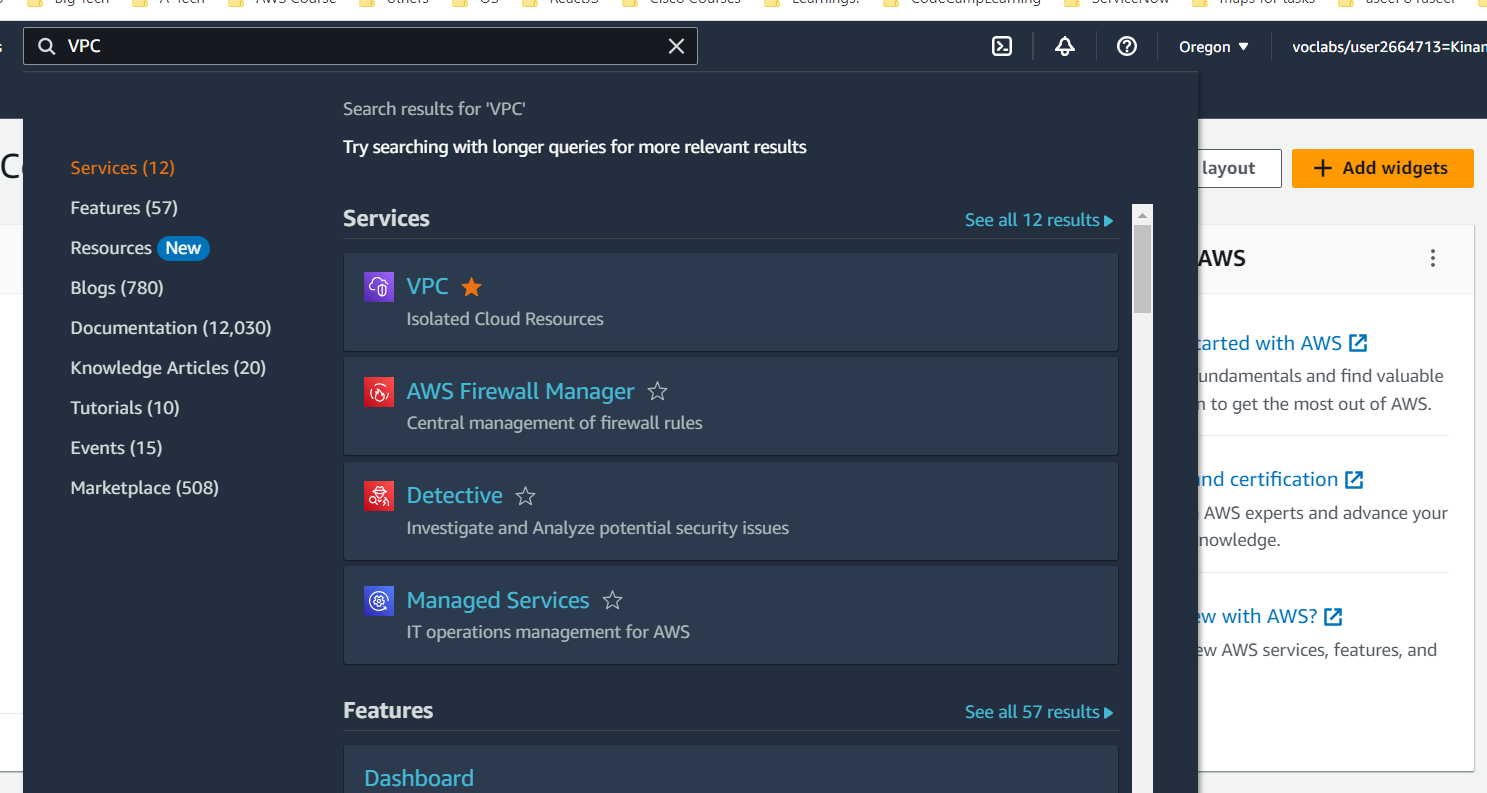
**Virtual Private Cloud (VPC) Setup and Configuration**

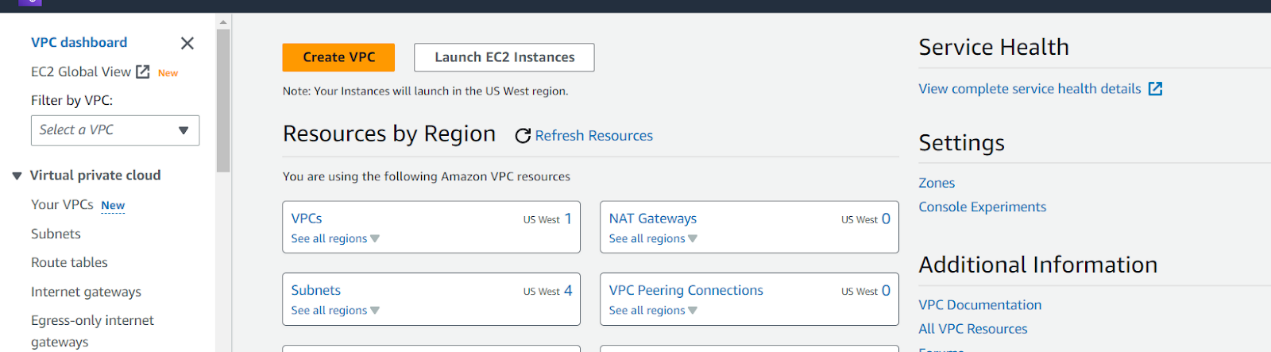
In this document, we walk through the process of setting up a VPC within AWS, configuring subnets, routing tables, and provisioning EC2 instances. The goal is to create a secure environment where an EC2 instance with SSH access can communicate with another EC2 instance hosting a MySQL server, without exposing the database to the public internet.

In order to complete this tutorial you must connect to amazon sandbox

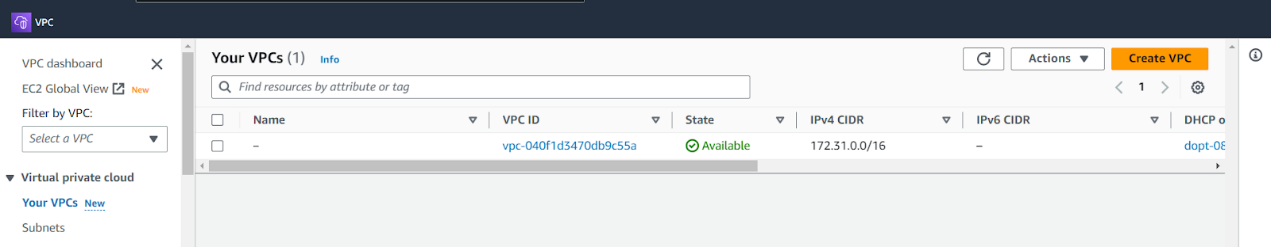
1) Create VPC

1.1) In the search bar, enter **VPC**  and **click** on the first option which is VPC****

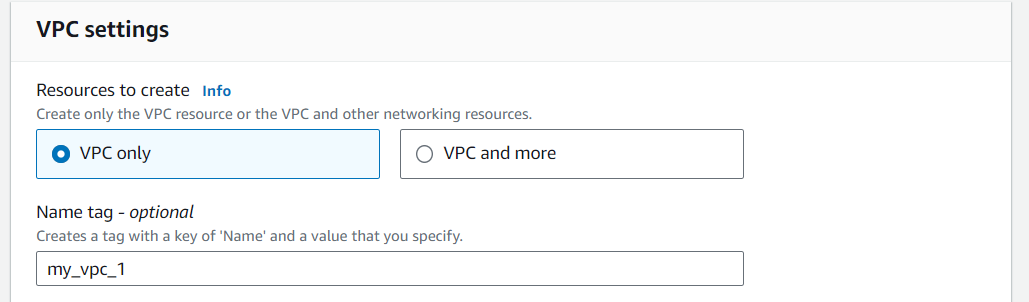
1.2) In the **navigation pane**, click **Your VPCs**. or click **Create VPC** and skip 1.3



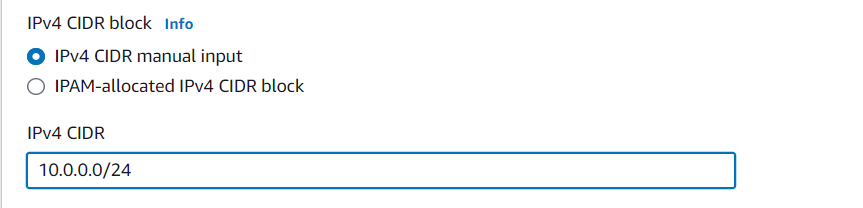
1.3) Click "**Create VPC**".

****

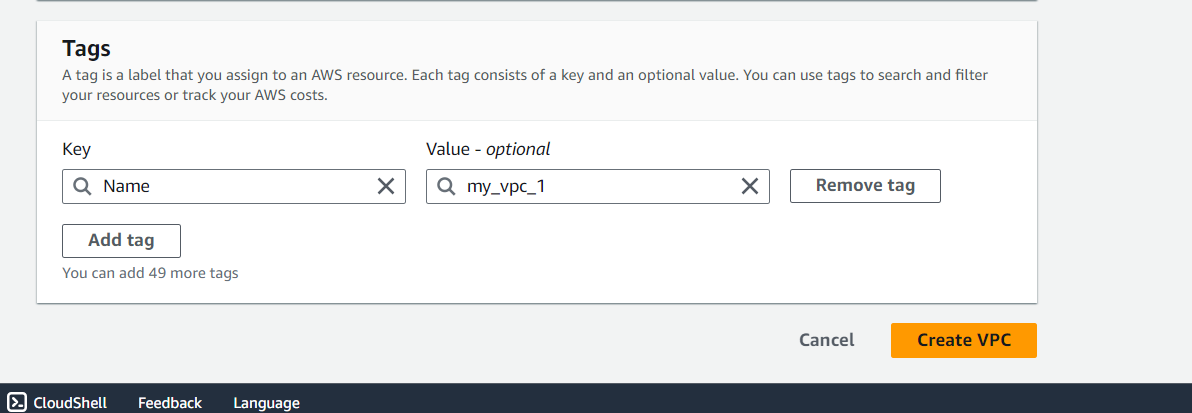
1.4) Enter a name for the VPC.

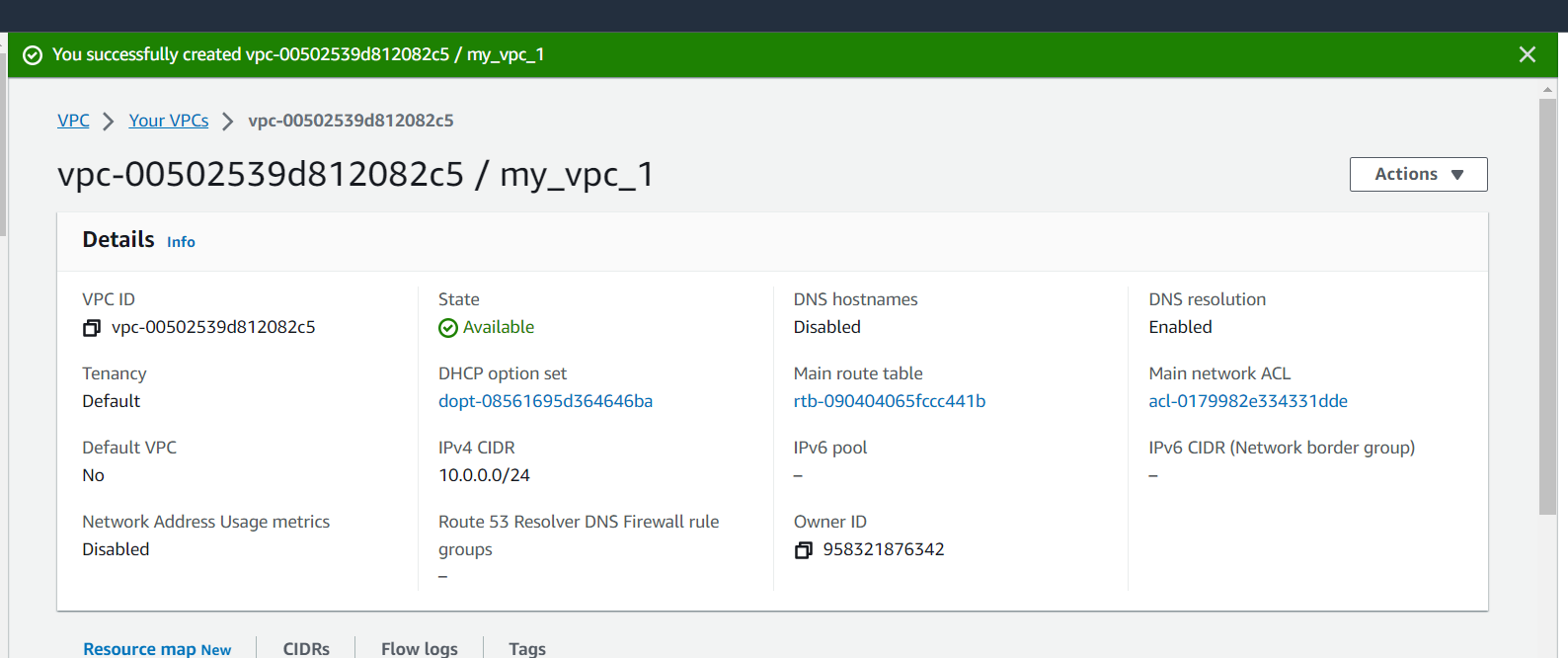


1.5) Specify an IPv4 CIDR block for the VPC.



1.6) Click "Create VPC".

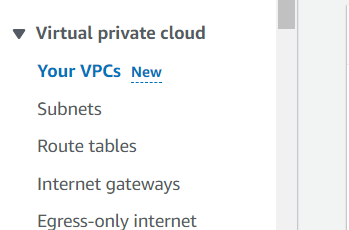
****

1.7) Then your supposed to get this message  


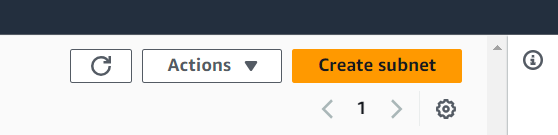
2) Add 4 subnets

2.1) 2 public subnets:

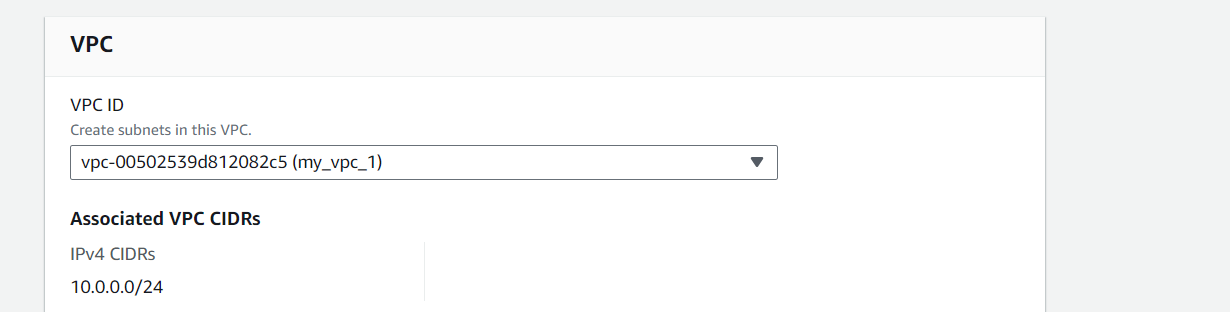
2.1.1) In the VPC Dashboard, click on "Subnets".



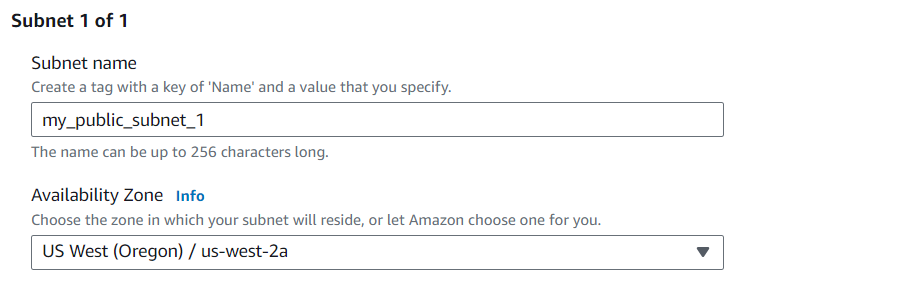
2.1.2) Click **Create subnet**.



2.1.3) Select the VPC you created.



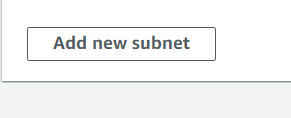
2.1.4) Specify a name and Choose the first Availability Zone

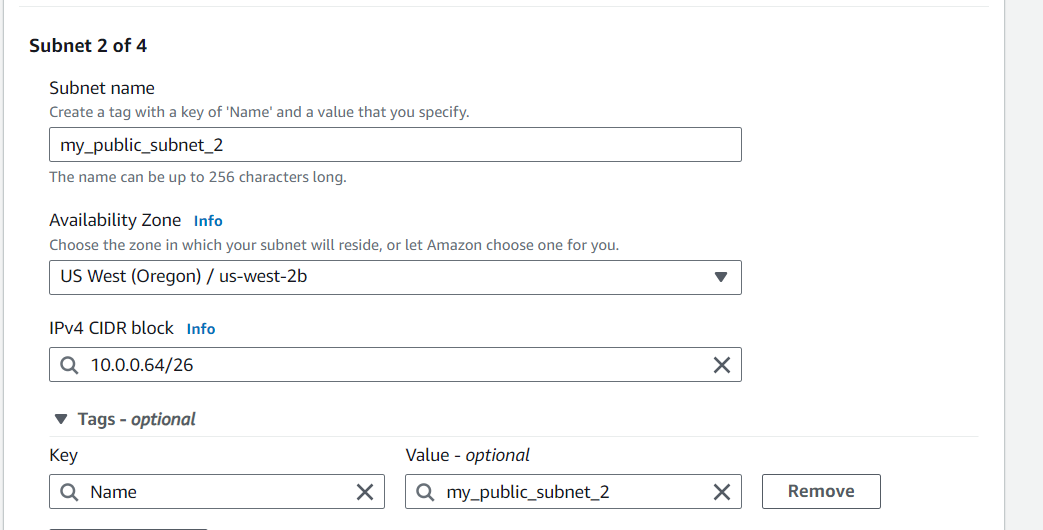


2.1.5) Specify CIDR block for the subnet.



2.1.6) Click add new subnet

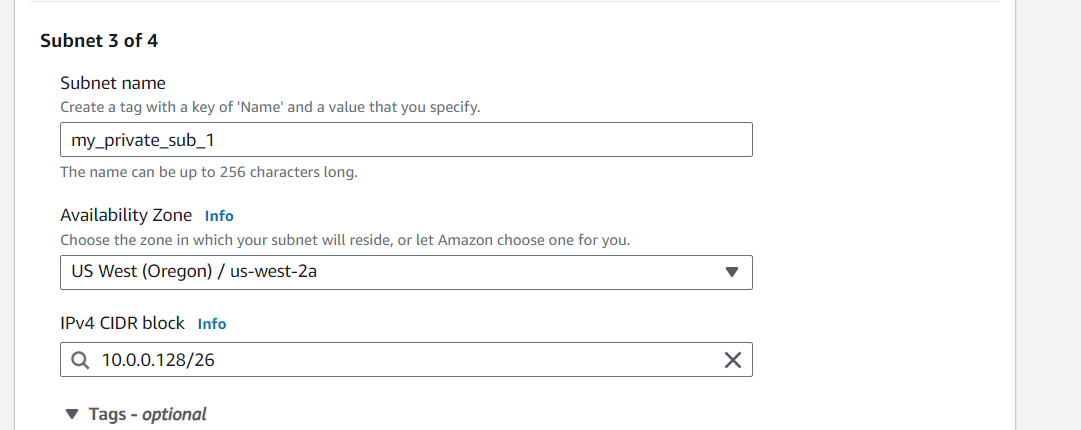


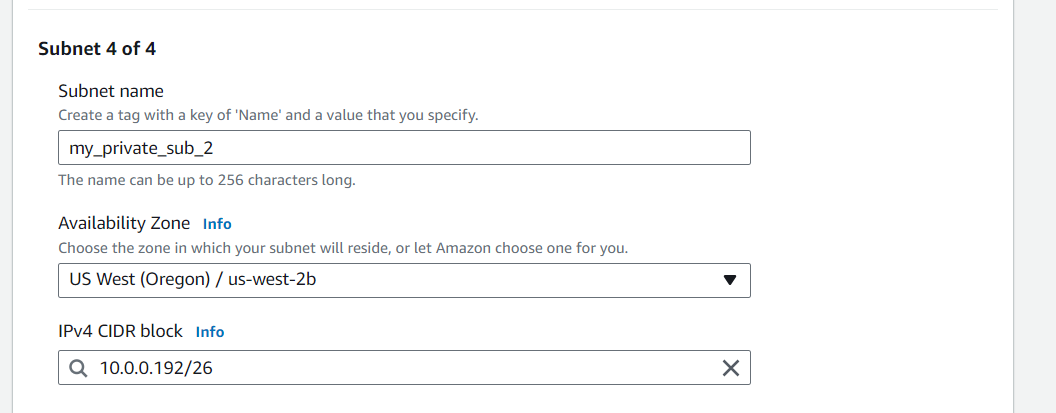
2.1.7) Repeat the process for the second private subnet in a different Availability Zone.  
**\*note the subrouting is done based on the vpc /24 notation chosen, basically we need 4 subnets which mean going done to /26 will do the job and we’ll give each net**  **64 addresses each .0 .64 .128 .192 all of them /26  
**

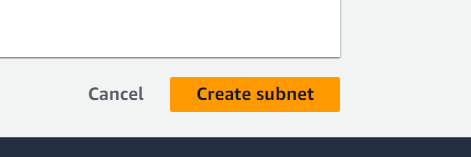
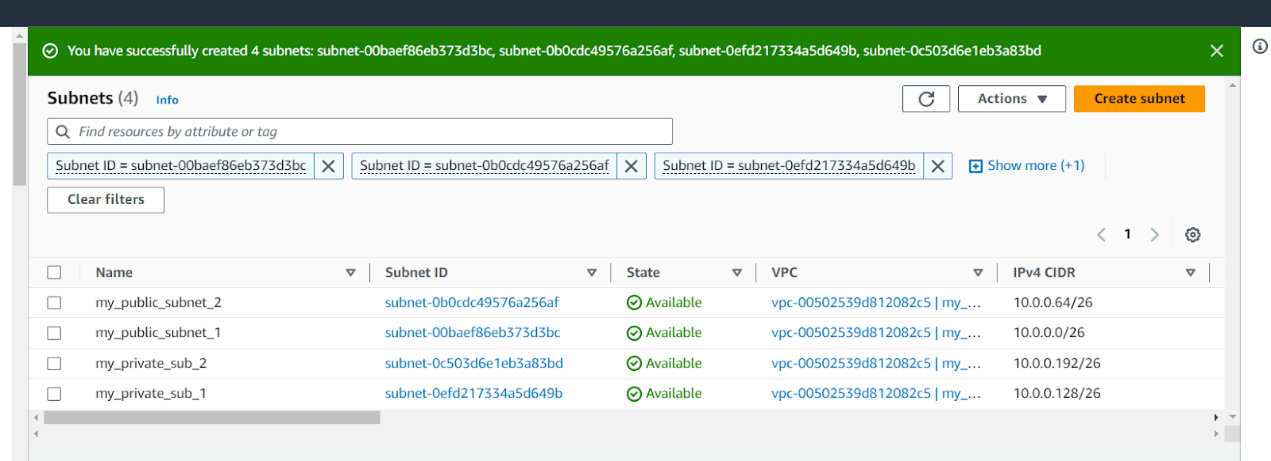
2.2) 2 private subnets:

2.2.1) Repeat the same steps as for the private subnets but choose different CIDR blocks.

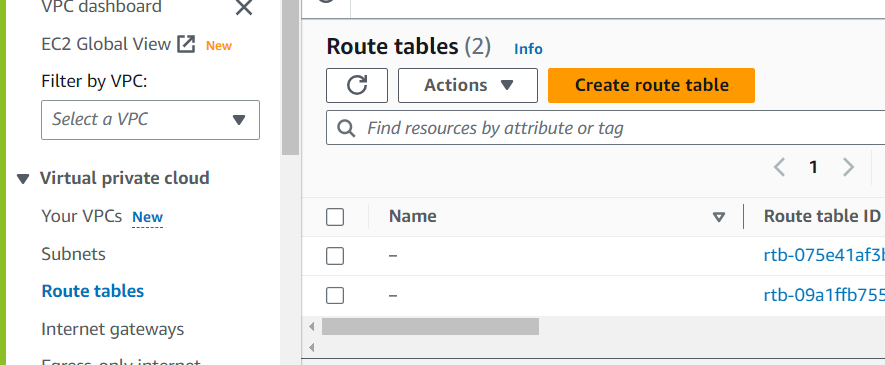
2.2.2) Choose the first Availability Zone for the first private subnet and the second Availability Zone for the second private subnet.





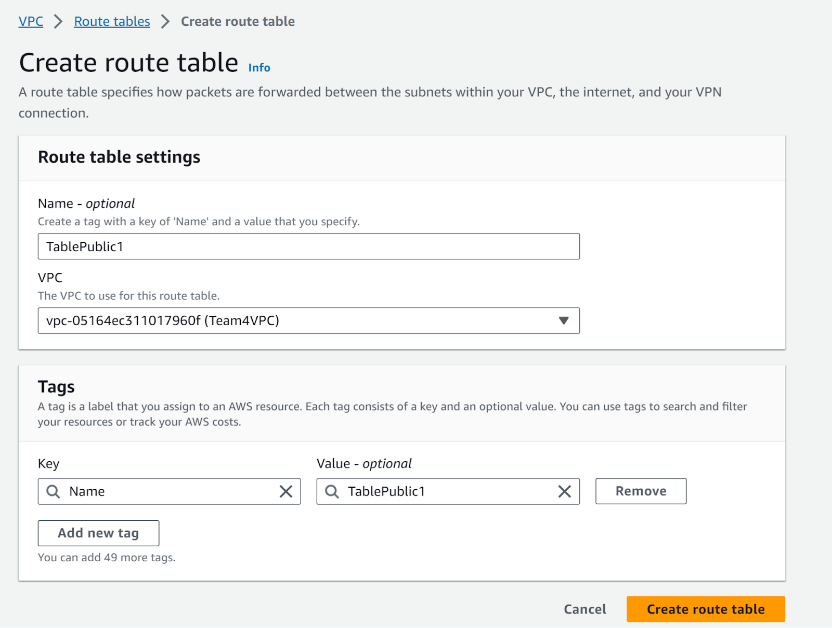
2.3) Click **Create subnet**  
  
This what should happen when finished creating  
  
  
  
3) 2 routing tables:

3.0) Click on the **Route tables left navigation and click on the route tables.**

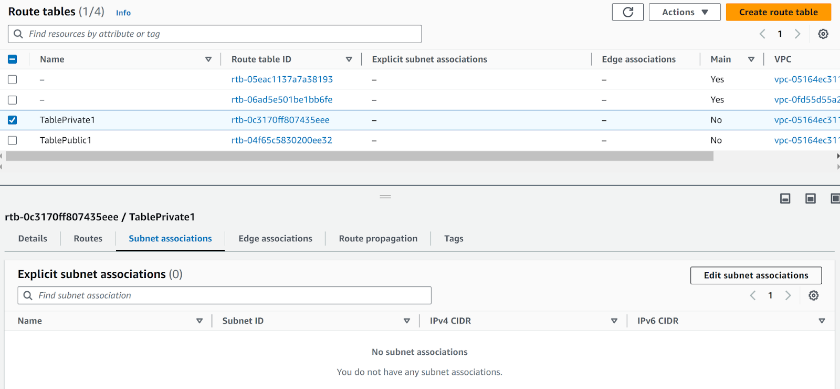
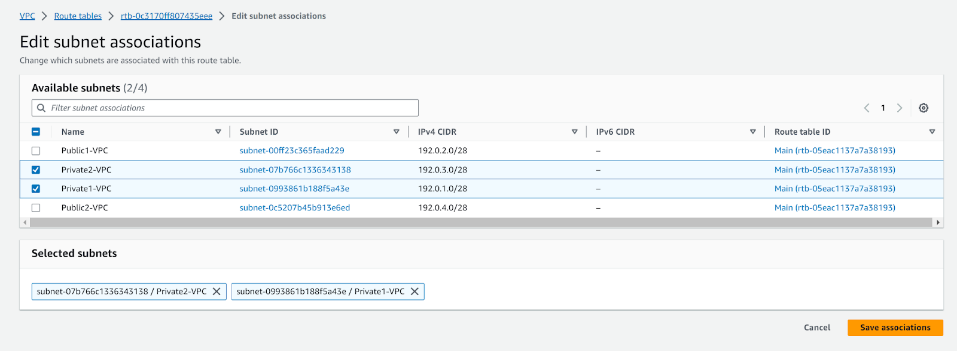


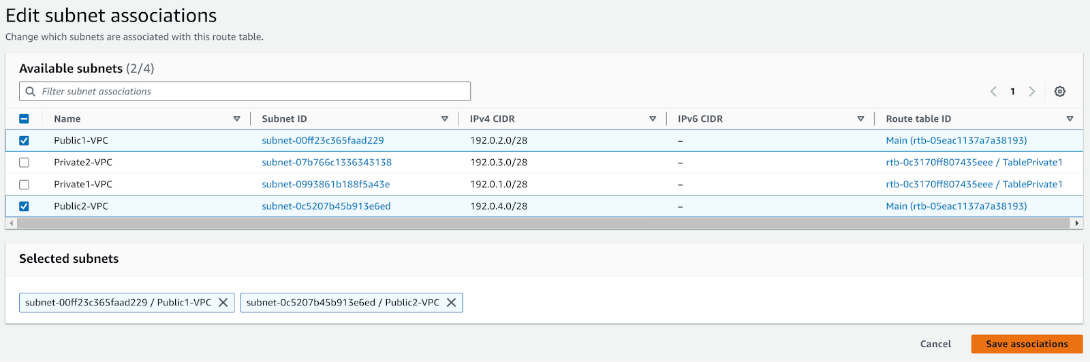
Name your route table. For VPC choose the VPC you’ve created.

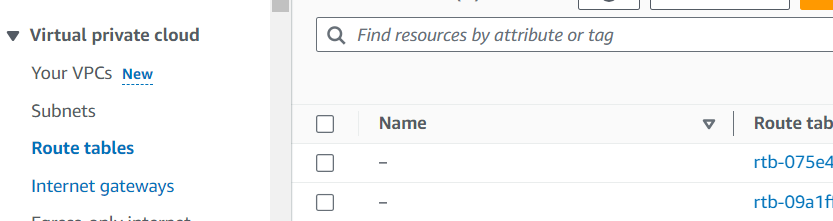
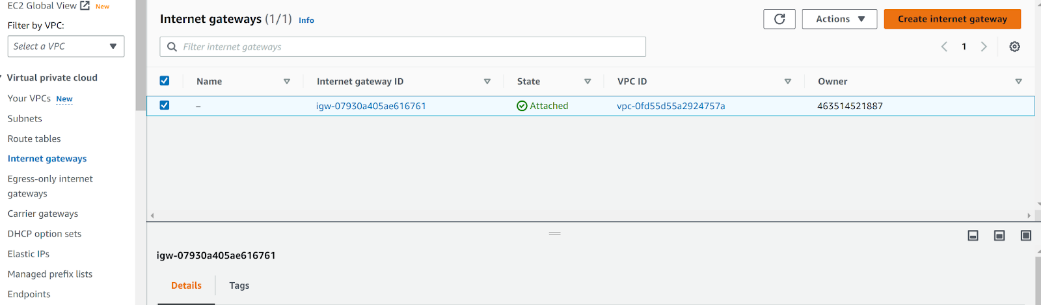
Create 2 route tables, 1 for the public subnets, and one for the private subnets.

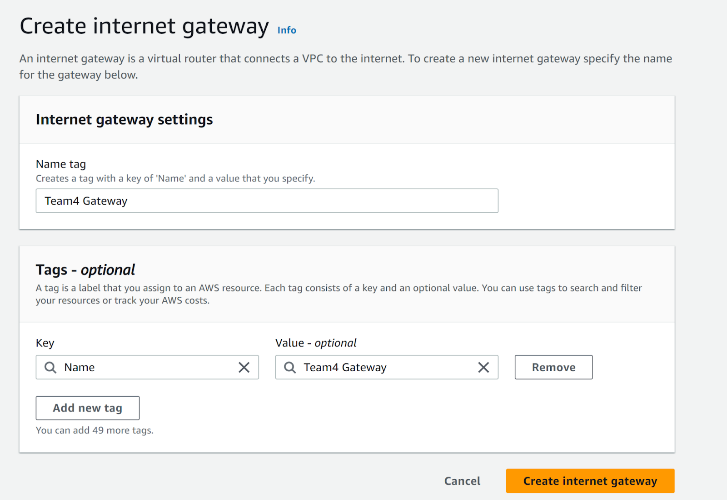
3.1)  

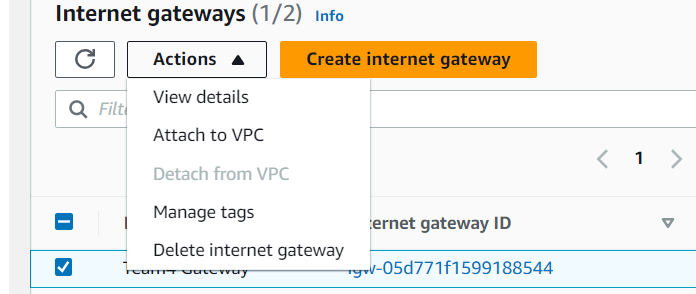
3.2)  Assign the route tables to their respective subnets.

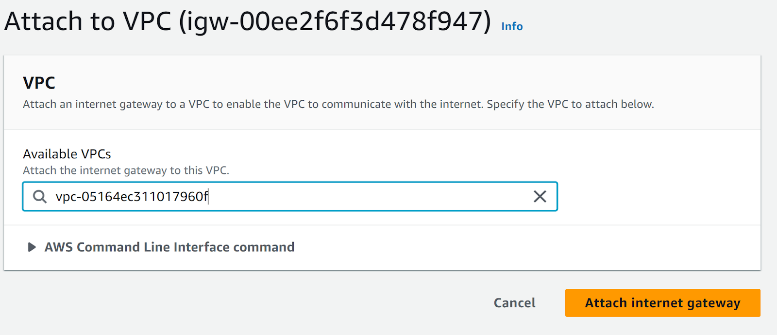
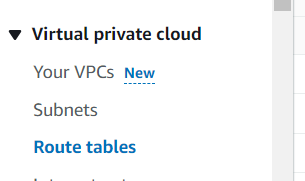
Choose the private subnet and then click on **subnet associations** and then click on **edit subnet association**  
Now click the respective (private) subnets and click **Save associations**

Now do the same thing with the public subnet  


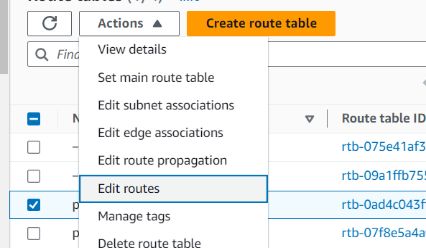
3.3) Create an internet gateway and attach it to the public subnet.  
First click on the **Internet gateways** on the **left nav pane  
**Now Click on the **Create internet gateway**  
  
Choose a name for the gateway

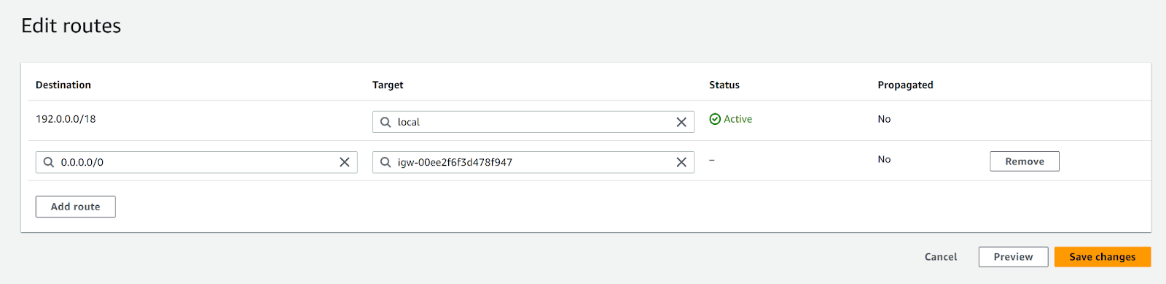


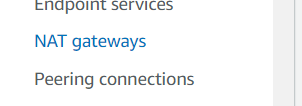
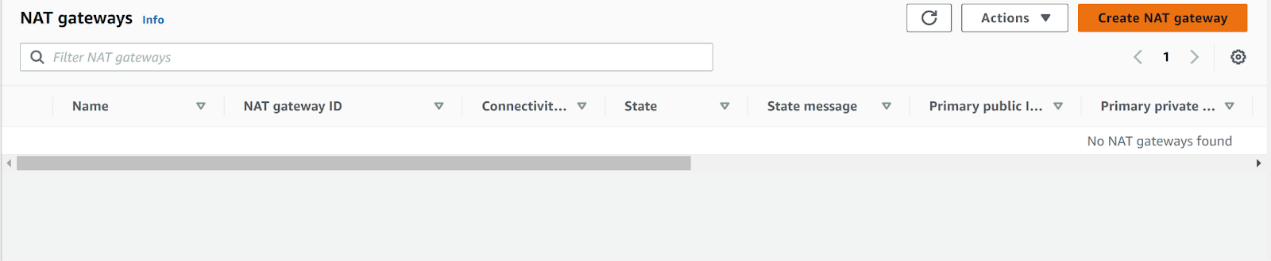
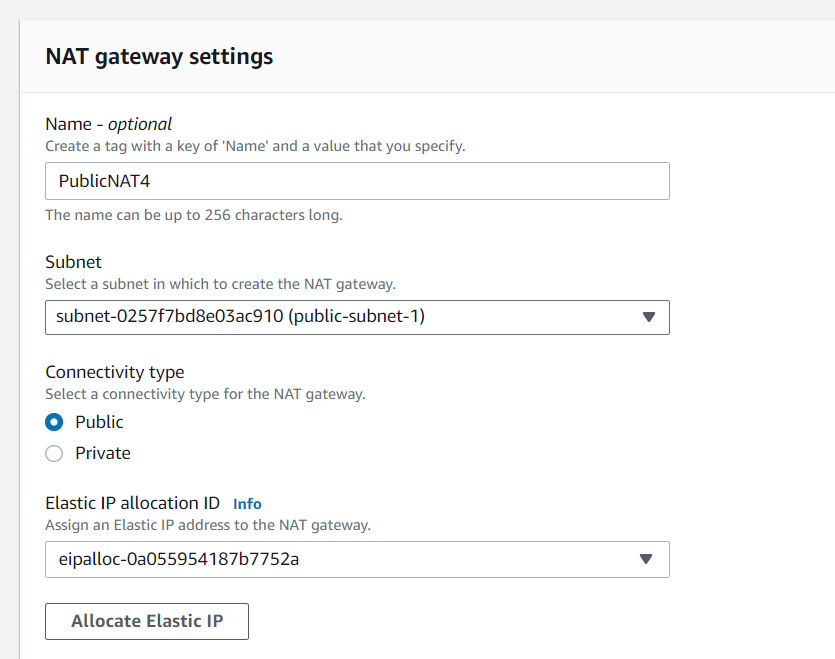
Go back to the **Internet gateways** and choose the new created gateway and then click on the **Attach to VPC**  
  
  
  
3.4) Attach the Gateway to the VPC you’ve created.

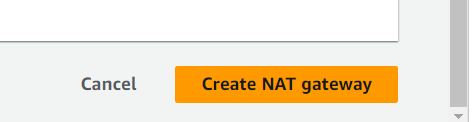
  
Now go back to the **route tables  
**

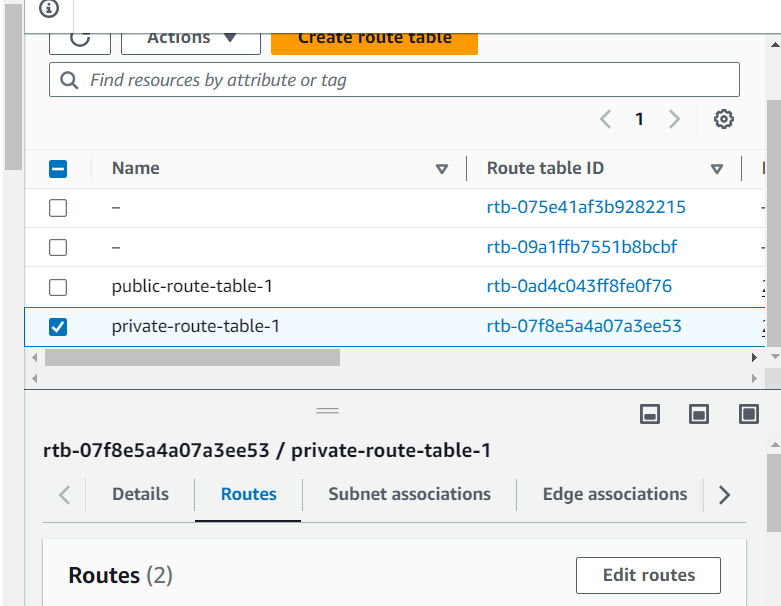
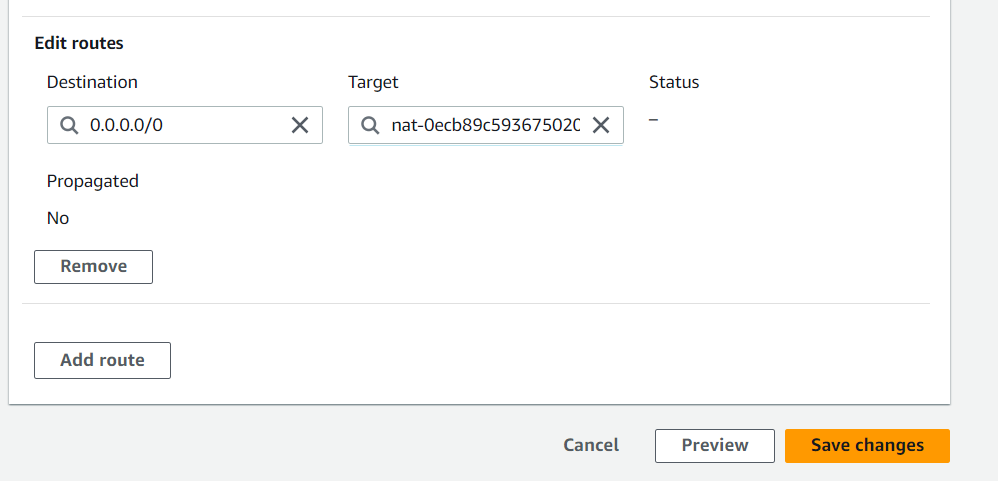
3.5) Attach the Gateway to the public subnet.

Go to the route table, choose the public route table then click on **edit routes** in the **actions**.  
  
Click on **Add route**  
And then insert the 0.0.0.0/0 CIDR notation (all the internet) with choosing the internet gateway made before



3.6) Create a NAT gateway for your private subnet.  
Now go to the **NAT gateways** in the **left navigation pane  
**Click on **Create NAT gateway**   
  
Choose name and relevant public subnet and **Public Connectivity type** and then click on **Allocate Elastic IP  
**

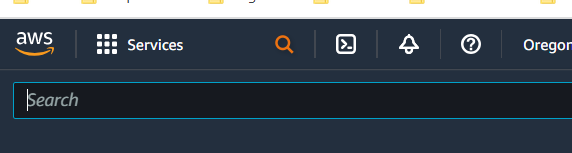
Then click on **Create NAT gateway**  


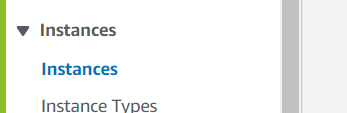
Now go back to the **routes table** and choose the **private subnet   
 **  
Then click on add the route and choose 0.0.0.0/0 Dest and the NATgateway created before  


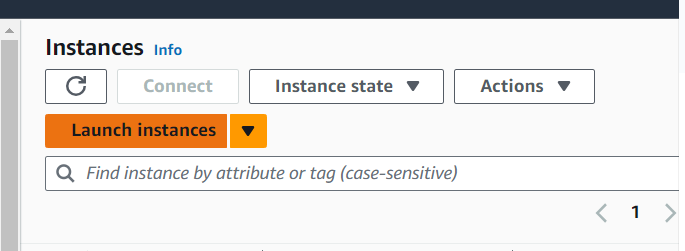
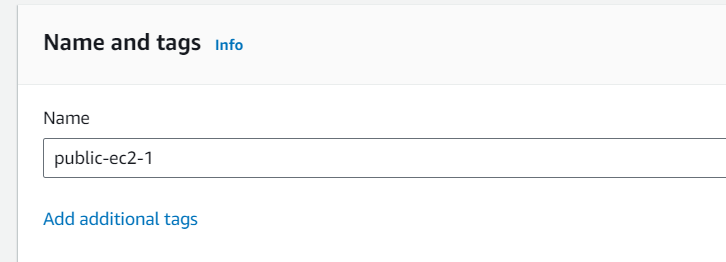
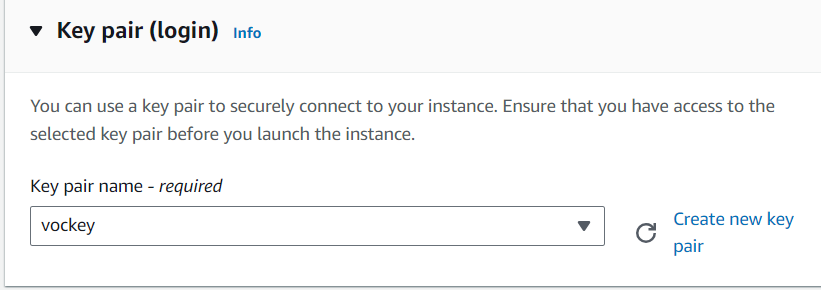
4) Build 2 EC2 instances

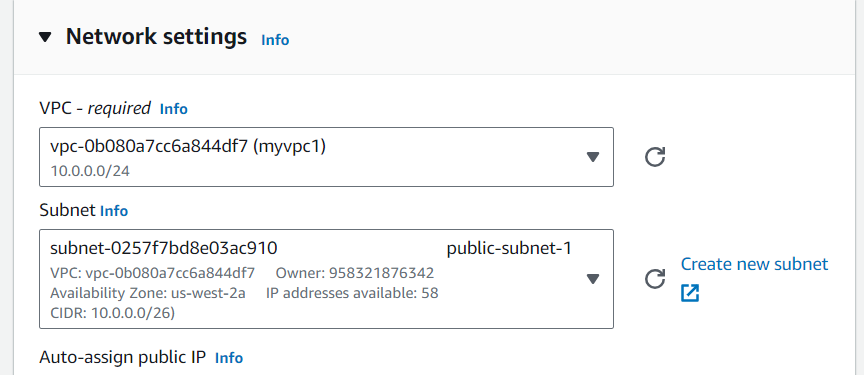
4.1) EC2 (Amazon Linux 2) with only SSH public access

4.1.1) search for EC2 in the services and go to it

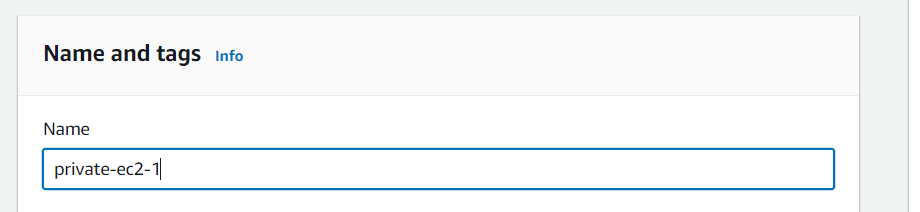


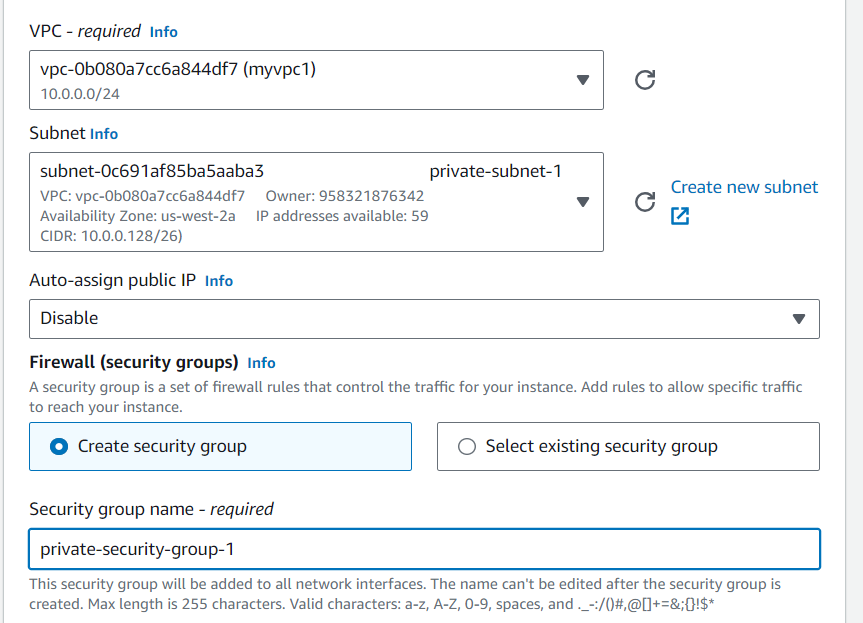
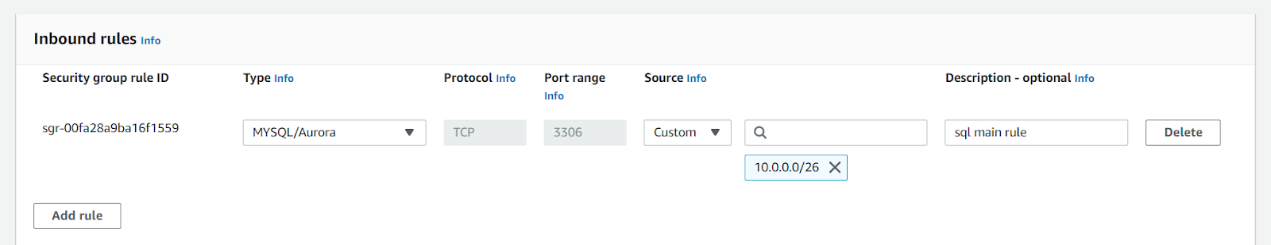
Go to instances in the left pane  
  
  
Click on Launch instances

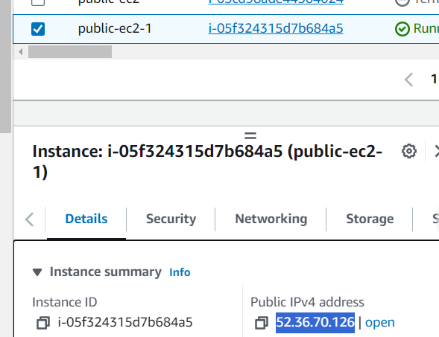
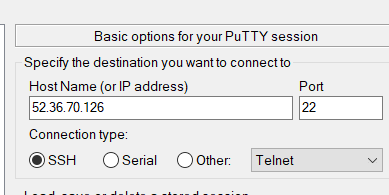
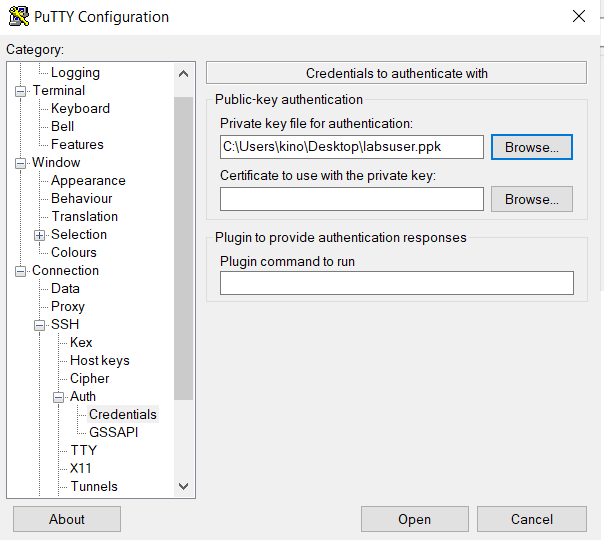
  
Name the instance  
  
Choose default lab key  


**Edit Network Settings** and choose **relevant VPC** and a **Public Subnet**   


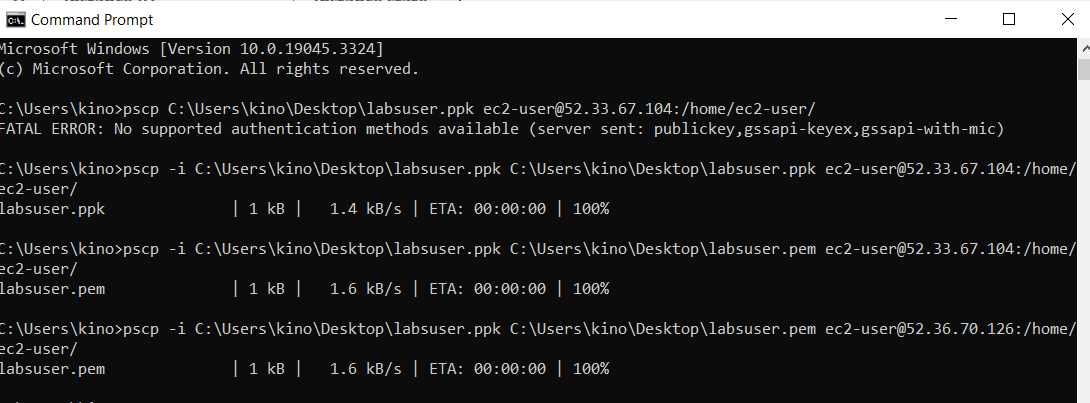
4.2) EC2 (Amazon Linux 2) with MySQL server not accessible publicly

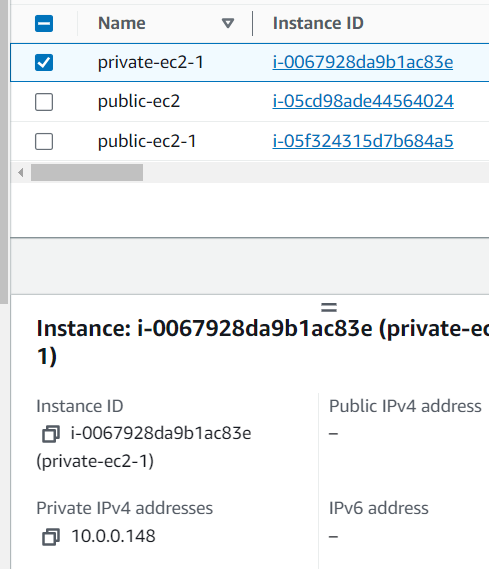
4.2.1) create a new instance and attach it to the same **VPC** and a **private subnet**  
Name it  
  
**Dont forget the key** and then click **Edit Network settings**

Then choose relevant VPC and Private subnet  
  
Configure the security group, name it, choose MYSQL type (port 3306) and the Source as the public subnet that want to use this instance.   


 5) Connect to ec2 instances  
5.1) Connect to the public instance  
5.1.1) in the instances page, choose the public instance you created and copy it’s public IPv4 address  
  
  
5.1.2) run putty and paste the ip   
  
5.1.3) go to   


5.1.4) Then click open and then click Accept  
5.1.5) login as ec2-user and Thats it, you’re in!

  
  
5.2) Connect to the private instance SSH Auth Credentials and locate your private key  
5.2.1)now in order to connect to the private instance you should copy the .pem key to the public instance using the .ppk key and then ssh to the private instance as following:  
5.2.1.a) run the following command from the windows CMD (or any shell)  
**pscp -i C:\path\to\labsuser.ppk C:\path\to\\labsuser.pem ec2-user@public\_instance\_ip:/home/ec2-user/**  


5.2.1.b) now run the following command in order to connect to the private instance from your public one  
  
**chmod 400 path/to/key**  
**ssh -i path/to/key private\_instance\_user@private\_instance\_ip**  
  
Then press enter  
5.2.2) now while you are in the private machine you can install the mysql package as follows:

**\***note that the MySQL is **not available** in the default **Amazon Linux 2** repositories. So we changed to using **Ubuntu based OS**

5.2.3) now run the following one by one to install and start the mysql server  
**sudo apt update**

**sudo apt install mysql-server  
  
Now go back to the public instance and install the mysql client package  
(you can exit the private using exit command or by opening another putty session)**Then you enter **sudo apt install mysql-client-core-8.0  
**