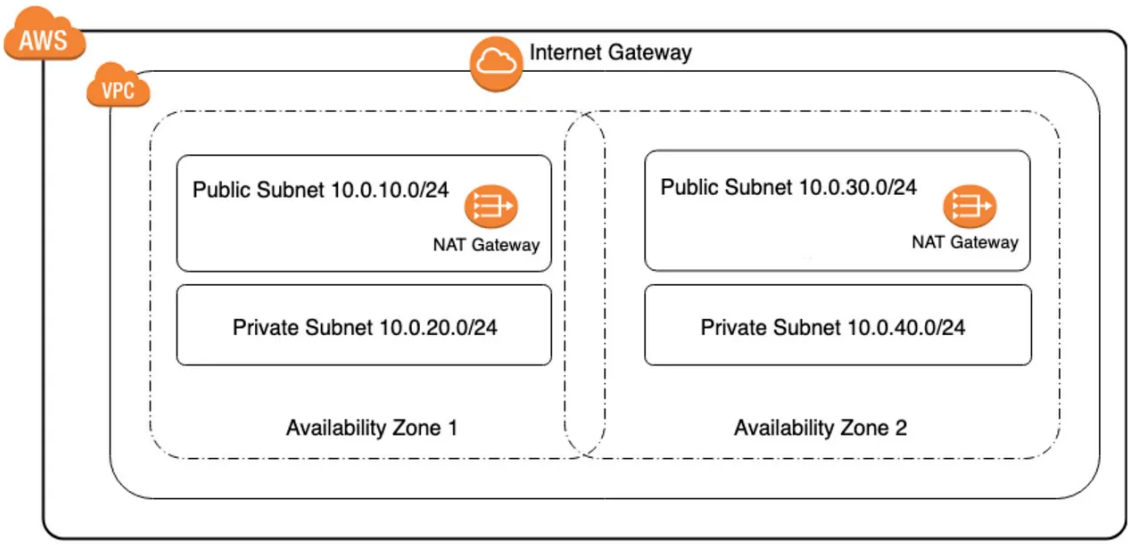
#4 Task: Create VPC, Public & Private Subnets.



What is a VPC and why is it important?

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

Amazon VPC enables you to build a virtual network in the AWS cloud — no VPNs, hardware, or physical data centers required. You can define your own network space, and control how your network and the Amazon EC2 resources inside your network are exposed to the Internet.

What is the importance of having private and public subnets?

The instances in the public subnet can send outbound traffic directly to the internet, whereas the instances in the private subnet can't. Instead, the instances in the private subnet can access the internet by using a network address translation (NAT) gateway that resides in the public subnet.

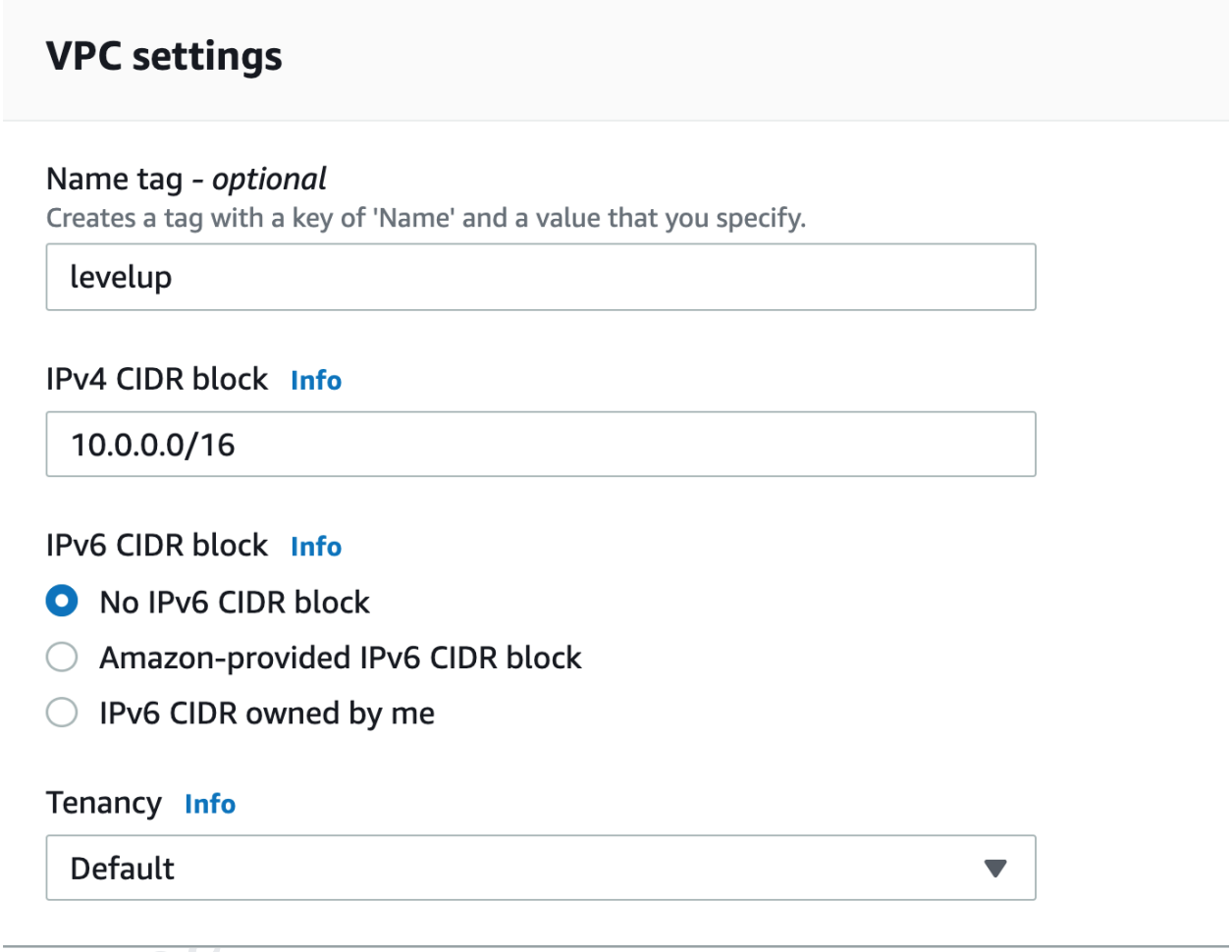
Requirements:

- AWS Console → https://aws.amazon.com/free/
- Patience…

The Process:

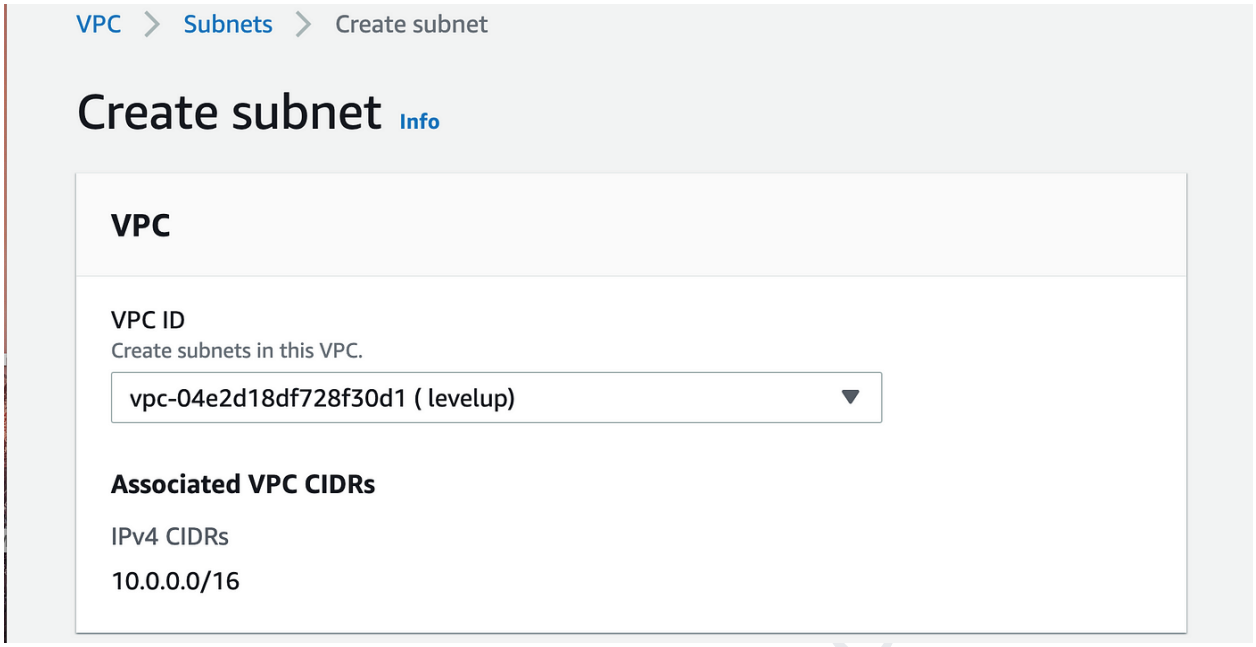
- Follow the steps below. Use the same IPv4 CIDR block numbers. You can make the names of the VPC, subnets, route tables, IGW, NAT gateways unique to whatever you want.

1. Create VPC from VPC Dashboard from AWS Console.

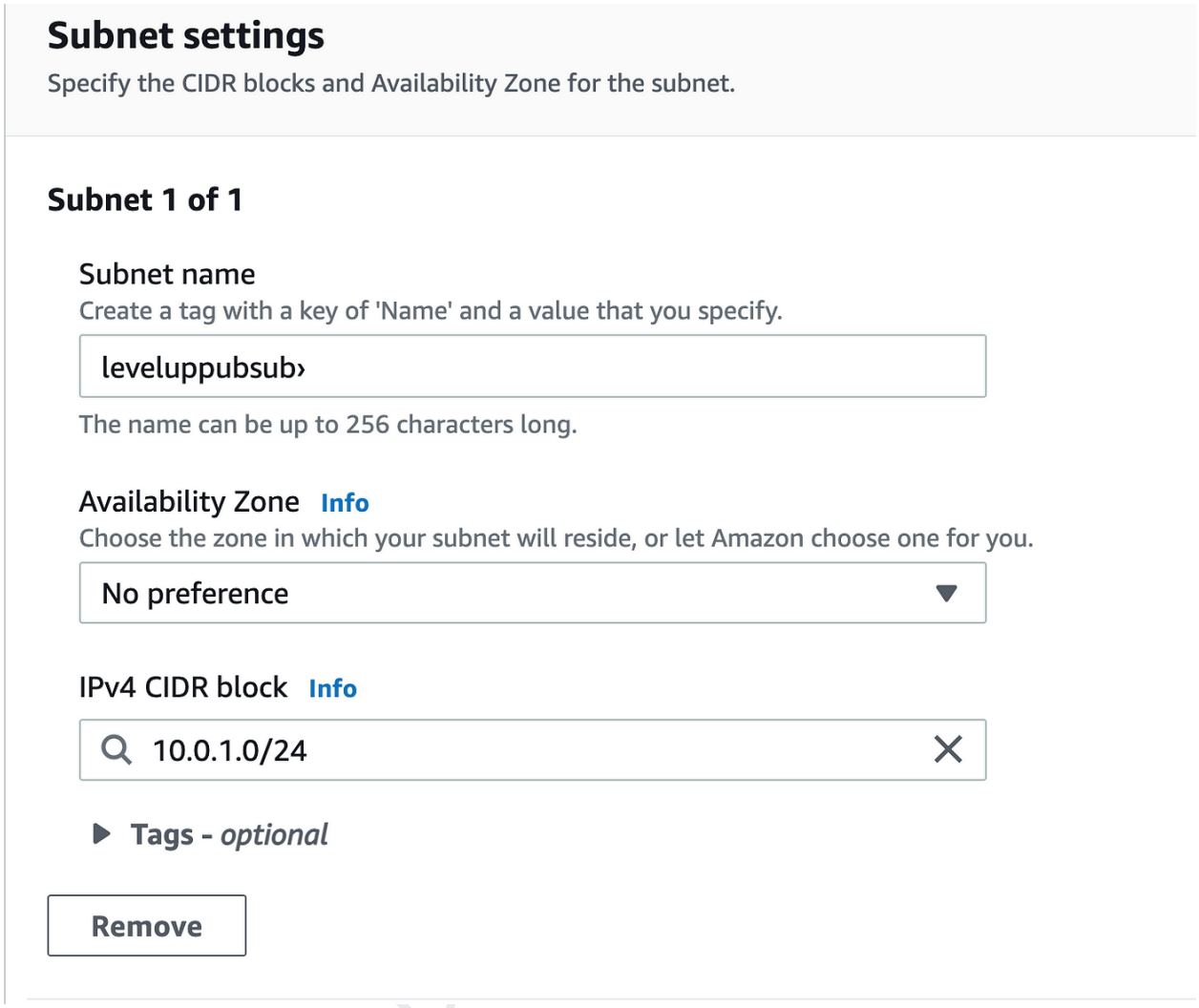


- Click on Create button which will create a VPC

2. Create subnets now, we'll start with creating a public subnet now.



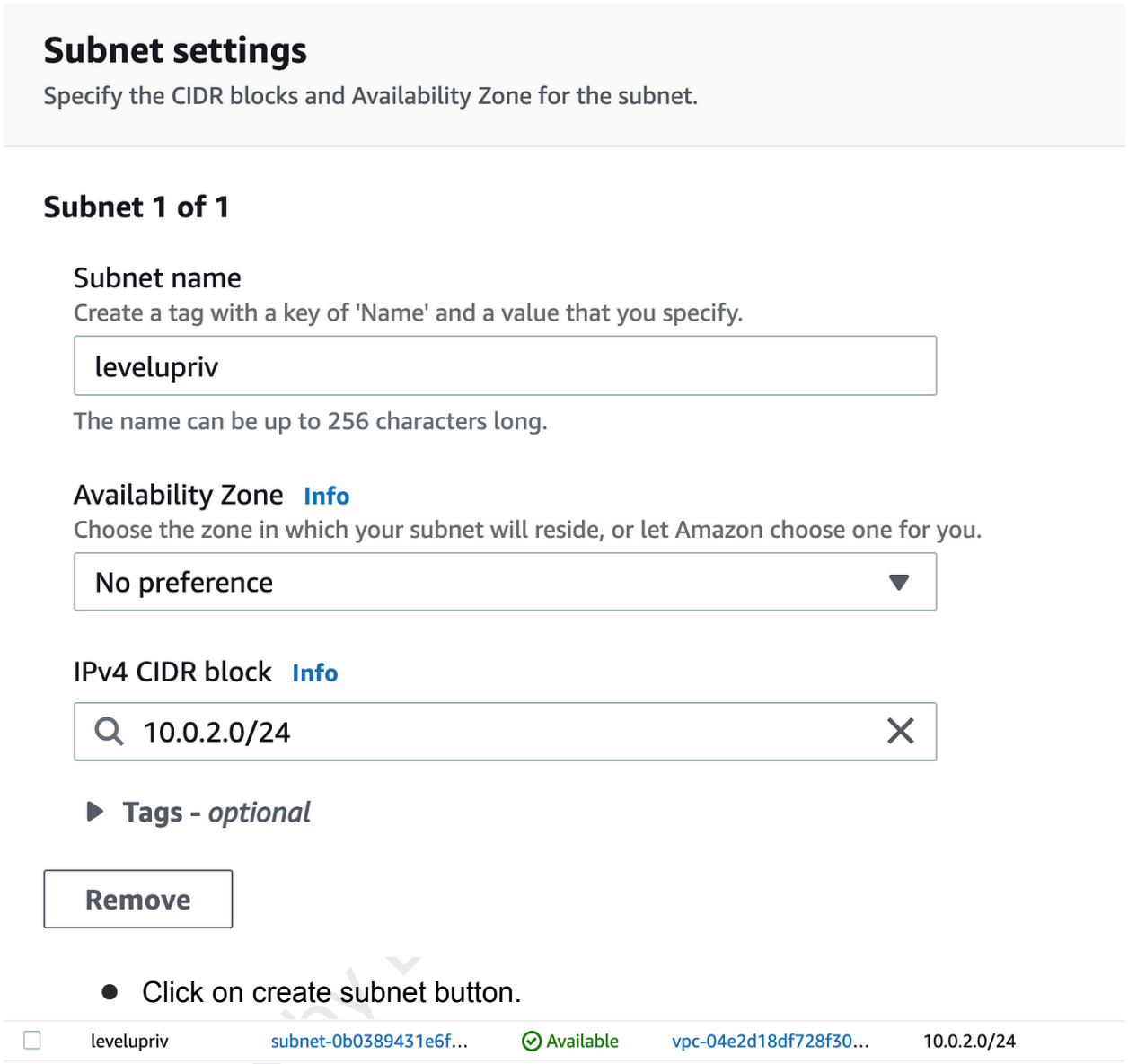
VPC ID is the vpc you created



Availability Zone I left as no preference for this particular project.

3. Let's create a private subnet now.

This can be created using Subnets options from left hand side list in VPC Dashboard. Same process as the public subnet, except we are using a different IPv4 CIDR block.

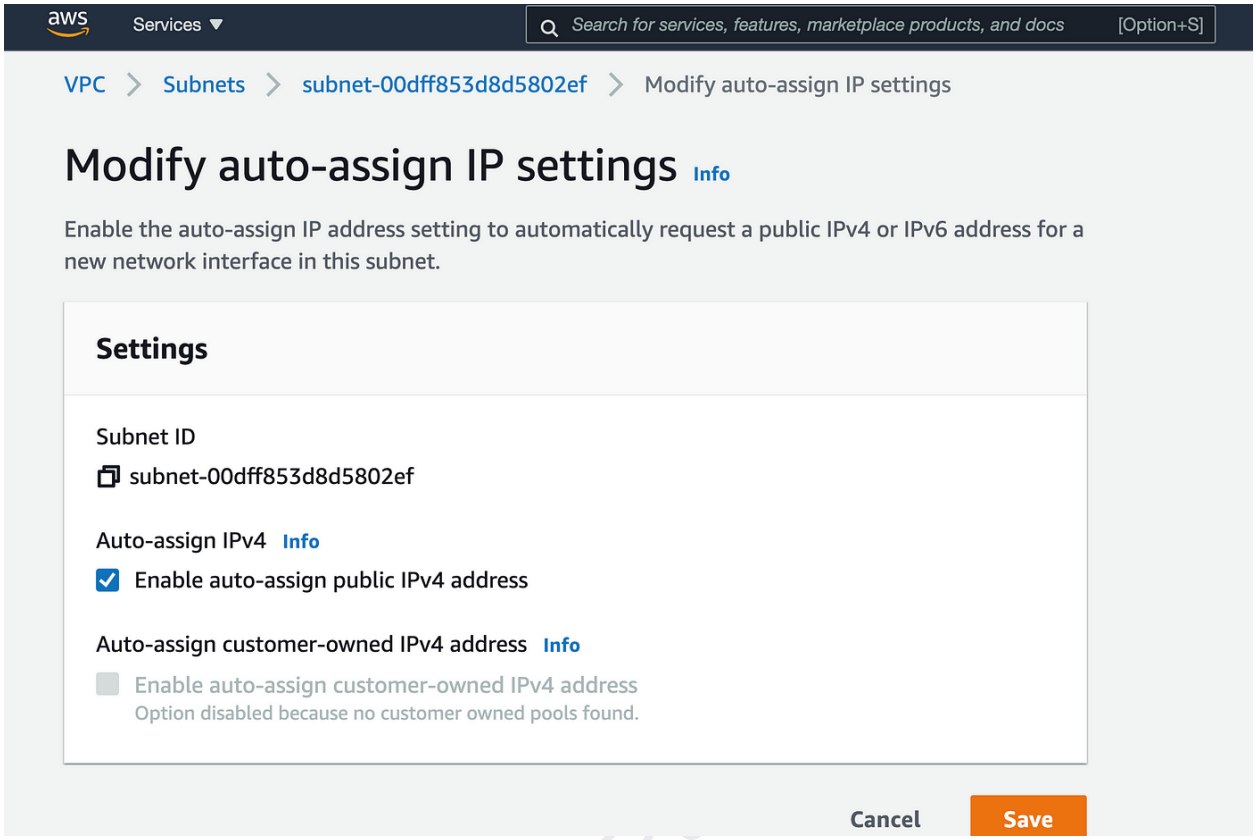


- Click on create subnet button.

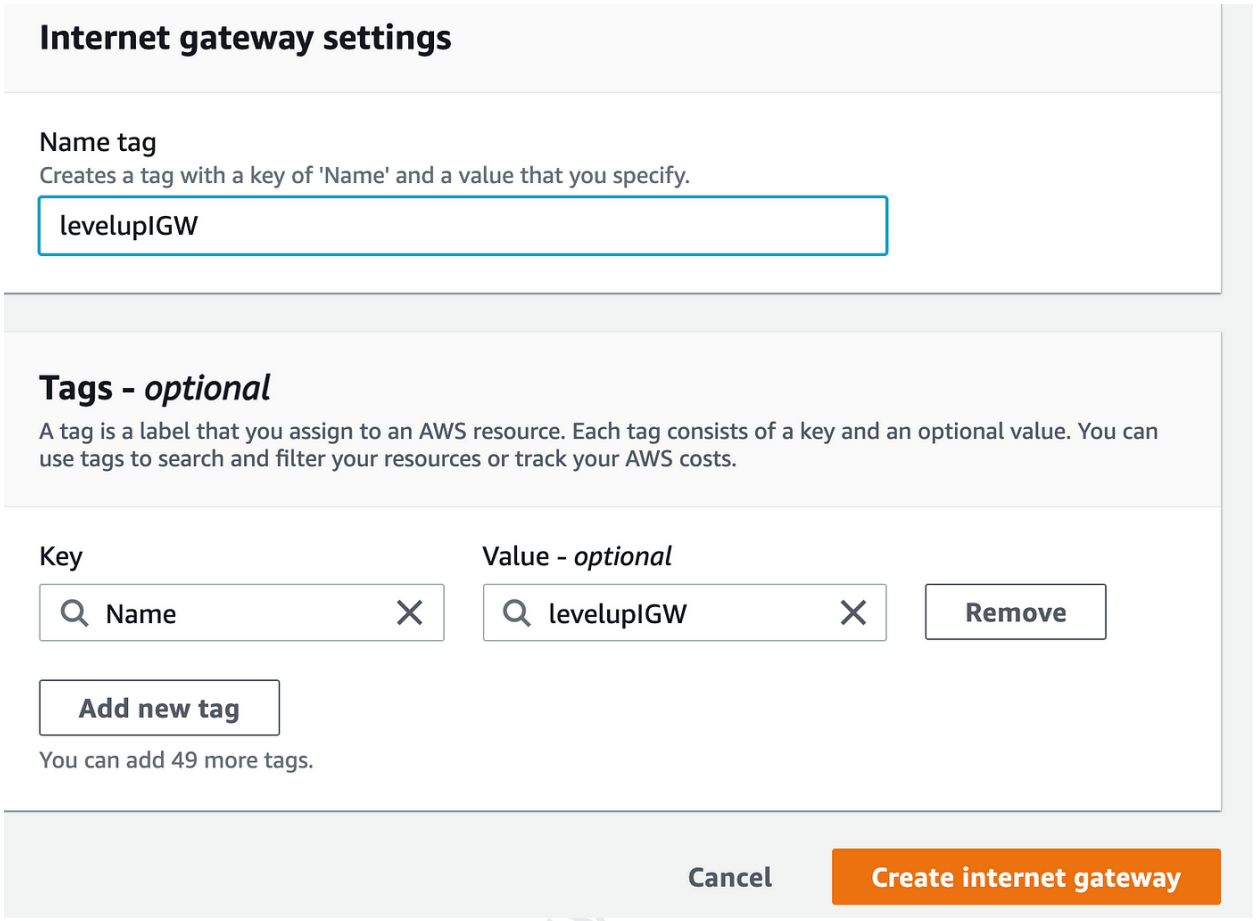
What is an AWS CIDR block?

- When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a Classless Inter-Domain Routing (CIDR) block; for example, 10.0. 0.0/16 . This is the primary CIDR block for your VPC.

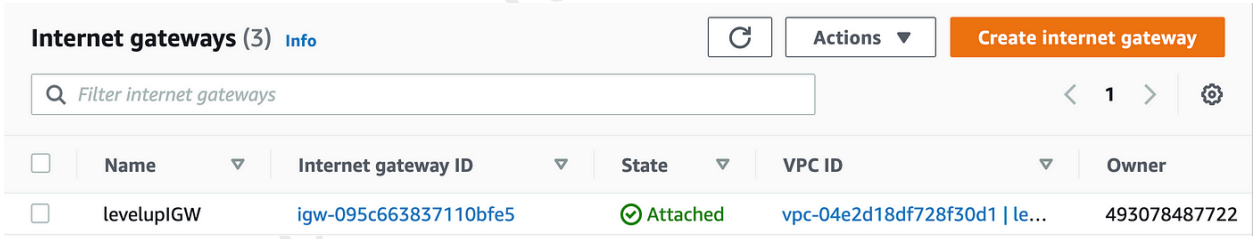
4. Modify Auto assign IP by right clicking on public subnet.



5. Create an Internet Gateway to use with our Public Subnet.



Yes, it's really as easy as just creating a tag.

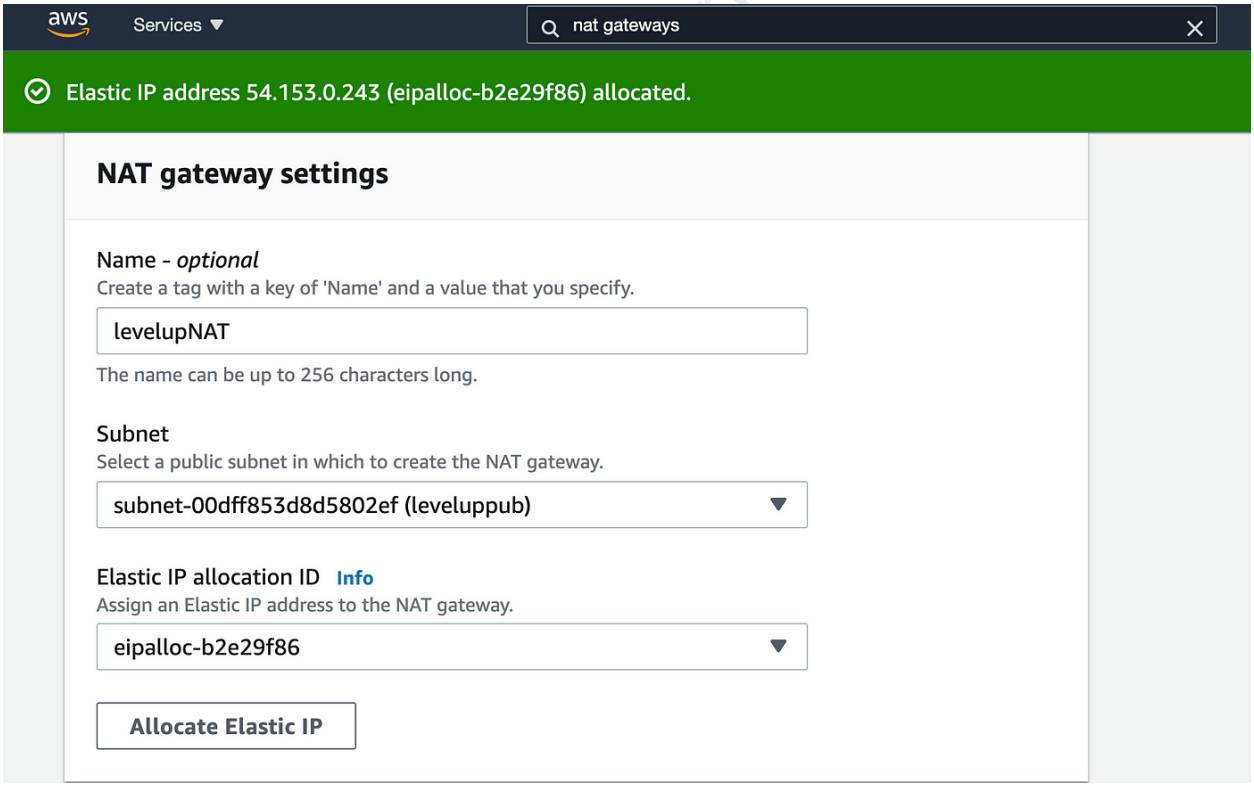


6. Attach Internet Gateway to VPC.

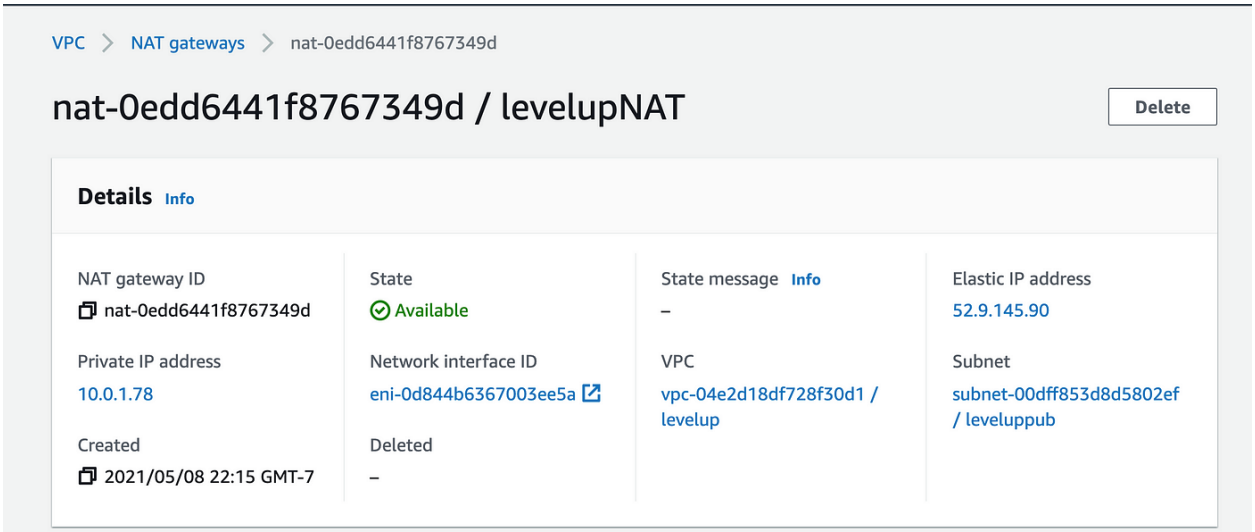
- When you right click on internet gateway, it will show you Attach to VPC option as below.
- Select the VPC you created and click on attach.

7. Create NAT Gateway.

- You can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances.
- Important: In order to access internet to your private subnet, NAT Gateway must be added to Public Subnet only.

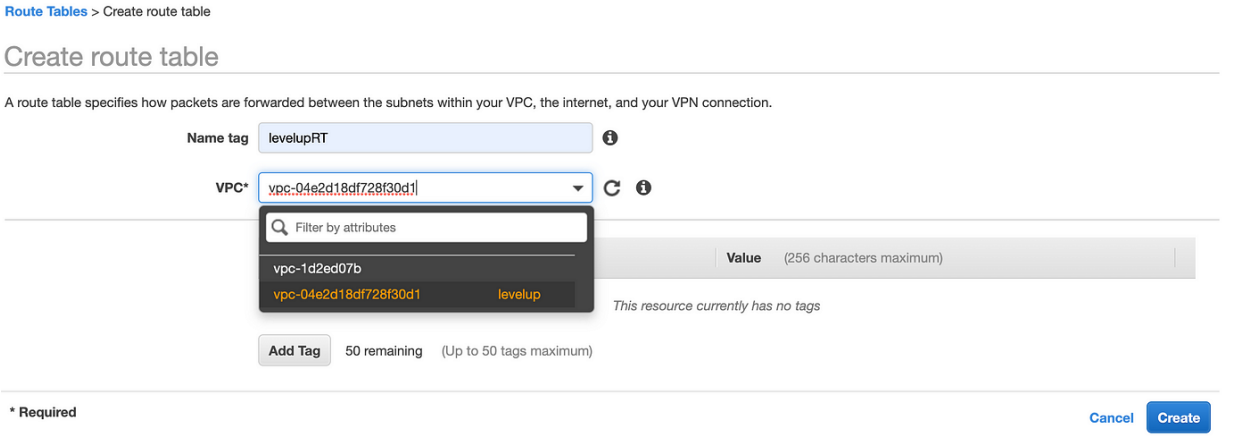


click on allocate elastic IP to attach elastic IP allocation ID

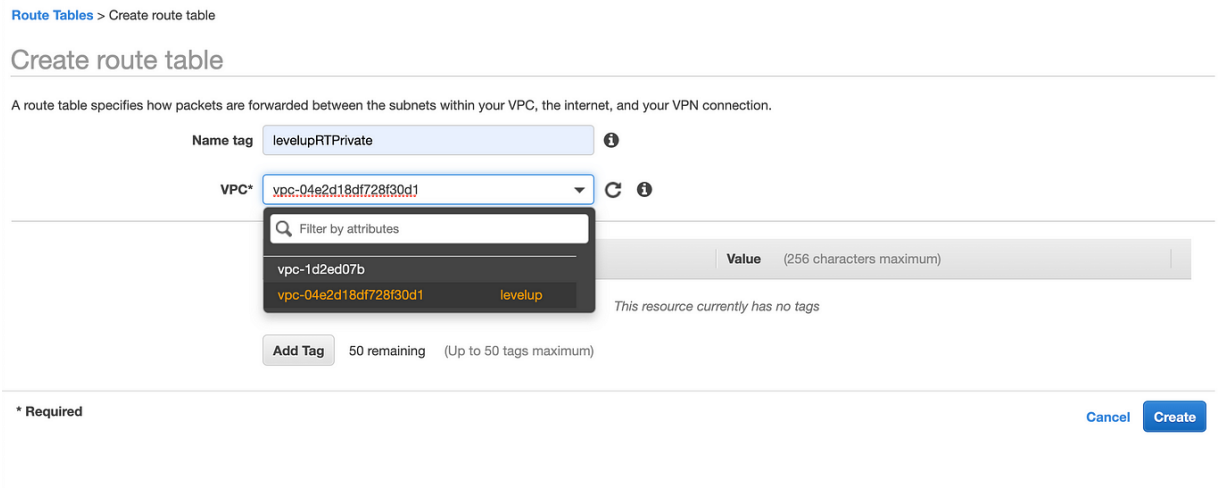


8. Create Route Tables.

- A route table contains a set of rules, called routes, that are used to determine where network traffic from your subnet or gateway is directed. To put it simply, a route table tells network packets which way they need to go to get to their destination.

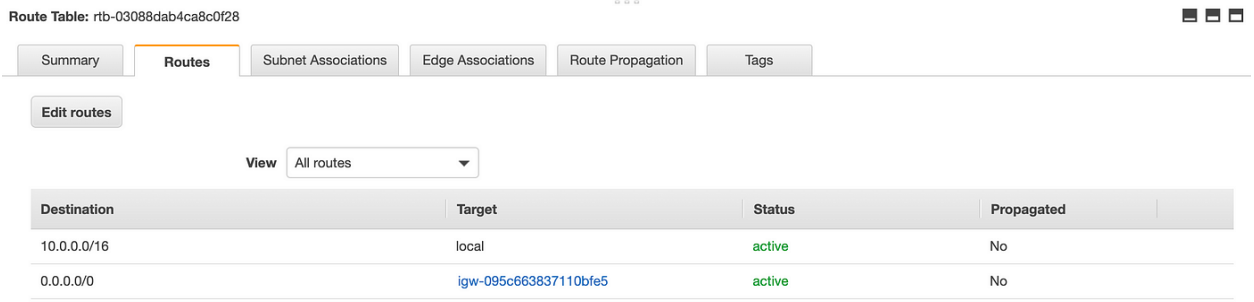


public route table



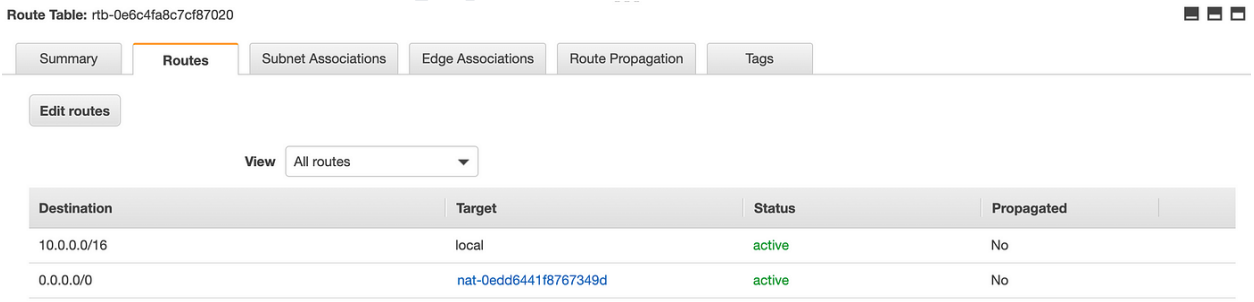
private route table

- Add Internet Gateway to Public Route Table. Click ADD routes and attach.



public route table

- Add NAT gateway to Private Route Table. Click ADD routes and attach.



private route table

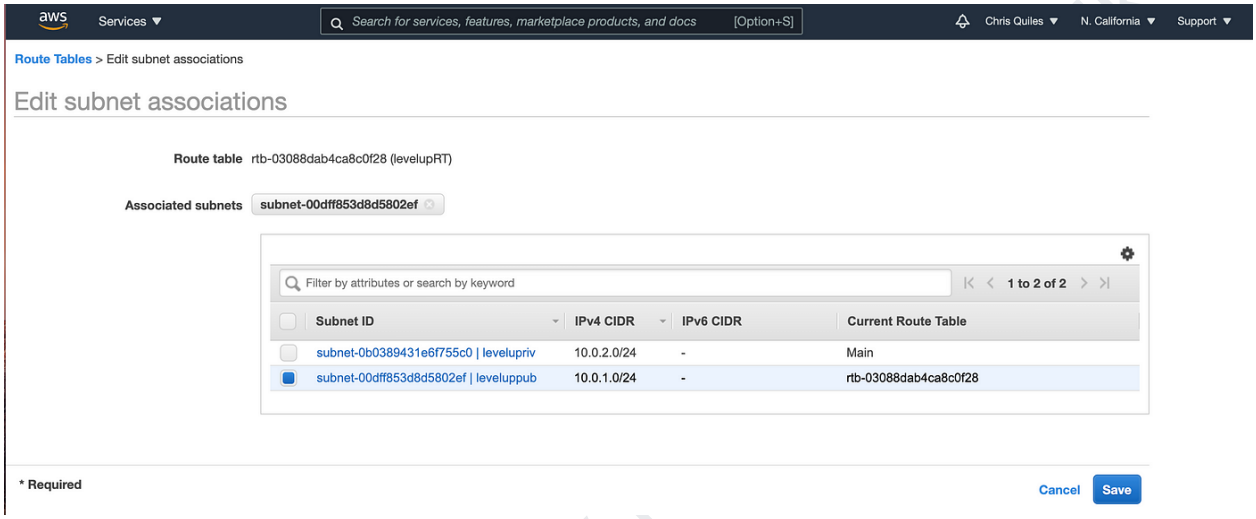
- Since we added NAT Gateway to public subnet, it will also have access to the internet.

9. Edit Subnet Association

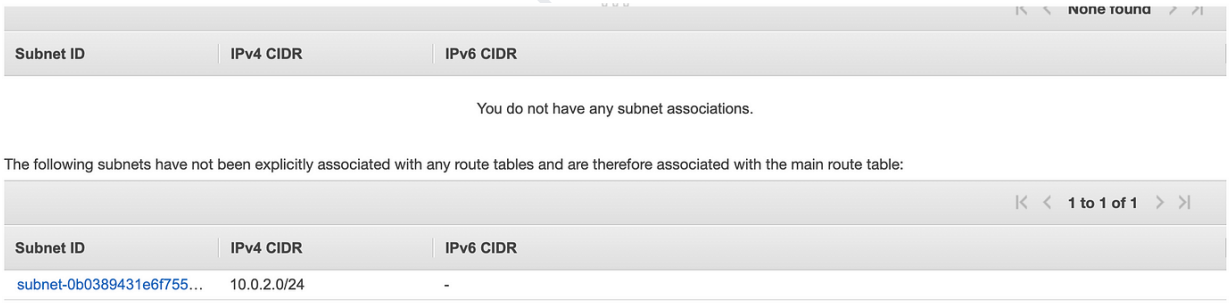
- Repeat steps for both your public and private Route Tables.

- Click Edit Subnet Association button.

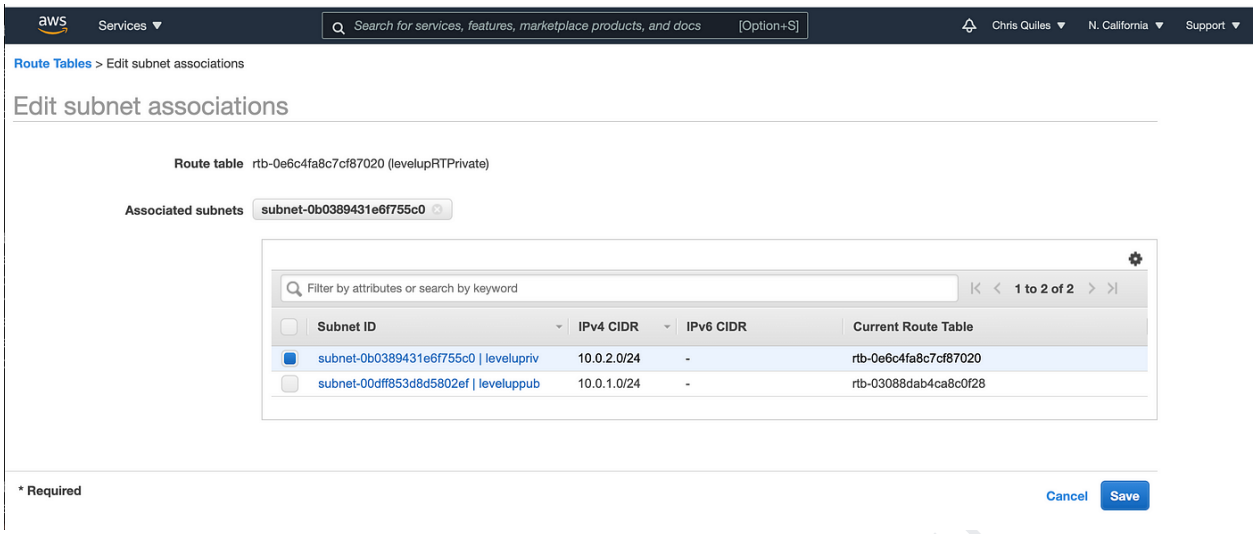
public route table



public route table



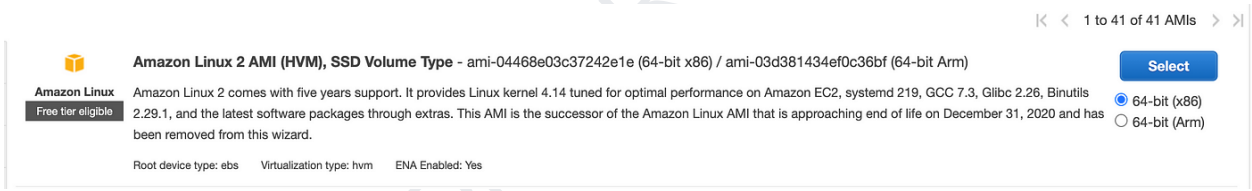
private route table



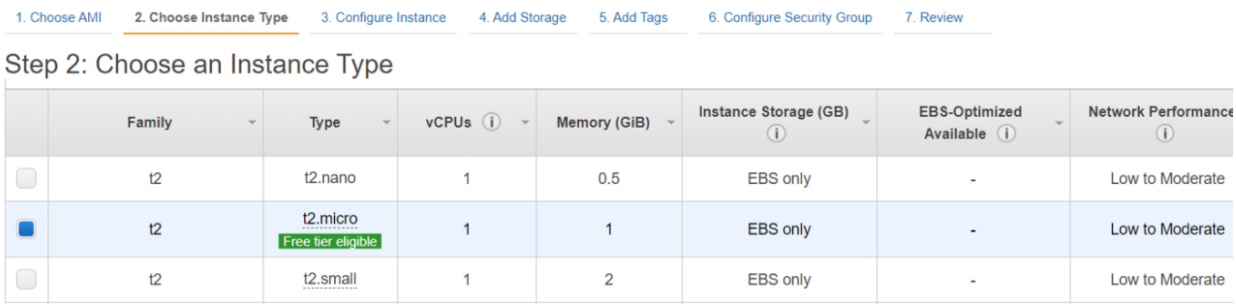
private route table

10. Create public and private EC2 instances.

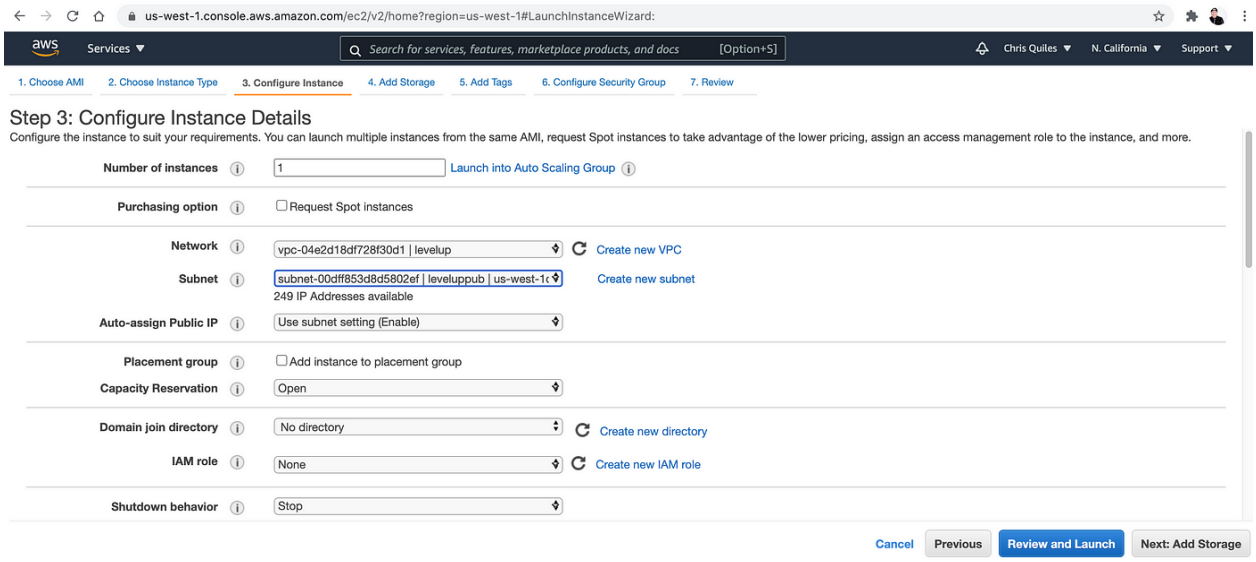
- Follow process for both public and private instances.
- Pay attention to steps for what is particular to a certain instance.
- First step, choose an AMI.



- Instance Type.

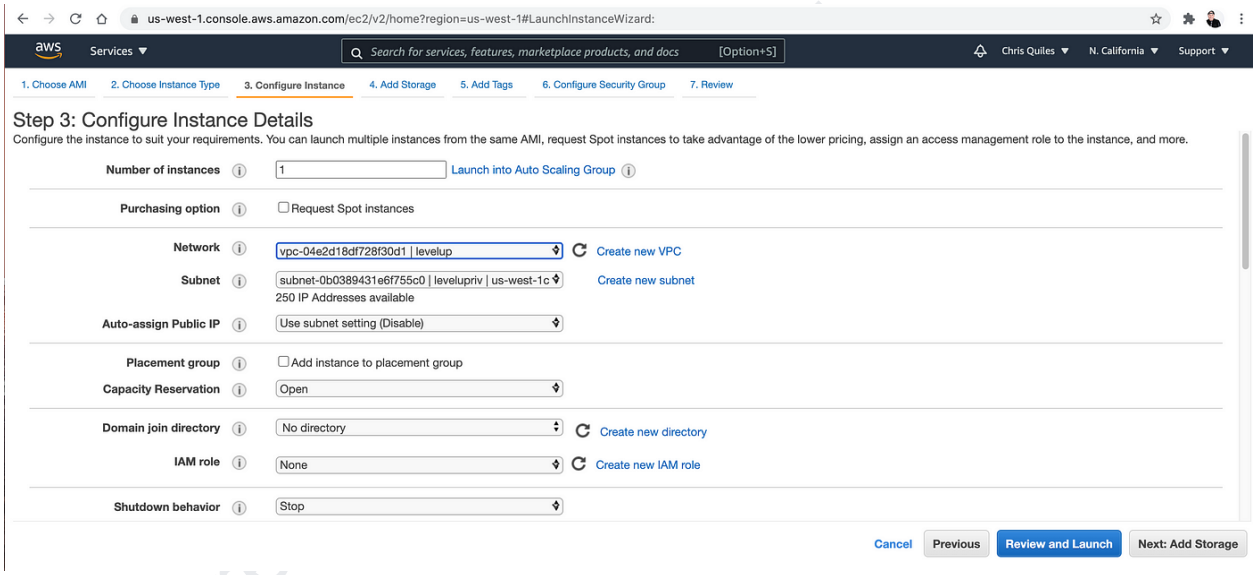


- Public Configure Details



public instance

- Private Configure Details

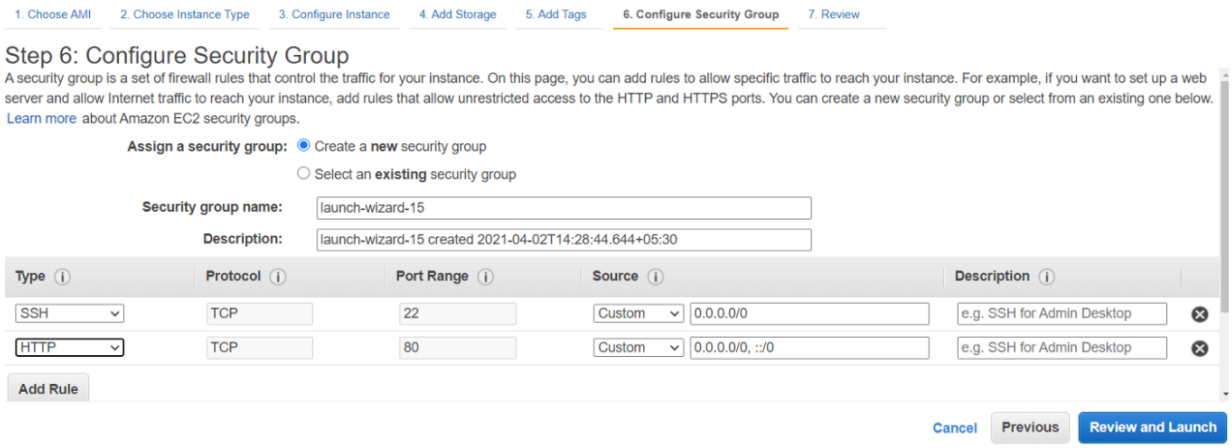


configure instance details should be the same for both instance, except the public and private subnets.

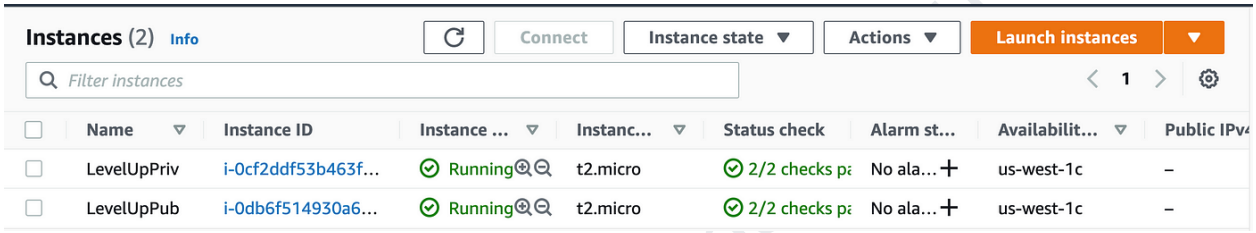
- Copy and paste User Data from below into PUBLIC instance.

```
#!/bin/bash
yum install httpd –y
yum update -y
service httpd start
chkconfig httpd on
```

- Configure Security Group. Make sure to add HTTP port 80 to both public and private instances. SSH port 22 will already be there when created.

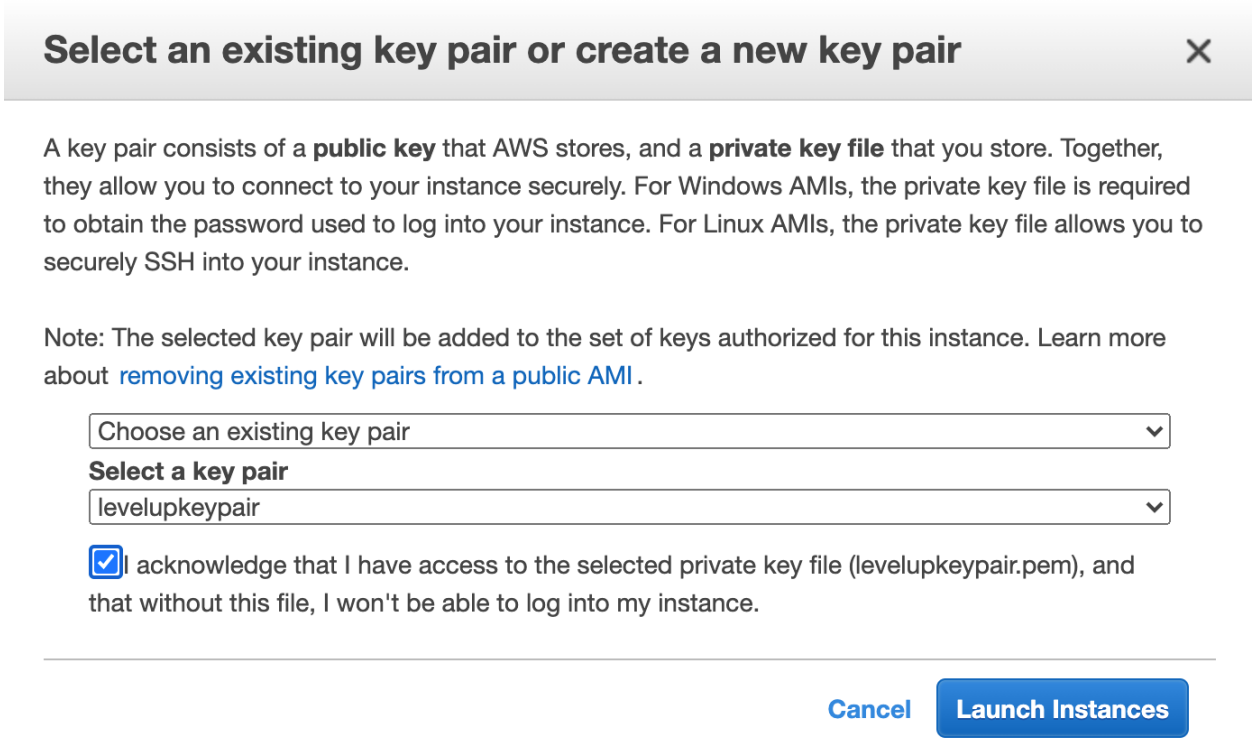


configure security group is the same for both public and private instances



public and private instances should be running as so

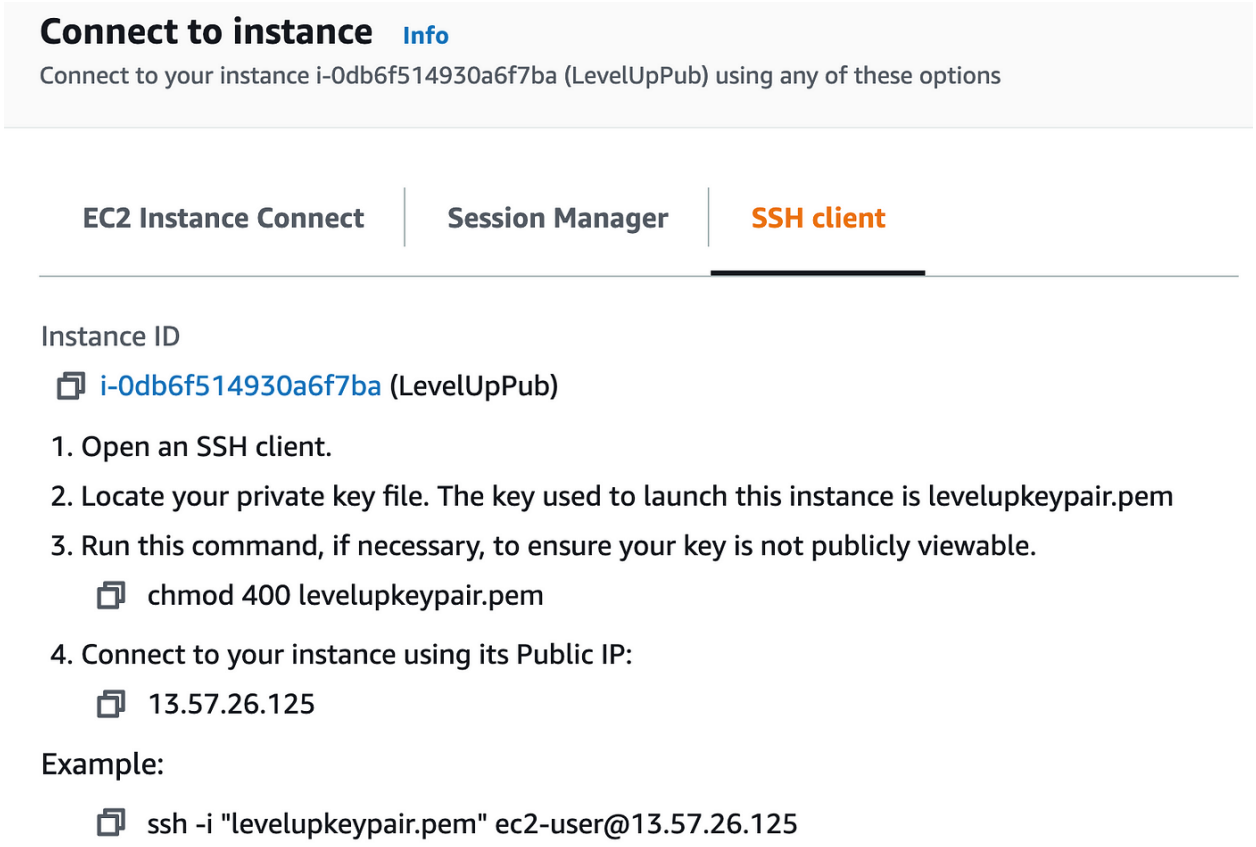
- Create a keypair or using an existing keypair.



11. Testing the EC2 Instances.

Public Instance:

- right click on box next to name of instance and click connect



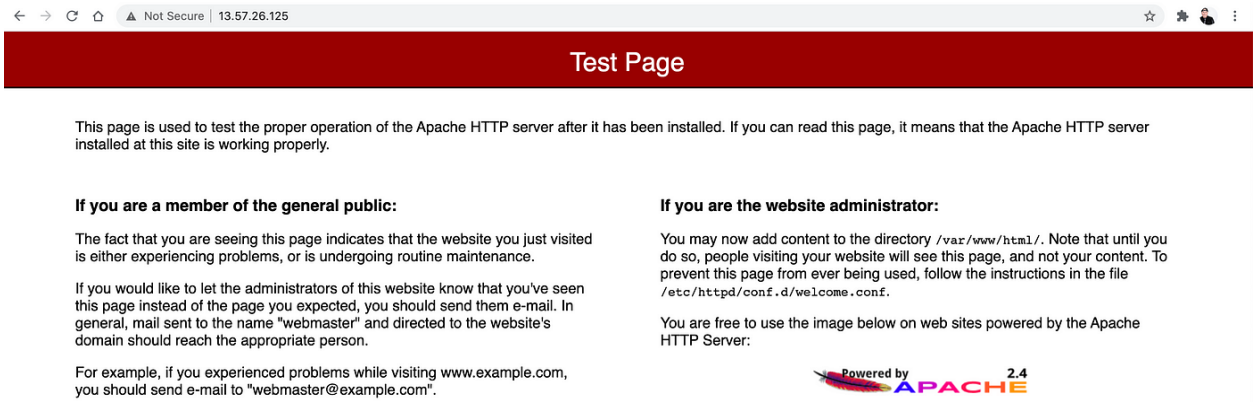
ssh steps to connecting to public instance

```
jerryc.quiles@Jerrys-MacBook-Pro ~/d/devops> chmod 400 levelupkeypair.pem
jerryc.quiles@Jerrys-MacBook-Pro ~/d/devops> ssh -i "levelupkeypair.pem" ec2-user@13.57.26.125
The authenticity of host '13.57.26.125 (13.57.26.125)' can't be established.
ECDSA key fingerprint is SHA256:2xIX63IFcwyXC62HpATCYhUim/duOvISzaGXo5PuP7E.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.57.26.125' (ECDSA) to the list of known hosts.
Last login: Sun May  9 06:44:18 2021 from c-73-241-240-172.hsd1.ca.comcast.net

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

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-1-233 ~]$
```

terminal of successfully connecting public instance



apache test page from our public instance IP address
Private Instance:

- We'll now connect to our private instance through our public instance.
- Inside you public instance create a file for your keypair.
- create a file for your keypair.

yum install vim
vim keypair.pem
:wq

- Copy and paste contents of keypair inside your newly created keypair.pem file, your keypair file will look like the following below.

chmod 600 keypair.pem
ssh -i keypair.pem ec2-user@private-ip-address

```
[ec2-user@ip-10-0-1-233 ~]$ vim levelupkeypair.pem
[ec2-user@ip-10-0-1-233 ~]$ chmod 600 levelupkeypair.pem
[ec2-user@ip-10-0-1-233 ~]$ ssh -i levelupkeypair.pem ec2-user@10.0.2.165
Last login: Sun May  9 06:52:49 2021 from 10.0.1.233

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

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-2-165 ~]$
```

12. Conclusion.

- In conclusion, the machines on a private subnet can access the Internet because the default route on a private subnet is not the VPC “Internet Gateway” object — it is an EC2 instance configured as a NAT instance. A NAT instance is an instance on a public subnet with a public IP, and specific configuration.