



The bridge to possible

Resilience and Scaling for Security Solutions in the Public Cloud

Eric Kostlan, Technical Marketing Engineer
BRKSEC-3457

CISCO *Live!*

#CiscoLiveAPJC



“Reconciliation”- Dustin Koa Art

```
ubuntu@Azure200 :~ $ ping -c 1 10.200.101.6 > /dev/null
ubuntu@Azure200 :~ $ ping -c 1 10.200.101.8 > /dev/null
ubuntu@Azure200 :~ $ arp -an
? (10.200.101.1) at 12:34:56:78:9a:bc [ether] on eth0
? (10.200.101.6) at 12:34:56:78:9a:bc [ether] on eth0
? (10.200.101.8) at 12:34:56:78:9a:bc [ether] on eth0
ubuntu@Azure200 :~ $
```

*“The public cloud is easy to learn.
Step 1. Forget everything you know
about networking.”*

Ancient Technical Marketing Maxim

Cisco Webex App

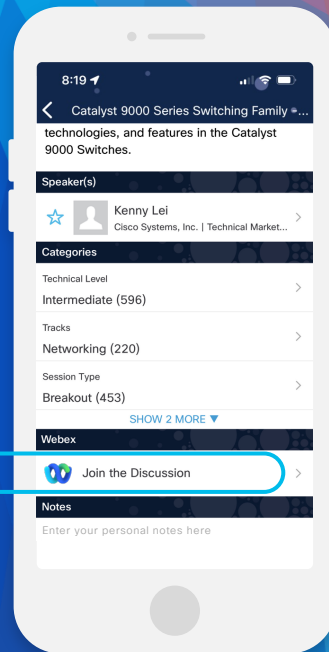
Questions?

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 December 22, 2023.

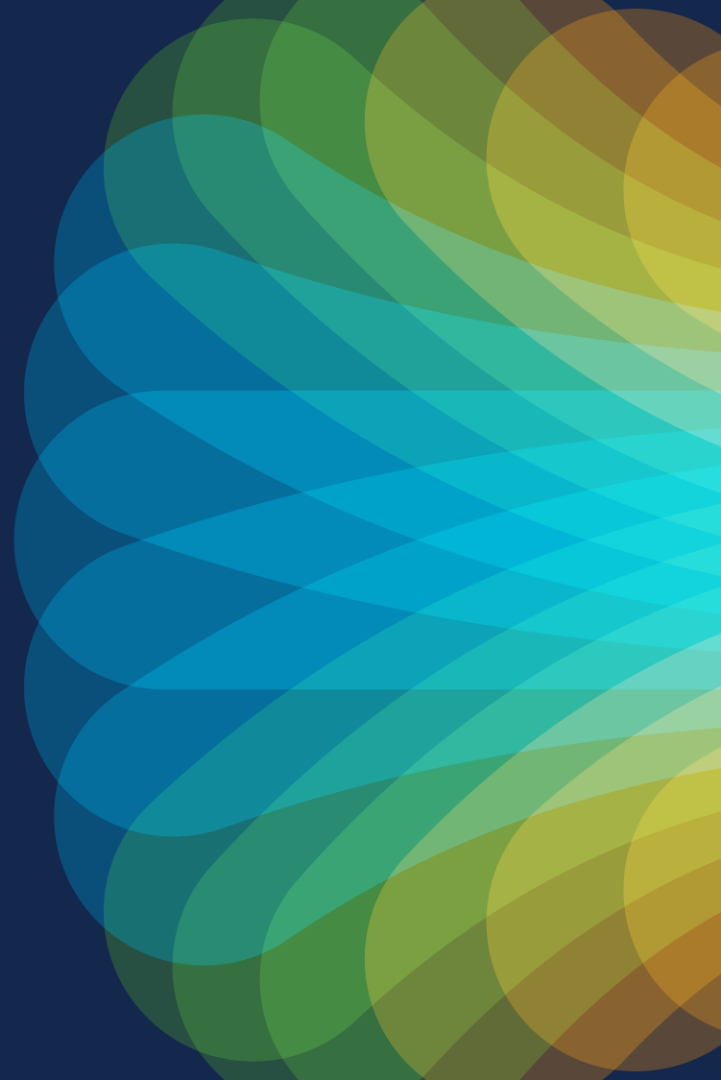


<https://ciscolive.ciscoevents.com/ciscolivebot/#BRKSEC-3457>

Agenda

- Introduction
- Load Balancers
- High Availability and Resilience
- Cisco Secure Firewall
- Cisco Multicloud Defense
- Cisco Secure Workload
- Conclusion

Introduction



“When the Internet was first introduced to the public in the 1990s, many people were skeptical about conducting financial transactions online due to security risks. The lack of established security measures and the fear of online fraud made it difficult for people to trust the Internet for commercial purposes.”

ChatGPT

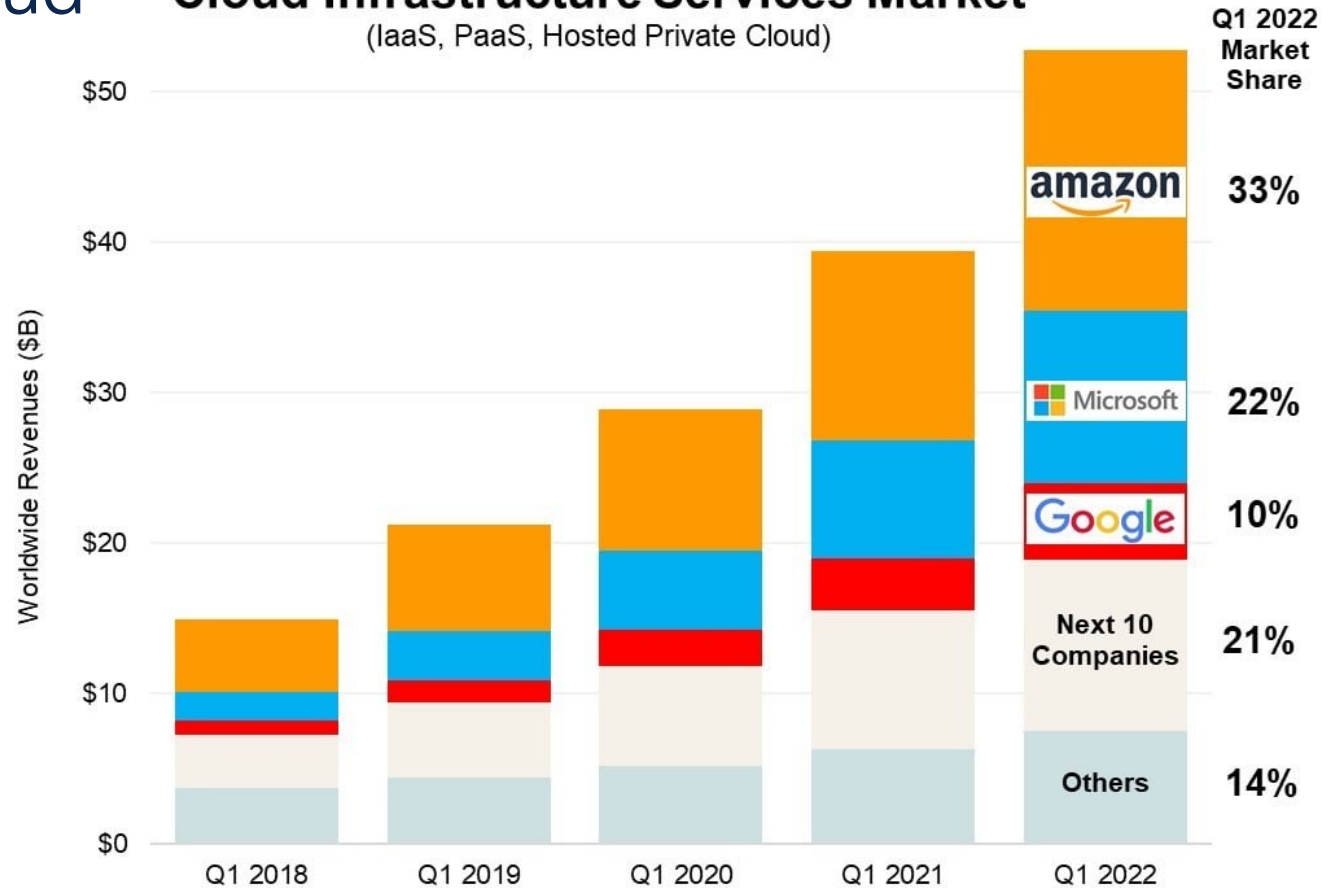
“But the value proposition was too great to wait for security. History repeats itself with the rush to the public cloud. Security can only attempt to keep up.”

Human

Public Cloud Market

Cloud Infrastructure Services Market

(IaaS, PaaS, Hosted Private Cloud)

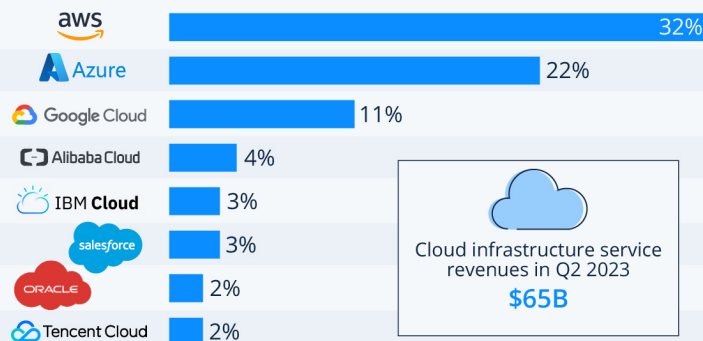


Source: Synergy Research Group

Public Cloud Market Share

Amazon Maintains Lead in the Cloud Market

Worldwide market share of leading cloud infrastructure service providers in Q2 2023*



* Includes platform as a service (PaaS) and infrastructure as a service (IaaS) as well as hosted private cloud services

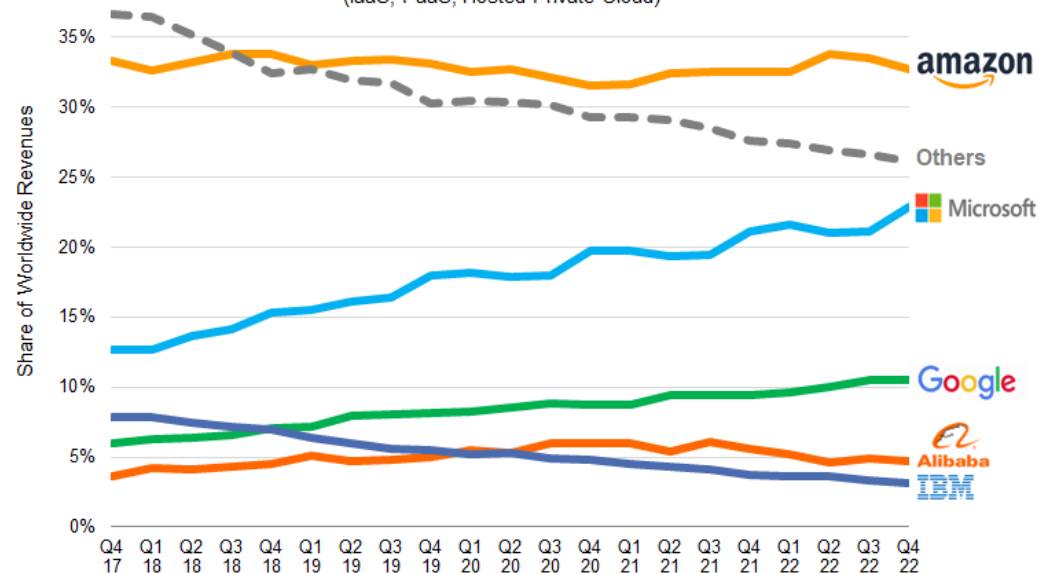
Source: Synergy Research Group



statista

Cloud Provider Market Share Trend

(IaaS, PaaS, Hosted Private Cloud)



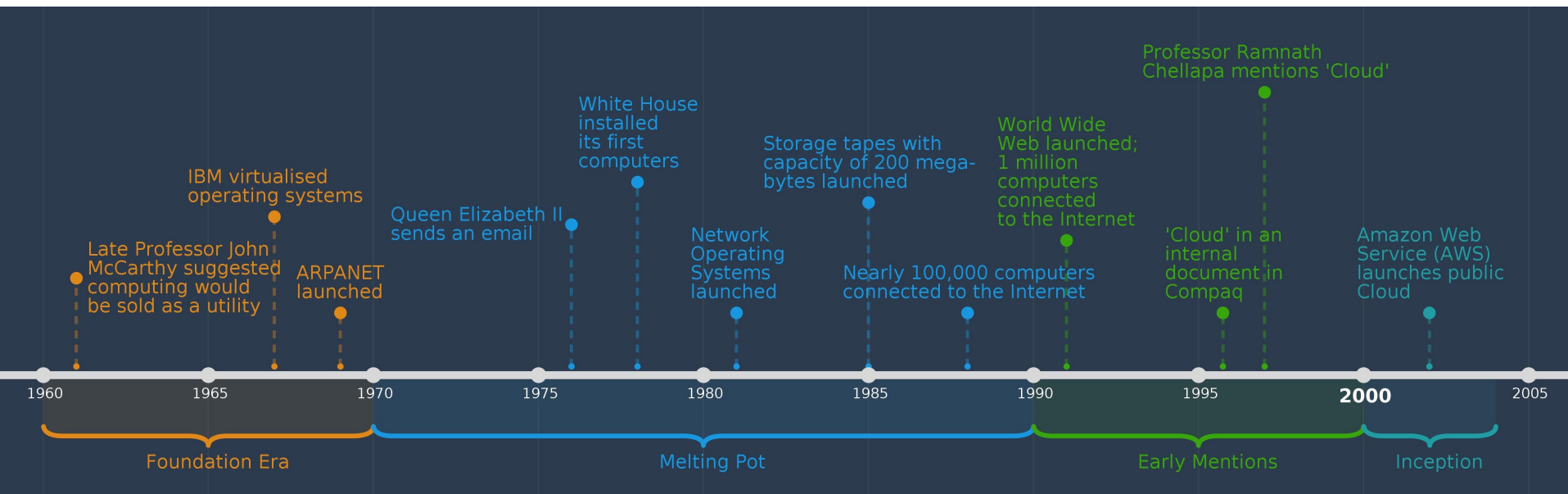
Source: Synergy Research Group

*“Keep in mind: ‘AWS’ stands for
‘Amazon Web Services’.”*

Ancient Technical Marketing Maxim

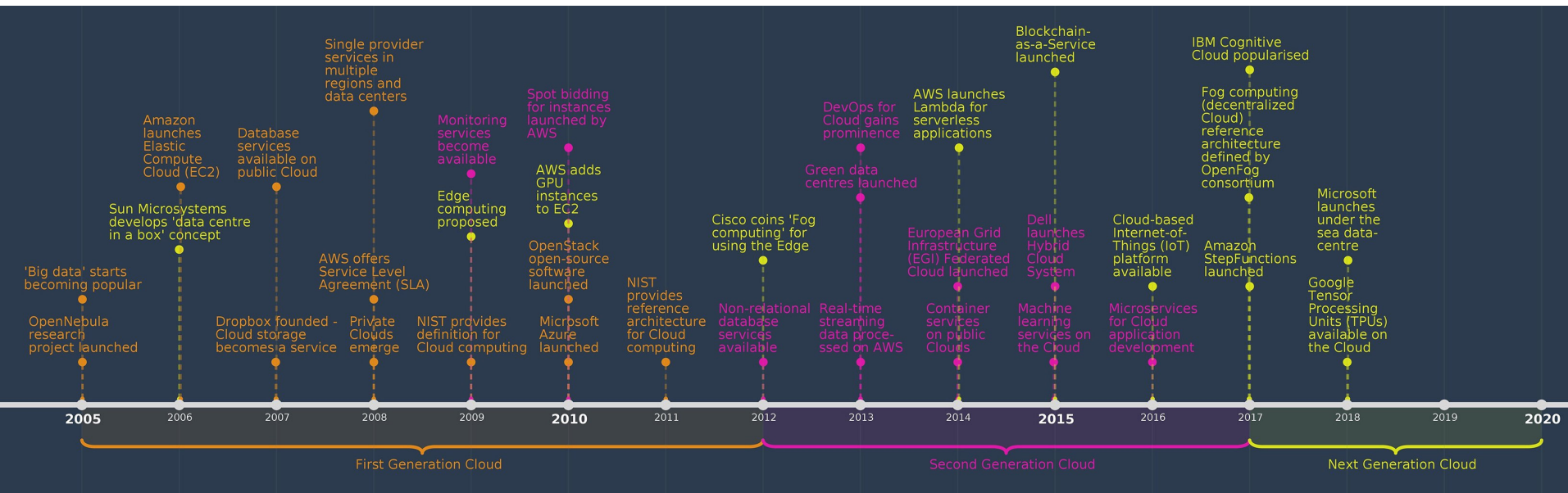
History of the Public Cloud

Source: Blesson Varghese, *History of the Cloud*, 2019



History of the Public Cloud

Source: Blesson Varghese, *History of the Cloud*, 2019



Public Cloud Security Coevolution

- Customers are rushing to deploy in the public cloud
 - Some customers think they can replicate their enterprise networks in the public cloud
 - Most know better
- Security vendors partner with public cloud providers to facilitate coevolution
 - Public cloud providers are evolving to prove broader networking capabilities
 - Example: gateway load balancers for service insertion

What is *Scale* in the Public Cloud?

- Scaling volume of traffic
 - Load Balancing
- Scaling connectivity
 - Remote access VPN
 - Intracloud connectivity (AWS Transit Gateway, Azure vWAN Hub)
 - Intercloud connectivity (AWS VPG, Azure VNG, Multicloud Defense*)
- Accelerating application development
 - Complex application dependencies and associated security challenges
- Scaling visibility – including dynamic attributes

*Feature coming soon

Load Balancers

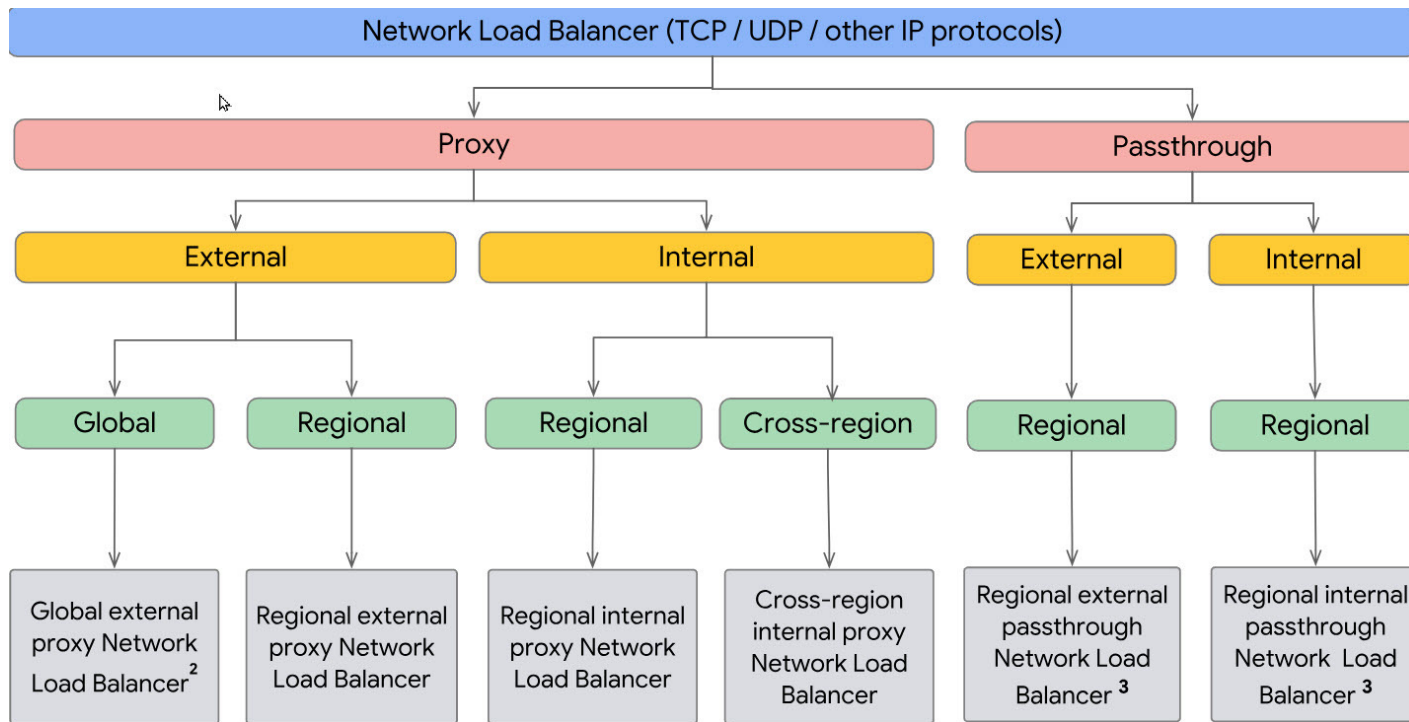
- Network Load Balancers
- Application Load Balancers
- Gateway Load Balancers
- DNS Load Balancers

Network Load Balancers (NLBs)

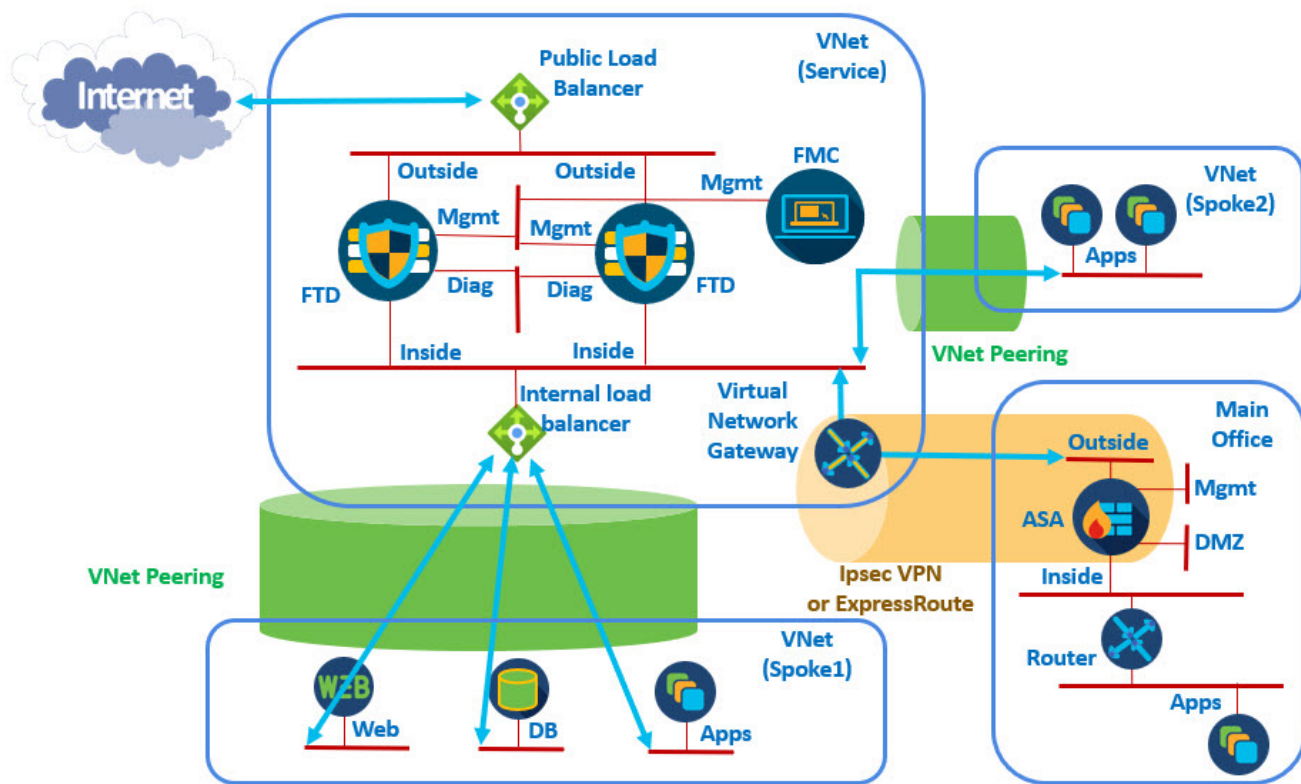
- External (public) load balancers
 - Frontend service definition
 - NAT
 - Destination NAT is motivated by behavior of backend servers. Typically, backend servers expect traffic to be addressed to them.
 - Not necessary for firewalls. You may want to preserve the destination IP if a single load balancer has multiple frontend IPs (Azure *Floating IP* feature)
- Internal load balancers
 - Used for east-west and outbound traffic
 - Azure ILB can handle all ports and protocols (HA port feature)
 - GCP ability to handle all ports and protocols is in review

GCP Network Load Balancers

<https://cloud.google.com/load-balancing/docs/choosing-load-balancer>



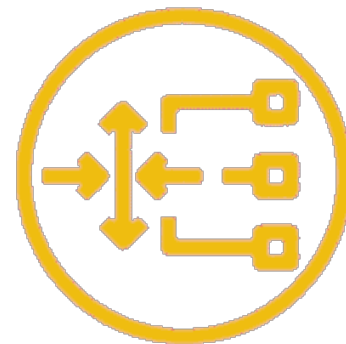
Load Balancer Sandwich (Azure)



Gateway Load Balancers (GLBs)

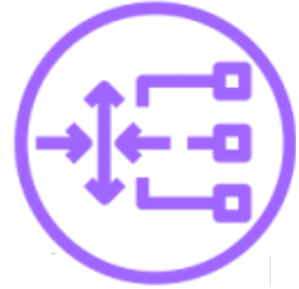
The right way to do service insertion in the public cloud!

- Traditional load balancers were designed to scale application servers, not firewalls
- Public cloud providers are adopting gateway load balancers (GWLBs) to address this need
 - AWS introduced it in November 2020
 - Azure introduced it in November 2021
- GWLBs allow transparent insertion of firewalls into the data path.

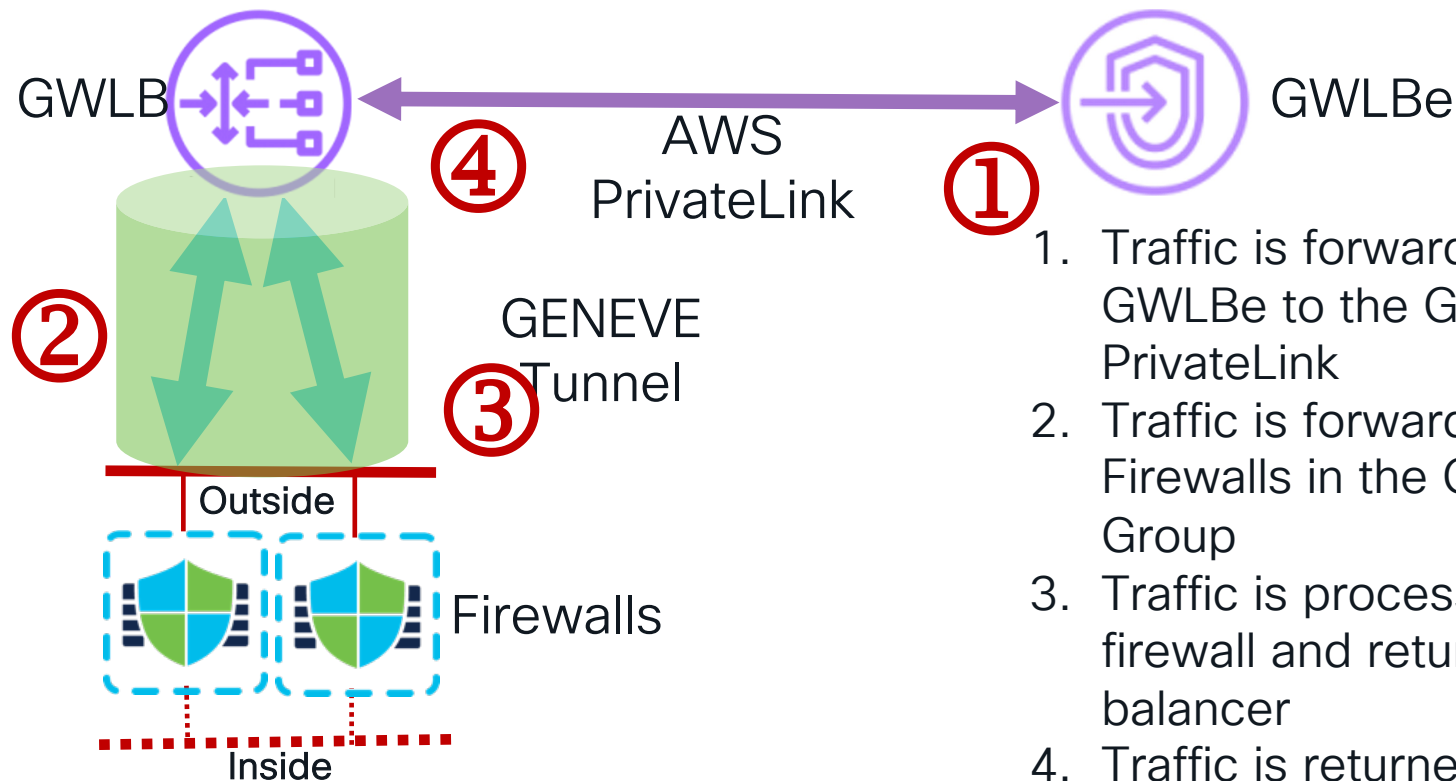


AWS GWLB, GWLB Endpoint (GWLBBe), and Endpoint Service

- GWLB
 - Forwards traffic to the firewalls **inside** a GENEVE tunnel
 - Load balances between firewalls in a Target Group
 - Does not modify packets or packet headers
 - Forwards all packets associated with a particular connection to the same firewall
- GWLB endpoint
- Endpoint service
 - At most one endpoint service per GWLB
 - Associates one or more GWLBes to a GWLB
 - Creates a private link between the GWLB and GWLBBe

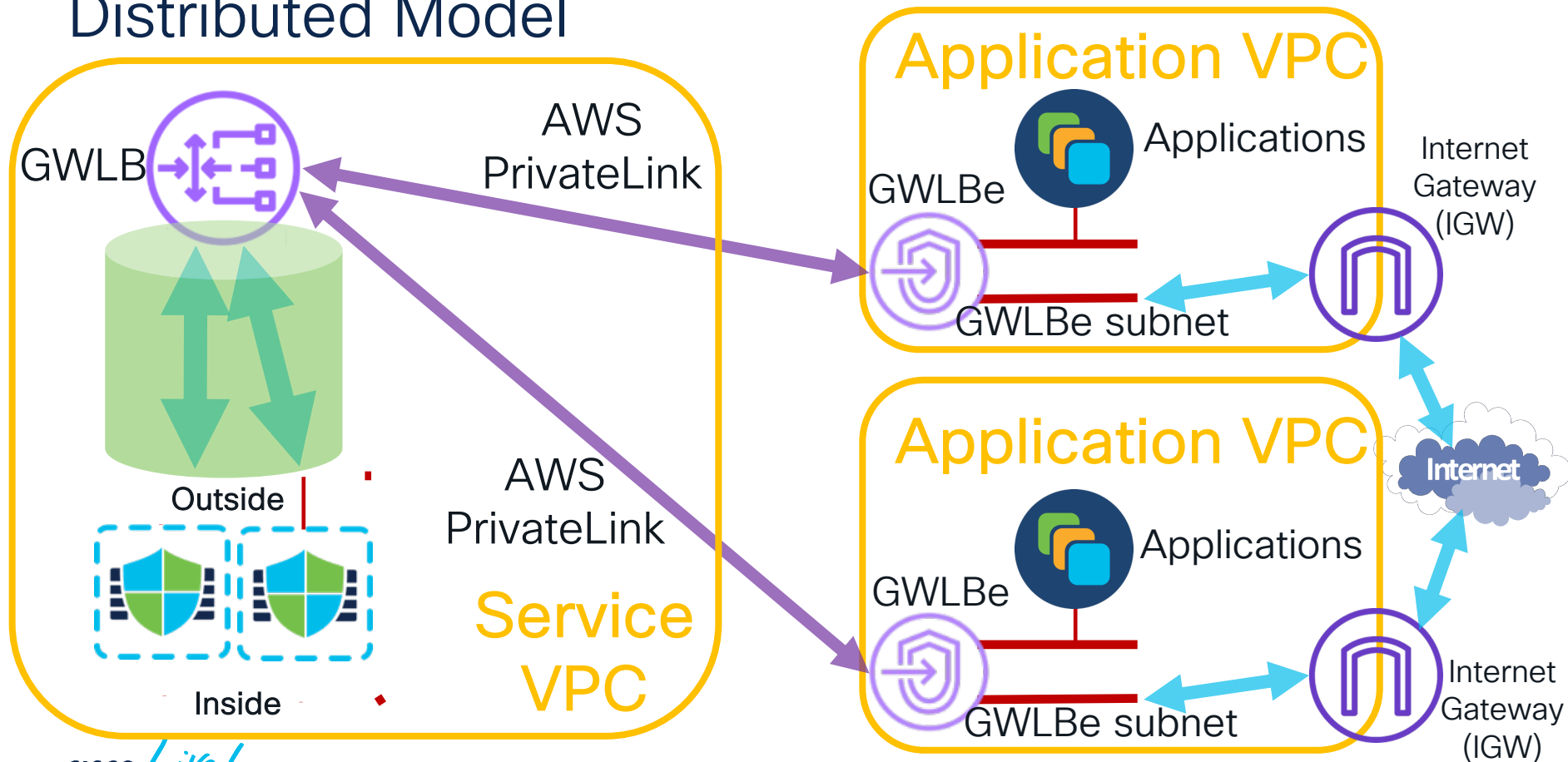


Traffic Flow Between the GWLBe, GWLB, and Firewalls

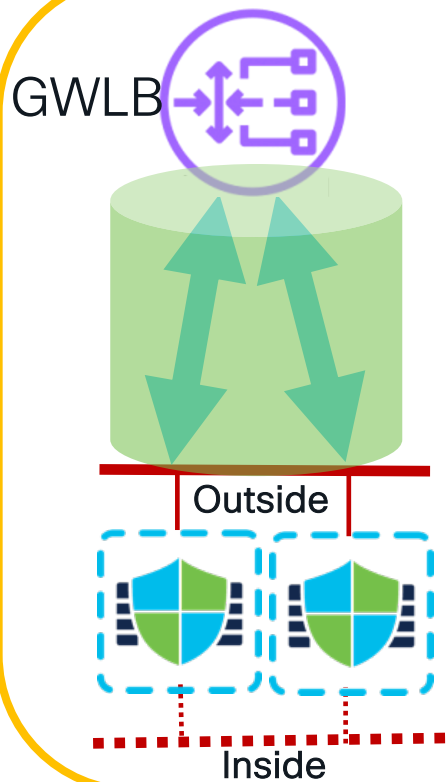


1. Traffic is forwarded from the GWLBe to the GWLB across PrivateLink
2. Traffic is forwarded to one of the Firewalls in the GWLB Target Group
3. Traffic is processed by the firewall and returned to the load balancer
4. Traffic is returned to the GWLBe

North/South and Outbound Traffic Distributed Model



East West Traffic – Problem



Service
VPC

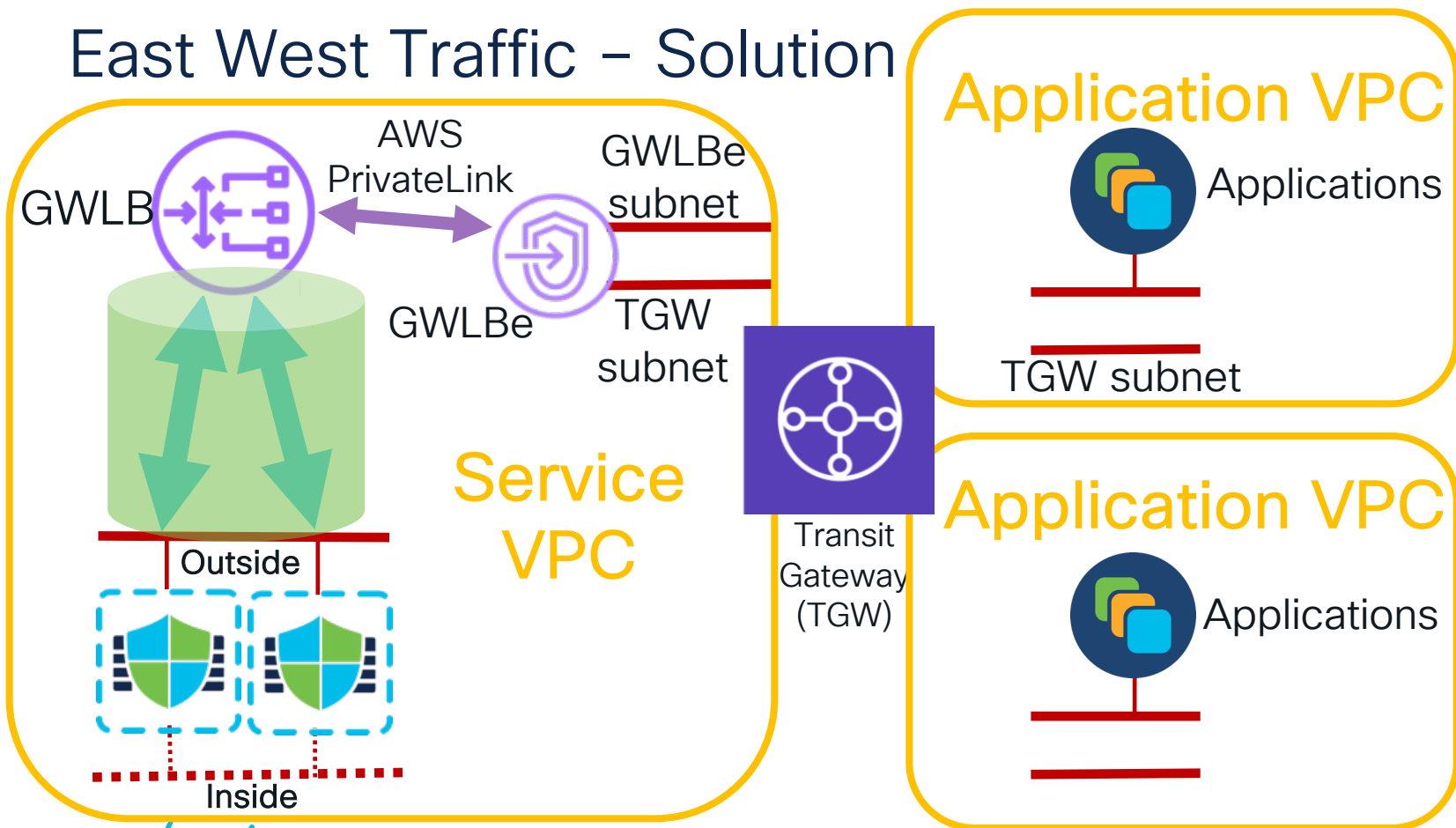
Application VPC



Application VPC

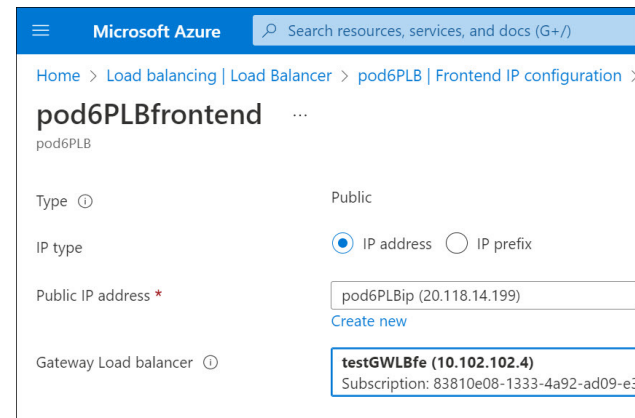
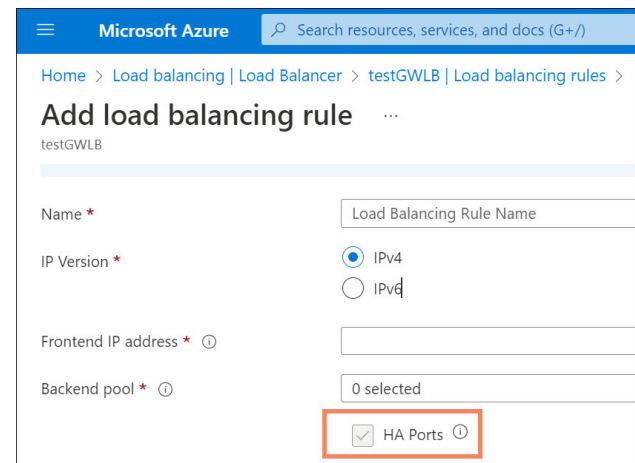


East West Traffic – Solution

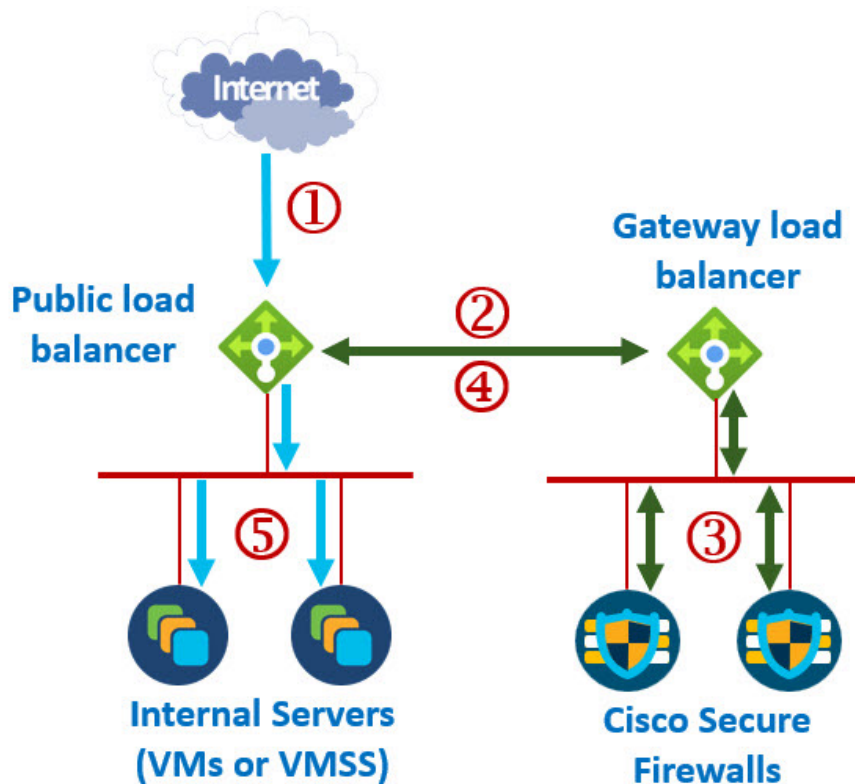


Azure GWLB

- Associated with a (standard SKU) public load balancer or network interface.
 - Network interface must have a (standard SKU) public IP.
 - Does not currently support East-West traffic.
- Transparently intercepts traffic
 - Requires no Azure routing changes
- Traffic received by the GWLB is load-balanced between backend pool devices
 - Uses VXLAN over UDP
 - Always load balances all ports and protocols

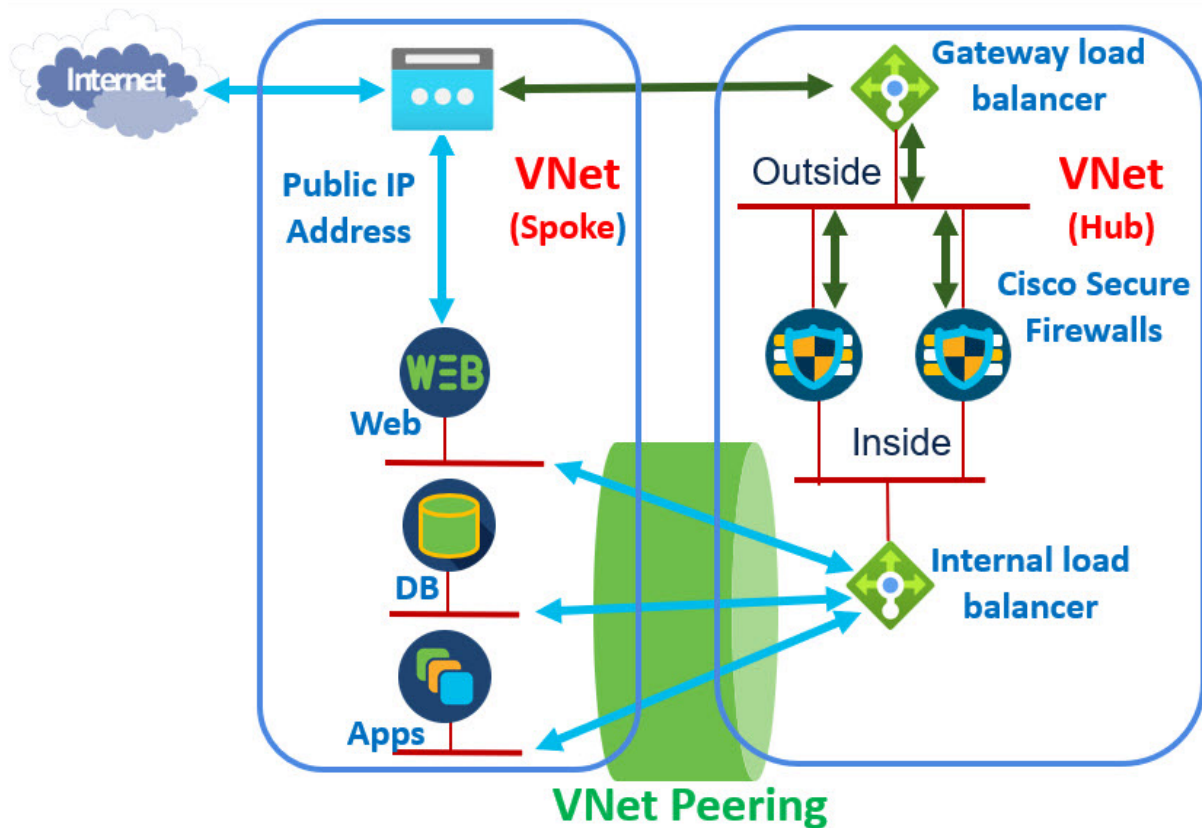


Packet Flow in Azure GWLB with Public Load Balancer



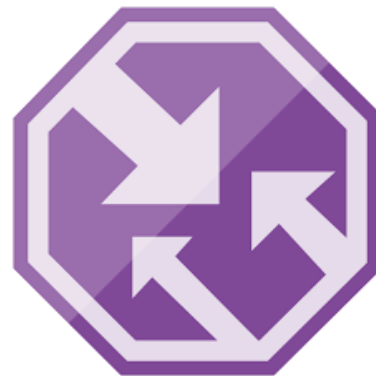
- ① Inbound flow uses public IP of public load balancer.
- ② Flow is forwarded transparently from the public load balancer to the gateway load balancer.
- ③ Flow is inspected by a firewall and returned to the gateway load balancer.
- ④ Flow is returned to the public load balancer.
- ⑤ Flow is forwarded to an internal server.

Combining Azure GWLB and ILB



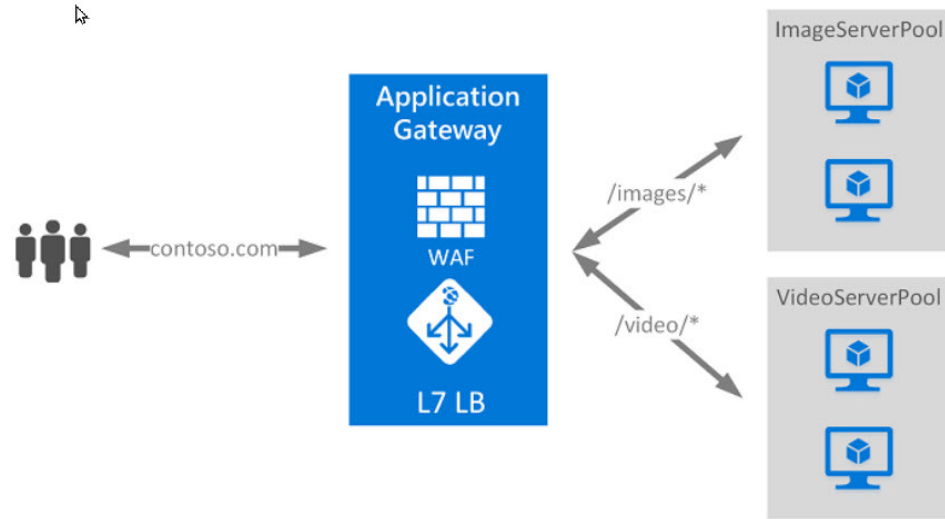
DNS Load Balancers

- Works by updating DNS
 - Modifies A/AAAA records or CNAME records
 - Data path does not go through separate entity
- Ideal for load balancing remote access VPN
- Examples:
 - AWS Route53
 - Azure Traffic Manager



Application Load Balancers (ALBs)

- Uses layer 7 information to load balance traffic
- Key use case is HTTP/HTTPS
 - Can direct different URLs (with the same FQDN) to different backend devices
 - Can direct requests to different backend devices based on media type
 - Can terminate TLS connects, offloading encryption/decryption from backend servers



Note: ALBs are not generally used in front of security solutions

Autoscale

- Security solutions (static) scaling can be expensive and wasteful
- To address this, many security solutions use autoscaling (adding and removing instances based on load)
- Typically used behind NLBs and GPBs (and ALBs for applications)
- Often utilize cloud native capabilities
 - AWS: auto scaling groups, lambda functions
 - Azure: virtual machine scale sets (VMSS), logic apps, function apps
 - GCP: instance group, Google functions
- Autoscale also can provide resilience as unhealthy nodes can be replaced

High Availability and Resilience



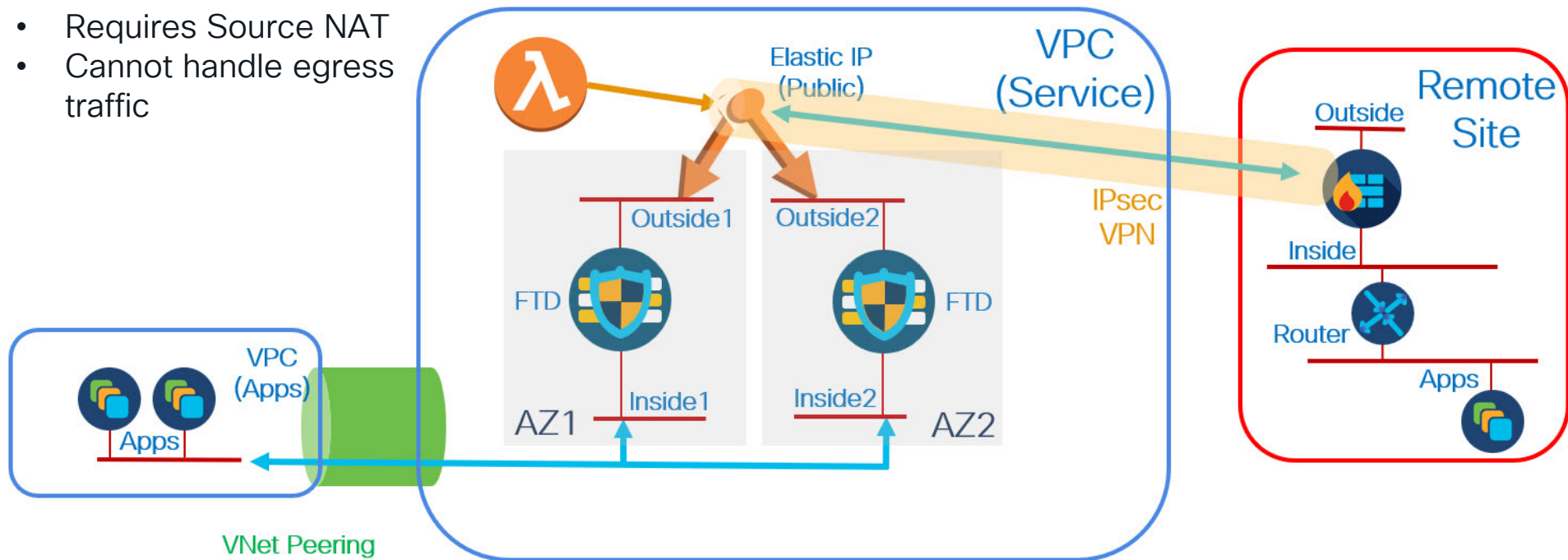
High Availability in the Public Cloud

- Approaches
 - Using load balancer health checks (usually active-active)
 - Using public cloud serverless infrastructure (usually active-standby)
 - AWS Lambda Layer
 - Azure Function Apps
 - GCP Functions
- Key limitations
 - Lack of layer two semantics -- cannot advertise new address with ARP
 - IP addresses are often pinned to the network interface objects
 - Time required for load balancers to detect unhealthy hosts is too long
 - Time required for API calls to public cloud providers is too long

Demonstration: IPsec Tunnel Failover Using AWS Lambda Functions

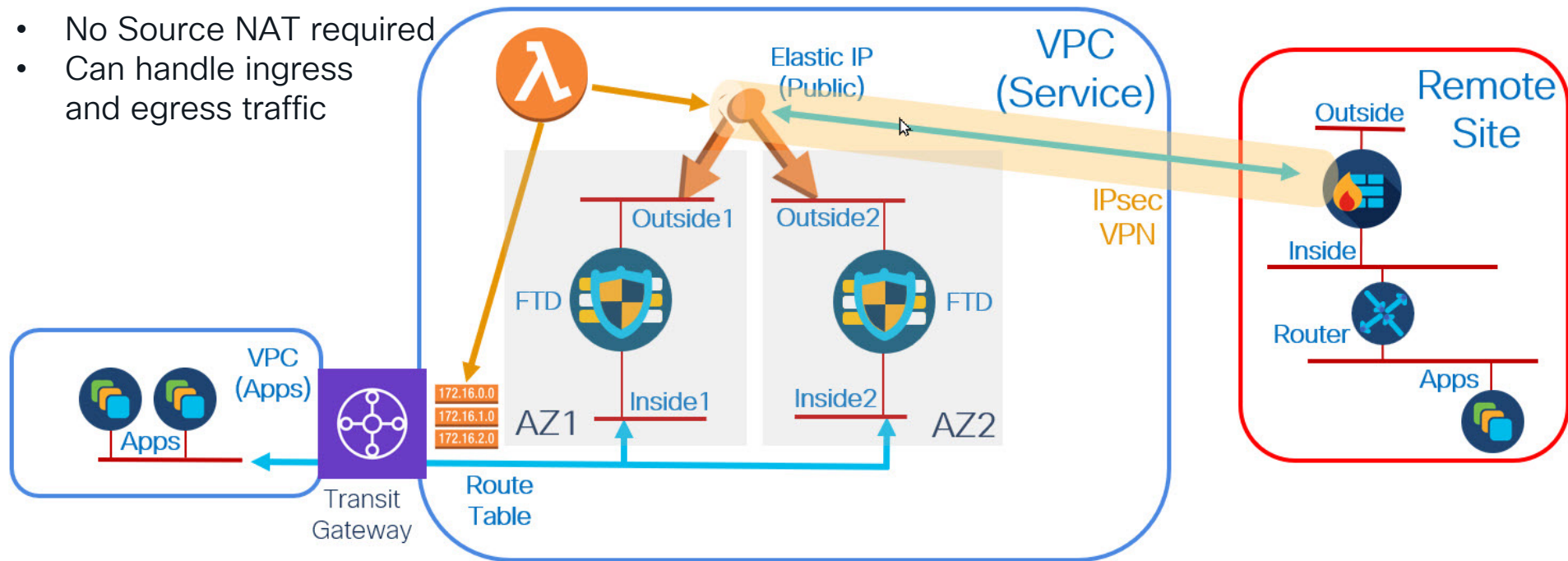
IPsec HA Scenario 1 – VPC Peering

- Requires Source NAT
- Cannot handle egress traffic

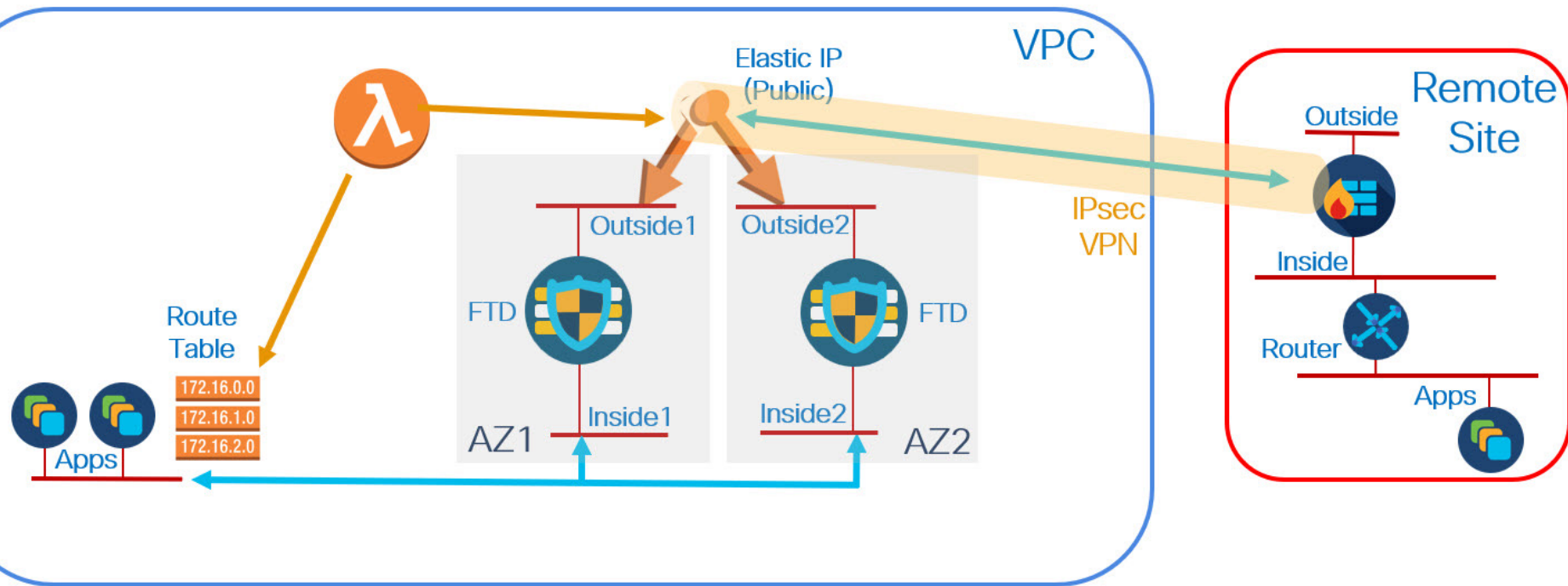


IPsec HA Scenario 2 – Transit Gateway

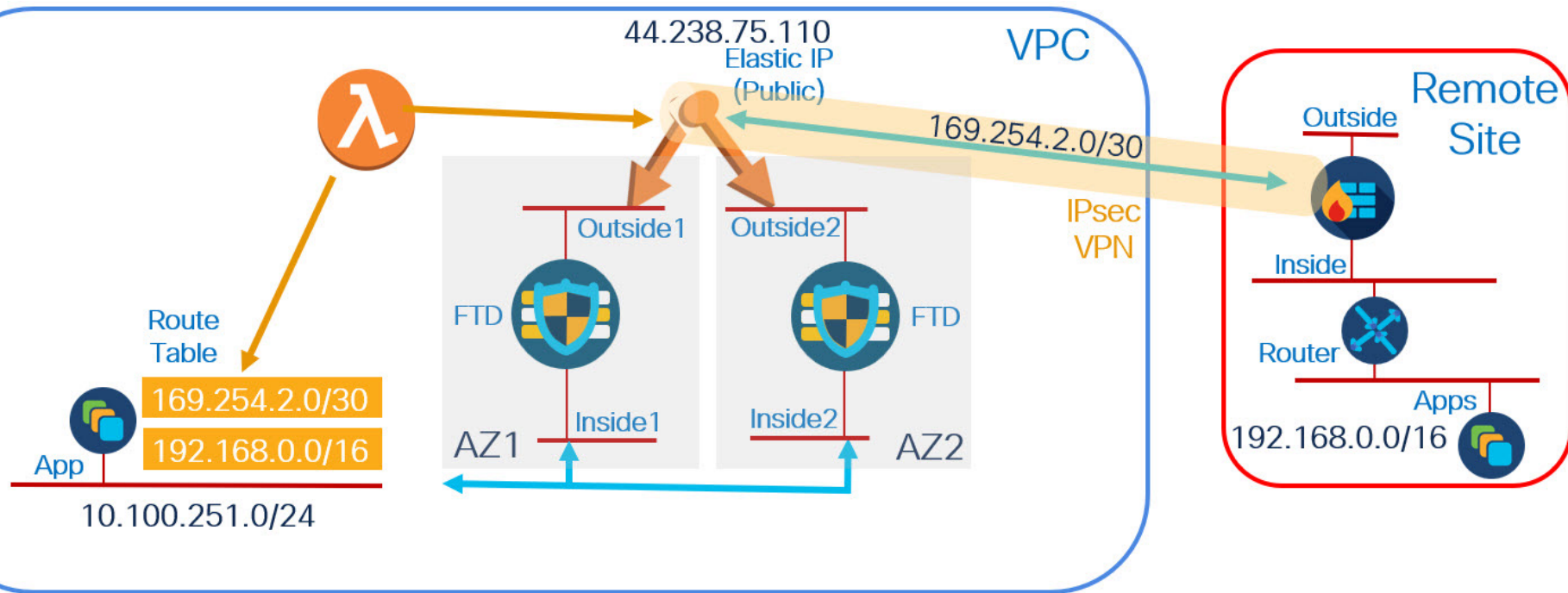
- No Source NAT required
- Can handle ingress and egress traffic



IPsec HA Scenario 3 – Simplified Test Deployment



IPsec HA Scenario 3 – Demo Deployment



Lambda Function

▼ Function overview [Info](#)

Export to Application ComposerDownload function

DiagramTemplate

 EKprobeFTDs

 Layers (0)

 EventBridge (CloudWatch Events)

+ Add trigger

+ Add destination

Description

-

Last modified

5 months ago

Function ARN

 arn:aws:lambda:us-west-2:399918941163:function:EKprobeFTDs

Function URL [Info](#)

-

Lambda Function Trigger

Triggers (1) [Info](#)



Fix errors

Edit

Delete

Add trigger

Find triggers

< 1 >



Trigger



EventBridge (CloudWatch Events): [EKrateTrigger](#)

arn:aws:events:us-west-2:399918941163:rule/EKrateTrigger

Rule state: **ENABLED**

▼ Details

Description: **Trigger every ten minutes**

Event bus: **default**

name: **EKrateTrigger**

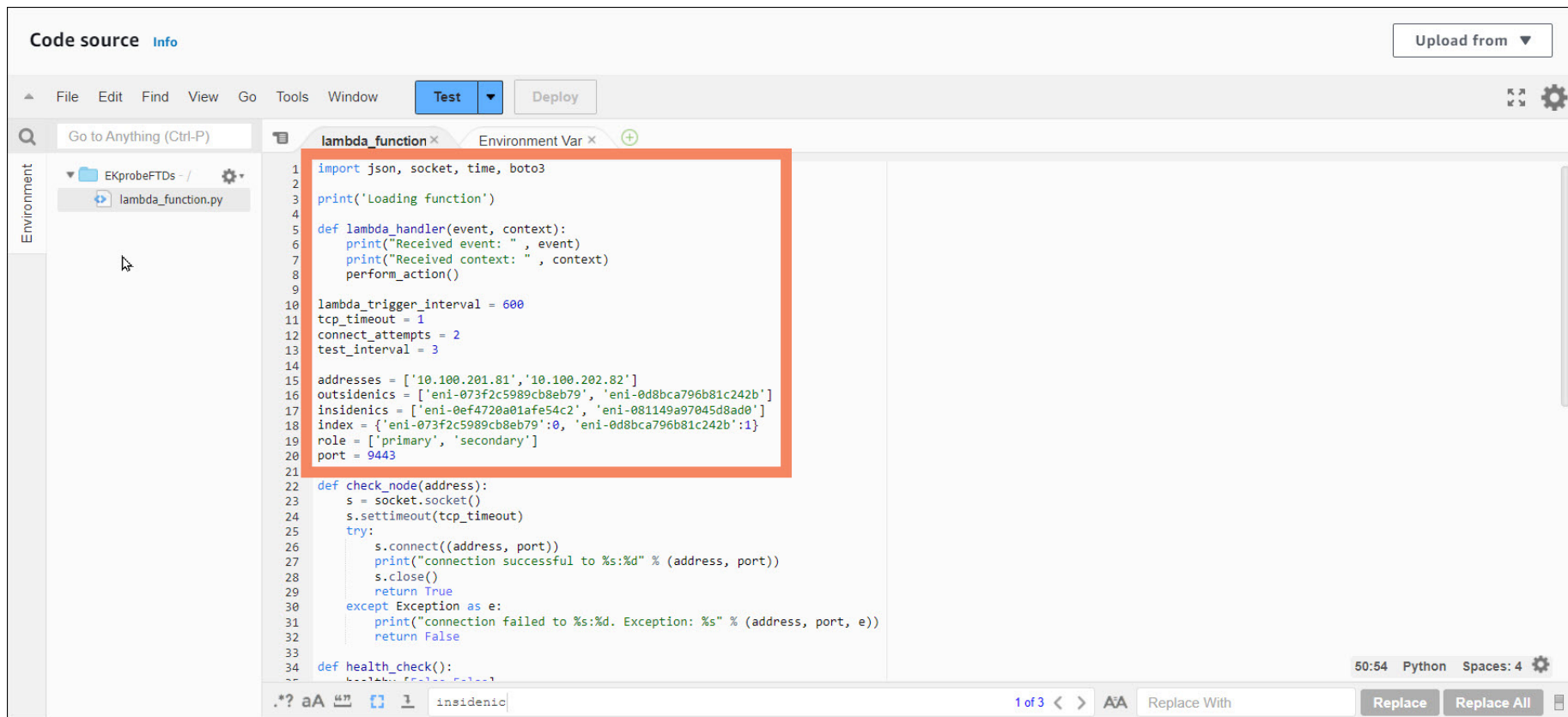
Schedule expression: **rate(10 minutes)**

Service principal: **events.amazonaws.com**

Statement ID: **lambda-6552cd2a-edab-4f25-b1ae-208812ca99dc**

url: **events/home#/rules/EKrateTrigger**

Lambda Function Code – Slide 1 of 3



The screenshot shows a code editor with a file named `lambda_function.py` open. The code is a Python lambda function for health checking. A red rectangular box highlights the following code block:

```
1 import json, socket, time, boto3
2
3 print('Loading function')
4
5 def lambda_handler(event, context):
6     print("Received event: " , event)
7     print("Received context: " , context)
8     perform_action()
9
10 lambda_trigger_interval = 600
11 tcp_timeout = 1
12 connect_attempts = 2
13 test_interval = 3
14
15 addresses = ['10.100.201.81', '10.100.202.82']
16 outsiders = ['eni-073f2c5989cb8eb79', 'eni-0d8bca796b81c242b']
17 insiders = ['eni-0ef4720a01afe54c2', 'eni-081149a97045d8ad0']
18 index = {'eni-073f2c5989cb8eb79':0, 'eni-0d8bca796b81c242b':1}
19 role = ['primary', 'secondary']
20 port = 9443
```

The rest of the code in the editor includes a `check_node` function that attempts to connect to the specified addresses and a `health_check` function.

Lambda Function Code – Slide 2 of 3

Code source [Info](#) Upload from ▼

File Edit Find View Go Tools Window Test ▼ Deploy

Go to Anything (Ctrl-P)

Environment

Environment Var x +

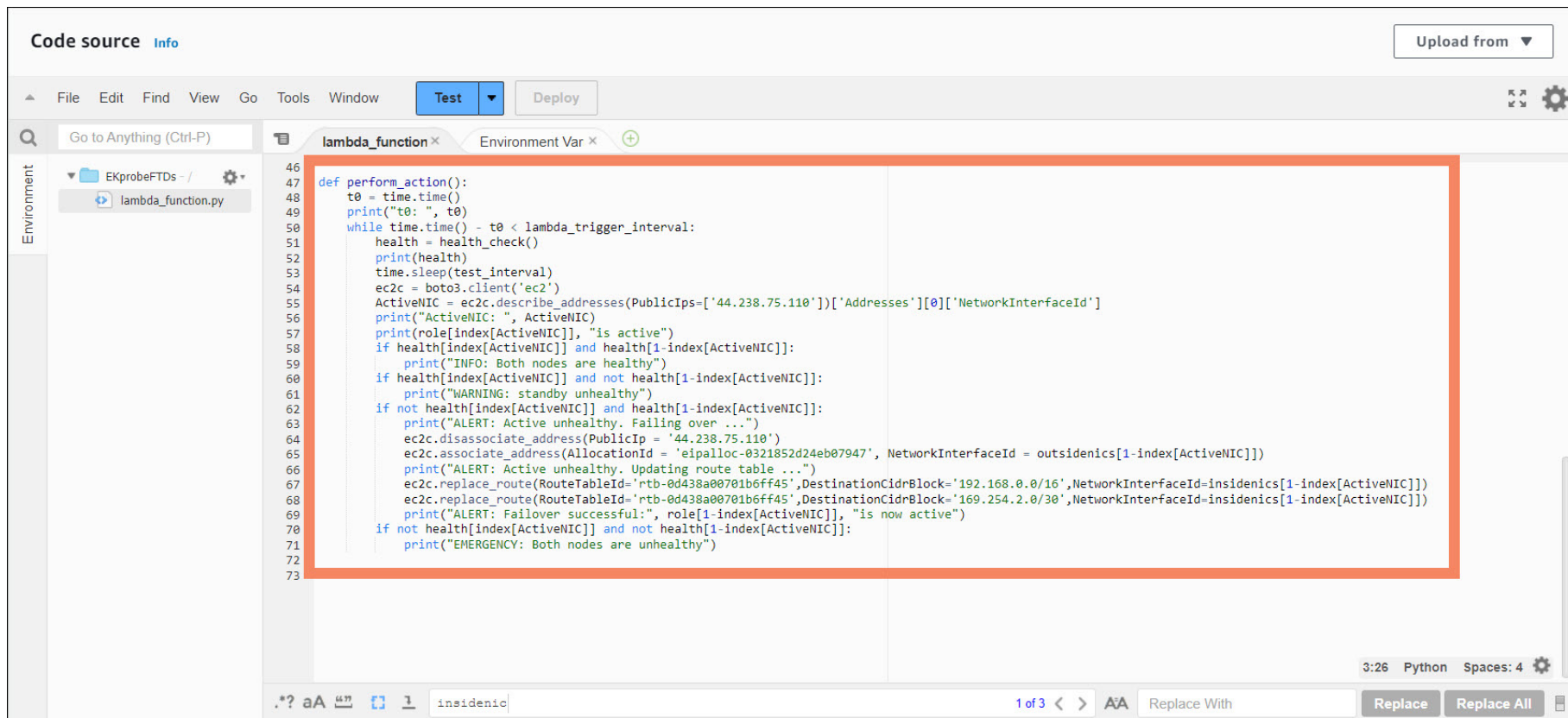
lamba_function x

```
22 def check_node(address):
23     s = socket.socket()
24     s.settimeout(tcp_timeout)
25     try:
26         s.connect((address, port))
27         print("connection successful to %s:%d" % (address, port))
28         s.close()
29         return True
30     except Exception as e:
31         print("connection failed to %s:%d. Exception: %s" % (address, port, e))
32         return False
33
34 def health_check():
35     healthy=[False,False]
36     for i in range(connect_attempts):
37         if not healthy[0]:
38             healthy[0] = check_node(addresses[0])
39         if not healthy[1]:
40             healthy[1] = check_node(addresses[1])
41     if not healthy[0]:
42         print("health check failed %d times: host %s marked unhealthy" % (connect_attempts, addresses[0]))
43     if not healthy[1]:
44         print("health check failed %d times: host %s marked unhealthy" % (connect_attempts, addresses[1]))
45     return healthy
46
47 def perform_action():
48     t0 = time.time()
49     print("t0: ", t0)
50     while time.time() - t0 < lambda_trigger_interval:
51         health = health_check()
52         print(health)
53         time.sleep(test_interval)
54         ec2c = boto3.client('ec2')
55         ActiveNIC = ec2c.describe_addresses(PublicIps=['44.238.75.110'])[0]['Addresses'][0]['NetworkInterfaceId']
```

67:106 Python Spaces: 4

insidenic 1 of 3 < > AA Replace With Replace Replace All

Lambda Function Code – Slide 3 of 3



The screenshot shows a code editor interface with a menu bar (File, Edit, Find, View, Go, Tools, Window), a toolbar with 'Test' and 'Deploy' buttons, and a search bar. The left sidebar shows the 'Environment' with a folder 'EKprobeFTDs - /' and a file 'lambda_function.py'. The main editor area displays the code for 'lambda_function.py', which is highlighted with an orange border. The code defines a 'perform_action()' function that checks the health of a network interface and performs failover actions if it becomes unhealthy. The status bar at the bottom shows '3:26 Python Spaces: 4' and a search bar with the text 'insidenic'.

```
46
47 def perform_action():
48     t0 = time.time()
49     print("t0: ", t0)
50     while time.time() - t0 < lambda_trigger_interval:
51         health = health_check()
52         print(health)
53         time.sleep(test_interval)
54         ec2c = boto3.client('ec2')
55         ActiveNIC = ec2c.describe_addresses(PublicIps=['44.238.75.110'])['Addresses'][0]['NetworkInterfaceId']
56         print("ActiveNIC: ", ActiveNIC)
57         print(role[index[ActiveNIC]], "is active")
58         if health[index[ActiveNIC]] and health[1-index[ActiveNIC]]:
59             print("INFO: Both nodes are healthy")
60         if health[index[ActiveNIC]] and not health[1-index[ActiveNIC]]:
61             print("WARNING: standby unhealthy")
62         if not health[index[ActiveNIC]] and health[1-index[ActiveNIC]]:
63             print("ALERT: Active unhealthy. Failing over ...")
64             ec2c.disassociate_address(PublicIp = '44.238.75.110')
65             ec2c.associate_address(AllocationId = 'eipalloc-0321852d24eb07947', NetworkInterfaceId = outsidenics[1-index[ActiveNIC]])
66             print("ALERT: Active unhealthy. Updating route table ...")
67             ec2c.replace_route(RouteTableId='rtb-0d438a00701b6ff45', DestinationCidrBlock='192.168.0.0/16', NetworkInterfaceId=insidenics[1-index[ActiveNIC]])
68             ec2c.replace_route(RouteTableId='rtb-0d438a00701b6ff45', DestinationCidrBlock='169.254.2.0/30', NetworkInterfaceId=insidenics[1-index[ActiveNIC]])
69             print("ALERT: Failover successful:", role[1-index[ActiveNIC]], "is now active")
70         if not health[index[ActiveNIC]] and not health[1-index[ActiveNIC]]:
71             print("EMERGENCY: Both nodes are unhealthy")
72
73
```

Last Updated: 01:51 PM

Refresh

+ Site to Site VPN

+ SASE Topology

▼ Select... ×

Refresh

Topology Name	VPN Type	Network Topology	Tunnel Status Distribution	IKEv1	IKEv2
> FTD81VPN	Route Based (VTI)	Point to Point	<div><div></div>1 - Tunnels</div>	✓	 
> FTD82VPN	Route Based (VTI)	Point to Point	<div><div></div>1 - Tunnels</div>	✓	 

Target Failover in AWS

- Moves existing connections from unhealthy nodes to healthy nodes.
- Only available for Gateway Load Balancers, disabled by default
- To configure this setting with the AWS CLI, use the **modify-target-group-attributes** command with the following key value pairs:
 - Key=**target_failover.on_deregistration** and Value= **no_rebalance** (default) or **rebalance**
 - Key=**target_failover.on_unhealthy** and Value= **no_rebalance** (default) or **rebalance**
- These attributes are also available in the AWS Cloud Formation Templates.

EC2 > Target groups > inspection-vpc-gwlb-tg-1 > Edit attributes

Edit attributes

Attributes Restore defaults

Deregistration delay Info
The time to wait for in-flight requests to complete while deregistering a target. During this time, the state of the target is draining.
 seconds
0-3600

Target failover — recommended Info
By default, the Gateway Load Balancer continues to send existing flows to targets, even if they have failed or are deregistered. You can use the toggle switch to rebalance existing flows to healthy targets.

☒ Rebalance flows

Flow stickiness Info
By default, the Gateway Load Balancer uses 5-tuple to maintain flow stickiness to a specific target appliance. You can modify the stickiness type and customize it to 3-tuple or 2-tuple.

☒ 5-tuple
Source IP, Source Port, Destination IP, Destination Port and Transport Protocol

☐ Customize flow stickiness
Allows you to customize the stickiness type to 3-tuple or 2-tuple.

Cancel Save changes

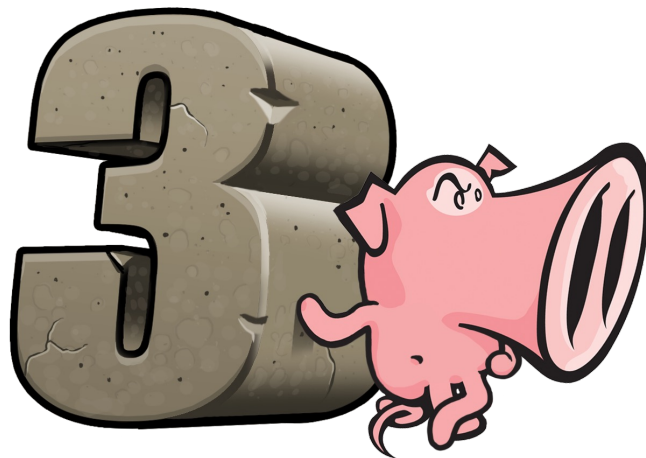
Target Failover in Azure and GCP

- Target Failover in Azure
 - Currently Azure GWLBs and NLBs do not perform target failover.
 - Even when a target becomes unhealthy, the load balancers forward the remaining packets of existing flows to the same host.
- Target Failover in GCP
 - Called connection persistence. This is a legacy feature in all GCP load balancers.
 - DEFAULT_FOR_PROTOCOL (default). Move existing UDP a healthy target. Keep (persist) sending TCP packets to the unhealthy target.
 - NEVER_PERSIST. Move existing UDP and TCP connections to a healthy target.
 - ALWAYS_PERSIST. Keep (persist) sending UDP and TCP packets to the unhealthy target.

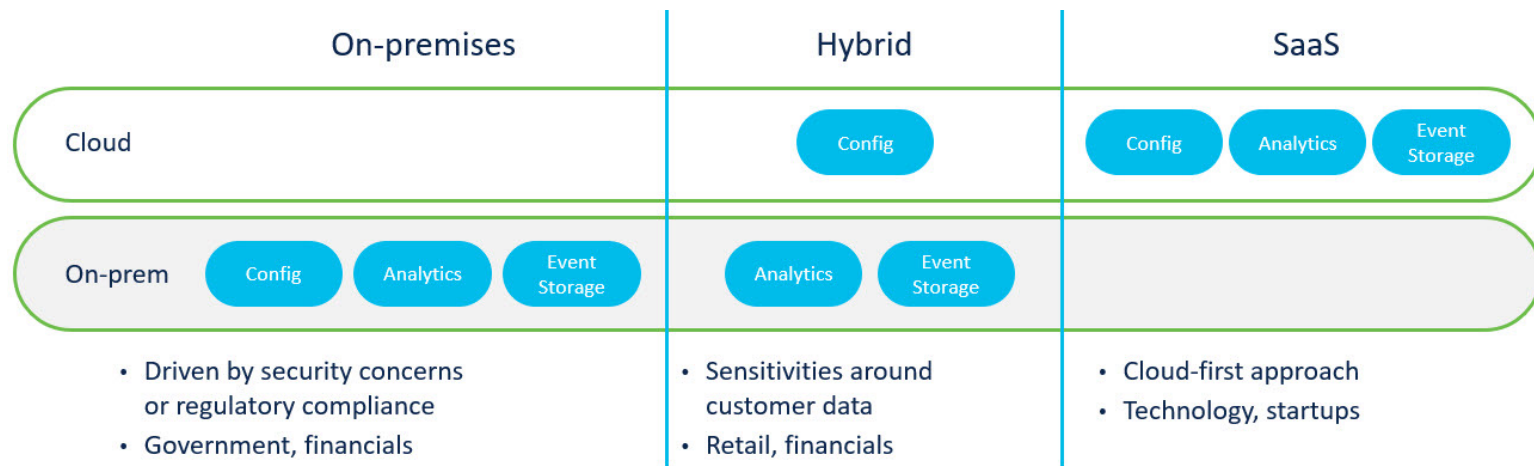
Cisco Secure Firewall

Cisco Secure Firewall Overview

- Stateful Firewall
 - Layer 1 (inline pairs/inline sets)
 - Layer 2 (bridge groups/BVIs)
 - Layer 3 and 4 (routed interfaces)
- NGIPS – Next-Generation Intrusion Prevention System
- AMP – Advanced Malware Protection
- VPN – Virtual Private Network
- URL – URL filtering
- SI – Security Intelligence



Firewall Management



- On Premises: Firewall Management Center (FMC)
- SaaS: Cisco Defense Orchestrator (CDO) with built-in cloud delivered FMC

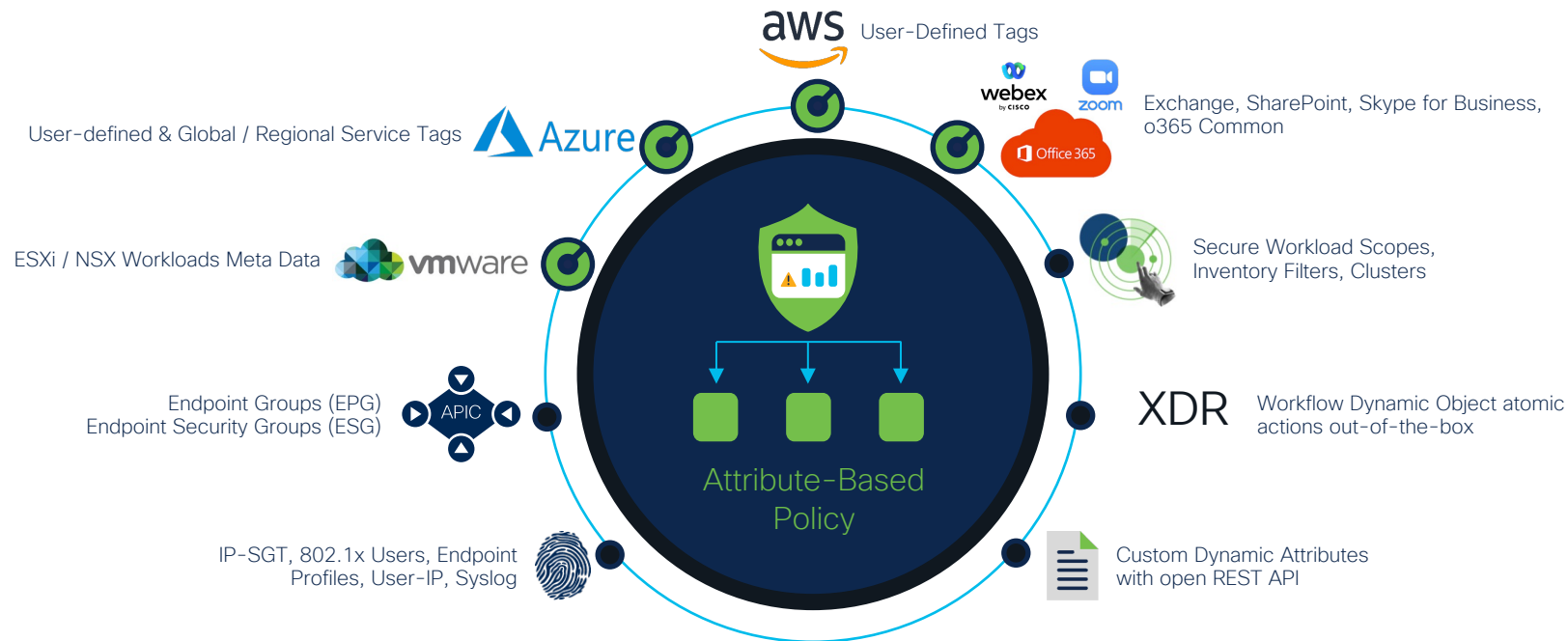
Cisco Secure Firewall in the Multi-cloud



- ▶ Clustering
- ▶ Dynamic Policy
- ▶ Better integration with public cloud infrastructure
- ▶ Infrastructure as Code and Automation

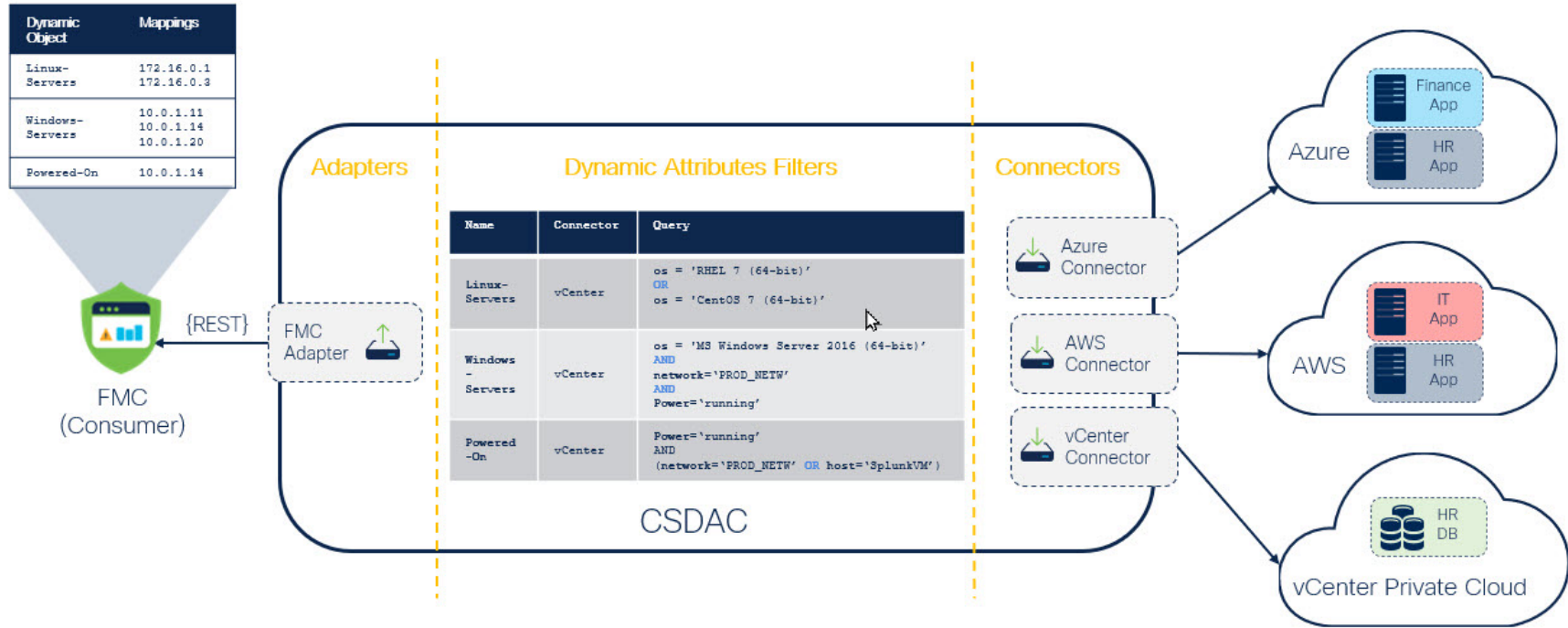
- ▶ Accelerated Networking
- ▶ Gateway Load balancer integration
- ▶ Auto Scaling
- ▶ Snapshot support

Dynamic Attributes and Security Policies



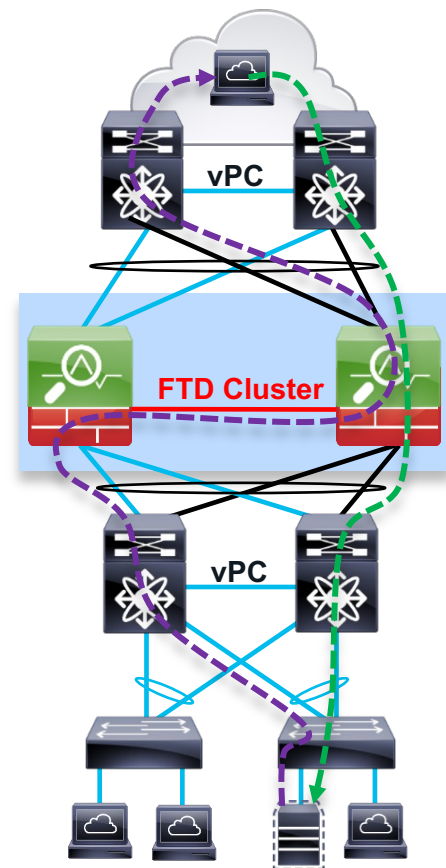
Cisco Secure Dynamic Attribute Connector (CSDAC)

Stand-alone VM, SaaS offering, or Integrated into on-premises FMC



Clustering Cisco Secure Firewall

- Combines multiple Cisco Secure Firewalls into a single logical device
 - Allows pay as you grow sizing – nodes can be added to existing clusters
 - Up to 16 nodes, depending on platform
- *May* provide resilience – dependent on external load balancing
 - Physical devices can use EtherChannel with LACP to provide resilience.
 - Virtual devices require load balancers capable of connection rehashing.
- Originally to cluster physical Cisco ASA firewall
 - Later extended to Cisco Secure Firewall Threat Defense (FTD)
 - Recently extended to virtual devices



Porting Cisco Clustering to the Public Clouding

Physical Cluster

- Data interfaces have two modes.
 - Individual interface mode – different IP addresses on different nodes (rarely used)
 - Spanned interface mode with EtherChannel
- CCL uses proprietary protocol over IP.
 - Increase MTU by 100 Bytes for cluster link.
- CCL uses broadcast for internode communication.
 - Uses dynamic node discovery

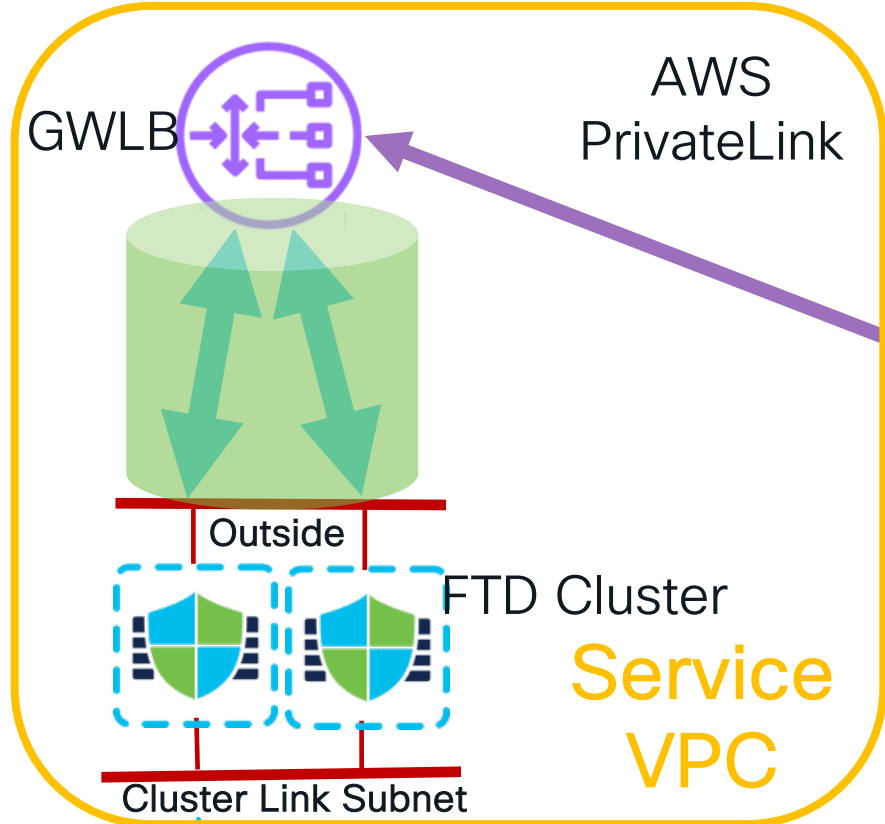
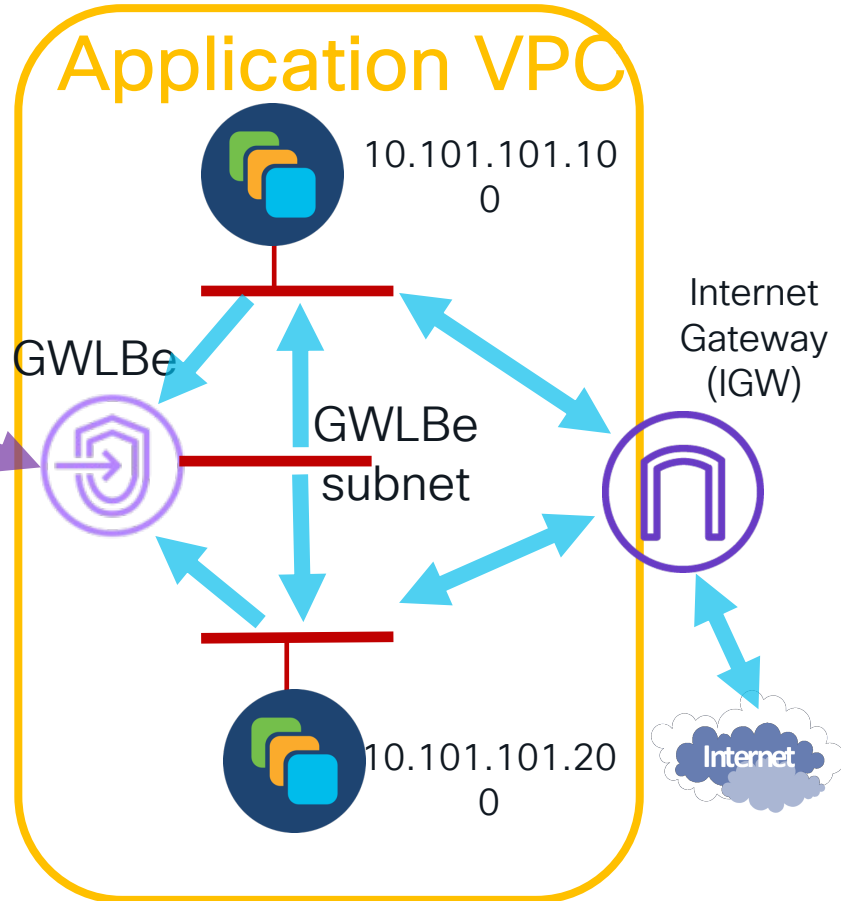
Virtual Cluster

- Data interfaces on each node use different IP addresses.
- CCL uses VXLAN over UDP.
 - Increase MTU by 160 Bytes for cluster link.
- CCL uses unicast for internode communication.
 - Requires static peer list

Demonstration: AWS Target Failover with Cisco Secure Firewall Cluster

Demonstration Topology

Inspect Intra-VPC traffic



Health Probe Settings

Defaults:

- Healthy threshold = 5
- Unhealthy threshold = 2
- Timeout = 5
- Interval = 10

Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

TCP

▼ Advanced health check settings

Restore defaults

Health check port

The port the load balancer uses when performing health checks on targets. By default, the health check port is the same as the target group's traffic port. However, you can specify a different port as an override.

☐ Traffic port

☒ Override

9443

1-65535

Healthy threshold

The number of consecutive health checks successes required before considering an unhealthy target healthy.

2

2-10

Unhealthy threshold

The number of consecutive health check failures required before considering a target unhealthy.

2

2-10

Timeout

The amount of time, in seconds, during which no response means a failed health check.

2

seconds

2-120

Interval

The approximate amount of time between health checks of an individual target

5

seconds

5-300

Details

arn:aws:elasticloadbalancing:us-west-2:399918941163:targetgroup/FTDvCluster/00d56815e5d166aaed

Target type	Protocol : Port	VPC	Load balancer
IP	GENEVE: 6081	vpc-0ce26a614c1c149d1 	FTDvClusterGLB 

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
2	 2	 0	 0	 0	 0

► **Distribution of targets by Availability Zone (AZ)**

Select values in this table to see corresponding filters applied to the Registered targets table below.

Targets

Monitoring

Health checks

Attributes

Tags

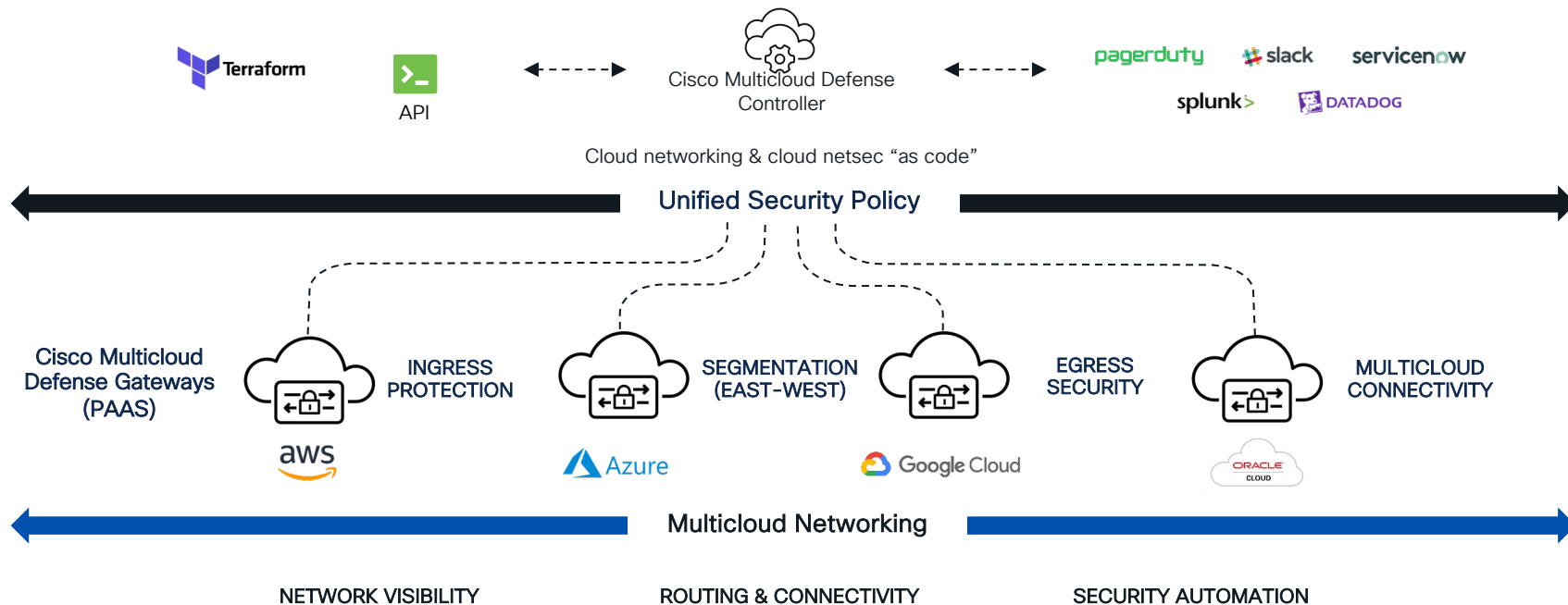
Health check settings

Edit

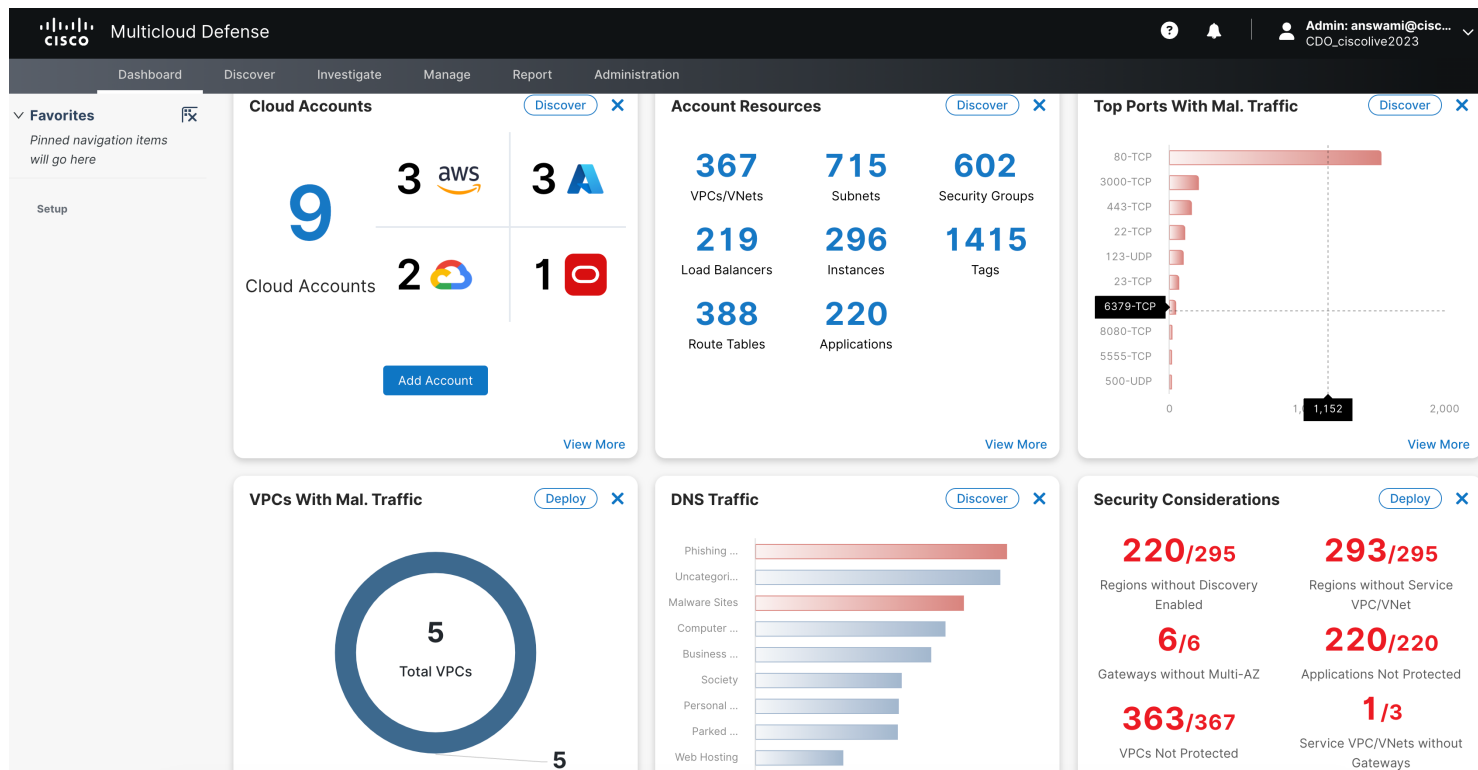
Protocol	Port	Healthy threshold	Unhealthy threshold
TCP	9443	2 consecutive health check successes	2 consecutive health check failures
Timeout	Interval		
2 seconds	5 seconds		

Cisco Multicloud Defense

Cisco Multicloud Defense



Multicloud Inventory and Visibility



Cisco Multicloud Defense Gateway



Ingress Gateway

- ✓ Reverse Proxy
- ✓ TLS decrypt
- ✓ WAF - L7 DoS
- ✓ IDS / IPS
- ✓ Antivirus
- ✓ Geo IP
- ✓ Malicious IP



Egress Gateway

Egress

- ✓ URL filtering
- ✓ Forward proxy
- ✓ TLS decrypt
- ✓ FQDN filtering*
- ✓ FQDN-based firewall policy
- ✓ DLP
- ✓ IDS / IPS
- ✓ Antivirus

East/West

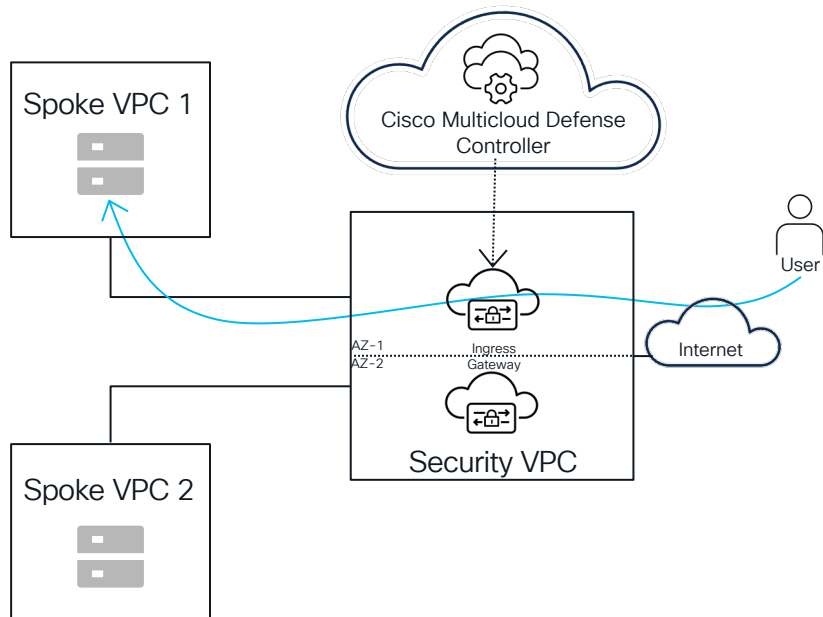
- ✓ FQDN filtering
- ✓ IPS / IDS
- ✓ Antivirus
- ✓ Micro-segmentation
- ✓ FQDN-based firewall policy
- ✓ TLS decrypt

** No TLS decryption is needed*

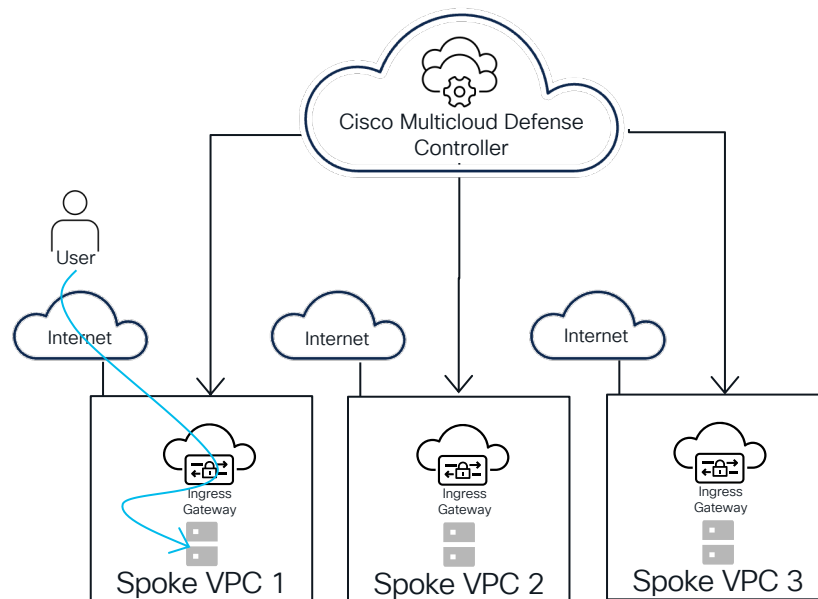
Centralized and Distributed Model

Ingress Traffic

Centralized Security Model



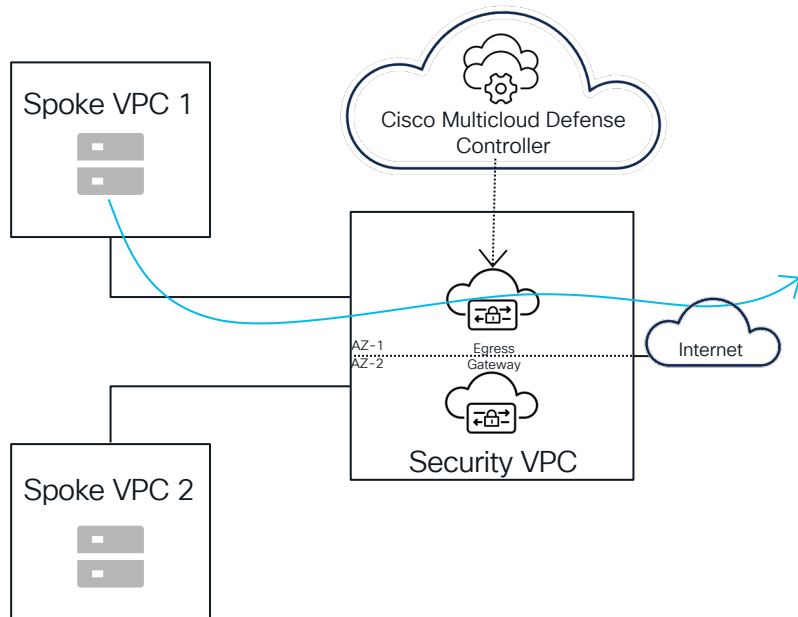
Distributed Security Model



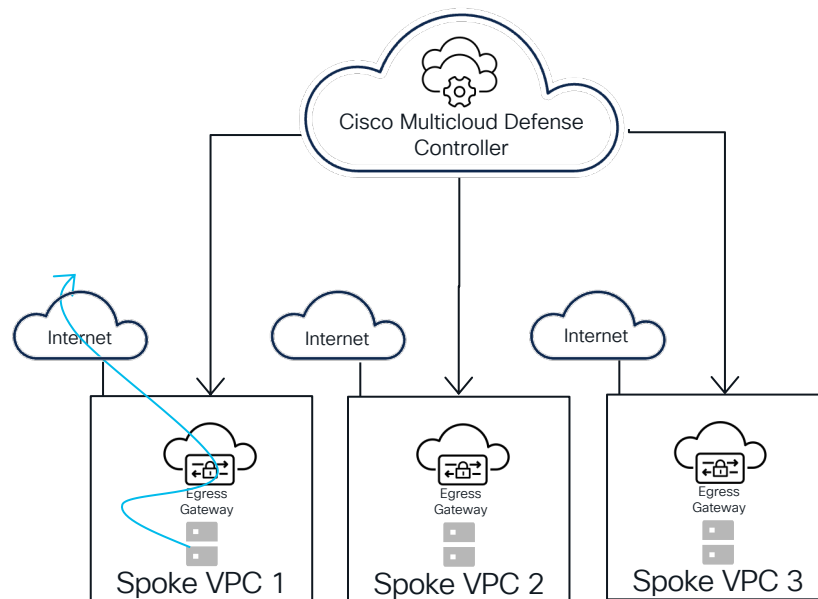
Centralized and Distributed Model

Egress Traffic to Internet

Centralized Security Model



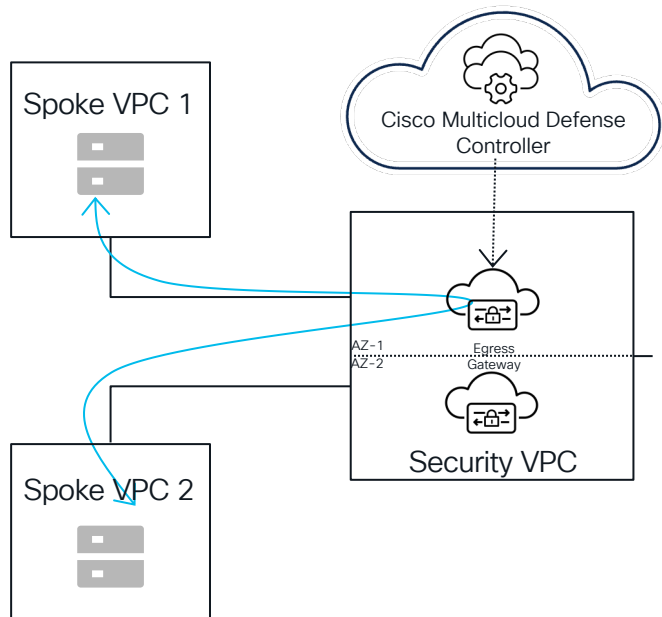
Distributed Security Model



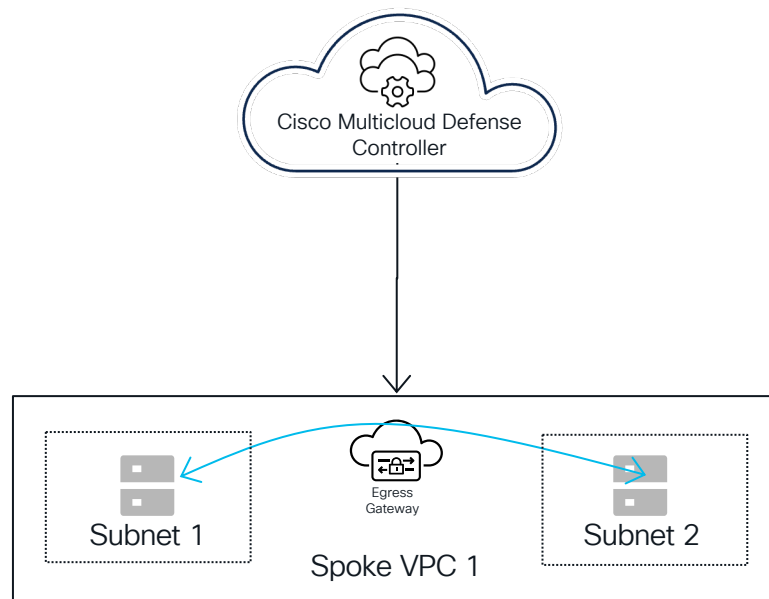
Centralized and Distributed Model

East-West Traffic through Egress Gateway

Centralized Security Model
Inter & Intra VPC



Distributed Security Model
Intra VPC



Connectivity for Centralized Model – Automation

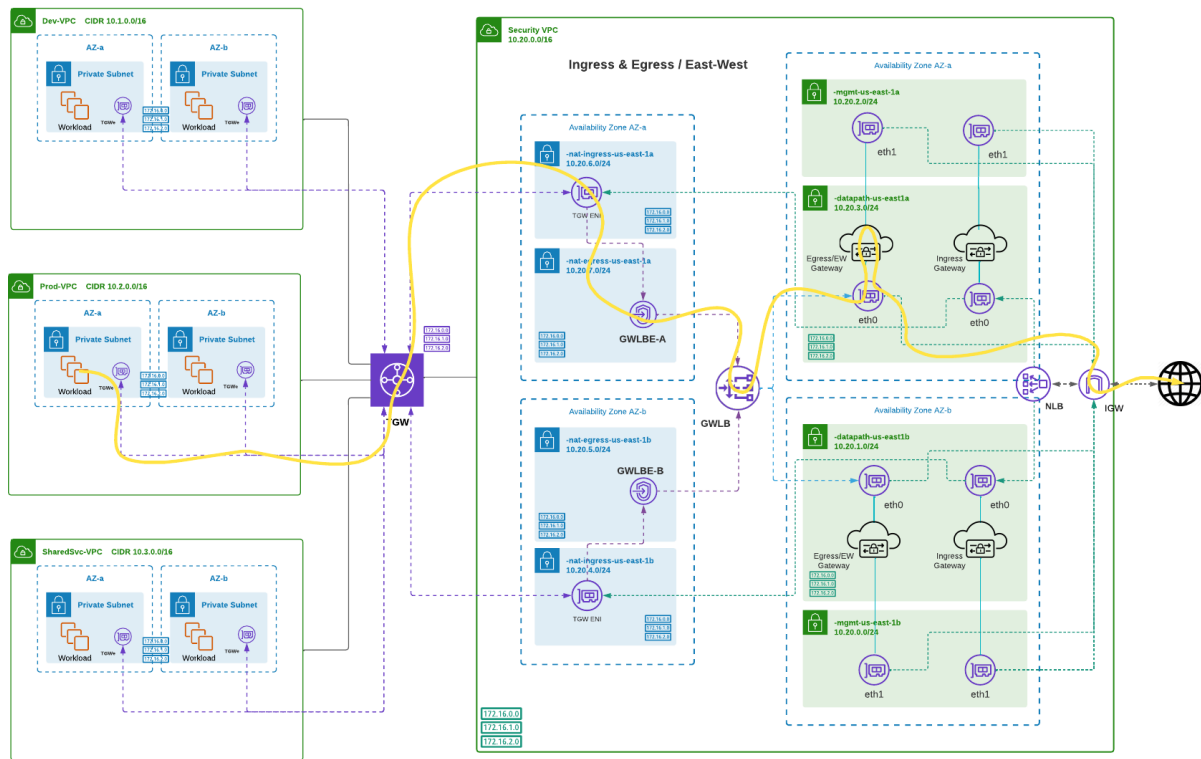
- (All clouds) Creation of Service VPC
- AWS
 - Deployment of new transit gateway
 - Integration with existing transit gateway
 - Updates VPC and TGW route tables
- Azure/GCP/OCI
 - Hub and spoke topology with VNet/VPC/VCN peering
 - Updates route tables
- (All clouds) Routing updates

Ingress



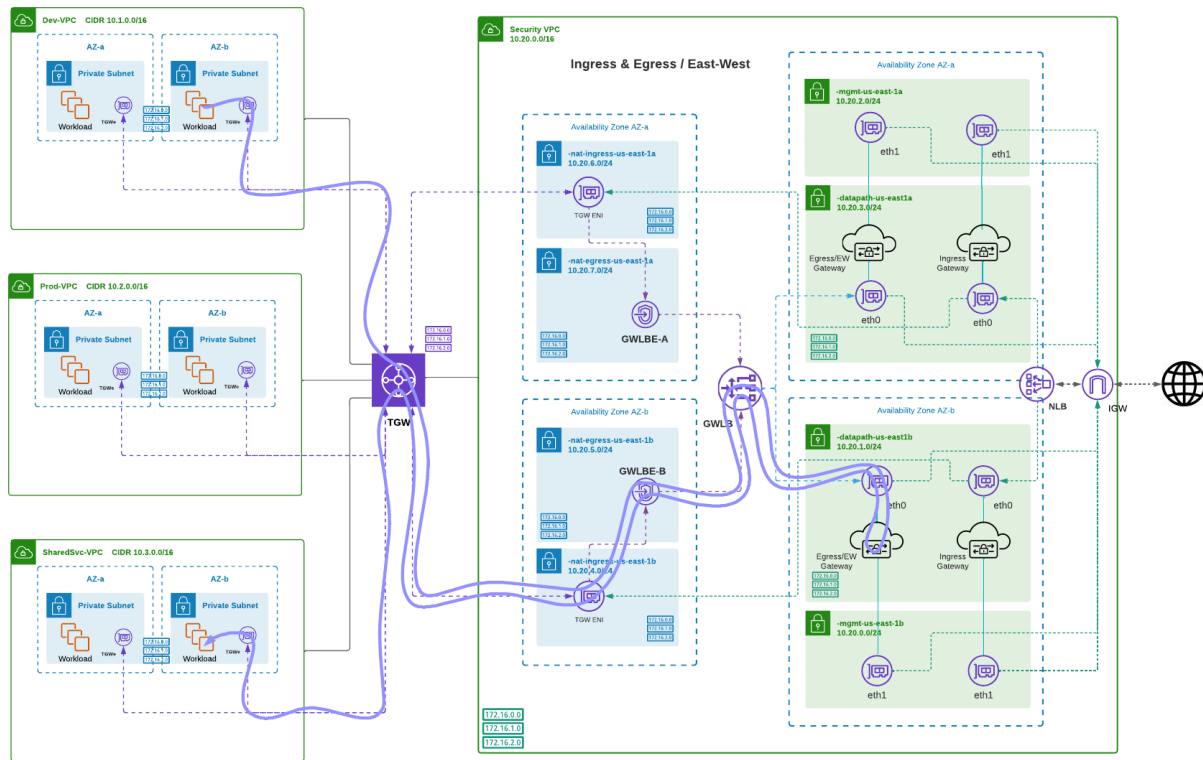
AWS Centralized Model

Egress



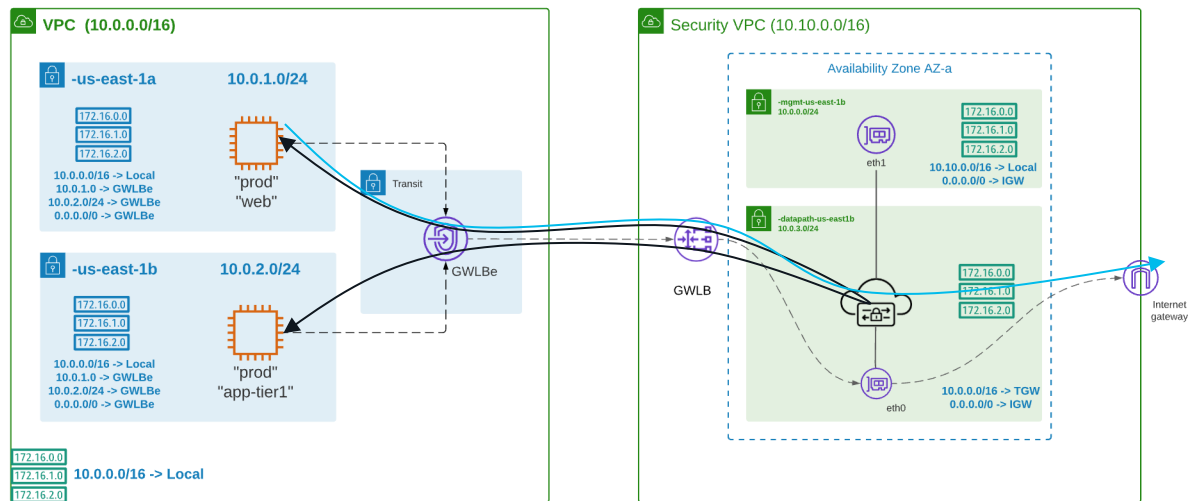
AWS Centralized Model

East-West Inter-VPC



AWS Centralized Model

East-West Intra-VPC



Scaling Intercloud Connectivity*

* Coming soon

Autoscaling

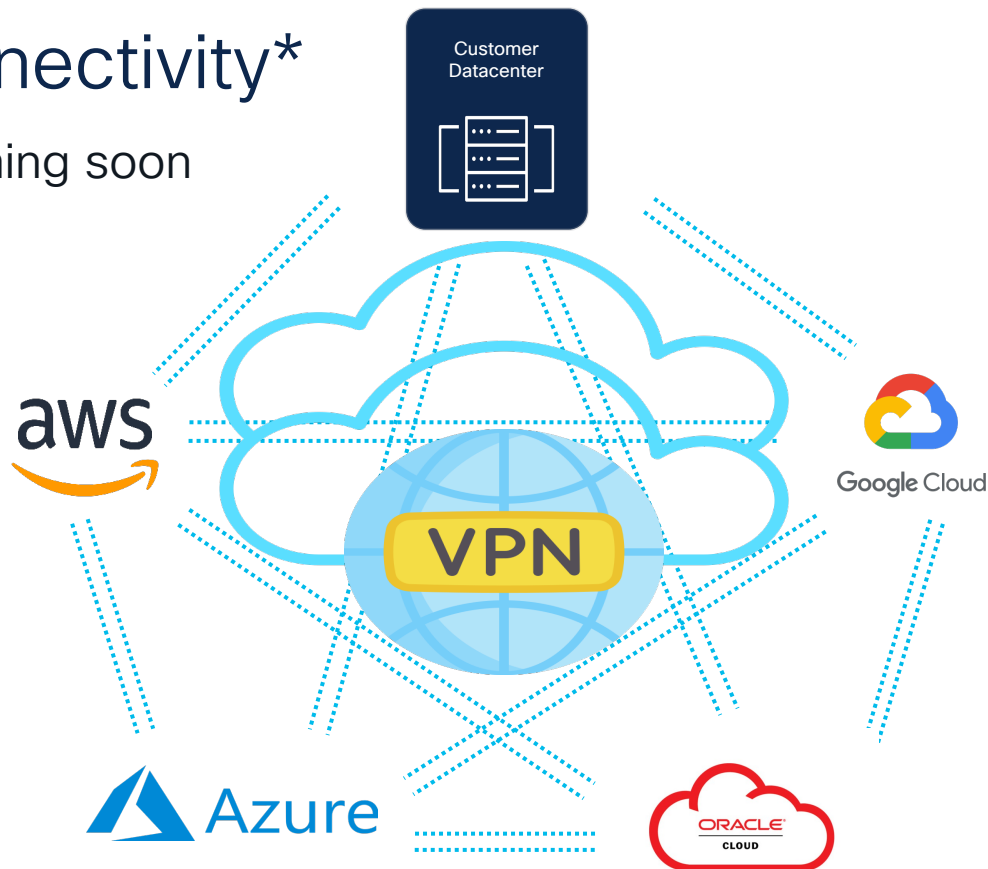
Self-Healing

Automation

Operation Simplicity

Distributed

Built-in Advanced Security

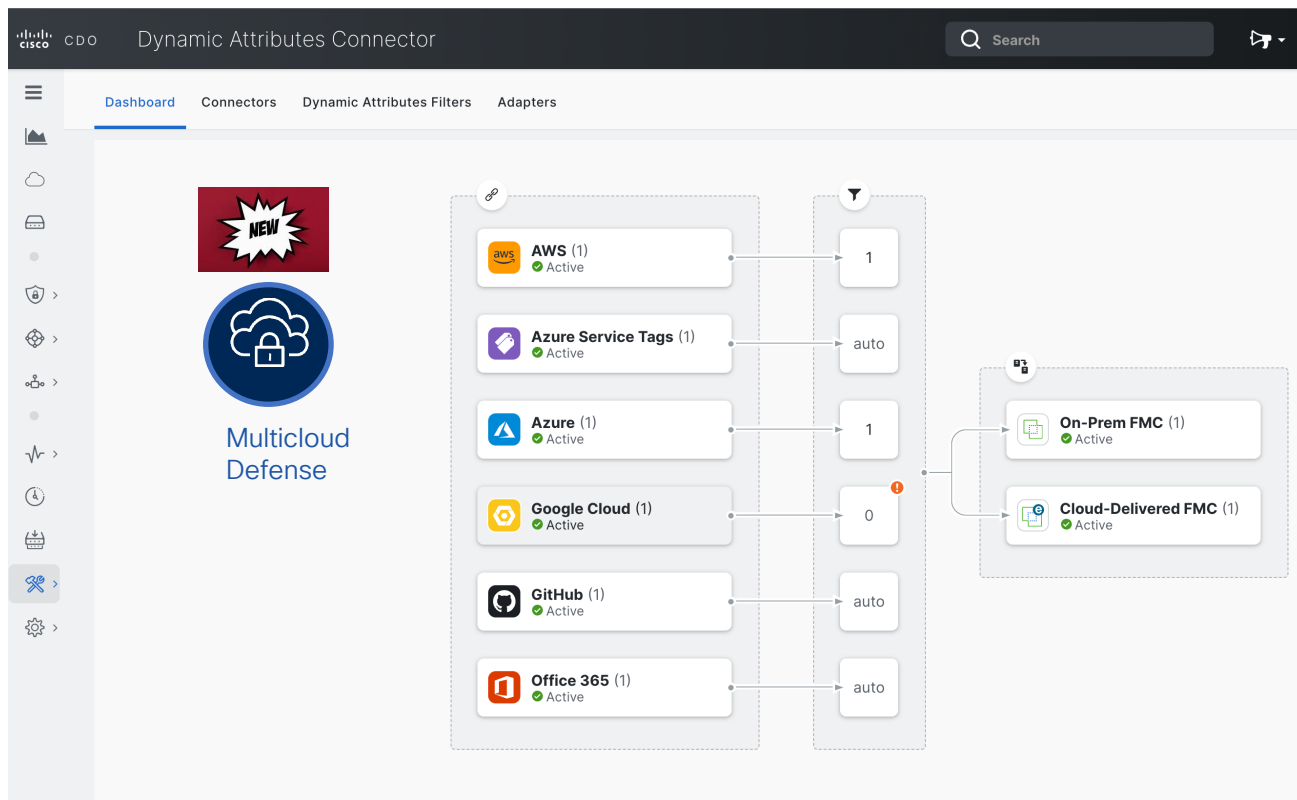


Demonstration: Cisco Multicloud Defense

Coming Soon to Multicloud Defense

- Integration with Cisco Secure Dynamic Access Connector (CSDAC)
 - Currently CSDAC only sends attributes to FMCs (and then to Cisco Secure Firewall)
- Shared Objects
 - Objects create on Cisco Secure Firewalls can be used by Multicloud Defense
- IPsec VPN
 - Multicloud Defense includes creation of site-to-site VPNs
 - Cloud to cloud
 - On-premise to cloud

Integration with CSDAC



Hide Menu

Dashboard

Multicloud Defense

Inventory

Configuration

Policies

Objects

VPN

Events & Monitoring

Analytics

Change Log

Jobs

Tools & Services

Settings

Devices

Templates

Search by Device Name, IP Address, or Serial Number

Displaying 4 of 4 results

All

ASA

FTD

☐	Name	Configuration Status	Connectivity
☐	Brisbane ASA	Synced	Online
☐	Melbourne FTD	Synced	Online
☐	Perth ASA	Synced	Online
☐	Sydney FTD	Synced	Online

Hide Menu

Secure Connect
Choice

Dashboard

Multicloud
Defense **New**

Inventory

Configuration

Policies

Objects

VPN and Zero Trust

Events & Monitoring

Analytics

Change Log

Jobs

Tools & Services

Settings

Welcome to Cisco Defense Orchestrator

Quick Actions

Inventory & Objects

+ Onboard

Connectivity States



Configuration States

1 Synced



0 Not Synced



0 Conflict Detected



Object Issues

Object status	Number of objects
All Objects	132
Inconsistent	4
Duplicate	20
Unused	56

Site-to-Site VPN

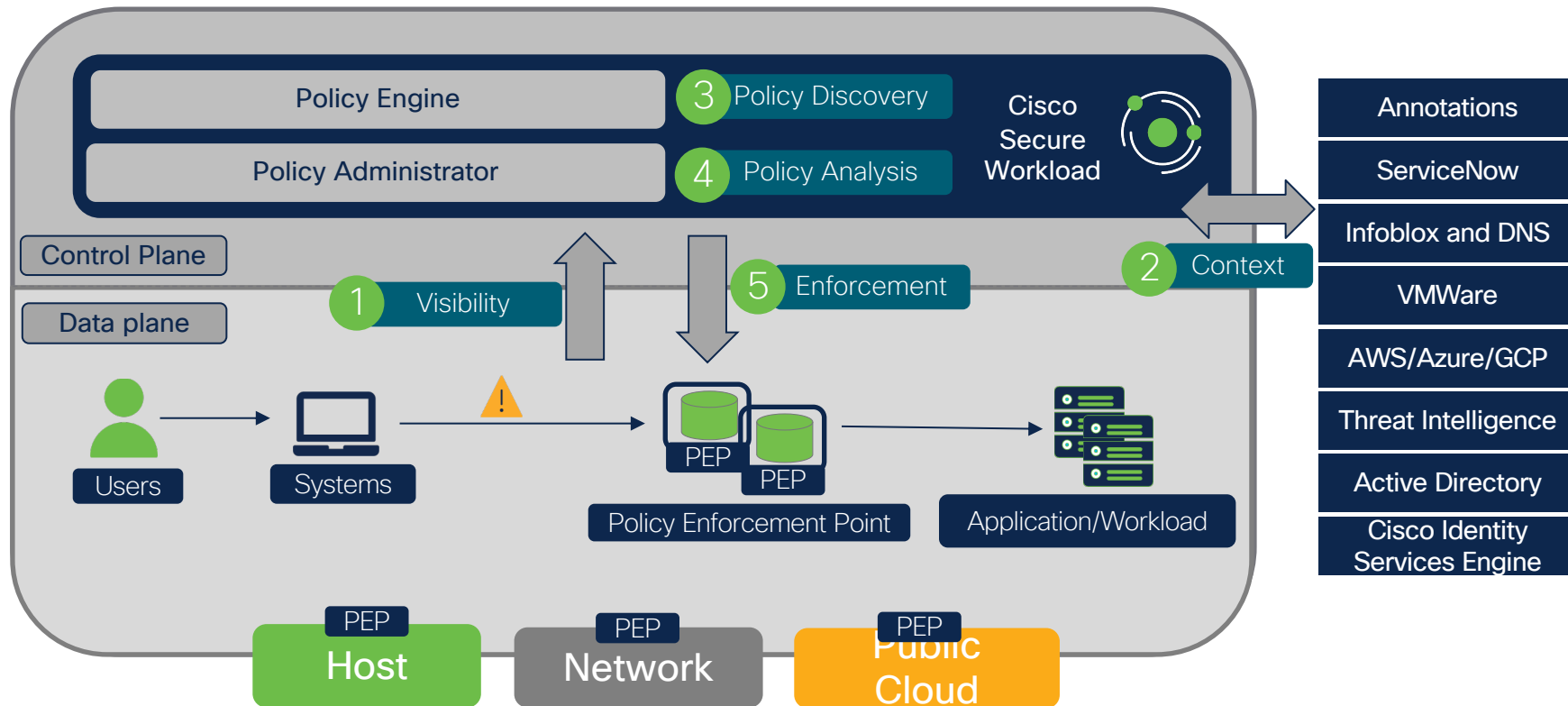
View All Tunnels

No Active Jobs

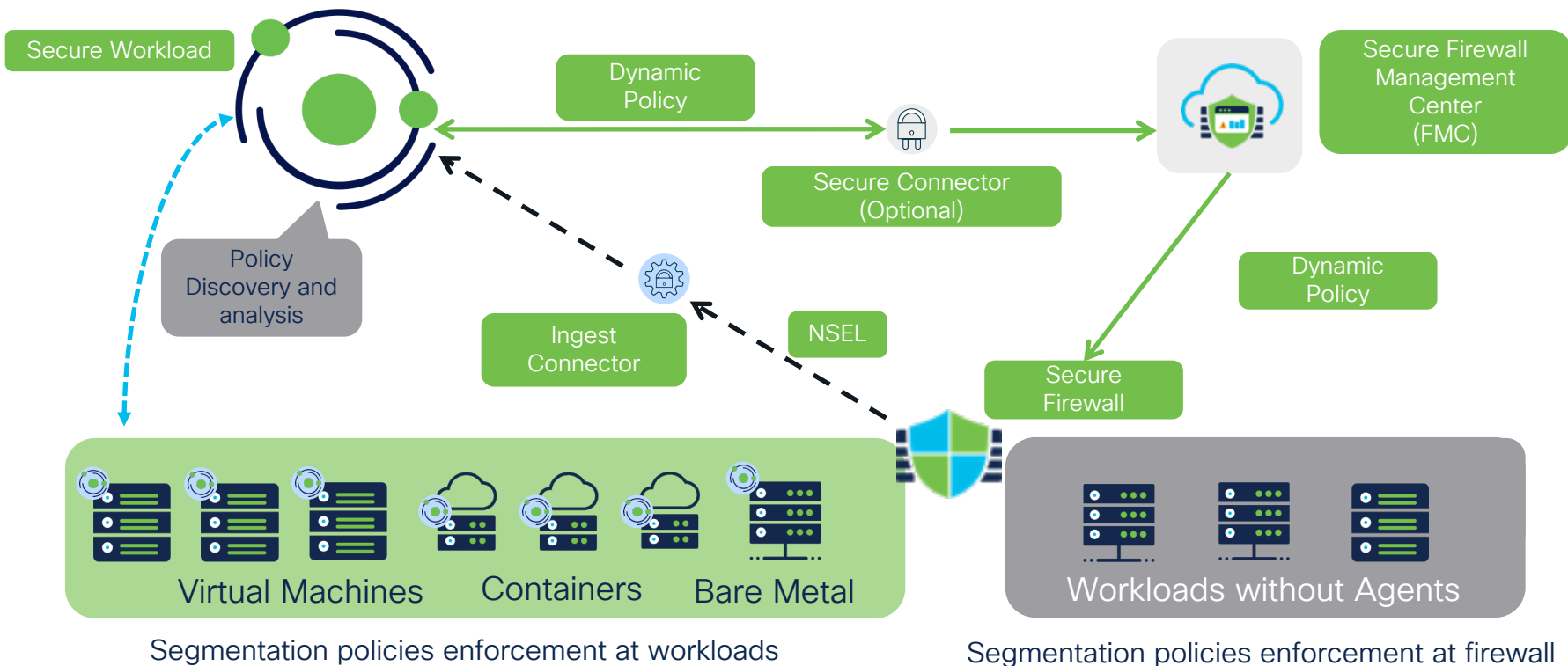
26 Background Tasks

Cisco Secure Workload

Cisco Secure Workload Architecture

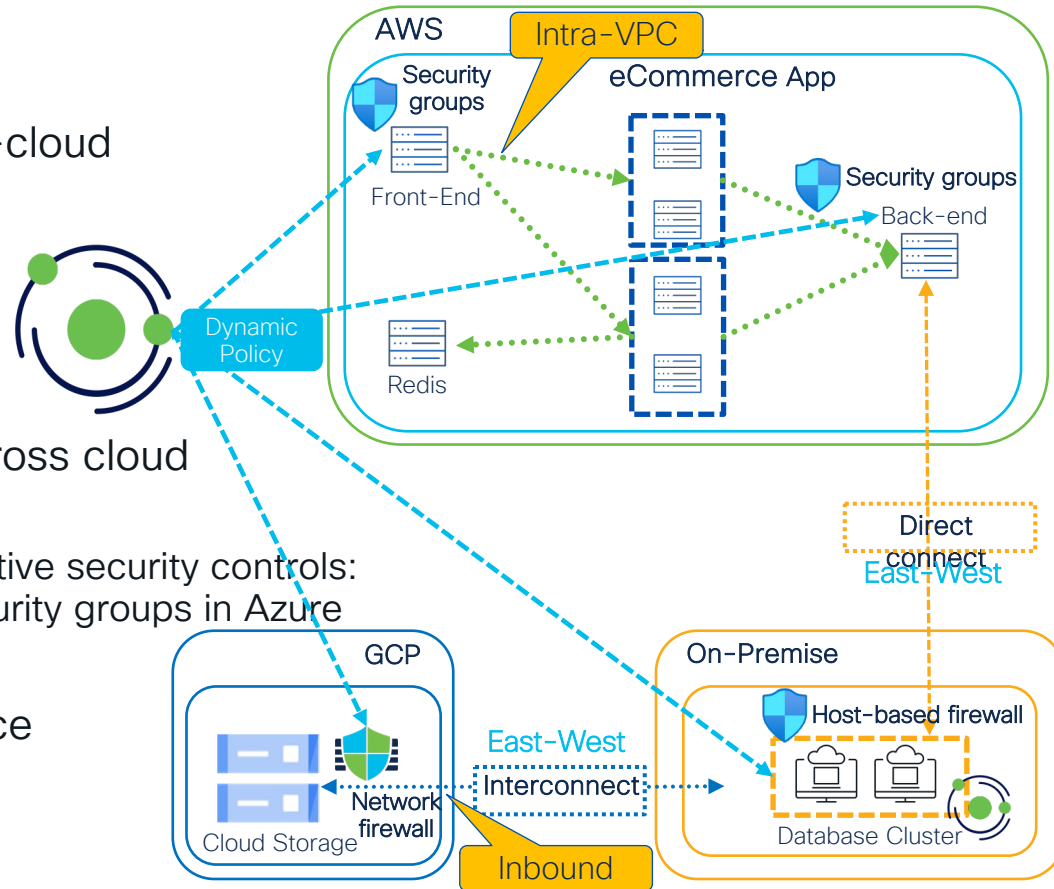


Agentless Segmentation Using Secure Firewall

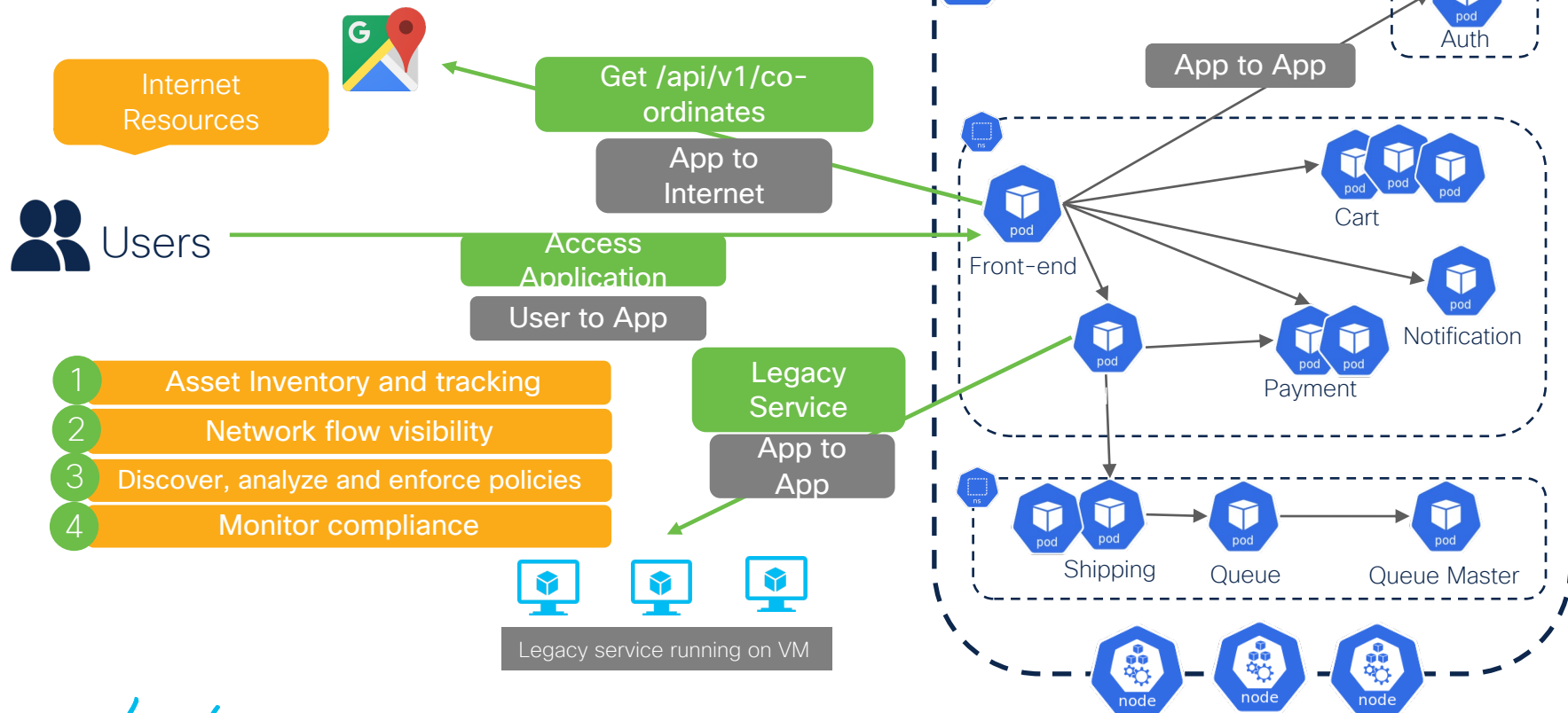


Multicloud Visibility and Segmentation

- Agentless flow visibility from multi-cloud environment
 - Support for AWS, Azure and GCP
- Contextual awareness using cloud tags and metadata
- Consistent policy orchestration across cloud providers
 - Agentless enforcement with cloud native security controls: security groups in AWS, network security groups in Azure and GCP native network firewall.
- Real-time update of policies at pace with cloud infrastructure updates.



Container Security



Conclusion



Did you know?

You can have a
one-on-one session with
a technical expert!

Visit Meet the Expert in The HUB
to meet, greet, whiteboard & gain
insights about your unique questions
with the best of the best.



Meet the Expert Opening Hours:

Tuesday	3:00pm – 7:00pm
Wednesday	11:15am – 7:00pm
Thursday	9:30am – 4:00pm
Friday	10:30am – 1:30pm

Session Surveys

We would love to know your feedback on this session!

- Complete a minimum of four session surveys and the overall event surveys to claim a Cisco Live T-Shirt



Continue your education



- Visit the Cisco Showcase 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

Participating in user research gives you a place to share your thoughts and experiences to influence the future of Cisco Secure products.

You'll hear from us at most once every 90 days.

You'll hear from us at most once every 90 days.

Participate is completely optional. You can opt out at any time.





The bridge to possible

Thank you

CISCO *Live!*

#CiscoLiveAPJC

The background is a vibrant, abstract graphic featuring a rainbow color palette. On the left, there are horizontal, wavy bands of color transitioning from red at the top to blue at the bottom. On the right, a bright white light source emits a series of sharp, triangular rays of color in a radial pattern, also using the rainbow spectrum. The overall effect is energetic and celebratory.

cisco *Live!*

Let's go

#CiscoLiveAPJC