



i i i i i i i i

You make **possible**

A horizontal row of nine stylized letter 'i' characters, each composed of a vertical bar with a small circular dot at the top. The colors of the bars alternate between blue, green, orange, and red. Below this graphic, the text "You make **possible**" is written in a white, sans-serif font. The first four words ("You make") are in a smaller font weight, while "possible" is in a bold font weight.



ConfigMon

Configuration Monitoring and
Compliance using Cisco DNA Center

Gabriel Zapodeanu
Technology Solutions Architect, Cisco Systems

 gzapodea@cisco.com

 @zapodeanu

GitHub github.com/zapodeanu

DEVWKS-2840

Cisco *live!*
June 9-13, 2019 • San Diego, CA

#CLUS



Cisco Webex Teams

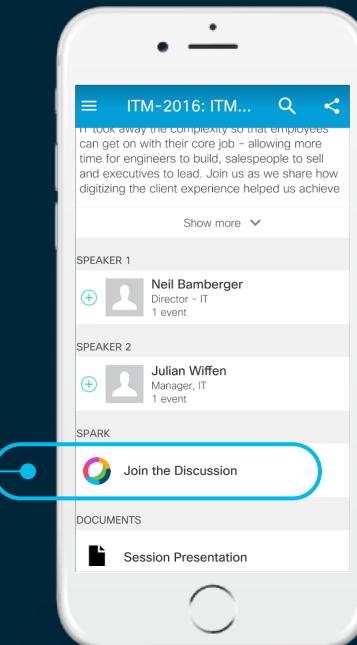
Questions?

Use Cisco Webex Teams 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 Webex Teams or go directly to the team space
- 4 Enter messages/questions in the team space

Webex Teams will be moderated by the speaker until June 16, 2019.



Gabriel Zapodeanu

- Technical Marketing Engineer, ENB, Cisco Systems



- Gabriel Zapodeanu is a Network Programmability Technical Marketing Engineer.
- He recently joined the Cisco Enterprise Networking Business, focused on controllers programmability – Cisco DNA Center, SD-WAN and ISE.
- Prior to his current role he was a Programmability Technology Architect, part of the Global Partner Organization, focused on partner technical enablement programs, APIs use case development across various technologies, as well as learning and certifications programs.
- After joining Cisco in 2006 he was an Systems Engineer in the West Area Enterprise Organization, Oregon USA. His work included large architectures for a variety of Enterprise customers from manufacturing, utilities, retail, transportation, healthcare, education, and financial industries.
- Gabriel completed his Cisco Business Architecture Certification #373 ,and recently, co-authored the book "IOS XE Programmability: Automating Device Lifecycle Management".

ConfigMon

Configuration Monitoring and Compliance using Cisco DNA Center

- The Challenge

- 70% of policy violations are due to user errors
- Configuration drifting

- The Goals

- Automated roll back of non-compliant changes
- Alert on all network configuration changes

- The Solution

- Integration between DNA Center, Cisco IOS XE, and Webex Teams
- ITSM Integration and Change Control Approval Process

- The Results

- Non-compliant configuration changes are mitigated in minutes
- Real time view of all device configuration changes

ConfigMon App

DNA Center

The screenshot shows the Cisco DNA Center Platform interface. In the top navigation bar, the 'Platform' tab is selected. Under the 'Developer Toolkit' section, there is a 'Command Runner' tool. It has two tabs: 'GET' and 'POST'. The 'GET' tab is active, showing the URL: `/api/v1/network-device/commands/read-only`. Below the URL, there are two buttons: 'Get all keywords of CLIs accepted by command runner' and 'Submit request for read-only CLIs'. To the right of the Command Runner, there is a 'Network Discovery' section.

```
def main():
    """The script will monitor device configuration changes. It could be executed on demand as in this lab,
    periodically (every 60 minutes, for example) or continuously.

    If a device has a configuration file for each DNA Center managed device, compare the existing cached file,
    with changes detected, identify the last user that configured the device, and create a new ServiceNow incident.

    Configuration changes can be due to device configuration, new configurations or approved in ServiceNow.

    Configuration changes can be due to:
        - no Access Control Lists changes
        - no IP address changes
        - no duplicated IPv4 addresses
    """

    # get the DNA C auth token
    dna_c_token = dna_api.get_dna_c_token(DNA_C_AUTH)
    print("DNA C auth token : ", dna_c_token['token'])

    # get the DNA C managed devices (all Extended wireless, for one location)
    all_devices_info = dna_api.get_all_device_info(dna_c_token)

    for device in all_devices_info:
        if device['fqdn'] == 'IOSXE':
            Routers = []
            for dev in device['hostnames']:
                if dev['fqdn'] == 'IOSXE':
                    Routers.append(dev['hostname'])
            all_devices_hostnames.append(device['hostname'])

    # get the DNA C auth token
    dna_c_token = dna_api.get_dna_c_token(DNA_C_AUTH)
    print("DNA C auth token : ", dna_c_token['token'])

    # get the DNA C managed devices (all Extended wireless, for one location)
    all_devices_info = dna_api.get_all_device_info(dna_c_token)

    for device in all_devices_info:
        if device['fqdn'] == 'IOSXE':
            Routers = []
            for dev in device['hostnames']:
                if dev['fqdn'] == 'IOSXE':
                    Routers.append(dev['hostname'])
            all_devices_hostnames.append(device['hostname'])
```

ServiceNow

The screenshot shows a ServiceNow Incident view for incident `INC00000004`. The summary indicates it's an 'Configuration Change Alert - IOSXE' created by user `IOSXE` on 2019-10-23 22:06:01. The details show the device is named `NYC-9300`, located in `Cisco New York`, and has management IP address `10.91.10.0`. The configuration change involved changing the `mg_server` from `17.16.32.40` to `17.16.32.43`. The incident was last updated on 2019-10-23 22:06:01 by `IOSCE`.



Open IOS XE, Guest Shell

ConfigMon Policies

- Rules:
 - No Access Control configurations changes
 - No logging configurations changes
 - Configurations should not create duplicated IPv4 addresses with clients or network devices interfaces
- Actions:
 - If not compliant, role back the entire configuration, test if rollback successful
 - If compliant, create incident in ServiceNow, ask for Change Control Manager to approve - save or rollback configuration
- Easy to create additional rules



Demo: ConfigMon

ConfigMon App

DNA Center

The screenshot shows the Cisco DNA Center Platform interface. In the top navigation bar, 'DESIGN', 'POLICY', 'PROVISION', 'ASSURANCE', and 'PLATFORM' are listed, with 'PLATFORM' being the active tab. Below the navigation is a sub-menu for 'Developer Toolkit' and 'Runtime Dashboard'. Under 'APIs', there's a 'Command Runner' section. It includes a table with two rows: one for 'Get all keywords of CLIs accepted by command runner' (Method: GET) and another for 'Get valid keywords' (Method: POST). Buttons for 'Run read-only commands on devices to get their real-time configuration' and 'Submit request for read-only CLIs' are also present.

This screenshot displays a configuration script intended for an Arista 7050 device. The script starts with a header indicating it will merge device configuration changes and execute them on demand. It then lists several commands related to interface configuration, including 'interface Vlan1', 'ip address 10.0.0.1 255.255.255.0', and 'no shutdown'. The script concludes with a 'commit' command.

ServiceNow

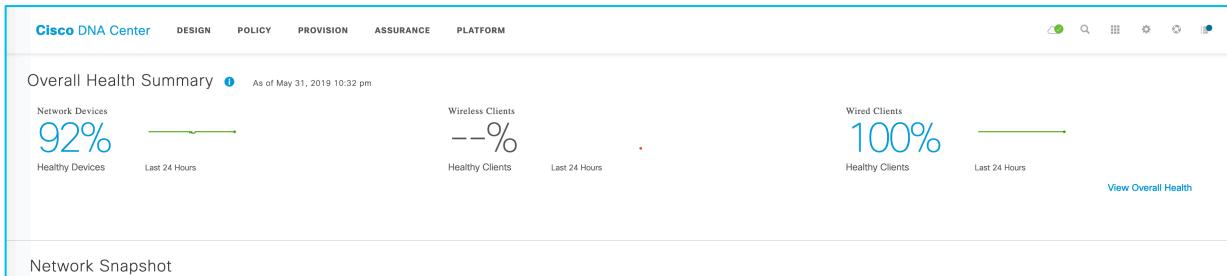
The ServiceNow ticket interface shows a ticket titled 'Ticket for Configuration Change - 2019-07-12 10:30:00'. The ticket details a configuration change for an Arista 7050 device, specifically changing the IP address to 10.0.0.1. The ticket status is 'Open' and it was created on 2019-07-12 10:30:00.

Webex Teams

The Webex Teams interface shows a channel named 'Notifications'. It displays four notifications from the 'ConfigMon' bot, each reporting a configuration change made by user 'ARISTA'. The notifications include the timestamp of the change (e.g., 2019-07-12 10:30:00), the device name ('arista7050'), and the specific configuration command executed.

Open IOS XE, Guest Shell

Cisco DNA Center



Complete network management system

- Single pane of glass for all devices
- End-to-end health information in real time
- Granular visibility
- Simplified workflows

Automation for provisioning

- Zero-touch deployment
- Device lifecycle management
- Policy enforcement

Analytics for assurance

- Verify intent of network settings
- Proactively resolve issues
- Reduce time spent troubleshooting

Platform for extensibility

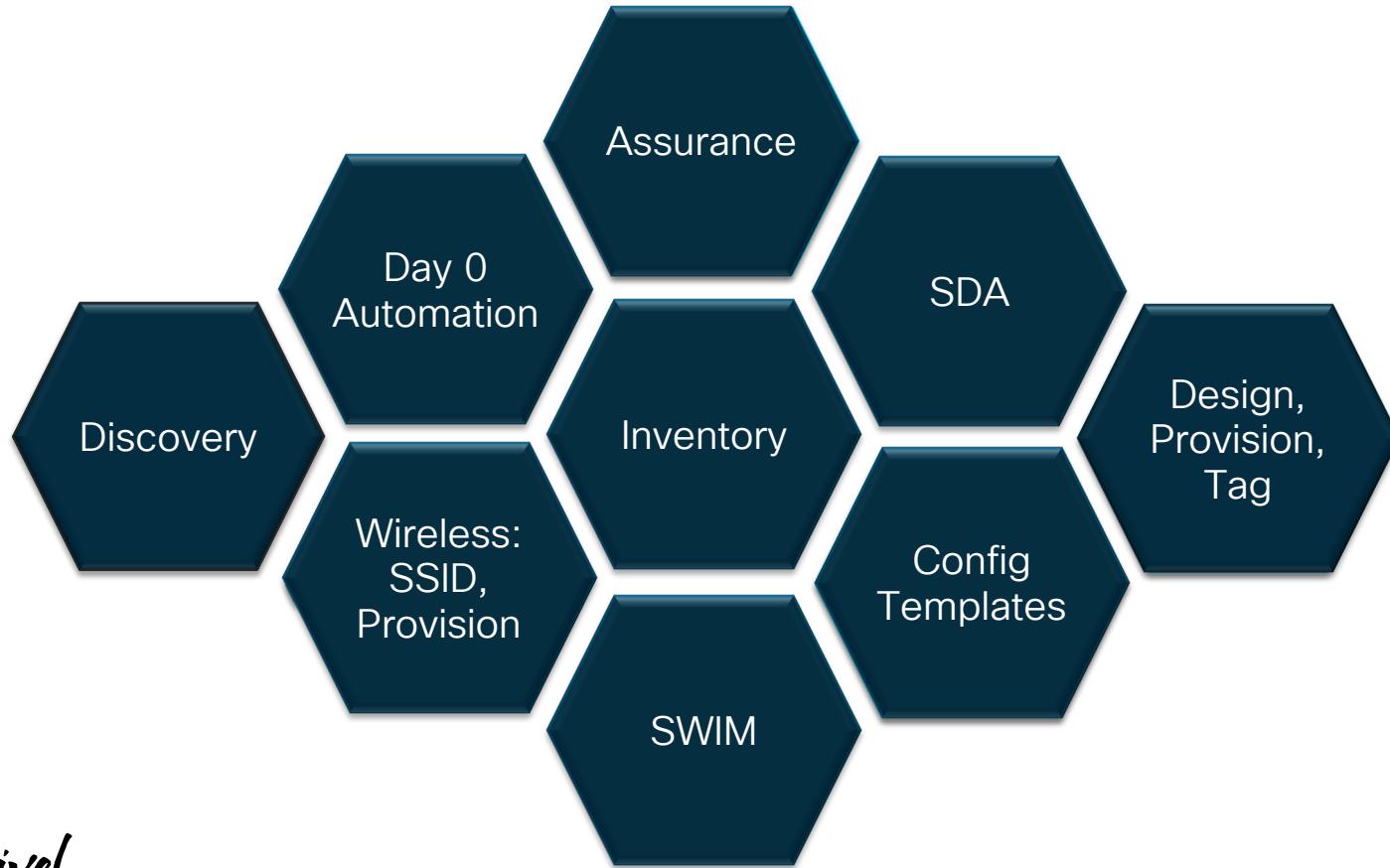
- Integrate APIs with third-party solutions
- Integrate and customize ServiceNow
- Evolve operational tools and processes

DNA Center Platform

The screenshot shows the Cisco DNA Center Platform interface. The top navigation bar includes links for DESIGN, POLICY, PROVISION, ASSURANCE, and PLATFORM. The PLATFORM link is highlighted with a yellow bar. Below the navigation is a header for the "Platform" section, indicating Version 1.0.3 - Released 2/8/2019. A secondary navigation bar includes Overview, Manage (with a dropdown arrow), Developer Toolkit (with a dropdown arrow), and Runtime Dashboard. The "Developer Toolkit" dropdown is open, showing options: APIs (which is selected and highlighted in green), Integration Flows, Data and Reports, and Multivendor support. The main content area is titled "APIs" and contains sections for "Know Your Network", "Site Management", "Connectivity", "Operational Tools", and "Authentication". The "Know Your Network" section is expanded, showing a description: "Know your Network APIs can be used to discover details about clients, sites, topology and devices. It also provides programmatic REST APIs to..." followed by a table for "Sites". The table has columns for Method (with a dropdown arrow), Name, and Description. It lists three API entries:

Method ▲	Name	Description
GET	Get Site Health	Returns Overall Health information for all sites
POST	Assign Device To Site	Assigns list of devices to a site
POST	Create Site Intent	Creates site with area/building/floor with specified hierarchy.

APIs at a Glance



API Docs

Get all keywords of CLIs accepted by command runner

X

DESCRIPTION

Get valid keywords

Method	URL
GET	https://10.93.130.30/dna/intent/api/v1/network-device-poller/cli/legit-reads

PARAMETERS

No Parameters available

RESPONSES

Response Codes

Code	Message
200	The request was successful. The result is contained in the response body.
204	The request was successful, however no content was returned.
206	The GET request included a Range Header, and the server responded with the partial content matching the range.
400	The client made a request that the server could not understand (for example, the request syntax is incorrect).
401	The client's authentication credentials included with the request are missing or invalid.
403	The server recognizes the authentication credentials, but the client is not authorized to perform this request.
404	The client made a request for a resource that does not exist.
409	The target resource is in a conflicted state (for example, an edit conflict where a resource is being edited by multiple users). Retrying the request later might succeed.
415	The client sent a request body in a format that the server does not support (for example, XML to a server that only accepts JSON).
500	The server could not fulfill the request.
501	The server has not implemented the functionality required to fulfill the request.
503	The service is temporarily unavailable.

Code Preview

Try It

- Info of what you need to send and what you expect to receive
- Try APIs without writing code
- Get help to write the API Requests in few programming languages

Cisco DNA Center Command Runner APIs

Command Runner

Method ▾	Name	Description
GET	Get all keywords of CLIs accepted by command runner	Get valid keywords
POST	Run read-only commands on devices to get their real-time configuration	Submit request for read-only CLIs

- Get all the commands supported
- Submit the command to device(s) using the device Id(s)
- Retrieve the output from the file specified in the response

Cisco DNA Center Device Detail API

Try Now

X

Method	Public URL :	
GET	https://10.93.130.30/dna/intent/api/v1/device-detail	
Name Query Params	Description	Value
timestamp*	Epoch time(in milliseconds) when the device data is required	1559884379000
searchBy*	MAC Address or Device Name value or UUID of the network device	:28f81-80eb-4c72-aada-fa082f8a5de9
identifier*	One of keywords : macAddress or uuid or nwDeviceName	uuid

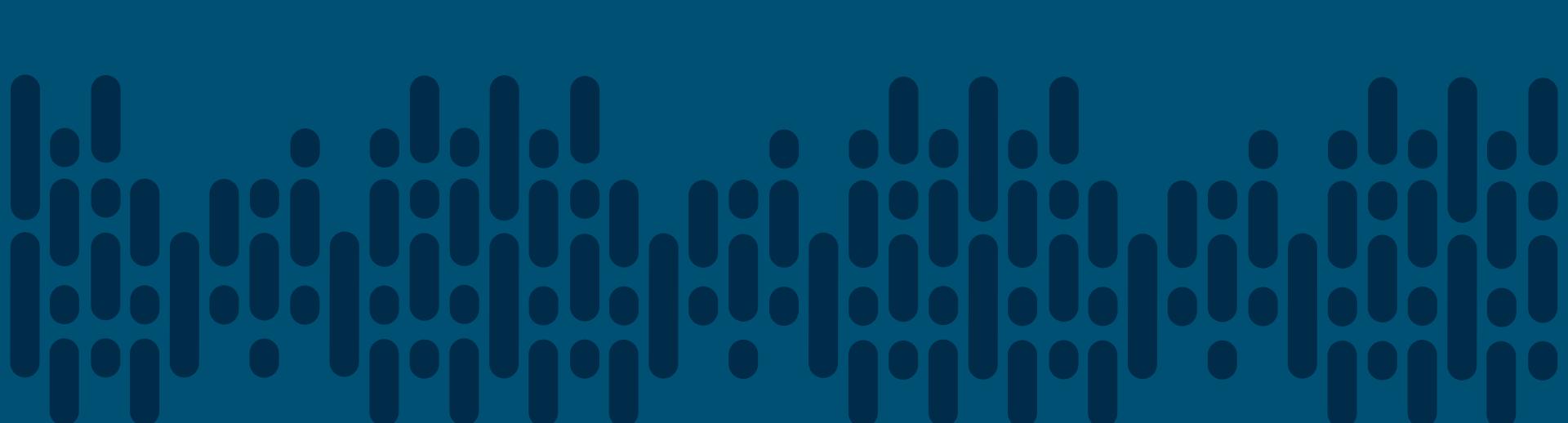
Response

```
1 {  
2   "response": {  
3     "managementIpAddress": "10.93.130.42",  
4     "haStatus": "Non-redundant",  
5     "serialNumber": "9XIF5ZTLRB",  
6     "communicationState": "REACHABLE",  
7     "nwDeviceName": "PDX-RN",  
8     "platformId": "CSR1000V",  
9     "nwDeviceId": "32c28f81-80eb-4c72-aada-fa082f8a5de9",  
10    "nwDeviceRole": "BORDER ROUTER",  
11    "nwDeviceFamily": "Routers",  
12    "macAddress": "00:1E:E5:D7:EF:00",  
13    "healthScore": 85  
14  }  
15}
```

Cancel

Run

- The REST API Request Requires:
 - Device Identifier
 - Type of identifier used
 - Timestamp – Epoch time in msec
- The REST API Response includes:
 - Location, family, operating system version, serial number, management IP address, health score, ...



Demo: Cisco DNA Center in Action

Hands-On

- Find a folder with the name “**DEVWKS_2840_US19**” on the desktop
- Use PyCharm and open the project **DEVWKS_2840_US19**
- **Edit the file “config.py”**
- Enter the environment variables provided to you
 - Cisco DNA Center password
 - ServiceNow Instance number and password
 - Update the IOS XE IP Address and Hostname with the device number
 - IOS XE password
- **Run the Python script ”intro_to_dna_center.py” in PyCharm**
- **Find the information for your device**

ConfigMon App

DNA Center

The screenshot shows the Cisco DNA Center Platform interface. The top navigation bar includes 'HOME', 'DEVICE', 'POLICY', 'PROVISION', 'ASSURANCE', and 'PLATFORM'. The 'PLATFORM' tab is selected, showing 'Cisco DNA Center Managed Device' status. Below the navigation, there are two main sections: 'Command Runner' and 'Network Discovery'. The 'Command Runner' section has a 'Hosted' dropdown set to 'DNA Center' and a 'Name' input field. The 'Network Discovery' section lists 'Discover new interfaces on port that have been configured' and 'Discover devices on user's Ixia'.

This script will merge device configuration changes. It will be executed on demand or in this tab.
It will collect the configuration file for each DNA Center managed device, compare with the existing saved file, and then merge the differences. It will then update the configuration file for each user that configured the device, and create a new backup file.
This script must be run by the user that configured the device, and must be run after the configuration has been committed.
This script must be run by the user that configured the device, and must be run after the configuration has been committed.
or execute current tasks changes
or execute current tasks changes
or execute current tasks changes

ServiceNow

The screenshot shows a ServiceNow ticket interface. The ticket details are as follows:

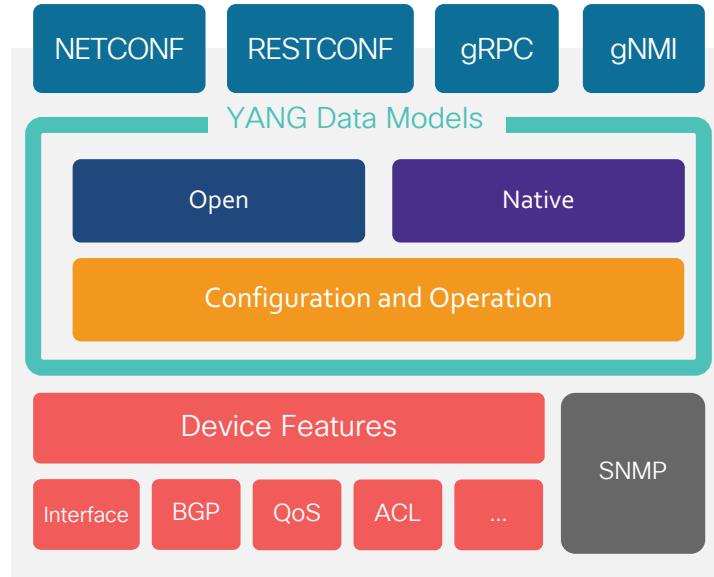
- System:** Configuration Change Item - MTC-0001
- Subject:** Configuration Change Item - MTC-0001 has been updated by user MTC001
- Description:** Configuration Change Item - MTC-0001 has been updated by user MTC001
- Category:** Configuration Change Item
- Type:** Configuration Change Item
- Priority:** Normal
- Owner:** MTC001
- Assignee:** MTC001
- Due Date:** 2019-09-12 12:00:00
- Last Update:** 2019-09-12 12:00:00
- Created:** 2019-09-12 12:00:00
- Updated:** 2019-09-12 12:00:00



Open IOS XE, Guest Shell

NETCONF and RESTCONF

- Options to program network devices:
 - **NETCONF** - Network Configuration Protocol
 - **RESTCONF** - REST-like access to the YANG Data Model
 - **gRPC** - open-source universal RPC framework, started by Google
 - **gNMI** - **gRPC Network Management Interface** – configuration and operational data, telemetry
 - gRPC carries gNMI

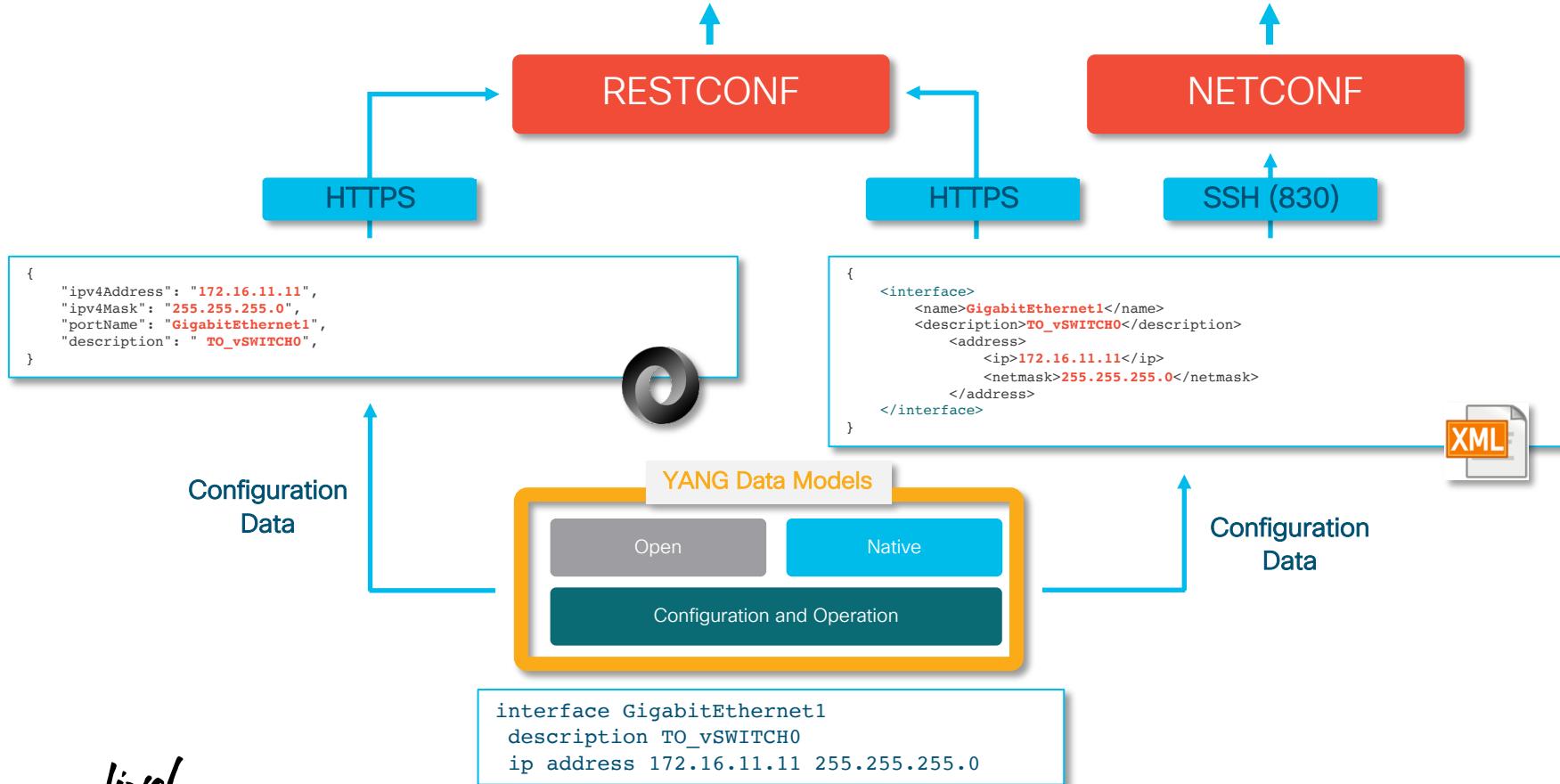


<http://cs.co/IOS-XE-Programmability-Book>

Protocol Summary

	NETCONF	RESTCONF	gRPC, gNMI
Standardization	RFC 6241	RFC 8040	Open Initiative
Encoding	XML	XML / JSON	JSON / GPB
Session-layer Service (RPC) Support	✓	✓	✓
Connection Oriented	✓	✗	✓
Session Security	SSH / SOAP	HTTPS	HTTPS

Network Device APIs



NETCONF and/or RESTCONF ?

```
<rpc-reply message-id="urn:uuid:50bf9d6e-7e5c-4182-ae6b-  
972a055ceef7" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"  
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">  
  <data>  
    <interfaces-state xmlns="urn:ietf:params:xml:ns:yang:ietf-  
    interfaces">  
      <interface>  
        <name>GigabitEthernet1</name>  
        <admin-status>up</admin-status>  
        <oper-status>up</oper-status>  
        <phys-address>00:0c:29:6c:81:06</phys-address>  
        <speed>1024000000</speed>  
        <statistics>  
          <in-octets>5432293472</in-octets>  
          <in-unicast-pkts>28518075</in-unicast-pkts>  
          .....  
          <out-octets>2901845514</out-octets>  
          <out-unicast-pkts>18850398</out-unicast-pkts>  
        </statistics>  
      </interface>  
    </interfaces-state>  
  </data></rpc-reply>
```



```
{  
  "ietf-interfaces:interface": {  
    "name": "GigabitEthernet1",  
    "admin-status": "up",  
    "oper-status": "up",  
    "last-change": "2018-01-17T21:49:17.000387+00:00",  
    "phys-address": "00:0c:29:6c:81:06",  
    "speed": 1024000000,  
    "statistics": {  
      "in-octets": 5425386232,  
      "in-unicast-pkts": 28489134,  
      .....  
      "out-octets": 2899535736,  
      "out-unicast-pkts": 18844784  
    }  
  }  
}
```



YANG Data Models



Demo: RESTCONF and NETCONF in Action

Hands-On

- Run the Python script “intro_to_NETCONF_RESTCONF.py” using PyCharm
- This script will:
 - Retrieve the device capabilities using RESTCONF
 - Find the device hostname using RESTCONF
 - Retrieve interface operational state using NETCONF
 - Retrieve the interface statistics using RESTCONF
 - Save running configuration to startup configuration using RESTCONF
 - Save running configuration to file using NETCONF
 - Rollback configuration from saved file using RESTCONF
- **Question – What are the “device capabilities”? How can we get them?**

ConfigMon App

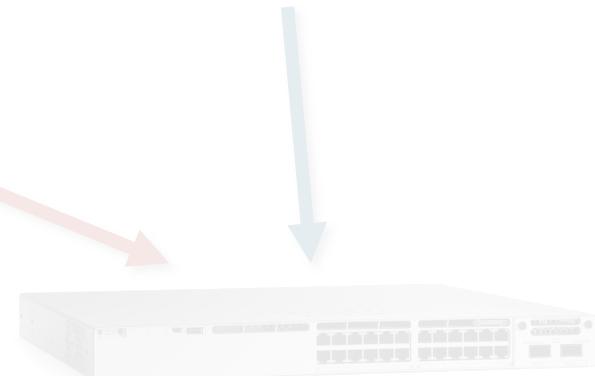
DNA Center

The screenshot shows the Cisco DNA Center Platform interface. In the top navigation bar, 'Platform' is selected. Below it, 'Devices' and 'Manage' are dropdown menus. Under 'Manage', 'Device Toolkit' is selected, which is further expanded to show 'Routine Dashboard'. The main content area displays a 'Command Runner' section with tabs for 'Normal', 'Power', 'Intelligent', and 'Script'. A note says 'Run command or script against multiple devices simultaneously'. Below this, there's a 'Network Discovery' section with a note 'Discover devices on your network using IP range or MAC address'.

This screenshot shows a configuration change log. It starts with a note: 'This script will merge device configuration changes. It will be executed on demand or in this tab.' It then lists several configuration items being updated, such as 'IOSXE Application Policy Range (N2K40)', 'Power N2', 'Access Logging Policy', 'Power N2', 'Power N2', and 'Power N2'. The log also notes 'Updated using API by user: OSCE' and 'Created using API by user: OSCE'.

ServiceNow

This screenshot shows a ServiceNow incident view for incident ID INC2010104. It displays a configuration change log. The log includes entries for 'IOSXE Application Policy Range (N2K40)' with details like 'Power N2', 'Access Logging Policy', 'Power N2', 'Power N2', and 'Power N2'. It also shows 'The device management IP address is 10.81.0.21' and 'The configuration change at 22:17:51 PDT Tue Oct 23 2018 by osce'. The log is created and updated by 'OSCE' using the API.



Open IOS XE, Guest Shell

Webex Teams

This screenshot shows a Webex Teams channel with several notifications. Each notification is for an 'iosxe' instance with a specific IP address (e.g., 10.10.10.10) and describes a 'Configuration Change Alert - IOSXE' that has been updated by user 'OSCE'. The notifications include details like 'Power N2', 'Access Logging Policy', and 'Power N2'. There are three notifications, each with a 'View Details' link.

ServiceNow

- ConfigMon will use REST APIs to create a new incident:
 - Device Information: location, type of device, software version, S/N, health score, management address
 - What is changed
 - Who made the change
 - Configuration compliance state
 - Approval request process
 - Save new config or rollback status

The screenshot shows a ServiceNow incident view with the title "Incident INC0011019". The interface displays five history items:

- Configuration changes not approved, Configuration rolled back successfully**
Updated using APIs by caller: APIUSER
- System Administrator NO**
- Waiting for Management Approval**
Updated using APIs by caller: APIUSER
- Approve these changes (YES/NO)?**
Passed ACL Policy
Passed Logging Policy
Passed Duplicate IPv4 Prevention
Updated using APIs by caller: APIUSER
- The configuration changes are**
 - +interface Loopback200
 - + no ip address
! Last configuration change at 22:59:15 UTC Tue Jun 4 2019 by cisco
Updated using APIs by caller: APIUSER
- Device location: Global/San Diego
Device family: CSR1000V
Device OS info: IOS-XE, 16.9.3
Device S/N: 9DV9GXABDZ
Device Health: 10/10
Device management IP address: 128.107.70.161**
Updated using APIs by caller: APIUSER



Demo: Create ServiceNow Incident

Hands-On

- Run the Python script “create_incident.py” using PyCharm
- This script will:
 - Find the ServiceNow sys_id for the API User account
 - Find out the device hostname using RESTCONF (for a specific IP address)
 - Create a new ServiceNow incident
- Troubleshooting – if not successful, verify the ServiceNow password in the config.py file
- Question – What is the user “sys_id”? Why do we need it?

ConfigMon App

DNA Center

The screenshot shows the Cisco DNA Center Platform interface. At the top, there are tabs for Device, Policy, Provision, Assurance, and Platform. The Platform tab is selected. Below the tabs, there are sections for Device Management (Devices, Manage), APIs (Command Runner, Data Management, Operations Tools, Network Discovery, Services Programs), and Network Discovery.

This screenshot shows the configuration interface for the ConfigMon App. It displays a script or configuration file with several commands related to device configuration and monitoring. Key lines include:

```
#!/bin/bash
# This script will merge device configuration changes. It will be executed on demand or in this job.
# It will gather the configuration file for each DNA Center managed device, compare with the existing config file,
# and then merge the differences. It will then upload the configuration file to each device, and create a new ServiceNow incident
# if there are any changes. It will also log the configuration changes to a log file and notifications of incoming changes
# will be sent to the configured email address.
# or specify device chassis changes
# or specify device IP address
# or specified IP address
```

ServiceNow

The screenshot shows a ServiceNow ticket titled "Incident - Configuration Change Alert - NYC-8500". The ticket details a configuration change made by user "IOSXE" on device "IOS-XE-0001" at 10:20:20 AM. The ticket is currently open and assigned to "IOS-XE".

Webex Teams

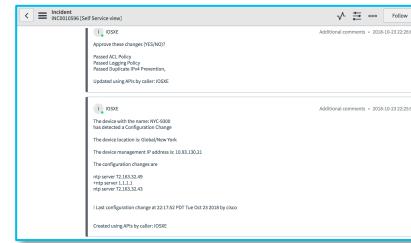
The screenshot shows a Cisco Webex Teams channel named "ServiceNow - Notifications". It displays a series of notifications from ServiceNow regarding configuration changes on device "IOS-XE-0001". Each notification includes the incident ID (e.g., INC00000004), description (e.g., "Configuration Change Alert - NYC-8500"), and the user who triggered it (e.g., "IOSXE").

Open IOS XE, Guest Shell

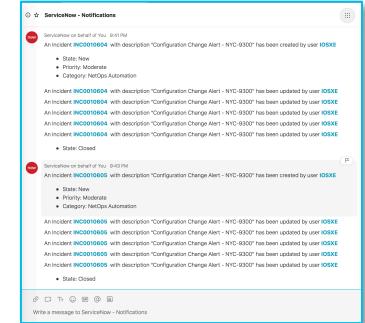
ServiceNow to WebEx Teams Integration

- Notifications sent from ServiceNow to WebEx Teams
- Create a Business Rule when to run the integration
- Sample code provided by WebEx teams
- Customize it for your environment

ServiceNow



New, updated,
or closed
Incident



Webex Teams

ServiceNow to Webex Teams

<https://apphub.webex.com/categories/all/bots/servicenow-3117>

Step B

Enter a Name of your choice, choose Incident from the Table dropdown and select Active and Advanced

```
return res;
},
var ciscoSpark = new Cisco();
var user = gs.getUser();
var gru = GlideScriptRecordUtil.get(current);
```

Note:

- You need a
- Only global
- selected ta

Above code contains the example code block (which is commented) for creating custom notifications. Refer the example to modify the code block as per your requirements. Please ensure to assign your custom notification to the **custom_message** variable, once you have created the custom notification.

Step D

ServiceNow Service Management

business

Script

```
if (changedField == "Insert") {
    if (changedValue == 2) changedValue = "High";
    if (changedValue == 3) changedValue = "Medium";
    if (changedValue == 4) changedValue = "Low";
    custom_message = "An " + changedField + " has been created with number " + current.number + " and description " + current.description;
} else if (current.operation() == "Delete") {
    custom_message = "A (" + table_name + ") " + current.number + " with description '" + current.description + "' has been deleted.";
} else if (current.operation() == "Insert" || previous.number == "") {
    custom_message = "A (" + table_name + ") " + current.number + " has been created by user " + user.name + " with description " + current.description;
} else {
    custom_message = "A (" + table_name + ") " + current.number + " with description '" + current.description + "' has been updated.";
}

//Following line should always be the last line of your custom code for the custom notification
obj.custom_message = custom_message;
ciscoSpark.post("https://runfile.build.io/run/mSP?h=Dtable=Incident&version=Jakarta", obj);
}}(current, previous);
```

Cancel Save

ServiceNow

ServiceNow to Webex Teams - Code

Business Rule
ServiceNow_to_Webex_Teams_bot

A business rule is a server-side script that runs when a record is displayed, inserted, deleted, or when a table is queried. Use business rules to automatically change values in form fields when the specified conditions are met. [More Info](#)

Name: ServiceNow_to_Webex_Teams_bot Application: Global

Table: Incident [incident] Active:

Advanced:

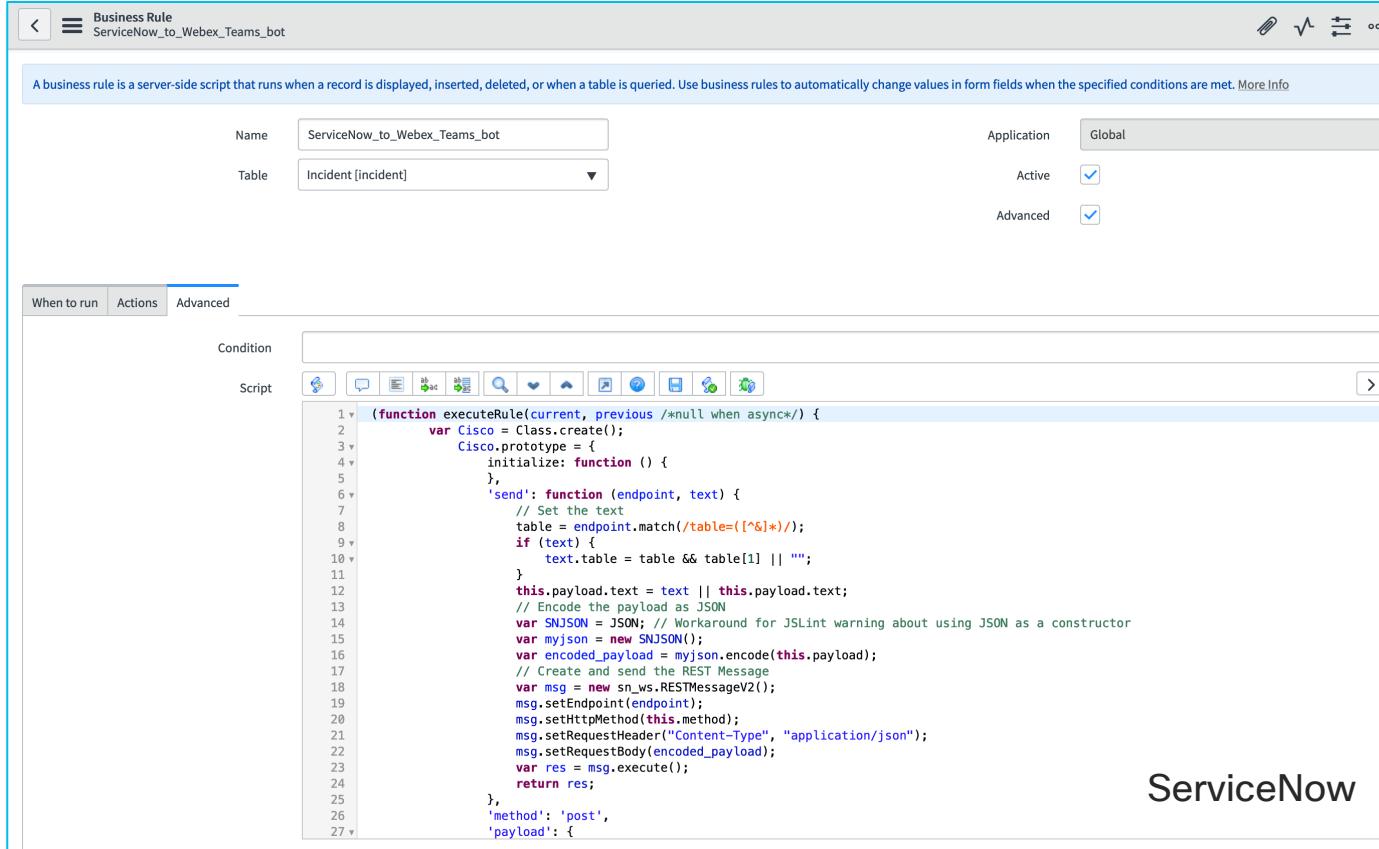
When to run Actions Advanced

Condition:

Script:

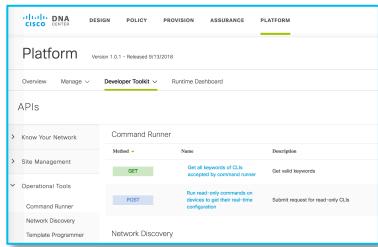
```
1 v  (function executeRule(current, previous /*null when async*/){  
2 v   var Cisco = Class.create();  
3 v     Cisco.prototype = {  
4 v       initialize: function () {  
5 v     },  
6 v       'send': function (endpoint, text) {  
7 v         // Set the text  
8 v         table = endpoint.match(/table=(\^&|*)/);  
9 v         if (text) {  
10 v           text.table = table && table[1] || "";  
11 v         }  
12 v         this.payload.text = text || this.payload.text;  
13 v         // Encode the payload as JSON  
14 v         var SNJSON = JSON; // Workaround for JSLint warning about using JSON as a constructor  
15 v         var myjson = new SNJSON();  
16 v         var encoded_payload = myjson.encode(this.payload);  
17 v         // Create and send the REST Message  
18 v         var msg = new sn_ws.RESTMessageV2();  
19 v         msg.setEndpoint(endpoint);  
20 v         msg.setHttpMethod(this.method);  
21 v         msg.setRequestHeader("Content-Type", "application/json");  
22 v         msg.setRequestBody(encoded_payload);  
23 v         var res = msg.execute();  
24 v         return res;  
25 v     },  
26 v     'method': 'post',  
27 v     'payload': {
```

ServiceNow



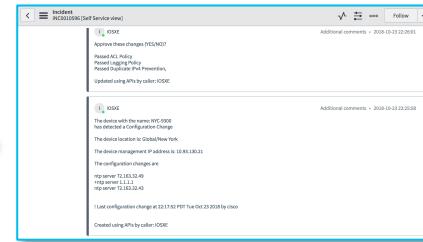
ConfigMon App

DNA Center

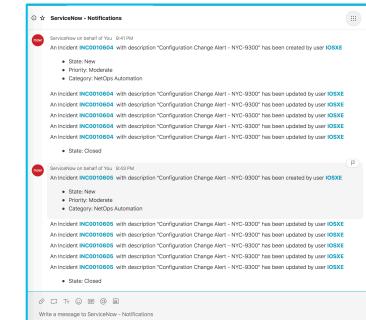


Open IOS XE, Guest Shell

ServiceNow



Webex Teams



Compliance Checks

- Rules:
 - No Access Control configurations changes
 - No logging configurations changes
 - Configurations should not create duplicated IPv4 addresses with clients or network devices interfaces
- Question:
 - How do I write a new rule, for example "No NTP configuration changes?"

Writing a Compliance Rule

- Save the running configuration as a baseline to a file
- Read current running configuration
- Identify delta, using the diff function
- Check for keywords that are matches, in the changed lines

```
interface Loopback2019
- no ip address
+ description CL US San Diego
+ ip address 19.19.19.19 255.255.255.255
```

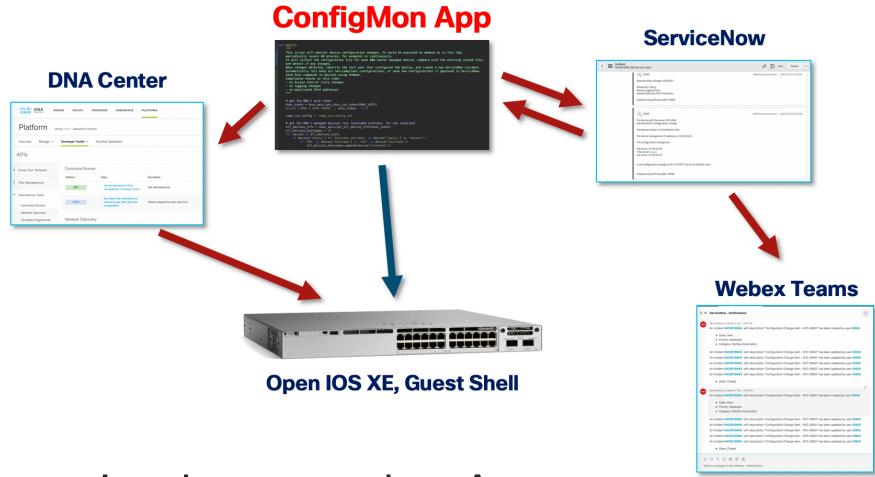
```
ip access-list extended NO_PING
deny icmp any any
permit ip any any
logging trap critical
logging host 10.93.130.30
+access-list 19 deny any
```

```
validation_result = 'Pass'
validation_comment = ''
if ('+access-list' in diff) or ('-access-list' in diff):
    updated_comment = '\nValidation against ACL changes failed'
    service_now_apis.update_incident(incident, updated_comment,
SNOW_DEV)
    validation_result = 'Failed'
else:
    validation_comment = '\nPassed ACL Policy'
```



Demo: ConfigMon

Hands-On



- We have tested all the software required to run the App
- Run the application “config_mon.py”
- SSH to the device using the public IP address and make a configuration change

ConfigMon Code

github.com/zapodeanu/DEVWKS_2840_US19

 [zapodeanu / DEVWKS_2840_US19](#)

[Watch](#) 0 [Star](#) 0 [Fork](#) 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Actions](#) [Projects 0](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

Repo for ConfigMon App - Cisco Live US 2019 session DEVWKS-2840 [Edit](#)

[Manage topics](#)

 12 commits  1 branch  0 releases  1 contributor  GPL-3.0

Branch: [master](#) [New pull request](#) [Create new file](#) [Upload files](#) [Find File](#) [Clone or download](#)

File	Last Commit	Time Ago
 zapodeanu updated for CL San Diego 2019	Latest commit 2251bea now	
 .gitignore	Updated for CL San Diego 2019	16 minutes ago
 LICENSE	Initial commit	2 days ago
 README.md	updated for CL San Diego 2019	3 hours ago
 config.py	updated for CL San Diego 2019	now
 config_mon.py	updated for CL San Diego 2019	27 minutes ago
 create_incident.py	updated for CL San Diego 2019	27 minutes ago
 dnac_apis.py	updated for CL San Diego 2019	3 hours ago
 intro_to_NETCONF_RESTCONF.py	updated for CL San Diego 2019	27 minutes ago

Other Sessions

- DEVNET-3841 "Project WhatsOp - IOS XE Messaging Platform"
 - Integration between Cisco DNA Center, IOS XE Guest Shell, ServiceNow, Webex Teams, PubNub, and Github
- BRKNMS-2935 "From Zero to Network Programmability in 120 Minutes"
 - Elastic Remote Network Access - Vendors remote access solution using Cisco DNA Center, RESTCONF, NETCONF, Webex Teams and ServiceNow

Enterprise Networks BookSprints

<http://cs.co/cat9000book>

<http://cs.co/sdabook>

<http://cs.co/wirelessbook>

<http://cs.co/programmabilitybook>

<http://cs.co/assurancebook>

<http://cs.co/sdwanbook>



Complete your online session evaluation



- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (starting on Thursday) to receive your Cisco Live water bottle.
- All surveys can be taken in the Cisco Live Mobile App or by logging in to the Session Catalog on cisco.cisco.com/us.

Cisco Live sessions will be available for viewing on demand after the event at cisco.cisco.com.

Continue your education



Demos in the
Cisco campus



Walk-in
self-paced labs



Meet the engineer
1:1 meetings



Related sessions



Thank you





A series of stylized lowercase 'i' characters arranged horizontally. The characters are colored in a repeating pattern: blue, green, blue, orange, red, orange, blue, green, blue, green. Each character consists of a short vertical stem with a small circular dot at the top.

You make **possible**