cisco Live!

Let's go

The bridge to possible

Using API's for Multidomain Inventory and Asset Management

William Nellis
@williamnellis
DEVNET-2103

Cisco Webex App

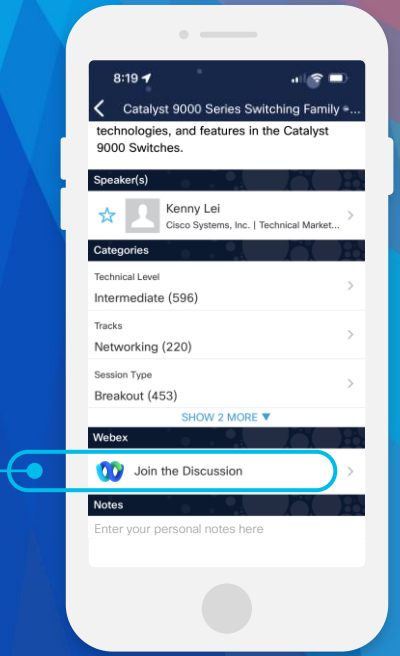
Questions?

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 June 9, 2023.



https://ciscolive.ciscoevents.com/ciscolivebot/#DEVENT-2103

Agenda

- What is the problem space?
- How the Solution Works
- Using the Solution
- Conclusion

About your Presenter

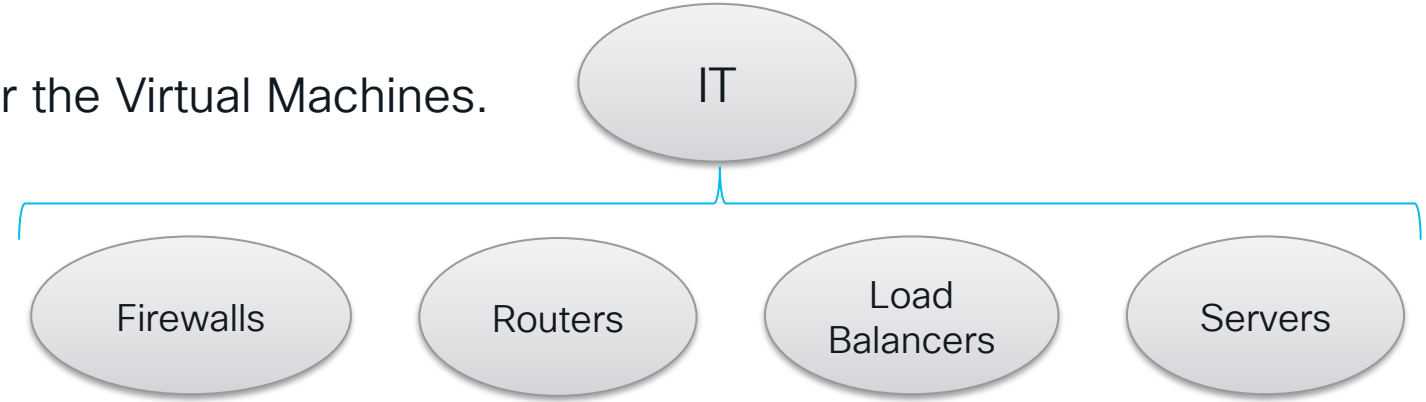
- 28 years in industry, 16 at Cisco
- Networking background,
- CCIEx2, CCDE, Azure Architect Expert
- Devnet Professional

The problem space

IT organizations are not required to support and audit just the routers.

Or the firewalls.

Or the Virtual Machines.



We have to do *ALL* of it.

Solution Design

- Simple and Easy to use
  - “so easy an Account Manager can run it, from the golf course”
- Real. Works on Real stuff. With real Data.
- Multidomain: Works on all Cisco Controller Products
- Snap in Framework for third party products

This is for networkers and developers with a…

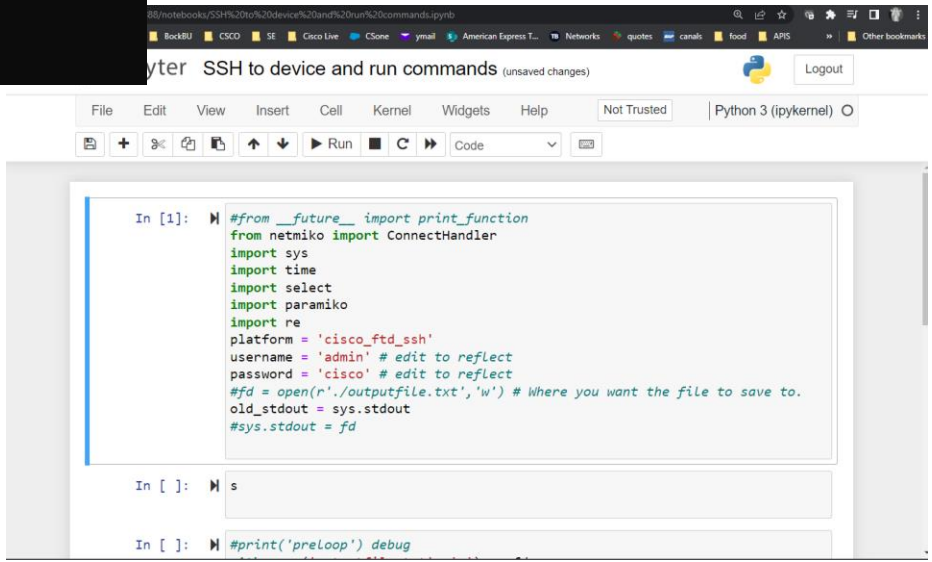
If you have a….	This demo can help by…
Networking Background	Easy to use Scripts that work with Modern API with no programming experience. A Leg up on Programmability.
Development Background	Showing Functional API and Authentication code to get to the data you need.

How this works

The Environment

```
(base) C:\Scripts\DeviceSSHRunCommands>jupyter notebook
[I 2023-03-22 15:45:27.912 LabApp] JupyterLab extension loaded from C:\Users\wnellis\Anaconda3\lib\site-packages\jupyter
lab
[I 2023-03-22 15:45:27.913 LabApp] JupyterLab application directory is C:\Users\wnellis\Anaconda3\share\jupyter\lab
[I 15:45:27.917 NotebookApp] Serving notebooks from local directory: C:\Scripts\DeviceSSHRunCommands
[I 15:45:27.918 NotebookApp] Jupyter Notebook 6.4.8 is running at:
[I 15:45:27.918 NotebookApp] http://localhost:8888/?token=4be053805ccf8e525bebfa50a35559b903f0af615c6326e5
[I 15:45:27.918 NotebookApp]  or http://127.0.0.1:8888/?token=4be053805ccf8e525bebfa50a35559b903f0af615c6326e5
[I 15:45:27.918 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 15:45:27.972 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///C:/Users/wnellis/AppData/Roaming/jupyter/runtime/nbserver-3280-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=4be053805ccf8e525bebfa50a35559b903f0af615c6326e5
     or http://127.0.0.1:8888/?token=4be053805ccf8e525bebfa50a35559b903f0af615c6326e5
[W 15:45:33.836 NotebookApp] Notebook SSH to device and run commands.ipynb is not trusted
[I 15:45:34.021 NotebookApp] Kernel started: 4cc2f1e9-d401-43cb-ac90-6e38838c7461, name: python3
[I 15:47:33.927 NotebookApp] Saving file at /SSH to device and run commands.ipynb
```

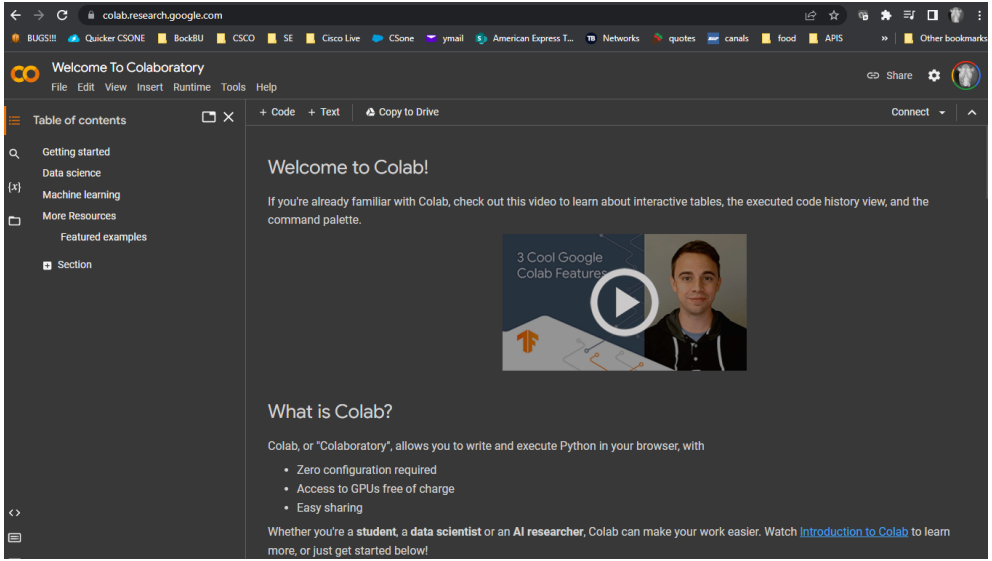


- Based off of Jupyter Notebook, but in cloud
- Go to directory with scripts
- Run : jupyter notebook

The Environment

- Google Colab
- Cloud Based Jupyter Notebook
- Colab.research.google.com

- Highly Encourage you to use personal account unless your work account has drive access.



How this works...

Uses Pandas Python library...

Get the data from “n” API. Put in Table.

Standardize tables...

Merge into one Single Asset Management Table.

Reconcile with ServiceNow.

Using this

“You gotta tame the beast before you let it out of the cage”

- To edit, you need to save a copy to your colab environment in your Google Drive.
- Your mileage may vary w/ a corporate account,
- Public sandboxes…
  - Rate limits.
  - Things change and there is a provision…
- Service Now Demo will NOT work for you until you get your service now developer environment.

Live Demo

http://cs.co/DEVNET-2103

Fill out your session surveys!

Attendees who fill out a minimum of four session surveys and the overall event survey will get **Cisco Live-branded socks** (while supplies last)!

Attendees will also earn 100 points in the **Cisco Live Challenge** for every survey completed.

These points help you get on the leaderboard and increase your chances of winning daily and grand prizes

Continue your education

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer 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

Thank you

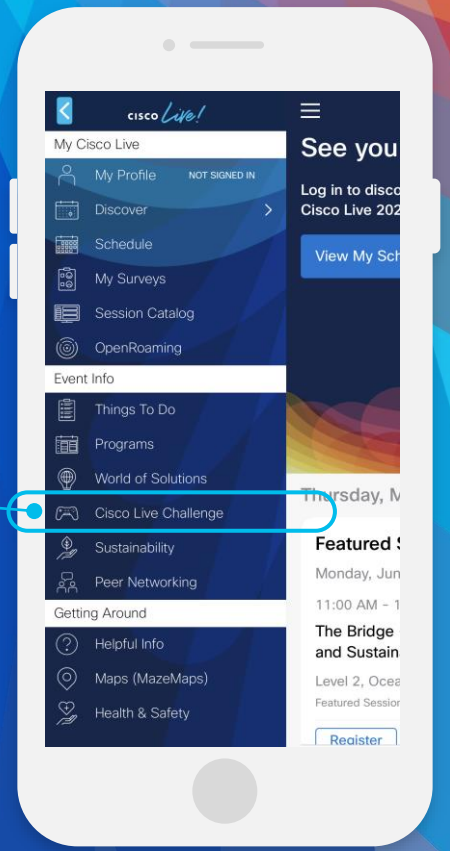
Cisco Live **Challenge**

Gamify your Cisco Live experience!

Get points for attending this session!

How:

1. Open the Cisco Events App.
2. Click on 'Cisco Live Challenge' in the side menu.
3. Click on View Your Badges at the top.
4. Click the + at the bottom of the screen and scan the QR code:



cisco Live!

Let's go

“Backup,

We need backup!”

Demo Logistics

xDomain Inventory DEVNET-2103

File Edit View Insert Runtime Tools Help All changes saved

Code + Text

Cross Domain Inventory

Before you even start... To Modify this, You do need a google account. AND IT PROBABLY NEEDS TO BE A PERSONAL ACCOUNT unless your Corporate account has google drive access. This is a read only sheet. To make the changes you need to, you need to go to File/Save a copy to drive. You need to use a personal google account. if its a corporate account you may not be able to save to Google drive questions to wnellis@cisco

this works best in chrome. in theory it works in other browsers. in practice, it may or may not.

This will give you a Read Write copy, that you can modify.

Questions or Comments to William Nellis wnellis@cisco.com

High Level Segments

- Functional Examples of Controller technologies,

xDomain Inventory DEVNET-2103

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Meraki

9 cells hidden

Cisco SD-Wan

9 cells hidden

Cisco DNA Center and Multi controller example

For instances in which you have Multiple Controllers, examples of which could be 1) More than 1 DNAC controller 2) More than 1 APIC Controller 3) Multiple Meraki instances, organizations,

you may want to be able to grab inventory from all instances and have them a part of the master inventory. An example of doing this in DNAC is done below, we define controllers, and iterate over the list, appending the data to a table.

[ ] 6 cells hidden

Intersight

[ ] 14 cells hidden

ACI

[ ] 7 cells hidden

Bringing it All Together

Detailed Code Example

```
# An Example of what we are about to do w/ multi controller

Controllers=[{'host':'host1', 'username':'username1','password':'password1'},{'host':'host2', 'username':'username2','password':'password2'},{'host':'host3', 'username':'username3','password':'password3'}]
for Controller in Controllers:
  print(f"host is {Controller['host']}, username is {Controller['username']}, password is {Controller['password']}")
#  print(Controller['host'])
```

```
host is host1, username is username1, password is password1
host is host2, username is username2, password is password2
host is host3, username is username3, password is password3
```

```
[ ] import requests
    from requests.auth import HTTPBasicAuth
    import json
    import pandas as pd

    #if you only have one dnac, you can just remove the second item in the list below...
    Controllers=[{'host':'sandboxdnac2.cisco.com','username':'devnetuser','password':'Cisco123!'},
                 {'host':'sandboxdnac.cisco.com','username':'devnetuser','password':'Cisco123!'}]

    bigTable=pd.DataFrame()

    for Controller in Controllers:
      DNAC=Controller['host']
      DNAC_USER=Controller['username']
      DNAC_PASSWORD=Controller['password']
      INTENT_API = f"https://{DNAC}/dna/intent/api/v1/"
      url = f'https://{DNAC}/dna/system/api/v1/auth/token'
      resp = requests.post(url, auth=HTTPBasicAuth(DNAC_USER, DNAC_PASSWORD), verify=False)
      print(resp.text)      #<<<<<<<<<<< You should see your token)
      token = resp.json()['Token']
      url = INTENT_API+"network-device"
      print(url)
      hdr={'x-auth-token': token}
      resp = requests.get(url, headers=hdr,verify=False)
      #print(resp.text)
      device_list=resp.json()
      dnac_inventory_df=pd.json_normalize(device_list['response'])
      dnac_inventory_df['DataSource']=INTENT_API
      bigTable=pd.concat([dnac_inventory_df,bigTable])
```

Make the Tables Consistent

EVERY DATA MERGE IS THE SAME LOGIC:

1. Create conscise report of domain that maps to "big inventory" schema... IE, just the serial, hostname, model...
2. Rename this concise reports columns to match... since host-name should equal hostname should equal Hostname, make it the same.
3. merge the domain specific concise data into "theBigInventory"

```
[ ] theBigInventory=pd.DataFrame(columns=('DataSource','Serial','Hostname','Model','IP Address','Version')) ### Define the table
```

Meraki data merge

```
[ ] #display(pd.concat([df1, df2], ignore_index = True))   #>>> each data frame has an index, (0,1,2,3, etc). ignore makes it reindex, so you dont end up with dup's two devices w/ same index in merged df
    #df.rename(columns = {'old_col1':'new_col1', 'old_col2':'new_col2'}, inplace = True) < consistent column name

    meraki_inventory_simple_df=meraki_inventory_df[['DataSource','serial','name','model','lanIp','firmware']]   #just the data we want
    meraki_inventory_simple_df.rename(columns={'serial':'Serial','name':'Hostname','model':'Model','lanIp':'IP Address','firmware':'Version'}, inplace=True) #rename columns to match
    theBigInventory=pd.concat([theBigInventory,meraki_inventory_simple_df], ignore_index = True)  #merge
```

Master List

- The Big Inventory list is below

```
theBigInventory
```

	DataSource	Serial	Hostname	Model	IP Address	Version
0	https://api.meraki.com/api/v1	Q2CV-V49B-RCMZ		MV71	192.168.0.25	Not running configured version
1	https://api.meraki.com/api/v1	Q2EK-2LYB-PCZP	Alex's MR84 - 1	MR84	None	Not running configured version
2	https://api.meraki.com/api/v1	Q2EK-3UBE-RRUY	Vegas Living Room MR84	MR84	192.168.0.20	Not running configured version
3	https://api.meraki.com/api/v1	Q2EK-ACGE-URXL		MR84	None	Not running configured version
4	https://api.meraki.com/api/v1	Q2EK-D4XP-235S	Vegas Balcony MR84	MR84	None	Not running configured version
5	https://api.meraki.com/api/v1	Q2EK-UKGM-XSD9	Sun Room	MR84	192.168.1.203	wireless-29-5-1
6	https://api.meraki.com/api/v1	Q2GV-7HEL-HC6C		MV12W	192.168.1.241	camera-4-18-1
7	https://api.meraki.com/api/v1	Q2HP-C2YW-KB2E	Office Switch	MS220-8P	192.168.1.227	switch-14-33
8	https://api.meraki.com/api/v1	Q2HP-EC87-M9B8	ms01-dl1	MS220-8P	192.168.128.2	switch-14-33
9	https://api.meraki.com/api/v1	Q2HP-W3HC-2C8D	ms01-dl3	MS220-8P	192.168.128.6	switch-14-33
10	https://api.meraki.com/api/v1	Q2HP-Y9R9-FK5Y	Basement Switch	MS220-8P	192.168.1.249	switch-14-33
11	https://api.meraki.com/api/v1	Q2JD-7RNY-EB7Z	Alex's MR32	MR32	192.168.1.16	wireless-25-13
12	https://api.meraki.com/api/v1	Q2KD-KWMU-7U92	ap01-dl2	MR42	192.168.128.3	wireless-29-5-1
13	https://api.meraki.com/api/v1	Q2LD-3Y7V-7Y2X	Basement AP	MR52	192.168.1.94	wireless-29-5-1
14	https://api.meraki.com/api/v1	Q2LD-D932-NRPU	ap01-dl3	MR52	192.168.1.142	wireless-29-5-1
15	https://api.meraki.com/api/v1	Q2LD-FGN3-VP7S	ap01-dl1	MR52	192.168.128.7	wireless-29-5-1
16	https://api.meraki.com/api/v1	Q2LD-GYL3-KEHX	1st Floor AP	MR52	192.168.1.215	wireless-29-5-1
17	https://api.meraki.com/api/v1	Q2LD-N2U5-D83H		MR52	24.144.215.84	Not running configured version
18	https://api.meraki.com/api/v1	Q2LD-X2S2-AG2U		MR52	None	Not running configured version
19	https://api.meraki.com/api/v1	Q2LD-ZWCZ-UA77	Office AP	MR52	192.168.1.177	wireless-29-5-1

Reconcile to Service Now

- Get Service Now Data
- Compare to existing list using Pandas function
- Push updated entries into Service Now

And now we merge... and create diff. this will show what is in system of truth and not system of record

```
# theBigInventory DataSource  Serial  Hostname  Model IP Address  Version

InventoryNotInSvcnow_df=theBigInventory.merge(svcnow_inventory_df, how = 'outer' ,indicator=True,left_on=["Hostname","IP Address","Model","Version"], right_on=["name","ip_address","model_number","firmware_version"]).loc[lambda x : x['_merge']=='left_only']
```

```
[ ] # Evaluate the diff
    InventoryNotInSvcnow_df
```

Thank you