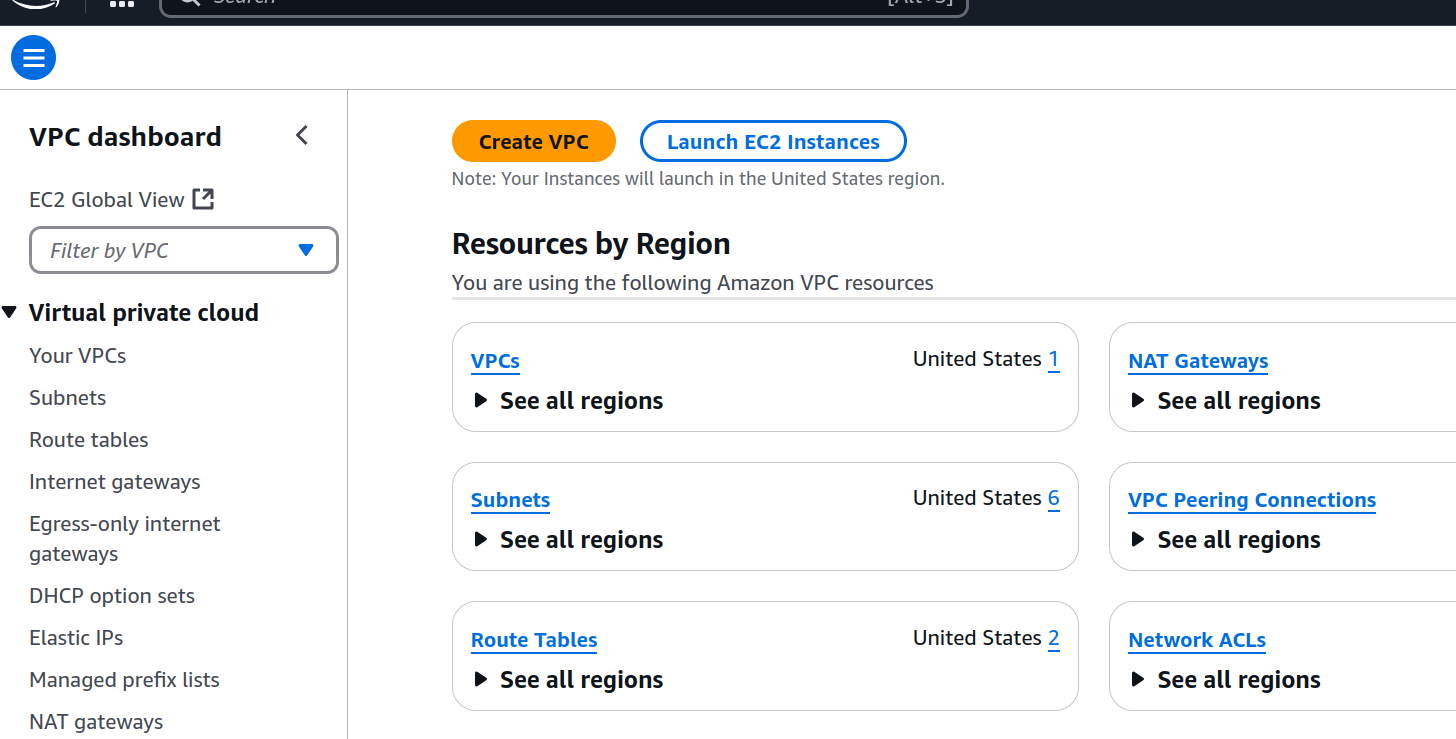
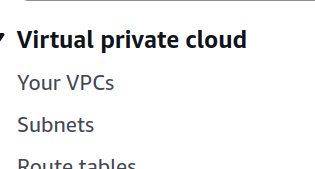
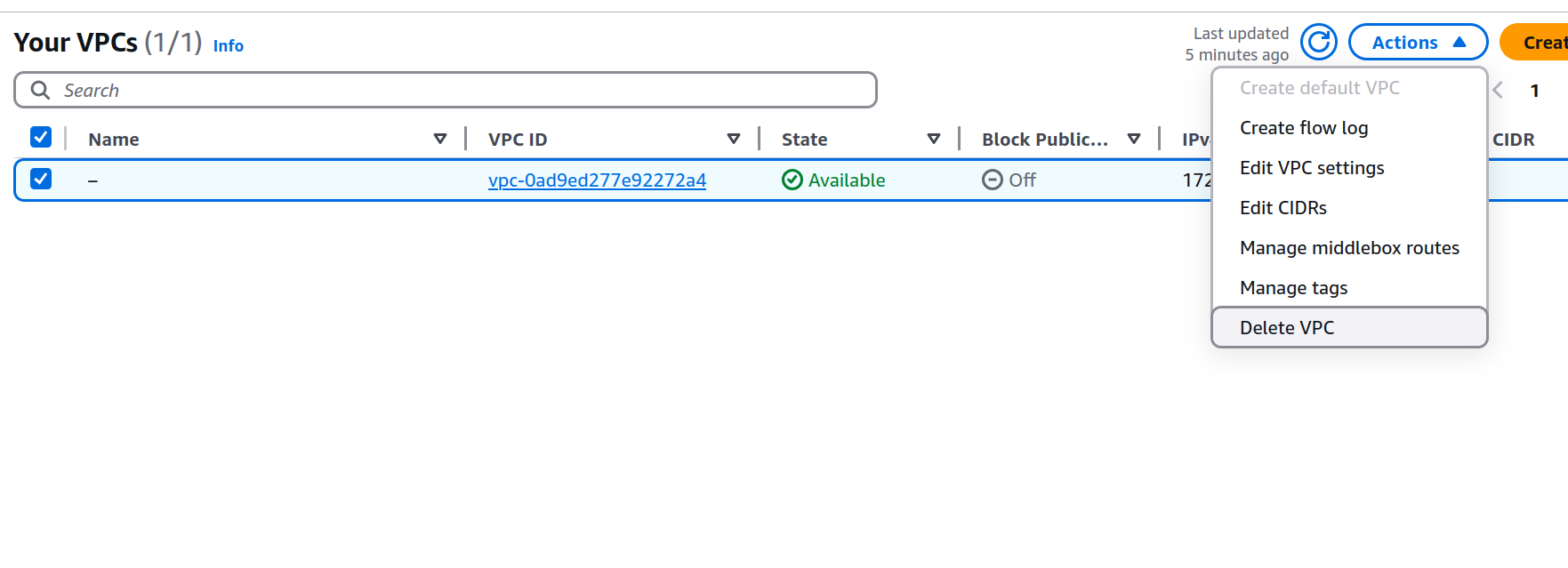
## Make VPC

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

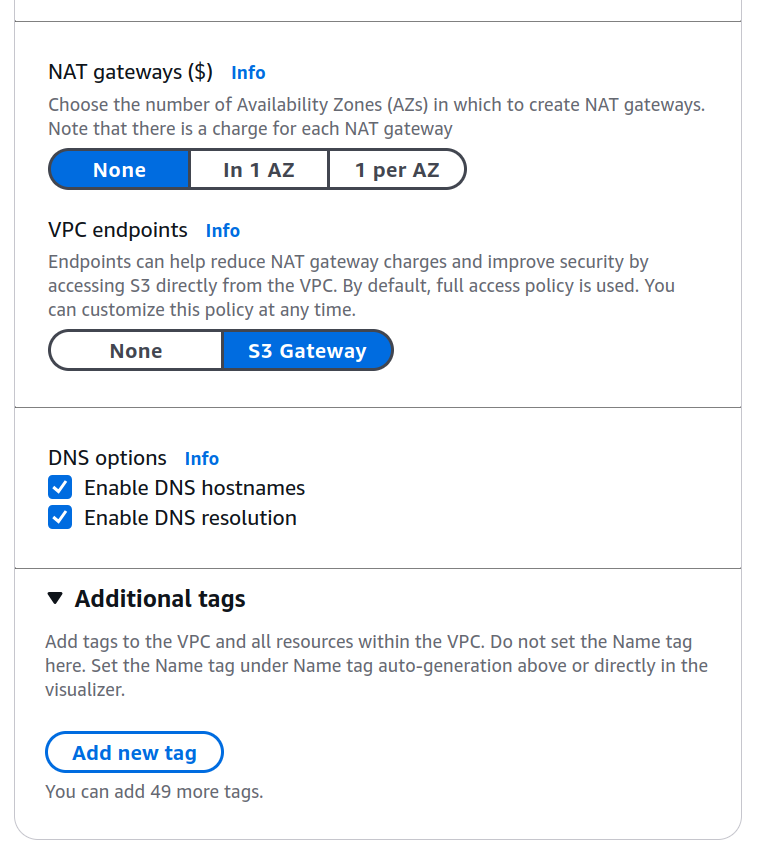
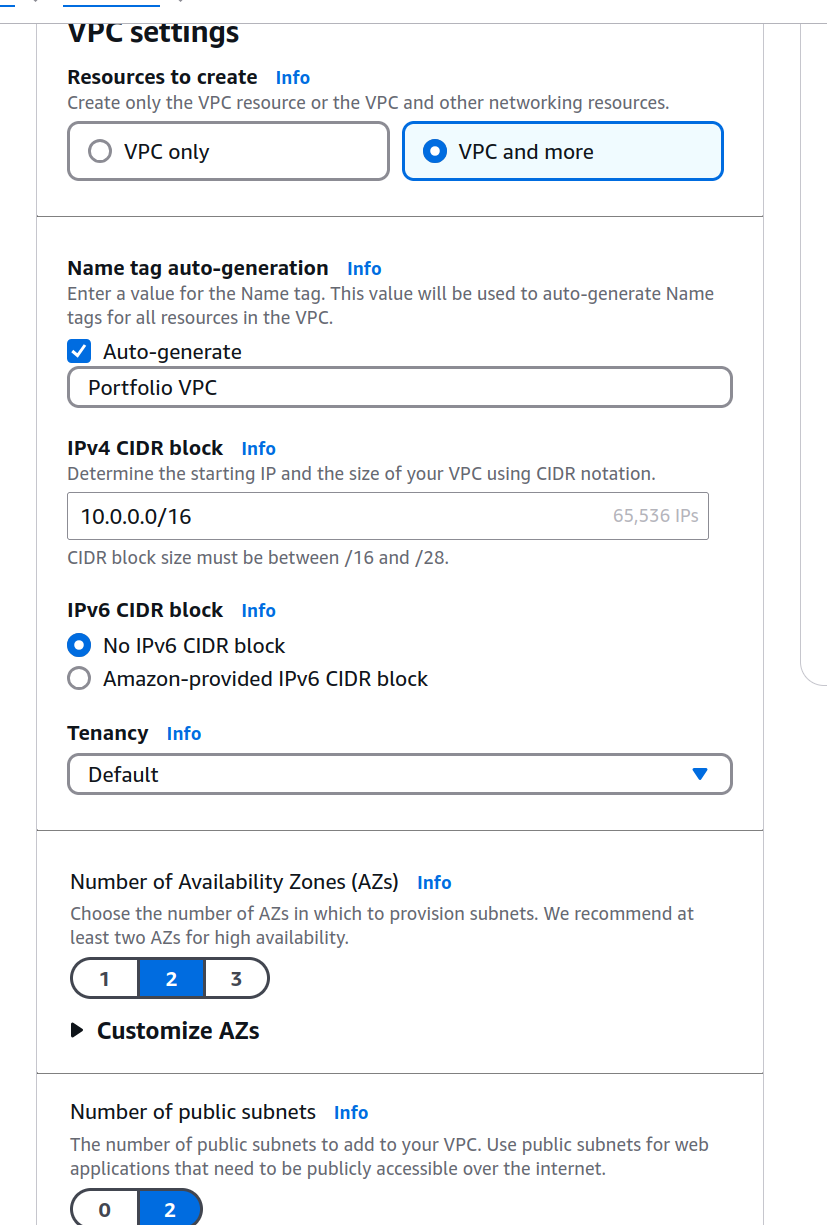




### Delete the default VPC

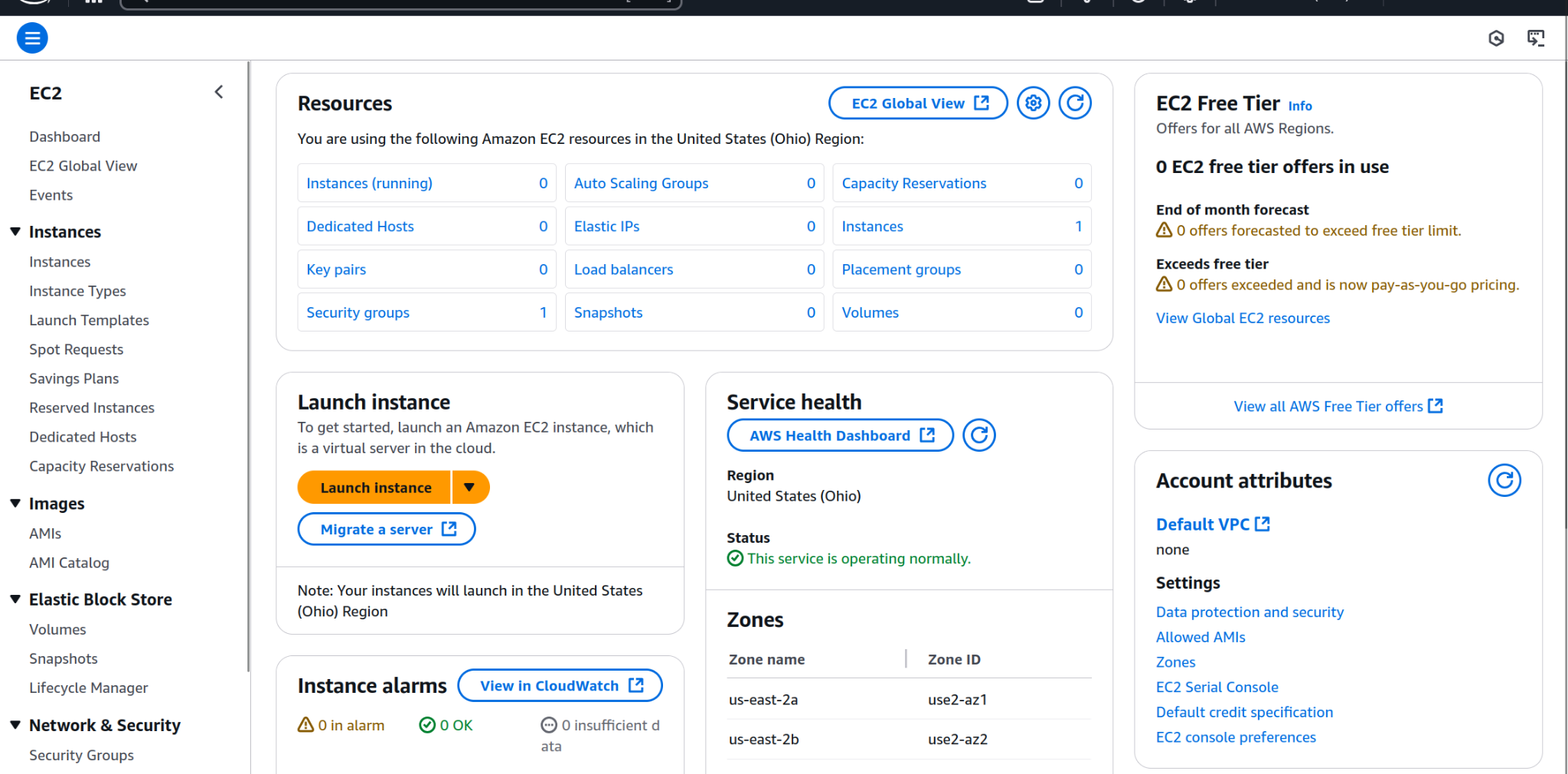


### Now create a new one with these settings

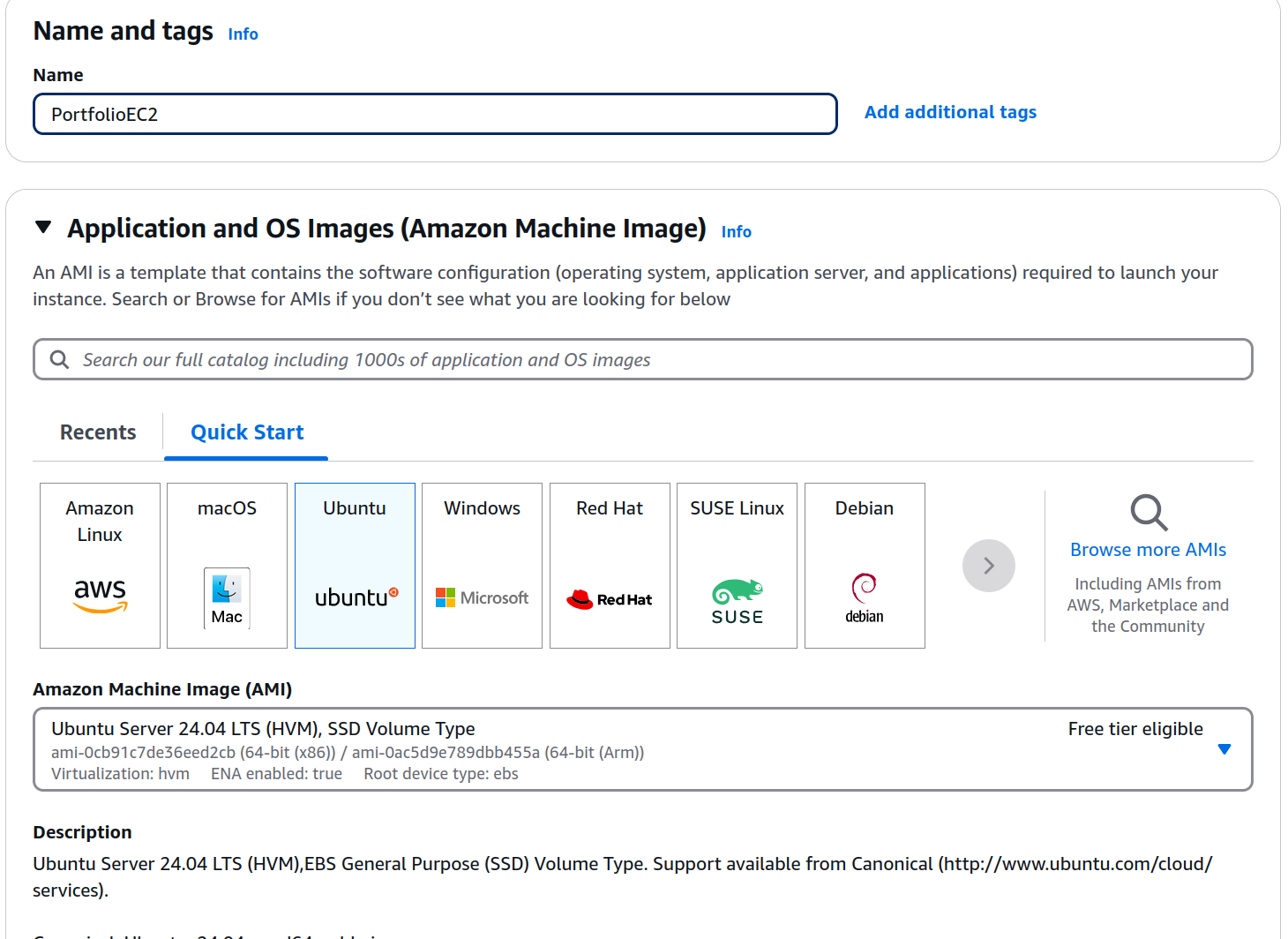


## 2) Making the EC2

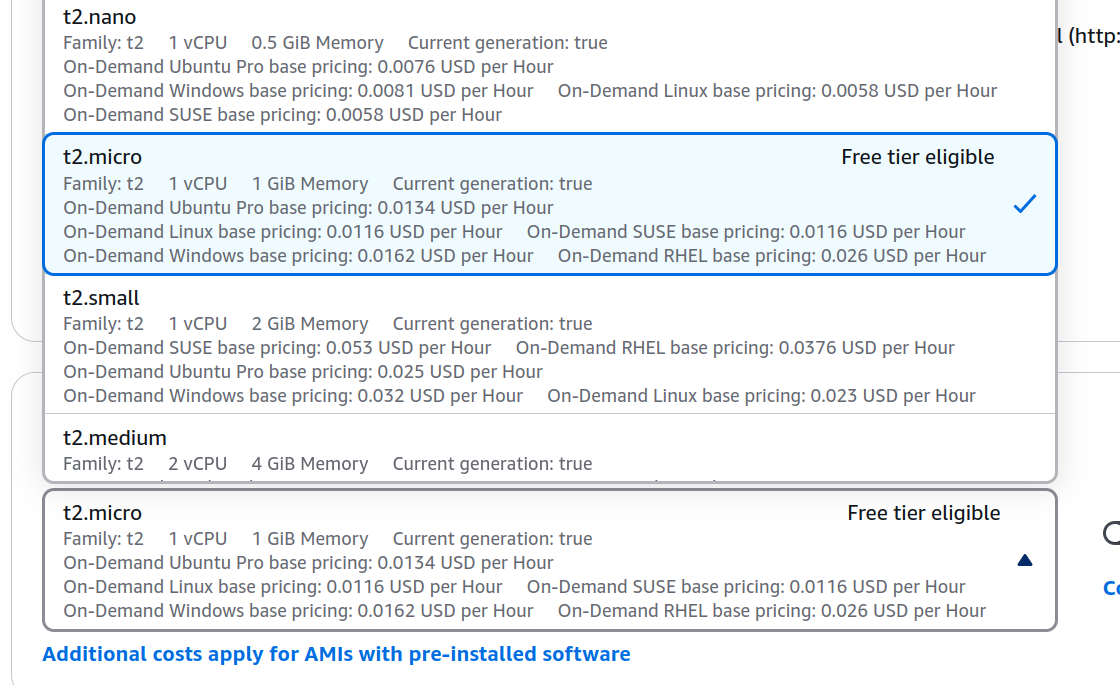
### Now go to EC2 Dashboard and launch the instance



### Name it and choose Ubuntu



### Make sure free tier is selected



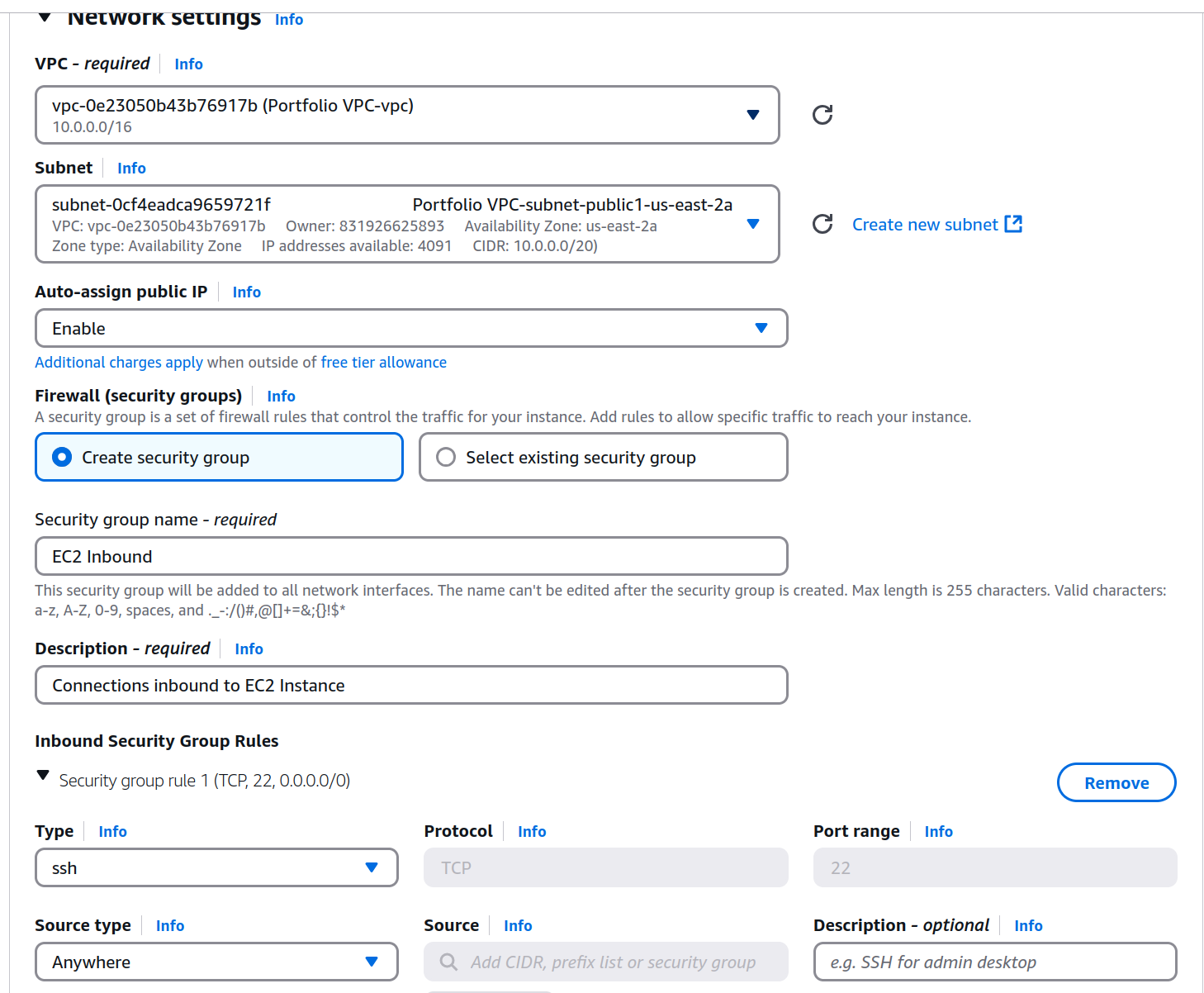
### Choose no key pair

We will connect to it through EC2 Connect

### Edit the network settings



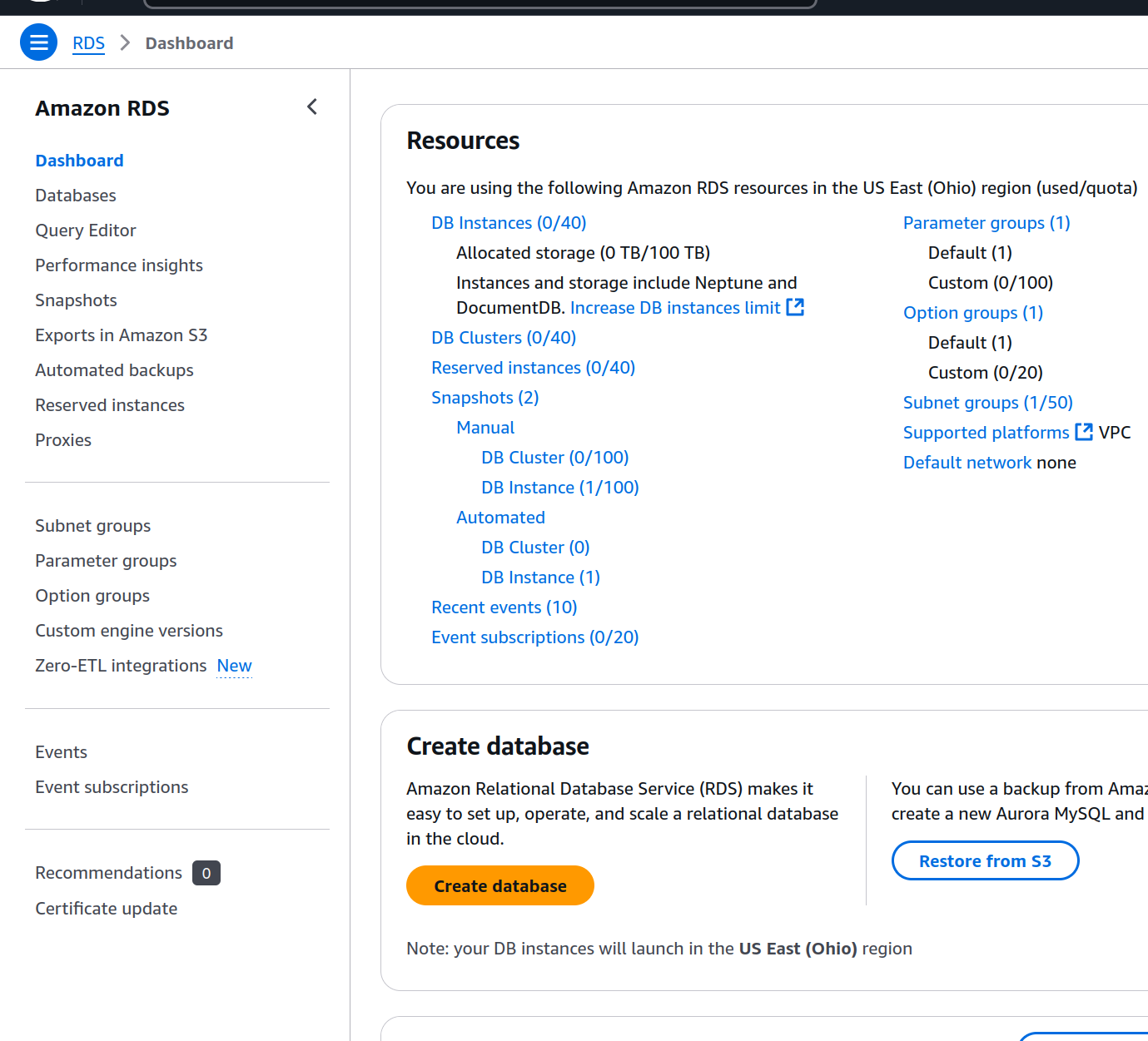
### And Use these settings



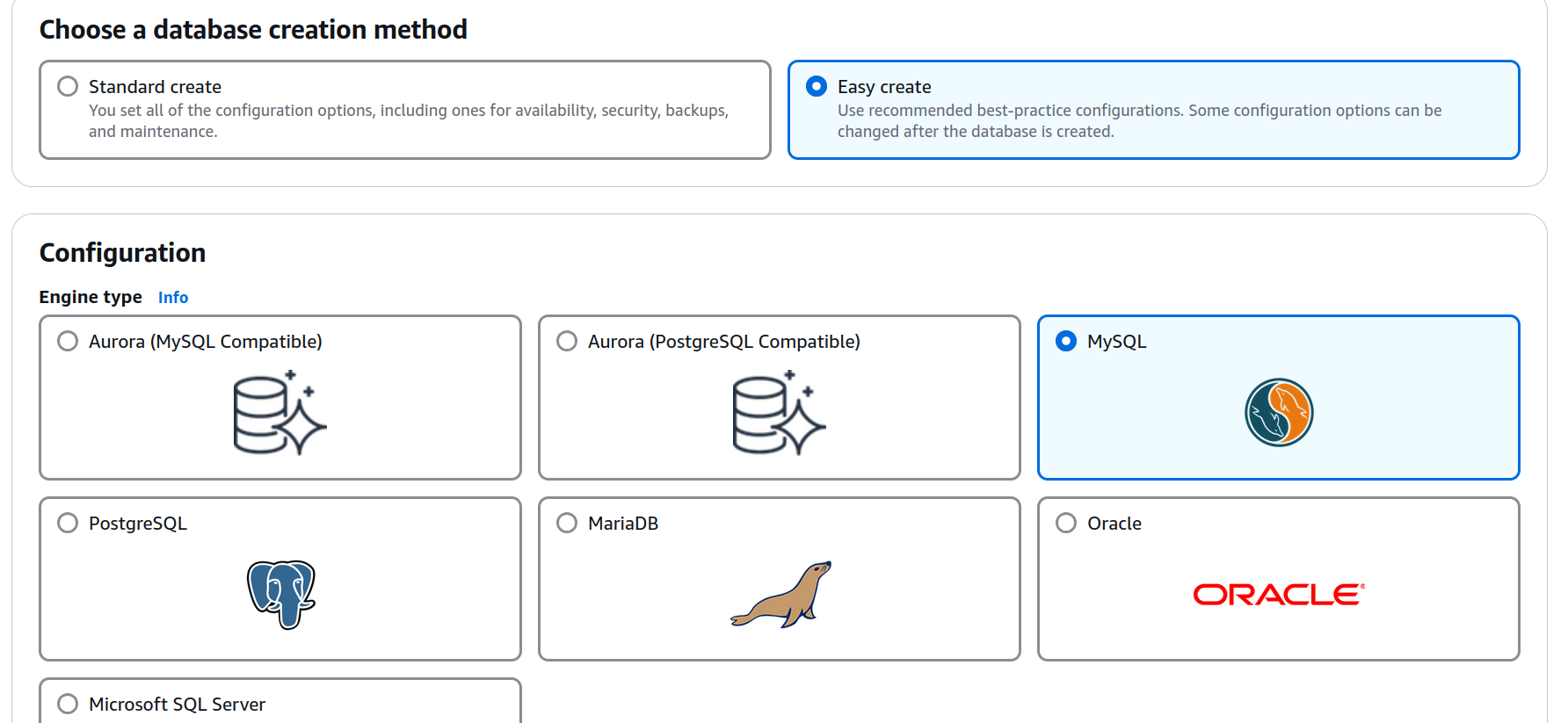
Now Launch instance and wait for it to load

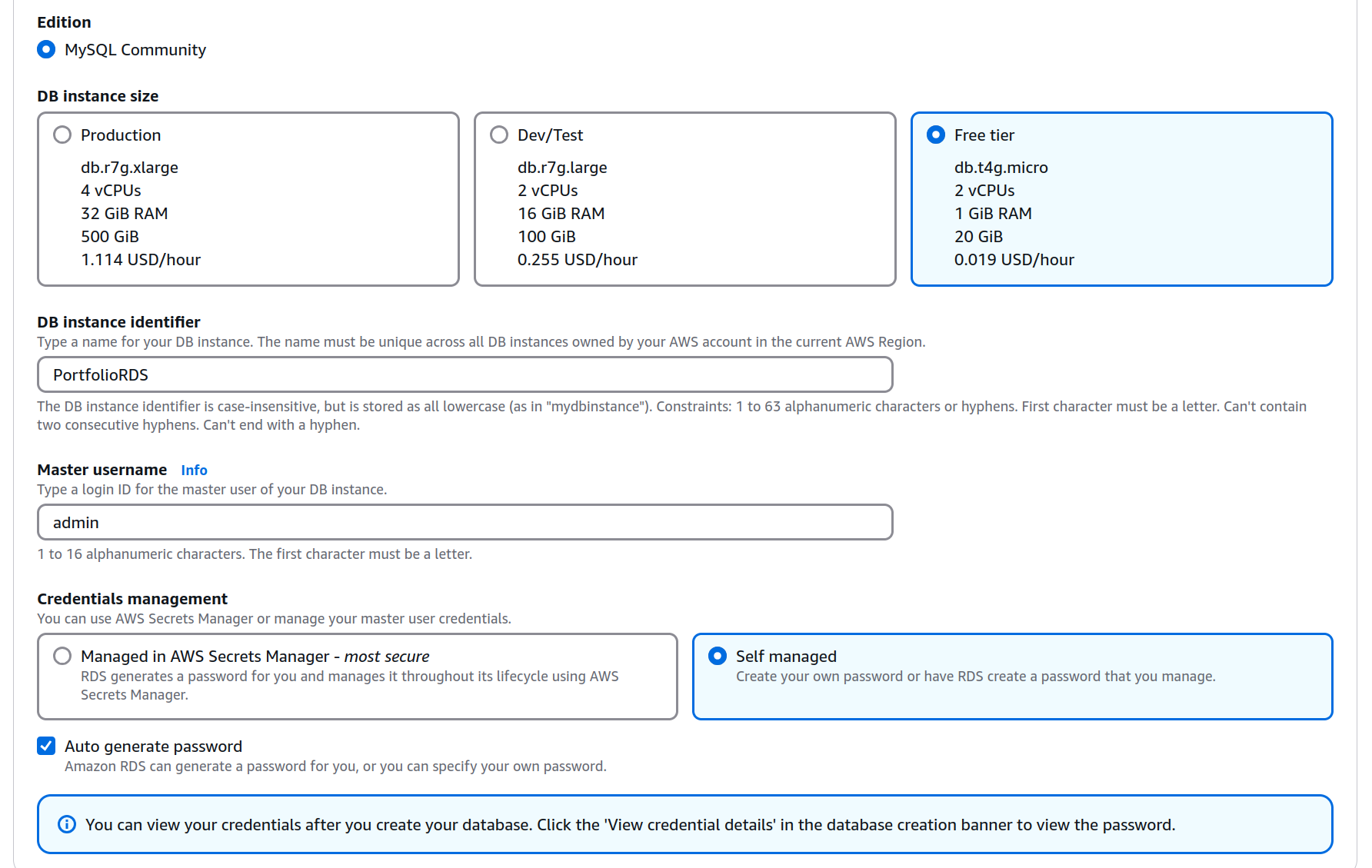
## 3) Making your RDS

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

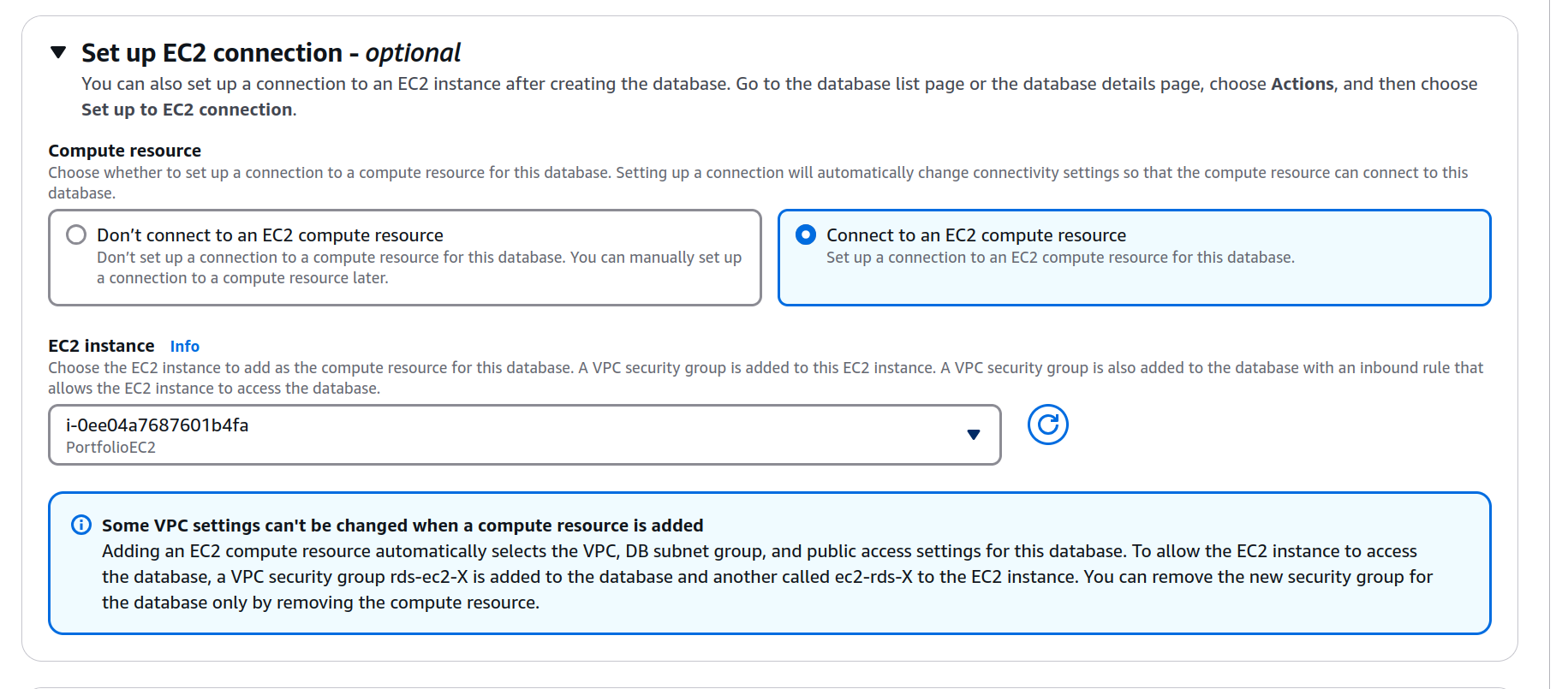


### Use these settings

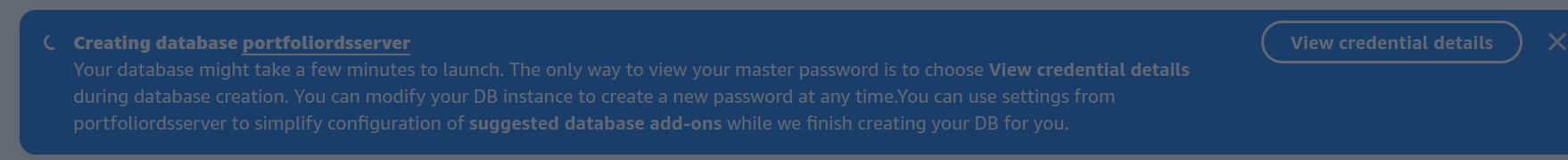


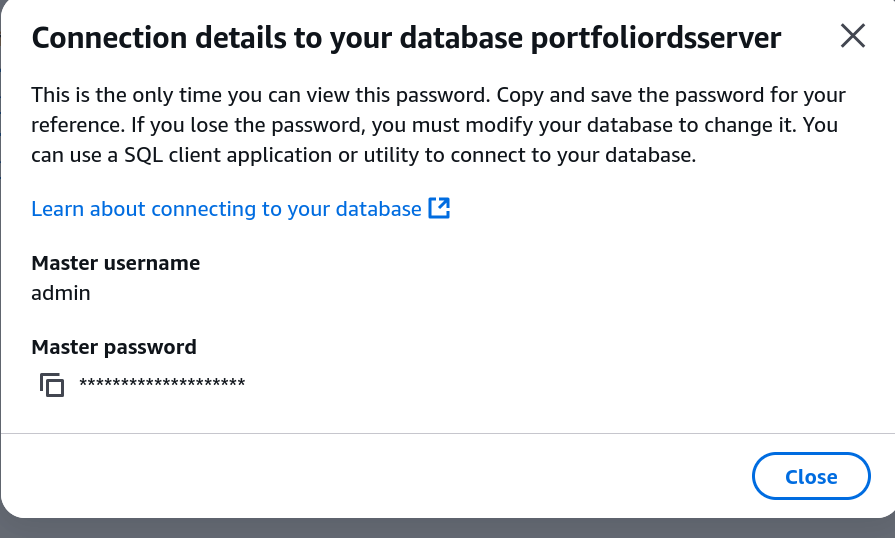


Make sure we set up an EC2 Connection to the one we just created

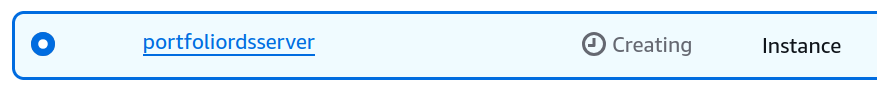


### Then create the database. Make sure you view credentials after creating! This will disappear if you navigate away from the page. Copy password and save it somewhere (notepad or somewhere else). YOU WILL USE THIS FOR THE REST OF YOUR PROJECTS, SO STORE SOMEWHERE SAFE. Should look like “avwmKKEE7H5xtfQcEm9O”





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

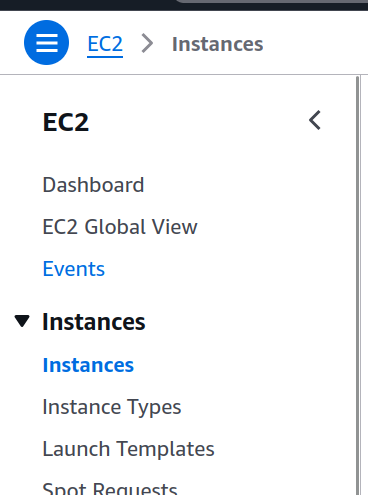


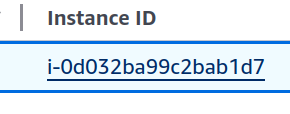
### Copy the endpoint somewhere (notepad) Should look like “portfoliordsserver.cjgoi4kygis9.us-east-2.rds.amazonaws.com”



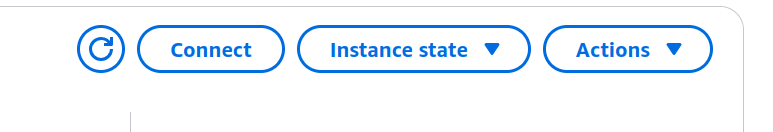
## 4) Connect to your EC2

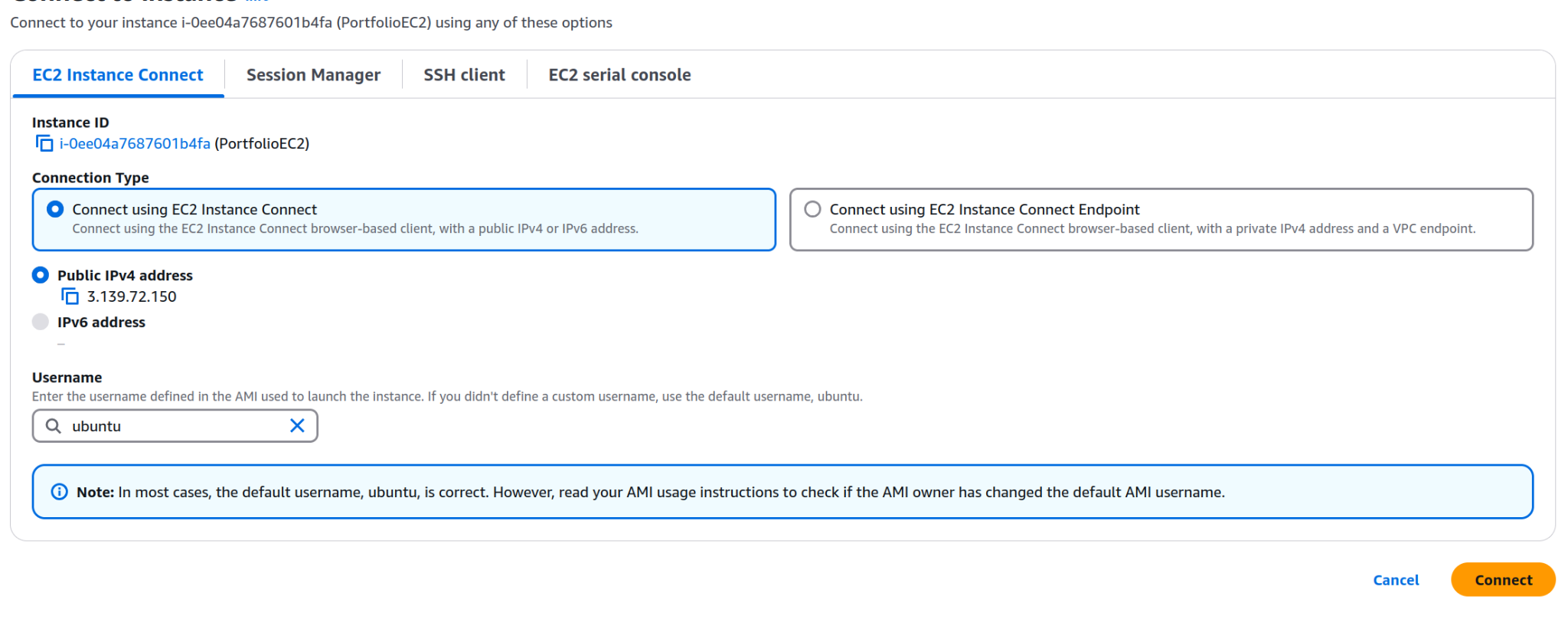
### Find your EC2 on the dashboard by going to instances. Click on the instance id that shows up.



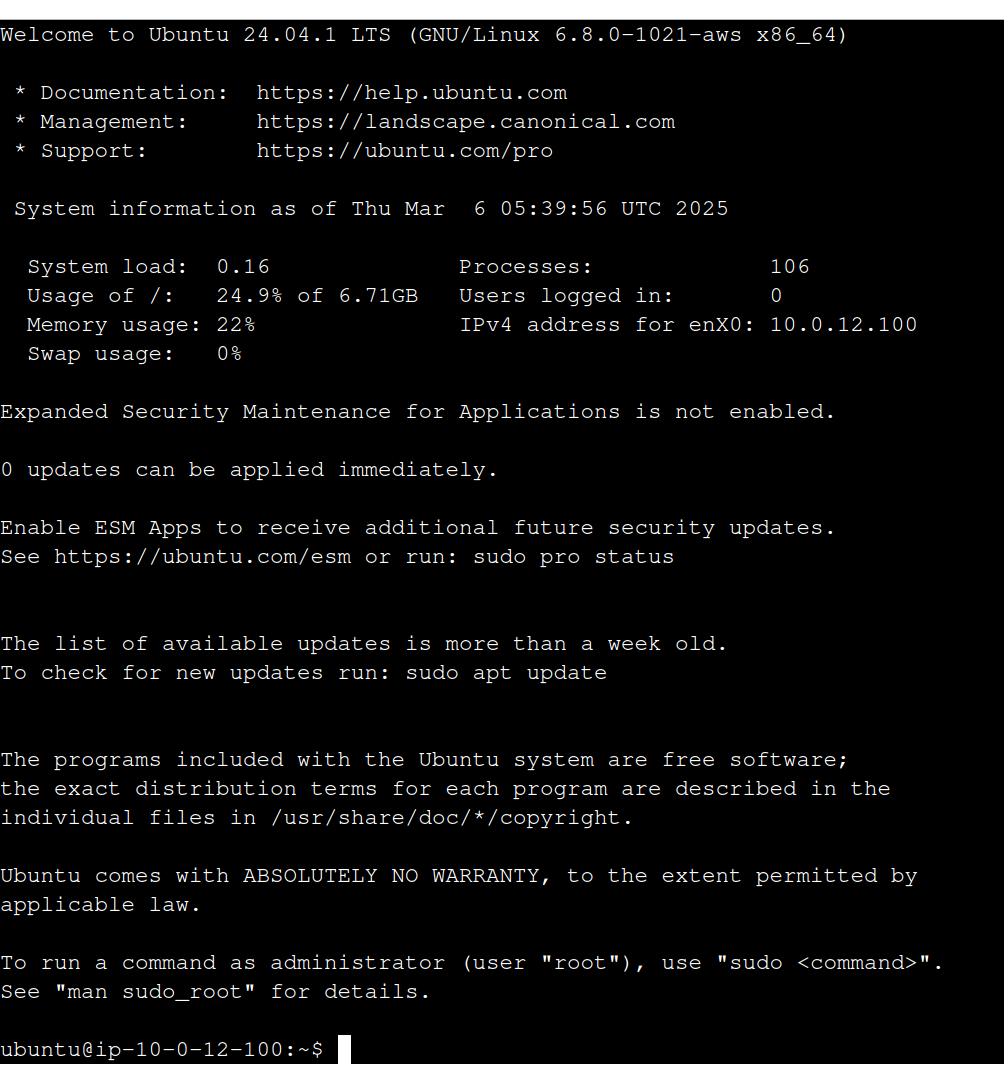


### Click on Connect





You should see this



### 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

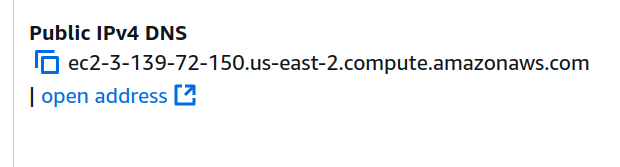
| create database forms; use forms; CREATE TABLE form\_submissions (  id INT AUTO\_INCREMENT PRIMARY KEY,  name VARCHAR(255),  subject VARCHAR(255),  email VARCHAR(255),  message TEXT,  created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ); |
| --- |

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



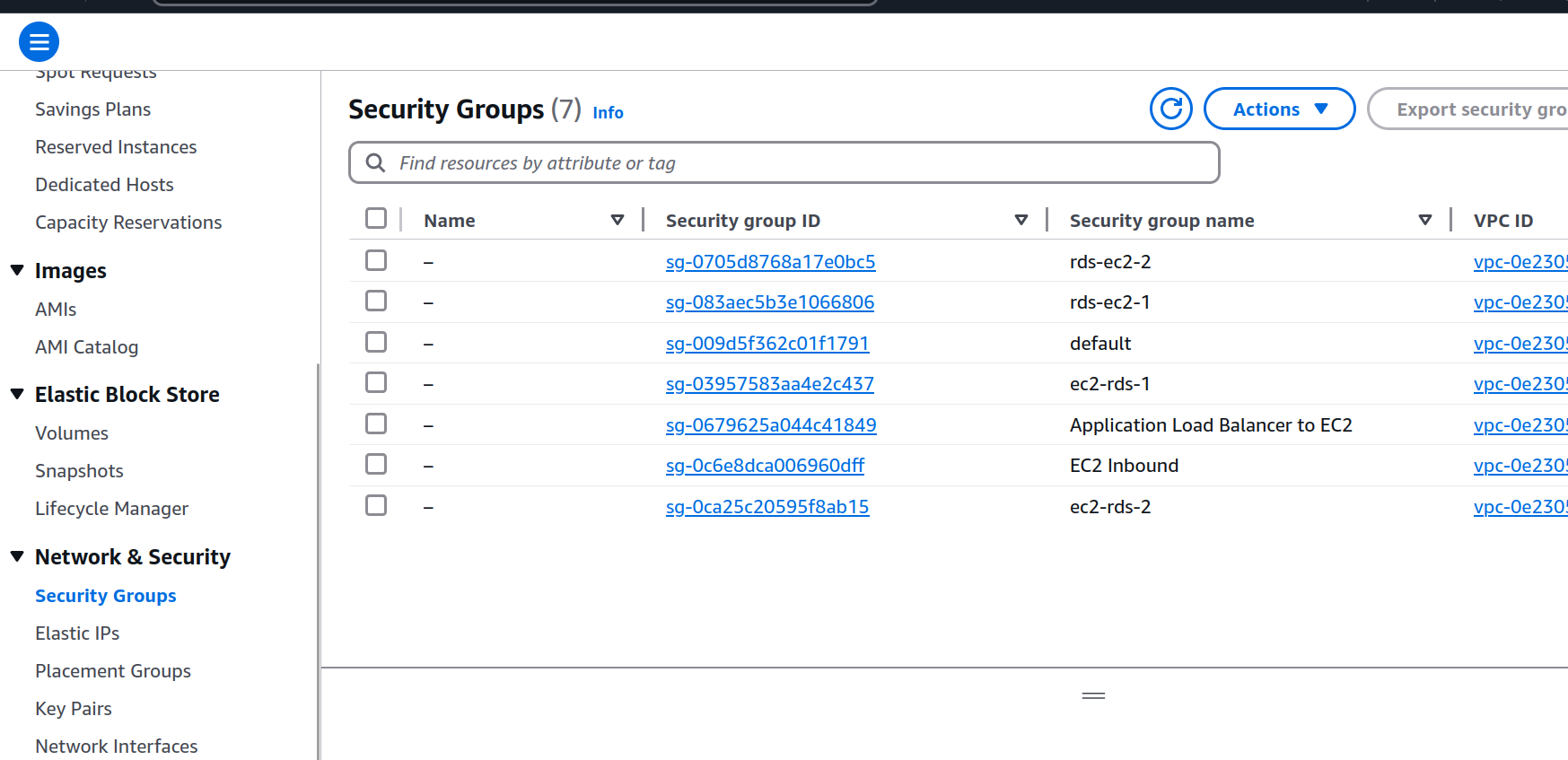
It doesn’t work (ignore that the picture doesn’t have port 5000)



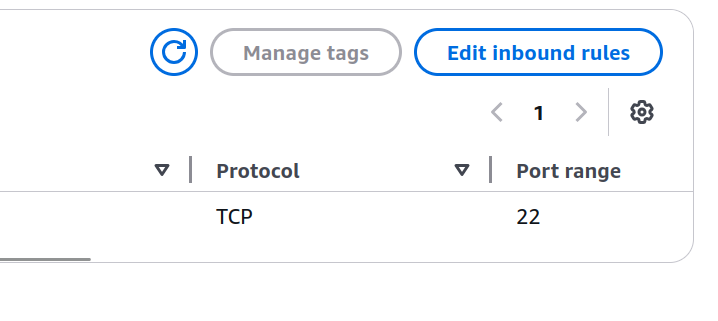
### Editing the security group

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.

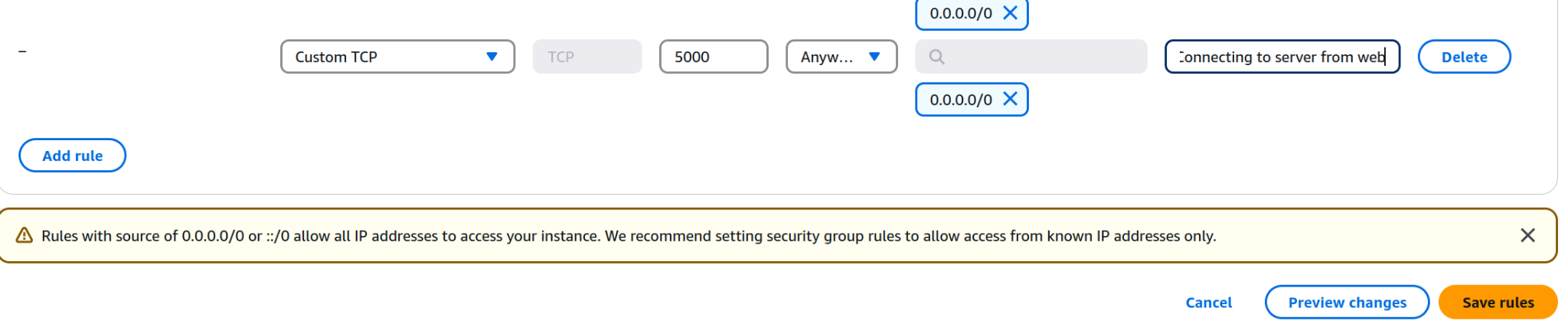
Click on your subnet from earlier (EC2 Inbound)



### Edit the Inbound rules



### Add this rule



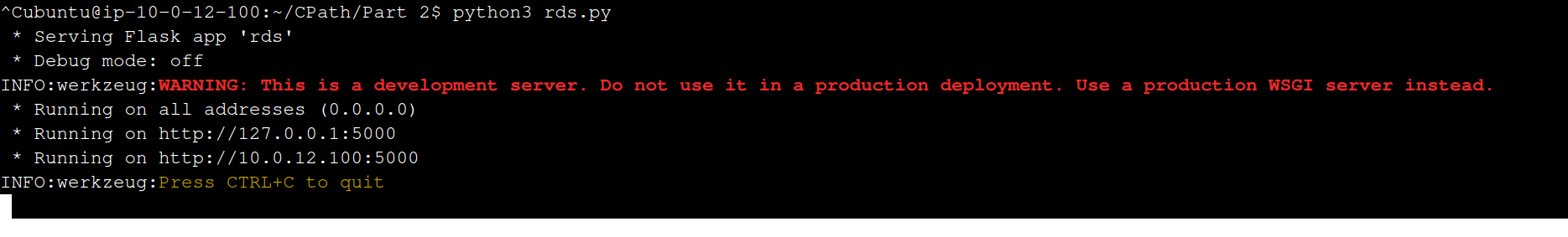
## Putting it all together

### Make sure your server is running

Go to your EC2 and start up the server again.

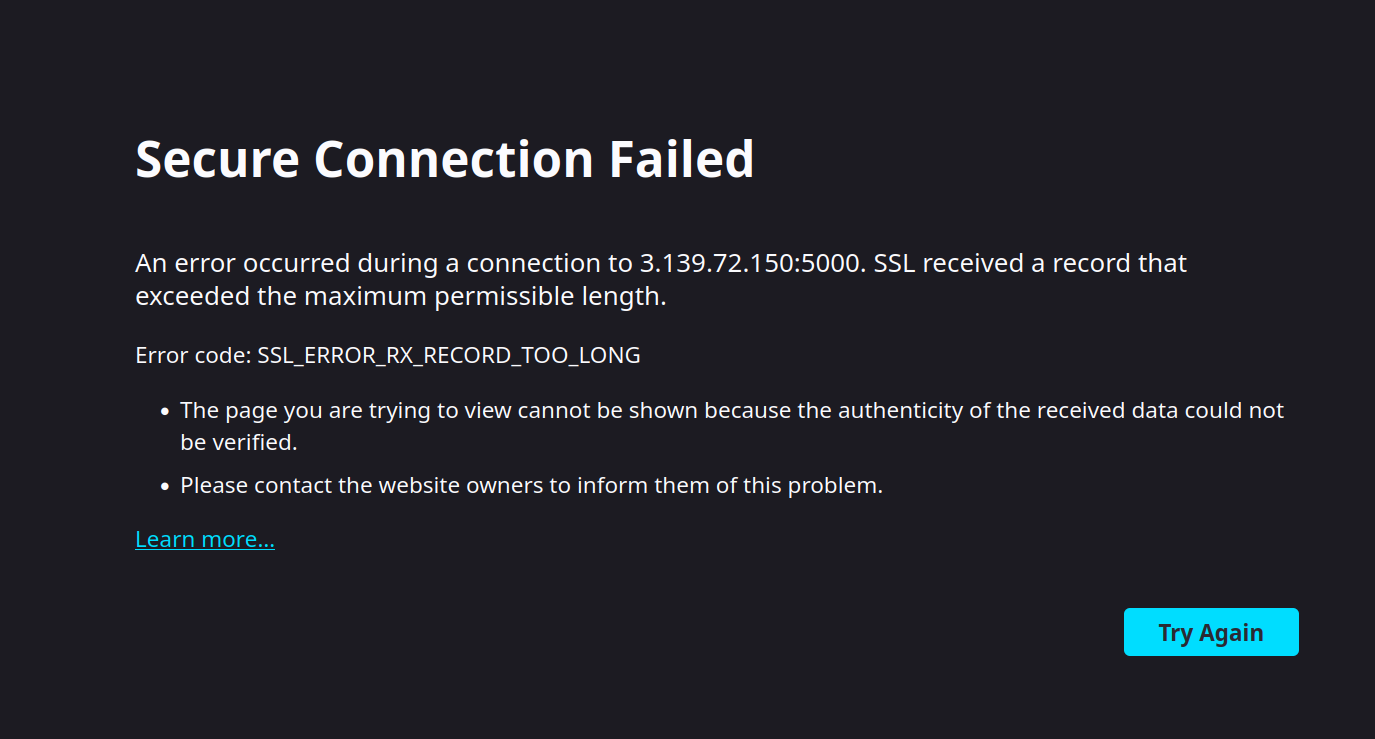
If you closed the tab, here’s the command To navigate to the folder (condensed from earlier section):

| cd /home/ubuntu/CPath/"Part 2" |
| --- |



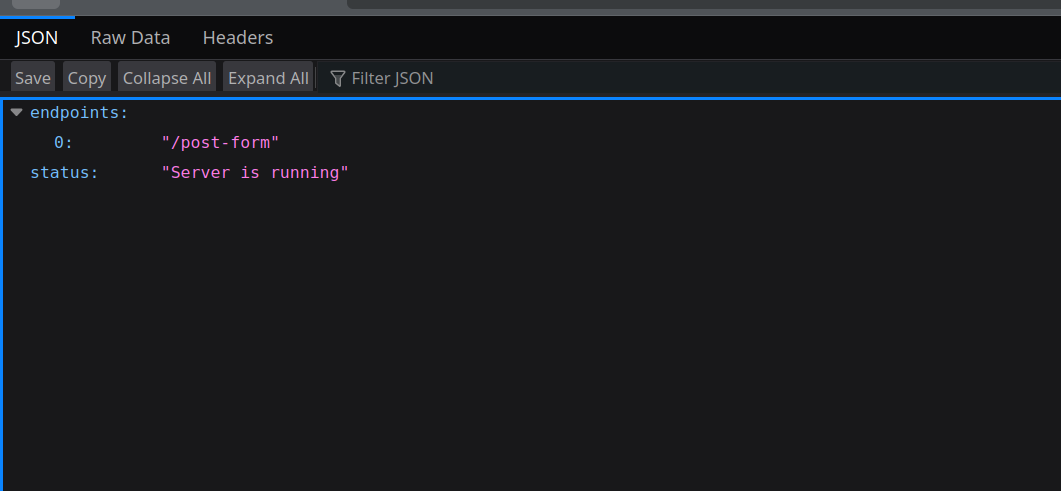
### Now connect again

If you see this

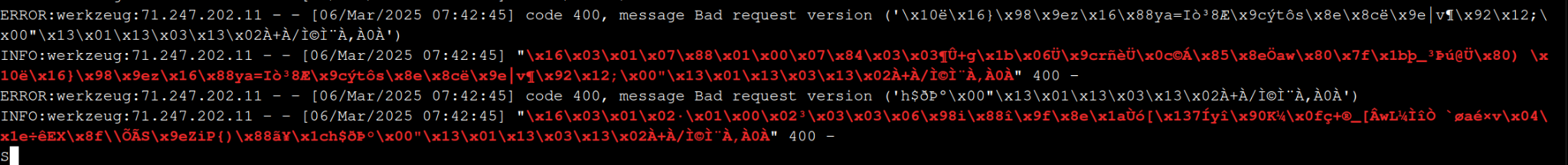


Go to the url and replace https with http. We don’t have SSL yet.

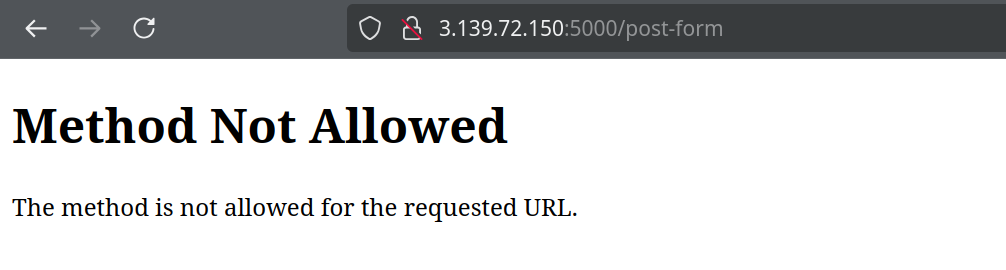
Success!



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



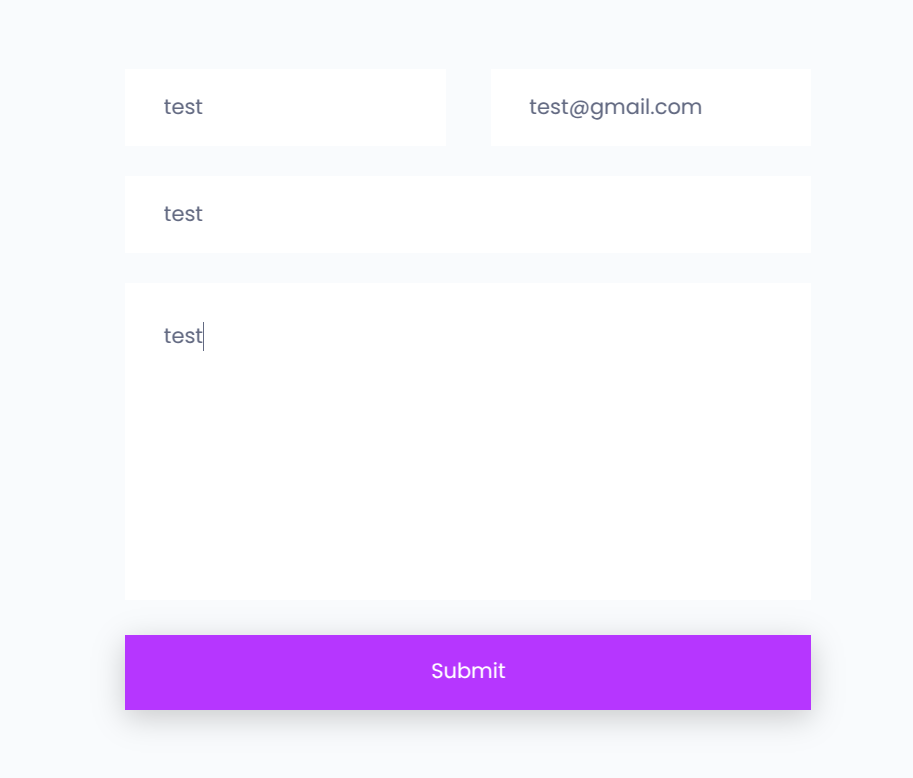
Not allowed, because we are making a GET request from url. We need to use it on our website instead.

### Send a form from our website

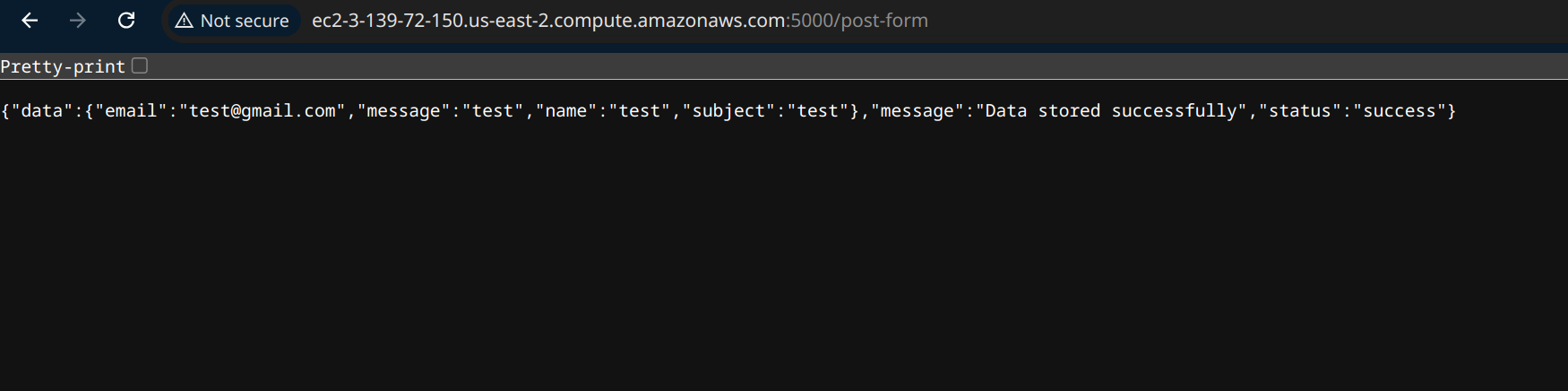
Add the attached form code to your website somewhere. Replace the action with the post-form url.

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



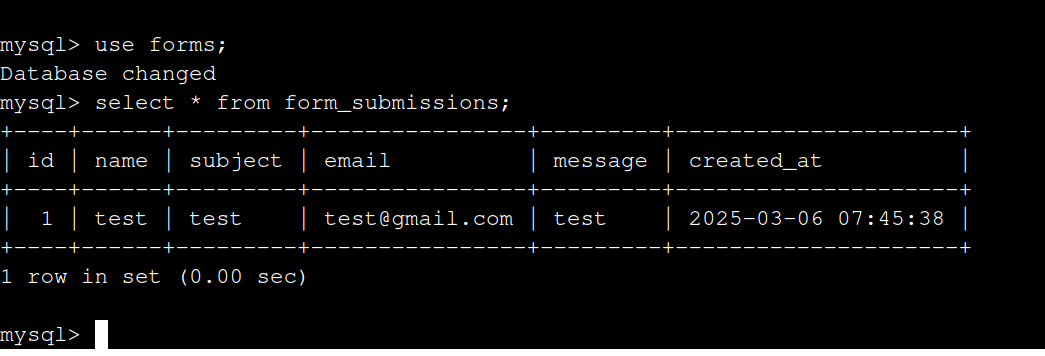
### Let’s see our data in the database.

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; |
| --- |



Success!