

Cross-Account VPC Peering Through Transit Gateway in AWS

Executive Summary

This document outlines the implementation of secure, scalable connectivity between two AWS accounts using AWS Transit Gateway. The configuration enables seamless VPC-to-VPC communication across accounts.

Environment Details

- Account A (Root Account): VPC-A (10.10.0.0/16)
- Account B (Dev Account): VPC-B (10.20.0.0/16)
- Transit Gateway shared from Account A → Account B
- EC2 instances placed in private subnets of each VPC.

Transit Gateway Creation (Account A)

Steps:

1. Navigate to VPC → Transit Gateways → Create TGW
2. Accept default settings or customize as per design

The screenshot shows the AWS VPC console interface. The top navigation bar includes links for VPC, EC2, IAM, S3, Billing and Cost Management, CloudWatch, AWS Auto Scaling, Elastic Container Registry, Route 53, and Elastic Kubernetes Service. The main content area is titled "Transit gateways (1/1) Info". It displays a single entry for "my-main-tgw" with ARN "tgw-081a2b101bf6d6d61", Owner ID "180294219747", and State "Available". Below this, a detailed view for "Transit gateway: tgw-081a2b101bf6d6d61 / my-main-tgw" is shown with the following table:

Transit gateway ID	State	Amazon ASN	DNS support
tgw-081a2b101bf6d6d61	Available	64512	Enable

Below the table, there are four more columns: "Transit gateway ARN" (arn:aws:ec2:ap-south-1:180294219747:transit-gateway:tgw-081a2b101bf6d6d61), "Default association route table" (Enable), "Association route table ID" (tgw-rtb-Obf9835ee7a84c1c0), and "Auto accept shared attachments" (Disable).

Resource Access Manager (RAM) Sharing

Steps:

1. Go to Resource Access Manager (RAM)
2. Create Resource Share → Add TGW
3. Add Account B as principal
4. Accept share from Account B

The screenshot shows the AWS RAM console interface. On the left, there's a sidebar with 'Resource Access Manager' and 'Shared by me' sections. Under 'Shared by me', it lists 'Resource shares', 'Shared resources', and 'Principals'. The main area is titled 'Summary' for a resource share named 'tgw'. It shows details like Name: tgw, Owner: 180294219747, Created on: 2025/11/29, ARN: arn:aws:ram:ap-south-1:180294219747:resource-share/631423ca-2ecb-4a05-8697-60e76d55f876, and Status: Active. Below this is a 'Shared resources (1)' section with a table showing one entry: tgw-081a2b101bf6d6d61, which is an ec2:TransitGateway and is associated.

This screenshot shows the AWS RAM console from the perspective of 'Developer-Account'. The sidebar and summary details are identical to the first screenshot. In the 'Shared resources (1)' section, the entry 'tgw-081a2b101bf6d6d61' now has a status of 'Associated' with a green checkmark, indicating that the resource share has been accepted by the receiving account.

Transit Gateway Attachments

Steps:

1. Create VPC-A attachment in Account A
2. Create VPC-B attachment in Account B
3. Accept required attachments

The screenshot shows two separate AWS VPC consoles side-by-side.

Top Console (Account ID: 1802-9421-9747):

- Left sidebar: VPC > Transit gateway attachments
- Table header: Transit gateway attachments (2) Info
- Table rows:
 - VPC-B: tgw-attach-0776a8d7ed2f2cad8, tgw-081a2b101bf6d6d61, Available
 - VPC-A: tgw-attach-0bb8f2d8e40b9a2b5, tgw-081a2b101bf6d6d61, Available

Bottom Console (Account ID: 7503-1144-0127):

- Left sidebar: VPC > Transit gateway attachments
- Table header: Transit gateway attachments (1) Info
- Table rows:
 - tgw-attachment: tgw-attach-0776a8d7ed2f2cad8, tgw-081a2b101bf6d6d61, Available, VPC

VPC Route Table Configuration

Each VPC Route Table must include:

- VPC-A → 10.20.0.0/16 → TGW
- VPC-B → 10.10.0.0/16 → TGW

Transit gateway route tables (1/1)

Name	Transit gateway route table ID	Transit gateway ID	State
tgw-rtb-0bf9835ee7a84c1c0	tgw-081a2b101bf6d6d61	Available	

Transit gateway route tables: tgw-rtb-0bf9835ee7a84c1c0

CIDR	Attachment ID	Resource ID	Route t...
10.10.0.0...	tgw-attach-0bb8f2d8e40b9a2b5	vpc-0a260ef3108a30136	VPC Static
10.20.0.0...	tgw-attach-0776a8d7ed2f2cad8	vpc-0b0090ab97b53e3f9	VPC Propagated

EC2 Deployment in Both VPCs

Launch EC2 instances in respective subnets.

Ensure:

- Correct VPC selection
- Security groups allow ICMP & SSH from the opposite CIDR

Instances (1/1)

Name	Instance ID	Instance state	Instance type	Status c...
VPC-A-Server	i-0e4a5b13bf3f786fc	Running	t2.micro	2/2 c...

i-0e4a5b13bf3f786fc (VPC-A-Server)

Instance ID i-0e4a5b13bf3f786fc	Public IPv4 address 13.233.106.102 open address	Private IPv4 addresses 10.10.8.253
IPv6 address -	Instance state Running	Public DNS ec2-13-233-106-102.ap-south-

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main area displays a table of instances with one row selected. The selected instance is 'VPC-B-Server' with Instance ID 'i-0c56d7c3f4af0e8aa'. The instance is listed as 'Running' with an 't2.micro' type. It has passed 2/2 checks and has a status check link. Below the table, the details for the selected instance are shown:

- Instance summary** (Info)
- Instance ID:** i-0c56d7c3f4af0e8aa
- IPv6 address:** -
- Public IPv4 address:** 3.109.2.114 | open address
- Private IPv4 addresses:** 10.20.19.36
- Instance state:** Running
- Public DNS:** ec2-3-109-2-114.ap-south-1.compute.amazonaws.com | open address

Connectivity Verification (Ping & SSH)

Validation tests:

- Ping VPC-A → VPC-B
- Ping VPC-B → VPC-A
- SSH from laptop into both EC2s
- SSH EC2-A → EC2-B and EC2-B → EC2-A

```
ubuntu@ip-10-10-8-253:~$ ping 10.20.19.36
PING 10.20.19.36 (10.20.19.36) 56(84) bytes of data.
64 bytes from 10.20.19.36: icmp_seq=1 ttl=63 time=1.46 ms
64 bytes from 10.20.19.36: icmp_seq=2 ttl=63 time=1.73 ms
64 bytes from 10.20.19.36: icmp_seq=3 ttl=63 time=1.07 ms

```

```
ubuntu@ip-10-10-8-253:~$ ssh -i tgw-key.pem ubuntu@10.20.19.36
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sun Nov 30 02:56:02 UTC 2025

System load: 0.0          Processes:           113
Usage of /: 28.4% of 6.71GB  Users logged in: 1
Memory usage: 22%          IPv4 address for enX0: 10.20.19.36
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Sun Nov 30 02:54:31 2025 from 49.37.169.142
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-20-19-36:~$
```

```
ubuntu@ip-10-20-19-36:~$ ping 10.10.8.253
PING 10.10.8.253 (10.10.8.253) 56(84) bytes of data.
64 bytes from 10.10.8.253: icmp_seq=1 ttl=63 time=1.59 ms
64 bytes from 10.10.8.253: icmp_seq=2 ttl=63 time=1.39 ms
64 bytes from 10.10.8.253: icmp_seq=3 ttl=63 time=0.885 ms
^C
--- 10.10.8.253 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.885/1.287/1.589/0.295 ms
ubuntu@ip-10-20-19-36:~$
```

```
ubuntu@ip-10-20-19-36:~$ ssh -i kmc-info-key.pem ubuntu@10.10.8.253
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sun Nov 30 03:00:53 UTC 2025

System load: 0.0          Processes:           112
Usage of /: 26.6% of 6.71GB  Users logged in: 1
Memory usage: 21%          IPv4 address for enX0: 10.10.8.253
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Sun Nov 30 02:54:39 2025 from 49.37.169.142
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-10-8-253:~$
```

Internal EC2-to-EC2 File Transfer (SCP)

Examples:

EC2-A → EC2-B:

```
scp -i tgw-key.pem file.txt ubuntu@10.20.x.x:/home/ubuntu/
```

EC2-B → EC2-A:

```
scp -i kmc-info-key.pem file.txt ubuntu@10.10.x.x:/home/ubuntu/
```

```
ubuntu@ip-10-10-8-253:~$ echo "This is a test file" > testA.txt
ubuntu@ip-10-10-8-253:~$ scp -i tgw-key.pem testA.txt ubuntu@10.20.19.36:/home/ubuntu/
testA.txt
ubuntu@ip-10-10-8-253:~$ ls
Myfile.txt  file.txt  testA.txt  testB.txt  tgw-key.pem
ubuntu@ip-10-10-8-253:~$
```

```
ubuntu@ip-10-20-19-36:~$ echo "This is a test file" > testB.txt
ubuntu@ip-10-20-19-36:~$ scp -i kmc-info-key.pem testB.txt ubuntu@10.10.8.253:/home/ubuntu/
testB.txt
ubuntu@ip-10-20-19-36:~$ ls
Myfile.txt  file.txt  kmc-info-key.pem  testA.txt  testB.txt
ubuntu@ip-10-20-19-36:~$
```

Final Validation Checklist

- ✓ TGW Created
- ✓ Share Accepted by Account B
- ✓ Attachments in 'Available' state
- ✓ TGW Route Table configured
- ✓ VPC Routes configured
- ✓ Ping works both ways
- ✓ SSH works both ways
- ✓ SCP transfers successful

12. Conclusion

The cross-account TGW setup is fully functional and validated. Both VPCs are securely connected using a scalable hub-and-spoke architecture.