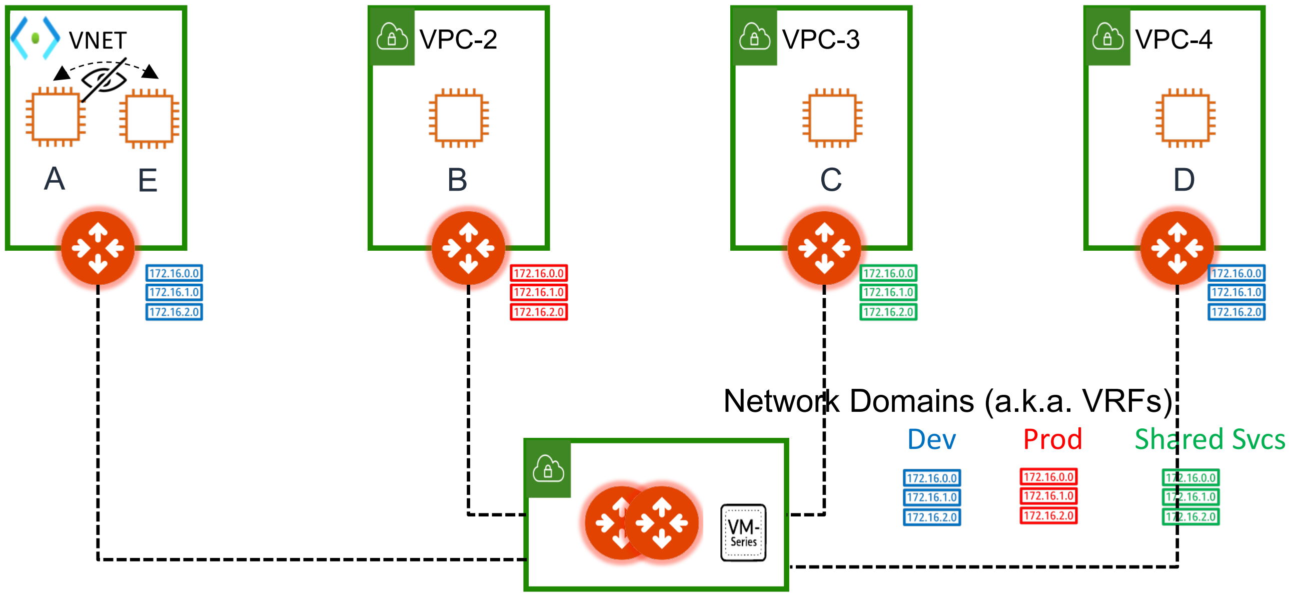
aviatrix

Distributed Firewalling

Distributed **Firewall** Problem Statement



Network Domain	Connects To
Prod	Shared Svcs
Dev	Shared Svcs
Shared Svcs	Prod, Dev

NGFWs can be a solution however an overkill for L3/L4 Rules & traffic has to be carried a long way to be inspected by the FW.

Firewall also doesn't have visibility and thus can't affect intra-VPC traffic

Distributed Firewall works by:

1.Leveraging the Aviatrix Spoke Gateways as Enforcement points.

2. Orchestate the provisioning of Azure NSGs, for Intra-VPC segmentation

The granularity of a Connection policy is at the Network Domain level thus it is not possible to:

1. Limit communication within the Network Domain (A to D)
2. Limit communication within the scope of the connection policy (A to C)

Diagram shows a single instance however in reality many instances will exist within each VPC.

Distributed Firewalling Basics

Distributed Firewalling* enforces policy exactly where needed across the entire network

Characteristics:

- Two components: Smart Groups & Rules
- Leveraging the Aviatrix Spoke Gateways as Enforcement points.
- Orchestating the provisioning of Azure NSGs, for Intra-VPC SmartGroup separation

* As of v3.2 of the CoPilot, Micro-Segmentation has been renamed to **Distributed Firewalling**

Smart Group

- **What is a Smart Group?**

A Smart Group identifies a group of resources that have similar policy requirements, that are confined in the same logical container.

- The members of a Smart Group can be classified using *three* methods:

  - CSP Tags
  - Resource Attributes
  - CIDR

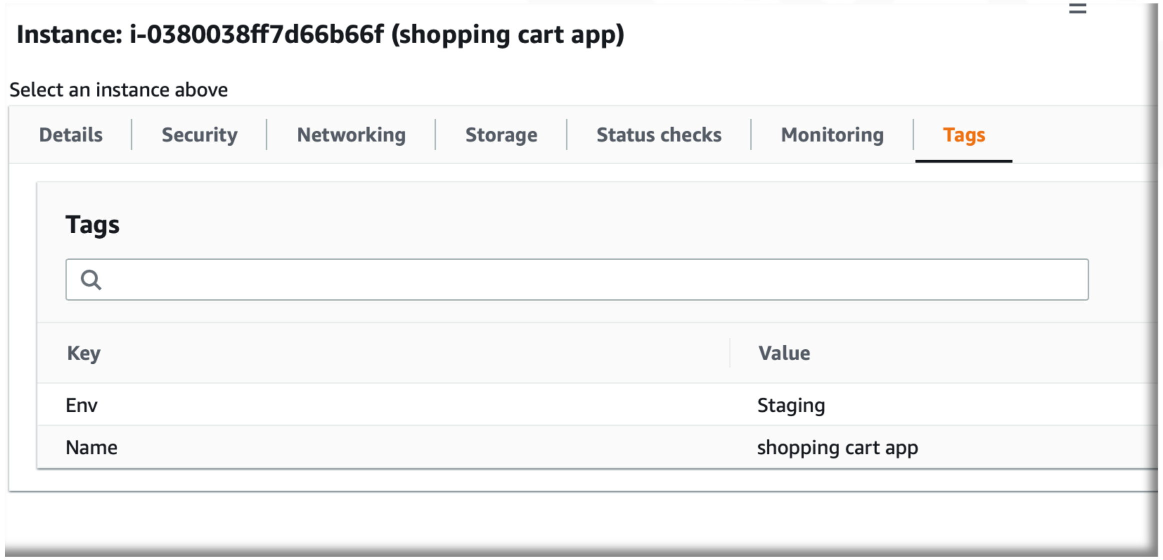
Classification Methods

CSP Tags (recommended)

- Tags are assigned to:
  - Instance
  - VPC/VNET
  - Subnet
- Tags are {Key, Value} pairs
- Eg: A VM hosting shopping cart application can be tagged with:

  {Key: Type, Value: Shopping cart app}

  {Key: Env, Value: Staging}



Instance: i-0380038ff7d66b66f (shopping cart app)

Select an instance above

Tags

Key	Value
Env	Staging
Name	shopping cart app

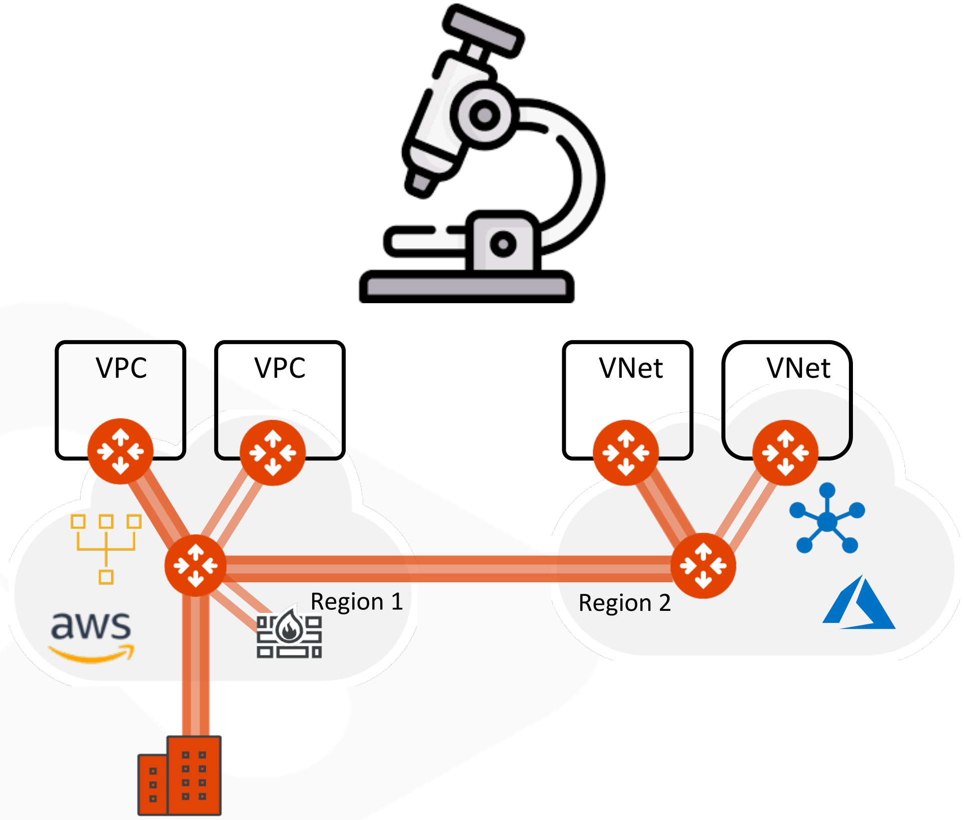
Resource attribute

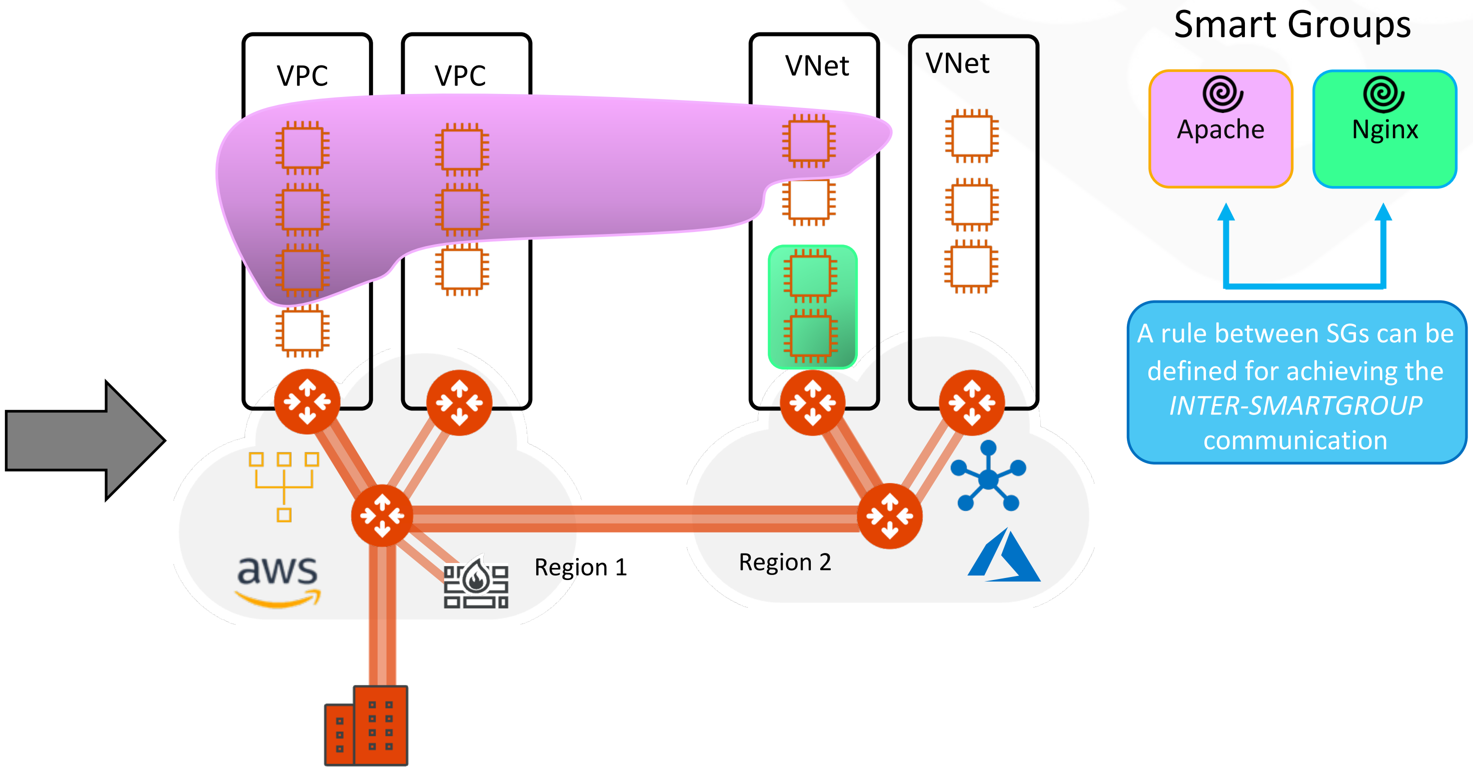
- Region Name, Account Name

IP Prefixes

- CIDR

Distributed Firewalling: Intra-rule vs. Inter-rule



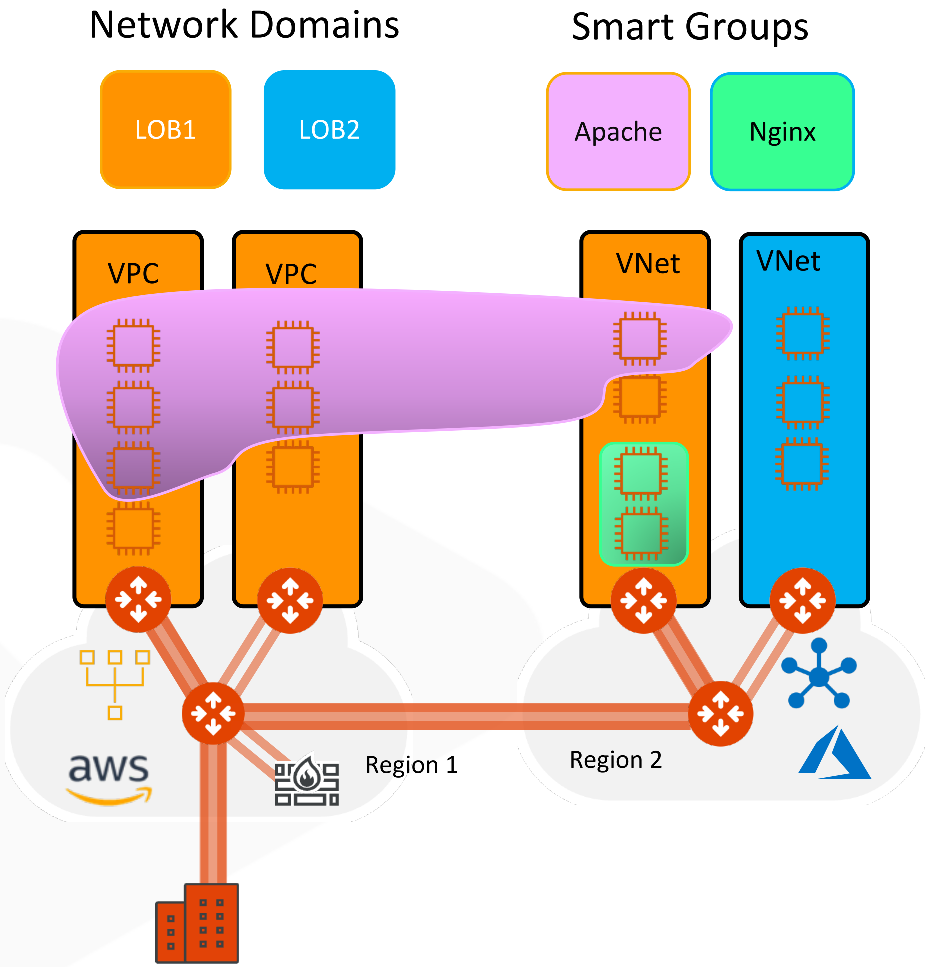


A rule between SGs can be defined for achieving the *INTER-SMARTGROUP* communication

- **INTRA-RULE**: is defined within a Smart Group, for dictating what kind of traffic is allowed/prohibited among all the instances that belong to that Smart Group

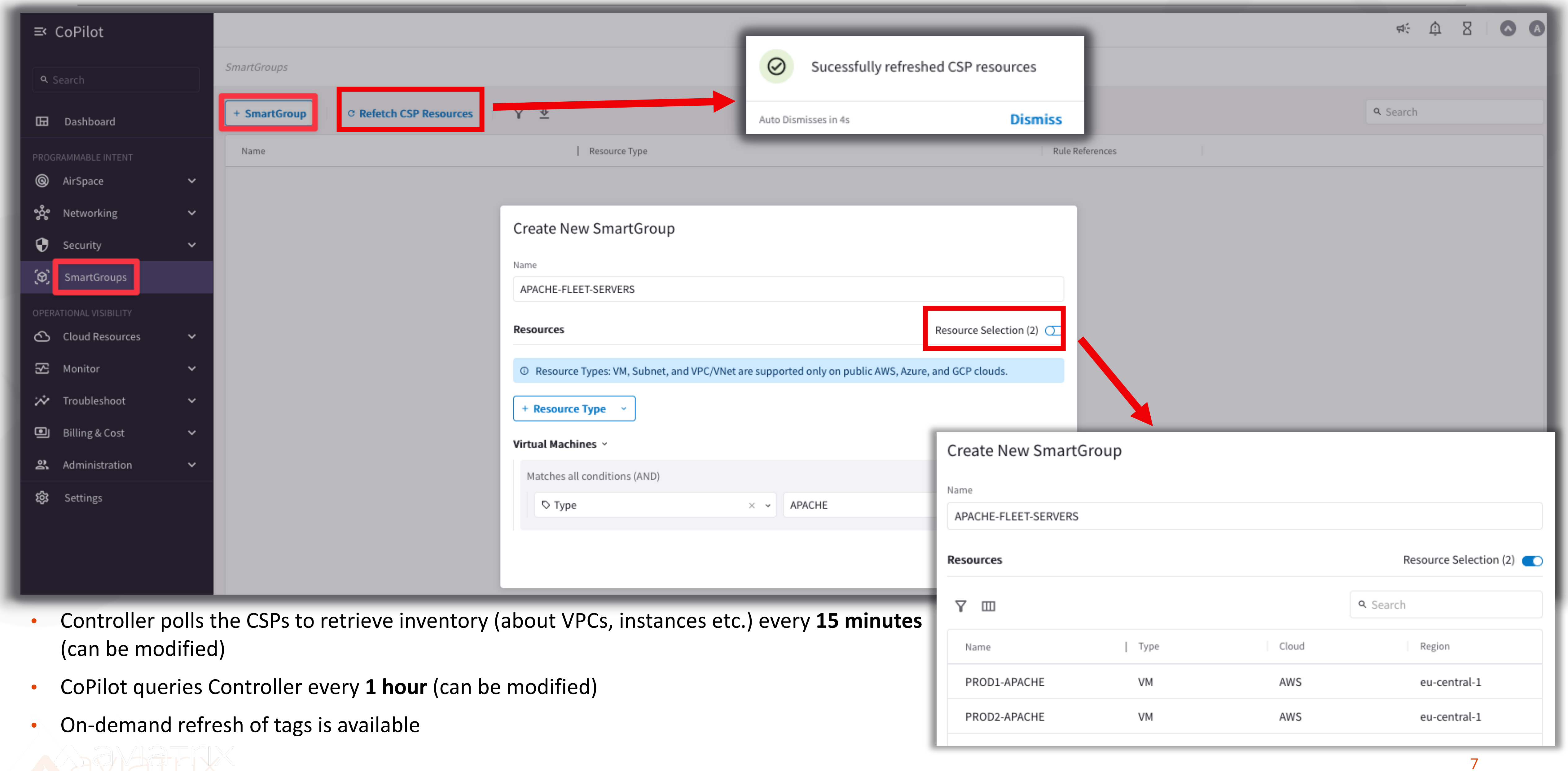
- **INTER-RULE:** is defined among Smart Groups, for dictating what kind of traffic is allowed/prohibited among two or more Smart Groups.

Network Segmentation & Distributed Firewalling Together



- **Scenario #1**: Smart Group defined within a Network Segment
- Network Segmentation and Distributed Firewalling are NOT mutually exclusive

Smart Groups Creation



Sucessfully refreshed CSP resources

Auto Dismisses in 4s
Dismiss

Create New SmartGroup

Name: APACHE-FLEET-SERVERS

Resources — Resource Selection (2)

ⓘ Resource Types: VM, Subnet, and VPC/VNet are supported only on public AWS, Azure, and GCP clouds.

+ Resource Type

Virtual Machines

Matches all conditions (AND)

Type: APACHE

Create New SmartGroup

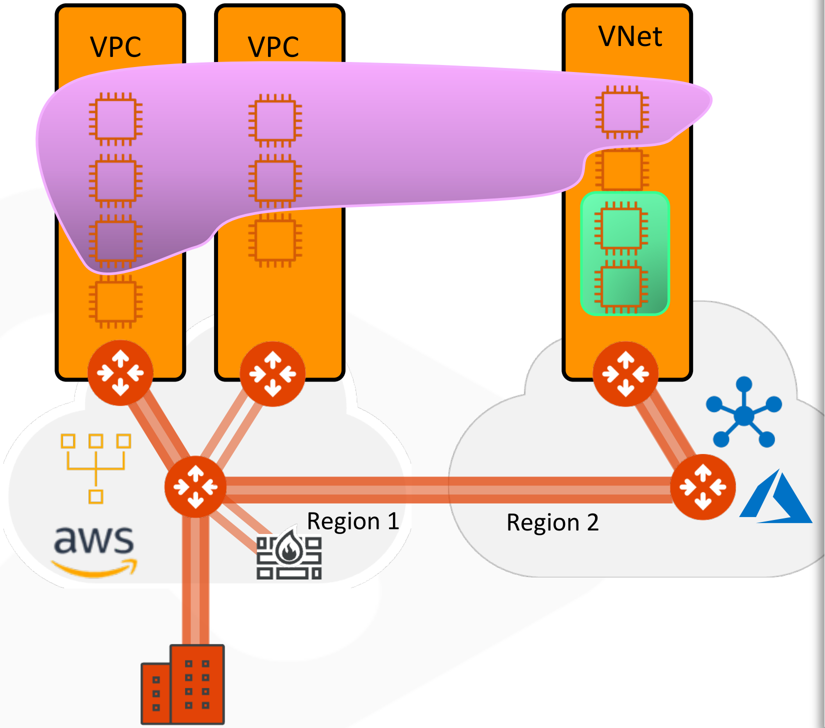
Name: APACHE-FLEET-SERVERS

Resources — Resource Selection (2)

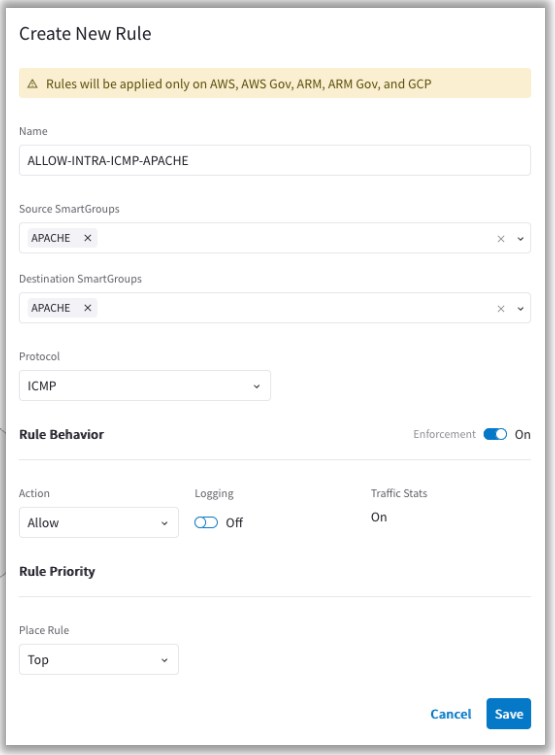
Name	Type	Cloud	Region
PROD1-APACHE	VM	AWS	eu-central-1
PROD2-APACHE	VM	AWS	eu-central-1

- Controller polls the CSPs to retrieve inventory (about VPCs, instances etc.) every **15 minutes** (can be modified)
- CoPilot queries Controller every **1 hour** (can be modified)
- On-demand refresh of tags is available

Distributed Firewalling Rules on Smart Groups



SmartGroup Apache



Create New Rule

⚠ Rules will be applied only on AWS, AWS Gov, ARM, ARM Gov, and GCP

Name: ALLOW-INTRA-ICMP-APACHE

Source SmartGroups: APACHE ×

Destination SmartGroups: APACHE ×

Protocol: ICMP

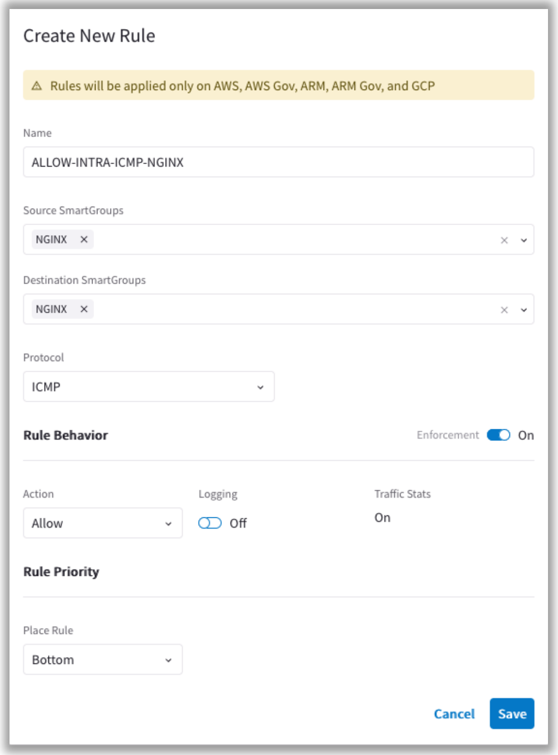
Rule Behavior — Enforcement: On

Action: Allow
Logging: Off
Traffic Stats: On

Rule Priority

Place Rule: Top

SmartGroup Nginx



Create New Rule

⚠ Rules will be applied only on AWS, AWS Gov, ARM, ARM Gov, and GCP

Name: ALLOW-INTRA-ICMP-NGINX

Source SmartGroups: NGINX ×

Destination SmartGroups: NGINX ×

Protocol: ICMP

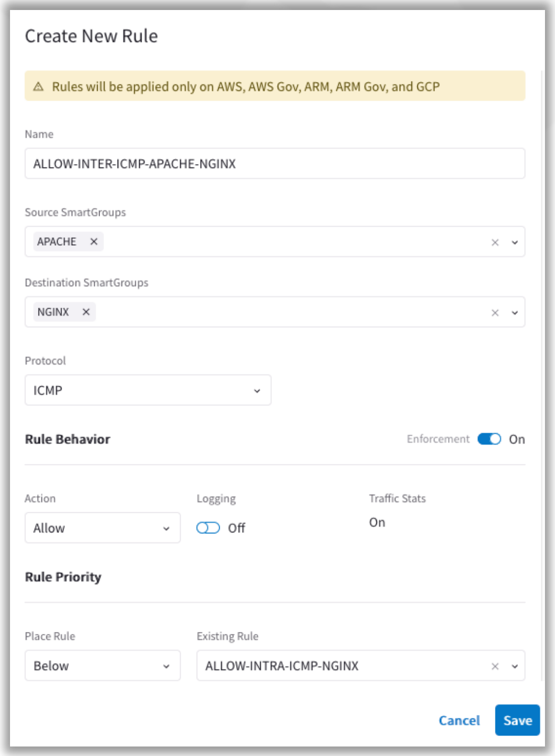
Rule Behavior — Enforcement: On

Action: Allow
Logging: Off
Traffic Stats: On

Rule Priority

Place Rule: Bottom

SmartGroup Apache ↔ SmartGroup Nginx



Create New Rule

⚠ Rules will be applied only on AWS, AWS Gov, ARM, ARM Gov, and GCP

Name: ALLOW-INTER-ICMP-APACHE-NGINX

Source SmartGroups: APACHE ×

Destination SmartGroups: NGINX ×

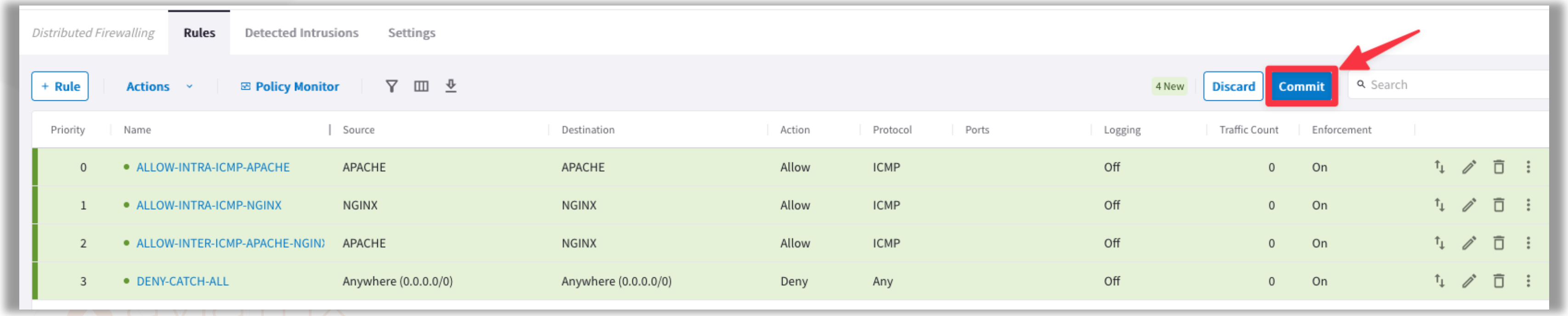
Protocol: ICMP

Rule Behavior — Enforcement: On

Action: Allow
Logging: Off
Traffic Stats: On

Rule Priority

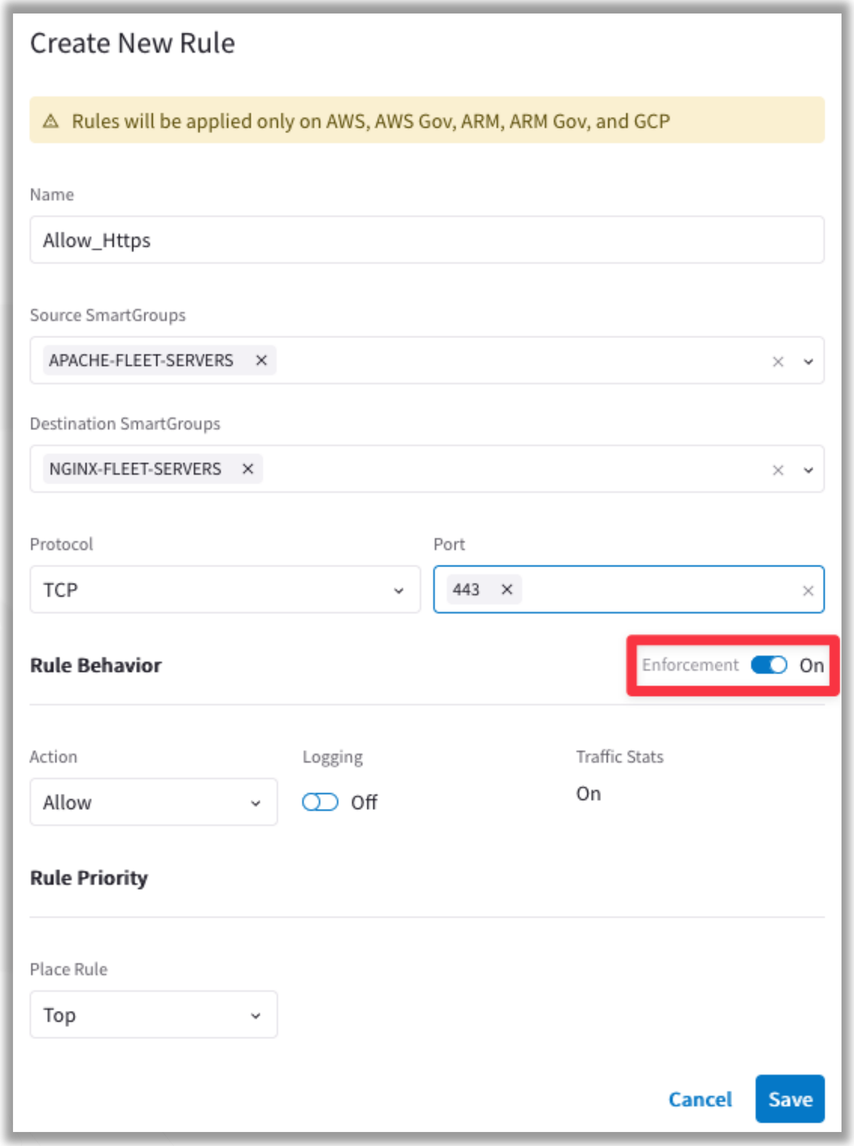
Place Rule: Below
Existing Rule: ALLOW-INTRA-ICMP-NGINX



Priority	Name	Source	Destination	Action	Protocol	Ports	Logging	Traffic Count	Enforcement
0	ALLOW-INTRA-ICMP-APACHE	APACHE	APACHE	Allow	ICMP		Off	0	On
1	ALLOW-INTRA-ICMP-NGINX	NGINX	NGINX	Allow	ICMP		Off	0	On
2	ALLOW-INTER-ICMP-APACHE-NGIN)	APACHE	NGINX	Allow	ICMP		Off	0	On
3	DENY-CATCH-ALL	Anywhere (0.0.0.0/0)	Anywhere (0.0.0.0/0)	Deny	Any		Off	0	On

- Rule changes are saved in **Draft** state.
- When you apply a rule to a SmartGroup, please keep in mind that there is an **Invisible Hidden Deny** at the very bottom.
- To save the changes click on “**Commit**”
- **Discard** will trash the changes
- Rule is **stateful**, this means that the return traffic is allowed automatically

Rule Enforcement



Create New Rule

⚠ Rules will be applied only on AWS, AWS Gov, ARM, ARM Gov, and GCP

Name: Allow_Https

Source SmartGroups: APACHE-FLEET-SERVERS

Destination SmartGroups: NGINX-FLEET-SERVERS

Protocol: TCP

Port: 443

Rule Behavior

Enforcement: On

Action: Allow

Logging: Off

Traffic Stats: On

Rule Priority

Place Rule: Top

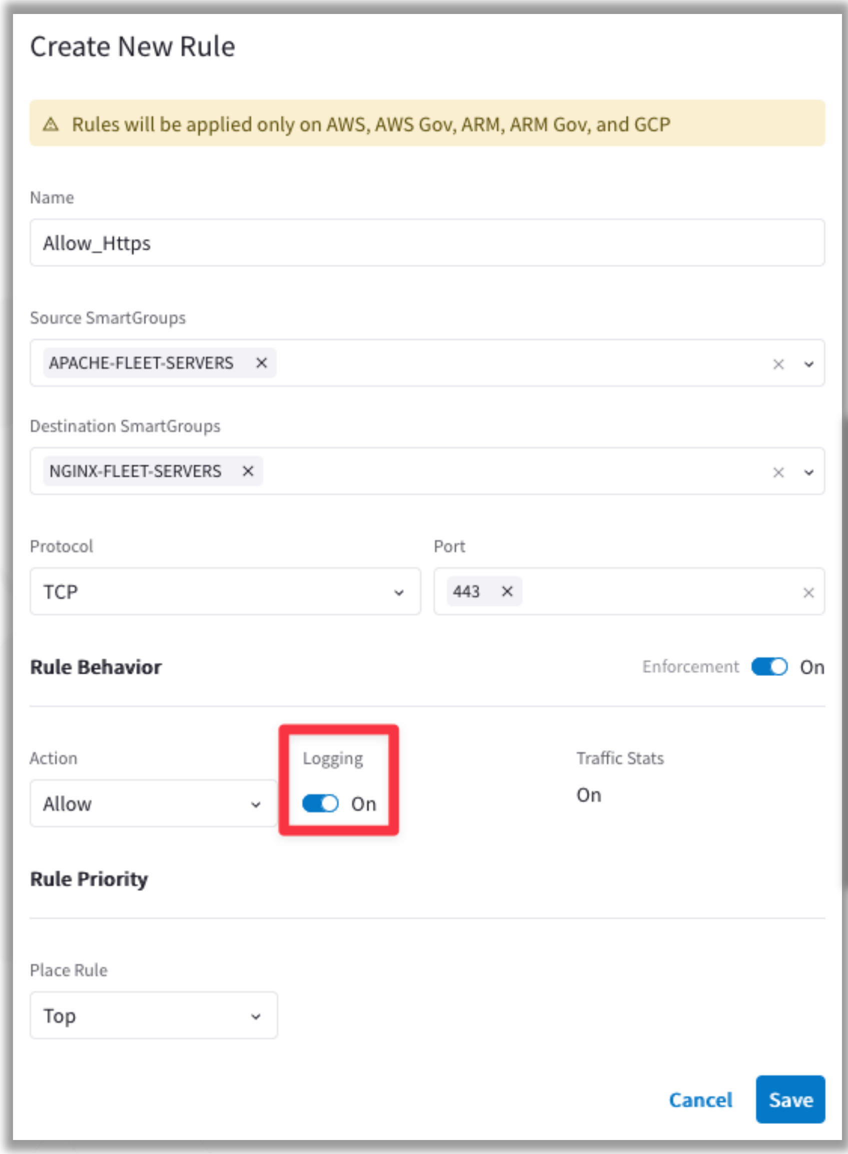
□ Enforcement ON

- Policy is enforced in the Data Plane

□ Enforcement OFF

- Policy is NOT enforced in the Data Plane
- The option provides a *Watch/Test* mode
- Common use case is with deny rule
- Watch what traffic hits the deny rule before enforcing the rule in the Data Plane.

Rule Logging



Create New Rule

⚠ Rules will be applied only on AWS, AWS Gov, ARM, ARM Gov, and GCP

Name: Allow_Https

Source SmartGroups: APACHE-FLEET-SERVERS ✕

Destination SmartGroups: NGINX-FLEET-SERVERS ✕

Protocol: TCP

Port: 443 ✕

Rule Behavior

Enforcement: On

Action: Allow

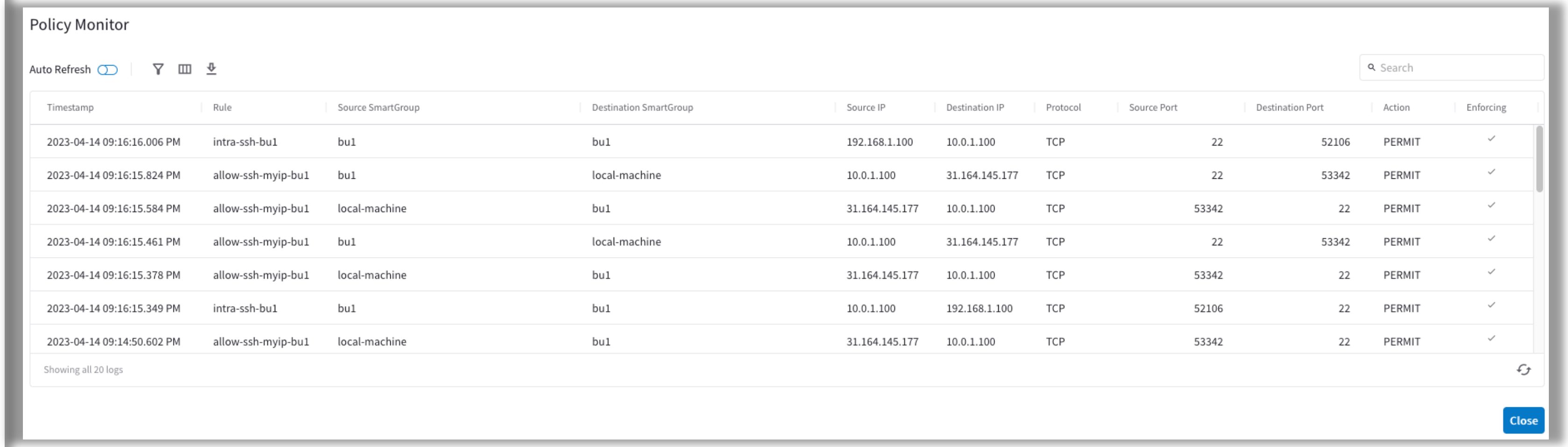
Logging: On

Traffic Stats: On

Rule Priority

Place Rule: Top

- ❑ **Logging can be turned ON/OFF per rule**
- ❑ **Configure Syslog to view the logs**



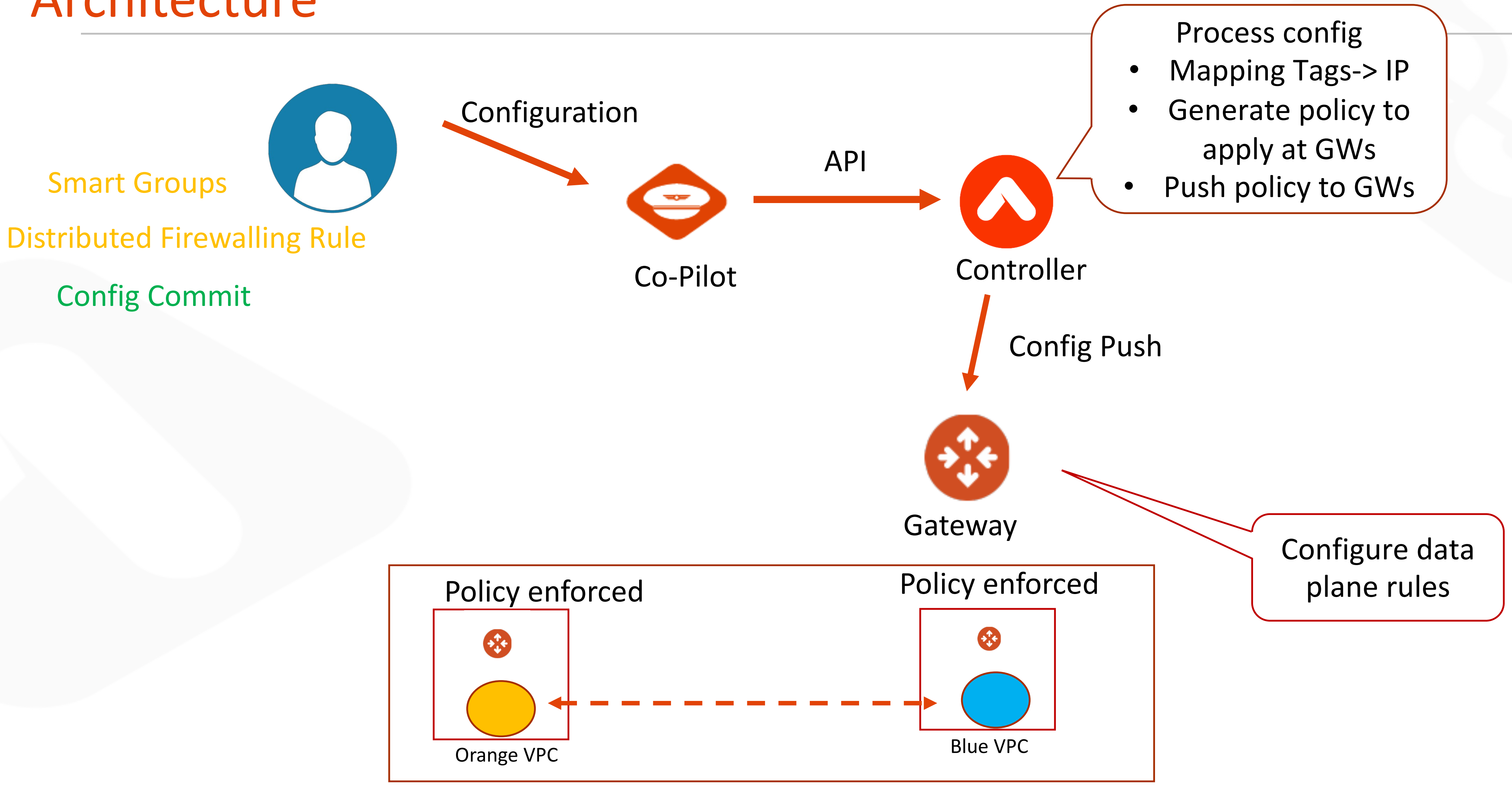
Policy Monitor

Auto Refresh

Timestamp	Rule	Source SmartGroup	Destination SmartGroup	Source IP	Destination IP	Protocol	Source Port	Destination Port	Action	Enforcing
2023-04-14 09:16:16.006 PM	intra-ssh-bu1	bu1	bu1	192.168.1.100	10.0.1.100	TCP	22	52106	PERMIT	✓
2023-04-14 09:16:15.824 PM	allow-ssh-myip-bu1	bu1	local-machine	10.0.1.100	31.164.145.177	TCP	22	53342	PERMIT	✓
2023-04-14 09:16:15.584 PM	allow-ssh-myip-bu1	local-machine	bu1	31.164.145.177	10.0.1.100	TCP	53342	22	PERMIT	✓
2023-04-14 09:16:15.461 PM	allow-ssh-myip-bu1	bu1	local-machine	10.0.1.100	31.164.145.177	TCP	22	53342	PERMIT	✓
2023-04-14 09:16:15.378 PM	allow-ssh-myip-bu1	local-machine	bu1	31.164.145.177	10.0.1.100	TCP	53342	22	PERMIT	✓
2023-04-14 09:16:15.349 PM	intra-ssh-bu1	bu1	bu1	10.0.1.100	192.168.1.100	TCP	52106	22	PERMIT	✓
2023-04-14 09:14:50.602 PM	allow-ssh-myip-bu1	local-machine	bu1	31.164.145.177	10.0.1.100	TCP	53342	22	PERMIT	✓

Showing all 20 logs

Architecture



Smart Groups

Distributed Firewalling Rule

Config Commit

Process config
- Mapping Tags-> IP
- Generate policy to apply at GWs
- Push policy to GWs

Configure data plane rules

aviatrix

Next: Security