**AWS VPC Setup with Bastion Server and Database Access**

**VPC and Subnets:**

* Created VPC (MyVPC) with CIDR block 10.0.0.0/16.
* Created two subnets:
  + WebSN (10.0.1.0/24) - Public subnet for web server.
  + DbSN (10.0.2.0/24) - Private subnet for database server.

**Network Access:**

* Enabled public IP for WebSN.
* Created Internet Gateway (MyIGW) and attached it to the VPC.
* Created Route Table (InternetRT) and attached it to both WebSN and MyIGW.
* This routing allows public internet access for WebSN.

**Web Server:**

* Launched an EC2 instance in WebSN with User Data script to install web server (Apache) and display "HelloWorld" index page.
* Security Group (WebSG) allows SSH and HTTP access from anywhere.

Additional Details -- User Data

#!/bin/bash

sudo su

yum update -y

yum install httpd -y

cd /var/www/html

echo "HelloWorld" > index.html

service httpd start

chkconfig httpd on

**Database Server:**

* Launched an EC2 instance in DbSN with MySQL/Aurora service enabled.
* Security Group (DbSG) allows MySQL/Aurora connection only from WebSN subnet (10.0.1.0/24).

Change Type from SSH to MYSQL/Aurora

**Type** **Source**

MYSQL/Aurora Custom 10.0.1.0/24

(MySQL Port is open to entire subnet)

Now, web server can pull data from database server.

**Database Administrator (DBA)** wants to create some files. Wants to perform maintenance activity.

Can he connect?

As DB Server is not having public IP and it is not having internet connectivity, DBA cannot connect.

For this, we need to create Bastion server/ Jump server in public subnet

**Bastion Server:**

* Set up a jump server (BastionServer) in WebSN with SSH access only from your IP address.
* Bastion Security Group (BastionSG) allows SSH access from your system IP.

(SSH port -- should be open to myself)

**Type** **Source**

SSH My IP

Now, Only I can connect to Bastion server through SSH

From the Bastion server, I should able to jump into Dbserver.

That means, DbServer SSH port should be open to Bastion server.

* Update DbSG to allow SSH access from BastionServer's private IP (10.0.1.66).

Goto Dbserver security group - DbSG (new tab)

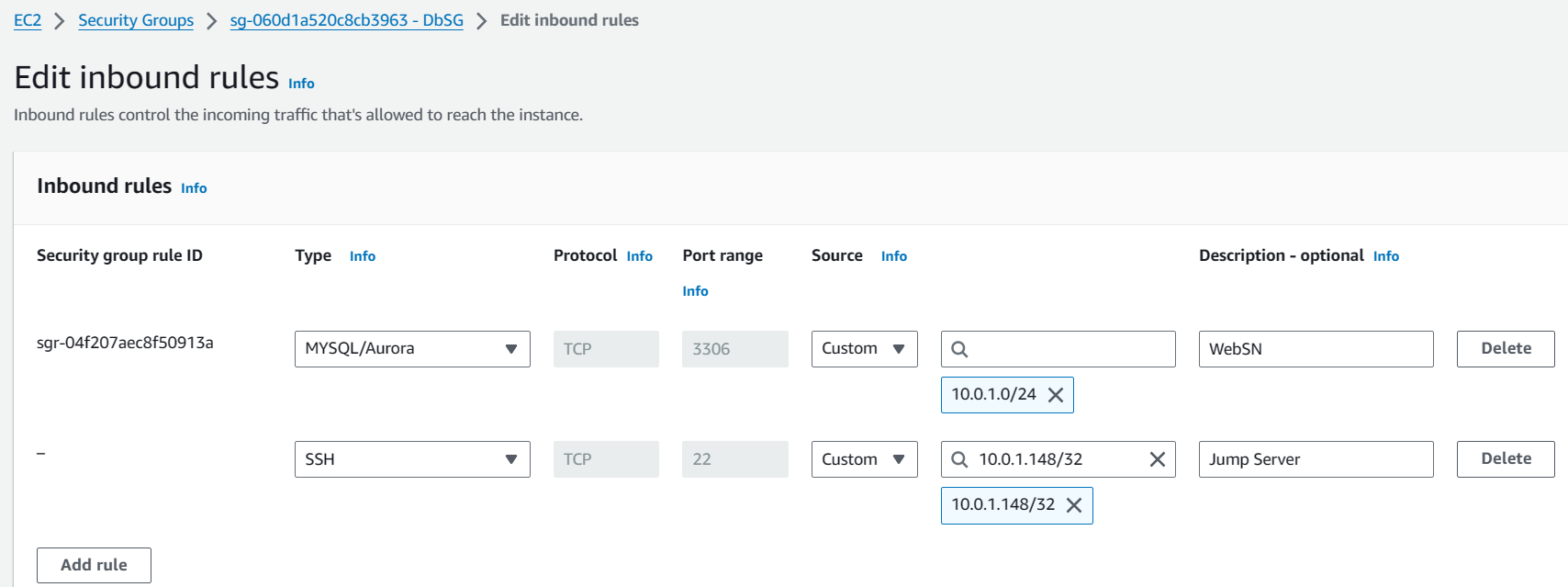
Select DbSG -- Inbound --Edit

Add Rule

**Type** **Source**

SSH Custom 10.0.1.66/32 (Private IP of bastion server)

Save.



**Connecting to Database Server:**

* From EC2 dashboard, connect to BastionServer.

$ sudo su

# yum update -y

* From BastionServer, use the copied SSH command with DbServer's PEM key to connect to the database server.

From bastion-- we need to jump to dbserver

Now, to connect to DBserver, we need to enter the details to DBserver in Bastion server

Select DbServer --- connect

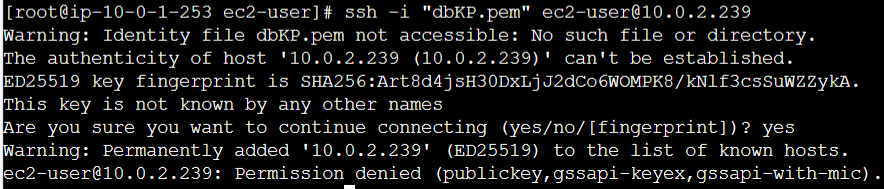
Copy the entire ssh command.

As we are connecting from linux to linux .pem file is enough.

Enter the ssh command.

eg:

# ssh -i "DbKP.pem" ec2-user@10.0.2.106



**Note:** To connect the .pem file need to be present in present working directory.

Now, we need to copy database server pem file in bastion server.

It is there in our windows machine.

Either we use WINSCP to transfer the file from windows to linux or just copy the pem file and paste it in bastion server with same name of pem file.

**Open WINSCP**

Connect to bastion server using Winscp

host name: user@ipaddress (bastion server)

advanced ---Authentication --- private key file - select the ppk file -- open -- ok - login

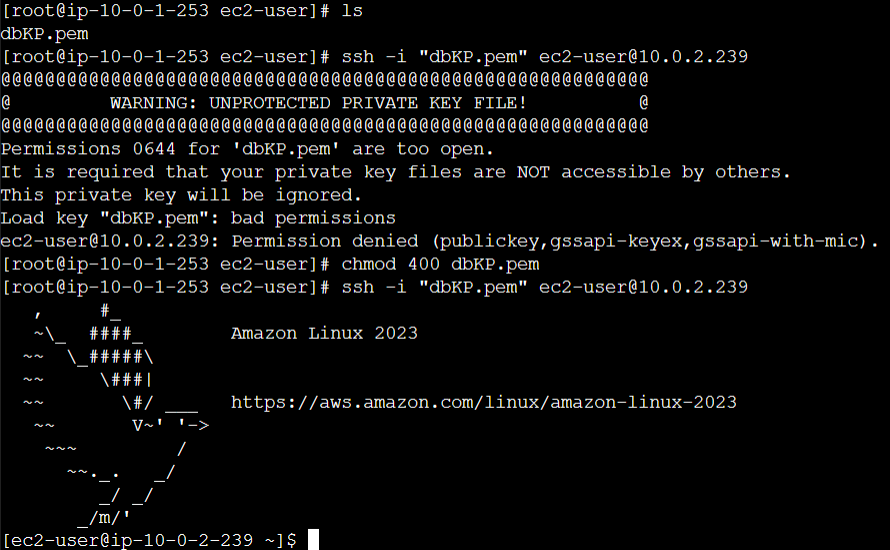
To get ppk from pem use puttygen

Now, drag and drop the database pem file to bastion server.

# ls (We should able to see the file)

Now connect to Dbserver by running the SSH command

# ssh -i "DbKP.pem" ec2-user@10.0.2.106



You are now connected to DBserver!!!

Now, In DB server, lets execute the following commands

$ sudo su

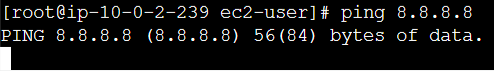
**Database Upgrade:**

* Tried to upgrade MySQL on DbServer, but failed due to lack of internet access.

Now I want to upgrade the latest version of MYSQL database

Command to upgrade MYSQL database

# ping 8.8.8.8



not successful.

We cannot install, as we are not having internet connection to private subnet.

To get internet connection, we create NAT server. (Network Address Translator)

The purpose of NAT is to provide internet to private subnet.

We need to create NAT in public subnet.

**NAT Gateway for Internet Access:**

* Create a NAT Gateway (myNAT) in WebSN subnet with an Elastic IP.
* Modify the default Route Table (NatRT) to route all traffic from DbSN subnet to the NAT Gateway for internet access.

In VPC Dashboard

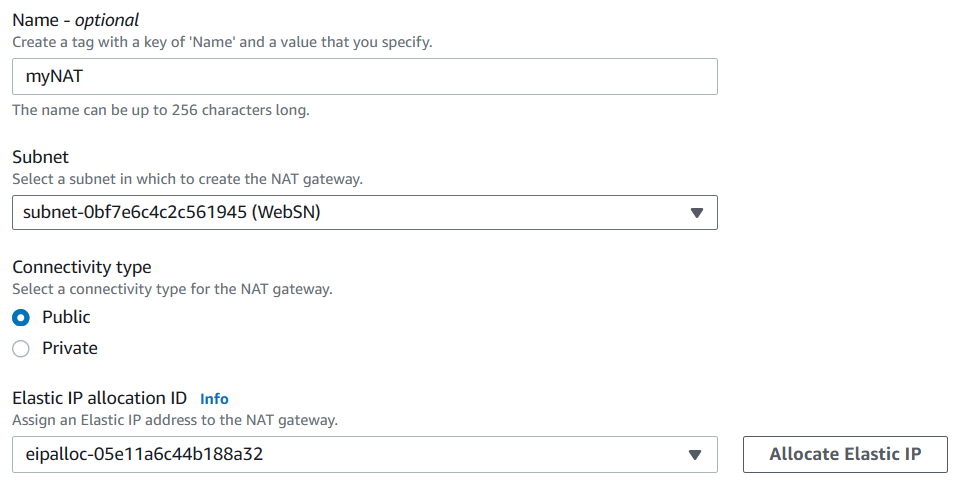
NAT Gateways > Create NAT Gateway (myNAT)

subnet: 10.0.1.0/24 (WebSN)

Create new EIP (allocate elastic Ip) > Create a NAT Gateway -- Close

NAT needs 2 min approx. to get created.

Name it as NAT



**What is Elastic IP?**

It is similar to Static IP

When we stop and start the EC2 Machines, public IP will change.

If you stop and start the machine, you want the same public IP, then we create Elastic IP

Elastic IP is nothing but static public IP

**Why do we need Elastic IP to NAT?**

If in case NAT is down, entire private subnet will not get internet.

Then we restart the NAT again, then it acquires new public IP

When NAT acquire new public IP, there could be connection issue.

So, we need Elastic IP to NAT

NAT is a closed box. It does not have any ports concept.

No one, can connect to NAT.

We cannot connect NAT to private Subnet.

So, we create RouteTable.

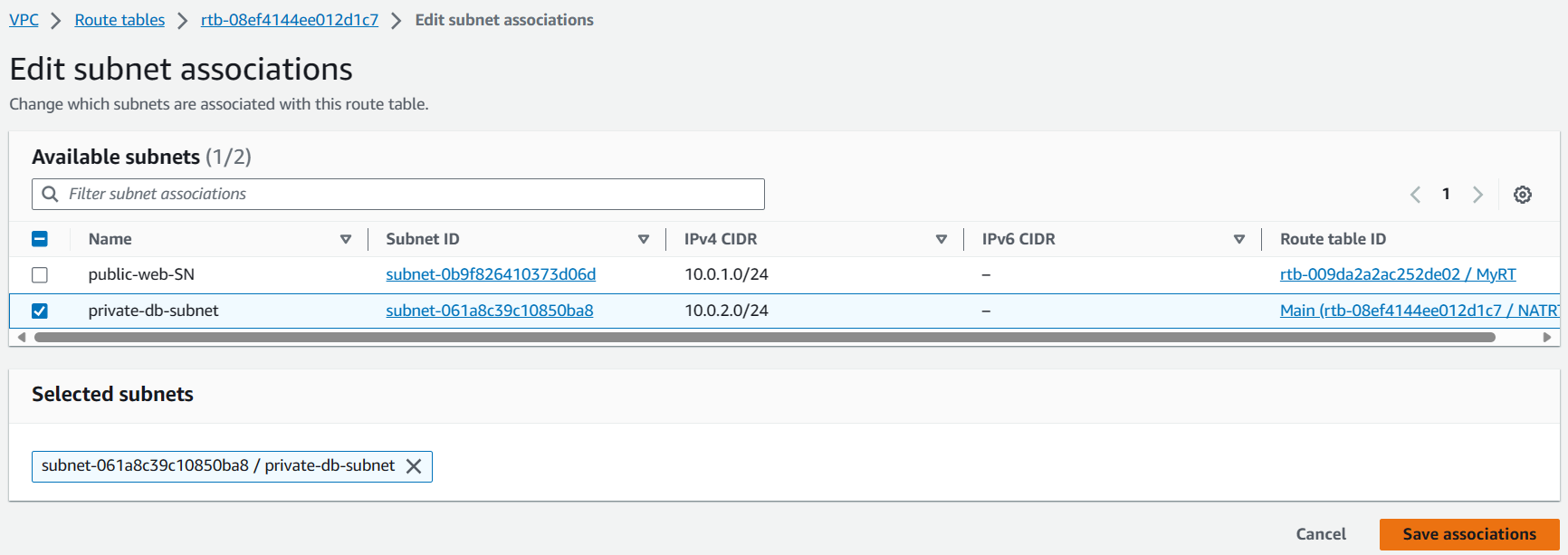
One end of RouteTable, I connect to NAT.

Another end of RouteTable, I Connect to private subnet.

Instead of creating new Route Table, we can use default RouteTable which was created, when we created VPC

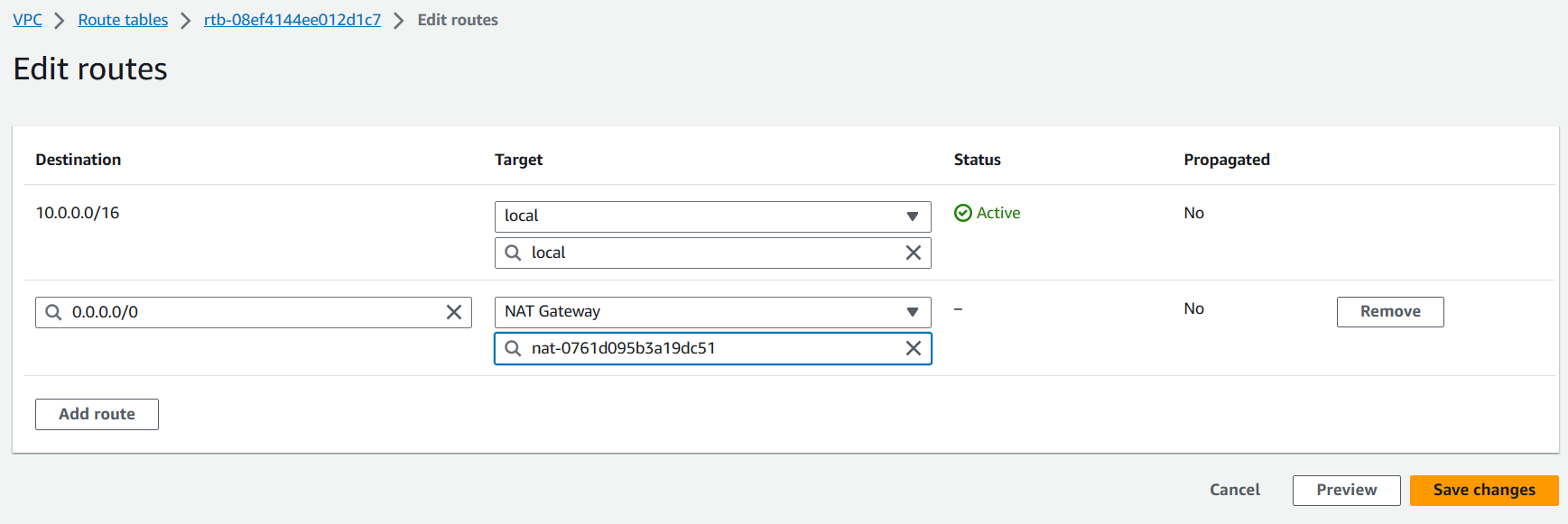
Let’s change the name of default RouteTable to NatRT

Select NatRT -- Subnet Associations -- Edit subnet Associations -- select private subnet-- save



Select NatRT -- Routes -- Edit Routes --Add Route -- Target: NAT Gateway (Select NAT)

Destination- 0.0.0.0/0 -- Save routes -- close

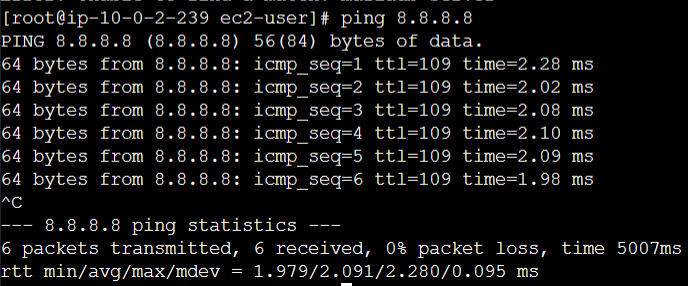


**Testing Internet Access:**

* Reconnected to DbServer and successfully ran yum install mysql -y to upgrade the database.

Test are we able to get internet to our DBServer.

Run the same command again

It Works!!

**Deletion Process:**

* Follow the order to safely delete resources:
  1. Delete NAT Gateway.
  2. Terminate EC2 Instances.
  3. Delete VPC.
  4. Release Elastic IPs. (Select Elastic IP from dashboard -- Actions -- Release Address)