TransitVPC TNSR Deployment Guide

Overview

AWS Transit VPC enables multiple Virtual Private Clouds (VPCs) - either geographically disparate and/or running in separate AWS accounts - to connect to a common VPC that serves as a global network transit center. Transit VPC simplifies cloud network management and minimizes the number of connections that must be configured and managed. Further, owing to its virtual implementation, no physical network equipment - or physical presence in a colocation transit hub is required.

This guide will help you to deploy Transit VPC in Amazon AWS. Traffic between VPCs is controlled and secured by an instance of the TNSR high speed router and firewall.

The TNSR instance is provisioned with 2 interfaces. The first interface is attached to the instance's kernel networking stack and is used to manage the instance via SSH or RESTCONF. The second interface is managed by the VPP packet processing daemon and is used to terminate IPsec tunnels which will be used to pass traffic securely between the VPCs and remote networks that will participate in the Transit VPC. The interfaces should each have an Elastic IP address assigned.

For deployment TransitVPC uses the CloudFormation template which contains basic parameters for running the TNSR instance. The CloudFormation template also creates roles, policies, and LambdaFunctions in AWS as well as setting up a schedule for the Lambda script.

Components used:

- CloudFormation template - **TransitVPC.json**. This template deploys a TransitVPC instance with two interfaces, configures and starts the Lambda function, and creates access roles
- TNSR instance. - tnsr-image-ami used as TransitInstance
- Lambda Function - **lambda/transit-vpc-poller.py.** This Function searches for specific TAGs on VGW VPC every minute.
- Python script - **lambda/transitvpc_ipsec.py.** This script establishes VPN connections with SpokeVPCs
- Systemd service - **service/transitvpc.service** with command to run python script
- Systemd timer **service/transitvpc.timer** with run service by scheduler

Preparatory actions:

- Create a new S3 bucket with basic security parameters and upload **lambda/transit-vpc-poller.zip to the bucket.** This archive contains the python script **lambda/transitvpc_ipsec.py** for AWS Lambda. You can use any Bucket name. This how-to will use the CloudFormation parameter- **S3BucketScript.**

Deployment steps

1. Open Amazon AWS Console and select "cloudformation" service.
2. Click choose a file, and choose CloudFormation template from your PC, then NEXT
3. Complete the following cloudformation required parameters:
    - StackName = Name in stacklist for managed stack
    - EnvName = Tag ENV: (for comfort)
    - InstanceAMI = existing tnsr AMI
    - BGP_ARN = autonomous system number
    - ALLOWSSH = your IP/SUBNET for ssh access. format 110.0.101.10/32
    - KeyName = choose ssh key for access into InstanceAMI
    - S3BucketScript = bucket where transit-vpc-poller.zip is stored
    - S3BucketConf = bucket where VPN configs are stored
    - S3Prefix = directory with config file
    - SpokeTag = Tag Key searching by Lambda
    - SpokeTagValue = Tag Value searching by Lambda
4. On the Options page click NEXT
5. On the Review page scroll down and choose check-box “I acknowledge that AWS CloudFormation might create IAM resources.” and click CREATE
6. Deployment will be completed in 2-3 minutes

Connect SpokeVPC to TransitVPC

1. Add **SpokeTag** : **SpokeTagValue** to VGW Spoke VPC
2. LambdaFunc will create VPN connection for TransitVPC in1-2 minutes and put in S3 Bucket file with VPN parameters with additional xml element status=create
3. TransitInstance will read configuration file and establish a VPN connection with each SpokeVPC having configuration file with status “create”
4. Add static route in VPNConnection if (needed)
5. Add routes in RouteTablesVPC via exising VGW

Remove SpokeVPC to TransitVPC

1. Remove **SpokeTag** or change its value from SpokeTagValue to something else in VGW SpokeVPC
2. LambdaFunc will update status in the VPN configuration file (created earlier) from “create” to “delete”
3. TransitInstance will find the updated file and delete the VPNp-connection with SpokeVPC,
   TransitInstance will also remove routes to SpokeVPC and delete conf file from S3

Configure TransitInstance:

Configure vpp interface. Interface should be configured such that the IPSec tunnel can establish a connection with SpokeVPC.
Use clixon_cli or su – tnsr

configure
interface VirtualFunctionEthernet0/6/0
ip address 10.10.0.101/24
enable
exit
route ipv4 table ipv4-VRF:0
route 0.0.0.0/0
next-hop 1 via 10.10.0.1 VirtualFunctionEthernet0/6/0
exit
exit
exit

Locations:

1. Put script **transitvpc_ipsec.py** in /opt/
2. Set executable flag on /opt/transitvpc_ipsec.py

   chmod +x /opt/transitvpc_ipsec.py

3. Put service **transitvpc.service** in /usr/lib/systemd/system/ and run following command to enable service

   systemctl enable transitvpc.service

4. Put timer **transitvpc.timer** in /usr/lib/systemd/system/ and run following command to enable timer

   systemctl enable transitvpc.timer
   systemctl start transitvpc.timer