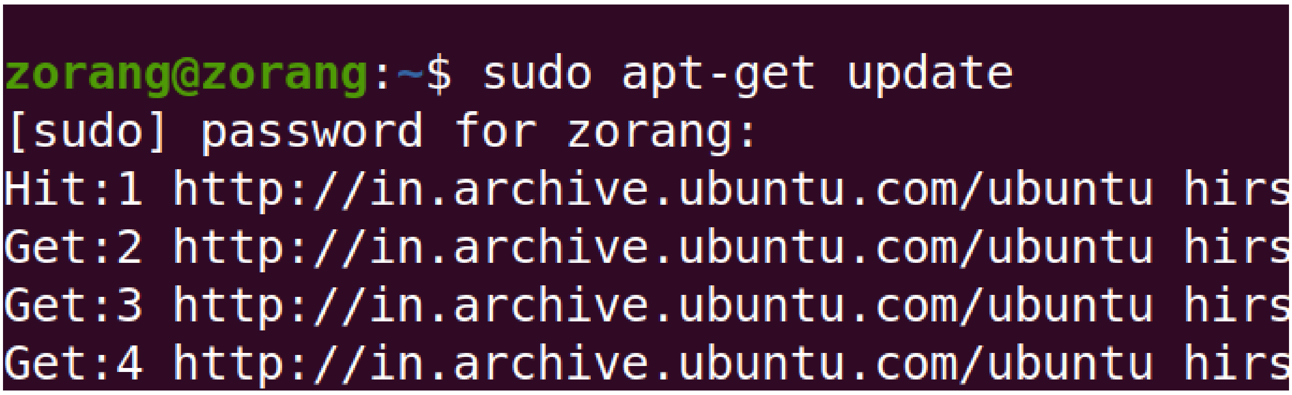
WordPress Installation

Step1 - First update your system repo

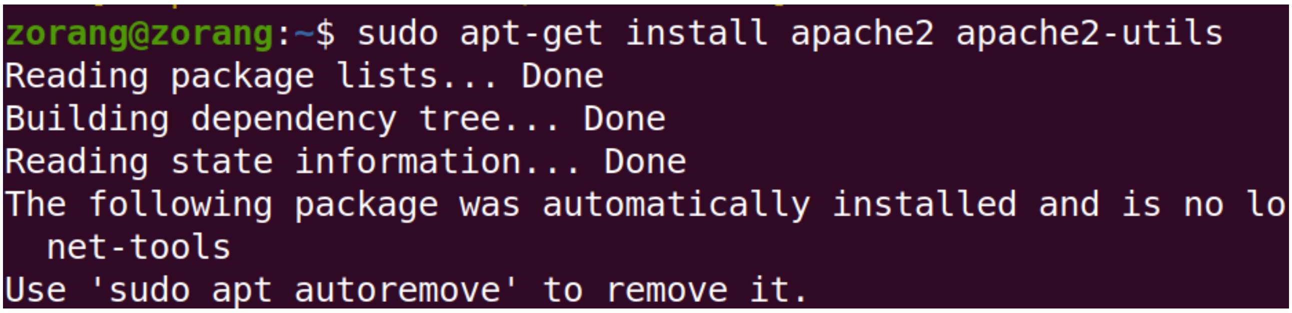
- Sudo apt-get update



```
zorang@zorang:~$ sudo apt-get update
[sudo] password for zorang:
Hit:1 http://in.archive.ubuntu.com/ubuntu hirs
Get:2 http://in.archive.ubuntu.com/ubuntu hirs
Get:3 http://in.archive.ubuntu.com/ubuntu hirs
Get:4 http://in.archive.ubuntu.com/ubuntu hirs
```

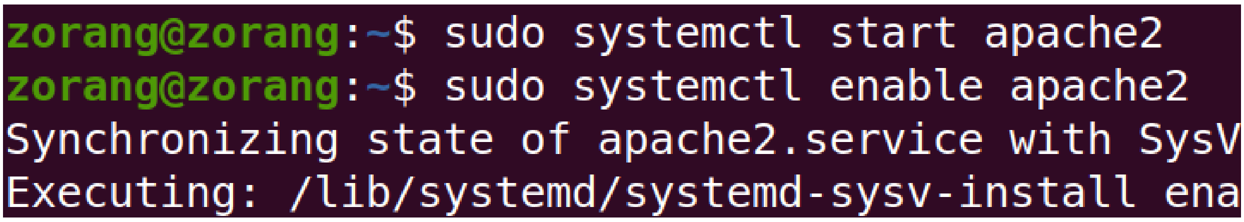
Step2 - Install apache2 and start and enable apache service

- `sudo apt-get install apache2 apache2-utils`



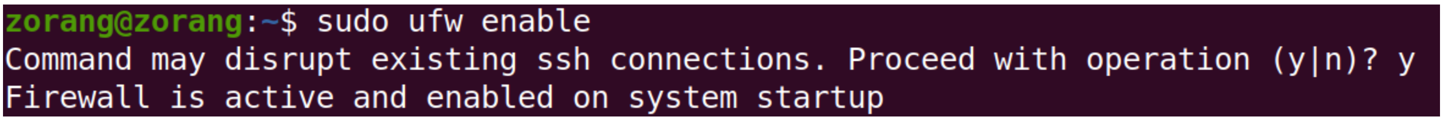
```
zorang@zorang:~$ sudo apt-get install apache2 apache2-utils
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no lo
  net-tools
Use 'sudo apt autoremove' to remove it.
```

- `sudo systemctl start apache2`
- `sudo systemctl enable apache2`



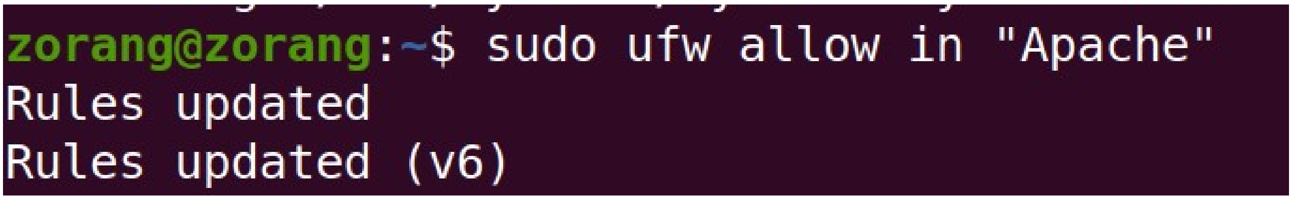
```
zorang@zorang:~$ sudo systemctl start apache2
zorang@zorang:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV
Executing: /lib/systemd/systemd-sysv-install ena
```

Step3 - Once you’ve started Apache, you then need to allow HTTP traffic on your UFW firewall



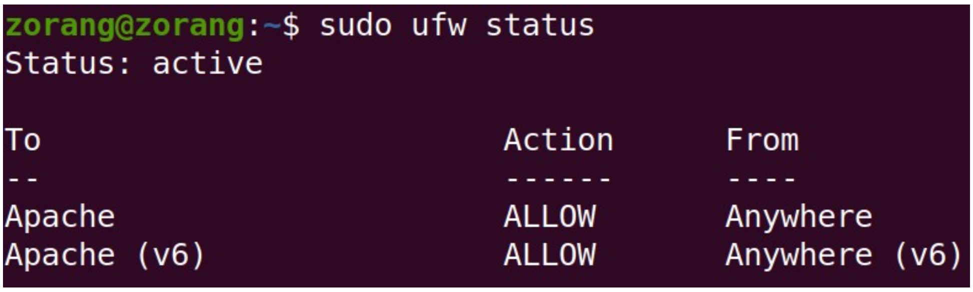
```
zorang@zorang:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
```

- sudo ufw allow in "Apache"



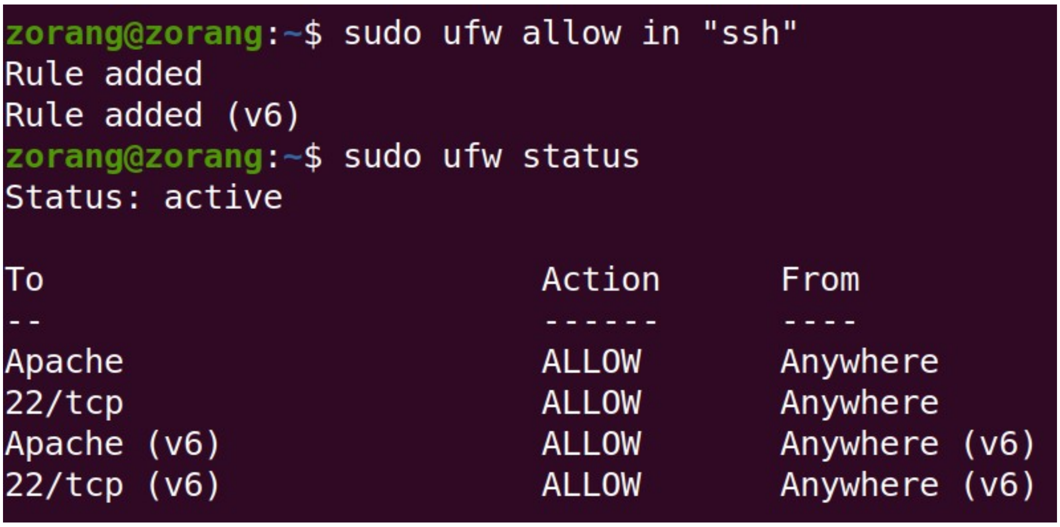
```
zorang@zorang:~$ sudo ufw allow in "Apache"
Rules updated
Rules updated (v6)
```

- sudo ufw status



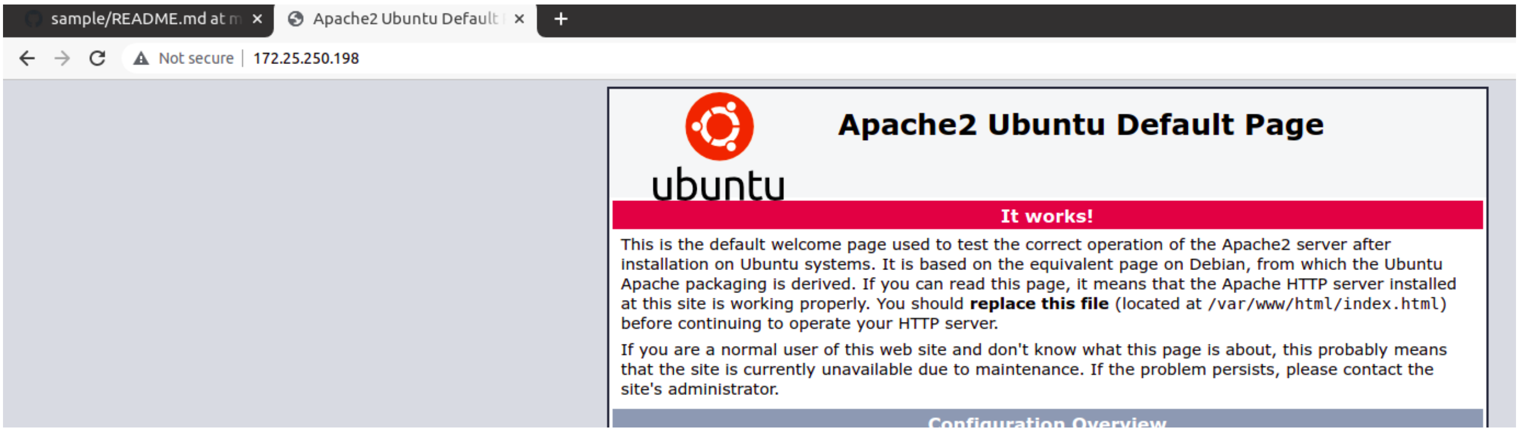
```
zorang@zorang:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
Apache                     ALLOW       Anywhere
Apache (v6)                ALLOW       Anywhere (v6)
```

Important - please make sure you allow ssh if your server is on virtual machine



```
zorang@zorang:~$ sudo ufw allow in "ssh"
Rule added
Rule added (v6)
zorang@zorang:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
Apache                     ALLOW       Anywhere
22/tcp                     ALLOW       Anywhere
Apache (v6)                ALLOW       Anywhere (v6)
22/tcp (v6)                ALLOW       Anywhere (v6)
```

To test whether the Apache server is running, open your web browser and enter the following URL in the address bar



Apache2 Ubuntu Default Page

It works!

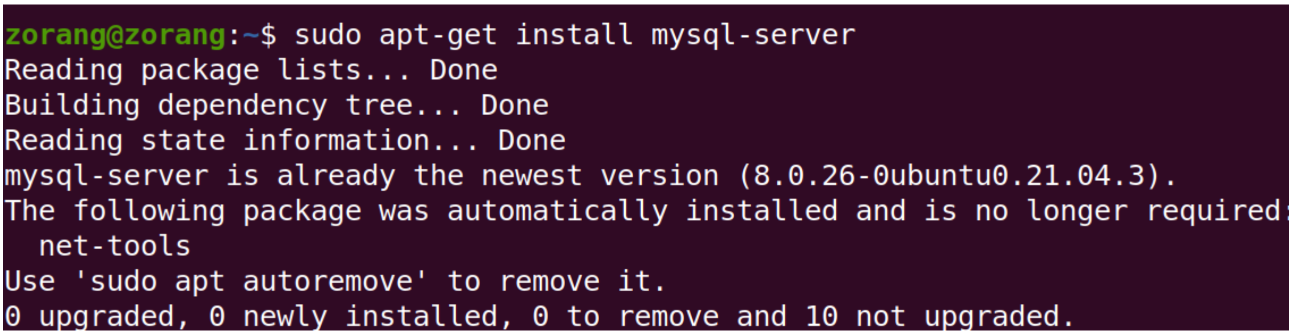
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Step4 - Now, Install mysql-server and set password for root

- `sudo apt-get install mysql-server`

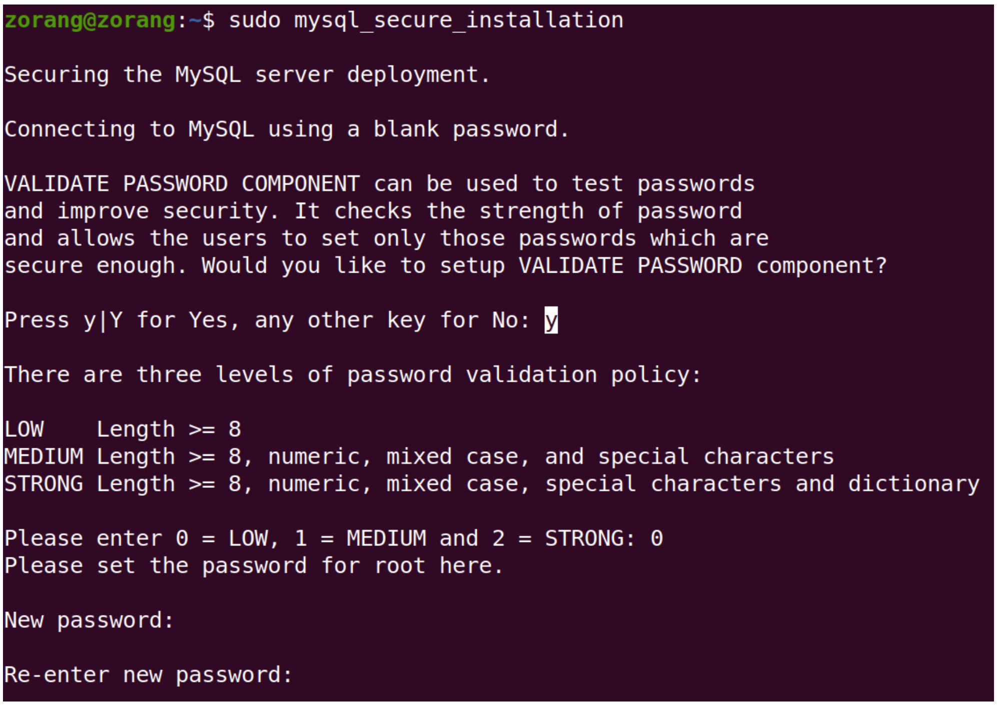


```
zorang@zorang:~$ sudo apt-get install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mysql-server is already the newest version (8.0.26-0ubuntu0.21.04.3).
The following package was automatically installed and is no longer required
  net-tools
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 10 not upgraded.
```

- `sudo mysql_secure_installation`

As shown in below image press Y to set root passsword

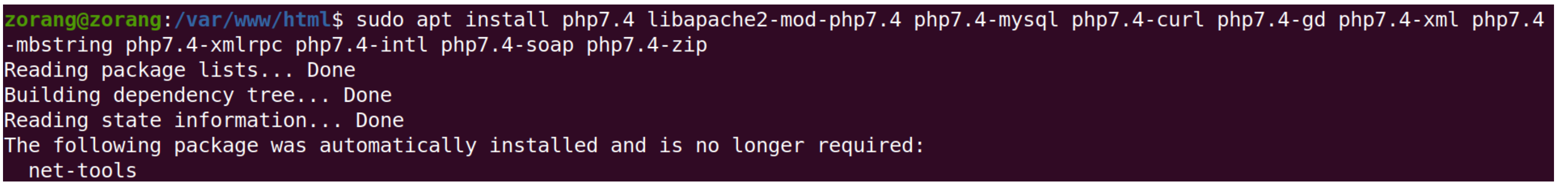
Choose password policy, and set root password like (Ritik@123456) and then press y for all.



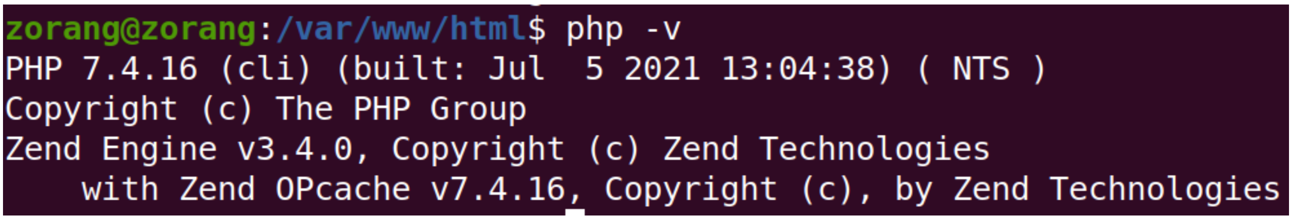
```
zorang@zorang:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y

There are three levels of password validation policy:

LOW    Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0
Please set the password for root here.

New password:

Re-enter new password:
```

Step5 - Install required php7.4 packages

- `sudo apt install php7.4 libapache2-mod-php7.4 php7.4-mysql php7.4-curl php7.4-gd php7.4-xml php7.4-mbstring php7.4-xmlrpc php7.4-intl php7.4-soap php7.4-zip`



```
zorang@zorang:/var/www/html$ sudo apt install php7.4 libapache2-mod-php7.4 php7.4-mysql php7.4-curl php7.4-gd php7.4-xml php7.4
-mbstring php7.4-xmlrpc php7.4-intl php7.4-soap php7.4-zip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  net-tools
```



```
zorang@zorang:/var/www/html$ php -v
PHP 7.4.16 (cli) (built: Jul  5 2021 13:04:38) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.16, Copyright (c), by Zend Technologies
```

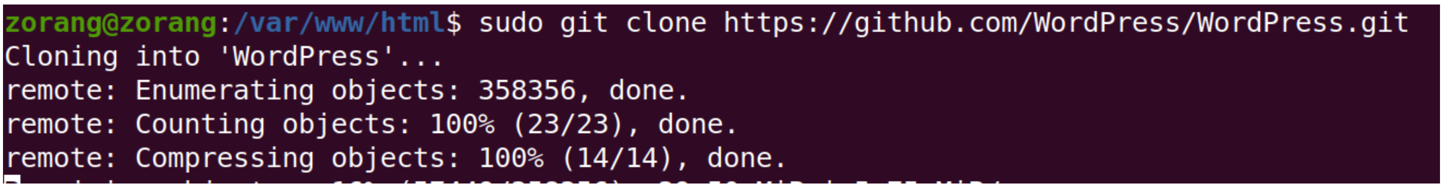
Step6 - Download the latest version of the WordPress from GitHub

1. First navigate to html dir

- `cd /var/www/html`

2. Clone WordPress repo from GitHub

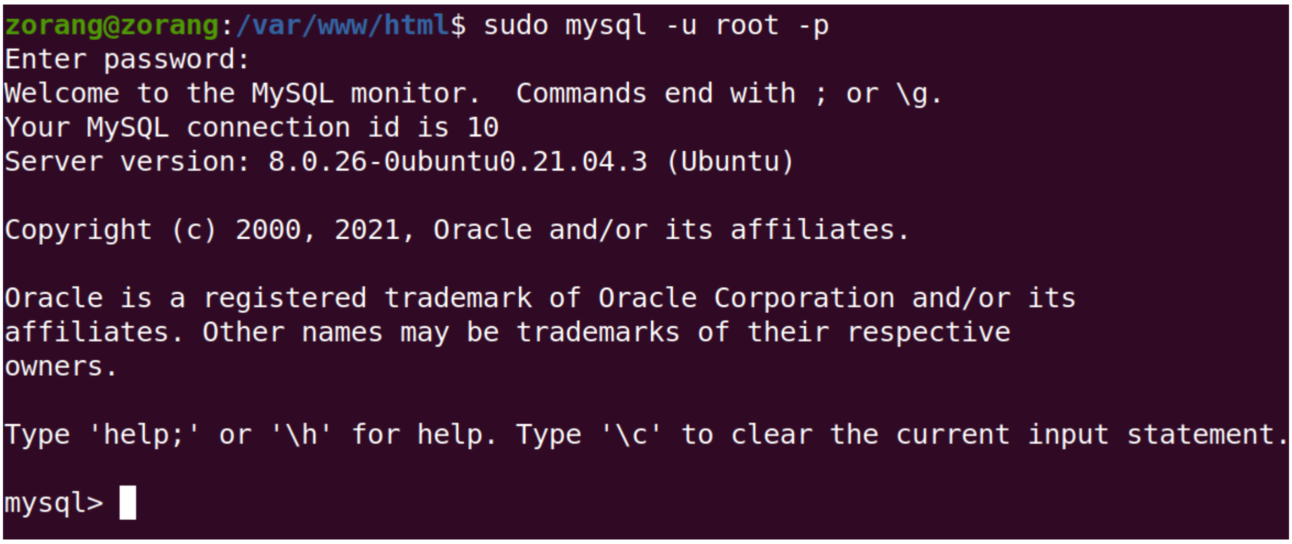
- `sudo git clone https://github.com/WordPress/WordPress.git`
- `sudo mv WordPress wordpress`



```
zorang@zorang:/var/www/html$ sudo git clone https://github.com/WordPress/WordPress.git
Cloning into 'WordPress'...
remote: Enumerating objects: 358356, done.
remote: Counting objects: 100% (23/23), done.
remote: Compressing objects: 100% (14/14), done.
```

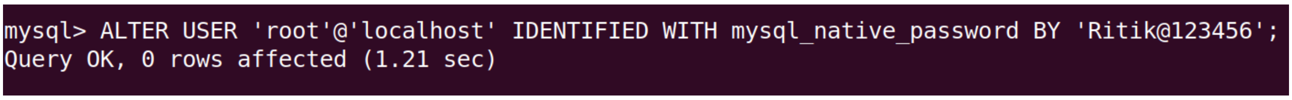
Step7 - Create user for wordpress and remember its password and give it permission to create database

- `sudo mysql -u root -p`



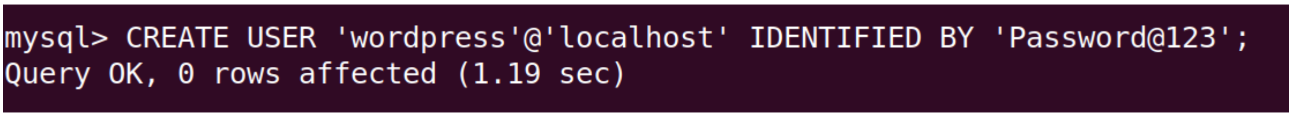
```
zorang@zorang:/var/www/html$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.26-0ubuntu0.21.04.3 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

- ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'Ritik@123456';



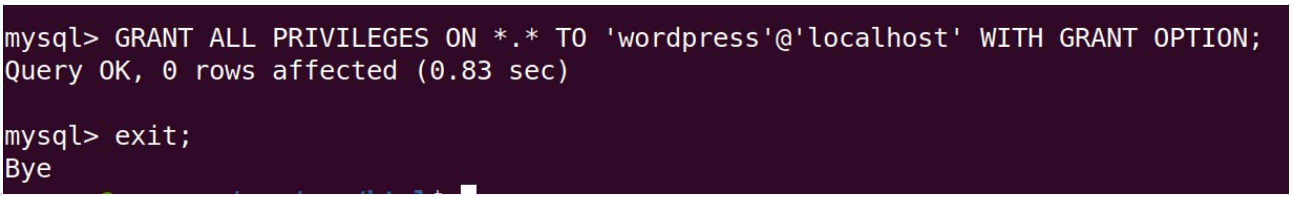
```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'Ritik@123456';
Query OK, 0 rows affected (1.21 sec)
```

- CREATE USER 'wordpress'@'localhost' IDENTIFIED BY 'Password@123';



```
mysql> CREATE USER 'wordpress'@'localhost' IDENTIFIED BY 'Password@123';
Query OK, 0 rows affected (1.19 sec)
```

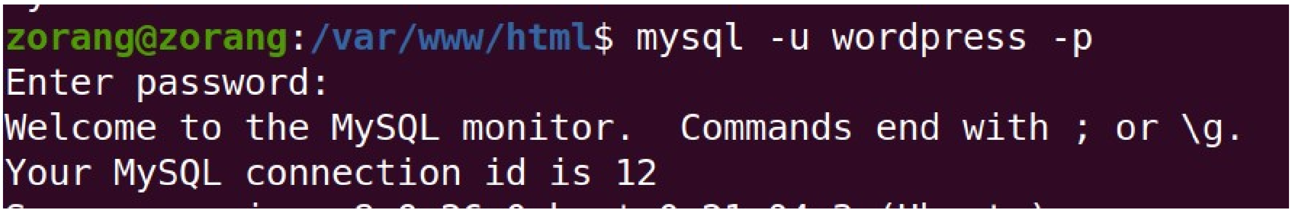
- GRANT ALL PRIVILEGES ON *.* TO 'wordpress'@'localhost' WITH GRANT OPTION;



```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'wordpress'@'localhost' WITH GRANT OPTION;
Query OK, 0 rows affected (0.83 sec)

mysql> exit;
Bye
```

Step8 - Now Create DataBase for wordpress

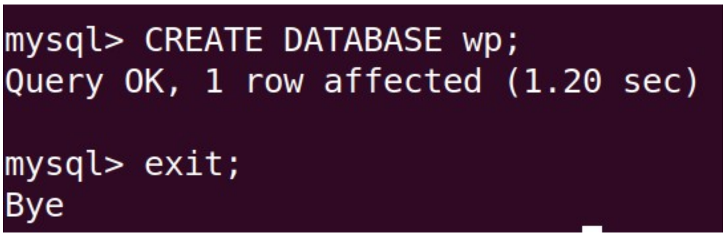
- mysql -u wordpress -p



```
zorang@zorang:/var/www/html$ mysql -u wordpress -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
```

Enter Password - Password@123 as above we given this password for Wordpress user

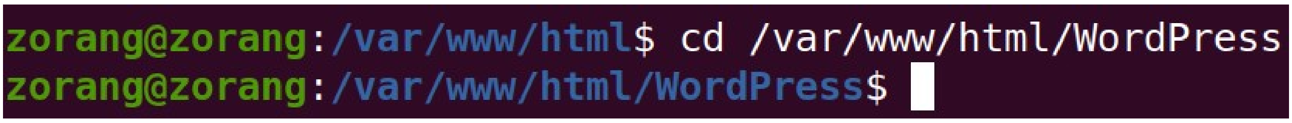
- CREATE DATABASE wp;



```
mysql> CREATE DATABASE wp;
Query OK, 1 row affected (1.20 sec)

mysql> exit;
Bye
```

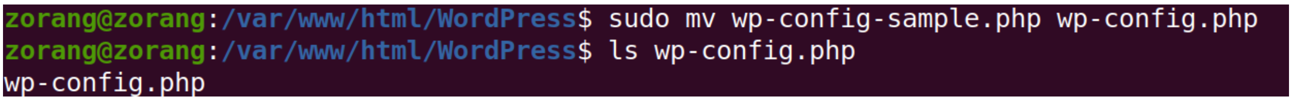
Step9 - Move sample file to activate it

- `cd /var/www/html/wordpress`



```
zorang@zorang:/var/www/html$ cd /var/www/html/WordPress
zorang@zorang:/var/www/html/WordPress$ 
```

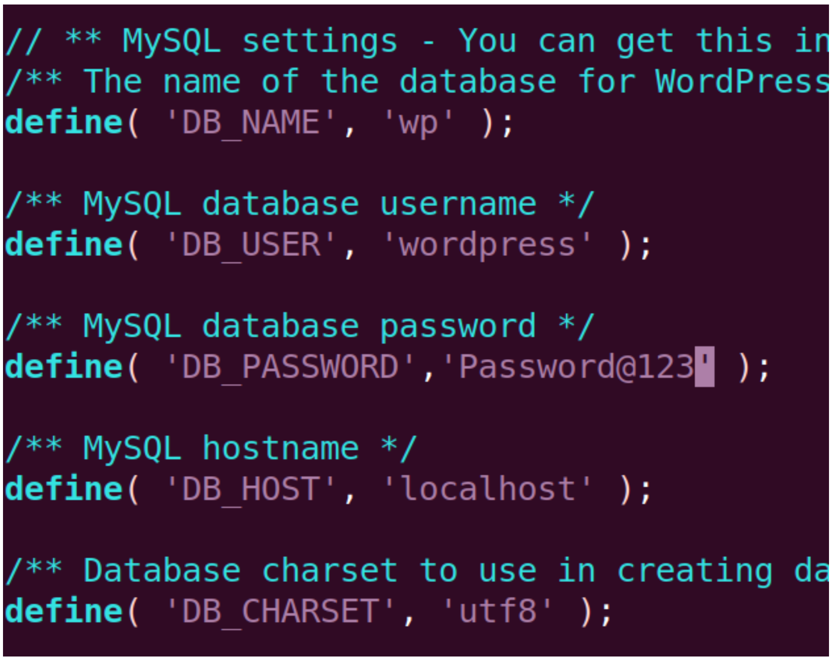
- `sudo mv wp-config-sample.php wp-config.php`



```
zorang@zorang:/var/www/html/WordPress$ sudo mv wp-config-sample.php wp-config.php
zorang@zorang:/var/www/html/WordPress$ ls wp-config.php
wp-config.php
```

Step10 - Now set DataBase User, DataBase Name and DataBase Password in wp-config.php file

- `sudo vim wp-config.php`



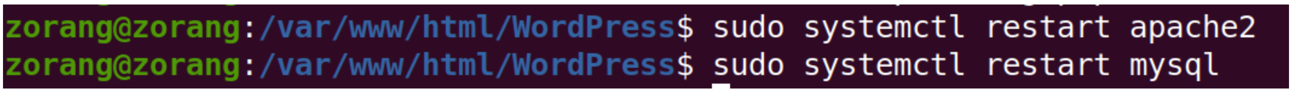
```
// ** MySQL settings - You can get this in
/** The name of the database for WordPress
define( 'DB_NAME', 'wp' );

/** MySQL database username */
define( 'DB_USER', 'wordpress' );

/** MySQL database password */
define( 'DB_PASSWORD','Password@123' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating da
define( 'DB_CHARSET', 'utf8' );
```

Give db_name = wp
Give db_user = wordpress
Give db_password = Password@123

Warning : Do not share your wp-config.php as this file contain you db password

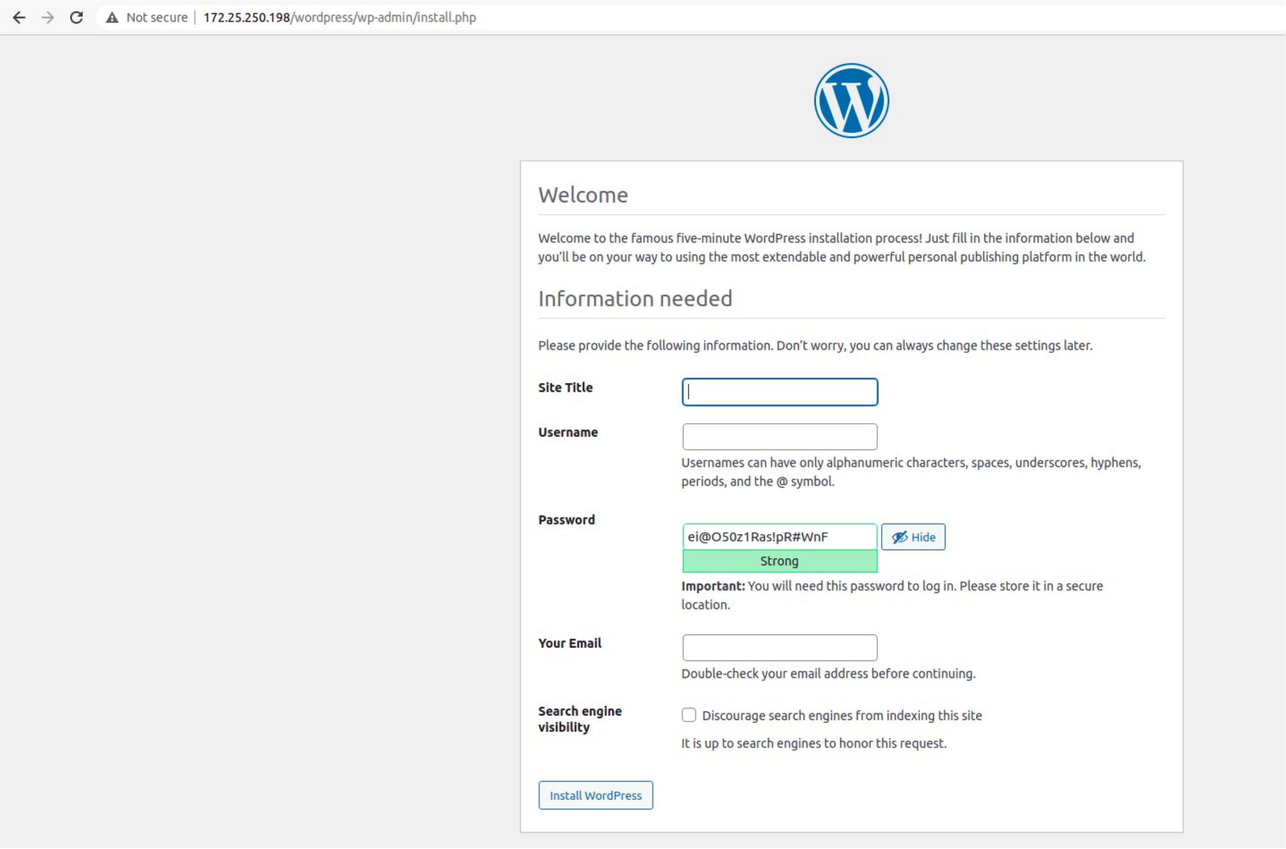
Step11 - Restart Apache and MySql service

- sudo systemctl restart apache2
- sudo systemctl restart mysql



```
zorang@zorang:/var/www/html/WordPress$ sudo systemctl restart apache2
zorang@zorang:/var/www/html/WordPress$ sudo systemctl restart mysql
```

Step12 - Go to Browser and type your server ip or server domain name



172.25.250.198/wordpress/wp-admin/install.php

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title	
Username	Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.
Password	ei@O50z1Ras!pR#WnF Hide Strong **Important:** You will need this password to log in. Please store it in a secure location.
Your Email	Double-check your email address before continuing.
Search engine visibility	□ Discourage search engines from indexing this site It is up to search engines to honor this request.

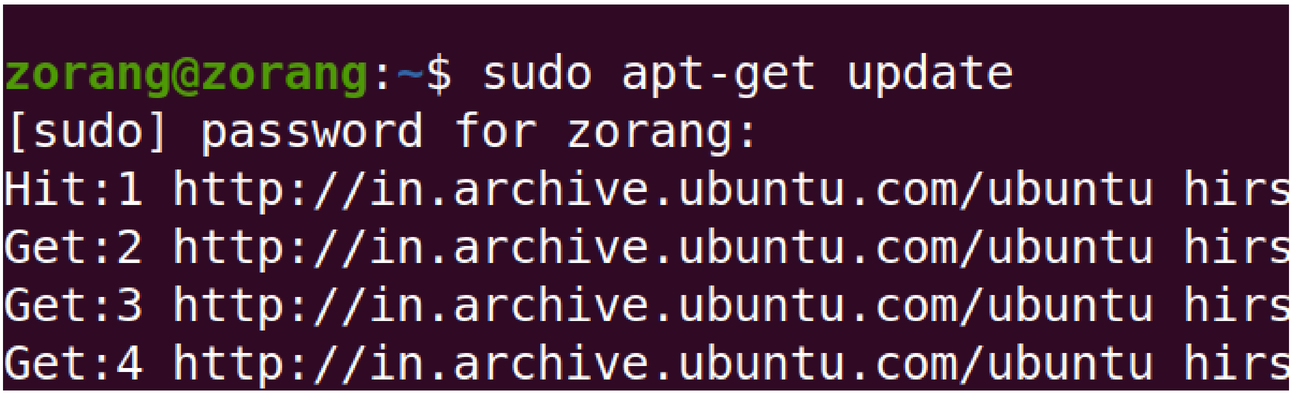
You can set below details on your WordPress installation Setup

```
Site Title = yoursitename
Username =  your-username
Password = StrongPassword
Email    = your.email@wordpress.com
```

Laravel Installation

Step1 - First update your system repo

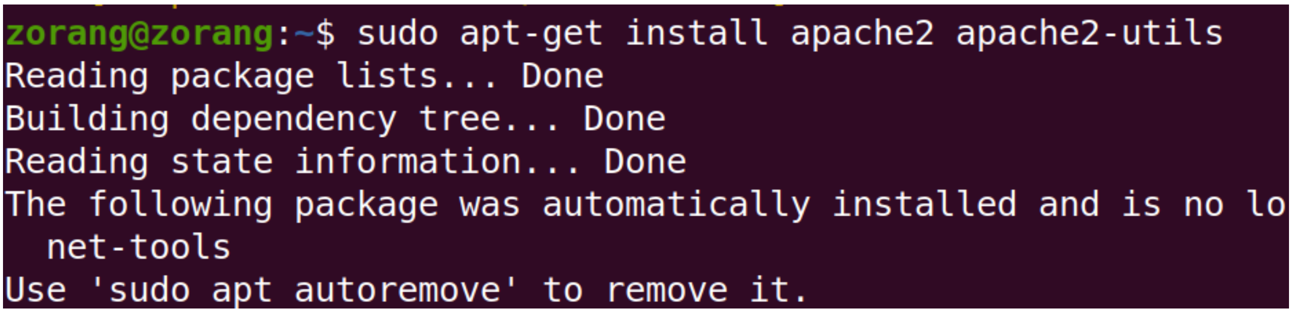
- Sudo apt-get update



```
zorang@zorang:~$ sudo apt-get update
[sudo] password for zorang:
Hit:1 http://in.archive.ubuntu.com/ubuntu hirs
Get:2 http://in.archive.ubuntu.com/ubuntu hirs
Get:3 http://in.archive.ubuntu.com/ubuntu hirs
Get:4 http://in.archive.ubuntu.com/ubuntu hirs
```

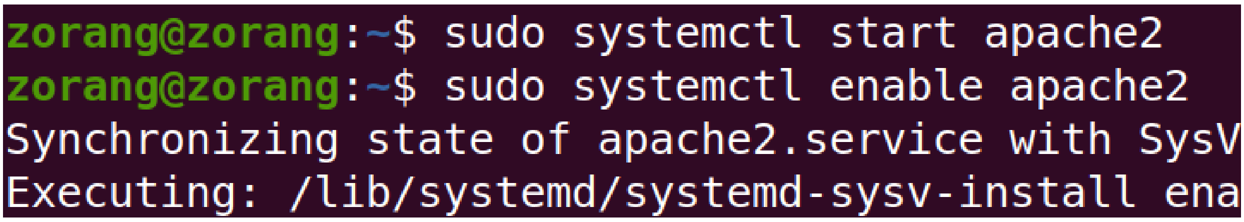
Step2 - Install apache2 and start and enable apache service

- `sudo apt-get install apache2 apache2-utils`



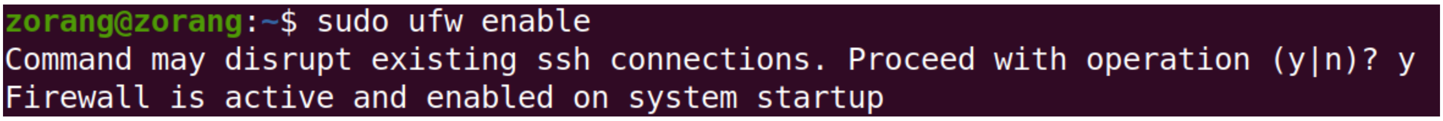
```
zorang@zorang:~$ sudo apt-get install apache2 apache2-utils
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no lo
  net-tools
Use 'sudo apt autoremove' to remove it.
```

- `sudo systemctl start apache2`
- `sudo systemctl enable apache2`



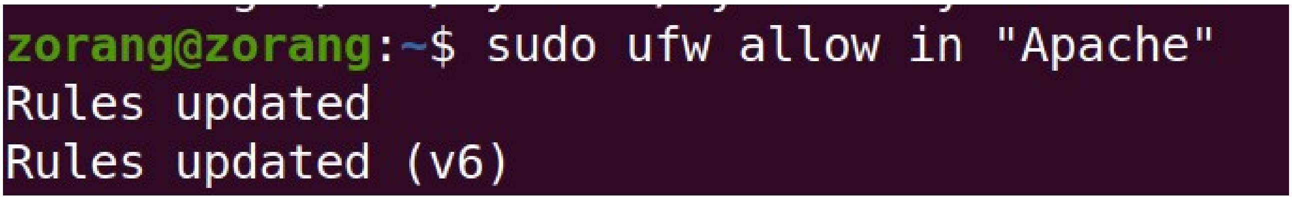
```
zorang@zorang:~$ sudo systemctl start apache2
zorang@zorang:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV
Executing: /lib/systemd/systemd-sysv-install ena
```

Step3 - Once you’ve started Apache, you then need to allow HTTP traffic on your UFW firewall



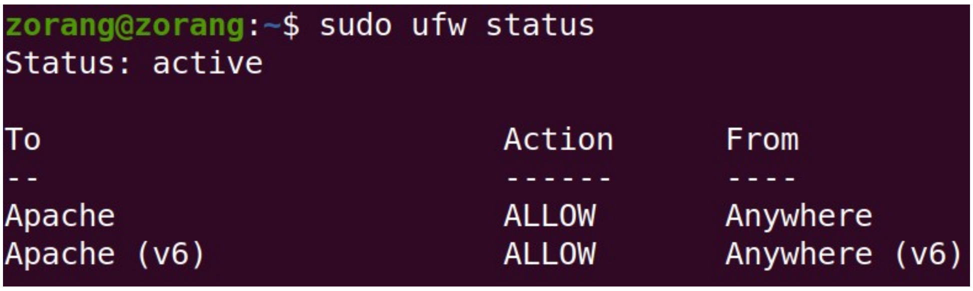
```
zorang@zorang:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
```

- `sudo ufw allow in "Apache"`



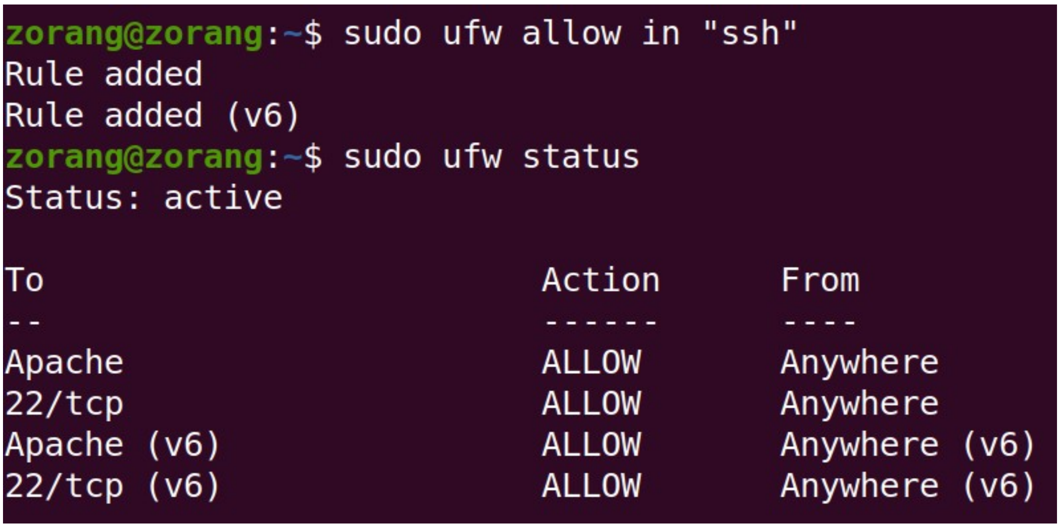
```
zorang@zorang:~$ sudo ufw allow in "Apache"
Rules updated
Rules updated (v6)
```

- `sudo ufw status`



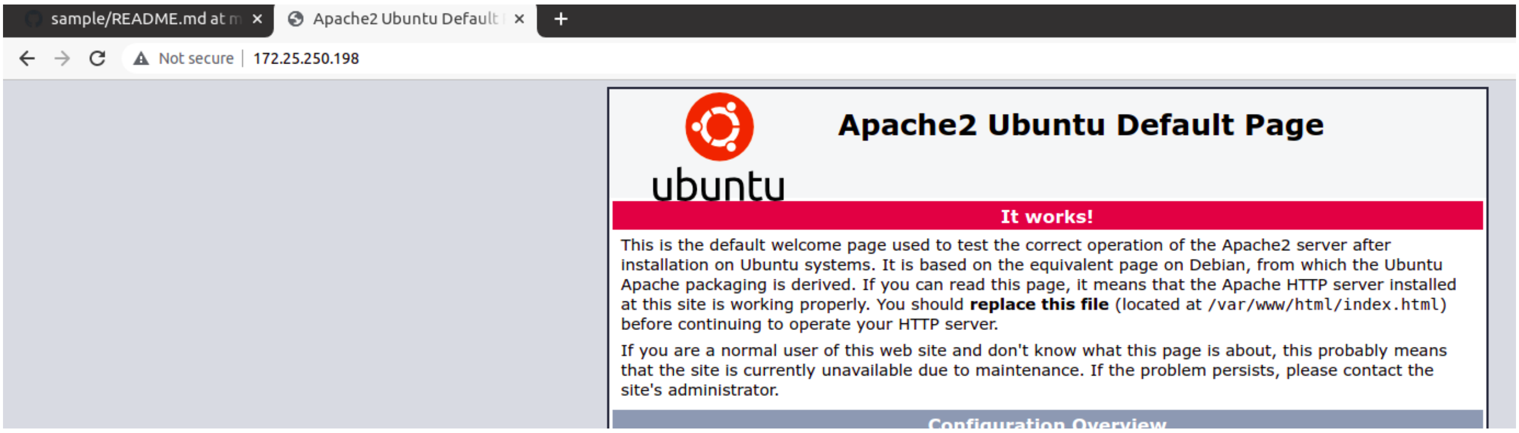
```
zorang@zorang:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
Apache                     ALLOW       Anywhere
Apache (v6)                ALLOW       Anywhere (v6)
```

Important - please make sure you allow ssh if your server is on virtual machine



```
zorang@zorang:~$ sudo ufw allow in "ssh"
Rule added
Rule added (v6)
zorang@zorang:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
Apache                     ALLOW       Anywhere
22/tcp                     ALLOW       Anywhere
Apache (v6)                ALLOW       Anywhere (v6)
22/tcp (v6)                ALLOW       Anywhere (v6)
```

To test whether the Apache server is running, open your web browser and enter the following URL in the address bar



Apache2 Ubuntu Default Page

It works!

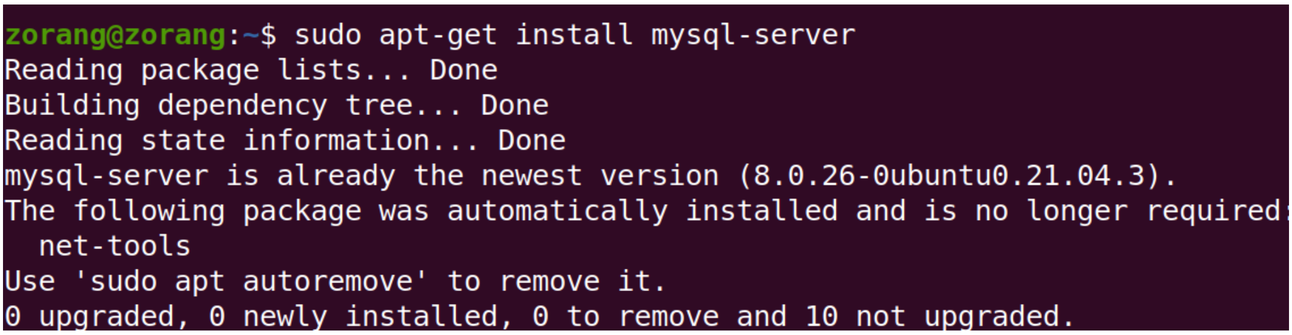
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Step4 - Now, Install mysql-server and set password for root

- `sudo apt-get install mysql-server`

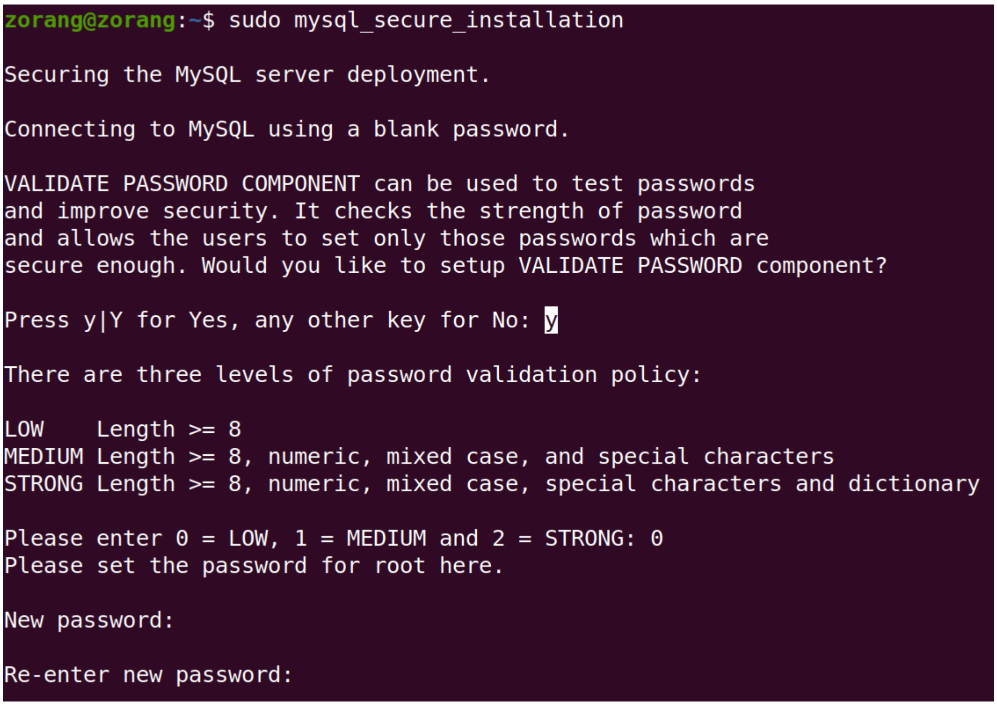


```
zorang@zorang:~$ sudo apt-get install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mysql-server is already the newest version (8.0.26-0ubuntu0.21.04.3).
The following package was automatically installed and is no longer required:
  net-tools
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 10 not upgraded.
```

- `sudo mysql_secure_installation`

As shown in below image press Y to set root passsword

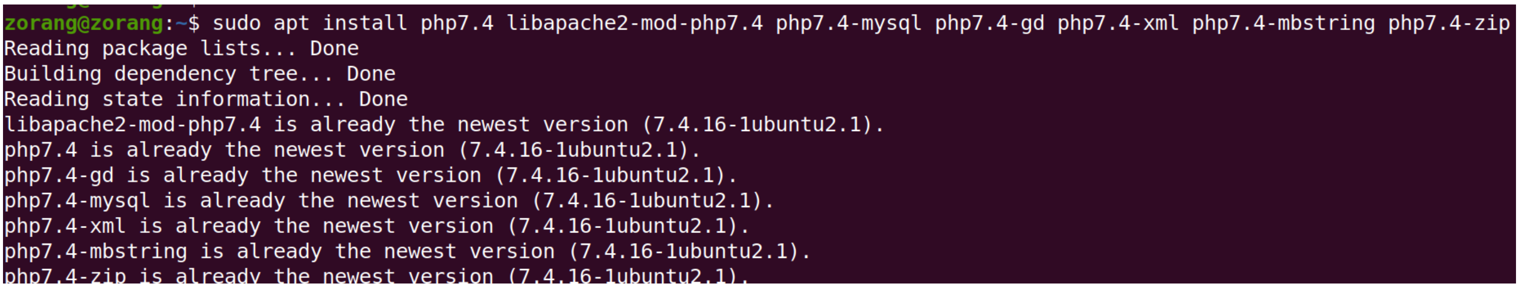
Choose password policy, and set root password like (Ritik@123456) and then press y for all.



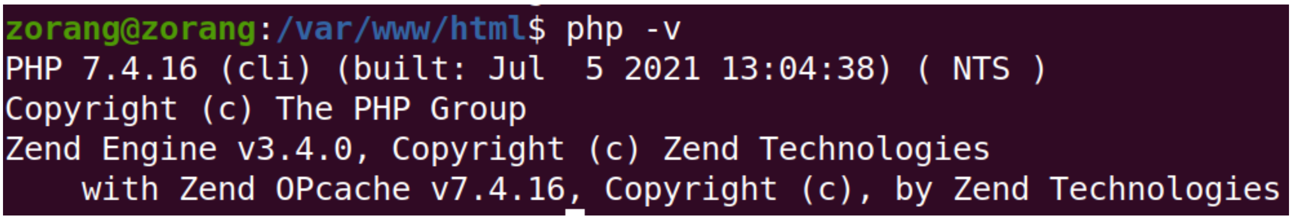
```
zorang@zorang:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y

There are three levels of password validation policy:

LOW    Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0
Please set the password for root here.

New password:

Re-enter new password:
```

Step5 - Install required php7.4 packages

- `sudo apt install php7.4 libapache2-mod-php7.4 php7.4-mysql php7.4-gd php7.4-xml php7.4-mbstring php7.4-zip`



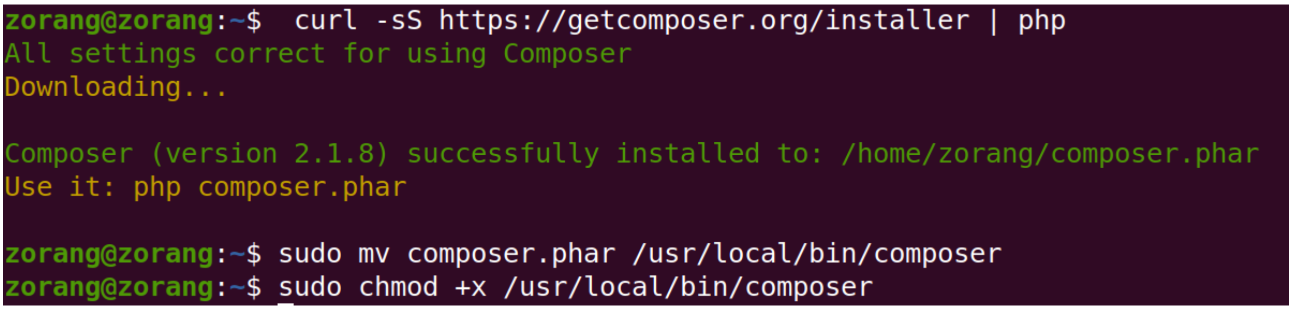
```
zorang@zorang:~$ sudo apt install php7.4 libapache2-mod-php7.4 php7.4-mysql php7.4-gd php7.4-xml php7.4-mbstring php7.4-zip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libapache2-mod-php7.4 is already the newest version (7.4.16-1ubuntu2.1).
php7.4 is already the newest version (7.4.16-1ubuntu2.1).
php7.4-gd is already the newest version (7.4.16-1ubuntu2.1).
php7.4-mysql is already the newest version (7.4.16-1ubuntu2.1).
php7.4-xml is already the newest version (7.4.16-1ubuntu2.1).
php7.4-mbstring is already the newest version (7.4.16-1ubuntu2.1).
php7.4-zip is already the newest version (7.4.16-1ubuntu2.1).
```



```
zorang@zorang:/var/www/html$ php -v
PHP 7.4.16 (cli) (built: Jul  5 2021 13:04:38) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.16, Copyright (c), by Zend Technologies
```

Step6 - Install composer (required for managing dependencies, libraries for laravel)

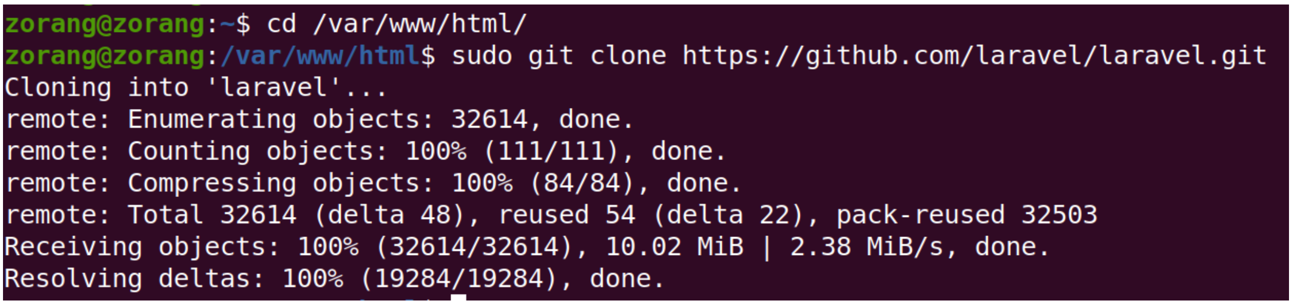
- `curl -sS https://getcomposer.org/installer | php`
- `curl -sS https://getcomposer.org/installer | php`
- `sudo chmod +x /usr/local/bin/composer`



```
zorang@zorang:~$  curl -sS https://getcomposer.org/installer | php
All settings correct for using Composer
Downloading...

Composer (version 2.1.8) successfully installed to: /home/zorang/composer.phar
Use it: php composer.phar

zorang@zorang:~$ sudo mv composer.phar /usr/local/bin/composer
zorang@zorang:~$ sudo chmod +x /usr/local/bin/composer
```

Step7 - Install the latest version of laravel

- `cd /var/www/html/`
- `sudo git clone https://github.com/laravel/laravel.git`



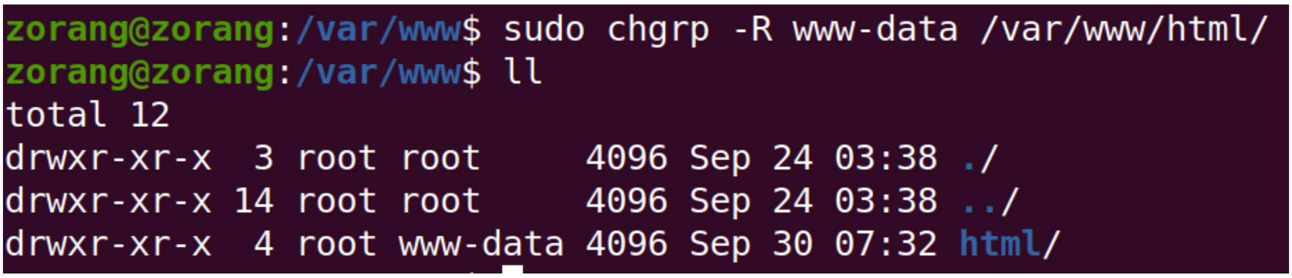
```
zorang@zorang:~$ cd /var/www/html/
zorang@zorang:/var/www/html$ sudo git clone https://github.com/laravel/laravel.git
Cloning into 'laravel'...
remote: Enumerating objects: 32614, done.
remote: Counting objects: 100% (111/111), done.
remote: Compressing objects: 100% (84/84), done.
remote: Total 32614 (delta 48), reused 54 (delta 22), pack-reused 32503
Receiving objects: 100% (32614/32614), 10.02 MiB | 2.38 MiB/s, done.
Resolving deltas: 100% (19284/19284), done.
```

or to install particular version go to laravel github profile

https://github.com/laravel/laravel/releases

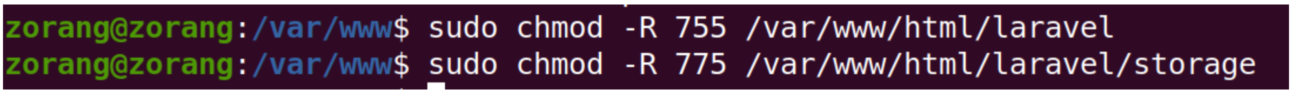
Step8 - Give proper permissions to html dir and laravel dir

- `sudo chgrp -R www-data /var/www/html/`



```
zorang@zorang:/var/www$ sudo chgrp -R www-data /var/www/html/
zorang@zorang:/var/www$ ll
total 12
drwxr-xr-x  3 root root      4096 Sep 24 03:38 ./
drwxr-xr-x 14 root root      4096 Sep 24 03:38 ../
drwxr-xr-x  4 root www-data 4096 Sep 30 07:32 html/
```

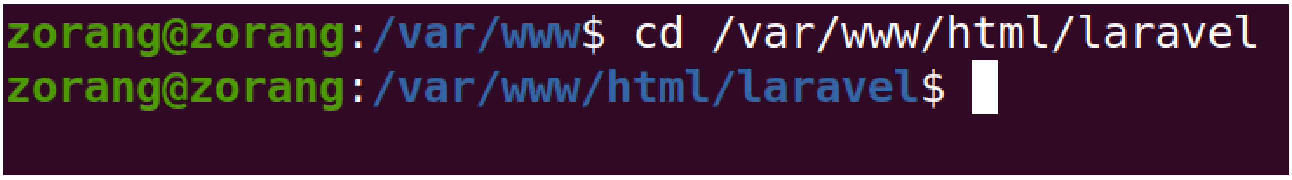
- `sudo chmod -R 755 /var/www/html/laravel`
- ` sudo chmod -R 775 /var/www/html/laravel/storage`



```
zorang@zorang:/var/www$ sudo chmod -R 755 /var/www/html/laravel
zorang@zorang:/var/www$ sudo chmod -R 775 /var/www/html/laravel/storage
```

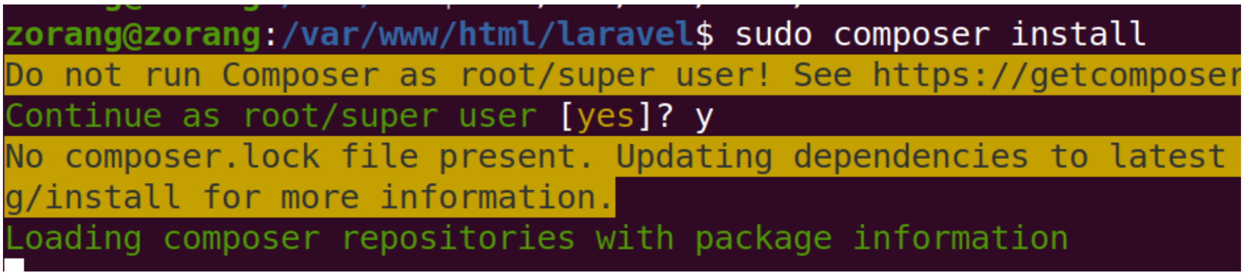
Step9 - Run composer install to downloads and installs all the libraries and dependencies outlined in laravel dir

- `cd /var/www/html/laravel`



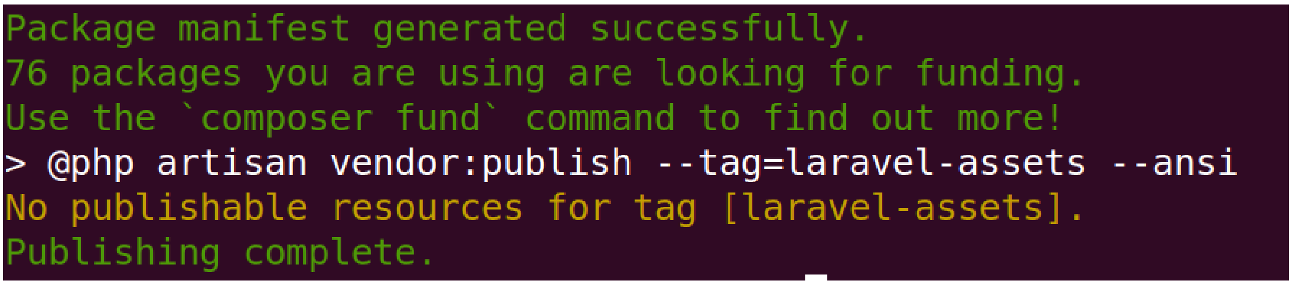
```
zorang@zorang:/var/www$ cd /var/www/html/laravel
zorang@zorang:/var/www/html/laravel$ 
```

- `sudo composer install`



```
zorang@zorang:/var/www/html/laravel$ sudo composer install
Do not run Composer as root/super user! See https://getcomposer
Continue as root/super user [yes]? y
No composer.lock file present. Updating dependencies to latest
g/install for more information.
Loading composer repositories with package information
```

Wait untill composer install dependencies, libraries for laravel application and after some time it shows complete.

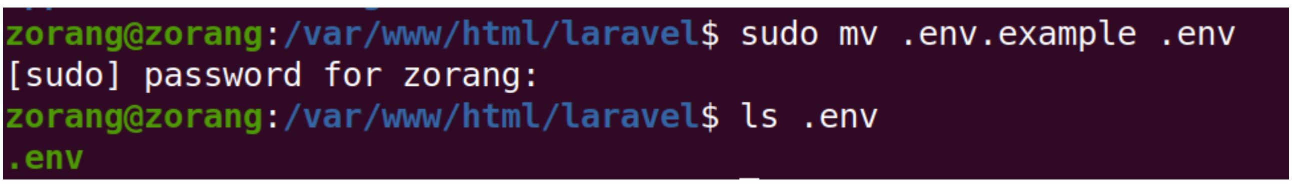


```
Package manifest generated successfully.
76 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi
No publishable resources for tag [laravel-assets].
Publishing complete.
```

Step10 - Move .env.example file to .env (It is a environment file to define things such as database connection settings, debug options, application URL, among other items that may vary depending on which environment the application is running.)

Warning - The environment configuration file contains sensitive information about your server, including database credentials and security keys. For that reason, you should never share this file publicly.

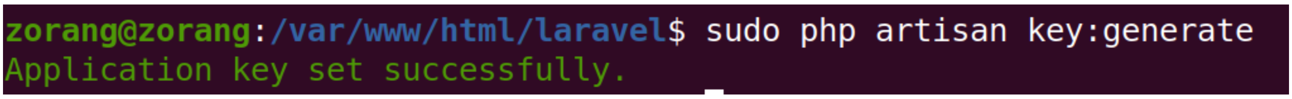
- `sudo mv .env.example .env`



```
zorang@zorang:/var/www/html/laravel$ sudo mv .env.example .env
[sudo] password for zorang:
zorang@zorang:/var/www/html/laravel$ ls .env
.env
```

Step11 - Now generate base64 random number encryption key, which used by the illuminate encrypter service.

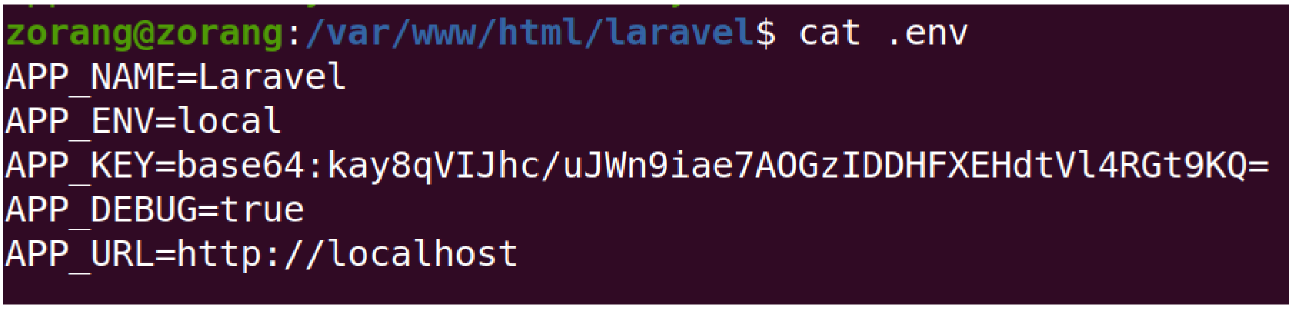
- `sudo php artisan key:generate`



```
zorang@zorang:/var/www/html/laravel$ sudo php artisan key:generate
Application key set successfully.
```

Step12 - Check the app_key get base64 ency. or not and you can also change the APP_NAME with the name of your application and APP_URL to the URL you need to access your Laravel application.

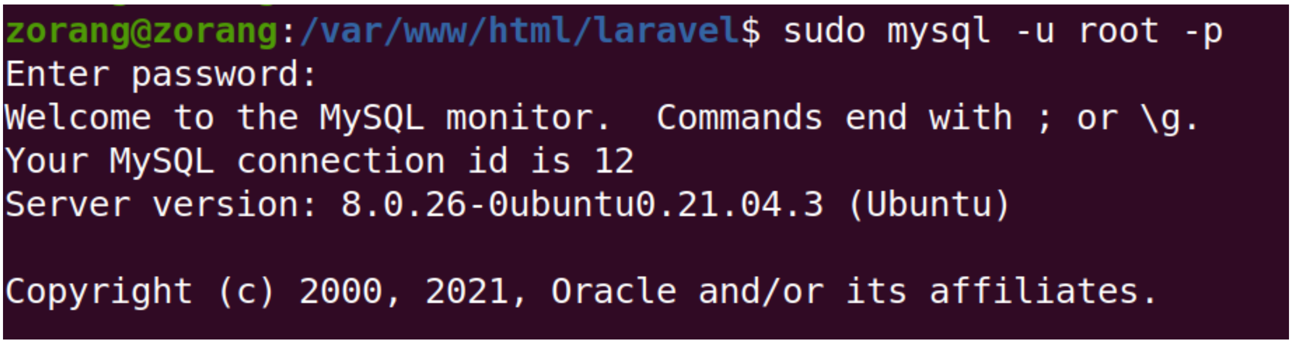
- `cat .env`



```
zorang@zorang:/var/www/html/laravel$ cat .env
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:kay8qVIJhc/uJWn9iae7AOGzIDDHFXEHdtVl4RGt9KQ=
APP_DEBUG=true
APP_URL=http://localhost
```

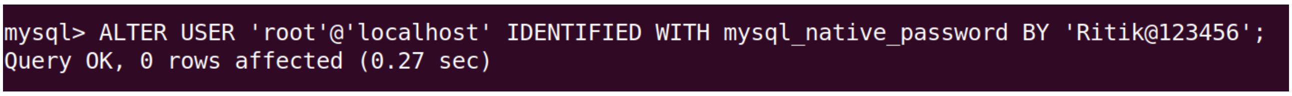
Step13 - Create user and database for laravel

- `sudo mysql -u root -p`



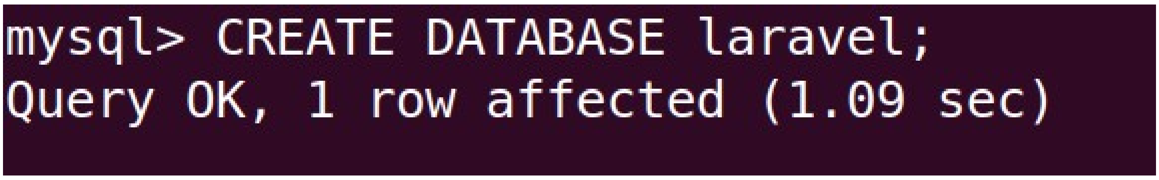
```
zorang@zorang:/var/www/html/laravel$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.26-0ubuntu0.21.04.3 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.
```

- `ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'Ritik@123456';`



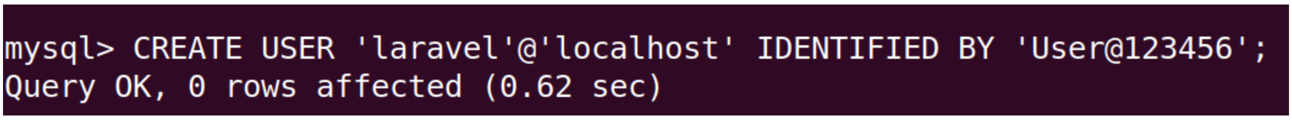
```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'Ritik@123456';
Query OK, 0 rows affected (0.27 sec)
```

- `CREATE DATABASE laravel;`



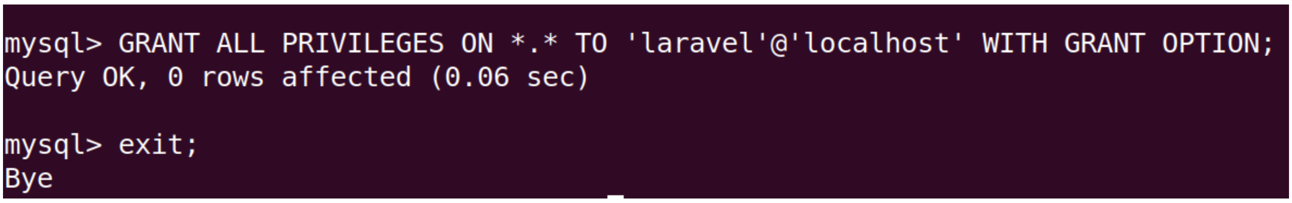
```
mysql> CREATE DATABASE laravel;
Query OK, 1 row affected (1.09 sec)
```

- `CREATE USER 'laravel'@'localhost' IDENTIFIED BY 'User@123456';`



```
mysql> CREATE USER 'laravel'@'localhost' IDENTIFIED BY 'User@123456';
Query OK, 0 rows affected (0.62 sec)
```

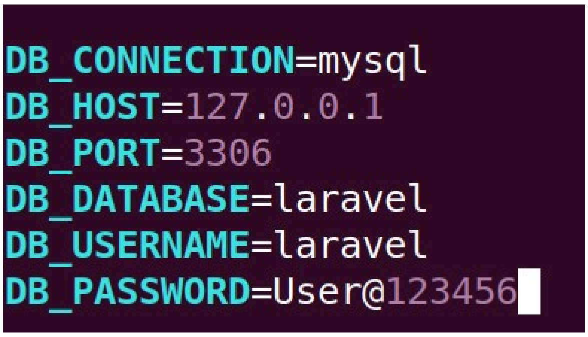
- `GRANT ALL PRIVILEGES ON *.* TO 'laravel'@'localhost' WITH GRANT OPTION;`



```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'laravel'@'localhost' WITH GRANT OPTION;
Query OK, 0 rows affected (0.06 sec)

mysql> exit;
Bye
```

Step14 - Now edit the .env file and update database settings

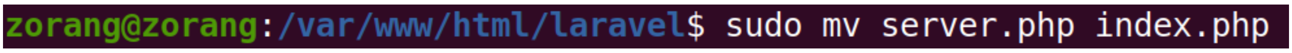
- `sudo vim .env`



```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel
DB_USERNAME=laravel
DB_PASSWORD=User@123456
```

Step15 - Move Server.php to index.php

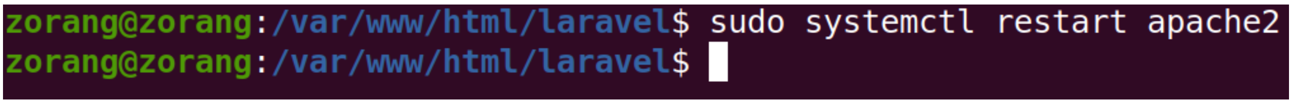
- `mv server.php index.php`



```
zorang@zorang:/var/www/html/laravel$ sudo mv server.php index.php
```

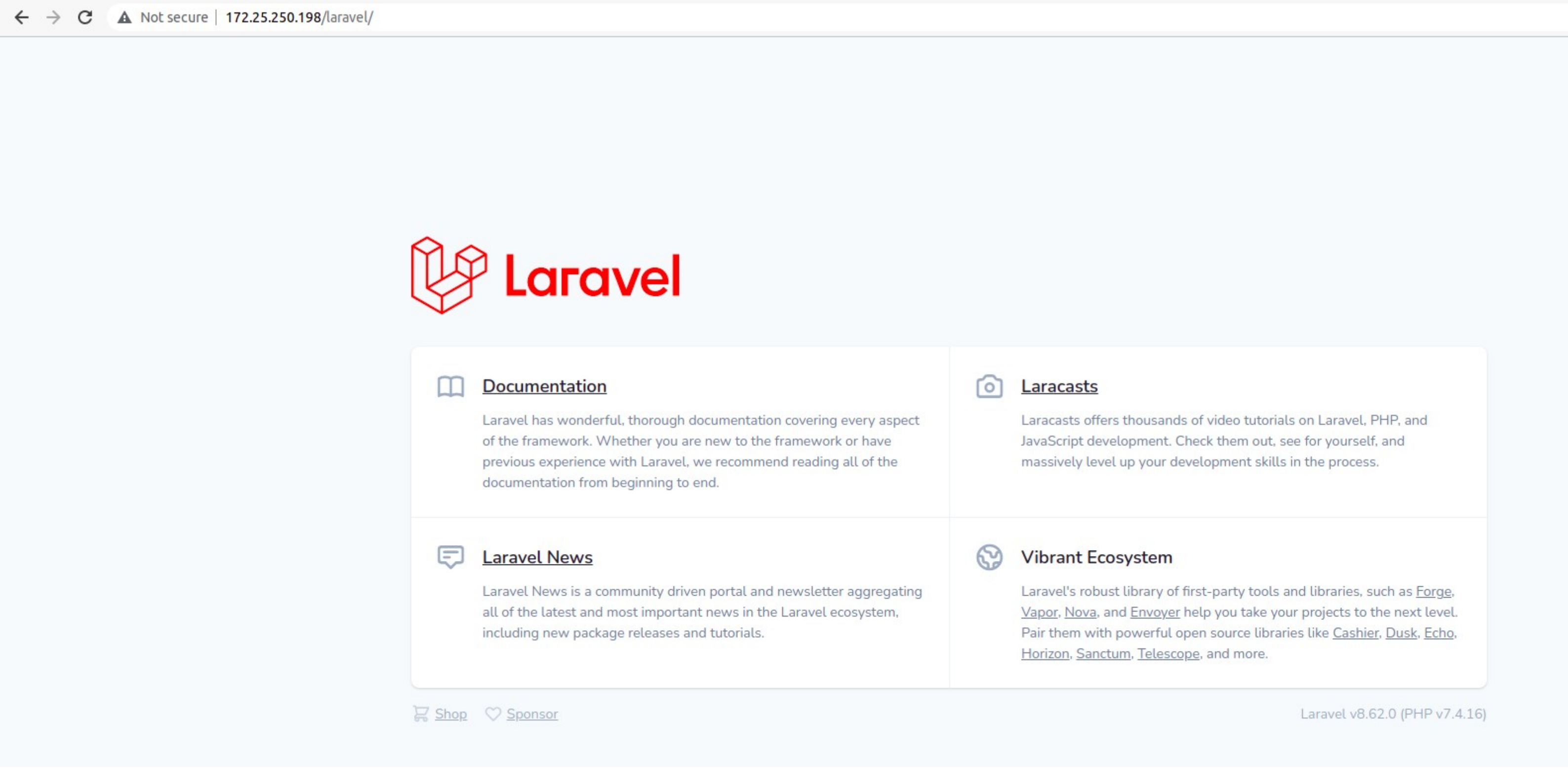
Step16 - Restart apache service

- `sudo systemctl restart apache2`



```
zorang@zorang:/var/www/html/laravel$ sudo systemctl restart apache2
zorang@zorang:/var/www/html/laravel$ 
```

Step17 - Go to browser and type your server ip/laravel or http://Serverdomain/laravel



Documentation

Laravel has wonderful, thorough documentation covering every aspect of the framework. Whether you are new to the framework or have previous experience with Laravel, we recommend reading all of the documentation from beginning to end.

Laracasts

Laracasts offers thousands of video tutorials on Laravel, PHP, and JavaScript development. Check them out, see for yourself, and massively level up your development skills in the process.

Laravel News

Laravel News is a community driven portal and newsletter aggregating all of the latest and most important news in the Laravel ecosystem, including new package releases and tutorials.

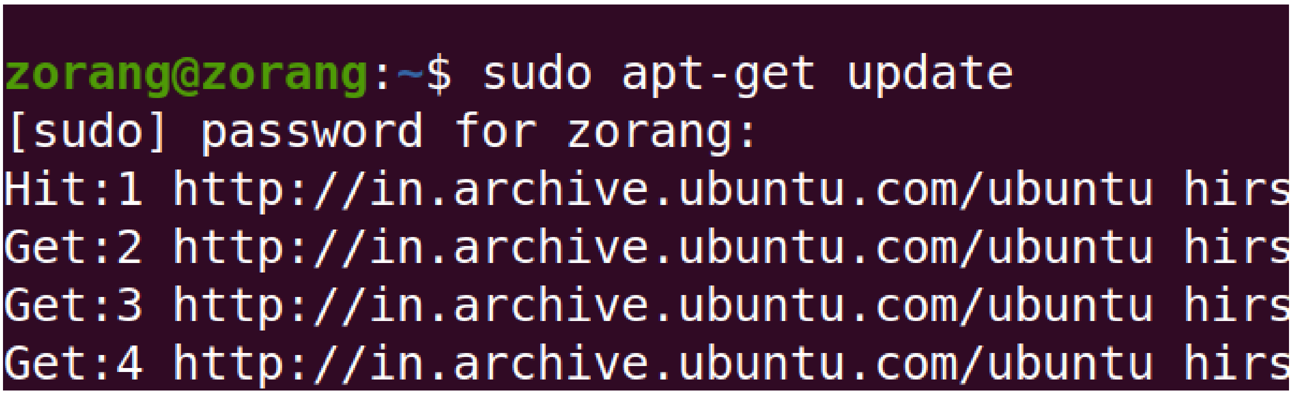
Vibrant Ecosystem

Laravel's robust library of first-party tools and libraries, such as Forge, Vapor, Nova, and Envoyer help you take your projects to the next level. Pair them with powerful open source libraries like Cashier, Dusk, Echo, Horizon, Sanctum, Telescope, and more.

Magento Installation

Step1 - First update your system repo

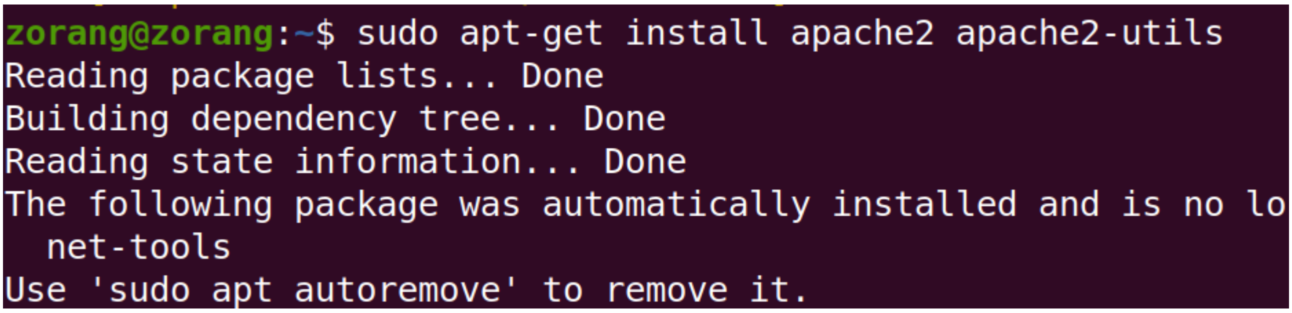
- Sudo apt-get update



```
zorang@zorang:~$ sudo apt-get update
[sudo] password for zorang:
Hit:1 http://in.archive.ubuntu.com/ubuntu hirs
Get:2 http://in.archive.ubuntu.com/ubuntu hirs
Get:3 http://in.archive.ubuntu.com/ubuntu hirs
Get:4 http://in.archive.ubuntu.com/ubuntu hirs
```

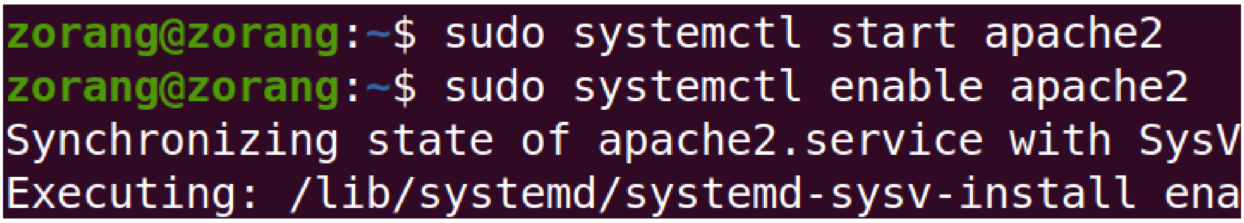
Step2 - Install apache2 and start and enable apache service

- `sudo apt-get install apache2 apache2-utils`



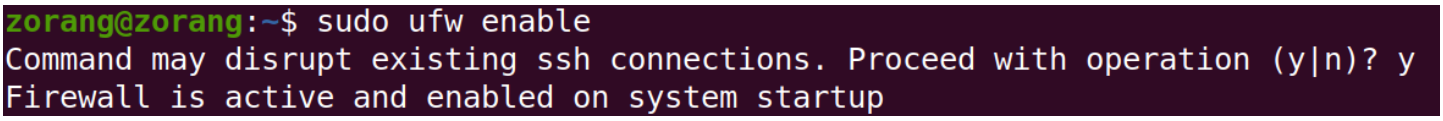
```
zorang@zorang:~$ sudo apt-get install apache2 apache2-utils
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no lo
  net-tools
Use 'sudo apt autoremove' to remove it.
```

- `sudo systemctl start apache2`
- `sudo systemctl enable apache2`



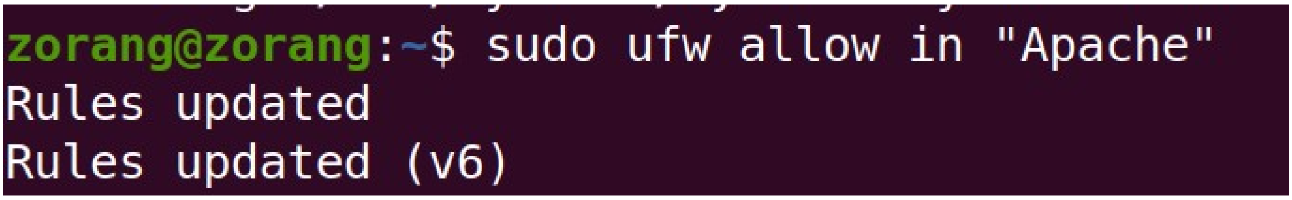
```
zorang@zorang:~$ sudo systemctl start apache2
zorang@zorang:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV
Executing: /lib/systemd/systemd-sysv-install ena
```

Step3 - Once you’ve started Apache, you then need to allow HTTP traffic on your UFW firewall



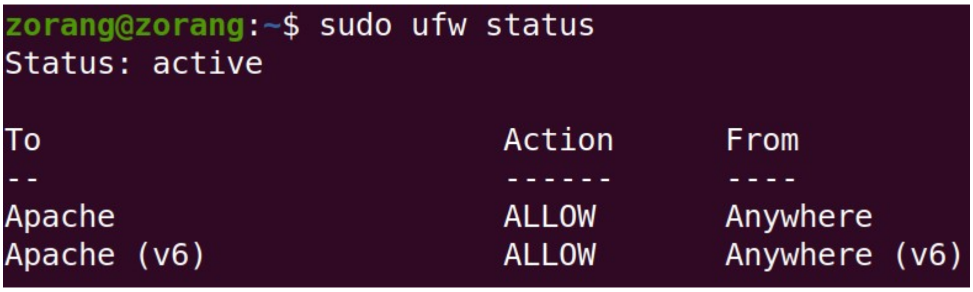
```
zorang@zorang:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
```

- `sudo ufw allow in "Apache"`



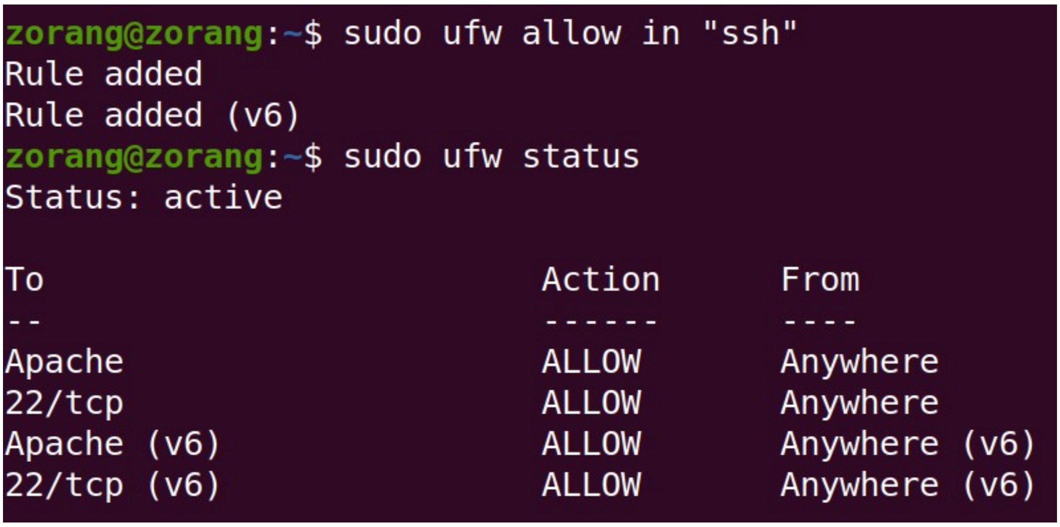
```
zorang@zorang:~$ sudo ufw allow in "Apache"
Rules updated
Rules updated (v6)
```

- `sudo ufw status`



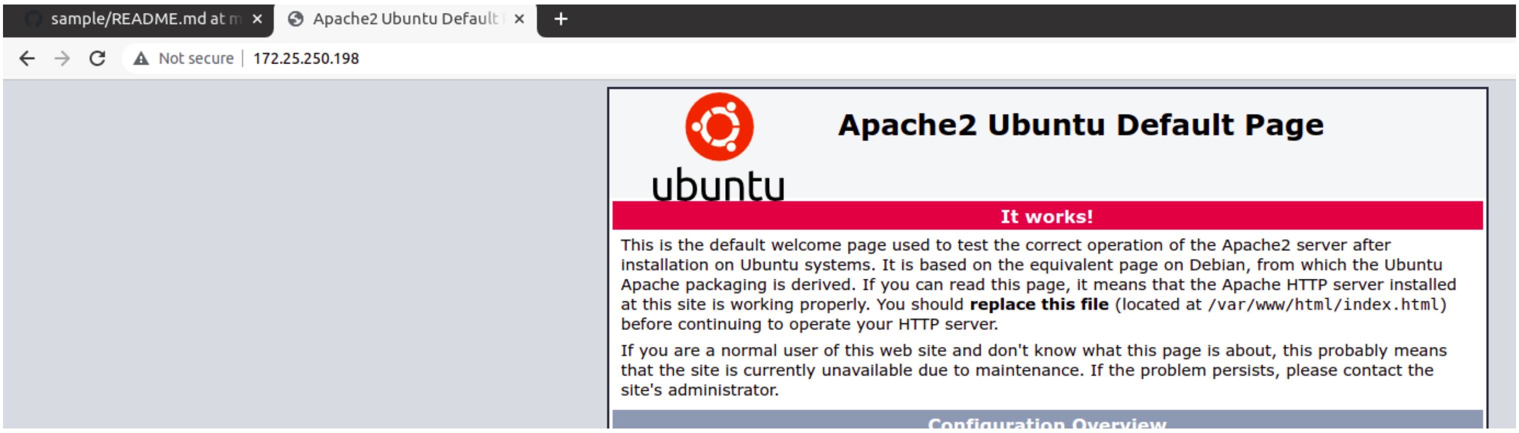
```
zorang@zorang:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
Apache                     ALLOW       Anywhere
Apache (v6)                ALLOW       Anywhere (v6)
```

Important - please make sure you allow ssh if your server is on virtual machine



```
zorang@zorang:~$ sudo ufw allow in "ssh"
Rule added
Rule added (v6)
zorang@zorang:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
Apache                     ALLOW       Anywhere
22/tcp                     ALLOW       Anywhere
Apache (v6)                ALLOW       Anywhere (v6)
22/tcp (v6)                ALLOW       Anywhere (v6)
```

To test whether the Apache server is running, open your web browser and enter the following URL in the address bar



Apache2 Ubuntu Default Page

It works!

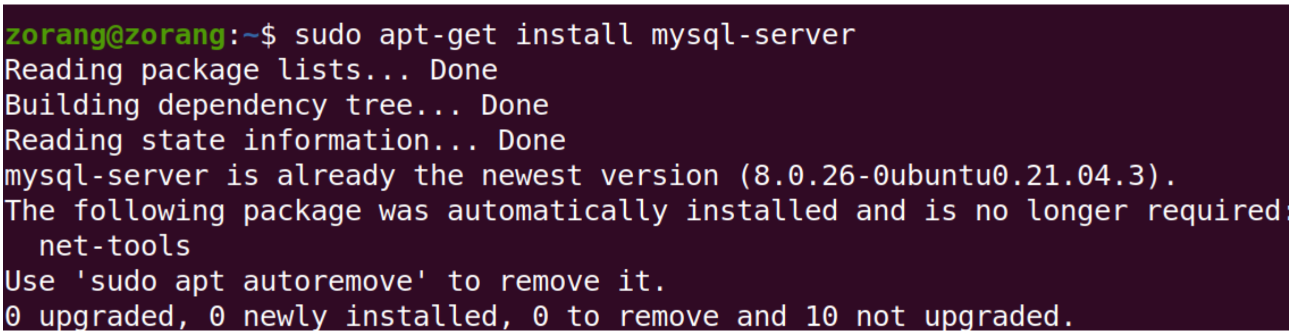
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Step4 - Now, Install mysql-server and set password for root

- `sudo apt-get install mysql-server`

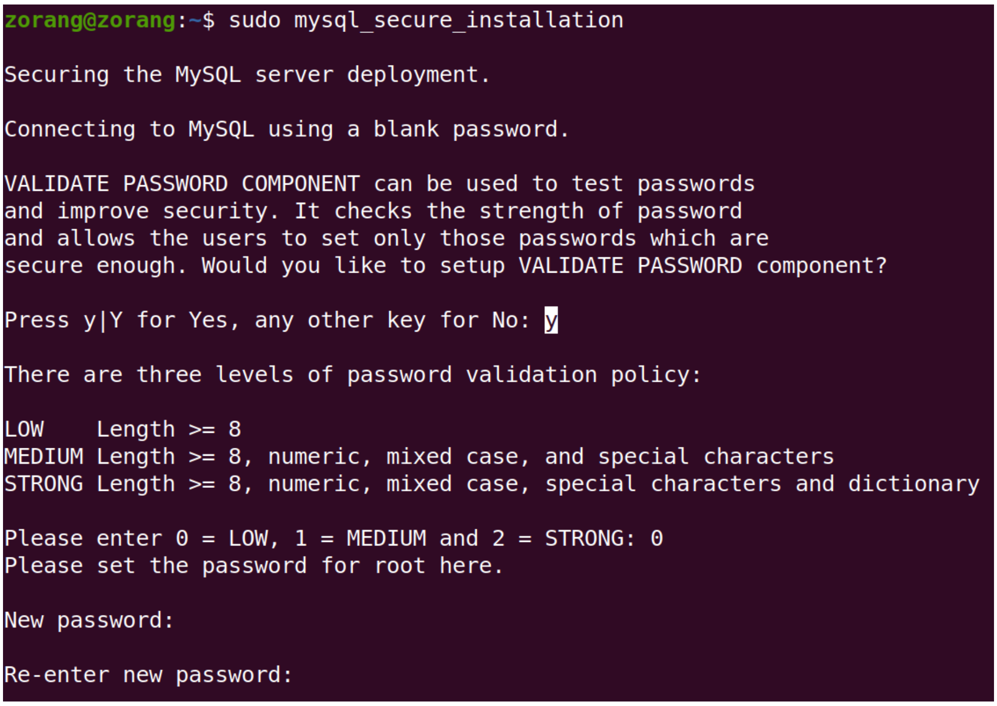


```
zorang@zorang:~$ sudo apt-get install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mysql-server is already the newest version (8.0.26-0ubuntu0.21.04.3).
The following package was automatically installed and is no longer required
  net-tools
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 10 not upgraded.
```

- `sudo mysql_secure_installation`

As shown in below image press Y to set root passsword

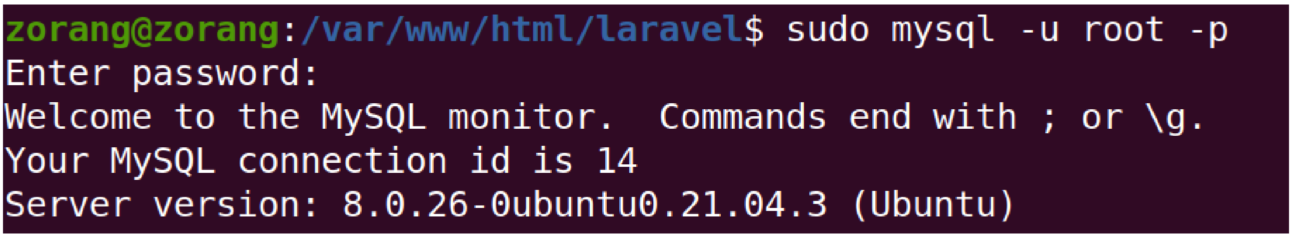
Choose password policy, and set root password like (Ritik@123456) and then press y for all.



```
zorang@zorang:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y

There are three levels of password validation policy:

LOW    Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0
Please set the password for root here.

New password:

Re-enter new password:
```

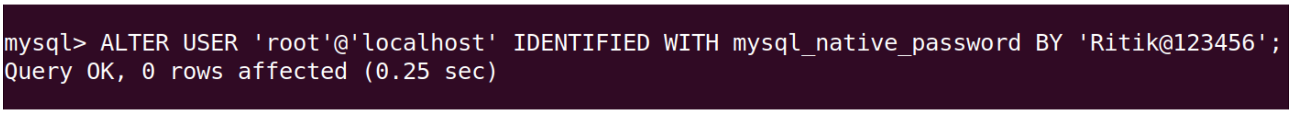
Step5 - Create user for magento and remember its password and give it permission to create database

- `sudo mysql -u root -p`



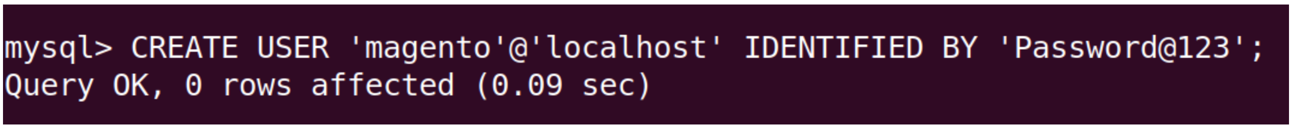
```
zorang@zorang:/var/www/html/laravel$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.26-0ubuntu0.21.04.3 (Ubuntu)
```

- `ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'Ritik@123456';`



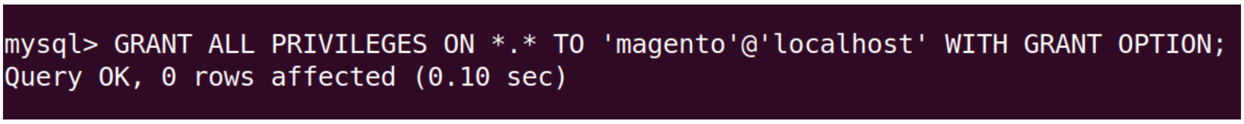
```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'Ritik@123456';
Query OK, 0 rows affected (0.25 sec)
```

- CREATE USER 'magento'@'localhost' IDENTIFIED BY 'Password@123';



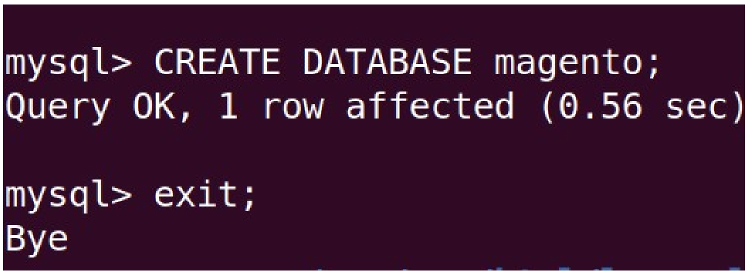
```
mysql> CREATE USER 'magento'@'localhost' IDENTIFIED BY 'Password@123';
Query OK, 0 rows affected (0.09 sec)
```

- GRANT ALL PRIVILEGES ON *.* TO 'magento'@'localhost' WITH GRANT OPTION;



```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'magento'@'localhost' WITH GRANT OPTION;
Query OK, 0 rows affected (0.10 sec)
```

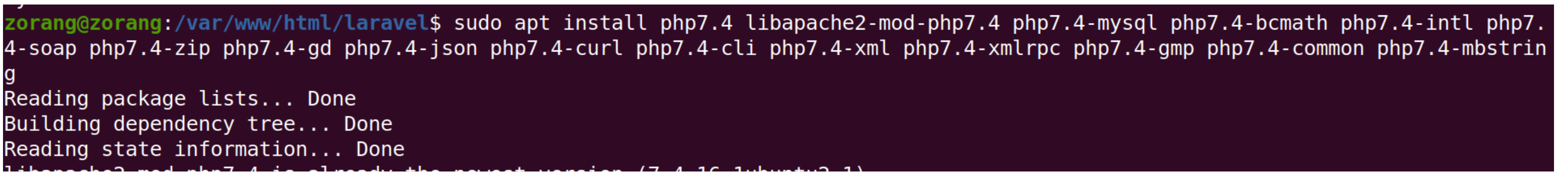
- CREATE DATABASE magento;



```
mysql> CREATE DATABASE magento;
Query OK, 1 row affected (0.56 sec)

mysql> exit;
Bye
```

Step6 - Install required php7.4 packages

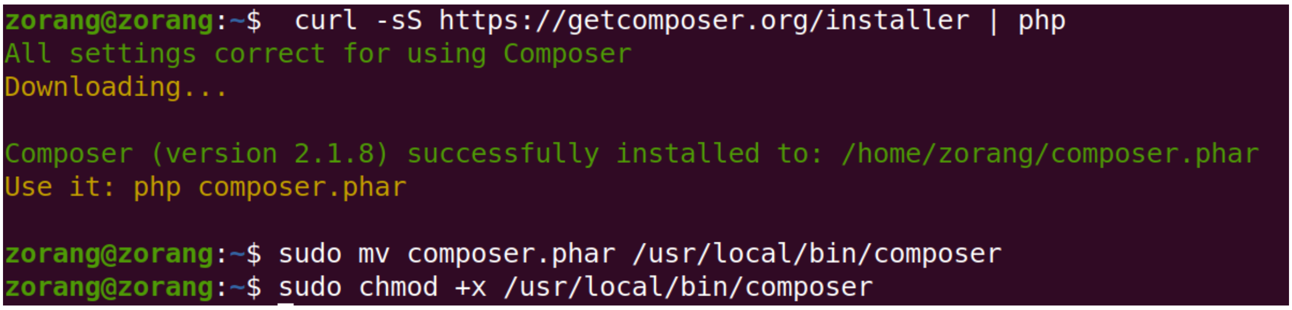
- sudo apt install php7.4 libapache2-mod-php7.4 php7.4-mysql php7.4-bcmath php7.4-intl php7.4-soap php7.4-zip php7.4-gd php7.4-json php7.4-curl php7.4-cli php7.4-xml php7.4-xmlrpc php7.4-gmp php7.4-common php7.4-mbstring



```
zorang@zorang:/var/www/html/laravel$ sudo apt install php7.4 libapache2-mod-php7.4 php7.4-mysql php7.4-bcmath php7.4-intl php7.
4-soap php7.4-zip php7.4-gd php7.4-json php7.4-curl php7.4-cli php7.4-xml php7.4-xmlrpc php7.4-gmp php7.4-common php7.4-mbstrin
g
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libapache2-mod-php7.4 is already the newest version (7.4.16-1ubuntu2.1).
```

Step7 - Install composer (required for managing dependencies, libraries for Magento)

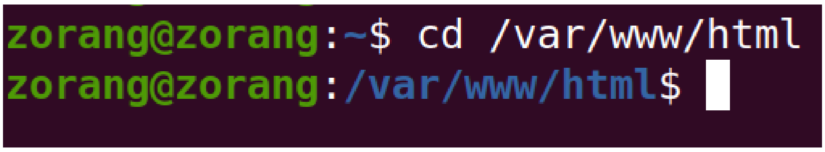
- `curl -sS https://getcomposer.org/installer | php`
- `curl -sS https://getcomposer.org/installer | php`
- `sudo chmod +x /usr/local/bin/composer`



```
zorang@zorang:~$  curl -sS https://getcomposer.org/installer | php
All settings correct for using Composer
Downloading...

Composer (version 2.1.8) successfully installed to: /home/zorang/composer.phar
Use it: php composer.phar

zorang@zorang:~$ sudo mv composer.phar /usr/local/bin/composer
zorang@zorang:~$ sudo chmod +x /usr/local/bin/composer
```

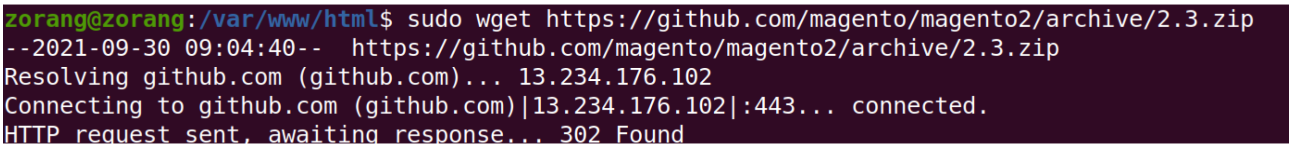
Step8 - Download the latest version of the Magento from GitHub (https://github.com/magento/magento2) or from official website(https://www.mageplaza.com/download-magento/).

- `cd /var/www/html`



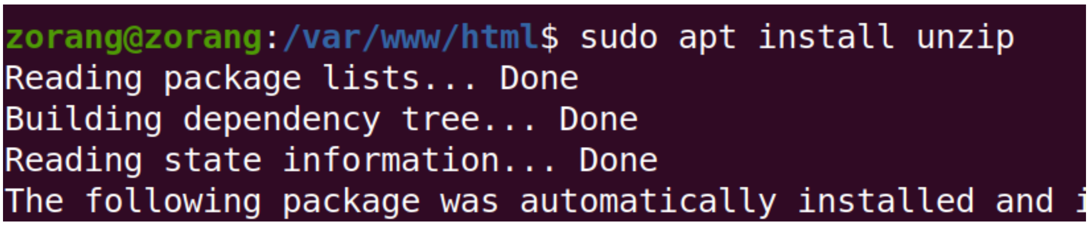
```
zorang@zorang:~$ cd /var/www/html
zorang@zorang:/var/www/html$ 
```

- sudo wget https://github.com/magento/magento2/archive/2.3.zip



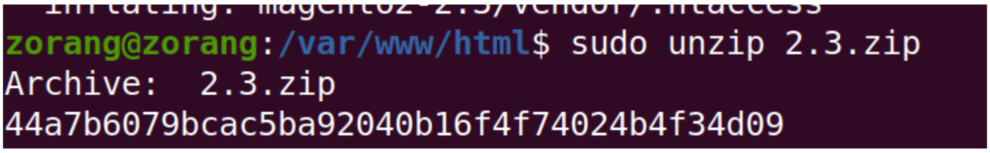
```
zorang@zorang:/var/www/html$ sudo wget https://github.com/magento/magento2/archive/2.3.zip
--2021-09-30 09:04:40--  https://github.com/magento/magento2/archive/2.3.zip
Resolving github.com (github.com)... 13.234.176.102
Connecting to github.com (github.com)|13.234.176.102|:443... connected.
HTTP request sent, awaiting response... 302 Found
```

- sudo apt install unzip



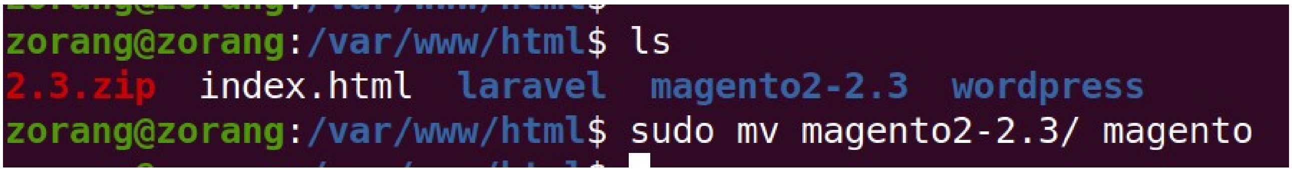
```
zorang@zorang:/var/www/html$ sudo apt install unzip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and i
```

- sudo unzip 2.3.zip



```
  inflating: magento2-2.3/vendor/.htaccess
zorang@zorang:/var/www/html$ sudo unzip 2.3.zip
Archive:  2.3.zip
44a7b6079bcac5ba92040b16f4f74024b4f34d09
```

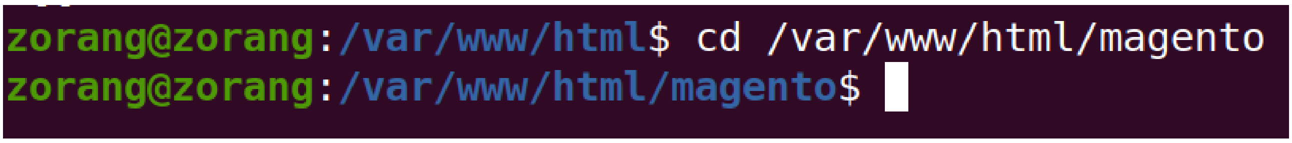
- sudo mv magento2-2.3/ magento



```
zorang@zorang:/var/www/html$
zorang@zorang:/var/www/html$ ls
2.3.zip   index.html   laravel   magento2-2.3   wordpress
zorang@zorang:/var/www/html$ sudo mv magento2-2.3/ magento
zorang@zorang:/var/www/html$
```

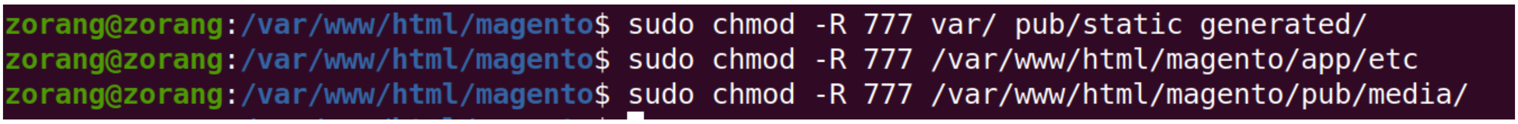
Step9 - Give proper permission to magento files so that magento application can run

- cd /var/www/html/magento



```
zorang@zorang:/var/www/html$ cd /var/www/html/magento
zorang@zorang:/var/www/html/magento$ 
```

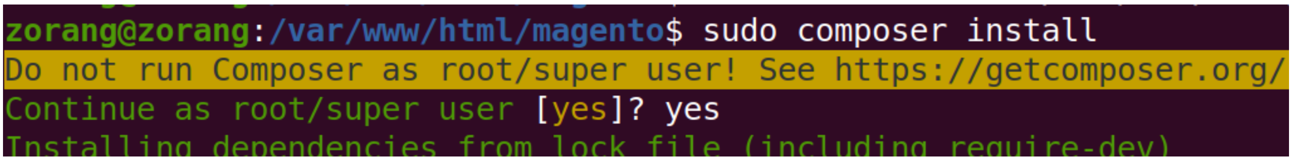
- `sudo chmod -R 777 var/ pub/static generated/`
- `sudo chmod -R 777 /var/www/html/magento/app/etc`
- `sudo chmod -R 777 /var/www/html/magento/pub/media/`



```
zorang@zorang:/var/www/html/magento$ sudo chmod -R 777 var/ pub/static generated/
zorang@zorang:/var/www/html/magento$ sudo chmod -R 777 /var/www/html/magento/app/etc
zorang@zorang:/var/www/html/magento$ sudo chmod -R 777 /var/www/html/magento/pub/media/
```

Step10 - Run composer install to downloads and installs all the libraries and dependencies outlined in Magento dir

- `sudo composer install`



```
zorang@zorang:/var/www/html/magento$ sudo composer install
Do not run Composer as root/super user! See https://getcomposer.org/
Continue as root/super user [yes]? yes
Installing dependencies from lock file (including require-dev)
```

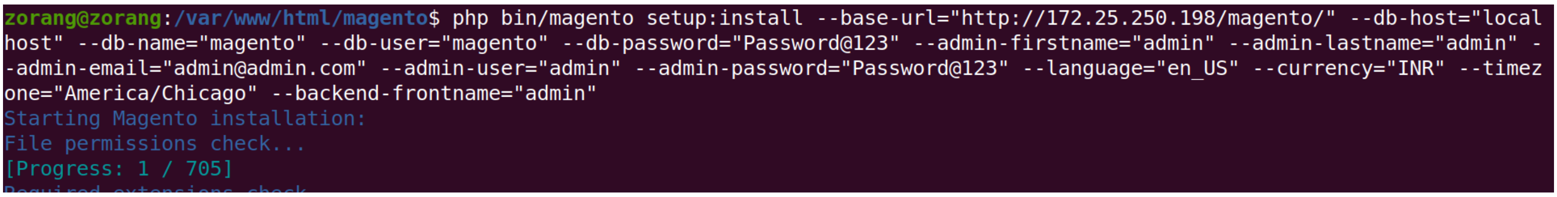
Step11 - Now run magento application

ℹ Edit the information to match your requirements and the configuration of your system:

- **base-url -** The location (URL) of your store. In this example, the store is installed on the localhost in the magento2.4 sub-directory.
- **db-host -** If Magento is on the same server as your database, use localhost. If you are using a separate database server, enter the hostname of that server.
- **db-name -** The name of the MySQL database created earlier.

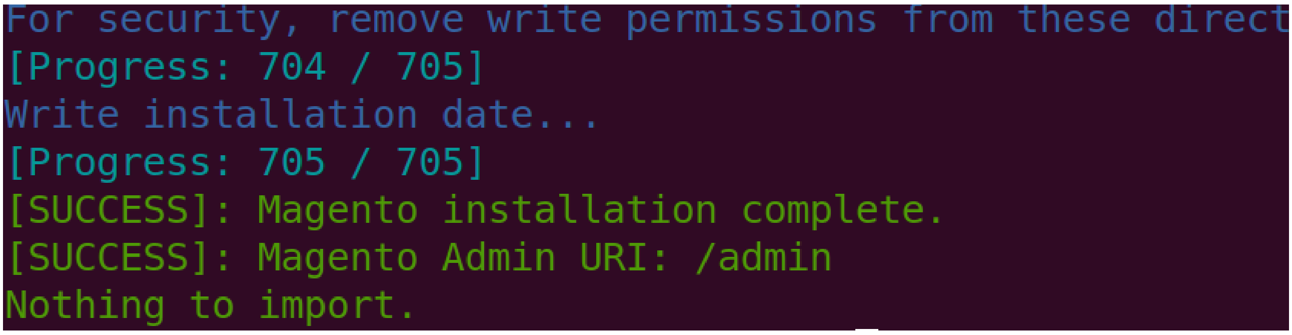
- **db-user -** Enter the username of your MySQL user.
- **db-password -** The password for your MySQL user.
- **admin-firstname and admin-lastname -** Set the full name for your Magento admin user.
- **admin-email -** Define a contact email for system notifications and password resets.
- **admin-user / admin-password -** Create the login credentials for the Magento Admin control panel.
- **language -** Defines the default language for your store.
- **currency -** Sets the base currency for your store.
- **timezone -** Regulates the default time zone for Magento.

-
  ```
  php bin/magento setup:install
  --base-url="http://172.25.250.198/magento/" --db-
  host="localhost" --db-name="magento" --db-user="magento" --db-
  password="Password@123" --admin-firstname="admin" --admin-
  lastname="admin" --admin-email="admin@admin.com" --admin-
  user="admin" --admin-password="Password@123" --
  language="en_US" --currency="INR" --timezone="America/Chicago"
  --backend-frontname="admin"
  ```



```
zorang@zorang:/var/www/html/magento$ php bin/magento setup:install --base-url="http://172.25.250.198/magento/" --db-host="local
host" --db-name="magento" --db-user="magento" --db-password="Password@123" --admin-firstname="admin" --admin-lastname="admin" -
-admin-email="admin@admin.com" --admin-user="admin" --admin-password="Password@123" --language="en_US" --currency="INR" --timez
one="America/Chicago" --backend-frontname="admin"
Starting Magento installation:
File permissions check...
[Progress: 1 / 705]
Required extensions check
```

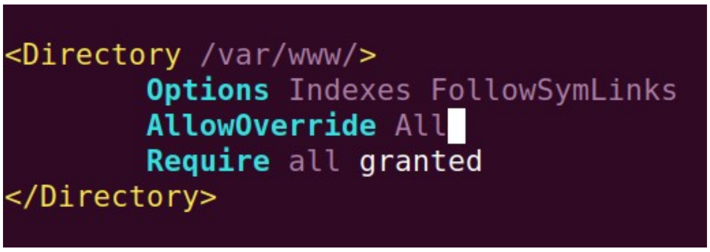
After some minutes magento installation is completed -



```
For security, remove write permissions from these direct
[Progress: 704 / 705]
Write installation date...
[Progress: 705 / 705]
[SUCCESS]: Magento installation complete.
[SUCCESS]: Magento Admin URI: /admin
Nothing to import.
```

Step12 - To run magento properly allow dir

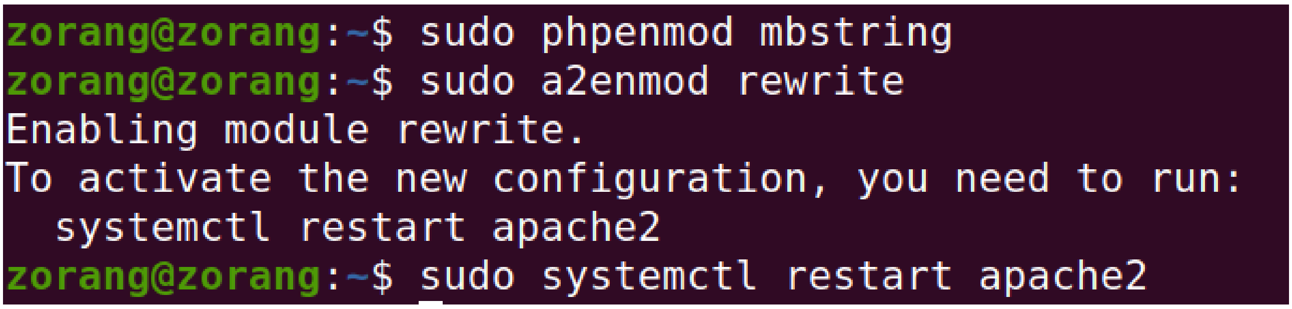
```
sudo vim /etc/apache2/apache2.conf
```



```
<Directory /var/www/>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
</Directory>
```

Step13 - Enable mbstring module and restart apache service

- `sudo phpenmod mbstring`
- `sudo a2enmod rewrite`
- `sudo systemctl restart apache2`



```
zorang@zorang:~$ sudo phpenmod mbstring
zorang@zorang:~$ sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  systemctl restart apache2
zorang@zorang:~$ sudo systemctl restart apache2
```

Step14 - Go to browser and type the url

(http://ip/magento/admin) or

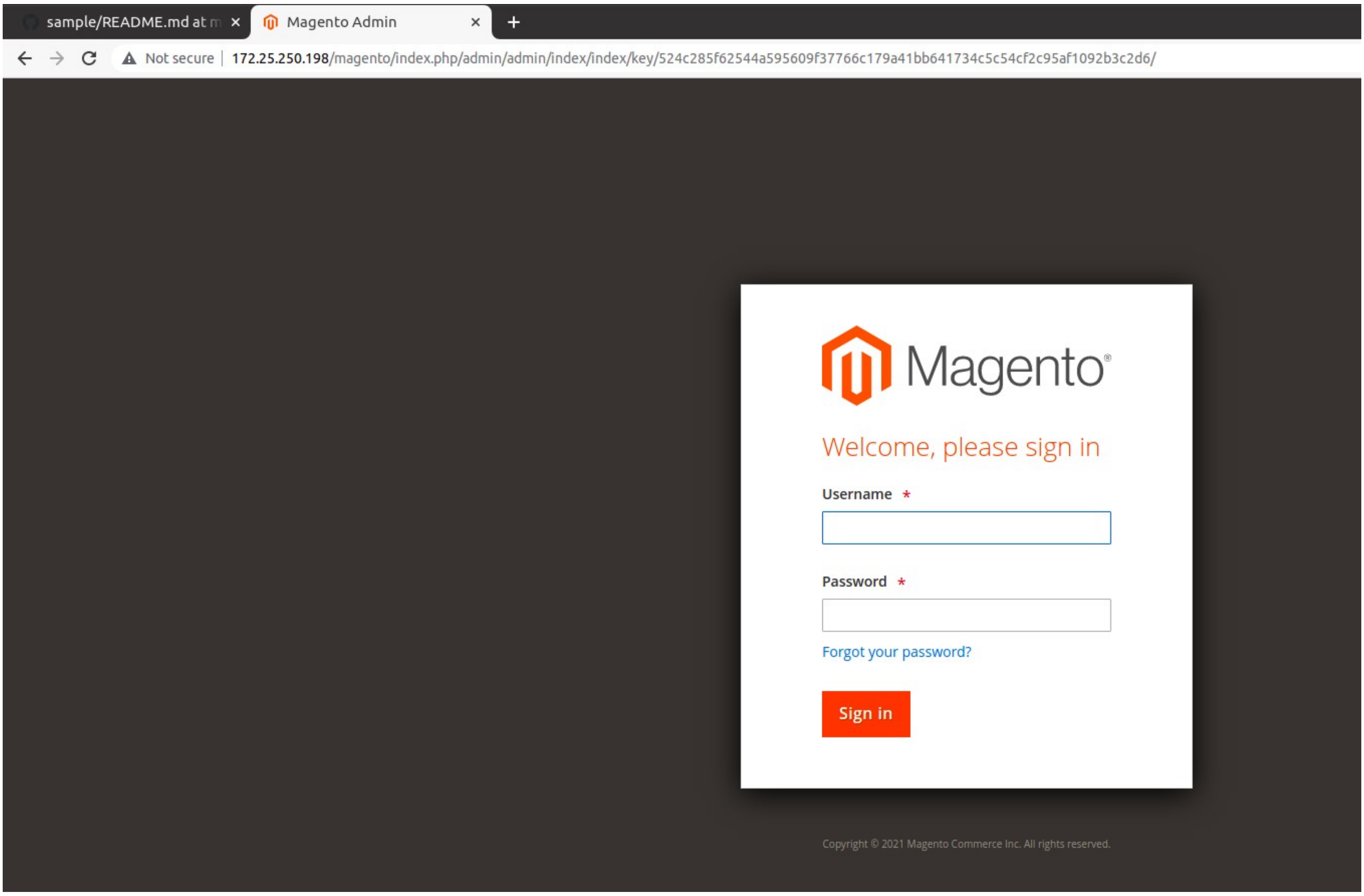
(http://domainname/magento/admin) as show below and username and password is

Username = admin

Password = Password@123

when you set while running magento as shown in below

-
  ```
  php bin/magento setup:install --base-url="http://172.25.250.198/magento/" --db-host="localhost" --db-name="magento" --db-user="magento" --db-password="Password@123" --admin-firstname="admin" --admin-lastname="admin" --admin-email="admin@admin.com" --admin-user="admin" --admin-password="Password@123" --language="en_US" --currency="INR" --timezone="America/Chicago" --backend-frontname="admin"
  ```



172.25.250.198/magento/index.php/admin/admin/index/index/key/524c285f62544a595609f37766c179a41bb641734c5c54cf2c95af1092b3c2d6/

Welcome, please sign in

Copyright © 2021 Magento Commerce Inc. All rights reserved.

Security Tips

Please visit below url to secure your web server

https://github.com/RG-linux-lover/WordPress-Laravel-Magento-Installation/tree/master/Security%20tips