

# TRANSIT GATEWAY

A transit gateway is a network transit hub that you can use to interconnect your virtual private clouds (VPCs) and on-premises networks. As your cloud infrastructure expands globally, inter-Region peering connects transit gateways together using the AWS Global Infrastructure. All network traffic between AWS data centers is automatically encrypted at the physical layer.

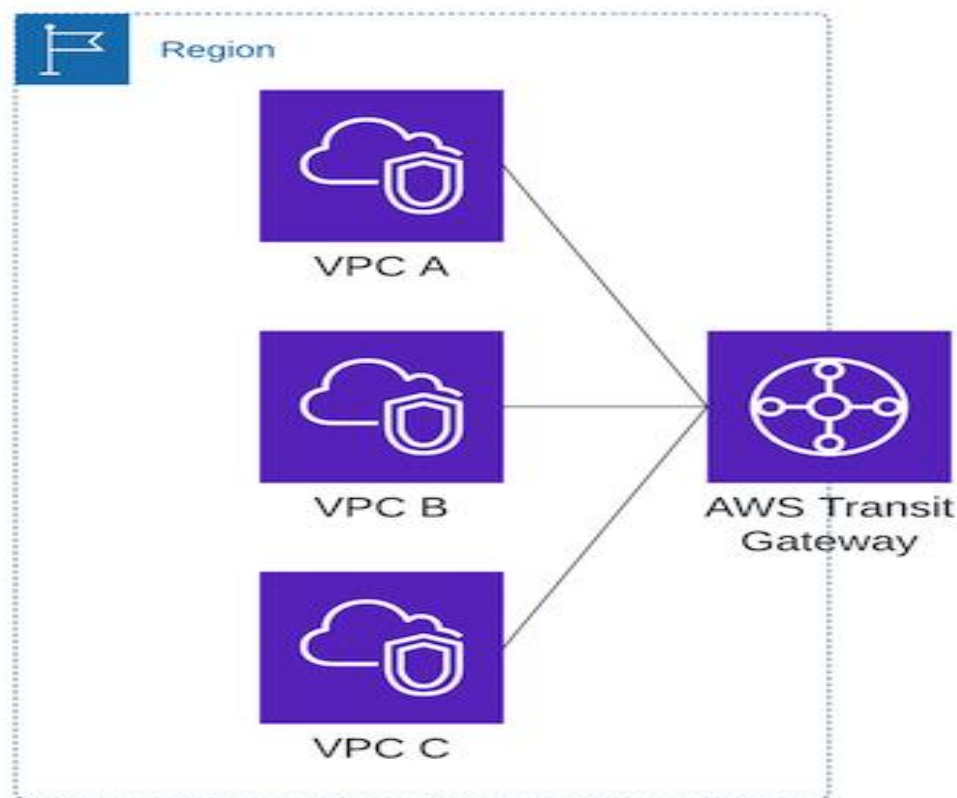


Fig1:Transit Gateway

## Components of Transit Gateway:

- **Transit Gateway** : Central hub that connects multiple VPCs and on-premises networks.
- **Attachments** : Connection between Transit Gateway and vcp's.
- **Route Tables** : Specifies how traffic should be routed between them.

## Steps to connect three vpc's to Transit Gateway :

- Open AWS Console and then open VPC.
- Click on Create VPC and select VPC .

The screenshot shows the 'Create VPC' page in the AWS Management Console. The breadcrumb navigation is 'VPC > Your VPCs > Create VPC'. The page title is '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.' The 'VPC settings' section includes: 'Resources to create' with 'VPC only' selected; 'Name tag - optional' with the value 'my-vpc-1'; 'IPv4 CIDR block' with 'IPv4 CIDR manual input' selected and the CIDR '120.10.0.0/16'; 'IPv6 CIDR block' with 'No IPv6 CIDR block' selected; and 'Tenancy' set to 'Default'. The 'Tags' section shows a key 'Name' and value 'my-vpc-1'. At the bottom are 'Cancel' and 'Create VPC' buttons.

- Create internal gateway to VPC-1 and attach to VPC-1

The screenshot shows the 'Create internet gateway' page in the AWS Management Console. The breadcrumb navigation is 'VPC > Internet gateways > Create internet gateway'. The page title is 'Create internet gateway' with an 'Info' link. A descriptive text states: '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.' The 'Internet gateway settings' section includes 'Name tag' with the value 'my-ig-1'. The 'Tags - optional' section shows a key 'Name' and value 'my-ig-1'. At the bottom are 'Cancel' and 'Create internet gateway' buttons.

VPC > Internet gateways > Attach to VPC (igw-06463519e6f14d960)

## Attach to VPC (igw-06463519e6f14d960) [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.

Q vpc-0ab6e9ad1f7b0749a



► AWS Command Line Interface command

Cancel

Attach internet gateway

- Create subnet-1-public for VPC-1

VPC > Subnets > Create subnet

## Create subnet [Info](#)

### VPC

#### VPC ID

Create subnets in this VPC.

vpc-0ab6e9ad1f7b0749a (my-vpc-1)

#### Associated VPC CIDRs

IPv4 CIDRs

120.10.0.0/16

### Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

#### Subnet 1 of 1

##### Subnet name

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

my-subnet-1

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.

Europe (Stockholm) / eu-north-1a

##### IPv4 VPC CIDR block [Info](#)

Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

120.10.0.0/16

##### IPv4 subnet CIDR block

120.10.1.0/24

256 IPs



#### Tags - optional

Key

Q Name



Value - optional

Q my-subnet-1



Remove

Add new tag

You can add 49 more tags.

Remove

Add new subnet

- Now we need to add the route tables , Click on Route Tables from the LHS panel and click on create route table

## Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

### Route table settings

#### Name - optional

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

#### VPC

The VPC to use for this route table.

### 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




You can add 49 more tags.



Next, we have to associate the subnet with routing table. For that select VPC-1-Route -> Click on Subnet associations -> Edit subnet associations, then select VPC-1 Public-Subnet1 -> Save associations.

### Edit subnet associations

Change which subnets are associated with this route table.

#### Available subnets (1/1)

<input checked="" type="checkbox"/>	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/>	my-subnet-1	subnet-09db6da59b940b4dc	120.10.1.0/24	-	rtb-01f951781bab97d69 / my-routetable-1

#### Selected subnets




Select the VPC-A-Route and go to Routes->Edit routes and add as per below , then click on save changes.

### Edit routes

Destination	Target	Status	Propagated
120.10.0.0/16	local	Active	No
<input type="text" value="0.0.0.0/0"/>	Internet Gateway	-	No
	igw-06463519e6f14d960		

- Now create another VPC with using of VPC and more

**Create VPC** [Info](#)

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances. Mouse over a resource to highlight the related resources.

**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 auto-generation** [Info](#)  
Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

☒ Auto-generate  
vpc-2

**IPv4 CIDR block** [Info](#)  
Determine the starting IP and the size of your VPC using CIDR notation.

120.20.0.0/16 65,536 IPs  
CIDR block size must be between /16 and /28.

**IPv6 CIDR block** [Info](#)  
☒ No IPv6 CIDR block  
☐ Amazon-provided IPv6 CIDR block

**Tenancy** [Info](#)  
Default

**Preview**

**Number of Availability Zones (AZs)** [Info](#)  
Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

1 2 3

Customize AZs

First availability zone  
eu-north-1b

**Number of public subnets** [Info](#)  
The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0 1

**Number of private subnets** [Info](#)  
The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

0 1 2

Customize subnets CIDR blocks

**NAT gateways (\$)** [Info](#)  
Choose the number of Availability Zones (AZs) in which to create NAT gateways. Note that there is a charge for each NAT gateway.

None In 1 AZ 1 per AZ

**VPC endpoints** [Info](#)  
Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

None S3 Gateway

**DNS options** [Info](#)  
☒ Enable DNS hostnames  
☒ Enable DNS resolution

Additional tags

Cancel Create VPC

- I Was Created VPC-2 in Stockholm (us north 1b) with 2 subnets (1 public, 1 Private) & 2 Route Tables(1 Public, 1 Private) with 1 internet gateway of IP (120.20.0.0/16).
- By the same process create VPC-3 with (120.30.0.0/16).

**Your VPCs (4)** [Info](#)

Q Search

<input type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/>	my-vpc-1	<a href="#">vpc-0ab6e9ad1f7b0749a</a>	Available	120.10.0.0/16	-
<input type="checkbox"/>	vpc-3-vpc	<a href="#">vpc-069103a92a4035f0a</a>	Available	120.30.0.0/16	-
<input type="checkbox"/>	-	<a href="#">vpc-00cc81bb0c5357758</a>	Available	172.31.0.0/16	-
<input type="checkbox"/>	vpc-2-vpc	<a href="#">vpc-085ea69c569ac6722</a>	Available	120.20.0.0/16	-

- Next Click on Transit Gateway and Create Transit Gateway with name (Transit Gateway-TG).

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.

**Description** [Info](#)  
Set the description of your transit gateway to help you identify it in the future.

### Configure the transit gateway

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

☒ **DNS support** [Info](#)

☒ **VPN ECMP support** [Info](#)

☒ **Default route table association** [Info](#)

☒ **Default route table propagation** [Info](#)

Transit gateways (1) [Info](#)

[Refresh](#) [Actions](#) [Create transit gateway](#)

<input type="checkbox"/>	Name <a href="#">↗</a>	Transit gateway ID	State
<input type="checkbox"/>	Transit Gateway-TG	<a href="#">tgw-0a349ec48ddeb766</a>	<span>Available</span>

- After creating Transit Gateway click on Transit Gateway Attachments as shown below.
- Create transit gateway attachments.

VPC > Transit gateway attachments > Create transit gateway attachment

## Create transit gateway attachment [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

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

**Transit gateway ID** [Info](#)

**Attachment type** [Info](#)

- Given name as TG-1.
- Select VPC-1 for attachmen.
- By the same process create attachments for TG-2 & TG-3 as Follow VPC-2 & VPC-3.

## Transit gateway attachments (5) [info](#)

Find transit gateway attachment by attribute or tag

	Name	Transit gateway attachment ID	Transit gateway ID	State	Resource type	Resource
	transit-gateway-2	<a href="#">tgw-attach-0a4e4c7ddb9e4d258</a>	<a href="#">tgw-0bad599222c11ba6b</a>	Deleted	Peering	tgw-C
	transit-gateway-3	<a href="#">tgw-attach-002a467e061193841</a>	<a href="#">tgw-0bad599222c11ba6b</a>	Available	VPC	<a href="#">vpc-0</a>
	transit-gateway-2	<a href="#">tgw-attach-044290778068f2e2c</a>	<a href="#">tgw-0bad599222c11ba6b</a>	Available	VPC	<a href="#">vpc-0</a>
	transitgateway-attac...	<a href="#">tgw-attach-073e54aa8eaa0f512</a>	<a href="#">tgw-0bad599222c11ba6b</a>	Deleted	VPC	<a href="#">vpc-0</a>
	transit-gateway-1	<a href="#">tgw-attach-0aa332da14166208e</a>	<a href="#">tgw-0bad599222c11ba6b</a>	Available	VPC	<a href="#">vpc-0</a>

- Next go to route tables and click on VPC-1.
- Click on Add Routes and add IP's of VPC-2 & VPC-3.

## Edit routes

Destination	Target	Status	Propagated	
120.10.0.0/16	local	Active	No	
	Q local			
Q 120.20.0.0/16	Transit Gateway	Active	No	<button>Remove</button>
	Q tgw-0bad599222c11ba6b			
Q 120.30.0.0/16	Transit Gateway	Active	No	<button>Remove</button>
	Q tgw-0bad599222c11ba6b			
Q 0.0.0.0/0	Internet Gateway	Active	No	<button>Remove</button>
	Q igw-06463519e6f14d960			

Add route

Cancel Preview Save changes

## rtb-01f951781bab97d69 / my-routetable-1

**Details**
[Info](#)

Route table ID  
rtb-01f951781bab97d69

VPC  
[vpc-0ab6e9ad1f7b0749a](#) | [my-vpc-1](#)

Main  
No

Owner ID  
980186824762

Explicit subnet associations  
[subnet-09db6da59b940b4dc](#) / [my-subnet-1](#)

Edge associations  
-

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

**Routes (4)**
Both

Filter routes

Destination	Target	Status	Propagated
0.0.0.0/0	<a href="#">igw-06463519e6f14d960</a>	Active	No
120.10.0.0/16	local	Active	No
120.20.0.0/16	<a href="#">tgw-0bad599222c11ba6b</a>	Active	No
120.30.0.0/16	<a href="#">tgw-0bad599222c11ba6b</a>	Active	No

- By configuring Route Tables Next we need to create 3 Instances for VPC-1, VPC-2 & VPC-3.
- Click on EC-2 instance. Then click on create instance.
- Give instance name as server-1 and select OS & Create Key Pair.
- Create Security Group & Add rules for SSH and HTTP as follows 22 & 80 port numbers.

- By the same process create instances for VPC-2 & VPC-3 with names as Server-2 & Server-3.

Key pair name - *required*

newba

Create new key pair

▼ Network settings

Info

VPC - *required*

Info

vpc-0ab6e9ad1f7b0749a (my-vpc-1)

120.10.0.0/16

Subnet

Info

subnet-09db6da59b940b4dc

my-subnet-1

VPC: vpc-0ab6e9ad1f7b0749a Owner: 980186824762  
Availability Zone: eu-north-1a IP addresses available: 249 CIDR: 120.10.1.0/24

Auto-assign public IP

Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups)

Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group

Select existing security group

Security group name - *required*

launch-wizard-4

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and \_-:/#@[]+=&~!|5\*

Description - *required*

Info

launch-wizard-4 created 2024-08-05T10:44:07.473Z

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Remove

Type

Info

ssh

Protocol

Info

TCP

Port range

Info

22

Number of instances

Info

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.5.2...read more  
ami-0b2777f4fd0d1712a

Virtual server type (instance type)

t3.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Launch instance

Review commands

## Inbound Security Group Rules

▼

Security group rule 1 (TCP, 22, 0.0.0.0/0)

Remove

Type | Info

ssh

Protocol | Info

TCP

Port range | Info

22

Source type | Info

Anywhere

Source | Info

Q Add CIDR, prefix list or security

0.0.0.0/0 X

Description - optional | Info

e.g. SSH for admin desktop

▼

Security group rule 2 (TCP, 80, 0.0.0.0/0)

Remove

Type | Info

HTTP

Protocol | Info

TCP

Port range | Info

80

Source type | Info

Custom

















Source | Info

Q Add CIDR, prefix list or security

0.0.0.0/0 X

Description - optional | Info

e.g. SSH for admin desktop

<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/>					All states ▾		
	Name  ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone ▾
<input checked="" type="checkbox"/>	my-instance-2	i-0a34b7fe992e66ae2	 Running  	t3.micro	 2/2 checks passed	<a href="#">View alarms</a> 	eu-north-1b
<input type="checkbox"/>	my-instance-3	i-071066c1466a1fd98	 Running  	t3.micro	 2/2 checks passed	<a href="#">View alarms</a> 	eu-north-1c
<input type="checkbox"/>	my-instance-1	i-0d0cc2ea95ec49baa	 Running  	t3.micro	 2/2 checks passed	<a href="#">View alarms</a> 	eu-north-1a



- After successful creation of instances connect to instances and install nginx and create an .html file for our recognition.
- By the same process we need to do in 3 servers.
- First connect to the server 1 in git bash and type sudo -i for the root user.
- And then type yum update -y && yum install nginx -y && cd /usr/share/nginx/html in git bash.
- Remove index.html file and then create index.html file and insert data to the file as “hi this is bhargav from north 1a”.
- By the same do in server-2 and server-3.

## Output From Server-1 :

```
[root@ip-120-10-1-155 html]# yum update -y && yum install nginx -y && cd /usr/share/nginx/html
Last metadata expiration check: 1:23:20 ago on Mon Aug 5 09:25:38 2024.
Dependencies resolved.
Nothing to do.
Complete!
Last metadata expiration check: 1:23:21 ago on Mon Aug 5 09:25:38 2024.
Package nginx-1:1.24.0-1.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-120-10-1-155 html]#
[root@ip-120-10-1-155 html]# cat index.html
hi this bhargav from north-1a
[root@ip-120-10-1-155 html]# systemctl restart nginx
[root@ip-120-10-1-155 html]# curl 120.10.1.155
hi this bhargav from north-1a
[root@ip-120-10-1-155 html]# curl 120.20.0.133:80
hi this is bhargav from north-1b
[root@ip-120-10-1-155 html]# curl 120.30.1.123:80
hi this is bhargav from north-1c
[root@ip-120-10-1-155 html]#
```

## Output From Server-2 :

```
balineni bhargav@BHARGAV MINGW64 ~/OneDrive/Desktop (master)
$ ssh -i "newba.pem" ec2-user@ec2-16-171-165-84.eu-north-1.compute.amazonaws.com

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

Last login: Mon Aug 5 09:38:06 2024 from 103.160.27.100
[ec2-user@ip-120-20-0-133 ~]$ sudo su
[root@ip-120-20-0-133 ec2-user]# yum update -y && yum install nginx -y && cd /usr/share/nginx/html
Last metadata expiration check: 0:29:11 ago on Mon Aug 5 09:36:09 2024.
Dependencies resolved.
Nothing to do.
Complete!
Last metadata expiration check: 0:29:12 ago on Mon Aug 5 09:36:09 2024.
Package nginx-1:1.24.0-1.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-120-20-0-133 html]# vi index.html
[root@ip-120-20-0-133 html]# systemctl restart nginx
[root@ip-120-20-0-133 html]# curl 120.20.0.133:80
hi this is bhargav from north-1b
[root@ip-120-20-0-133 html]# curl 120.10.1.155:80
hi this bhargav from north-1a
[root@ip-120-20-0-133 html]# curl 120.30.1.123:80
hi this is bhargav from north-1c
[root@ip-120-20-0-133 html]#
```

## Output From Server-3 :

```
balineni bhargav@BHARGAV MINGW64 ~/OneDrive/Desktop (master)
$ ssh -i "newba.pem" ec2-user@ec2-13-53-172-204.eu-north-1.compute.amazonaws.com

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

Last login: Mon Aug 5 09:39:08 2024 from 103.160.27.100
[ec2-user@ip-120-30-1-123 ~]$ sudo su
[root@ip-120-30-1-123 ec2-user]# yum update -y && yum install nginx -y && cd /usr/share/nginx/html
Last metadata expiration check: 0:30:26 ago on Mon Aug 5 09:36:57 2024.
Dependencies resolved.
Nothing to do.
Complete!
Last metadata expiration check: 0:30:27 ago on Mon Aug 5 09:36:57 2024.
Package nginx-1:1.24.0-1.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-120-30-1-123 html]# vi index.html
[root@ip-120-30-1-123 html]# systemctl restart nginx
[root@ip-120-30-1-123 html]# curl 120.30.1.123:80
hi this is bhargav from north-1c
[root@ip-120-30-1-123 html]# curl 120.20.0.133:80
hi this is bhargav from north-1b
[root@ip-120-30-1-123 html]# curl 120.10.1.155:80
hi this bhargav from north-1a
[root@ip-120-30-1-123 html]#
```

## **Problems with Transit Vpc:**

- Instance Based.
- Additional EC2 Cost.
- Software Licensing Cost.
- Availability Issues.
- Bandwidth Limitations of EC2.

## **Conclusion :**

AWS Transit Gateway simplifies cloud network architectures by acting as a hub to interconnect your VPCs, VPNs, and data centers. It eliminates complex mesh topologies and provides easy scalability, centralized management, and secure network segmentation. As your cloud footprint grows, Transit Gateway is key to maintaining a simple, efficient, and secure network topology.

Balineni Bhargav

[bargavrambalineni@gmail.com](mailto:bargavrambalineni@gmail.com)

7995693436

Batch No : 127