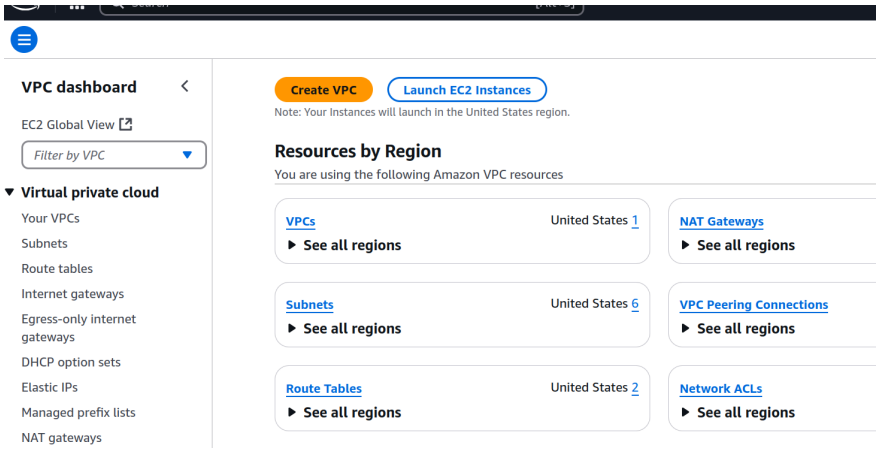
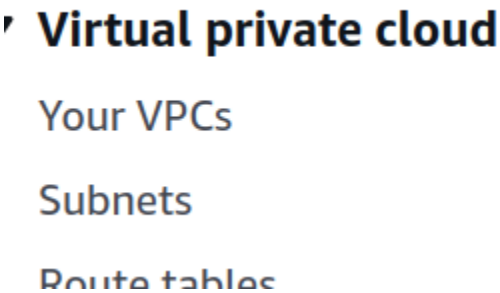
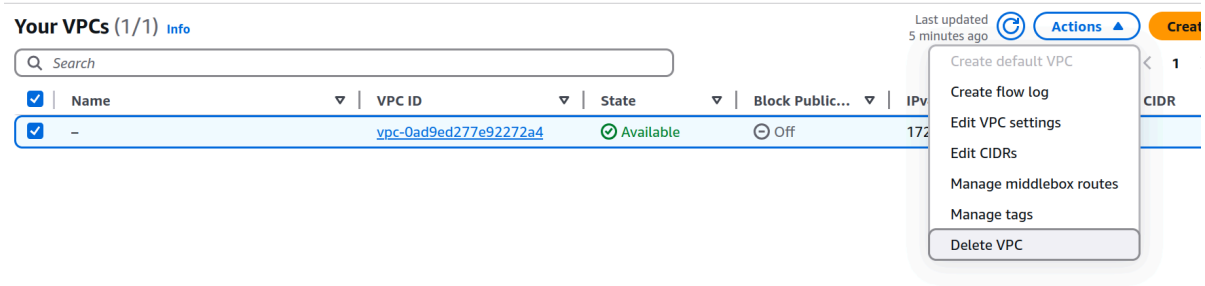
Make VPC

1) Go to dashboard and look up VPC in search bar. Go to VPC dashboard, and then go to your VPCS

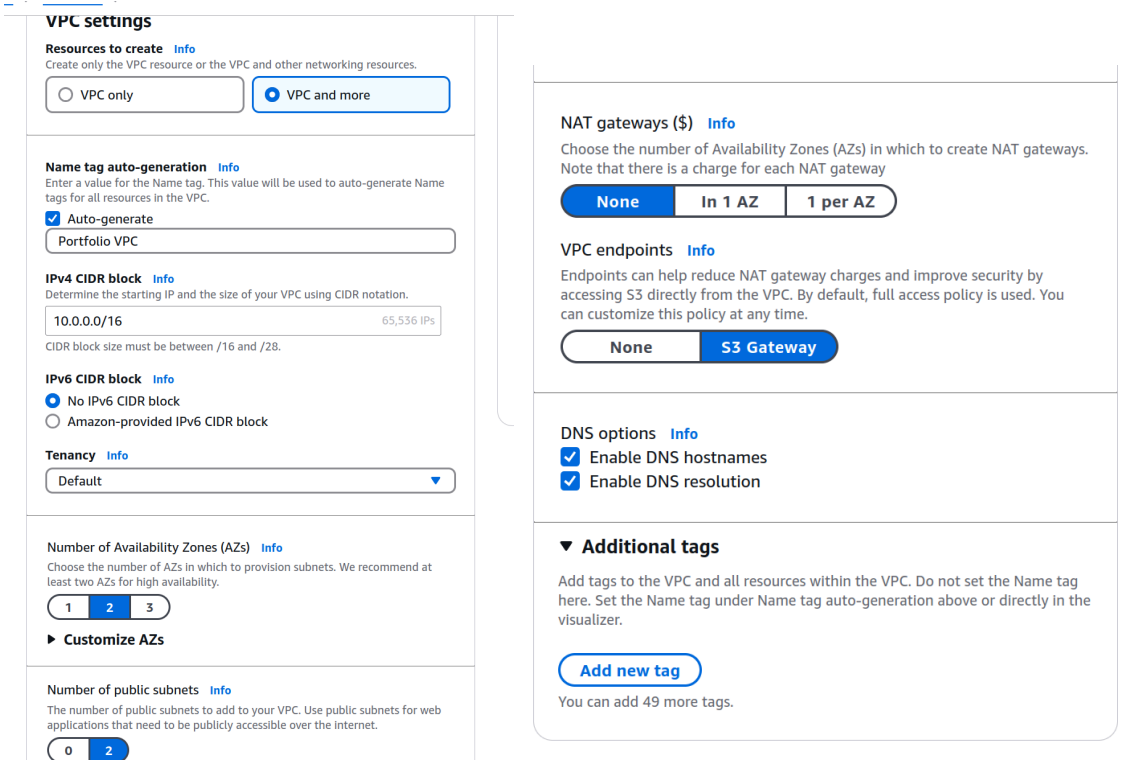




Delete the default VPC

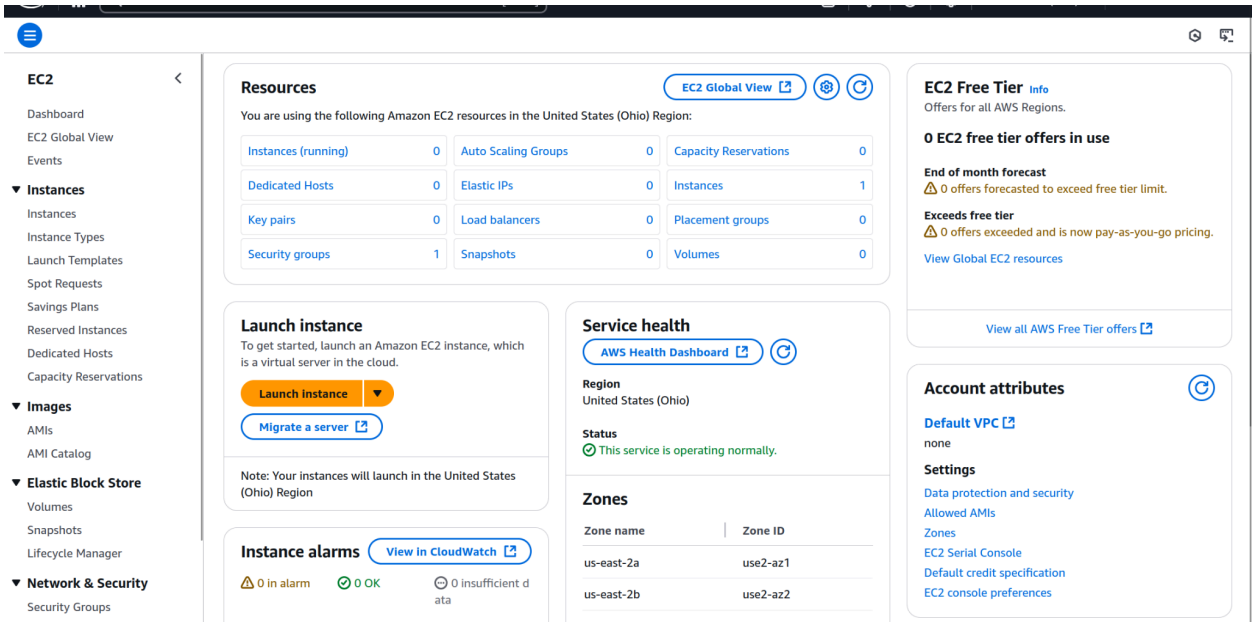


Now create a new one with these settings

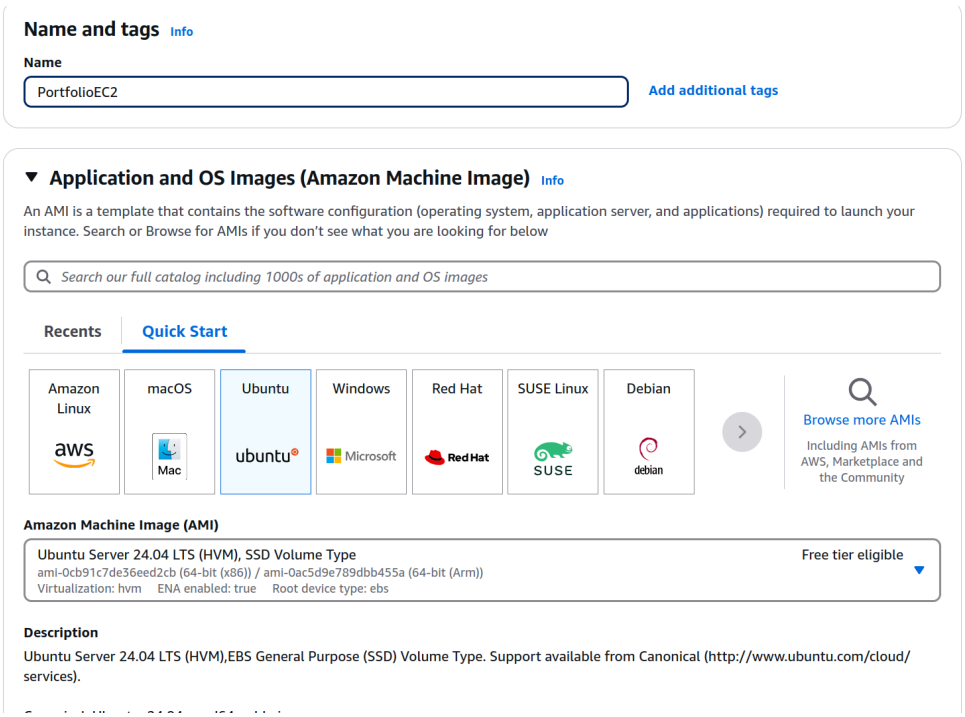


2) Making the EC2

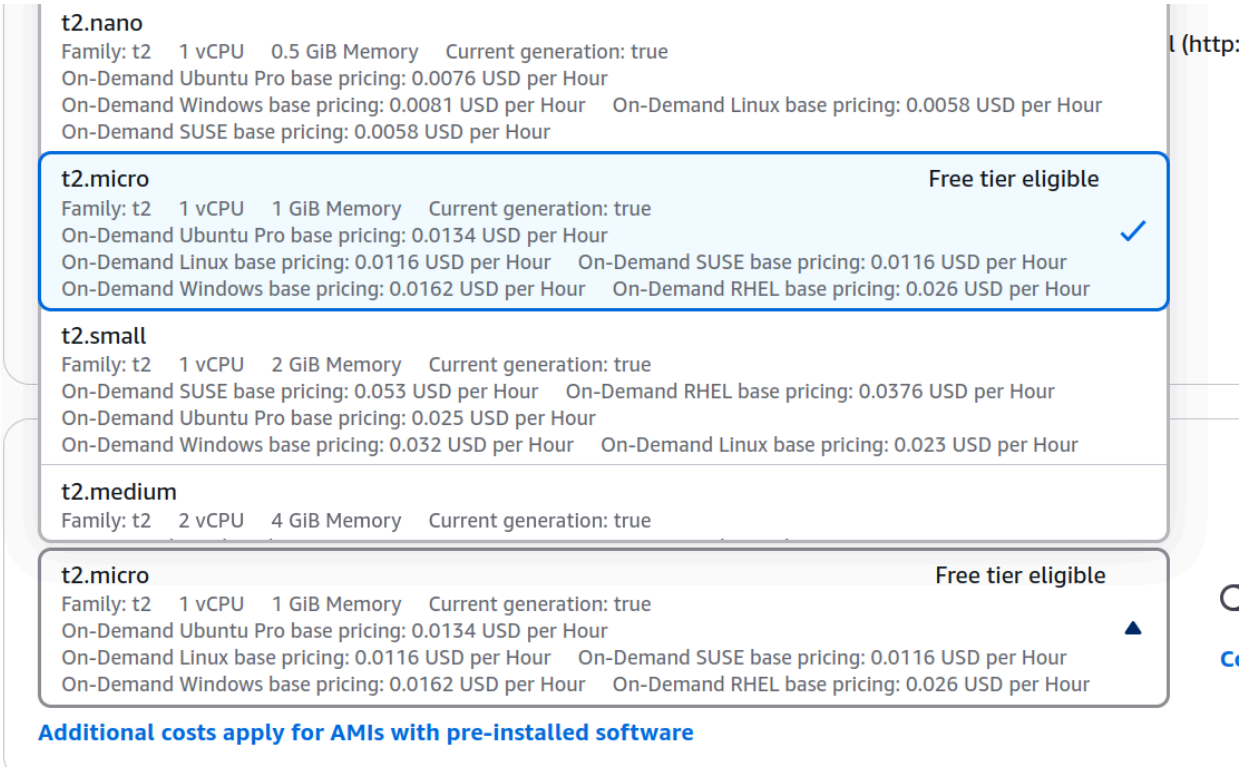
Now go to EC2 Dashboard and launch the instance



Name it and choose Ubuntu



Make sure free tier is selected

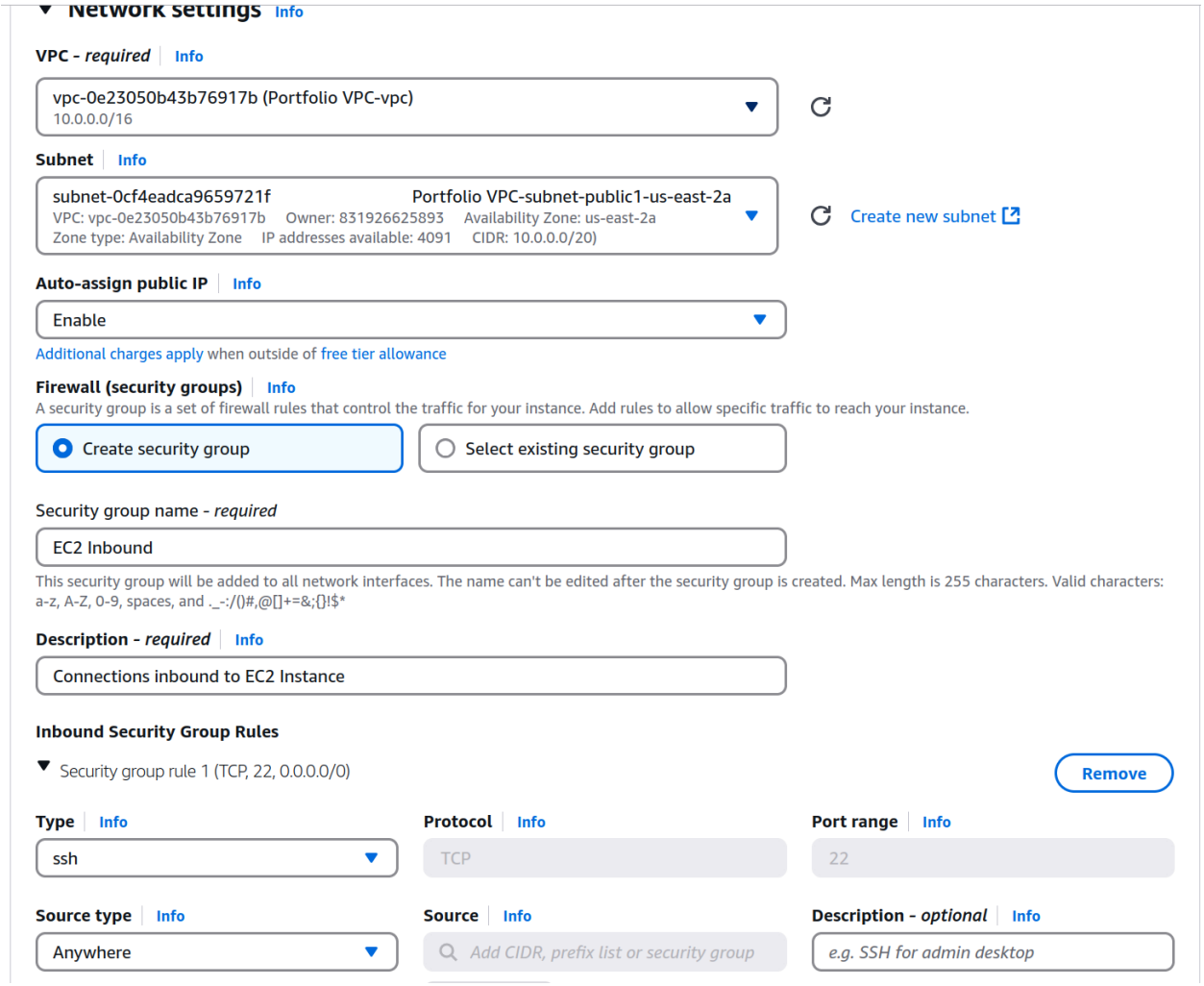


Choose no key pair

We will connect to it through EC2 Connect

Edit the network settings

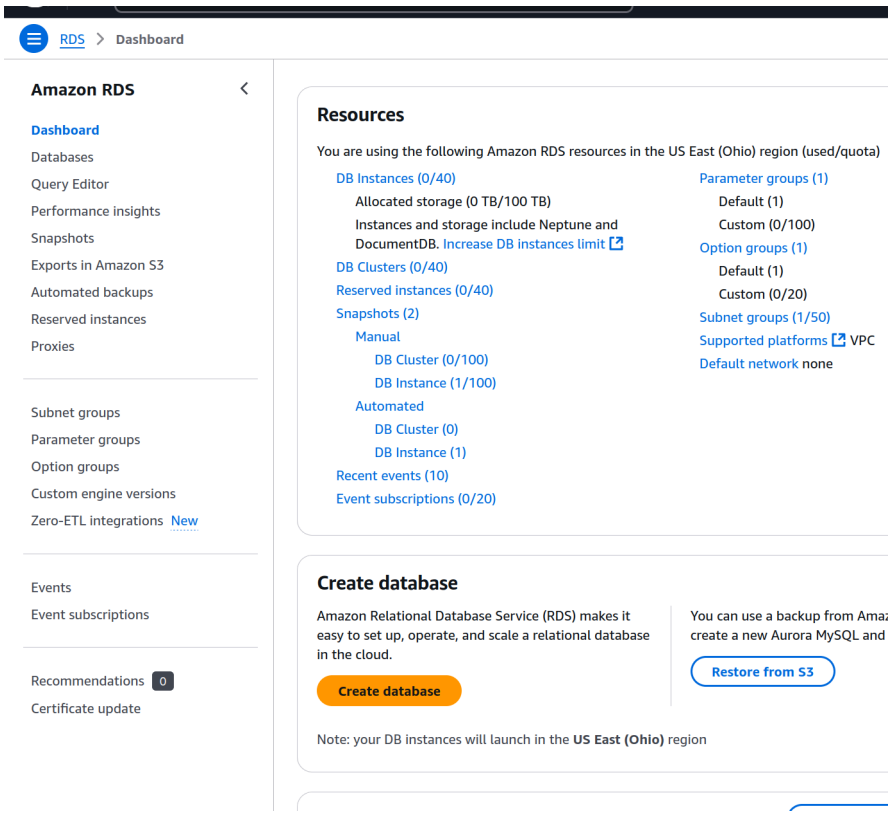
And Use these settings



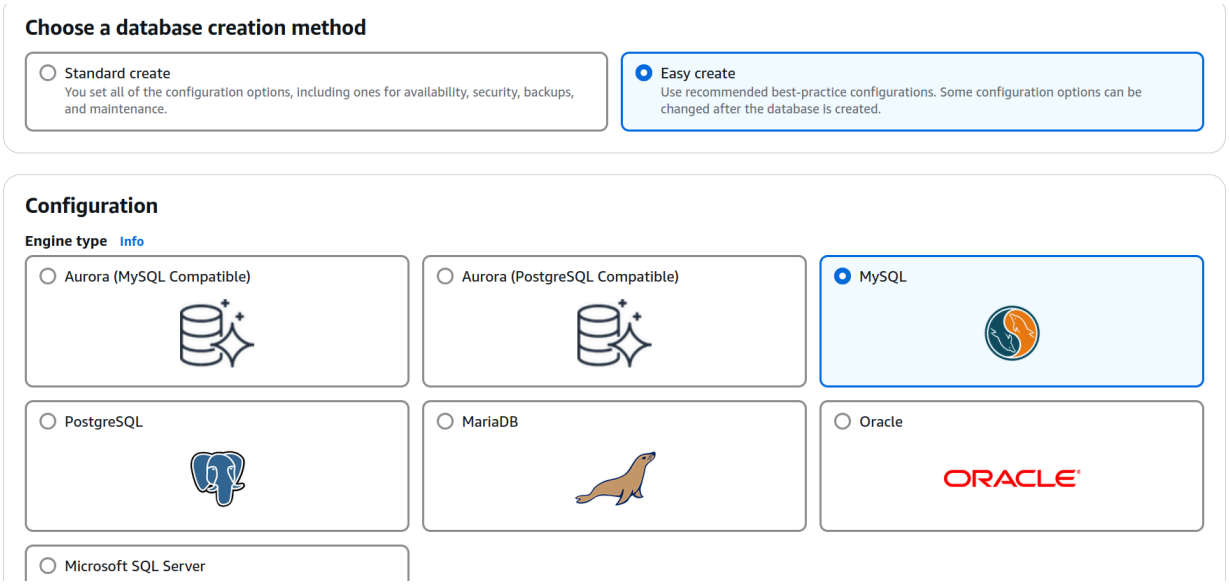
Now Launch instance and wait for it to load

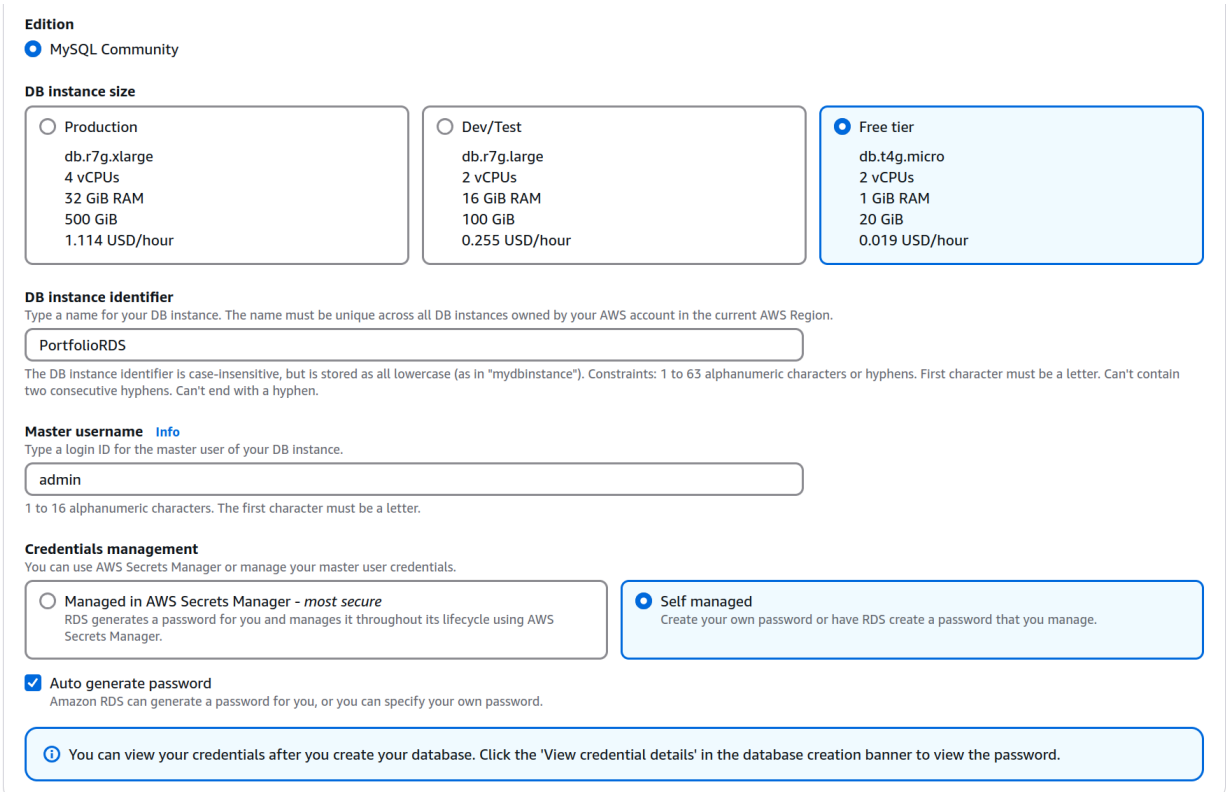
3) Making your RDS

Search up RDS (Should show something like Aurora and RDS) and go to the RDS Dashboard. Then click on create database

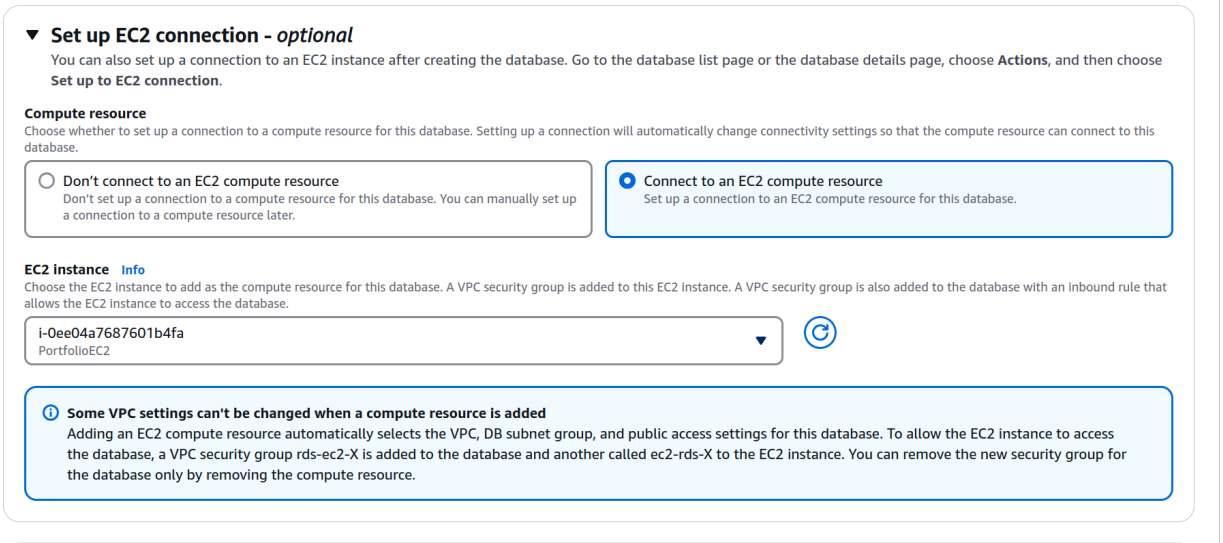


Use these settings

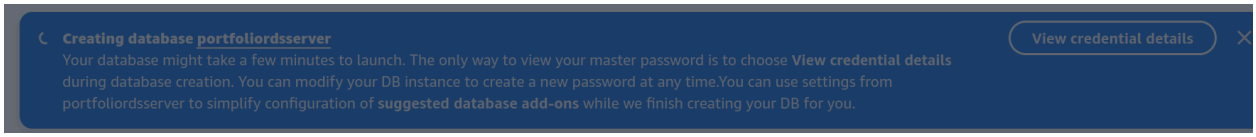


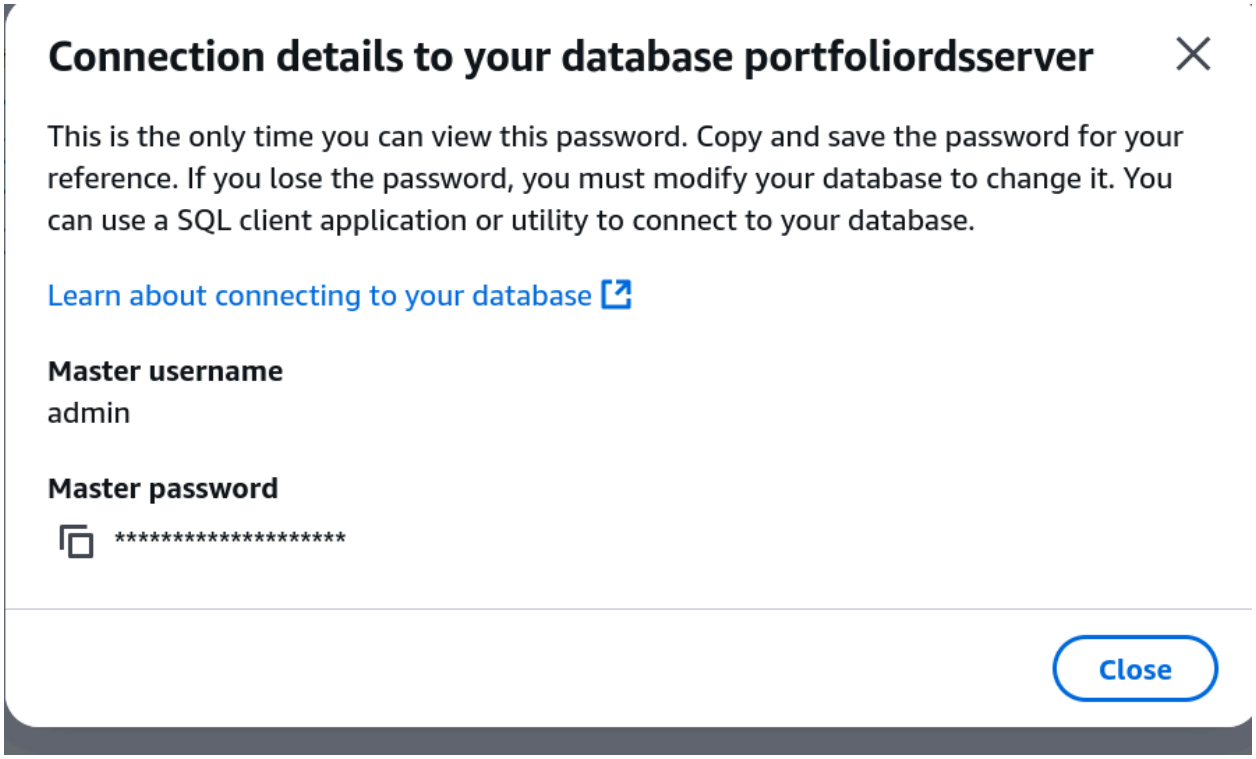


Make sure we set up an EC2 Connection to the one we just created



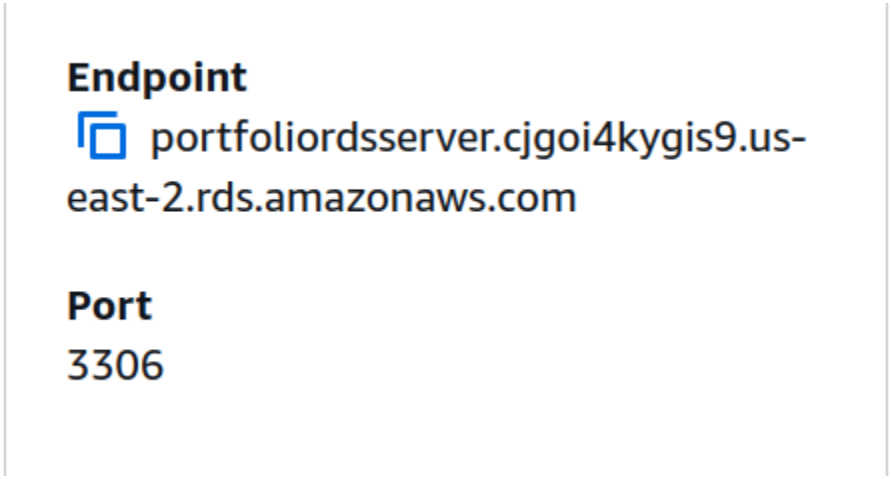
Then create the database. Make sure you view credentials after creating! This will disappear if you navigate away from the page. Copy password and save it somewhere (notepad or somewhere else). YOU WILL USE THIS FOR THE REST OF YOUR PROJECTS, SO STORE SOMEWHERE SAFE. Should look like “avwmKKEE7H5xtfQcEm9O”





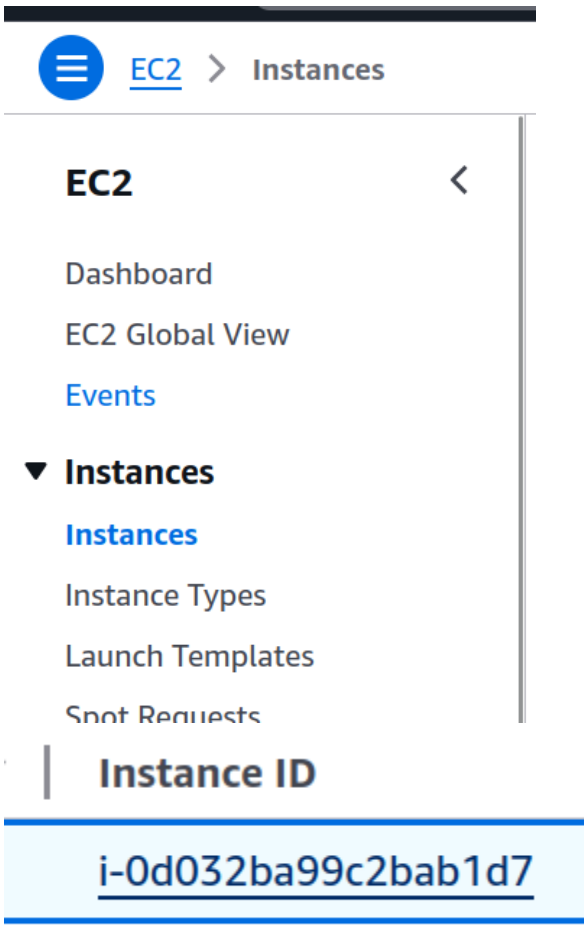
Wait for your RDS to finish creating. Then click on it

Copy the endpoint somewhere (notepad) Should look like “portfoliordsserver.cjgoi4kygis9.us-east-2.rds.amazonaws.com”

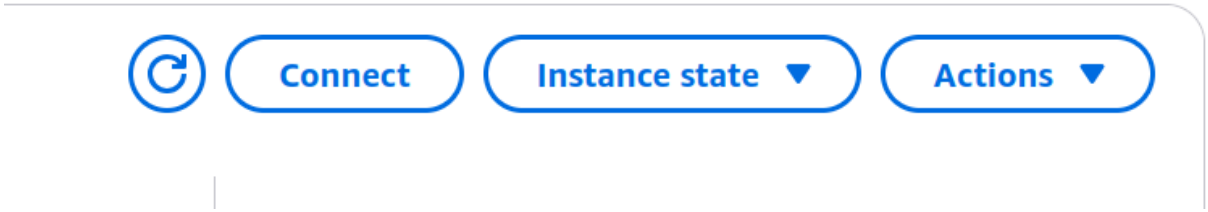


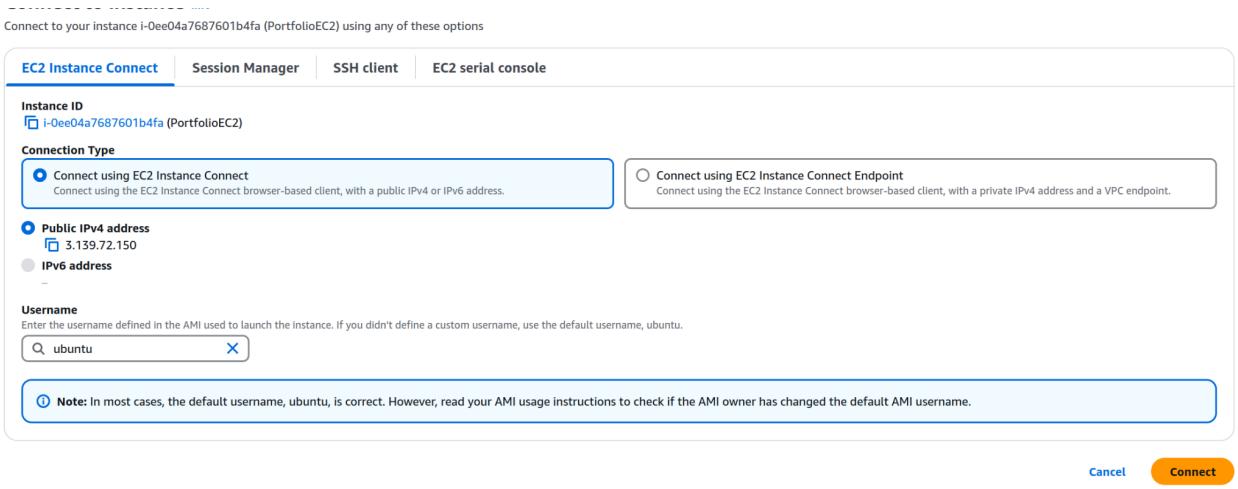
4) Connect to your EC2

Find your EC2 on the dashboard by going to instances. Click on the instance id that shows up.



Click on Connect





You should see this

```
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Thu Mar  6 05:39:56 UTC 2025

  System load:  0.16              Processes:             106
  Usage of /:   24.9% of 6.71GB   Users logged in:       0
  Memory usage: 22%               IPv4 address for enX0: 10.0.12.100
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-12-100:~$
```

Enter these commands (One at a time please):

Here we install updates to linux system. Then we install necessary python packages for the server. USE CTRL+SHIFT+V to paste in a terminal rather than CTRL+V

```
sudo apt update
```

```
sudo apt upgrade

sudo apt install python3-dotenv
sudo apt install python3-flask
sudo apt install python3-pymysql
```

Getting the code for our server

```
git clone https://github.com/K12-NYU-Center/CPath.git
cd CPath
cd "Part 2"
```

Making an .env file. This is to store important credentials

```
nano .env
```

paste this in. Remove the bracketed part and put your saved endpoint and password earlier:

```
DB_HOST={rds endpoint}
DB_USER=admin
DB_PASSWORD={password saved earlier}
DB_NAME=forms
DB_PORT=3306
```

Ctrl-x then enter to save

Test the server

Type in the command:

```
python3 rds.py
```

If it works with no errors, press Ctrl+c to quit it

5) Connect to your RDS MYSQL Server

More bash commands.

Install your mysql command line client. This let's us connect to the RDS server

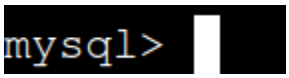
```
sudo apt install mysql-client
```

Now connect to the server. Use your saved endpoint earlier

```
mysql -h {endpoint} -u admin -p
```

It should prompt you for your password. Copy paste it in.

You should see this



MySQL Commands:

Enter these into the mysql server

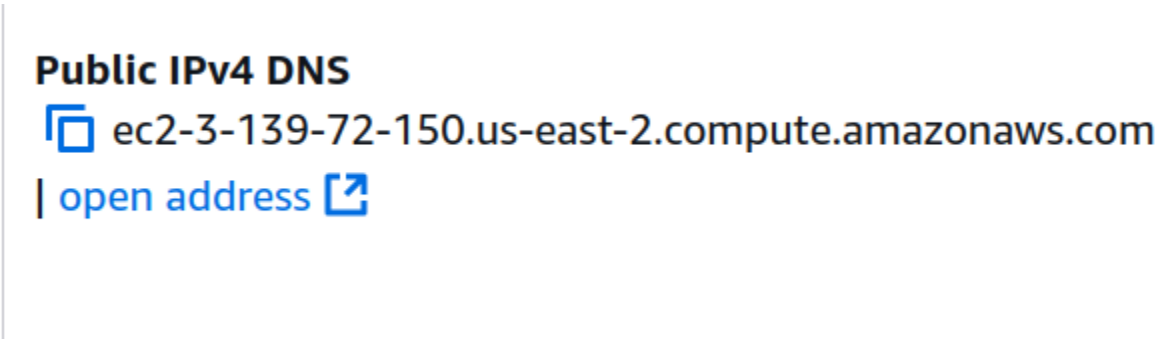
```
create database forms;
use forms;
CREATE TABLE form_submissions (
      id INT AUTO_INCREMENT PRIMARY KEY,
      name VARCHAR(255),
      subject VARCHAR(255),
      email VARCHAR(255),
      message TEXT,
      created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Now type quit to quit the mysql server

Setting up access from the web

Try connecting your EC2 Instance from web

Try connecting. Remember to add port :5000 to end of it, because we are using port :5000 for the server. Example: https://ec2-3-139-82-150.us-east-2.compute.amazonaws.com:5000

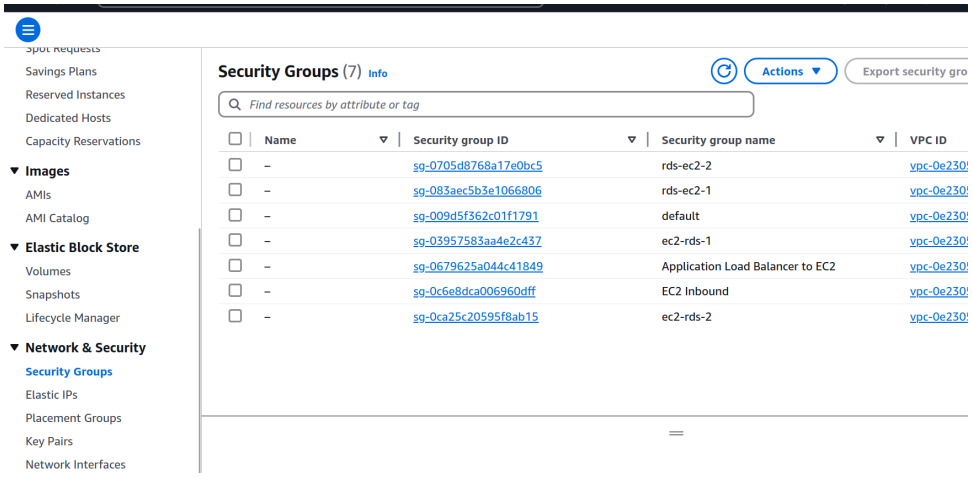


It doesn’t work (ignore that the picture doesn’t have port 5000)

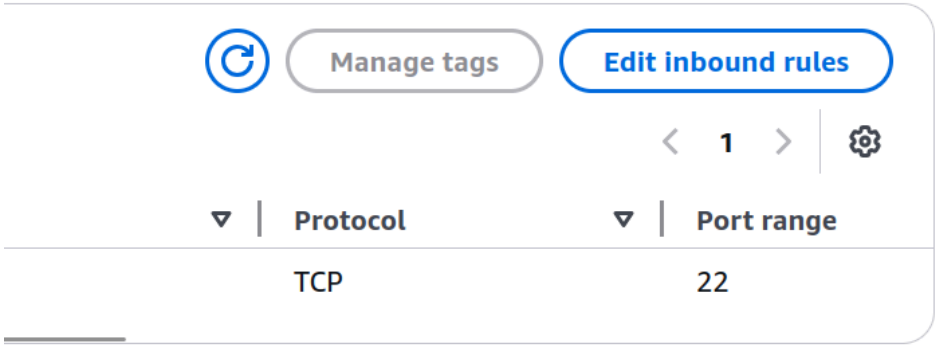
Editing the security group

We need to allow access from the web for our EC2 with security groups. It’s not enough to just have it in a public subnet.

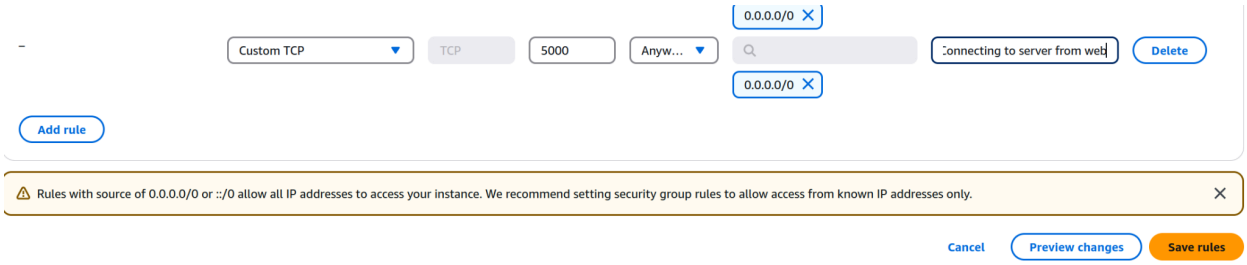
Click on your subnet from earlier (EC2 Inbound)



Edit the Inbound rules



Add this rule



Putting it all together

Make sure your server is running

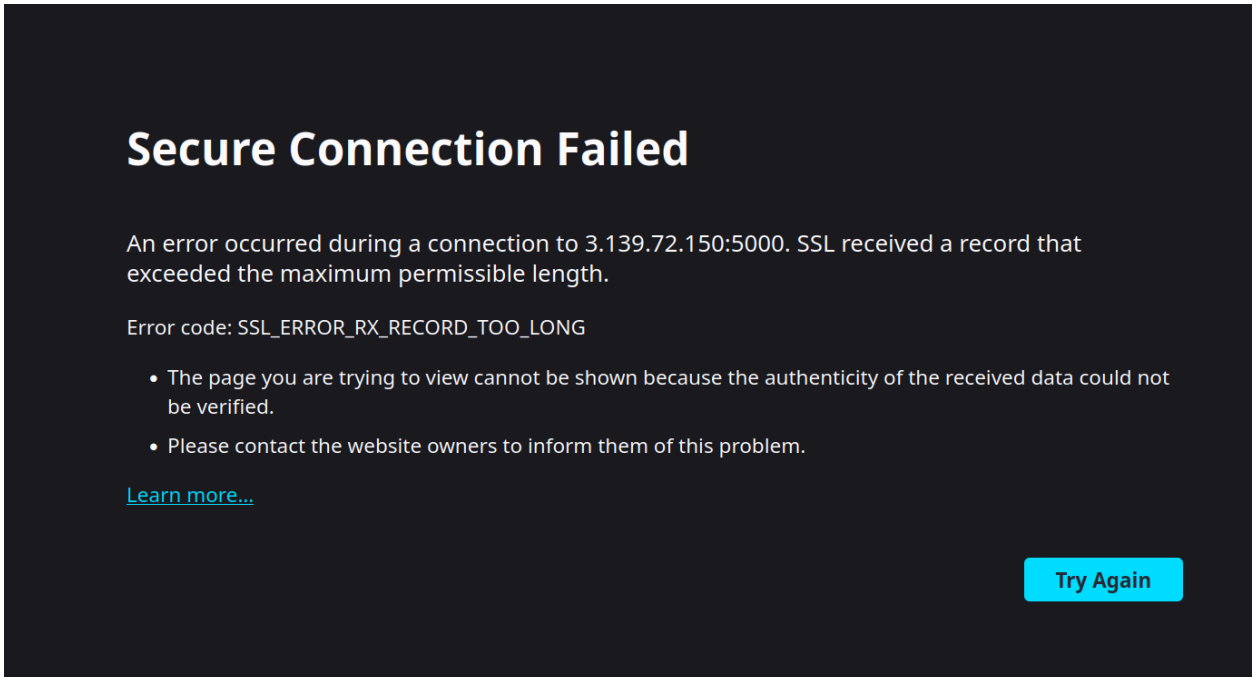
Go to your EC2 and start up the server again.
If you closed the tab, here's the command To navigate to the folder (condensed from earlier section):

```
cd /home/ubuntu/CPath/"Part 2"
```

```
^Cubuntu@ip-10-0-12-100:~/CPath/Part 2$ python3 rds.py
 * Serving Flask app 'rds'
 * Debug mode: off
INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://10.0.12.100:5000
INFO:werkzeug:Press CTRL+C to quit
```

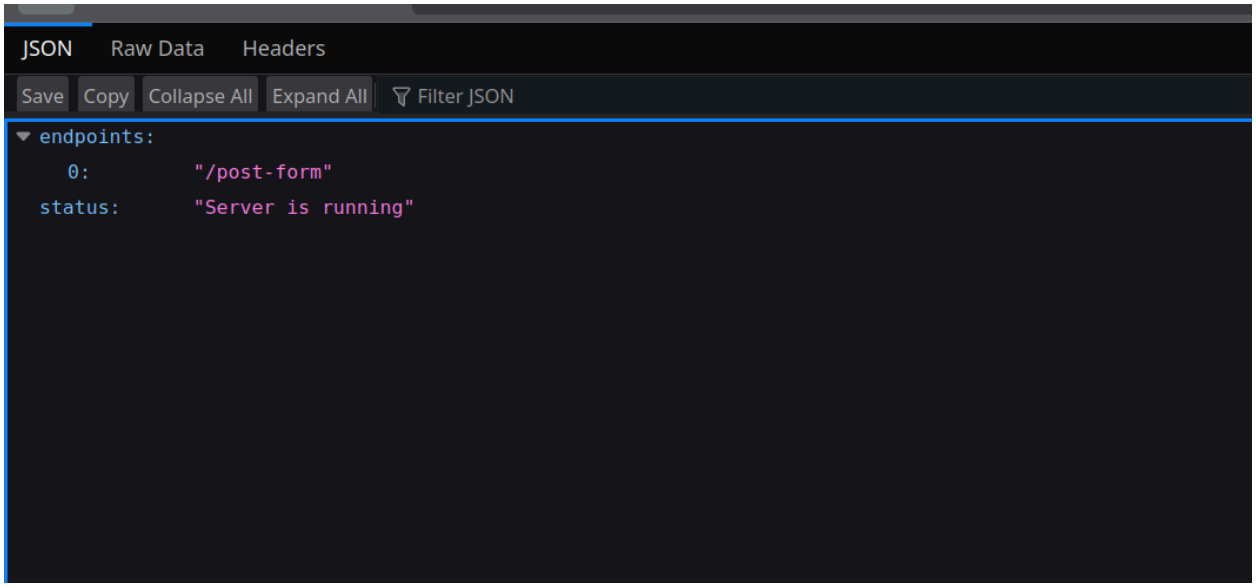
Now connect again

If you see this

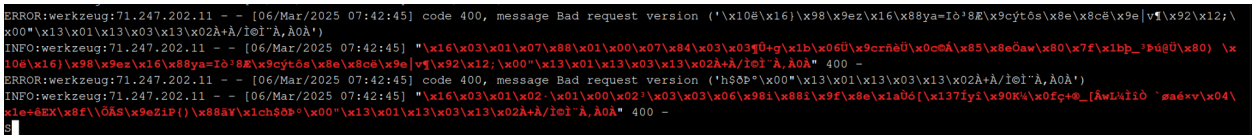


Go to the url and replace https with http. We don't have SSL yet.

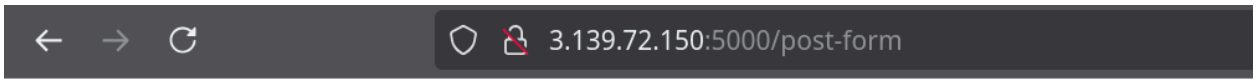
Success!



If you see something like this in your logs. Don’t worry. It’s just when you try to initiate https connection with http server



Try going to /post-form



Method Not Allowed

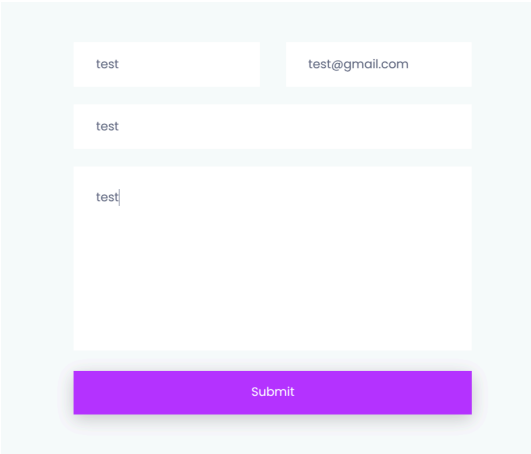
The method is not allowed for the requested URL.

Not allowed, because we are making a GET request from url. We need to use it on our website instead.

Send a form from our website

Add the attached form code to your website somewhere. Replace the action with the post-form url.
Example: http://ec2-3-139-72-150.us-east-2.compute.amazonaws.com:5000/post-form

Send the form (Remember not to send anything sensitive. HTTP is unencrypted)



(If you’re using the brownie website, click on the submit letters. The button is slightly broken)

You should see something like this

Let’s see our data in the database.

Press ctrl+C on your EC2 to exit the flask server. Connect to your RDS again

Now lets see our table

```
Use forms;
select * from form_submissions;
```

```
mysql> use forms;
Database changed
mysql> select * from form_submissions;
+----+------+---------+----------------+---------+---------------------+
| id | name | subject | email          | message | created_at          |
+----+------+---------+----------------+---------+---------------------+
|  1 | test | test    | test@gmail.com | test    | 2025-03-06 07:45:38 |
+----+------+---------+----------------+---------+---------------------+
1 row in set (0.00 sec)

mysql>
```

Success!