

Configuración de CIMC a través del módulo API Ansible Rest

Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Descripción general de CIMC API](#)

[Configuraciones](#)

[1. Busque la clase o DN del objeto administrado \(MO\) CIMC](#)

[1 bis. Utilice la API para iniciar sesión en el CIMC y recuperar la información de cookies](#)

[1 ter. Uso del método de consulta de API configResolveDn para recuperar toda la información de objetos administrados \(MO\)](#)

[Ejemplo 1: Consulte la clase y el DN de la zona horaria](#)

[Ejemplo 2: Consulte la clase y el DN del nombre de host](#)

[2. Administración de CIMC a través de API REST](#)

[Utilice el método API configResolveClass para recuperar información](#)

[Utilice el método API configConfMo para modificar la configuración](#)

[3. Ejemplo de flujo de trabajo de configuración de automatización CIMC](#)

[Información Relacionada](#)

Introducción

Este documento describe cómo configurar Cisco Integrated Management Controller (CIMC) a través del módulo API Ansible REST.

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

- UCS CIMC
- API
- Ansible

Componentes Utilizados

- UCS C220-M4, versión 4.1(2f)
- Un cliente que ejecuta postman y ansible versión 2.14.5

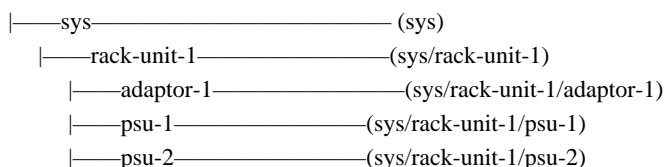
La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Descripción general de CIMC API

Todos los componentes físicos y lógicos que componen Cisco UCS están representados en un modelo de información de gestión jerárquica (MIM), también denominado MIT. Cada nodo del árbol representa un objeto administrado (MO) o un grupo de objetos que contiene su estado administrativo y su estado operativo.

La estructura jerárquica comienza en la parte superior (sys) y contiene los nodos primarios y secundarios. Cada nodo de este árbol es un objeto administrado y cada objeto de Cisco UCS tiene un nombre distinguido (DN) único que describe el objeto y su ubicación en el árbol. Los objetos administrados son abstracciones de los recursos de Cisco UCS, como CPU, DIMM, tarjetas adaptadoras, ventiladores y unidades de fuentes de alimentación.

Ilustración de la estructura CIMC MIM:



Nomenclatura de Objetos:

- DN: El nombre distintivo permite identificar de forma inequívoca un objeto de destino.
- RN: El nombre relativo identifica un objeto dentro del contexto de su objeto principal.

Por ejemplo, este nombre distinguido:

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

Se compone de 4 nombres relativos:

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

API utilizada en este artículo:

- Autenticación: `aaaLogin`. Método inicial para iniciar sesión. Utilice el método `aaaLogin` para obtener una cookie válida.
- Consulta: `configResolveDn`. Recupera objetos por DN.
- Configuración: `configConfMo`. El método `configConfMo` se utiliza para configurar una o más propiedades en un objeto administrado (MO). El MO que se va a configurar se identifica de forma única mediante un nombre distinguido (DN).



Nota:

Muchos métodos de consulta incluyen un argumento `inHierarchical` que acepta un valor booleano (`true/yes` o `false/no`). Cuando se establece en `true`, este argumento hace que el método devuelva todos los objetos secundarios de la jerarquía.

Configuraciones

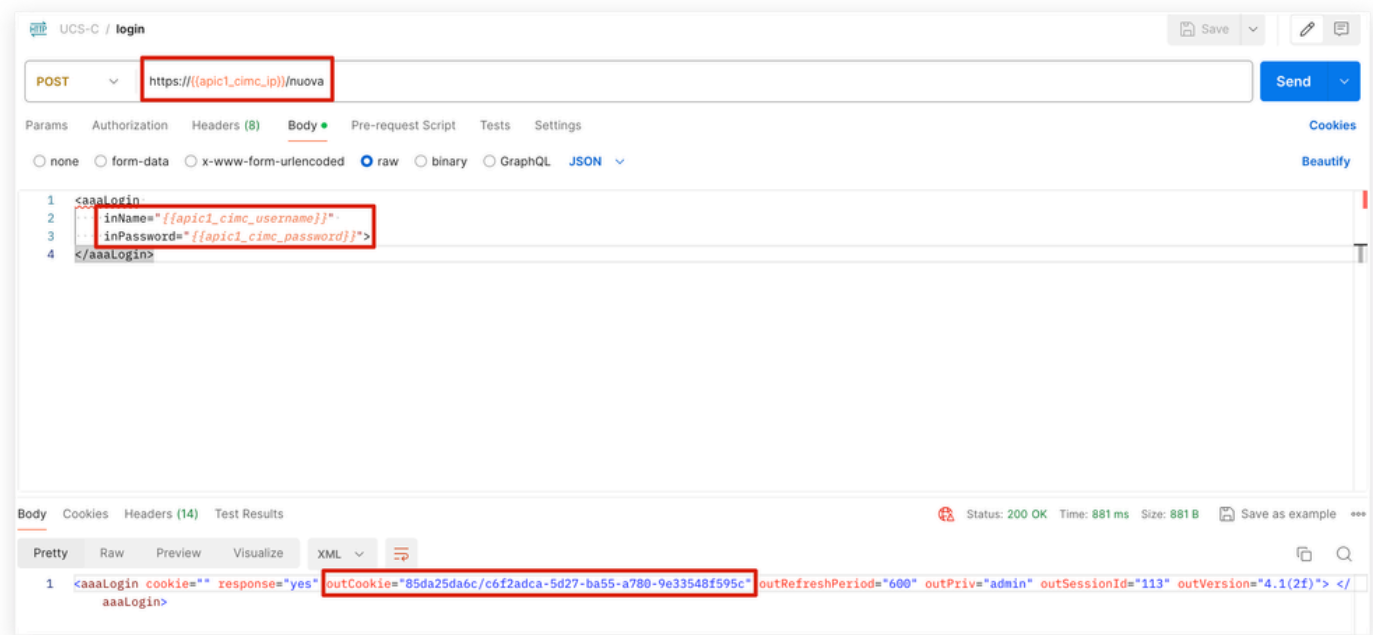
1. Busque la clase o DN del objeto administrado (MO) de CIMC

Para automatizar la configuración de CIMC a través de su API, es fundamental determinar la clase específica o la información de nombre distinguido (DN) asociada a los objetos administrados (MO) que desea configurar.

1 bis. Utilice la API para iniciar sesión en el CIMC y recuperar la información de cookies

Envíe una solicitud POST a https://{{apic_cimc_ip}}/nuova y especifique el método `aaaLogin`. Introduzca un nombre de usuario y una contraseña.

Copie la cookie de la respuesta API.



Alternativamente, puede utilizar curl para obtener la información de la cookie.

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

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

1 ter. Uso del método de consulta de API `configResolveDn` para recuperar toda la información de objetos administrados (MO)

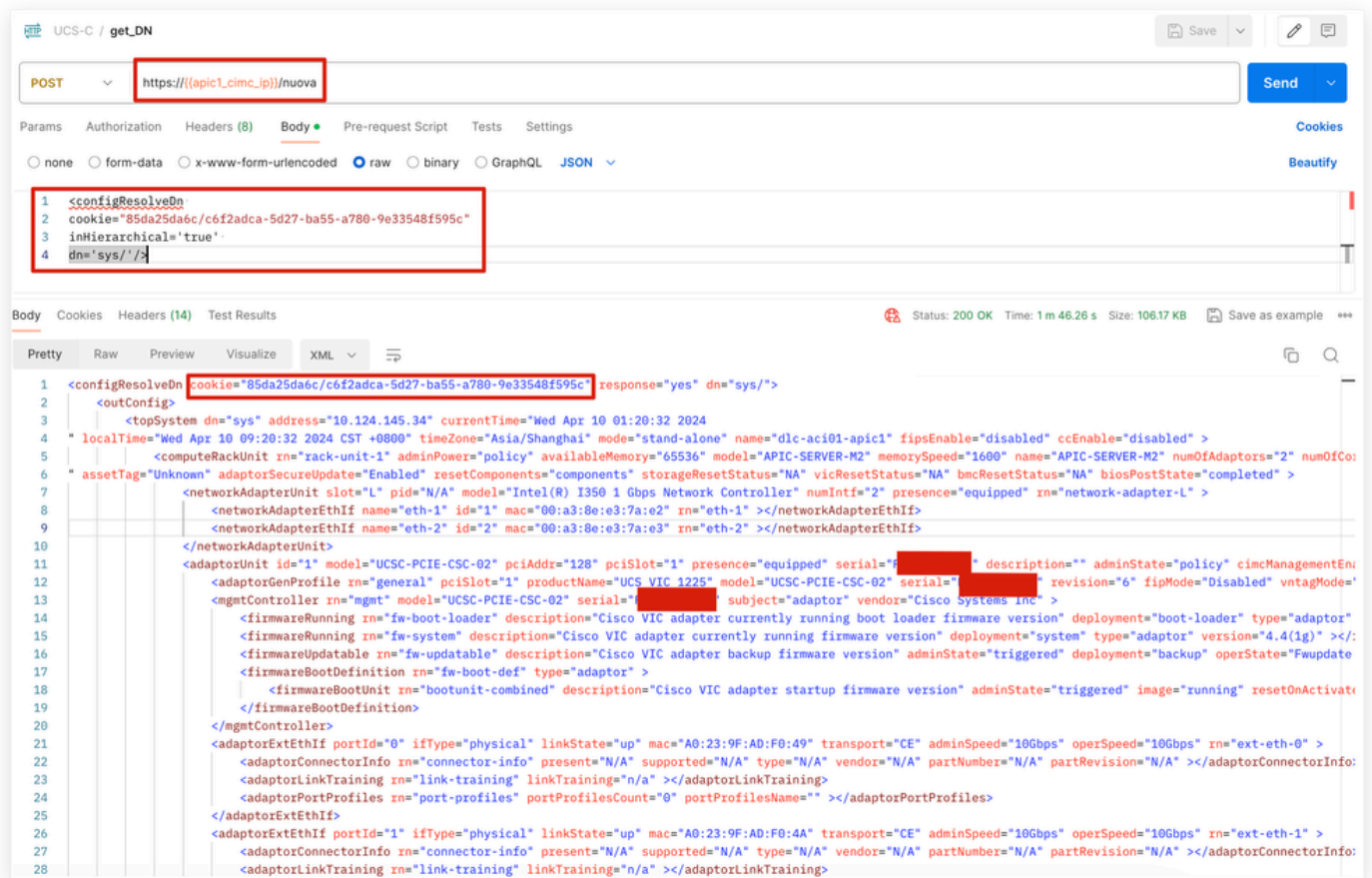
Al utilizar `configResolveDn` con `inHierarchical="true"` y `dn="sys/"`, recupera toda la información de objetos administrados (MO) de CIMC.

`configResolveDn`: El método `configResolveDn` recupera un único objeto administrado para un DN especificado.

`inHierarchical=true`: Cuando se establece en `true`, devuelve toda la información del nodo secundario. Esta combinación permite obtener información MO de todos los nodos y subnodos de CIMC.

`dn="sys/"`: Esta es la raíz superior del MIT.

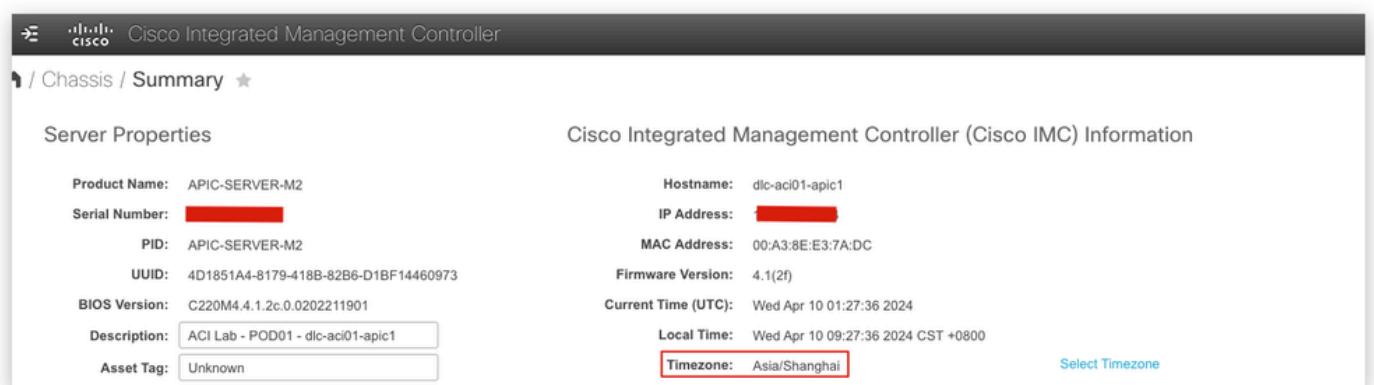
Respuesta de API:



Copie la respuesta de ejecución de Postman en un editor de texto como Notepad, PyCharm o Visual Studio Code para realizar búsquedas posteriores de la clase y el DN basados en MO.

Ejemplo 1: Consulte la clase y el DN de la zona horaria

La zona horaria configurada en la actual GUI de CIMC es "Asia/Shanghai".



Busque 'Asia/Shanghai' en los resultados devueltos por Postman en el paso 1b. La zona horaria es "Asia/Shanghai", la clase es "topSystem", y el DN es "sys".

```
<#root>
```

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

```
<topSystem
```

```
dn="sys"
```

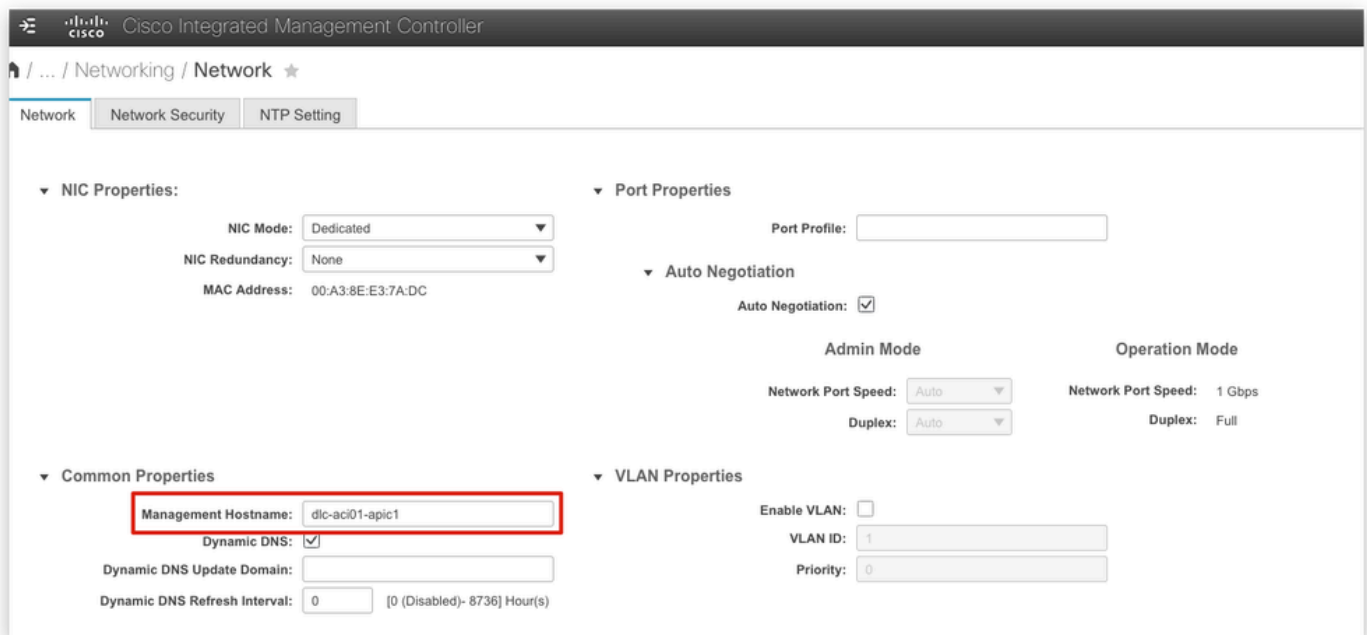
```
address="a.b.c.d" currentTime="Wed Apr 10 01:05:12 2024"  
" localTime="Wed Apr 10 09:05:12 2024 CST +0800"
```

```
timeZone="Asia/Shanghai"
```

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

Ejemplo 2: Consulte la clase y el DN del nombre de host

El nombre de host configurado en la GUI de CIMC actual es "dlc-aci01-apic1".



Busque "dlc-aci01-apic1" en los resultados devueltos por Postman. El nombre de host es "dlc-aci01-apic1", la clase es "mgmtIf" y la ejecución es "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" >
```

Luego, consulte CIMC visore desde https://CIMC_IP/visore.html, el nombre de host "dlc-aci01-apic1" corresponde al DN= `sys/rack-unit-1/mgmt/if-1`.

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. Administración de CIMC a través de API REST

- En el paso 1, ya ha identificado la clase y el nombre distinguido (DN) correspondientes al objeto administrado (MO).
- Puede utilizar el módulo Ansible `community.general.imc_rest` para administrar CIMC mediante API. La información detallada se refiere a: [imc_rest module Documentación](#)

Utilice el método API `configResolveClass` para recuperar información

`configResolveClass`: El método recupera un objeto administrado en una clase determinada. Si `inHierarchical=true`, los resultados contienen elementos secundarios. Tomando la versión del firmware de consulta como ejemplo, utilice el método API `configResolveClass` y especifique el `classID` del MO.

Salida de contenido de script Ansible:

```
<#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'  
  
>
```

Utilice el método API `configConfMo` para modificar la configuración

Para modificar la configuración de MO mediante la API CIMC, utilice el método `configConfMo`. Este método está diseñado para configurar o modificar los ajustes de un MO específico. Al invocar `configConfMo`, es importante proporcionar la clase exacta o la información DN del MO que desea modificar.

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

Salida de contenido de script Ansible:

<#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. Ejemplo de flujo de trabajo de configuración de automatización CIMC

Cisco APIC es un software de controlador de Cisco ACI instalado en la serie UCS C220. El flujo de trabajo ilustra el proceso automatizado de recreación de imágenes del software 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

Ejemplo de módulo Ansible:



Nota: El ejemplo solo incluye información de contenido, el módulo de análisis completo se refiere al módulo `community.general.imc_rest` del sitio web oficial de 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
```

Información Relacionada

[Servidores de montaje en bastidor Cisco UCS Guía del programador de la API XML de Cisco IMC](#)

[community.general.imc_rest module - Administrar el hardware de Cisco IMC a través de su API REST](#)

[Referencia del modelo de información de UCS Manager](#)

Acerca de esta traducción

Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).