

The Cisco Live! logo, featuring the word "CISCO" in a dark blue, sans-serif font, followed by "Live!" in a dark blue, script font.

CISCO *Live!*

The text "Let's go" in a large, dark blue, sans-serif font, positioned to the left of a bright white sunburst graphic that radiates across the right side of the image.

Let's go

#CiscoLiveAPJC



The bridge to possible

Become a Security NetOps Ninja

- Automating your security deployments and operations

Ling Yang - Customer Delivery Security Architect

Hari Haran S M - Customer Delivery Engineering Technical Leader

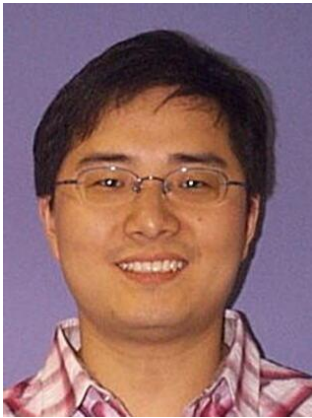
BRKSEC-2707

CISCO *Live!*

#CiscoLiveAPJC

About Your Speaker

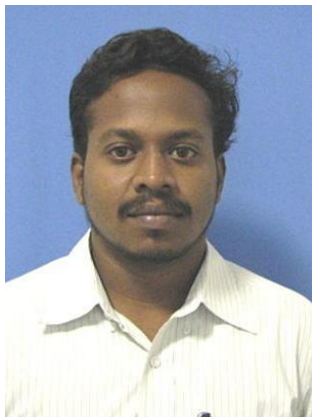
Ling Yang lingya@cisco.com



- Joined Cisco in 2005
- Delivery Security Architect in Customer Success (CX) Australia
- CIO-100 Honoree for innovation in IT in 2014
- M.S.C.S. USYD (Master of Science in Computer Science, University of Sydney)
- 4 x CCIE in Enterprise Infrastructure, Security, Service Provider and Storage Networking
- 18 years experience in designing, building and testing network solutions for regional and global customers covering Cisco security products, virtualization, service provider and core routing technologies



About Your Speaker



Hariharan S M

hasm@cisco.com

Hariharan is a Technical Leader in Security TAC CX. He has multi domain expertise including Switching, on-prem conferencing, voice gateways, VPN and AAA. He has 16 years of industry experience and been with Cisco over 12 years. He is customer obsessed and spent most of his career as TAC Engineer.

He is passionate about automation with a moto “Make things Simple”. He has experience in creating full stack application and some devops.

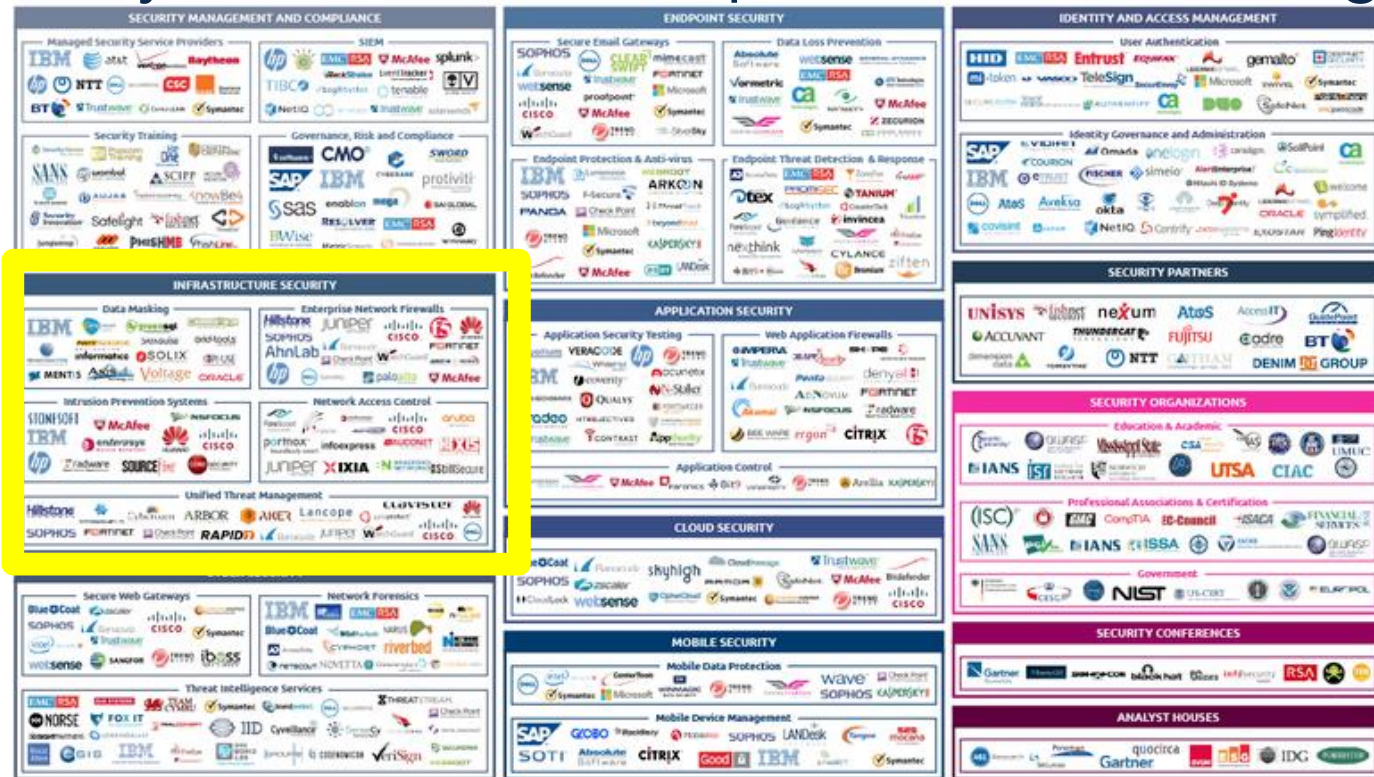
Agenda

CISCO *Live!*

- Security Infrastructure Operational Challenges
- Automating Security Deployments and Operations
 - Selecting “Best” Tools
 - Infrastructure Provisioning
 - Policy Management
 - Use Cases and Demo
- One More Thing
 - Use Cases and Demo
- Wrap-Up
- Q&A

Security Infrastructure Operational Challenges

Security Infrastructure Operational Challenges



Source: <https://www.digitalguardian.com/blog/information-security-industryscape>

Security Infrastructure Operational Challenges

Gartner Prediction No. 1: By 2025, **70%** of organizations will implement structured infrastructure automation to deliver flexibility and efficiency, up from 20% in 2021.

- Infrastructure automation should be seen as an ongoing discipline, as it requires proper planning and coordination to drive productivity.
- When beginning an automation journey, focus on productivity and efficiency as initial drivers, enabling teams to offload routine work to automation and improve the volume of work that can be executed.

Gartner Prediction No. 3: By 2027, **35%** of application workloads will not be optimal or ready for cloud delivery, down from 55% in 2022

- Infrastructure will soon reach a state where approximately one-third of workloads will run in the cloud
- one-third can migrate or move to the cloud with increased complexity
- one-third will not run in the cloud because of complexity, performance, economics or regulatory constraints

Source: <https://www.gartner.com/en/articles/4-predictions-for-i-o-leaders-on-the-path-to-digital-infrastructure>

Security Infrastructure Operational Challenges



Automated Provisioning



Version Control



Configuration
Management



Continuous Integration
- Delivery



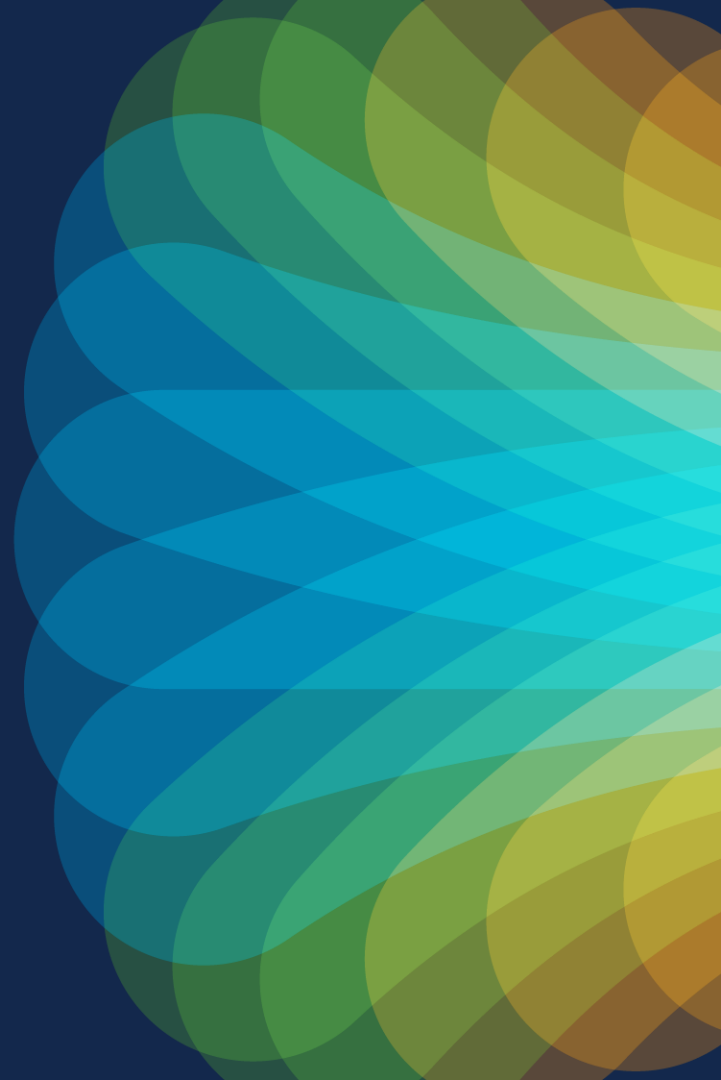
Scalability



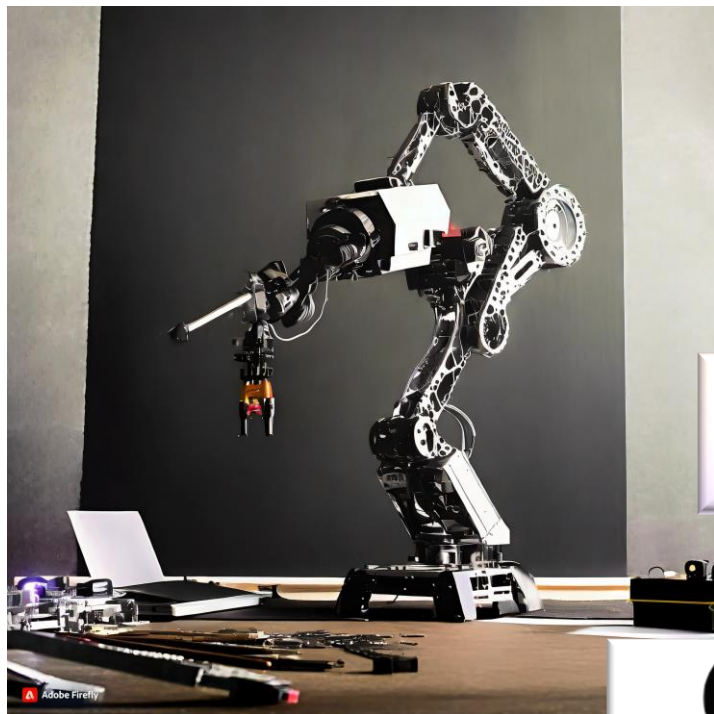
Infrastructure and right policy



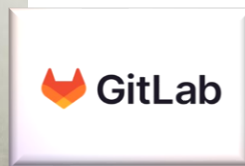
Automating Security Deployment and Operations



Selecting the “Best” Tools (Plural “tool-s”)



An AI Generated
Image





The Better Question... *“Which tools are better together for What you need?”*

Often said by those who have implemented large scale network automation

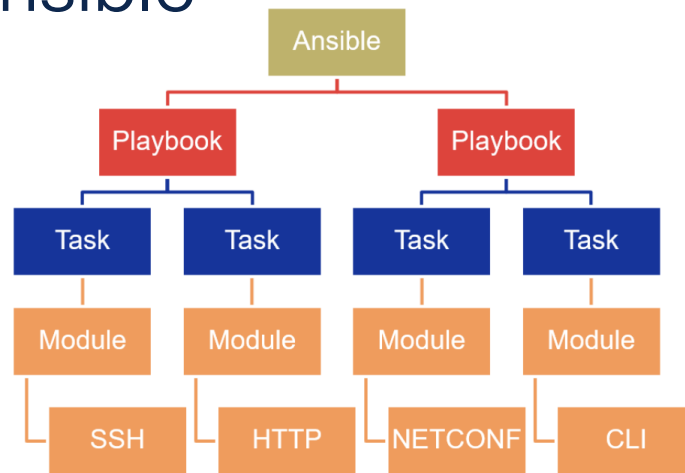
Selecting the “Best” tools – Ansible

Ansible is an [open-source](#) software [provisioning](#), configuration management, and application-deployment tool enabling [infrastructure as code](#).

It runs on many Unix- like systems, and can configure both Unix- like systems as well as Microsoft Windows.

It includes its own [declarative language](#) to describe system configuration.

Ansible is agentless, temporarily connecting remotely via [SSH](#) or [API](#) to do its tasks.



Ansible was designed to automate compute, storage and other non-networking infrastructure, especially in the cloud.

The most tedious and error-prone parts of network automation are updating and deleting configuration. Developing with Ansible involves troubleshooting whitespaces and configuration dependencies

Selecting the “Best” tools – Python



Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or **glue language** to connect existing components together.

Selecting the “Best” Tools – Battle Suit

Orchestration

Python

Configuration
Management

Python

Python

Python

Python

Automated Provisioning

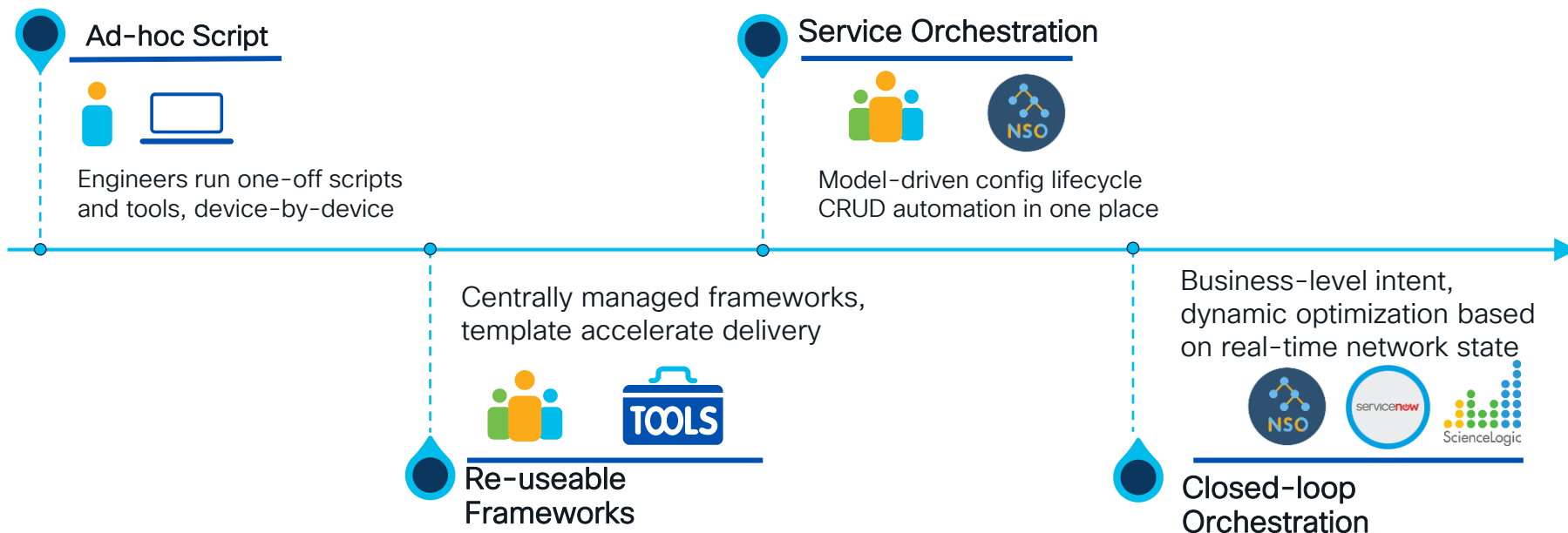


Version Control: GitLab, GitHub ...

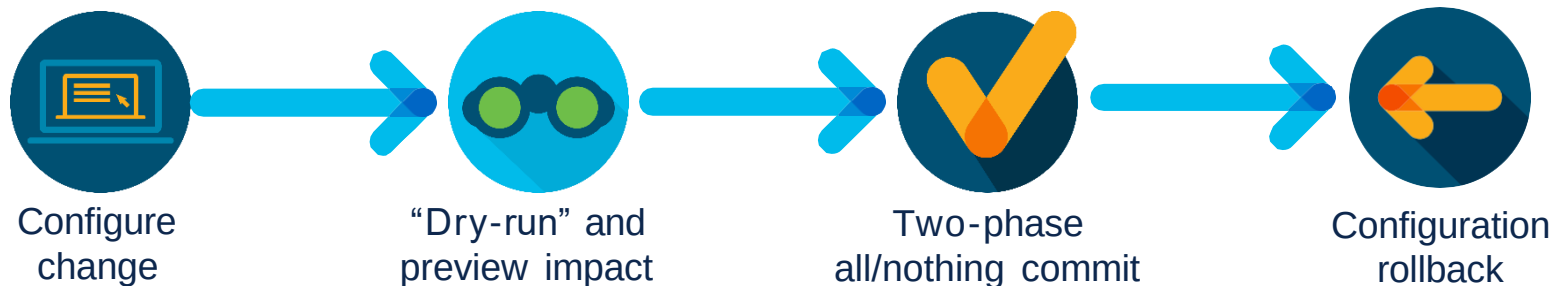
Continuous Integration –
Delivery: Jenkins, GitLab ...

An AI Generated
Image

Maturity Journey



Selecting the “Best” Tools – Transaction



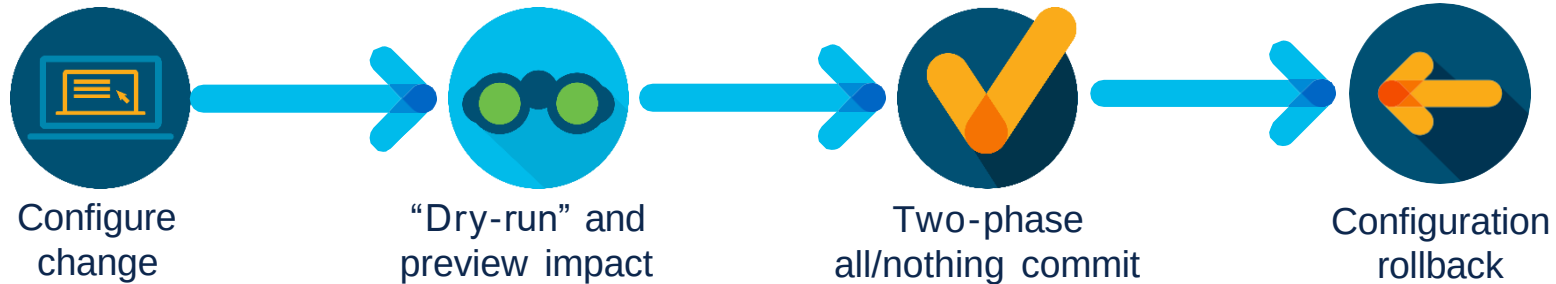
Transaction – independent unit of work (ACID)

- Atomicity – all operations treated as single unit
- Consistency – data is in consistent state
- Isolation – concurrent transactions are possible
- Durability – result is permanent

Why Transaction is Important

- Ability to perform more than one operation in an atomic way
- Perform completely or not at all
- Restoration of state upon failure (all or nothing)
- Simplifies support for rollback
- Automation and programmability is possible

Selecting the “Best” Tools – Transaction



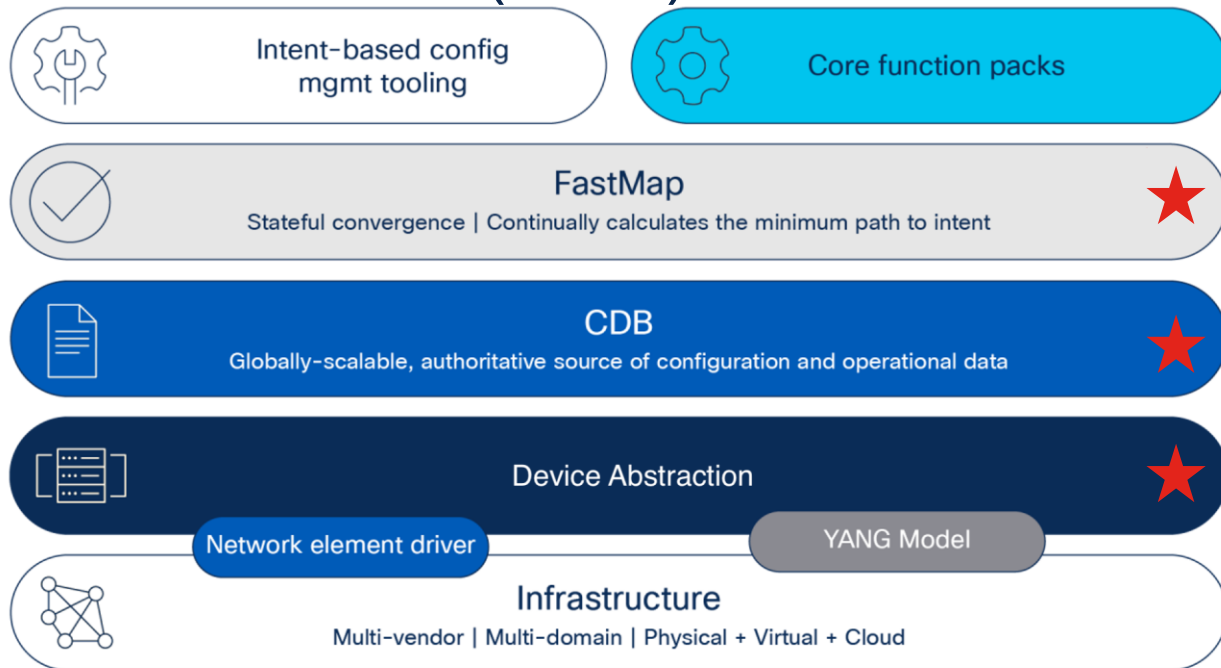
Transactional Devices

- Support for transactions (Netconf with commit and confirmed-commit capabilities)
- Integration into Network Orchestrator
- Programmability and automation
- Rollback is usually supported or simpler to perform

Non-Transactional Device

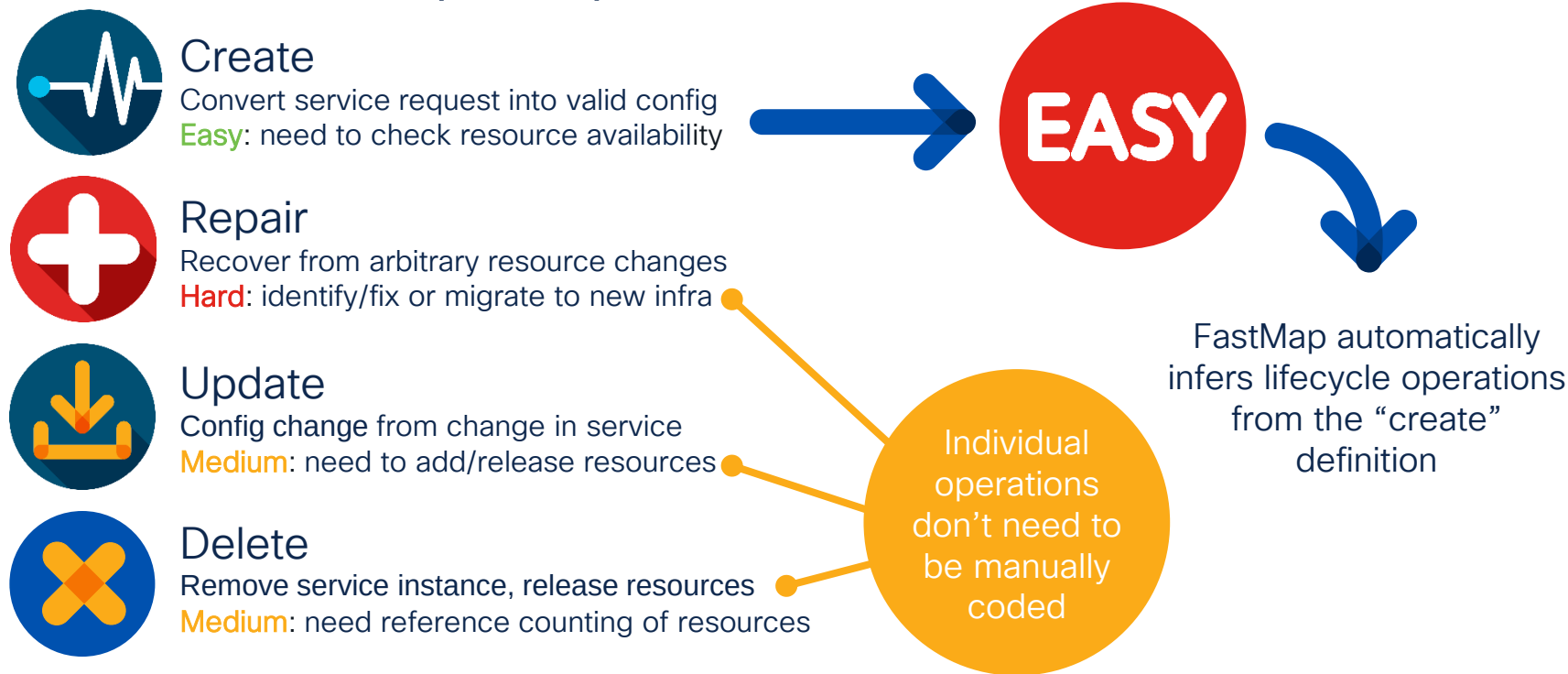
- SNMP device, CLI devices
- One operation can bring device into failed state
- Knowledge of command orders and rules is required
- Cannot be easily automated,
- cheap to start but more expensive to operate
- No simple rollback

Selecting the “Best” Tools – Network Services Orchestrator (NSO)

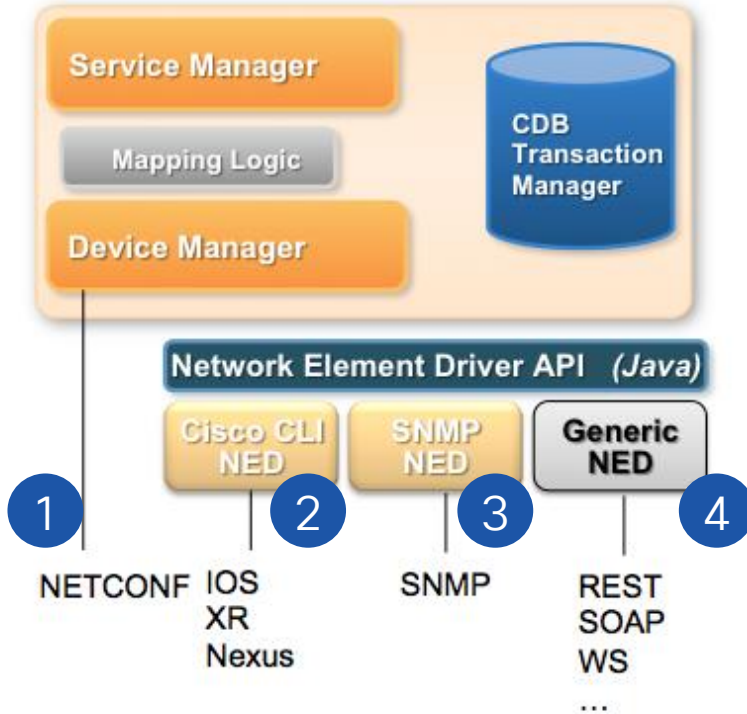


- Model-driven, end-to-end service lifecycle and customer experience focused
- Seamless integration with northbound tooling
- Loosely-coupled and modular architecture leveraging open APIs and standard protocols
- Orchestration across multi-domain and multi-layer for network-wide, centralized policy and services

Selecting the “Best” Tools – Network Services Orchestrator (NSO) – FASTMAP



Selecting the “Best” Tools – Network Services Orchestrator (NSO) – Network Element Driver (NED)



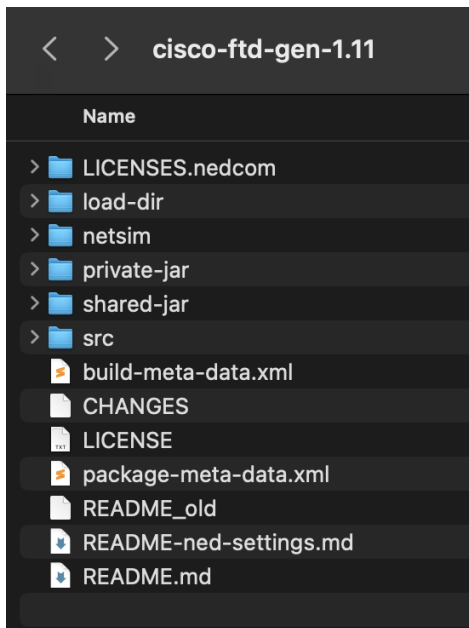
There are four categories of NEDs, depending on the device interface:

1. NETCONF NED: a device that supports NETCONF
2. CLI NED: any device with a CLI that resembles a Cisco CLI
3. SNMP NED: an SNMP device
4. Generic NED: proprietary protocols like REST, and third-party CLIs

Selecting the “Best” Tools – Network Services Orchestrator (NSO) – NED Multi-Vendor and Customization



Selecting the “Best” Tools – Network Services Orchestrator (NSO) – NED Security Product



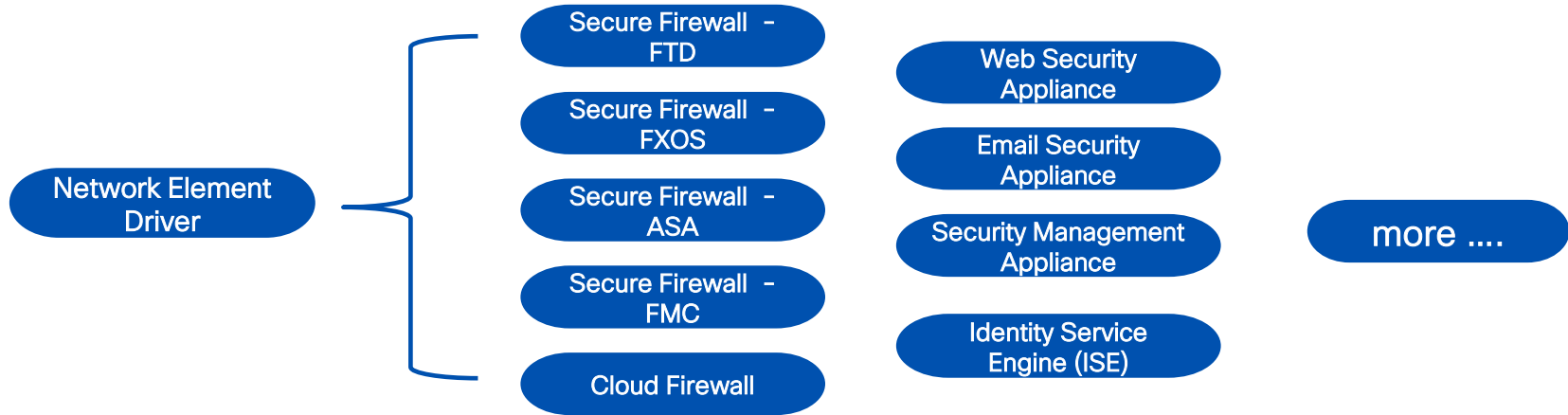
 [ncs-6.1-cisco-ftd-1.11.5.signed.bin](#) 2023-05-17 07:37 3.8M

 [ncs-6.1-cisco-ftd-1.11.6.signed.bin](#) 2023-05-22 16:02 3.7M

```
cli:ned-diff-move-after ">interface" {  
    cli:arguments "when-set";  
}  
cli:ned-diff-move-before ">>interface" {  
    cli:arguments "when-delete|prio=+1";  
}
```

```
leaf name {  
    //tailf:cli-diff-delete-after "/ftd/object/udpports/name";  
    tailf:cli-diff-dependency "/ftd/object/udpports[name=current()/../name]";  
    tailf:cli-diff-dependency "/ftd/object/tcpports[name=current()/../name]";  
    type string;  
}
```

Selecting the “Best” Tools – Network Services Orchestrator (NSO) – Cisco Security Product



- Cisco security products are not transactional devices
- Non-transactional devices can be made transactional via NED

Selecting the “Best” tools – Connecting dots



Selecting the “Best” tools – Connecting dots

Ansible is good at stitching together pieces into a workflow. NSO is good at configuring and deploying network services

Ansible needs NSO for:

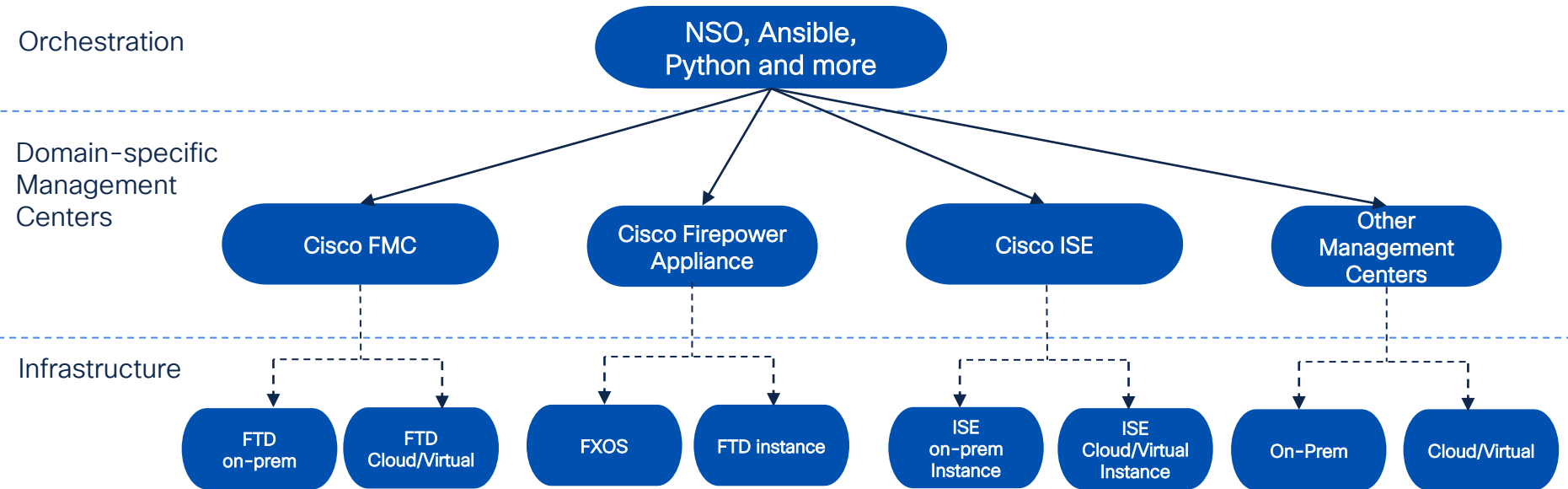
- Service life-cycle management
- Configuration transactions that are stateful
- Device management with NSO NEDs(Network Element Drivers)
- Configuration Sync-from / Sync-to between devices
- Device Configuration Auto / Partial rollback in case of failure
- Commit-queues for faster response times

NSO Modules for Ansible:

- nso_action - to run actions in NSO
- nso_config - to manage configuration through NSO
- nso_query - to query the NSO data store
- nso_show - to show subtree data from NSO data store
- nso_verify - to verify configuration and operational data in NSO

Source: <https://community.cisco.com/t5/nso-developer-hub-blogs/nso-and-ansible-together-complement-each-other-s-strengths/ba-p/4138534>

Connecting the Dots - Building Blocks



Selecting the “Best” Tools – Improved Battle Suit

Automated Provisioning: [NSO](#), [Python](#), [Ansible](#) ...

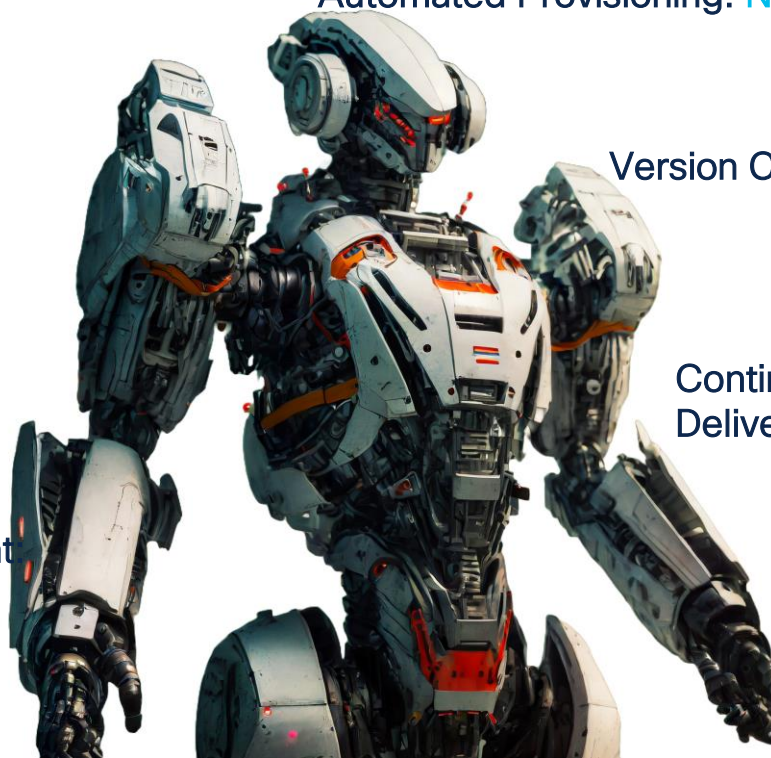
Orchestration: [NSO](#),
[Python](#) ...

Version Control: [GitLab](#), [GitHub](#) ...

Scalability and Ease of Use:
[NSO](#), [Python](#) ...

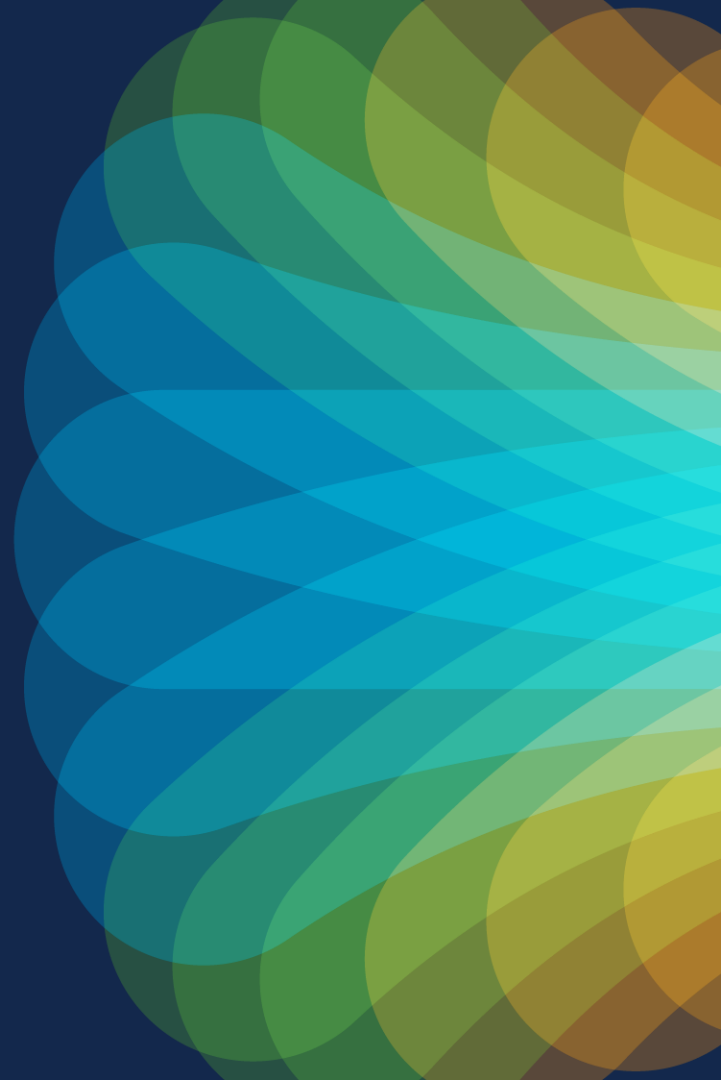
Continuous Integration –
Delivery: [Jenkins](#), [GitLab](#) ...

Configuration Management
[NSO](#), [Ansible](#)...



An AI Generated
Image

Automation Concepts in infrastructure Provisioning



Use Case and Demo – Test the Battle Suit

Build NSO Portal (customised portal)

- Infrastructure Provisioning
- Disaster Recovery



CISCO *Live!*

An AI Generated
Image

A screenshot of a login portal for CiscoLive 2023 Melbourne. The portal is white with a blue background. It features the Cisco logo at the top, followed by the event name "CiscoLive 2023 Melbourne" and the session ID "BRKSEC-2707". Below this, there are input fields for "Username" (containing "Admin") and "Password" (containing "*****"). A blue "Sign In" button is at the bottom.


CiscoLive 2023
Melbourne
BRKSEC-2707

Username

Password

[Sign In](#)

Firewall Vision: Network, Workload, and Cloud

Single security management plane to abstract end-to-end policy intent from enforcement point specific configuration.

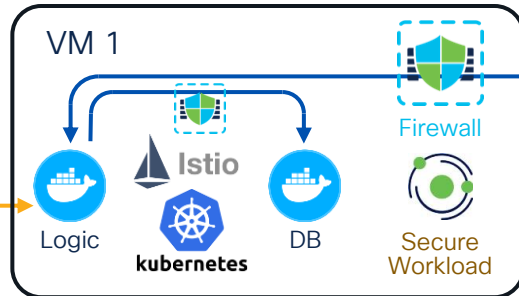
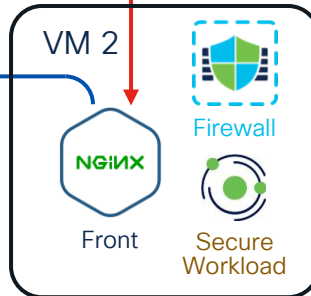
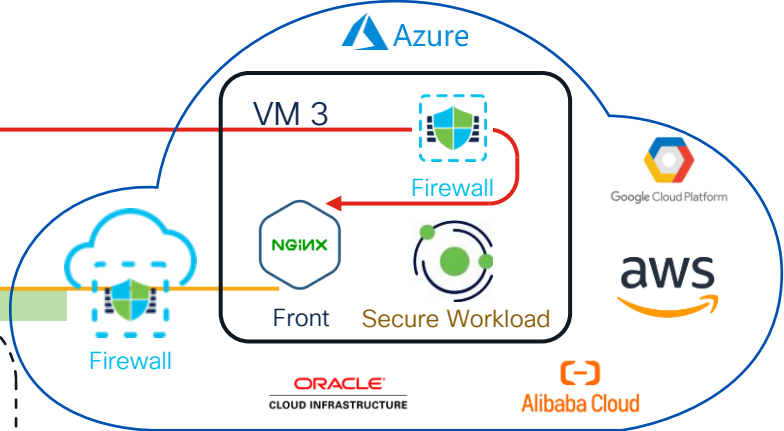
Private or cloud-delivered **Firewall** inspects application edge, implements Zero Trust Network Access (ZTNA), continuously applies full stack of inline security services.

Extranet



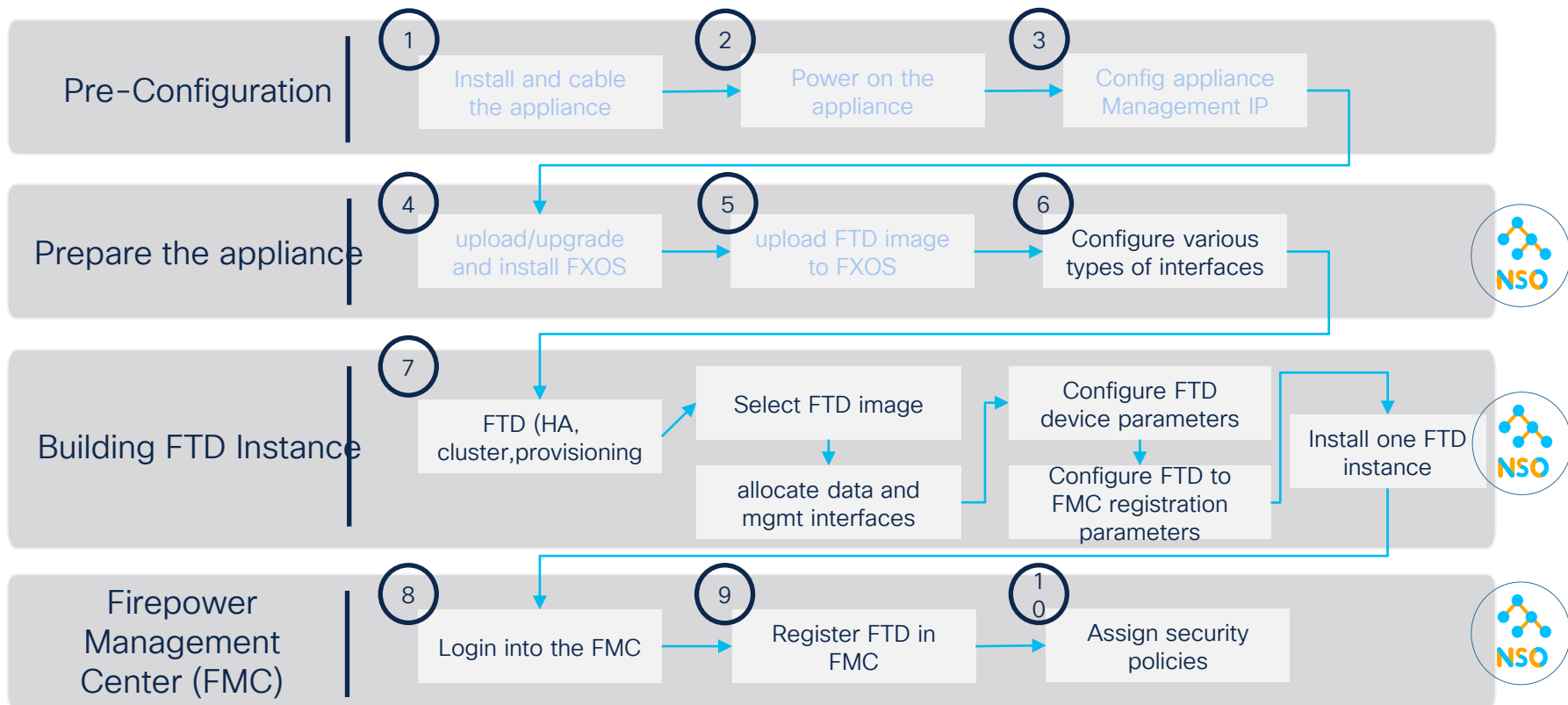
Firewall

Private Cloud



Secure Workload protects host OS at process and file levels, selectively inspects network and service mesh traffic with inline **Firewall** and API controls, integrates with public cloud and cloud-native orchestrators for posture and policy, consumes policy as code from DevOps tools.

Firepower Provisioning



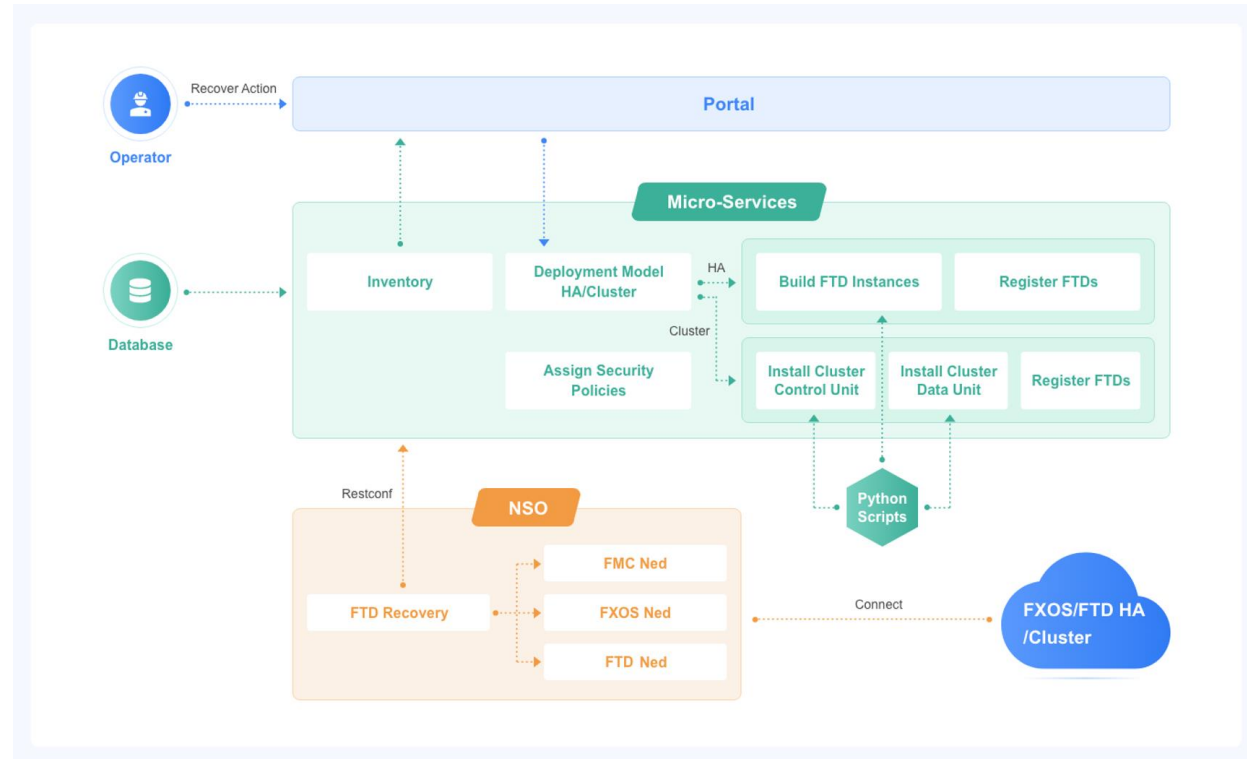
Firepower Provisioning – Firepower Solution

Who Is It For

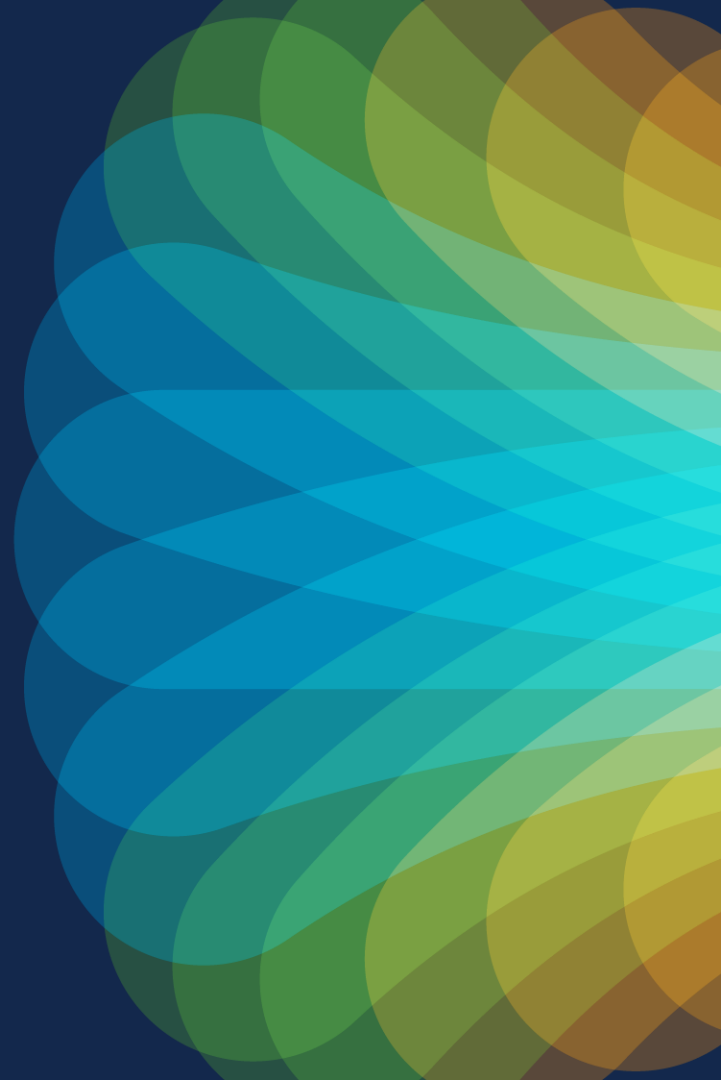
Any customer that requires automation for multiple ASAs, FTDs, or FXOS appliances.

What Does It Do

Encapsulates CLI, FXOS API , NSO NEDs as ways to be run consistently against multiple devices in parallel.



Demo



Use Case and Demo – Infrastructure Provisioning

CiscoLive 2023 Melbourne

BRKSEC-2707 | Maintenance

[Back](#)

Disaster Recovery

Inventory	HA Name	Role	Location	Management IP	Action
BRKSEC2707_FTD_1	BRKSEC2707_HA	Primary	fxos-1	10.66.181.190	Rebuild
BRKSEC2707_FTD_2	BRKSEC2707_HA	Secondary	fxos-2	10.66.181.191	Rebuild

Inventory	Cluster Name	Role	Location	Management IP	Action
BRKSEC2707_CL_FTD_1	BRKSEC2707	Control	fxos-1	10.66.181.195	Rebuild
BRKSEC2707_CL_FTD_2	BRKSEC2707	Data	fxos-2	10.66.181.196	Rebuild

Use Case and Demo – Infrastructure Provisioning

The image is a composite of three screenshots demonstrating infrastructure provisioning.

Top Left: React App - Maintenance

Disaster Recovery

Inventory	HA Name	Role	Location	Management IP	Acti
BRKSEC2707_FTD_1	BRKSEC2707_HA	Primary	fxos-1	10.66.181.190	Reb
BRKSEC2707_FTD_2	BRKSEC2707_HA	Secondary	fxos-2	10.66.181.191	Reb

Inventory	Cluster Name	Role	Location	Management IP	AI
BRKSEC2707_CL_FTD_1	BRKSEC2707	Control	fxos-1	10.66.181.195	R
BRKSEC2707_CL_FTD_2	BRKSEC2707	Data	fxos-2	10.66.181.196	R

Bottom Left: Terminal Window

```
2023-11-28 12:40:27,257 - apscheduler.executors.default - INFO - Running job "update_interval (trigger: interval[0:05:00], next run at: 2023-11-28 12:45:27 AEDT)" (scheduled at 2023-11-28 12:40:27.256556+11:00)
2023-11-28 12:40:27,259 - aps - INFO - The scheduling task interval not change, still 5 minutes
2023-11-28 12:40:27,259 - apscheduler.executors.default - INFO - Job "update_interval (trigger: interval[0:05:00], next run at: 2023-11-28 12:45:27 AEDT)" executed successfully
2023-11-28 12:40:27,261 - apscheduler.executors.default - INFO - Running job "call_api (trigger: interval[0:05:00], next run at: 2023-11-28 12:45:27 AEDT)" (scheduled at 2023-11-28 12:40:27.261012+11:00)
2023-11-28 12:40:27,262 - aps - INFO - =====> Start >>> FMC-Primary

2023-11-28 12:41:20,519 - aps - INFO - =====> Success End >>> FMC-Primary
2023-11-28 12:41:20,519 - aps - INFO - =====> Start >>> ISE
2023-11-28 12:41:24,688 - aps - INFO - =====> Success End >>> ISE
2023-11-28 12:41:24,688 - apscheduler.executors.default - INFO - Job "call_api (trigger: interval[0:05:00], next run at: 2023-11-28 12:45:27 AEDT)" executed successfully
```

Right: Firepower Chassis Manager GUI

Overview Interfaces **Logical Devices** Security Engine Platform Settings System Tools Help admin

Logical Device List (2 Container Instances) 78% (54 of 70) Cores Available

Refresh Add

Application	Version	Resource Profile	Management IP	Gateway	Management Port	Status
FTD	7.2.5.208	8CPU	10.66.181.196	10.66.181.1	Ethernet1/6	Online

Interface Name **Type** **Attributes**

- Port-channel3 data
- Port-channel4 data
- Port-channel48 cluster

Attributes

- Cluster Operational Status : in-cluster
- FIREPOWER-MGMT-IP : 10.66.181.196
- CLUSTER-ROLE : data
- CLUSTER-IP : 127.2.2.1
- MGMT-URL : https://
- UUID : b6e47edo-8ccd-11ee-9028-87c261560ef

BRKSEC2707_FTD_2 Standalone Status:ok

7 Successful Login in last 24 hrs - View Details | Mon Nov 27 2023 at 04:54:28 from - 10.137.84.67

Expand All Download Device List Report

☐	Name	Model	V..	Chassis	Licenses	Access Control Policy	Auto RollBack	
☒	BRKSEC2707_CL_FTD_1(Control) Snort 3 10.66.181.195 - Routed	Firepower 4140 with FTD	7....	FRP4100-1x Security Moc	Essentials, IPS (2 more...)	BRKSEC_2...	N/A	⋮
☒	BRKSEC2707_CL_FTD_2 Snort 3 10.66.181.196 - Routed	Firepower 4140 with FTD	7....	FRP4100-2x Security Moc	Essentials, IPS (2 more...)	BRKSEC_2...	N/A	⋮
☐	BRKSEC2707_HA High Availability							⋮

Use Case and Demo – Infrastructure Provisioning

The image displays four screenshots related to infrastructure provisioning:

- React App:** Shows a web interface for disaster recovery management. It includes a table with columns: Inventory, HA Name, Role, Location, Management IP, and Action. The table lists two FTD instances (BRKSEC2707_FTD_1 and FTD_2) and two CL_FTD instances (CL_FTD_1 and CL_FTD_2).
- Firepower Chassis Manager:** Shows the 'Logical Devices' tab. It displays a table of logical devices, including BRKSEC2707_CL_FTD_1 and BRKSEC2707_FTD_1, with details on their applications, versions, resource profiles, and management IP/gateway.
- Terminal:** Shows the output of a command executed on a root@fp-app-server:1... prompt. The output displays logs for the 'fmc_api' and 'apscheduler' components, indicating successful API calls and scheduled tasks.
- Cisco ISE:** Shows a 'Successful Login' message and a list of devices under the 'BRKSEC2707_HA' group. The devices listed are 'BRKSEC2707_FTD_1(Primary)' and 'BRKSEC2707_FTD_2(Secondary)', both with status 'Active' and 'Standby' respectively.

Automation Concepts in Policy Management

Use Case and Demo – Battle Suit Special Features

- Policy drift detection
 - FTD Policy Change Detection
 - ISE Policy Change Detection
- Firewall Access Control Policy Provisioning



CISCO *Live!*

An AI Generated
Image



The image shows a login interface for CiscoLive 2023 Melbourne. It features the Cisco logo at the top, followed by the event name and a session ID. Below this are fields for Username and Password, and a Sign In button.

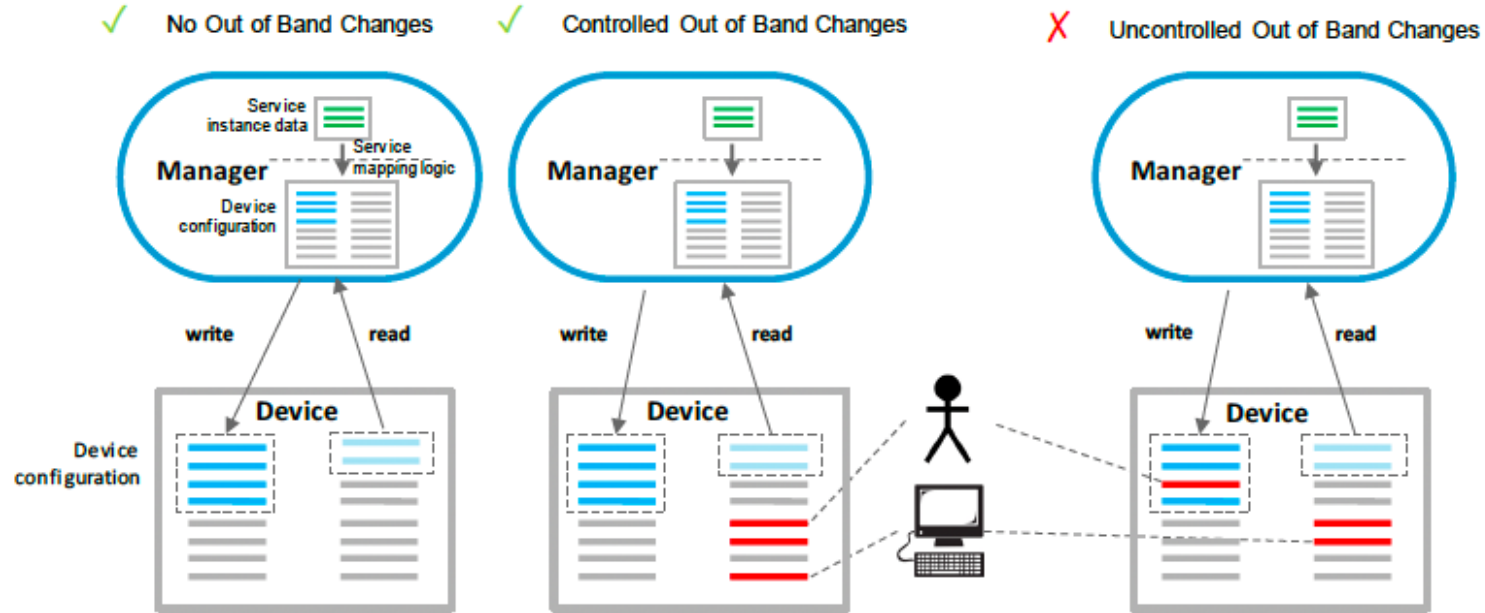
CiscoLive 2023
Melbourne
BRKSEC-2707

Username
Admin

Password

Sign In

Policy Drift Detection



Policy Drift Detection

- Policy drift detection
 - Firepower
 - ISE
- Transaction ID to create working configuration template, customer can mark it as the golden template.
- Tools such as NSO to query ISE PAN(configurable time interval), Firepower FMC and routers/ switches to identify policy deviation.
- Report in the alarm section
 - acknowledgeable by admin
 - admin can issue authorization to revert config back to the marked golden template

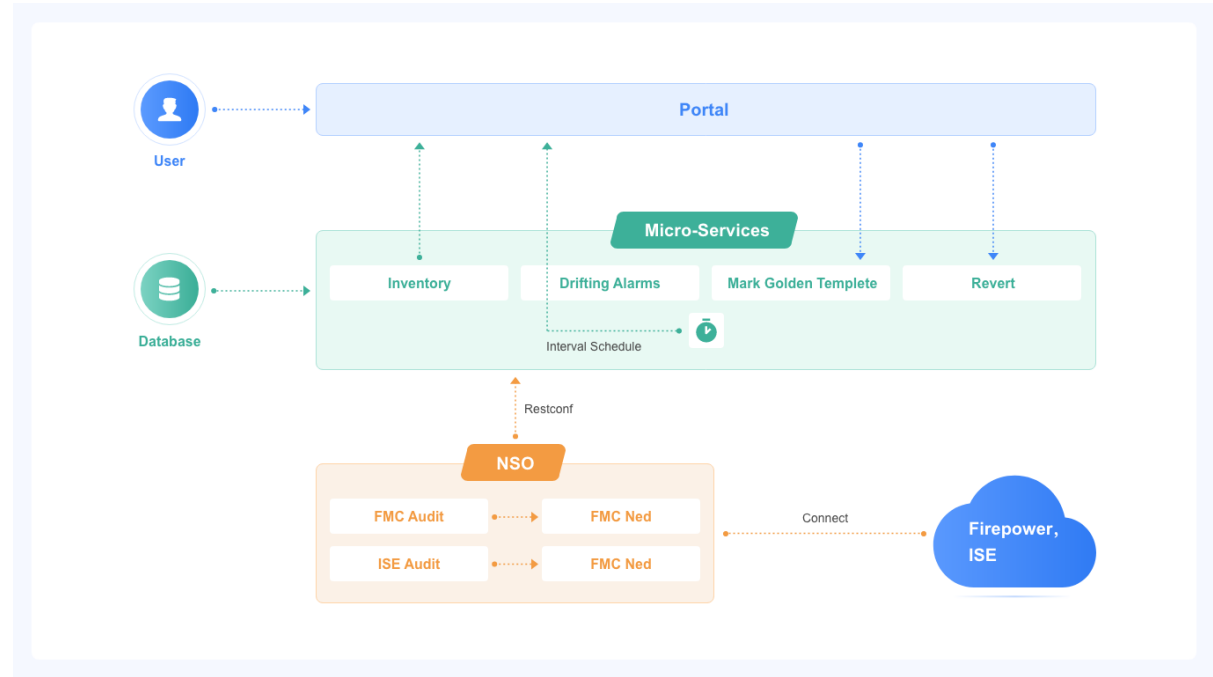
Policy Drift Detection

Who Is It For

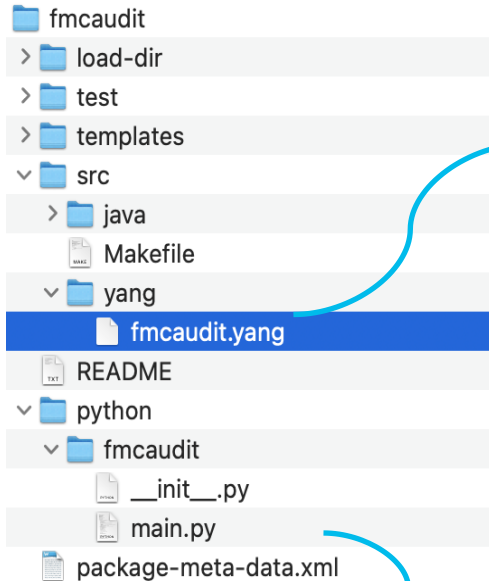
- Unknown device state due to multiple groups executing concurrent changes.
- Inconsistency between desired state and the running state.

What Does It Do

Encapsulates CLI, FXOS API , NSO NEDs as ways to be run consistently against multiple devices in parallel.



Policy Drift Detection



```
fmcaudit.yang

container action {
  tailf:action synctofmc {
    tailf:actionpoint 'syncto-actionpoint';
    input {
    }
    output {
      leaf status {
        type uint16;
      }
      leaf message {
        type string;
      }
    }
  }
  tailf:action compare {
    tailf:actionpoint 'compare-actionpoint';
    input {
    }
    output {
      leaf status {
        type uint16;
      }
      leaf message {
        type string;
      }
    }
  }
  tailf:action rebasefromfmc {
    tailf:actionpoint 'rebasefromfmc-actionpoint';
    input {
    }
    output {
      leaf status {
        type uint16;
      }
      leaf message {
        type string;
      }
    }
  }
}
```

<https://{{nso-server}}:{{nso-port}}/restconf/data/fmcaudit:compare>

<https://{{nso-server}}:{{nso-port}}/restconf/data/fmcaudit:rebasefromfmc>

<https://{{nso-server}}:{{nso-port}}/restconf/data/fmcaudit:synctofmc>

Restconf

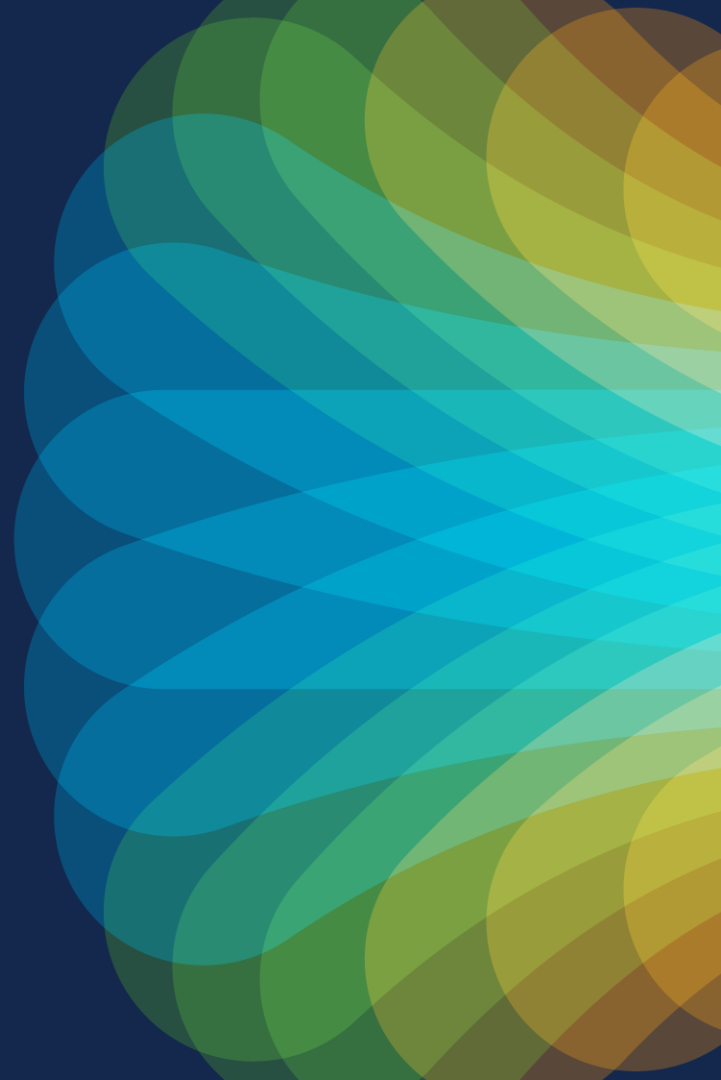
```
class synctofmc(ncs.dp.Action):
class compare(ncs.dp.Action):
class rebase(ncs.dp.Action):
```

Implement Code Base

Invoke

cisco-fmc-gen-1.6

Demo



Use Case and Demo – Policy Drift Detection

CiscoLive 2023 Melbourne

BRKSEC-2707 | Security

Back

Drifting Detection

NO.	Model	Host Name	Management IP	Version
-----	-------	-----------	---------------	---------

root@fp-app-server:~#

```
2023-11-28 12:15:27,261 - apscheduler.executors.default - INFO - Running job "call_api (trigger: interval[0:05:00], next run at: 2023-11-28 12:20:27 AEDT)" (scheduled at 2023-11-28 12:15:27,261812+11:00)
2023-11-28 12:15:27,261 - aps - INFO - ===== Start >>> FMC-Primary
2023-11-28 12:16:25,295 - aps - INFO - ===== Success End >>> FMC-Primary
2023-11-28 12:16:25,295 - aps - INFO - ===== Start >>> ISE
2023-11-28 12:16:29,717 - aps - INFO - ===== Success End >>> ISE
2023-11-28 12:16:29,717 - apscheduler.executors.default - INFO - Job "call_api (trigger: interval[0:05:00], next run at: 2023-11-28 12:20:27 AEDT)" executed successfully
2023-11-28 12:17:00,957 - werkzeug - INFO - 10.67.182.89 -- [28/Nov/2023 12:17:00] "GET /api/security/ HTTP/1.1" 200 -
2023-11-28 12:17:03,290 - werkzeug - INFO - 10.67.182.89 -- [28/Nov/2023 12:17:03] "GET /api/security/ HTTP/1.1" 200 -
```

Identity Services Engine

Intuitive network security

Username

Password

Login

[English](#) | [日本語](#)

[Problems logging in?](#)

© 2022 Cisco Systems, Inc. Cisco, Cisco Systems and Cisco Systems logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries. Cisco ISE utilizes open source software from various components. [View third-party licenses and notices](#)

Use Case and Demo

- Policy drifting detection
 - FTD Policy Change Detection
 - ISE Policy Change Detection
- Firewall Access Control Policy Provisioning



CISCO *Live!*

An AI Generated
Image

A screenshot of a login interface for CiscoLive 2023 Melbourne. The interface is white with a blue background. It features the Cisco logo at the top, followed by the event name "CiscoLive 2023 Melbourne" and the session ID "BRKSEC-2707". Below this, there are input fields for "Username" (containing "Admin") and "Password" (containing "*****"). A blue "Sign In" button is at the bottom.

Cisco

CiscoLive 2023
Melbourne

BRKSEC-2707

Username

Admin

Password

Sign In

Firewall Policy Provisioning – Path Finding

Requests:

- Can you configure the FW01 & FW02 firewalls to allow my access?
- Can you configure firewall to allow my access?
- What vendor/type of firewalls are present in path?
Who can configure? When?
- Which team can configure the firewalls in my path to access AAA server?
- How many firewalls are there in my path to access AAA server?

.....



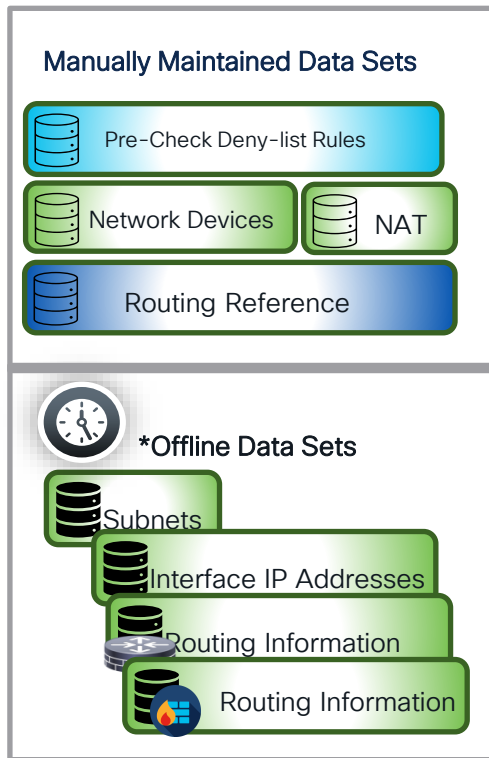
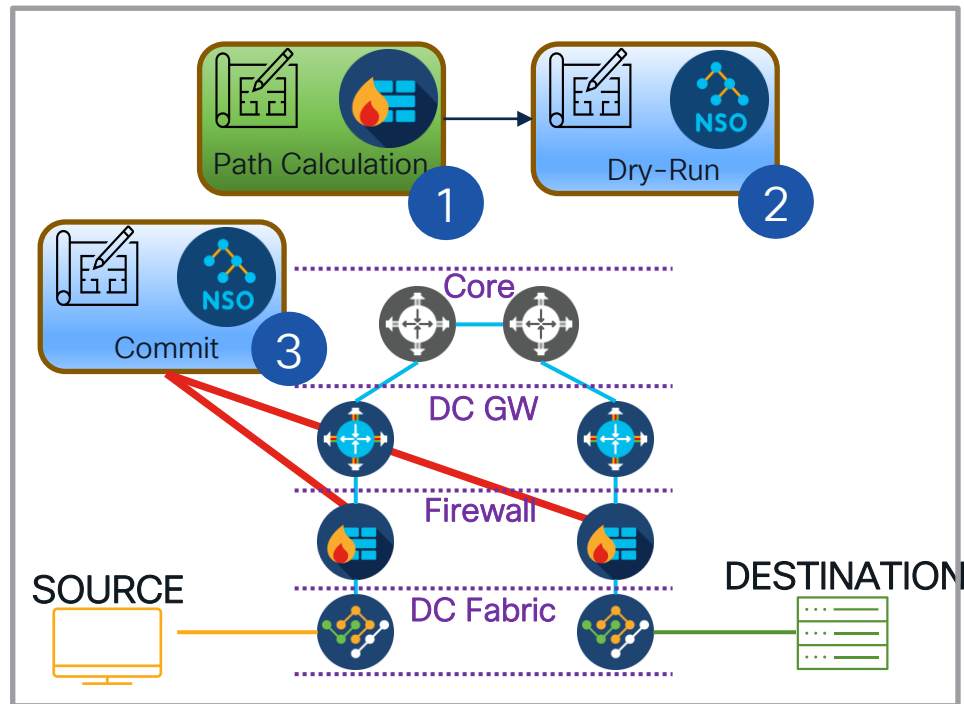
Challenges:

- Find **the path** and identify **firewalls** between source to destination
- Find the **teams** controlling the firewalls
- **Multi-vendor** firewalls
 - technology & skills
- **Change Management & Approvals**
 - Long & cumbersome approval process
- **Implementation Challenges**
 - Availability of skilled resources
 - Errors, **rollback**, **dry-run**, **post-checks**, reporting

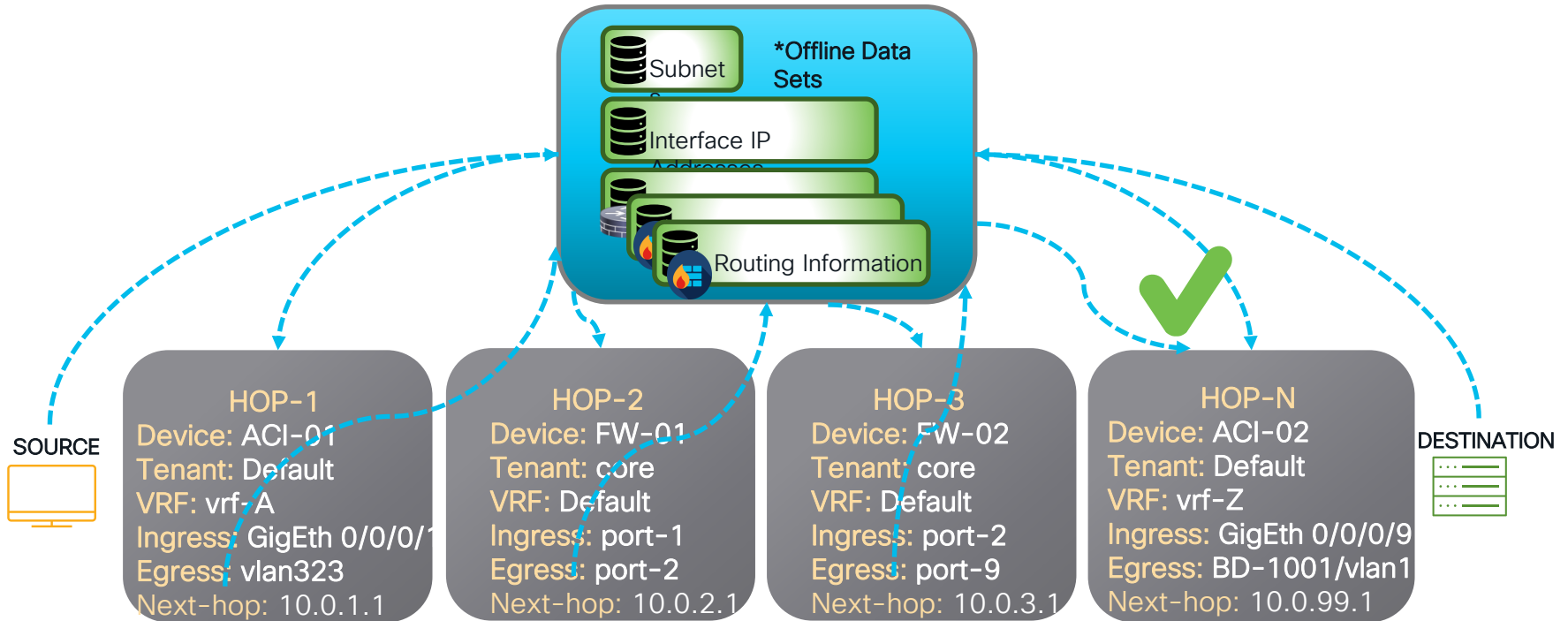
...

Firewall Policy Provisioning - Path Finding

request intent! allow access from SOURCE to DESTINATION on PORT



Firewall Policy Provisioning - Path Finding



Firewall Policy Provisioning - Path Finding



What is the path from source to destination?

Path Group

SRC IP: 192.168.110.25, Dest IP: 192.168.112.73

Forward Path:



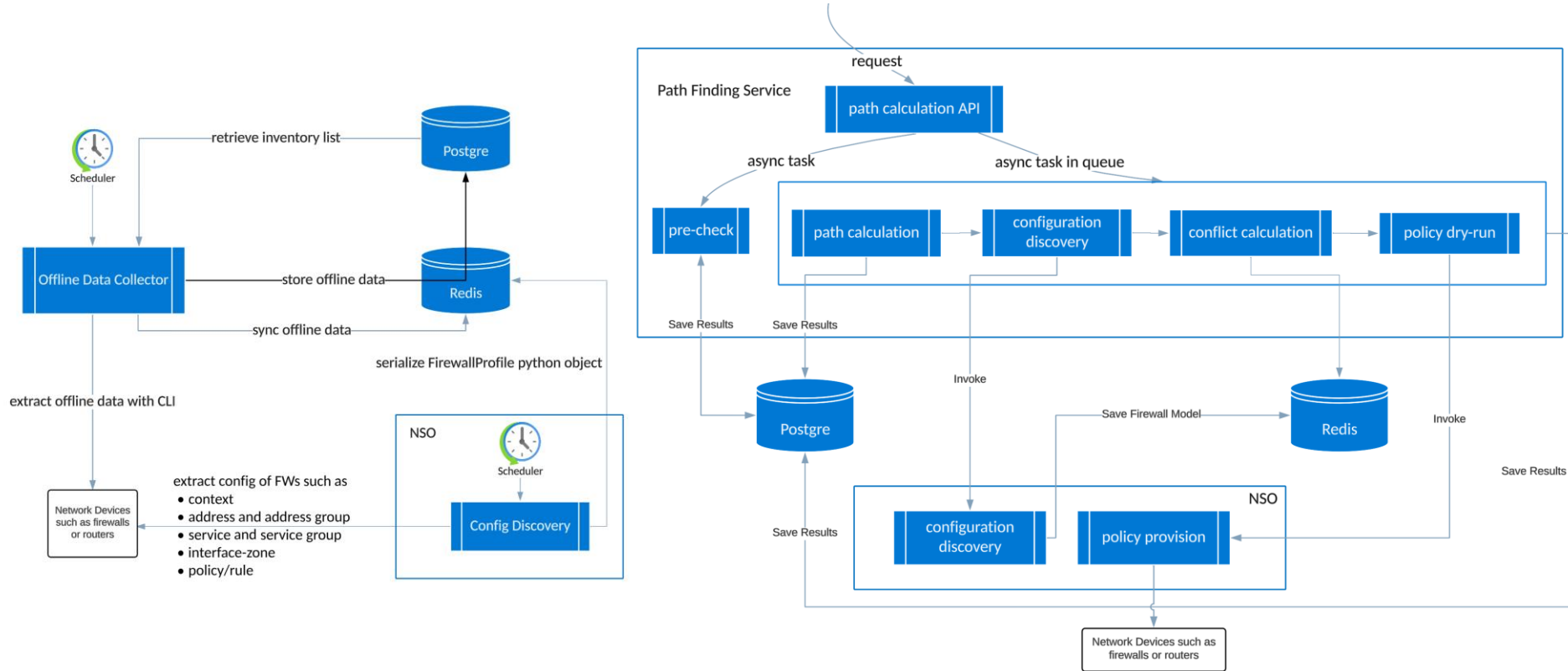
Routing Evidence:

Device	Context	VRF	Destination IP	Egress Interface	Next Hop	Next Hop Node Name	Proposed Routing
APIC	Network.Sector	VRF107	0.0.0.0/0	ETH1/1,ETH1/2	10.26.0.1	FTD-1	No
FTD-1	Default	Default	0.0.0.0/0	ae1.1995	10.20.8.33	DC-Core	No
DC-Core	Default	Default	0.0.0.0/0	Vlan5	10.20.8.9	FTD-2	No
FTD-2	WAN	Default	10.0.0.0/8	EDN-130	10.20.4.1	SC-RDC47-02-net	No
SC-RDC47-02-net	Default	EDN	10.32.10.0/24	Po421.1	10.4.21.234	SC1160501-Core	No
SC1160501-Core	WAN	Default	10.32.10.0/24	ETH1/8	10.30.0.18	SC1160501-AGG	No
SC1160501-AGG	Default	Default	0.0.0.0/0	Vlan20	0.0.0.0	N/A	No

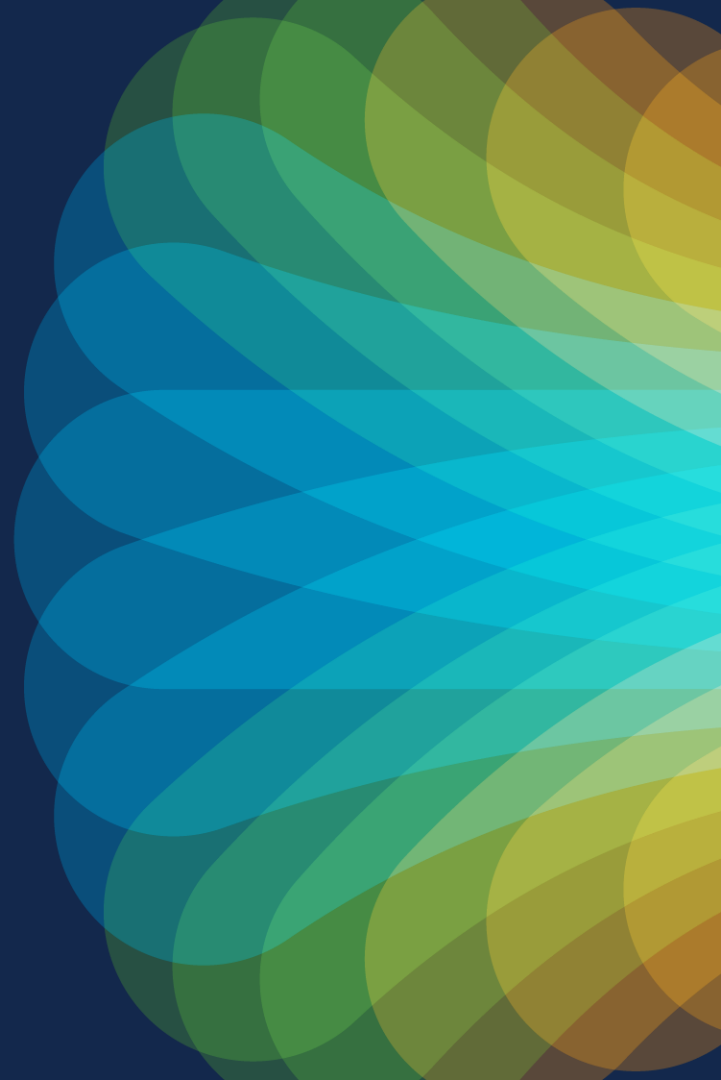
Dry-Run Result

```
Device      Dry-Run Result
FTD-1      ["FTD-1": {
  cisco-fmc:policy {
    accesspolicies FTD_Policy {
      # after accessrules abde
      + accessrules USC6000254_rule_3_1 {
      +   action ALLOW;
      +   sendEventsToFMC true;
      +   logFiles false;
      +   logBegin false;
      +   logEnd true;
      +   sourceZones {
      +     objects outside;
      +   }
      +   destinationPorts {
      +     objects NTP-TCP {
      +       protocol TCP
      +       type ProtocolPortObject;
      +     }
      +   }
      +   destinationZones {
      +     objects inside;
      +   }
      +   sourceNetworks {
      +     literals 192.168.110.25;
      +   }
      +   destinationNetworks {
      +     literals 192.168.112.73
      +   }
      + }
      + enabled;
    }
  }
}
```

Firewall Policy Provisioning - Path Finding



Demo



Use Case and Demo – Path Finding

A screenshot of a login interface for CiscoLive 2023 Melbourne. The interface is centered on a dark blue background with abstract, wavy, light blue lines. A white rectangular box contains the Cisco logo at the top, followed by the text "CiscoLive 2023 Melbourne" and "BRKSEC-2707". Below this, there are two input fields: "Username" with a user icon and the text "Admin", and "Password" with a key icon and masked characters "*****". A blue "Sign In" button is at the bottom of the white box.


CiscoLive 2023
Melbourne
BRKSEC-2707

Username

Password

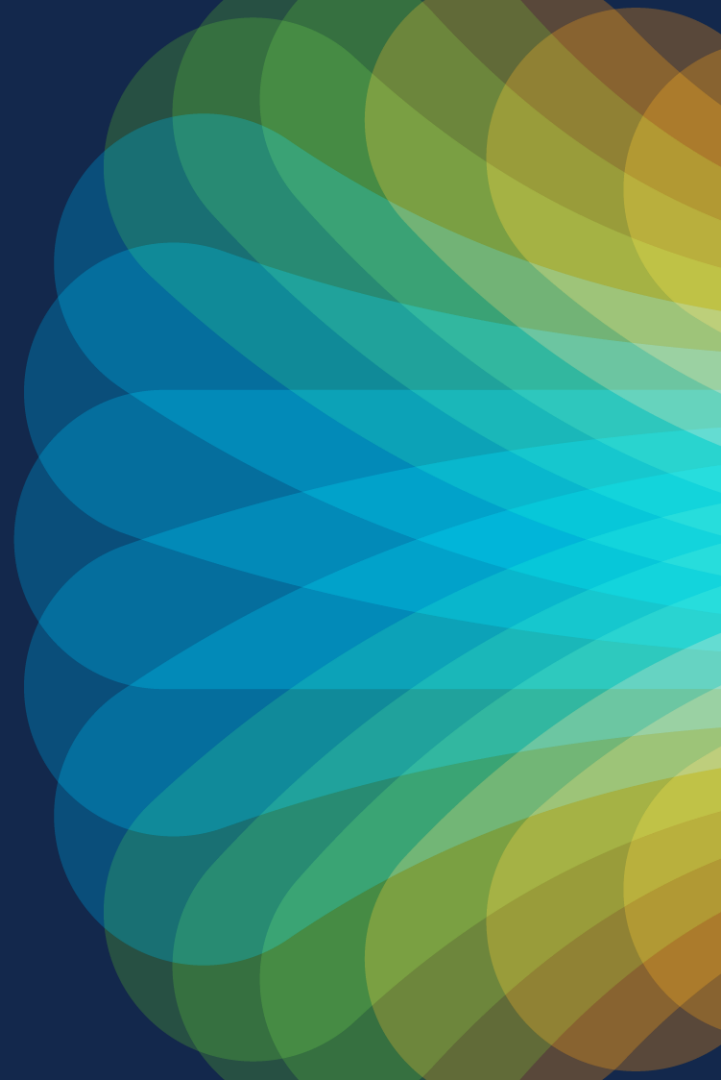
[Sign In](#)

Value Proposition Summary

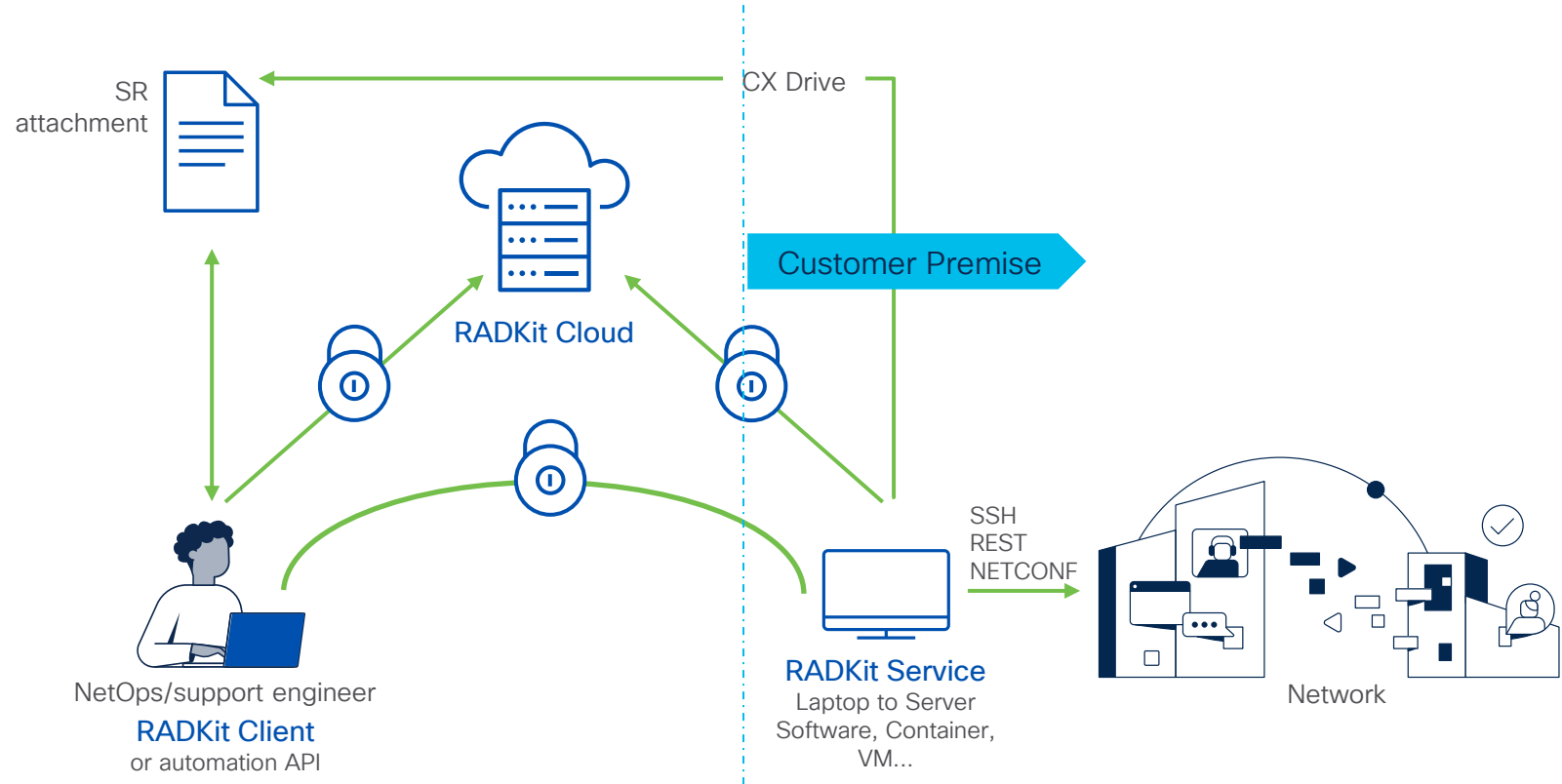
Use Case	Before Automation Baseline		After Automation Improvement
2 x Firepower Instance Provisioning	~ 1 hours		~15 - 20 mins
Path ACL Rules Provisioning	~		~ 2 - 3 mins
Policy Change Detection	~		~ 1-5 mins

One More Thing...

NetOps using Radkit



RADKit General Architecture



RADKit: Query at scale

1. Grouping devices
2. Collecting basic information from multiple devices
3. Simultaneous command collection
4. Parse the output: RADkit genie

RADKit: Authenticate client with certificate

```
from radkit_client import certificate_login

def get_client():
    print("Creating new Radkit session")
    client = certificate_login(
        identity="hasm@cisco.com",
        ca_path="./certs/chain.pem",
        cert_path="./certs/cert.pem",
        key_path="./certs/cert_key.pem",
        private_key_password=os.environ["CERT_KEY"],
    )
    return client
```

RADKit: Connect to service

```
from radkit_client import create_context, certificate_login

with create_context():
    client = get_client()
    svc = client.service("legv-ek83-27ez").wait()
```

1. Connect to service
2. Wait makes the connection synchronous
3. Inventories are fetched

RADKit: Get output from devices

```
sw1 = svc.inventory["pod3-sw1"]  
cmd_output = sw1.exec("show version")  
  
ios_devices = svc.inventory.filter("device_type", "IOS")  
cmd_output = ios_devices.exec("show version").wait()
```

1. Select single device
2. Filter devices based on attributes
3. Get the output of any commands
4. Use array, to get output of multiple commands

RADKit: Parse output with Genie

```
>>> import radkit_genie as rkg
>>>
>>> cmd = svc.inventory['pod3-sw1'].exec("show version").wait()
>>> parsed = rkg.parse(cmd, os="iosxe")
>>> parsed['pod3-sw1']['show version'].data['version'] | print
{'xe_version': '16.09.04', 'version_short': '16.9', 'platform': 'Catalyst L3 Switch', 'version': '16.9.4', 'image_id': 'CAT3K_CAA-UNIVERSALK9-M', 'label': 'RELEASE SOFTWARE (fc2)', 'os': 'IOS-XE', 'location': 'Fuji', 'image_type': 'production image', 'copyright_years': '1986-2019', 'compiled_date': 'Thu 22-Aug-19 17:33', 'compiled_by': 'mcpre', 'rom': 'IOS-XE ROMMON', 'bootldr': 'CAT3K_CAA Boot Loader (CAT3K_CAA-HBOOT-M) Version 4.78, RELEASE SOFTWARE (P)', 'hostname': 'Switch', 'uptime': '26 weeks, 3 days, 23 hours, 16 minutes', 'uptime_this_cp': '26 weeks, 3 days, 23 hours, 19 minutes', 'returned_to_rom_by': 'Power Failure or Unknown', 'system_image': 'flash:cat3k_caa-universalk9.16.09.04.SPA.bin', 'last_reload_reason': 'Power Failure or Unknown', 'license_package': {'ipservicesk9': {'license_level': 'ipservicesk9', 'license_type': 'Smart License', 'next_reload_license_level': 'ipservicesk9'}, 'None': {'license_level': 'None', 'license_type': 'Subscription Smart License', 'next_reload_license_level': 'None'}}, 'chassis': 'WS-C3850-48P', 'main_mem': '832441', 'processor_type': 'MIPS', 'rtr_type': 'Edison', 'chassis_sn': 'FOC2032U1GE', 'number_of_intfs': {'Virtual Ethernet': '2', 'Gigabit Ethernet': '52', 'Ten Gigabit Ethernet': '4'}, 'mem_size': {'non-volatile configuration': '2048', 'physical': '4194304'}, 'disks': {'crashinfo.:': {'disk_size': '252000', 'type_of_disk': 'Crash Files'}, 'flash.:': {'disk_size': '1611414', 'type_of_disk': 'Flash'}, 'webui.:': {'disk_size': '0', 'type_of_disk': 'WebUI ODM Files'}}, 'curr_config_register': '0x102', 'switch_num': {'2': {'ports': '56', 'model': 'WS-C3850-48P', 'sw_ver': '16.9.4', 'sw_image': 'CAT3K_CAA-UNIVERSALK9', 'mode': 'BUNDLE', 'uptime': '26 weeks, 3 days, 23 hours, 19 minutes', 'active': True, 'mac_address': '00:b0:e1:bf:59:80', 'mb_assembly_num': '73-15800-08', 'mb_sn': 'FOC20325CMR', 'model_rev_num': 'AB0', 'mb_rev_num': 'A0', 'model_num': 'WS-C3850-48P', 'system_sn': 'FOC2032U1GE'}}}
```

1. Command output as json
2. Supports only windows
3. Not available for all commands / products

RADKit standalone script contd...

```
with create_context():
    client = get_client()
    svc = client.service("legv-ek83-27ez").wait()
    ios_devices = svc.inventory.filter("device_type", "IOS")
    cmd_output = ios_devices.exec("show version").wait()
    out_parsed = rkg.parse(cmd_output, os="iosxe")
    for dev in out_parsed.keys():
        logger.info(f"Version of {dev} : {out_parsed[dev]['show version'].data['version']['version']}")
```

Connects to radkit service

Group device using Filter

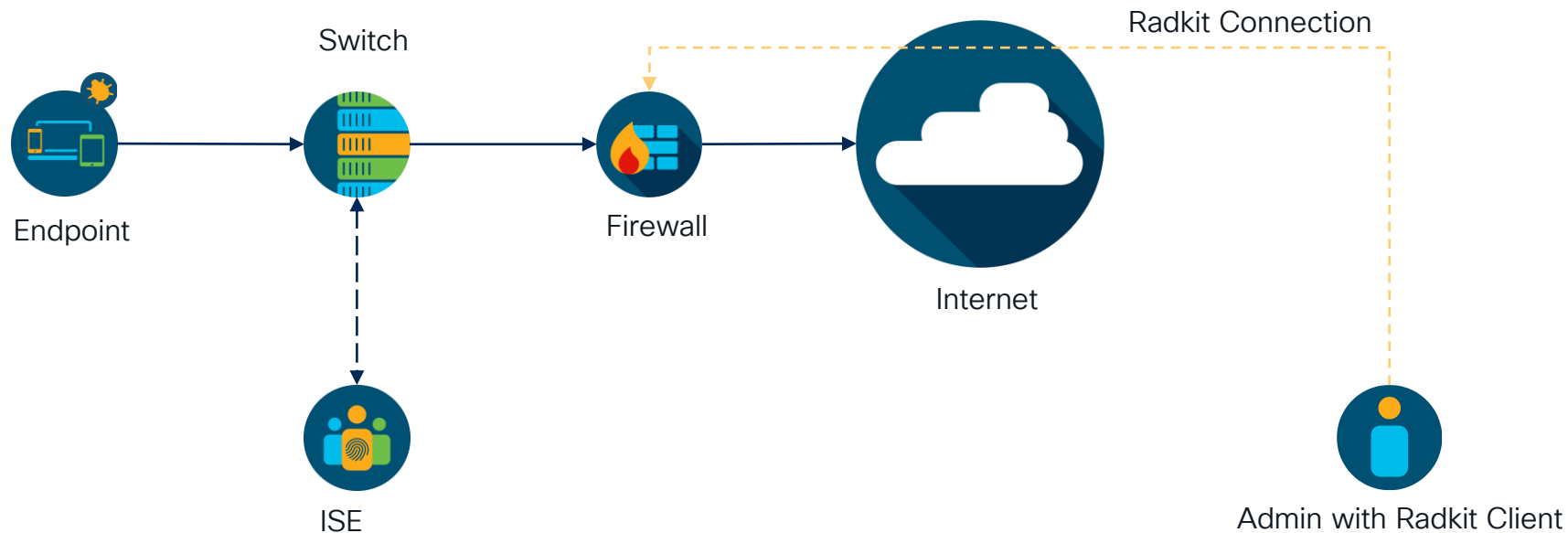
Get output from filtered device

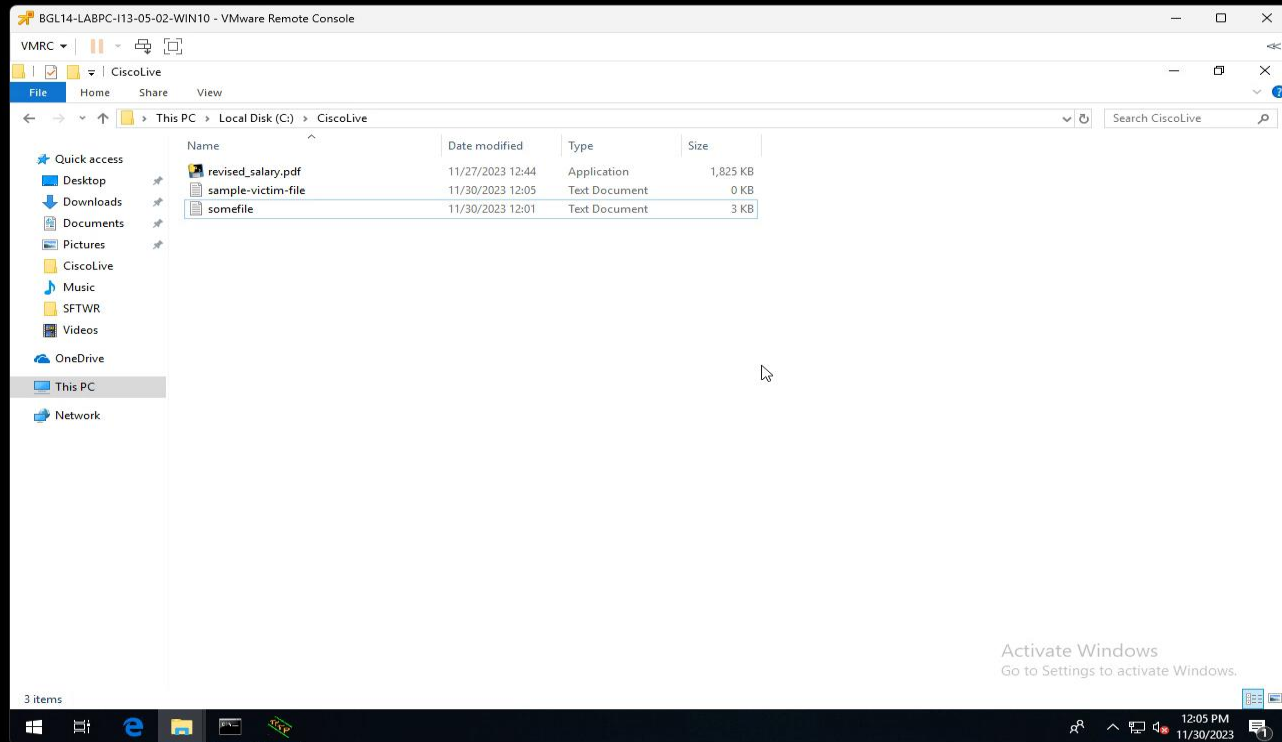
Parse the output using Genie

RADKit : Script output

```
hasm@lab-device:/ciscolive$ python demo1.py
Creating new Radkit session
26-Nov-2023 10:19:31 INFO      Client Created
26-Nov-2023 10:19:46 INFO      Filtered all IOS devices
26-Nov-2023 10:19:59 INFO      Gathered show version from all IOS devices
26-Nov-2023 10:19:59 INFO      Parsed the output using RADKit Genie
26-Nov-2023 10:19:59 INFO      Version of pod2-sw1 : 16.3.2
26-Nov-2023 10:19:59 INFO      Version of pod1-sw3 : 15.2(4)E1
26-Nov-2023 10:19:59 INFO      Version of pod2-sw2 : 16.6.4
26-Nov-2023 10:19:59 INFO      Version of pod3-sw1 : 16.9.4
26-Nov-2023 10:19:59 INFO      Version of pod1-sw2 : 16.9.4
26-Nov-2023 10:19:59 INFO      Version of pod1-sw1 : 15.2(7)E3
```

RADKit : Threat hunting





RADKit : FMC API

```
def get_incidents(fmc, uuid):
    api = fmc.swagger.paths['/api/fmc_tid/v1/domain/{domainUUID}/tid/incident']
    api_call = api.get(domainUUID=uuid).wait()
    resp = api_call.result.json
    return resp

def get_incident_details(fmc, uuid, id):
    detail = {}
    uuid = fmc.swagger.metadata['domains']['Global']
    api = fmc.swagger.paths['/api/fmc_tid/v1/domain/{domainUUID}/tid/incident/{objectId}']
    api_call = api.get(domainUUID=uuid, objectId=id).wait()
    resp = api_call.result.json
    return resp

with create_context():
    fmc = svc.inventory['demofmc']
    fmc.update_swagger().wait()
    uuid = fmc.swagger.metadata['domains']['Global']
    all_incidents = get_incidents(fmc, uuid)
    for incident in all_incidents:
        details = get_incident_details(fmc, uuid, incident)
```

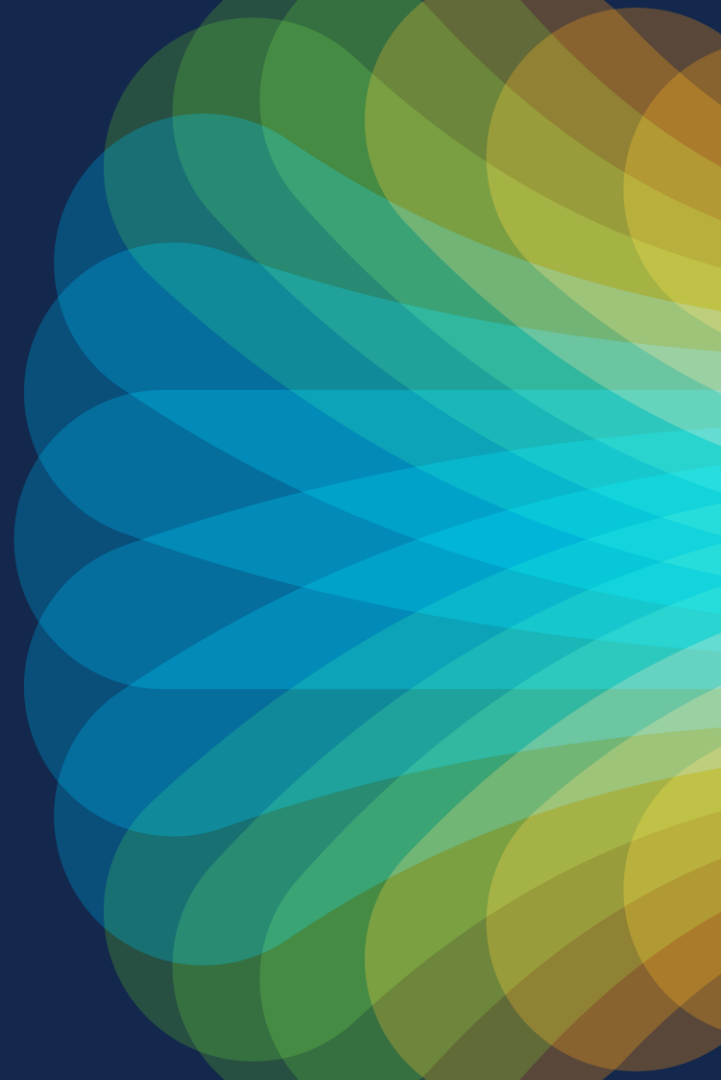
RADKit : ISE API

```
def get_session_from_ise(ise, ip):
    api = ise.http.get('/admin/API/mnt/Session/ActiveList', headers={"Content-Type": "application/xml", "Accept": "application/xml"}).wait()
    sess_dict = json.loads(json.dumps(xmltodict.parse(api.result.text)))
    sessions = []
    mac = ""
    if isinstance(sess_dict["activeList"]["activeSession"], dict):
        sessions.append(sess_dict["activeList"]["activeSession"])
    else:
        sessions = sess_dict["activeList"]["activeSession"]
    for sessn in sessions:
        if sessn["framed_ip_address"] == ip:
            mac = sessn["calling_station_id"]
    api = ise.http.get(f'/admin/API/mnt/Session/MACAddress/{mac}', headers={"Content-Type": "application/xml", "Accept": "application/xml"}).wait()
    details_dict = json.loads(json.dumps(xmltodict.parse(api.result.text)))
    sess_dict = details_dict['sessionParameters']
    return sess_dict

def ise_block_endpoint(ise, mac):
    endpointId = ""
    api = ise.http.get(f'/ers/config/endpoint/name/{mac}').wait()
    resp = api.result.json
    endpointId = resp['ERSEndPoint']['id']

    indata = {"ERSEndPoint" : {'groupId': 'aa000c30-8bff-11e6-996c-525400b48521', 'staticGroupAssignment': True, 'mac': mac}}
    api = ise.http.put(f'/ers/config/endpoint/{endpointId}', json=indata, headers={"Content-Type": "application/json", "Accept": "application/json"}).wait()
    return
```

Wrap Up



Wrap Up

- Summary
- Resource list
 - NSO: developer.cisco.com/docs/search/?products=NSO
 - Firepower: developer.cisco.com/docs/search/?products=Firepower
 - ISE:
developer.cisco.com/docs/search/?products=Identity+Services+Engine
 - Security: developer.cisco.com/docs/search/?categories=Security
 - CX Security IaC demo in the World of Solution
 - Cisco RADKit: <https://radkit.cisco.com>

CISCO *Live!*

Did you know?

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

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



Meet the Expert Opening Hours:

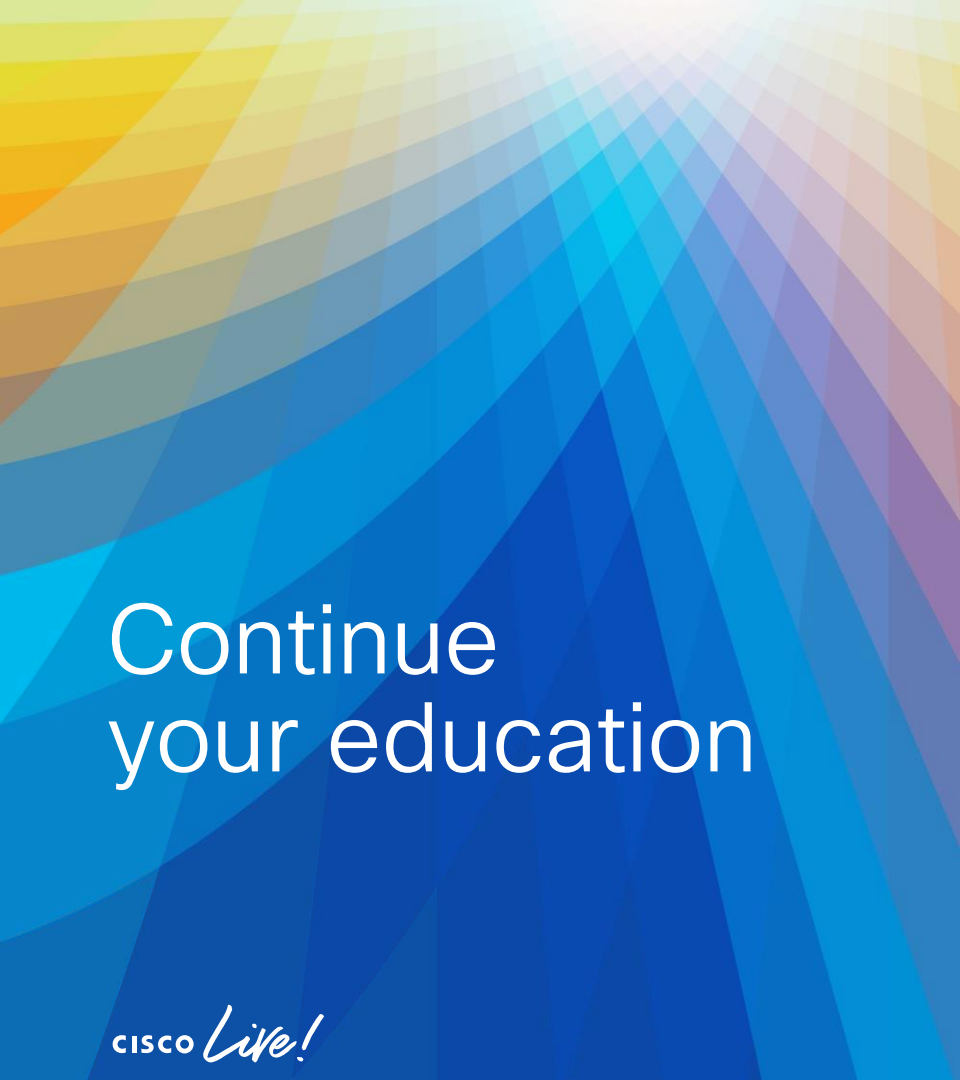
Tuesday	3:00pm – 7:00pm
Wednesday	11:15am – 7:00pm
Thursday	9:30am – 4:00pm
Friday	10:30am – 1:30pm

Session Surveys

We would love to know your feedback on this session!

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





Continue your education



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



The bridge to possible

Thank you

CISCO *Live!*

#CiscoLiveAPJC

The background features a vibrant, multi-colored abstract design. On the left, there are horizontal, wavy bands of color in shades of red, orange, yellow, and green. On the right, a bright white light source emits a series of colorful rays in shades of blue, green, and yellow, creating a sunburst effect.

cisco *Live!*

Let's go

#CiscoLiveAPJC