# **WEB STACK-IMPLEMENTATION (LAMP)**

 The goal of this project is to describe the concepts of Continuous Integration, Continuous Delivery / Deployment and DevOps on a Lamp web stack.

     Never heard about LAMP stack? No - Okay.

LAMP => LINUX, APACHE, MySQL, PHP

All together it's called a stack, just like when you have heard some developers describe themselves as Mern or Mean stack developers.

Mern => mongodb, express, reactjs and nodejs

Mean => mongodb, express, angularjs and nodejs

Pern => postgresqldb, express, react and nodejs

# **Tldr;**

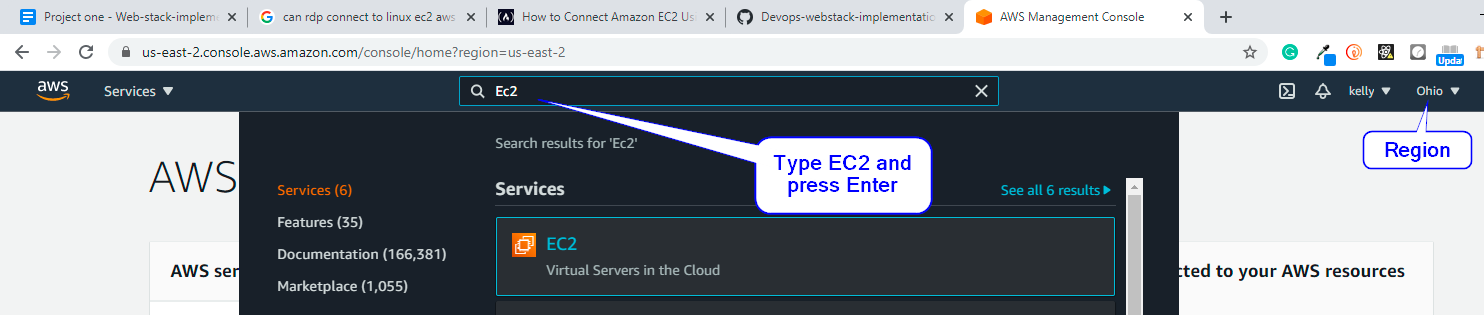
#Video link

# **Prerequisites:**

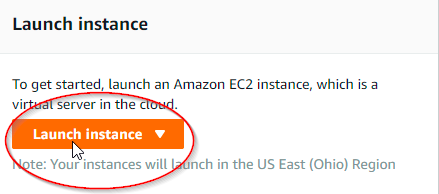
* Aws account running an EC2 instance
* Internet connection
* Fundamental Knowledge of downloading and installing
* Basics Linux skills

**Implementation**

* Open your PC browser and login to <https://aws.amazon.com/>
* A region is selected by default (change if necessary), from the search bar type EC2 and click.

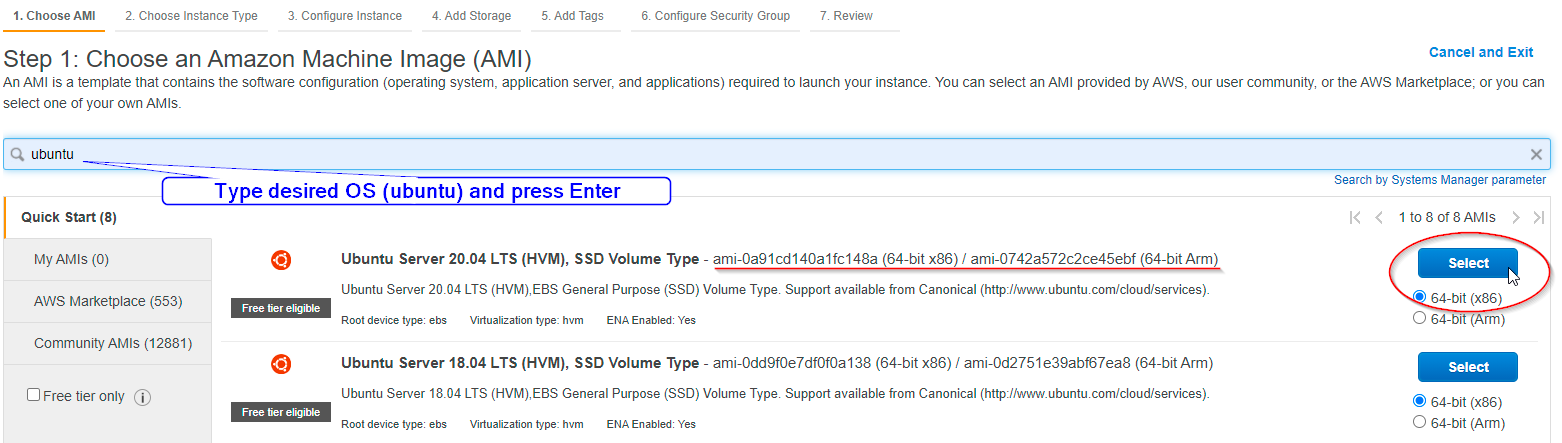


* From the Ec2 dashboard, click on the button “Launch instance” to start using a virtual server.

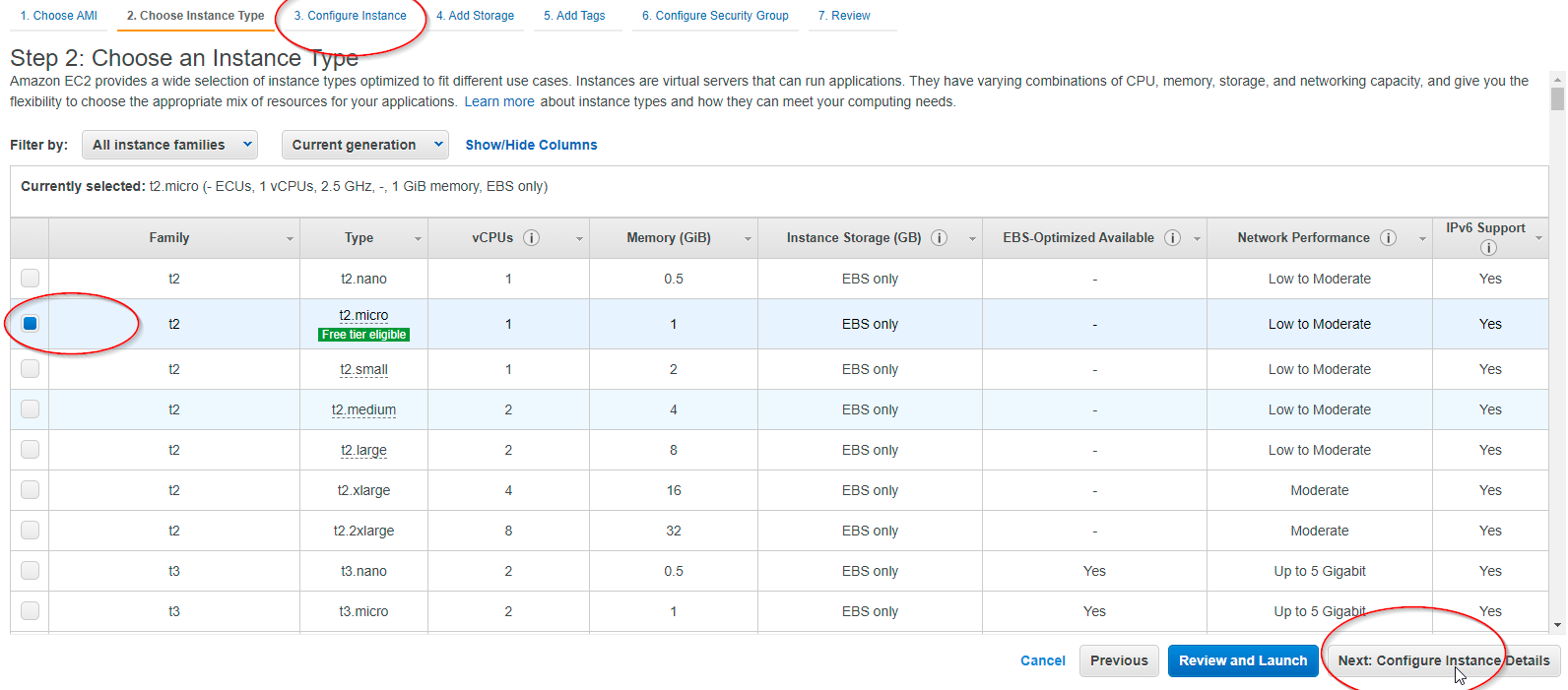


* An AMI window displays, type “Ubuntu” on the search bar and hit enter, or scroll down to select “Ubuntu Server 20.04 LTS (HVM), SSD Volume Type” based on your system architecture.

  Note: the AMI (Amazon machine image) is always different from user to user



* The next step of configuring our EC2 is to select the instance type, preferably a **t2 micro - Free tier**. Then click (3) configure instance showing at the top or click next configuration details at the bottom.



**Move to next step**

* To configure the instance, we will leave all default but scroll to the bottom and on the advanced details section, in the user data column add below script as shown on the screenshot.

**#!/bin/bash**

**apt update -y**

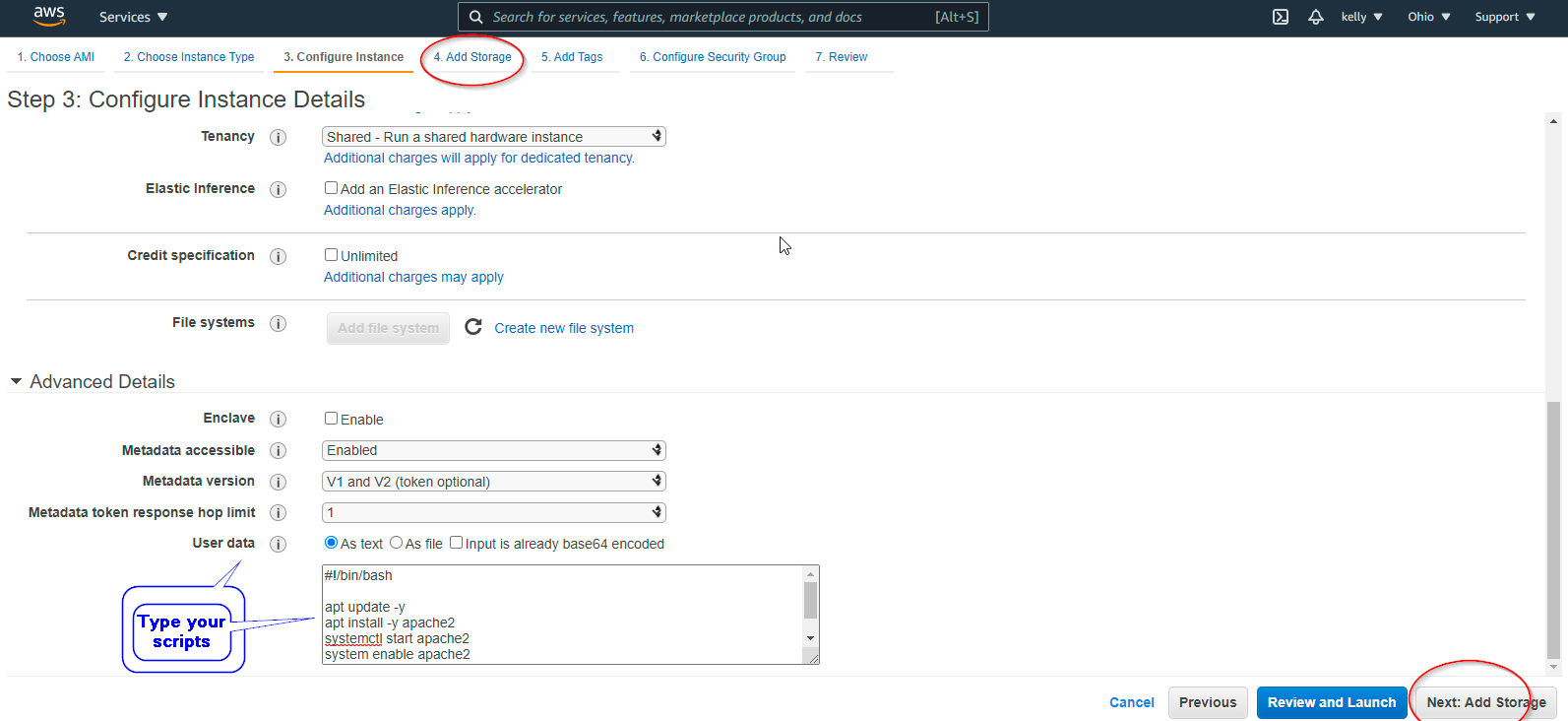
**apt install -y apache2**

**systemctl start apache2**

**system enable apache2**

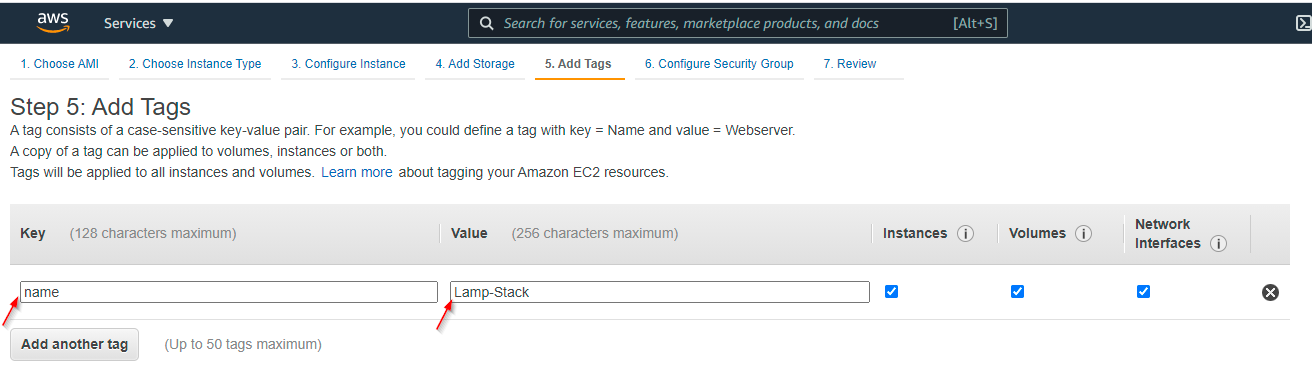
**echo "Hello buddy, salutations to you from $(hostname -f)" > /var/www/html/index.html**

**Move to next step**



* Move to tab 5 to Add tags to our EC2 instance, I have deliberately skipped tab 4 to choose the default storage volume given by AWS.

Tags are key-value paired fields and help to categorize your AWS resource, now click ADD TAG to assign a unique name and move to next.



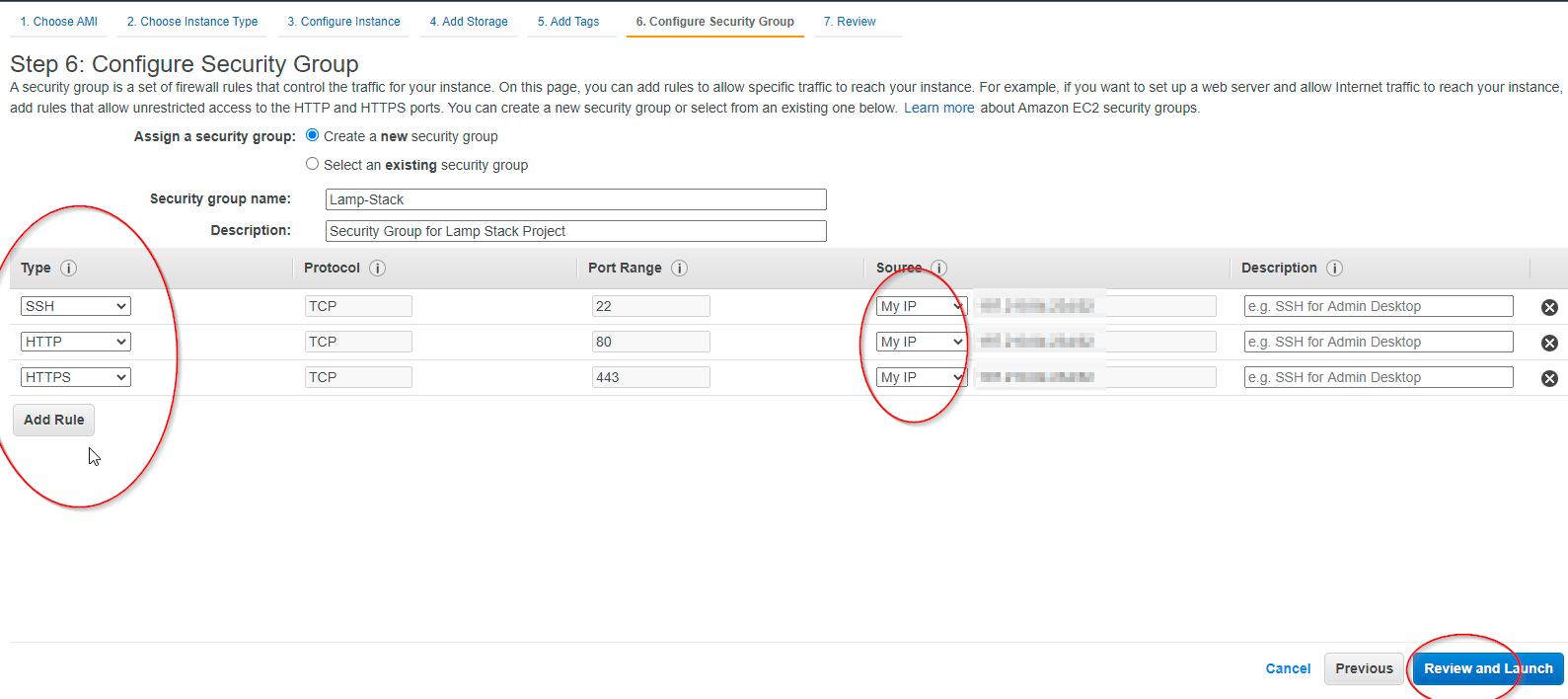
**Move to Next**

* We will modify the default security group by giving access to **port 80, 22 and 443**.

 Reason: The security Group are set of firewall rules which denies and grant access to our EC2 instance,

To access the EC2 instance with a console, we

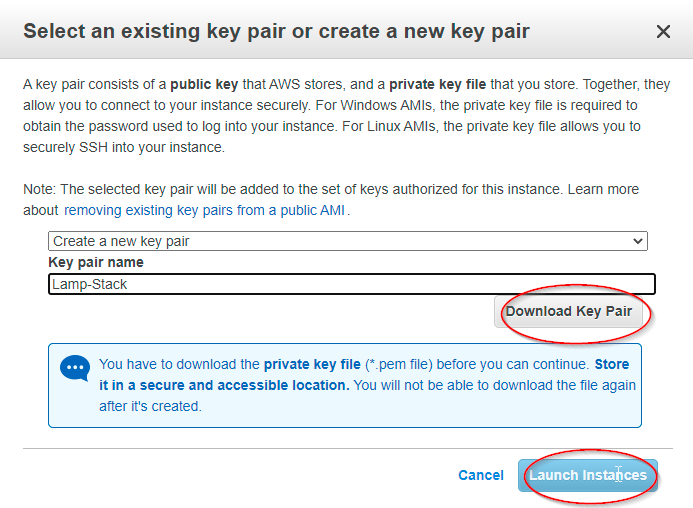
You may add descriptions on the last column.

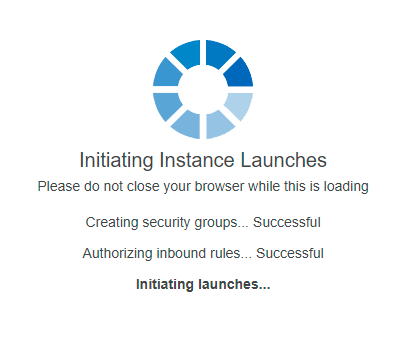


* Click review and launch

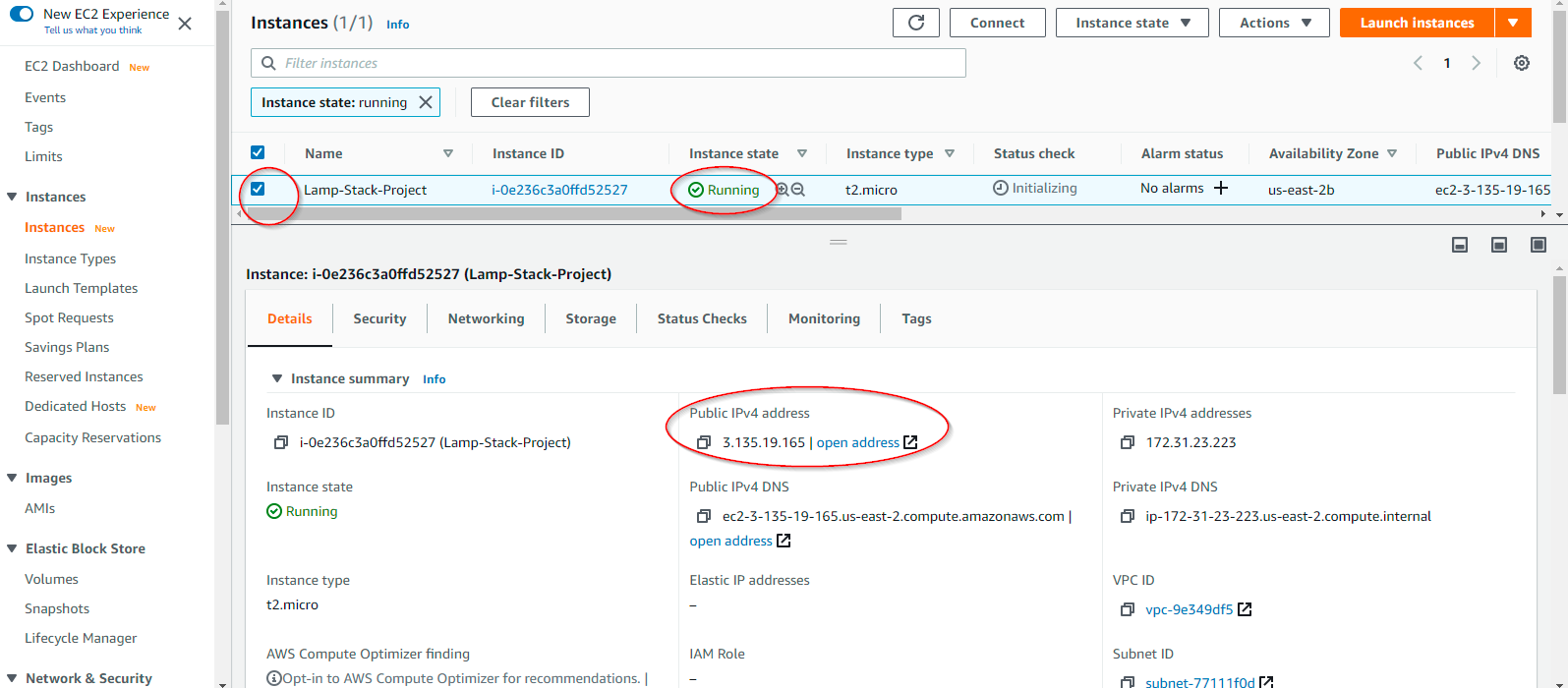
You will get a Prompt to Create a Private Key File, feel free to choose an existing one, if it already exists on the same PC.

**Download the key file to a good location, to be used later, Then Launch.**





Done? Good Job, let’s get to business now.

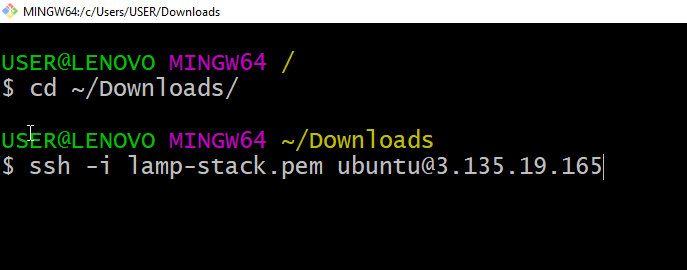


Copy your own Public IP as shown on the above screenshot, now it’s time to use the console

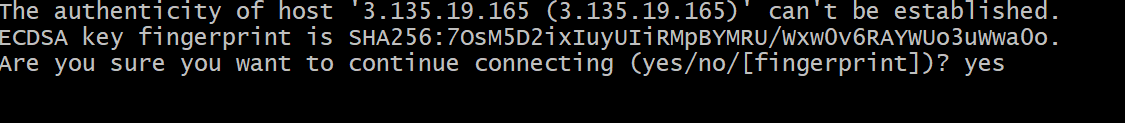
**Yay!!!**

Open git bash or putty or mobaxterm, whichever console is suitable, else download.

We are using git bash here:



**Type YES, to connect.**

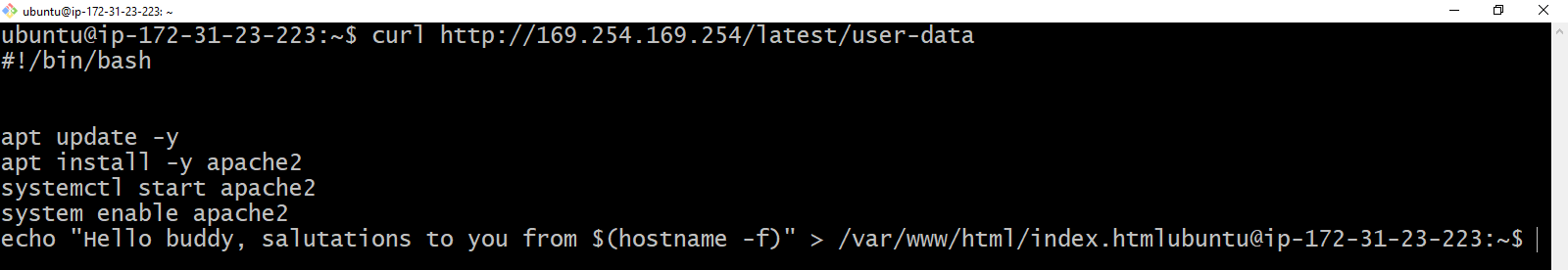


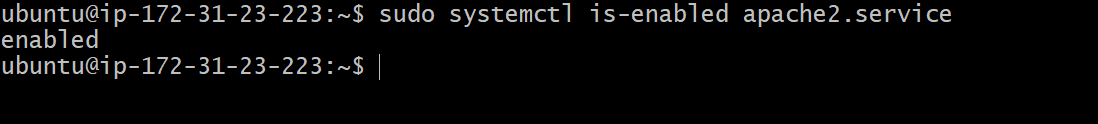
You have now connected to the EC2 instance via SSH

**Type clear**, to have a neat console and proceed.

We will now check if our userdata scripts were loaded.

Type: curl http://169.254.169.254/latest/user-data

By default EC2 user is given sudo privilege

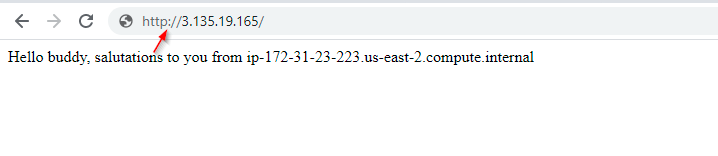


Run this code:

**$ sudo systemctl is-enabled apache2  
or**

**$ sudo systemctl status apache2**

To check if apache has been installed and visit <http://public ip> on browser to confirm.

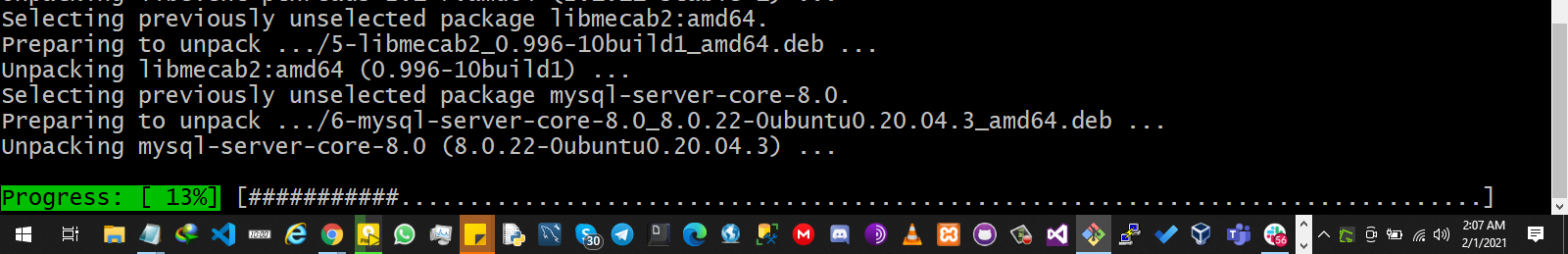


Great!

So from the LAMP stack, we have got Linux and Apache ready, let’s get MySQL running now.

Run this code:

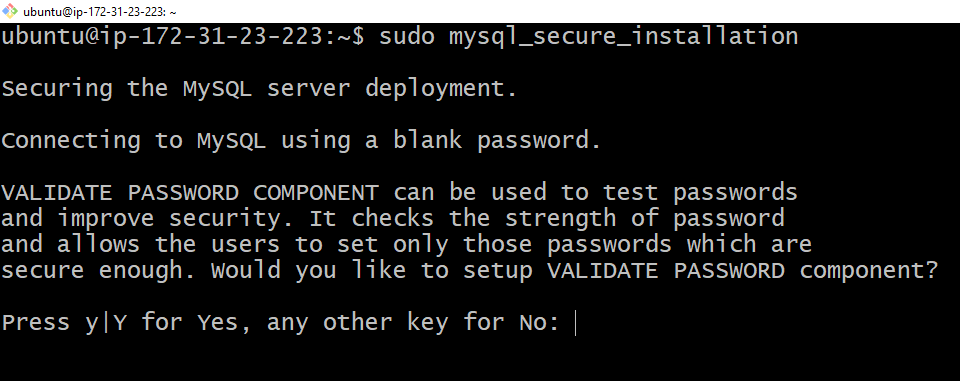
**$ sudo apt install MySQL-server –y**



**Next, we need to configure MySQL to secure authentication.**

Run this code:

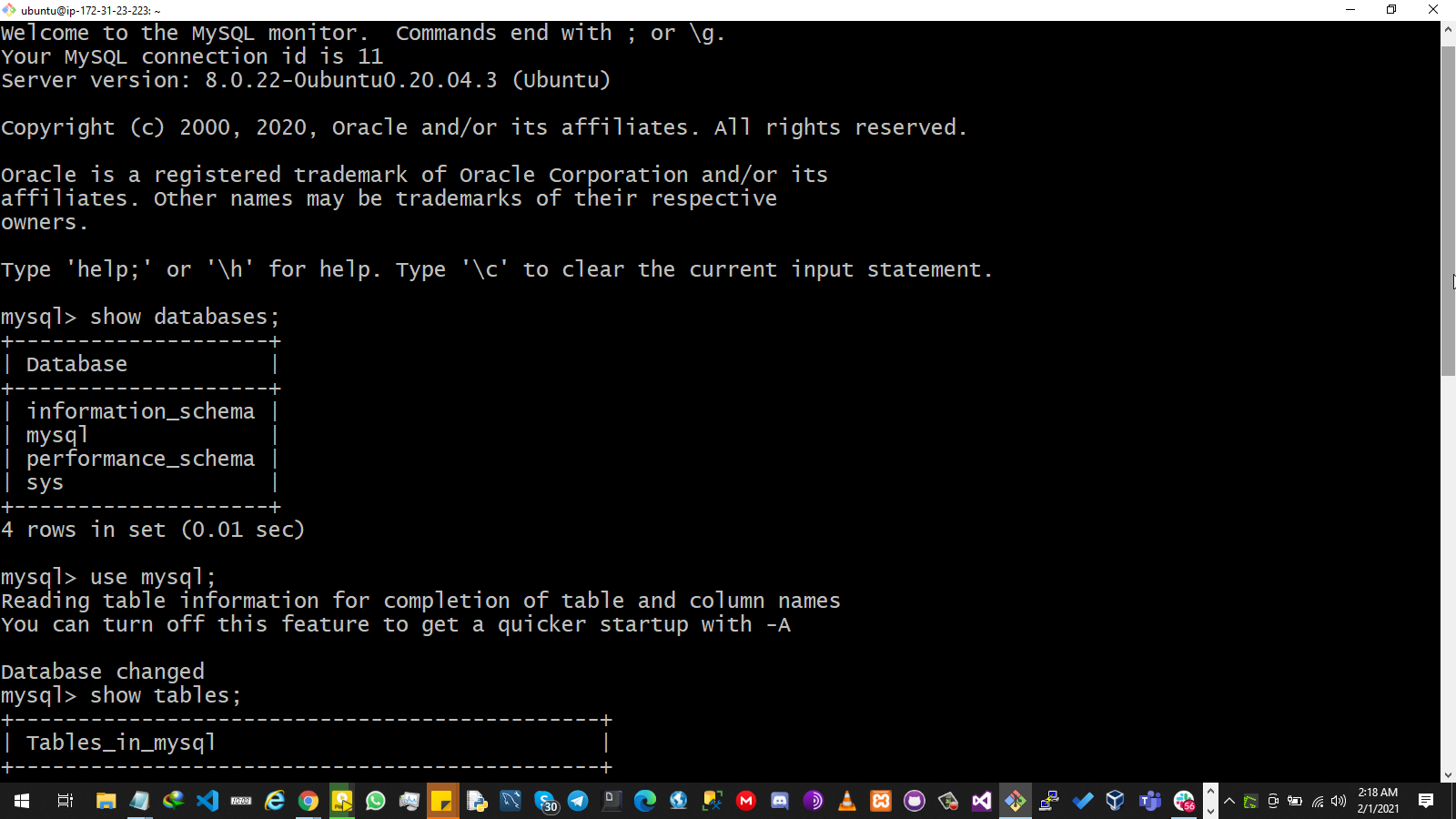
**$ sudo mysql\_secure\_installation**



Type yes to continue, on the next prompt choose 0 or 2 and enter desired password to continue, then type y or yes to continue for ALL prompt.

Great MYSQL is installed and configured for use, we could test by running below code:

**$ sudo MySQL**



Type exit to leave MySQL console editor.

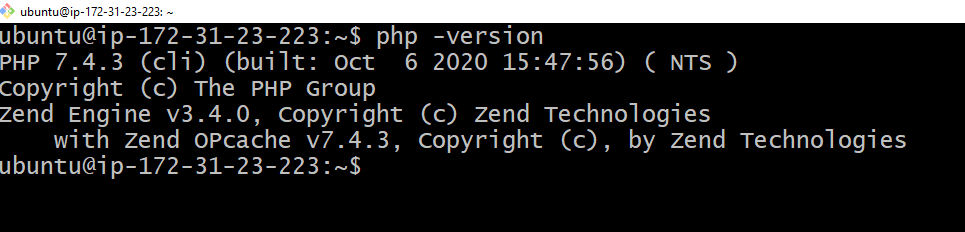
**MySQL> exit**

Linux, Apache and MySQL checked, now let’s install PHP and required dependencies using below command:

$ sudo apt install php libapache2-mod-php php-MySQL -y

Now it’s installed, check PHP version using this command:

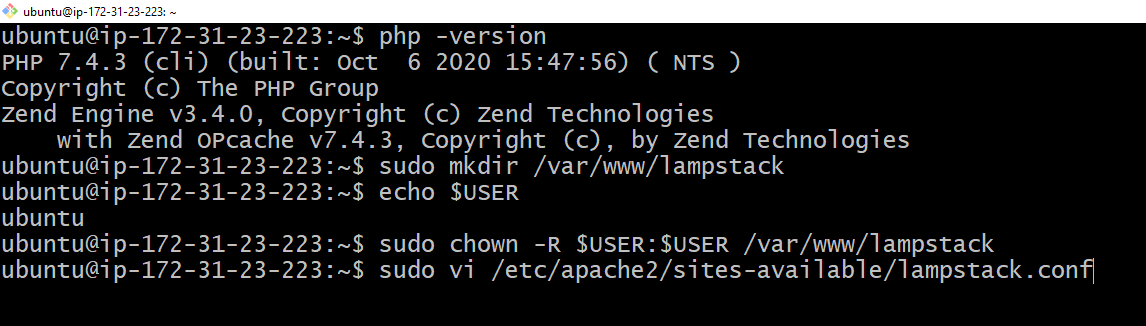
**php -version**



Next step, let’s make a dir. for our site directory, Run below Command

**$ sudo mkdir /var/www/lampstack**

**$ sudo chown -R $USER:$USER /var/www/lampstack**



Run below command, this is done to edit the new site directory:

**$ sudo vi /etc/apache2/sites-available/lampstack.conf**

**Type “I” without the quotes to type the below virtual host file in the config created, then press ESC and exit with “ :wq “ command**

**<VirtualHost \*:80>**

**ServerName lampstack**

**ServerAlias www.lampstack**

**ServerAdmin webmaster@localhost**

**DocumentRoot /var/www/lampstack**

**ErrorLog ${APACHE\_LOG\_DIR}/error.log**

**CustomLog ${APACHE\_LOG\_DIR}/access.log combined**

**</VirtualHost>**

Next check the content of that directory using below command

**$ sudo ls /etc/apache2/sites-available**

Result =>  000-default.conf default-ssl.conf lampstack.conf

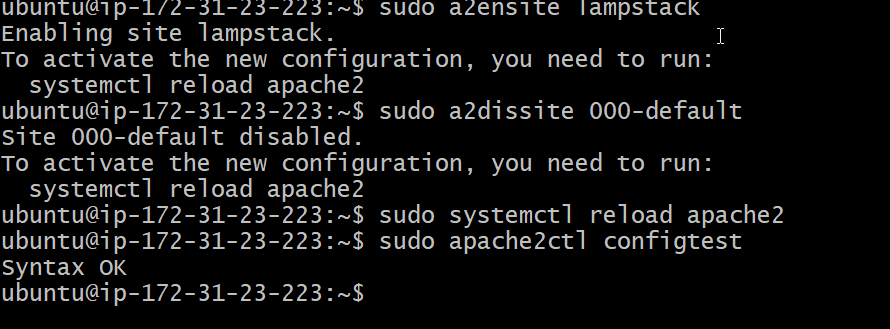
We need to tell apache to enable this new directory to serve our site and disable the default site directory:

**$ sudo a2ensite lampstack**

**$ sudo a2dissite 000-default**

**$ sudo apache2ctl configtest**

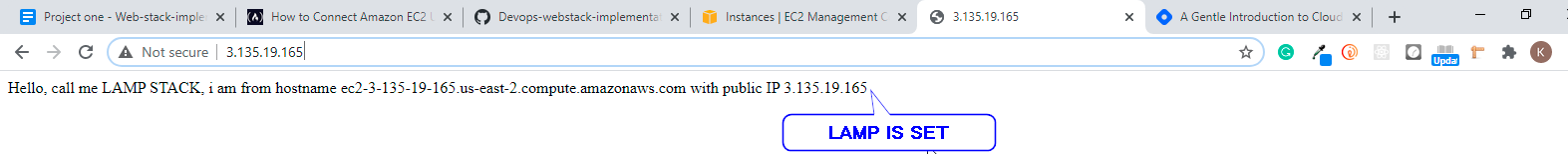
**$ sudo systemctl reload apache2**



Good, finally, let’s add an index file content to our new site directory

**$ sudo echo 'Hello, call me LAMP STACK, I am from hostname' $(curl -s** [**http://169.254.169.254/latest/meta-data/public-hostname**](http://169.254.169.254/latest/meta-data/public-hostname)**) 'with public IP' $(curl -s** [**http://169.254.169.254/latest/meta-data/public-ipv4**](http://169.254.169.254/latest/meta-data/public-ipv4)**) > /var/www/lampstack/index.html**

Refresh browser and check now, Hola!!!

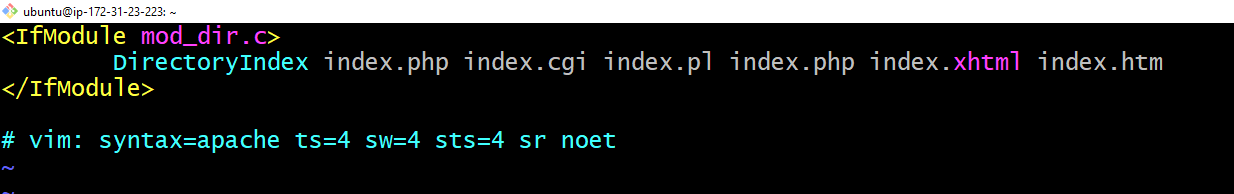


**http://<public ip>:80 is set.**

 To cap it all up, if you need to serve php files, then we need to tweak a file, and make index.php the first directory index as shown below.

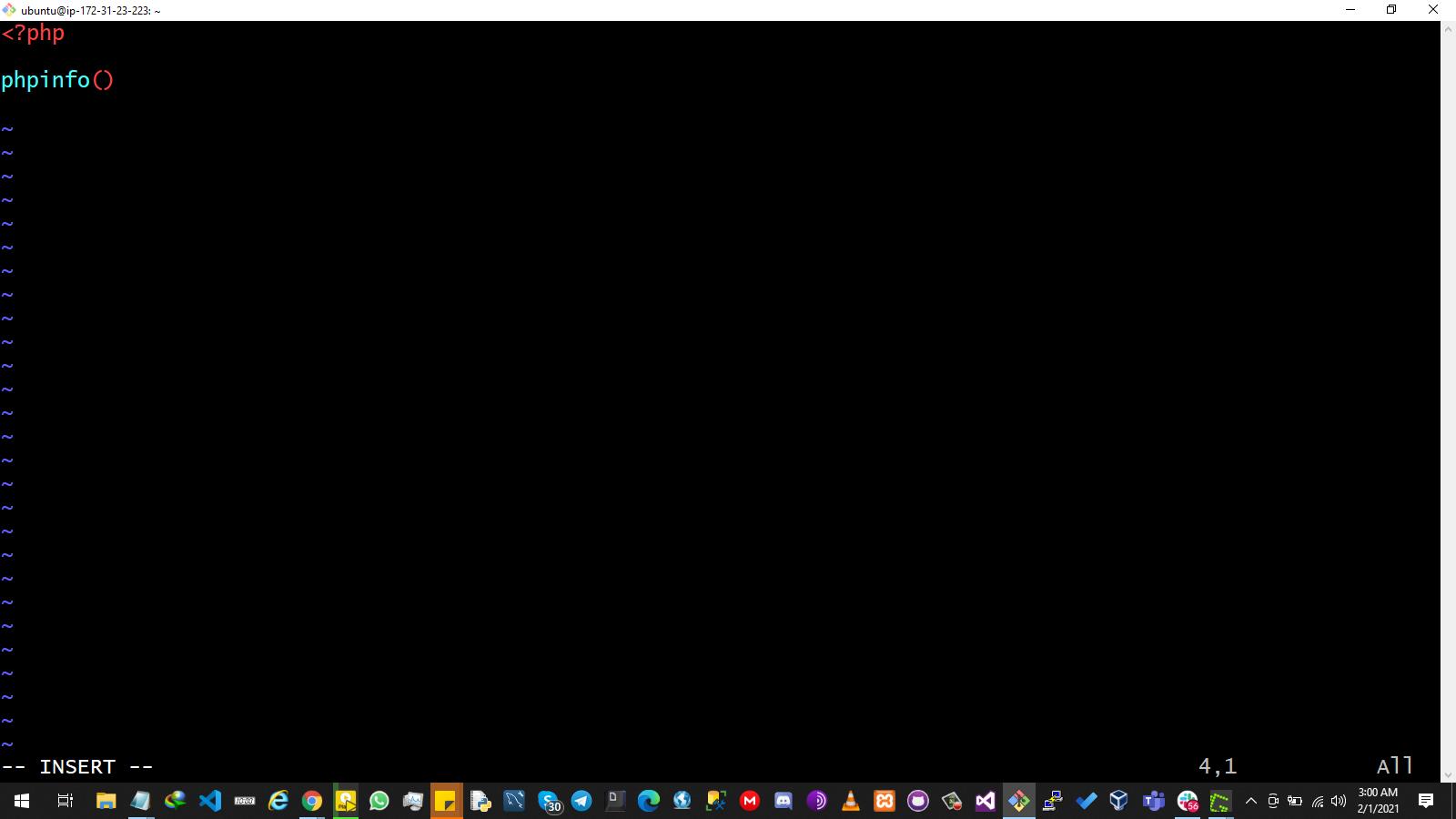
**Run this command**

**$ sudo vi /etc/apache2/mods-enabled/dir.conf**



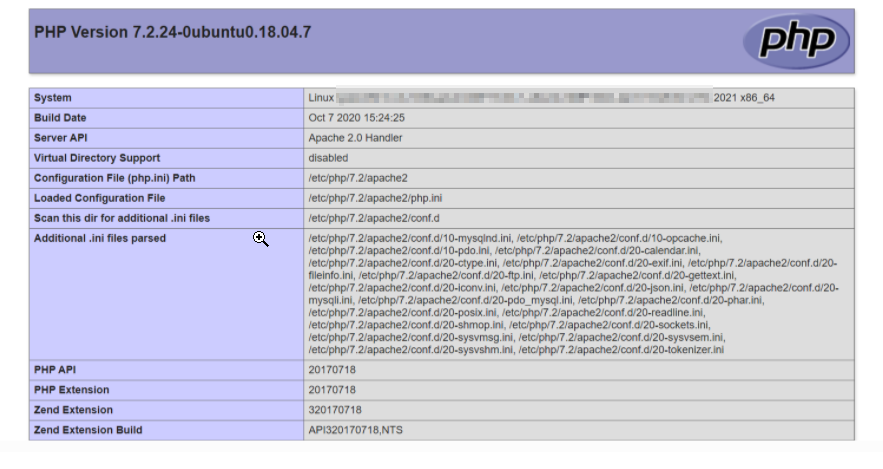
Then, let’s edit the index.php file to add contents, Run below command

**$ vi /var/www/lampstack/index.php**



Then reload apache to effect change at reboot.

**$ sudo systemctl reload apache2**

Then refresh and check your browser.

Thank you, this is the minimum requirement to set up an AWS instance with Linux, Apache, MySQL and PHP for a web project.

Hope this was informative.

PS: Remember to terminate your EC2 instance.

