Capstone project - Migrating to cloud

# Details of the project:

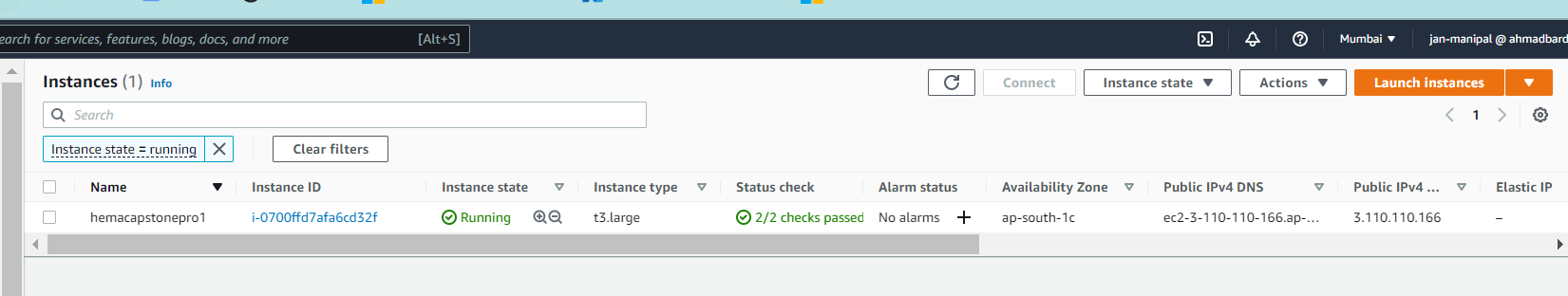
Migrating the data of the database from primary instance that is located in Mumbai region to the secondary region Ohio without losing any data.

Task1:

Deploy sample python webapp on linux server along with mysql as backend for storing sample database in Mumbai region.

# 1.Creating an Ubuntu linux sever:

Created an Ubuntu linux primary server in Mumbai region.

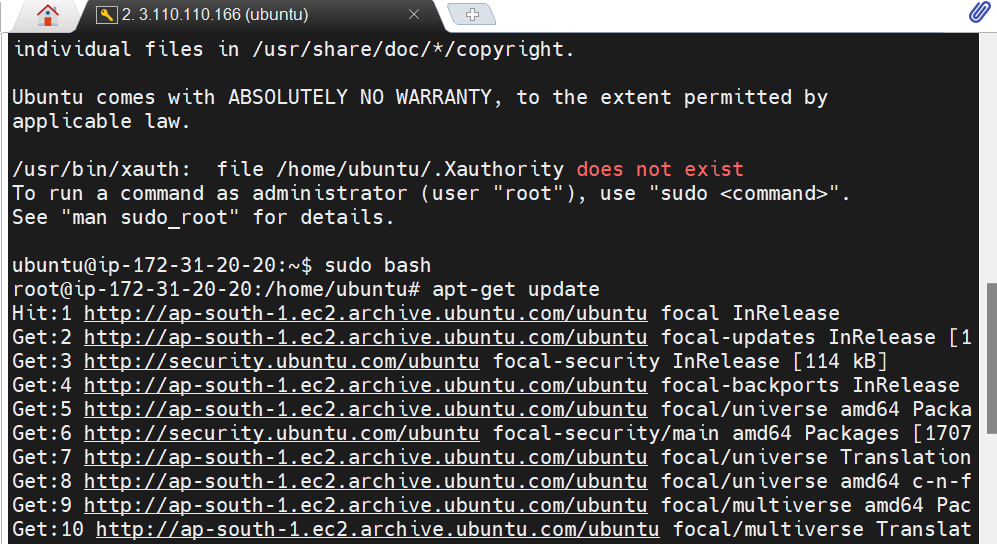


# 2.Install all required dependencies:

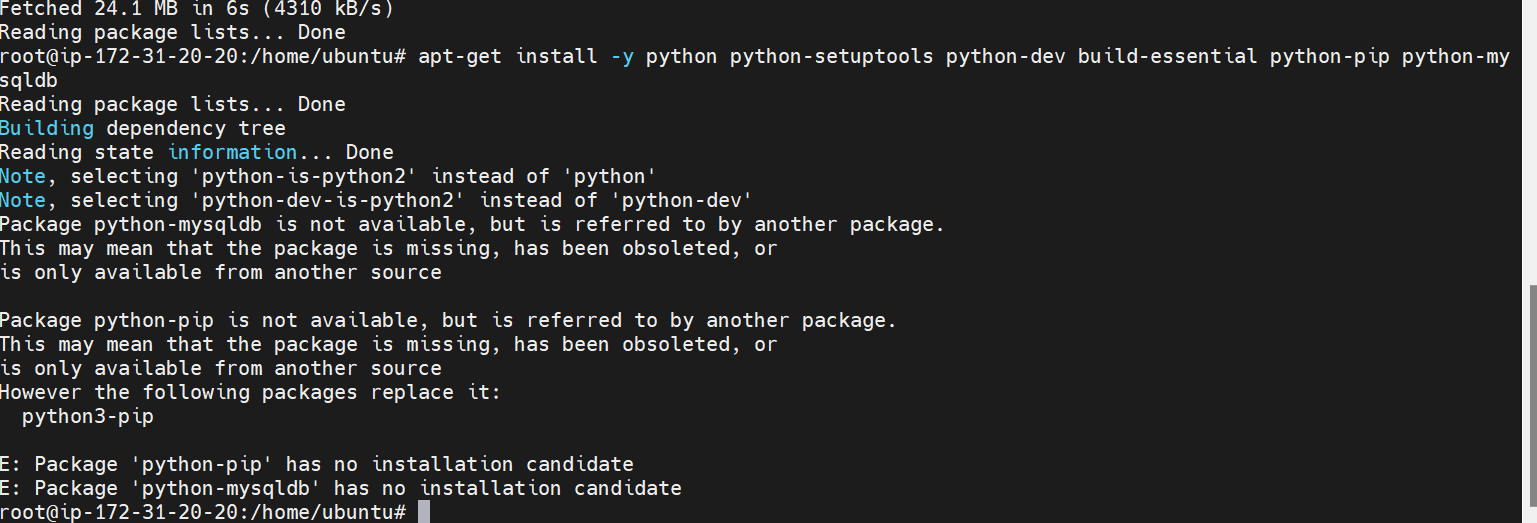
# **Step 1:** Python and its dependencies

# sudo su

# apt-get update

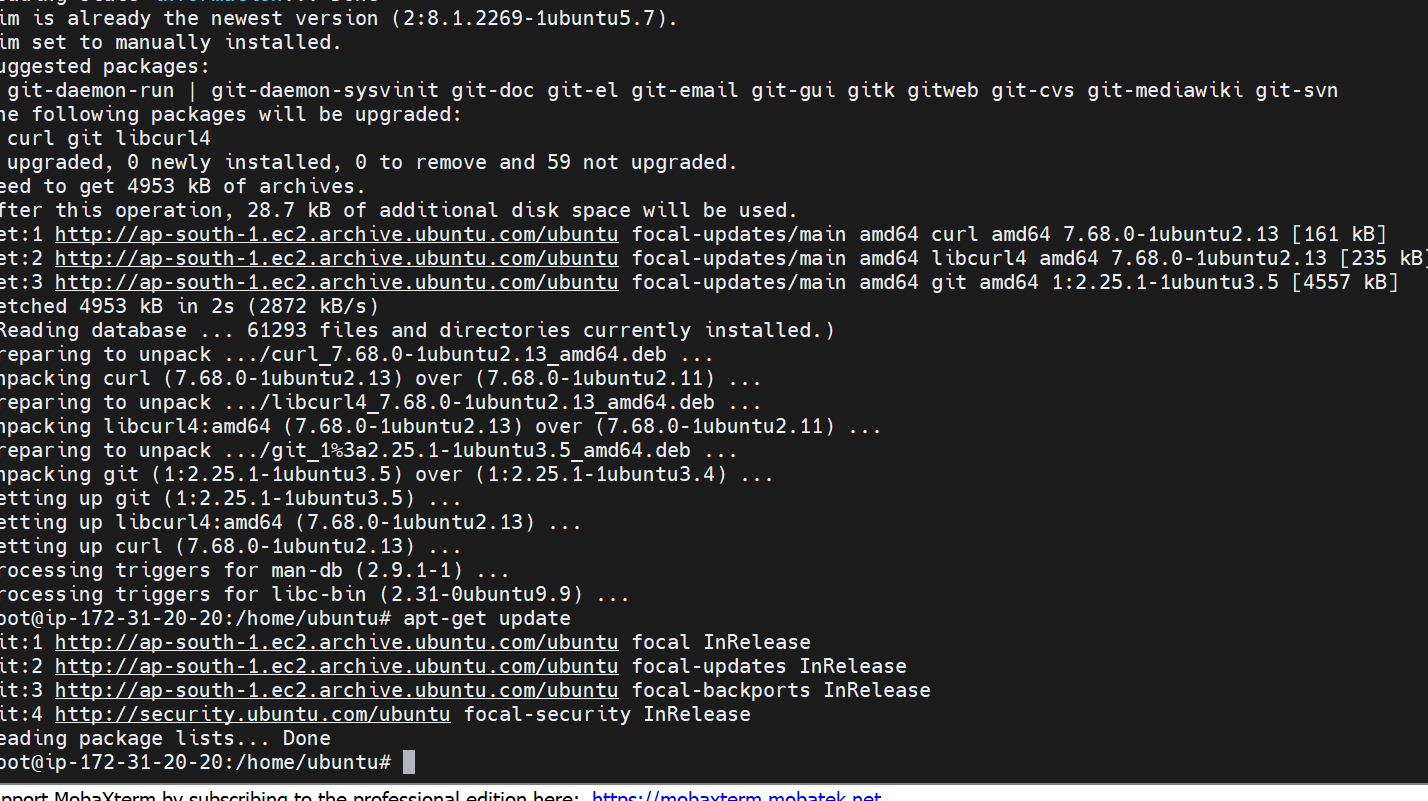


# apt-get install -y python python-setuptools python-dev build-essential python-pip python-mysqldb



**Step 2:** Installing other dependencies

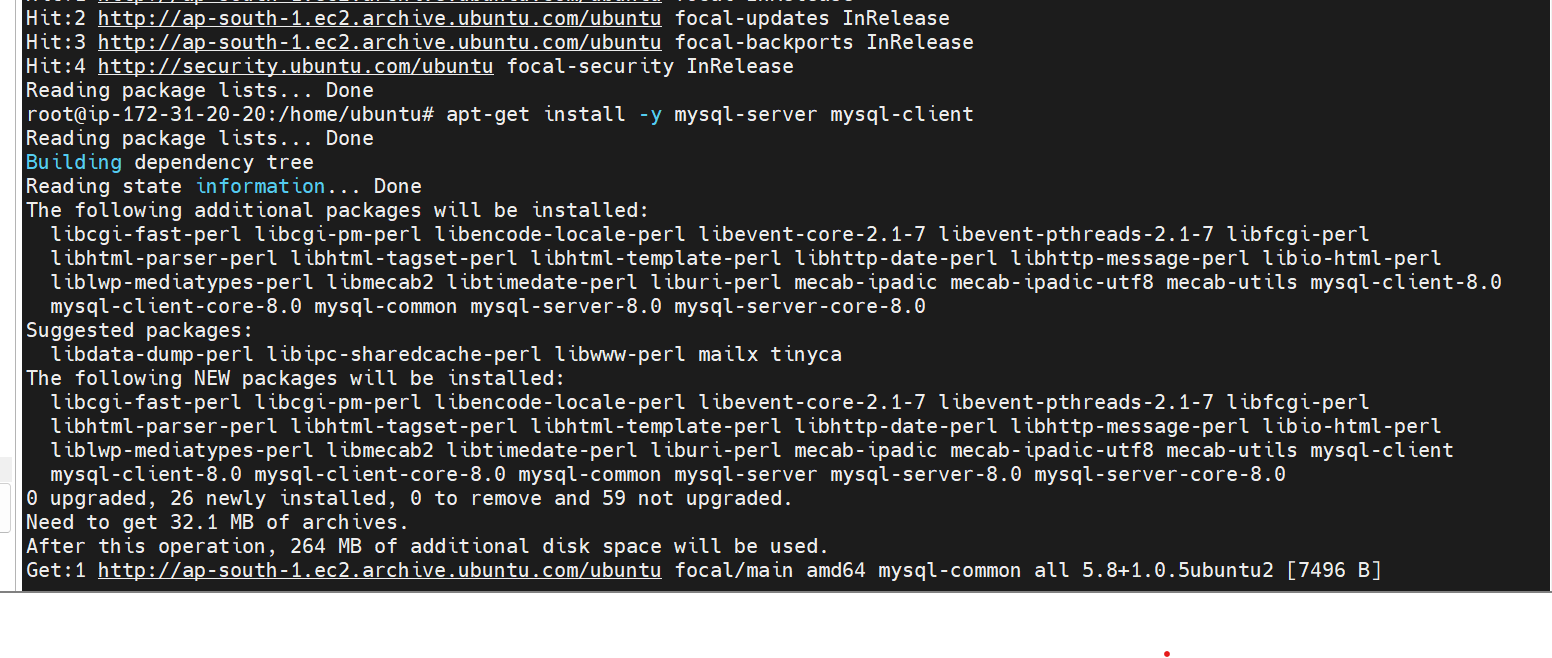
# apt-get install git vim curl -y

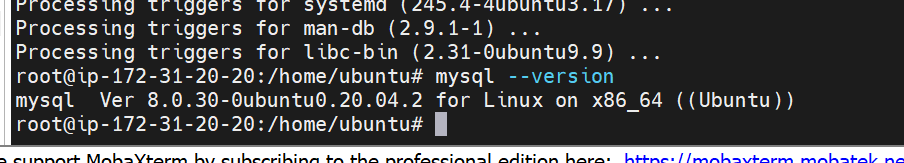


# 3.Install and Configure Database:

**Step 1:** Install MySQL database

# apt-get install -y mysql-server mysql-client



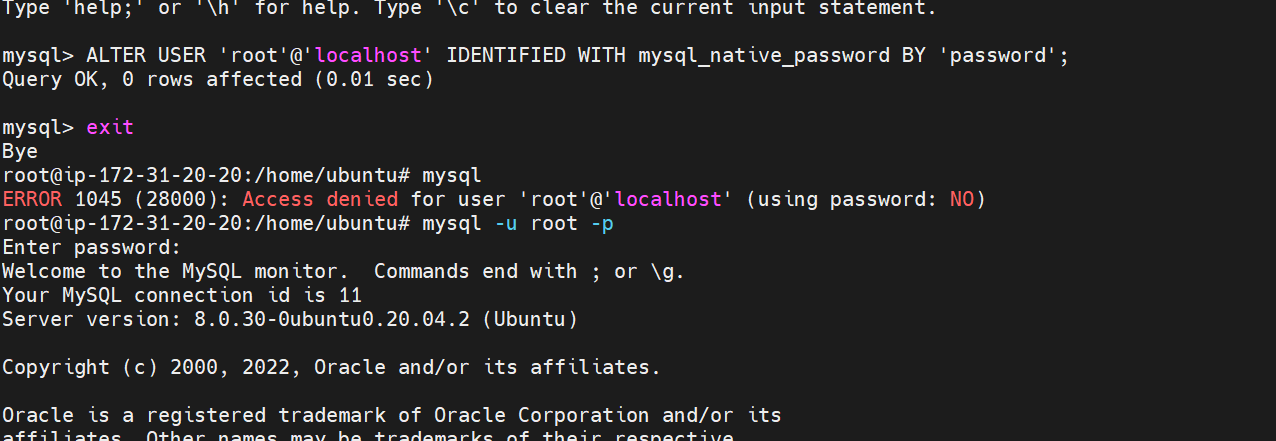


Step2: Adding root access

# Sudo mysql

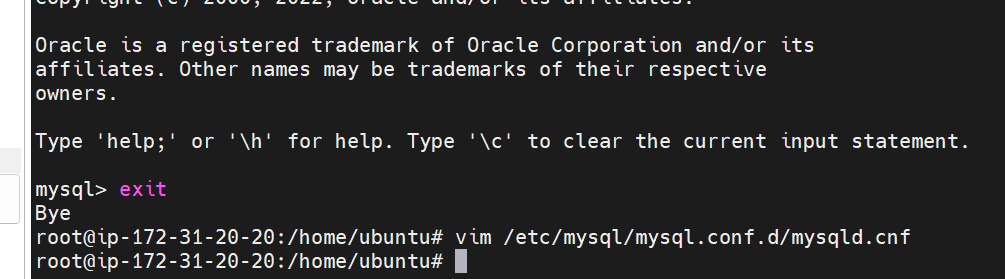
Setting up password to access my sql

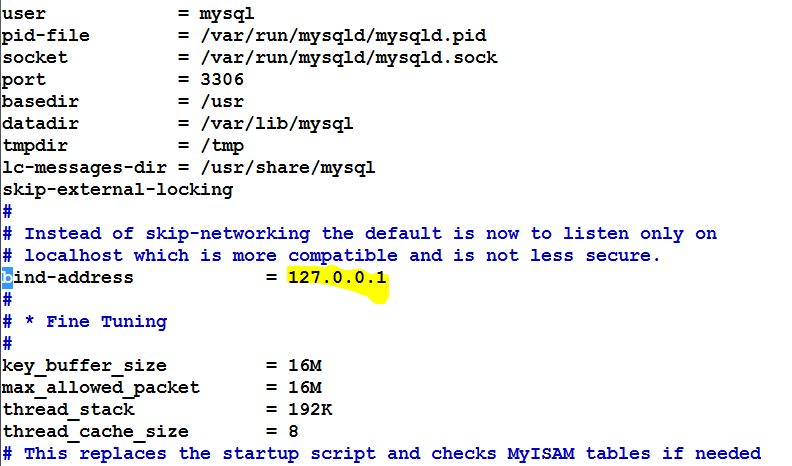
# ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'password';



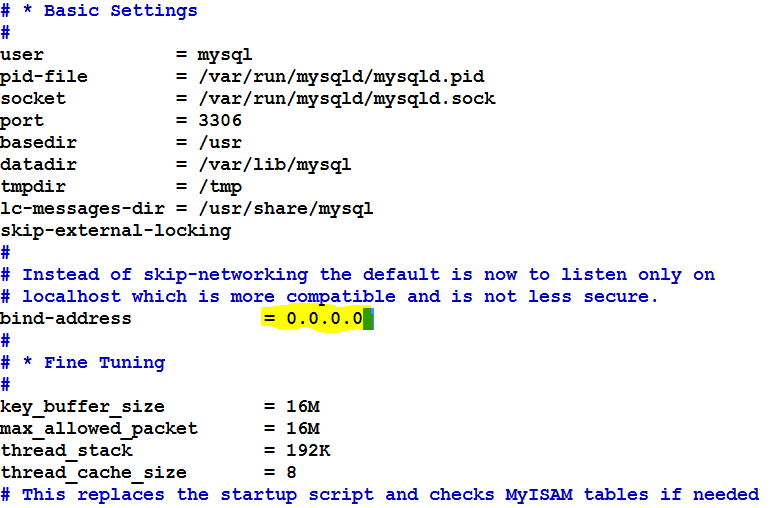
# 4.Start Database Service

**Step 1:** Edit mySQL Configuration file



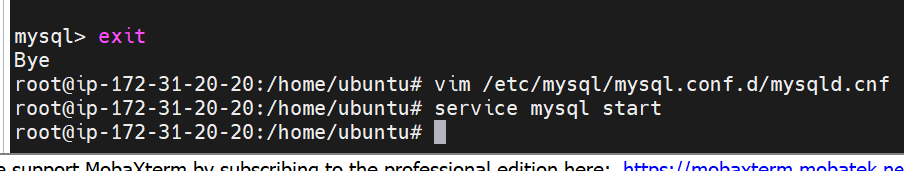


bind-address = 0.0.0.0



**Step 2:** Start the database service

# service mysql start



**Step 3:** Create database and database users

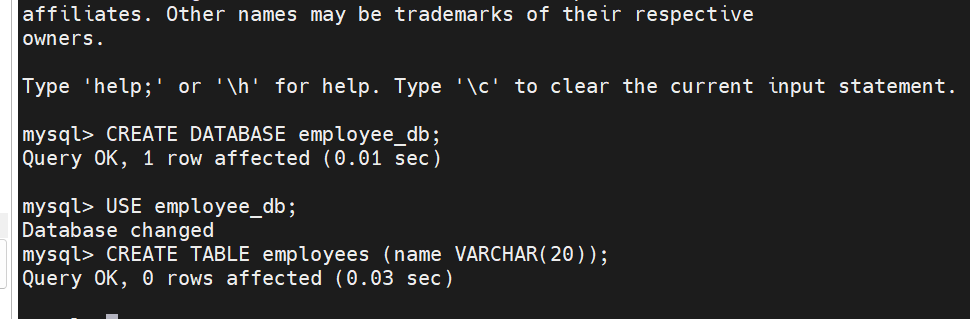
Creating database ‘employee\_db’ and table ‘employees’

# mysql -u root -p

mysql> CREATE DATABASE employee\_db;

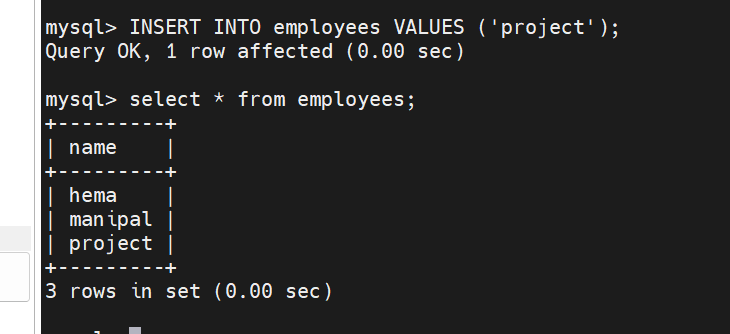
mysql> USE employee\_db;

mysql> CREATE TABLE employees (name VARCHAR(20));



**Step 4:** Insert some test data

mysql> INSERT INTO employees VALUES ('hema');

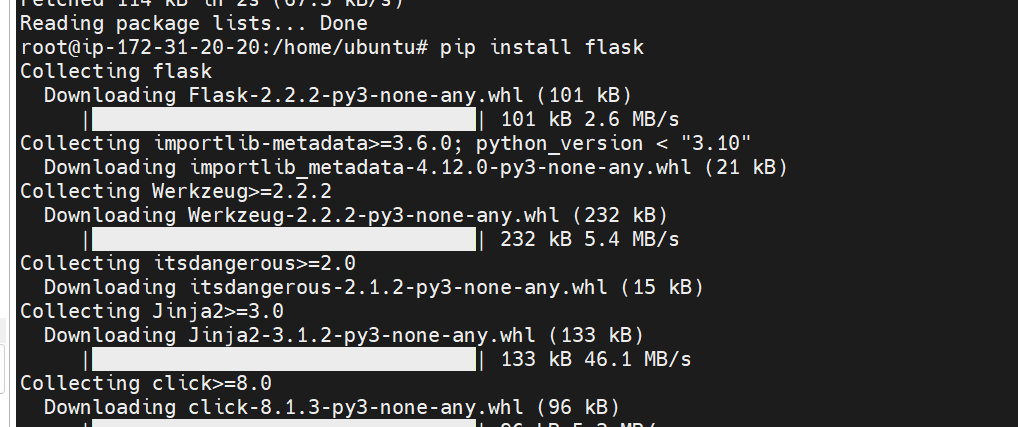


# 5.Install and Configure Web Server

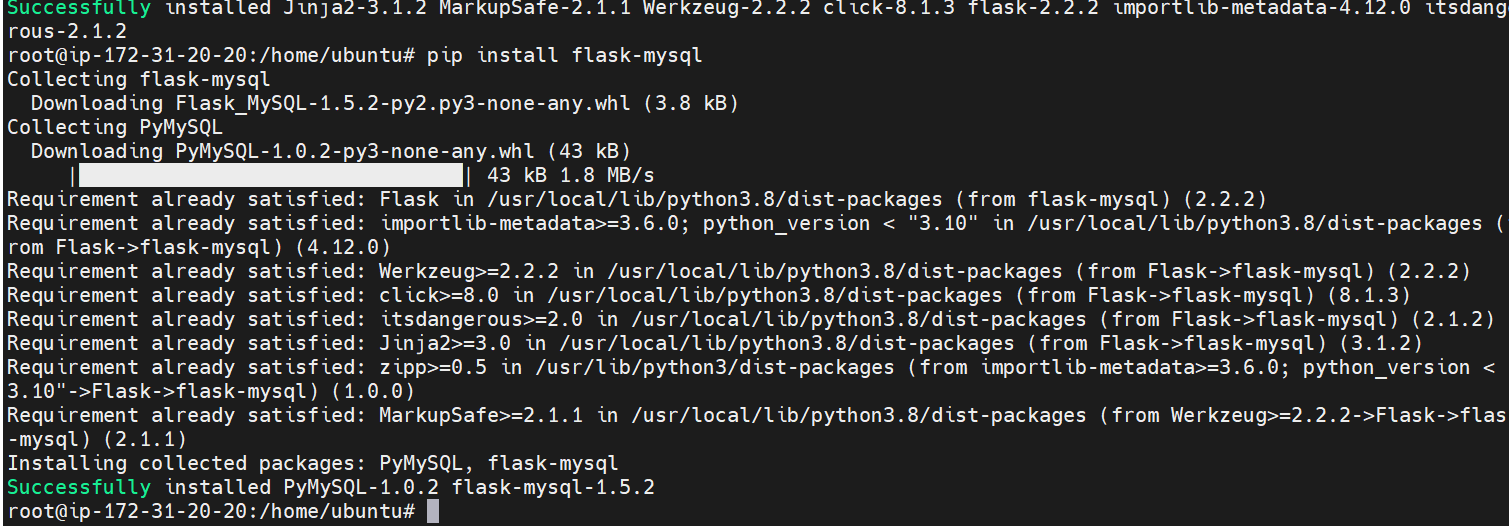
**Step 1:** Install Python flask dependency

Install and Configure Web Server

# pip install flask

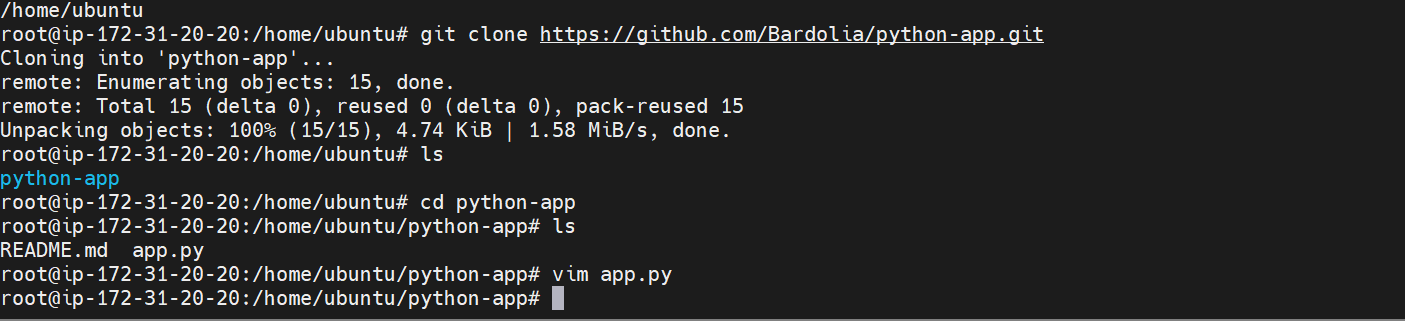


# pip install flask-mysql



**Step 2:** Copy app.py or download it from source repository

# git clone https://github.com/Bardolia/python-app.git

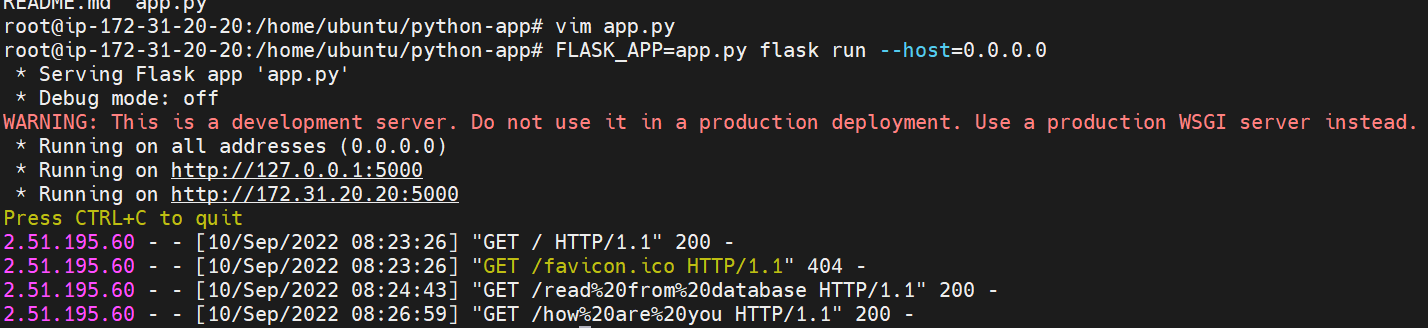


# cd python-app

**Step 3:** Configure database credentials and parameters

# vim app.py

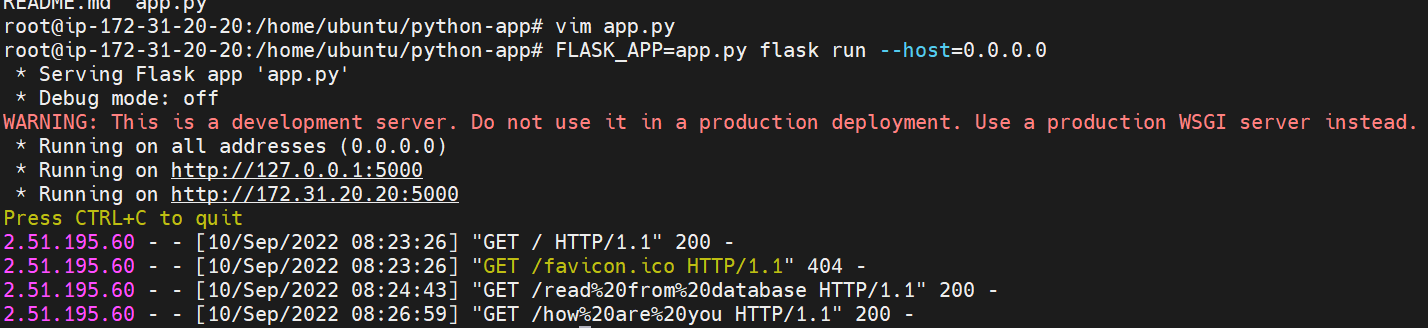
Change the db\_user as ‘root’ and password as ‘password’ which we defined while installing database.



# 6.Start Web Server

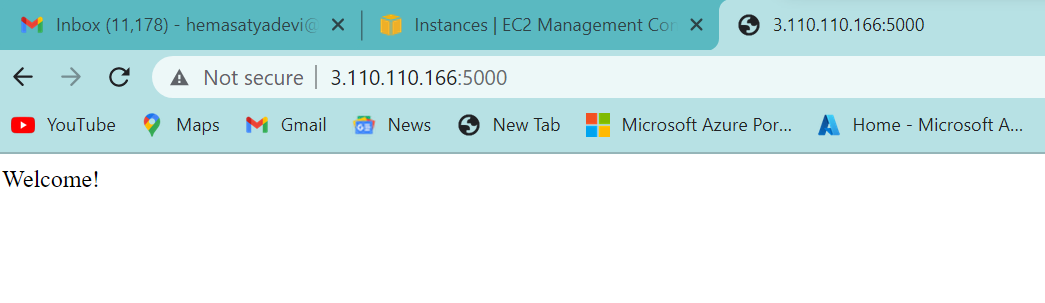
**Step 1:** Start web server

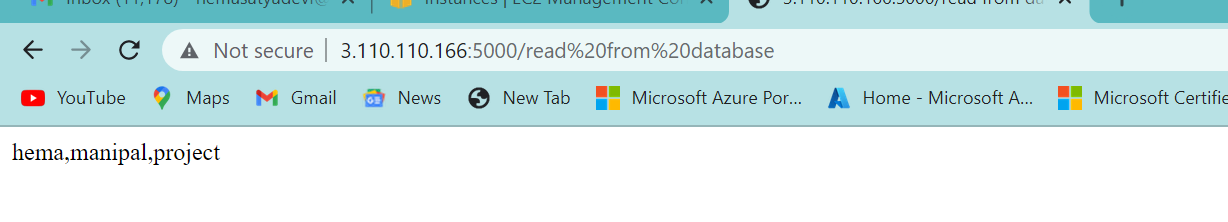
FLASK\_APP=app.py flask run --host=0.0.0.0

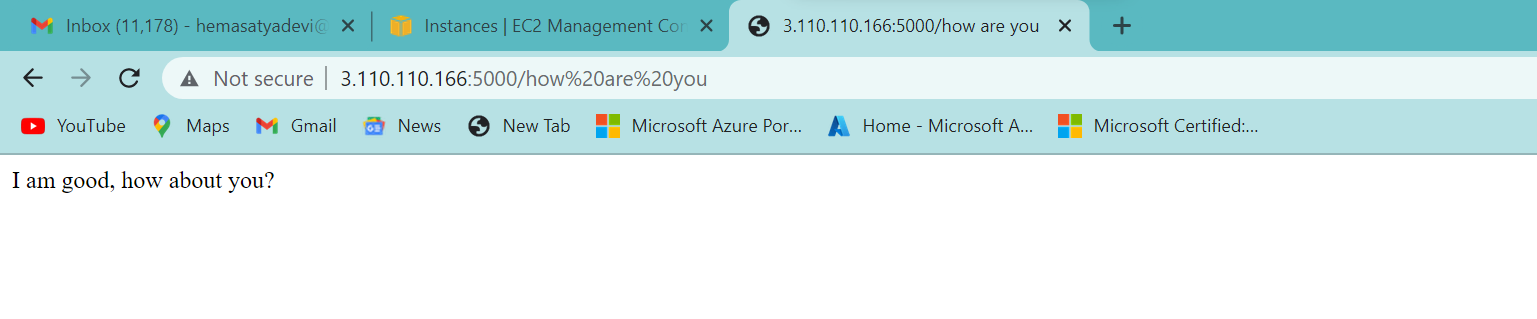


# 7.Test

**Step 1:** Open a browser and go to URL





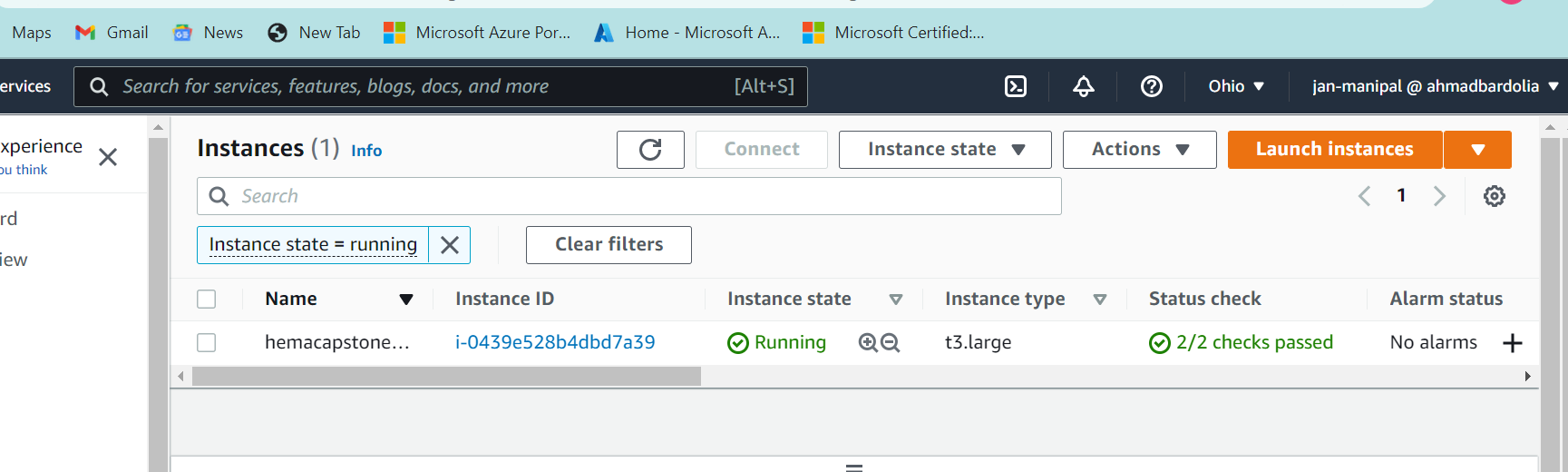


Task2:

Setup replica of existing poc on the cloud of your choice by creating seperate vpc security group private keypair to take remote access and than deploying the sample python webapp and the database server on the said server.

# 1.Creating an Ubuntu linux sever:

Created a replica of an Ubuntu Linux secondary server in Ohio region.

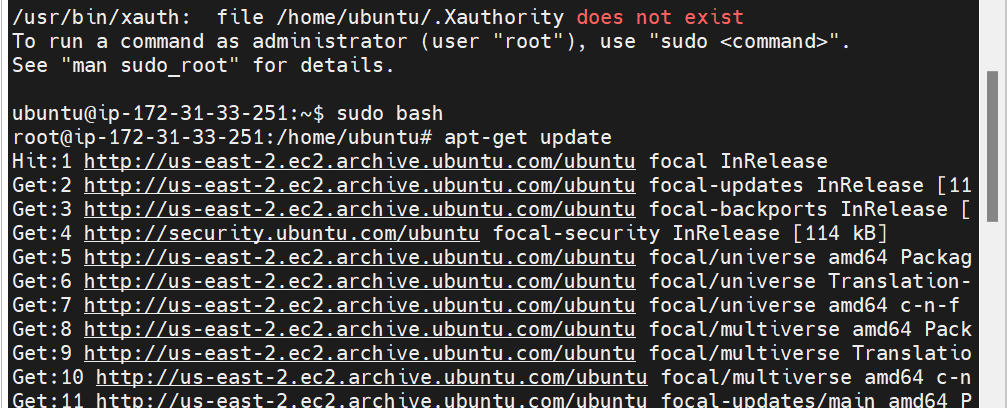


# 2.Install all required dependencies:

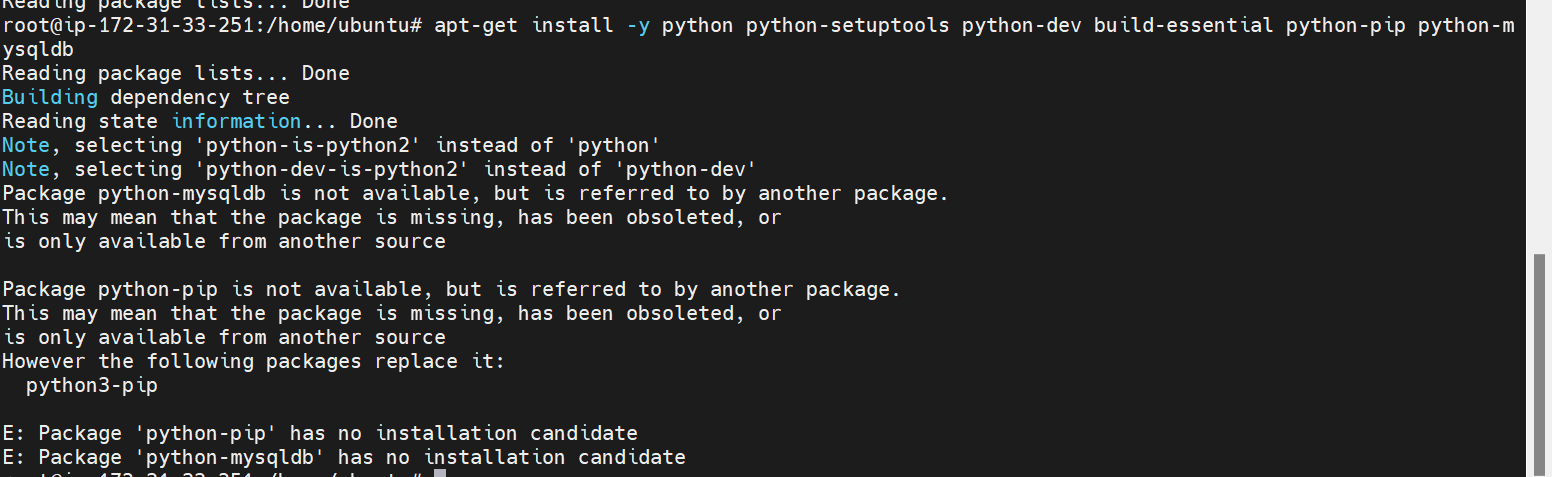
**Step 1:** Python and its dependencies

# sudo su

# apt-get update

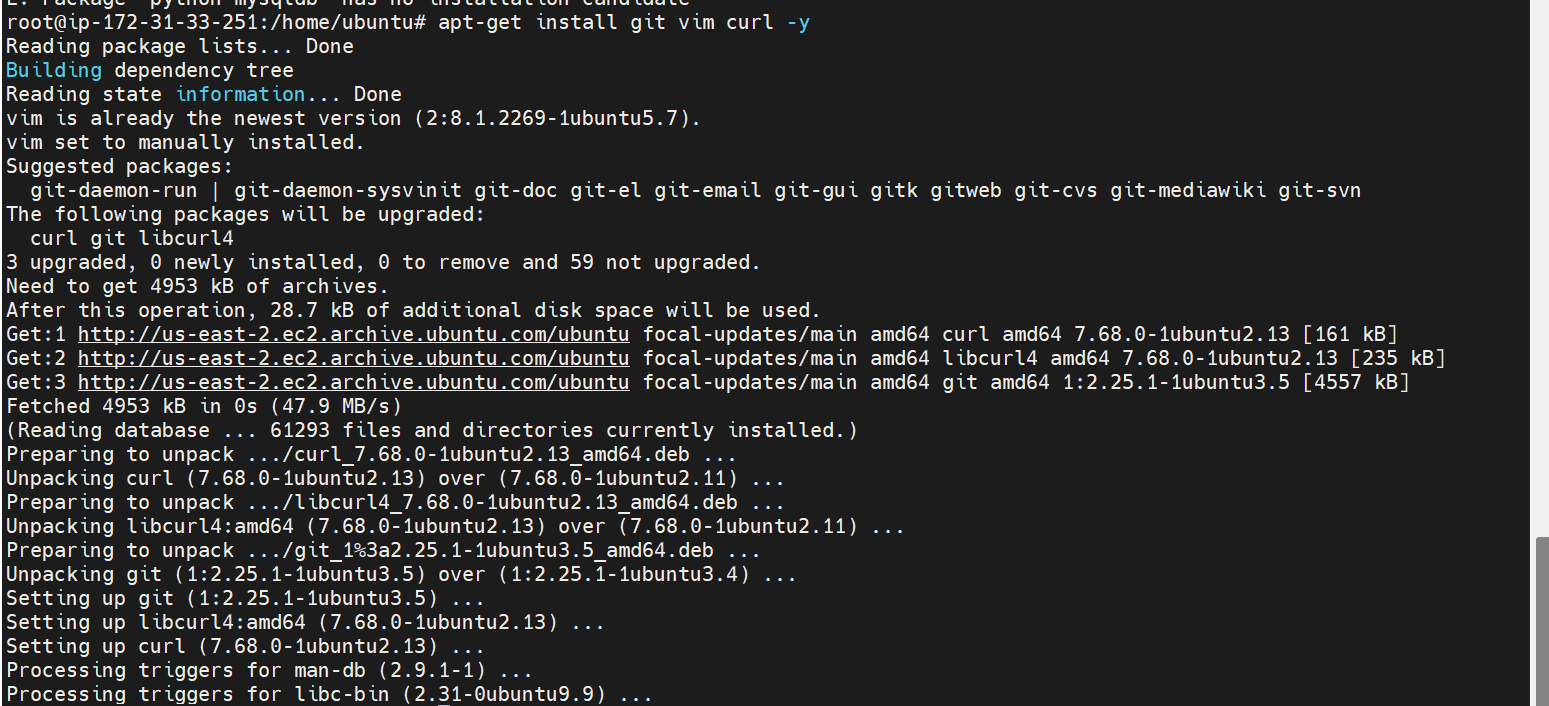


# apt-get install -y python python-setuptools python-dev build-essential python-pip python-mysqldb



**Step 2:** Installing other dependencies

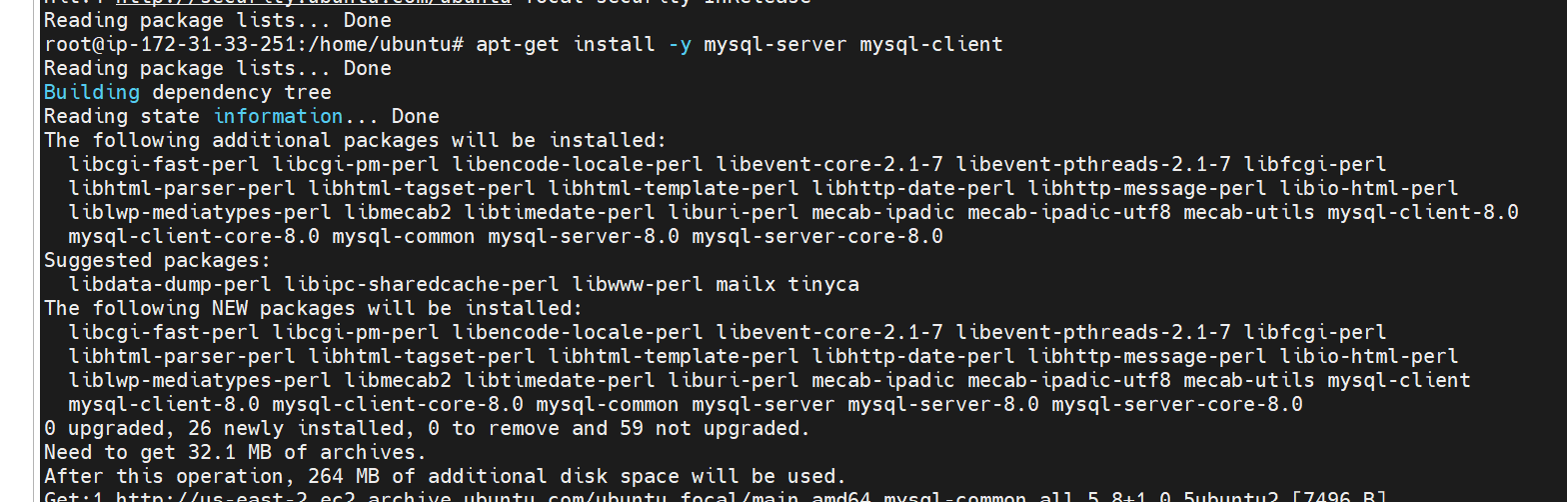
# apt-get install git vim curl -y

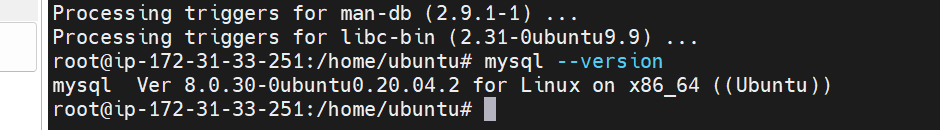


# 3.Install and Configure Database:

**Step 1:** Install MySQL database

# apt-get install -y mysql-server mysql-client



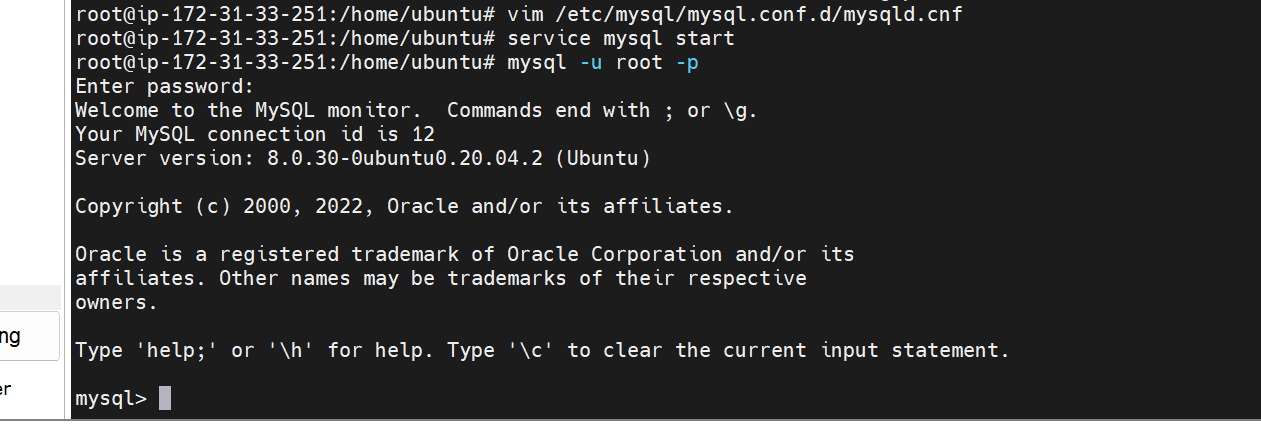


Step2: Adding root access

# Sudo mysql

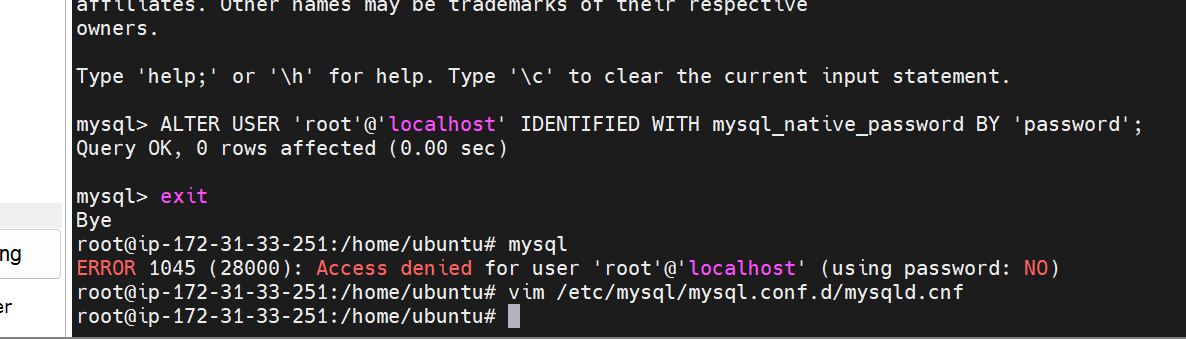
Setting up password to access my sql

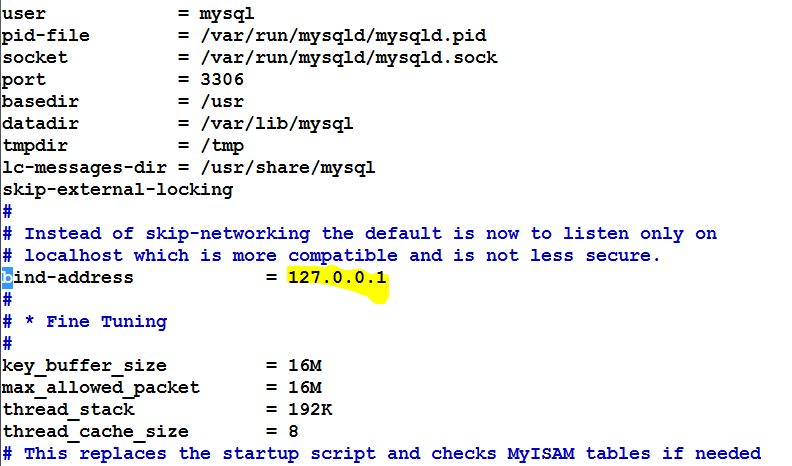
# ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'password';



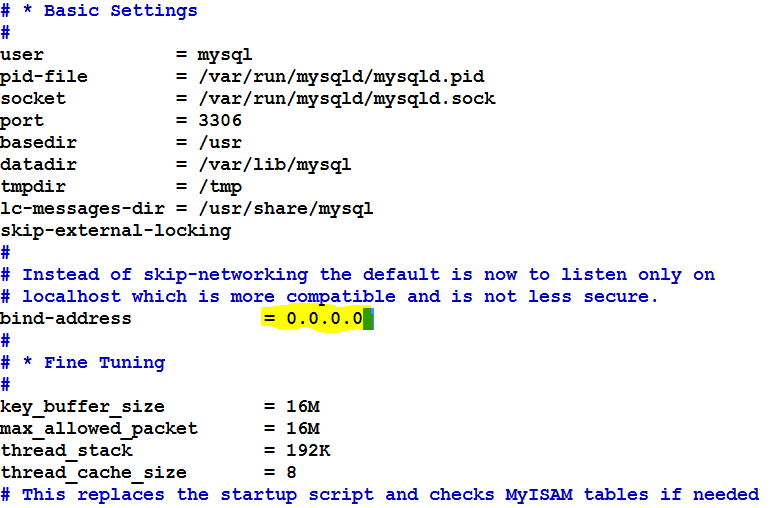
# 4.Start Database Service

**Step 1:** Edit mySQL Configuration file



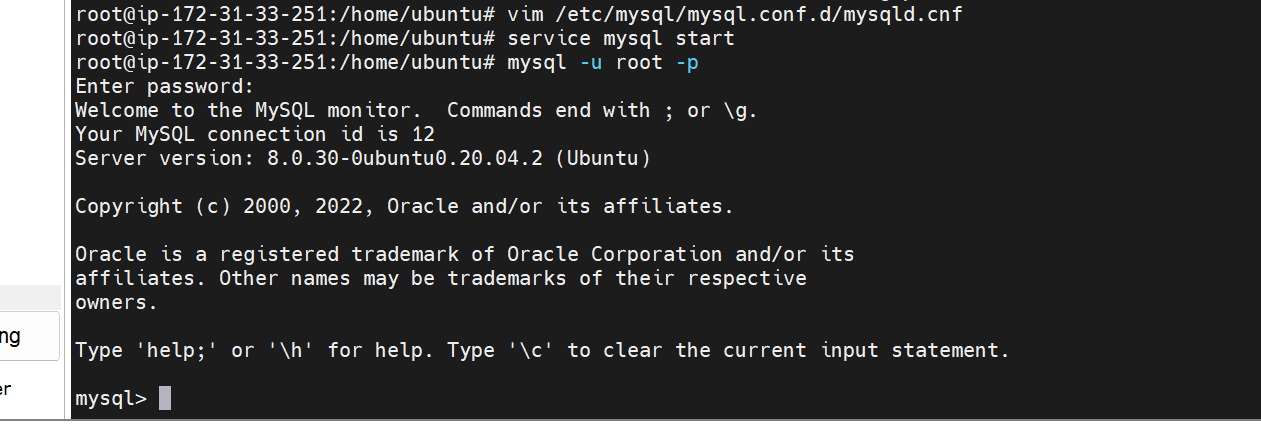


bind-address = 0.0.0.0



**Step 2:** Start the database service

# service mysql start



**Step 3:** Create database and database users

Creating database ‘employee\_db’

# mysql -u root -p

mysql> CREATE DATABASE employee\_db;

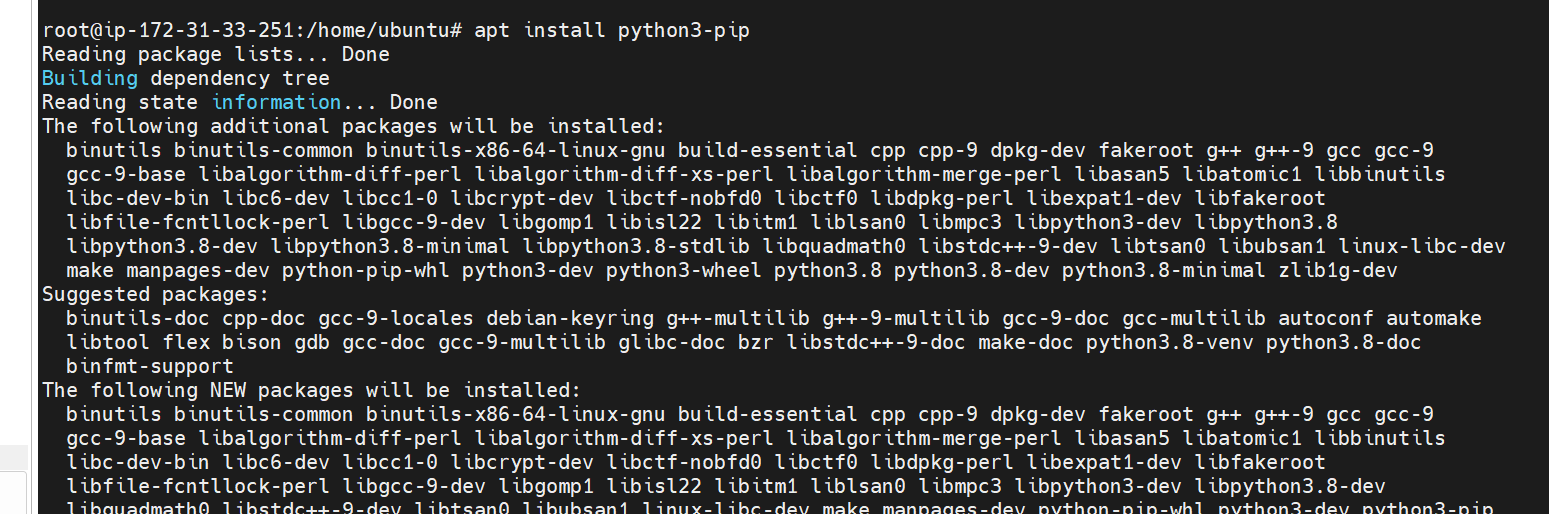
mysql> USE employee\_db;

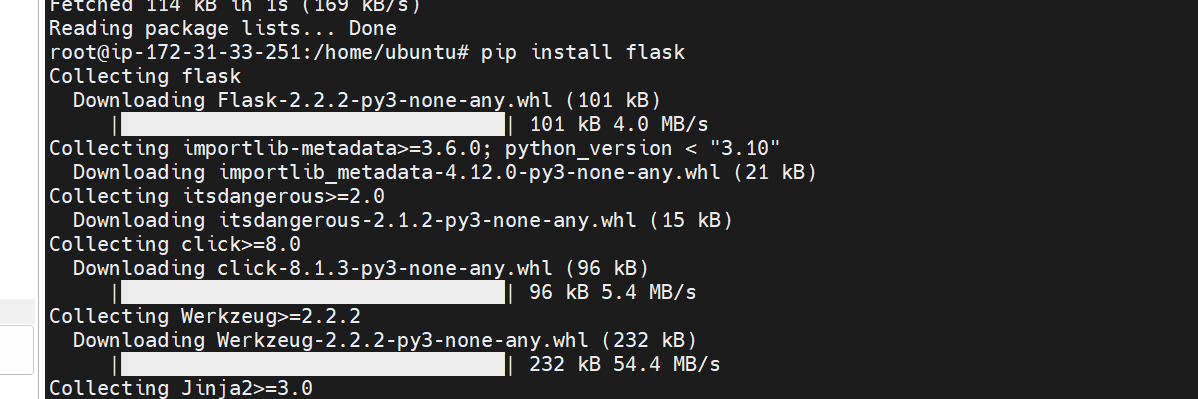
# 5.Install and Configure Web Server

**Step 1:** Install Python flask dependency

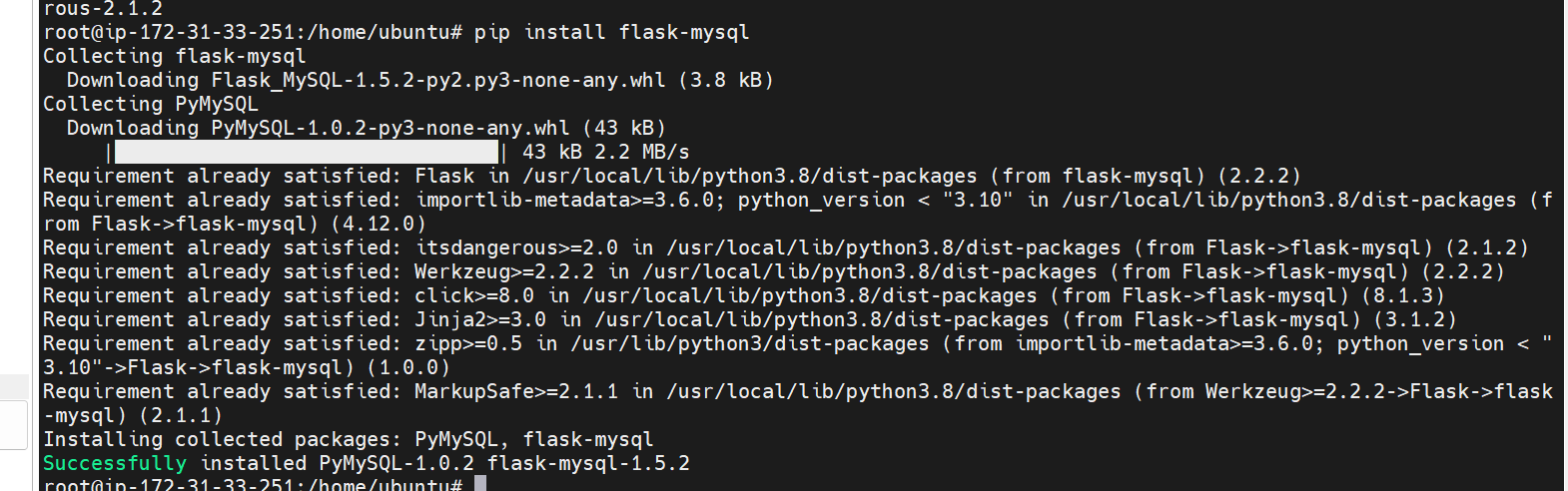
Install and Configure Web Server

# pip install flask



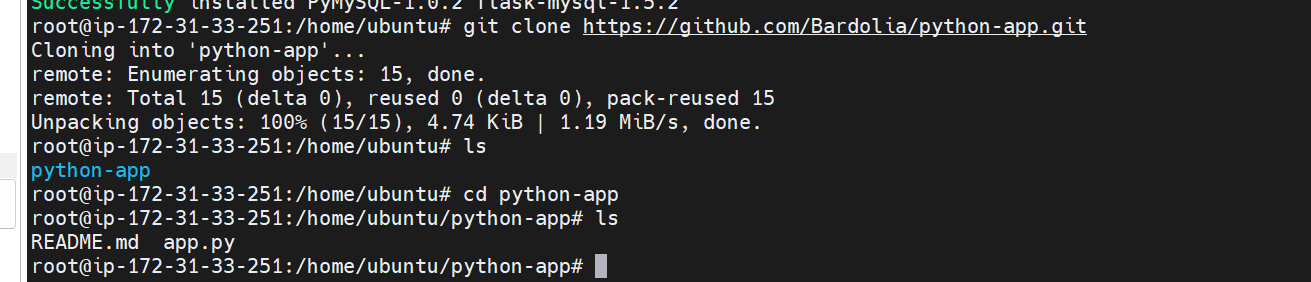


# pip install flask-mysql



**Step 2:** Copy app.py or download it from source repository

# git clone https://github.com/Bardolia/python-app.git

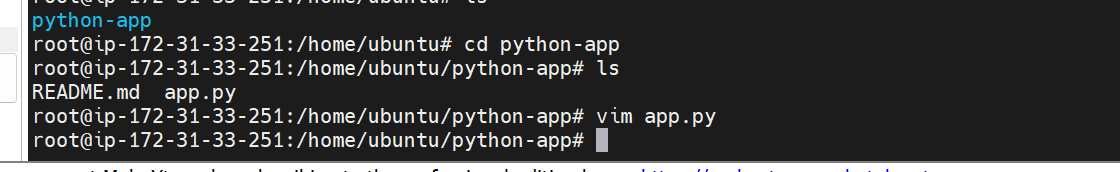


# cd python-app

**Step 3:** Configure database credentials and parameters

# vim app.py

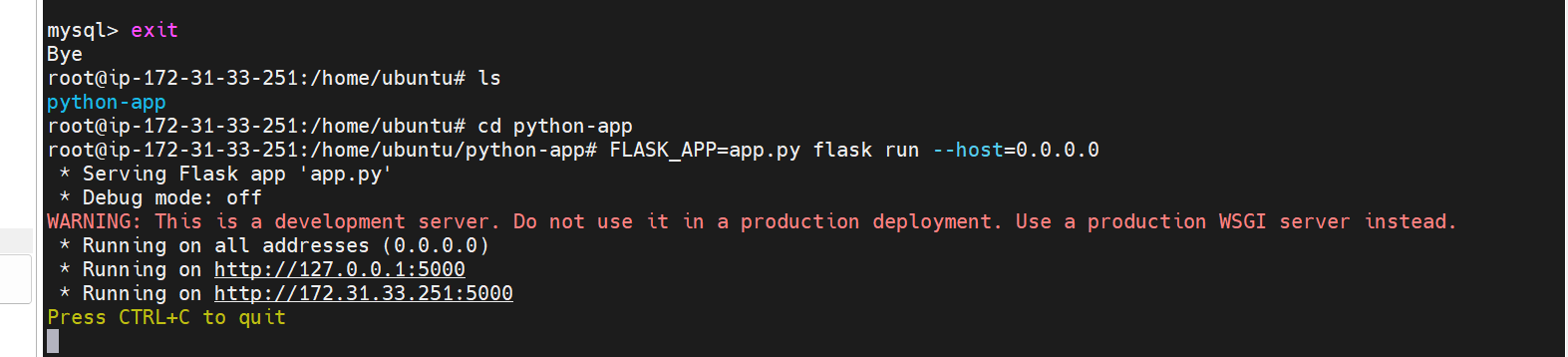
Change the db\_user as ‘root’ and password as ‘password’ which we defined while installing database.



# 6.Start Web Server

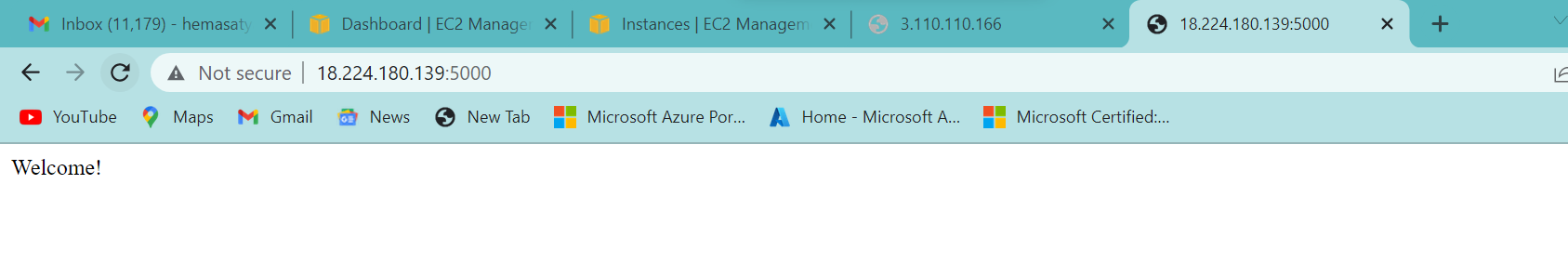
**Step 1:** Start web server

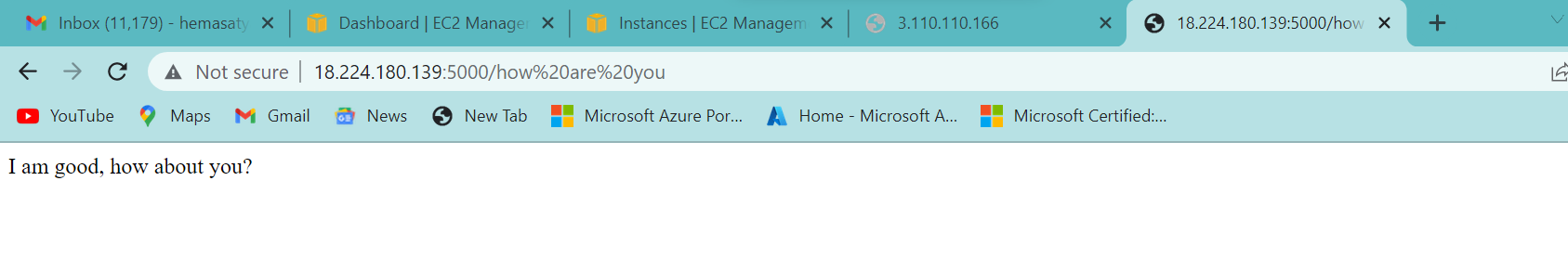
FLASK\_APP=app.py flask run --host=0.0.0.0



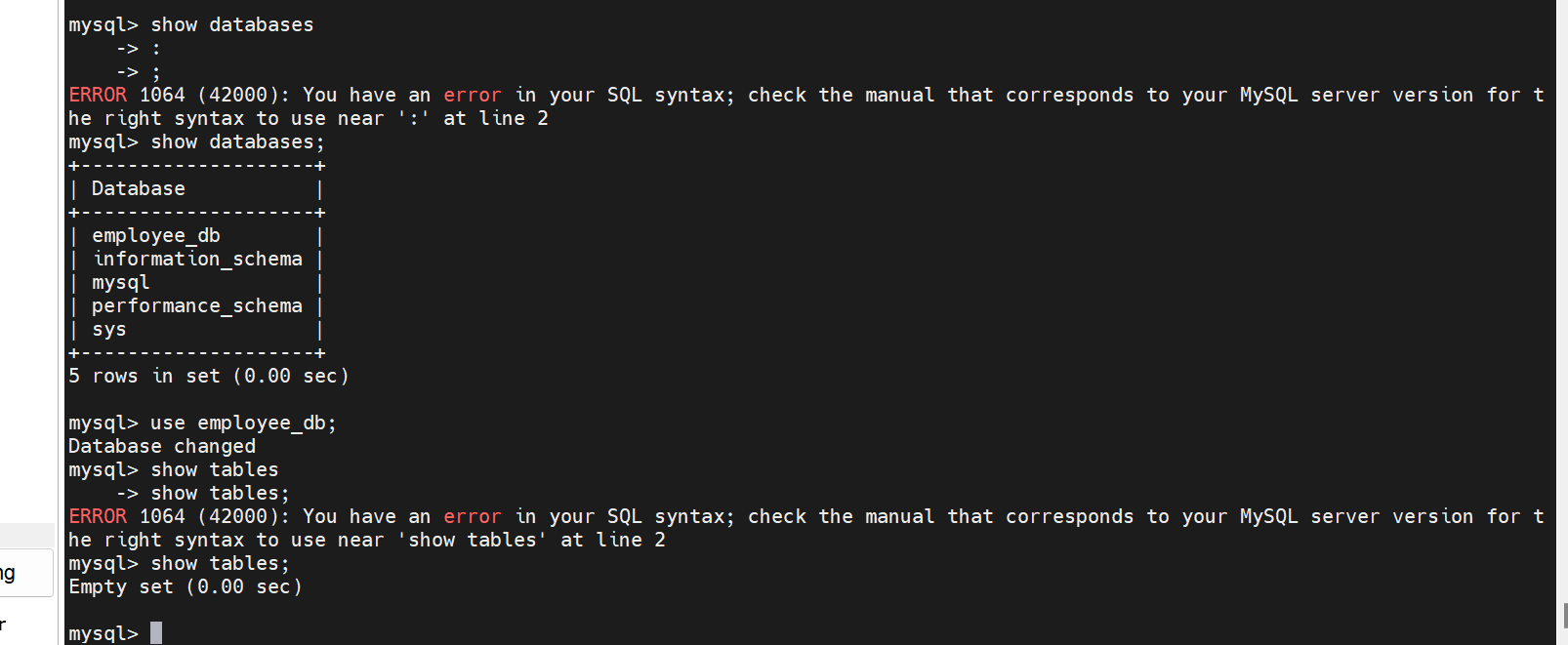
# 7.Test

**Step 1:** Open a browser and go to URL





Note: we don’t fine any tables in the secondary database server.



Task 3 :

Migrate the on-premise database onto the database server running on the cloud and note down the time it took for performing said task.

# 1.Take Backup of Databases from primary Server:

# mysqldump -u root -p employee\_db > employee\_db.sql

# 2.Copy the Data to secondary DB Server

Copy the content of the pem file so that you can take remote access to your cloud instance from your primary server.

# vim key.pem

-----BEGIN RSA PRIVATE KEY-----

MIIEowIBAAKCAQEAgr4cq7d5aMseWLMsER+zK0gtUuxthpvzzdffszzklMX1mWJriQkoEVZNe5p7bTlTfh

7Ksm0WEdtT2BSn1PhMG7uHqLtGk2kijIg5ikqf65hAvNfgd\S\UqBDdyto1WW1Nk2zSdVGGMvm8HjUHo0C

OgVlr5qisA2zc/NqV3FwKBpcF+GzDmZZy6MlU+1LkuJwczyTEVfUCep9u6L88aYuGQbAdEL/gvYPsfdsdf

6mVPSgmSJn/CH+Lj+jQCp8iOIocZHBS4yIyIoa/HXNGHhSc8226ojnPn/g8oQsv6/gAxwhlnMGsKVsvfsf

DUWOlyzQ5vA7BtFLUwnM4J7n4DvUVMOuOMR/KihU+zzjxR/B0NNeMQIDAQABAoIBAChbkvIWsq18saFDG4

daOfuyyWAnKL1yHPxYdVwI6e5apRNx5Avq/alMYKKU7yCqH3HBA4TMU5GziFAdICmHc8bjvl6BYKW3RS23

nEPFvw+QINQwwwdVtU2aFvlENxoYSTW4Cm3yHa8t0d3NDn1XvaY3YEoysi6GYuRpDu0G+zIY0jNT62fsf3

YJFDWtiW1M3766HLid5S1yMeTVAHKQmRa9KhoViusIy8KKgqVYEj6sdfsdgYOOvTsJ5rNUZETMFgzf32QC

TYhquu8GaehBC6sDOwTUuTefQw497m0pNiwyIgu5ECi4GeiWxqZ/Kg3sB36YsdgsdUislnK8QVTluQNSgL

SyX9MUHS51R7A5YP4SsQQaHyqAECgYEAuBFndfq4wfIU1QKKsQpYl1BSfAviI9Ssfdfsd1voVsSgt3mcdA

ei7qYsIgn4szkkKgfPXSGhwtgCqALNqk972wa0cdcoKUQsZ33Pfshfsdf9YzBvgasf0dqiQfSlz7J3Irsw

+I2Zz8Ga4Gh5E1xmBz5Efw3Mmua1sXT6q7+gtu6+pwMJ2BvSe/ECgdfYEAtdXptVsffsQ+fWgqlXZ5TtrD

xUFYr53/HIk0+ZaaTdkMJSfB6rOA+AtSB0HCR9tbzni3ABYFJbPzG+o+Nfdsfdkawufc2WPg91qyOPrZPv

T580i0TLgIqFy6Ok0WYVjFJ2X3WrbsjfklhfkjHJKSLFHsjkldhfsklDFJHFFJKLDFHFDJKLFHKLDFJdhFKLJFHFFJKLFHJKLFHSJKLFHFKJLFs2RmoQdaGhu6pU28+ow4xQsKrnIwt6j2a0pusfTWsdffzASLBeRkEC

gYEAlZbnWzVYo53mmjpHlpvO5aIMdu19ahs7XMu39vEc0Ofimam5KGuV8bsfsfffNCRTJKlWN9zWRrAgcI

ReunOAN79aNE5kQC08XRsc3292n37EkEGtoxFaMKgiIKujohEU6a2wH/qTKee0KTVH2hUxsfdsdfRW3Oh0

bvSfp6bDqQeb/8TzYiMySzECgYBv9LA7rA35qswlUFeqXHmTFwPixRXQ3Q3X6olBCzUf2Hz9/IVxzsffsf

3Ix52VwkYAQroRmTnPs4Vo6FjnkVc/rVu6YYue4KHBAXoHEFBhIU+ZuOnjt4wV32a7MyvYcSiWGvasfFDS

aRsffsfLjbKJmVtJZev6zCOjfMUg3Vil2uzpVnvNBCodteeHhwQKBgAzTTUths0Y+Qs57sOySrm1sdQT4b

h69Nui7aEFFnMuoDqTcoQqm07CXzNFdBOkWvUAEZ2sUCChMjn9QzgZQpTBsBrsA4pFDbBbkC7c4esdfgsg

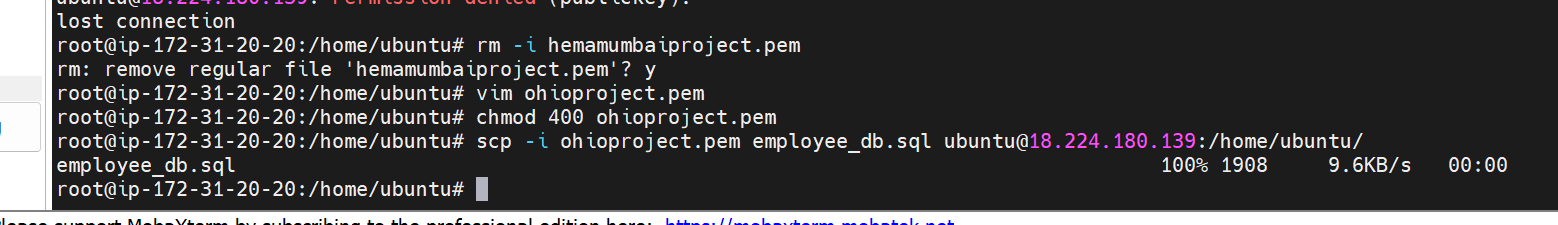
xm0K9rOer126rOADHzIQEteSsuAgfCXzj9kEAZ3nUhteTMktFQmhRE+Zb4Pu4OOBnGgWzdsssfsfsdffss

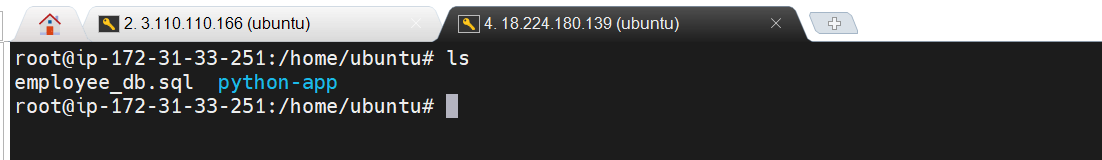
-----END RSA PRIVATE KEY-----

remotely copy the sql dump onto the cloud server

# chmod 400 key.pem

# scp -i key.pem employee\_db.sql [ubuntu@18.224.180.139:/home/ubuntu/](mailto:ubuntu@18.224.180.139:/home/ubuntu/)



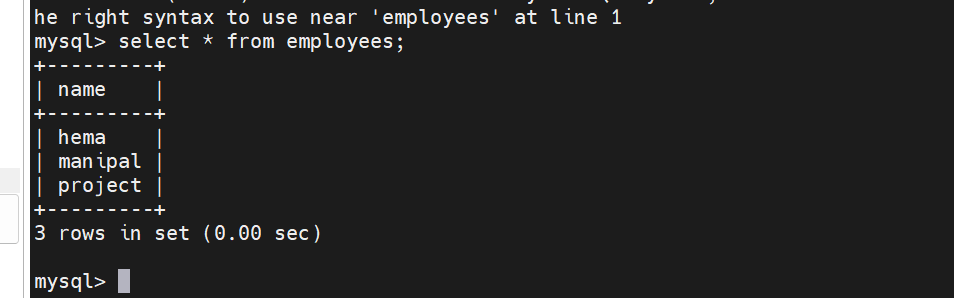


Data is copied to the secondary database as shown above.

# Verify the migration

Restore the database on the cloud server

# mysql -u root -p employee\_db < employee\_db.sql



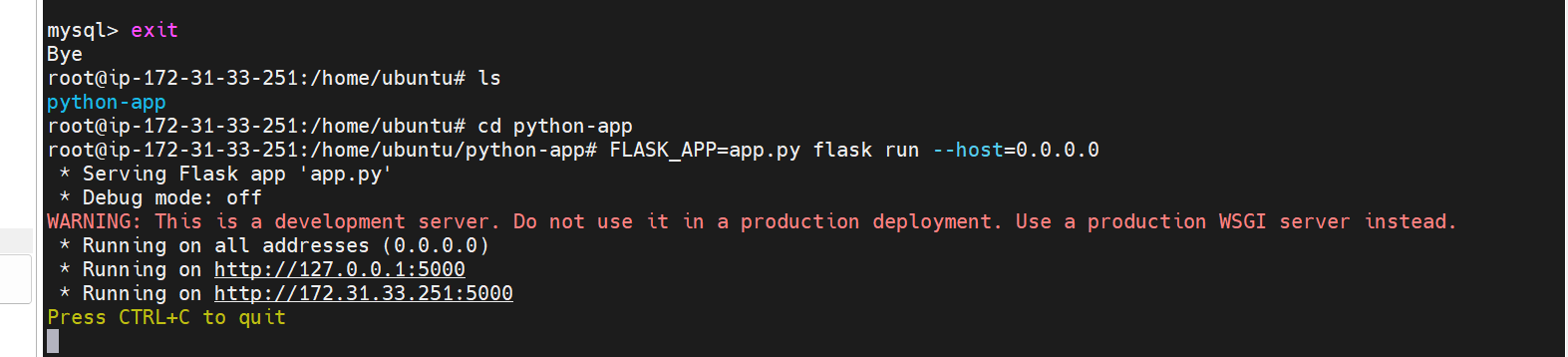
# Start Web Server

Run the migrated web server back on the cloud with existing database.

**Start web server**

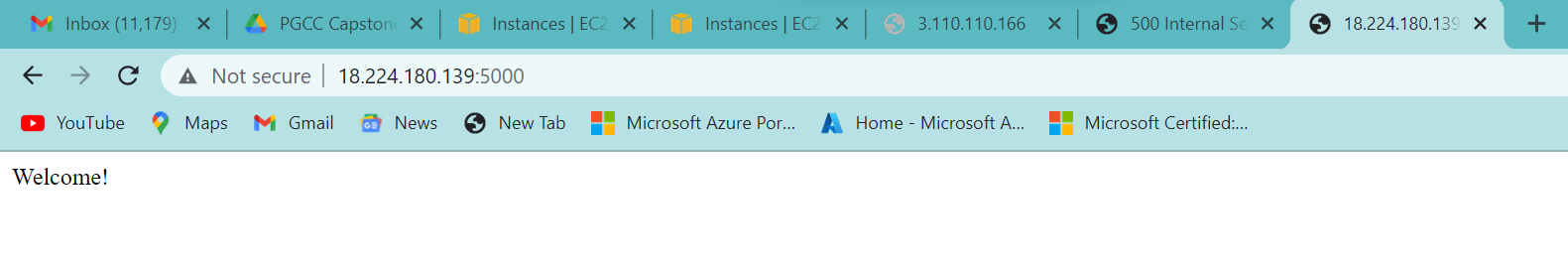
# cd /home/ubuntu/python-app/

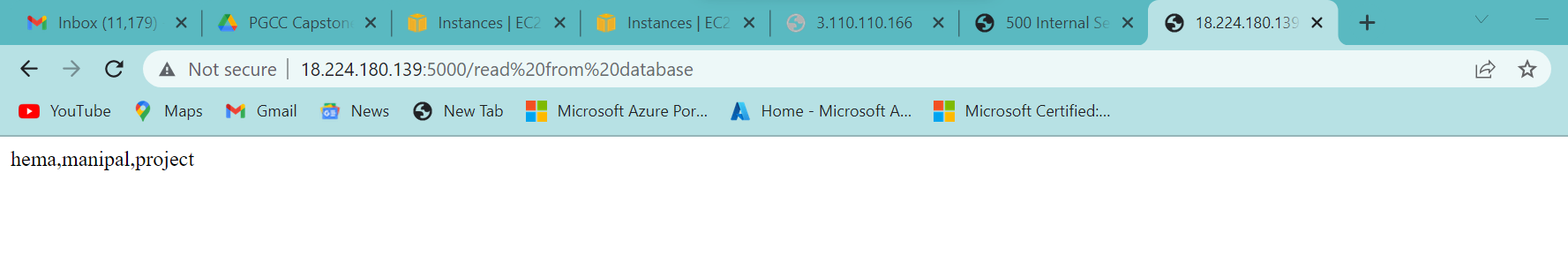
FLASK\_APP=app.py flask run --host=0.0.0.0



# Test

**Open a browser and go to URL**





Successfully completed the migrating data from primary database server in Mumbai region to secondary database server in Ohio region.

Task 4 :

Replicate all the dependencies application libraries and configuration from the primary server to the server running on the cloud.

# Install all required dependencies

# **Step 1:** Python and its dependencies

# sudo su

# apt-get update

**Step 2:** Installing other dependencies

# apt-get install git vim curl -y