Operations, Troubleshooting and Visibility

ACE Solutions Architecture Team

Operational Challenges in Public Cloud

Evidential Data

When working with Cloud Providers, often customer is challenged to prove providers fault/issues

Unfamiliar Toolset

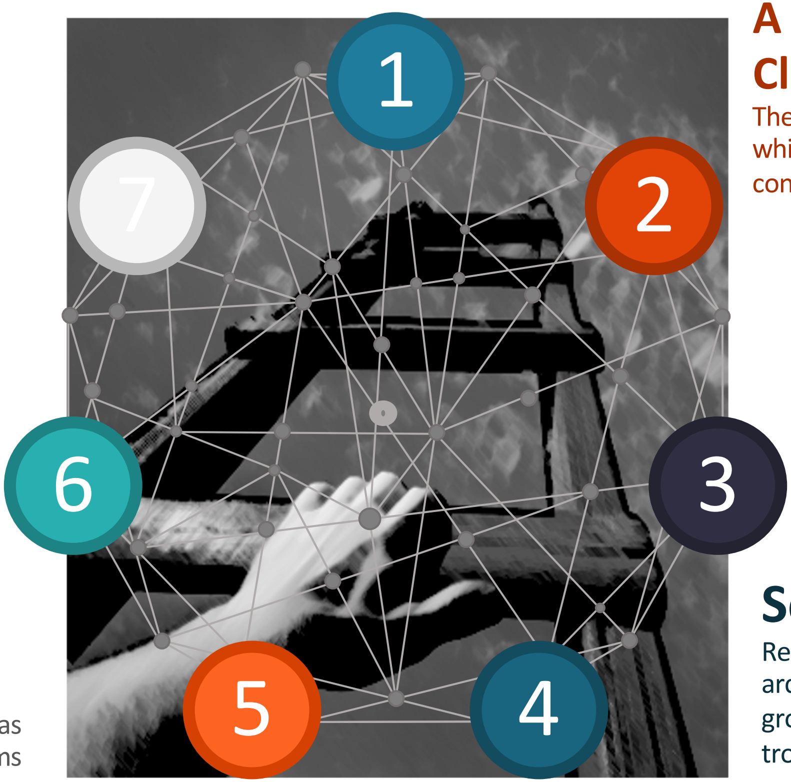
Native cloud lacks familiar tools like ping, packet capture, traceroute

Blackbox – No visibility

Native cloud constructs want you to trust all is well always. No visibility into logs, current state, routing tables, etc.

Infrastructure as Code

Solves agility problem, creates support issues as tier-1 is not able to troubleshoot code problems



A Flat World in Public Cloud

There is a lack of hierarchy in the cloud which means its hard to insert security, control and visibility

Tier-3 becomes Tier-1

Frontline support teams don't have the skill and tools in public cloud requiring senior network engineers to assist with most support issues

Scaling Out

Real problems are experienced when architecture scales out as it very quickly grows to be complex and very hard to troubleshoot

aviatrix
ACE
Aviatrix Certified Engineer

Infrastructure as Code

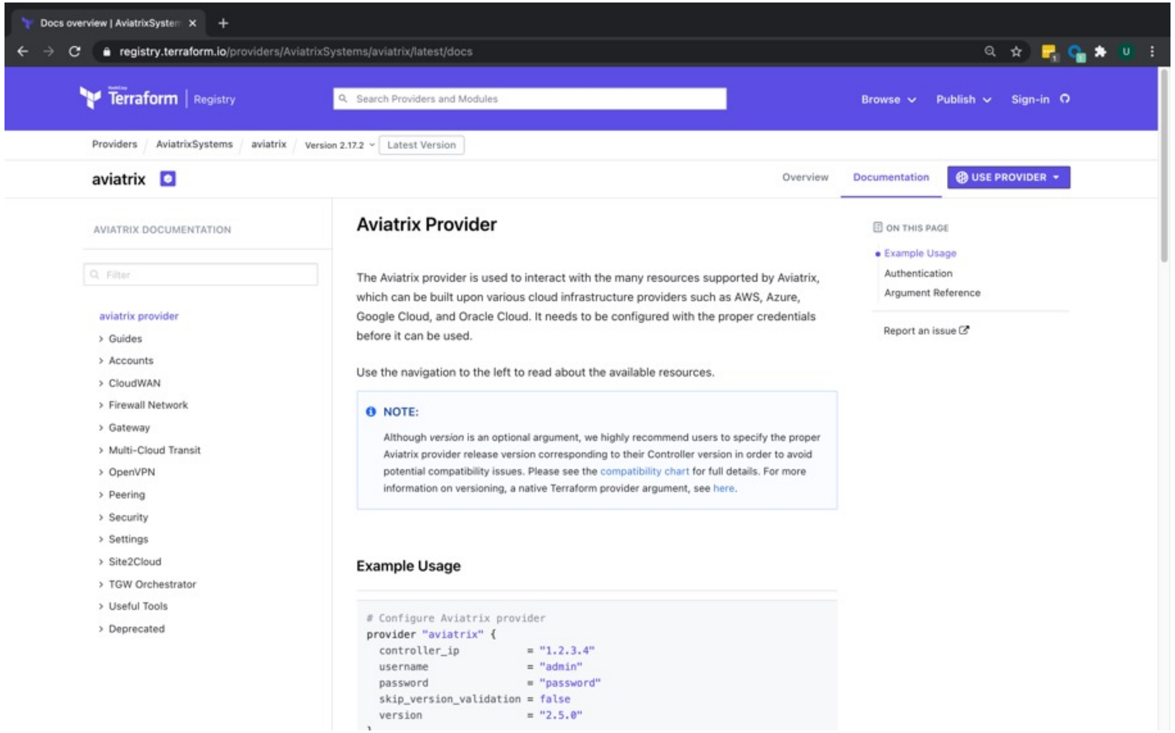
aviatrix

What it is

- Use Infrastructure as Code to provision and manage any cloud, infrastructure, or service
- Write declarative configuration files – define desired state
- Plan and predict changes
- Create reproducible infrastructure – if resource already exists, it won’t recreate it
- Maintains knowledge of resources in a database called **State**
    - State maps config to real world

Aviatrix Terraform Provider

- Multi-lingual entity responsible for API interactions with CSPs
- Exposes resources in those CSPs for any account/subscription that has been onboarded
- Feature parity with Controller code



Aviatrix Provider

The Aviatrix provider is used to interact with the many resources supported by Aviatrix, which can be built upon various cloud infrastructure providers such as AWS, Azure, Google Cloud, and Oracle Cloud. It needs to be configured with the proper credentials before it can be used.

Use the navigation to the left to read about the available resources.

NOTE:

Although *version* is an optional argument, we highly recommend users to specify the proper Aviatrix provider release version corresponding to their Controller version in order to avoid potential compatibility issues. Please see the compatibility chart for full details. For more information on versioning, a native Terraform provider argument, see here.

Example Usage

```
# Configure Aviatrix provider
provider "aviatrix" {
  controller_ip           = "1.2.3.4"
  username                = "admin"
  password                = "password"
  skip_version_validation = false
  version                 = "2.5.0"
```

Aviatrix Terraform Resources – Examples

- # Create an Aviatrix AWS Gateway

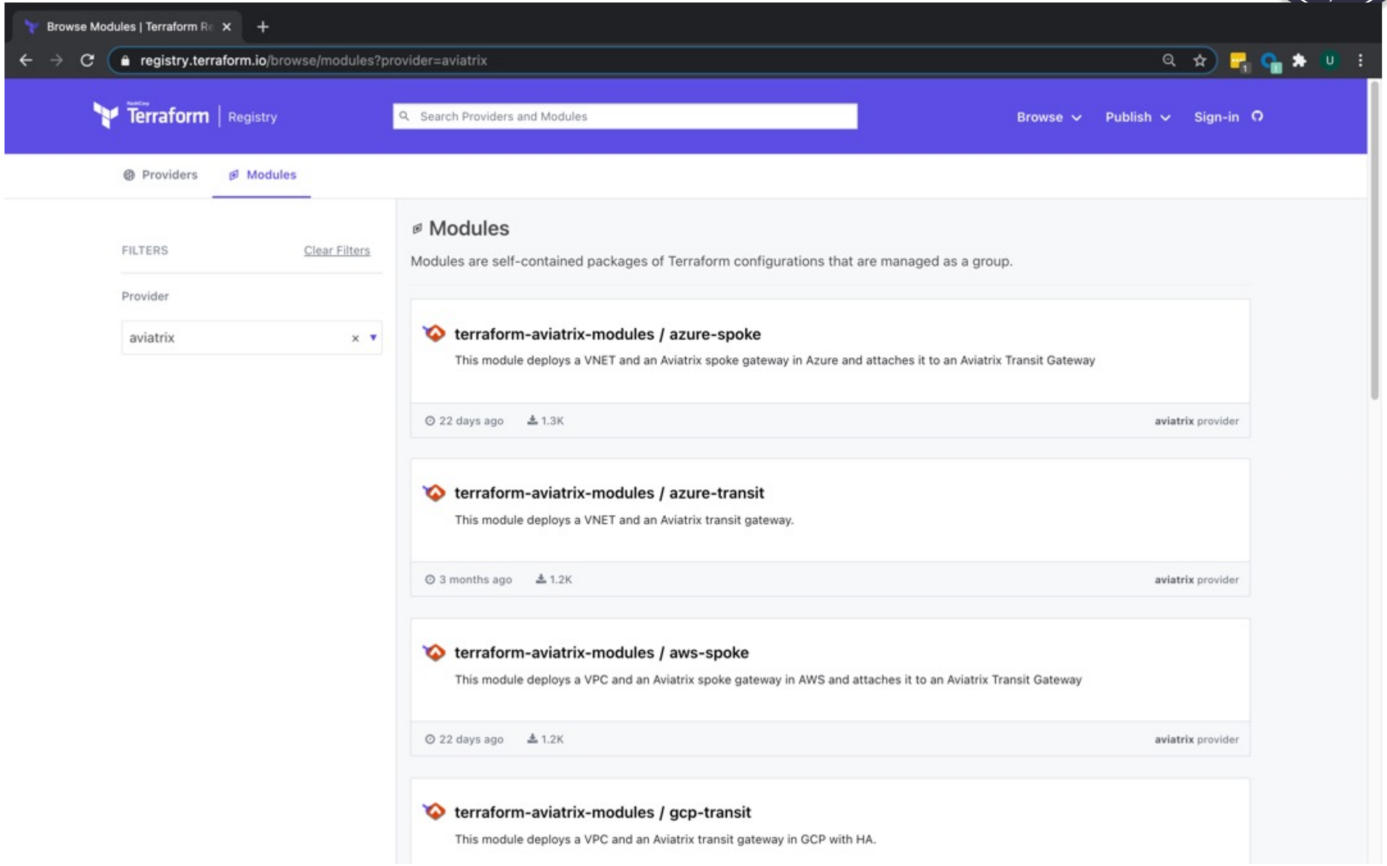
```
resource "aviatrix_gateway"
"test_gateway_aws" {

   cloud_type   = 1

   account_name = "devops-aws"

  gw_name      = "avtx-gw-1"

  vpc_id       = "vpc-abcdef"

  vpc_reg      = "us-west-1"

  gw_size      = "t2.micro"

   subnet       = "10.0.0.0/24"

}
```

- # Create an Aviatrix Azure Gateway

```
resource "aviatrix_gateway"
"test_gateway_azure" {

  cloud_type    = 8

  account_name = "devops-azure"

  gw_name      = "avtx-gw-azure"

  vpc_id       = "gateway:test-gw-123"

  vpc_reg      = "West US"

  gw_size      = "Standard_D2"

  subnet       = "10.13.0.0/24"

}
```

Aviatrix Terraform Modules

- ***"Repeatable++"***
- Similar to the concepts of libraries, packages, or modules found in most programming languages
- Provide many of the same benefits
- ~10X reduction in lines of code
- Can be found on Terraform Registry



Browse Modules | Terraform Re
registry.terraform.io/browse/modules?provider=aviatrix
Terraform | Registry
Search Providers and Modules
Browse
Publish
Sign-in
Providers
Modules
FILTERS
Clear Filters
Provider
aviatrix

Modules

Modules are self-contained packages of Terraform configurations that are managed as a group.

terraform-aviatrix-modules / azure-spoke
This module deploys a VNET and an Aviatrix spoke gateway in Azure and attaches it to an Aviatrix Transit Gateway
22 days ago · 1.3K · **aviatrix** provider

terraform-aviatrix-modules / azure-transit
This module deploys a VNET and an Aviatrix transit gateway.
3 months ago · 1.2K · **aviatrix** provider

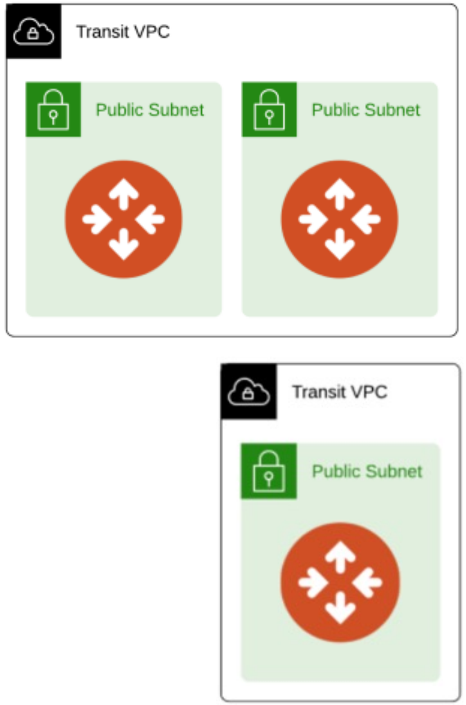
terraform-aviatrix-modules / aws-spoke
This module deploys a VPC and an Aviatrix spoke gateway in AWS and attaches it to an Aviatrix Transit Gateway
22 days ago · 1.2K · **aviatrix** provider

terraform-aviatrix-modules / gcp-transit
This module deploys a VPC and an Aviatrix transit gateway in GCP with HA.

Aviatrix Terraform Module – Example

- # Create a VPC and a set of Aviatrix transit gateways.

```
module "transit_aws_1" {

  source  = "terraform-aviatrix-modules/mc-transit/aviatrix"

  version = ”1.1.2”

  cloud   = “aws”

  cidr    = "10.1.0.0/20"

  region  = "eu-west-1"

  account = "AWS-account"

}

ha_gw set to true by default
```



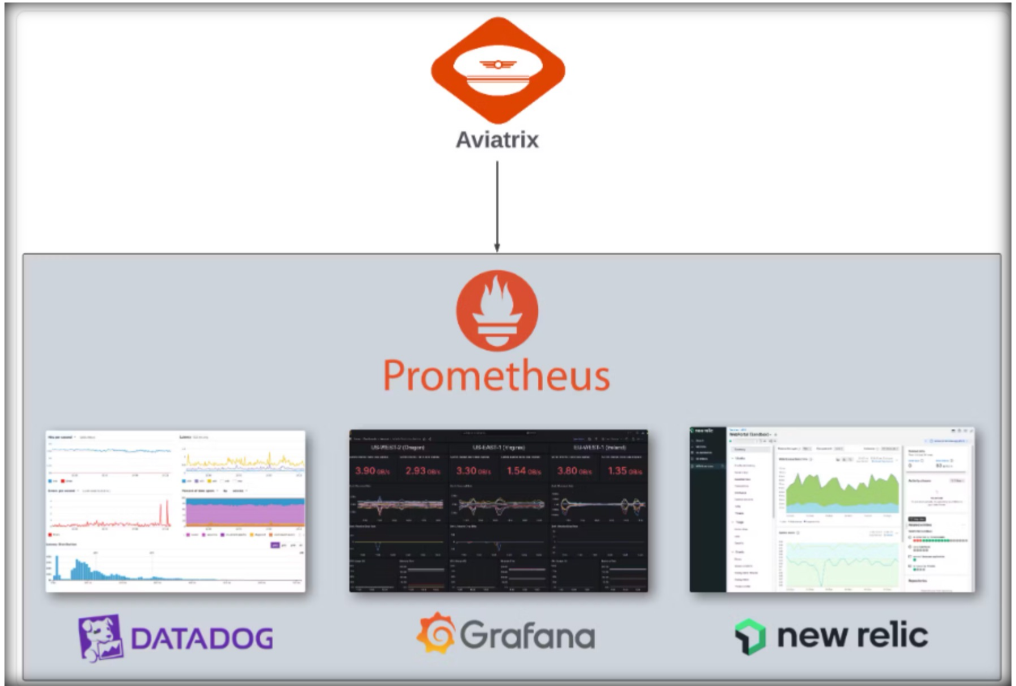
aviatrix
ACE
Aviatrix Certified Engineer

Network Insights API

aviatrix

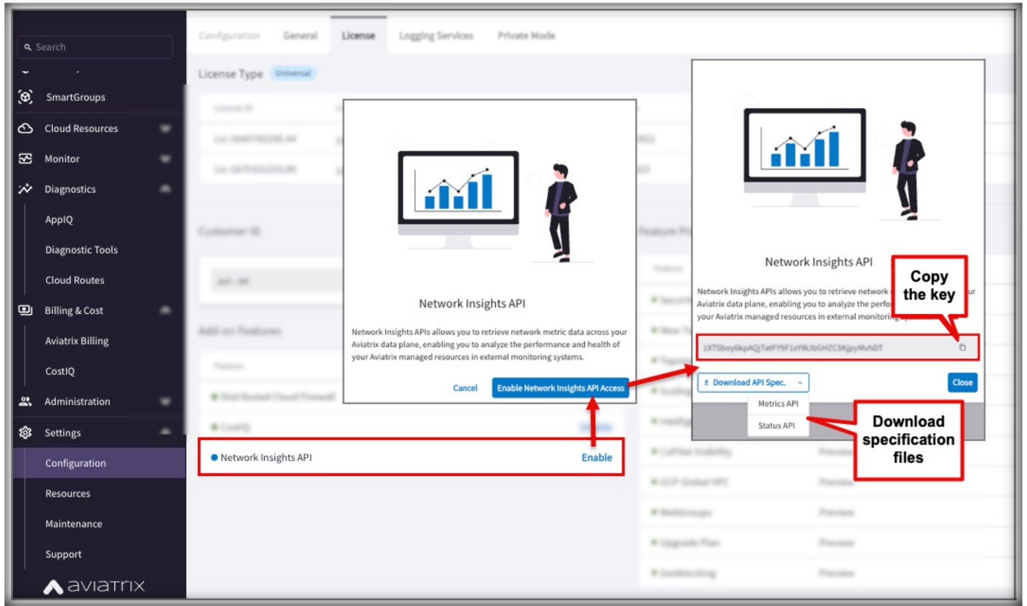
Network Insights API (part.1)

- The Aviatrix Network Insights API allows you to retrieve network metric and status data across your Aviatrix data plane. Using the metric and status APIs, you can integrate with ***third-party tools*** for data analysis and visualization of the performance and health of your Aviatrix-managed resources. The APIs also support data retention for compliance.



Network Insights API (part.2)

- The Network Insights API supports ***Prometheus*** and JSON formats. All data transmissions are encrypted using industry-standard protocols.
- An **API key** is used to authenticate requests for your Aviatrix services.
  - The Aviatrix API uses port 443, the same port as the CoPilot UI. Ensure that port 443 is accessible and not restricted by any Security Groups.



Network Insights API

Network Insights APIs allows you to retrieve network metric data across your Aviatrix data plane, enabling you to analyze the performance and health of your Aviatrix managed resources in external monitoring systems.

aviatrix

ACE

Aviatrix Certified Engineer

Aviatrix Sandbox Starter Tool

Build your own MCNA Transit at ~$1/hr

- Goal: Fast path for Customers and Partners to deploy Aviatrix multi-cloud transit foundation with minimal cost
- Turn-key solution to deploy Aviatrix Controller + MCNA in AWS and Azure + test instances with extreme simplicity and flexibility

- Can be deployed in 3 different ways:
  - Local (BYO Docker)
  - AMI
  - AMI with Terraform module

Description	Unit Cost	Quantity	Hourly Cost	Cost for 8 hours	Cost for 24 hours
Aviatrix Controller in AWS (t3.large)	$0.09	1	$0.09		
Aviatrix Gateway in AWS (t2.micro)	$0.01	3	$0.03		
Test instances in AWS (t2.micro)	$0.01	2	$0.02		
Aviatrix Encrypted Peering (AWS)	$0.23	2	$0.46		
Total Cost for AWS-only Transit + 2 Spokes			$0.60	$4.80	$14.40

Extending into Azure

Description	Unit Cost	Quantity	Hourly Cost	Cost for 8 hours	Cost for 24 hours
Aviatrix Gateway in Azure (B1s)	$0.01	3	$0.03		
Aviatrix Encrypted Peering (Azure)	$0.23	2	$0.46		
Aviatrix Transit Peering (between AWS and Azure)	$0.70	1	$0.70		
Total Cost for MCNA (including minimal network egress charges)			$1.19	$9.52	$28.56

User guide: https://community.aviatrix.com/t/g9hx9jh

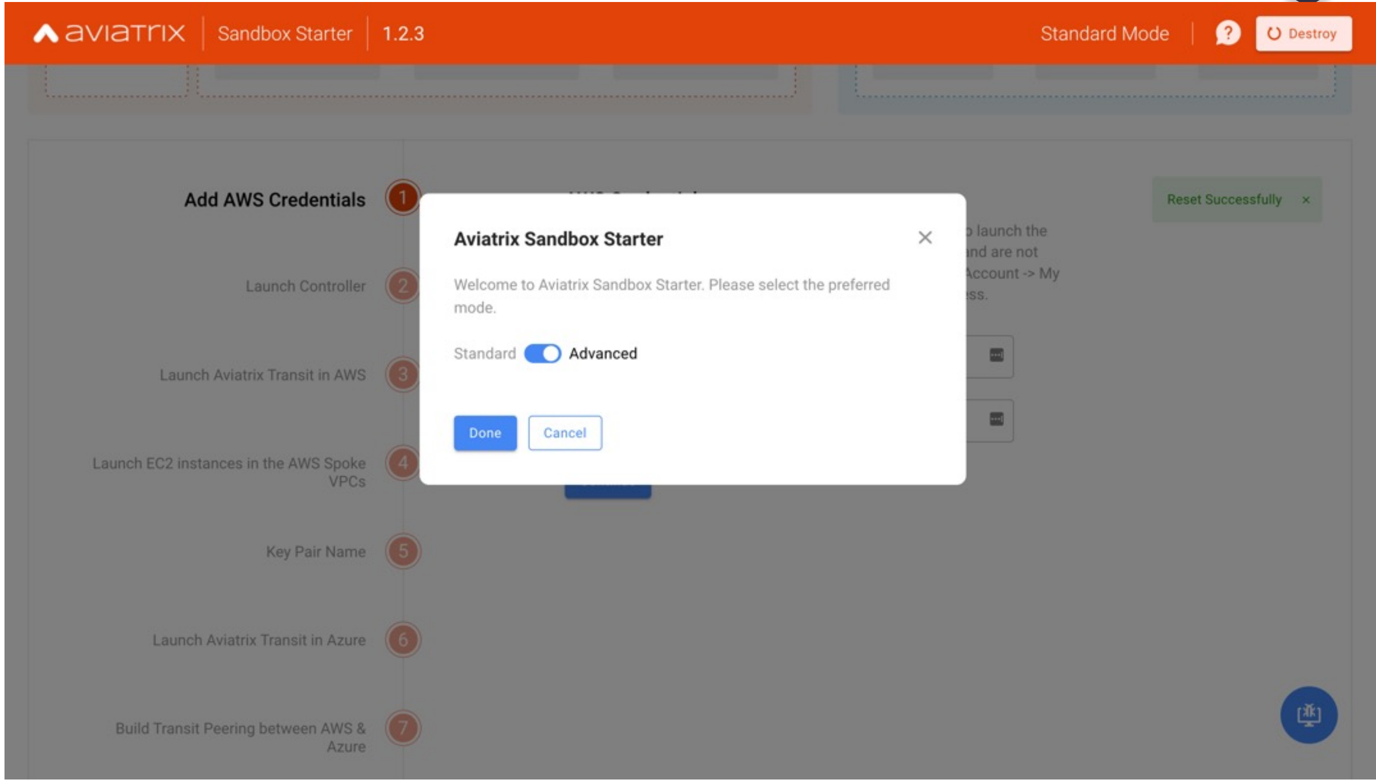
What Sandbox Starter Tool Builds

- Controller launch (Metered or BYOL)
  - VPC and all networking
  - Security Groups
  - Key pairs
  - IAM roles and policies (only if they don’t already exist)
  - EC2 instance
  - Username and password
  - Software upgrade
  - AWS account onboarding
  - Configuring License (BYOL)

- MCNA launch
  - Azure account onboarding
  - AWS VPCs and Azure VNets
    - Spoke and Transit
  - ActiveMesh Transit in AWS
    - Spoke gateways
    - Transit gateways
    - Spoke attachment to Transit
  - Same ActiveMesh Transit in Azure
  - Transit peering between AWS and Azure

Sandbox Starter Tool Modes

- Standard Mode
  - Fixed regions, resource names, and CIDR blocks
- Advanced Mode
  - Customizable regions, resource names, and CIDR blocks


aviatrix
Sandbox Starter
1.2.3
Standard Mode
Destroy
Add AWS Credentials
Launch Controller
Launch Aviatrix Transit in AWS
Launch EC2 instances in the AWS Spoke VPCs
Key Pair Name
Launch Aviatrix Transit in Azure
Build Transit Peering between AWS & Azure
Reset Successfully

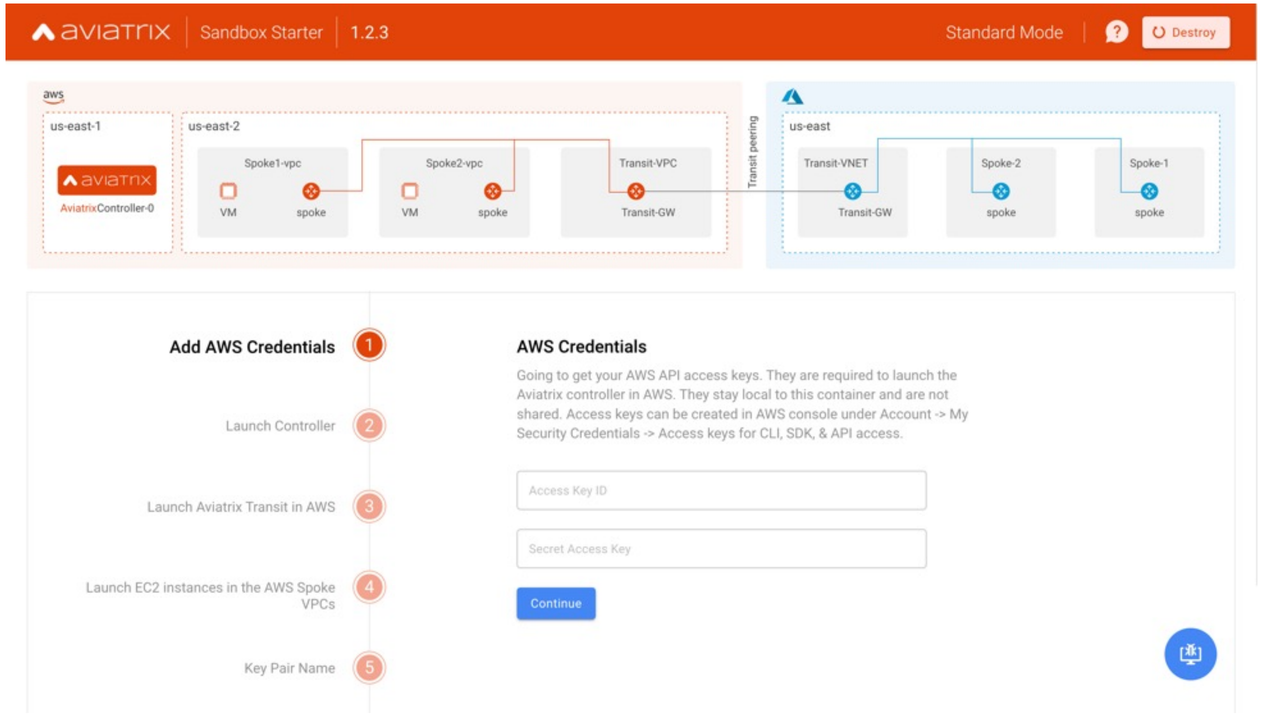
Aviatrix Sandbox Starter

Welcome to Aviatrix Sandbox Starter. Please select the preferred mode.

Standard [✓ Advanced]

Done Cancel

Sandbox Starter Tool Workflow Start



AWS Credentials

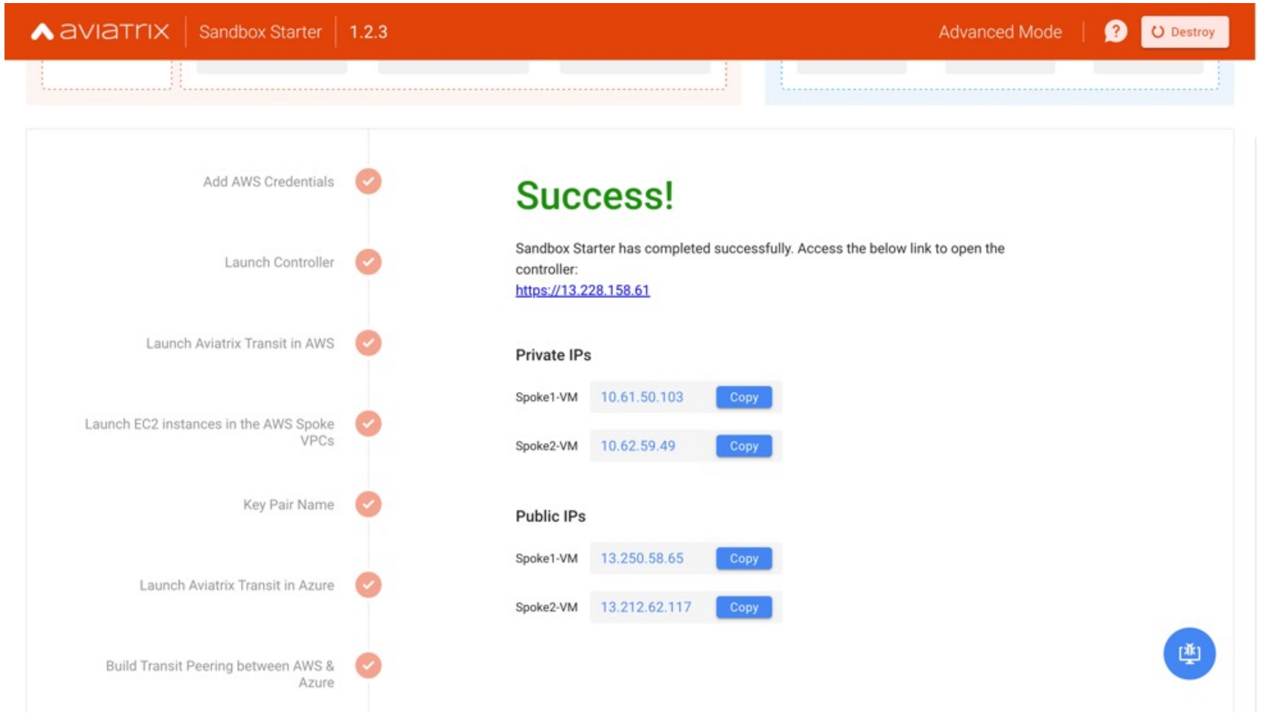
Going to get your AWS API access keys. They are required to launch the Aviatrix controller in AWS. They stay local to this container and are not shared. Access keys can be created in AWS console under Account -> My Security Credentials -> Access keys for CLI, SDK, & API access.

Access Key ID

Secret Access Key

Continue

Sandbox Starter Tool Workflow Completion


aviatrix
Sandbox Starter
1.2.3
Advanced Mode
Destroy
Add AWS Credentials
Launch Controller
Launch Aviatrix Transit in AWS
Launch EC2 instances in the AWS Spoke VPCs
Key Pair Name
Launch Aviatrix Transit in Azure
Build Transit Peering between AWS & Azure

Success!

Sandbox Starter has completed successfully. Access the below link to open the controller:
https://13.228.158.61

Private IPs

Spoke1-VM	10.61.50.103	Copy
Spoke2-VM	10.62.59.49	Copy

Public IPs

Spoke1-VM	13.250.58.65	Copy
Spoke2-VM	13.212.62.117	Copy

aviatrix

ACE

Aviatrix Certified Engineer

Next: Lab 11 Terraform and NetworkInsight API

aviatrix