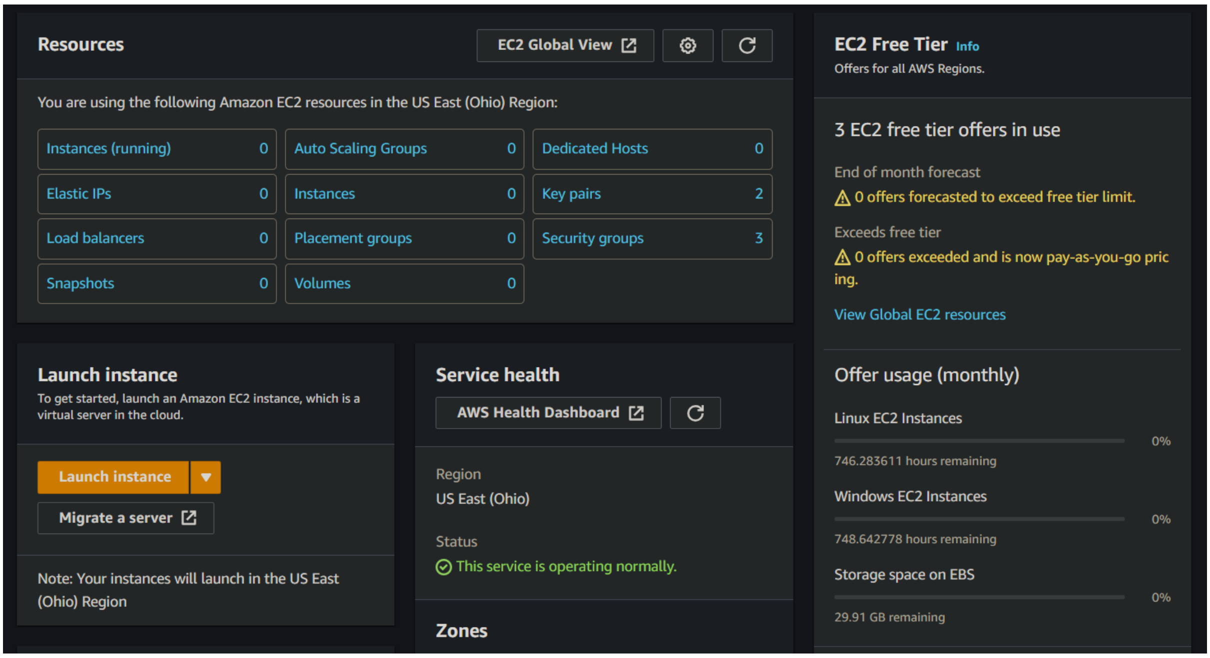
Project Title: - “Scalable MySQL Containerization using Docker on AWS”

Step:-1 Go to AWS Management Console and logged using your credentials

Step:-2 Go to EC2 dashboard and click on launch instance.



Resources

You are using the following Amazon EC2 resources in the US East (Ohio) Region:

Instances (running)	0	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	0	Key pairs	2
Load balancers	0	Placement groups	0	Security groups	3
Snapshots	0	Volumes	0		

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

Service health

Region
US East (Ohio)

Status
This service is operating normally.

Zones

EC2 Free Tier Info

Offers for all AWS Regions.

3 EC2 free tier offers in use

End of month forecast
⚠ 0 offers forecasted to exceed free tier limit.

Exceeds free tier
⚠ 0 offers exceeded and is now pay-as-you-go pricing.

View Global EC2 resources

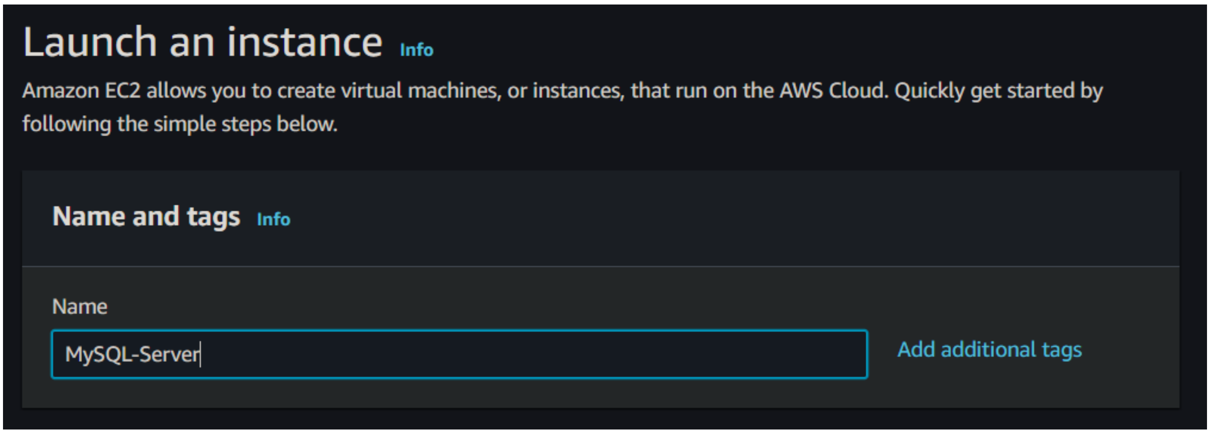
Offer usage (monthly)

Linux EC2 Instances 0%
746.283611 hours remaining

Windows EC2 Instances 0%
748.642778 hours remaining

Storage space on EBS 0%
29.91 GB remaining

Step:-3 Launch an Instance → Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. [Name and tags]



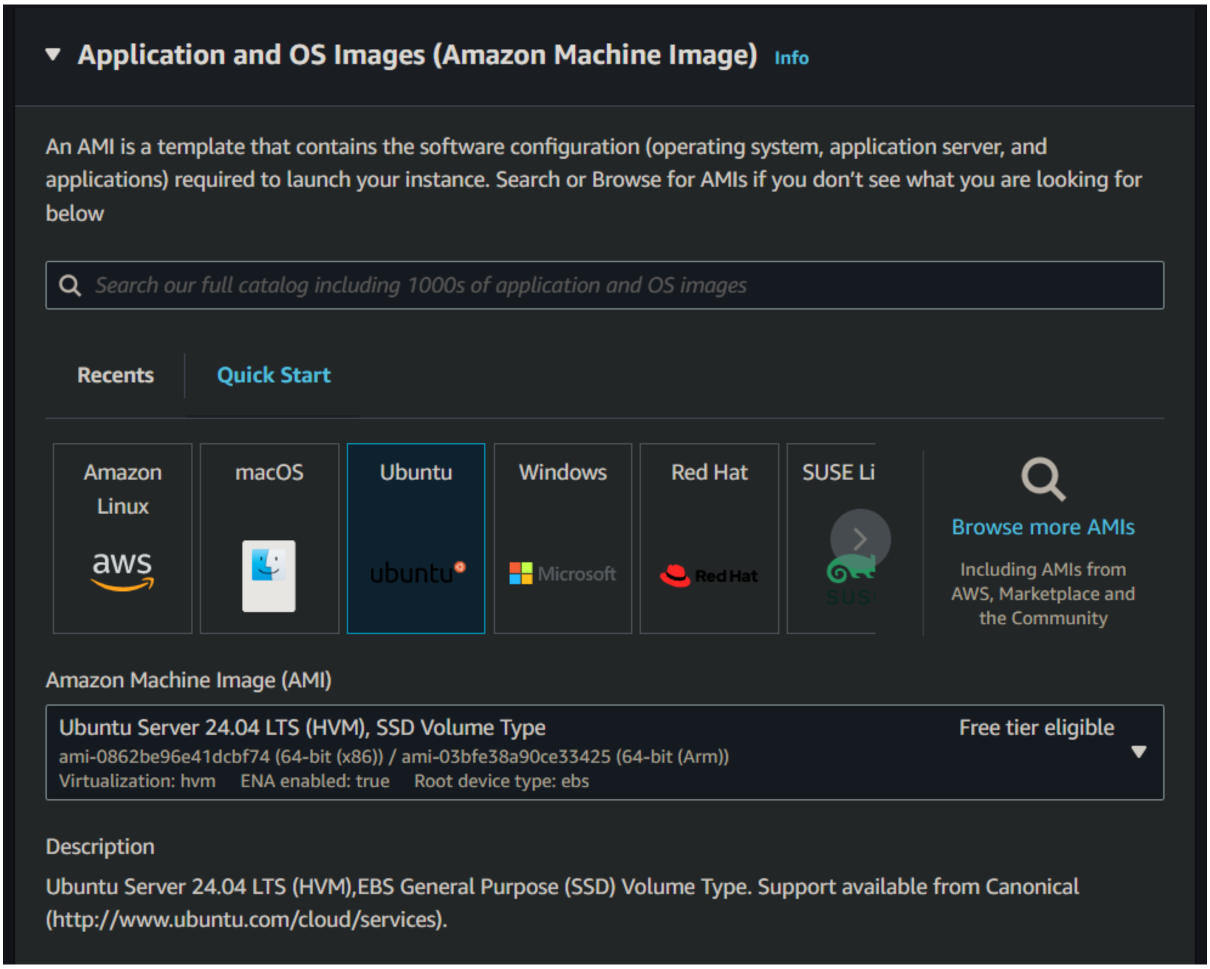
Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: MySQL-Server

Step:-4 Select Application and OS Images [Amazon Machine Image] → An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or browse for AMIs if you don’t see what you are looking for below.



▼ Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

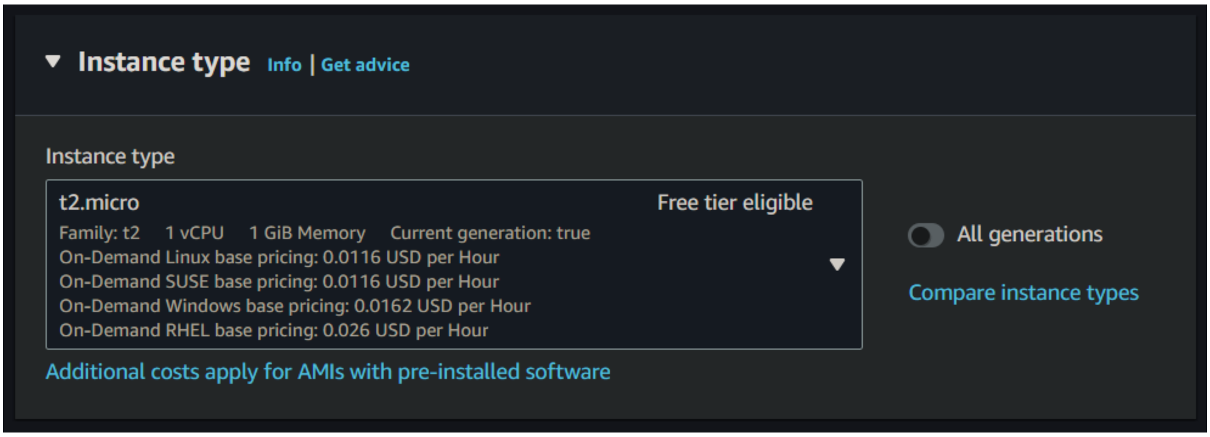
Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-0862be96e41dcbf74 (64-bit (x86)) / ami-03bfe38a90ce33425 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Step:-5 Select Instance Type



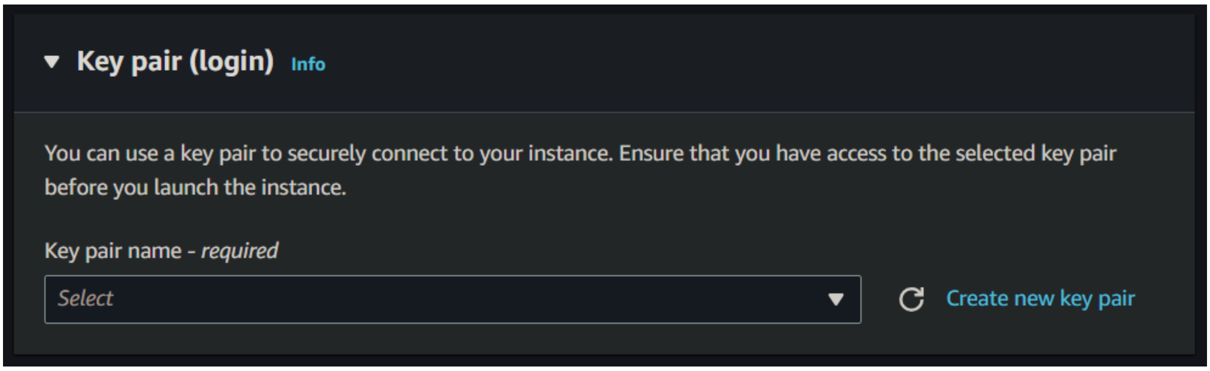
▼ Instance type

Instance type

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0116 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour

Additional costs apply for AMIs with pre-installed software

Step:-6 Key pair (login) [Create new key pair] you can use key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

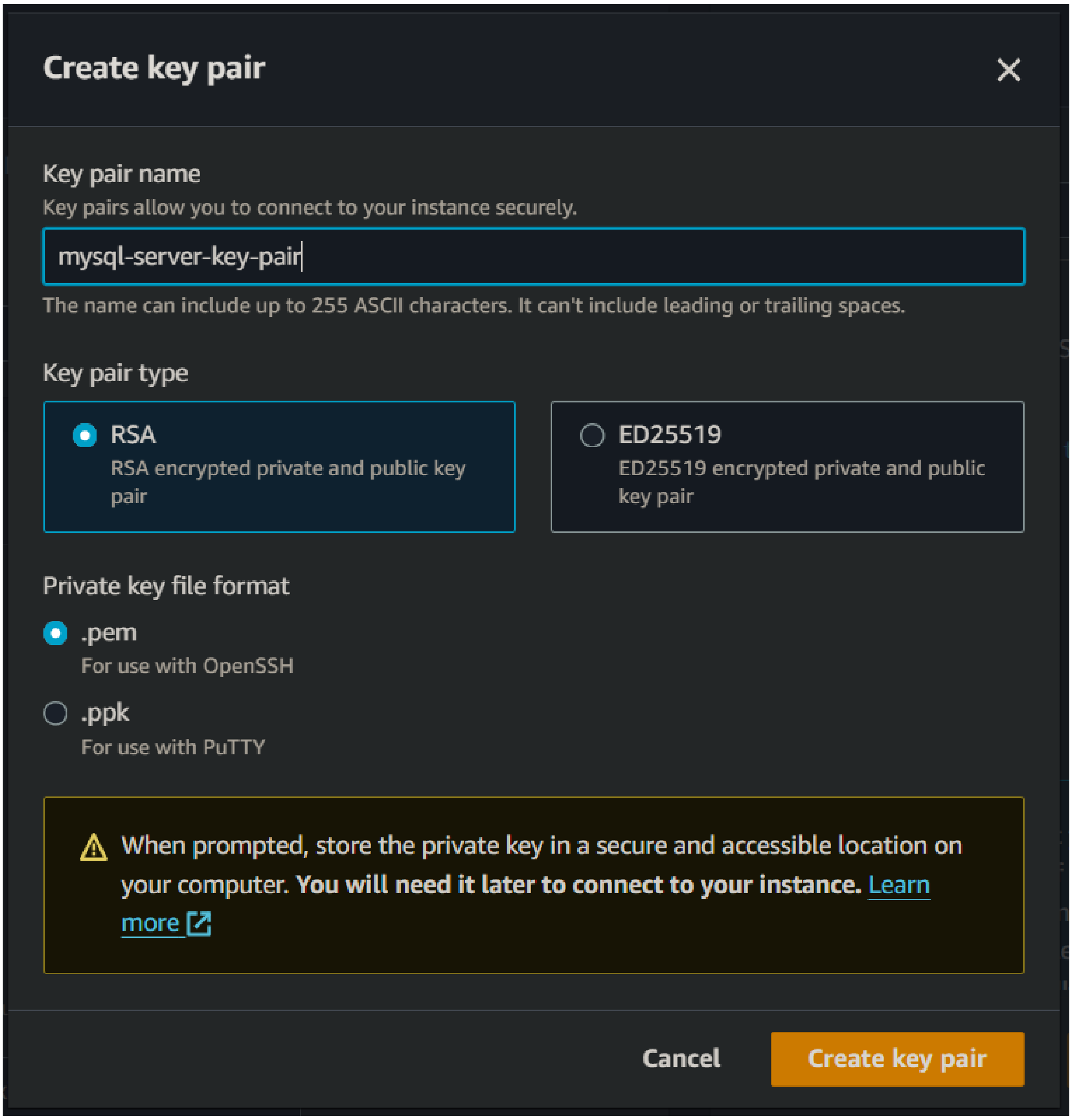


▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Select ▼



Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

mysql-server-key-pair

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

● RSA
RSA encrypted private and public key pair

○ ED25519
ED25519 encrypted private and public key pair

Private key file format

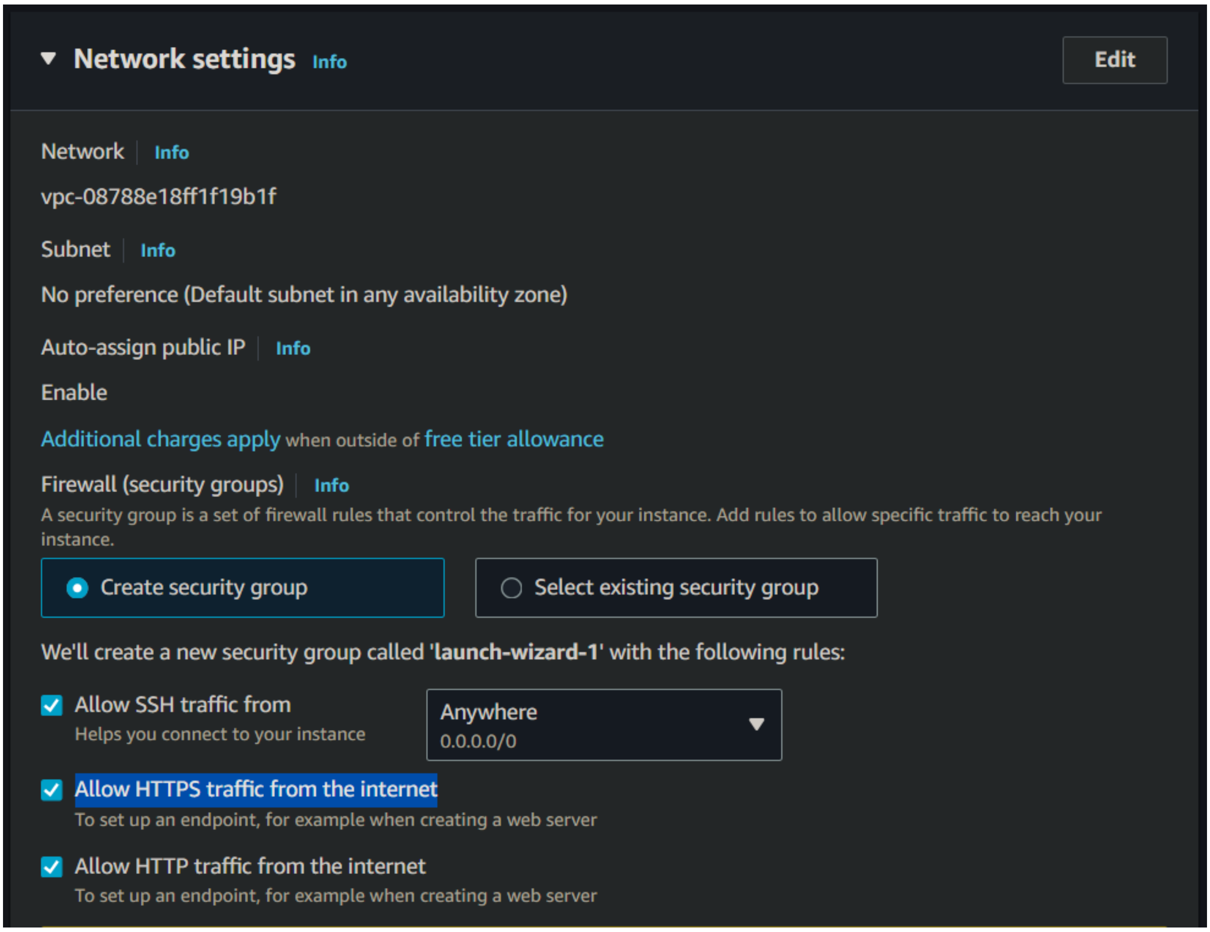
● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** Learn more

Step:-7 Network Settings → Click on edit button → Create new security group name as "**mysql-server-sg**"

1. Allow **SSH** traffic from anywhere
2. Allows **HTTPs** traffic from the internet
3. Allow **HTTP** traffic from the internet



Network | Info

vpc-08788e18ff1f19b1f

Subnet | Info

No preference (Default subnet in any availability zone)

Auto-assign public IP | Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) | Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

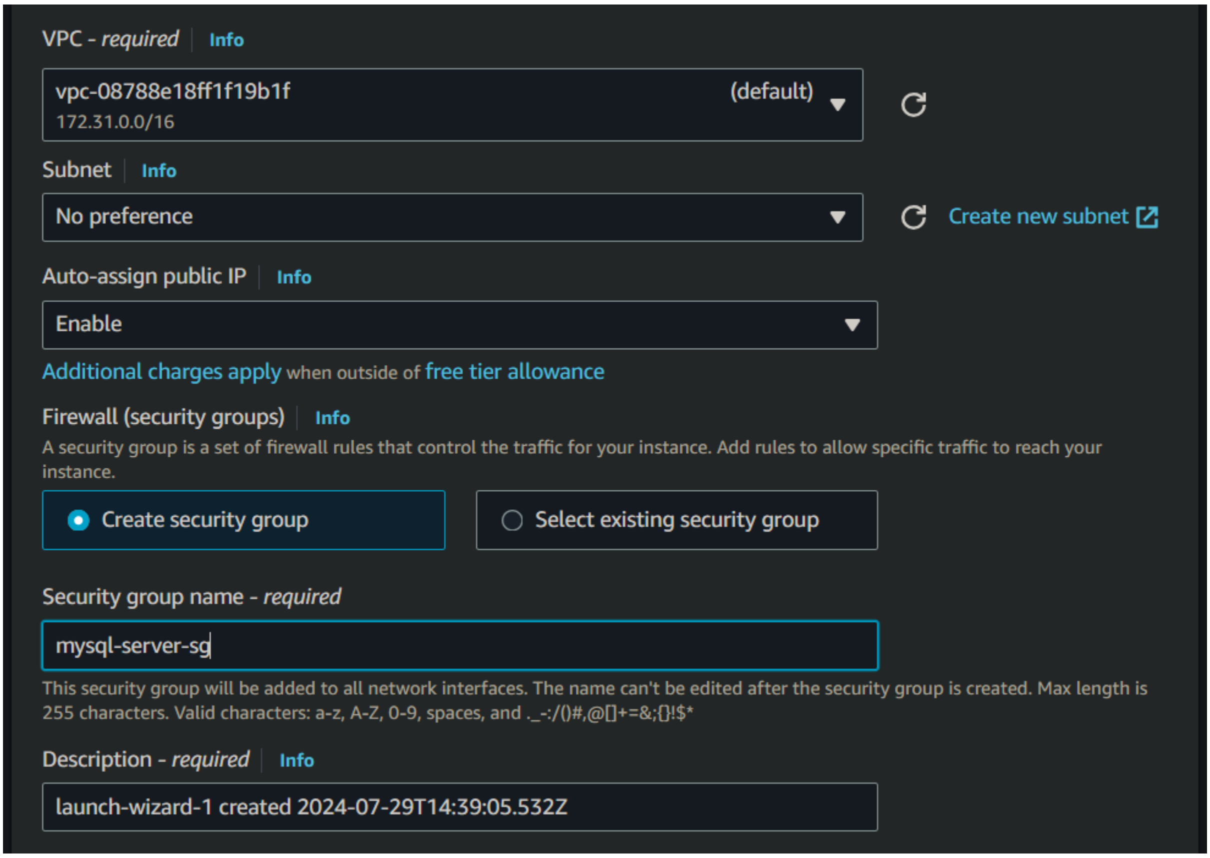
● Create security group ○ Select existing security group

We'll create a new security group called '**launch-wizard-1**' with the following rules:

☑ Allow SSH traffic from — Anywhere 0.0.0.0/0
Helps you connect to your instance

☑ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

☑ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server



VPC - *required* Info

vpc-08788e18ff1f19b1f (default)
172.31.0.0/16

Subnet Info

No preference

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

Security group name - *required*

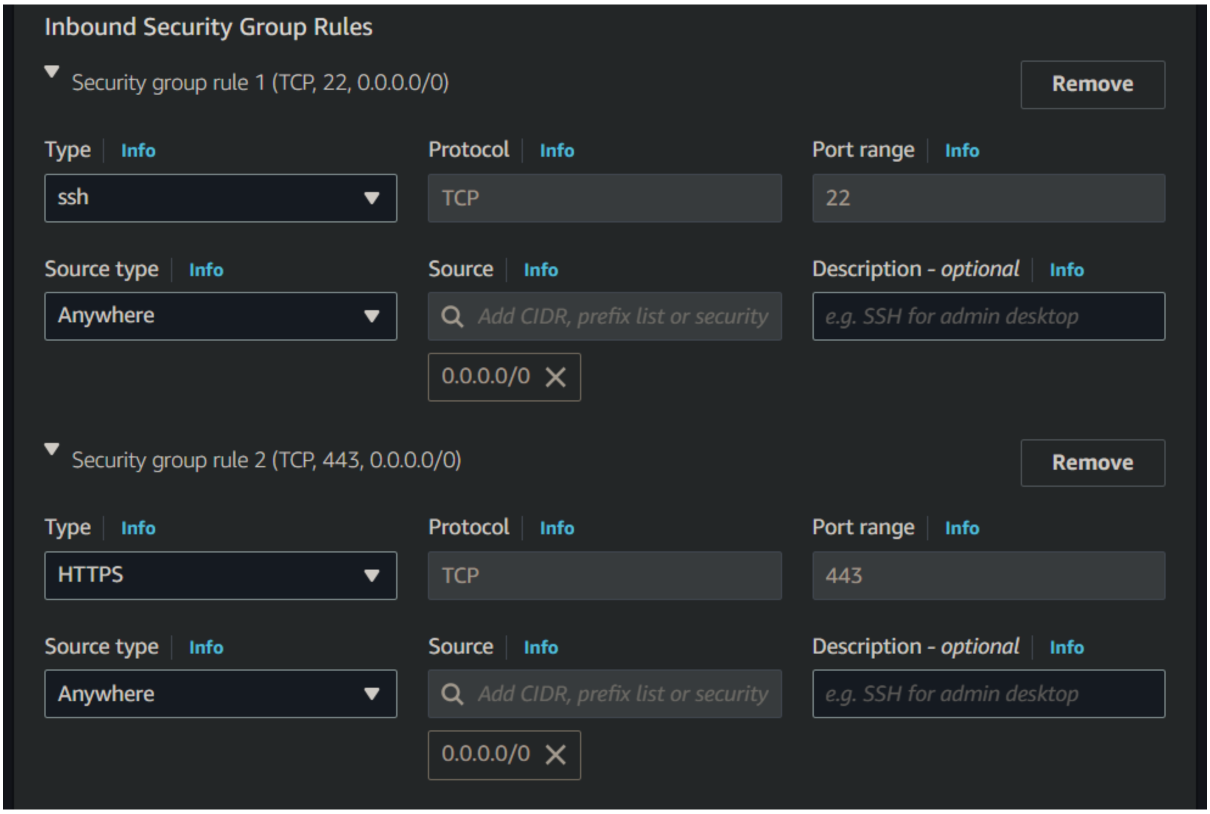
mysql-server-sg

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*

Description - *required* Info

launch-wizard-1 created 2024-07-29T14:39:05.532Z

Allow the following protocol **SSH** to anywhere because to connect to remote server from local machine, also allow **HTTP & HTTPS** to anywhere for inbound traffic.



Inbound Security Group Rules

Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type	Protocol	Port range
ssh	TCP	22

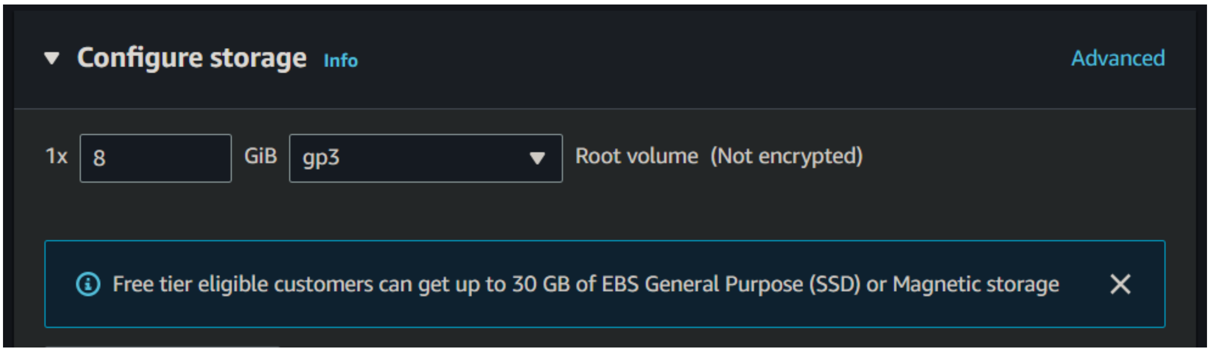
Source type	Source	Description - *optional*
Anywhere	Add CIDR, prefix list or security 0.0.0.0/0	e.g. SSH for admin desktop

Security group rule 2 (TCP, 443, 0.0.0.0/0)

Type	Protocol	Port range
HTTPS	TCP	443

Source type	Source	Description - *optional*
Anywhere	Add CIDR, prefix list or security 0.0.0.0/0	e.g. SSH for admin desktop

Step:-8 Configure storage → free tier eligible customers can get up to 30 GB of EBS (Elastic Block Store) of General Purpose (SSD) or magnetic storage.

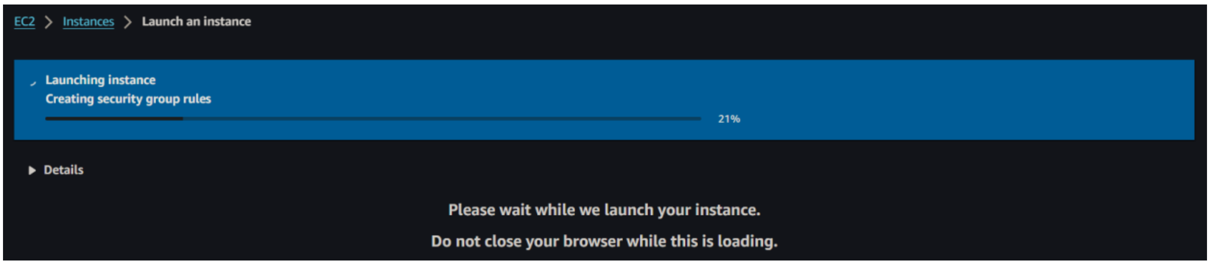


▼ Configure storage Info Advanced

1x 8 GiB gp3 ▼ Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Step:-9 Launch instance



EC2 > Instances > Launch an instance

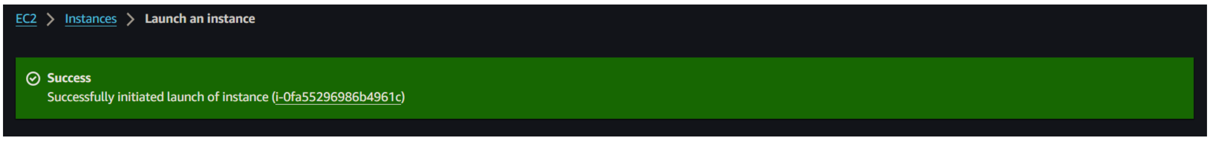
Launching instance
Creating security group rules
21%

▶ Details

Please wait while we launch your instance.

Do not close your browser while this is loading.

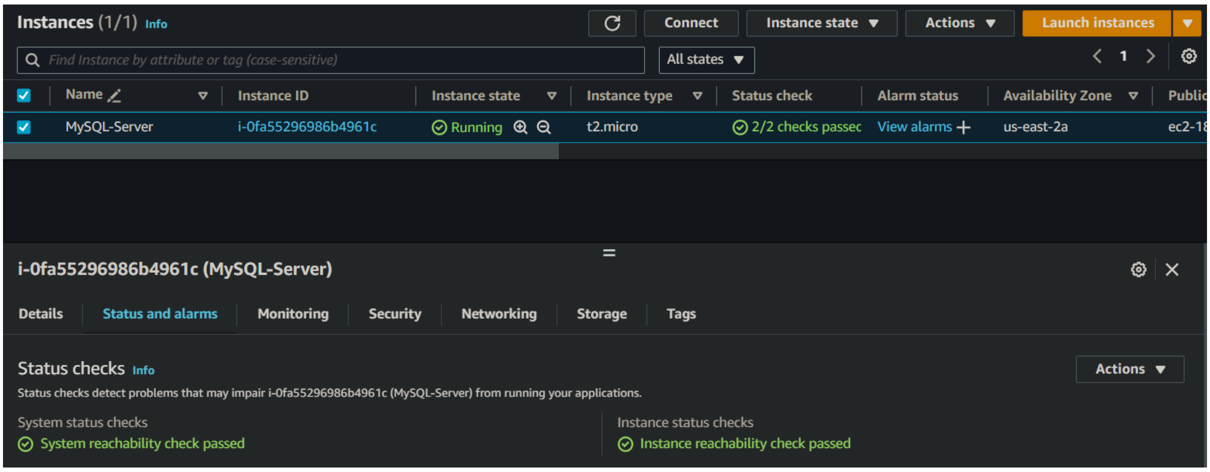
Step:-10 Instance is launch successfully click on instances to go on running instances.



EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-0fa55296986b4961c)

Step:-11 go to instance and click on the server name and check for status check as 2/2 checked passed that is means **system reachability checked passed** and **instance reachability checked passed**. If those are passed then your instance is ready to connect.



Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive)

All states ▼

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public
☑	MySQL-Server	i-0fa55296986b4961c	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-2a	ec2-18

i-0fa55296986b4961c (MySQL-Server)

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

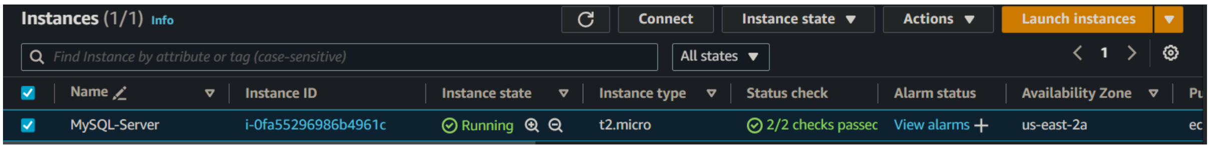
Status checks Info

Status checks detect problems that may impair i-0fa55296986b4961c (MySQL-Server) from running your applications.

System status checks
✓ System reachability check passed

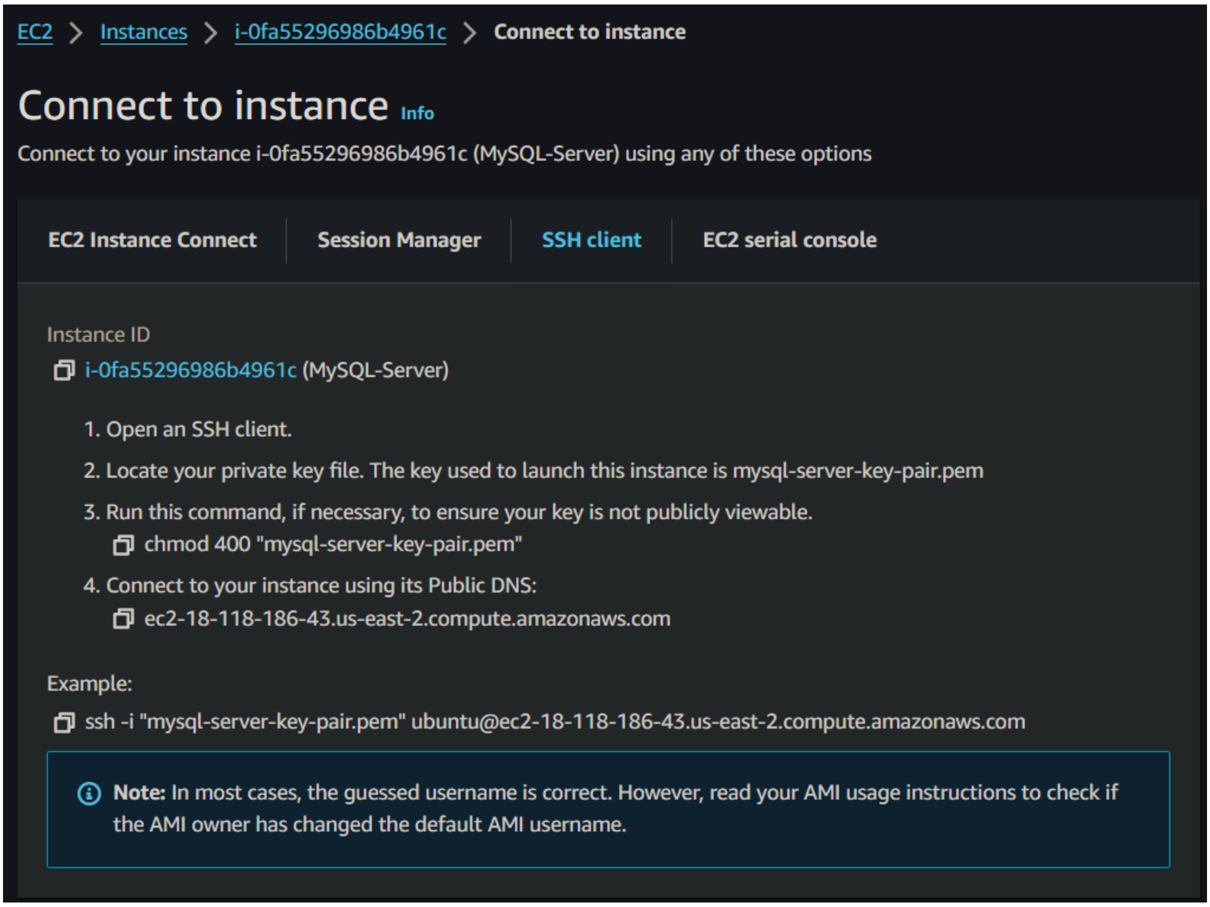
Instance status checks
✓ Instance reachability check passed

Step:-12: click on name i.e. **MySQL-Server** → **Connect** → **SSH Client** → Copy the SSH command shown in the below diagram.



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pu
☑	MySQL-Server	i-0fa55296986b4961c	Running	t2.micro	2/2 checks passec	View alarms +	us-east-2a	ec

[Command: **ssh -i "mysql-server-key-pair.pem" ubuntu@ec2-18-118-186-43.us-east-2.compute.amazonaws.com**]



Connect to instance Info

Connect to your instance i-0fa55296986b4961c (MySQL-Server) using any of these options

Instance ID

i-0fa55296986b4961c (MySQL-Server)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is mysql-server-key-pair.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   `chmod 400 "mysql-server-key-pair.pem"`
4. Connect to your instance using its Public DNS:
   `ec2-18-118-186-43.us-east-2.compute.amazonaws.com`

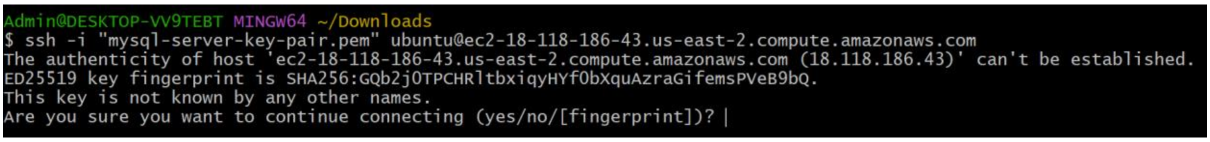
Example:

```
ssh -i "mysql-server-key-pair.pem" ubuntu@ec2-18-118-186-43.us-east-2.compute.amazonaws.com
```

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

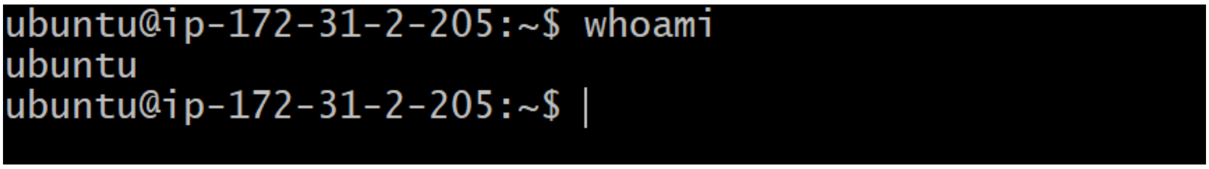
Step:-13 open your terminal from local machine and connect remote machine using SSH [command **"ssh -i "mysql-server-key-pair.pem" ubuntu@ec2-18-118-186-43.us-east-2.compute.amazonaws.com"**], here in my case I am using Git Bash to connect to my remote server from local server.

Change directory to ~/downloads folder where your **"mysql-server-key-pair.pem"** file is located and run here and here run SSH command to connect to remote server **"MySQL-Server"**



```
Admin@DESKTOP-VV9TEBT MINGW64 ~/Downloads
$ ssh -i "mysql-server-key-pair.pem" ubuntu@ec2-18-118-186-43.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-18-118-186-43.us-east-2.compute.amazonaws.com (18.118.186.43)' can't be established.
ED25519 key fingerprint is SHA256:GQb2jOTPCHRltbxiqyHYfObXquAzraGifemsPVeB9bQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? |
```

Step:-14 Verify from which user you are logged in to remote server [type command: **whoami**]



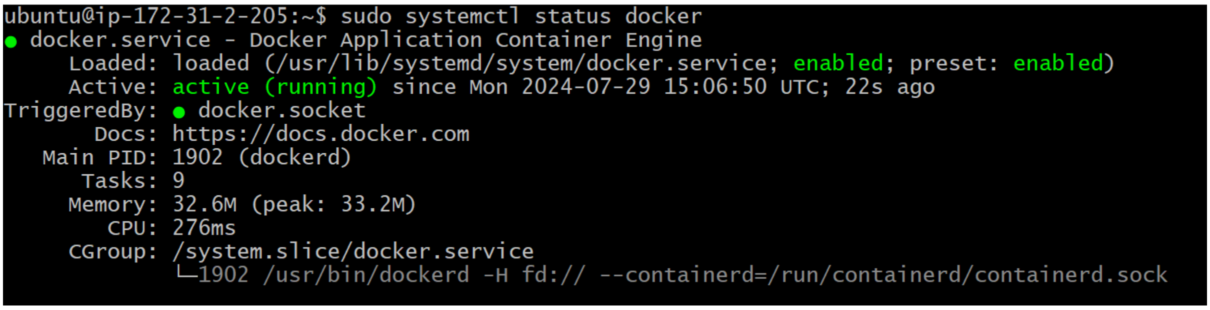
```
ubuntu@ip-172-31-2-205:~$ whoami
ubuntu
ubuntu@ip-172-31-2-205:~$ |
```

Step: - 15 update the package list of **MySQL-Server** using [command: **sudo apt-get update**] after upgradation is completed install Docker on the **MySQL-Server** using [command: **sudo apt-get install docker.io**].

Once installation is done Start and Enable Docker using

[Command: **sudo systemctl status docker**]

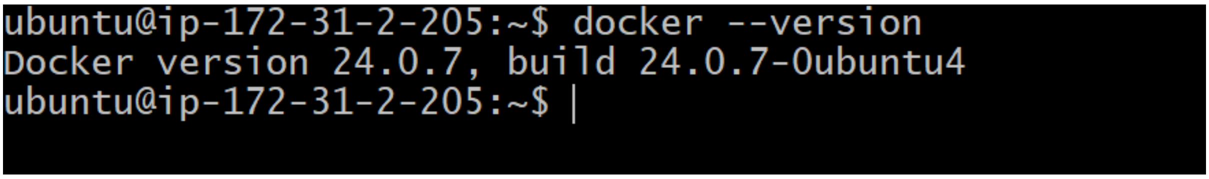
[Command: **sudo systemctl enable docker**]



```
ubuntu@ip-172-31-2-205:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: active (running) since Mon 2024-07-29 15:06:50 UTC; 22s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 1902 (dockerd)
      Tasks: 9
     Memory: 32.6M (peak: 33.2M)
        CPU: 276ms
     CGroup: /system.slice/docker.service
             └─1902 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

Verify Docker installation:

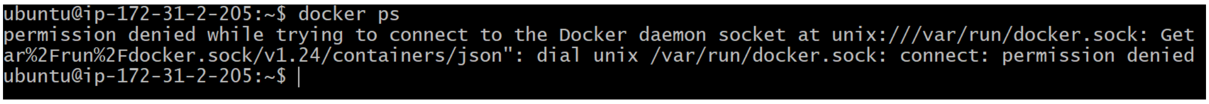
[Command: **docker --version**]



```
ubuntu@ip-172-31-2-205:~$ docker --version
Docker version 24.0.7, build 24.0.7-0ubuntu4
ubuntu@ip-172-31-2-205:~$ |
```

Check and verifies Docker installation run a [command: **docker ps**] to check running containers → permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker/sock.

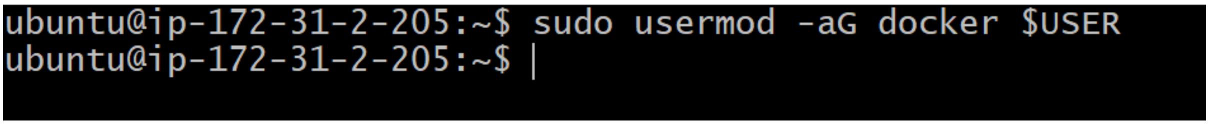
To resolve this error add the current user into the docker group.



```
ubuntu@ip-172-31-2-205:~$ docker ps
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get
ar%2Frun%2Fdocker.sock/v1.24/containers/json": dial unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-2-205:~$ |
```

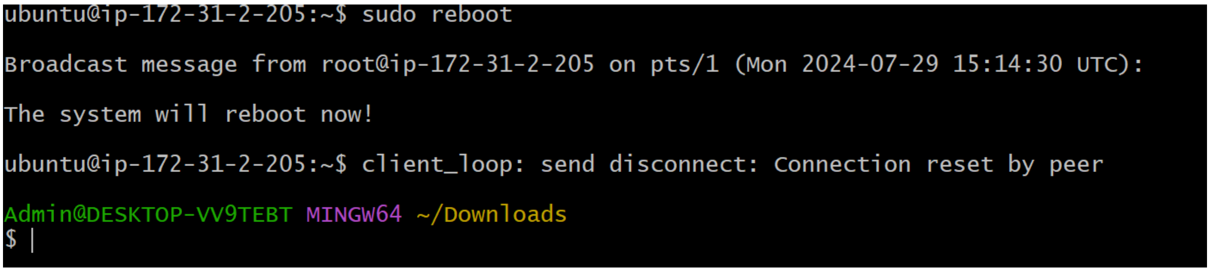
Add current logged in user into the docker group using the [command: **sudo usermod –aG docker $USER**].

Check and verifies is current logged in user Ubuntu is added or not in the docker group [command: **cat /etc/group**].



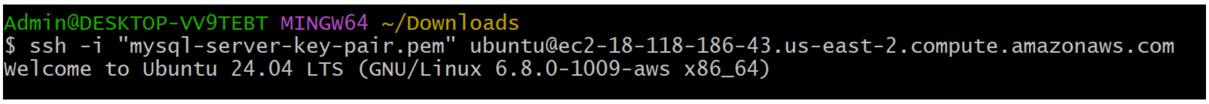
```
ubuntu@ip-172-31-2-205:~$ sudo usermod -aG docker $USER
ubuntu@ip-172-31-2-205:~$ |
```

Check and verifies Docker installation run a [command: **docker ps**] to check running containers → permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker/sock. Again this error came so you need to reboot the system and post reboot SSH into remote server [command: **sudo reboot**].



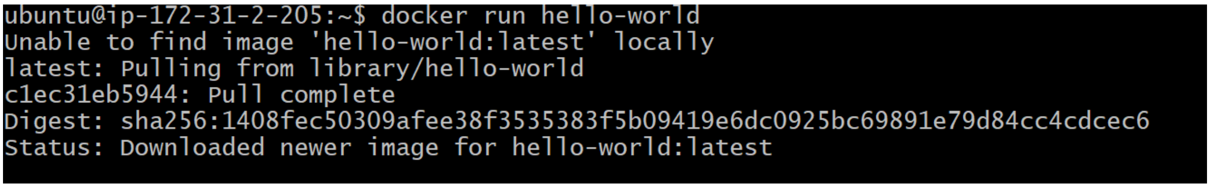
```
ubuntu@ip-172-31-2-205:~$ sudo reboot

Broadcast message from root@ip-172-31-2-205 on pts/1 (Mon 2024-07-29 15:14:30 UTC):

The system will reboot now!

ubuntu@ip-172-31-2-205:~$ client_loop: send disconnect: Connection reset by peer

Admin@DESKTOP-VV9TEBT MINGW64 ~/Downloads
$ |
```

Connect to **MySQL-Sevrer** using a SSH command.



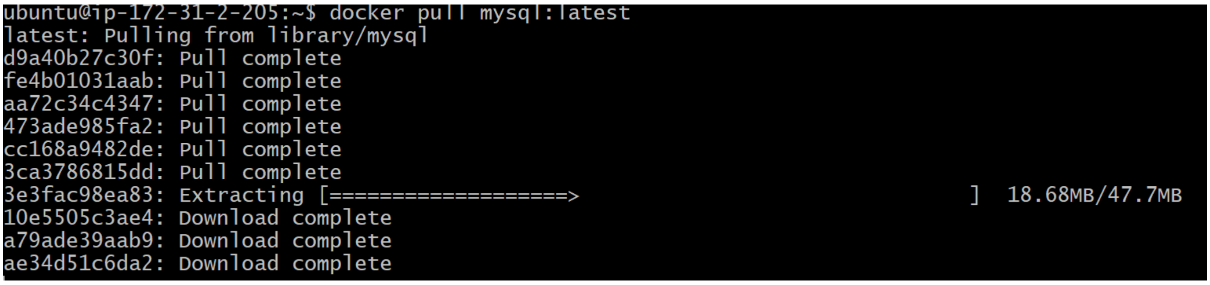
```
Admin@DESKTOP-VV9TEBT MINGW64 ~/Downloads
$ ssh -i "mysql-server-key-pair.pem" ubuntu@ec2-18-118-186-43.us-east-2.compute.amazonaws.com
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1009-aws x86_64)
```

Verify the Docker installation → Note: Dokcer installed successfully.



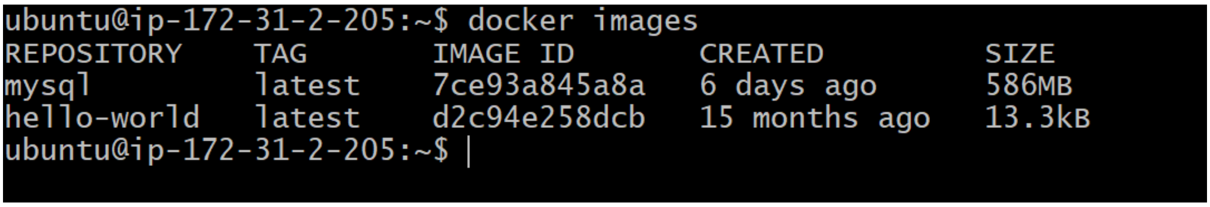
```
ubuntu@ip-172-31-2-205:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:1408fec50309afee38f3535383f5b09419e6dc0925bc69891e79d84cc4cdcec6
Status: Downloaded newer image for hello-world:latest
```

Step:-16 Pull the **MySQL Docker Image from DockerHub** using [command: **sudo docker pull mysql:latest**]



```
ubuntu@ip-172-31-2-205:~$ docker pull mysql:latest
latest: Pulling from library/mysql
d9a40b27c30f: Pull complete
fe4b01031aab: Pull complete
aa72c34c4347: Pull complete
473ade985fa2: Pull complete
cc168a9482de: Pull complete
3ca3786815dd: Pull complete
3e3fac98ea83: Extracting [==================>                                ]  18.68MB/47.7MB
10e5505c3ae4: Download complete
a79ade39aab9: Download complete
ae34d51c6da2: Download complete
```

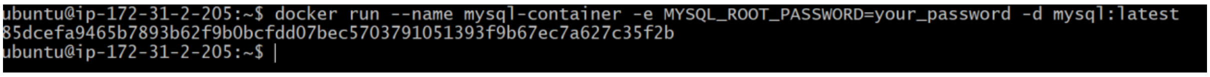
Check and verify if mysql image is downloaded or not using [command: **docker images**]



```
ubuntu@ip-172-31-2-205:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
mysql         latest    7ce93a845a8a   6 days ago      586MB
hello-world   latest    d2c94e258dcb   15 months ago   13.3kB
ubuntu@ip-172-31-2-205:~$ |
```

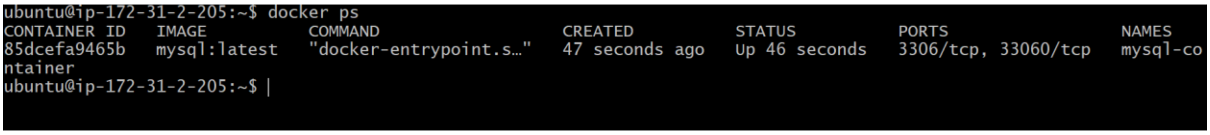
Step:-17 run the MySQL Docker container → Create a container and set the root password (replace **your_password** with a strong password)

[Command: **docker run --name mysql-container -e MYSQL_ROOT_PASSWORD=your_password -d mysql:latest**]



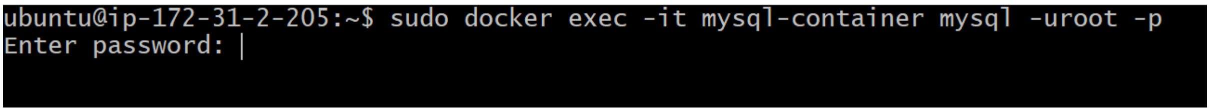
```
ubuntu@ip-172-31-2-205:~$ docker run --name mysql-container -e MYSQL_ROOT_PASSWORD=your_password -d mysql:latest
85dcefa9465b7893b62f9b0bcfdd07bec5703791051393f9b67ec7a627c35f2b
ubuntu@ip-172-31-2-205:~$ |
```

Check and verifies if Docker container is create or not using [command: **docker ps**]

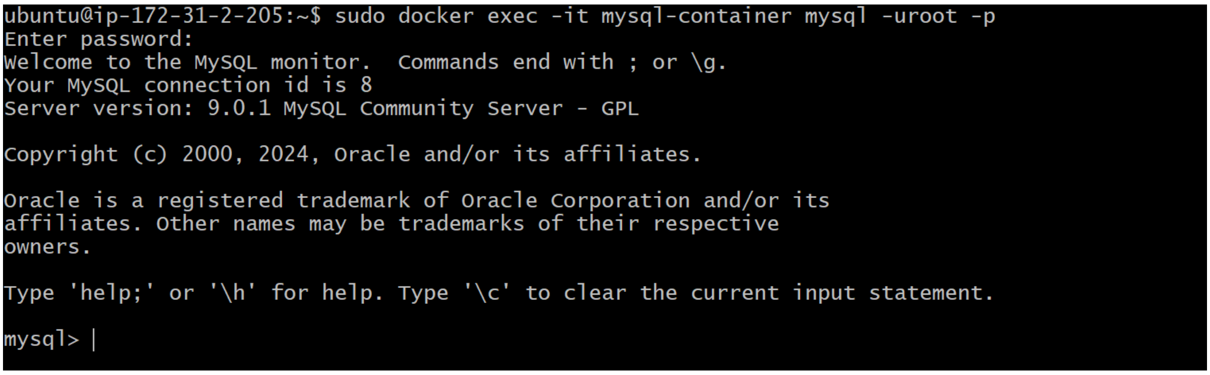


```
ubuntu@ip-172-31-2-205:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS                 NAMES
85dcefa9465b   mysql:latest   "docker-entrypoint.s…"   47 seconds ago   Up 46 seconds   3306/tcp, 33060/tcp   mysql-co
ntainer
ubuntu@ip-172-31-2-205:~$ |
```

Step:-18 access the MySQL container using [**sudo docker exec -it mysql-container mysql -uroot -p**]

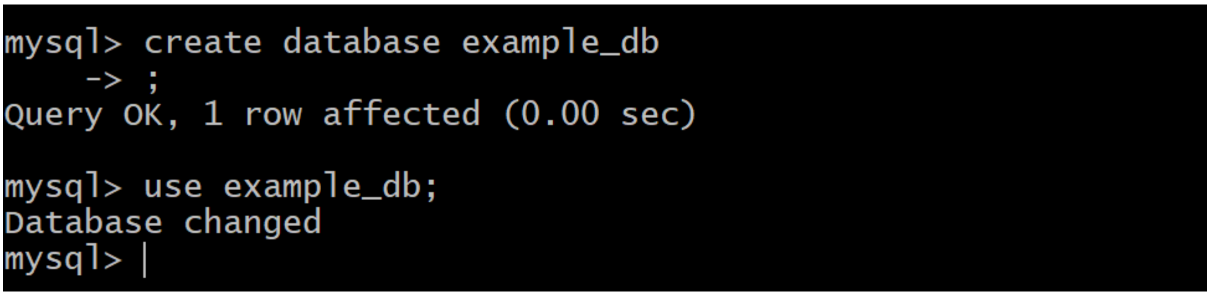


```
ubuntu@ip-172-31-2-205:~$ sudo docker exec -it mysql-container mysql -uroot -p
Enter password: |
```



```
ubuntu@ip-172-31-2-205:~$ sudo docker exec -it mysql-container mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 9.0.1 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

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> |
```

Step:-19 Create a Database and Table, and Perform **CRUD** Operations



```
mysql> create database example_db
    -> ;
Query OK, 1 row affected (0.00 sec)

mysql> use example_db;
Database changed
mysql> |
```

Step:-19 create a table name as **“employee”**

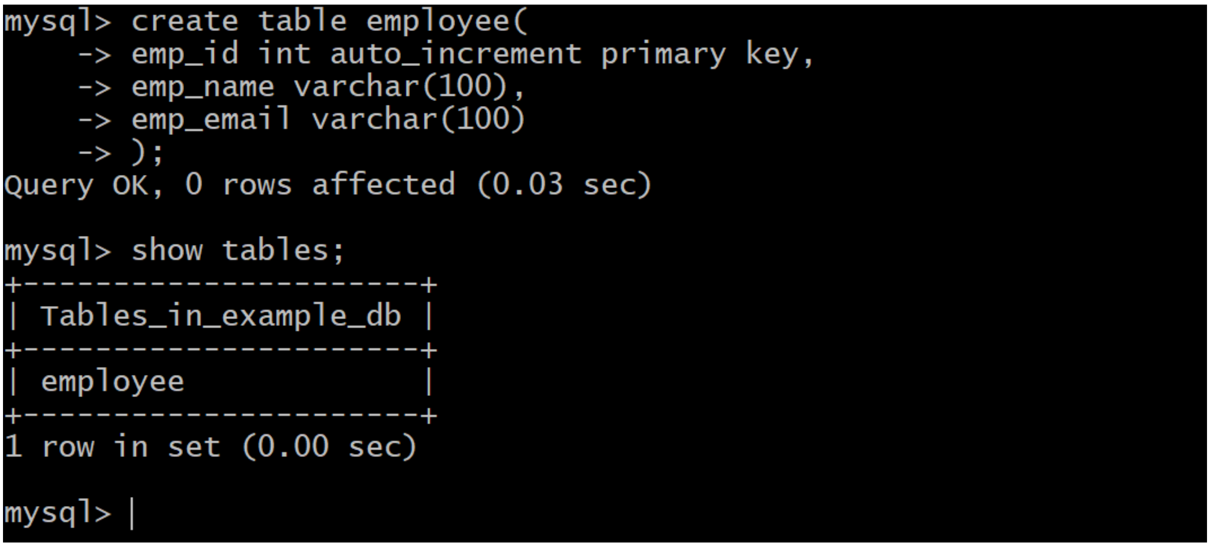
create table employee(

emp_id int auto_increment primary key,

emp_name varchar(100),

emp_email varchar(100)

);



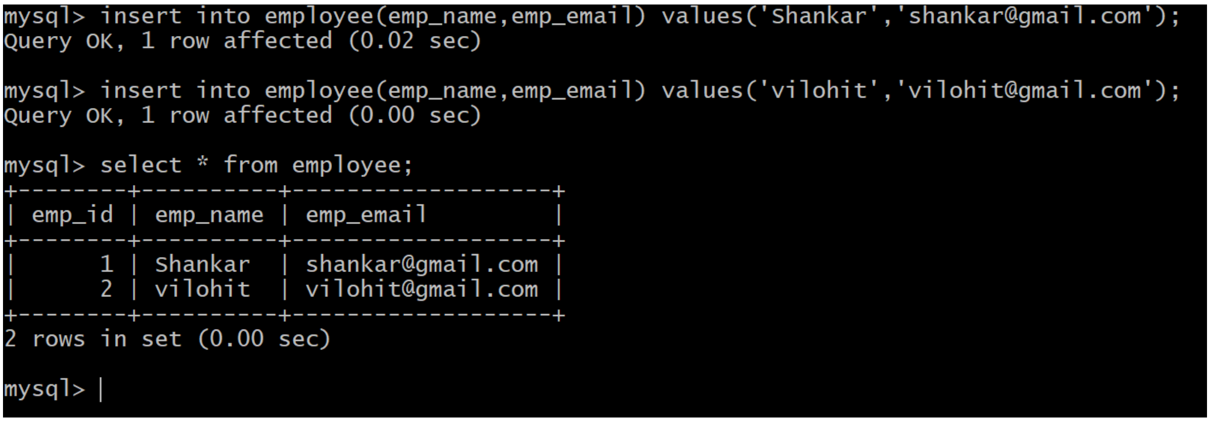
```
mysql> create table employee(
    -> emp_id int auto_increment primary key,
    -> emp_name varchar(100),
    -> emp_email varchar(100)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> show tables;
+----------------------+
| Tables_in_example_db |
+----------------------+
| employee             |
+----------------------+
1 row in set (0.00 sec)

mysql> |
```

Step:-20 Insert data into employee table and select data from employee table.

insert into employee(emp_name,emp_email)
values(‘Shankar’,’shankar@gmail.com’);

insert into employee(emp_name,emp_email)
values(‘vilohit’,’vilohit@gmail.com’);

select * from employee;



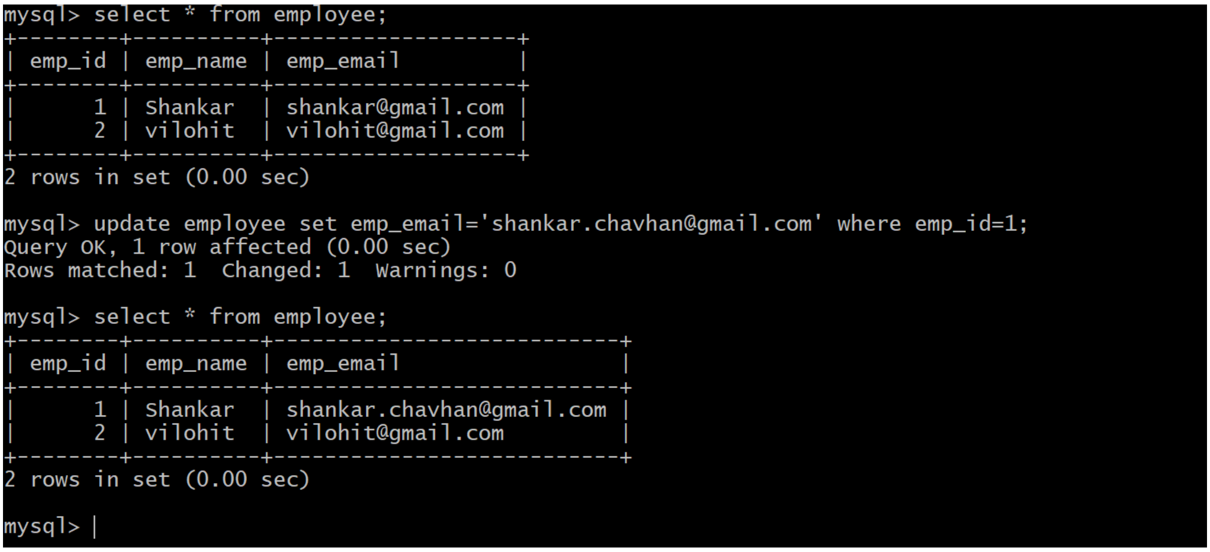
```
mysql> insert into employee(emp_name,emp_email) values('Shankar','shankar@gmail.com');
Query OK, 1 row affected (0.02 sec)

mysql> insert into employee(emp_name,emp_email) values('vilohit','vilohit@gmail.com');
Query OK, 1 row affected (0.00 sec)

mysql> select * from employee;
+--------+----------+-------------------+
| emp_id | emp_name | emp_email         |
+--------+----------+-------------------+
|      1 | Shankar  | shankar@gmail.com |
|      2 | vilohit  | vilohit@gmail.com |
+--------+----------+-------------------+
2 rows in set (0.00 sec)

mysql> |
```

Step:-21 Update data in employee table

select * from employee;

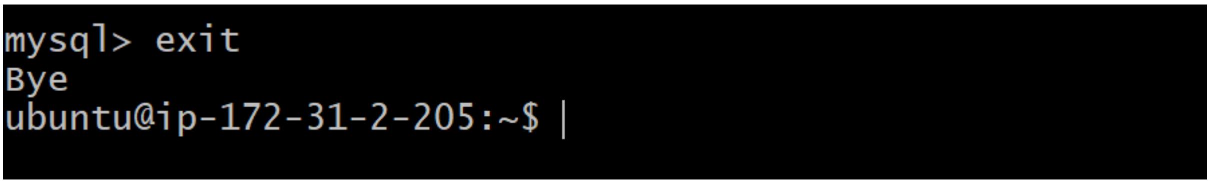
update employee set emp_email='shankar.chavhan@gmail.com' where emp_id=1;

select * from employee;



```
mysql> select * from employee;
+--------+----------+-------------------+
| emp_id | emp_name | emp_email         |
+--------+----------+-------------------+
|      1 | Shankar  | shankar@gmail.com |
|      2 | vilohit  | vilohit@gmail.com |
+--------+----------+-------------------+
2 rows in set (0.00 sec)

mysql> update employee set emp_email='shankar.chavhan@gmail.com' where emp_id=1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from employee;
+--------+----------+---------------------------+
| emp_id | emp_name | emp_email                 |
+--------+----------+---------------------------+
|      1 | Shankar  | shankar.chavhan@gmail.com |
|      2 | vilohit  | vilohit@gmail.com         |
+--------+----------+---------------------------+
2 rows in set (0.00 sec)

mysql> |
```

Note: to clear the mysql shell type (ctrl + d) and to exit from mysql shell type (exit).



```
mysql> exit
Bye
ubuntu@ip-172-31-2-205:~$ |
```

Step:-22 Secure Your Instance and Data

1. **Limit Security Group Access**:
   Restrict access to the MySQL port (3306) to specific IP addresses.

2. **Regular Backups**:
   Consider regular backups of your MySQL data, either using Docker volumes or other backup solutions.

Follow me on **LinkedIn & GitHub** for project updates and insights!

~ Shankar Chavhan