**Transit gateway**

VPC peering is used to connect two VPCs so they can communicate with each other easily. But while working with a large number of VPC connectivity it becomes complicated, as the number of VPCs increases. The network become complex.

For example, if you want to connect 5 VPCs directly, you would need to create over 10 individual peering connections. Managing and maintaining all these links can be difficult and confusing.

To overcome this disadvantage we uses the **AWS Transit gateway.**

**AWS Transit gateway:**

AWS Transit Gateway is a **fully managed service** by AWS that connects VPCs and on-premises networks through a **centralized hub**. This design reduces the need for complex peering relationships (like VPC peering) and simplifies management. Think of it as a central router that simplifies network architecture and management, especially in complex, distributed environments.

Without a Transit Gateway, if you have multiple VPCs that need to communicate with each other, you would typically set up a mesh of VPC peering connections. This becomes incredibly complex and hard to manage as the number of VPCs grows (e.g., for N VPCs, you need N\*(N-1)/2 peering connections).

Transit Gateway acts as a central hub. Each VPC (or on-premises network) connects only to the Transit Gateway, simplifying the routing table management and reducing the number of connections.

**Features:**

* **Centralized Connectivity Hub:** Acts as a central point for connecting thousands of VPCs and on-premises networks.
* **Route Tables:** Each Transit Gateway has its own route tables. You can associate VPC attachments and VPN attachments with specific route tables to control traffic flow and implement network segmentation.
* **Attachment Types:**
* **VPC Attachments:** Connects your Amazon VPCs to the Transit Gateway.
* **VPN Attachments:** Connects your on-premises networks to the Transit Gateway via IPsec VPN/VPG tunnels.
* **Direct Connect Gateway Attachments:** Connects your Direct Connect Gateways to the Transit Gateway, allowing your on-premises network to access multiple VPCs through a single Direct Connect connection.
* **Peering Attachments:** Allows you to peer two Transit Gateways in different AWS regions, enabling inter-region connectivity.
* **Traffic Segmentation:** Use separate route tables and associations to isolate traffic between different environments (e.g., production, development, shared services).
* **Multicast Support:** Supports IP multicast routing, which is useful for applications that require one-to-many communication (e.g., video streaming, financial services).
* **Network Manager:** A feature within AWS Transit Gateway that provides a unified view and management of your global network across AWS and on-premises resources. It includes network events, topology visualizations, and performance monitoring.
* **Shared Services VPC Integration:** Easily route traffic to and from a shared services VPC (e.g., for centralized firewalls, DNS, or other network appliances) that is attached to the Transit Gateway.
* **Load Balancing:** Transit Gateway can distribute traffic among multiple equal-cost paths to a destination, providing a form of load balancing.
* **Inter-Region Peering:** Connects Transit Gateways across different AWS regions, enabling global network architectures with simplified routing.
* **Flexible Routing Options:** Supports static routes, propagated routes from VPCs, and learned routes from VPNs and Direct Connect.
* **Flow Logs:** Integrates with Amazon CloudWatch Flow Logs to capture detailed information about the IP traffic going to and from network interfaces on the Transit Gateway, useful for monitoring and troubleshooting.

**Why Use Transit Gateway?**

* **Scalability**: Connect thousands of VPCs and on-premises networks.
* **Simplicity**: Replaces complex peering meshes with a hub-and-spoke model.
* **Cost-effective**: Reduces the number of VPNs and Direct Connect links needed.
* **Security**: Allows traffic filtering using Transit Gateway route tables and integration with Network Firewall.
* **High performance**: Built on the AWS backbone for low latency and high throughput.

**Key Features of Transit Gateway**

| **Feature** | **Description** |
| --- | --- |
| **Centralized routing** | TGW uses route tables to control traffic between VPCs and VPNs. |
| **Support for thousands of attachments** | Supports large-scale architectures with thousands of VPCs. |
| **Multicast support** | Enables multicast traffic across VPCs (e.g., for media streaming). |
| **Transit Gateway Peering** | Connect TGWs in different AWS regions (global network). |
| **Integration with Direct Connect** | Route on-premises traffic directly via Direct Connect Gateway. |
| **Bandwidth control and traffic inspection** | Integrate with firewalls and control throughput per attachment. |
| **High availability** | Built-in redundancy and failover capabilities. |
| **VPN and AWS Site-to-Site VPN support** | Easily connect on-prem networks using IPsec tunnels. |

**Note1:** VPC peering is leads to mesh model, whereas Transit Gateway form hub-and-spoke model.

**Note2:** Transit Gateway just acts as a Load balancer in the point of establishing Networking connections.

**Note3:** Transit Gateway uses **route tables** to control how traffic is routed between connected networks (VPCs, VPNs, Direct Connect, etc.).

Block Diagram:

Region-02

Region-01

VPC-02

VPC-03

VPC-01

VPC-03

VPC-02

VPC-01

TGW

TGW

Fig: Transit Gateway connectivity between two different regions.

Let’s work with the transit gateway practically.

Process: Now let’s create a two VPC, in each VPC create a single instance in it. Do this in two different regions. And establish the connection/communication between these all VPC using Transit gateway.