Soliz, Joseph
Simplilearn Post-Graduate Cloud Computing
PGP CC - AWS Cloud Fundamentals
6/28/2024

AWS Fundamentals - Course-End Project 1

Creating a VPC with Database and EC2 Instances

Objective

To design and construct an Amazon Virtual Private Cloud (VPC) architecture that includes an EC2 instance within a public subnet and a database instance within a private subnet

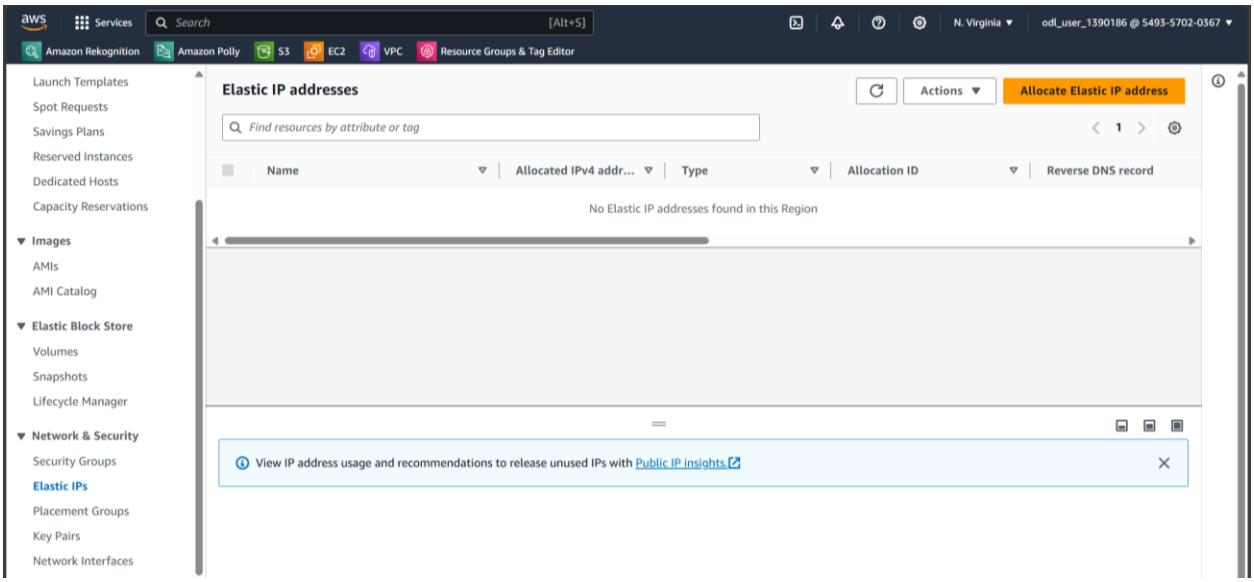
Tasks:

1. Allocating an Elastic IP address
2. Creating a VPC with public and private subnets
3. Creating an additional private subnet
4. Creating a VPC security group
5. Creating a VPC security group for a private DB instance
6. Creating a DB subnet group
7. Creating a DB instance in the private subnet
8. Creating an EC2 instance in the public subnet

Step 1: Allocating an Elastic IP address

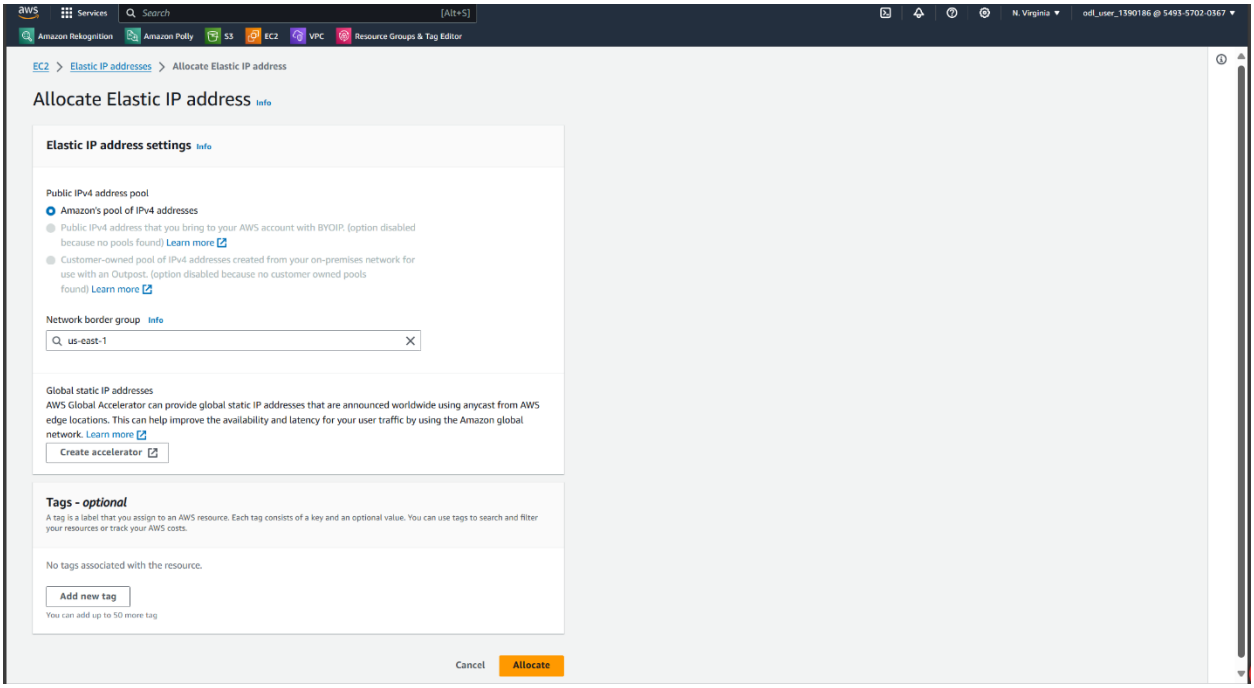
Go to EC2 Dashboard. To your left under Network & Security, Click on Elastic IP.

To create an elastic IP, click on **Allocate Elastic IP address**:

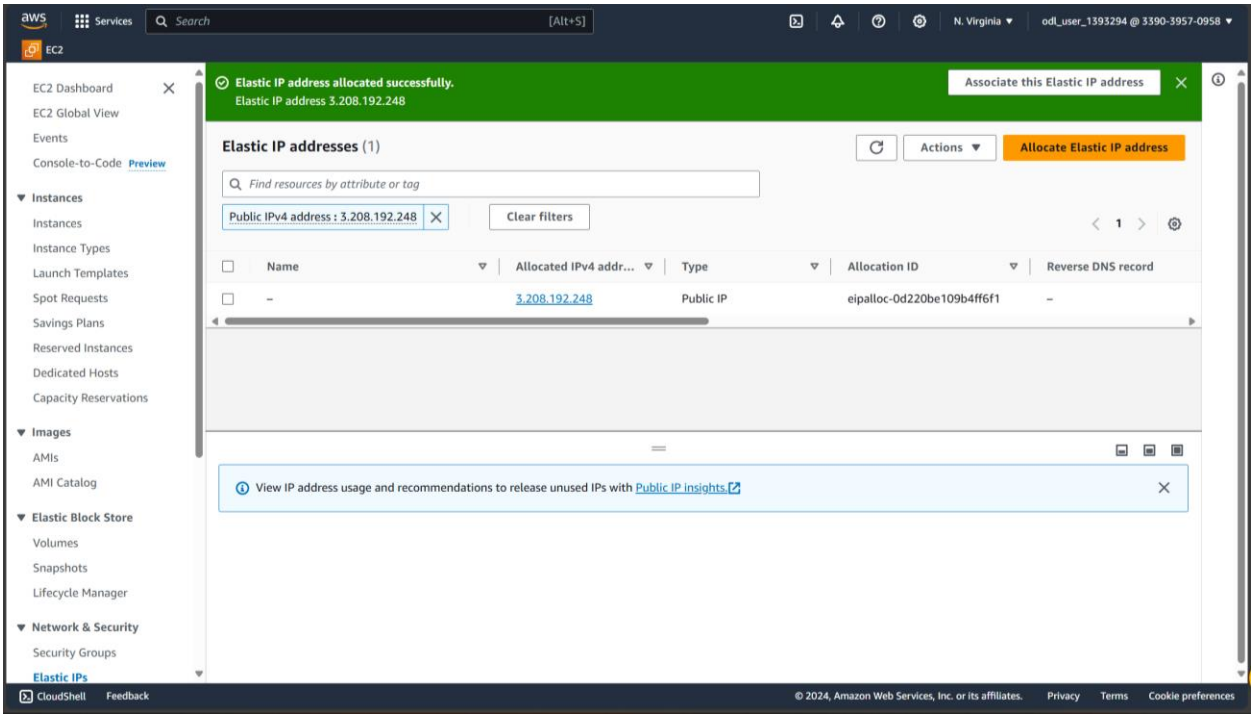


In Allocate Elastic IP address, be sure that Amazon's pool of IPv4 addresses is selected under Public IPv4 addresses.

Your network border group should match with the Availability Zone you are under.



Click on **Allocate**.

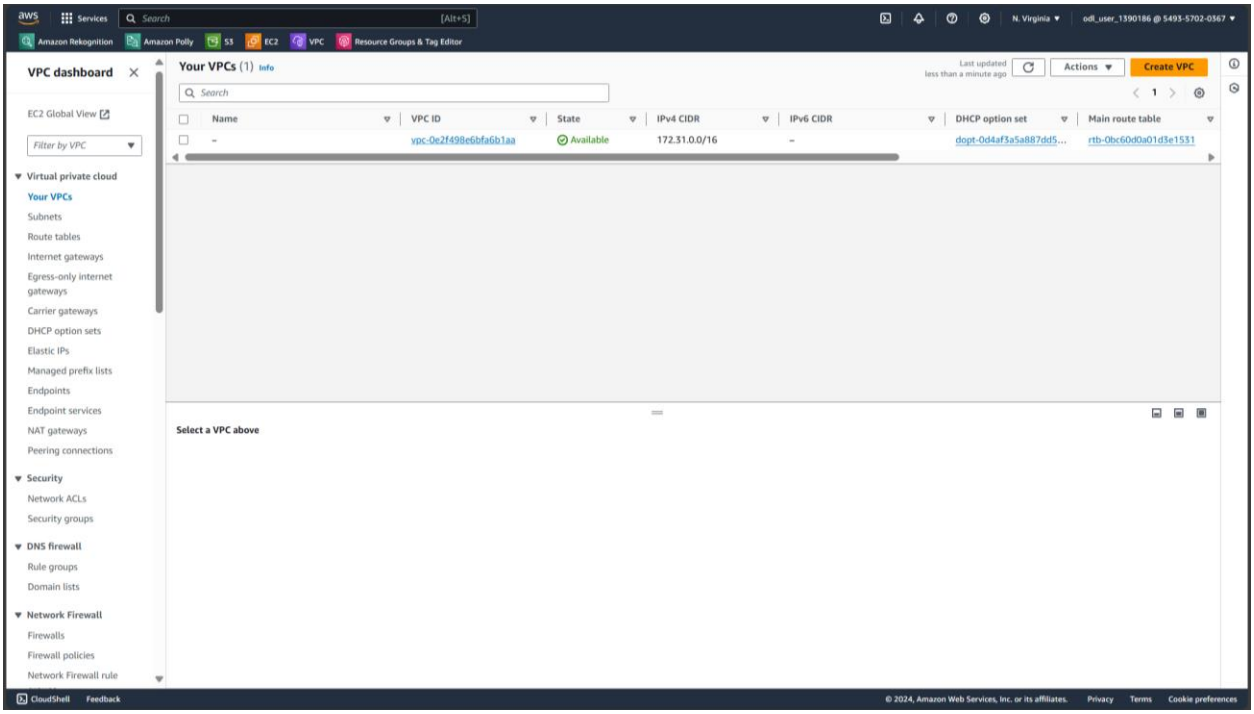


You have your Elastic IP address all set.

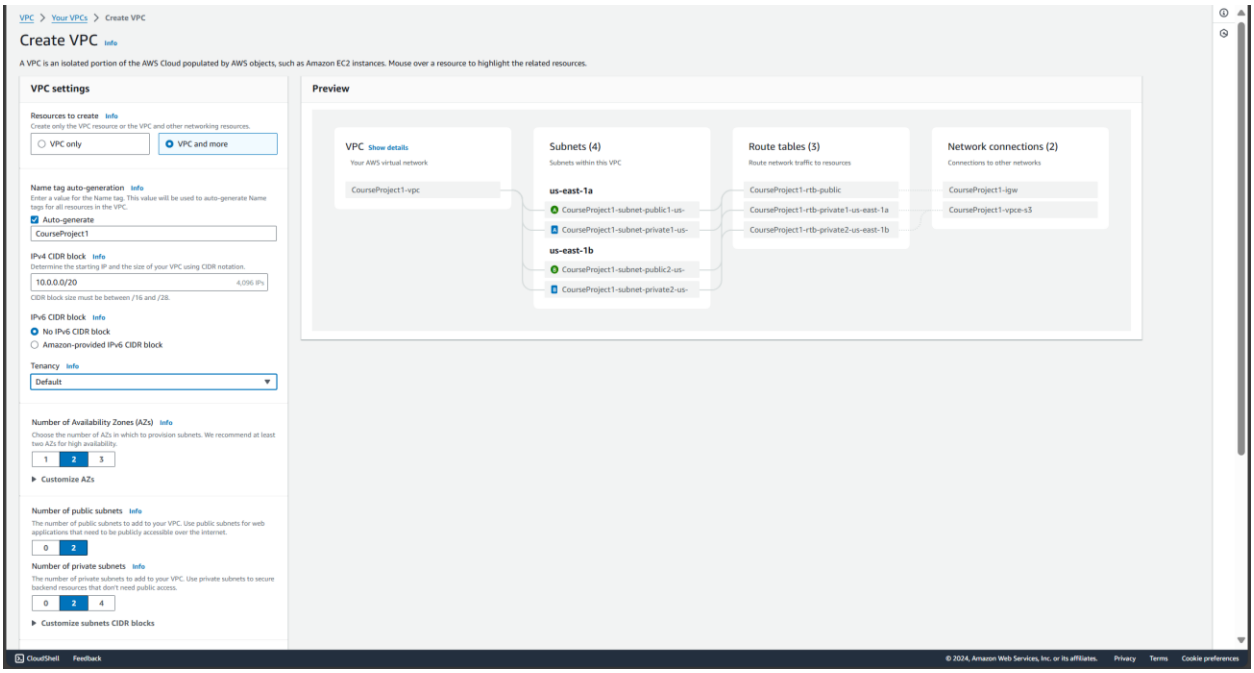
Step 2: Creating a VPC with public and private subnets

Most of these relate to Lesson 6 LEP.

Go to the VPC dashboard in AWS. Already you have a default VPC.

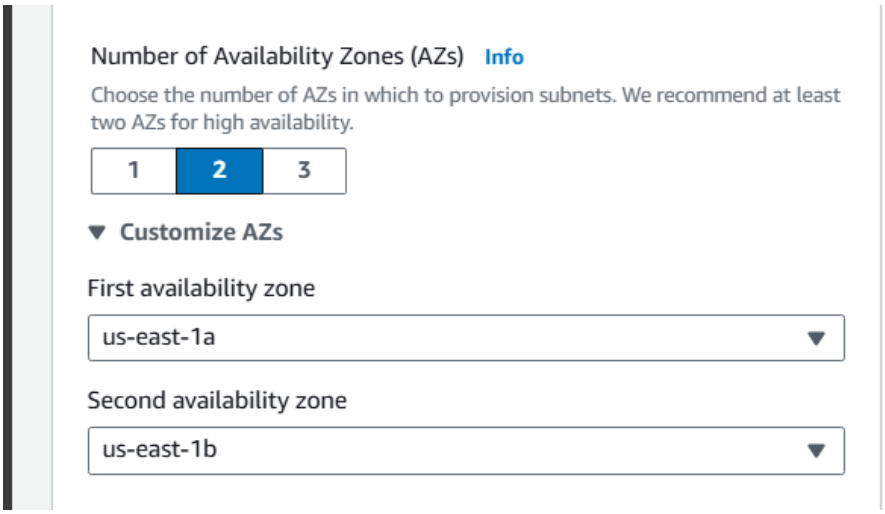


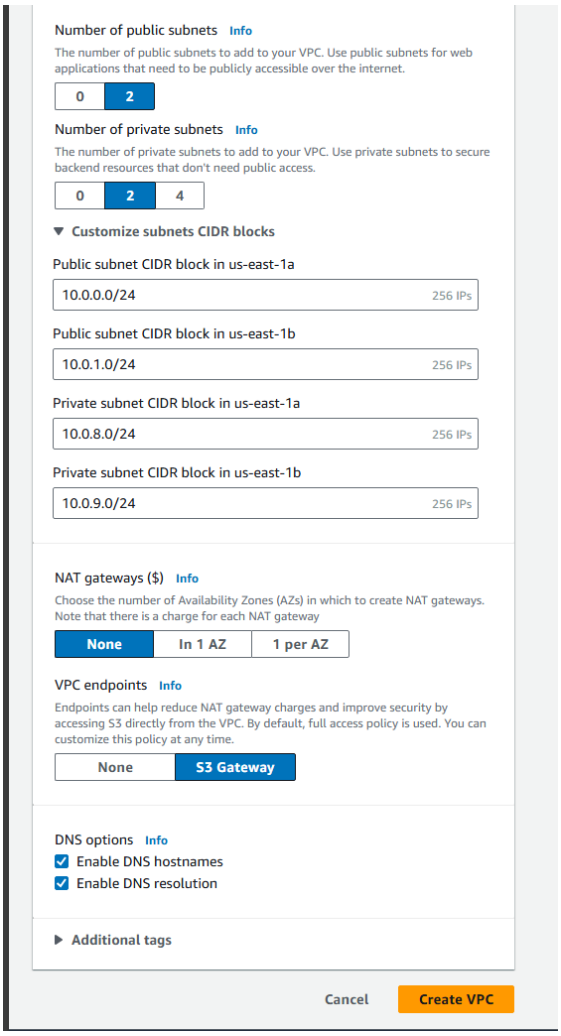
To create a VPC, click on **Create VPC**.



You have multiple options from VPC only that allows you to choose to create the VPC by itself and the subnets separately. Or to save time, you can choose “VPC and More” where you create a VPC and a set of subnets.

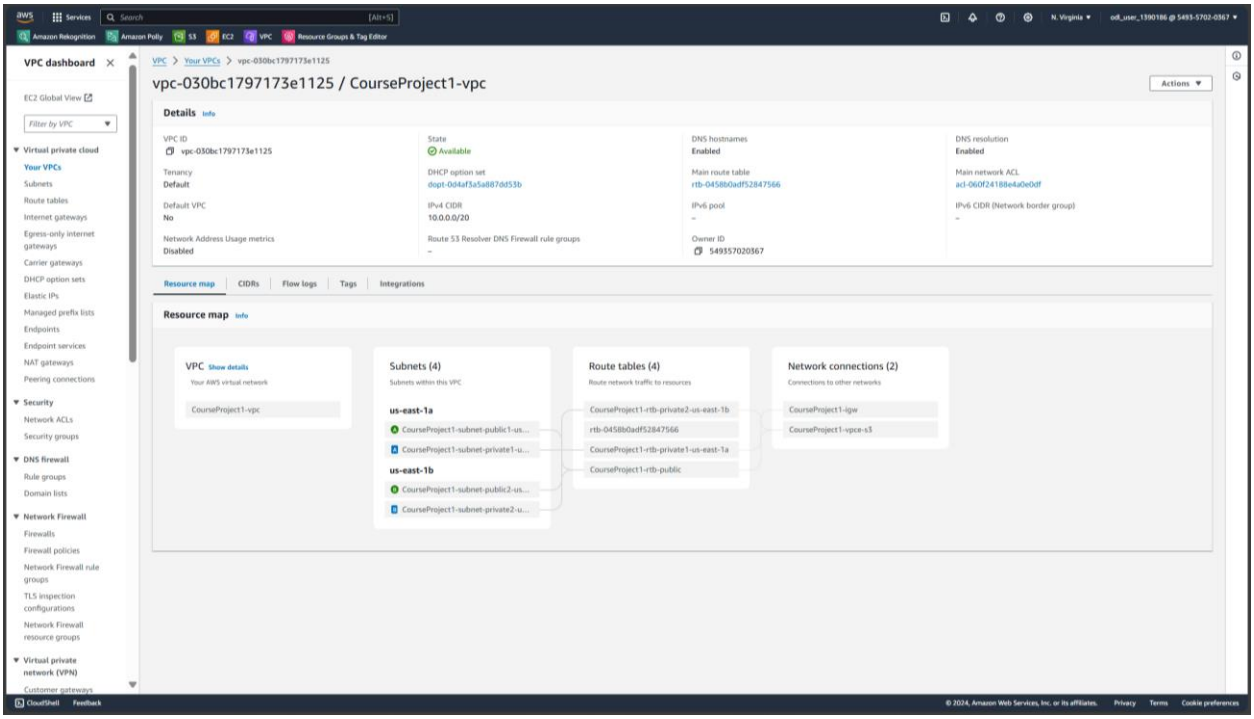
In this case, we choose our IPv4 CIDR block for our VPC to be 10.0.0.0/20





Our customize block comes out as 10.0.0.0/24, 10.0.1.0/24, 10.0.8.0/24, 10.0.9.0/24. This will be useful when creating an additional subnet.

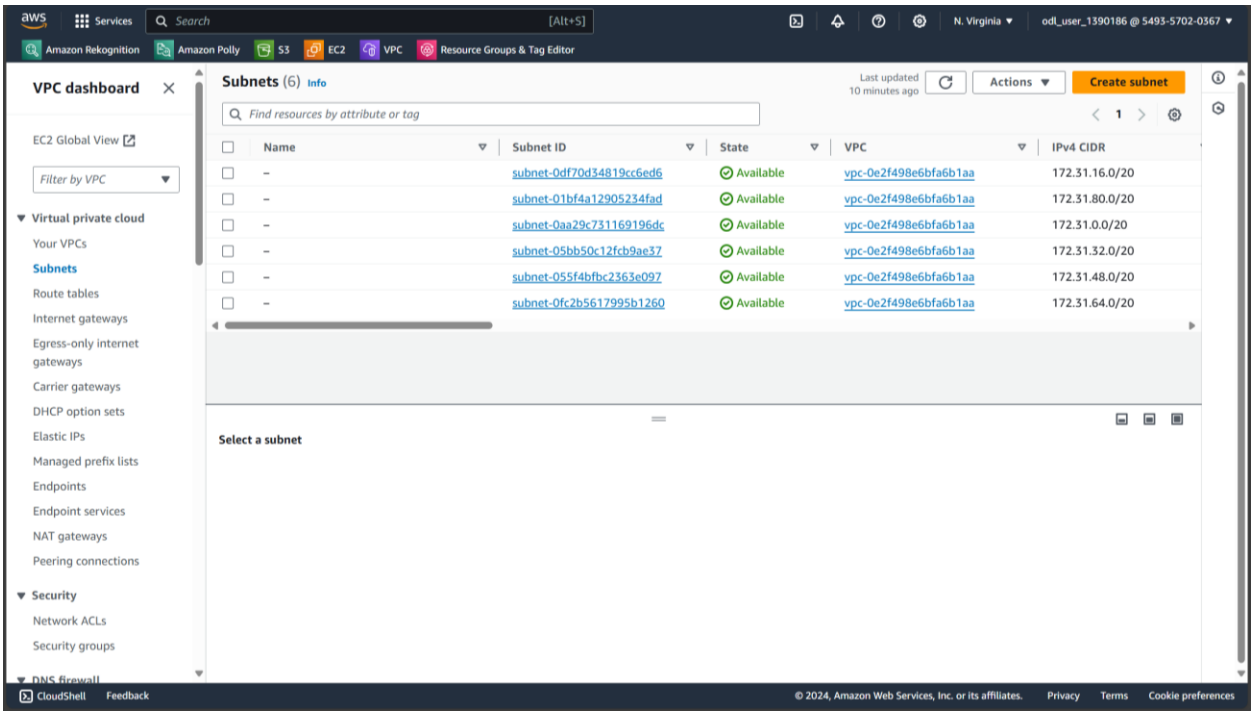
Once done, select **Create VPC**.



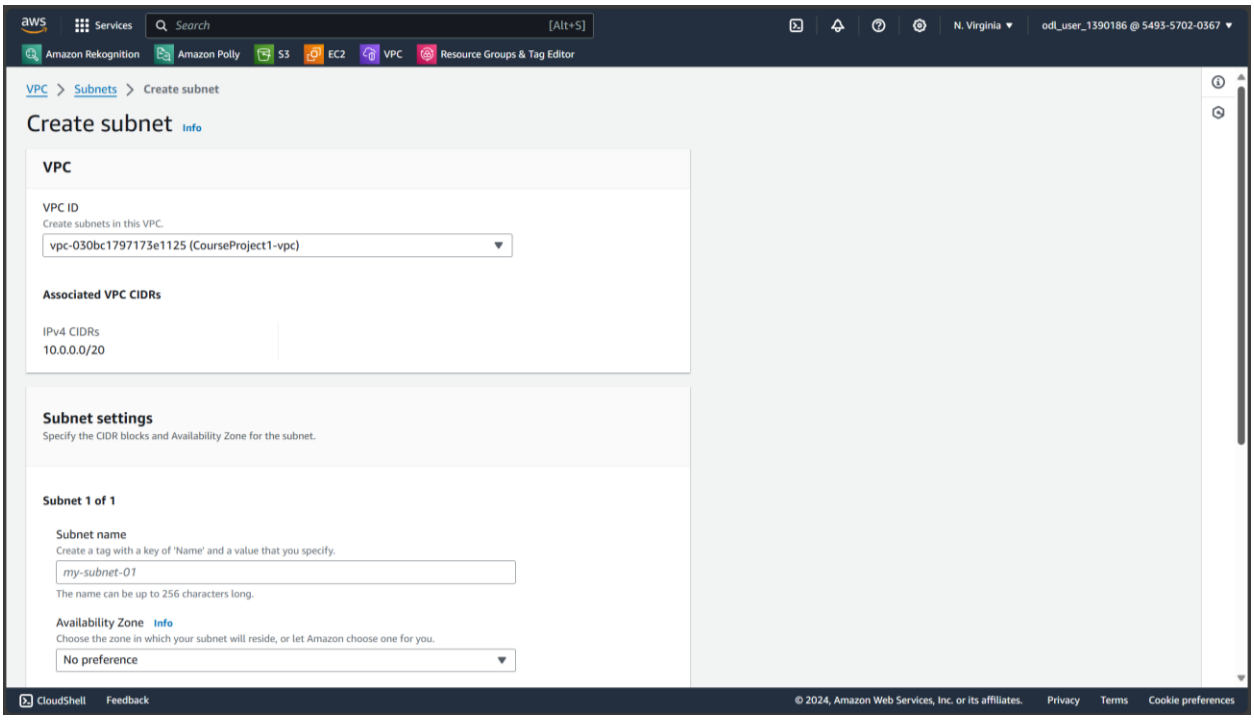
VPC created

Step 3. Creating Additional Subnet

Following on the Subnet page, go to Create Subnet.



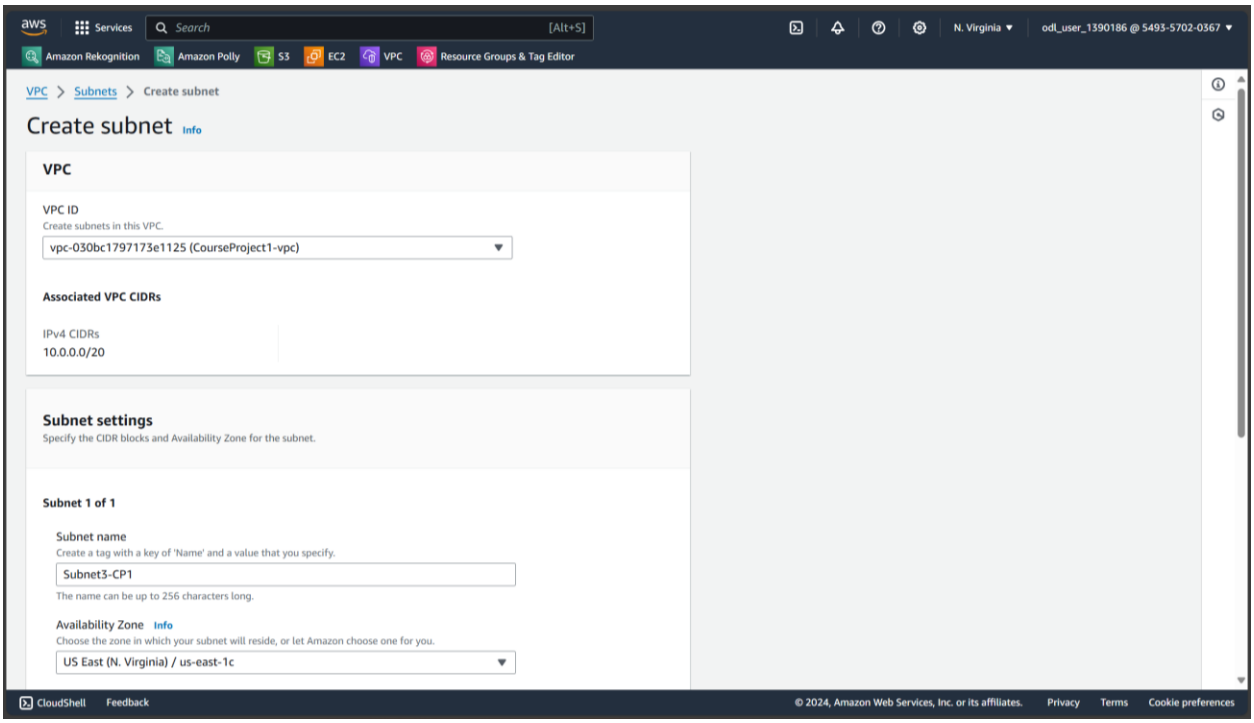
Select the VPC you want to connect to. In this case, “CourseProject1”. Give your subnet a name of your choosing.



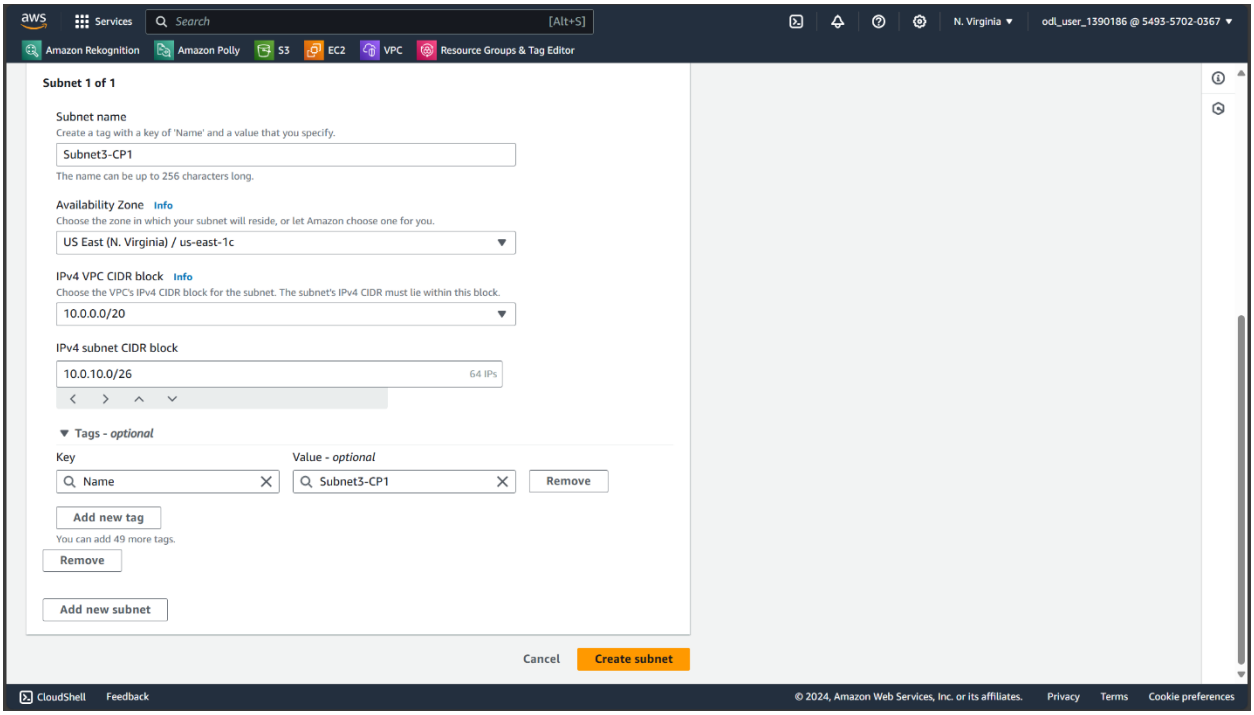
Choose an AZ that hasn’t been taken:

As your first four are us-east-1b, us-east-1b, us-east-1a, us-east-1a

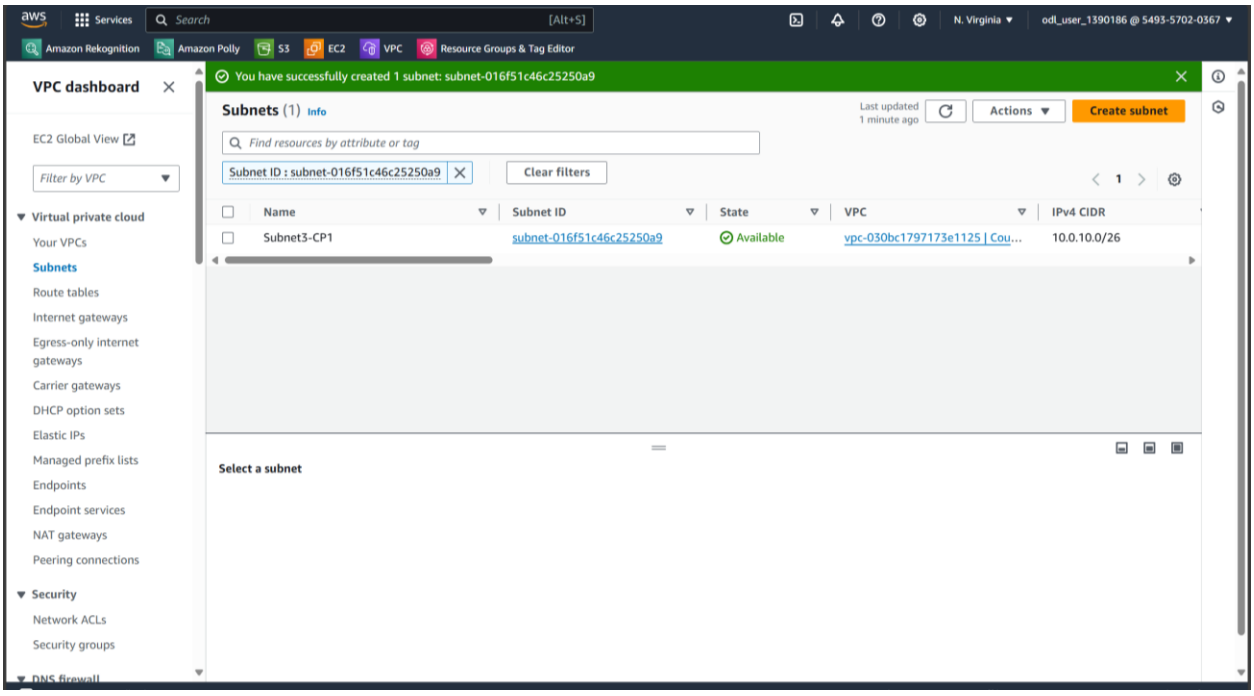
Or select No preference. In this case, we will do us-east-1c



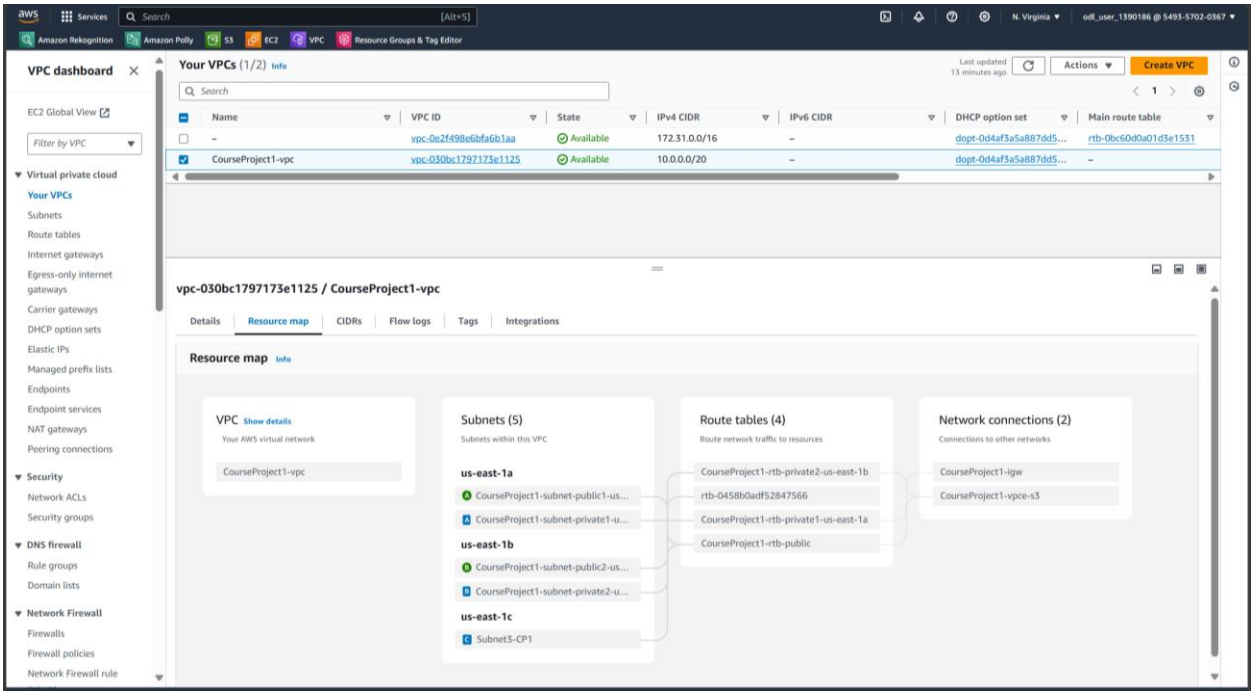
Choose your IPv4 VPC CIDR block. Make sure it’s higher than the other subnets.



Create Subnet.

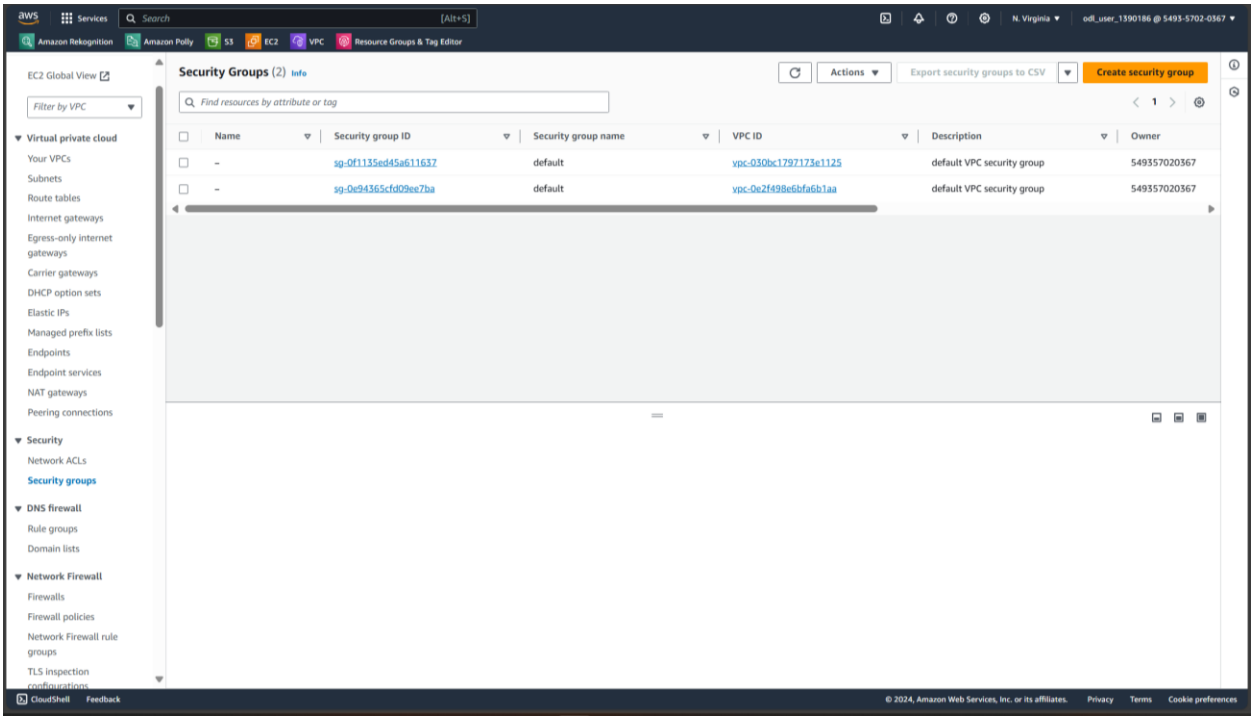


You can see it shown in your resource map for your project

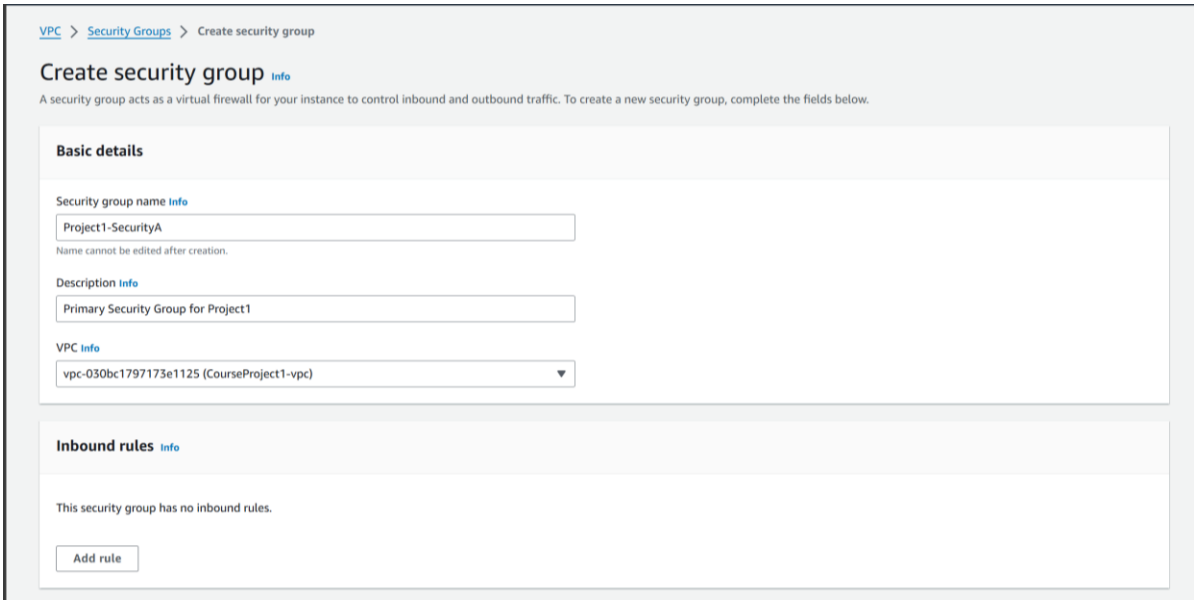


Step 4. Creating a VPC security group

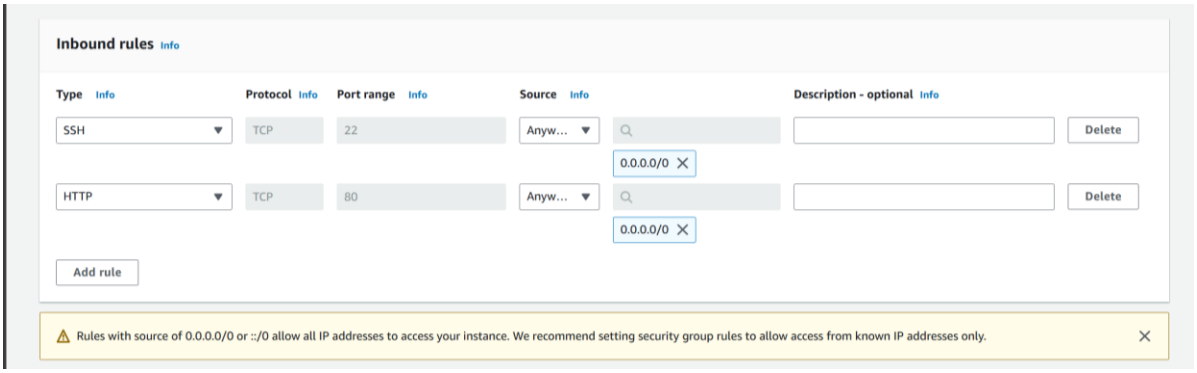
Go to Security Groups on the left in the VPC Dashboard under Security. Select Create Security Group.



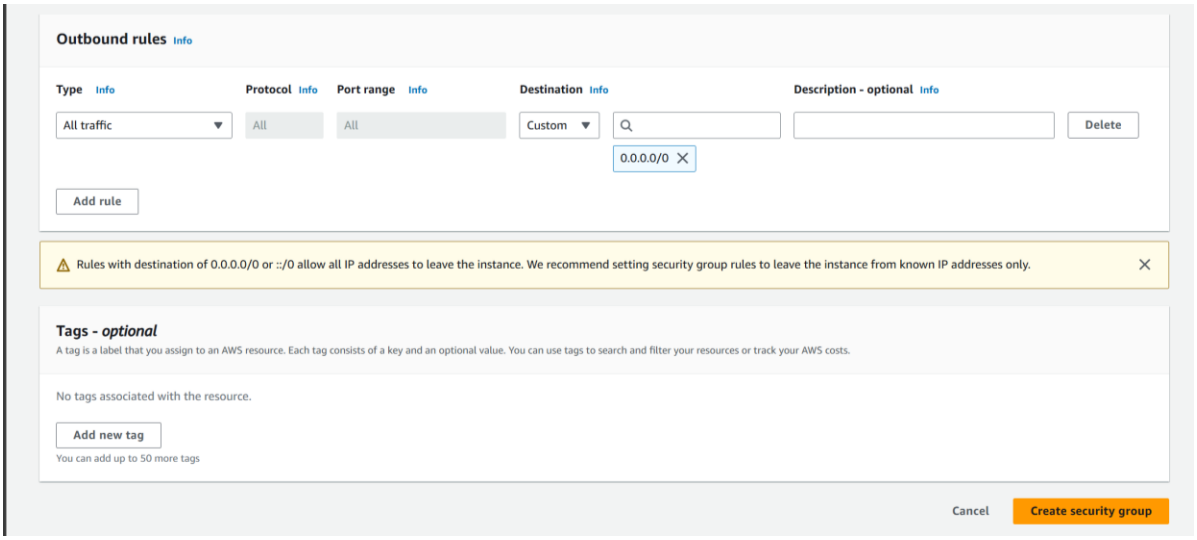
In Create Security Group, type in the name you want to call your security group (Option to provide a description info) and select the VPC you want to attach it to.



For Inbound Rules

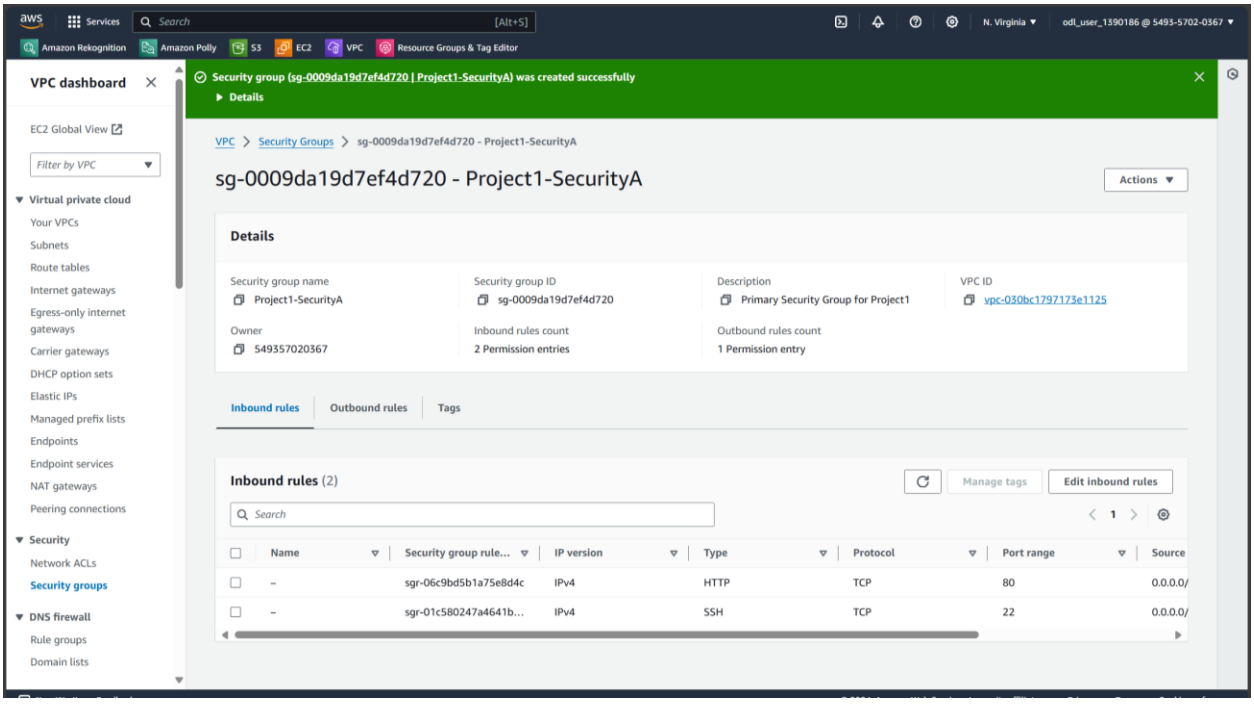


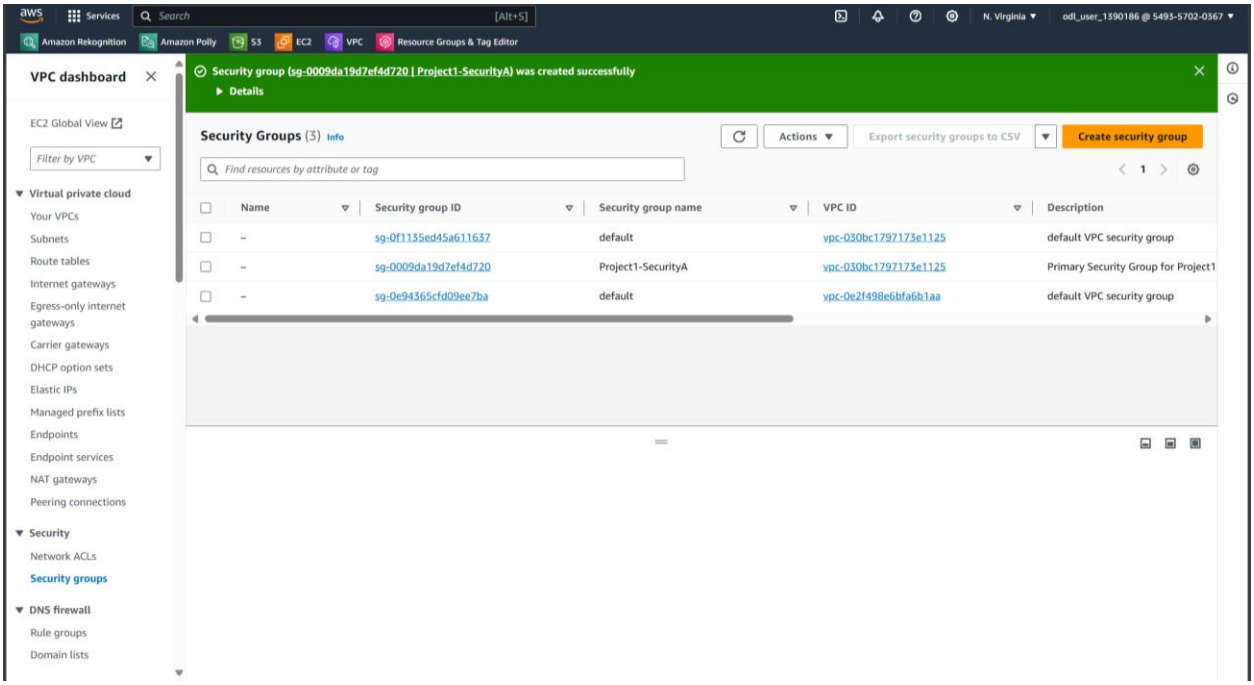
While it would be safer to have your SSH be your IP address. For this example, will have it be Anywhere-IPv4 0.0.0.0/0



Be sure outbound rules have the destination set to Custom and be 0.0.0.0/0

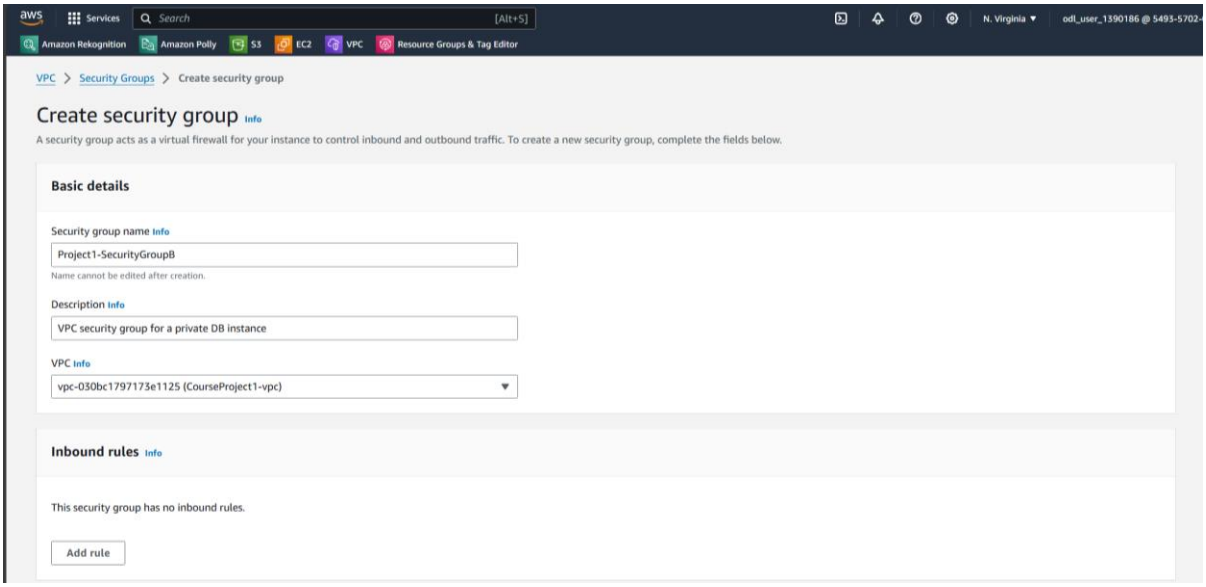
Click on **Create security group**.



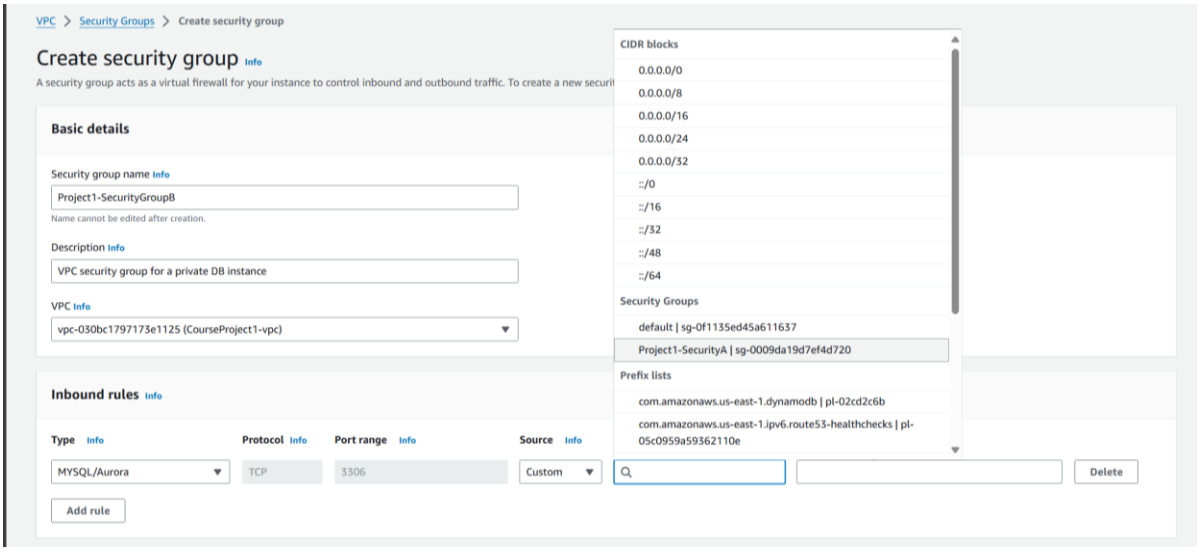


Step 5. Creating a VPC security group for a private DB instance

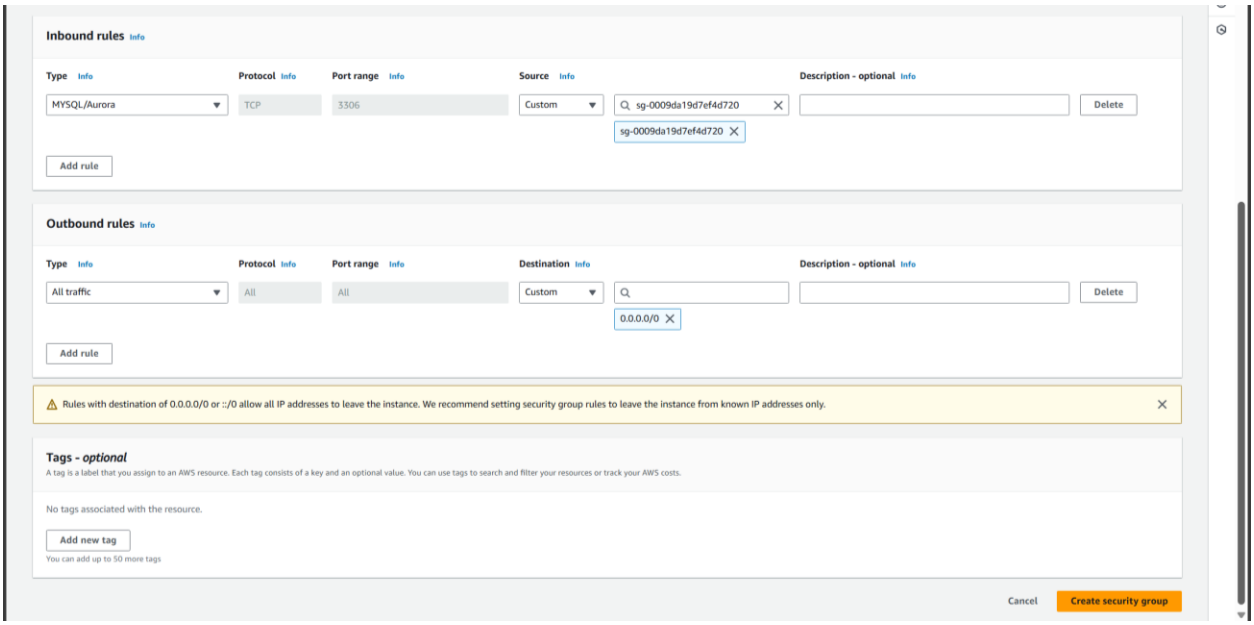
Repeat the same process. Go to Create Security Group. In the page, type your custom name, description, and the VPC you want your security group to be affiliated with.



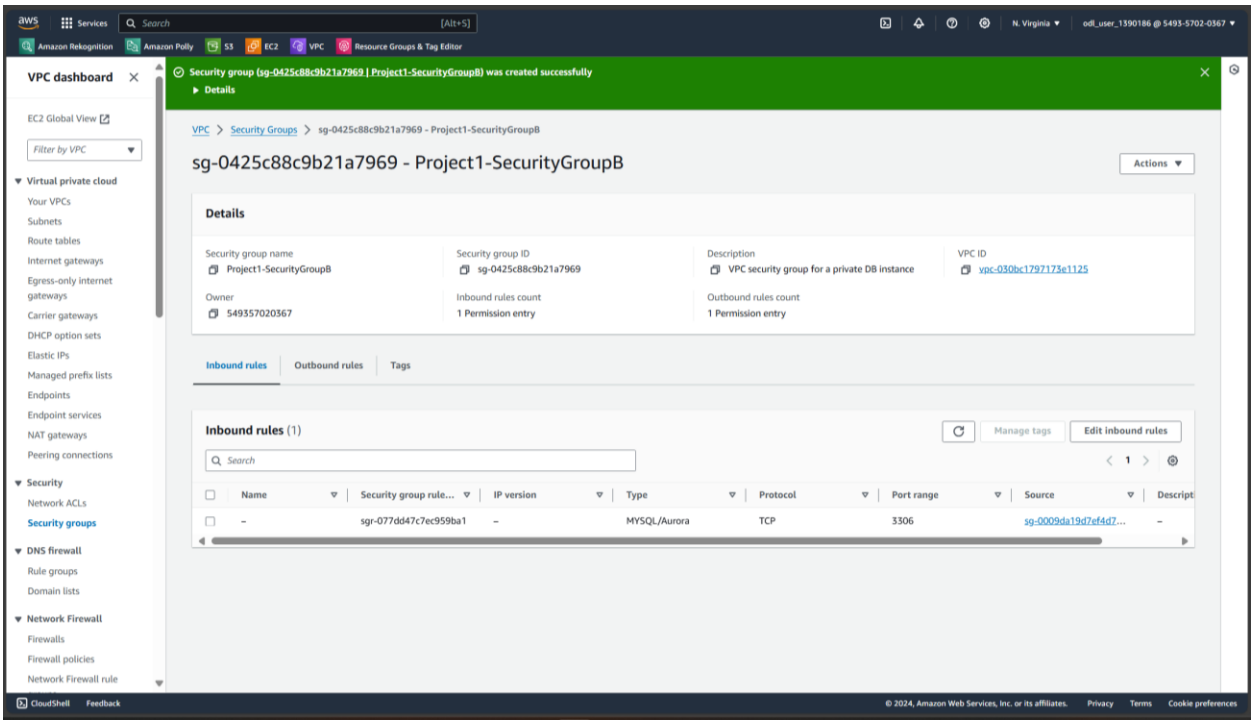
For Inbound Rules, since we want this to be for a private DB instance, we can have it set to “MySQL/Aurora” for Type and the source to your custom Security Group.



Once set, be sure Outbound Rules are the same as before



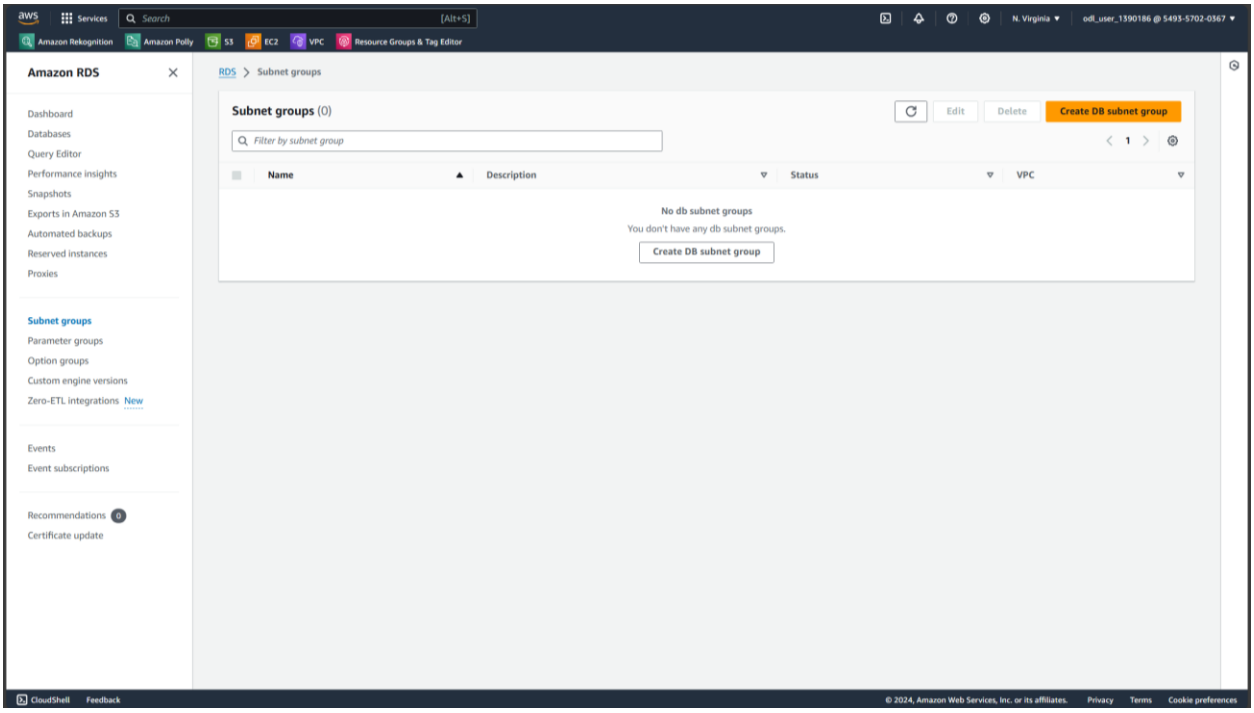
Click on **Create security group**.



Step 6: Creating a DB subnet group

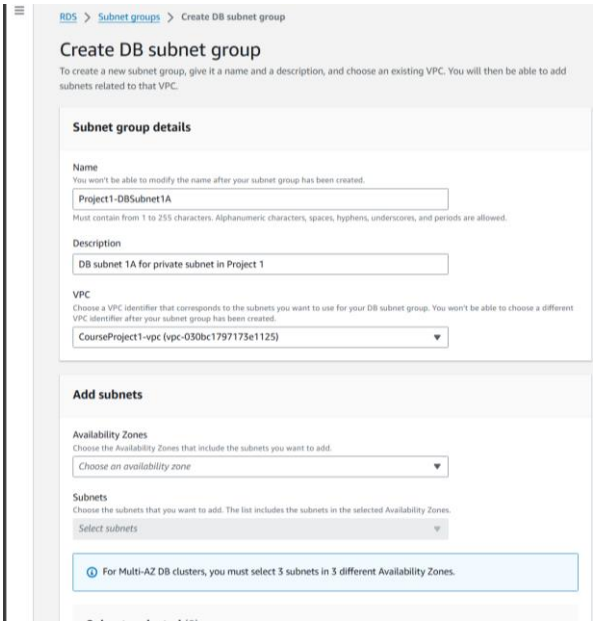
Go to RDS. Under the RDS Dashboard, select Subnet groups.

Select '**Create DB subnet group'**.

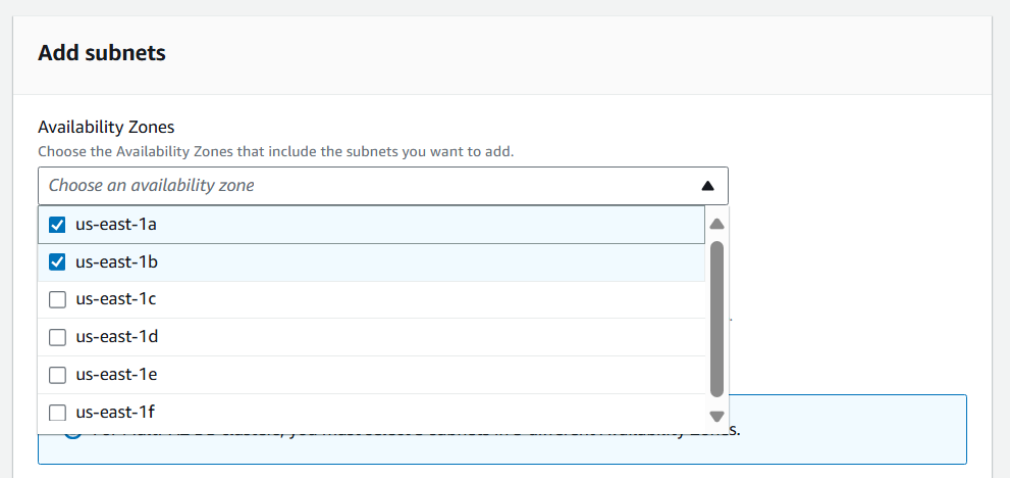


In Create DB subnet group:

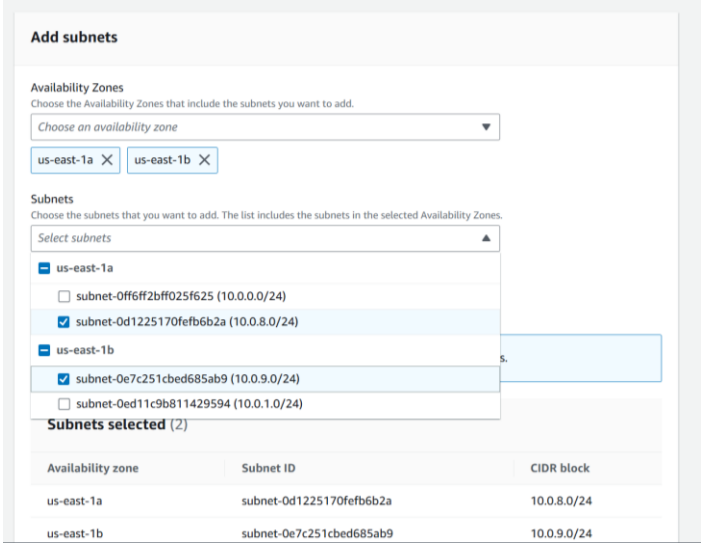
Type in the name for you DB subnet group and a description if you want to. Be sure to select the VPC you want to connect to “CourseProject1-vpc”.



To add subnets, first select the Availability Zones you want to connect to. Be sure they are the AZs that your subnets for your VPC are in.



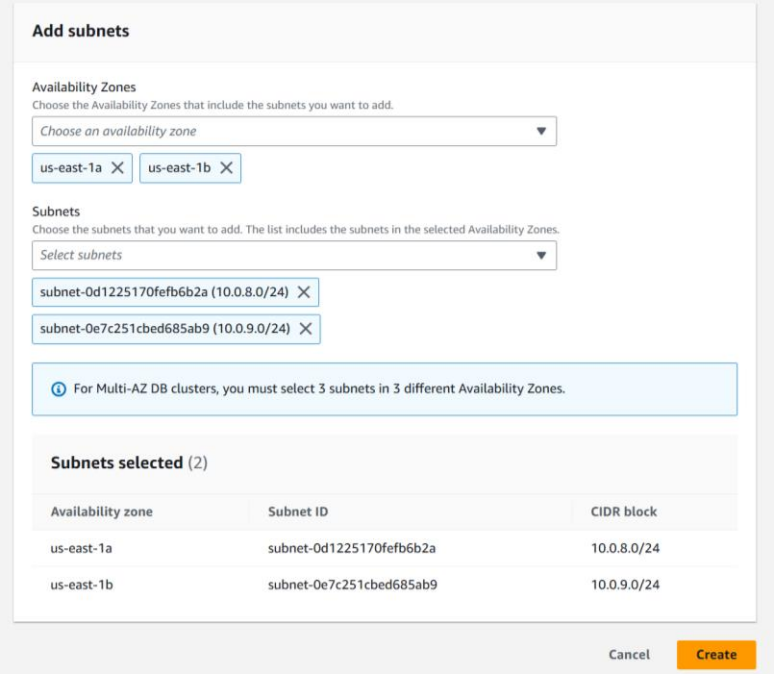
Next select the subnets you wish to connect to. Be sure they are the subnets that match with your VPC.



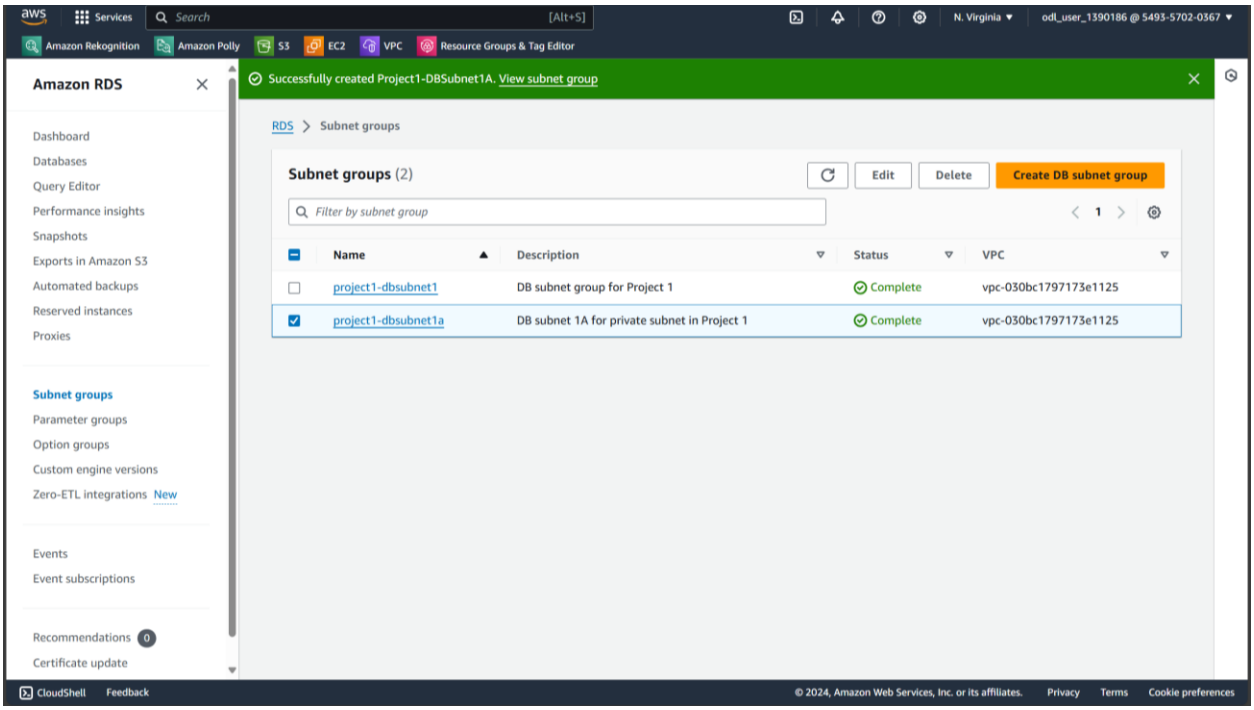
In this case, we are creating a private subnet group for the following step. Make sure you select the correct subnets that are your private subnets from VPC.

- Us-east-1a, 10.0.8./24
- Us-east-1b, 10.0.9.0/24

Creating a multi-AZ DB clusters.



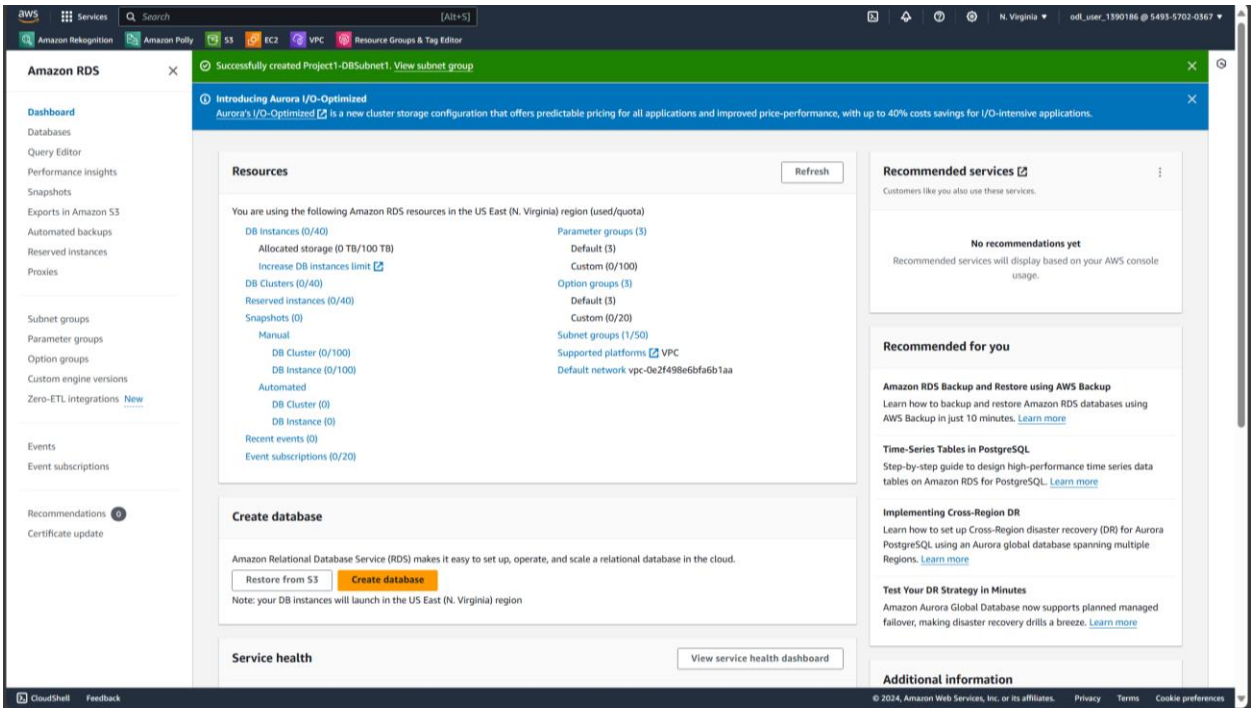
Click **Create**.



Your DB subnet is created among other subnets.

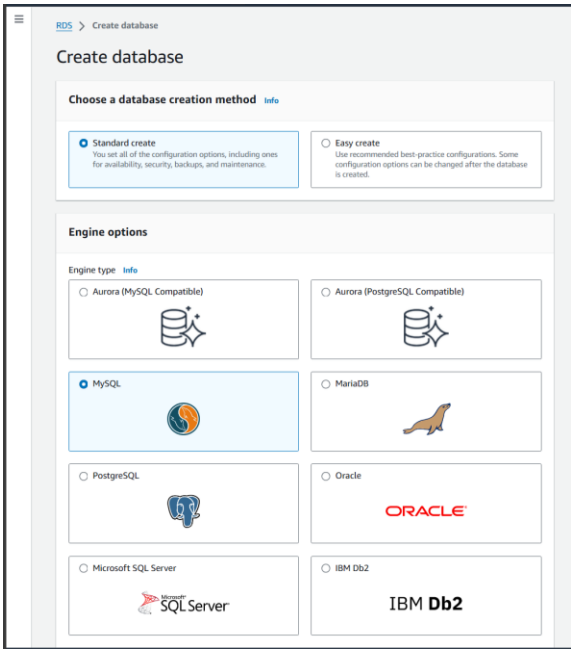
Step 7. Creating a DB instance in the private subnet

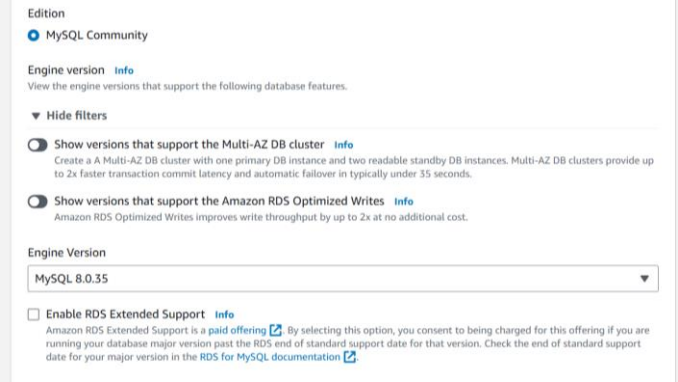
Go to RBS Dashboard. Click on “Create database”



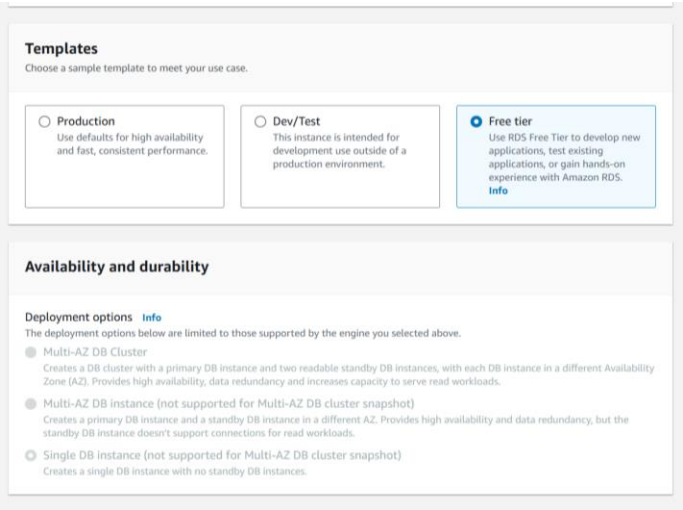
In Create Database:

Have ‘Standard create’ selected. For Engine Options, click on for your choice of engine. For this example, we are choosing “MySQL”.

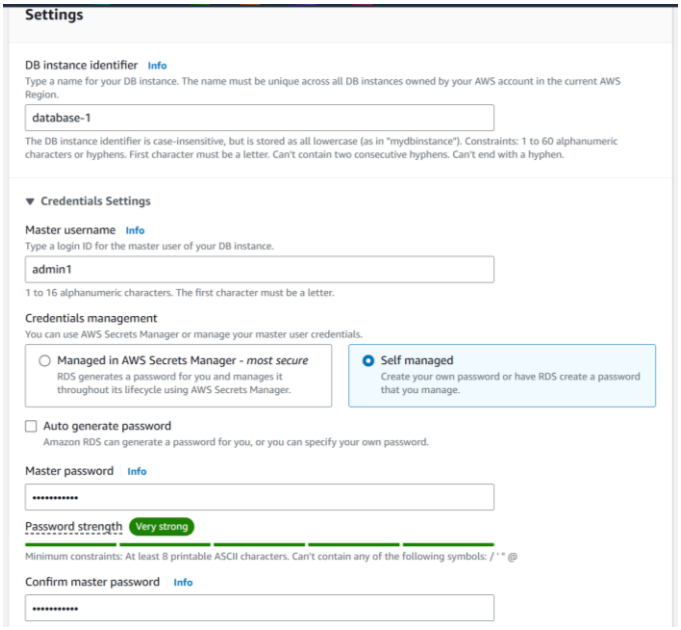




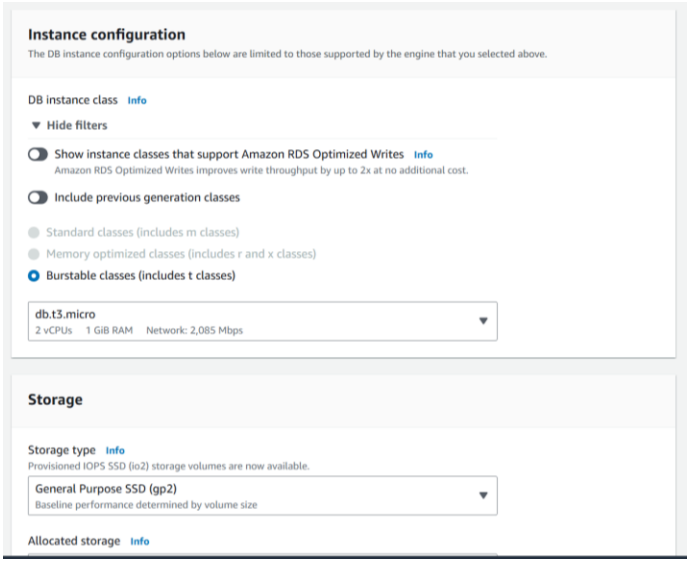
Be sure to choose “MySQL Community” is selected and you have the latest version of your engine selected.



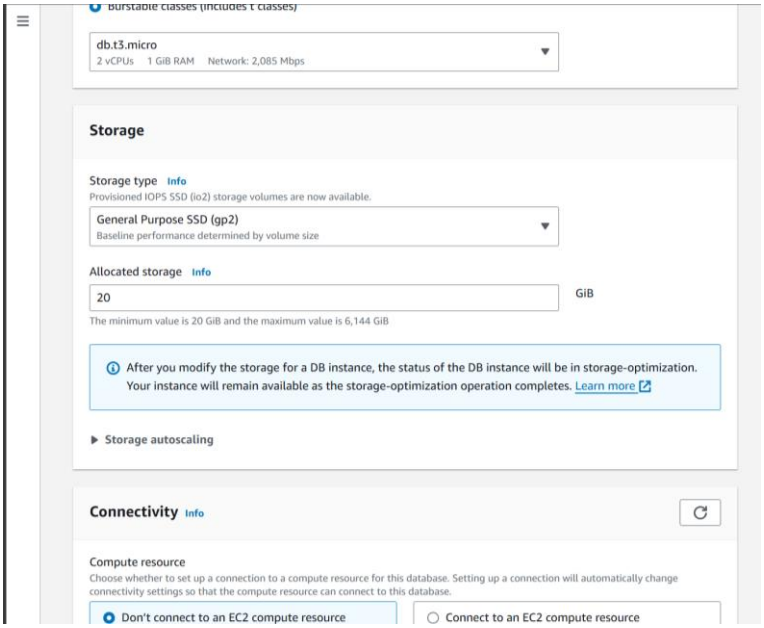
To save cost, select Free tier under Templates.



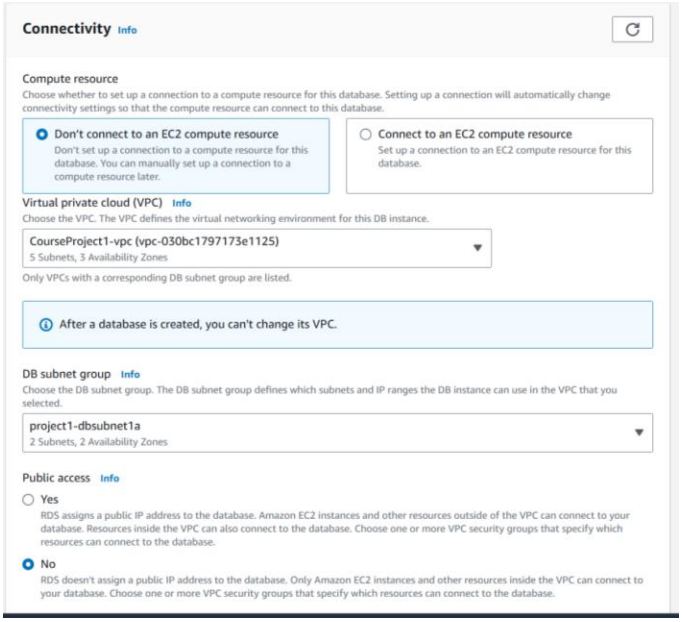
In Settings, type a name for DB instance identifier. Under Credentials Settings, create your username. To create and manage your own password, select “Self Managed” and unselect “Auto generate password”. Type in your own password and confirm.



In instance configuration, select db.t3.micro.



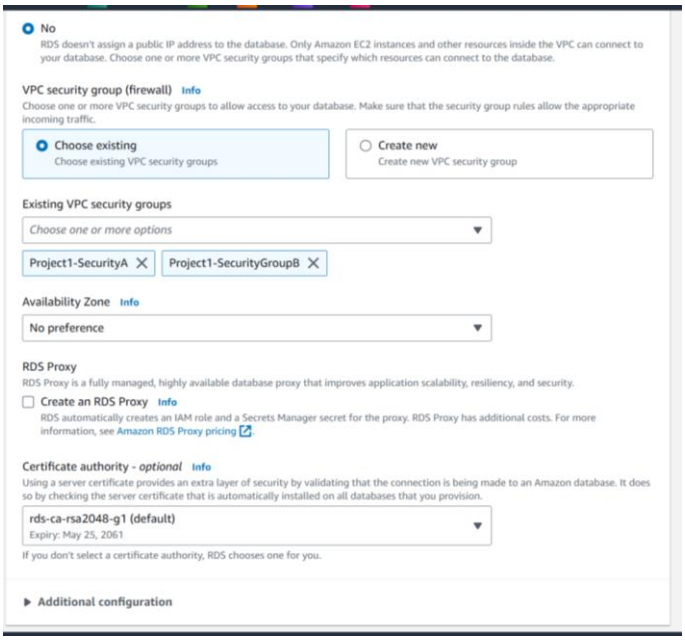
For Storage, have Storage type be gp2 and allocated storage 20 GiB.



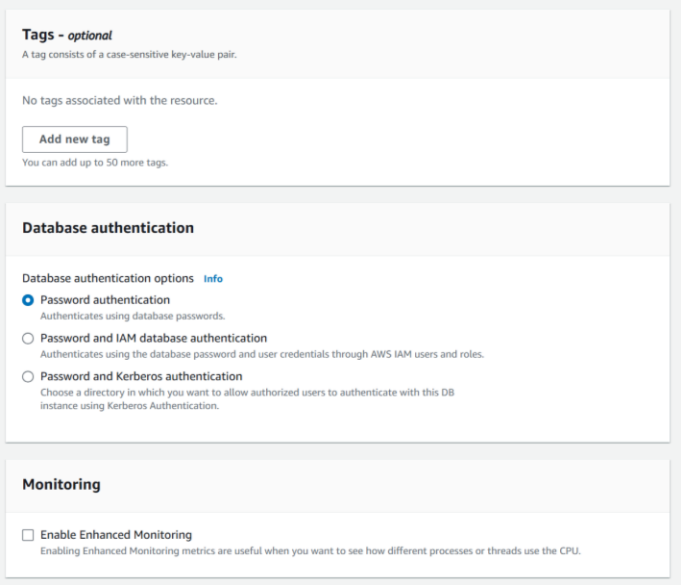
Under Connectivity, be sure your compute resource is NOT connected to EC2 compute resource.

Connect to your VPC “CourseProject1”. Connect to your DB subnet group. In this case, since it is for a private subnet, select the one you have set for your private subnets in VPC.

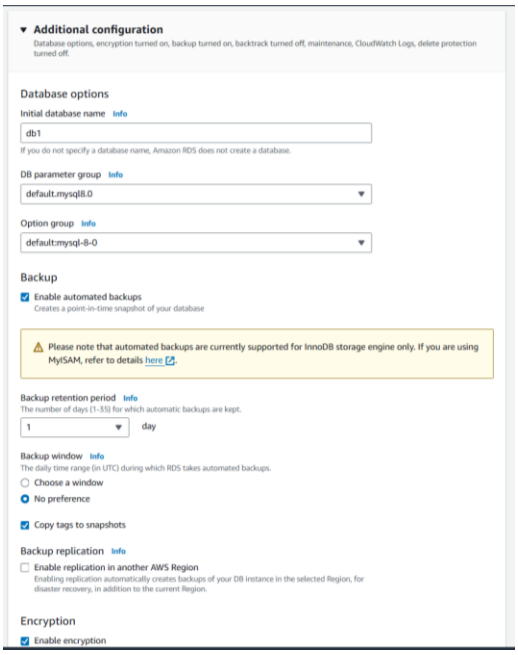
Click on “No” for Public Access.



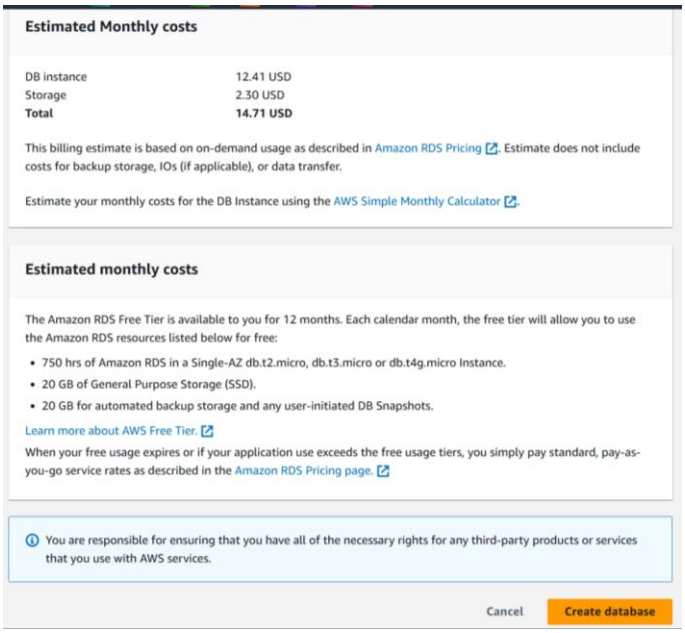
Choose your existing VPC security group. Here, we select both “Project1-SecurityA” for SSH and HTTP access and “Project1-SecurityGroupB” that is set for a private DB instance.



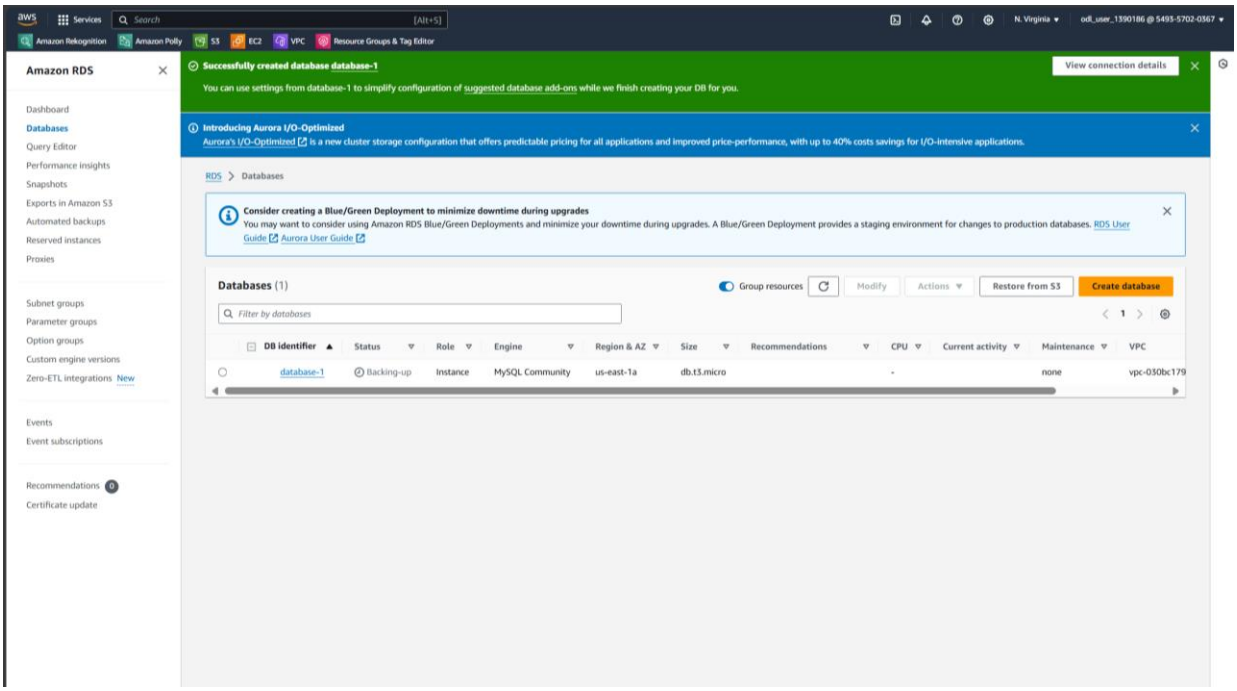
Be sure your Database authentication is set to Password authentication.



In Database options, create a name for your initial database name. The rest can be left alone.

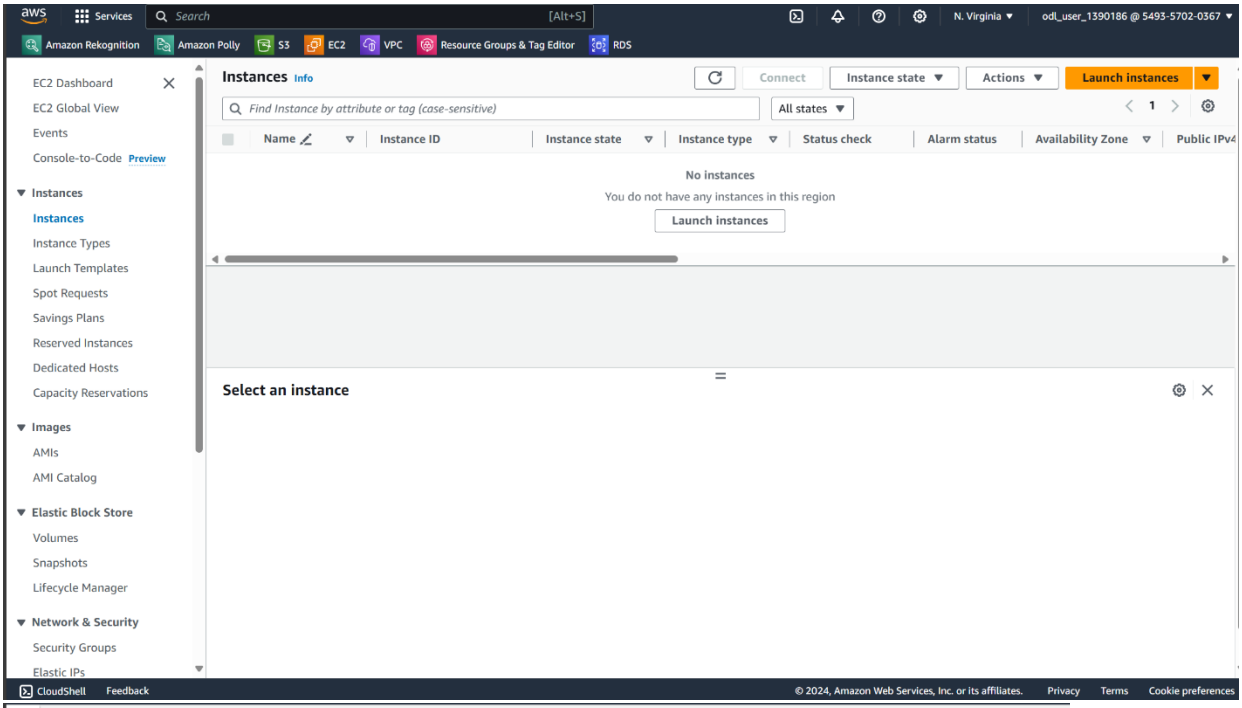


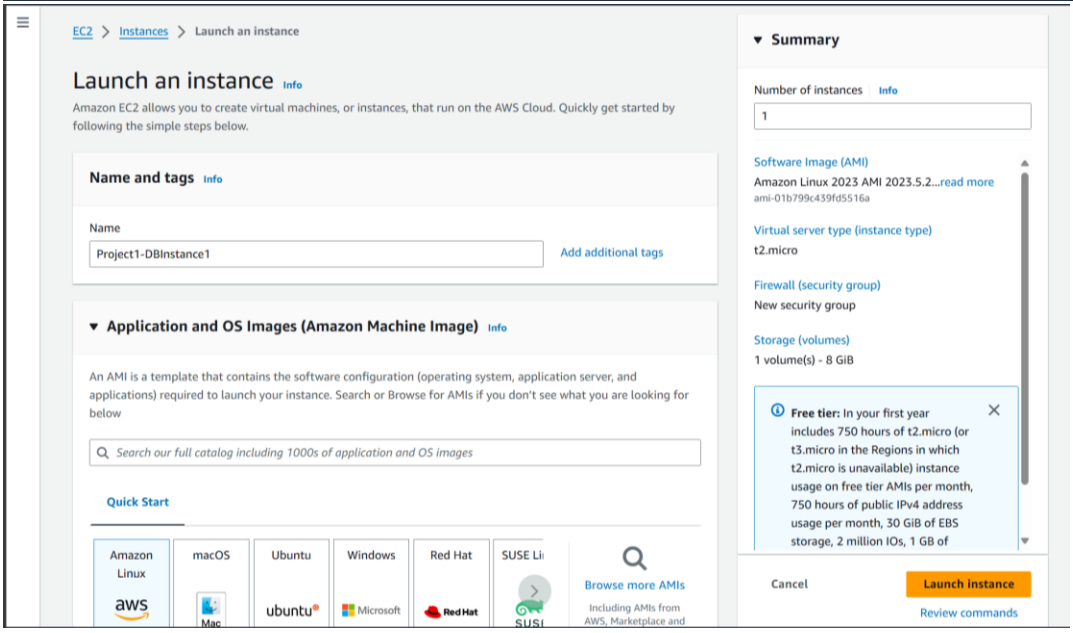
Click on **Create Database**.



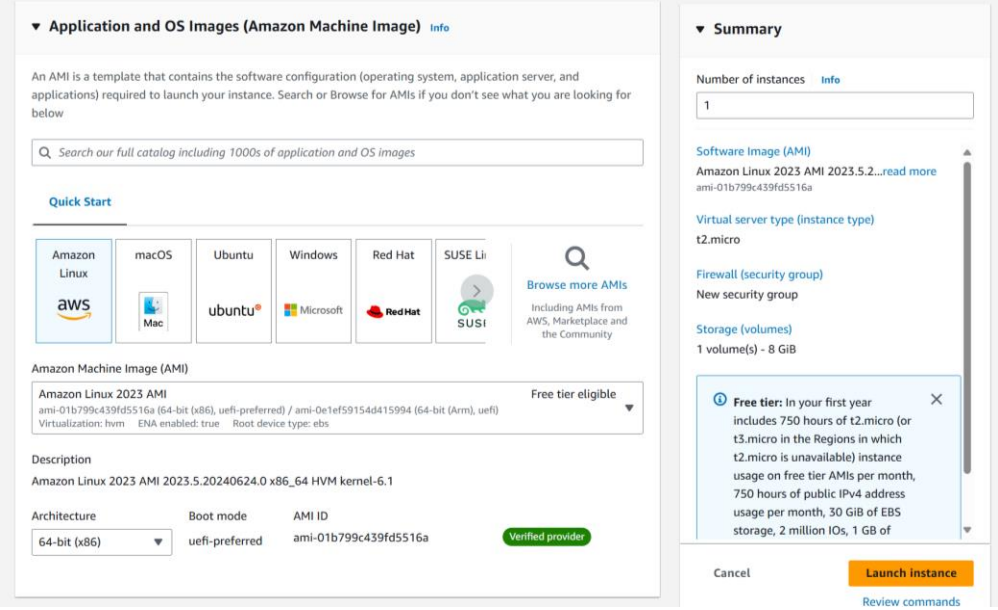
Step 8: Creating an EC2 instance in the public subnet

To create instance, go to EC2. Under the EC2 dashboard, select Instances. In Instances, select “Launch Instances” to begin process.

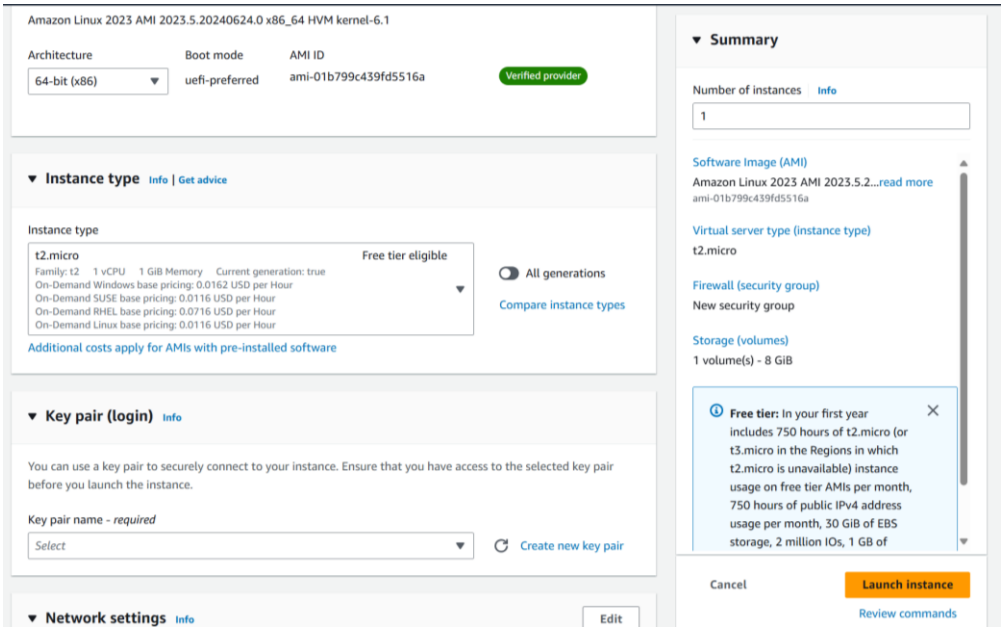




Create a name for your instance.

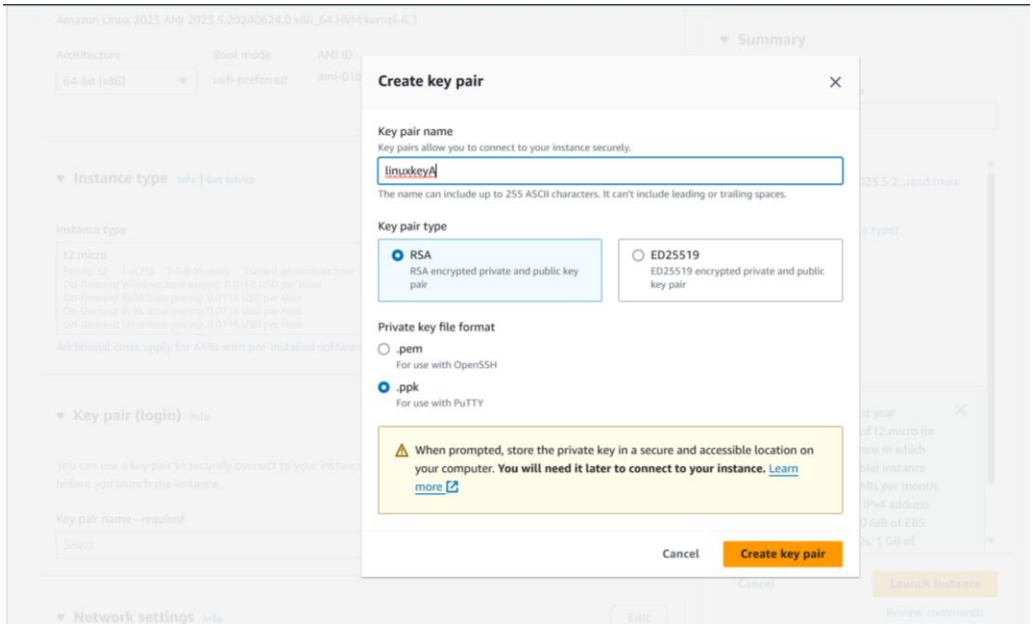


Select your AMI for your instance. For this example, we will be using Amazon Linux. Be sure it is Free tier eligible.



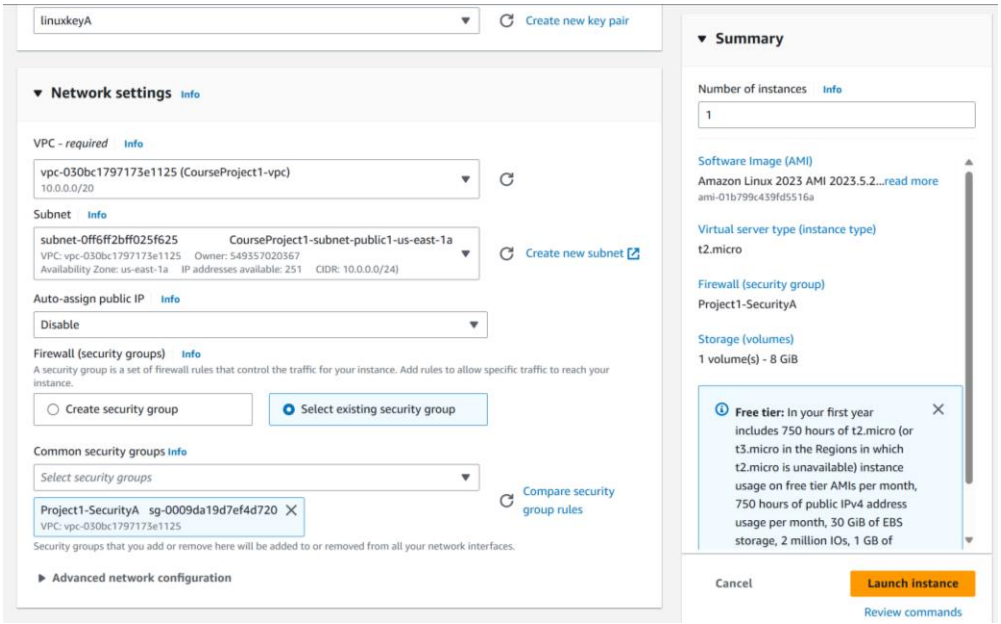
Select your instance type. In this case, you want t2.micro to be free tier.

Create a key pair for your instance.

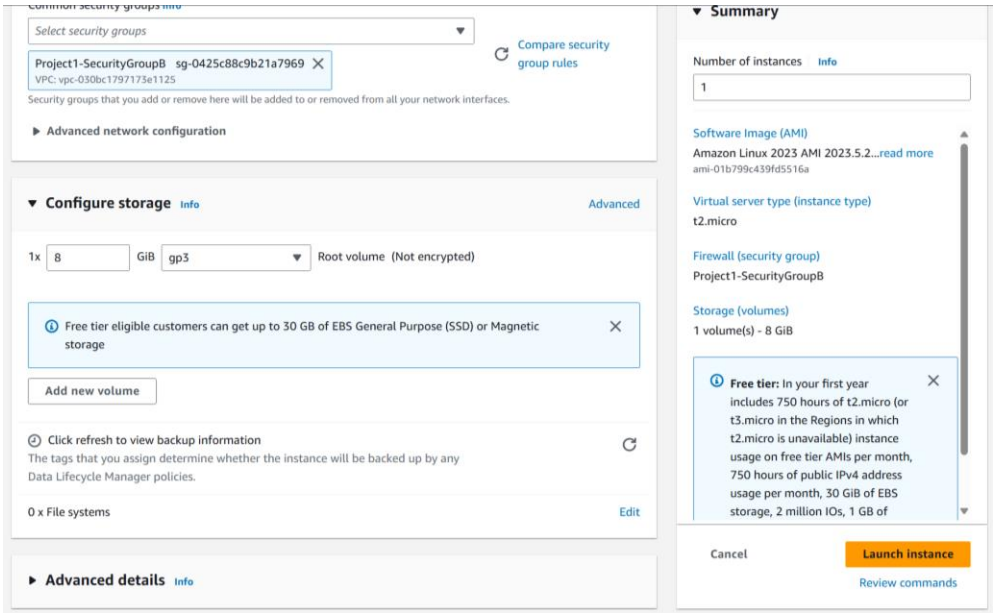


Create a name for your key pair. For the Linux AMI, we will be using PuTTY to log-in so have file format in ".ppk". Create key pair.

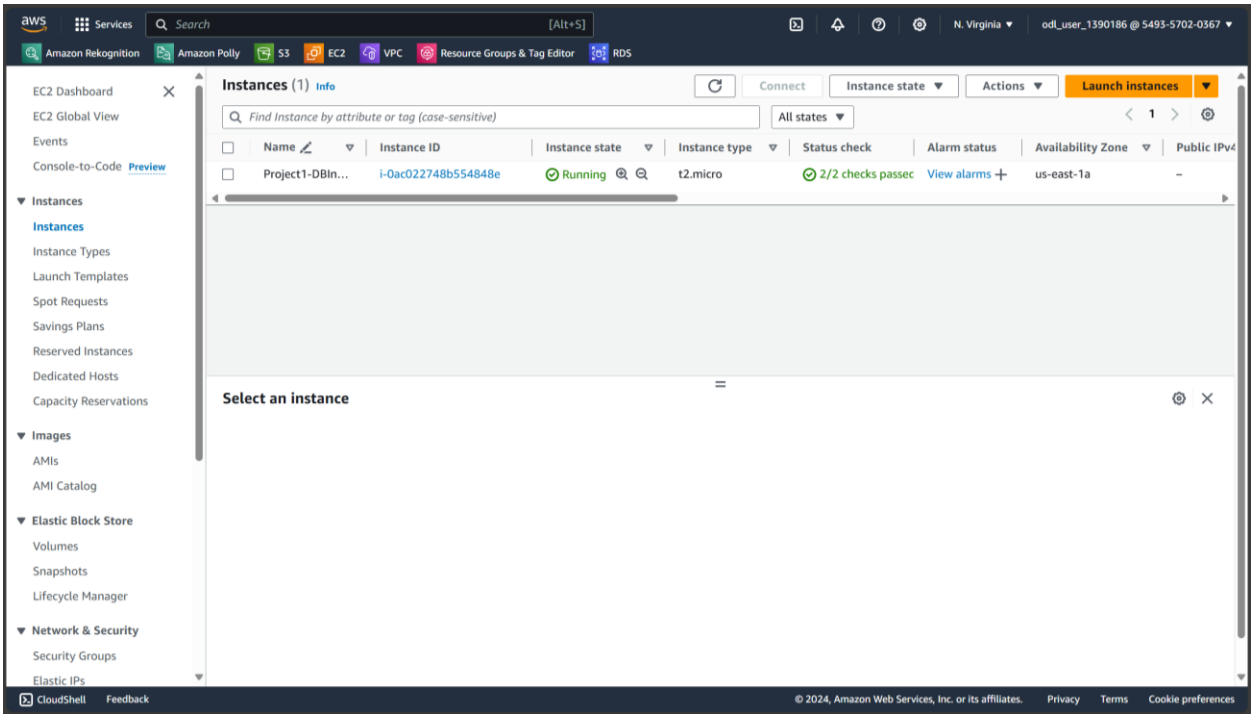
Go to Network Settings. Select Edit in order to connect to your VPC.



Select your VPC. For subnet, choose the public subnet of your VPC. For security group choose the security group you made from Step 5.

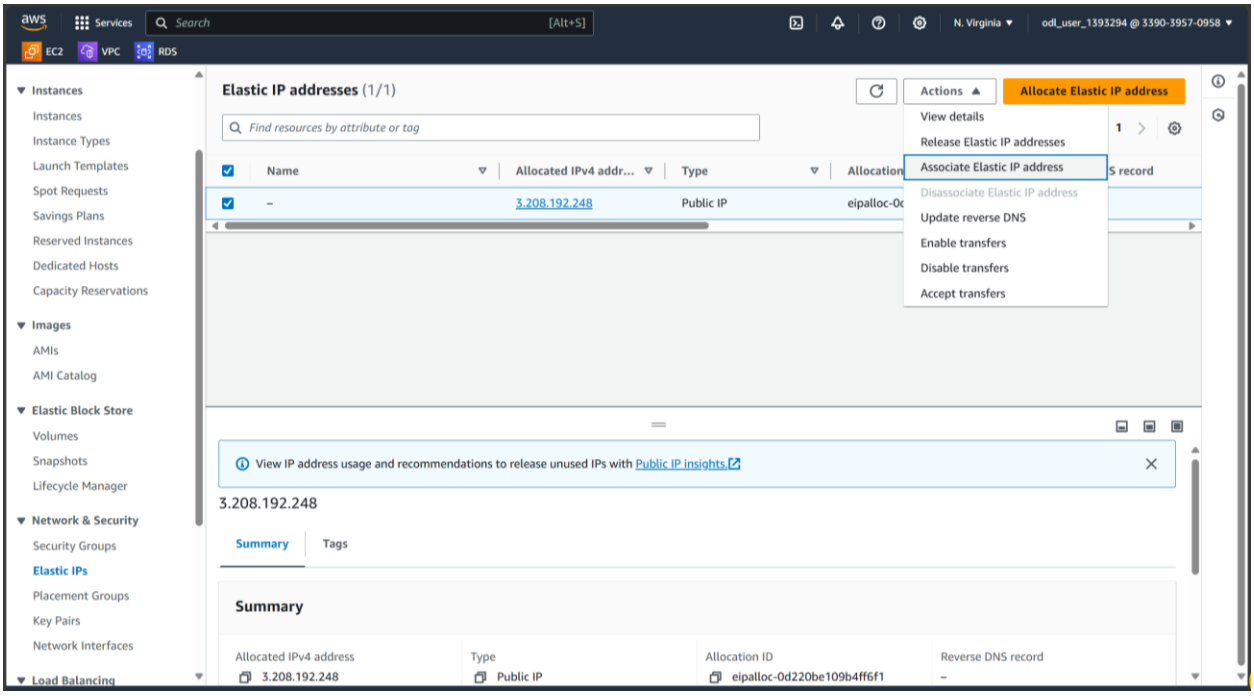


Have Configure Storage set at gp3. Review all your settings. End with “Launch instance”.

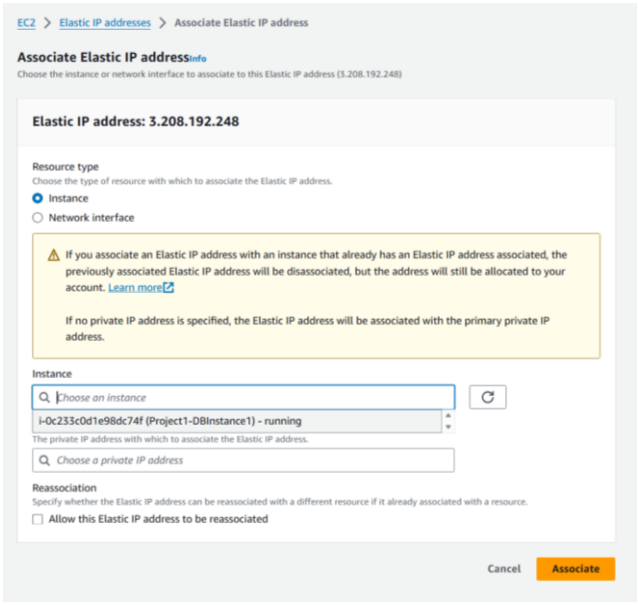


In a few minutes, the instance should have “Running” for Instance State and “2/2 checks passed” for Status check.

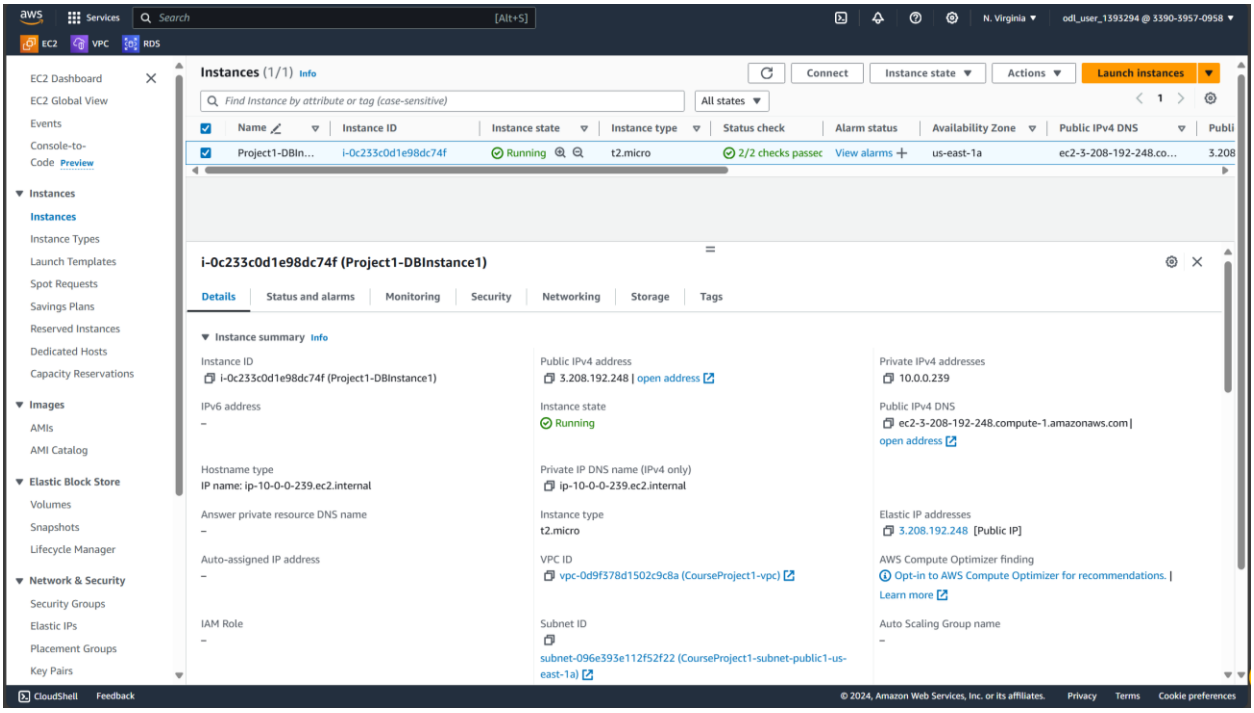
Returning to Step 1 to provide a public IPv4 address for instance



Go to Elastic IPs and select your elastic IP. Under Actions, select “Associate Elastic IP address.”



Select Instance and select the instance you want to associate the IP address with. Once selected, click on “**Associate**”.

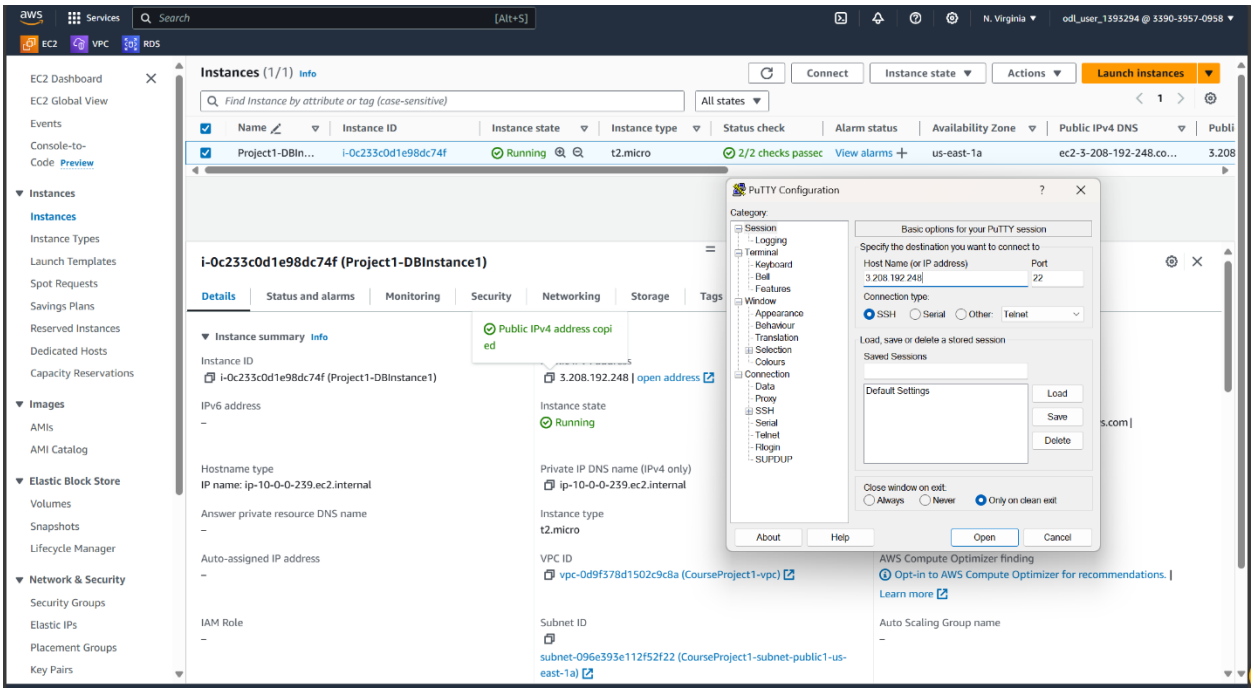


Now your instance has a public address from elastic IP.

Step 9: Run VPC on PuTTY

PuTTY App

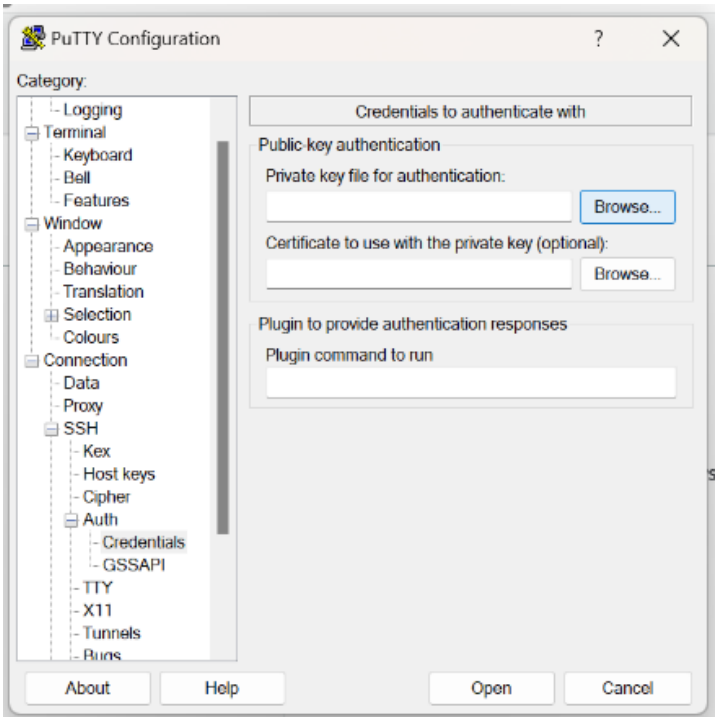
Open the PuTTY app.

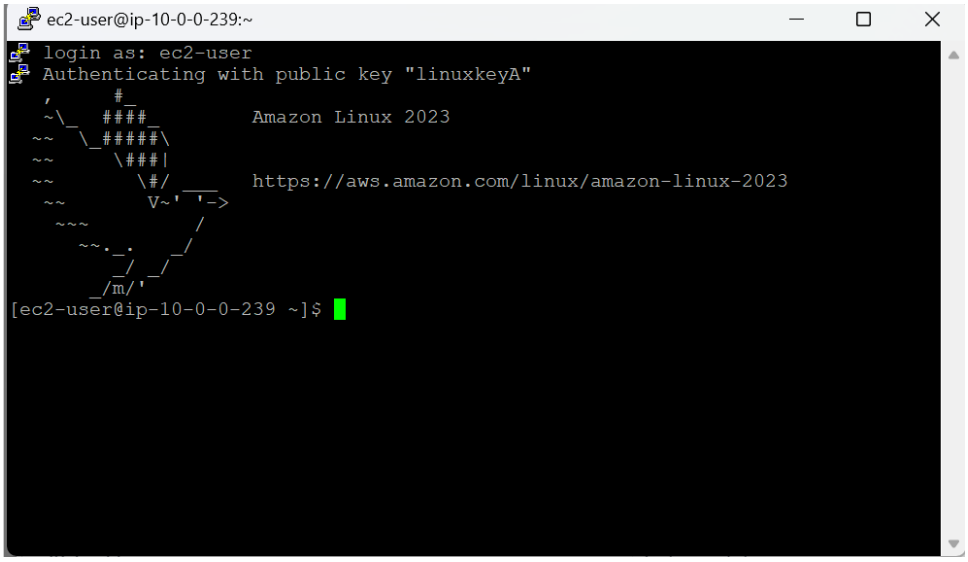


Be sure your instance is Running. Copy your public IP and paste in PuTTY in Host Name

Have it at Port 22 and Select SSH.

Under SSH, under Authorization and Credentials. Browse for your private key (ppk) for Linux and install it. Then click open.



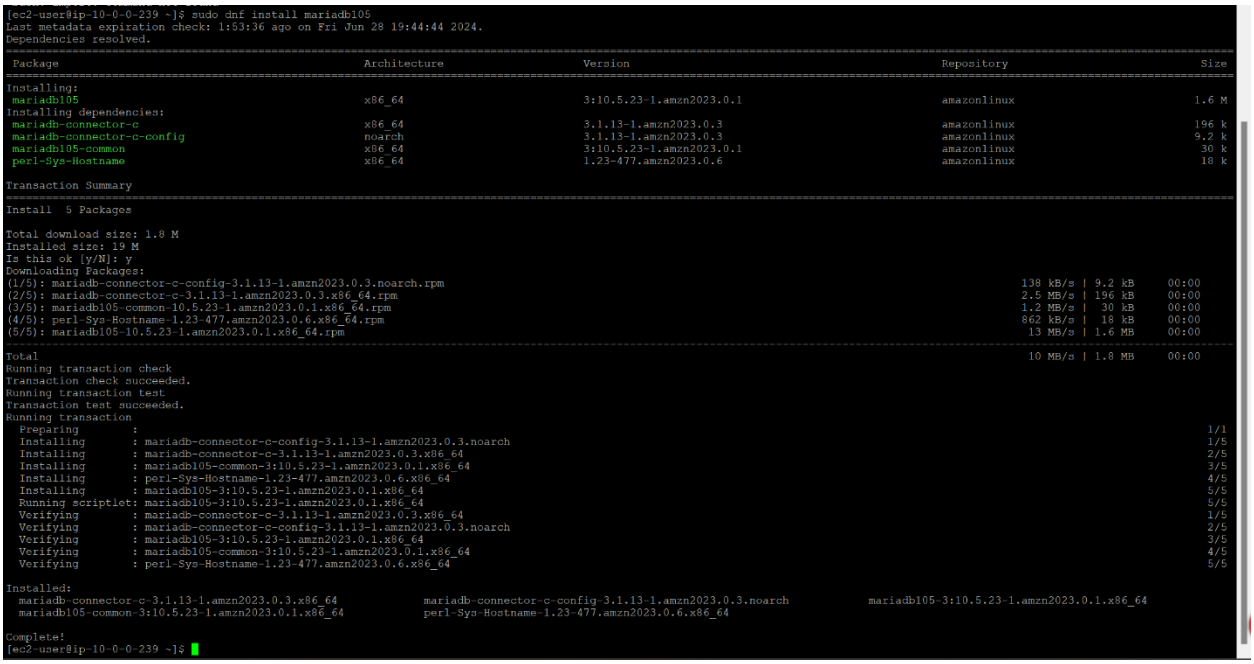


Once open type in your user: ec2-user

Download SQL

Following these steps from AWS:
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ConnectToInstance.html#mysql-install-cli

- sudo dnf install mariadb105 to install MySQL command-line client on Amazon Linux 2023
- be sure to type “y” for yes to install



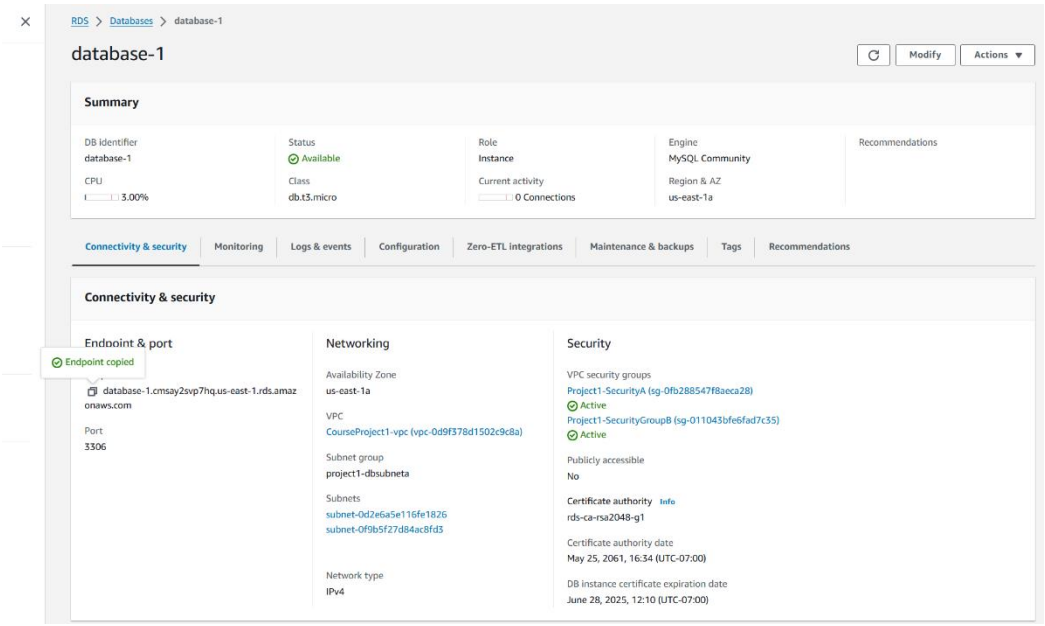
To check if mySQL is installed run: mysql –version

```
[ec2-user@ip-10-0-0-239 ~]$ mysql --version
mysql  Ver 15.1 Distrib 10.5.23-MariaDB, for Linux (x86_64) using  EditLine wrapper
[ec2-user@ip-10-0-0-239 ~]$
```

You can run “man mysql” to get a manual on you MySQL

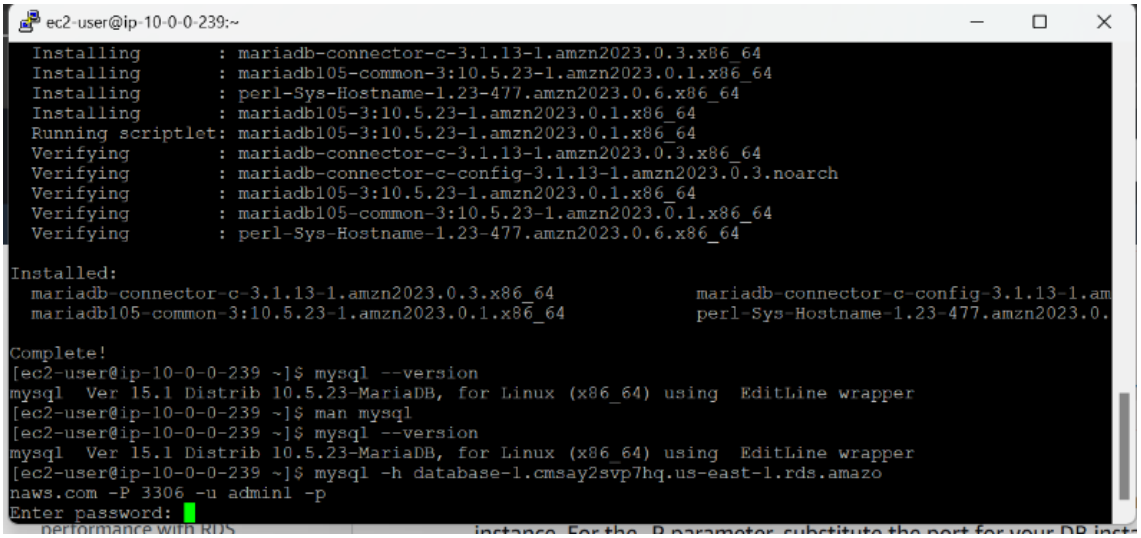
Connect DB

You can follow these steps from AWS
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ConnectToInstance.html

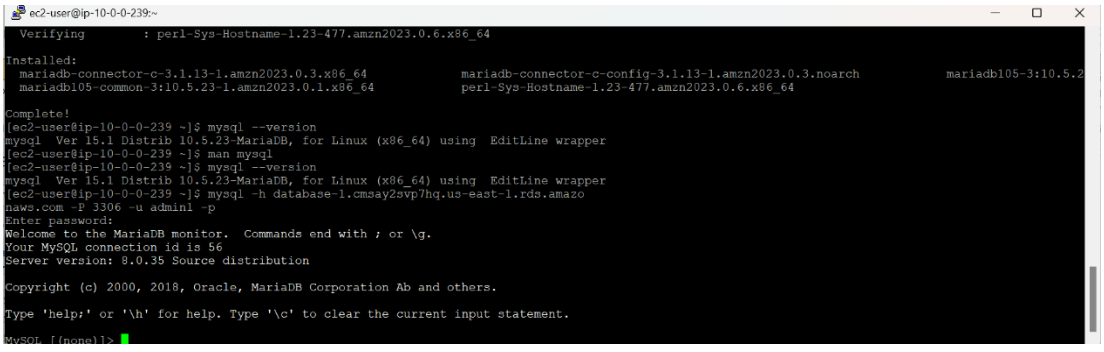


Copy the endpoint address of your database. Then in the terminal type in:

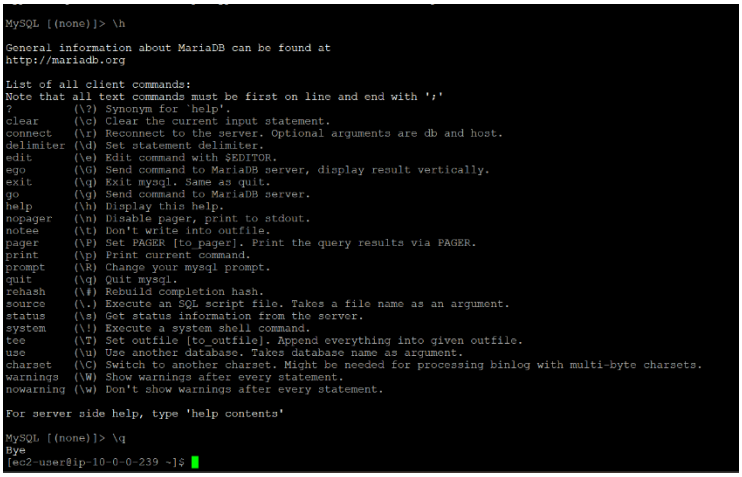
mysql -h *endpoint_address* -P *port_number* -u *master_username_for_DB* -p



Type in your password for administrator of your DB.



You are connected to your MySQL database through terminal.



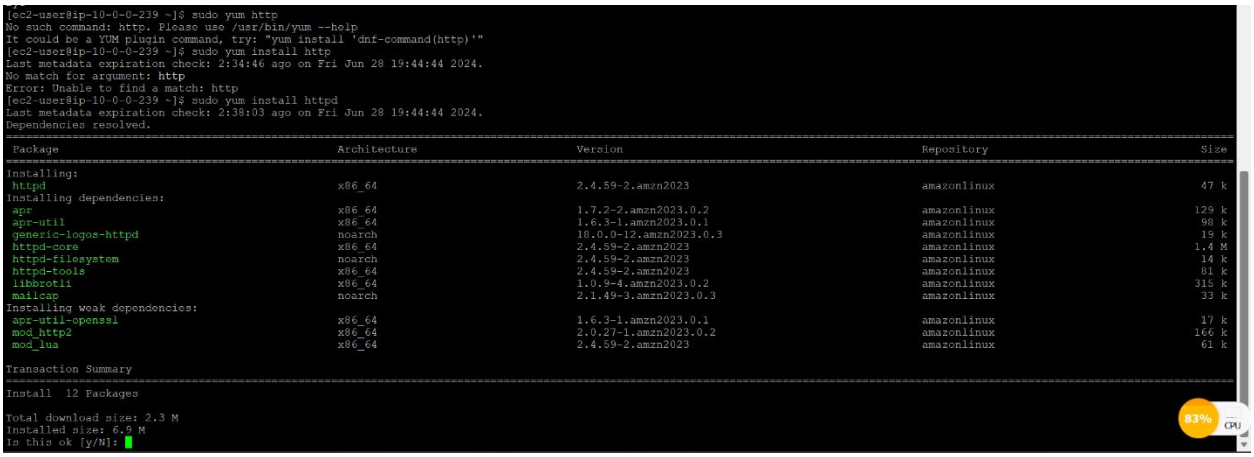
Can run \h as an option.

For HTTP Access

Run Http/Apache Web services to check if you can access it on browser.

Directions: [Compiling and Installing - Apache HTTP Server Version 2.4]

Run: sudo yum install httpd



After installed, check your status of your httpd/Apache Web server.

Run: status httpd service

To enable program, run:

- sudo systemctl enable httpd
- sudo systemctl start httpd

HTTPD is enabled and allow access to database by copying your IPv4 address of instance and paste it on your browser.

The following steps are how to create a VPC and connect to an EC2 Instance and Database.

Finish