

cisco *Live!*

Let's go



The bridge to possible

Zero Touch Provisioning & Configuration Management of Cisco FTD in Azure

Using Terraform and Ansible

Ed McNicholas, Sr. Cybersecurity Architect
@don't have one

cisco Live!

DEVNET-2150

Cisco Webex App

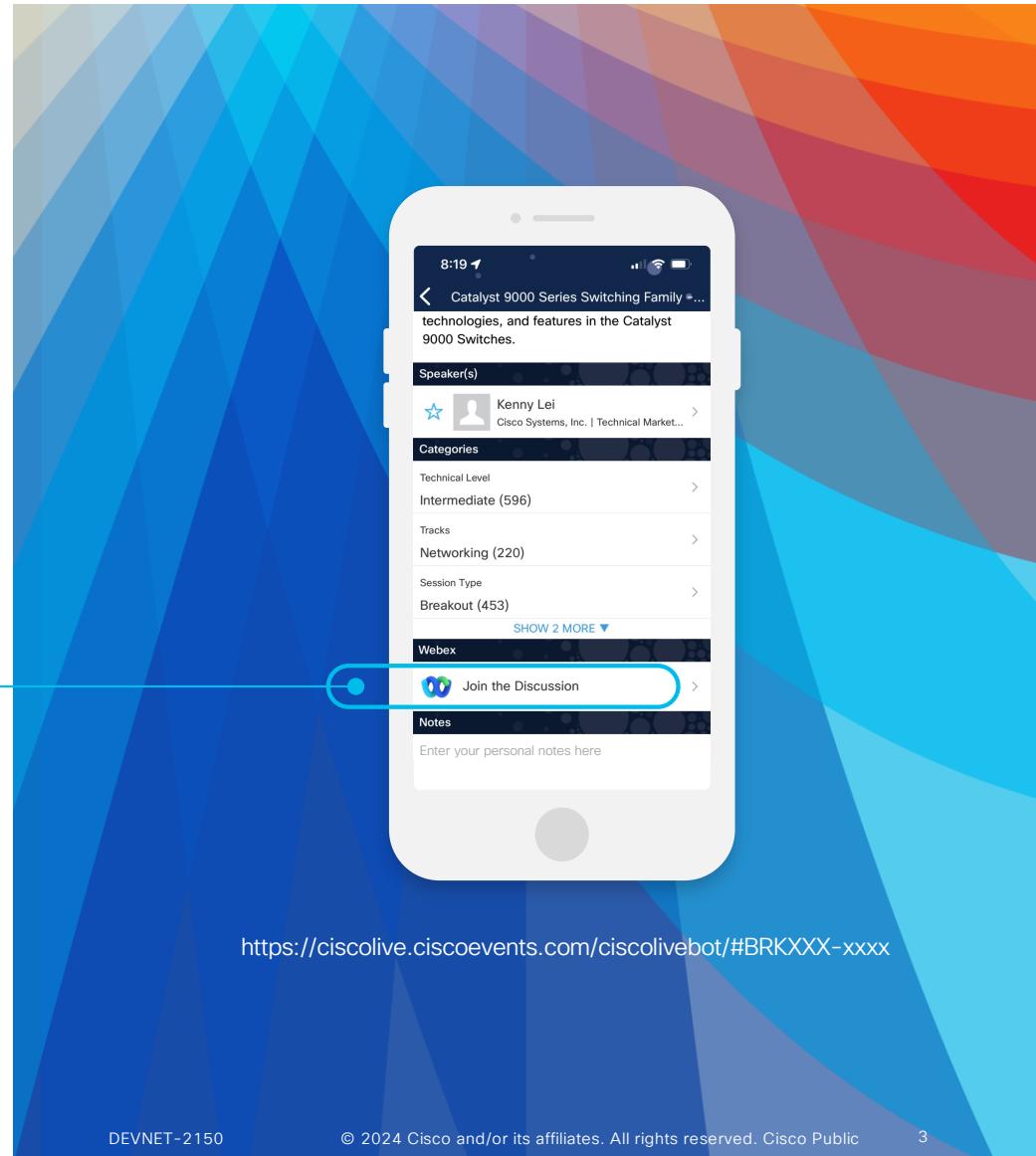
Questions?

Use Cisco Webex App to chat with the speaker after the session

How

- 1 Find this session in the Cisco Live Mobile App
- 2 Click “Join the Discussion”
- 3 Install the Webex App or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until February 23, 2024.



Agenda

cisco Live!

- Introduction
- Azure
- Terraform
- Ansible
- Conclusion

Introduction

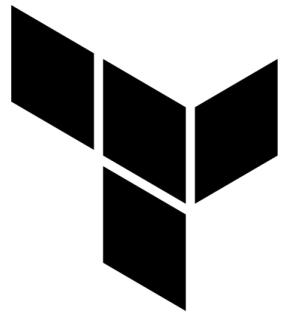
cisco *Live!*



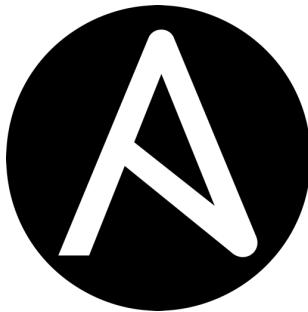
Automating FTD in Azure



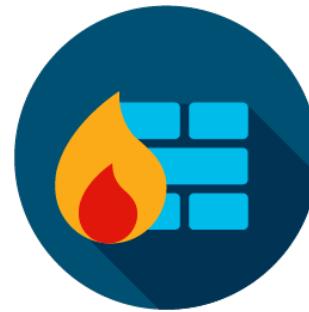
Azure



Terraform



Ansible



Secure
Firewall



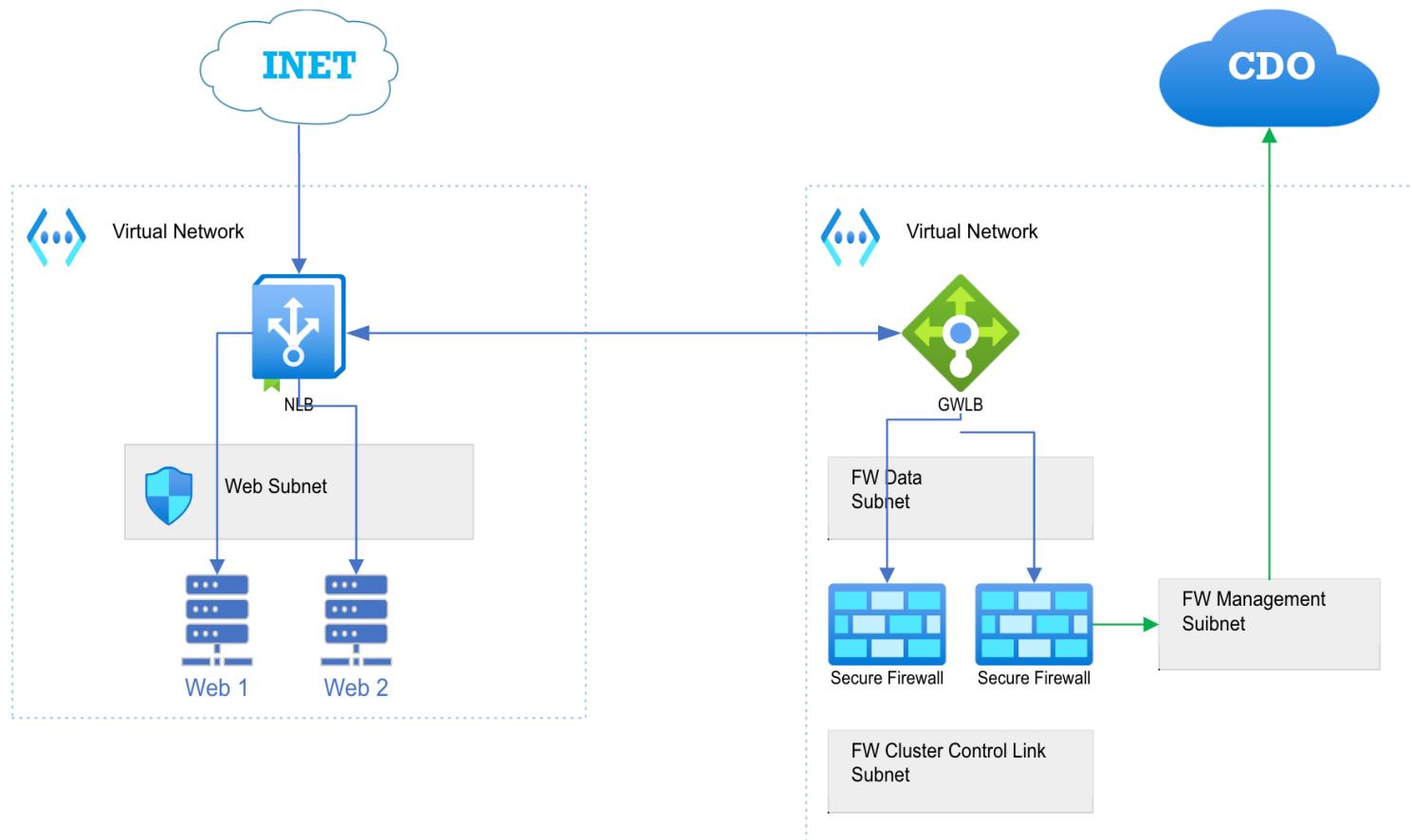
Defense
Orchestrator

GitHub Repo - https://github.com/emcnicholas/Azure_TF_GWLB.git

DEVNET-2150

6

The Architecture



Before diving in headfirst

Let's talk terminology

cisco *Live!*



Ansible – Imperative & Mutable

- Shallow Learning Curve
- Agentless
- Supports Every Device from Every Vendor
- Modular
- Open-Source Community and Vendor Commercial Support
- Common Toolset with server / DevOps / infra teams

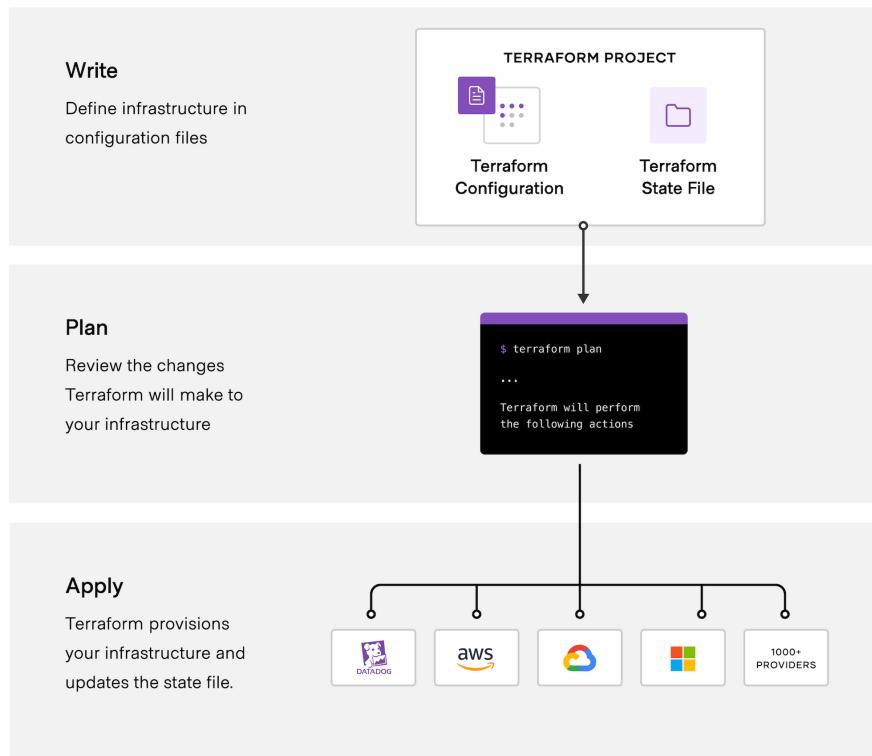


Terraform – Declarative & Immutable

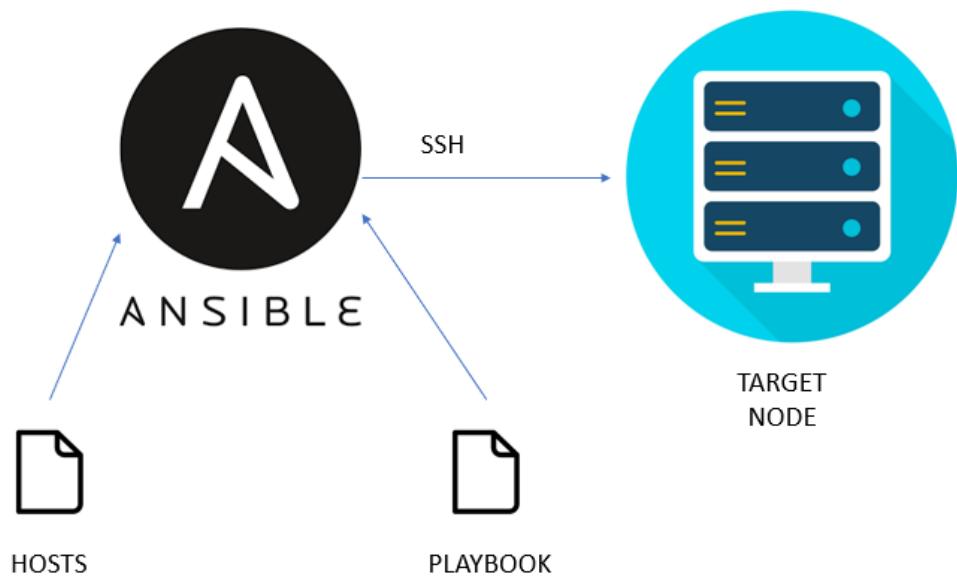
- Developed by HashiCorp
- Initial release in July 2014
- Designed to be a full Infrastructure as Code (IaC) management tool for Cloud Infrastructure Provisioning
- Completely written in Go, creating a single binary file 
- Fully declarative leveraging HashiCorp Configuration Language (HCL)



Terraform to Deploy



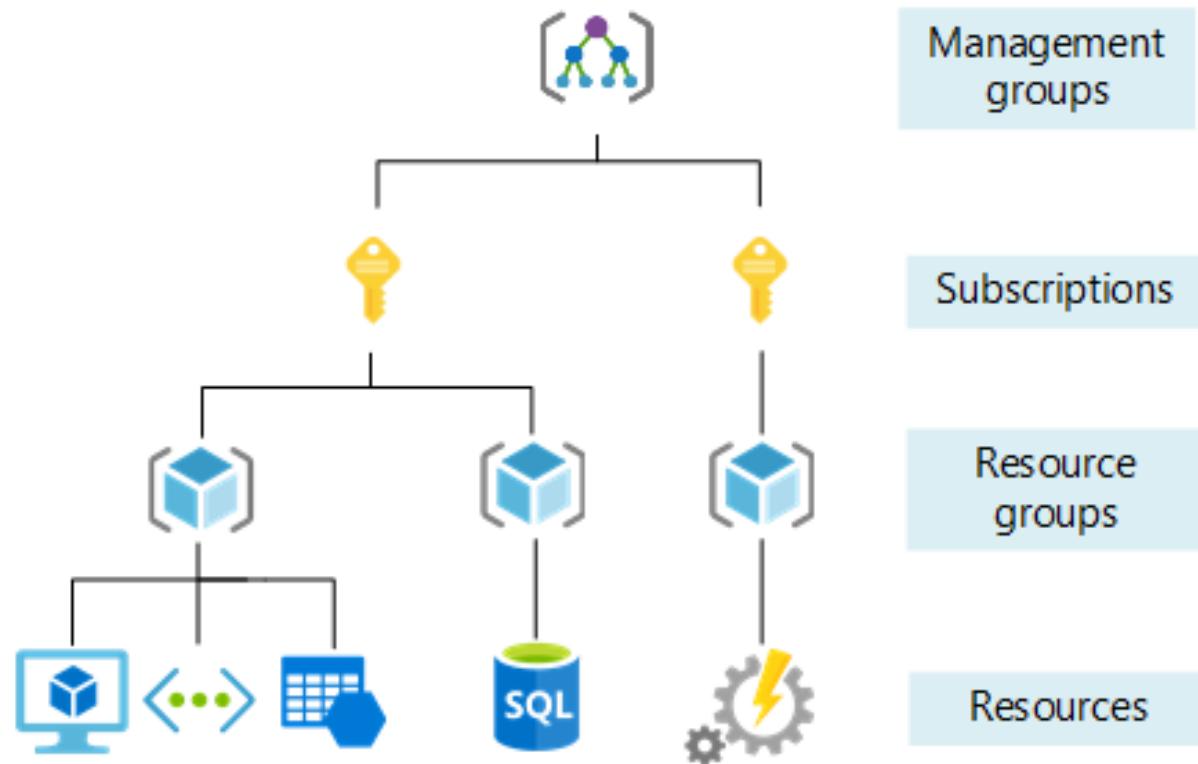
Ansible to Configure



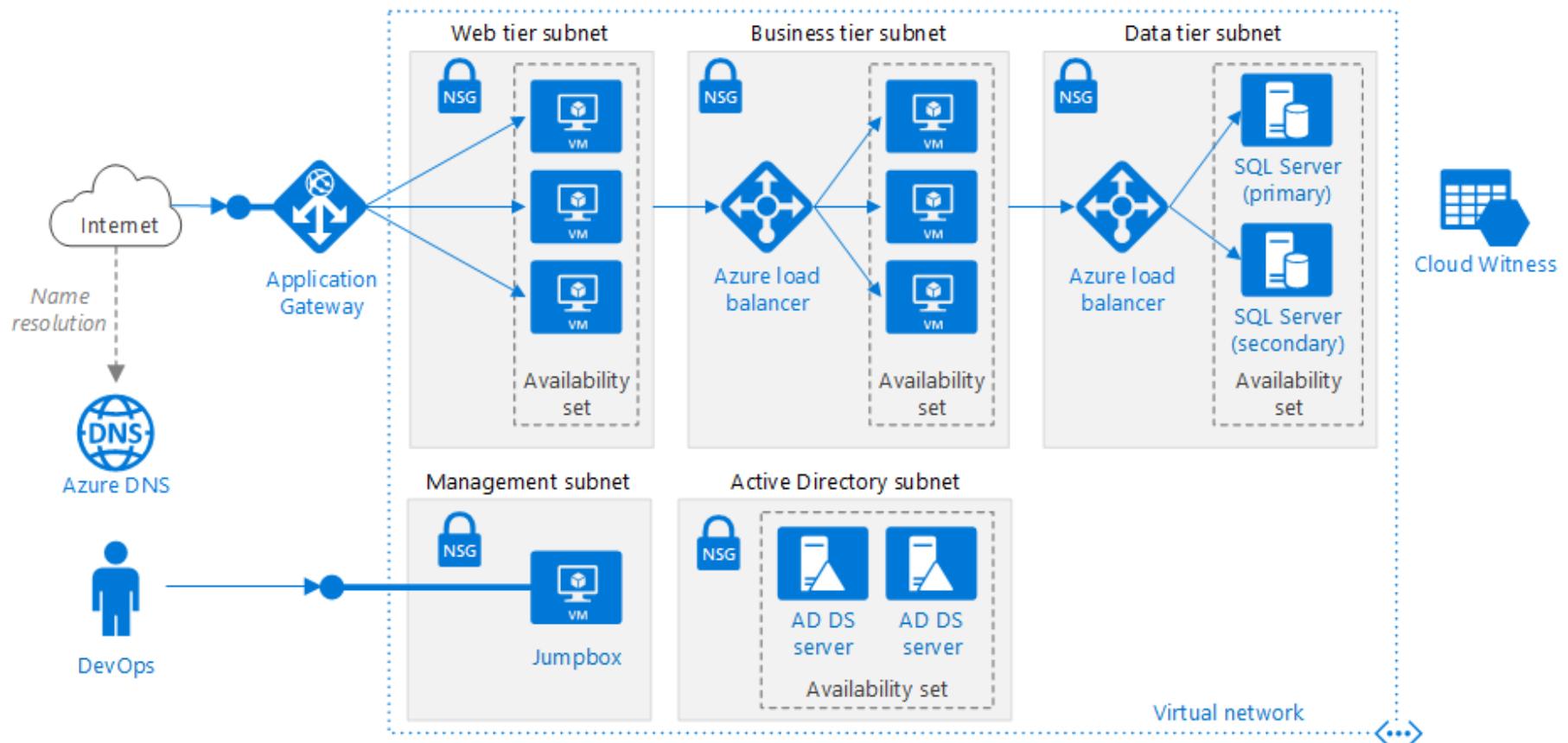
Azure

cisco *Live!*

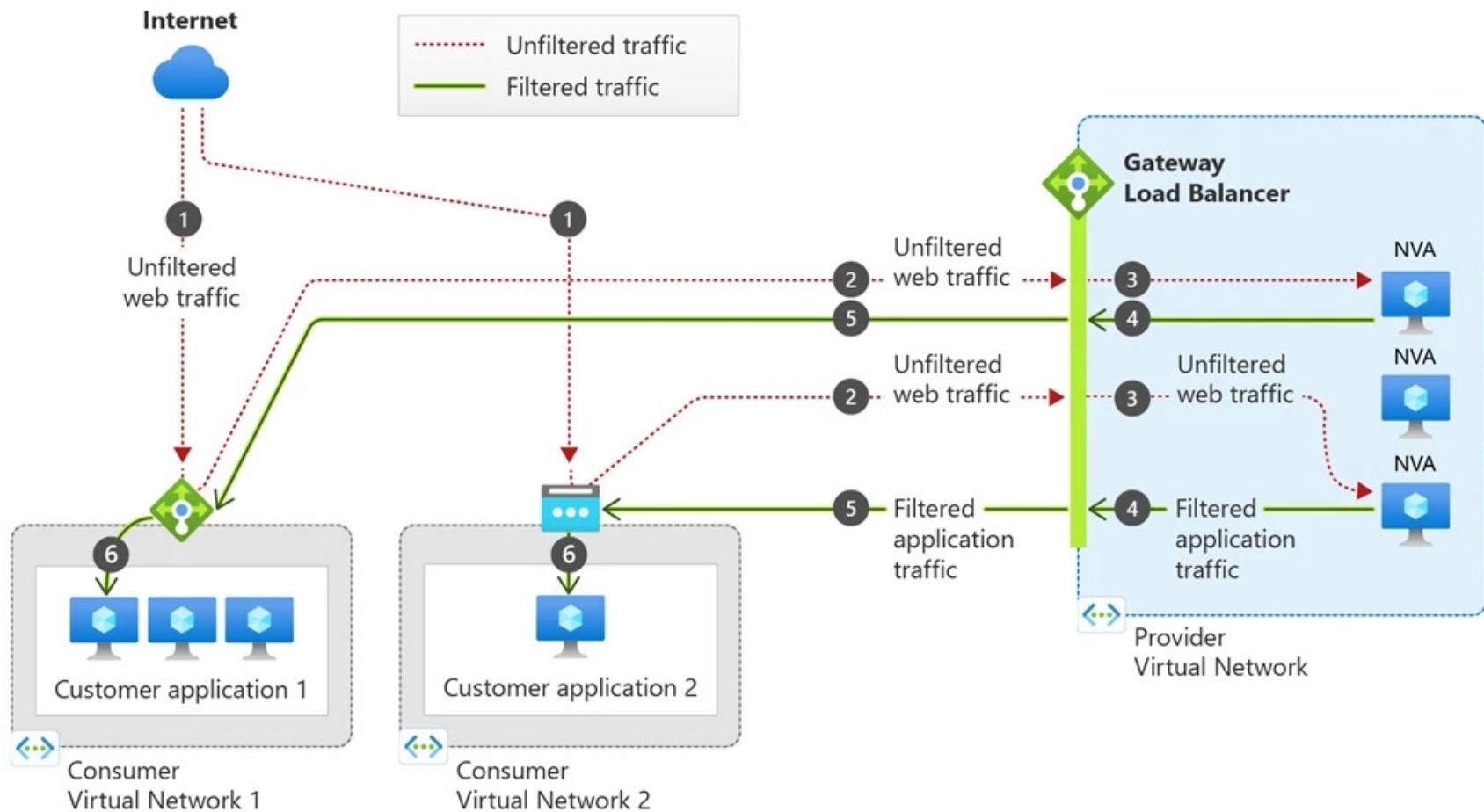
Azure Resources



Azure Resources



Azure Gateway Load Balancer



Terraform

cisco *Live!*

Providers

- Microsoft Azure
 - <https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs>
- Cisco FMC
 - <https://registry.terraform.io/providers/CiscoDevNet/fmc/latest/docs>
- Cisco CDO
 - <https://registry.terraform.io/providers/CiscoDevNet/cdo/latest/docs>

```
terraform {  
  required_providers {  
    azurerm = {  
      source  = "hashicorp/azurerm"  
      version = "3.89.0"  
    }  
    fmc   = {  
      source  = "CiscoDevNet/fmc"  
      version = ">=1.4.5"  
    }  
    cdo   = {  
      source  = "CiscoDevNet/cdo"  
      version = ">=0.7.2, <1.0.0"  
    }  
  }  
  
provider "azurerm" {  
  features {}  
}  
  
provider "fmc" {  
  is_cdfmc  = true  
  cdo_token = var.cdo_token  
  fmc_host  = var.cdfMC  
  cdfmc_domain_uuid = var.cdfmc_domain_uuid  
}  
  
provider "cdo" {  
  base_url  = var.cdo_base_url  
  api_token = var.cdo_token  
}
```

GWLB Virtual Network and Subnets

```
resource "azurerm_virtual_network" "gwlb" {
    name          = "gwlb-net"
    address_space = ["10.100.0.0/16"]
    resource_group_name = azurerm_resource_group.gwlb.name
    location      = azurerm_resource_group.gwlb.location
}

resource "azurerm_subnet" "fw_management" {
    name          = "fw-management"
    resource_group_name = azurerm_resource_group.gwlb.name
    virtual_network_name = azurerm_virtual_network.gwlb.name
    address_prefixes     = [cidrsubnet(azurerm_virtual_network.gwlb.address_space[0], 8, 1)]
}

resource "azurerm_subnet" "fw_data" {
    name          = "fw-data"
    resource_group_name = azurerm_resource_group.gwlb.name
    virtual_network_name = azurerm_virtual_network.gwlb.name
    address_prefixes     = [cidrsubnet(azurerm_virtual_network.gwlb.address_space[0], 8, 2)]
}

resource "azurerm_subnet" "fw_ccl" {
    name          = "fw-ccl"
    resource_group_name = azurerm_resource_group.gwlb.name
    virtual_network_name = azurerm_virtual_network.gwlb.name
    address_prefixes     = [cidrsubnet(azurerm_virtual_network.gwlb.address_space[0], 8, 3)]
```

GWLB Virtual Network and Subnets

 **gwlb-net** | Subnets ☆ ...

Virtual network

«

+ Subnet + Gateway subnet ↻ Refresh | 👤 Manage users trash Delete

🔍

Name ↑↓	IPv4 ↑↓	IPv6 ↑↓	Available IPs
fw-management	10.100.1.0/24	-	248
fw-data	10.100.2.0/24	-	249
fw-ccl	10.100.3.0/24	-	250

🔗 Overview

📅 Activity log

👤 Access control (IAM)

🏷 Tags

✖ Diagnose and solve problems

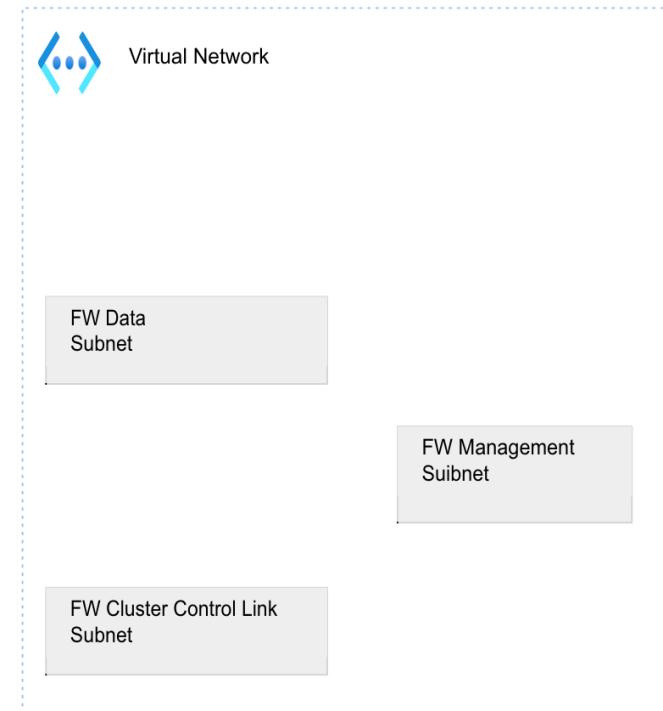
Settings

🔗 Address space

🔗 Connected devices

🔗 Subnets

GWLB Virtual Network and Subnets



Web App Virtual Network and Subnets

```
resource "azurerm_virtual_network" "web" {
    name          = "web-net"
    address_space = ["10.1.0.0/16"]
    resource_group_name = azurerm_resource_group.gwlb.name
    location      = azurerm_resource_group.gwlb.location
}

resource "azurerm_subnet" "web" {
    name          = "web-subnet"
    resource_group_name = azurerm_resource_group.gwlb.name
    virtual_network_name = azurerm_virtual_network.web.name
    address_prefixes = [cidrsubnet(azurerm_virtual_network.web.address_space[0], 8, 1)]
}
```

Web App Virtual Network and Subnets

The screenshot shows a web application interface for managing a virtual network named "web-net".

Header: **web-net | Subnets** (with a star icon) and three dots for more options.

Left Sidebar:

- Search bar
- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems

Settings Section:

- Address space
- Connected devices
- Subnets (highlighted with a gray background)

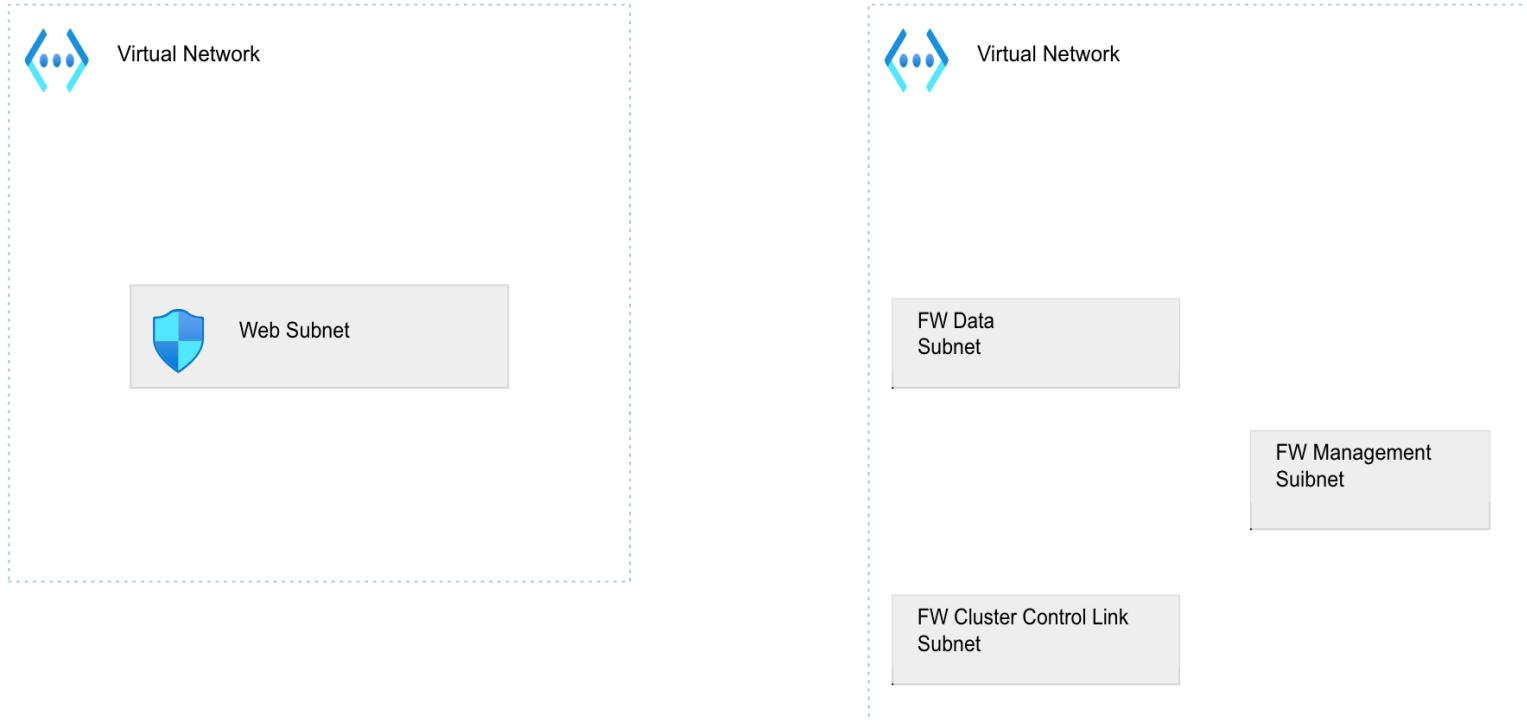
Top Action Bar:

- Search bar: Search subnets
- Buttons: + Subnet, + Gateway subnet, Refresh

Table View:

Name ↑↓	IPv4 ↑↓
web-subnet	10.1.1.0/24

Web App Virtual Network and Subnets



Web Facing Load Balancer in Web VNET

```
resource "azurerm_public_ip" "web_lb" {
  name          = "web-ip"
  location      = azurerm_resource_group.gwlb.location
  resource_group_name = azurerm_resource_group.gwlb.name
  allocation_method = "Static"
  sku           = "Standard"
}

resource "azurerm_lb" "web" {
  name          = "web-lb"
  location      = azurerm_resource_group.gwlb.location
  resource_group_name = azurerm_resource_group.gwlb.name
  sku           = "Standard"

  frontend_ip_configuration {
    name          = "web-lb-ip"
    public_ip_address_id = azurerm_public_ip.web_lb.id
    gateway_load_balancer_frontend_ip_configuration_id = azurerm_lb.fw.frontend_ip_configuration[0].id
  }
}
```

Web Facing Load Balancer in Web VNET

web-lb | Frontend IP configuration ☆ ...

Load balancer

<< + Add ⟳ Refresh ↗ Give feedback

Filter by name...

Name ↑	IP address ↑
web-lb-ip	23.100.31.160 (web-ip)

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Frontend IP configuration

Web Facing Load Balancer Configuration

```
resource "azurerm_lb_backend_address_pool" "web" {
    loadbalancer_id = azurerm_lb.web.id
    name           = "web-servers"
}

#...

resource "azurerm_network_interface_backend_address_pool_association" "web" {
    network_interface_id      = azurerm_network_interface.web.id
    ip_configuration_name     = "web-nic-ip"
    backend_address_pool_id   = azurerm_lb_backend_address_pool.web.id
}

resource "azurerm_lb_probe" "http_probe" {
    loadbalancer_id = azurerm_lb.web.id
    name           = "http-probe"
    protocol       = "Http"
    request_path   = "/"
    port           = 80
}

resource "azurerm_lb_rule" "web" {
    loadbalancer_id          = azurerm_lb.web.id
    name                     = "HTTP"
    protocol                 = "Tcp"
    frontend_port            = 80
    backend_port              = 80
    frontend_ip_configuration_name = "web-lb-ip"
    backend_address_pool_ids = [azurerm_lb_backend_address_pool.web.id]
    probe_id                 = azurerm_lb_probe.http_probe.id
    disable_outbound_snat = true
}

resource "azurerm_lb_outbound_rule" "web" {
    name           = "web-outbound"
    loadbalancer_id = azurerm_lb.web.id
    protocol       = "All"
    backend_address_pool_id = azurerm_lb_backend_address_pool.web.id
    allocated_outbound_ports = 512

    frontend_ip_configuration {
        name = "web-lb-ip"
    }
}
```

Web Facing Load Balancer Configuration

The screenshot shows two pages from the Azure portal related to a load balancer named "web-lb".

Top Page: Overview of the Load Balancer

- Resource group:** gwlb-rg
- Location:** East US
- Subscription:** sec-automation-azure
- Subscription ID:** 61b020af-0c48-4132-b2cf-2b162c4c42a2
- SKU:** Standard
- Backend pool:** web-servers (1 virtual machine)
- Load balancing rule:** HTTP (Tcp/80)
- Health probe:** http-probe (Http:80)
- NAT rules:** 0 inbound
- Tier:** Regional

Bottom Page: Backend pools

The page displays the configuration of the "Backend pools" for the load balancer.

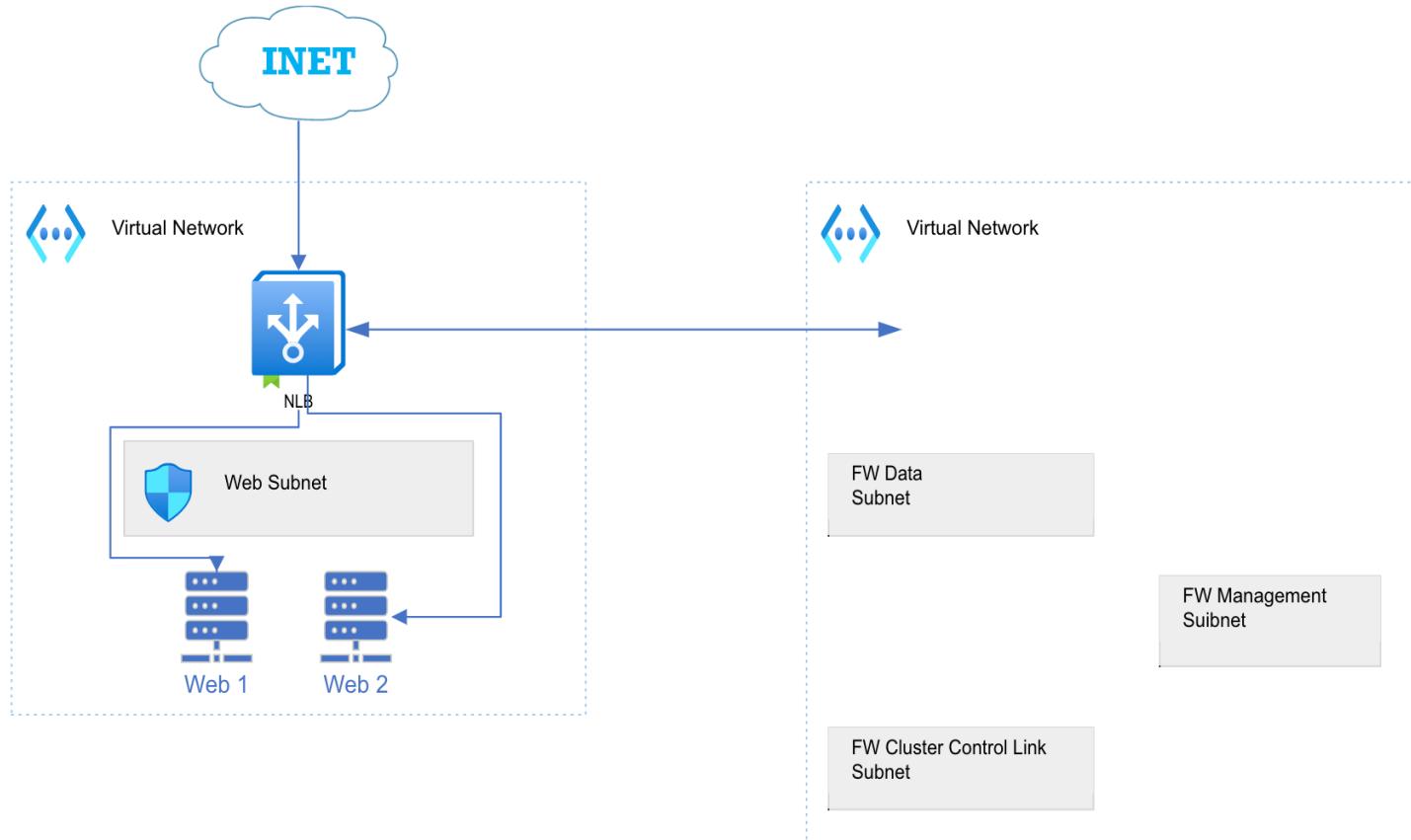
Overview:

The backend pool is a critical component of the load balancer. The backend pool defines the group of resources that will serve traffic for a given load-balancing rule. [Learn more.](#)

Table: Backend pools

Backend pool	Resource Name	IP address	Network interface
web-servers (1)	web	10.1.1.4	web-nic

Web Facing Load Balancer Configuration



Gateway Load Balancer Configuration

```
resource "azurerm_lb" "fw" {
    name          = "fw-lb"
    location      = azurerm_resource_group.gwlb.location
    resource_group_name = azurerm_resource_group.gwlb.name
    sku           = "Gateway"

    frontend_ip_configuration {
        name      = "fw-lb-ip"
        subnet_id = azurerm_subnet.fw_data.id
    }
}

resource "azurerm_lb_backend_address_pool" "fw" {
    loadbalancer_id = azurerm_lb.fw.id
    name           = "firewalls"
    tunnel_interface {
        identifier = 800
        type       = "Internal"
        protocol   = "VXLAN"
        port       = 10800
    }

    tunnel_interface {
        identifier = 801
        type       = "External"
        protocol   = "VXLAN"
        port       = 10801
    }
}
```

Gateway Load Balancer Configuration

fw-lb | Frontend IP configuration

Load balancer

Search



Add

Refresh

Give feedback

Overview

Activity log

Access control (IAM)

Tags

fw-lb | Backend pools

Load balancer

Search



Add Refresh

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Frontend IP configuration

Backend pools

Health probes

cisco Live!

Filter by name...

Name ↑↓

fw-lb-ip

IP address ↑↓

10.100.2.4

firewalls

fw-lb

Name *

Virtual network ⓘ

firewalls

gwlb-net

Backend Pool Configuration

NIC

IP address

Gateway load balancer configuration

Configuration settings on how the traffic is redirected to and from the gateway appliances.

Protocol ⓘ

VXLAN

Internal and External

Internal

External

10800

800

10801

801

The backend pool is a critical component of the load balancer. The backend pool defines the group of resources that will serve traffic for a given load-balancing rule. [Learn more.](#)

Backend pool

Resource Name

IP address

Network interface

Internal port * ⓘ

External port * ⓘ

External identifier * ⓘ

DEVNET-2150

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

30 30

Gateway Load Balancer Configuration

```
resource "azurerm_lb_backend_address_pool_address" "fw" {
  name          = "fw-lb-pool"
  backend_address_pool_id = azurerm_lb_backend_address_pool.fw.id
  virtual_network_id   = azurerm_virtual_network.gwlb.id
  ip_address         = azurerm_network_interface.fw_data.ip_configuration[0].private_ip_address
}

resource "azurerm_lb_probe" "tcp_12345" {
  loadbalancer_id = azurerm_lb.fw.id
  name           = "tcp-12345"
  protocol       = "Tcp"
  port           = 12345
}

resource "azurerm_lb_rule" "gwlb" {
  loadbalancer_id      = azurerm_lb.fw.id
  name                 = "All-Traffic"
  protocol             = "All"
  frontend_ip_configuration_name = "fw-lb-ip"
  frontend_port        = 0
  backend_port         = 0
  load_distribution    = "SourceIP"
  backend_address_pool_ids = [azurerm_lb_backend_address_pool.fw.id]
  probe_id             = azurerm_lb_probe.tcp_12345.id
}
```

Home > fw-lb | Health probes >

tcp-12345 ...

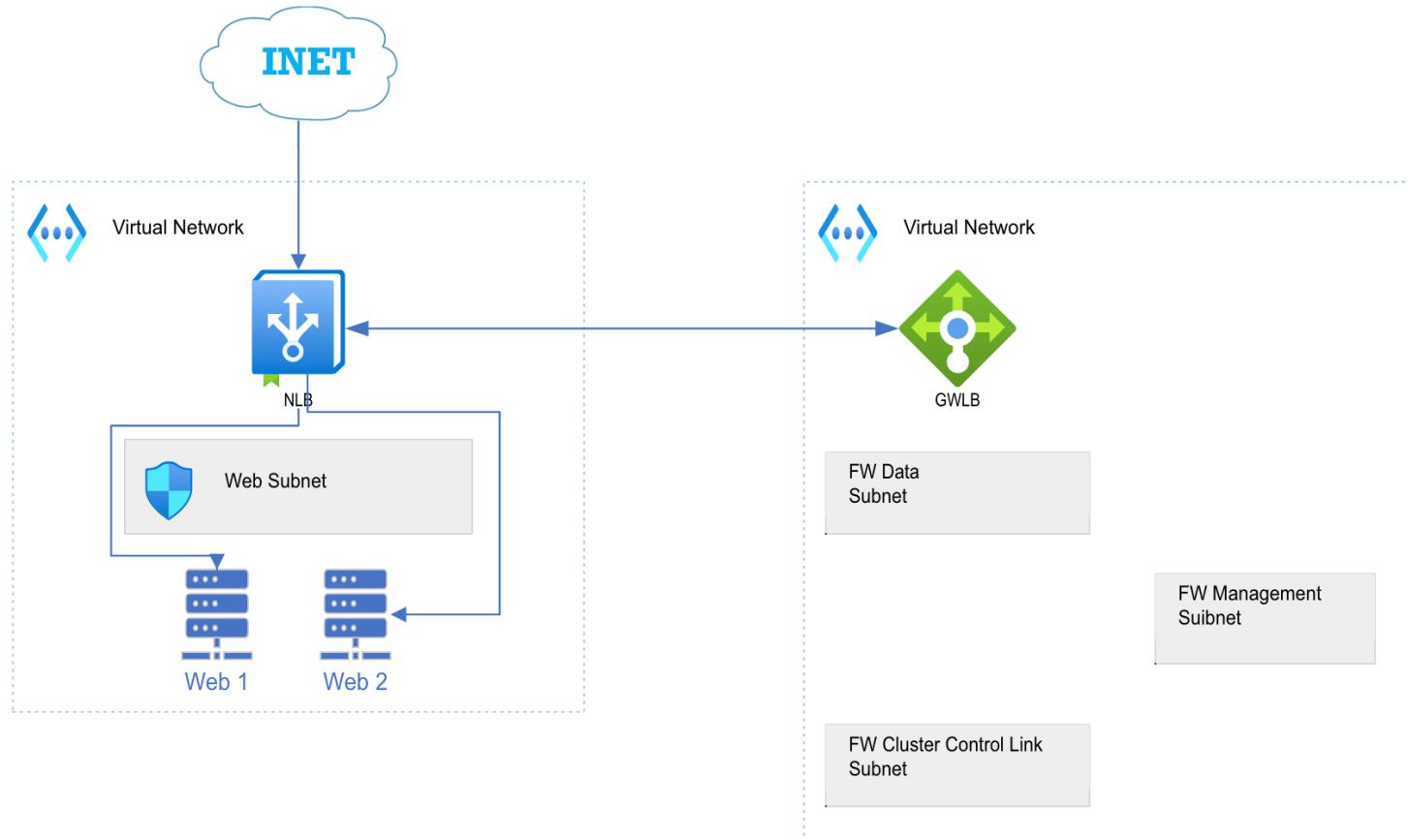
fw-lb

i Health probes are used to check the status of a backend pool instance. If a probe fails, then no new connections will be sent to that backend instance until it passes again.

i Unhealthy threshold, otherwise known as the property `numberOfUnsuccessfulProbes`. If the number of unsuccessful probes reaches this value, then the instance is marked as unhealthy immediately after 1 probe regardless of the property's configured interval.

Name *	<input type="text" value="tcp-12345"/>
Protocol *	<input type="text" value="TCP"/>
Port * i	<input type="text" value="12345"/>
Interval (seconds) * i	<input type="text" value="15"/>
Used by * i	<input type="text" value="All-Traffic"/>

Gateway Load Balancer Configuration



Cisco Secure Firewall (aka FTD)

```
resource "azurerm_public_ip" "ftd-pub-ip" {
    name          = "ftd-pub-ip"
    location      = azurerm_resource_group.gwlb.location
    resource_group_name = azurerm_resource_group.gwlb.name
    allocation_method = "Static"
    sku           = "Standard"
}

resource "azurerm_network_interface" "fw_management" {
    name          = "fw-management-nic"
    location      = azurerm_resource_group.gwlb.location
    resource_group_name = azurerm_resource_group.gwlb.name

    ip_configuration {
        name          = "fw-management-nic-ip"
        subnet_id     = azurerm_subnet.fw_management.id
        private_ip_address_allocation = "Dynamic"
        public_ip_address_id       = azurerm_public_ip.ftd-pub-ip.id
    }
}
```

Cisco Secure Firewall (aka FTD)

The image shows two screenshots of Azure resource management pages.

ftd-pub-ip (Public IP address)

Overview

Resource group (move) : [gwlb-rg](#)

Location (move) : East US

Subscription (move) : [sec-automation-azure](#)

Subscription ID : 61b020af-0c48-4132-b2cf-2b162c4c42a2

SKU : Standard

Tier : Regional

IP address : 23.101.130.227

DNS name : -

Associated to : [fw-management-nic](#)

Virtual machine : [ftd-azure](#)

Routing preference : Microsoft network

fw-management-nic (Network interface)

Overview

Resource group (move) : [gwlb-rg](#)

Location (move) : East US

Subscription (move) : [sec-automation-azure](#)

Subscription ID : 61b020af-0c48-4132-b2cf-2b162c4c42a2

Accelerated networking : Disabled

Virtual network/subnet : [gwlb-net/fw-management](#)

Private IPv4 address : 10.100.1.6

Public IPv4 address : 23.101.130.227 (ftd-pub-ip)

Private IPv6 address : -

Public IPv6 address : -

Attached to : [ftd-azure \(Virtual machine\)](#)
[ftd-ssh-ns \(Network security group\)](#)

Type : Regular

Cisco Secure Firewall (aka FTD)

```
resource "azurerm_network_interface" "fw_data" {
    name          = "fw-data-nic"
    location      = azurerm_resource_group.gwlb.location
    resource_group_name = azurerm_resource_group.gwlb.name

    ip_configuration {
        name          = "fw-data-nic-ip"
        subnet_id     = azurerm_subnet.fw_data.id
        private_ip_address_allocation = "Dynamic"
    }
}

resource "azurerm_network_interface" "fw_ccl" {
    name          = "fw-ccl-nic"
    location      = azurerm_resource_group.gwlb.location
    resource_group_name = azurerm_resource_group.gwlb.name

    ip_configuration {
        name          = "fw-cl-nic-ip"
        subnet_id     = azurerm_subnet.fw_ccl.id
        private_ip_address_allocation = "Dynamic"
    }
}
```

Cisco Secure Firewall (aka FTD)

fw-data-nic ⭐ ...

Network interface

Search Overview Activity log Access control (IAM) Tags

Resource group ([move](#)) : [gwlb-rg](#) Private IPv4 address : 10.100.2.5
Location ([move](#)) : East US Public IPv4 address : -
Subscription ([move](#)) : [sec-automation-azure](#) Private IPv6 address : -
Subscription ID : 61b020af-0c48-4132-b2cf-2b162c4c42a2 Public IPv6 address : -
Accelerated networking : Disabled Attached to : [ftd-azure \(Virtual machine\)](#)
Virtual network/subnet : [gwlb-net/fw-data](#) Type : Regular

fw-ccl-nic ⭐ ...

Network interface

Search Overview Activity log Access control (IAM) Tags

Resource group ([move](#)) : [gwlb-rg](#) Private IPv4 address : 10.100.3.4
Location ([move](#)) : East US Public IPv4 address : -
Subscription ([move](#)) : [sec-automation-azure](#) Private IPv6 address : -
Subscription ID : 61b020af-0c48-4132-b2cf-2b162c4c42a2 Public IPv6 address : -
Accelerated networking : Disabled Attached to : [ftd-azure \(Virtual machine\)](#)
Virtual network/subnet : [gwlb-net/fw-ccl](#) Type : Regular

Cisco Defense Orchestrator

```
# Create default Access Control Policy
resource "fmc_access_policies" "access_policy" {
    name          = "${var.name}-Access-Policy"
    default_action = "block"
}

resource "cdo_ftd_device" "ftd" {
    access_policy_name = fmc_access_policies.access_policy.name
    licenses          = ["BASE", "MALWARE", "URLFilter", "THREAT"]
    name              = "ftd-azure"
    virtual           = true
    performance_tier = "FTDv30"
}
```

Cisco Defense Orchestrator

The screenshot shows the Cisco Defense Orchestrator web interface. The top navigation bar includes the Cisco logo, a search bar, and various system status indicators. The main content area displays a list of Access Control Policies on the left and a detailed view of a selected policy on the right.

Left Panel (Access Control Policy List):

- Default Access Control Policy
- Default Access Control Policy with default action block
- ftd_registration-Access-Policy
- FTDv-Access-Policy
- home-acp
- home network
- ice-cube-Access-Policy
- sock-shop-Access-Policy
- Test_Policy

The "ice-cube-Access-Policy" row is highlighted with a blue border.

Right Panel (Selected Policy View):

Dashboard: Multicloud Defense (New)

Inventory: Configuration

Policies: Targeting 1 devices (*Out-of-date on 1 targeted devices*)

Devices: ftd-azure (FTD Cluster, 1 device)
Configuration Status: Synced
Connectivity Status: Online

Search Results: Displaying 1 of 1 results

Name	Configuration Status	Connectivity Status
ftd-azure	Synced	Online

FTD Instance Configuration

```
resource "azurerm_linux_virtual_machine" "ftd" {
    name                  = "ftd-azure"
    computer_name         = "ftd-azure"
    location              = azurerm_resource_group.gwlb.location
    resource_group_name   = azurerm_resource_group.gwlb.name
    network_interface_ids = [
        azurerm_network_interface.fw_management.id,
        azurerm_network_interface.fw_diagnostic.id,
        azurerm_network_interface.fw_data.id,
        azurerm_network_interface.fw_ccl.id
    ]
    size                  = "Standard_D3_v2"
    admin_username         = "azadmin"
    admin_password         = var.admin_password
    disable_password_authentication = false
}
```

FTD Instance Configuration

```
custom_data = base64encode(jsonencode(
{
    "AdminPassword": var.admin_password,
    "Hostname": "ftd-azure",
    "FirewallMode": "Routed",
    "ManageLocally": "No",
    "FmcIp": "${var.cdFMC}",
    "FmcRegKey": "${cdo_ftd_device_ftd.reg_key}",
    "FmcNatId": "${cdo_ftd_device_ftd.nat_id}",
    "Cluster": {
        "CclSubnetRange": "${cidrhost(azurerm_subnet.fw_ccl.address_prefixes[0],1)} ${cidrhost(azurerm_subnet.fw_ccl.address_prefixes[0],32)}",
        "ClusterGroupName": "ftd-azure",
        "HealthProbePort": "12345",
        "GatewayLoadBalancerIP": "${azurerm_lb.fw.frontend_ip_configuration[0].private_ip_address}",
        "EncapsulationType": "vxlan",
        "InternalPort": "10800",
        "ExternalPort": "10801",
        "InternalSegId": "800",
        "ExternalSegId": "801"
    }
})
```

FTD Instance Configuration

ftd-azure | Networking

Virtual machine

Search

Feedback Attach network interface Detach network interface

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Networking

fw-management-nic fw-diagnostic-nic fw-data-nic fw-ccl-nic

IP configuration fw-management-nic-ip (Primary)

Network Interface: fw-management-nic

Virtual network/subnet: gwlb-net/fw-management

Effective security rules

NIC Public IP: 23.101.130.227

Troubleshoot VM connection issues

NIC Private IP: 10.100.1.6

Topology

Accelerated networking: Disabled

FTD Instance Configuration

Defense Orchestrator
FMC / Devices / Secure Firewall Interfaces

Analysis Policies Devices

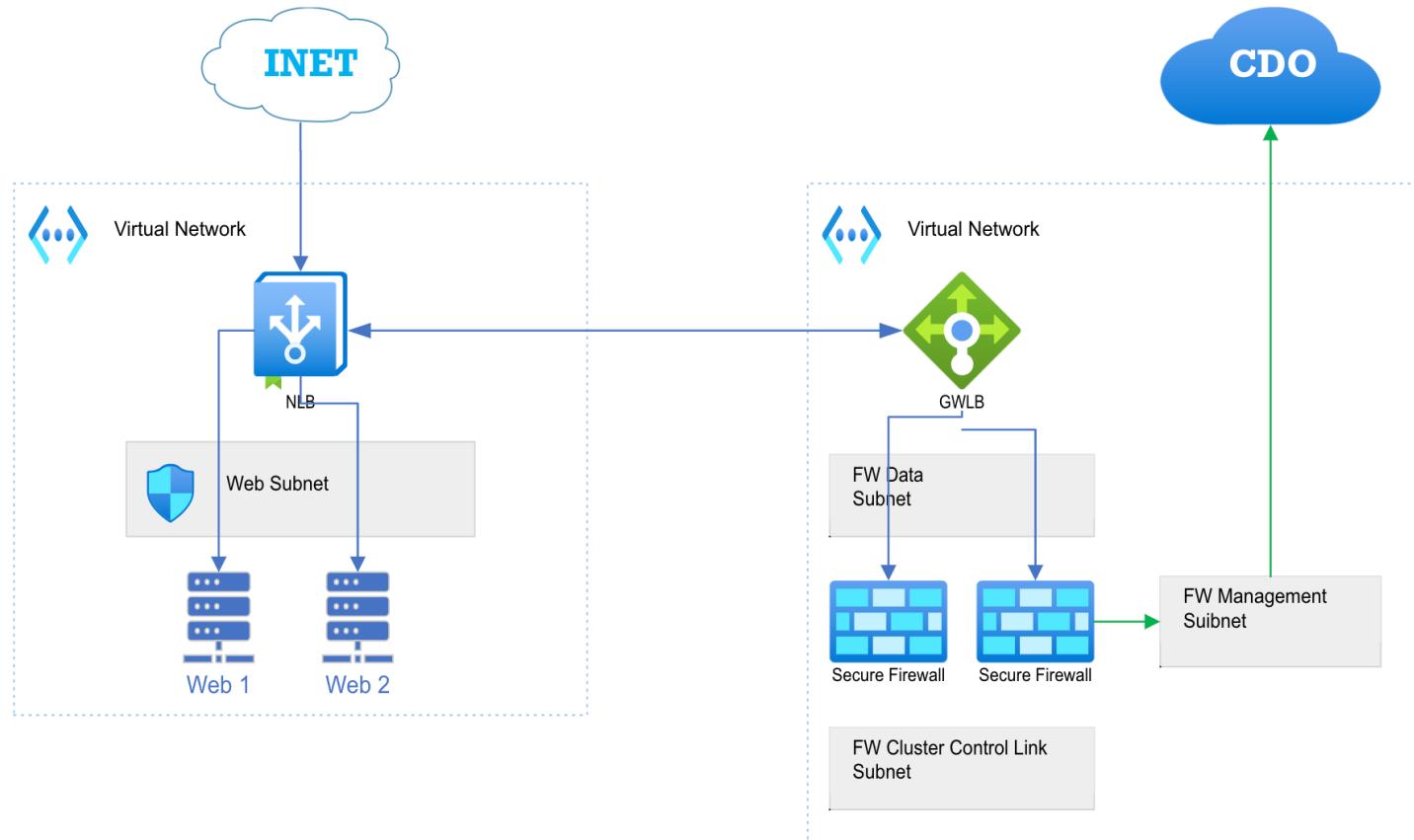
ftd-azure

Cisco Firepower Threat Defense for Azure

Cluster Device Routing Interfaces Inline Sets DHCP VTEP

	Interface	Logical Name	Type	Security Zones
	Diagnostic0/0	management	Physical	
	GigabitEthernet0/0	vxlan_tunnel	Physical	
	GigabitEthernet0/1	ccl_link	Physical	
	vni1		VNIInt...	
	vni2	GWLB-backend...	VNIInt...	

FTD Instance Configuration



Ansible

cisco *Live!*

Ansible Collections

- Cisco FMC
 - <https://github.com/CiscoDevNet/FMCAnsibleCisco>
- Cisco CDO
 - <https://github.com/CiscoDevNet/ansible-cisco-cdo/tree/main/docs>

Cisco Secure Firewall Management Center (FMC) Ansible Collection

An Ansible Collection that automates configuration management and execution of operational tasks on Cisco Secure Firewall Management Centre (FMC) devices using FMC REST API.

This module has been tested against the following ansible versions: 2.9.17, 2.10.5 This module has been tested against the following cisco Secure Firewall Management Center versions: 7.0, 7.1, 7.2, 7.3

Included Content

The collection contains one Ansible module:

- [fmc_configuration.py](#) - manages device configuration via REST API. The module configures virtual and physical devices by sending HTTPS calls formatted according to the REST API specification.

Installing this collection

You can install the Cisco DCNM collection with the Ansible Galaxy CLI:

```
ansible-galaxy collection install cisco.fmcansible
```

Create Inventory File from Terraform Resources

```
resource "local_file" "host_file" {
  content      = <<-EOT
  ---
  all:
    hosts:
      cdFMC:
        ansible_host: ${var.cdFMC}
        ansible_network_os: cisco.fmcansible.fmc
        ansible_httpapi_port: 443
        ansible_httpapi_use_ssl: True
        ansible_httpapi_validate_certs: True
        web_lb_public_ip: ${azurerm_public_ip.web_lb.ip_address}
    EOT
  ● filename = "${path.module}/hosts.yaml"
}
```

Inventory File Created (YAML)

```
---
all:
  hosts:
    cdFMC:
      ansible_host: cisco-edmcnich.app.us.cdo.cisco.com
      ansible_network_os: cisco.fmcansible.fmc
      ansible_httpapi_port: 443
      ansible_httpapi_use_ssl: True
      ansible_httpapi_validate_certs: True
      web_lb_public_ip: 23.100.31.160
```

Fetch FMC Domain and FTD Devices

```
# Fetch Data from FMC
- name: Get Domain UUID
  cisco.fmcansible.fmc_configuration:
    operation: getAllDomain
    register_as: domain
- name: Get Devices
  cisco.fmcansible.fmc_configuration:
    operation: getAllDevice
    path_params:
      domainUUID: '{{ domain[0].uuid }}'
    filters:
      name: ftd-azure
    register_as: device_list
  until: device_list is defined
  retries: 60
  delay: 5
```

Fetch HTTP Object and Access Policy

```
- name: Get Port Object HTTP
  cisco.fmcansible.fmc_configuration:
    operation: getAllProtocolPortObject
    path_params:
      domainUUID: '{{ domain[0].uuid }}'
    filters:
      name: HTTP
    register_as: http
```

```
- name: Get Access Policy
  cisco.fmcansible.fmc_configuration:
    operation: getAllAccessPolicy
    path_params:
      domainUUID: '{{ domain[0].uuid }}'
    filters:
      name: ftd-azure-Access-Policy
    register_as: accesspolicy
```

Create Host Object

```
- name: Web Load Balancer
  cisco.fmcansible.fmc_configuration:
    operation: upsertHostObject
    data:
      name: web_lb_public_ip
      value: '{{ web_lb_public_ip }}'
      type: Host
    path_params:
      domainUUID: '{{ domain[0].uuid }}'
    register_as: web_lb_public_ip
```

Outbound Rule

```
- name: Access Rule 1
  cisco.fmcansible.fmc_configuration:
    operation: upsertAccessRule
    data:
      name: Permit Outbound
      type: accessrule
      action: ALLOW
      section: mandatory
      enabled: true
      sendEventsToFMC: true
      logBegin: true
      logEnd: true
      sourceNetworks:
        objects:
          - id: '{{ web_lb_public_ip.id }}'
            name: '{{ web_lb_public_ip.name }}'
            type: '{{ web_lb_public_ip.type }}'
      newComments:
        - 'Outbound Traffic'
    path_params:
      section: 'Mandatory'
      containerUUID: '{{ accesspolicy[0].id }}'
      domainUUID: '{{ domain[0].uuid }}'
    register_as: accessrule1
```

Inbound Rule

CISCO Live!

```
- name: Access Rule 2
cisco.fmcansible.fmc_configuration:
    operation: upsertAccessRule
    data:
        name: Access to Web Server
        type: accessrule
        action: ALLOW
        section: mandatory
        enabled: true
        sendEventsToFMC: true
        logBegin: true
        logEnd: true
        destinationNetworks:
            objects:
                - id: '{{ web_lb_public_ip.id }}'
                  name: '{{ web_lb_public_ip.name }}'
                  type: '{{ web_lb_public_ip.type }}'
        destinationPorts:
            objects:
                - id: '{{ HTTP.id }}'
                  name: '{{ HTTP.name }}'
                  type: '{{ HTTP.type }}'
        newComments:
            - 'Web Server'
        path_params:
            section: 'Mandatory'
            containerUUID: '{{ accesspolicy[0].id }}'
            domainUUID: '{{ domain[0].uuid }}'
            register_as: accessrule2
```

Access Policy

Defense Orchestrator FMC / Policies / Access Control / Policy Editor

Analysis Policies Devices Objects Integration

Return Home Deploy 12 ? edmcnich@cisco.com SECURE

Return to Access Control Policy Management ice-cube-Access-Policy

Packets → Prefilter Rules → Decryption → Security Intelligence → Identity → Access Control More

Switch to Legacy UI Analyze Discard Save Targeted: 1 device

Type to search Total 3 rules Add Category Add Rule

Name	Action	Source			Destination			Applications	Users
		Zones	Networks	Ports	Zones	Networks	Ports		
Mandatory (-)									
1 Permit Outbound	Allow	Any	web_lb_public_ip	Any	Any	Any	Any	Any	Any
2 Inbound to Web Server	Allow	Any	Any	Any	Any	web_lb_public_ip	HTTP	Any	Any
Default (-)									
3 Deny_Any	Block	Any	Any	Any	Any	Any	Any	Any	Any

Conclusion

cisco *Live!*

Conclusion

- Provisioned Cisco Secure Firewall into Microsoft Azure using Terraform
 - Deployed the Firewall cluster into a Service VNET behind an Azure Gateway Load Balancer
 - Deployed web application into Web App VNET behind a Standard Load Balancer
 - Linked the Standard Load Balancer to the Gateway Load Balancer to service chain Ingress/Egress for the Web App to the Secure Firewall Cluster
- Configured the Secure Firewall Policy using Ansible
 - Created dynamic object using Terraform resources
 - Created Policy rules to allow access to and from the Web App

Fill out your session surveys!



Attendees who fill out a minimum of four session surveys and the overall event survey will get **Cisco Live-branded t-shirt** (while supplies last)!

All surveys can be taken in the Cisco Events Mobile App or by logging in to the Session Catalog and clicking the "Attendee Dashboard" at <https://www.ciscolive.com/emea/learn/sessions/session-catalog.html>





The bridge to possible

Thank you

cisco Live!



cisco *Live!*

Let's go