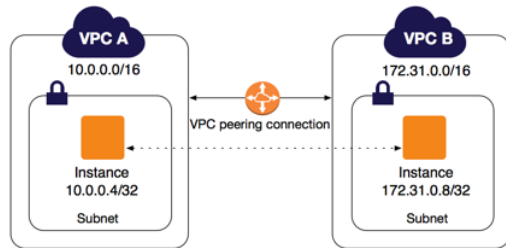


Amazon VPC Peering

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses. Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account. The VPCs can be in different regions (also known as an inter-region VPC peering connection).

<https://docs.aws.amazon.com/vpc/latest/peering/working-with-vpc-peering.html>



Step 1:- Create two VPC with different range of ip address.

Name:- Robo-VPC A (10.5.0.0/16) and Robo-VPC B (192.168.0.0/16)

Create VPC

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances.

Name tag:

IPv4 CIDR block*:

IPv6 CIDR block: ☐ No IPv6 CIDR Block ☐ Amazon provided IPv6 CIDR block ☐ IPv6 CIDR owned by me

Tenancy:

Create VPC **Actions**

Create VPC

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances.

Name tag:

IPv4 CIDR block*:

IPv6 CIDR block: ☐ No IPv6 CIDR Block ☐ Amazon provided IPv6 CIDR block ☐ IPv6 CIDR owned by me

Tenancy:

VPC Dashboard

Filter by VPC:

Virtual Private Cloud

Your VPCs

Name	VPC ID	State	IPv4 CIDR
Robo-VPC_B	vpc-0270047c44bf6a9c0	available	192.168.0.0/16
Robo-VPC_A	vpc-0ed9cdbb73b3a703f	available	10.5.0.0/16

Step2:-Create three subnets in that for Robo-VPC A assign two subnets and for Robo-VPC B assign one for Robo-VPC B and make sure to select different Availability Zone for private and public.

Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between /16 and /30.

Name tag:

VPC*:

Availability Zone:

VPC CIDRs:

CIDR	Status
10.5.0.0/16	associated

IPv4 CIDR block*:

Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between /16 and /30.

Name tag:

VPC*:

Availability Zone:

VPC CIDRs:

CIDR	Status
10.5.0.0/16	associated

IPv4 CIDR block*:

Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between /16 and /28. IPv6 block sizes must be a /64 CIDR block.

Name tag

VPC*

Availability Zone

VPC CIDRs

CIDR	Status
192.168.0.0/16	associated

IPv4 CIDR block*

<input type="checkbox"/>	Robo-VPC_B_SUB_private	subnet-0383e13f3b598c6d8	available	vpc-0270047c44bf6a9c0 Robo-VPC_B	192.168.4.0/24
<input type="checkbox"/>	Robo-VPC_A_SUB_public	subnet-0d74ba8f9af394f63	available	vpc-0ed9cddb73b3a703f Robo-VPC_A	10.5.0.0/24
<input type="checkbox"/>	Robo-VPC_A_SUB_private	subnet-0eb1ed7925be13a28	available	vpc-0ed9cddb73b3a703f Robo-VPC_A	10.5.1.0/24

Note:- Here for testing purpose I have selected Availability Zone:-<US-east-2b> for private subnets on both VPC and Availability Zone:-<US-east-2a> for public subnet in Robo-VPC A

Step3:-Create a route tables for public and private subnets.

Name tag Name tag Name tag

VPC* VPC* VPC*

<input checked="" type="checkbox"/>	Robo-VPC_RT_pub_sub	rtb-03f8e03b2a75a7388	vpc-0ed9cddb73b3a703f Robo-VPC_A	898051851723
<input checked="" type="checkbox"/>	Robo-VPC_RT_private_sub	rtb-0f513e0741951ea06	vpc-0ed9cddb73b3a703f Robo-VPC_A	898051851723
<input checked="" type="checkbox"/>	Robo-VPC-B_RT_private_sub	rtb-00942d294c1fdbf53	vpc-0270047c44bf6a9c0 Robo-VPC_B	898051851723

Note:- Here two route table(private&public) has been created Robo-VPC A and one route table (private) has been created for Robo-VPC B

Step4:-Create peering connection between Robo-VPC A and Robo-VPC B

@Click on Peering Connections → Create Peering Connection → enter the Name tag → select VPC (Requester)* < Robo-VPC A > →

aws Services Resource Groups

NAI Gateways

Peering Connections

Create Peering Connection

Actions

Create Peering Connection

Peering connection name tag

Select a local VPC to peer with

VPC (Requester)*

CIDRs

CIDR	Status	Status Reason
10.5.0.0/16	associated	

Select another VPC to peer with

Account ☒ My account
☐ Another account

Region ☒ This region (us-east-2)
☐ Another Region

VPC (Acceptor)* vpc-0270047c44bf6a9c0

CIDRs	CIDR	Status	Status Reason
	192.168.0.0/16	● associated	

✓ Success

A VPC peering connection (pcx-045cdb74631b976ed) has been requested.

Requester VPC owner	898051851723 (This account)	Acceptor VPC owner	898051851723 (This account)
Requester VPC ID	vpc-0ed9cdbb73b3a703f	Acceptor VPC ID	vpc-0270047c44bf6a9c0
Requester VPC Region	us-east-2	Acceptor VPC Region	us-east-2
Requester VPC CIDRs	10.5.0.0/16	Acceptor VPC CIDRs	-

@@Successfully created and its pending for acceptance.

Name	Peering Connection	Status	Requester VPC	Acceptor VPC	Requester CIDRs	Acceptor CIDRs
Chitti_robo2.0	pcx-045cdb74631...	● Pending Acceptance	vpc-0ed9cdbb73b...	vpc-0270047c44bf...	10.5.0.0/16	-

@Click on Actions → then select Accept Request

Create Peering Connection Actions

Filter by tags and attributes or

Chitti_robo2.0 pcx-045c...

- Accept Request
- Reject Request
- Delete VPC Peering Connection
- Edit DNS Settings
- Add/Edit Tags

Name	Peering Connection	Status	Requester VPC	Acceptor VPC	Requester CIDRs	Acceptor CIDRs
Chitti_robo2.0	pcx-045cdb74631...	● Active	vpc-0ed9cdbb73b...	vpc-0270047c44bf...	10.5.0.0/16	192.168.0.0/16

Step5:-Create a internet gateway and attach the same VPC

@Click on Internet Gateways → Create internet gateway → enter the name tag → after creation then attach the same to Robo-VPC A

Robo-VPC_A_public_sub-IGW	igw-00baed9ac43...	detached	Robo-VPC_A_public_sub-IGW	igw-00baed9ac43...	attached
---------------------------	--------------------	----------	---------------------------	--------------------	----------

Step5.1 click on route tables and add rule for target IGW. Routes → Edit routes → Add route

@Select public routes Robo-VPC RT_pub_sub and edit route.
Edit routes

Destination	Target	Status
10.5.0.0/16	local	active
0.0.0.0/0	igw-	
Add route		
igw-00baed9ac436f65ed Robo-VPC_A_public_sub-IGW		

@Select private route Robo-VPC_RT_private_sub and add target as peering connection.
Edit routes

Destination	Target	Status	Propagated
10.5.0.0/16	local	active	No
192.168.4.0/24	pcx-		No

Add route

pcx-045cdb74631b976ed Chitti_robo2.0

@Select private route Robo-VPC-B_RT_private_sub add
Edit routes

Destination	Target	Status	Propagated
192.168.0.0/16	local	active	No
10.5.1.0/24	pcx-		No

Add route

pcx-045cdb74631b976ed Chitti_robo2.0

Step5.1 Associate respective subnet.

@Select Robo-VPC_RT_pub_sub → edit subnet associations

Name	Route Table ID	VPC ID	Owner
Robo-VPC_RT_pub_sub	rtb-03f8e03b2a75a7388	vpc-0ed9cddb73b3a703f Robo-VPC_A	898051851723
Robo-VPC_RT_private_sub	rtb-0f513e0741951ea06	vpc-0ed9cddb73b3a703f Robo-VPC_A	898051851723
Robo-VPC-B_RT_private_sub	rtb-00942d294c1fdbf53	vpc-0270047c44dbf6a9c0 Robo-VPC_B	898051851723

Route Table: rtb-03f8e03b2a75a7388

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Edit subnet associations

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

You do not have any subnet associations.

Edit subnet associations

Route table rtb-03f8e03b2a75a7388 (Robo-VPC_RT_pub_sub)

Associated subnets subnet-0d74ba8f9af394f63

Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table
subnet-0eb1ed7925be13a28 Robo-VPC_A_SUB_private	10.5.1.0/24	-	rtb-0f513e0741951ea06
subnet-0d74ba8f9af394f63 Robo-VPC_A_SUB_public	10.5.0.0/24	-	Main

@Select Robo-VPC_RT_private_sub → edit subnet associations
Edit subnet associations

Route table rtb-0f513e0741951ea06 (Robo-VPC_RT_private_sub)

Associated subnets subnet-0eb1ed7925be13a28

Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table
subnet-0eb1ed7925be13a28 Robo-VPC_A_SUB_private	10.5.1.0/24	-	rtb-03f8e03b2a75a7388
subnet-0d74ba8f9af394f63 Robo-VPC_A_SUB_public	10.5.0.0/24	-	Main

@Select Robo-VPC-B_RT_private_sub → edit subnet associations
Edit subnet associations

Route table rtb-00942d294c1fdbf53 (Robo-VPC-B_RT_private_sub)

Associated subnets subnet-0383e13f3b598c6d8

Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table
subnet-0383e13f3b598c6d8 Robo-VPC_B_SUB_private	192.168.4.0/24	-	Main

Step6:-Launch instance

@Create instance with public subnet.

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances [Launch into Auto Scaling Group](#)

Purchasing option ☐ Request Spot instances

Network [Create new VPC](#)

Subnet [Create new subnet](#)
251 IP Addresses available

Auto-assign Public IP

Placement group ☐ Add instance to placement group

Capacity Reservation [Create new Capacity Reservation](#)

IAM role [Create new IAM role](#)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

@Create instance with private subnet.

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances [Launch into Auto Scaling Group](#)

Purchasing option ☐ Request Spot instances

Network [Create new VPC](#)

Subnet [Create new subnet](#)
251 IP Addresses available

Auto-assign Public IP

Placement group ☐ Add instance to placement group

Capacity Reservation [Create new Capacity Reservation](#)

IAM role [Create new IAM role](#)

@Create instance with private subnet of VPC B

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances [Launch into Auto Scaling Group](#)

Purchasing option ☐ Request Spot instances

Network [Create new VPC](#)

Subnet [Create new subnet](#)
251 IP Addresses available

Auto-assign Public IP

Placement group ☐ Add instance to placement group

Capacity Reservation [Create new Capacity Reservation](#)

IAM role [Create new IAM role](#)

Step 7:-Assign EIP to public instance subnet.

@Click on Elastic IPs → click on Allocate Elastic IP address → click allocate.

Elastic IP addresses (1/1) [Refresh](#) [Actions](#) [Allocate Elastic IP address](#)

[<](#) [1](#) [>](#) [Settings](#)

<input checked="" type="checkbox"/>	Name	Public IPv4 address	Allocation ID	Associated instance ID
<input checked="" type="checkbox"/>		18.189.211.232	eipalloc-02127d5c8cc6c8add	-

@Click on Actions → Associate Elastic IP address

Elastic IP addresses (1/1)

Filter Elastic IP addresses

Name	Public IPv4 address	Allocation ID
	18.189.211.232	eipalloc-02127d5c8cc6c8add

Elastic IP address: 18.189.211.232

Resource type
Choose the type of resource with which to associate the Elastic IP address.

☒ Instance
☐ Network interface

⚠ If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account. [Learn more.](#)

Instance

Choose an instance

- i-0921fe1140205ff14 (VPC_B_private) - running
- i-05a0c6040156a82c4 (VPC_A_public) - running**
- i-0ec87625eb12b0538 (VPC_A_Private) - running

Reassociation
Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.

☐ Allow this Elastic IP address to be reassociated

Public IPv4 address: 18.189.211.232 X Clear filters

Name	Public IPv4 address	Allocation ID	Associated instance ID
	18.189.211.232	eipalloc-02127d5c8cc6c8add	i-05a0c6040156a82c4

Step 8:- Now try to access server via public ip.

```
[centos@ip-10-5-0-192 ~]$ uptime
03:50:07 up 1:03, 1 user, load average: 0.14, 0.08, 0.05
[centos@ip-10-5-0-192 ~]$ uname -a
Linux ip-10-5-0-192.us-east-2.compute.internal 3.10.0-957.1.3.el7.x86_64
x86_64 GNU/Linux
[centos@ip-10-5-0-192 ~]$ hostname
ip-10-5-0-192.us-east-2.compute.internal
[centos@ip-10-5-0-192 ~]$
```

@Copy the .pem key via winscp to public instance and then try to access private instance.

#chmod 600 AWS-SSHKEY.pem

#ssh -i AWS-SSHKEY.pem centos@10.5.1.67

```
[centos@ip-10-5-0-192 ~]$ ssh -i AWS-SSHKEY.pem centos@10.5.1.67
[centos@ip-10-5-1-67 ~]$ uname -a
Linux ip-10-5-1-67.us-east-2.compute.internal 3.10.0-957.1.3.el7.x86_64
x86_64 GNU/Linux
[centos@ip-10-5-1-67 ~]$ w
 04:02:45 up 1:12, 1 user, load average: 0.00, 0.01, 0.03
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
centos    pts/0    10.5.0.192    04:02    5.00s  0.01s  0.01s  w
[centos@ip-10-5-1-67 ~]$
```

@Now come out from VPC-A private instance and try to copy .pem key from public instance to private.

#scp -r -i AWS-SSHKEY.pem AWS-SSHKEY.pem centos@10.5.1.67:~

```
[centos@ip-10-5-0-192 ~]$ scp -r -i AWS-SSHKEY.pem AWS-SSHKEY.pem centos@10.5.1.67:~
AWS-SSHKEY.pem 100% 1692 869.9KB/s 00:00
[centos@ip-10-5-0-192 ~]$
```

@again ssh to VPC-A private instance and from there try to copy .pem key to VPC-B private and access via ssh.

#ssh -i AWS-SSHKEY.pem [centos@10.5.1.67](#)

```
[centos@ip-10-5-1-67 ~]$ w
 04:09:18 up  1:19,  1 user,  load average: 0.00, 0.01, 0.03
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
centos    pts/0    10.5.0.192    04:07    6.00s  0.02s  0.01s  w
[centos@ip-10-5-1-67 ~]$
```

#ping 192.168.4.104

```
[centos@ip-10-5-1-67 ~]$ ping 192.168.4.104
PING 192.168.4.104 (192.168.4.104) 56(84) bytes of data.
64 bytes from 192.168.4.104: icmp_seq=1 ttl=64 time=0.422 ms
64 bytes from 192.168.4.104: icmp_seq=2 ttl=64 time=0.545 ms
```

#ssh -i AWS-SSHKEY.pem [centos@192.168.4.104](#)

```
[centos@ip-10-5-1-67 ~]$ ssh -i AWS-SSHKEY.pem centos@192.168.4.104
The authenticity of host '192.168.4.104 (192.168.4.104)' can't be established.
ECDSA key fingerprint is SHA256:LTwp3RF8TKb/KA72d/eWVt3dSVsY+3l4aPfyY00OPD8.
ECDSA key fingerprint is MD5:8d:0f:8b:22:c5:88:8c:4b:e9:c2:c1:50:b2:0b:57.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.4.104' (ECDSA) to the list of known hosts.
```

#w

#hostname -i

```
[centos@ip-192-168-4-104 ~]$ w
 04:11:44 up  1:16,  1 user,  load average: 0.00, 0.01, 0.04
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
centos    pts/0    10.5.1.67     04:11    0.00s  0.02s  0.02s  w
[centos@ip-192-168-4-104 ~]$ hostname -i
fe80::4c8:d9ff:fe2d:de42%eth0 192.168.4.104
[centos@ip-192-168-4-104 ~]$
```

Note:- Now we have created VPC peering and You can establish peering relationships between VPCs across different AWS Regions (also called Inter-Region VPC Peering). This allows VPC resources including EC2 instances, Amazon RDS databases and Lambda functions that run in different AWS Regions to communicate with each other using private IP addresses, without requiring gateways, VPN connections, or separate network appliances. The traffic remains in the private IP space. All inter-region traffic is encrypted with no single point of failure, or bandwidth bottleneck. Traffic always stays on the global AWS backbone, and never traverses the public internet, which reduces threats, such as common exploits, and DDoS attacks. Inter-Region VPC Peering provides a simple and cost-effective way to share resources between regions or replicate data for geographic redundancy.