Ansible Rest API 모듈을 통해 CIMC 구성

목차

소개

이 문서에서는 Ansible REST API 모듈을 통해 CIMC(Cisco Integrated Management Controller)를 구성하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- UCS CIMC
- API
- 앤서블

사용되는 구성 요소

- UCS C220-M4, 버전 4.1(2f)
- postman 및 ansible 버전 2.14.5를 실행하는 클라이언트

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든

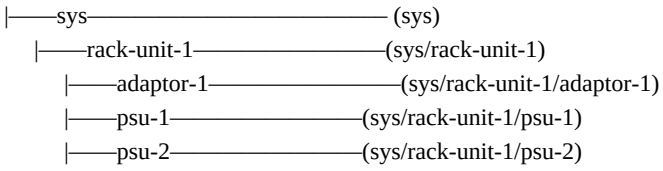
명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

CIMC API 개요

Cisco UCS를 구성하는 모든 물리적 및 논리적 구성 요소는 MIM(Hierarchical Management Information Model)(MIT라고도 함)으로 표현됩니다. 트리의 각 노드는 관리 상태와 작동 상태를 포함하는 MO(Managed Object) 또는 객체 그룹을 나타냅니다.

계층 구조는 최상위(sys)에서 시작하며 상위 및 하위 노드를 포함합니다. 이 트리의 각 노드는 관리 객체이며 Cisco UCS의 각 객체에는 트리에서의 객체 및 위치를 설명하는 고유한 DN(식별 이름)이 있습니다. 관리 객체는 CPU, DIMM, 어댑터 카드, 팬 및 전원 공급 장치와 같은 Cisco UCS 리소스의 추상화입니다.

CIMC MIM 구조의 그림:



개체 이름 지정:

- DN: DN을 사용하면 대상 객체를 명확하게 식별할 수 있습니다.
- RN: 상대 이름은 상위 객체의 컨텍스트 내에서 객체를 식별합니다.

예를 들어, 이 DN은 다음과 같습니다.

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

는 4개의 상대 이름으로 구성됩니다.

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

이 문서에 사용된 API:

- 인증: aaaLogin. 초기 로그인 방법입니다. aaaLogin 메서드를 사용하여 유효한 쿠키를 가져옵니다.

- Query: configResolveDn. DN으로 개체를 검색합니다.
- 설정: configConfMo. configConfMo 메서드는 MO(Managed Object)에서 하나 이상의 속성을 구성하는 데 사용됩니다. 구성할 MO는 DN(Distinguished Name)으로 고유하게 식별됩니다.

참고:

많은 쿼리 메서드에는 부울 값(true/yes 또는 false/no)을 받는 inHierarchical 인수가 포함됩니다. 이 인수를 true로 설정하면 메서드가 계층 구조 내의 모든 자식 개체를 반환합니다.

설정

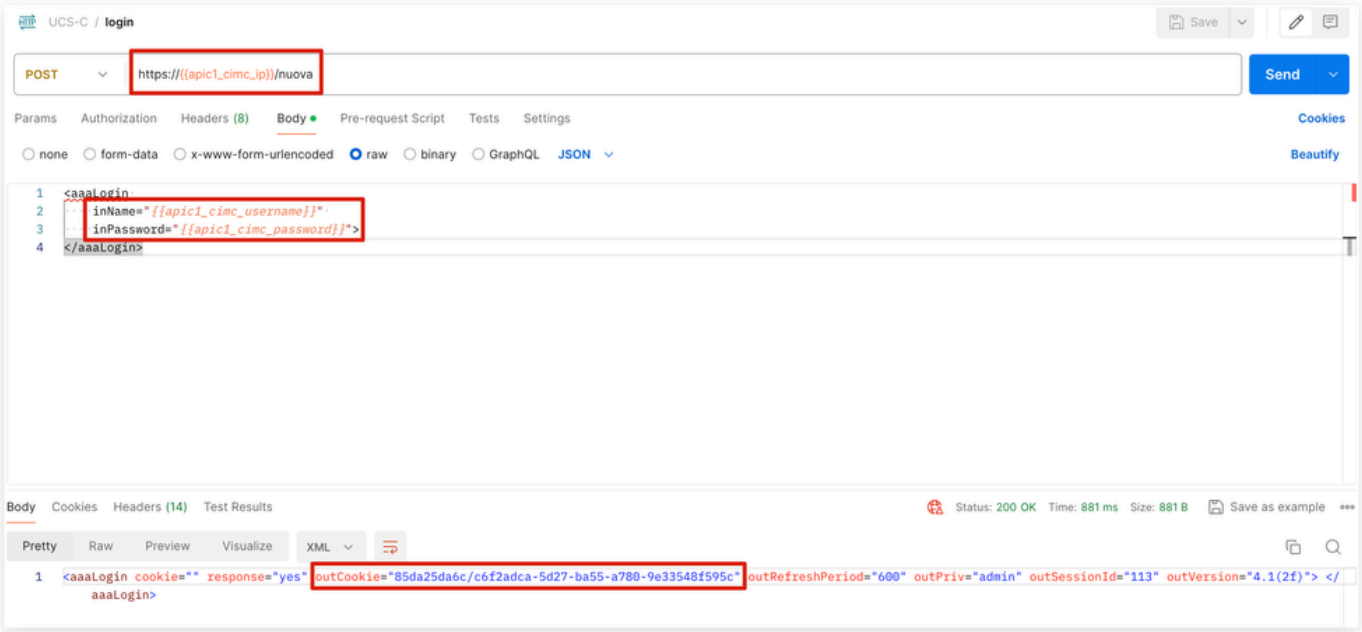
1. CIMC MO(Managed Object)의 클래스 또는 DN 찾기

API를 통해 CIMC 컨피그레이션을 자동화하려면 구성하려는 MO와 관련된 특정 클래스 또는 DN(Distinguished Name) 정보를 결정해야 합니다.

1a. API를 사용하여 CIMC에 로그인하고 쿠키 정보를 검색합니다

https://{{apic_cimc_ip}}/nuova로 POST 요청을 보내고 aaaLogin 메서드를 지정합니다. 사용자 이름과 비밀번호를 입력합니다.

API 응답에서 쿠키를 복사합니다.



또는 curl을 사용하여 쿠키 정보를 가져올 수 있습니다.

curl -k -d "

" https://apic_cimc_ip/nuova

10억 API 쿼리 메서드 configResolveDn을 사용하여 모든 MO(Managed Object) 정보 검색

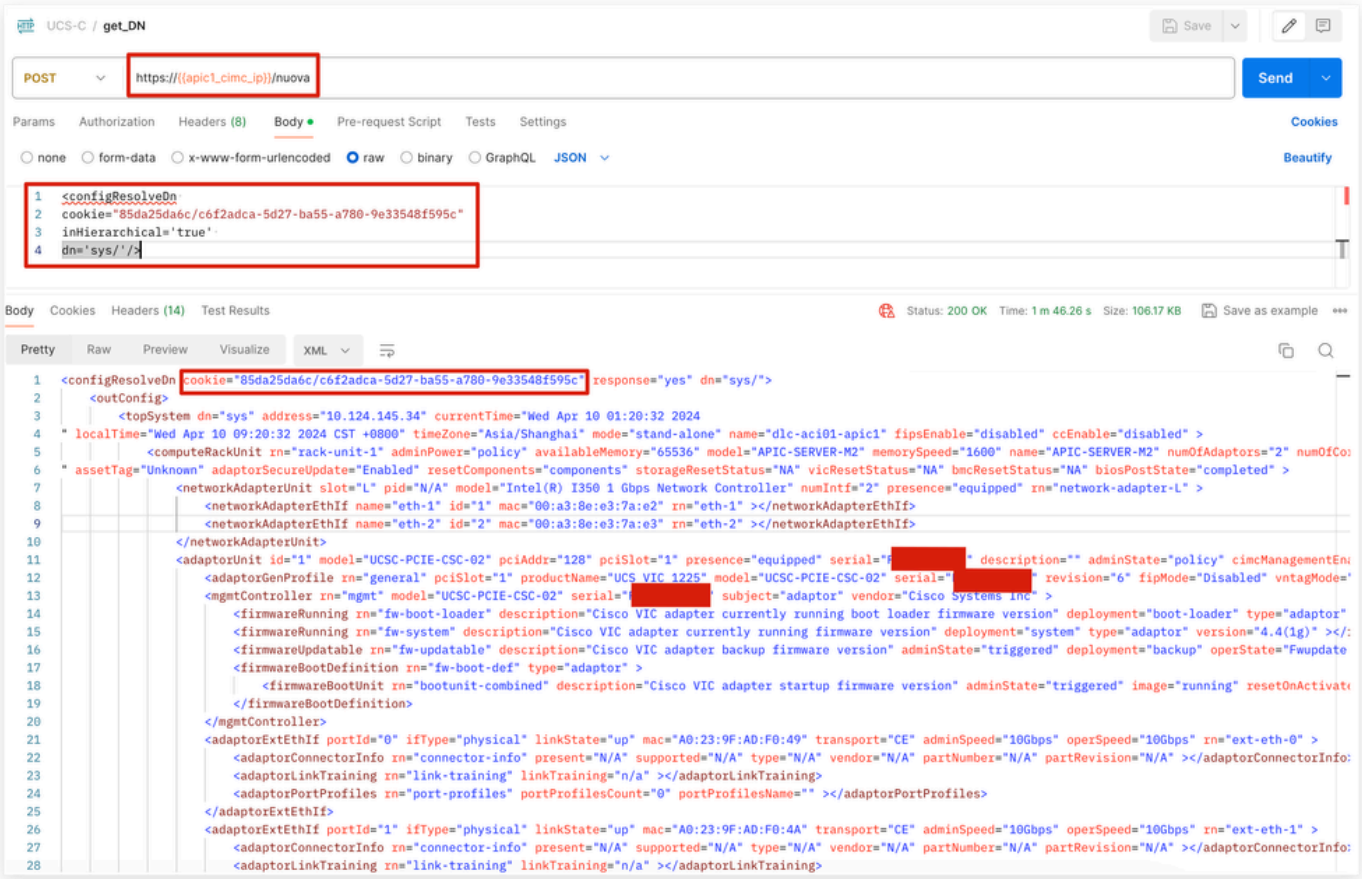
inHierarchical="true" 및 dn="sys/"와 함께 configResolveDn을 사용하는 경우 CIMC에서 모든 MO(Managed Object) 정보를 검색합니다.

configResolveDn: configResolveDn 메서드는 지정된 DN에 대한 단일 관리 개체를 검색합니다.

inHierarchical=true: true로 설정하면 모든 자식 노드 정보가 반환됩니다. 이 조합을 통해 CIMC에서 모든 노드 및 하위 노드 MO 정보를 가져올 수 있습니다.

dn= " sys/ ": 이것이 MIT의 최상위 루트입니다.

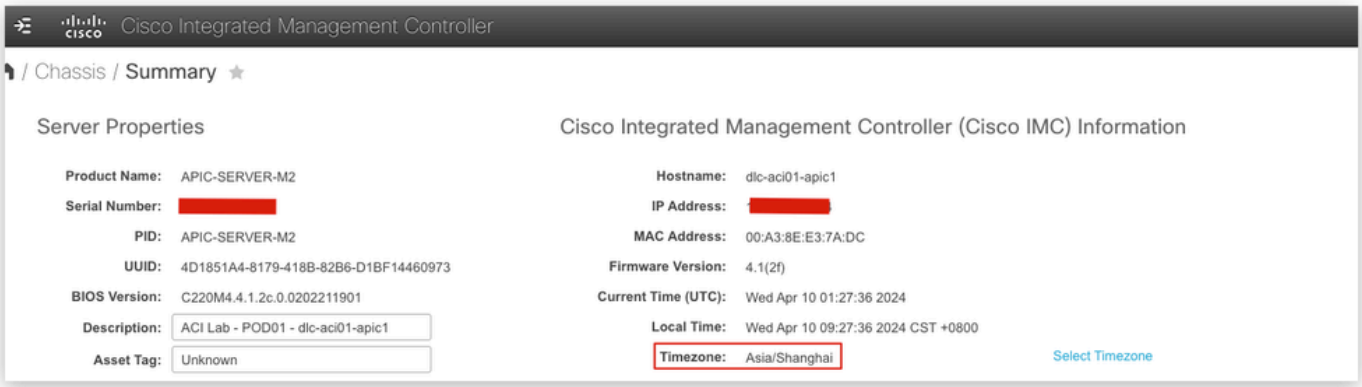
API 응답:



Postman 실행 응답을 Notepad, PyCharm 또는 Visual Studio Code와 같은 텍스트 편집기에 복사하여 MO를 기반으로 클래스 및 DN을 후속 검색합니다.

예 1: 표준 시간대의 클래스 및 DN 쿼리

현재 CIMC GUI에 구성된 시간대는 "Asia/Shanghai"입니다.



1b단계에서 Postman이 반환한 결과에서 'Asia/Shanghai'를 검색하세요. 표준 시간대는 "Asia/Shanghai"이고 클래스는 "topSystem"이며 DN은 "sys/"입니다.

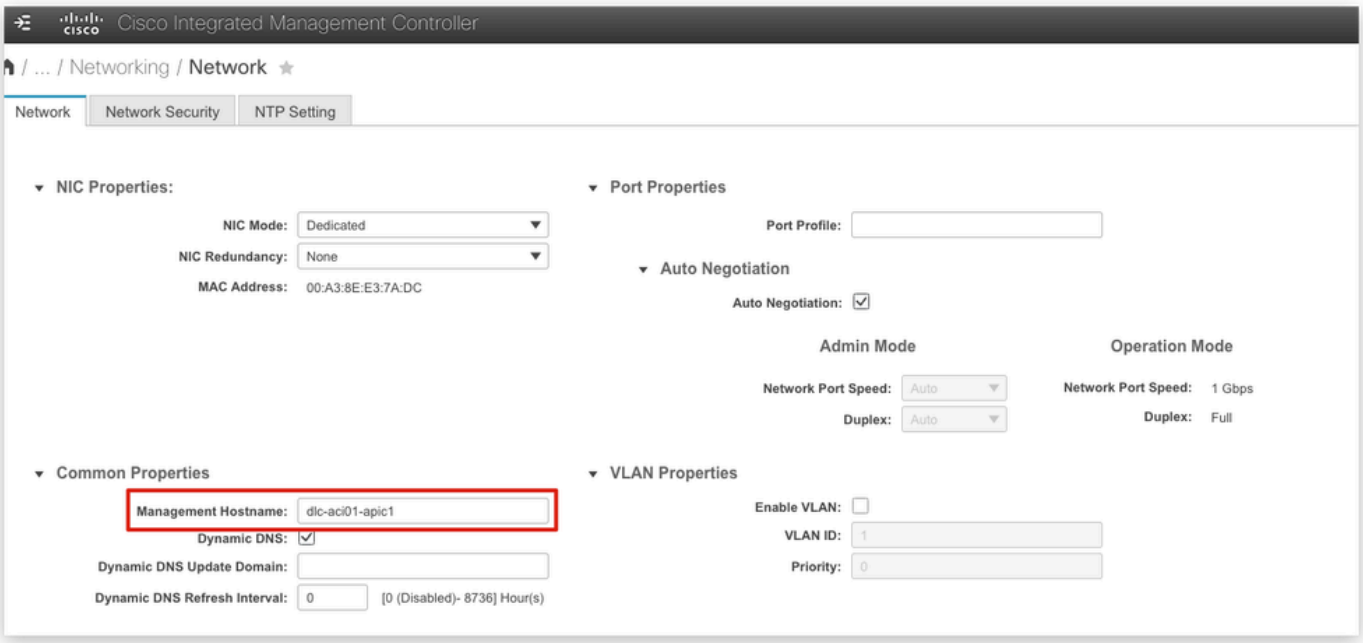
<#root>

```
<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" >
```

예 2: 호스트 이름의 클래스 및 DN 쿼리

현재 CIMC GUI에 구성된 호스트 이름은 "dlc-aci01-apic1"입니다.

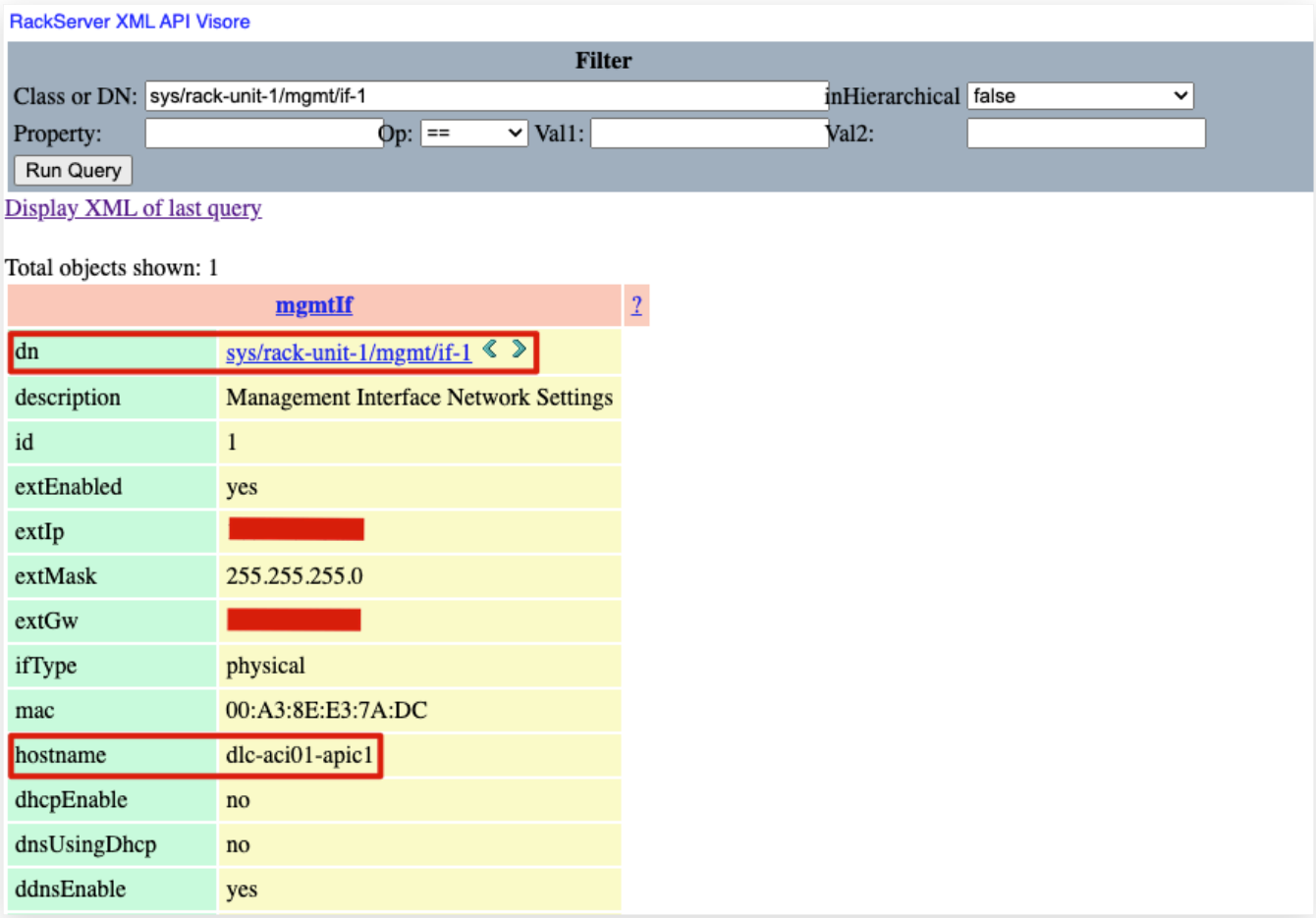


Postman에서 반환된 결과에서 "dlc-aci01-apic1"을 검색합니다. 호스트 이름은 "dlc-aci01-apic1"이고, 클래스는 "mgmtIf"이며, rn은 "if-1"입니다.

<#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" >
```

그런 다음 https://CIMC_IP/visore.html에서 CIMC visore를 쿼리합니다. 호스트 이름 "dlc-aci01-apic1"은 DN= `sys/rack-unit-1/mgmt/if-1`에 해당합니다.



2. REST API를 통한 CIMC 관리

- 1단계에서는 MO(Managed Object)에 해당하는 클래스 및 DN(Distinguished Name)을 이미 확인했습니다.
- Ansible community.general.imc_rest 모듈을 사용하여 API를 통해 CIMC를 관리할 수 있습니다. 세부 정보 참조: imc_rest_module 설명서

API 메서드 configResolveClass를 사용하여 정보 검색

구성 확인 클래스: 이 메서드는 지정된 클래스에서 관리되는 개체를 검색합니다. inHierarchical=true이면 결과에 하위 항목이 포함됩니다. 펌웨어 버전 쿼리를 예로 들면 API 메서드 configResolveClass를 사용하고 MO의 classID를 지정합니다.

실행 가능한 스크립트 콘텐츠 출력:

<#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: |

<

configResolveClass

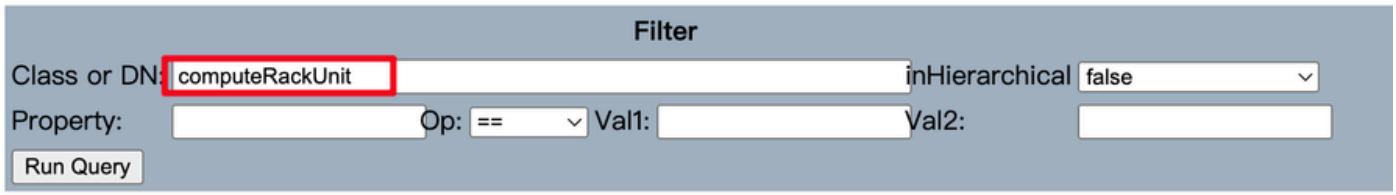
inHierarchical='false'

classId='firmwareRunning'

/>

API 메서드 configConfMo를 사용하여 컨피그레이션을 수정합니다.

CIMC API를 사용하여 MO 컨피그레이션을 수정하려면 configConfMo 메서드를 사용합니다. 이 방법은 특정 MO의 설정을 구성하거나 수정할 수 있도록 설계되었습니다. configConfMo를 호출할 때는 수정하려는 MO의 정확한 클래스 또는 DN 정보를 제공해야 합니다.



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

실행 가능한 스크립트 콘텐츠 출력:

<#root>

```
- name: change CIMC description
  community.general.imc_rest:
    hostname: '{{ imc_hostname }}'
    username: '{{ imc_username }}'
    password: '{{ imc_password }}'
    validate_certs: false
```

content: |

<

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

/>

examples:

3. CIMC 자동화 구성 워크플로 예

Cisco APIC는 UCS C220 Series에 설치된 Cisco ACI 컨트롤러 소프트웨어입니다. 이 워크플로에서는 APIC 소프트웨어 재이미지화의 자동화된 프로세스를 보여줍니다.

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 모듈의 예:

참고: 이 예에는 콘텐츠 정보만 포함되며, 전체 가능 모듈은 Ansible 공식 웹 사이트의 community.general.imc_rest 모듈을 참조합니다

<#root>

```
- name: Login to CIMC with pre-check
   content: |
```

<

configResolveClass

inHierarchical='false'

classId

='firmwareRunning'/>

<

configResolveClass

inHierarchical='false'

classId

='faultInst'/>

<

configResolveClass

inHierarchical='false'

classId

='equipmentTpm'/>

```
- name: IMC update CIMC infra info
    content: |
```

<

mgmtIf

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

hostname="dlc-aci01-apic1"/>

<

computeRackUnit

dn="sys/rack-unit-1"

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

<

topSystem

dn="sys"

timeZone="Asia/Shanghai"/>

<

commNtpProvider

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

ntpServer1="ntp.esl.cisco.com"/>

```
- name: Update CIMC configurations
    content: |
```

```
        <
```

lsbootVMedia

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

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

```
        <
```

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"

 >

        <

computeRackUnit

dn="sys/rack-unit-1"

 adminPower="hard-reset-immediate" />

```
# 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: |
```

```
        <
```

lsbootVMedia

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

```
 name="CIMC-map" status='removed'/>
```

```
        <
```

commVMediaMap

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

volumeName="ACI-automation" status='removed' >

<

lsbootStorage

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

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

<

```
computeRackUnit dn="sys/rack-unit-1"
 adminPower="up" />
```

```
  delegate_to: localhost
  tags:
    - retrieve_CIMC_status
```

관련 정보

Cisco UCS 랙 마운트 서버 Cisco IMC XML API 프로그래머 가이드

community.general.imc_rest 모듈 - REST API를 통해 Cisco IMC 하드웨어 관리

UCS Manager 정보 모델 참조

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.