cisco Live!

Let's go

The bridge to possible

Cisco Catalyst Center-as-Code Network Compliance Use Cases

Gabi Zapodeanu
Principal Technical Marketing Engineer, Cisco Systems
@zapodeanu
DEVNET-2634

Cisco Webex App

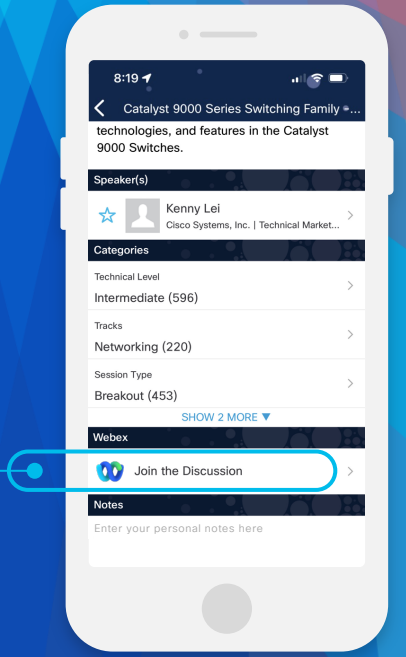
Questions?

Use Cisco Webex App to chat
with the speaker after the session

How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until December 22, 2023.



https://ciscolive.ciscoevents.com/ciscolivebot/#DEVNET-2634

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Agenda

- Platforms
- Catalyst Center Compliance
- Custom Network Compliance
- Developer Resources
- Lessons Learned

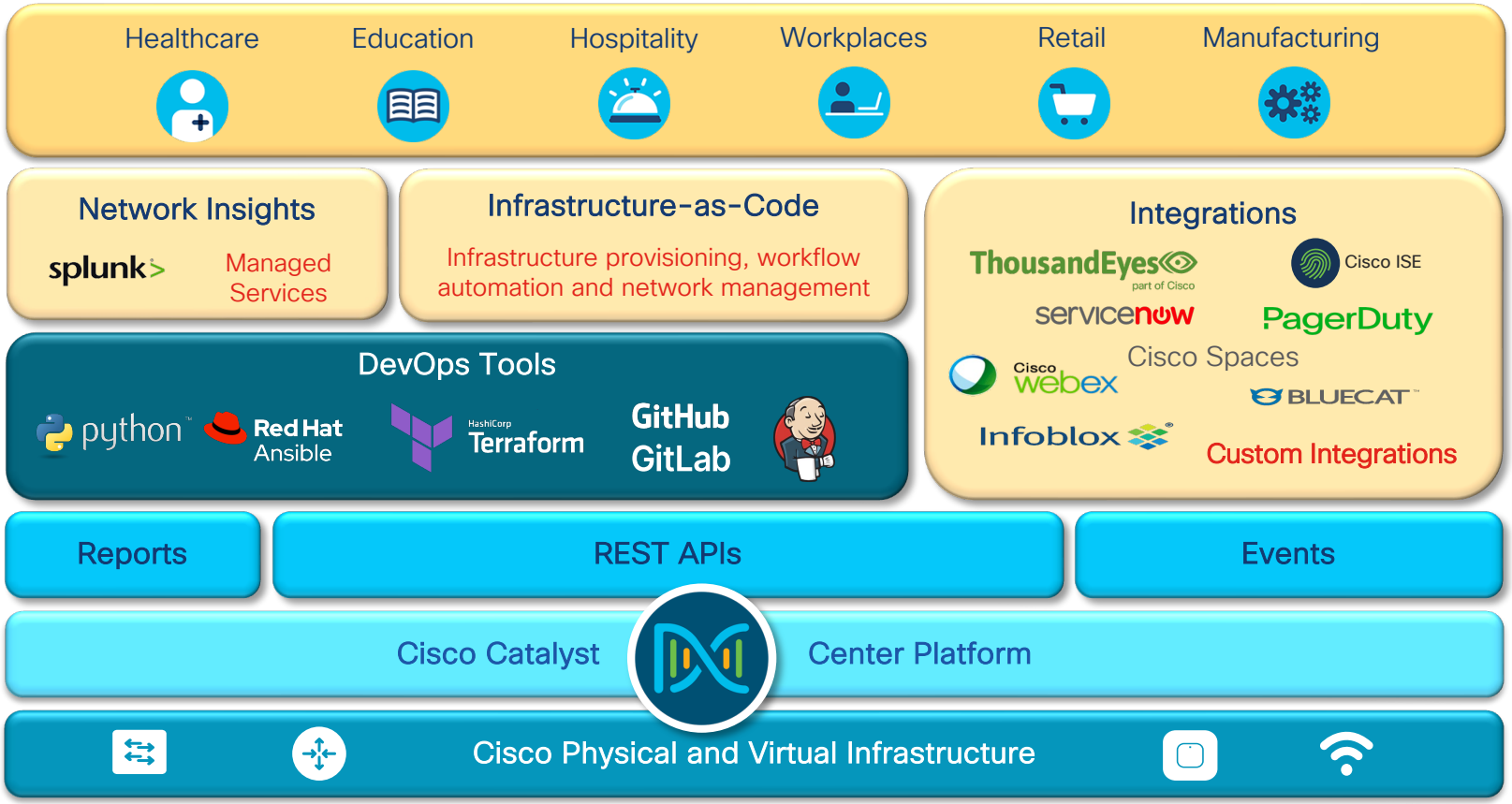
© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Agenda

- Platforms
- Catalyst Center Compliance
- Custom Network Compliance
- Developer Resources
- Lessons Learned

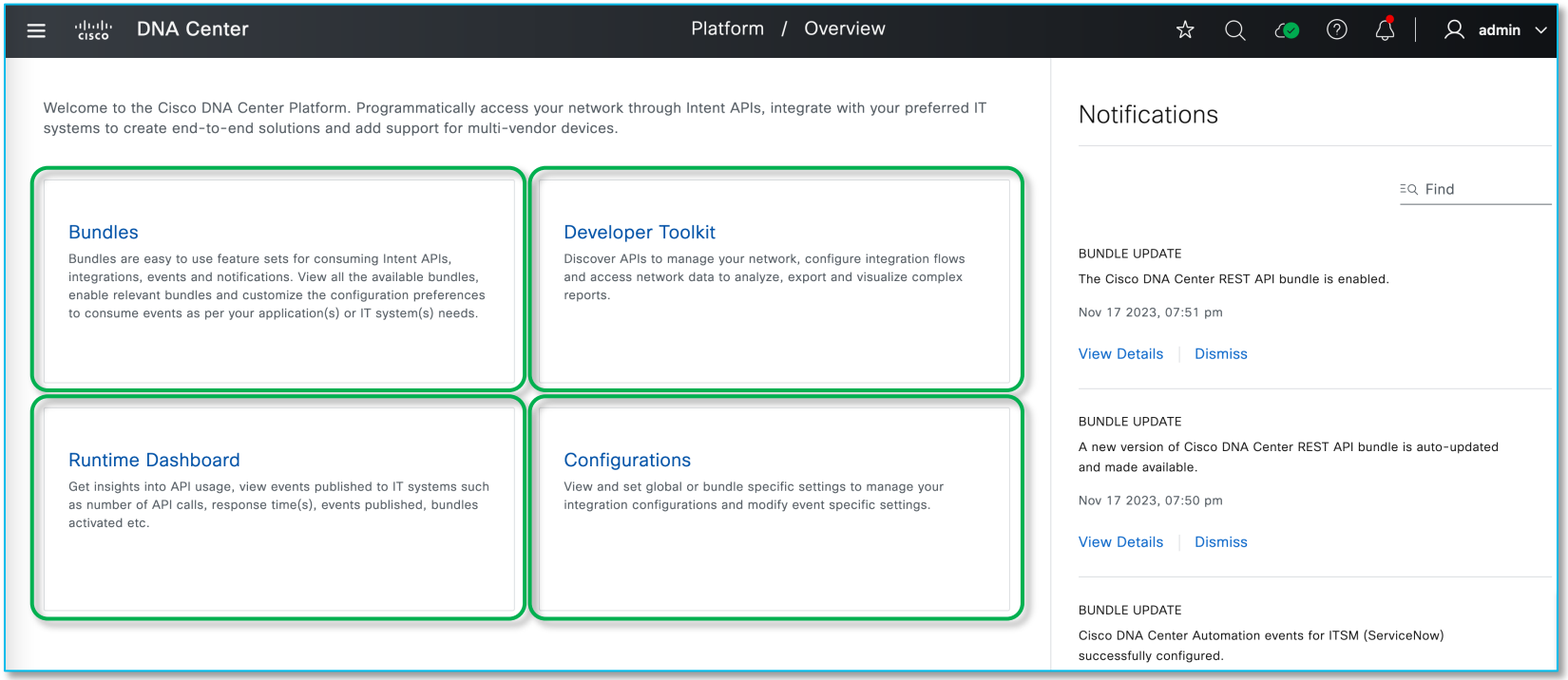
© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Cisco Catalyst Center Platform



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

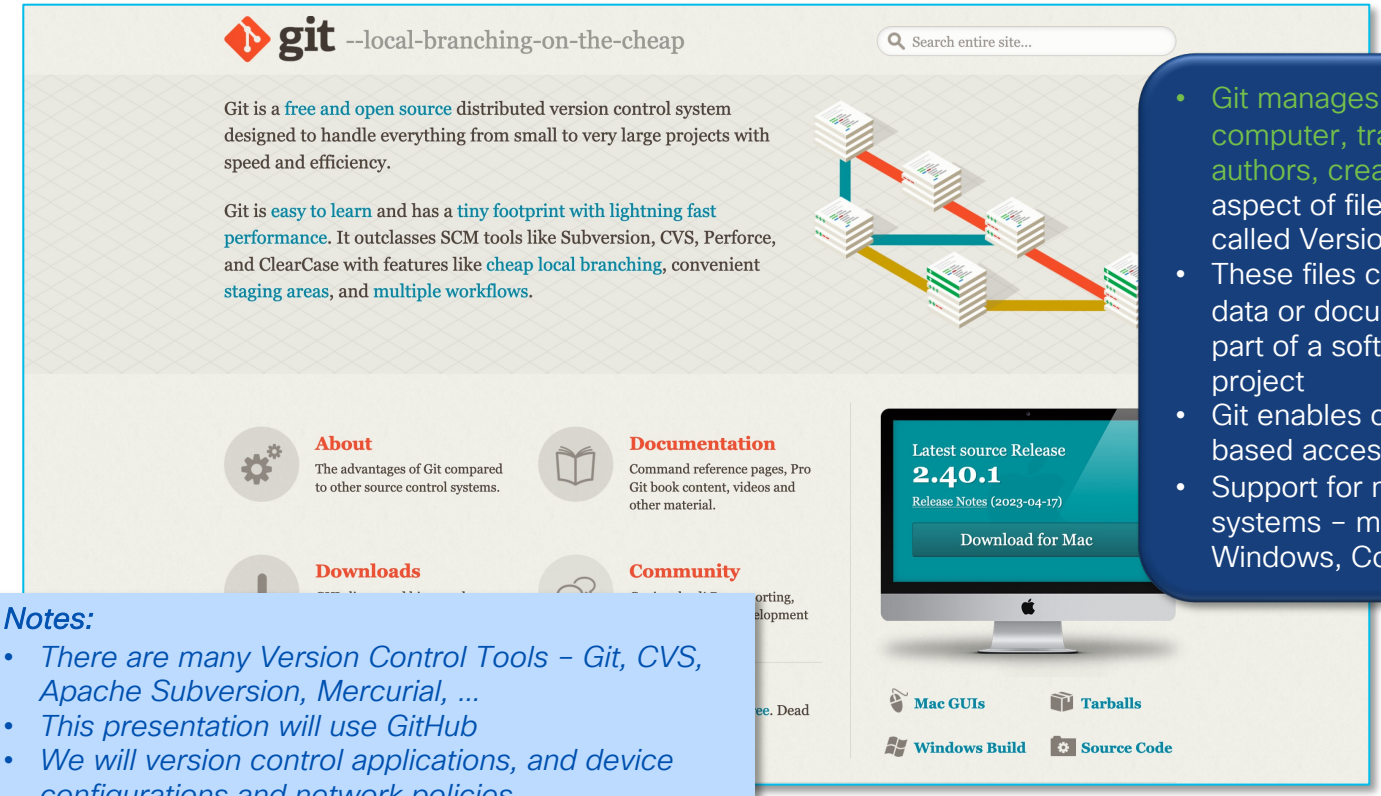
Catalyst Center Platform – Overview



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Version Control

Ref: https://git-scm.com/



- Git manages files on your computer, tracking changes, authors, creating diffs. This aspect of file management is called Version Control
- These files can be applications, data or documentation that is part of a software development project
- Git enables collaboration, role-based access and security
- Support for major operating systems – macOS, Linux, Windows, Containers

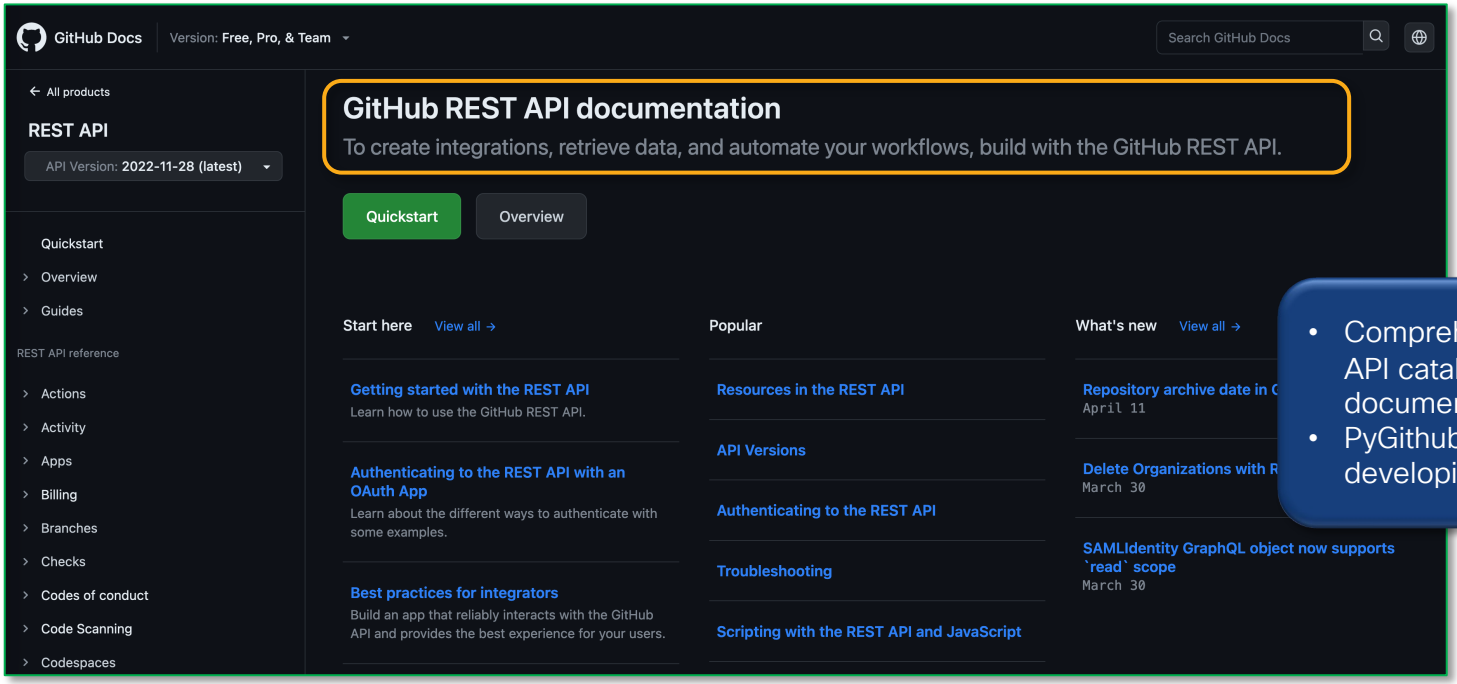
Notes:

- *There are many Version Control Tools – Git, CVS, Apache Subversion, Mercurial, …*
- *This presentation will use GitHub*
- *We will version control applications, and device configurations and network policies*

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

GitHub APIs

https://docs.github.com/en/rest



- Comprehensive REST API catalog, very well documented
- PyGithub library if developing in Python

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Why Cisco Catalyst Center and Version Control?

+

Version Control Platforms

GitHub

Git Version Control Platforms

- Track changes of your automation apps, network state, and Cisco Catalyst Center intent configs
- May be used as single source of truth for your network compliance policies
- Robust role-based access for teams and developers

Will support **Cisco Catalyst Center Infrastructure-as-Code** solutions

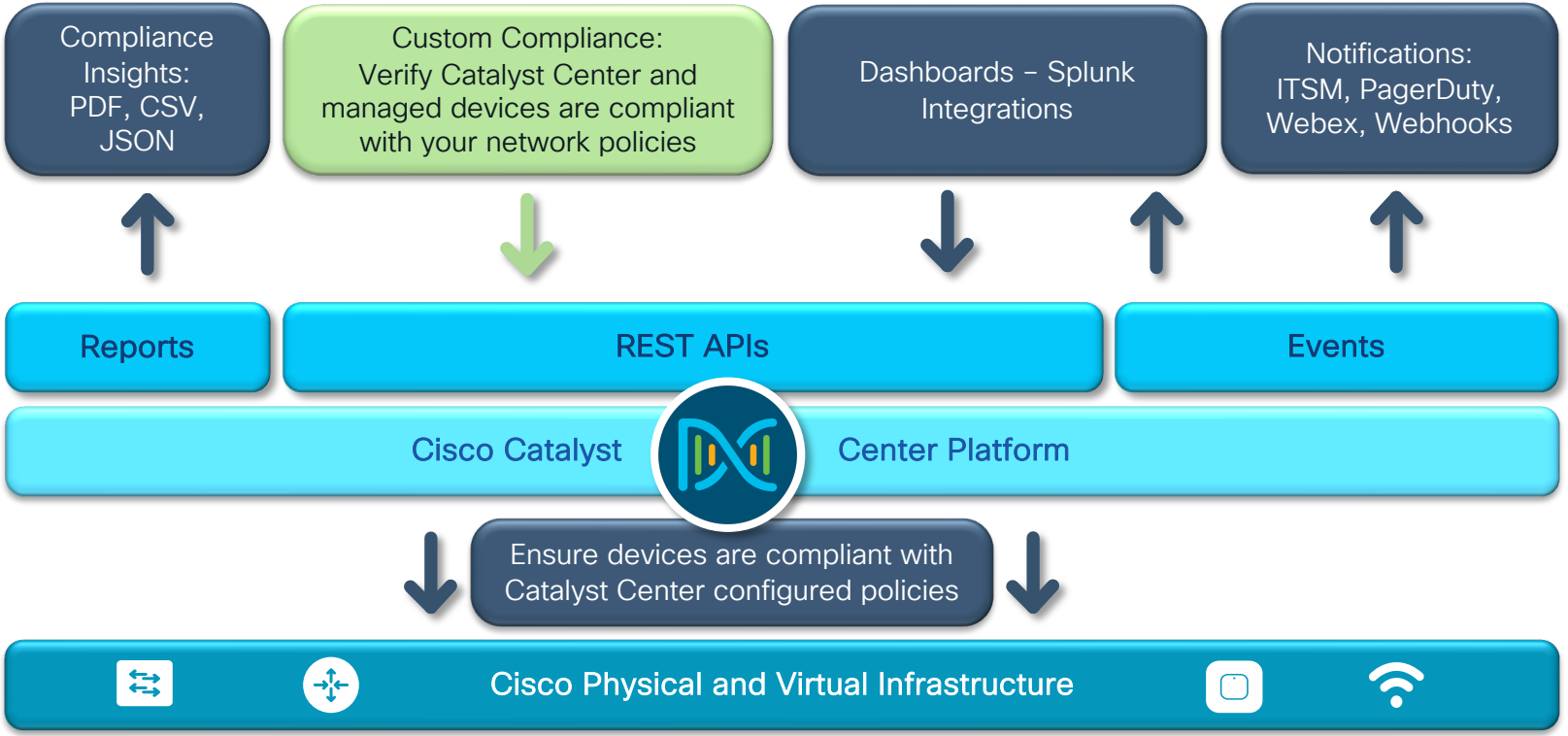
© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Agenda

- Platforms
- Catalyst Center Compliance
- Custom Network Compliance
- Developer Resources
- Lessons Learned

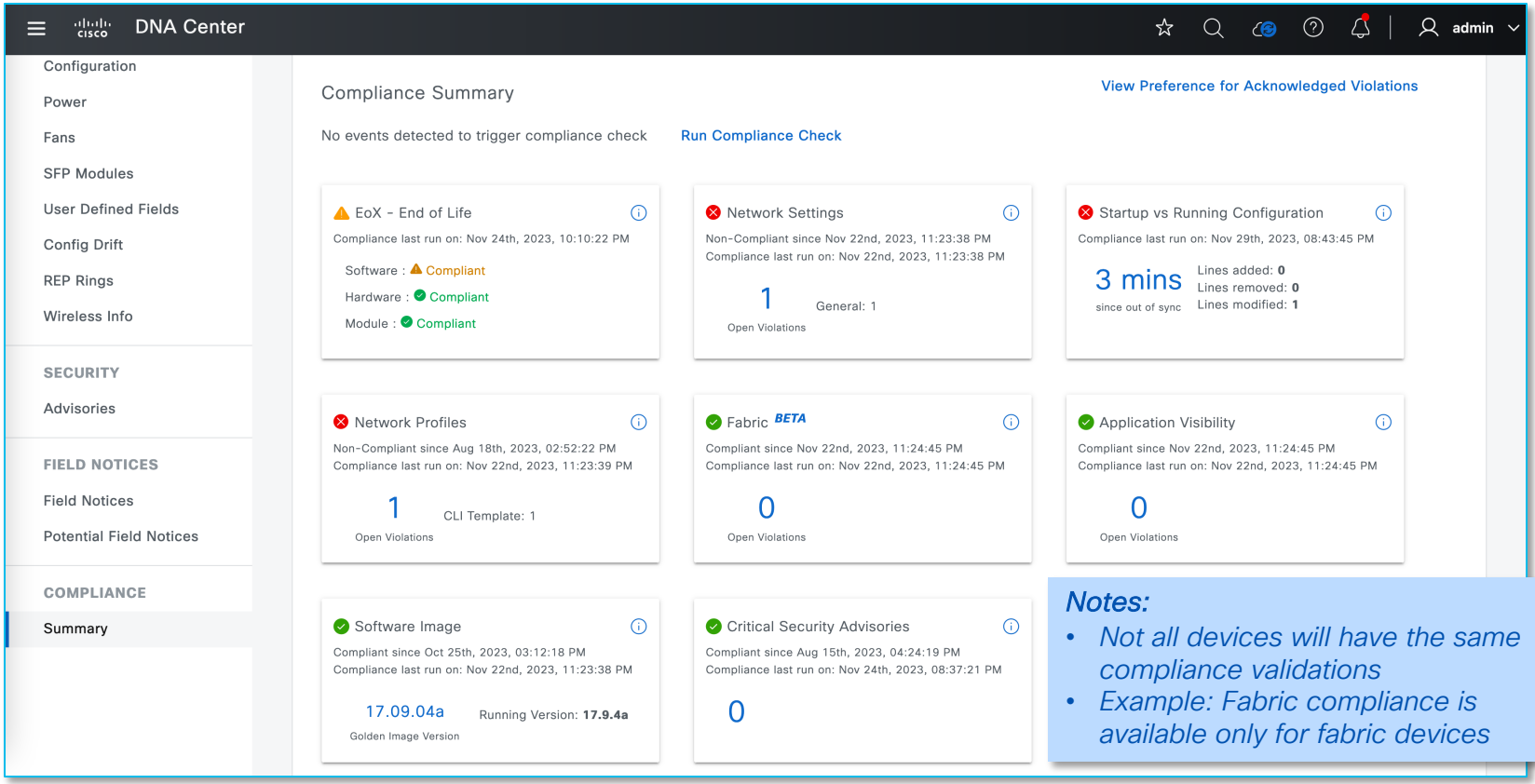
© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Network Compliance + Custom Compliance



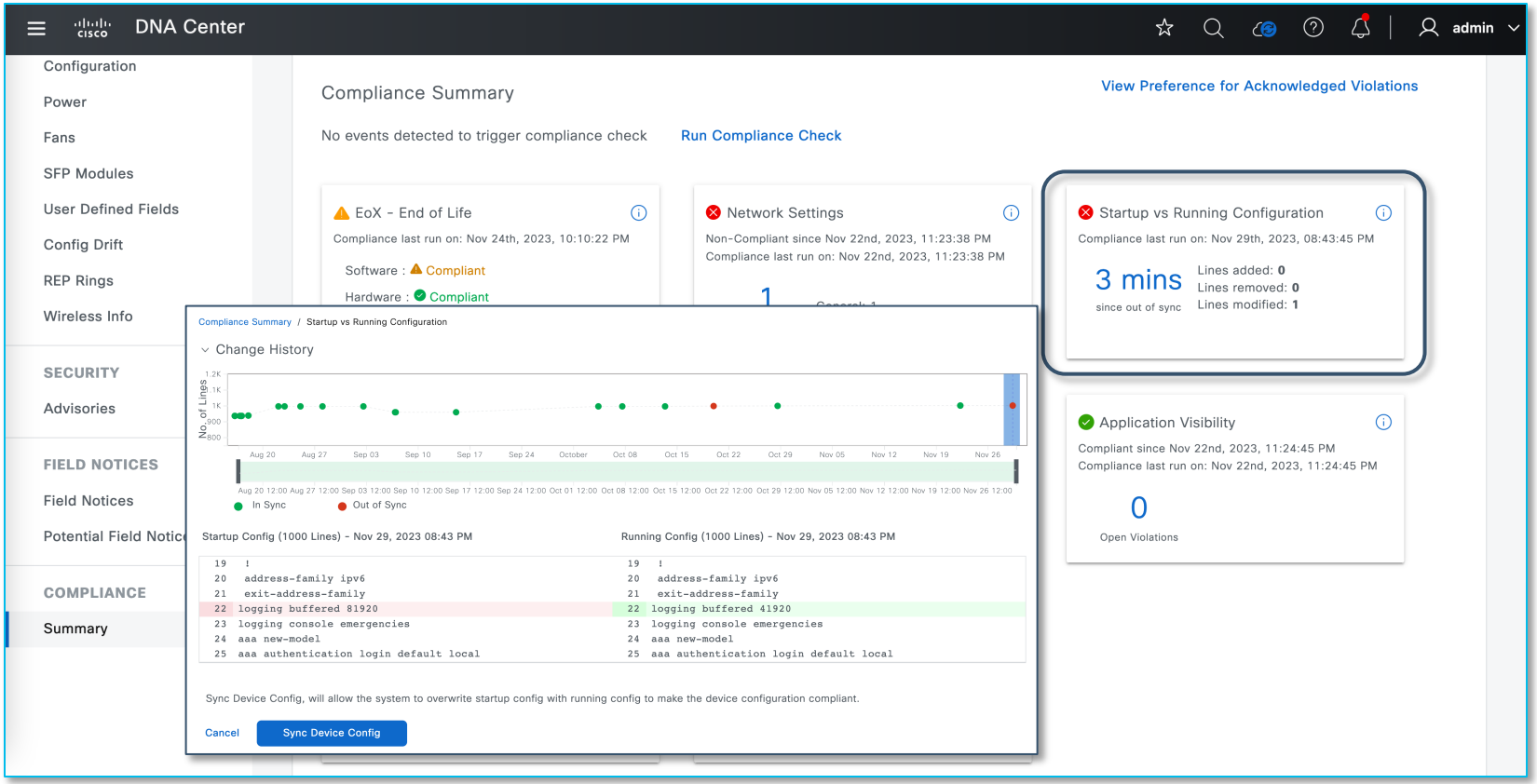
© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Catalyst Center Network Compliance



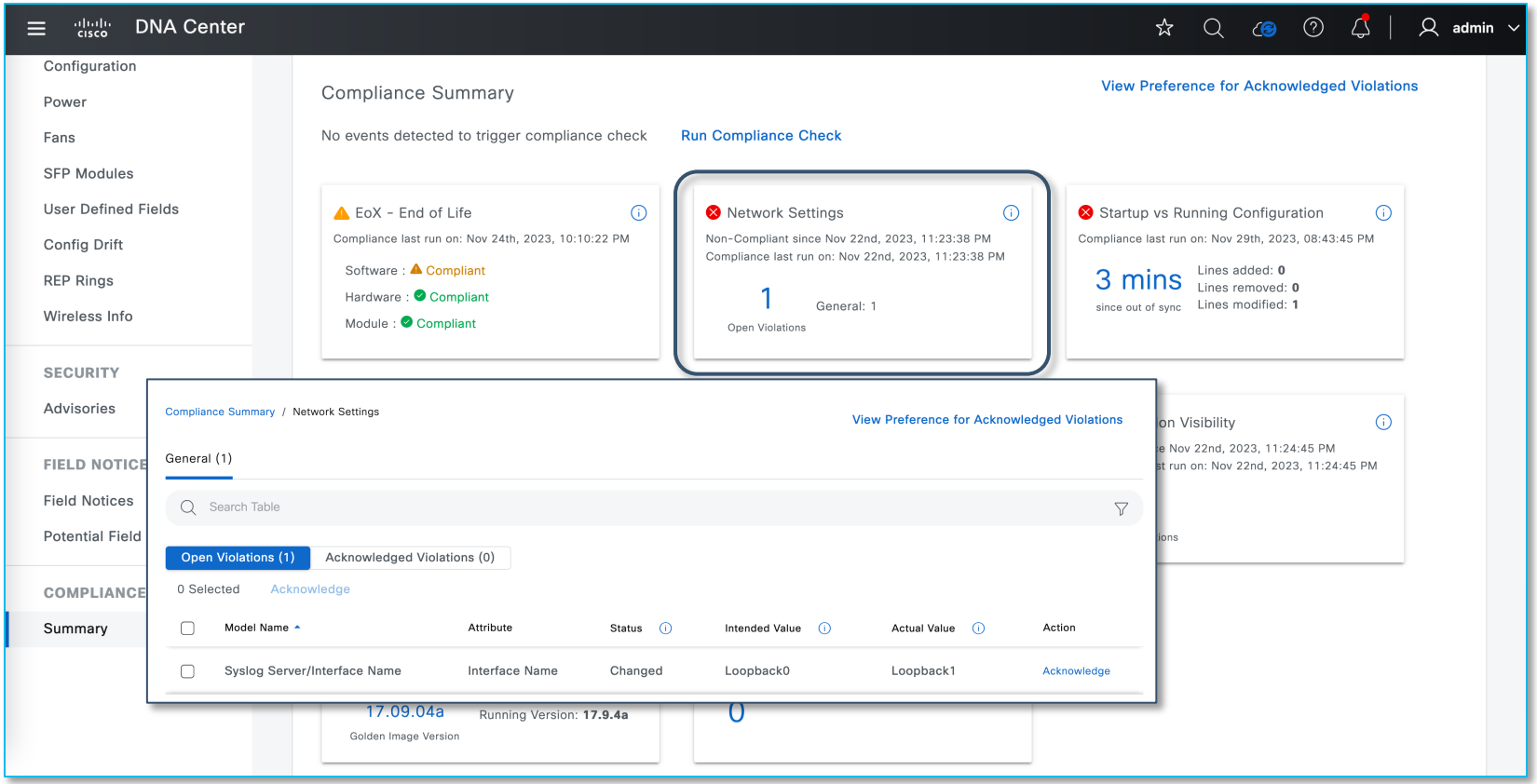
© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Startup vs Running Configuration



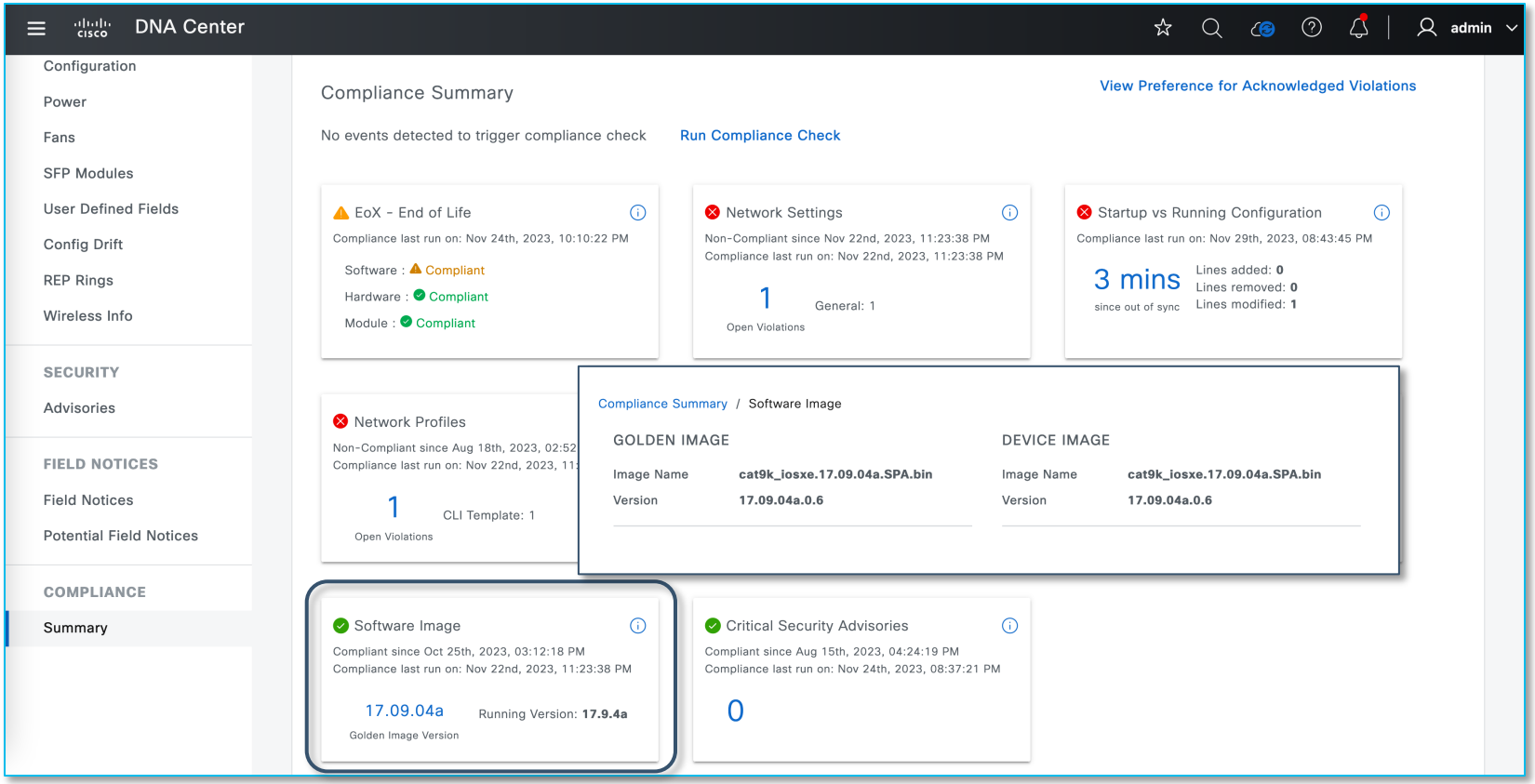
© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Network Settings



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

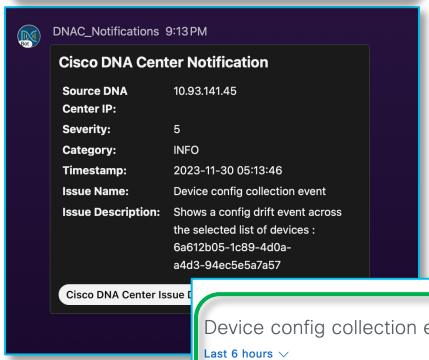
Software Image



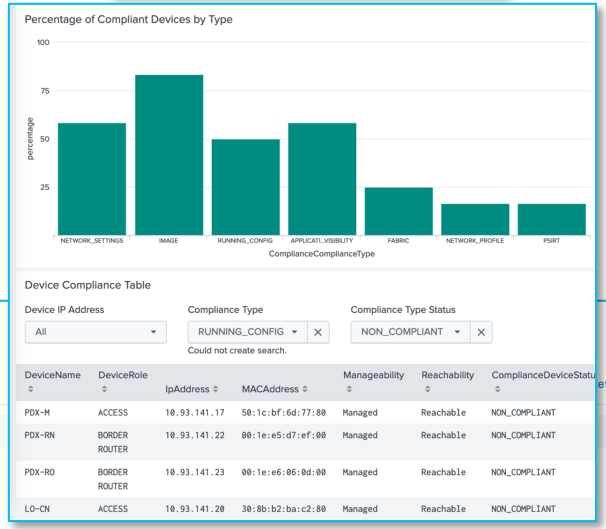
© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Compliance Events, Reports and Dashboards

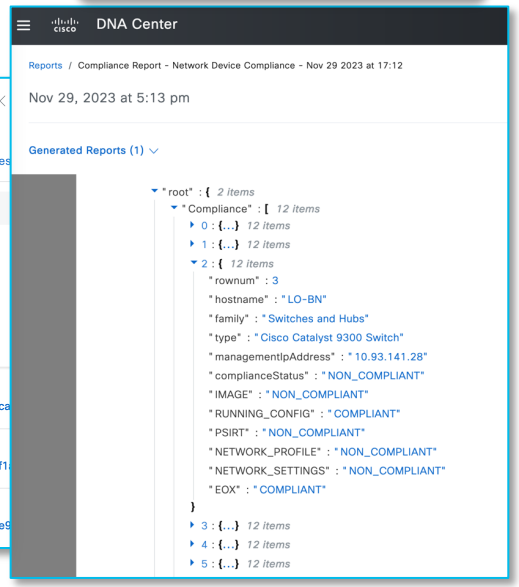
Webex Notifications

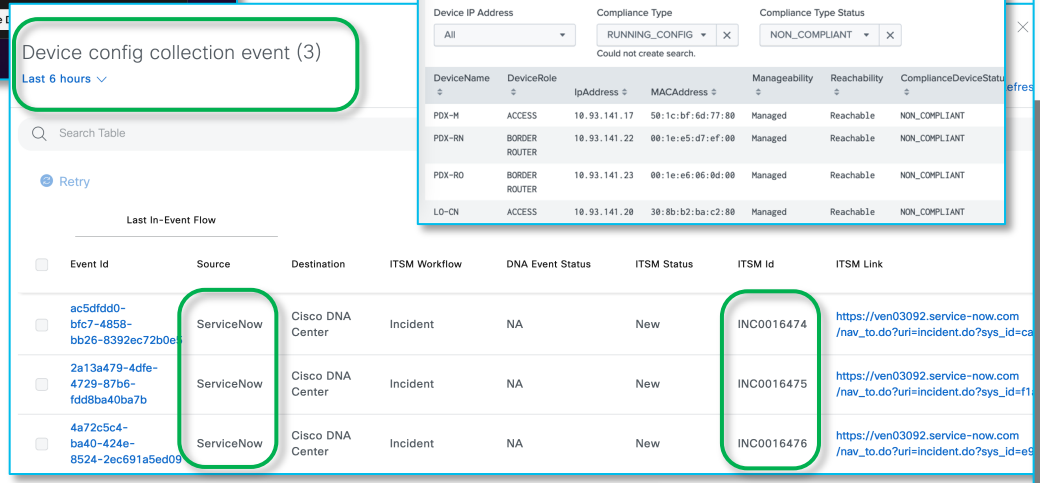


Splunk Dashboard



Report

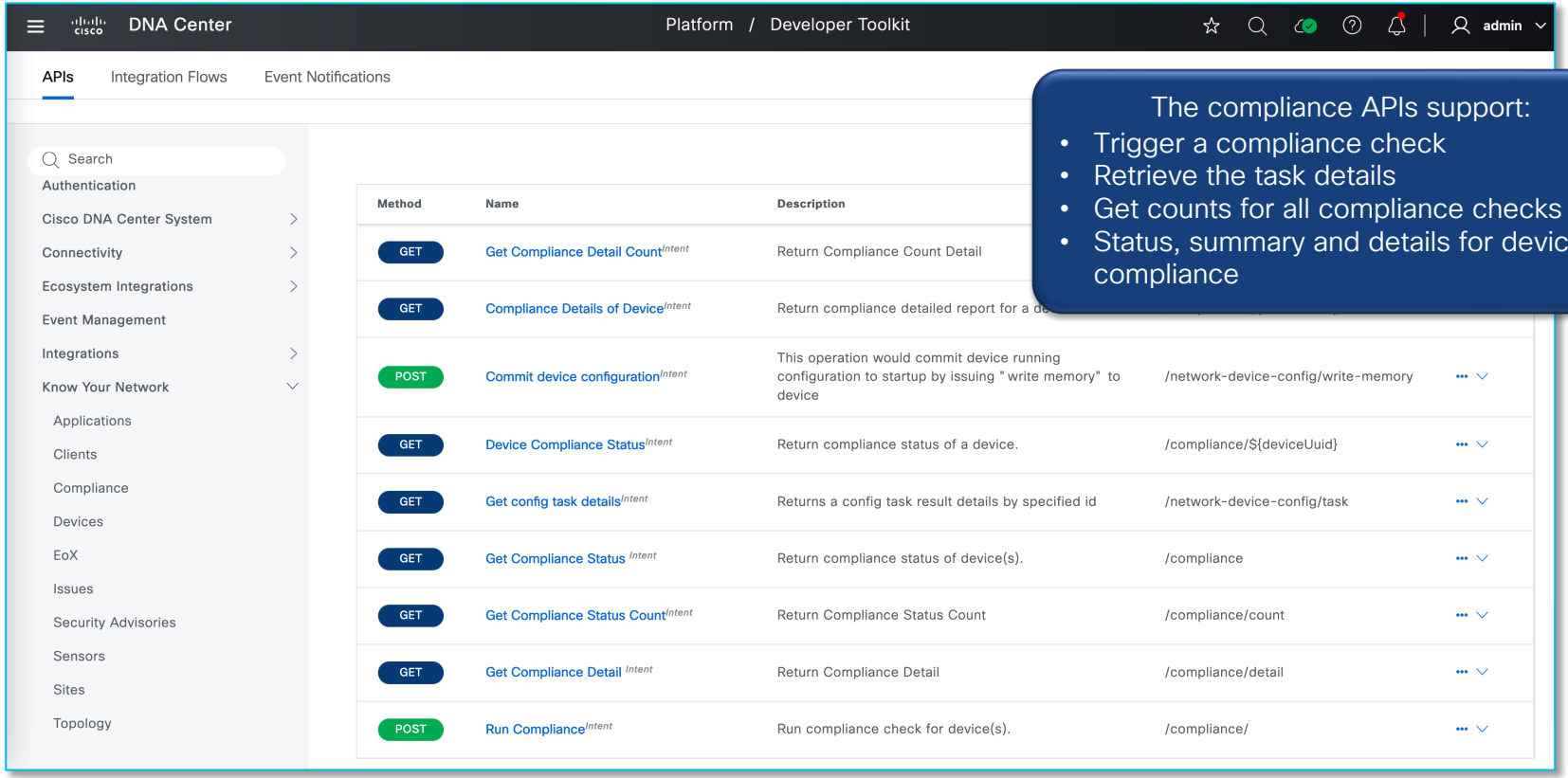




ServiceNow Incidents

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Catalyst Center Compliance APIs



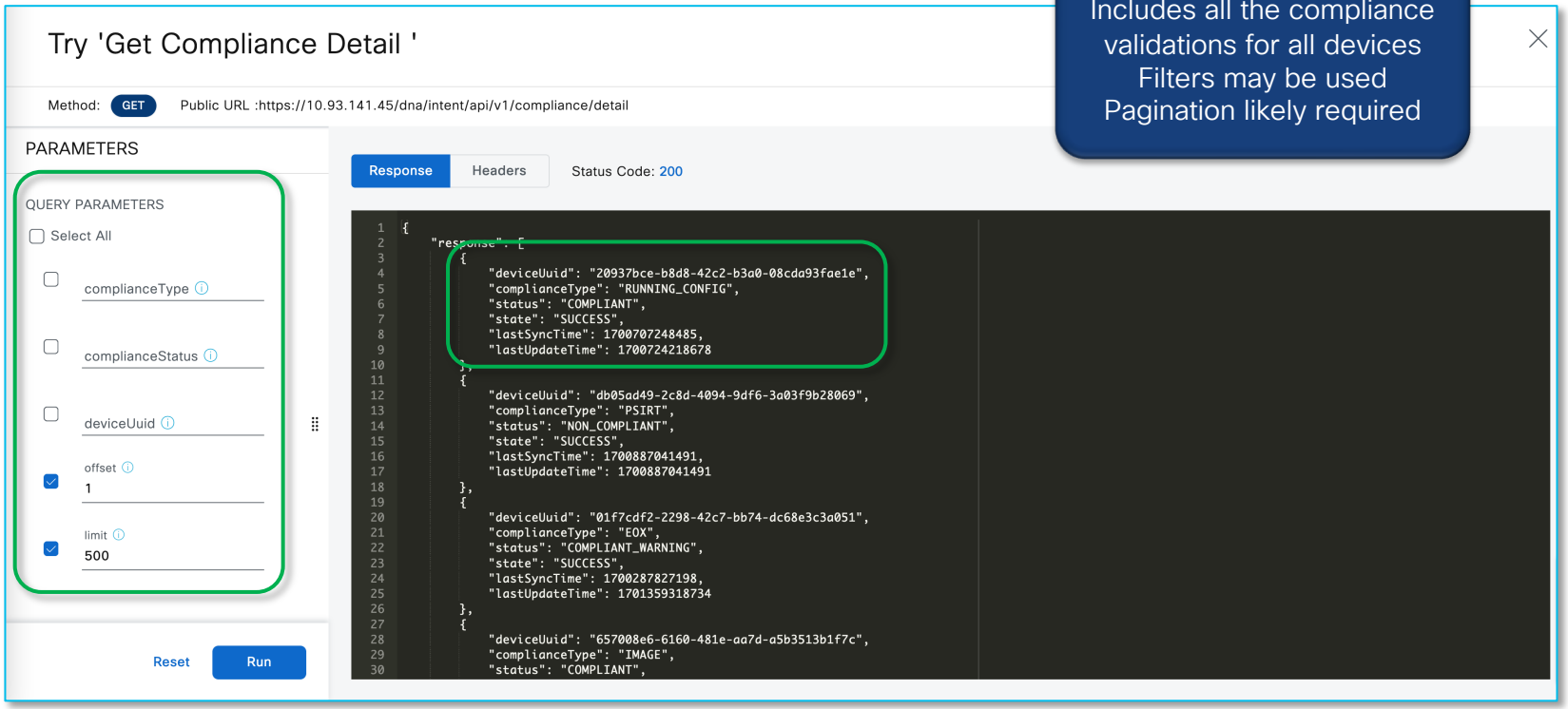
The compliance APIs support:

- Trigger a compliance check
- Retrieve the task details
- Get counts for all compliance checks
- Status, summary and details for device compliance

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

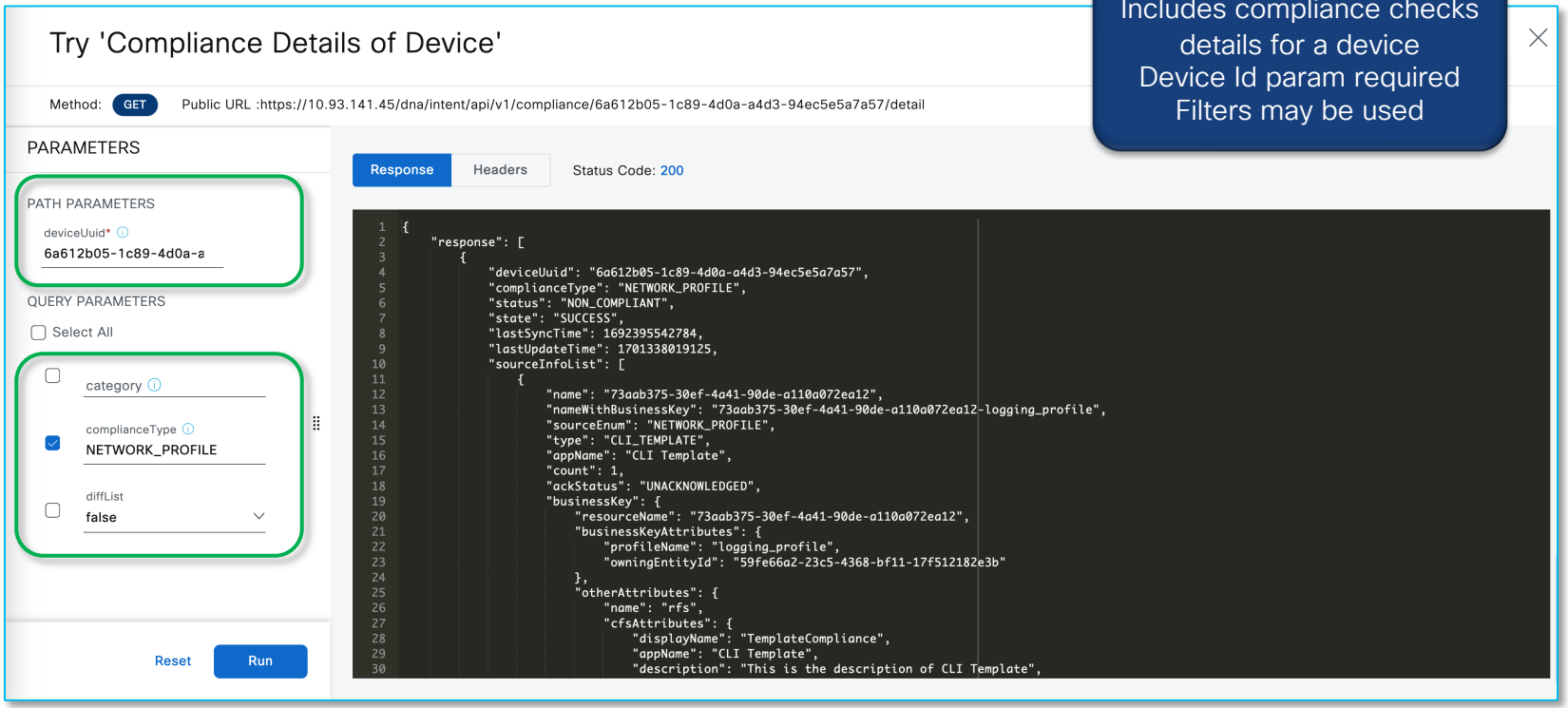
Get Compliance Details API

Includes all the compliance validations for all devices
Filters may be used
Pagination likely required



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

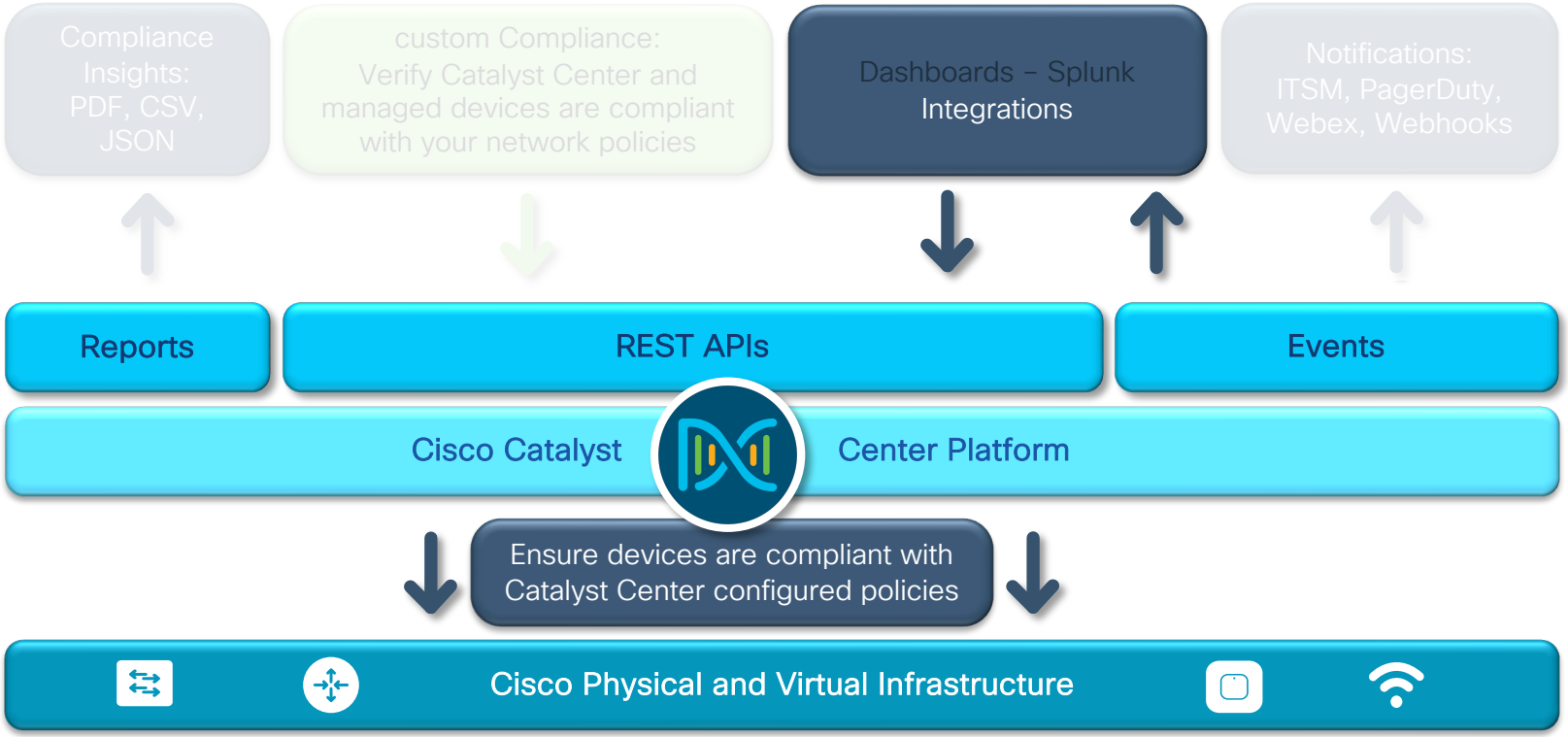
Device Compliance Details API



Includes compliance checks details for a device
Device Id param required
Filters may be used

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Network Compliance Integrations



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Catalyst Center Non-Compliant Devices

App Workflow

Resources:
- Cisco Catalyst Center
- Python SDK

Non-Compliant Devices App:

- Retrieve the compliance status for all devices
- Using the device unique identifiers, find out the device hostname
- Create a report for all non-compliant devices

REST APIs

Catalyst Center Compliance and Inventory

Save report to file, JSON formatted

Note:

- *Catalyst Center APIs enable integrations with 3rd party platforms to access network compliance details*
- *Device details may be enriched or filtered by device family and roles, fabric roles, location, software version …*
- *Example: compliance status for all Access switches, at location PDX, running version 17.3.7*

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Compliance App Python Code

```
65
66      get_compliance_response = catalyst_center_api.compliance.get_compliance_detail()
67      network_compliance_info = get_compliance_response['response']
68      logging.info(' Collected Catalyst Center network compliance state')
69
70      # sort the list of compliance status, by device Id
71      network_compliance = sorted(network_compliance_info, key=lambda x: x['deviceUuid'])
72
73      # identify all compliance checks in environment
74      compliance_type = []
75      for item in network_compliance:
76          item_compliance = item['complianceType']
77          if item_compliance not in compliance_type:
78              compliance_type.append(item_compliance)
79
80      logging.info(' Type of compliance checks: ' + json.dumps(compliance_type))
81
82      # create report for non-compliant devices for each compliance type
83      compliance_report = {}
84      for item in compliance_type:
85          compliance_report.update({item: []})
86
87      # loop through each item in compliance and append to report, to the specific category
88      for item in network_compliance:
89          if item['status'] == 'NON_COMPLIANT':
90              item_compliance = item['complianceType']
91              device_id = item['deviceUuid']
92              device_info = catalyst_center_api.devices.get_device_by_id(id=device_id)
93              device_hostname = device_info['response']['hostname']
94              compliance_report[item_compliance].append(device_hostname)
95
96      logging.info(' Non-compliant devices report completed: ')
```

Get all devices compliance and sort output

Identify all types of compliance types

Create non-compliant devices report by identifying the device hostnames

Notes:

- *Only two API endpoints called*
- *Sorting the compliance API response by device Id, minimizes the number of API calls made to identify the device hostname*

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Compliance App Run

```
INFO:root: Application "catalyst_center_compliance.py" Start, 2023-12-03 15:37:30
INFO:root: Collected Catalyst Center network compliance state
INFO:root: Type of compliance checks: [
    "EOX",
    "NETWORK_SETTINGS",
    "PSIRT",
    "IMAGE",
    "RUNNING_CONFIG",
    "APPLICATION_VISIBILITY
    "NETWORK_PROFILE",
    "FABRIC"
]
INFO:root: Non-compliant de
INFO:root: {
    "EOX": [
        "LO-EDGE"
    ],
    "NETWORK_SETTINGS": [
        "PDX-M",
        "LO-CN",
        "LO-EDGE",
        "PDX-CORE1",
        "LO-BN"
    ],
    "IMAGE": [
        "LO-BN"
    ],
    "RUNNING_CONFIG": [
        "PDX-RO",
        "PDX-M",
        "NYC-RO",
        "SP",
        "PDX-RN",
        "LO-CN"
    ],
    "APPLICATION_VISIBILITY": [],
    "NETWORK_PROFILE": [
        "NYC-ACCESS",
        "PDX-STACK",
        "PDX-M",
        "PDX-RN",
        "LO-CN",
        "LO-EDGE",
        "PDX-CORE1",
        "LO-BN"
    ],
    "FABRIC": []
}
INFO:root: Saved the non-compliant devices report to file "compliance_report.json"
```

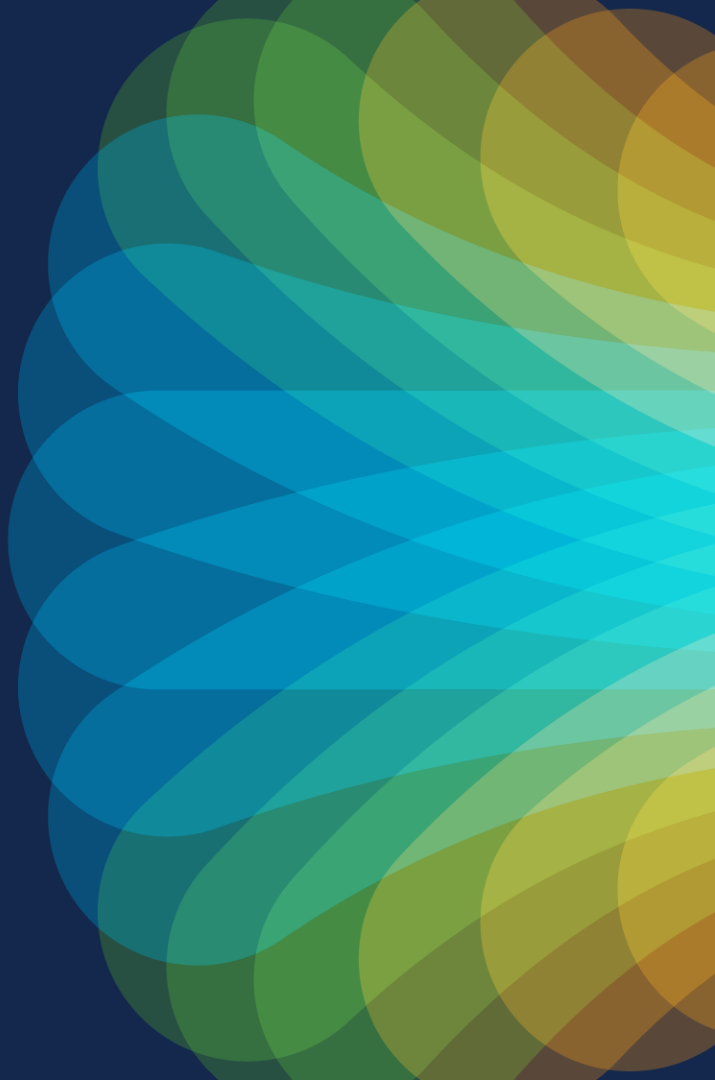
```
INFO:root: Non-compliant Core devices report completed:
INFO:root: {
    "EOX": [],
    "NETWORK_SETTINGS": [
        "PDX-CORE1"
    ],
    "PSIRT": [
        "PDX-CORE1"
    ],
    "IMAGE": [],
    "RUNNING_CONFIG": [],
    "APPLICATION_VISIBILITY": [],
    "NETWORK_PROFILE": [
        "PDX-CORE1"
    ],
    "FABRIC": []
}
INFO:root: Saved the Core non-compliant devices report to file "compliance_report.json"
INFO:root: End of Application "catalyst_center_compliance.py" Run: 2023-12-03 15:37:47
```

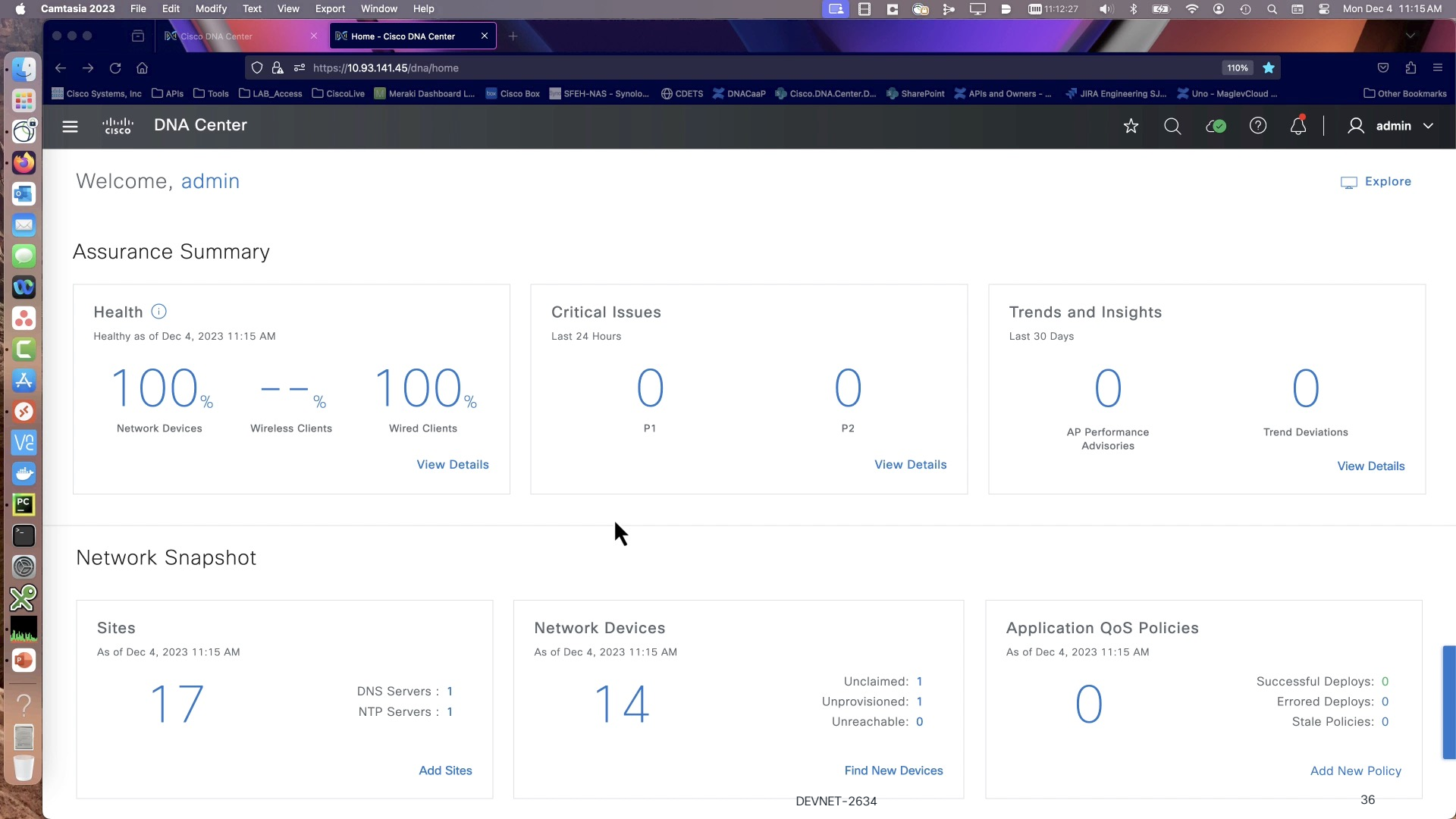
Use this file for:

- ITSM change management
- Remediation IaC workflows

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Catalyst Center Compliance Demo



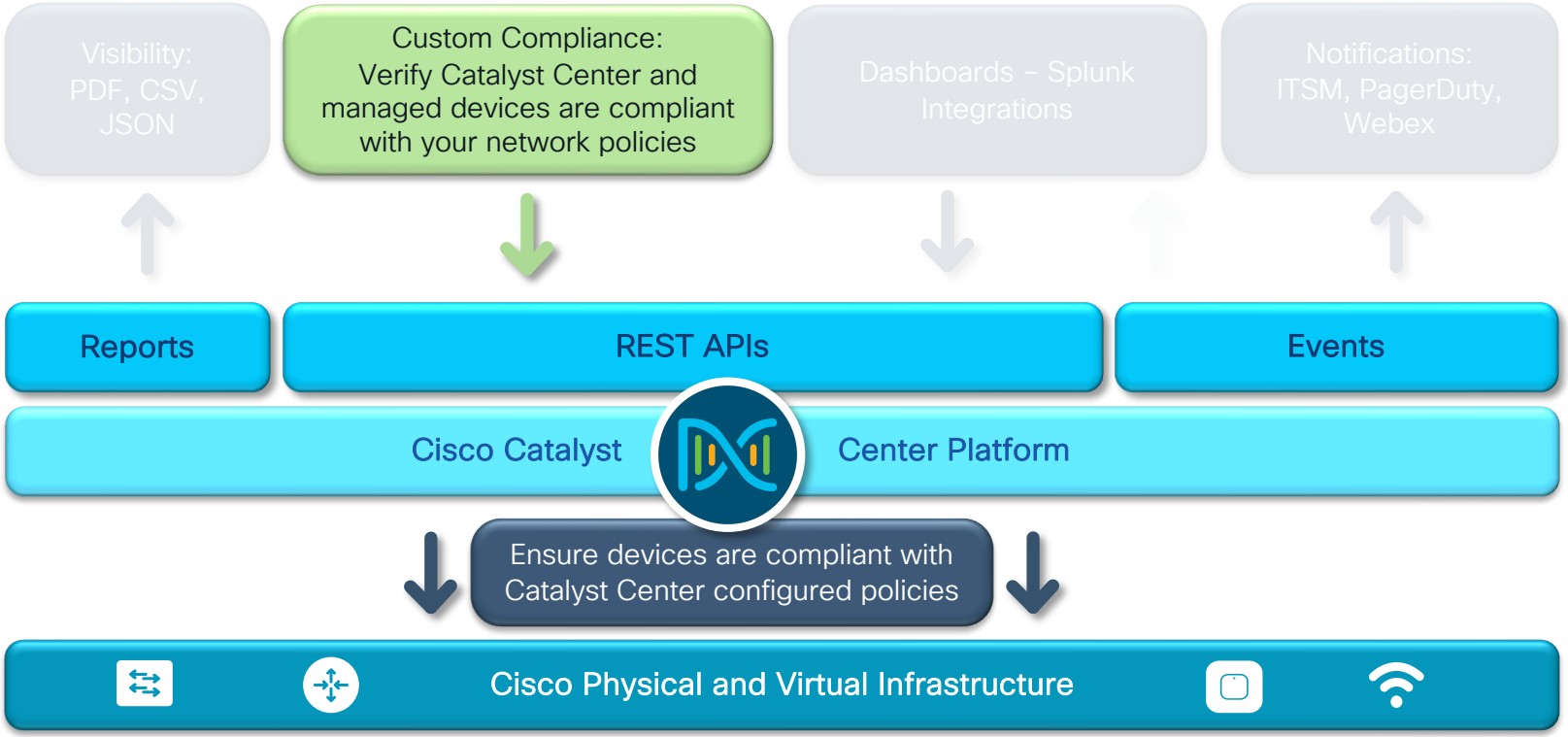

Camtasia 2023 File Edit Modify Text View Export Window Help
Mon Dec 4 11:15 AM
Cisco DNA Center
Home - Cisco DNA Center
https://10.93.141.45/dna/home
110%
Cisco Systems, Inc APIs Tools LAB_Access CiscoLive Meraki Dashboard L... Cisco Box SFEH-NAS - Synolo... CDETS DNACaaP Cisco.DNA.Center.D... SharePoint APIs and Owners - ... JIRA Engineering SJ... Uno - MaglevCloud ... Other Bookmarks
DNA Center
admin
Welcome, admin
Explore
Assurance Summary
Health
Healthy as of Dec 4, 2023 11:15 AM
100 %
Network Devices
-- %
Wireless Clients
100 %
Wired Clients
View Details
Critical Issues
Last 24 Hours
0
P1
0
P2
View Details
Trends and Insights
Last 30 Days
0
AP Performance Advisories
0
Trend Deviations
View Details
Network Snapshot
Sites
As of Dec 4, 2023 11:15 AM
17
DNS Servers : 1
NTP Servers : 1
Add Sites
Network Devices
As of Dec 4, 2023 11:15 AM
14
Unclaimed: 1
Unprovisioned: 1
Unreachable: 0
Find New Devices
Application QoS Policies
As of Dec 4, 2023 11:15 AM
0
Successful Deploys: 0
Errored Deploys: 0
Stale Policies: 0
Add New Policy

Agenda

- Platforms
- Catalyst Center Compliance
- Custom Network Compliance
- Developer Resources
- Lessons Learned

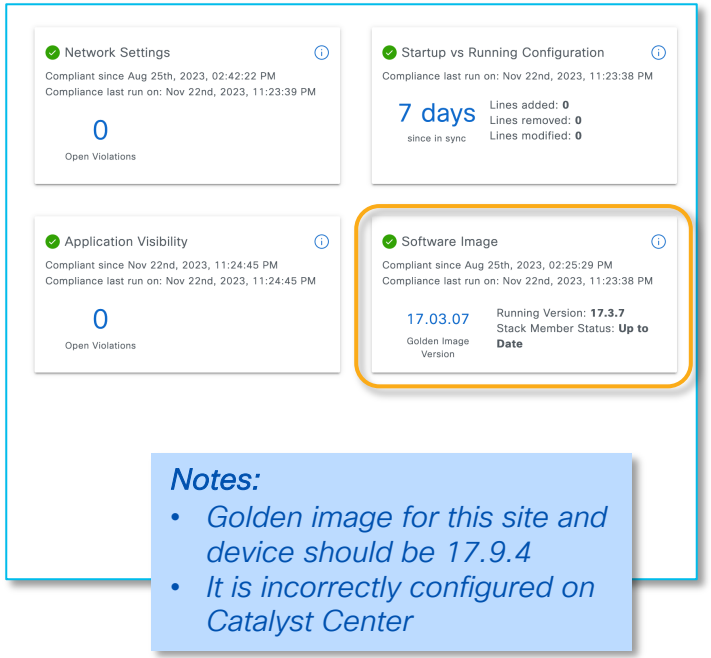
© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Custom Network Compliance



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Custom Network Compliance



- The device is compliant with Catalyst Center network settings and runs the golden image
- Is it compliant with your network policies?
- For a continuous compliance validation with corporate policies, we need to:
  - Verify intent is configured correctly on Catalyst Center
  - Catalyst Center network compliance will validate devices are compliant with intent

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Custom Network Compliance

App Workflow

Resources:
- Cisco Catalyst Center
- Python SDK
- GitHub Repo

GitHub — Intent Configurations

REST APIs

Custom Network Compliance App:

- Retrieve the compliance intent from GitHub
- Verify the Catalyst Center configurations meet the defined intent
- Create a report, save to file, option to push to GitHub

REST APIs

Catalyst Center Compliance, Inventory, Settings, Configs

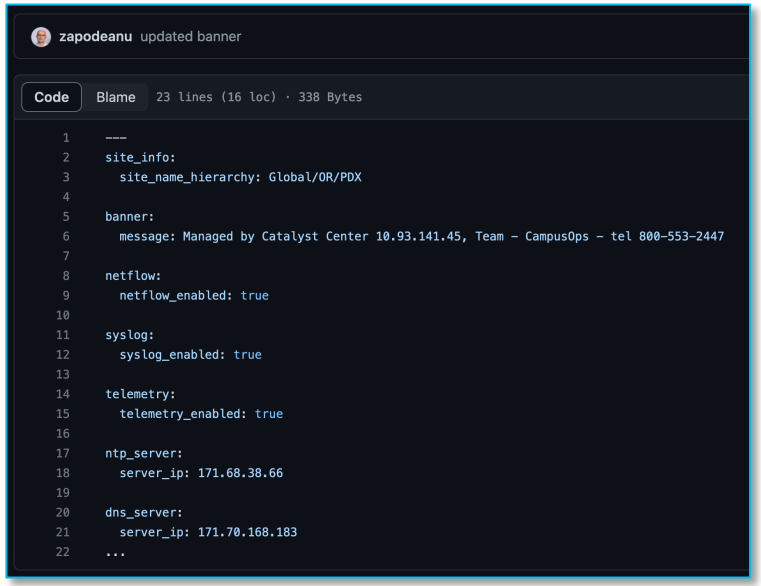
Save report, JSON formatted, optionally push to 3rd party platform

Notes:
- *Supports multi-Catalyst Center intent compliance validations*
- *App developed by developer + network engineer*

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

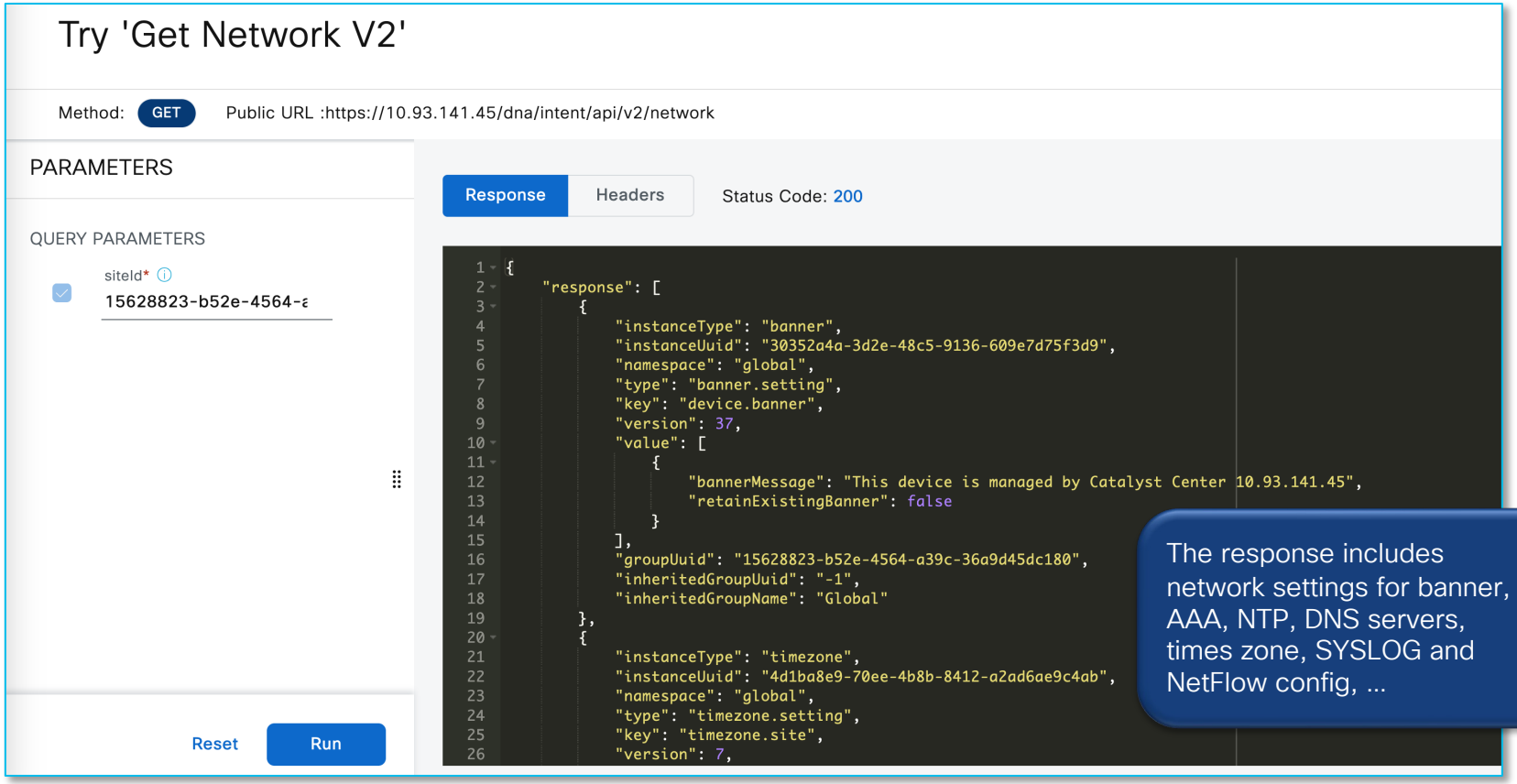
Custom Network Settings Compliance

- Pull from GitHub the file with the custom network settings
- Parse the banner message, NTP and DNS servers' intent configuration
- Verify if intent matches the Catalyst Center configurations



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Catalyst Center Network Settings API



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

GitHub APIs

```python
def get_private_repos(username, github_token):
    """
    This function will return private repos for user referred by the User Access Token
    :param: username: user for which to return the repos
    :param: github_token: Personal access token for user
    :return: repos_list
    """
    url = GITHUB_URL + '/user/repos?type=private'
    header = {'Accept': 'application/vnd.github+json', 'Authorization': 'token ' + GITHUB_TOKEN}
    response = requests.get(url, headers=header, verify=True)
    response_json = response.json()
    repos_list = []
    for repo in response_json:
        repos_list.append(repo['name'])
    return repos_list
```

```python
def get_repo_file_content(username, repo_name, file_name):
    """
    This function will return the content of the file from the repo
    :param username: GitHub username
    :param repo_name: GitHub repo
    :param file_name: file name
    :return: return the file content
    """
    url = GITHUB_URL + '/repos/' + username + '/' + repo_name + '/contents/' + file_name
    header = {'Authorization': 'token ' + GITHUB_TOKEN}
    response = requests.get(url, headers=header)
    response_json = response.json()
    file_content = response_json['content']
    file_content_encoding = response_json.get('encoding')
    if file_content_encoding == 'base64':
        file_content = base64.b64decode(file_content)
    return file_content
```

zapodeanu updated banner

Code | Blame | 23 lines (16 loc) · 338 Bytes | Code 55% faster with GitHub Copilot

```yaml
---
site_info:
  site_name_hierarchy: Global/OR/PDX

banner:
  message: Managed by Catalyst Center 10.93.141.45, Team - CampusOps - tel 800-553-2447

netflow:
  netflow_enabled: true

syslog:
  syslog_enabled: true

telemetry:
  telemetry_enabled: true

ntp_server:
  server_ip: 171.68.38.66

dns_server:
  server_ip: 171.70.168.183
...
```

- Get all repos, including the private repos, for user
- Retrieve the intent configuration file, in this case for network settings

Note:

- *Verify which GitHub APIs provide private repos info*

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Network Settings Compliance App Python Code

```
66
67      # get all repos for user
68      repos = github_apis.get_private_repos(username=GITHUB_USERNAME, github_token=GITHUB_TOKEN)
69
70      # verify if repo exists
71      if GITHUB_REPO not in repos:
72          logging.info(' Repo "' + GITHUB_REPO + '" not found!')
73          return
74      logging.info(' Repo "' + GITHUB_REPO + '" found!')
75
76      # get the network settings intent file
77      intent_network_settings = github_apis.get_repo_file_content(username=GITHUB_USERNAME, repo_name=GITHUB_REPO,
78                                                                  file_name=FILE_NAME)
79      logging.info(' File "' + FILE_NAME + '" found!')
80
81      # decode the YAML file
82      intent_config = yaml.safe_load(intent_network_settings)
83
84      # parse the input data
85      site_name_hierarchy = intent_config['site_info']['site_name_hierarchy']
86      banner_message = intent_config['banner']['message']
87      ntp_server = intent_config['ntp_server']['server_ip']
88      dns_server = intent_config['dns_server']['server_ip']
89
90      logging.info(' Intent network settings from GitHub:')
91      logging.info('   Site hierarchy: ' + site_name_hierarchy)
92      logging.info('   Banner: ' + banner_message)
93      logging.info('   NTP server: ' + ntp_server)
94      logging.info('   DNS server: ' + dns_server)
95
```

Get from GitHub, private repo, the intent config file

Parse the YAML file, in this case Network Settings

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Compliance App Python Code – continued

```
100
101     # get the site Id for the site with the name
102     response = catalyst_center_api.sites.get_site(name=site_name_hierarchy)
103     site_id = response['response'][0]['id']
104     logging.info(' The site id for the site: "' + site_name_hierarchy + '" is: ' + site_id)
105
106     # collect the network settings for site
107     response = catalyst_center_api.network_settings.get_network_v2(site_id)
108     site_network_settings = response['response']
109     logging.info(' Collected the site settings')
110
111     # verify the settings for NTP
112     network_settings_ntp_status = {'ntp': 'not_compliant'}
113     for item in site_network_settings:
114         if item['instanceType'] == 'ip' and item['key'] == 'ntp.server' and item['value'][0] == ntp_server:
115             network_settings_ntp_status.update({'ntp': 'compliant'})
116             break
117
118     # verify the settings for DNS
119     network_settings_dns_status = {'dns': 'not_compliant'}
120     for item in site_network_settings:
121         if item['instanceType'] == 'dns' and item['key'] == 'dns.server' and item['value'][0]['primaryIpAddress'] == dns_server:
122             network_settings_dns_status.update({'dns': 'compliant'})
123             break
124
125     # verify the settings for Banner message
126     network_settings_banner_status = {'banner': 'not_compliant'}
127     for item in site_network_settings:
128         if item['instanceType'] == 'banner' and item['value'][0]['bannerMessage'] == banner_message:
129             network_settings_banner_status.update({'banner': 'compliant'})
130             break
131
```

Get the site unique identifier and network settings

Validate the GitHub intent matches the configured settings for site

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Intent vs Configured Network Settings

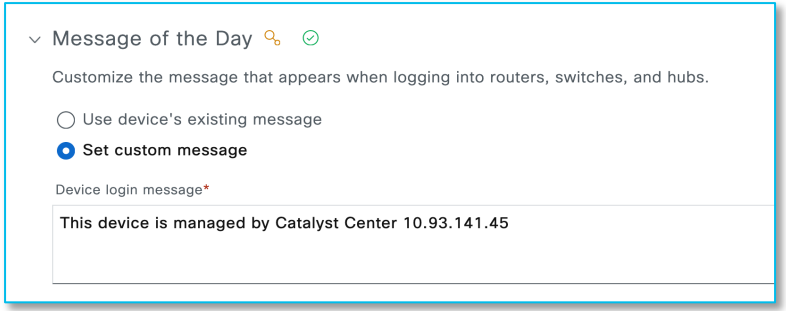
zapodeanu updated banner

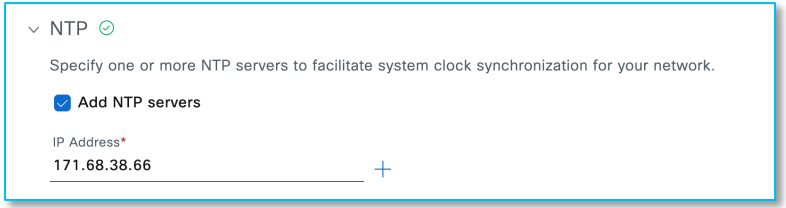
Code | Blame | 23 lines (16 loc) · 338 Bytes | Code 55% faster with GitHub Copilot

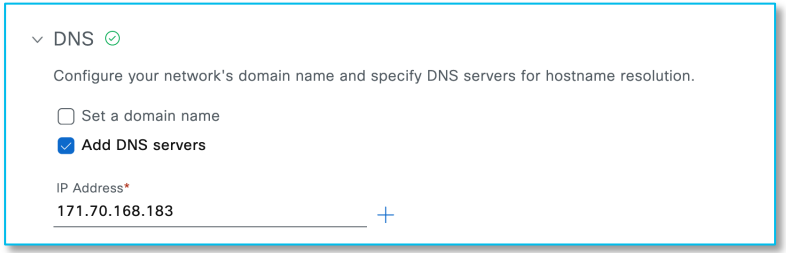
```yaml
---
site_info:
  site_name_hierarchy: Global/OR/PDX

banner:
  message: Managed by Catalyst Center 10.93.141.45, Team - CampusOps - tel 800-553-2447

netflow:
  netflow_enabled: true

syslog:
  syslog_enabled: true

telemetry:
  telemetry_enabled: true

ntp_server:
  server_ip: 171.68.38.66

dns_server:
  server_ip: 171.70.168.183
...
```

Note:

- *Build your data model for your Custom compliance policy*







© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Network Settings Compliance App Run

```
INFO:root: Application "network_settings_compliance.py" Start, 2023-12-04 01:01:43
INFO:root: Repo "compliance_state" found!
INFO:root: File "intent_network_settings.yaml" found!
INFO:root: Intent network settings from GitHub:
INFO:root:   Site hierarchy: Global/OR/PDX
INFO:root:   Banner: Managed by Catalyst Center 10.93.141.45, Team - CampusOps - tel 800-553-2447
INFO:root:   NTP server: 171.68.38.66
INFO:root:   DNS server: 171.70.168.183
INFO:root: The site id for the site: "Global/OR/PDX" is: 15628823-b52e-4564-a39c-36a9d45dc180
INFO:root: Collected the site settings
INFO:root: Network Settings compliance report:
INFO:root:   {
    "site_hierarchy": "Global/OR/PDX",
    "compliance_status": {
        "dns": "compliant",
        "ntp": "compliant",
        "banner": "not_compliant",
        "banner_intent": "Managed by Catalyst Center 10.93.141.45, Team - CampusOps - tel 800-553-2447",
        "banner_configured": "This device is managed by Catalyst Center 10.93.141.45"
    }
}
INFO:root: Saved the network settings report to file "network_settings_report.json"
INFO:root: End of Application "network_settings_compliance.py" Run: 2023-12-04 01:01:48
```

Use the report file for:

- 3rd party integrations (ITSM)
- Remediation IaC workflow update settings + reprovision devices

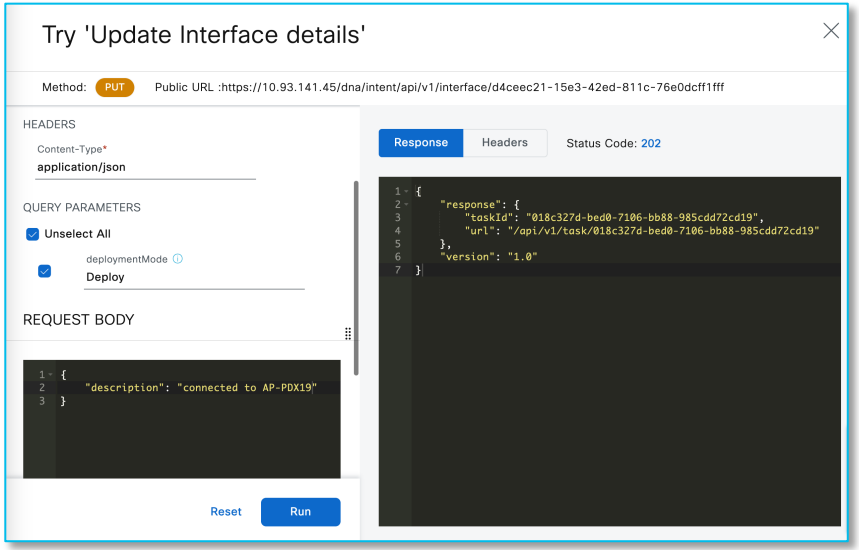
Compliant Settings

Non-Compliant settings and details for intent and configured

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

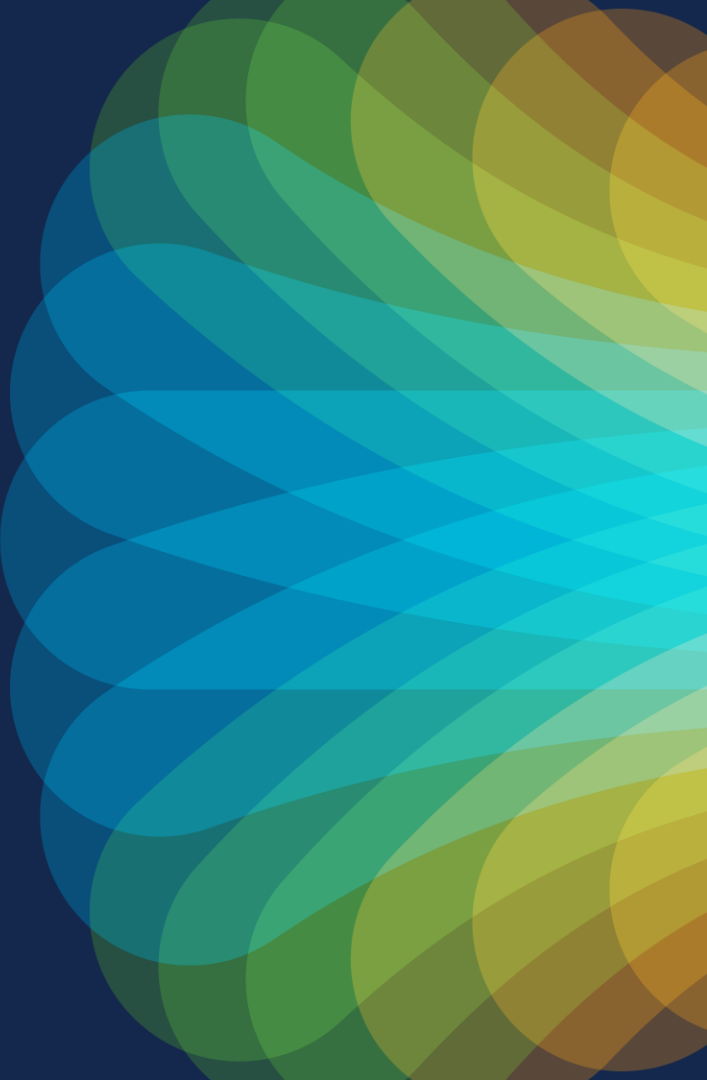
Custom Compliance

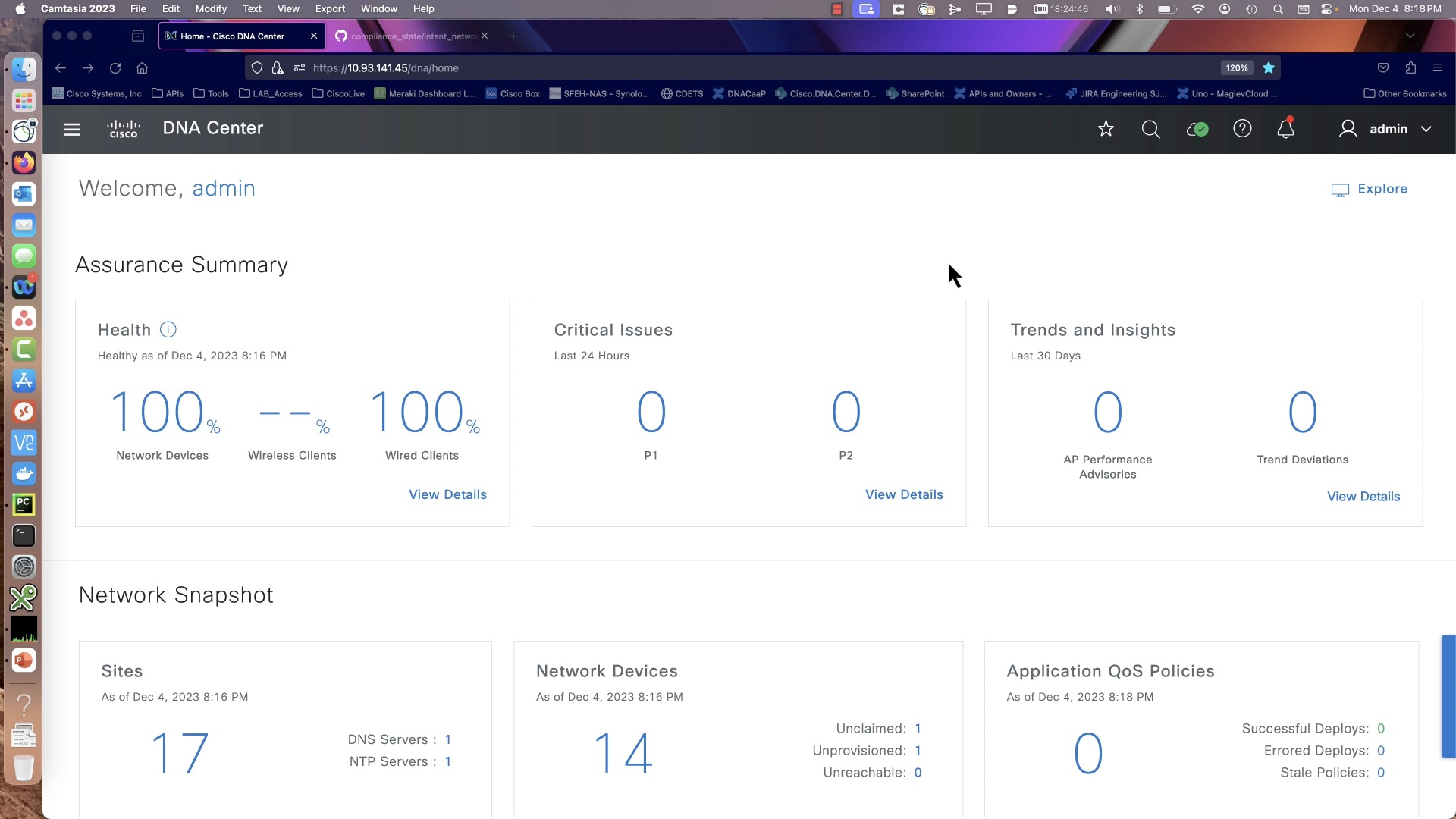
- Using the same workflow, we may verify network settings for hundreds or thousands of sites
- Maintain a sync between projects and templates that are configured on Catalyst Center with those defined on GitHub
- Golden software image configuration and tags
- Automate time zone config based on device location
- Interface descriptions and operations based on your policies
- Integrations and ITSM change management



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

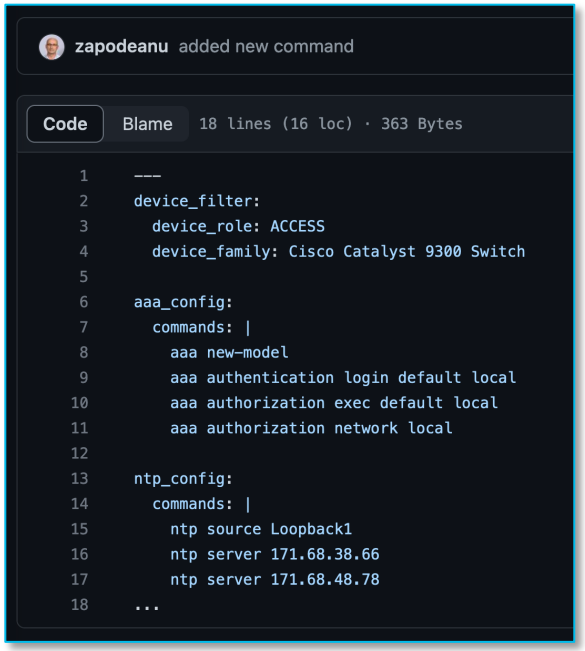
Network Settings Compliance Demo




Camtasia 2023 File Edit Modify Text View Export Window Help
Mon Dec 4 8:18 PM
Home - Cisco DNA Center
https://10.93.141.45/dna/home
120%
DNA Center
admin
Welcome, admin
Explore
Assurance Summary
Health
Healthy as of Dec 4, 2023 8:16 PM
100%
Network Devices
--%
Wireless Clients
100%
Wired Clients
View Details
Critical Issues
Last 24 Hours
0
P1
0
P2
View Details
Trends and Insights
Last 30 Days
0
AP Performance Advisories
0
Trend Deviations
View Details
Network Snapshot
Sites
As of Dec 4, 2023 8:16 PM
17
DNS Servers : 1
NTP Servers : 1
Network Devices
As of Dec 4, 2023 8:16 PM
14
Unclaimed: 1
Unprovisioned: 1
Unreachable: 0
Application QoS Policies
As of Dec 4, 2023 8:18 PM
0
Successful Deploys: 0
Errored Deploys: 0
Stale Policies: 0

Custom Device Config Compliance

- Pull from GitHub the file with the intent device configs
- Verify the desired configuration matches the configuration of devices managed by Catalyst Center
- Create report



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Custom Device Config Compliance Workflow

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

How to Compare Text Files in Python

Python “difflib.Differ” Class

- Compares sequences of lines of text and producing human-readable differences or deltas.
- It will compare sequences of lines, and sequences of characters within similar (near-matching) lines.
- Each line of a Differ delta begins with a two-letter code

Code	Meaning
'- '	line unique to sequence 1
'+ '	line unique to sequence 2
' '	line common to both sequences
'? '	line not present in either input sequence

Ref: https://docs.python.org/3/library/difflib.html

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Config Compliance Check – Calculate Diff

"cli_commands.txt"

```
aaa new-model
aaa authentication login default local
aaa authorization exec default local
aaa authorization network local
```

"device_config.txt"

```
logging buffered 20960
logging console emergencies
aaa new-model
!
aaa authentication login default local
aaa authorization exec default
!
aaa session-id common
clock timezone PST -8 0
```

"cli_difference"

```
+ logging buffered 20960
+ logging console emergencies
  aaa new-model
+ !
  aaa authentication login default local
- aaa authorization exec default local
?                                   ------
+ aaa authorization exec default
- aaa authorization network local
+ !
+ aaa session-id common
+ clock timezone PST -8 0
```

Calculate using the "difflib.Differ" class the delta between CLI commands and the device config

- Lines starting with " " are in both files
- Lines starting with "- " are missing in the second file
- Lines starting with "+ " are missing in the first file

Notes:

- *Order of the param files is important*
- *In this use case: config_diff.compare ("cli_commands.txt", "device_config.txt")*

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Config Compliance Check – Create Report

"cli_difference"

```
+ logging buffered 20960
+ logging console emergencies
  aaa new-model
+ !
  aaa authentication login default local
- aaa authorization exec default local
?                                  ------
+ aaa authorization exec default
- aaa authorization network local
+ !
+ aaa session-id common
+ clock timezone PST -8 0
```

- From the "cli_difference" select the lines that start with "- "
- These are the commands existing in the "cli_commands.txt" and not in the device config
- If there no lines starting with "- " device is compliant with the Custom policy

Notes:

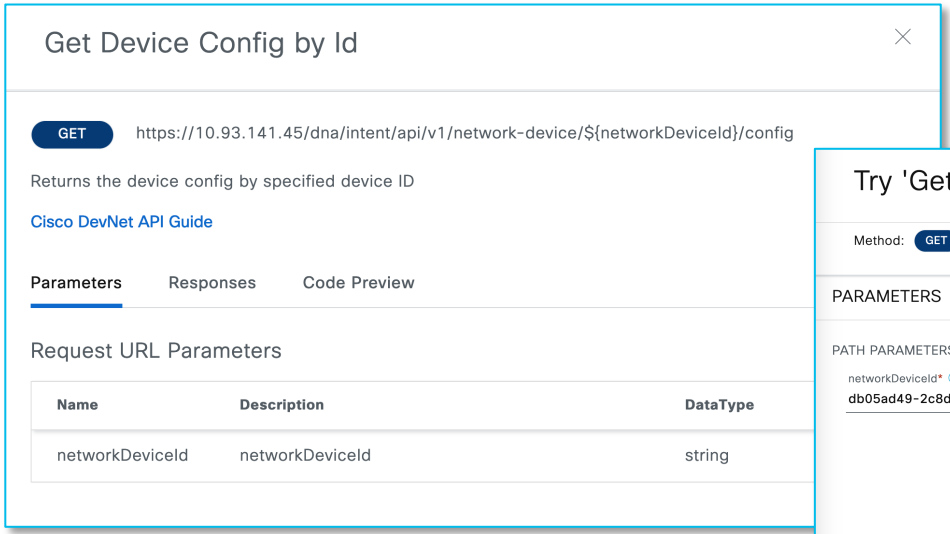
- *The difflib library has few other options to compare text*
- *There are other libraries that may be used.*

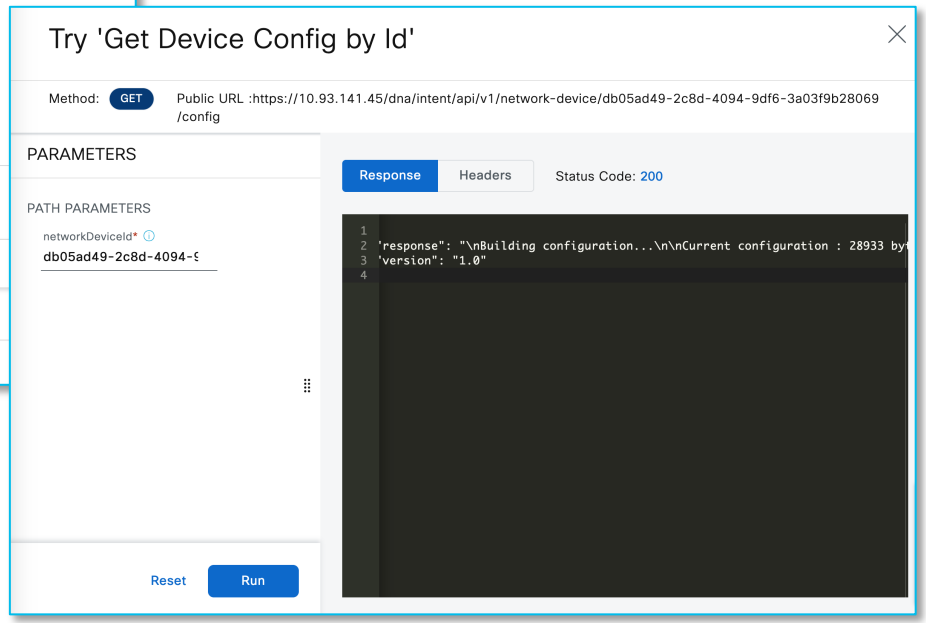
Missing commands

```
aaa authorization exec default local
aaa authorization network local
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Get Device Config By Device ID API





- The configuration of the device is based on the last time Catalyst Center synchronized the device
- To collect real-time device configurations, use the command-runner APIs

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Device Config Compliance App Code

```python
# verify if repo exists
if GITHUB_REPO not in repos:
    logging.info(' Repo "' + GITHUB_REPO + '" not found!')
    return
logging.info(' Repo "' + GITHUB_REPO + '" found!')

# get the device config  compliance intent file
file_content = github_apis.get_repo_file_content(username=GITHUB_USERNAME,
                                                 repo_name=GITHUB_REPO,
                                                 file_name=FILE_NAME)
logging.info(' File "' + FILE_NAME + '" found!')

# parse the input data
intent_config = yaml.safe_load(file_content)
aaa_config = intent_config['aaa_config']['commands']
ntp_config = intent_config['ntp_config']['commands']

# parse the device policy
device_role = intent_config['device_filter']['device_role']
device_family = intent_config['device_filter']['device_family']

logging.info(' Device configs from GitHub:')
logging.info('   aaa_
logging.info('   ntp_
```

Pull from GitHub repo and the file with the device compliance policy, and parse data

Get the device count from Catalyst Center, retrieve the device list, enrich with location and SDA device role

```python
# collect device inventory
# get the device count
response = catalyst_center_api.devices.get_device_count()
device_count = response['response']
logging.info(' Number of devices managed by Catalyst Center: ' + str(device_count))

# get the device info list
offset = 1
limit = 500
device_list = []
while offset <= device_count:
    response = catalyst_center_api.devices.get_device_list(offset=offset)
    offset += limit
    device_list.extend(response['response'])
logging.info(' Collected the device list from Catalyst Center')

# create device inventory, add location and fabric roles

device_inventory = []
for device in device_list:
    # select which inventory to add the device to
    if device.family != "Unified AP":...

logging.info(' Retrieved the device location and fabric role')
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Device Config Compliance App Code – continued

```
# collect the device configs for the devices that match the role
logging.info(' Compliance checks for devices that match the device filter')
for item in device_inventory:
    if item['role'] == device_role and item['device_family'] == device_family:
        response = catalyst_center_api.devices.get_device_config_by_id(item['device_id'])
        device_config = response['response']
        with open(item['hostname'] + '_config.txt', 'w') as f:
            f.write(device_config)
        logging.info(' Saved the device config ' + item['hostname'] + '_config.txt')
        logging.info(' Device: ' + item['hostname'] + ' :')

        # check common lines between the two files - device config compliance commands and running config
        aaa_config_file = open('aaa_config.txt', 'r').readlines()
        ntp_config_file = open('ntp_config.txt', 'r').readlines()
        config_file = open(item['hostname'] + '_config.txt', 'r').readlines()

        aaa_difference = difflib.Differ()
        aaa_commands_list = []
        for line in aaa_difference.compare(aaa_config_file, config_file):
            if line.startswith('-'):
                clean_line = line.strip()
                aaa_commands_list.append(clean_line)
        if len(aaa_commands_list) == 0:
            logging.info('    - AAA config check passed')
        else:
            logging.info('    - AAA config check failed, missing commands:')
            for command in aaa_commands_list:
                logging.info('        ' + command)
```

Call Catalyst Center API to get the device config, save config to file
Use the difflib>Differ class
Validate the “aaa_config.txt” commands are in the device config

Validate the “ntp_config.txt” commands are in the device config

```
ntp_difference = difflib.Differ()
ntp_commands_list = []
for line in ntp_difference.compare(ntp_config_file, config_file):
    if line.startswith('-'):
        clean_line = line.strip()
        ntp_commands_list.append(clean_line)
if len(ntp_commands_list) == 0:
    logging.info('    - NTP config check passed')
else:
    logging.info('    - NTP config check failed, missing commands:')
    for command in ntp_commands_list:
        logging.info('        ' + command)
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Device Config Compliance App Logging

zapodeanu added new command

Code | Blame | 18 lines (16 loc) · 363 Bytes

```
---
device_filter:
  device_role: ACCESS
  device_family: Cisco Catalyst 9300

aaa_config:
  commands: |
    aaa new-model
    aaa authentication login default local
    aaa authorization exec default local
    aaa authorization network local

ntp_config:
  commands: |
    ntp source Loopback1
    ntp server 171.68.38.66
    ntp server 171.68.48.78
...
```

```
INFO:root: Application "device_config_compliance.py" Start, 2023-12-03 17:55:41
INFO:root: Repo "compliance_state" found!
INFO:root: File "custom_network_compliance.yaml" found!
INFO:root: Device configs from GitHub:
INFO:root:
aaa new-model
aaa authentic
aaa authoriza
aaa authoriza

INFO:root:
ntp source Lo
ntp server 17
ntp server 17
```

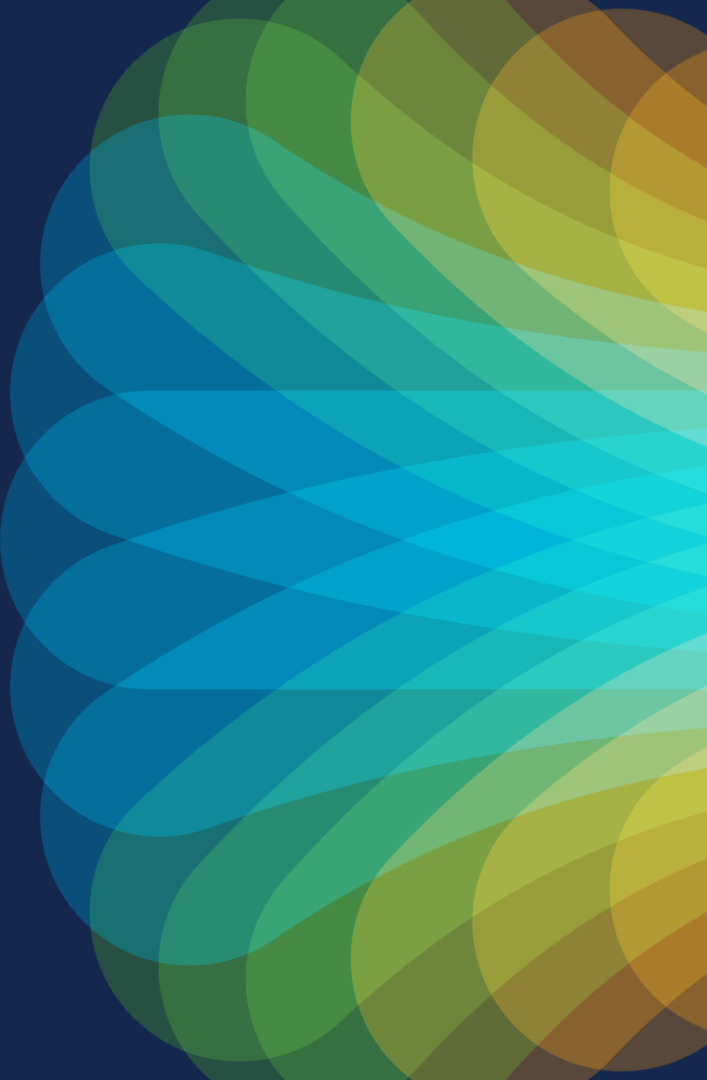
```
INFO:root: Compliance device filter:
INFO:root:   device_role: ACCESS
INFO:root:   device_family: Cisco Catalyst 9300 Switch
INFO:root: Number of devices managed by Cat
INFO:root: Collected the device list from
INFO:root: Retrieved the device location a
INFO:root: Saved the device inventory to f
INFO:root: Compliance checks for devices t
INFO:root: Saved the device config LO-CN_c
INFO:root: Device: LO-CN :
INFO:root:    - AAA config check failed, missing commands:
INFO:root:        - aaa authorization network local
INFO:root:    - NTP config check failed, missing commands:
INFO:root:        - ntp source Loopback1
INFO:root:        - ntp server 171.68.48.78
INFO:root: Saved the device config PDX-STACK_config.txt
INFO:root: Device: PDX-STACK :
INFO:root:    - AAA config check failed, missing commands:
INFO:root:        - aaa authorization network local
INFO:root:    - NTP config check failed, missing commands:
INFO:root:        - ntp server 171.68.48.78
INFO:root: End of Application "device_config_compliance.py" Run: 2023-12-03 17:56:01
```

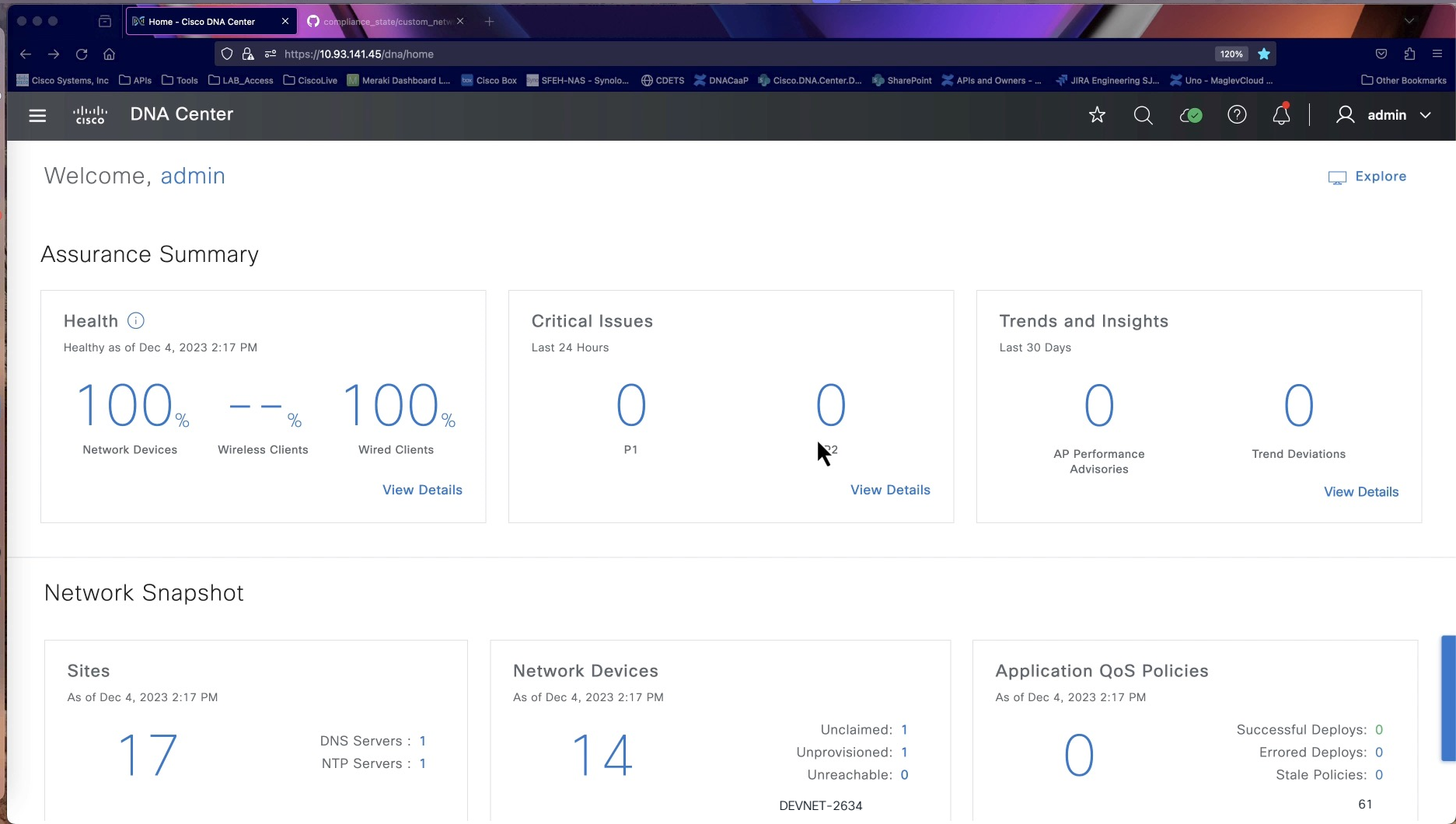
Use this report for:

- ITSM change management
- Remediation IaC workflows, create the CLI template and deploy to devices

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

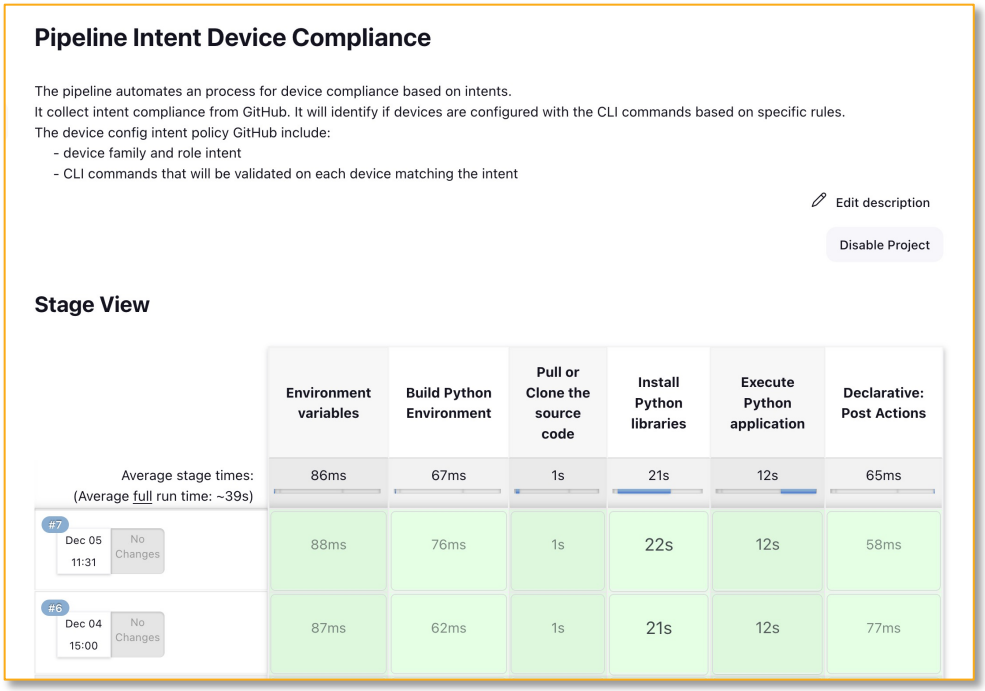
Device Compliance Demo




Camtasia 2023 File Edit Modify Text View Export Window Help
Mon Dec 4 2:17 PM
Home - Cisco DNA Center
https://10.93.141.45/dna/home
120%
DNA Center
admin
Welcome, admin
Explore
Assurance Summary
Health
Healthy as of Dec 4, 2023 2:17 PM
100%
Network Devices
--%
Wireless Clients
100%
Wired Clients
View Details
Critical Issues
Last 24 Hours
0
P1
0
P2
View Details
Trends and Insights
Last 30 Days
0
AP Performance Advisories
0
Trend Deviations
View Details
Network Snapshot
Sites
As of Dec 4, 2023 2:17 PM
17
DNS Servers : 1
NTP Servers : 1
Network Devices
As of Dec 4, 2023 2:17 PM
14
Unclaimed: 1
Unprovisioned: 1
Unreachable: 0
DEVNET-2634
Application QoS Policies
As of Dec 4, 2023 2:17 PM
0
Successful Deploys: 0
Errored Deploys: 0
Stale Policies: 0
61

Options to Run Compliance Applications

- On-demand or on your automation platform
- Triggered by APIs or events
- Scheduled
- Part of your IaC pipelines, example Jenkins pipeline



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

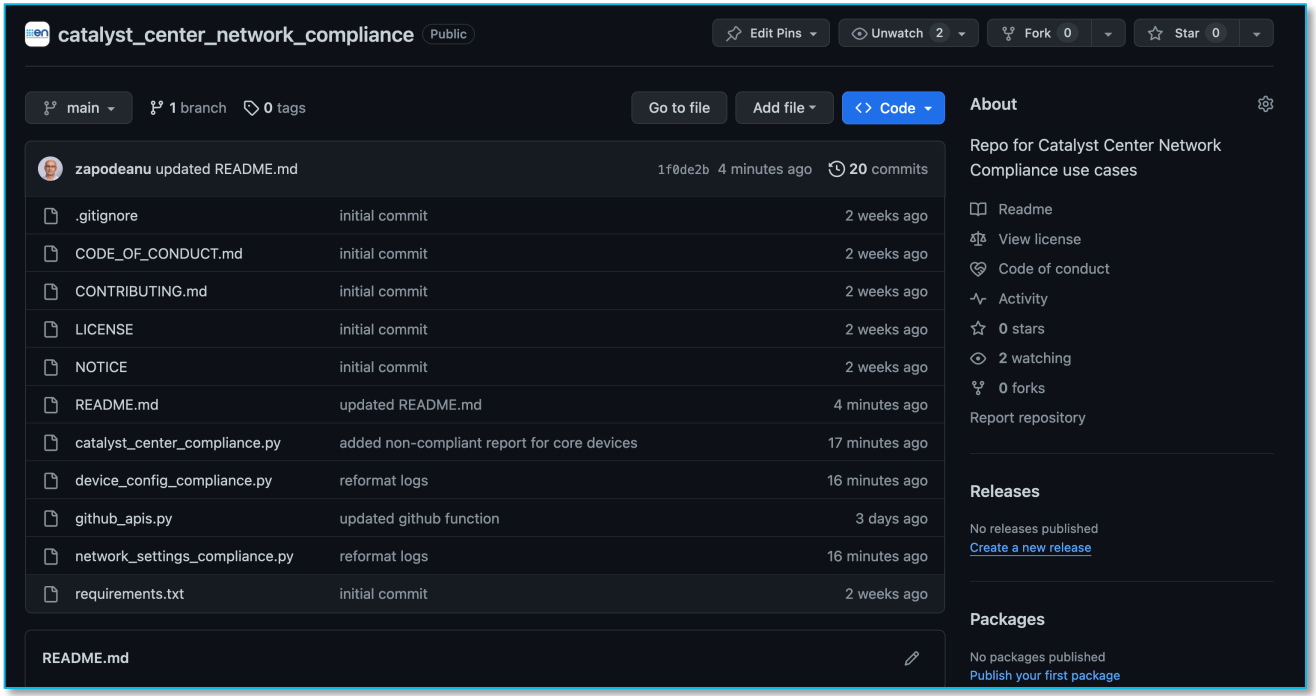
Agenda

- Platforms
- Catalyst Center Compliance
- Custom Network Compliance
- Developer Resources
- Lessons Learned

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

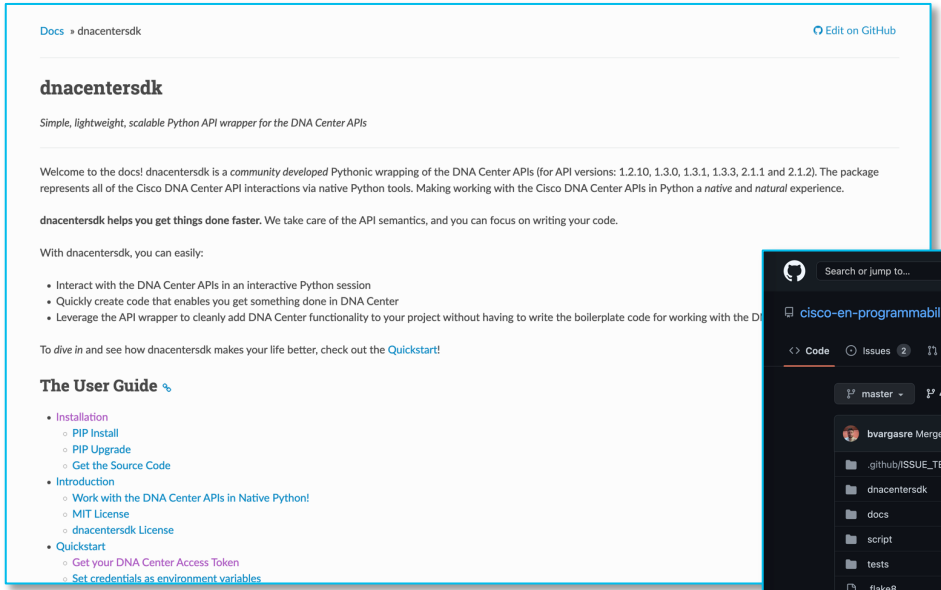
Catalyst Center Network Compliance Use Cases

https://github.com/cisco-en-programmability/catalyst_center_network_compliance



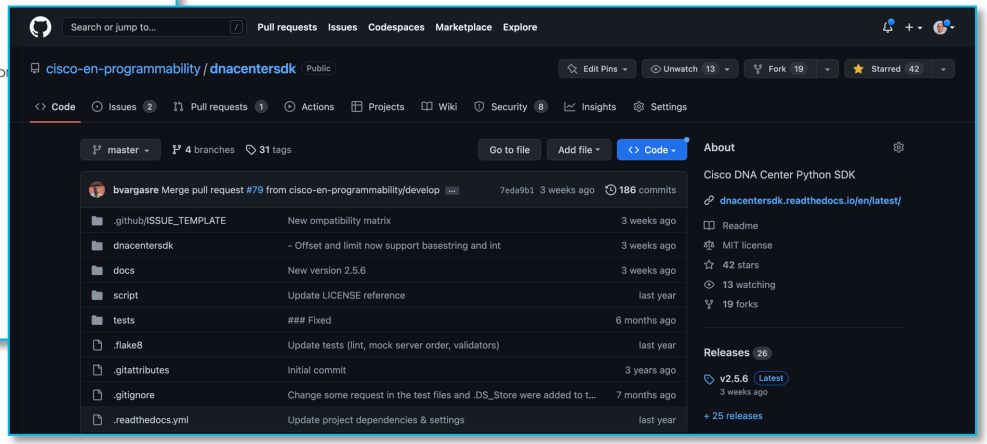
© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Python SDK
Cisco DNA Center Version Support 1.2.10 to 2.3.5.3



https://dnacentersdk.readthedocs.io

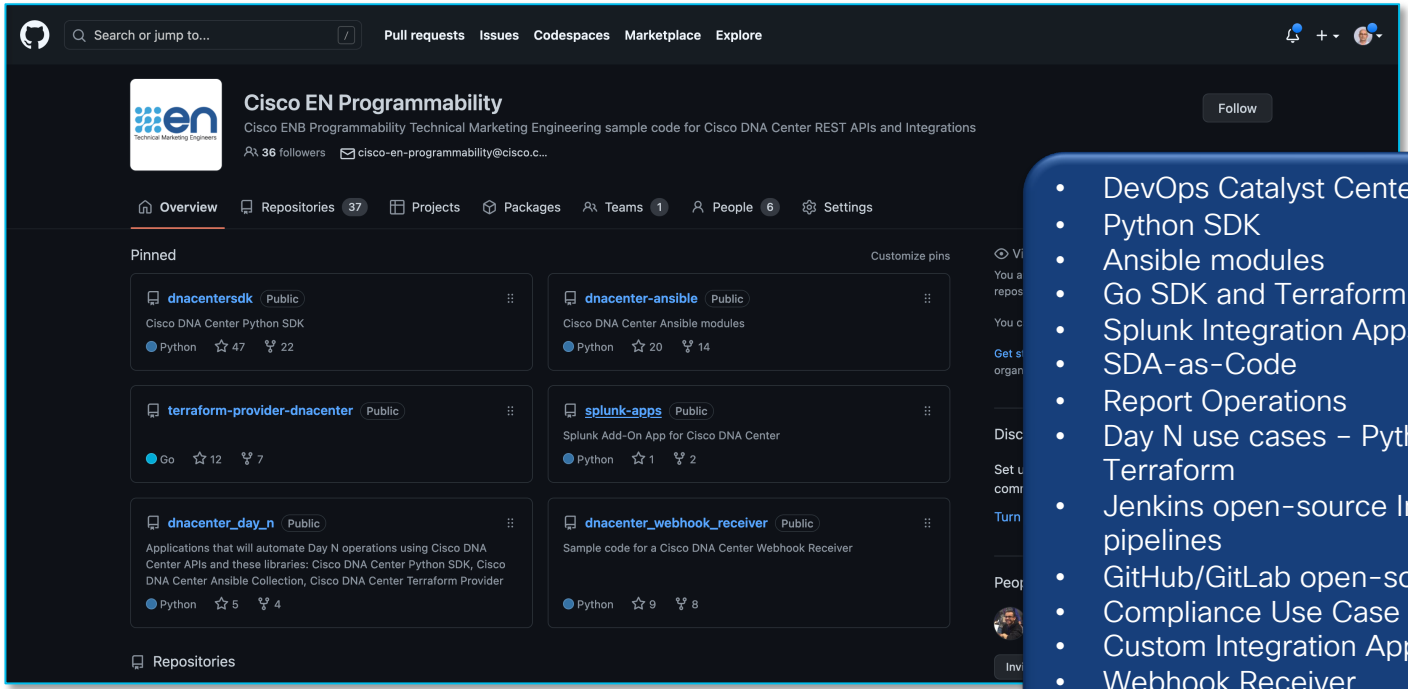
- Includes all Cisco Catalyst Center REST APIs
- Support for Cisco Catalyst Center version 2.3.5.3, (2.3.7.3 pending release)
- First SDK version August 2019
- Downloads last month 6,200+



https://github.com/cisco-en-programmability/dnacentersdk

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Cisco EN Programmability GitHub Org



https://github.com/cisco-en-programmability

- DevOps Catalyst Center use cases
- Python SDK
- Ansible modules
- Go SDK and Terraform provider
- Splunk Integration Apps
- SDA-as-Code
- Report Operations
- Day N use cases – Python SDK, Ansible, Terraform
- Jenkins open-source Integration and pipelines
- GitHub/GitLab open-source Integrations
- Compliance Use Case
- Custom Integration App
- Webhook Receiver
- Other sample code

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

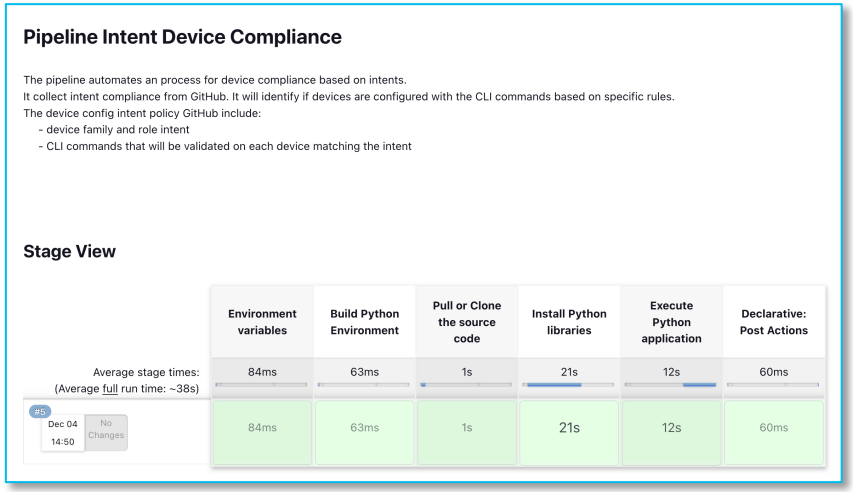
Agenda

- Platforms
- Catalyst Center Compliance
- Custom Network Compliance
- Developer Resources
- Lessons Learned

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Summary

- Catalyst Center has rich network compliance API, providing consistent intent configurations, profiles and policies
- Use existing compliance features and add your compliance workflows for a continuous compliance process
- Compliance insights and reporting, options to automate compliance drift




```
INFO:root: Compliance checks for devices that match the device filter
INFO:root: Saved the device config LO-CN_config.txt
INFO:root: Device: LO-CN :
INFO:root:    - AAA config check failed, missing commands:
INFO:root:        - aaa authorization network local
INFO:root:    - NTP config check failed, missing commands:
INFO:root:        - ntp source Loopback1
INFO:root:        - ntp server 171.68.48.78
INFO:root: Saved the device config PDX-STACK_config.txt
INFO:root: Device: PDX-STACK :
INFO:root:    - AAA config check failed, missing commands:
INFO:root:        - aaa authorization network local
INFO:root:    - NTP config check failed, missing commands:
INFO:root:        - ntp server 171.68.48.78
INFO:root: End of Application "device_config_compliance.py" Run: 2023-12-03 19:51:20
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Other Sessions and Activities This Week

SESSION CODE	SESSION TITLE	ROOM	SESSION DATE	SESSION TIME
DEVNET-2632	Version Control Tools Integrations – Cisco Catalyst Center Infrastructure-as-Code Use Cases	DevNet Theatre	12/08/23	09:15 AM
MTE-1042	Meet the Expert - Gabi Zapodeanu	Meet the Expert 1 – The HUB	12/06/23	03:00 PM

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

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!

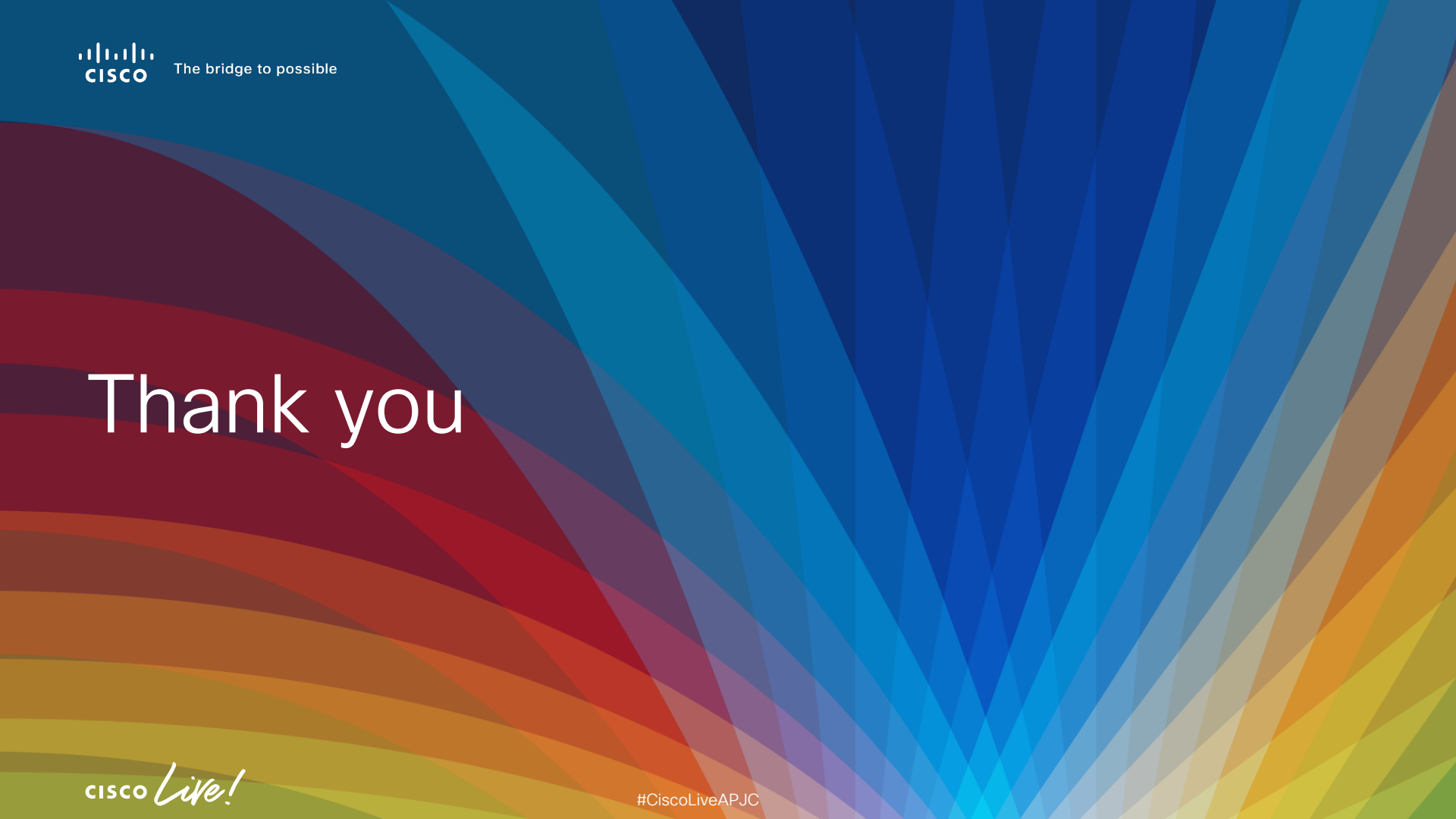
- Complete a minimum of four session surveys and the overall event surveys to claim a Cisco Live T-Shirt

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

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

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public


cisco
The bridge to possible
Thank you
cisco Live!
#CiscoLiveAPJC

cisco Live!

Let's go

#CiscoLiveAPJC