

Cisco APIC-EM Network Python Sandbox v1.1

最后更新日期：2015 年 12 月 11 日

关于本解决方案

思科应用策略基础设施控制器企业模块 (APIC-EM) 是一款软件定义网络 (SDN) 控制器，专用于协调和管理组成思科基础设施的局域网 (LAN) 和广域网 (WAN)。APIC-EM 解决方案由以下四大应用组成：网络 PnP、IWAN、路径跟踪和网络资产。可以从 GUI 或 RESTful API 查看所有这些应用。

思科应用策略基础设施控制器企业模块 (Cisco APIC-EM) 可以统一企业网络自动化和管理。自动化 APIC-EM 控制器的首选方法是通过编程语言 *python*。

本演示是创建和部署自动化 APIC-EM 控制器功能的 python 脚本的沙盒环境。

关于本演示

本演示包括显示如何使用思科 APIC-EM 网络 PnP 的场景，包括：

- [场景 1：使用 PyCharm 和运行 Python 脚本](#)
- [场景 2：通过 Swagger 探索 API](#)

要求

本演示无需终端路由器。

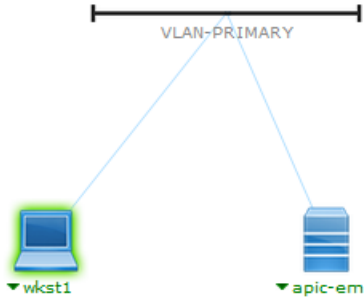
表1. 要求

必备	可选
<ul style="list-style-type: none">• 笔记本电脑	<ul style="list-style-type: none">• Cisco AnyConnect

拓扑

此内容包含预配置的用户和组件，旨在说明解决方案脚本场景和功能。大多数组件都能通过预定义管理用户帐户进行完整的配置。要查看用于访问组件的 IP 地址和用户帐户凭证，可点击活动会话的 **Topology（拓扑）** 菜单中的组件图标，也可点击需要使用这些凭证的场景步骤中的组件图标。

图 1. 拓扑



开始演示

演示前的准备

在向现场观众进行演示之前，我们强烈建议您浏览本文档的内容，并使用活动会话进行实操。这有助于您熟悉文档和内容结构。

准备是成功演示的关键。

执行以下步骤安排内容会话并配置演示环境。

1. 浏览至 dcloud.cisco.com，选择最接近您的位置，然后利用 Cisco.com 凭证登录。
2. 如果这是您首次将路由器与 dCloud 结合使用，则注册并配置路由器。[[显示操作方法](#)]
3. 安排会话。[[显示操作方法](#)]
4. 测试连接。[[显示操作方法](#)]
5. 在 **My Dashboard（我的控制面板）** > **My Sessions（我的会话）** 中检查会话状态是否为 **Active（活动）**。

注意：会话可能最多需要 30 分钟才能变为活动会话。

6. 点击 **View（查看）** 打开活动会话。
7. 为获取最佳性能，利用 **Cisco AnyConnect VPN** [[显示操作方法](#)] 和笔记本电脑上的本地 **RDP 客户端** [[显示操作方法](#)] 连接至工作站。
 - 工作站 1: **198.18.133.36**，用户名: **administrator**，密码: **C1sco12345**

注意：您也可以使用 Cisco dCloud 远程桌面客户端 [[显示操作方法](#)] 连接到工作站。dCloud 远程桌面客户端最适合用于访问交互最少的活动会话。然而，许多用户在使用该方法时都遇到连接问题和性能问题。

场景1. 使用 PyCharm 和运行 Python 脚本

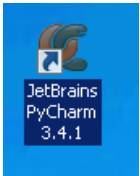
注意：本演示提供 APIC-EM GUI 访问权限，其中，预填写的设备数据库可用。**无连接至本演示的实际设备。**要求发现新设备的任何 API 功能都将无效。

或者，可以连接至 APIC-EM 和通过公共 IP 地址写入自定义脚本。有关该操作步骤的详细信息，请查看[附录 A: 通过公共 IP 连接](#)。

步骤

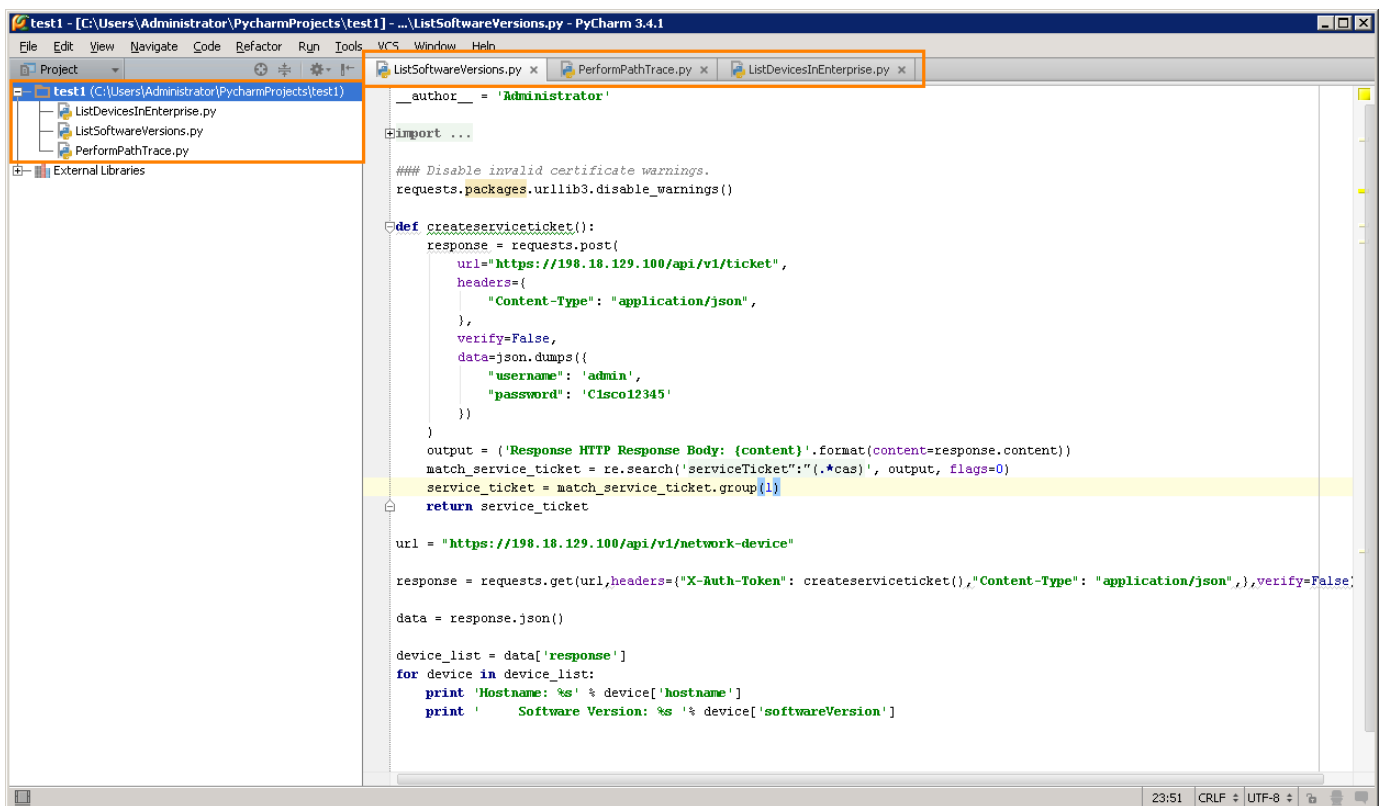
1. 从桌面上点击 **JetBrains PyCharm 3.4.1** 图标。

图 2. JetBrains PyCharm 3.4.1



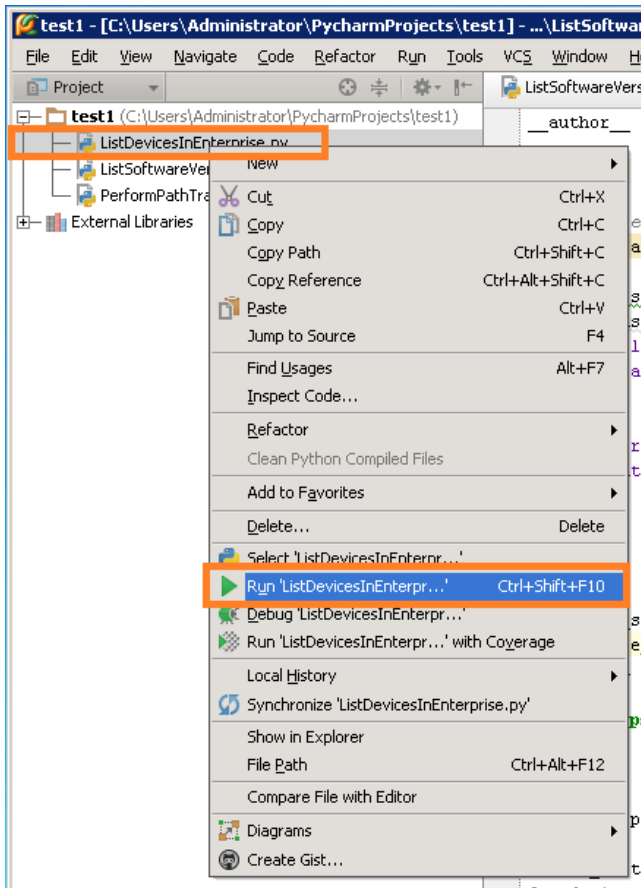
2. 这会加载 **PyCharm project listing (PyCharm 项目列表)**，其中，预建脚本已列出。

图 3. PyCharm 项目列表



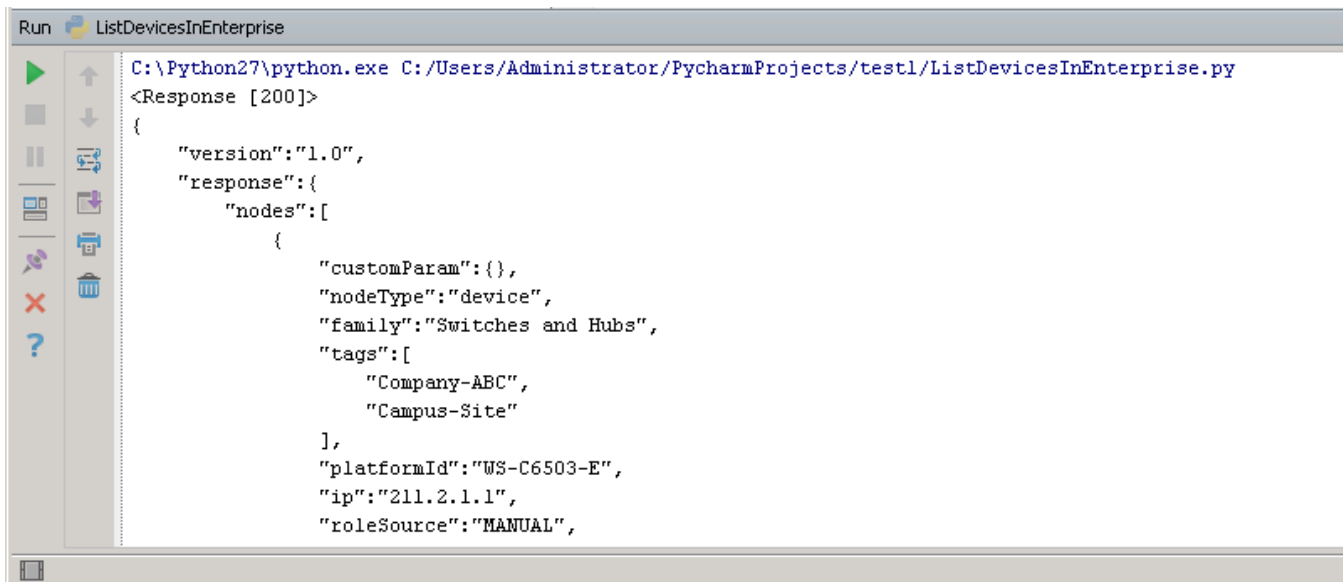
3. 右键点击 **ListDevicesInEnterprise.py** 脚本。右键点击并选择 **Run（运行）** 命令。

图 4. ListDevicesInEnterprise.py 运行命令



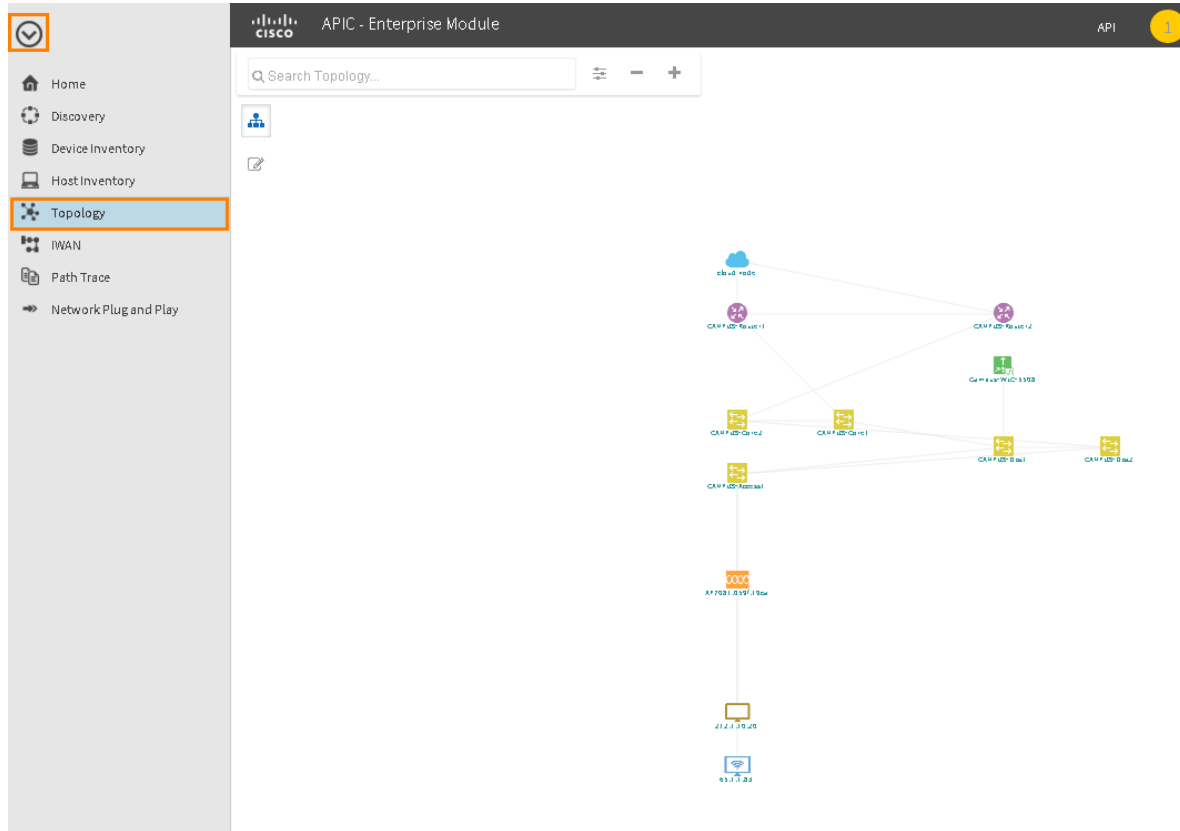
4. 脚本在“Run”（运行）窗口中执行，并在 APIC-EM 中列出设备。

图 5. 执行脚本



5. 打开浏览器并启动 **APIC-EM**。接受任何警告并使用 **userid admin**（用户名 **admin**）和 **password C1sco12345**（密码 **C1sco12345**）登录。
6. 如果左侧菜单未展开，则**点击箭头**，然后选择 **Topology**（拓扑）。此时会显示“APIC-EM Topology”（APIC-EM 拓扑）页面。

图 6. APIC-EM 拓扑



7. 选择 **Device Inventory**（设备资产）。此时会显示所有调出设备列表。这些设备与之前“Run”（运行）窗口中的设备相匹配。

图 7. 设备列表

Device Name	IP Address	Device Status	Up Time	Last Updated Time	Last Inventory Collection Status
AP7081-059f19ca	55.1.1.3	Reachable	NA	a few seconds ago	Managed
CAMPUS-Access1	212.1.10.1	Reachable	12 days, 0:18:10.78	14 minutes ago	Managed
CAMPUS-Core1	211.3.1.1	Reachable	329 days, 21:40:06.38	14 minutes ago	Managed
CAMPUS-Core2	211.2.1.1	Reachable	329 days, 20:41:35.46	14 minutes ago	Managed
CAMPUS-Dist1	55.1.1.100	Reachable	12 days, 0:33:05.06	14 minutes ago	Managed
CAMPUS-Dist2	212.3.1.2	Reachable	2 days, 22:50:49.57	14 minutes ago	Managed
CAMPUS-Router1	210.1.1.1	Reachable	329 days, 22:20:15.59	14 minutes ago	Managed
CAMPUS-Router2	210.2.1.1	Reachable	58 days, 17:20:48.47	14 minutes ago	Managed
Campus-WLC-5508	172.28.97.253	Reachable	209 days, 3:06:04.37	14 minutes ago	Managed

8. 点击 **CAMPUS-Core2** 设备名称显示详细设备信息。该信息与显示于之前“Run”（运行）窗口中的信息相匹配。

图 8. CAMPUS-Core2

Interface Name	MAC Address	Status
FortyGigabitEthernet3/1	70:ca:9b:8f:6c:6c	Down
FortyGigabitEthernet3/2	70:ca:9b:8f:6c:6d	Down
GigabitEthernet1/1	24:e9:b3:3f:b1:c0	Up
Vlan1	24:e9:b3:3f:b1:c0	Down
GigabitEthernet1/2	24:e9:b3:3f:b1:c0	Down
GigabitEthernet1/3	24:e9:b3:3f:b1:c0	Up
TenGigabitEthernet1/4	24:e9:b3:3f:b1:c0	Up
FortyGigabitEthernet3/3	70:ca:9b:8f:6c:6e	Down
FortyGigabitEthernet3/4	70:ca:9b:8f:6c:6f	Down
TenGigabitEthernet1/5	24:e9:b3:3f:b1:c0	Down

9. 点击 **X** 关闭窗口。

10. 返回 PyCharm 并运行 **ListSoftwareVersions.py** 脚本。

图 9. ListSoftwareVersions.py 脚本

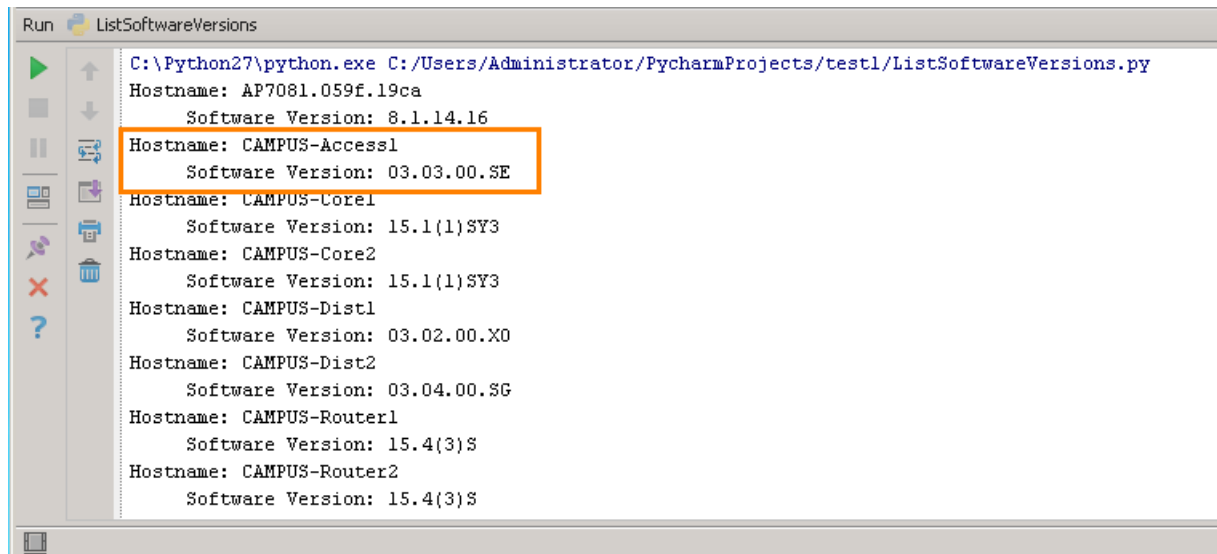
```

{
  "version": "1.0",
  "response": {
    "nodes": [
      {
        "name": "CAMPUS-Core2",
        "family": "Switches and Hubs",
        "author": "Adminis
    ]
  }
}

```

11. 脚本在“Run”（运行）窗口中执行，并会列出各设备的主机名和软件版本。

图 10. 主机名和软件版本



```

Run ListSoftwareVersions
C:\Python27\python.exe C:/Users/Administrator/PycharmProjects/test1/ListSoftwareVersions.py
Hostname: AP7081.059f.19ca
  Software Version: 8.1.14.16
Hostname: CAMPUS-Access1
  Software Version: 03.03.00.SE
Hostname: CAMPUS-Core1
  Software Version: 15.1(1)SY3
Hostname: CAMPUS-Core2
  Software Version: 15.1(1)SY3
Hostname: CAMPUS-Dist1
  Software Version: 03.02.00.X0
Hostname: CAMPUS-Dist2
  Software Version: 03.04.00.SG
Hostname: CAMPUS-Router1
  Software Version: 15.4(3)S
Hostname: CAMPUS-Router2
  Software Version: 15.4(3)S
  
```

12. 指向 `ListSoftwareVersions.py` 选项卡显示脚本详细信息，包括如何格式化输出的相关信息。

图 11. `ListSoftwareVersions.py` 选项卡



```

)
output = ('Response HTTP Response Body: {content}'.format(content=response.content))
match_service_ticket = re.search('serviceTicket": "(.*cas)', output, flags=0)
service_ticket = match_service_ticket.group(1)
return service_ticket

url = "https://198.18.129.100/api/v1/network-device"

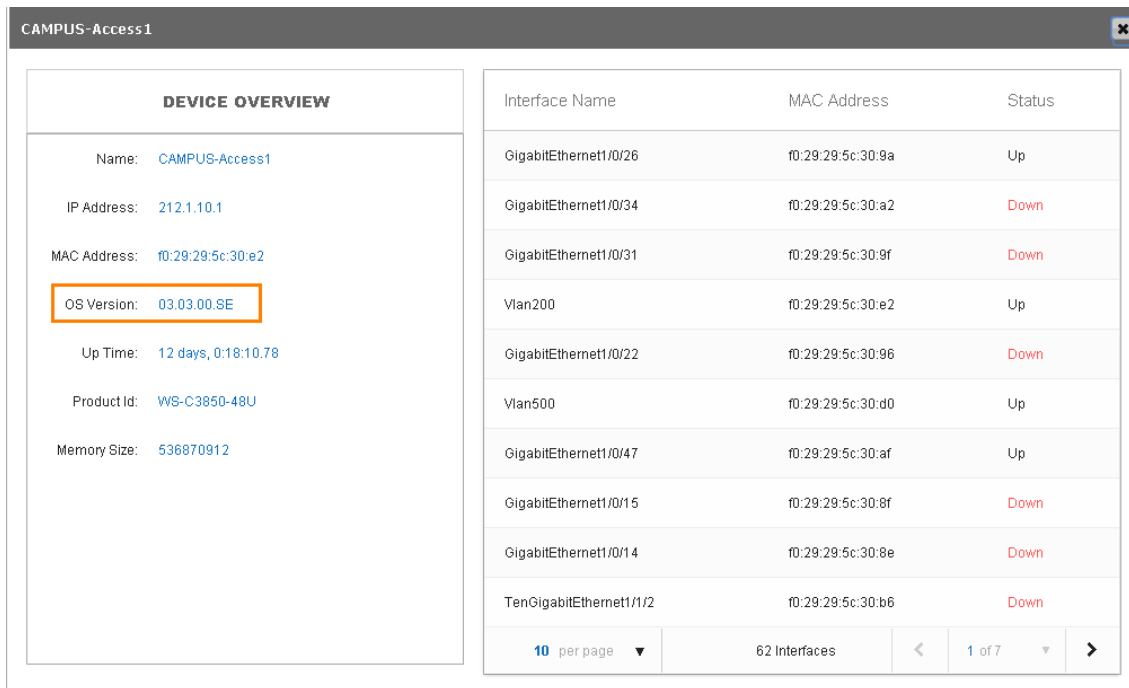
response = requests.get(url, headers={"X-Auth-Token": createserviceticket(), "Content-Type": "application/json"}, verify=False)

data = response.json()

device_list = data['response']
for device in device_list:
    print 'Hostname: %s' % device['hostname']
    print '    Software Version: %s' % device['softwareVersion']
  
```


13. 返回 APIC-EM 窗口，然后单击列表中的 **CAMPUS-Access1**。指出 OS 版本与“Run”（运行）窗口中所列的软件版本相同。

图 12. CAMPUS-Access1 OS 版本



The screenshot shows the APIC-EM interface for the device 'CAMPUS-Access1'. On the left, the 'DEVICE OVERVIEW' section displays the following details:

- Name: CAMPUS-Access1
- IP Address: 212.1.10.1
- MAC Address: f0:29:29:5c:30:e2
- OS Version: **03.03.00.SE** (highlighted with a red box)
- Up Time: 12 days, 0:18:10.78
- Product Id: WS-C3850-48U
- Memory Size: 536870912

On the right, a table lists the interfaces and their status:

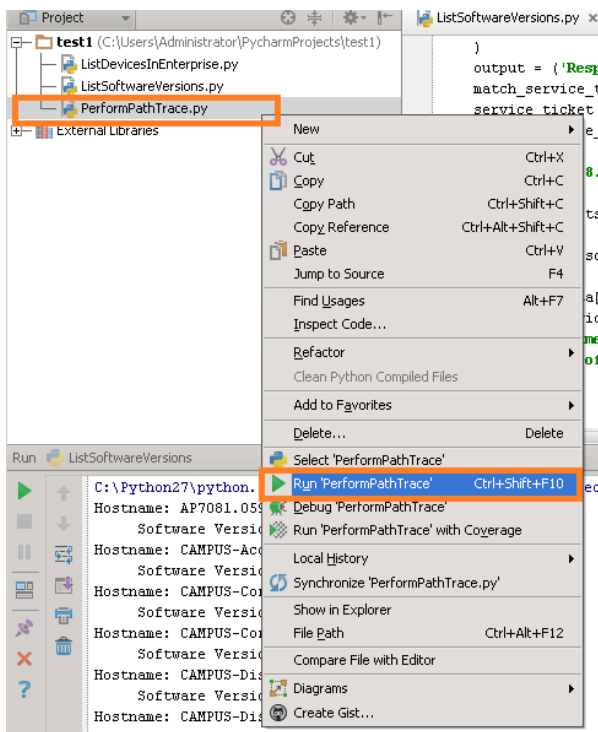
Interface Name	MAC Address	Status
GigabitEthernet1/0/26	f0:29:29:5c:30:9a	Up
GigabitEthernet1/0/34	f0:29:29:5c:30:a2	Down
GigabitEthernet1/0/31	f0:29:29:5c:30:9f	Down
Vlan200	f0:29:29:5c:30:e2	Up
GigabitEthernet1/0/22	f0:29:29:5c:30:96	Down
Vlan500	f0:29:29:5c:30:d0	Up
GigabitEthernet1/0/47	f0:29:29:5c:30:af	Up
GigabitEthernet1/0/15	f0:29:29:5c:30:8f	Down
GigabitEthernet1/0/14	f0:29:29:5c:30:8e	Down
TenGigabitEthernet1/1/2	f0:29:29:5c:30:b6	Down

At the bottom of the interface table, there are navigation controls: '10 per page', '62 Interfaces', and '1 of 7'.

14. 单击 X 关闭窗口。

15. 返回 PyCharm 并运行 **PerformPathTrace.py** 脚本。

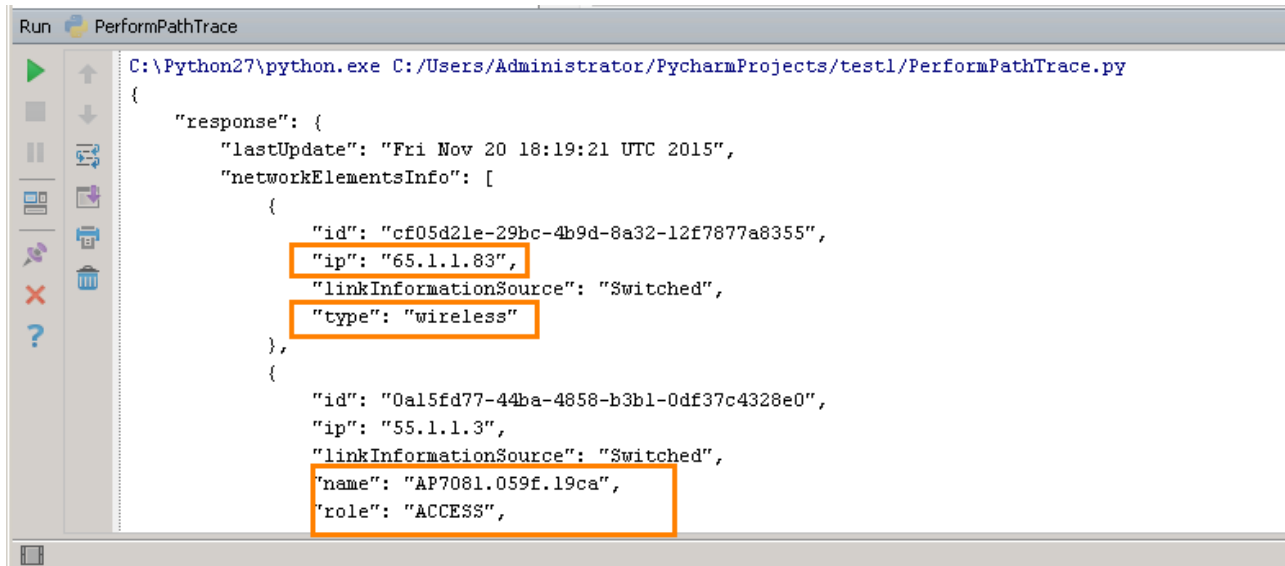
图 13. PerformPathTrace.py



The screenshot shows the PyCharm IDE interface. In the 'Project' view on the left, the file 'PerformPathTrace.py' is selected and highlighted with a red box. A context menu is open over this file, with the 'Run PerformPathTrace' option highlighted in blue. The menu also shows other options like 'Copy', 'Paste', 'Find Usages', and 'Refactor'. The 'Run' window at the bottom shows the output of the script, listing software versions for various hosts, including 'CAMPUS-Access1'.

16. 脚本在“Run（运行）”窗口中执行，并显示通过网络的路径流量，包括源 IP、网络类型和 AP 信息等。

图 14. 路径跟踪



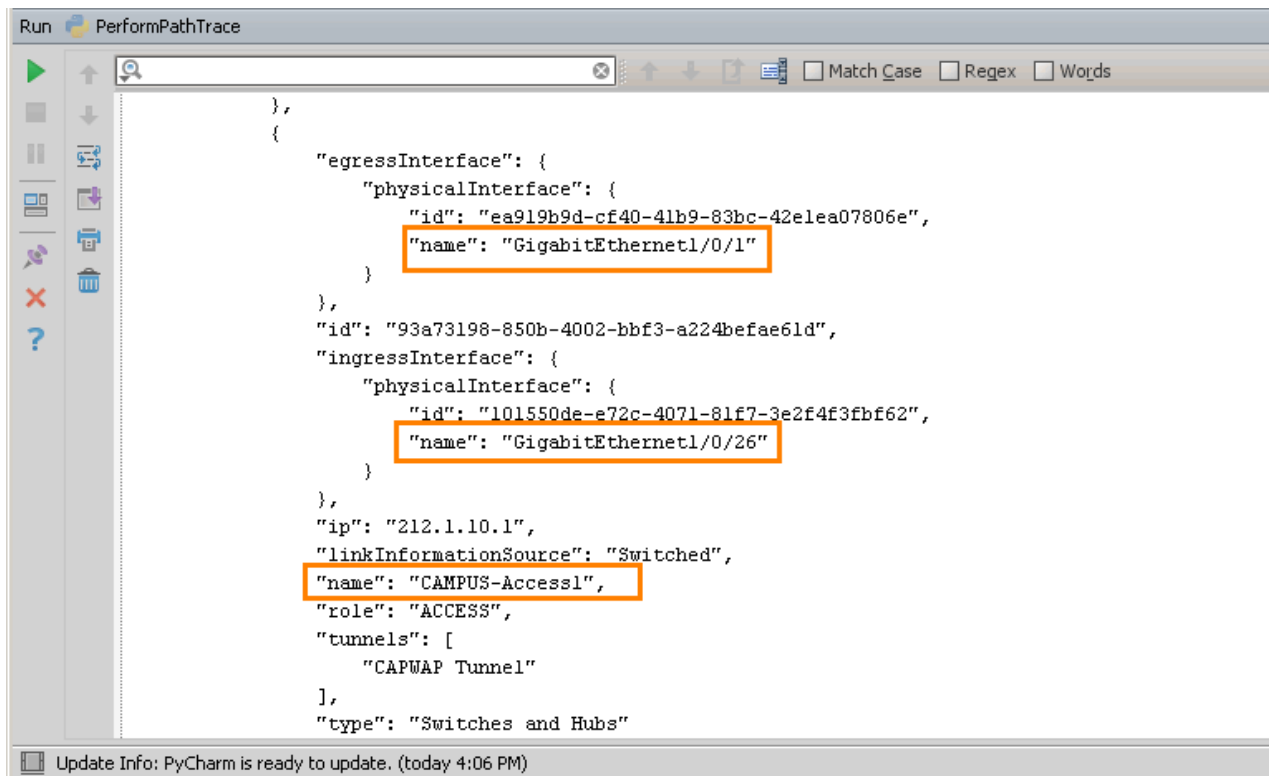
```

Run PerformPathTrace
C:\Python27\python.exe C:/Users/Administrator/PycharmProjects/test1/PerformPathTrace.py
{
  "response": {
    "lastUpdate": "Fri Nov 20 18:19:21 UTC 2015",
    "networkElementsInfo": [
      {
        "id": "cf05d21e-29bc-4b9d-8a32-12f7877a8355",
        "ip": "65.1.1.83",
        "linkInformationSource": "Switched",
        "type": "wireless"
      },
      {
        "id": "0a15fd77-44ba-4858-b3b1-0df37c4328e0",
        "ip": "55.1.1.3",
        "linkInformationSource": "Switched",
        "name": "AP7081.059f.19ca",
        "role": "ACCESS",
      }
    ]
  }
}

```

17. 向下滚动“Path Trace”（路径跟踪）窗口并指明 **egress and ingress interface**（传出和传入接口）信息。此时会显示流量进出该轨迹的物理接口。以图形方式展示路径跟踪时，信息应匹配。

图 15. 出口和入口信息



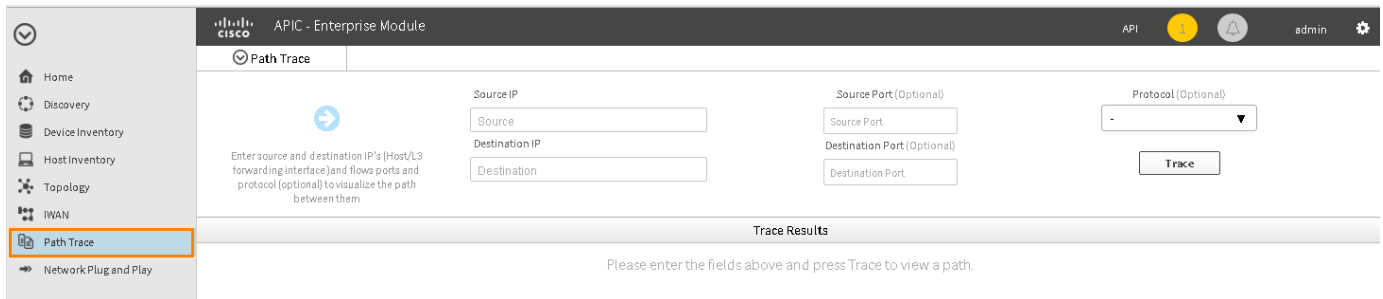
```

Run PerformPathTrace
},
{
  "egressInterface": {
    "physicalInterface": {
      "id": "ea919b9d-cf40-41b9-83bc-42e1ea07806e",
      "name": "GigabitEthernet1/0/1"
    }
  },
  "id": "93a73198-850b-4002-bbf3-a224befae61d",
  "ingressInterface": {
    "physicalInterface": {
      "id": "101550de-e72c-4071-81f7-3e2f4f3fbf62",
      "name": "GigabitEthernet1/0/26"
    }
  },
  "ip": "212.1.10.1",
  "linkInformationSource": "Switched",
  "name": "CAMPUS-Access1",
  "role": "ACCESS",
  "tunnels": [
    "CAPWAP Tunnel"
  ],
  "type": "Switches and Hubs"
}
Update Info: PyCharm is ready to update. (today 4:06 PM)

```

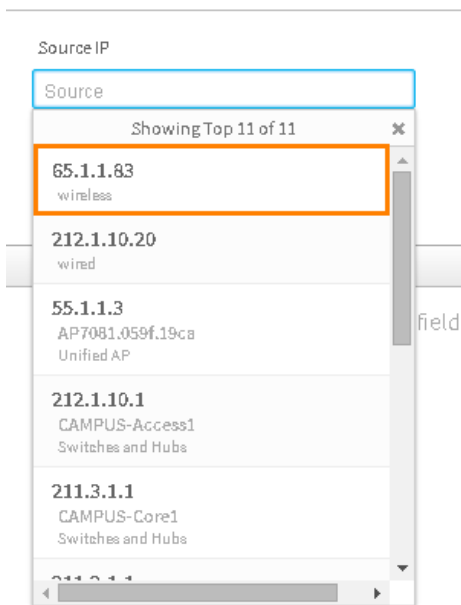
18. 返回 APIC-EM 窗口，然后单击左侧菜单中的 **Path Trace（路径跟踪）**。此时会显示“Path Trace”（路径跟踪）窗口。

图 16. “Path Trace”（路径跟踪）窗口



19. 在 **Source IP（源 IP）** 下拉列表中，选择 **Wireless client（无线客户端）**。

图 17. Wireless client（无线客户端）



20. 在 **DestinationIP**（目标 IP）下拉列表中，选择 **Wired client**（有线客户端）。

图 18. Wired client（有线客户端）

Source IP
65.1.1.83

Destination IP
Destination

Showing Top 11 of 11

65.1.1.83	wireless
212.1.10.20	wired
55.1.1.3	AP7081.059f.19ca Unified AP
212.1.10.1	CAMPUS-Access1 Switches and Hubs
211.3.1.1	CAMPUS-Core1 Switches and Hubs
211.3.1.1	

21. 点击 **Trace**（跟踪）。跟踪路径时，系统将打开一个对话框。

图 19. 查找路径

Path Trace

Enter source and destination IP's (Host/L3 forwarding interface) and flows ports and protocol (optional) to visualize the path between them

Source IP: 65.1.1.83

Destination IP: 212.1.10.20

Source Port (Optional): Source Port

Destination Port (Optional): Destination Port

Protocol (Optional): -

Trace

Trace Results

Finding path

||

./_tasks

This may take a minute

22. “Path Trace”（路径跟踪）窗口显示相同设备。

图 20. 路径跟踪

APIC - Enterprise Module

Path Trace | IPs: 65.1.1.83 → 212.1.10.20

Trace Results

View Small | Show Reverse | Scroll Lock | Show Duplicate Devices | View in Topology

65.1.1.83 (wireless) → AP7081.059f.19ca (Unified AP) → CAMPUS-Access1 (Switches and Hubs) → CAMPUS-Dist1 (Switches and Hubs) → Campus-WLC-5508 (WLC) → CAMPUS-Dist1 (Switches and Hubs) → CAMPUS-Access1 (Switches and Hubs) → 212.1.10.20

65.1.1.83
IP: 65.1.1.83
Type: wireless
Link Source: Switched

AP7081.059f.19ca
IP: 55.1.1.3
Type: Unified AP
Link Source: Switched
Tunnels: CAPWAP Tunnel

CAMPUS-Access1
IP: 212.1.10.1
Type: Switches and Hubs
Link Source: Switched
Tunnels: CAPWAP Tunnel
Ingress Interface: GigabitEthernet1/0/26
Egress Interface: GigabitEthernet1/0/1

CAMPUS-Dist1
IP: 55.1.1.100
Type: Switches and Hubs
Link Source: Switched
Tunnels: CAPWAP Tunnel

23. 点击 **CAMPUS-Access1** 并指明传入和传出接口信息相匹配。

图 21. CAMPUS-Access1 入口和出口

APIC - Enterprise Module

Path Trace | IPs: 65.1.1.83 → 212.1.10.20

Trace Results

View Small | Show Reverse | Scroll Lock | Show Duplicate Devices | View in Topology

65.1.1.83 (wireless) → AP7081.059f.19ca (Unified AP) → **CAMPUS-Access1 (Switches and Hubs)** → CAMPUS-Dist1 (Switches and Hubs) → Campus-WLC-5508 (WLC) → CAMPUS-Dist1 (Switches and Hubs) → CAMPUS-Access1 (Switches and Hubs) → 212.1.10.20

CAMPUS-Access1
IP: 212.1.10.1
Type: Switches and Hubs
Link Source: Switched
Tunnels: CAPWAP Tunnel
Ingress Interface: GigabitEthernet1/0/26
Egress Interface: GigabitEthernet1/0/1

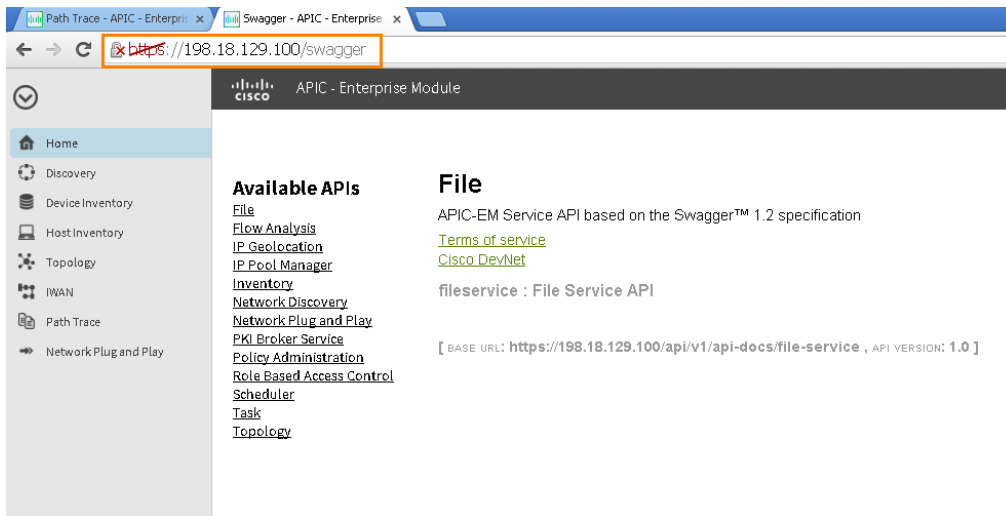
CAMPUS-Dist1
IP: 55.1.1.100
Type: Switches and Hubs

场景2. 通过 Swagger 探索 API

步骤

1. 打开浏览器并启动 **APIC-EM**。接受任何警告并使用 **userid admin**（用户名 **admin**）和 **password C1sco12345**（密码 **C1sco12345**）登录。
2. 点击打开新选项卡并浏览至 <https://198.18.129.100/swagger>。
3. 此时会显示可用 API。

图 22. 可用 API



4. 要添加用户，请依次点击 **Role Based Access Control**（基于角色的访问控制）> **User Management API**（用户管理 API）> **addUser**（添加用户）。

图 23. “Role Based Access Control”（基于角色的访问控制）> “User Management API”（用户管理 API）> “addUser”（添加用户）

The screenshot displays the Cisco APIC - Enterprise Module interface. The left sidebar lists various API categories, with "Role Based Access Control" highlighted. The main content area shows the "Role Based Access Control" configuration for the "User Management API". The "User" role is selected, and the "addUser" endpoint is highlighted in orange.

Method	Endpoint	Action
PUT	/user	updateUser
POST	/user	addUser
GET	/user	getUsers
GET	/user/passphrase/auto	getAutoPassphrase
GET	/user/passphrase/auto/{seedPhrase}	getAutoPassphrase
GET	/user/password-policy/invalid-attempt-count	getAttemptCount
PUT	/user/password-policy/invalid-attempt-count	updateAttemptCount
PUT	/user/password-policy/lock-expiry-time	updateLockExpiryTimeout
GET	/user/password-policy/lock-expiry-time	getLockExpiry
PUT	/user/status	updateUserStatus
GET	/user/status/{username}	getUserStatus
GET	/user/{username}	getUser
DELETE	/user/{username}	deleteUser

5. 点击“Data Type”（数据类型）下方的 **Model Schema（模型方案）**，然后点击代码以填写新用户的“Value”（值）参数。

图 24. 模型方案

POST /user addUser

Implementation Notes
This method is used to add a new user. The password is excluded from the json schema below, but will still need to be included.

Response Class
Model | Model Schema

SuccessResult {
 version (string, optional),
 response (string, optional)
}

Response Content Type: application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
user	<pre>{ "username": "", "authorization": [{ "scope": "", "role": "" }] }</pre>	user	body	Model Model Schema

Parameter content type: application/json

Error Status Codes

HTTP Status Code	Reason
200	This Request is OK
202	This Request is Accepted
403	This user is Forbidden Access to this Resource
401	Not Authorized Yet, Credentials to be supplied
404	No Resource Found

Try it out!

6. 在空白“Value”（值）参数的问号间输入以下信息：

- 用户名：test
- 密码：test123@
- 范围：ALL
- 角色：ROLE_ADMIN

注意：添加密码行，严格按照：“password”: “test123@”, 格式 - 请勿忘记在行末添加逗号。

7. 点击 Try it out! (试用!)

图 25. “Value” (值) 参数

Parameters

Parameter	Value
user	<pre style="border: 1px solid #ccc; padding: 5px;">{ "username": "test", "password": "test123@", "authorization": [{ "scope": "ALL", "role": "ROLE_ADMIN" }] }</pre> <p>Parameter content type: application/json</p>

Error Status Codes

HTTP Status Code	Reason
200	This Request is OK
202	This Request is Accepted
403	This user is Forbidden Access to this Resource
401	Not Authorized Yet, Credentials to be supplied
404	No Resource Found

Try it out!
[Hide Response](#)

8. 用户创建成功。点击添加用户行中的 **POST** 隐藏 AddUser 信息。

图 26. 用户创建成功

Request URL

https://198.18.129.100/api/v1/user

Response Body

```
{
  "response": "User created successfully",
  "version": "1.0"
}
```


Response Code

201

Response Headers

```
{
  "Pragma": "no-cache",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Frame-Options": "SAMEORIGIN",
  "Content-Type": "application/json; charset=UTF-8",
  "Access-Control-Allow-Origin": "https://198.18.129.100",
  "Access-Control-Expose-Headers": "",
  "Cache-Control": "no-cache, no-store",
  "Access-Control-Allow-Credentials": "false",
  "Connection": "close"
}
```

9. 依次点击 **Role Based Access Control（基于角色的访问控制）** > **user: User Management API（用户：用户管理 API）** > **Show/Hide（显示/隐藏）**。

图 27. “Role Based Access Control”（基于角色的访问控制） > “user: User Management API”（用户：用户管理 API） > “Show/Hide”（显示/隐藏）

Available APIs

- [File](#)
- [Flow Analysis](#)
- [IP Geolocation](#)
- [IP Pool Manager](#)
- [Inventory](#)
- [Network Discovery](#)
- [Network Plug and Play](#)
- [PKI Broker Service](#)
- [Policy Administration](#)
- Role Based Access Control**
- [Scheduler](#)
- [Task](#)
- [Topology](#)

Role Based Access Control

APIC-EM Service API based on the Swagger™ 1.2 specification

[Terms of service](#)
[Cisco DevNet](#)

role : Role Description API		Show/Hide	List Operations	Expand Operations	Raw
ticket : Ticket Management API		Show/Hide	List Operations	Expand Operations	Raw
user : User Management API		Show/Hide	List Operations	Expand Operations	Raw
PUT	/user				updateUser
POST	/user				addUser
GET	/user				getUsers
GET	/user/passphrase/auto				getAutoPassphrase
GET	/user/passphrase/auto/{seedPhrase}				getAutoPassphrase
GET	/user/password-policy/invalid-attempt-count				getAttemptCount
PUT	/user/password-policy/invalid-attempt-count				updateAttemptCount
PUT	/user/password-policy/lock-expiry-time				updateLockExpiryTimeout
GET	/user/password-policy/lock-expiry-time				getLockExpiry
PUT	/user/status				updateUserStatus
GET	/user/status/{username}				getUserStatus
GET	/user/{username}				getUser
DELETE	/user/{username}				deleteUser

10. 点击 **Delete（删除）** 显示如何删除用户。
11. 在用户名值中输入 **test**，然后点击 **Try it out!（试用！）** 从数据库中删除用户。

图 28. 删除用户

DELETE
/user/{username}
deleteUser

Implementation Notes
 This method is used to delete a user.
 Admin permission is required.
 It is possible to delete your own user.

Response Class
 Model | Model Schema

SuccessResult {
 version (string, optional),
 response (string, optional)
}

Response Content Type: application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
username	test	username	path	string

Error Status Codes

HTTP Status Code	Reason
200	This Request is OK
403	This user is Forbidden Access to this Resource
401	Not Authorized Yet, Credentials to be supplied
404	No Resource Found

Try it out!
Hide Response

图 29. 删除确认消息

Request URL

https://198.18.129.100/api/v1/user/test

Response Body

```
{
  "response": "User deleted successfully",
  "version": "1.0"
}
```

Response Code

200

Response Headers

```
{
  "Pragma": "no-cache",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Frame-Options": "SAMEORIGIN",
  "Content-Type": "application/json;charset=UTF-8",
  "Access-Control-Allow-Origin": "https://198.18.129.100",
  "Access-Control-Expose-Headers": "",
  "Cache-Control": "no-cache, no-store",
  "Access-Control-Allow-Credentials": "false",
  "Connection": "close"
}
```

12. 返回 **APIC-EM** 窗口，然后探索任何其他 API 选项。

图 30. 可用 API

Available APIs

[File](#)

[Flow Analysis](#)

[IP Geolocation](#)

[IP Pool Manager](#)

[Inventory](#)

[Network Discovery](#)

[Network Plug and Play](#)

[PKI Broker Service](#)

[Policy Administration](#)

[Role Based Access Control](#)

[Scheduler](#)

[Task](#)

[Topology](#)

附录 A. 通过公共 IP 连接

作为使用 wkst1 连接和运行 pyCharm 的替代方案，可以通过公共 IP 连接至 APIC-EM。这让您能够写入自己的自定义脚本并能够在自身开发环境中测试 API。

步骤

1. 从“Session Details”（会话详情）选项卡中检索 APIC-EM 的 **Public IP Address**（公用 IP 地址）。

图 31. 公用 IP 地址

Session Details

Session ID: 211142
 End Time : 12/11/2015 20:00:00
 Owner: rgiorgi ← 用户名
 Last Modified: 12/2/2015
 VPN Available: true
 Session Devices: N/A
 Virtual Center: 1

Public Address	Private Address	Description
64.100.10.129 ← 公用 IP 地址	198.18.129.100	apic-em-path

2. 将该 IP 用作主机，开始构建脚本。使用在“Session Details”（会话详情）选项卡中作为用户名的所有者 ID 和密码 C1sco12345 登录。

```

16  ### Create Service Ticket.
17  def createserviceticket():
18      response = requests.post(
19          url="https://64.100.10.95/api/v1/ticket",
20          headers={
21              "Content-Type": "application/json",
22          },
23          verify=False,
24          data=json.dumps({
25              "username": 'dschuema',
26              "password": 'C1sco12345'
27          })
28      )
29      output = ('Response HTTP Response Body: {content}'.format(content=response.content))
30      match_service_ticket = re.search('serviceTicket': "(.*cas)", output, flags=0)
31      service_ticket = match_service_ticket.group(1)
32      return service_ticket
  
```



美洲总部
Cisco Systems, Inc.
加州圣何西

亚太地区总部
Cisco Systems (USA) Pte.Ltd.
新加坡

欧洲总部
Cisco Systems International BV
荷兰阿姆斯特丹

思科在全球设有 200 多个办事处。地址、电话号码和传真号码均列在思科网站 www.cisco.com/go/offices 中。

思科和思科徽标是思科和/或其附属公司在美国和其他国家或地区的商标或注册商标。有关思科商标的列表，请访问此 URL：www.cisco.com/go/trademarks。本文提及的第三方商标均归属其各自所有者。使用“合作伙伴”一词并不暗示思科和任何其他公司存在合伙关系。(1110R)