

CISCO *Live!*

GO BEYOND

#CiscoLiveAPJC



Automating Cisco Secure Firewall in Public Cloud

Using Terraform

Siddhu Warriar, Principal Engineer
Ed McNicholas, Cybersecurity Architect
DEVNET-1023

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

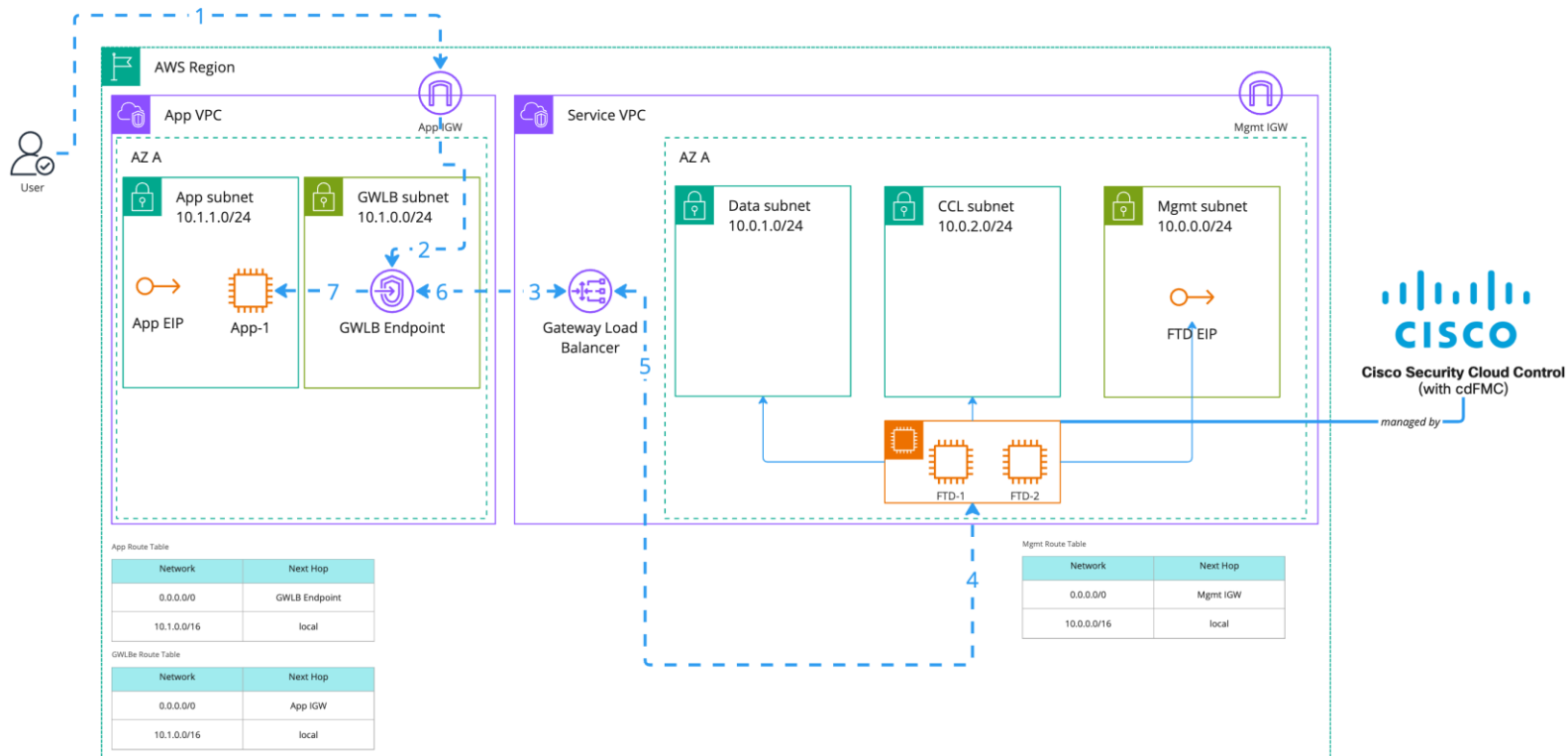
Webex spaces will be moderated by the speaker until November 15, 2024.



<https://ciscolive.ciscoevents.com/ciscolivebot/#DEVNET-1023>



Deployment Topology



There's a learning lab!

The screenshot shows the Cisco DevNet Learning Labs Center interface. The top navigation bar includes the Cisco DevNet logo, the text 'Learning Labs Center', and links for 'SIGN UP FREE' and 'LOG IN'. The main header area displays the lab title 'Secure Firewall Automation AWS GWLB Cisco Live' on the left and a hamburger menu icon followed by the same title on the right. A left sidebar contains a table of contents with 10 items: 'Introduction', 'Setting up the Terraform Environment', 'Service VPC', 'Application VPC', 'Security Groups', 'Cisco Secure Firewall', 'Gateway Load Balancer', 'Routing', 'Application' (highlighted in blue), 'Access Control Policy', and 'Firewall Registration'. The main content area is titled 'Introduction' and contains a paragraph explaining the lab's purpose: using Terraform to provision a Secure Firewall cluster into a Service VPC behind an AWS Gateway Load Balancer (GWLB). It mentions Cisco Secure Firewall Threat Defense and the use of the AWS Gateway Load Balancer (GWLB) to drive simple, agile, and efficient security in the cloud. The text states that this integration simplifies the insertion of Cisco Secure Firewall in AWS with Geneve protocol (RFC 8926) encapsulation, making architectures more scalable by removing the need for source network address translation (SNAT) in the traffic path. Below the introduction is a 'Lab Overview' section, which begins with a paragraph stating that the following diagram shows a scalable architecture for protecting ingress traffic with Cisco Secure Firewall and AWS Gateway Load Balancer. The text continues, recommending the creation of a service VPC with an AWS Gateway Load Balancer and Cisco Secure Firewall virtual appliances in the backend pool of the gateway load balancer. It notes that gateway load balancers talk to these firewalls using Geneve encapsulation, eliminating the need for SNAT, since the embedded virtual network interface (VNI) information. A 'Start Learning' button is located at the bottom right of the lab overview section. On the far right, there is a dark-themed panel with a terminal window showing some code, a 'Code as you learn!' prompt, and a message to 'Continue the learning lab to unlock the full interactive experience.'

<http://cs.co/fw-gwlb>

Basics: Terraform Terminology

Terraform Providers

```
provider "cdo" {  
  base_url = var.base_url  
  api_token = var.cdo_api_token  
}
```



```
provider "aws" {  
  region = var.aws_region  
}
```



```
provider "fmc" {  
  fmc_host      = data.cdo_cdfmc.current.hostname  
  is_cdfmc      = true  
  cdo_token     = var.cdo_api_token  
  cdfmc_domain_uuid = data.cdo_cdfmc.current.domain_uuid  
}
```



```
provider "vsphere" {  
  user      = var.vsphere_username  
  password  = var.vsphere_password  
  vsphere_server = var.vsphere_server  
  
  # if you have a self-signed cert  
  allow_unverified_ssl = var.allow_unverified_ssl  
}
```

Terraform Resources

```
resource "cdo_ftd_device" "ftd-1" {  
  name           = local.ftd_name_1  
  access_policy_name = "Default Access Control Policy"  
  performance_tier = "FTDv5"  
  virtual        = true  
  licenses       = ["BASE"]  
}
```



```
resource "fmc_access_policies" "fmc_access_policy" {  
  name           = "${local.ftd_name}-Terraform Access Policy"  
  default_action = "block"  
}
```



A resource is a component of your
Terraform-managed infrastructure.

vmware



```
resource "aws_route_table" "private_route_tables" {  
  count = var.num_az_zones  
  vpc_id = aws_vpc.vpc.id  
  
  Damir Suleymanov, 2 years ago | 3 authors (Damir Suleymanov and others)  
  tags = {  
    Name = "${var.base_name}-private-route-table-${lookup(var.az_subname_map, count.index)}"  
  }  
}
```

```
resource "vsphere_virtual_machine" "vm" {  
  name           = var.fqdn  
  resource_pool_id = var.vm_datastore_map["${var.vm_cluster_host}"]  
  datastore_id   = data.vsphere_datastore.datastore.id  
  folder         = var.vm_folder  
  wait_for_guest_net_timeout = false  
  
  num_cpus = 4  
  memory   = 8192  
  guest_id = data.vsphere_virtual_machine.template.guest_id  
  
  scsi_type = data.vsphere_virtual_machine.template.scsi_type  
  
  Damir Suleymanov, 4 years ago | 2 authors (Damir Suleymanov and others)  
  disk {  
    label           = "disk0"  
    size            = data.vsphere_virtual_machine.template.disks[0].size  
    eagerly_scrub   = data.vsphere_virtual_machine.template.disks[0].eagerly_scrub  
    thin_provisioned = data.vsphere_virtual_machine.template.disks[0].thin_provisioned  
  }  
}
```

cisco *Live!*

Terraform Data Sources

```
data "cdo_tenant" "current" {}
```



```
data "aws_region" "current" {}
```

```
data "fmc_access_policies" "acp" {  
  name = "FTD ACP"  
}
```



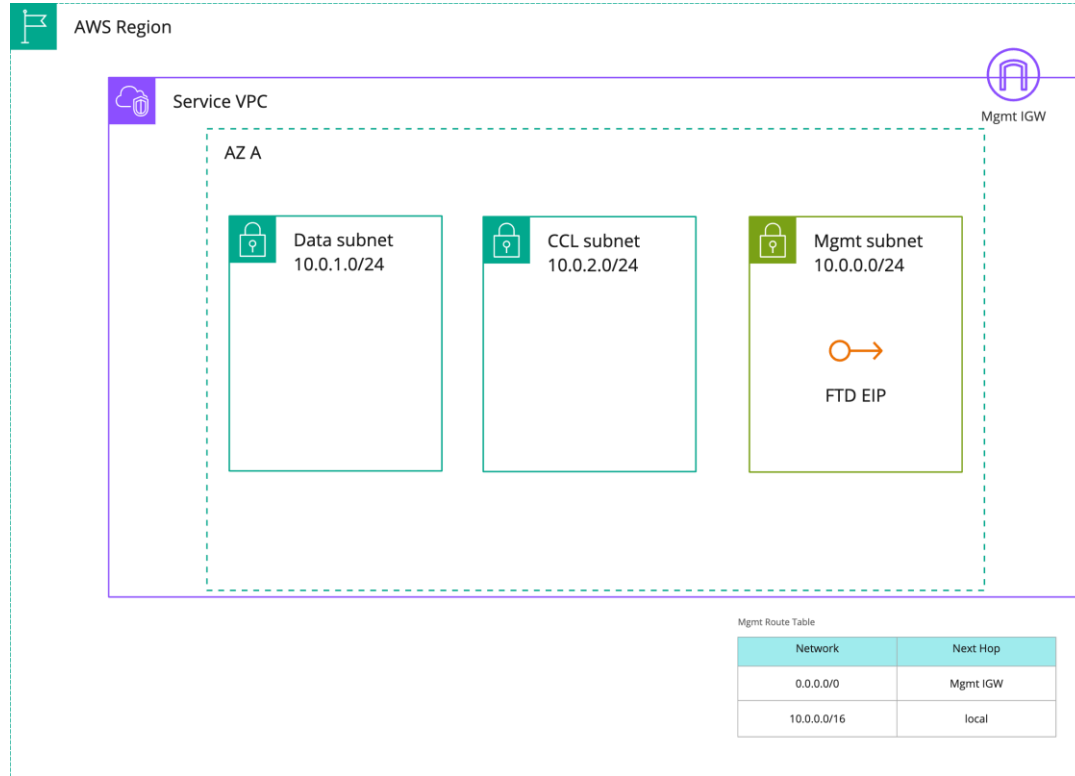
A data source is a read-only view into infrastructure components defined outside your Terraform code.

vmware

```
data "vsphere_datacenter" "dc" {  
  name = var.datacenter  
}
```

Create the Service VPC

Topology



Terraform: Defining the VPC

You, 2 minutes ago | 1 author (You)

Service VPC

You, 2 minutes ago | 1 author (You)

```
resource "aws_vpc" "srvc_vpc" {  
  cidr_block           = var.srvc_cidr  
  enable_dns_support   = true  
  enable_dns_hostnames = true  
  instance_tenancy     = "default" Not committed yet  
  
  tags = {  
    Name = "${var.env_name} Service VPC"  
  }  
}
```

Terraform: Defining the Subnets

Service Subnets

You, 3 minutes ago | 1 author (You)

```
resource "aws_subnet" "mgmt_subnet" {  
  vpc_id            = aws_vpc.srvc_vpc.id  
  cidr_block        = var.mgmt_subnet  
  availability_zone = var.aws_az  
  tags = {  
    Name = "${var.env_name} Service Mgmt Subnet"  
  }  
}
```

You, 3 minutes ago | 1 author (You)

```
resource "aws_subnet" "data_subnet" {  
  vpc_id            = aws_vpc.srvc_vpc.id  
  cidr_block        = var.data_subnet  
  availability_zone = var.aws_az  
  tags = {  
    Name = "${var.env_name} Service Data Subnet"  
  }  
}
```

```
resource "aws_subnet" "ccl_subnet" {  
  vpc_id            = aws_vpc.srvc_vpc.id  
  cidr_block        = var.ccl_subnet  
  availability_zone = var.aws_az  
  tags = {  
    Name = "${var.env_name} Service CCL Subnet"  
  }  
}
```

Terraform: Defining the Internet Gateway

```
# Service Mgmt IGW
You, 4 minutes ago | 1 author (You)
resource "aws_internet_gateway" "mgmt_igw" {
  vpc_id = aws_vpc.srvc_vpc.id
  You, 4 minutes ago | 1 author (You)
  tags = {
    Name = "${var.env_name} Service Mgmt-IGW"
  }
}
```

Terraform: Defining the Routes

```
# Mgmt Route Table
You, 5 minutes ago | 1 author (You)
resource "aws_route_table" "mgmt_route_table" {
  vpc_id = aws_vpc.srvc_vpc.id
  You, 5 minutes ago | 1 author (You)
  tags = {
    Name = "${var.env_name} Service Mgmt Route Table"
  }
}

# Mgmt Default Route Routes
You, 5 minutes ago | 1 author (You)
resource "aws_route" "mgmt_default_route" {
  depends_on      = [aws_internet_gateway.mgmt_igw]
  route_table_id  = aws_route_table.mgmt_route_table.id
  destination_cidr_block = "0.0.0.0/0"
  gateway_id      = aws_internet_gateway.mgmt_igw.id
}

# Mgmt Route Associations
You, 5 minutes ago | 1 author (You)
resource "aws_route_table_association" "mgmt_association" {
  subnet_id      = aws_subnet.mgmt_subnet.id
  route_table_id = aws_route_table.mgmt_route_table.id
}
```

Terraform: Apply Changes

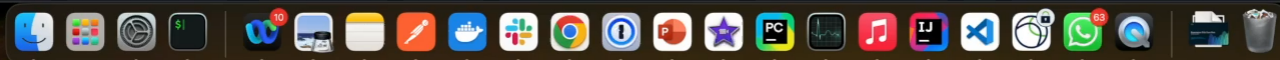
- 1 Open your Terminal
- 2 Type terraform init and press the Enter key
- 3 Type terraform apply and press the Enter key

tmuxinator start mac

~/w/cdo/cl/secure-firewall-automation/aw/gwlb-v2 main !4 71 >

9s 15:35:43

cdo-mac siwarrie 7:0 <servicesZ 4:Terraform 5:Infra 6:Deployments 7:clapj24* 8:cdo-public-api-docsZ 9:zsh- 15:35:47 10-Nov-24 Siddhus-MacBook-Pro-2.local



What it looks like on the AWS Console

The screenshot displays the AWS Management Console interface, specifically the VPC dashboard. The left sidebar shows the navigation menu with options like EC2 Global View, Virtual private cloud, and various networking resources. The main content area is divided into three panels: 'Your VPCs', 'Internet gateways', and 'Subnets'.

Panel 1: Your VPCs (1/1)

Search: service X Clear filters

☒	Name	VPC ID	State	IPv4 CIDR
☒	gwlb-test6 Service VPC	vpc-07de556bdbadab8a0	Available	10.0.0.0/16

vpc-07de556bdbadab8a0 / gwlb-test6 Service VPC

Details | Resource map | CIDRs | Flow logs | Tags | Integrations

Details

Property	Value
VPC ID	vpc-07de556bdbadab8a0
State	Available
DNS hostnames	Enabled
DHCP option set	dopt-06ceba3bfc43f3b74
Tenancy	Default
Default VPC	No
IPv4 CIDR	10.0.0.0/16
Main route table	rtb-07ac2749190fa628d
IPv6 pool	-

Panel 2: Internet gateways (1/1)

Search: service X Clear filters

☒	Name	Internet gateway ID	State	VPC ID
☒	gwlb-test6 Service Mgmt-IGW	igw-0b411380607e8bd51	Attached	vpc-07de556bdbadab8a0 gwlb-test6...

igw-0b411380607e8bd51 / gwlb-test6 Service Mgmt-IGW

Details | Tags

Details

Property	Value
Internet gateway ID	igw-0b411380607e8bd51
State	Attached
VPC ID	vpc-07de556bdbadab8a0 gwlb-test6 Service VPC

Panel 3: Subnets (3)

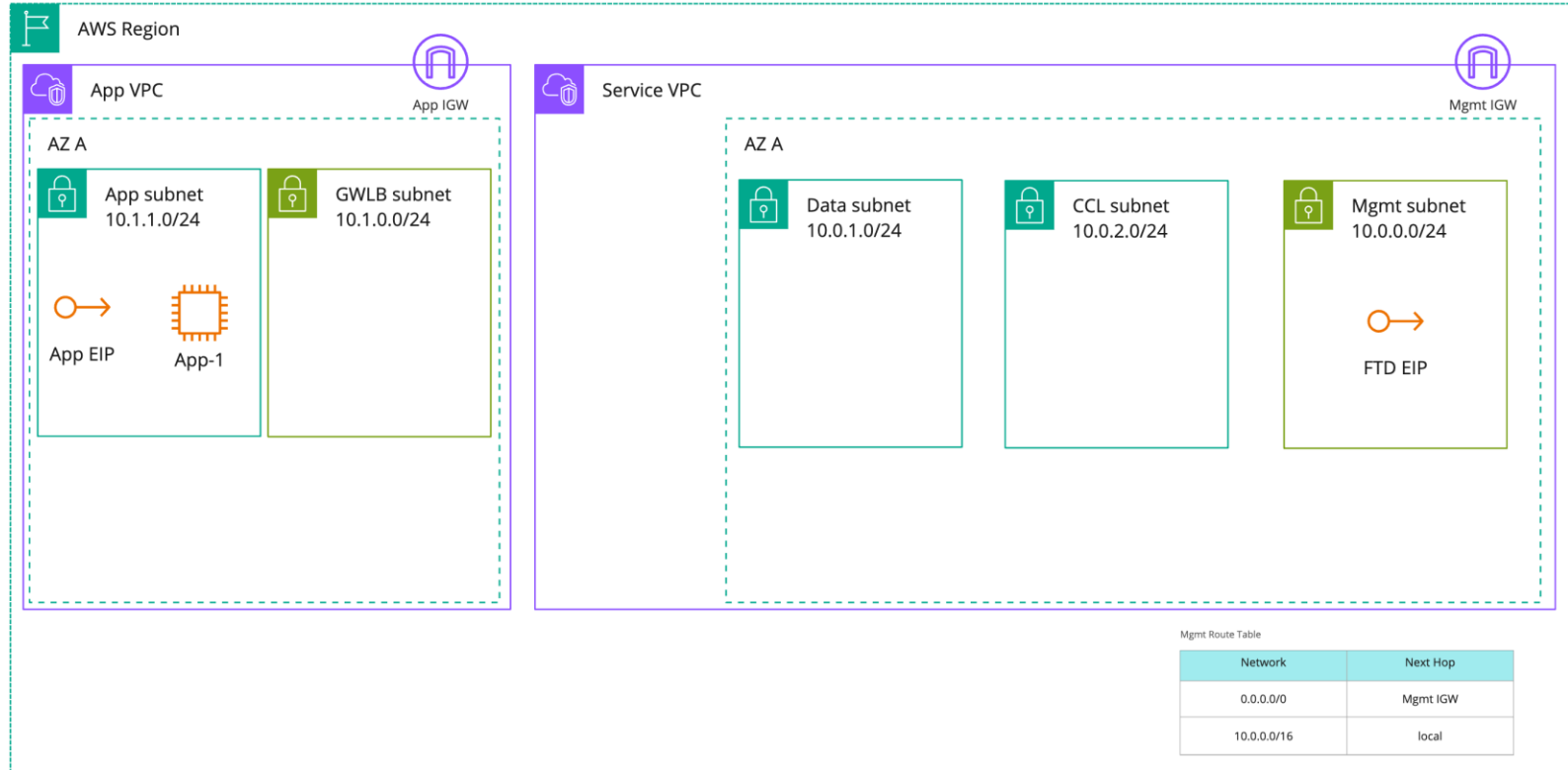
Search: Find resources by attribute or tag

service X Clear filters

☐	Name	Subnet ID	State	VPC	IPv4 CIDR
☐	gwlb-test6 Service Mgmt Subnet	subnet-011bccb8fbc6fdbf8	Available	vpc-07de556bdbadab8a0 gwlb-test6 Service VPC	10.0.0.0/24
☐	gwlb-test6 Service CCL Subnet	subnet-0ba64defd4c0d32c1	Available	vpc-07de556bdbadab8a0 gwlb-test6 Service VPC	10.0.2.0/24
☐	gwlb-test6 Service Data Subnet	subnet-0bfa74422498815a9	Available	vpc-07de556bdbadab8a0 gwlb-test6 Service VPC	10.0.1.0/24

Create the App VPC

Application VPC



Terraform: Defining the VPC

```
# App VPC
```

```
You, 4 minutes ago | 1 author (You)
```

```
resource "aws_vpc" "app_vpc" {  
  cidr_block           = var.app_cidr  
  enable_dns_support   = true  
  enable_dns_hostnames = true  
  instance_tenancy     = "default"  
  You, 4 minutes ago | 1 author (You)  
  tags = {  
    Name = "${var.env_name }-App-VPC"  
  }  
}
```

Terraform: Defining the Subnets

```
resource "aws_subnet" "gwlbe_subnet" {  
  vpc_id          = aws_vpc.app_vpc.id  
  cidr_block      = var.gwlbe_subnet  
  availability_zone = var.aws_az  
  You, 4 minutes ago | 1 author (You)  
  tags = {  
    Name = "${var.env_name } GWLBe Subnet"  
  }  
}
```

```
resource "aws_subnet" "app_subnet" {  
  vpc_id          = aws_vpc.app_vpc.id  
  cidr_block      = var.app_subnet  
  availability_zone = var.aws_az  
  You, 4 minutes ago | 1 author (You)  
  tags = {  
    Name = "${var.env_name } App Subnet"  
  }  
}
```

Terraform: Defining the Internet Gateway

```
resource "aws_internet_gateway" "app_igw" {  
  vpc_id = aws_vpc.app_vpc.id  
  You, 5 minutes ago | 1 author (You)  
  tags = {  
    Name = "${var.env_name } IGW"  
  }  
}
```

Terraform: Apply Changes

- 1 Open your Terminal
- 2 Type terraform apply and press the Enter key

~/W/cdo/cl/secure-firewall-automation/aw/gwlb-v2 main !5 72 > INT 15:50:43



What it looks like on the AWS Console

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peering connections

Security

Network ACLs

Security groups

VPC dashboard

EC2 Global View

Filter by VPC:

Select a VPC

Your VPCs (1/2) Info

Search

gwlb

Clear filters

	Name	VPC ID	State	IPv4 CIDR	IPv6
☐	gwlb-test6 Service VPC	vpc-07de556bdbadab8a0	Available	10.0.0.0/16	-
☒	gwlb-test6-App-VPC	vpc-0a0564082a96d340e	Available	10.1.0.0/16	-

vpc-0a0564082a96d340e / gwlb-test6-App-VPC

Details | Resource map | CIDRs | Flow logs | Tags | Integrations

Details

VPC ID

vpc-0a0564082a96d340e

State

Available

DNS hostnames

Enabled

Tenancy

Default

DHCP option set

dopt-06ceb3bfc4f3f374

Main route table

rtb-040f6852fcc4a093

Default VPC

No

IPv4 CIDR

10.1.0.0/16

IPv6 pool

-

Network Address Usage metrics

Disabled

Route 53 Resolver DNS Firewall rule groups

-

Owner ID

471112506928

Subnets (5) Info

Search

gwlb

Clear filters

	Name	Subnet ID	State	VPC	IPv4 CIDR
☐	gwlb-test6 App Subnet	subnet-019af9be7c93d0b18	Available	vpc-0a0564082a96d340e gwlb-test6-App-VPC	10.1.1.0/24
☐	gwlb-test6 GWLB Subnet	subnet-06f322c4981ddf750	Available	vpc-0a0564082a96d340e gwlb-test6-App-VPC	10.1.0.0/24
☐	gwlb-test6 Service CCL Subnet	subnet-0ba64defd4c0d32c1	Available	vpc-07de556bdbadab8a0 gwlb-test6 Service VPC	10.0.2.0/24
☐	gwlb-test6 Service Data Subnet	subnet-0bfa74422498815a9	Available	vpc-07de556bdbadab8a0 gwlb-test6 Service VPC	10.0.1.0/24
☐	gwlb-test6 Service Mgmt Subnet	subnet-011bccb8f6e6fdbf8	Available	vpc-07de556bdbadab8a0 gwlb-test6 Service VPC	10.0.0.0/24

VPC dashboard

EC2 Global View

Filter by VPC:

Select a VPC

Internet gateways (1/2) Info

Search

gwlb

Clear filters

	Name	Internet gateway ID	State	VPC ID
☒	gwlb-test6 IGW	igw-080963e296e704f3a	Attached	vpc-0a0564082a96d340e gwlb-test6-App-VPC
☐	gwlb-test6 Service Mgmt-IGW	igw-0b411380607e8bd51	Attached	vpc-07de556bdbadab8a0 gwlb-test6 Service VPC

igw-080963e296e704f3a / gwlb-test6 IGW

Details | Tags

Details

Internet gateway ID

igw-080963e296e704f3a

State

Attached

VPC ID

vpc-0a0564082a96d340e | gwlb-test6-App-VPC

Define Security Groups

Let's open the doors out wide

```
resource "aws_security_group" "allow_all" {
  name        = "Allow All"
  description = "Allow all traffic"
  vpc_id      = aws_vpc.srv_vpc.id

  ingress {
    from_port = 0
    to_port   = 0
    protocol  = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port = 0
    to_port   = 0
    protocol  = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "${var.env_name} Service SG"
    app  = "service"
  }
}
```

```
resource "aws_security_group" "app_allow_all" {
  name        = "Allow All"
  description = "Allow all traffic"
  vpc_id      = aws_vpc.app_vpc.id

  ingress {
    from_port = 0
    to_port   = 0
    protocol  = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

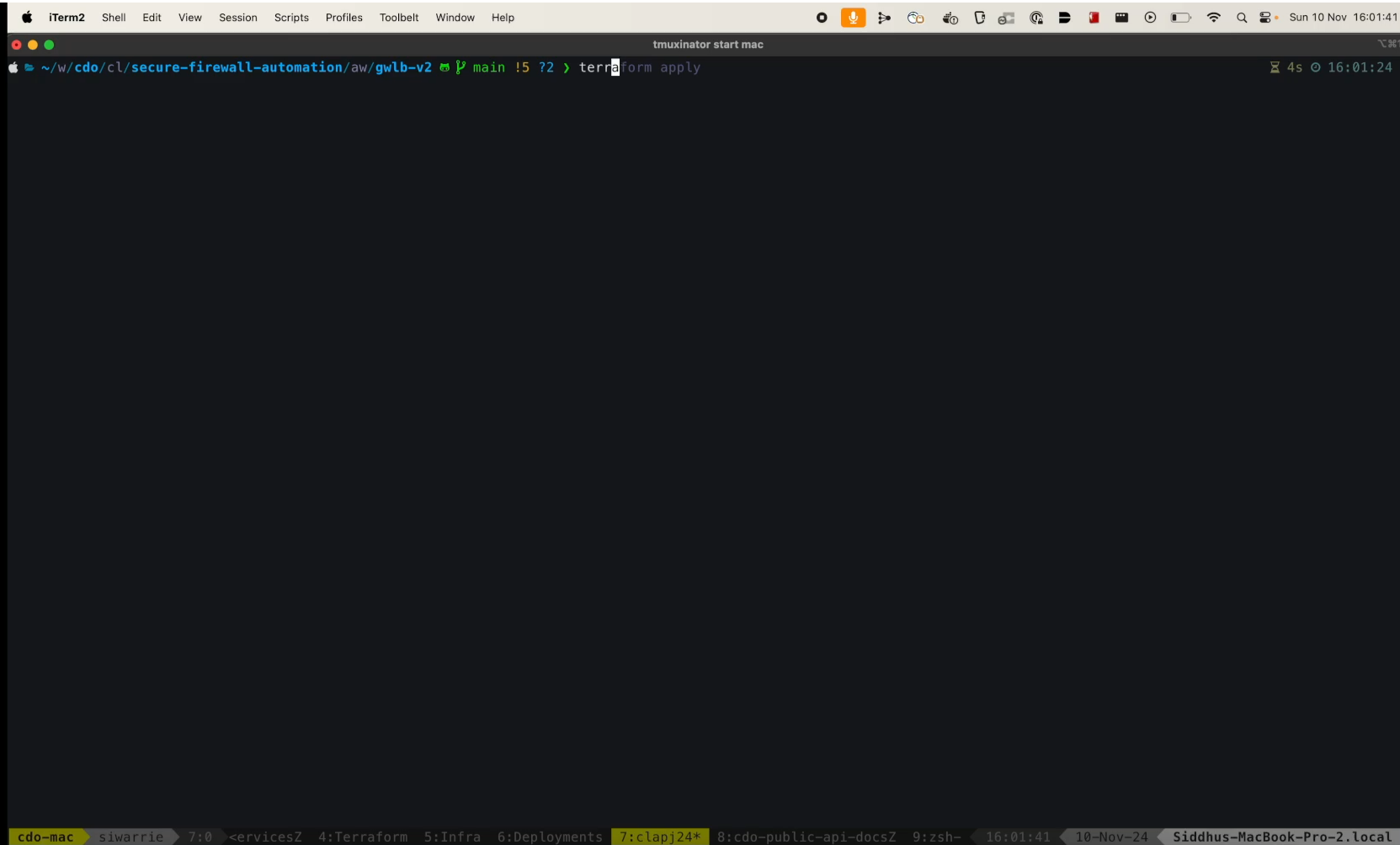
  egress {
    from_port = 0
    to_port   = 0
    protocol  = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "${var.env_name} Service SG"
    app  = "service"
  }
}
```

Security Groups Wide Open!?! No worries, We've got a Cisco Secure Firewall...

Terraform: Apply Changes

- 1 Open your Terminal
- 2 Type terraform apply and press the Enter key



Configure Cisco Security Cloud Control

Terraform: Get cdFMC in SCC tenant

```
data "cdo_cdfmc" "current" {  
}  
  
provider "fmc" {  
  fmc_host      = data.cdo_cdfmc.current.hostname  
  is_cdfmc      = true  
  cdo_token     = var.scc_api_token  
  cdfmc_domain_uuid = data.cdo_cdfmc.current.domain_uuid  
}
```

Assumption: Your Security Cloud Control tenant has a cdFMC provisioned

Terraform: Define default access policy

```
resource "fmc_access_policies" "access_policy" {  
  name          = "${var.env_name}-Access-Policy"  
  default_action = "block"  
}
```

Terraform: Create FTD resource in SCC

```
resource "cdo_ftd_device" "ftd-1" {  
  name                = var.ftd_name  
  access_policy_name  = resource.fmc_access_policies.access_policy.name  
  performance_tier    = "FTDv10"  
  virtual             = true  
  licenses            = ["BASE", "THREAT", "MALWARE", "URLFilter"]  
  lifecycle {  
    ignore_changes = [licenses]  
  }  
}
```

This generates a configure manager CLI you can paste into your FTD

Terraform: Apply Changes

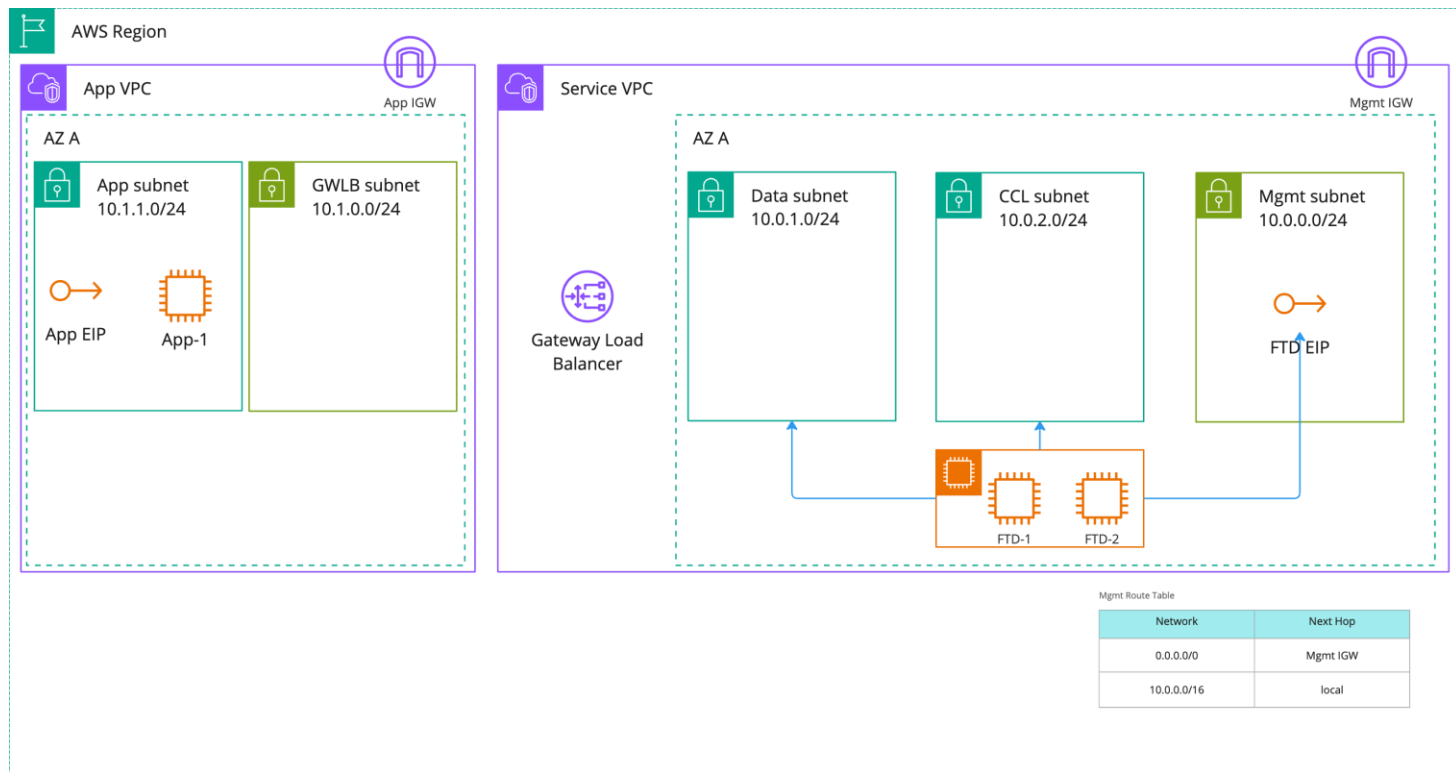
- 1 Open your Terminal
- 2 Type terraform apply and press the Enter key

~/W/cdo/cl/secure-firewall-automation/aw/gwlb-v2 main !7 74 >



Deploy Cisco Secure Firewall Threat Defense (FTD)

Cisco Secure Firewall Threat Defense (FTD)



Terraform: Define AMI

```
data "aws_ami" "ftdv" {
  most_recent = true
  owners      = ["67959333241"]
  #include_deprecated = true
  filter {
    name     = "name"
    values   = [var.ftd_version]
  }
  filter {
    name     = "product-code"
    values   = ["a8sxy6easi2zumgtyr564z6y7"]
  }
}
```

Terraform: Define Network Interfaces

```
resource "aws_network_interface" "ftd_ccl" {  
  description      = "ftd_ccl_if"  
  subnet_id        = aws_subnet.ccl_subnet.id  
  security_groups  = [aws_security_group.allow_all.id]  
  source_dest_check = false  
  tags = {  
    Name = "${var.env_name} Service FTD CCL"  
  }  
}
```

You, 2 minutes ago | 1 author (You)

Terraform: Define AWS Instance - User Data

```
# FTD Firewalls
resource "aws_instance" "ftd" {
  ami           = data.aws_ami.ftdv.id
  instance_type = "c5.xlarge"
  key_name      = aws_key_pair.public_key.key_name
  user_data_replace_on_change = true
  user_data     = <<-EOT
  {
    "AdminPassword":"${var.ftd_pass}",
    "Hostname":"${var.env_name}-FTDv",
    "ManageLocally":"No",
    "FmcIp": "${data.cdo_cdfmc.current.hostname}",
    "FmcRegKey":"${cdo_ftd_device.ftd_1.reg_key}",
    "FmcNatId":"${cdo_ftd_device.ftd_1.nat_id}",
    "Cluster":{
      "CclSubnetRange":"10.0.2.4 10.0.2.30",
      "ClusterGroupName":"${var.env_name}-cluster",
      "Geneve":"Yes",
      "HealthProbePort":"443"
    }
  }
  EOT
}
```

*Values from
Security Cloud Control*



Terraform: Define AWS Instance – Network Interfaces

```
network_interface {
  network_interface_id = aws_network_interface.ftd_management.id
  device_index         = 0
}

network_interface {
  network_interface_id = aws_network_interface.ftd_diagnostic.id
  device_index         = 1
}

network_interface {
  network_interface_id = aws_network_interface.ftd_data.id
  device_index         = 2
}

network_interface {
  network_interface_id = aws_network_interface.ftd_ccl.id
  device_index         = 3
}

tags = {
  Name = "${var.env_name} FTD"
}
```

Terraform: Elastic IP for FTD

```
resource "aws_eip" "ftd-mgmt-EIP" {  
    #vpc    = true  
    depends_on = [aws_internet_gateway.mgmt_igw,aws_instance.ftd]  
    tags = {  
        Name = "${var.env_name} Service FTD Mgmt EIP"  
        app  = "service"  
    }  
}  
  
resource "aws_eip_association" "ftd-mgmt-ip-association" {  
    network_interface_id = aws_network_interface.ftd_management.id  
    allocation_id         = aws_eip.ftd-mgmt-EIP.id  
}
```

Terraform: Wait for FTD to boot up

```
resource "aws_eip_association" "ftd-mgmt-ip-association" {
  network_interface_id = aws_network_interface.ftd_management.id
  allocation_id        = aws_eip.ftd-mgmt-EIP.id
}

resource "null_resource" "wait_for_ssh" {
  provisioner "remote-exec" {
    connection {
      type      = "ssh"
      host      = aws_eip.ftd-mgmt-EIP.public_ip
      user      = "admin"
      password   = local.ftd_pass
      timeout    = "20m" # Increase the timeout to 20 minutes, as FTD takes a while to boot
    }

    inline = [
      "echo FTD is ready"
    ]
  }

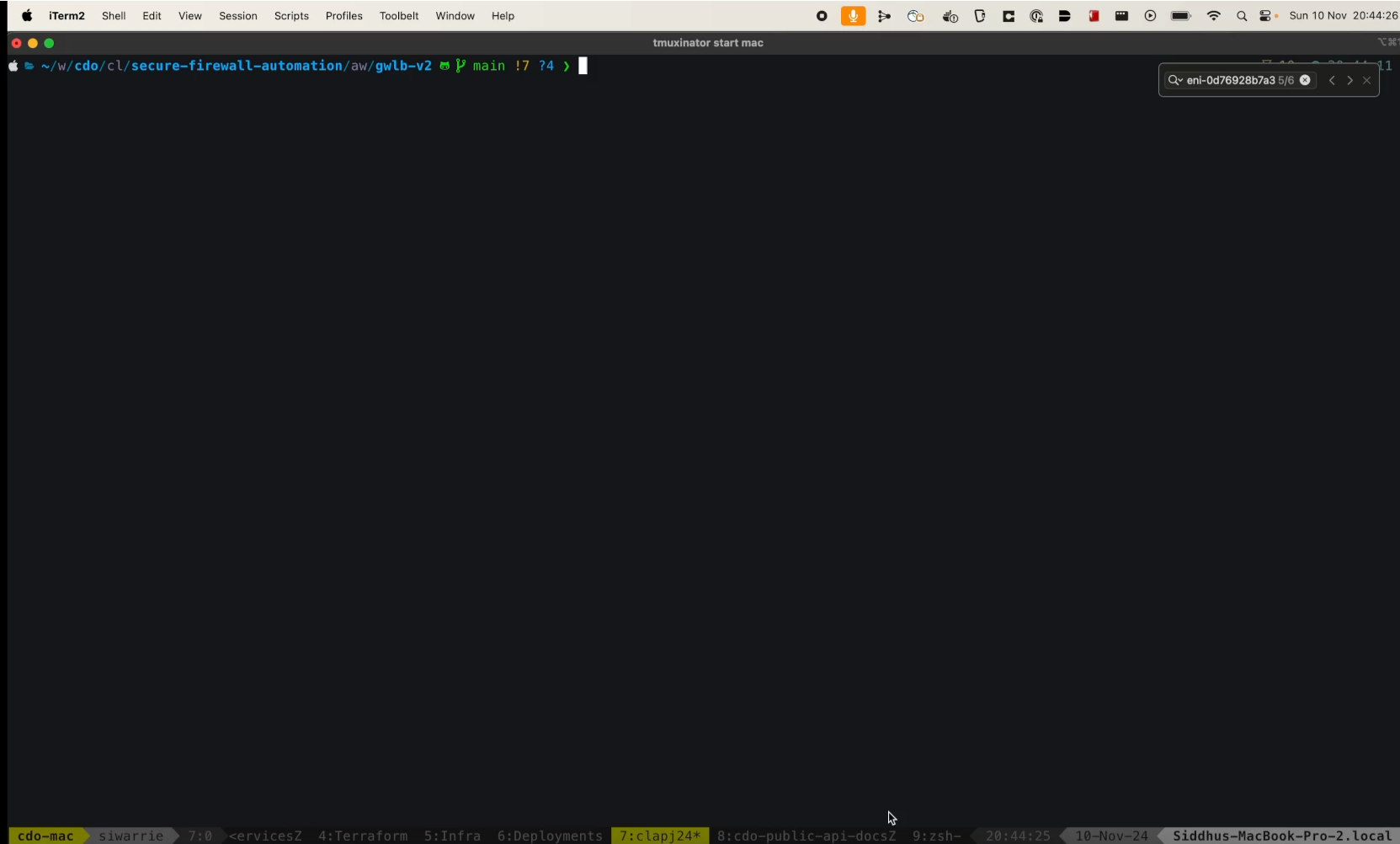
  depends_on = [aws_instance.ftd, aws_eip_association.ftd-mgmt-ip-association]
}
```

Terraform: Register FTD to Security Cloud Control

```
resource "cdo_ftd_device_onboarding" "ftds" {  
  ftd_uid      = cdo_ftd_device.ftd_1.id  
  depends_on = [aws_eip_association.ftd-mgmt-ip-association]  
}
```

Terraform: Apply Changes

- 1 Open your Terminal
- 2 Type terraform apply and press the Enter key



What it looks like on the AWS Console

The screenshot displays the AWS Management Console interface for EC2 instances. At the top, the 'Instances (1/1)' section is active, showing a single instance named 'gwlb FTD' with ID 'i-054b916b8f315250f'. The instance is in a 'Running' state, using a 'c5.xlarge' instance type, and is located in the 'ap-southeast-2a' availability zone. The public IPv4 address is '3.106.12.253'. Below the instance list, the 'Details' tab is selected for the instance 'i-054b916b8f315250f (gwlb FTD)'. The details section shows the instance summary, including the instance ID, public IPv4 address, private IPv4 addresses, instance state (Running), public IPv4 DNS, and private IP DNS name.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
gwlb FTD	i-054b916b8f315250f	Running	c5.xlarge	Initializing	View alarms	ap-southeast-2a	ec2-3-106-12-253.ap-s...	3.106.12.253

i-054b916b8f315250f (gwlb FTD)

- Details
- Status and alarms
- Monitoring
- Security
- Networking
- Storage
- Tags

Instance summary

Instance ID i-054b916b8f315250f	Public IPv4 address 3.106.12.253 open address	Private IPv4 addresses 10.0.1.143 10.0.2.68 10.0.0.10 10.0.0.182
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-3-106-12-253.ap-southeast-2.compute.amazonaws.com open address
Hostname type IP name is 10.0.0.10.ap-southeast-2.compute.internal	Private IP DNS name (IPv4 only) ip-10-0-0-10.ap-southeast-2.compute.internal	

What it looks like on Security Cloud Control

The screenshot shows the Cisco Security Cloud Control interface. The top navigation bar includes the Cisco logo, 'Security Cloud Control', and 'Security Devices'. A search bar is present with the placeholder text 'Type \'Ctrl\' + \'/\' to search'. The main content area displays a table of devices under the 'Devices' tab. The table has columns for Name, Configuration Status, and Connectivity. The 'gwib-cluster' is selected, and its details are shown on the right. The details include Location (n/a), Model (Cisco Firepower Threat Defense for AWS), Serial (9AWEQC9NXL), PCB Serial (n/a), Version (7.2.9), Onboarding Method (Registration Key), and Snort Version (3.1.21.900-7). The 'gwib-cluster' is also shown as a cluster with 1 device. The 'aws-ftd-cluster' is shown as a control node. The 'gwib-cluster' is in a 'Synced' state, and the 'aws-ftd-cluster' is in a 'Syncing' state. The 'gwib-cluster' is online, and the 'aws-ftd-cluster' is online. The 'gwib-cluster' is selected, and its details are shown on the right. The details include Location (n/a), Model (Cisco Firepower Threat Defense for AWS), Serial (9AWEQC9NXL), PCB Serial (n/a), Version (7.2.9), Onboarding Method (Registration Key), and Snort Version (3.1.21.900-7). The 'gwib-cluster' is also shown as a cluster with 1 device. The 'aws-ftd-cluster' is shown as a control node. The 'gwib-cluster' is in a 'Synced' state, and the 'aws-ftd-cluster' is in a 'Syncing' state. The 'gwib-cluster' is online, and the 'aws-ftd-cluster' is online. The 'gwib-cluster' is selected, and its details are shown on the right. The details include Location (n/a), Model (Cisco Firepower Threat Defense for AWS), Serial (9AWEQC9NXL), PCB Serial (n/a), Version (7.2.9), Onboarding Method (Registration Key), and Snort Version (3.1.21.900-7).

Name	Configuration Status	Connectivity
Alice-Springs FTD	Synced	Online
Goulburn FTD	Synced	Online
gwib-cluster FTD Cluster 1 device	Synced	Online
aws-ftd-cluster FTD Control Node	Syncing	Online

Device Details for gwib-cluster (FTD):

- Location: n/a
- Model: Cisco Firepower Threat Defense for AWS
- Serial: 9AWEQC9NXL
- PCB Serial: n/a
- Version: 7.2.9
- Onboarding Method: Registration Key
- Snort Version: 3.1.21.900-7

Cluster: aws-ftd-cluster (Control Node)

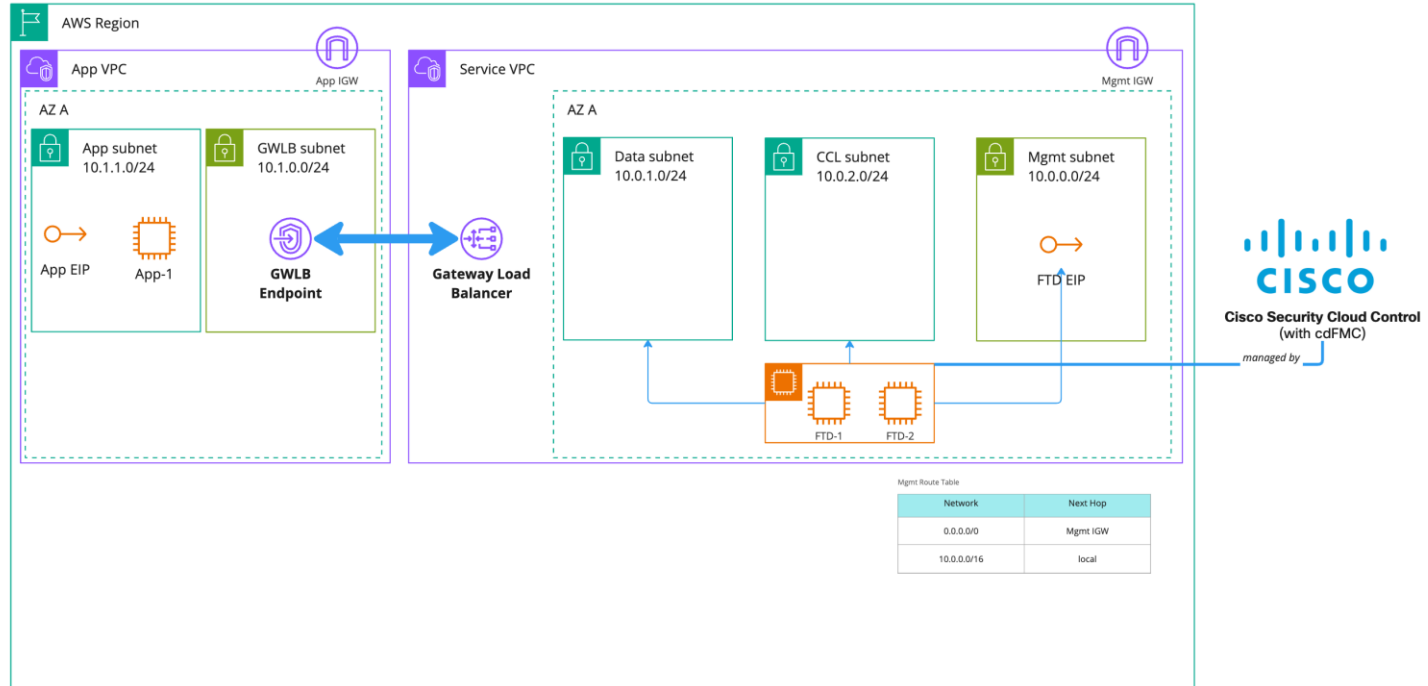
Synced: Your device's configuration is up-to-date.

Device Actions:

- Check for Changes
- Manage Licenses
- Workflows
- Remove
- Cluster Live Status

Deploy Gateway Load Balancer

Gateway Load Balancer



Terraform: Define Gateway Load Balancer

- Uses GENEVE to tunnel traffic through firewall
- Health Check on port 443
- Uses Source IP and Destination IP for target stickiness (default is round robin)

```
# Gateway Load Balancing related resources
resource "aws_lb" "gwlb" {
  name                        = "${var.env_name}-gwlb"
  load_balancer_type         = "gateway"
  subnets                   = [aws_subnet.data_subnet.id]
  enable_cross_zone_load_balancing = true

  tags = {
    Name = "${var.env_name} Service GWLB"
    app  = "service"
  }
}

# Target group is IP based since FTD's are provisioned with multiple interfaces
resource "aws_lb_target_group" "ftd" {
  name        = "${var.env_name}-ftdtg"
  protocol    = "GENEVE"
  vpc_id      = aws_vpc.srv_vpc.id
  target_type = "ip"
  port        = 6081
  stickiness {
    type = "source_ip_dest_ip"
  }
  health_check {
    port        = 443
    protocol    = "TCP"
  }
  tags = {
    Name = "${var.env_name} Service GWLB TG"
    app  = "service"
  }
}
```

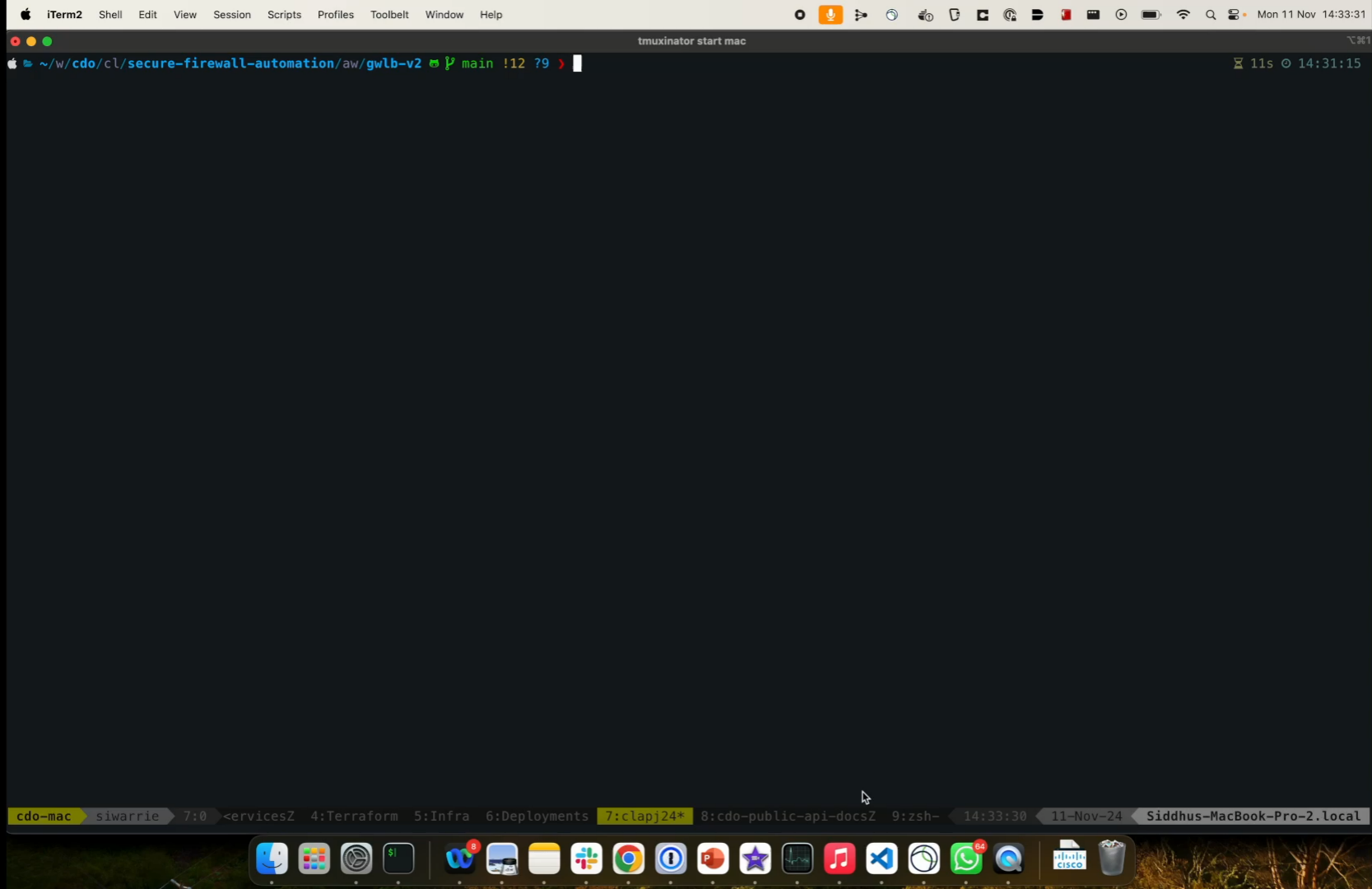
Terraform: Define GWLB Endpoint Service

```
# Endpoint Service
resource "aws_vpc_endpoint_service" "gwlb" {
  acceptance_required      = false
  gateway_load_balancer_arns = [aws_lb.gwlb.arn]
  tags = {
    Name = "${var.env_name} Service GWLB EP Service"
    app  = "service"
  }
}

# GWLB Endpoints. One is required for each AZ in App VPC
resource "aws_vpc_endpoint" "fw" {
  service_name      = aws_vpc_endpoint_service.gwlb.service_name
  vpc_endpoint_type = aws_vpc_endpoint_service.gwlb.service_type
  vpc_id            = aws_vpc.app_vpc.id
  tags = {
    Name = "${var.env_name} GWLBe"
  }
}
```

Terraform: Apply Changes

- 1 Open your Terminal
- 2 Type terraform apply and press the Enter key



On the AWS Console: Gateway Load Balancer

EC2 Dashboard

EC2 Global View

Events

Console-to-Code [Preview](#)

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations [New](#)

▼ Images

AMIs

[AMI Catalog](#)

▼ Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

▼ Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

▼ Load Balancing

[Load Balancers](#)

EC2 > Load balancers

Introducing resource map for Network Load Balancers

Resource map is a visual representation of the relationships between load balancer resources and provides the ability to view, explore, and troubleshoot the architecture of your load balancers on the load balancers detail page. Share feedback to help us improve your experience.

Load balancers (1/1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

☒	Name	DNS name	State	VPC ID	Availability Zones	Type	D
☒	gwlb-test6-gwlb	-	Active	vpc-07de556bdbadab...	us-east-1a (use1-az1)	gateway	M

Load balancer: gwlb-test6-gwlb

Details

Listeners

Network mapping

Monitoring

Integrations

Attributes

Tags

Details

Load balancer type

Gateway

Availability Zones

[subnet-0bfa74422498815a9](#) [us-east-1a \(use1-az1\)](#)

Load balancer ARN

[arn:aws:elasticloadbalancing:us-east-1:471112506928:loadbalancer/gwy/gwlb-test6-gwlb/07816ea1d5b1af20](#)

Status

Active

Date created

May 23, 2024, 15:13 (UTC-04:00)

VPC

[vpc-07de556bdbadab8a0](#)

cisco *Live!*

#CiscoLiveAPJC

DEVNET-1023

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

56

On the AWS Console: GWLB Target Group

EC2 > Target groups

Target groups (1/1) [Info](#) Refresh Actions Create target group

1 match

gwlb × Clear filters < 1 > ⚙️

☒	Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
☒	gwlb-test6-ftdtg	arn:aws:elasticloadbalanci...	6081	GENEVE	IP	gwlb-test6-gwlb	vpc-07de556bdbadab8a0

Target group: gwlb-test6-ftdtg ×

Details **Targets** Monitoring Health checks Attributes Tags

Registered targets (1) Refresh Deregister Register targets

< 1 > ⚙️

☐	IP address	Port	Zone	Health status	Health status details
☐	10.0.1.59	6081	us-east-1a	Healthy	-

On the AWS Console: GWLB Endpoint Service

VPC dashboard ×

EC2 Global View

Filter by VPC:

Select a VPC

▼ Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peering connections

▼ Security

Network ACLs

Security groups

▼ DNS firewall

Endpoint services (1/1) [Info](#)

☒	Name	Service ID	Types	Service name	State	Availability Zones	Acceptance req...
☒	gwlb-test6 Service GWLB EP Service	vpce-svc-0a20f72ba1db23719	GatewayLoadBalan...	com.amazonaws.vpce.us-east-1.vpce-sv...	Available	us-east-1a (use1-az1)	No

vpce-svc-0a20f72ba1db23719 / gwlb-test6 Service GWLB EP Service

Details

Load balancers

Allow principals

Endpoint connections

Notifications

Monitoring

Contributor Insights

Tags

Details

<div>Service ID</div> <div> vpce-svc-0a20f72ba1db23719</div>	<div>Types</div> <div>GatewayLoadBalancer</div>	<div>Service name</div> <div> com.amazonaws.vpce.us-east-1.vpce-svc-0a20f72ba1db23719</div>	<div>State</div> <div>Available</div>
<div>Network Load Balancers ARNs</div> <div>–</div>	<div>Gateway Load Balancers ARNs</div> <div> arn:aws:elasticloadbalancing:us-east-1:471112506928:loadbalancer/gwy/gwlb-test6-gwlb/07816ea1d5b1af20</div>	<div>Availability Zones</div> <div>us-east-1a (use1-az1)</div>	<div>Acceptance required</div> <div>No</div>
<div>DNS names</div> <div>–</div>	<div>Private DNS name</div> <div>–</div>	<div>Domain verification status</div> <div>Info</div>	<div>Domain verification name</div> <div>Info</div>
<div>Domain verification type</div> <div>Info</div>	<div>Domain verification value</div> <div>Info</div>	<div>Supported IP address type</div> <div> ipv4</div>	<div>–</div>

CISCO *Live!*

#CiscoLiveAPJC

DEVNET-1023

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

58

On the AWS Console: GWLB Endpoint

VPC dashboard

EC2 Global View

Filter by VPC:
Select a VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peering connections

Security

Network ACLs

Endpoints (1/1)

Search

Name	VPC endpoint ID	VPC ID	Service name	Endpoint type
gwlb-test6 GWLB	vpce-05bde45636ca1dc3a	vpc-0a0564082a96d340e gwlb-test6-...	com.amazonaws.vpce.us-east-1.vpce-svc-0a20f72ba1d...	GatewayLoadBalancer

vpce-05bde45636ca1dc3a / gwlb-test6 GWLB

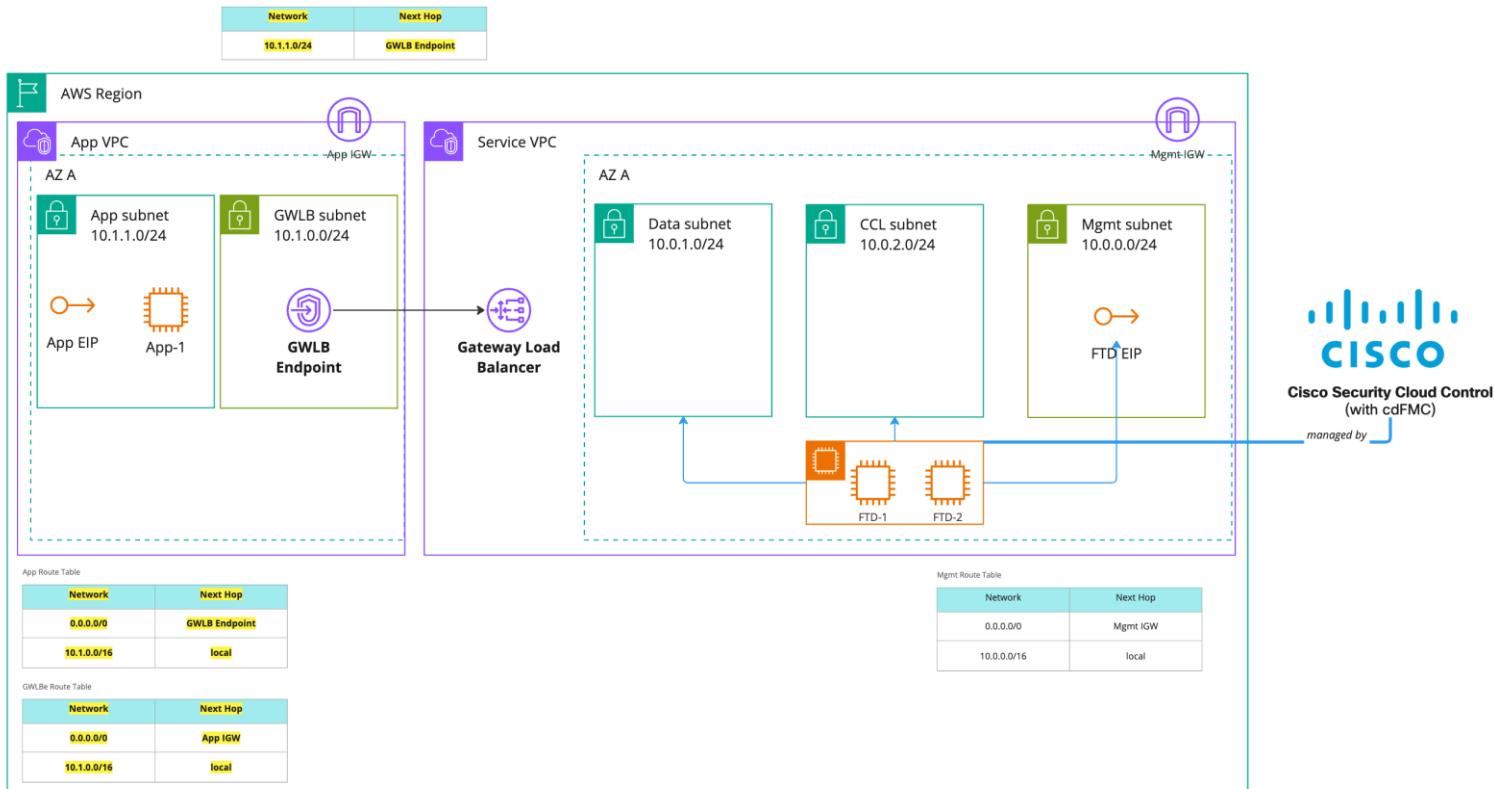
DetailsSubnetsNotificationMonitoringTags

Details

Endpoint ID vpce-05bde45636ca1dc3a	Status Available	Creation time Thursday, May 23, 2024 at 15:15:45 EDT	Endpoint type GatewayLoadBalancer
VPC ID vpc-0a0564082a96d340e (gwlb-test6-App-VPC)	Status message -	Service name com.amazonaws.vpce.us-east-1.vpce-svc-0a20f72ba1db23719	Private DNS names enabled -
IP address type ipv4			

Define Routes

Topology



Terraform: App Subnet Route Table

```
# App Route Table
resource "aws_route_table" "app_route_table" {
  vpc_id = aws_vpc.app_vpc.id
  tags = {
    Name = "${var.env_name } App Route Table"
  }
}

# App Default route to GWLB Endpoint
resource "aws_route" "app_default_route" {
  depends_on = [aws_vpc_endpoint.fw, aws_vpc_endpoint_subnet_association.fw]
  route_table_id      = aws_route_table.app_route_table.id
  destination_cidr_block = "0.0.0.0/0"
  vpc_endpoint_id      = aws_vpc_endpoint.fw.id
}

# App Subnet Route Association to App Route Table
resource "aws_route_table_association" "app1_association" {
  subnet_id      = aws_subnet.app_subnet.id
  route_table_id = aws_route_table.app_route_table.id
}
```

Terraform: GWLB Subnet Route Table

```
# GWLB Route Table
resource "aws_route_table" "gwlb_route_table" {
  vpc_id = aws_vpc.app_vpc.id
  tags = {
    Name = "${var.env_name } GWLB Route Table"
  }
}

#
# GWLB Default route to Internet Gateway
resource "aws_route" "gwlb_default_route" {
  depends_on = [aws_internet_gateway.app_igw]
  route_table_id      = aws_route_table.gwlb_route_table.id
  destination_cidr_block = "0.0.0.0/0"
  gateway_id          = aws_internet_gateway.app_igw.id
}

#
## GWLB Subnet Route Associations to GWLB Route Table
resource "aws_route_table_association" "gwlb_association" {
  subnet_id      = aws_subnet.gwlb_subnet.id
  route_table_id = aws_route_table.gwlb_route_table.id
}
```

Terraform: App IGW Route Table

```
# App IGW Route Table
resource "aws_route_table" "app_igw_route_table" {
  vpc_id = aws_vpc.app_vpc.id
  tags = {
    Name = "${var.env_name } IGW Route Table"
  }
}

# App IGW route to App Subnet via GWLBe
resource "aws_route" "app1_igw_route_app1_subnet" {
  depends_on = [aws_vpc_endpoint.fw, aws_vpc_endpoint_subnet_association.fw]
  route_table_id      = aws_route_table.app_igw_route_table.id
  destination_cidr_block = var.app_subnet
  vpc_endpoint_id      = aws_vpc_endpoint.fw.id
}

## App IGW Associations to App1 IGW Route Table
resource "aws_route_table_association" "app_igw_association" {
  gateway_id      = aws_internet_gateway.app_igw.id
  route_table_id = aws_route_table.app_igw_route_table.id
}
```

Terraform: Apply Changes

- 1 Open your Terminal
- 2 Type terraform apply and press the Enter key

What it looks like on the AWS Console

VPC dashboard X

EC2 Global View

Filter by VPC:
Select a VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peering connections

Route tables (1/4) info

Find resources by attribute or tag

App X Clear filters

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
☐ gwlb-test6 IGW Route Table	rtb-0c9c6e1791089f66a	-	igw-080963e296e704f3a	No	vpc-0a0564082a96d340e gwlb-test6-App-VPC
☒ gwlb-test6 GWLB Route Table	rtb-00396b1c551c5ae48	subnet-06f322c4981d6f...	-	No	vpc-0a0564082a96d340e gwlb-test6-App-VPC
☐ gwlb-test6 App Route Table	rtb-0352163a6a8b1d530	subnet-019a9b0e7c95d0...	-	No	vpc-0a0564082a96d340e gwlb-test6-App-VPC
☐ -	rtb-04a0f6852fcca093	-	-	Yes	vpc-0a0564082a96d340e gwlb-test6-App-VPC

rtb-00396b1c551c5ae48 / gwlb-test6 GWLB Route Table

Details Routes Subnet associations Edge associations Route propagation Tags

Routes (2)

Filter routes

Destination	Target	Status	Propagated
0.0.0.0/0	igw-080963e296e704f3a	Active	No
10.1.0.0/16	local	Active	No

VPC dashboard X

EC2 Global View

Filter by VPC:
Select a VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peering connections

Route tables (1/4) info

Find resources by attribute or tag

App X Clear filters

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
☒ gwlb-test6 IGW Route Table	rtb-0c9c6e1791089f66a	-	igw-080963e296e704f3a	No	vpc-0a0564082a96d340e gwlb-test6-App-VPC
☐ gwlb-test6 GWLB Route Table	rtb-00396b1c551c5ae48	subnet-06f322c4981d6f...	-	No	vpc-0a0564082a96d340e gwlb-test6-App-VPC
☐ gwlb-test6 App Route Table	rtb-0352163a6a8b1d530	subnet-019a9b0e7c95d0...	-	No	vpc-0a0564082a96d340e gwlb-test6-App-VPC
☐ -	rtb-04a0f6852fcca093	-	-	Yes	vpc-0a0564082a96d340e gwlb-test6-App-VPC

rtb-0c9c6e1791089f66a / gwlb-test6 IGW Route Table

Details Routes Subnet associations Edge associations Route propagation Tags

Routes (2)

Filter routes

Destination	Target	Status	Propagated
10.1.0.0/16	local	Active	No
10.1.1.0/24	vpc-05bde45636ca1db3a	Active	No

VPC dashboard X

EC2 Global View

Filter by VPC:
Select a VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

Internet gateways (1/2) info

Search

gwlb X Clear filters

Name	Internet gateway ID	State	VPC ID
☒ gwlb-test6 IGW	igw-080963e296e704f3a	Attached	vpc-0a0564082a96d340e gwlb-test6-...
☐ gwlb-test6 Service Mgmt-IGW	igw-0b411380607e8bd51	Attached	vpc-07d65568dbadab8a0 gwlb-test6-...

igw-080963e296e704f3a / gwlb-test6 IGW

Details Tags

Details

Internet gateway ID	State	VPC ID
igw-080963e296e704f3a	Attached	vpc-0a0564082a96d340e gwlb-test6-App-VPC

Define App Workload

Terraform: Define AMI

```
data "aws_ami" "ami_linux" {  
  most_recent = true  
  owners      = ["amazon"]  
  filter {  
    name     = "name"  
    values   = ["amzn2-ami-hvm*"]  
  }  
  filter {  
    name     = "root-device-type"  
    values   = ["ebs"]  
  }  
  filter {  
    name     = "architecture"  
    values   = ["x86_64"]  
  }  
}
```

Terraform: Create EC2 Instance

```
# Instances in App VPC
resource "aws_instance" "app" {
  ami            = data.aws_ami.ami_linux.id
  instance_type  = "t2.micro"
  key_name       = aws_key_pair.public_key.key_name
  subnet_id     = aws_subnet.app_subnet.id
  private_ip    = var.app_server
  associate_public_ip_address = true
  vpc_security_group_ids = [
    aws_security_group.app_allow_all.id
  ]
  tags = {
    Name = "${var.env_name} App Server"
  }
}
```

Terraform: Apply Changes

- 1 Open your Terminal
- 2 Type terraform apply and press the Enter key

```

~/.w/cdo/cl/secure-firewall-automation/aw/gwlb-v2 main !12 ?9 > terraform apply
random_string.id: Refreshing state... [id=9633]
random_password.ftd_pass: Refreshing state... [id=none]
data.cdo_cdfmc.current: Reading...
tls_private_key.key_pair: Refreshing state... [id=39be45bd15f0c53138a1e200d0ae2cf0f2444899]
local_file.this: Refreshing state... [id=5c759ae0ee978a7862acf988232ffe92cadceca1]
data.cdo_cdfmc.current: Read complete after 0s [id=93924a24-cba0-46b3-8dcd-1f5079b407dd]
fmc_access_policies.access_policy: Refreshing state... [id=0222F49A-929F-0ed3-0000-004294972936]
aws_key_pair.public_key: Refreshing state... [id=gwlb-9633-key]
data.aws_ami.ami_linux: Reading...
aws_vpc.srv_vpc: Refreshing state... [id=vpc-051e1a5229a535bb2]
aws_vpc.app_vpc: Refreshing state... [id=vpc-0ca378f46d5090210]
data.aws_ami.ftdv: Reading...
cdo_ftd_device.ftd_1: Refreshing state... [id=dedb9c58-56c7-4b2b-971e-bb5dd04152ae]

```



Define Access Control Policy

Terraform: Configure IPS policy

```
data "fmc_ips_policies" "ips_policy" {  
  name = "Security Over Connectivity"  
}
```

```
# IPS Policy  
resource "fmc_ips_policies" "ips_policy" {  
  name           = "${var.env_name}-ips_policy"  
  inspection_mode = "DETECTION"  
  basepolicy_id  = data.fmc_ips_policies.ips_policy.id  
}
```

Terraform: Allow outbound access on 80/443

```
resource "fmc_access_rules" "access_rule_1" {
  depends_on = [fmc_access_policies.access_policy]
  acp         = fmc_access_policies.access_policy.id
  section     = "mandatory"
  name        = "${var.env_name}_permit_outbound"
  action      = "allow"
  enabled     = true
  send_events_to_fmc = true
  log_files   = false
  log_begin   = true
  log_end     = true
  source_networks {
    source_network {
      id = fmc_network_objects.app_subnet.id
      type = "Network"
    }
  }
  destination_ports {
    destination_port {
      id = data.fmc_port_objects.http.id
      type = "TCPPortObject"
    }
    destination_port {
      id = data.fmc_port_objects.https.id
      type = "TCPPortObject"
    }
  }
  ips_policy = fmc_ips_policies.ips_policy.id
  new_comments = ["${var.env_name} outbound web traffic"]
}
```

Terraform: Allow inbound SSH access

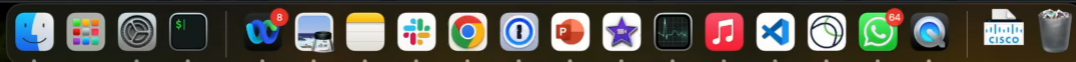
```
resource "fmc_access_rules" "access_rule_2" {
  depends_on = [fmc_access_policies.access_policy]
  acp        = fmc_access_policies.access_policy.id
  section    = "mandatory"
  name       = "${var.env_name}_access_to_app_server"
  action     = "allow"
  enabled    = true
  send_events_to_fmc = true
  log_files  = false
  log_begin  = true
  log_end    = true
  destination_networks {
    destination_network {
      id = fmc_host_objects.app_server.id
      type = "Host"
    }
  }
  destination_ports {
    destination_port {
      id = data.fmc_port_objects.ssh.id
      type = "TCPPortObject"
    }
  }
  ips_policy = fmc_ips_policies.ips_policy.id
  new_comments = ["${var.env_name} ssh to app server"]
}
```

Terraform: Deploy Changes to FTD

```
data "fmc_devices" "device" {  
  name = var.ftd_name  
}  
  
resource "fmc_ftd_deploy" "gwlb_cluster" {  
  device = data.fmc_devices.device.id  
  ignore_warning = false  
  force_deploy = false  
}
```

Terraform: Apply Changes

- 1 Open your Terminal
- 2 Type terraform apply and press the Enter key



What it looks like on Security Cloud Control

The screenshot displays the Cisco Security Cloud Control interface for managing the 'gwlb-access-policy'. The left sidebar contains navigation links for Home, Analysis, Policies (selected), Devices, and Objects. The top navigation bar shows a breadcrumb trail: Packets → Prefilter Rules → Decryption → Security Intelligence → Identity → Access Control (selected) → More. Below the breadcrumb is a search bar with the text 'Type to search' and a filter icon. The main content area shows a table of rules under the 'Mandatory (1 - 2)' group. The table has columns for Name, Action, Source (Zones and Networks), and Destination (Zones, Networks, and Ports). Two rules are listed: 'gwlb_permit_outbound' and 'gwlb_access_to_app_serv..', both with an 'Allow' action.

[Return to Access Control Policy Management](#)

gwlb-access-policy

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

Type to search Total 2 rules

	Name	Action	Source		Destination		
			Zones	Networks	Zones	Networks	Ports
☐ ▾	Mandatory (1 - 2)						
☐	1 gwlb_permit_outbound	→ Allow	Any	gwlb-app_subnet	Any	Any	HTTP HTTPS
☐	2 gwlb_access_to_app_serv..	→ Allow	Any	Any	Any	gwlb-app_server	SSH

What it looks like on Security Cloud Control (II)

The screenshot displays the Cisco Security Cloud Control web interface. On the left is a sidebar with navigation icons for Home, Analysis, Policies (highlighted with a blue bar), and Devices. The main content area has a header with 'Intrusion Policies' (active) and 'Network Analysis Policies'. Below the header is a search bar with the placeholder text 'Search by Intrusion Policy, Description, or Base Policy'. A link 'Show Snort 3 Sync status' with an information icon is also present. The main content is a table with four columns: 'Intrusion Policy', 'Description', 'Base Policy', and 'Usage Information'. One row is visible, showing 'gwlb-ips_policy' as the intrusion policy, which is linked to the 'Security Over Connectivity' base policy. The usage information indicates it is an 'Access Control Policy' with 'No Zero Trust Application Policy' and is targeting 1 device.

Intrusion Policy	Description	Base Policy	Usage Information
gwlb-ips_policy		Security Over Connectivity	1 Access Control Policy <i>No Zero Trust Application Policy</i> Targeting 1 device

Complete Your Session Evaluations



Complete a minimum of 4 session surveys and the Overall Event Survey to claim a **Cisco Live T-Shirt**.



Complete your surveys in the **Cisco Live mobile app**.

Continue your education



- Visit the Cisco Stand for related demos
- Book your one-on-one Meet the Expert meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

Contact me at: siwarrie@cisco.com on e-mail or Webex



Thank you

CISCO *Live!*

#CiscoLiveAPJC