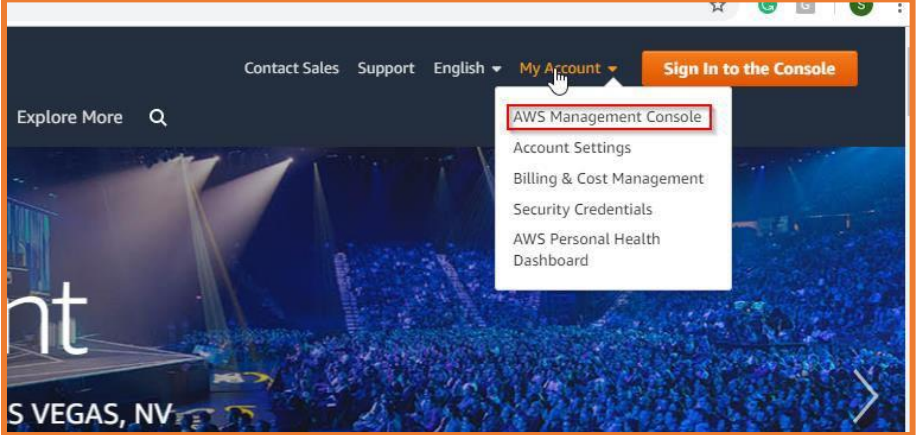
PROJECT 1 – SOLUTION

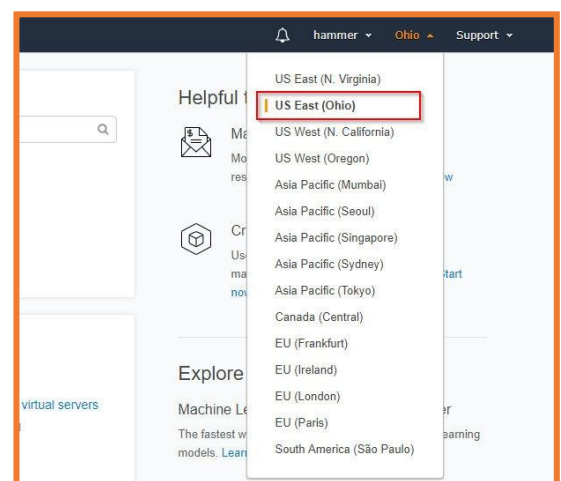
* **Connect your system with your EC2 Instance**

First you need to install Putty on your system and then connect it with your EC2 instance. Below are the steps for it:

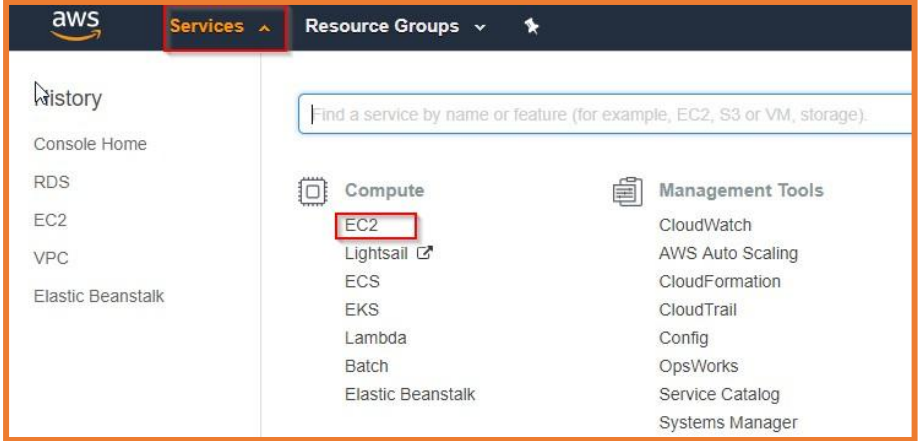
• First sign into the AWS Management Console



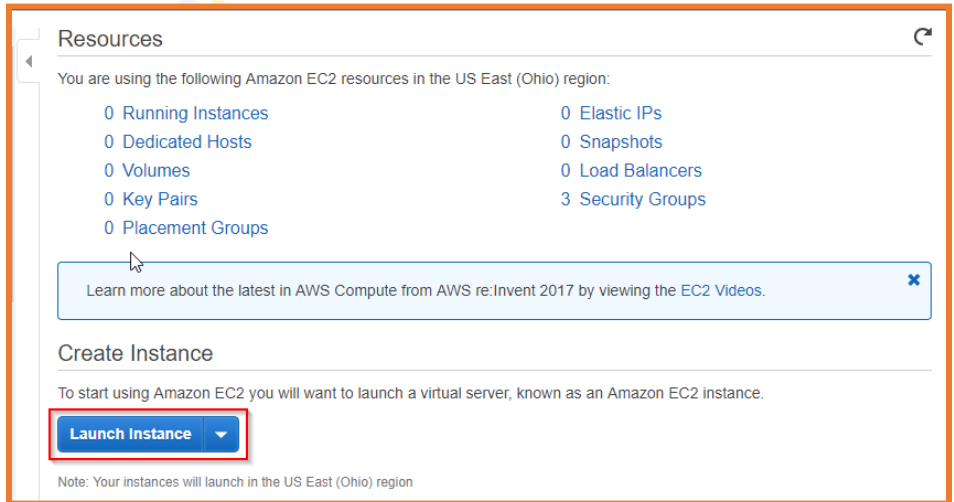
* Select any region you want, like we’ve selected Ohio here



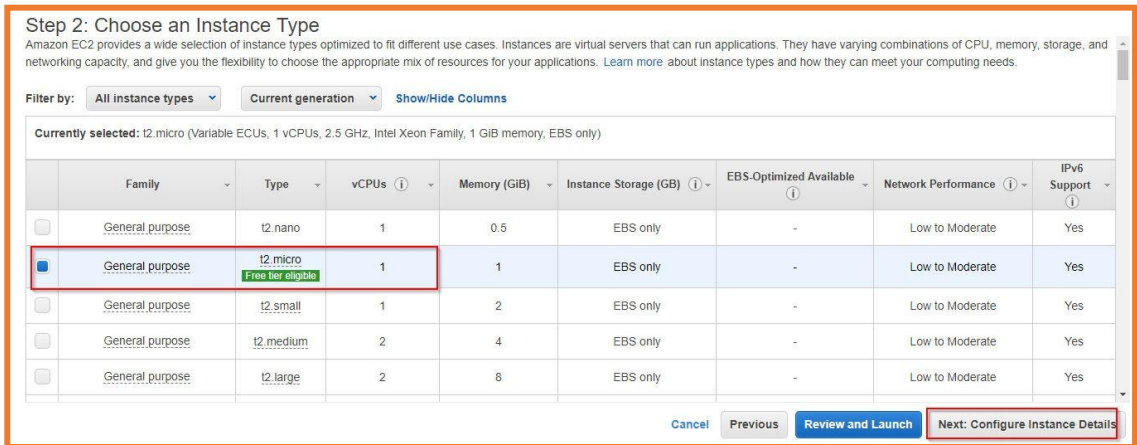
In the Services section, you must see Compute where you need to choose EC2



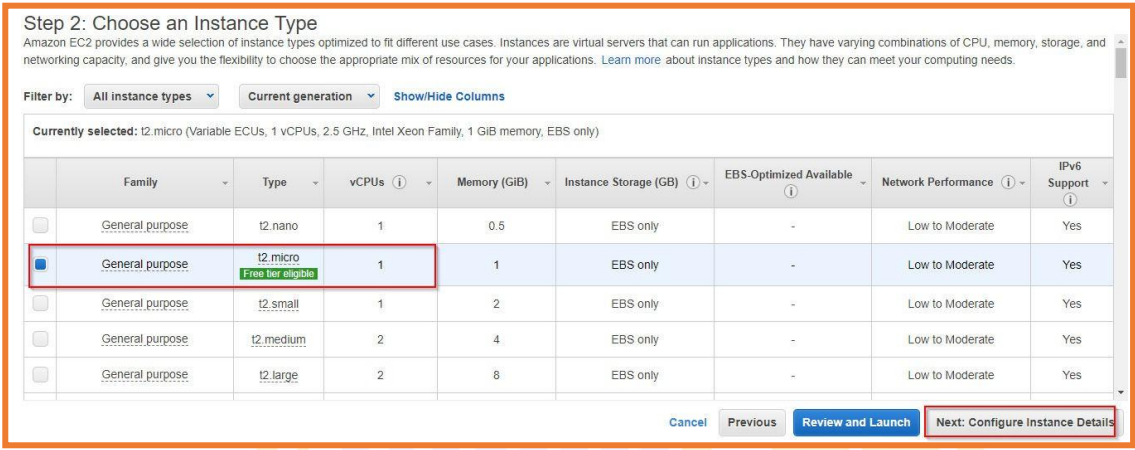
Then you will see in the Create section, there is Launch Instance option, select it



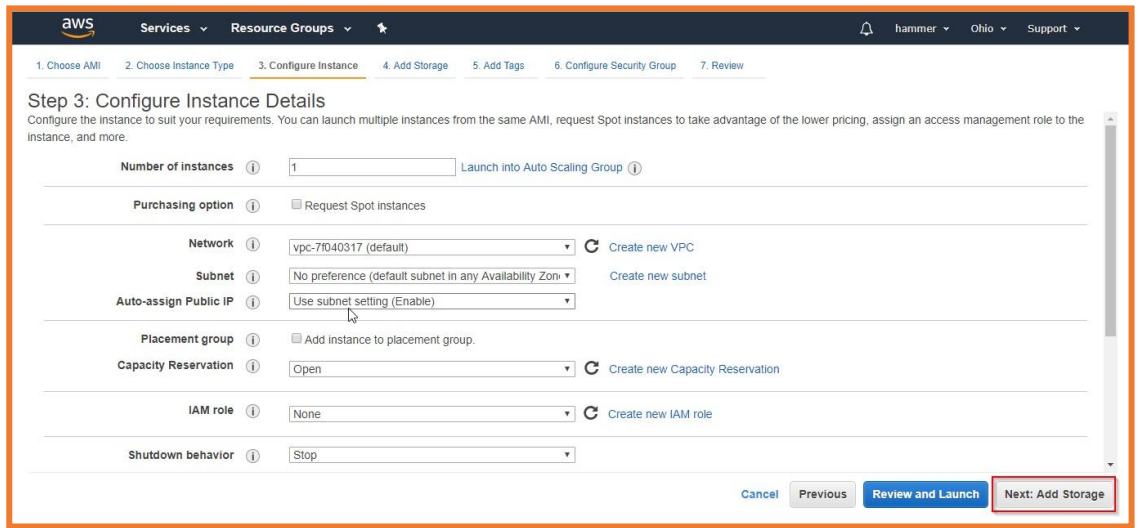
Then Select an AMI or Amazon Machine Image



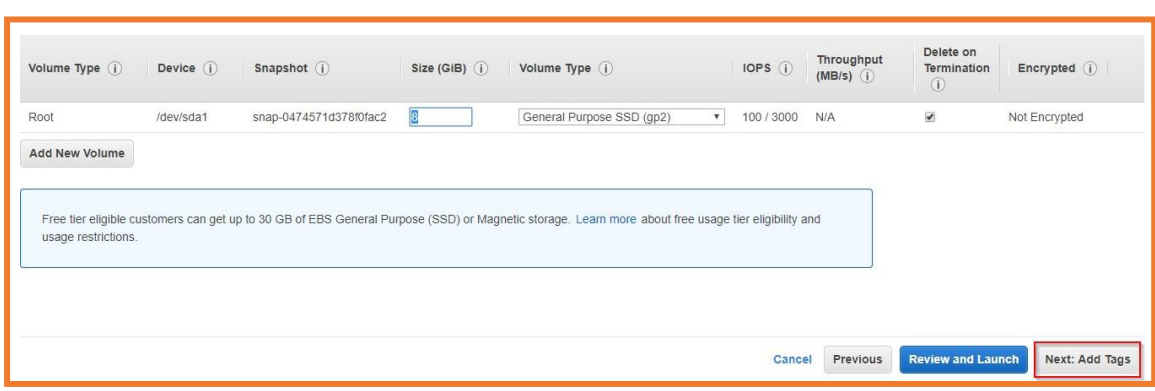
Choose your instance type, we’re choosing Free tier for the demo purpose



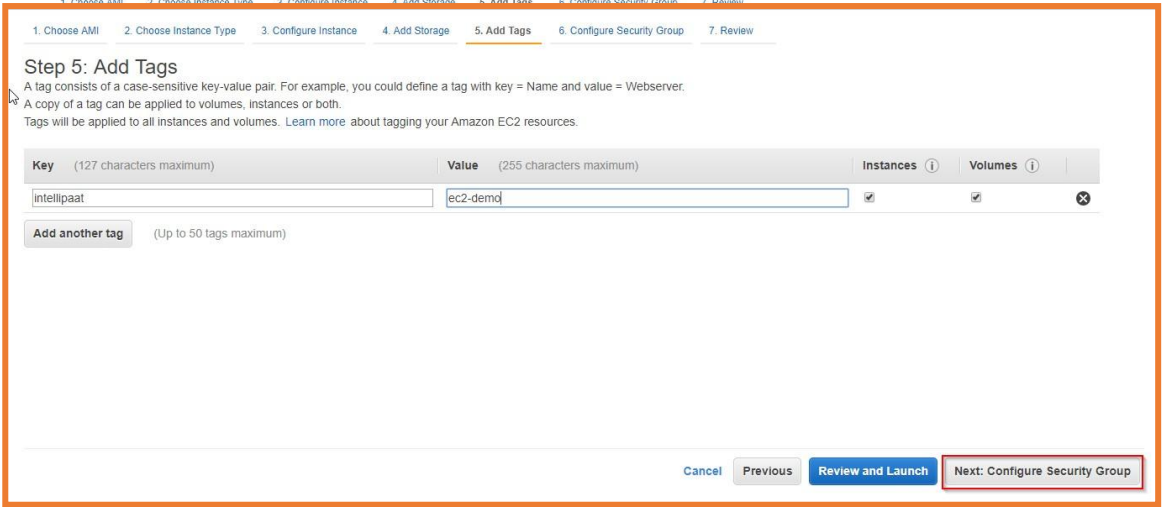
Next step is to configure your instance details and then there will and Add storage option, select it



Then click on Add Tags



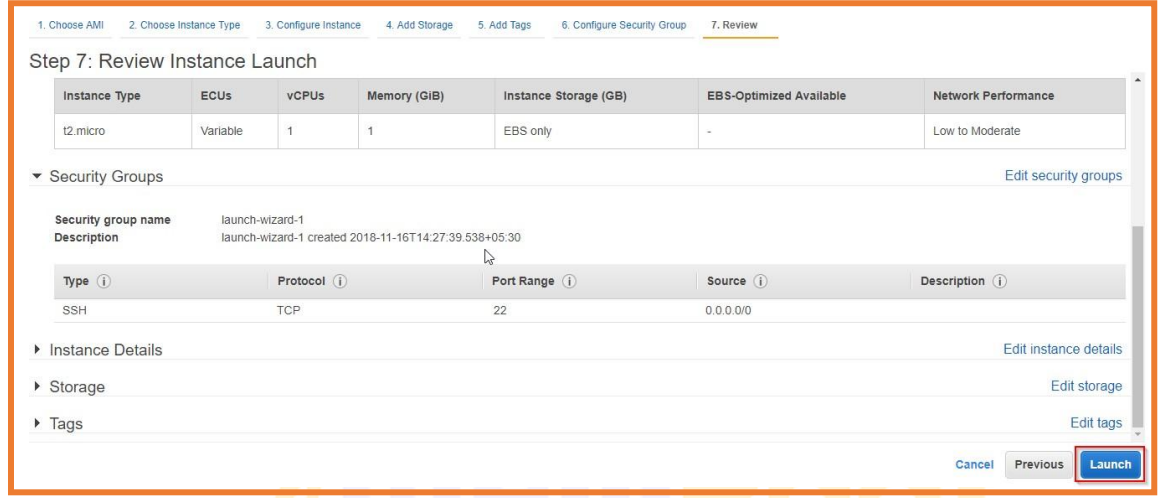
Add tags then name the key and a value, click Configure Security Group



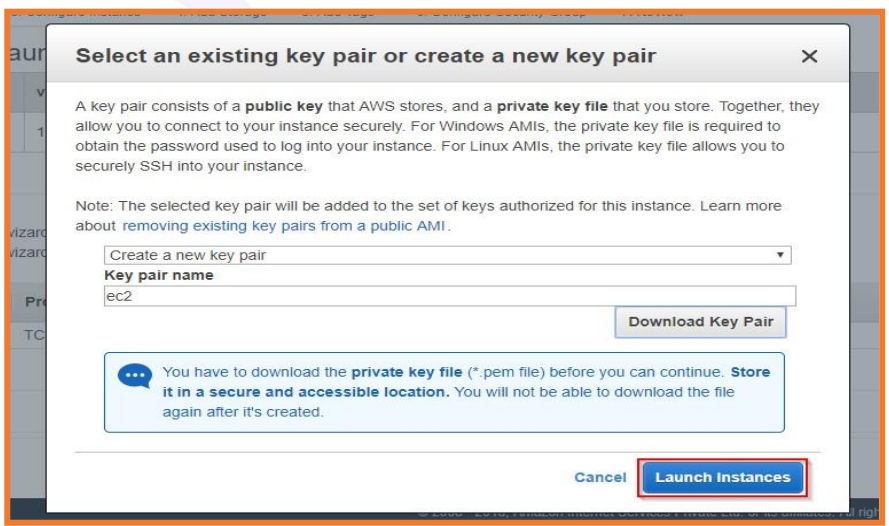
Keep the configuration of security group as it



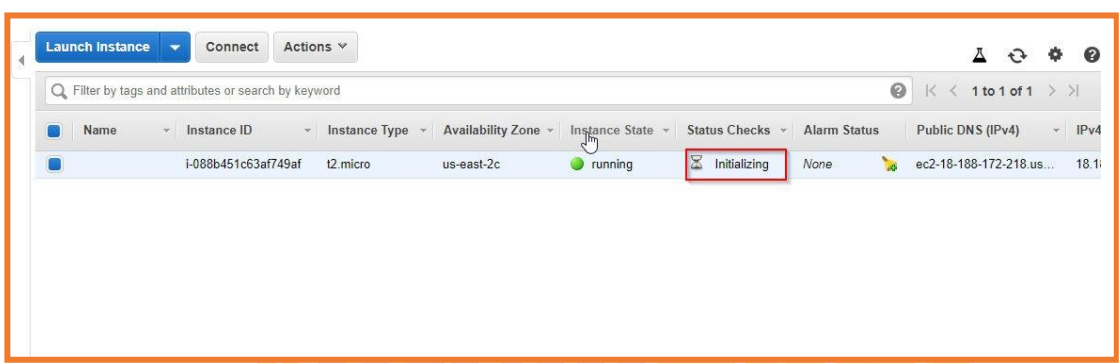
Then click Review & Launch



Then Create a key pair, download it and then Launch your instance

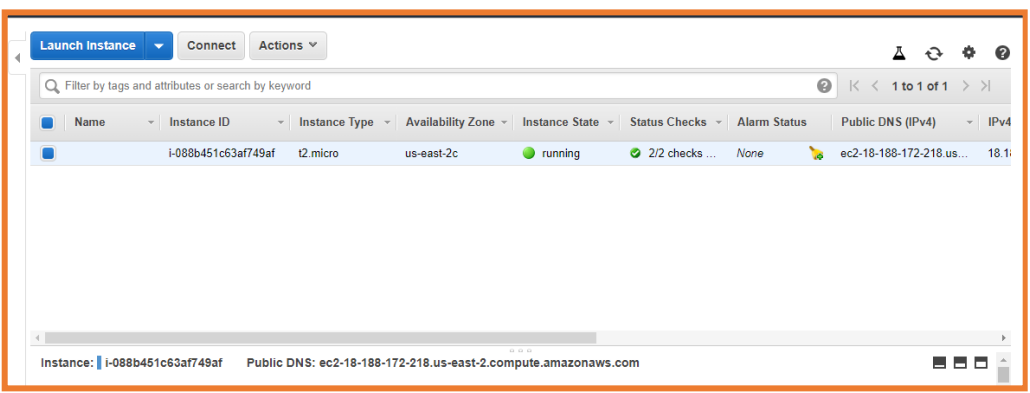


Status

You will be able to see in your status that your Instance is on Initializing stage 

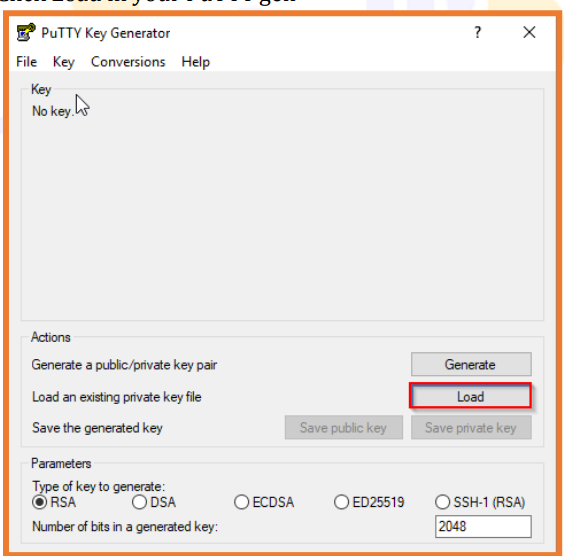


Then after few minutes, you will see that now your instance is in running stage

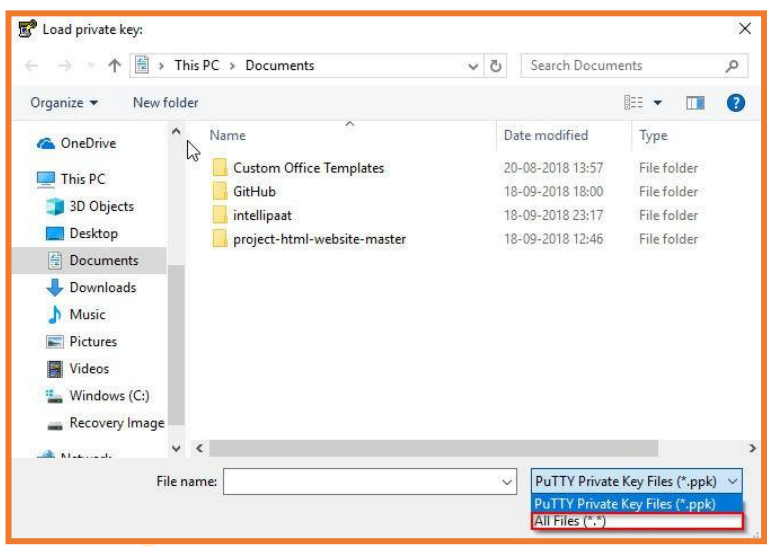


Now it’s time to convert your private key using PuTTYgen PuTTY won’t be able to support this .pem file, so you’d require a PuTTY gen tool which can convert your .pem file into .ppk format, because you need a .ppk file in order to connect it with your instance

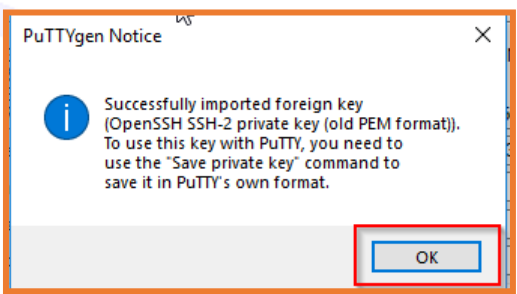
Click Load in your PuTTY gen



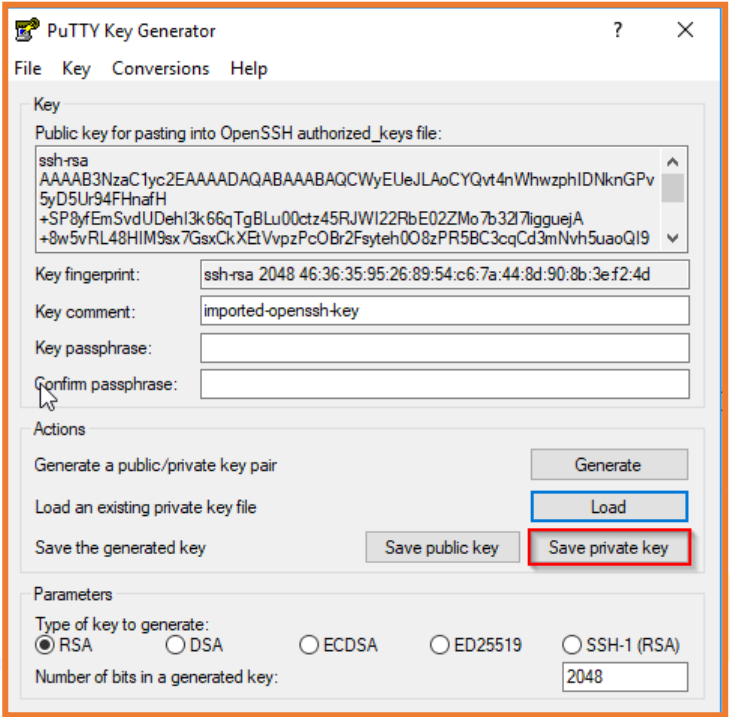
PuTTY key gen always shows the .ppk format file, so go to the right bottom bar and select the All files option as shown below



Then select the folder where you downloaded this keypair and load it there • You will see this option then click OK



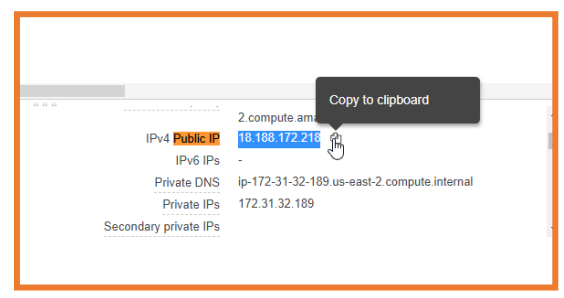
Then click on Save the Private key, PuTTY gen will give a warning about saving the key without Key passphrase, click Yes and specify the same name for your file that you gave it in the key pair



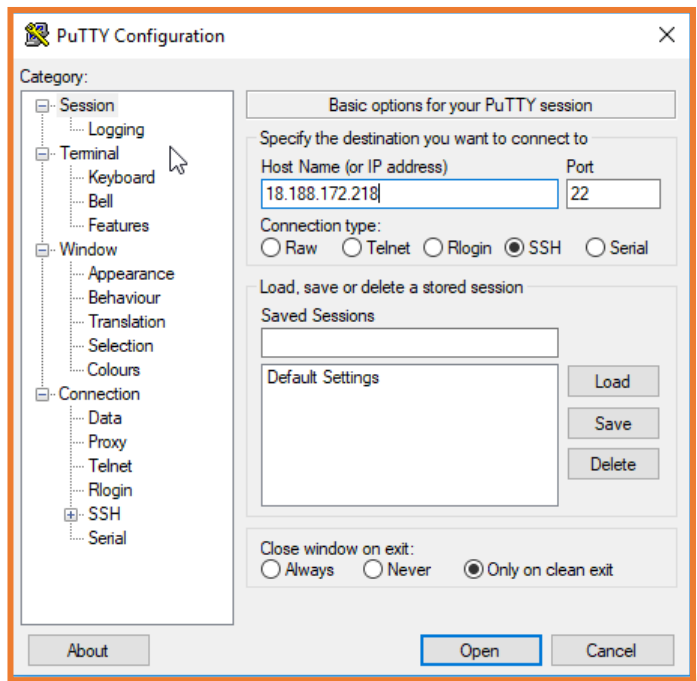
Now you will see that in your folder, the .ppk file is already added with that name you had given (in our case, it’s ec2)

**Connecting to your EC2 Instance using SSH & PuTTY**

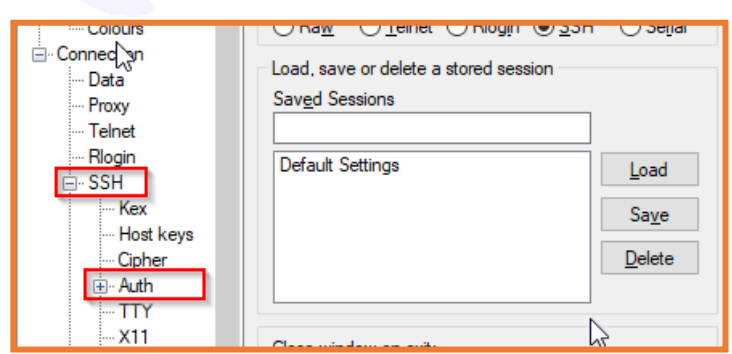
• First open PuTTY.exe then in the Host Name box, add the Public IP of your Instance



Copy paste this Public IP in your PuTTY Hostname

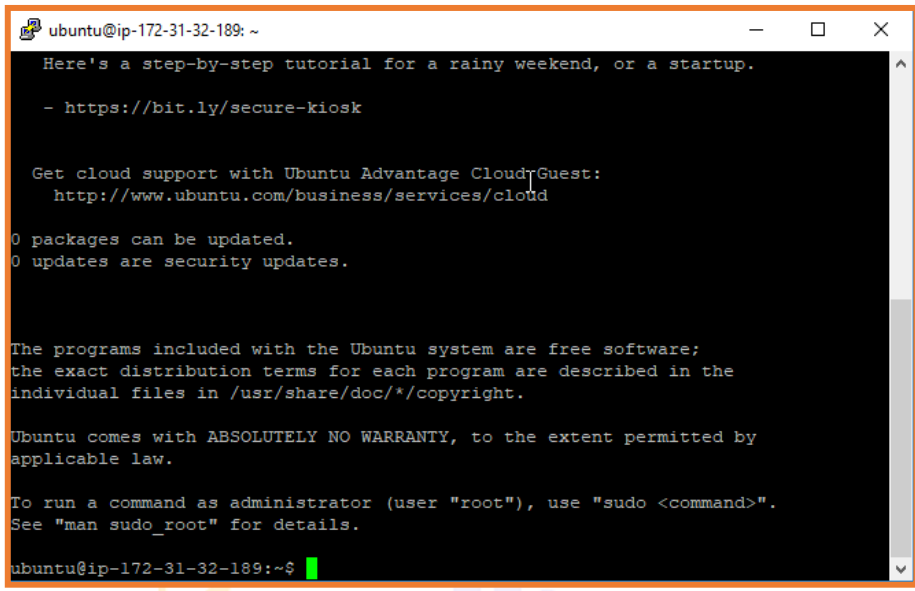


Then in the category list, expand the SSH and Click on AUTH (but don’t expandit)



Then Click Open

Login as per your OS, in our case it is ubuntu, so we will Login as: Ubuntu



* First Update your system using the command sudo apt-get update

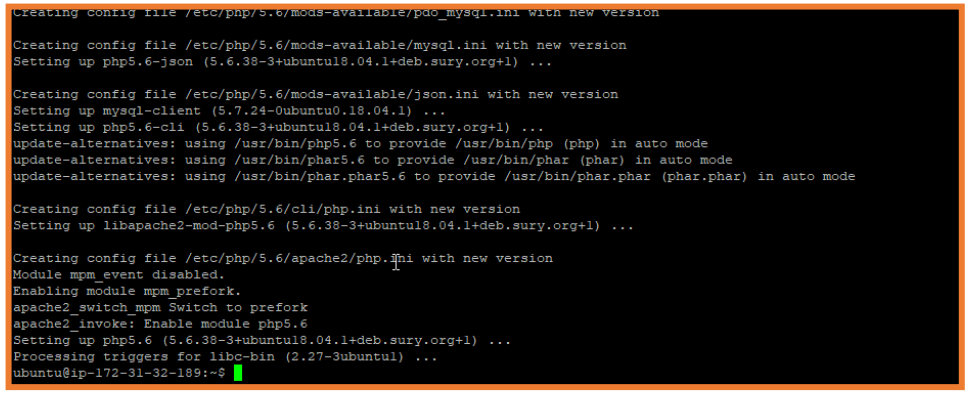
• Then use this command in PuTTY to install Apache2 sudo apt-get install apache2

• Then install php-mysql using the following command

sudo add-apt-repository -y ppa:ondrej/php

sudo apt install php5.6 mysql-client php5.6-mysqli

Now everything is updated in your system



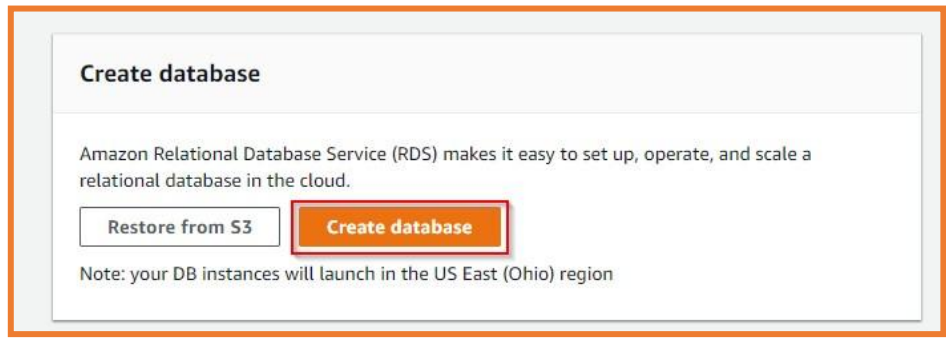
Now we connect mysql with the RDS

• Go to your AWS Management Console

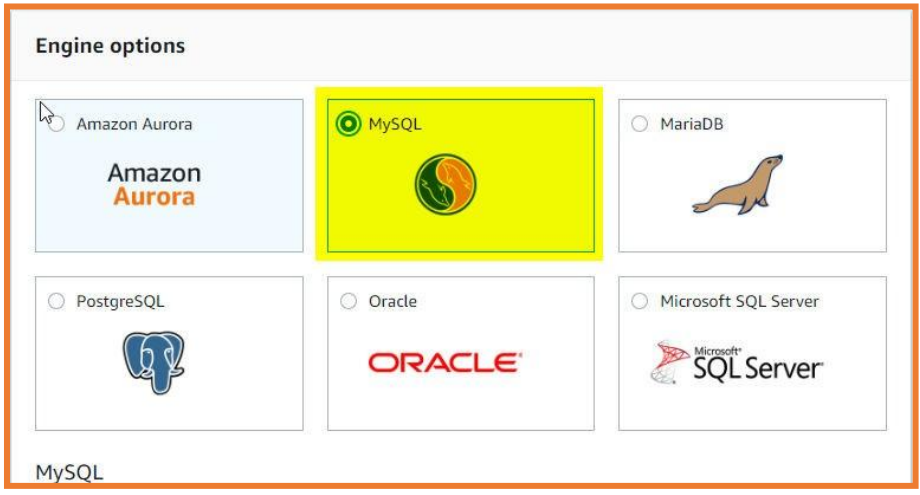
• Select RDS



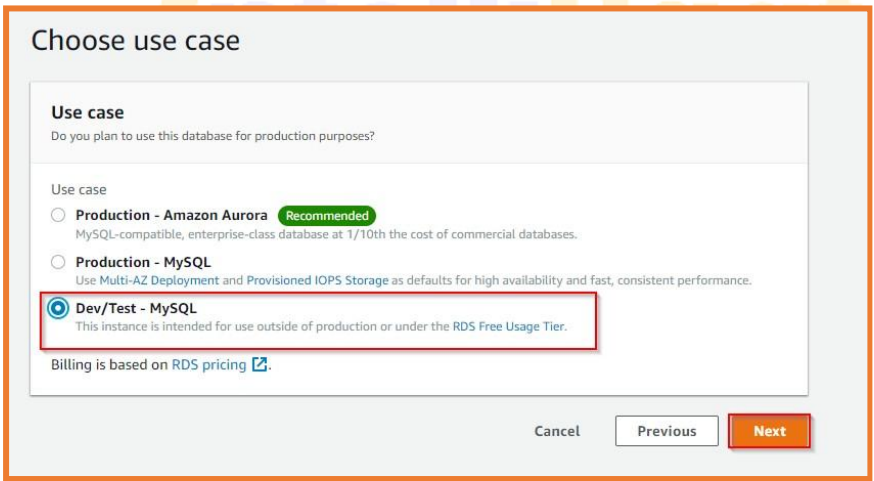
Then click on Create Database



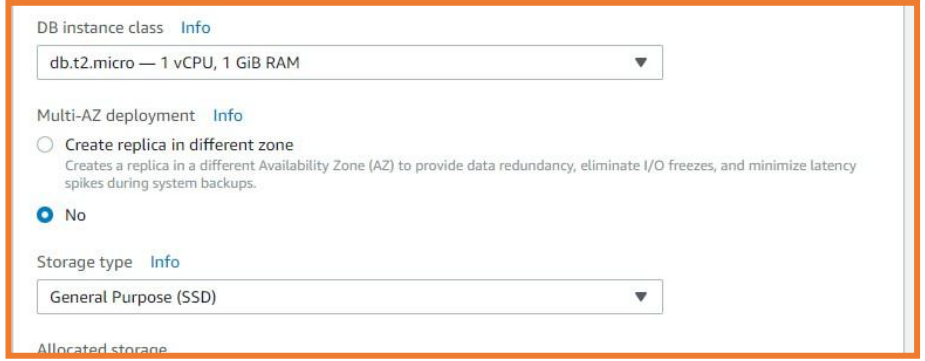
Select the MySQL Engine and click Next



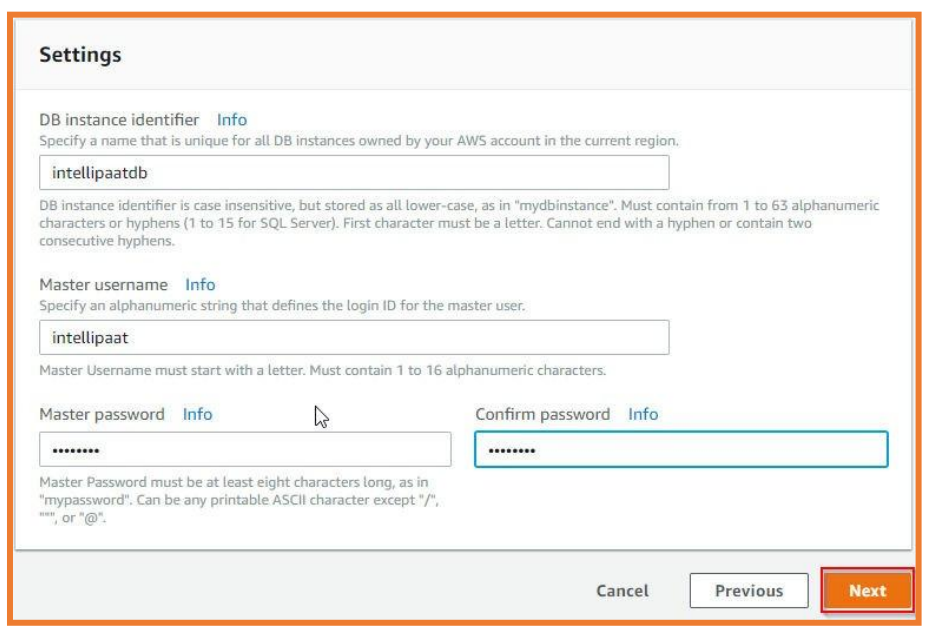
Since we’re using it for the demo purpose, so we’ll choose the Dev/Test -MySQL option only and then click Next



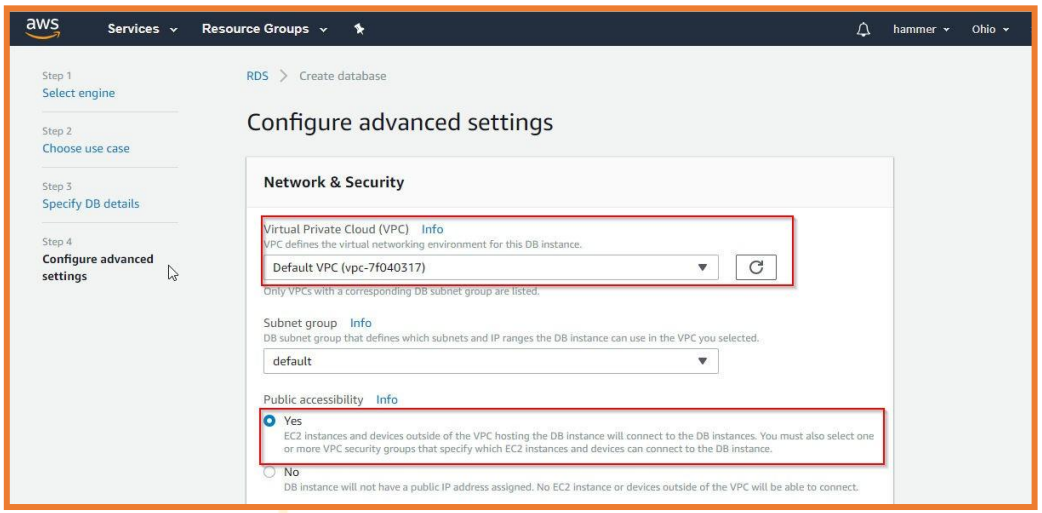
Specify DB Details, make sure to choose only db.t2.micro in DB Instance Class



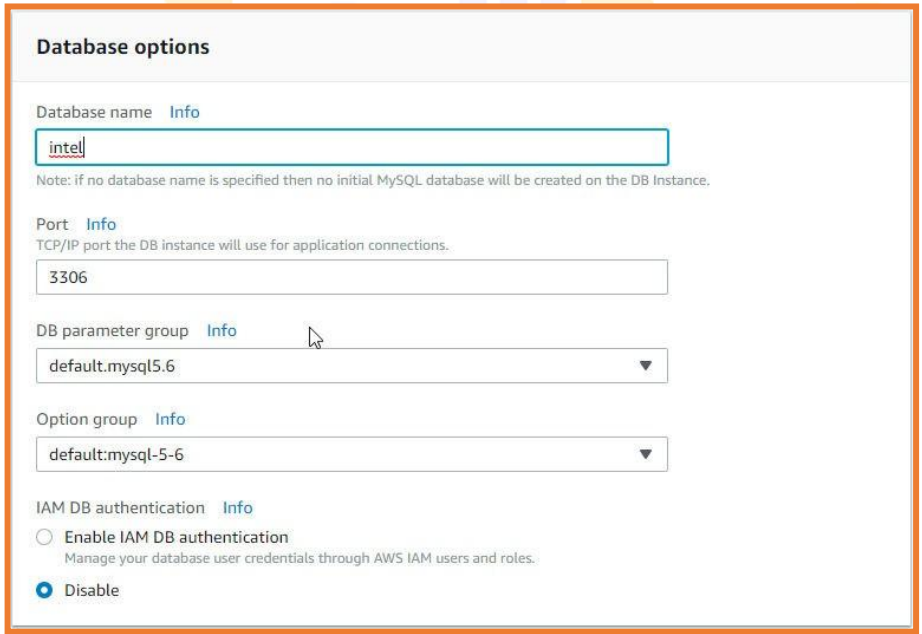
Enter these credentials (Note: Make sure you remember these credentials, as they will be required for connecting the RDS with your PuTTY



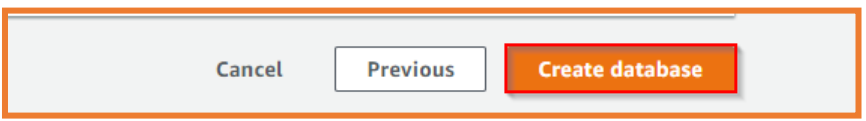
Then in the Configure Advanced Option, make sure to keep the VPC as default, along with the Public Accessibility as Yes



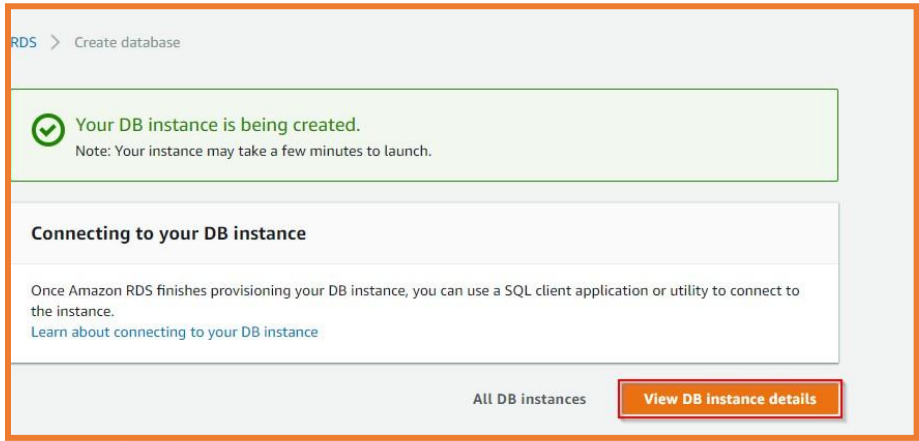
In the Database Options, name the Database and keep the other artefacts as it is



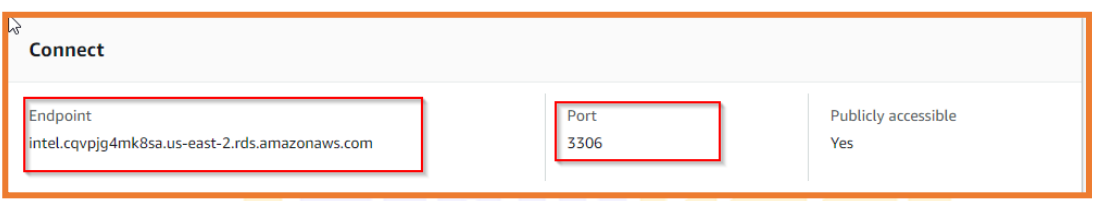
Then click on Create database



Then you can check your instance status



It may take few minutes for RDS to go from Initial to Running stage, you will observe that Endpoint and Port are not yet available (wait for few minutes)



Also, make sure to change some security configuration in the RDS • Go to your EC2 Instance Security Groups and select your group ID



Then go to RDS Security groups and select the Inbound rules panel there and click on Add Rule



Then paste the EC2 Security ID in Source> Custom > Security Group by keeping the Type as MYSQL/Aurora



Now go back to your PuTTY and use this command as shown below

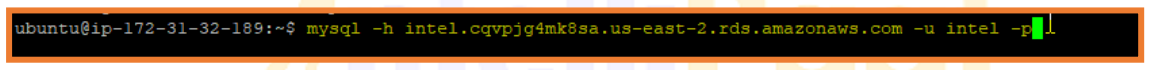
mysql -h hostname -u username -p

NOTE:

o In place of hostname, make sure to use your Endpoint from RDS

o Username which you created

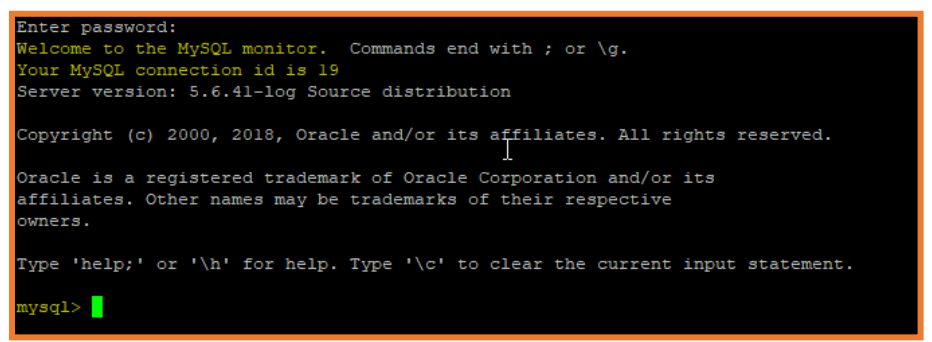
Here, we’re using our own Endpoint and username and password used



Use the command as shown below

• After this, it will ask for your password, in our case, password is: intel123

• Then it will show that you’re connected to the mysql



Filezilla

* Now install Filezilla

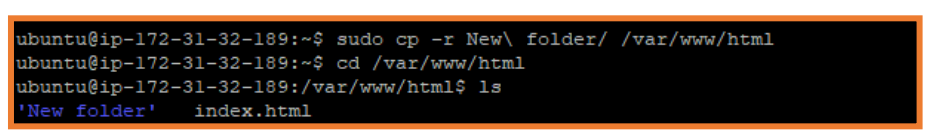
• In order to connect it, enter hostname as the Endpoint of EC2 and Username as Ubuntu and no need to keep the password, then quickconnect.



Now your Filezilla is connected with your EC2 instance

Create a ‘New Folder’ of your website in your Desktop

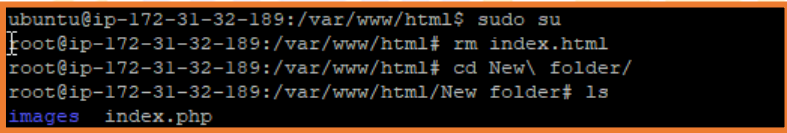
And copy paste it in your Filezilla Remote Site path: /home/ubuntu



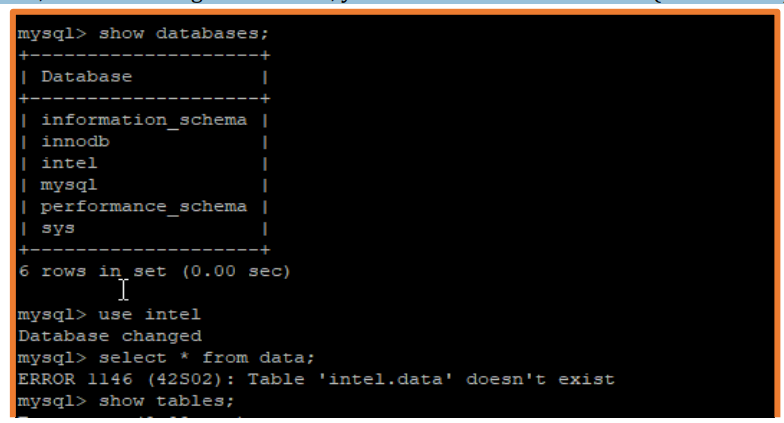
Now go back to your PuTTY, where you will see that it contains the index.html file

Now you need to remove this ‘index.html’ file and add ‘index.php’ in its place

For that you need to use “sudo su” and remove this file using remove command

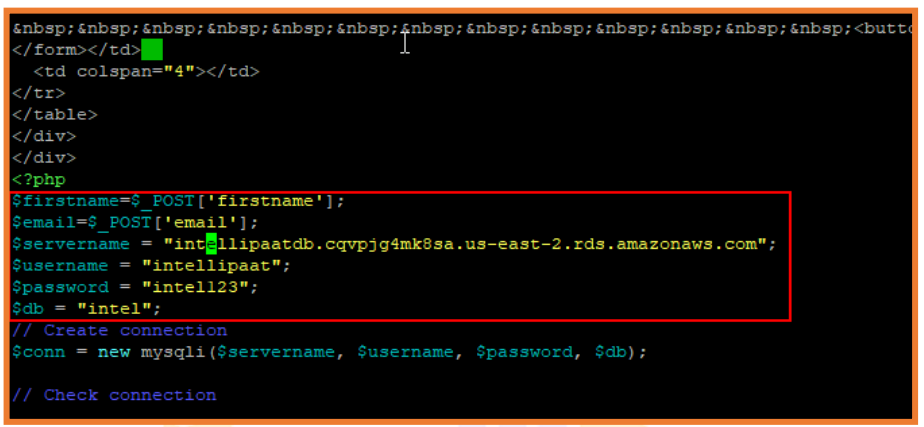


Also, before running this website, you need to create a table in it (its database)

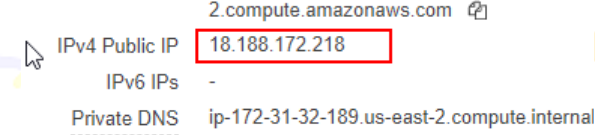


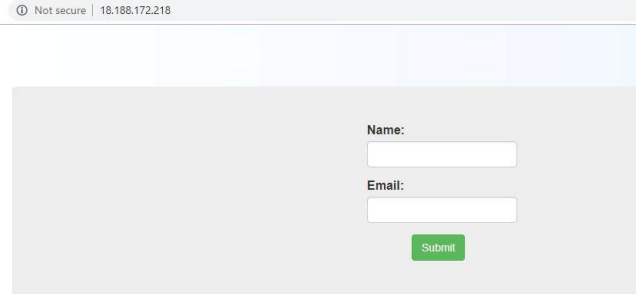
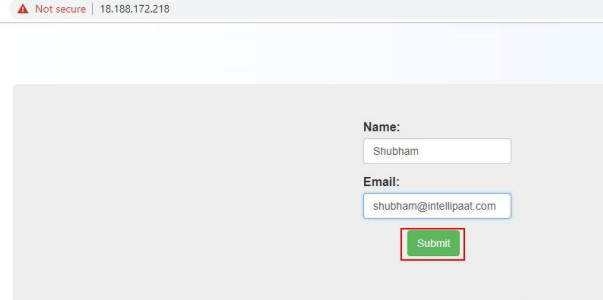
Now go to the path where website files are kept and run the index.php file by using sudo nano index.php

Now after this, GNU nano will pop up where you have to make changes in your code, you have to check if in your server name, the endpoint of your RDS is there along with username, password and db name



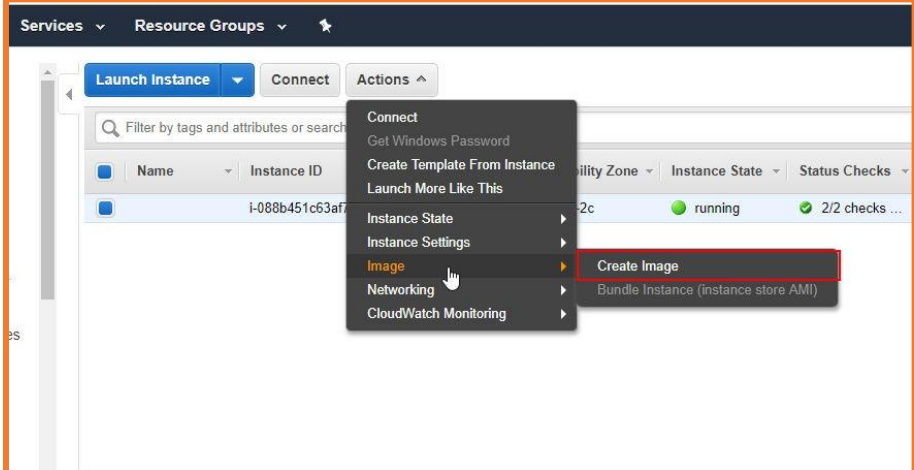
Now when you will try, and copy paste the Public IP of your EC2 Instance

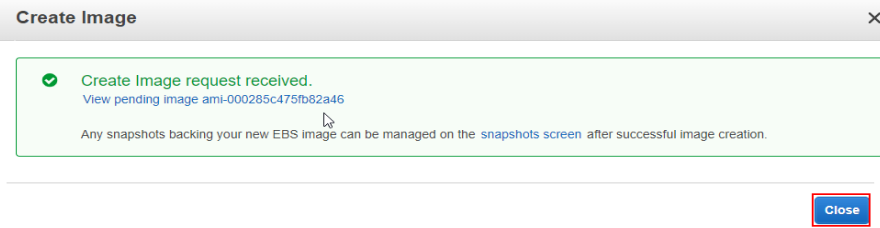


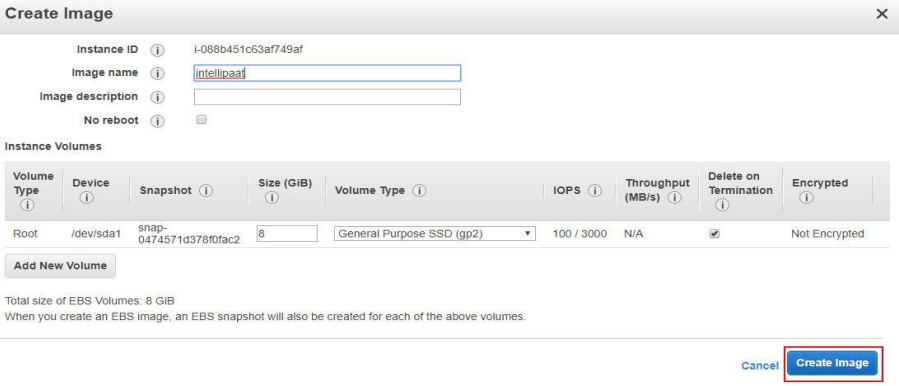
* After copying this IP to your browser, you will observe that your website is working on it
* 
* Now when you enter these details in this website, you will see the following result
* 

AUTO SCALING

Now, we’ll do the autoscaling of our website by going to our EC2 Instance and then click on Actions and Create Image







Then further, activate its autoscaling and then its classic load balancer which directs the traffic to your website directly