# Project - 1: Deploying a
# Multi-Tier Website Using AWS EC2

**Assignment Submitted By:-Hitesh Chauhan**
**Course Offered: -Advanced Cloud Computing and Devops**
**Assignment By: -Intellipaat**
**Trainer: -Puneet Gavri**
**Date Of Submission: -04/12/2024**

## Description:

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic**.**

## Problem Statement:

Company ABC wants to move their product to AWS. They have the following things set up right now:
1. MySQL DB
2. Website (PHP)

The company wants high availability on this product, therefore wants Auto Scaling to be enabled on this website.

**Steps To Solve:**
1. Launch an EC2 Instance
2. Enable Auto Scaling on these instances (minimum 2)
3. Create an RDS Instance
4. Create Database & Table in RDS instance:
a. Database name: intel
b. Table name: data
c. Database password: intel123
5. Change hostname in website
6. Allow traffic from EC2 to RDS instance
7. Allow all-traffic to EC2 instance

# Project- 1: Solution

1.Login to the AWS Console providing your credentials.
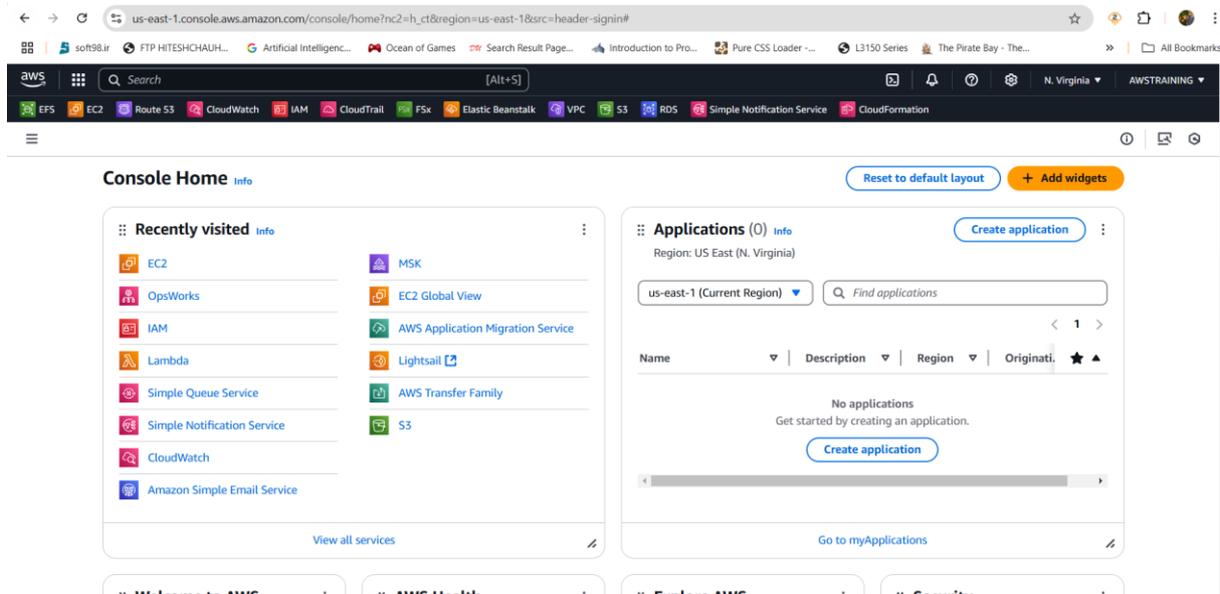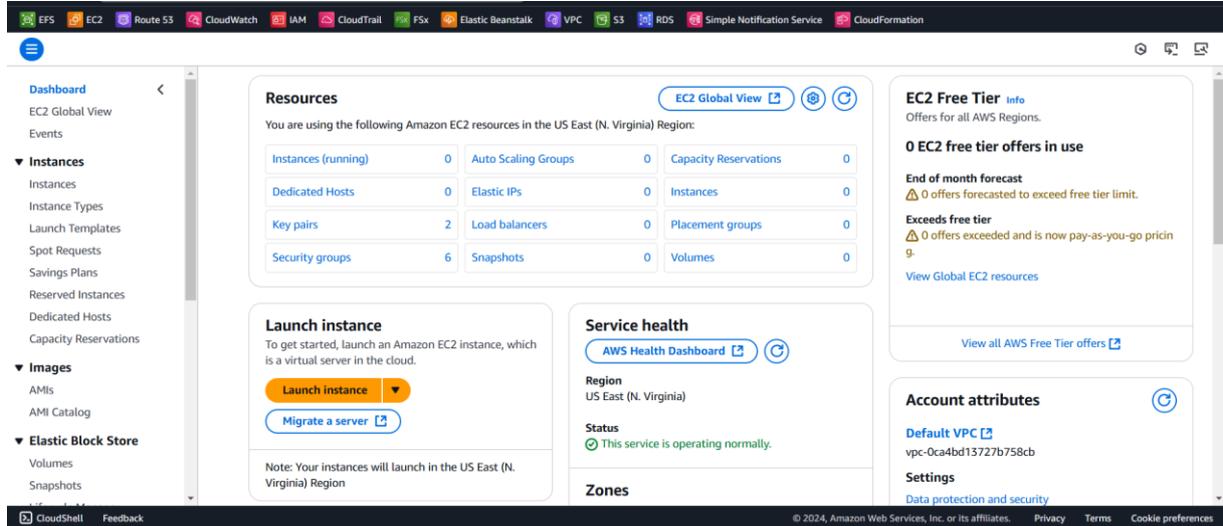


2. Select any region you want. We've selected N.Virgania here
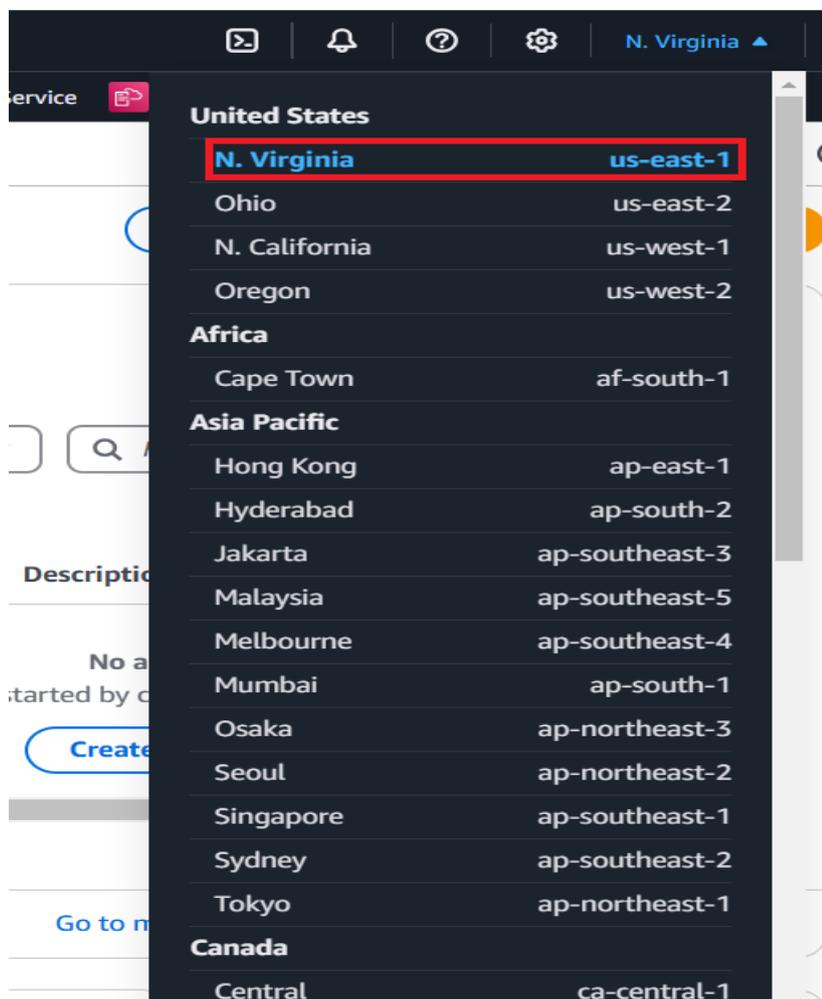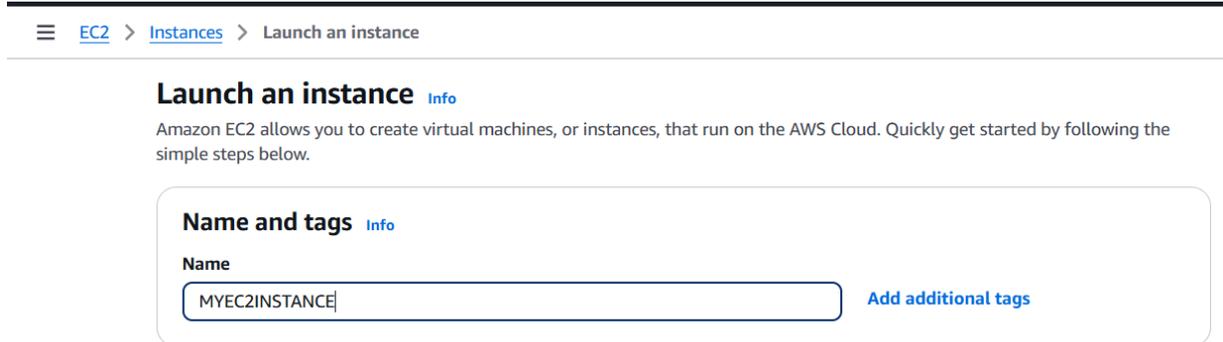
Then Click EC2 and after click the ec2 you will open this page.

3. In the **Services** section, you'll see **Compute** where you need to choose **EC2.**

4. Then in the Create Instance section, select the **Launch Instance** option



Assign ec2 instance name



5. Then Select an **AMI** or **Amazon Machine Image**

## 6. Choose your instance type. We're choosing Free tier for demo purposes



## 7.Choose Your Key pair for connect to the instance.



## 8.Choose Network Setting And Security Group Setting.

9. Configure your instance details and then select the

**Add storage** option



10.Then Click Launch Instance

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

Initializing stage.
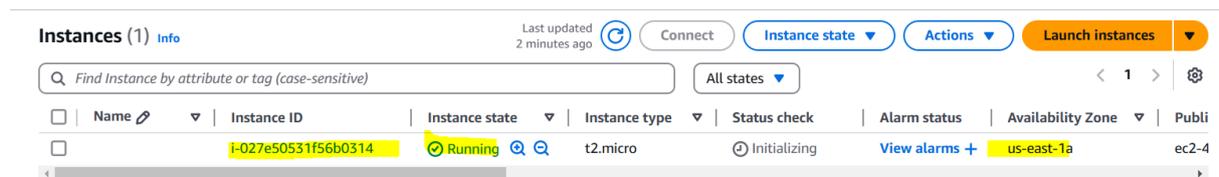


MobaXterm won't be able to support this **.pem** file, so you'd require a PuTTY gen tool which can convert your **.pem** file into **.ppk** format, because you need a **.ppk** file in order to connect it with your instance.
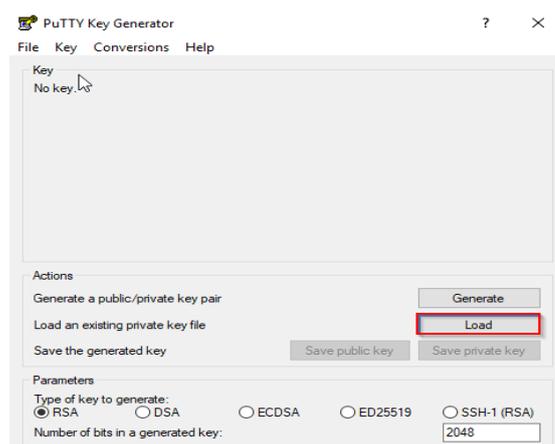
11. PuTTYgen key always shows the .ppk format file, so go to the right bottom bar and select the All files option as shown below
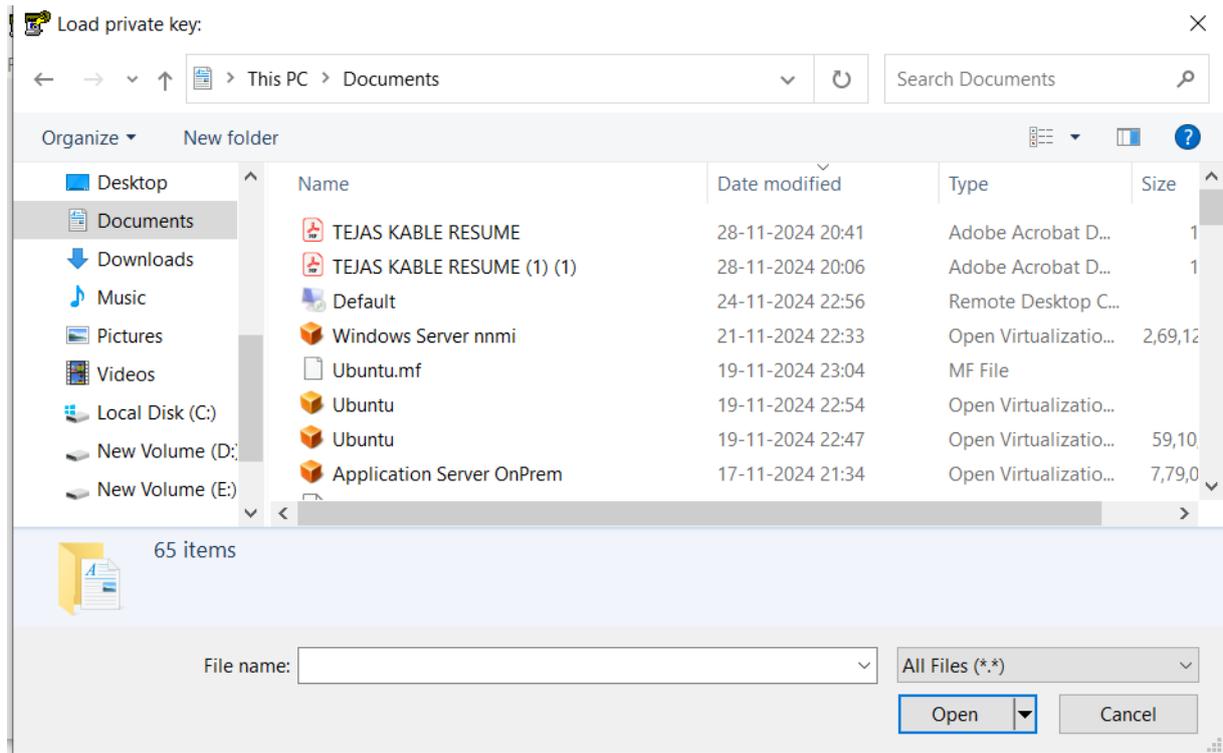


12. Then select the folder where you downloaded this keypair and load it there

13. You will see this option then click OK
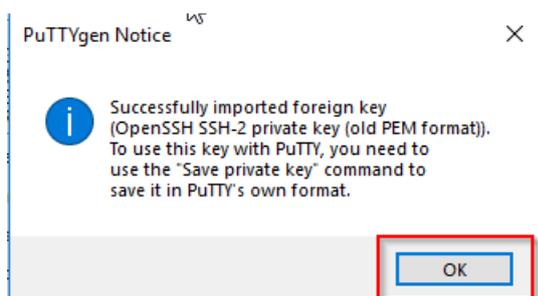


14.Then click on Save the Private key, PuTTY gen will give a warning about saving the key without Key passphrase, click Yes and specify the same name for your file that you gave it in the key pair.
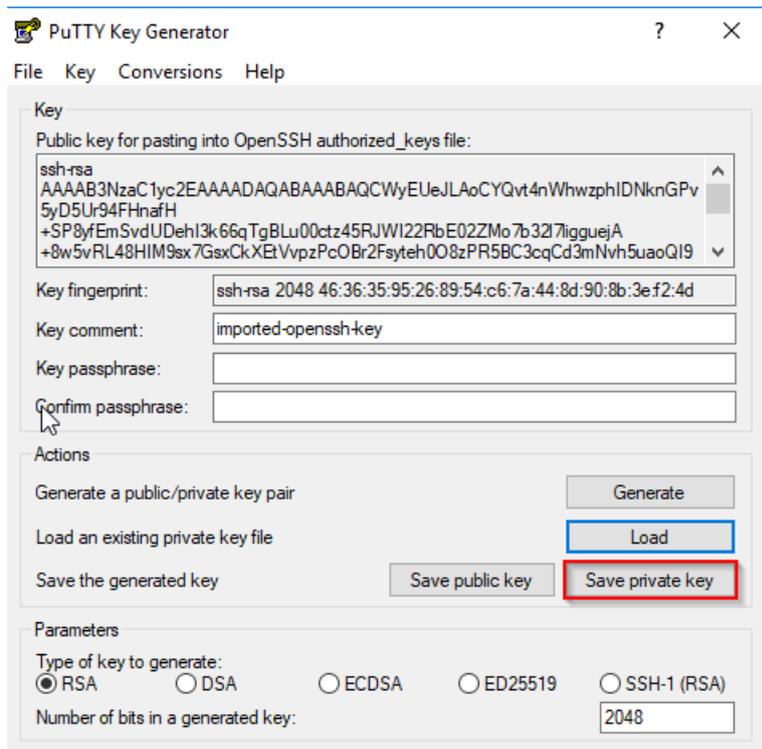
15. Then click on Save the Private key, PuTTY gen will give a warning about saving the key without Key passphrase, click Yes and specify the same name for your file that you gave it in the key pair.

16. 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 & Mobaxterm:**

First open MobaXterm then in the Host Name box, add the Public IP of

your Instance.

Open MobaXterm And Click Sessions
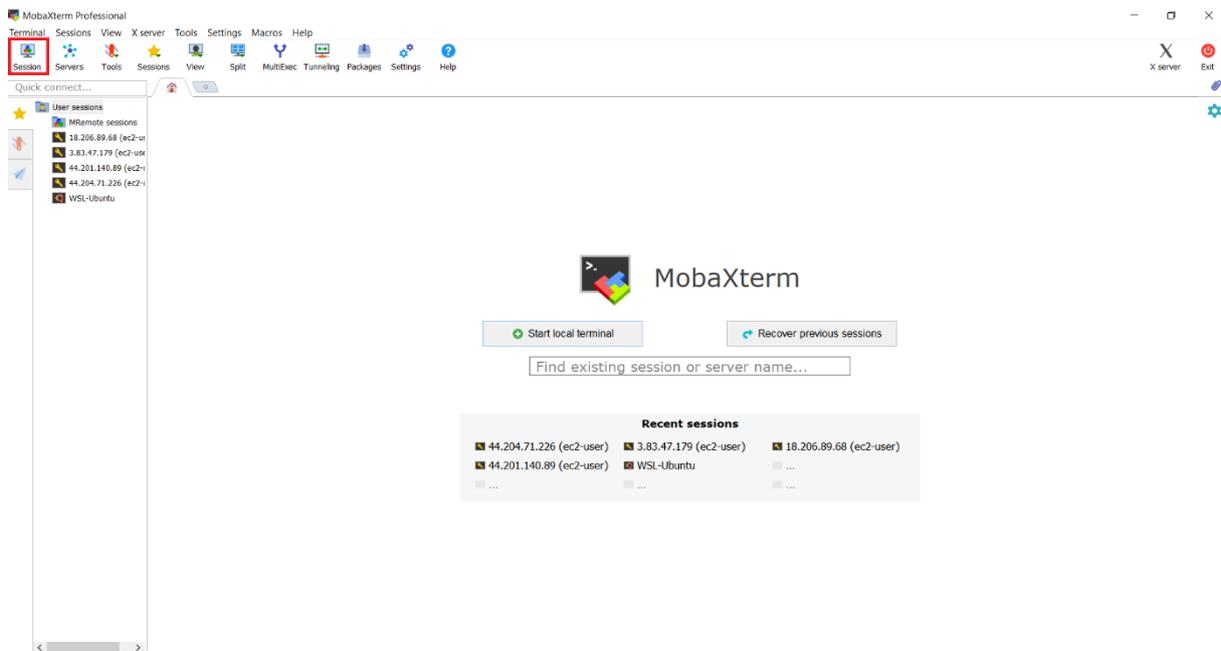


After Click the Session.now we choose ssh in this case.



After Click the SSH,you will fill the details like

Remote Host:-44.204.46.79

Specify Username:-Ubuntu

Then We need to upload the private key .ppk file format.we got converted through puttygen.then click ok.
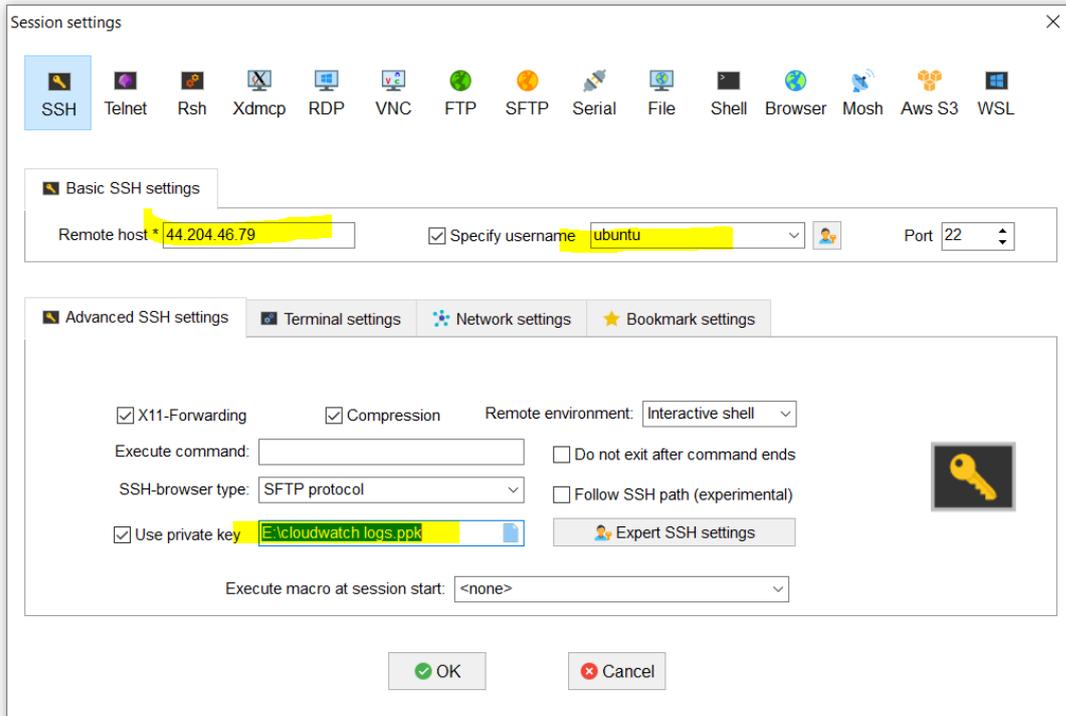
After Click ok You will prompt Connection to public ip.now click accept.
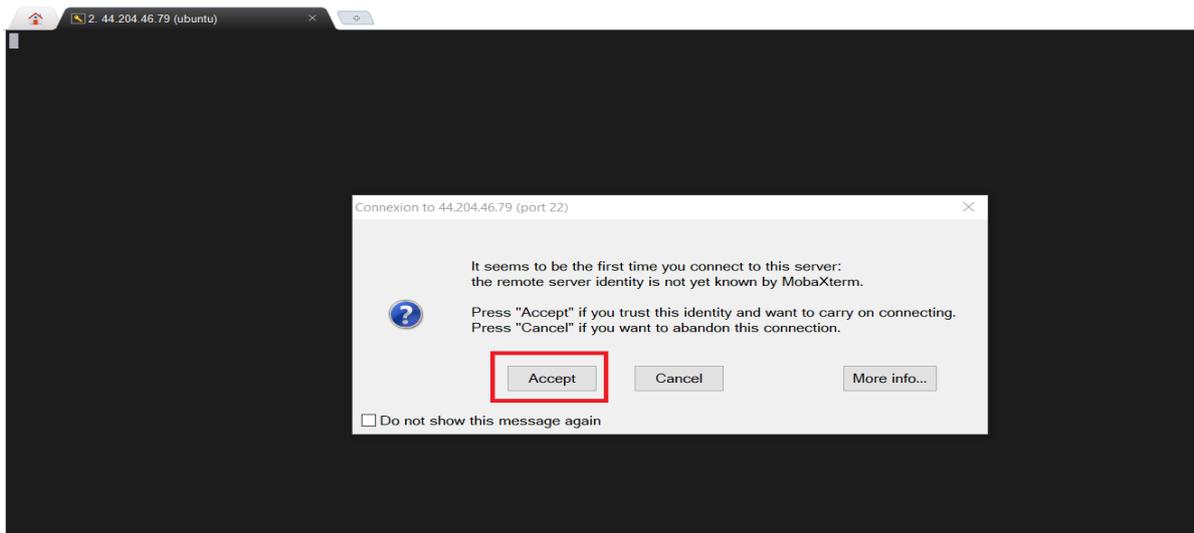


After Click the accept you will; connect the instance through mobaxterm.

After Connection the Instance we need to perform the below task from server.

First Update your system using the command

**sudo apt-get update**



Then use this command in MobaXterm to install Apache2

**sudo apt-get install apache2**

```
root@ip-172-31-82-41:/home/ubuntu# sudo apt-get install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 60 not upgraded.
Need to get 2084 kB of archives.
After this operation, 8094 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

```
Enabling module mpm_event.
Enabling module authz_core.
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /usr/lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /usr/lib/systemd/system/apache-htcacheclean.service.
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-82-41:/home/ubuntu#
```

Then install php-mysql using the following command

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

```
root@ip-172-31-82-41:/home/ubuntu# sudo add-apt-repository -y ppa:ondrej/php
PPA publishes dbgsym, you may need to include 'main/debug' component
Repository: 'Types: deb
URIs: https://ppa.launchpadcontent.net/ondrej/php/ubuntu/
Suites: noble
Components: main
'
Description:
Co-installable PHP versions: PHP 5.6, PHP 7.x, PHP 8.x and most requested extensions are included. Only Supported Ubuntu Releases (https://wiki.ubuntu.com/Releases) a
re provided.

Debian oldstable and stable packages are provided as well: https://deb.sury.org/#debian-dpa

You can get more information about the packages at https://deb.sury.org

BUGS&FEATURES: This PPA now has a issue tracker:
https://deb.sury.org/#bug-reporting

CAVEATS:
1. If you are using php-gearman, you need to add ppa:ondrej/pkg-gearman
2. If you are using apache2, you are advised to add ppa:ondrej/apache2
3. If you are using nginx, you are advised to add ppa:ondrej/nginx-mainline
   or ppa:ondrej/nginx

PLEASE READ: If you like my work and want to give me a little motivation, please consider donating regularly: https://donate.sury.org/

WARNING: add-apt-repository is broken with non-UTF-8 locales, see
https://github.com/oerdnj/deb.sury.org/issues/56 for workaround:

# LC_ALL=C.UTF-8 add-apt-repository ppa:ondrej/php
More info: https://launchpad.net/~ondrej/+archive/ubuntu/php
Adding repository.
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:5 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble InRelease [24.4 kB]
Get:6 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main amd64 Packages [126 kB]
Get:7 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main Translation-en [39.0 kB]
Fetched 189 kB in 3s (63.4 kB/s)
Reading package lists... Done
root@ip-172-31-82-41:/home/ubuntu#
```

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



```
Fetched 189 kB in 3s (63.4 kB/s)
Reading package lists... Done
root@ip-172-31-82-41:/home/ubuntu# sudo apt install php5.6 mysql-client php5.6-mysqli
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'php5.6-mysql' instead of 'php5.6-mysqli'
The following additional packages will be installed:
  debsuryorg-archive-keyring libapache2-mod-php5.6 libpcre3 mysql-client-8.0 mysql-client-core-8.0 mysql-common php-common php5.6-cli php5.6-common php5.6-json
  php5.6-opcache php5.6-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  debsuryorg-archive-keyring libapache2-mod-php5.6 libpcre3 mysql-client mysql-client-8.0 mysql-client-core-8.0 mysql-common php-common php5.6 php5.6-cli
  php5.6-common php5.6-json php5.6-mysql php5.6-opcache php5.6-readline
0 upgraded, 15 newly installed, 0 to remove and 60 not upgraded.
Need to get 6928 kB of archives.
After this operation, 76.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libpcre3 amd64 2:8.39-15build1 [248 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-core-8.0 amd64 8.0.40-0ubuntu0.24.04.1 [2765 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 mysql-common all 5.8+1.1.0build1 [6746 B]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-8.0 amd64 8.0.40-0ubuntu0.24.04.1 [22.5 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client all 8.0.40-0ubuntu0.24.04.1 [9416 B]
Get:6 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main amd64 debsuryorg-archive-keyring all 2024.02.05+ubuntu24.04.1+deb.sury.org+1 [4332 B]
Get:7 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main amd64 php-common all 2:95+ubuntu24.04.1+deb.sury.org+1 [13.2 kB]
Get:8 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main amd64 php5.6-common amd64 5.6.40-79+ubuntu24.04.1+deb.sury.org+1 [686 kB]
Get:9 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main amd64 php5.6-json amd64 5.6.40-79+ubuntu24.04.1+deb.sury.org+1 [19.0 kB]
Get:10 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main amd64 php5.6-opcache amd64 5.6.40-79+ubuntu24.04.1+deb.sury.org+1 [68.0 kB]
Get:11 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main amd64 php5.6-readline amd64 5.6.40-79+ubuntu24.04.1+deb.sury.org+1 [13.9 kB]
Get:12 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main amd64 php5.6-cli amd64 5.6.40-79+ubuntu24.04.1+deb.sury.org+1 [1386 kB]
Get:13 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main amd64 libapache2-mod-php5.6 amd64 5.6.40-79+ubuntu24.04.1+deb.sury.org+1 [1340 kB]
Get:14 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main amd64 php5.6 all 5.6.40-79+ubuntu24.04.1+deb.sury.org+1 [191 kB]
Get:15 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main amd64 php5.6-mysql amd64 5.6.40-79+ubuntu24.04.1+deb.sury.org+1 [154 kB]
Fetched 6928 kB in 12s (589 kB/s)
Selecting previously unselected package debsuryorg-archive-keyring.
(Reading database ... 68559 files and directories currently installed.)
Preparing to unpack .../00-debsuryorg-archive-keyring_2024.02.05+ubuntu24.04.1+deb.sury.org+1_all.deb ...
Unpacking debsuryorg-archive-keyring (2024.02.05+ubuntu24.04.1+deb.sury.org+1) ...
Selecting previously unselected package libpcre3:amd64.
Preparing to unpack .../01-libpcre3_2%3a8.39-15build1_amd64.deb ...
Unpacking libpcre3:amd64 (2:8.39-15build1) ...
Selecting previously unselected package php-common.
```

```
Creating config file /etc/php/5.6/mods-available/mysqli.ini with new version

Creating config file /etc/php/5.6/mods-available/pdo_mysql.ini with new version

Creating config file /etc/php/5.6/mods-available/mysql.ini with new version
Setting up php5.6-opcache (5.6.40-79+ubuntu24.04.1+deb.sury.org+1) ...

Creating config file /etc/php/5.6/mods-available/opcache.ini with new version
Setting up php5.6-readline (5.6.40-79+ubuntu24.04.1+deb.sury.org+1) ...

Creating config file /etc/php/5.6/mods-available/readline.ini with new version
Setting up php5.6-cli (5.6.40-79+ubuntu24.04.1+deb.sury.org+1) ...
update-alternatives: using /usr/bin/php5.6 to provide /usr/bin/php (php) in auto mode
update-alternatives: using /usr/bin/phar5.6 to provide /usr/bin/phar (phar) in auto mode
update-alternatives: using /usr/bin/phar.phar5.6 to provide /usr/bin/phar.phar (phar.phar) in auto mode

Creating config file /etc/php/5.6/cli/php.ini with new version
Setting up libapache2-mod-php5.6 (5.6.40-79+ubuntu24.04.1+deb.sury.org+1) ...

Creating config file /etc/php/5.6/apache2/php.ini with new version
Module mpm_event disabled.
Enabling module mpm_prefork.
apache2_switch_mpm Switch to prefork
apache2_invoke: Enable module php5.6
Setting up php5.6 (5.6.40-79+ubuntu24.04.1+deb.sury.org+1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Processing triggers for php5.6-cli (5.6.40-79+ubuntu24.04.1+deb.sury.org+1) ...
Processing triggers for libapache2-mod-php5.6 (5.6.40-79+ubuntu24.04.1+deb.sury.org+1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-82-41:/home/ubuntu#
```
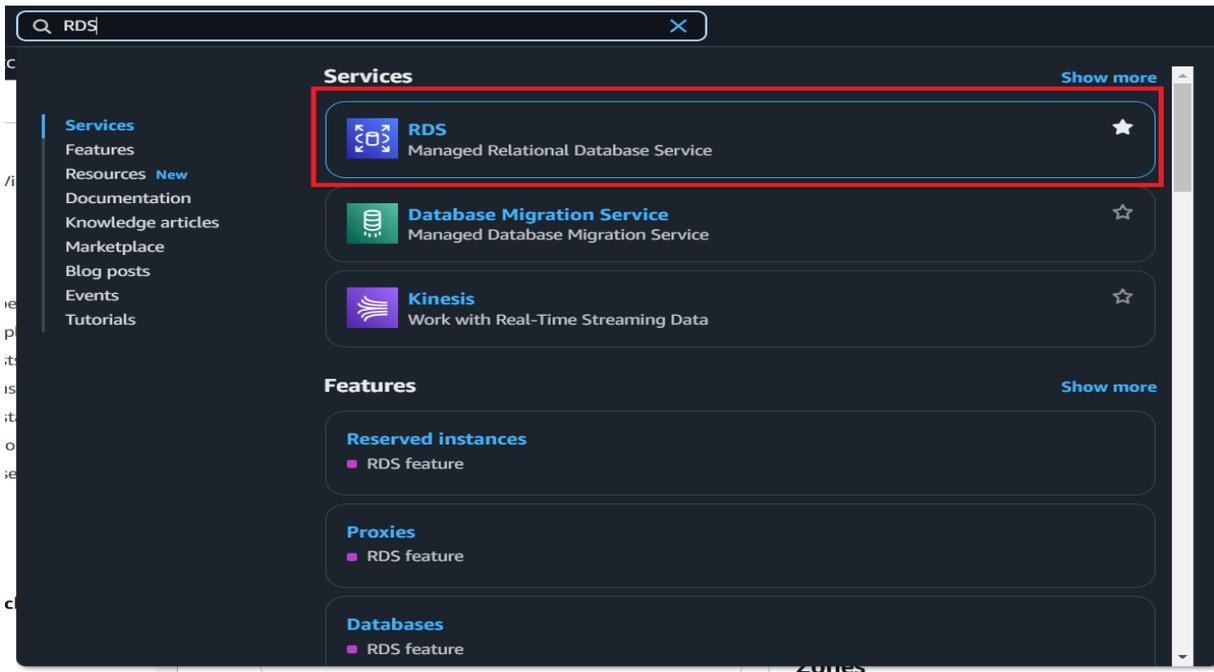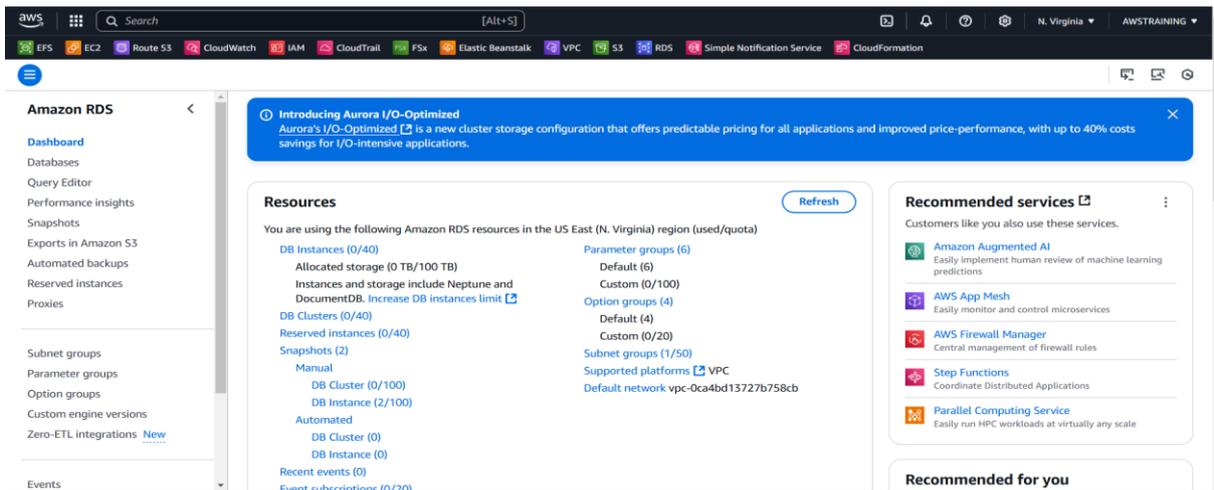
Now we connect mysql with the RDS:

1.Go To AWS Management Console.
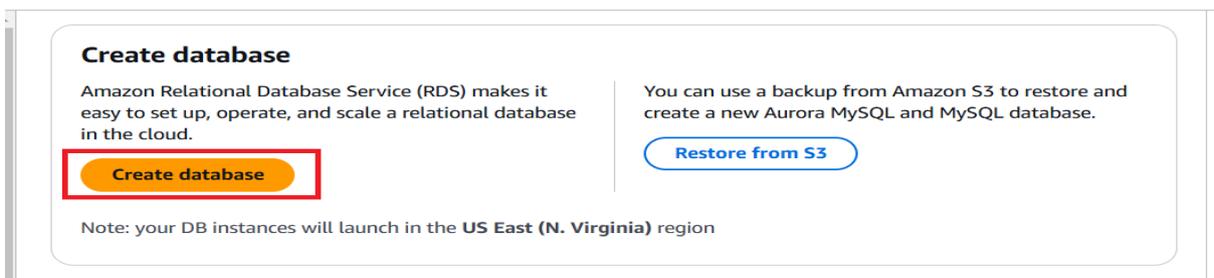
Search RDS

Click the database and create database.



Choose create database.



Select the MySQL Engine and click Next

Select The Templates Dev/Test



Then Availability And Durability Will be Multi-AZ DB Cluster.



Then we need to set credential of db.



Specify DB Details, make sure to choose in DB

## Instance Class

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

**DB instance class** | Info

▼ **Hide filters**

⬤ Show instance classes that support Amazon RDS Optimized Writes  Info
   Amazon RDS Optimized Writes improves write throughput by up to 2x at no
   additional cost.

🔘 Standard classes (includes m classes)
⚪ Memory optimized classes (includes r classes)
⚪ Compute optimized classes (includes c classes)

db.m5d.large (supports Amazon RDS Optimized Writes)
2 vCPUs   8 GiB RAM   Network: Up to 4,750 Mbps   Instance storage: 75 GiB NVMe SSD   ▼

Specify DB Details, make sure to choose only db.m5d.large 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
MobaXterm)

**Connectivity** Info

**Compute resource**

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

🔘 **Don't connect to an EC2 compute resource**
   Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

⚪ Connect to an EC2 compute resource
   Set up a connection to an EC2 compute resource for this database.

**Virtual private cloud (VPC)** Info
Choose the VPC. The VPC defines the virtual networking environment for this DB cluster.

Default VPC (vpc-0ca4bd13727b758cb)
6 Subnets, 6 Availability Zones   ▼

Only VPCs with a corresponding DB subnet group are listed.

ⓘ After a database is created, you can't change its VPC.

**DB subnet group** Info
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB cluster can use in the VPC that you selected.

default-vpc-0ca4bd13727b758cb
6 Subnets, 6 Availability Zones   ▼

**Public access** Info
⬤ Yes
   RDS assigns a public IP address to the cluster. Amazon EC2 instances and other resources outside of the VPC can connect to your cluster. Resources inside the VPC can also connect to the cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster.

## Select VPC Security group

**VPC security group (firewall)** Info
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

🔘 **Choose existing**
   Choose existing VPC security groups

⚪ Create new
   Create new VPC security group

**Existing VPC security groups**

Choose one or more options   ▼

default ✕

Database Authentication will be default as password authentication.



Then click on Create database.



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)



In few minutes, you will be able to see the Endpoint and Port.

Also, make sure to change some security configuration in the RDS

Go to your EC2 Instance Security Groups and select your group ID
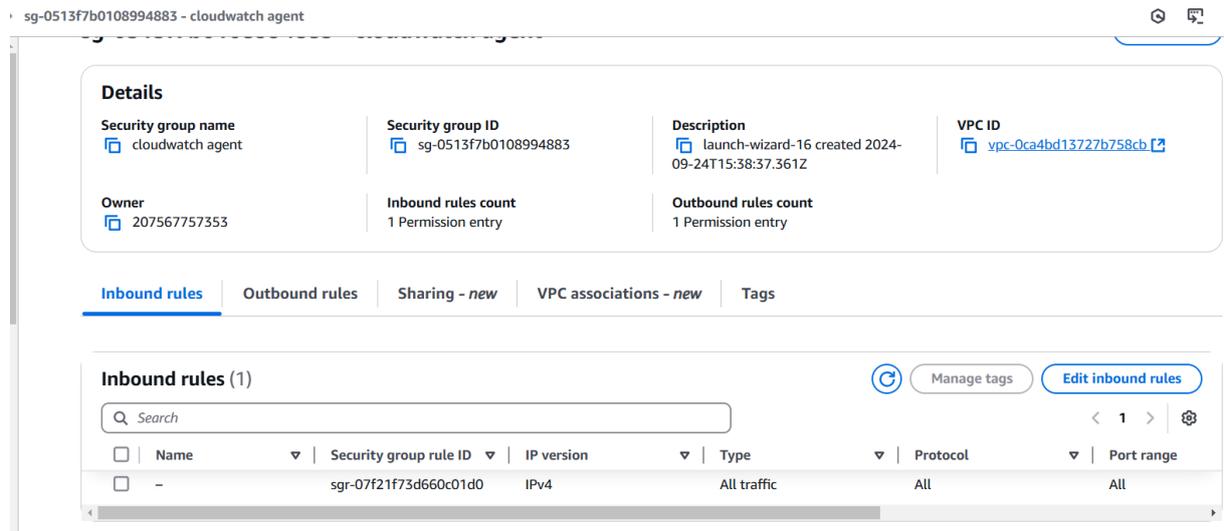


Now go back to your mobaxterm and use this command as shown below

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

Username which you created Here, we're using our own Endpoint and username and password used

I used third party software is mysql workbench.
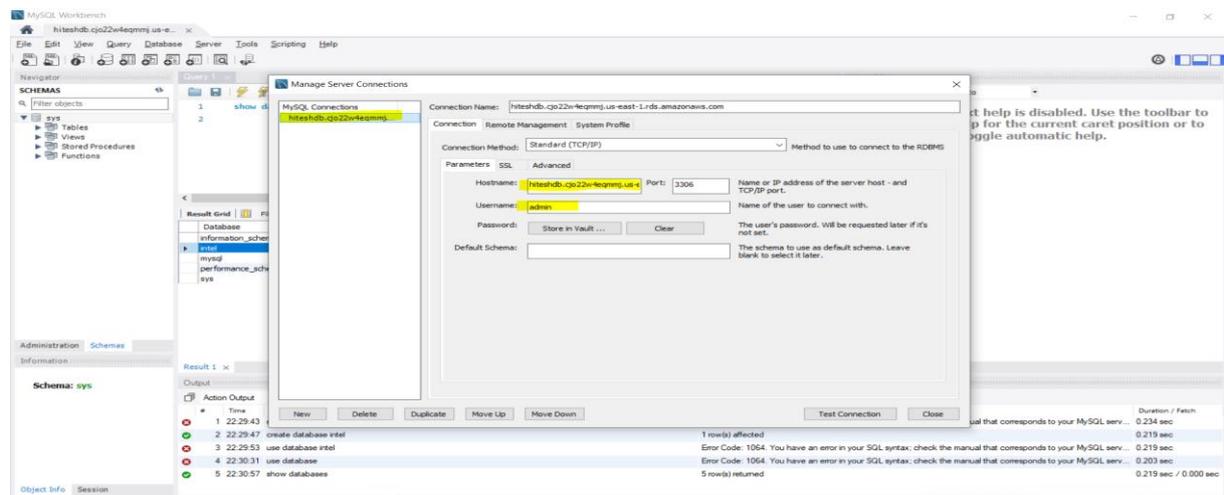


Now go to the path where website files are kept and run the index.php file by using sudo nano index.php.

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

```
            <butto
</form></td>
  <td colspan="4"></td>
</tr>
</table>
</div>
</div>
<?php
$firstname=$_POST['firstname'];
$email=$_POST['email'];
$servername = "intellipaatdb.cqvpjg4mk8sa.us-east-2.rds.amazonaws.com";
$username = "intellipaat";
$password = "intell23";
$db = "intel";
// Create connection
$conn = new mysqli($servername, $username, $password, $db);

// Check connection
```

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

**Instance summary for i-088a174d9f67aff3e** Info

Updated 36 minutes ago

| | | |
|---|---|---|
| **Instance ID**<br>i-088a174d9f67aff3e | **Public IPv4 address**<br>3.231.165.130 \| open address | **Private IPv4 addresses**<br>172.31.12.168 |
| **IPv6 address**<br>– | **Instance state**<br>⊘ Running | **Public IPv4 DNS**<br>ec2-3-231-165-130.compute-1.amazonaws.com \|<br>open address |

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.
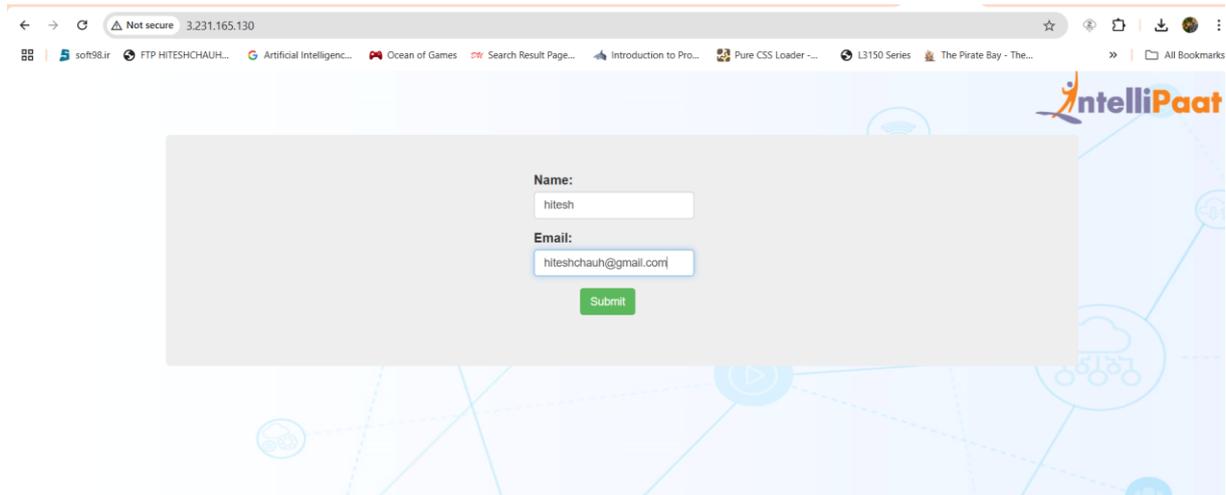


## Auto Scaling:

Now, we'll do the autoscaling of our website by going to our EC2 Instance and then click on Actions and Create Image.
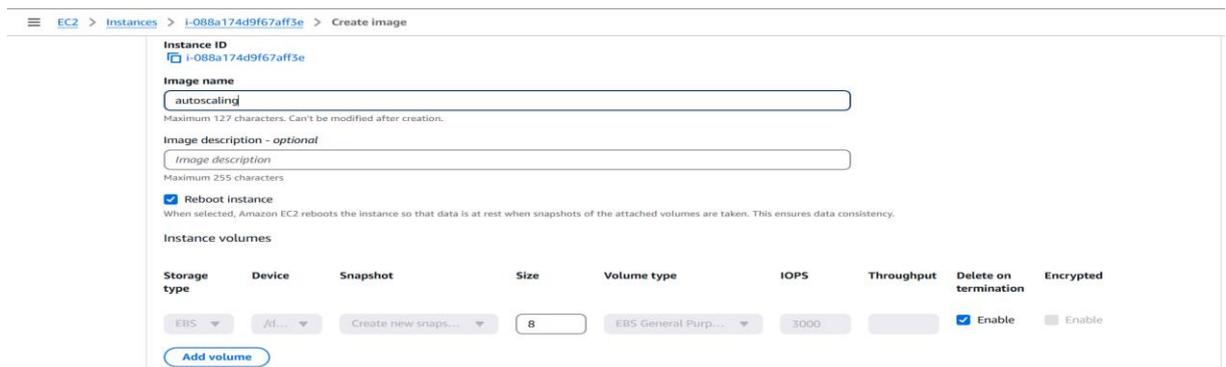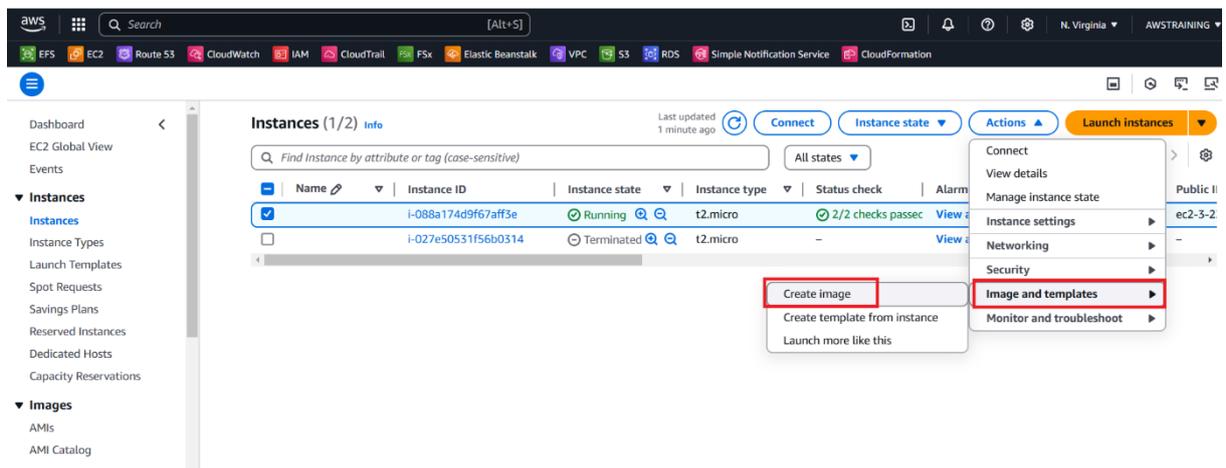
During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

**Tags** - *optional*
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

- ⦿ **Tag image and snapshots together**
  Tag the image and the snapshots with the same tag.

- ○ **Tag image and snapshots separately**
  Tag the image and the snapshots with different tags.

No tags associated with the resource.

**Add new tag**

You can add up to 50 more tags.

Cancel      **Create image**

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

**Snapshots** (1) Info      Recycle Bin    Actions ▼    **Create snapshot**

Owned by me ▼    🔍 Search                                                    ‹ 1 ›  ⚙

| Snapshot ID ▽ | Volume size ▽ | Description ▽ | Storage tier ▽ | Snapshot status ▽ | Started ▽ | Progress |
|---|---|---|---|---|---|---|
| snap-01452d5ba3260721d | 8 GiB | Created by CreateImage(i-... | Standard | ⊘ Completed | 2024/12/05 23:02 GMT+5:... | ⊘ Available (100%) |

That's IT.