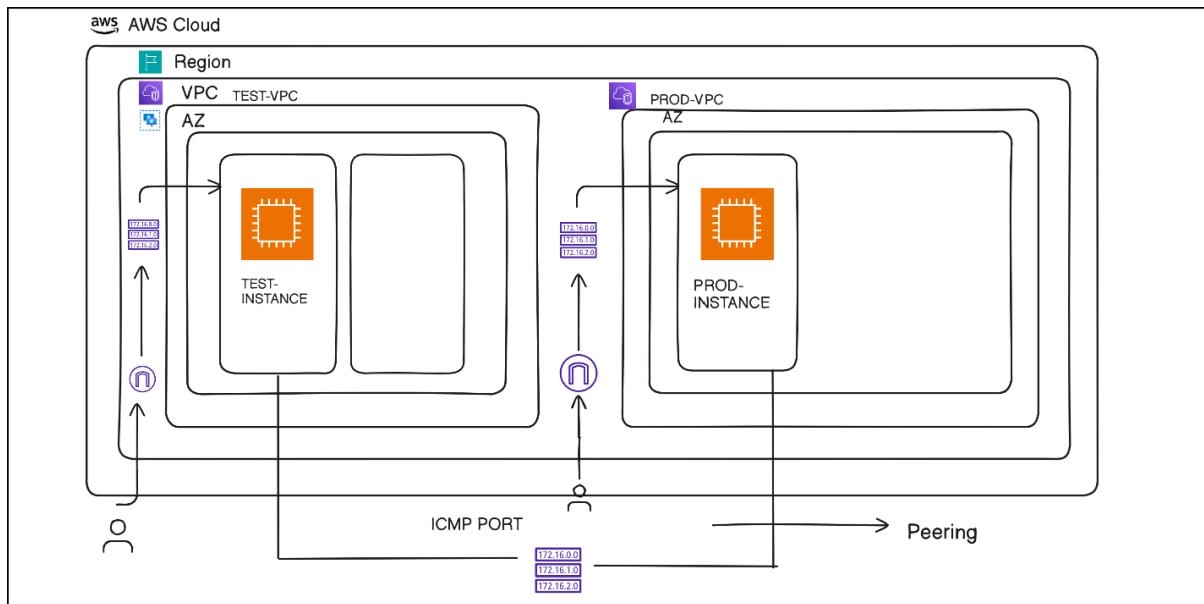


VPC Peering Project

Project Overflow:



Steps to do VPC Peering in AWS:

1. Go to the VPC page.
2. Click on "Create VPC" and provide the following information in the required fields.
 - a. VPC name - "test-vpc."
 - b. CIDR - "10.0.0.0/16"
 - c. Click on Create VPC.

VPCs VPC encryption controls - new					
Your VPCs (1/3) Info			Last updated 6 minutes ago	Actions	Create VPC
<input type="text"/> Find VPCs by attribute or tag					
Name	VPC ID	State	Encryption c...	Encryption control .	
<input type="checkbox"/> prod-vpc	vpc-0d6f9fa85e2177e5a	Available	-	-	
<input type="checkbox"/> -	vpc-085b239eccf83bc44	Available	-	-	
<input checked="" type="checkbox"/> test-vpc	vpc-05ad6920775719138	Available	-	-	

3. Go to the Subnets option.
4. Click on Create subnet and provide the following information in the required fields.
 - a. VPC ID - "test-vpc."
 - b. Name of the subnet - "public-test-subnet."
 - c. Availability zones - "us-east-2a"

- d. Divide the range of CIDR - “10.0.0.0/24”
- e. Click on “Create Subnet”.

Subnets (1/5) Info				
<input type="checkbox"/>	Name	Subnet ID	State	VPC
<input checked="" type="checkbox"/>	public-test-subnet	subnet-00e9041fae747c437	Available	vpc-05ad6920775719138 test...
<input type="checkbox"/>	-	subnet-0337b579399f0fa5e	Available	vpc-085b239eccf83bc44
<input type="checkbox"/>	-	subnet-0e767b9cab75ed1fe	Available	vpc-085b239eccf83bc44
<input type="checkbox"/>	-	subnet-070649e85e3346b83	Available	vpc-085b239eccf83bc44
<input type="checkbox"/>	prod-public-subnet	subnet-03d79089a01141115	Available	vpc-0d6f9fa85e2177e5a prod...

5. Go to the route table.
6. Click on Create route table and provide the following information in the required fields.
 - a. Name of table - “test-rtb.”
 - b. Select VPC - “test-vpc.”
 - c. Click on “Create route table.”

Route tables (1/5) Info					
<input type="checkbox"/>	Name	Route table ID	Explicit subnet associ...	Edge associations	Main
<input type="checkbox"/>	-	rtb-055d2adae34ecdc79	-	-	Yes
<input checked="" type="checkbox"/>	test-rtb	rtb-04dd5e7dc55379ee6	subnet-00e9041fae747c...	-	No
<input type="checkbox"/>	prod-rtb	rtb-09d5b55486141e6c8	subnet-03d79089a01141...	-	No
<input type="checkbox"/>	-	rtb-017f165c79351e665	-	-	Yes
<input type="checkbox"/>	-	rtb-0d524292b33ae5c9f	-	-	Yes

- To allow the traffic from the outside world to create an internet gateway.
7. Go to the internet gateway.
 8. Click on “create internet gateway”.
 9. Give the name of the gateway and create it.
 - a. Name - “test-igw”

Internet gateways (1/3) Info			
Actions Create internet gateway			
Name	Internet gateway ID	State	VPC ID
<input checked="" type="checkbox"/> test-igw	igw-01bf5422785226ccb	Attached	vpc-05ad6920775719138 test
<input type="checkbox"/> -	igw-069980e44fa2c829b	Attached	vpc-085b239eccf83bc44
<input type="checkbox"/> prod-igw	igw-08feb00d4979c269c	Attached	vpc-0d6f9fa85e2177e5a prod

10. Attach your interway gateway to the created VPC - “test-vpc”.

11. Go to the EC2.

12. Click on “Launch instance” and provide the following information in the required fields.

- a. Name of instance - “test-instance.”
- b. Choose the AMI - “ubuntu.”
- c. Choose the instance-type - “t2.micro”.
- d. Give the key pair created or create a new pair - “aws-key.”
- e. In the networking section, go to edit and edit the VPC details.
- f. Choose VPC -”test-vpc.”
- g. Choose a subnet.
- h. Create a new security group that allows SSH and HTTP.
- i. Launch the instance.

Instances (1/2) Info					
Connect Actions Launch instances					
Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input checked="" type="checkbox"/> test-instance	i-0d70a6eb5c7dcf105	Running	t2.micro	2/2 checks passed	View alarms
<input type="checkbox"/> prod-instance	i-0fb581fae68f99950	Running	t2.micro	2/2 checks passed	View alarms

Now do the routing through the route tables so the requests reach the test-instance server.

13. Go to the VPC and route table options.

14. Click on the “test-rtb” checkbox and below the Route option.

rtb-04dd5e7dc55379ee6 / test-rtb						
Details	Routes	Subnet associations	Edge associations	Route propagation	Tags	
Routes (3)						
Edit routes						
Destination	Target	Status	Propagated	Route Origin		

15. Edit routes and allow the routing of the internet gateway with this route table, and save the changes.

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	Internet Gateway	Active	No	CreateRoute
igw-01bf5422785226ccb				

Add route Cancel Preview Save changes

16. Go to the subnet associations and attach your “public-test-subnet” to the route table “test-rtb” by clicking on edit subnet associations.

Name	Route table ID	Explicit subnet associ...	Edge associations	Main
-	rtb-055d2adae34ecdc79	-	-	Yes
<input checked="" type="checkbox"/> test-rtb	rtb-04dd5e7dc55379ee6	subnet-00e9041fae747c...	-	No
<input type="checkbox"/> prod-rtb	rtb-09d5b55486141e6c8	subnet-03d79089a01141...	-	No
-	rtb-017f165c79351e665	-	-	Yes

rtb-04dd5e7dc55379ee6 / test-rtb

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

Explicit subnet associations (1)

Edit subnet associations

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/> public-test-subnet	subnet-00e9041fae747c437	10.0.0.0/24	-	rtb-04dd5e7dc55379ee6 / test-rtb

Selected subnets

subnet-00e9041fae747c437 / public-test-subnet X

Cancel Save associations

Now your “test-instance” is connected to the internet and ready to use.

17. Do the same process for creating “prod-vpc” as well.

Points to remember:

1. While creating “prod-instance” in the security group, give IP address ranges “198.168.0.0/16” in the HTTP request because we are creating prod-vpc, so it could be private, that’s why specific IP ranges.

EC2 > Security Groups > sg-00e0d29231d31dcc2 - launch-wizard-2 > Edit inbound rules

Edit inbound rules Info

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

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0627da7424fc283ac	SSH	TCP	22	Cus... ▾	0.0.0.0/0 X
sgr-08c8ba370c5beba0e	HTTP	TCP	80	Cus... ▾	192.168.0.0/16 X

Add rule

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

Buttons: Cancel, Preview changes, Save rules

VPC Peering Method:

1. Go to the VPC and click on the “peering connections” option.
2. Give the following information in the settings and create peering between the two servers.
3. From “test-vpc” to “prod-vpc”.

VPC > Peering connections > Create peering connection

Create peering connection

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them privately. Info

Peering connection settings

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

Select a local VPC to peer with

VPC ID (Requester)

VPC CIDRs for vpc-05ad6920775719138 (test-vpc)

CIDR	Status	Status reason
10.0.0.0/16	Associated	-

Buttons: Cancel, Preview, Create peering connection

Select another VPC to peer with

Account

 My account
 Another account

Region

 This Region (us-east-2)
 Another Region

VPC ID (Acceptor)

vpc-0d6f9fa85e2177e5a (prod-vpc)

VPC CIDRs for vpc-0d6f9fa85e2177e5a (prod-vpc)

CIDR	Status	Status reason
198.168.0.0/16	Associated	-

- To allow peering between the two VPCs, we need to accept the request just like a friend request on social media, and then only the two VPCs can communicate with each other.

Peering connections (1/1) [Info](#)

Find peering connections by attribute or tag

Name	Peering connection ID	Status
test-prod-peering	pcx-0d756e9858c08f720	Active

Actions ▾ [Create peering connection](#)

- [View details](#)
- Accept request
- [Reject request](#)
- [Edit DNS settings](#)
- [Manage tags](#)
- [Delete peering connection](#)

pcx-0d756e9858c08f720 / test-prod-peering

[Details](#) [DNS](#) [Route tables](#) [Tags](#)

Then, only the status will be active for peering.

- To allow communication, we need a route table between two VPCs so our message can go to the VPCs.
- Go to the VPC and route table option.
- Route the “test-rtb” with “prod-rtb” by adding their IP ranges.

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

Last updated 9 minutes ago [Actions](#) [Create route table](#)

Find route tables by attribute or tag

<input type="checkbox"/>	Name	Route table ID	Explicit subnet associations	Edge associations	Main
<input type="checkbox"/>	-	rtb-055d2adae34ecdc79	-	-	Yes
<input checked="" type="checkbox"/>	test-rtb	rtb-04dd5e7dc55379ee6	subnet-00e9041fae747c...	-	No
<input type="checkbox"/>	prod-rtb	rtb-09d5b55486141e6c8	subnet-03d79089a01141...	-	No
<input type="checkbox"/>	-	rtb-017f165c79351e665	-	-	Yes

rtb-04dd5e7dc55379ee6 / test-rtb

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

Routes (3)

Filter routes

Destination	Target	Status	Propagated	Route Origin
Q 198.168.0.0/16	local	<input checked="" type="radio"/> Active	No	CreateRoute
Q 0.0.0.0/0	Internet Gateway	<input checked="" type="radio"/> Active	No	CreateRoute

[Edit routes](#)

8. We are allowing routing from “test-instance” to “prod-instance” and vice versa in the form of IP addresses.

Prod IP Address

[Edit routes](#)

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	<input checked="" type="radio"/> Active	No	CreateRouteTable
Q 198.168.0.0/16	Peering Connection	<input checked="" type="radio"/> Active	No	CreateRoute
Q 0.0.0.0/0	Internet Gateway	<input checked="" type="radio"/> Active	No	CreateRoute

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

Test IP Address

[Edit routes](#)

Destination	Target	Status	Propagated	Route Origin
198.168.0.0/16	local	<input checked="" type="radio"/> Active	No	CreateRouteTable
Q 10.0.0.0/16	Peering Connection	<input checked="" type="radio"/> Active	No	CreateRoute
Q 0.0.0.0/0	Internet Gateway	<input checked="" type="radio"/> Active	No	CreateRoute

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

9. Open both instances - test and prod, and try to “PING” - to send the messages or packets to the other server to check whether the servers are getting the requests or not.

When you try to do a ping, it won't work because the "PING" command only works with the ICMP protocol.

10. Add an ICMP request to the security group.

In "test-instance", add the IP address of prod-instance, ie, 192.168.0.0/16.

Edit inbound rules Info

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

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-00738234d72e2b1b4	All ICMP - IPv4	ICMP	All	Cus... ▾	198.168.0.0/16 X
sgr-03032dd4cc39e4c8	HTTP	TCP	80	Cus... ▾	0.0.0.0/0 X
sgr-032658995fe718e25	SSH	TCP	22	Cus... ▾	0.0.0.0/0 X

Add rule

In "prod-instance", add the IP Address of test-instance, ie, 10.0.0.0/16.

Edit inbound rules Info

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

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0627da7424fc283ac	SSH	TCP	22	Cus... ▾	0.0.0.0/0 X
sgr-08c8ba370c5beba0e	HTTP	TCP	80	Cus... ▾	192.168.0.0/16 X
sgr-07dfc32cfb01c9a5f	All ICMP - IPv4	ICMP	All	Cus... ▾	10.0.0.0/16 X

11. Now you can do "PING" in the "test instance" to check, and now it will communicate with "prod-instance".

From 198.168.0.178 icmp_seq=9 Destination Host Unreachable
^C
--- 198.168.0.178 ping statistics ---
10 packets transmitted, 0 received, +9 errors, 100% packet loss, time 9246ms
pipe 3
ubuntu@ip-10-0-0-154:~\$ ping 198.168.0.178
PING 198.168.0.178 (198.168.0.178) 56(84) bytes of data.
64 bytes from 198.168.0.178: icmp_seq=1 ttl=64 time=0.742 ms
64 bytes from 198.168.0.178: icmp_seq=2 ttl=64 time=0.570 ms
64 bytes from 198.168.0.178: icmp_seq=3 ttl=64 time=0.494 ms
64 bytes from 198.168.0.178: icmp_seq=4 ttl=64 time=0.615 ms
64 bytes from 198.168.0.178: icmp_seq=5 ttl=64 time=0.505 ms
64 bytes from 198.168.0.178: icmp_seq=6 ttl=64 time=0.481 ms
64 bytes from 198.168.0.178: icmp_seq=7 ttl=64 time=0.451 ms
64 bytes from 198.168.0.178: icmp_seq=8 ttl=64 time=0.408 ms
64 bytes from 198.168.0.178: icmp_seq=9 ttl=64 time=0.578 ms
^C
--- 198.168.0.178 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8217ms
rtt min/avg/max/mdev = 0.408/0.533/0.742/0.097 ms
ubuntu@ip-10-0-0-154:~\$ []

i-0d70a6eb5c7dcf105 (test-instance)
PublicIPs: 3.17.156.106 PrivateIPs: 10.0.0.154

12. Now you can do “PING” in the “prod-instance” to check, and now it will communicate with “test-instance”.

ubuntu@ip-198-168-0-178:~\$ ping 10.0.0.154
PING 10.0.0.154 (10.0.0.154) 56(84) bytes of data.
64 bytes from 10.0.0.154: icmp_seq=1 ttl=64 time=0.442 ms
64 bytes from 10.0.0.154: icmp_seq=2 ttl=64 time=0.466 ms
64 bytes from 10.0.0.154: icmp_seq=3 ttl=64 time=0.562 ms
64 bytes from 10.0.0.154: icmp_seq=4 ttl=64 time=0.622 ms
64 bytes from 10.0.0.154: icmp_seq=5 ttl=64 time=0.603 ms
64 bytes from 10.0.0.154: icmp_seq=6 ttl=64 time=0.493 ms
64 bytes from 10.0.0.154: icmp_seq=7 ttl=64 time=0.428 ms
64 bytes from 10.0.0.154: icmp_seq=8 ttl=64 time=0.419 ms
^C64 bytes from 10.0.0.154: icmp_seq=9 ttl=64 time=0.407 ms
64 bytes from 10.0.0.154: icmp_seq=10 ttl=64 time=0.403 ms
64 bytes from 10.0.0.154: icmp_seq=11 ttl=64 time=0.444 ms
64 bytes from 10.0.0.154: icmp_seq=12 ttl=64 time=0.463 ms
^C
--- 10.0.0.154 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11257ms
rtt min/avg/max/mdev = 0.403/0.479/0.622/0.072 ms
ubuntu@ip-198-168-0-178:~\$ []

i-0fb581fae68f99950 (prod-instance)
PublicIPs: 18.117.186.238 PrivateIPs: 198.168.0.178

Enjoy :)