# WEB STACK-IMPLEMENTATION (LAMP)

The goal of this project is to describe the concepts and process of Continuous Integration, Continuous Delivery / Deployment and DevOps practices using 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

Lemp => Linux, nginx, mysql and php

Mean => Mongodb, express, angularjs and nodejs

Pern => Postgresqldb, express, react and nodejs

## Tldr;

#Video link

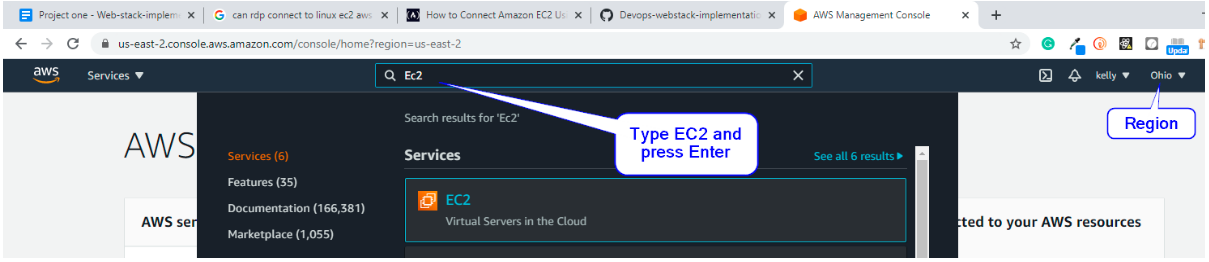
## Prerequisites:

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

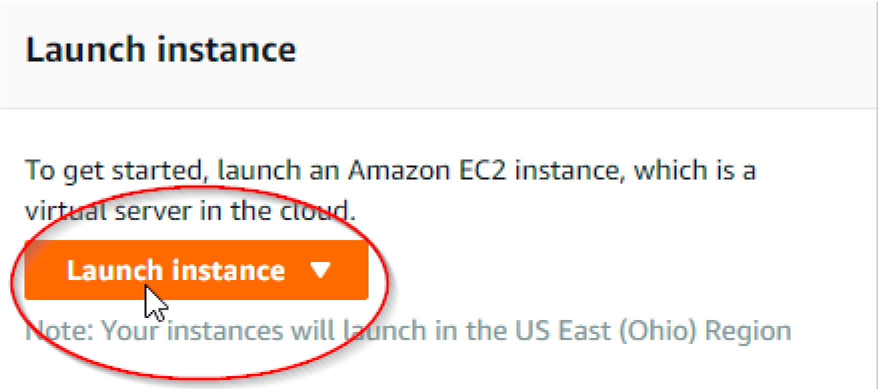
# Implementation

- 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.


Project one - Web-stack-implem
can rdp connect to linux ec2 aws
How to Connect Amazon EC2 Us
Devops-webstack-implementatio
AWS Management Console
us-east-2.console.aws.amazon.com/console/home?region=us-east-2
aws
Services ▼
Ec2
kelly ▼
Ohio ▼
Region
Search results for 'Ec2'
Type EC2 and press Enter
AWS
Services (6)
Features (35)
Documentation (166,381)
Marketplace (1,055)
Services
See all 6 results ▸
EC2
Virtual Servers in the Cloud
AWS ser
ted to your AWS resources

- From the Ec2 dashboard, click on the button "Launch instance" to start using a virtual server.



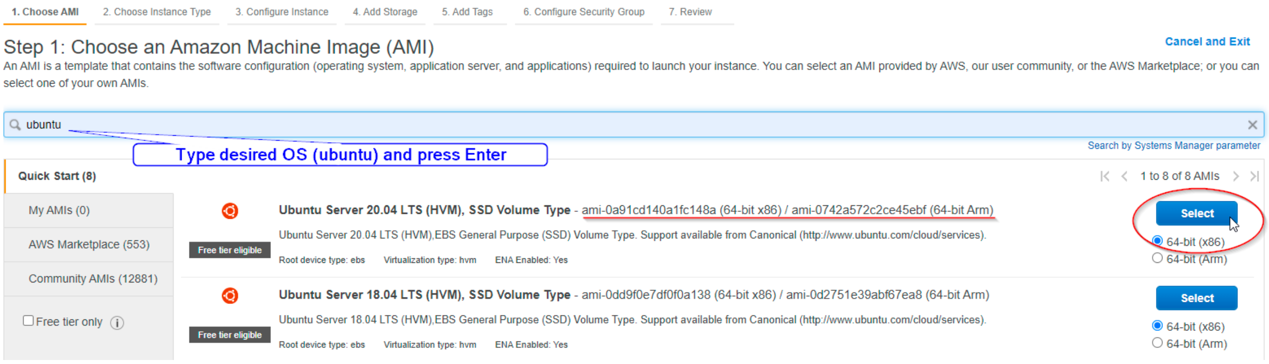
Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the US East (Ohio) Region

- 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



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

ubuntu

Type desired OS (ubuntu) and press Enter

**Ubuntu Server 20.04 LTS (HVM), SSD Volume Type** - ami-0a91cd140a1fc148a (64-bit x86) / ami-0742a572c2ce45ebf (64-bit Arm)

Free tier eligible

Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select ● 64-bit (x86) ○ 64-bit (Arm)

**Ubuntu Server 18.04 LTS (HVM), SSD Volume Type** - ami-0dd9f0e7df0f0a138 (64-bit x86) / ami-0d2751e39abf67ea8 (64-bit Arm)

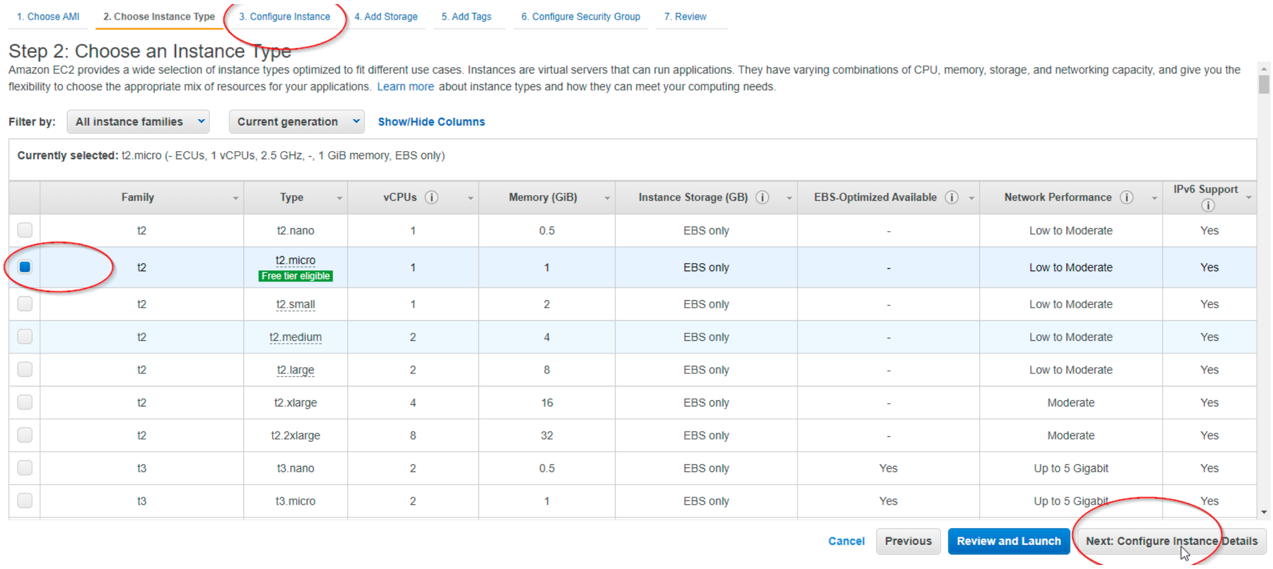
Free tier eligible

Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select ● 64-bit (x86) ○ 64-bit (Arm)

- 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.



Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance families · Current generation · Show/Hide Columns

**Currently selected:** t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

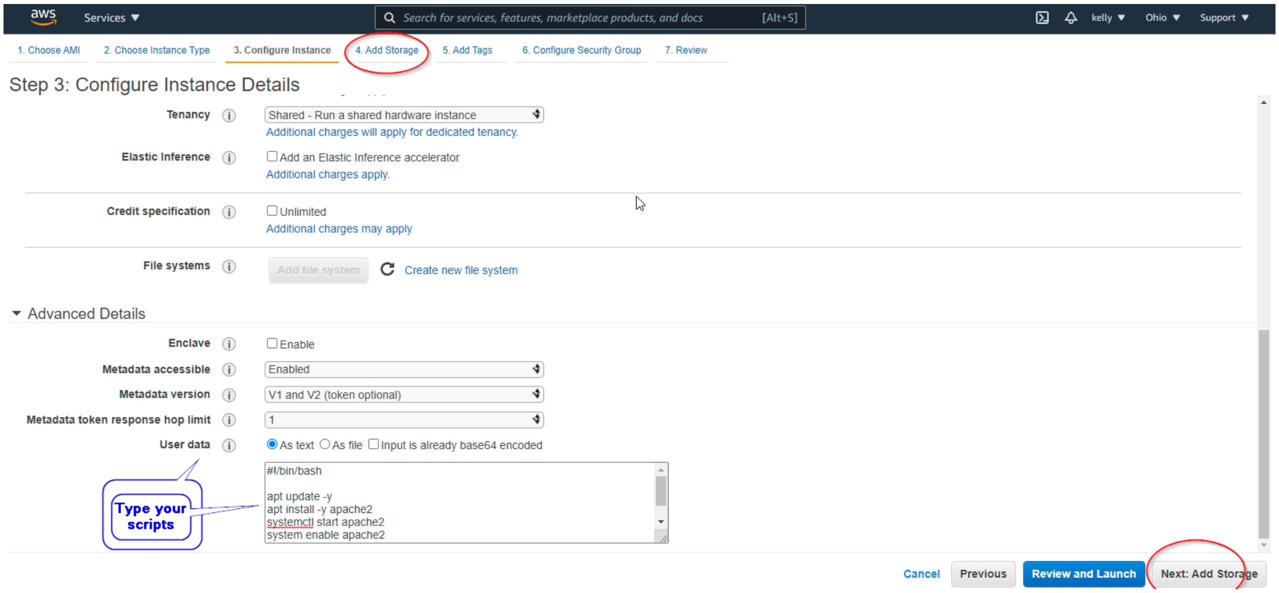
| | Family | Type | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance | IPv6 Support |
|---|---|---|---|---|---|---|---|---|
| □ | t2 | t2.nano | 1 | 0.5 | EBS only | - | Low to Moderate | Yes |
| ☑ | t2 | t2.micro<br>Free tier eligible | 1 | 1 | EBS only | - | Low to Moderate | Yes |
| □ | t2 | t2.small | 1 | 2 | EBS only | - | Low to Moderate | Yes |
| □ | t2 | t2.medium | 2 | 4 | EBS only | - | Low to Moderate | Yes |
| □ | t2 | t2.large | 2 | 8 | EBS only | - | Low to Moderate | Yes |
| □ | t2 | t2.xlarge | 4 | 16 | EBS only | - | Moderate | Yes |
| □ | t2 | t2.2xlarge | 8 | 32 | EBS only | - | Moderate | Yes |
| □ | t3 | t3.nano | 2 | 0.5 | EBS only | Yes | Up to 5 Gigabit | Yes |
| □ | t3 | t3.micro | 2 | 1 | EBS only | Yes | Up to 5 Gigabit | Yes |

**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



Step 3: Configure Instance Details

Tenancy: Shared - Run a shared hardware instance
Additional charges will apply for dedicated tenancy.

Elastic Inference: □ Add an Elastic Inference accelerator
Additional charges apply.

Credit specification: □ Unlimited
Additional charges may apply

File systems: Add file system | Create new file system

Advanced Details

Enclave: □ Enable

Metadata accessible: Enabled

Metadata version: V1 and V2 (token optional)

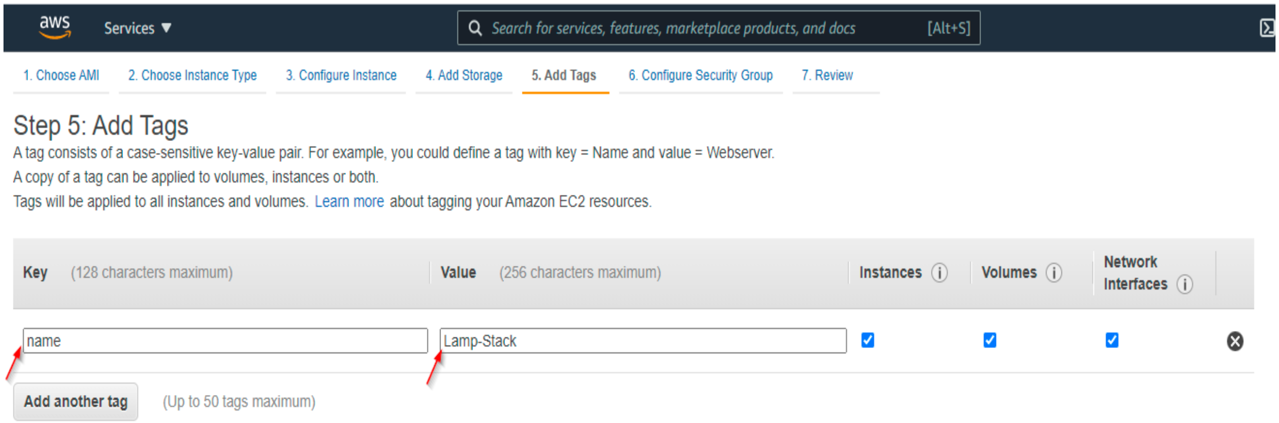
Metadata token response hop limit: 1

User data: ● As text ○ As file □ Input is already base64 encoded

```
#!/bin/bash

apt update -y
apt install -y apache2
systemctl start apache2
system enable apache2
```

- 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.



aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

| Key (128 characters maximum) | Value (256 characters maximum) | Instances | Volumes | Network Interfaces | |
|---|---|---|---|---|---|
| name | Lamp-Stack | ☑ | ☑ | ☑ | ⊗ |

Add another tag (Up to 50 tags maximum)

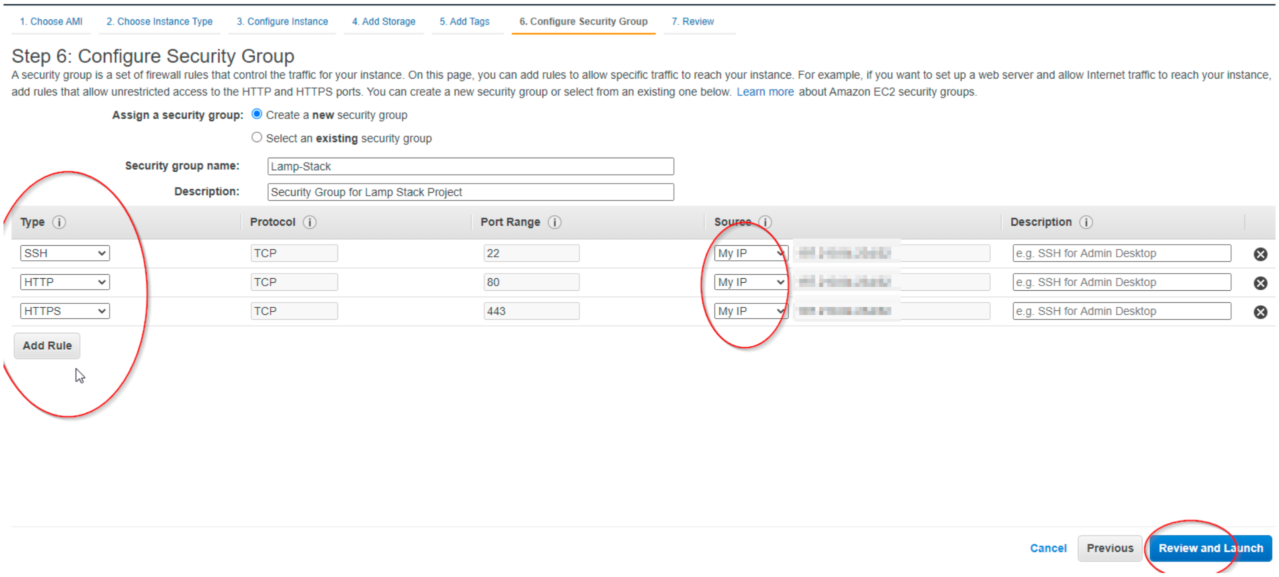
**Move to Next**

- 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.



1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: Lamp-Stack

Description: Security Group for Lamp Stack Project

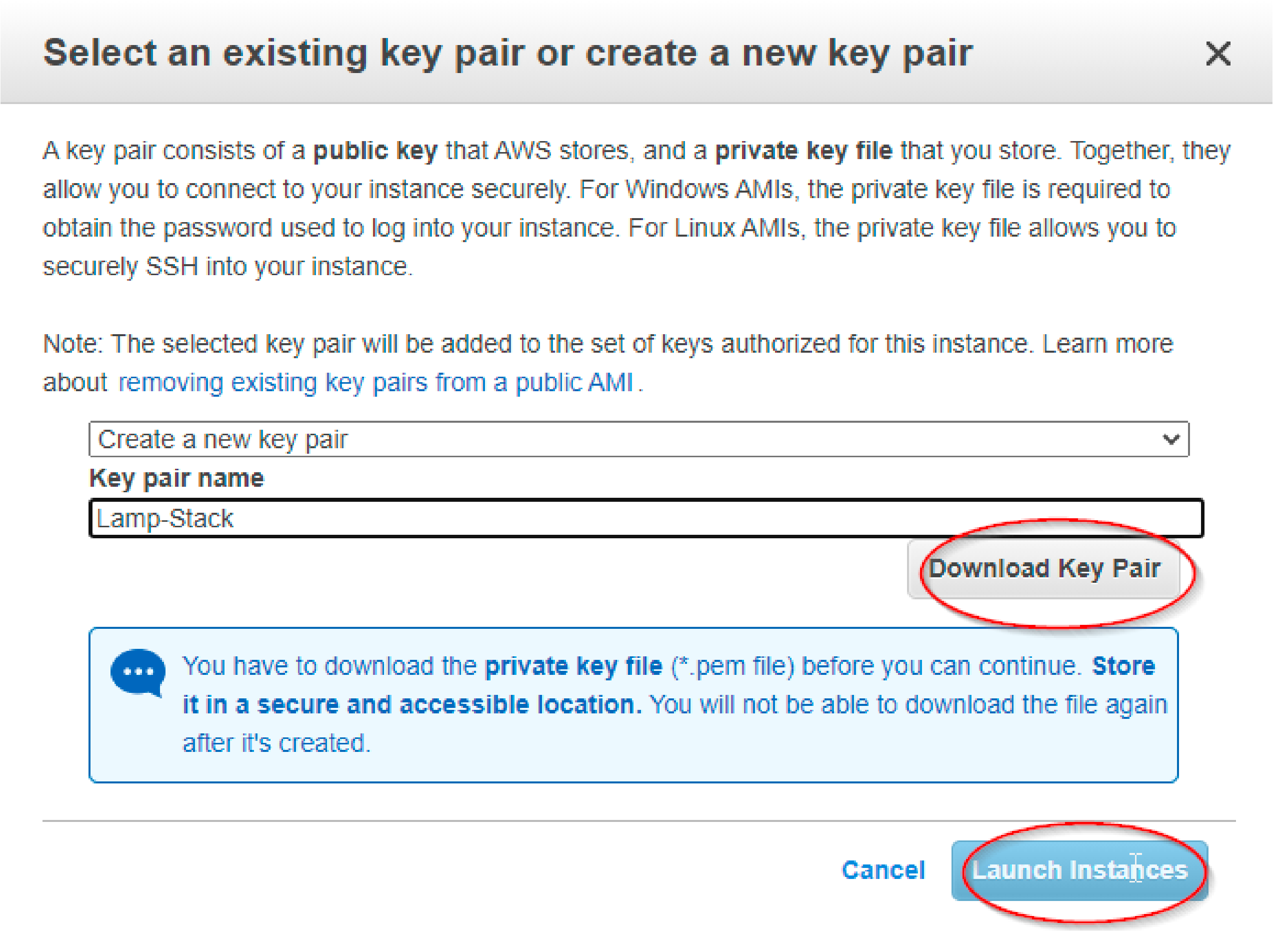
| Type | Protocol | Port Range | Source | Description | |
|---|---|---|---|---|---|
| SSH | TCP | 22 | My IP [redacted] | e.g. SSH for Admin Desktop | ⊗ |
| HTTP | TCP | 80 | My IP [redacted] | e.g. SSH for Admin Desktop | ⊗ |
| HTTPS | TCP | 443 | My IP [redacted] | e.g. SSH for Admin Desktop | ⊗ |

Add Rule

- 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.**



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

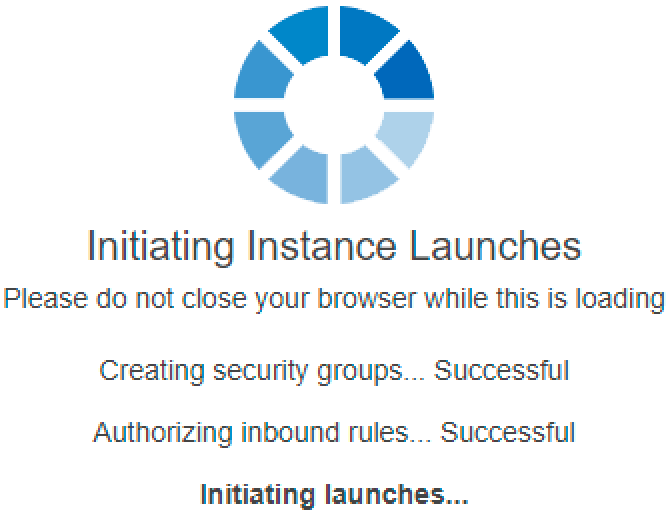
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI .

Create a new key pair

**Key pair name**

Lamp-Stack

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.



Initiating Instance Launches

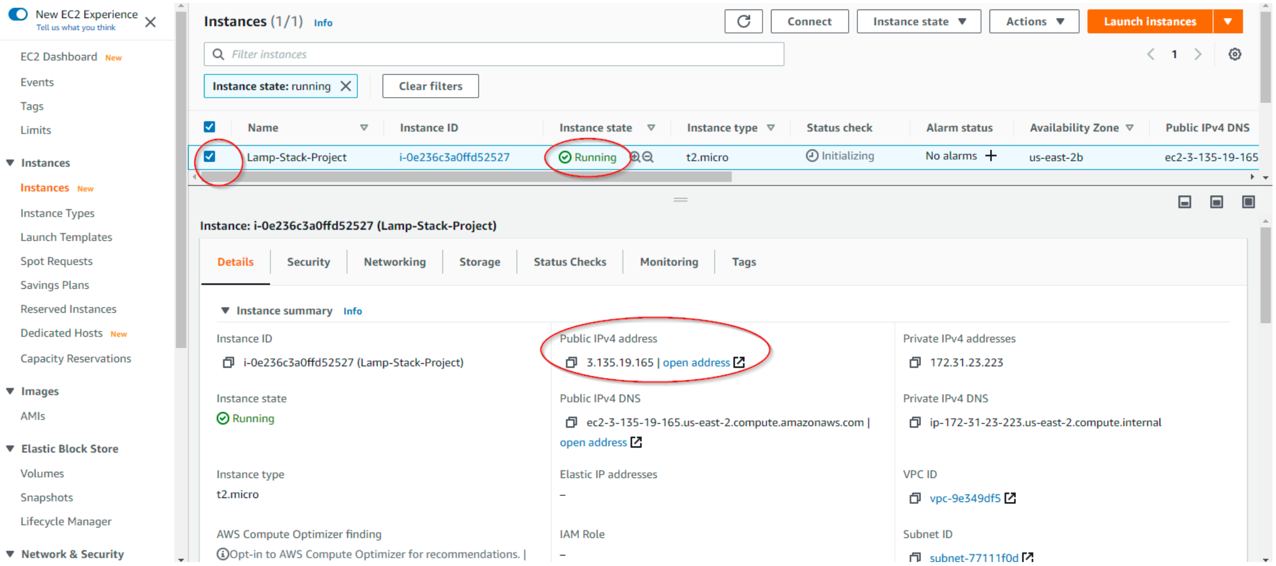
Please do not close your browser while this is loading

Creating security groups... Successful

Authorizing inbound rules... Successful

**Initiating launches...**

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



Instances (1/1) Info

Instance state: running

| ☑ | Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS |
|---|---|---|---|---|---|---|---|---|
| ☑ | Lamp-Stack-Project | i-0e236c3a0ffd52527 | Running | t2.micro | Initializing | No alarms + | us-east-2b | ec2-3-135-19-165 |

Instance: i-0e236c3a0ffd52527 (Lamp-Stack-Project)

▼ Instance summary Info

Instance ID
i-0e236c3a0ffd52527 (Lamp-Stack-Project)

Public IPv4 address
3.135.19.165 | open address

Private IPv4 addresses
172.31.23.223

Instance state
Running

Public IPv4 DNS
ec2-3-135-19-165.us-east-2.compute.amazonaws.com | open address

Private IPv4 DNS
ip-172-31-23-223.us-east-2.compute.internal

Instance type
t2.micro

Elastic IP addresses
–

VPC ID
vpc-9e349df5

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. |

IAM Role
–

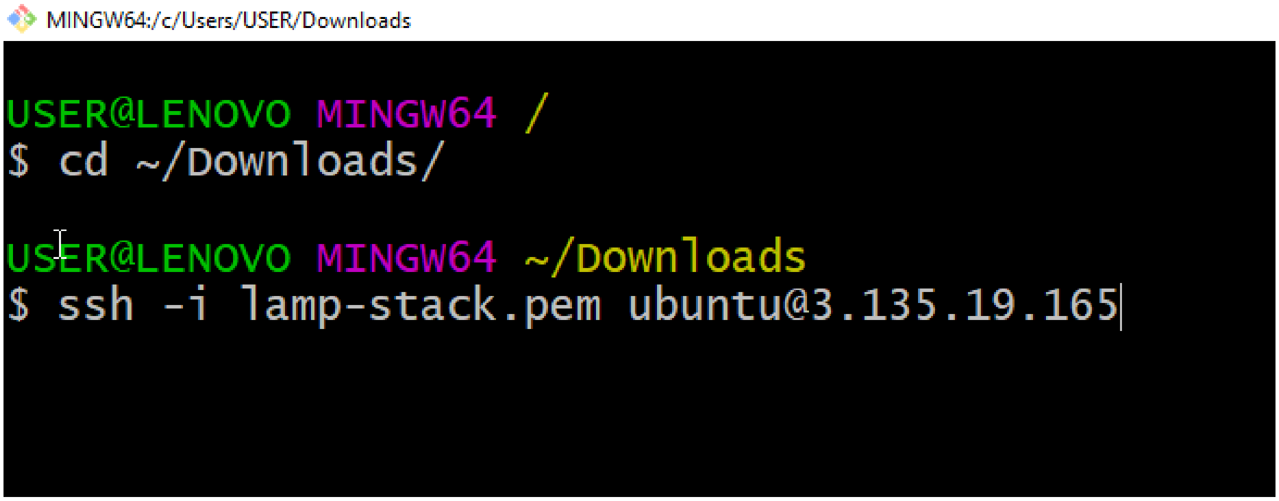
Subnet ID
subnet-77111f0d

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

**Yay!!!**

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

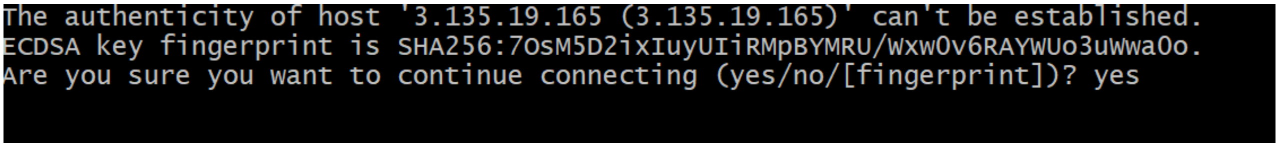
We are using git bash here:



MINGW64:/c/Users/USER/Downloads

```
USER@LENOVO MINGW64 /
$ cd ~/Downloads/

USER@LENOVO MINGW64 ~/Downloads
$ ssh -i lamp-stack.pem ubuntu@3.135.19.165
```

**Type YES, to connect.**



```
The authenticity of host '3.135.19.165 (3.135.19.165)' can't be established.
ECDSA key fingerprint is SHA256:7OsM5D2ixIuyUIiRMpBYMRU/Wxw0v6RAYWUo3uWwa0o.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

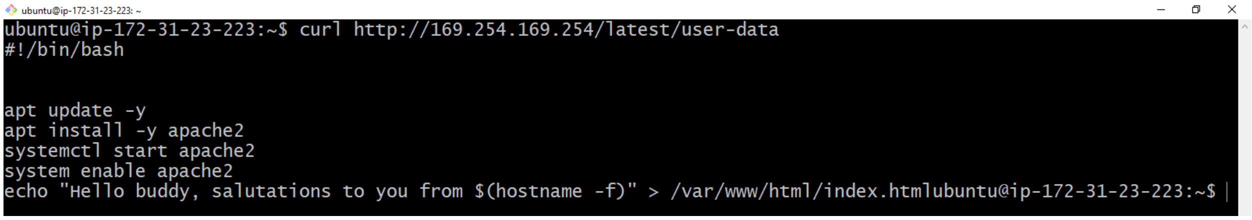
You have now connected to the EC2 instance via SSH

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

We will now check if our userdata scripts were loaded.

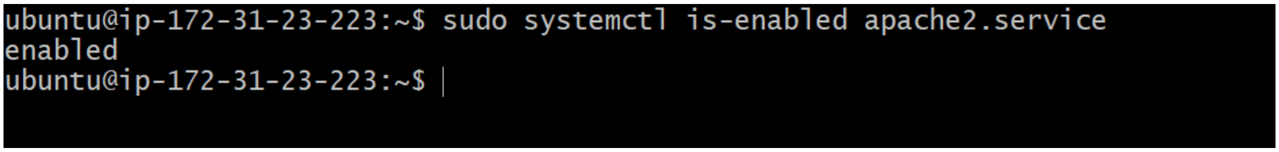
Type: curl http://169.254.169.254/latest/user-data

By default EC2 user is given sudo privilege



ubuntu@ip-172-31-23-223: ~

```
ubuntu@ip-172-31-23-223:~$ curl http://169.254.169.254/latest/user-data
#!/bin/bash


apt update -y
apt install -y apache2
systemctl start apache2
system enable apache2
echo "Hello buddy, salutations to you from $(hostname -f)" > /var/www/html/index.htmlubuntu@ip-172-31-23-223:~$
```

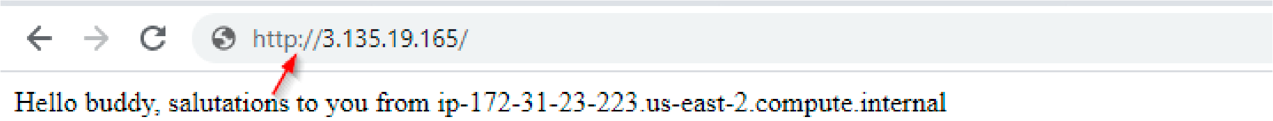


```
ubuntu@ip-172-31-23-223:~$ sudo systemctl is-enabled apache2.service
enabled
ubuntu@ip-172-31-23-223:~$ 
```

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.



http://3.135.19.165/

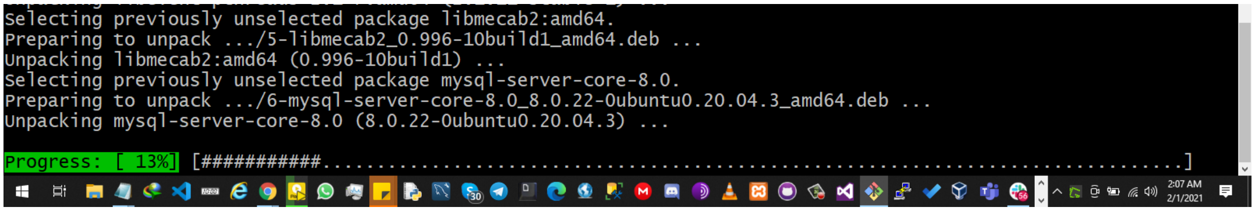
Hello buddy, salutations to you from ip-172-31-23-223.us-east-2.compute.internal

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
```

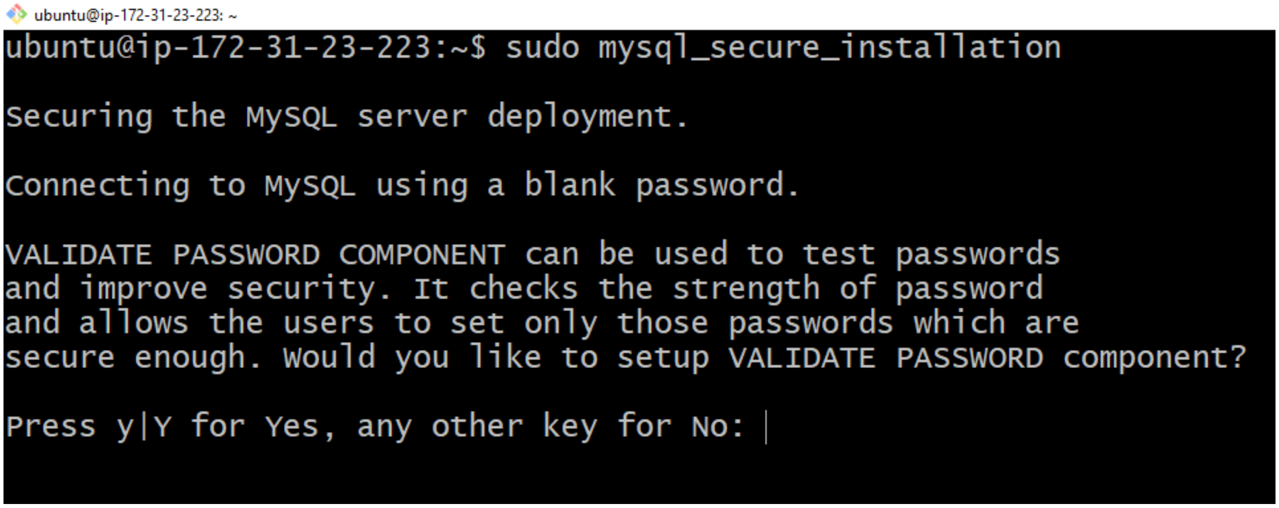


```
Selecting previously unselected package libmecab2:amd64.
Preparing to unpack .../5-libmecab2_0.996-10build1_amd64.deb ...
Unpacking libmecab2:amd64 (0.996-10build1) ...
Selecting previously unselected package mysql-server-core-8.0.
Preparing to unpack .../6-mysql-server-core-8.0_8.0.22-0ubuntu0.20.04.3_amd64.deb ...
Unpacking mysql-server-core-8.0 (8.0.22-0ubuntu0.20.04.3) ...

Progress: [ 13%] [###########...........................................................................]
```

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

Run this code:

```
$ sudo mysql_secure_installation
```

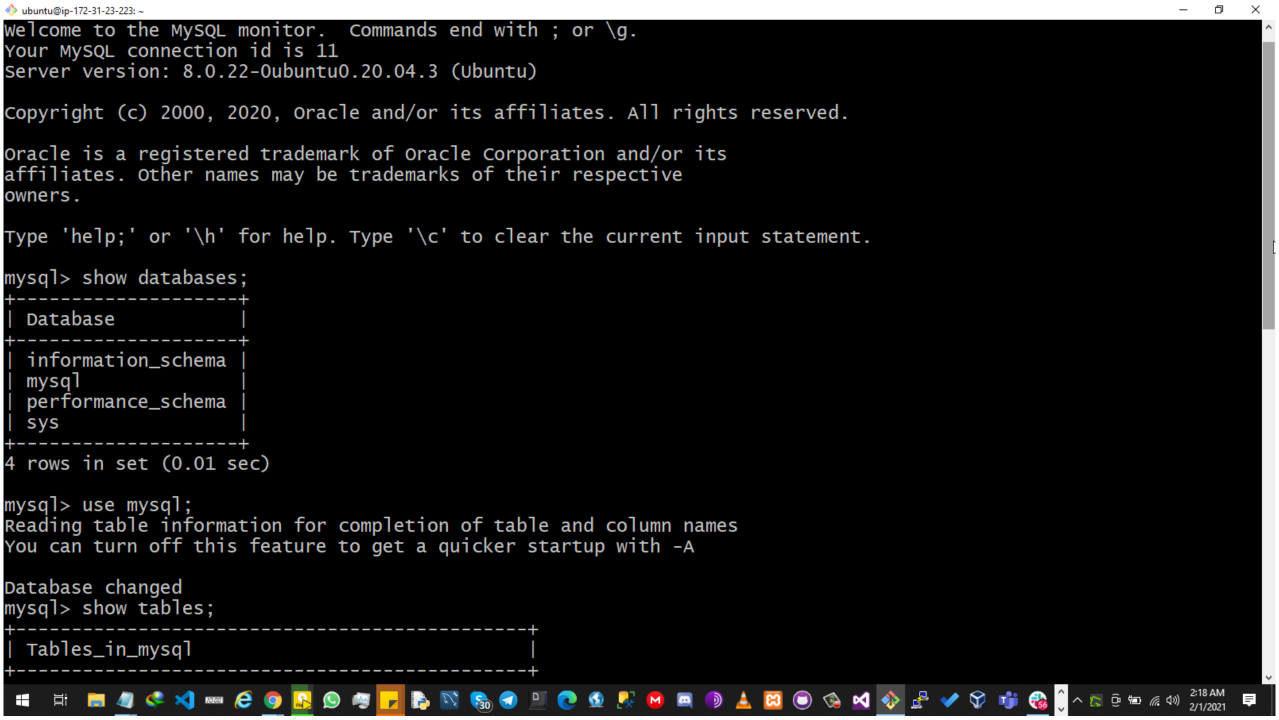


```
ubuntu@ip-172-31-23-223:~$ sudo mysql_secure_installation

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: 
```

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
```



```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.22-0ubuntu0.20.04.3 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

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> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.01 sec)

mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+---------------------------------------------+
| Tables_in_mysql                             |
+---------------------------------------------+
```

Type exit to leave MySQL console editor.

```
MySQL> exit
```

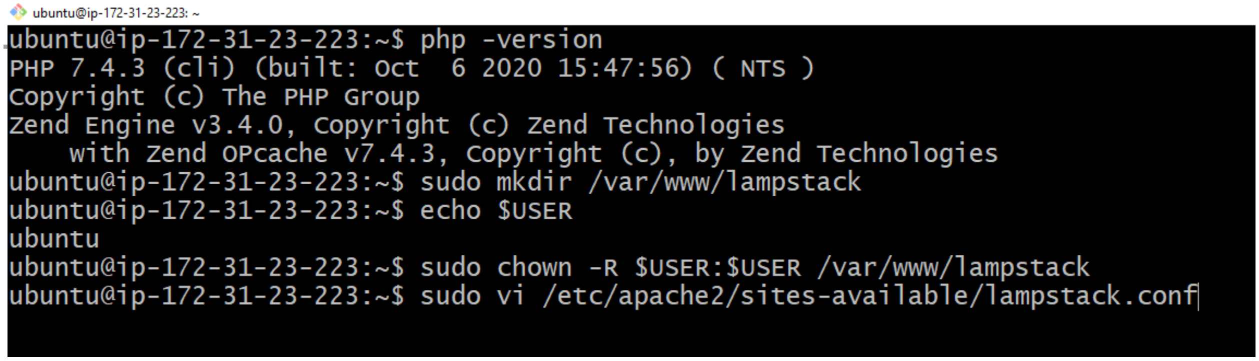
## PHP INSTALLATION

**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
```



ubuntu@ip-172-31-23-223: ~

```
ubuntu@ip-172-31-23-223:~$ php -version
PHP 7.4.3 (cli) (built: Oct  6 2020 15:47:56) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
ubuntu@ip-172-31-23-223:~$ sudo mkdir /var/www/lampstack
ubuntu@ip-172-31-23-223:~$ echo $USER
ubuntu
ubuntu@ip-172-31-23-223:~$ sudo chown -R $USER:$USER /var/www/lampstack
ubuntu@ip-172-31-23-223:~$ sudo vi /etc/apache2/sites-available/lampstack.conf
```

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>
```

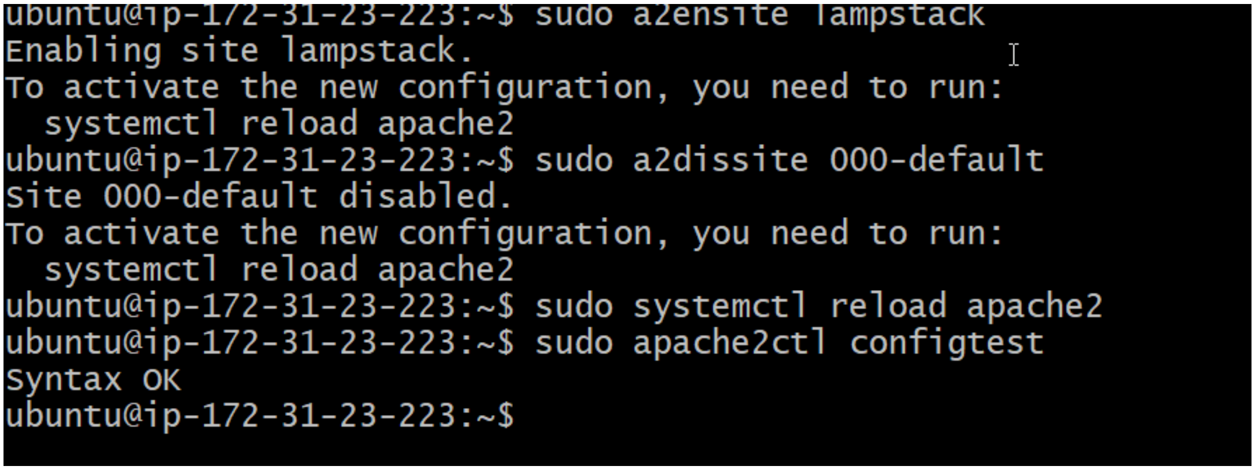
Next check the content of that directory using below command

```
$ 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
```

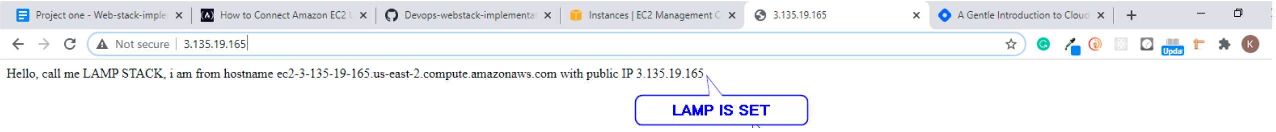


```
ubuntu@ip-172-31-23-223:~$ sudo a2ensite lampstack
Enabling site lampstack.
To activate the new configuration, you need to run:
  systemctl reload apache2
ubuntu@ip-172-31-23-223:~$ sudo a2dissite 000-default
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
ubuntu@ip-172-31-23-223:~$ sudo systemctl reload apache2
ubuntu@ip-172-31-23-223:~$ sudo apache2ctl configtest
Syntax OK
ubuntu@ip-172-31-23-223:~$
```

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) 'with public IP' $(curl -s
http://169.254.169.254/latest/meta-data/public-ipv4) > /var/www/lampstack/index.html
```

Refresh browser and check now, Hola!!!


Project one - Web-stack-imple
How to Connect Amazon EC2
Devops-webstack-implementa
Instances | EC2 Management C
3.135.19.165
A Gentle Introduction to Cloud
Not secure | 3.135.19.165

Hello, call me LAMP STACK, i am from hostname ec2-3-135-19-165.us-east-2.compute.amazonaws.com with public IP 3.135.19.165

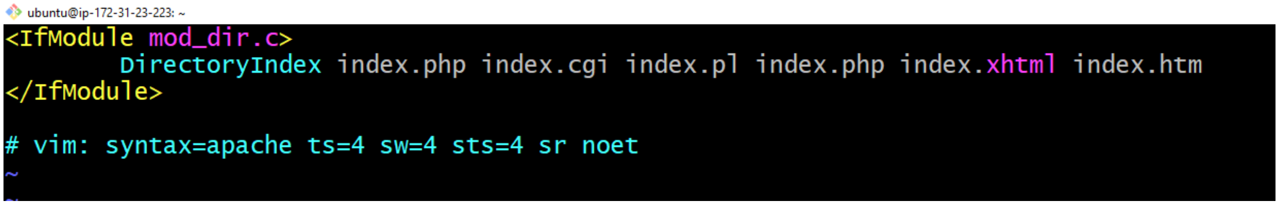
LAMP IS SET

**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**

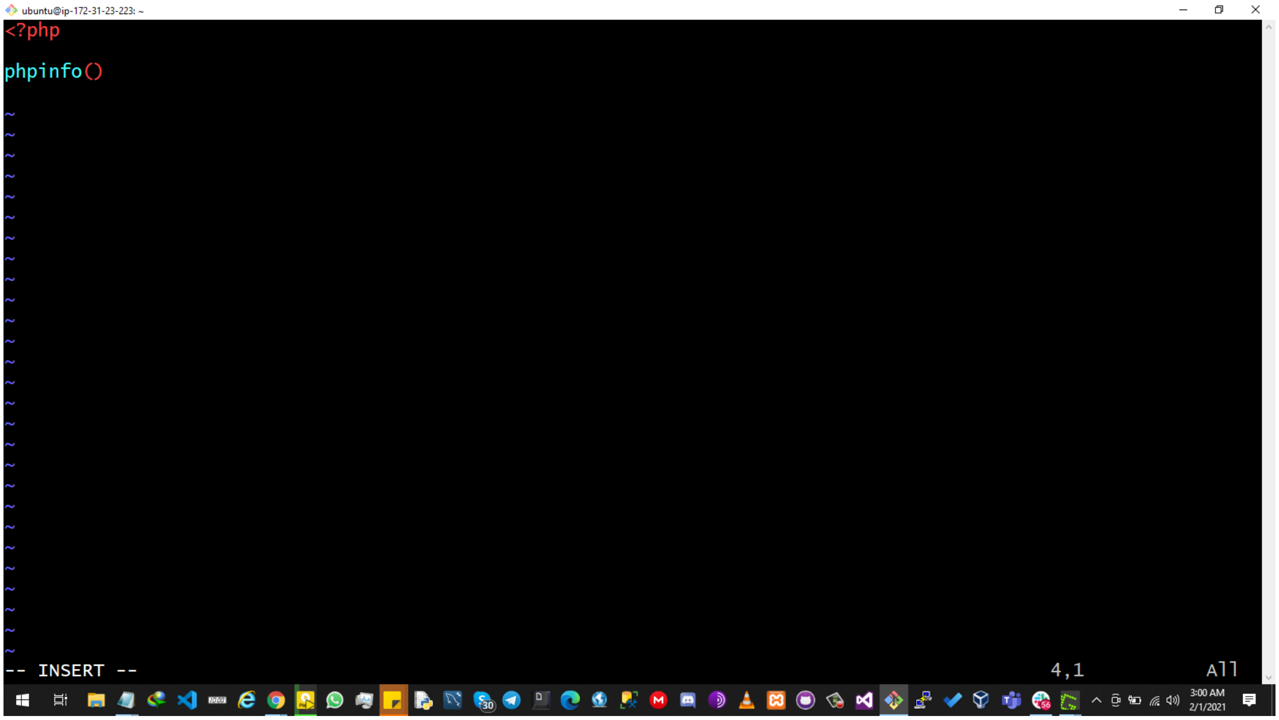
```
$ sudo vi /etc/apache2/mods-enabled/dir.conf
```



```
<IfModule mod_dir.c>
        DirectoryIndex index.php index.cgi index.pl index.php index.xhtml index.htm
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
~
```

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

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

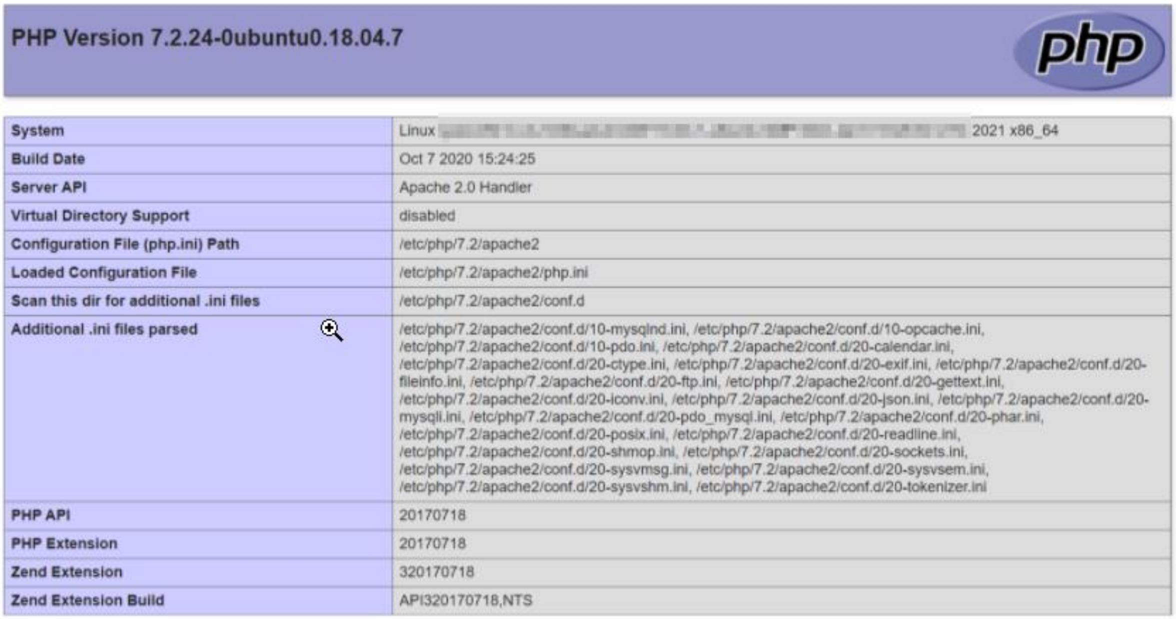


```
<?php

phpinfo()
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                  4,1           All
```

Then reload apache to effect change at reboot.

```
$ sudo systemctl reload apache2
```

Then refresh and check your browser.



PHP Version 7.2.24-0ubuntu0.18.04.7

| | |
|---|---|
| **System** | Linux [redacted] 2021 x86_64 |
| **Build Date** | Oct 7 2020 15:24:25 |
| **Server API** | Apache 2.0 Handler |
| **Virtual Directory Support** | disabled |
| **Configuration File (php.ini) Path** | /etc/php/7.2/apache2 |
| **Loaded Configuration File** | /etc/php/7.2/apache2/php.ini |
| **Scan this dir for additional .ini files** | /etc/php/7.2/apache2/conf.d |
| **Additional .ini files parsed** | /etc/php/7.2/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mysqli.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.2/apache2/conf.d/20-sysvsem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini |
| **PHP API** | 20170718 |
| **PHP Extension** | 20170718 |
| **Zend Extension** | 320170718 |
| **Zend Extension Build** | API320170718,NTS |

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.