Installation of LAMP stack on Amazon Linux OS on AWS Ec2 Instance

LAMP is a software stack that is widely used for building and hosting dynamic web applications. It offers a low cost of ownership, easy customization, and a large user community. The components of the LAMP stack are highly compatible and can be easily integrated, making it a popular choice for web developers.

The acronym stands for Linux, Apache, MySQL, and PHP. Let's look at each component in detail:

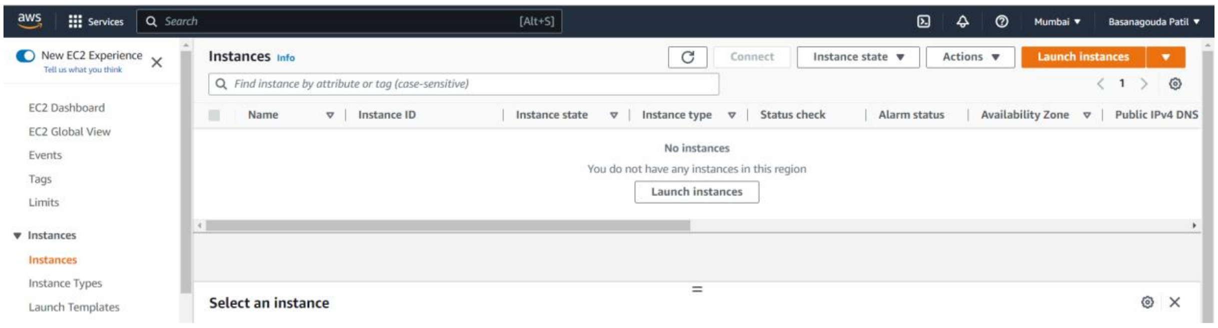
Linux: Linux is an open-source operating system that is widely used for web hosting purposes. It provides a stable, secure, and scalable environment for web applications to run. Linux is flexible and can be easily configured to meet the requirements of different web applications.

Apache: Apache is a widely used open-source web server software that serves HTTP requests and manages the flow of data between the client and server. It is highly configurable, with a large number of modules that can be added to extend its functionality. Apache is also highly scalable, and can handle a large number of concurrent connections, making it suitable for high-traffic web applications.

MySQL: MySQL is an open-source relational database management system (RDBMS) that is used to store and retrieve data in a structured manner. It is widely used in web applications due to its reliability, scalability, and ease of use. MySQL provides a high level of security, with built-in encryption and access control mechanisms to ensure the protection of sensitive data.

PHP: PHP is a server-side scripting language that is used to write dynamic web pages and applications. PHP code is executed on the server, and the results are sent back to the client in the form of HTML or other formats. PHP provides a large number of functions and libraries that make it easy to perform tasks such as accessing databases, generating dynamic content, and handling user input.

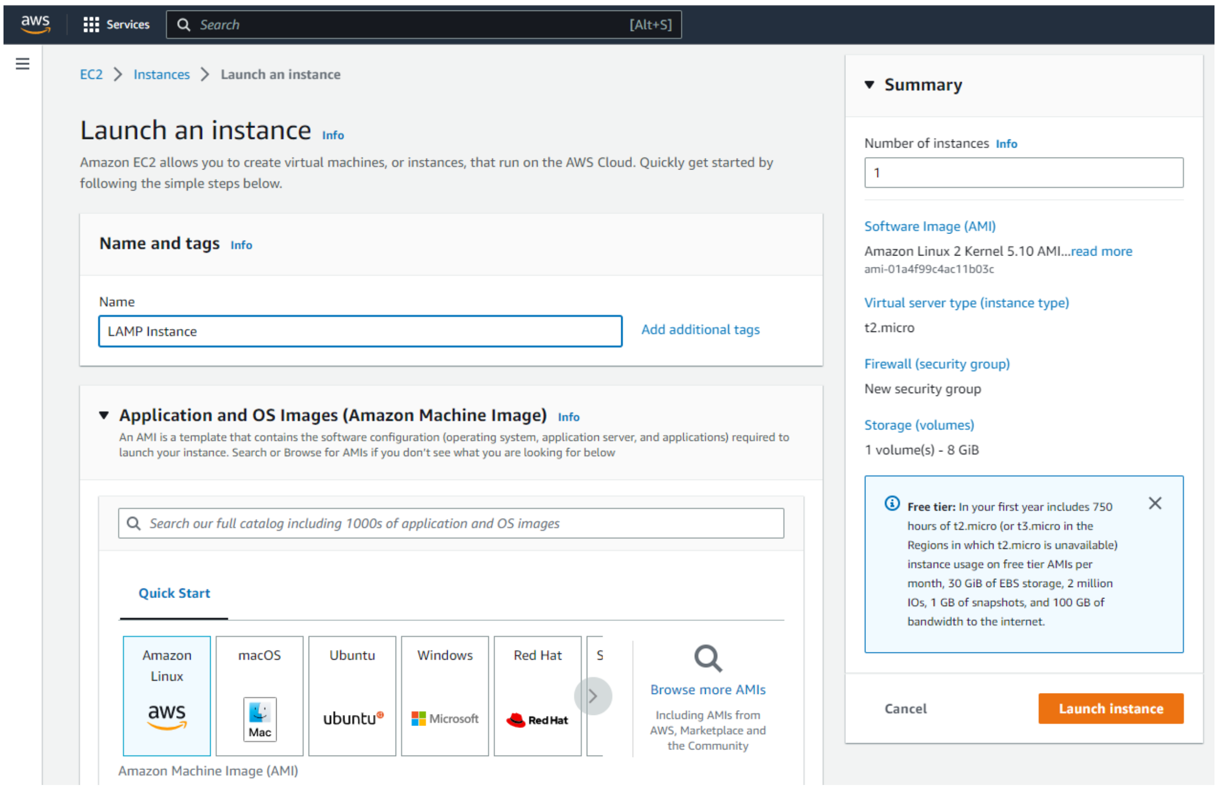
Step 1: Creating an Amazon Linux OS in AWS Ec2 Instance



No instances

You do not have any instances in this region

Create AWS account and search Ec2 in search bar and click on Launch Instance.



EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: LAMP Instance

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Quick Start: Amazon Linux, macOS, Ubuntu, Windows, Red Hat

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...read more ami-01a4f99c4ac11b03c

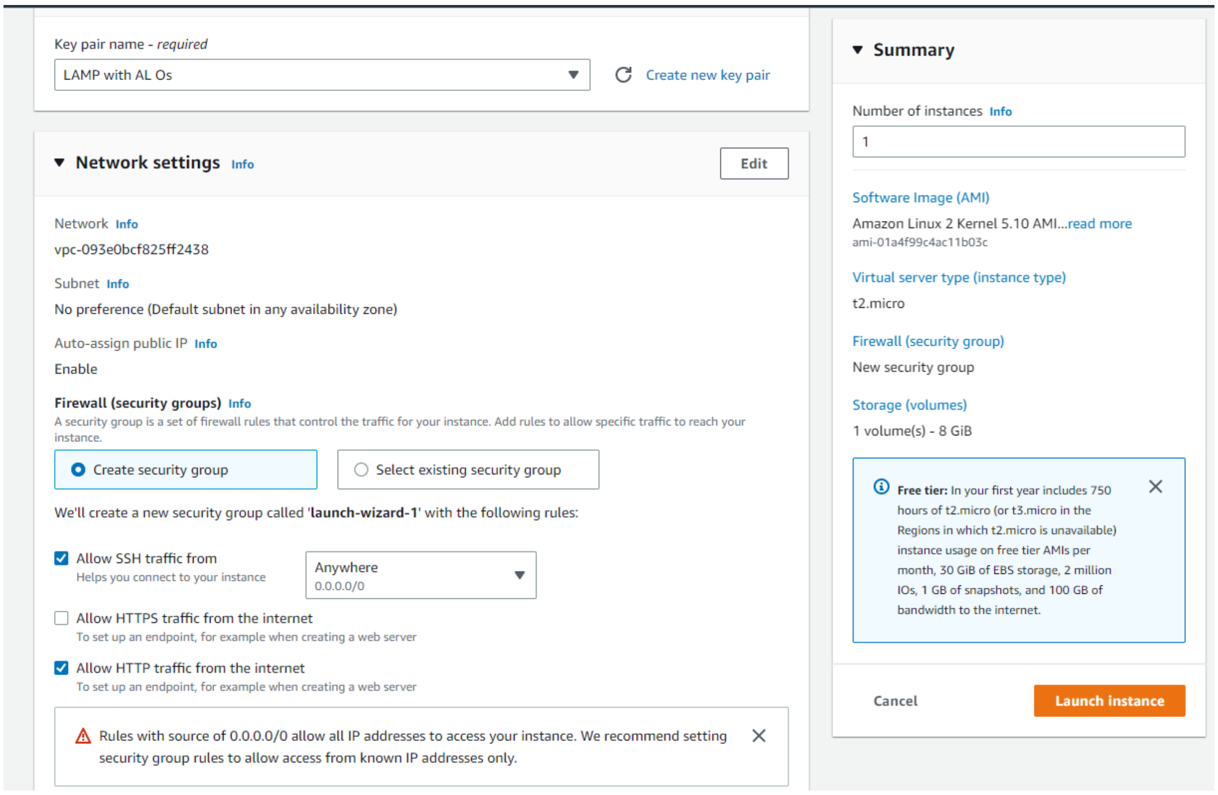
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Creating a Ec2 instance with Amazon Linux OS , naming it LAMP instance



Key pair name - *required*: LAMP with AL Os

Network settings Info

Network: vpc-093e0bcf825ff2438

Subnet: No preference (Default subnet in any availability zone)

Auto-assign public IP: Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

[✓ Create security group] Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

✓ Allow SSH traffic from — Anywhere 0.0.0.0/0
Helps you connect to your instance

☐ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

✓ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...read more ami-01a4f99c4ac11b03c

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Create a key pair and in network settings allow HTTP traffic then click on launch instance.

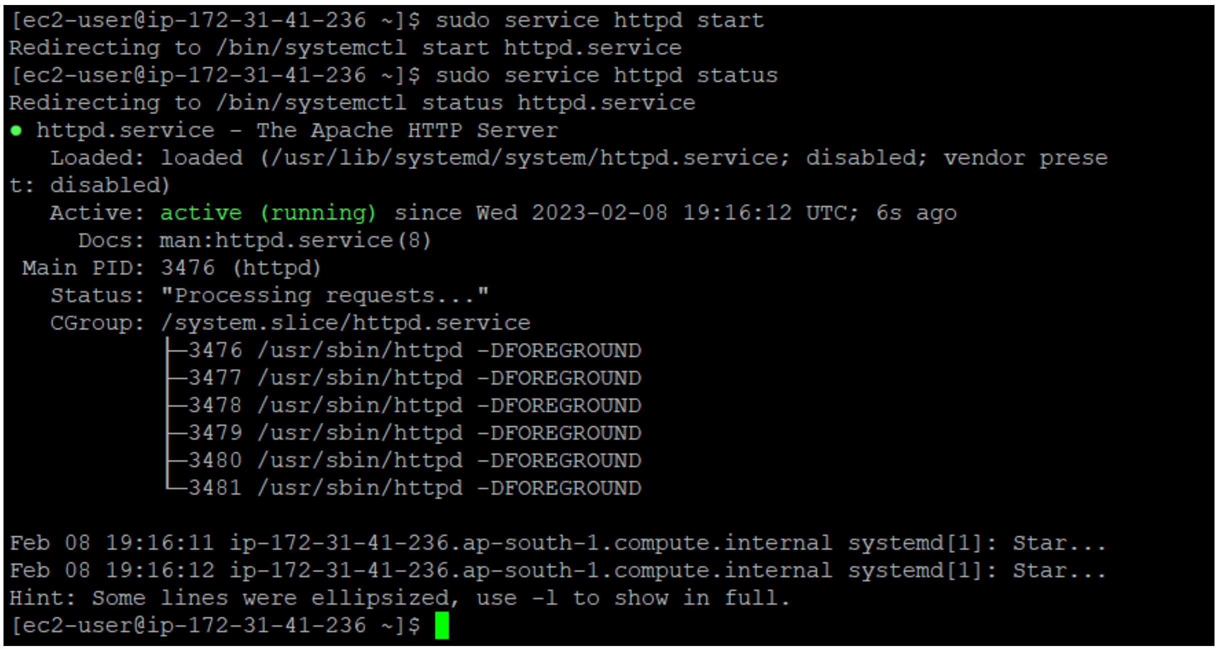
Step 2 : Install and Run Apache (httpd) in Amazon Linux OS

“httpd” is Apache package name in Amazon linux, whereas in ubuntu OS it is called as “apache2”

Commands need to be run.

sudo yum install httpd -y

sudo service httpd start
sudo service httpd status



```
[ec2-user@ip-172-31-41-236 ~]$ sudo service httpd start
Redirecting to /bin/systemctl start httpd.service
[ec2-user@ip-172-31-41-236 ~]$ sudo service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor prese
t: disabled)
   Active: active (running) since Wed 2023-02-08 19:16:12 UTC; 6s ago
     Docs: man:httpd.service(8)
 Main PID: 3476 (httpd)
   Status: "Processing requests..."
   CGroup: /system.slice/httpd.service
           ├─3476 /usr/sbin/httpd -DFOREGROUND
           ├─3477 /usr/sbin/httpd -DFOREGROUND
           ├─3478 /usr/sbin/httpd -DFOREGROUND
           ├─3479 /usr/sbin/httpd -DFOREGROUND
           ├─3480 /usr/sbin/httpd -DFOREGROUND
           └─3481 /usr/sbin/httpd -DFOREGROUND

Feb 08 19:16:11 ip-172-31-41-236.ap-south-1.compute.internal systemd[1]: Star...
Feb 08 19:16:12 ip-172-31-41-236.ap-south-1.compute.internal systemd[1]: Star...
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-41-236 ~]$ 
```

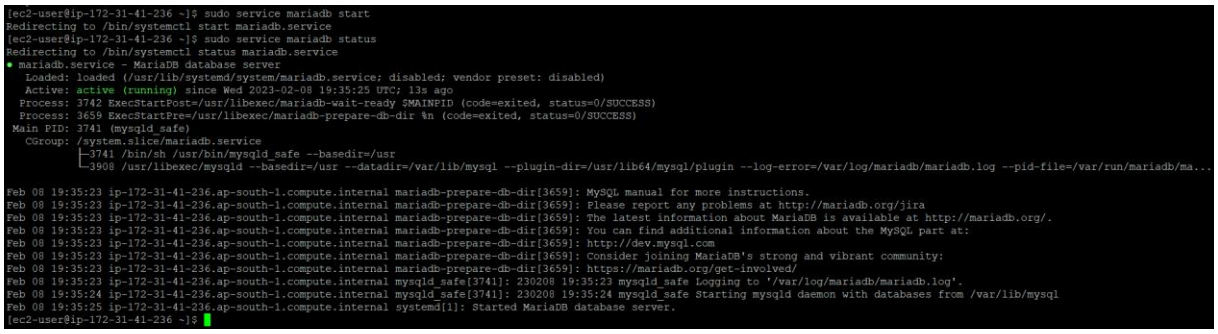
Apache install and started successfully. A part is completed for LAMP stack.

Step 3 : Install and Run mysql in Amazon Linux OS

Latest Upgraded version of Mysql is called Mariadb. we can say that mysql and mariadb both are same above mysql version 5.8 and in Amazon Linux 2 mysql is available as “mariadb-server”.

Commands need to be run.

sudo yum install mariadb-server -y
sudo service mariadb start
sudo service mariadb status

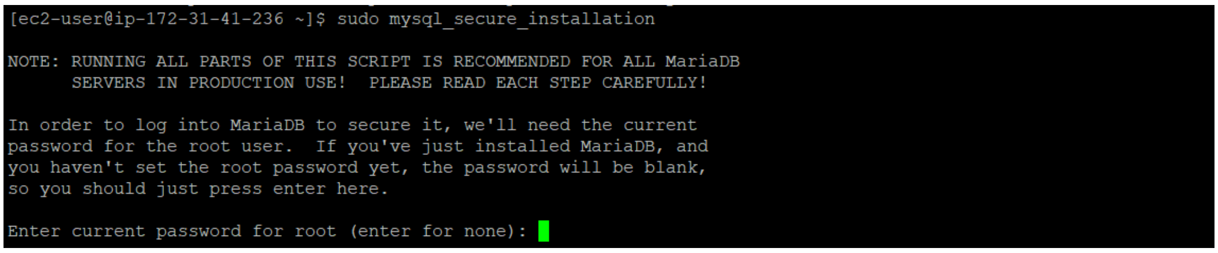


```
[ec2-user@ip-172-31-41-236 ~]$ sudo service mariadb start
Redirecting to /bin/systemctl start mariadb.service
[ec2-user@ip-172-31-41-236 ~]$ sudo service mariadb status
Redirecting to /bin/systemctl status mariadb.service
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; vendor preset: disabled)
   Active: active (running) since Wed 2023-02-08 19:35:25 UTC; 13s ago
  Process: 3742 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited, status=0/SUCCESS)
  Process: 3659 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, status=0/SUCCESS)
 Main PID: 3741 (mysqld_safe)
   CGroup: /system.slice/mariadb.service
           ├─3741 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
           └─3908 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-dir=/usr/lib64/mysql/plugin --log-error=/var/log/mariadb/mariadb.log --pid-file=/var/run/mariadb/ma...

Feb 08 19:35:23 ip-172-31-41-236.ap-south-1.compute.internal mariadb-prepare-db-dir[3659]: MySQL manual for more instructions.
Feb 08 19:35:23 ip-172-31-41-236.ap-south-1.compute.internal mariadb-prepare-db-dir[3659]: Please report any problems at http://mariadb.org/jira
Feb 08 19:35:23 ip-172-31-41-236.ap-south-1.compute.internal mariadb-prepare-db-dir[3659]: The latest information about MariaDB is available at http://mariadb.org/.
Feb 08 19:35:23 ip-172-31-41-236.ap-south-1.compute.internal mariadb-prepare-db-dir[3659]: You can find additional information about the MySQL part at:
Feb 08 19:35:23 ip-172-31-41-236.ap-south-1.compute.internal mariadb-prepare-db-dir[3659]: http://dev.mysql.com
Feb 08 19:35:23 ip-172-31-41-236.ap-south-1.compute.internal mariadb-prepare-db-dir[3659]: Consider joining MariaDB's strong and vibrant community:
Feb 08 19:35:23 ip-172-31-41-236.ap-south-1.compute.internal mariadb-prepare-db-dir[3659]: https://mariadb.org/get-involved/
Feb 08 19:35:23 ip-172-31-41-236.ap-south-1.compute.internal mysqld_safe[3741]: 230208 19:35:23 mysqld_safe Logging to '/var/log/mariadb/mariadb.log'.
Feb 08 19:35:24 ip-172-31-41-236.ap-south-1.compute.internal mysqld_safe[3741]: 230208 19:35:24 mysqld_safe Starting mysqld daemon with databases from /var/lib/mysql
Feb 08 19:35:25 ip-172-31-41-236.ap-south-1.compute.internal systemd[1]: Started MariaDB database server.
[ec2-user@ip-172-31-41-236 ~]$ 
```

Mariadb install and running successfully. we also need to set password for it.

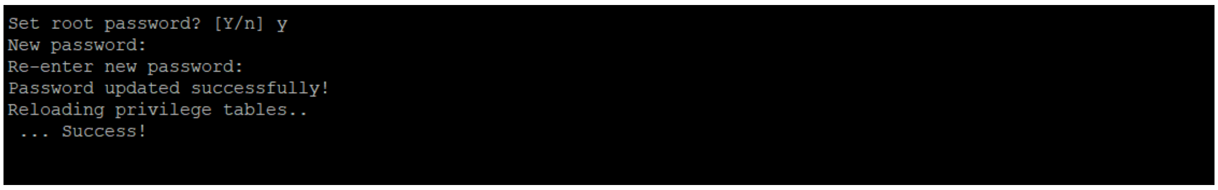
Command to set a password for it to make it more secure

sudo mysql_secure_installation



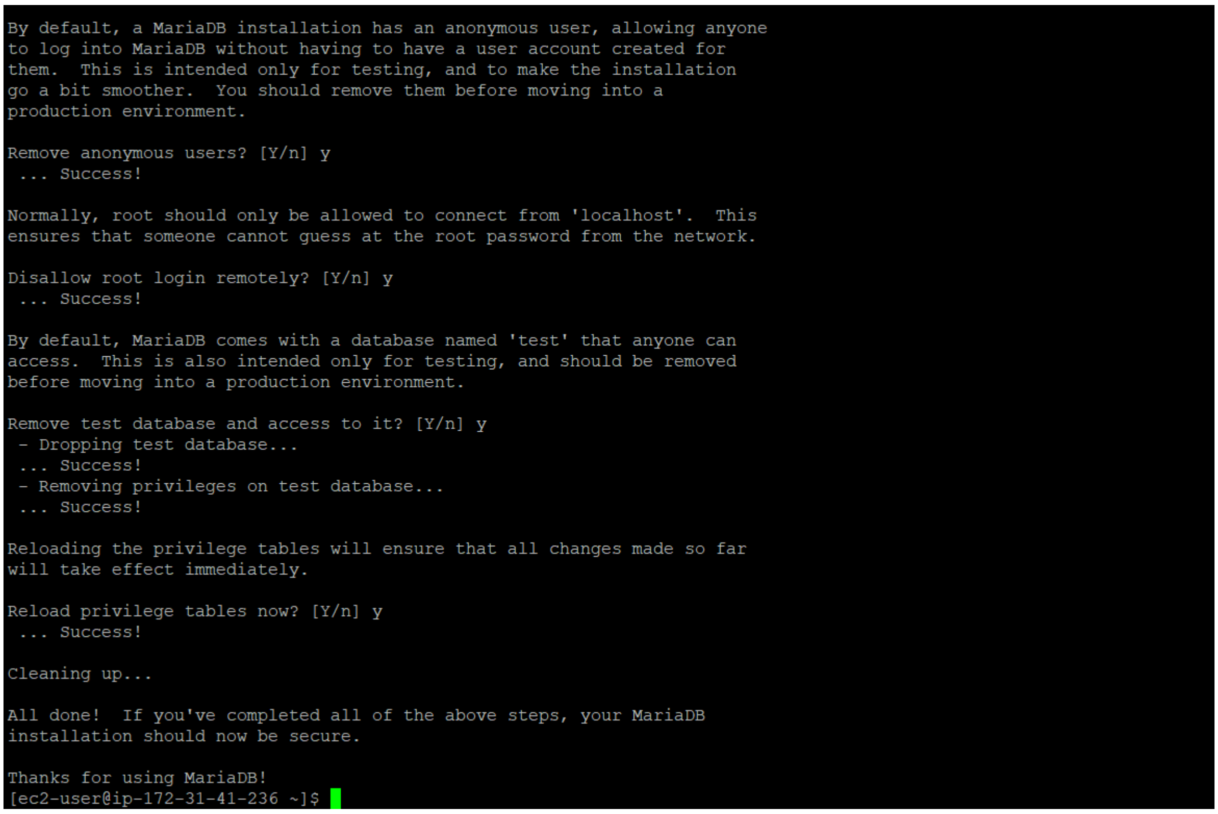
```
[ec2-user@ip-172-31-41-236 ~]$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none): 
```

it asks for current password , just press ENTER on keyboard as its not having any password



```
Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
 ... Success!
```

Now it asks for setting root password , say yes (press y) and enter new password , root is default usern name in mysql/mariadb while entering password cursor does not moves , so still enter password.

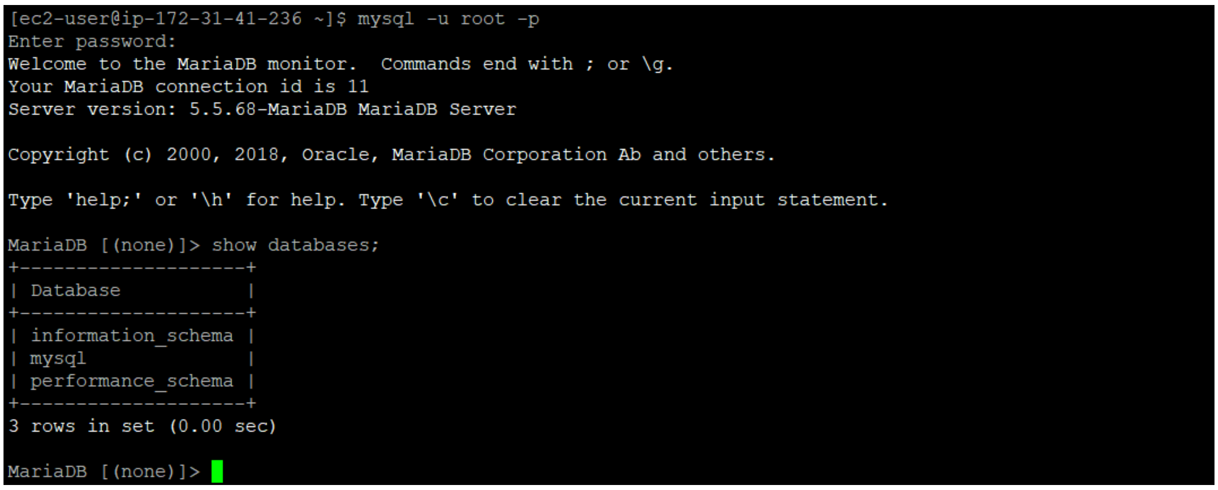


```
By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
 ... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
 ... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
[ec2-user@ip-172-31-41-236 ~]$
```

Now, it will ask some questions like do you want to remove anonymous users? , Disallow root login remotely? etc.. press y until it finishes and tells Thanks for using mariaDB.

Now you can login to mysql/mariadb by running below command

mysql -u root -p



```
[ec2-user@ip-172-31-41-236 ~]$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 11
Server version: 5.5.68-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
+--------------------+
3 rows in set (0.00 sec)

MariaDB [(none)]>
```

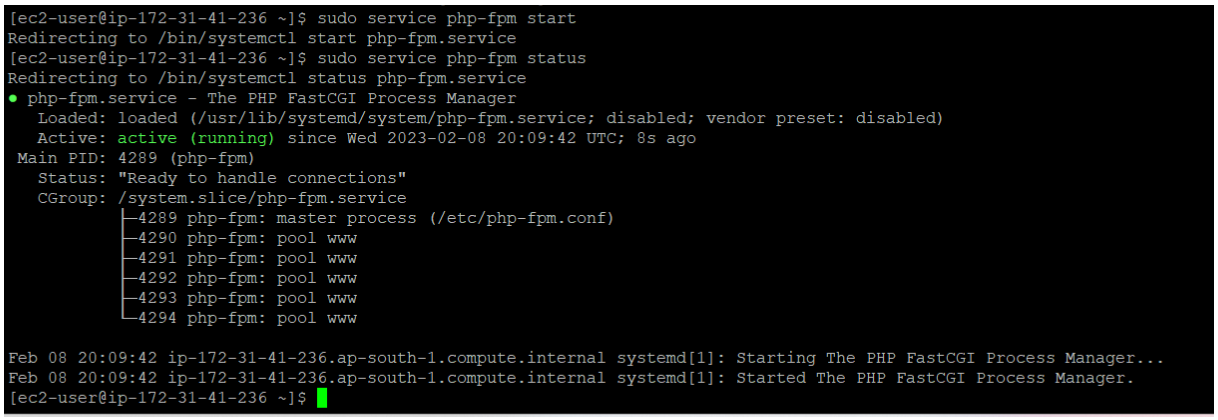
Enter password which you have set in above steps, you have successfully connected to mariadb. To check databases details run command **show databases; .** To exit from mariadb entire **exit** and press enter. M part is completed for LAMP stack.

Step 4 : Install and Run php in Amazon Linux OS

To install php in amazon linux OS packet manage is amazon-linux-extras and php8.0 is it will install php version of 8.0

Commands need to be run.

sudo amazon-linux-extras install php8.0 -y
sudo service php-fpm start
sudo service php-fpm status



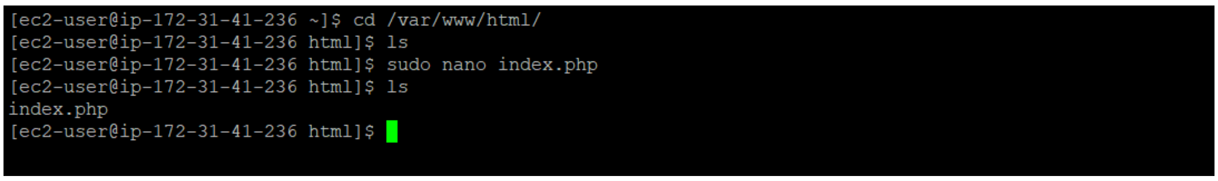
```
[ec2-user@ip-172-31-41-236 ~]$ sudo service php-fpm start
Redirecting to /bin/systemctl start php-fpm.service
[ec2-user@ip-172-31-41-236 ~]$ sudo service php-fpm status
Redirecting to /bin/systemctl status php-fpm.service
● php-fpm.service - The PHP FastCGI Process Manager
   Loaded: loaded (/usr/lib/systemd/system/php-fpm.service; disabled; vendor preset: disabled)
   Active: active (running) since Wed 2023-02-08 20:09:42 UTC; 8s ago
 Main PID: 4289 (php-fpm)
   Status: "Ready to handle connections"
   CGroup: /system.slice/php-fpm.service
           ├─4289 php-fpm: master process (/etc/php-fpm.conf)
           ├─4290 php-fpm: pool www
           ├─4291 php-fpm: pool www
           ├─4292 php-fpm: pool www
           ├─4293 php-fpm: pool www
           └─4294 php-fpm: pool www

Feb 08 20:09:42 ip-172-31-41-236.ap-south-1.compute.internal systemd[1]: Starting The PHP FastCGI Process Manager...
Feb 08 20:09:42 ip-172-31-41-236.ap-south-1.compute.internal systemd[1]: Started The PHP FastCGI Process Manager.
[ec2-user@ip-172-31-41-236 ~]$
```

Php is install and running successfully.

Now go to apache web server directory using its default path /var/www/html and create a php file called index.php to test our php installation and its working.

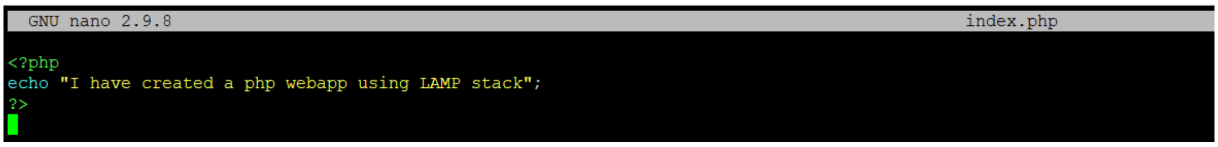
Commands to run

cd /var/www/html/
sudo nano index.php



```
[ec2-user@ip-172-31-41-236 ~]$ cd /var/www/html/
[ec2-user@ip-172-31-41-236 html]$ ls
[ec2-user@ip-172-31-41-236 html]$ sudo nano index.php
[ec2-user@ip-172-31-41-236 html]$ ls
index.php
[ec2-user@ip-172-31-41-236 html]$
```

Gone to html directory and create a php file

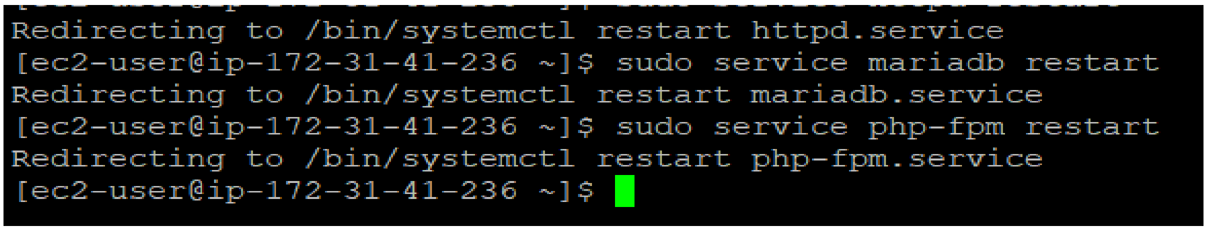


```
  GNU nano 2.9.8                                                                  index.php

<?php
echo "I have created a php webapp using LAMP stack";
?>
```

Added some content in index.php file. M part is completed for LAMP stack.

Now once restart all services using below commands

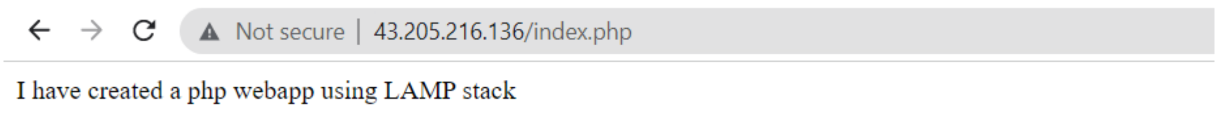
sudo service httpd restart
sudo service mariadb restart
sudo service php-fpm restart



```
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-41-236 ~]$ sudo service mariadb restart
Redirecting to /bin/systemctl restart mariadb.service
[ec2-user@ip-172-31-41-236 ~]$ sudo service php-fpm restart
Redirecting to /bin/systemctl restart php-fpm.service
[ec2-user@ip-172-31-41-236 ~]$
```

Now, copy public IP of your Ec2 instance and past it in web-browser with index.php path.

http://43.205.216.136/index.php



Not secure | 43.205.216.136/index.php

I have created a php webapp using LAMP stack

Yeappp..! LMAP stack is install successfully and php page is running successfully.