Cisco Live!
January 29 - February 2, 2018 · Barcelona
Managing Cisco UCS with the Python SDK

John McDonough, Technical Leader – Developer Evangelist
Questions?

Use Cisco Spark to communicate with the speaker after the session

How

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

cs.co/ciscolivebot#DEVNET-2060
Agenda – DEVNET 2060

- Managing Cisco UCS with the Python SDK
  - Installation Overview
  - Connect / Query / Filter / Dump XML
  - Get Help / Variable Inspection / Meta Data
  - Configure / Transactions
  - Compare / Sync / Code Generation

- Conclusion
UCS Python SDK
Installation
Cisco UCS Python SDKs - Install

- Hosted on Github
  - UCS Manager SDK
    - Source – https://github.com/CiscoUcs/ucsmsdk
    - Samples – https://github.com/CiscoUcs/ucsmsdk_samples
    - Documents – https://CiscoUcs.github.io/ucsmsdk_docs

- UCS IMC SDK
  - Source – https://github.com/CiscoUcs/imcsdk
  - Documents – https://ciscoucs.github.io/imcsdk_docs

- UCS Central Python SDK
  - Source – https://github.com/CiscoUcs/ucscentralsdk
Cisco UCS Python SDKs - Install

• Install Python 2.7.X or 3.5.X / 3.6.X
  • Or both UCS Python SDKs can coexist!

• pip – Preferred Installer Program (package manager)
  • Installs latest “Release” from https://pypi.python.org/pypi

    pip install ucsmsdk

• git – Install from github.com – works for ucsmsdk, imcsdk, ucscentralsdk
  • Installs latest SDK source code from https://github.com/CiscoUcs/ucsmsdk

    git clone https://github.com/CiscoUcs/ucsmsdk.git
    cd ucsmsdk
    C:\Python27\python.exe .\setup.py install ← Python 2

    'C:\Program Files\Python35\python.exe' .\setup.py install ← Python 3
Connect / Query / Filter / Dump XML
from ucsmsdk.ucshandle import UcsHandle
handle = UcsHandle("198.18.133.91","admin","password")
handle.login()

handle.cookie

blades = handle.query_classid("ComputeBlade")
len(blades)

for blade in blades:
    print blade.dn, blade.serial, blade.model

handle.set_dump_xml()
blades = handle.query_classid("ComputeBlade")
handle.unset_dump_xml()
Connect / Query / Filter / Dump XML

```python
filter_exp=('model","UCSB-B200-M4")'
blades = handle.query_classid("ComputeBlade",filter_str=filter_exp)
len(blades)

filter_exp=('model","UCSB-B200-M4", type="eq")'
blades = handle.query_classid("ComputeBlade",filter_str=filter_exp)
len(blades)

filter_exp=('model","ucsB-B200-m4", flag="I")'

for blade in blades:
    print blade.dn

blade_by_dn = handle.query_dn("sys/chassis-1/blade-2")
print blade_by_dn
```
blades_and_chassis = handle.query_classids("ComputeBlade","EquipmentChassis")

print blades_and_chassis
print blades_and_chassis['ComputeBlade']

for blade in blades_and_chassis['ComputeBlade']:
    print blade.dn

for chassis in blades_and_chassis['EquipmentChassis']:
    print chassis.dn

blade_and_chassis = handle.query_dns("sys/chassis-1/blade-1","sys/chassis-1")
print blade_and_chassis
print blade_and_chassis['sys/chassis-1/blade-1'].dn
handle.logout()
Get Help / Variable Inspection / Metadata

```python
vars(handle)
dir(UcsHandle)
help(UcsHandle)

from ucsmsdk.ucscoreutils import get_meta_info
meta = get_meta_info(class_id="FabricVlan")
print meta

meta = get_meta_info(class_id="FabricVlan", include_prop=False, show_tree=False)
print meta

meta = get_meta_info(class_id="FabricVlan", include_prop=False, show_tree=True)
print meta
```
Configure / Transactions
from ucsmsdk.ucshandle import UcsHandle
from ucsmsdk.mometa.fabric.FabricVlan import FabricVlan
handle = UcsHandle("198.18.133.91", "admin", "password")
handle.login()

fabric_lan_cloud = handle.query_classid("FabricLanCloud")
vlan = FabricVlan(parent_mo_or_dn=fabric_lan_cloud[0], sharing="none",
                  name="vlan100", id="100", mcast_policy_name="",
                  policy_owner="local", default_net="no", pub_nw_name="",
                  compression_type="included")

handle.add_mo(vlan)
handle.commit()
handle.logout()
from ucsmsdk.ucshandle import UcsHandle
from ucsmsdk.mometa.fabric.FabricVlan import FabricVlan
handle = UcsHandle("192.168.220.212", "admin", "password")
handle.login()

fabric_lan_cloud = handle.query_classid("FabricLanCloud")
vlans = ['200', '300']

for vlan in vlans:
    vlan = FabricVlan(parent_mo_or_dn=fabric_lan_cloud[0], sharing="none",
                       name="vlan" + vlan, id=vlan, mcast_policy_name="",
                       policy_owner="local", default_net="no", pub_nw_name="",
                       compression_type="included")
    handle.add_mo(vlan)

handle.commit()
handle.logout()
from ucsmsdk.ucshandle import UcsHandle
from ucsmsdk.utils import comparesyncmo

source_ucs = UcsHandle("192.168.220.212", "admin", "password")
target_ucs = UcsHandle("192.168.220.209", "admin", "password")
source_ucs.login()
target_ucs.login()

source_ucs_vlans = source_ucs.query_classid("fabricVlan")
target_ucs_vlans = target_ucs.query_classid("fabricVlan")

difference_vlans = comparesyncmo.compare_ucs_mo(target_ucs_vlans, source_ucs_vlans)

# print the difference to the console
compareresyncmo.write_mo_diff(difference_vlans)

compareresyncmo.sync_ucs_mo(target_ucs, difference_vlans, delete_not_present=True)

source_ucs.logout()
target_ucs.logout()
Code Generation

1. control-option/alt-q (MAC)
   Ctrl-Alt-q (Windows)
2. Click Record XML
3. Do Configuration
4. Click Stop XML Recording
5. Download XML file
6. Feed XML file to UCS Python SDK conversion method

```python
from ucsmsdk.utils.converttopython import convert_to_ucs_python
convert_to_ucs_python(xml=True, literal_path="C:\\Users\\demouser\\Downloads\\vlan_ops_xmlReq.log")
```
Conclusion
Get hands-on with APIs and SDKs in the **DevNet Zone**

**LEARN | CODE | INSPIRE | CONNECT**

In the DevNet Zone we offer:

- **Classrooms** – Learn about a specific topic/technology
- **NEW! Hack, Defend and Win!** – Embrace your inner hacker, then defend the network! Black Hat, White Hat Security Challenge
- **DevNet Sandbox** – Explore free technology-packed 24x7 labs and play IoT foosball! Go for a 5-minute mini-hack challenge or complete all three in 15-minutes to win
- **Learning Labs** – Find an open seat to start coding with self-paced tutorials (*experts available*)
- **Panels** – Get inspired & interact with industry experts
- **Connected Women’s Event** – Join Susie Wee on Monday for this event... followed by a DevNet Zone open house
- **NEW! Application Developer Zone and Challenge** – Win a special-edition t-shirt and bragging rights...complete a learning lab, a workshop, and an application developer demo
- **Demos** – Exciting demos and inspiring partner solutions showcase
- **Workshops** – code with an instructor in a hands-on session

Join DevNet!
developer.cisco.com/join/CLEUR18

All DevNet members win a prize, visit the Info Desk to learn more!

Cisco Live Barcelona
January 29 – February 2, 2018

@CiscoDevNet | #DevNet | #CLEUR
Hack, Defend, and Win!

Come play the **Black Hat, White Hat Security Challenge** in the DevNet Zone!

>>> Hack

Embrace your inner hacker and try to access the network

>>> Defend

Be the heavily-armed Cisco defender that saves the day

>>> Win

Collect your prize and claim your bragging rights

@CiscoDevNet
#DevNet | #CLEUR
Questions?
Use Cisco Spark to communicate with the speaker after the session

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

cs.co/ciscolivebot#DEVNET-2060
• Please complete your Online Session Evaluations after each session

• Complete 4 Session Evaluations & the Overall Conference Evaluation (available from Thursday) to receive your Cisco Live T-shirt

• All surveys can be completed via the Cisco Live Mobile App or the Communication Stations

Don’t forget: Cisco Live sessions will be available for viewing on-demand after the event at www.ciscolive.com/global/on-demand-library/.
Continue Your Education

- DevNet: https://developer.cisco.com
- DevNet Learning Labs: https://learninglabs.cisco.com
- Demos in the Cisco campus
- Walk-in Self-Paced Labs
- Tech Circle
- Meet the Engineer 1:1 meetings
- Related sessions
Got more questions? Come find me!

:jomcdono@cisco.com

:@johnamcdonough

:http://github.com/movinalot

:@CiscoDevNet

:facebook.com/ciscodevnet

:http://github.com/CiscoDevNet
Thank you