**Cybersecurity 401**

**Module 4 - Cloud Security**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# **Lab 17 - Cloud Network Security**

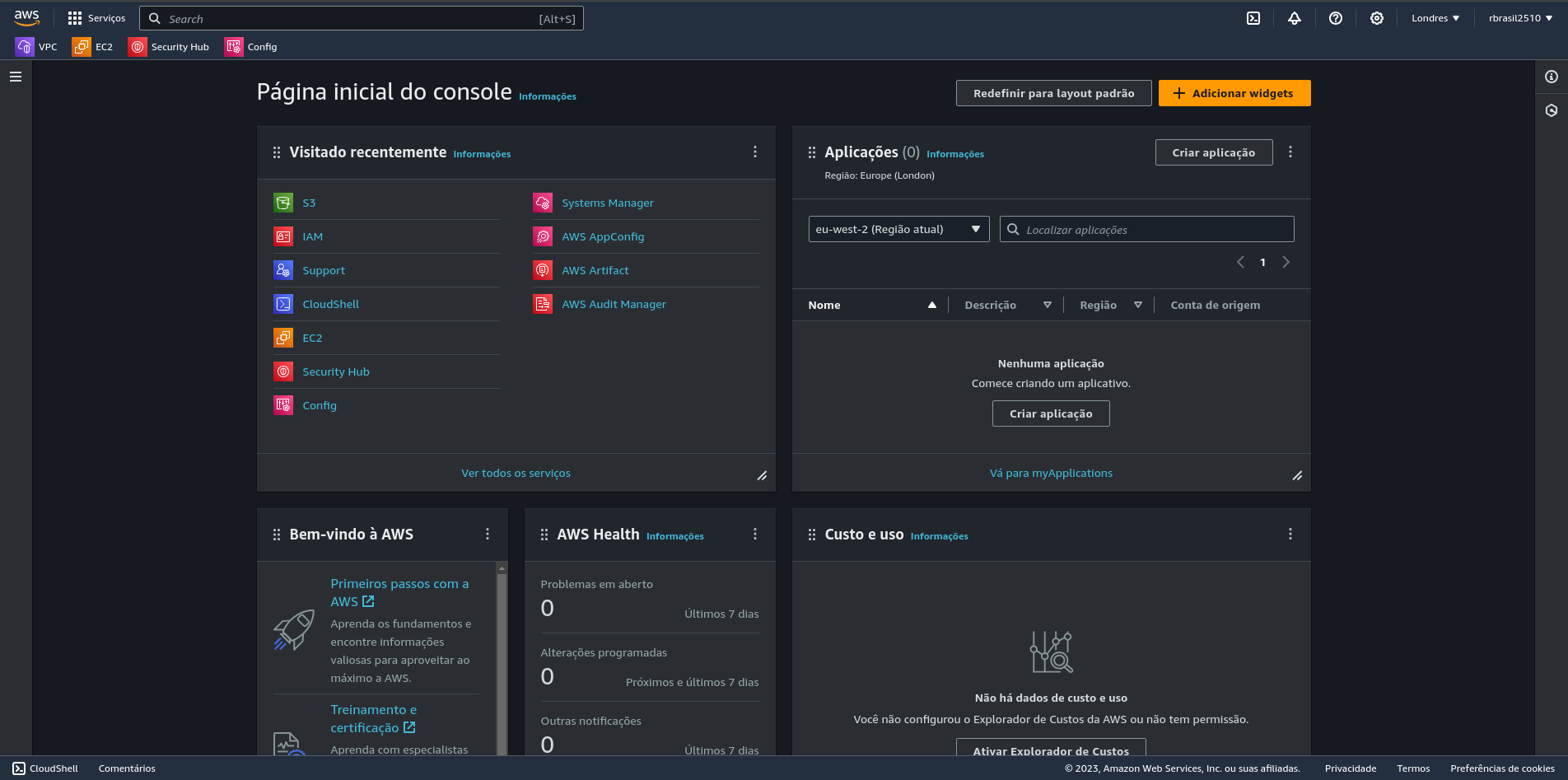
# 

**| Rodrigo Brasil 12/2023 |**

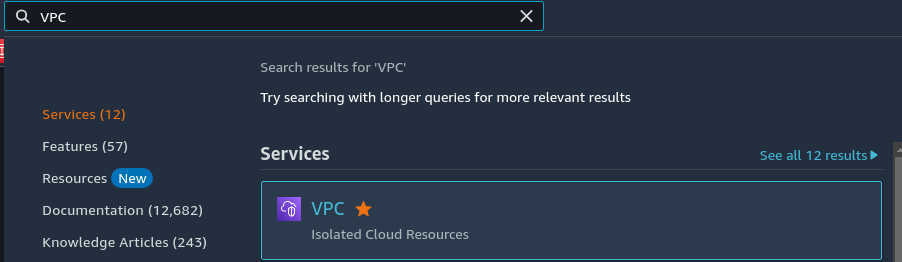
## Part 1: VPC and Subnets Creation

### VPC Creation

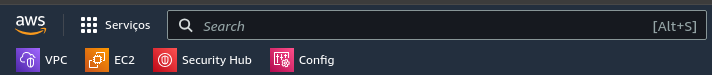
Let’s start with creating a VPC first.

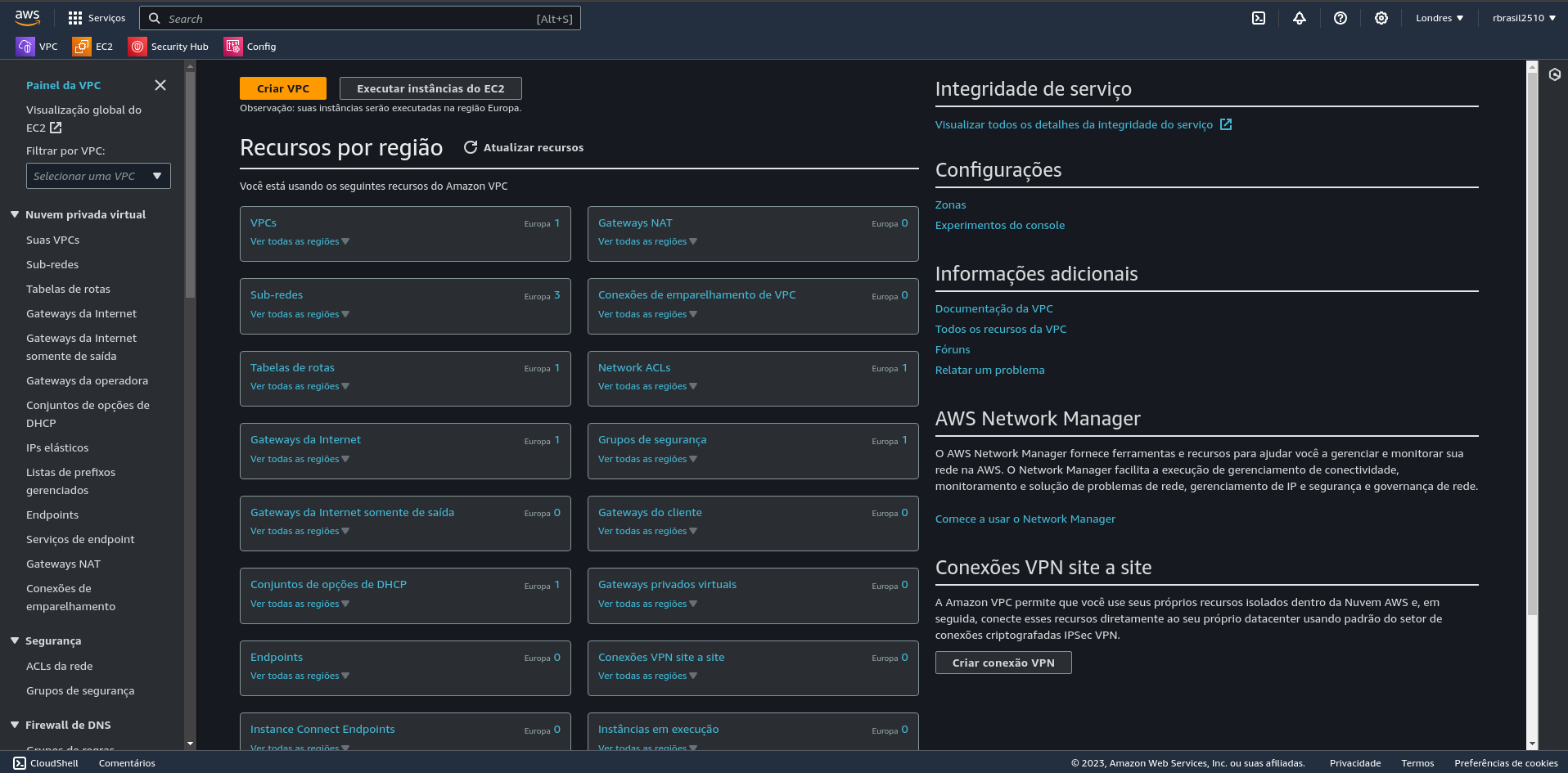


From our AWS dashboard, locate the search bar, it's on the top of the dashboard to the left



On the search bar, search for VPC for easier access later click on the star to add to our favorites

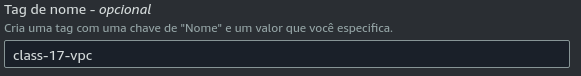
They are located under the search bar



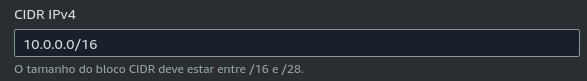
To create a VPC we are going to click on the Orange button saying “**Create VPC**”



Now we will start configuring our VPC, first let's start with naming it.

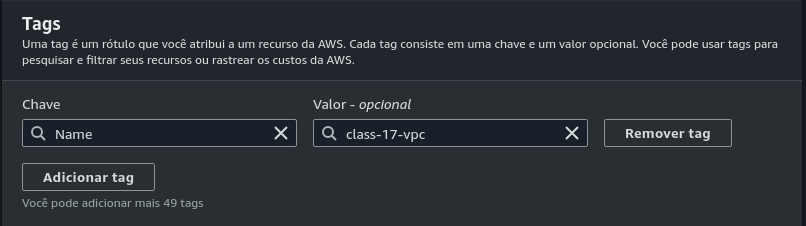


For this exercise im going to call it class-17-vpc

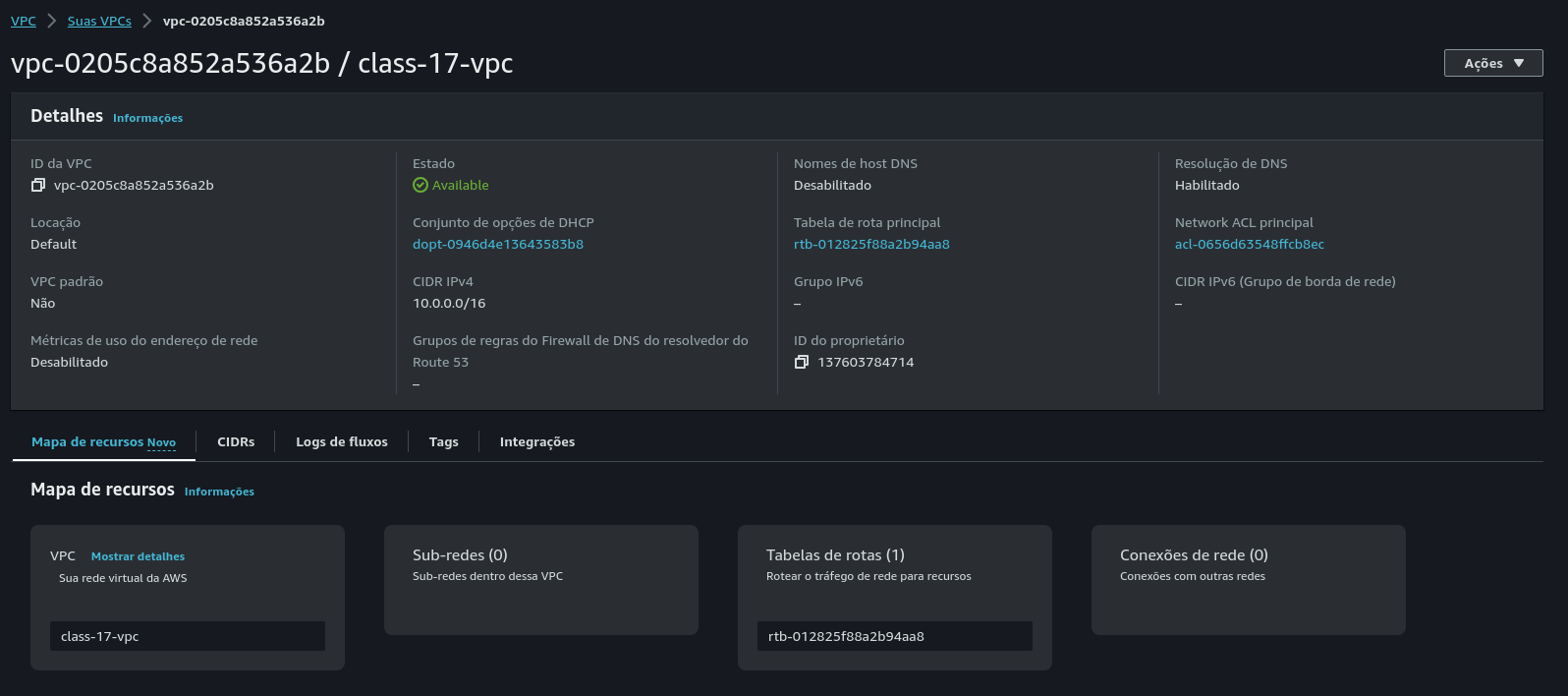


Now we need to give it a CIDR Block

For this exercise we will give it “**10.0.0.0/16**”



Now leave everything on default but make sure the tag is has the same name has the VPC

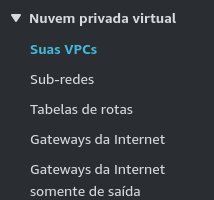


After creating it it will redirect us to its page

### Subnet Creation

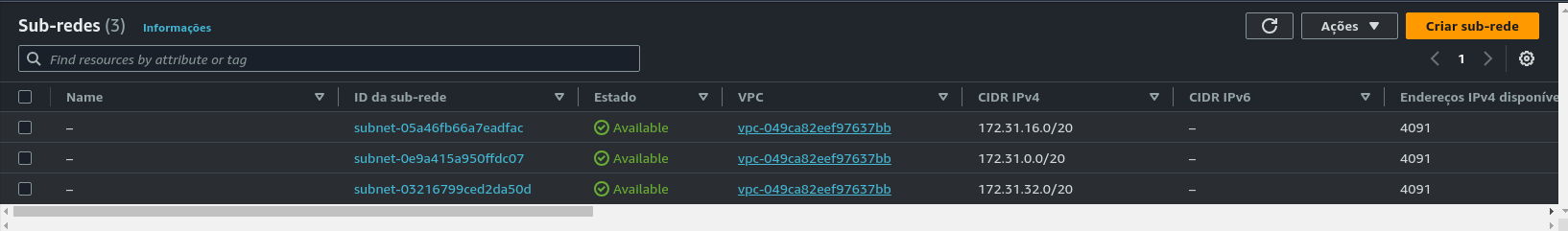
Now we are going to create the subnets

we are going to create 2, a public subnet and a private subnet



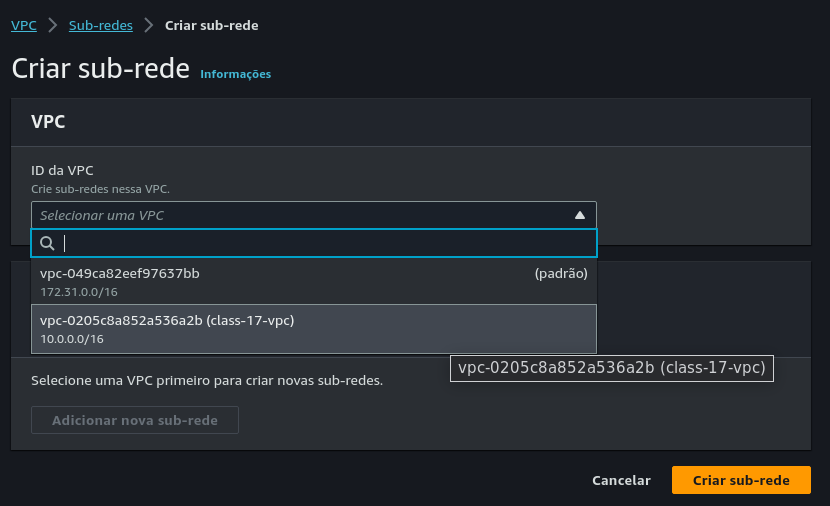
On the VPC page, on the left there will be a collum

in there we will find the subnets



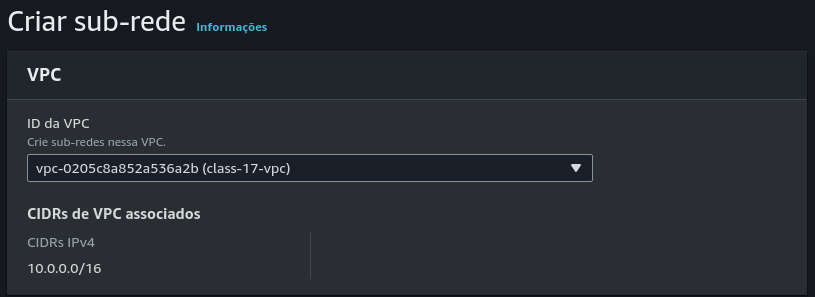
It seems i already have some subnets created, but we will create a new one anyway

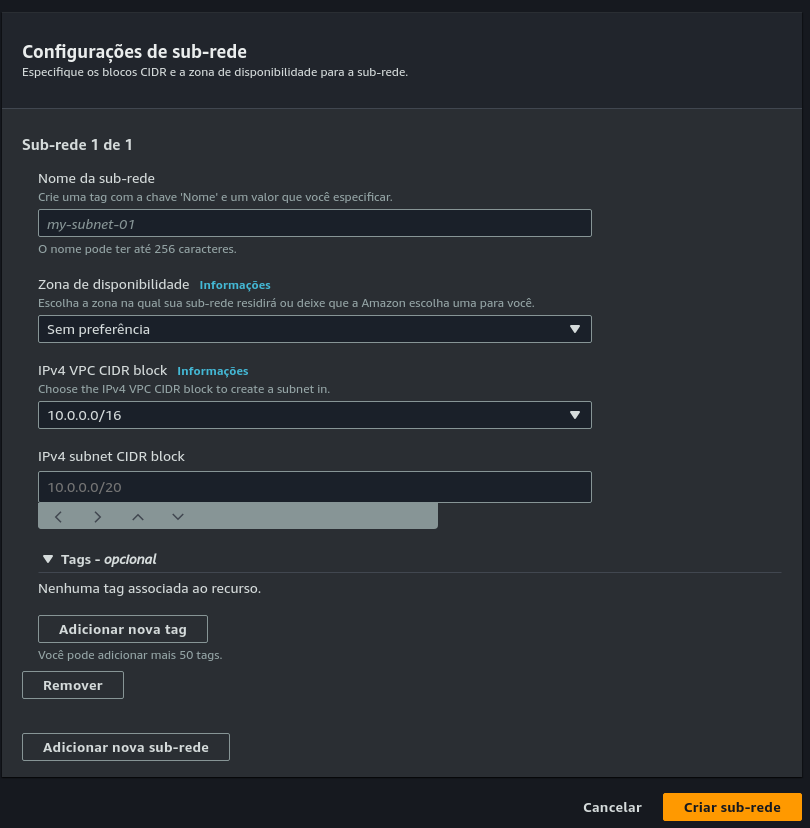
To do so click the orange button “**Create sub-net**”



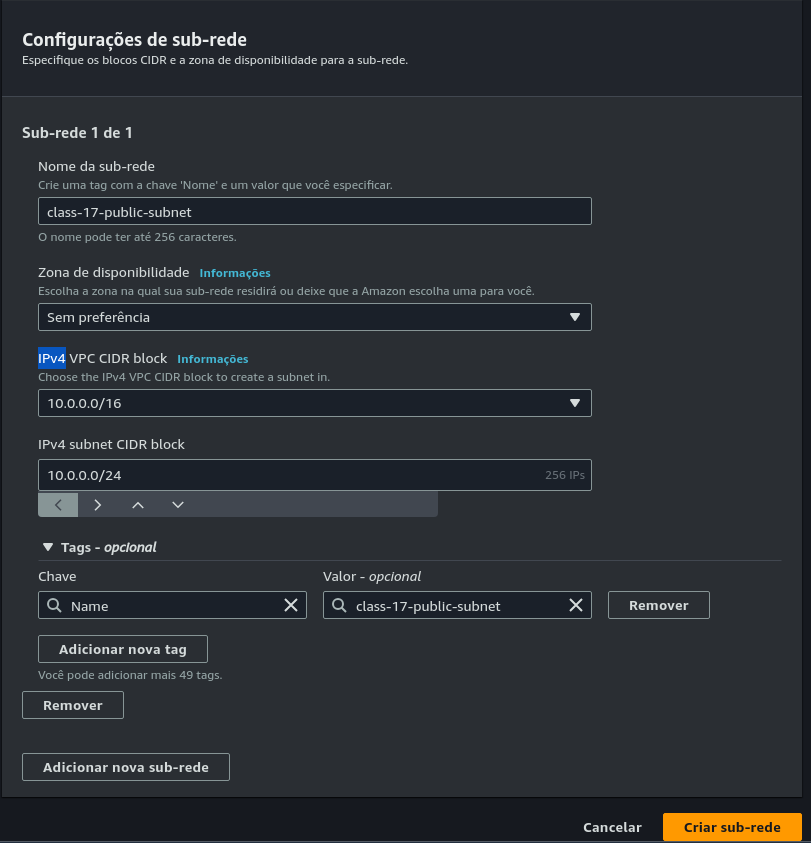
Now let’s configure our subnet

First it will ask what VPC we want to use, we are going to select the one we created on the previous step (**class-17-vpc**)





After selecting the VPC we need to give it a name, we will call it **class-17-public-subnet**



Next we need to give the subnet a CIDR Block, we want 250 hosts so lets calculate this:

VPC CIDR: 10.0.0.0/16 (32 bits total)

Number of bits required for hosts: 8 bits

Subnet size = 32 (total bits) - 8 (bits for hosts) = 24 bits

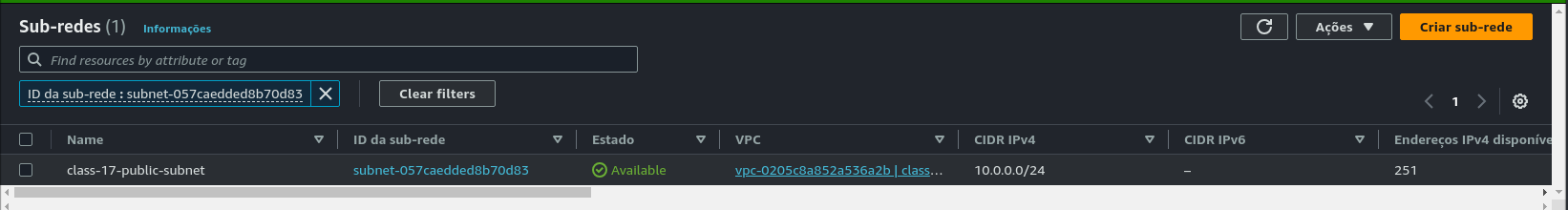
So, our subnet CIDR block would be 10.0.0.0/24.

We can use a subnet calculator for easier calculation

This one is very good:

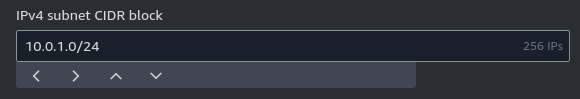
<https://www.davidc.net/sites/default/subnets/subnets.html>

We are going to leave the rest at its default and click the **create subnet** button to create the subnet



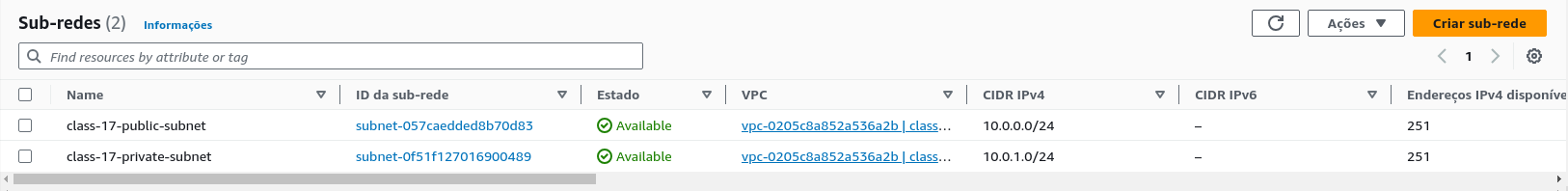
After creating it will redirect us back you our subnets page and it will show our subnet

Now we will create the private subnet



The configuration is the same has the public but on the subnet CIDR Block we are going to use **10.0.1.0/24**

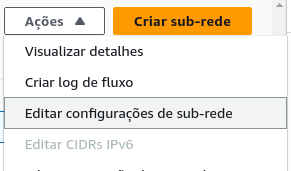
After creating the private subnet



on our subnet page we need to enable auto-assign an IP to our public subnet



to do so we select the public subnet



Click the actions button and click edit subnet configurations



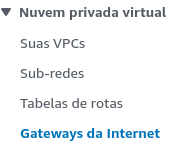
In the configurations, we check the box that says **Enable auto-assign public IPv4 address**

And save

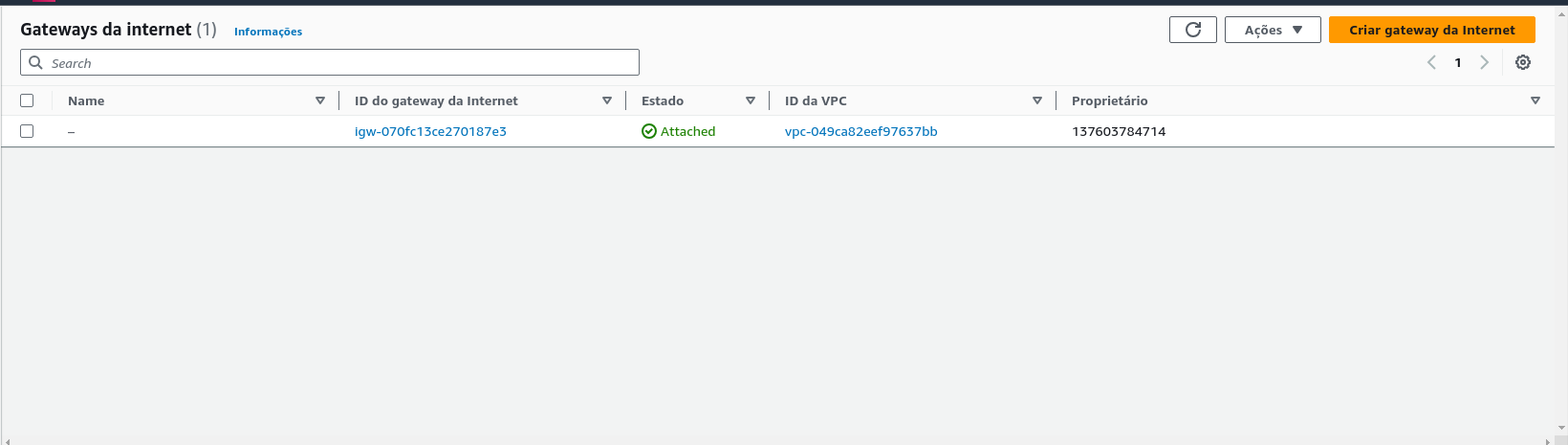
## 

## Part 2: Create an Internet Gateway

Now we are going to create a internet gateway

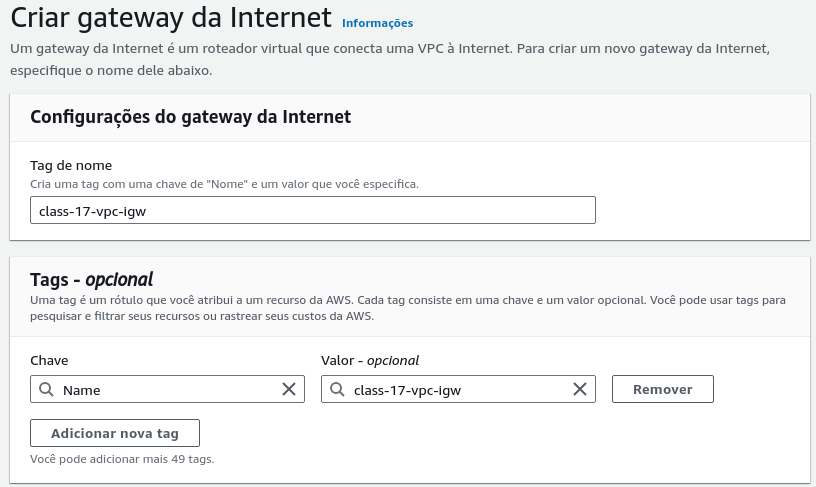


To do so were we left of on the subnets, on the sidebar on the left there will be our internet gateway



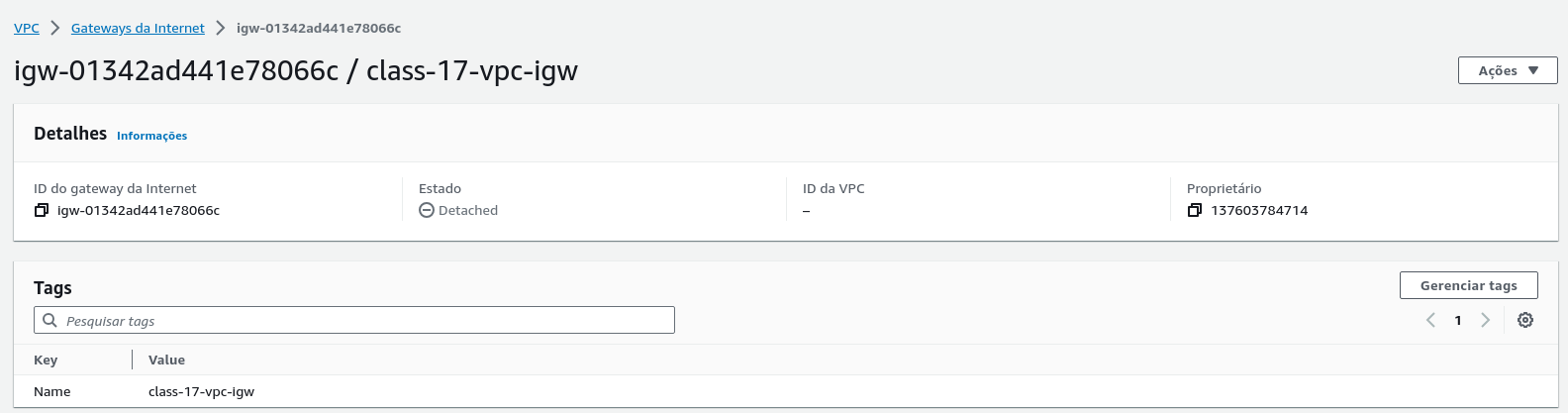
After clicking it will look like this

Click the **Create Internet Gateway** button to create it



This is our creation page. Here we only need to give it a name and its tag

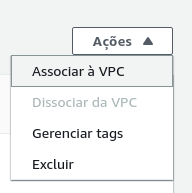
The tag usually is given automatically after giving it a name



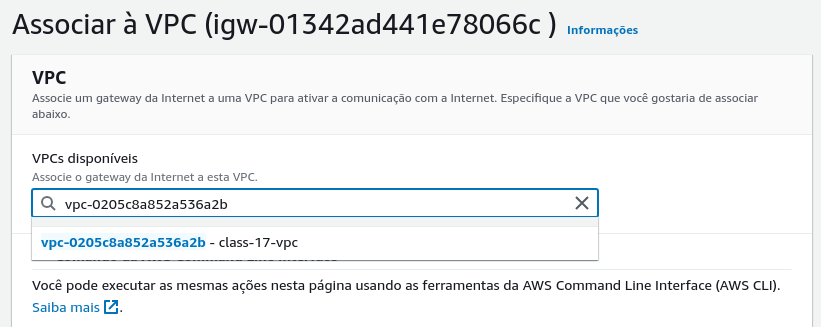
After creating it, it will redirect us to the IGW created

We can observe that the status is **Detached**

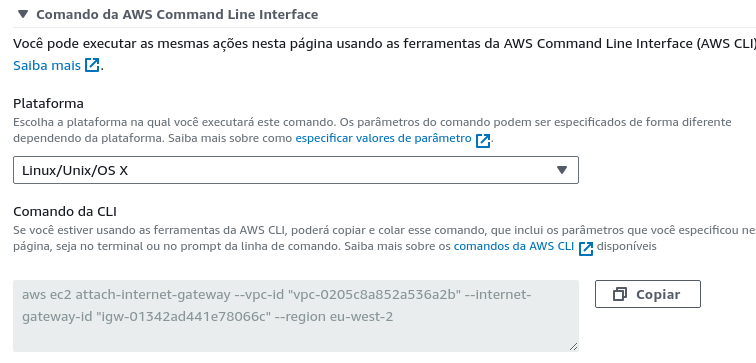
To change this we need to associate a VPC to our gateway



To do so we click on the actions button and select associate to a VPC

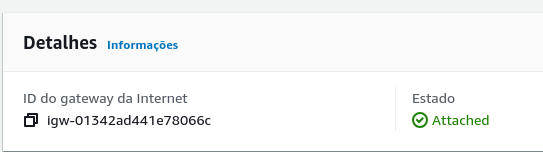


Inside we only need to select our VPC that we created



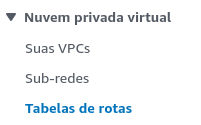
We can also use the AWS CLI in linux to do this with the command:

“**aws ec2 attach-internet-gateway --vpc-id "vpc-0205c8a852a536a2b" --internet-gateway-id "igw-01342ad441e78066c" --region eu-west-2**”

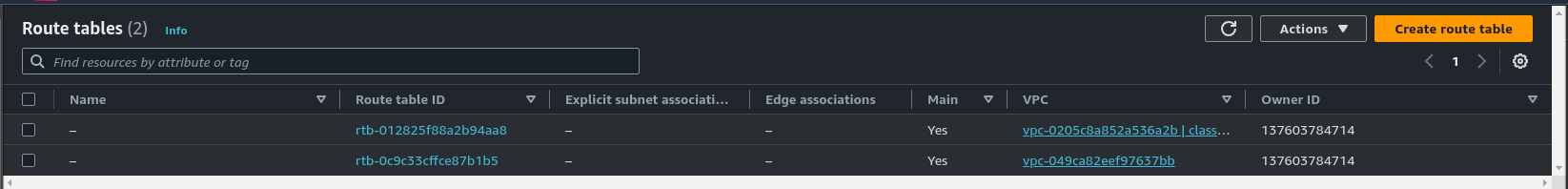


Now we can see the status changed from **Detached** to **Attached**

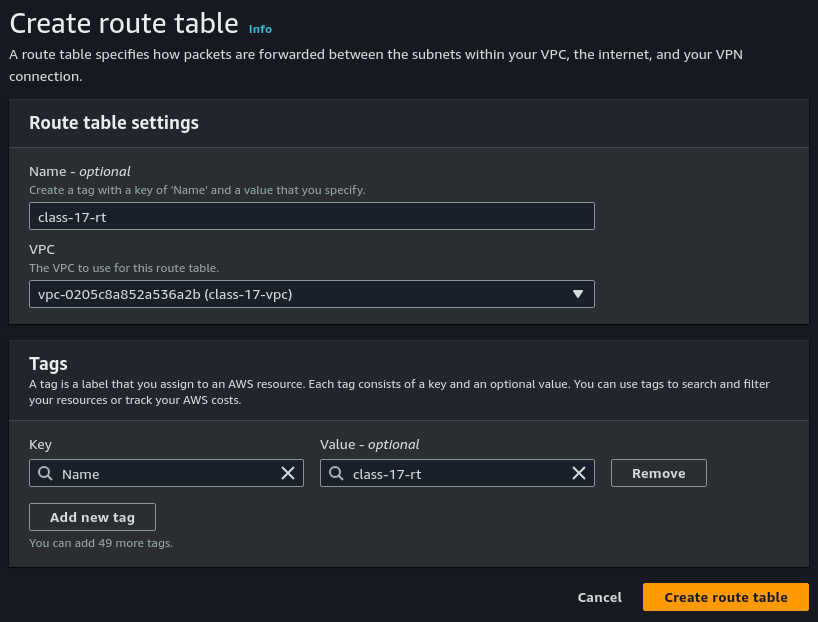
### Configure Route Table



From where we are on our panel to the left select the routing table



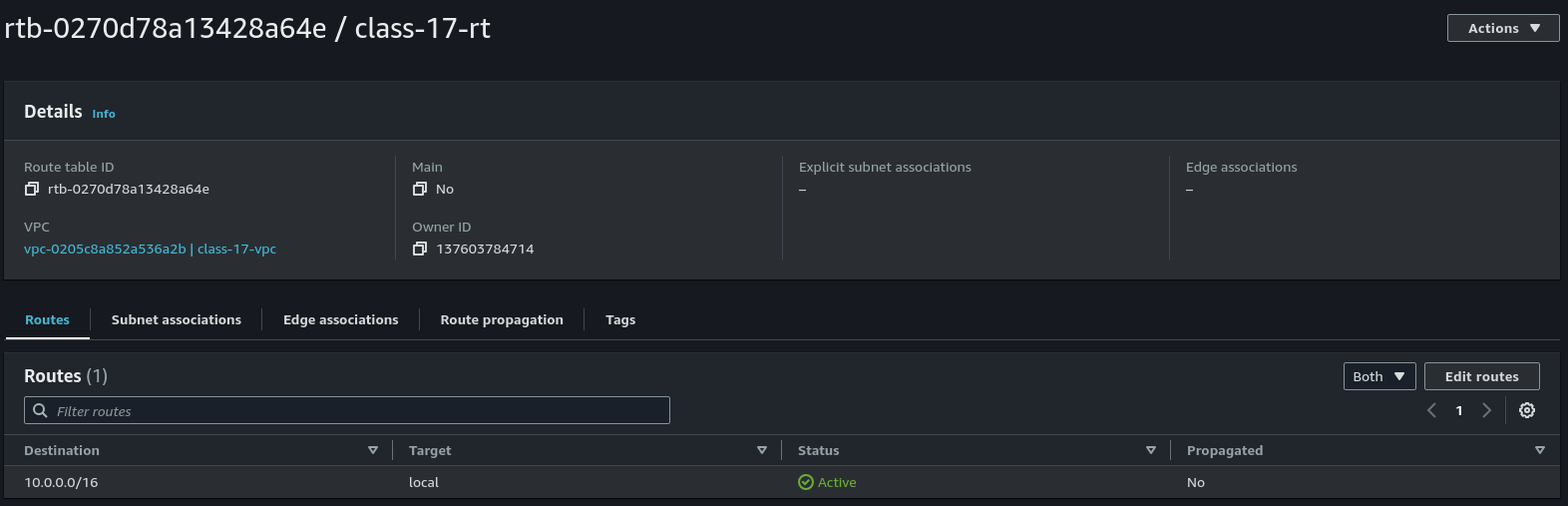
Inside our routing table we will create a new on by clicking on the Create route table



Inside it we only need to give it a name and select our VPC

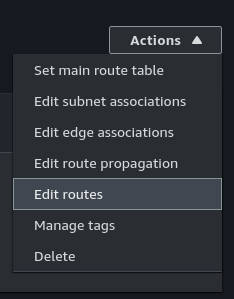
The name i gave for mine is **“class-17-rt”**

And click on the create route table to create it

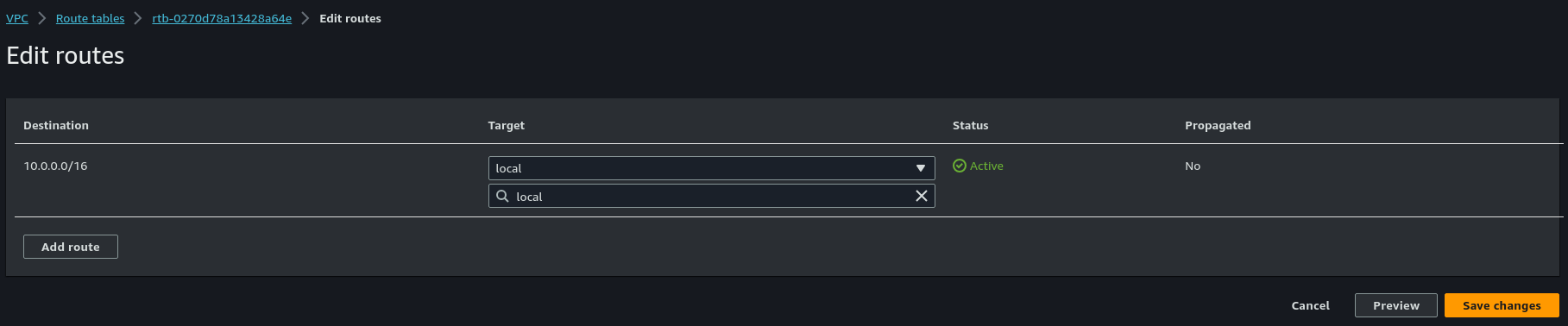


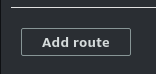
After creating it will show us our routing table

From here

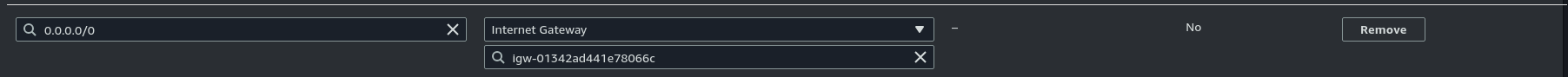


we will click the actions button and click edit routes



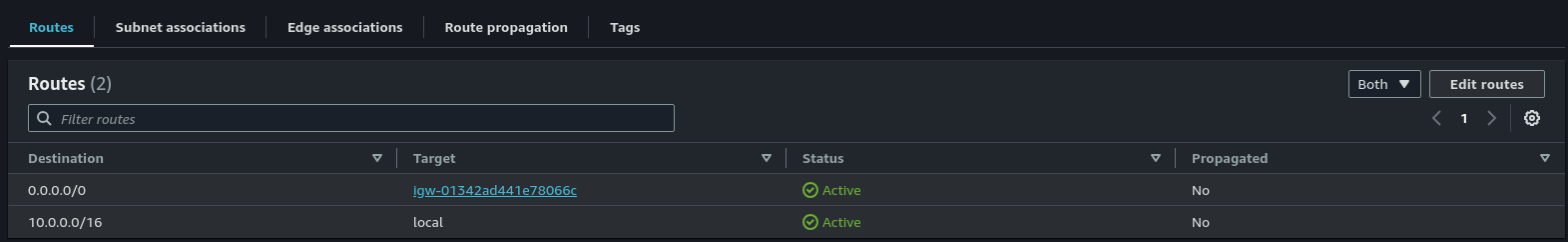


Inside it we will click the add route button

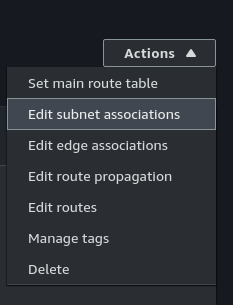


And provide our Destination ip that will be 0.0.0.0/0

And our target will be our previously created internet gateway

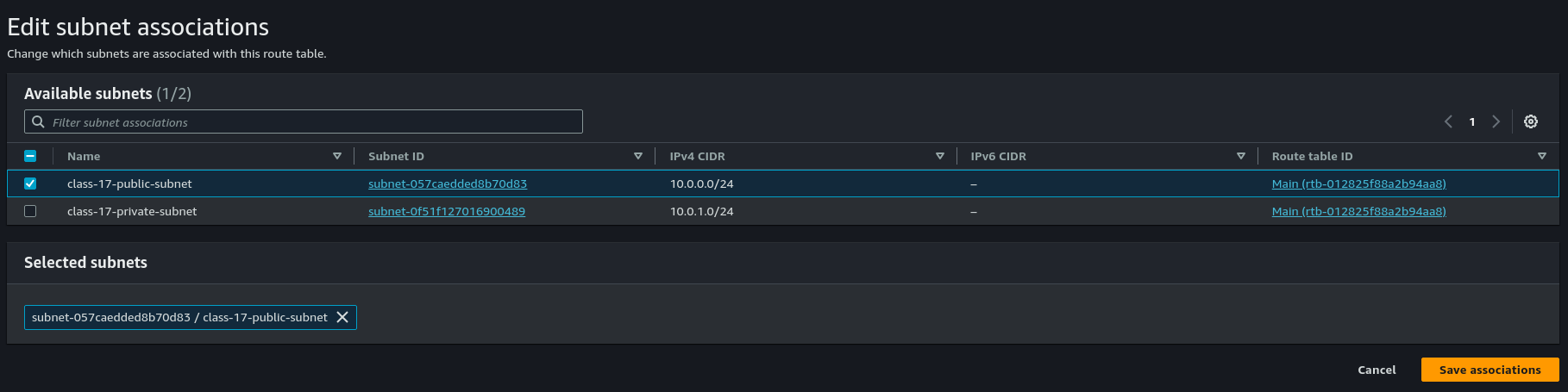


After creating our new route will show up on our routing table



Now we need to associate our public subnet to it

Click actions again and select subnet associations



Inside it we will select our public subnet and save.

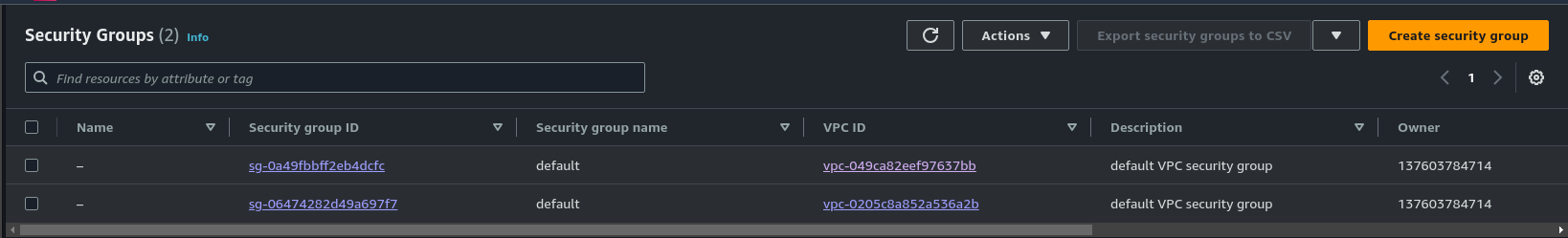
## 

## Part 3: Security Groups

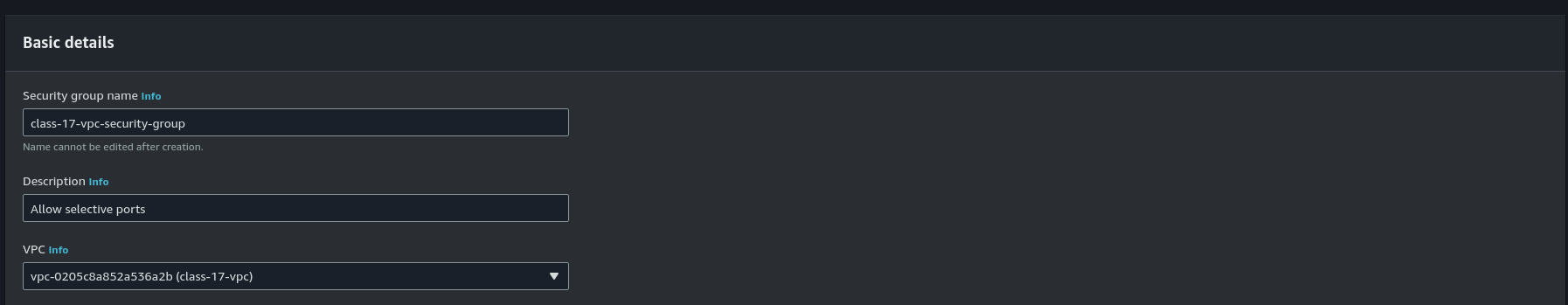
Now we need to create a new security group to associate to our VPC



To do so we will navigate on our left panel, to security and click security groups

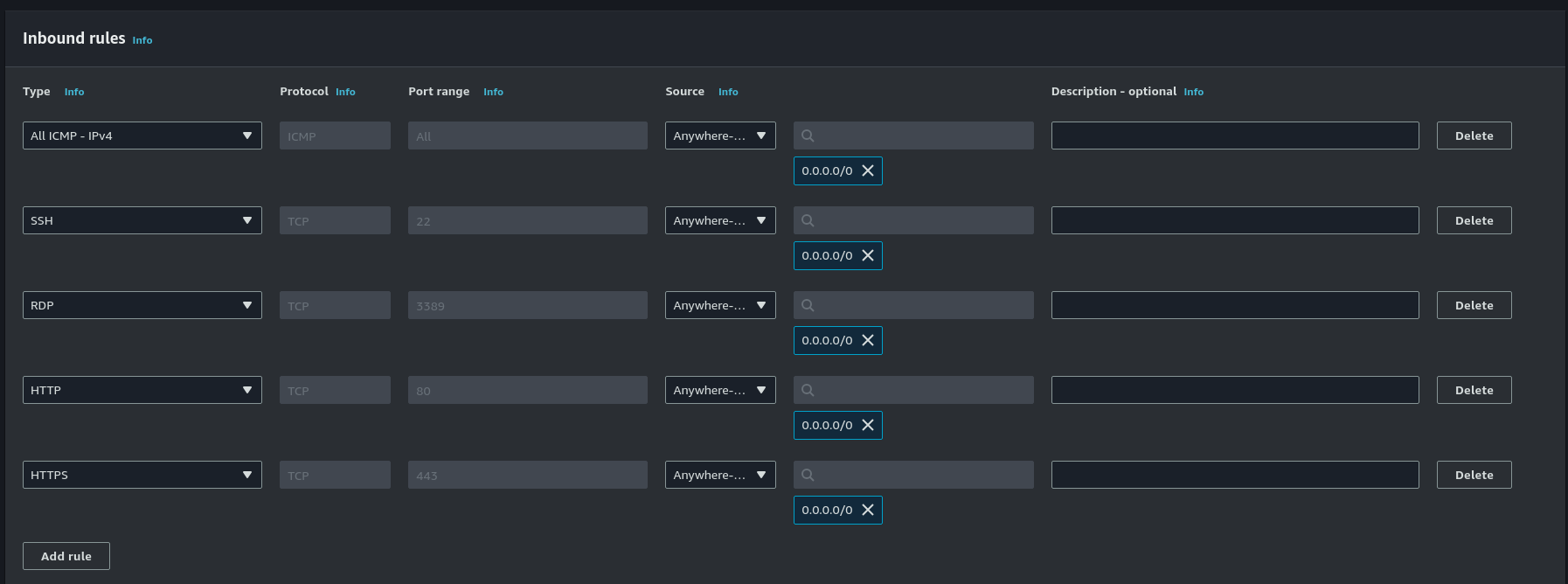


Inside it we will click Create security group



To start, on our basic details we will give it a name, it will be **class-17-vpc-security-group**

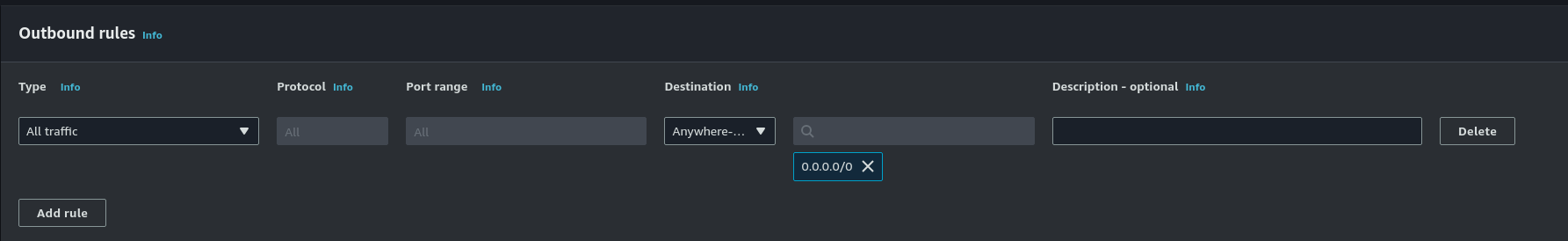
On the description we will say **Allow selective ports** and on VPC we will select our VPC



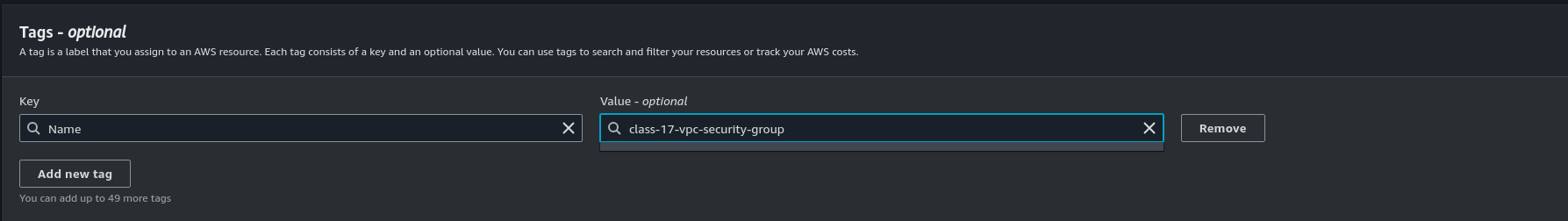
On our inbound rules we will had 5 rules:

1. All ICMP-IPV4
2. SSH
3. RDP
4. HTTP
5. HTTPS

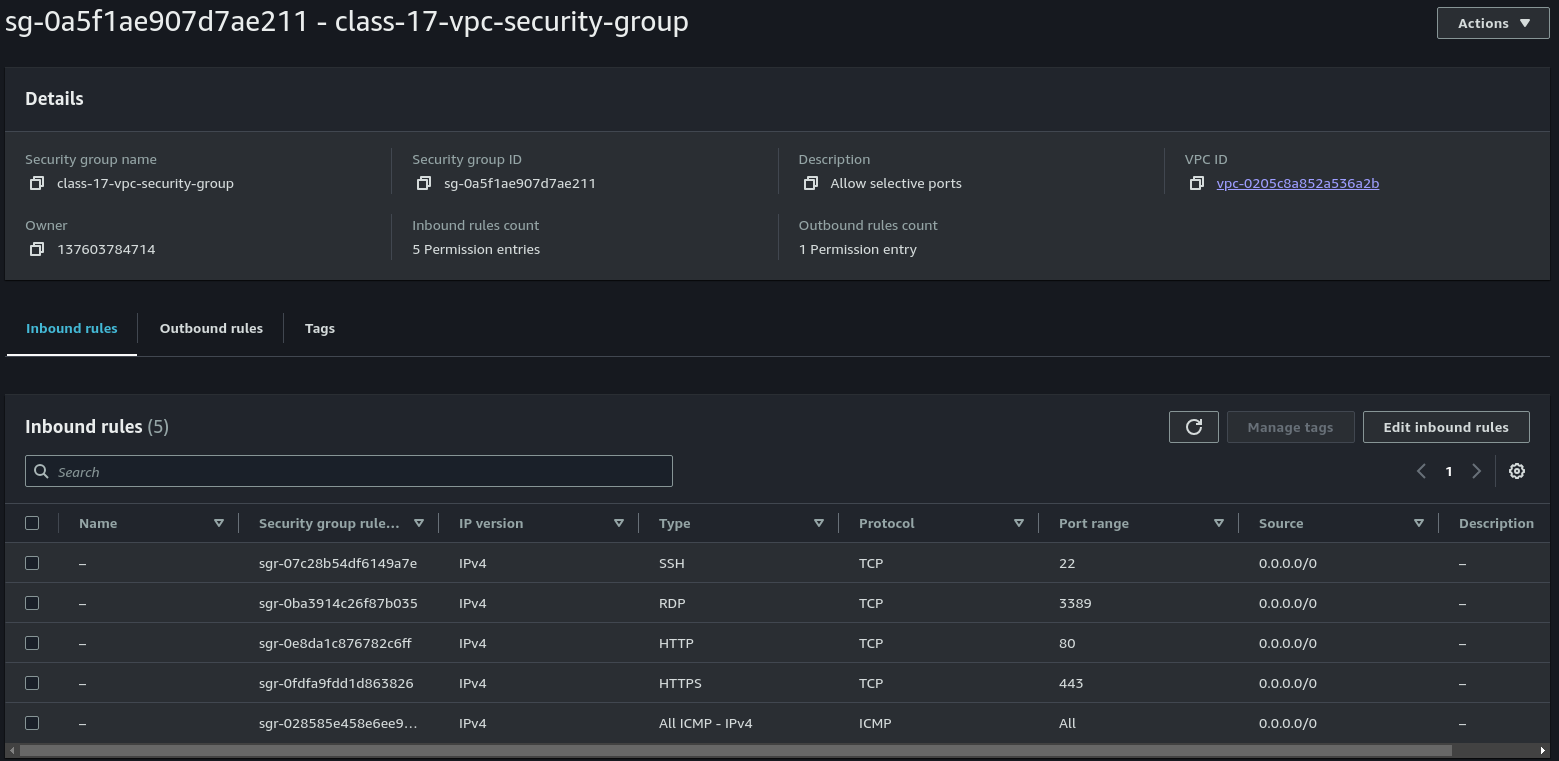
With the source set to anywhere



On the outbound rules leave the type at all traffic and on the destination select anywhere-IPV4



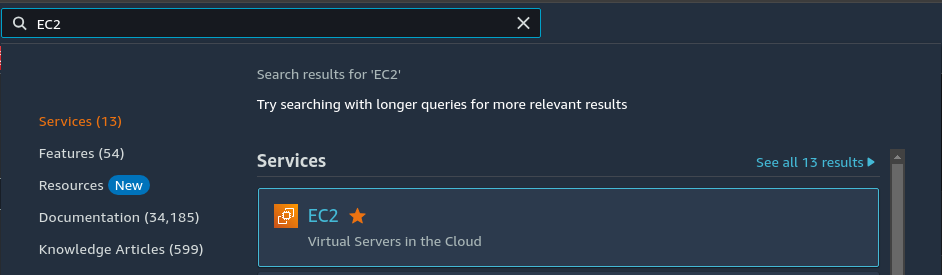
And for last create a tag if there is none, give it a name and a value for this rule it will be **class-17-vpc-security-group**



If successful our security group page should look like this

## Part 4: Instance Deployment

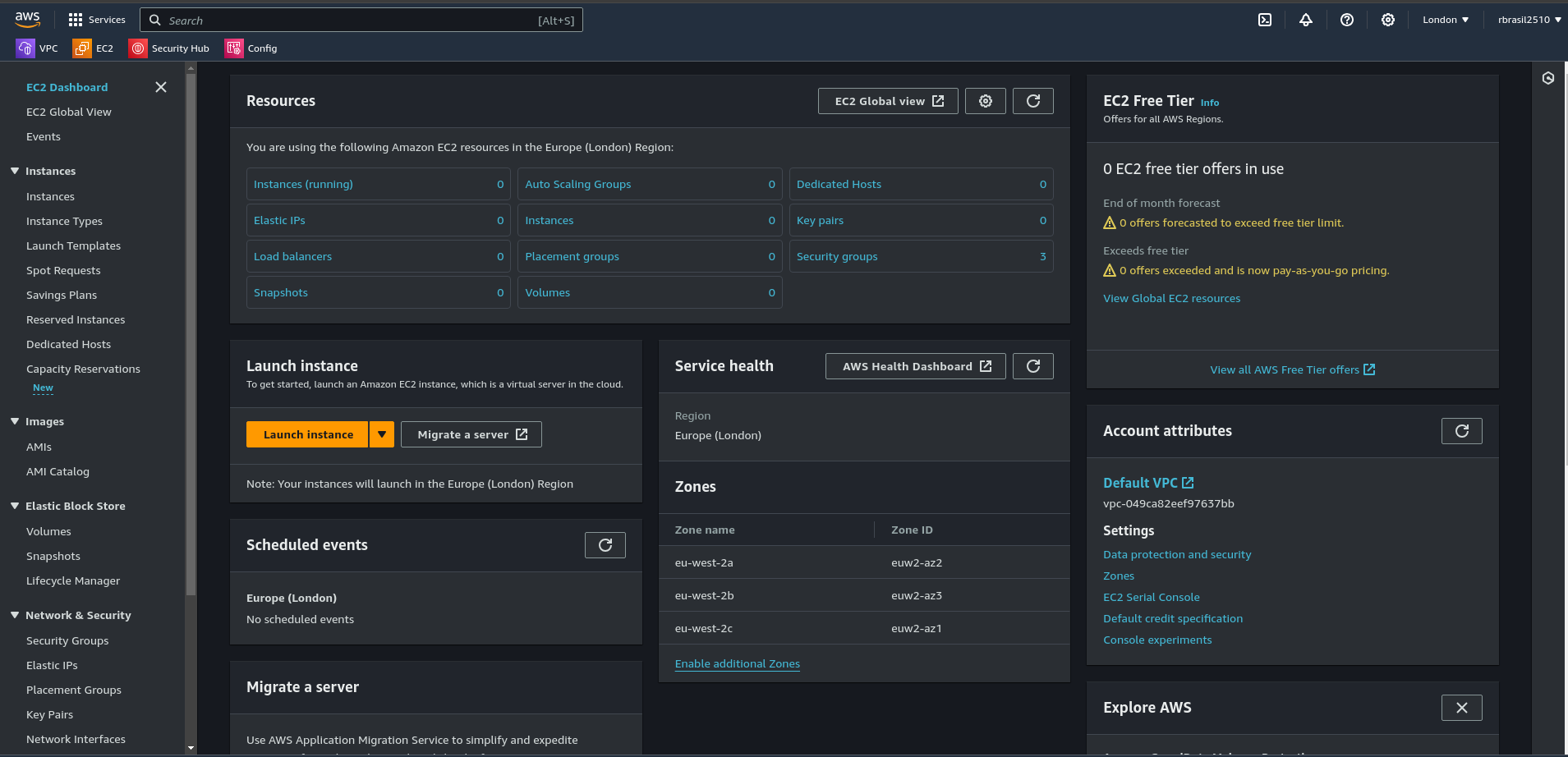
Now we will build 2 Ubuntu Server instances



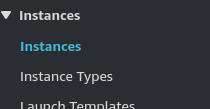
To do this we need to switch you the EC2 service

Search on the search bar located on the top of the page for EC2

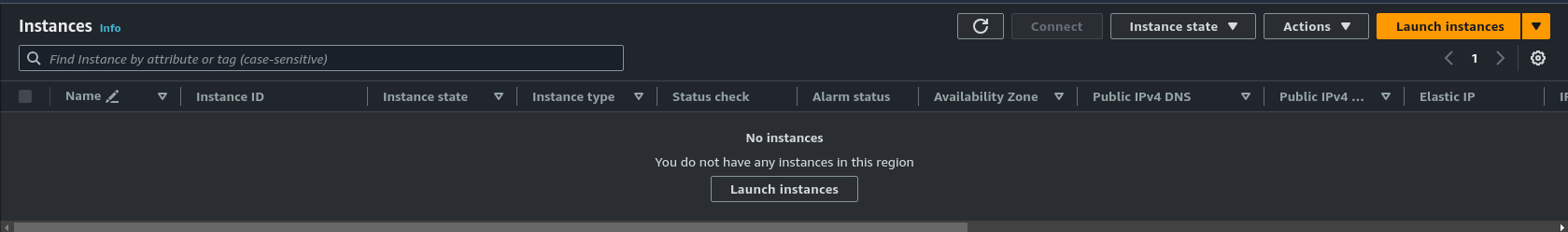
It should be the first match, feel free to save it out the hotbar



The EC2 page should look like this. From here on our left panel

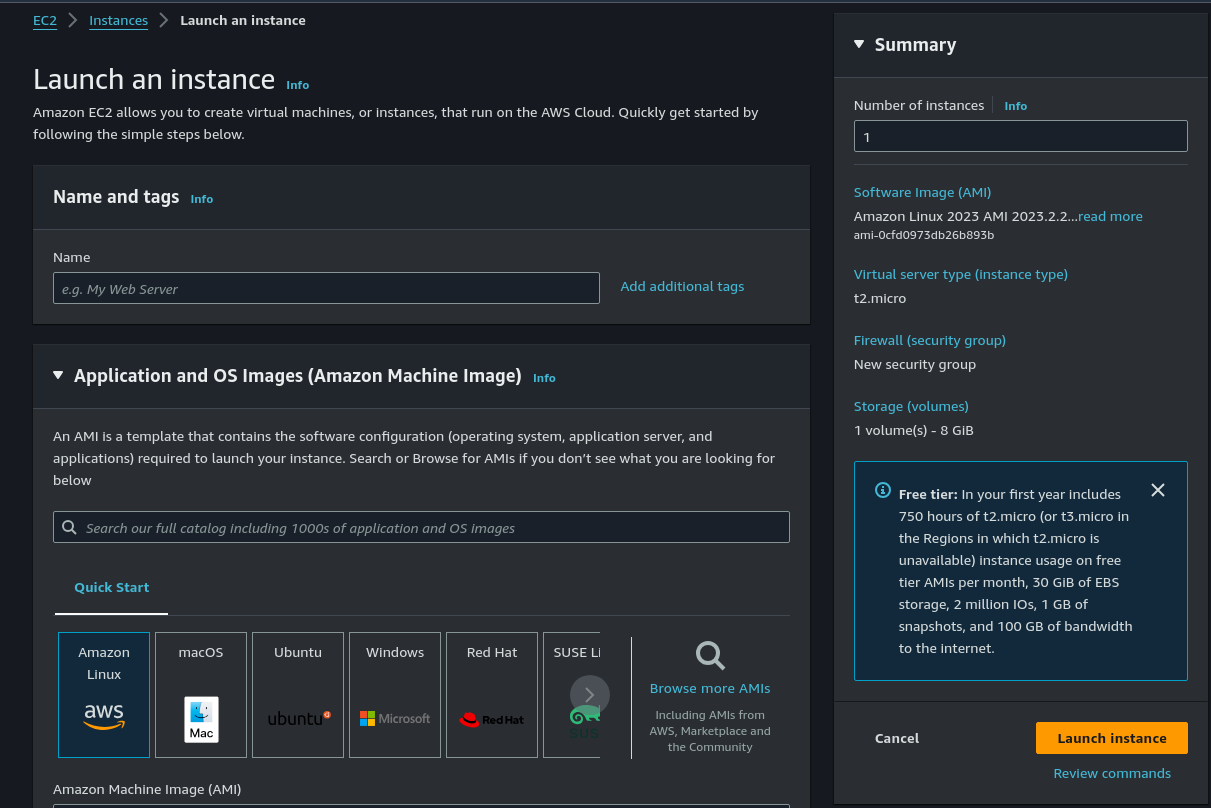


There will be the instances

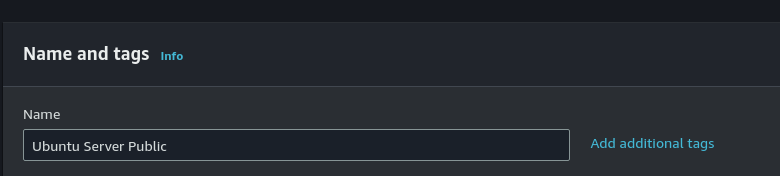


The instances page should look like this

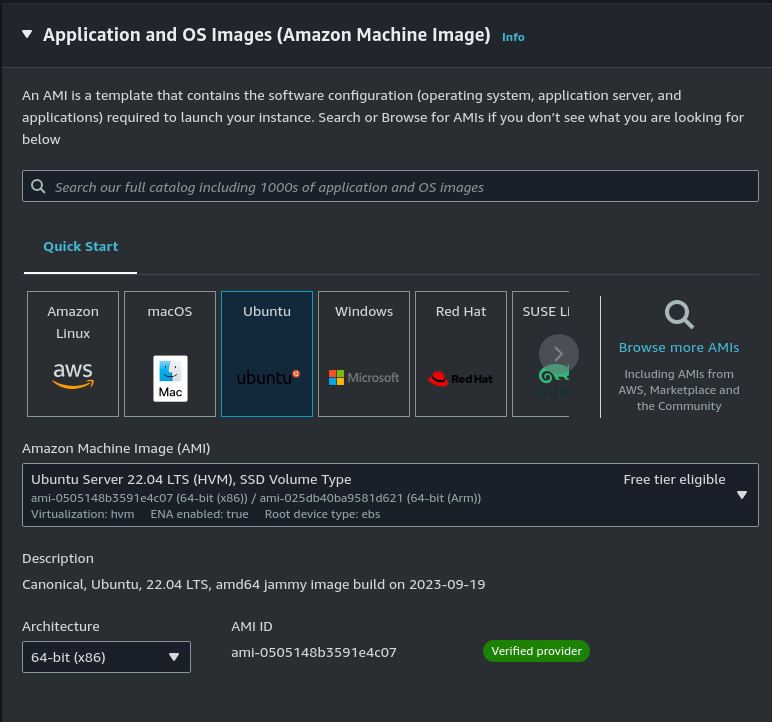
To create our first Instance click on the Launch instances button, it can be either the gray button on the center or the orange button on the top right



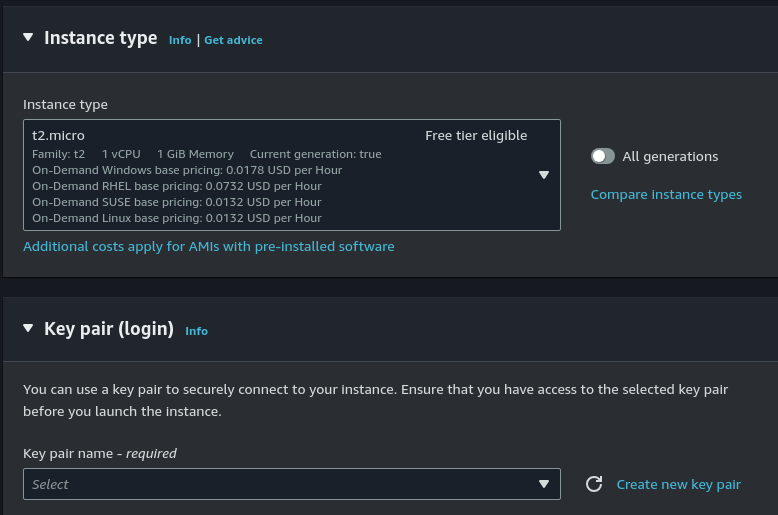
The launch instance page will look like this



From here we will give it a name, this is will be our public server so we will call it **Ubuntu Server Public**

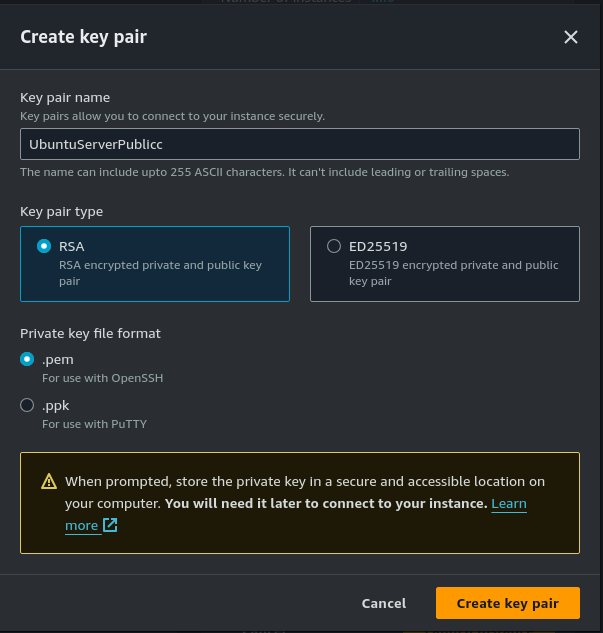


Now we will select the OS, we want **ubuntu server** so select ubuntu and it will automatically give us **ubuntu server**, if not feel free to select it from the dropdown box and make sure it says **free tier eligible**

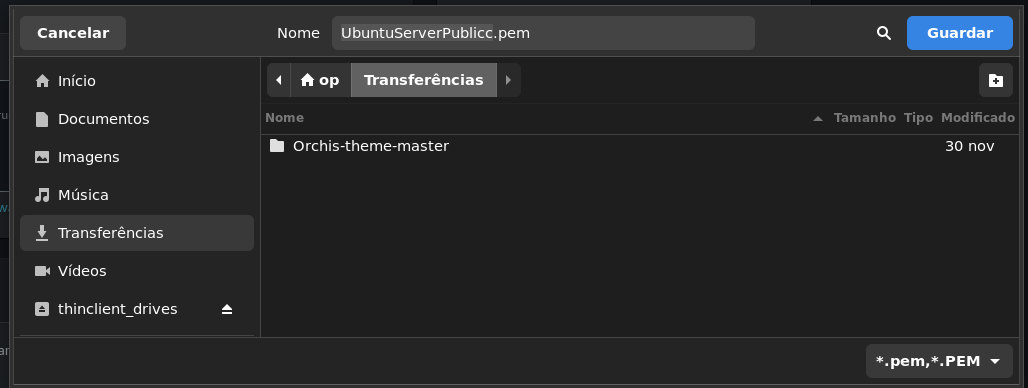
****

In the instance type leave it at default and on key pair click **Create new key pair**

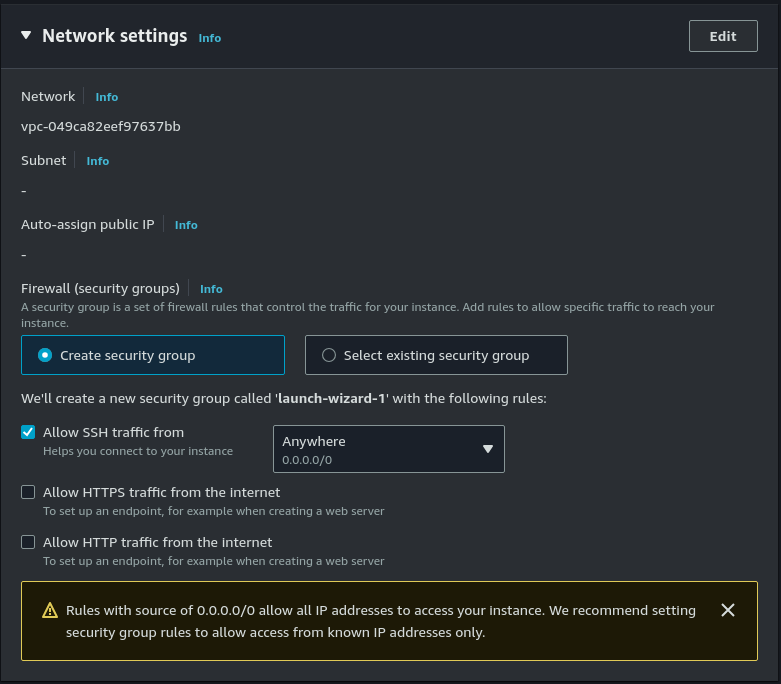
****

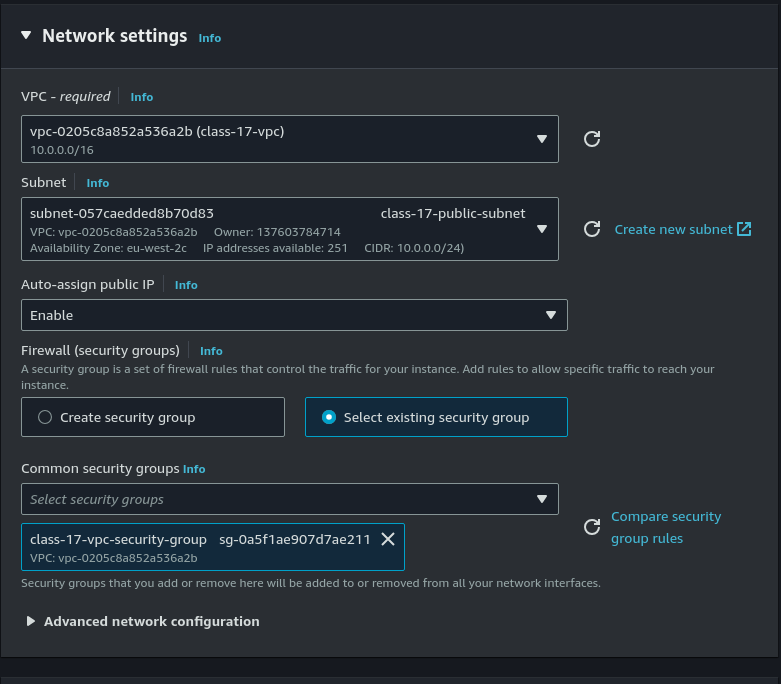
****

This menu will give it a name, for this key i will call it **UbuntuServerPublicc** and leave everything else at default

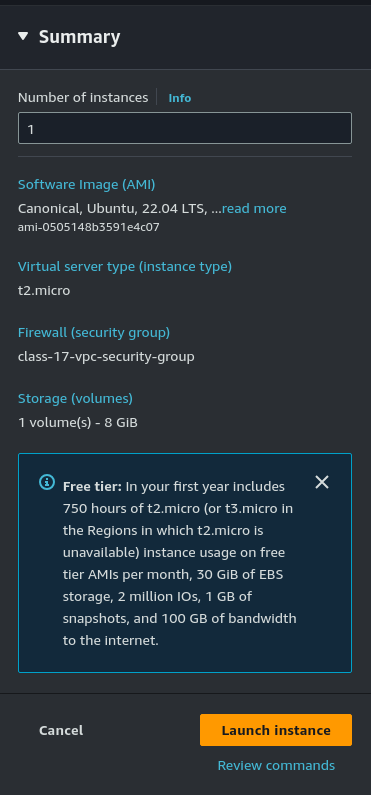


It will ask where we want to save the key on our computer for this i will simply save it on the Downloads directory

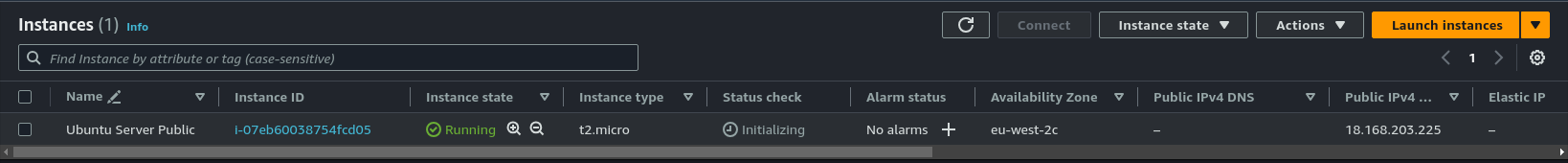
Now on the network settings we will click the edit button



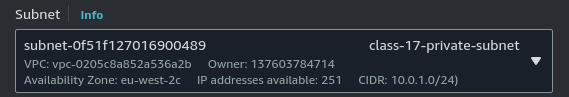
Now we will select our VPC from the dropdown, next select our subnet, at last click the “**Select existing security group**” and select our previously created security group



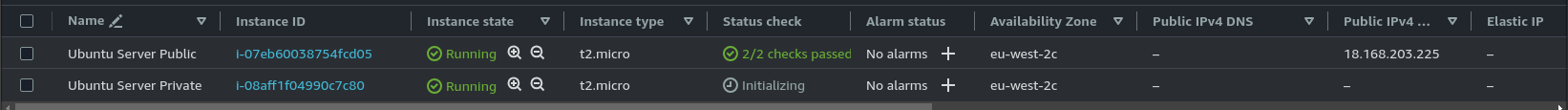
Leave the rest at its default and click **Launch instance** to create the instance



On our instances page we can see our instance created



The steps are the same for the private instance except on the subnet we will give it the private subnet instead.



Our instances page should look like this



From our local pc if we try to ssh to our public instance we can successfully enter it



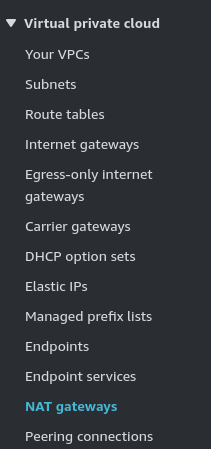
But if we try to ssh to the private instance it we won't be able to which is what we want

## 

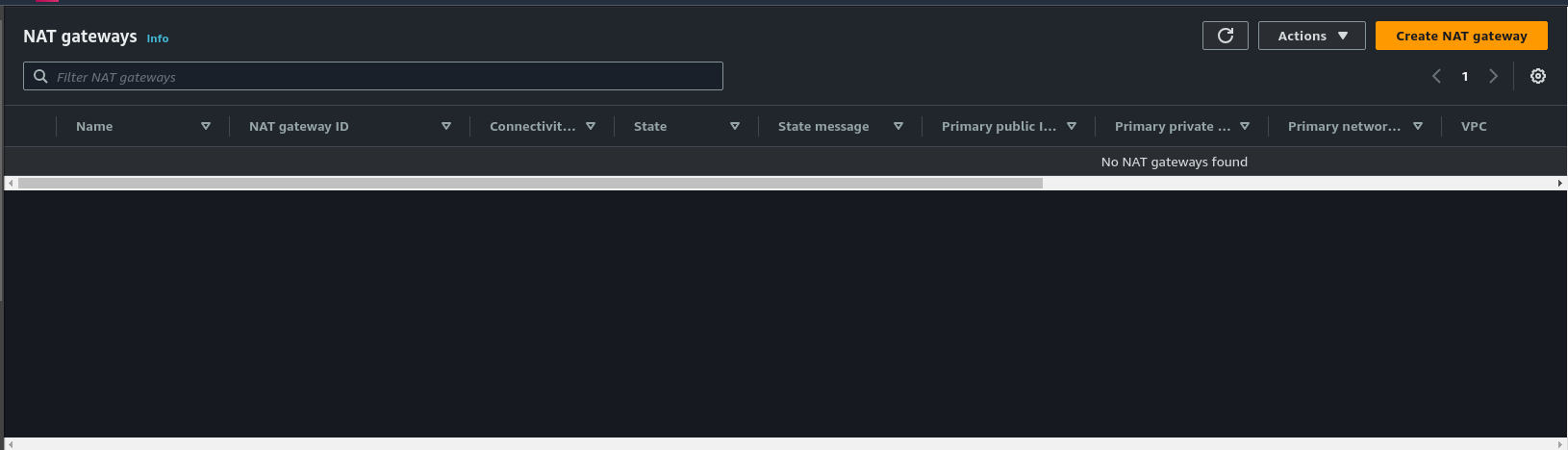
## 

## Part 5: NAT Gateway

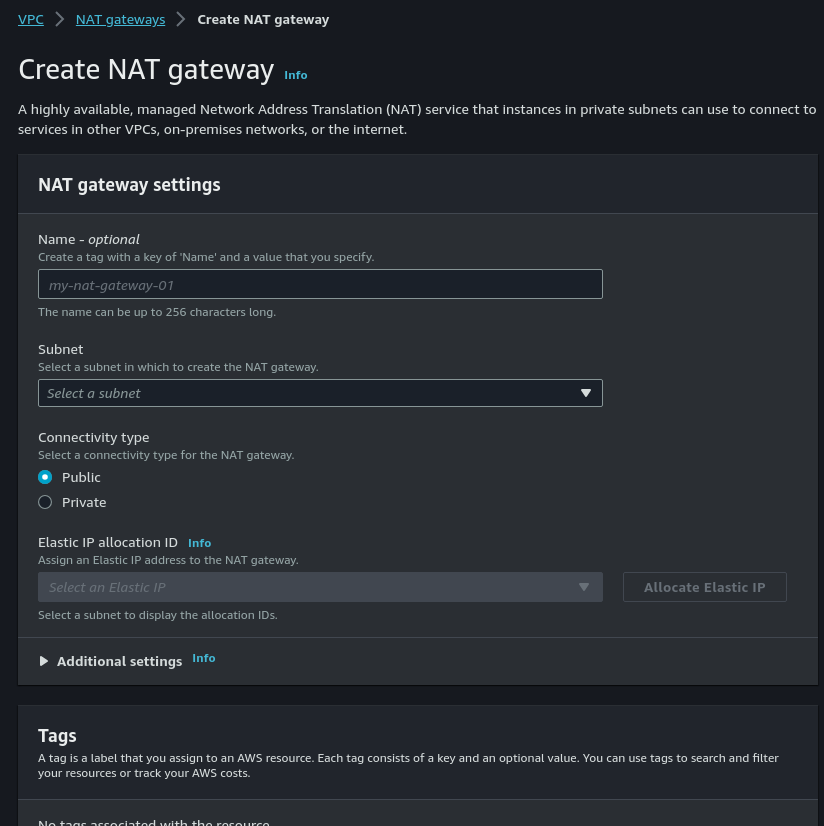
To create a NAT Gateway we need to go back to our VPC page

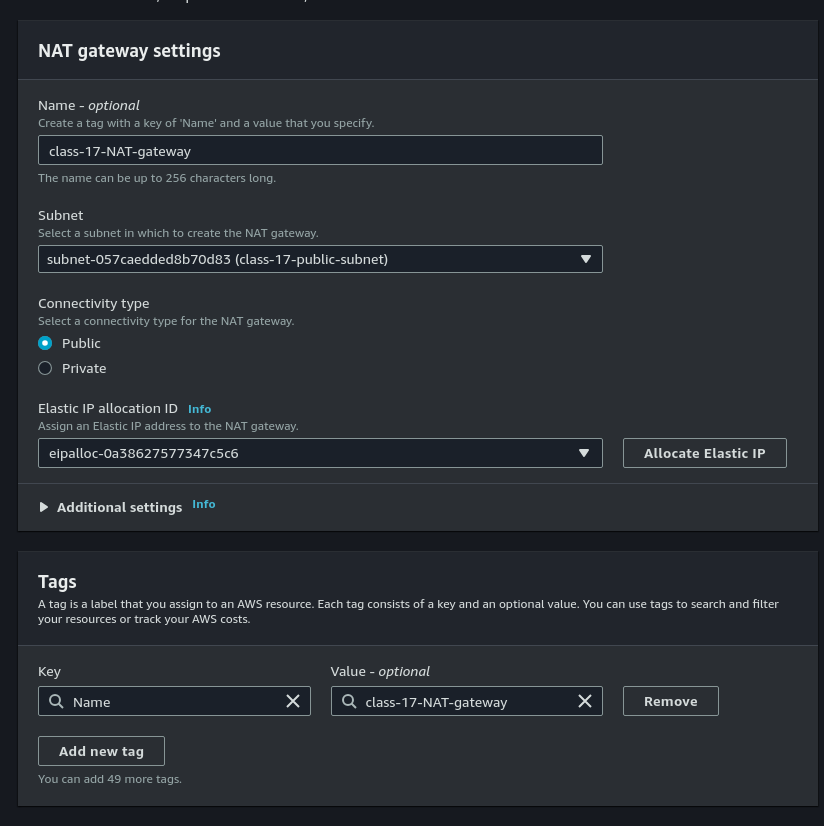


On our left panel we will find NAT gateways



NAT gateway page will look like this





Firstly we give it a name we will give it **class-17-NAT-gateway**

on our subnet we select our public subnet

leave the connectivity type on Public

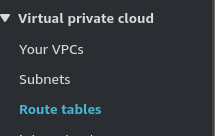
and click the **Allocate Elastic IP** button to assign one to our NAT gateway

On our Tags we will give it a value and it will be class-17-NAT

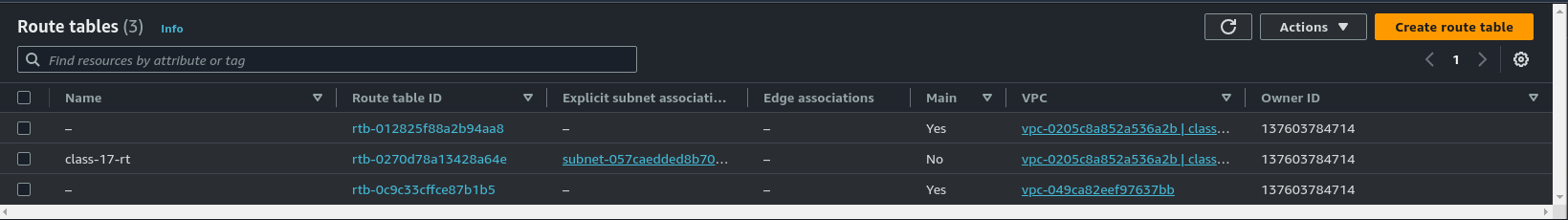
## 

## Part 6: Configure Route Table for NAT Gateway

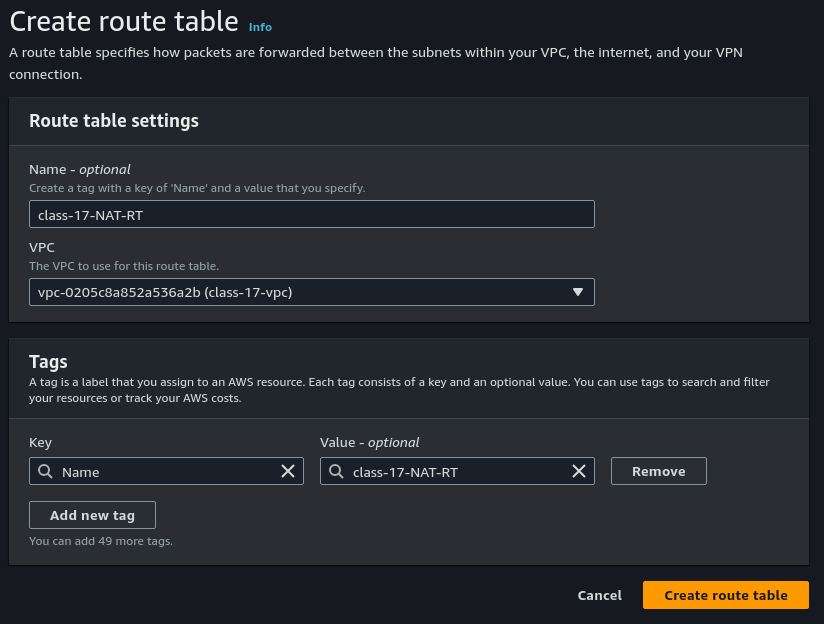
We will create a new routing table for our NAT gateway



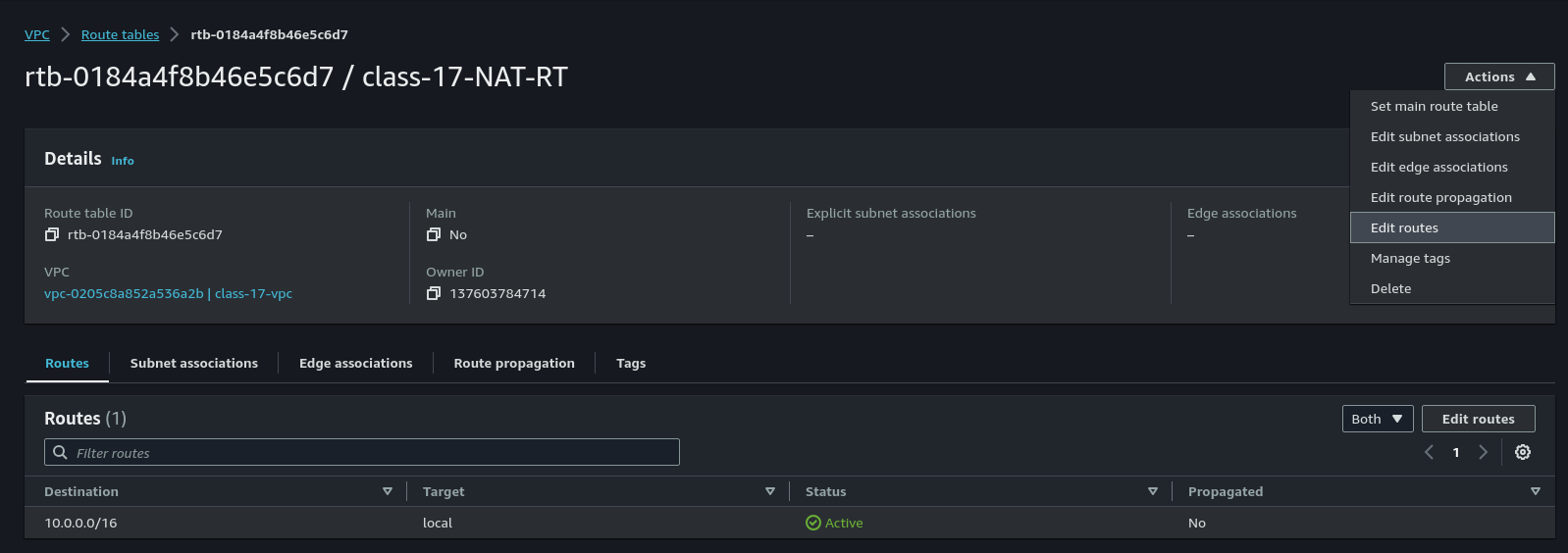
So again from our left panel we will select Route tables



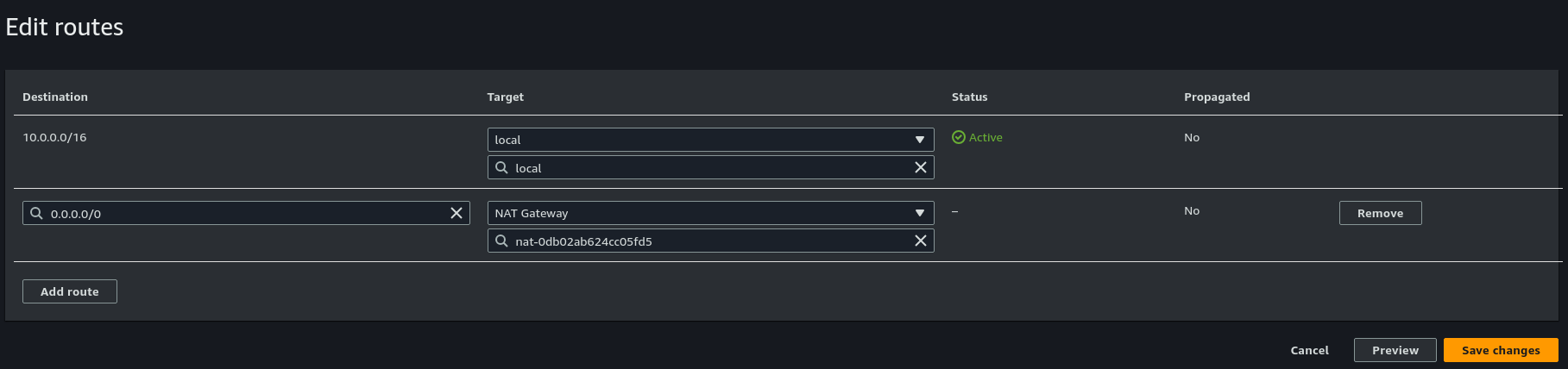
Click **Create route table**



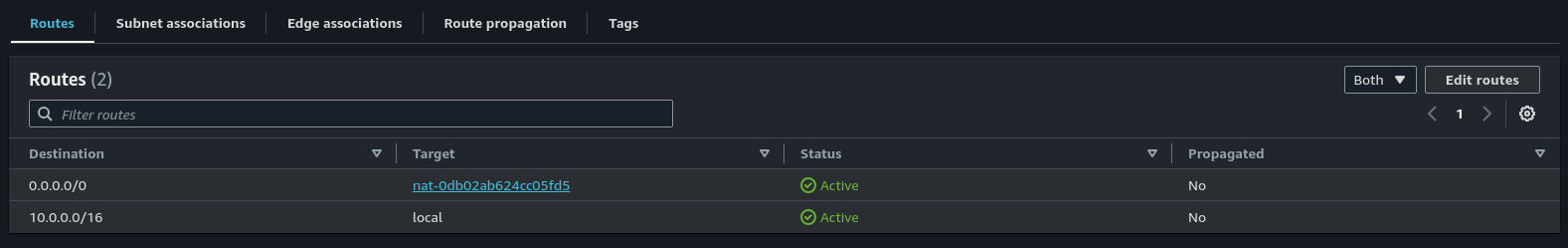
Give it a name, select our VPC and create the route table



From our route table page we will click the action button and select from the dropdown edit route



Inside click the add route button on the destination we will use 0.0.0.0/0 and on the target we will select our NAT gateway



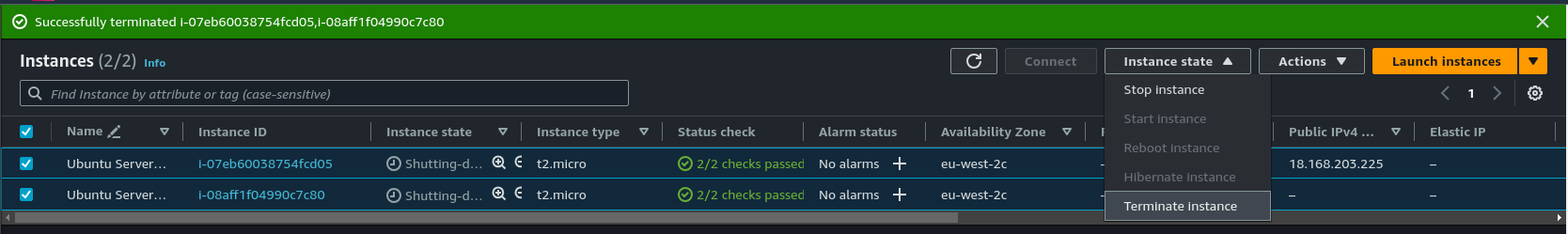
After creating it will show up on our routes

## 

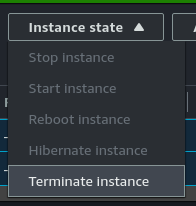
## 

## Part 7: Wrap Up

Now that we are done, to avoid incurring cloud charges we are going to terminate our EC2 Instances



To terminate our instances we simply select them



and on instance state we select **Terminate instance**