

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

Create a vpc with the name my-vpc.

The screenshot shows the 'Create VPC' configuration page in the AWS Management Console. The top navigation bar includes the AWS logo, a search bar with the placeholder 'Search [Alt+S]', and a breadcrumb trail: 'VPC > Your VPCs > Create VPC'. The main section is titled 'Create VPC' with an 'Info' link. A descriptive text states: 'A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.' Below this, the 'VPC settings' section is titled 'VPC settings'. It contains a 'Resources to create' dropdown where 'VPC only' is selected. A 'Name tag - optional' field contains the value 'my-vpc'. Under 'IPv4 CIDR block', 'IPv4 CIDR manual input' is selected, and the CIDR block '10.0.0.0/16' is entered. Under 'IPv6 CIDR block', 'No IPv6 CIDR block' is selected. The entire form is contained within a light gray box.

Create a public subnet .

The screenshot shows the AWS VPC Subnets creation interface. At the top, there's a navigation bar with the AWS logo, a search bar containing 'Search' and a keyboard shortcut '[Alt+S]'. Below the navigation, the path 'VPC > Subnets > Create subnet' is shown. The main section is titled 'Associated VPC CIDRs' and lists 'IPv4 CIDRs' as '10.0.0.0/16'. The next section, 'Subnet settings', is titled 'Specify the CIDR blocks and Availability Zone for the subnet.' It contains three fields: 'Subnet name' (input: 'my-public-subnet'), 'Availability Zone' (input: 'Europe (Stockholm) / eun1-az1 (eu-north-1a)'), and 'IPv4 VPC CIDR block' (input: '10.0.0.0/16').

## Create a private subnet.

This screenshot shows the same 'Create a private subnet.' interface as the previous one, but with different input values. In the 'Subnet name' field, 'my-private-subnet' is entered. In the 'Availability Zone' dropdown, 'No preference' is selected. In the 'IPv4 VPC CIDR block' dropdown, '10.0.0.0/16' is chosen. A new field, 'IPv4 subnet CIDR block', is present with the value '10.0.16.0/20' and a note indicating it covers '4,096 IPs'.

## 2. Enable VPC peering for cross-region.

Click on peering connections and give the details and give cross-region and vpc id of accepter and create peering.

The screenshot shows the 'Create peering connection' wizard. Step 1: 'Peering connection settings'. It asks for a name (optional) and has a text input field with 'ohio-peering'. Step 2: 'Select a local VPC to peer with'. It shows a dropdown for 'VPC ID (Requester)' set to 'vpc-0721df79a41acf611 (default)'. Step 3: 'VPC CIDRs for vpc-0721df79a41acf611 (default)'. A table lists one CIDR: 172.31.0.0/16, status 'Associated', and reason '-'. Step 4: 'Select another VPC to peer with'. It shows two options: 'My account' (selected) and 'Another account'. A note says 'Accept the peering request in the another region.'

Accept the peering request in the another region.

The screenshot shows the 'Peering connections' list. It displays one connection named 'pcx-01459bd03baf45920' with status 'Pending acceptance'. The requester VPC is 'vpc-0721df79a41acf611'. On the right, there's an 'Actions' menu with options: 'View details', 'Accept request' (which is highlighted), 'Reject request', 'Edit DNS settings', 'Manage tags', and 'Delete peering connection'. Below the list, a detailed view for 'pcx-01459bd03baf45920' is shown with tabs for 'Details', 'DNS', 'Route tables', and 'Tags'. The 'Details' tab shows Requester owner ID and Acceptor owner ID, both linking to 'vpc-0721df79a41acf611'. The VPC Peering connection ARN is also listed.

Go to routetable and edit with another region's cidr range.do same process in the another region also.

Destination	Target	Status	Propagated	Route Origin
172.31.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	Internet Gateway	Active	No	CreateRoute
10.0.0.0/16	Peering Connection pcx-01459bd03baf45920	-	No	CreateRoute
Add route				

Connect with public ip and ping another region's private ip to check the connection.

```
[root@ip-172-31-32-120 ~]# ping 10.0.15.89
PING 10.0.15.89 (10.0.15.89) 56(84) bytes of data.
64 bytes from 10.0.15.89: icmp_seq=416 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=417 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=418 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=419 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=420 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=421 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=422 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=423 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=424 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=425 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=426 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=427 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=428 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=429 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=430 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=431 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=432 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=433 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=434 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=435 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=436 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=437 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=438 ttl=127 time=13.7 ms
```

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

Create peering connection and give details of another region of different account.

The screenshot shows the 'Create peering connection' page in the AWS VPC console. At the top, there's a navigation bar with the AWS logo, a search bar, and a 'Create peering connection' button. Below the navigation, the breadcrumb path shows 'VPC > Peering connections > Create peering connection'. The main section is titled 'Create peering connection' with a sub-section 'Peering connection settings'. It includes fields for 'Name - optional' (containing 'devendra-peering'), 'Select a local VPC to peer with' (with 'VPC ID (Requester)' set to 'vpc-0721df79a41acf611 (default)'), and 'VPC CIDRs for vpc-0721df79a41acf611 (default)' (listing '172.31.0.0/16' with status 'Associated'). Below this, there's a section 'Select another VPC to peer with' with an 'Account' dropdown showing 'Another account' selected. A note at the bottom says 'Your VPC peering connection (pcx-018d9775539cbe1e1) has been established.'

Accept the request from receiver's account.

The screenshot shows the 'Peering connections' list page in the AWS VPC console. It displays a single peering connection entry with the status 'Established'. The 'Actions' button is highlighted in blue. A green banner at the top of the list indicates that the connection has been established. The list includes columns for 'CIDR', 'Status', and 'Status reason'. A search bar at the bottom allows filtering by attribute or tag.

Add routes in sender's account and receiver's account.

The screenshot shows the 'Edit routes' section of the AWS VPC console. It lists several routes:

- Destination: 172.31.0.0/16, Target: local, Status: Active, Propagated: No, Route Origin: CreateRouteTable.
- Destination: Q\_ 0.0.0.0, Target: Internet Gateway, Status: Active, Propagated: No, Route Origin: CreateRoute.
- Destination: Q\_ 192.168.0.0/24, Target: Peering Connection, Status: -, Propagated: No, Route Origin: CreateRoute.
- Destination: Q\_ ppx-018d9775539cbe1e1, Target: Q\_ ppx-018d9775539cbe1e1 (devendra-peering), Status: -, Propagated: No, Route Origin: -.

Buttons at the bottom include 'Add route', 'Cancel', 'Preview', and 'Save changes'.

login with your public ip and ping<reciver's private I'd> to check the connectivity.

```
[root@ip-172-31-13-26 ~]# ping 10.0.2.10
PING 10.0.2.10 (10.0.2.10) 56(84) bytes of data.
64 bytes from 10.0.2.10: icmp_seq=1 ttl=126 time=2.32 ms
64 bytes from 10.0.2.10: icmp_seq=2 ttl=126 time=1.22 ms
64 bytes from 10.0.2.10: icmp_seq=3 ttl=126 time=1.13 ms
64 bytes from 10.0.2.10: icmp_seq=4 ttl=126 time=1.14 ms
64 bytes from 10.0.2.10: icmp_seq=5 ttl=126 time=1.13 ms
64 bytes from 10.0.2.10: icmp_seq=6 ttl=126 time=1.15 ms
64 bytes from 10.0.2.10: icmp_seq=7 ttl=126 time=1.13 ms
64 bytes from 10.0.2.10: icmp_seq=8 ttl=126 time=1.12 ms
```

## 4. Set up a VPC Transit Gateway.

Create 4 vpc's.

The screenshot shows the 'Your VPCs' page in the AWS VPC dashboard. It lists four VPCs:

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
default	vpc-0564861ba65055cc3	Available	Off	172.31.0.0/16	-
vpn1	vpc-03c8a8c0d50a003c9	Available	Off	10.0.0.0/28	-
vpn2	vpc-0b335a455a76659b5	Available	Off	10.0.1.0/28	-
vpn3	vpc-0a2c328e7a4fde62d	Available	Off	10.0.2.0/28	-

A message at the bottom states: 'VPCs: vpc-0564861ba65055cc3, vpc-03c8a8c0d50a003c9, vpc-0b335a455a76659b5, vpc-0a2c328e7a4fde62d'.

Create subnets to the vpc's.

You have successfully created 1 subnet: subnet-06a1c6459d5b9ae38

Subnets (5) <a href="#">Info</a>						
<a href="#">Actions</a> <a href="#">Create subnet</a>						
	Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
<input type="checkbox"/>	pub-subnet-default	<a href="#">subnet-02560f8bc88fe708a</a>	<span>Available</span>	<a href="#">vpc-0564861ba65055cc3   defa...</a>	<input type="checkbox"/> Off	172.31.0.0/16
<input type="checkbox"/>	pri-subnet-default	<a href="#">subnet-0d4ea6ce22b4e7e6c</a>	<span>Available</span>	<a href="#">vpc-0564861ba65055cc3   defa...</a>	<input type="checkbox"/> Off	172.31.16.0/16
<input type="checkbox"/>	vpc1-subnet	<a href="#">subnet-0d95feb509064af46</a>	<span>Available</span>	<a href="#">vpc-03c8a8c0d50a003c9   vpc1</a>	<input type="checkbox"/> Off	10.0.0.0/24
<input type="checkbox"/>	vpc2-subnet	<a href="#">subnet-039d3584bd67b82db</a>	<span>Available</span>	<a href="#">vpc-0b335a455a76659b5   vpc2</a>	<input type="checkbox"/> Off	10.0.1.0/24
<input type="checkbox"/>	vpc3-subnet	<a href="#">subnet-06a1c6459d5b9ae38</a>	<span>Available</span>	<a href="#">vpc-0a2c328e7a4fde62d   vpc3</a>	<input type="checkbox"/> Off	10.0.2.0/24

## Create a transit gateway.

You successfully created tgw-0f4b076d1cff84df5 / my-transitgateway.

You can visualize and monitor your Transit Gateway(s) from the AWS Network Manager [ ] . Register your Transit Gateway by creating a global network [ ] to get started.

Transit gateways (1/1) <a href="#">Info</a>						
<a href="#">Actions</a> <a href="#">Create</a>						
	Name	Transit gateway ID	State			
<input checked="" type="checkbox"/>	my-transitgateway	<a href="#">tgw-0f4b076d1cff84df5</a>	<input type="checkbox"/> Pending			

## Create 4 transit gateway attachments and attach to 4 vpc's.

You successfully created VPC attachment tgw-attach-0be6fe8d40f1c0282 / vpc3-tga.

Transit gateway attachments (4/4) <a href="#">Info</a>						
<a href="#">Actions</a> <a href="#">Create transit gateway attachment</a>						
	Name	Transit gateway attachment ID	Transit gateway ID	State	Resource type	Resource ID
<input checked="" type="checkbox"/>	vpc1-tga	<a href="#">tgw-attach-021713307685d0ea0</a>	<a href="#">tgw-0f4b076d1cff84df5</a>	<input type="checkbox"/> Pending	VPC	<a href="#">vpc-03c8a8c0d50a003c9</a>
<input checked="" type="checkbox"/>	vpc2-tga	<a href="#">tgw-attach-05a10c5c0760a1751</a>	<a href="#">tgw-0f4b076d1cff84df5</a>	<input type="checkbox"/> Pending	VPC	<a href="#">vpc-0b335a455a76659b5</a>
<input checked="" type="checkbox"/>	vpc3-tga	<a href="#">tgw-attach-0be6fe8d40f1c0282</a>	<a href="#">tgw-0f4b076d1cff84df5</a>	<input type="checkbox"/> Pending	VPC	<a href="#">vpc-0a2c328e7a4fde62d</a>
<input checked="" type="checkbox"/>	default-tga	<a href="#">tgw-attach-0e1a25d309a54ff95</a>	<a href="#">tgw-0f4b076d1cff84df5</a>	<span>Available</span>	VPC	<a href="#">vpc-0564861ba65055cc3</a>

Transit gateway attachment IDs

tgw-attach-021713307685d0ea0, tgw-attach-05a10c5c0760a1751, tgw-attach-0be6fe8d40f1c0282, tgw-attach-0e1a25d309a54ff95

## Create 4 instances for 4 vpc's.

**Instances (4/6) Info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public
pri-ec2	i-05f85f900276ae46e	Terminated	t3.micro	-	<a href="#">View alarms +</a>	us-west-1a	-
<input checked="" type="checkbox"/> vpc3-ec2	i-0aa3f349b0aa2fb9a	Running	t3.micro	Initializing	<a href="#">View alarms +</a>	us-west-1a	-
<input checked="" type="checkbox"/> vpc1-ec2	i-012697a6b4e6ba3cc	Running	t3.micro	Initializing	<a href="#">View alarms +</a>	us-west-1a	-
<input type="checkbox"/> pub-ec2	i-0e782b07ee52add85	Terminated	t3.micro	-	<a href="#">View alarms +</a>	us-west-1c	-
<input checked="" type="checkbox"/> default-ec2	i-04c06ab5dbc089490	Running	t3.micro	3/3 checks passed	<a href="#">View alarms +</a>	us-west-1c	ec2-5
<input checked="" type="checkbox"/> vpc2-ec2	i-034198f5fabdefc00	Running	t3.micro	Initializing	<a href="#">View alarms +</a>	us-west-1c	-

**4 instances selected**

[Monitoring](#)

[Configure CloudWatch agent](#)

Go to route table and edit the routes give permissions for all vpc's.

**VPC dashboard**

**Route tables**

**rtb-0c3340b3f557c1b0d**

**Details**

Updated routes for rtb-0c3340b3f557c1b0d / default-rt successfully

**Routes (5)**

Destination	Target	Status	Propagated	Route Origin
0.0.0.0/0	igw-021348c6279231955	Active	No	Create Route
10.0.0.0/28	tgw-0f4b076d1cff84df5	Active	No	Create Route
10.0.1.0/28	tgw-0f4b076d1cff84df5	Active	No	Create Route
10.0.2.0/28	tgw-0f4b076d1cff84df5	Active	No	Create Route
172.31.0.0/16	local	Active	No	Create Route Table

We need to create internet gateways for 4 vpc's.

**Internet gateways (1/2) Info**

Name	Internet gateway ID	State	VPC ID
<input checked="" type="checkbox"/> default-internetgateway	igw-021348c6279231955	Attached	vpc-0564861ba65055cc3   default
<input type="checkbox"/> vpc2-internetgateway	igw-04effb92417c5492c	Attached	vpc-0b335a455a76659b5   vpc2

**igw-021348c6279231955**

Login with public ip address and ping any instance private ip address.

```
[root@ip-172-31-13-26 ~]# ping 10.0.2.10
PING 10.0.2.10 (10.0.2.10) 56(84) bytes of data.
64 bytes from 10.0.2.10: icmp_seq=1 ttl=126 time=2.32 ms
64 bytes from 10.0.2.10: icmp_seq=2 ttl=126 time=1.22 ms
64 bytes from 10.0.2.10: icmp_seq=3 ttl=126 time=1.13 ms
64 bytes from 10.0.2.10: icmp_seq=4 ttl=126 time=1.14 ms
64 bytes from 10.0.2.10: icmp_seq=5 ttl=126 time=1.13 ms
64 bytes from 10.0.2.10: icmp_seq=6 ttl=126 time=1.15 ms
64 bytes from 10.0.2.10: icmp_seq=7 ttl=126 time=1.13 ms
64 bytes from 10.0.2.10: icmp_seq=8 ttl=126 time=1.12 ms
^C
--- 10.0.2.10 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7009ms
rtt min/avg/max/mdev = 1.121/1.292/2.321/0.389 ms
[root@ip-172-31-13-26 ~]# ping 10.0.0.6
PING 10.0.0.6 (10.0.0.6) 56(84) bytes of data.
64 bytes from 10.0.0.6: icmp_seq=1 ttl=126 time=1.93 ms
64 bytes from 10.0.0.6: icmp_seq=2 ttl=126 time=1.08 ms
64 bytes from 10.0.0.6: icmp_seq=3 ttl=126 time=1.08 ms
64 bytes from 10.0.0.6: icmp_seq=4 ttl=126 time=1.09 ms
64 bytes from 10.0.0.6: icmp_seq=5 ttl=126 time=1.05 ms
64 bytes from 10.0.0.6: icmp_seq=6 ttl=126 time=1.07 ms
^C
--- 10.0.0.6 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 1.048/1.217/1.931/0.319 ms
```

## 5. Set up a VPC Endpoint.

Open ec2 instance with public ip address.

```
MUJU_SK@DESKTOP-LU541U4 MINGW64 ~/Downloads
$ ssh -i california.pem ec2-user@54.183.26.154
The authenticity of host '54.183.26.154 (54.183.26.154)' can't be established.
ED25519 key fingerprint is SHA256:zLM1ROVXJkpgA5yITcCKbmqx5QeERCIiEu1lD3zkvzk.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.183.26.154' (ED25519) to the list of known hosts.

      #_
     ~\_\ #####
     ~~ \####\      Amazon Linux 2023
     ~~ \|##|
     ~~ \#/ ,-->  https://aws.amazon.com/linux/amazon-linux-2023
     ~~ .-. / \
     ~~ / ,--/ \
     _/m/ |   |

[ec2-user@ip-172-31-6-12 ~]$ sudo su-
sudo: su-: command not found
[ec2-user@ip-172-31-6-12 ~]$ sudo su -
[root@ip-172-31-6-12 ~]# aws --version
aws-cli/2.27.57 Python/3.9.23 Linux/6.1.150-174.273.amzn2023.x86_64 source/x86_64.amzn.2023
[root@ip-172-31-6-12 ~]# |
```

Create an end point for public instance.

## Create endpoint Info

Create the type of VPC endpoint that supports the service, service network or resource to which you want to connect.

### Endpoint settings

Specify a name and select the type of endpoint.

#### Name tag - optional

Creates a tag with a key of 'Name' and a value that you specify. Tags help you find and manage your endpoint.

s3-endpoint

#### Type Info

Select a category

##### AWS services

Connect to services provided by Amazon with an Interface endpoint, or a Gateway endpoint

##### PrivateLink Ready partner service

Connect to SaaS services which have an Interface endpoint. Uses AWS Private

##### EC2 Instance Connect Endpoint

An elastic network interface that allows you to connect to resources in a private subnet

##### Resources

Connect to resources like Amazon Relational Database Service endpoint. Uses AWS Private

### Services (1/2)

Search

Service Name = com.amazonaws.us-west-1.s3

Service Name	Owner	Type
<input type="radio"/> com.amazonaws.us-west-1.s3	amazon	Interface
<input checked="" type="radio"/> com.amazonaws.us-west-1.s3	amazon	Gateway

### Network settings

Select the VPC in which to create the endpoint

#### VPC

Create the VPC endpoint in the VPC in the same AWS Region from which you will access a resource.

vpc-0564861ba65055cc3

### Route tables (1/2) Info

vpc-0564861ba65055cc3

**Route tables (1/2)** [Info](#) Search

<input type="checkbox"/>	Name	▼	Route Table ID	▼	Main
<input type="checkbox"/>	pub-route-table		<a href="#">rtb-0c3340b3f557c1b0d (pub-route-table)</a>		Yes
<input checked="" type="checkbox"/>	pri-routetable		<a href="#">rtb-0b334e4702f6e460c (pri-routetable)</a>		No

- When you use an endpoint, the source IP addresses from your instances in your affected subnets for accessing the AWS service may be dropped. Ensure that you do not use endpoints to access services that require public IP addresses.

[rtb-0b334e4702f6e460c](#) **Policy** [Info](#)

PC endpoint policy controls access to the service.

 **Full access**

Allow access by any user or service within the VPC using credentials from any Amazon Web Services accounts to any resources in the VPC.

We can see endpoint is connected to bucket.

```
[root@ip-172-31-6-12 .aws]# cd ~
[root@ip-172-31-6-12 ~]# aws s3 ls
2025-09-23 12:39:35 my-s3bucket0123
[root@ip-172-31-6-12 ~]# |
```