

Configuration de CIMC via le module d'API Ansible Rest

Table des matières

[Introduction](#)

[Conditions préalables](#)

[Exigences](#)

[Composants utilisés](#)

[Présentation de l'API CIMC](#)

[Configurations](#)

[1. Recherchez la classe ou le DN de l'objet géré CIMC \(MO\)](#)

[1 bis. Utilisez l'API pour vous connecter au CIMC et récupérer les informations relatives aux cookies](#)

[1 ter. Utilisation de la méthode de requête API configResolveDn pour récupérer toutes les informations d'objet géré \(MO\)](#)

[Exemple 1 : Interroger la classe et le DN du fuseau horaire](#)

[Exemple 2 : Recherchez la classe et le DN du nom d'hôte](#)

[2. Gestion de CIMC via l'API REST](#)

[Utilisez la méthode API configResolveClass pour récupérer des informations](#)

[Utilisez la méthode API configConfMo pour modifier la configuration](#)

[3. Exemple de workflow de configuration d'automatisation CIMC](#)

[Informations connexes](#)

Introduction

Ce document décrit comment configurer le contrôleur de gestion intégré Cisco (CIMC) via le module API Ansible REST.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- UCS CIMC
- API
- Ansible

Composants utilisés

- UCS C220-M4, version 4.1(2f)
- Un client exécutant postman et Ansible version 2.14.5

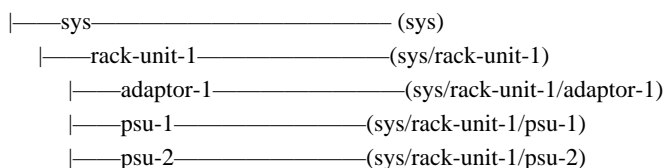
The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

Présentation de l'API CIMC

Tous les composants physiques et logiques de Cisco UCS sont représentés dans un modèle d'informations de gestion hiérarchique (MIM), également appelé MIT. Chaque noeud de l'arborescence représente un objet géré (MO) ou un groupe d'objets qui contient son état administratif et son état opérationnel.

La structure hiérarchique commence au sommet (sys) et contient les noeuds parent et enfant. Chaque noeud de cette arborescence est un objet géré et chaque objet de Cisco UCS possède un nom distinctif (DN) unique qui décrit l'objet et sa place dans l'arborescence. Les objets gérés sont des abstractions des ressources Cisco UCS, telles que les processeurs, les barrettes DIMM, les cartes d'extension, les ventilateurs et les blocs d'alimentation.

Illustration de la structure MIM du CIMC :



Nom d'objet :

- DN : Le nom distinctif vous permet d'identifier sans ambiguïté un objet cible.
- RN : Le nom relatif identifie un objet dans le contexte de son objet parent.

Par exemple, ce nom distinctif :

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

Se compose de 4 noms relatifs :

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

API utilisée dans cet article :

- Authentification: aaaLogin. Méthode initiale de connexion. Utilisez la méthode aaaLogin pour obtenir un cookie valide.
- Requête: configResolveDn. Récupère les objets par DN.
- Configuration: configConfMo. La méthode configConfMo est utilisée pour configurer une ou plusieurs propriétés dans un objet géré (MO). L'organisation géographique à configurer est identifiée de manière unique par un nom distinctif (DN).



Remarque :

De nombreuses méthodes de requête incluent un argument inHierarchical qui accepte une valeur booléenne (true/yes ou false/no). Lorsque cet argument a la valeur true, la méthode renvoie tous les objets enfants de la hiérarchie.

Configurations

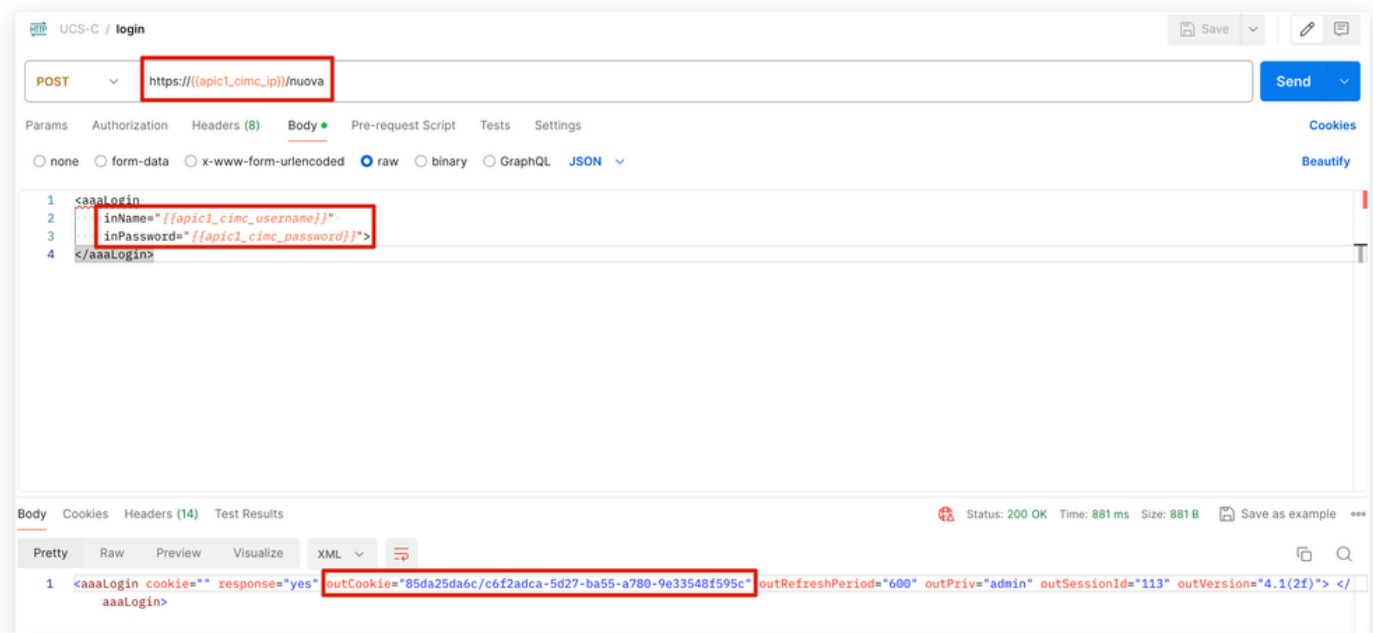
1. Recherchez la classe ou le DN de l'objet géré CIMC

Pour automatiser la configuration de CIMC via son API, il est essentiel de déterminer la classe spécifique ou les informations de nom distinctif (DN) associées aux objets gérés (MO) que vous souhaitez configurer.

1 bis. Utilisez l'API pour vous connecter au CIMC et récupérer les informations relatives aux cookies

Envoyez une demande POST à https://{{apic_cimc_ip}}/nuova et spécifiez la méthode aaaLogin. Entrez un nom d'utilisateur et un mot de passe.

Copiez le cookie de la réponse de l'API.



Vous pouvez également utiliser curl pour récupérer les informations relatives aux cookies.

```
curl -k -d "
```

```
" https://apic_cimc_ip/nuova
```

1 ter. Utilisation de la méthode de requête API `configResolveDn` pour récupérer toutes les informations d'objet géré (MO)

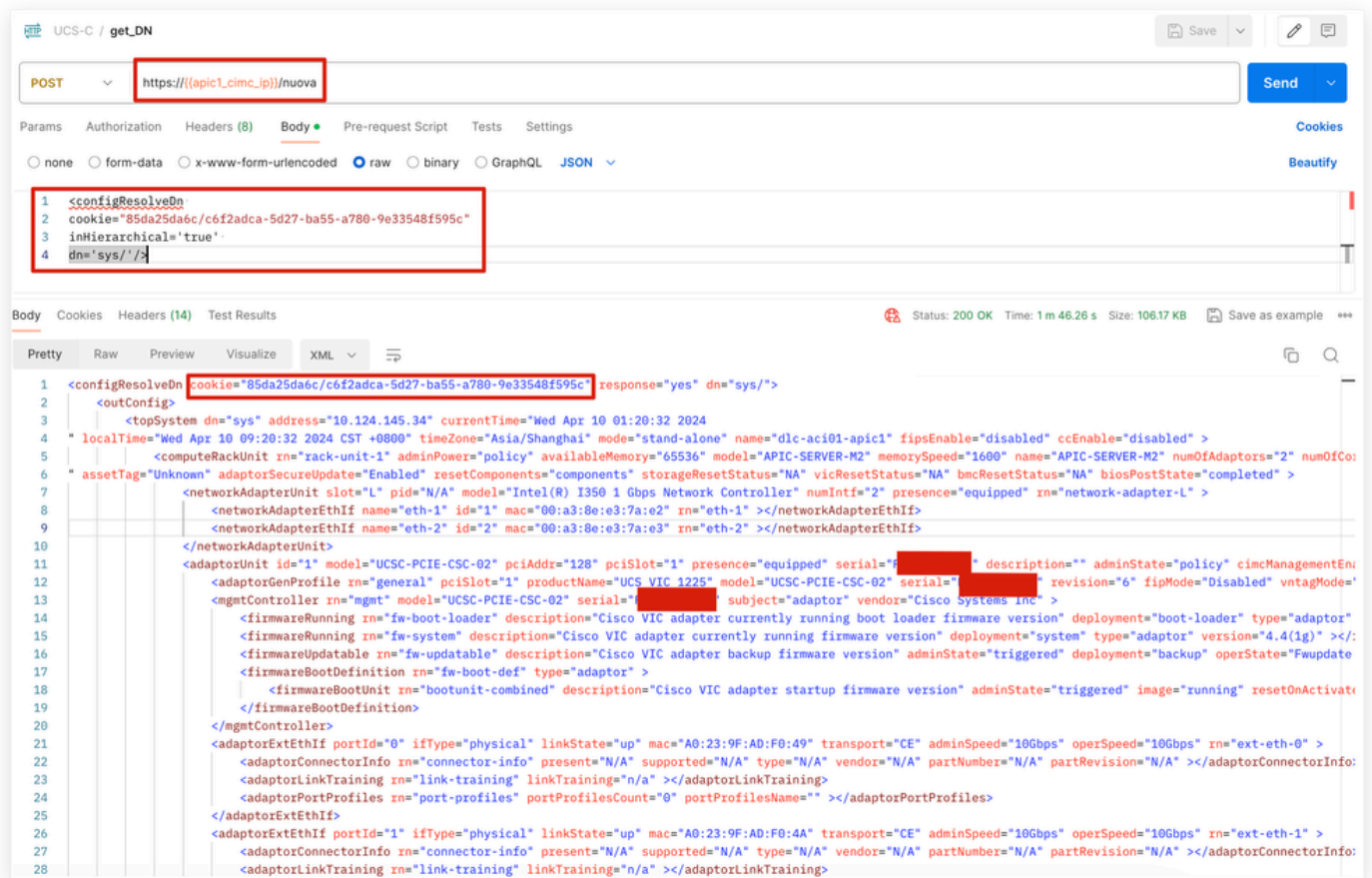
Lors de l'utilisation de `configResolveDn` avec `inHierarchical="true"` et `dn="sys/"`, il récupère toutes les informations d'objet géré (MO) à partir de CIMC.

`configResolveDn` : La méthode `configResolveDn` récupère un seul objet géré pour un nom unique spécifié.

`inHierarchical=true` : Lorsque cette propriété a la valeur `true`, elle renvoie toutes les informations de noeud enfant. Cette combinaison permet d'extraire toutes les informations MO des noeuds et sous-noeuds de CIMC.

`dn="sys/"`: C'est la racine supérieure du MIT.

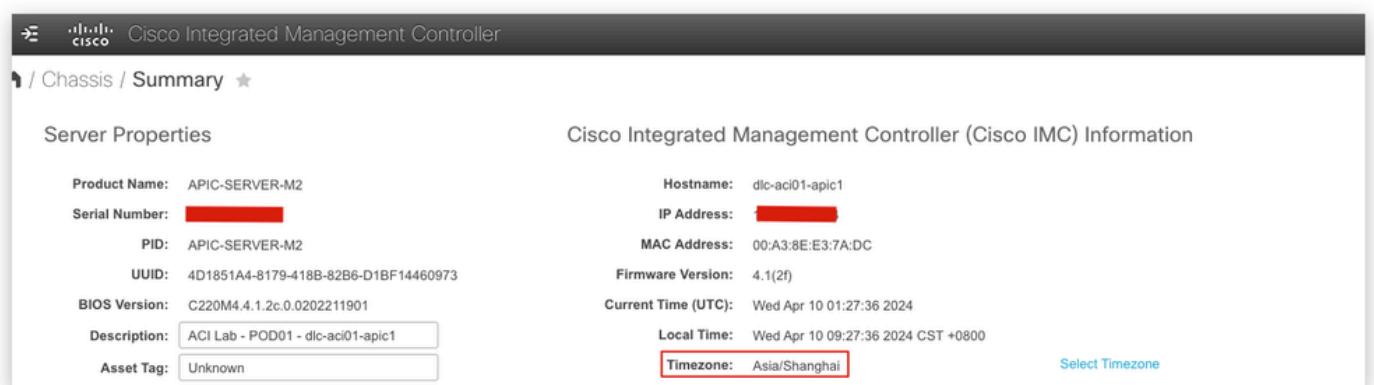
Réponse de l'API :



Copiez la réponse d'exécution de Postman dans un éditeur de texte tel que Notepad, PyCharm ou Visual Studio Code pour une recherche ultérieure de la classe et du DN basée sur MO.

Exemple 1 : Interroger la classe et le DN du fuseau horaire

Le fuseau horaire configuré dans l'interface graphique actuelle du CIMC est « Asia/Shanghai ».



Recherchez 'Asia/Shanghai' dans les résultats renvoyés par Postman à l'étape 1b. Le fuseau horaire est "Asia/Shanghai", la classe est "topSystem", et le DN est "sys/".

<#root>

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

<topSystem

dn="sys"

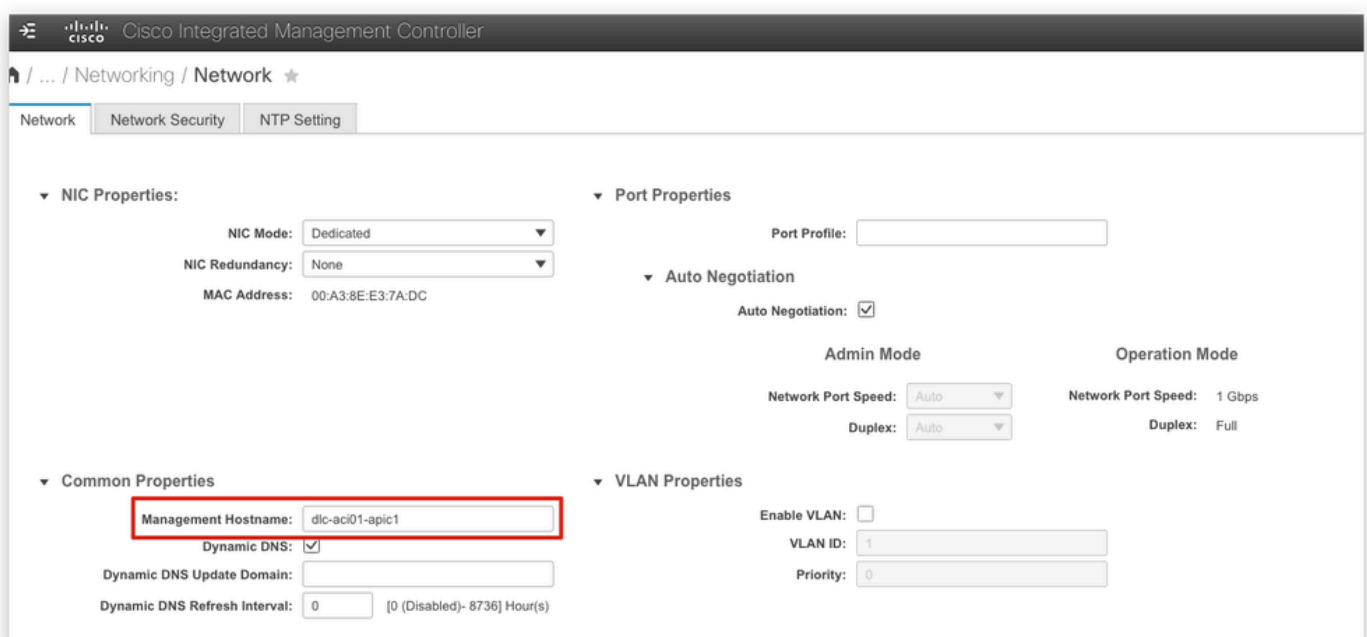
address="a.b.c.d" currentTime="Wed Apr 10 01:05:12 2024"
" localTime="Wed Apr 10 09:05:12 2024 CST +0800"

timeZone="Asia/Shanghai"

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

Exemple 2 : Recherchez la classe et le DN du nom d'hôte

Le nom d'hôte configuré dans l'interface utilisateur graphique actuelle du CIMC est « dlc-aci01-apic1 ».



Recherchez "dlc-aci01-apic1" dans les résultats renvoyés par Postman. Le nom d'hôte est « dlc-aci01-apic1 », la classe est « mgmtIf » et le nom d'exécution est « 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=":" v
v6SlaacIp=":" v6dhcpEnable="no" v6dnsUsingDhcp="no" v6dnsPreferred=":" v6dnsAlternate=":" subject="b
adminNetSpeed="auto" adminDuplex="auto" operNetSpeed="1Gbps" operDuplex="full" >

Ensuite, interrogez CIMC visore à partir de https://CIMC_IP/visore.html, le nom d'hôte « dlc-aci01-

apic1 » correspond au DN= `sys/rack-unit-1/mgmt/if-1`.

RackServer XML API Visore

Filter

Class or DN: inHierarchical
Property: Op: Val1: Val2:

[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 | | |
| extMask | 255.255.255.0 | |
| extGw | | |
| ifType | physical | |
| mac | 00:A3:8E:E3:7A:DC | |
| hostname | dlc-aci01-apic1 | |
| dhcpEnable | no | |
| dnsUsingDhcp | no | |
| ddnsEnable | yes | |

2. Gestion de CIMC via l'API REST

- À l'étape 1, vous avez déjà identifié la classe et le nom distinctif (DN) correspondant à l'objet géré (MO).
- Vous pouvez utiliser le module Ansible `community.general.imc_rest` pour gérer CIMC via l'API. Les informations détaillées se rapportent à : [imc_rest module Documentation](#)

Utilisez la méthode API `configResolveClass` pour récupérer des informations

`configResolveClass` : La méthode récupère l'objet managé dans une classe donnée. Si `inHierarchical=true`, les résultats contiennent des enfants. En prenant la version du microprogramme comme exemple, utilisez la méthode API `configResolveClass` et spécifiez l'ID de classe du MO.

Sortie de contenu de script réutilisable :

```
<#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'
```

```
/>
```

Utilisez la méthode API `configConfMo` pour modifier la configuration

Pour modifier la configuration de MO à l'aide de l'API CIMC, utilisez la méthode `configConfMo`. Cette méthode est conçue pour configurer ou modifier les paramètres d'un mode de gestion spécifique. Lors de l'appel de `configConfMo`, il est important de fournir les informations exactes de classe ou de DN du MO que vous souhaitez modifier.

Filter

Class or DN: inHierarchical

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 |

Sortie de contenu de script réutilisable :

<#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. Exemple de workflow de configuration d'automatisation CIMC

Le contrôleur Cisco APIC est un logiciel de contrôleur Cisco ACI installé sur la gamme UCS C220. Le workflow illustre le processus automatisé de recréation d'image du logiciel APIC.

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

Exemple de module réutilisable :



Remarque : L'exemple inclut uniquement des informations de contenu, le module ansible complet fait référence au module `community.general.imc_rest` du site officiel Ansible

<#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.es1.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 speed
```

```
- 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/vmmmap-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
```

Informations connexes

[Serveurs rack Cisco UCS Guide du programmeur de l'API XML Cisco IMC](#)

[community.general.imc_rest module - Gestion du matériel Cisco IMC via son API REST](#)

[Référence du modèle d'informations UCS Manager](#)

À propos de cette traduction

Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.