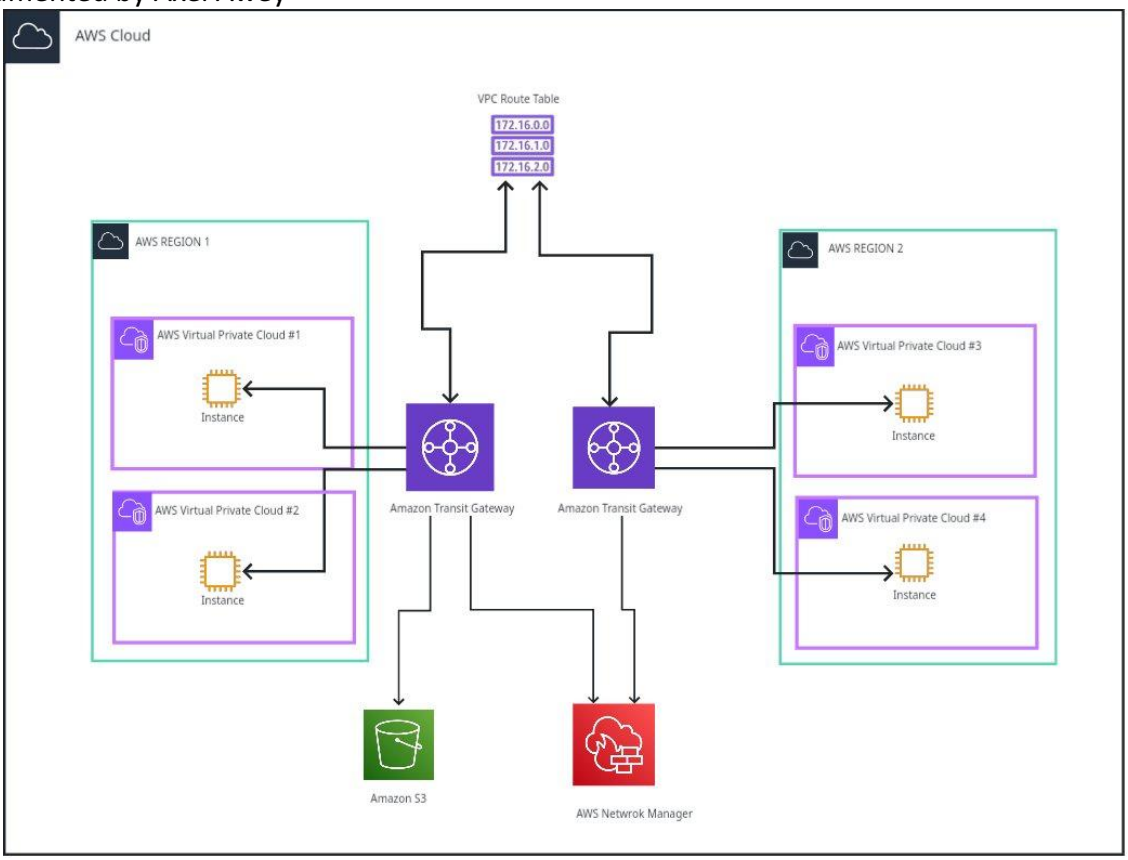
AWS Transit Gateway Setup & Analysis Documentation

Documented by Axel Awey



Overview

This is documentation on the demo of a fully functional AWS Transit Gateway Setup, as well as Analysis with detailed explanations of the traffic flow across the VPCs located in different and multiple Availability Zones and Regions.

The AWS Services utilized in this project are;

- AWS Virtual Private Cloud (VPC)
- AWS Elastic Cloud Compute (EC2)
- AWS Transit Gateway
- AWS CloudWatch
- AWS Network Manager

With an AWS Transit Gateway Setup, this gives customers the opportunity to ;

- Streamline the customer's architecture - meaning they are able to manage the growth of the setup and more based of the complexity of the setup.
- Completely improves security of the architecture
- Enables lift and shift for customer's for them to be able to migrate their resources from on-premises to the cloud or to primarily move whatever resources they want to share amongst each other.

With the appropriate routing, resources can be shared amongst users and customers across different availability zones and AWS Regions as well.

A Transit Gateway gives users and customers the opportunity to attach their VPC and VPN connections and also to route traffic between them across different Availability Zones, it also works across multiple AWS Accounts. Users can even go to the extent of utilizing AWS RAM to share and distribute resources in your transit gateway with the other accounts. The Transit Gateway acts as a interconnection hub where users can not only connect their VPCs but also their devices and on-premises sites.

Transit Gateways support Dynamic and Static Routing; Static routing is a network configuration method where routes are manually entered in to a router's routing table and Dynamic routing where routers automatically update and disclose the routing information in order to determine the right path for data transmission. In a Transit Gateway setup, each VPC in their respective Availability Zones must be associated with a single route table and that route table is used to decide the next distribution for the traffic coming from the resource attachment. The route table allows for both IPv4 and IPv6 CIDR blocks with VPCs being the targets.

This is optional for users and customers that prefer to segment or to isolate their network. Additional route tables can be added within the transit gateway. To isolate their network, a sample routing architecture would something similar to this where one VPC is routed and associated with a particular route table and the other VPC associated with a completely different route table, This creates a completely isolated network inside of the Transit Gateway.

For successful outcome of this setup it would also require a Transit Gateway Attachment for each of the VPCs in each region as well as a Transit Gateway Peering Connection.

In my setup, I had 2 VPCs launched in 2 AWS Regions along with a Transit Gateway in each of them. Each VPC was configured with public and private subnets across multiple Availability Zones, and EC2 Instances were deployed for connetivity testing. The Transit Gateways were attached to their respective VPCs, and routing was configured to allow inter-VPC communication aross regions.

To capture and analyze network traffic, I enabled VPC Flow Logs for all VPCs and Transit Gateway Flow Logs for both Transit Gateways. Logs were configured to publish to Amazon CloudWatch Logs, enabling real-time filtering by source/destination, traffic type, and status. This gave clear insights into accepted/rejected traffic and routing behavior during testing.

I configured AWS Network Manager to visualize and monitor the multi-VPC, multi-Region architecture. A global network was created, and both Transit Gateways were registered under it. This provided centralized visibility of the network topology, route propagation, and link health, as well as an aggregated view of CloudWatch metrics and flow log data for easier troubleshooting.

In this project, I overcame Transit Gateway routing challenges, ensuring all necessary route propagations and associations were configured. Combined with flow logs and Network Manager, I was able to validate end-to-end connectivity and optimize network performance across regions.