day -4 AWS -- VPC and Dynamo DB

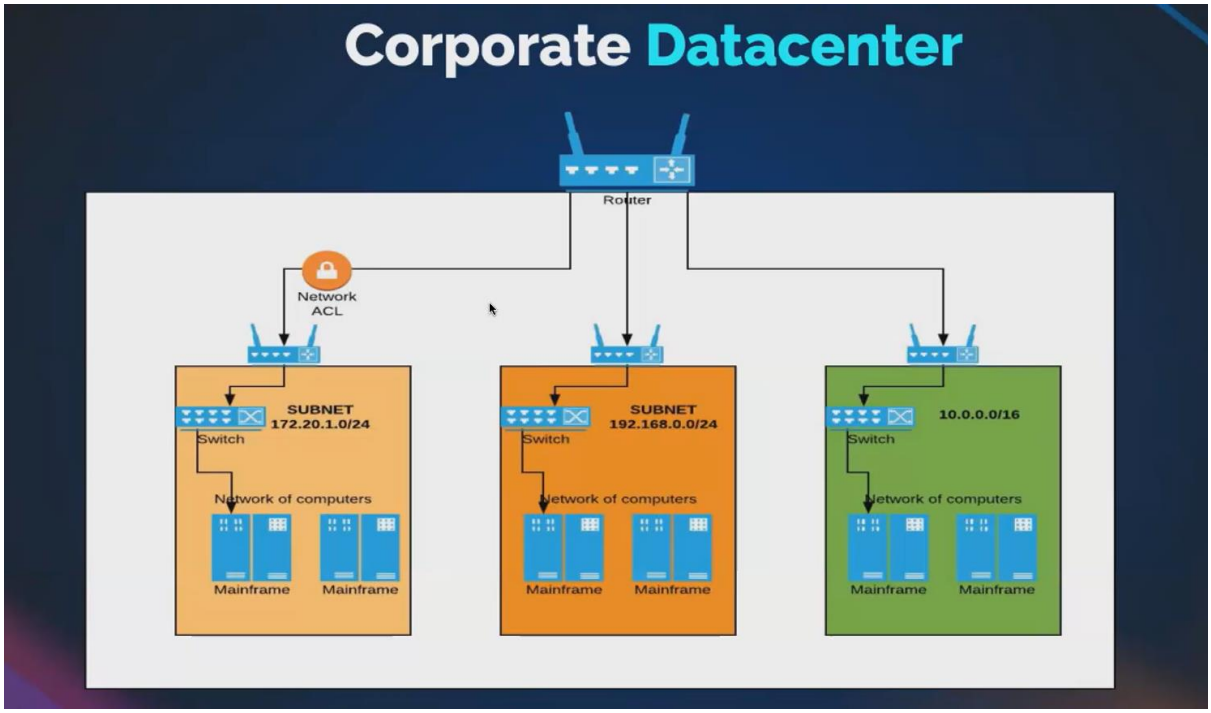
VPC:

- **Vpc is a virtual private cloud**
- **We just logged into our aws console and we straight away. We created the resources. We launched S3 bucket. We launched ebs, we launched Ec 2 .**
- **What your aws will do. It will launch your resources in the default Vpc. Only if you see your Ec2 or anything, if you go and check the Vpc. It would have created in the default.**

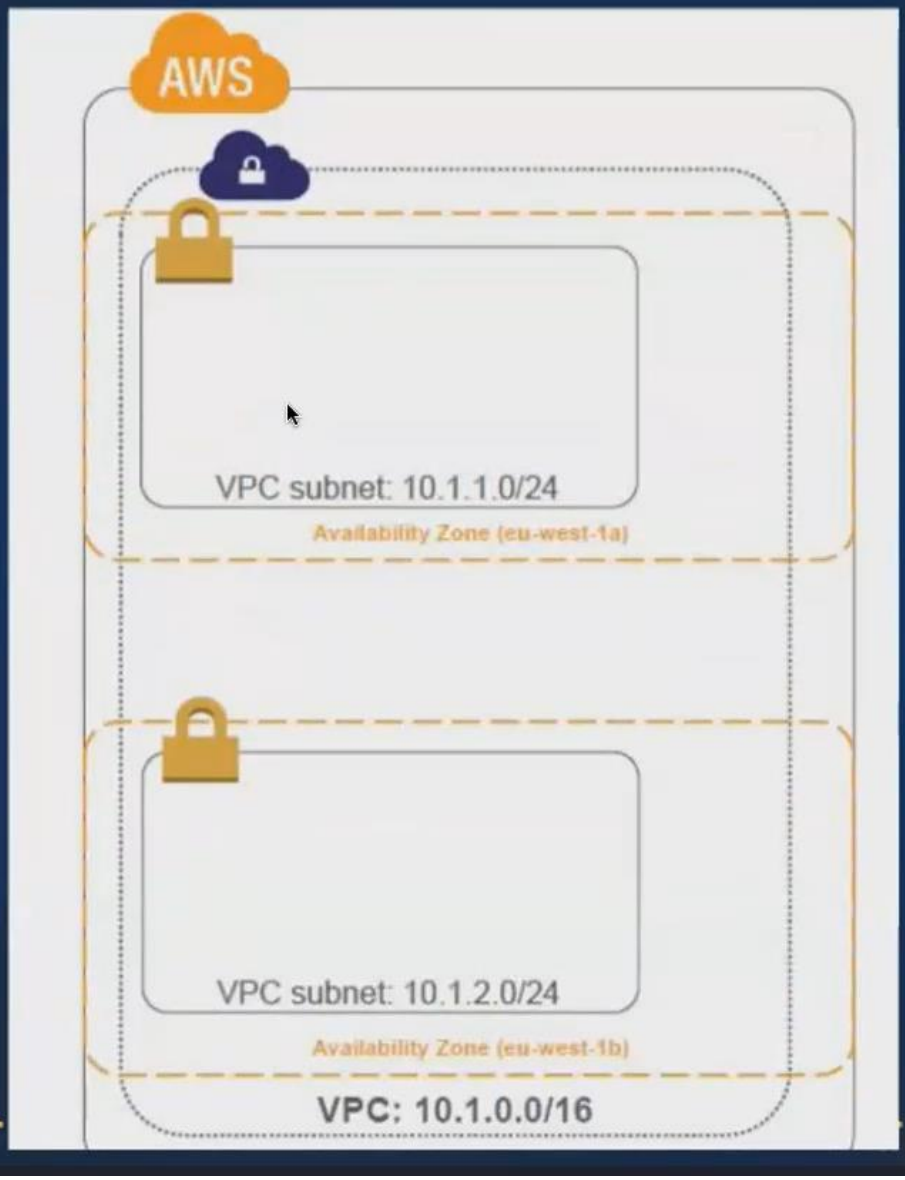
Why we are going for vpc?

- you have 2 applications, 2 different organizations. They create their Ec 2 machine. They wanted to launch their Ec2 machine in the same availability zone.
- **So their privacy has been reduced. So you're sharing your application. If one application is hacked, another application will also be hacked. So in order to avoid all those things, to ensure security, every organization start by creating before launching any resource, they start by creating a Vpc.**
- vpc is like a <u>**logical separation**</u>**. Okay, so it is an isolated within your public cloud. You have a Aws cloud which is public**
- **You're creating a logical isolation for your resources. You create your isolator a small part in your aws, cloud, and launch all your aws resources within that area.**

- **you can define your own Vpc IP address. You can create your own subnets.**
- **Network divided into subnets. Each subnets have own Ip,Switches and all .**


Corporate Datacenter
Router
Network ACL
SUBNET 172.20.1.0/24
Switch
Network of computers
Mainframe
Mainframe
SUBNET 192.168.0.0/24
Switch
Network of computers
Mainframe
Mainframe
10.0.0.0/16
Switch
Network of computers
Mainframe
Mainframe

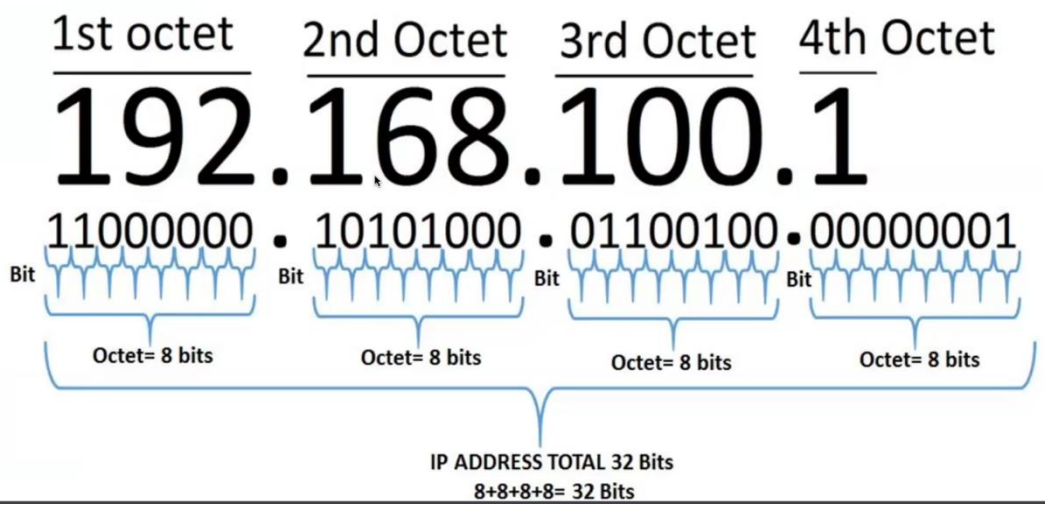
VPC Network

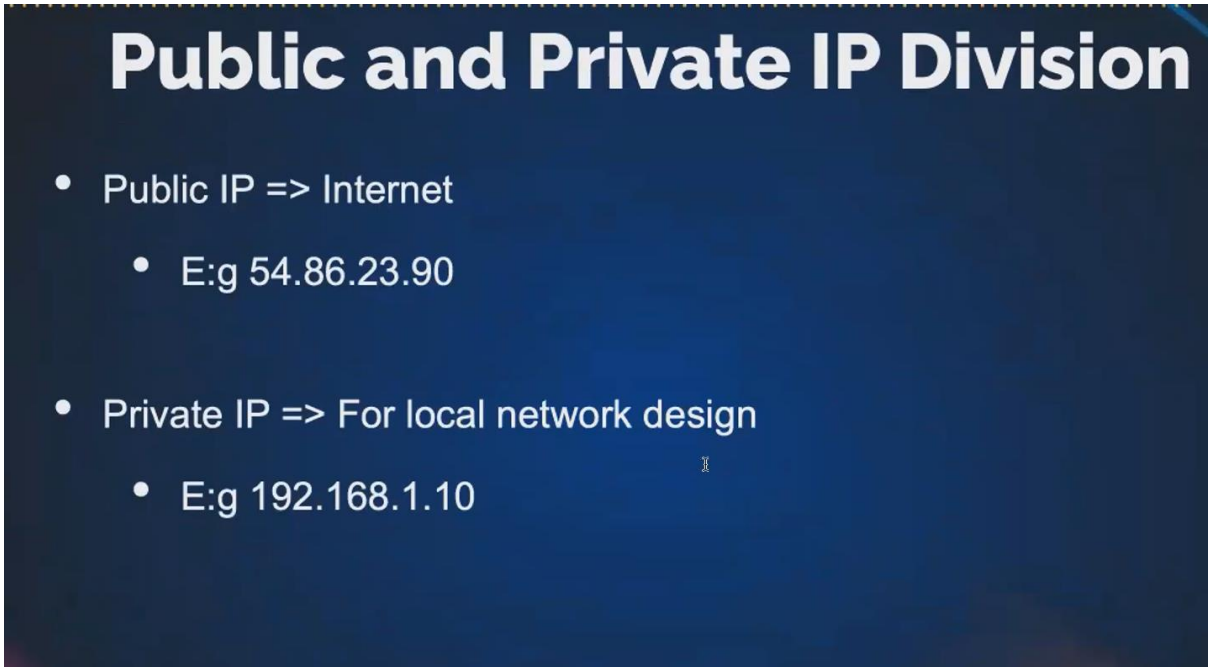

AWS
VPC subnet: 10.1.1.0/24
Availability Zone (eu-west-1a)
VPC subnet: 10.1.2.0/24
Availability Zone (eu-west-1b)
VPC: 10.1.0.0/16

Virtual Private Cloud (VPC)

- VPC is a logical data center within an AWS Region.
- virtual private cloud is an on-demand configurable pool of shared computing resources allocated within a public cloud environment.
- Control over network environment, select IP address range, subnets and configure route tables and gateways.

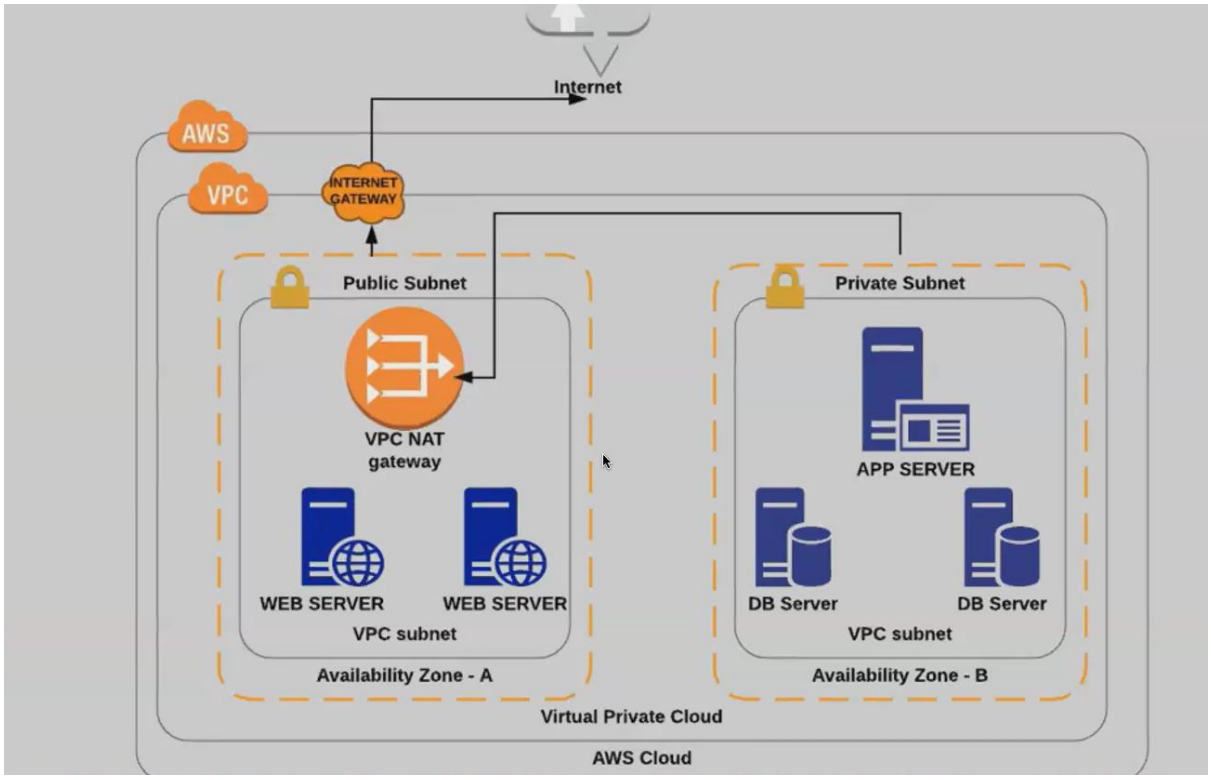
IPv4 Address





Aws will decide by default. It will assign the IP address for your ec2.

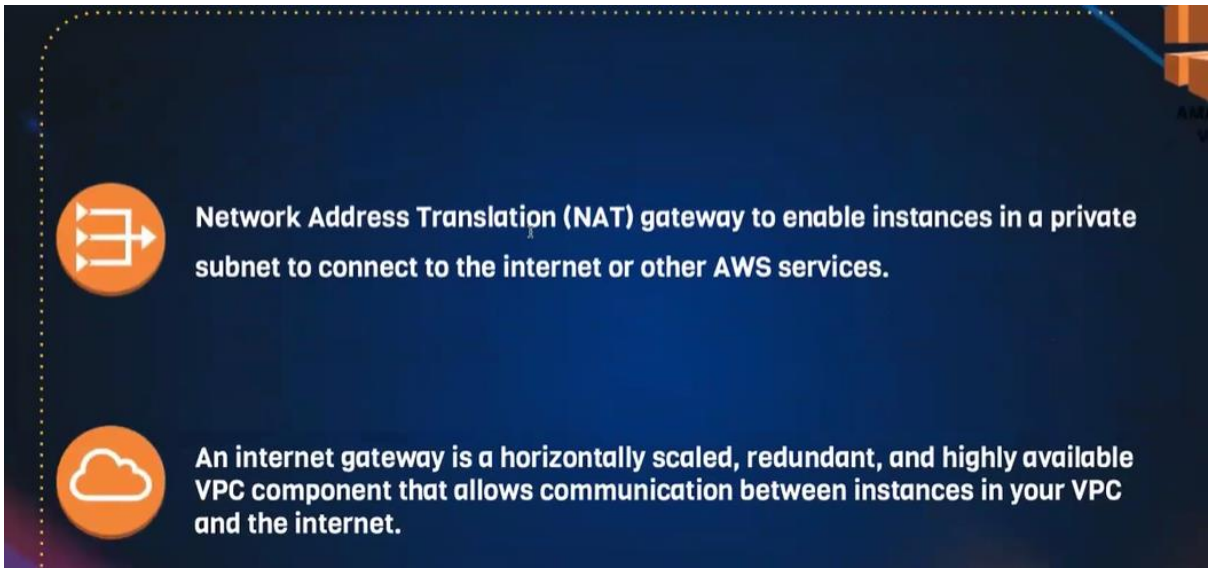
when you're creating your own Vpc. You can decide the IP address, range, and how many IP address.



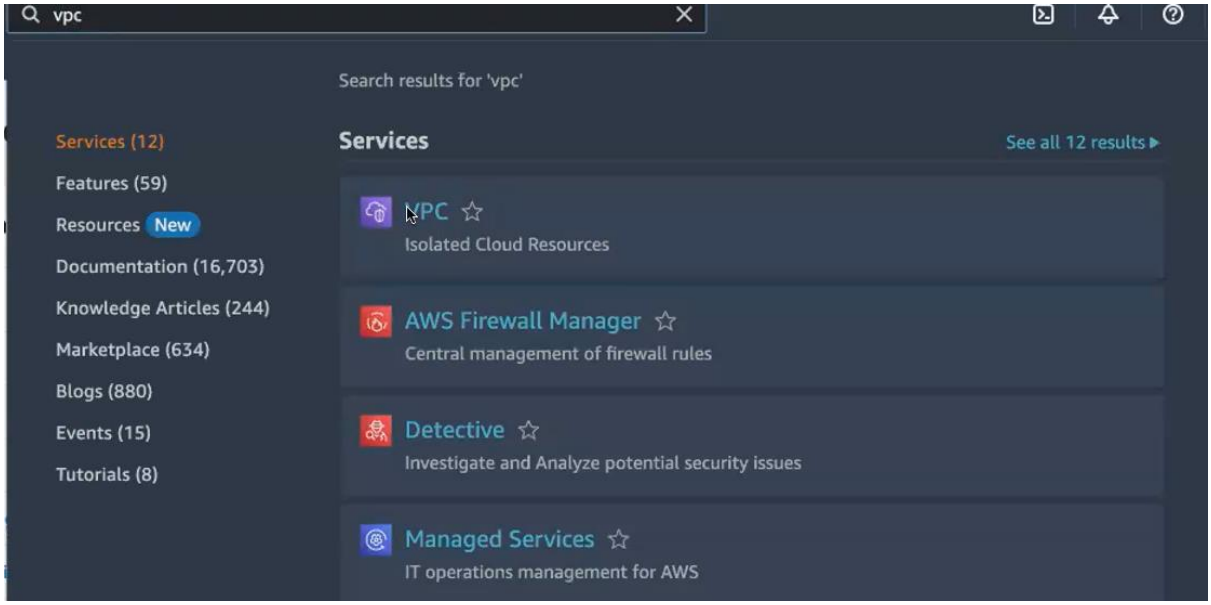
- **Aws cloud within the Aws, we launch the Vpc within Vpc, you launch subnets, you can launch multiple subnets. So you can launch either a public subnet or a private subnet.**
- **If you're launching a public subnet, public subnet can connect with the Internet.**
- **Public subnet means the resources that we are launching inside the public subnet can talk to the Internet how they can talk.**
- **They can talk with the help of Internet gateway. So you have to launch a component called <u>Internet Gateway</u>.**
- **So with the help of Internet gateway, the resources that we have placed inside the public subnet.**

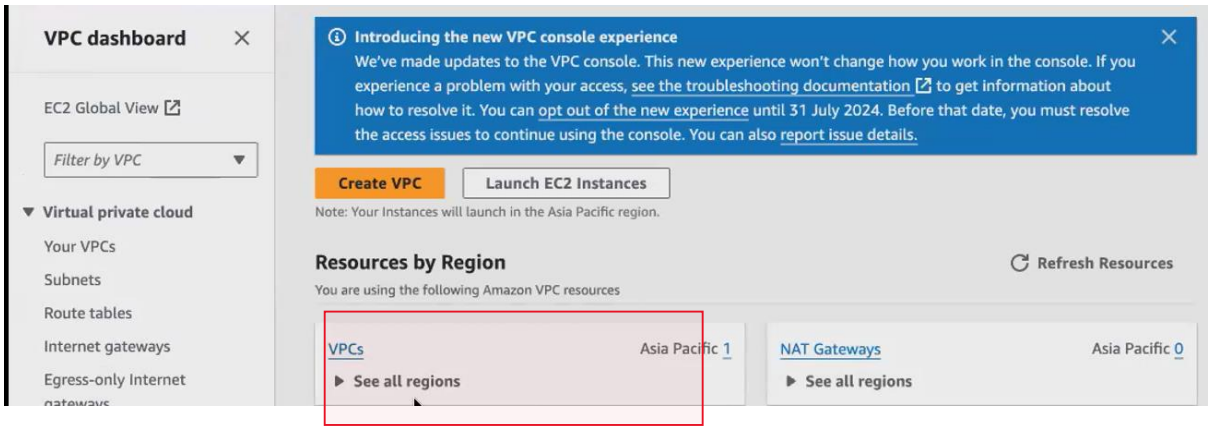
- **certain resources you want to restrict from the outside world. So those resources you place it inside private subnet. for example: web application**
- **in web application. You have a front end. You have an application layer, and you have a database layer. Obviously, front end has to communicate with the outside world. Okay, where in your application layer, where you write all your logic, your database layer.**

- everything has to be kept private, so you cannot. You cannot allow outside world to access your applications. Logic for the database. So we place our database server and the application server inside the private subnet, so that no anyone through the Internet cannot access your resources, that if we are placing it inside the private subnet.
- if you want to connect. So here you only have the outbound traffic, which means, if my database wants to do a update or do a patchwork in that case, it has to connect with the Internet. So during that time only your in private subnet can connect with the Internet that do not directly it has to connect with an At gateway and via Nat Gateway. You will be connecting to the Internet gateway. Nat Gateway means network address translation.
- So this Nat gateway. What it will do. It will hide your resources. That is your database Ec2. It will not expose the IP address of your database to machine. Instead, it will it will mask the IP address, and it will connect with the Internet, so that just to protect us for the security reason.
- if you're using a Nat gateway, your address will be translated, it will assign a elastic IP address, a different IP address will be assigned, and it will be communicating to the Internet.
- 2 types of gateways:
- 1. Nat gateway
- 2. Internet gateway

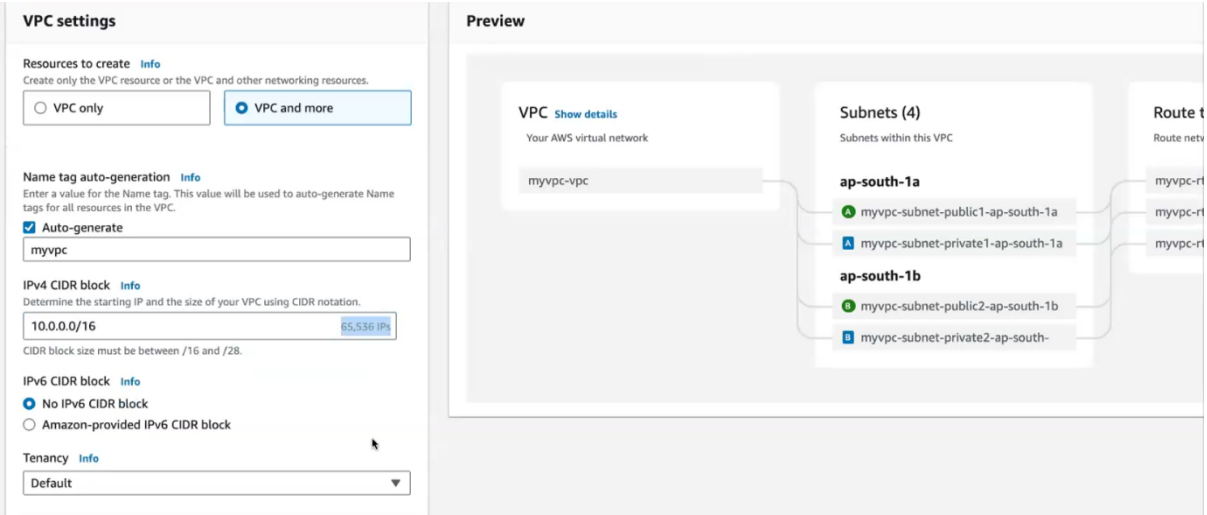


How to create VPC:



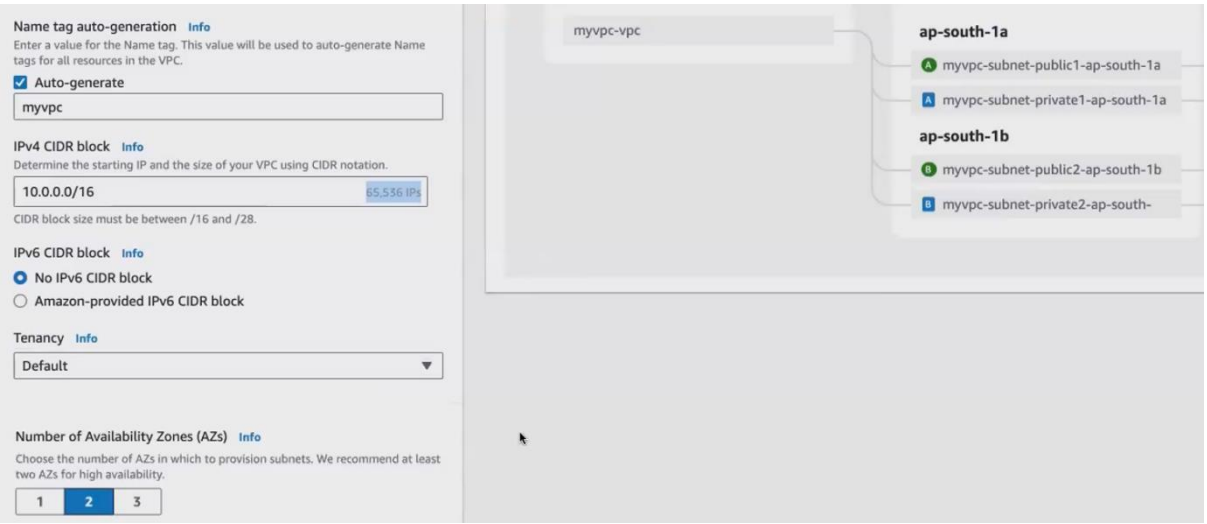


This is default vpc. But we create vpc



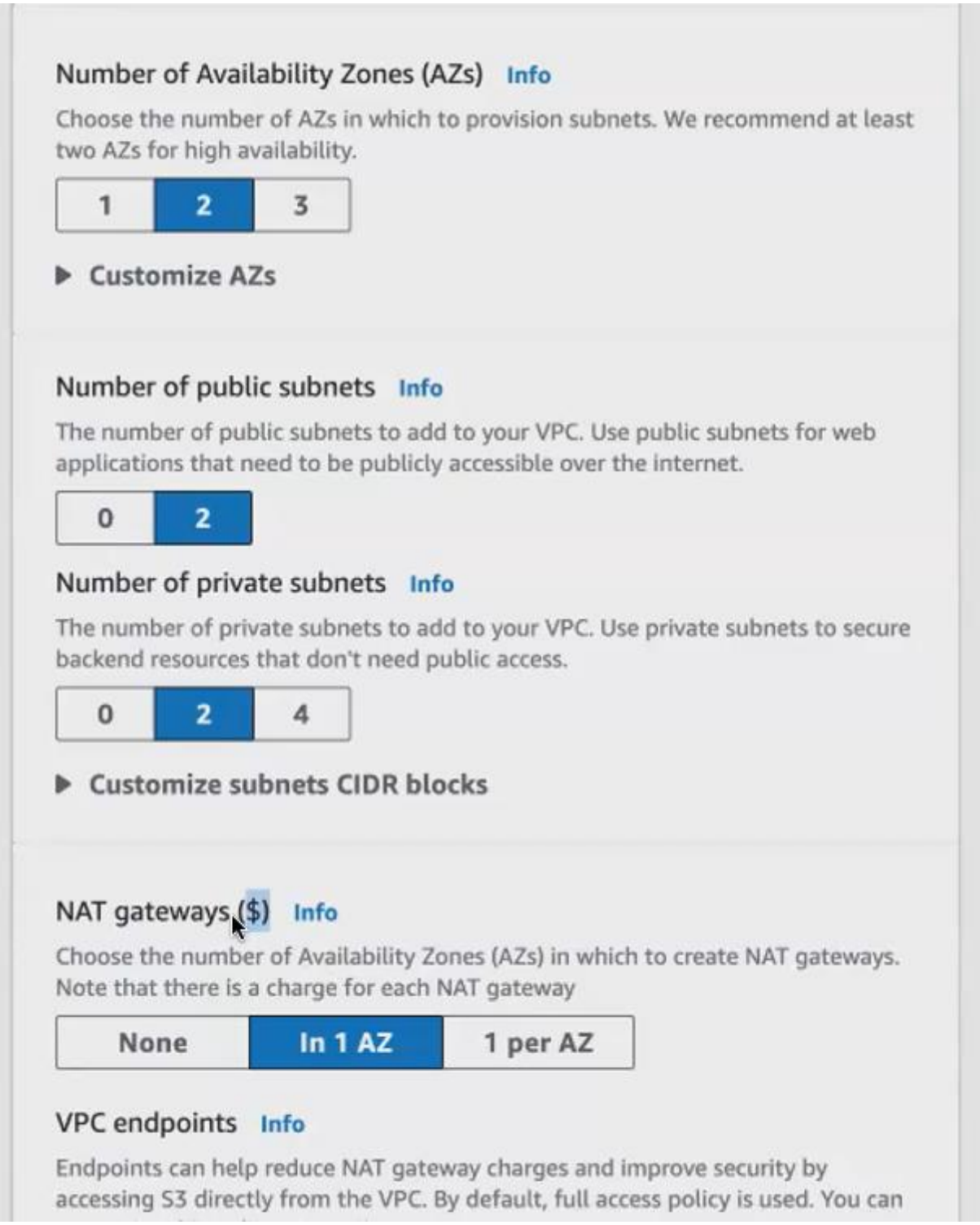
10.0.0./16 – 65,536 ip add created

Myvpc -name of my vpc



AZ 2 means 2 public and 2 private n/w created

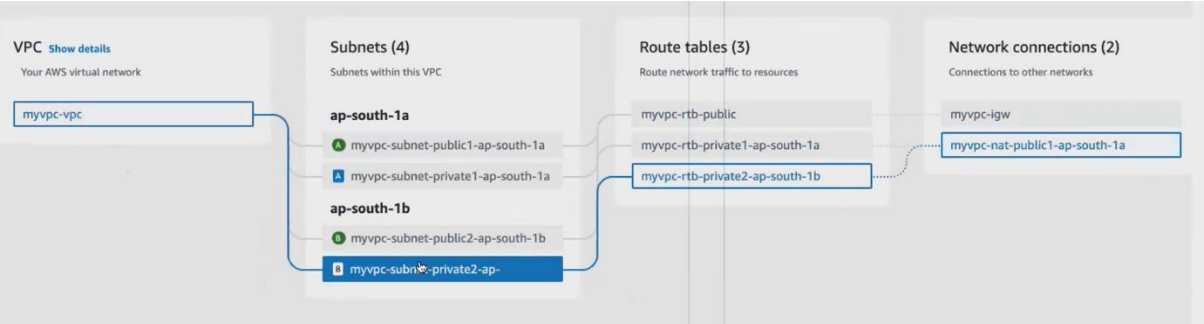
AZ 3 means 3 public and 3 private n/w created



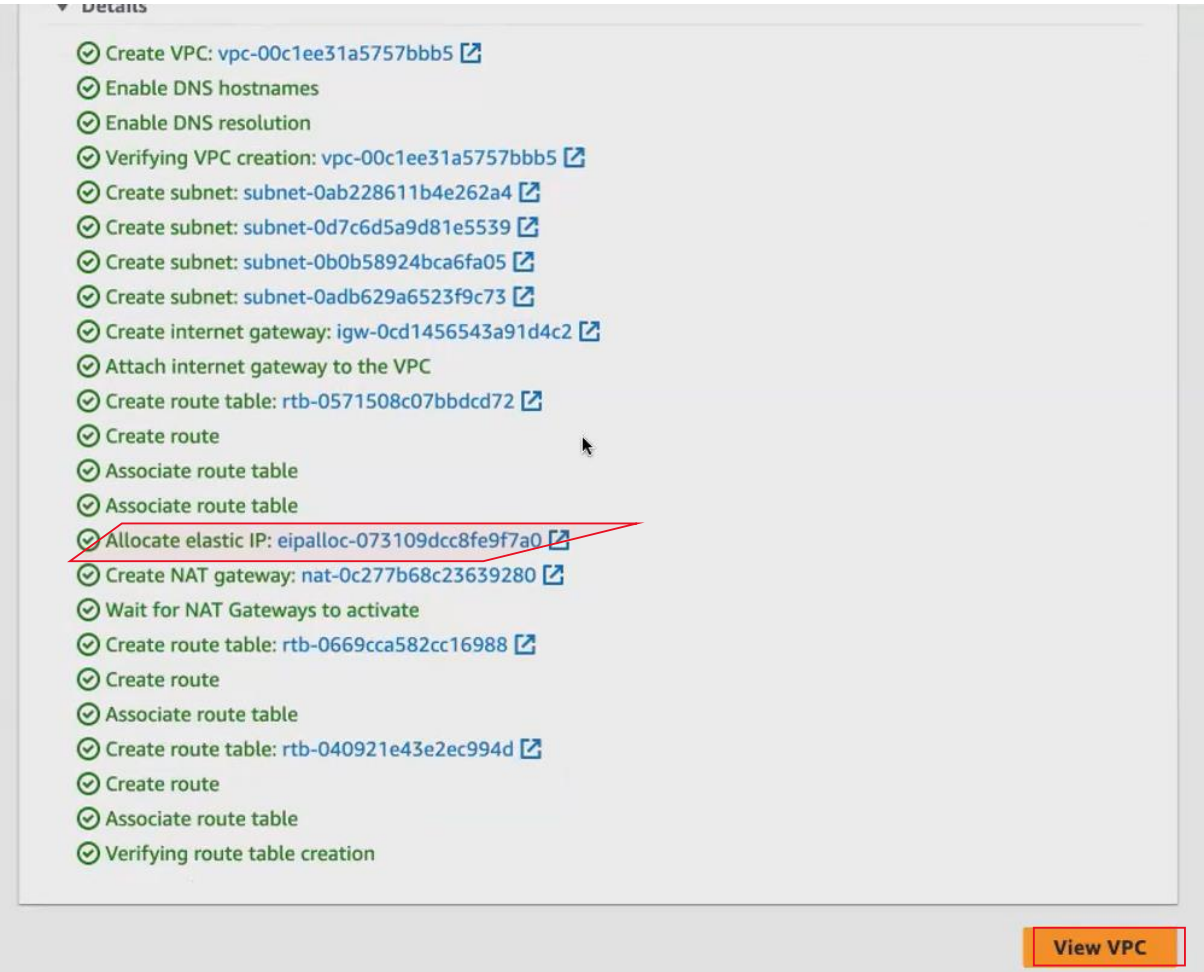
Nat gateway chargeable so select as 1 AZ

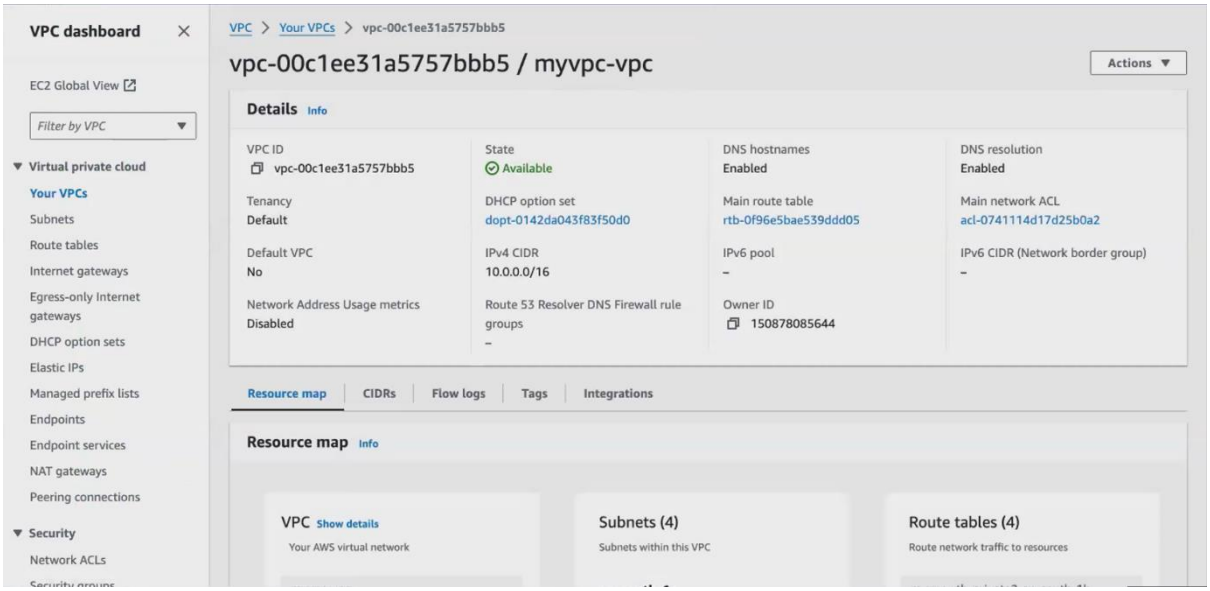
Vpc endpoint give none.

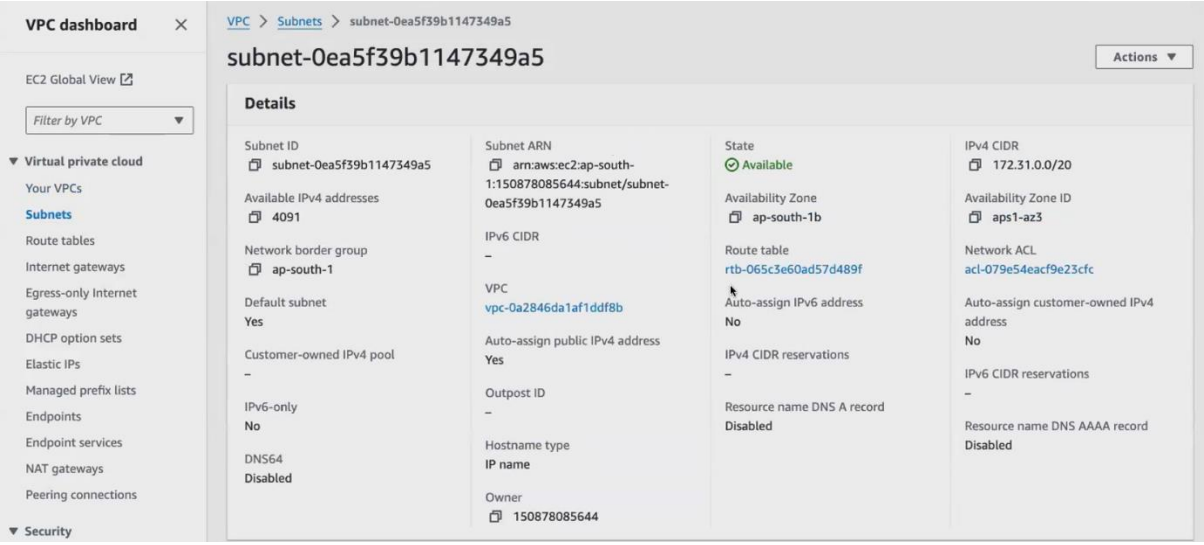
This is my vpc network architecture



Click create VPC all resources install one by one manually. Elastic ip is optional. Nat is enough

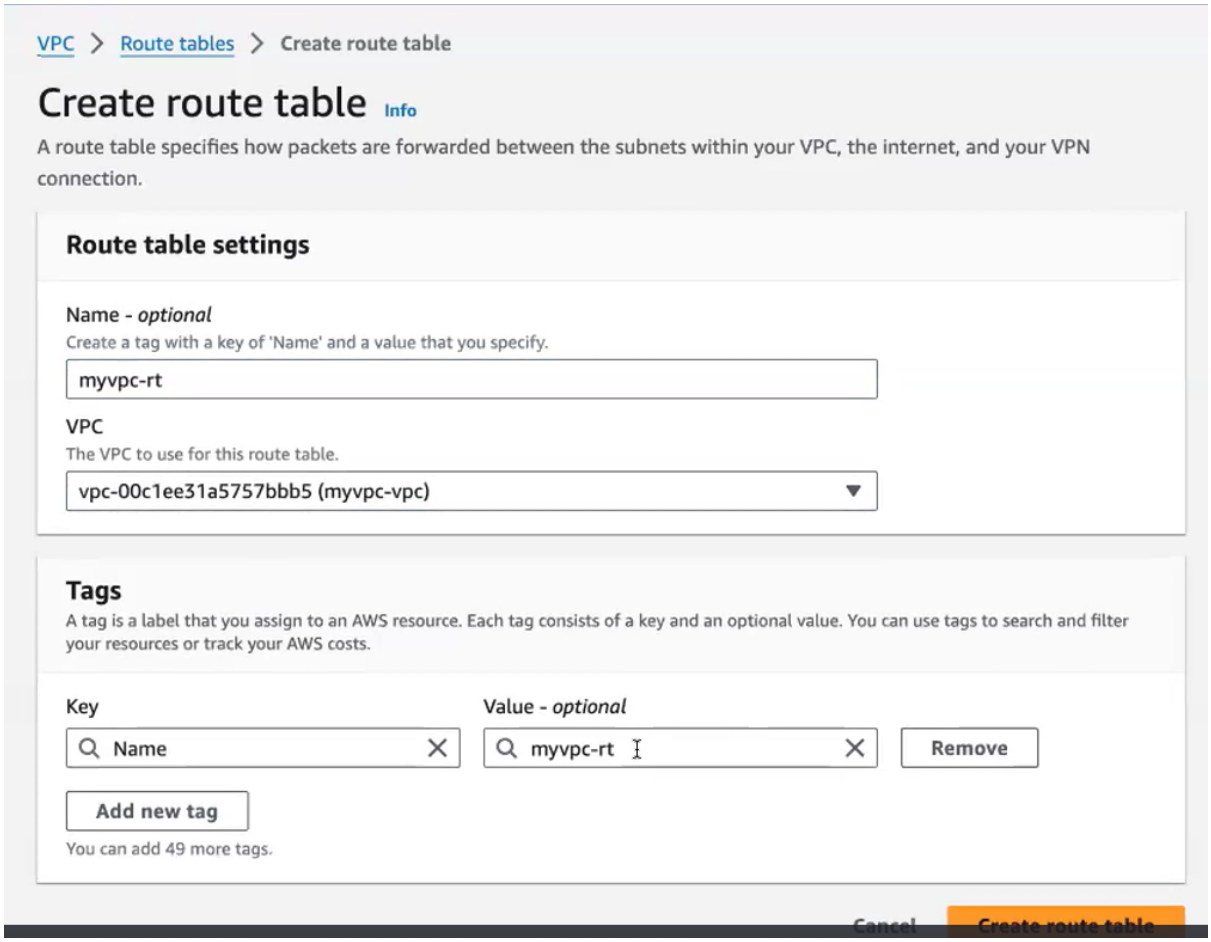






Create route table to associate our public subnet to internet gateway

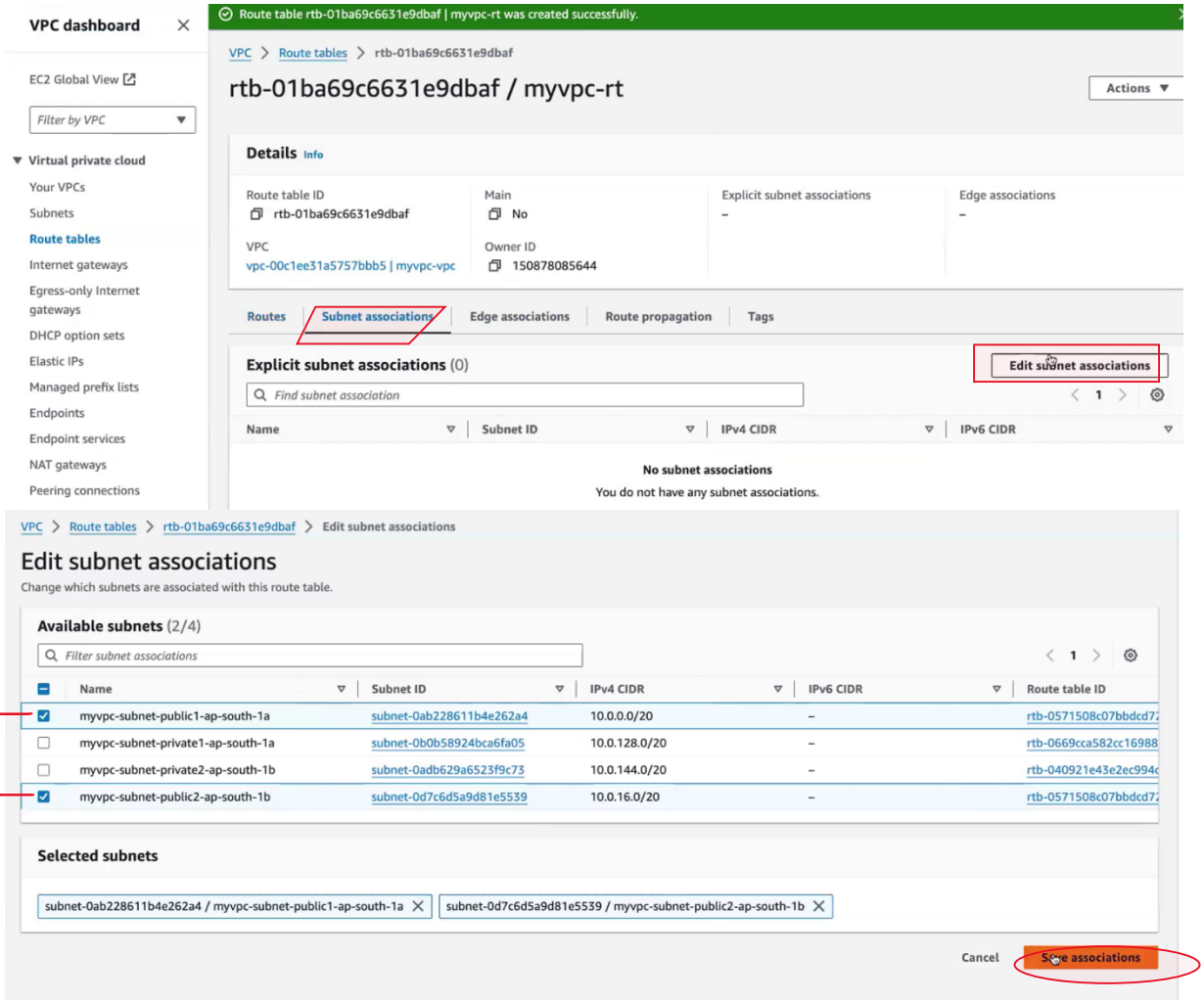
- now you have all the components separate. So you have your public subnet ,Internet gateway, private subnet, Nat Gateway. So we will be connecting all these things via Route Table.



Myvpc-vpc name my vpc

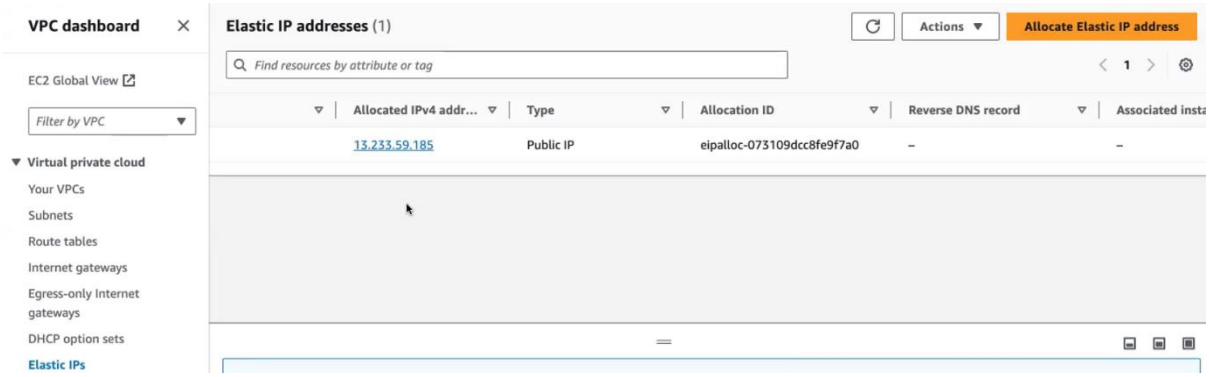
Now r.t created

How-to associate with public subnet



we'll be creating an NAT gateway. So when you create an NATgateway elastic IP will be created. So your eip means elastic IP address will be created. So whenever you're connecting your private, the resources with your Nat Gateway.

it will communicate will only expose the elastic IP address rather than exposing the private Ec2. Our private ip add now shown .only elastic IP address will show



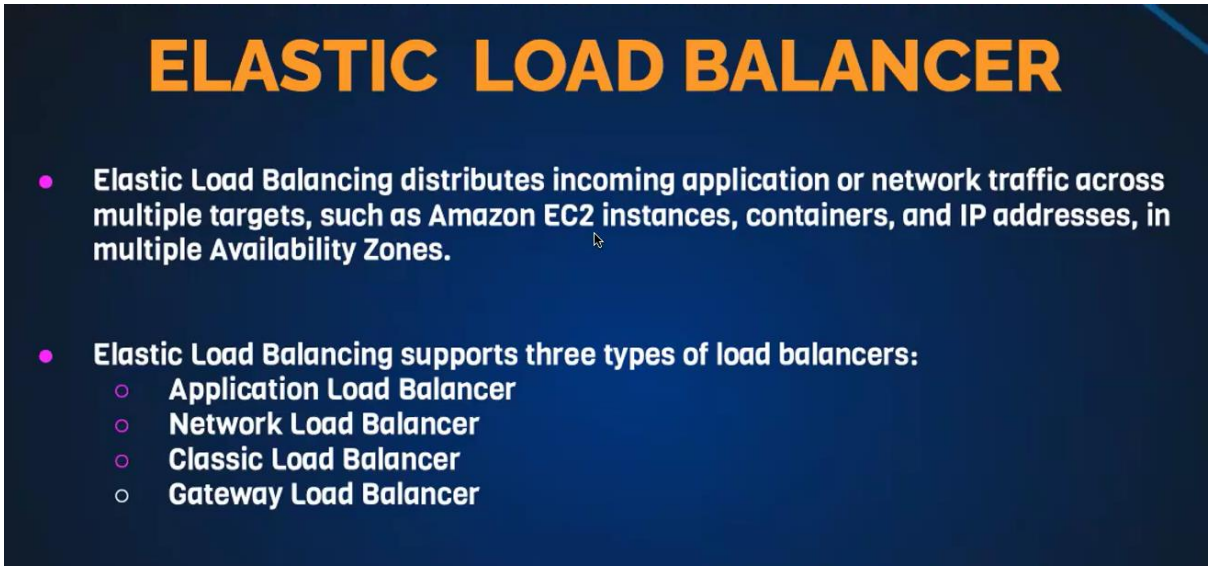
Elastic load balancer:

- **aws elastic load balancer needs. So which distributes the traffic to your resources. So, for example, you're hosting application. You're hosting a web-based application in your Ec2. Mission.**

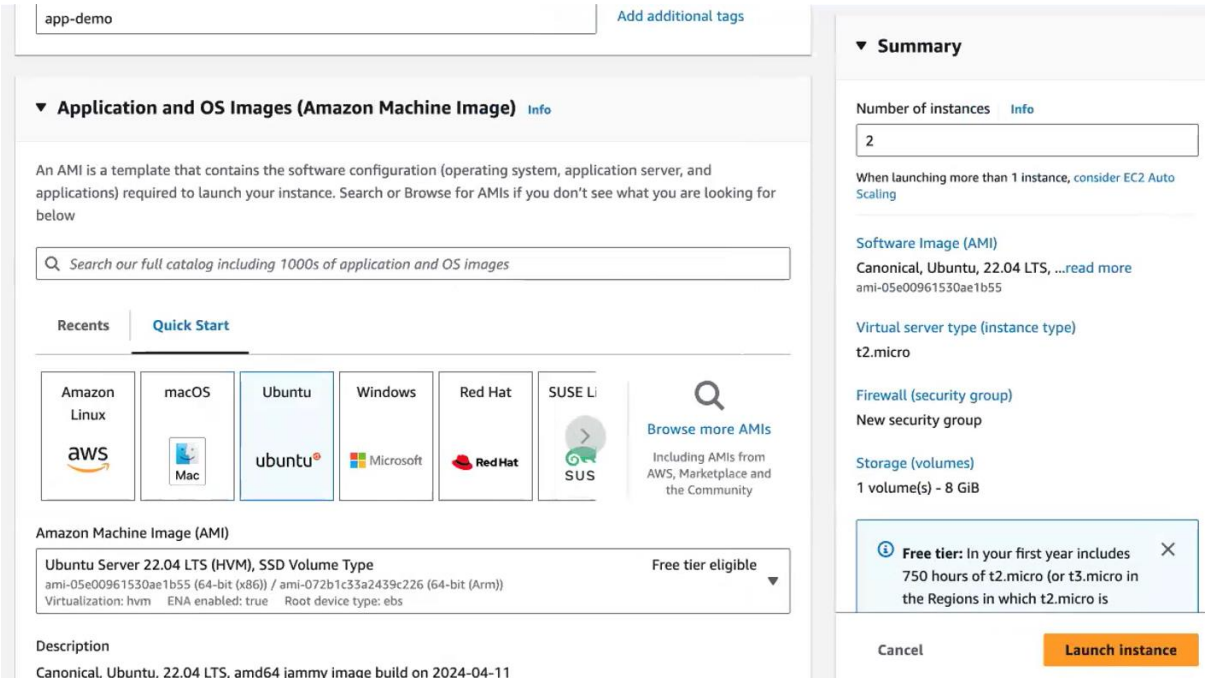
- **So you have some 10 Ec2 running which is hosting your application. You have a e-commerce website. And this e-commerce website is hosted in those 10 Ec2. You have user requests coming in to access the application.**
- **your client request will be coming in to access the application. So if, for some reason, if the traffic is high, so you have all of a sudden your application demand grows. So your user size also grows. So what happened? Your app your request? All the request goes to your Ec2 .so your aws will not know to how to route the traffic among all your Ec2 your aws will not know, aws randomly sends the traffic to any one of your Ec. 2**
- **you have 10 Ec2 running, so you have set the load for all the Ec2. So one Ec2 will be capable of handling only this much amount of load. It can only handle 500 requests per second.**
- **if the client who have you? You, you have some 100 requests, or you have some 1,000 requests hitting the same ec2 per second.**
- **So what will happen? Your Ec2 will go down it. It is not capable enough. It is not build enough to handle that much amount of load or that much amount of request. In that case, what we do, we build a load balancer in front of your Ec2. So you have a load balancer, and behind the load balancer, you create your service. You create your Ec2 servers, and inside the Ec2 servers, you host your application.**
- **you in front of all these servers you have your load, balancer. So when the clients sends the request, it will hit your load, balancer.so your load balancer, will get the request from your client, and it will distribute equally among all your Ec2**

- **so it will distribute the traffic. It will break down the traffic, and it will distribute it equally among all your Ec2. So instead of one machine receiving all 1,000 requests, every machine will receive 100 requests.**
- **So what will happen. Your machine can also respond quickly.**

- **We go for elastic load balancer. <u>Elastic load Balancer means it can scale up and scale down.</u>**
- **based on the applications requirement. Your aws can scale the traffic, and it can scale down the traffic also.**



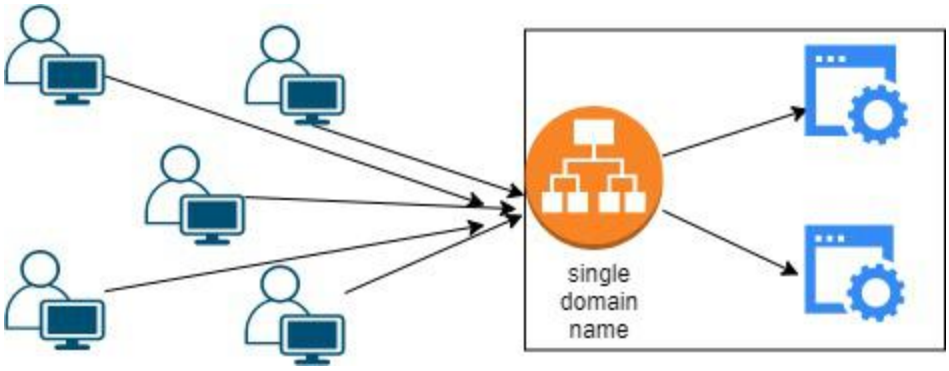
before launching the load balancer, will launch some 2 Ec2.



https://docs.google.com/document/d/1abBZAHj0Mq5RFIb8v3lwO_fOXgqCWpquEF3-GjuK2S4/edit

Elastic load balancer

Elastic load balancer is a service provided by Amazon in which the incoming traffic is efficiently automatically distributed across a group of backend servers in a manner that increases speed and performance.



Application Load Balancer - HTTP and HTTPS traffic routing
This load balancer works at the Application layer of the OSI Model
Network Load Balancer - This type of load balancer works at the transport layer(TCP/SSL) of the OSI model.
Gateway Load Balancer - It's capable of handling millions of requests per second.

Launch ec2

Userdata amazon linux ec2

```
#!/bin/bash
sudo yum update -y
sudo amazon-linux-extras install nginx1 -y
sudo systemctl enable nginx
sudo systemctl start nginx
```

```
Ubuntu

#!/bin/bash
sudo apt update -y
sudo apt install nginx -y
sudo systemctl enable nginx
sudo systemctl start nginx
```

```
echo "<h1>Hello World from $(hostname -f)</h1>" >
/var/www/html/index.html
```

Or

```
#!/bin/bash
sudo apt update
sudo apt install apache2 wget unzip -y
wget https://www.tooplate.com/zip-
templates/2132_clean_work.zip
unzip 2132_clean_work.zip
sudo cp -r 2132_clean_work/* /var/www/html/
sudo systemctl restart apache2
```

Or

```
#!/bin/bash

# Installing Dependencies
echo "##########################################"
echo "Installing packages."
echo "##########################################"
sudo apt-get update > /dev/null
sudo apt-get install wget unzip apache2 -y > /dev/null
echo

# Start & Enable Service
echo "##########################################"
echo "Start & Enable Apache2 Service"
echo "##########################################"
sudo systemctl start apache2
sudo systemctl enable apache2
echo

# Creating Temp Directory
echo "##########################################"
echo "Starting Artifact Deployment"
echo "##########################################"
mkdir -p /tmp/webfiles
cd /tmp/webfiles
echo

wget https://www.tooplate.com/zip-templates/2098_health.zip >
/dev/null
unzip 2098_health.zip > /dev/null
sudo cp -r 2098_health/* /var/www/html/
echo

# Bounce Service
echo "##########################################"
echo "Restarting Apache2 service"
```

```
echo "##########################################"
systemctl restart apache2
echo

# Clean Up
echo "##########################################"
echo "Removing Temporary Files"
echo "##########################################"
rm -rf /tmp/webfiles
echo

sudo systemctl status apache2
ls /var/www/html/
```

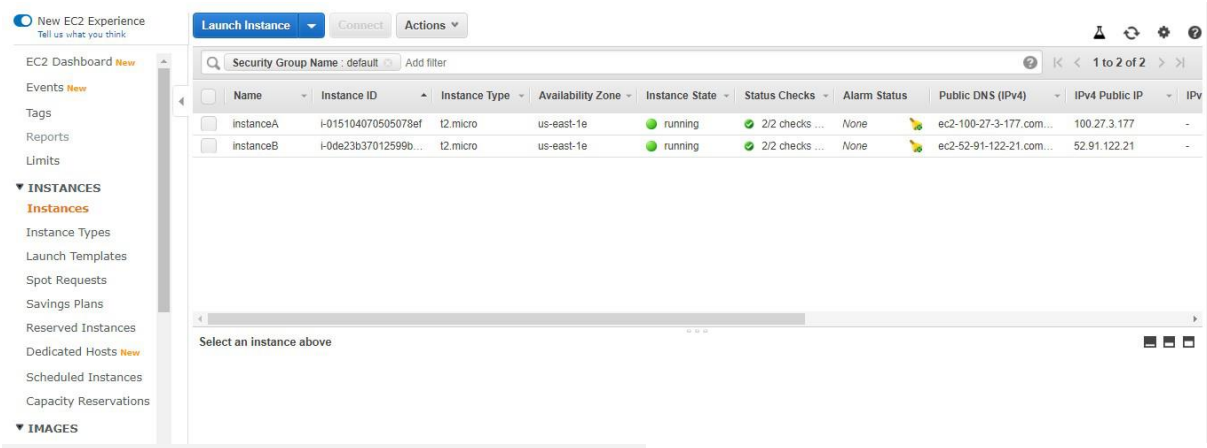
==========
=

```
CentOs / Amazon Linux
#!/bin/bash
# Use this for your user data (script from top to bottom)
# install httpd (Linux 2 version)
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "<h1>Hello World from $(hostname -f)</h1>" >
/var/www/html/index.html
```

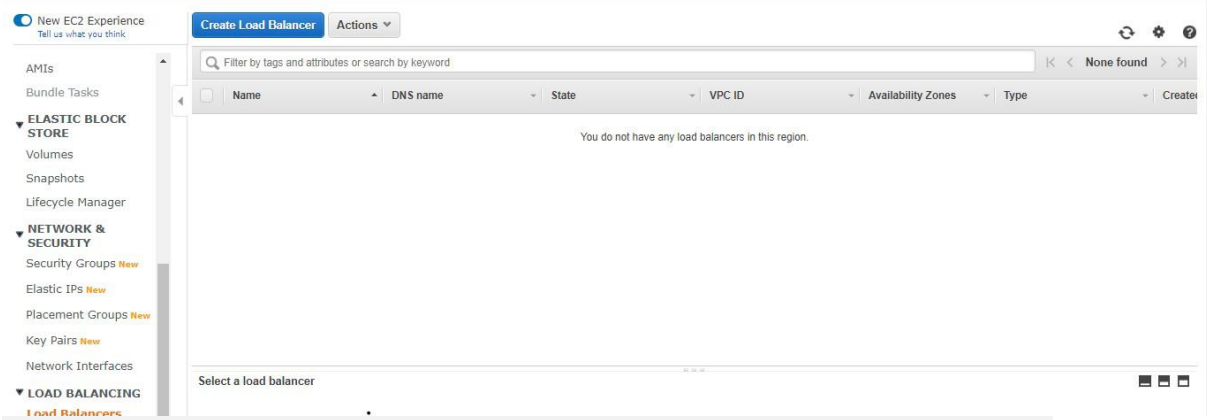
```
#!/bin/bash
#install httpd
sudo yum update -y
sudo yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "<h1>Hello World from $(hostname -f)</h1>" >
/var/www/html/index.html
```

Steps to configure an Application load balancer in AWS:

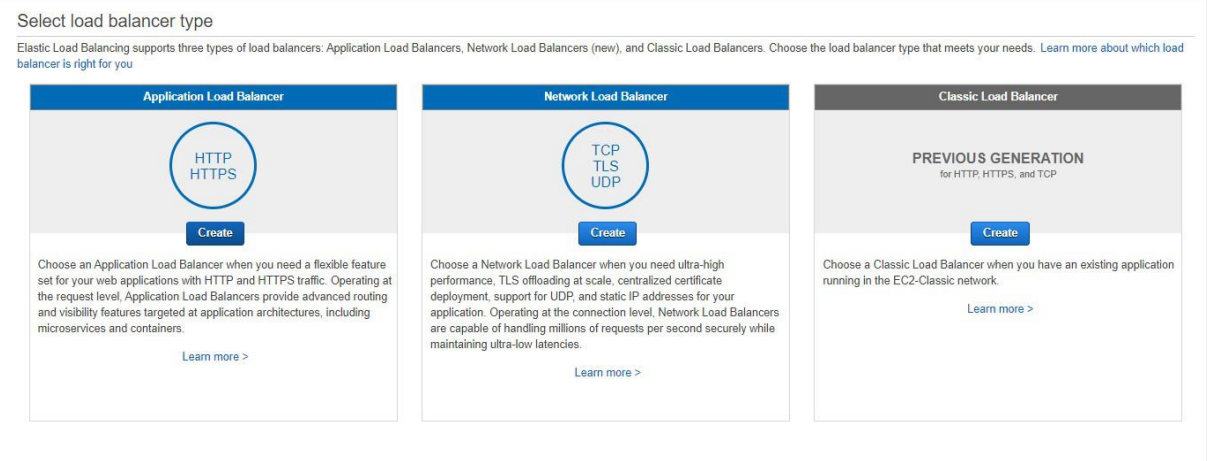
Step 1: Launch the two instances on the AWS management console named Instance A and Instance B. Go to services and select load balancer



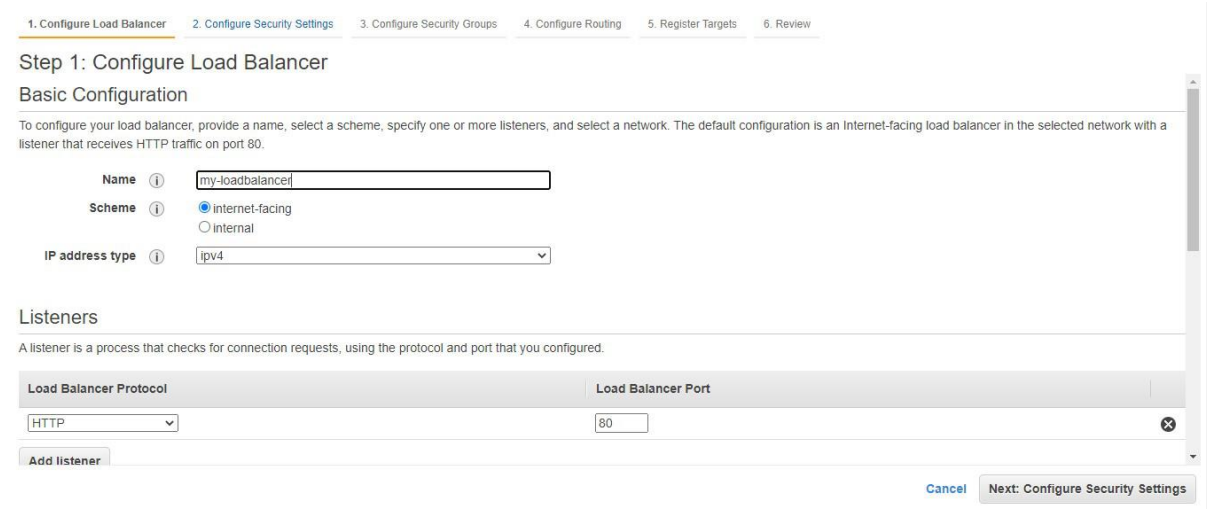
Step 2: Click on create load balancer.



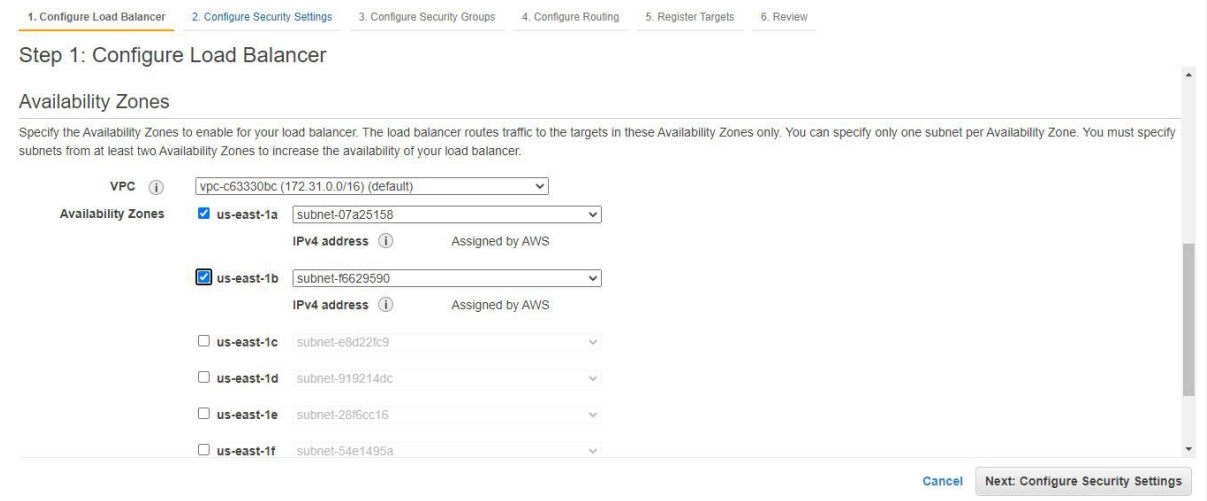
Step 3: Select Application Load Balancer and click on create.



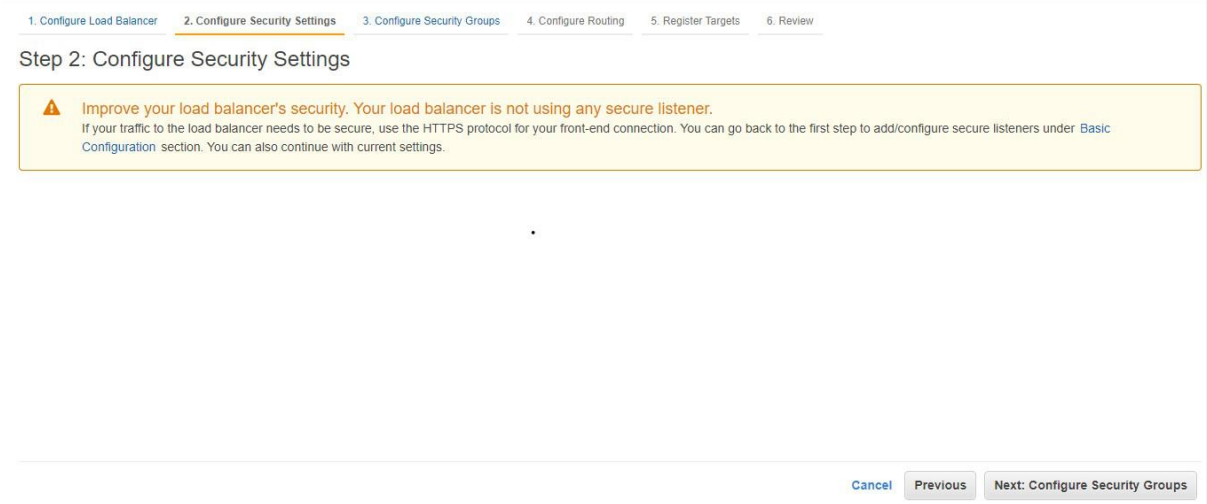
Step 4: Here you are required to configure the load balancer. Write the name of the load balancer. Choose the scheme as internet facing.



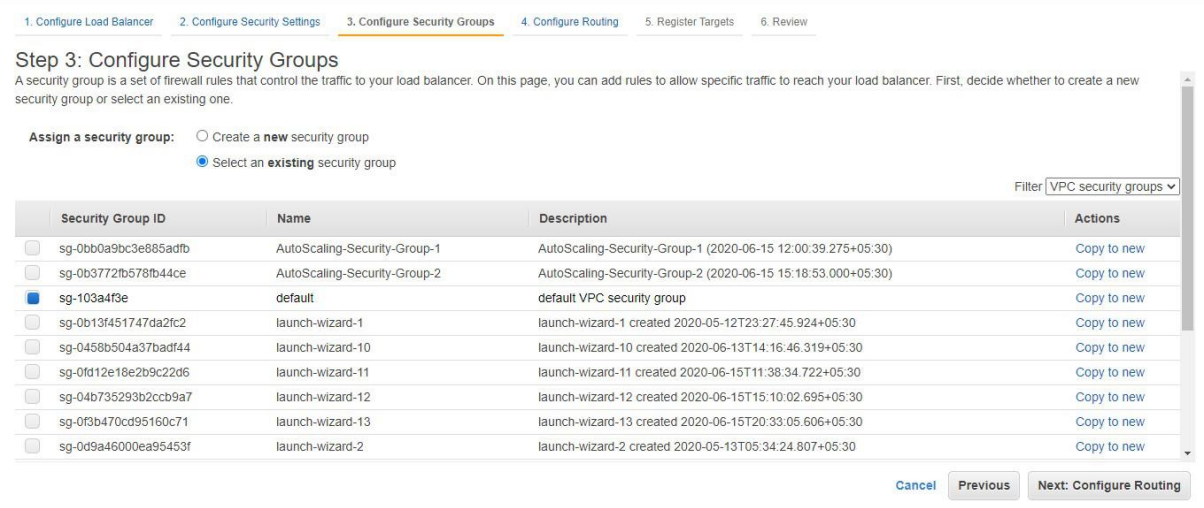
Step 5: Add at least 2 availability zones. Select us-east-1a and us-east-1b



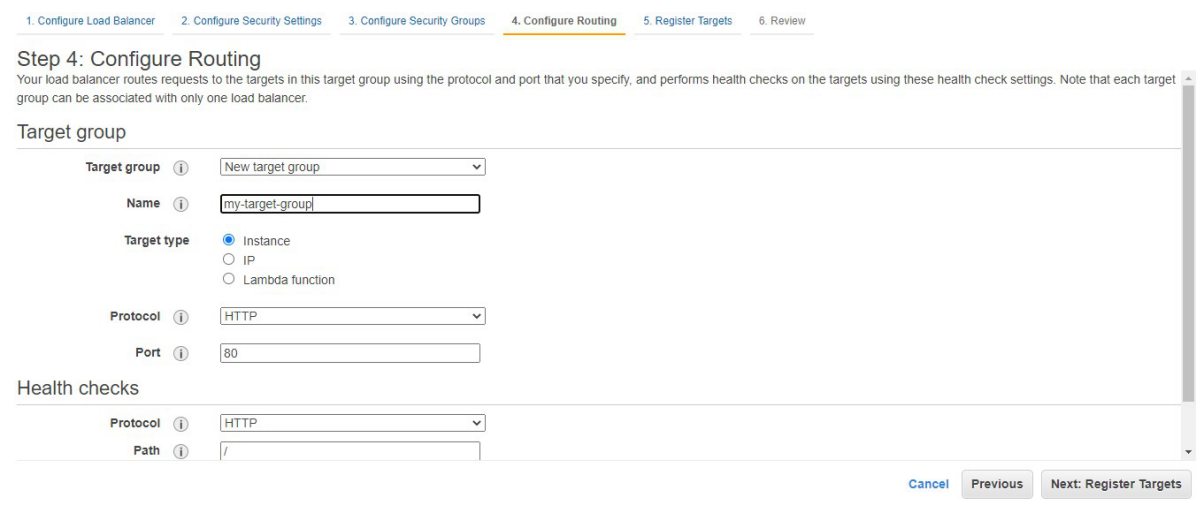
Step 6: We don't need to do anything here. Click on Next: Configure Security Groups



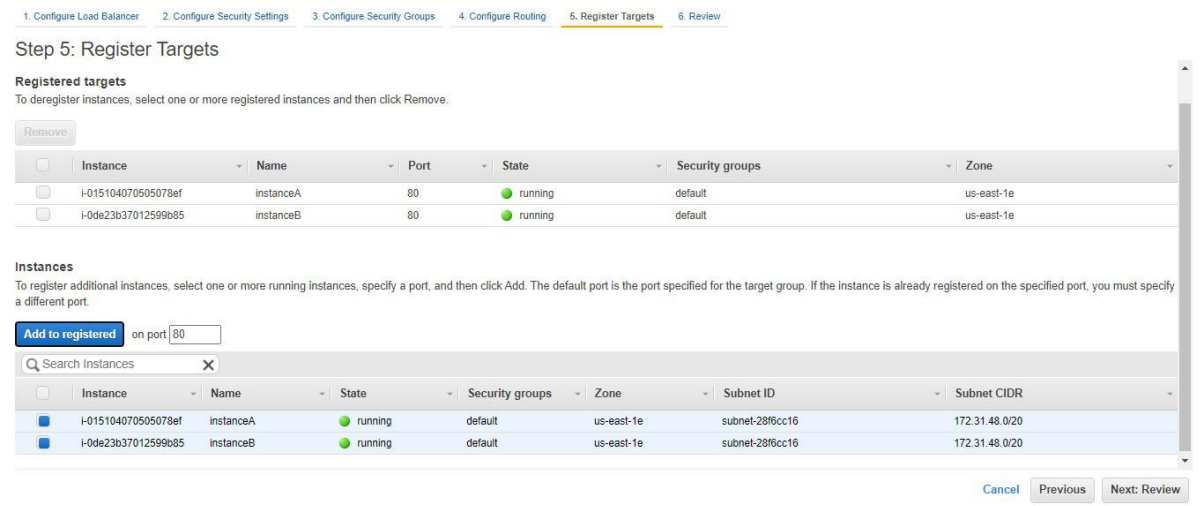
Step 7: Select the default security group. Click on Next: Configure Routing



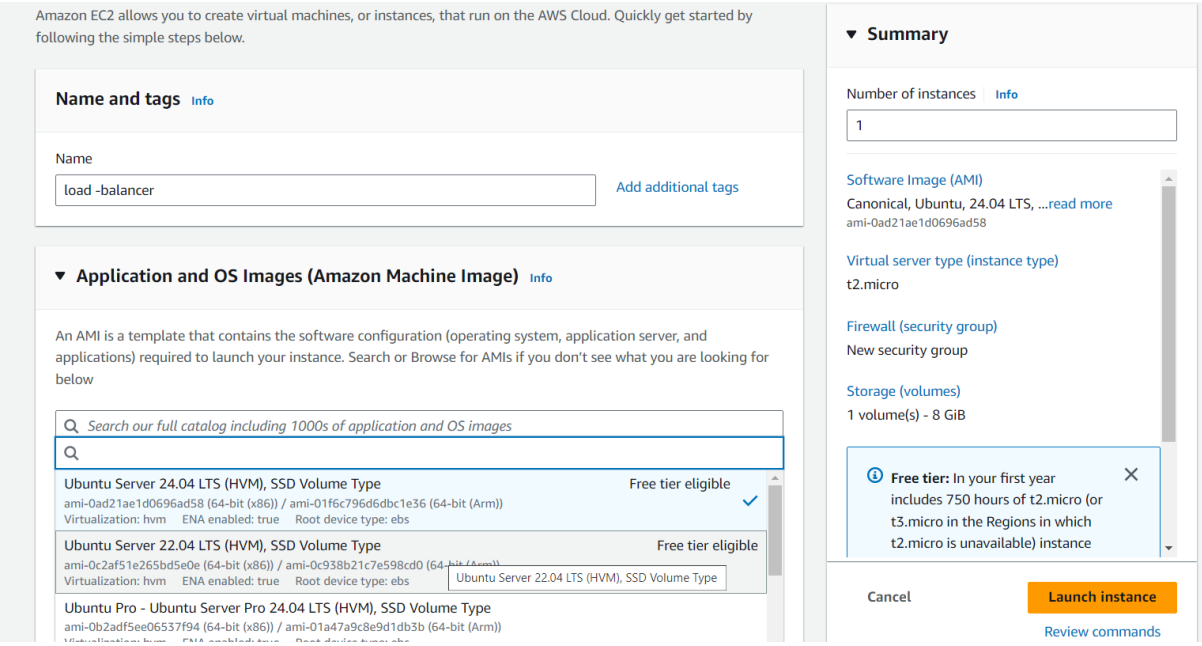
Step 8: Choose the name of the target group to be my-target-group. Click on Next: Register Targets.



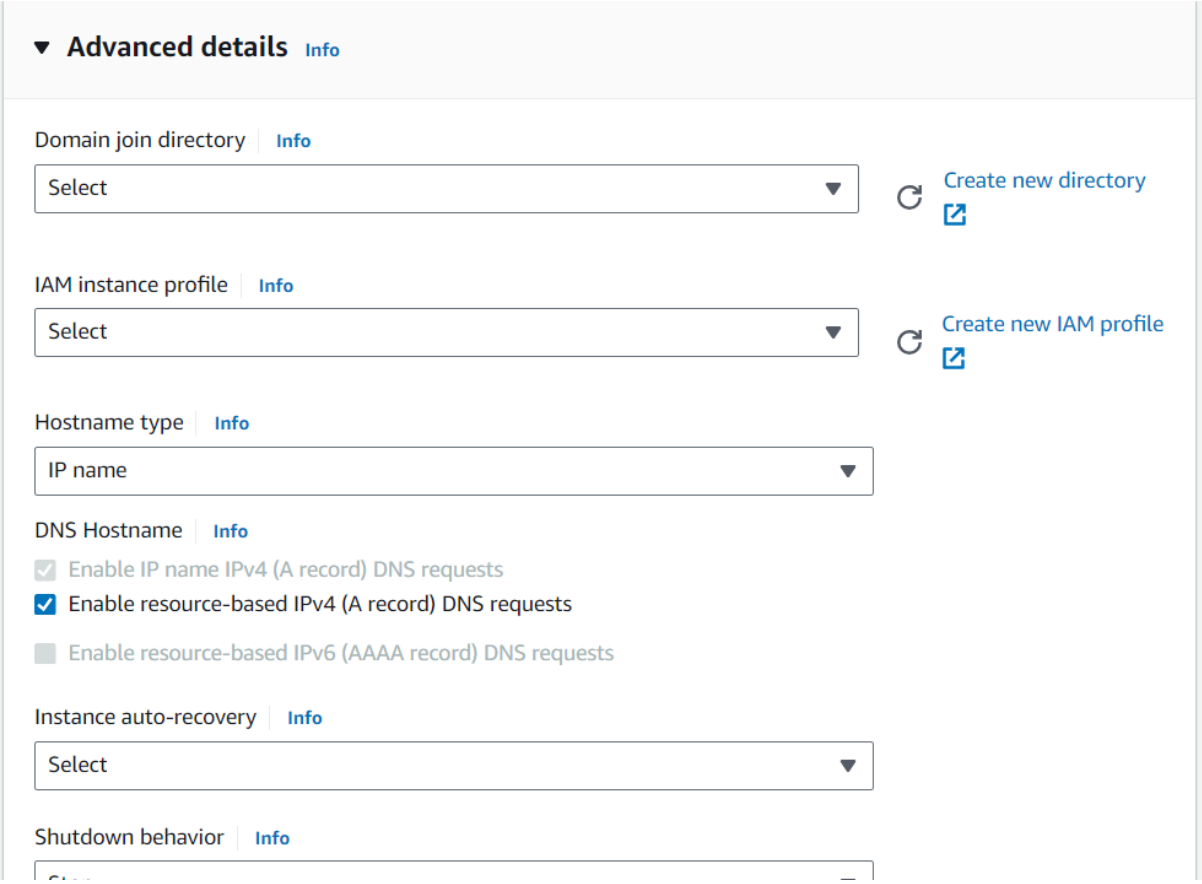
Step 9: Choose instance A and instance B and click on Add to registered. Click on Next: Review.



Step 10: Review all the configurations and click on create



Select ubuntu latest version.click advanced set



docs.google.com/document/d/1abBZAHj0Mq5RFlb8v3IwO_fOXgqCWpquEF3-GjuK2S4/edit

Elastic load balancer

File Edit View Tools Help

Outline

– Steps to configure an Applicatio...

```
sudo systemctl start nginx
echo "<h1>Hello World from $(hostname -f)</h1>" >
/var/www/html/index.html
```

Or

```
#!/bin/bash
sudo apt update
sudo apt install apache2 wget unzip -y
wget https://www.tooplate.com/zip-templates/2132_clean_work.zip
unzip 2132_clean_work.zip
sudo cp -r 2132_clean_work/* /var/www/html/
sudo systemctl restart apache2
```

Copy the bash script and paste it in user data

2

Allow tags in metadata Info

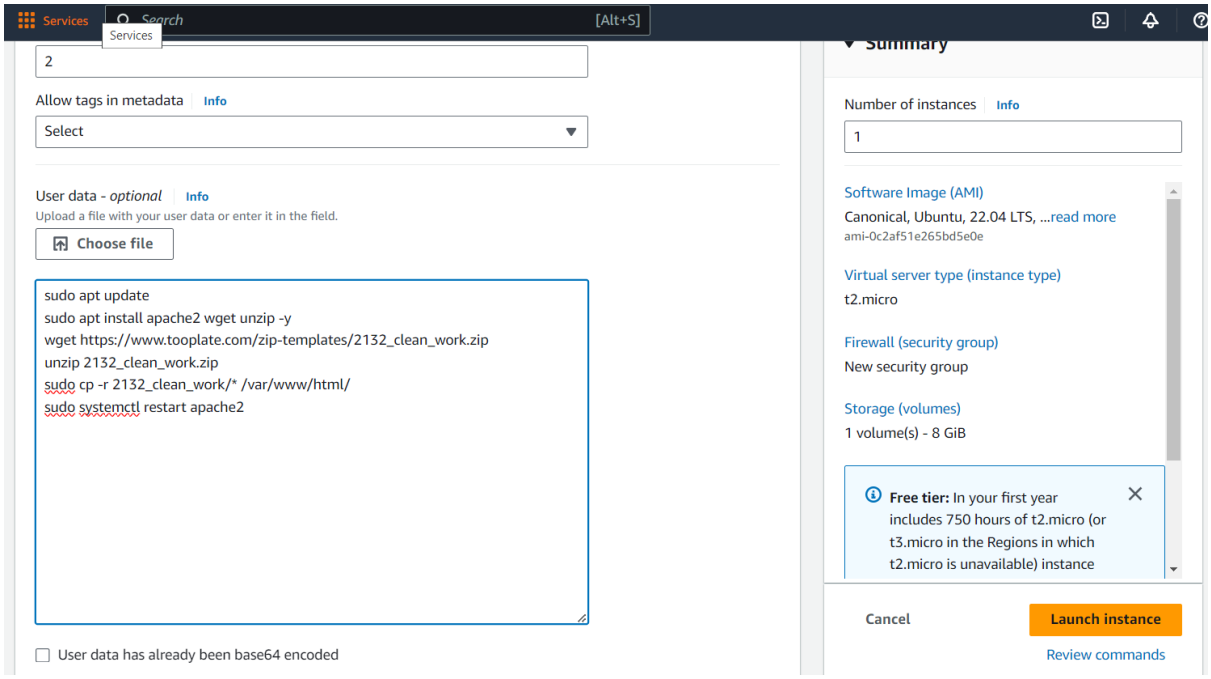
Select

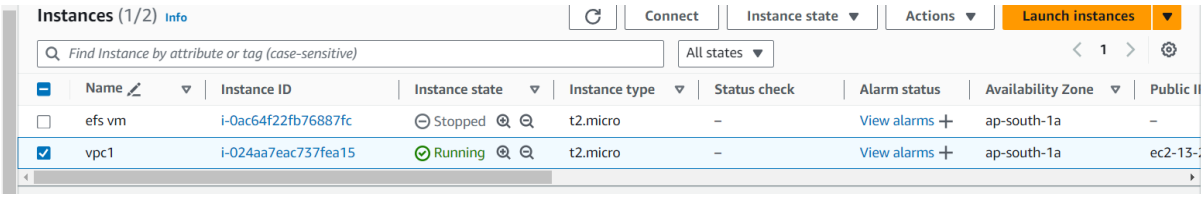
User data - *optional* Info

Upload a file with your user data or enter it in the field.

Choose file

```
sudo apt update
sudo apt install apache2 wget unzip -y
wget https://www.tooplate.com/zip-templates/2132_clean_work.zip
unzip 2132_clean_work.zip
sudo cp -r 2132_clean_work/* /var/www/html/
sudo systemctl restart apache2
```



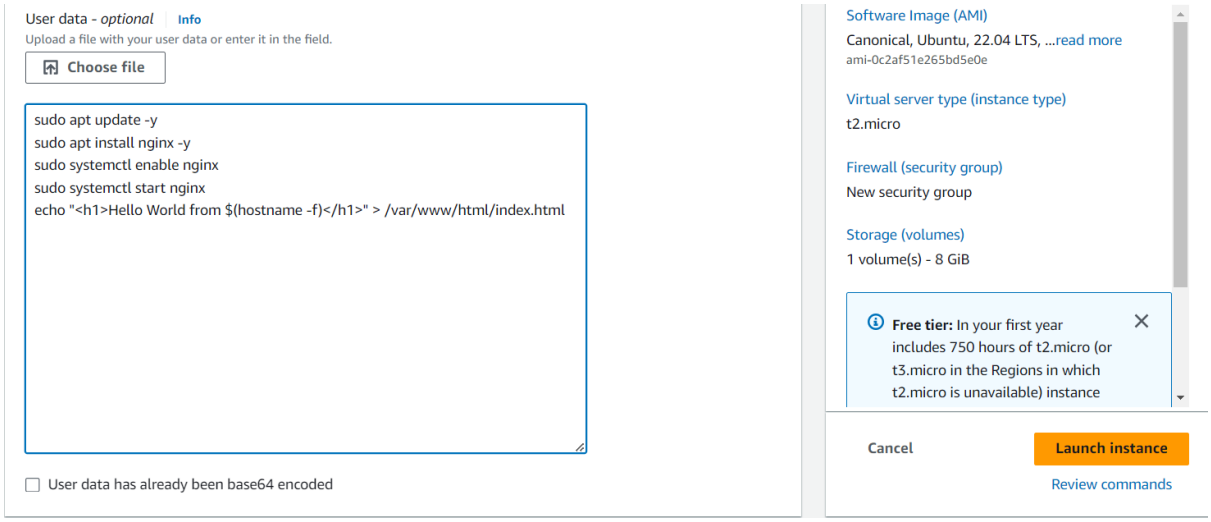


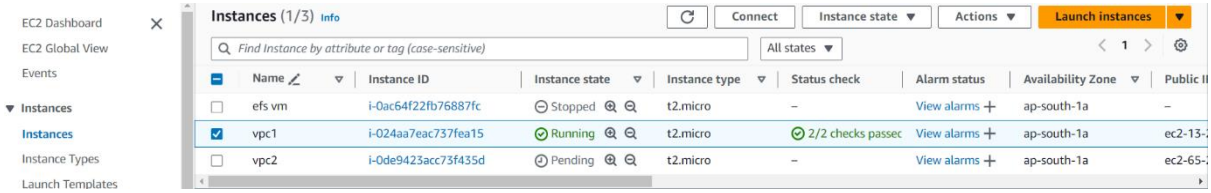
Another instance created for another application. Follow the same steps

---

```
Ubuntu

#!/bin/bash
sudo apt update -y
sudo apt install nginx -y
sudo systemctl enable nginx
sudo systemctl start nginx
echo "<h1>Hello World from $(hostname -f)</h1>" >
/var/www/html/index.html
Or
```





2 instances created for 2 diff applications

Add inbound rule for port no:80 for both instance