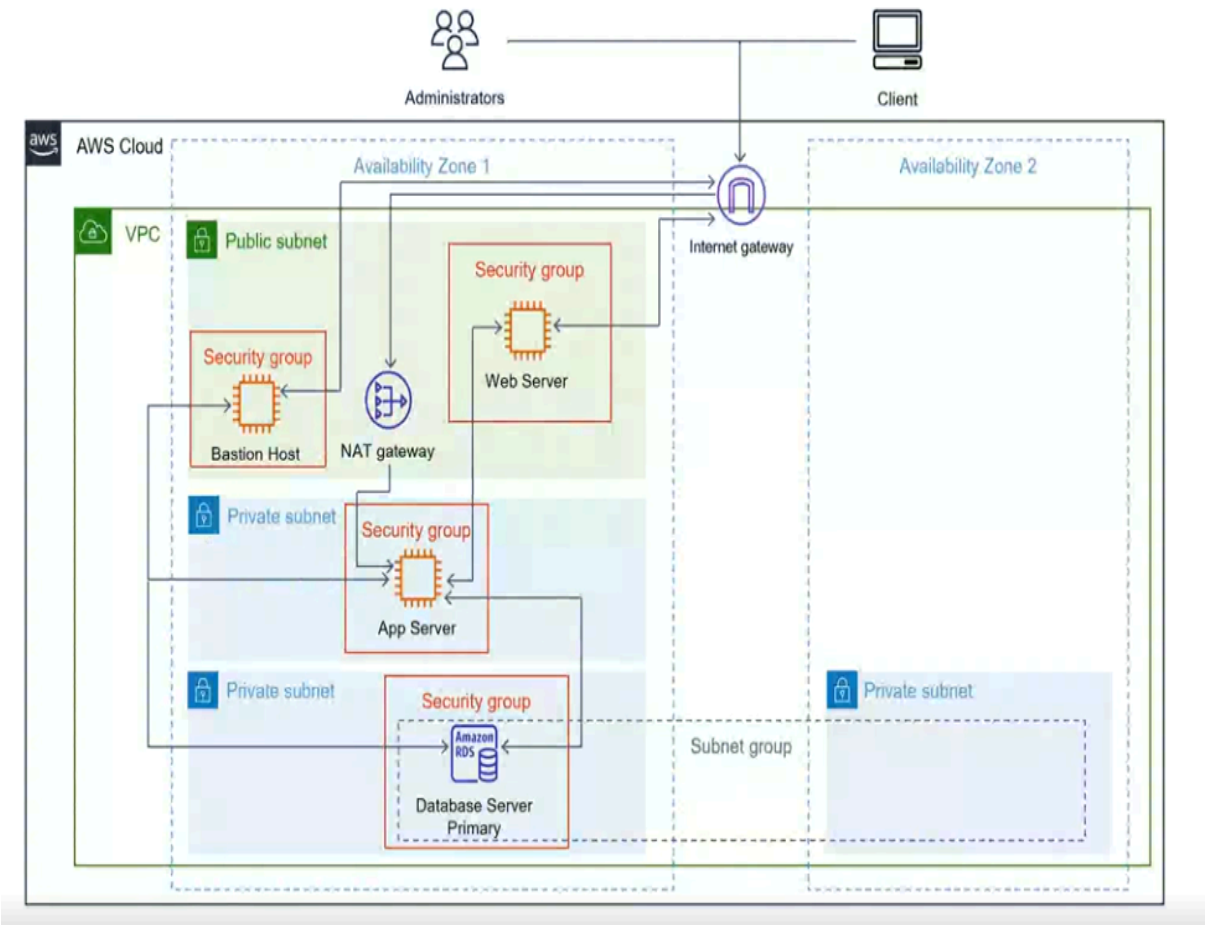
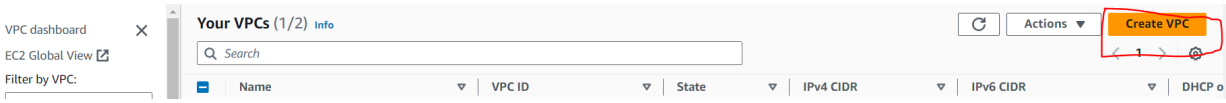
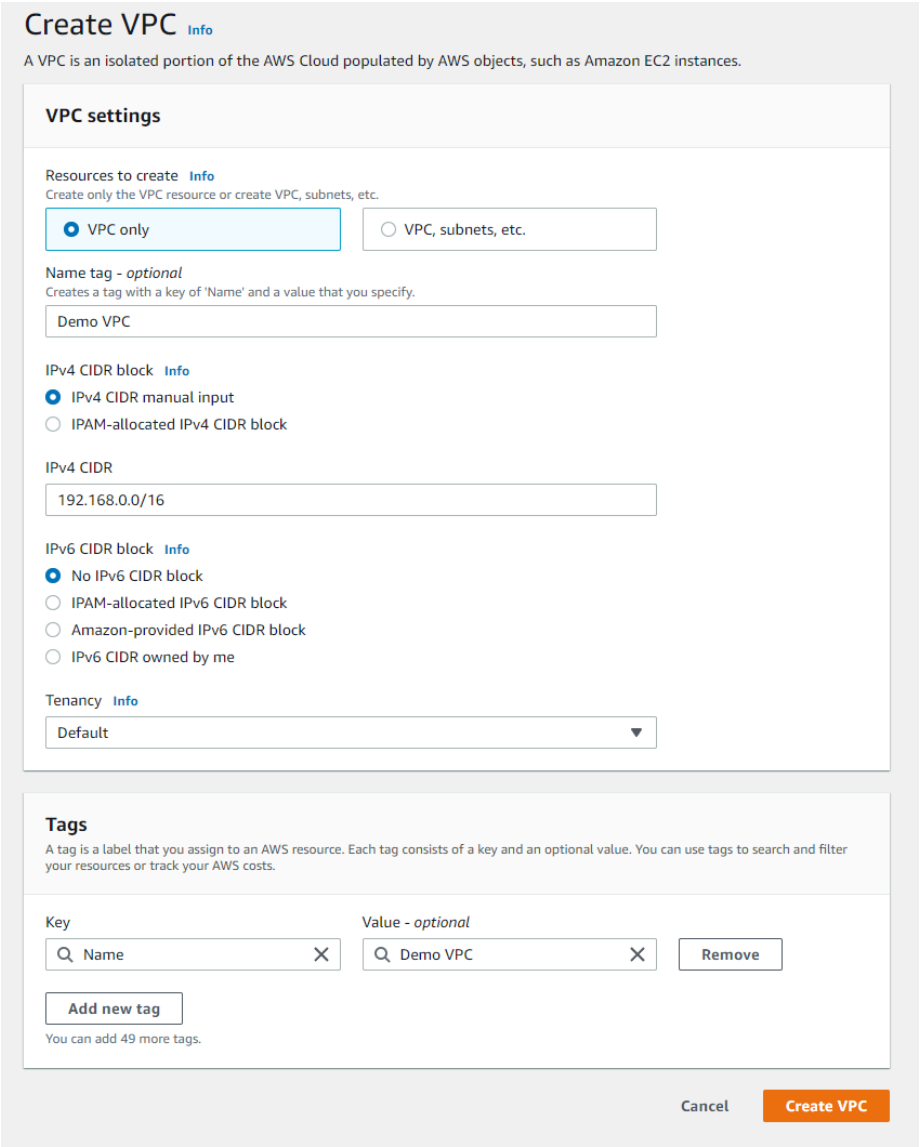


Step 1: Create a VPC and Subnets as well as routing and security groups

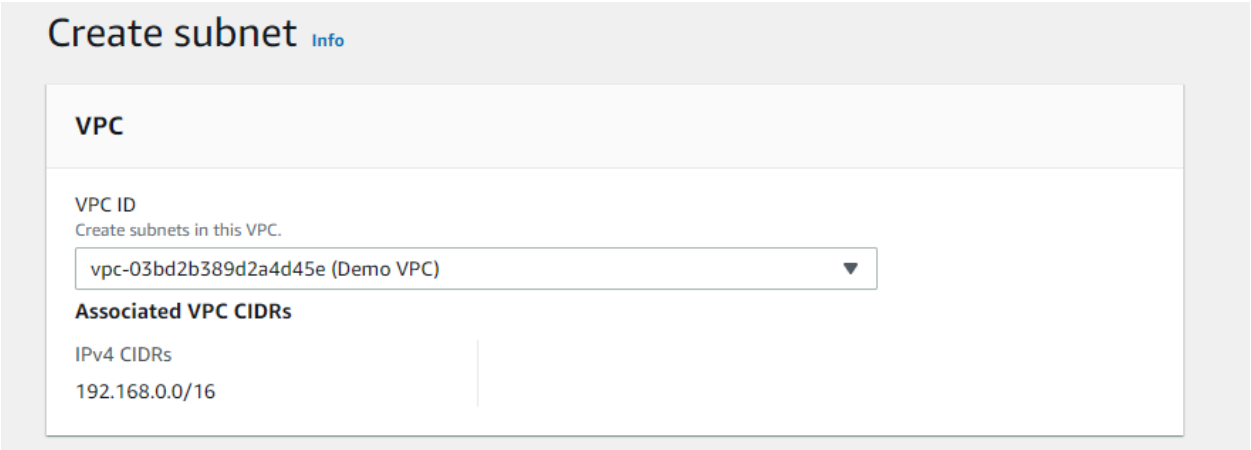
- Go to “Your VPCs” from the VPC service on the AWS management console and click on the orange “Create VPC” button.



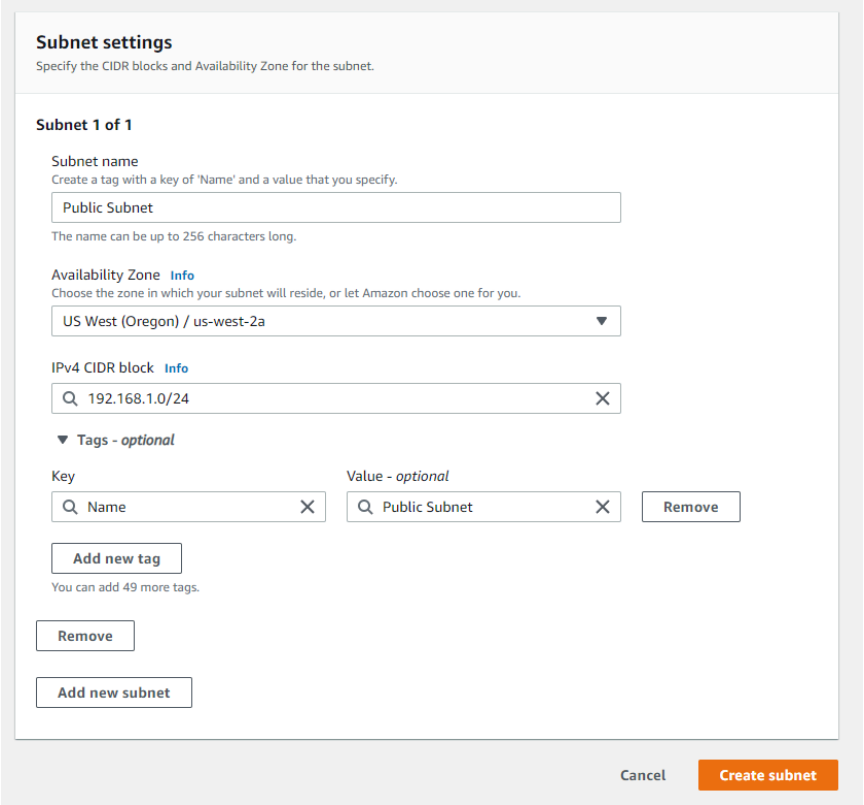
- Only create a vpc here and give it a name. You are free to make your own name or follow along with the one put here
- GIve it a 192.168.0.0/16 CIDR block and leave everything else as default. Click create.



- To create your subnets go to Subnets on the left hand side of the VPC service and click on it
- Add your VPC ID to where it asks



- Assign it a name letting you know what it is your first public subnet
- Put it in any availability zone and give it a CIDR of 192.168.1.0/24



- Add a second subnet and name it Private Subnet 1 or something to let you know it is your first private subnet
- Put it in the same availability zone as the first subnet you made and give it a CIDR of 192.168.2.0/24

Subnet 2 of 2

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

Private Subnet 1

The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

US West (Oregon) / us-west-2a

IPv4 CIDR block Info

192.168.2.0/24

Tags - *optional*

Key	Value - *optional*	
Name	Private Subnet 1	Remove

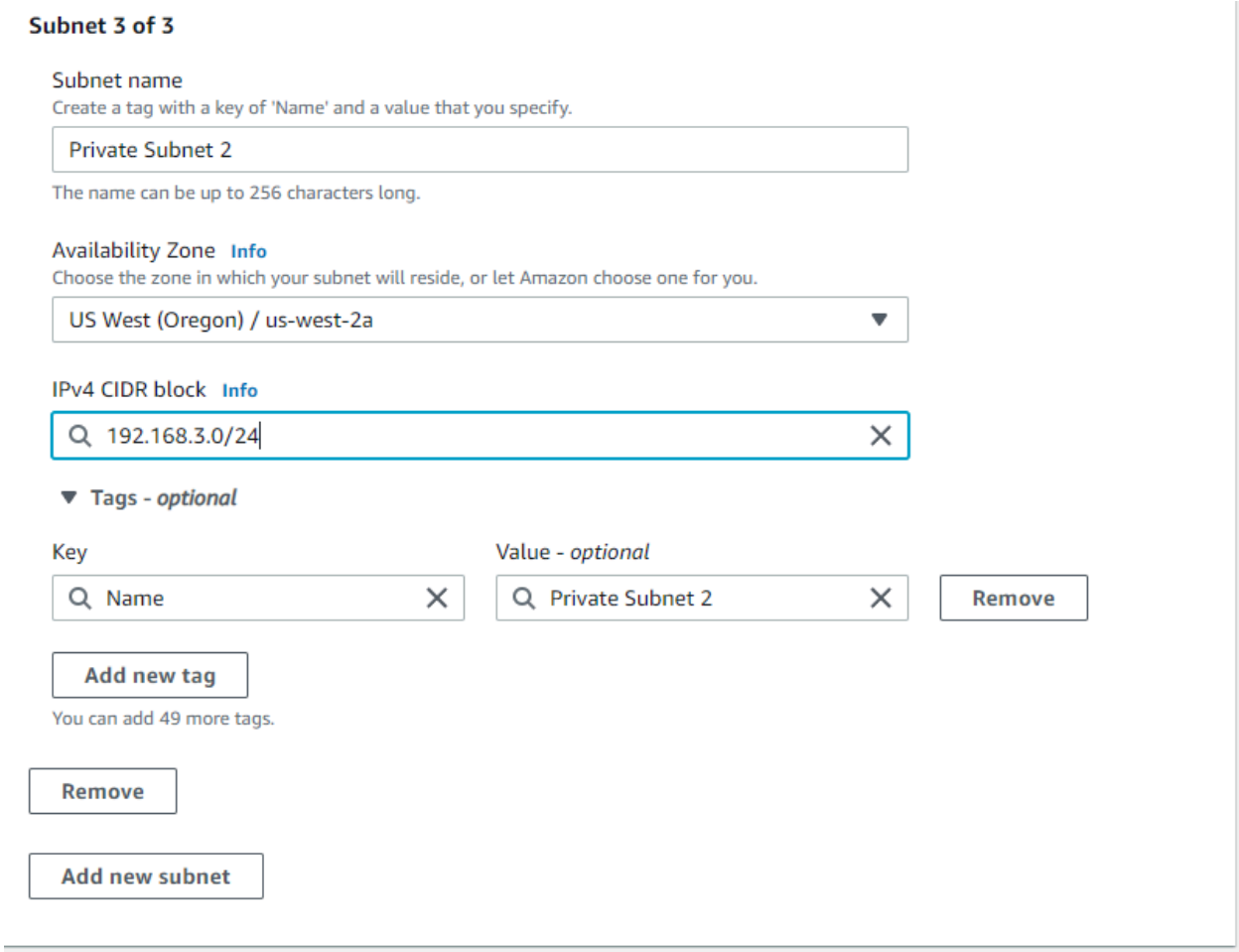
Add new tag

You can add 49 more tags.

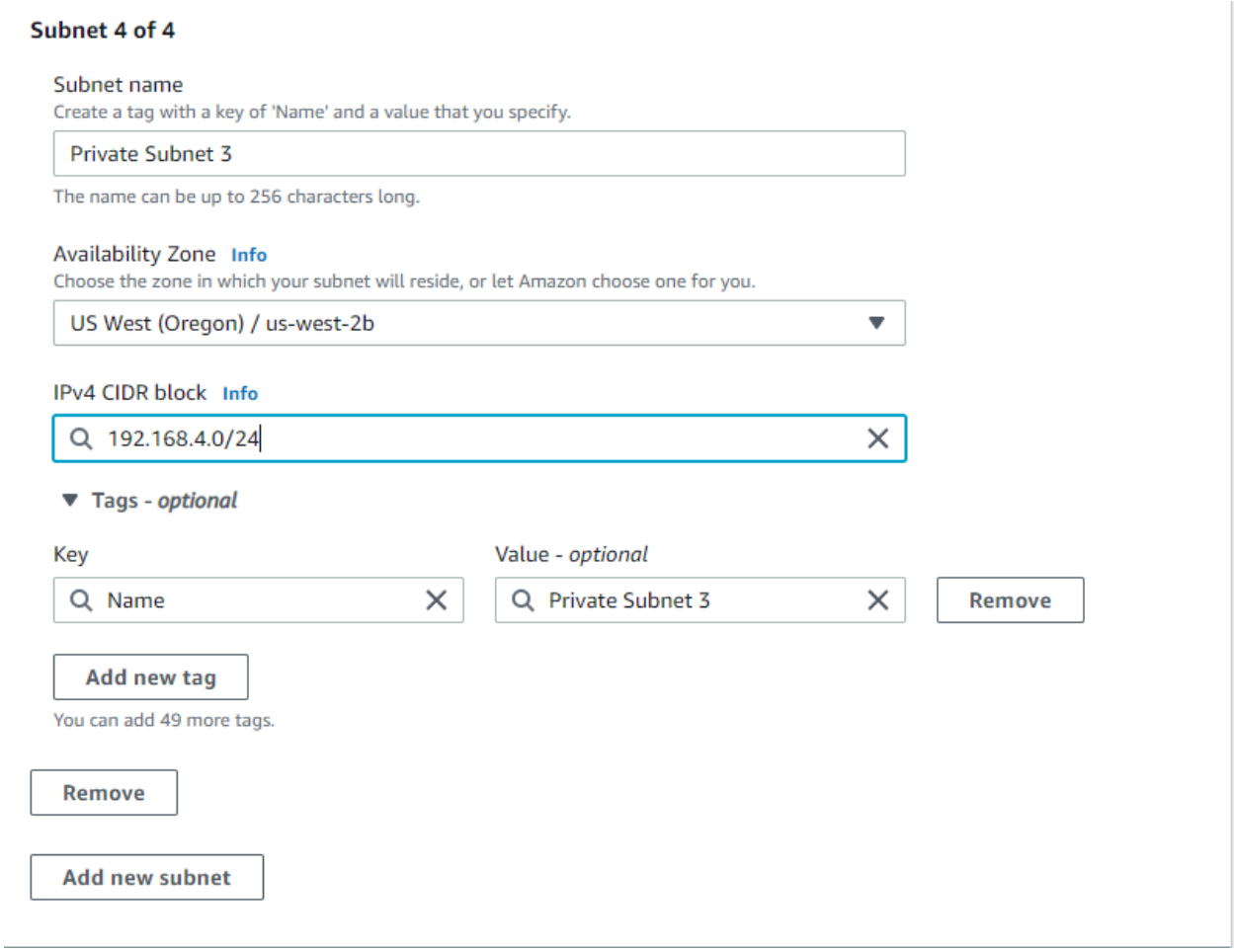
Remove

Add new subnet

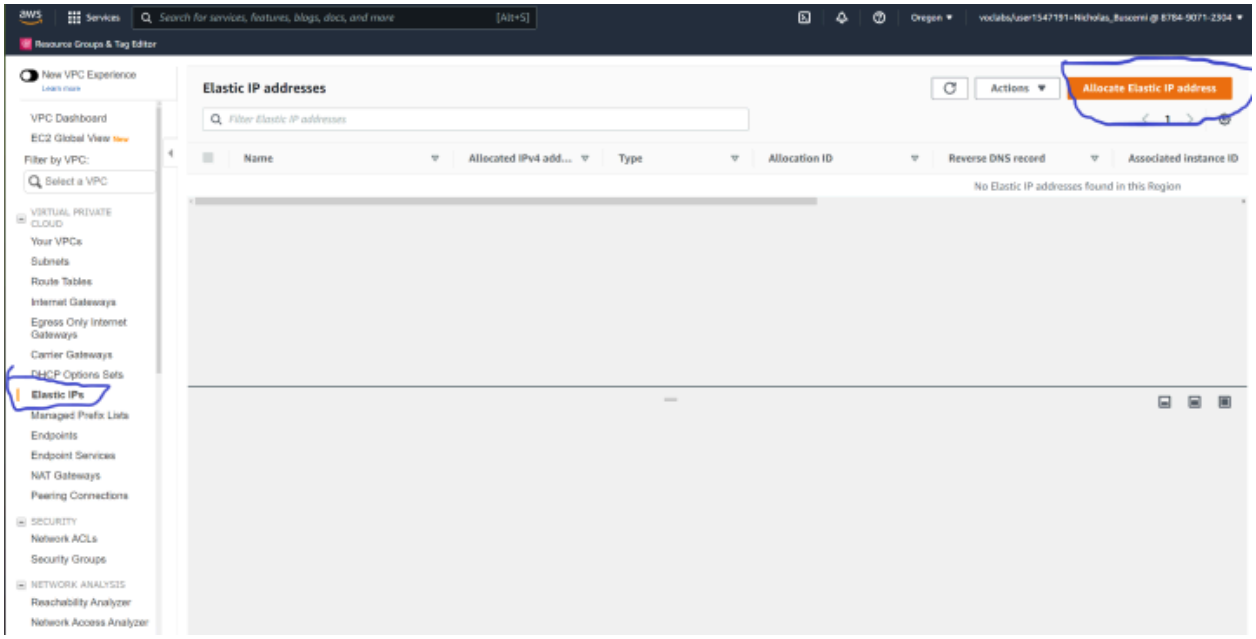
- Add a third subnet and assign a name letting you know it is the second private subnet you will be making
- Put it in the same availability zone as your first public subnet and give it a CIDR of 192.168.3.0/24



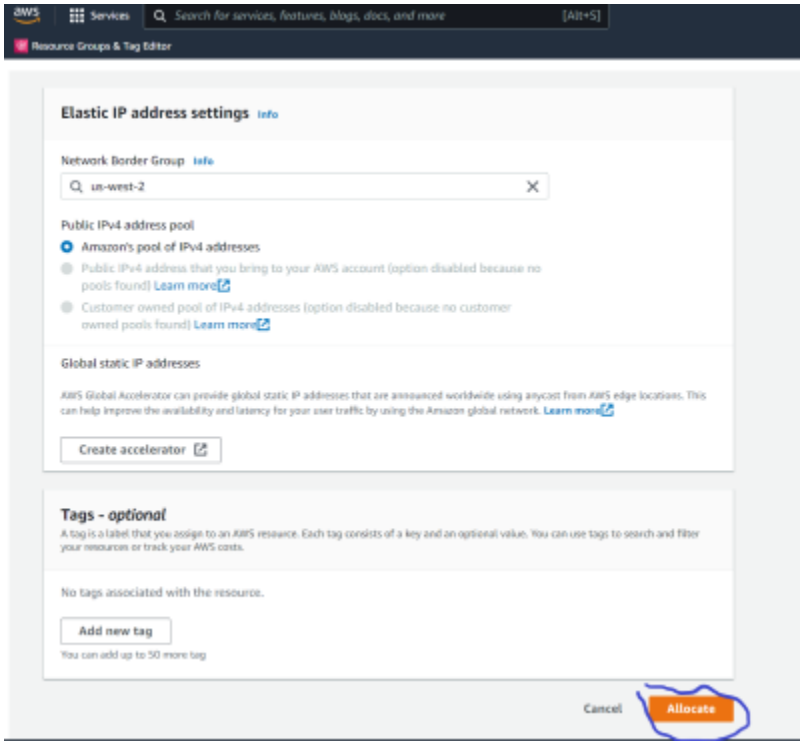
- Add a fourth and final subnet and give it a name letting you know it is the third private subnet
- Put it in a different availability zone from the rest of your subnets and give it a CIDR of 192.168.4.0/24



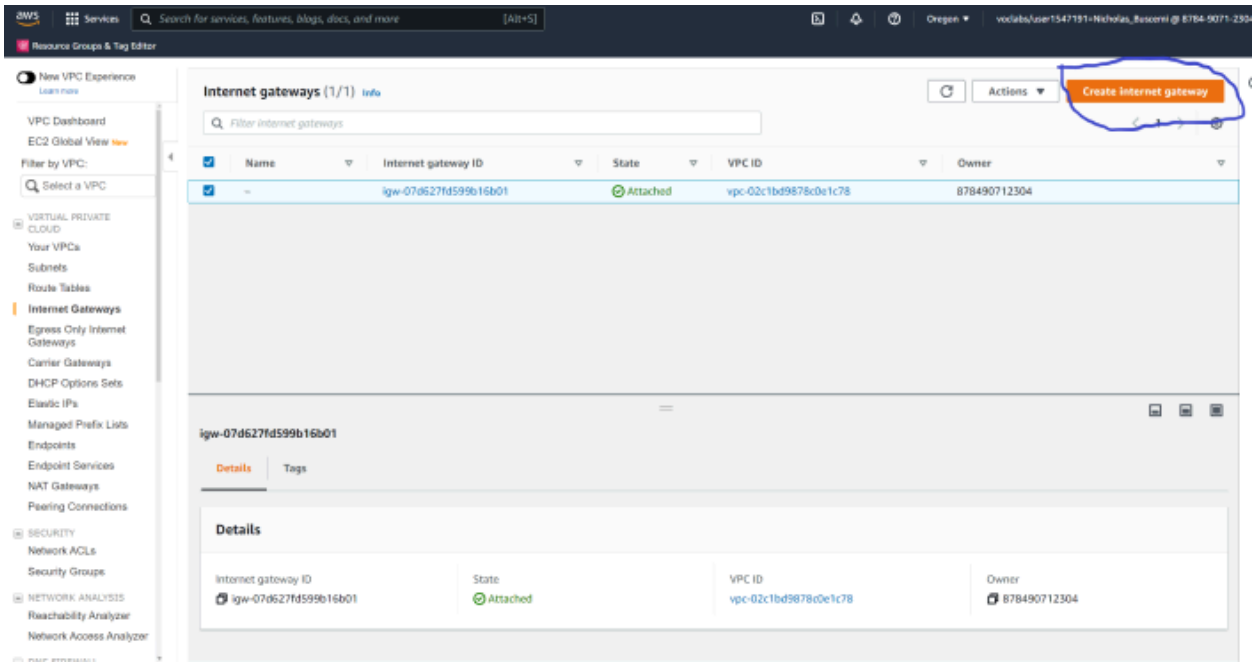
- Set up for route tables
- Allocate an Elastic IP address by going to Elastic IPs on the left hand side and click “Allocate Elastic IP address”



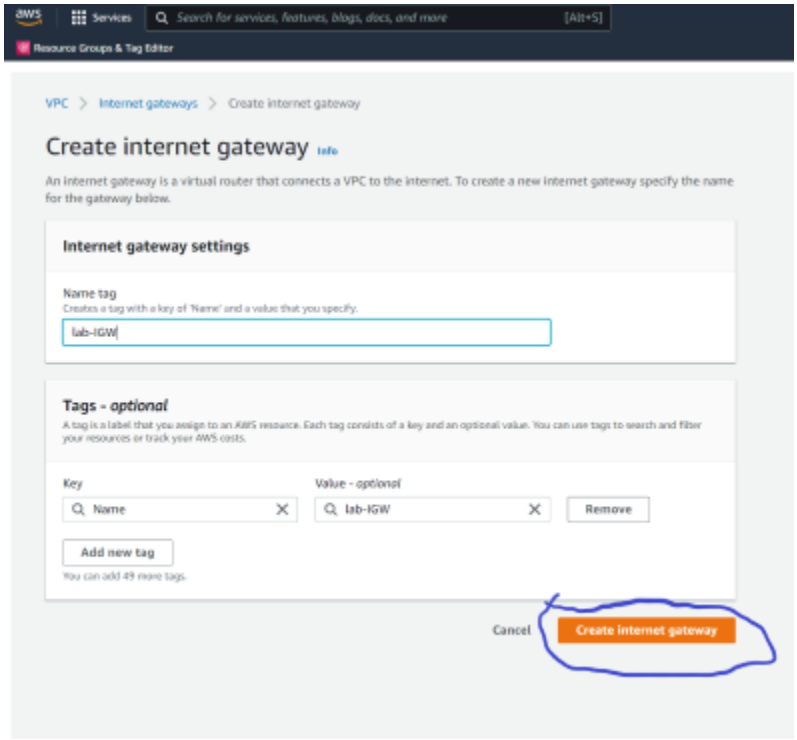
- Everything should be good as default but make sure that you are in the same region you have been creating everything in and then press “Allocate”. You can also add a name tag if you wish but it isn’t necessary



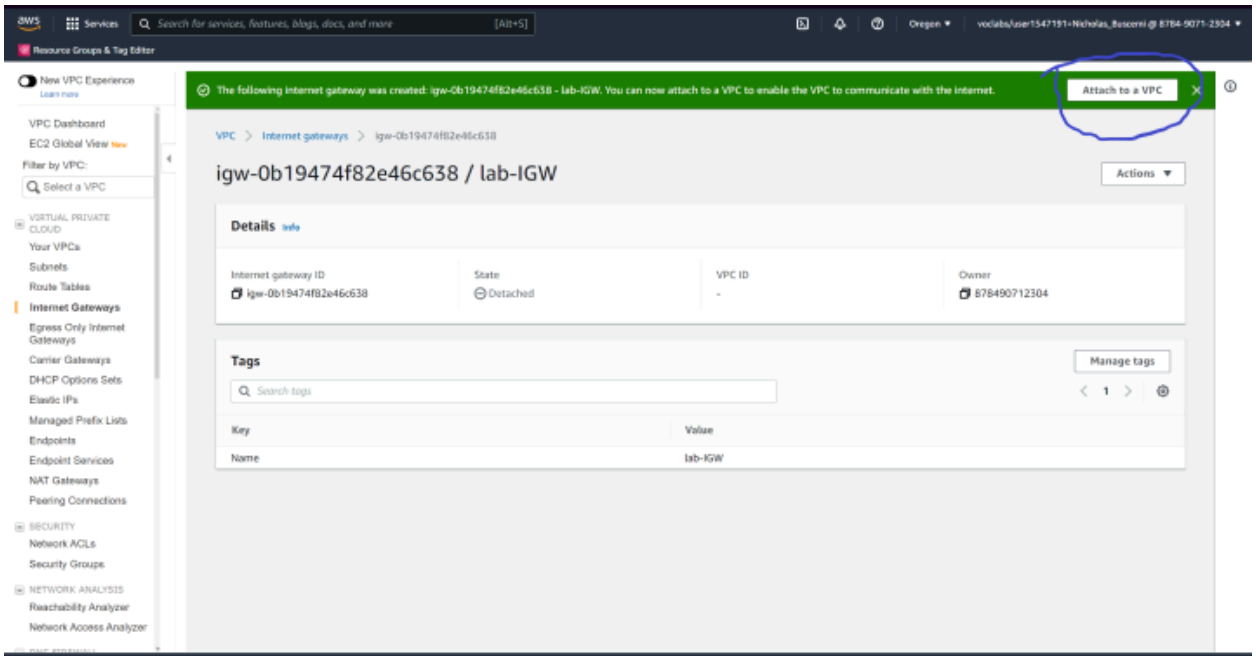
- Now create an internet gateway and attach it to the VPC by going to Internet Gateways on the left hand side and clicking “Create Internet Gateway”



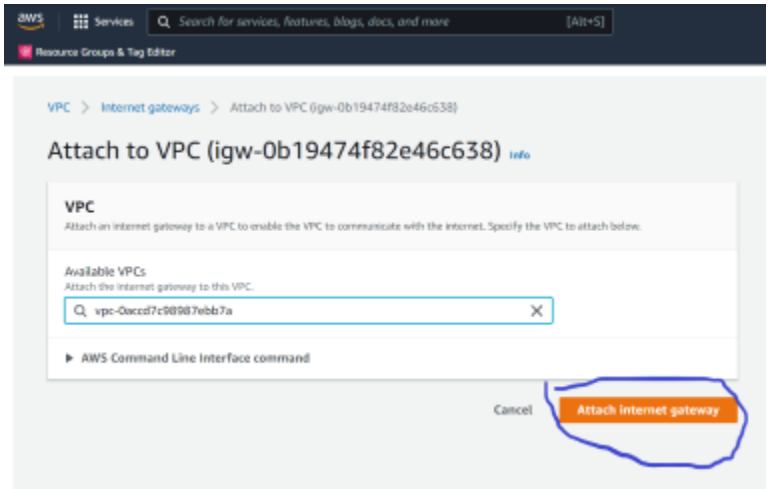
- Name it something similar to what is below and then click “Create Internet Gateway”



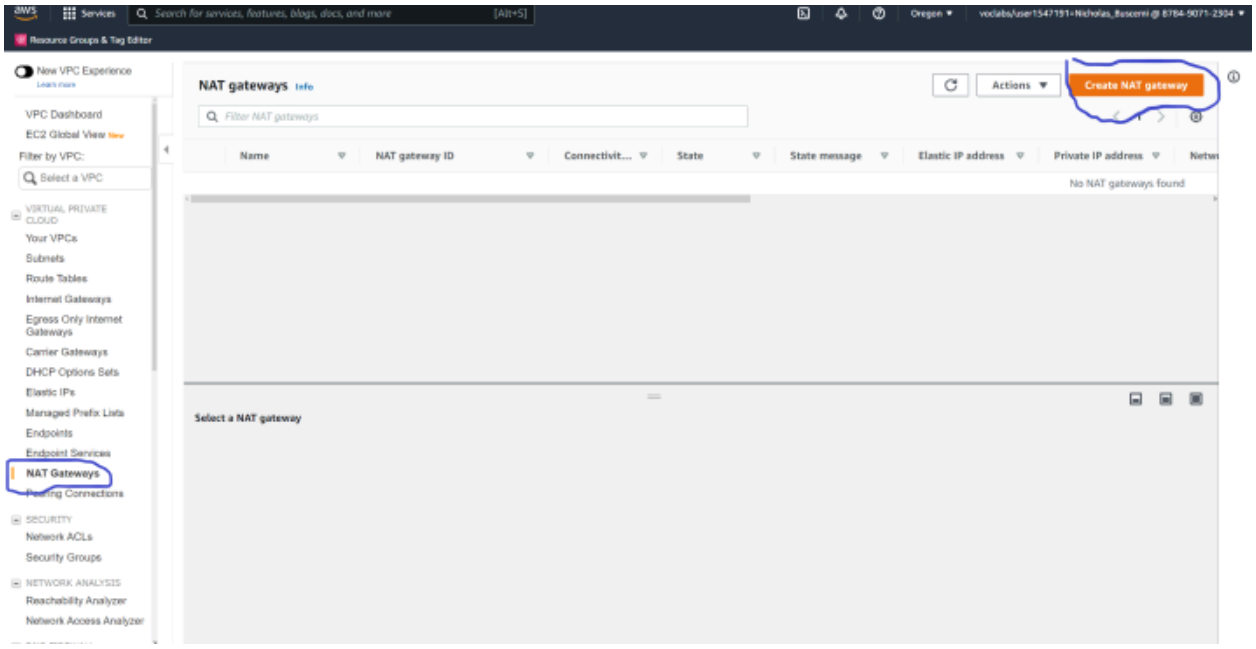
- Once it is created attach it to your VPC by clicking “Attach to a VPC” on the top of the screen



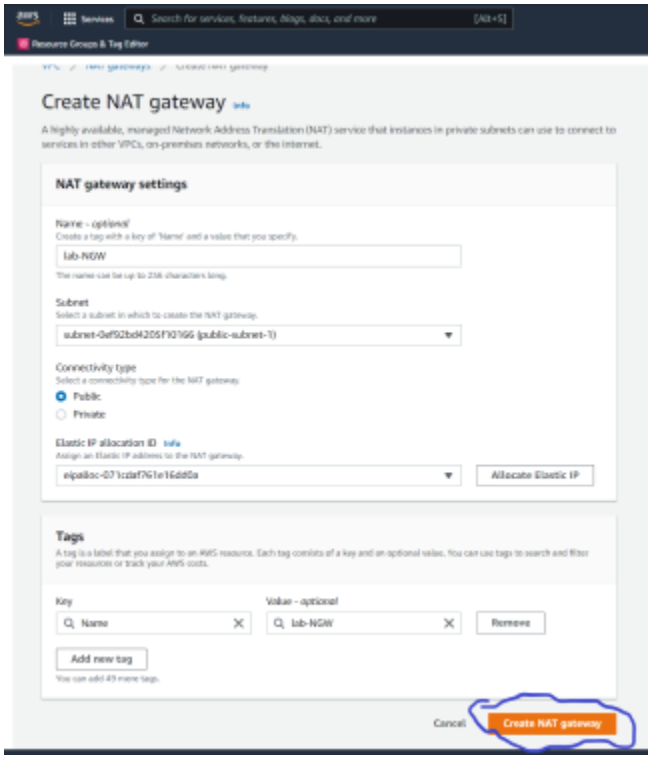
- Click the drop down and select your vpc that you made



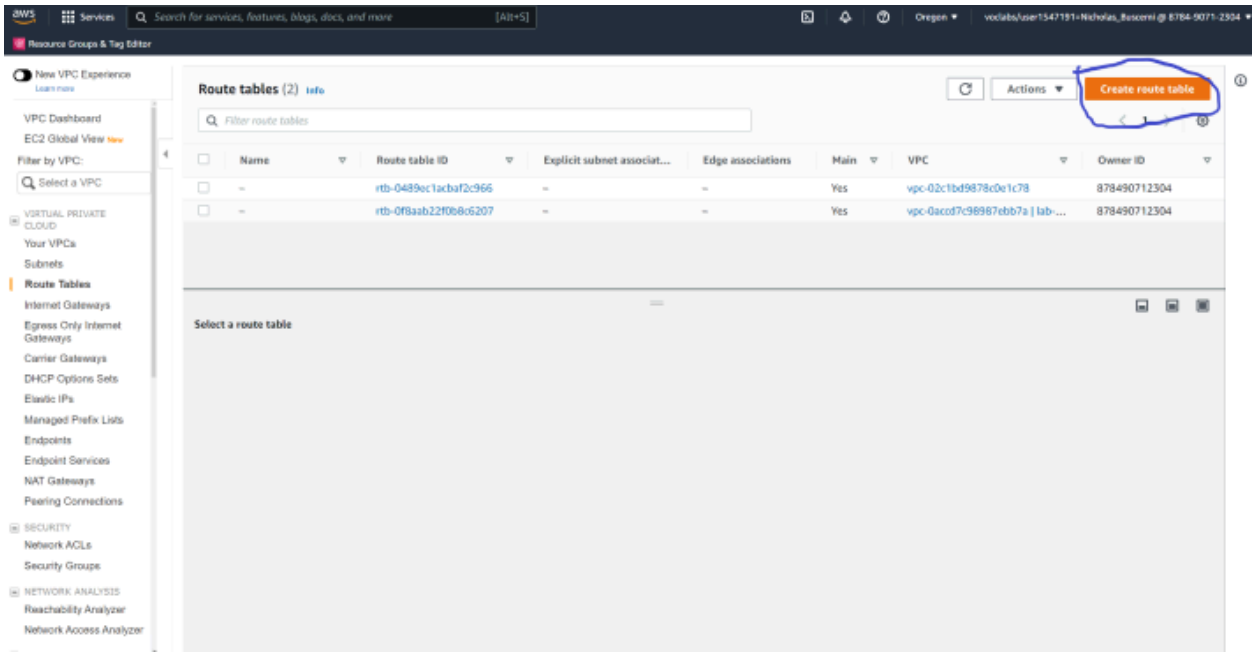
- Create a NAT Gateway by clicking on Nat Gateways on the left hand side and then clicking “Create NAT Gateway”



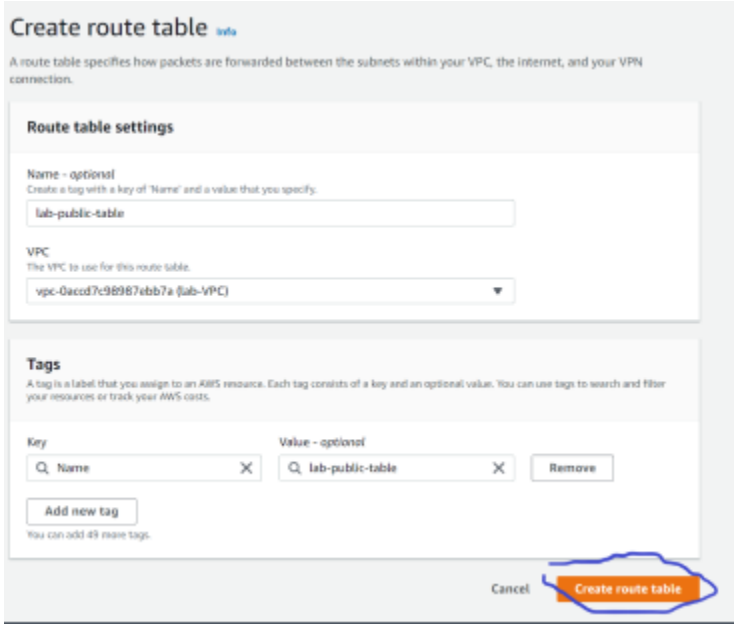
- Give it a name similar to the one below and assign it to a **public subnet**
- Click the drop down for Elastic IPs and click the one you created previously
- Click “Create NAT gateway”



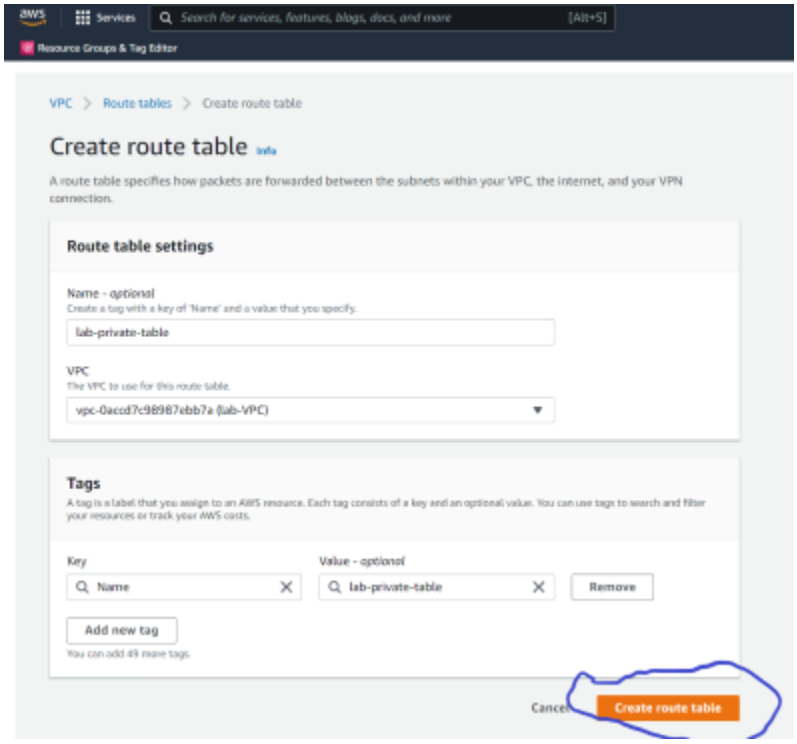
- Create Route Tables by first heading to “Route Tables” on the left hand side
- Click “Create route table”



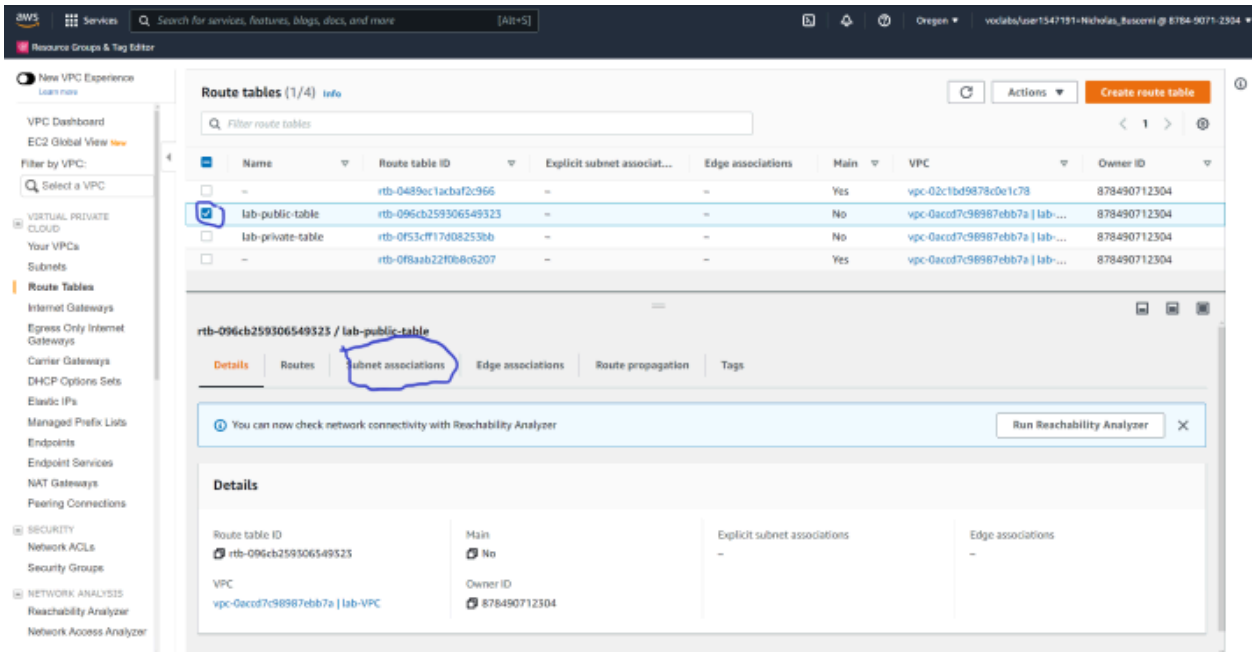
- Give it a name letting you know this is the public route table for your lab
- Assign your VPC to it and click “Create route table”



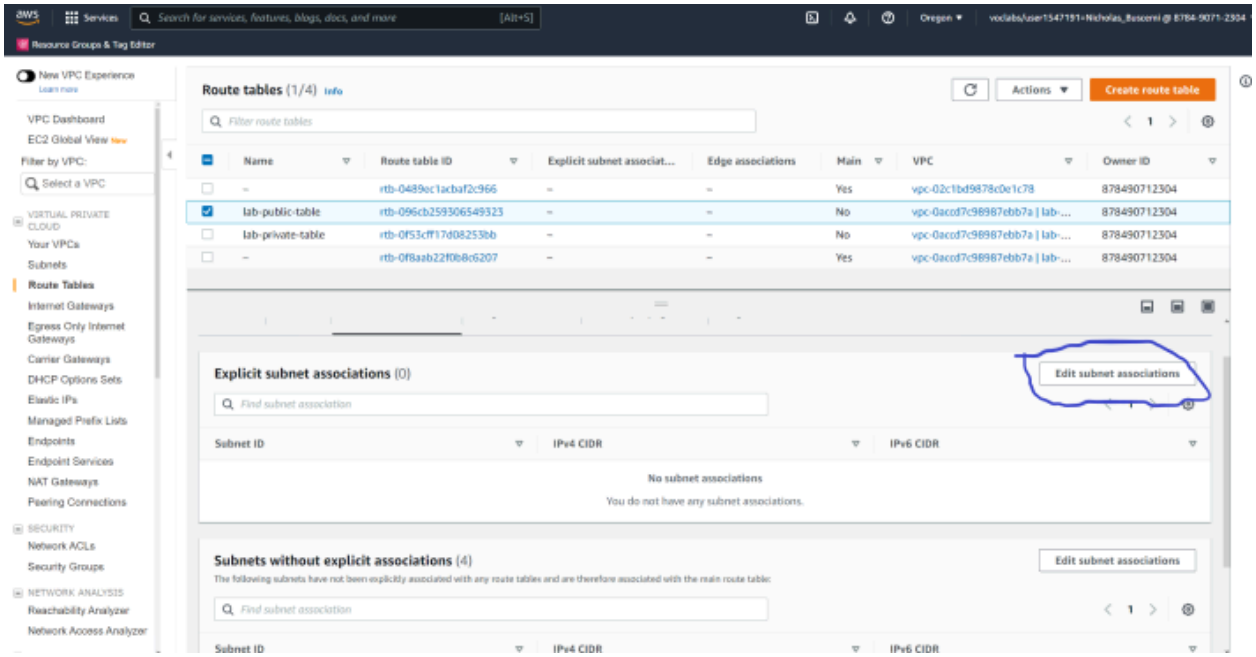
- Make a second route table naming it something to let you know that this is the private route table for your lab and assign your VPC to it



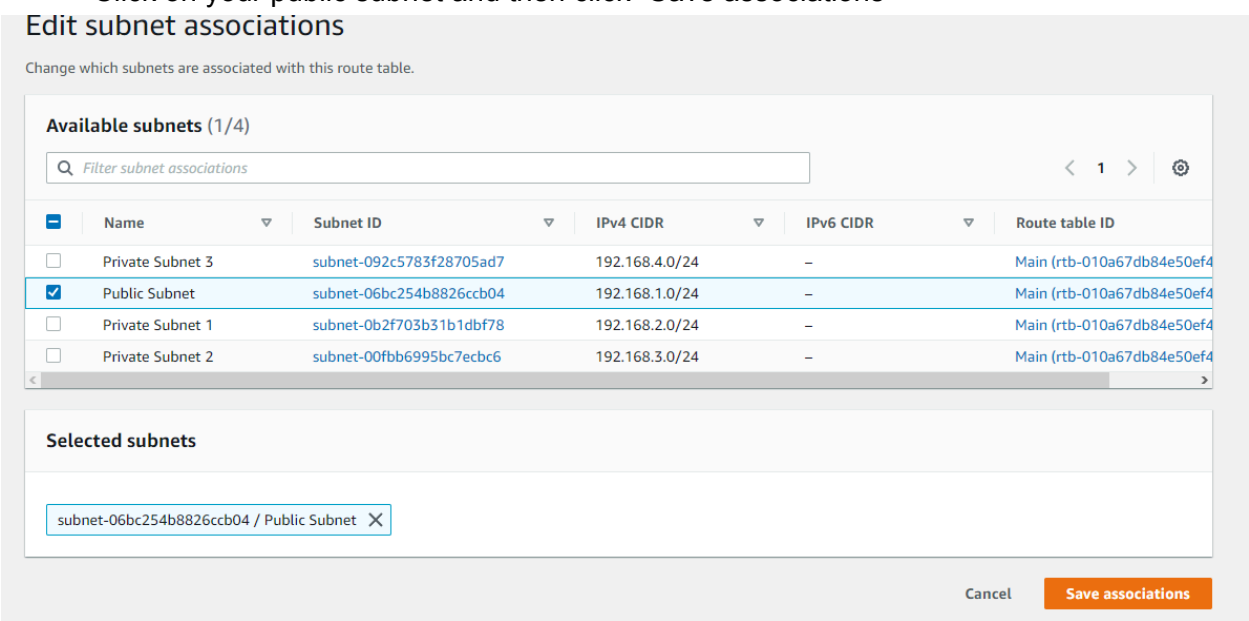
- Now associate your subnets with their respective route table
- Click on the public route table and click on “Subnet association” next to “Details”



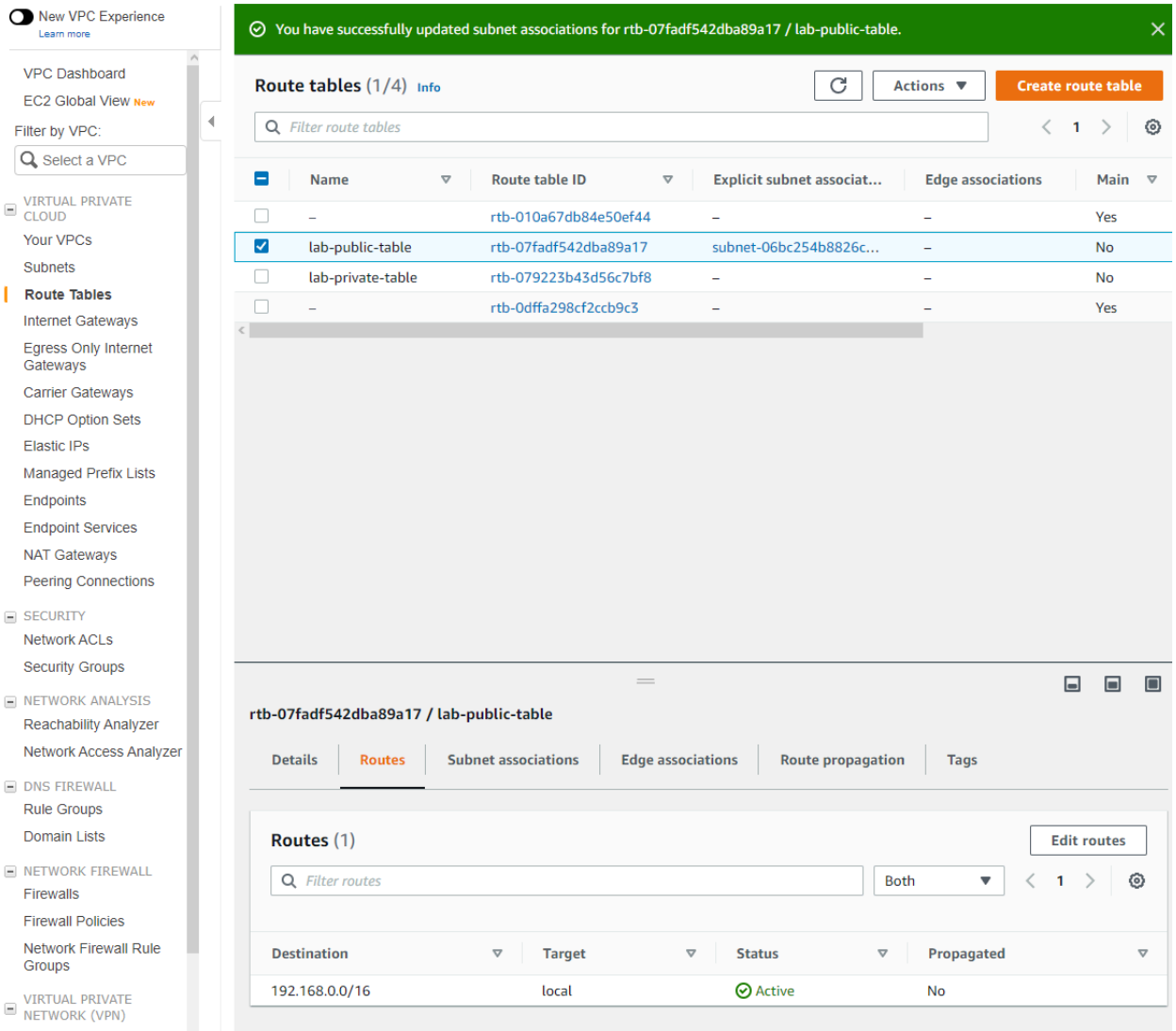
- Click on “Edit subnet associations”



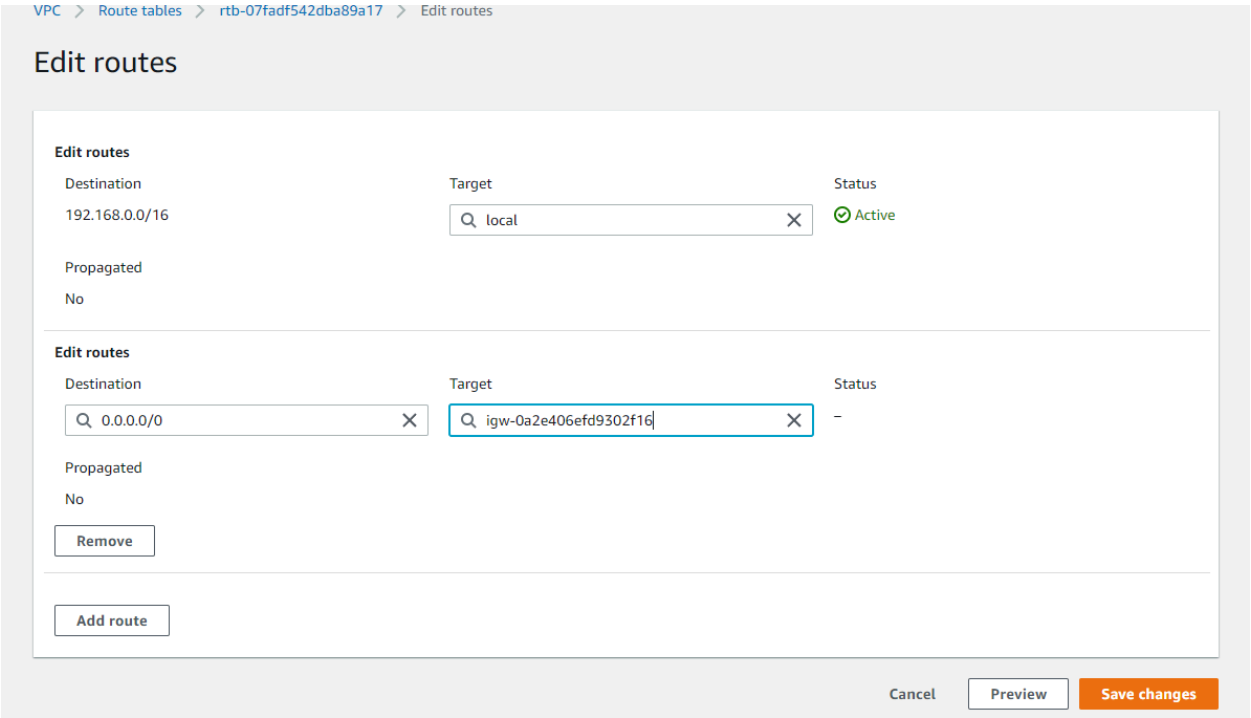
- Click on your public subnet and then click “Save associations”



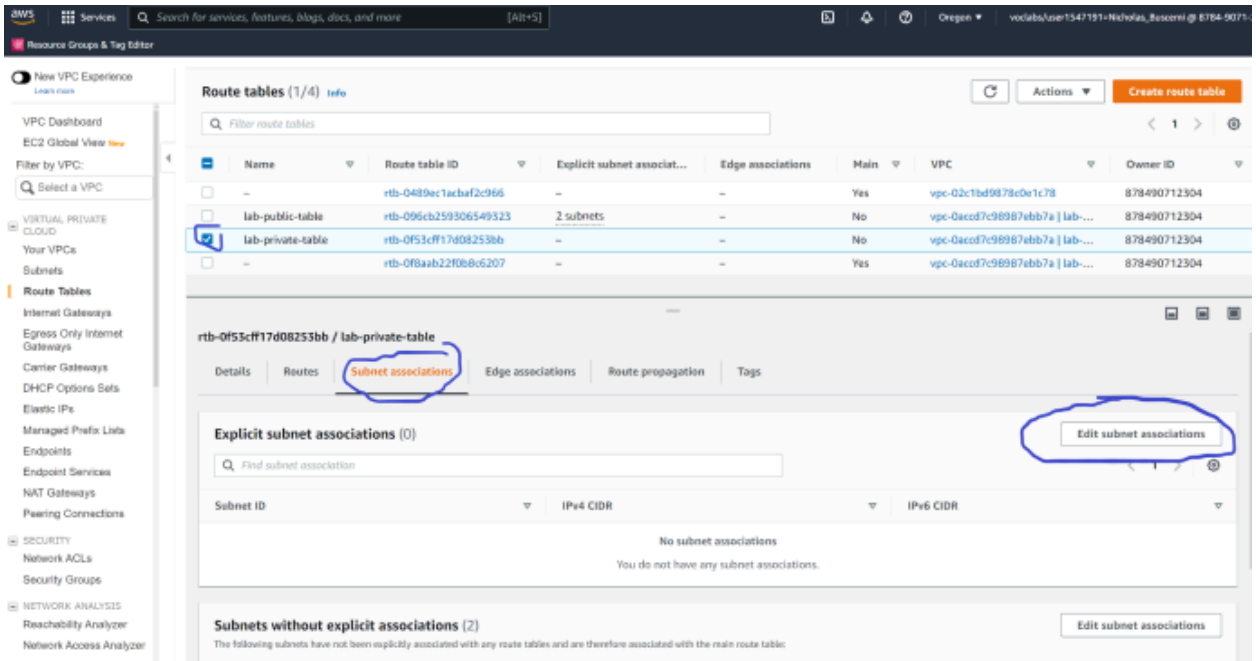
- Now add a route to our public route table to get access to the internet gateway
- Click on “Routes” next to “Details” and click “Edit routes”



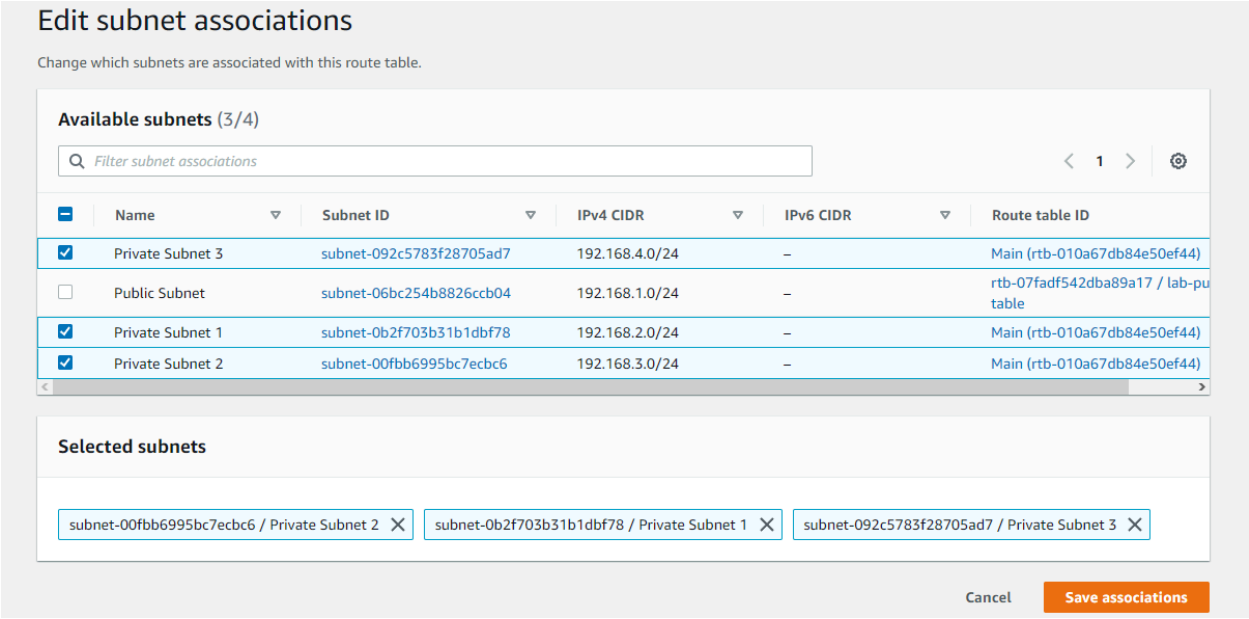
- Add a new route having a destination of anywhere and a target of your internet gateway and click “Save changes”



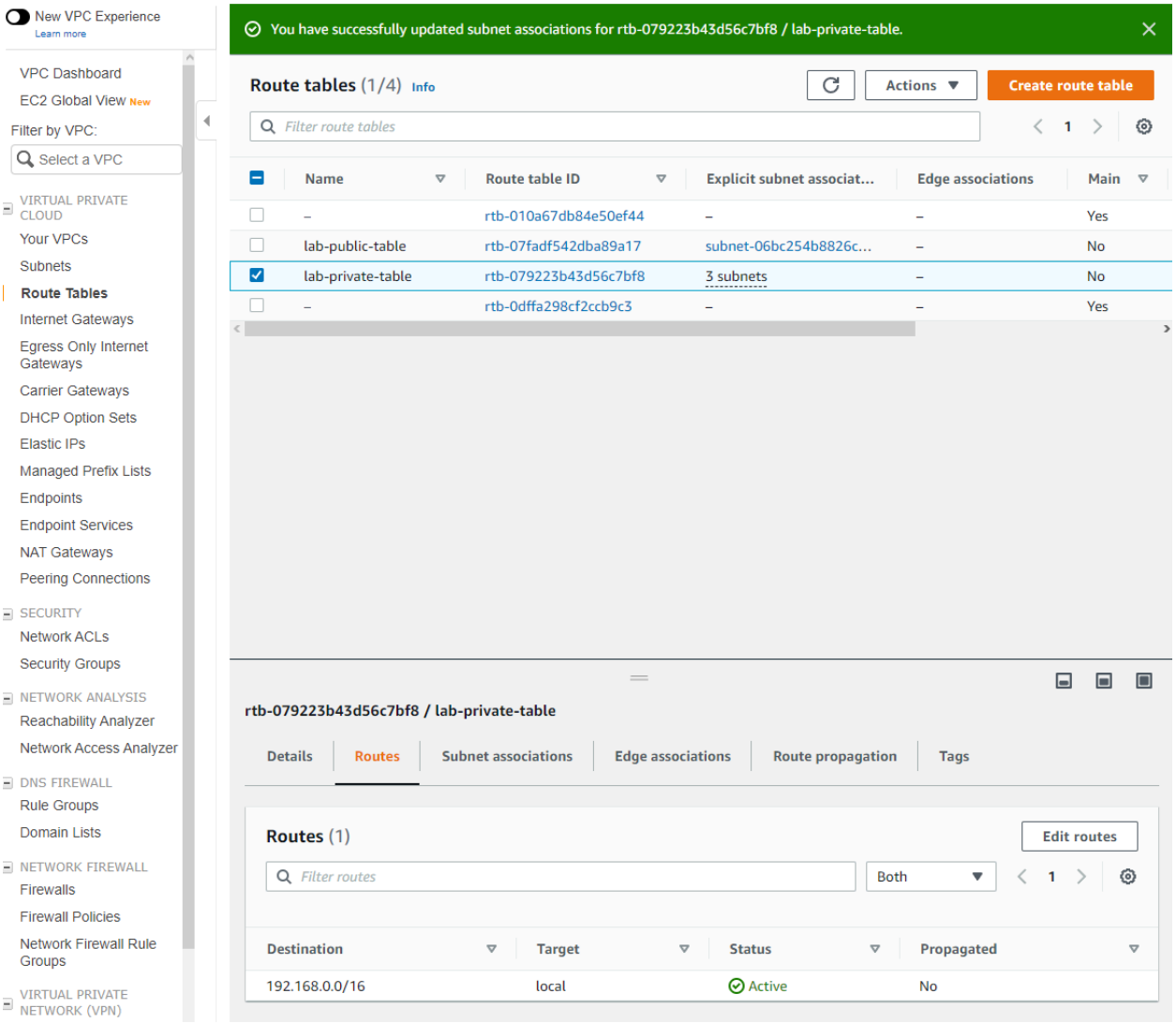
- Do the same thing for your private route table by clicking on it and going to its subnet associations and editing them



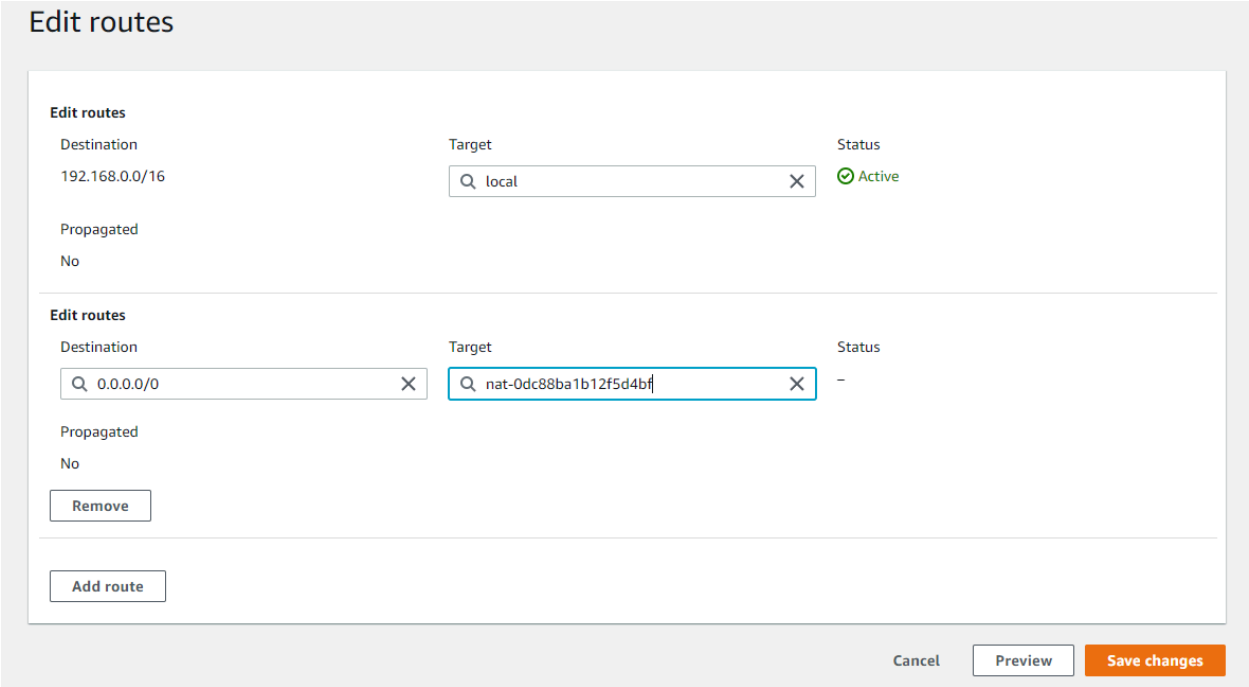
- Click on all three of your private subnets and save the associations

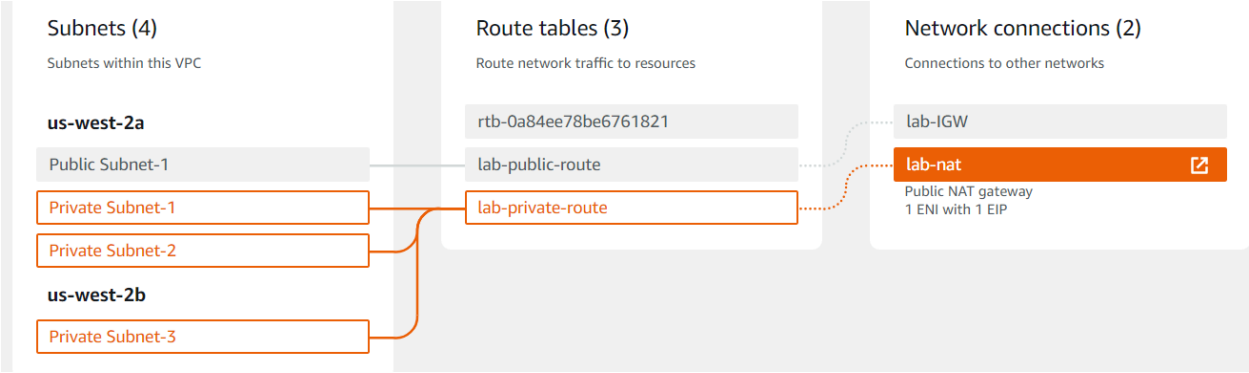


- Go to edit the routes of the private table

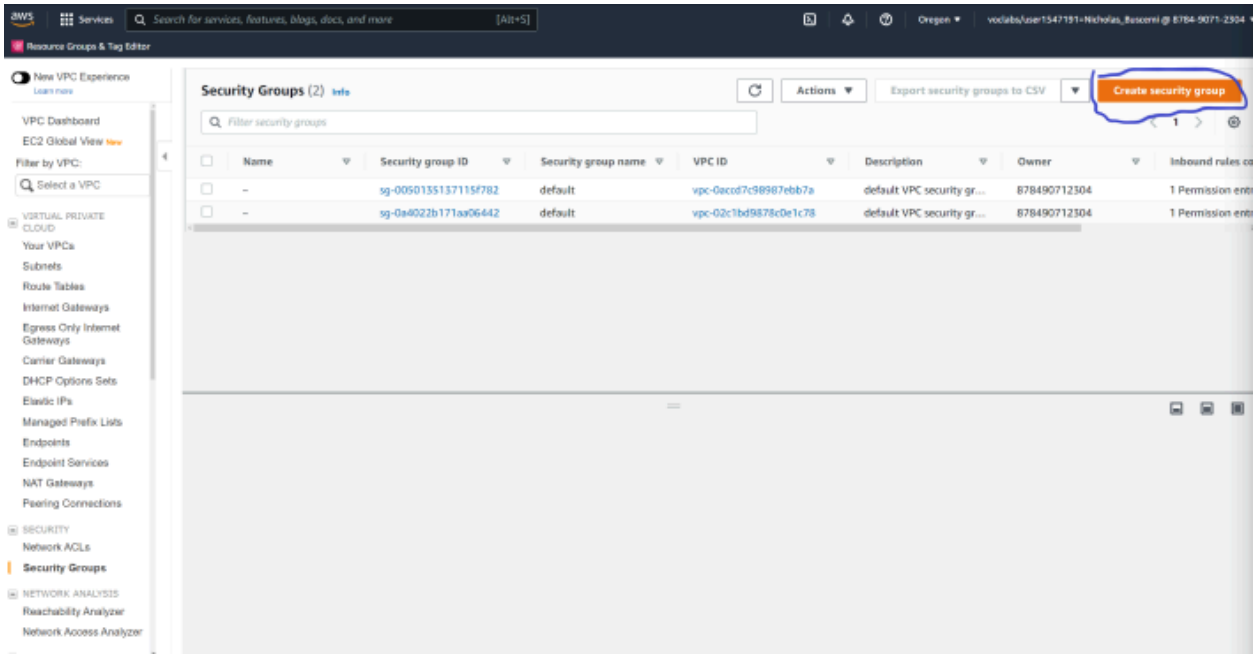


- Add a route to the private table that has a destination of anywhere and a target of your Nat gateway that you made earlier

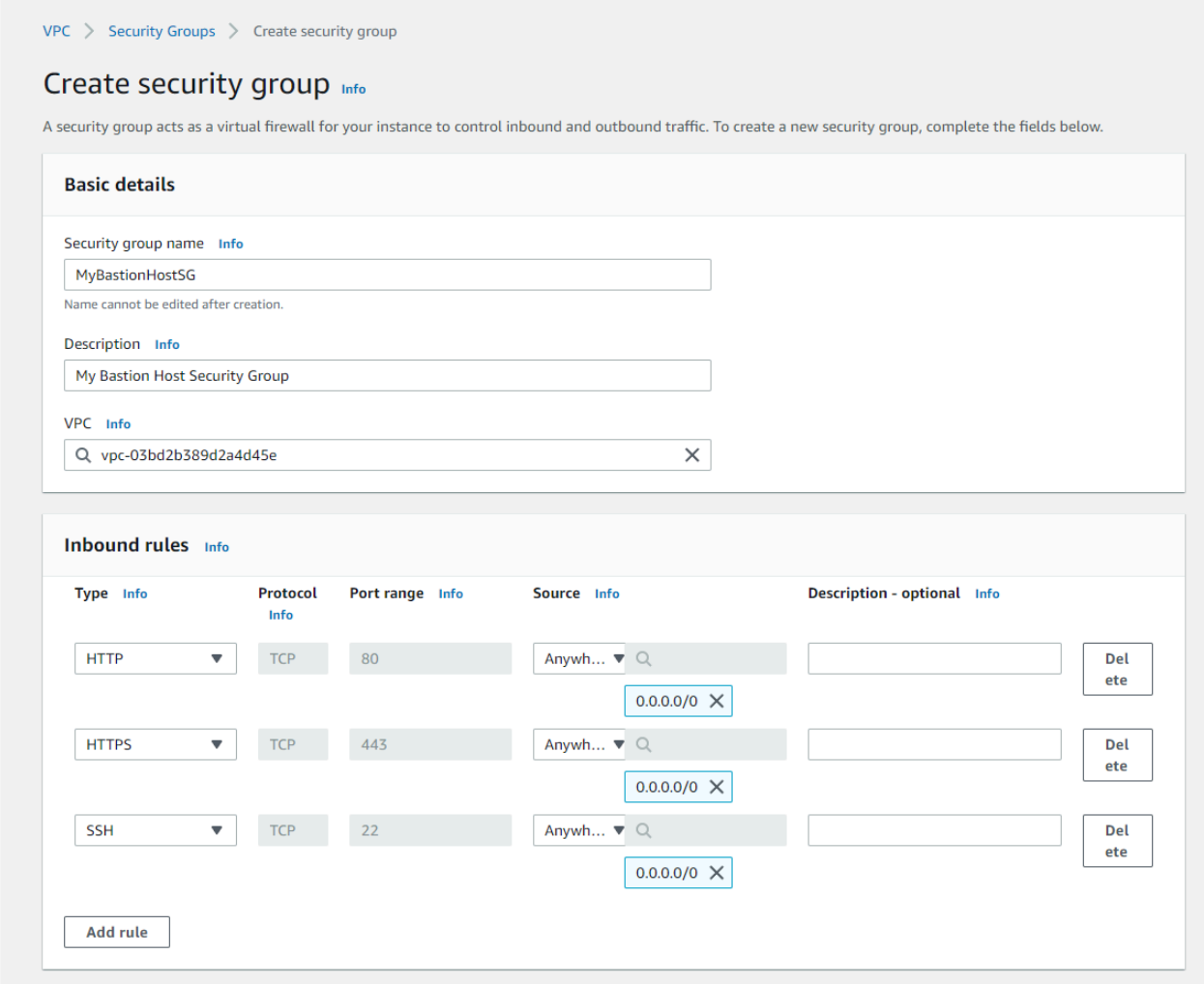




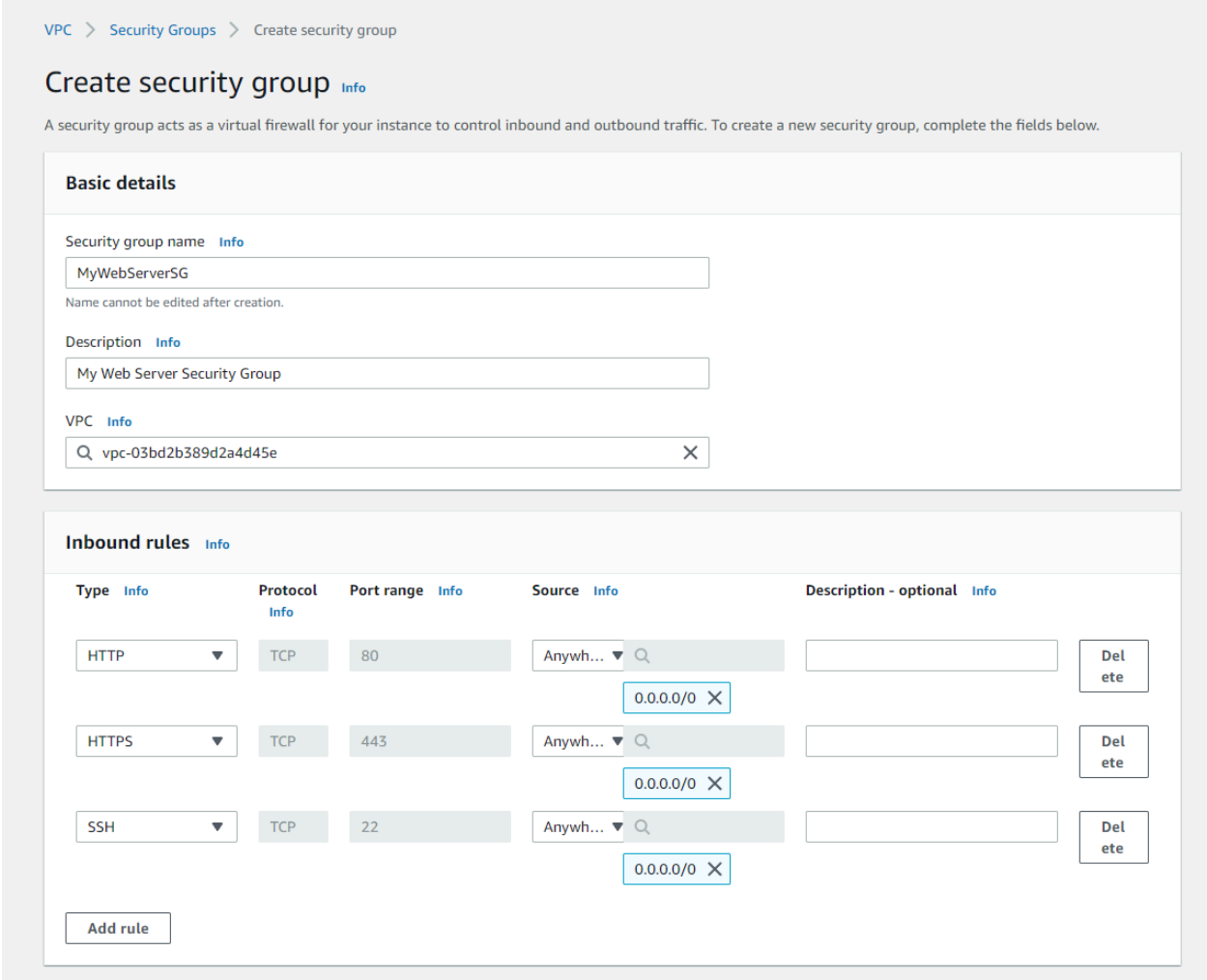
- Now to create our security groups (One for our bastion host, web server, app server, and our database) we will head to Security Groups on the left and click “Create security group”



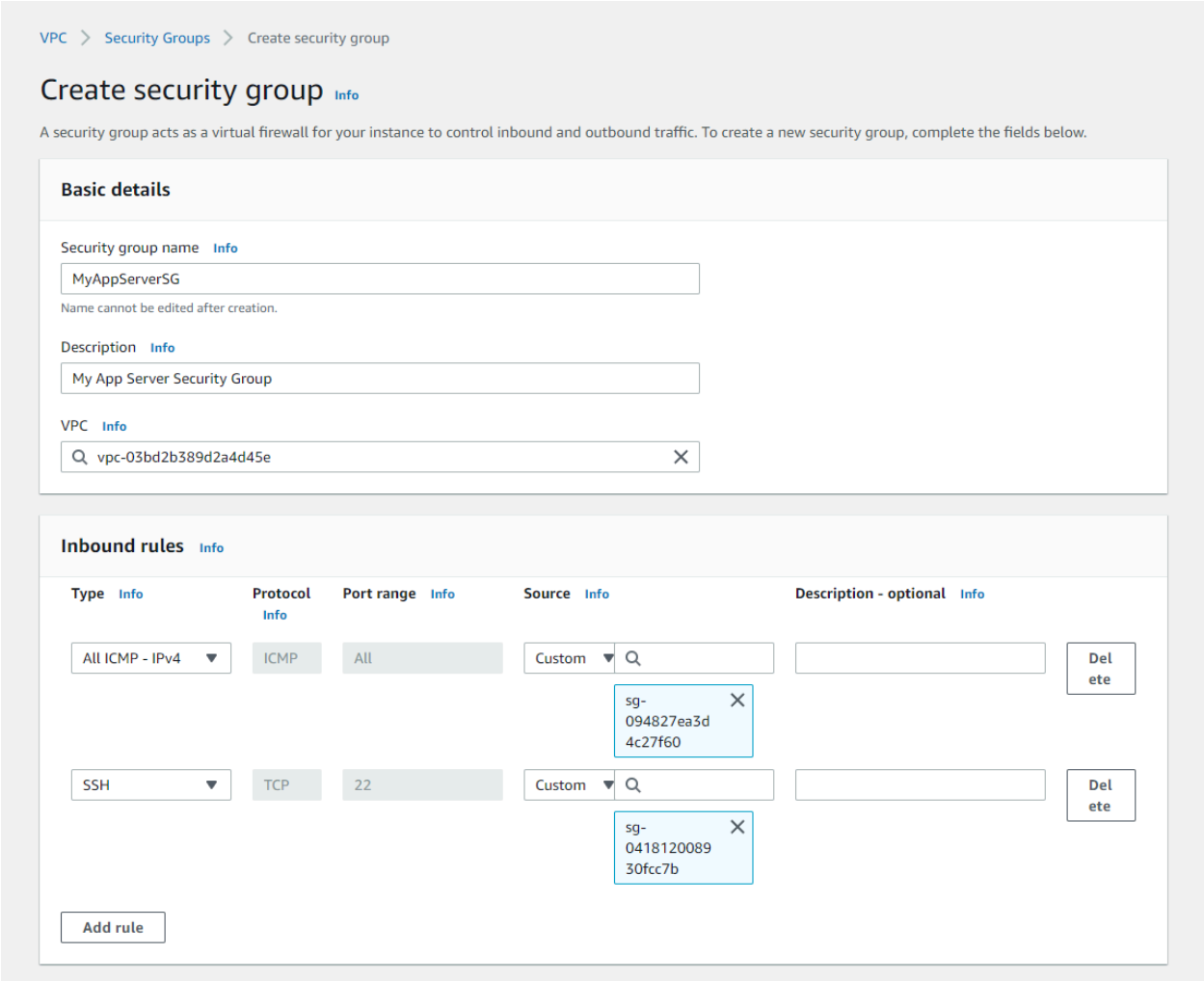
- Give it a name and description letting you know it is for a bastion host
- Assign your VPC to it
- Give it three inbound rules, one for SSH using your IP and one for HTTP using 0.0.0.0/0 as well as https using 0.0.0.0/0.



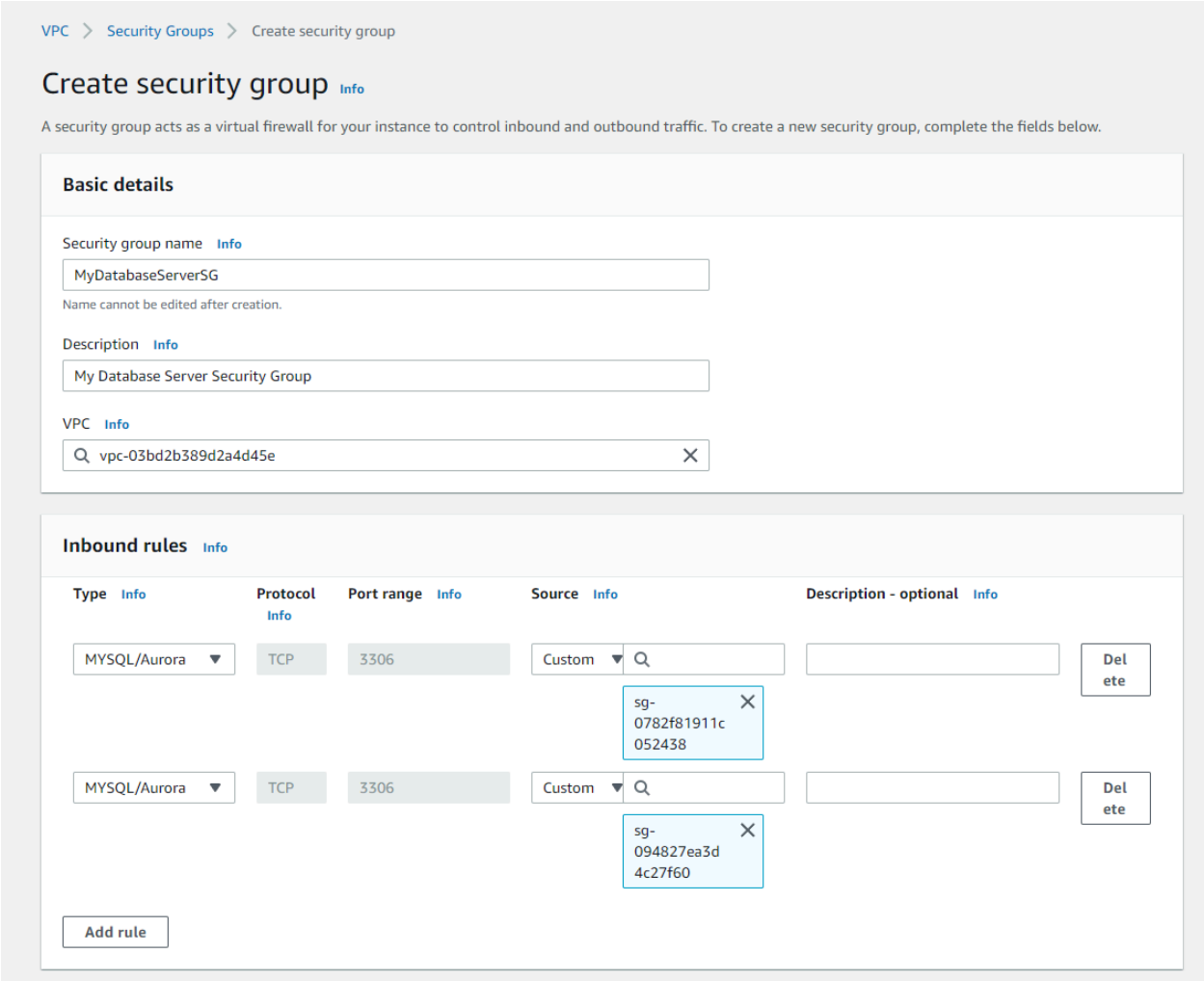
- Create another security group
- Give it a name and description letting you know it is for a Web server
- Assign your VPC to it
- Give it the same inbound rules as the Bastion Host security group



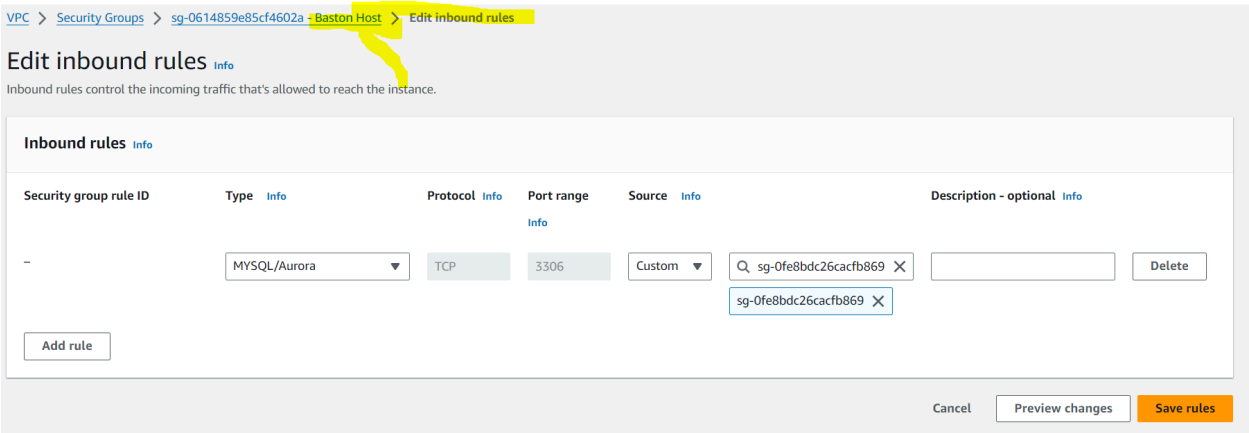
- Create another security group
- Give it a name and description letting you know it is for an **app server**
- Assign your VPC to it
- Give it an inbound rule for **All ICMP -IPv4** with **a source of your web server SG** and **another inbound rule for SSH** with a **source** of your **bastion host SG**



- Create one final security group
- Give it a name and description letting you know it is for a database server
- Assign your VPC to it
- Give it two inbound rules **both for MYSQL/Aurora** and give one of them a **source of your app server SG and the other one a source of your bastion host SG**

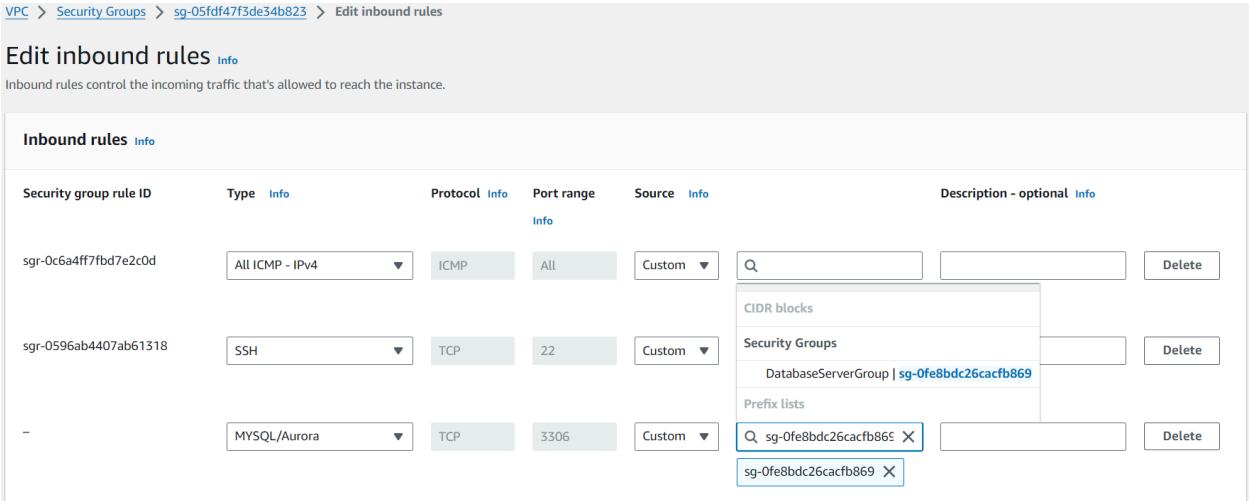


- Go back to your **bastion host inbound rules** and add one more for MYSQL/Aurora and a source of your **database SG**



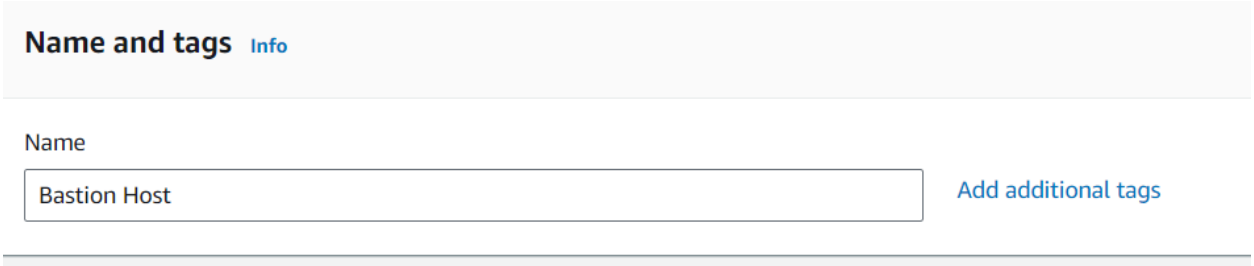
- Go back to your web server inbound rules and add one more for All ICMP - IPv4 and a source of your app server SG.

- Go back to your app server inbound rules and add one more for MYSQL/Aurora and a source of your database SG and then an HTTP and HTTPS rule both with a source of 0.0.0.0/0



Step 2: Create Servers

- Create Bastion Host



- Select Amazon Linux 2 AMI

Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Li

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type — Free tier eligible

ami-0895022f3dac85884 (64-bit (x86)) / ami-09ebdd80c5c138f65 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20240223.0 x86_64 HVM gp2

Architecture: 64-bit (x86)

AMI ID: ami-0895022f3dac85884 — Verified provider

- Select t2.micro

Instance type Info | Get advice

Instance type

t2.micro — Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Linux base pricing: 0.0116 USD per Hour

On-Demand SUSE base pricing: 0.0116 USD per Hour

On-Demand Windows base pricing: 0.0162 USD per Hour

On-Demand RHEL base pricing: 0.0716 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

- Select the key pair

Key pair (login) Info

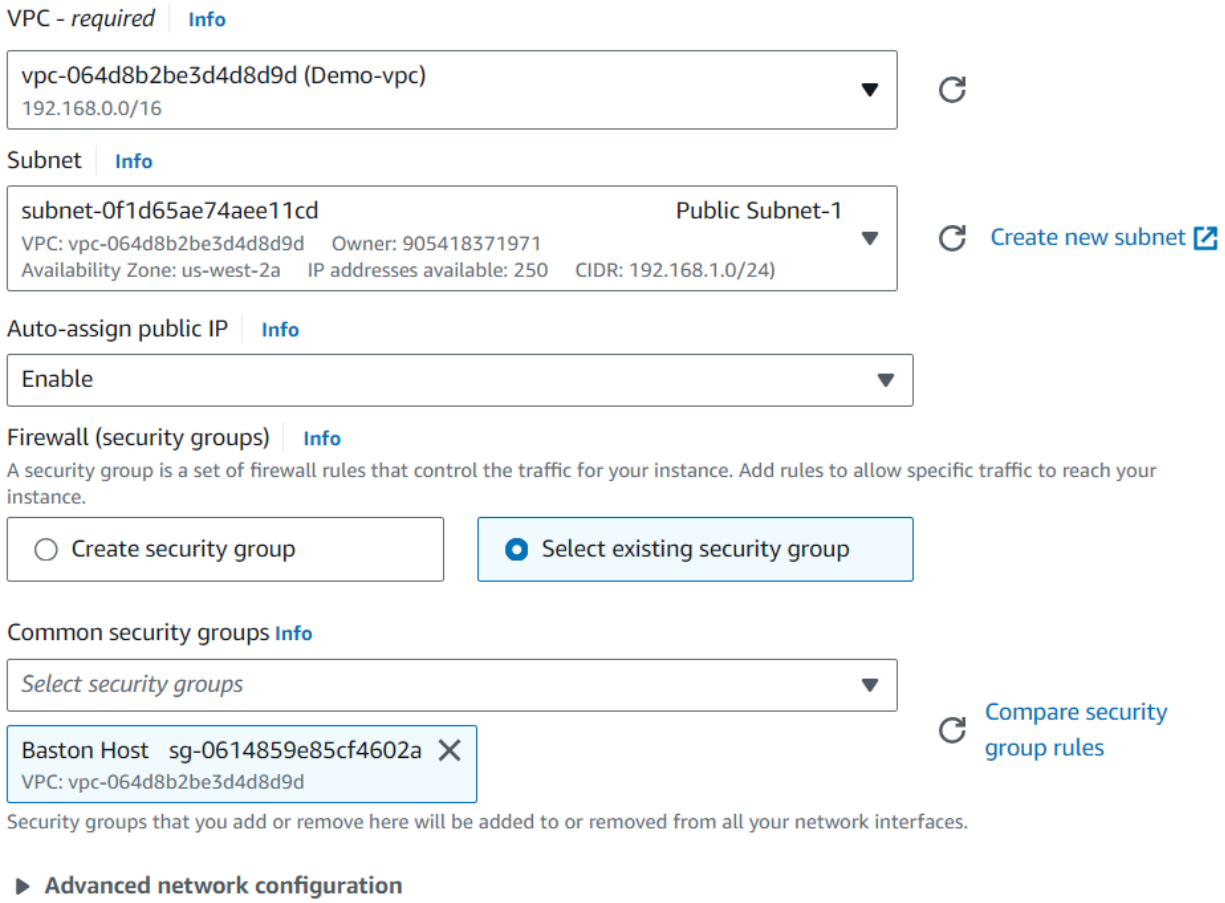
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

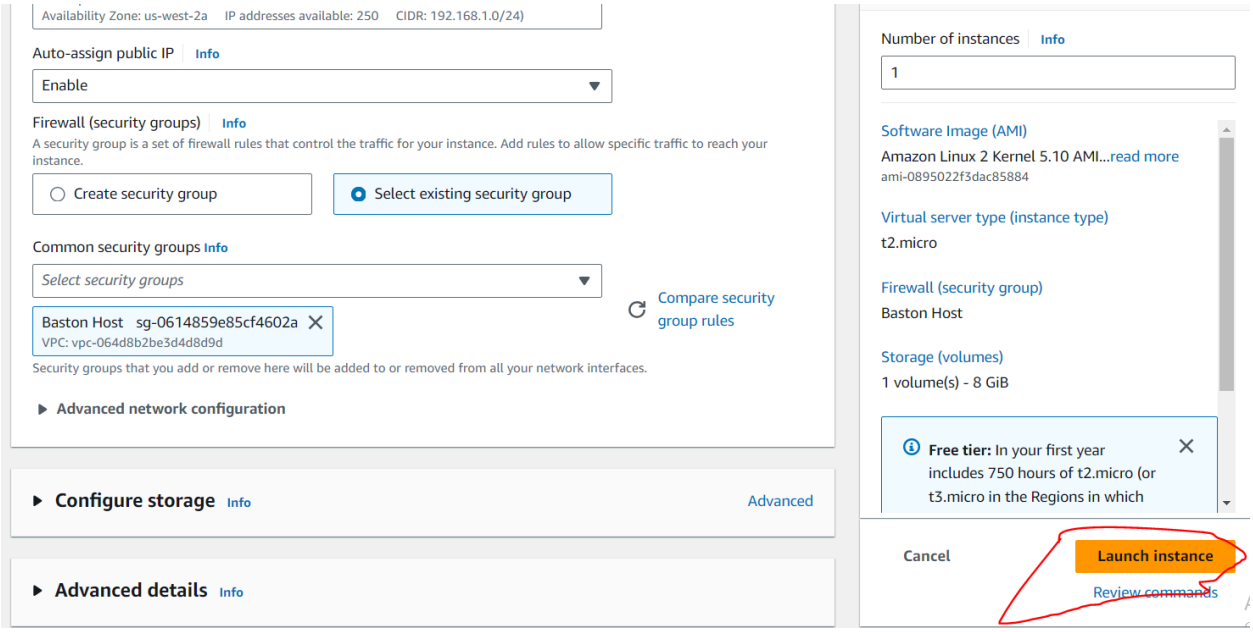
vockey

Create new key pair

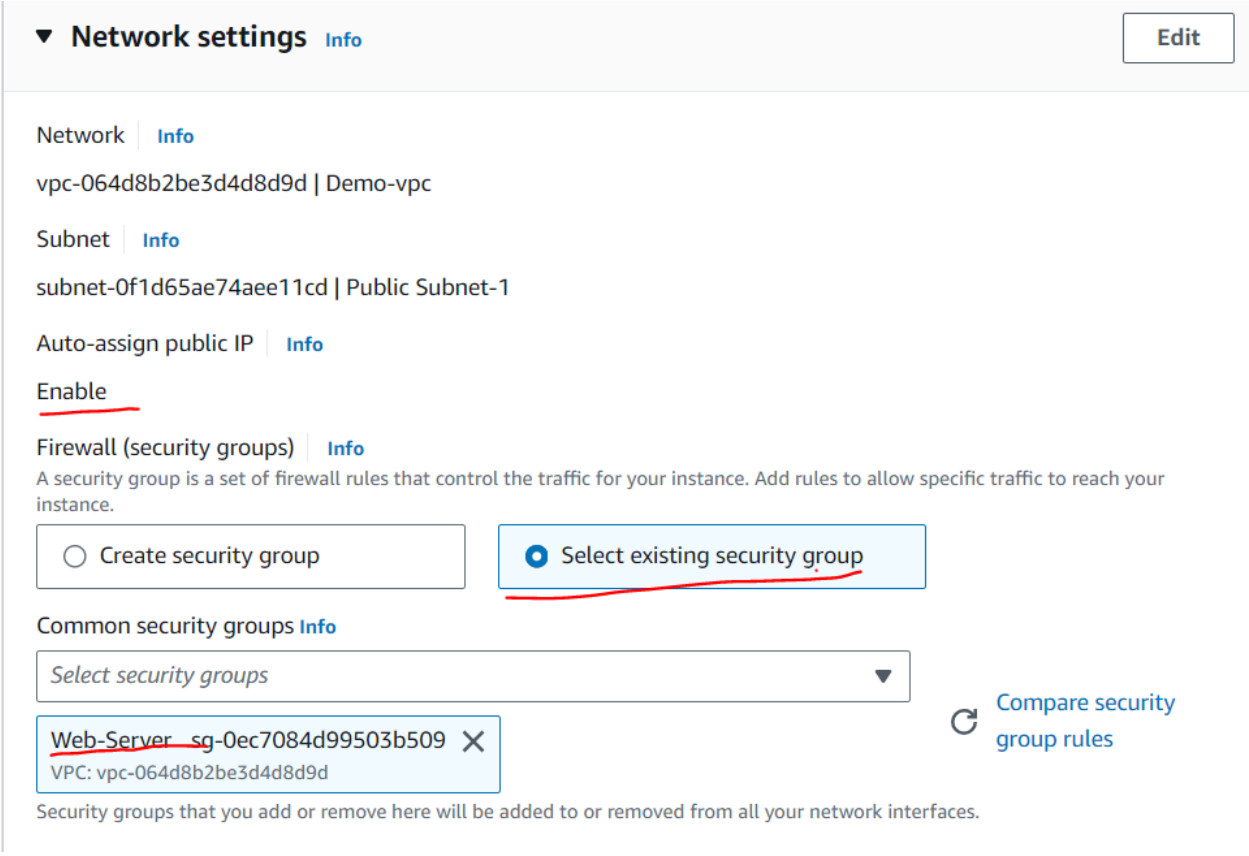
- Put in your VPC and Public Subnet, enable auto assign public IP and Select an existing group and select your Bastion Host SG.



- Storage leaves default.



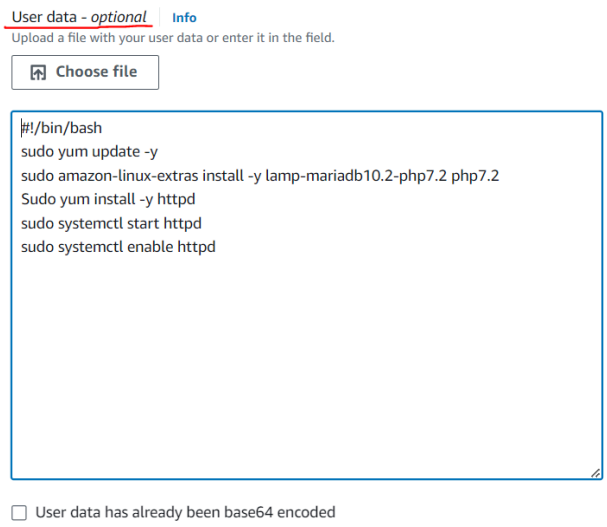
- Follow the same steps to create the Web Server
- Follow along like previously and change your network, subnet, and enable auto assign public ip

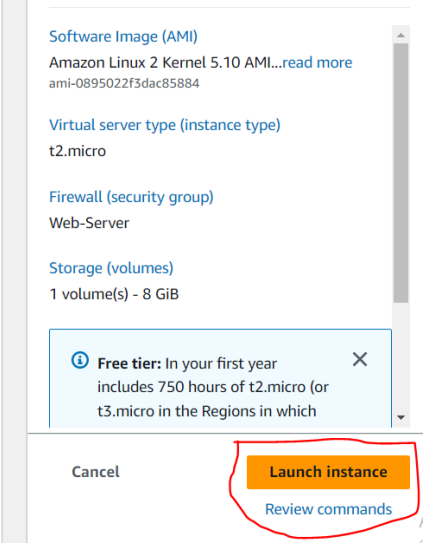


- Storage leave default

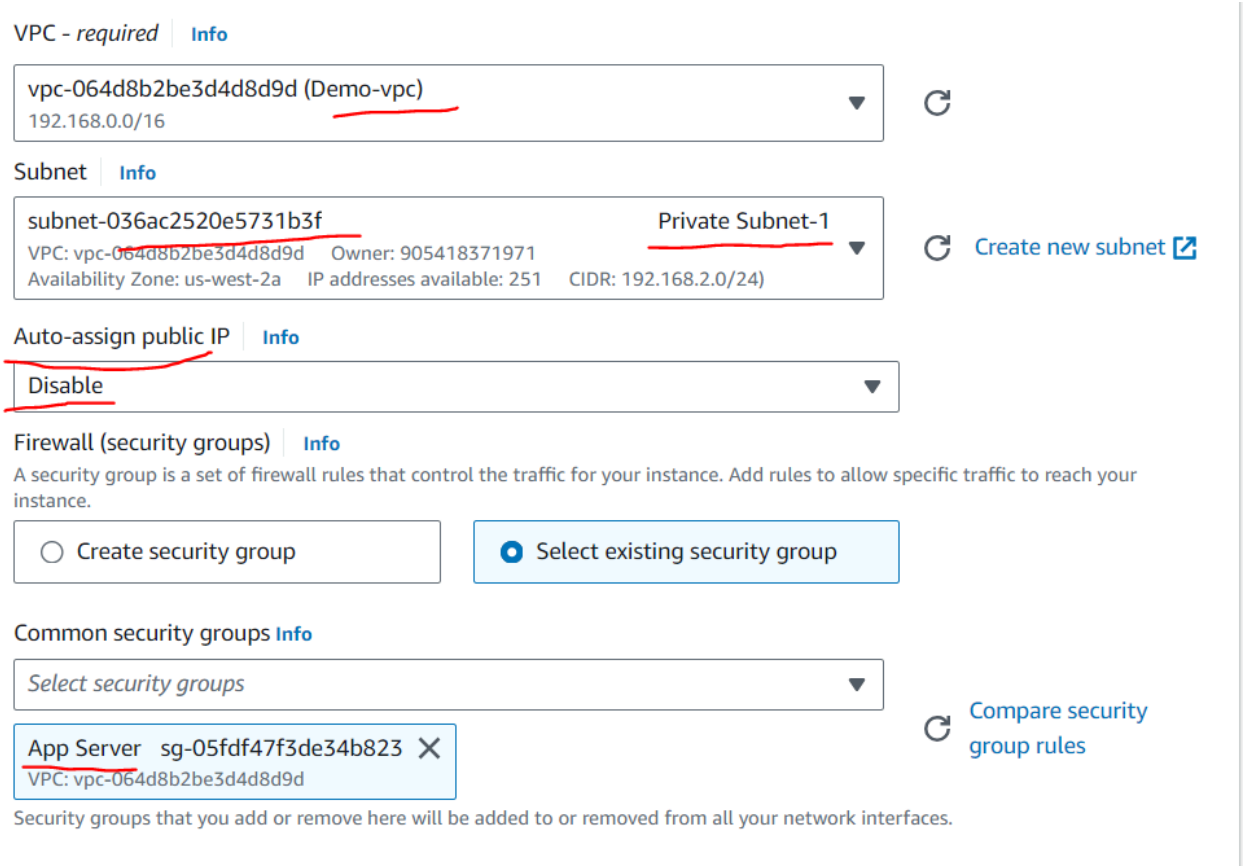
- Then go to user data and type this into it to set up the web server

```
#!/bin/bash
sudo yum update -y
sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
sudo yum install -y httpd
sudo systemctl start httpd
sudo systemctl enable httpd
```





- Creating an App server
- Put in your VPC and then choose **Private Subnet 1** for the subnet and leave auto assign public ip disabled



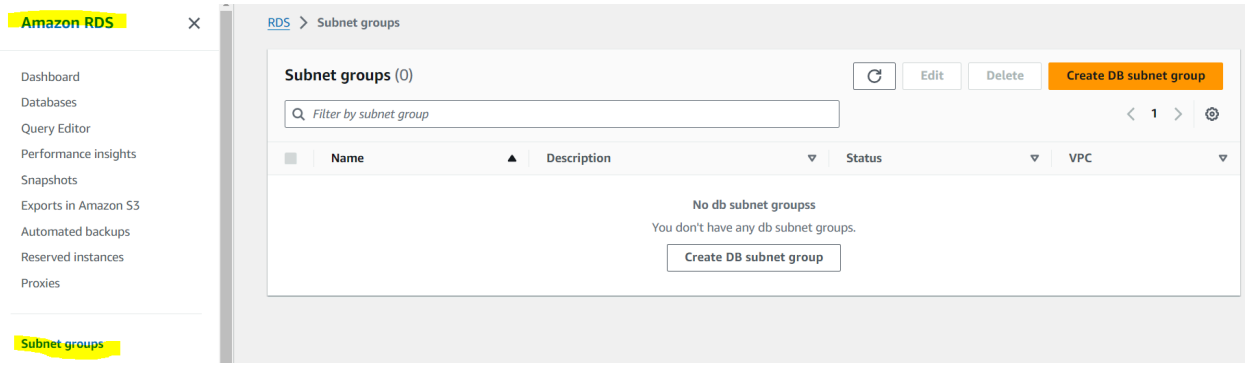
- Then go into Advanced Detail section add the below script in user data column

```
#!/bin/bash
sudo yum update -y
sudo yum install -y mariadb-server
sudo service mariadb start
```

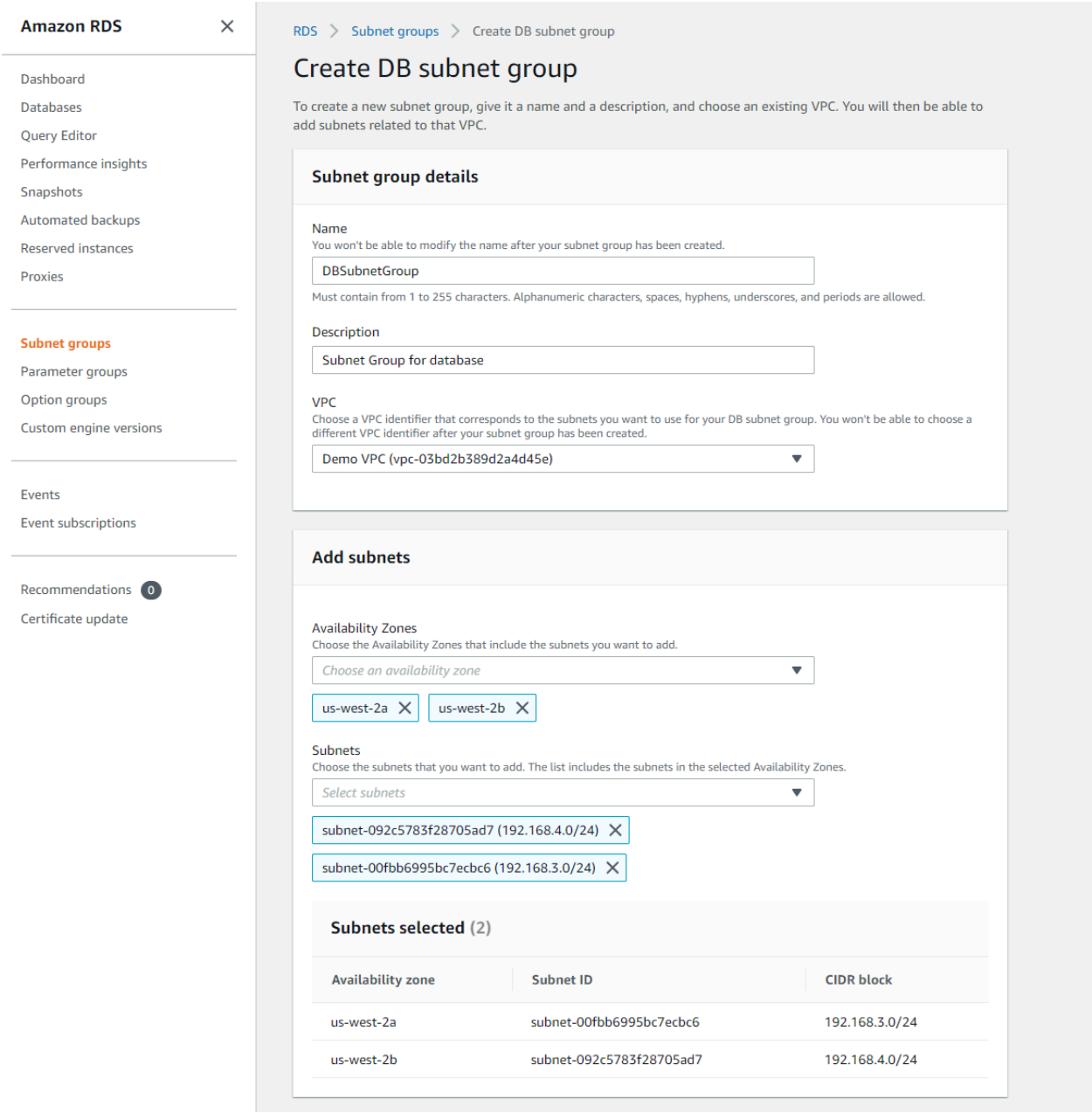
Note:
To check version of mariadb use **mysql -v**

Step 3: Create a Database

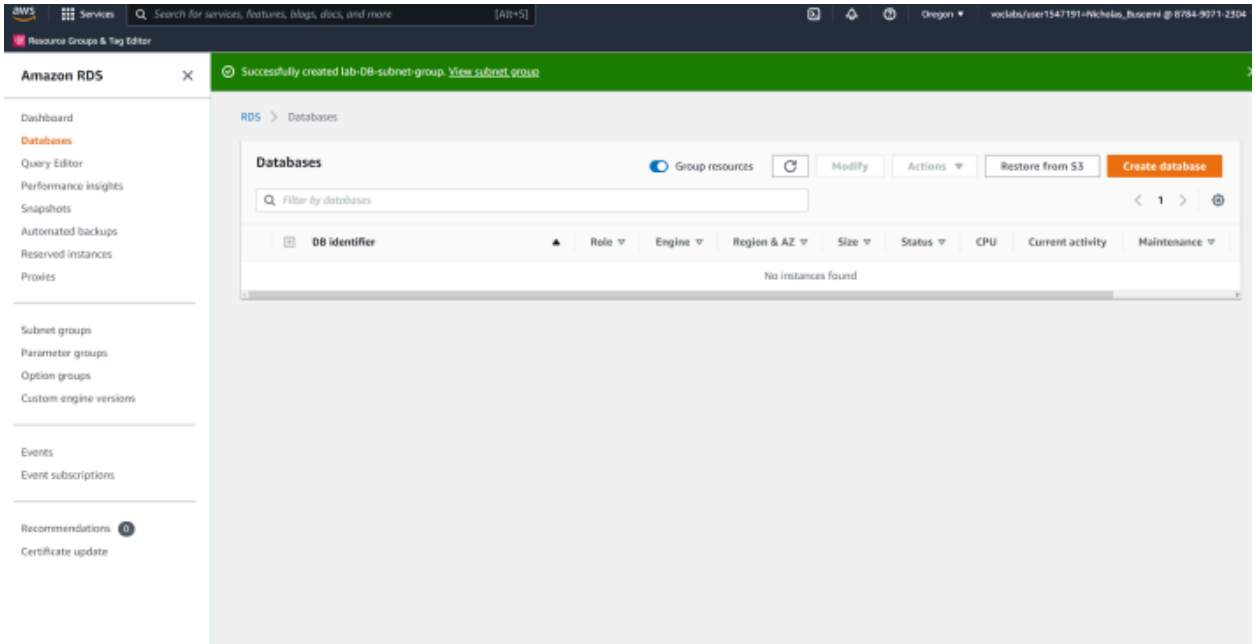
- From the management console search RDS.
- In left pane, find **DB subnet group.** Create a DB Subnet group.



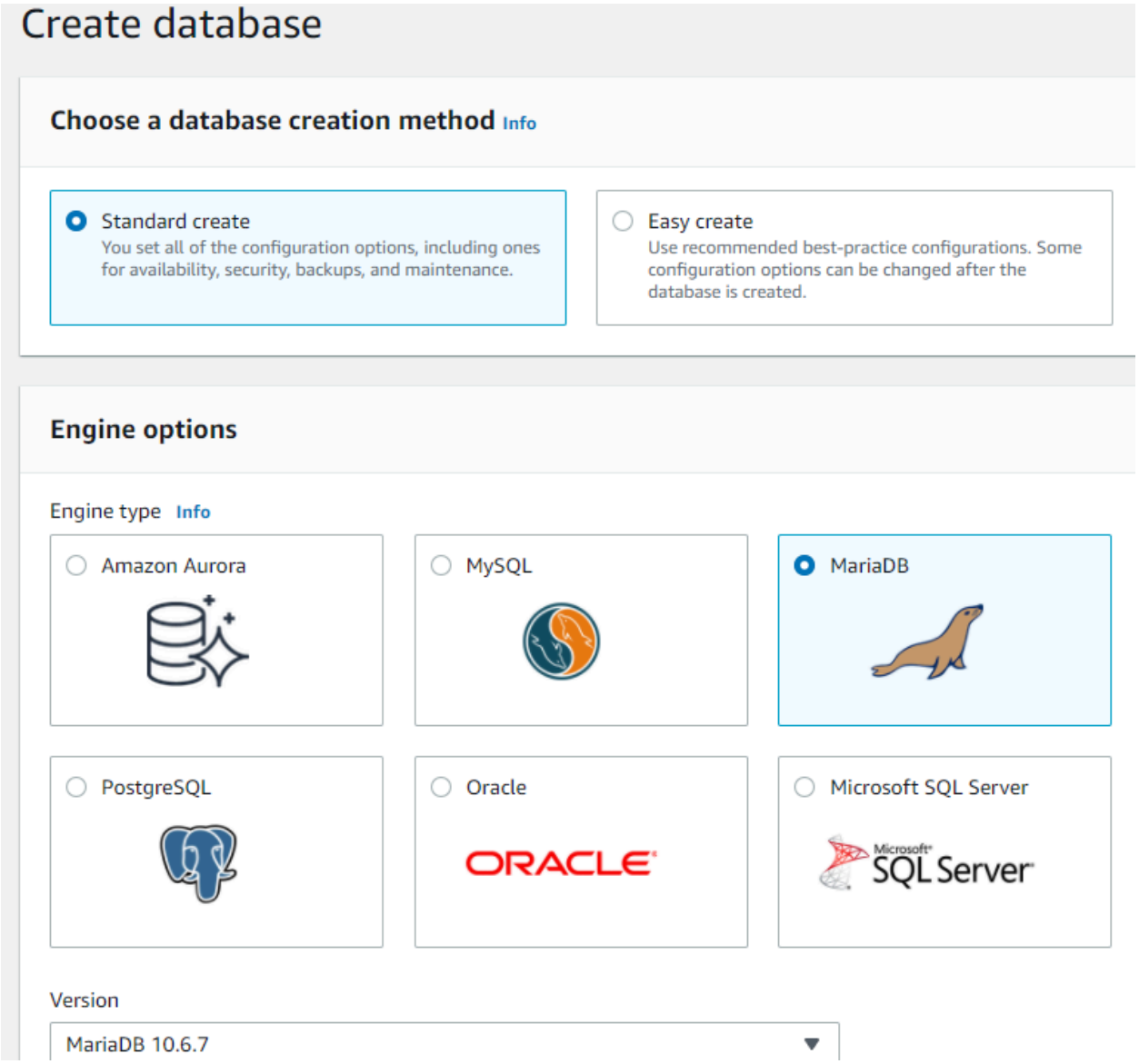
- Give it a name and description letting you know what it is and then assign your VPC to it
- Put in the availability zones you used for your subnets
- Select subnets 3 and 4
- Click create



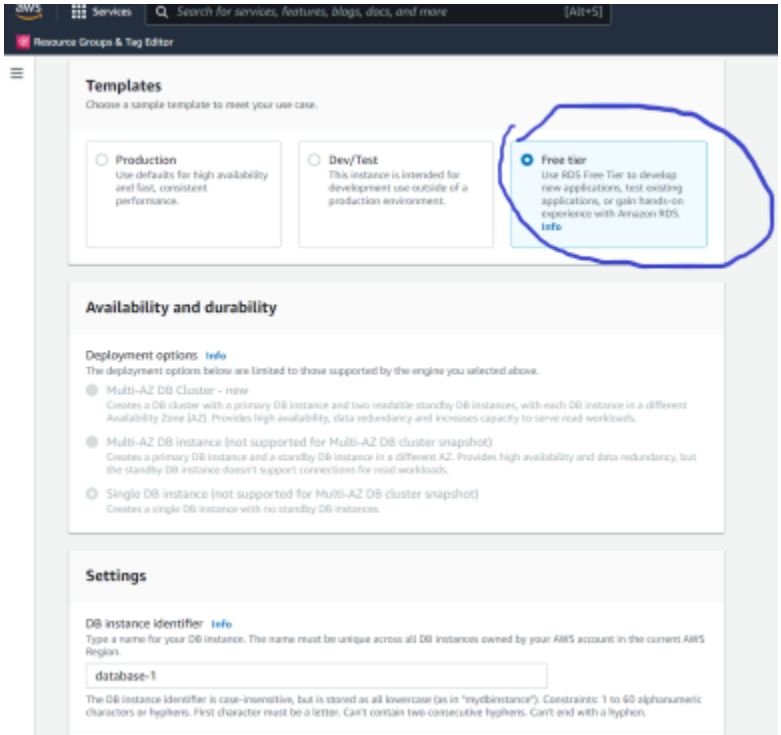
- Go to Databases on the left hand side and click on “Create Database”


Services
Search for services, features, blogs, docs, and more
[Alt+S]
Oregon
Resource Groups & Tag Editor
Amazon RDS
Successfully created lab-DB-subnet-group. View subnet group
Dashboard
Databases
Query Editor
Performance insights
Snapshots
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Custom engine versions
Events
Event subscriptions
Recommendations 0
Certificate update
RDS > Databases
Databases
Group resources
Modify
Actions
Restore from S3
Create database
Filter by databases
DB identifier
Role
Engine
Region & AZ
Size
Status
CPU
Current activity
Maintenance
No instances found

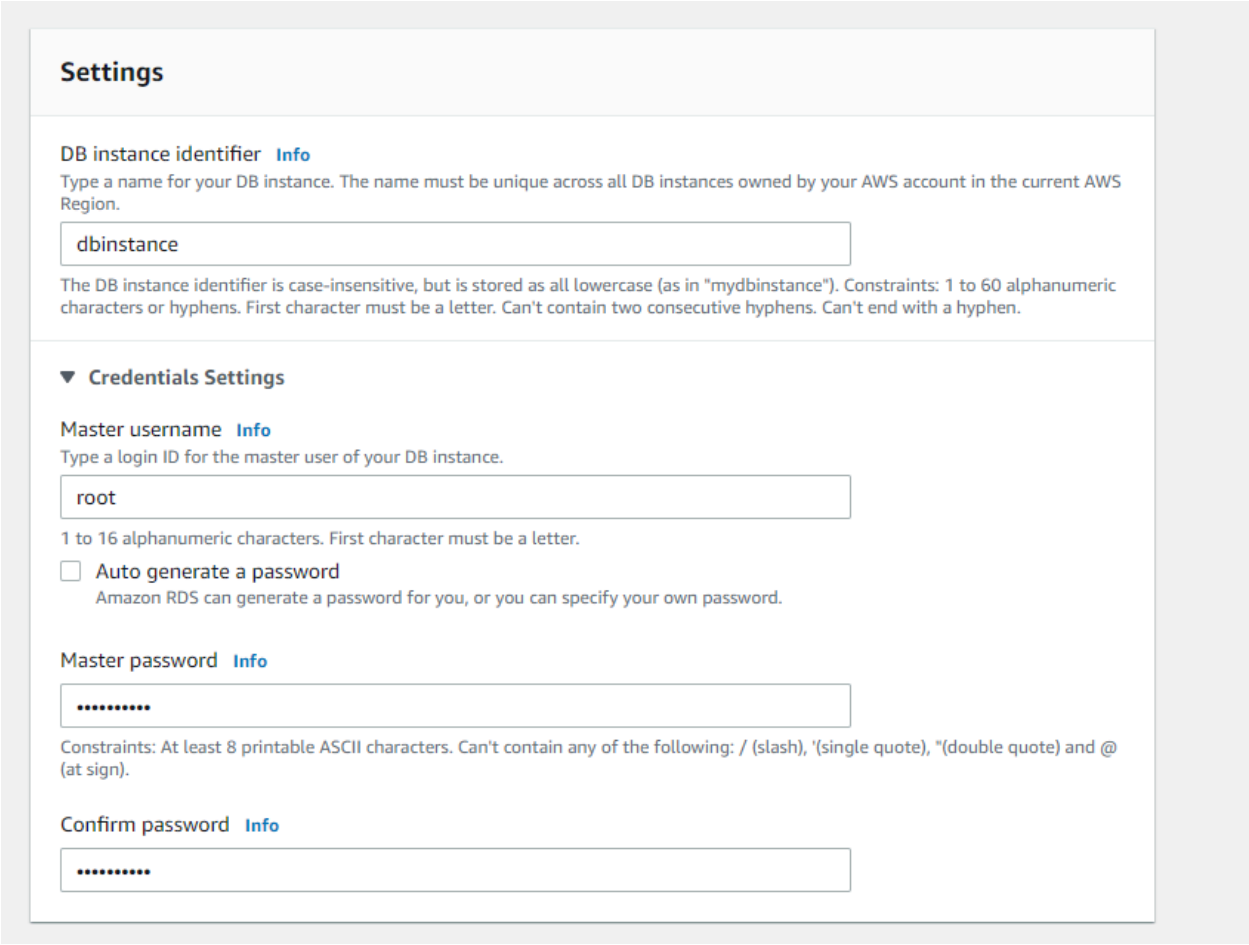
- Click on Standard create and MariaDB for the engine type



- Make sure you click on Free tier here



- Give it an identifier you can easily identify it with
- Give it a master username or leave it as default admin. For the purpose of these instructions I will be using **root**
- Give it a password that you write down somewhere else to make sure you have the correct one. For the purpose of these instructions I will be using **Re:Start!9**



- Everything between this and the last step is left default
- Assign your vpc
- Make sure your subnet group is listed under the subnet group section
- Public access is no
- Choose existing VPC security groups
- Remove the default security group and add your database security group
- Select your first availability zone as well

Connectivity

Virtual private cloud (VPC) Info

VPC that defines the virtual networking environment for this DB instance.

Demo VPC (vpc-03bd2b389d2a4d45e)

Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change its VPC.

Subnet group Info

DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

dbsubnetgroup

Public access Info

Yes

Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.

No

RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

VPC security group

Choose a VPC security group to allow access to your database. Ensure that the security group rules allow the appropriate incoming traffic.

Choose existing

Choose existing VPC security groups

Create new

Create new VPC security group

Existing VPC security groups

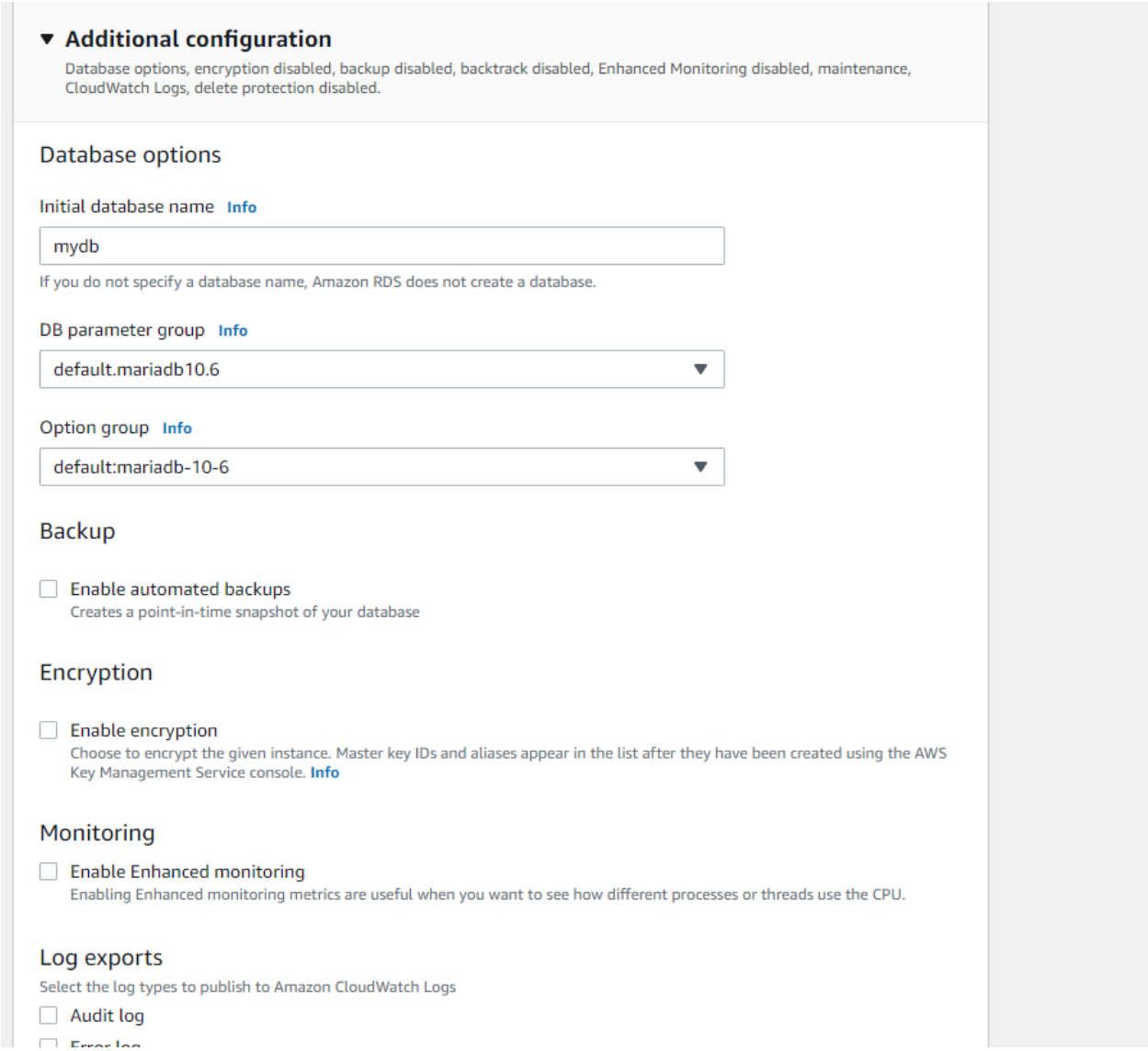
Choose VPC security groups

MyDatabaseServerSG

Availability Zone Info

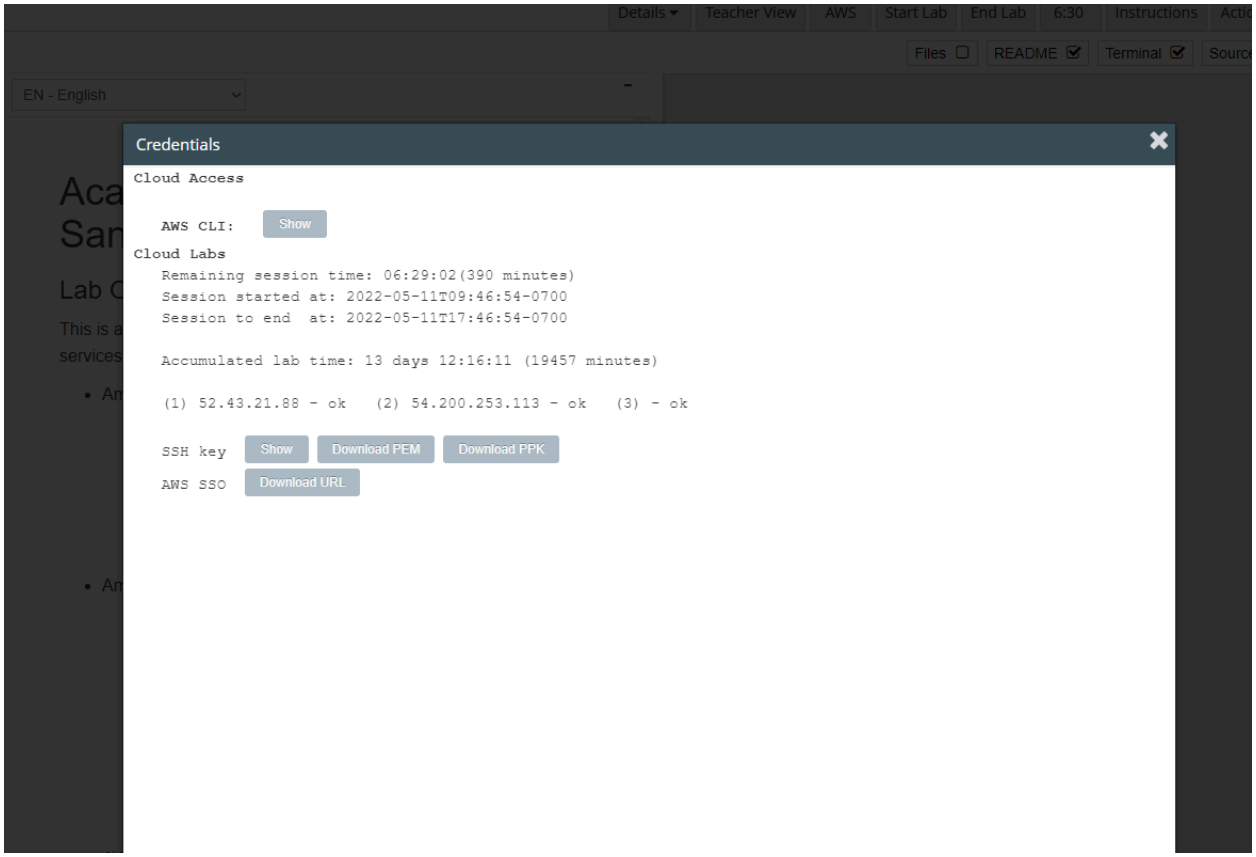
us-west-2a

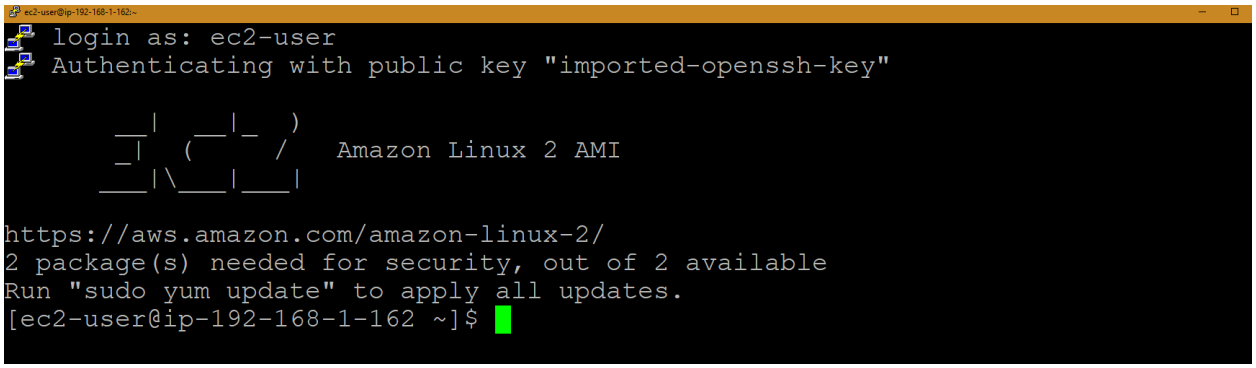
- Scroll down to Additional configuration on the bottom and give it an initial database name and save it in the same spot as your password since it will be used later
- Disable automated backups and encryption since they are not needed (These are normally best practice to leave enabled but the database will spin up faster with those checked off as they are not needed).
- Scroll down all the way to the bottom and create your database



Step 4: Test connections

- SSH into your Bastion Host after downloading both the pem and ppk files from the lab environment





- This is for windows only as I only have a windows machine to work on, sorry mac and linux users
- Go into your powershell and type this command out
    - Pscp -scp -P 22 -i '.\Downloads\labsuser.ppk' -| user ec2-user '.\Downloads\labsuser.pem' ec2-user@bastion-host-public-ip:/home/ec2-user
    - Replace bastion-host-public-ip with the public ip address of your bastion host
- Hit enter and it should upload those keys to your bastion host for use on other servers

```
PS C:\Users\nickb> Pscp -scp -P 22 -i '.\Downloads\labsuser.ppk' -l user ec2-us
.200.253.113:/home/ec2-user
pscp: ec2-user: No such file or directory

labsuser.pem              | 1 kB |   1.6 kB/s | ETA: 00:00:00 | 100%
PS C:\Users\nickb>
```

- Test on your ssh to see if the file is uploaded by using ls. Should return something like this

```
[ec2-user@ip-192-168-1-162 ~]$ ls
labsuser.pem
[ec2-user@ip-192-168-1-162 ~]$
```

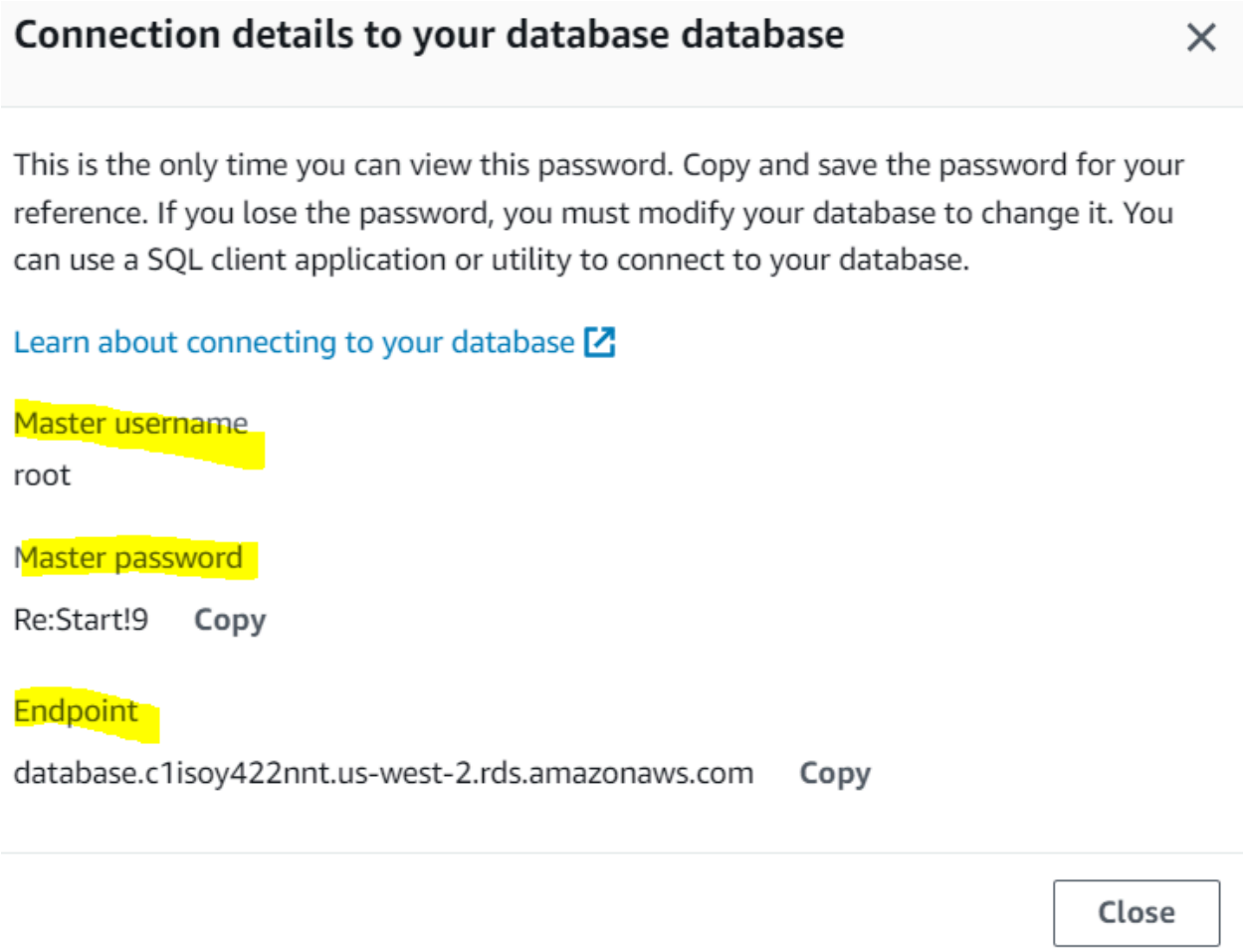
- Change file permissions for the file we just downloaded to our bastion host by typing
    - chmod 400 labsuser.pem
- Then ssh into our app server by typing
    - ssh -i my-key-pair.pem ec2-user@app-server-private-ip
    - Replace my-key-pair with the name of your key
    - Replace app-server-private-ip with your app server's private ip address
- Type yes when it prompts you to
- Use ls to see that you are now ssh into a different server since there is no more key

```
[ec2-user@ip-192-168-1-162 ~]$ chmod 400 labsuser.pem
[ec2-user@ip-192-168-1-162 ~]$ ssh -i labsuser.pem ec2-user@192.168
.2.172
The authenticity of host '192.168.2.172 (192.168.2.172)' can't be e
stablished.
ECDSA key fingerprint is SHA256:RURnbNWL6+XNSA3+S9k0FtM1Fy0aAHcv4z7
qLtKlm7A.
ECDSA key fingerprint is MD5:ce:f3:58:79:65:eb:ae:de:6a:3c:51:5c:89
:4b:69:88.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.2.172' (ECDSA) to the list of k
nown hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 2 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-192-168-2-172 ~]$ ls
[ec2-user@ip-192-168-2-172 ~]$
```

- Use ping and the private ip address of your web server to ping the web server and see it connect

```
[ec2-user@ip-192-168-2-172 ~]$ ping 192.168.1.252
PING 192.168.1.252 (192.168.1.252) 56(84) bytes of data.
64 bytes from 192.168.1.252: icmp_seq=1 ttl=255 time=0.486 ms
64 bytes from 192.168.1.252: icmp_seq=2 ttl=255 time=0.441 ms
64 bytes from 192.168.1.252: icmp_seq=3 ttl=255 time=0.450 ms
64 bytes from 192.168.1.252: icmp_seq=4 ttl=255 time=0.483 ms
^C
--- 192.168.1.252 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3052ms
rtt min/avg/max/mdev = 0.441/0.465/0.486/0.019 ms
[ec2-user@ip-192-168-2-172 ~]$
```

- Test out connecting to the database by typing out mysql –user=root -password='Re:Start!9' –host=database-server-endpoint
- Replace database-server-endpoint with the database server endpoint
- Type show databases; to see your database from the app server



```
[ec2-user@ip-192-168-2-31 ~]$ mysql --user=root --password='Re:Start!9' --host=d
atabase.clisoy422nnt.us-west-2.rds.amazonaws.com
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 83
Server version: 10.11.6-MariaDB managed by https://aws.amazon.com/rds/

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| innodb             |
| mydb               |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.01 sec)

MariaDB [(none)]>
```

- This concludes the lab