Problem Statement:

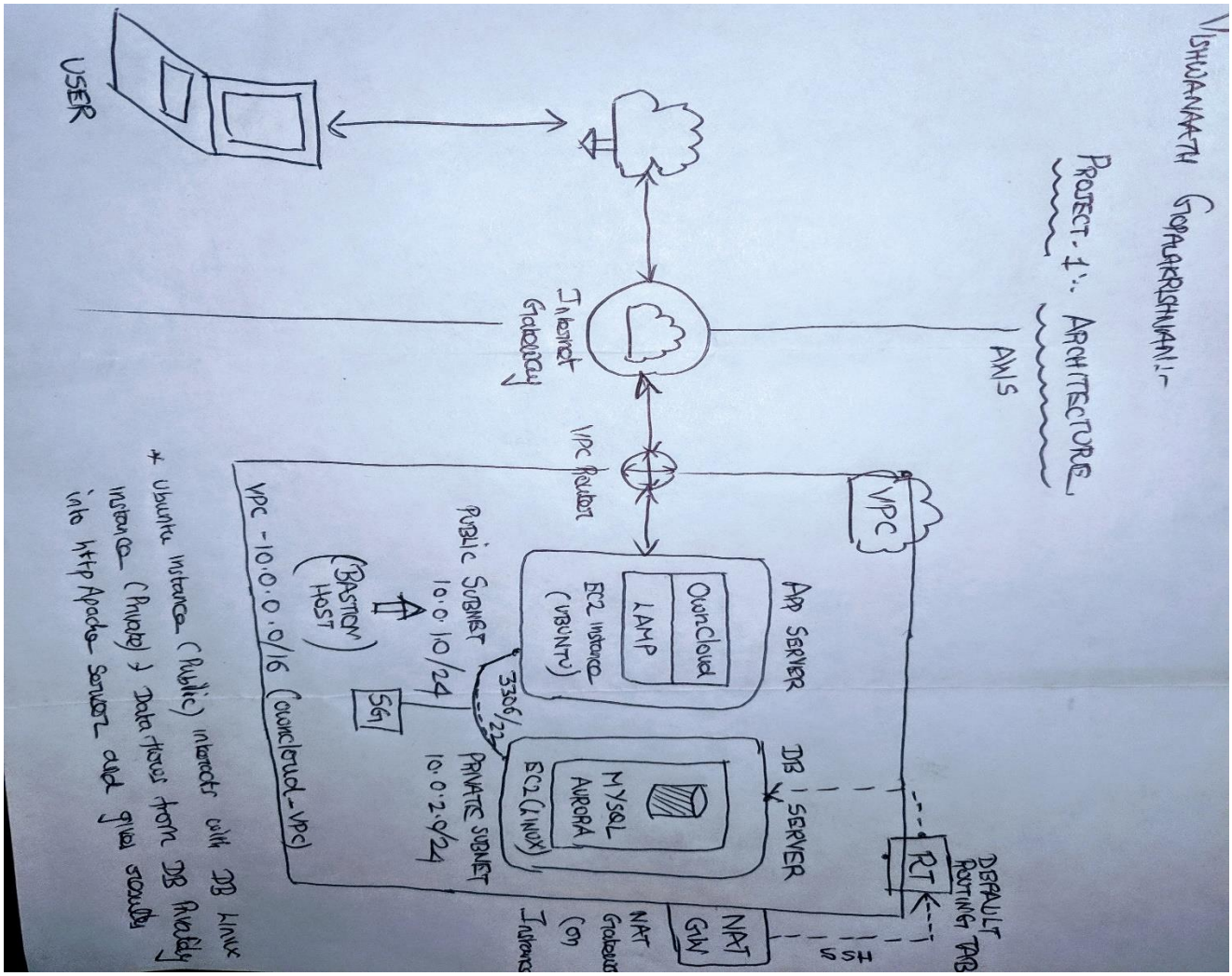
Creating a Fileshare and sync solution using ownCloud and AWS Project: Own cloud Solution - Vishwanaath Gopalakrishnan (18 Nov 2019)

Architecture and Screenshots listed in the document-

Learnings & Observations:

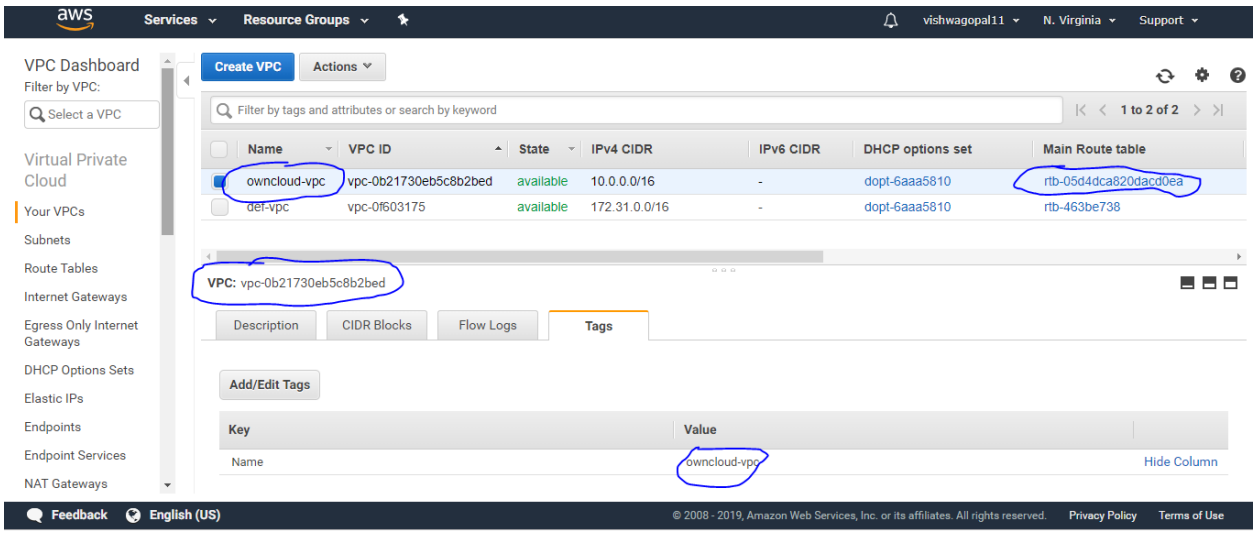
- Private Subnet created to host the DB server instance
- NAT gateway used to establish connection to the Private subnet only from the Public subnet by setting up route tables and NAT gateway and letting it act as Bastion host

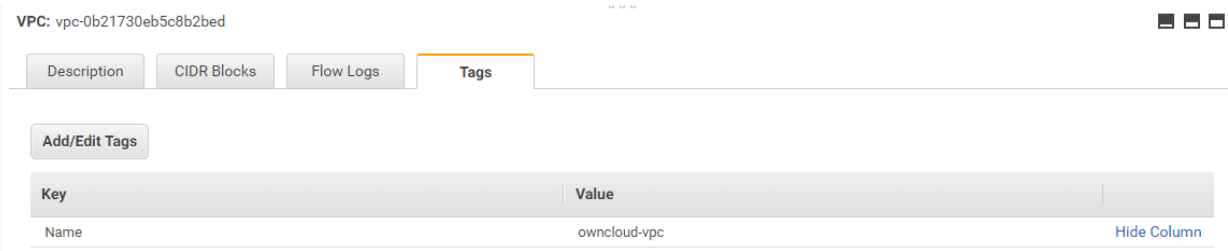
Architecture Diagram:



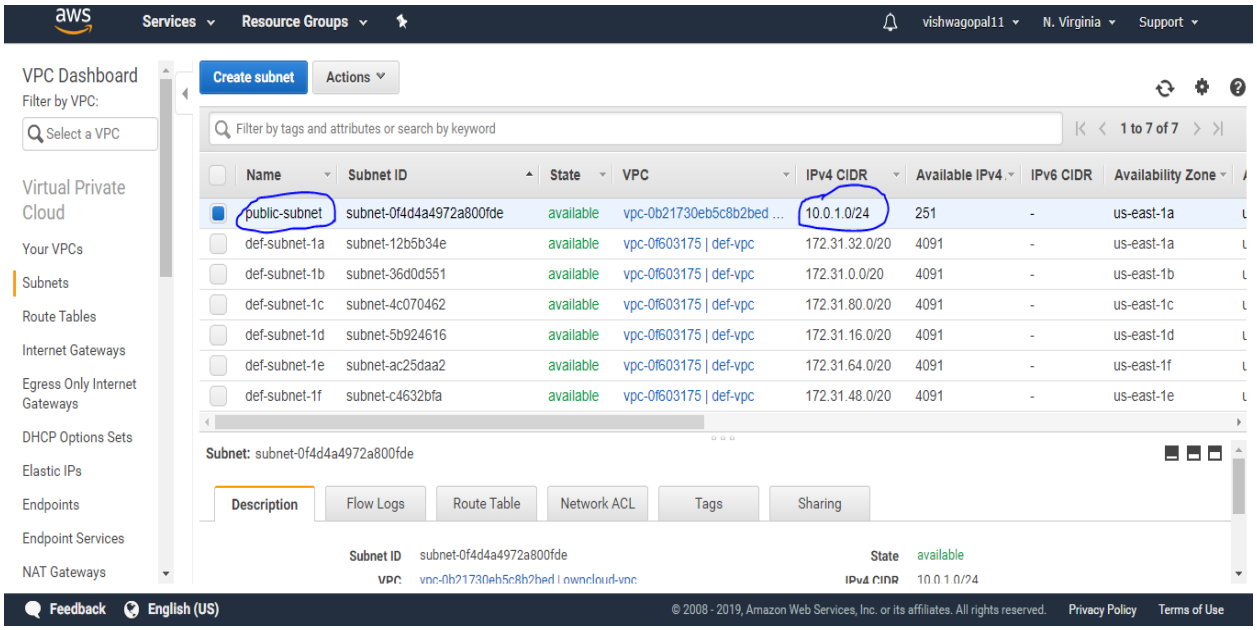
Implementation:

- Create a Custom VPC with the name owncloud-vpc with CIDR block 10.0.0.0/16

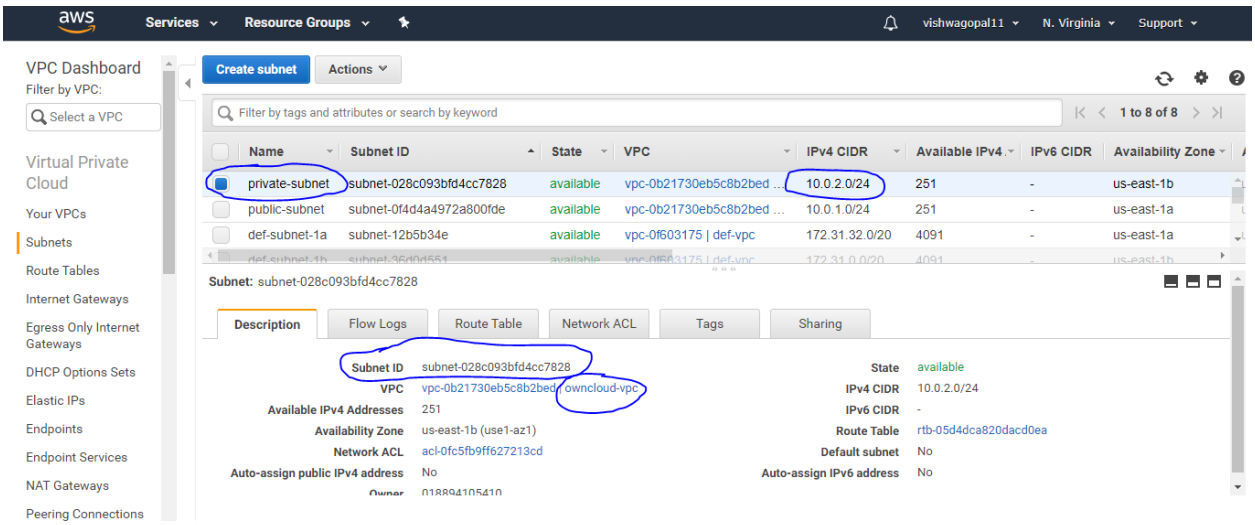




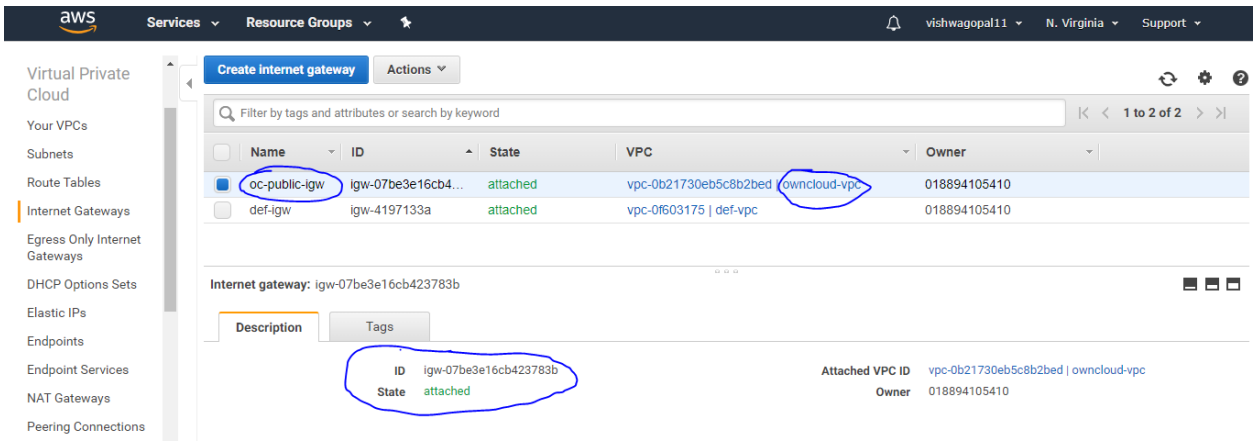
- Create a Public Subnet with a CIDR block 10.0.1.0/24 that will allow 251 IP addresses under the owncloud-vpc. This is where the owncloud LAMP App server will be launched (AZ -1a)



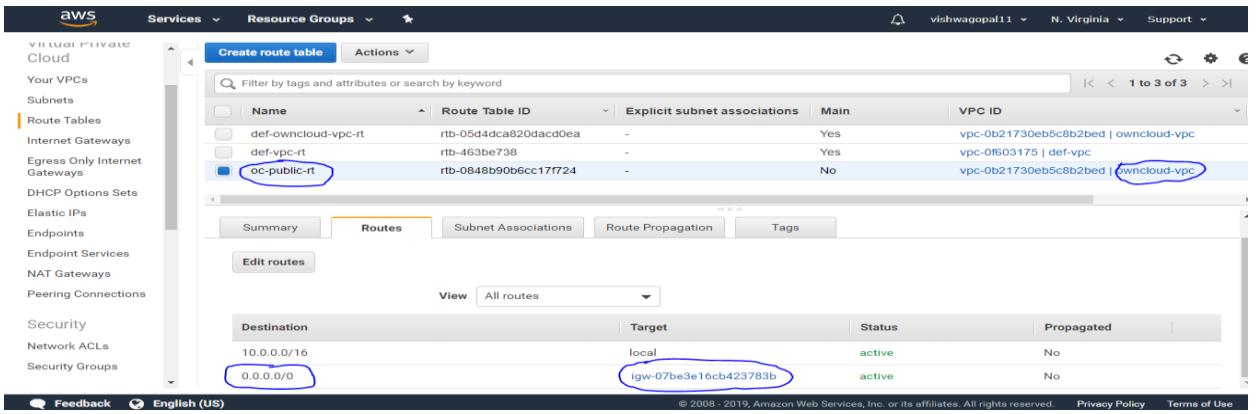
- Create a Private subnet with a CIDR block 10.0.2.0/24 that will allow 251 IP addresses also under the owncloud-vpc. This is where the MQ SQL Database server will be launched (AZ – 1b)



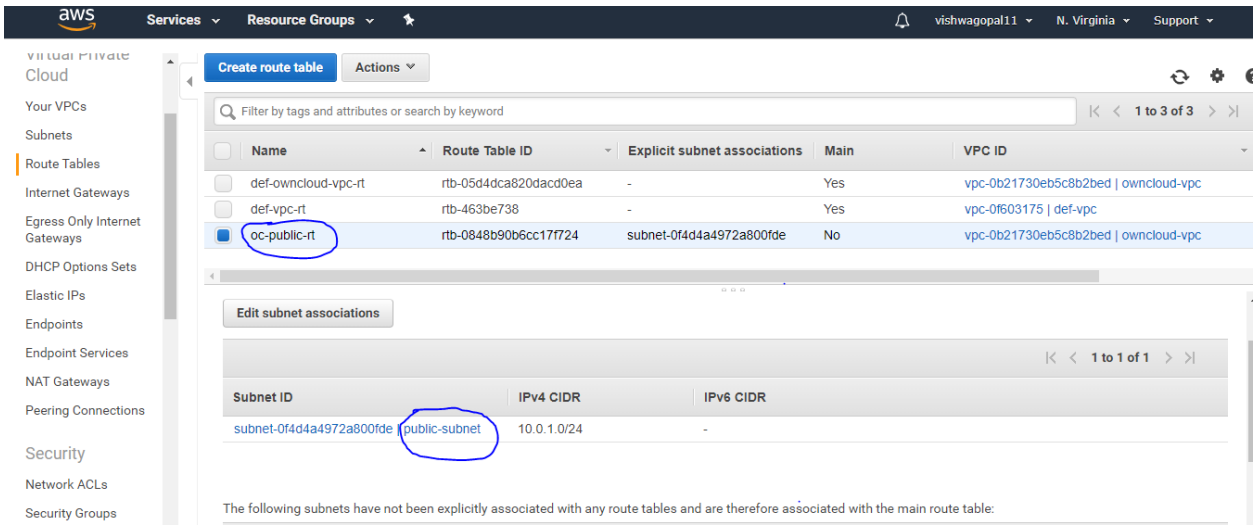
- Connect the Public Subnet to the Internet using an Internet gateway.
- Create a Custom Internet gateway as oc-public-igw and associate it with the owncloud-vpc



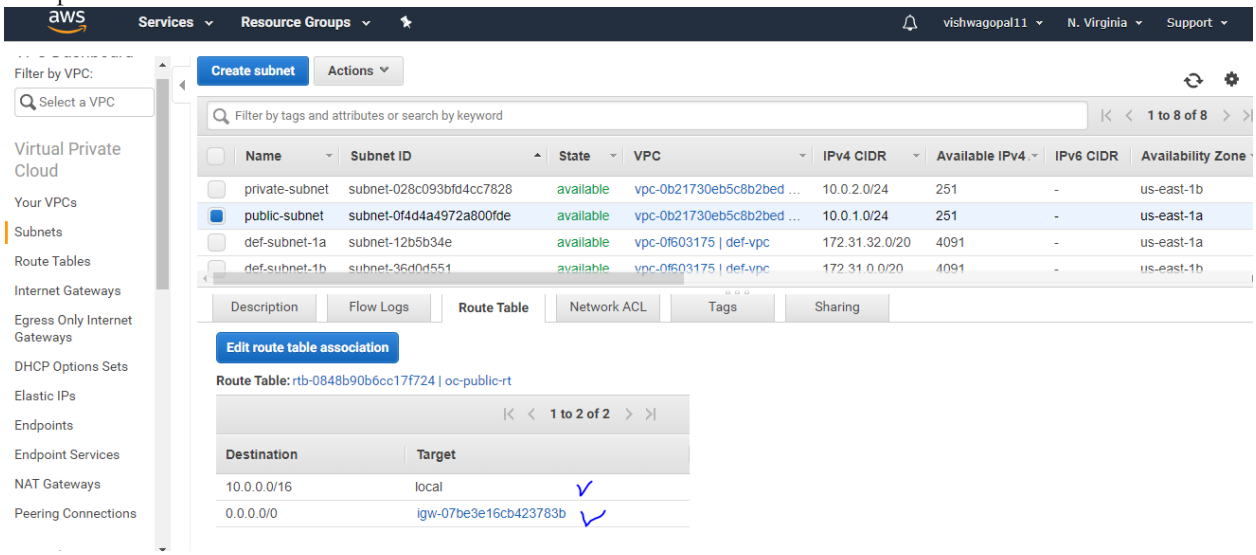
- Create a custom Route table oc-public-rt to connect the Public subnet in the VPC to the internet
- Add a global destination route CIDR block 0.0.0.0/0 and set the target to the Internet gateway oc-public-igw and save the route



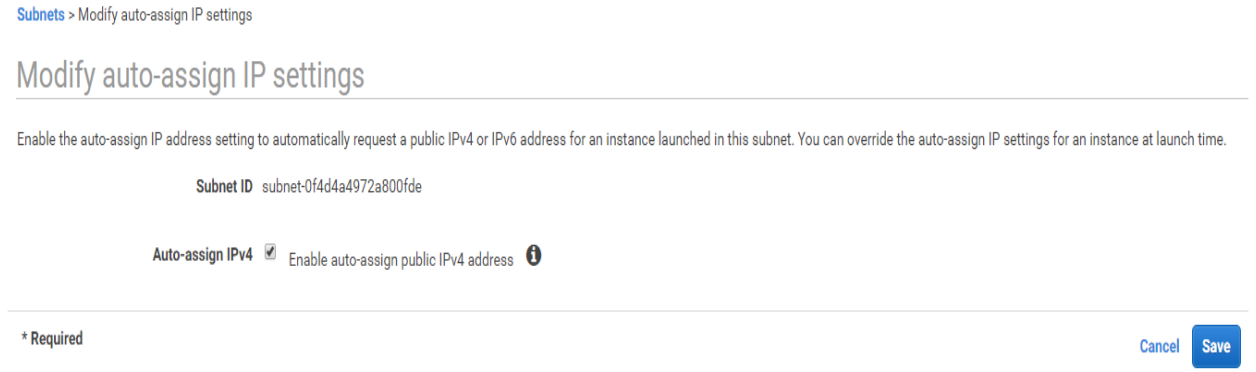
- Under Subnet associations, associate the route table created to the Public-subnet.

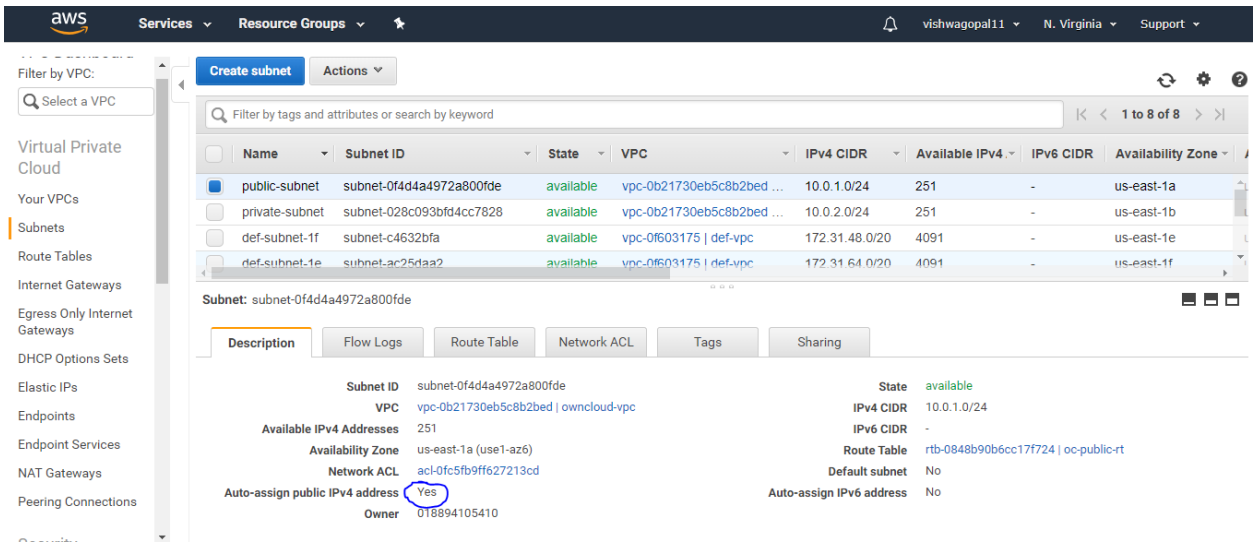


- The public subnet will now be associated with two route table entries as below

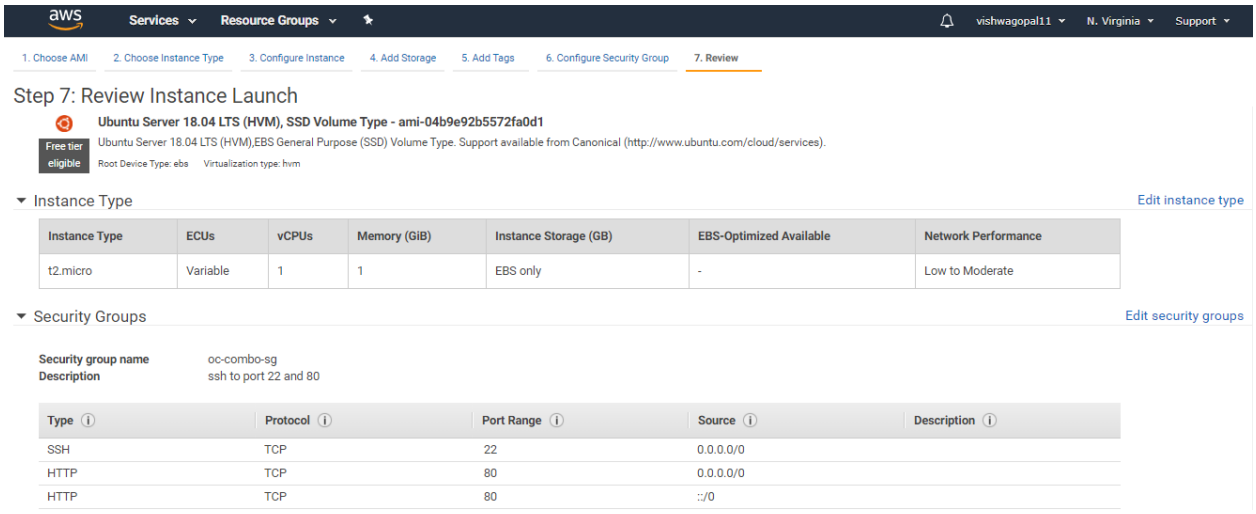


- Enable the auto assign IP settings for the Public subnet to allow the EC2 instance to get an IP address when launched

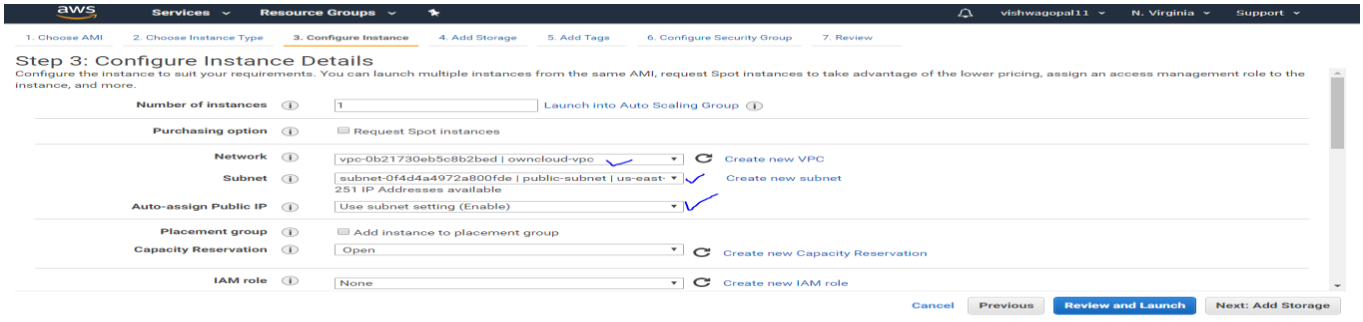




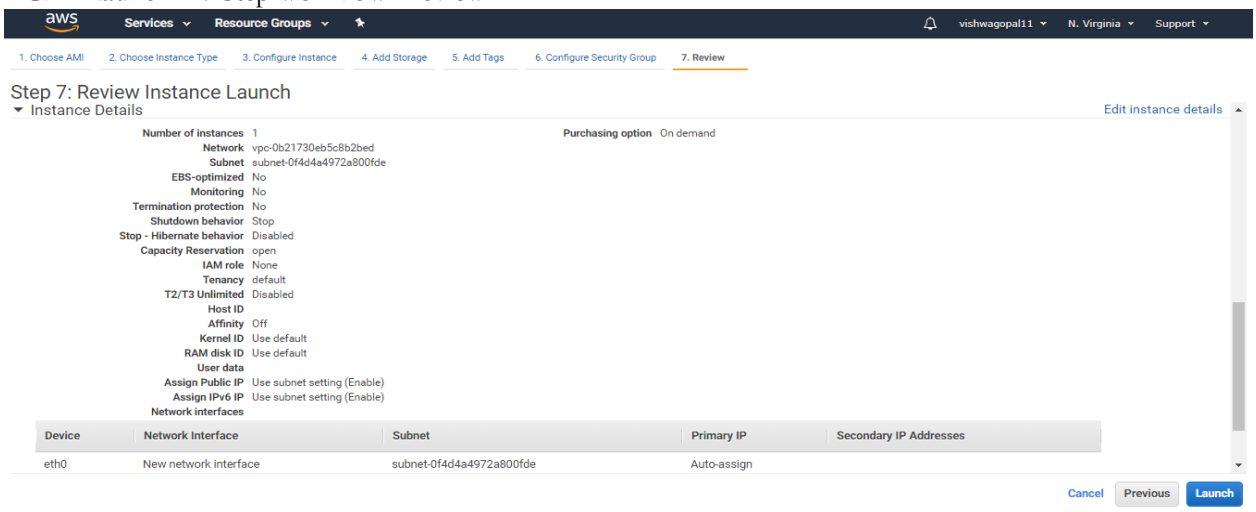
- Create EC2 instance in the Public subnet and proceed with the steps to setup the owncloud and LAMP applications in the EC2 instance as per the 7-step workflow

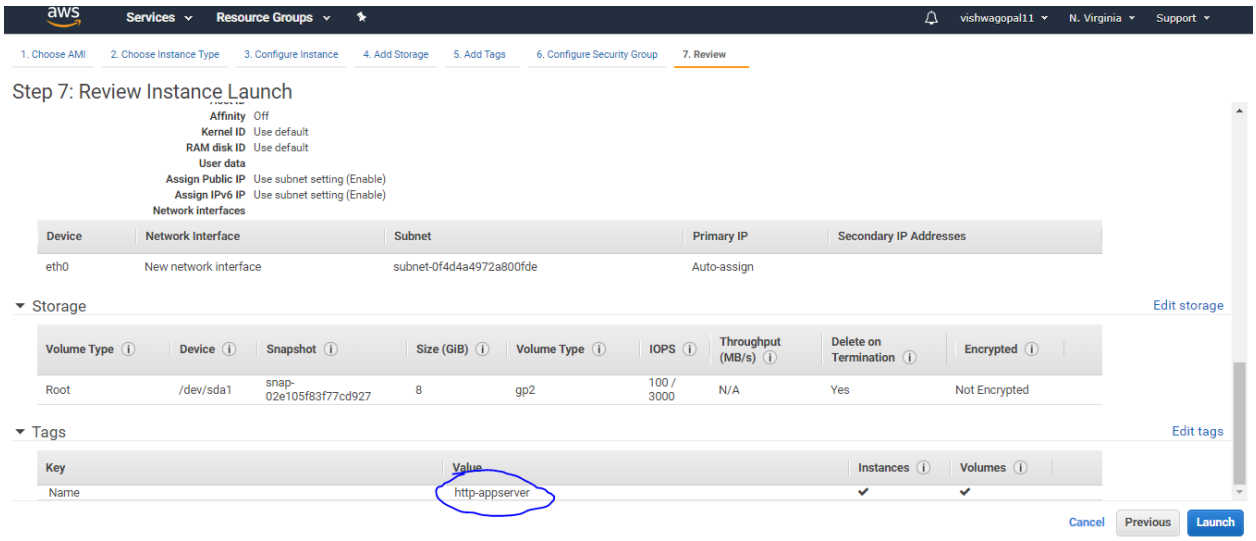


- Setup security group to port 22 and port 28 as oc-combo-sg
- Ensure that the EC2 instance is associated with the correct VPC and subnet as below

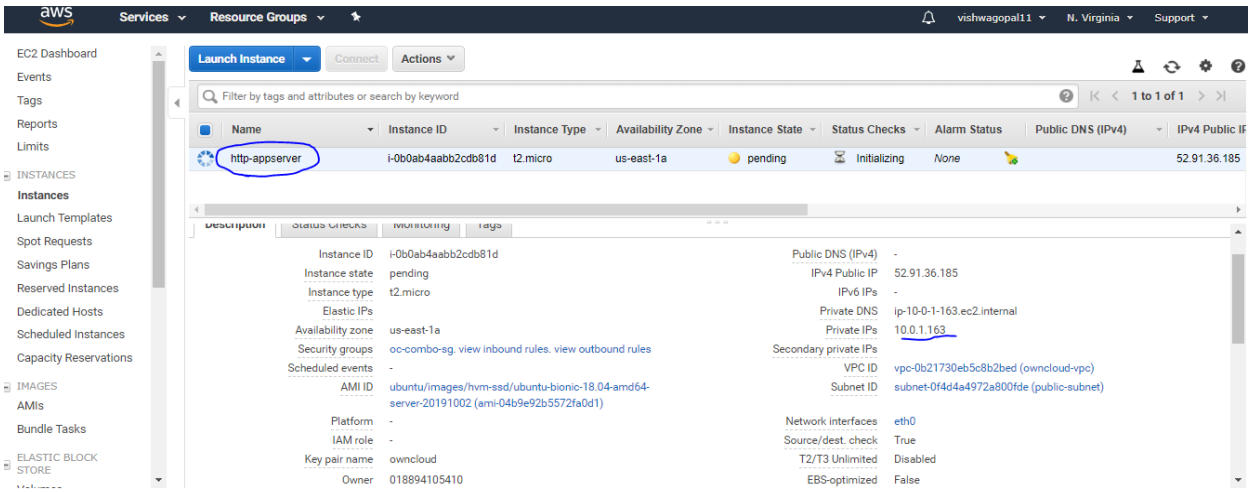


- EC2 – Launch – 7 Step workflow - review

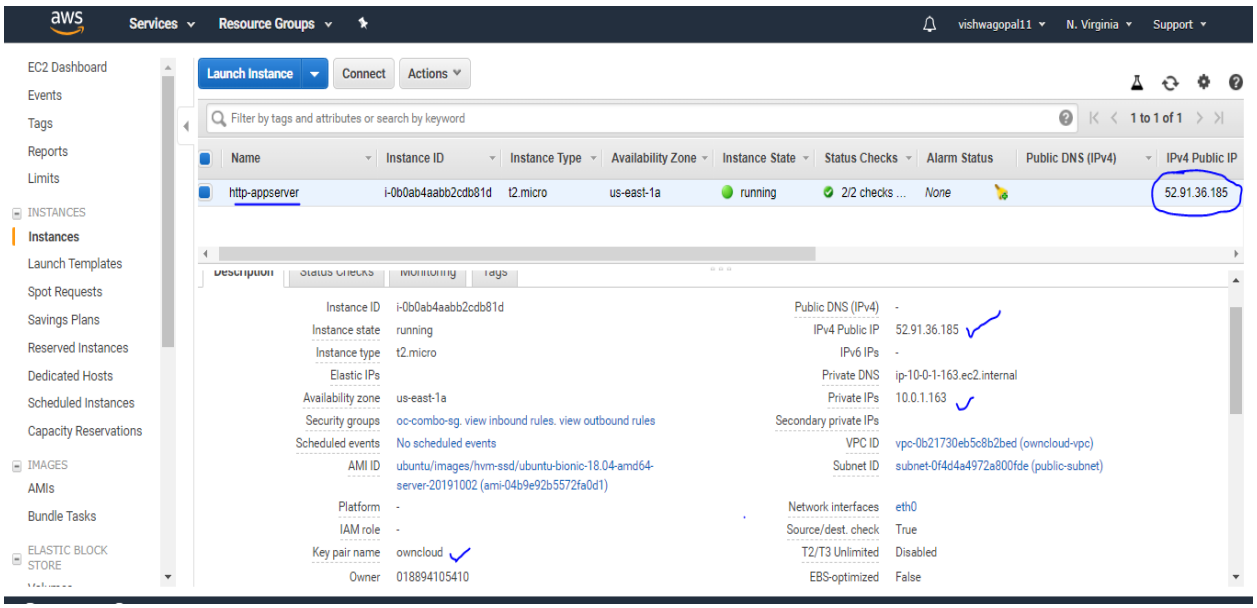


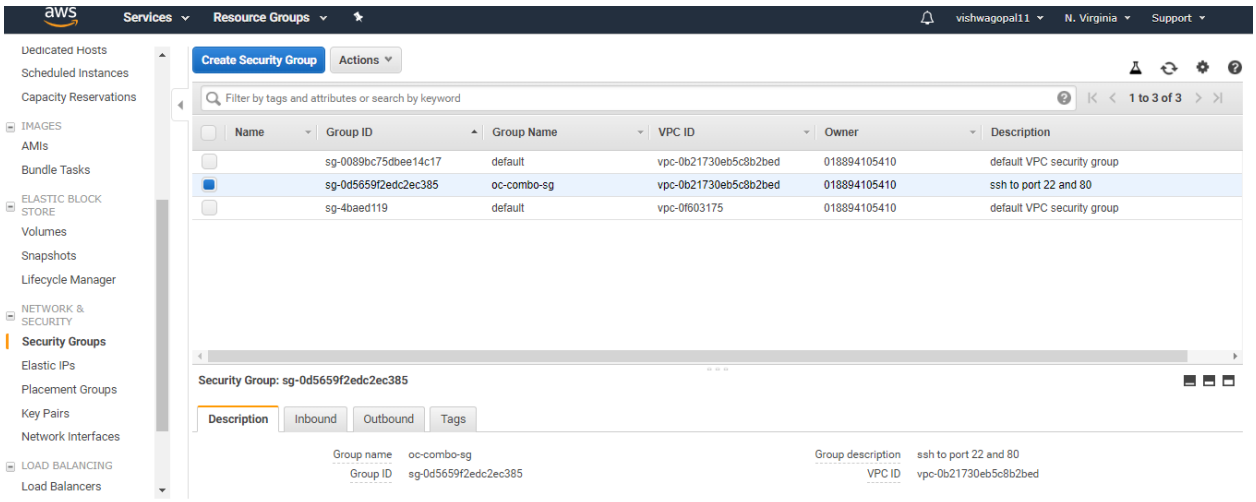


- Launch the EC2 instance after creating a keypair as owncloud.pem and download the keypair file to the system and ensure that the EC2 instance is created in the public subnet's IP range



- SSH into the EC2 instance using the owncloud.pem file using the IPV4 public IP address 52.91.36.185 after ensuring the security group is setup for both ports 22 and 80





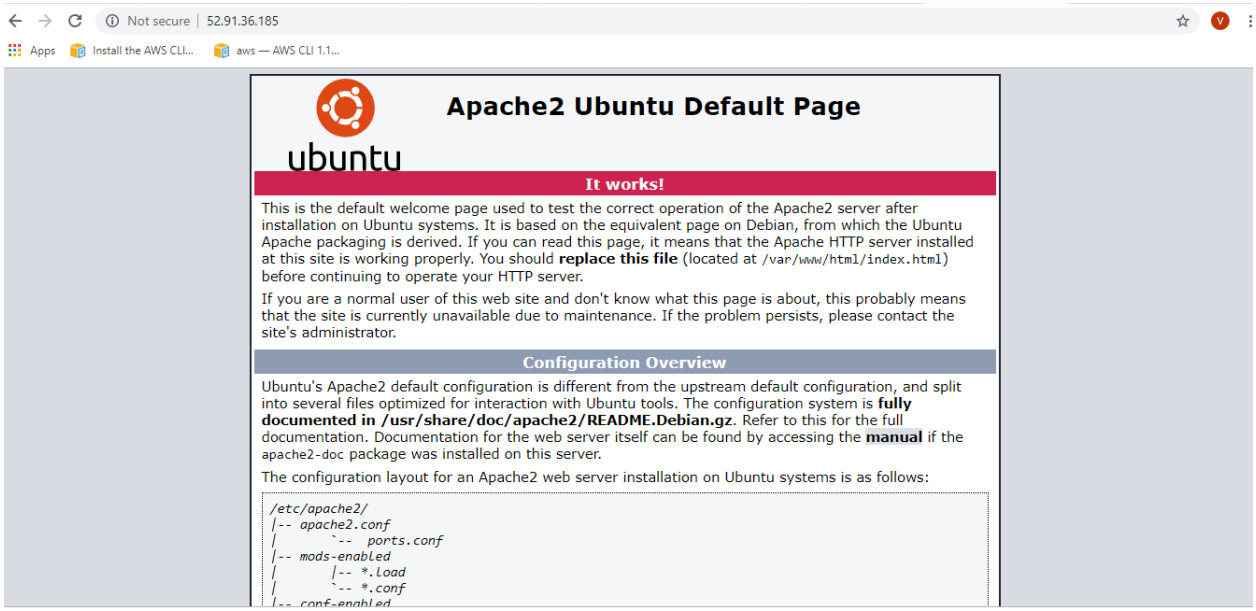
- Once the security groups are created with appropriate ports, the console can be used to get into the instance

Logging into the console

- SSHed into the instance and executed the below steps
- Install apache web server using following commands
  1. sudo apt-get update
  2. sudo apt-get install apache2

```
ubuntu@DESKTOP-49DC5BB:/mnt/c/users/vishw/downloads$ sudo su
[sudo] password for ubuntu:
root@DESKTOP-49DC5BB:/mnt/c/Users/Vishw/Downloads# ssh -i owncloud.pem ubuntu@52.91.36.185
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1051-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Nov 18 03:35:04 UTC 2019

  System load:  0.0               Processes:           88
  Usage of /:   15.8% of 7.69GB   Users logged in:     0
  Memory usage: 16%               IP address for eth0: 10.0.1.163
  Swap usage:   0%


51 packages can be updated.
24 updates are security updates.


Last login: Mon Nov 18 03:15:09 2019 from 73.61.110.39
ubuntu@ip-10-0-1-163:~$ sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease
Fetched 88.7 kB in 0s (240 kB/s)
Reading package lists... Done
ubuntu@ip-10-0-1-163:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2 is already the newest version (2.4.29-1ubuntu4.11).
0 upgraded, 0 newly installed, 0 to remove and 47 not upgraded.
ubuntu@ip-10-0-1-163:~$
```

- Validate installation by accessing public ip of EC2 instance in browser



- Use the following commands to install php - sudo apt install php libapache2-mod-php php-mysql
- Make index.php as the default first load page 1. Edit /etc/apache2/mods-enabled/dir.conf file and make index.php as first access page DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm 2. Restart the web server - sudo systemctl restart apache2

```
ubuntu@ip-10-0-1-163:~$ sudo apt install php libapache2-mod-php
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php7.2 libsodium23 php-common php7.2 php7.2-cli php7.2-common php7.2-json php7.2-opcache php7.2-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php libapache2-mod-php7.2 libsodium23 php php-common php7.2 php7.2-cli php7.2-common php7.2-json php7.2-opcache php7.2-readline
0 upgraded, 11 newly installed, 0 to remove and 47 not upgraded.
Need to get 4013 kB of archives.
After this operation, 17.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Abort.
ubuntu@ip-10-0-1-163:~$
```

- Edit /etc/apache2/mods-enabled/dir.conf file and make index.php as first access page DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm

```
ubuntu@ip-10-0-1-163:/$ cat /etc/apache2/mods-enabled/dir.conf file
<IfModule mod_dir.c>
        DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
cat: file: No such file or directory
ubuntu@ip-10-0-1-163:/$ vi /etc/apache2/mods-enabled/dir.conf file
2 files to edit
ubuntu@ip-10-0-1-163:/$ ls -ltr /etc/apache2/mods-enabled/dir.conf file
ls: cannot access 'file': No such file or directory
lrwxrwxrwx 1 root root 26 Nov 18 03:20 /etc/apache2/mods-enabled/dir.conf -> ../mods-available/dir.conf
ubuntu@ip-10-0-1-163:/$ sudo su
root@ip-10-0-1-163:/# cat /etc/apache2/mods-enabled/dir.conf file
<IfModule mod_dir.c>
        DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
cat: file: No such file or directory
root@ip-10-0-1-163:/# vi cat /etc/apache2/mods-enabled/dir.conf file
3 files to edit
root@ip-10-0-1-163:/# vi /etc/apache2/mods-enabled/dir.conf file
2 files to edit
root@ip-10-0-1-163:/# cat /etc/apache2/mods-enabled/dir.conf file
<IfModule mod_dir.c>
        DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
cat: file: No such file or directory
root@ip-10-0-1-163:/#
```

- Restart using sudo systemctl restart apache2

```
Select root@ip-10-0-1-163: /
<IfModule mod_dir.c>
        DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
cat: file: No such file or directory
root@ip-10-0-1-163:/# sudo systemctl restart apache2
root@ip-10-0-1-163:/# pwd
/
root@ip-10-0-1-163:/#
```

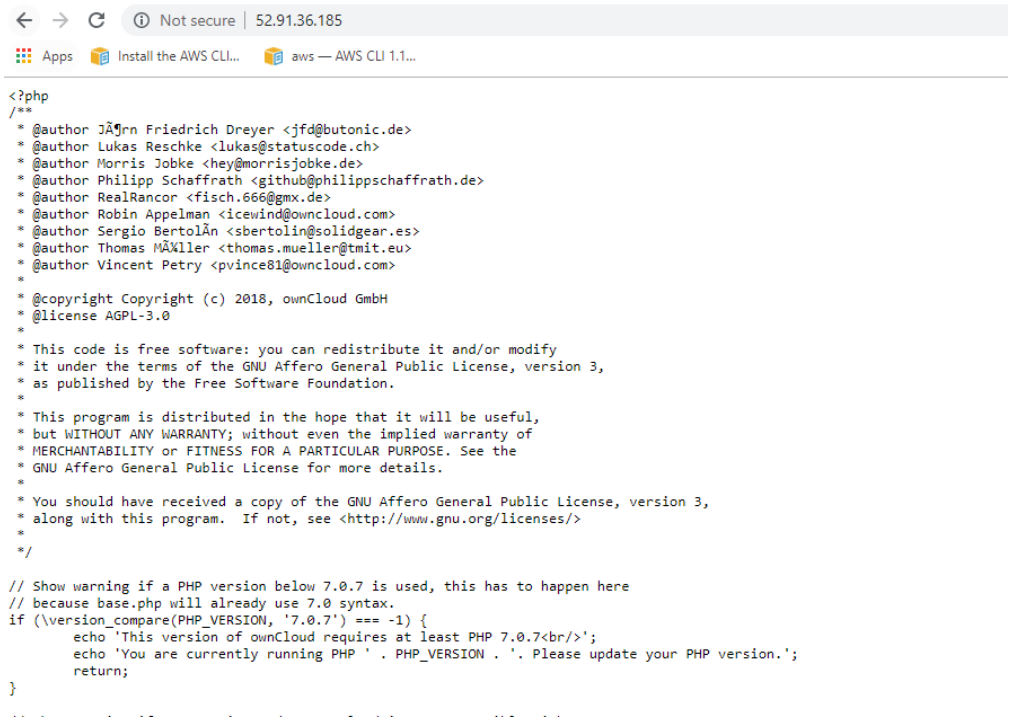
- Install owncloud on ubuntu 18.04 using the commands provided

1. curl https://download.owncloud.org/download/repositories/10.2/Ubuntu_18.04/Release.key | sudo apt-key add –
2. echo 'deb http://download.owncloud.org/download/repositories/10.2/Ubuntu_18.04/ /' | sudo tee /etc/apt/sources.list.d/owncloud.list
3. sudo apt update 4. sudo apt install php-bz2 php-curl php-gd php-imagick php-intl php-mbstring php-xml php-zip owncloud-files

root@ip-10-0-1-163: /

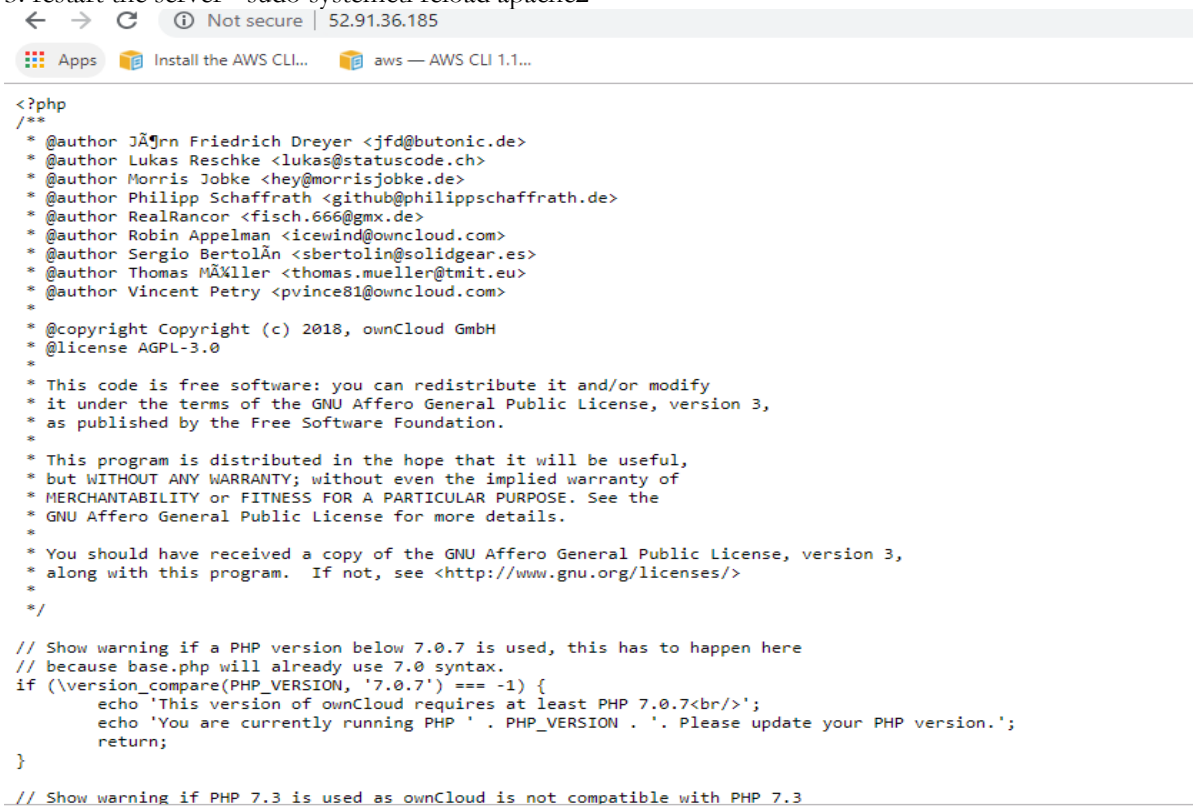
```
Setting up php7.2-xml (7.2.24-0ubuntu0.18.04.1) ...

Creating config file /etc/php/7.2/mods-available/dom.ini with new version

Creating config file /etc/php/7.2/mods-available/simplexml.ini with new version

Creating config file /etc/php/7.2/mods-available/wddx.ini with new version

Creating config file /etc/php/7.2/mods-available/xml.ini with new version

Creating config file /etc/php/7.2/mods-available/xmlreader.ini with new version

Creating config file /etc/php/7.2/mods-available/xmlwriter.ini with new version

Creating config file /etc/php/7.2/mods-available/xsl.ini with new version
Setting up php7.2-zip (7.2.24-0ubuntu0.18.04.1) ...

Creating config file /etc/php/7.2/mods-available/zip.ini with new version
Setting up php-bz2 (1:7.2+60ubuntu1) ...
Setting up libmagickcore-6.q16-3:amd64 (8:6.9.7.4+dfsg-16ubuntu6.8) ...
Setting up libgd3:amd64 (2.2.5-4ubuntu0.3) ...
Setting up php-zip (1:7.2+60ubuntu1) ...
Setting up php-xml (1:7.2+60ubuntu1) ...
Setting up php7.2-cli (7.2.24-0ubuntu0.18.04.1) ...
update-alternatives: using /usr/bin/php7.2 to provide /usr/bin/php (php) in auto mode
update-alternatives: using /usr/bin/phar7.2 to provide /usr/bin/phar (phar) in auto mode
update-alternatives: using /usr/bin/phar.phar7.2 to provide /usr/bin/phar.phar (phar.phar) in auto mode

Creating config file /etc/php/7.2/cli/php.ini with new version
Setting up php7.2-phpdbg (7.2.24-0ubuntu0.18.04.1) ...
update-alternatives: using /usr/bin/phpdbg7.2 to provide /usr/bin/phpdbg (phpdbg) in auto mode

Creating config file /etc/php/7.2/phpdbg/php.ini with new version
Setting up libmagickwand-6.q16-3:amd64 (8:6.9.7.4+dfsg-16ubuntu6.8) ...
Setting up libavahi-client3:amd64 (0.7-3.1ubuntu1.2) ...
Setting up libcups2:amd64 (2.2.7-1ubuntu2.7) ...
Setting up php7.2-gd (7.2.24-0ubuntu0.18.04.1) ...

Creating config file /etc/php/7.2/mods-available/gd.ini with new version
Setting up php-imagick (3.4.3~rc2-2ubuntu4) ...
Setting up libcupsfilters1:amd64 (1.20.2-0ubuntu3.1) ...
Setting up libcupsimage2:amd64 (2.2.7-1ubuntu2.7) ...
Setting up php-gd (1:7.2+60ubuntu1) ...
Setting up libgs9:amd64 (9.26~dfsg+0-0ubuntu0.18.04.12) ...
Setting up ghostscript (9.26~dfsg+0-0ubuntu0.18.04.12) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
root@ip-10-0-1-163:/#
```

- Accessing the IPV4 address 52.91.36.185 throgh address bar gives the required results zs below



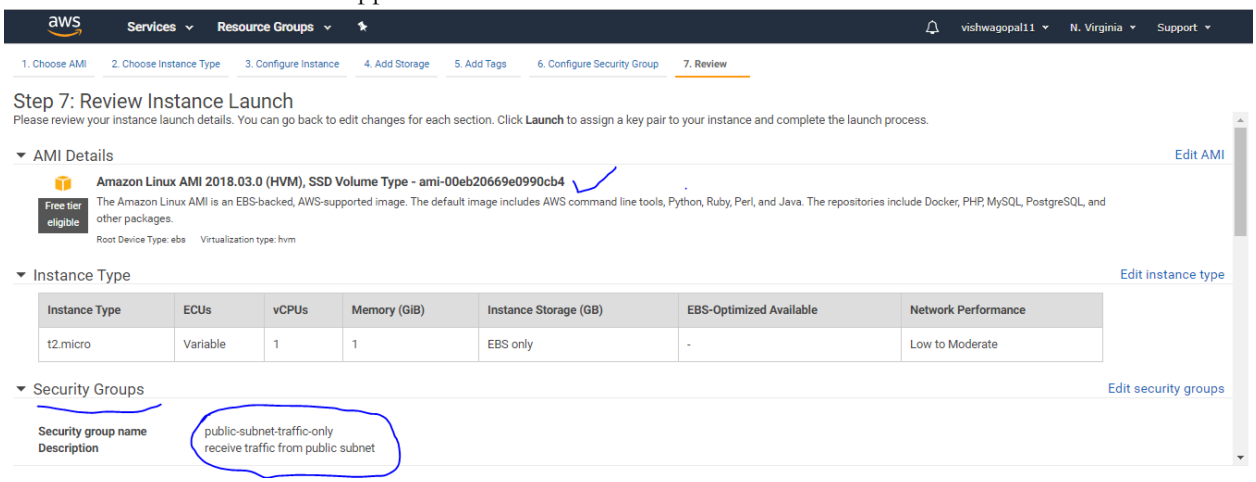
- Change default site directory to owncloud files directory using sudo user

1. edit /etc/apache2/sites-enabled/000-default.conf 2. update directory root path to /var/www/owncloud
3. restart the server - sudo systemctl reload apache2

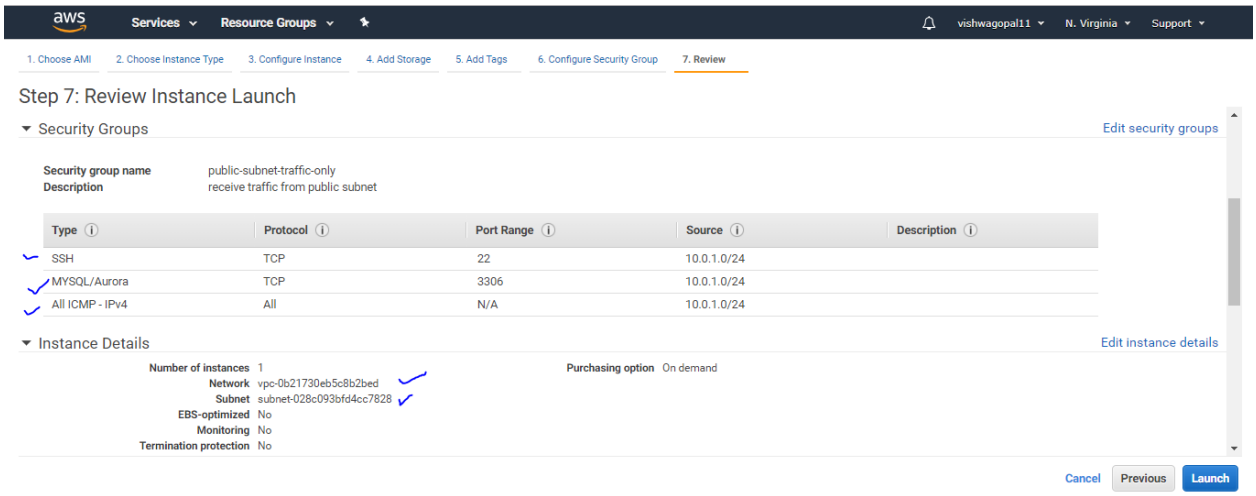


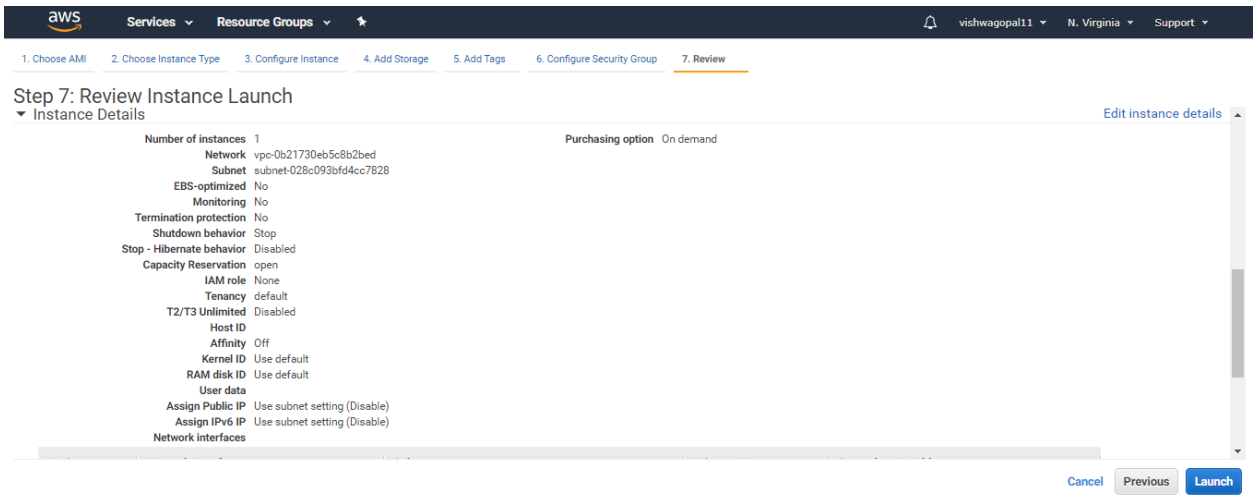
DB Server Setup in EC2:

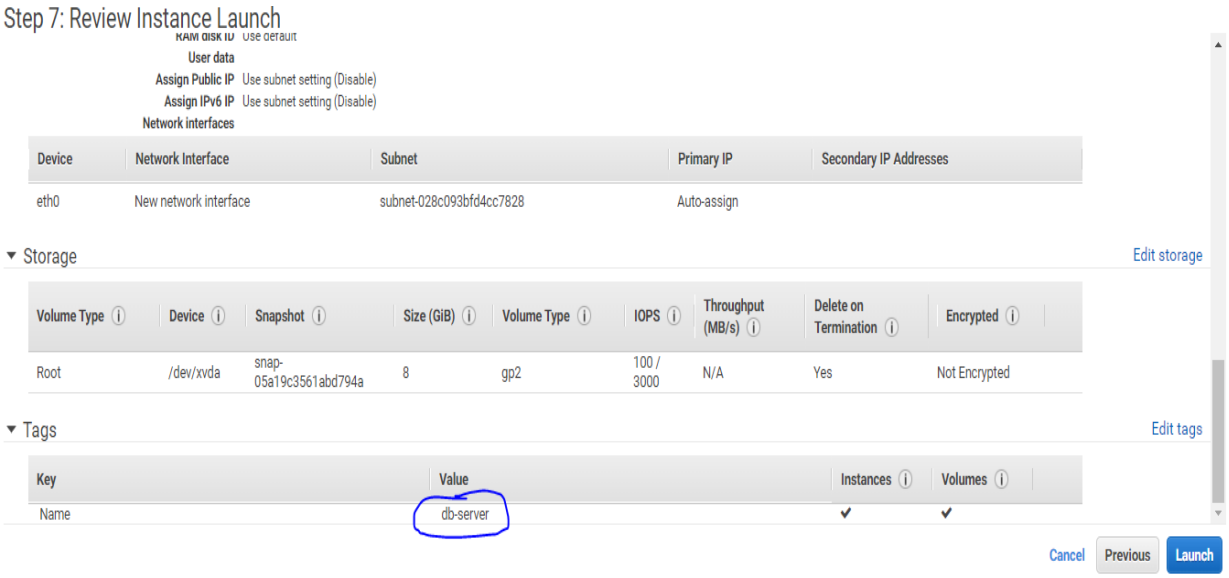
Setting up DB server on the Private Subnet and connecting via NAT gateway to allow interaction between the appserver and DB server



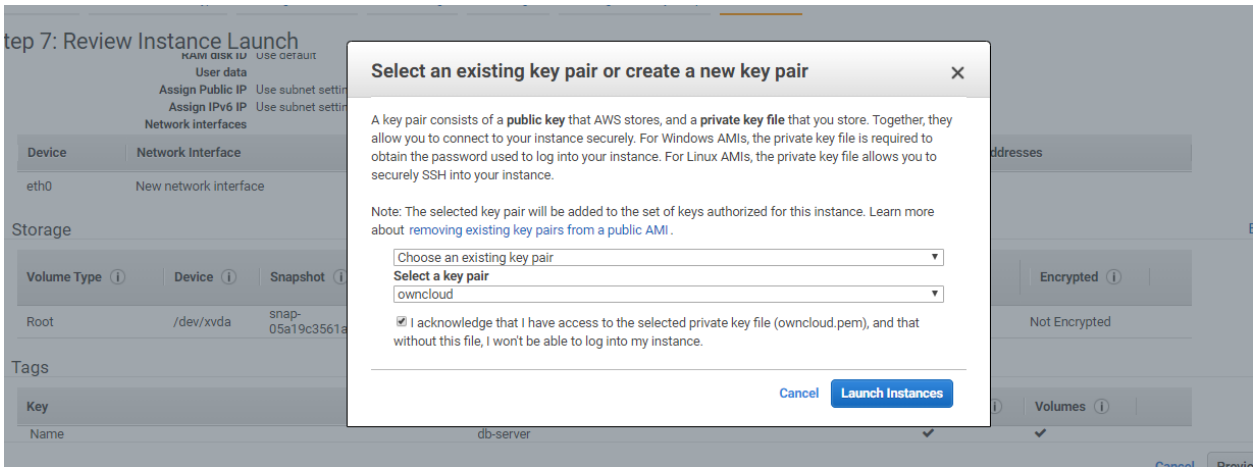
- DB server – Private subnet – Linux instance review pages

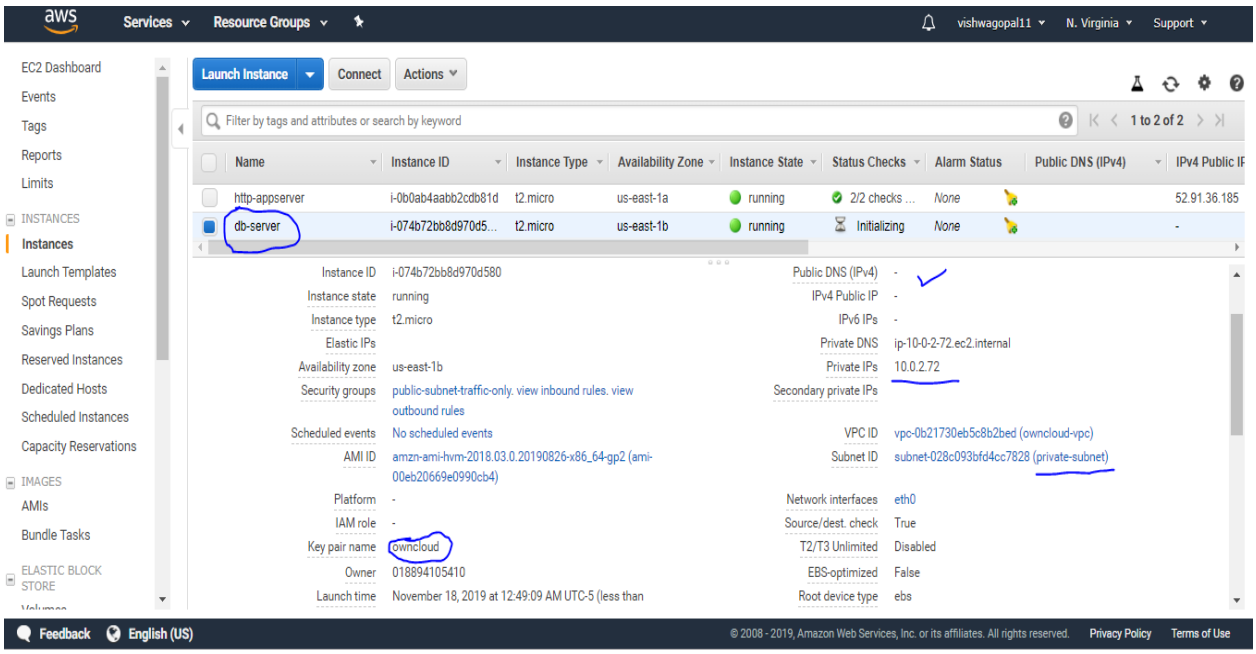






- Key pair selection in the EC2 instance creation steps

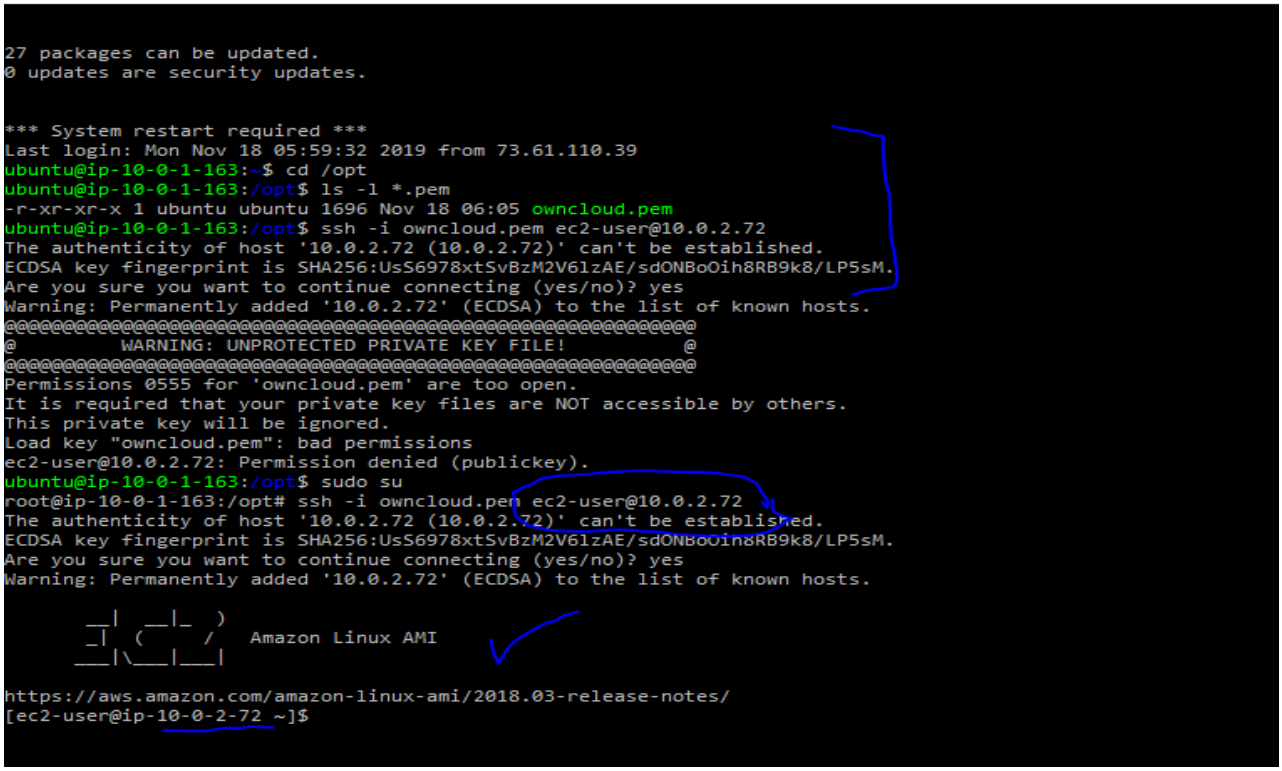




- The DB server is now created and in order to interact with the app server internet private connection is required
- This will need to be setup via NAT gatway by connecting to Publiv subnet that acts as the Bastion Host
- To SSH into the Private subnet DB instance – The .pem file is required.

```
ec2-user@ip-10-0-2-72:~
51 packages can be updated.
24 updates are security updates.


Last login: Mon Nov 18 03:35:05 2019 from 73.61.110.39
ubuntu@ip-10-0-1-163:~$
ubuntu@ip-10-0-1-163:~$
ubuntu@ip-10-0-1-163:~$ cd /opt/
ubuntu@ip-10-0-1-163:/opt$ sudo chown ubuntu:ubuntu -R /opt
ubuntu@ip-10-0-1-163:/opt$ ls -al
total 8
drwxr-xr-x  2 ubuntu ubuntu 4096 Oct  2 17:08 .
drwxr-xr-x 23 root   root   4096 Nov 18 06:01 ..
ubuntu@ip-10-0-1-163:/opt$ exit
logout
Connection to 52.91.36.185 closed.
root@DESKTOP-49DC5BB:/mnt/c/Users/Vishw/Downloads# ls - al *.pem
ls: cannot access '-': No such file or directory
ls: cannot access 'al': No such file or directory
 gl.pem  'glpem1 (1).pem'  'glpem1 (2).pem'   greatlearning.pem   owncloud.pem
root@DESKTOP-49DC5BB:/mnt/c/Users/Vishw/Downloads# scp -i owncloud.pem ./owncloud.pem ubuntu@52.91.36.185:/opt
owncloud.pem
root@DESKTOP-49DC5BB:/mnt/c/Users/Vishw/Downloads#
root@DESKTOP-49DC5BB:/mnt/c/Users/Vishw/Downloads# ssh -i owncloud.pem ubuntu@52.91.36.185
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1051-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Nov 18 06:05:53 UTC 2019

  System load:  0.01              Processes:           94
  Usage of /:   21.5% of 7.69GB   Users logged in:     0
  Memory usage: 20%               IP address for eth0: 10.0.1.163
  Swap usage:   0%


27 packages can be updated.
0 updates are security updates.
```

ec2-user@ip-10-0-2-72:~



EC2 instances are accessed via ssh as per the screenshot

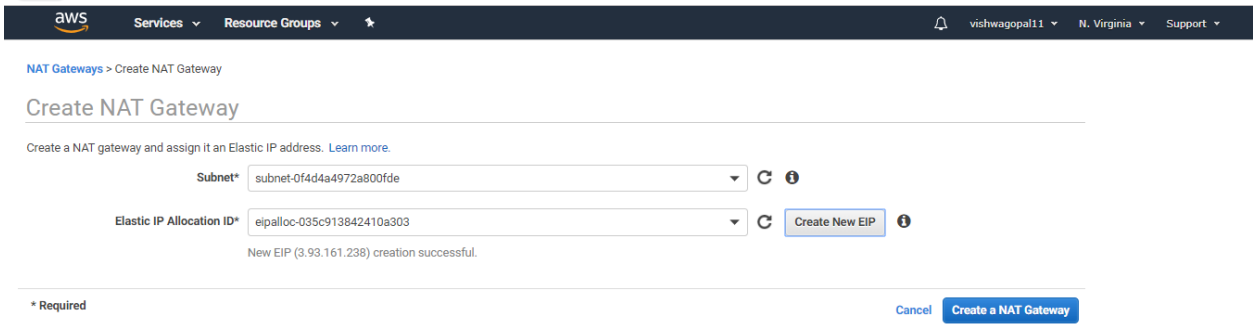
- As a first step we copy the owncloud.pem file from the ubuntu public subnet EC2 instance into an opt folder where we change the owner of the file after which we use the scp command to copy the file into the EC2 instance and then ssh into the private DB instance using the Private IP address
- Public EC2 insatnce – Ubuntu (Username: Ubuntu)
- Private EC2 instance – Linux (Username: ubuntu)
- The pem file can be SSHed using the super user role from either of the instances but it does not connect to the internet

```
[ec2-user@ip-10-0-2-72 ~]$ sudo yum update
Failed to set locale, defaulting to C
Loaded plugins: priorities, update-motd, upgrade-helper
Could not retrieve mirrorlist http://repo.us-east-1.amazonaws.com/latest/main/mirror.list error was
12: Timeout on http://repo.us-east-1.amazonaws.com/latest/main/mirror.list: (28, 'Connection timed out after 5001 milliseconds')


 One of the configured repositories failed (Unknown),
 and yum doesn't have enough cached data to continue. At this point the only
 safe thing yum can do is fail. There are a few ways to work "fix" this:

     1. Contact the upstream for the repository and get them to fix the problem.

     2. Reconfigure the baseurl/etc. for the repository, to point to a working
        upstream. This is most often useful if you are using a newer
        distribution release than is supported by the repository (and the
        packages for the previous distribution release still work).

     3. Disable the repository, so yum won't use it by default. Yum will then
        just ignore the repository until you permanently enable it again or use
        --enablerepo for temporary usage:

            yum-config-manager --disable <repoid>

     4. Configure the failing repository to be skipped, if it is unavailable.
        Note that yum will try to contact the repo. when it runs most commands,
        so will have to try and fail each time (and thus. yum will be be much
        slower). If it is a very temporary problem though, this is often a nice
        compromise:

            yum-config-manager --save --setopt=<repoid>.skip_if_unavailable=true

Cannot find a valid baseurl for repo: amzn-main/latest
[ec2-user@ip-10-0-2-72 ~]$
```

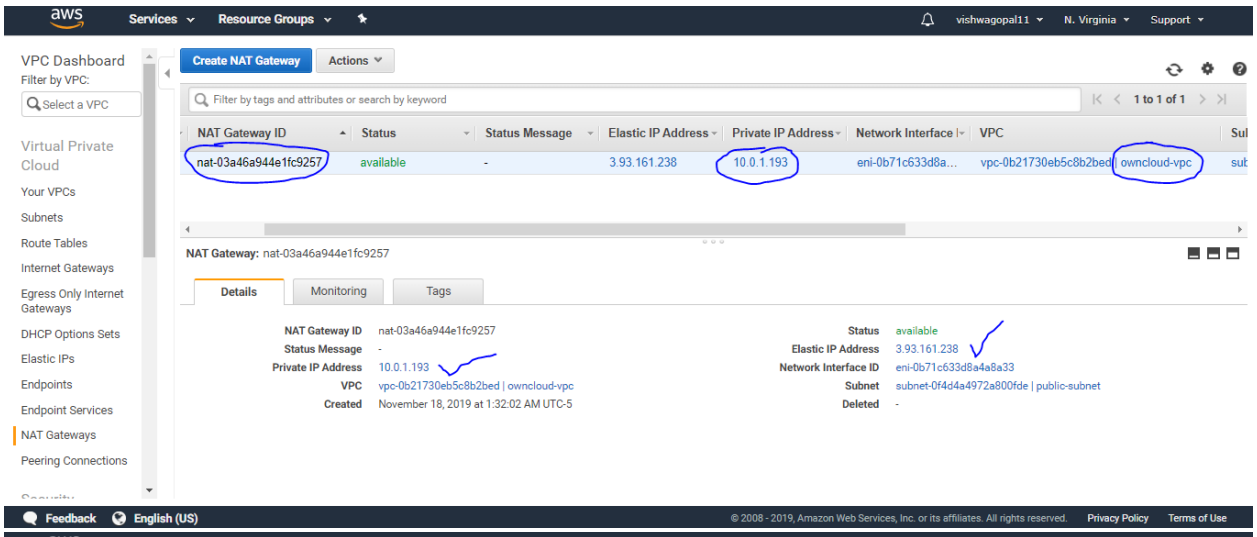
NAT Gateway creation:

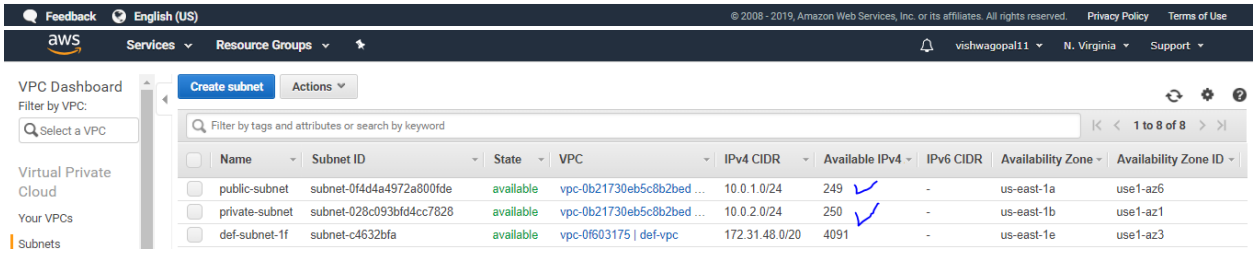
To let the DB server connect to the internet, a NAT gateway needs to be created. And then a custom route table needs to be created that will route traffic only between the public EC2 instance which has the app server and the private DB server instance. For the current solution, the default route table will be used for the NAT gateway setup

NAT gateway created will need to be attached the owncloud VPC which has the internet connectivity

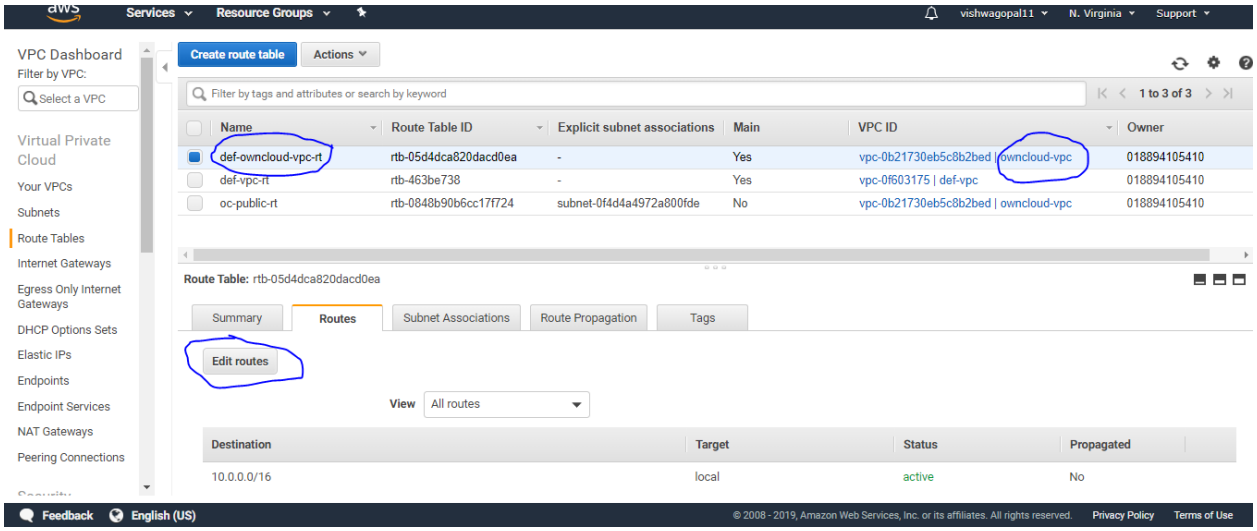


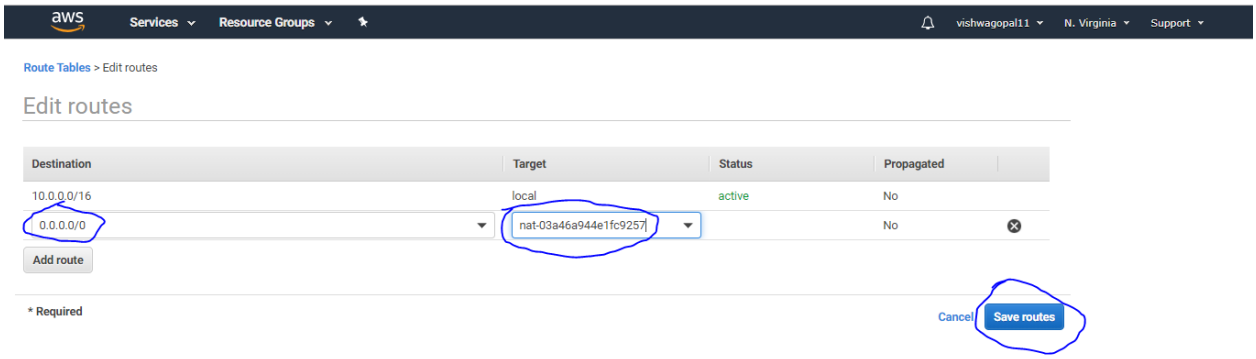
The NAT gateway created will need to be associated with a route table (selecting the default route table def-owncloud-vpc-rt).

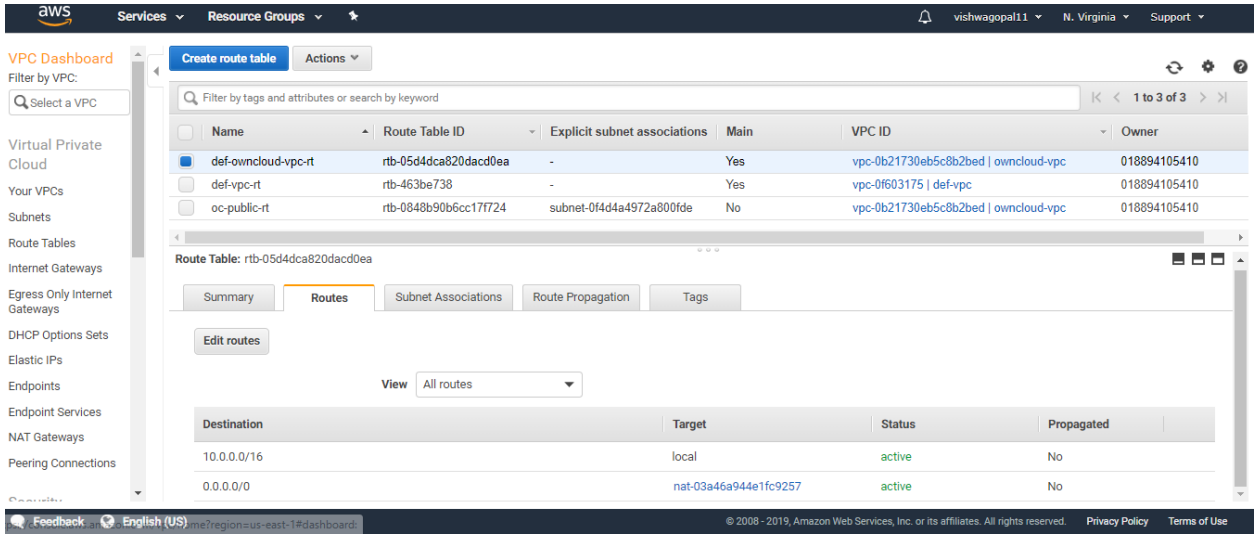




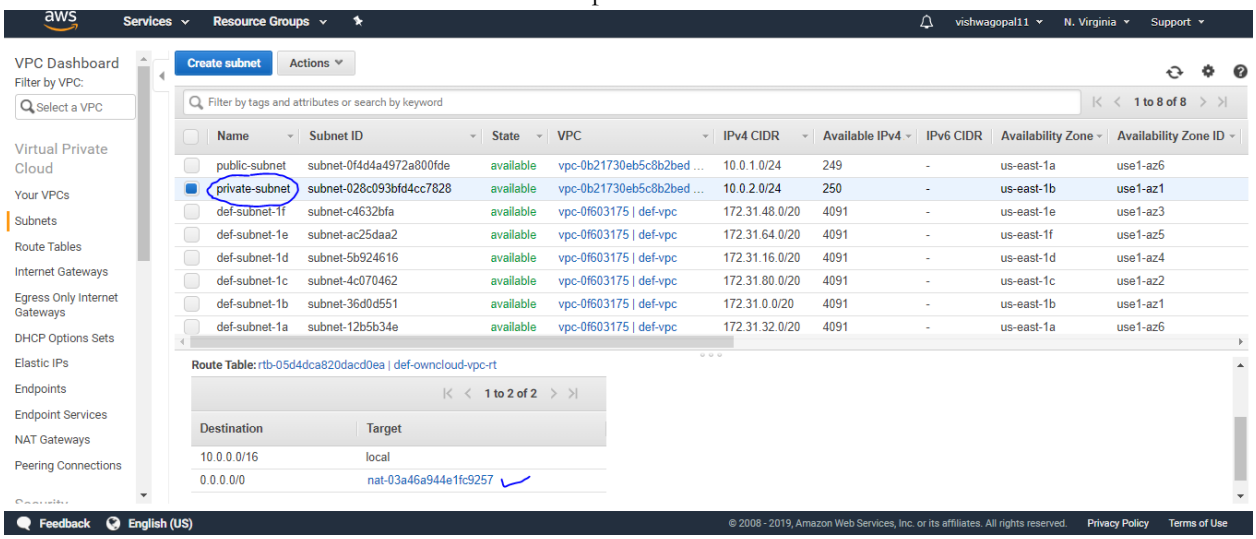
Associate the default route table with the NAT gateway instance through subnet associations and adding the route to the destination:



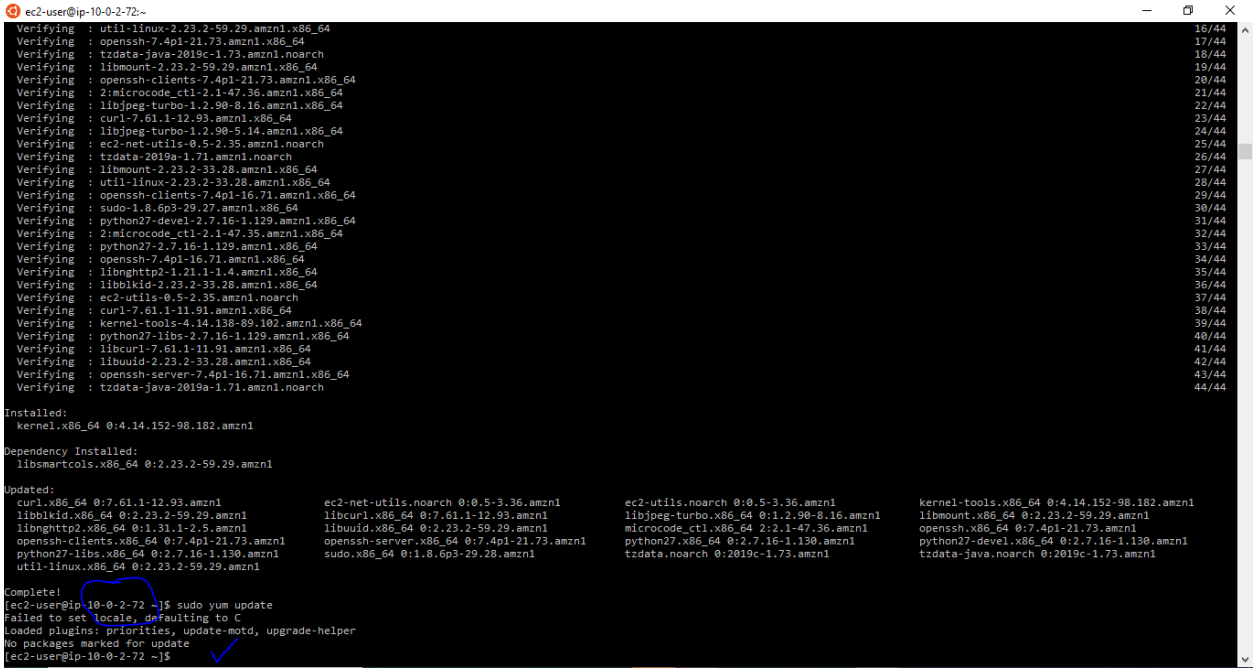




Default VPC route table is now associated with the private subnet

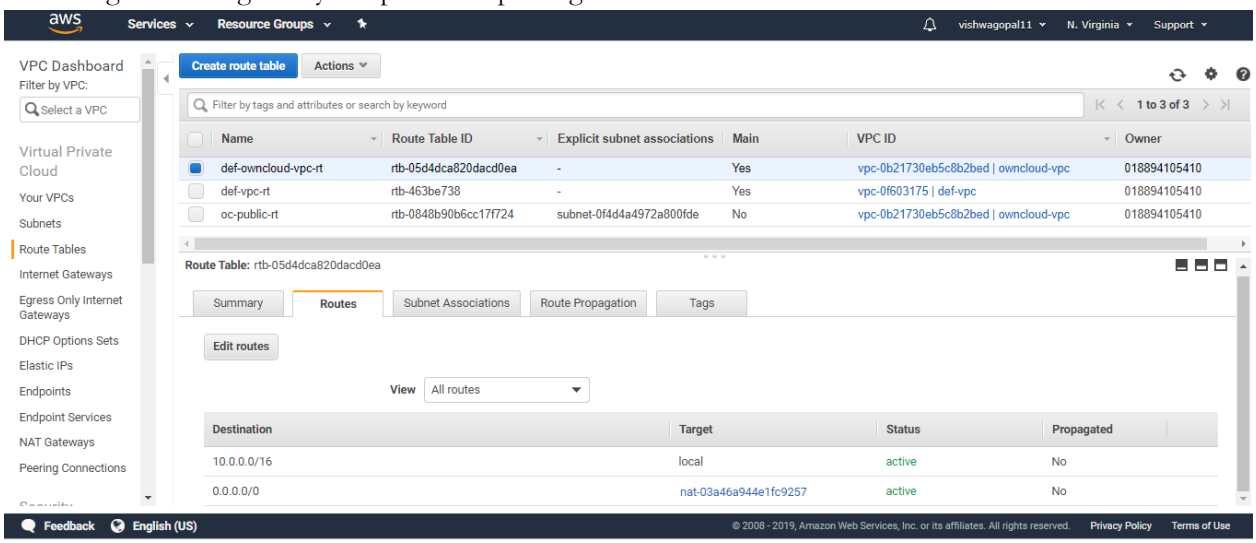


Sudo yum update on the private instance now works



Note: NAT instances can also be created to achieve the connectivity of the private subnet with a default VPC via a route table setup (NAT instances can be found via Community instances)

Validating the NAT gateway setup before updating the DB server instance via the console



Executing the below commands in the DB server console with ec2-user@ip-10-0-2-72

sudo apt-get install mysql-server

sudo mysql_secure_installation

sudo mysql

Create new DB and user with all privileges

CREATE DATABASE owncloud

CREATE USER ‘owncloud’ @ ‘localhost’ IDENTIFIED BY ‘password’ ;

CREATE USER ‘owncloud’ @ ‘%’ IDENTIFIED BY ‘password’ ;

GRANT ALL PRIVILEGES ON *.* to owncloud@localhost IDENTIFIED BY 'password' WITH GRANT OPTION; GRANT ALL PRIVILEGES ON *.* to owncloud@'%' IDENTIFIED BY 'password' WITH GRANT OPTION; FLUSH PRIVILEGES;
EXIT;

Change the bind address

sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf
sudo systemctl restart mysql

On web server install mysql client

mysql -uowncloud -h -p

Mysql-server DB successfully installed:

Screenshot accessing the DB installation in the console and the commands were executed