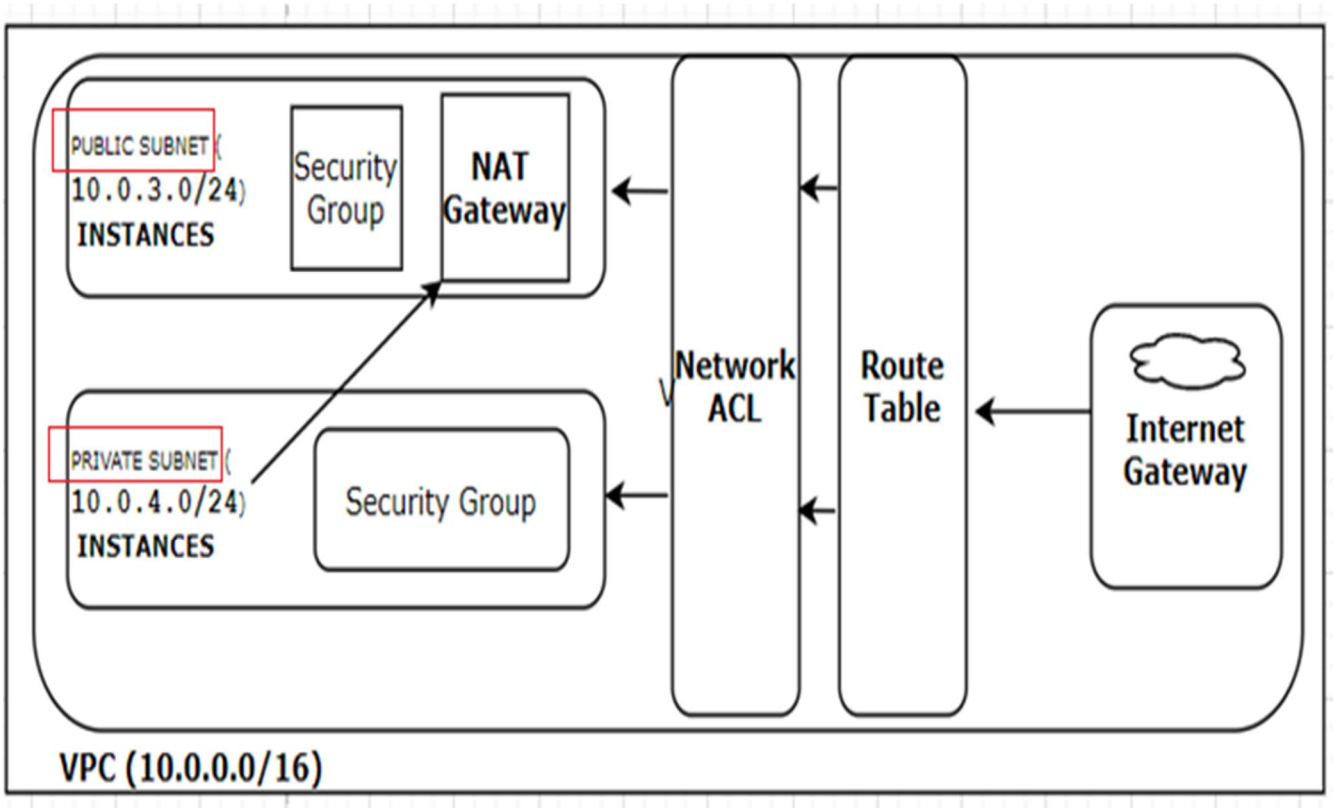
Problem Resolution:

1. Create a VPC with CIDR (10.0.0.0/16).
2. Create public subnet (10.0.3.0/24) need to be accessible from outside. And a web server instances need to attach in this subnet which need to access from outside.
3. Create private subnet (10.0.4.0/24) need to connect with NAT gateway. And a MySQL DB server will create here. This subnet can't be access from outside directly.
4. In all cases there need to be custom Network ACL and Route table and Internet Gateway instead of default.

Diagram for VPC Cloud Setup:

Below is the diagram for proposed solution of the problem statement.



Implementation:

Step1: Create your VPC

Login to your AWS account, From the Services Tab → Select VPC →then Select Your VPC → click on "**Create VPC**" → specify the followings

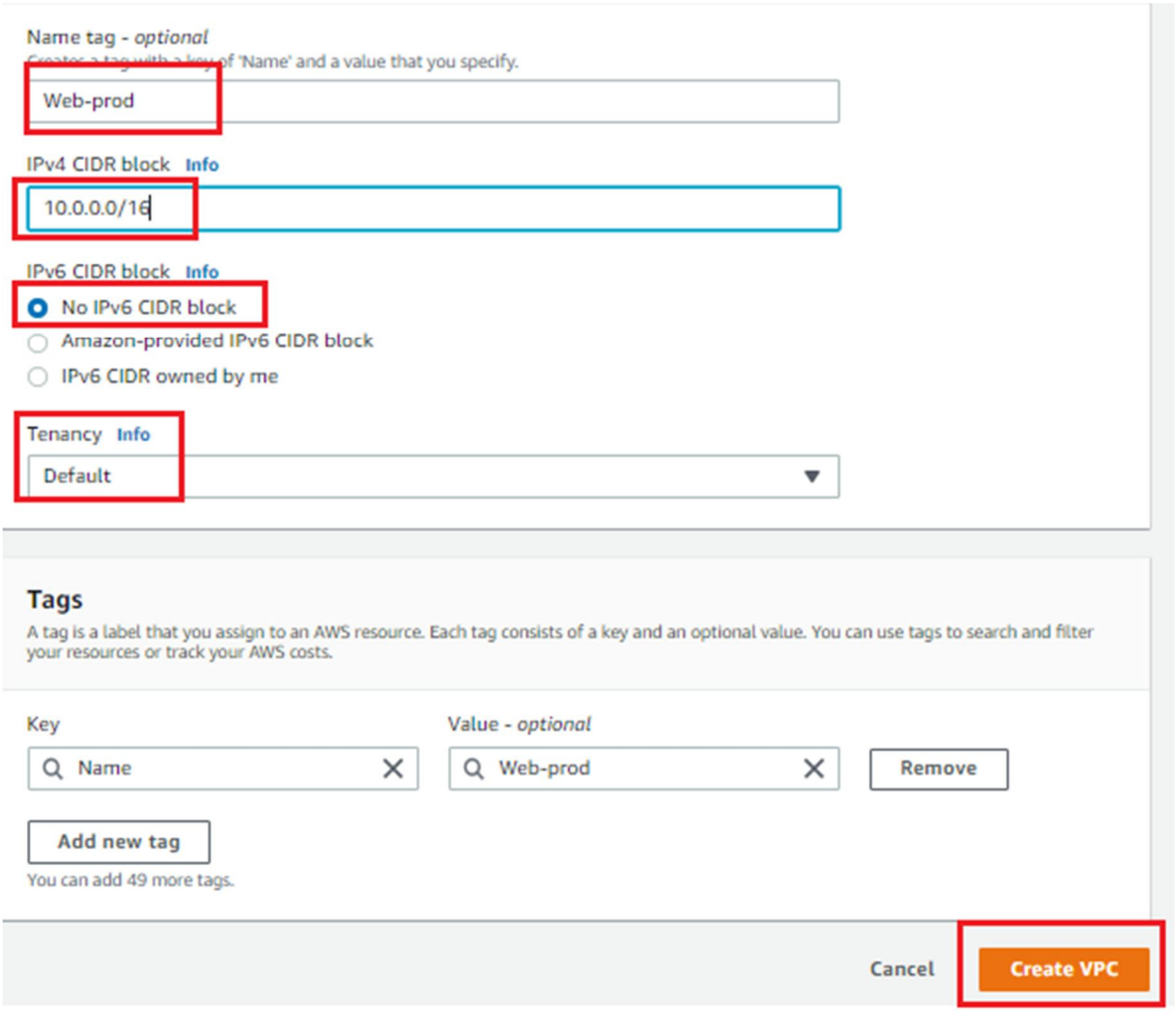
VPC Name = **Web-Prod**

IPV4 CIDR = **10.0.0.0/16**

No IPv6 CIDR block

Tenancy = **Default**

Click on "**Yes,Create**" option



Step 2: Create Subnets

From the **VPC Dashboard** click on **Subnets** option and then click on Create TWO **Subnet**

Create Prod-Public subnet

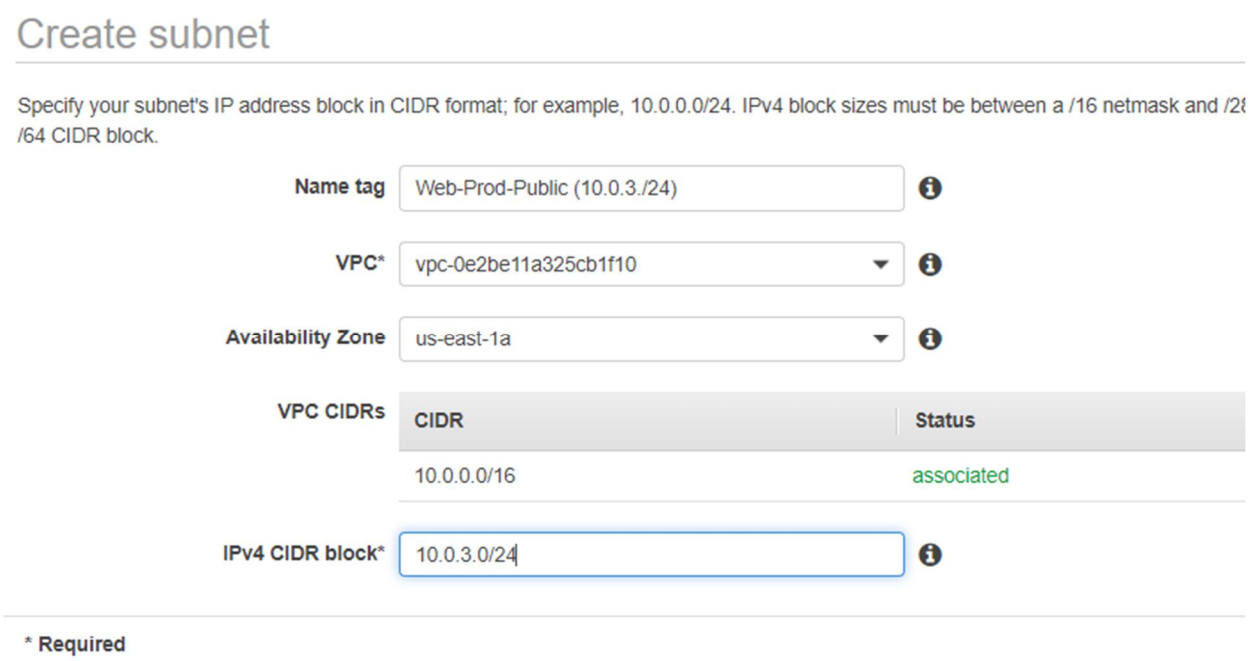
Name Tag = **Web-Prod-Public (10.0.3.0/24)**

vpc = **vpc-06b15f1d4d924b0a9 Web-prod**

Availability Zone = **us-east-1a**

IPv4 CIDR block = **10.0.3.0/24**

Click on "**Yes,Create**" option



Create Prod-Private subnet

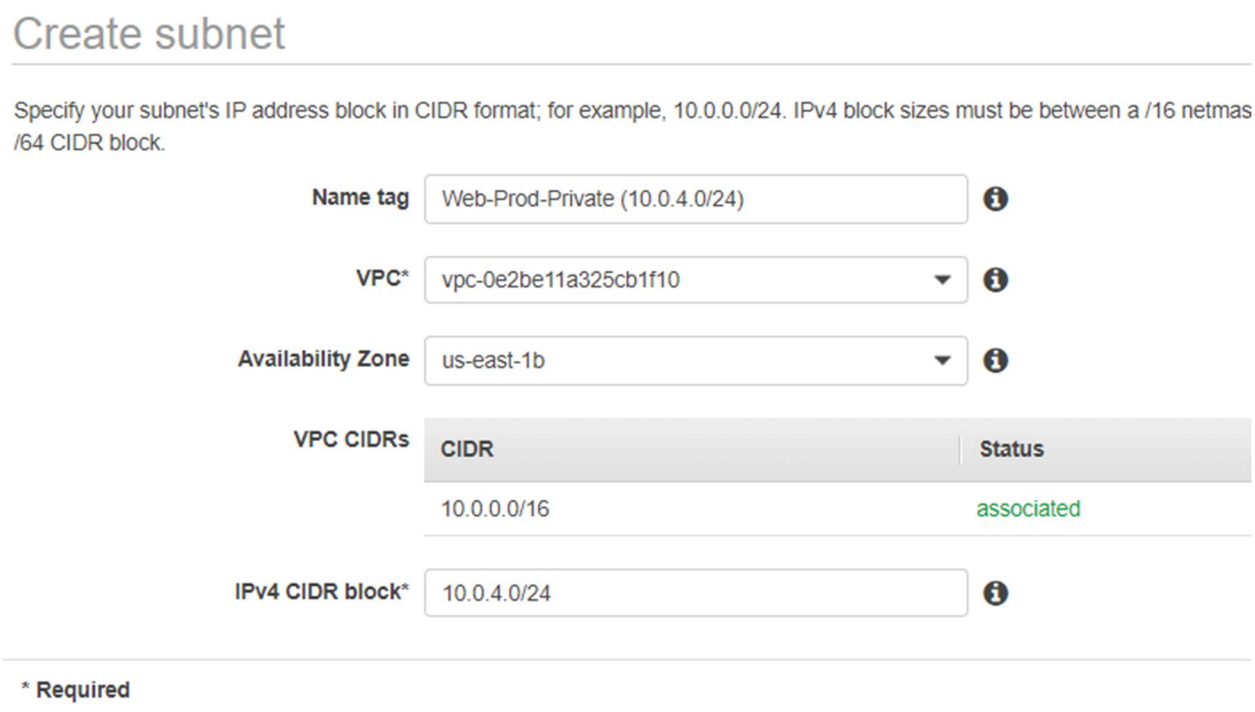
Name Tag = **Web-Prod-Private (10.0.4.0/24)**

vpc = **vpc-06b15f1d4d924b0a9 Web-prod**

Availability Zone = **us-east-1b**

IPv4 CIDR block = **10.0.4.0/24**

Click on "Yes,Create" option

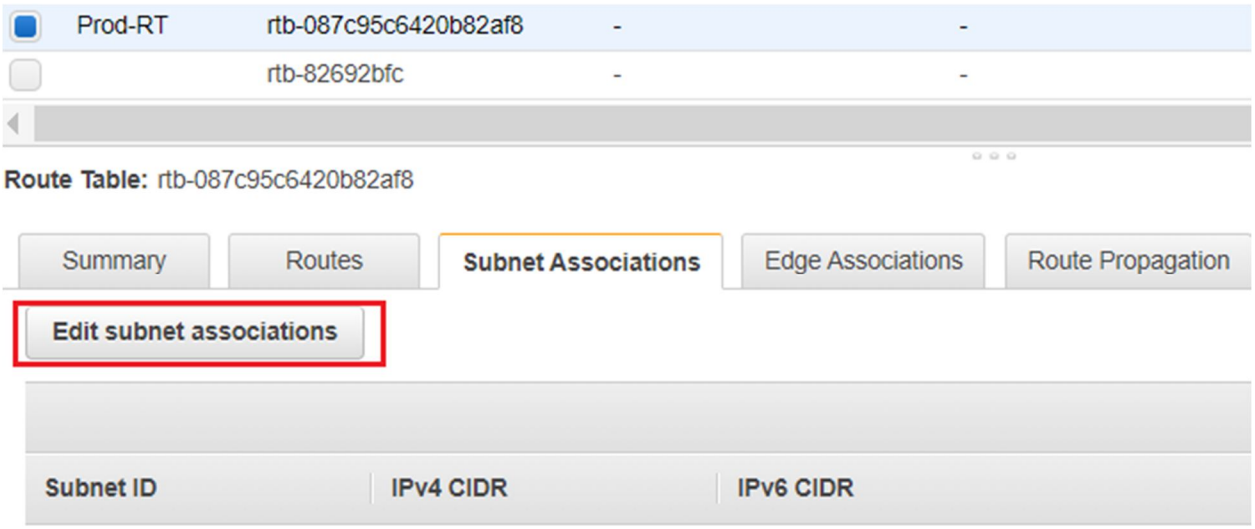


Step 3 Create a Route table and Associate it with your VPC

From VPC Dashboard there is an option create a Route table. Click on "**Create Route Table**" and specify the followings:

Name tag = **Prod-RT**

VPC = **Web-Prod**



Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-084aadbee52f3fb2...	10.0.3.0/24	-

The following subnets have not been explicitly associated with any route tables and are ther

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-0e4a70918f7e0c5...	10.0.4.0/24	-

Step 4: Create Internet Gateway (igw) and attached it to your VPC and modify the route table

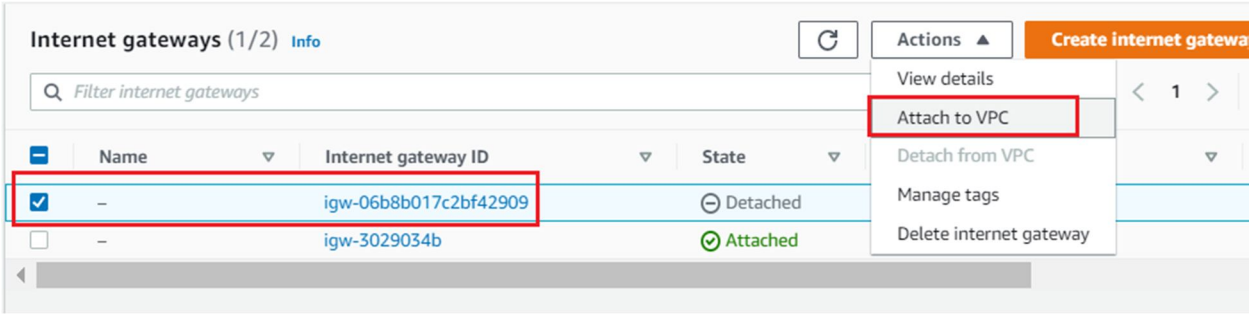
From VPC dashboard there is an option to create Internet gateway. Specify the Name of Internet gateway.

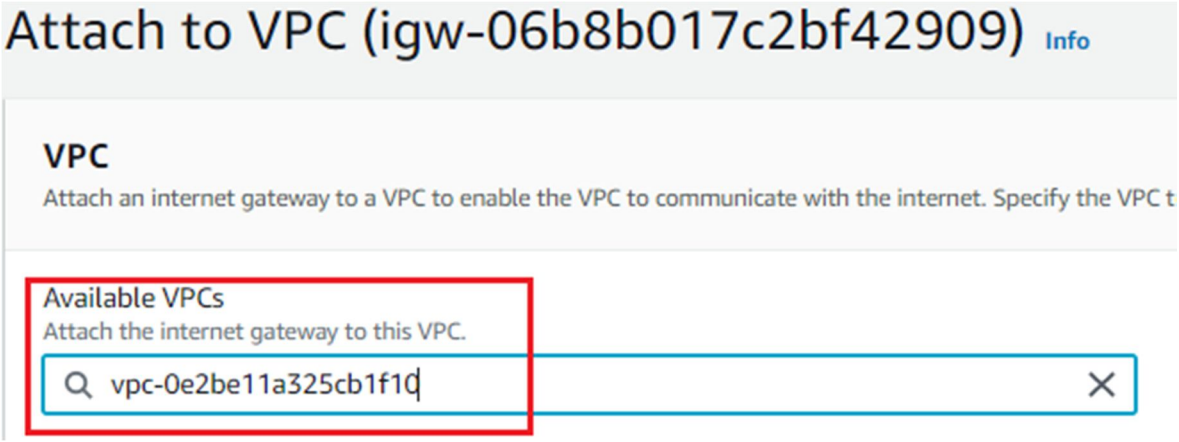
Name tag
Creates a tag with a key of 'Name' and a value that you specify.
my-internet-gateway

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.

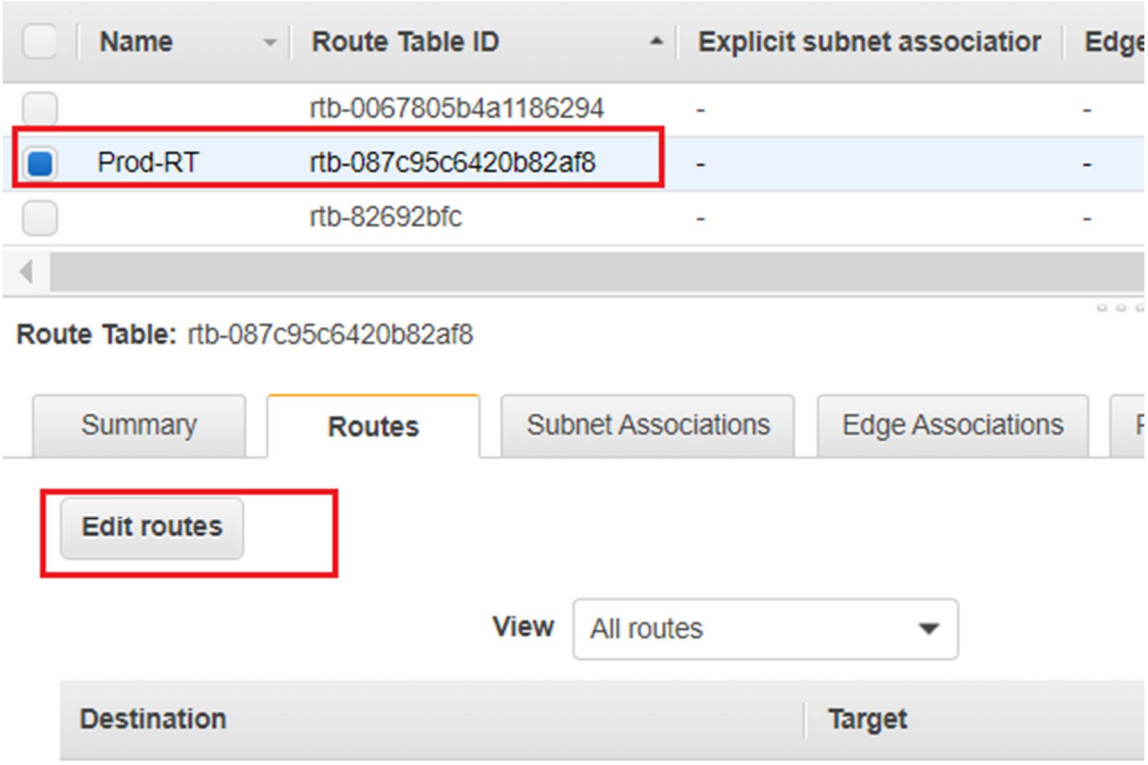
Key	Value - *optional*	
Web-IGW	Web-IGW	Remove

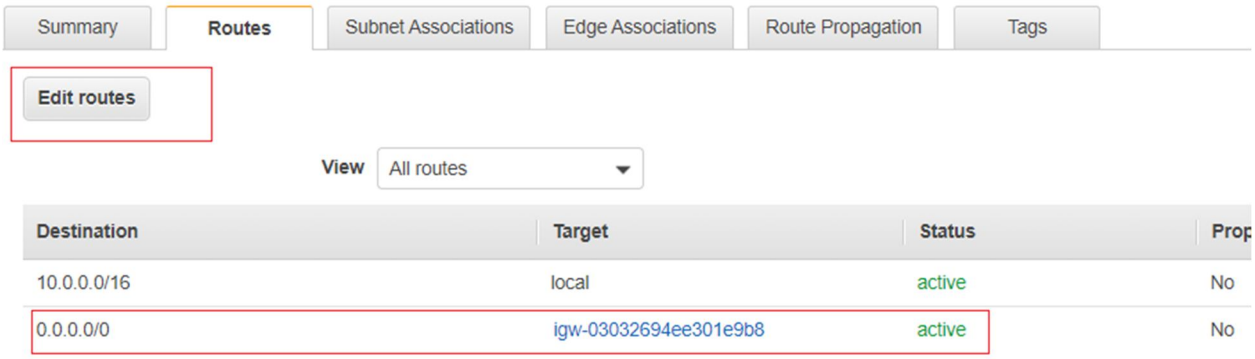
Once the Internet gateway is created, attached it to your VPC, Select and Right Click Your Internet gateway and then Select the "**Attach to VPC**" option and specify your VPC , in here it is **Web-Prod**



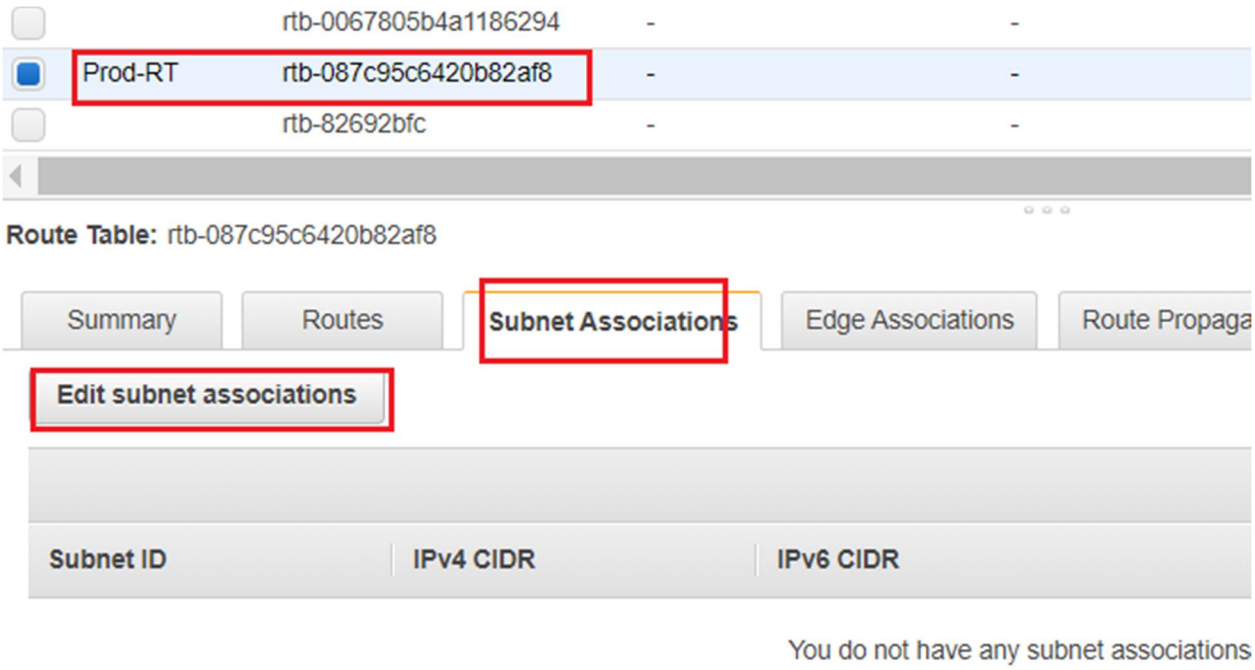


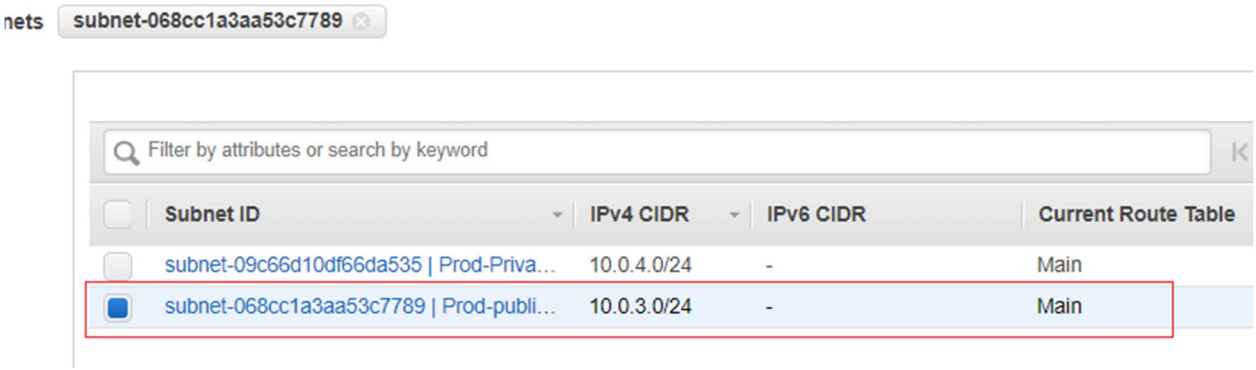
Now Add Route to your route Table for Internet, go to **Route Tables** Option, Select your Route Table, In my case it is "Prod-RT ", click on Route Tab and Click on **Edit** and the click on "**add another route**" Mention Destination IP of Internet as **"0.0.0.0/0"** and in the target option your Internet gateway will be populated automatically





Then again go to Edit Route table option and click on **"Edit subnet association"** and **add subnet 10.0.3.0/24** , then this subnet will be publicly above. But 10.0.4.0/24 will not be publicly available so this subnet will not add here



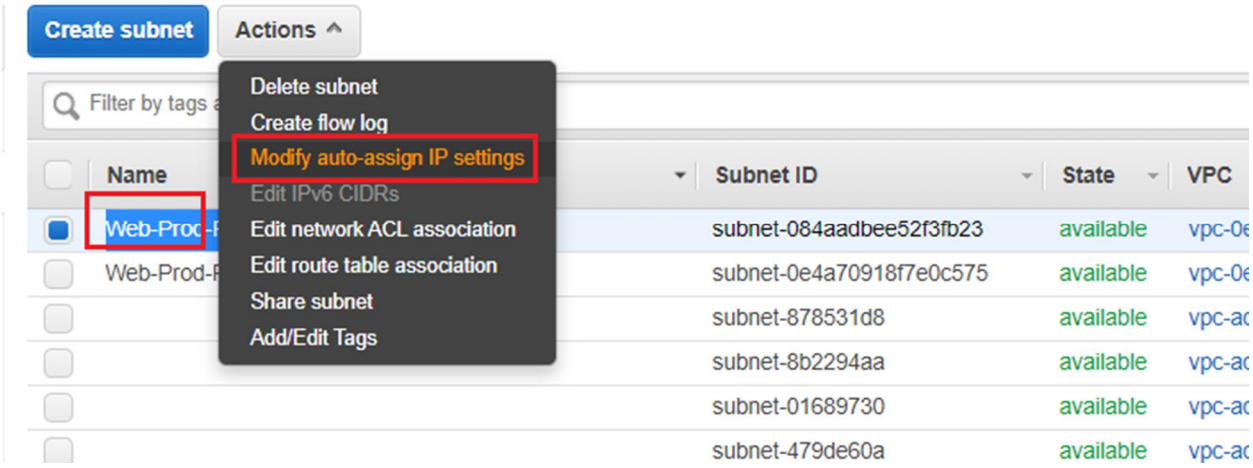


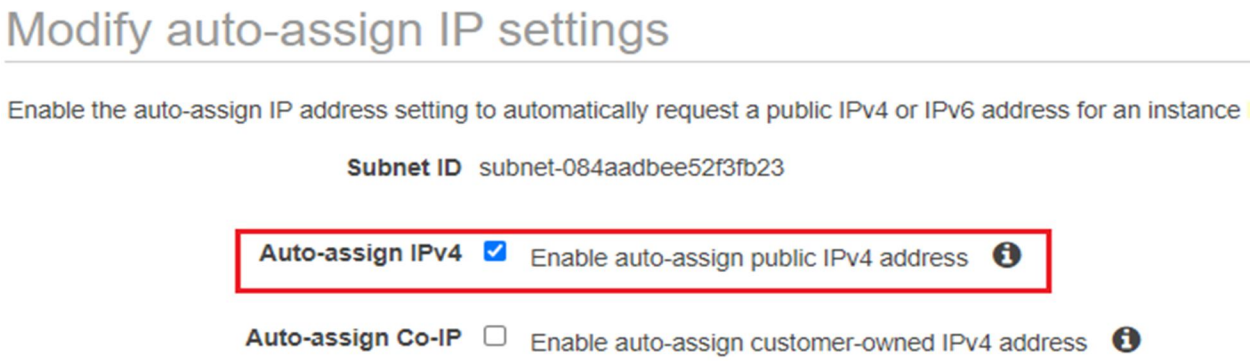
Step 5: Modify IP settings at VPC subnet section for Public Subnet

Services -> VPC -> Subnets

Select Public subnet **Web-Prod-Public (10.0.3./24)** -> **Actions** -> **Modify auto-assign IP settings**

Enable auto-assign public IPv4 address

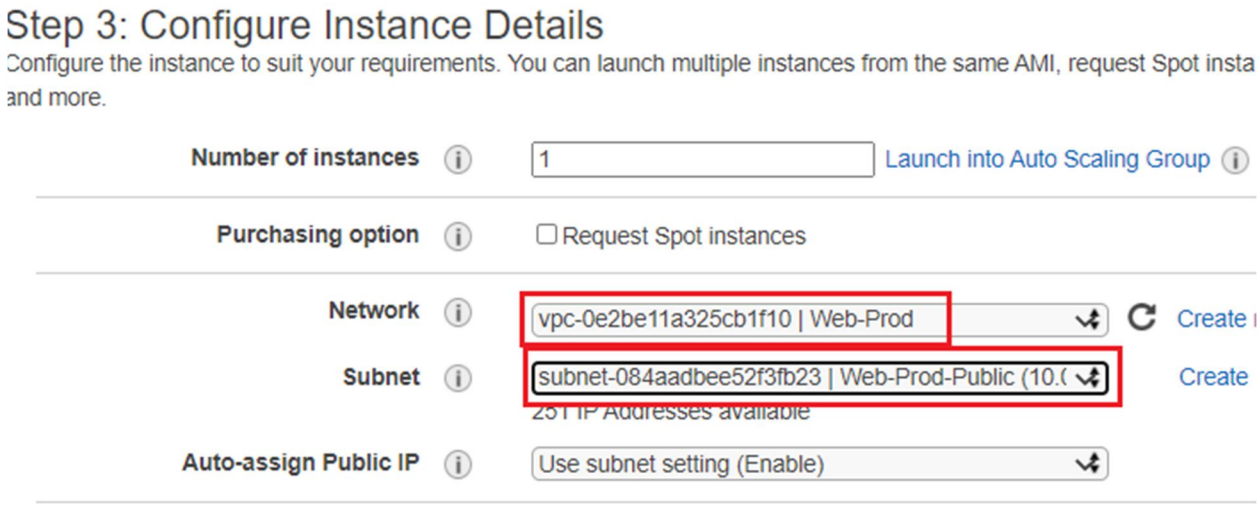




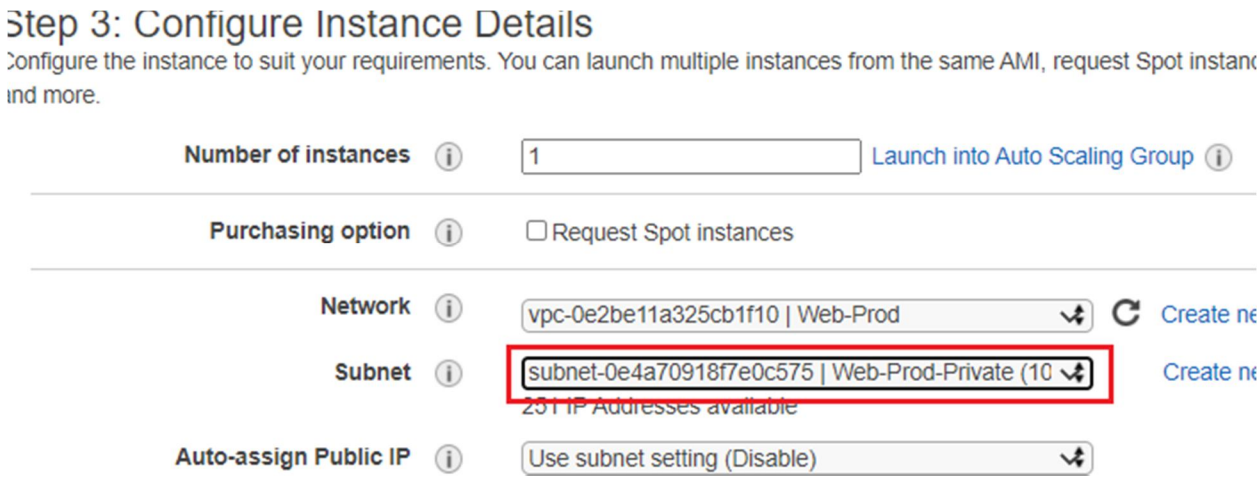
Step 6: Launch Web server and DB Server Instance

Now launch Web server and DB server in EC2 console and associate Web server with public subnet and DB server with private subnet. Also create a new security group with allow port 22, 443 & port 80 also create key pair as per your requirement. In this case key pair is using existing.

Web Server Network and subnet



DB Server Network and subnet



Security group

Assign a security group: ◉ Create a **new** security group

○ Select an **existing** security group

Security group name: vpc-web-init

Description: launch-wizard-3 created 2020-10-11

Type	Protocol	Port Range
SSH	TCP	22
HTTP	TCP	80
HTTPS	TCP	443

Step 7: Login to Web server instances and configure web server
as it is connected and with public ip and has real IP assigned. Configure web server in the instances.

```
asif@DESKTOP-37QUITS:~/.ssh$ ssh ec2-user@3.237.92.91 -i ec2keyp.pem
The authenticity of host '3.237.92.91 (3.237.92.91)' can't be established.
ECDSA key fingerprint is SHA256:wEzuD7Y/OWoEPoUxCuZ44lmPFKiJCGt2MKill+C7FKM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '3.237.92.91' (ECDSA) to the list of known hosts.

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

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 13 available
Run "sudo yum update" to apply all updates.
```

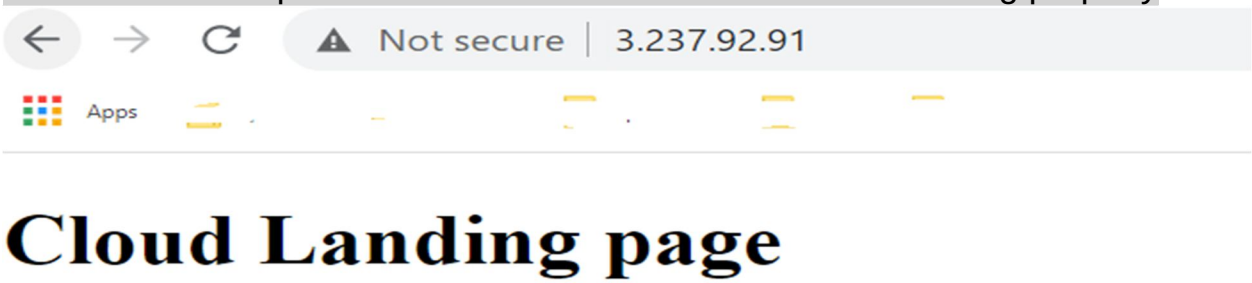
sudo su
yum update –y
yum install httpd –y

cd /var/www/html/
nano index.html
<!DOCTYPE html>

```
<html>
<head>
        <title> Landing page</title>
</head>
<body>
        <h1> Cloud Landing page </h1>
</body>
</html>
```

service httpd start
chkconfig httpd on

Browse instance public IP and cofirm that web server is working properly



Step 8: Login DB server instance and install MySQL

As DB server instances is not connected with public subnet so it can't be connected from outside. So for DB server instance need to create custom security group thus we can login from Web server subnet. Please do followings:

Step 8.1 Configure custom Security group for DB server and attach it with DB instances

Services -> Security Group -> Create Security Group

Name : DBInstancesSecurityGroup

Description: DBInstancesSecurityGroup

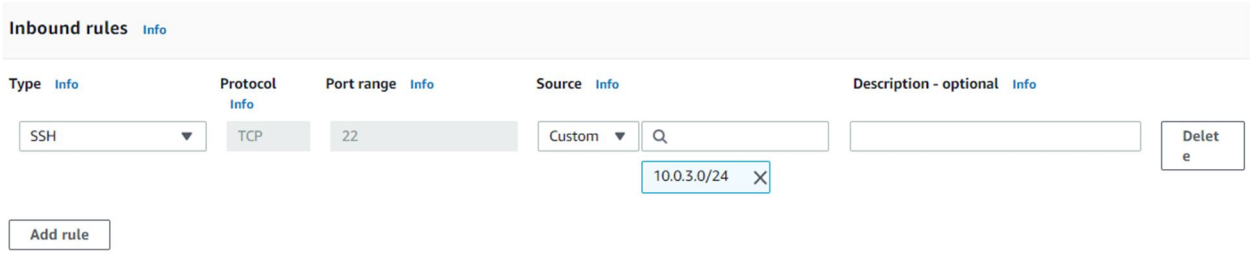
VPC = **Web-prod**

Add inbound Rule

Type = SSH

Port = 22

Custom Source CIDR = 10.0.3.0/24



Then Services -> EC2 -> Click DB Server instance -> Actions -> Networking -> Change Security Group -> Add security group “DBInstancesSecurityGroup” -> Remove default -> Click Save

Now we can login to DB instance private IP from Web server instance as showing below

```
[ec2-user@ip-10-0-3-137 ~]$ ssh -i privatekey.pem ec2-user@10.0.4.94
The authenticity of host '10.0.4.94 (10.0.4.94)' can't be established.
ECDSA key fingerprint is SHA256:5VBnFlSWjHZW4zqcFTVYGU4yAMalb0JAjyUIl8K4DmY.
ECDSA key fingerprint is MD5:d7:3d:4e:b3:2e:14:54:ab:73:bf:d1:ae:61:3e:18:07.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.4.94' (ECDSA) to the list of known hosts.

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

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-4-94 ~]$
```

Step 8.2 Connect DB instance with NAT gateway

Now need to connect DB instance with NAT gateway cause from DB instance internet access is not allowed so from DB instance we can’t install MySQL DB package. Do following to create NAT gateway :

Services -> NAT gateway -> Create NAT gateway -> Then do followings

Name = DBInstance-NATgateway

Subnet = 10.0.3.0/24 (need to select the public subnet)

Click “Allocate Elastic IP”

Click “Create NAT gateway”

It will take sometime to create NAT gateway.

Then go to VPC -> Route Table -> Select routing table which create default -> Edit Route Table and add following

0.0.0.0/0 NAT

Now we will be able to access internet from DB instance via NAT gateway and install MySQL DB there.

```
[root@ip-10-0-4-94 ec2-user]# yum install mysql -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package mariadb.x86_64 1:5.5.64-1.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=================================================================================
```

Step 9: Create Network ACL

Services -> VPC -> Network ACLs > Create network ACL

Name Tag = Web-NACL

Vpc = Web-Prod

Click Web-NACL and Edit inbound rules

Add rule from number 100 and allow ssh , http, https as below :

Details | **Inbound Rules** | Outbound Rules | Subnet associations | Tags

Edit inbound rules

View All rules

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	SSH (22)	TCP (6)	22	0.0.0.0/0	ALLOW
150	HTTP (80)	TCP (6)	80	0.0.0.0/0	ALLOW
200	HTTPS (443)	TCP (6)	443	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

Now Edit outbound Rules and edit port and allow port ssh, http, https and also allow port range 1024-65535 (ephemeral ports) as below picture :

Details | Inbound Rules | **Outbound Rules** | Subnet associations | Tags

Edit outbound rules

View All rules

Rule #	Type	Protocol	Port Range	Destination	Allow / Deny
100	SSH (22)	TCP (6)	22	0.0.0.0/0	ALLOW
150	HTTP (80)	TCP (6)	80	0.0.0.0/0	ALLOW
200	HTTPS (443)	TCP (6)	443	0.0.0.0/0	ALLOW
250	Custom TCP Rule	TCP (6)	1024 - 65535	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

Then click Web-NACL -> Actions -> Edit Subnet Associations -> and add public subnet

So subnet 10.0.3.0/24 is now associated with Web-NACL

Details | Inbound Rules | Outbound Rules | **Subnet associations** | Tags

Edit subnet associations

Filter by tags and attributes or search by keyword

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-068cc1a3aa...	10.0.3.0/24	-

Now we can still browse out landing web page at web server public IP which means NACL is working and allowing traffic to/from web server.

RollBack

- ➔ First terminate instances
- ➔ Delete NAT Gateway
- ➔ De-attach subnet from route table
- ➔ De-attach internet gateway from vpc and delete
- ➔ Delete subnets from vpc
- ➔ Delete VPC