



Operations, Visibility, and Troubleshooting in Public Cloud

Operational Challenges in Public Cloud

Evidential Data

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

Unfamiliar Toolset

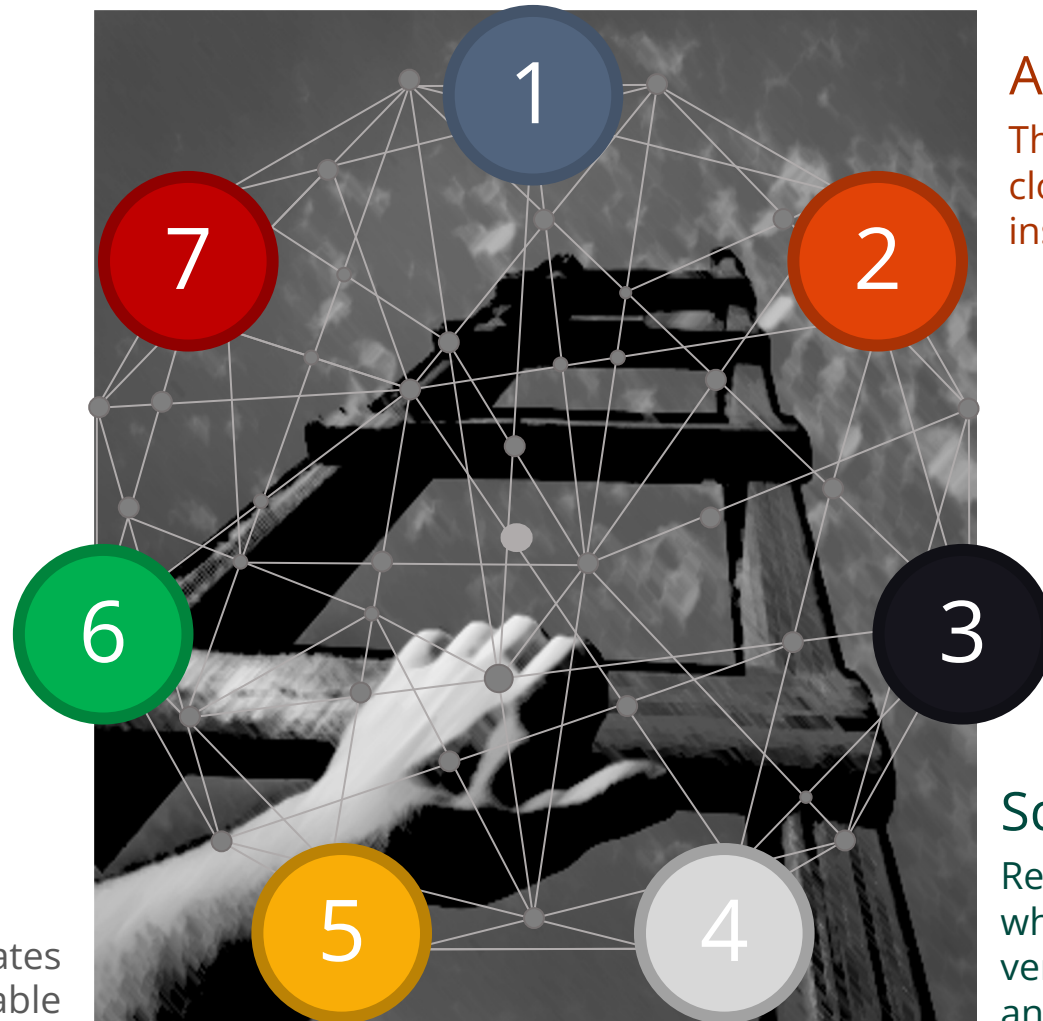
Native cloud lacks familiar tools like ping, packet capture, traceroute

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

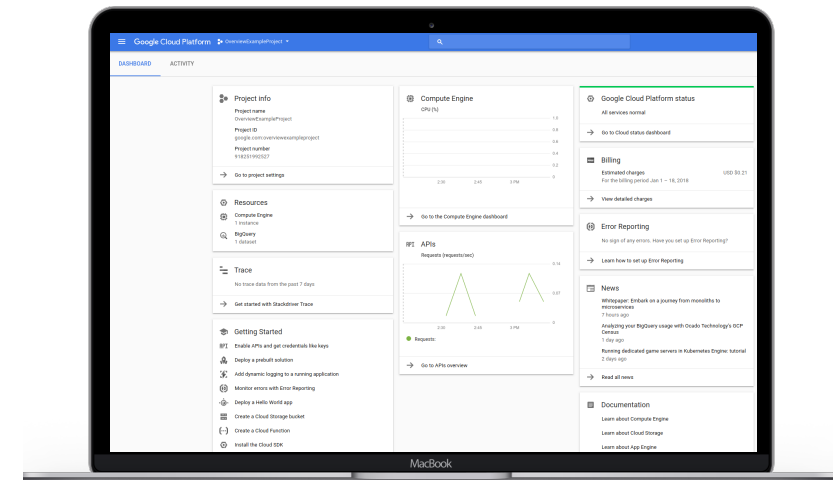
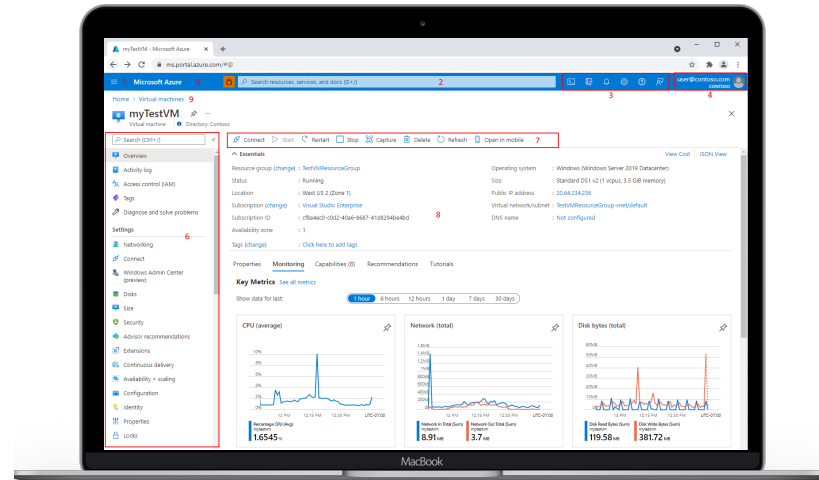
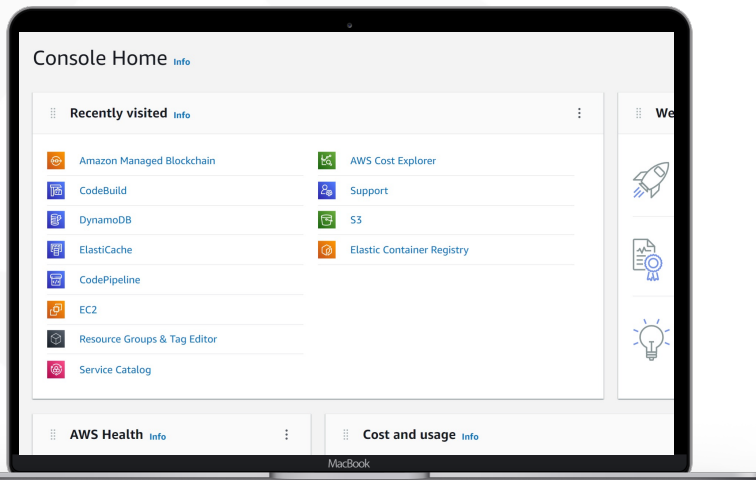
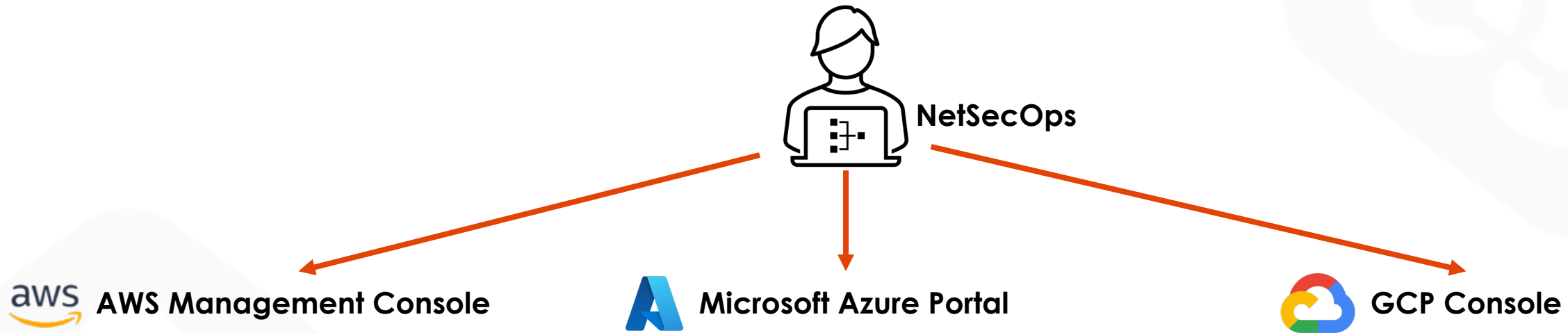
Tier-3 becomes Tier-1

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

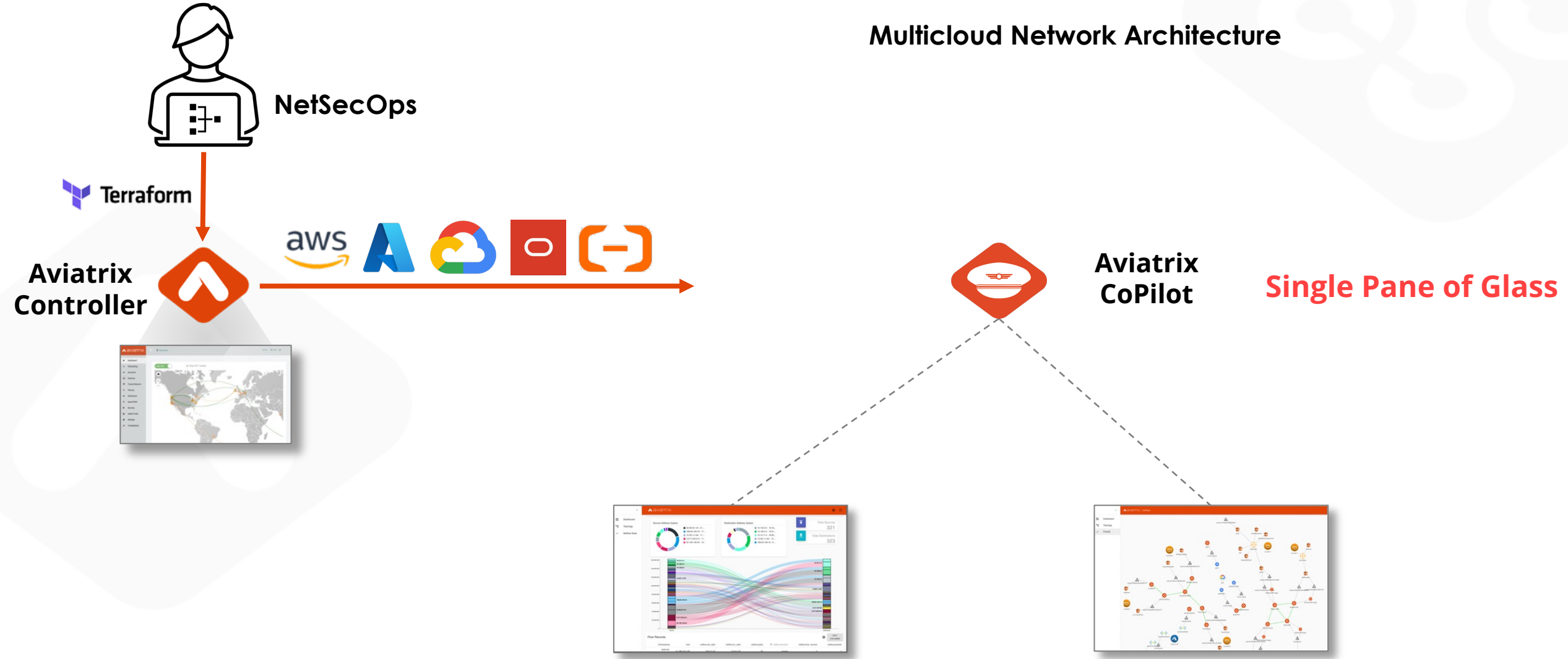
Real problems are experienced when architecture scales out as it very quickly grows to be complex and very hard to troubleshoot

Deploying and Operating Multicloud with Native Constructs



Deploying and Operating Multicloud with Aviatrix

Multicloud Network Architecture



Multicloud – Multi-Account







- Single pane of glass to manage all cloud accounts
- Support for AWS, AWS Gov, Azure, GCP etc. using same workflows, terminologies and tools
- Periodic Account Audits
 - To make sure they are intact and have needed IAM Roles, Policies and Trust Relationships (with Primary Accounts)
 - Notification sent if audit fails

Cloud Accounts

Account Name	Cloud	Account Number / ID	RBAC Group	Audit Status
aws-account	AWS	[REDACTED]	admin, local-students	Pass
azure-account	Azure ARM	[REDACTED]	admin, local-students	Pass
edge_admin	Aviatrix		admin, local-students	Pass
gcp-account	GCP	aviatrix-lab100	admin, local-students	Pass

Connect Cloud Account

Account Name

 Standard     

AWS Azure GCP OCI Alibaba Edge CSP

IAM Role-Based ☒

🔔 Launch the [CloudFormation Script](#) to establish the trust with your primary access account. (Skip if you have already run the script).

AWS Account Number

AWS App Role ARN Optional

AWS EC2 Role ARN Optional

Add to RBAC Groups Optional

☐ I have run the CloudFormation script to set up this secondary access account

Cancel Save



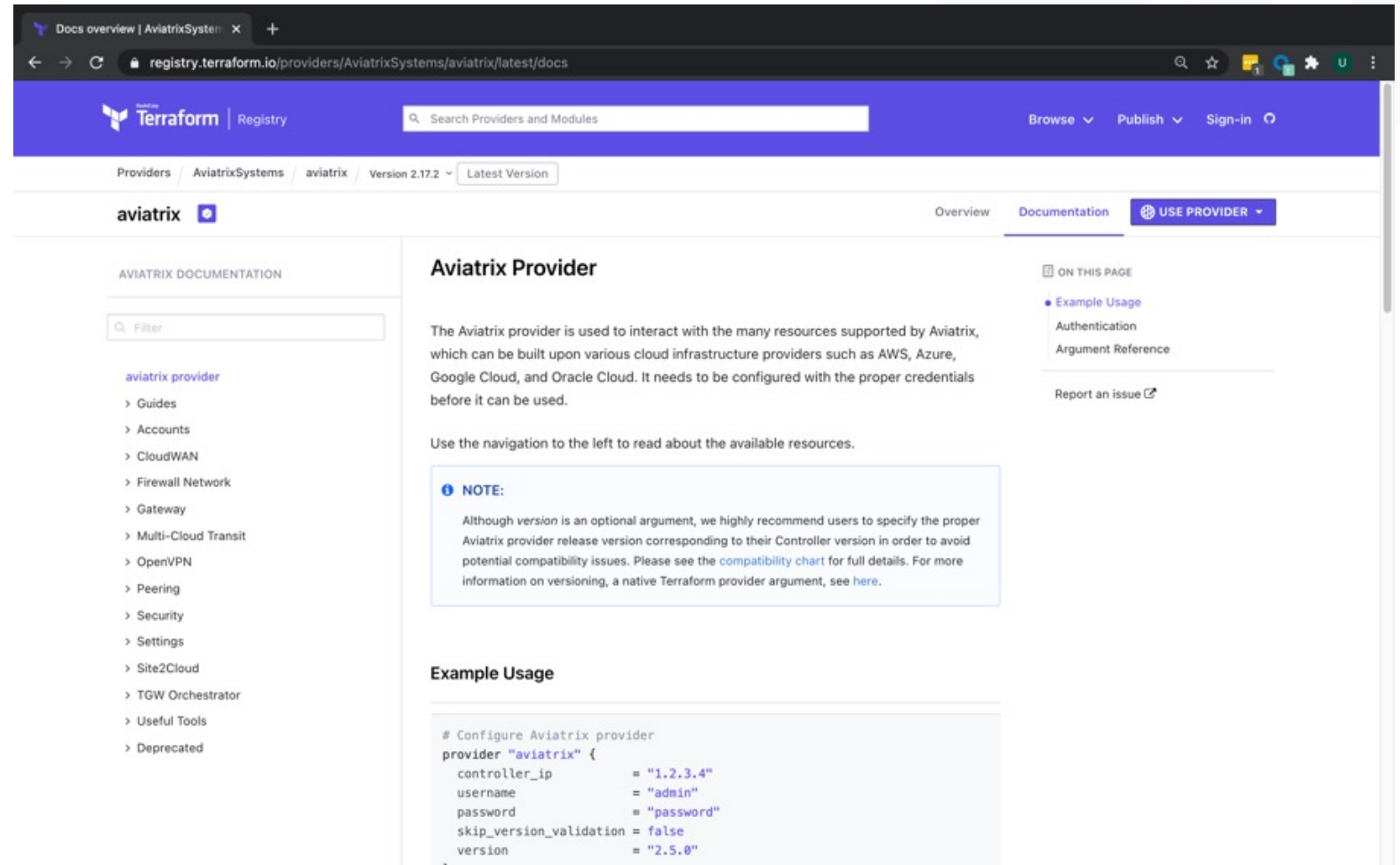
Infrastructure as Code

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

Aviatrx 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



The screenshot shows the Terraform Registry page for the Aviatrx provider. The page is titled "Aviatrx Provider" and includes a navigation sidebar on the left with links to "Guides", "Accounts", "CloudWAN", "Firewall Network", "Gateway", "Multi-Cloud Transit", "OpenVPN", "Peering", "Security", "Settings", "Site2Cloud", "TGW Orchestrator", "Useful Tools", and "Deprecated". The main content area contains a "NOTE" section stating: "Although version is an optional argument, we highly recommend users to specify the proper Aviatrx 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](#)." Below the note is an "Example Usage" section with a code block showing the configuration for the Aviatrx provider:

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

The right sidebar contains a "ON THIS PAGE" section with links to "Example Usage", "Authentication", and "Argument Reference", along with a "Report an issue" link.

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"

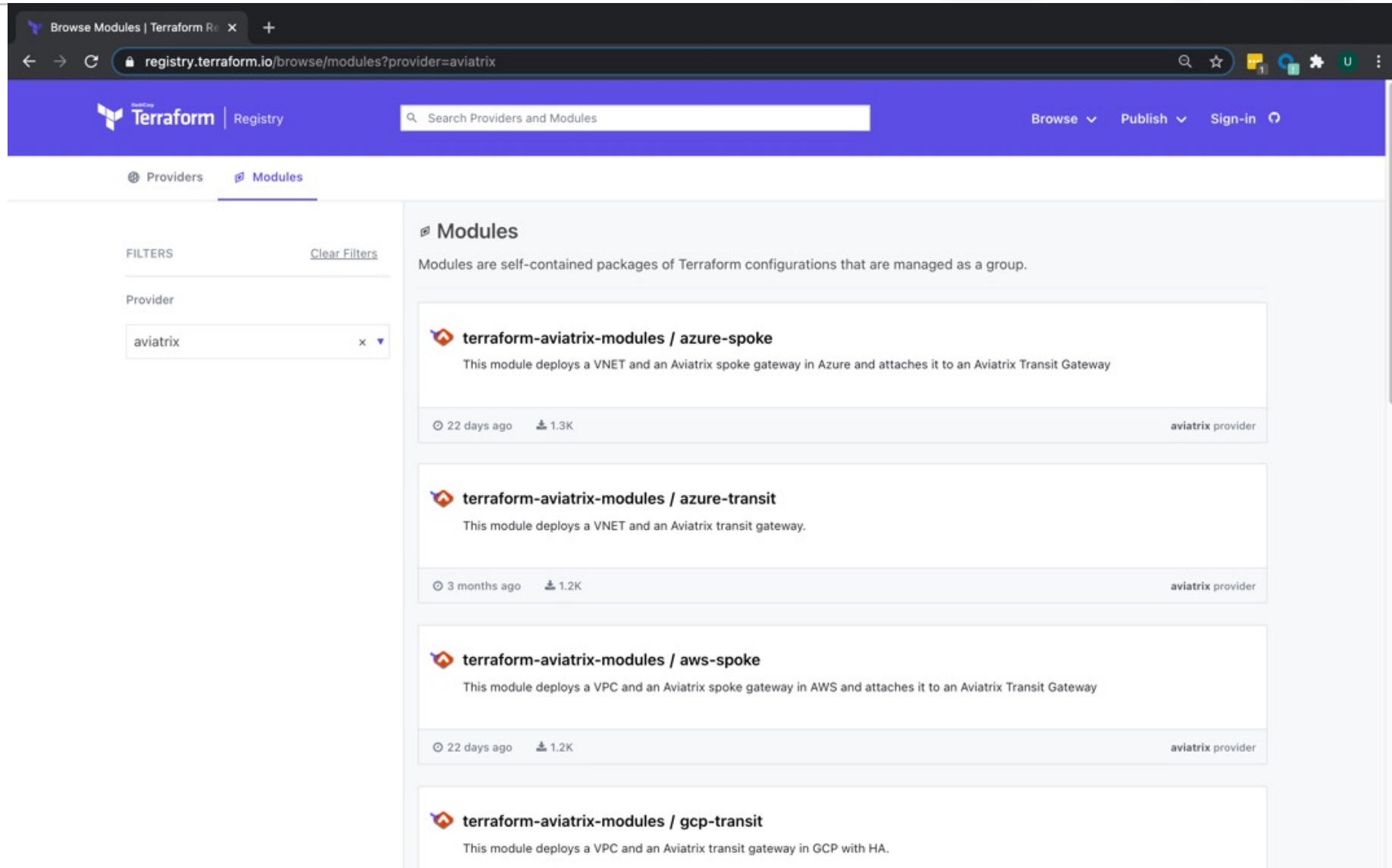
    gw_size      = "Standard_D2"

    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





Next: UI Walkthrough [Tour]



Aviatrix ApplQ

AppIQ – End-to-End Application Path Inspection and Troubleshooting



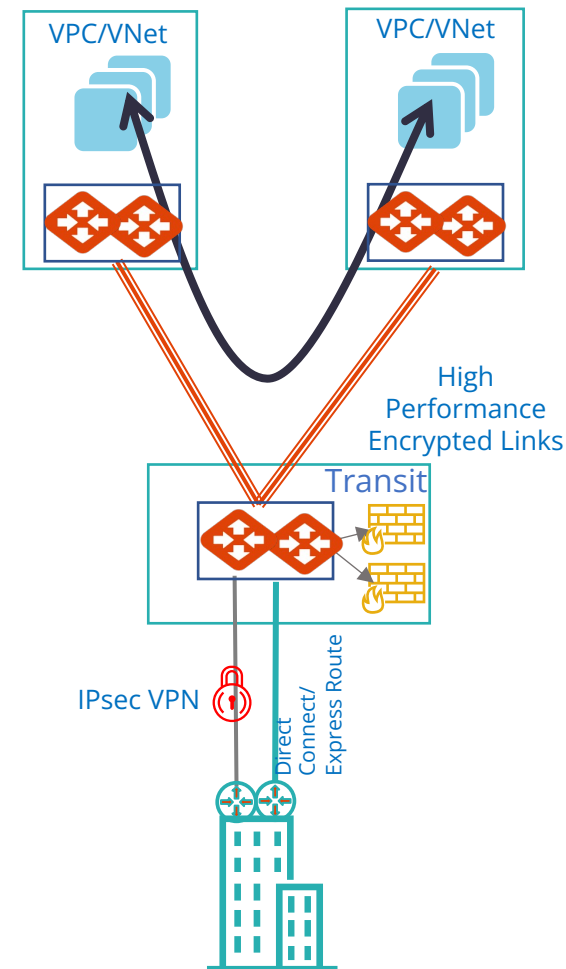
- Topology
- Gateway Telemetry
- FlowIQ
- FlightPath

Source to Destination

1. Source Instance name
2. Source IP address
3. Source VPC ID
4. Source Subnet ID
5. Source Route Table ID
6. Source Outbound Rules in Security Group used
7. Source outbound NACL rule
8. Source Transit Routing or VPC Peering Route
9. Destination Instance name
10. Destination IP address
11. Destination VPC ID
12. Destination Subnet ID
13. Destination Route Table ID
14. Destination Inbound NACL rule
15. Destination Inbound Rules in Security Group used

Return Traffic

16. Source Subnet ID
17. Source Route Table ID
18. Source outbound NACL rule
17. Source Transit Routing or VPC Peering Route
18. Destination Instance name
19. Destination IP address
20. Destination VPC ID
21. Destination Subnet ID
22. Destination Route Table ID
23. Destination Route used in Routing table
24. Destination Inbound NACL rule
25. Stateful FW rules



Familiar Troubleshooting Tools (Enterprise-grade Tools)



- ICMP-based tool (Ping, Traceroute, Trace Path)
- Active sessions
- Interface stats
- TCP-UDP connectivity tools
- Packet Capture (.pcap file)
- NetFlow (FlowIQ)

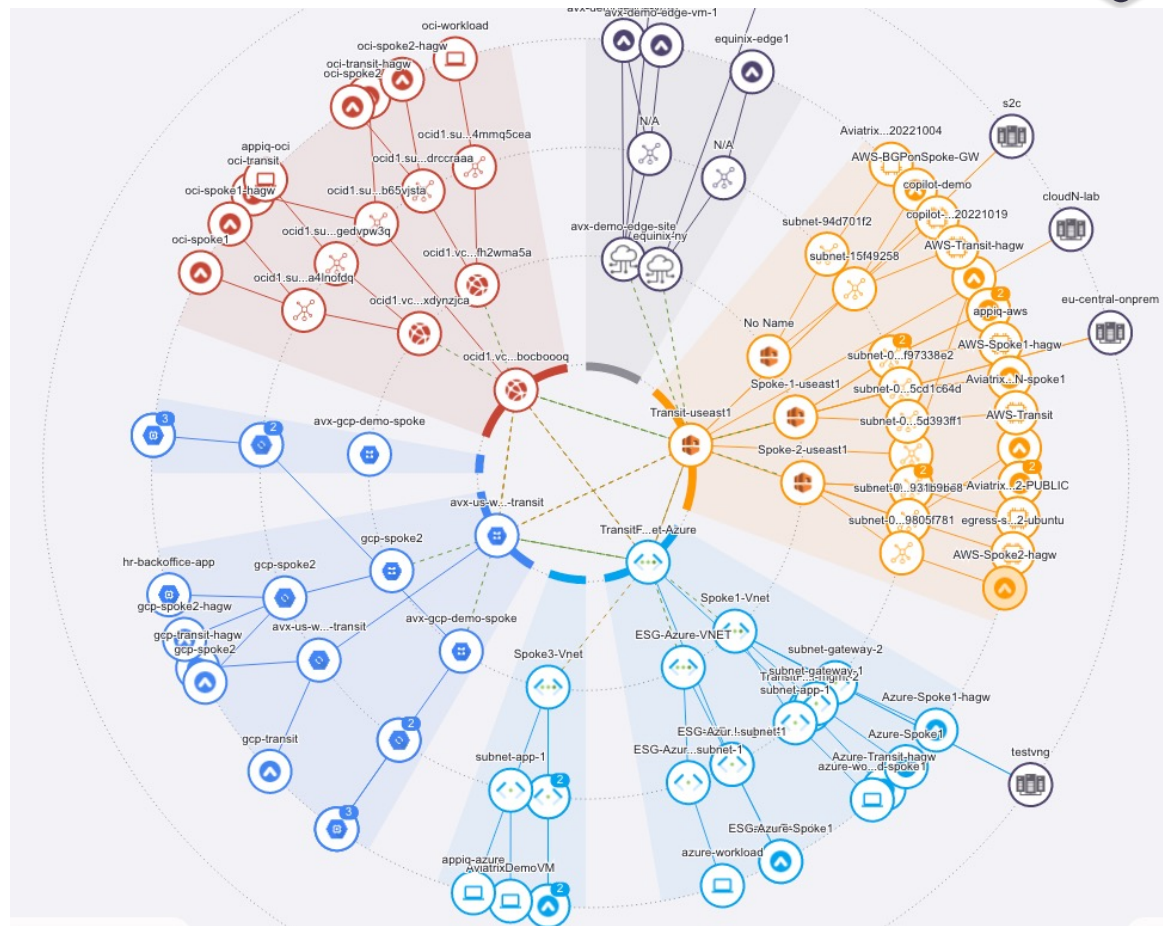
Virtual Machine
AWS-Spoke2-hagw

Properties

Search

Attribute	Value
Instance ID	i-026fc883af7ef50e3
Name	AWS-Spoke2-hagw
Type	Virtual Machine
Cloud	AWS
Account Name	Aviatrix-Demo
Region	us-east-1
VPC ID	vpc-04edee14459bd1f4b
Subnet ID	subnet-013f65a299805f781
Public IP	18.214.48.229
Private IP	10.3.131.75
Public DNS N...	ec2-18-214-48-229.compute-1.ama...
Private DNS N...	ip-10-3-131-75.ec2.internal
Associated Ga...	AWS-Spoke2-hagw
Instance Status	running

Diagnostic Tools



Gateway Diagnostics for AWS-Spoke2-hagw

Ping Traceroute Tracepath Tracelog Test Connectivity Active Sessions Interface Stats >

Destination: 1.1.1.1 Interface: Use Route Table ▼ Ping



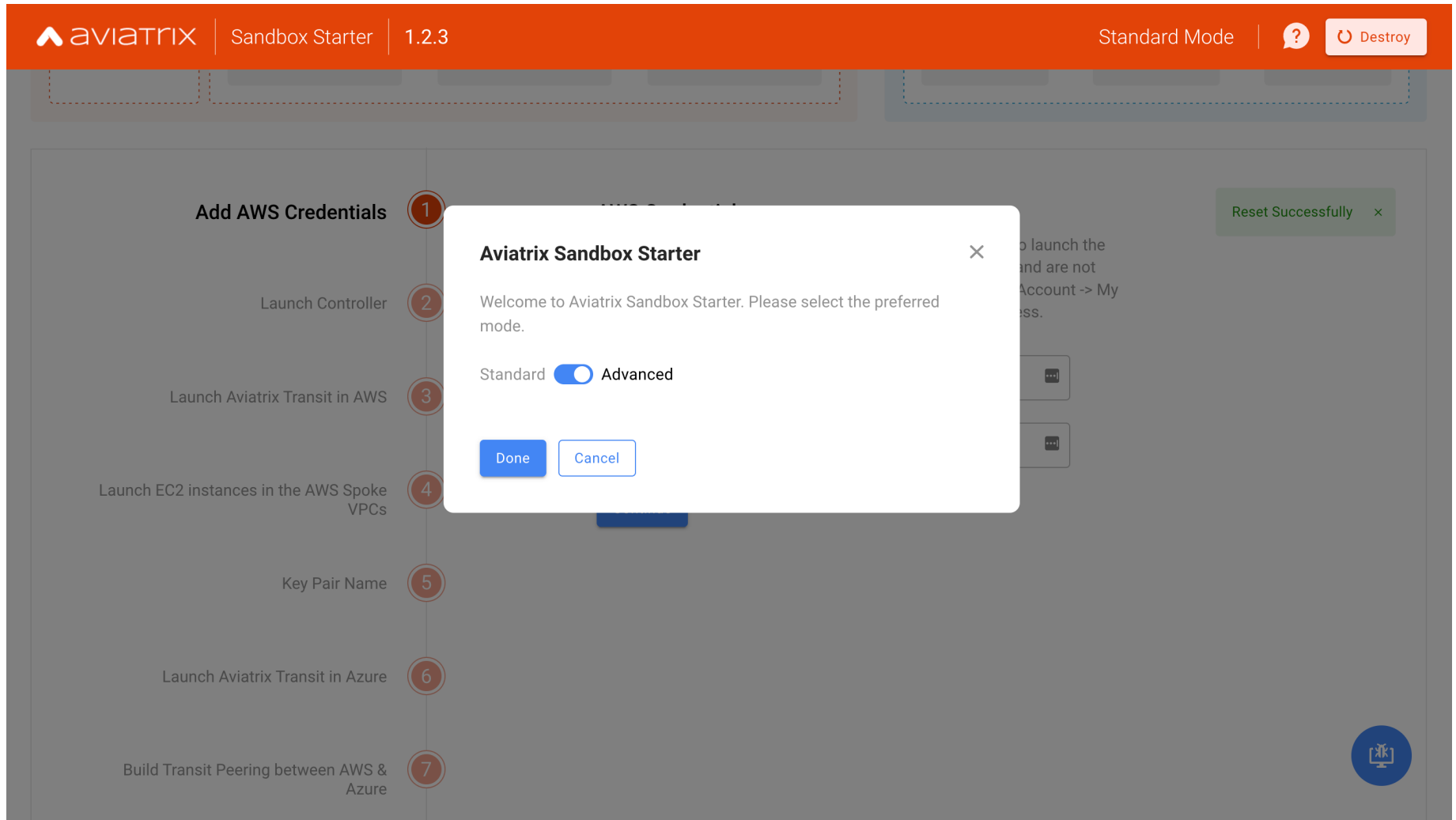
Aviatrix Learning and Testing Tool

For Pre-upgrade /SRE/Learning/Study/etc.

Sandbox Starter Tool Modes



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



Sandbox Starter Tool Workflow Start



aviatrix

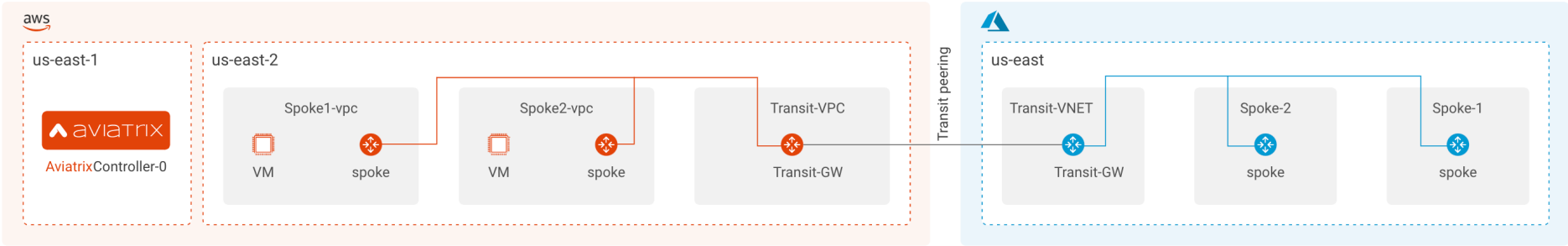
Sandbox Starter

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

1

2

3

4

5

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

