



IaC and Network Insights API

ACE Team



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

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

The screenshot shows the Terraform Registry page for the **aviatrix** provider. The page is titled "Aviatrix Provider" and includes a search bar, navigation tabs (Overview, Documentation, USE PROVIDER), and a sidebar with a list of resources under "AVIATRIX DOCUMENTATION". The main content area describes the provider's purpose and includes a "NOTE" about versioning. An "Example Usage" section shows a Terraform configuration snippet for the provider.

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

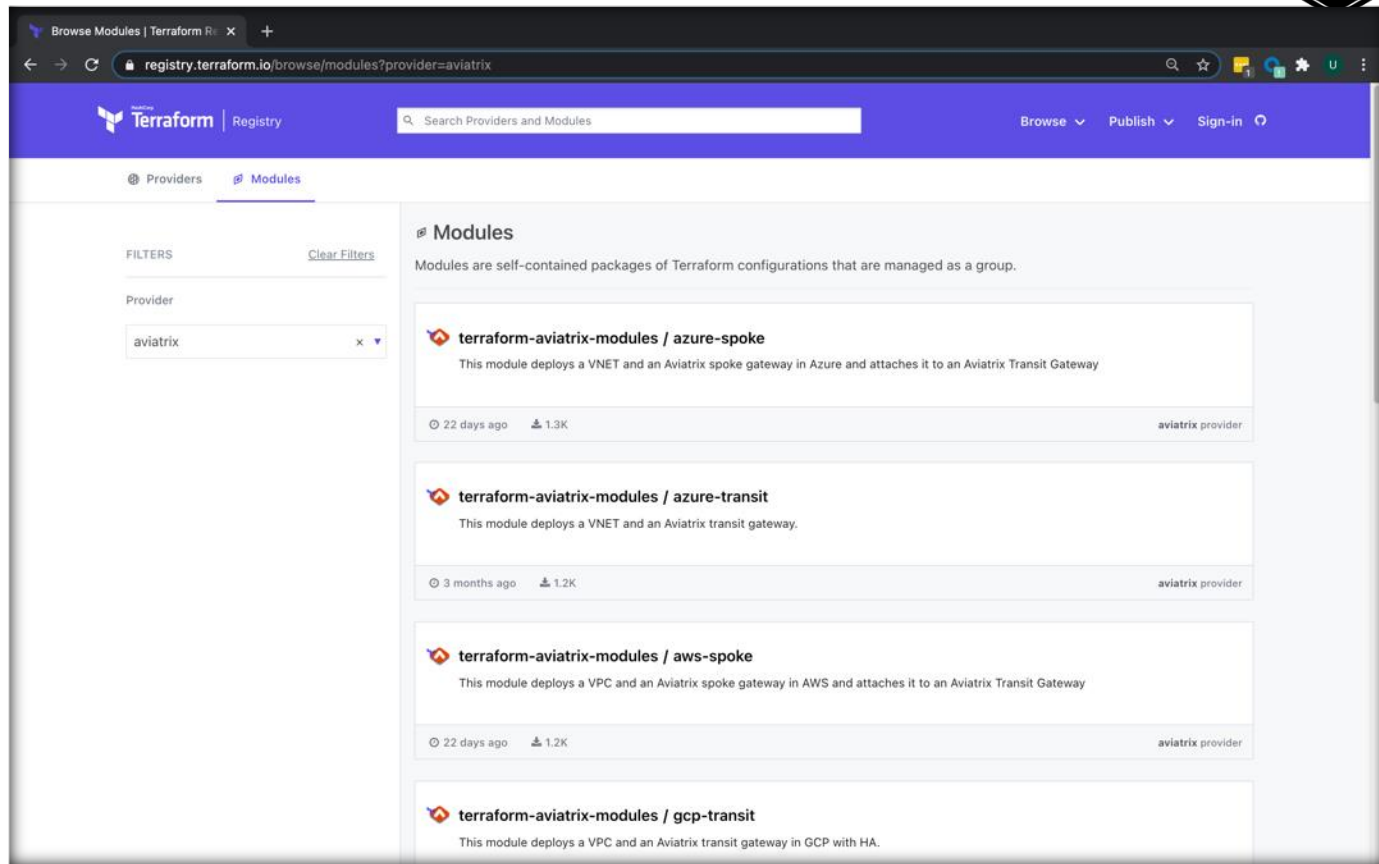
    account_name = "devops-azure"

    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

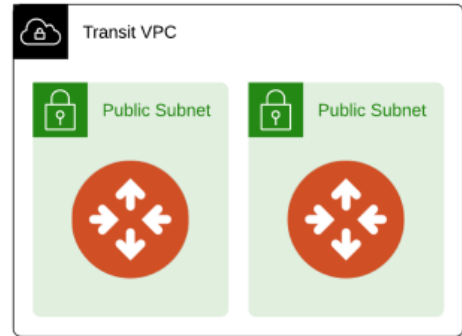
- ***“Repeatable++”***
- Similar to the concepts of libraries, packages, or modules found in most programming languages
- Provide many of the same benefits
- ~10X reduction in lines of code
- Can be found on Terraform Registry



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
```

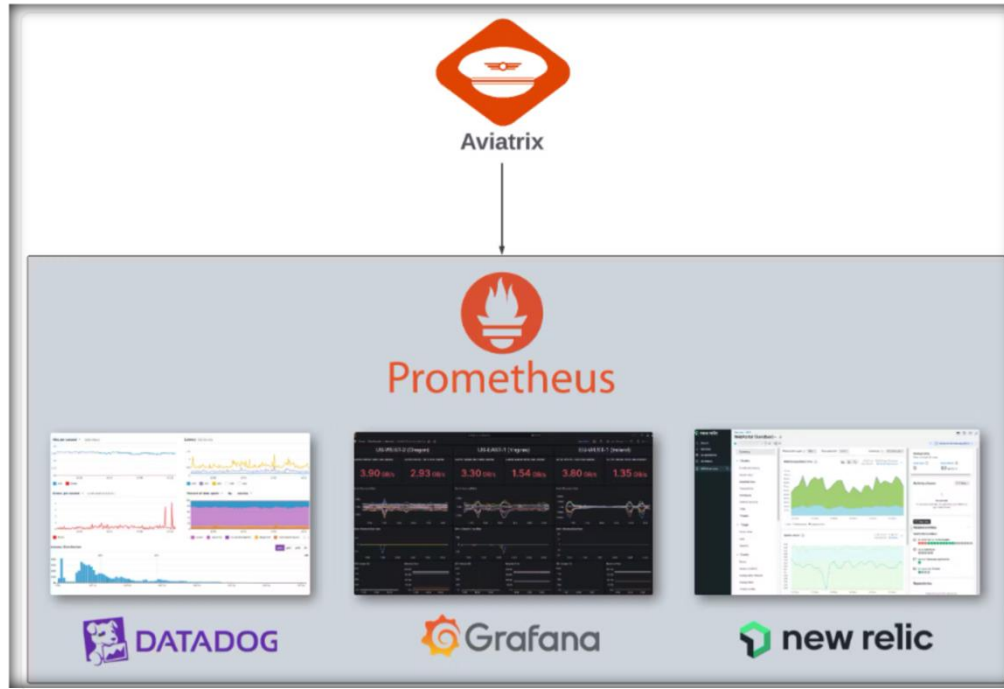




Network Insights API

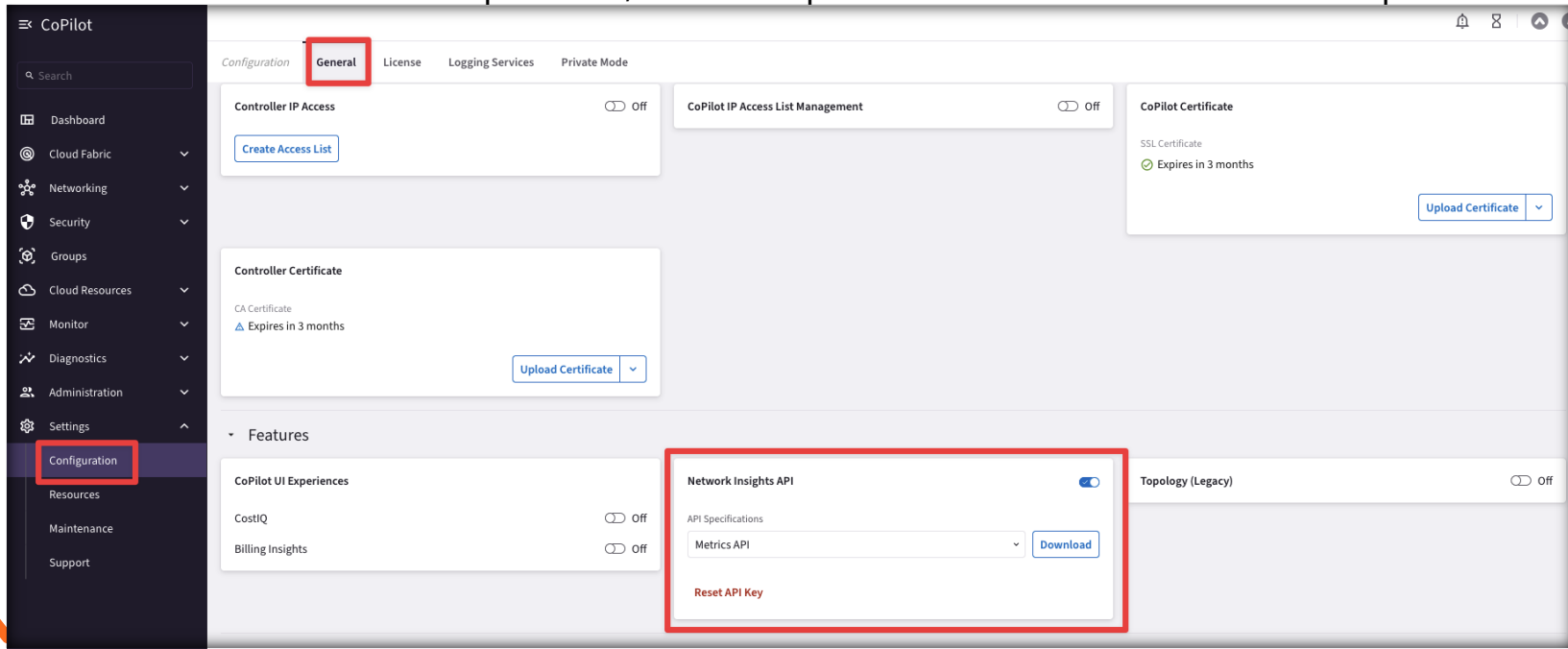
Network Insights API (part.1)

- 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



The screenshot displays the Aviatrix CoPilot Configuration interface. The left sidebar contains a navigation menu with the following items: Dashboard, Cloud Fabric, Networking, Security, Groups, Cloud Resources, Monitor, Diagnostics, Administration, Settings, Configuration (highlighted with a red box), Resources, Maintenance, and Support. The main content area is titled 'Configuration' and has a sub-tab 'General' (also highlighted with a red box). The 'General' tab is divided into several sections: 'Controller IP Access' (with a toggle switch set to 'Off' and a 'Create Access List' button), 'Controller Certificate' (showing 'CA Certificate' and 'Expires in 3 months' with an 'Upload Certificate' button), 'CoPilot IP Access List Management' (with a toggle switch set to 'Off'), 'CoPilot Certificate' (showing 'SSL Certificate' and 'Expires in 3 months' with an 'Upload Certificate' button), 'Features' (with a toggle switch set to 'Off'), 'CoPilot UI Experiences' (with a toggle switch set to 'Off'), 'CostIQ' (with a toggle switch set to 'Off'), 'Billing Insights' (with a toggle switch set to 'Off'), 'Network Insights API' (with a toggle switch set to 'On', a dropdown menu for 'API Specifications' set to 'Metrics API', a 'Download' button, and a 'Reset API Key' link), and 'Topology (Legacy)' (with a toggle switch set to 'Off').



Next: Lab 12 - Terraform and
Network Insights API