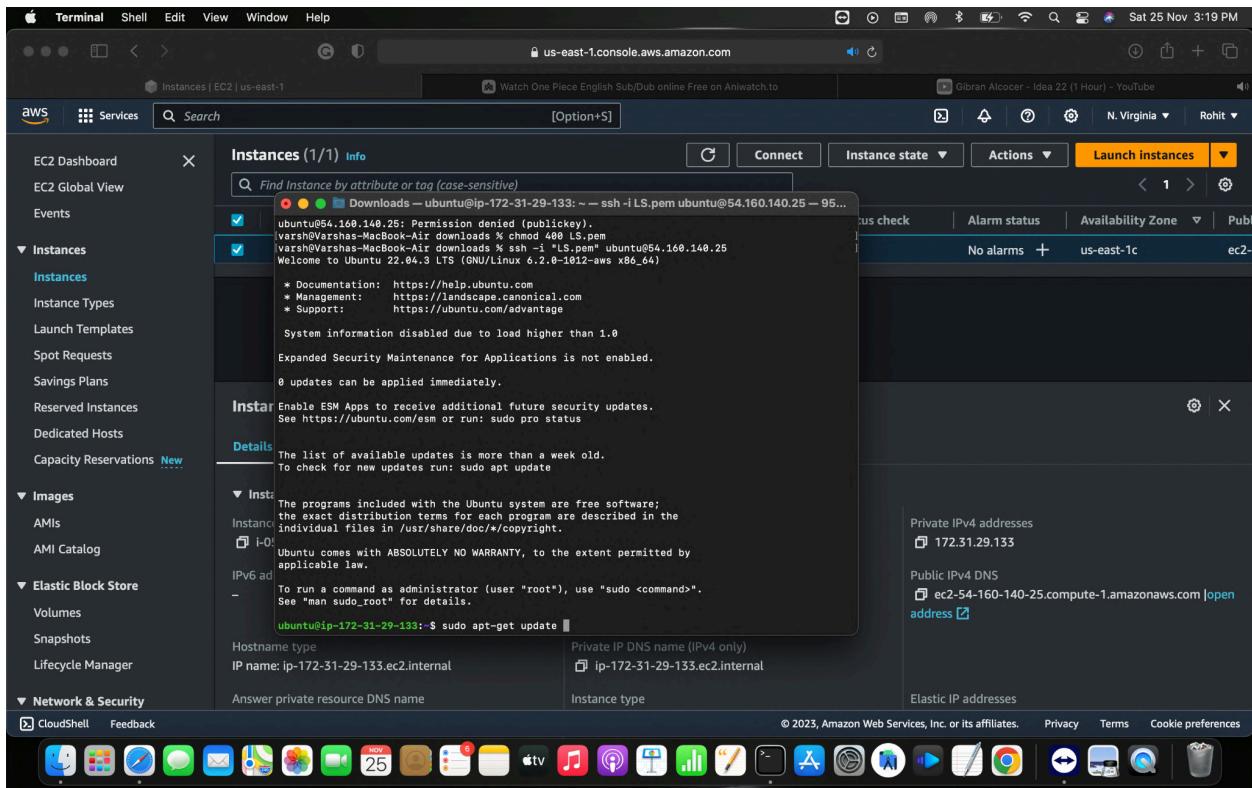


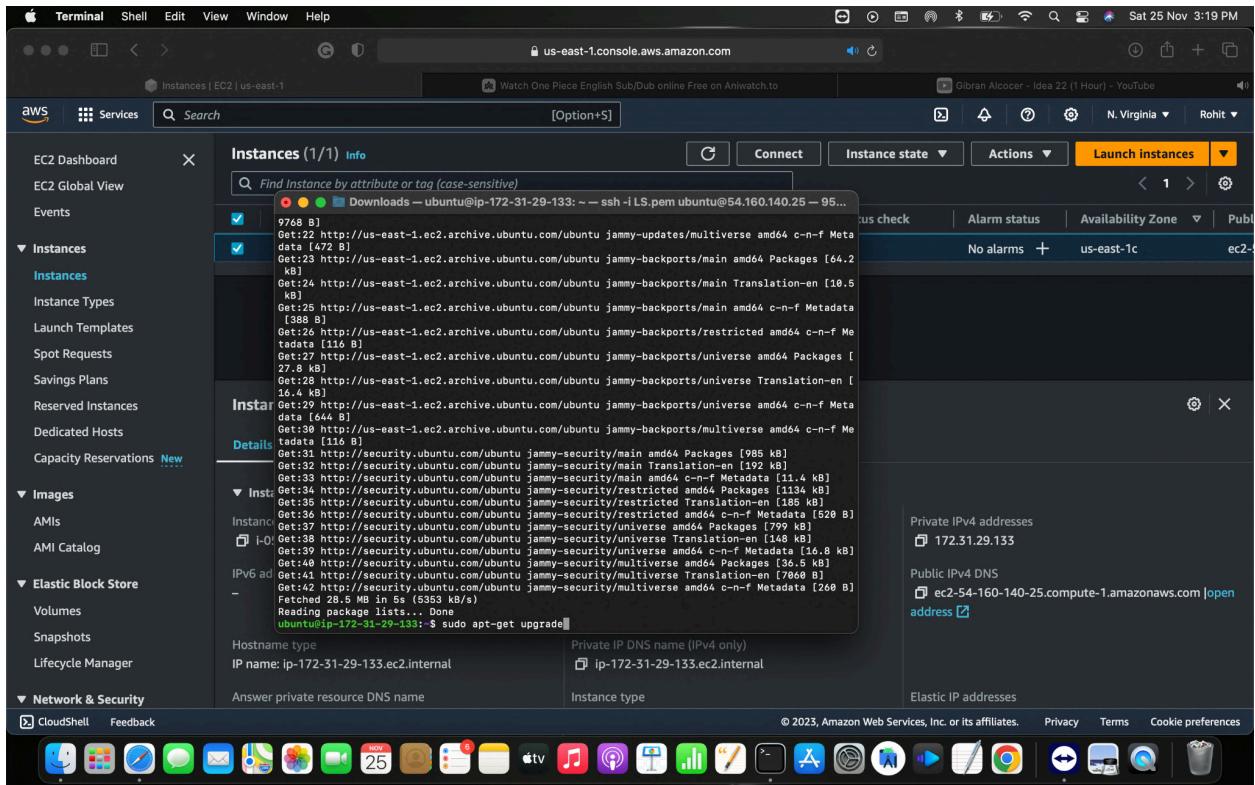
Connecting LAMP stack on EC2 instance and installing wordpress

1 - create an instance of whatever OS you prefer, **Ubuntu or linux**. Then connect the instance through cmd or mac terminal. I am using ubuntu. For that go to your cmd or mac terminal. Then fire the **cd downloads** command to locate where the key pair is. Then fire **ssh -i "yourkeyname" username@publicipaddress** command for connecting.

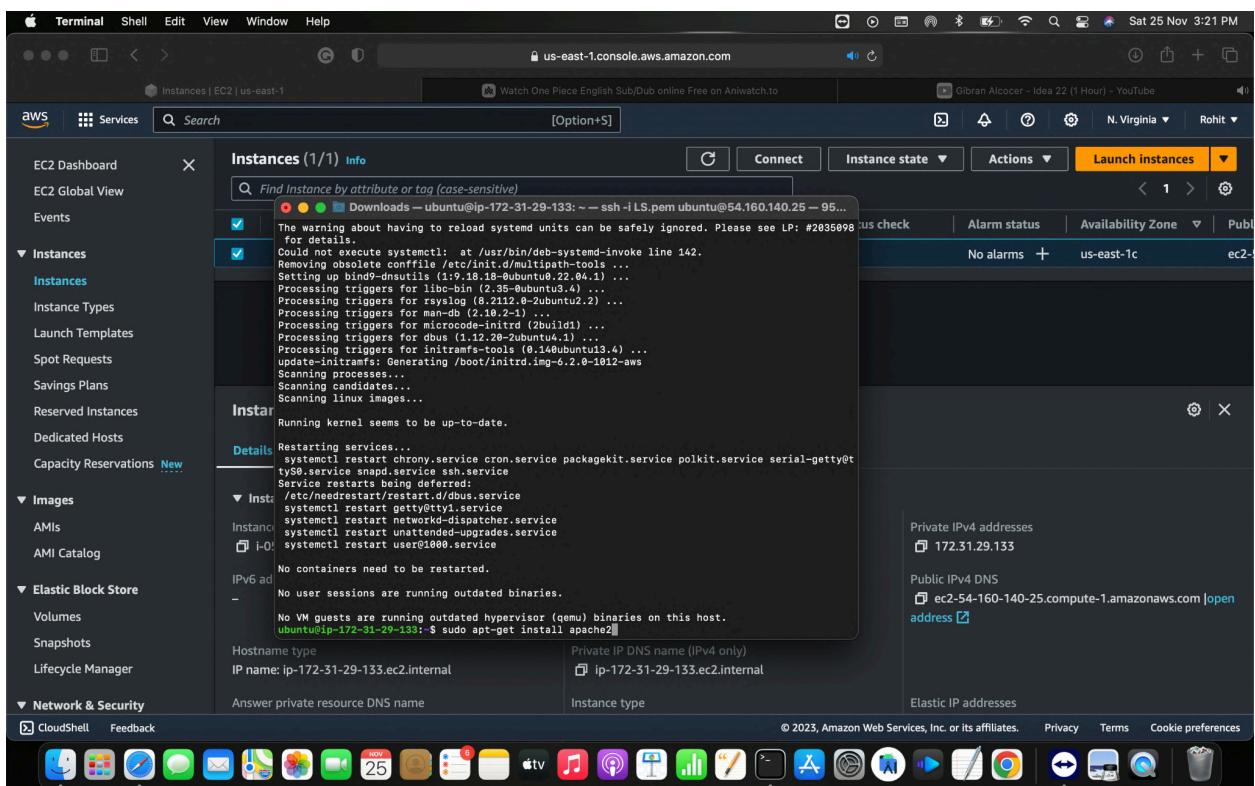
2 - after successfully connecting, first update the ubuntu. For that, fire the **sudo apt-get update** command.



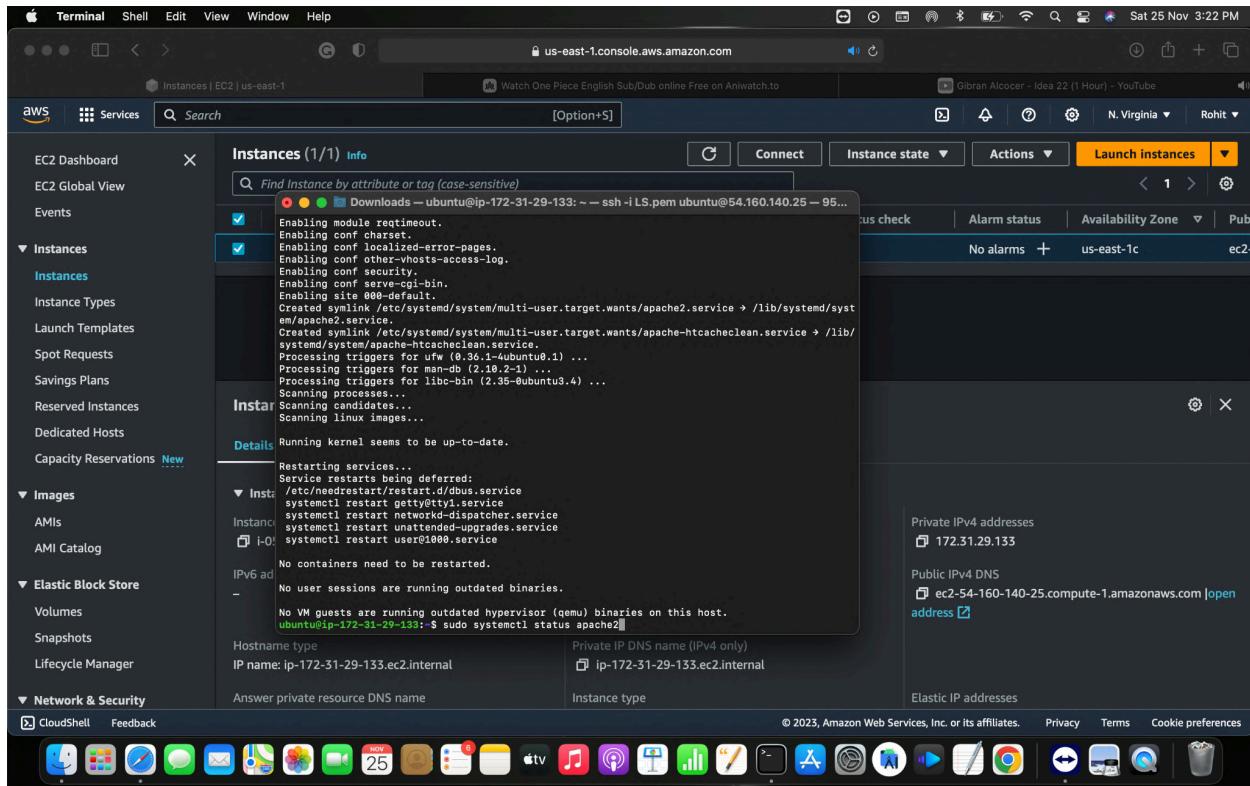
3 - then upgrade it. For that fire **sudo apt-get upgrade** command.



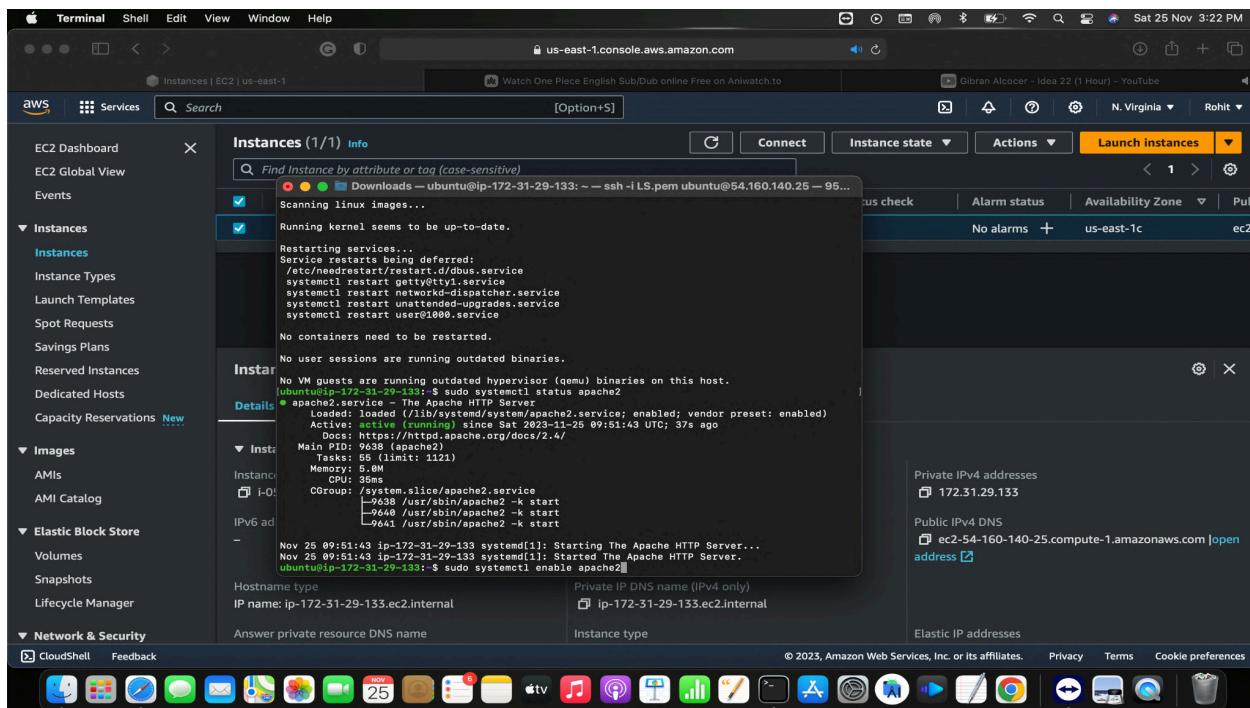
4 - Now, first install apache. For that fire **sudo apt-get install apache2** command.



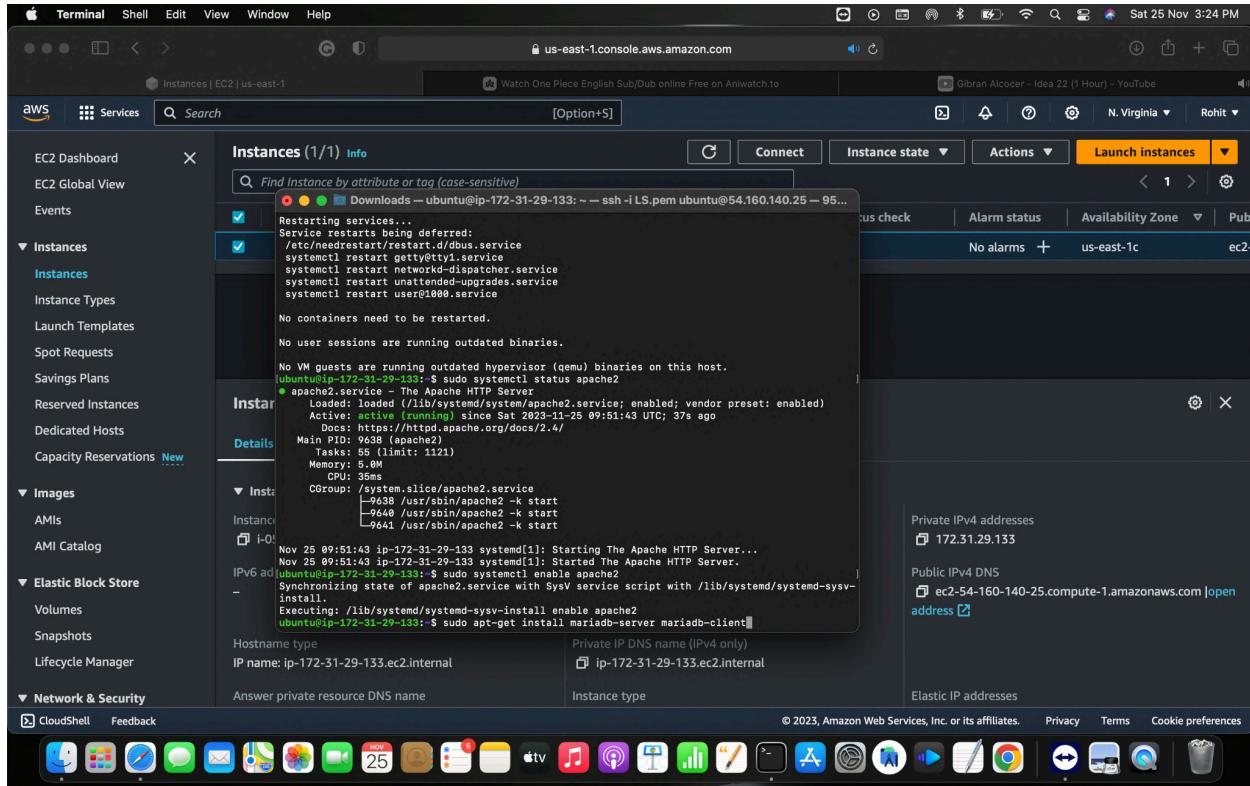
5 - then check the status if it's active or not by firing **sudo systemctl status apache2** command.



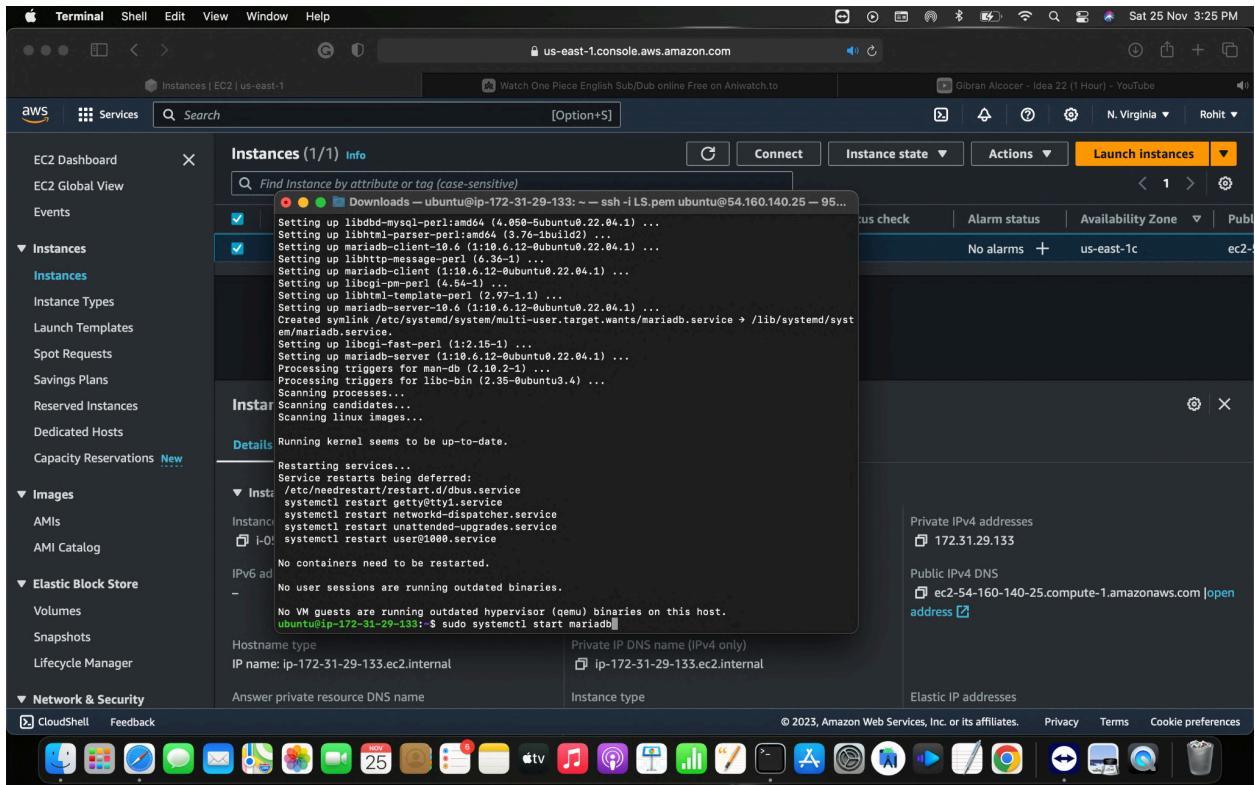
6 - then enable it by firing **sudo systemctl enable apache2** command.



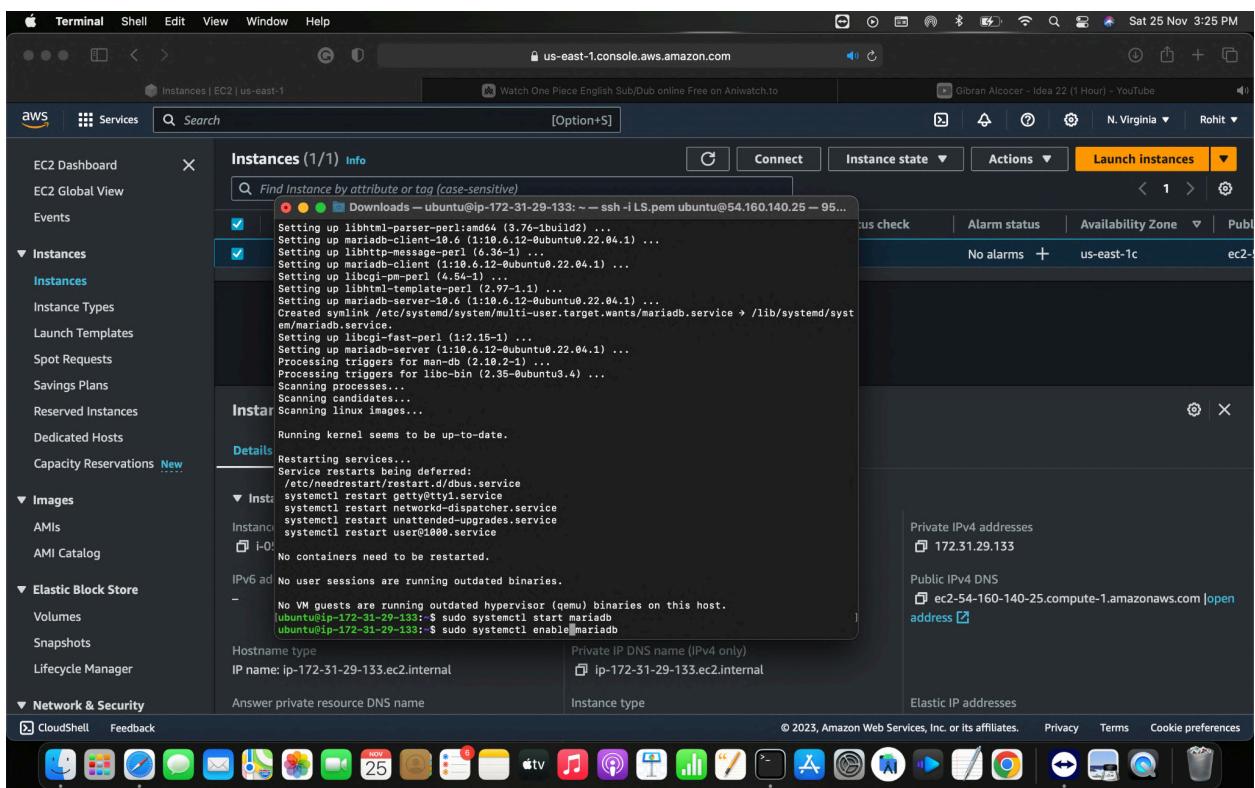
7 - then we are going to install **mariadb**. MariaDB is an open source and forever-free MySQL alternative that offers greater efficiency, enhanced database performance, and support for various data types through multiple storage engines. For installing mariadb, fire **sudo apt-get install mariadb-server mariadb-client** command.



8 - after installing, start mariadb by firing **sudo systemctl start mariadb** command.



9 - then enable mariadb by firing **sudo systemctl enable mariadb** command.



10 - Now, we are going to install **mysql**. For that fire **sudo mysql_secure_installation** command.

The screenshot shows a Mac OS X desktop environment with a Terminal window open in the CloudShell interface. The terminal session is connected to an EC2 instance in the us-east-1 region. The user has run the command `sudo mysql_secure_installation`, which outputs the following text:

```
Setting up libcgpm-perl (4.54-1) ...
Setting up libcgpm-temperate-perl (2.07-1.1) ...
Setting up mariadb-server-10.0 (1:10.0.12-0ubuntu0.22.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /lib/systemd/system/mariadb.service.
Setting up libcgpm-fast-perl (1:2.15-1) ...
Setting up mariadb-server (1:10.6.12-0ubuntu0.22.04.1) ...
Processing triggers for man-db (2.18.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.4) ...
Scanning processes...
Scanning candidates...
Scanning linux images...
Running kernel seems to be up-to-date.

Insta[REDACTED] Restarting services...
Service restarts being deferred:
/etc/init.d/dbus.service
systemctl restart getty@tty1.service
systemctl restart networkd-dispatcher.service
systemctl restart unattended-upgrades.service
systemctl restart user@1000.service
Insta[REDACTED] No containers need to be restarted.

Instanc[REDACTED] No user sessions are running outdated binaries.

i-0[REDACTED] No VM guests are running outdated hypervisor (qemu) binaries on this host.

IPV6 ad[REDACTED] [REDACTED]

Hostname type IP name: ip-172-31-29-133.ec2.internal Private IP DNS name (IPv4 only)
Instance type
```

The terminal window also displays the AWS CloudShell interface, showing the AWS logo, search bar, and various service links like EC2 Dashboard, Services, and Instances. The status bar at the bottom indicates the date and time as Sat 25 Nov 3:26 PM.

11 - when installing, create a password for mariadb as proceeding. After creating the password click on **y=>y=>y=>y** and mariadb and mysql is installed.

```
Setting up libgio-html-perl (1.084+2) ...
Setting up libglib-perl:amd64 (1.643-3build3) ...
Setting up libglib2.0-perl:amd64 (72.1-1) ...
Setting up libhttpdate-perl (2.3300+2) ...
Setting up liblbdctl6:amd64 (72.1-1) ...
Setting up liblfcgi-perl:amd64 (0.82+ds-1build1) ...
Setting up libluring2:amd64 (2.1-2build1) ...
Setting up libpmem1:amd64 (1.11.1-3build1) ...
Setting up libpmix-perl:amd64 (1.643-3build3) ...
Setting up mariadb-server-core-10.6 (1:10.6.12-0ubuntu0.22.04.1) ...
Setting up mariadb-client-perl (6.05-1) ...
Setting up mariadb-client-core-10.6 (1:10.6.12-0ubuntu0.22.04.1) ...
Setting up liblbbd-mysql-perl:amd64 (4.058-Subuntu0.22.04.1) ...
Setting up libhtml-parser-perl:amd64 (3.76-1build2) ...
Setting up mariadb-client-10.6 (1:10.6.12-0ubuntu0.22.04.1) ...
Setting up mariadb-client-core-10.6 (1:10.6.12-0ubuntu0.22.04.1) ...
Setting up mariadb-client-libs (1:10.6.12-0ubuntu0.22.04.1) ...
Setting up mariadb-client-perl (4.54-1) ...
Setting up libhtml-template-perl (2.97-1.1) ...
Setting up mariadb-server-10.6 (1:10.6.12-Subuntu0.22.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /lib/systemd/system/mariadb.service.

Instance Type: t2.micro
Launch Type: On-Demand
Processing triggers for libc-bin (2.35-0ubuntu3.4) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for mariadb (1:10.6.12-0ubuntu0.22.04.1) ...
Spot Request: Scanning processes...
Savings Plan: Scanning candidates...
Scanning linux images...

Running kernel seems to be up-to-date.

Dedicated: Restarting services...
Capacity Rebalance: Service restarts being deferred:
    /etc/neorestart/restart.d/dbus.service
    systemctl restart getty@tty1.service
    systemctl restart networkd-dispatcher.service
    systemctl restart unattended-upgrades.service
    systemctl restart user@1000.service

Images: No containers need to be restarted.

AMI Catalog: No user sessions are running outdated binaries.

Elastic Block Store: No VM guests are running outdated hypervisor (qemu) binaries on this host.

Volumes: NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!
Snapshots: 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 haven't set the root password yet, you should just press enter here.

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

```
Setting up liburi-perl (5.10-1) ...
Setting up liblbbi-perl:amd64 (1.643-3build3) ...
Setting up mariadb-server-core-10.6 (1:10.6.12-0ubuntu0.22.04.1) ...
Setting up libhttp-date-perl (6.05-1) ...
Setting up mariadb-client-core-10.6 (1:10.6.12-0ubuntu0.22.04.1) ...
Setting up liblbbd-mysql-perl:amd64 (4.058-Subuntu0.22.04.1) ...
Setting up mariadb-client-libs (1:10.6.12-0ubuntu0.22.04.1) ...
Setting up mariadb-client-perl (6.36-1) ...
Setting up mariadb-client-perl (6.36-1) ...
Setting up libhttp-message-perl (4.54-1) ...
Setting up libhtml-template-perl (2.97-1.1) ...
Setting up mariadb-server-10.6 (1:10.6.12-Subuntu0.22.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /lib/systemd/system/mariadb.service.

Instance Type: t2.micro
Launch Type: On-Demand
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.4) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Running kernel seems to be up-to-date.

Dedicated: Restarting services...
Capacity Rebalance: Service restarts being deferred:
    /etc/neorestart/restart.d/dbus.service
    systemctl restart getty@tty1.service
    systemctl restart networkd-dispatcher.service
    systemctl restart unattended-upgrades.service
    systemctl restart user@1000.service

Images: No user sessions are running outdated binaries.

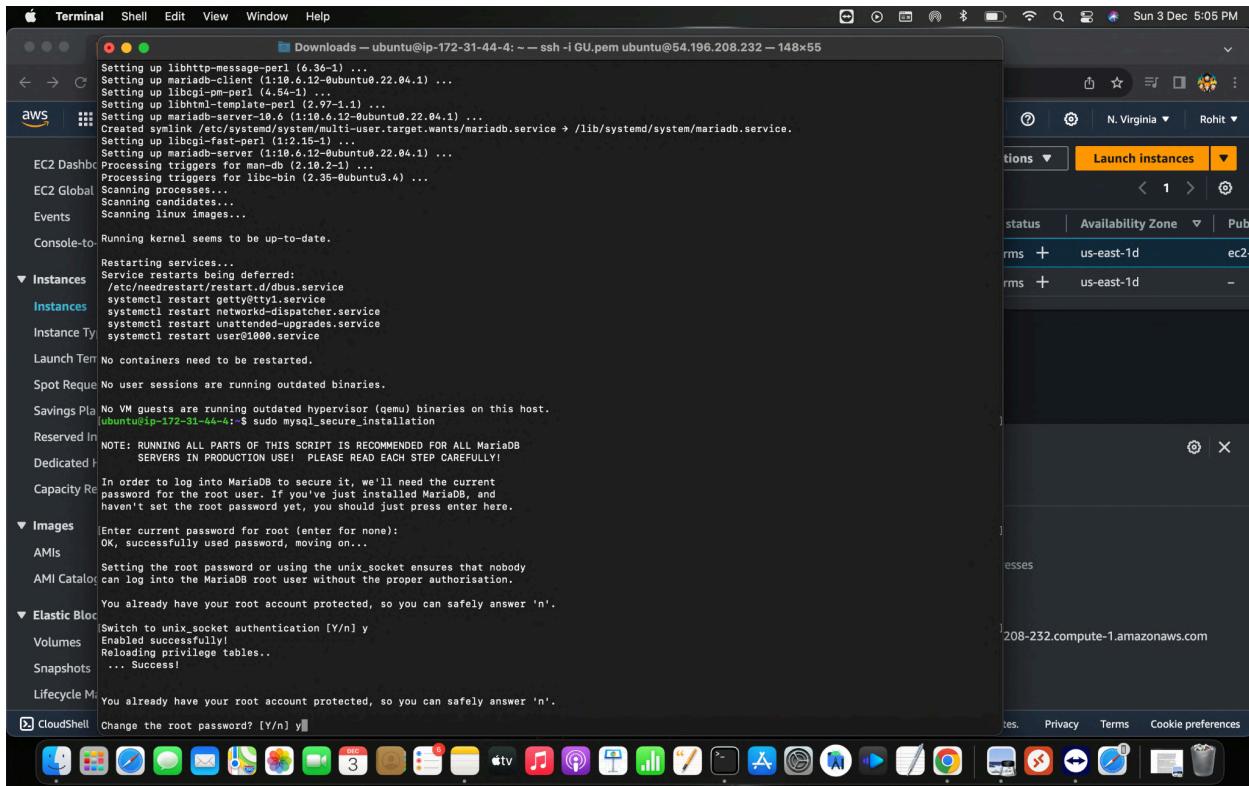
Elastic Block Store: No VM guests are running outdated hypervisor (qemu) binaries on this host.

Volumes: NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!
Snapshots: 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 haven't set the root password yet, you should just press enter here.

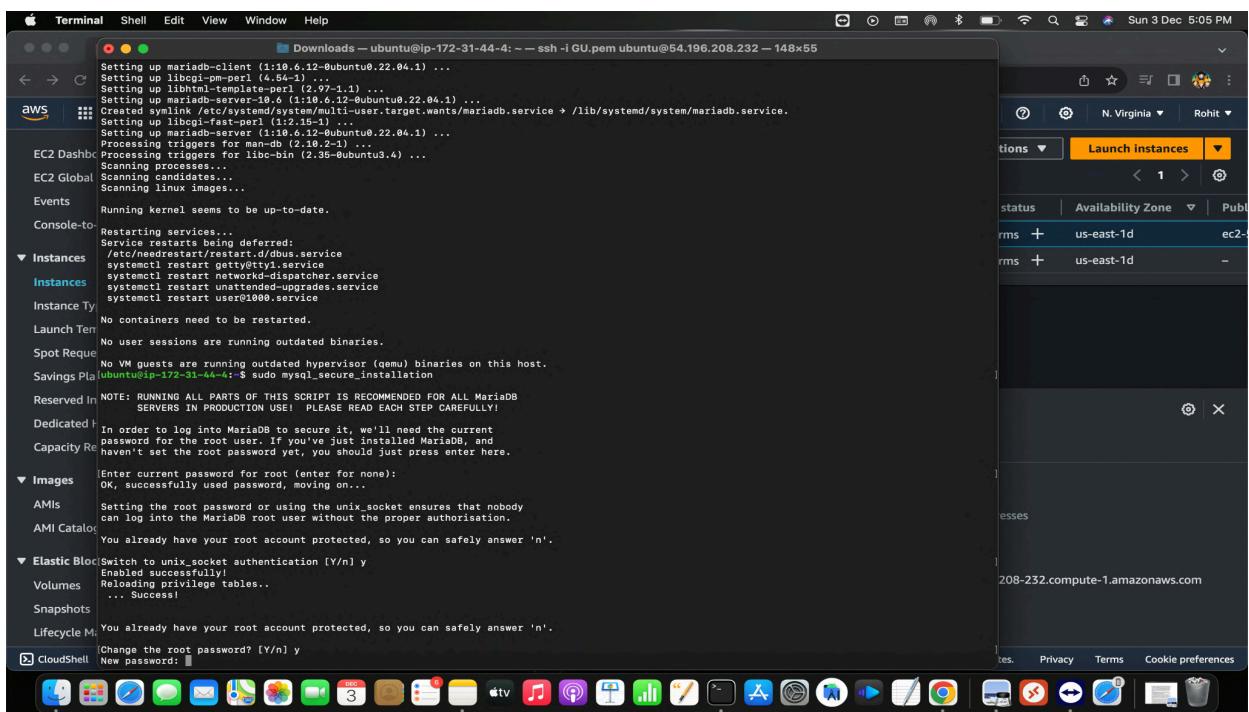
CloudShell: Enter current password for root (enter for none): 
[ubuntu@ip-172-31-44-4: ~ ssh -i GU.pem ubuntu@54.196.208.232 - 148x55] $ sudo mysql_secure_installation
OK, successfully used password, moving on...
Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.
CloudShell: Switch to unix_socket authentication [Y/n] y
```

12 - this is where you will change the password. Type **y** and hit enter.



13 - Write a new password and hit enter and again rewrite the password and hit enter.



14 - After this, again keep typing Y and hit enter.

The screenshot shows a Mac desktop with two terminal windows and the AWS Management Console. The top terminal window is titled 'Downloads — ubuntu@ip-172-31-44-4: ~ — ssh -i GU.pem ubuntu@54.196.208.232 — 148x55'. It displays the output of a command to secure a MySQL installation, including prompts for root password changes and privilege table reloads. The bottom terminal window is identical, showing the same command and output. The AWS Management Console sidebar is visible on the right, showing 'Instances' and 'CloudShell' sections.

```
Running kernel seems to be up-to-date.
Restarting services...
Service restarts being deferred:
/etc/init.d/dbus.service
systemctl restart getty@tty1.service
systemctl restart networkd-dispatcher.service
systemctl restart unattended-upgrades.service
systemctl restart user@1000.service

EC2 Dashboard
EC2 Global
Events
Console-to-Instances
Instances
Launch Template
Spot Request
Savings Plan
Reserved Instances
Dedicated Instances
Capacity Reserve
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Events
CloudShell
```

```
aws
```

```
EC2 Dashboard
EC2 Global
Events
Console-to-Instances
Instances
Launch Template
Spot Request
Savings Plan
Reserved Instances
Dedicated Instances
Capacity Reserve
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Events
CloudShell
```

```
ubuntu@ip-172-31-44-4: ~ $ 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
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables...
... Success!
```

```
ubuntu@ip-172-31-44-4: ~ $ 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
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables...
... Success!
```

```
CloudShell
```

```
Remove anonymous users? [Y/n] y
```

```
Disallow root login remotely? [Y/n] y
```

The screenshot shows a Mac desktop with two terminal windows and the AWS Management Console. The top terminal window is titled 'Downloads — ubuntu@ip-172-31-44-4: ~ — ssh -i GU.pem ubuntu@54.196.208.232 — 148x55'. It displays the output of a command to secure a MySQL installation, including prompts for root password changes and privilege table reloads. The bottom terminal window is identical, showing the same command and output. The AWS Management Console sidebar is visible on the right, showing 'Instances' and 'CloudShell' sections.

```
Running kernel seems to be up-to-date.
Restarting services...
Service restarts being deferred:
/etc/init.d/dbus.service
systemctl restart getty@tty1.service
systemctl restart networkd-dispatcher.service
systemctl restart unattended-upgrades.service
systemctl restart user@1000.service

EC2 Dashboard
EC2 Global
Events
Console-to-Instances
Instances
Launch Template
Spot Request
Savings Plan
Reserved Instances
Dedicated Instances
Capacity Reserve
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Events
CloudShell
```

```
aws
```

```
EC2 Dashboard
EC2 Global
Events
Console-to-Instances
Instances
Launch Template
Spot Request
Savings Plan
Reserved Instances
Dedicated Instances
Capacity Reserve
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Events
CloudShell
```

```
ubuntu@ip-172-31-44-4: ~ $ 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
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables...
... Success!
```

```
ubuntu@ip-172-31-44-4: ~ $ 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
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables...
... Success!
```

```
CloudShell
```

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

```
Disallow root login remotely? [Y/n] y
```

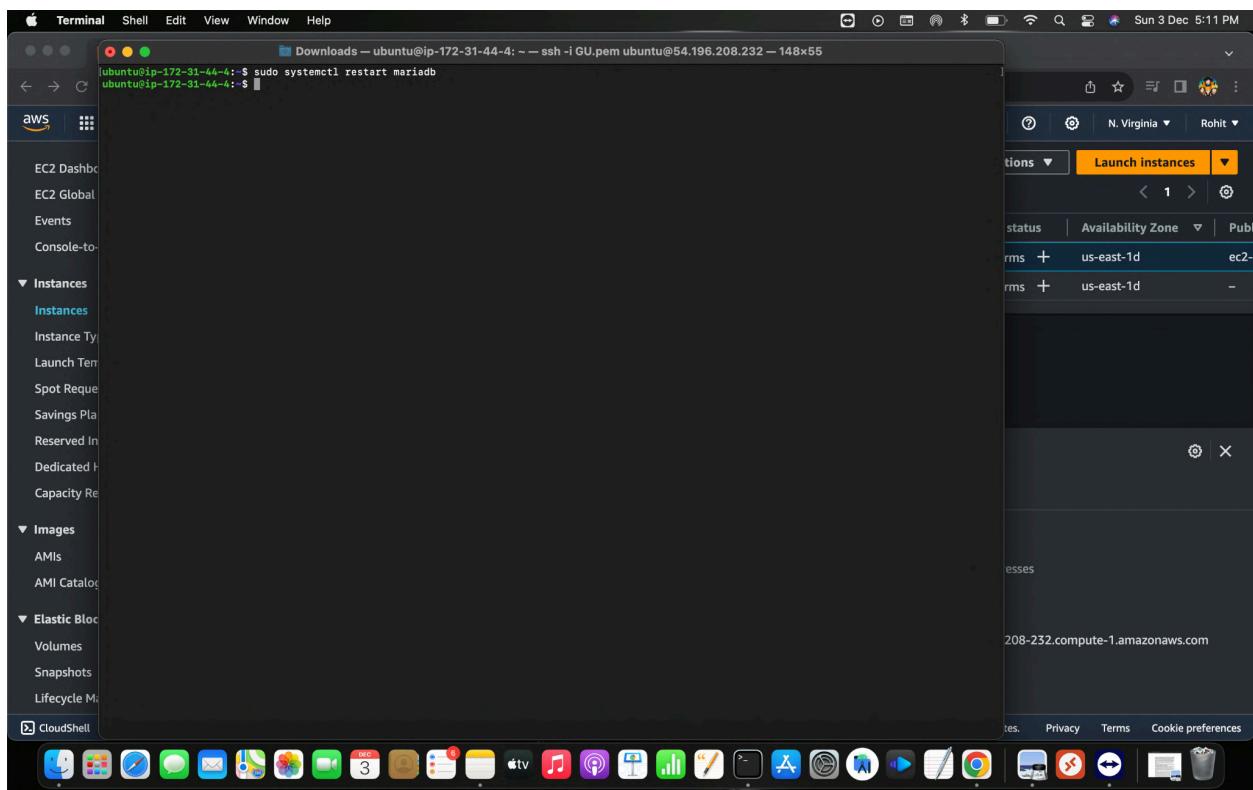
A screenshot of a Mac OS X desktop environment. On the left, a terminal window titled 'Downloads — ubuntu@ip-172-31-44-4: ~ — ssh -i GU.pem ubuntu@54.196.208.232 — 148x55' displays a script for securing a MariaDB installation. The script asks for the current root password, sets a new one, updates privilege tables, and removes anonymous users. It also disables root login from the network and removes the 'test' database. On the right, a browser window shows the AWS CloudWatch Metrics interface for an instance named '208-232.compute-1.amazonaws.com' in the N. Virginia region. The metrics tab is selected, showing various monitoring data over time.

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
[ubuntu@ip-172-31-44-4: ~] $ 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  
haven't set the root password yet, you should just press enter here.  
  
Enter current password for root (enter for none):  
OK, successfully used password, moving on...  
  
Setting the root password or using the unix_socket ensures that nobody  
can log into the MariaDB root user without the proper authorisation.  
  
You already have your root account protected, so you can safely answer 'n'.  
Switch to unix socket authentication [Y/n] y  
Enabled successfully!  
Reloading privilege tables...  
... Success!  
  
Launch Tem...  
Spot Reque...  
Savings Pla...  
Reserved In...  
Dedicated I...  
Capacity Re...  
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.  
  
▼ Instances  
Instances  
Instance Ty...  
Launch Tem...  
Spot Reque...  
Savings Pla...  
Reserved In...  
Dedicated I...  
Capacity Re...  
AMIs  
AMI Catalog...  
... Success!  
  
▼ Elastic Blo...  
Volumes  
Snapshots  
Lifecycle M...  
CloudShell Remove test database and access to it? [Y/n] y  
  
[CloudShell Remove test database and access to it? [Y/n] y]
```

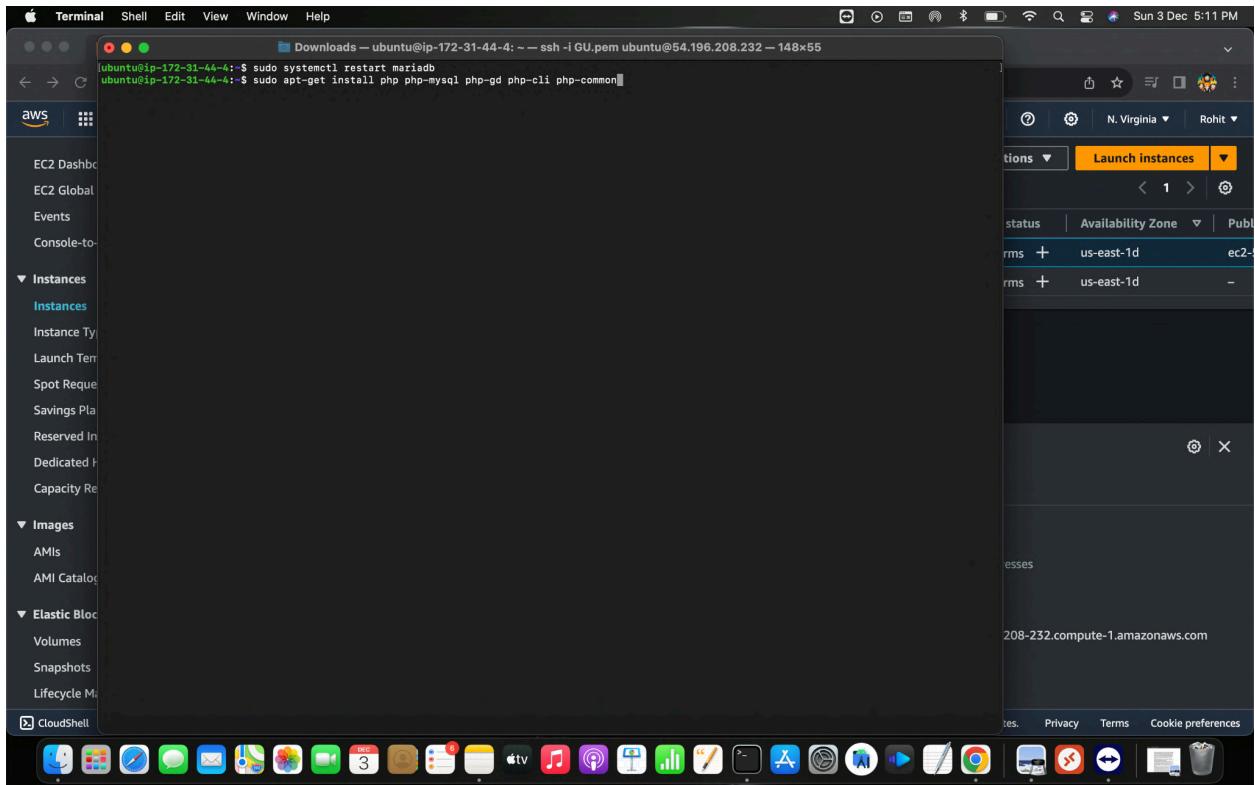
A second screenshot of a Mac OS X desktop environment, identical to the first one. It shows the same terminal window with the MySQL security script running and the same AWS CloudWatch Metrics interface for the instance '208-232.compute-1.amazonaws.com'. The terminal output shows the script has completed its tasks, including dropping the 'test' database and reloading privilege tables.

```
haven't set the root password yet, you should just press enter here.  
[Enter current password for root (enter for none):  
OK, successfully used password, moving on...  
Setting the root password or using the unix_socket ensures that nobody  
can log into the MariaDB root user without the proper authorisation.  
  
EC2 Dashba...  
You already have your root account protected, so you can safely answer 'n'.  
Switch to unix_socket authentication [Y/n] y  
Enabled successfully!  
Reloading privilege tables...  
... Success!  
  
Console-to-...  
Instances  
Instances  
Instance Ty...  
Launch Tem...  
Spot Reque...  
Savings Pla...  
Reserved In...  
Dedicated I...  
Capacity Re...  
AMIs  
AMI Catalog...  
... Success!  
  
▼ Elastic Blo...  
Volumes  
Snapshots  
Lifecycle M...  
CloudShell 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.  
CloudShell Reload privilege tables now? [Y/n] y
```

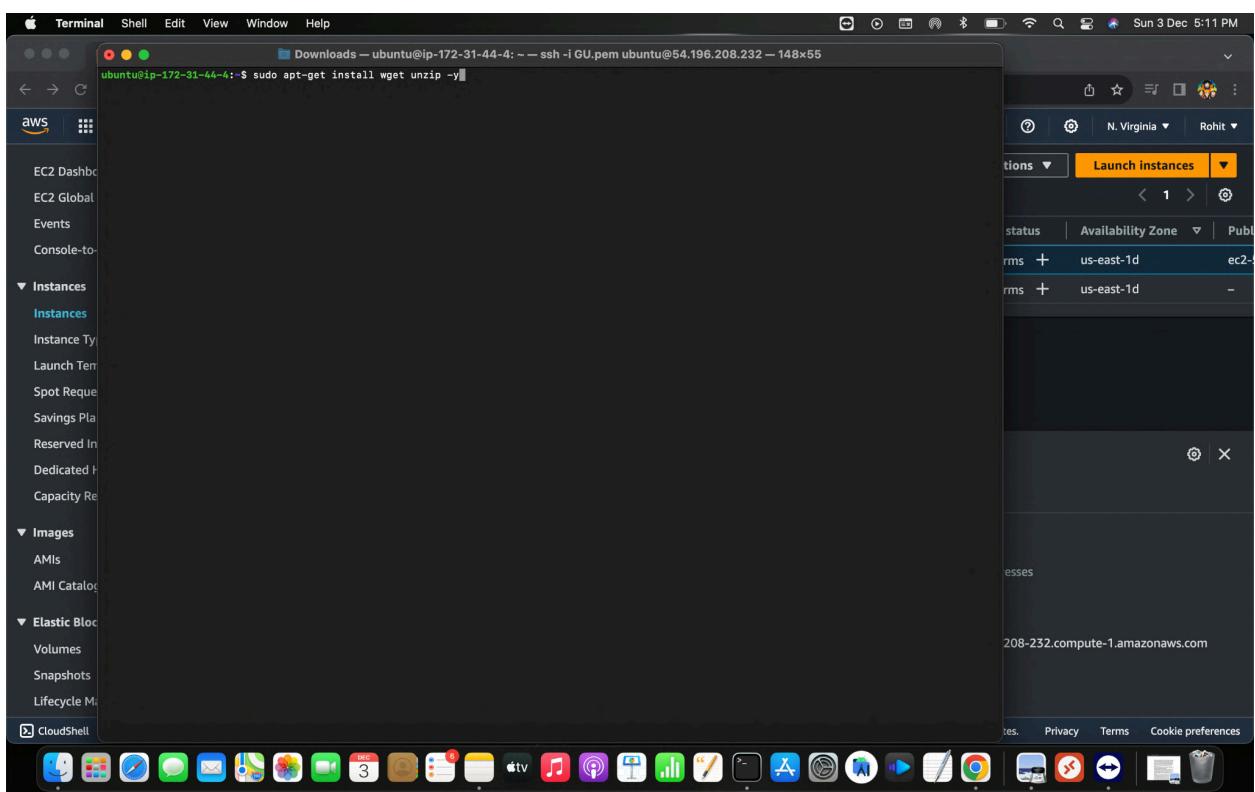
15 - Now, restart mariadb by firing **sudo systemctl restart mariadb** command.



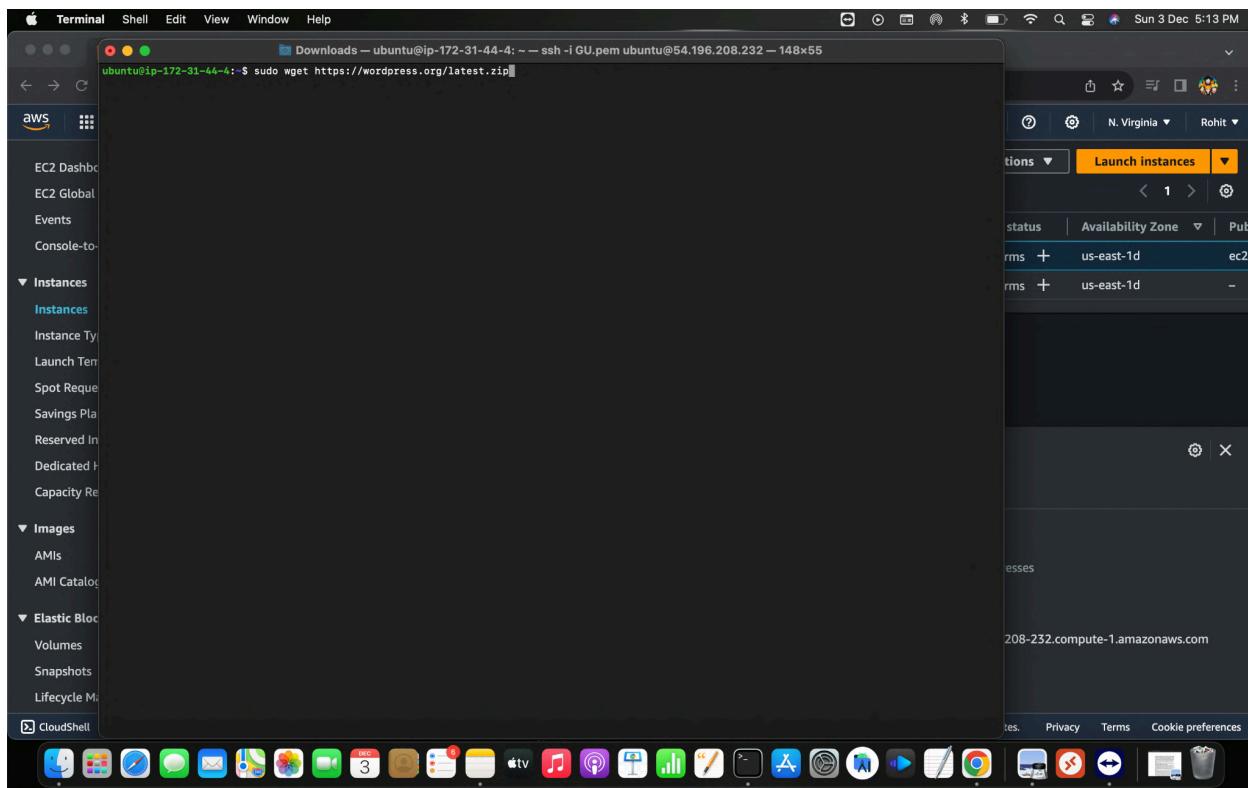
16 - Now, we are going to install **php**. For that fire **sudo apt-get install php php-mysql php-gd php-cli php-common** command.



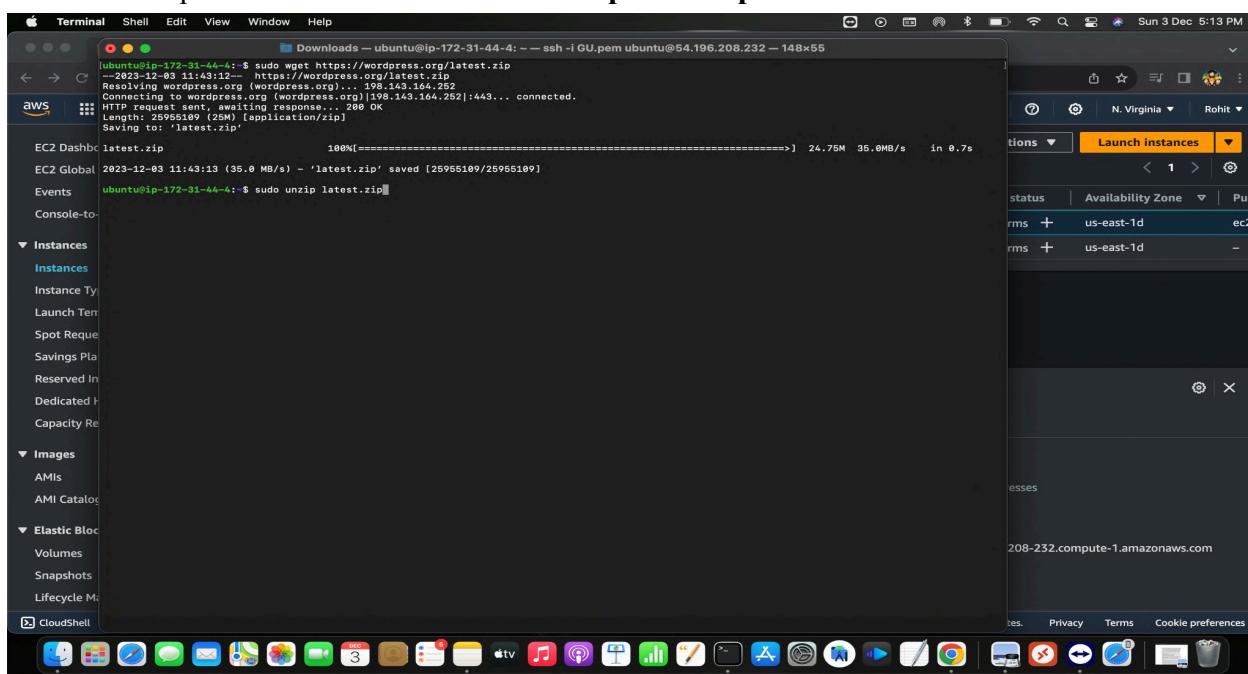
17 - Now, we need wordpress. For that first fire **sudo apt-get install wget unzip -y** command.



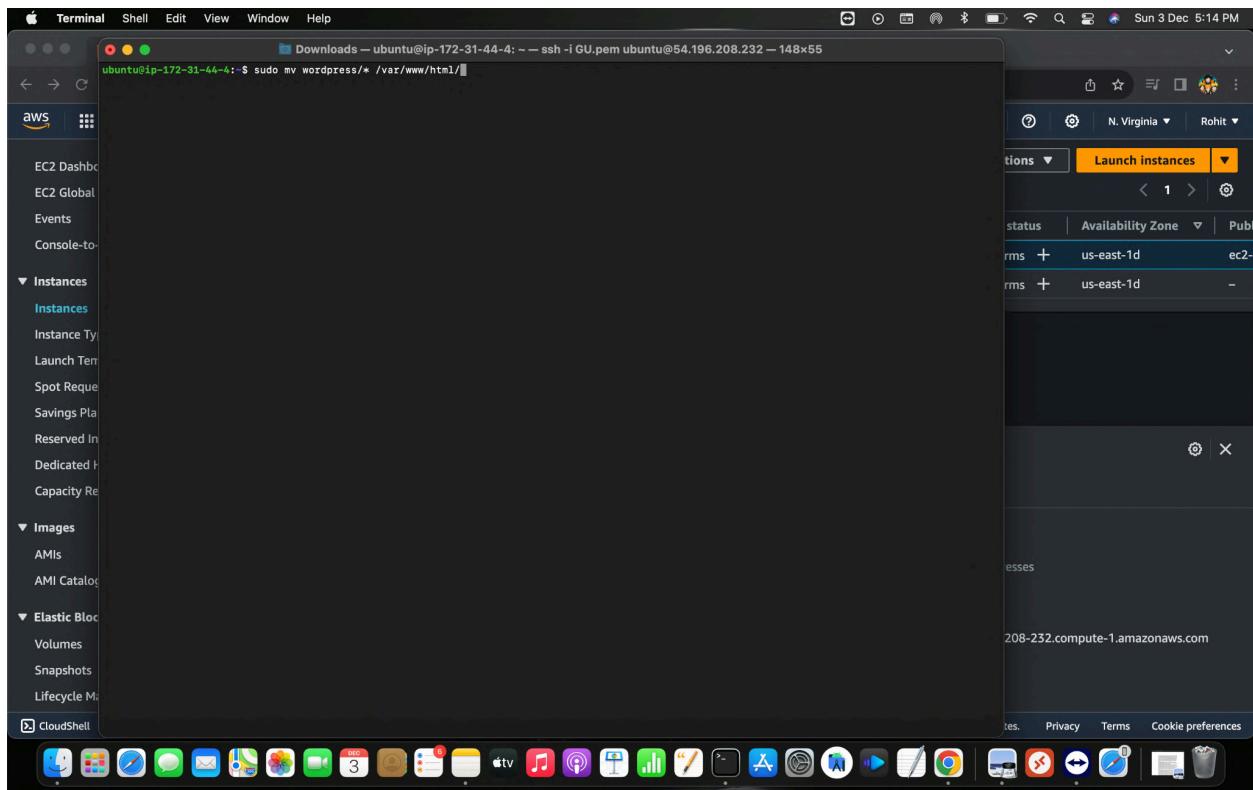
18 - Now, second go to the official website of wordpress and copy the download link of wordpress then we are going to paste it in cmd. After copying, fire **sudo wget pasteyourlink** command.



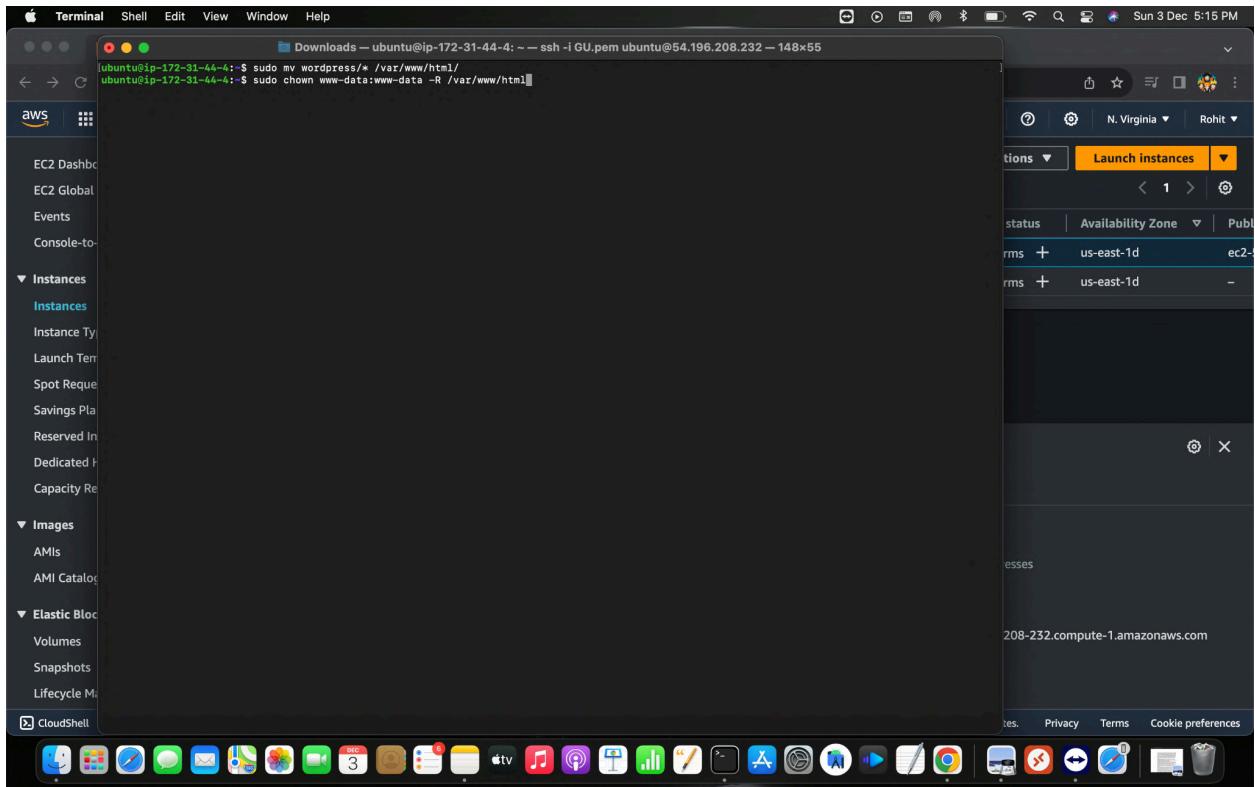
19 - then unzip the link. For that fire **sudo unzip latest.zip** command.



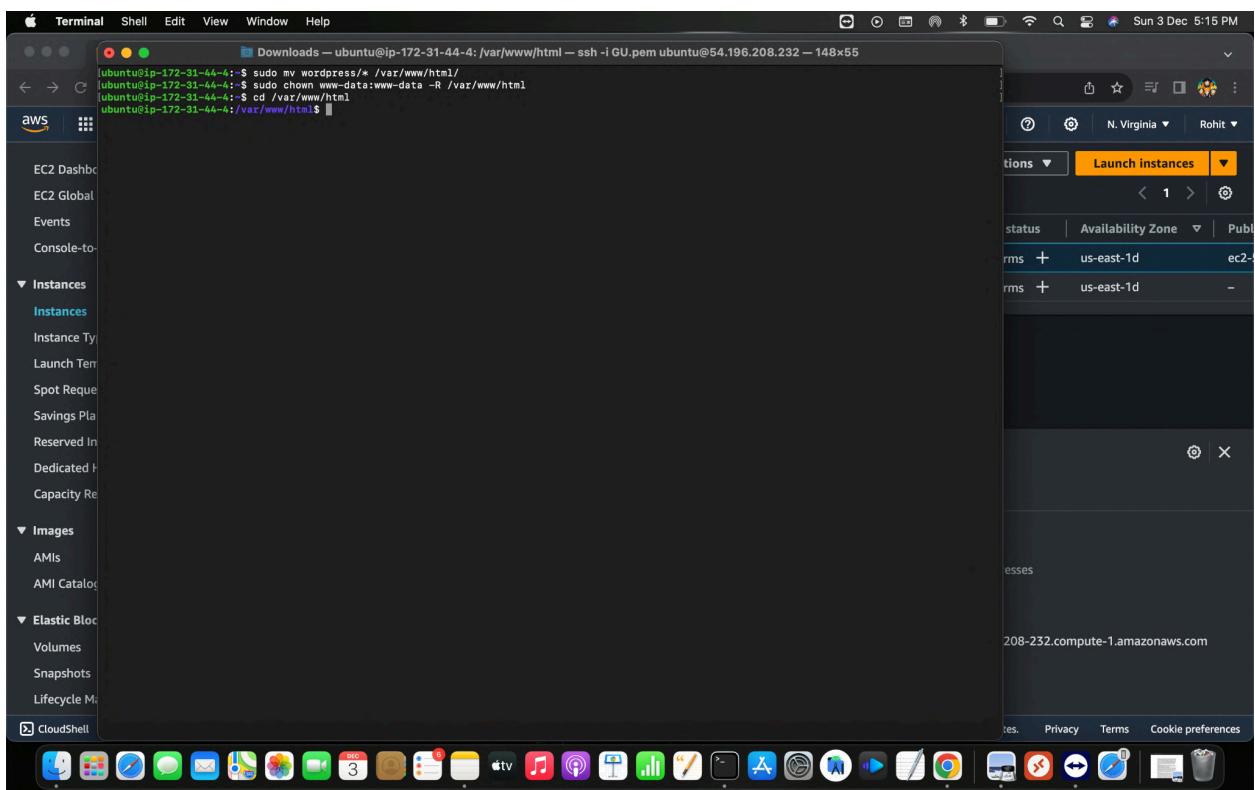
20 - Now move that file to the directory. For that fire **sudo mv wordpress/* /var/www/html/** command.



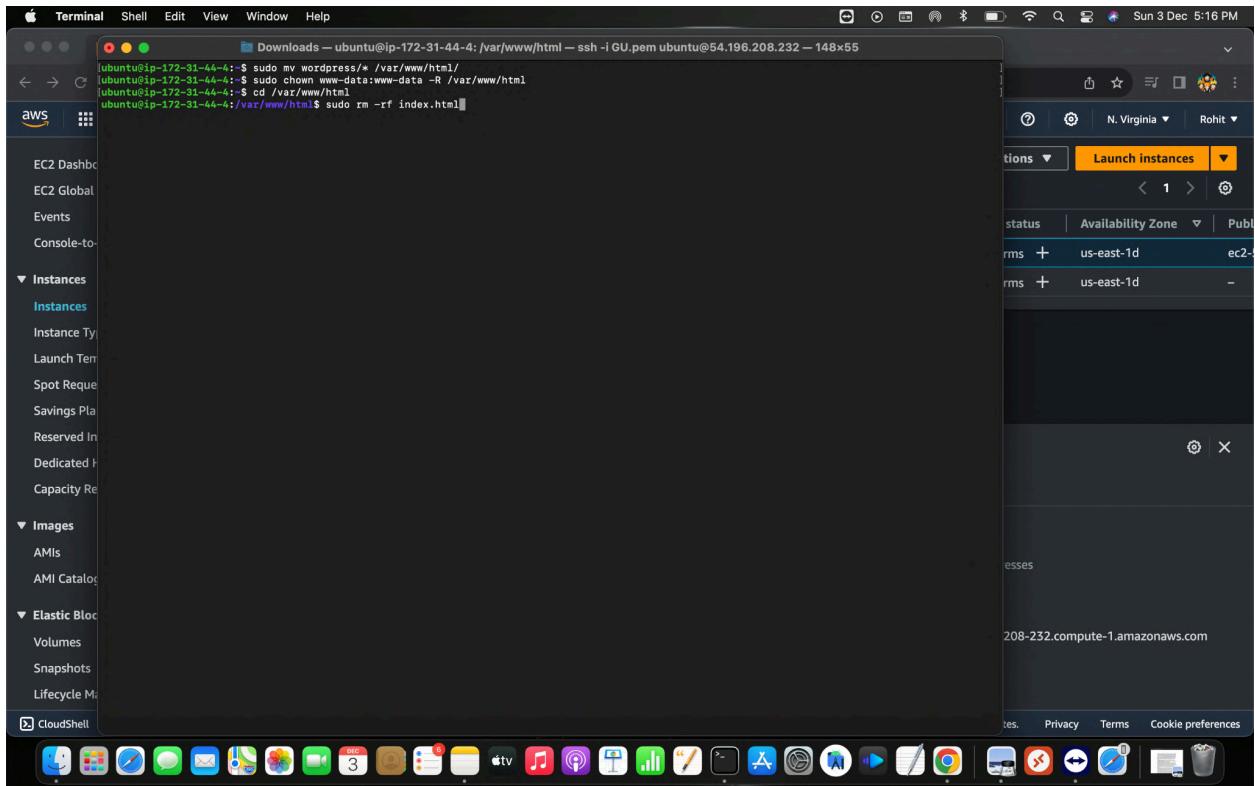
21 - Now, change the ownership of that file by firing **sudo chown www-data:www-data -R /var/www/html/** command.



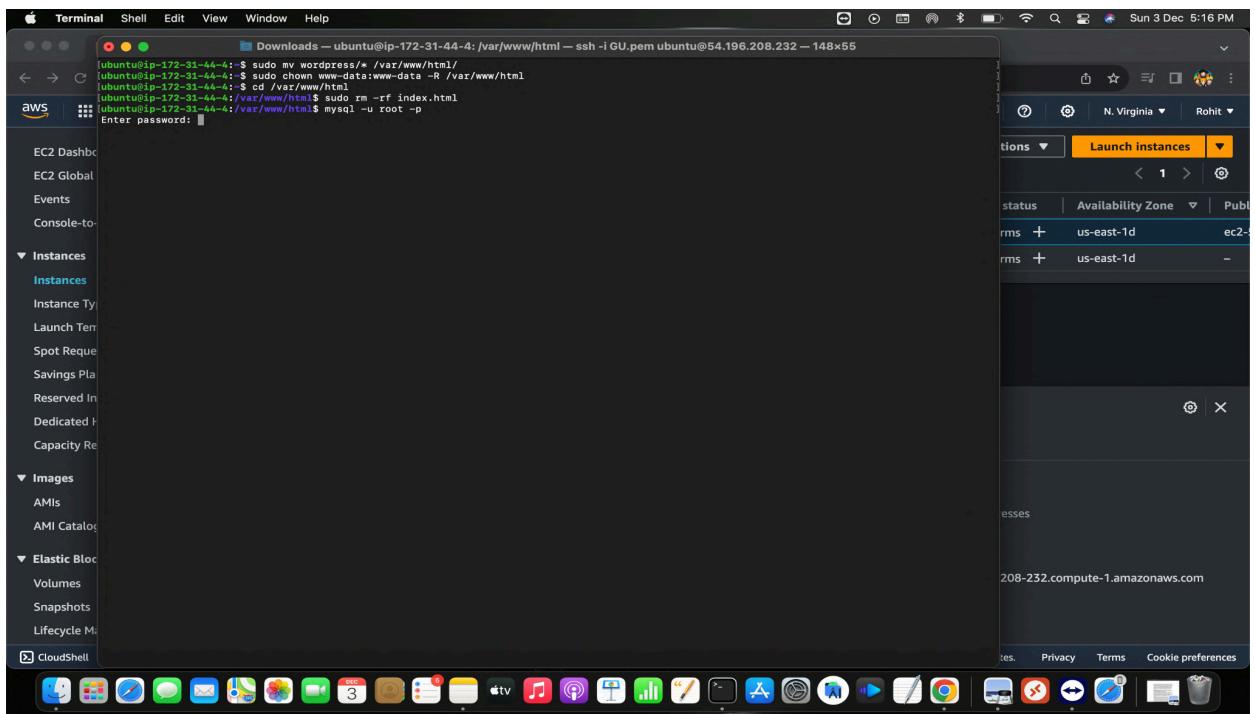
22 - Now, go to the directory. For that fire **cd /var/www/html** command.



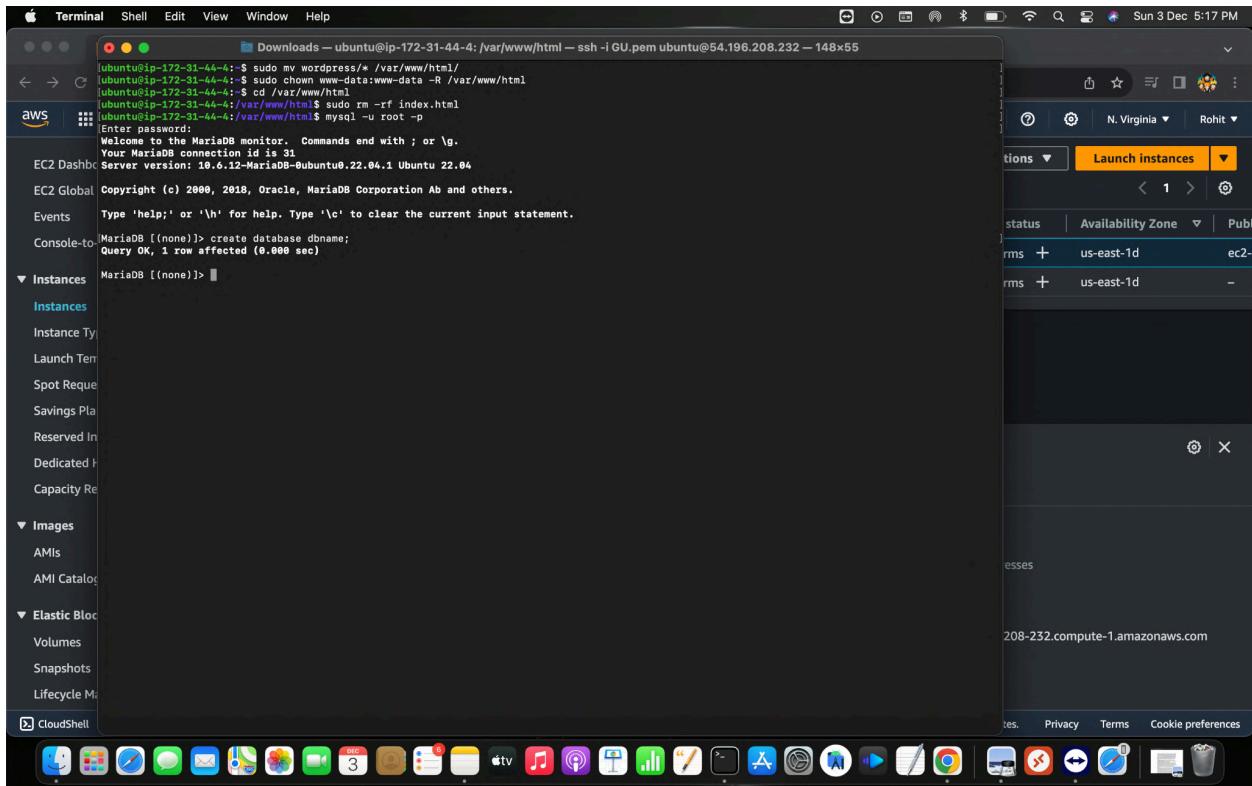
23 - Now, delete the existing index.html file by firing **sudo rm -rf index.html** command.



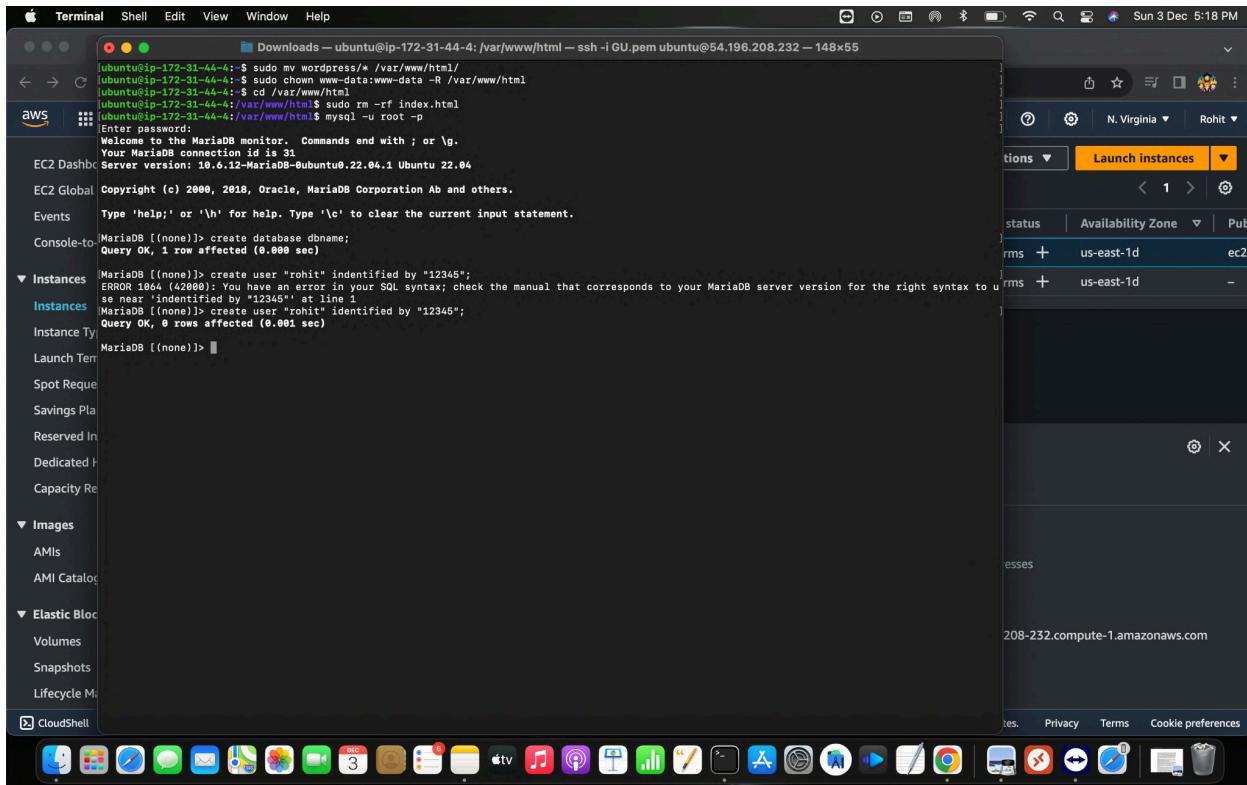
24 - Now, login to mysql by firing **mysql -u root -p** command.



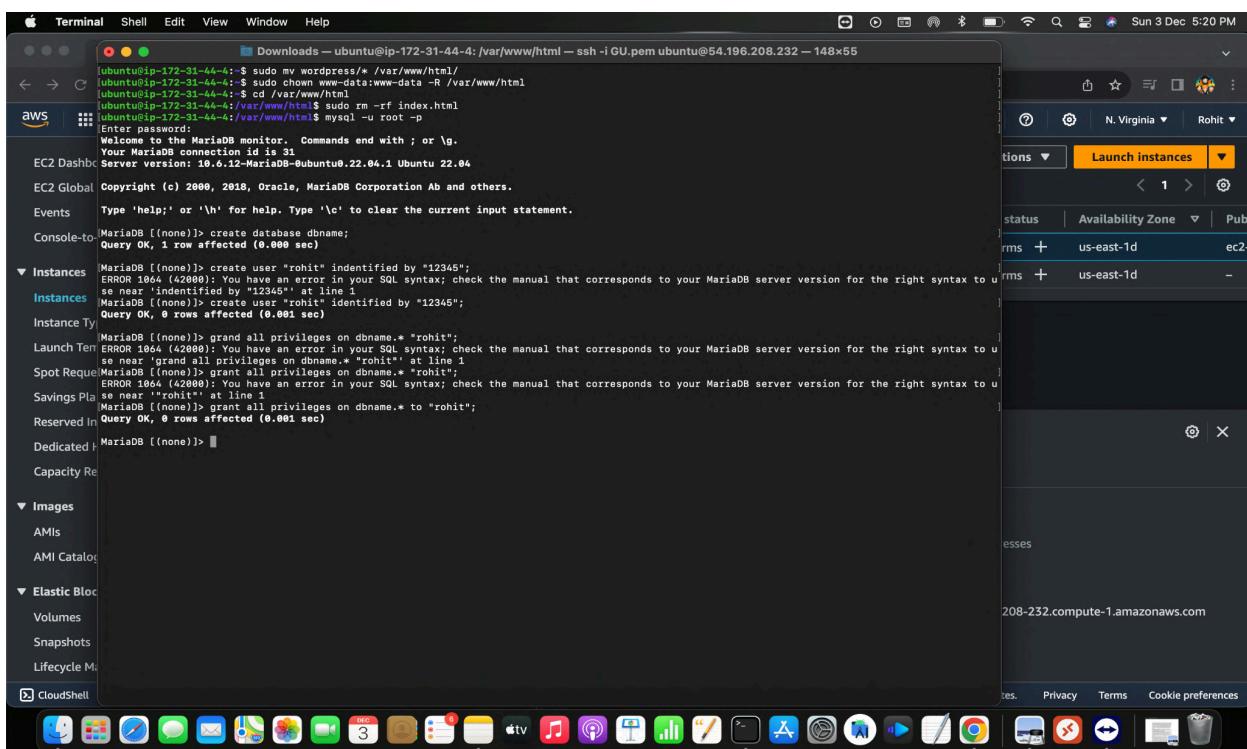
25 - first create a database. Type **create database databasename;** and hit enter.



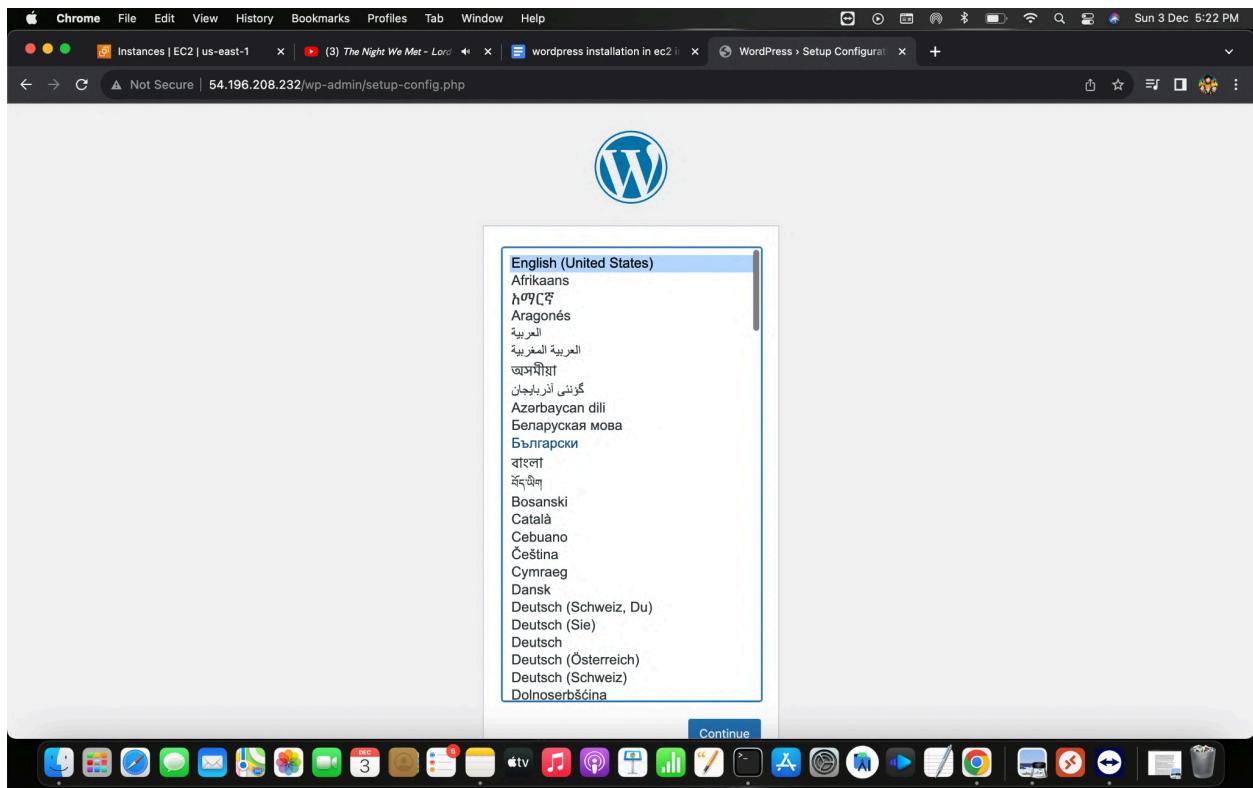
26 - Now, we create a user and create a password. For that, type **create user "username" identified by "password";** and hit enter.



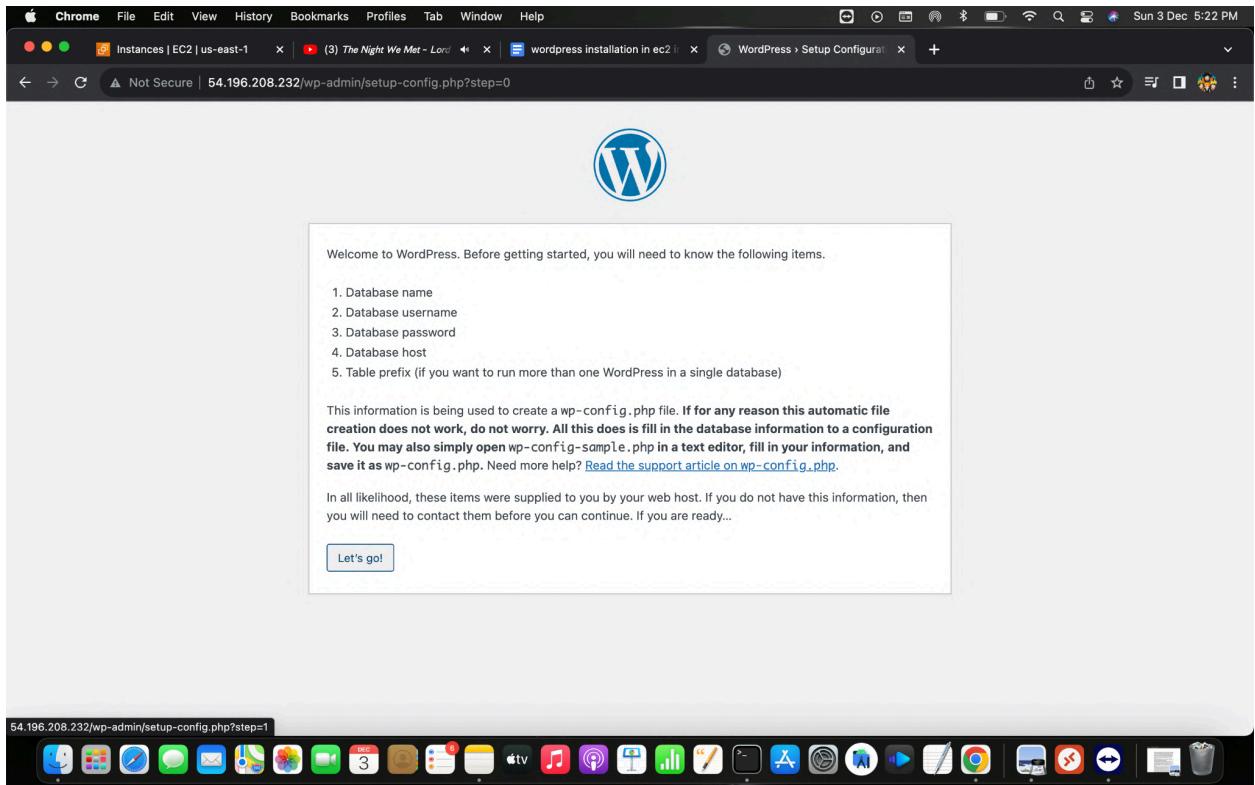
27 - now all privileges to the database. Type **grant all privileges databaseName.* to “username”** and hit enter. And lastly type **quit** and hit enter.



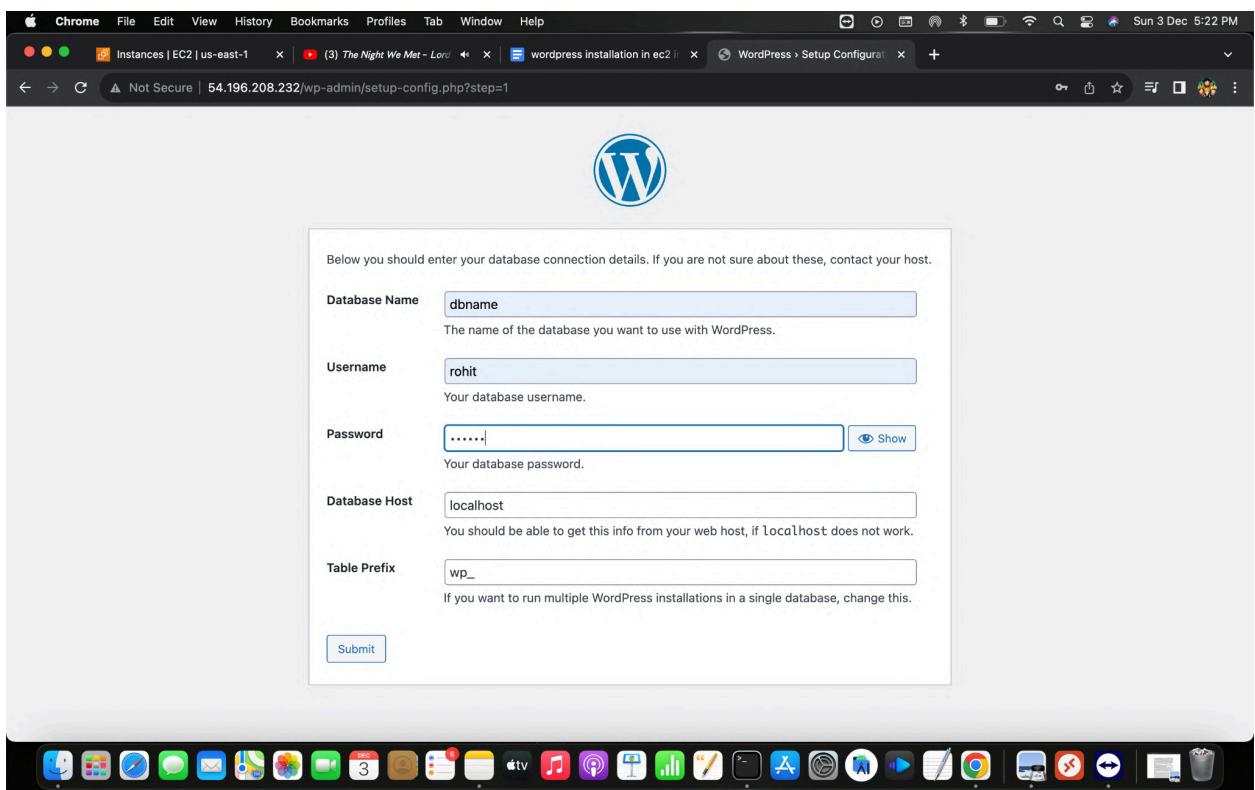
28 - copy the public ip address and paste it to the browser and hit enter. Wordpress page will be open. Select language and click continue.



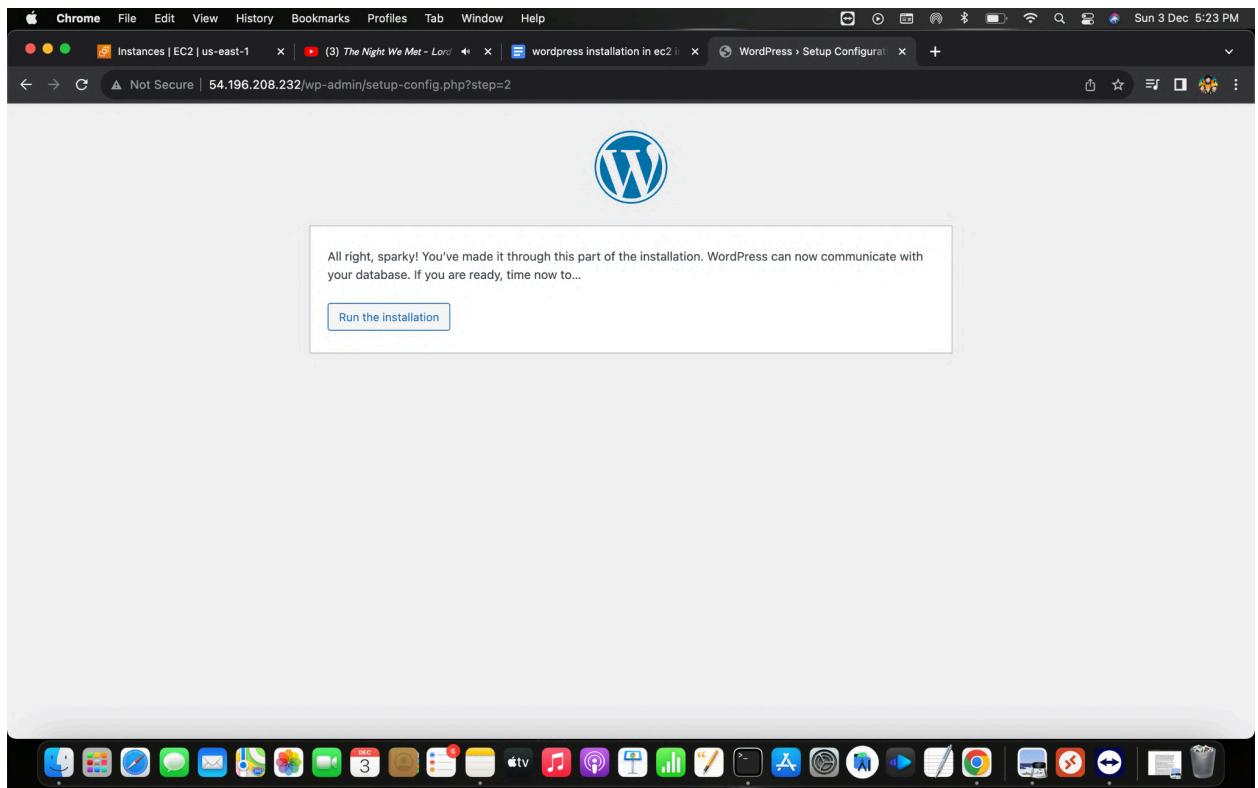
29 - click on let's go.



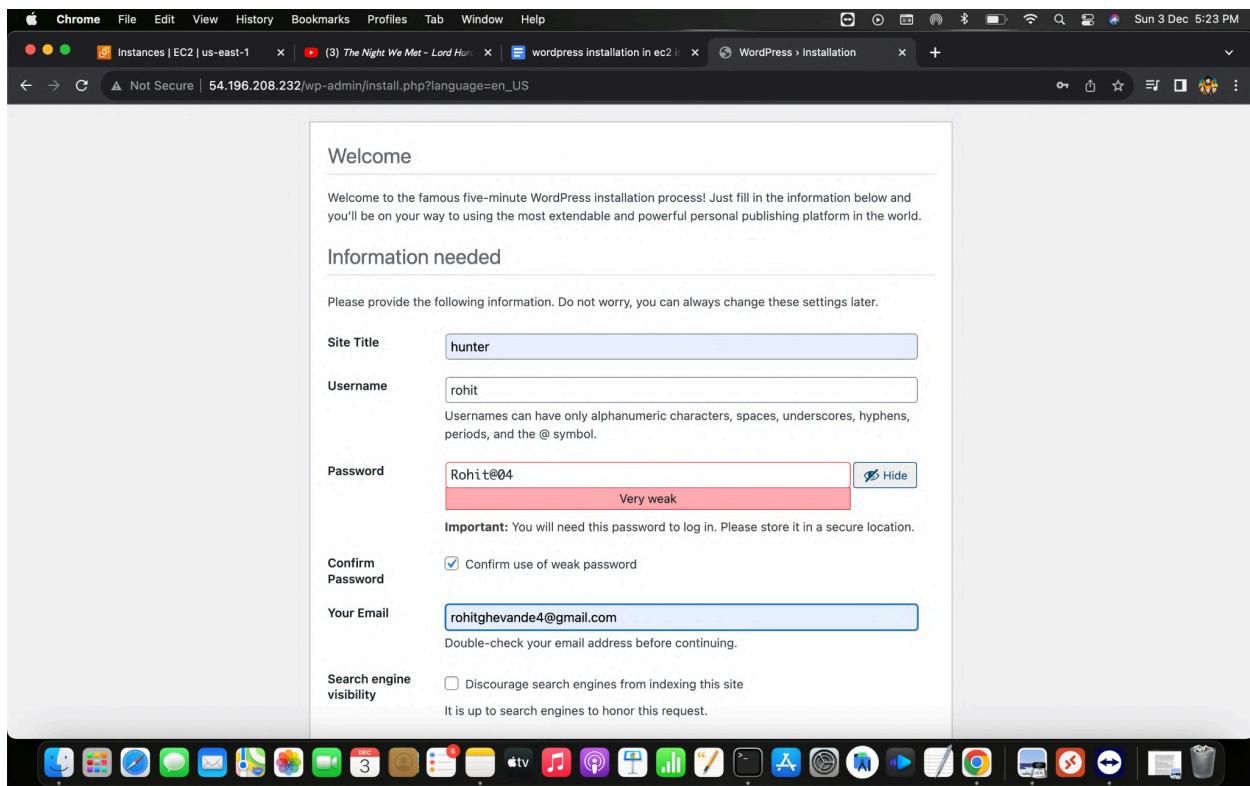
30 - type the database name we created and username and password. Then hit submit.



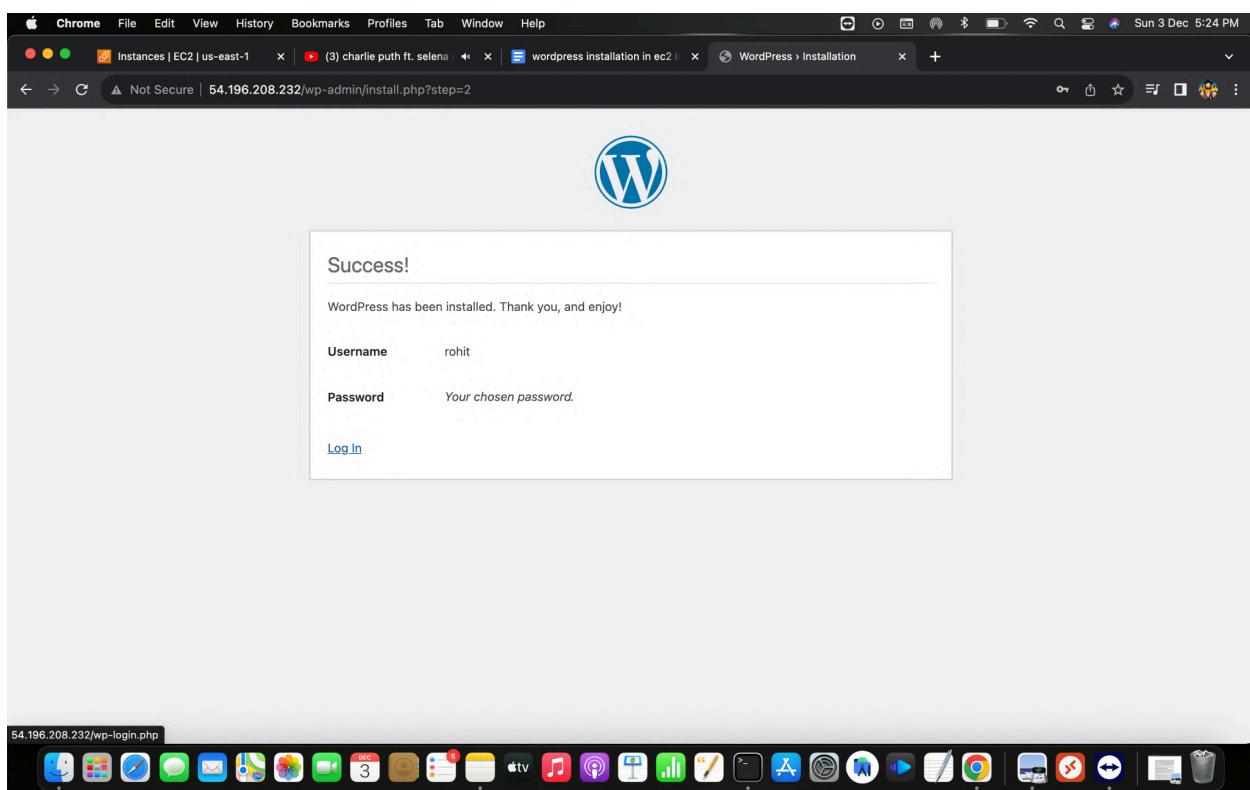
31 - click on **run the installation**.



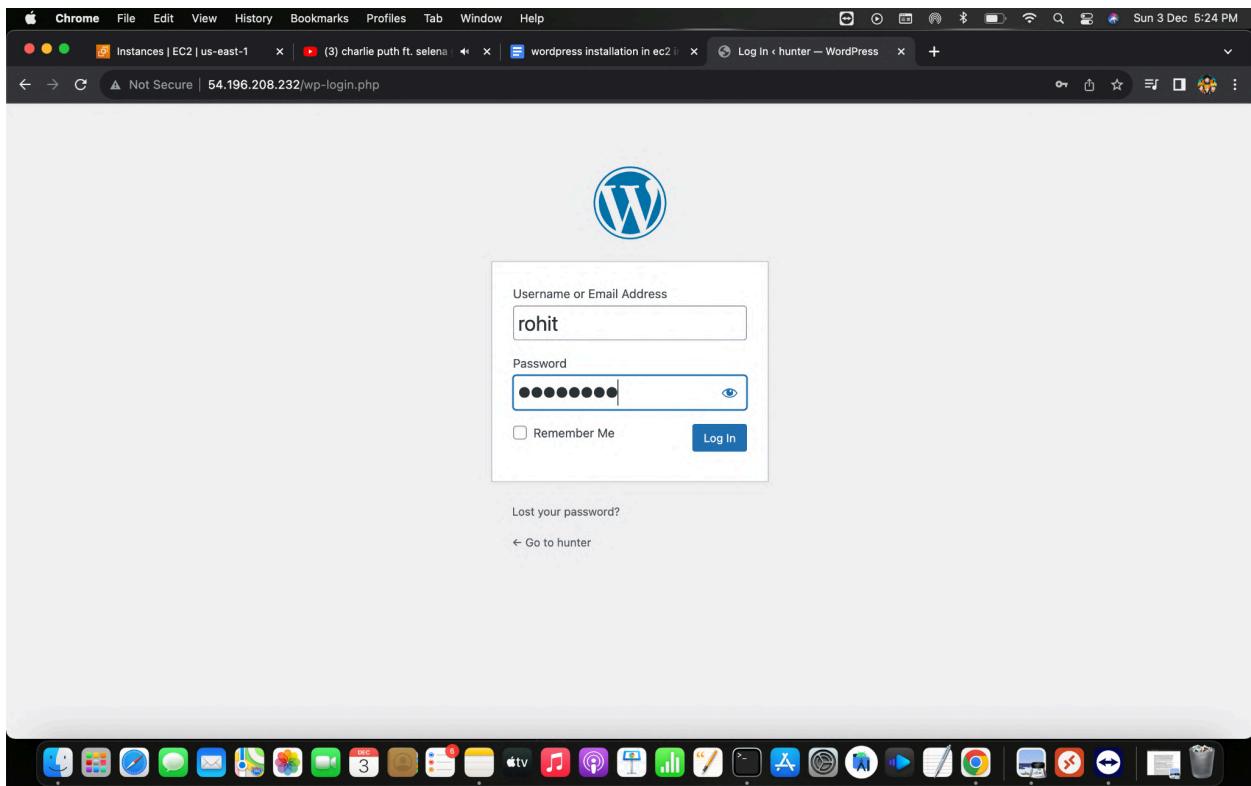
32 - add a title. Add new username, change the default password and type new password, then add gmail id and hit submit.



33 - after creating a profile, click on **log in**.



34 - add the created username and password and click on log in.



35 - welcome to wordpress.

