Implementation Guide:

Aviatrix Multi-Account Transit Network
# Table of Contents

- Foreword ............................................................................................................................................................. 3
- Solution overview and features .......................................................................................................................... 4
- Architecture diagram........................................................................................................................................... 4
- Pre-requisites ...................................................................................................................................................... 5
- Best Practices .................................................................................................................................................... 12
- Solution Estimated Pricing................................................................................................................................ 12
- FAQs................................................................................................................................................................... 12
- Additional resources.......................................................................................................................................... 12
- Partner contact information.............................................................................................................................. 12
Foreword

Aviatrix Transit Networking is an advanced networking solution to connect multiple Amazon Virtual Private Cloud (VPC) across regions in the AWS cloud. Implementing this solution, allows setting up a multi-region enterprise grade network in AWS and leveraging the AWS Control Tower account creation workflow to enable automated provisioning of new VPCs and their attachment into the enterprise network infrastructure.

The purpose of this AWS Implementation Guide is to enable AWS Marketplace customers to seamlessly activate, deploy and configure the AWS Transit Gateway in AWS Control Tower environment while leveraging the resources pre-configured by AWS Control Tower as part of the initialization.
Solution overview and features

AWS Control Tower allows setting up network baselines such as creating VPCs in multiple regions (no IP address management), configuring CIDR block ranges, and more. However, there are other networking services like AWS Transit Gateway (TGW), or AWS Direct Connect (DX) that need to be configured explicitly on the AWS Control Tower environment. As of June 2020, configuring the Transit Gateway as part of the networking baselines in Account Factory is not natively supported. This joint solution leverages the Aviatrix platform to provide advanced networking functionality and serve as a network factory for newly provisioned/enrolled AWS accounts under AWS Control Tower.

New accounts need to be provisioned from AWS Control Tower Account Factory. This solution also covers existing account onboarding as a child account in AWS Control Tower using the Enroll existing account functionality.

The Aviatrix cloud network platform delivers:
- Advanced Networking & Security while maintaining the simplicity and automation of cloud.
- Enables enterprise IT to easily deploy a high-availability, Multi-Cloud Network Data Plane with high performance encryption, multi-cloud network segmentation and unparalleled operational telemetry
- Brings day-two operational troubleshooting and visibility to help pinpoint traffic anomalies and suspicious behavior, resolve connectivity problems faster, and to share network health metrics and dynamic network resource maps with staff and management.

Architecture diagram

Figure 1 Aviatrix Multi-account Transit Network Architecture Diagram
This integrated solution leverages the Aviatrix platform to add “Network Factory” capabilities to AWS Control Tower by provisioning new VPC and attaching it to the shared AWS network infrastructure each time a new account is created/onboarded. It ensures VPC CIDRs are unique in the customer environment.

The solution relies on AWSControlTowerExecution cross account AWS Identity and Access Management (IAM) role created by AWS Control Tower in every child account. This cross account IAM role is used to create additional IAM roles that Aviatrix Controller uses to provision and orchestrate the network infrastructure. The deployment CF template creates a new VPC and sets up a Python3 runtime AWS Lambda that is triggered by the CreateManagedAccount Control Tower lifecycle event to automate the per account VPC creation. It utilizes AWS Resource Access Manager to share the AWS Transit Gateway with the account and create a VPC attachment into the enterprise network.

Pre-requisites

1) Use one of your existing AWS accounts or create Network Infra account in the Infrastructure OU
2) Subscribe to Aviatrix Secure Networking Platform Metered - 24x7 Support on AWS Marketplace.
   https://aws.amazon.com/marketplace/pp/B086T2RVTF

   Click on the Continue to Subscribe button.

3) Launch Aviatrix Controller using this template and follow the instructions at the below URL

   The installation does the following:
   a) Creates two IAM roles i.e aviatix-role-ec2 and aviatix-role-app. The associated policies can be referenced at
      https://s3-us-west-2.amazonaws.com/aviatrix-download/iam_assume_role_policy.txt
      https://s3-us-west-2.amazonaws.com/aviatrix-download/IAM_access_policy_for_CloudN.txt
   b) Launches a new Amazon EC2 instance (t3.large) for the Aviatrix Controller
   c) Assigns new EIP to the instance
   d) Creates a new Security Group called AviatrixSecurityGroup

4) The steps in the startup guide also ensure that the AWS account in which Aviatrix Controller is launched is onboarded to Aviatrix platform. Once the controller is up and running, access the UI via https://<EIP>.

   Login to controller and do the following:
   a. Follow the instructions at the below URL to onboard all the existing AWS Control Tower accounts such as master account, log_archive, audit, network accounts and more.
      https://docs.aviatrix.com/HowTos/aviatrix_account.html#setup-additional-access-account-for-aws-cloud
   b. If using the AWS Transit Gateway to connect VPCs, create TGW in AWS Control Tower home region
c. **Alternatively** if using the Aviatrix Data Plane to connect VPCs, from the Aviatrix Controller UI, first create a Transit VPC and then launch a pair of Aviatrix Transit Gateways in AWS Control Tower home region as shown below.
d. Create HA gateway for redundancy

5) Using the AWS CLI to securely store Aviatrix Controller private IP address and credentials in AWS Systems Manager -> Parameter Store

aws ssm put-parameter --type "SecureString" --name "/aviatrix/controller/username" --value "admin"
aws ssm put-parameter --type "SecureString" --name "/aviatrix/controller/password" --value "XXXXX"
aws ssm put-parameter --type "SecureString" --name "/aviatrix/controller/ip_address" --value "<private IP>"

Alternatively using the AWS console
If you are new to AWS, see Getting Started with AWS: https://aws.amazon.com/getting-started/.


To get started with AWS Control Tower, check out the https://docs.aws.amazon.com/controltower/latest/userguide/getting-started-with-control-tower.html

Deployment and Configuration Steps

**Step 1.1:** To get started, log in to AWS Control Tower master account. Deploy the solution AWS CloudFormation template using the following S3 URL

https://aviatrix-cloudformation-templates.s3-us-west-2.amazonaws.com/vpc_attach.template

**NOTE 1:** The lambda supports Aviatrix Controller version 5.3 and above.

**NOTE 2:** If upgrading Aviatrix Controller to 6.1 and above, the lambda package needs to be updated. Either the Cloudformation template can be deleted and recreated (this will automatically pull the latest lambda package) or the lambda zip can be updated using the AWS console.

Input parameters for the template:

- a) CIDR for VPC into which the Lambda will be launched
- b) Address space allocation for two private subnets in this VPC
- c) AWS account number Aviatrix Controller resides in
- d) Name of AWS or Aviatrix Transit Gateway to which newly created VPCs should attach
- e) If using Aviatrix Transit GW, set avtxtransit to true (Default is false)
- f) Address Range in which VPC CIDRs will be assigned (Default: 10.0.0.0/8)
- g) Generated VPC CIDR mask length (Default: 24)

The template creates a Lambda triggered by **CreateManagedAccount** Cloudwatch event from AWS Control Tower. When triggered the Lambda does the following in the newly created account:

- a) Assmes **AWSControlTowerExecution** role to create aviatrix-role-ec2 and aviatrrix-role-app roles
- b) Onboards the newly created AWS account onto the Aviatrix Controller
- c) Creates a VPC with first available /24 CIDR ensuring it is unique among all the VPCs across all AWS accounts already onboarded on the controller
- d) Attaches the VPC to the AWS or Aviatrix Transit Gateway

**Step 1.2:** From the AWS Console, verify that the CloudFormation stack was successfully deployed
**Step 1.3:** To ensure the Lambda is able to communicate with the Aviatrix Controller privately, attach Aviatrix Controller VPC and the AviatrixControlTowerLambdaVPC to the Transit Gateway in the Default_Domain.
Step 1.4: From the Aviatrix Controller, confirm that both VPCs are attached to the TGW in the Default_Domain

Testing the solution

Step 1.1: Using the Lambda test events to simulate the workflow, below abbreviated CreateManagedAccount event can be used to test the setup. Update the three parameters highlighted in red to appropriate values for your environment.

```
{
    "detail-type": "AWS Service Event via CloudTrail",
    "source": "aws.controltower",
    "region": "us-east-1",
    "detail": {
        "userIdentity": {
            "accountId": "*************",
            "invokedBy": "AWS Internal"
        },
        "eventName": "CreateManagedAccount",
        "awsRegion": "us-east-1",
        "serviceEventDetails": {
            "createManagedAccountStatus": {
                "account": {
                    "accountName": "YYYYYYY",
                    "accountId": "XXXXXXXXXXXX"
                }
            }
        }
    }
}
```

- **accountId**: Account Id of a managed AWS account within AWS Control Tower
- **accountName**: Name of the AWS Account
- **awsRegion**: region in which you have the AWS Control Tower setup.

This creates a VPC in the account specified in the event and a corresponding VPC attachment to the AWS Transit Gateway.

Step 1.2: Use the following workflow to delete the resources created by the test

1. Detach the VPC created above from the shared TGW
2. From “Useful Tools” menu, click on the three vertical dots and delete the VPC created.

3. Delete the account that was onboarded by the lambda

Using the solution
Once deployment and testing is successful you can use this automation as part of your AWS Control Tower setup. Anytime you use the Account factory to create a new account or enroll an existing account, the solution will

- Create a VPC in the new account using a CIDR block which is non-overlapping with the CIDR blocks of other VPCs setup by this solution
- Create a VPC attachment from this VPC to the AWS Transit Gateway

This can be verified by using the AWS Console as well as the Aviatrix Controller UI.
Best Practices

- Securely store the Aviatrix Controller credentials in SSM parameter store
- Before launching Aviatrix Controller, ensure elastic IP limits have not been reached

Solution Estimated Pricing

- Number of VPC-to-VPC IPSec Tunnel Connections within AWS – $0.19 per hour
- Estimated EC2 Infrastructure Cost - $0.0832 per hour (based on t3.large)

FAQs

https://docs.aviatrix.com/HowTos/tgw_faq.html

Additional resources

https://docs.aviatrix.com/index.html

Partner contact information

info@aviatrix.com