Multi-tier Web Appication using AWS

Name: Disha Patel

Abstract **- In this project, we will deploy a multi-tier application on the AWS Cloud and configure the fundamental networking. VPC, ALB, EC2, and RDS are the four AWS services that we will use in this project. We'll put a web application live using a multi-tier architecture. In order to build the root network, we first create a multi-tier architecture using AWS' virtual private cloud (VPC). We'll build two different kinds of networks—one private and one public—each of which will include vital details about the web application, including a load balancer, databases, servers, and applications.**

1.INTRODUCTION

We use AWS services to host multi-tier web applications. People occasionally have security issues with their web applications, such as website hacking, data leakage, and information loss. In this project, we're attempting to create a network that is secure, isn't connected to the internet, and can store vital data about applications, such as databases, servers, and webpages. This network safeguards critical application information so that it cannot be hacked or leaked online. Additionally, we'll build a second network using an AWS elastic load balancer that is connected to the internet and allows us to access the project URL.

2. Multi-tier Website Architecture

This is the project's basic architecture, which explains how it will proceed. You can see a schematic of two networks in the image below; one is network-accessible, meaning it is connected to the internet, and the other is not.

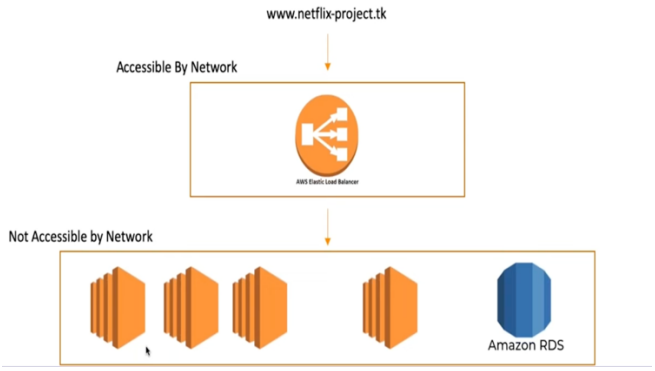

Fig. 2.1 This is a website architecture.

In the above picture accessible network include AWS elastic load balancer and it is connected with internet. Your incoming traffic is automatically split among numerous targets, including EC2 instances, containers, and IP addresses in one or more Availability Zones, thanks to elastic load balancing. It keeps track of the wellbeing of the registered targets, only sending traffic to those that are in good shape.

Load balancer acts as the client's sole point of contact. This makes your application more accessible. As your needs change, you can add and delete instances from your load balancer without impairing the general flow of requests to your service. Your load balancer is scaled through elastic load balancing as the volume of traffic to your application changes over time. The vast majority of workloads can be scaled automatically using elastic load balancing.

The protocol and port that you set are used by a listener to monitor for connection requests from clients. The listener then transmits requests to one or more registered instances using the configured protocol and port. You modify your load balancer by adding one or more listeners. To ensure that the load balancer only routes requests to healthy instances, you can configure health checks, which are used to keep an eye on the condition of the registered instances.

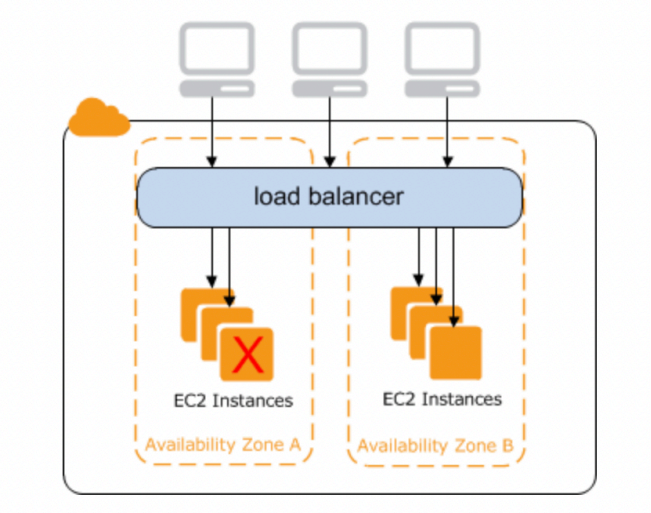

Fig. 2.2 Load balancer

It's critical to maintain roughly the same number of instances registered with the load balancer in each Availability Zone in order to guarantee that your registered instances can handle the request load in each Availability Zone. The requests are split equally across the two Availability Zones, for instance, if you have ten instances in us-west-2a and two instances in us-west-2b. As a result, the ten instances in us-west-2a and the two instances in us-west-2b both serve the same volume of traffic. Instead, each Availability Zone has to have six instances.

3. Services used in this project

(a) Amazon Elastic compute cloud

To launch our website, we'll use the EC2 service. It is an online service that offers safe, scalable computational capacity. For developers, it is intended to make web-scale cloud computing simpler. The straightforward web service interface for Amazon EC2 makes it easy for you to get and set up capacity. In the Amazon Web Services (AWS) Cloud, it offers scalable processing power. By using Amazon EC2, you can develop and deploy apps more quickly because you won't need to make an upfront hardware investment. AWS EC2 is a top IaaS illustration. For businesses who want to host cloud-based applications, EC2 provides scalable infrastructure.

AWS provides virtual servers; EC2 users do not own the physical servers.

(b) Virtual Private Cloud (VPC)

You can launch AWS resources into a virtual network that you have defined using Amazon Virtual Private Cloud (Amazon VPC). With the advantages of AWS's scalable infrastructure, this virtual network resembles a conventional network that you would run in your own data center. In our project, I first establish a primary network called the website and give it a range of IP addresses. Our IP address range will be 10.1.0.0/16. Our system's principal network, which also includes other subnets, is this one.

A private cloud computing environment housed inside of a public cloud is known as a virtual private cloud (VPC). A VPC basically creates logically segregated areas of a public cloud to offer a virtual private environment. The benefits of a VPC include access to all features and functionalities of the public cloud computing provider hosting your VPC as well as the ability to effectively use your own private cloud employing secure service.

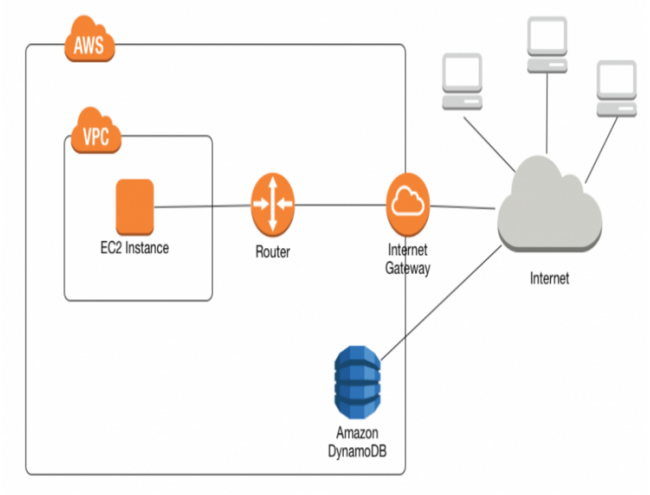

Fig. 3.1 VPC network

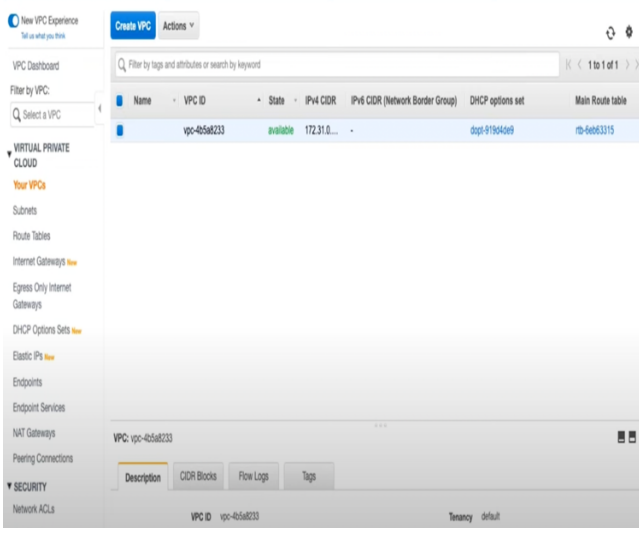

Fig. 3.2 Create VPC

Create VPC with the name NETFLIX and add IP address range to the VPC here we can see the Netflix VPC created in below picture.

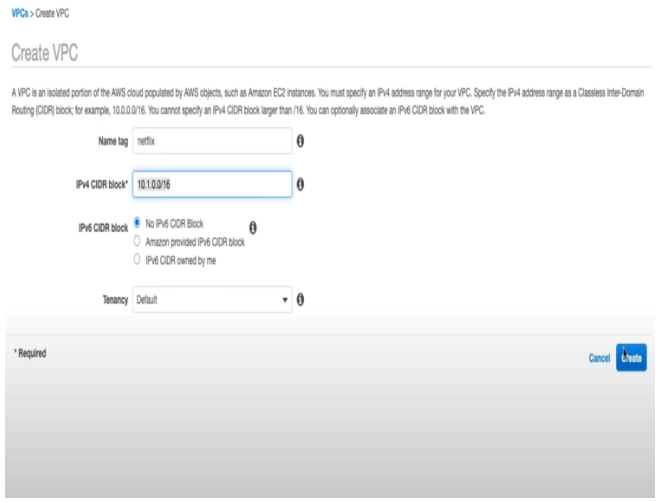

Fig. 3.3 Create VPC with Name and IP address range

Fig. 3.4 VPC Created

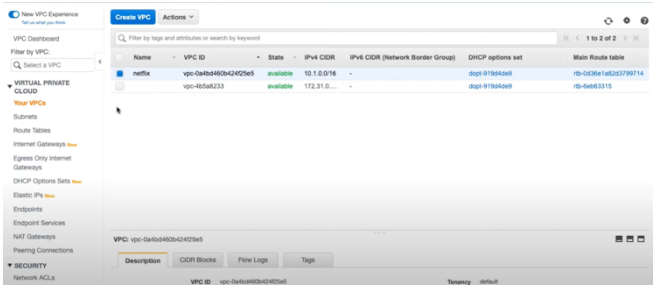

Fig. 3.5 Netflix Network

4. Sub Networks in AWS

A subnetwork is a divided portion of a larger network. More specifically, subnets divide an IP network logically into numerous, smaller network pieces. Data is sent from one computer to another via the internet using the Internet Protocol (IP).

A range of IP addresses in your VPC is called a subnet.Subnets can be attached to AWS resources like EC2 instances and RDS DB instances.Instances can be grouped together by creating subnets based on your operational and security requirements. One or more subnets, or IP address ranges, make up each VPC network. Regional resources known as subnets have corresponding IP address ranges.

We use two different subnets for the deployment of websites, referred to as the private subnet and the public subnet. Public subnets come with internet access whereas private ones do not. Additionally, because both subnets are connected to the same network, they can communicate with one another even without a network connection.

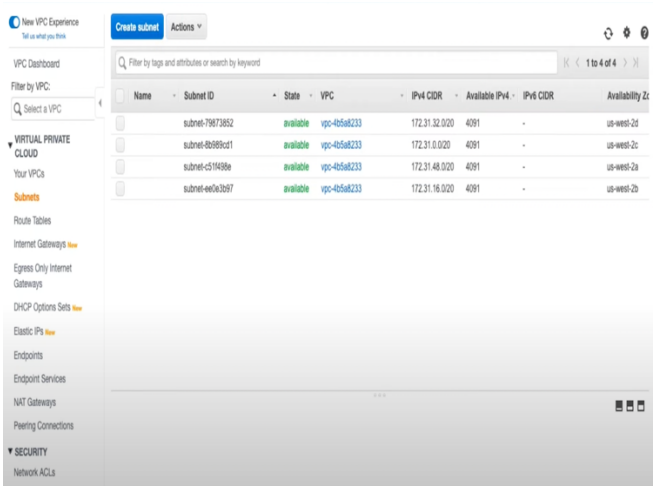

Fig. 4.1 Create Subnet

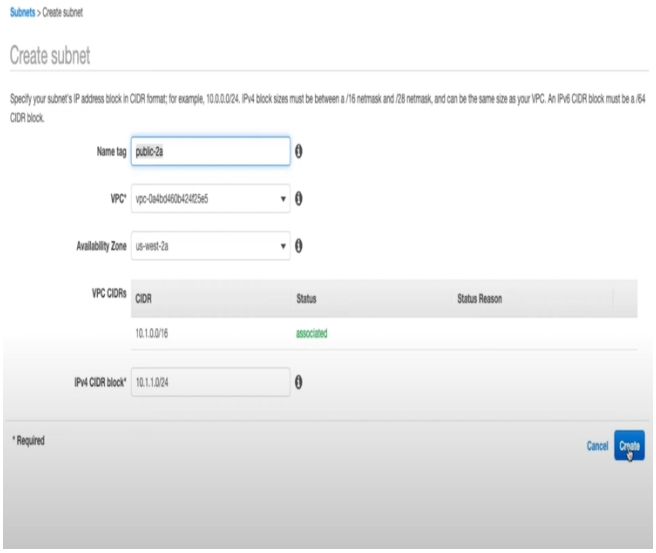

Fig. 4.2 Create Subnet with public-2a name

With Netflix VPC, I established a public subnet and assigned availability zone 2a. Additionally, I must include the 10.1.1.0/24 IP address range, which is part of the Netflix VPC range.

Fig. 4.3 Public-2a subnet created

We can see in the below picture that Public network has been created with the name of public-2a.

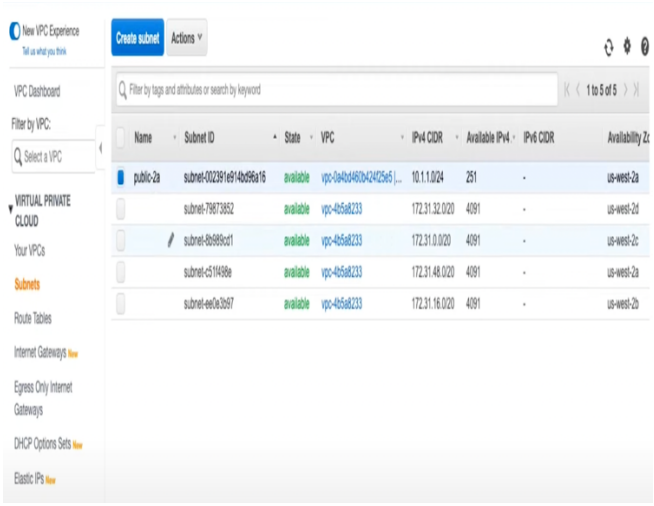

Fig. 4.4

Now create second public network with the region and Netflix VPC and assign IO address too.

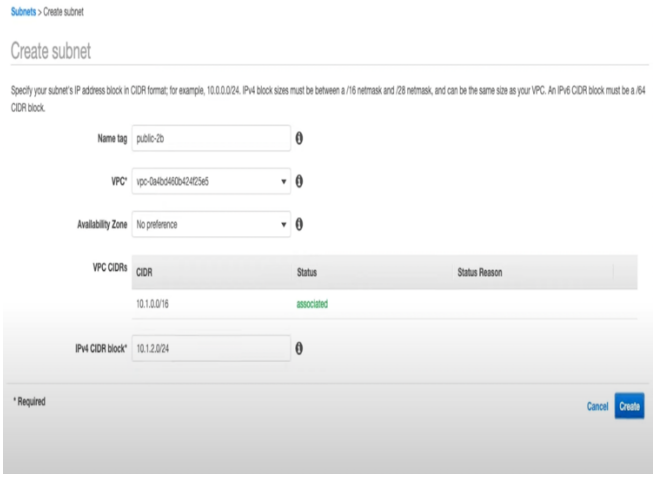

Fig. 4.5 Create Subnet with public-2b name

With Netflix VPC, I established a public subnet and assigned availability zone 2b. Additionally, I must include the 10.1.2.0/24 IP address range, which is part of the Netflix VPC range.

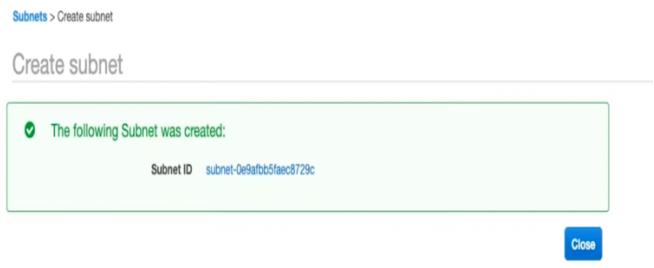

Fig. 4.6 public-2b subnet created

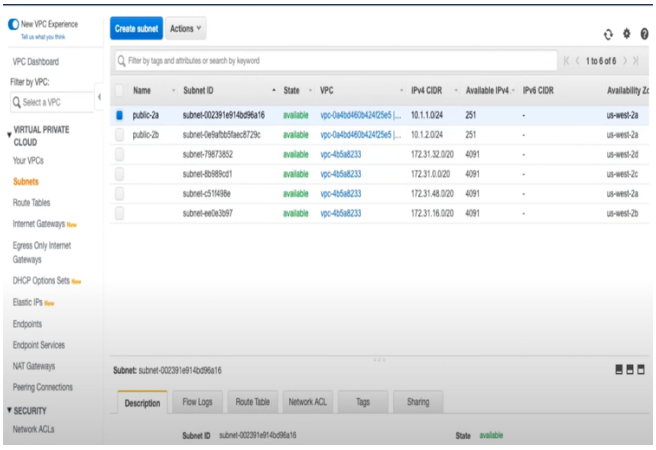

Fig. 4.7 two public subnets created

Now we are going to create private network to store datasets, servers, applications and websites. We will create 2 private subnets one is private-2a and other is private-2b.

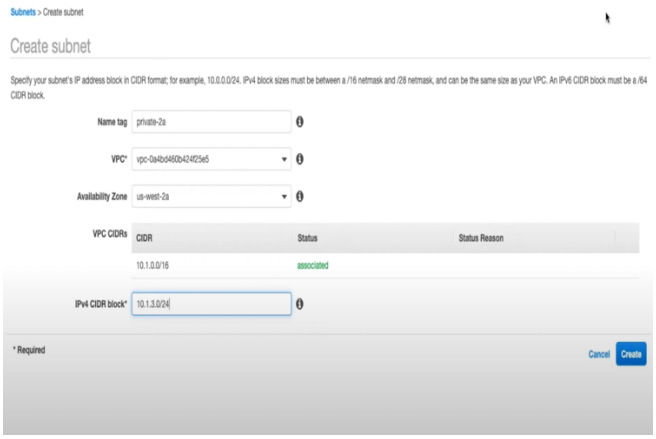

Fig. 4.8 Create subnet with Name private-2a and IP address range

With Netflix VPC, I established a private subnet and assigned availability zone 2a. Additionally, I must include the 10.1.3.0/24 IP address range, which is part of the Netflix VPC range.

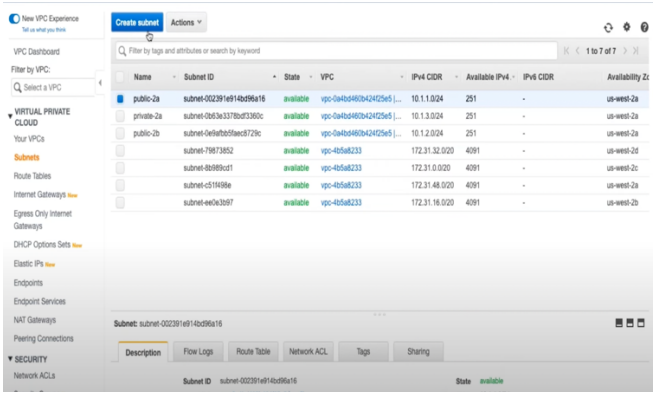

Fig. 4.9 created private-2a subnet

Now we will create the another private network with the region 2b and IP address range.

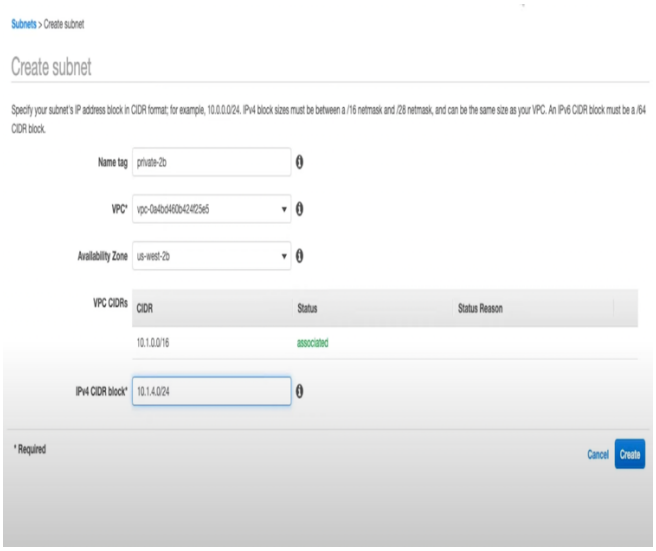

Fig. 4.10 create private-2b subnet

With Netflix VPC, I established a private subnet and assigned availability zone 2b. Additionally, I must include the 10.1.4.0/24 IP address range, which is part of the Netflix VPC range.

Fig. 4.11 subnet created succesfully

Here we can see that the public-2a and private-2a subnets have availibility zone ID 2a and for public-2b and private-2b subnets we have availibility zone ID 2b.

5. Internet gateways access for subnet

An internet gateway is a highly available, redundant, horizontally scaled VPC component that lets your VPC communicate with the internet.Both IPv4 and IPv6 traffic are supported.It has no impact on your network traffic's bandwidth or availability.

If a resource in your public subnets has a public IPv4 address or an IPv6 address, an internet gateway enables the resource to connect to the internet.

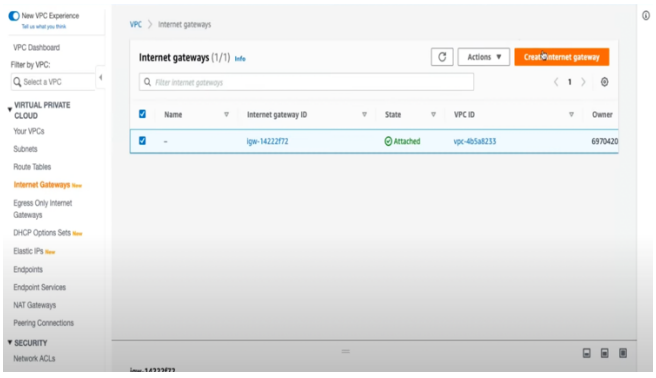

Fig. 5.1 Internet gateway

The public IPv4 or IPv6 address can be used by resources on the internet to connect to resources in your subnet. For instance, an internet gateway enables you to use your local computer to connect to an EC2 instance on AWS.

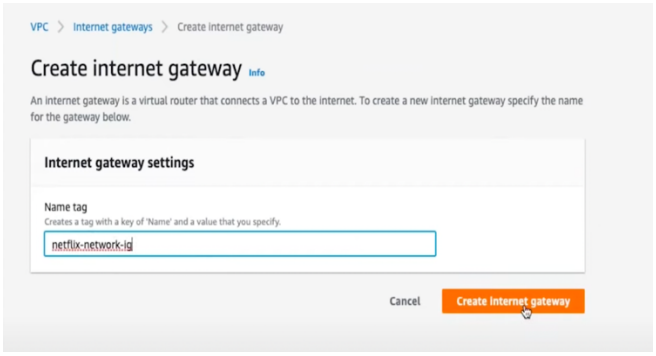

Fig. 5.2 create internet gateway

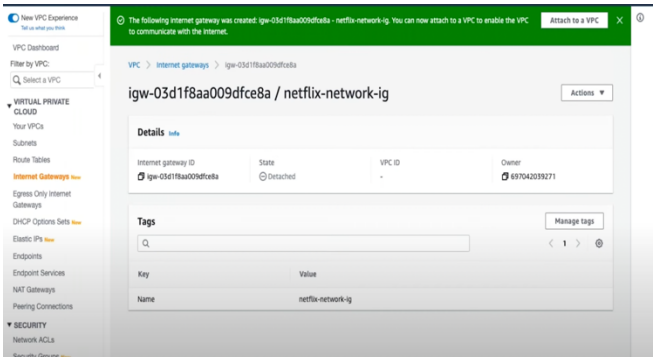

Fig. 5.3 created internet gateway

A subnet is referred to be a public subnet if it is connected to a route table that has a route to an internet gateway. A private subnet is one that is connected to a route table but does not have a route to an internet gateway.

You can provide a route for the internet gateway to any locations that are implicitly unknown to the route table (0.0.0.0/0 for IPv4 or::/0 for IPv6) in your public subnet's route table. Alternately, you can limit the scope of the route to a smaller set of IP addresses, such as the Elastic IP addresses of additional Amazon EC2 instances outside your VPC or the public IPv4 addresses of your company's externally accessible endpoints on AWS.

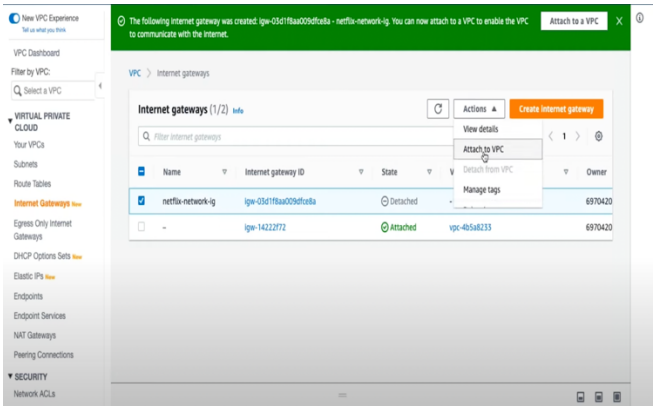

Fig. 5.4 Attach to VPC

After creating internet gateway attached it to netflix VPC than select the VPC that is netflix and attach internet gateway.

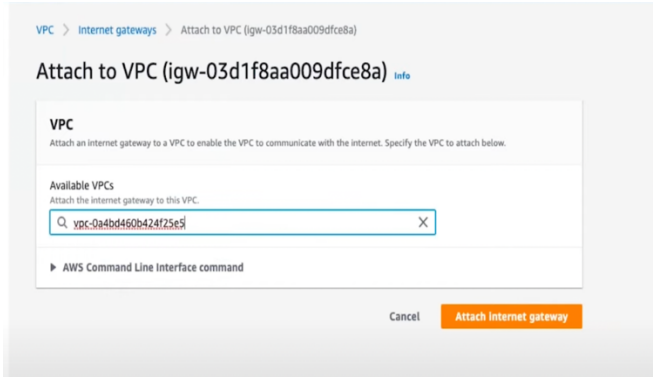

Fig. 5.5 attached internet gateway

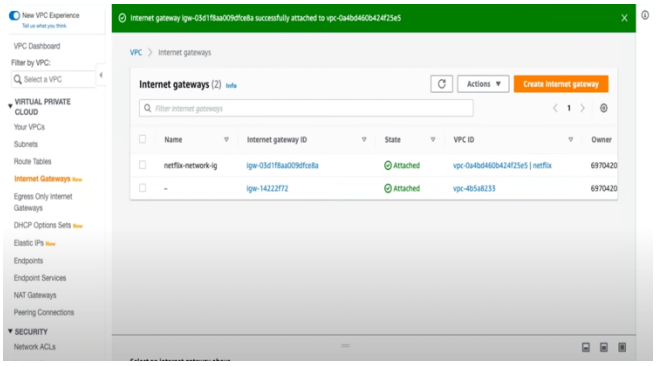

Fig. 5.6 successfully attched internet gateways

Now our internet gateway is attched to my network but still the resources that are going to be in my network they will not be able to access it. The reason for that is whenever you deployed any resources on cloud it is not directly deploy in main network that you created. It is always deployed in sub part of network which is nothing but the subnets that we created. So the subnet study of deployed we have to configure them to connect to this internet gateway. Now internet gateway is part of our network but subnets are not aware about that internet network

6. Route Tables

Routes are a set of rules in a route table that decide where network traffic from your gateway or subnet goes.

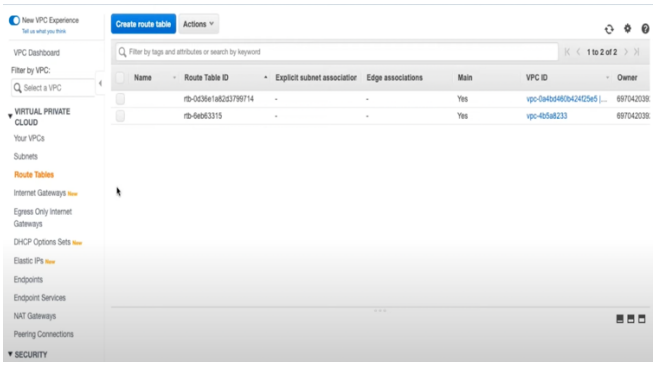

Fig. 6.1 Create route tables

A routing table is a set of rules used to decide where data packets traveling over an Internet Protocol (IP) network will go. It is typically viewed as a table.Routing tables are utilized by all IP-enabled devices, including routers and switches.

Route tables are assigned to distinct subnets within a VPC.All subnets would be assigned to a single route table that was created in a VPC.In a VPC, you can either create multiple route tables or leave the default route table alone.

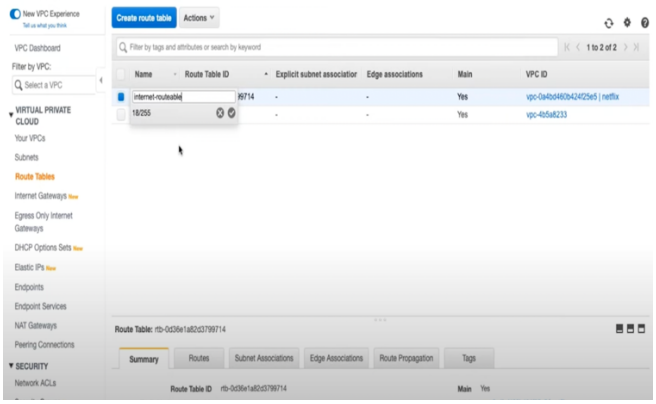

Fig. 6.2 Create route table

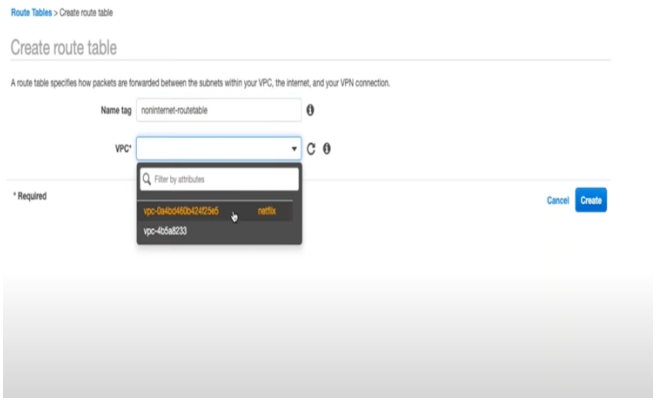

Fig. 6.3 Create route table for noninternet subnet

Now attached the subnets which will have internet access for that go to subnet association and edit subnet association to add subnets that is public-2a and public-2b.

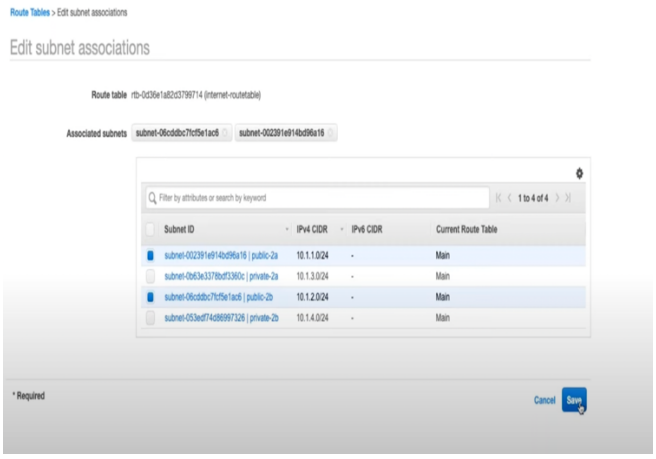

Fig. 6.4 Assign subnets

Now subnets have been attached to the route tables now we will specifity a route to the internet gateway to this route table and than all the subnets which are connected to these route table will be able to see that internet gateway

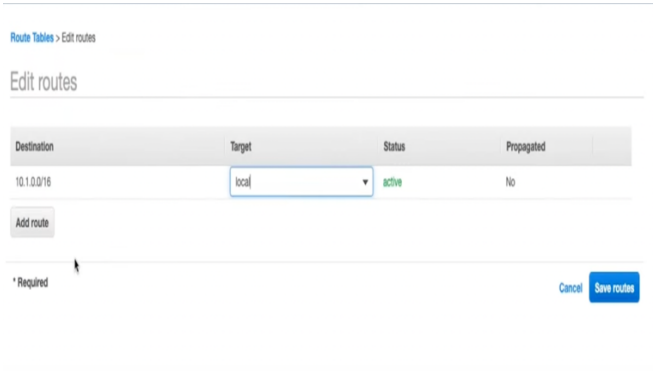

Fig. 6.5 Add route

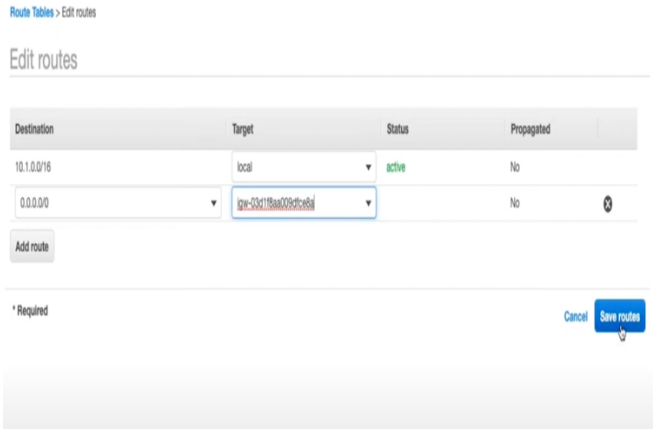

Fig. 6.6 assign destination IP address

Fig. 6.7 Assign subnets to route table

Now attached the subnets which will have not internet access for that go to subnet association and edit subnet association to add subnets that is private-2a and private-2b.

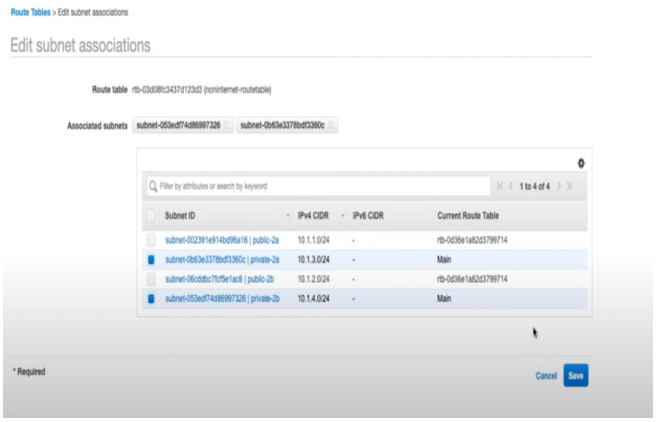

Fig. 6.8 Assign subnets to non internet subnets

Now we have two route table configued and we have our network configured. Now the resources which can talk to each other on internet or which basically have internet access they will need public IP address. If they do not have public IP be outside from the netflix network will not be able to talk to them.

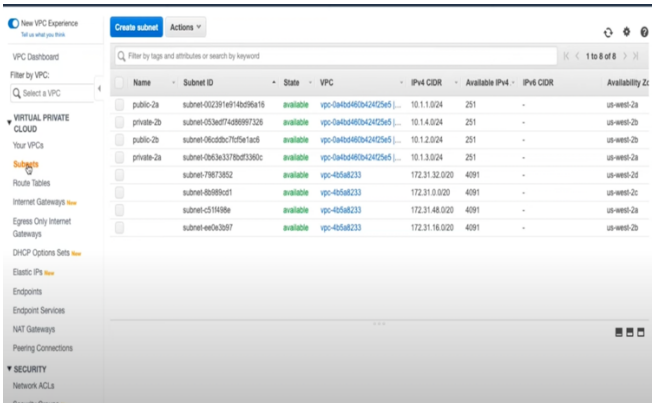

Fig. 6.9

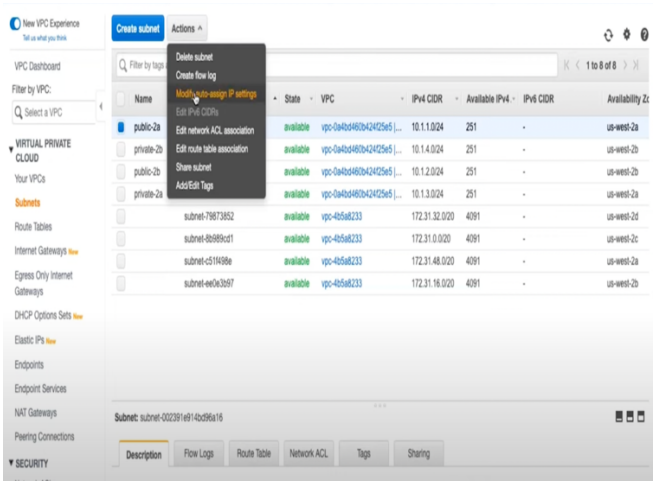

Fig. 6.10 Modify auto assign IP addresses

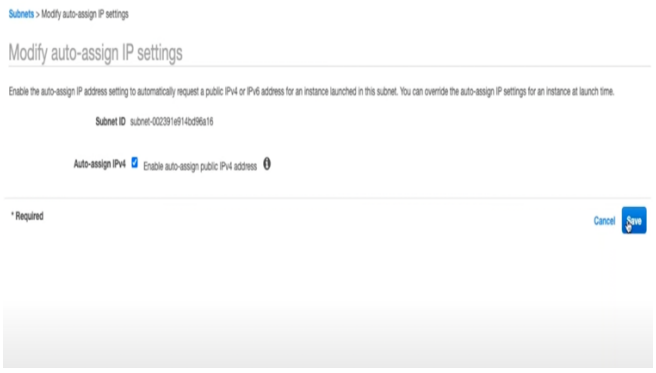

Fig. 6.11enables auto assign IPv4

Now all the resources which we are going to deploy in public-2a are going to have a public IP address. It is a basic IP address which can be listout from the intrenet. All the 10.1.0.0 IP address series are part of local network that is netflix network. when we talk about internet, it is also a network. Internet has it's own IP addresses which are given to the people when they want to connect to each other.

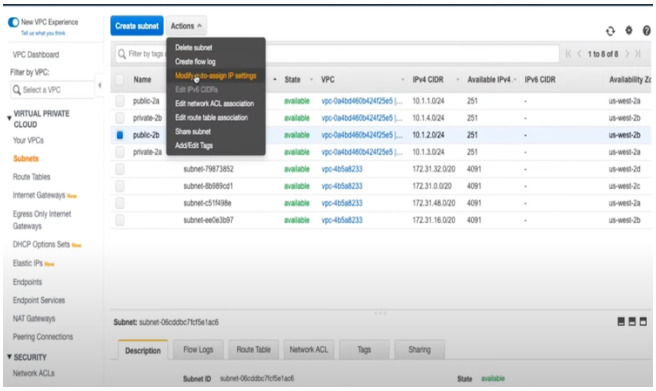

Fig. 6.12 Modify auto assign IP addresses

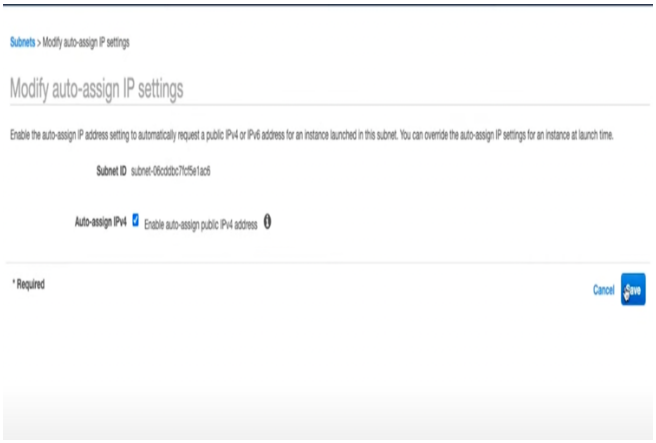

Fig. 6.13 enables auto assign IPv4

Now all the resources that we are going to deploy in public-2a and public-2b will have an IP address.

7. EC2 Instances

There are numerous instance types optimized for various use cases available in Amazon EC2.You have the flexibility to select the appropriate combination of resources for your applications by using instance types, which include a variety of combinations of CPU, memory, storage, and networking capacity.

An Amazon Machine Image (AMI), which contains an operating system, applications, and configurations, is created during EC2 setup. Users can then launch virtual machines as needed once that AMI has been registered with EC2 and loaded to the Amazon Simple Storage Service (S3).

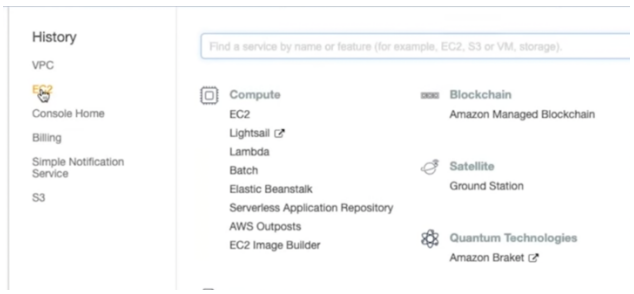

Fig. 7.1 EC2 service

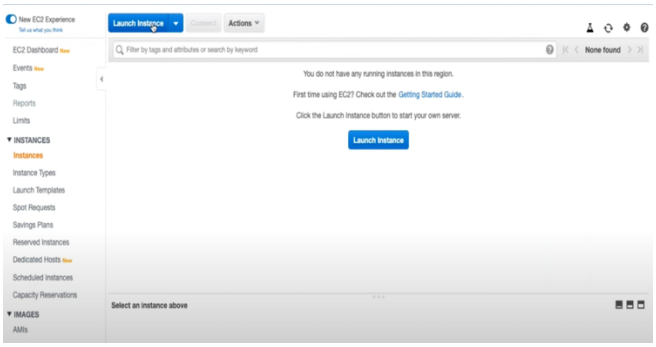

Fig. 7.2 lauch instances

Now when we want to rent a server on cloud the first thing is to decide the what kind of operating system that we want to run on that server. Let me select linux operating system to run my project.

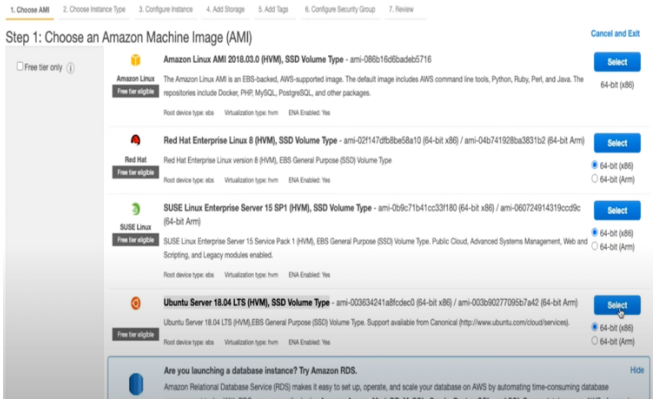

Fig. 7.3 Operating system

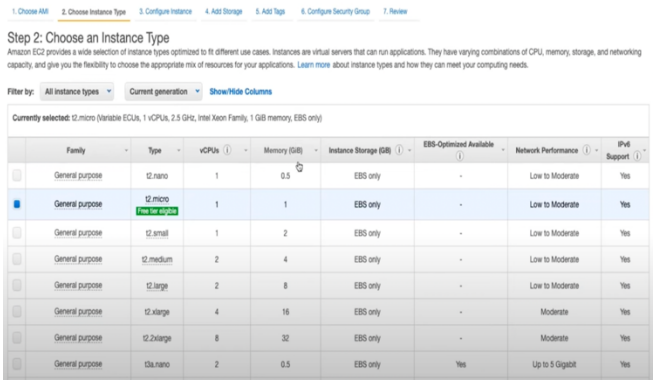

Fig. 7.4

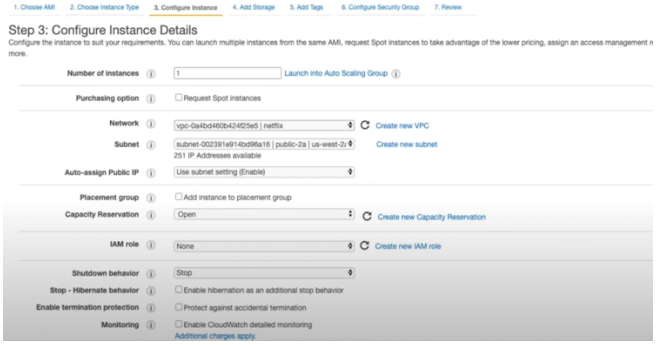

Fig. 7.5 configure instance details

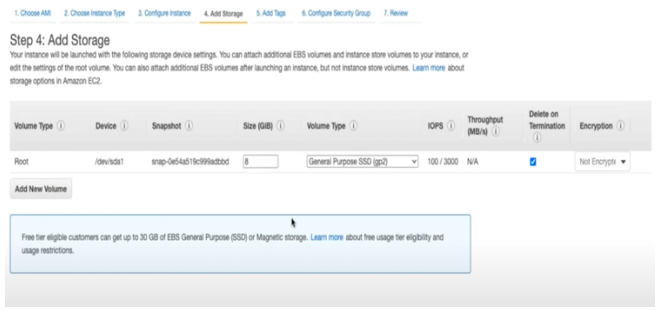

Fig. 7.6 Add storage

Now when you talk about firewalls, you basically referring to security groups in AWS. My servers will be governed by a firewall which is called a security group. I can create a new firewall and can choose an existing firewall as well.

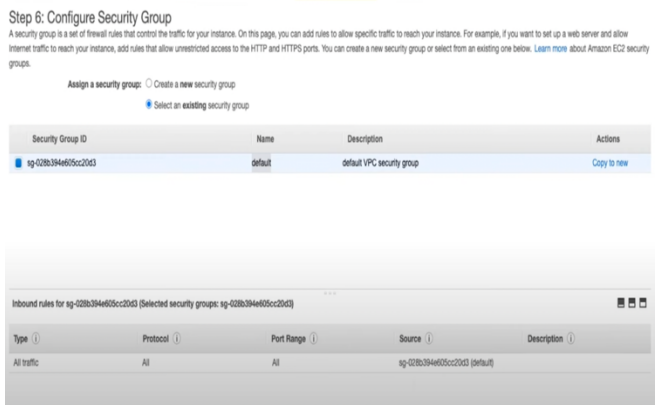

Fig. 7.7 select default firewall

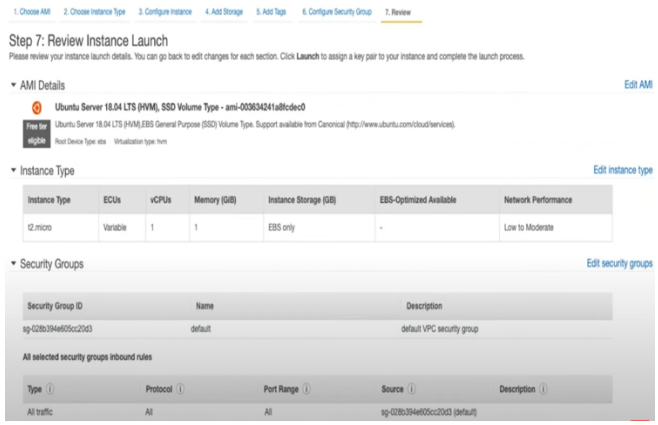

Fig. 7.8 Review and lunch instance

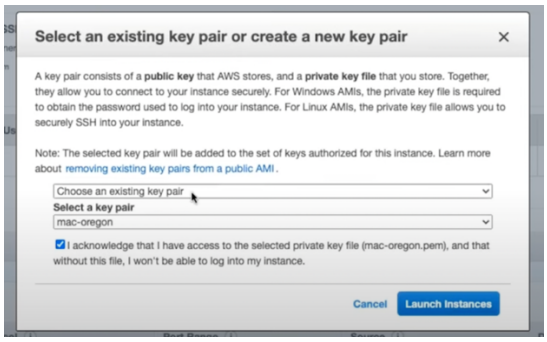

Fig. 7.9 authentication

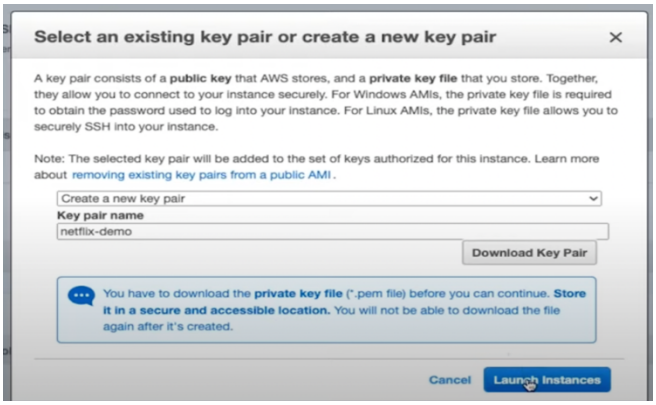

Fig. 7.10 create key pair

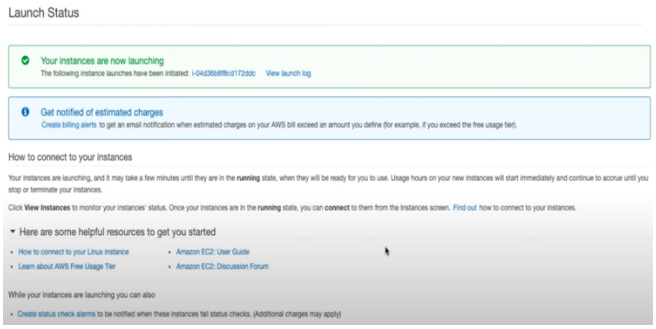

Fig. 7.11 successfully lunch instance

The instance will launch successfully and we launch it under public server.

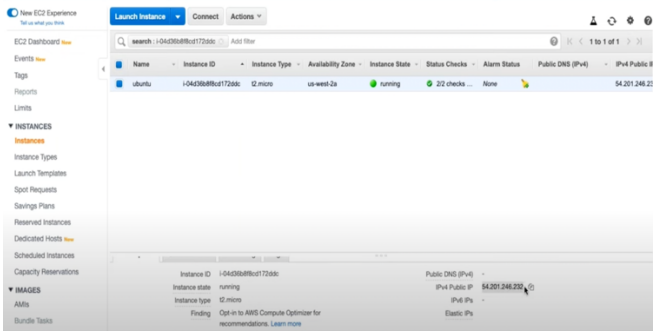

Fig. 7.12 ubuntu instance

Now select security group for ubuntu instance. It is default security so change setting there.

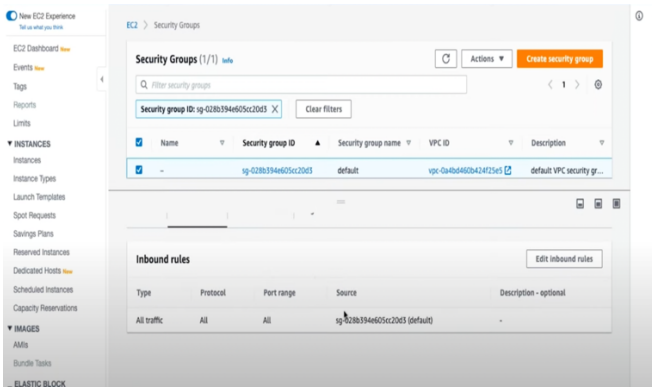

Fig. 7.13 change inbound rule

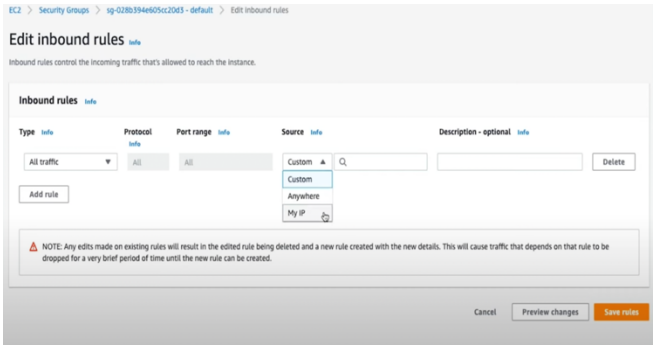

Fig. 7.14

Fig. 7.15

Write above mention command in terminal

Fig. 7.16

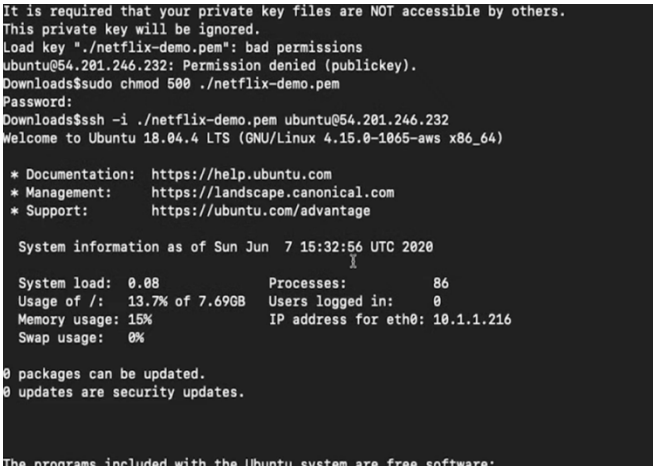

Fig. 7.17 lauch instances

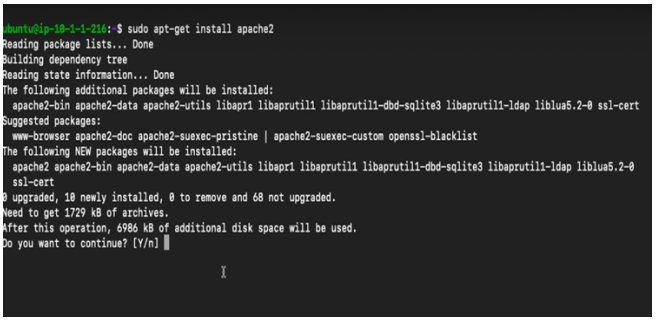

Fig. 7.18 lauch instances

8. References

[1] D. K. gorge and M. K. L. Lee, “Web application hosting on AWS using aws services,” *J. Comput. Graph. Statist.*, vol. 24, no. 2, pp. 379–393, Jun. 2015, doi: 10.1080/1034545.2015.903443.

[2] B. Rieder, Engines of Order:AWS service for multi-tier architecure and deployment: Amsterdam Univ. Press, 2020.

[3] I. Boglaev, “Aws service and instances to store and use the data with mutliple services and web application,” *J. Comput. Math.*, vol. 34, no. 3, pp. 262–284, May 2016, doi: 10.4208/jcm.1512-m2015-4545