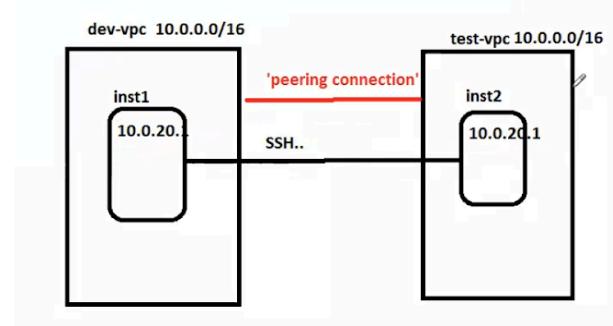


AWS Peering Connection

The diagram depicts two VPCs, namely 'dev-vpc' and 'test-vpc.' Each VPC contains a respective instance: 'dev-vpc' has an instance named 'inst1,' and 'test-vpc' has an instance named 'inst2.' The objective is to establish a connection between 'inst1' and 'inst2.'



But we have limitation

if we perform ssh connection between these two instances, it won't be possible.

We can not change the CIDR, once the VPCs are created

So the solution for this problem is **peering connection**.

Peering connections in the context of AWS (Amazon Web Services) refer to the establishment of network connections between two separate VPCs (Virtual Private Clouds) or between a VPC and an on-premises network. This connectivity is achieved through the AWS Peering feature, and it serves several important needs:

- 1. **Inter-VPC Communication:**** Peering connections enable secure and direct communication between different VPCs within the same or different AWS regions. This is useful when you have multiple VPCs for different workloads or teams, and you need them to communicate with each other.
- 2. **Resource Sharing:**** Peering allows resources in one VPC to access resources in another VPC. This is particularly beneficial when you want to share data, services, or applications across different VPCs while maintaining isolation and security.
- 3. **Cost Savings:**** Instead of using the public internet or dedicated connections, peering connections provide a cost-effective way for VPCs to communicate with each other. Data transfer between peered VPCs is typically faster and doesn't incur additional data transfer costs.
- 4. **Simplified Network Architecture:**** Peering connections simplify the network architecture by allowing VPCs to communicate directly without the need for complex routing configurations. This ease of communication can be crucial in scenarios where services in different VPCs need to interact seamlessly.

5. **Hybrid Cloud Connectivity:** Peering connections can be used to establish connectivity between an AWS VPC and an on-premises data centre. This is especially useful in hybrid cloud architectures where certain workloads or services reside on-premises while others are hosted in the AWS cloud.

6. **Improved Security and Control:** Peering connections can be configured with route tables and security groups to control and secure traffic between the peered VPCs. This helps in implementing fine-grained access controls and ensures that only authorised communication occurs between VPCs.

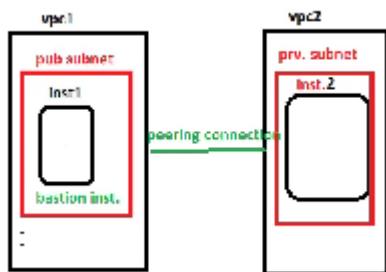
7. **Global Reach:** AWS peering connections can be established not only within the same region but also across different AWS regions, allowing for global communication between VPCs.

In summary, peering connections in AWS facilitate efficient, secure, and cost-effective communication between different VPCs or between a VPC and an on-premises network, contributing to a more flexible and scalable cloud infrastructure.

=====

Questions

Q1 How we can connect private instances to public instances



We have 2 virtual private clouds. vpc1 and vpc2 .

vpc1 has a public subnet and inside it, there is instance1, vpc2 has a private subnet and inside it, there is instance2. We want to connect vpc1 and vpc2.

->Using a peering connection.

Q2 What is a peering connection?

-> A peering connection in AWS is like a virtual bridge that connects two separate virtual networks. It allows these networks to communicate with each other as if they were part of the same network, even though they are distinct and separate entities. This connection is private, which means that the data exchanged between the networks doesn't travel over the public internet, ensuring security and better performance.

Q3 Bastion instance?

-> A Bastion instance, also known as a Bastion host or a Jump box, acts as an intermediary server between your local machine and the private instances within your VPC. It provides a secure point of entry into the private network. You can connect to the Bastion instance over SSH (Secure Shell) from your local machine, and from there, you can use the Bastion instance to access other instances within the private network.

simple example

****Bastion Host (or Bastion Instance):****

A Bastion host is a server that acts as an intermediary or a secure gateway between clients (such as system administrators) and private servers or resources within a network. The Bastion host is usually placed in a DMZ (Demilitarized Zone), a network segment that is isolated from the internal network and the internet. Its primary purpose is to provide a controlled and monitored entry point for accessing internal resources.

****Use Case Example: Secure Remote Access to Private Instances in AWS:****

Let's consider an example within the context of AWS:

1. **Network Setup:**

- You have a Virtual Private Cloud (VPC) in AWS, and within this VPC, there are private instances (e.g., databases, application servers) that should not be directly accessible from the internet.

- You also have a Bastion instance placed in a public subnet in the same VPC. This Bastion instance is the only server exposed to the internet.

2. **Accessing Private Instances:**

- System administrators or authorised users connect to the Bastion instance using SSH or another secure protocol.
- Once connected to the Bastion instance, they can then use it as a jumping-off point to access the private instances within the VPC.

3. **Security Measures:**

- The Bastion host is hardened and configured with strong security measures, including strict access controls, monitoring, and logging.
- Access to the Bastion host may be restricted using key pairs, multi-factor authentication, or other secure authentication methods.

4. **Bastion as a Secure Gateway:**

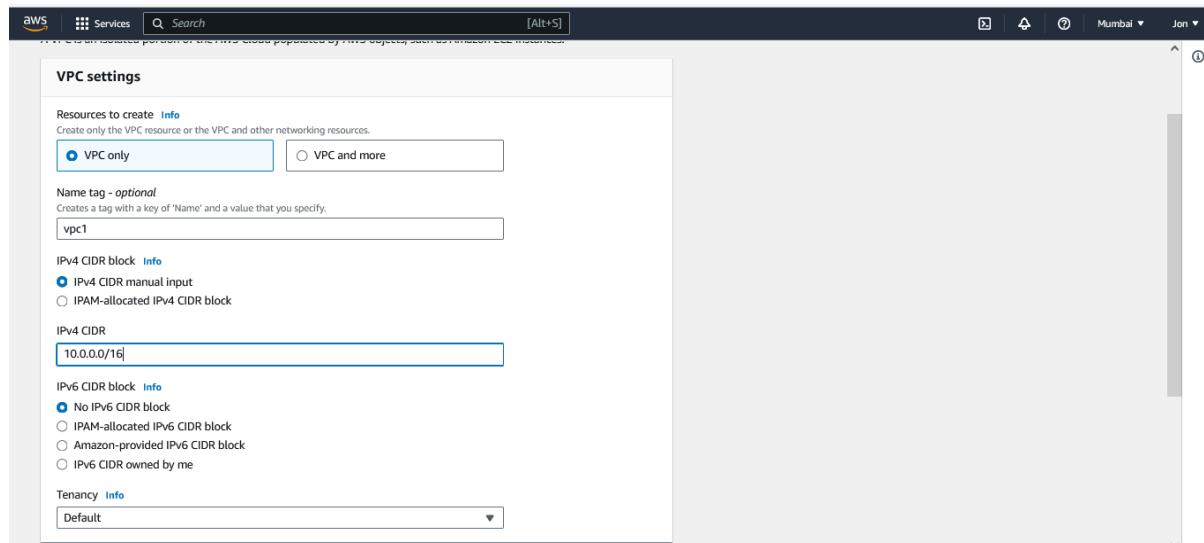
- The Bastion instance acts as a secure gateway, allowing administrators to access internal resources without exposing them directly to the internet.
- After connecting to the Bastion instance, administrators can use it to SSH or RDP into the private instances.

Benefits:

- Enhanced security: Private instances are not directly exposed to the internet, reducing the attack surface.
- Centralised access control: Access to internal resources can be tightly controlled and monitored through the Bastion host.
- Auditing and logging: All access to internal resources can be logged and audited through the Bastion instance.

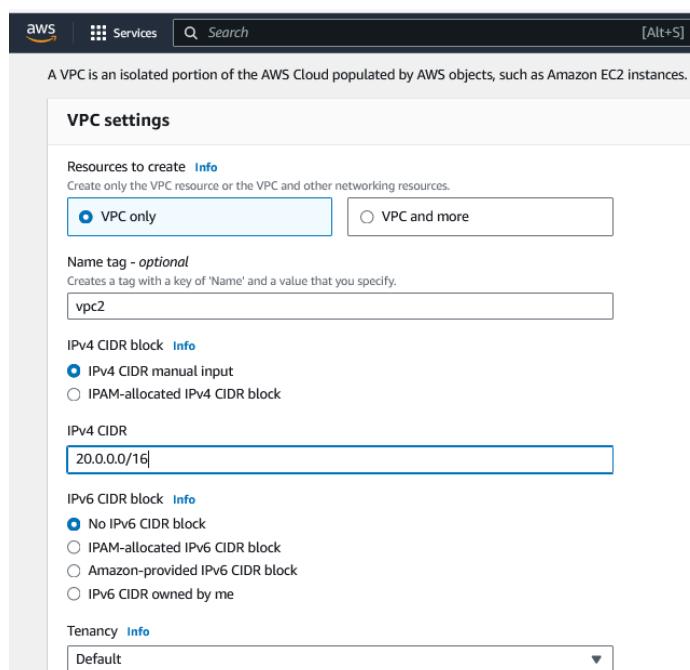
In summary, a Bastion instance serves as a secure entry point for accessing private resources within a network, and it helps maintain a more secure and controlled environment.

Step 1:-Create vpc1



The screenshot shows the 'VPC settings' page for creating a new VPC. The 'Resources to create' section is set to 'VPC only'. The 'Name tag - optional' field contains 'vpc1'. Under 'IPv4 CIDR block', 'IPv4 CIDR manual input' is selected, and the CIDR block '10.0.0.0/16' is entered. The 'IPv6 CIDR block' section shows 'No IPv6 CIDR block' selected. The 'Tenancy' dropdown is set to 'Default'. A 'Tags' section is visible at the bottom, showing a single tag 'Name: vpc1'.

Step 2:- Create vpc2



The screenshot shows the 'VPC settings' page for creating a new VPC. The 'Resources to create' section is set to 'VPC only'. The 'Name tag - optional' field contains 'vpc2'. Under 'IPv4 CIDR block', 'IPv4 CIDR manual input' is selected, and the CIDR block '20.0.0.0/16' is entered. The 'IPv6 CIDR block' section shows 'No IPv6 CIDR block' selected. The 'Tenancy' dropdown is set to '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
<input type="text" value="Name"/> <input type="button" value="X"/>	<input type="text" value="vpc2"/> <input type="button" value="X"/>
<input type="button" value="Remove tag"/>	
<input type="button" value="Add tag"/>	

You can add 49 more tags

For VPC1

Step 3:- Create subnet for vpc1

VPC > [Subnets](#) > Create subnet

Create subnet [Info](#)

VPC

VPC ID
Create subnets in this VPC.

Associated VPC CIDRs
IPv4 CIDRs

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 CIDR block [Info](#)

Tags - optional

Key	Value - optional
<input type="text" value="Name"/> <input type="button" value="X"/>	<input type="text" value="vpc1-pub-subnet"/> <input type="button" value="X"/>
<input type="button" value="Remove"/>	
<input type="button" value="Add new tag"/>	

You can add 49 more tags.

Step 4 :- Create a internet gateway for vpc1 as it is a public vpc

aws Services Search [Alt+S]

Create internet gateway Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

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

vpc1-pub-igw

Tags - optional
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="vpc1-pub-igw"/>

Add new tag
You can add 49 more tags.

Create internet gateway

Step 5:- Attach internet gateway to vpc1

Actions -> Attach to VPC

aws Services Search [Alt+S]

The following internet gateway was created: igw-0ade2658bb0aba6aa. You can now attach to a VPC to enable the VPC to communicate with the internet. [Attach to a VPC](#)

VPC dashboard X
EC2 Global View New

Filter by VPC: [Select a VPC](#)

Virtual private cloud
Your VPCs New
Subnets
Route tables
Internet gateways
Egress-only internet gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services

[VPC](#) > [Internet gateways](#) > igw-0ade2658bb0aba6aa

igw-0ade2658bb0aba6aa / vpc1-pub-igw

Details <small>Info</small>		Actions <small>▲</small>	
Internet gateway ID	igw-0ade2658bb0aba6aa	State	Detached
		VPC ID	-
		Owner	115185172188

Tags

Manage tags	
<input type="text" value="Search tags"/>	
Key	Value
Name	vpc1-pub-igw

Select vpc1

VPC > [Internet gateways](#) > Attach to VPC (igw-0ade2658bb0aba6aa)

Attach to VPC (igw-0ade2658bb0aba6aa) [Info](#)

VPC
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs
Attach the internet gateway to this VPC.

[X](#)

▶ AWS Command Line Interface command

[Cancel](#) [Attach internet gateway](#)

Step 6:- Create route table for **vpc1**

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.
 [X](#)

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 - <i>optional</i>
<input type="text" value="Name"/> X	<input type="text" value="vpc1-pub-rwt"/> X Remove

[Add new tag](#)

You can add 49 more tags.

[Cancel](#) [Create route table](#)

Step 7:- **vpc1** subnet association

Subnet association -> Edit subnet association

select vpc1-pub-subnet -> Click Save association

Step 8 :- Edit routes

Route tables ->Select vpc1->Routes->Edit routes

Target -> Internet Gateway

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	igw-0ade2658bb0aba6aa	-	No

Add route

Cancel **Preview** **Save changes**

Step 9:- Nacl for vpc1

Security -> nacl -> Create nacl

Network ACL settings

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

VPC
VPC to use for this network ACL.

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 - <i>optional</i>
<input type="text" value="Name"/>	<input type="text" value="vpc1-pub-nacl"/>

Add tag
You can add 49 more tags

Cancel **Create network ACL**

Step 10 :- Edit inbound rules for nacl of vpc1

Security -> nacl -> Select vpc1 -> Inbound rules -> Edit inbound rules

Step 11 :- Create a security group for vpc1

Security -> Security groups -> Create security group ->

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
SSH	TCP	22	Anywh...	0.0.0.0/0

Add rule Delete

For VPC2

Step 1:- Create subnet for vpc2 (Private subnet)

Subnets (8)

Name	Subnet ID	State	VPC	IPv4 CIDR
-	subnet-0941d6d773441c43a	Available	vpc-0f8f0da2d5913c923	172.31.16.0/20
adhi-vpc-subnet-pv	subnet-07ee4c40b71ecb2e4	Available	vpc-00784dd42f224120d adhi...	20.0.11.0/24
sakshi-vpc-subnet-pv	subnet-0ce5679e9989202a6	Available	vpc-0c0927f14731ea0c9 saks...	20.0.11.0/24

Create subnet

VPC ID
Create subnets in this VPC.
vpc-08cfb80918a46bbc6 (vpc2)

AWS Services Search [Alt+S]

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
vpc2-prv-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.
Asia Pacific (Mumbai) / ap-south-1a

IPv4 CIDR block [Info](#)
20.0.0.0/24

Tags - optional

Key	Value - optional
Q Name	Q vpc2-prv-subnet

[Add new tag](#)
You can add 49 more tags.
[Remove](#)

[Add new subnet](#)

Cancel [Create subnet](#)

Step 2:- NAT gateway

AWS Services Search [Alt+S]

NAT gateways (2) [Info](#)

Filter NAT gateways

Name	NAT gateway ID	Connectivity	State	Primary public IP	Primary
sakshi-vpc-nat-gate...	nat-0735c5fbe173ca9ab	Public	Available	3.6.212.107	20.0.11...
adhi-vpc-nat-gateway	nat-031701a0922bd1e81	Public	Available	13.232.223.160	20.0.11...

Actions [Create NAT gateway](#)

AWS Services Search [Alt+S]

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.
vpc2-prv-natway

The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.
subnet-053756ab8eabf75ea (vpc2-prv-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.
eipalloc-0ebebd13ec7bef060

[Allocate Elastic IP](#)

Additional settings [Info](#)

aws Services Search [Alt+S]

Select a subnet in which to create the NAT gateway.

subnet-053756ab8abf75ea (vpc2-prv-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.

eipalloc-0ebcd13ec7bef060 [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	Value - optional
<input type="text"/> Name	<input type="text"/> vpc2-prv-natway

[Add new tag](#)
You can add 49 more tags.

Cancel [Create NAT gateway](#)

Step 3:- Route table

aws Services Search [Alt+S]

VPC dashboard [X](#)

EC2 Global View [New](#)

Filter by VPC: [Select a VPC](#)

Virtual private cloud

Your VPCs [New](#)

Subnets

Route tables [Route tables](#)

Internet gateways

Route tables (10) [Info](#)

Find resources by attribute or tag

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
adhi-vpc-subnet-pv-rtb	rtb-051e6282f6c5fcc58	subnet-07ee4c40b71ec...	-	No	vpc-00784dd42f2241
-	rtb-07a31a1702c682190	-	-	Yes	vpc-00784dd42f2241
-	rtb-0bb4879711026ebc9	-	-	Yes	vpc-08cfb80918a46b
sakshi-vpc-subnet-pv-rtb	rtb-038f4c972a759b632	-	-	No	vpc-0c0927f14731ea
-	rtb-0dd0fff7e967983e	-	-	Yes	vpc-00a60c42f9d857
adhi-vpc-subnet-pub-rtb	rtb-0717f911607e6d75e	subnet-09dc4614ec1993...	-	No	vpc-00784dd42f2241

aws Services Search [Alt+S]

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.

vpc2-prv-rtb

VPC
The VPC to use for this route table.

vpc-08cfb80918a46bbc6 (vpc2)

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.

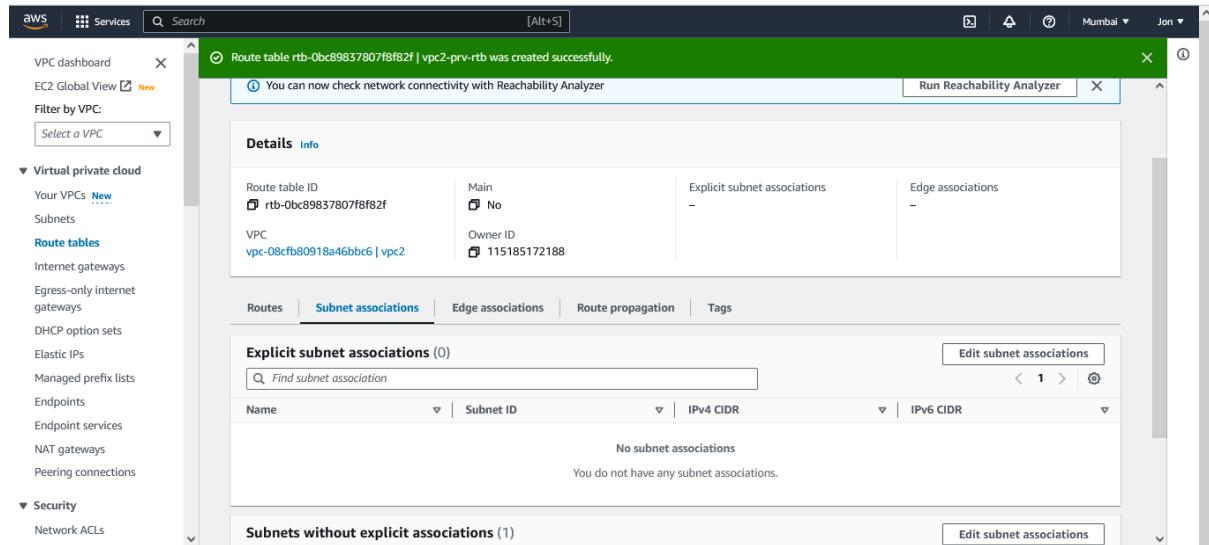
Key	Value - optional
<input type="text"/> Name	<input type="text"/> vpc2-prv-rtb

[Add new tag](#)
You can add 49 more tags.

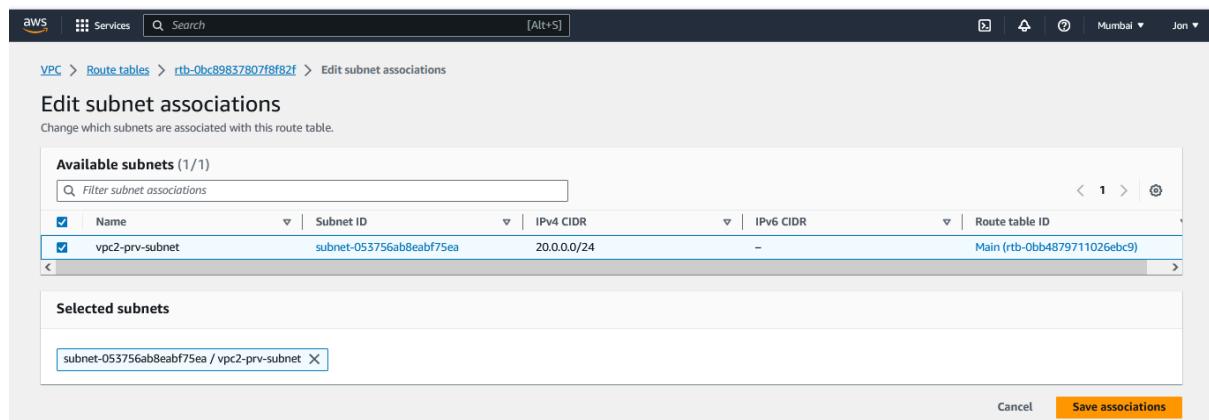
Cancel [Create route table](#)

Step 4:- Subnet Associations

Edit subnet associations



The screenshot shows the AWS VPC Route Table Details page. At the top, a green banner indicates that the route table was created successfully. Below the banner, a message says "You can now check network connectivity with Reachability Analyzer" with a "Run Reachability Analyzer" button. The main area shows the "Details" tab for a route table with the ID "rtb-0bc89837807f8f82f". The "Main" status is set to "No". The "Owner ID" is listed as "vpc-08cfb80918a46bbc6 | vpc2" and the "Last modified" date is "11185172188". The "Subnet associations" tab is selected, showing a table with columns: Name, Subnet ID, IPv4 CIDR, and IPv6 CIDR. The table is empty, displaying the message "No subnet associations" and "You do not have any subnet associations." Other tabs include "Routes", "Edge associations", "Route propagation", and "Tags".



The screenshot shows the "Edit subnet associations" dialog box. The title is "Edit subnet associations" and the sub-section is "Change which subnets are associated with this route table." The "Available subnets" section lists one subnet: "vpc2-prv-subnet" with Subnet ID "subnet-053756ab8eabf75ea", IPv4 CIDR "20.0.0.0/24", and Route table ID "Main (rtb-0bb4879711026ebc9)". The "Selected subnets" section shows the same subnet listed. At the bottom are "Cancel" and "Save associations" buttons.

Step 5:- Peering connection

Peering connection -> Create peering connection

Peering connections Info

Filter peering connections

Name	Peering connection ID	Status	Requester VPC	Acceptor VPC	Requester
No peering connection found					

Select a peering connection above

Requestor -> **vpc1**

Acceptor -> **vpc2**

Peering connection settings

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

vpc1tovpc2-peering

Select a local VPC to peer with

VPC ID (Requester)

vpc-00a60c42f9d857c70 (vpc1)

VPC CIDRs for vpc-00a60c42f9d857c70 (vpc1)

CIDR	Status	Status reason
10.0.0.0/16	Associated	-

Select another VPC to peer with

Account

My account

Another account

Region

This Region (ap-south-1)

Another Region

VPC ID (Acceptor)

vpc-08cfb80918a46bbc6 (vpc2)

Region

This Region (ap-south-1)

Another Region

VPC ID (Acceptor)

vpc-08cfb80918a46bbc6 (vpc2)

VPC CIDRs for vpc-08cfb80918a46bbc6 (vpc2)

CIDR	Status	Status reason
20.0.0.0/16	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	vpc1tovpc2-peering

Add new tag

You can add 49 more tags.

Cancel **Create peering connection**

Accept

Peering connections (1/1) **Info**

Name	Peering connection ID	Status	Requester VPC
vpc1tovpc2-pe...	px-0ead6739c1de2b159	Pending acceptance	vpc-00a60c42f9d857c70 / vpc1

Actions

Accept request

Reject request

Edit DNS settings

Manage tags

Delete peering connection

Accept VPC peering connection request **Info**

Are you sure you want to accept this VPC peering connection request? (px-0ead6739c1de2b159 / vpc1tovpc2-peering)

Requester VPC	Acceptor VPC	Requester CIDRs
vpc-00a60c42f9d857c70 / vpc1	vpc-08cfb80918a46bbc6 / vpc2	10.0.0.0/16
Acceptor CIDRs	Requester Region	Acceptor Region
-	Mumbai (ap-south-1)	Mumbai (ap-south-1)
Requester owner ID	Acceptor owner ID	
115185172188 (This account)	115185172188 (This account)	

Cancel **Accept request**

Step 5:- Route table association

Edit routes

The screenshot shows the AWS VPC Route Tables page. The left sidebar includes options for VPC dashboard, EC2 Global View, Filter by VPC, Virtual private cloud, Route tables, Security, and Network ACLs. The main content area displays a table of route tables with columns for Name, Route table ID, Explicit subnet association, Edge associations, Main, and VPC. One route table, 'vpc2-prv-rtb', is selected. Below the table is a 'Routes (1)' section with a table showing a single route to '20.0.0.0/16' with a target of 'local' and an active status.

The screenshot shows the 'Edit routes' page for the selected route table. It lists two routes: one to '20.0.0.0/16' with a target of 'local' and another to '0.0.0.0/0' with a target of 'pcx-0ead6739c1de2b159'. The 'Add route' button is visible at the bottom left, and 'Save changes' is at the bottom right.

Step 6:- Create nacl

The screenshot shows the 'Create network ACL' page. It includes sections for 'Network ACL settings' (Name: 'vpc2-prv-nacl', VPC: 'vpc-08cfb80918a46bbc6 (vpc2)'), 'Tags' (Key: 'Name', Value: 'vpc2-prv-nacl'), and a summary section indicating 49 more tags can be added. The 'Create network ACL' button is at the bottom right.

Edit nacl inbound rules

aws Services Search [Alt+S] Mumbai Jon

VPC dashboard EC2 Global View New

Filter by VPC: Select a VPC

Virtual private cloud Your VPCs New

Subnets

Route tables

Internet gateways

Egress-only internet gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peerings

Network ACLs (1/11) Info

Find resources by attribute or tag

Name Network ACL ID Associated with Default VPC ID

vpc1-pub-nacl acl-05b2630709206b9ec - No vpc-00a60c42f9d857c70 / vpc1

vpc2-prv-nacl acl-06d7096be30a64beb - No vpc-08cfb80918a46bbc6 / vpc2

acl-06d7096be30a64beb / vpc2-prv-nacl

Details Inbound rules Outbound rules Subnet associations Tags

You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer

Inbound rules (1)

Filter inbound rules

Rule number Type Protocol Port range Source Allow/Deny

100 SSH (22) TCP (6) 22 10.0.0.1/16 Allow

* All traffic All All 0.0.0.0/0 Deny

Edit inbound rules

Add new rule Sort by rule number

Cancel Preview changes Save changes

aws Services Search [Alt+S] Mumbai Jon

VPC Network ACLs acl-06d7096be30a64beb / vpc2-prv-nacl Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the VPC.

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	SSH (22)	TCP (6)	22	10.0.0.1/16	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

Add new rule Sort by rule number

Cancel Preview changes Save changes

Step 7:- Create a security group

aws Services Search [Alt+S] Mumbai Jon

Your VPCs New

Subnets

Route tables

Internet gateways

Egress-only internet gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Security Groups (10) Info

Export security groups to CSV Create security group

Filter security groups

Name	Security group ID	Security group name	VPC ID	Description	Owner
-	sg-0bcd6e36268d6f37	default	vpc-0c0927f14731ea0c9	default VPC security gr...	115185172188
-	sg-0361906cd830bd323	vpc1-pub-sg	vpc-0f80da2d5913c923	vpc1-pub-sg	115185172188
-	sg-08706a52516a87af9	default	vpc-00a60c42f9d857c70	default VPC security gr...	115185172188
-	sg-01b2048f727f69a7	default	vpc-00784dd42f224120d	default VPC security gr...	115185172188
-	sg-07ab4d2df7a001b55	launch-wizard-1	vpc-0f80da2d5913c923	launch-wizard-1 create...	115185172188
adhi-vpc-sg-pv	sg-048e8038ade81144e	adhi-vpc-sg-pv	vpc-00784dd42f224120d	adhi-vpc-sg-pv	115185172188

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info
 Name cannot be edited after creation.

Description Info

VPC Info
 X

Inbound rules Info

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
SSH	TCP	22	Custom	<input type="text" value="10.0.0.0/16"/> <small>X</small>

[Add rule](#)

Outbound rules Info

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Destination <small>Info</small>	Description - optional <small>Info</small>
All traffic	All	All	Custom	<input type="text" value="0.0.0.0"/> <small>X</small>

[Add rule](#)

Tags - optional
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"/> <small>X</small>	<input type="text" value="vpc2-prv-sg"/> <small>X</small>

[Add new tag](#) You can add up to 49 more tags

[Cancel](#) [Create security group](#)

Creating instances

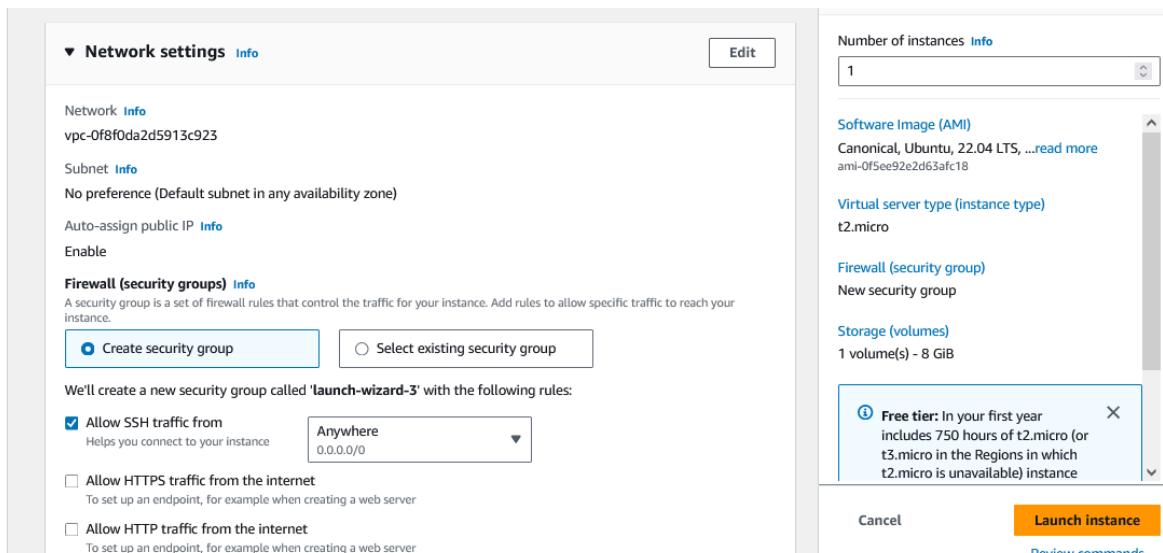
Public Instance

Launch instance

Step 1:-Now giving name as :- vpc1-public-instance

Step to Launch instance are same , only the difference is in Network settings

Ec2-> Launch instance -> Network settings ->Edit

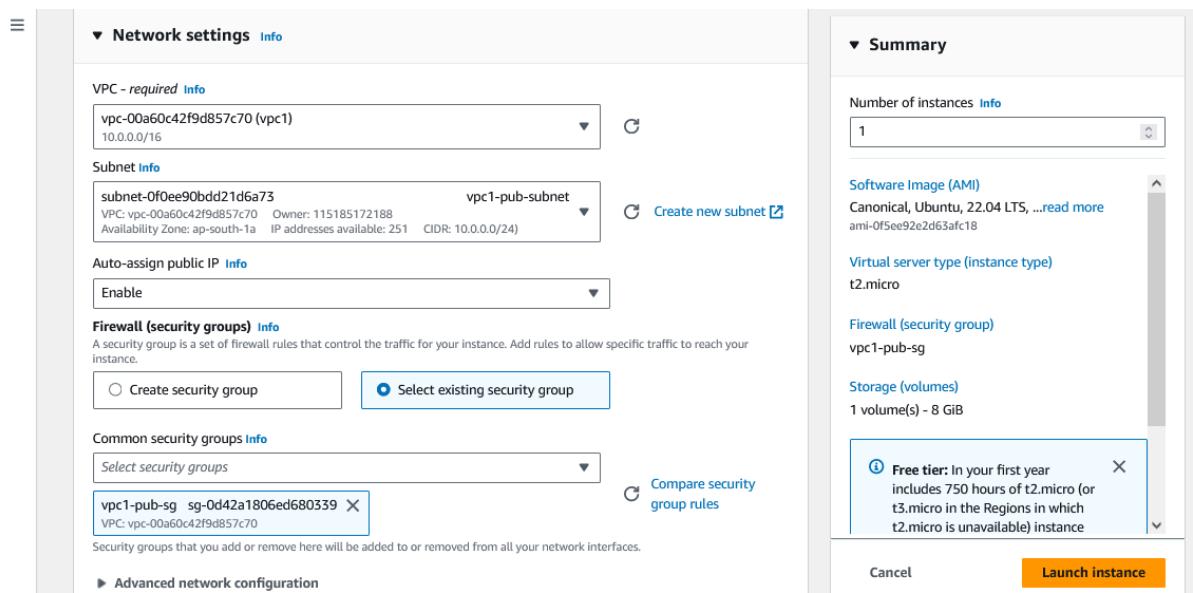


VPC -> select vpc1

Subnet -> select subnet1

Auto assign public IP -> Enable

Firewall:- select existing sg -> vpc 1



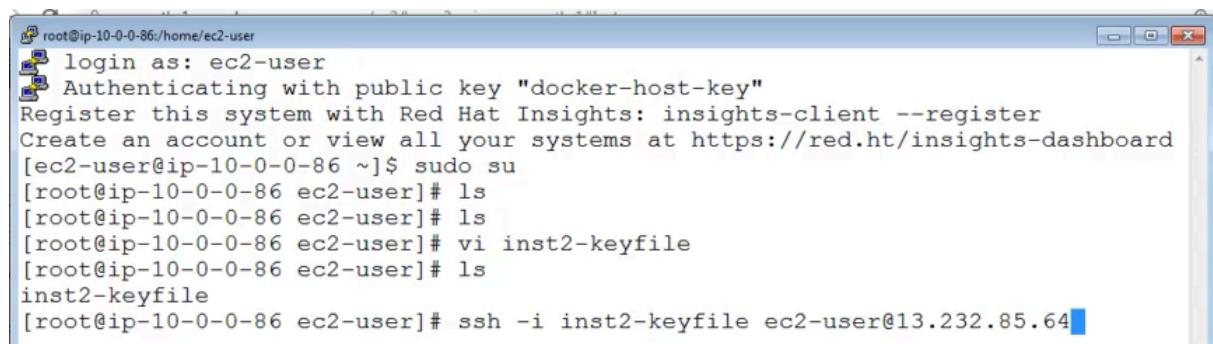
Follow the same process and create private instance (Select vpc2)

Step 2:-Now giving name as:- vpc2-private-instance

Follow the same process and create private instance (Select vpc2)

Step 3:- Do ssh means putty related configuration of vpc1-pub-instance

Now you have to connect from instance 1 to instance 2 .i.e. from vpc1 instance we want to connect to vpc2 instance



root@ip-10-0-0-86:/home/ec2-user
login as: ec2-user
Authenticating with public key "docker-host-key"
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
[ec2-user@ip-10-0-0-86 ~]\$ sudo su
[root@ip-10-0-0-86 ec2-user]# ls
[root@ip-10-0-0-86 ec2-user]# ls
[root@ip-10-0-0-86 ec2-user]# vi inst2-keyfile
[root@ip-10-0-0-86 ec2-user]# ls
inst2-keyfile
[root@ip-10-0-0-86 ec2-user]# ssh -i inst2-keyfile ec2-user@13.232.85.64

(Public ip of vpc2)

Run the above commands on vpc1-public-terminal