

# Setting up Transit Gateway and VPC Endpoints for a MultiVPC Architecture.

## SCENARIO:

A large organization is migrating its on-premises infrastructure to the AWS cloud.

The organization's architecture involves multiple VPCs for different departments and applications, each requiring secure communication with centralized services and external resources.

The IT team needs to design and implement a scalable and efficient network

architecture to accommodate the organization's growth and ensure robust connectivity between VPCs and external services.

Create 3 VPC's one for public and 2 for private.

The screenshot shows the AWS VPC settings page. At the top, there is a navigation bar with the AWS logo, a search bar, and a 'Create VPC' button. Below the navigation bar, a sub-navigation bar shows the path: VPC > Your VPCs > Create VPC. A note below the path states: 'A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.' The main section is titled 'VPC settings'. Under 'Resources to create', the 'VPC only' option is selected. In the 'Name tag - optional' field, the value 'vpc-01' is entered. Under 'IPv4 CIDR block', the CIDR block '172.168.0.0/24' is specified. Under 'IPv6 CIDR block', the 'No IPv6 CIDR block' option is selected. The page has a light gray background with white text and blue highlights for selected options.

Your VPCs (3/5) [Info](#)

Last updated less than a minute ago

[Actions](#) [Create VPC](#)

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
my-vpc	vpc-0b3ab1c3d8a1c0bdb	Available	Off	10.0.0.0/16	-
<input checked="" type="checkbox"/> vpc-01	vpc-0af726d176c49904e	Available	Off	172.168.0.0/24	-
<input checked="" type="checkbox"/> vpc-02	vpc-0b96ede37d17a943c	Available	Off	10.0.1.0/28	-
<input checked="" type="checkbox"/> vpc-03	vpc-0f6b4aea545ed4a65	Available	Off	10.0.0.0/28	-

VPCs: vpc-0af726d176c49904e, vpc-0b96ede37d17a943c, vpc-0f6b4aea545ed4a65

## Create subnets for the vpc's that you are created.

[VPC](#) > Subnets

VPC dashboard [AWS Global View](#) [Filter by VPC](#)

Virtual private cloud [Your VPCs](#) [Subnets](#) [Route tables](#) [Internet gateways](#) [Egress-only internet gateways](#) [DHCP option sets](#)

You have successfully created 1 subnet: subnet-0578113227cc73af7

Subnets (3/8) [Info](#)

Last updated less than a minute ago

[Actions](#) [Create subnet](#)

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
<input checked="" type="checkbox"/> vpc-1-subnet	subnet-0420ecddb617afb9	Available	vpc-0af726d176c49904e   vpc-01	Off	172.168.0.0
<input checked="" type="checkbox"/> vpc-2-subnet	subnet-03044c10e4caa8517	Available	vpc-0b96ede37d17a943c   vpc-02	Off	10.0.1.0/28
<input checked="" type="checkbox"/> vpc-3-subnet	subnet-0578113227cc73af7	Available	vpc-0f6b4aea545ed4a65   vpc-03	Off	10.0.0.0/28

Subnets: subnet-0420ecddb617afb9, subnet-03044c10e4caa8517, subnet-0578113227cc73af7

## Create a transit gateway

[VPC](#) > [Transit gateways](#) > Create transit gateway

### Create transit gateway [Info](#)

A transit gateway (TGW) is a network transit hub that interconnects attachments (VPCs and VPNs) within the same AWS account or across AWS accounts.

#### Details - optional

##### Name tag

Creates a tag with the key set to Name and the value set to the specified string.

my-tg

##### Description [Info](#)

Set the description of your transit gateway to help you identify it in the future.

description

#### Configure the transit gateway

##### Amazon side Autonomous System Number (ASN) [Info](#)

ASN

DNS support [Info](#)

Security Group Referencing support [Info](#)

VPN ECMP support [Info](#)

Default route table association [Info](#)

Create transit gateway attachments for three vpc's. and give the subnet associations, for public subnet attach the internet gateway.

The screenshot shows the 'Transit gateway attachments' section of the AWS Management Console. It lists three attachments:

Name	Transit gateway attachment ID	Transit gateway ID	State	Resource type	Resource ID
vpc-tga1	tgw-attach-00142779ec62afb98	tgw-04639e643a6fbbe0d	Pending	VPC	vpc-0af7
vpc-tga2	tgw-attach-05cadd678e417fa1a	tgw-04639e643a6fbbe0d	Pending	VPC	vpc-0b9f
vpc-tga3	tgw-attach-0efa3c2dcc6557a80	tgw-04639e643a6fbbe0d	Pending	VPC	vpc-0f6b

Create 3 instances with the 3 vpc's.

The screenshot shows the 'Instances' section of the AWS Management Console. It lists four instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
my-vpc-1	i-06fe0f2d7d054cf38	Running	t3.micro	Initializing	<a href="#">View alarms</a>	eu-north-1c	-
vpc-3-instance	i-00299d1d857ebefb	Running	t3.micro	-	<a href="#">View alarms</a>	eu-north-1c	-
peering-europe	i-001bcffa94af95490	Stopped	t3.micro	-	<a href="#">View alarms</a>	eu-north-1a	-
vpc-2-instance	i-00a2341e815728c75	Running	t3.micro	Initializing	<a href="#">View alarms</a>	eu-north-1a	-

Three instances are selected: my-vpc-1, vpc-3-instance, and vpc-2-instance.

Go to first subnet and edit routes add remaining two vpc's cidr range .

Login with public ip of 1 instance and from that public to ping another account's private ip address.

```
MUJU SK@DESKTOP-LU541U4 MINGW64 ~/Downloads
$ ssh -i "AWS.pem" ec2-user@16.16.233.250
The authenticity of host '16.16.233.250 (16.16.233.250)' can't be established.
ED25519 key fingerprint is SHA256:ZHCp0387IS01bgjyOsGYpr7tAnArYC3C2CN82dyoja4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '16.16.233.250' (ED25519) to the list of known hosts.

      #_
  ~\_\_ #####_          Amazon Linux 2023
  ~~ \_\_\#\#\#\_\_
  ~~   \#\#\#
  ~~     \#/   __  https://aws.amazon.com/linux/amazon-linux-2023
  ~~       V~'  '->
  ~~~           /
  ~~.  /  _/
  _/m/ , _/
[ec2-user@ip-10-0-0-5 ~]$ sudo su -
[root@ip-10-0-0-5 ~]# ping 172.168.0.161
PING 172.168.0.161 (172.168.0.161) 56(84) bytes of data.
64 bytes from 172.168.0.161: icmp_seq=41 ttl=126 time=0.629 ms
64 bytes from 172.168.0.161: icmp_seq=42 ttl=126 time=0.288 ms
64 bytes from 172.168.0.161: icmp_seq=43 ttl=126 time=0.290 ms
64 bytes from 172.168.0.161: icmp_seq=44 ttl=126 time=0.285 ms
64 bytes from 172.168.0.161: icmp_seq=45 ttl=126 time=0.302 ms
64 bytes from 172.168.0.161: icmp_seq=46 ttl=126 time=0.290 ms
64 bytes from 172.168.0.161: icmp_seq=47 ttl=126 time=0.291 ms
64 bytes from 172.168.0.161: icmp_seq=48 ttl=126 time=0.295 ms
```

## 2. Configure VPC endpoints to securely access AWS services without internet gateways or NAT gateways, ensuring data privacy and minimizing exposure to external threats.

Launch one public instance and one private instance using private subnet and public subnet.

The screenshot shows the AWS CloudWatch Metrics interface. At the top, there's a table of EC2 instances with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. Two instances are selected: 'private-instance' and 'public-instance'. Below the table, it says '2 instances selected'. Under the 'Monitoring' tab, there are four metrics: CPU utilization (%), Network in (bytes), Network out (bytes), and Network packets in (count). There are also buttons for 'Configure CloudWatch agent', 'Investigate with AI - new', and 'Explore related'.

Go to S3 and create a bucket.

The screenshot shows the AWS S3 Buckets interface. It displays a single bucket named 'vpc-challenge7'. The 'Objects' tab is selected, showing 0 objects. A message indicates that objects are fundamental entities stored in Amazon S3. There are buttons for 'Upload' and 'Actions' (Copy S3 URI, Copy URL, Download, Open, Delete). Below the table, it says 'No objects' and 'You don't have any objects in this bucket.' with a 'Upload' button.

Connect ssh with the public instance

```
MUJU SK@DESKTOP-LU541U4 MINGW64 ~/Downloads
$ ssh -i "AWS.pem" ec2-user@ec2-13-60-99-219.eu-north-1.compute.amazonaws.com
,
#_
~\_ ####_ Amazon Linux 2023
~~ \####\ https://aws.amazon.com/linux/amazon-linux-2023
~~ \|##|
~~ \|#/ __ V~` ,`->
~~| / |
~~| / |
/_m/` / |
Last login: Wed Oct 1 09:37:02 2025 from 103.143.169.218
[ec2-user@ip-172-31-36-222 ~]$ sudo su -
Last login: Wed Oct 1 09:37:20 UTC 2025 on pts/1
[root@ip-172-31-36-222 ~]# 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-172-31-36-222 ~]# |
```

Go to aws configure and give the security key and password .

```
[root@ip-172-31-36-222 ~]# aws configure
AWS Access Key ID [None]: AKIATNTADWLTSYF63H5C
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json
[root@ip-172-31-36-222 ~]# aws s3 ls
2025-09-29 10:11:05 bucketnew57
2025-09-29 11:34:52 n-virginia-bucket6
2025-10-01 09:35:38 vpc-challenge7
[root@ip-172-31-36-222 ~]# |
```

Copy the pem key in local and create a file and paste the pem key in that file.

Add permission to that file chmod 400 test.pem

Ssh -i test.pem ec2-user@<private ip>

```
[root@ip-172-31-36-222 ~]# vi test.pem
[root@ip-172-31-36-222 ~]# chmod 400 test.pem
[root@ip-172-31-36-222 ~]# ssh -i test.pem ec2-user@172.31.36.222
The authenticity of host '172.31.36.222 (172.31.36.222)' can't be established.
ED25519 key fingerprint is SHA256:4w+Vu2FUk9B7oy077R7k9udiFTt6LnLXKRF2teKboSE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.36.222' (ED25519) to the list of known hosts.

,
#          Amazon Linux 2023
~~ \_\#\#\#
~~ \_\#\#\#\_
~~   \#\#
~~     \#/ , __> https://aws.amazon.com/linux/amazon-linux-2023
~~
~~ ._. / \
~~ / \_ / \
~~ /m/ , / \
Last login: wed oct  1 09:38:21 2025 from 103.143.169.218
[ec2-user@ip-172-31-36-222 ~]$ |
```

After that aws configure.

```
Last login: wed oct  1 09:38:21 2025 from 103.143.169.218
[ec2-user@ip-172-31-36-222 ~]$ aws configure
AWS Access Key ID [None]: AKIATNTADWLTSYF63H5C
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json
[ec2-user@ip-172-31-36-222 ~]$ |
```

- Now go to vpc
- Navigate to end points
- Select aws service
- Then select s3 and end point type express
- Then select gateway
- Create endpoint the endpoint has been created
- Now you can access the s3 from the private instance : aws

s3 ls

Create an end point.

The screenshot shows the AWS VPC Endpoints console with the following details:

**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.

my-endpoint

**Type**

Select a category

<input checked="" type="radio"/> AWS services Connect to services provided by Amazon with an Interface endpoint, or a Gateway endpoint	<input type="radio"/> PrivateLink Ready partner services Connect to SaaS services which have AWS Service Ready designation with an Interface endpoint. Uses AWS PrivateLink	<input type="radio"/> AWS Marketplace Connect to SaaS with an Interface endpoint
<input type="radio"/> EC2 Instance Connect Endpoint An elastic network interface that allows you to connect to resources in a private subnet	<input type="radio"/> Resources Connect to resources like Amazon Relational Database Services (RDS) with a Resource endpoint. Uses AWS PrivateLink	<input type="radio"/> Service networks Connect to VPC I Uses AWS Privat
<input type="radio"/> Endpoint services that use NLBs and GWLBs Find services shared with you by service name. Connect to a Network LoadBalancer (NLB) service with an Interface endpoint or to a Gateway LoadBalancer (GWLB) service with a Gateway Load Balancer endpoint		

**Services (1/1)**

Service Name = com.amazonaws.eu-north-1.s3express

Clear filters

Service Name	Owner	Type	Region
com.amazonaws.eu-north-1.s3express	amazon	Gateway	eu-north-

**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-0db9d4fb39c8bc078 (default vpc)

**Route tables (1/3) [Info](#)**

Name	Route Table ID	Main	Associated Id
-	<a href="#">rtb-0b42283383e9a976a</a>	Yes	<a href="#">subnet-0446b22818aa451b1 (default-public-subnet)</a>
default-public-subnet	<a href="#">rtb-08080318cb07dc856 (default-public-subnet)</a>	No	<a href="#">subnet-0cc304f411ffbe14c (default-public-subnet)</a>
<input checked="" type="checkbox"/> default-private-subnet	<a href="#">rtb-04fa92c9c736f604a (default-private-subnet)</a>	No	<a href="#">subnet-05523948fb6f25c67 (default-private-subnet)</a>

When you use an endpoint, the source IP addresses from your instances in your affected subnets for accessing the AWS service in the same region will be private IP addresses, not public IP addresses. Existing endpoints from your affected subnets to the AWS service that use public IP addresses may be dropped. Ensure that you don't have critical tasks running when you create or modify an endpoint.

[rtb-04fa92c9c736f604a](#) [X](#)

#### Policy [Info](#)

VPC endpoint policy controls access to the service.

Full access

If we see in the routetable the endpoint is created for private subnet.

**VPC dashboard < Route tables**

AWS Global View [Edit](#)

Filter by VPC [▼](#)

**Virtual private cloud**

- Your VPCs
- Subnets
- Route tables**
- Internet gateways
- Egress-only internet gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- NAT gateways
- Peering connections

**Security**

- Network ACLs
- Security groups

**Route tables (1/6) [Info](#)**

Name	Route table ID	Explicit subnet assoc...	Edge associations	Main	VPC
-	<a href="#">rtb-0b42283383e9a976a</a>	-	-	Yes	<a href="#">vpc-0db9d</a>
pri-route-table-task0	<a href="#">rtb-00d57739fabcdf9da</a>	-	-	No	<a href="#">vpc-0b5ab</a>
-	<a href="#">rtb-0729da802662a0c5f</a>	-	-	Yes	<a href="#">vpc-0b5ab</a>
<input checked="" type="checkbox"/> default-private-subnet	<a href="#">rtb-04fa92c9c736f604a</a>	<a href="#">subnet-05523948fb6f25c67</a>	-	No	<a href="#">vpc-0db9d</a>
default-public-subnet	<a href="#">rtb-08080318cb07dc856</a>	<a href="#">subnet-0cc304f411ffbe14c</a>	-	No	<a href="#">vnr-0rth9d</a>

**rtb-04fa92c9c736f604a / default-private-subnet**

Details	Routes	Subnet associations	Edge associations	Route propagation	Tags
<b>Routes (2)</b>					
<a href="#">Filter routes</a>					
Destination	Target	Status	Propagated	Route Origin	
<a href="#">pl-05ed086c</a>	<a href="#">vpc-e-04dadd688d236484f</a>	<a href="#">Active</a>	No	Create Route	
172.31.0.0/16	local	<a href="#">Active</a>	No	Create Route Table	

```
[root@ip-172-31-36-222 ~]# aws configure
AWS Access Key ID [None]: AKIATNTADWLTSYF63H5C
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json
[root@ip-172-31-36-222 ~]# aws s3 ls
2025-09-29 10:11:05 bucketnew57
2025-09-29 11:34:52 n-virginia-bucket6
2025-10-01 09:35:38 vpc-challenge7
[root@ip-172-31-36-222 ~]# |
```