

Configurar o CIMC através do módulo API Ansible Rest

Contents

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Visão geral da API do CIMC](#)

[Configurações](#)

[1. Localize a classe ou o DN do MO \(Objeto Gerenciado\) do CIMC](#)

[1-A. Use a API para fazer login no CIMC e recuperar as informações do cookie](#)

[1-B. Usando o método de consulta API configResolveDn para recuperar todas as informações do Objeto Gerenciado \(MO\)](#)

[Exemplo 1: Consulte a classe e o DN do fuso horário](#)

[Exemplo 2: Consultar a classe e o DN do nome do host](#)

[2. Gerenciamento do CIMC via API REST](#)

[Use o método de API configResolveClass para recuperar Informações](#)

[Use o método de API configConfMo para modificar a configuração](#)

[3. Exemplo de fluxo de trabalho de configuração de automação do CIMC](#)

[Informações Relacionadas](#)

Introdução

Este documento descreve como configurar o Cisco Integrated Management Controller (CIMC) através do módulo API REST Ansible.

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- CIMC UCS
- API
- Ansible

Componentes Utilizados

- UCS C220-M4, versão 4.1(2f)
- Um cliente que executa o postman e o ansible versão 2.14.5

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

Visão geral da API do CIMC

Todos os componentes físicos e lógicos que compõem o Cisco UCS são representados em um modelo de informações de gerenciamento hierárquico (MIM), também conhecido como MIT. Cada nó na árvore representa um objeto gerenciado (MO) ou grupo de objetos que contém seu estado administrativo e seu estado operacional.

A estrutura hierárquica começa na parte superior (sys) e contém nós pai e filho. Cada nó nessa árvore é um objeto gerenciado e cada objeto no Cisco UCS tem um nome distinto (DN) exclusivo que descreve o objeto e seu lugar na árvore. Objetos gerenciados são abstrações dos recursos do Cisco UCS, como CPUs, DIMMs, placas adaptadoras, ventoinhas e unidades de fonte de alimentação.

Ilustração da estrutura do CIMC MIM:

```
|-----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)
```

Nomeação de objetos:

- DN: O nome distinto permite que você identifique sem ambiguidade um objeto de destino.
- RN: O nome relativo identifica um objeto dentro do contexto de seu objeto pai.

Por exemplo, este nome distinto:

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

É composto por 4 nomes relativos:

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

API usada neste artigo:

- Autenticação: aaaLogin. Método inicial de login. Use o método aaaLogin para obter um cookie válido.
- Consulta: configResolveDn. Recupera objetos por DN.
- Configuração: configConfMo. O método configConfMo é usado para configurar uma ou mais propriedades em um Objeto Gerenciado (MO). O MO a ser configurado é identificado exclusivamente por um DN (Distinguished Name - Nome Distinto).



Note:

Muitos métodos de consulta incluem um argumento inHierarchical que aceita um valor booleano (true/yes ou false/no). Quando definido como verdadeiro, este argumento faz com que o método retorne todos os objetos filho dentro da hierarquia.

Configurações

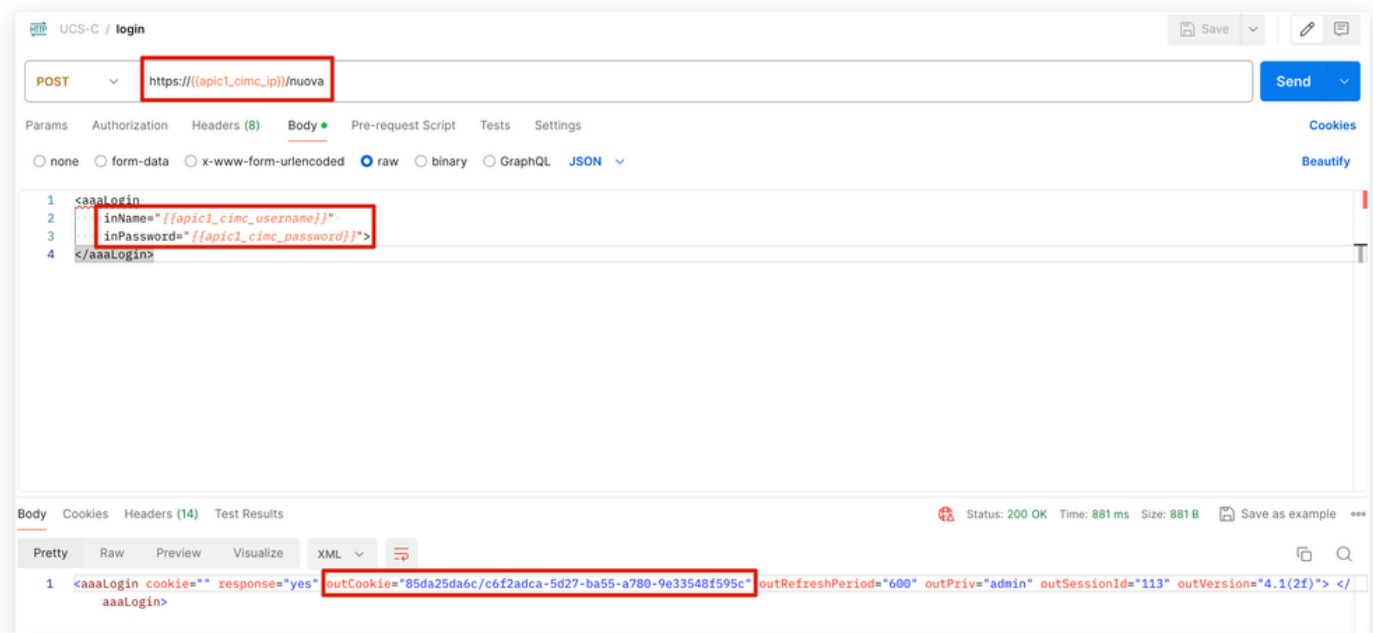
1. Localize a classe ou o DN do Objeto Gerenciado (MO) do CIMC

Para automatizar a configuração do CIMC por meio de sua API, é essencial determinar a classe específica ou as informações de nome distinto (DN) associadas aos objetos gerenciados (MOs) que você deseja configurar.

1-A. Use a API para fazer login no CIMC e recuperar as informações do cookie

Envie uma solicitação POST para https://{{apic_cimc_ip}}/nuova e especifique o método aaaLogin. Insira um nome de usuário e uma senha.

Copie o cookie da resposta da API.



Alternativamente, você pode utilizar a curva para buscar as informações do cookie.

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

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

1-B. Usando o método de consulta API configResolveDn para recuperar todas as informações do Objeto Gerenciado (MO)

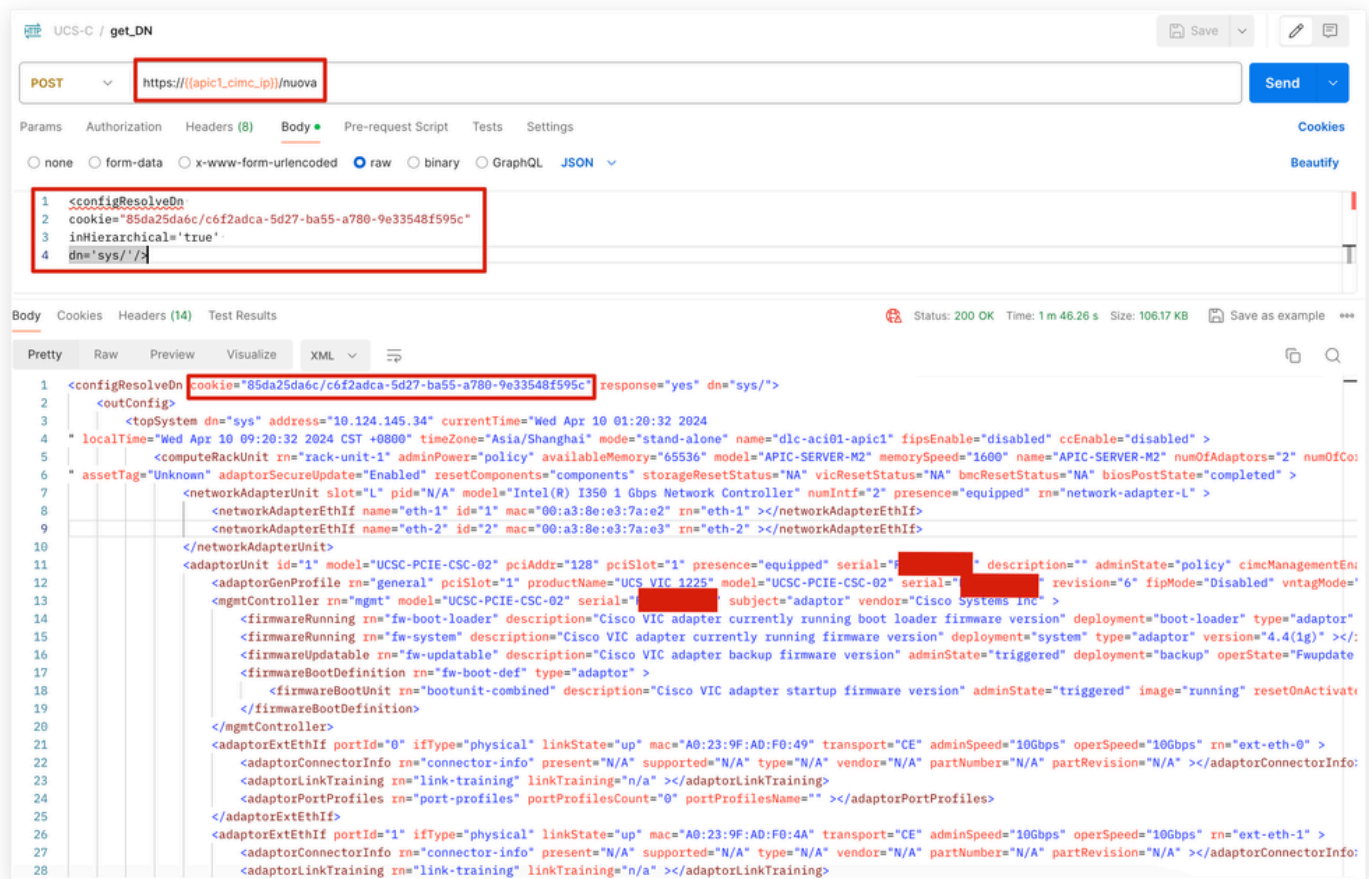
Ao usar configResolveDn com inHierarchical="true" e dn="sys/", ele recupera todas as informações do Objeto Gerenciado (MO) do CIMC.

configResolveDn: O método configResolveDn recupera um único objeto gerenciado para um DN especificado.

inHierarchical=true: Quando definido como verdadeiro, retorna todas as informações do nó filho. Essa combinação permite buscar todas as informações de MO de nós e subnós do CIMC.

dn="sys/": Essa é a raiz superior do MIT.

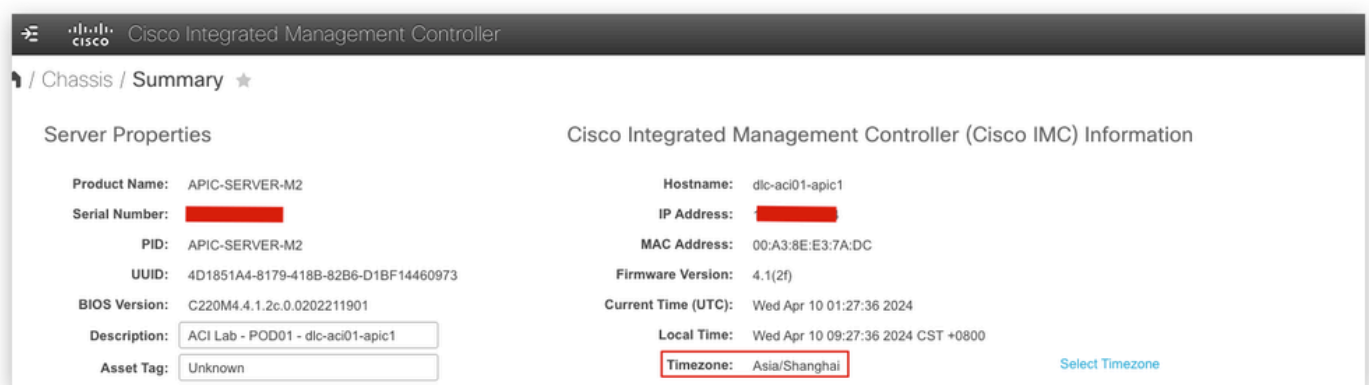
Resposta de API:



Copie a resposta de execução do Postman para um editor de texto como Notepad, PyCharm ou Visual Studio Code para pesquisa da classe e DN com base em MO.

Exemplo 1: Consulte a classe e o DN do fuso horário

O fuso horário configurado na GUI atual do CIMC é "Ásia/Xangai".



Procure 'Ásia/Xangai' a partir dos resultados retornados por Postman na etapa 1b. O fuso horário é "Ásia/Xangai", a classe é "topSystem", e o DN é "sys/".

<#root>

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

```
<topSystem
```

```
dn="sys"
```

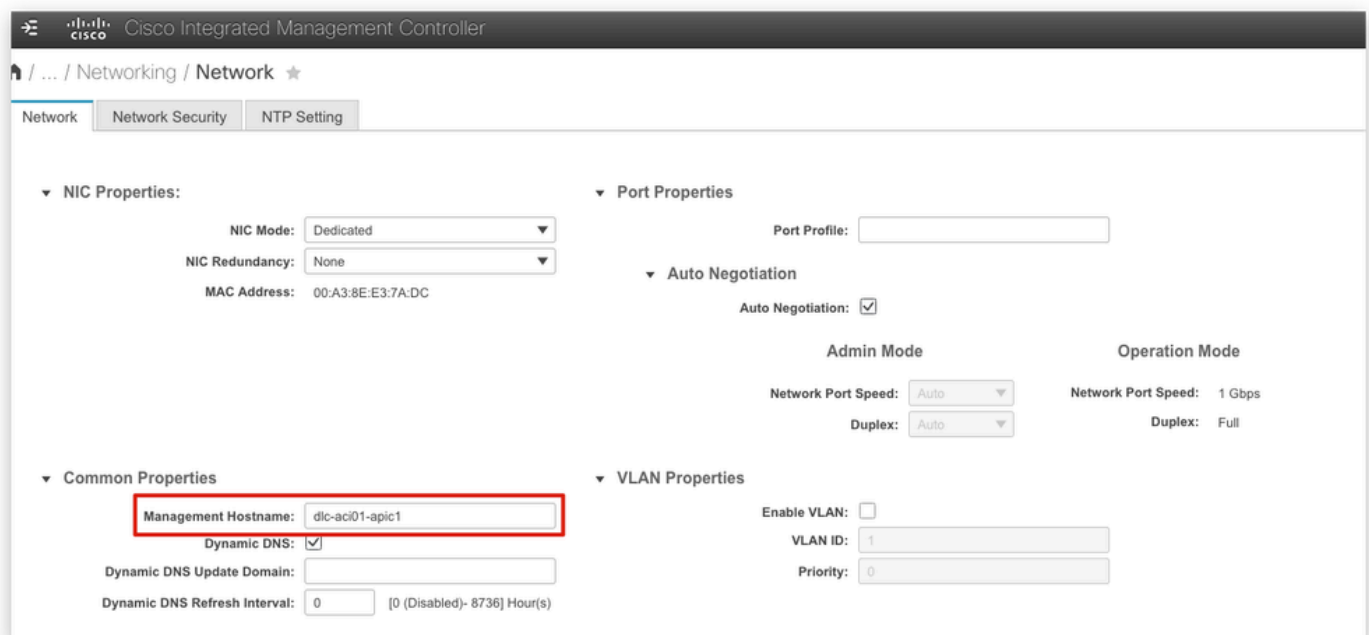
```
address="a.b.c.d" currentTime="Wed Apr 10 01:05:12 2024"  
" localTime="Wed Apr 10 09:05:12 2024 CST +0800"
```

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

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

Exemplo 2: Consultar a classe e o DN do nome do host

O nome de host configurado na GUI do CIMC atual é "dlc-aci01-apic1".



Procure "dlc-aci01-apic1" nos resultados retornados por Postman. O nome do host é "dlc-aci01-apic1", a classe é "mgmtIf" e o rn é "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" >
```

Em seguida, consulte o CIMC visore em https://CIMC_IP/visore.html, o nome de host "dlc-aci01-apic1" corresponde ao 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. Gerenciamento do CIMC via API REST

- Na etapa 1, você já identificou a classe e o nome distinto (DN) correspondentes ao objeto gerenciado (MO).
- Você pode usar o módulo Ansible `community.general.imc_rest` para gerenciar o CIMC via API. Informações detalhadas referem-se a: [imc_rest module Documentação](#)

Use o método de API `configResolveClass` para recuperar informações

`configResolveClass` O método recupera o objeto gerenciado em uma determinada classe. Se `inHierarchical=true`, os resultados conterão filhos. Tomando como exemplo a versão de firmware de consulta, use o método de API `configResolveClass` e especifique o `classID` do MO.

Saída de conteúdo 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'  
>
```

Use o método de API `configConfMo` para modificar a configuração

Para modificar a configuração do MO usando a API do CIMC, use o método `configConfMo`. Este método foi projetado para definir ou modificar as configurações de um MO específico. Ao chamar `configConfMo`, é importante fornecer a classe exata ou informações DN do MO que você deseja 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

Saída de conteúdo 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. Exemplo de fluxo de trabalho de configuração de automação do CIMC

O Cisco APIC é um software controlador da Cisco ACI instalado no UCS C220 Series. O fluxo de trabalho ilustra o processo automatizado de recriação de imagens do 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

Exemplo de módulo Ansible:



Note: O exemplo inclui apenas informações de conteúdo, módulo ansible completo refere-se a community.general.imc_rest módulo do site oficial 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
```

Informações Relacionadas

[Servidores em rack Cisco UCS Guia do programador de API XML IMC da Cisco](#)

[community.general.imc_rest module - Gerencie o hardware Cisco IMC por meio de sua API REST](#)

[Referência do modelo de informações do UCS Manager](#)

Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.