

Create 2 VPC's (VPC1 should be public and VPC 2 should be private)
communication between 2 VPC's ping should be worked, AWS peering.

Task:

Create a 2 VPC's (VPC1 should be public and VPC 2 should be private) communication between 2 VPC's ping should be worked, AWS peering:

- VPC (Virtual Private Cloud):

VPC allows you to create a private network within the AWS cloud. You can define your own network topology, set up subnets, and configure network security and routing.

- Route Tables:

Route tables are an essential component of the Virtual Private Cloud (VPC) networking infrastructure. They are used to control the traffic flow within a VPC, determine how traffic is routed between subnets, and define how traffic is directed to the internet or other AWS resources.

- Internet Gateway:

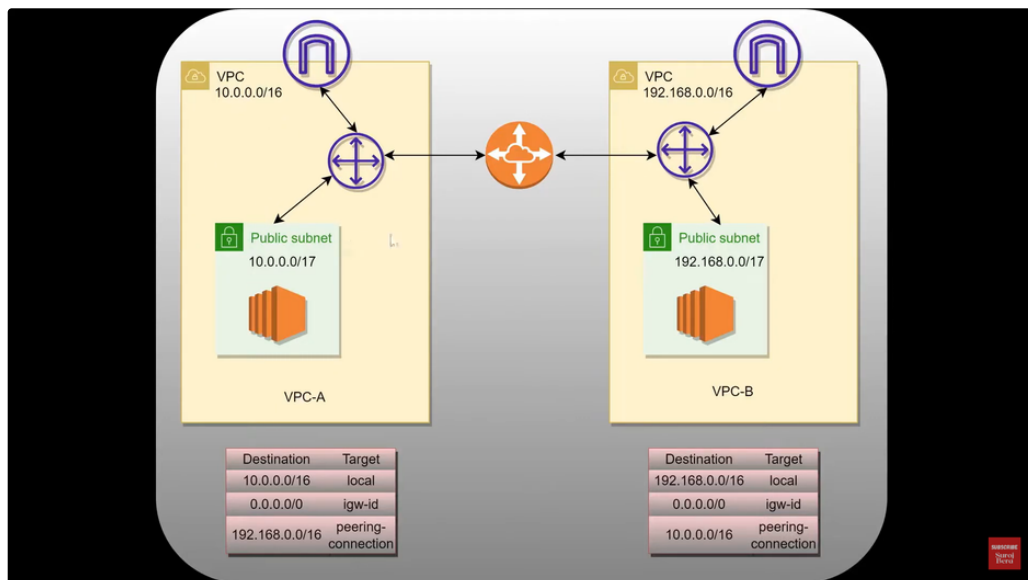
Internet Gateway is a crucial component for enabling communication between resources within your Virtual Private Cloud (VPC) and the public internet.

- VPC Peering Connection:

VPC Peering Connection is a network connection between two Virtual Private Clouds (VPCs) that allows them to communicate with each other as if they were on the same network. VPC Peering is a valuable feature for connecting VPCs within the same AWS account or across different accounts, enabling secure and private communication between resources in separate VPCs.

- Security Groups:

Security Group is a fundamental component for controlling inbound and outbound network traffic to and from Amazon Elastic Compute Cloud (EC2) instances and other AWS resources. Security Groups act as virtual firewalls for your resources, allowing you to define rules that specify which traffic is allowed or denied.



- Create 2 VPC's named as vpcA and vpcB, With IPV4 CIDR Manual Input. IPV4 for vpcA will be 10.0.0.0/16, IPV4 for vpcB will be 192.168.0.0/16.

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.

vpca

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

- Now create two internet gateways with names of igwA and igwB. Attach internet gateways with the VPC's created.

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.

igwA

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	
Q Name	Q igwA	Remove

[Add new tag](#)

You can add 49 more tags.

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

[VPC](#) > [Internet gateways](#) > Attach to VPC (igw-09ecec6275a7f73e7)

Attach to VPC (igw-09ecec6275a7f73e7) [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.

► AWS Command Line Interface command

Cancel

Attach internet gateway

- Now create two subnets in two different VPC's, those will be public subnets. VPC ID should be given individually to both as vpcA and vpcB.

Create subnet [Info](#)

VPC

VPC ID

Create subnets in this VPC.

Associated VPC CIDRs

IPv4 CIDRs

10.0.0.0/16

- Subnet names of vpcA is public-subnet-vpcA, name of B will be public-subnet-vpcB. Select the availability zone, Mention IPv4 subnet CIDR block as 10.0.0.0/17 and 192.168.0.0/17 and create subnets.

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 IPv4 VPC CIDR block to create a subnet in.

IPv4 subnet CIDR block

 32,768 IPs

< > ^ v

▼ Tags - optional

Key	Value - optional	
<input type="text" value="Name"/>	<input type="text" value="public-subnet-vpcA"/>	Remove

- After creating the subnets, select the subnet and edit subnet settings, Enable the auto-assign IP settings to automatically request a public IPv4 or IPv6 address for a new network interface in this subnet.

VPC > Subnets > subnet-06ac629d9dfea362f > Edit subnet settings

Edit subnet settings [Info](#)

Subnet

Subnet ID	Name
subnet-06ac629d9dfea362f	public-subnet-vpcA

Auto-assign IP settings [Info](#)

Enable the auto-assign IP settings to automatically request a public IPv4 or IPv6 address for a new network interface in this subnet.

☒ Enable auto-assign public IPv4 address [Info](#)

☐ Enable auto-assign customer-owned IPv4 address [Info](#)
Option disabled because no customer owned pools found.

- Now go to route table, here we can find the route tables of vpcA, vpcB. Select the route table and go to subnet associations and navigate to edit subnet associations.

Route tables (1/5) [Info](#)

Find resources by attribute or tag

	Name	Route table ID	Explicit subnet associati...	Edge associations	M
<input type="checkbox"/>	-	rtb-07106eac811bf2506	-	-	Ye
<input type="checkbox"/>	Default-vpc	rtb-0ba29d51e77a55793	-	-	Ye
<input type="checkbox"/>	-	rtb-026380ed2e52a26bf	subnet-0eb4b4ee79320c...	-	Ye
<input checked="" type="checkbox"/>	-	rtb-084600be2665a7d7d	-	-	Ye

rtb-084600be2665a7d7d

Details Routes **Subnet associations** Edge associations Route propagation Tags

Explicit subnet associations (0) [Edit subnet associations](#)

Find subnet association

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
------	-----------	-----------	-----------

- And attach the subnet with the route table by selecting it and save the associations.

VPC > Route tables > rtb-084600be2665a7d7d > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/1)

Filter subnet associations

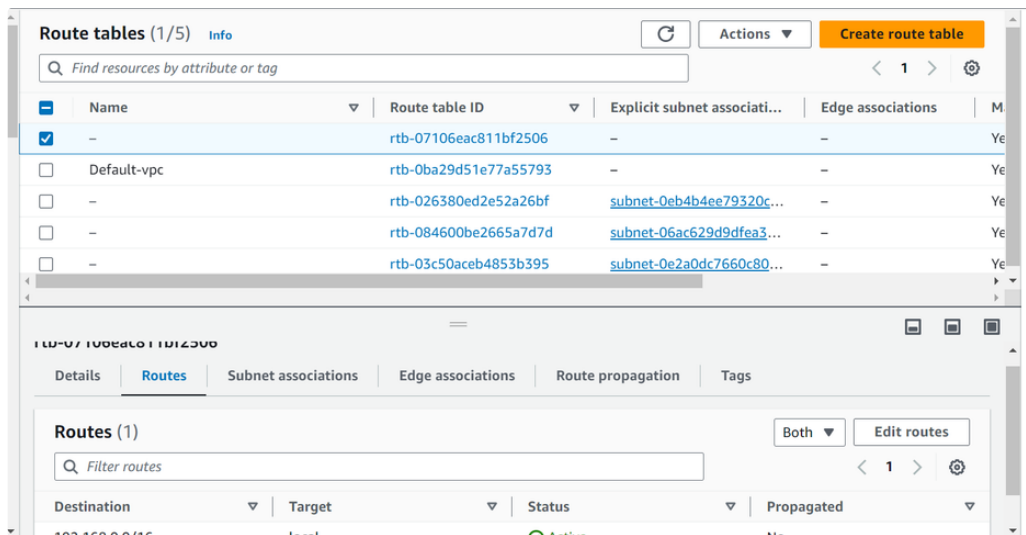
	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/>	public-subnet-vpcA	subnet-06ac629d9dfea362f	10.0.0.0/17	-	Main (rtb-084600be2665a7d7d)

Selected subnets

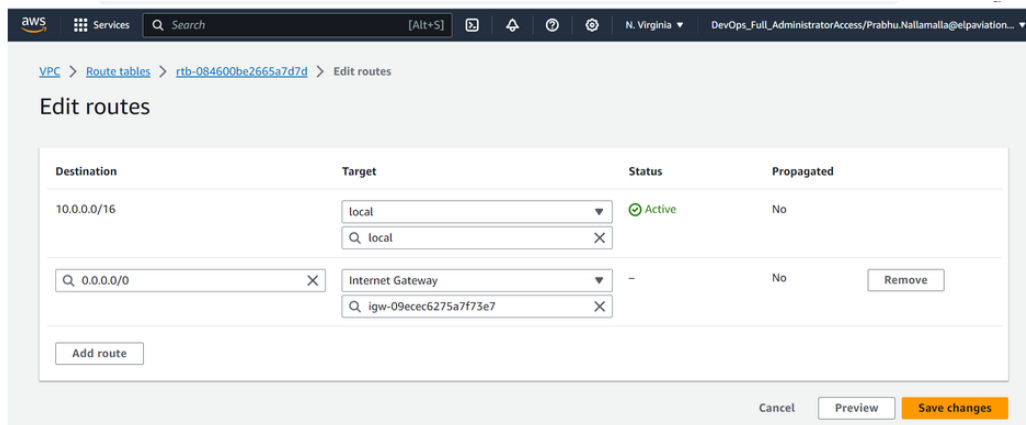
subnet-06ac629d9dfea362f / public-subnet-vpcA X

Cancel [Save associations](#)

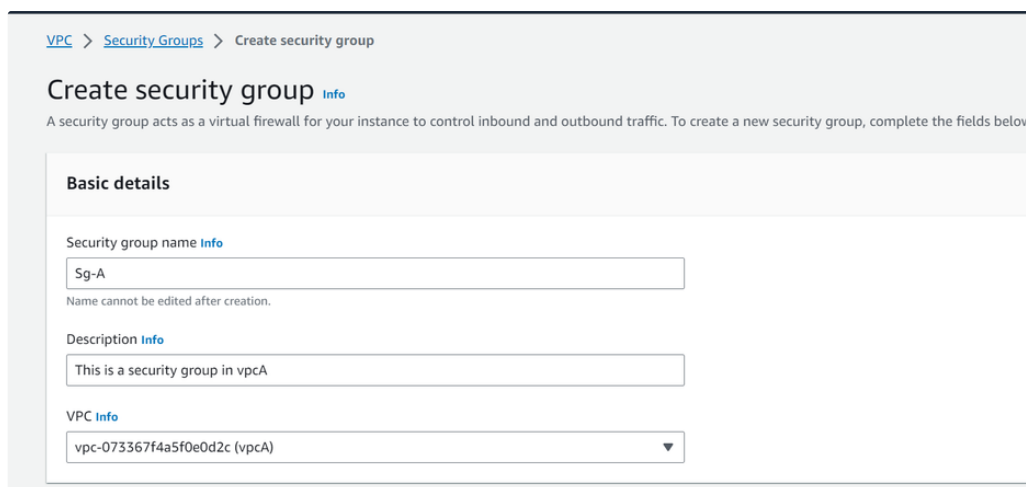
- Now go to the routes, and edit the routes. Defaultly the destination is 10.0.0.0/16 and target is local, which means it will works for local network.



- Now we need to add route, by giving destination as 0.0.0.0/0 and target will be internet gateway for igwA for vpcA and igwB for vpcB



- Create two security groups with names Sg-A, description will be This is a security group in vpcA. And using vpcA.
- Create two security groups with names Sg-B, description will be This is a security group in vpcB. And using vpcB.



- Now create the inbound rules of SSH, All ICMP-IPv4, using 10.0.0.0/16 and 192.168.0.0/16 in both vpcA and vpcB. Outbound rules will be default.

Inbound rules [info](#)

Type info	Protocol info	Port range info	Source info	Description - optional info	
SSH	TCP	22	An... 0.0.0.0/0		Delete
All ICMP - IPv4	ICMP	All	Cu... 10.0.0.0/16		Delete
All ICMP - IPv4	ICMP	All	Cu... 192.168.0.0/16		Delete

[Add rule](#)

- Now create two EC2 instances, proceed without a key pair. And edit network settings, select the respective VPC, we can find the respected subnet also.
- Auto assign public IP is enabled. Select existing security group, and create EC2 instances and launch the instances.
- Check the internet connection by using ping google.com.
- Now try to ping with private IP of same instance. And it will work.
- Now try to ping with opposite instance private IP's, here it won't work as expected.
- Create a peering connection, give a name as vpcAtovpcB, and select vpcA for requester and vpcB as acceptor.

Peering connection settings

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

vpcAtovpcB

Select a local VPC to peer with

VPC ID (Requester)

vpc-00be8ff64514da14d (vpc-A)

VPC CIDRs for vpc-00be8ff64514da14d (vpc-A)

CIDR	Status	Status reason
10.0.0.0/16	✔ Associated	-

Select another VPC to peer with

Account

☒ My account

☐ Another account

Region

☒ This Region (us-east-1)

☐ Another Region

☒ This Region (us-east-1)
☐ Another Region

VPC ID (Acceptor)

vpc-0af0b351c4e3395cf (vpcB) ▼

VPC CIDRs for vpc-0af0b351c4e3395cf (vpcB)

CIDR	Status	Status reason
192.168.0.0/16	✔ 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

Q Name X Q vpcAtovpcB X Remove

Add new tag

You can add 49 more tags.

Cancel Create peering connection

- After creating peering connection, select the connection and accept it. Now click on edit the route tables go to route tables, select and edit the route add 192.168.0.0/16 with target of peering connection of vpcAtovpcB. repeat the same step with 10.0.0.0/16 in route table B.

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local Q local X	✔ Active	No
Q 0.0.0.0/0 X	Internet Gateway Q igw-09ecec6275a7f73e7 X	✔ Active	No Remove
Q 192.168.0.0/16 X	Peering Connection Q pcx-0847e915361676dab X	-	No Remove

Add route

Cancel Preview Save changes

- Go to ec2 instances and try to ping the IP address of different networks, now it will work.

```

ubuntu@ip-192-168-101-89:~$ ping 10.0.2.234
PING 10.0.2.234 (10.0.2.234) 56(84) bytes of data.
64 bytes from 10.0.2.234: icmp_seq=1 ttl=64 time=0.359 ms
64 bytes from 10.0.2.234: icmp_seq=2 ttl=64 time=0.486 ms
64 bytes from 10.0.2.234: icmp_seq=3 ttl=64 time=0.441 ms
^C
--- 10.0.2.234 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2028ms
rtt min/avg/max/mdev = 0.359/0.428/0.486/0.052 ms
ubuntu@ip-192-168-101-89:~$
  
```

i-07fbb1bb718cccb35 (ec2B)

PublicIPs: 54.235.45.93 PrivateIPs: 192.168.101.89

- Now try to ping using the public IP's, it will not work. Go to security group and edit the inbound rule. By ICMP, give target as public IP of ec2-A and ec2-B.
- Go to security group and edit the inbound rule. By ICMP, give target as public IP of ec2-B and ec2-A.

0.0.0.0/0 X

All ICMP - IPv4 ICMP All Cu... Q 54.196.236.236 X

54.196.236.236/32 X

All ICMP - IPv4 ICMP All Cu... Q 54.235.45.93/32 X

54.235.45.93/32 X

Add rule Delete Delete

- Now check the ping connection by using the both public IP's on both the instances it will work.

```
ubuntu@ip-192-168-101-89:~$ ping 54.196.236.236
PING 54.196.236.236 (54.196.236.236) 56(84) bytes of data.
64 bytes from 54.196.236.236: icmp_seq=1 ttl=63 time=0.573 ms
64 bytes from 54.196.236.236: icmp_seq=2 ttl=63 time=0.604 ms
^C
--- 54.196.236.236 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1024ms
rtt min/avg/max/mdev = 0.573/0.588/0.604/0.015 ms
ubuntu@ip-192-168-101-89:~$
```

i-07fbb1bb718cccb35 (ec2B)

PublicIPs: 54.235.45.93 PrivateIPs: 192.168.101.89