通过Ansible Rest API模块配置CIMC

目录

简介

本文档介绍如何通过Ansible REST API模块配置思科集成管理控制器(CIMC)。

先决条件

要求

Cisco 建议您了解以下主题：

- UCS CIMC
- API
- Ansible

使用的组件

- UCS C220-M4，版本4.1(2f)
- 运行postman和ansible版本2.14.5的客户端

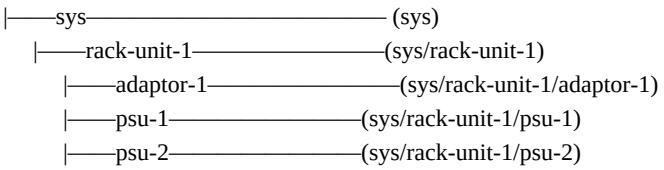
本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

CIMC API概述

Cisco UCS的所有物理和逻辑组件都以分层管理信息模型(MIM)（也称为MIT）表示。树中的每个节点代表一个托管对象(MO)或一组对象，其中包含其管理状态和运行状态。

分层结构从顶部(sys)开始，包含父节点和子节点。此树中的每个节点都是一个托管对象，Cisco UCS中的每个对象都有一个唯一的可分辨名称(DN)，用于描述对象及其在树中的位置。托管对象是Cisco UCS资源的抽象，例如CPU、DIMM、适配器卡、风扇和电源设备。

CIMC MIM结构图示：



对象命名：

- DN:可分辨名称使您能够明确标识目标对象。
- RN:相对名称标识父对象上下文中的对象。

例如，此可分辨名称：

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

由4个相对名称组成：

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

本文中使用的API:

- 身份验证:aaaLogin。初始登录方法。使用aaaLogin方法获取有效的Cookie。
- 查询： configResolveDn。按DN检索对象。
- 配置:configConfMo。configConfMo方法用于在受管对象(MO)中配置一个或多个属性。 要配置的MO由可分辨名称(DN)唯一标识。

注意：

许多查询方法都包含inHierarchical参数，该参数接受布尔值（true/yes或false/no）。 设置为true时，此参数导致方法返回层次结构中的所有子对象。

配置

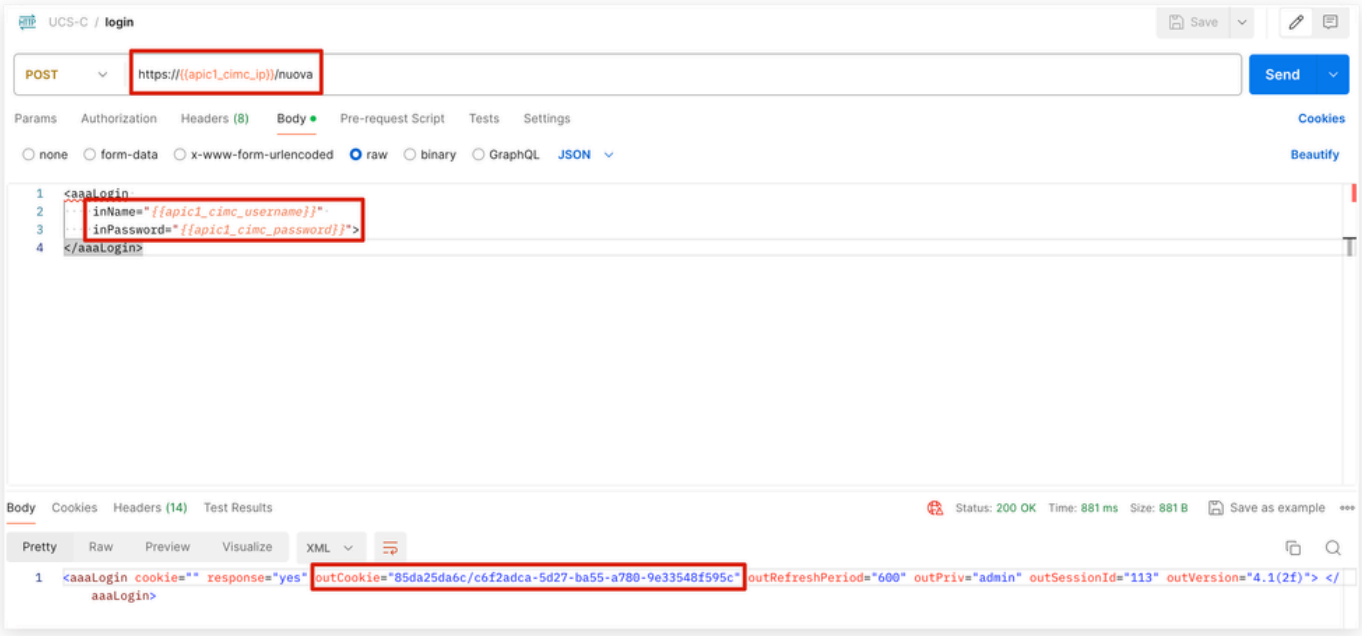
1.查找CIMC托管对象(MO)的类或DN

要通过其API自动配置CIMC，必须确定与您想要配置的托管对象(MO)关联的特定类或可分辨名称(DN)信息。

1a。使用API登录到CIMC并检索cookie信息

将POST请求发送到https://{{apic_cimc_ip}}/nuova 并指定aaaLogin方法。输入用户名和密码。

从API响应复制cookie。



或者，您可以使用curl获取cookie信息。

curl -k -d "

" https://apic_cimc_ip/nuova

1b.使用API查询方法configResolveDn检索所有托管对象(MO)信息

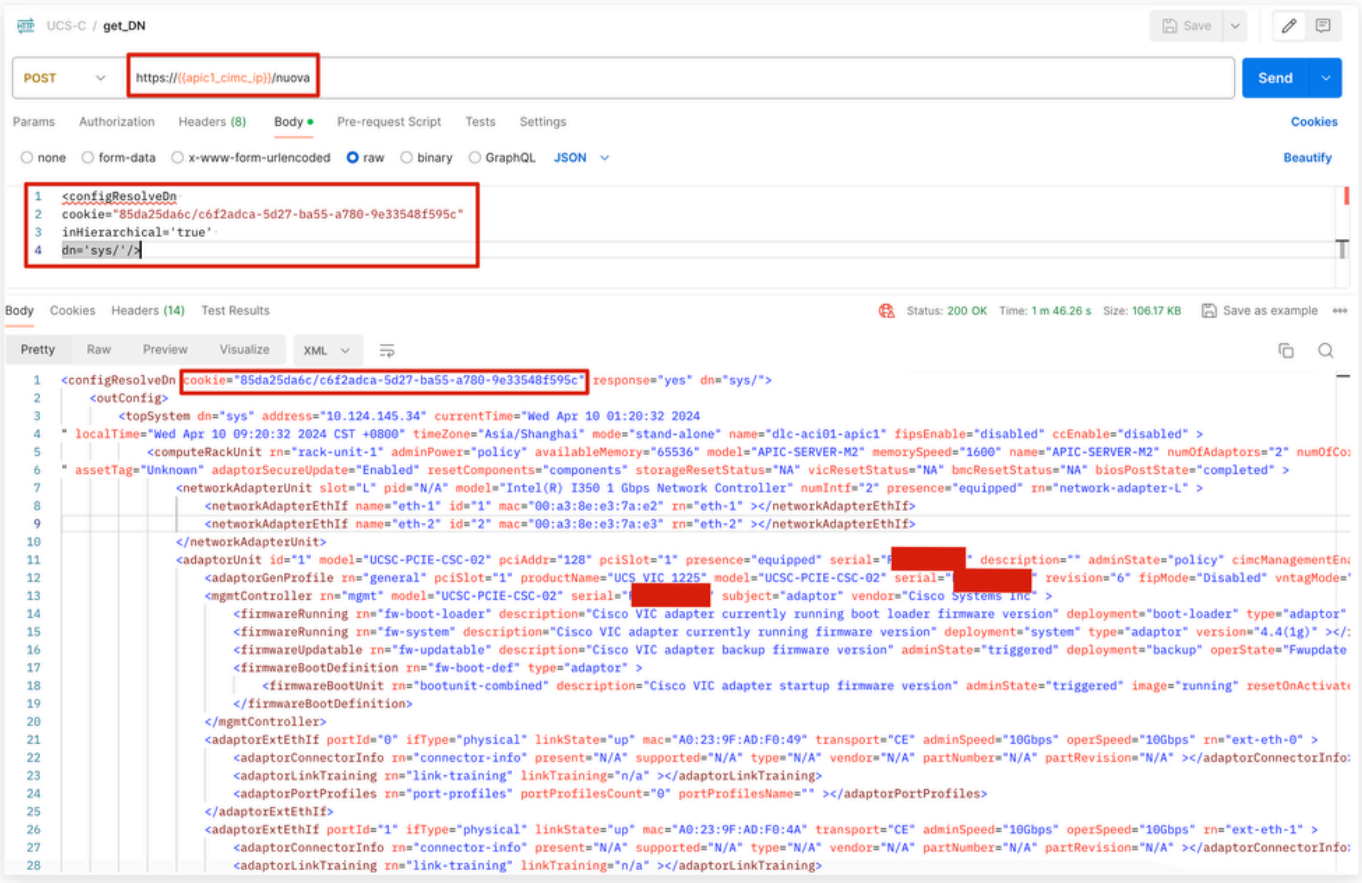
将configResolveDn与inHierarchical="true"和dn="sys/"配合使用时，它会从CIMC检索所有托管对象(MO)信息。

configResolveDn:configResolveDn方法为指定DN检索单个托管对象。

inHierarchical=true:设置为true时，返回所有子节点信息。此组合允许从CIMC获取所有节点和子节点的MO信息。

dn="sys/":这是MIT的根目录。

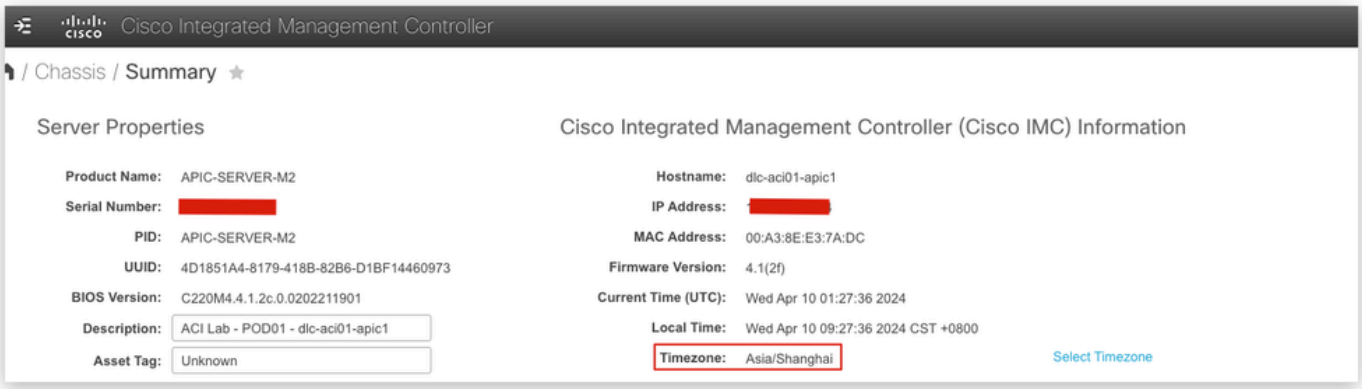
API响应：



将Postman执行响应复制到文本编辑器（如Notepad、PyCharm或Visual Studio Code），以便以后搜索基于MO的类和DN。

示例 1：查询时区的类和DN

当前CIMC GUI中配置的时区为“Asia/Shanghai”。



从Postman在步骤1b中返回的结果中搜索“Asia/Shanghai”。时区为“Asia/Shanghai”，类别为“topSystem”,DN为“sys/”。

<#root>

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

dn="sys"

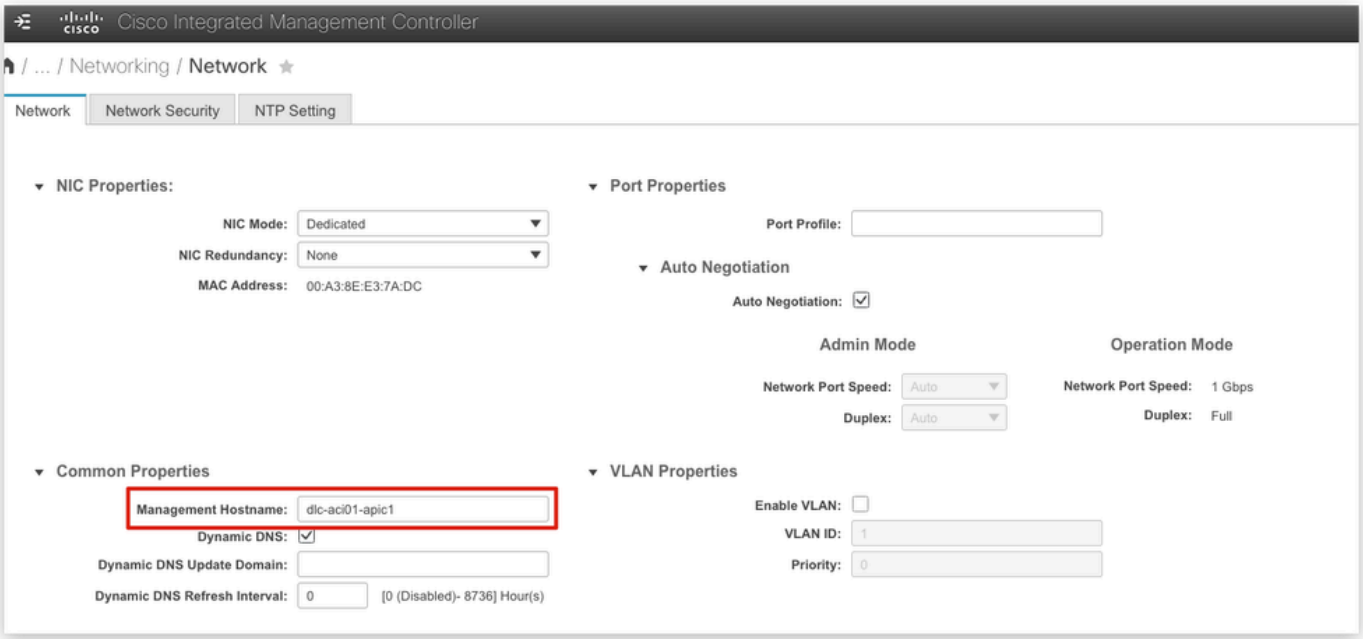
```
 address="a.b.c.d" currentTime="Wed Apr 10 01:05:12 2024
" localTime="Wed Apr 10 09:05:12 2024 CST +0800"
```

timeZone="Asia/Shanghai"

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

示例 2：查询主机名的类和DN

当前CIMC GUI中配置的主机名为“dlc-aci01-apic1”。



在Postman返回的结果中搜索“dlc-aci01-apic1”。主机名为“dlc-aci01-apic1”，类为“mgmtIf”，rn为“if-1”。

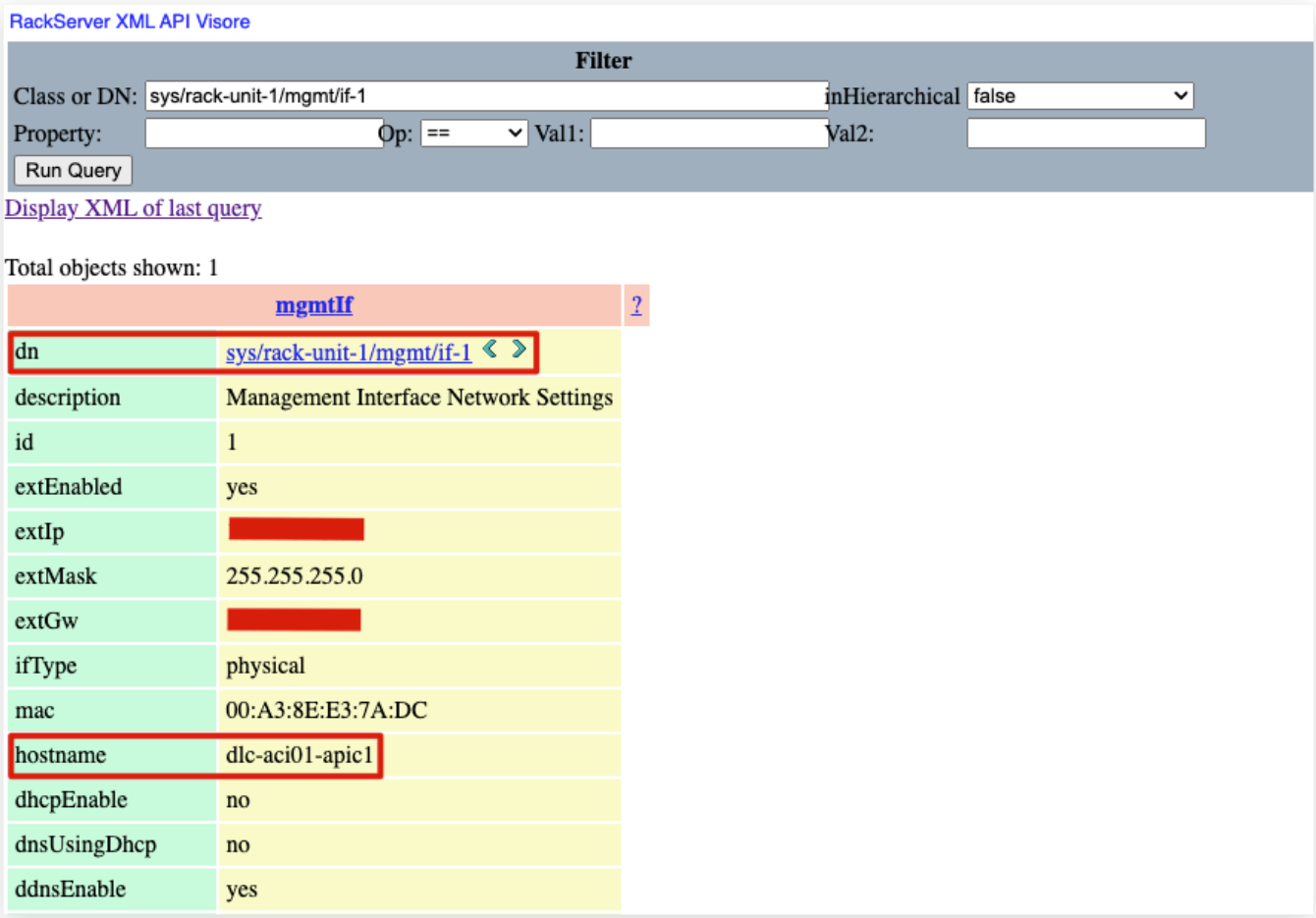
<#root>

<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" >

然后，从https://CIMC_IP/visore.html查询CIMC visore，主机名“dlc-aci01-apic1”对应于DN=“sys/rack-unit-1/mgmt/if-1”。



2.通过REST API管理CIMC

- 在第1步中，您已标识与托管对象(MO)对应的类和可分辨名称(DN)。
- 您可以使用Ansible community.general.imc_rest模块通过API管理CIMC。有关详细信息，请参阅imc_rest_module文档

使用API方法configResolveClass检索信息

configResolveClass:该方法检索给定类中的托管对象。如果inHierarchical=true，则结果包含子级。以查询固件版本为例，使用API方法configResolveClass并指定MO的classID。

可能的脚本内容输出：

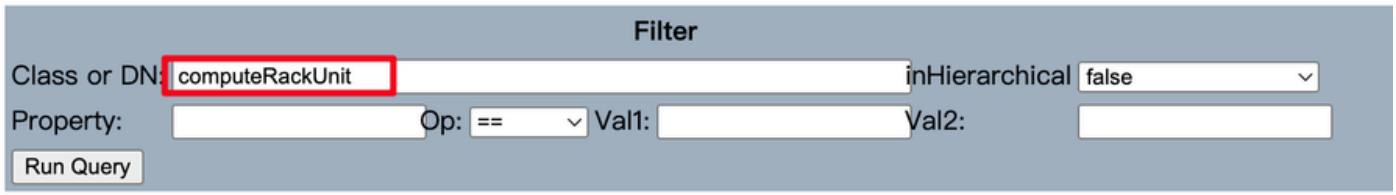
<#root>

```
- 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'
/>
```

使用API方法configConfMo修改配置

要使用CIMC API修改MO的配置，请使用configConfMo方法。此方法用于配置或修改特定MO的设置。调用configConfMo时，提供要修改的MO的确切类或DN信息非常重要。



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

可能的脚本内容输出：

<#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自动化配置工作流程示例

思科APIC是安装在UCS C220系列上的思科ACI控制器软件。该工作流程说明了重新映像APIC软件的自动化流程。

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模块示例：

注意：示例仅包含内容信息，full ansible module是指Ansible官方网站中的community.general.imc_rest module

---

<#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
```

相关信息

[Cisco UCS机架式服务器Cisco IMC XML API程序员指南]

[community.general.imc_rest模块 — 通过其REST API管理Cisco IMC硬件]

[UCS Manager信息模型参考]

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。