**2 Tier Architecture**

**Prerequisites:**

1. AWS Account (Avoid root user)

2. Enable the MFA for Your account

3. Terraform

4. AWS CLI

5. VS code

Do Git clone <https://github.com/gayatripawar2001/web-db-deployment.git>

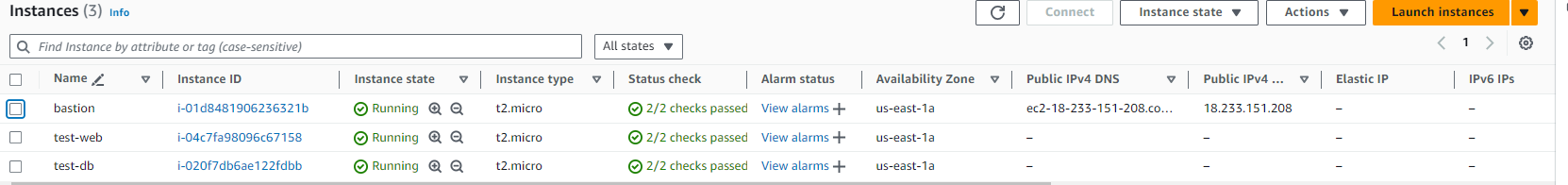
Go to Terraform script diectory

Do aws configure and run the below command

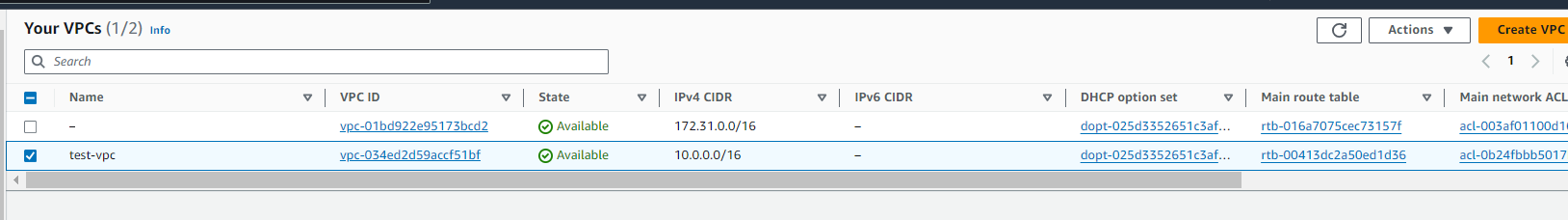
1. Terraform init
2. Terraform plan
3. Terrafrom apply

After terraform successful…

* **EC2**



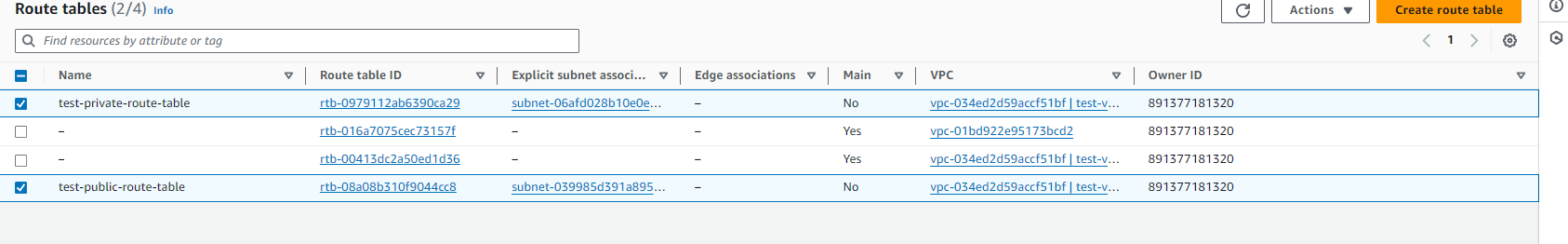
* **VPC**



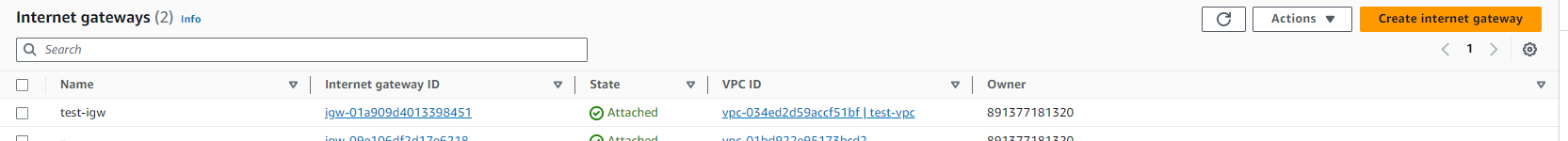
* **Subnets**



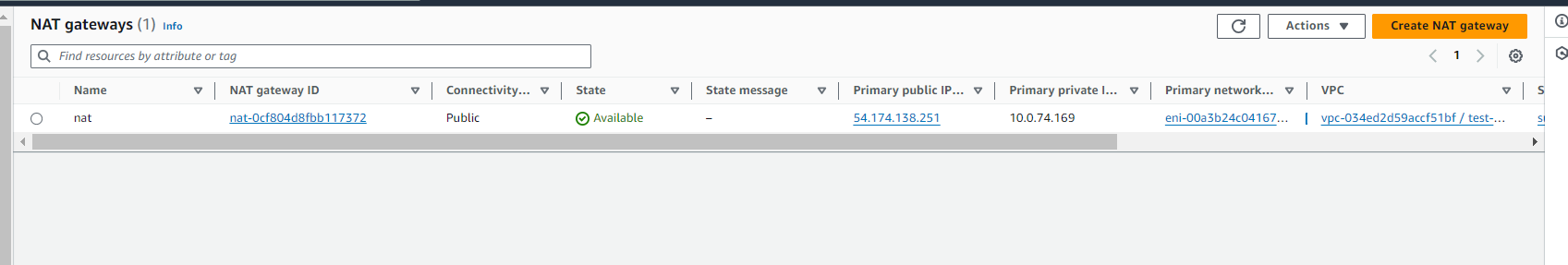
* **Route tables**



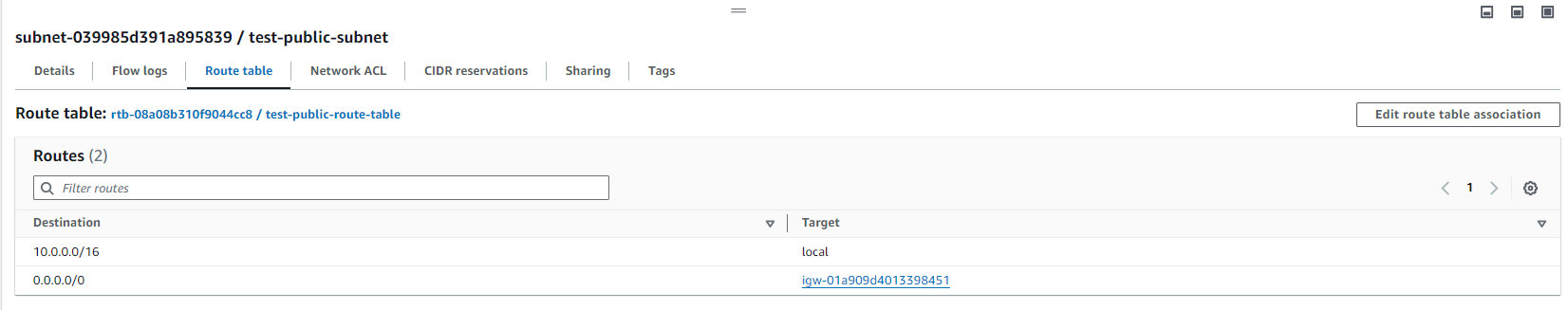
* **Igw**



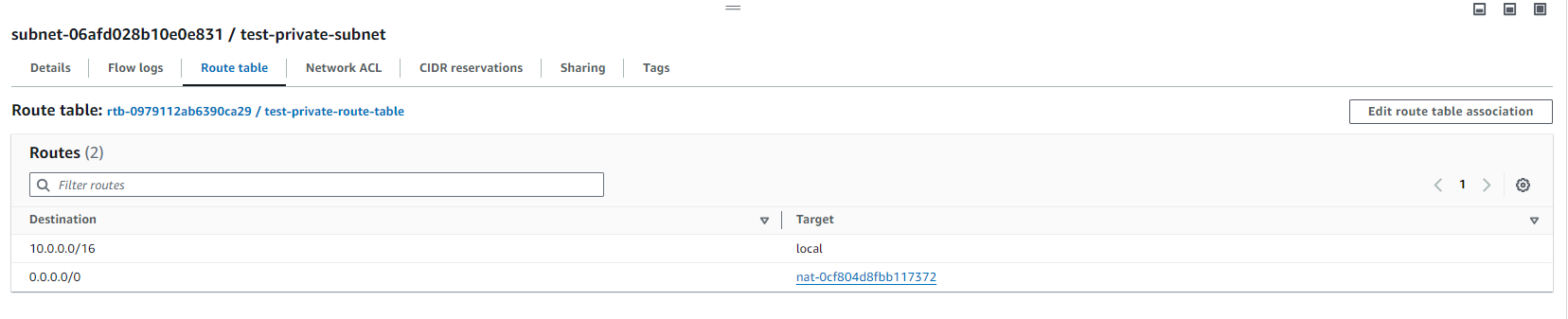
* **Nat Gateway**



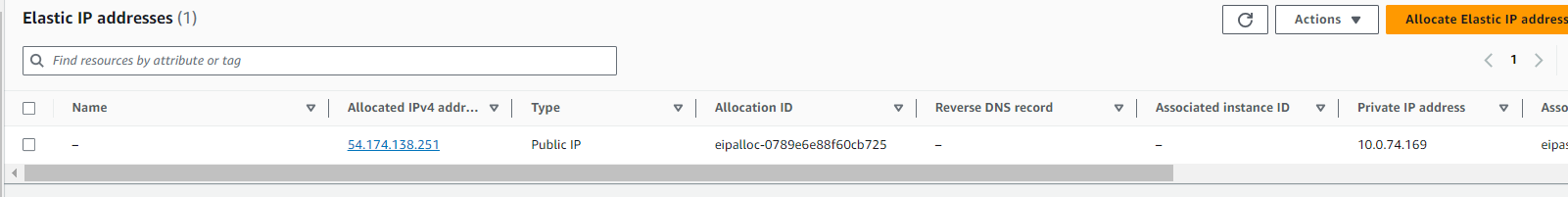
* **Public subnet route assocition**



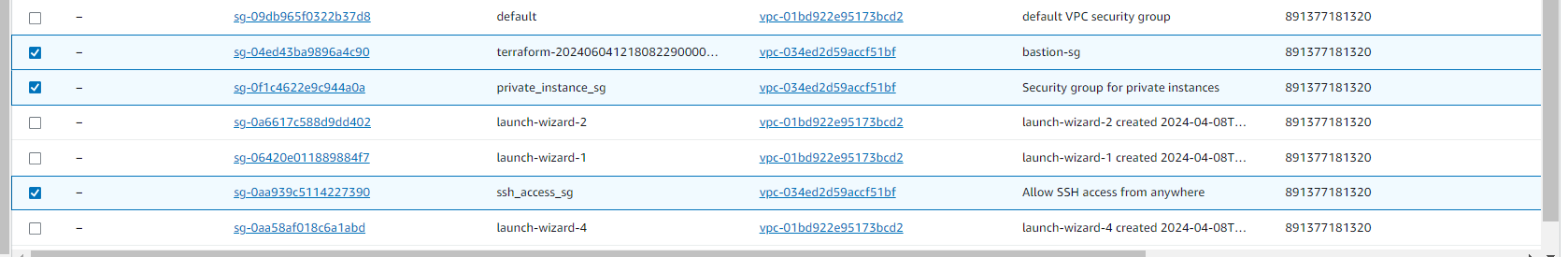
* **Private subnet route association**



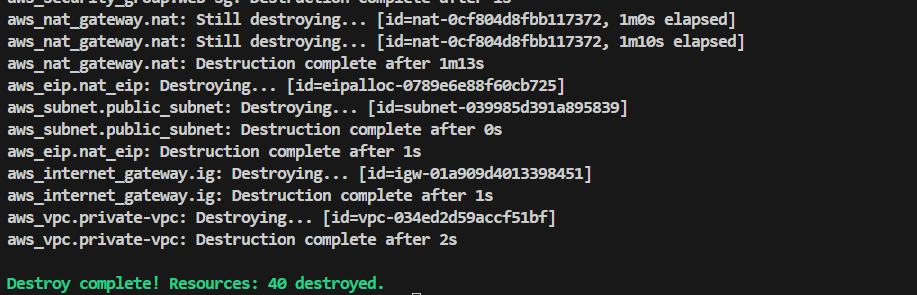
* **Eip**



* **Security groups**



* **Terraform destroy**



SSH into the bastion host to connect to the private web server

* **setting up the web application on one ec2 instance using flask**

**Run the below script**

**Flask.sh**

#!/bin/bash

sudo apt update

sudo apt install -y python3 python3-venv

mkdir flaskapp\_gunicorn

cd flaskapp\_gunicorn

python3 -m venv venv\_gunicorn

source venv\_gunicorn/bin/activate

pip install flask

cat <<EOF > app.py

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def hello():

return 'Hello world!'

if \_\_name\_\_ == '\_\_main\_\_':

app.run()

EOF

**Provide all the details related to the database in code**

**Init\_db.py**

import os

import psycopg2

conn = psycopg2.connect(

host="server\_private\_ip",

database="db",

user=os.environ['DB\_USERNAME'],

password=os.environ['DB\_PASSWORD'])

# Open a cursor to perform database operations

cur = conn.cursor()

# Execute a command: this creates a new table

cur.execute('DROP TABLE IF EXISTS books;')

cur.execute('CREATE TABLE books (id serial PRIMARY KEY,'

'Name varchar (50) NOT NULL,'

'age integer NOT NULL,:'

)

# Insert data into the table

cur.execute('INSERT INTO books (name, age)'

'VALUES (%s, %s)',

('Gayatri',

18)

)

cur.execute('INSERT INTO books (name, age)'

'VALUES (%s, %s)',

('krishna',

20)

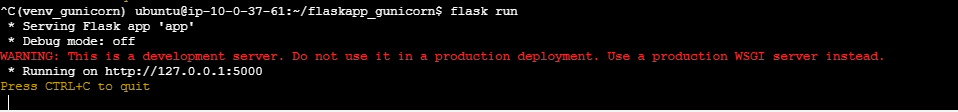
)

conn.commit()

cur.close()

conn.close()

Flask run



Flask application is running….

SSH into the DB server using bastion host

* **On db server install PostgreSQL Installation, Run the below script**

**PostgreSQL**

#!/bin/bash

sudo apt update

sudo apt install -y wget ca-certificates

wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -

sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ $(lsb\_release -cs)-pgdg main" >> /etc/apt/sources.list.d/pgdg.list'

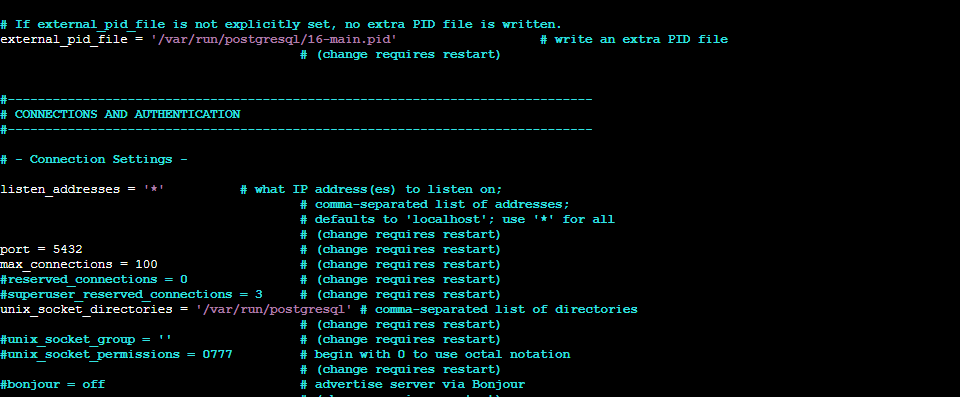
sudo apt update

sudo apt install -y postgresql postgresql-contrib

**Configuration settings**

Change the configuration of the postgresql.conf

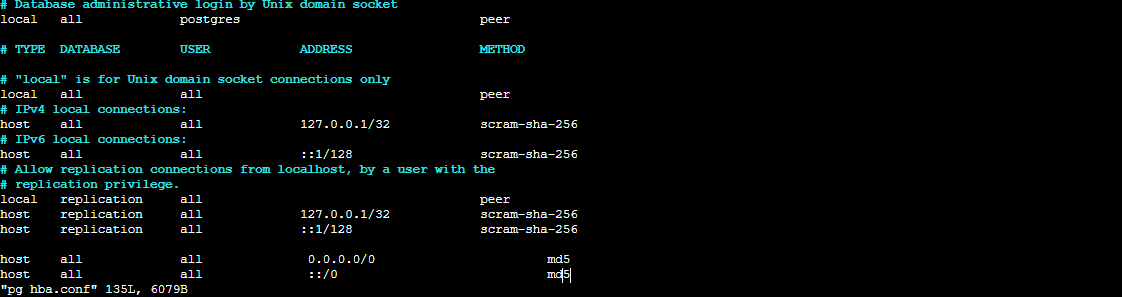
/etc/postgresql-- uncomment the listen\_address and add the \* as value



Now add the belwo lines of code in to the file pg\_hba.conf

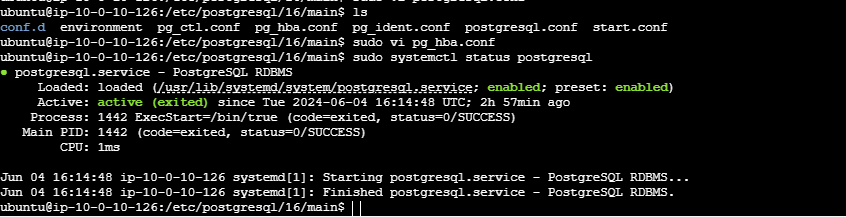
host all all 0.0.0.0/0 md5

host all all ::/0 md5



Now Restart the postgressql service

Sudo systemctl restart postgresql

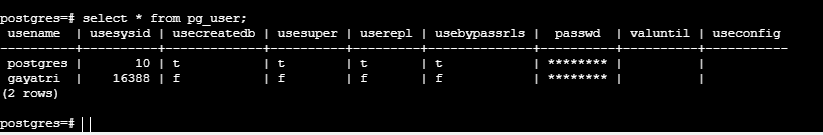


**Create a user , database and table**

sudo -u postgres psql

CREATE USER gayatri WITH PASSWORD '123';

GRANT ALL PRIVILEGES ON DATABASE user TO gayatri;



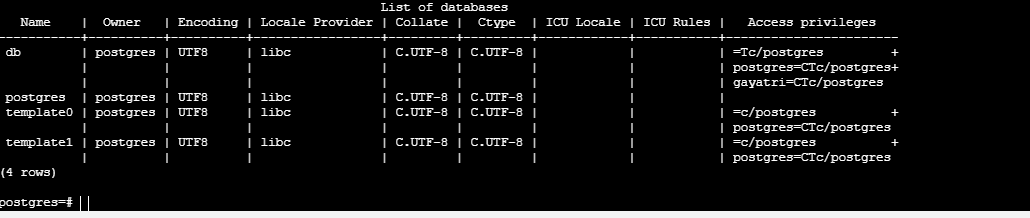
CREATE DATABASE db;

CREATE TABLE test (

name varchar(30),

age integer

);

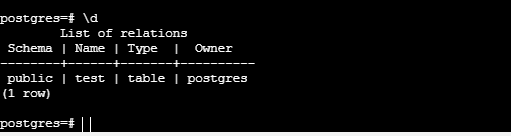


INSERT INTO test (name, age)

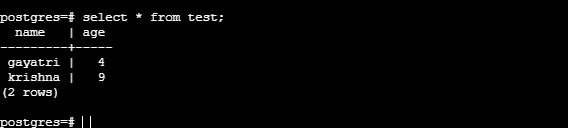
VALUES ('Dolly', '7'),

('gayatri', '4'),

('krishna', '9');

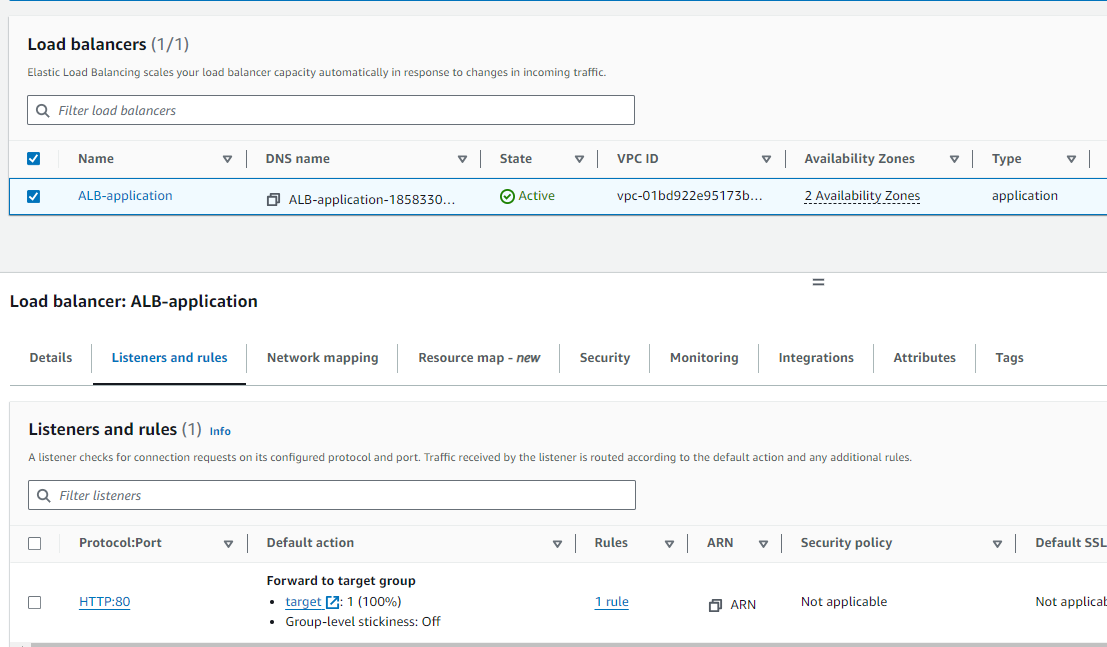


SELECT \* FROM test;

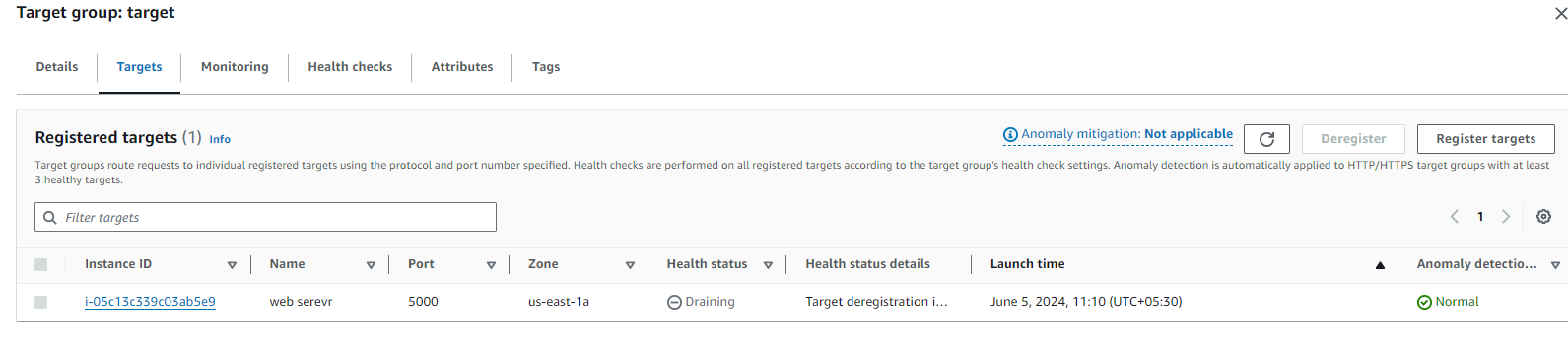


**Accessing the application from ALB**

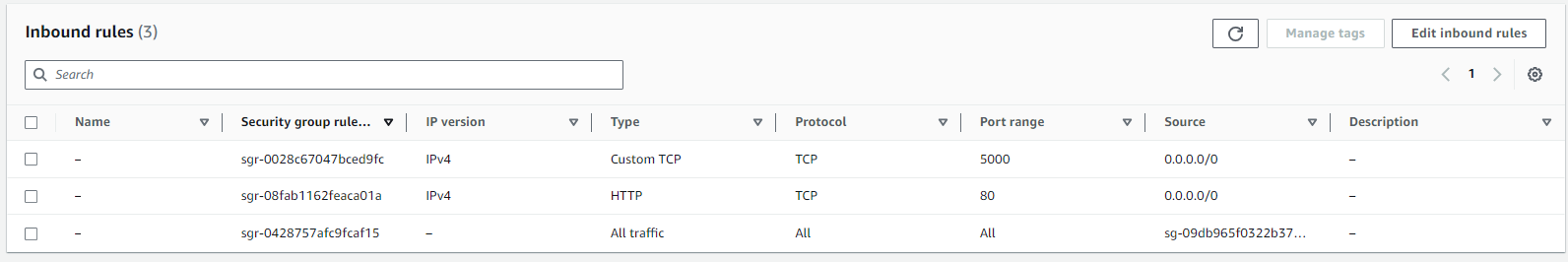
Create a ALB and Target groups



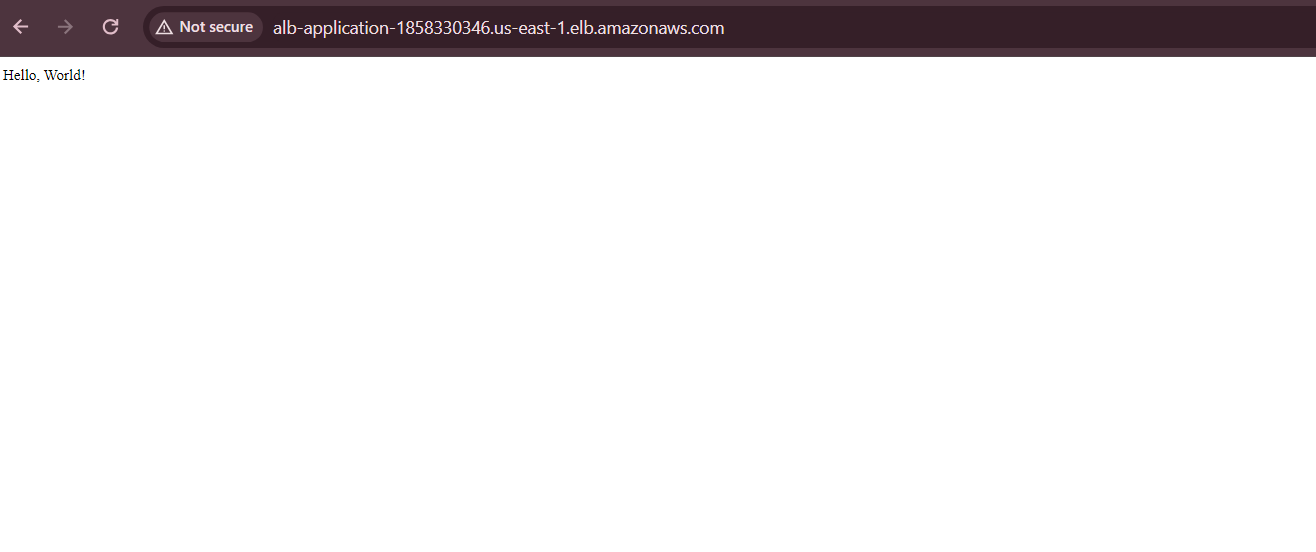
Target group added the registered targets fro port 5000



ALB security group allow 80 and port 5000 from all source

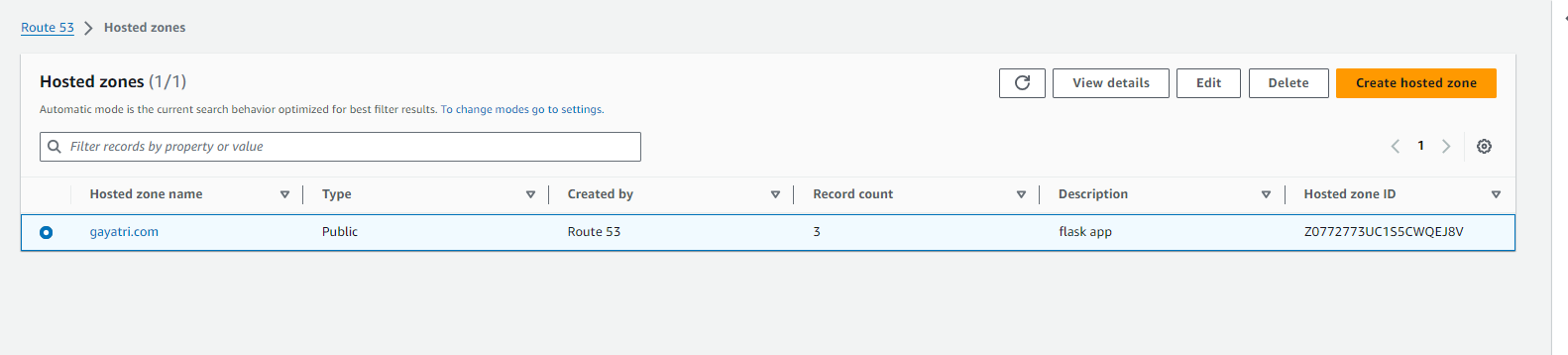


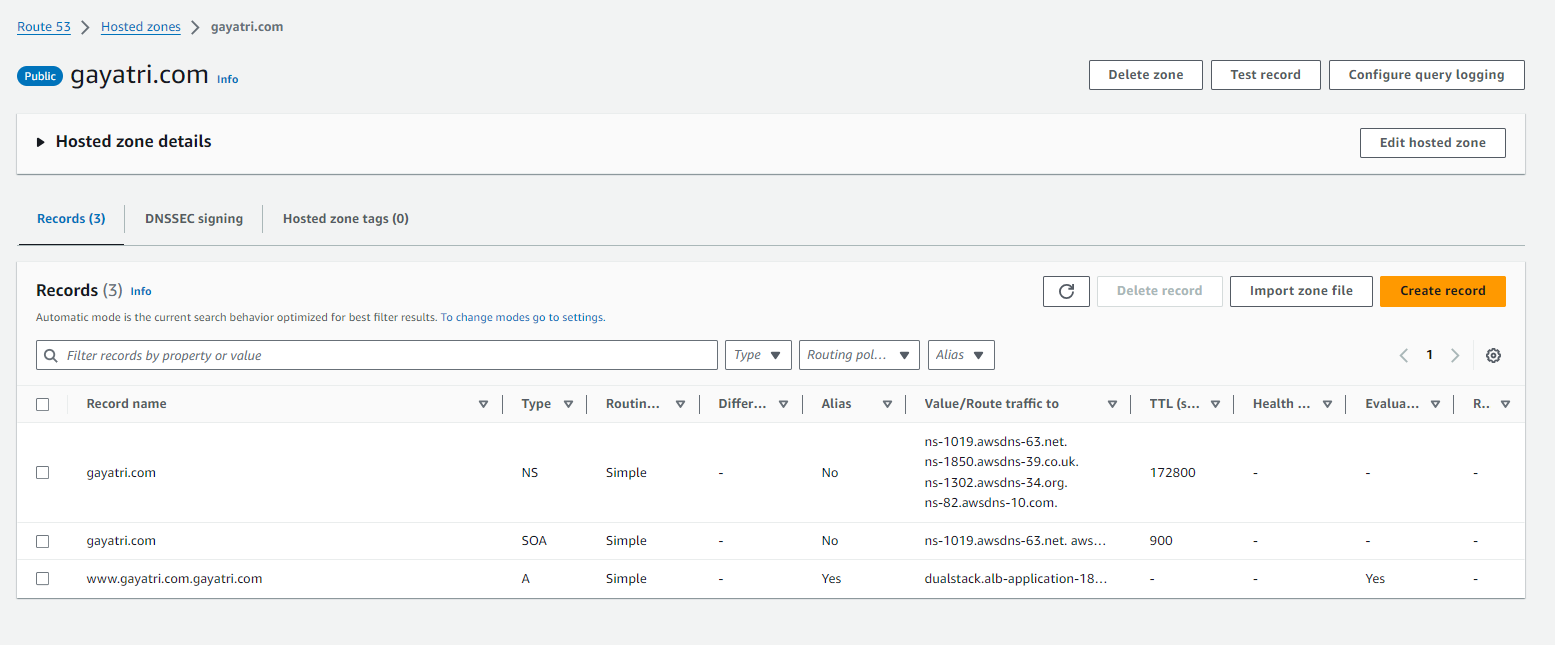
Run the flask app and access it through the alb dns



**Creating the Hosted Zone for application**

Created the hosted dns for application which points to ALB





---------------------------------------------------------END------------------------------------------------------------