Configure CIMC through Ansible Rest API Module

Contents

Introduction

This document describes how to configure Cisco Integrated Management Controller(CIMC) through the Ansible REST API module.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- UCS CIMC
- API
- Ansible

Components Used

- UCS C220-M4,version 4.1(2f)
- A client running postman and ansible version 2.14.5

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

CIMC API Overview

All the physical and logical components that comprise Cisco UCS are represented in a hierarchical management information model (MIM), also referred to as the MIT. Each node in the tree represents a managed object (MO) or group of objects that contains its administrative state and its operational state.

The hierarchical structure starts at the top (sys) and contains parent and child nodes. Each node in this tree is a managed object and each object in Cisco UCS has a unique distinguished name (DN) that describes the object and its place in the tree. Managed objects are abstractions of the Cisco UCS resources, such as CPUs, DIMMs, adapter cards, fans, and power supply units.

Illustration of the CIMC MIM Structure:

```
|——sys———————————————— (sys)
    |——rack-unit-1————————————(sys/rack-unit-1)
        |——adaptor-1————————————(sys/rack-unit-1/adaptor-1)
        |——psu-1————————————(sys/rack-unit-1/psu-1)
        |——psu-2————————————(sys/rack-unit-1/psu-2)
```

Object Naming:

- DN: The distinguished name enables you to unambiguously identify a target object.
- RN: The relative name identifies an object within the context of its parent object.

For example, this distinguished name:

```
<dn = "sys/rack-unit-1/adaptor-1/host-eth-eth2"/>
```

Is composed of 4 relative names:

```
topSystem MO: rn="sys"
computeRackUnit MO: rn ="rack-unit-1"
adaptorUnit MO: rn="adaptor-<id>"
adaptorHostEthIf MO: rn="host-eth-<id>"
```

API Used in this Article:

- Authentication: **aaaLogin.** Initial method for logging in. Use the aaaLogin method to get a valid cookie.

- Query: **configResolveDn.** Retrieves objects by DN.

- Configuration: **configConfMo.** The configConfMo method is used to configure one or more properties in a Managed Object (MO). The MO to be configured is uniquely identified by a Distinguished Name (DN).

Note:

Many query methods include a inHierarchical argument that accepts a Boolean value (true/yes or false/no). When set to true, this argument causes the method to return all child objects within the hierarchy.

Configurations

1. Find the class or DN of the CIMC Managed Object (MO)

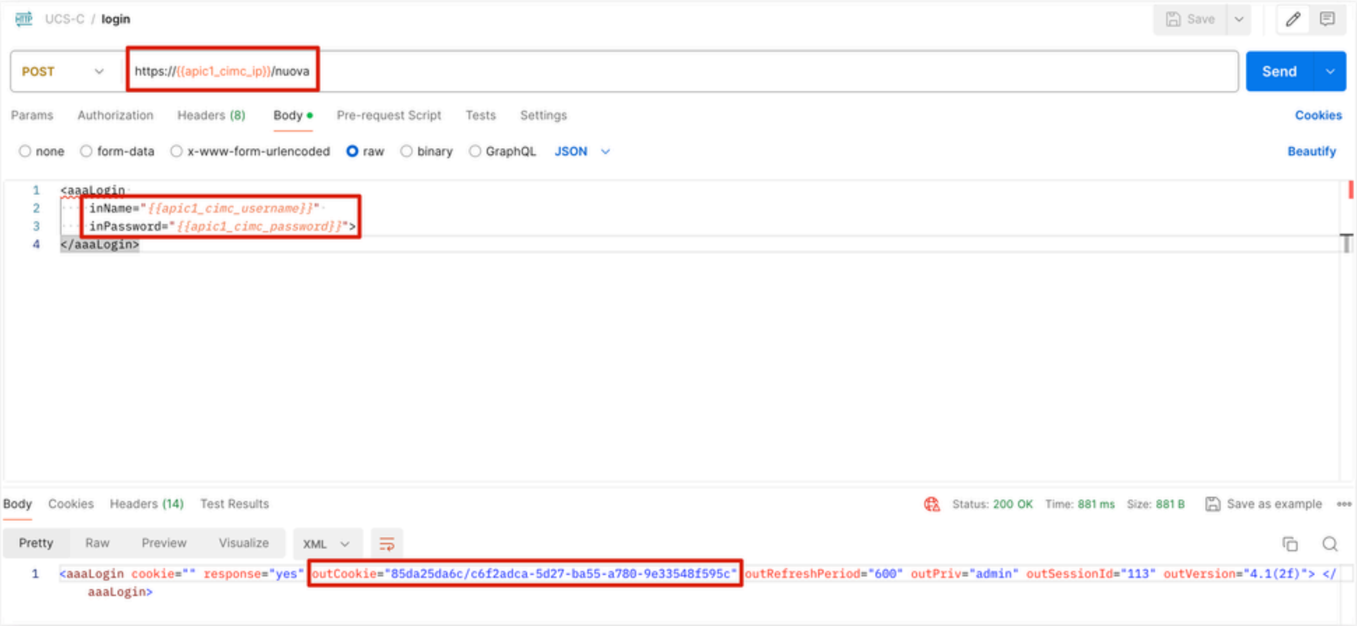
To automate the configuration of CIMC via its API, it is essential to determine the specific class or Distinguished Name (DN) information associated with the Managed Objects (MOs) you wish to configure.

1a. Use the API to log in to the CIMC and retrieve the cookie information

Send a POST request to https://{{apic_cimc_ip}}/nuova and specify the aaaLogin method. Input a username and password.

```
<aaaLogin
   inName='{{ cimc_username }}'
   inPassword={{ cimc_password }}'>
</aaaLogin>
```

Copy the cookie from the API response.



UCS-C / login

POST https://{{apic1_cimc_ip}}/nuova

```
<aaaLogin
   inName="{{apic1_cimc_username}}"
   inPassword="{{apic1_cimc_password}}">
</aaaLogin>
```

Status: 200 OK Time: 881 ms Size: 881 B

```
<aaaLogin cookie="" response="yes" outCookie="85da25da6c/c6f2adca-5d27-ba55-a780-9e33548f595c" outRefreshPeriod="600" outPriv="admin" outSessionId="113" outVersion="4.1(2f)"> </
    aaaLogin>
```

Alternatively, you can utilize curl to fetch the cookie information.

```
curl -k -d "<aaaLogin inName='{{ cimc_username }} ' inPassword='{{ cimc_password }}'></aaaLogin>" https://apic_cimc_ip/nuova
```

1b. Using the API query method configResolveDn to retrieve all Managed Object (MO) information

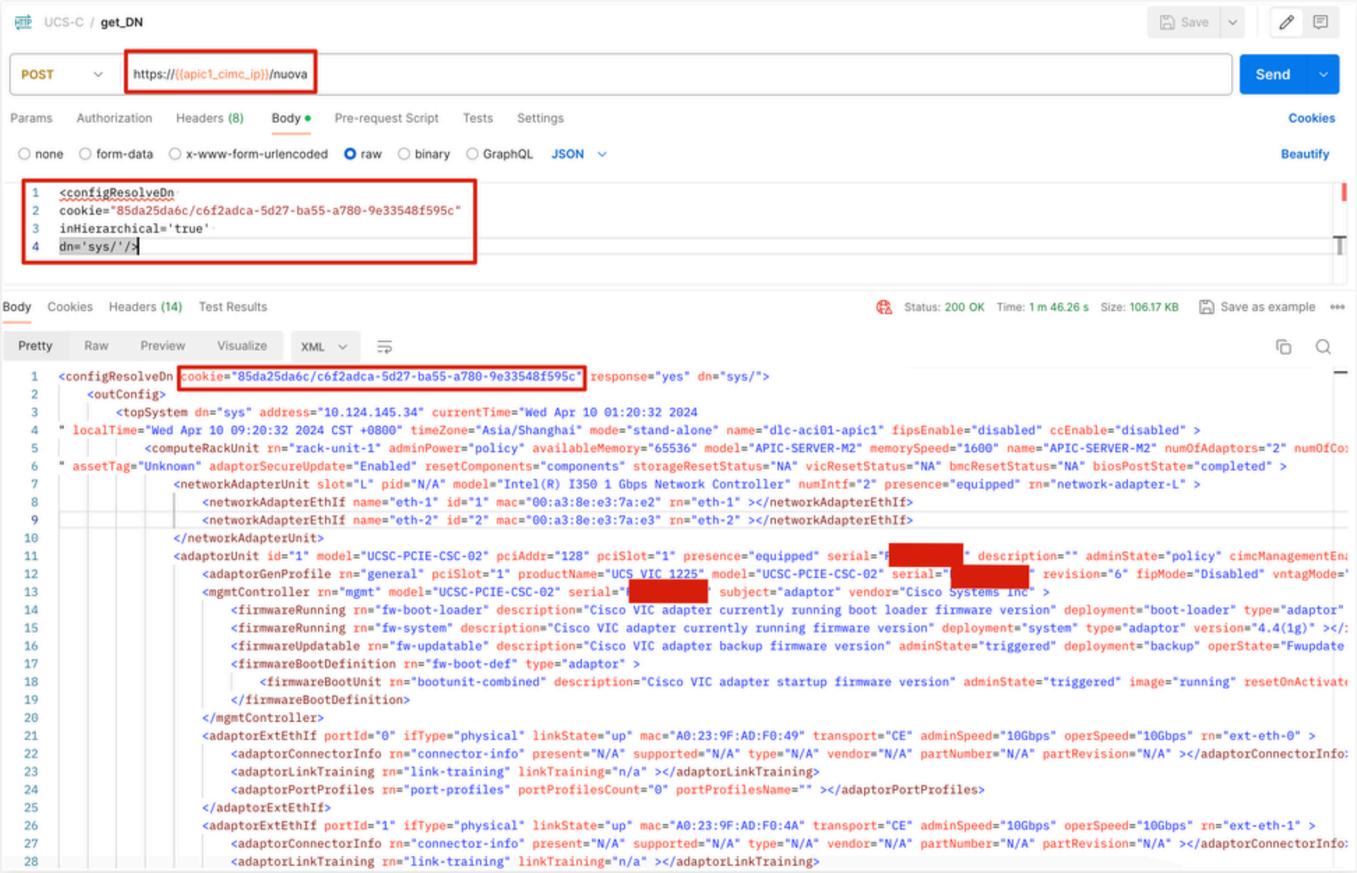
When using **configResolveDn** with **inHierarchical**="**true**" and **dn="sys/**", it retrieves all Managed Object (MO) information from CIMC.

configResolveDn: The configResolveDn method retrieves a single managed object for a specified DN.

inHierarchical=true: When set to true, it returns all child node information. This combination enables fetching all nodes and sub-nodes MO information from CIMC.

dn="sys/": This is the top root of MIT.

API Response:



https://{{apic1_cimc_ip}}/nuova

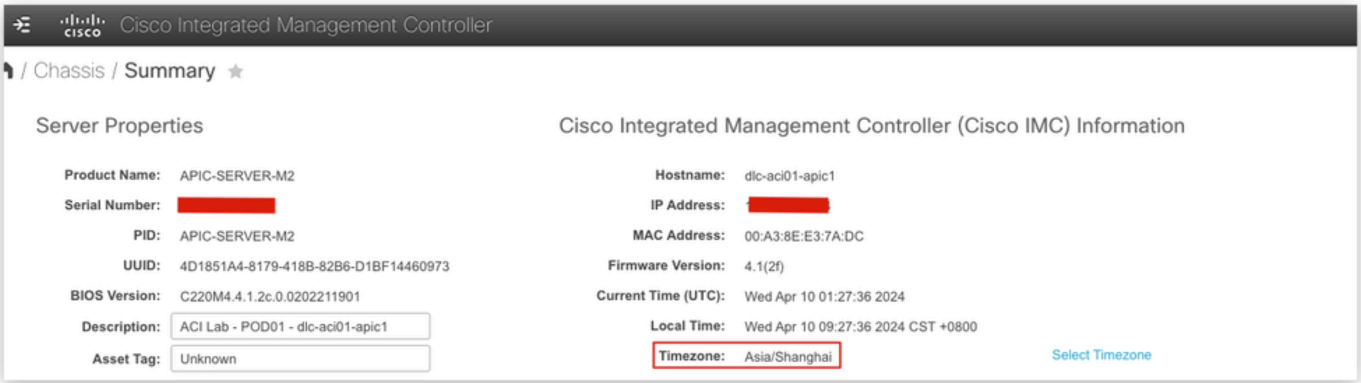
```
<configResolveDn
cookie="85da25da6c/c6f2adca-5d27-ba55-a780-9e33548f595c"
inHierarchical='true'
dn='sys/'/>
```

```
<configResolveDn cookie="85da25da6c/c6f2adca-5d27-ba55-a780-9e33548f595c" response="yes" dn="sys/">
    <outConfig>
        <topSystem dn="sys" address="10.124.145.34" currentTime="Wed Apr 10 01:20:32 2024
" localTime="Wed Apr 10 09:20:32 2024 CST +0800" timeZone="Asia/Shanghai" mode="stand-alone" name="dlc-aci01-apic1" fipsEnable="disabled" ccEnable="disabled" >
            <computeRackUnit rn="rack-unit-1" adminPower="policy" availableMemory="65536" model="APIC-SERVER-M2" memorySpeed="1600" name="APIC-SERVER-M2" numOfAdaptors="2" numOfCo:
" assetTag="Unknown" adaptorSecureUpdate="Enabled" resetComponents="components" storageResetStatus="NA" vicResetStatus="NA" bmcResetStatus="NA" biosPostState="completed" >
                <networkAdapterUnit slot="L" pid="N/A" model="Intel(R) I350 1 Gbps Network Controller" numIntf="2" presence="equipped" rn="network-adapter-L" >
                    <networkAdapterEthIf name="eth-1" id="1" mac="00:a3:8e:e3:7a:e2" rn="eth-1" ></networkAdapterEthIf>
                    <networkAdapterEthIf name="eth-2" id="2" mac="00:a3:8e:e3:7a:e3" rn="eth-2" ></networkAdapterEthIf>
                </networkAdapterUnit>
                <adaptorUnit id="1" model="UCSC-PCIE-CSC-02" pciAddr="128" pciSlot="1" presence="equipped" serial="[redacted]" description="" adminState="policy" cimcManagementEn:
                    <adaptorGenProfile rn="general" pciSlot="1" productName="UCS VIC 1225" model="UCSC-PCIE-CSC-02" serial="[redacted]" revision="6" fipMode="Disabled" vntagMode='
                    <mgmtController rn="mgmt" model="UCSC-PCIE-CSC-02" serial="[redacted]" subject="adaptor" vendor="Cisco Systems Inc" >
                        <firmwareRunning rn="fw-boot-loader" description="Cisco VIC adapter currently running boot loader firmware version" deployment="boot-loader" type="adaptor"
                        <firmwareRunning rn="fw-system" description="Cisco VIC adapter currently running firmware version" deployment="system" type="adaptor" version="4.4(1g)" ></:
                        <firmwareUpdatable rn="fw-updatable" description="Cisco VIC adapter backup firmware version" adminState="triggered" deployment="backup" operState="Fwupdate
                        <firmwareBootDefinition rn="fw-boot-def" type="adaptor" >
                            <firmwareBootUnit rn="bootunit-combined" description="Cisco VIC adapter startup firmware version" adminState="triggered" image="running" resetOnActivat
                        </firmwareBootDefinition>
                    </mgmtController>
                    <adaptorExtEthIf portId="0" ifType="physical" linkState="up" mac="A0:23:9F:AD:F0:49" transport="CE" adminSpeed="10Gbps" operSpeed="10Gbps" rn="ext-eth-0" >
                        <adaptorConnectorInfo rn="connector-info" present="N/A" supported="N/A" type="N/A" vendor="N/A" partNumber="N/A" partRevision="N/A" ></adaptorConnectorInfo:
                        <adaptorLinkTraining rn="link-training" linkTraining="n/a" ></adaptorLinkTraining>
                        <adaptorPortProfiles rn="port-profiles" portProfilesCount="0" portProfilesName="" ></adaptorPortProfiles>
                    </adaptorExtEthIf>
                    <adaptorExtEthIf portId="1" ifType="physical" linkState="up" mac="A0:23:9F:AD:F0:4A" transport="CE" adminSpeed="10Gbps" operSpeed="10Gbps" rn="ext-eth-1" >
                        <adaptorConnectorInfo rn="connector-info" present="N/A" supported="N/A" type="N/A" vendor="N/A" partNumber="N/A" partRevision="N/A" ></adaptorConnectorInfo:
                        <adaptorLinkTraining rn="link-training" linkTraining="n/a" ></adaptorLinkTraining>
```

Copy the Postman execution response to a text editor such as Notepad, PyCharm, or Visual Studio Code for subsequent searching of the class and DN based on MO.

Example 1: Query the class and DN of the timezone

The timezone configured in the current CIMC GUI is "Asia/Shanghai".



Server Properties

Product Name: APIC-SERVER-M2
Serial Number: [redacted]
PID: APIC-SERVER-M2
UUID: 4D1851A4-8179-418B-82B6-D1BF14460973
BIOS Version: C220M4.4.1.2c.0.0202211901
Description: ACI Lab - POD01 - dlc-aci01-apic1
Asset Tag: Unknown

Cisco Integrated Management Controller (Cisco IMC) Information

Hostname: dlc-aci01-apic1
IP Address: [redacted]
MAC Address: 00:A3:8E:E3:7A:DC
Firmware Version: 4.1(2f)
Current Time (UTC): Wed Apr 10 01:27:36 2024
Local Time: Wed Apr 10 09:27:36 2024 CST +0800
Timezone: Asia/Shanghai

Search for **'Asia/Shanghai'** from the results returned by Postman in step 1b. The timezone is **"Asia/Shanghai",** the class is **"topSystem",** and the DN is **"sys/".**

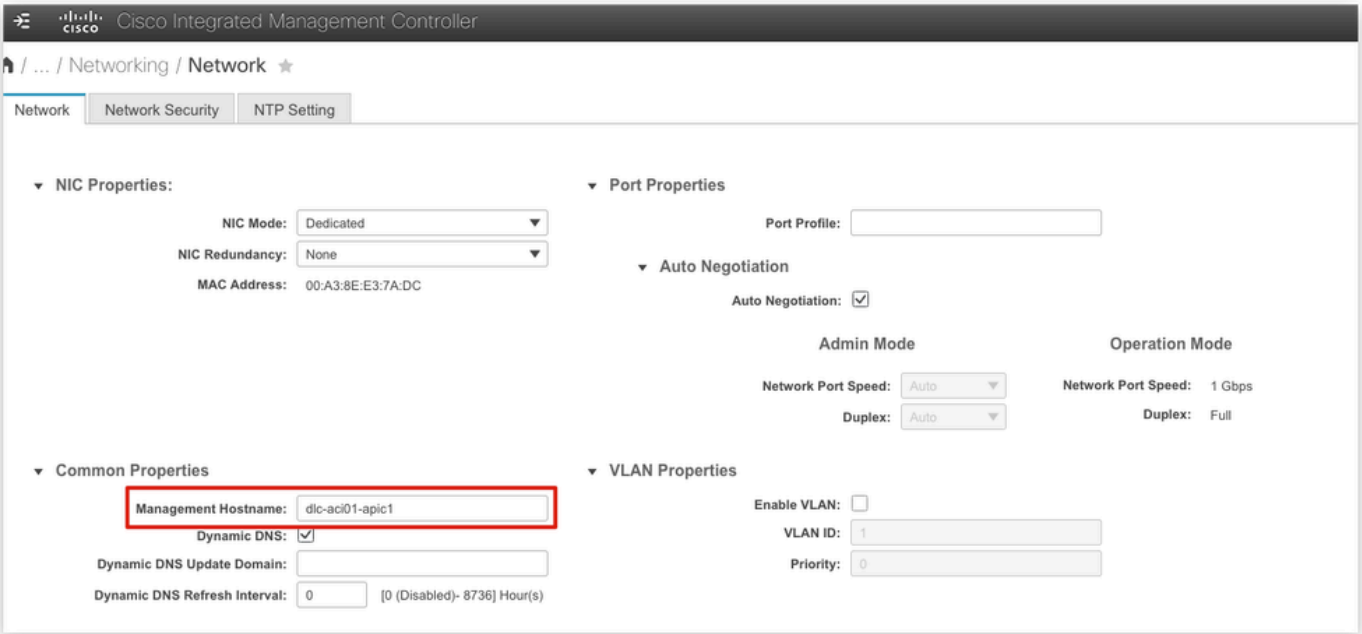
<#root>

```
<configResolveDn cookie="85da25da6c/c6f2adca-5d27-ba55-a780-9e33548f595c" response="yes" dn="sys/">
    <outConfig>
        <topSystem
```

```
dn="sys"

 address="a.b.c.d" currentTime="Wed Apr 10 01:05:12 2024
" localTime="Wed Apr 10 09:05:12 2024 CST +0800"

timeZone="Asia/Shanghai"

mode="stand-alone" name="dlc-aci01-apic1" fipsEnable="disabled" ccEnable="disabled" >
```

Example 2: Query the class and DN of the hostname

The hostname configured in the current CIMC GUI is "dlc-aci01-apic1".



Cisco Integrated Management Controller

/ ... / Networking / Network

Network | Network Security | NTP Setting

▾ NIC Properties:

NIC Mode: Dedicated

NIC Redundancy: None

MAC Address: 00:A3:8E:E3:7A:DC

▾ Port Properties

Port Profile:

▾ Auto Negotiation

Auto Negotiation: ☑

Admin Mode

Network Port Speed: Auto

Duplex: Auto

Operation Mode

Network Port Speed: 1 Gbps

Duplex: Full

▾ Common Properties

Management Hostname: dlc-aci01-apic1

Dynamic DNS: ☑

Dynamic DNS Update Domain:

Dynamic DNS Refresh Interval: 0 [0 (Disabled)- 8736] Hour(s)

▾ VLAN Properties

Enable VLAN: □

VLAN ID: 1

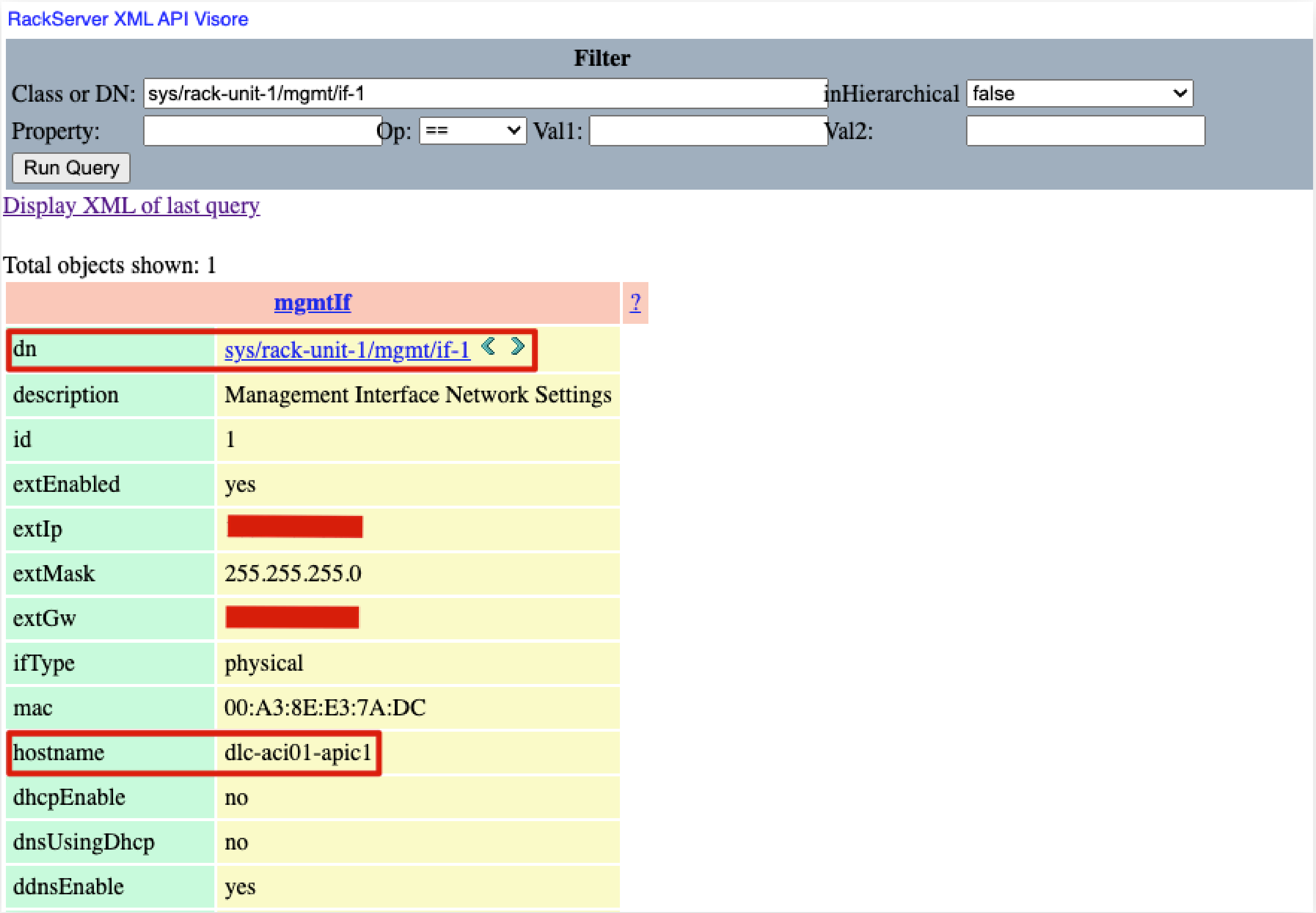
Priority: 0

Search for "dlc-aci01-apic1" in the results returned by Postman. The hostname is "dlc-aci01-apic1", the class is "mgmtIf", and the rn is "if-1".

<#root>

```
<mgmtIf rn="if-1" description="Management Interface Network Settings" id="1" extEnabled="yes" extIp="a.
ifType="physical" mac="00:A3:8E:E3:7A:DC"

hostname="dlc-aci01-apic1"

dhcpEnable="no" dnsUsingDhcp="no" ddnsEnable="yes" ddnsDomain=""
dnsPreferred="a.b.c.z" dnsAlternate="0.0.0.0" ddnsRefreshInterval="0" nicMode="dedicated" vicSlot="0" n
vlanEnable="no" vlanId="1" vlanPriority="0" portProfile="" v6extEnabled="no" v6extIp="::" v6extGw="::" 
v6SlaacIp="::" v6dhcpEnable="no" v6dnsUsingDhcp="no" v6dnsPreferred="::" v6dnsAlternate="::" subject="b
adminNetSpeed="auto" adminDuplex="auto" operNetSpeed="1Gbps" operDuplex="full" >
```

Then, query CIMC visore from https://CIMC_IP/visore.html, the hostname "dlc-aci01-apic1" corresponds to the DN= `sys/rack-unit-1/mgmt/if-1`.



RackServer XML API Visore

Filter

Class or DN: sys/rack-unit-1/mgmt/if-1 inHierarchical false

Property: Op: == Val1: Val2:

Run Query

Display XML of last query

Total objects shown: 1

mgmtIf	
dn	sys/rack-unit-1/mgmt/if-1
description	Management Interface Network Settings
id	1
extEnabled	yes
extIp	[redacted]
extMask	255.255.255.0
extGw	[redacted]
ifType	physical
mac	00:A3:8E:E3:7A:DC
hostname	dlc-aci01-apic1
dhcpEnable	no
dnsUsingDhcp	no
ddnsEnable	yes

2. Managing CIMC via REST API

- In step 1, you have already identified the class and distinguished name (DN) corresponding to the managed object (MO).
- You can use the Ansible **community.general.imc_rest** module to manage CIMC via API. Detail info refers to: imc_rest module Documentation

Use the API method configResolveClass to retrieve Information

configResolveClass: The method retrieves managed object in a given class. If inHierarchical=true, the results contain children. Taking querying firmware version as an example, use API method configResolveClass and specify the classID of the MO.

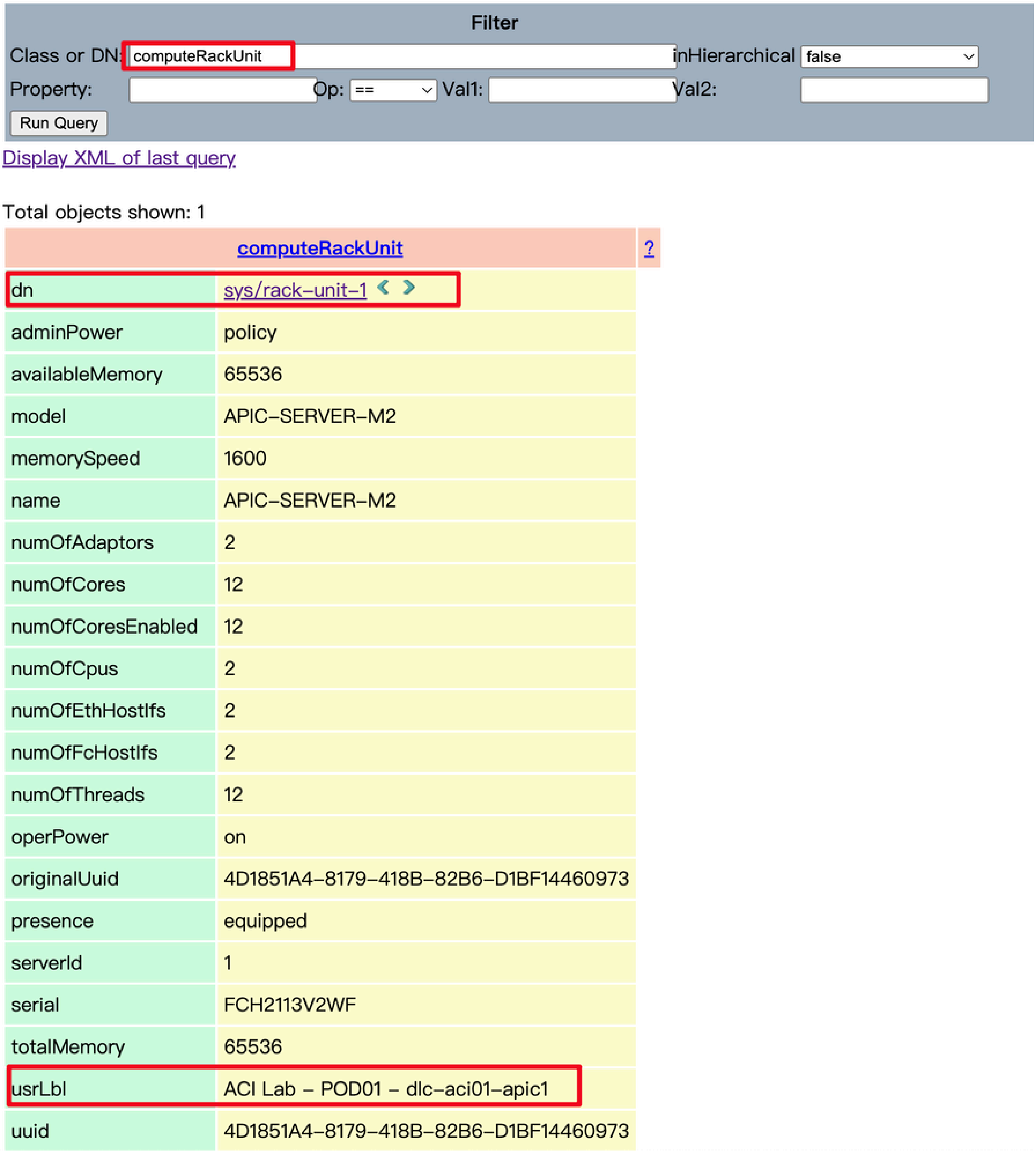
Ansible script content output:

<#root>

```
- name: IMC login and check
  community.general.imc_rest:
    hostname: '{{ imc_hostname }}'
    username: '{{ imc_username }}'
    password: '{{ imc_password }}'
    validate_certs: false  # only do this when you trust the network!
    content: |
      <!-- firmware version -->
  <

configResolveClass

inHierarchical='false'

classId='firmwareRunning'

/>
```

Use API method configConfMo to modify the Configuration

To modify the configuration of MO using the CIMC API, utilize the configConfMo method. This method is designed to configure or modify the settings of a specific MO. When invoking configConfMo, it is important to provide the exact class or DN information of the MO you wish to modify.



Class or DN: computeRackUnit inHierarchical false

Property: Op: == Val1: Val2:

Display XML of last query

Total objects shown: 1

computeRackUnit	?
dn	sys/rack-unit-1
adminPower	policy
availableMemory	65536
model	APIC-SERVER-M2
memorySpeed	1600
name	APIC-SERVER-M2
numOfAdaptors	2
numOfCores	12
numOfCoresEnabled	12
numOfCpus	2
numOfEthHostIfs	2
numOfFcHostIfs	2
numOfThreads	12
operPower	on
originalUuid	4D1851A4-8179-418B-82B6-D1BF14460973
presence	equipped
serverId	1
serial	FCH2113V2WF
totalMemory	65536
usrLbl	ACI Lab - POD01 - dlc-aci01-apic1
uuid	4D1851A4-8179-418B-82B6-D1BF14460973

Ansible script content output:

<#root>

```
  - name: change CIMC description
    community.general.imc_rest:
      hostname: '{{ imc_hostname }}'
      username: '{{ imc_username }}'
      password: '{{ imc_password }}'
      validate_certs: false
      content: |
    <configConfMo><inConfig>
     <
computeRackUnit dn="sys/rack-unit-1" usrLbl="new_lab_CIMC_description"
/>
        </inConfig></configConfMo>examples:
```

3. CIMC Automation Configuration Workflow Example

Cisco APIC is Cisco ACI controller software installed on UCS C220 series. The workflow illustrates the automated process of re-imaging APIC software.

1. Login to CIMC with pre-check
  - Retrieve firmware version
  - Retrieve faults
  - Retrieve TPM status
2. Update CIMC configurations
  - Update management hostname
  - Update Description
  - Update Timezone
  - Update ntp
  - Enable SOL
  - Update CIMC mapping vmedia
  - Update CIMC boot order to CIMC-map
  - Reboot CIMC
3. Ansible run shell expect to monitor installation status and enter iso link for APIC installation speed up
4. Retrieve CIMC post installation status
  - Update CIMC boot order back to HDD
  - Power-on `host`

Ansible module example:

Note: The example only includes content info, full ansible module refers to community.general.imc_rest module from Ansible official website

```
<#root>

- name: Login to CIMC with pre-check
   content: |
     <!-- firmware version -->
     <

configResolveClass

 inHierarchical='false'

classId

='firmwareRunning'/>

      <!-- fault -->
      <

configResolveClass
```

```
 inHierarchical='false'

classId

='faultInst'/>

      <!-- TPM status -->
      <

configResolveClass

 inHierarchical='false'

classId

='equipmentTpm'/>


- name: IMC update CIMC infra info
    content: |
      <!-- Configure hostname -->
      <configConfMo><inConfig>
        <

mgmtIf



dn="sys/rack-unit-1/mgmt/if-1"

 hostname="dlc-aci01-apic1"/>
      </inConfig></configConfMo>

      <!-- Configure description -->
      <configConfMo><inConfig>
        <

computeRackUnit



dn="sys/rack-unit-1"

 usrLbl="ACI Lab - POD01 - dlc-aci01-apic1"/>
      </inConfig></configConfMo>

      <!-- Configure timeZone -->
      <configConfMo><inConfig>
        <

topSystem



dn="sys"

 timeZone="Asia/Shanghai"/>
      </inConfig></configConfMo>

      <!-- Configure NTP -->
      <configConfMo><inConfig>
        <

commNtpProvider



dn="sys/svc-ext/ntp-svc"
```

```
 ntpServer1="ntp.esl.cisco.com"/>
      </inConfig></configConfMo>


- name: Update CIMC configurations
    content: |
      <!-- IMC change boot order to CIMC-map -->
      <configConfMo><inConfig>
        <

lsbootVMedia



dn="sys/rack-unit-1/boot-precision/vm-CIMC-map"

 name="CIMC-map" type="VMEDIA" subtype="cimc-mapped-dvd"  order="1" state="Enabled" />
      </inConfig></configConfMo>

      <!-- CIMC mapping vmedia -->
      <configConfMo><inConfig>
        <

commVMediaMap

 volumeName="ACI-automation" map="www" remoteShare="http://a.b.c.d/Images/ACI/4/4.2/" remoteFile="aci-a

dn="sys/svc-ext/vmedia-svc/vmmap-ACI-automation"

 ></commVMediaMap>
      </inConfig></configConfMo>

      <!-- CIMC hard reset -->
      <configConfMo><inConfig>
        <

computeRackUnit



dn="sys/rack-unit-1"

 adminPower="hard-reset-immediate" />
      </inConfig></configConfMo>

# Ansible run shell expect to monitor installation status and enter iso link for APIC installation spee
- name: copy apic init script to
  template:
    src: "init.sh"
    dest: /tmp/init.sh
  delegate_to: localhost

- name: Make script executable
  file:
    path: /tmp/init.sh
    mode: "+x"
  delegate_to: localhost
  tags:
    - render
    - init

- name: Run the generated script
  command: /tmp/init.sh
  delegate_to: localhost
  changed_when: no
  tags:
    - script

- name: Retrieve CIMC post installation status
    content: |
      <!-- delete exists CIMC mapping -->
      <configConfMo><inConfig>
        <

lsbootVMedia


dn="sys/rack-unit-1/boot-precision/vm-CIMC-map"

 name="CIMC-map" status='removed'/>
      </inConfig></configConfMo>
      <!-- delete exists CIMC mapping vmedia -->
      <configConfMo><inConfig>
        <

commVMediaMap


dn="sys/svc-ext/vmedia-svc/vmmap-ACI-automation"

 volumeName="ACI-automation" status='removed' ></commVMediaMap>
      </inConfig></configConfMo>
      <!-- Reconfigure IMC to boot from storage -->
      <configConfMo><inConfig>
        <

lsbootStorage


dn="sys/rack-unit-1/boot-policy/storage-read-write"

 access="read-write" order="1" type="storage"/>
      </inConfig></configConfMo>

      <!-- CIMC power up -->
      <configConfMo><inConfig>
        <

computeRackUnit dn="sys/rack-unit-1"

 adminPower="up" />
      </inConfig></configConfMo>
  delegate_to: localhost
  tags:
    - retrieve_CIMC_status
```

Related Information

Cisco UCS Rack-Mount Servers Cisco IMC XML API Programmer's Guide

community.general.imc_rest module – Manage Cisco IMC hardware through its REST API

UCS Manager Information Model Reference