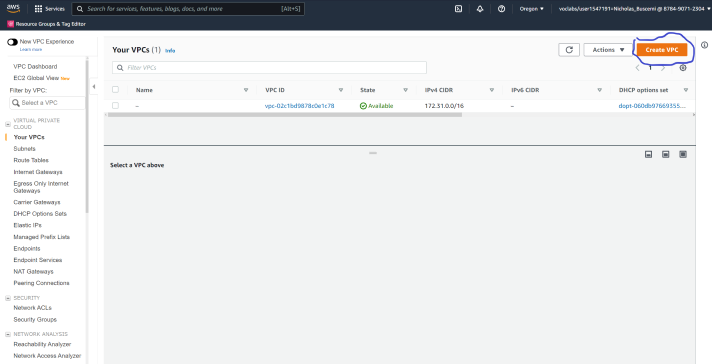
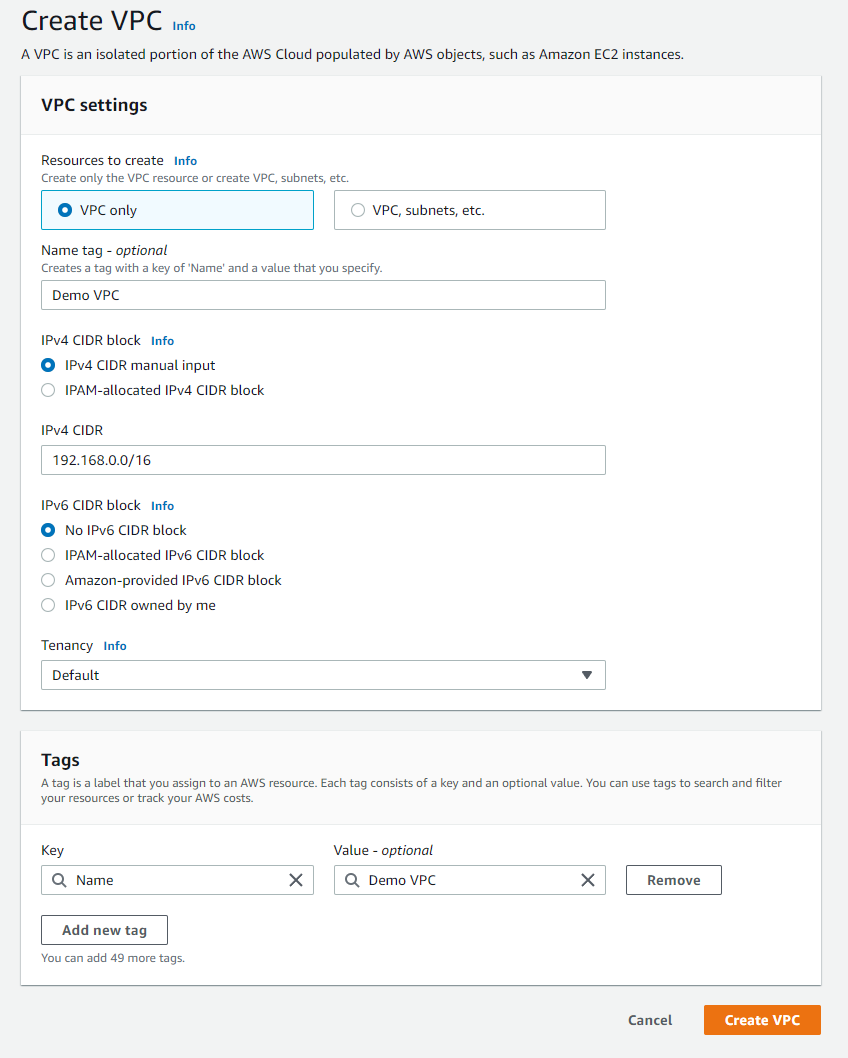
Step 1: Create a VPC and Subnets as well as routing and security groups

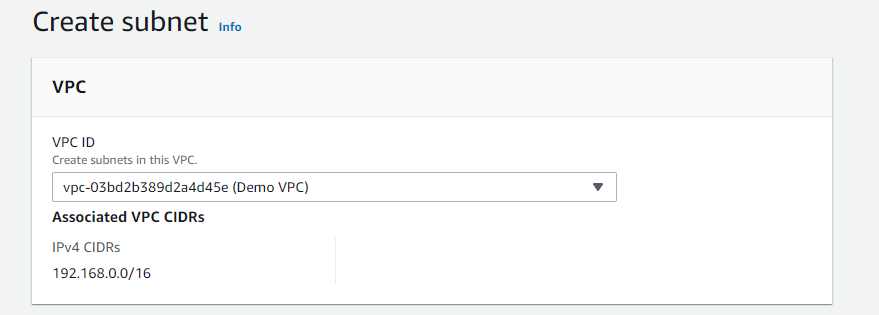
* Go to “Your VPCs” from the VPC service on the AWS management console and click on the orange “Create VPC” button



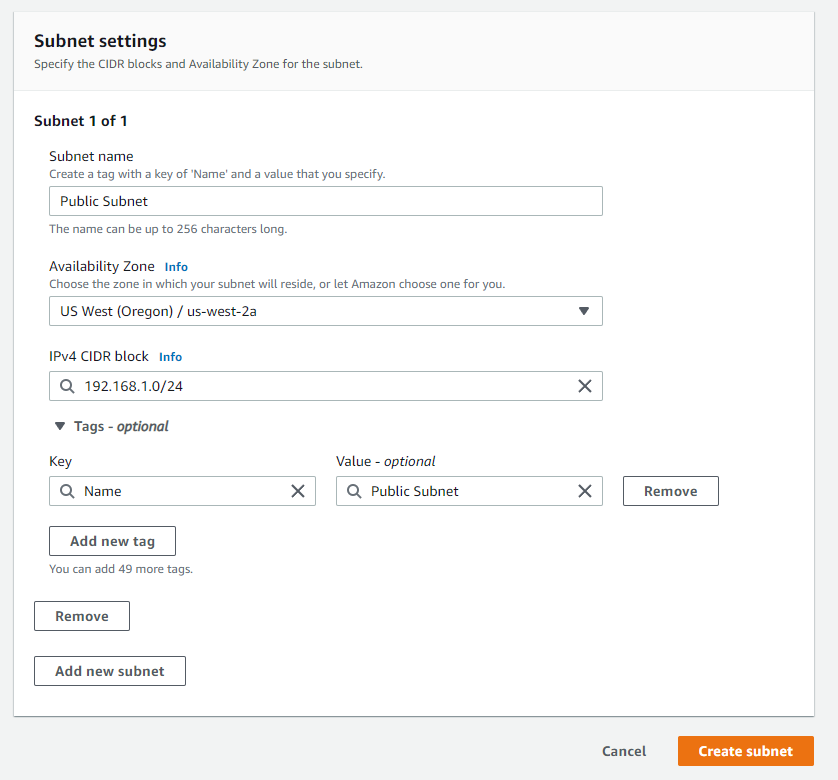
* Only create a vpc here and give it a name. You are free to make your own name or follow along with the one put here
* GIve it a 192.168.0.0/16 CIDR block and leave everything else as default. Click create.



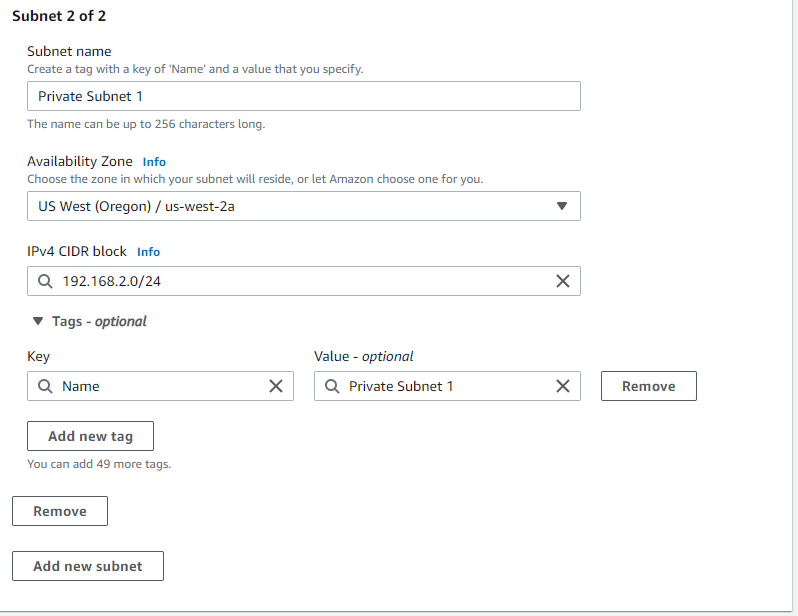
* To create your subnets go to Subnets on the left hand side of the VPC service and click on it
* Add your VPC ID to where it asks



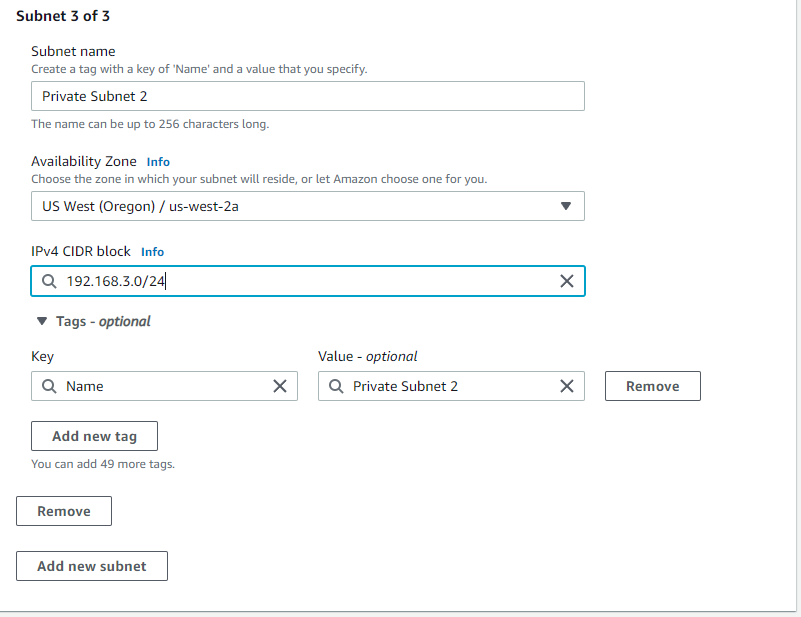
* Assign it a name letting you know what it is your first public subnet
* Put it in any availability zone and give it a CIDR of 192.168.1.0/24



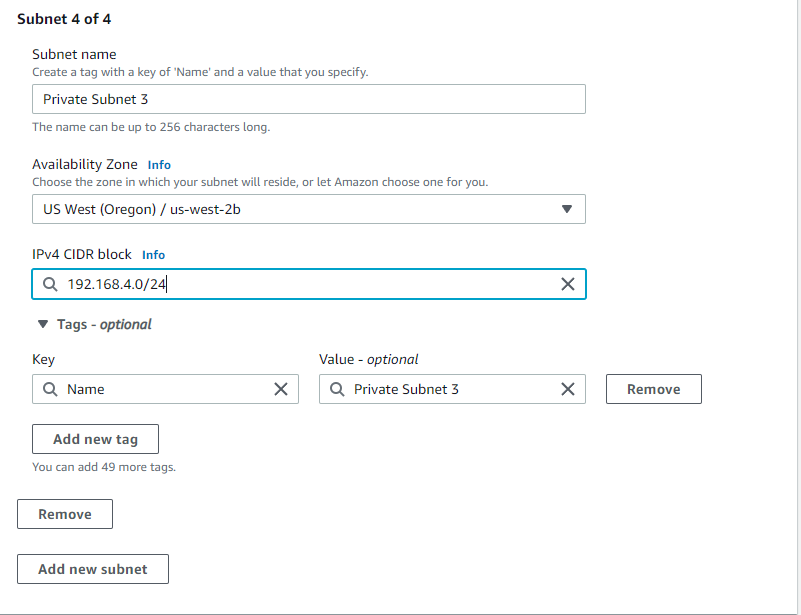
* Add a second subnet and name it Private Subnet 1 or something to let you know it is your first private subnet
* Put it in the same availability zone as the first subnet you made and give it a CIDR of 192.168.2.0/24



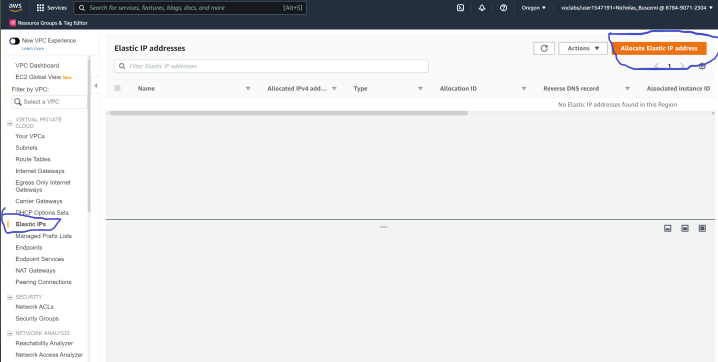
* Add a third subnet and assign a name letting you know it is the second private subnet you will be making
* Put it in the same availability zone as your first public subnet and give it a CIDR of 192.168.3.0/24



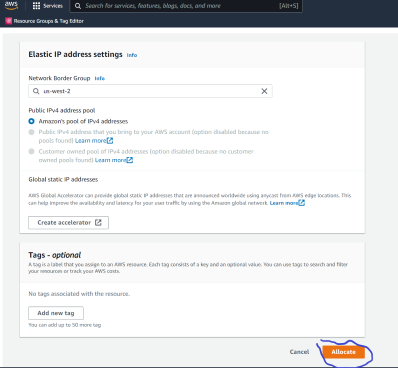
* Add a fourth and final subnet and give it a name letting you know it is the third private subnet
* Put it in a different availability zone from the rest of your subnets and give it a CIDR of 192.168.4.0/24



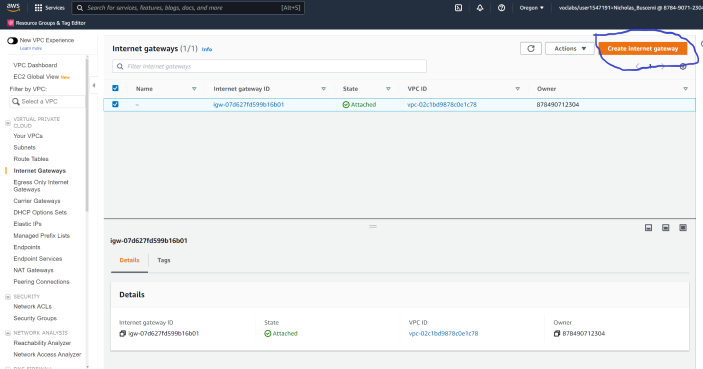
* Set up for route tables
* Allocate an Elastic IP address by going to Elastic IPs on the left hand side and click “Allocate Elastic IP address”



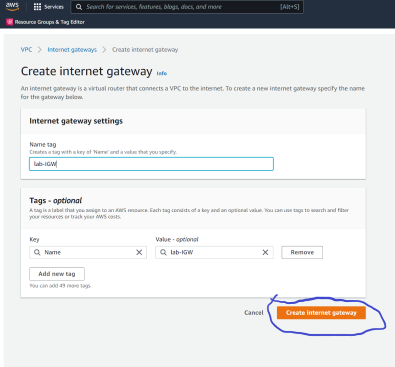
* Everything should be good as default but make sure that you are in the same region you have been creating everything in and then press “Allocate”. You can also add a name tag if you wish but it isn’t necessary



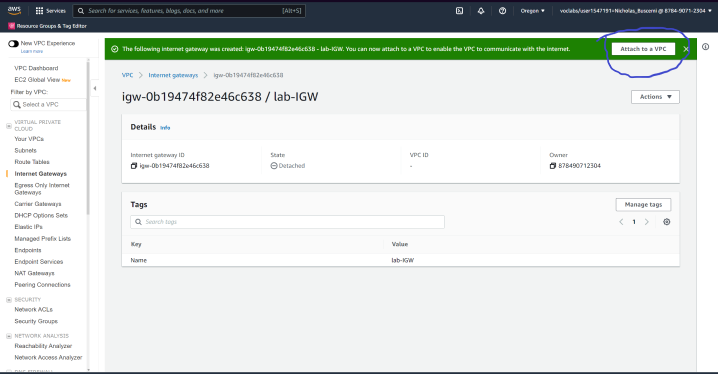
* Now create an internet gateway and attach it to the VPC by going to Internet Gateways on the left hand side and clicking “Create Internet Gateway”



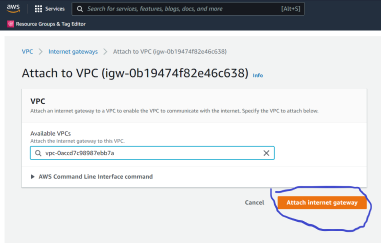
* Name it something similar to what is below and then click “Create Internet Gateway”



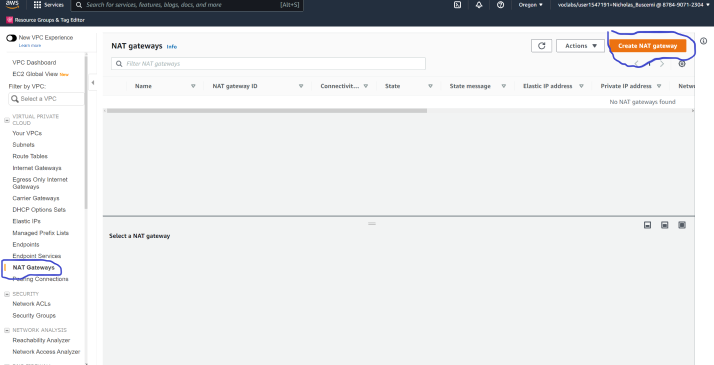
* Once it is created attach it to your VPC by clicking “Attach to a VPC” on the top of the screen



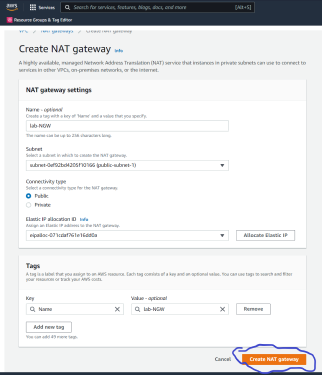
* Click the drop down and select your vpc that you made



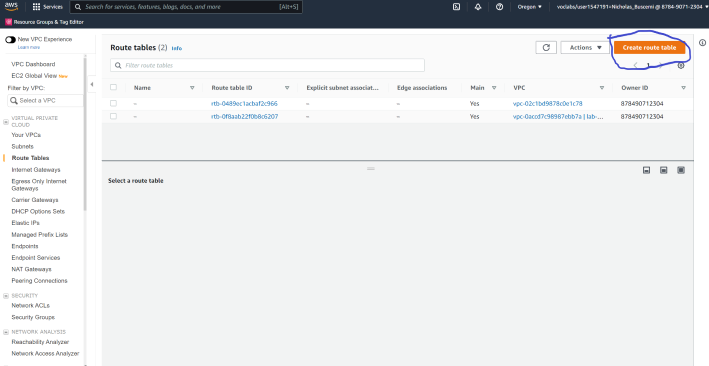
* Create a NAT Gateway by clicking on Nat Gateways on the left hand side and then clicking “Create NAT Gateway”



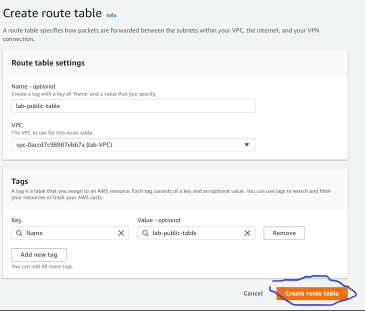
* Give it a name similar to the one below and assign it to a public subnet
* Click the drop down for Elastic IPs and click the one you created previously
* Click “Create NAT gateway”



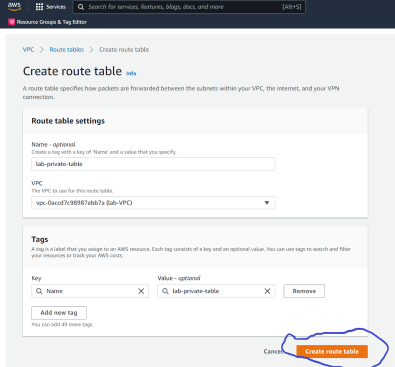
* Create Route Tables by first heading to “Route Tables” on the left hand side
* Click “Create route table”



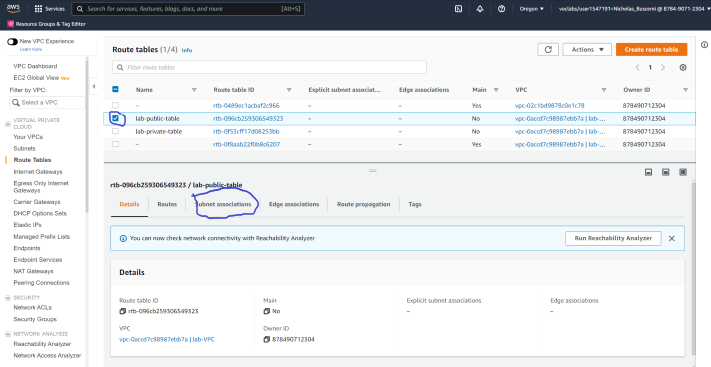
* Give it a name letting you know this is the public route table for your lab
* Assign your VPC to it and click “Create route table”



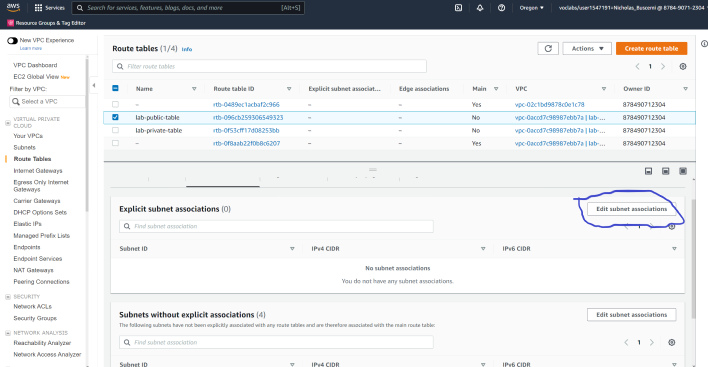
* Make a second route table naming it something to let you know that this is the private route table for your lab and assign your VPC to it



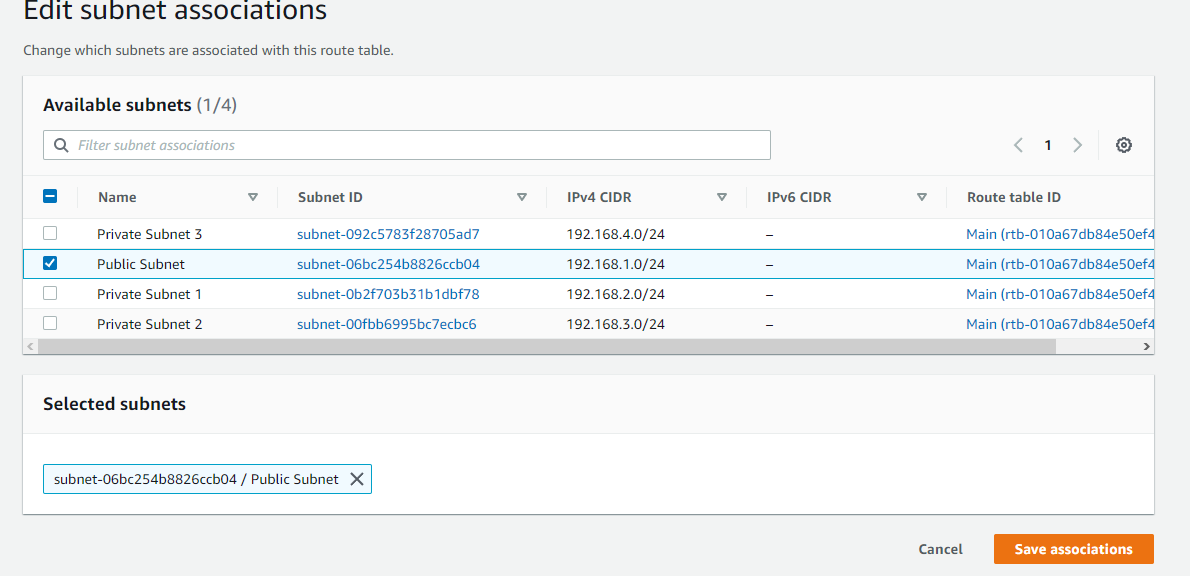
* Now associate your subnets with their respective route table
* Click on the public route table and click on “Subnet association” next to “Details”



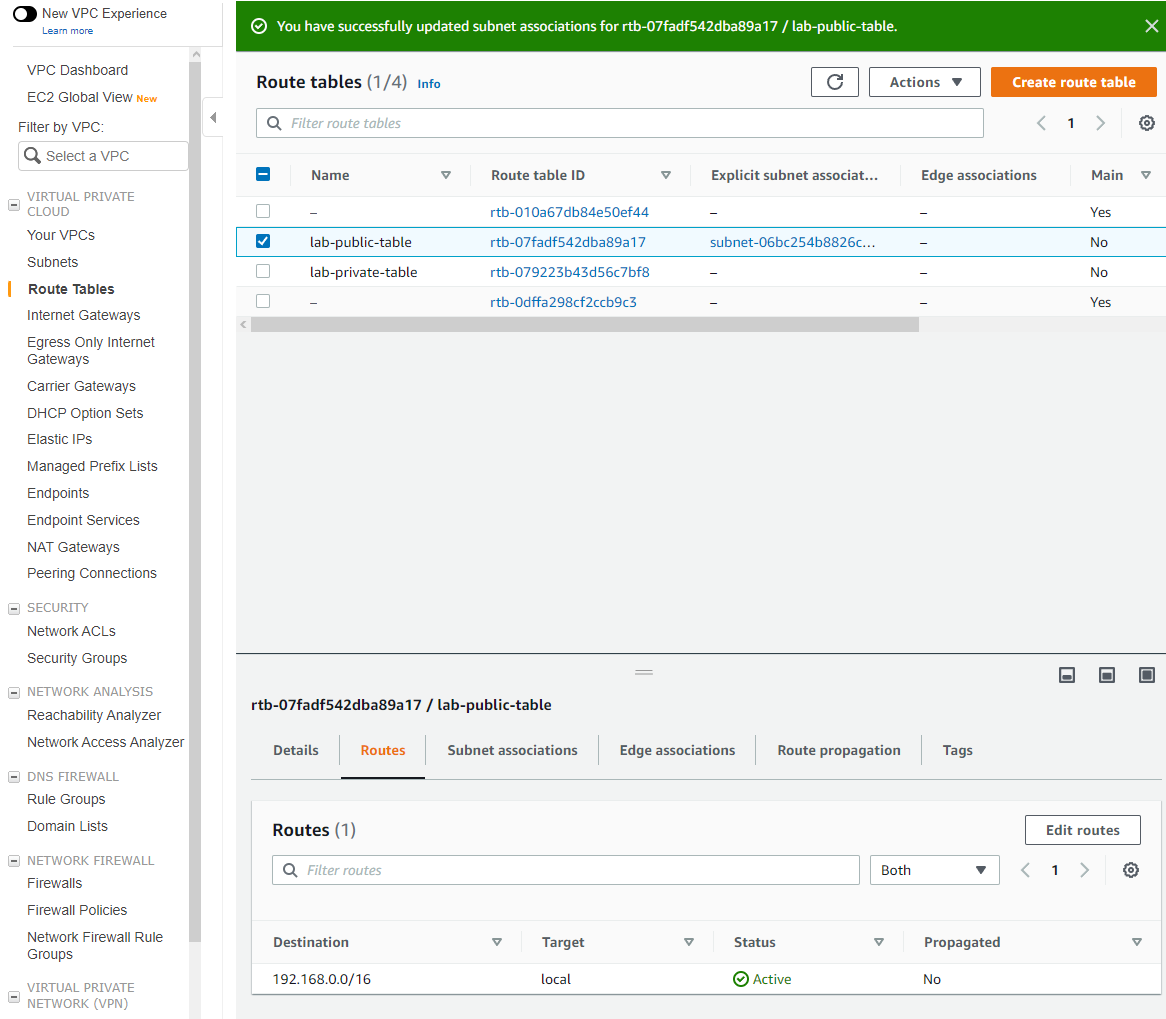
* Click on “Edit subnet associations”



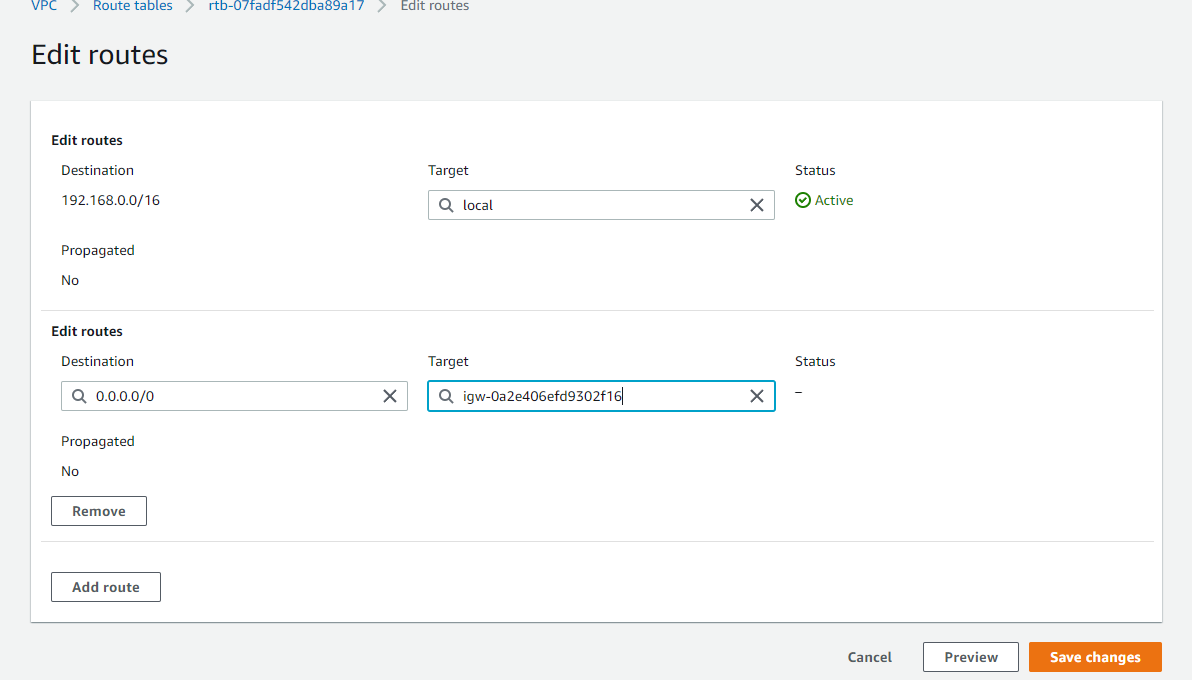
* Click on your public subnet and then click “Save associations”



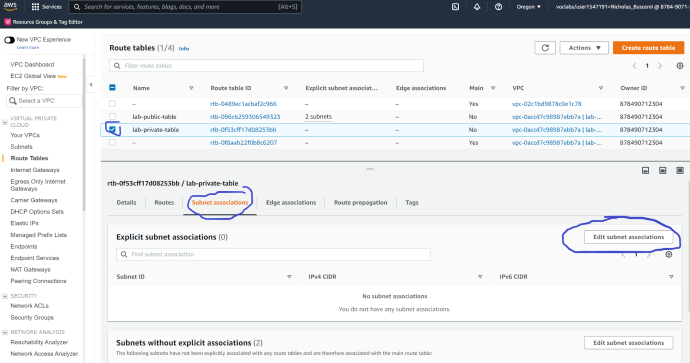
* Now add a route to our public route table to get access to the internet gateway
* Click on “Routes” next to “Details” and click “Edit routes”



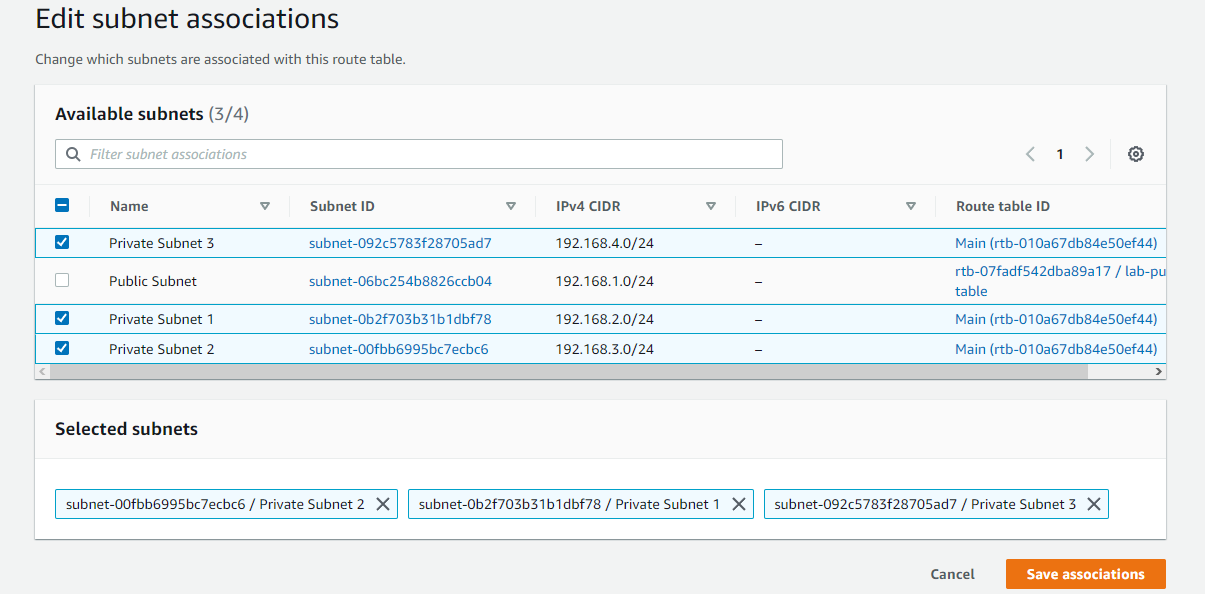
* Add a new route having a destination of anywhere and a target of your internet gateway and click “Save changes”



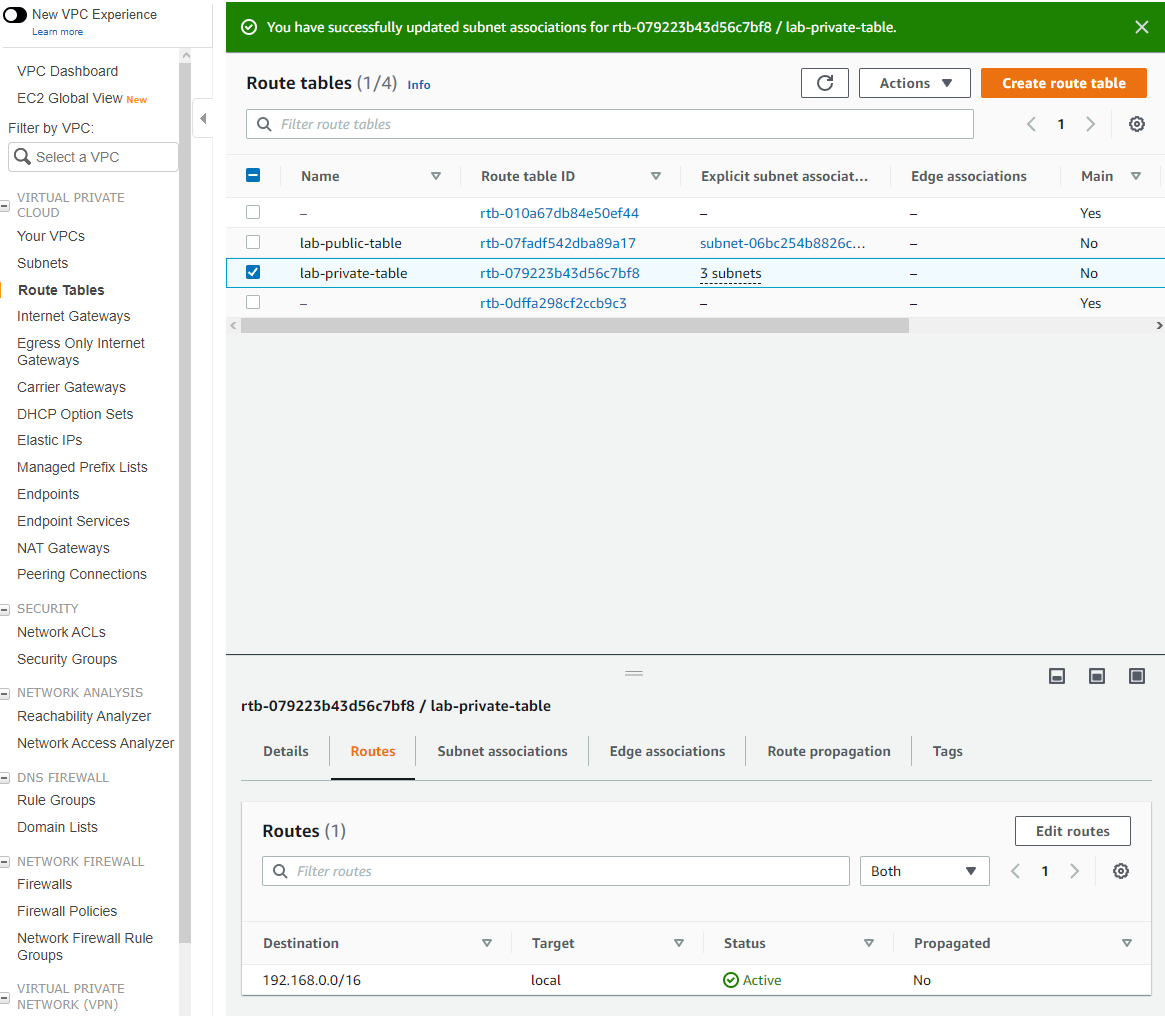
* Do the same thing for your private route table by clicking on it and going to its subnet associations and editing them



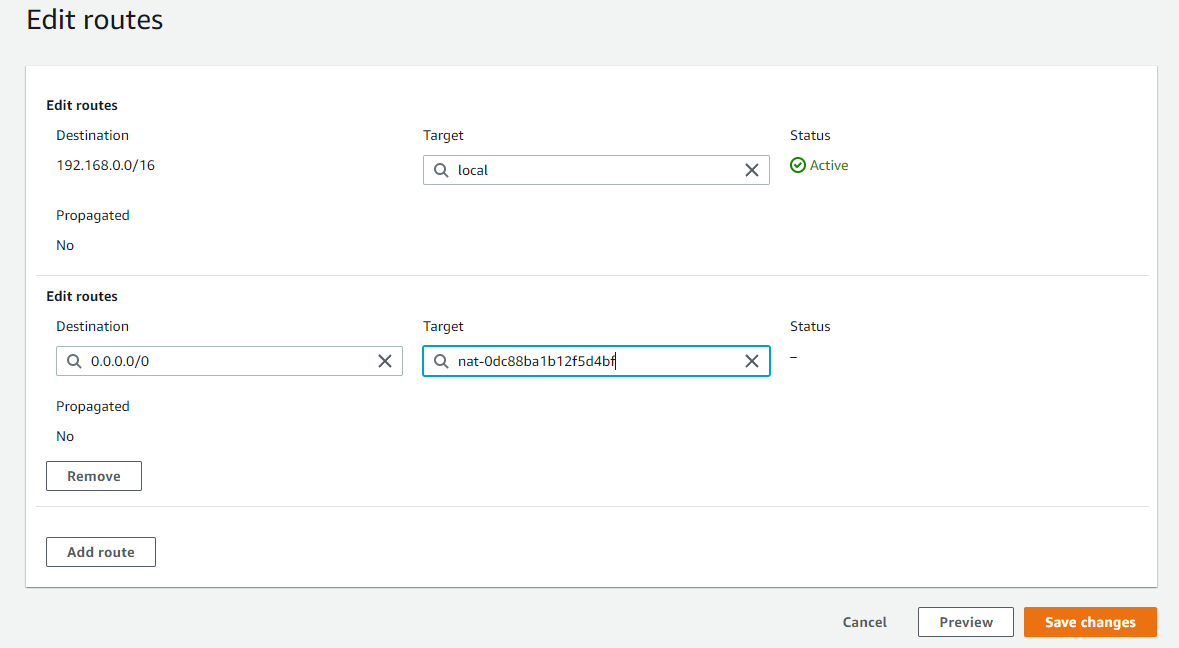
* Click on all three of your private subnets and save the associations



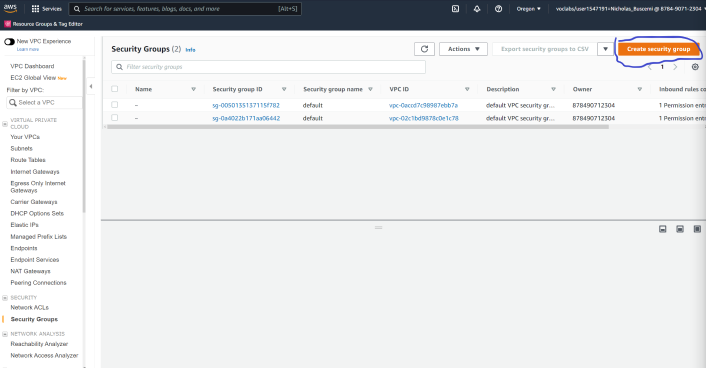
* Go to edit the routes of the private table



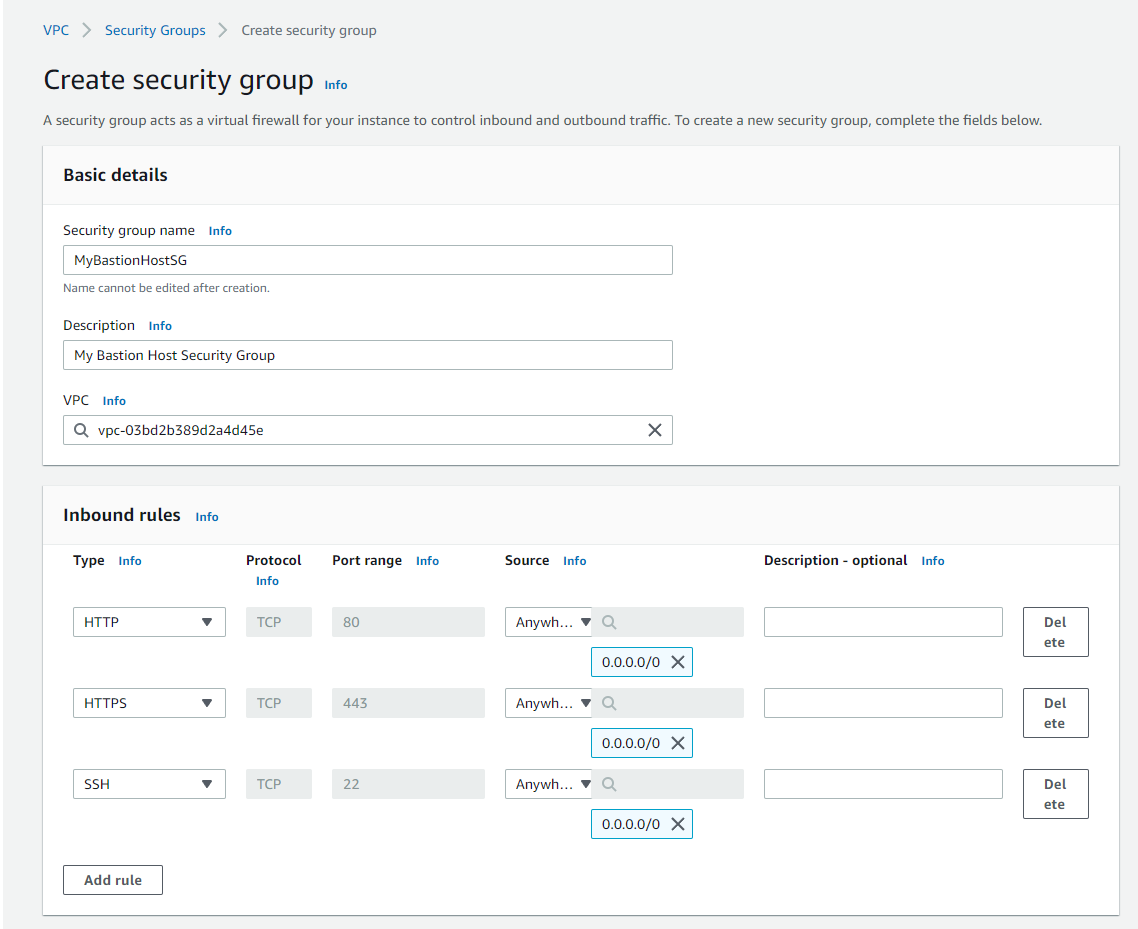
* Add a route to the private table that has a destination of anywhere and a target of your Nat gateway that you made earlier



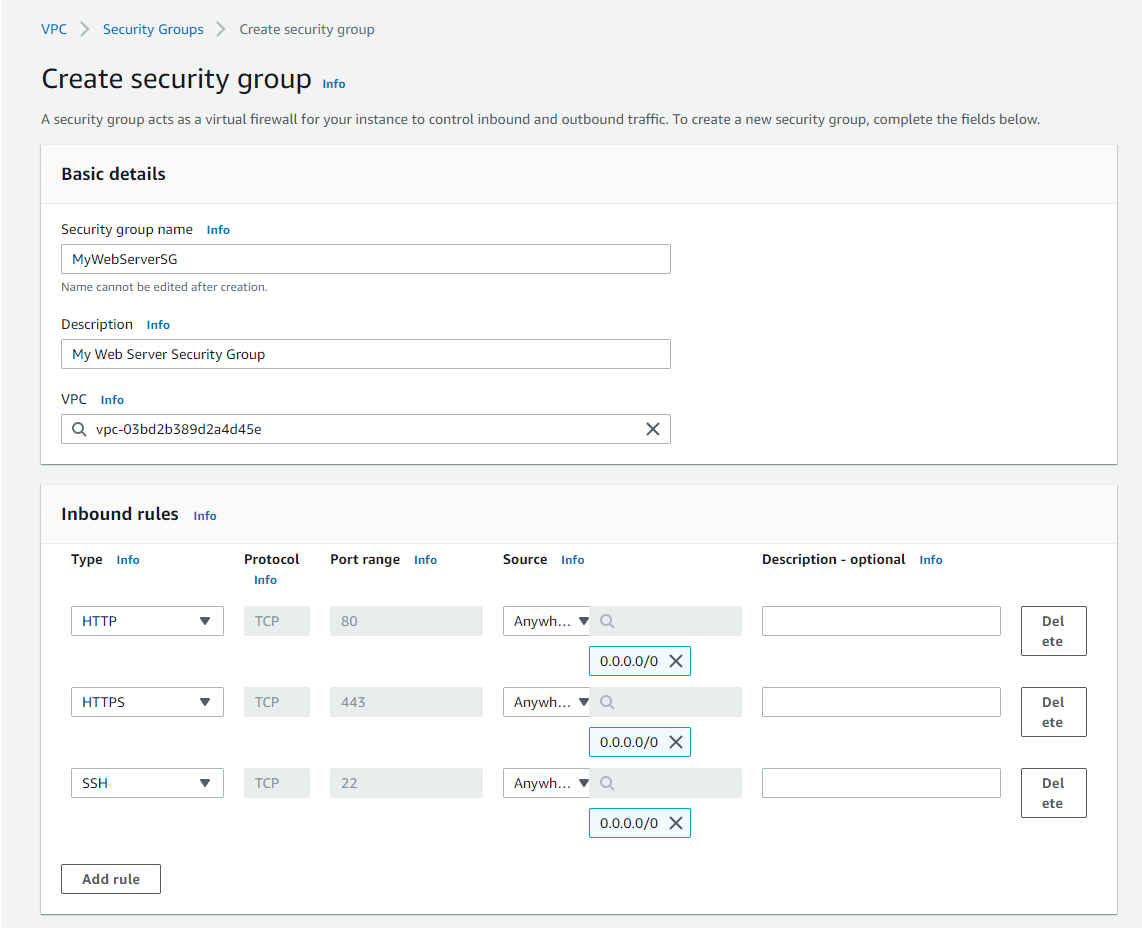
* Now to create our security groups (One for our bastion host, web server, app server, and our database) we will head to Security Groups on the left and click “Create security group”



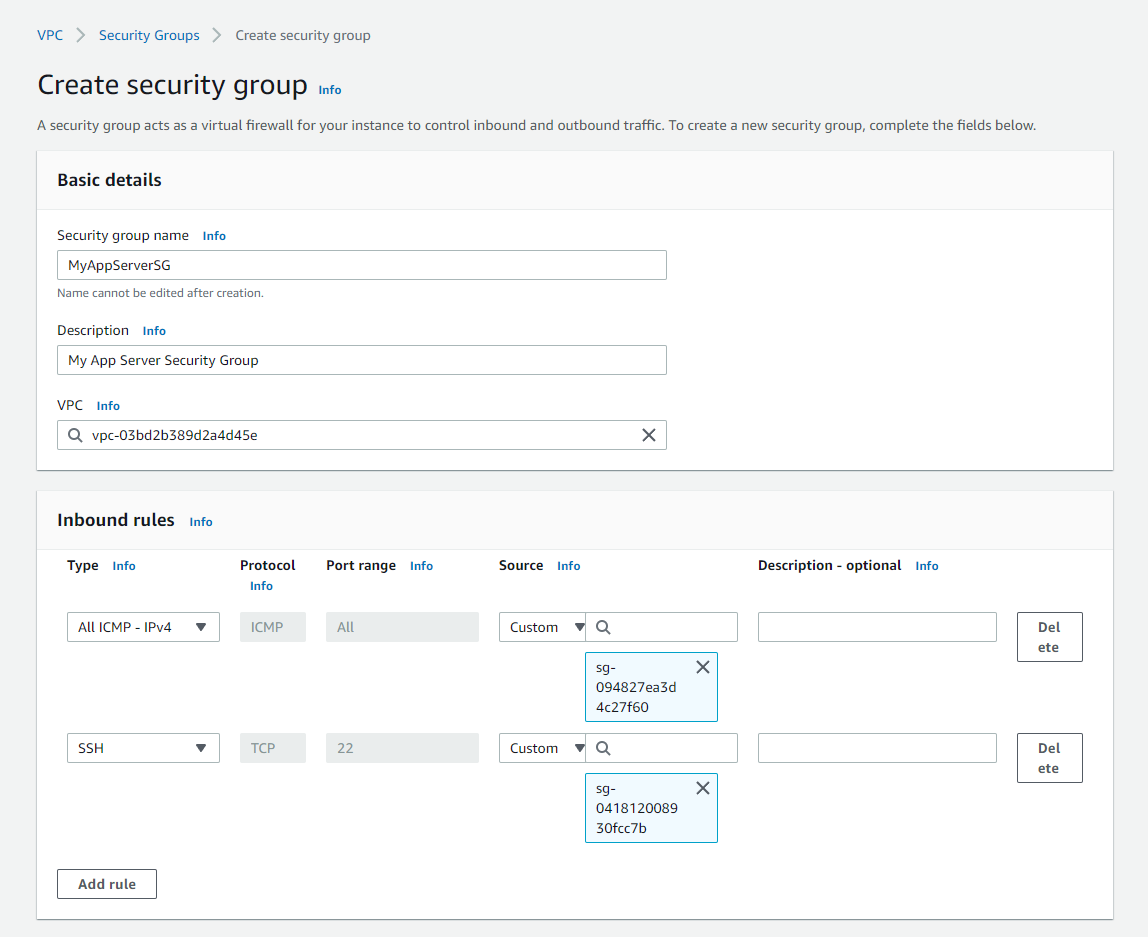
* Give it a name and description letting you know it is for a bastion host
* Assign your VPC to it
* Give it three inbound rules, one for SSH using your IP and one for HTTP using 0.0.0.0/0 as well as https using 0.0.0.0/0



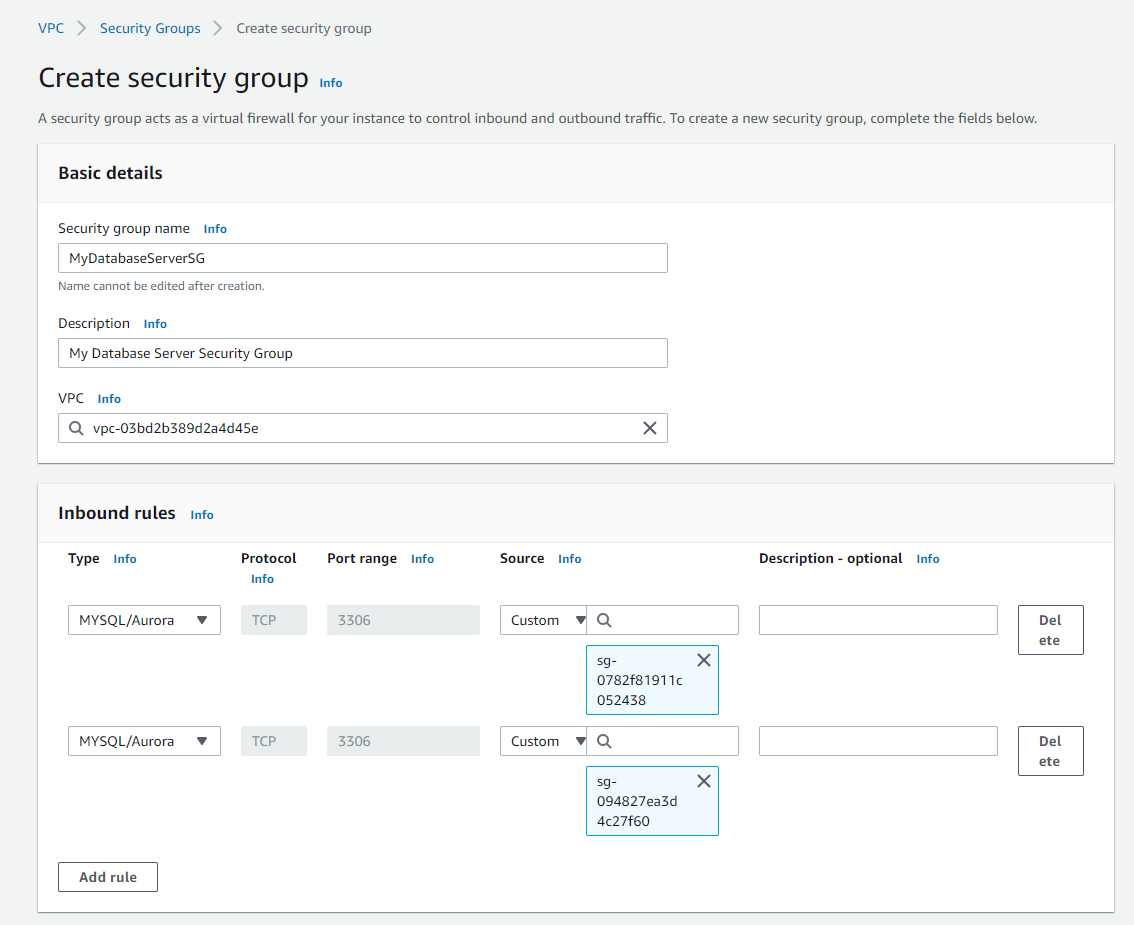
* Create another security group
* Give it a name and description letting you know it is for a Web server
* Assign your VPC to it
* Give it the same inbound rules as the Bastion Host security group



* 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



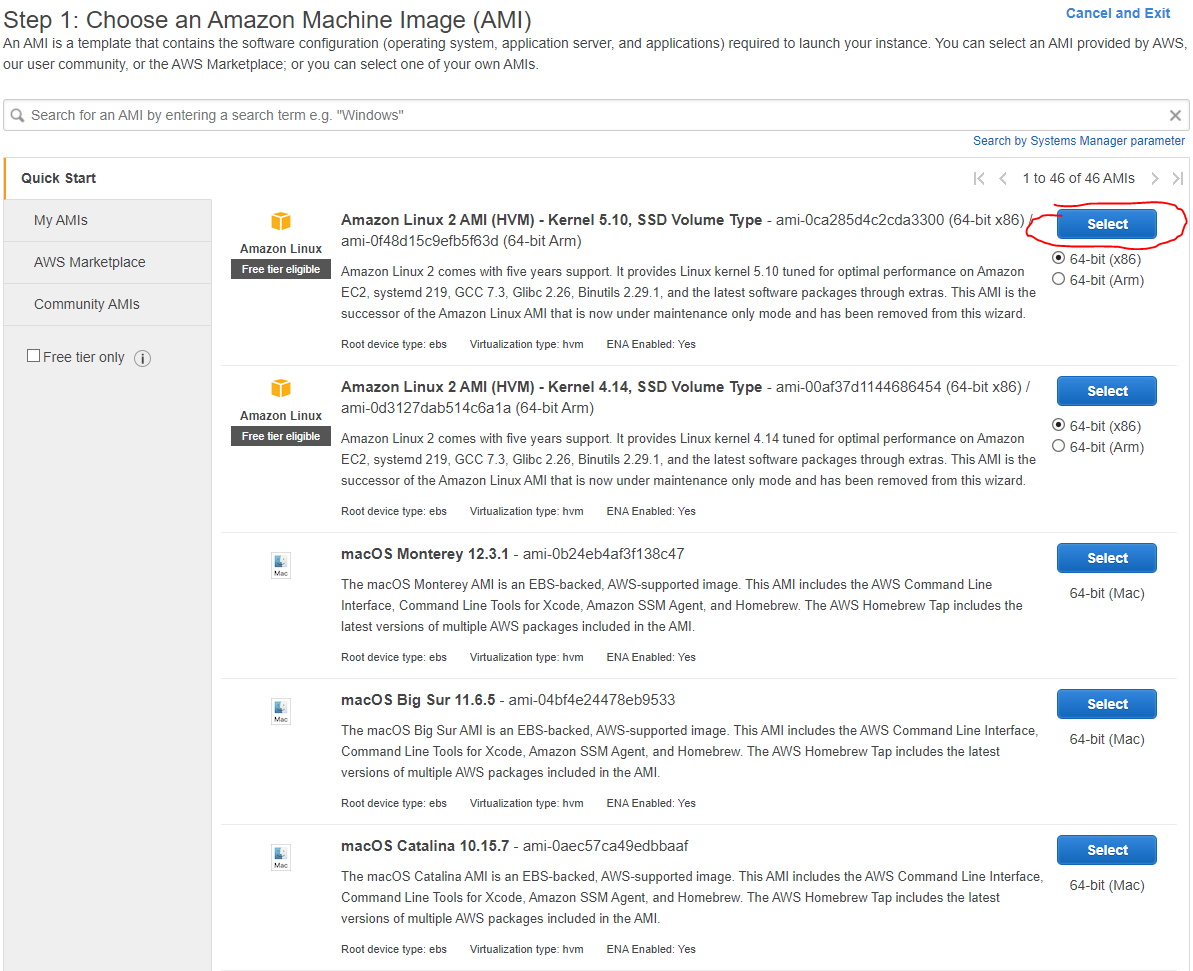
* Create one final security group
* Give it a name and description letting you know it is for a database server
* Assign your VPC to it
* Give it two inbound rules both for MYSQL/Aurora and give one of them a source of your app server SG and the other one a source of your bastion host SG



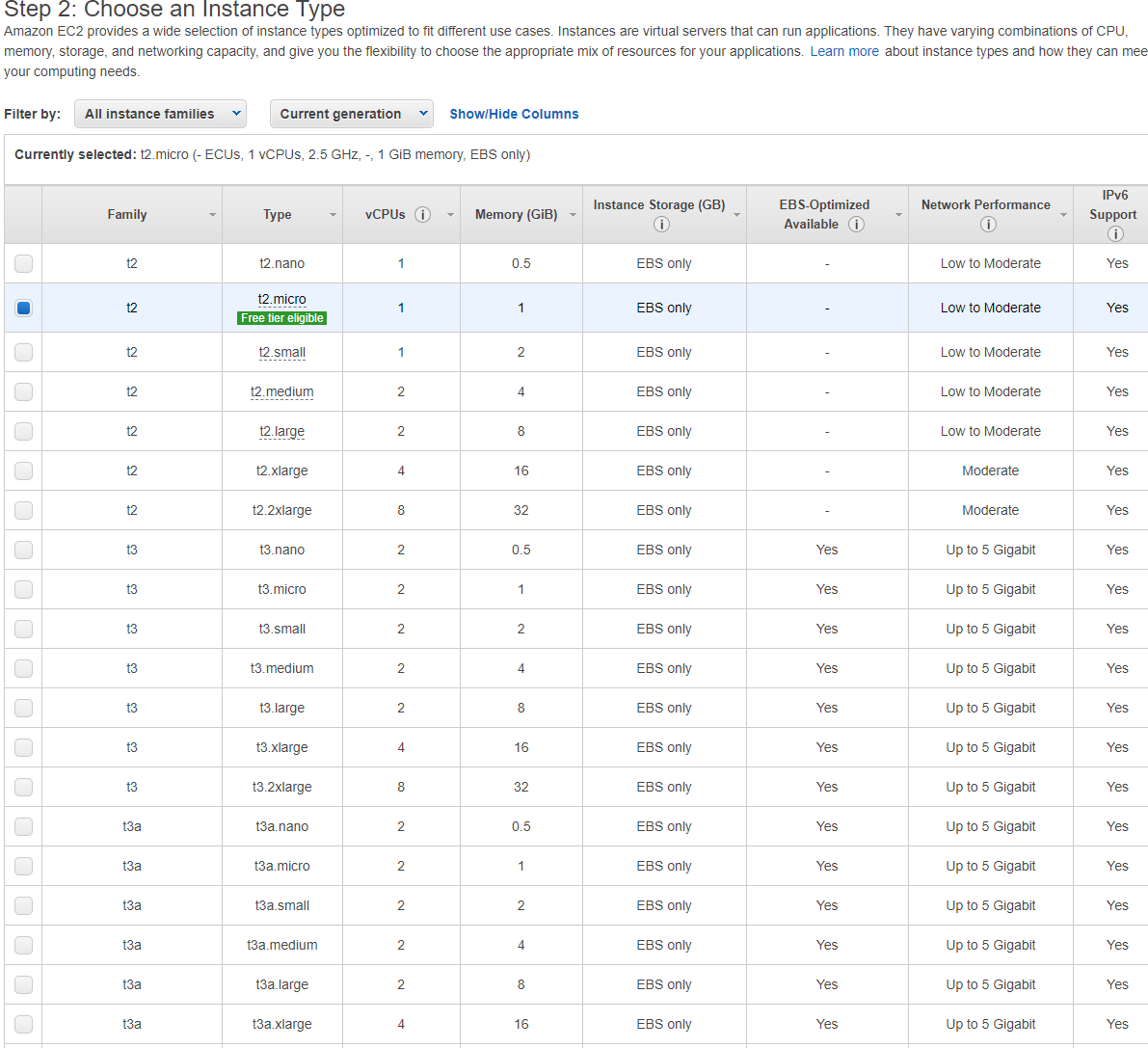
* Go back to your bastion host inbound rules and add one more for MYSQL/Aurora and a source of your database SG
* Go back to your web server inbound rules and add one more for All ICMP - IPv4 and a source of your app server SG
* Go back to your app server inbound rules and add one more for MYSQL/Aurora 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

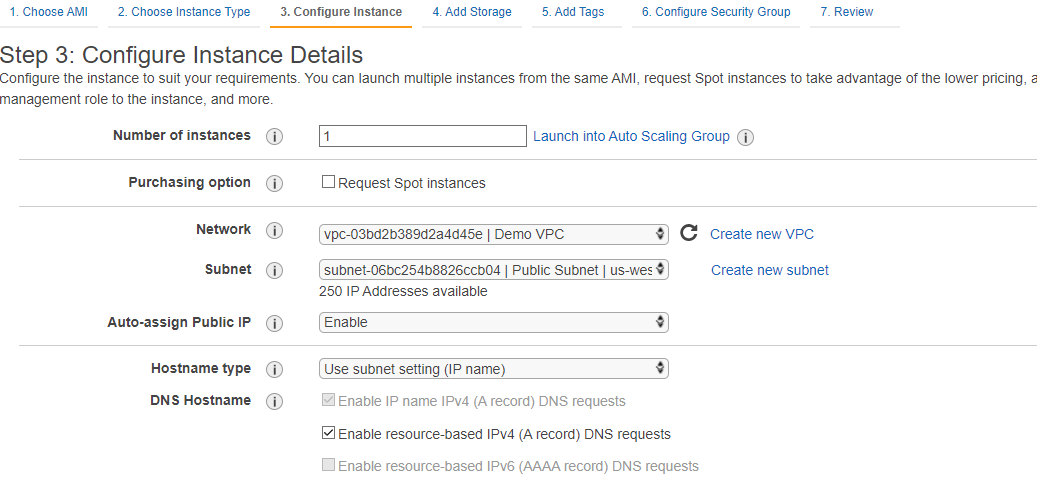
* Create Bastion Host
* Select Amazon Linux 2 AMI



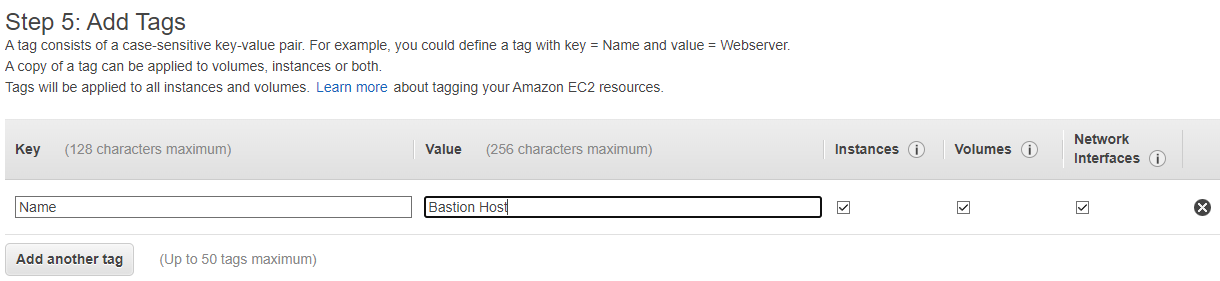
* Select t2.micro



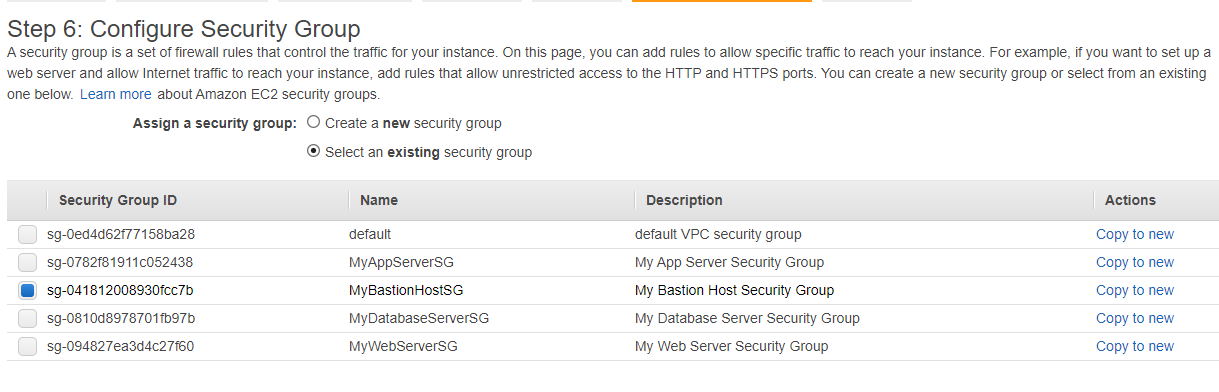
* Put in your VPC and Public Subnet and enable auto assign public IP



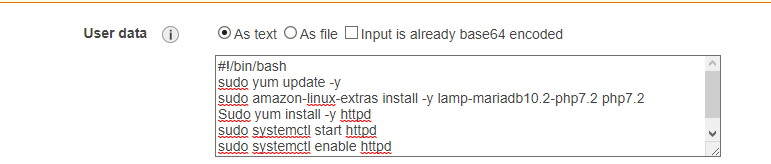
* Storage leave default
* Add a name tag to let you know this is the Bastion Host



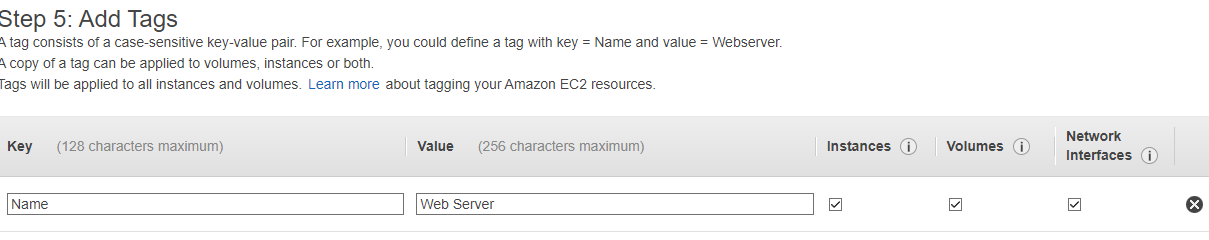
* Select an existing group and select your Bastion Host SG



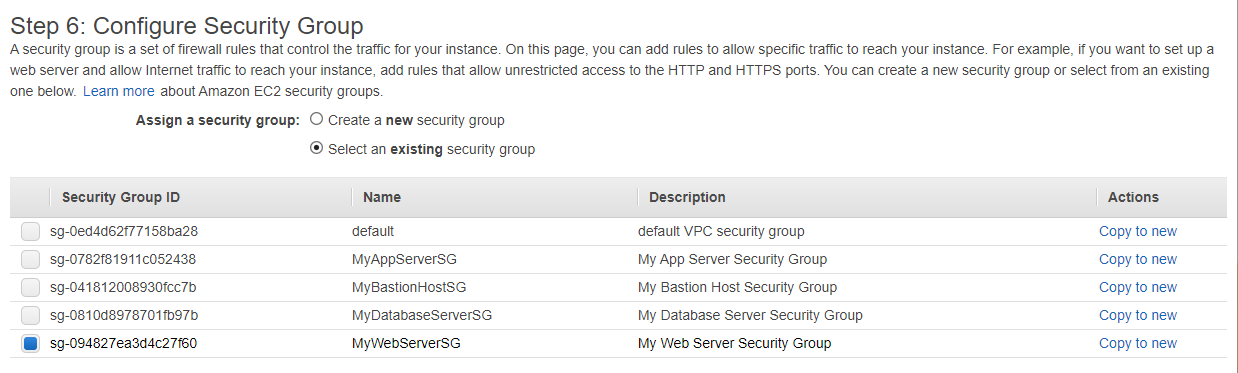
* Launch and choose an existing keypair. This can be downloaded from the lab page
* To make the Web Server follow the same steps until you get to Step 3
* Follow along like previously and change your network, subnet, and enable auto assign public ip
* Then go to user data and type this into it to set 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



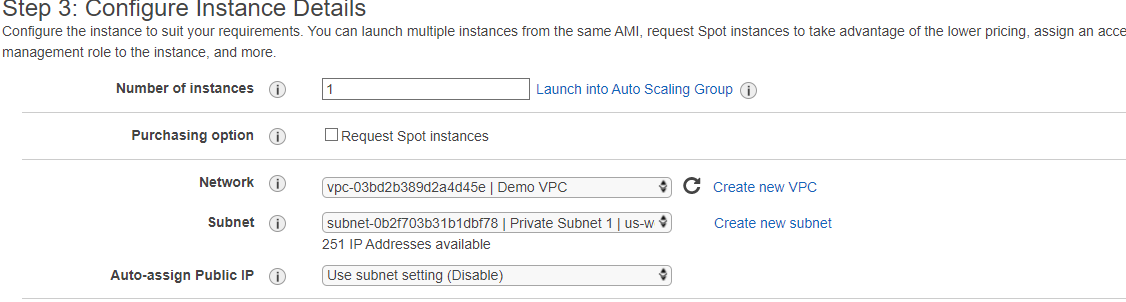
* Storage leave default
* Give it a name tag letting you know it is the Web Server



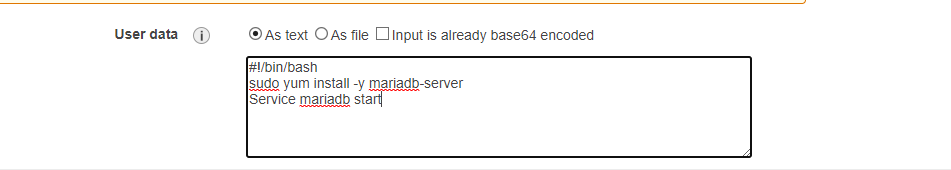
* Select an existing security group and select your web server SG



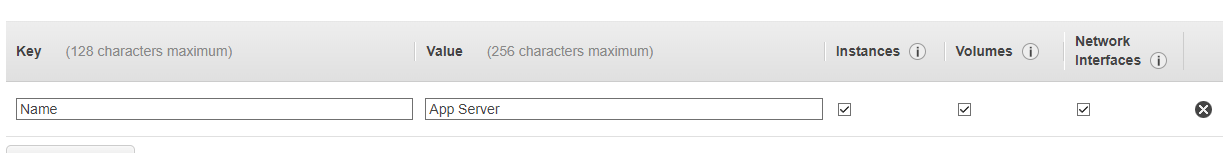
* Just like before launch and use the existing keypair
* Follow the same steps once again to create the app server until you get to step 3
* Put in your VPC and then choose Private Subnet 1 for the subnet and leave auto assign public ip disabled



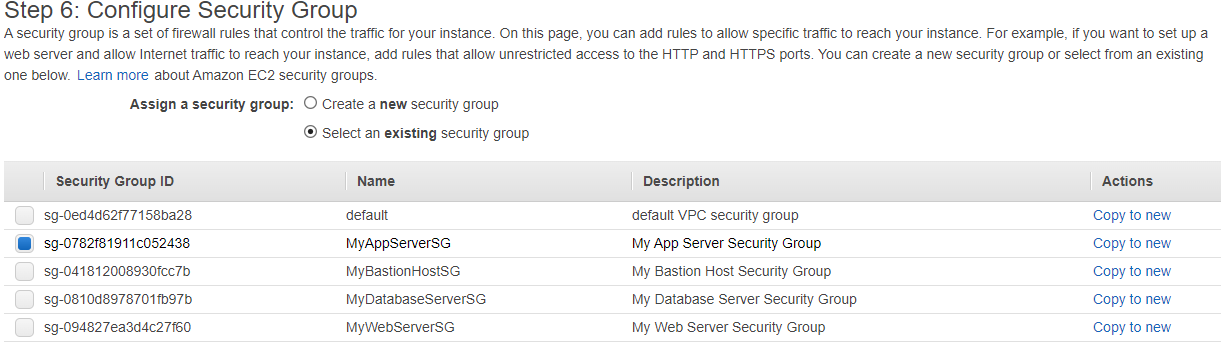
* Then go into metadata and type this out to set up a database server on our app server
  + #!/bin/bash
  + sudo yum install -y mariadb-server
  + Sudo service mariadb start (note that the image is incorrect, make sure to add sudo)



* Give it a name letting you know it is the App Server



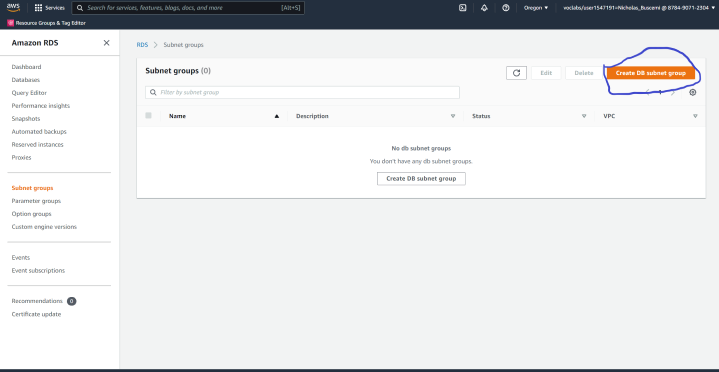
* Select an existing security group and select the app server SG



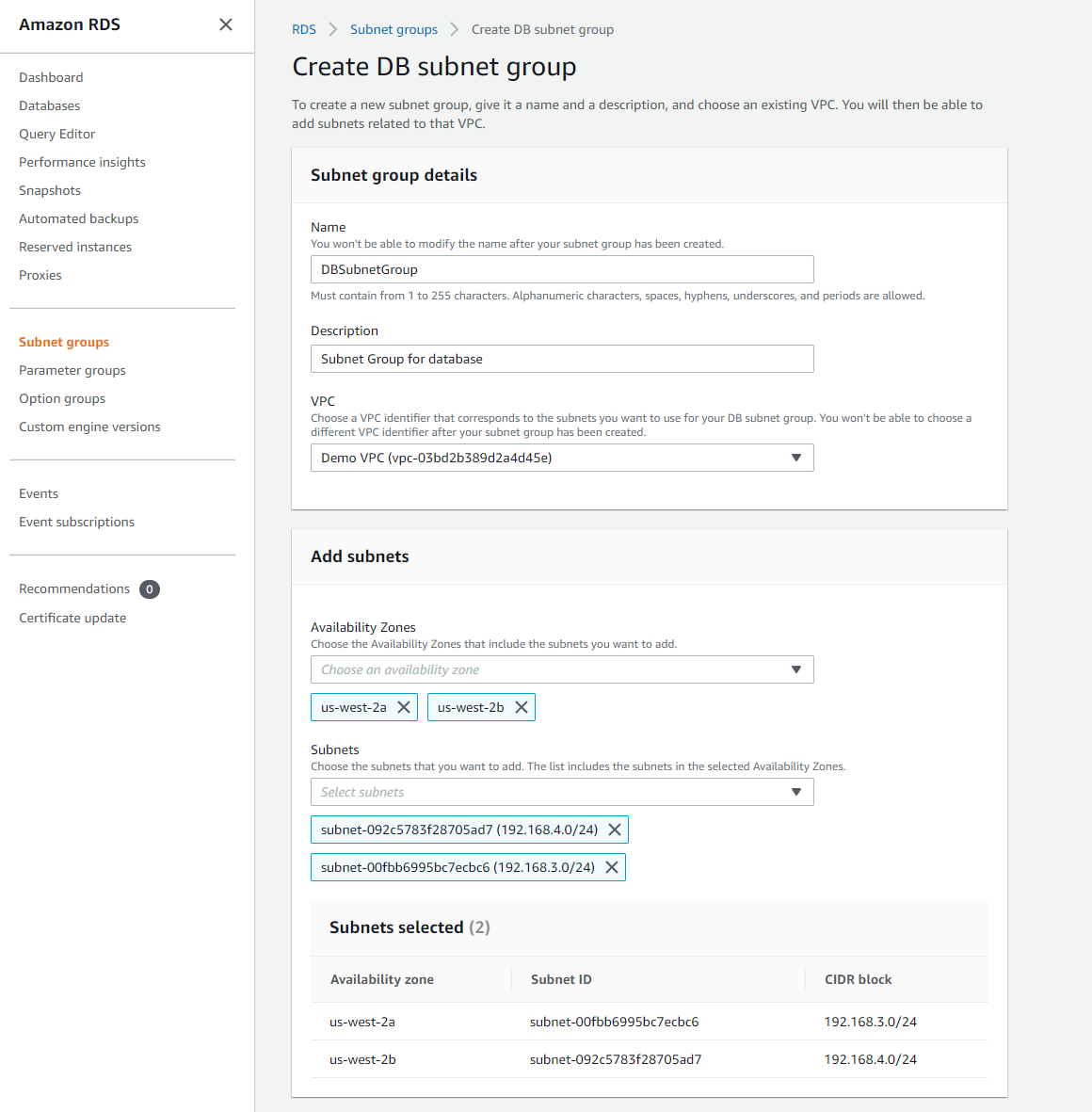
* Just like before launch and use the existing keypair

Step 3: Create a Database

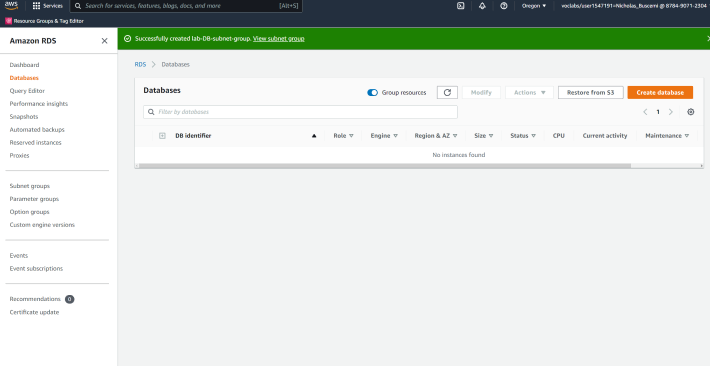
* Create a DB subnet group by first heading to the Amazon RDS service page on the AWS management console
* Click on Subnet Groups on the left hand side and the click on “Create DB subnet group”



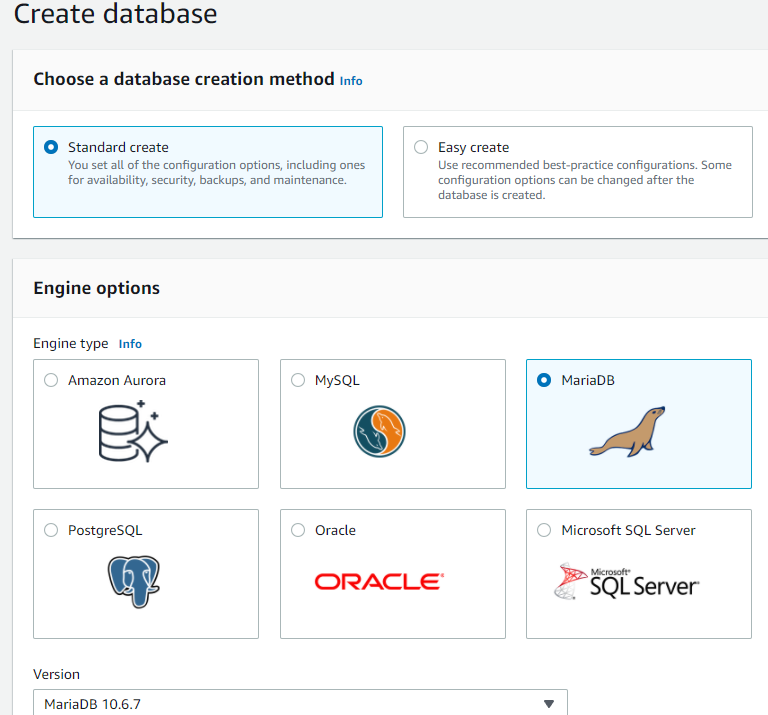
* Give it a name and description letting you know what it is and then assign your VPC to it
* Put in the availability zones you used for your subnets
* Select subnets 3 and 4
* Click create



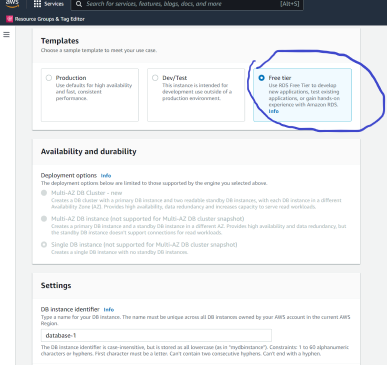
* Go to Databases on the left hand side and click on “Create Database”



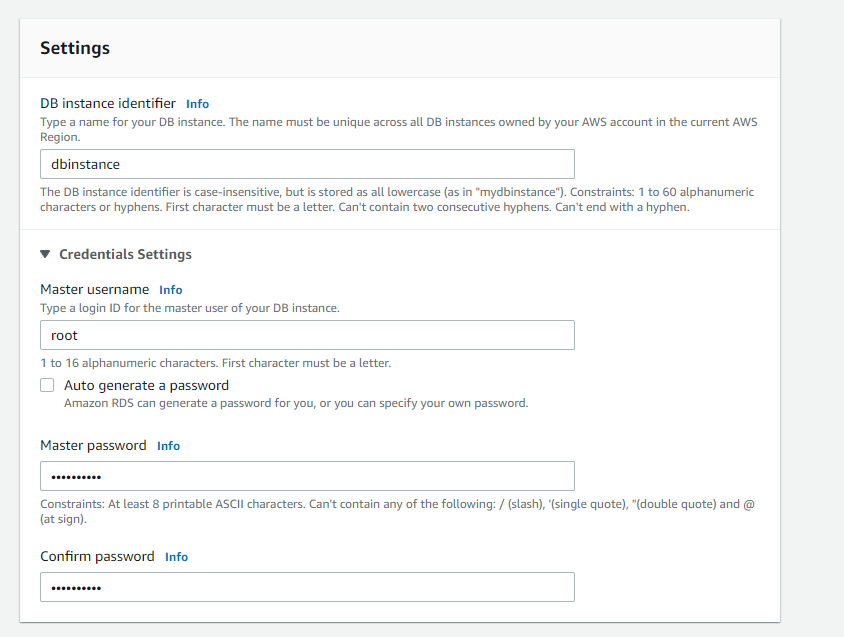
* Click on Standard create and MariaDB for the engine type



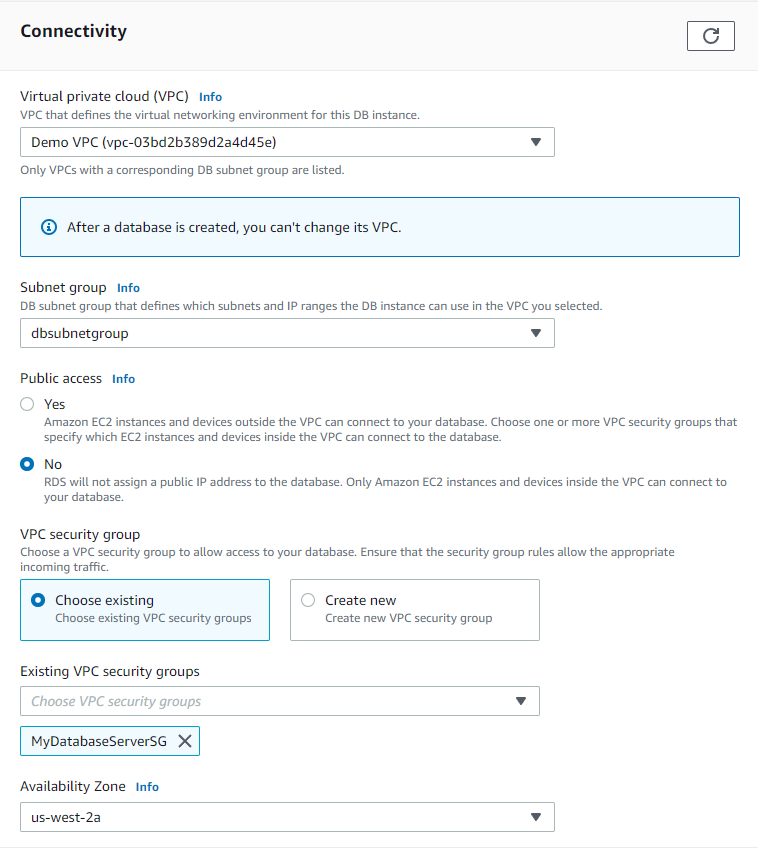
* Make sure you click on Free tier here



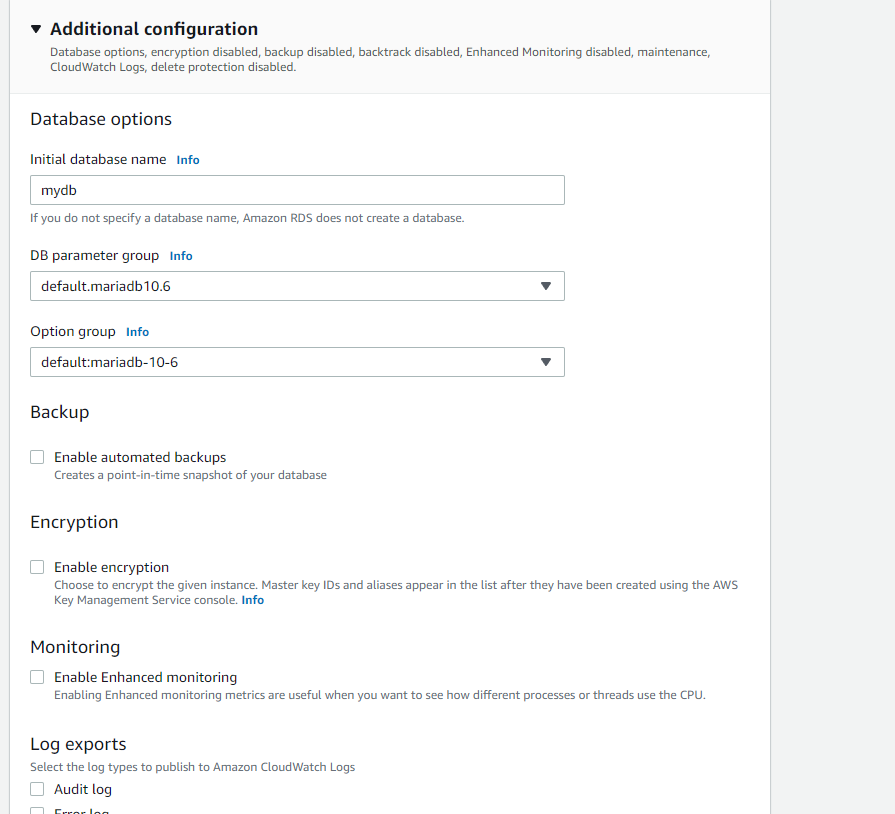
* Give it an identifier you can easily identify it with
* Give it a master username or leave it as default admin. For the purpose of these instructions I will be using root
* Give it a password that you write down somewhere else to make sure you have the correct one. For the purpose of these instructions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

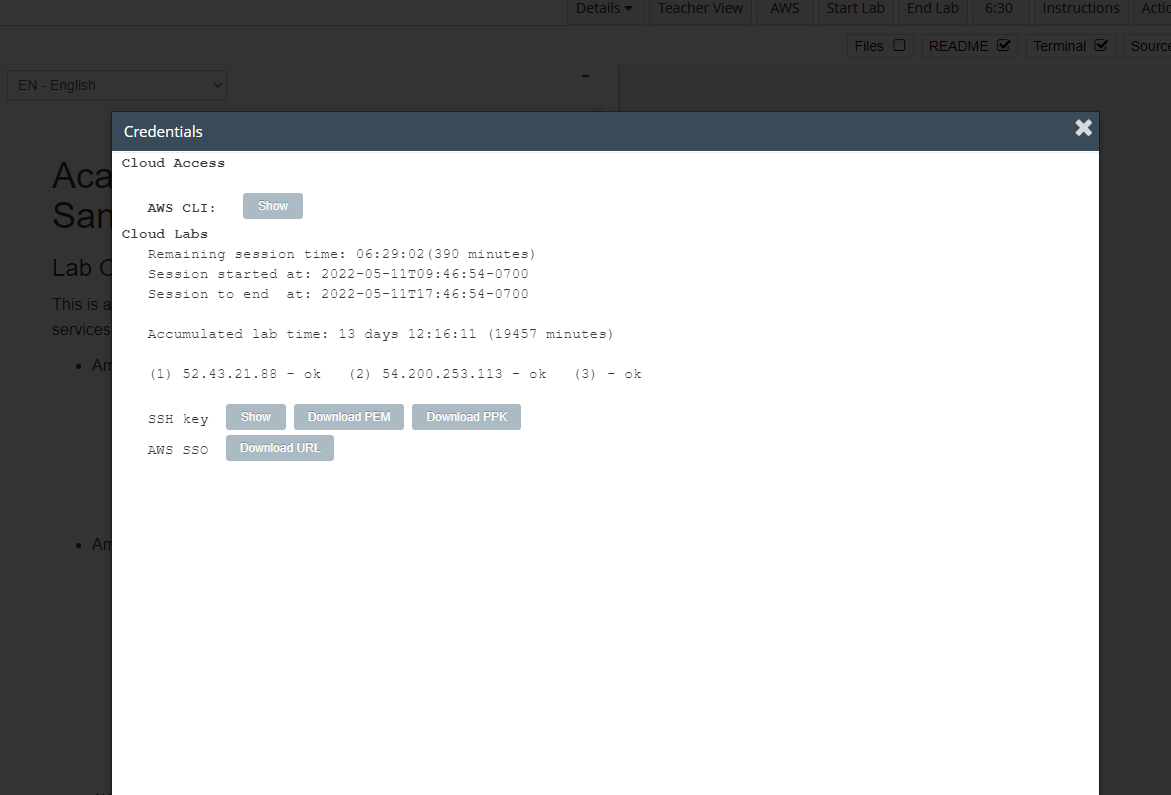


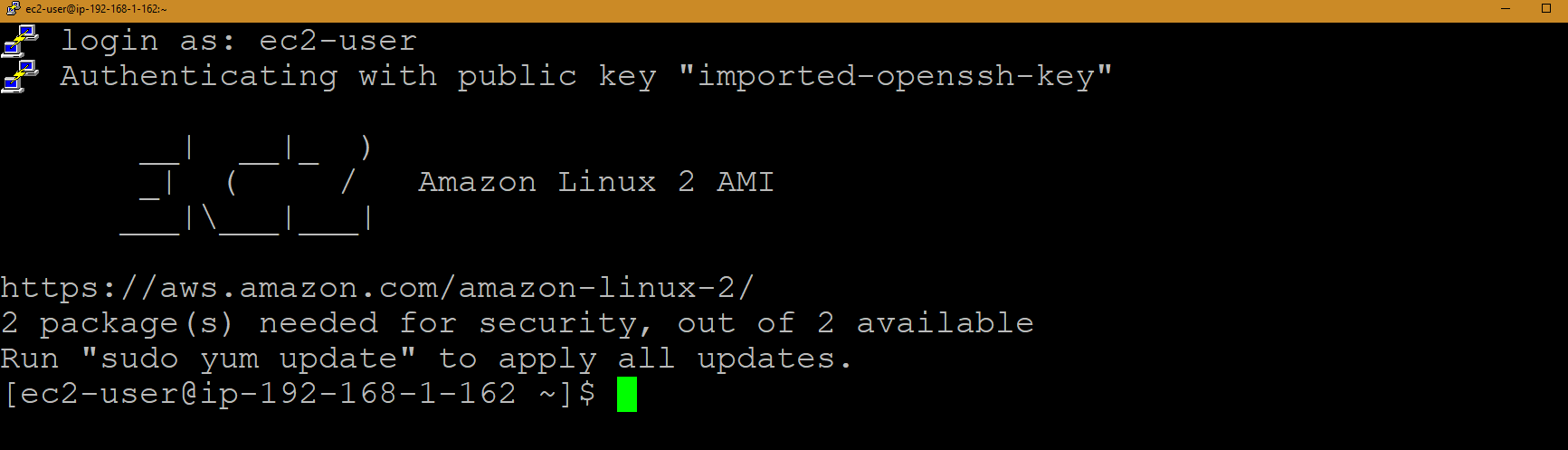
* Scroll down to Additional configuration on the bottom and give it an initial database name and save it in the same spot as your password since it will be used later
* Disable automated backups and encryption since they are not needed (These are normally best practice to leave enabled but the database 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



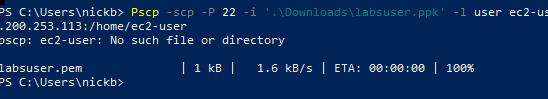
Step 4: Test connections

* Ssh into your Bastion Host after downloading both the pem and ppk files from the lab environment

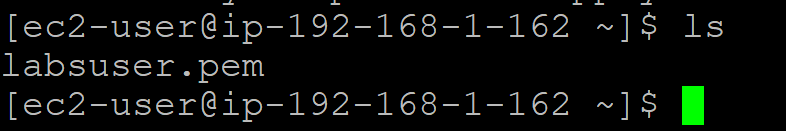




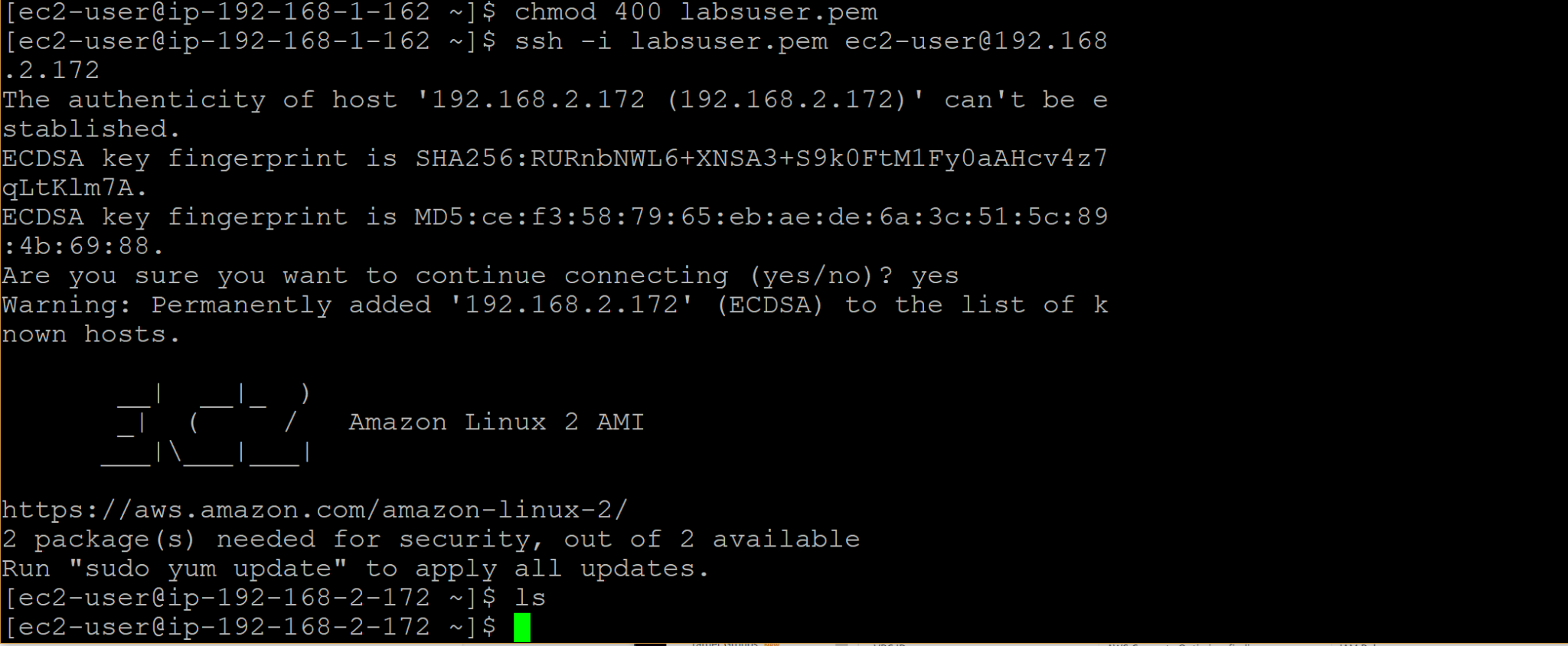
* This is for windows only as I only have a windows machine to work on, sorry mac and linux users
* Go into your powershell and type this command out
  + Pscp -scp -P 22 -i ’.\Downloads\labsuser.ppk’ -| user ec2-user ‘.\Downloads\labsuser.pem’ ec2-user@bastion-host-public-ip:/home/ec2-user
  + Replace 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



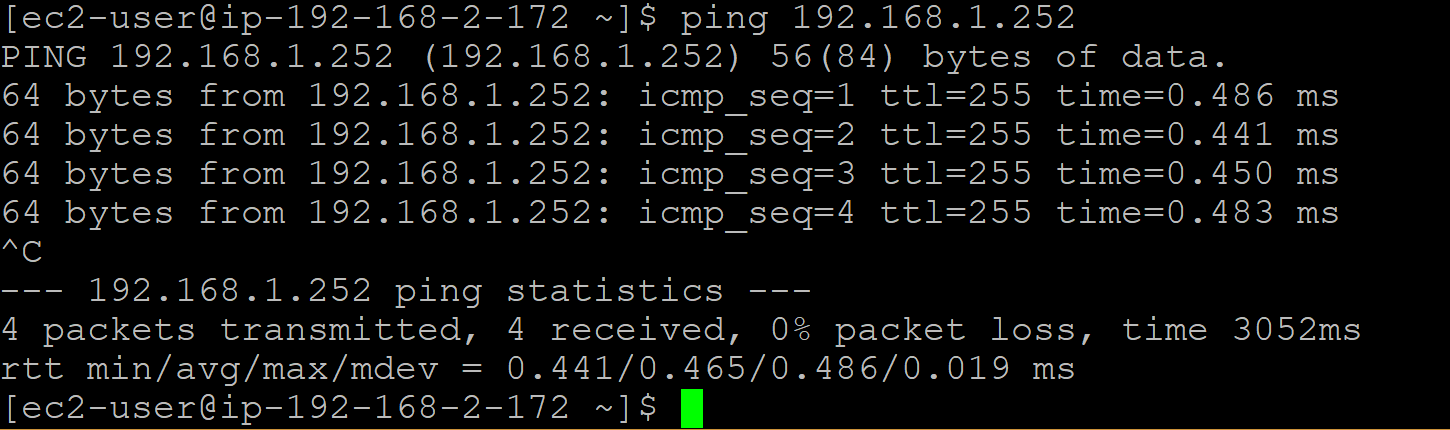
* Test on your ssh to see if the file is uploaded by using ls. Should return something like this



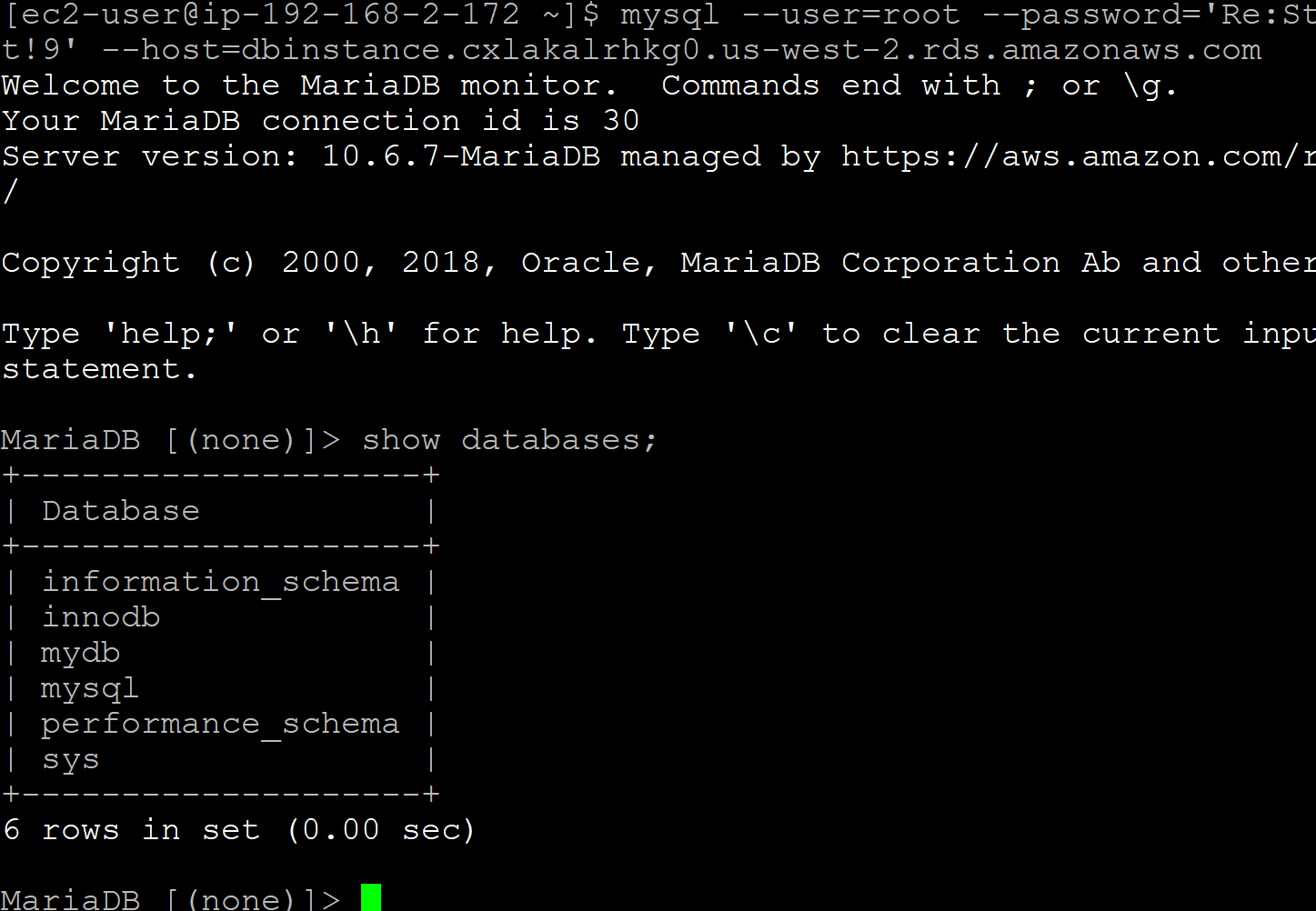
* Change file permissions for the file we just downloaded to our bastion host by typing
  + chmod 400 labsuser.pem
* Then ssh into our app server by typing
  + ssh -i my-key-pair.pem ec2-user@app-server-private-ip
  + Replace my-key-pair with the name of your key
  + Replace app-server-private-ip with 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



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



* 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



* This concludes the lab