

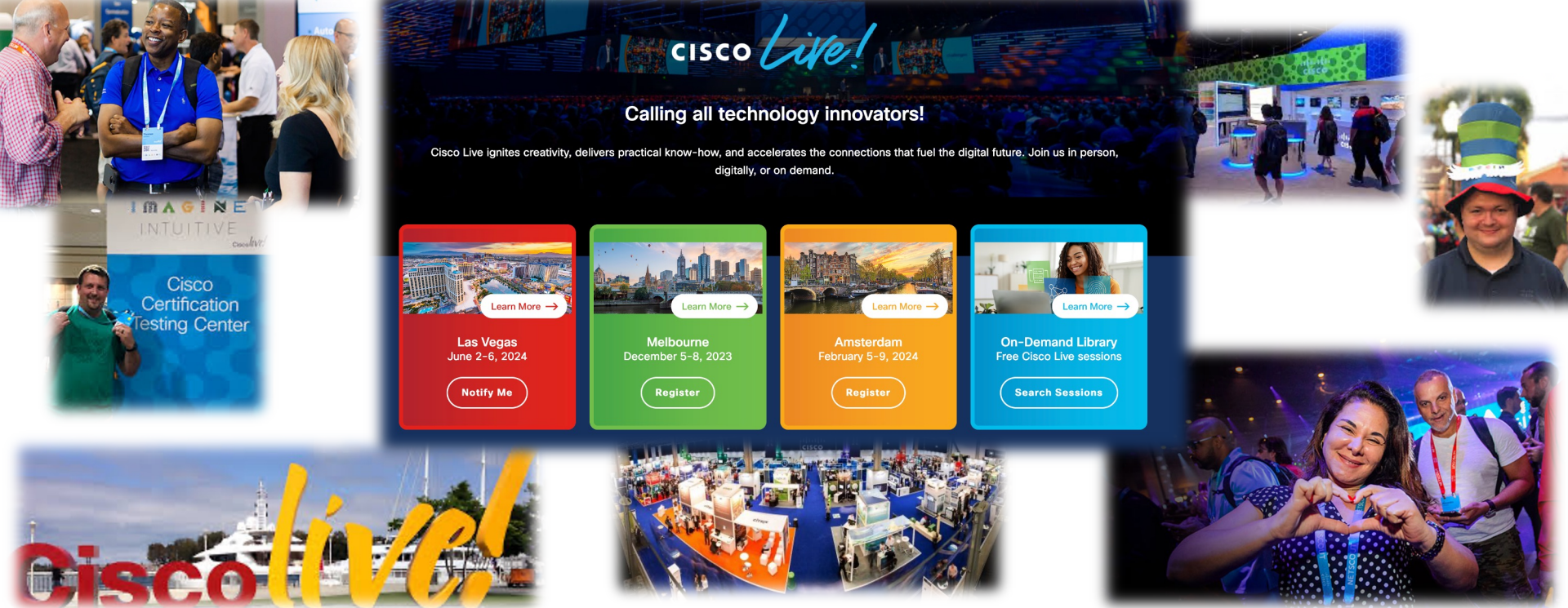


The bridge to possible



Unlocking Telemetry and Instrumentation: Cisco's Journey with InfluxDB

Jason Davis
Distinguished Engineer
April 9, 2024



Since 1989 CiscoLive (Networkers)
The network industries' premiere event
for education, inspiration and fun



Fira de Barcelona



San Diego Convention Center



Las Vegas



Messe Berlin

Held in multiple theatres each year;
US and Europe events are large
requiring our own event IT support staff

CiscoLive - An Example



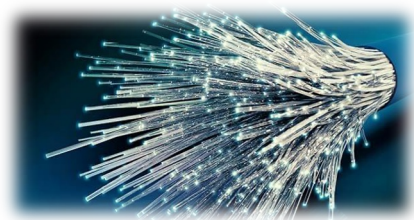
Over 2,300
wireless access points



28,000 attendees
with ~74K mobile devices



650 network switches



Triple 100 Gigabit/second
Internet links

Servers
Storage
Firewalls

Monitoring/
Provisioning
Apps



mobile containerized data center

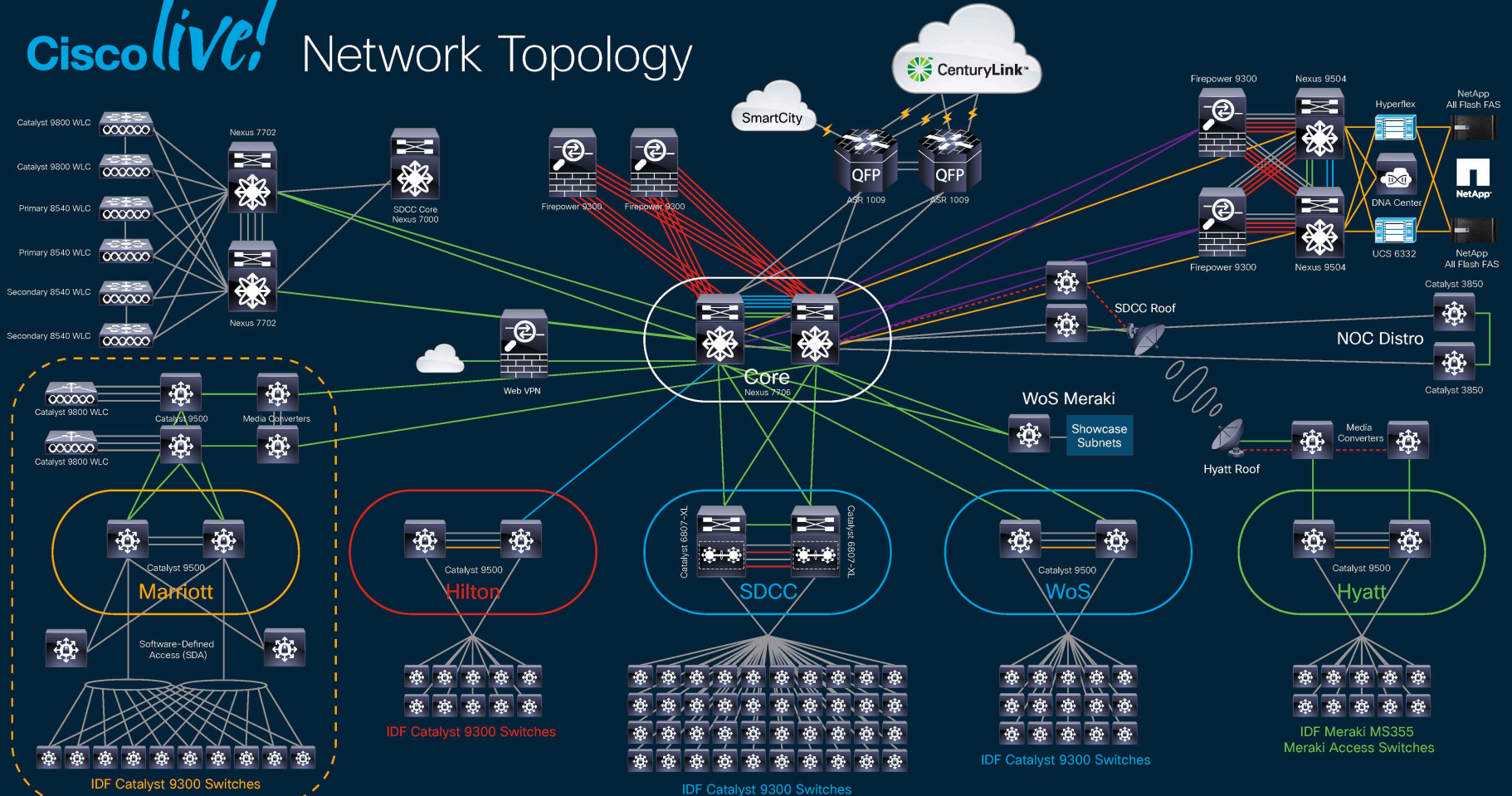
Build-Out: Starting from Scratch



| SUN | MON | TUE | WED | THU | FRI | SAT |
|--------|--------|--------|--------------|--------|-----|-----|
| May 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 28 | 29 | 30 | 31 | June 1 | 2 | 3 |
| | Travel | Set-up | CiscoLive !! | | | |
| 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 |



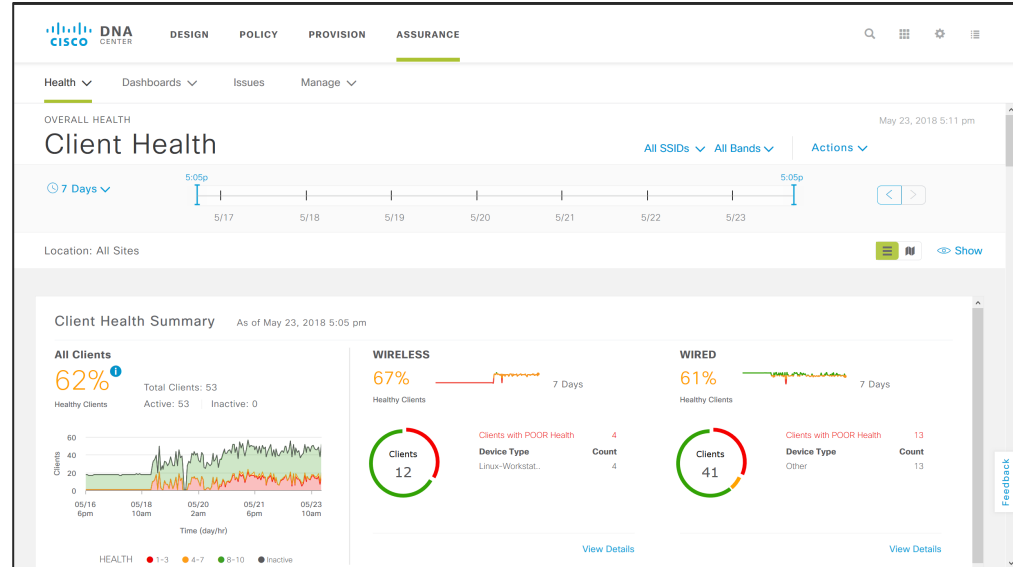
Cisco live! Network Topology



Monitoring Strategy 1 of 3

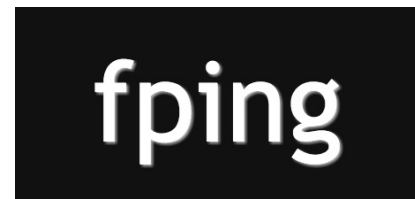
- *Leverage Commercial Management products*

Catalyst Center, Prime Network Registrar, Cisco Telemetry Broker, Umbrella, Meraki Dashboard, WLC, ThousandEyes, etc.



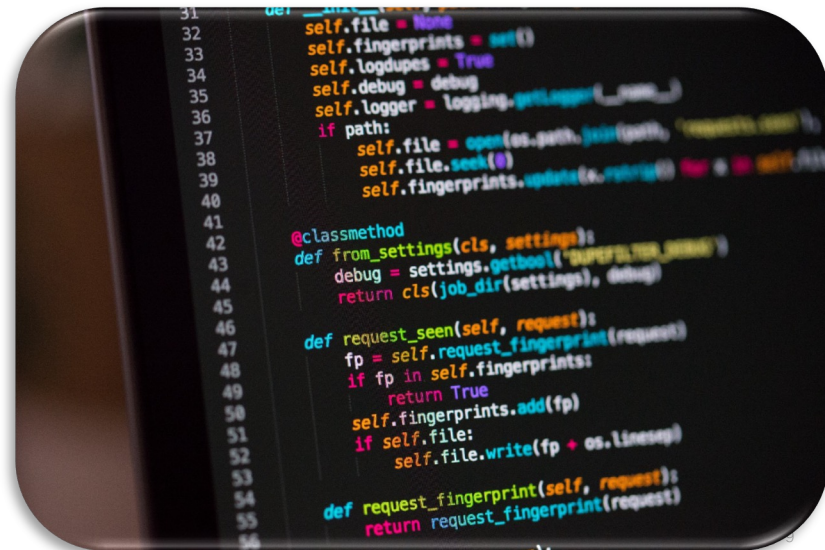
Monitoring Strategy 2 of 3

- *Use Open-source solutions to supplement needs specific to our use-case*



Monitoring Strategy 3 of 3

- Do purposeful development to fit our unique requirements*

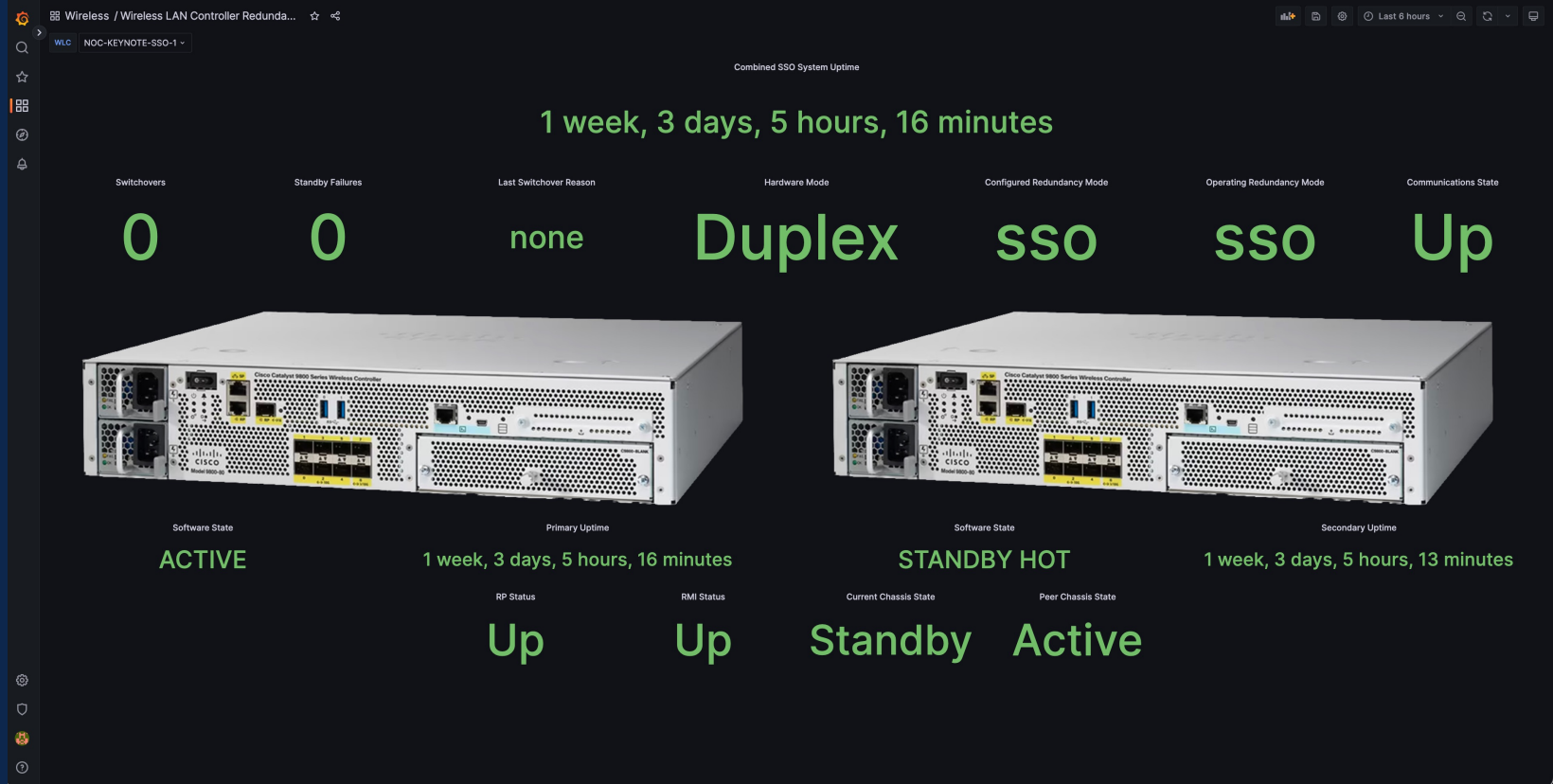


The Journey (with databases)



OLAP/OLTP... TSDB?





Python script 'SSH2Influx' gathering CLI command output of WLC pairwise health, put into InfluxDB with Grafana showing combined health

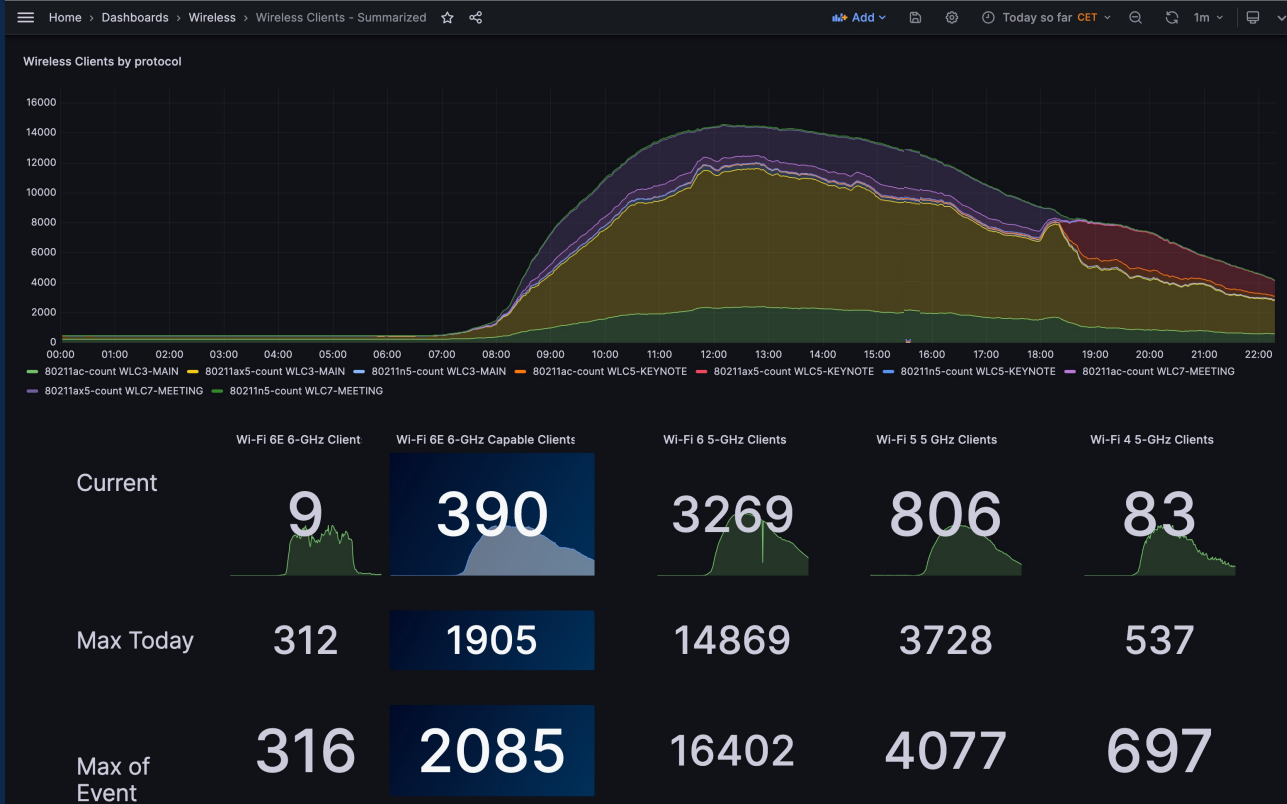
```

24 ▾ inventory:
25     hosts:
26         - host: NOC-MBCC-SSO-1
27         - host: NOC-KEYNOTE-SSO-1
28         - host: NOC-MGM-LUX-SSO-1
29
30 vars:
31     globalcommands:
32         - cmd: show process cpu platform | incl Pid|wncd
33           parsespec: 101
34         - cmd: show wireless loadbalance tag affinity wncd 0
35           parsespec: 200
36         - cmd: show wireless loadbalance tag affinity wncd 1
37           parsespec: 201
38         - cmd: show wireless loadbalance tag affinity wncd 2
39           parsespec: 202
40         - cmd: show wireless loadbalance tag affinity wncd 3
41           parsespec: 203
42         - cmd: show wireless loadbalance tag affinity wncd 4
43           parsespec: 204
44         - cmd: show wireless loadbalance tag affinity wncd 5
45           parsespec: 205
46         - cmd: show wireless loadbalance tag affinity wncd 6
47           parsespec: 206
48         - cmd: show wireless loadbalance tag affinity wncd 7
49           parsespec: 207
50         - cmd: show redundancy
51           parsespec: 301
52
53     parsespecs:
54         commands:
55             - command: show process cpu platform | incl Pid|wncd
56               parsespec: 101
57             regex: >-
58                 \s\d+\s+\d+\s+(\d+)\%\s+(\d+)\%\s+(\d+)\%\s+.\s+\d+\s+(\S+)
59             measurement: wireless-wncd-proc
60             matchtype: multiple
61             match1: fiveSecCPU
62             match1fieldtype: key
63             match1datatype: integer
64             match2: oneMinCPU
65             match2fieldtype: key
66             match2datatype: integer
67             match3: fiveMinCPU
68             match3fieldtype: key
69             match3datatype: integer
70             match4: wncdInstance
71             match4fieldtype: tag
72             match4datatype: string
73         - command: show wireless loadbalance tag affinity wncd 0
74           parsespec: 200
75           regex: >-

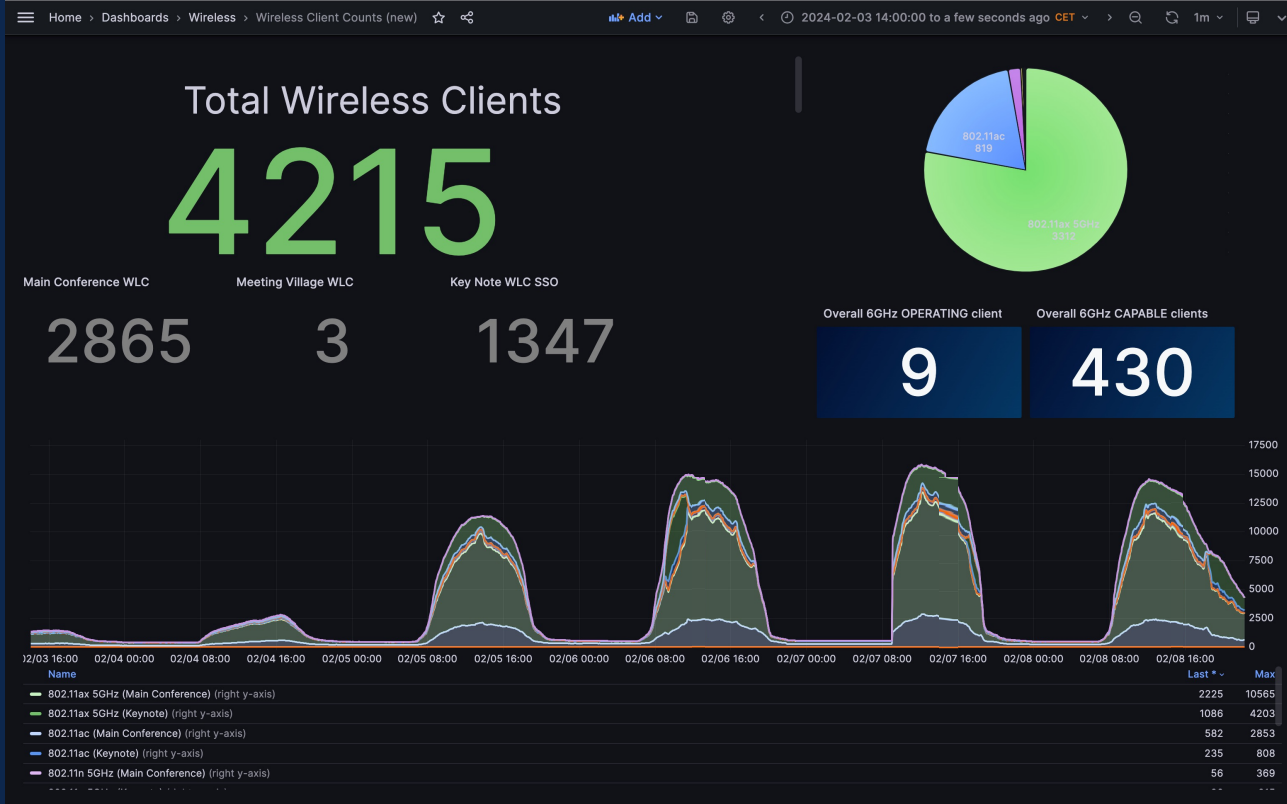
```

<https://github.com/cisco-open/ssh2influx>



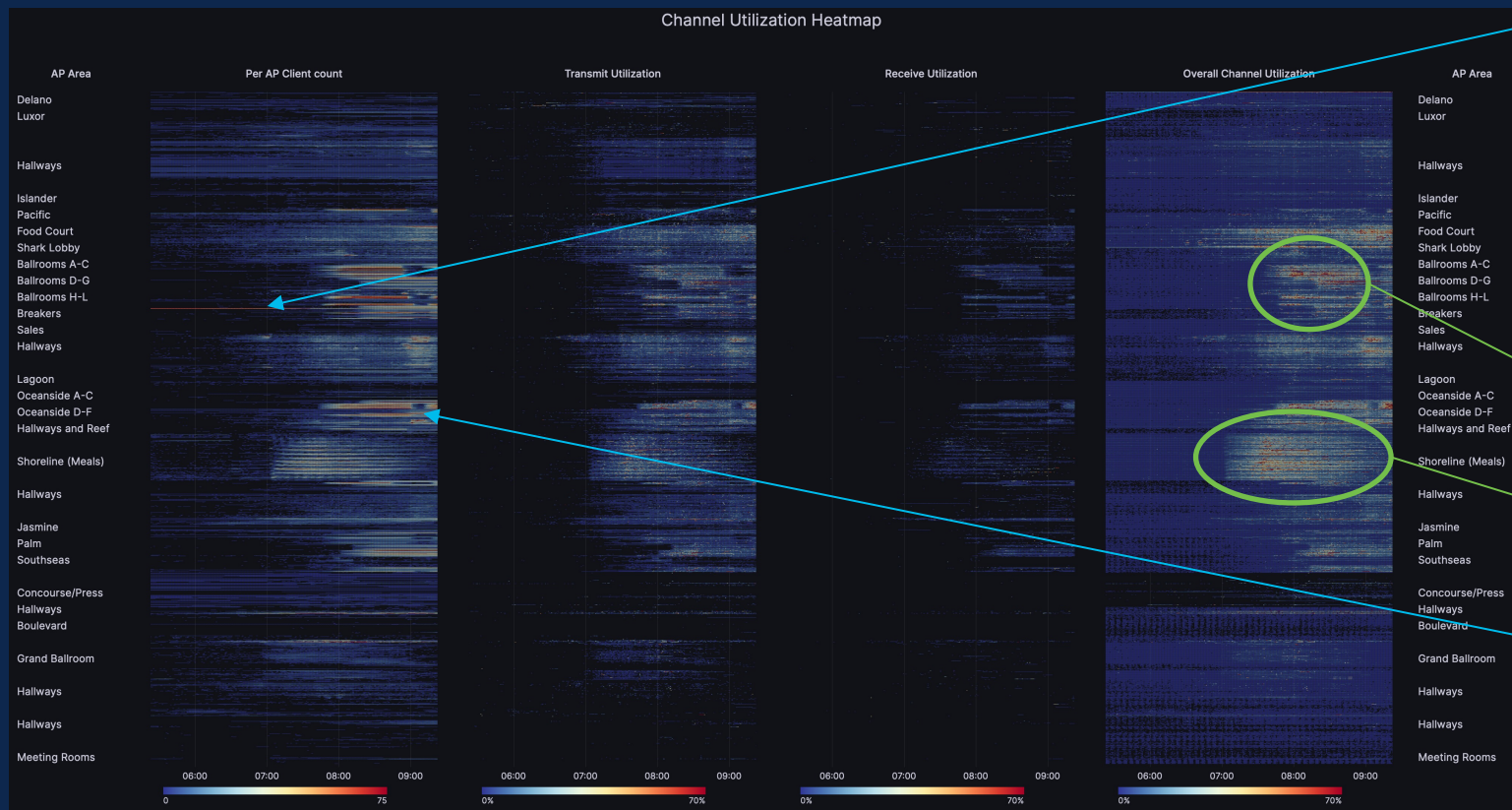


Python script 'NETCONF2Influx' gathering YANG model data of wireless client counts via NETCONF – put into InfluxDB with Grafana showing breakdown of wireless clients by IEEE standard and pre-defined queries of current, daily and event max counts



Python script 'NETCONF2Influx' gathering YANG model data of wireless client counts via NETCONF – put into InfluxDB with Grafana showing breakdown of wireless clients by IEEE standard -- layout specialized for multi-day view and more prominent kiosk mode

Channel Utilization Heatmap – Day 1



This is the AP in the dedicated “tablet charging room”

First sessions of the day

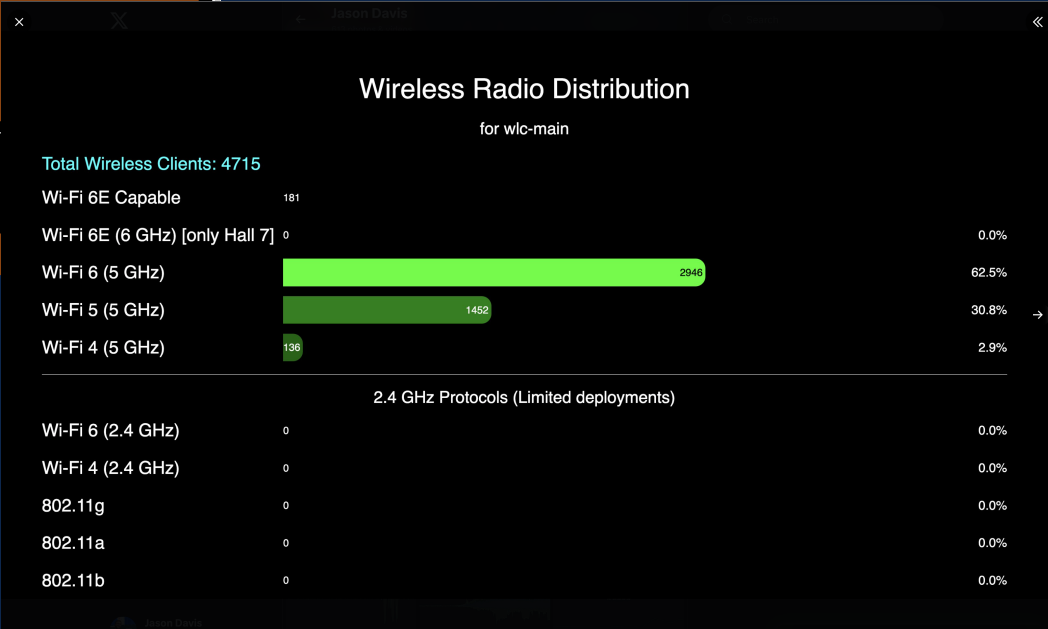
Breakfast

Attendees moving between session rooms

| Client Distribution by SSID and Wireless Standard | | | | | | | | | |
|---|---------|---------------|-----------------|--------|---------------|-----------------|---------|---------|---------|
| SSID | Wi-Fi6E | Wi-Fi6 (5GHz) | Wi-Fi6 (2.4GHz) | Wi-Fi5 | Wi-Fi4 (5GHz) | Wi-Fi4 (2.4GHz) | 802.11g | 802.11a | 802.11b |
| CiscoLive | 0 | 938 | 0 | 268 | 39 | 0 | 0 | 0 | 0 |
| CiscoLive-WPA3 | 8 | 209 | 0 | 33 | 22 | 0 | 0 | 0 | 0 |
| CL-OPS | 7 | 51 | 0 | 9 | 0 | 0 | 0 | 0 | 0 |
| eduroam | 0 | 53 | 0 | 9 | 0 | 0 | 0 | 0 | 0 |
| None | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| Office_RAI | 0 | 0 | 2 | 0 | 0 | 4 | 0 | | |
| OpenRoaming | 1 | 81 | 0 | 6 | 0 | 0 | 0 | | |

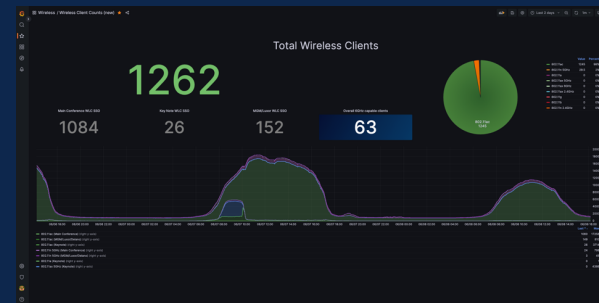
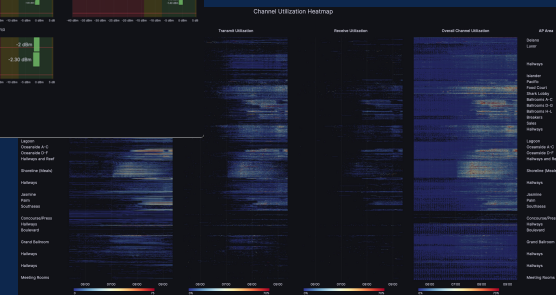
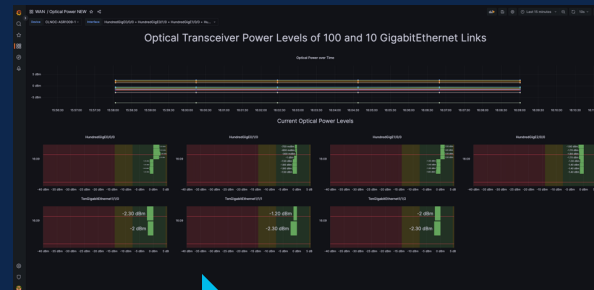
| Overall Counts | | | | | | | |
|----------------|---------|---------------|-----------------|--------|---------------|-----------------|---------|
| SSID | Wi-Fi6E | Wi-Fi6 (5GHz) | Wi-Fi6 (2.4GHz) | Wi-Fi5 | Wi-Fi4 (5GHz) | Wi-Fi4 (2.4GHz) | 802.11g |

Updated Every 2 Minutes



The Network is the Demo





gNMI

gRPC

NETCONF

RESTCONF

REST

SNMP

CLI



Design
considering
varied data
sources

It's all about the
Apey Eyes

APIs...
get it!?



tl;dr

CMSE - Cisco Metrics Search Engine

- **Enable Devs** to find Cisco instrumentation & telemetry options needed to enrich solution development across the diverse spectrum of Cisco products
- **For Cisco**...enhances our broad portfolio of product offers with a 'One Cisco' strategy supporting Full-Stack Observability and Monitoring

A Car analogy...



Article

Talk

Read

Edit

View history

Search Wikipedia

Not logged in - Talk - Contributions - Create account - Log in

OBD-II PIDs

From Wikipedia, the free encyclopedia

OBD-II PIDs (On-board diagnostics **Parameter IDs**) are codes used to request data from a vehicle, used as a diagnostic tool.

SAE standard J1979 defines many OBD-II PIDs. All on-road vehicles and trucks sold in North America are required to support a subset of these codes, primarily for state mandated emissions inspections. Manufacturers also define additional PIDs specific to their vehicles. Though not mandated, many motorcycles also support OBD-II PIDs.

In 1996, light duty vehicles (less than 8,500 lb or 3,900 kg) were the first to be mandated followed by medium duty vehicles (8,500–14,000 lb or 3,900–6,400 kg) in 2005.^[1] They are both required to be accessed through a standardized [data link connector](#) defined by SAE J1962.

Heavy duty vehicles (greater than 14,000 lb or 6,400 kg) made after 2010,^[1] for sale in the US are allowed to support OBD-II diagnostics through SAE standard J1939-13 (a round diagnostic connector) according to CARB in title 13 CCR 1971.1. Some heavy duty trucks in North America use the SAE J1962 OBD-II diagnostic connector that is common with passenger cars, notably Mack and Volvo Trucks, however they use 29 bit CAN identifiers (unlike 11 bit headers used by passenger cars).

Contents

hide

1 Services / Modes

2 Standard PIDs

2.1 Service 01 - Show current data

2.2 Service 02 - Show freeze frame data

2.3 Service 03 - Show stored Diagnostic Trouble Codes (DTCs)

2.4 Service 04 - Clear Diagnostic Trouble Codes and stored values

2.5 Service 05 - Test results, oxygen sensor monitoring (non CAN only)

2.6 Service 09 - Request vehicle information

2.7 Blower encoded PIDs

2.7.1 Service 01 PID 00 - Show PIDs supported

2.7.2 Service 01 PID 01 - Monitor status since DTCs cleared

2.7.3 Service 01 PID 41 - Monitor status this drive cycle

2.7.4 Service 01 PID 78 and 79 - Exhaust Gas temperature (EGT) Bank 1 and Bank 2

2.7.5 Service 03 (no PID required) - Show stored Diagnostic Trouble Codes

2.7.6 Service 09 PID 08 - In-use performance tracking for spark ignition engines

2.7.7 Service 09 PID 09 - In-use performance tracking for compression ignition engines

2.8 Enumerated PIDs

2.8.1 Service 01 PID 00 - Fuel system status

2.8.2 Service 01 PID 12 - Commanded secondary air status

2.8.3 Service 01 PID 1C - OBD standards this vehicle conforms to

2.8.4 Service 01 PID 51 - Fuel Type Coding

3 Non-standard PIDs

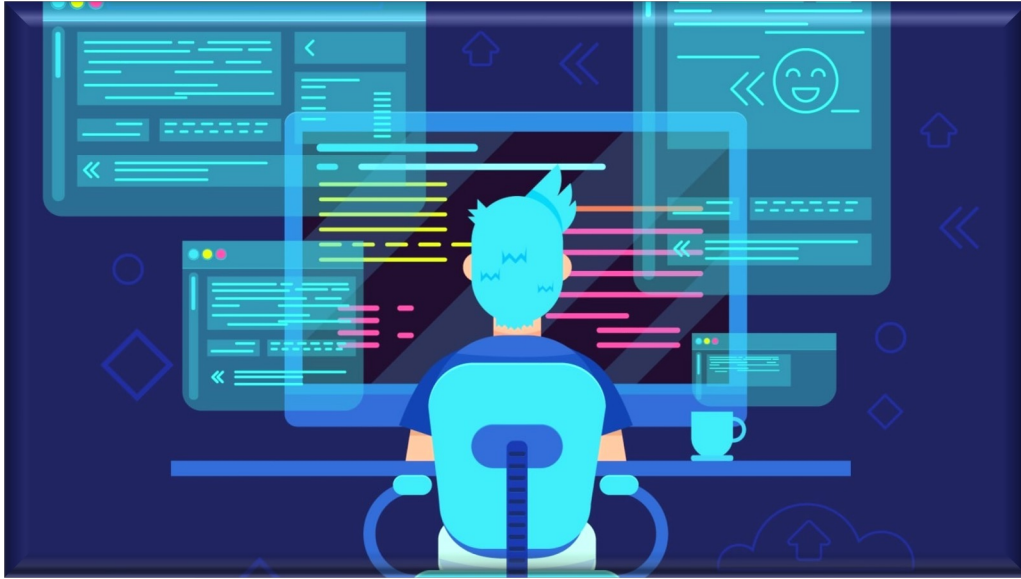
4 CAN (11-bit) bus format

4.1 Query

4.2 Response

5 See also

How Does a Dev Ideate?



Build



Buy



Re-use



Issue

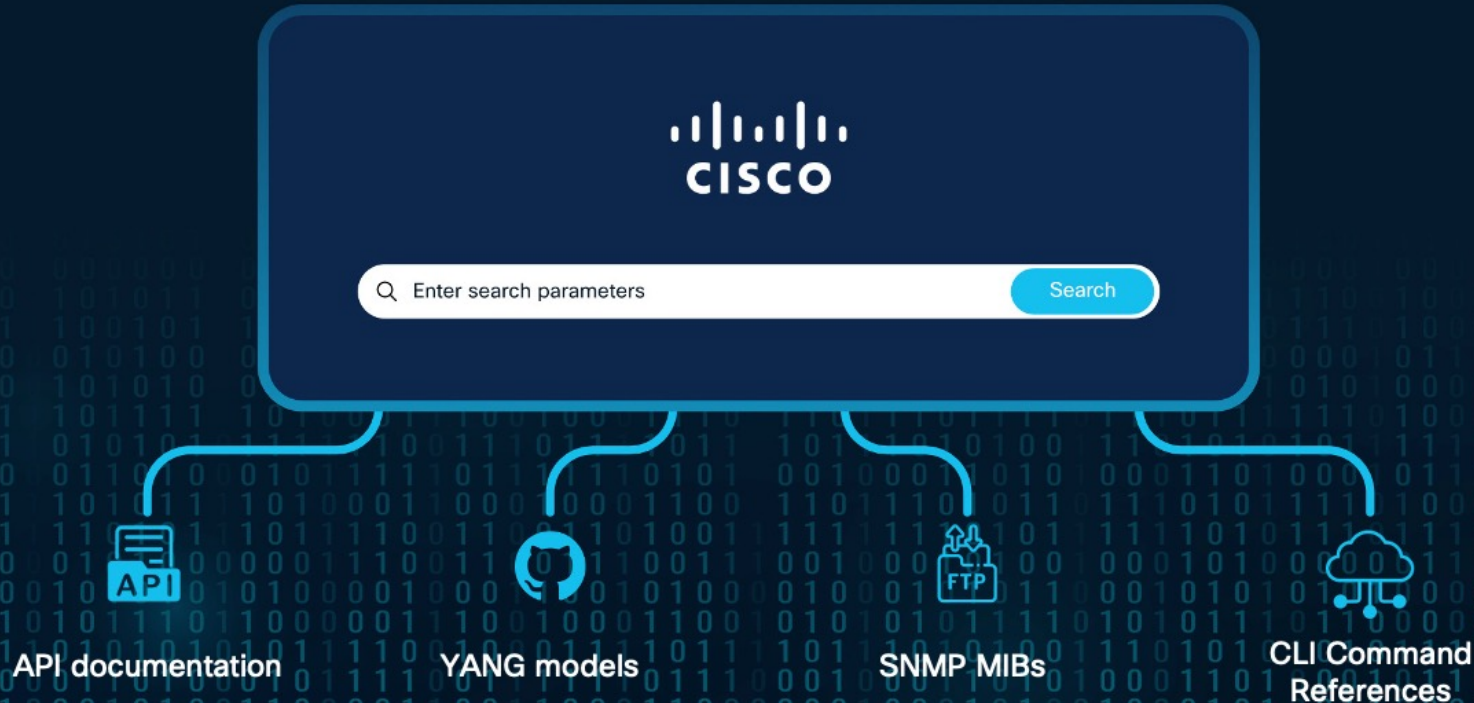
Customer-Devs are largely *unaware* of telemetry and instrumentation options

APIs, YANG Model, SNMP
MIB and CLI command
references are *spread across*
1,000s of different web-docs


Because of complexity
Devs don't engage

A Search Engine for Telemetry, Instrumentation, Metrics

Cisco Metrics Search Engine



CMSE

 show users

Search

All

APIs


YANG Model

SNMP MIB

CLI

Refine your search

10,000+ results (0.12 seconds)

 Rate your search experience!

OS



Version



Products



Categories



Reset

show user

OS: Cisco IOS XR

Version:

To display all **user** groups and task IDs associated with the currently logged-in **user**, use the **show user**

Show code snippets



show user

OS: Cisco IOS XR

Version:

To display all **user** groups and task IDs associated with the currently logged-in **user**, use the **show user**

Show code snippets



show users

OS: Cisco NX-OS

Version: 7.0(3)J3(1)

Other 10 versions ▾

show users

Show code snippets



show users

OS: Cisco NX-OS

Version:

To display the **user** session information for a virtual device context (VDC), use the **show users** command. **show users** This...

Show code snippets



show users

OS: Cisco IOS XE

Version:

To display information about the active lines on the router, use the **show users** command in **user EXEC**

Show code snippets



show users

OS: Cisco IOS XE

Version:

To display information about the active lines on the router, use the **show users** command in **user EXEC**

Show code snippets



show users

OS: Cisco NX-OS

Version: 6.1(2)J2(2)

Other 10 versions ▾

show users

[read-only

show users

OS: Cisco NX-OS

Version: 10.1(x)

Other 10 versions ▾

show users

[read-only

show users

OS: Cisco NX-OS


Version: 10.2(x)

Other 10 versions ▾

show users

[read-only

CMSE

 show users

Search

All

APIs

YANG Model

SNMP MIB

CLI

Refine your search

OS



Version



Products




Categories



Reset

10,000+ results (0.12 seconds)

 Rate your search experience!

show user

OS: Cisco IOS XR

Version:

To display all **user** groups and task IDs associated with the currently logged-in **user**, use the **show user**

Show code snippets



show user

OS: Cisco IOS XR

Version:

To display all **user** groups and task IDs associated with the currently logged-in **user**, use the **show user**

Show code snippets



show users

OS: Cisco NX-OS

Version:

To display the **user** session information for a virtual device context (VDC), use the **show users** command. **show users** This...

Show code snippets



show users

OS: Cisco IOS XE

Version:

To display information about the active lines on the router, use the **show users** command in **user EXEC**

Show code snippets



show users

OS: Cisco NX-OS

Version: 6.1(2)J2(2)

Other 10 versions

show users [read-only

show users

OS: Cisco NX-OS

Version: 10.1(x)

Other 10 versions

show users [read-only (

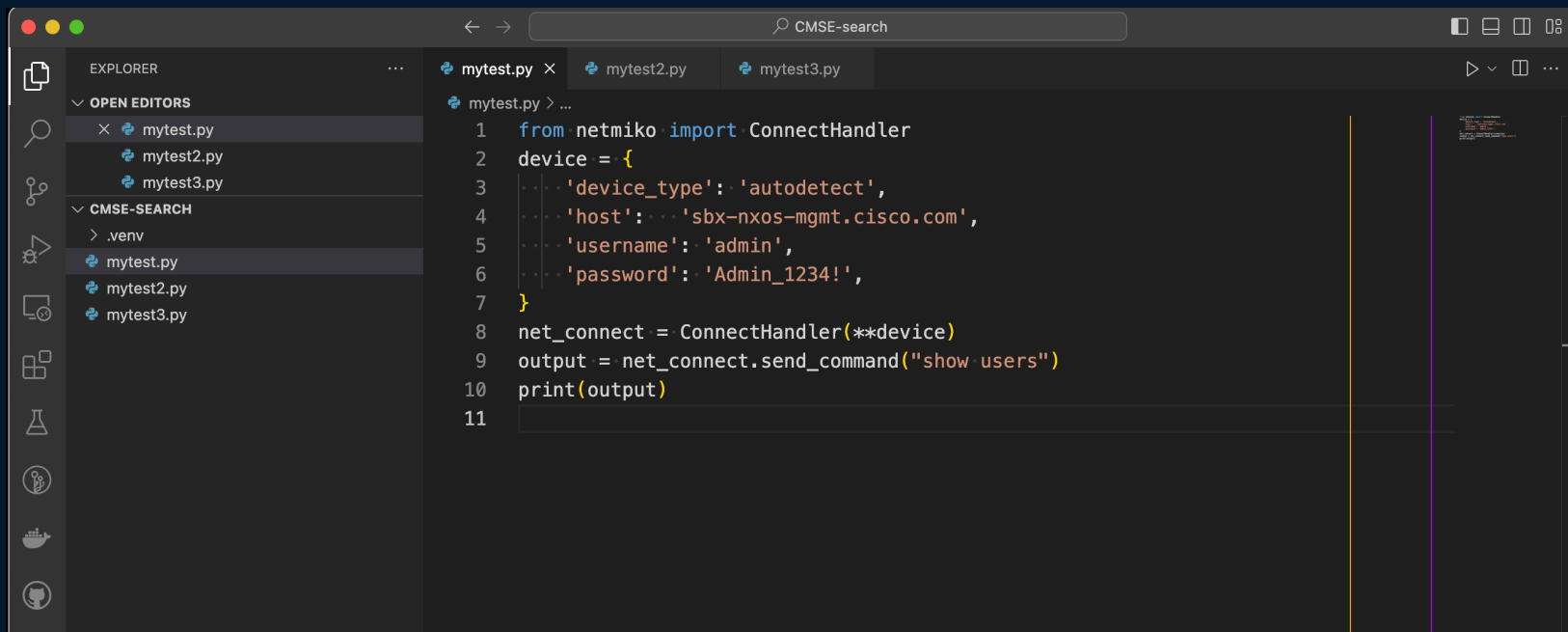
Code Snippet Command Reference



show users

Version: 7.0(3)J3(1)

```
1 from netmiko import ConnectHandler
2 device = {
3     'device_type': 'autodetect',
4     'host': 'CHANGEME',
5     'username': 'CHANGEME',
6     'password': 'CHANGEME',
7 }
8 net_connect = ConnectHandler(**device)
9 output = net_connect.send_command("show users")
10 print(output)
```

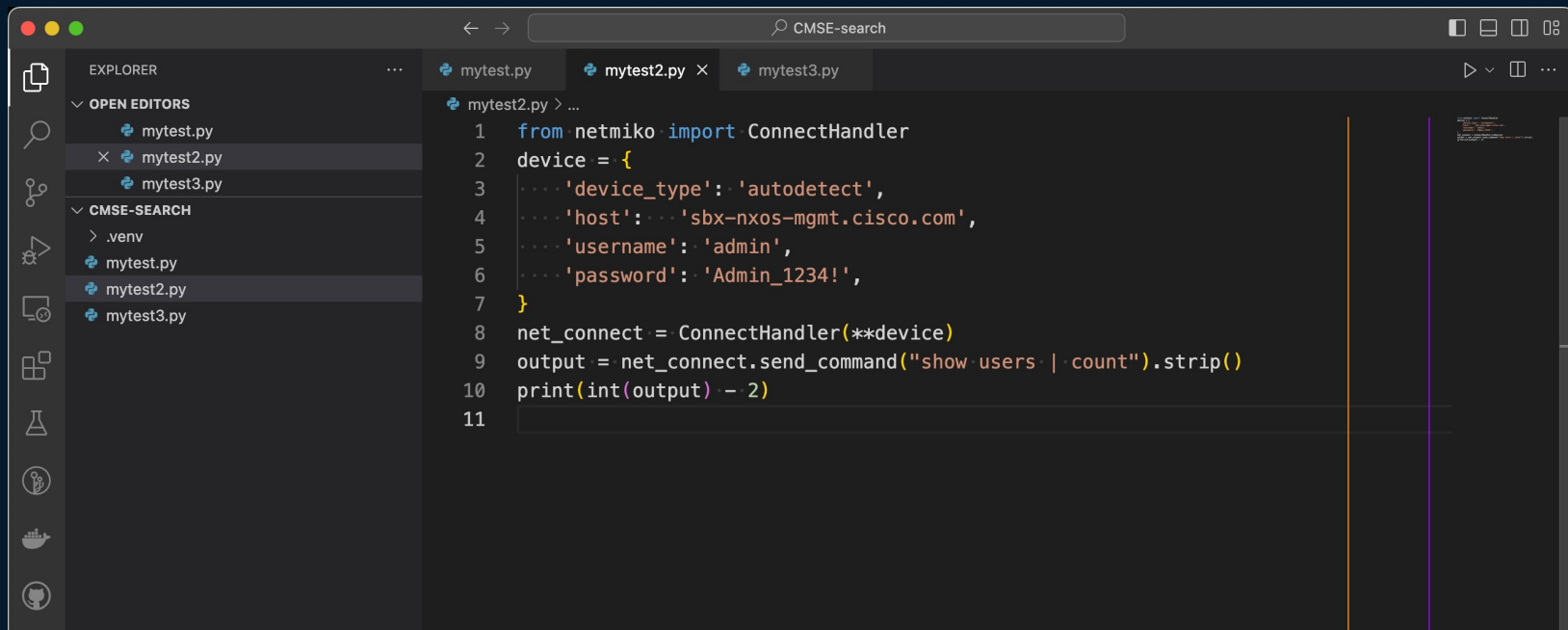


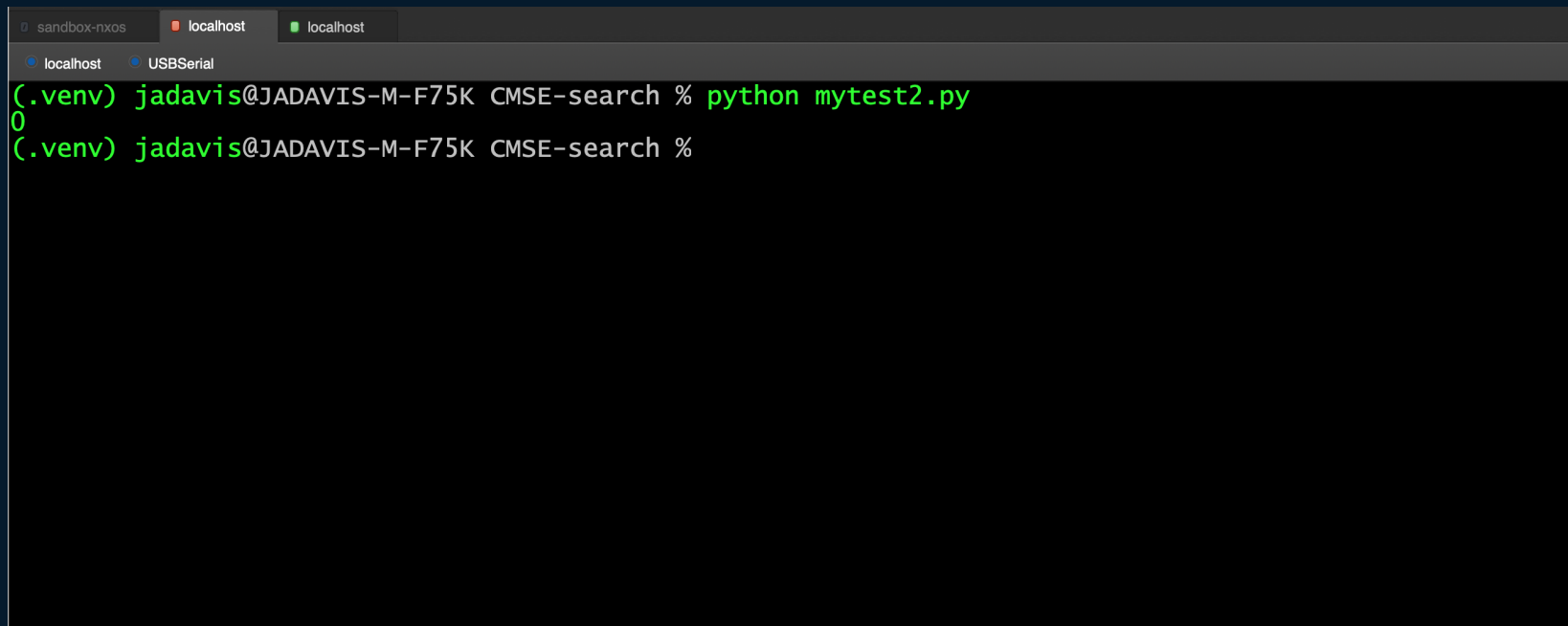
sandbox-nxoslocalhostlocalhost

localhostUSBSerial

```
(.venv) jadavis@JADAVIS-M-F75K CMSE-search % python mytest.py
NAME      LINE      TIME      IDLE      PID COMMENT
admin     pts/5      Apr  2 11:44      .      17519 (10.10.20.28) session=ssh *
```

```
(.venv) jadavis@JADAVIS-M-F75K CMSE-search %
```



A terminal window with a dark background. The title bar at the top shows three tabs: 'sandbox-nxos' (with a magnifying glass icon), 'localhost' (with a red square icon), and another 'localhost' (with a green square icon). Below the tabs, there are two active sessions: 'localhost' (with a blue dot icon) and 'USBSerial' (with a blue dot icon). The main area of the terminal displays the following text in green:
(.venv) jadavis@JADAVIS-M-F75K CMSE-search % python mytest2.py
0
(.venv) jadavis@JADAVIS-M-F75K CMSE-search %

EXPLORER

OPEN EDITORS

- mytest.py
- nxostest.py
- mytest2.py
- mytest3.py

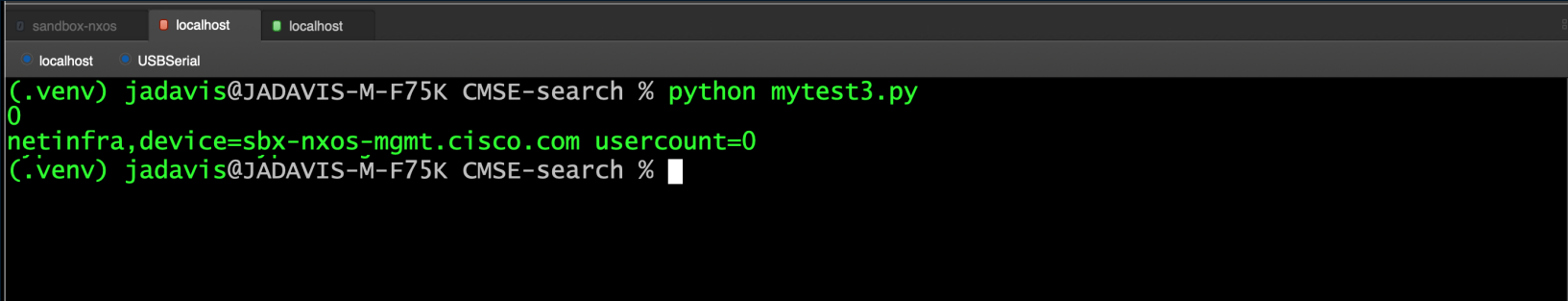
CMSE-SEARCH

- .venv
- mytest.py
- mytest2.py
- mytest3.py
- nxostest.py

mytest3.py > ...

```
1 from netmiko import ConnectHandler
2 from influxdb_client_3 import InfluxDBClient3, Point
3
4 device = {
5     'device_type': 'autodetect',
6     'host': 'sbx-nxos-mgmt.cisco.com',
7     'username': 'admin',
8     'password': 'Admin_1234!',
9 }
10 net_connect = ConnectHandler(**device)
11 output = net_connect.send_command("show users | count").strip()
12 print(int(output) - 2)
13
14 with InfluxDBClient3(token="N31gsUv...", host="https://...",
15                      org="c56de710ea5186...", database="jadavis's Bucket") as client:
16     point = f"netinfra,device={device['host']} usercount={int(output) - 2}"
17     #point = Point("netinfra").tag("device", f"{device['host']}").field("usercount", int(output) - 2)
18     print(point)
19
20 client.write(record=point)
21
22
23
```

Enhance with basic write to Influx code



A terminal window with a dark background and green text. The window has three tabs at the top: 'sandbox-nxos', 'localhost' (selected), and another 'localhost'. Below the tabs, there are two sub-tabs: 'localhost' (selected) and 'USBSerial'. The terminal shows the following text:

```
(.venv) jadavis@JADAVIS-M-F75K CMSE-search % python mytest3.py
0
netinfra,device=sbx-nxos-mgmt.cisco.com usercount=0
(.venv) jadavis@JADAVIS-M-F75K CMSE-search %
```


Load Data

Data Explorer

Dashboards

Tasks

Alerts

Settings

Help & Support

Data Explorer

Switch to old Data Explorer

Graph

CUSTOMIZE

Local

SAVE AS



Query 1 (0.09s)

+

View Raw Data

CSV

Refresh

Past 1h

SCRIPT EDITOR

SUBMIT

FROM

Search buckets

jadavis's Bucket

_monitoring

_tasks

+ Create Bucket

MEASUREMENT

Search measurements

netinfra

All Measurements

Filter

device

Search device tag values

sbx-nxos-mgmt.cisco.c...

Filter

_field

Search fields

usercount

WINDOW PERIOD

CUSTOM

AUTO

auto (10s)

Fill missing values

AGGREGATE FUNCTION

CUSTOM

AUTO

mean

median

last

Call To Action

- Try it out – it's live NOW!

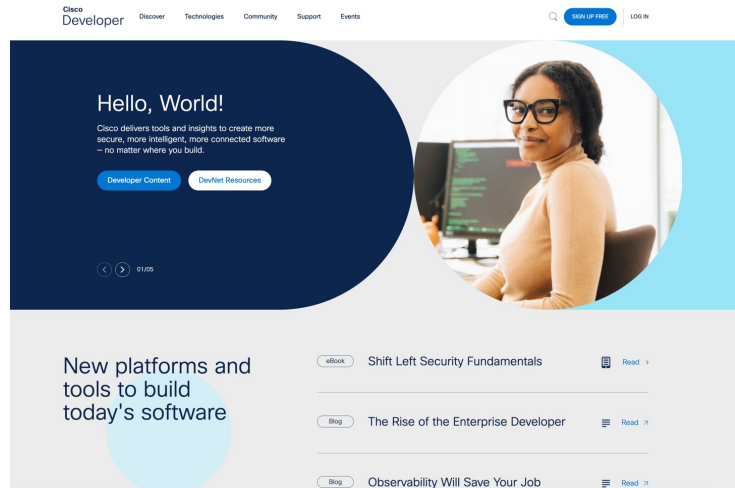
<https://developer.cisco.com/cmse>



- If you have feedback, <mailto://cmse-interest@external.cisco.com>



<https://developer.cisco.com>



Training & Certification

API Guides

Learning Labs

Code Exchange

Reservable Resources

Developer Forums

Tools

Resources for...

DevOps/SRE

Enterprise
AppDevs



Infra
Engineers



The bridge to possible