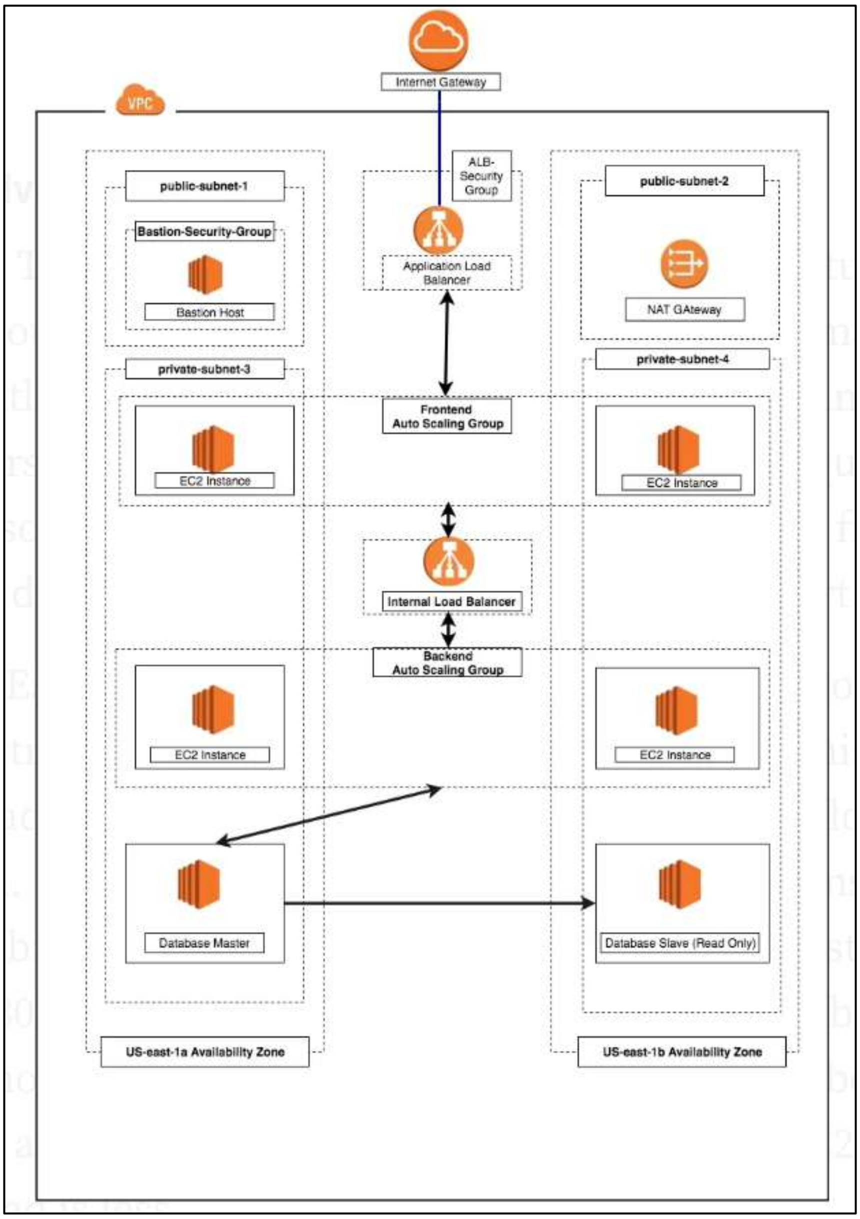
## 1. Challenge #1

A 3-tier environment is a common setup. Use a tool of your choosing/familiarity create these resources on a cloud environment (Azure/AWS/GCP). Please remember we will not be judged on the outcome but more focusing on the approach, style and reproducibility.

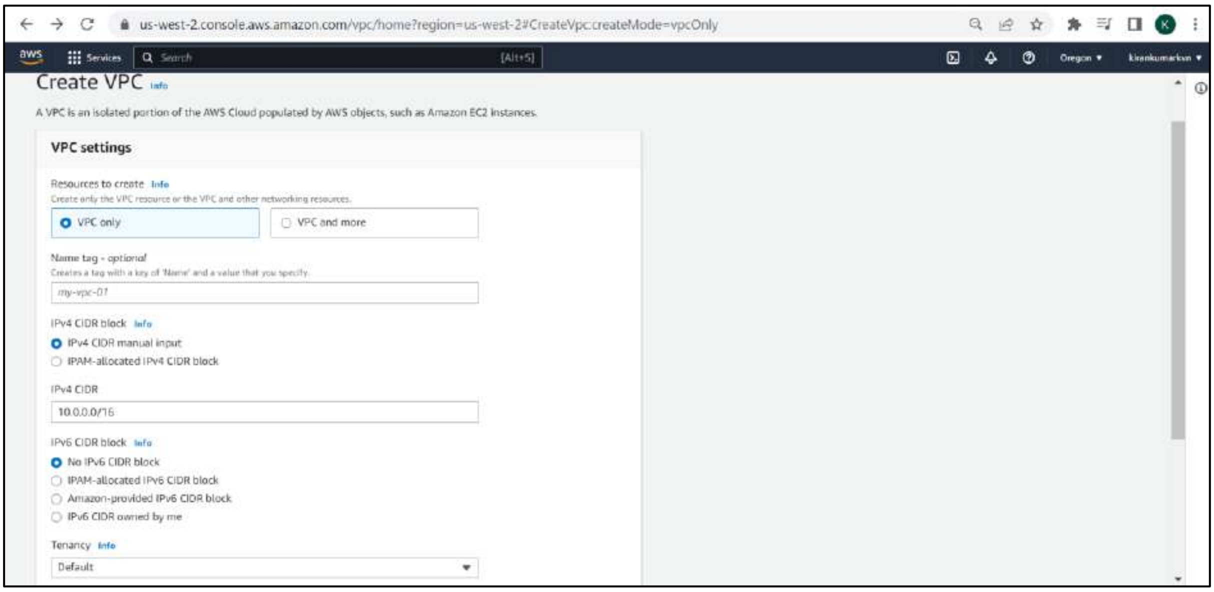
## My Response :

I have created a 3-tier environment in AWS. Here, I have used these AWS services to design and build a three-tier cloud infrastructure: Elastic Compute Cloud (EC2), Auto Scaling Group, Virtual Private Cloud(VPC), Elastic Load Balancer (ELB), Security Groups and the Internet Gateway. This infrastructure is designed to be highly available and fault tolerant.



**Following is what I did :**

1. **First, I setup a Virtual Private Cloud (VPC):** Using VPC, I can manage my AWS resources in a more secure and scalable manner. I went to the VPC section of the AWS services, and then clicked on the 'Create VPC' button. I gave my VPC a name and a CIDR block of 10.0.0.0/16.



us-west-2.console.aws.amazon.com/vpc/home?region=us-west-2#CreateVpc:createMode=vpcOnly
aws
Services
Search
[Alt+S]
Oregon
kirankumarkvn

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.
● VPC only ○ VPC and more

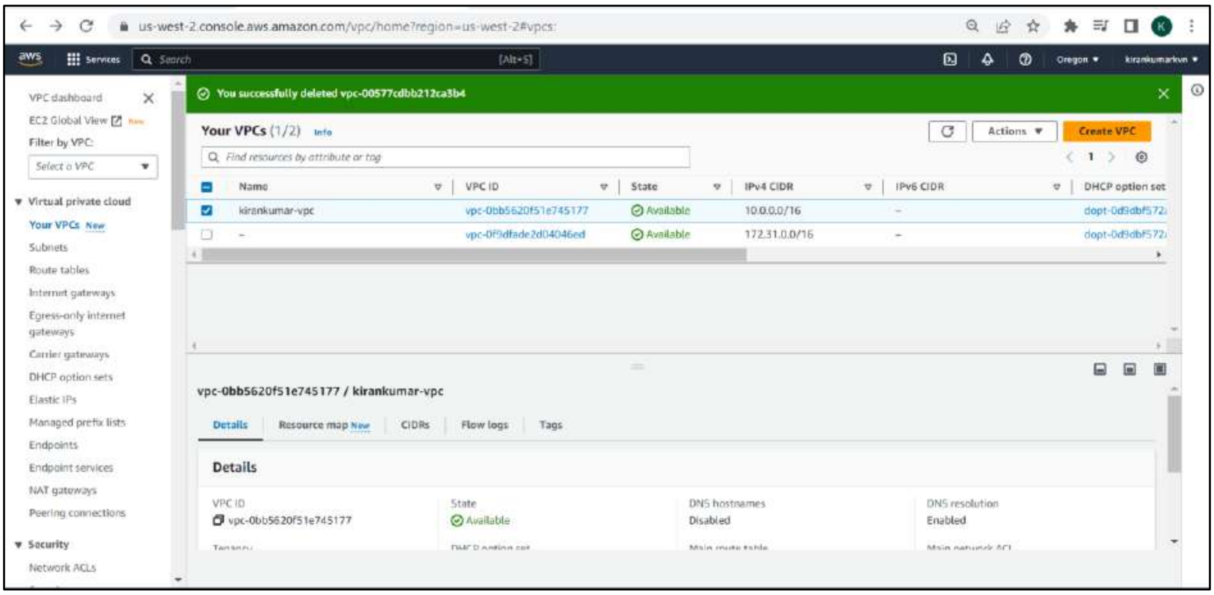
Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.
my-vpc-01

IPv4 CIDR block Info
● IPv4 CIDR manual input
○ IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/16

IPv6 CIDR block Info
● No IPv6 CIDR block
○ IPAM-allocated IPv6 CIDR block
○ Amazon-provided IPv6 CIDR block
○ IPv6 CIDR owned by me

Tenancy Info
Default



You successfully deleted vpc-00577cdbb212ca3b4

Your VPCs (1/2) Info

Find resources by attribute or tag

| | Name | VPC ID | State | IPv4 CIDR | IPv6 CIDR | DHCP option set |
|---|---|---|---|---|---|---|
| ☑ | kirankumar-vpc | vpc-0bb5620f51e745177 | Available | 10.0.0.0/16 | – | dopt-0d9dbf572 |
| □ | – | vpc-0f9dfade2d04046ed | Available | 172.31.0.0/16 | – | dopt-0d9dbf572 |

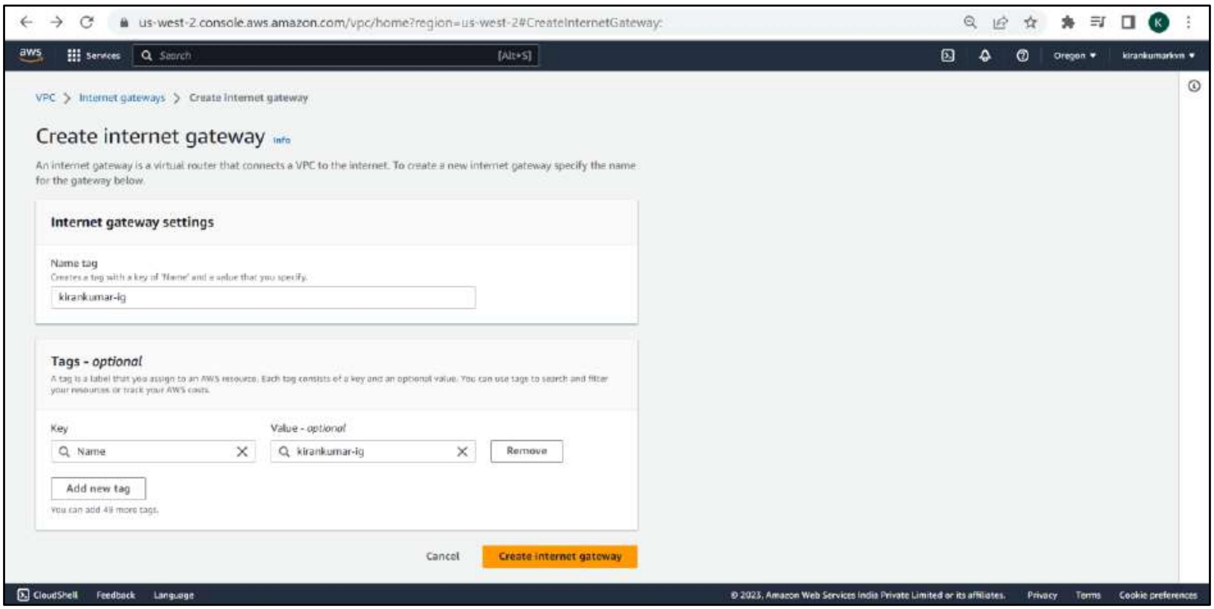
vpc-0bb5620f51e745177 / kirankumar-vpc

Details | Resource map New | CIDRs | Flow logs | Tags

Details

| VPC ID | State | DNS hostnames | DNS resolution |
|---|---|---|---|
| vpc-0bb5620f51e745177 | Available | Disabled | Enabled |

2. **Then, I setup the Internet Gateway**: The Internet Gateway allows communication between the EC2 instances in the VPC and the internet. To create the Internet Gateway, I navigated to the Internet Gateways page and then clicked on 'Create internet gateway' button.


us-west-2.console.aws.amazon.com/vpc/home?region=us-west-2#CreateInternetGateway:
Services
Search [Alt+S]
Oregon ▾
kirankumarkvm ▾

VPC > Internet gateways > Create internet gateway

## Create internet gateway Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

**Internet gateway settings**

Name tag
Creates a tag with a key of 'Name' and a value that you specify.
kirankumar-ig

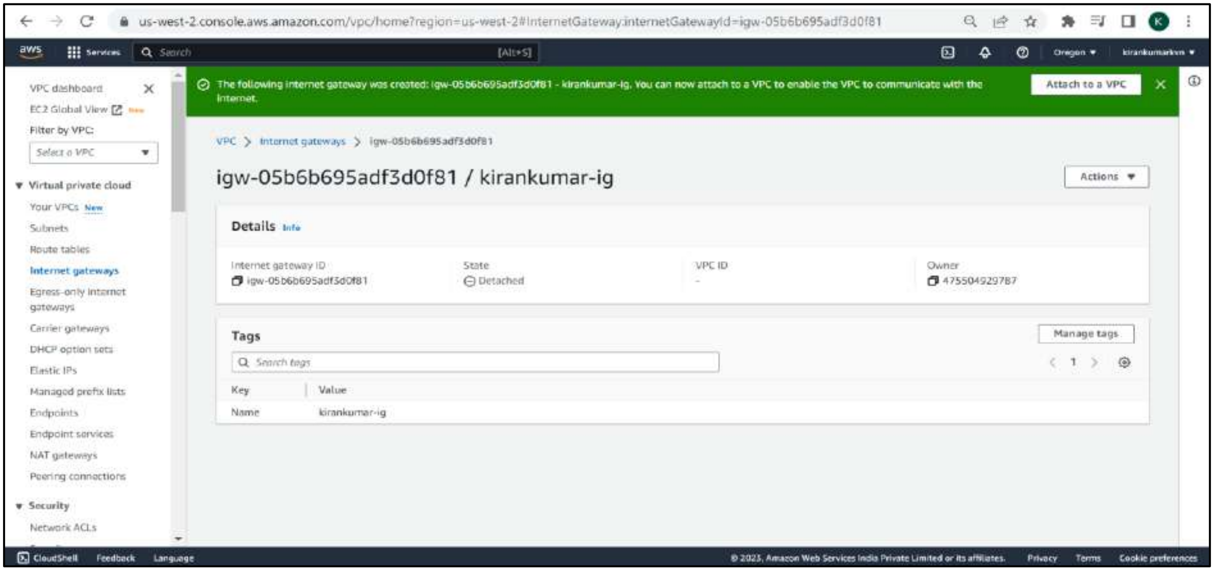
**Tags - *optional***
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

| Key | Value - *optional* | |
|---|---|---|
| Name | kirankumar-ig | Remove |

Add new tag
You can add 49 more tags.

Cancel | Create internet gateway

CloudShell Feedback Language
© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences



The following internet gateway was created: igw-05b6b695adf3d0f81 - kirankumar-ig. You can now attach to a VPC to enable the VPC to communicate with the internet. Attach to a VPC

VPC > Internet gateways > igw-05b6b695adf3d0f81

## igw-05b6b695adf3d0f81 / kirankumar-ig

Actions ▾

**Details** Info

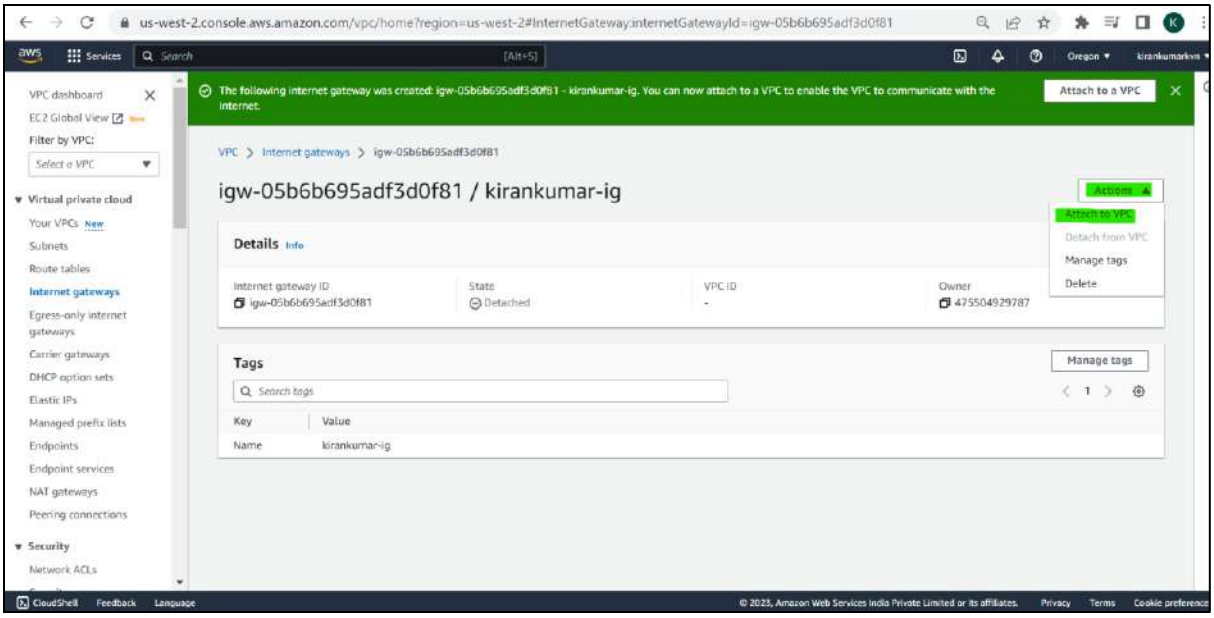
| Internet gateway ID | State | VPC ID | Owner |
|---|---|---|---|
| igw-05b6b695adf3d0f81 | Detached | - | 475504929787 |

**Tags** Manage tags

Search tags

| Key | Value |
|---|---|
| Name | kirankumar-ig |

CloudShell Feedback Language
© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences



The following internet gateway was created: igw-05b6b695adf3d0f81 - kirankumar-ig. You can now attach to a VPC to enable the VPC to communicate with the internet. Attach to a VPC

VPC > Internet gateways > igw-05b6b695adf3d0f81

## igw-05b6b695adf3d0f81 / kirankumar-ig

**Details** Info

| Internet gateway ID | State | VPC ID | Owner |
|---|---|---|---|
| igw-05b6b695adf3d0f81 | Detached | - | 475504929787 |

**Tags** Manage tags

Search tags

| Key | Value |
|---|---|
| Name | kirankumar-ig |

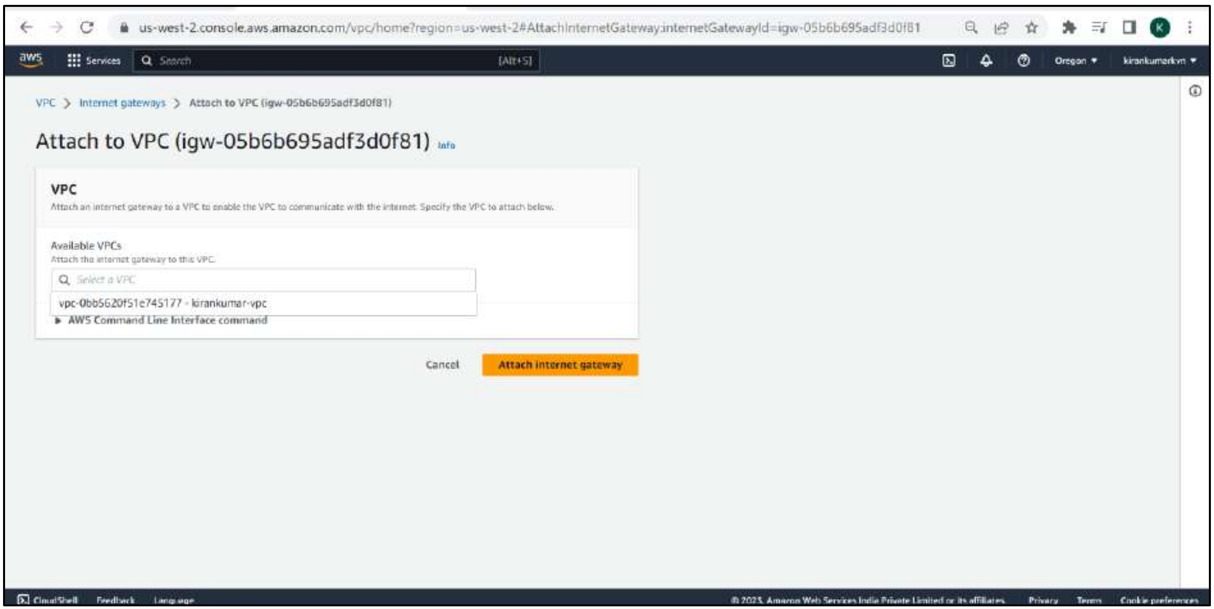
CloudShell Feedback Language
© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preference

I attached my VPC to the internet gateway like this :

a. Selected the 'internet gateway',

b. Clicked on the 'Actions' button and then selected 'Attach to VPC',

c. Selected the VPC to attach the internet gateway and clicked 'Attach'.


us-west-2.console.aws.amazon.com/vpc/home?region=us-west-2#AttachInternetGateway:internetGatewayId=igw-05b6b695adf3d0f81
aws
Services
Search
[Alt+S]
Oregon ▾
kirankumarkvn ▾
VPC > Internet gateways > Attach to VPC (igw-05b6b695adf3d0f81)
Info
▸ AWS Command Line Interface command
Cancel
Attach internet gateway
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences

Attach to VPC (igw-05b6b695adf3d0f81)

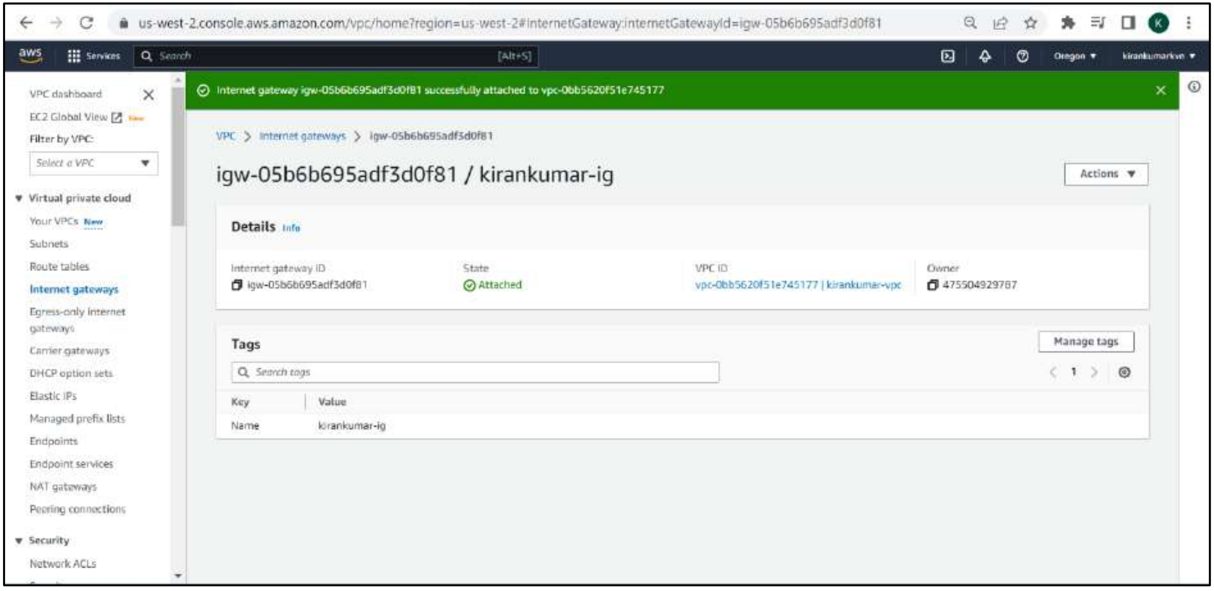
VPC

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs

Attach the internet gateway to this VPC.

vpc-0bb5620f51e745177 - kirankumar-vpc



Internet gateway igw-05b6b695adf3d0f81 successfully attached to vpc-0bb5620f51e745177

igw-05b6b695adf3d0f81 / kirankumar-ig

Details

Internet gateway ID: igw-05b6b695adf3d0f81
State: Attached
VPC ID: vpc-0bb5620f51e745177 | kirankumar-vpc
Owner: 475504929787

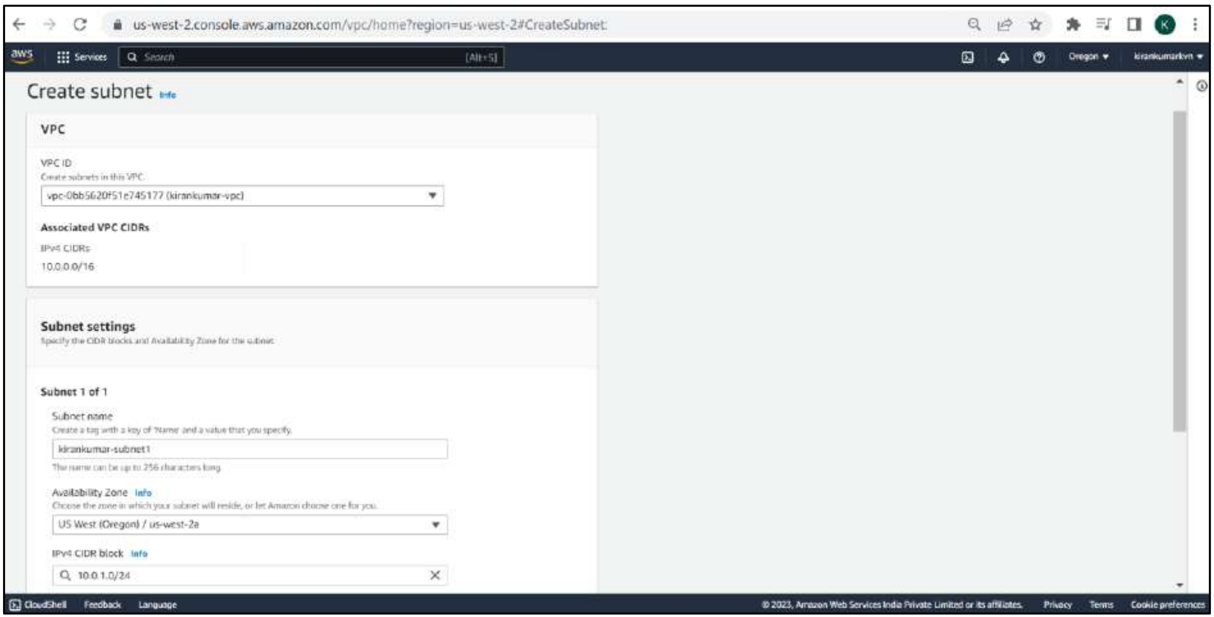
Tags

| Key | Value |
|---|---|
| Name | kirankumar-ig |

3. **Created 4 Subnets:** The subnet is a way for us to group our resources within the VPC with their IP range. A subnet can be public or private. EC2 instances within a public subnet have public IPs and can directly access the internet while those in the private subnet does not have public IPs and can only access the internet through a NAT gateway. For my setup, we created the following subnets with the corresponding IP ranges.

kirankumar-subnet1(public) | CIDR (10.0.1.0/24) | Availability Zone (us-east-1a)
kirankumar-subnet2(public) | CIDR (10.0.2.0/24) | Availability Zone (us-east-1b)

kirankumar-subnet3(private)| CIDR (10.0.3.0/24) | Availability Zone (us-east-1a)
kirankumar-subnet4(private)| CIDR(10.0.4.0/24) | Availability Zone (us-east-1b)


us-west-2.console.aws.amazon.com/vpc/home?region=us-west-2#CreateSubnet:
aws
Services
Search
[Alt+S]
Oregon ▾
kirankumarkvn ▾

Create subnet Info

VPC

VPC ID
Create subnets in this VPC
vpc-0bb5620f51e745177 (kirankumar-vpc)

Associated VPC CIDRs
IPv4 CIDRs
10.0.0.0/16

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

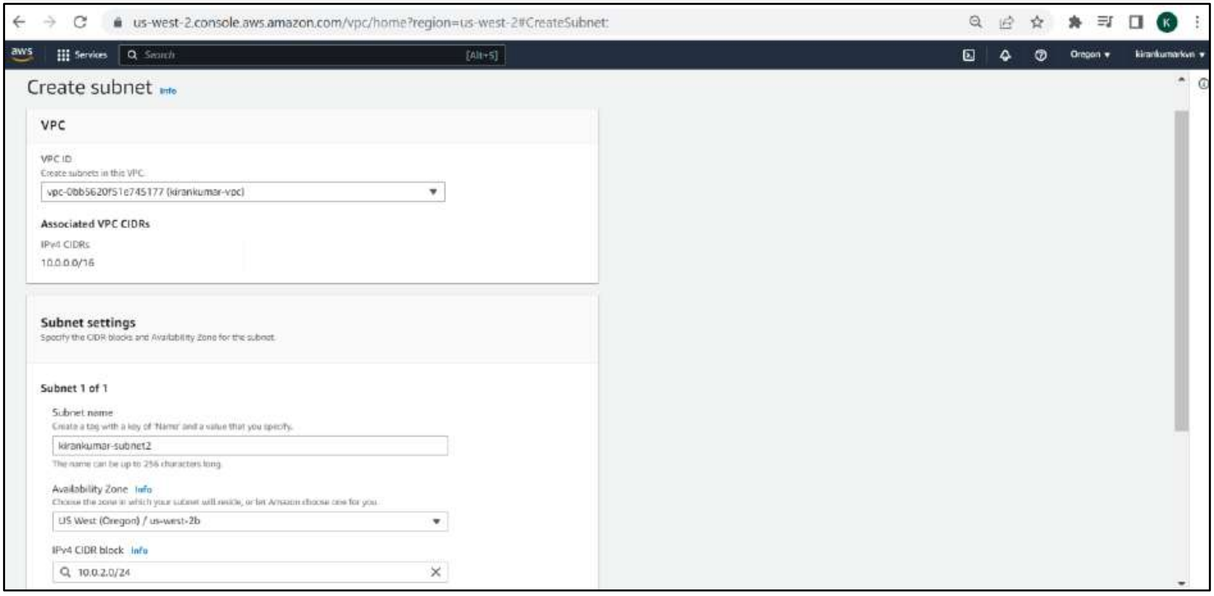
Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
kirankumar-subnet1
The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US West (Oregon) / us-west-2a

IPv4 CIDR block Info
10.0.1.0/24

CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences


us-west-2.console.aws.amazon.com/vpc/home?region=us-west-2#CreateSubnet:
aws
Services
Search
[Alt+S]
Oregon ▾
kirankumarkvn ▾

Create subnet Info

VPC

VPC ID
Create subnets in this VPC
vpc-0bb5620f51e745177 (kirankumar-vpc)

Associated VPC CIDRs
IPv4 CIDRs
10.0.0.0/16

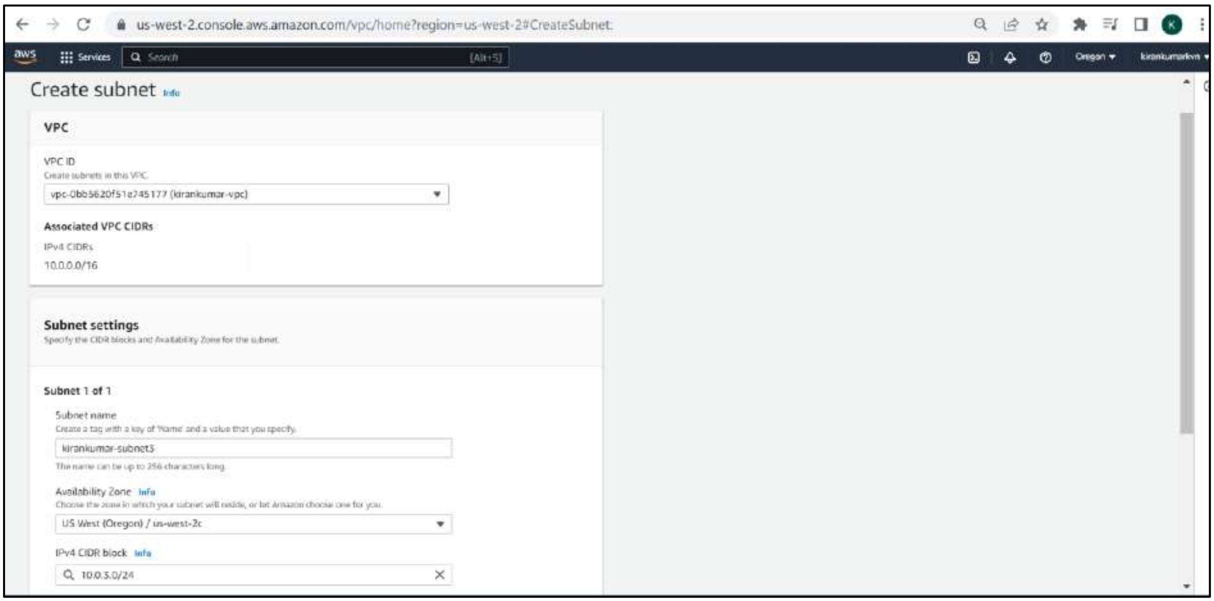
Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
kirankumar-subnet2
The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US West (Oregon) / us-west-2b

IPv4 CIDR block Info
10.0.2.0/24



us-west-2.console.aws.amazon.com/vpc/home?region=us-west-2#CreateSubnet:

## Create subnet

**VPC**

VPC ID
Create subnets in this VPC.
vpc-0bb5620f51e745177 (kirankumar-vpc)

**Associated VPC CIDRs**
IPv4 CIDRs
10.0.0.0/16

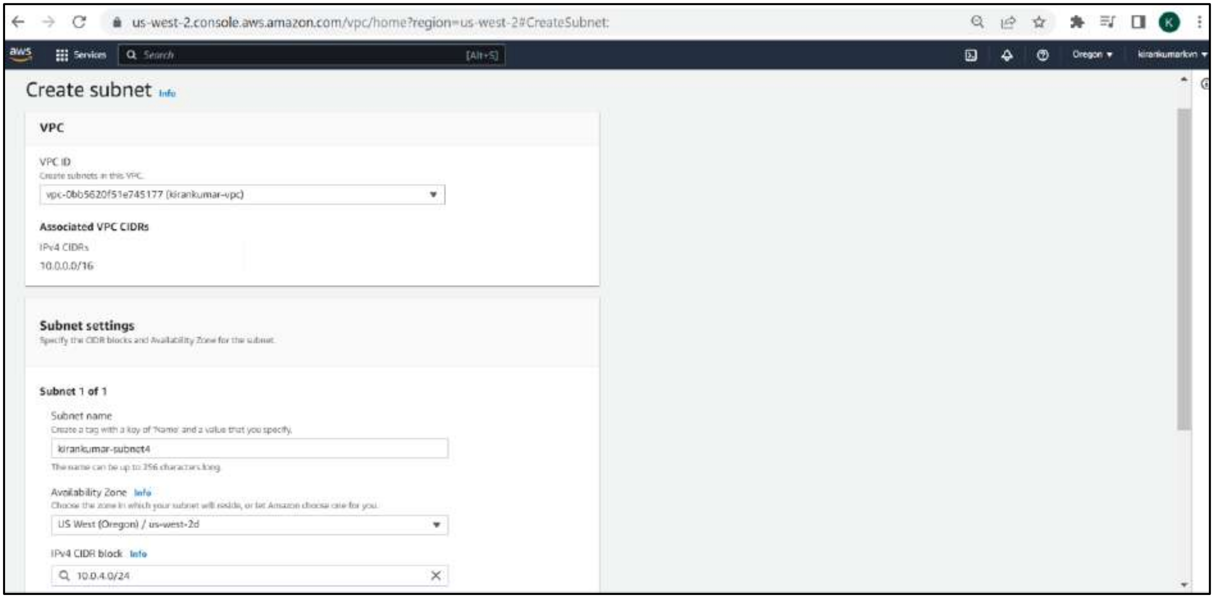
**Subnet settings**
Specify the CIDR blocks and Availability Zone for the subnet.

**Subnet 1 of 1**

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
kirankumar-subnet3
The name can be up to 256 characters long.

Availability Zone
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US West (Oregon) / us-west-2c

IPv4 CIDR block
10.0.3.0/24



us-west-2.console.aws.amazon.com/vpc/home?region=us-west-2#CreateSubnet:

## Create subnet

**VPC**

VPC ID
Create subnets in this VPC.
vpc-0bb5620f51e745177 (kirankumar-vpc)

**Associated VPC CIDRs**
IPv4 CIDRs
10.0.0.0/16

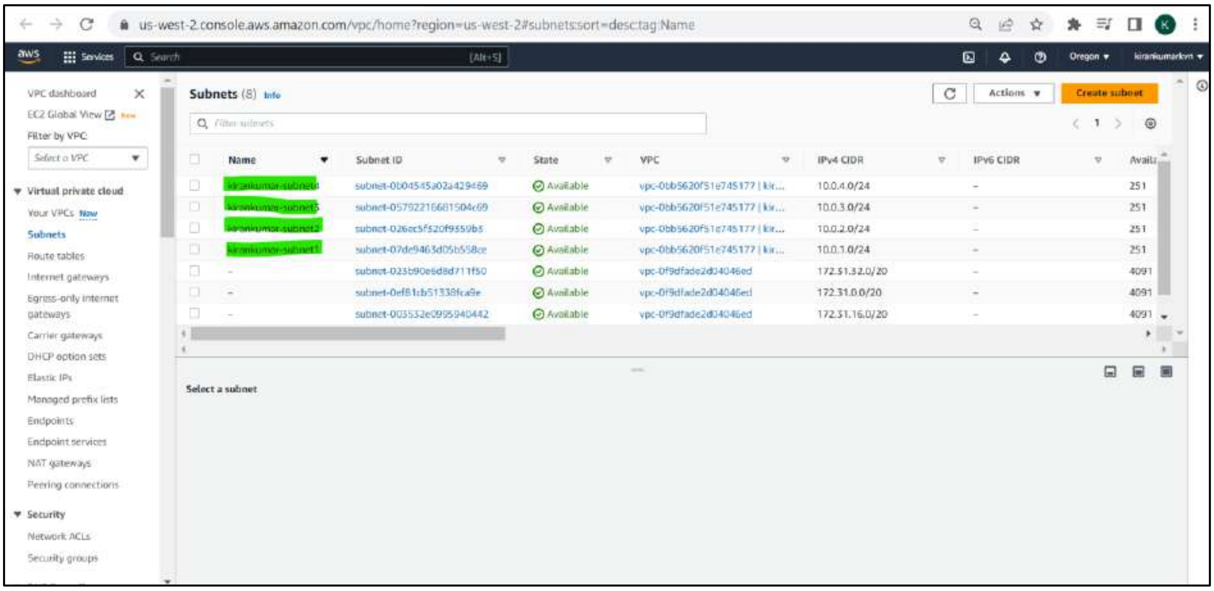
**Subnet settings**
Specify the CIDR blocks and Availability Zone for the subnet.

**Subnet 1 of 1**

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
kirankumar-subnet4
The name can be up to 256 characters long.

Availability Zone
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US West (Oregon) / us-west-2d

IPv4 CIDR block
10.0.4.0/24



us-west-2.console.aws.amazon.com/vpc/home?region=us-west-2#subnets:sort=desc:tag:Name

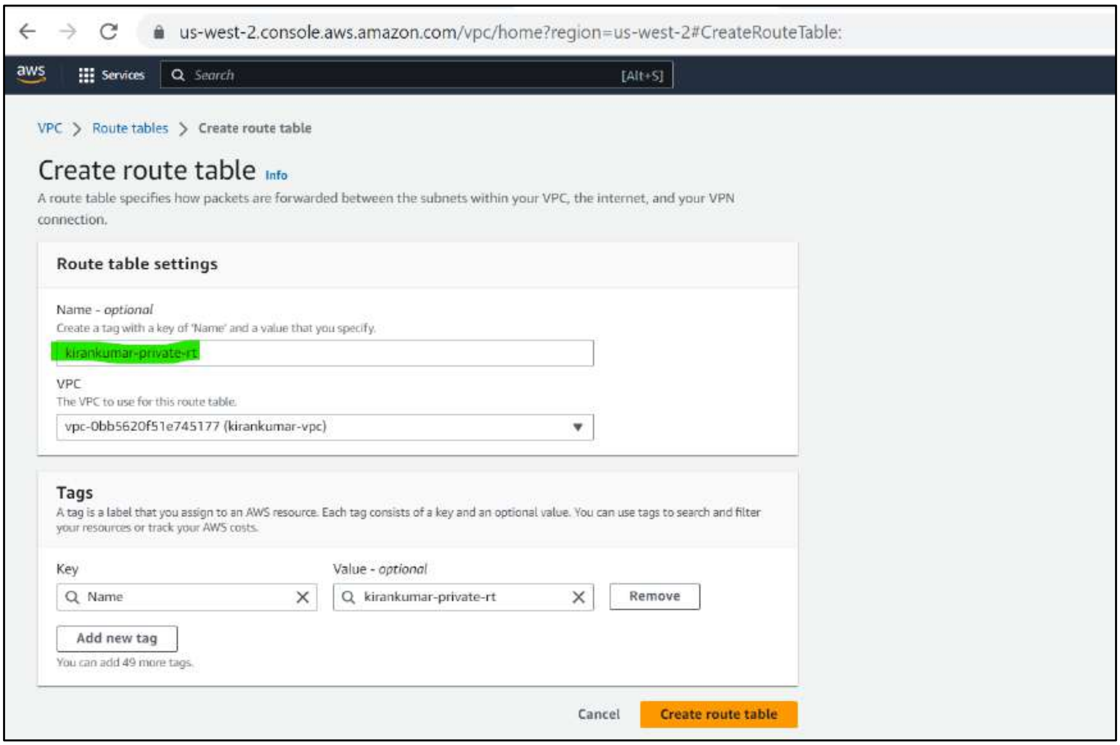
**Subnets** (8)

| Name | Subnet ID | State | VPC | IPv4 CIDR | IPv6 CIDR | Availa |
|---|---|---|---|---|---|---|
| kirankumar-subnet4 | subnet-0b04545a02a429469 | Available | vpc-0bb5620f51e745177 \| kir... | 10.0.4.0/24 | – | 251 |
| kirankumar-subnet3 | subnet-05792216681504c69 | Available | vpc-0bb5620f51e745177 \| kir... | 10.0.3.0/24 | – | 251 |
| kirankumar-subnet2 | subnet-026ec5f520f9359b5 | Available | vpc-0bb5620f51e745177 \| kir... | 10.0.2.0/24 | – | 251 |
| kirankumar-subnet1 | subnet-07de9463d05b558ce | Available | vpc-0bb5620f51e745177 \| kir... | 10.0.1.0/24 | – | 251 |
| – | subnet-023b90e6d8d711f50 | Available | vpc-0f9dfade2d04046ed | 172.31.32.0/20 | – | 4091 |
| – | subnet-0ef81cb51338fca9e | Available | vpc-0f9dfade2d04046ed | 172.31.0.0/20 | – | 4091 |
| – | subnet-003532e0995940442 | Available | vpc-0f9dfade2d04046ed | 172.31.16.0/20 | – | 4091 |

Select a subnet

**4. Created Two Route Tables**: Route tables is a set of rules that determines how data moves within our network. We need two route tables; private route table and public route table. The public route table will define which subnets that will have direct access to the internet ( i.e., public subnets) while the private route table will define which subnet goes through the NAT gateway (i.e., private subnet).

To create route tables, I navigated over to the 'Route Tables' page and clicked on 'Create route table' button.



us-west-2.console.aws.amazon.com/vpc/home?region=us-west-2#CreateRouteTable:

VPC > Route tables > Create route table

Create route table

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

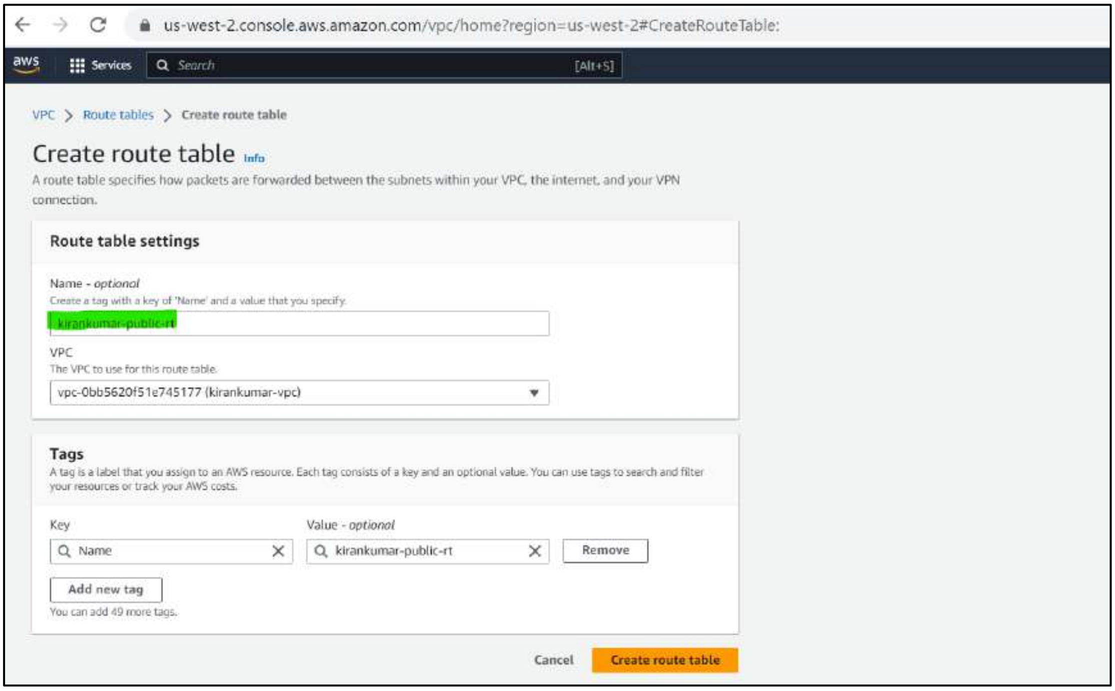
Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
kirankumar-private-rt

VPC
The VPC to use for this route table.
vpc-0bb5620f51e745177 (kirankumar-vpc)

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

| Key | Value - *optional* |
|---|---|
| Name | kirankumar-private-rt |

You can add 49 more tags.



us-west-2.console.aws.amazon.com/vpc/home?region=us-west-2#CreateRouteTable:

VPC > Route tables > Create route table

Create route table

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
kirankumar-public-rt

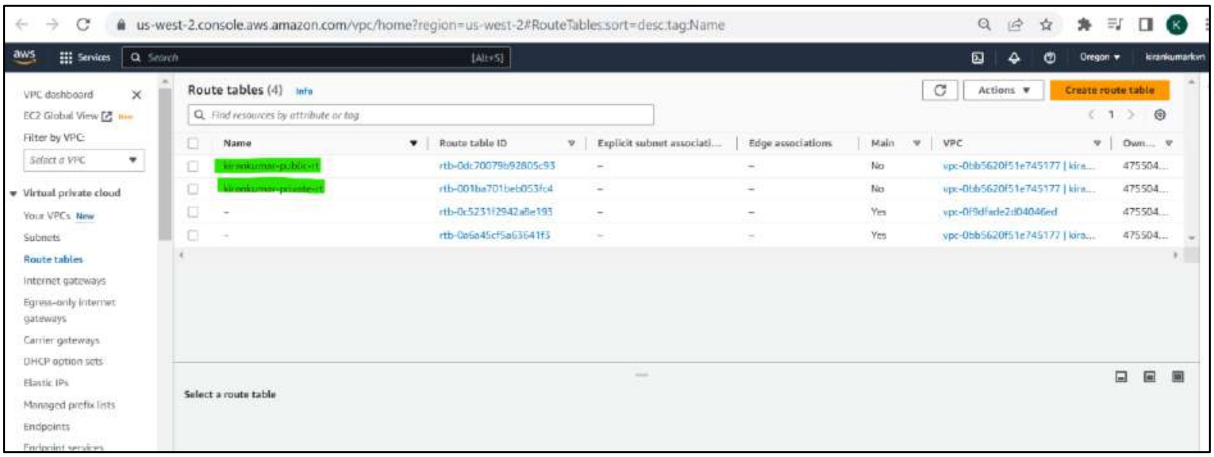
VPC
The VPC to use for this route table.
vpc-0bb5620f51e745177 (kirankumar-vpc)

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

| Key | Value - *optional* |
|---|---|
| Name | kirankumar-public-rt |

You can add 49 more tags.

The public and the private subnet needs to be associated with the public and the private route table respectively. To do that, I selected the route table and then chose the Subnet Association tab.



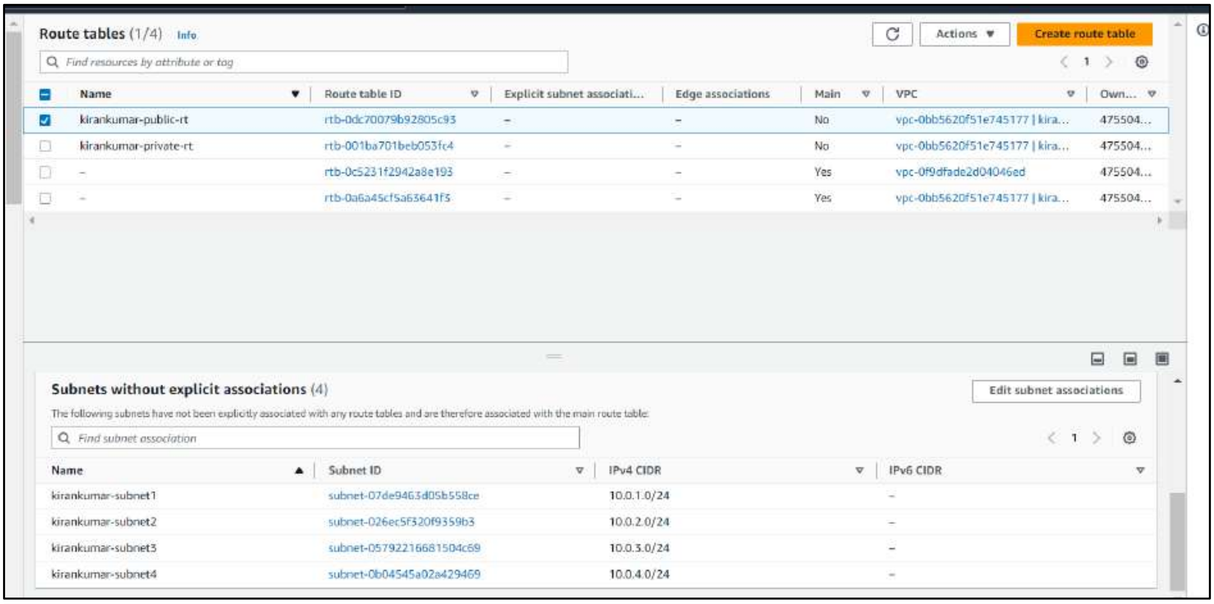
us-west-2.console.aws.amazon.com/vpc/home?region=us-west-2#RouteTables:sort=desc:tag:Name

Route tables (4) Info

Find resources by attribute or tag

| □ | Name | Route table ID | Explicit subnet associati... | Edge associations | Main | VPC | Own... |
|---|---|---|---|---|---|---|---|
| □ | kirankumar-public-rt | rtb-0dc70079b92805c93 | – | – | No | vpc-0bb5620f51e745177 \| kira... | 475504... |
| □ | kirankumar-private-rt | rtb-001ba701beb053fc4 | – | – | No | vpc-0bb5620f51e745177 \| kira... | 475504... |
| □ | - | rtb-0c5231f2942a8e193 | – | – | Yes | vpc-0f9dfade2d04046ed | 475504... |
| □ | - | rtb-0a6a45cf5a63641f3 | – | – | Yes | vpc-0bb5620f51e745177 \| kira... | 475504... |

Select a route table



Route tables (1/4) Info

Find resources by attribute or tag

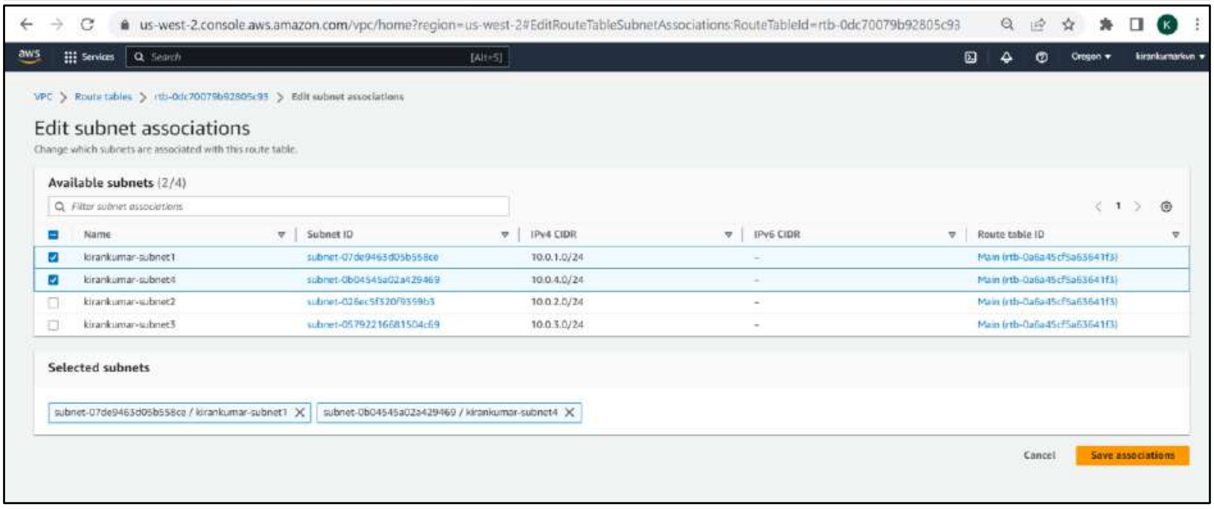
| ▣ | Name | Route table ID | Explicit subnet associati... | Edge associations | Main | VPC | Own... |
|---|---|---|---|---|---|---|---|
| ☑ | kirankumar-public-rt | rtb-0dc70079b92805c93 | – | – | No | vpc-0bb5620f51e745177 \| kira... | 475504... |
| □ | kirankumar-private-rt | rtb-001ba701beb053fc4 | – | – | No | vpc-0bb5620f51e745177 \| kira... | 475504... |
| □ | – | rtb-0c5231f2942a8e193 | – | – | Yes | vpc-0f9dfade2d04046ed | 475504... |
| □ | – | rtb-0a6a45cf5a63641f5 | – | – | Yes | vpc-0bb5620f51e745177 \| kira... | 475504... |

Subnets without explicit associations (4)

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Find subnet association

| Name | Subnet ID | IPv4 CIDR | IPv6 CIDR |
|---|---|---|---|
| kirankumar-subnet1 | subnet-07de9463d05b558ce | 10.0.1.0/24 | – |
| kirankumar-subnet2 | subnet-026ec5f320f9359b3 | 10.0.2.0/24 | – |
| kirankumar-subnet3 | subnet-05792216681504c69 | 10.0.3.0/24 | – |
| kirankumar-subnet4 | subnet-0b04545a02a429469 | 10.0.4.0/24 | – |



us-west-2.console.aws.amazon.com/vpc/home?region=us-west-2#EditRouteTableSubnetAssociations:RouteTableId=rtb-0dc70079b92805c93

VPC > Route tables > rtb-0dc70079b92805c93 > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (2/4)

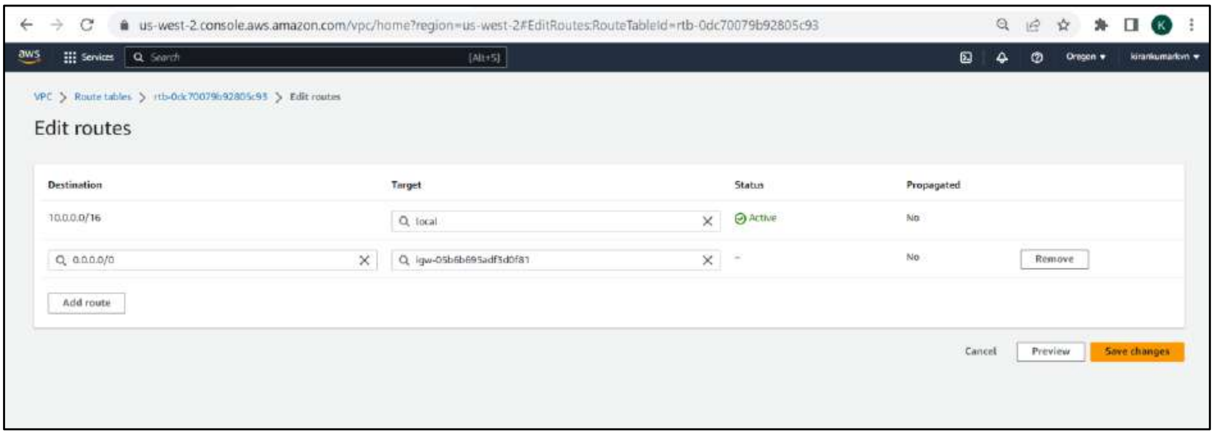
Filter subnet associations

| ▣ | Name | Subnet ID | IPv4 CIDR | IPv6 CIDR | Route table ID |
|---|---|---|---|---|---|
| ☑ | kirankumar-subnet1 | subnet-07de9463d05b558ce | 10.0.1.0/24 | – | Main (rtb-0a6a45cf5a63641f3) |
| ☑ | kirankumar-subnet4 | subnet-0b04545a02a429469 | 10.0.4.0/24 | – | Main (rtb-0a6a45cf5a63641f3) |
| □ | kirankumar-subnet2 | subnet-026ec5f320f9359b3 | 10.0.2.0/24 | – | Main (rtb-0a6a45cf5a63641f3) |
| □ | kirankumar-subnet3 | subnet-05792216681504c69 | 10.0.3.0/24 | – | Main (rtb-0a6a45cf5a63641f3) |

Selected subnets

subnet-07de9463d05b558ce / kirankumar-subnet1 ✕ subnet-0b04545a02a429469 / kirankumar-subnet4 ✕

We also need to route the traffic to the internet through the internet gateway for our public route table. To do that, I selected the ‘public route table’ and then chose the ‘Routes’ tab :



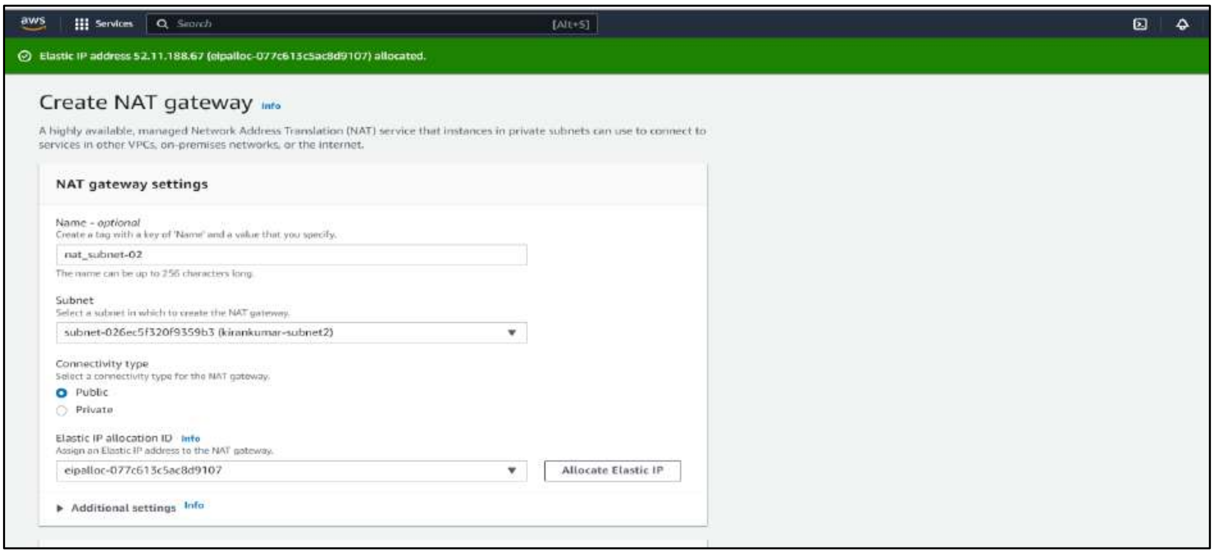
us-west-2.console.aws.amazon.com/vpc/home?region=us-west-2#EditRoutes:RouteTableId=rtb-0dc70079b92805c93

VPC > Route tables > rtb-0dc70079b92805c93 > Edit routes

Edit routes

| Destination | Target | Status | Propagated | |
|---|---|---|---|---|
| 10.0.0.0/16 | local | Active | No | |
| 0.0.0.0/0 | igw-05b6b695adf3d0f81 | - | No | Remove |

**5. Create the NAT Gateway:** The NAT gateway enables the EC2 instances in the private subnet to access the internet. The NAT Gateway is an AWS managed service for the NAT instance. To create a NAT gateway, I navigated to the NAT Gateways page, and then clicked on the Create NAT Gateway.



Elastic IP address 52.11.188.67 (eipalloc-077c613c5ac8d9107) allocated.

Create NAT gateway Info

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

NAT gateway settings

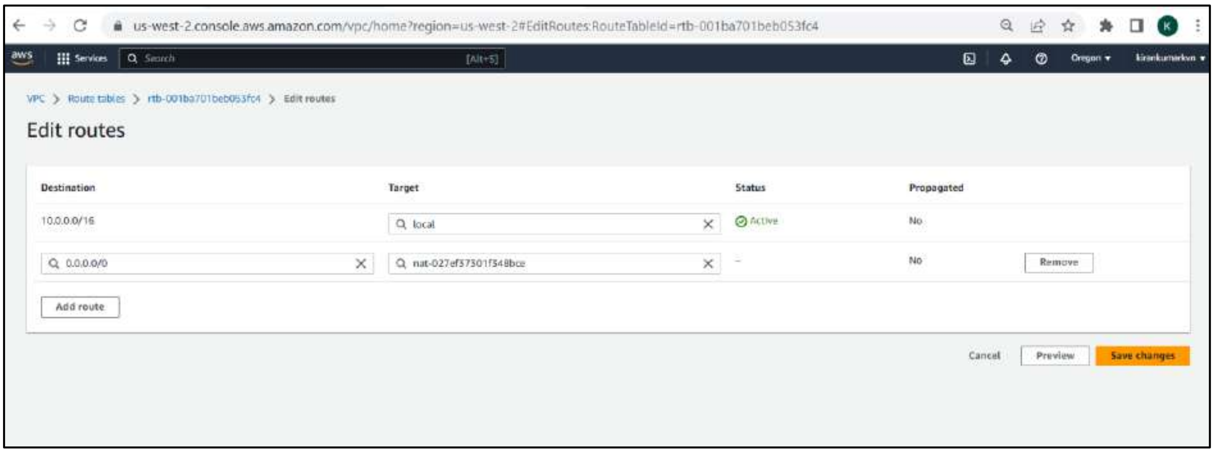
Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
nat_subnet-02
The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.
subnet-026ec5f320f9359b3 (kirankumar-subnet2)

Connectivity type
Select a connectivity type for the NAT gateway.
● Public
○ Private

Elastic IP allocation ID Info
Assign an Elastic IP address to the NAT gateway.
eipalloc-077c613c5ac8d9107

Then, I edited the private route table to make use of the NAT gateway to access the internet.



us-west-2.console.aws.amazon.com/vpc/home?region=us-west-2#EditRoutes:RouteTableId=rtb-001ba701beb053fc4

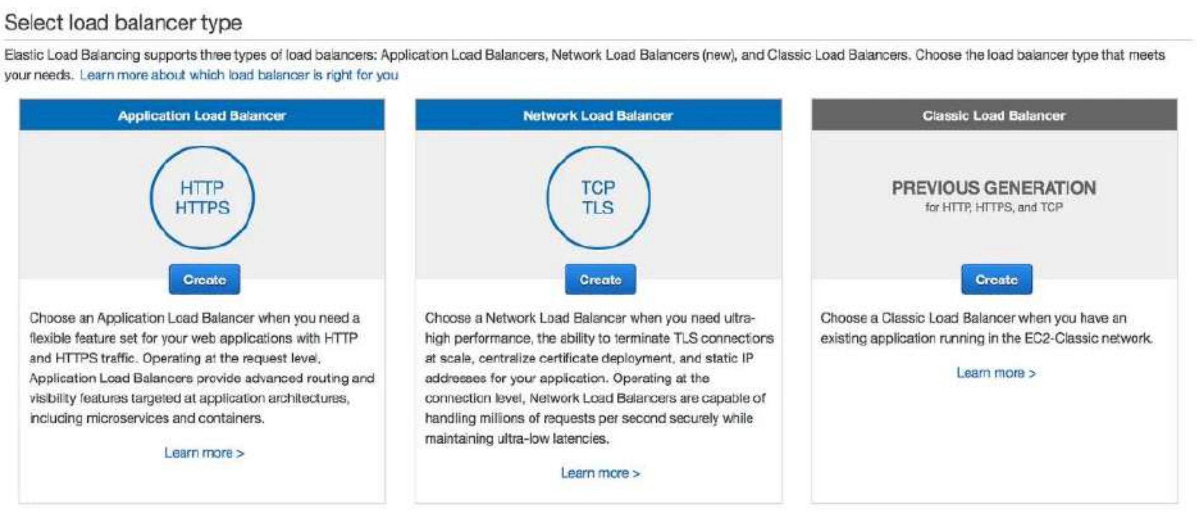
VPC > Route tables > rtb-001ba701beb053fc4 > Edit routes

Edit routes

| Destination | Target | Status | Propagated | |
|---|---|---|---|---|
| 10.0.0.0/16 | local | Active | No | |
| 0.0.0.0/0 | nat-027ef37301f348bce | - | No | Remove |

**6. Create Elastic Load Balancer:** As per my design, my frontend will can only accept traffic from the elastic load balancer which connects directly with the internet gateway while my backend will receive traffic through the internal load balancer. The load balancer distributes the load across the EC2 instances serving that application. If the application uses sessions, sessions can be stored in either the Elastic Cache or the DynamoDB. To create the two load balancers, I navigated to the Load Balancer page and clicked on Create Load Balancer.

a. Select the Application Load Balancer



Select load balancer type

Elastic Load Balancing supports three types of load balancers: Application Load Balancers, Network Load Balancers (new), and Classic Load Balancers. Choose the load balancer type that meets your needs. Learn more about which load balancer is right for you

Choose an Application Load Balancer when you need a flexible feature set for your web applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Choose a Network Load Balancer when you need ultra-high performance, the ability to terminate TLS connections at scale, centralize certificate deployment, and static IP addresses for your application. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

Choose a Classic Load Balancer when you have an existing application running in the EC2-Classic network.

Select Application Load Balancer

b. Gave name to the Load Balancer, one for internet facing that communicates with the frontend and the internal for backend.
c. Under Availability Zone, for the internet facing Load Balancer, I selected the two public subnets while for my internal Load Balancer, I selected the two private subnets.
d. Under the Security Group, I only allowed the ports that my application needs. I allowed HTTP port 80 & HTTPS port 443 on my internet facing load balancer. For my internal load balancer, I only opened the port that the backend runs on, which is port 3000. I made this port only open to the security group of the frontend which allowed only the frontend to have access to this port.
e. Under the Configure Routing, I configured my Target Group to have the Target type of instance which is needed to create my Auto Scaling Group.
f. I skipped the Register Targets & then clicked on the Create button.

**7. 'Auto Scaling' Group**: With Auto Scaling Group, I adjusted the size of EC2 instances. To create an Auto Scaling Group, I navigated to the Auto Scaling Group page, clicked on the 'Create Auto Scaling Group' button.

a. Under the Configure details, gave the Launch Configuration a name called : kirankumar-Frontend-LC.
b. Under the security group, I allowed only the ports 8080, 443 & 3000.
c. I reviewed the Configuration and clicked on 'Create Launch Configuration'. Then I created & downloaded a new key pair.
d. Under the Configure scaling policies, I added one instance when the CPU >= 80% & to scale down when the CPU <= 50%.
e. After all this setup, click on 'Create Auto Scaling group' button.

**8. Bastion Host:** The bastion host is just an EC2 instance that sits in the public subnet. The best practice is to only allow SSH to this instance from your trusted IP. To create a bastion host, I navigated to the EC2 instance page and created an EC2 instance (with public IP) in the kirankumar-public-subnet-1 subnet within my VPC. I allowed SSH from my private instances from the Bastion Host.

This is how, with manual configurations, I setup a three-tier architecture in AWS.