

Start Now Cisco UCS and Intersight Programmability

John McDonough – DevNet Developer Advocate
@johnamcdonough

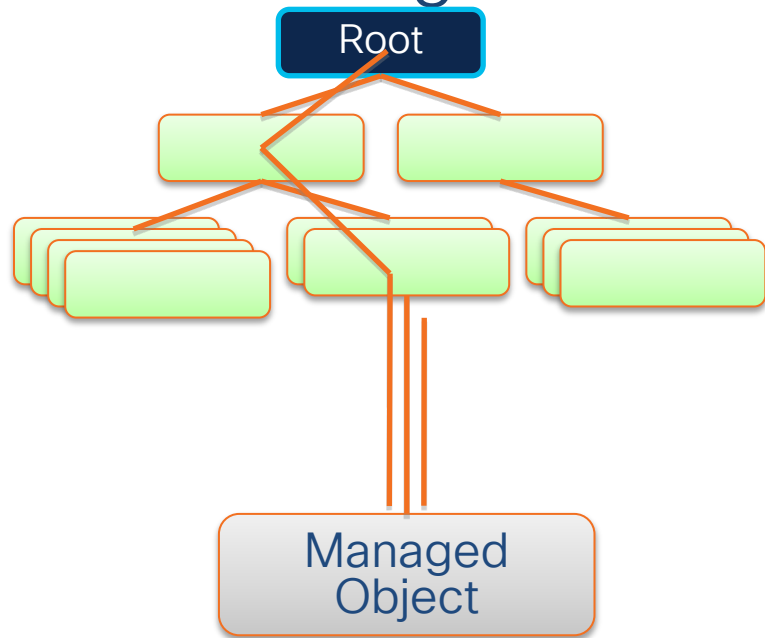


Agenda

- Introduction
- Cisco UCS
 - PowerTool
 - Python SDK
 - Ansible
- Intersight
 - REST API – Native Python
 - Ansible

Cisco UCS

UCS Management Information Model



Everything is an Object

Objects are hierarchically organized

Objects belong to a Class: Card, Port, VNIC...

Objects have Class Inheritance
computeBlade is a subclass of computePhysical

Objects represent physical & logical entities

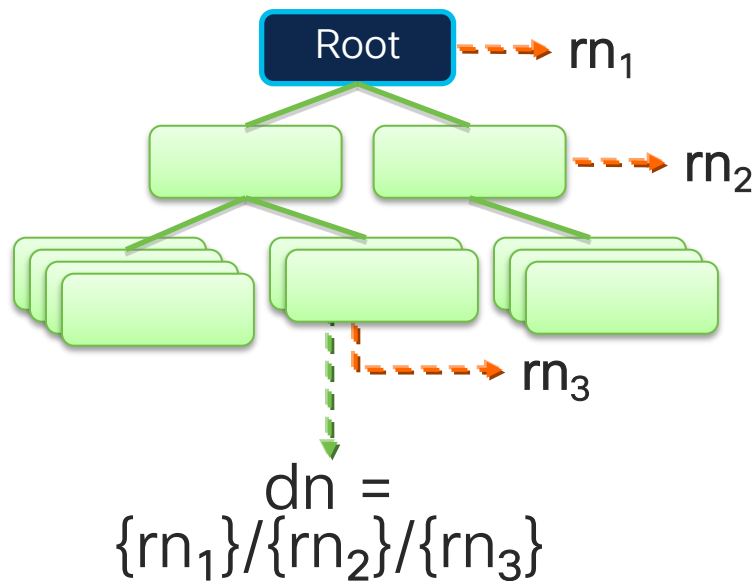
Objects are uniquely identified by a DN

Objects have attributes

UCS XML Database contains comprehensive system information

- Discovered components
- System configuration
- Operational status including statistics and faults

Distinguished Names



RN: identifies an object relative to its parent object.

DN: Slash delimited sequence of RNs

DN: Immutable

DN: Provides a fully qualified path to object

DN: unambiguously identifies a target object.

Example:

```
<dn = "sys/chassis-5/blade-2/adaptor-1/host-eth-2"/>
```

is composed of the following relative names:

topSystem MO: rn="sys"

equipmentChassis MO: rn="chassis- $\langle id \rangle$ "

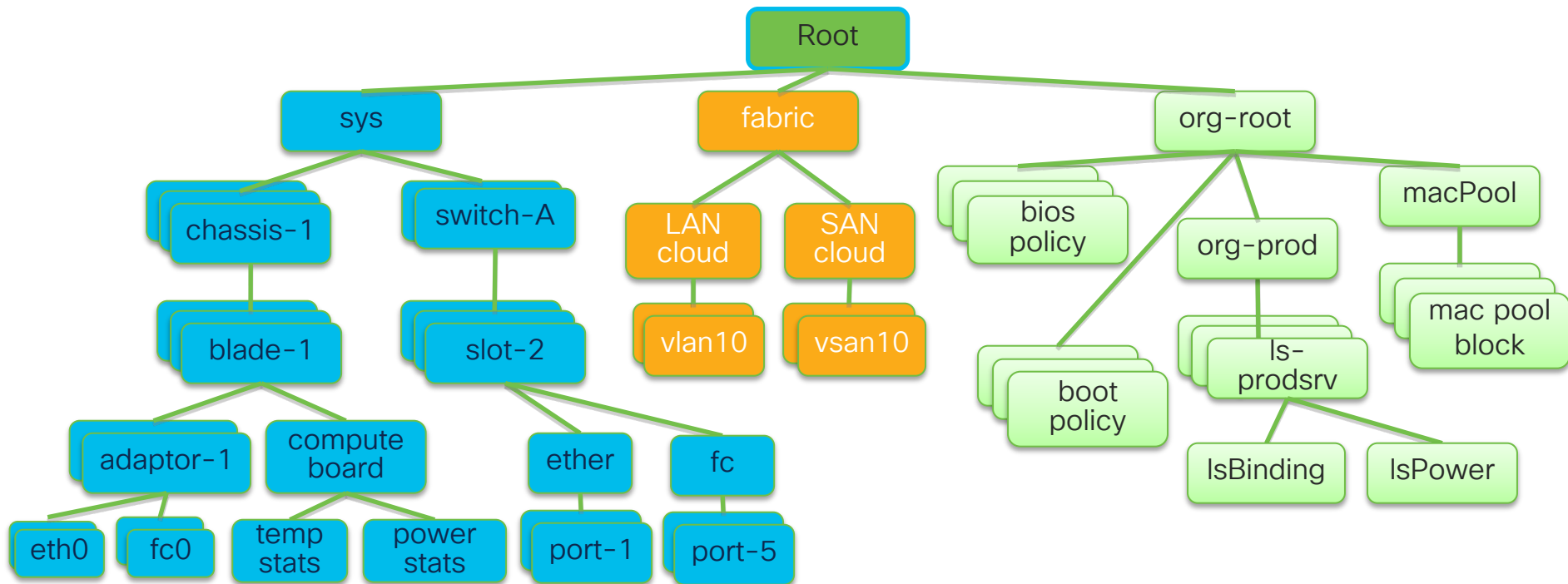
computeBlade MO: rn = "blade- $\langle slotId \rangle$ "

adaptorUnit MO: rn="adaptor- $\langle id \rangle$ "

adaptorHostEthIf MO: rn="host-eth- $\langle id \rangle$ "

UCS Manger – Object Model

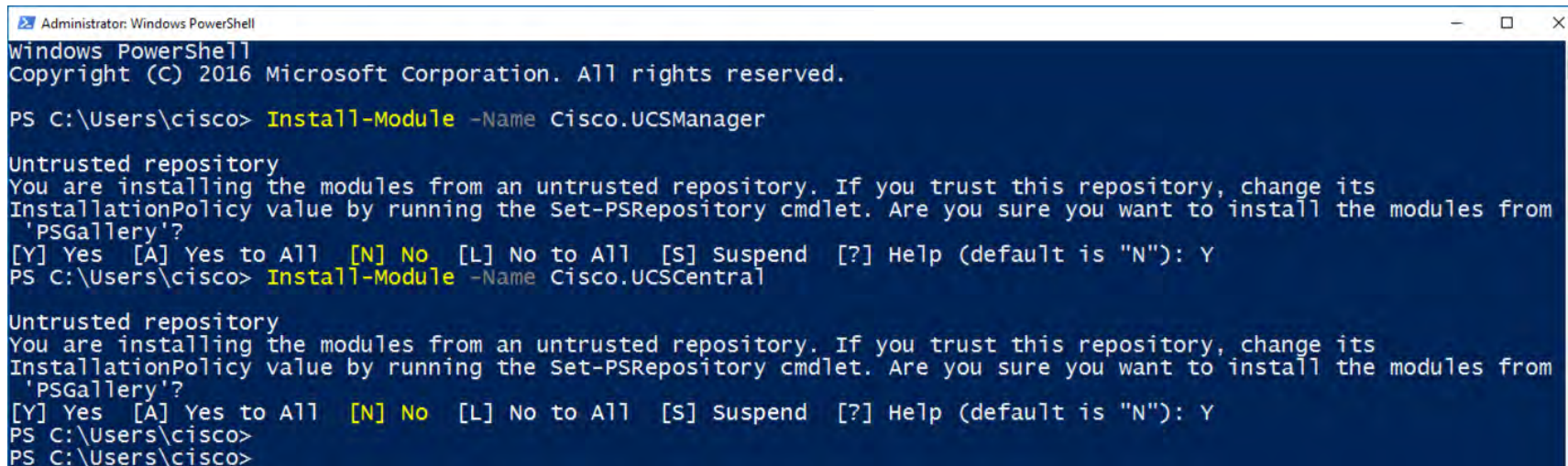
Managed Object Tree



UCS PowerTool

Windows PowerShell – UCS Powertool Install-Module / Count the Cmdlets

- Install what you need – use Install-Module



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

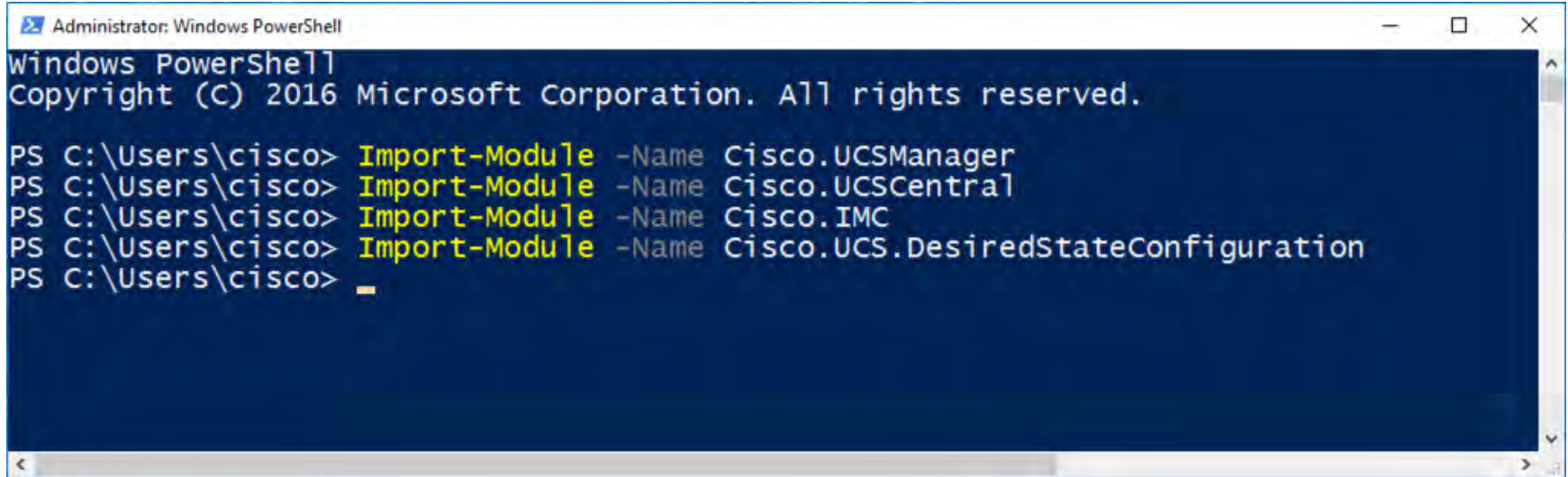
PS C:\Users\cisco> Install-Module -Name Cisco.UCSManager

Untrusted repository
You are installing the modules from an untrusted repository. If you trust this repository, change its
InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from
'PSGallery'?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): Y
PS C:\Users\cisco> Install-Module -Name Cisco.UCSCentral

Untrusted repository
You are installing the modules from an untrusted repository. If you trust this repository, change its
InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from
'PSGallery'?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): Y
PS C:\Users\cisco>
PS C:\Users\cisco>
```

- PowerShell C:\> Get-Command -Module Cisco.UCSManager | Measure → 2472
- PowerShell C:\> Get-Command -Module Cisco.IMC | Measure → 828
- PowerShell C:\> Get-Command -Module Cisco.UCSCentral | Measure → 2398
- PowerShell C:\> Get-Command -Module Cisco.UCS.DesiredStateConfiguration | Measure → 2
- PowerShell C:\> Get-Command -Module Cisco.UCS.Core | Measure → 16
- 5716 Cmdlets and Functions in total

Windows PowerShell – UCS Powertool Import-Module



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\cisco> Import-Module -Name Cisco.UCSManager
PS C:\Users\cisco> Import-Module -Name Cisco.UCSCentral
PS C:\Users\cisco> Import-Module -Name Cisco.IMC
PS C:\Users\cisco> Import-Module -Name Cisco.UCS.DesiredStateConfiguration
PS C:\Users\cisco> _
```

Linux PowerShell Core – UCS PowerTool Core Download / Install / Count the Cmdlets

- Download from [community.cisco.com](https://community.cisco.com/t5/cisco-developed-ucs-integrations/cisco-ucs-powertool-core-suite-for-powershell-core-modules-for/ta-p/3985798)
 - <https://community.cisco.com/t5/cisco-developed-ucs-integrations/cisco-ucs-powertool-core-suite-for-powershell-core-modules-for/ta-p/3985798>
- Latest Version – 2.5.3.0 Beta – ucspowertoolcore_v2.5.3.0.zip

```
Welcome to Cisco UCS PowerTool Core Suite 2.5.3.0 Beta

Log in to a UCS Manager Domain:
To generate PowerShell code from a running UCS Manager GUI:
To show object metadata for all UCS PowerTool commands:

Log in to an IMC of standalone UCS server:
Generate PowerShell code from a running IMC of standalone UCS server:
Show object metadata for all IMC PowerTool cmdlets:

Log in to a UCS Central server:
To generate PowerShell code from a running UCS Central GUI:
To show object metadata for all UCS Central PowerTool commands:
Once you've connected, display the details of active session(s):

If you need more help, visit the UCS PowerTool user community:
```

| | | |
|---------------------|---------------------|----------------------------|
| Connect-Ucs | Connect-Imc | Connect-UcsCentral |
| ConvertTo-UcsCmdlet | ConvertTo-ImcCmdlet | ConvertTo-UcsCentralCmdlet |
| Get-UcsCmdletMeta | Get-ImcCmdletMeta | Get-UcsCentralCmdletMeta |
| | | Get-UcsPSSession |

<https://communities.cisco.com/ucsintegrations>

- PS /home/user/> Get-Command -Module Cisco.UCSManager | Measure → 2470
- PS /home/user/> Get-Command -Module Cisco.IMC | Measure → 826
- PS /home/user/> Get-Command -Module Cisco.UCSCentral | Measure → 2395
- PS /home/user/> Get-Command -Module Cisco.UCS.Core | Measure → 15
- 5706 Cmdlets and Functions in total

Linux PowerShell Core – UCS PowerTool Core Download / Install / Count the Cmdlets

```
JOMCDONO-M-V1ZP:~ jomcdono$ pwsh
```

```
PowerShell 6.2.3
```

```
Copyright (c) Microsoft Corporation. All rights reserved.
```

```
https://aka.ms/pscore6-docs
```

```
Type 'help' to get help.
```

```
PS movinalot ~ > ./local/share/powershell/Modules/Start-UcsPowerTool.ps1
```

```
Welcome to Cisco UCS PowerTool Core Suite 2.5.3.0 Beta
```

```
Log in to a UCS Manager Domain:
```

```
To generate PowerShell code from a running UCS Manager GUI:
```

```
To show object metadata for all UCS PowerTool commands:
```

```
Connect-Ucs
```

```
ConvertTo-UcsCmdlet
```

```
Get-UcsCmdletMeta
```

```
Log in to an IMC of standalone UCS server:
```

```
Generate PowerShell code from a running IMC of standalone UCS server:
```

```
Show object metadata for all IMC PowerTool cmdlets:
```

```
Connect-Imc
```

```
ConvertTo-ImcCmdlet
```

```
Get-ImcCmdletMeta
```

```
Log in to a UCS Central server:
```

```
To generate PowerShell code from a running UCS Central GUI:
```

```
To show object metadata for all UCS Central PowerTool commands:
```

```
Once you've connected, display the details of active session(s):
```

```
Connect-UcsCentral
```

```
ConvertTo-UcsCentralCmdlet
```

```
Get-UcsCentralCmdletMeta
```

```
Get-UcsPSSession
```

```
If you need more help, visit the UCS PowerTool user community:
```

```
https://communities.cisco.com/ucsintegrations
```

```
pwsh -NoExit -File ~/.local/share/powershell/Modules/Start-UcsPowerTool.ps1
```

Connect / Query / Dump XML

> ex-01.ps1 ✕

DEVWKS-2061 > > ex-01.ps1

```
1  # ex_01.ps1
2  # Connect
3  Connect-Ucs -Name 198.18.133.91
4  Get-UcsPSSession
5  Get-help Connect-Ucs
6
7  # Query UCS Manager
8  Get-UcsBlade
9  Get-UcsBlade | Select-Object Dn,Serial,Model
10 Get-UcsBlade -Xml
11
12 # Get Cmdlets are Class Based but can retrieve a Single Object
13 Get-UcsBlade -Dn sys/chassis-3/blade-1 -Xml | Select-Object Dn, Serial, Model
14
```


Query – Filter

> ex-02.ps1 ✕

DEVWKS-2061 ▶ > ex-02.ps1

```
1  # ex_02.ps1
2  # PowerTool is case sensitive by default
3  Get-UcsBlade -Model UCSB-B200-M4 | Select-Object Dn, Serial, Model
4  Get-UcsBlade -Model ucsb-B200-M4 | Select-Object Dn, Serial, Model
5
6  # PowerShell is case insensitive by default
7  Get-UcsBlade | ?{$_ .Model -eq 'ucsb-B200-M4'} | Select-Object Dn, Serial, Model
8
9  # Filter Strings
10 Get-UcsBlade -Filter 'Model -ilike UcsB-b200-M3' -Xml | Select-Object Dn, Model | more
11 Get-UcsBlade -Filter 'Model -ilike UcsB-b200-M4' -Xml | Select-Object Dn, Model | more
12
```

Configure

```
ex-06.ps1 x
DEVWKS-2061 > ex-06.ps1 ...
1 # ex-06.ps1
2 # Configure
3
4 # Add a VLAN
5 Get-UcsLanCloud | Add-UcsVlan -Name CiscoLive -Id 500
6
7 # Add an organization
8 $ucsOrg = Add-UcsOrg -Name CiscoLiveOrg
9 Add-UcsServiceProfile - $ucsOrg )rg -Name CiscoLiveSP
10 Add-UcsBootPolicy -Org $ucsOrg -Name CL_BP -RebootOnUpdate yes -BootMode legacy | Add-UcsLsbootVirtualMedia -Access "read-only-local" -LunId "0" -Order 1
11
12 Get-UcsServiceProfile -Name CiscoLiveSP | Set-UcsServiceProfile -BootPolicyName CL_BP -Force
13 Add-UcsServiceProfile -Org $ucsOrg -Name CiscoLiveSP -BootPolicyName CL_BP -ModifyPresent
14 Get-UcsServiceProfile -Name CiscoLiveSP
15
16 Get-UcsOrg -Name CiscoLiveOrg | Get-UcsBootPolicy -Name CL_BP | Remove-UcsBootPolicy -Force
17 Get-UcsServiceProfile -Name CiscoLiveSP
18
```

Transactions

> ex-07.ps1 x

DEVWKS-2061 > ex-07.ps1 > ...

```
1 # ex-07.ps1
2 // Transactions
3 $ucsLanCloud = Get-UcsLanCloud
4
5 Start-UcsTransaction
6 101..104 | %{Add-Ucsvlan -LanCloud $ucsLanCloud -Name vlan$_ -id $_}
7 Complete-UcsTransaction
8
9 Start-UcsTransaction
10 105..120 | %{Add-Ucsvlan -LanCloud $ucsLanCloud -Name vlan$_ -id $_}
11 Complete-UcsTransaction
12
13 Start-UcsTransaction
14 Get-UcsVlan | ?{$_.id -ge 102 -and $_.id -le 120} | Remove-UcsVlan -Force | Out-Null
15 Complete-UcsTransaction | Out-Null
16
```

Compare / Sync

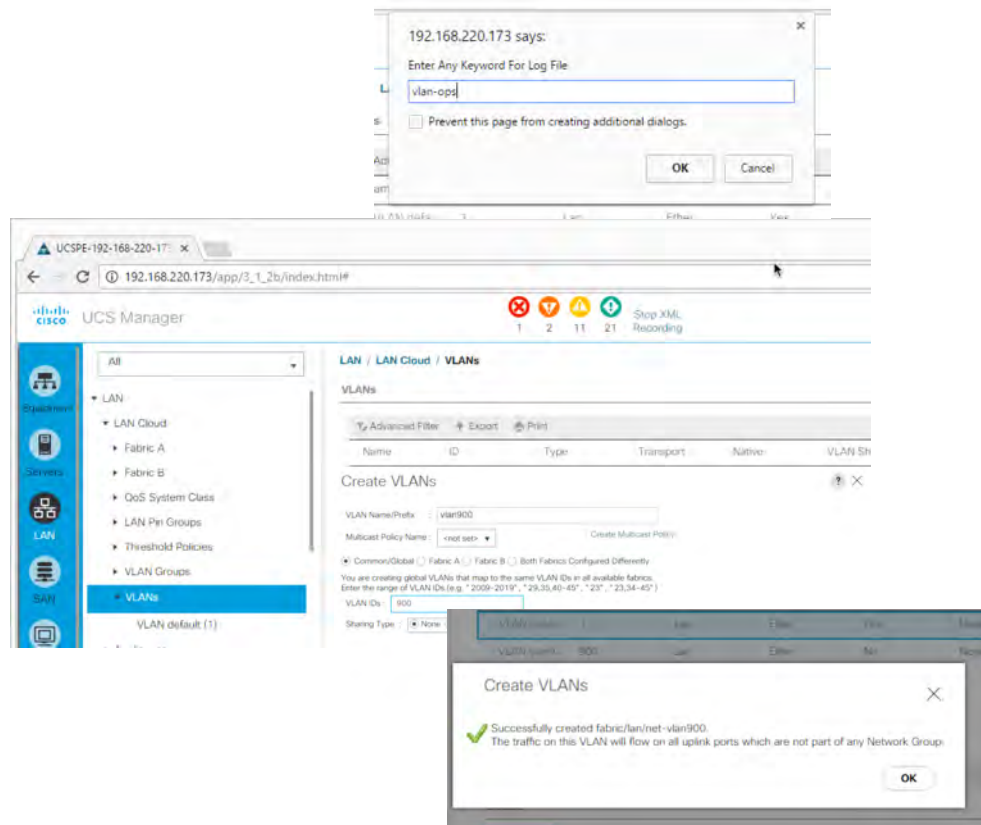
> ex-08.ps1 ✕

DEVWKS-2061 > ex-08.ps1 ▶ ...

```
1 # ex-08.ps1
2 # Compare / Sync
3 $ucsCred = Get-Credential
4 $ucsSource = Connect-Ucs -Name 198.18.133.91 -Credential $ucsCred -NotDefault
5 $ucsTarget = Connect-Ucs -Name 198.18.134.249 -Credential $ucsCred -NotDefault
6
7 Compare-UcsMo $(Get-UcsVlan -Ucs $ucsTarget) $(Get-UcsVlan -Ucs $ucsSource)
8
9 Sync-UcsMo -Ucs $ucsTarget $(Compare-UcsMo $(Get-UcsVlan -Ucs $ucsTarget) $(Get-UcsVlan -Ucs $ucsSource)) -WhatIf -DeleteNotPresent
```


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
ConvertTo-UcsCmdlet



ConvertTo-UcsCmdlet -Xml -LiteralPath C:\Users\demouser\Downloads\org_ops_xmlReq.log

UCS Python SDK

Cisco UCS Python SDKs – Install

- Hosted on Github
 - UCS Manager SDK – 0.9.10
 - Source – <https://github.com/CiscoUcs/ucsmsdk>
 - Samples – https://github.com/CiscoUcs/ucsmsdk_samples
 - Documents – https://CiscoUcs.github.io/ucsmsdk_docs
 - UCS IMC SDK – 0.9.7
 - Source – <https://github.com/CiscoUcs/imcsdk>
 - Documents – https://ciscoucs.github.io/imcsdk_docs
 - UCS Central Python SDK – 0.9.0.1
 - Source – <https://github.com/CiscoUcs/ucscentralsdk>



python



GitHub

Cisco UCS Python SDKs – Install

- Install Python 2.7.X or 3.5.X and greater
 - **Both** Python versions of UCS Python SDKs can coexist!
 - pip – Preferred Installer Program (package manager)
 - Installs latest “Release” from <https://pypi.python.org/pypi>

```
pip install ucsm sdk
pip install ucsc sdk
pip install imc sdk
```
 - git – Install from github.com – works for ucsm sdk, imc sdk, ucsc sdk
 - Installs latest SDK source code from <https://github.com/CiscoUcs/ucsm sdk>

```
git clone https://github.com/CiscoUcs/ucsm sdk/
cd ucsm sdk
sudo make install
```



Connect / Query / Dump XML

ex-01.py x

DEVWKS-2060 x ex-01.py x ...

```
1  """
2      ex-01.py
3  """
4
5  from ucsm.sdk.ucshandle import UcsHandle
6  handle = UcsHandle("198.18.133.91", "admin", "password")
7  handle.login()
8
9  handle.cookie
10
11  blades = handle.query_classid("ComputeBlade")
12  len(blades)
13
14  for blade in blades:
15      print blade.dn, blade.serial, blade.model
16
17  handle.set_dump_xml()
18  blades = handle.query_classid("ComputeBlade")
19  handle.unset_dump_xml()
```

Query - Filter

```
ex-02.py x
DEVWKS-2060 ▸ ex-02.py ▸ ...
1  """
2  ▸ ex-02.py
3  """
4
5  filter_exp='(model,"UCSB-B200-M4")'
6  blades = handle.query_classid("ComputeBlade",filter_str=filter_exp)
7  len(blades)
8
9  filter_exp='(model,"UCSB-B200-M4", type="eq")'
10 blades = handle.query_classid("ComputeBlade",filter_str=filter_exp)
11 len(blades)
12
13 filter_exp='(model,"UCSB-B200-M4", type="ne")'
14 blades = handle.query_classid("ComputeBlade",filter_str=filter_exp)
15 len(blades)
16
17 filter_exp='(model,"ucsB-B200-m4", flag="I")'
18 blades = handle.query_classid("ComputeBlade",filter_str=filter_exp)
19 len(blades)
20
21 for blade in blades:
22     print blade.dn, blade.model
```

Query - Returns

```
ex-03.py x
DEVWKS-2060 ▸ ex-03.py ▸ ...
1  """
2  ex-03.py
3  """
4
5  # List of Objects Returned
6  blades = handle.query_classid("ComputeBlade")
7
8  # Single Object Returned
9  blade_by_dn = handle.query_dn("sys/chassis-1/blade-2")
10 print blade_by_dn
11
12 # Dictionary of Object Lists Returned
13 blades_and_chassis = handle.query_classids("ComputeBlade","EquipmentChassis")
14 print blades_and_chassis
15
16 print blades_and_chassis['ComputeBlade']
17 for blade in blades_and_chassis['ComputeBlade']:
18     print blade.dn
19
20 # Dictionary of Objects Returned
21 blade_and_chassis = handle.query_dns("sys/chassis-1/blade-1","sys/chassis-2")
22 print blade_and_chassis
23 print blade_and_chassis['sys/chassis-3/blade-1'].dn
```

Configure

ex-05.py x

DEVWKS-2060 ▸ ex-05.py ▸ ...

```
1  """
2      ex-05.py
3  """
4
5  from ucsmsdk.ucshandle import UcsHandle
6  from ucsmsdk.mometa.fabric.FabricVlan import FabricVlan
7  handle = UcsHandle("198.18.133.91", "admin", "password")
8  handle.login()
9
10 fabric_lan_cloud = handle.query_classid("FabricLanCloud")
11 vlan = FabricVlan(parent_mo_or_dn=fabric_lan_cloud[0],
12                   name="vlan100",
13                   id="100")
14
15 handle.add_mo(vlan)
16 handle.commit()
17 handle.logout()
```


Transactions

ex-06.py x

DEVWKS-2060 ▸ ex-06.py ▸ ...

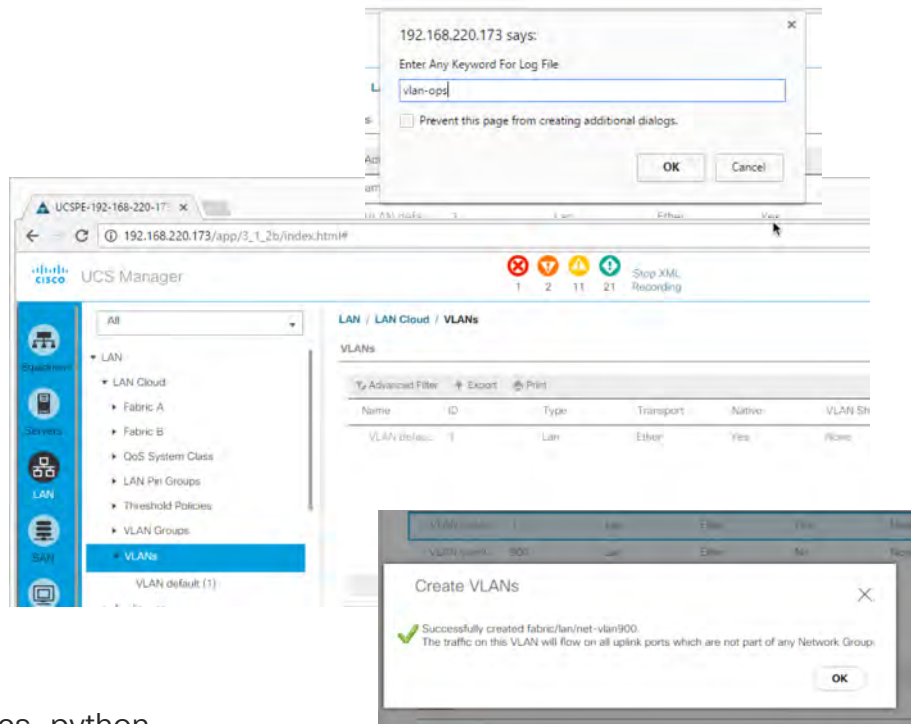
```
1  """
2  ex-06.py
3  """
4
5  from ucsm.sdk.ucshandle import UcsHandle
6  from ucsm.sdk.mometa.fabric.FabricVlan import FabricVlan
7  handle = UcsHandle("198.18.133.91", "admin", "password")
8  handle.login()
9
10 fabric_lan_cloud = handle.query_classid("FabricLanCloud")
11 vlans = ['200', '300', '400', '500']
12
13 for vlan in vlans:
14     vlan = FabricVlan(parent_mo_or_dn=fabric_lan_cloud[0],
15                       name="vlan" + vlan,
16                       id=vlan)
17
18     handle.add_mo(vlan)
19
20 handle.commit()
21 handle.logout()
```

Compare / Sync

```
ex-07.py x
DEVWKS-2060 ex-07.py ...
1  """
2      ex-07.py
3  """
4
5  from ucsm.sdk.ucshandle import UcsHandle
6  from ucsm.sdk.utils import comparesyncmo
7  source_ucs=UcsHandle("198.18.133.91", "admin", "password")
8  target_ucs=UcsHandle("198.18.134.249", "admin", "password")
9  source_ucs.login()
10 target_ucs.login()
11
12 source_ucs_vlans=source_ucs.query_classid("fabricVlan")
13 target_ucs_vlans=target_ucs.query_classid("fabricVlan")
14
15 difference_vlans=comparesyncmo.compare_ucs_mo(target_ucs_vlans, source_ucs_vlans)
16
17 # print the difference to the console
18 comparesyncmo.write_mo_diff(difference_vlans)
19
20 comparesyncmo.sync_ucs_mo(target_ucs, difference_vlans, delete_not_present=True)
21
22 source_ucs.logout()
23 target_ucs.logout()
24
```

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



```
from ucsm.sdk.utils.converttopython import convert_to_ucs_python
```

```
# Windows
```

```
convert_to_ucs_python(xml=True, literal_path="C:\\User\\demouser\\Downloads\\vlan-ops_xmlReq.log")
```

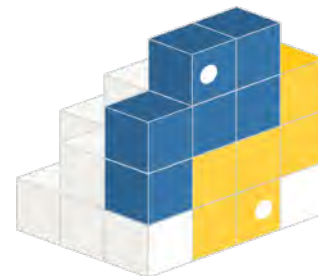
```
# Linux
```

```
convert_to_ucs_python(xml=True, literal_path="/Users/demouser/Downloads/vlan-ops_xmlReq.log")
```

UCS Ansible

Ansible Collections – Install

- Install Python 2.7.X or 3.7.X **OR BOTH!**
 - pip – Preferred Installer Program (package manager)
 - Installs latest “Release” from <https://pypi.python.org/pypi>
`pip install ansible`
- Cisco Intersight and UCS Collections – Modules in Ansible Galaxy
 - Details at <https://galaxy.ansible.com/cisco/ucs>
 - Details at <https://galaxy.ansible.com/cisco/intersight>
 - ansible-galaxy commands to install/update without updates to Ansible core
`ansible-galaxy collection install cisco.ucs`
 - ansible-doc has usage/examples, e.g., `ansible-doc cisco.ucs.ucs_vlans`
`ansible-galaxy collection install cisco.intersight`
 - ansible-doc has usage/examples, e.g., `ansible-doc cisco.Intersight.Intersight_facts`



UCS Ansible Collection

- Ensure Latest Ansible – 2.9 includes UCS Modules
`pip install ansible -U`
- Install UCS Python SDK – ucsmsdk
`pip install ucsmsdk -U`
- Install UCS Collection
`ansible-galaxy collection install cisco.ucs`
- Install/Update UCS Manager Ansible – ucs-ansible
`git clone https://github.com/CiscoUcs/ucsm-ansible`
- Ansible with UCS just like Ansible with other devices

UCS Ansible Collection Playbook

```
# Example Playbook: cisco.ucs.ucs_dns_server
- hosts: ucs
  connection: local
  gather_facts: false

  tasks:
    - name: Test that we have a UCS hostname, UCS username, and UCS password
      fail:
        msg: 'Please define the following variables: ucs_hostname, ucs_username and ucs_password.'
      when: ucs_hostname is not defined or ucs_username is not defined or ucs_password is not defined
      vars:
        # use "<<: *login_info" to substitute the information below in each task
        # this is not required, however it makes the playbook shorter.
        login_info: &login_info
        hostname: "{{ ucs_hostname }}"
        username: "{{ ucs_username }}"
        password: "{{ ucs_password }}"

    - name: Configure DNS server
      cisco.ucs.ucs_dns_server:
        <<: *login_info
        dns_server: 10.10.10.10
        description: DNS Server IP address
        state: present
        delegate_to: localhost
```

Intersight Python

Intersight REST API

<https://github.com/movinalot/intersight-rest-api>

The screenshot shows the GitHub repository page for `movinalot / intersight-rest-api`. The repository has 22 commits, 1 branch, 0 packages, 0 releases, and 1 contributor. The latest commit is `68cdaf2` on Jan 27. The repository contains three files: `key` (update for developer user, 3 months ago), `.gitignore` (wrong repo, 3 months ago), and `Intersight.postman_environment.json` (updates to collection and environment, 11 months ago).

Search or jump to...

Pull requests Issues Marketplace Explore

movinalot / intersight-rest-api

Watch 0 Star 0 Fork 1

Code Issues 2 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights Settings

Intersight REST API interactions with Postman and Python

Manage topics

22 commits 1 branch 0 packages 0 releases 1 contributor MIT

Branch: master New pull request

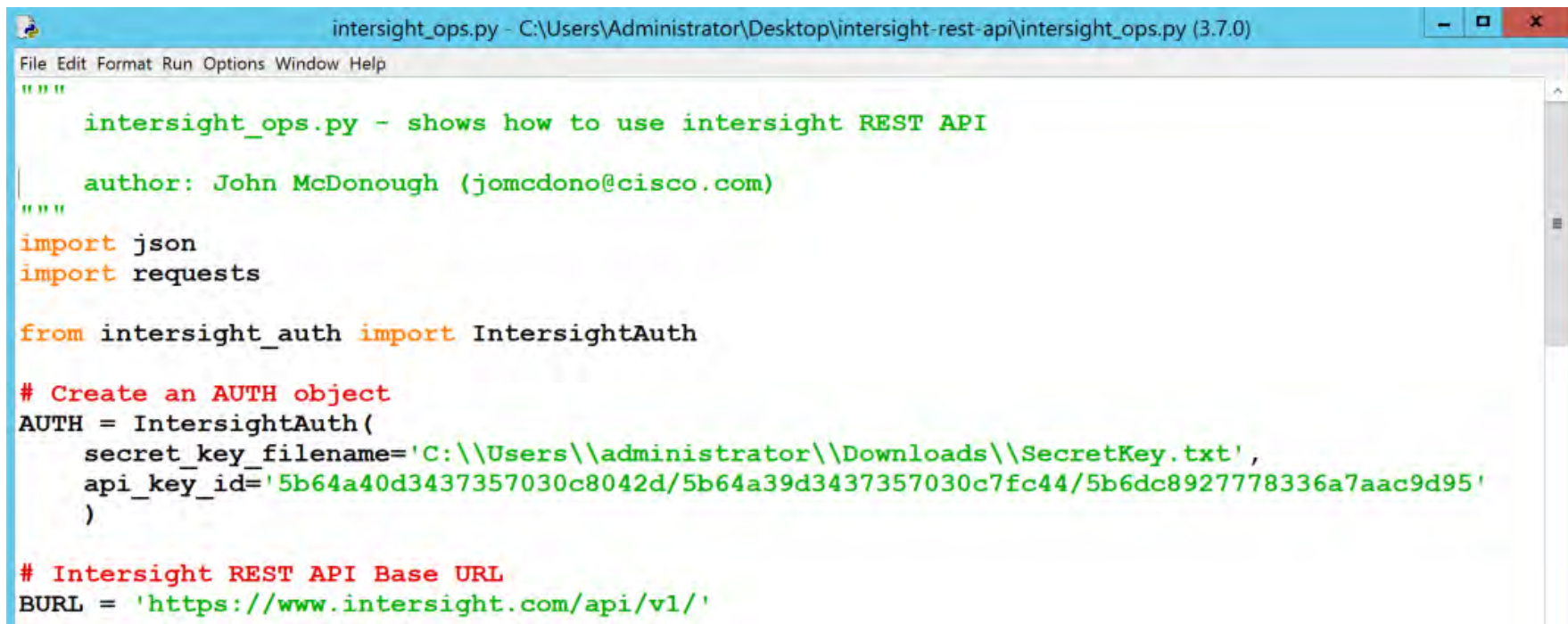
Create new file Upload files Find file Clone or download

movinalot wrong repo Latest commit 68cdaf2 on Jan 27

| | | |
|-------------------------------------|---------------------------------------|---------------|
| key | update for developer user | 3 months ago |
| .gitignore | wrong repo | 3 months ago |
| Intersight.postman_environment.json | updates to collection and environment | 11 months ago |

Python – Update Code with Credentials

Update intersight_ops.py with SecretKey.txt and API Key ID



```
intersight_ops.py - C:\Users\Administrator\Desktop\intersight-rest-api\intersight_ops.py (3.7.0)
File Edit Format Run Options Window Help
"""
    intersight_ops.py - shows how to use intersight REST API

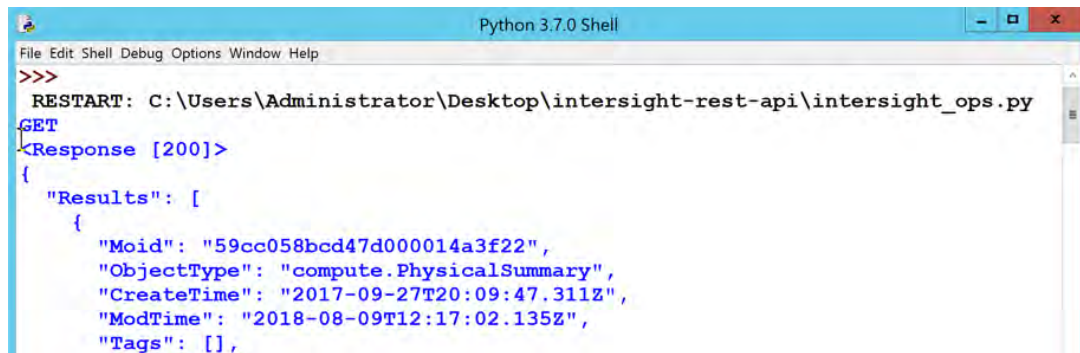
    author: John McDonough (jomcdono@cisco.com)
"""
import json
import requests

from intersight_auth import IntersightAuth

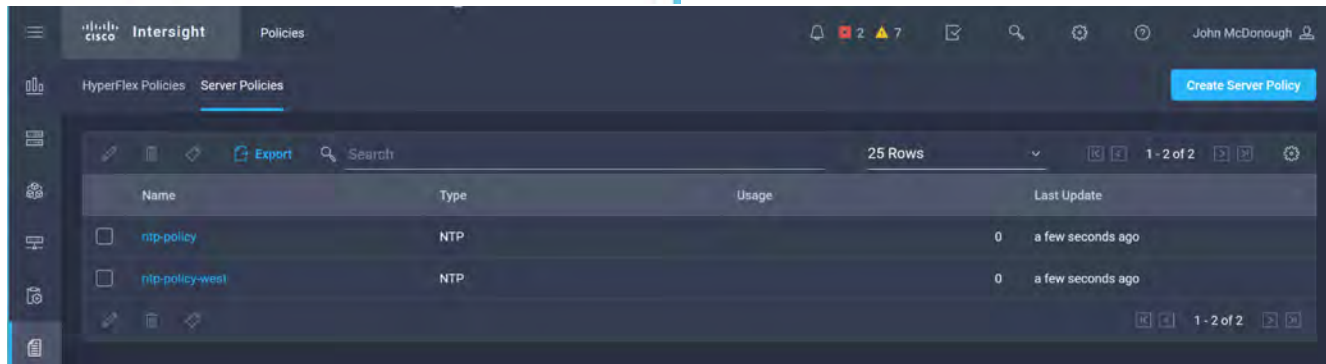
# Create an AUTH object
AUTH = IntersightAuth(
    secret_key_filename='C:\\Users\\administrator\\Downloads\\SecretKey.txt',
    api_key_id='5b64a40d3437357030c8042d/5b64a39d3437357030c7fc44/5b6dc8927778336a7aac9d95'
)

# Intersight REST API Base URL
BURL = 'https://www.intersight.com/api/v1/'
```

Python – Run the Code



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
>>> RESTART: C:\Users\Administrator\Desktop\intersight-rest-api\intersight_ops.py
GET
<Response [200]>
{
  "Results": [
    {
      "Moid": "59cc058bcd47d000014a3f22",
      "ObjectType": "compute.PhysicalSummary",
      "CreateTime": "2017-09-27T20:09:47.311Z",
      "ModTime": "2018-08-09T12:17:02.135Z",
      "Tags": [],
```

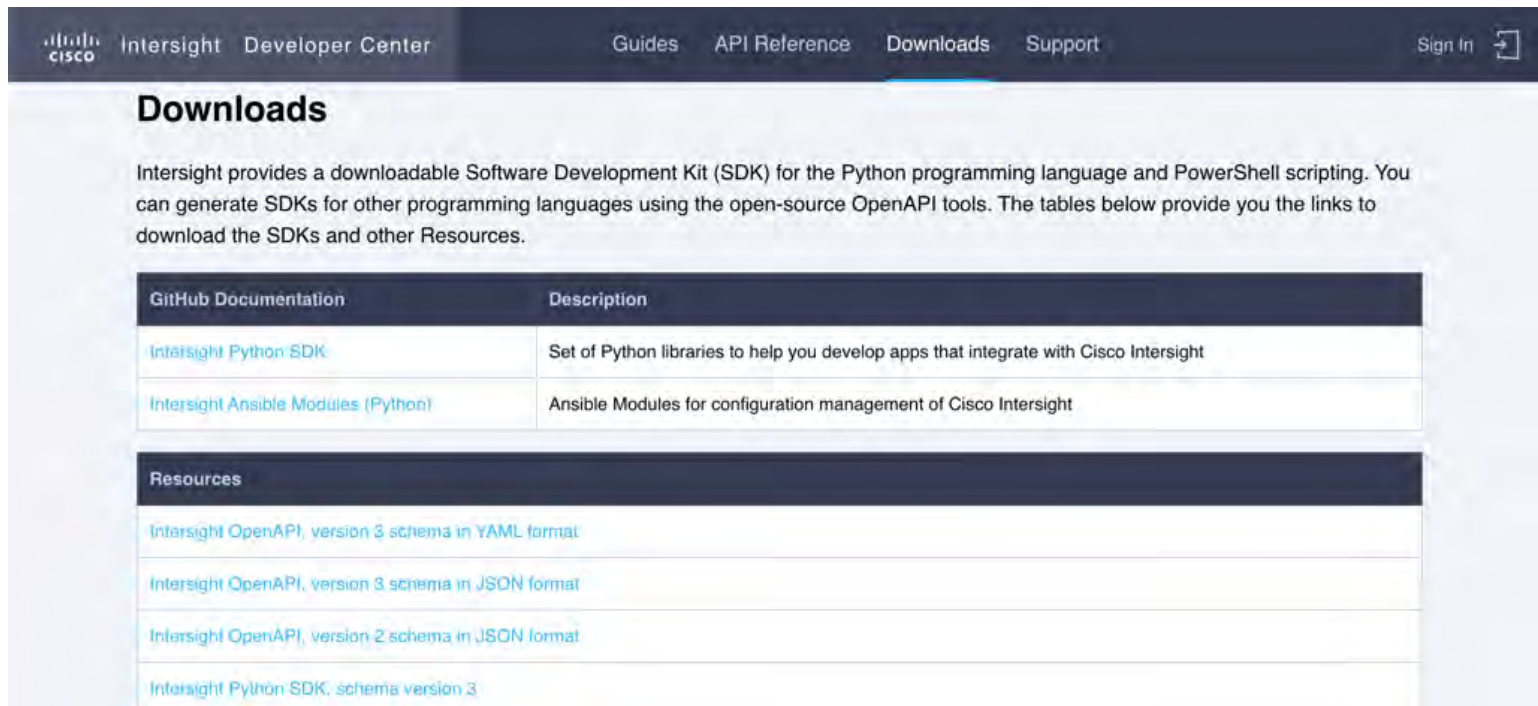


The screenshot shows the Cisco Intersight web interface. The 'Policies' tab is selected, and the 'Server Policies' sub-tab is active. A table displays the current policies. The table has columns for Name, Type, Usage, and Last Update. There are two rows of data, both of type 'NTP'. The first row is 'ntp-policy' and the second is 'ntp-policy-west'. Both have a usage of '0' and were updated 'a few seconds ago'. The interface also shows a 'Create Server Policy' button and pagination information indicating 25 rows and 1 of 2 items.

| Name | Type | Usage | Last Update |
|------------------------------------------|------|-------|-------------------|
| <input type="checkbox"/> ntp-policy | NTP | 0 | a few seconds ago |
| <input type="checkbox"/> ntp-policy-west | NTP | 0 | a few seconds ago |

Intersight Python SDK

<https://intersight.com/apidocs/downloads/>



The screenshot shows the Cisco Intersight Developer Center interface. The top navigation bar includes the Cisco logo, 'Intersight Developer Center', and links for 'Guides', 'API Reference', 'Downloads' (which is highlighted), and 'Support'. A 'Sign In' button with a user icon is on the right. The main content area is titled 'Downloads' and contains a paragraph explaining that Intersight provides a downloadable SDK for Python and PowerShell, and also offers SDKs for other languages via OpenAPI tools. Below this are two tables. The first table, 'GitHub Documentation', lists the 'Intersight Python SDK' and 'Intersight Ansible Modules (Python)' with their descriptions. The second table, 'Resources', lists links to OpenAPI schemas in YAML and JSON formats for versions 3 and 2, and a link to the Python SDK schema version 3.

Downloads

Intersight provides a downloadable Software Development Kit (SDK) for the Python programming language and PowerShell scripting. You can generate SDKs for other programming languages using the open-source OpenAPI tools. The tables below provide you the links to download the SDKs and other Resources.

| GitHub Documentation | Description |
|-----------------------------------------------------|---------------------------------------------------------------------------------------|
| Intersight Python SDK | Set of Python libraries to help you develop apps that integrate with Cisco Intersight |
| Intersight Ansible Modules (Python) | Ansible Modules for configuration management of Cisco Intersight |

| Resources |
|---------------------------------------------------------------------|
| Intersight OpenAPI, version 3 schema in YAML format |
| Intersight OpenAPI, version 3 schema in JSON format |
| Intersight OpenAPI, version 2 schema in JSON format |
| Intersight Python SDK, schema version 3 |

Intersight Ansible

Intersight Ansible Collection

- Ensure Latest Ansible – 2.9 includes UCS Modules
`pip install ansible -U`
- Install Intersight Collection
`ansible-galaxy collection install cisco.Intersight`
- Ansible with Intersight just like Ansible with other devices

Intersight Ansible Collection Playbook

```
---
# Example Playbook: cisco.intersight.intersight_rest_api
- hosts: "{{ group | default('Intersight_Servers') }}"
  connection: local
  gather_facts: false
  vars:
    # Create an anchor for api_info that can be used throughout the file
    api_info: &api_info
    api_private_key: "{{ api_private_key }}"
    api_key_id: "{{ api_key_id }}"
    api_uri: "{{ api_uri | default(omit) }}"
    validate_certs: "{{ validate_certs | default(omit) }}"
    state: "{{ state | default(omit) }}"
  tasks:
    - name: Update server firmware
      cisco.intersight.intersight_rest_api:
        <<: *api_info
        resource_path: /firmware/Upgrades
        query_params:
          $filter: "Server.Moid eq '{{ server_moid }}'"
        update_method: post
```

Explore More

- UCS Learning Labs

Python <https://developer.cisco.com/learning/modules/ucs-python-sdk-introduction>

PowerTool <https://developer.cisco.com/learning/modules/ucs-powertool-introduction>

- UCS Developer Center

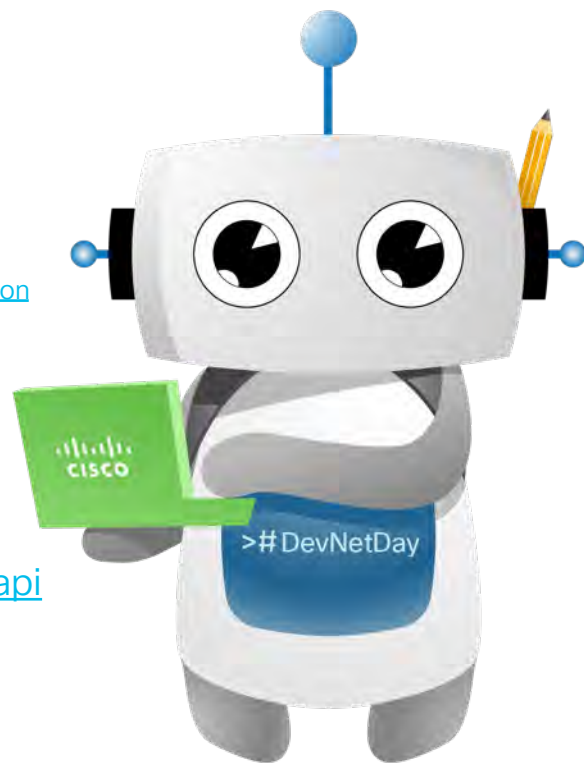
<https://developer.cisco.com/site/ucs-dev-center/>

- Intersight Learning Labs

<https://developer.cisco.com/learning/modules/intersight-rest-api>

- Intersight Developer Center

<https://developer.cisco.com/site/intersight/>



Thank you



Possibilities

#CiscoLive | #DevNetDay