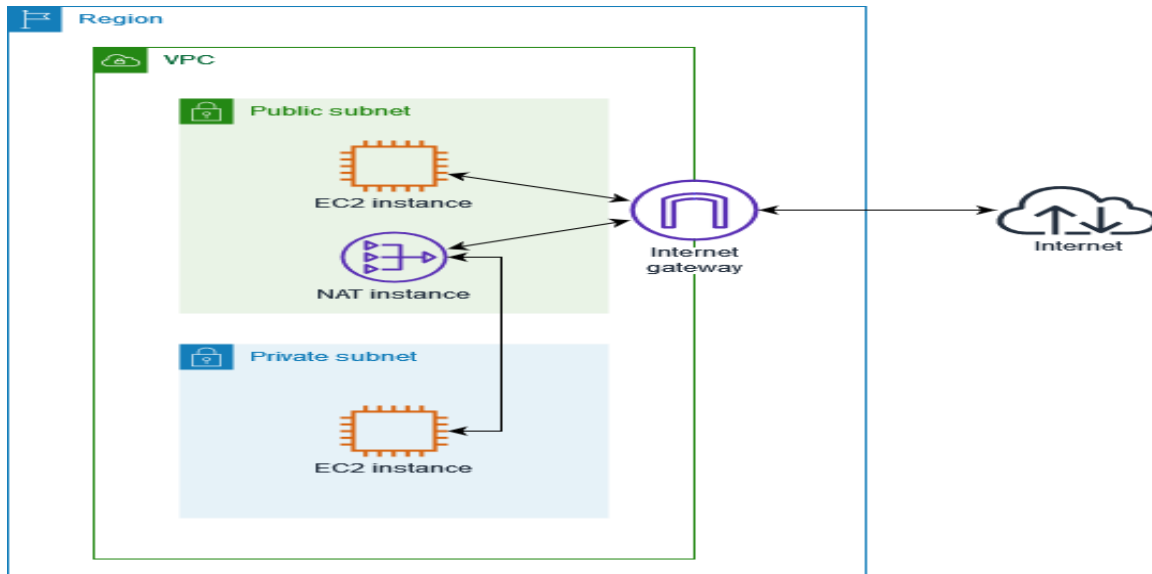


## WEEK :10

### AIM: Create and Configure Amazon Virtual Private Cloud (VPC).



#### TASK

Create your own VPC

Create Public subnet

Create Private subnet

Create Internet Gateway

Attach Internet Gateway to your VPC

Create Public Routing Table, associate subnet and add routing rules

Create Private Routing table, associate subnet and add routing rules

Launch an instance in Public network

Launch an instance in Private network

Create Nat Gateway

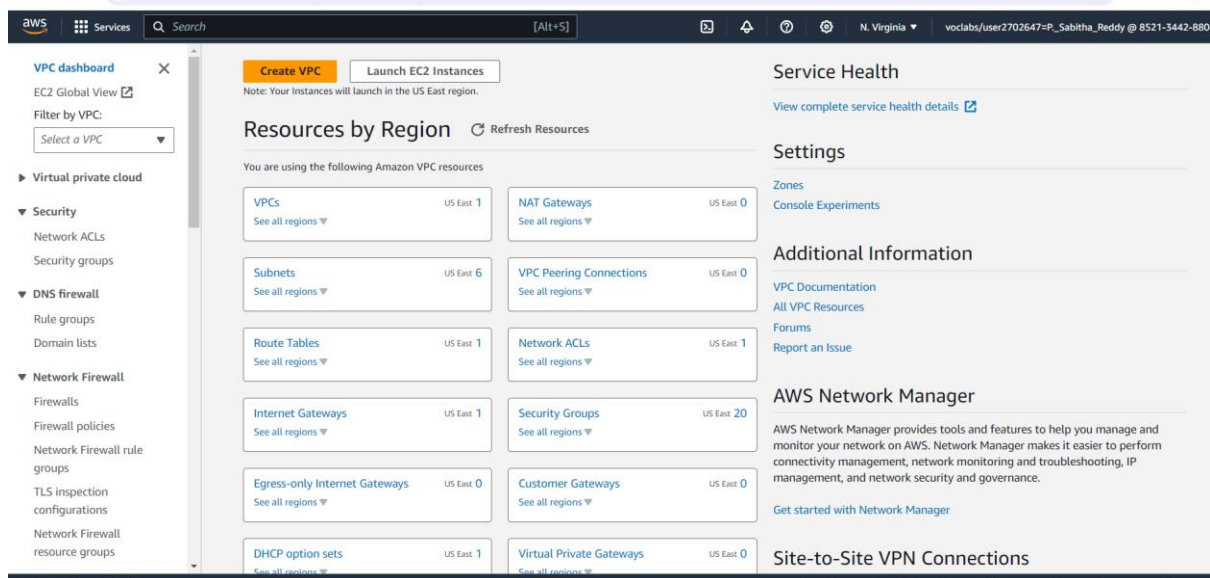
Connect to public instance and check internet connectivity

Connect to private instance and check internet connectivity

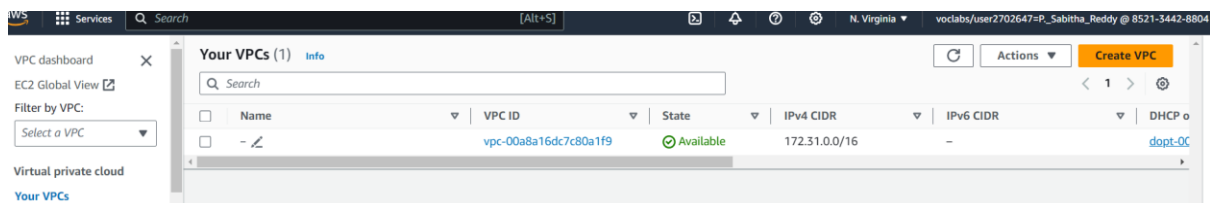
Amazon Virtual Private Cloud (Amazon VPC) enables you to launch Amazon Web Services (AWS) resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

**Step1:** Open AWS console,

Search for VPC in Search Bar,  
Click on **VPC**



On **VPC Dashboard Panel**,  
Click on **YOUR VPC**,  
Click on **CREATE VPC** Button



On Create VPC page,  
For Name Tag→lab\_vpc,  
For IPv4CIDR Block→10.0.0.0/16  
Leave remaining fields as default,  
Click on **CREATE VPC** Button.

## VPC settings

### Resources to create [Info](#)

Create only the VPC resource or the VPC and other networking resources.

☒ VPC only

☐ VPC and more

### Name tag - *optional*

Creates a tag with a key of 'Name' and a value that you specify.

lab\_vpc

### IPv4 CIDR block [Info](#)

☒ IPv4 CIDR manual input

☐ IPAM-allocated IPv4 CIDR block

### IPv4 CIDR

10.0.0.0/16

CIDR block size must be between /16 and /28.

### IPv6 CIDR block [Info](#)

☒ No IPv6 CIDR block

☐ IPAM-allocated IPv6 CIDR block

☐ Amazon-provided IPv6 CIDR block

☐ IPv6 CIDR owned by me

### Tenancy [Info](#)

Default

### IPv4 CIDR

10.0.0.0/16

CIDR block size must be between /16 and /28.

### IPv6 CIDR block [Info](#)

☒ No IPv6 CIDR block

☐ IPAM-allocated IPv6 CIDR block

☐ Amazon-provided IPv6 CIDR block

☐ IPv6 CIDR owned by me

### Tenancy [Info](#)

Default

## 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*



Remove tag

Add tag

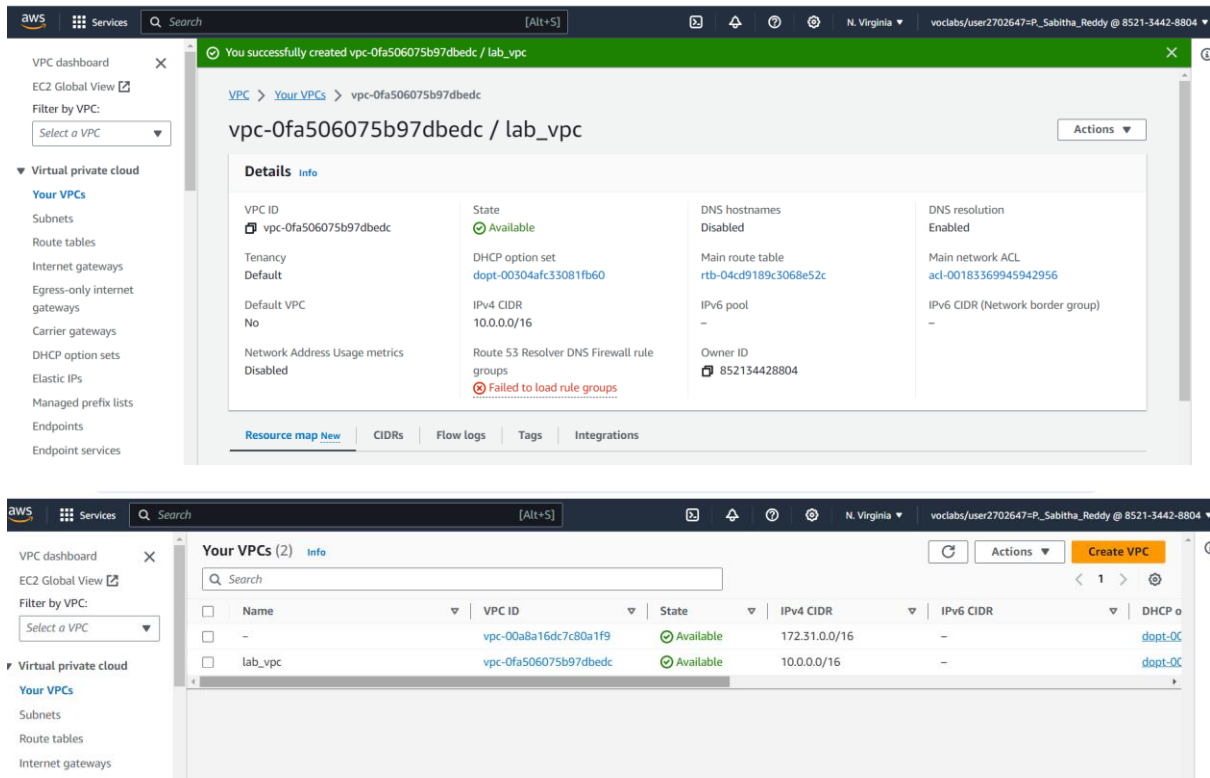
You can add 49 more tags

Cancel

Create VPC

# Verify

Lab\_vpc is Created.

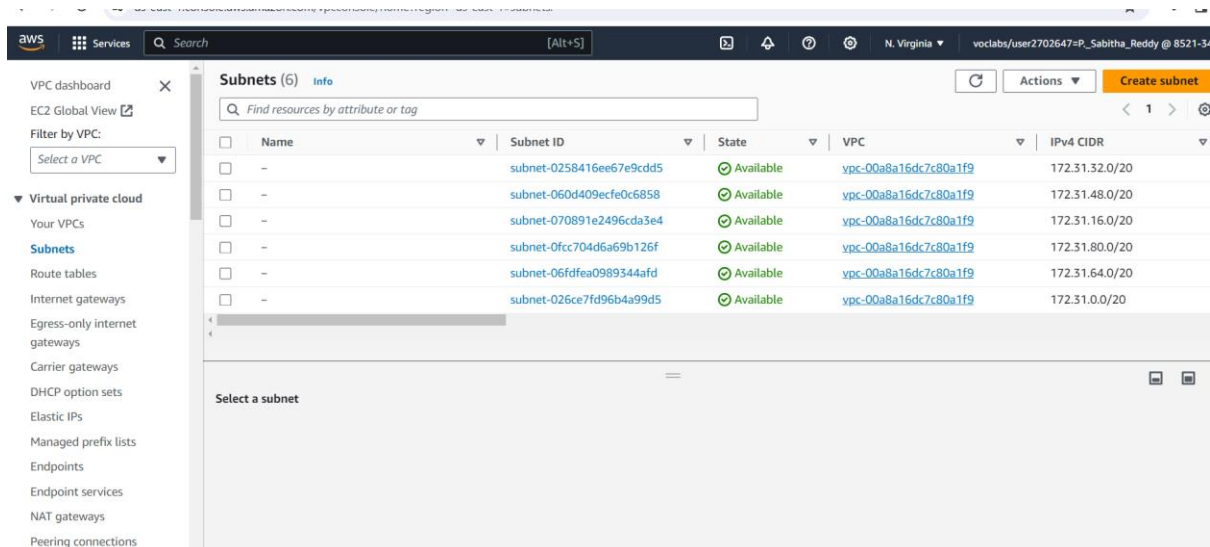


## Step2:

To Create Public Subnet

Click on Subnet

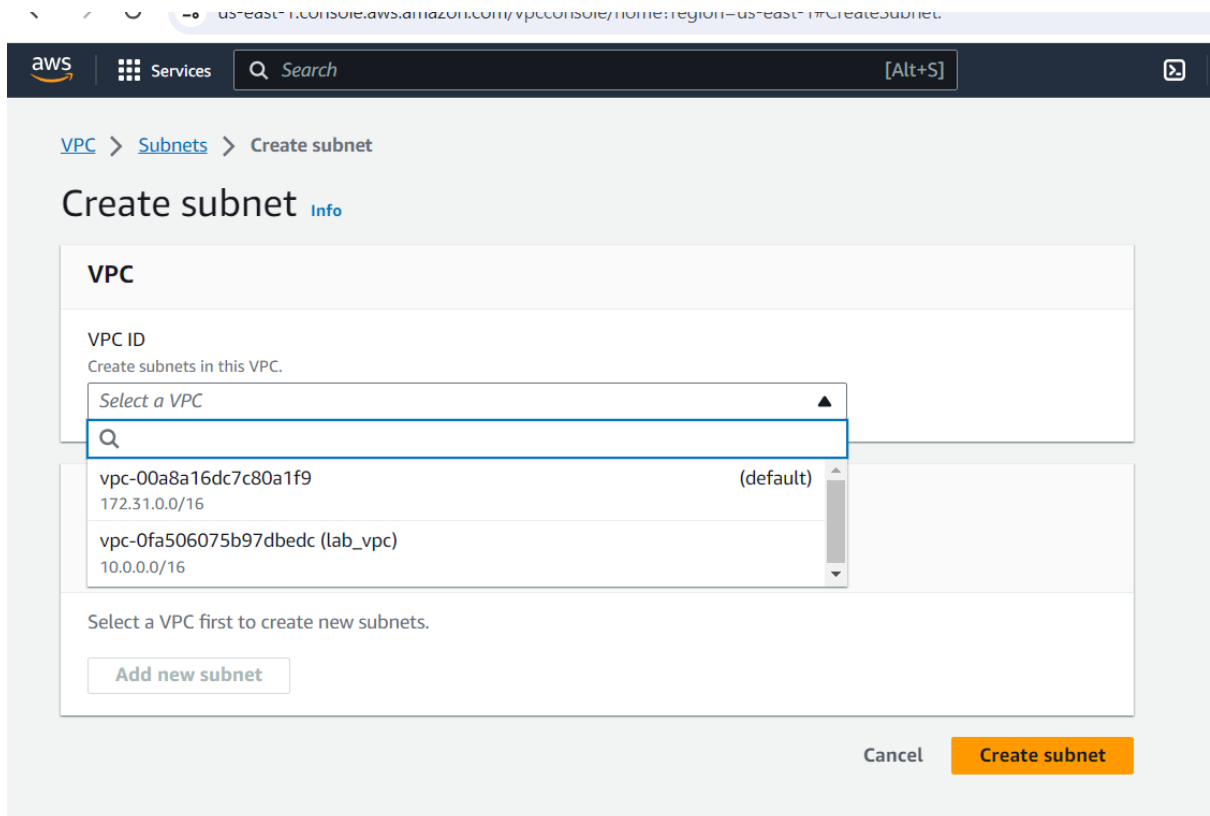
Click on Create Subnet button



On Create Subnet page

For VPC Id: lab\_vpc

Click on Create Subnet button



For Subnet Name → public\_subnet

Availability Zone → US East(N.Virginia)/us-east-1a

IPv4 VPC CIDR block → 10.0.0.0/16

IPv4 subnet CIDR block → 10.0.0.0/24

Click on Create Subnet button

VPC ID

Create subnets in this VPC.

vpc-0fa506075b97dbedc (lab\_vpc) ▼

Associated VPC CIDRs

IPv4 CIDRs

10.0.0.0/16

Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name

Create a tag with a key of 'Name' and a value that you specify.

my-subnet-01

The name can be up to 256 characters long.

Availability Zone [Info](#)

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

No preference ▼

IPv4 VPC CIDR block [Info](#)

Choose the IPv4 VPC CIDR block to create a subnet in.

### Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

#### Subnet 1 of 1

##### Subnet name

Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

##### Availability Zone [Info](#)

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

##### IPv4 VPC CIDR block [Info](#)

Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

##### IPv4 subnet CIDR block

256 IPs

#### ▼ Tags - optional

Key

Value - optional

## Step 3:

Click on ADD NEW SUBNET BUTTON

For Subnet Name → private\_subnet

Availability Zone → US East(N.Virginia)/us-east-1a

IPv4 VPC CIDR block → 10.0.0.0/16

IPv4 subnet CIDR block → 10.0.1.0/24

Click on Create Subnet button

## Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

### Subnet 1 of 1

#### Subnet name

Create a tag with a key of 'Name' and a value that you specify.

Private\_subnet

The name can be up to 256 characters long.

#### Availability Zone

[Info](#)

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

United States (N. Virginia) / us-east-1a

#### IPv4 VPC CIDR block

[Info](#)

Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

10.0.0.0/16

#### IPv4 subnet CIDR block

10.0.1.0/24

256 IPs

#### Tags - optional

##### Key

Q Name

##### Value - optional

Q Private\_subnet

Remove

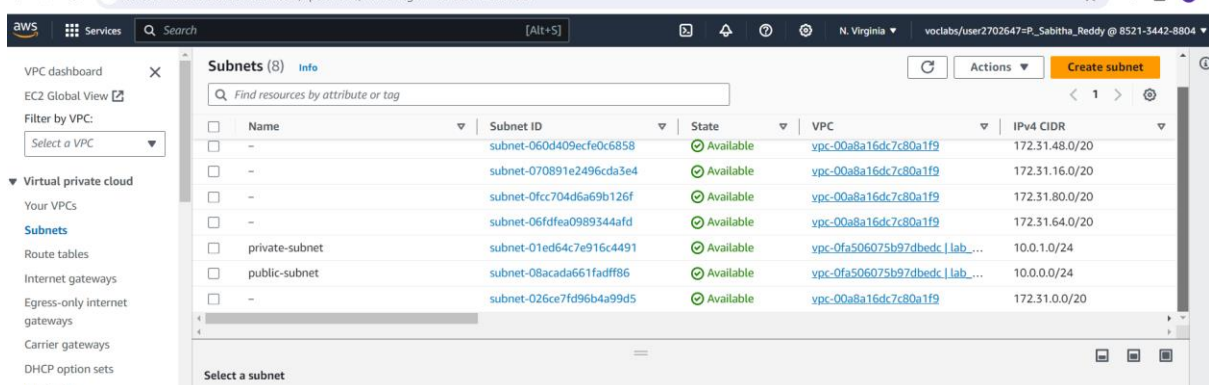
Add new tag

You can add 49 more tags.

Remove

Add new subnet

Verify public, private subnets are created.



The screenshot shows the AWS VPC console with a list of subnets. The table has columns for Name, Subnet ID, State, VPC, and IPv4 CIDR. There are 8 subnets listed, including private and public subnets.

Name	Subnet ID	State	VPC	IPv4 CIDR
-	subnet-060d409ecfe0c6858	Available	vpc-00a8a16dc7c80a1f9	172.31.48.0/20
-	subnet-070891e2496cda3e4	Available	vpc-00a8a16dc7c80a1f9	172.31.16.0/20
-	subnet-0fcc704d6a69b126f	Available	vpc-00a8a16dc7c80a1f9	172.31.80.0/20
-	subnet-06fdfea0989344afd	Available	vpc-00a8a16dc7c80a1f9	172.31.64.0/20
private-subnet	subnet-01ed64c7e916c4491	Available	vpc-0fa506075b97dbedc   lab...	10.0.1.0/24
public-subnet	subnet-08acada661fadff86	Available	vpc-0fa506075b97dbedc   lab...	10.0.0.0/24
-	subnet-026ce7fd96b4a99d5	Available	vpc-00a8a16dc7c80a1f9	172.31.0.0/20

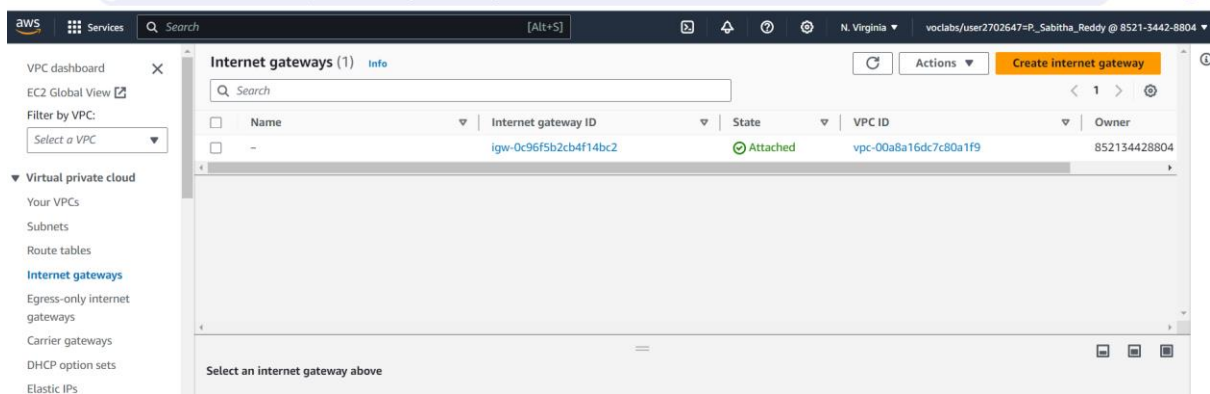
**Step 4: Create INTERNET GATEWAY and ATTACH TO VPC**

In VPC Dashboard Panel

Click on Internet Gateways

Click on Create Internet Gateway button





In Create Internet Gateway page

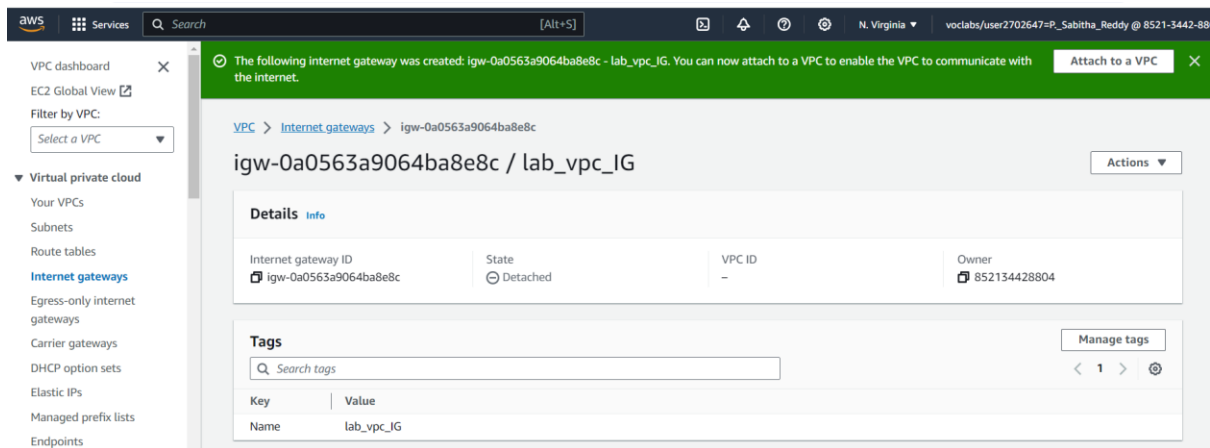
For Name Tag → lab\_vpc\_IG

Click on Create Internet Gateway button

The screenshot shows the 'Create internet gateway' page. It has a breadcrumb trail: 'VPC > Internet gateways > Create internet gateway'. The main heading is 'Create internet gateway' with an 'Info' link. A descriptive paragraph explains that an internet gateway is a virtual router. Below this is the 'Internet gateway settings' section, which includes a 'Name tag' field with the value 'lab\_vpc\_IG'. There is also a 'Tags - optional' section with a table for adding tags. The table has columns for 'Key' and 'Value - optional'. One tag is added with key 'Name' and value 'lab\_vpc\_IG'. At the bottom right are 'Cancel' and 'Create internet gateway' buttons.

Verify

Internet Gateway created.



## Step 5:

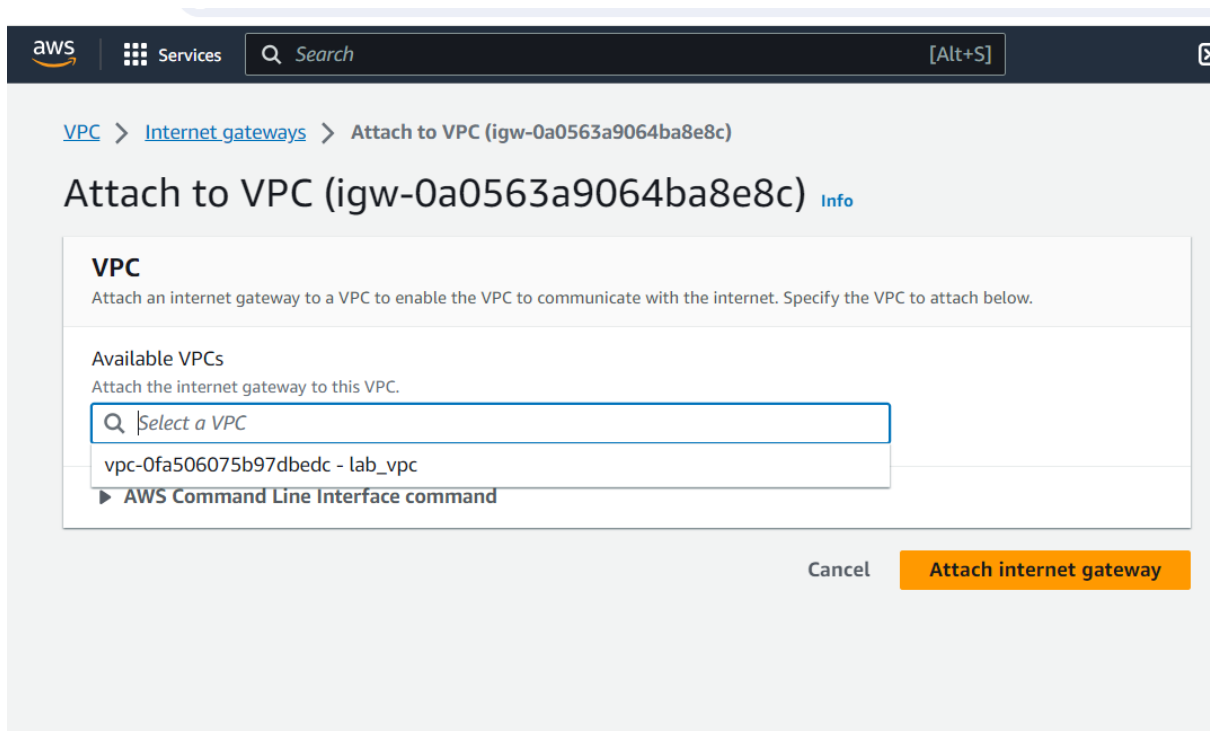
Select lab\_vpc\_IG

Click on ATTACH to VPC

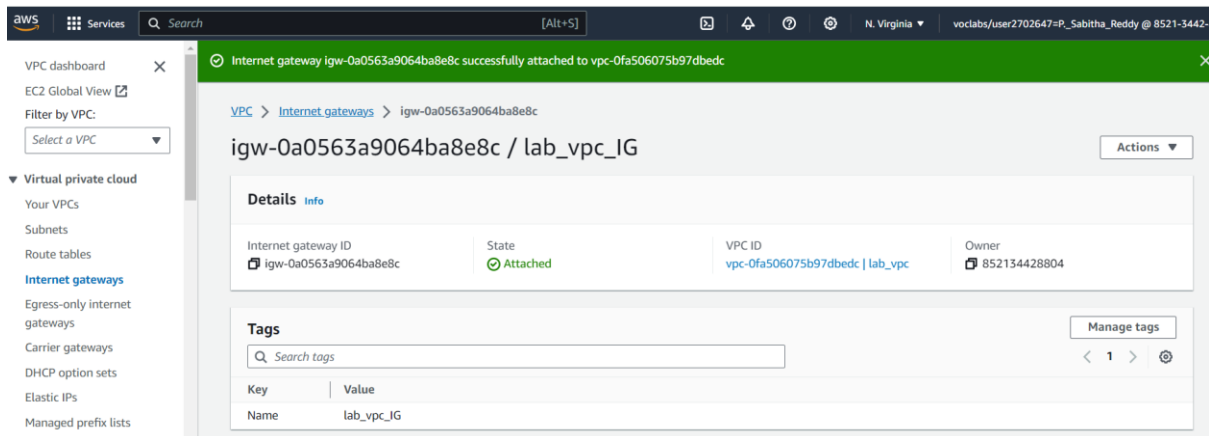
In ATTACH to VPC box

For VPC→lab\_vpc

Click on attach internet gateway button.



**Verify** Internet gateway is connected to your VPC

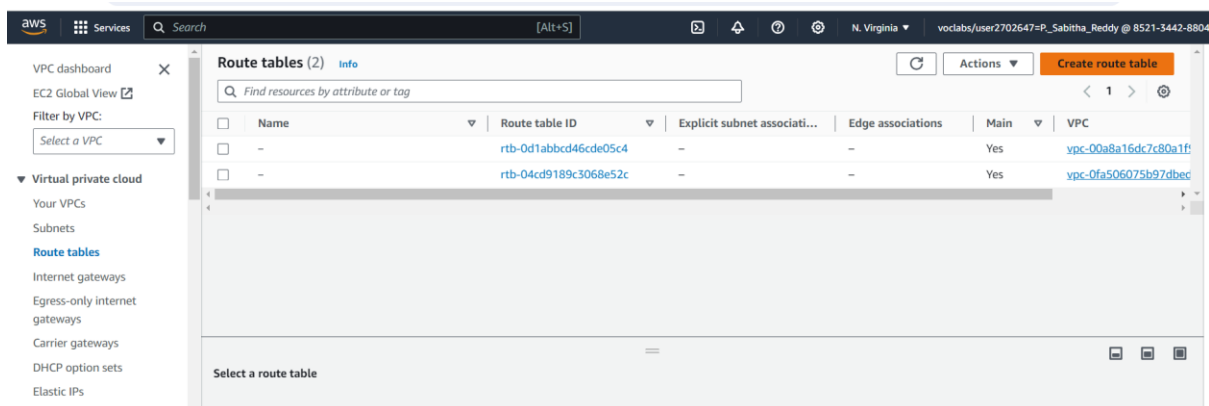


**Step 6:** Create Public Routing Table, associate subnet and add routing rules

On VPC Dashboard panel

Click on Route Table

Click on Create route table button



On route table box

For Name Tag → PUBLIC-RT-lab\_vpc

For VPC → lab\_vpc

Click on Create route table button

VPC > Route tables > Create route table

## Create route table [Info](#)

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

### Route table settings

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

**VPC**  
The VPC to use for this route table.

### 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	
<input type="text" value="Name"/>	<input type="text" value="PUBLIC-RT-lab_vpc"/>	<input type="button" value="Remove"/>

You can add 49 more tags.

**Verify ,**

PUBLIC-RT-lab\_vpc Is created.

**Step 7:**

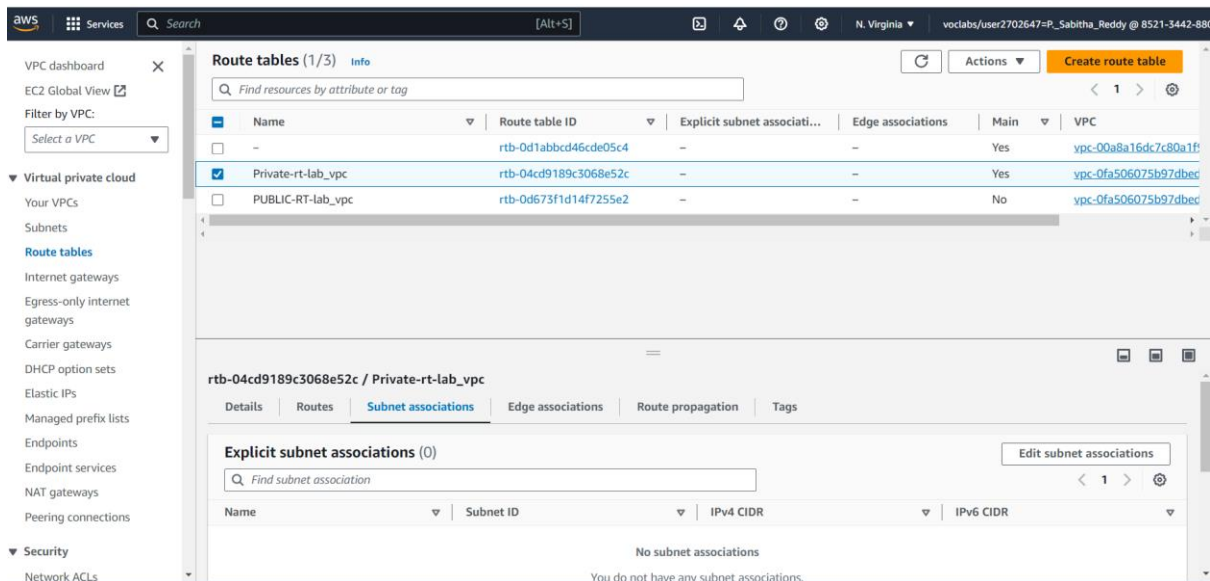
For Name Tag→PRIVATE-RT-lab\_vpc

For VPC→ lab\_vpc

Click on Create route table button

Select Private-RT-lab\_vpc →Actions

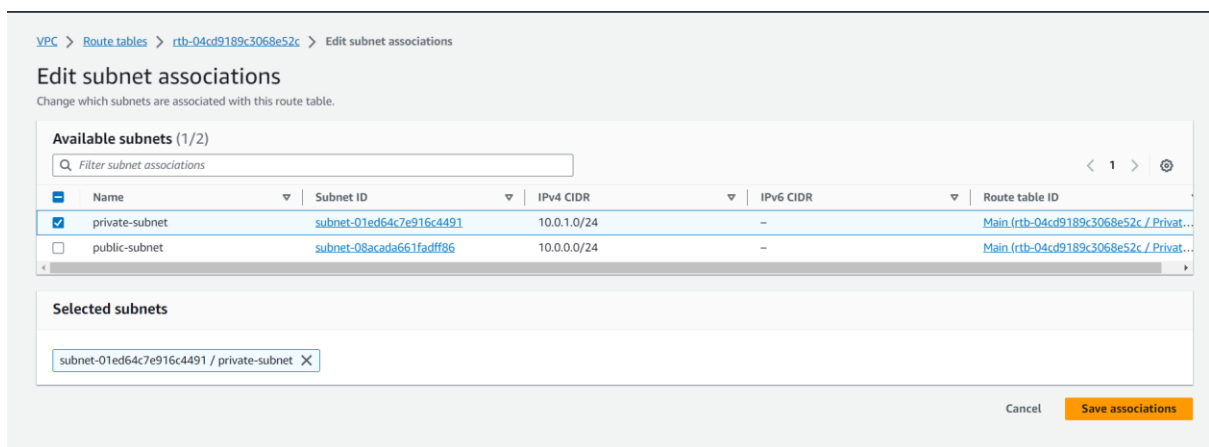
Click on subnet associations



Click on Edit subnet associations

Select check box of private\_subnet → 10.0.1.0/24

Click on save associations

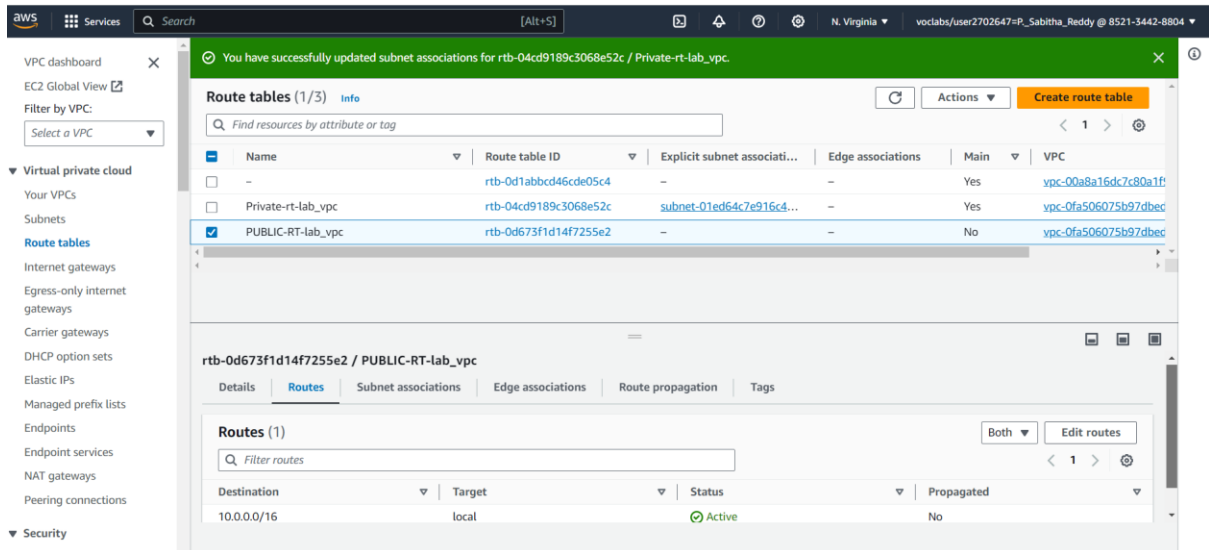


Verify private\_subnet is associated with routing table .

Select PUBLIC-RT-lab\_vpc → Actions

Click on subnet associations

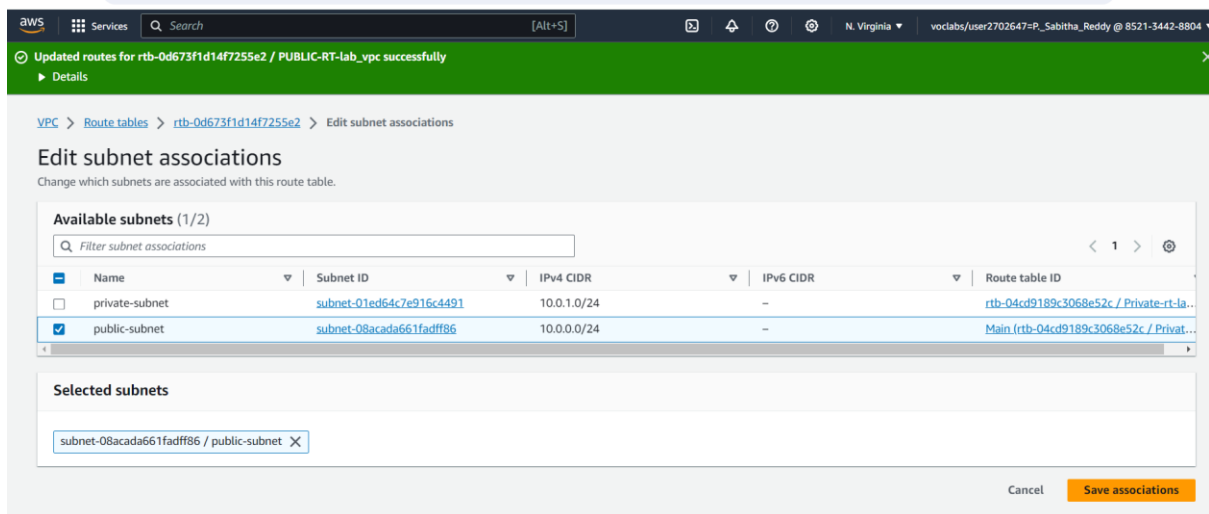
Click on Edit subnet associations



Select check box of public\_subnet → 10.0.0.0/24

Click on save associations

Verify public\_subnet is associated with routing table



Select PUBLIC-RT-lab\_vpc → Actions

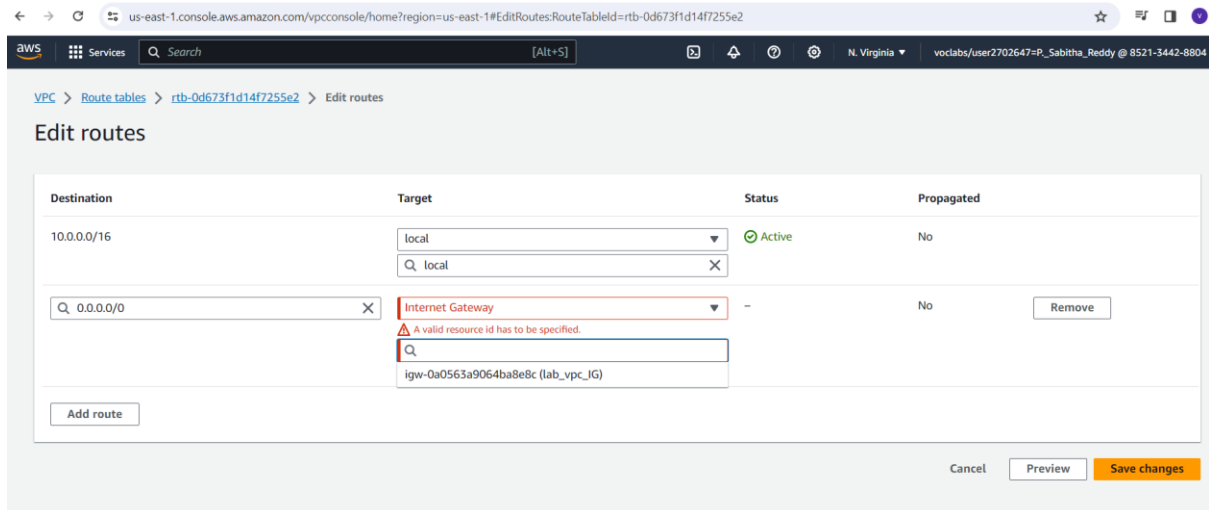
Click on Edit routes button,

Click on add route button,

For Destination → 0.0.0.0/0

Target → internet gateway → igw-0a0563a9064ba8e8c(lab\_vpc\_IG)

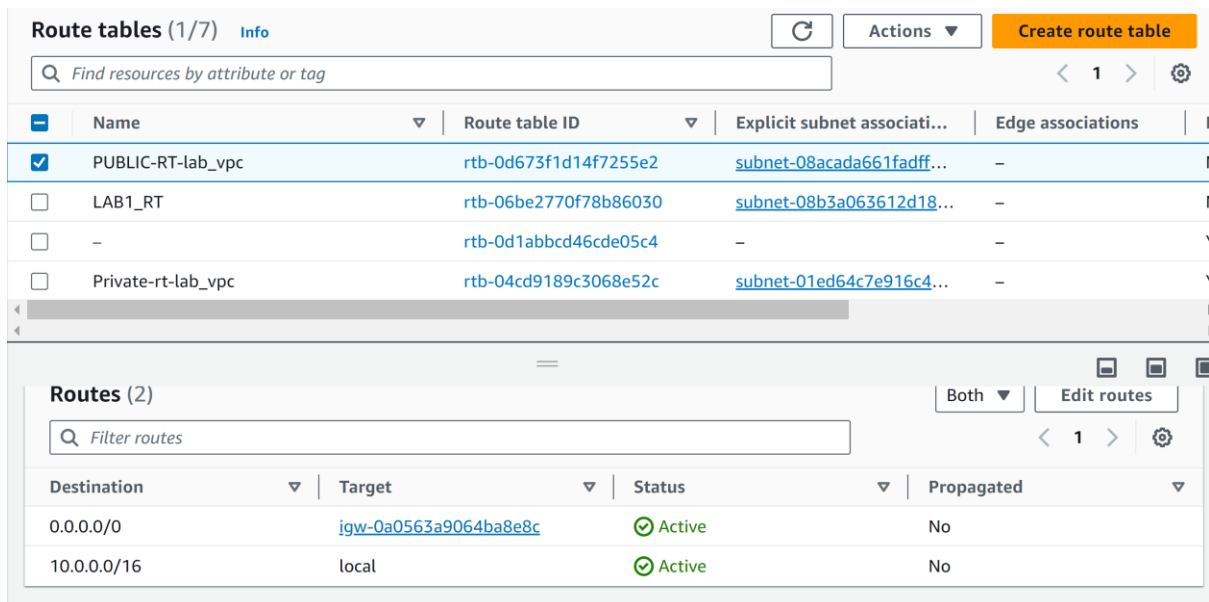
Click on Save changes button



Verification

PUBLIC-RT-lab\_vpc is added through Internet Gateway

Verify Status column is Active.



**Step 8:**

Create Amazon Linux Instance with lab\_vpc

## Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

### Name and tags [Info](#)

Name

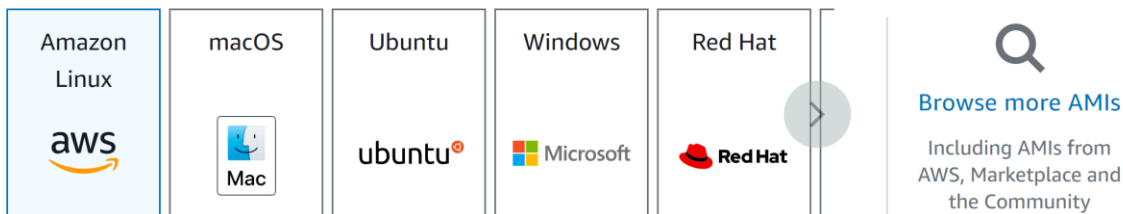
[Add additional tags](#)

### ▼ Application and OS Images (Amazon Machine Image) [Info](#)

Select AMI → amazon linux, instance type → t2.micro,

Recents

**Quick Start**



#### Amazon Machine Image (AMI)

Amazon Linux 2023 AMI

ami-0230bd60aa48260c6 (64-bit (x86)) / ami-04c97e62cb19d53f1 (64-bit (Arm))

Virtualization: hvm    ENA enabled: true    Root device type: ebs

Free tier eligible ▼

#### Description

Amazon Linux 2023 AMI 2023.2.20231113.0 x86\_64 HVM kernel-6.1

Click on create new key pair,

Key pair name → vpc\_linux

Click on create key pair



**Create key pair**

Key pair name  
Key pairs allow you to connect to your instance securely.  
vpc\_linux  
The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ RSA  
RSA encrypted private and public key pair

☐ ED25519  
ED25519 encrypted private and public key pair

Private key file format

☒ .pem  
For use with OpenSSH

☐ .ppk  
For use with PuTTY

Cancel Create key pair

Click on Edit on Network settings

▼ **Network settings** Info Edit

Network Info  
vpc-00a8a16dc7c80a1f9

Subnet Info  
No preference (Default subnet in any availability zone)

Auto-assign public IP Info  
Enable

For VPC → lab\_vpc,

Subnet → public-subnet

Auto-assign public IP → enable

▼ Network settings

Info

VPC - required

Info

vpc-0fa506075b97dbedc (lab\_vpc)

10.0.0.0/16

▼

↻

Subnet

Info

subnet-08acada661fadff86

public-subnet

▼

↻

Create new subnet

↗

VPC: vpc-0fa506075b97dbedc

Owner: 852134428804

Availability Zone: us-east-1a

IP addresses available: 251

CIDR: 10.0.0.0/24)

Auto-assign public IP

Info

Enable

▼

Firewall (security groups)

Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group

☐ Select existing security group

Configure storage settings are default.

Click on launch Instance

aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user2702647=P\_Sabitha\_Reddy @

Key pair name - required

vpc\_lab\_key

▼

↻

Create new key pair

▼ Network settings

Info

VPC - required

Info

vpc-0fa506075b97dbedc (lab\_vpc)

10.0.0.0/16

▼

↻

Subnet

Info

subnet-01ed64c7e916c4491

private-subnet

▼

↻

Create new subnet

↗

VPC: vpc-0fa506075b97dbedc

Owner: 852134428804

Availability Zone: us-east-1a

IP addresses available: 251

CIDR: 10.0.1.0/24)

Auto-assign public IP

Info

Enable

▼

Firewall (security groups)

Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group

☐ Select existing security group

Security group name - required

launch-wizard-21

▼

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and \_-:/[]@!+=&:[]\$\*

▼ Summary

Number of instances

Info

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.2.2...read more

ami-0230bd60aa48260c6

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance

Cancel

Launch instance

Review commands

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Remove

Type | Info

ssh ▼

Protocol | Info

TCP

Port range | Info

22

Source type | Info

Anywhere ▼

Source | Info

Q Add CIDR, prefix list or security group

0.0.0.0/0 ✕

Description - optional | Info

e.g. SSH for admin desktop

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0)

Remove

Type | Info

HTTP ▼

Protocol | Info

TCP

Port range | Info

80

Source type | Info

Anywhere ▼

Source | Info

Q Add CIDR, prefix list or security group

0.0.0.0/0 ✕

Description - optional | Info

e.g. SSH for admin desktop

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

✕

Add security group rule

► Advanced network configuration

## Step 9:

Create one more instance in same vpc and subnet is Private-subnet

▼ **Network settings** [Info](#)

VPC - *required* [Info](#)

vpc-0fa506075b97dbedc (lab\_vpc)  
10.0.0.0/16

Subnet [Info](#)

subnet-01ed64c7e916c4491 private-subnet  
VPC: vpc-0fa506075b97dbedc Owner: 852134428804 Availability Zone: us-east-1a  
IP addresses available: 251 CIDR: 10.0.1.0/24

Auto-assign public IP [Info](#)

Enable

**Firewall (security groups)** [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

Security group name - *required*

launch-wizard-28

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and . \_ - / ( ) # , @ [ ] + = & ; { } ! \$ \*

Description - *required* [Info](#)

launch-wizard-28 created 2023-11-28T10:12:23.329Z

Configure storage settings are default.

Click on launch Instance

So here we created two amazon linux instances with same VPC and SUBNETS are PRIVATE & PUBLIC subnets.

**Step 10:**

**Create NAT Gateway :**  
On VPC Dashboard panel → NAT Gateways

**NAT gateways** (1) [Info](#)

Find resources by attribute or tag

Name	NAT gateway ID	Connectivity...	State	State message	Primary public I...	Primary private I...	Primary network...	VPC
lab_vpc_nat	nat-03854956efc2b621e	Public	Available	-	52.206.123.29	10.0.0.48	eni-0d89d27bf8edc...	vpc-0f395b4f6e820e6

Click on Create NAT Gateway button → **Name tag: PTInstance\_NAT**

**Subnet :** Public\_subnet,

**Connectivity type:** Public,

**Elastic IP allocation ID:** Click on allocate Elastic IP button

[/PC](#) > [NAT gateways](#) > Create NAT gateway

### Create NAT gateway Info

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

**NAT gateway settings**

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

PTInstance\_NAT

The name can be up to 256 characters long.

**Subnet**  
Select a subnet in which to create the NAT gateway.

subnet-01ba0146d1634d3b7 (Public\_subnet)

**Connectivity type**  
Select a connectivity type for the NAT gateway.

☒ Public  
☐ Private

**Elastic IP allocation ID** Info  
Assign an Elastic IP address to the NAT gateway.

Select an Elastic IP

[Allocate Elastic IP](#)

► **Additional settings** Info

**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**

Q Name

**Value - optional**

Q PTInstance\_NAT

[Remove](#)

[Add new tag](#)

You can add 49 more tags.

[Cancel](#) [Create NAT gateway](#)

🔔 NAT gateway nat-00e5a05dbd11862b8 | PTInstance\_NAT was created successfully.

**NAT gateways (2)** Info

Find resources by attribute or tag

	Name	NAT gateway ID	Connectivity...	State	State message	Primary public I...	Primary private I...	Primary network...	VPC
<input type="radio"/>	lab_vpc_nat	nat-03854956efc2b621e	Public	Available	–	52.206.123.29	10.0.0.48	eni-0d89d27bf8edc...	vpc-0f395b4f6e820e6
<input checked="" type="radio"/>	PTInstance_NAT	nat-00e5a05dbd11862b8	Public	Available	–	18.209.91.206	10.0.0.211	eni-0321d39ce3442...	vpc-0f395b4f6e820e6

Then Attach this NAT Gateway to Private Route Table,  
Click on **Route Tables** → select Private-RT-lab\_vpc  
→ **Actions** → edit Routes → add route →

**Edit routes**

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
<input type="text" value="Q local"/>			
<a href="#">Add route</a>			

[Cancel](#) [Preview](#) [Save changes](#)

**Destination** is Public IP address, 0.0.0.0/0, **Target :** NAT Gateway, select PTInstance\_NAT

> [rtb-07f59c314636ddc13](#) > Edit routes

### Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	NAT Gateway		No

[Add route](#) [Cancel](#) [Preview](#) [Save changes](#)

Click on Save Changes.

### Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	NAT Gateway		No

[Add route](#) [Cancel](#) [Preview](#) [Save changes](#) [Actions](#)

rtb-07f59c314636ddc13 / Private-RT-lab\_vpc

**Details** Info
 

Route table ID  
rtb-07f59c314636ddc13

VPC  
vpc-0f395b4f6e820e678 | lab\_vpc

**Main**

Main  
No

Owner ID  
794256376414

**Explicit subnet associations**  
[subnet-06542bb83bb34a8ee / Private\\_subnet](#)

**Edge associations**  
 -

[Routes](#)
[Subnet associations](#)
[Edge associations](#)
[Route propagation](#)
[Tags](#)

**Routes (2)**

[Filter routes](#)

Destination	Target	Status	Propagated
0.0.0.0/0	nat-00e5a05dbd11862b8	Active	No
10.0.0.0/16	local	Active	No

**Step 11:**  
**Connect to public instance and check internet connectivity.**

Connect the Public Linux instance with **SSH client** connection or **EC2 Instance Connect**.

If connecting with **SSH Client**, open command prompt in local system and change path for folder where your Linux instance key pair available in local system. Then paste the instance path.

**Connect to instance** [Info](#)

Connect to your instance i-0a567ad3fdafd3827 (serverinstance) using any of these options

## EC2 Instance Connect

## Session Manager

SSH client

## EC2 serial console

Instance ID

 i-0a567ad3fdafd3827 (serverinstance)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is `serverkey.pem`.
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
`❏ chmod 400 "serverkey.pem"`
4. Connect to your instance using its Public DNS:  
`❏ ec2-3-83-217-134.compute-1.amazonaws.com`

Example:

```
ssh -i "serverkey.pem" ec2-user@ec2-3-83-217-134.compute-1.amazonaws.com
```

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

[illegible]

To check internet Connectivity in Linux instance  
Command: ping www.google.com

```

ec2-user@ip-10-0-0-100 ~
Amazon Linux 2023

https://aws.amazon.com/linux/amazon-linux-2023

Last login: Tue Mar 18 07:28:47 2025 from 119.235.52.196
[ec2-user@ip-10-0-0-100 ~]$ ping www.google.com
PING www.google.com (142.250.31.99) 56(64) bytes of data:
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=1 ttl=58 time=2.66 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=2 ttl=58 time=2.64 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=3 ttl=58 time=2.65 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=4 ttl=58 time=2.42 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=5 ttl=58 time=2.64 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=6 ttl=58 time=2.41 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=7 ttl=58 time=2.65 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=8 ttl=58 time=2.61 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=9 ttl=58 time=2.96 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=10 ttl=58 time=2.79 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=11 ttl=58 time=2.61 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=12 ttl=58 time=2.34 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=13 ttl=58 time=2.53 ms
^C
--- www.google.com ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12021ms
rtt min/avg/max/mdev = 2.337/2.608/2.957/0.157 ms
[ec2-user@ip-10-0-0-100 ~]$

```

If data packet's are transmitted without any loss then can understood that instance have internet connection.  
with this we can say that Public Instance have internet connectivity.

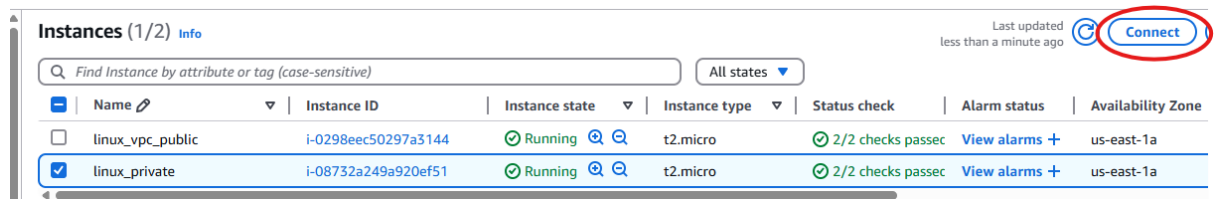
## Step 12:

### Connect to private instance and check internet connectivity.

Connect the Private Linux instance with **SSH client** connection or **EC2 Instance Connect**.

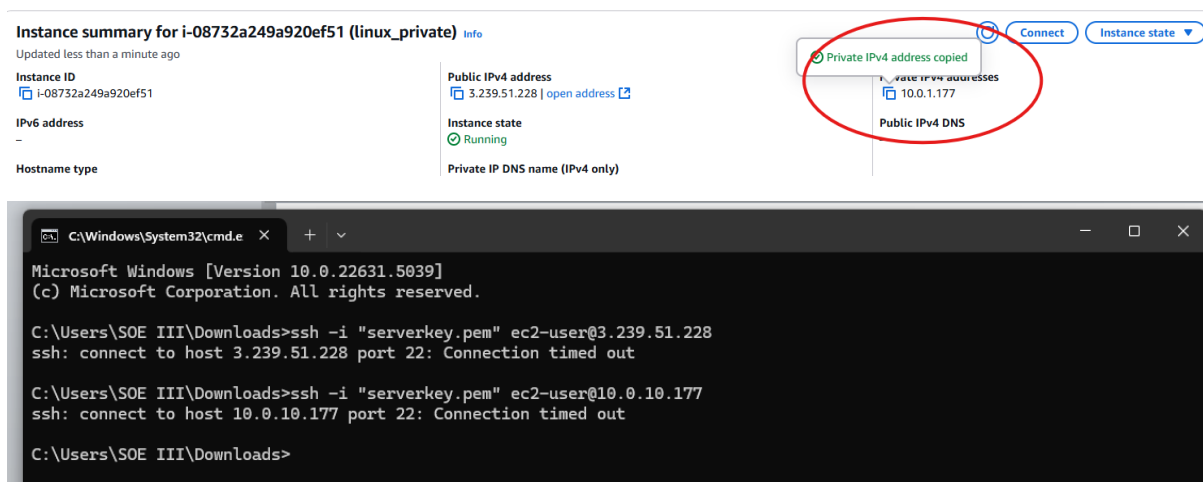
If connecting with **SSH Client**, open command prompt in local system and change path for folder where your Linux instance key pair available in local system.

Then paste the instance path.



Instances (1/2) Info							
Last updated less than a minute ago <a href="#">Connect</a>							
Find Instance by attribute or tag (case-sensitive) All states							
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	linux_vpc_public	i-0298eec50297a3144	Running	t2.micro	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1a
<input checked="" type="checkbox"/>	linux_private	i-08732a249a920ef51	Running	t2.micro	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1a

To connect with the private linux instance need to use Private ip Address



Instance summary for i-08732a249a920ef51 (linux\_private) Info

Updated less than a minute ago

Instance ID: i-08732a249a920ef51

IPv6 address: -

Hostname type: -

Public IPv4 address: 3.239.51.228 | [open address](#)

Instance state: Running

Private IP DNS name (IPv4 only): -

Private IPv4 address copied: 10.0.1.177

Public IPv4 DNS: -

```
C:\Windows\System32\cmd.e x + v
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SOE III\Downloads>ssh -i "serverkey.pem" ec2-user@3.239.51.228
ssh: connect to host 3.239.51.228 port 22: Connection timed out

C:\Users\SOE III\Downloads>ssh -i "serverkey.pem" ec2-user@10.0.10.177
ssh: connect to host 10.0.10.177 port 22: Connection timed out

C:\Users\SOE III\Downloads>
```



## Connect to instance [info](#)

Connect to your instance i-08732a249a920ef51 (linux\_private) using any of these options

EC2 Instance Connect   Session Manager   **SSH client**   EC2 serial console

Instance ID  
i-08732a249a920ef51 (linux\_private)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is serverkey.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
chmod 400 "serverkey.pem"
4. Connect to your instance using its Public IP:  
3.239.51.228

Example:  
ssh -i "serverkey.pem" ec2-user@3.239.51.228

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

```
C:\Windows\System32\cmd.e  X  +  v
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

C:\Users\S0E III\Downloads>ssh -i "serverkey.pem" ec2-user@3.239.51.228
ssh: connect to host 3.239.51.228 port 22: Connection timed out

C:\Users\S0E III\Downloads>ssh -i "serverkey.pem" ec2-user@10.0.10.177
ssh: connect to host 10.0.10.177 port 22: Connection timed out

C:\Users\S0E III\Downloads>
```

With Both Public and Private IP addresses also we can't connect the instance from outside because it is launched in the Private Subnet.

## To connect the private instance

From **public instance** we can connect the private instance.

For that Key pair file should available in the Public instance.

To copy files from local system to Public Linux instance need to use **SCP** command:

## Steps to connect private instance

### In the public Linux Instance CMD:

Execute below commands:

➤ **Change user from ec2-user to root:**

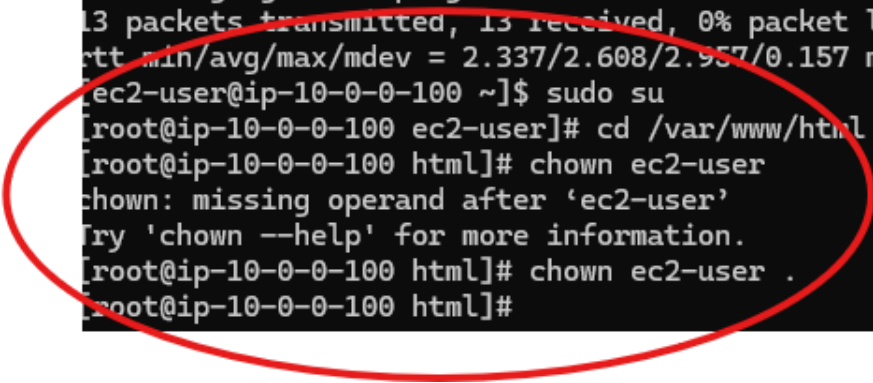
### Sudo Su

➤ **Open the html folder in public linux instace:**

**Cd /var/www/html**

➤ **Give owner permission to ec2-user**

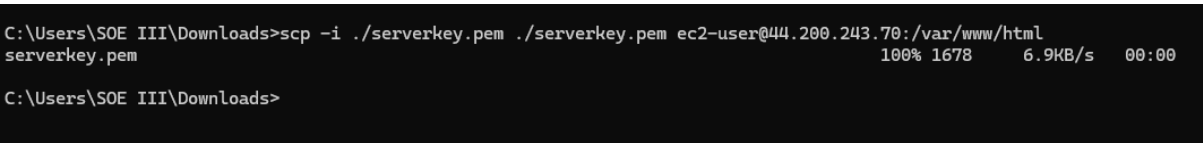
**Chown ec2-user .**



```
--- www.google.com ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12021ms
rtt min/avg/max/mdev = 2.337/2.608/2.957/0.157 ms
ec2-user@ip-10-0-0-100 ~]$ sudo su
root@ip-10-0-0-100 ec2-user]# cd /var/www/html
root@ip-10-0-0-100 html]# chown ec2-user
chown: missing operand after 'ec2-user'
Try 'chown --help' for more information.
root@ip-10-0-0-100 html]# chown ec2-user .
root@ip-10-0-0-100 html]#
```

**Open Cmd in Local System:**

- ✓ Open path where key pair file available in the command prompt
- ✓ Scp -i ./serverkey.pem ./serverkey.pem ec2-user@44.200.243.70 :/var/www/html



```
C:\Users\S0E III\Downloads>scp -i ./serverkey.pem ./serverkey.pem ec2-user@44.200.243.70:/var/www/html
serverkey.pem
100% 1678 6.9KB/s 00:00
C:\Users\S0E III\Downloads>
```

Successfully pem file transferred to public Linux instance

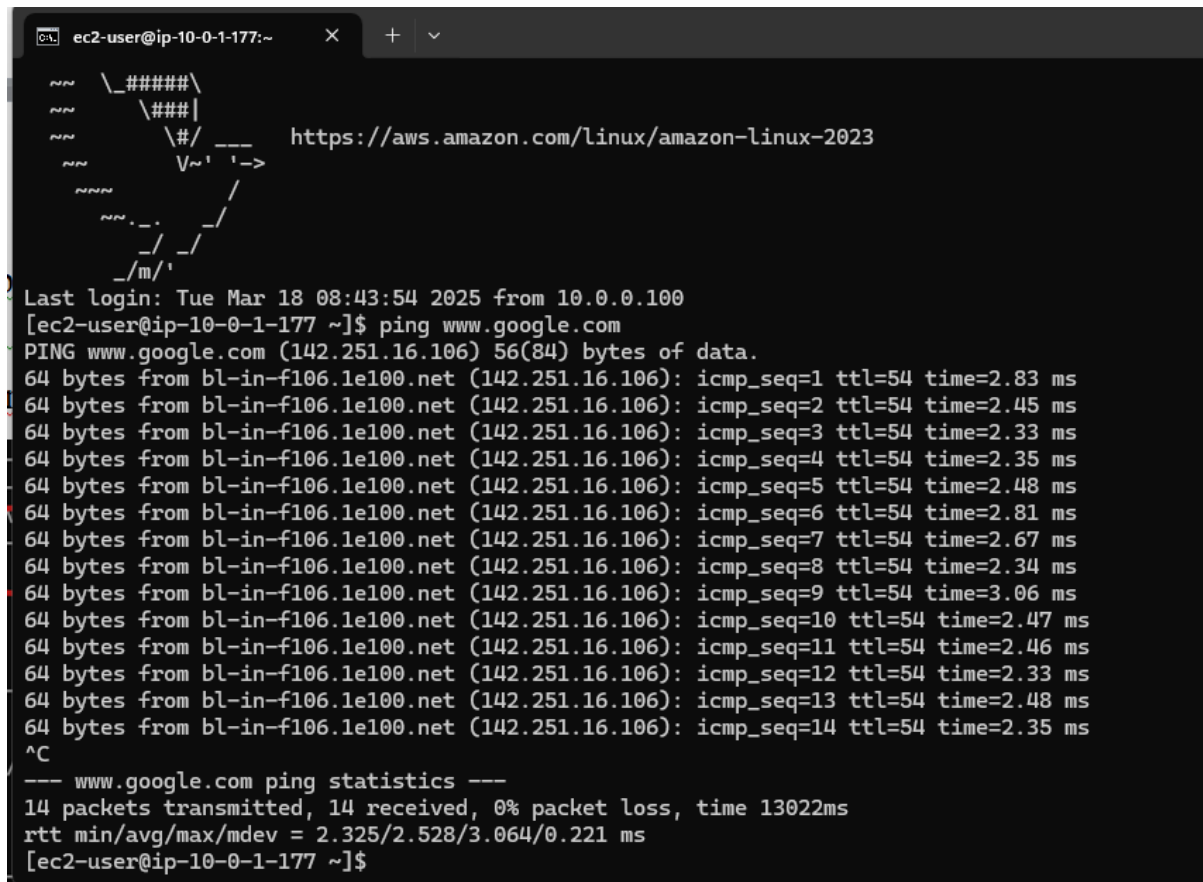
Check the file is available in public Linux instance

Use command in linux instance : **ls**



Check the internet connectivity in private linux instance

**Command:** ping [www.google.com](https://www.google.com)

A screenshot of a terminal window with a dark background. The window title is 'ec2-user@ip-10-0-1-177:~'. In the top left corner, there is a small ASCII art logo of a cat. The terminal shows the command 'ping www.google.com' and its output. The output indicates that 14 packets were transmitted and received with 0% packet loss. The round-trip time (rtt) statistics are: min/avg/max/mdev = 2.325/2.528/3.064/0.221 ms. The prompt '[ec2-user@ip-10-0-1-177 ~]\$' is visible at the bottom.

```
ec2-user@ip-10-0-1-177:~  
Last login: Tue Mar 18 08:43:54 2025 from 10.0.0.100  
[ec2-user@ip-10-0-1-177 ~]$ ping www.google.com  
PING www.google.com (142.251.16.106) 56(84) bytes of data:  
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=1 ttl=54 time=2.83 ms  
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=2 ttl=54 time=2.45 ms  
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=3 ttl=54 time=2.33 ms  
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=4 ttl=54 time=2.35 ms  
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=5 ttl=54 time=2.48 ms  
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=6 ttl=54 time=2.81 ms  
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=7 ttl=54 time=2.67 ms  
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=8 ttl=54 time=2.34 ms  
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=9 ttl=54 time=3.06 ms  
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=10 ttl=54 time=2.47 ms  
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=11 ttl=54 time=2.46 ms  
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=12 ttl=54 time=2.33 ms  
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=13 ttl=54 time=2.48 ms  
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=14 ttl=54 time=2.35 ms  
^C  
--- www.google.com ping statistics ---  
14 packets transmitted, 14 received, 0% packet loss, time 13022ms  
rtt min/avg/max/mdev = 2.325/2.528/3.064/0.221 ms  
[ec2-user@ip-10-0-1-177 ~]$
```

If data packets are transmitted without any loss then can understood that instance have internet connection.

With this we can say that Public Instance have internet connectivity.