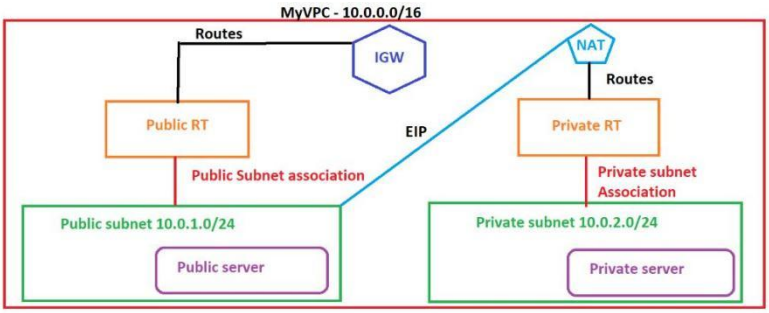
AWS Virtual Private Cloud

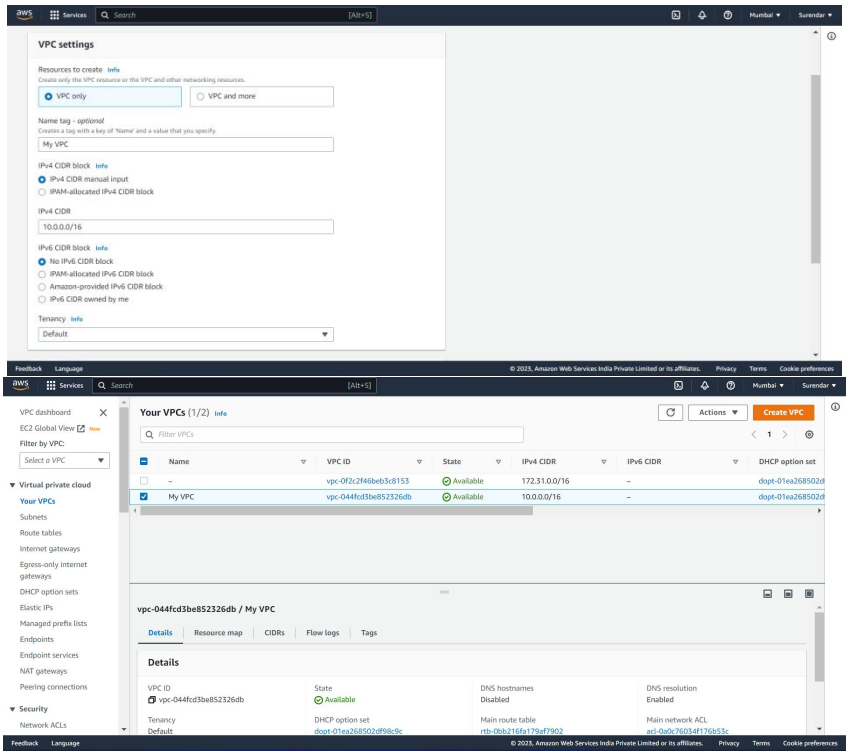
Steps to Create and Configure your VPC for Public and Private Server in Windows.

My VPC Architecture



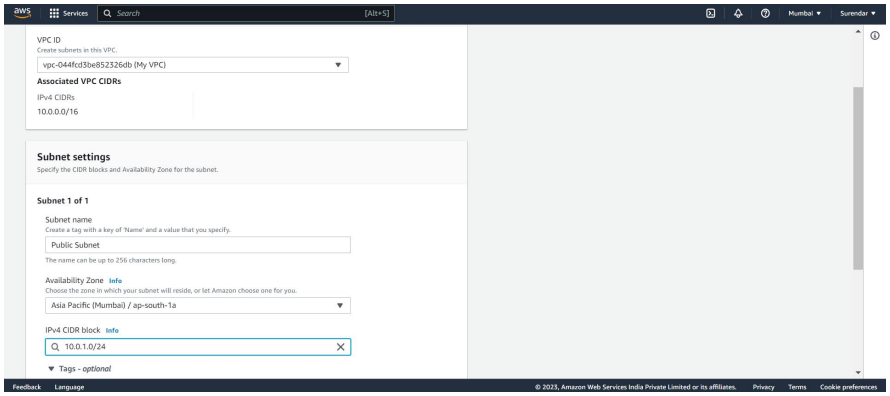
STEP 1

In yourVPC - Create VPC - Set a name (**My VPC**) - IPv4 CIDR (**10.0.0.0/16**) - Tenancy (**Default**) - **create a VPC**.



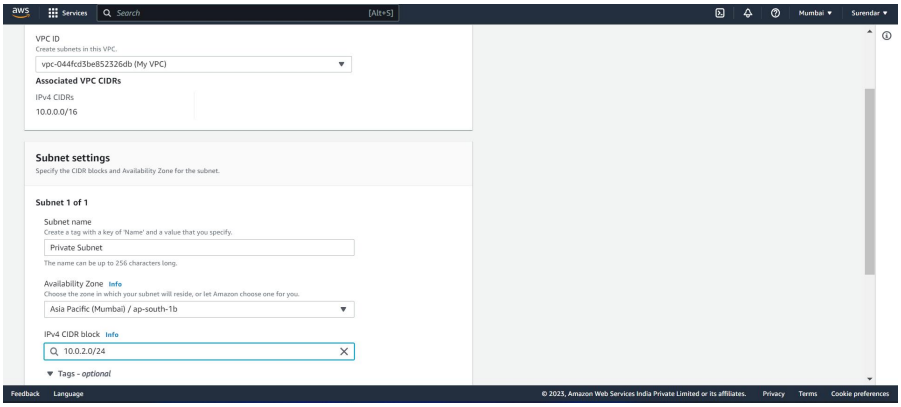
STEP 2

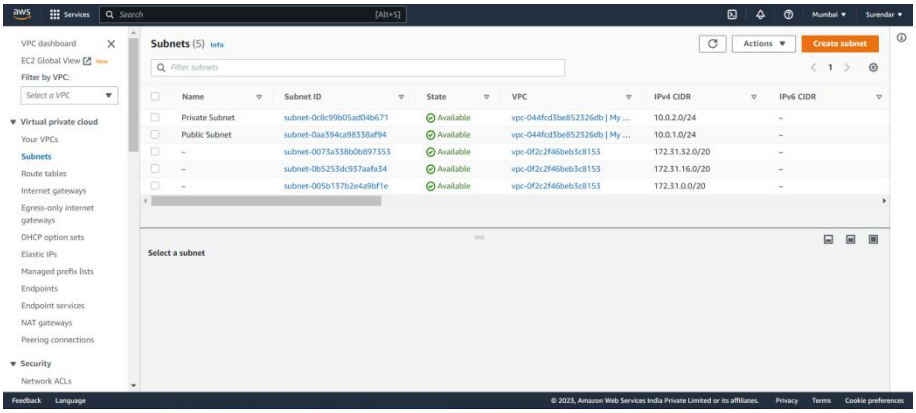
In **Subnets** - Create **Subnet** - Set a name (**Public Subnet**) - Avail Zone (**1a**) - IPv4 CIDR block (**10.0.1.0/24**) - create a public subnet.



STEP 3

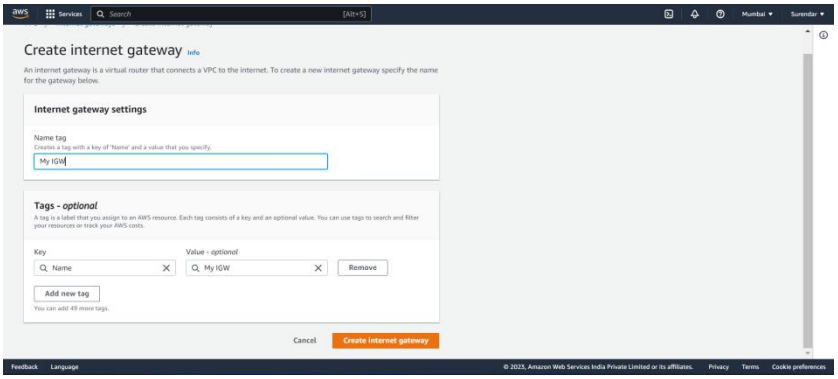
Create **Subnet** - Set a name (**Private Subnet**) - Avail Zone (**1b**) - IPv4 CIDR block (**10.0.2.0/24**) - create a private subnet.



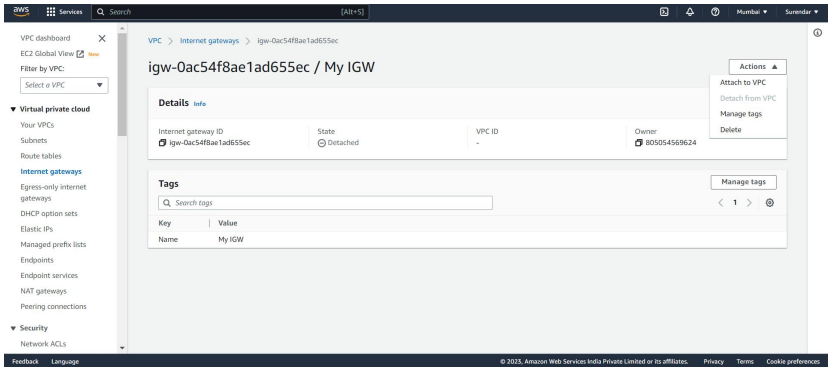


STEP 4

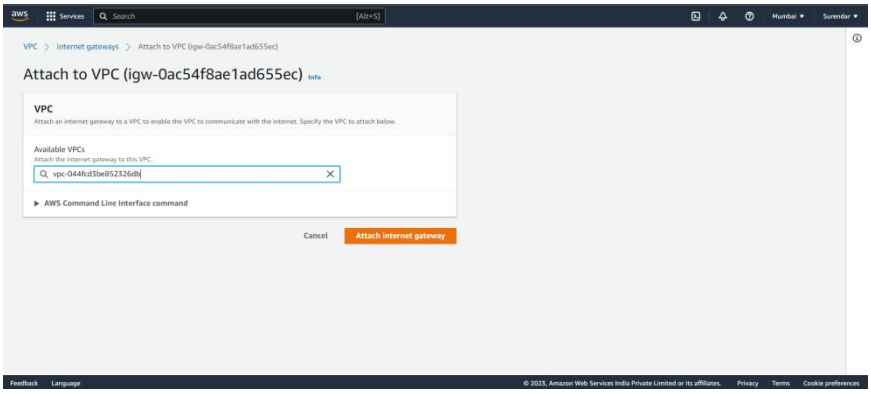
In Internet Gateway - Create Internet Gateway - Set a name (**MY IGW**) - **create**.



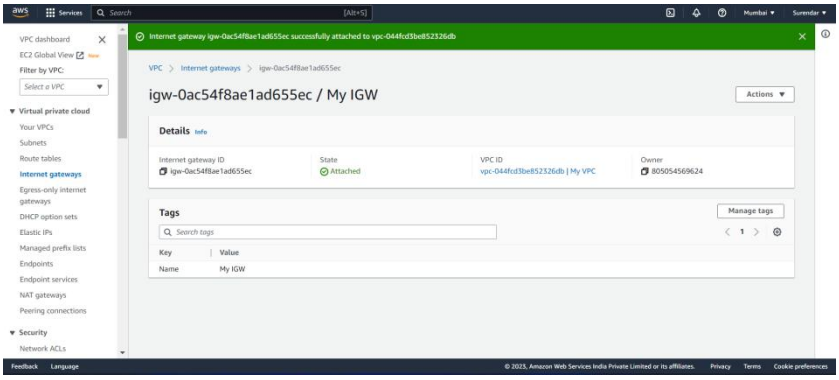
After created IGW it will be in **Detached** - In **Action** - Select **Attach to VPC**.



Choose **MY VPC** - Attach internet Gateway.

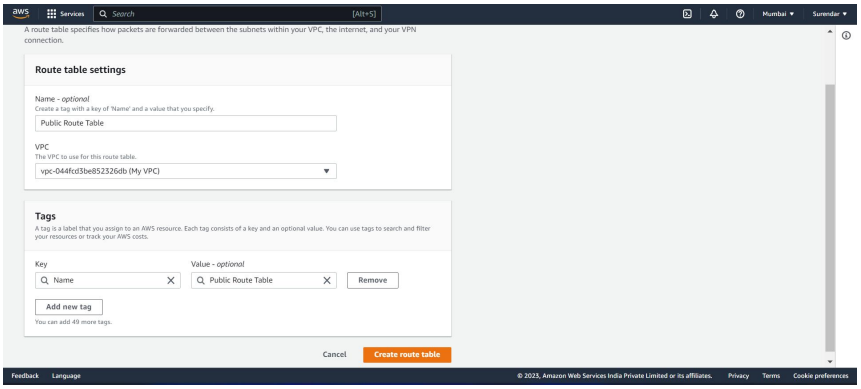


Now Successfully **Attached** My VPC in Internet gateway.

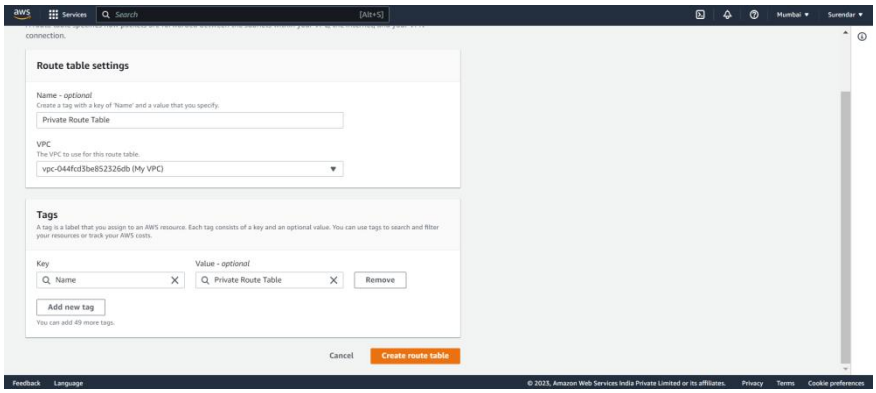


STEP 5

In Route Tables - Create - Set a name (**Public Route Table**) - Choose **My VPC** - Create route table.



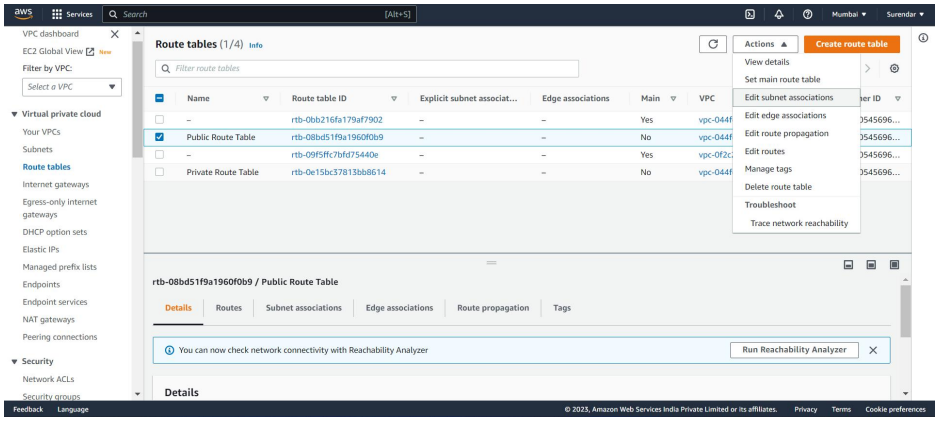
Create - Set a name (**Private Route Table**) - Choose **My VPC** - Create route table.



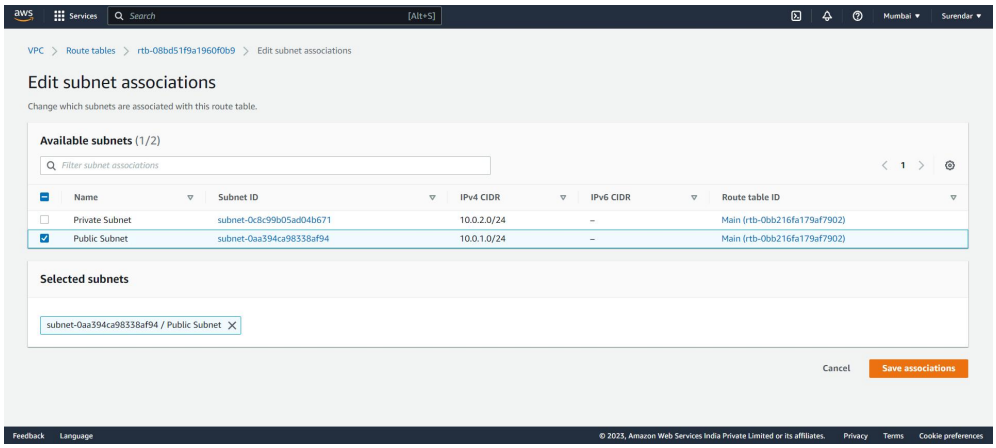
STEP 6

create a subnet association for both public and private route tables

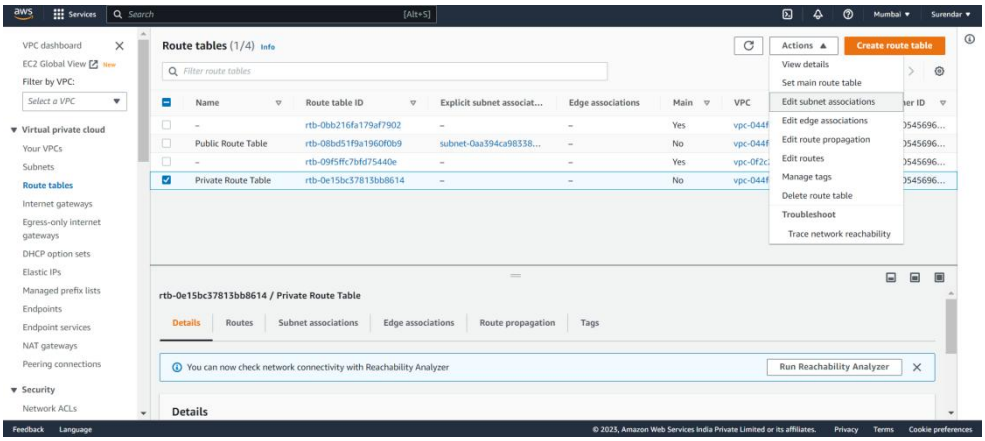
In route table - Select **Public route table** - Action - edit subnet Association.



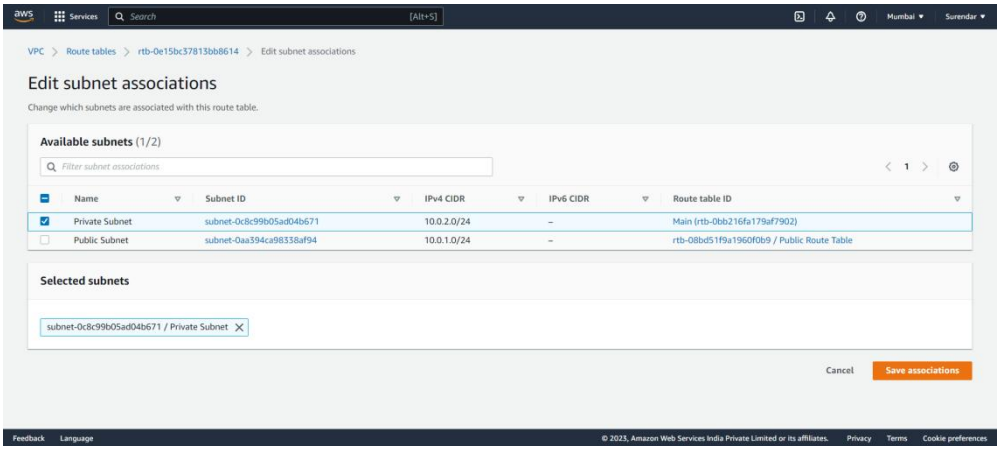
Choose **Public Subnet** - Save association.



Select **Private route table** - Action - edit subnet Association.

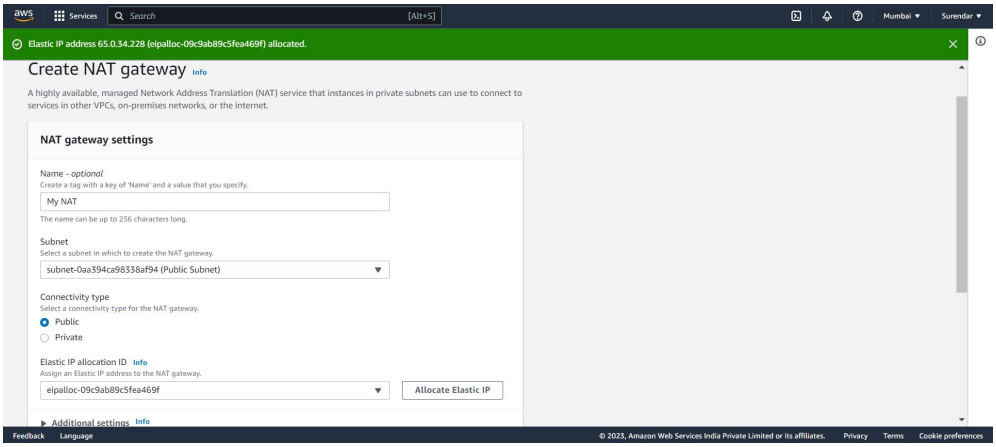


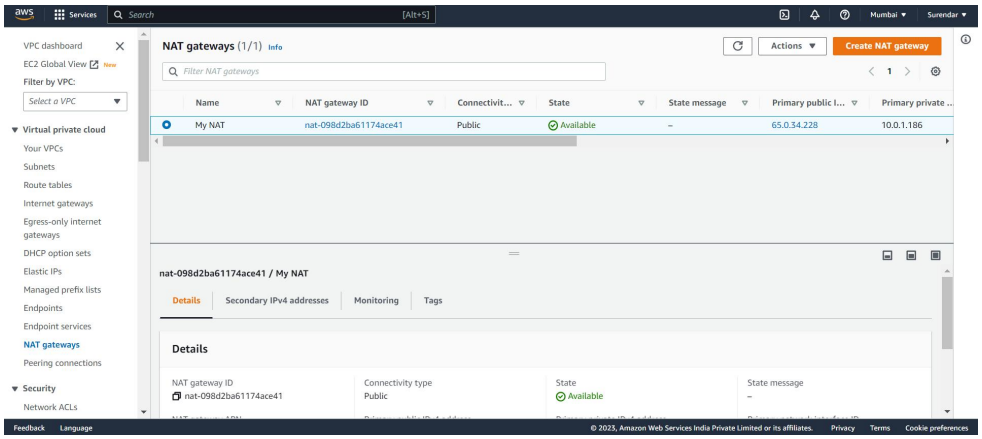
Choose **Private Subnet** - Save association.



STEP 7

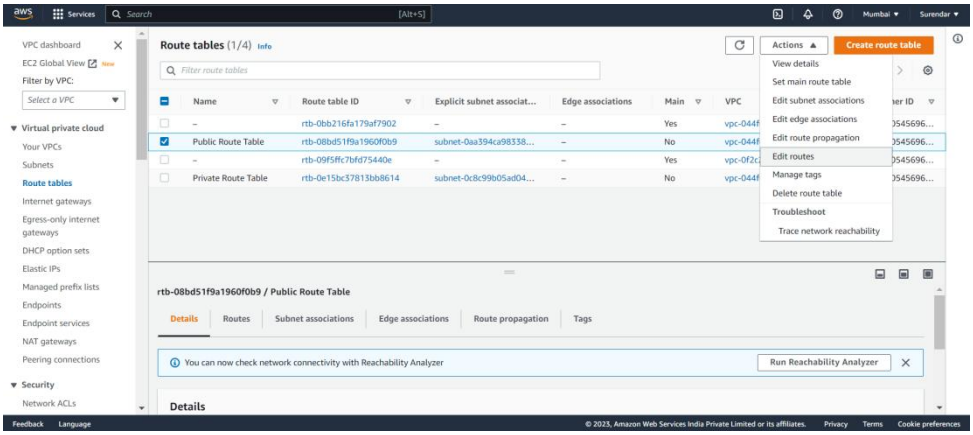
In NAT gateway - create NAT gateway - Set a name (**My NAT**) - Subnet choose (**Public Subnet**) - **Allocate Elastic IP** - create NAT gateway.



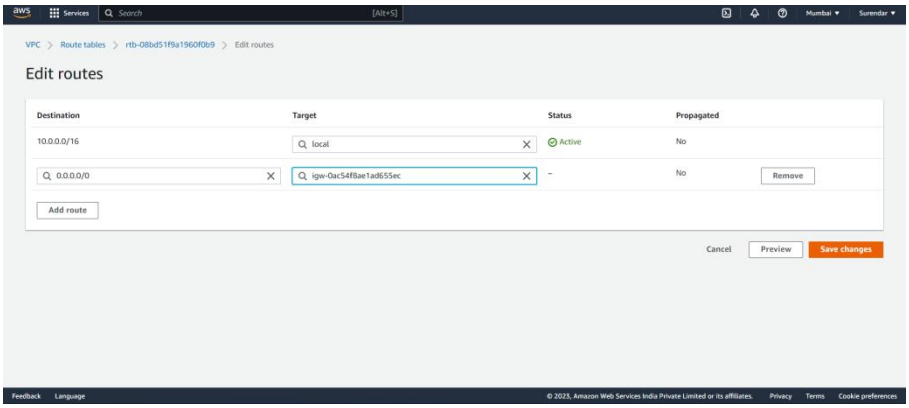


STEP 8

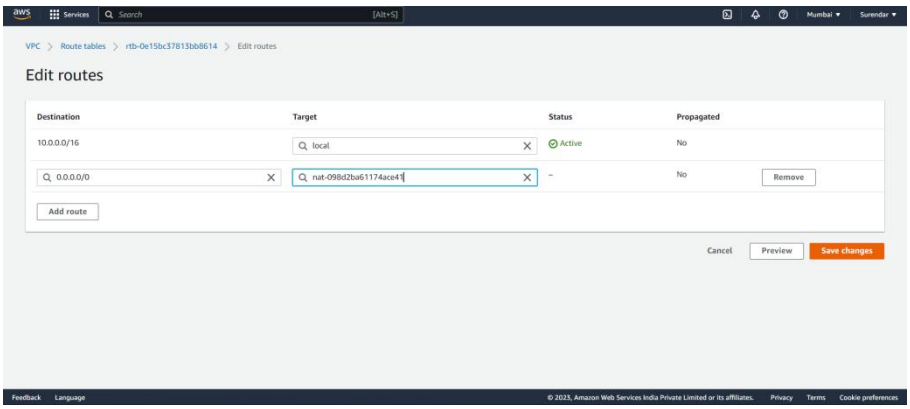
Create a Route - select **Public route table** - Actions - **edit routes**.



Destination (**0.0.0.0/0**) open IP - Target (**Internet Gateway IP**) - save changes.



Select (**Private route table**) - Action - **edit routes** - Destination (**0.0.0.0/0**) open IP - Target choose **NAT gateway (my NAT)** - save changes.



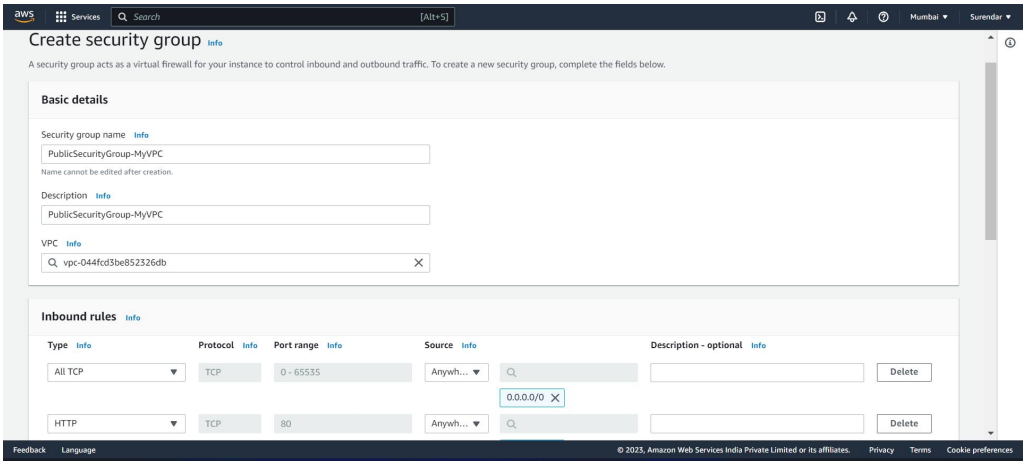
STEP 9

In security group - create -Set a name (**PublicSecurityGroup-MyVPC**)-Description (**PublicSecurityGroup-MyVPC**) - Remove default VPC - Add **My VPC** -

Inbound Rule - Type :**All TCP** - Source: **Anywhere IPv4** -

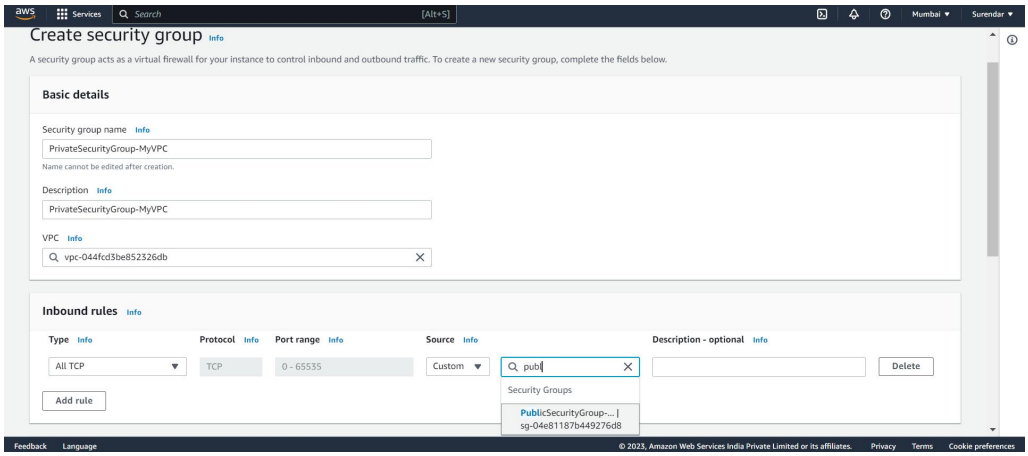
Type :**HTTP** - Source:**Anywhere IPv4**.

Create Security group.



create -Set a name (**PrivateSecurityGroup-MyVPC**)-Description (**PrivateSecurityGroup-MyVPC**) - Remove default VPC - Add **My VPC** -

Inbound Rule - Type :**All TCP** - Source:**PublicSecuritygroup** - Create Security Group.

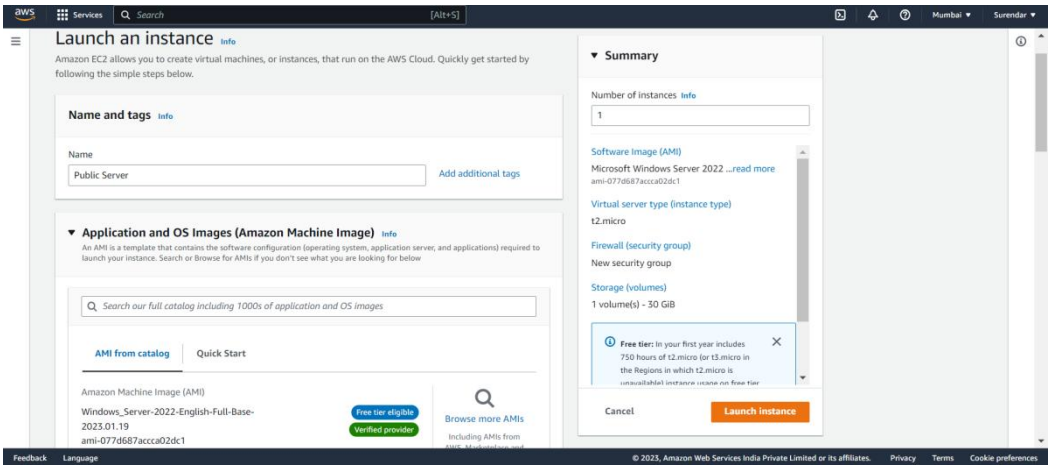


STEP 10

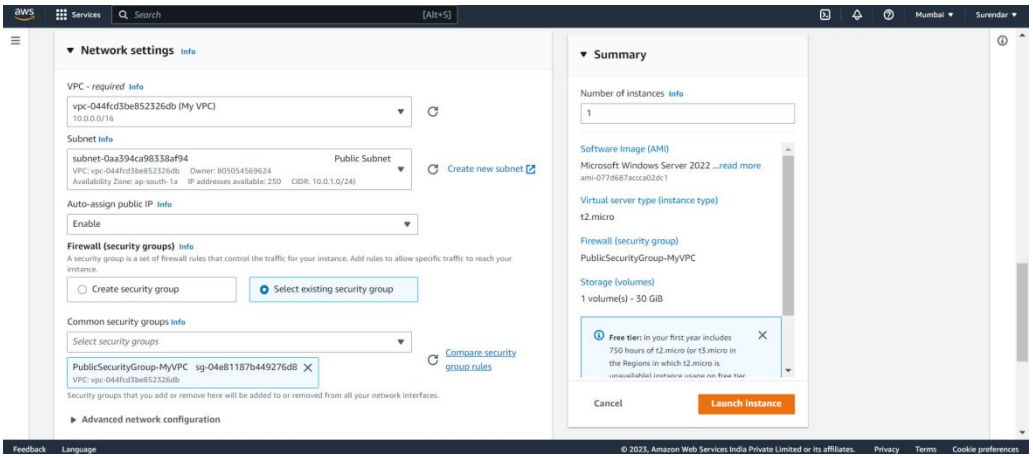
Then, In EC2 - launch 2 instance in windows server as a name of Public and Private server.

Create a Public Instance

Set a name (**Public Server**) - choose **Windows OS** in AMI - Choose .pem key file.

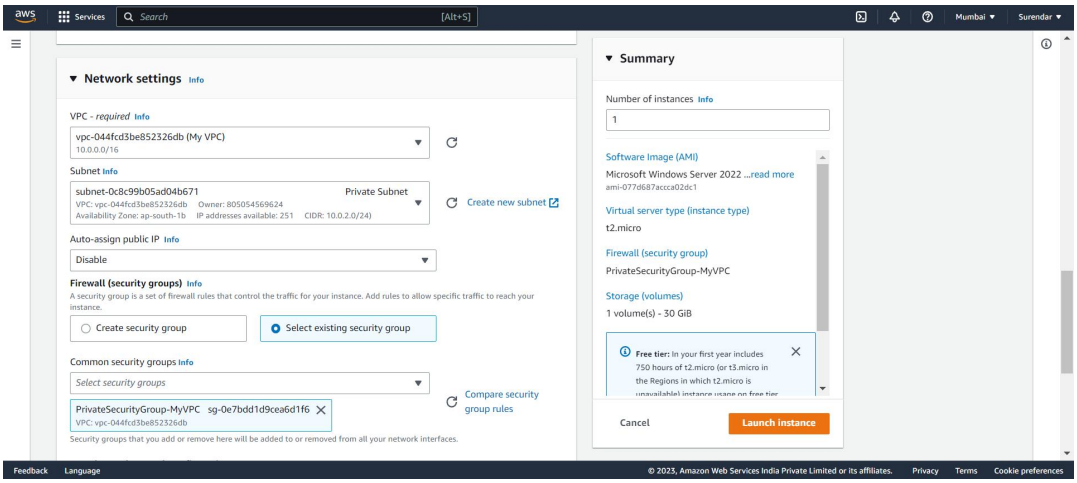


Network setting - edit - choose **My VPC** - Subnet choose **Public Subnet** - Auto-assign publicIP (**Enable**) - Comon Security group Choose (**publicsecuritygroup-MyVPC**) - Launch instance.



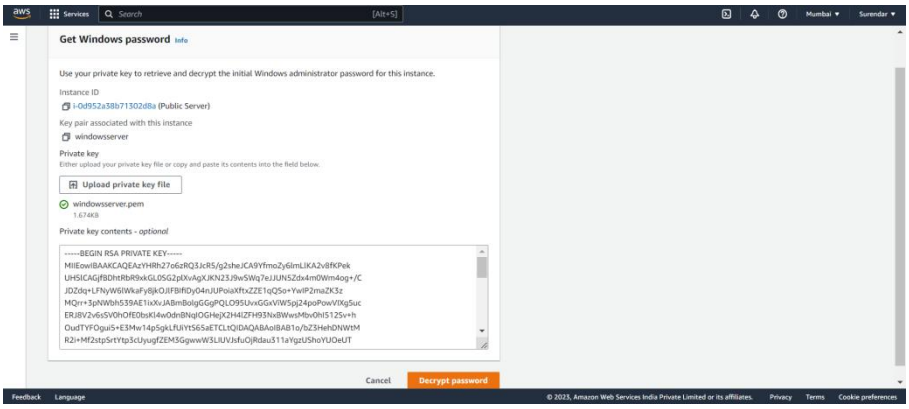
Now create private instance

Set a name (**Private Server**) - Choose **windows OS** in AMI - Choose **.pem key file** - Network setting (edit) -VPC (**My VPC**)- Subnet choose (**Private Subnet**) - Auto-assign publicIP (**Disable**) - common security group choose (**Privatesecuritygroup-MyVPC**) - Launch Instance.



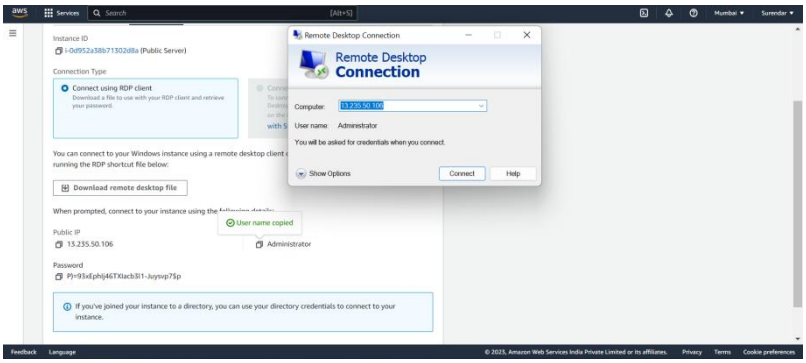
STEP 11

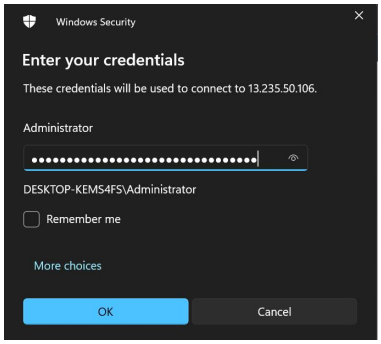
Connect the **Public server instance** - **decrypt password**

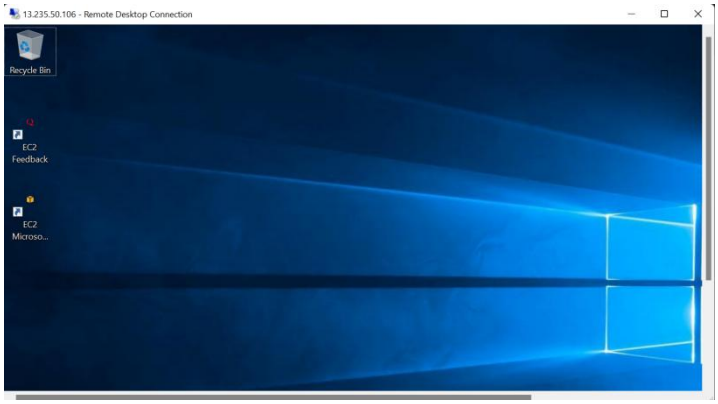


STEP 12

Open Remote Desktop Connection - Connect the **public Windows server** using Specific credentials.







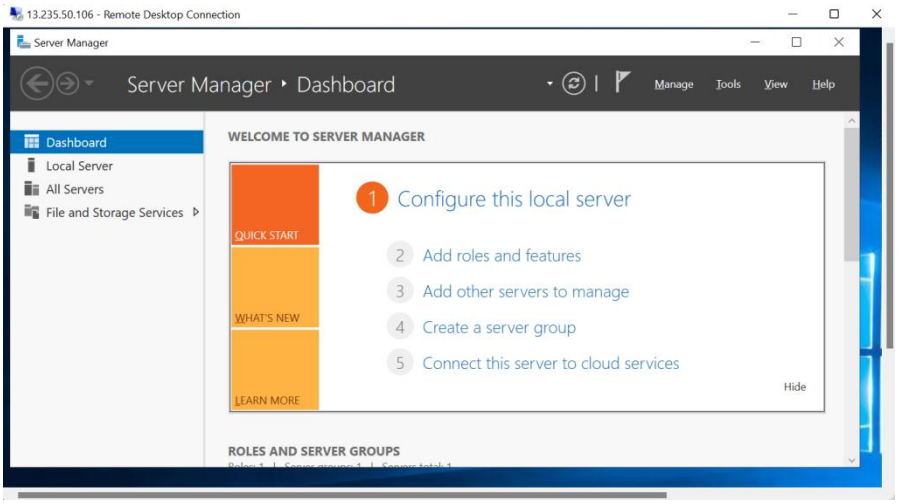
STEP 13

After Successfully connect the Public windows server in RDP.

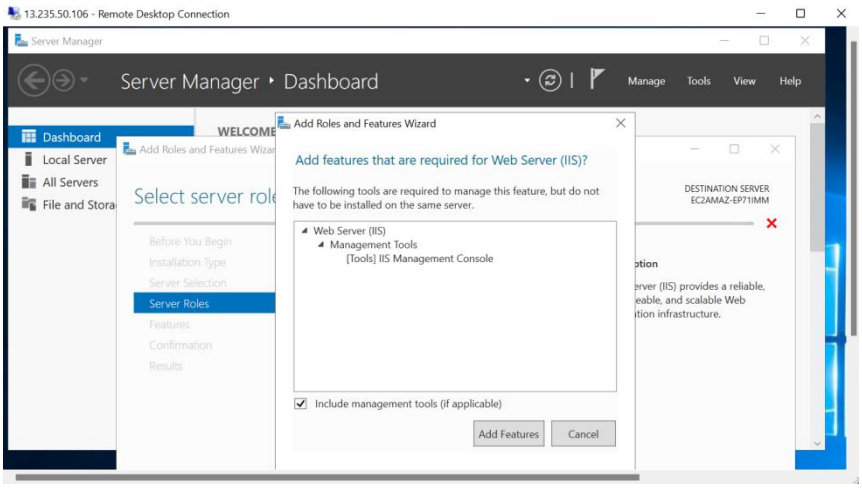
Now, Install webserver (IIS) to host a website through Public server.

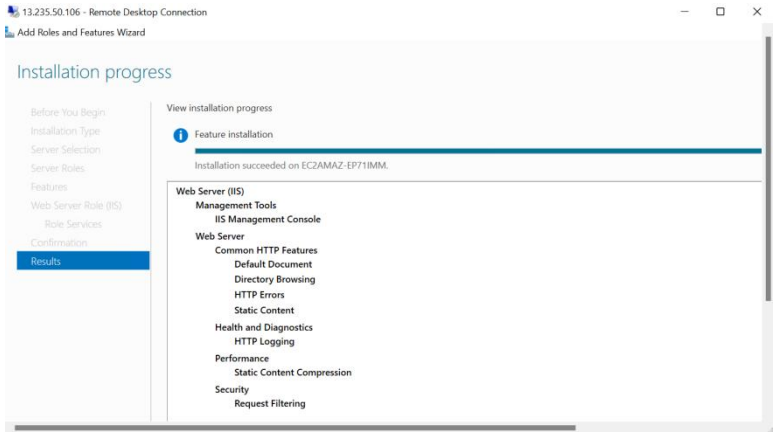
In Remote Desktop Connection

Open Server manager - Add roles and features.



Select Web Server IIS - Install IIS.



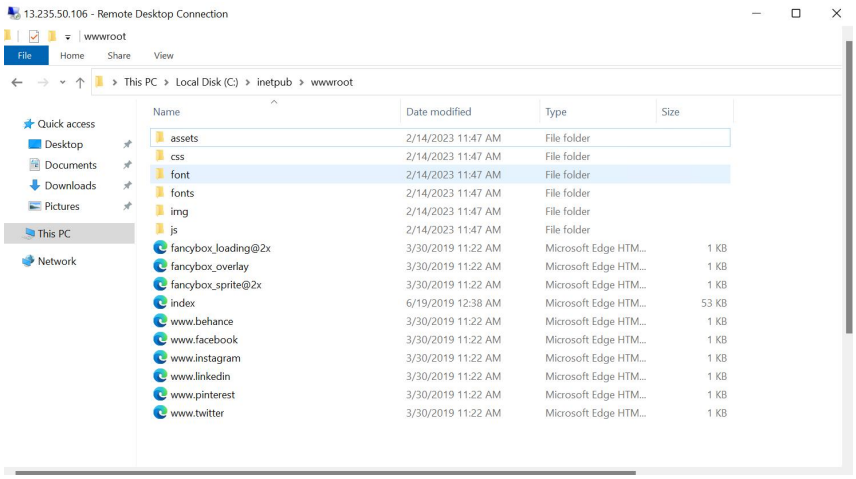


STEP 14

In, Remote Desktop Connection

After Successfully installed **IIS webserver in Public server.**

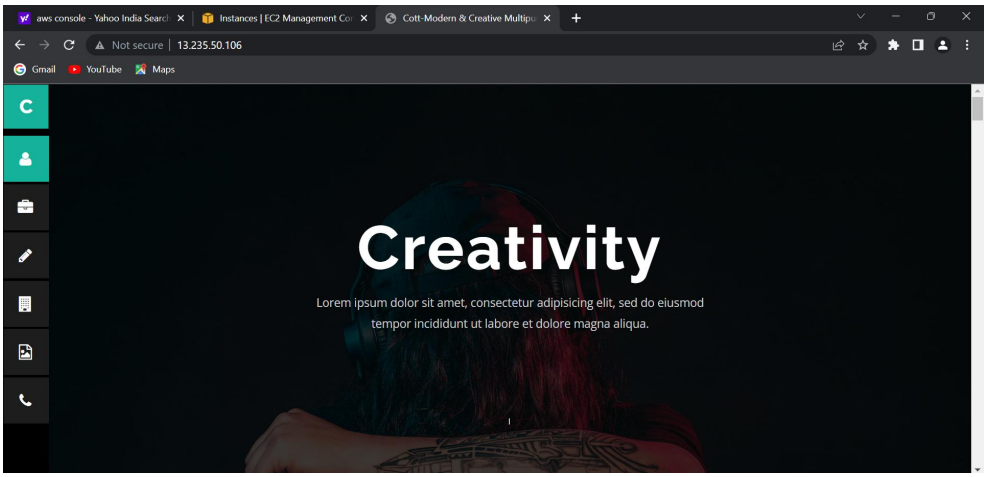
Open filebrowser - Localdisk C - inetpub - [**www.root**] - remove default files - paste website content files.



STEP 15

In chrome

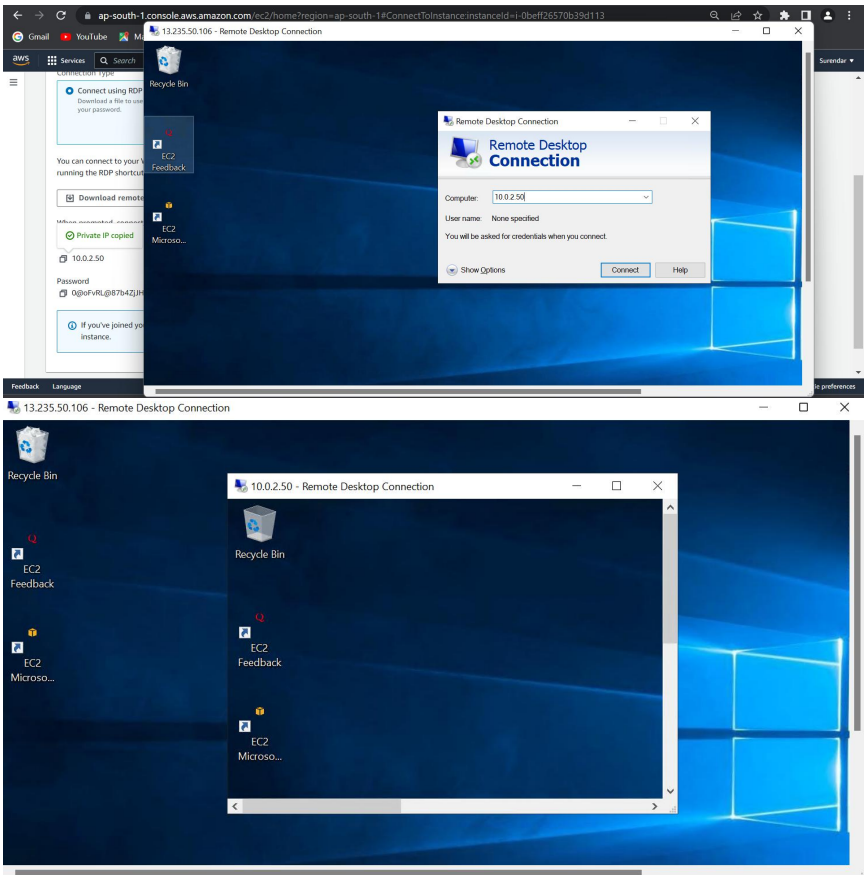
Paste Public server Instance IPv4 address as (http://13.235.50.106)



Successfully config yourVPC and hosted a website in public server.

STEP 16

Connect the **private server Instance** - **Decrypt Password** - After login as a public server in RDP - Open **Remote Desktop connection** - enter the **private server specific credentials**



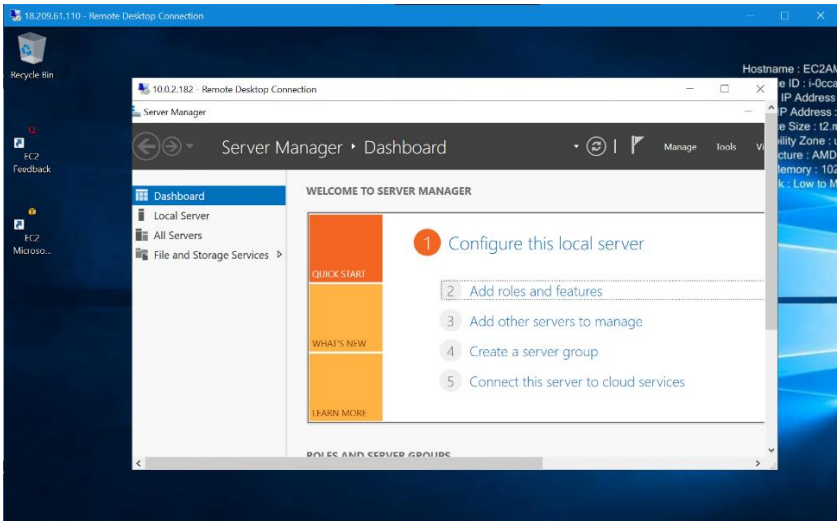
STEP 17

After Successfully connect the Private windows server in Public windows Server using RDP.

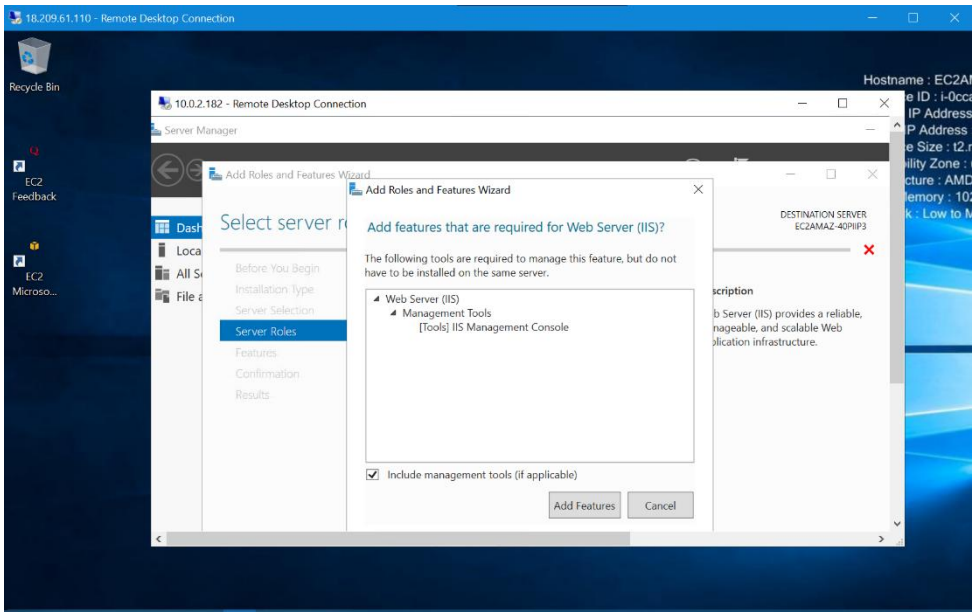
Now, Install webserver (IIS) to host a website through Private Server IP address.

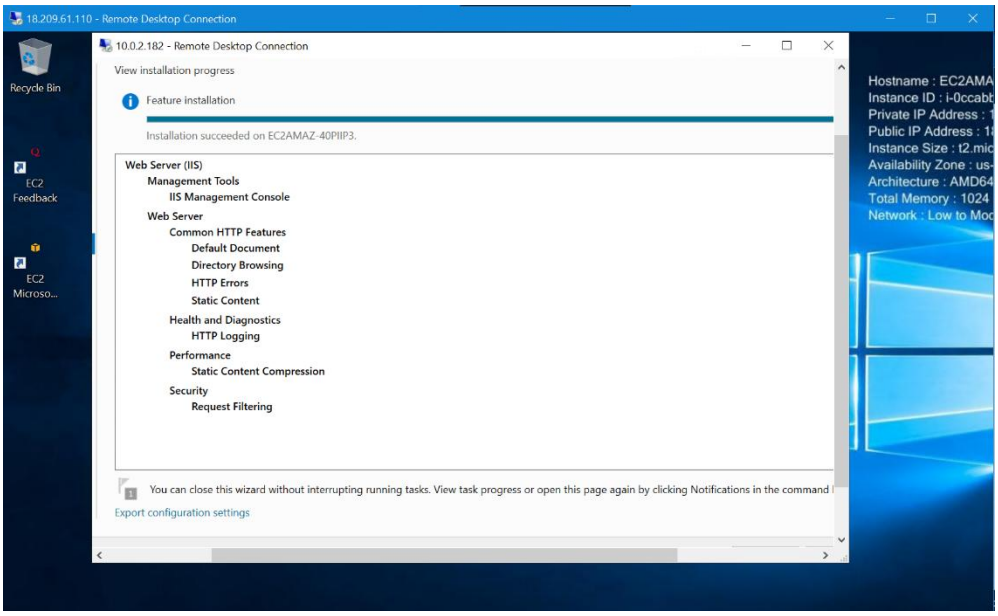
In Remote Desktop Connection

Open Server manager - Add roles and features.



Select Web Server IIS - Install IIS.



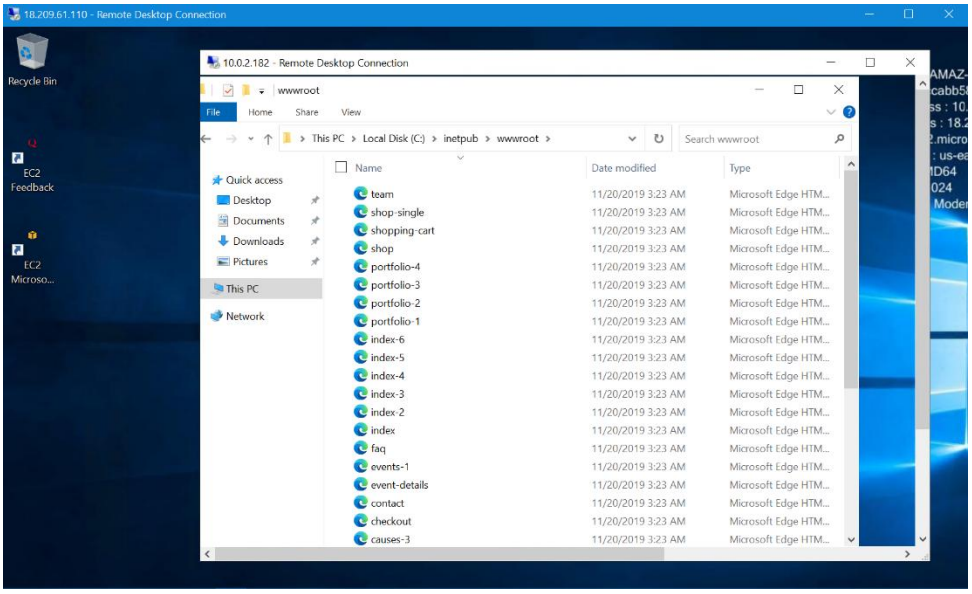


STEP 18

In, Remote Desktop Connection

After Successfully installed **IIS webserver in Private server.**

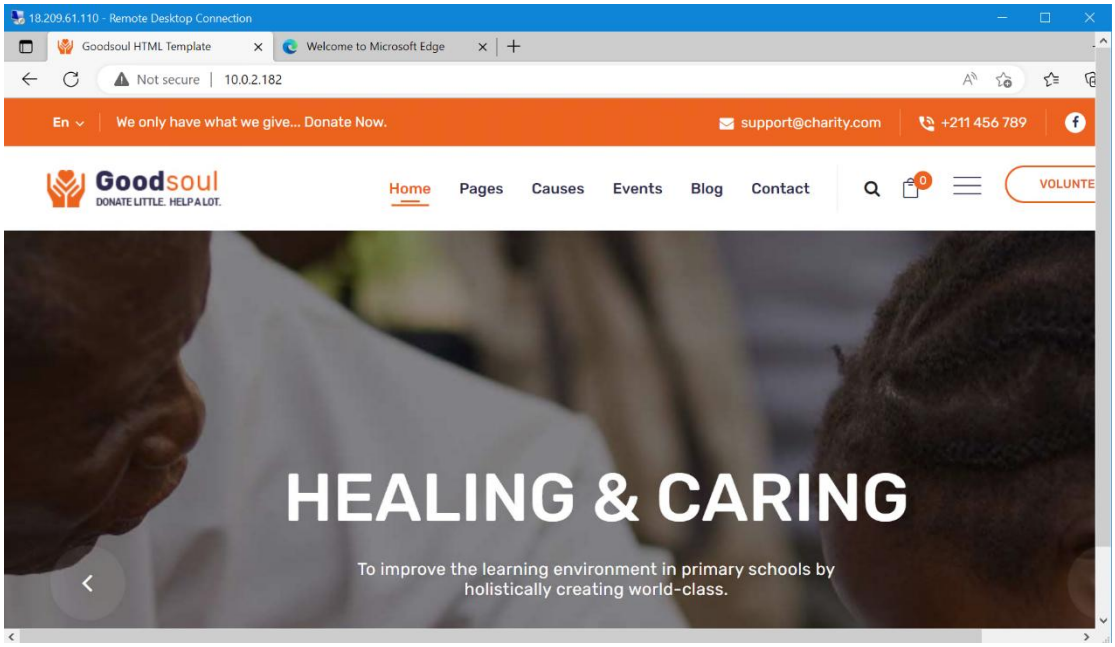
Open filebrowser - Localdisk C - inetpub - [**www.root**] - remove default files - paste website content files.



STEP 19

Open Chrome in Public Server.

Paste Private server Private IPv4 address as (http://10.0.2.182)

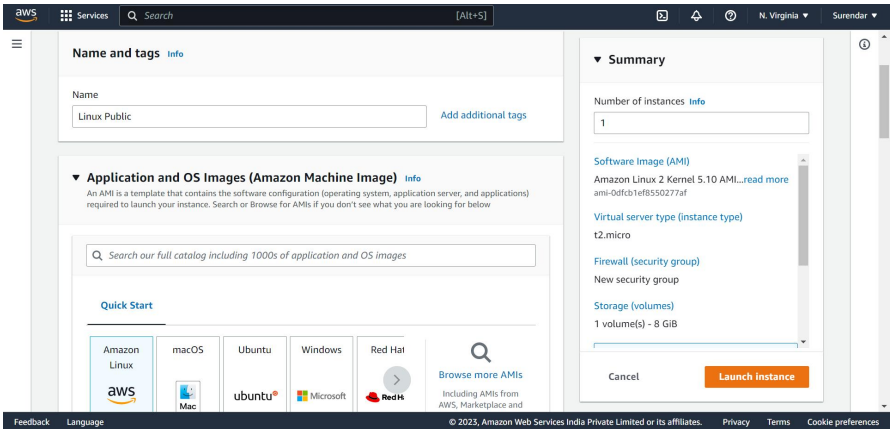


Successfully Hosted the Private Server Website through Public Server using RDP.

STEPS to Host a Website through Public Linux Server in Amazon Linux after successfully created and configured Your VPC.

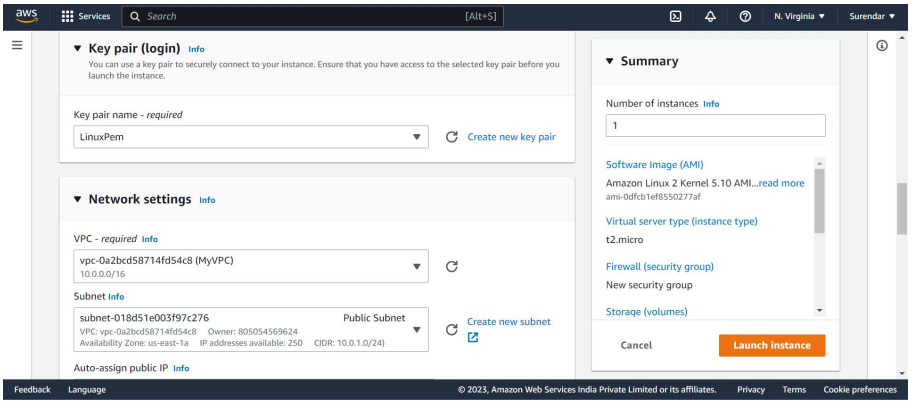
STEP 1

Launch instance - Set a name - choose Amazon Linux in AMI.



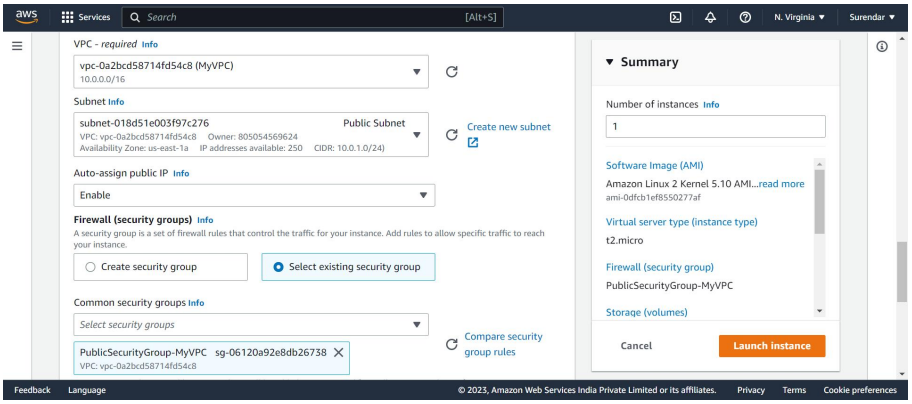
STEP 2

Create a new linuxpem key pair file (NOTE: create .ppk keyfile also if connecting the linux server with putty / winscp)



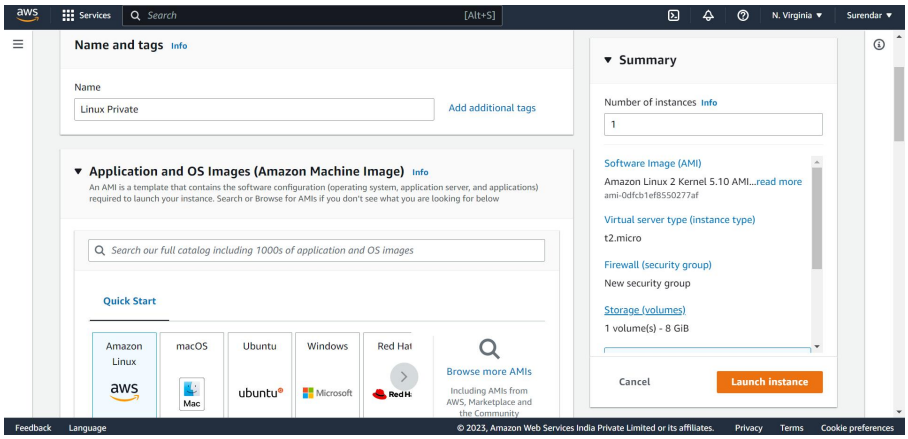
STEP 3

In Network Setting (edit) - choose MYVPC - Subnet (PublicSubnet) - Auto-Assign (Enable) - choose Existing PublicSecuritygroup-MyVPC - Launch Instance.



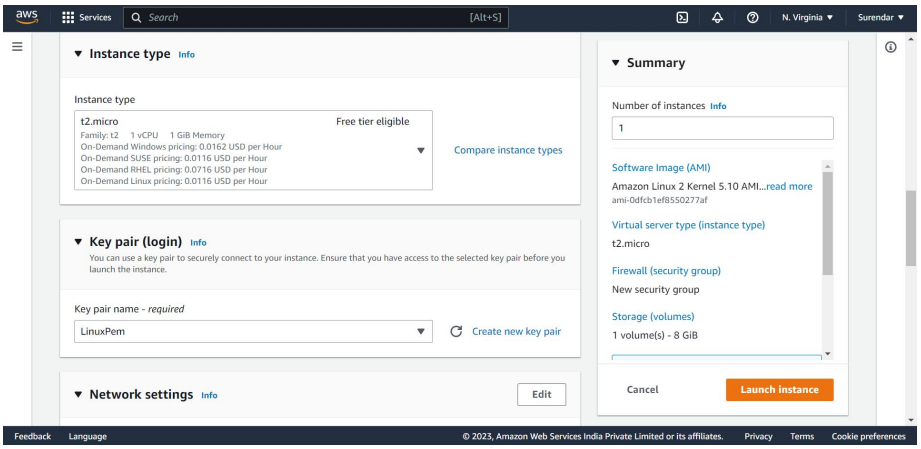
STEP 4

Launch instance for Private Linux Server - Choose Amazon Linux in AMI.



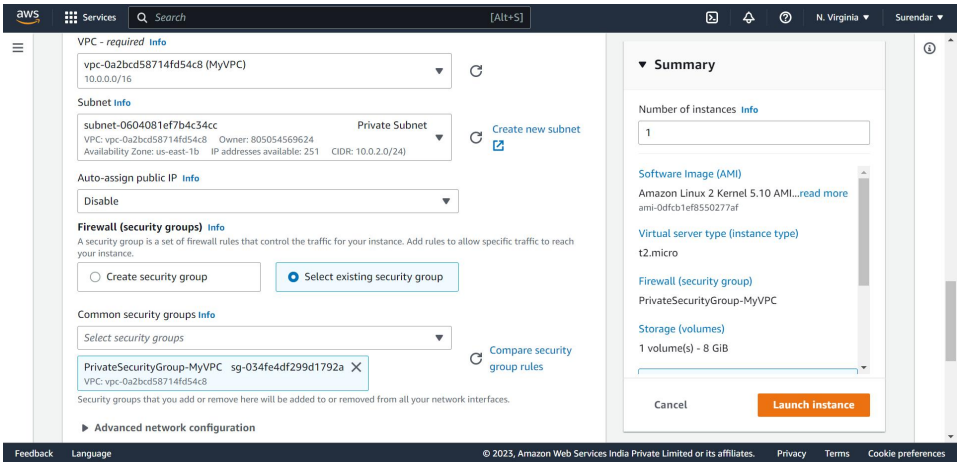
STEP 5

Choose same pem keyfile which have been chosen for Linux Public server.



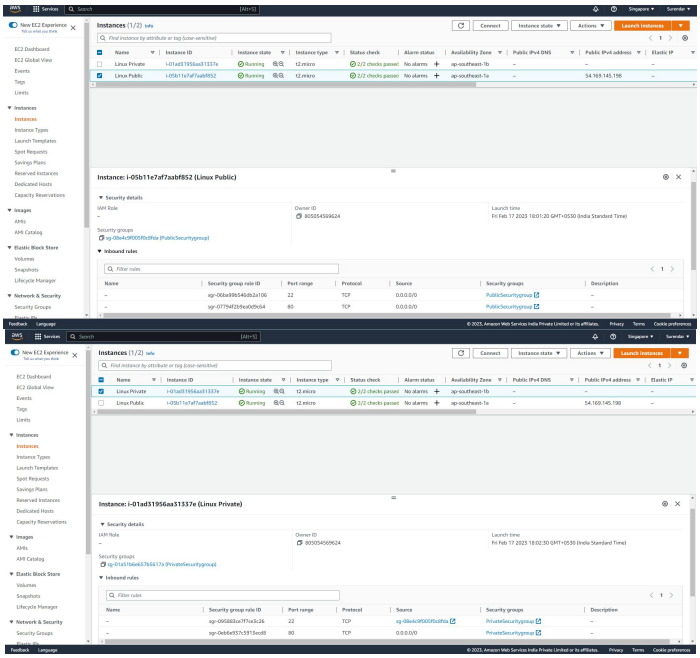
STEP 6

In network Setting (edit) - choose MyVPC - choose Private Subnet - Auto assign Public IP (Disable) - Choose Existing PrivateSecurityGroup-MyVPC - Launch instance.



STEP 7

Ensure both Security group in inbound rule, is the SSH port and HTTP port are added.



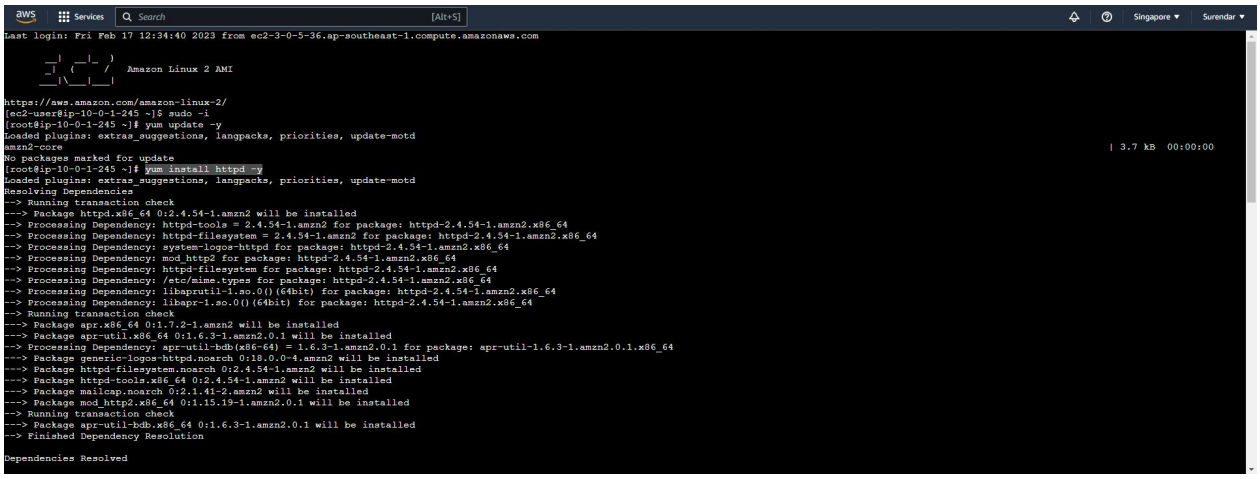
STEP 8

After connected the Public Linux server - Follow below commands steps and install httpd service.

$sudo -i

#yum update -y

#yum install httpd -y



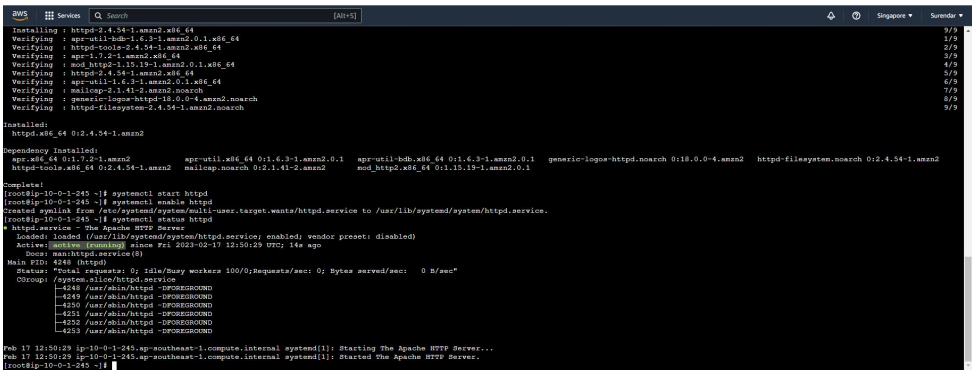
STEP 9

After installed the httpd service - start the httpd service and check the status (Active Running) (Enabled)

#systemctl start httpd

#systemctl enable httpd

#systemctl status httpd



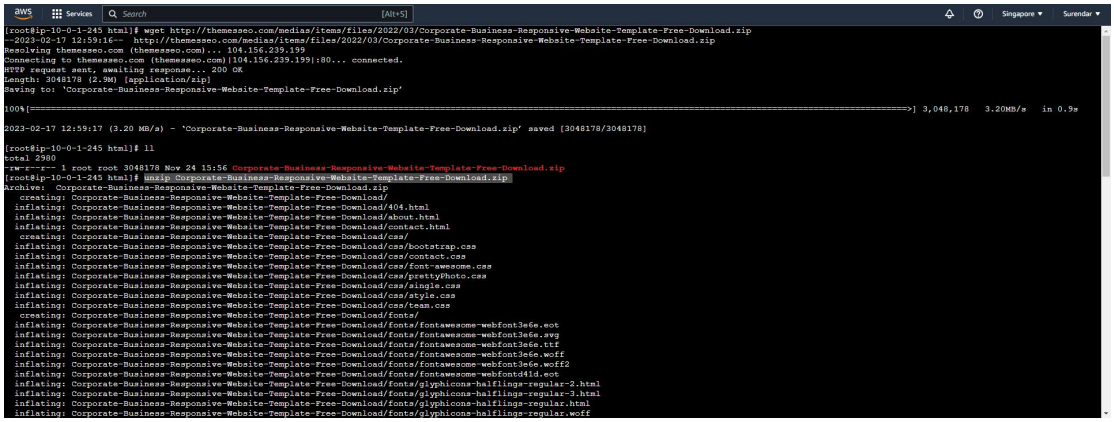
STEP 10

Download the Application website template in linux server using wget command in /var/www/html path

#cd /var/www/html

#wget <website .zip URL>

#unzip <website zip filename>



STEP 11

After unzipped the website template - move all the website content inside website template dir to /var/www/html path

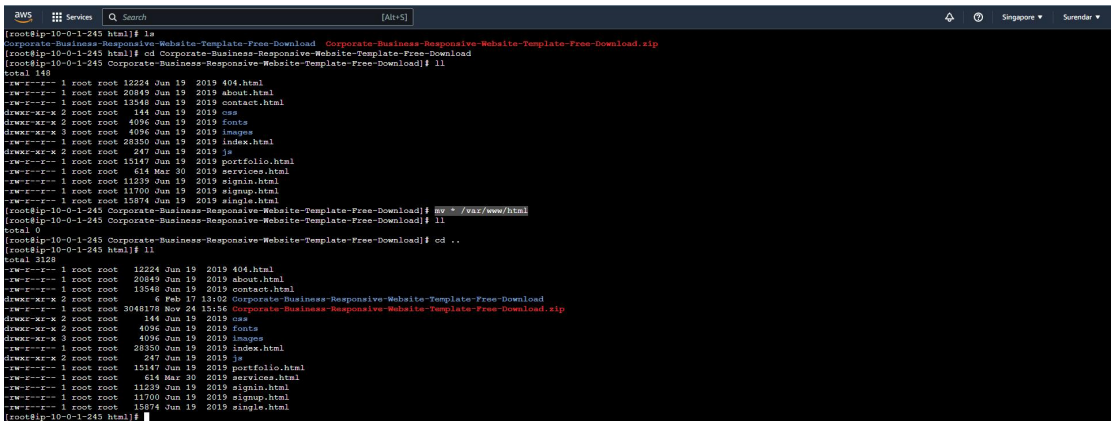
#cd <website template directory name>

#mv * /var/www/html

#cd ..

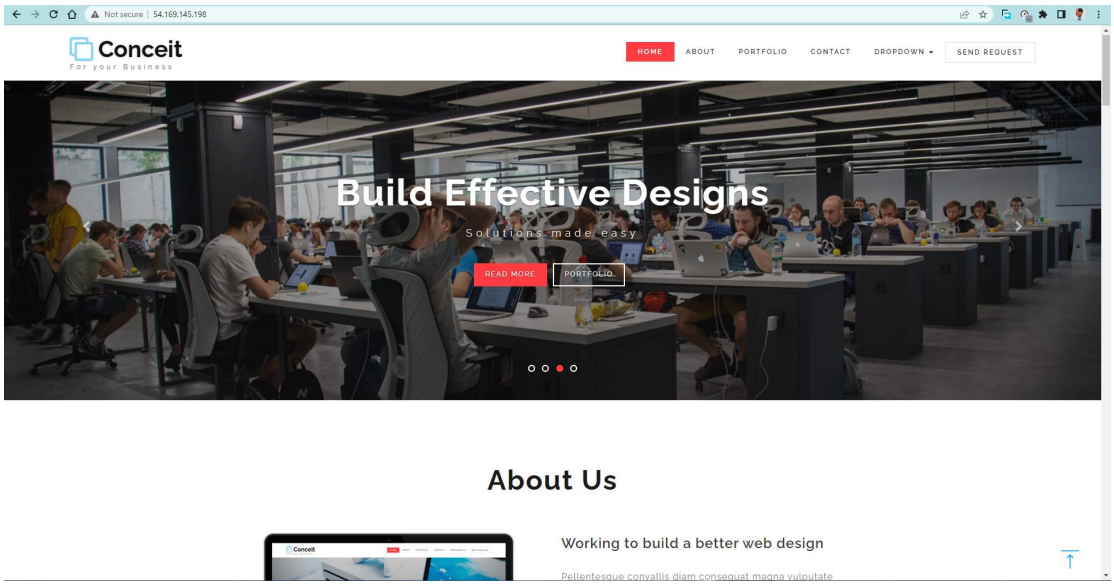
#ls

All website content inside website template dir are moved into /var/www/html directory.



STEP 12

Copy Ipv4 Public IP for linux public server and paste in chrome tab as http://54.169.145.198

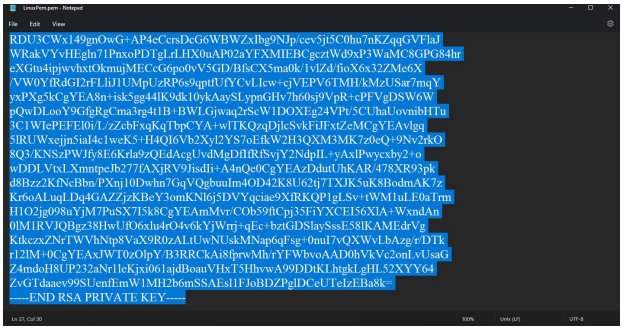


Successfully hosted application Website through Linux Public server in Amazon Linux.

Steps To login a private Linux server in Public linux server and Ensuring Private Linux Server is Pinging after adding a website content files in /var/www/html path.

STEP 13

Open the Linuxpem .pem keyfile in notepad and copy all the content



STEP 14

In LinuxPublic Server

#cd ~

#vi pemfile

<paste all the .pem keyfile content >

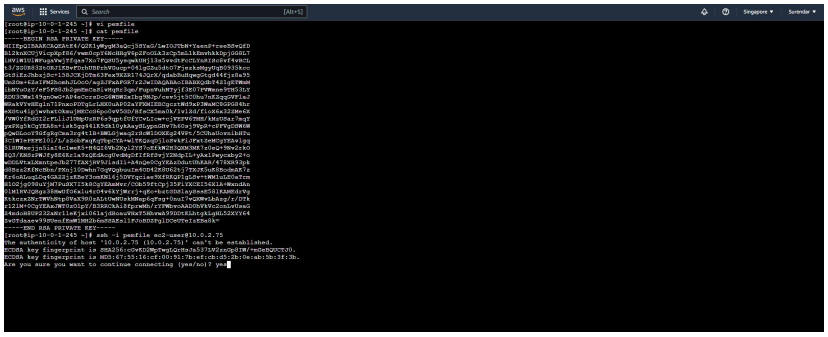
:wq!

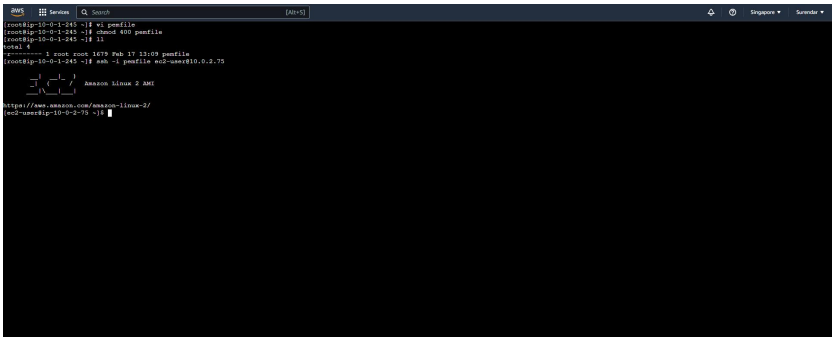
#chmod 400 pemfile

Now Login private linux server inside public Linux Server using ssh command

#ssh -i pemfile ec2-user@10.0.2.75

Yes





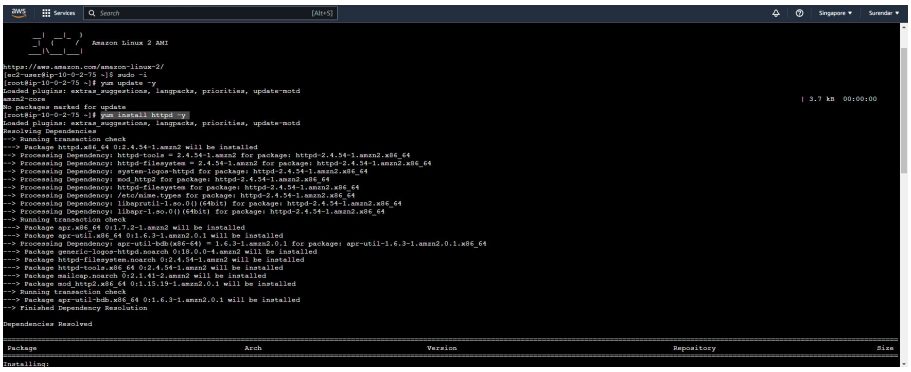
Successfully login private linux server inside public linux server using ssh command.

STEP 15

$sudo -i

#yum update -y

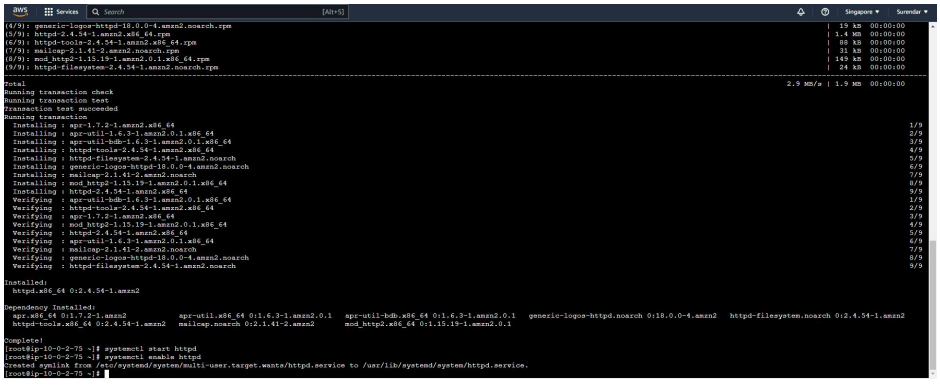
#yum install httpd -y



STEP 16

#systemctl start httpd

#systemctl enable httpd

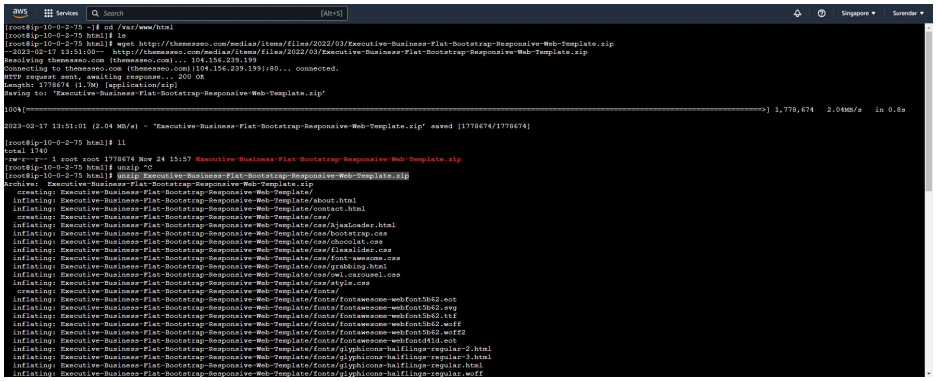


STEP 17

#cd /var/www/html

#wget <website .zip URL>

#unzip <website zip filename>



STEP 18

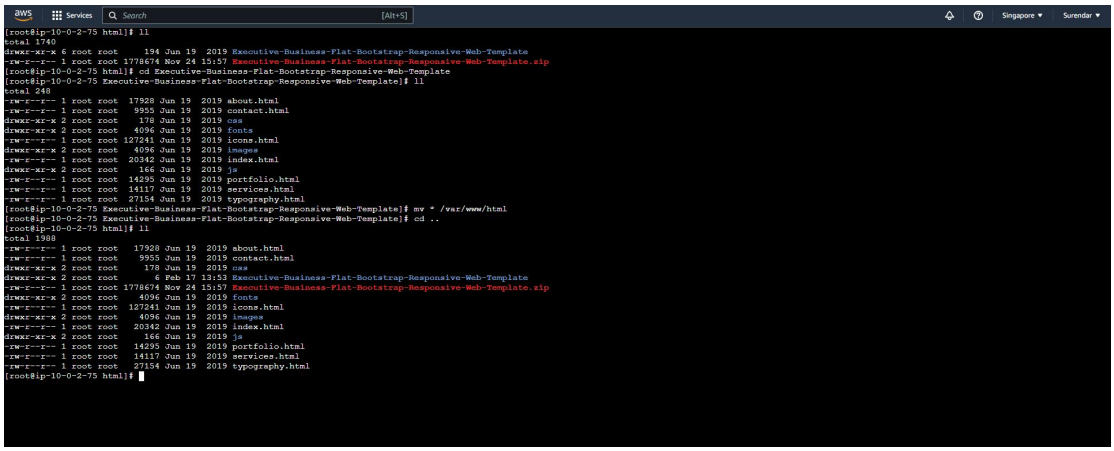
#cd <website template directory name>

#mv * /var/www/html

#cd ..

#ls

All website content inside website template dir are moved into /var/www/html directory.



STEP 19

Ping Linux Private server private IPv4 using ping command

#ping 10.0.2.75