

VPC-TASK2

1. Create one VPC, with 1 public subnet and 1 private subnet.

The screenshot shows the 'Create VPC' wizard in the AWS VPC service. The 'VPC settings' section is active, showing two options: 'VPC only' (selected) and 'VPC and more'. A 'Name tag - optional' field contains 'VPC_general'. Under 'IPv4 CIDR block', 'IPv4 CIDR manual input' is selected, and the value '192.168.0.0/16' is entered. The bottom of the screen includes standard AWS navigation links like CloudShell, Feedback, and Console Mobile App, along with copyright and legal information.

- Search for VPC and create VPC
- VPC setting → VPC only
- Select IPv4 CIDR manual input
- IPV4 CIDR as 192.168.0.0/16
- Click on create CIDR.

The screenshot shows the 'VPC dashboard' page. A success message at the top states 'You successfully created vpc-04f856db8114e4c37 / VPC_general'. The main area displays the details of the newly created VPC, including its ID, state (Available), and various configuration settings like DNS resolution, network ACL, and CIDR. The left sidebar shows a tree view of VPC resources like Subnets and Route tables. The bottom of the screen includes standard AWS navigation links and copyright information.

- It shows successfully created VPC.

VPC-TASK2

The screenshot shows the 'Edit subnet settings' page for a subnet named 'Private_general'. The 'Auto-assign IP settings' section has the 'Enable auto-assign public IPv4 address' checkbox unchecked. The 'Resource-based name (RBN) settings' section has the 'Enable resource name DNS A record on launch' checkbox unchecked.

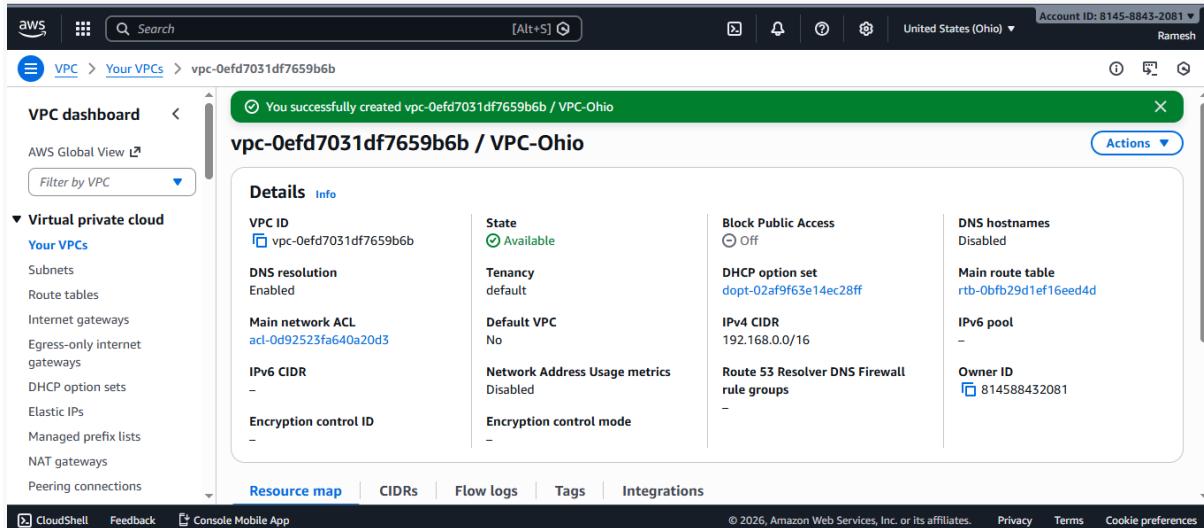
- Click on VPC, select on subnet create subnet as private subnet and public subnet.
- For private subnet we can should not enable auto-assign public IP4 address in edit subnet settings.
- CIDR Range should be within VPC range.

The screenshot shows the 'Edit subnet settings' page for a subnet named 'public_general'. The 'Auto-assign IP settings' section has the 'Enable auto-assign public IPv4 address' checkbox checked. The 'Resource-based name (RBN) settings' section has the 'Enable resource name DNS A record on launch' checkbox unchecked.

- Click on VPC, select on subnet create subnet as public subnet.
- For public subnet we can should enable auto-assign public IP4 address in edit subnet settings.
- CIDR Range should be within VPC range.

VPC-TASK2

2. Enable VPC peering for cross-region.

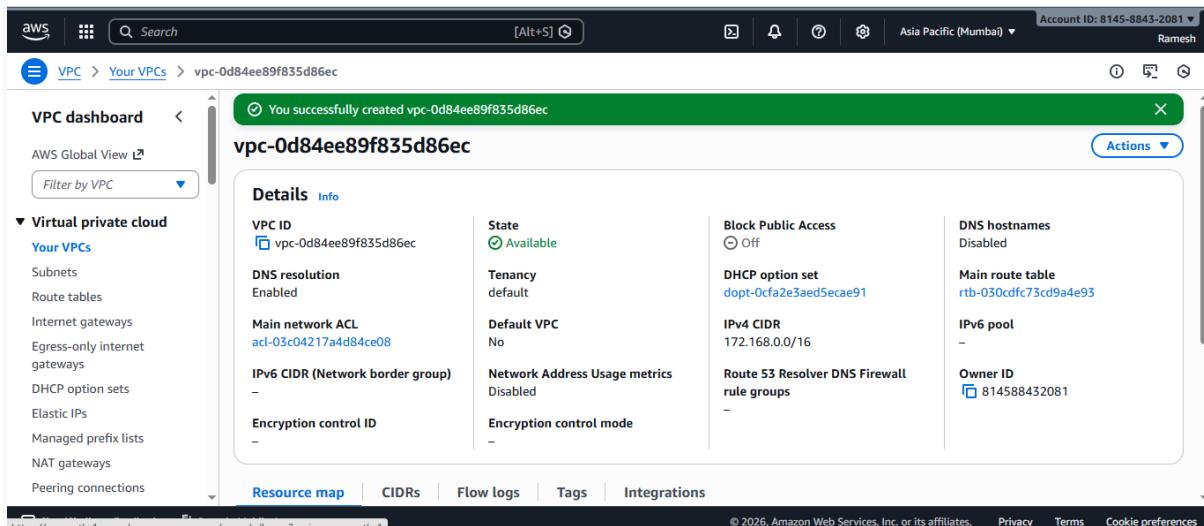


The screenshot shows the AWS VPC dashboard for the Ohio region. A green success message at the top states: "You successfully created vpc-0efd7031df7659b6b / VPC-Ohio". The main card displays the details of the newly created VPC:

Details		Info	
VPC ID	vpc-0efd7031df7659b6b	State	Available
DNS resolution	Enabled	Tenancy	default
Main network ACL	acl-0d92523fa640a20d3	Default VPC	No
IPv6 CIDR	-	Network Address Usage metrics	Disabled
Encryption control ID	-	Encryption control mode	-
		Block Public Access	Off
		DHCP option set	dopt-02af9f63e14ec28ff
		IPv4 CIDR	192.168.0.0/16
		Route 53 Resolver DNS Firewall rule groups	-
		DNS hostnames	Disabled
		Main route table	rtb-0fb29d1ef16eed4d
		IPv6 pool	-
		Owner ID	814588432081

Below the main card are tabs for Resource map, CIDRs, Flow logs, Tags, and Integrations. The bottom of the page includes links for CloudShell, Feedback, Console Mobile App, and standard footer links.

- Search for VPC, click on create VPC for ohio.
- Give IPV4 CIDR as manual input.
- Click on create VPC.



The screenshot shows the AWS VPC dashboard for the Mumbai region. A green success message at the top states: "You successfully created vpc-0d84ee89f835d86ec". The main card displays the details of the newly created VPC:

Details		Info	
VPC ID	vpc-0d84ee89f835d86ec	State	Available
DNS resolution	Enabled	Tenancy	default
Main network ACL	acl-03c04217a4d84ce08	Default VPC	No
IPv6 CIDR (Network border group)	-	Network Address Usage metrics	Disabled
Encryption control ID	-	Encryption control mode	-
		Block Public Access	Off
		DHCP option set	dopt-0cfa2e3ae05eca91
		IPv4 CIDR	172.168.0.0/16
		Route 53 Resolver DNS Firewall rule groups	-
		DNS hostnames	Disabled
		Main route table	rtb-030cdcf73cd9a4e93
		IPv6 pool	-
		Owner ID	814588432081

Below the main card are tabs for Resource map, CIDRs, Flow logs, Tags, and Integrations. The bottom of the page includes links for CloudShell, Feedback, Console Mobile App, and standard footer links.

- Search for VPC, click on create VPC for Mumbai Region.
- Give IPV4 CIDR as manual input.
- Click on create VPC.

VPC-TASK2

The screenshot shows the AWS VPC dashboard. On the left, there's a sidebar with 'VPC dashboard' and 'Virtual private cloud' sections. The main area is titled 'Your VPCs (1/2)' and shows a table with two rows:

Name	VPC ID	Status	Encryption c...	Encryption control ...
ohio	vpc-0ac272c235584af9	Available	-	-
Mumbai	vpc-0d84ee89f835d86ed	Available	-	-

- In Your VPCs we can see all VPCs what we have created.
- We can create two VPCs for VPC cross peering connection

The screenshot shows the 'Create peering connection' wizard. Step 1: 'Select a local VPC to peer with'. It asks for a 'Name - optional' (connect_mumbai) and shows a dropdown for 'VPC ID (Requester)' with 'vpc-0efd7031df7659b6b (VPC-Ohio)' selected. Below it, a table lists 'VPC CIDRs for vpc-0efd7031df7659b6b (VPC-Ohio)':

CIDR	Status	Status reason
192.168.0.0/16	Associated	-

- In VPC we have to search for Peering connections
- Click on Create peering connection
- Select VPC ID (requester)
- Select another VPC to peer with Account type as my account or another Account.

The screenshot shows the 'Create peering connection' wizard. Step 2: 'Select a peer VPC'. It shows 'Region' options ('This Region (us-east-2)' and 'Another Region') and a dropdown for 'VPC ID (Acceptor)' with 'vpc-0d84ee89f835d86ed' selected. Below it, there's a 'Tags' section:

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	connect_mumbai

Cancel Create peering connection

VPC-TASK2

- Select region and
- Acceptor ID-VPC-id
- Click on create on peer connection

The screenshot shows the AWS VPC Peering Connections dashboard. A green banner at the top indicates a peering connection request has been made. The main card displays the details of the peering connection:

pcx-0970d1062ab0a5c7e / connect_mumbai	
Details Info	
Requester owner ID	814588432081
Peering connection ID	pcx-0970d1062ab0a5c7e
Status	Initiating Request to 814588432081
Expiration time	Wednesday, January 14, 2026 at 01:31:10 GMT+5:30
Acceptor owner ID	814588432081
Requester VPC	vpc-0efd7031df7659b6b / VPC-Ohio
Requester CIDRs	192.168.0.0/16
Requester Region	Ohio (us-east-2)
VPC Peering connection ARN	arn:aws:ec2:us-east-2:814588432081:vpc-peering-connection/pcx-0970d1062ab0a5c7e
Acceptor VPC	vpc-0d84ee89f835d86ec
Acceptor CIDRs	-
Acceptor Region	Mumbai (ap-south-1)

- Above image shows peering requesting send to other region through peering connections.

The screenshot shows the AWS VPC Peering Connections dashboard. A blue banner at the top indicates pending acceptance for a peering connection request. The main card displays the details of the peering connection:

pcx-0970d1062ab0a5c7e	
Details Info	
Requester owner ID	814588432081
Peering connection ID	pcx-0970d1062ab0a5c7e
Status	Pending Acceptance by 814588432081
Expiration time	Wednesday, January 14, 2026 at 01:31:10 GMT+5:30
Acceptor owner ID	814588432081
Requester VPC	vpc-0efd7031df7659b6b
Requester CIDRs	192.168.0.0/16
Requester Region	Ohio (us-east-2)
VPC Peering connection ARN	arn:aws:ec2:ap-south-1:814588432081:vpc-peering-connection/pcx-0970d1062ab0a5c7e
Acceptor VPC	vpc-0d84ee89f835d86ec / Mumbai
Acceptor CIDRs	-
Acceptor Region	Mumbai (ap-south-1)

- We can notification of pending acceptance of peering connection as shown above.

VPC-TASK2

The screenshot shows the AWS VPC Peering Connections page. A specific peering connection, "pcx-0970d1062ab0a5c7e", is selected. The status is "Pending acceptance". The "Actions" menu is open, showing options: "Accept request", "Reject request", "Edit DNS settings", "Manage tags", and "Delete peering connection".

Details

Requester owner ID	Acceptor owner ID
814588432081	814588432081

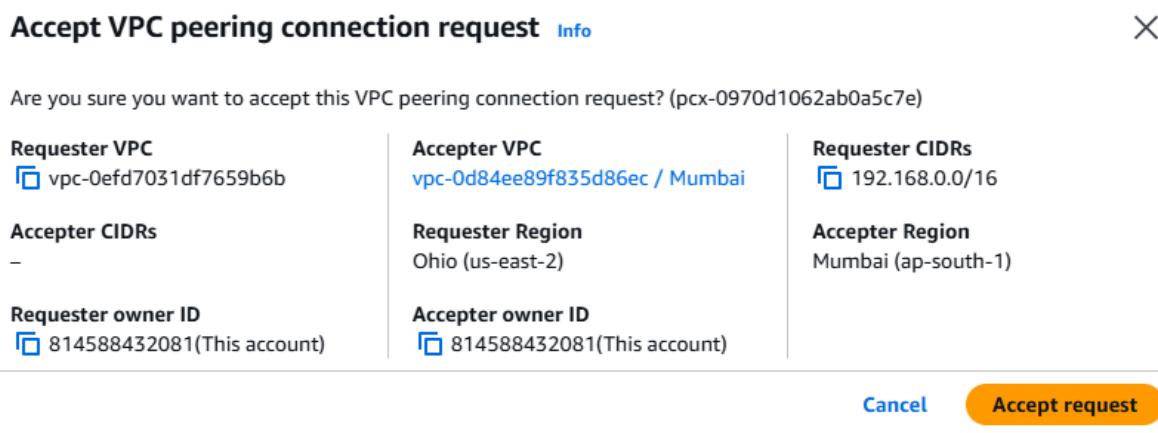
Peering connection ID	Requester VPC
pcx-0970d1062ab0a5c7e	vpc-0efd7031df7659b6b

Status	Requester CIDs	Requester Region	Acceptor VPC	Acceptor CIDs	Acceptor Region
Pending Acceptance by 814588432081	192.168.0.0/16	Ohio (us-east-2)	vpc-0d84ee89f835d86ec / Mumbai	-	Mumbai (ap-south-1)

Expiration time: Wednesday, January 14, 2026 at 01:31:10 GMT+5:30

DNS | **Route tables** | **Tags**

- Click on actions so that we can accept request and reject request.



- After clicking on accept request so that we can accept peering connection.

The screenshot shows the AWS VPC Peering Connections page. The peering connection "pcx-0970d1062ab0a5c7e" is now listed with a green status bar indicating it has been established. The "Actions" menu is open, showing options: "Modify my route tables now" and "Actions".

Details

Requester owner ID	Acceptor owner ID	VPC Peering connection ARN
814588432081	814588432081	arn:aws:ec2:ap-south-1:814588432081:vpc-peering-connection/pcx-0970d1062ab0a5c7e

Peering connection ID	Requester VPC	Acceptor VPC
pcx-0970d1062ab0a5c7e	vpc-0efd7031df7659b6b	vpc-0d84ee89f835d86ec / Mumbai

Status	Requester CIDs	Requester Region	Acceptor CIDs	Acceptor Region
Provisioning	192.168.0.0/16	Ohio (us-east-2)	172.168.0.0/16	Mumbai (ap-south-1)

Expiration time: -

- Above image say successful of peering connection established.

VPC-TASK2

The screenshot shows the AWS VPC Route Tables page. A green success message at the top reads: "Route table rtb-057db1ef3b020155f | Peering-ohio was created successfully." Below this, the route table details are shown: Route table ID is rtb-057db1ef3b020155f, Main is No, Owner ID is vpc-0efd7031df7659b6b | VPC-Ohio. The Routes tab is selected, showing one route entry: Destination 192.168.0.0/16, Target local, Status Active, Propagated No, and Route Origin Create Route Table.

- Click on Route table and create route table.

The screenshot shows the AWS VPC Route Tables page. A green success message at the top reads: "Updated routes for rtb-057db1ef3b020155f / Peering-ohio successfully." Below this, the route table details are shown: Route table ID is rtb-057db1ef3b020155f, Main is No, Owner ID is vpc-0efd7031df7659b6b | VPC-Ohio. The Routes tab is selected, showing two route entries: Destination 192.168.0.0/16, Target local, Status Active, Propagated No, and Route Origin Create Route Table.

- Click on edit routed and so that we can add peer connection id.

VPC-TASK2

← → ⚙ ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#EditRoutes:RouteTableId=rtb-07a9665576c376a9f

Account ID: 8145-8843-2081 Ramesh

aws Search [Alt+S] 🌐 🔍

Asia Pacific (Mumbai) ▾

VPC > Route tables > rtb-07a9665576c376a9f > Edit routes

Edit routes

Destination	Target	Status	Propagated	Route Origin
172.16.0.0/16	local	Active	No	CreateRouteTable
172.31.0.0/16	Peering Connection	Active	No	CreateRoute

Add route

Cancel Preview Save changes

CloudShell Feedback Console Mobile App

© 2026, Amazon Web Services, Inc. or its affiliates.

Privacy Terms Cookie preferences

- Here we can add peering connections as shown and save changes.

```
ec2-user@ip-172-31-34-220:~
```

```
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ ssh -i "ohio_keypair.pem" ec2-user@ec2-3-138-186-90.us-east-2.compute.amazonaws.com
ssh: connect to host ec2-3-138-186-90.us-east-2.compute.amazonaws.com port 22: Connection timed out

user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ ssh -i "ohio_keypair.pem" ec2-user@ec2-3-138-186-90.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-3-138-186-90.us-east-2.compute.amazonaws.com (3.138.186.90)' can't be established.
ED25519 key fingerprint is SHA256:fX+qpM9s5UaUCYPoXp7J2D8yvak72ok04Vkb/nq7XzS.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-138-186-90.us-east-2.compute.amazonaws.com' (ED25519) to the list of known hosts.

          _#_
  ,~\ _##_      Amazon Linux 2023
~~ \_\####\ \
~~   \##|      https://aws.amazon.com/linux/amazon-linux-2023
~~     '#/ __
~~     V~' .->
~~       / \
~~     .-. /
~~     / \
~~   / \
/_m/ ,\

[ec2-user@ip-172-31-34-220 ~]$ ping 172.31.34.220
PING 172.31.34.220 (172.31.34.220) 56(84) bytes of data.
64 bytes from 172.31.34.220: icmp_seq=1 ttl=127 time=0.025 ms
64 bytes from 172.31.34.220: icmp_seq=2 ttl=127 time=0.021 ms
64 bytes from 172.31.34.220: icmp_seq=3 ttl=127 time=0.033 ms
64 bytes from 172.31.34.220: icmp_seq=4 ttl=127 time=0.022 ms
64 bytes from 172.31.34.220: icmp_seq=5 ttl=127 time=0.025 ms
64 bytes from 172.31.34.220: icmp_seq=6 ttl=127 time=0.032 ms
64 bytes from 172.31.34.220: icmp_seq=7 ttl=127 time=0.032 ms
64 bytes from 172.31.34.220: icmp_seq=8 ttl=127 time=0.037 ms
64 bytes from 172.31.34.220: icmp_seq=9 ttl=127 time=0.024 ms
64 bytes from 172.31.34.220: icmp_seq=10 ttl=127 time=0.023 ms
64 bytes from 172.31.34.220: icmp_seq=11 ttl=127 time=0.023 ms
64 bytes from 172.31.34.220: icmp_seq=12 ttl=127 time=0.023 ms
```

- Connection of EC2 which is created before.
 - Checking of ping with private ip.

VPC-TASK2

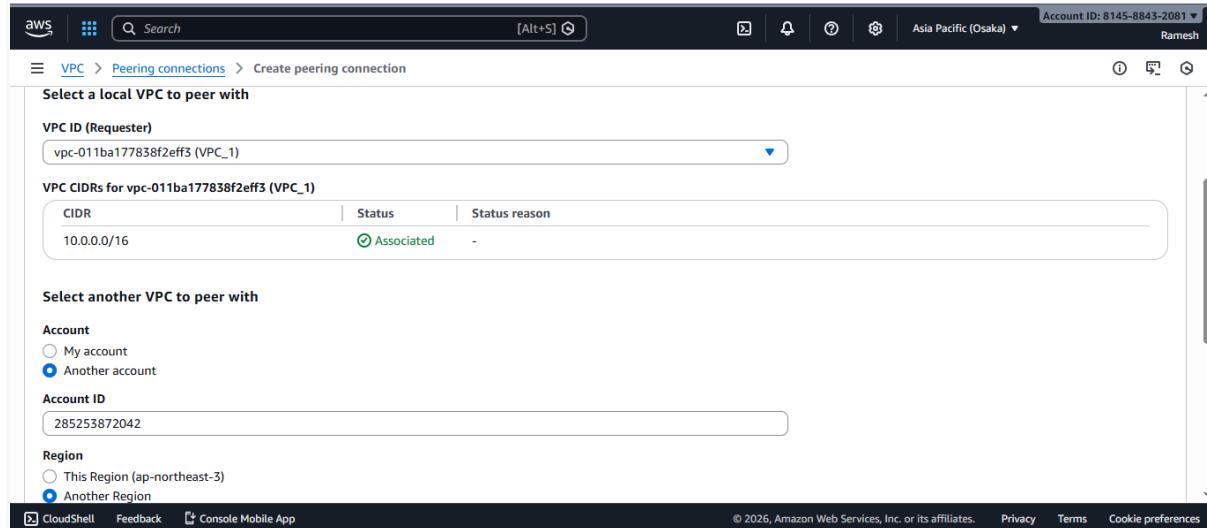
```
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
aws.com "Mumbai_keypair.pem" ec2-user@ec2-65-2-69-128.ap-south-1.compute.amazonaws
The authenticity of host 'ec2-65-2-69-128.ap-south-1.compute.amazonaws.com (65.2.69.128)' can't be established.
ED25519 key fingerprint is SHA256:KA8sF1hanODA03kVdg9V8Xa4wzDafpbelMTEiZ0nfLc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-65-2-69-128.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.

          #
          #####      Amazon Linux 2023
~~ \#####\
~~ \|##|
~~  \#/  ___ https://aws.amazon.com/linux/amazon-linux-2023
~~   V~' :->
~~   /
~~-. /-
~/m/'

[ec2-user@ip-172-31-4-29 ~]$ ping 172.31.4.29
PING 172.31.4.29 (172.31.4.29) 56(84) bytes of data.
64 bytes from 172.31.4.29: icmp_seq=1 ttl=127 time=0.022 ms
64 bytes from 172.31.4.29: icmp_seq=2 ttl=127 time=0.029 ms
64 bytes from 172.31.4.29: icmp_seq=3 ttl=127 time=0.029 ms
64 bytes from 172.31.4.29: icmp_seq=4 ttl=127 time=0.030 ms
64 bytes from 172.31.4.29: icmp_seq=5 ttl=127 time=0.029 ms
64 bytes from 172.31.4.29: icmp_seq=6 ttl=127 time=0.033 ms
64 bytes from 172.31.4.29: icmp_seq=7 ttl=127 time=0.036 ms
64 bytes from 172.31.4.29: icmp_seq=8 ttl=127 time=0.027 ms
64 bytes from 172.31.4.29: icmp_seq=9 ttl=127 time=0.027 ms
64 bytes from 172.31.4.29: icmp_seq=10 ttl=127 time=0.031 ms
64 bytes from 172.31.4.29: icmp_seq=11 ttl=127 time=0.031 ms
64 bytes from 172.31.4.29: icmp_seq=12 ttl=127 time=0.030 ms
64 bytes from 172.31.4.29: icmp_seq=13 ttl=127 time=0.031 ms
64 bytes from 172.31.4.29: icmp_seq=14 ttl=127 time=0.029 ms
64 bytes from 172.31.4.29: icmp_seq=15 ttl=127 time=0.027 ms
64 bytes from 172.31.4.29: icmp_seq=16 ttl=127 time=0.031 ms
64 bytes from 172.31.4.29: icmp_seq=17 ttl=127 time=0.028 ms
```

- Checking connection for both EC2.

3. Enable VPC peering for cross-account (you can collaborate with your friend to do this task).



- Select a local VPC to peer with VPC ID (requester).
 - Select another VPC to peer with account as another account
 - Fill account Id
 - Select Region as another region.

VPC-TASK2

The screenshot shows the AWS VPC Peering connections page. A green banner at the top states: "A VPC peering connection ppx-0364d2a435f8180c8 / Friend1_VPC has been requested. The owner of vpc-021d059b33d95eef5 must accept the peering connection." Below this, the "Peering connections (1) Info" section lists one connection:

Name	Peering connection ID	Status	Requester VPC	Acceptor VPC
Friend1_VPC	ppx-0364d2a435f8180c8	Pending acceptance	vpc-011ba177838f2eff3 / VPC_1	vpc-021d059b33d95eef5

Below the table, a message says "Select a peering connection above".

- Click on Create peering connection which is send to acceptor.

The screenshot shows the AWS VPC Peering connections page. A modal dialog titled "Accept VPC peering connection request" is open over the list of connections. The modal contains the following information:

Requester VPC	Acceptor VPC	Requester CIDRs	Acceptor CIDRs
vpc-011ba177838f2eff3	vpc-021d059b33d95eef5 / New-VPC-1	10.0.0.0/16	10.0.0.0/16

At the bottom of the modal, there are "Cancel" and "Accept request" buttons. The "Accept request" button is highlighted with a yellow background.

- From other account this shows for accept request

VPC-TASK2

The screenshot shows the AWS VPC Route Tables interface. In the 'Edit routes' section, there is a table with columns: Destination, Target, Status, Propagated, and Route Origin. One row has '10.0.0.0/16' as the destination, 'local' as the target (status is Active), and 'CreateRouteTable' as the route origin. Another row is being edited for '172.168.0.0/16', with 'Peering Connection' as the target (status is -) and 'pcx-0364d2a435f8180c8' as the route origin. There are 'Add route', 'Remove', 'Cancel', 'Preview', and 'Save changes' buttons at the bottom.

- Click on Route tables and edit route tables and edit routes
- Add friend CIDR and peering connection
- Click on Save changes

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Thu Jan  8 06:18:14 2026 from 15.168.105.162
[ec2-user@ip-10-0-232-74 ~]$ ping 178.168.0.4
PING 178.168.0.4 (178.168.0.4) 56(84) bytes of data.
64 bytes from 178.168.0.4: icmp_seq=1 ttl=44 time=273 ms
64 bytes from 178.168.0.4: icmp_seq=2 ttl=44 time=273 ms
64 bytes from 178.168.0.4: icmp_seq=3 ttl=44 time=273 ms
64 bytes from 178.168.0.4: icmp_seq=4 ttl=44 time=272 ms
64 bytes from 178.168.0.4: icmp_seq=5 ttl=44 time=273 ms
64 bytes from 178.168.0.4: icmp_seq=6 ttl=44 time=272 ms
64 bytes from 178.168.0.4: icmp_seq=7 ttl=44 time=273 ms
64 bytes from 178.168.0.4: icmp_seq=8 ttl=44 time=273 ms

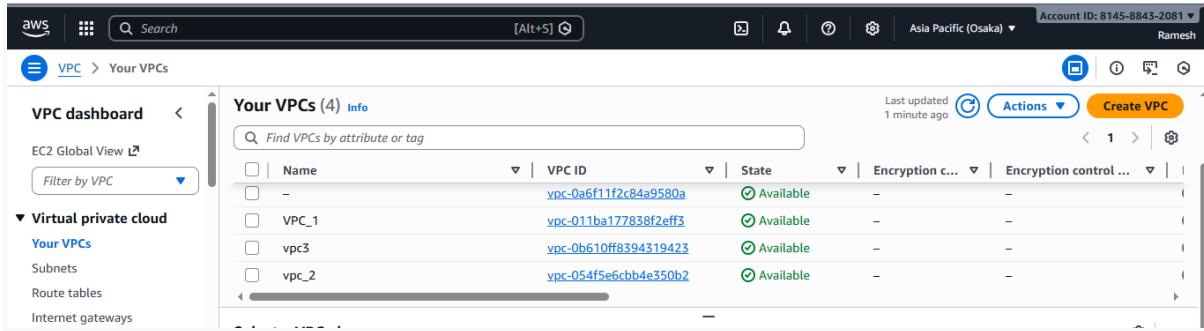
i-0a5486ac4cf7bb459 (ec2_vpc1)
PublicIPs: 15.168.164.148 PrivateIPs: 10.0.232.74
```

- Check ping ip address with friend ip and ping is connected.

VPC-TASK2

4. Set up a VPC Transit Gateway.

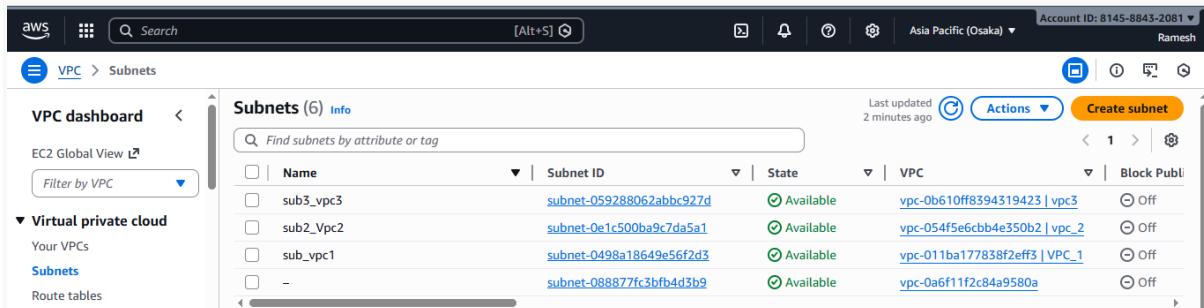
- Transit Gateway allows you to connect many VPCs and on-prem networks through a single gateway instead of managing many VPC peering connections.



The screenshot shows the AWS VPC dashboard with the title "Your VPCs". It lists four VPCs: "VPC_1", "vpc3", "vpc_2", and "vpc_1". Each entry includes the VPC ID, state (Available), and options for Actions and Encryption control.

Name	VPC ID	State	Actions	Encryption control
vpc_1	vpc-011ba177838f2eff3	Available		
vpc3	vpc-0b610ff8394319423	Available		
vpc_2	vpc-054f5e6ccb4e350b2	Available		

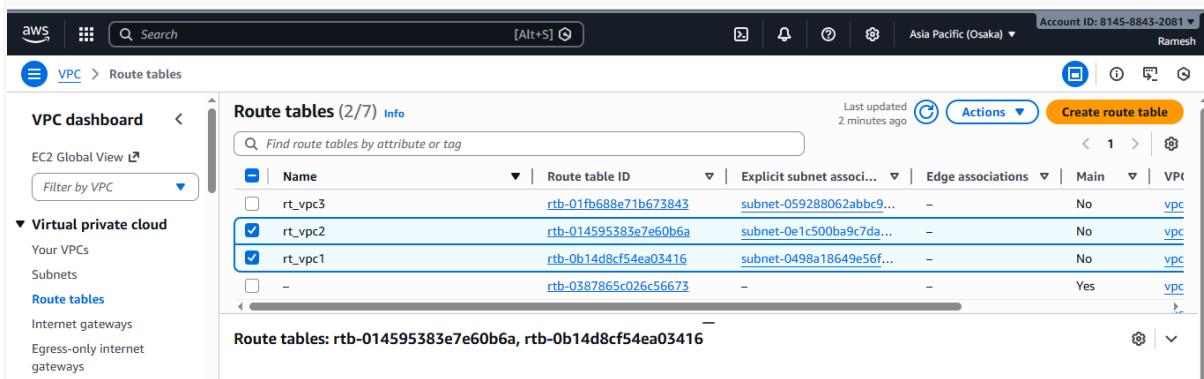
- Click on VPC and create VPC as per our requirement with CIDR range
- Click on your VPC and it shows VPCs present in it.



The screenshot shows the AWS Subnets page with the title "Subnets". It lists six subnets: "sub3_vpc3", "sub2_vpc2", "sub_vpc1", and three others. Each entry includes the Subnet ID, state (Available), VPC, and options for Actions and Block Public.

Name	Subnet ID	State	VPC	Actions	Block Public
sub3_vpc3	subnet-059288062abb927d	Available	vpc-0b610ff8394319423 vpc3		Off
sub2_vpc2	subnet-0e1c500ba9c7da5a1	Available	vpc-054f5e6ccb4e350b2 vpc_2		Off
sub_vpc1	subnet-0498a18649e56f2d3	Available	vpc-011ba177838f2eff3 VPC_1		Off
-	subnet-088877fc3bfb4d3b9	Available	vpc-0a6f11f2c84a9580a		Off

- Click on subnet and create subnet with in VPC CIDR as per our requirement.
- For that subnet we should add corresponding VPC



The screenshot shows the AWS Route tables page with the title "Route tables". It lists three route tables: "rt_vpc3", "rt_vpc2", and "rt_vpc1". Each entry includes the Route table ID, explicit subnet associations, edge associations, main status, and VPC. A note at the bottom states "Route tables: rtb-014595383e7e60b6a, rtb-0b14d8cf54ea03416".

Name	Route table ID	Explicit subnet assoc...	Edge associations	Main	VPC
rt_vpc3	rtb-01fb688e71b673843	subnet-059288062abb927d	-	No	vpc-0a6f11f2c84a9580a
rt_vpc2	rtb-014595383e7e60b6a	subnet-0e1c500ba9c7da5a1	-	No	vpc-054f5e6ccb4e350b2
rt_vpc1	rtb-0b14d8cf54ea03416	subnet-0498a18649e56f2d3	-	No	vpc-011ba177838f2eff3
-	rtb-0387865c026c56673	-	-	Yes	vpc-0a6f11f2c84a9580a

- Click on create on route table and add VPC to them.

VPC-TASK2

The screenshot shows the AWS VPC dashboard with the 'Internet gateways' section selected. There are four Internet gateways listed:

Name	Internet gateway ID	State	VPC ID
VPC1-IGW	igw-045b0b61ad73347c9	Attached	vpc-011ba177838f2eff3 VPC_1
vpc3_igw	igw-066fe3da44bcbe6e6	Attached	vpc-0b610ff8394319423 vpc3
vpc2_igw	igw-0c54a4ce357c3260c	Attached	vpc-054f5e6ccb4e350b2 vpc_2
-	igw-0e105d7d902b484a3	Attached	vpc-0a611f2c84a9580a

- Click on create Internet gateway for VPC present in our region.

The screenshot shows the AWS EC2 dashboard with the 'Instances' section selected. There are three instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
ec2_vpc3	i-089140accbeb0664f	Running	t2.micro	2/2 checks passed	View alarms +	ap-northeast-1
ec2_vpc1	i-0a5486ac4cf7bb459	Running	t2.micro	2/2 checks passed	View alarms +	ap-northeast-1
ec2_vpc2	i-0f5b23712455b1baf	Running	t3.micro	3/3 checks passed	View alarms +	ap-northeast-1

- Click on EC2 and click on instance and launch Instances as required.

The screenshot shows the AWS VPC dashboard with the 'Transit gateways' section selected. There is one transit gateway listed:

Name	Transit gateway ID	Owner ID	State
transit_gateway	tgw-0792864d88e9f1dea	814588432081	Available

Details for the transit gateway:

Transit gateway ID	State	Amazon ASN	DNS support
tgw-0792864d88e9f1dea	Available	64512	Enable

- Search for Transit gateways and create transit gateway

VPC-TASK2

The screenshot shows the AWS VPC console with the path: VPC > Transit gateway attachments. A success message at the top states: "You successfully created VPC attachment tgw-attach-025d758423d0f9547 / tg_attach_3." The main table lists three transit gateway attachments:

Name	Transit gateway attachment ID	Transit gateway ID	Status	VPC
tg_attach_3	tgw-attach-025d758423d0f9547	tgw-0792864d88e9f1dea	Pending	vp...
tg_attach_1	tgw-attach-0915d02e7ee4acbe7	tgw-0792864d88e9f1dea	Available	vp...
tg_attach_2	tgw-attach-0f19b3aa0dc3ee28	tgw-0792864d88e9f1dea	Pending	vp...

- Click on transit gateway attachments and create transit gateway attachment.

The screenshot shows the AWS VPC console with the path: VPC > Route tables > rtb-01fb688e71b673843 > Edit routes. The "Edit routes" section displays several route entries:

Destination	Target	Status	Propagated	Route Origin
12.0.0.0/16	local	Active	No	CreateRouteTable
10.0.0.0/16	Transit Gateway	Active	No	CreateRoute
11.0.0.0/16	Transit Gateway	Active	No	CreateRoute
0.0.0.0/0	Internet Gateway	Active	No	CreateRoute

Buttons for "Add route" and "Save changes" are visible at the bottom.

- In VPC and select Route tables and edit routes
- Add transit gateways CIDR and add internet gateway.
- Click on save changes.

VPC-TASK2

```
ec2-user@ip-12-0-1-103:~  
This host key is known by the following other names/addresses:  
~/ssh/known_hosts:81: 15.168.62.107  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '56.155.29.17' (ED25519) to the list of known hosts.  
#  
### Amazon Linux 2023  
####  
### https://aws.amazon.com/linux/amazon-linux-2023  
#  
V~ .-->  
-->  
-->  
/m/  
Last login: Wed Jan 7 21:08:50 2026 from 115.96.62.22  
[ec2-user@ip-12-0-1-103 ~]$ ping 12.0.1.103  
PING 12.0.1.103 (12.0.1.103) 56(84) bytes of data.  
64 bytes from 12.0.1.103: icmp_seq=1 ttl=127 time=0.016 ms  
64 bytes from 12.0.1.103: icmp_seq=2 ttl=127 time=0.025 ms  
64 bytes from 12.0.1.103: icmp_seq=3 ttl=127 time=0.026 ms  
64 bytes from 12.0.1.103: icmp_seq=4 ttl=127 time=0.026 ms  
^C  
--- 12.0.1.103 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3109ms  
rtt min/avg/max/mdev = 0.016/0.023/0.026/0.004 ms  
[ec2-user@ip-12-0-1-103 ~]$ ping 10.0.232.74  
PING 10.0.232.74 (10.0.232.74) 56(84) bytes of data.  
64 bytes from 10.0.232.74: icmp_seq=1 ttl=126 time=1.16 ms  
64 bytes from 10.0.232.74: icmp_seq=2 ttl=126 time=1.22 ms  
64 bytes from 10.0.232.74: icmp_seq=3 ttl=126 time=0.673 ms  
64 bytes from 10.0.232.74: icmp_seq=4 ttl=126 time=1.44 ms  
^C  
--- 10.0.232.74 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3024ms  
rtt min/avg/max/mdev = 0.673/1.121/1.441/0.279 ms  
[ec2-user@ip-12-0-1-103 ~]$ ping 11.0.1.21  
PING 11.0.1.21 (11.0.1.21) 56(84) bytes of data.  
64 bytes from 11.0.1.21: icmp_seq=1 ttl=126 time=2.18 ms  
64 bytes from 11.0.1.21: icmp_seq=2 ttl=126 time=0.891 ms  
64 bytes from 11.0.1.21: icmp_seq=3 ttl=126 time=1.23 ms  
64 bytes from 11.0.1.21: icmp_seq=4 ttl=126 time=0.758 ms  
^X64 bytes from 11.0.1.21: icmp_seq=5 ttl=126 time=0.847 ms  
^C  
--- 11.0.1.21 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 4037ms  
rtt min/avg/max/mdev = 0.758/1.181/2.181/0.524 ms  
[ec2-user@ip-12-0-1-103 ~]$ |
```

- Above image shows VPC transit gateway among all EC2.

VPC-TASK2

5. Set up a VPC Endpoint.

- An endpoint is a specific entry or exit point used to communicate with a service, application, or network.
Think of it as an address where requests are sent and responses are received.

The screenshot shows the 'Create endpoint' page in the AWS VPC console. The 'Endpoint settings' section is visible, showing a 'Name tag - optional' field containing 'ep_Neelima'. The 'Type' section is expanded, showing several options: 'AWS services' (selected), 'PrivateLink Ready partner services', 'AWS Marketplace services', 'EC2 Instance Connect Endpoint', 'Resources', and 'Service networks'. At the bottom of the page, there are links for CloudShell, Feedback, Console Mobile App, and a copyright notice for 2026.

- In VPC select Endpoints and
- Click on create on endpoint.
- Select type as AWS services.

The screenshot shows the 'Service Region' selection step in the 'Create endpoint' process. It shows the 'Enable Cross Region endpoint' checkbox is checked, and the region selected is 'Asia Pacific (Osaka) (ap-northeast-3)'. Below this, a message indicates 'Showing services available in service region: Asia Pacific (Osaka) (ap-northeast-3)'. A table titled 'Services (1/2)' lists one service: 'com.amazonaws.ap-northeast-3.s3' (Owner: amazon, Type: Interface). The bottom of the page includes standard AWS navigation links.

- Select Service Region
- In services select for service which is associate with the region.

VPC-TASK2

The screenshot shows the 'Create endpoint' page in the AWS VPC service. In the 'Network settings' section, the VPC dropdown is set to 'vpc-006e2888d698c709e (Endpoint_VPC)'. Below it, under 'Additional settings', there is a 'Route tables (2)' section. It lists two route tables: 'rtb-092127b36b118bfe3' (Main) and 'rtb-0cc7e9a7e6343e4dc (EP_rt)' (No). Both route tables are associated with the same 'Associated Id'.

Name	Route Table ID	Main	Associated Id
-	rtb-092127b36b118bfe3	Yes	-
EP_rt	rtb-0cc7e9a7e6343e4dc (EP_rt)	No	-

- In Networking setting Select VPC which is created for VPC endpoints.
- Select Route tables which are created for endpoints.

The screenshot shows the 'Create endpoint' page. In the 'Route tables (1/2)' section, the 'rtb-0cc7e9a7e6343e4dc (EP_rt)' route table is selected. A note below explains that using private IP addresses for endpoints ensures security. At the bottom of the page, the 'Policy' section is visible, showing 'Full access' selected.

When you use an endpoint, the source IP addresses from your instances in your affected subnets for accessing the AWS service in the same region will be private IP addresses, not public IP addresses. Existing connections from your affected subnets to the AWS service that use public IP addresses may be dropped. Ensure that you don't have critical tasks running when you create or modify an endpoint.

Full access

Allow access by any user or service within the VPC using credentials from any Amazon Web Services accounts to any resources in this Amazon Web Services service. All policies — IAM user policies, VPC endpoint policies, and Amazon Web Services service-specific policies (e.g. Amazon S3 bucket policies, any S3 ACL policies) — must grant the necessary permissions for access to succeed.

- Click on policy and select full access.
- Save it.

VPC-TASK2

The screenshot shows the AWS VPC Endpoints console. A green success message at the top states "Successfully created VPC endpoint vpce-0c213ee3185b334f7". Below it, a table titled "Endpoints (1/1)" lists one endpoint named "ep_Neelima" with ID "vpce-0c213ee3185b334f7" and status "Available". The table has columns for Name, VPC endpoint ID, Endpoint type, and Status. At the bottom, there's a "Details" section for the endpoint.

- Above image shows endpoints created successfully

The screenshot shows the AWS Subnets console. A green success message at the top states "You have successfully created 2 subnets: subnet-0b86a5cccd44191be, subnet-0d812caa618c7b28a". Below it, a table titled "Subnets (2)" lists two subnets: "pub_sub_ep" and "prvt_sub_Endpoint". The table has columns for Name, Subnet ID, State, VPC, and Block Public. Both subnets are listed as "Available".

- Click on subnet and create subnets with VPC which is associate with it

The screenshot shows the AWS Internet Gateways console. A green success message at the top states "Internet gateway igw-017599eeba4a0dc7d successfully attached to vpc-006e2888d698c709e". Below it, a table titled "igw-017599eeba4a0dc7d / Endpoint_IGW" shows details for the internet gateway, including its ID, state (Attached), VPC ID, and owner. It also displays a "Tags (1)" section with a single tag: "Name" with value "Endpoint_IGW".

- Click on Internet gateway and create IGW and route edit route and internet gateway to public subnet.

VPC-TASK2

```
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ scp -i osakakeypair.pem osakakeypair.pem ec2-user@56.155.87.114:keys/
osakakeypair.pem                                         100% 1678      8.0KB/s   00:00

user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ ssh -i osakakeypair.pem ec2-user@56.155.87.114
,          #
~\_      #####
~~ \####\           Amazon Linux 2023
~~ \###|
~~  \|#/ __ https://aws.amazon.com/linux/amazon-linux-2023
~~    V~, .->
~~     /
~~.-
~/ -/
~/m/

Last login: Thu Jan  8 11:15:42 2026 from 103.203.172.230
[ec2-user@ip-14-0-1-11 ~]$ mkdir -p ~/keys
[ec2-user@ip-14-0-1-11 ~]$ ssh -i osakakeypair.pem ec2-user@56.155.87.114
oooooooooooooooooooooooooooooooooooooooooooooooooooo
@           WARNING: UNPROTECTED PRIVATE KEY FILE!          @
oooooooooooooooooooooooooooooooooooooooooooooooooooo
Permissions 0444 for 'osakakeypair.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "osakakeypair.pem": bad permissions
ec2-user@56.155.87.114: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-14-0-1-11 ~]$ chmod 400 ~/keys/osakakeypair.pem
[ec2-user@ip-14-0-1-11 ~]$ ls -l ~/keys/osakakeypair.pem
-r-----. 1 ec2-user ec2-user 1678 Jan  8 11:19 /home/ec2-user/keys/osakakeypair.pem
[ec2-user@ip-14-0-1-11 ~]$ ssh -i ~/keys/osakakeypair.pem ec2-user@10.0.2.4

^C
[ec2-user@ip-14-0-1-11 ~]$ ssh -i ~/keys/osakakeypair.pem ec2-user@10.0.2.4
```

- Connect to ec2 instance which created for endpoints.

```
root@ip-14-0-1-11:~
[ec2-user@ip-14-0-1-11 ~]$ sudo su -
[root@ip-14-0-1-11 ~]# ping 14.0.1.11
PING 14.0.1.11 (14.0.1.11) 56(84) bytes of data.
64 bytes from 14.0.1.11: icmp_seq=1 ttl=127 time=0.015 ms
64 bytes from 14.0.1.11: icmp_seq=2 ttl=127 time=0.027 ms
64 bytes from 14.0.1.11: icmp_seq=3 ttl=127 time=0.025 ms
64 bytes from 14.0.1.11: icmp_seq=4 ttl=127 time=0.026 ms
^C
--- 14.0.1.11 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3128ms
rtt min/avg/max/mdev = 0.015/0.023/0.027/0.004 ms
[root@ip-14-0-1-11 ~]# ping 14.0.2.4
PING 14.0.2.4 (14.0.2.4) 56(84) bytes of data.
64 bytes from 14.0.2.4: icmp_seq=1 ttl=127 time=0.865 ms
64 bytes from 14.0.2.4: icmp_seq=2 ttl=127 time=0.890 ms
64 bytes from 14.0.2.4: icmp_seq=3 ttl=127 time=1.38 ms
64 bytes from 14.0.2.4: icmp_seq=4 ttl=127 time=1.04 ms
64 bytes from 14.0.2.4: icmp_seq=5 ttl=127 time=1.47 ms
^C
--- 14.0.2.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4079ms
rtt min/avg/max/mdev = 0.865/1.128/1.469/0.249 ms
[root@ip-14-0-1-11 ~]# |
```

VPC-TASK2

- Check ping connection for private ip with public subnet ip.

```
[root@ip-14-0-1-11 ~]# aws configure
AWS Access Key ID [*****KPRI]: AKIA33KKBB3I7ZCCKPRI
AWS Secret Access Key [*****H6Ph]: 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
Default region name [text]: ap-south-1
Default output format [text]: text
[root@ip-14-0-1-11 ~]# aws s3 ls
2025-10-15 05:45:47 aws-athena-query-results-814588432081-us-east-2-0pf98ayv
2025-09-10 18:25:51 aws-cloudtrail-logs-814588432081-0a7db287
2025-08-15 15:11:11 demo-wrerwe
2025-11-03 03:45:13 dummy-buck516
2025-08-14 10:18:05 josh-1-2
2025-08-11 14:43:06 kavya54321
2025-08-12 16:54:23 kvk24
2025-10-15 04:08:25 nam-etl-516
2025-11-10 05:46:40 nfs-data12345
2025-08-13 14:49:28 s3-life-cycle1
2026-01-06 19:09:30 s3neelima
2025-11-03 03:41:53 venkat-516
2025-10-10 19:38:28 venkey-s3-516
[root@ip-14-0-1-11 ~]# |
```

To access AWS first we have to configure through IAM

- \$aws configure
- AWS access key ID:
- AWS Secret Access ID:
- Default region name:
- Default output format:
- Access one of the service like S3 service through it
- AWS s3 ls it shows all files present it.