AWS TRANSIT VPC WITH FORTIGATE NEXT-GENERATION FIREWALL
This Fortinet Transit VPC Deployment Guide covers architectural design details and configuration steps for deploying a Transit VPC on Amazon Web Services (AWS). It includes links to AWS CloudFormation templates that launch, configure, and deploy the needed AWS services to deploy the Transit VPC solution on AWS.

The guide is intended for IT infrastructure architects, administrators, and DevOps professionals who have practical experience architecting for AWS.

AWS TRANSIT VPC CONCEPT

Transit VPC is a concept of having a Virtual Private Cloud (VPC) network with virtualized firewalls in AWS. The Transit VPC acts as the hub to all the traffic passing between the Spoke VPCs and the on-premises data center. For example, three branch offices (“spokes”) connect to a central office (the “hub”) over VPN links. The on-premises customer data center is a “spoke” from the perspective of the firewall inside the Transit VPC, with similar security and routing policies applied to it.

Amazon Virtual Private Cloud (Amazon VPC) provides customers with the ability to create as many virtual networks as they need, as well as different options for connecting those networks to each other and to non-AWS infrastructure. There are two common strategies for connecting multiple, geographically dispersed VPCs and remote networks to:

- Implement a hub-and-spoke network topology that routes all traffic through a network Transit center (a Transit VPC)
- Create a meshed network that uses individual connections between all networks.

Both approaches can create an efficient and available Transit network.

The Transit VPC in AWS consists of a VPC network that can be split into two or more subnets. Each of these subnets is associated with its own route tables and security groups. The Transit VPC is defined for an AWS Region, with the VPC subnets being equally divided among two Availability Zones per Region. This helps provide redundancy and high availability for the purposes of maintaining that high availability.

TRANSIT VPC CLOUDFORMATION TEMPLATE

The Transit VPC is orchestrated and brought online using custom CloudFormation templates. The parameters for creating the components of Transit VPC are defined in the CloudFormation template.

FORTINET TRANSIT VPC DEPLOYMENT IN AWS

As stated, one of the common strategies for connecting multiple, geographically dispersed VPCs and remote networks is to create a Transit VPC that serves as a global network Transit center.

A Transit VPC simplifies network management and minimizes the number of connections required to connect multiple VPCs and remote networks.
This design can save time and effort and reduce costs, as it is implemented virtually without the traditional expense of establishing a physical presence in a co-location Transit hub or deploying physical network gear.

The CloudFormation template configures and runs FortiGate in AWS. The solution below is used to display FortiGate’s features in AWS:
- FortiGate VPN appliance in a typical hub-and-spoke network topology to connect multiple VPCs across Regions and Accounts
- Automate VPN connection configuration and setup as new spoke VPCs join the topology
- Support both BYOL and pay-as-you-go licensing options

**HIGH-LEVEL NETWORK ARCHITECTURE**

In the diagram below, each FortiGate will be connected to the entire Spoke VPCs independently and in a redundant manner. Both FortiGates operate in Active-Active mode.
The core components for this solution include:

- AWS VPC, VGW
- Fortinet NGFW Appliances—FortiGate (BYOL/pay as you go)
- AWS S3
- AWS EC2
- AWS Lambda
- AWS CloudWatch
- AWS IAM Roles
- AWS KMS

**HIGH-LEVEL SOLUTION DESIGN SUMMARY**

1. The solution deploys a Transit VPC
   - In a user-provided AWS region.
   - In a user-provided CIDR.
2. It deploys one or more spoke VPCs.
   - Spoke VPCs can be spread across multiple regions
   - Spoke VPCs can be in one or more AWS Accounts and are connected to the Transit network.
   - Spoke VPCs do not overlap with each other or with the Transit VPC.
3. This highly available design deploys two FortiGate VPN appliances into separate Availability Zones of a dedicated Transit VPC.
   - Each FortiGate instance has an associated Amazon CloudWatch alarm that enables automatic recovery of the instance if the EC2 hardware fails.
4. Spoke VPCs are connected to the Transit network through dynamically routed VPN connections between their Virtual Private Gateways (VGWs) and the FortiGate instances.
5. The user has an option to select FortiGate BYOL or pay as you go. If the user opts for the BYOL option, the associated license file should be uploaded as an accessible URL endpoint (AWS S3 or others) and provided as an input to the template.
6. FortiManager optionally can be deployed in a separate subnet in the Transit VPC. However, the integration to Fortinet Appliances has to be performed manually.

The automated process for adding a new spoke VPC, as part of this solution, is as follows:

7. Every five minutes, an Amazon CloudWatch event invokes the VGW Poller Lambda function, which iterates through each AWS Region of one or more customer accounts, searching for appropriately tagged Spoke VGWs (default tag key transitvpc:spoke, default tag value true) that do not have existing Transit VPC VPN connections.
8. When the VGW Poller identifies an applicable Spoke VGW, it creates the corresponding customer gateways (if required) and VPN connections to each FortiGate Appliance, and then saves this connection information to an Amazon S3 bucket using S3 SSE-KMS. All data in the S3 bucket is encrypted using a solution-specific AWS KMS managed customer master key (CMK).
9. The S3 Put event invokes the VPN Configurator Lambda function, which parses the VPN connection information and generates the necessary config files to create new VPN connections.
10. The VPN Config (Lambda function) pushes the configuration to the VPN Appliance instances using SSH.
11. As soon as the VPN configuration is applied onto the FortiGate instances, the VPN tunnels come up and Border Gateway Protocol (BGP) neighbor relationships are established to the Spoke VPCs.

**GETTING STARTED**

1. Open AWS console with your credentials and select the deployment location.
2. Click on Services and navigate to CloudFormation under Management Tools.
3. Click on **Create Stack**.

4. On **Select Template** page, choose to upload a template to Amazon S3 and click on **Next**.
5. A **Specify Details** page opens. Enter the **Stack name**, and in the **Parameters** section, enter **AutomateUserPwd** and give the **FortiGateKeyName** (the key pair should be present in the AWS account).

6. Enter the **Date of the deployment being created** in the **TagCreated** section on the same page and click on **Next**, which leads to the **Options** page.
7. On this page, enter the **Tags** giving values for the **Key** section and **Value** section and click on **Next**.

8. A **Review** page opens, which gives the overview of the details entered.
9. Check the box **I acknowledge that AWS CloudFormation might create IAM resources** and click on **Create**.

10. This leads to the **Create Stack** page, which shows the progress of the stack created. Click on the **Outputs** sections below to view the events.
11. The following are the resources created after the CloudFormation deployment:

- VPC
- 2 FortiGate EC2 Instances
- S3 Bucket
- Lambda Function: This solution uses two AWS Lambda functions, the VGW Poller and the Worker Config.
- IAM Roles and Policies

12. We can access the FortiGate console by copying the Value of the Key from Outputs of CloudFormation and browsing it.

13. Give the admin username and password relevant to the console.
14. Open another AWS account.

15. Click on **Services** and navigate to **CloudFormation** under **Management Tools**.

16. Click on **Create Stack**.

17. On the **Select Template** page, choose **Upload a template to Amazon S3** and click on **Next**.
18. A **Specify Details** page opens. Enter the **Stack name**. In the **Parameters** section, enter **S3BucketName** of the Transit VPC S3 bucket that is used to read the Transit endpoints and store configuration files. Enter the **Date of the deployment being created** in TagCreated and click on **Next**.

19. This creates a Lambda function.

For more information or support on FortiGate Transit VPC design, please contact [aws@fortinet.com](mailto:aws@fortinet.com). For CloudFormation download and feedback, please visit [Github Fortinet Solutions](https://github.com/fortinetsolutions).

**REFERENCE:**