DOCUMENTATION OF AWS PROJECT

Task Allocated by : Satyanarayana Sir

Mentoring & Monitoring : Teja Sir

Team Lead : B.Devi Veera Naga Sri

Team :

Ashok .P

Surya Kumari .M

Dhana Ganesh. N

Dhana Satya Sriya. N

Devi Veera Naga Sri. B

Platform Used : AMAZON WEB SERVICES

Services used :

- Elastic Compute Cloud (EC2)
- Virtual Private Cloud (VPC)
- Subnets
- Route tables
- Internet gateway
- Peering connections
- Security groups

Abstract

AWS users who need to access services or resources within a private network typically connect to the bastion host and VPC peering . Though both functionality is similar , there is a slight difference . The primary purpose of a bastion host is to serve as a secure gateway for remote access to instances or resources within a single VPC. It acts as a controlled entry point for users or administrators to connect to specific resources within the same VPC. Bastion hosts are used for secure remote management and access control . While the VPC peering is to establish network-level connectivity between two or more Virtual Private Clouds (VPCs). It enables resources in one VPC to communicate directly with resources in another VPC, essentially extending the network. VPC peering is used for inter-VPC communication and resource sharing. In essence, VPC peering is about connecting separate VPCs for inter-VPC communication, while a bastion host is focused on providing secure remote access to resources within a single VPC. Their roles and functions are fundamentally different, addressing distinct networking and security needs.

Elastic Compute Cloud (EC2) :

Amazon Elastic Compute Cloud (Amazon EC2) provides on-demand, scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 reduces hardware costs so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. You can add capacity (scale up) to handle compute-heavy tasks, such as monthly or yearly processes, or spikes in website traffic. When usage decreases, you can reduce capacity (scale down) again.Amazon EC2 provides different instance types to enable you to choose the CPU, memory, storage, and networking capacity that you need to run your applications.

Benfits of EC2 instance :

- ➢ Amazon Elastic Compute Cloud (Amazon EC2) provides secure, resizable compute capacity in the cloud.
- ➢ Access reliable, scalable infrastructure on demand.
- ➢ Optimize performance and cost with flexible options.

Steps to launch instance :

- Select a region
- Navigate to the EC2 console
- Create the EC2 instance
- Name instance
- Choose an Amazon machine image
- Choose an instance type
- Create key pair
- Edit network settings
- Configure storage
- Launch instance

Virtual Private Cloud (VPC) :

Virtual Private Cloud (Amazon VPC), you can launch AWS resources in a logically isolated virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS. You can provide multiple layers of security, including security groups and network access control lists, to help control access to Amazon EC2 instances in each subnet. For example, you can create a public-facing subnet for web servers that can access to the internet and can also place your backend system such as databases or application servers to a private-facing subnet.

Steps to Create a VPC:

1. Login to your AWS Console.
2. Create your VPC with Valid CIDR and name.
3. Click Subnet and create your subnet with public subnet valid name & VPC.
4. Valid subnet range which is valid IPv4 CIDR block.
5. Repeat steps 2 & 3, with Private Subnet too.

What can we do with a VPC:

- ➢ Launch instances in a subnet of your choosing. We can choose our own subnet addressing.
- ➢ We can assign custom IP address ranges in each subnet.
- ➢ We can configure route tables between subnets.
- ➢ We can create an internet gateway and attach it to our VPC.
- ➢ It provides much better security control over your AWS resources.
- ➢ We can assign security groups to individual instances.
- ➢ We also have subnet network access control lists (ACLS).

Subnetwork or subnet :

Subnetwork or subnet is a logical subdivision of an IP network. The practice of dividing a network into two or more networks is called subnetting. AWS provides two types of subnetting one is Public which allow the internet to access the machine and another is private which is hidden from the internet. A subnet is a range of IP addresses in your VPC. You can create AWS resources, such as EC2 instances, in specific subnets.

Route tables:

A route table contains a set of rules, called routes, that are used to determine where network traffic from your VPC is directed. You can explicitly associate a subnet with a particular route table. Otherwise, the subnet is implicitly associated with the main route table. Each route in a route table specifies the range of IP addresses where you want the traffic to go (the destination) and the gateway, network interface, or connection through which to send the traffic (the target).

Internet Gateway(IGW):

Internet Gateway is a horizontally scalable, redundant, and highly available VPC component that enables the communication between instances in a VPC and the Internet. It acts as a bridge between a VPC and the Internet and allows you to route Internet traffic to your instances. Internet Gateways also provide security features such as security groups and network access control lists to control inbound and outbound traffic, as well as a way to connect to on-premises resources via VPN or Direct Connect.

Security Groups:

An AWS security group acts as a virtual firewall for your EC2 instances to control incoming and outgoing traffic. Both inbound and outbound rules control the flow of traffic to and traffic from your instance, respectively. AWS Security Groups help you secure your cloud environment by controlling how traffic will be allowed into your EC2 machines. With Security Groups, you can ensure that all the traffic that flows at the instance level is only through your established ports and protocols.

VPC Peering Connection:

VPC Peering connection is a networking connection between two VPCs that enables you to route traffic between them privately (using private IPv4 or IPv6 addresses). Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, with a VPC in another AWS account or with a VPC in a different AWS Region. VPC peering connection is a one to one relationship between two VPCs. You can create multiple VPC peering connections for each VPC, but transitive peering relationships are not supported. You can modify a VPC peering connection to enable instances in their VPC to communicate with linked EC2-Classic instances in the peer VPC.AWS uses the existing infrastructure of a VPC to create a VPC peering connection; it is neither a gateway nor a VPN connection, and does not rely on a separate piece of physical hardware.

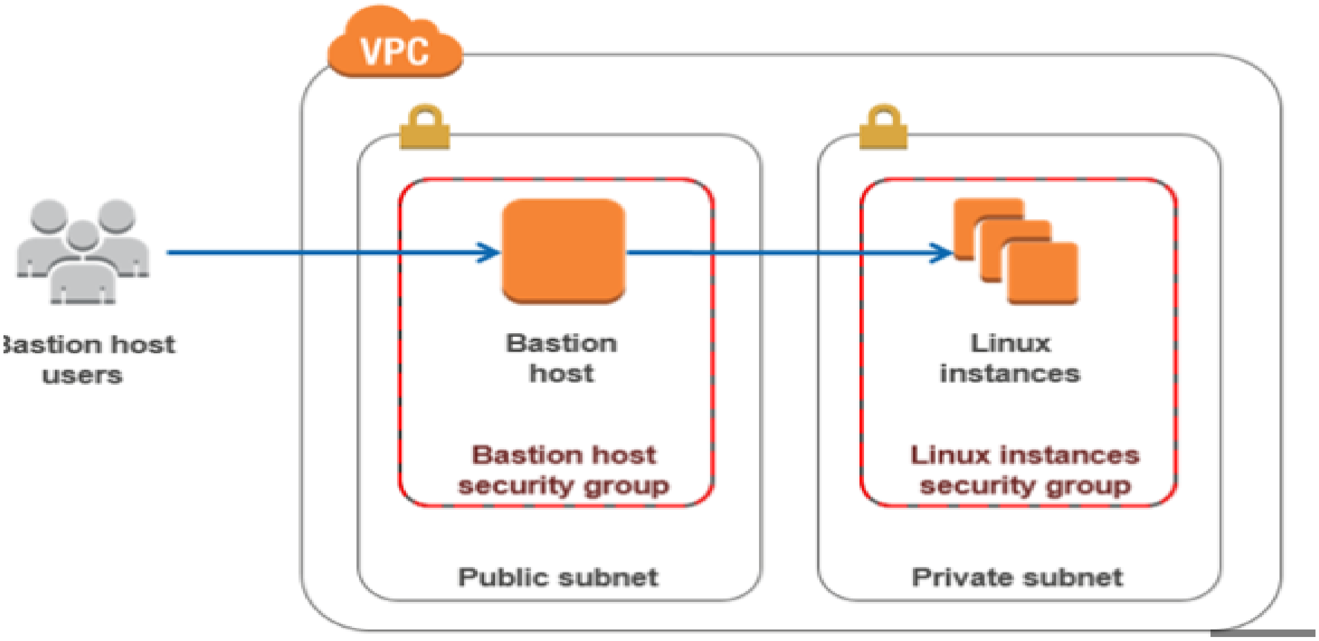
Rules:

- VPC peering connection cannot be created between VPCs that have matching or overlapping CIDR blocks.

- VPC peering does not support transitive peering relationships. If A is peered with B, B is peered with C, A is not peered with C.

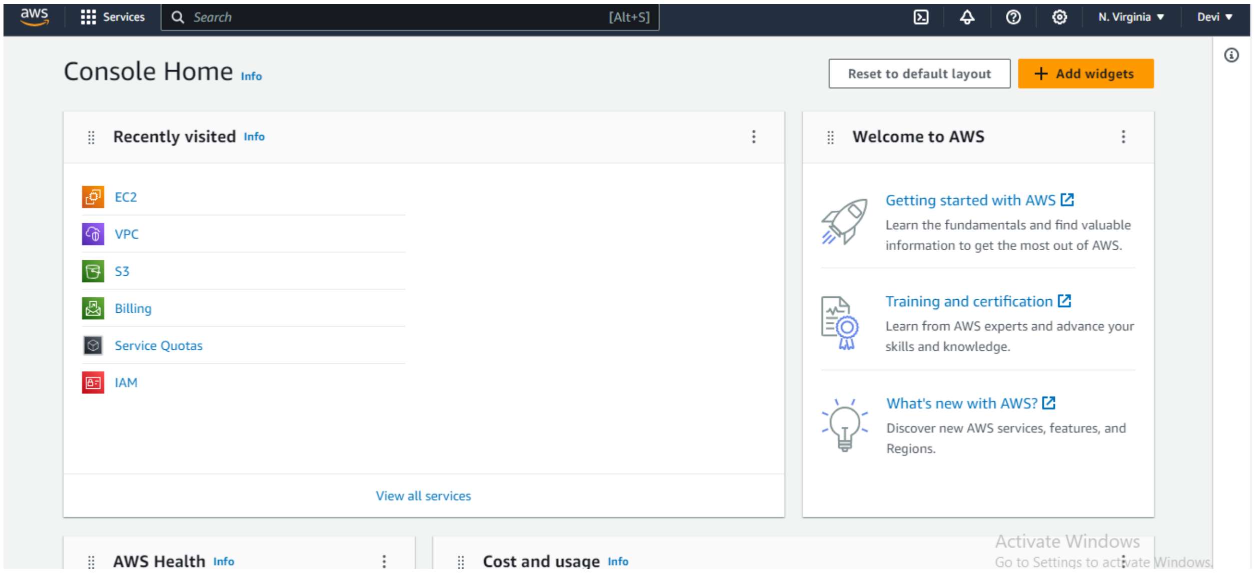
BASTION HOST

A bastion host is a server used to manage / access to a private network from a public network. Sometimes called a jump box or jump server. Bastion host basically provides an entry point into the private networks which are to be connected to the external network securing from the attacks. A bastion host has both internal and external IP addresses. If users want to connect the internal instance without using external IP addresses then it can connect to a Bastion host and then connect to your internal instances from that Bastion host. While using Bastion service you have to log in first to your Bastion host and then directed to the private instances**.**



Steps Followed:

- Log in to AWS Management Console.



Getting started with AWS

Learn the fundamentals and find valuable information to get the most out of AWS.

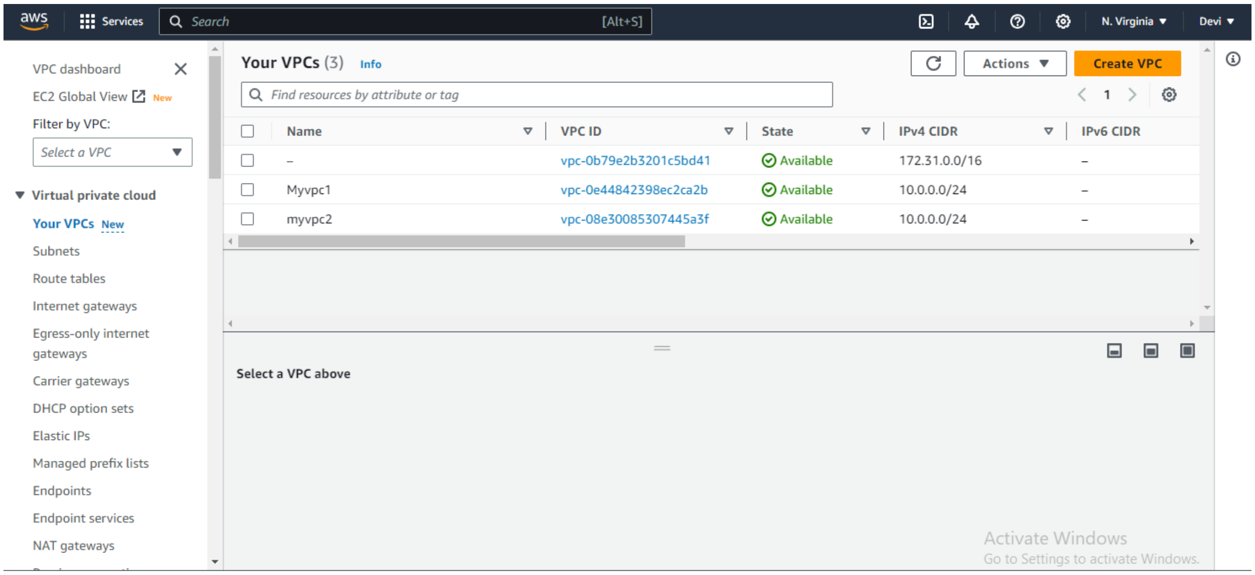
Training and certification

Learn from AWS experts and advance your skills and knowledge.

What's new with AWS?

Discover new AWS services, features, and Regions.

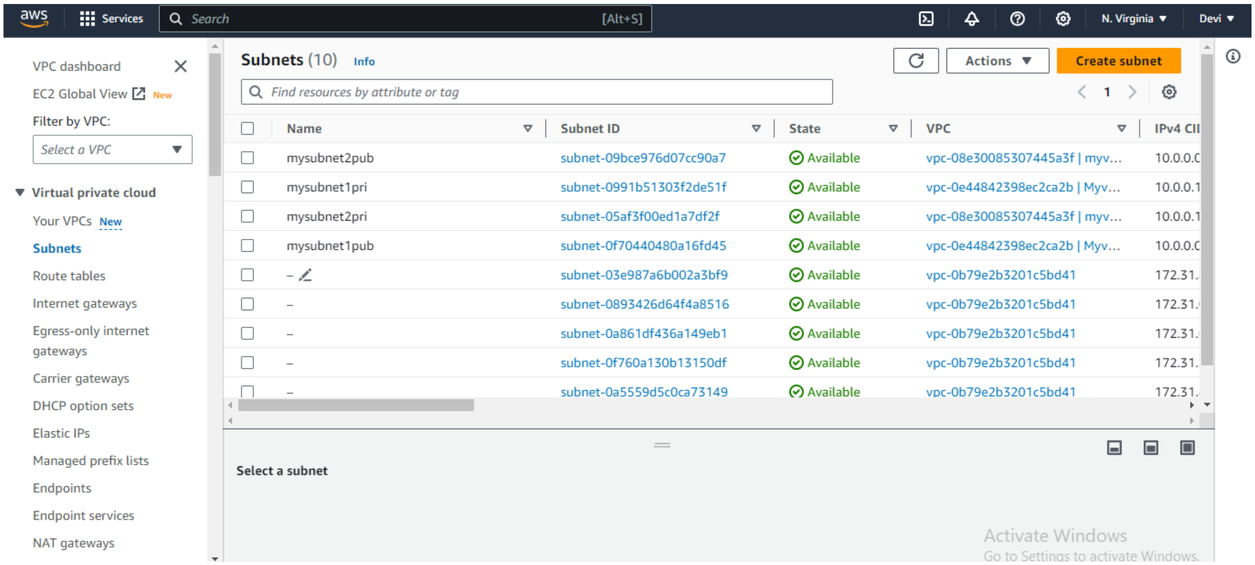
- Create VPC with required CIDR range.



Your VPCs (3) Info

□	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
□	–	vpc-0b79e2b3201c5bd41	✓ Available	172.31.0.0/16	–
□	Myvpc1	vpc-0e44842398ec2ca2b	✓ Available	10.0.0.0/24	–
□	myvpc2	vpc-08e30085307445a3f	✓ Available	10.0.0.0/24	–

- Divide the VPC into subnets – Private and Public subnets

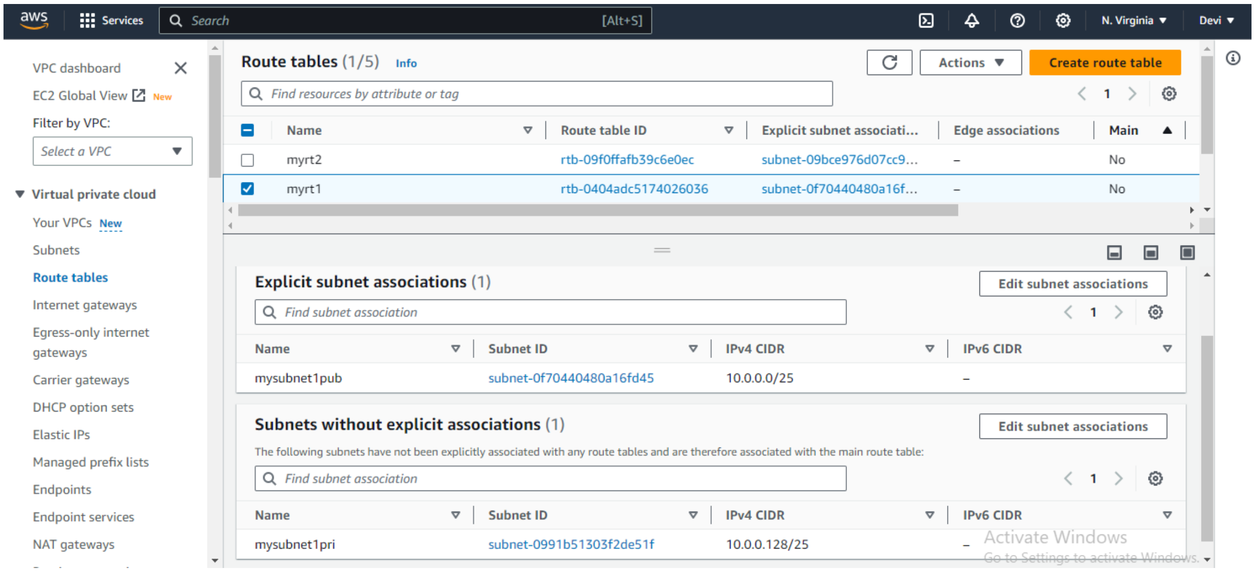


Subnets (10) Info

□	Name	Subnet ID	State	VPC	IPv4 CI
□	mysubnet2pub	subnet-09bce976d07cc90a7	Available	vpc-08e30085307445a3f | myv...	10.0.0.0
□	mysubnet1pri	subnet-0991b51303f2de51f	Available	vpc-0e44842398ec2ca2b | Myv...	10.0.0.1
□	mysubnet2pri	subnet-05af3f00ed1a7df2f	Available	vpc-08e30085307445a3f | myv...	10.0.0.1
□	mysubnet1pub	subnet-0f70440480a16fd45	Available	vpc-0e44842398ec2ca2b | Myv...	10.0.0.0
□	–	subnet-03e987a6b002a3bf9	Available	vpc-0b79e2b3201c5bd41	172.31.
□	–	subnet-0893426d64f4a8516	Available	vpc-0b79e2b3201c5bd41	172.31.
□	–	subnet-0a861df436a149eb1	Available	vpc-0b79e2b3201c5bd41	172.31.
□	–	subnet-0f760a130b13150df	Available	vpc-0b79e2b3201c5bd41	172.31.
□	–	subnet-0a5559d5c0ca73149	Available	vpc-0b79e2b3201c5bd41	172.31.

Select a subnet

- Create Route tables and attach to corresponding subnets.



Route tables (1/5) Info

☐	Name	Route table ID	Explicit subnet associati...	Edge associations	Main
□	myrt2	rtb-09f0ffafb39c6e0ec	subnet-09bce976d07cc9...	–	No
☑	myrt1	rtb-0404adc5174026036	subnet-0f70440480a16f...	–	No

Explicit subnet associations (1)

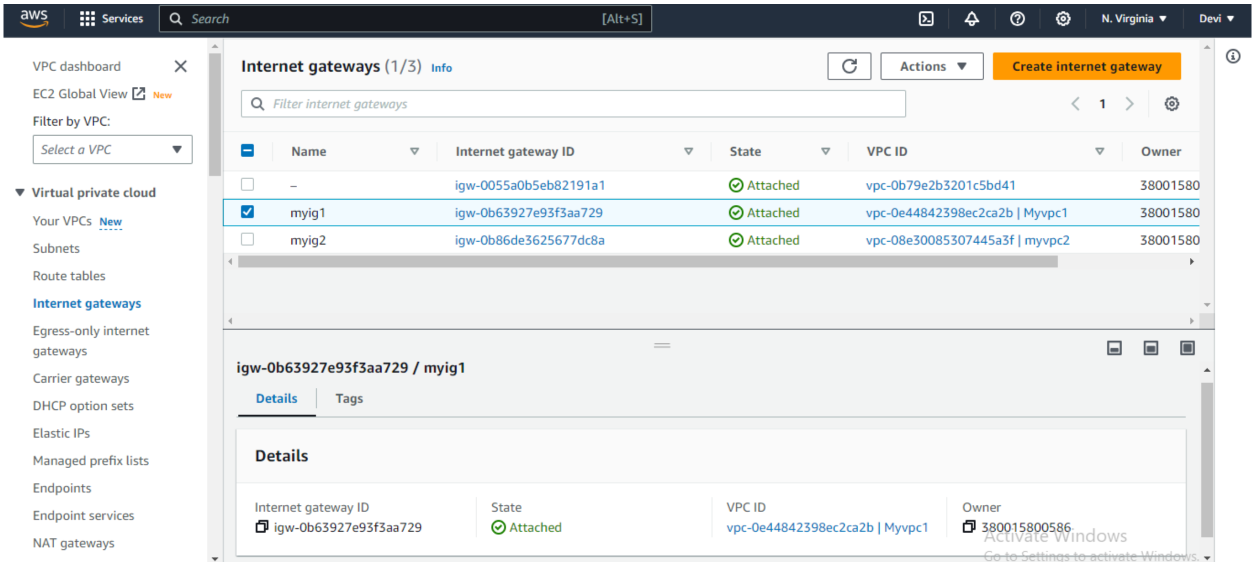
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
mysubnet1pub	subnet-0f70440480a16fd45	10.0.0.0/25	–

Subnets without explicit associations (1)

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
mysubnet1pri	subnet-0991b51303f2de51f	10.0.0.128/25	–

- Create Internet gateway and attach to the VPC.



Internet gateways (1/3) Info

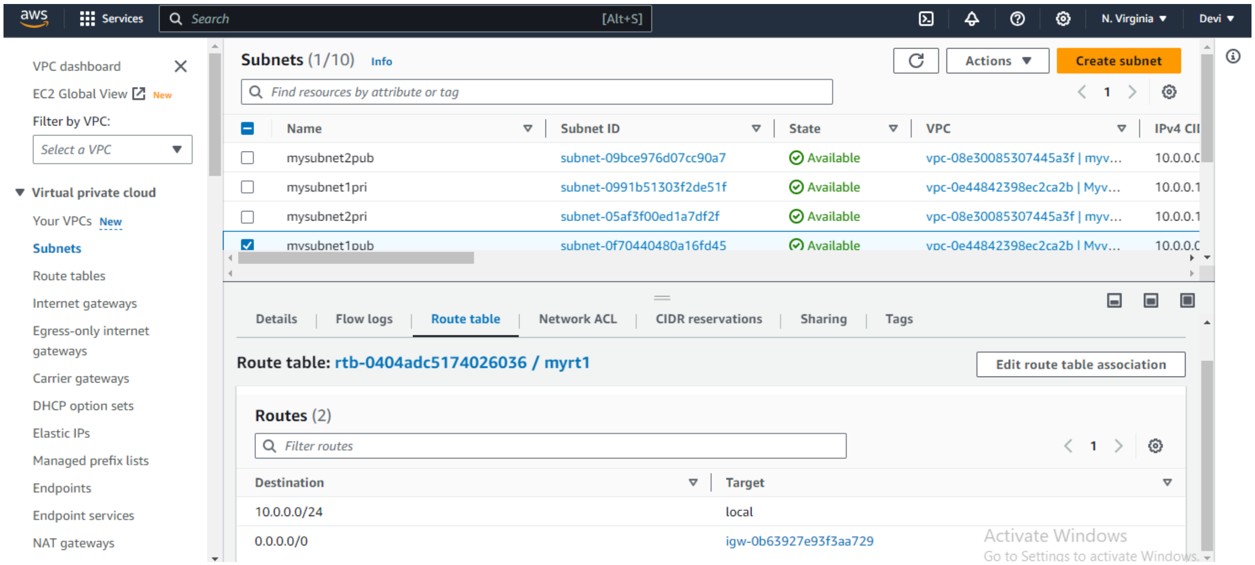
	Name	Internet gateway ID	State	VPC ID	Owner
□	–	igw-0055a0b5eb82191a1	Attached	vpc-0b79e2b3201c5bd41	38001580
☑	myig1	igw-0b63927e93f3aa729	Attached	vpc-0e44842398ec2ca2b | Myvpc1	38001580
□	myig2	igw-0b86de3625677dc8a	Attached	vpc-08e30085307445a3f | myvpc2	38001580

igw-0b63927e93f3aa729 / myig1

Details

Internet gateway ID	State	VPC ID	Owner
igw-0b63927e93f3aa729	Attached	vpc-0e44842398ec2ca2b | Myvpc1	380015800586

- Attach Internet gateway to the public subnet inorder to provide internet to the public subnets.



Subnets (1/10) Info

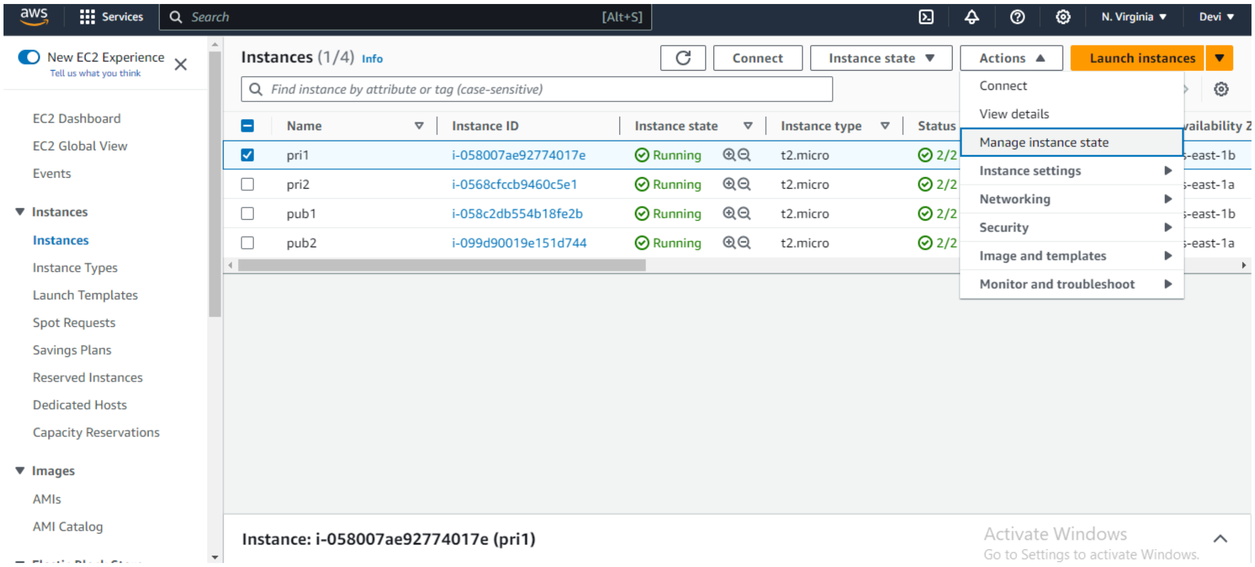
	Name	Subnet ID	State	VPC	IPv4 CII
□	mysubnet2pub	subnet-09bce976d07cc90a7	Available	vpc-08e30085307445a3f | myv...	10.0.0.C
□	mysubnet1pri	subnet-0991b51303f2de51f	Available	vpc-0e44842398ec2ca2b | Myv...	10.0.0.1
□	mysubnet2pri	subnet-05af3f00ed1a7df2f	Available	vpc-08e30085307445a3f | myv...	10.0.0.1
☑	mysubnet1pub	subnet-0f70440480a16fd45	Available	vpc-0e44842398ec2ca2b | Myv...	10.0.0.C

Route table: rtb-0404adc5174026036 / myrt1

Routes (2)

Destination	Target
10.0.0.0/24	local
0.0.0.0/0	igw-0b63927e93f3aa729

- Launch EC2 instance in each subnet. Select OS, create key pair, edit network settings. Enable associate IP address for public network and disable for private network.



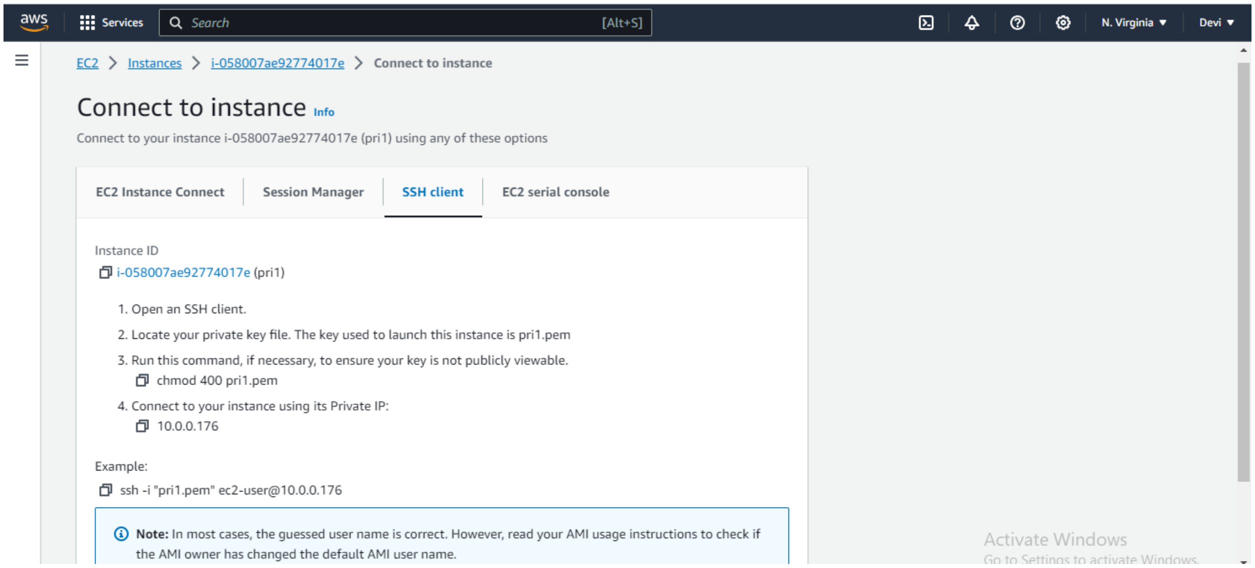
Instances (1/4) Info

Find instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type	Status	vailability Z
☑	pri1	i-058007ae92774017e	✓ Running	t2.micro	✓ 2/2	-east-1b
□	pri2	i-0568cfccb9460c5e1	✓ Running	t2.micro	✓ 2/2	-east-1a
□	pub1	i-058c2db554b18fe2b	✓ Running	t2.micro	✓ 2/2	-east-1b
□	pub2	i-099d90019e151d744	✓ Running	t2.micro	✓ 2/2	-east-1a

Instance: i-058007ae92774017e (pri1)

- Now connect to the instance.



Connect to instance Info

Connect to your instance i-058007ae92774017e (pri1) using any of these options

Instance ID

i-058007ae92774017e (pri1)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is pri1.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.

```
chmod 400 pri1.pem
```

4. Connect to your instance using its Private IP:

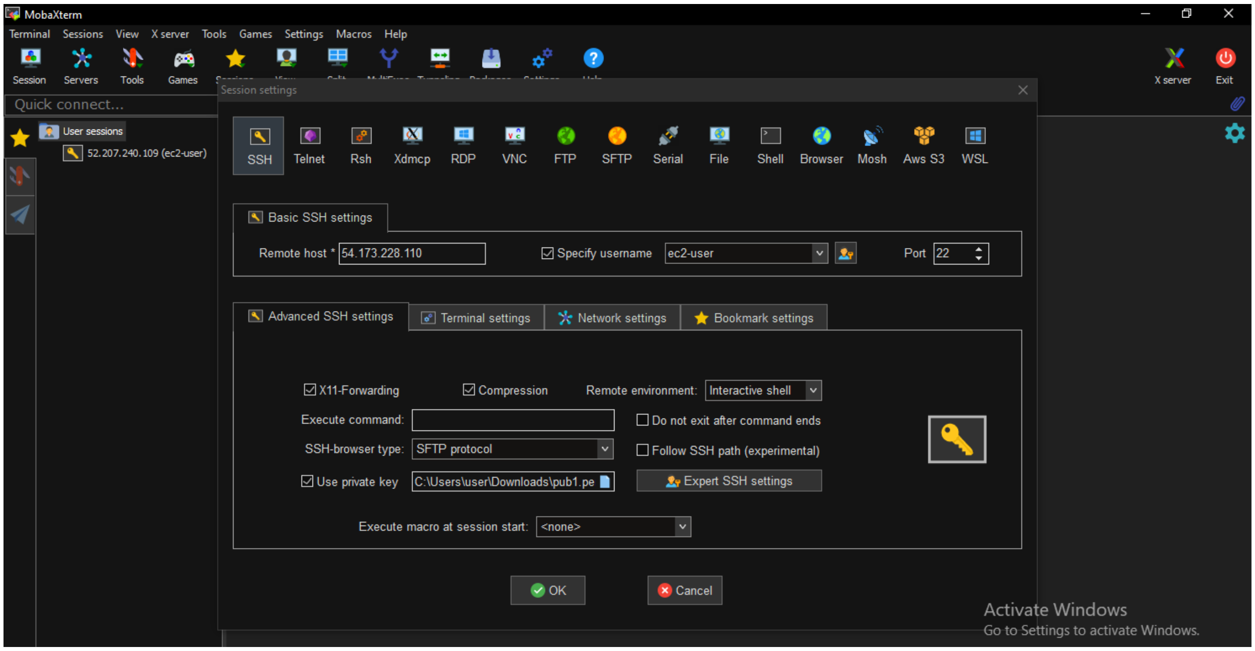
```
10.0.0.176
```

Example:

```
ssh -i "pri1.pem" ec2-user@10.0.0.176
```

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

- Here we are connecting through MobaXterm.
  - ➪ Select SSH session.
  - ➪ Remote host - our public IP
  - ➪ Username - ec2-user
  - ➪ Finally upload the public key pair in advanced settings.


MobaXterm
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games
X server Exit
Quick connect...
User sessions
52.207.240.109 (ec2-user)
Session settings
SSH Telnet Rsh Xdmcp RDP VNC FTP SFTP Serial File Shell Browser Mosh Aws S3 WSL
Basic SSH settings
Advanced SSH settings
Terminal settings
Network settings
Bookmark settings
Expert SSH settings
OK
Cancel
Activate Windows
Go to Settings to activate Windows.

Remote host * 54.173.228.110
☑ Specify username ec2-user
Port 22

☑ X11-Forwarding
☑ Compression
Remote environment: Interactive shell
Execute command:
☐ Do not exit after command ends
SSH-browser type: SFTP protocol
☐ Follow SSH path (experimental)
☑ Use private key C:\Users\user\Downloads\pub1.pe
Execute macro at session start: <none>

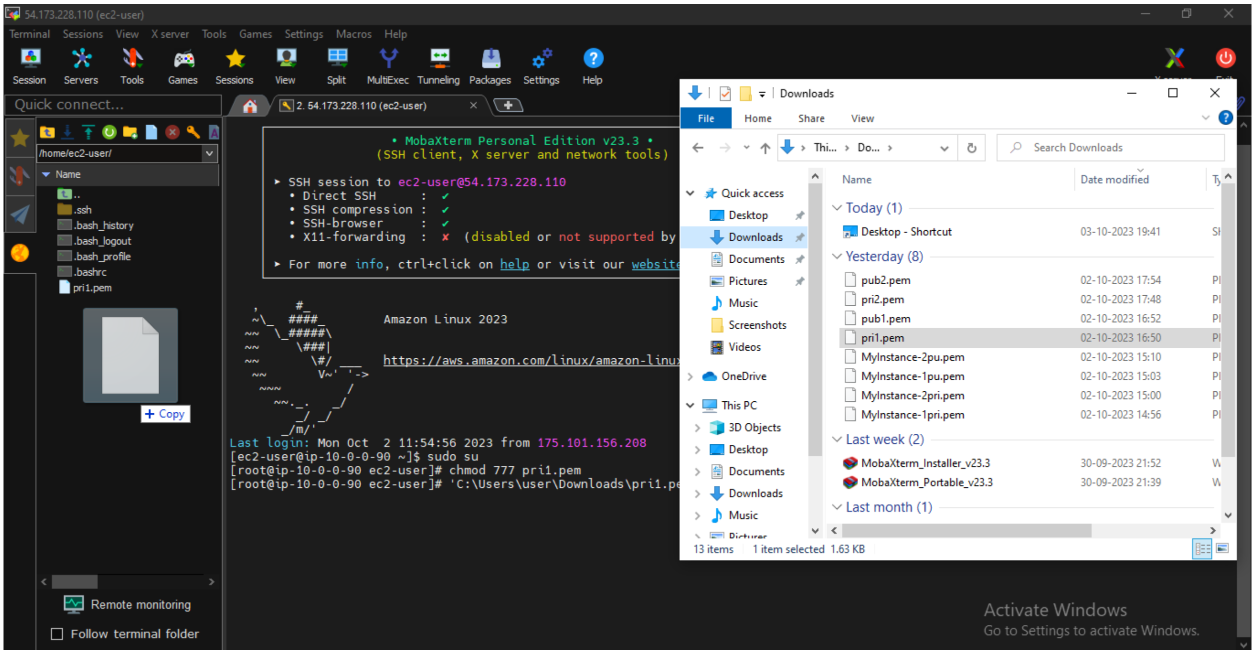
- We will connect to the public instance successfully. Now by using some commands we will connect private instance.

  - ➪ ***sudo su*** — Change to root account

  - ➪ ***chmod 777 privatekeypair.pem*** — Giving access to read, write, execute

  Drag the private key pair file to left side.

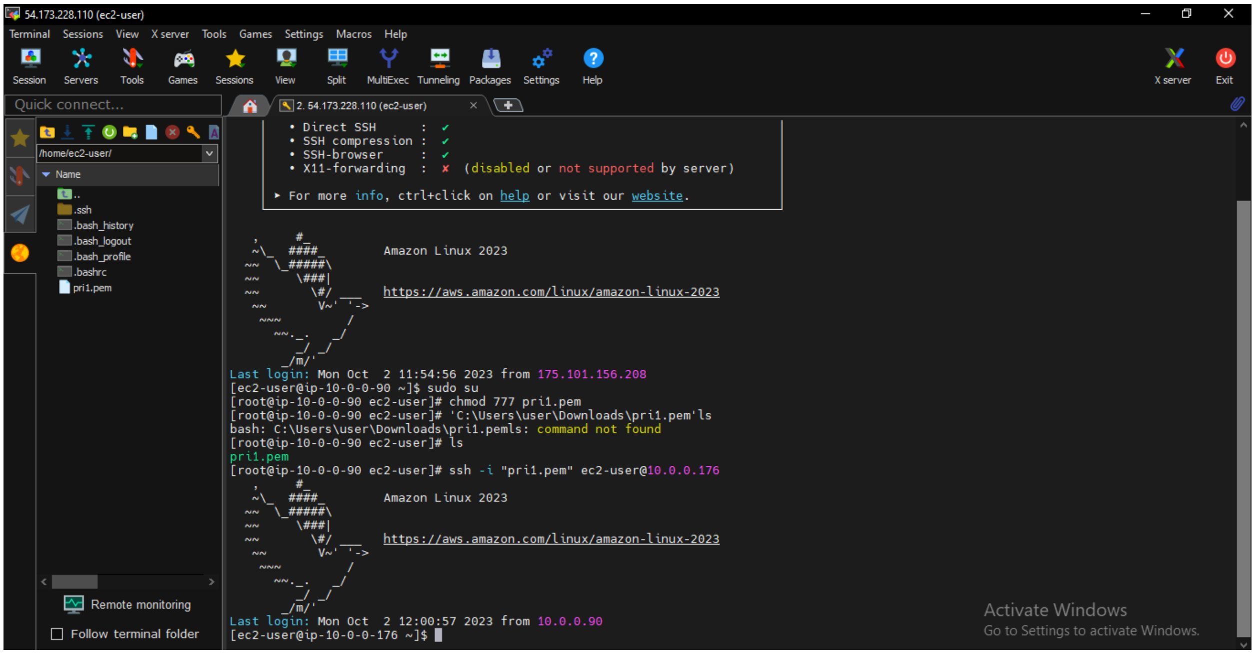
- ➪ ***ssh -i "privatekeypair.pem" ec2-user@privateip***



```
        • MobaXterm Personal Edition v23.3 •
      (SSH client, X server and network tools)

 ▸ SSH session to ec2-user@54.173.228.110
   • Direct SSH       :  ✓
   • SSH compression  :  ✓
   • SSH-browser      :  ✓
   • X11-forwarding   :  ✘  (disabled or not supported by

 ▸ For more info, ctrl+click on help or visit our website

   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Mon Oct  2 11:54:56 2023 from 175.101.156.208
[ec2-user@ip-10-0-0-90 ~]$ sudo su
[root@ip-10-0-0-90 ec2-user]# chmod 777 pri1.pem
[root@ip-10-0-0-90 ec2-user]# 'C:\Users\user\Downloads\pri1.pe
```

Name	Date modified
Today (1)	
Desktop - Shortcut	03-10-2023 19:41
Yesterday (8)	
pub2.pem	02-10-2023 17:54
pri2.pem	02-10-2023 17:48
pub1.pem	02-10-2023 16:52
pri1.pem	02-10-2023 16:50
MyInstance-2pu.pem	02-10-2023 15:10
MyInstance-1pu.pem	02-10-2023 15:03
MyInstance-2pri.pem	02-10-2023 15:00
MyInstance-1pri.pem	02-10-2023 14:56
Last week (2)	
MobaXterm_Installer_v23.3	30-09-2023 21:52
MobaXterm_Portable_v23.3	30-09-2023 21:39
Last month (1)	

- Finally, we connected the private instance through public instance

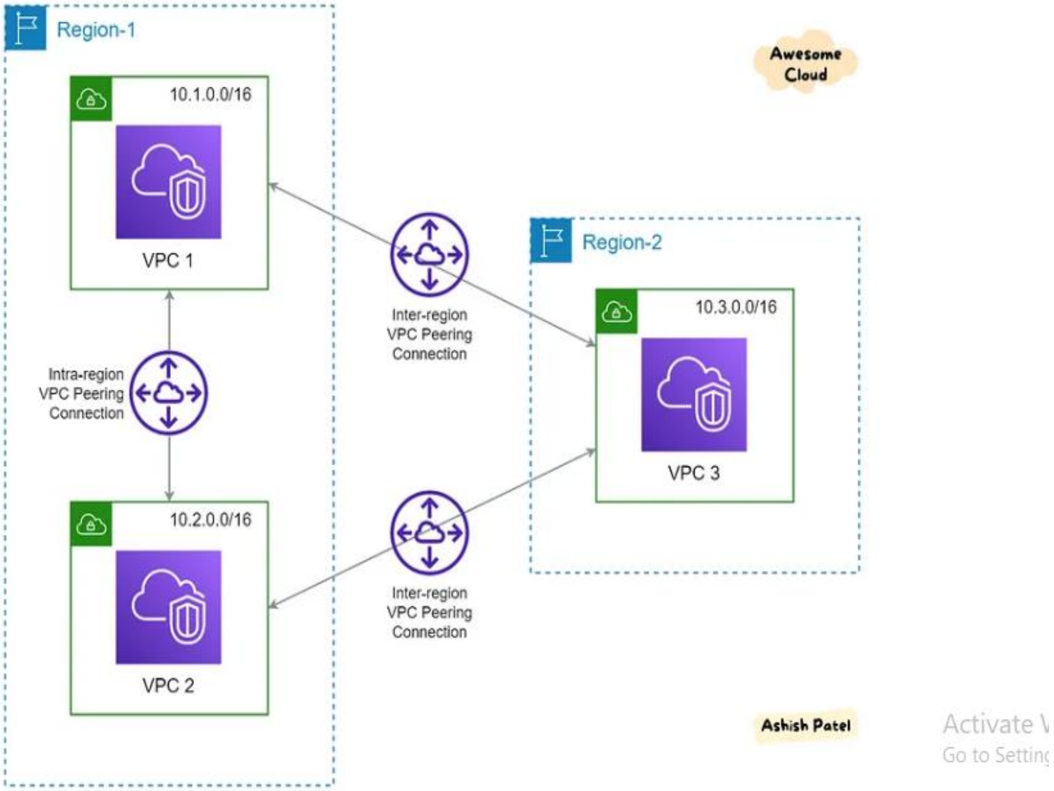


```
   • Direct SSH       :  ✓
   • SSH compression  :  ✓
   • SSH-browser      :  ✓
   • X11-forwarding   :  ✘  (disabled or not supported by server)

 ▸ For more info, ctrl+click on help or visit our website.

   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Mon Oct  2 11:54:56 2023 from 175.101.156.208
[ec2-user@ip-10-0-0-90 ~]$ sudo su
[root@ip-10-0-0-90 ec2-user]# chmod 777 pri1.pem
[root@ip-10-0-0-90 ec2-user]# 'C:\Users\user\Downloads\pri1.pem'ls
bash: C:\Users\user\Downloads\pri1.pemls: command not found
[root@ip-10-0-0-90 ec2-user]# ls
pri1.pem
[root@ip-10-0-0-90 ec2-user]# ssh -i "pri1.pem" ec2-user@10.0.0.176
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Mon Oct  2 12:00:57 2023 from 10.0.0.90
[ec2-user@ip-10-0-0-176 ~]$
```

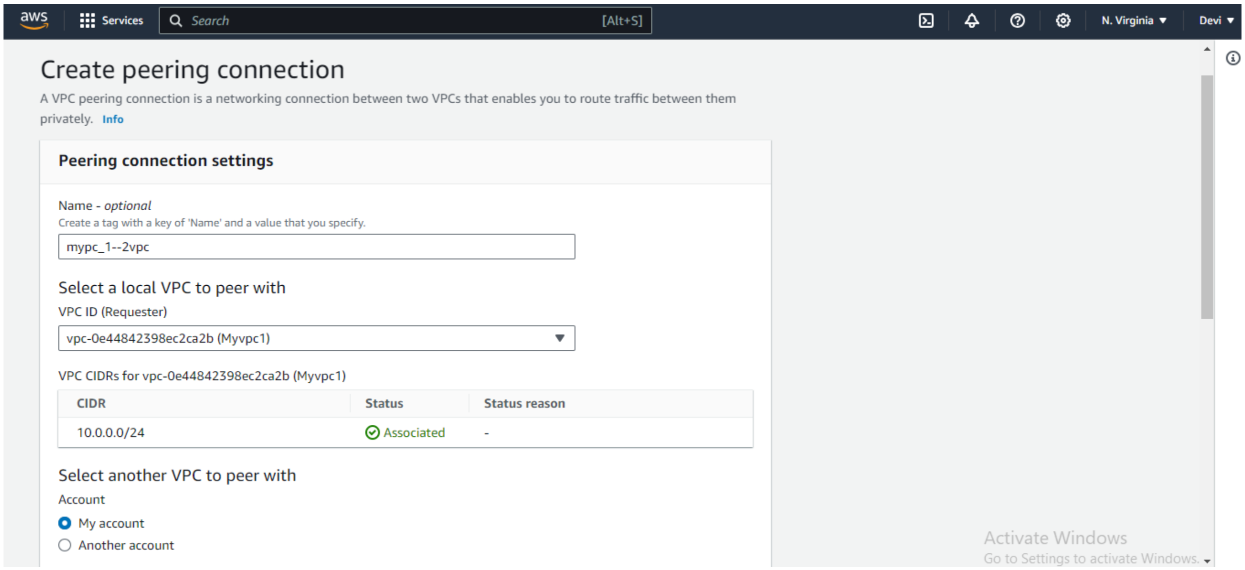
54.173.228.110 (ec2-user)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
2. 54.173.228.110 (ec2-user)
/home/ec2-user/
Name
.ssh
.bash_history
.bash_logout
.bash_profile
.bashrc
pri1.pem
Remote monitoring
Follow terminal folder
Activate Windows
Go to Settings to activate Windows.

VPC PEERING

- VPC peering is connecting private instance of one VPC from public subnet instance of other VPC. It may be between two VPCs of the same regions or different regions or between two VPC of different accounts.
- IPs of each VPCs should not match/overlap. Must be different.
- This is one to one connection.



- Create peering connection between two VPC’s of different IP ranges.



Create peering connection

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them privately. Info

Peering connection settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
mypc_1--2vpc

Select a local VPC to peer with

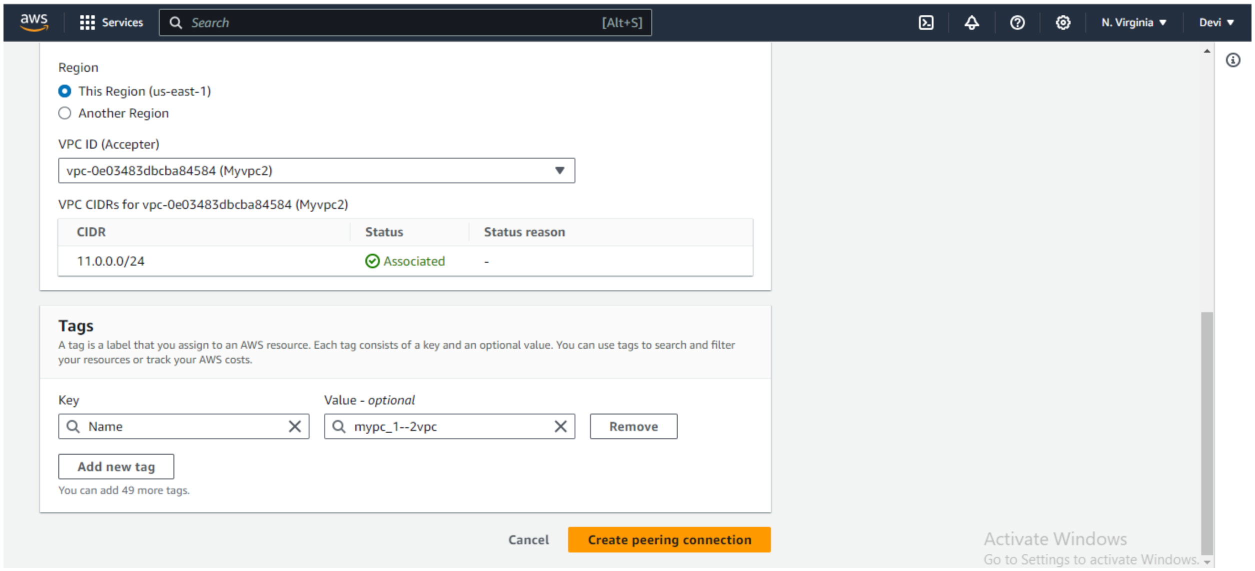
VPC ID (Requester)
vpc-0e44842398ec2ca2b (Myvpc1) ▾

VPC CIDRs for vpc-0e44842398ec2ca2b (Myvpc1)

CIDR	Status	Status reason
10.0.0.0/24	✓ Associated	-

Select another VPC to peer with

Account
● My account
○ Another account



Region
● This Region (us-east-1)
○ Another Region

VPC ID (Accepter)
vpc-0e03483dbcba84584 (Myvpc2) ▾

VPC CIDRs for vpc-0e03483dbcba84584 (Myvpc2)

CIDR	Status	Status reason
11.0.0.0/24	✓ Associated	-

Tags

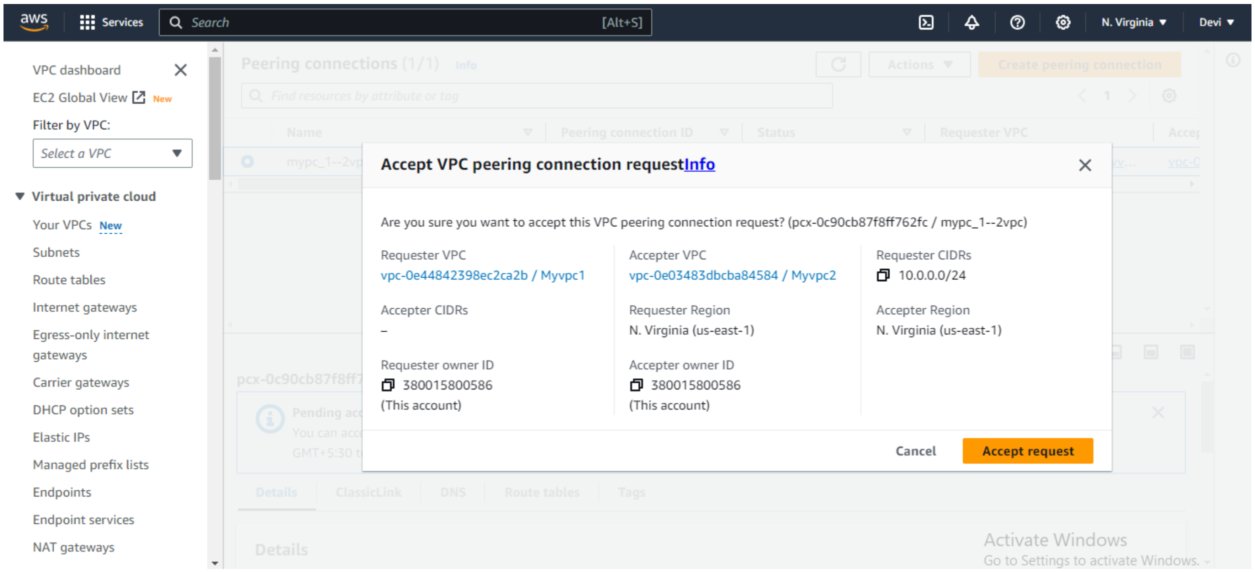
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*	
Name	mypc_1--2vpc	Remove

Add new tag
You can add 49 more tags.

Cancel Create peering connection

- Send request from one VPC to other and accept the request.

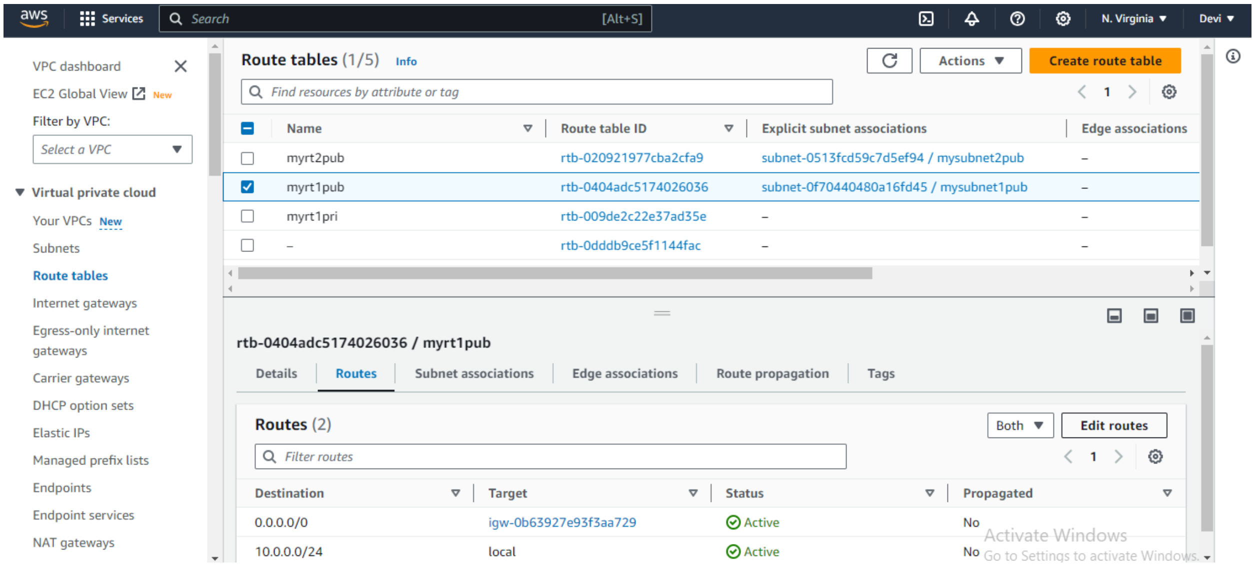


Accept VPC peering connection requestInfo

Are you sure you want to accept this VPC peering connection request? (pcx-0c90cb87f8ff762fc / mypc_1--2vpc)

Requester VPC	Accepter VPC	Requester CIDRs
vpc-0e44842398ec2ca2b / Myvpc1	vpc-0e03483dbcba84584 / Myvpc2	10.0.0.0/24
Accepter CIDRs	Requester Region	Accepter Region
–	N. Virginia (us-east-1)	N. Virginia (us-east-1)
Requester owner ID	Accepter owner ID	
380015800586 (This account)	380015800586 (This account)	

- Add 2nd private IP to 1st public IP and vice versa.



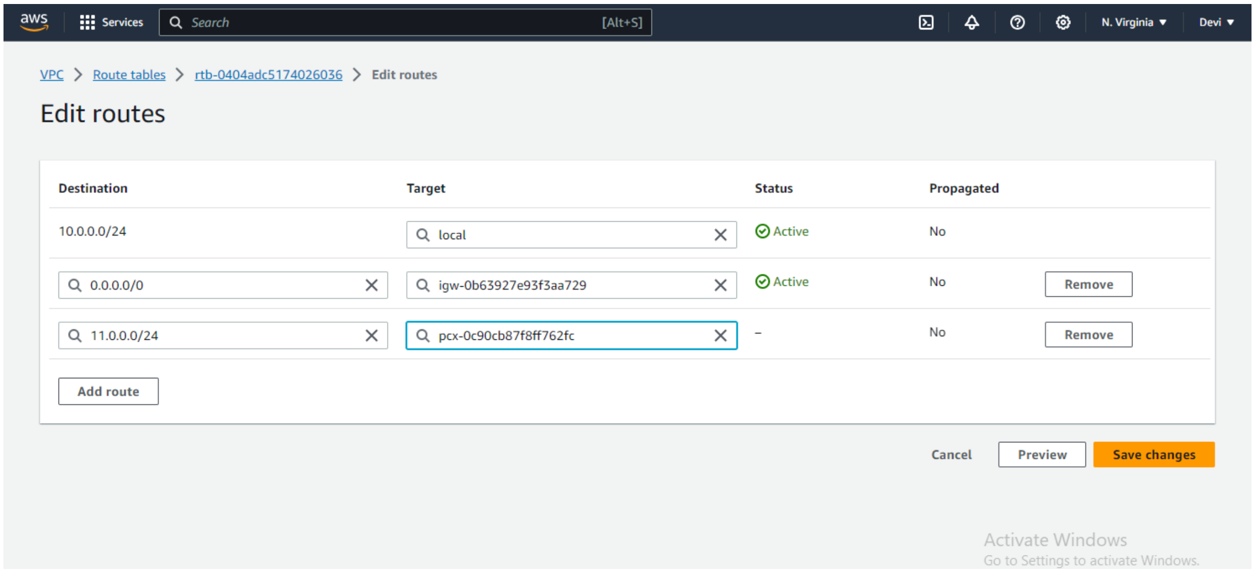
Route tables (1/5) Info

	Name	Route table ID	Explicit subnet associations	Edge associations
☐	myrt2pub	rtb-020921977cba2cfa9	subnet-0513fcd59c7d5ef94 / mysubnet2pub	–
☑	myrt1pub	rtb-0404adc5174026036	subnet-0f70440480a16fd45 / mysubnet1pub	–
☐	myrt1pri	rtb-009de2c22e37ad35e	–	–
☐	–	rtb-0dddb9ce5f1144fac	–	–

rtb-0404adc5174026036 / myrt1pub

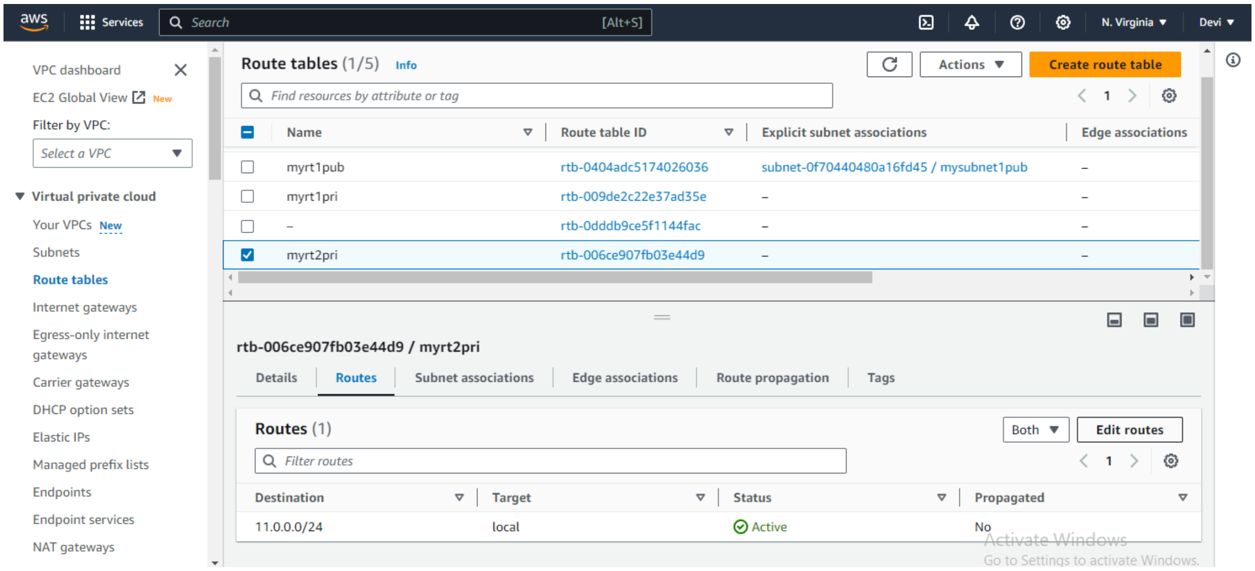
Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0b63927e93f3aa729	Active	No
10.0.0.0/24	local	Active	No


aws
Services
Search
[Alt+S]
N. Virginia
Devi
VPC > Route tables > rtb-0404adc5174026036 > Edit routes
Add route
Cancel
Preview
Save changes
Activate Windows
Go to Settings to activate Windows.

Edit routes

Destination	Target	Status	Propagated	
10.0.0.0/24	local	✓ Active	No	
0.0.0.0/0	igw-0b63927e93f3aa729	✓ Active	No	Remove
11.0.0.0/24	pcx-0c90cb87f8ff762fc	–	No	Remove



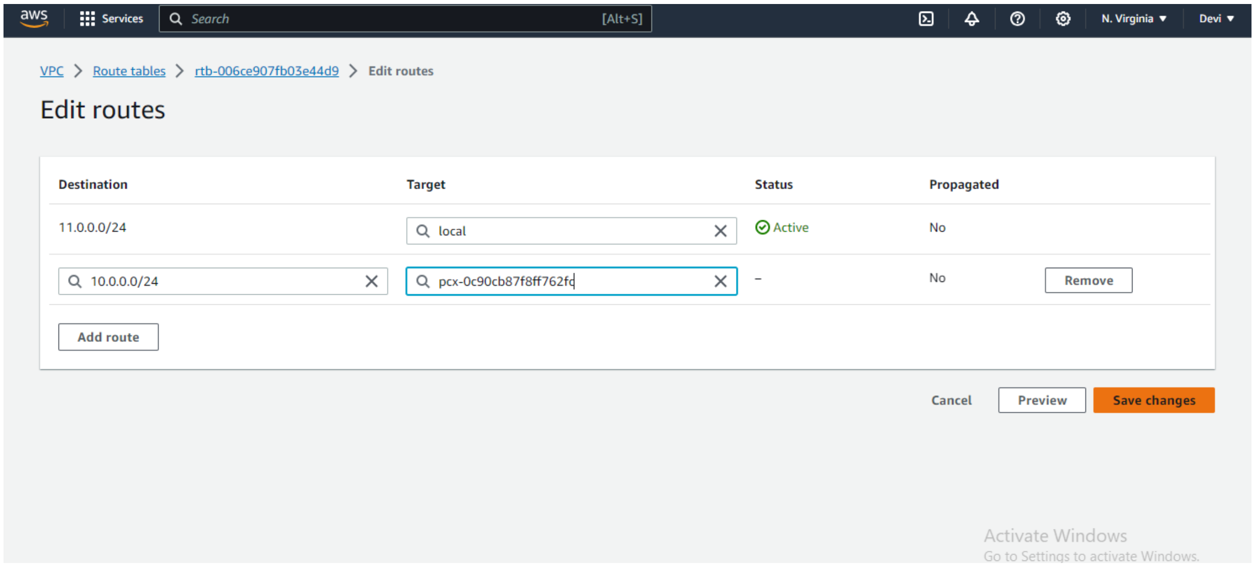
Route tables (1/5)

	Name	Route table ID	Explicit subnet associations	Edge associations
□	myrt1pub	rtb-0404adc5174026036	subnet-0f70440480a16fd45 / mysubnet1pub	–
□	myrt1pri	rtb-009de2c22e37ad35e	–	–
□	–	rtb-0dddb9ce5f1144fac	–	–
☑	myrt2pri	rtb-006ce907fb03e44d9	–	–

rtb-006ce907fb03e44d9 / myrt2pri

Routes (1)

Destination	Target	Status	Propagated
11.0.0.0/24	local	✓ Active	No

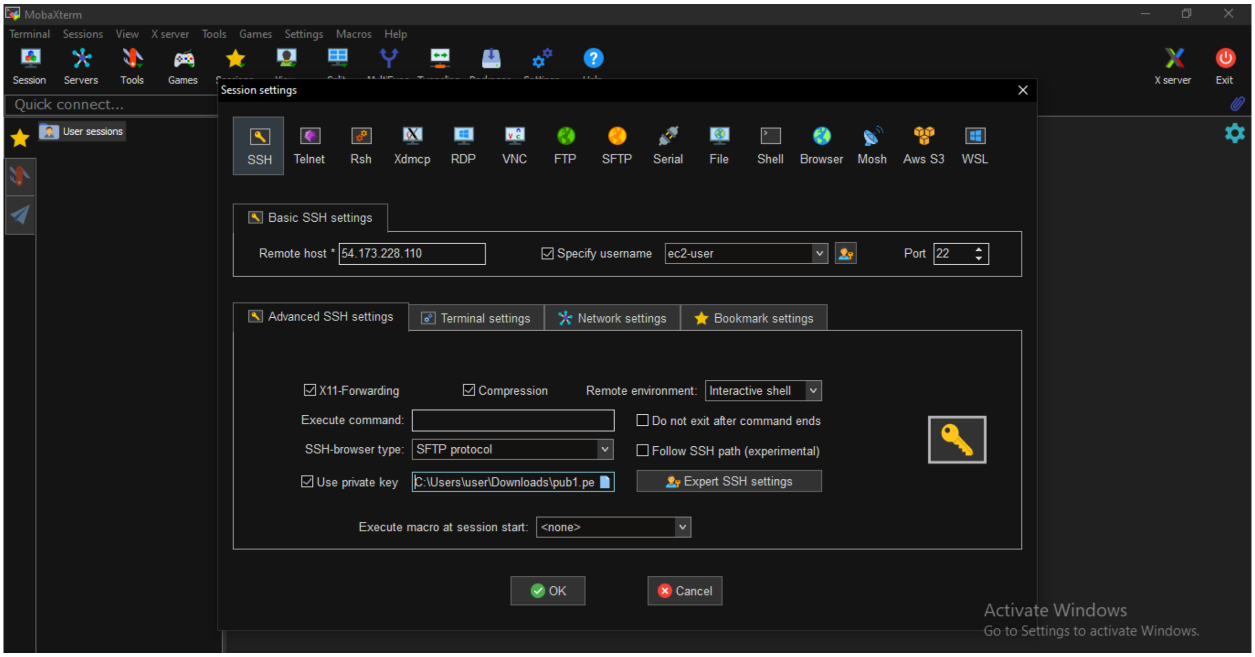


VPC > Route tables > rtb-006ce907fb03e44d9 > Edit routes

Edit routes

Destination	Target	Status	Propagated
11.0.0.0/24	local	Active	No
10.0.0.0/24	pcx-0c90cb87f8ff762fd	-	No

- Now connect through MobaXterm and ping the two VPC’s.



Session settings

Basic SSH settings

Remote host * 54.173.228.110 ☑ Specify username ec2-user Port 22

Advanced SSH settings

☑ X11-Forwarding ☑ Compression Remote environment: Interactive shell

Execute command: ☐ Do not exit after command ends

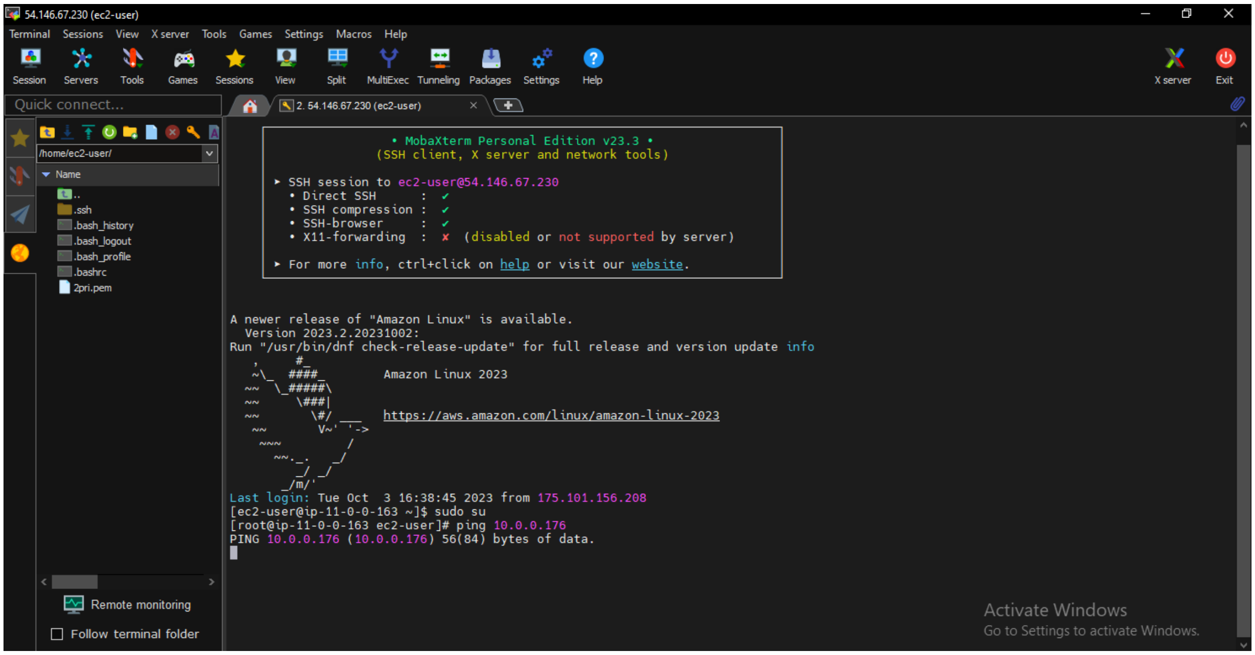
SSH-browser type: SFTP protocol ☐ Follow SSH path (experimental)

☑ Use private key C:\Users\user\Downloads\pub1.pe

Execute macro at session start: <none>

MobaXterm
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games
X server Exit
Quick connect...
User sessions
SSH Telnet Rsh Xdmcp RDP VNC FTP SFTP Serial File Shell Browser Mosh Aws S3 WSL
Terminal settings
Network settings
Bookmark settings
Expert SSH settings
OK
Cancel
Activate Windows
Go to Settings to activate Windows.

- Finally we connected.


54.146.67.230 (ec2-user)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
2. 54.146.67.230 (ec2-user)
/home/ec2-user/
Name
..
.ssh
.bash_history
.bash_logout
.bash_profile
.bashrc
2pri.pem
Remote monitoring
□ Follow terminal folder
Activate Windows
Go to Settings to activate Windows.

```
               • MobaXterm Personal Edition v23.3 •
             (SSH client, X server and network tools)

 ➤ SSH session to ec2-user@54.146.67.230
   • Direct SSH      :  ✓
   • SSH compression :  ✓
   • SSH-browser     :  ✓
   • X11-forwarding  :  ✘  (disabled or not supported by server)

 ➤ For more info, ctrl+click on help or visit our website.


A newer release of "Amazon Linux" is available.
  Version 2023.2.20231002:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Oct  3 16:38:45 2023 from 175.101.156.208
[ec2-user@ip-11-0-0-163 ~]$ sudo su
[root@ip-11-0-0-163 ec2-user]# ping 10.0.0.176
PING 10.0.0.176 (10.0.0.176) 56(84) bytes of data.
```

CONCLUSION :

We finally conclude that AWS provides a number of efficient, secure connectivity options to help users. In conclusion, a bastion host is an essential component for securing an AWS VPC environment, providing enhanced security, simplified access, scalability, and easy management. By using a bastion host, you can ensure secure and controlled access to private instances within the VPC environment, reduce complexity and overheads, and improve overall performance and security. And VPC peering connection helps the user to increase data flow speed. If users possess several AWS accounts then they can peer the VPCs across numerous accounts to build a file-sharing network.

----------------------- THE END ---------------------------