Cloud Plus Plus Services

Configure Custom Amazon VPC and provision web server in public subnet of the VPC.

Objectives:

1. Learn to design and implement custom Amazon VPC.
2. Learn to provision a web server in public subnet.
3. Learn to provision NAT gateway.

Step 1: In AWS Management Console, go to VPC service. In VPC side Panel, click Your VPCs. Your default VPC is visible in this window. Click on Create VPC.

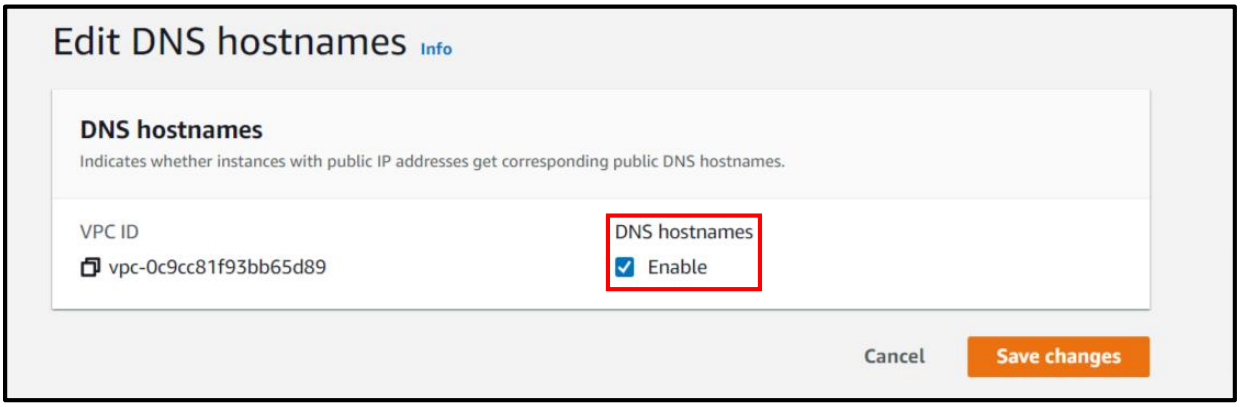
In VPC settings:

Name tag – myVPC

IPv4 CIDR block – 10.0.0.0/16

Keep rest default, scroll down and click on Create VPC. In the next successful creation window, click on Actions -> Edit DNS Hostnames.

Check the Enable box and Save changes.



Step 2: Go back to VPC dashboard. Click on Subnets. Here there would be existing subnets of default VPC. Click on Create Subnets.

Select myVPC from drop down.

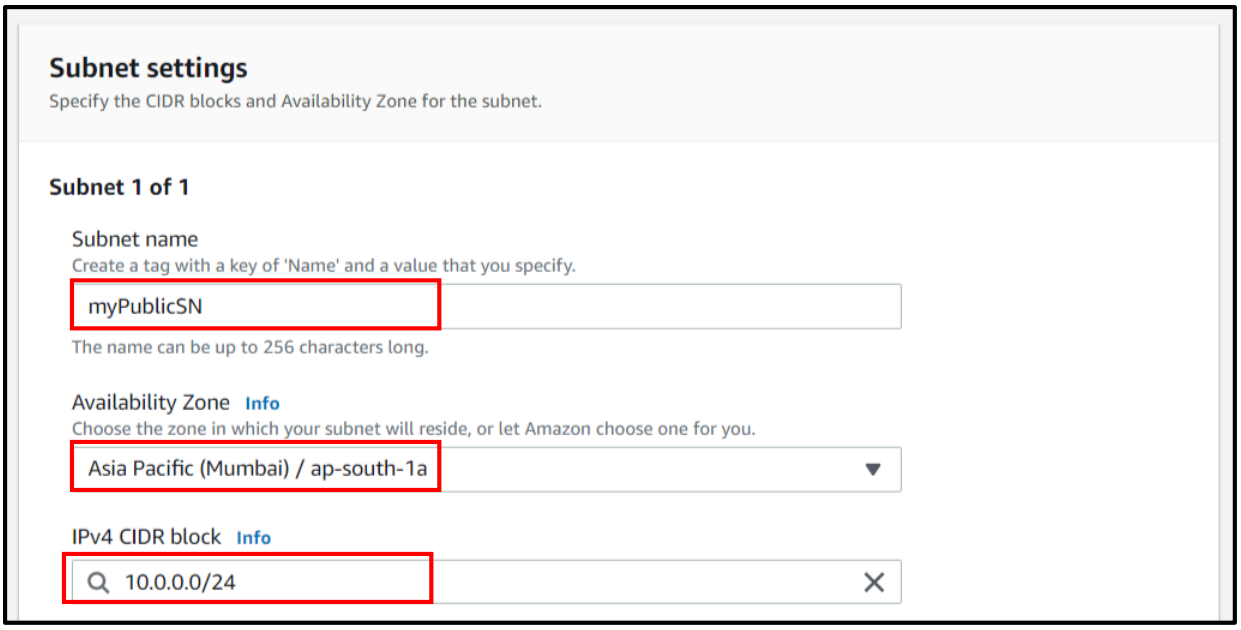
Configure Subnet settings as follows:

Subnet Name: myPublicSN

Availability Zone: ap-south-1a

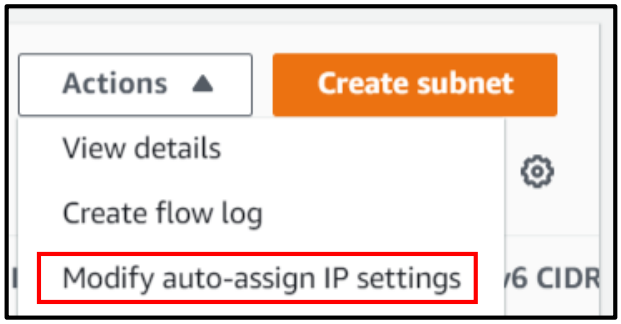
(The above values may vary according to your own region, subnets and requirements)

IPv4 CIDRs: 10.0.0.0/24

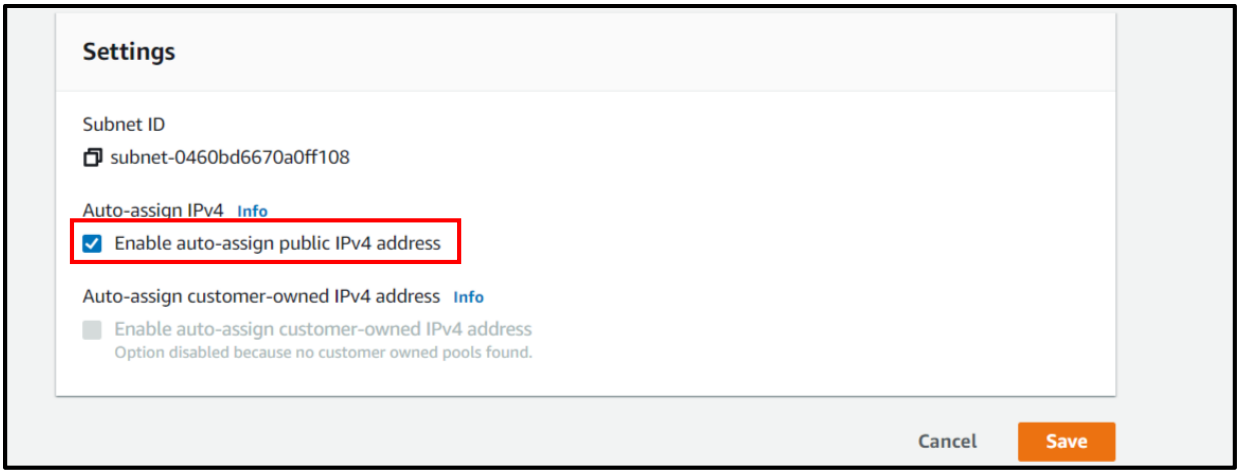


Keep the rest values default, scroll down and click on Create Subnet.

Select the **myPublicSN**, go to Actions -> Modify auto-assign IP settings



Check the **Enable auto assign public IPv4 address**. And click Save.



Go back to **Subnets**, click on **Create Subnets**. Select **myVPC** from drop down.

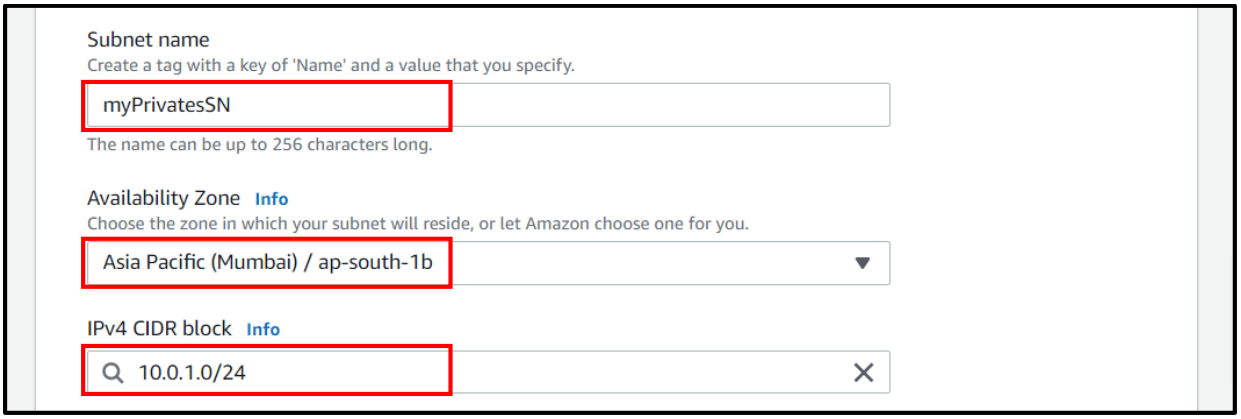
Configure private subnet according to following configurations:

Subnet Name: myPrivateSN

Availability Zone: ap-south-1b

(The above values may vary according to your own region, subnets and requirements)

IPv4 CIDRs: 10.0.1.0/24

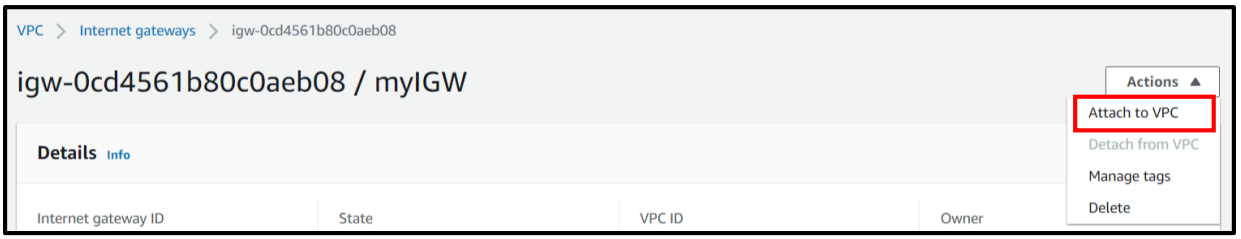


Step 3: In VPC console side panel, go to Internet Gateways. Click on Create internet gateway.

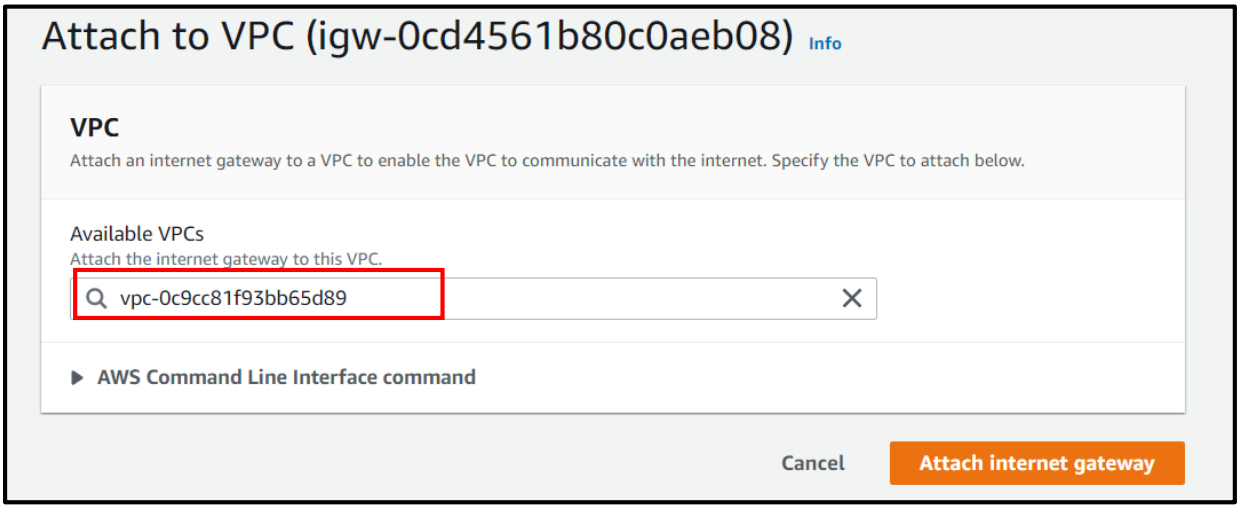
Provide **Name tag** – myIGW.

Keep the defaults as is, scroll down and click on Create internet gateway.

In the successful creation window, click on Actions -> Attach to VPC.



Select myVPC from drop down and click on Attach internet gateway.



Step 4: Go back to VPC service console. Select Route Tables. Here we create 2 route tables.

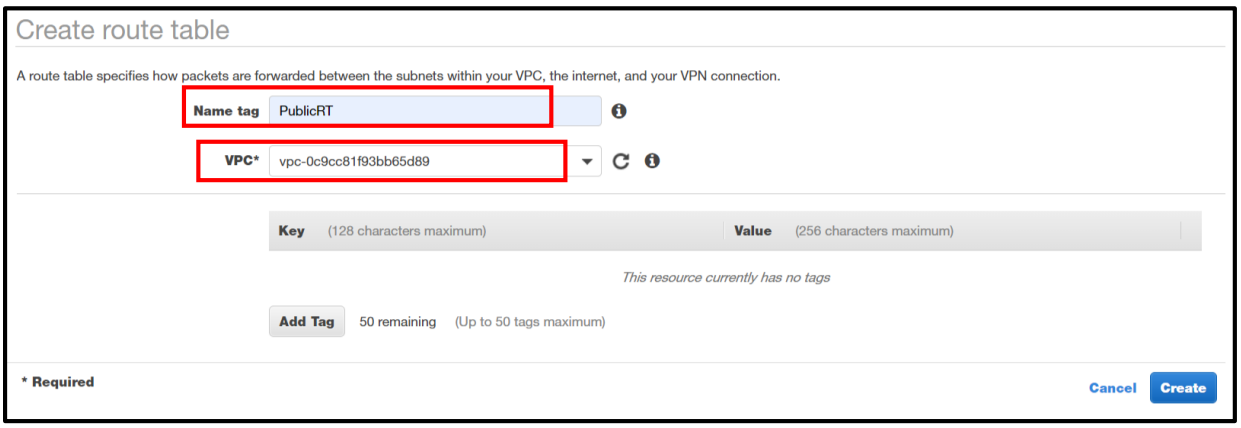
Click on Create route table.

Provide following configurations:

Name Tag – PublicRT

VPC – myVPC (from drop down)

Click on Create button.



Follow the same procedure to create Private route table.

Go back to Route Tables. Click on Create route table.

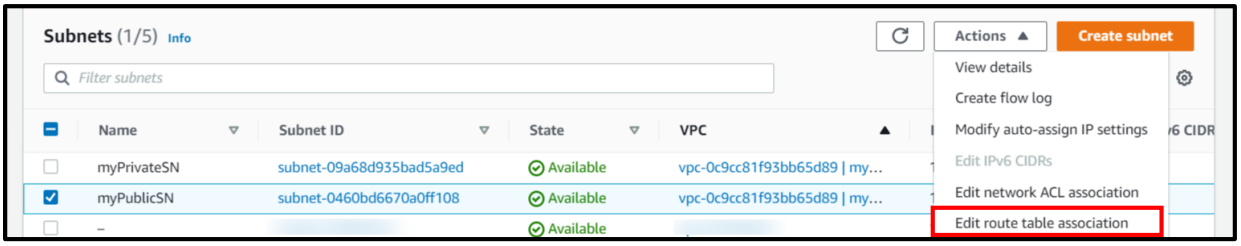
Provide following configurations:

Name Tag – PrivateRT

VPC – myVPC (from drop down)

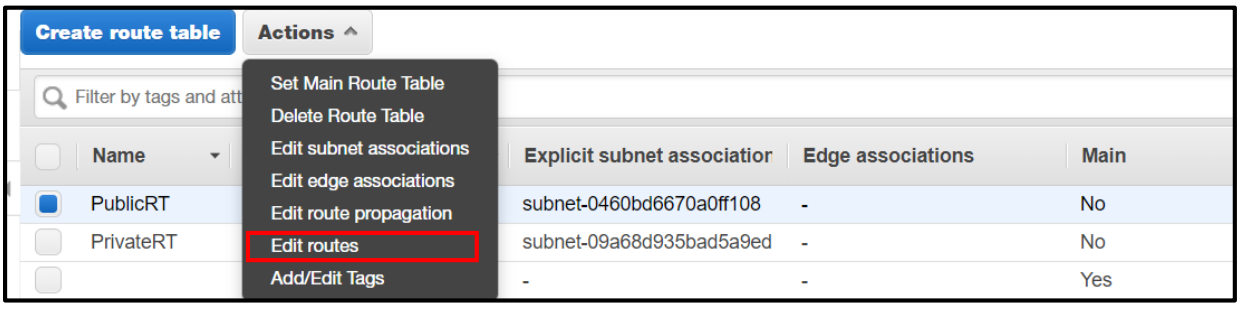
Click on Create button.

Step 5: Go back to **Subnets**. Select **myPublicSN**. Go to Actions -> Edit route table associations.



Select the **PublicRT** from drop down. Click on Save.

Go to Actions -> Edit Routes.

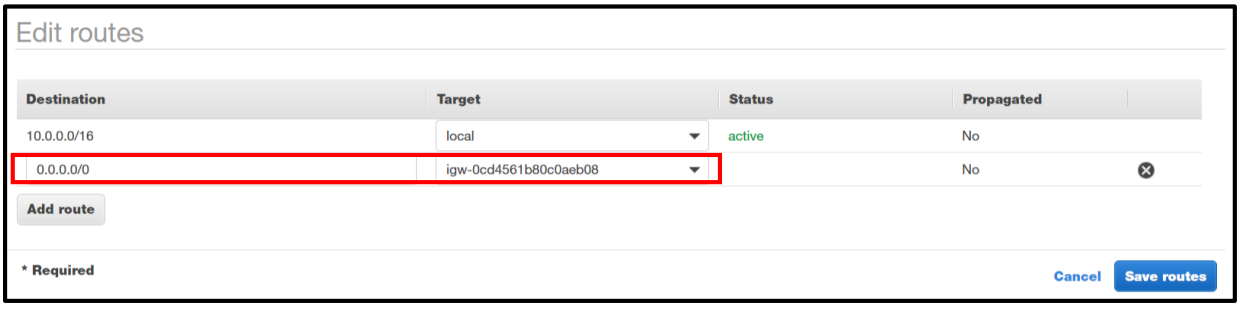


Configure as follows:

Destination: 0.0.0.0/0

Target: myIGW (from drop down)

Click on Save Routes.



Go back to **Subnets**. Select **myPrivateSN**. Go to Actions -> Edit route table associations.

Select the **PrivateRT** from drop down. Click on Save.

Step 6: In VPC console side panel, click Security Groups under **Security**.

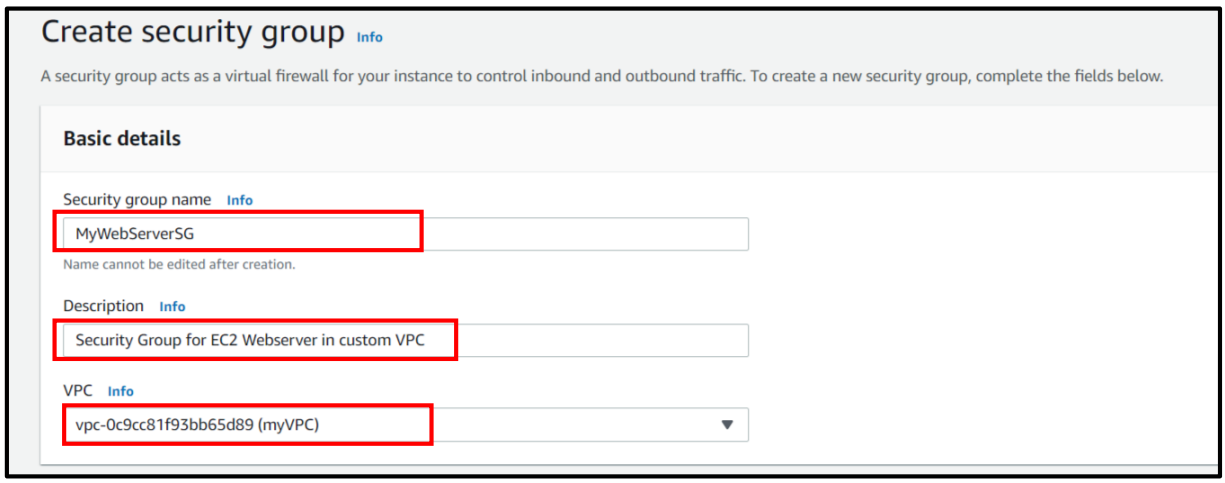
Click on Create security group.

Configure it as follows:

Security group name: MyWebServerSG

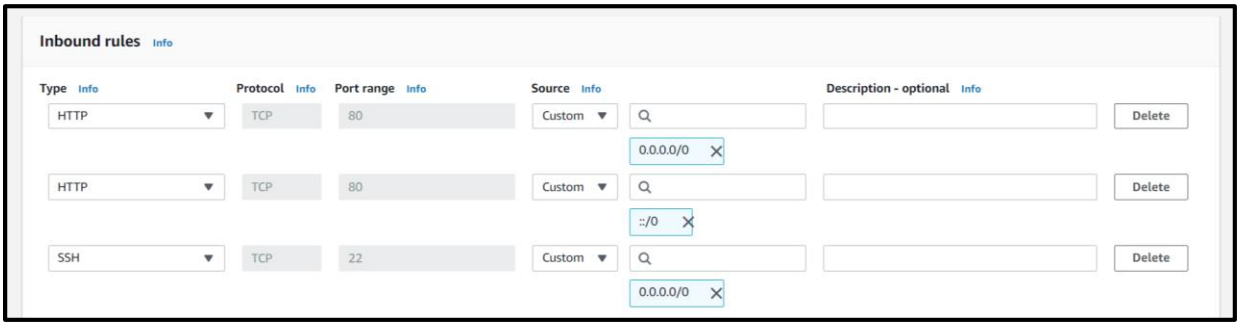
Description: Security Group for EC2 Webserver in custom VPC

Select myVPC from drop down.



Add three rules under **Inbound Rules**:

1. **Type**: HTTP
   Source: 0.0.0.0/0
2. **Type**: HTTP
   Source: ::/0
3. **Type**: SSH
   Source: 0.0.0.0/0



After adding the rules, scroll down and click on Create security group.

Now also create a database security group. Go to Security Groups under **Security**.

Click on Create security group. Configure it as follows:

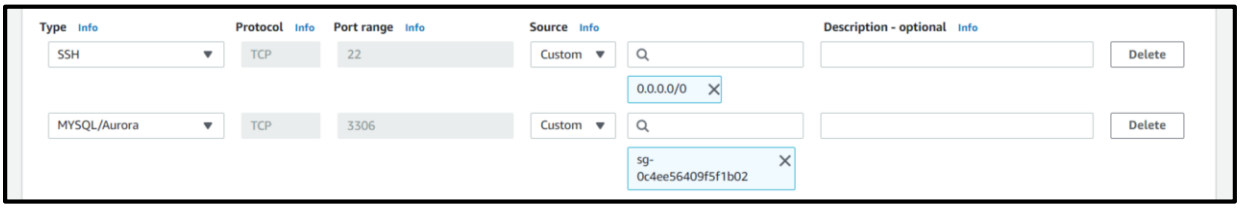
Security group name: MyDatabaseSG

Description: Security Group for RDS Database in custom VPC

Select myVPC from drop down.

Add three rules under Inbound Rules:

1. **Type**: SSH
   Source: 0.0.0.0/0
2. **Type**: MYSQL/Aurora
   Source: MyWebServerSG (The Security group created above for ec2 webserver instance creation)



After adding the rules, scroll down and click on Create security group.

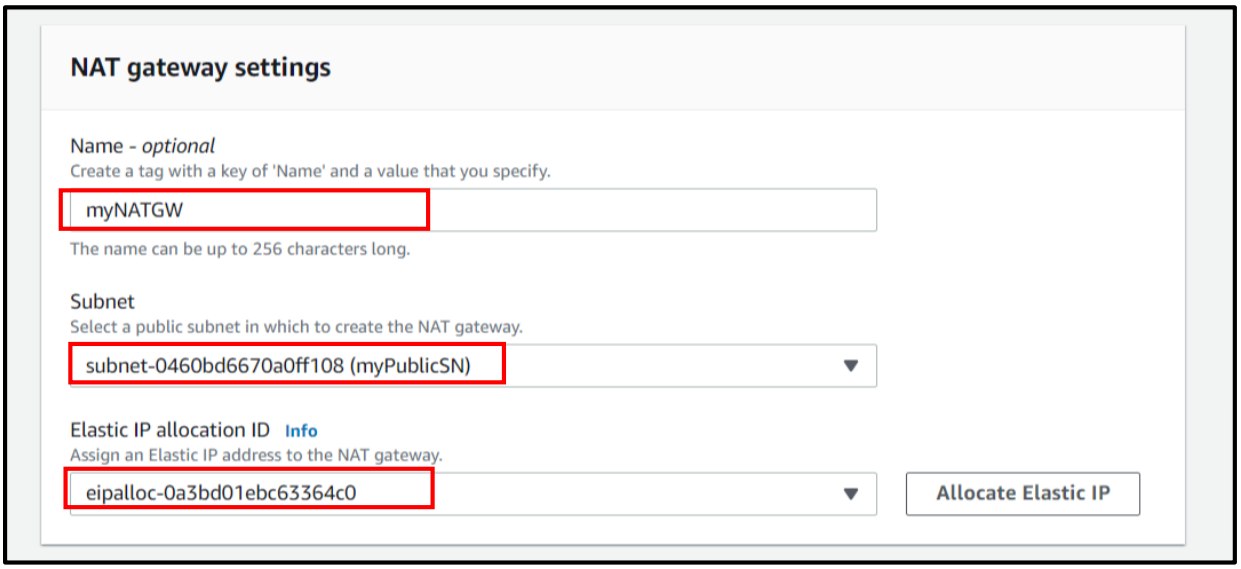
Step 7: In VPC console, go to NAT Gateways. Click on Create NAT gateway.

Configure as follows:

Name: myNATGW

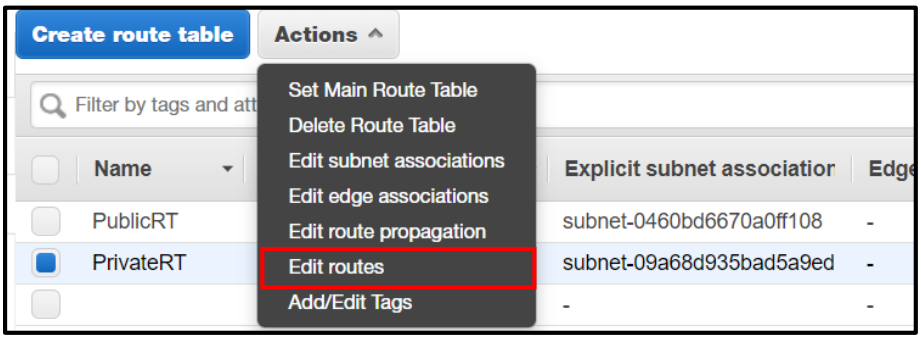
Subnet: myPublicSN

Elastic IP allocation ID: *Select the Allocate Elastic IP option*



Scroll down and click on Create NAT gateway.

Go back to **Route Tables**. Select PrivateRT. Go to Action -> Edit Routes.

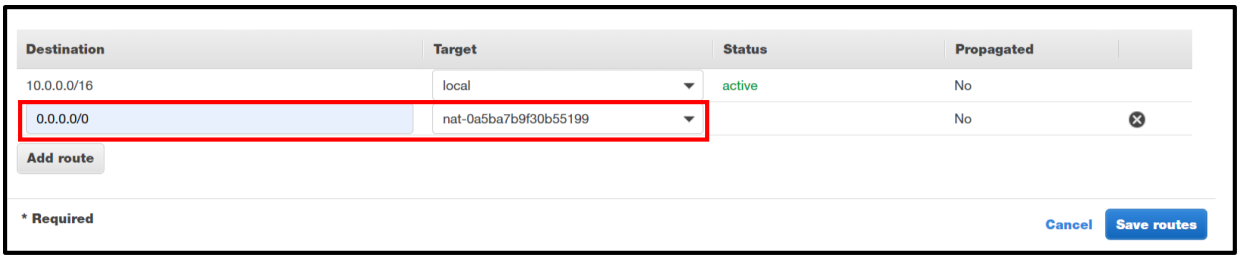


Add route with following configuration:

Destination: 0.0.0.0/0

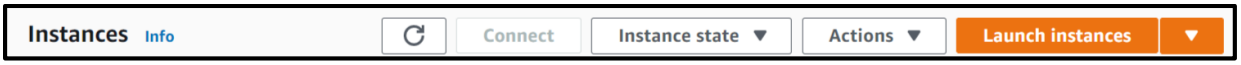
Target: myNATGW

Click on Save routes.

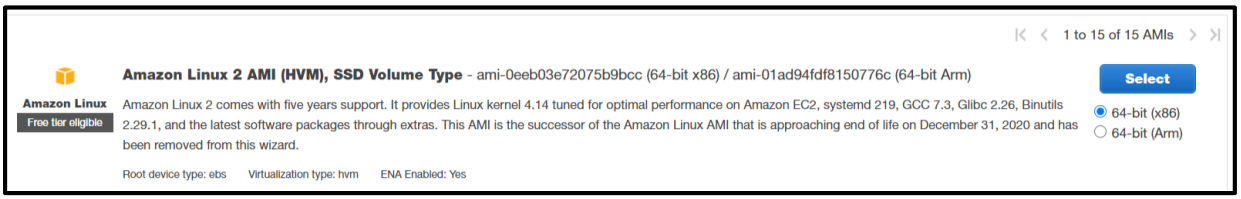


Step 8: In AWS console go to EC2 services. Select Instances.

Click on Launch instances.



Select Linux 2 AMI.



In next step keep the default **t2.micro**.

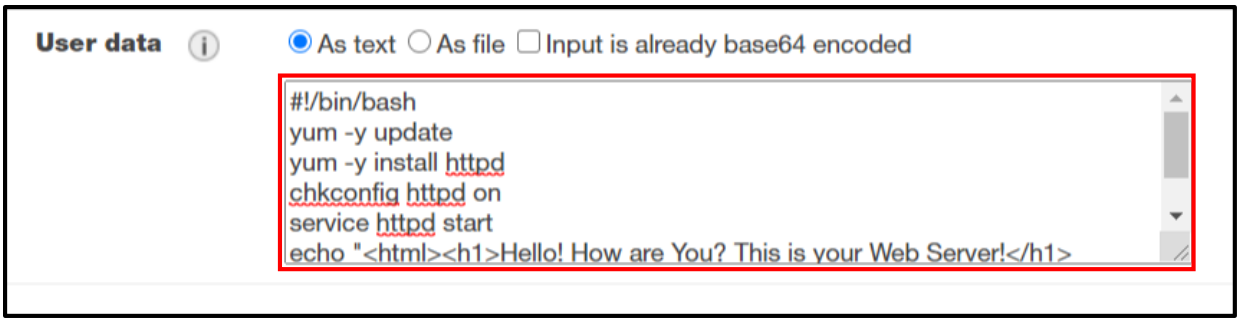
In next step, Configure Instance Details, select:

Network: myVPC

Subnet: myPublicSN

Scroll down to **Advanced Details**. Under **User data**, in the text box, provide the following script:

```
#!/bin/bash
yum -y update
yum -y install httpd
chkconfig httpd on
service httpd start
echo "<html><h1>Hello! How are You? This is your Web Server!</h1></html>" > /var/www/html/index.html
```



In next step keep default storage selection.

In next step add tags as follows:

Key: Name

Value: MyWebServerForVPC

In next step, click on **Select exiting security groups** radio button. Select the MyWebServerSG created previously.

Select the key pair, and launch the instance.

Copy the **Public IPv4 address**, run it in a different tab and test the web server instance.

If you no longer need this infrastructure, make sure to dissociate and release the elastic IP address, delete the EC2 instance, NAT gateway and the custom VPC.

Was this document helpful? YES / NO

Document Created by	Version
Parag Deshpande	05-MAR-2020