atmosphere'22

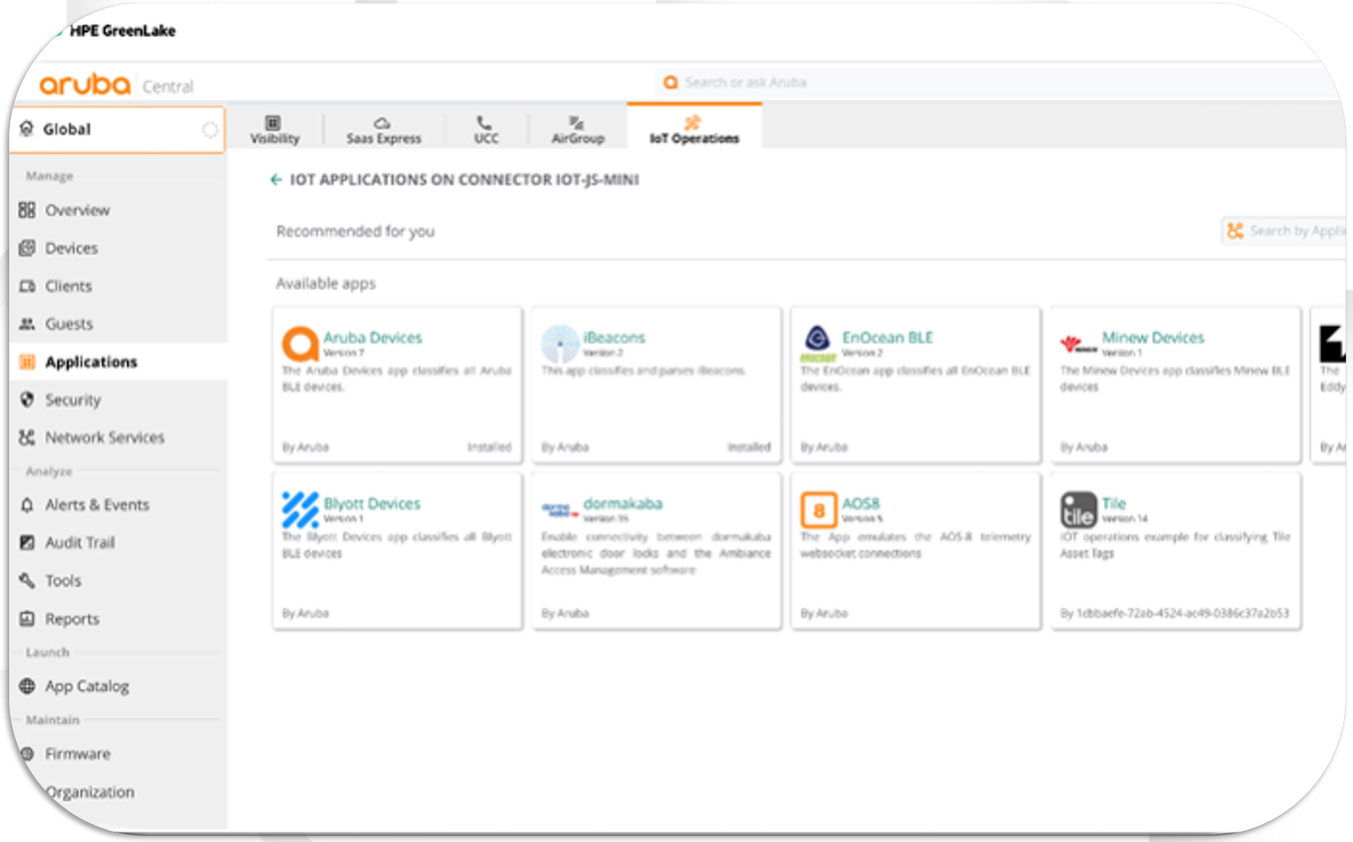
MAKING CONNECTIONS, ANYWHERE

如何借助Central Rest API、Streaming API及Webhook实现第三方集成及网络自动化

Aruba大客户技术顾问 李丽军

April 2022

aruba
a Hewlett Packard Enterprise company

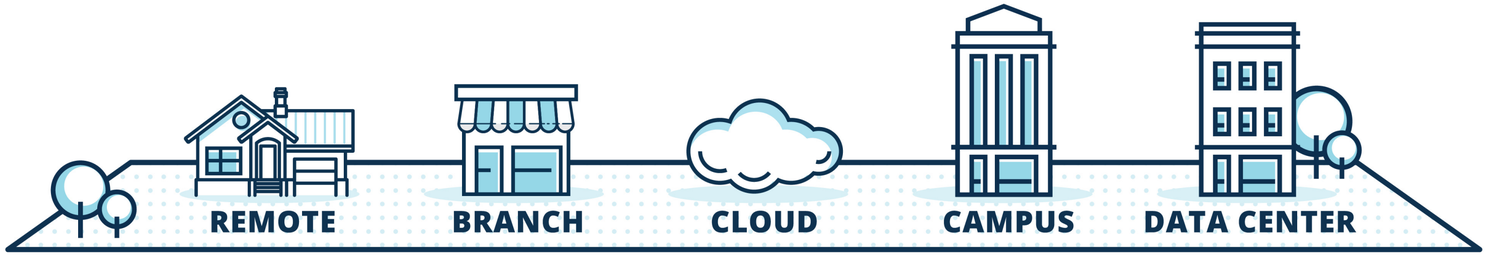


Aruba Central:

不仅仅是一个管理系统…

ARUBA 边缘服务平台(ESP)之下的

网络自动化



ARUBA CENTRAL

ANALYZE AND ACT

- Onboarding
- Provisioning
- Orchestration
- Analytics
- Location
- Management
- AIOps

PROTECT

- Visibilty
- Authentication
- Continuous Monitoring
- Policy Enforcement
- Zero Trust
- SASE
- Edge-to-Cloud Security

CONNECT

- Wireless
- Wired
- SD-WAN
- 5G
- IoT
- Unified Infrastructure

AS-A-SERVICE

API

API

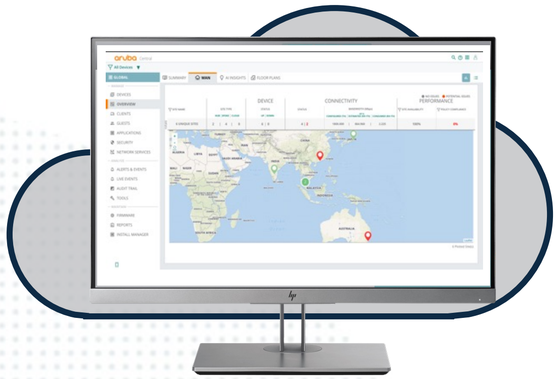
API

Aruba Central云管理平台

•通过统一管理和基于人工智能驱动的运维操作打破管理孤岛

云原生统一管理平台

REMOTE BRANCH CLOUD CAMPUS DATA CENTER



部署灵活性

SaaS On-Prem Managed Services

统一基础架构集中管理

APs Switches SD-WAN Gateways VPN

基于AI驱动的数据分析 & 自动化

Onboarding Provisioning Orchestration Analytics Contact Tracing

内置安全

Unified Threat Management Authentication Continuous Monitoring Policy Enforcement Visibility

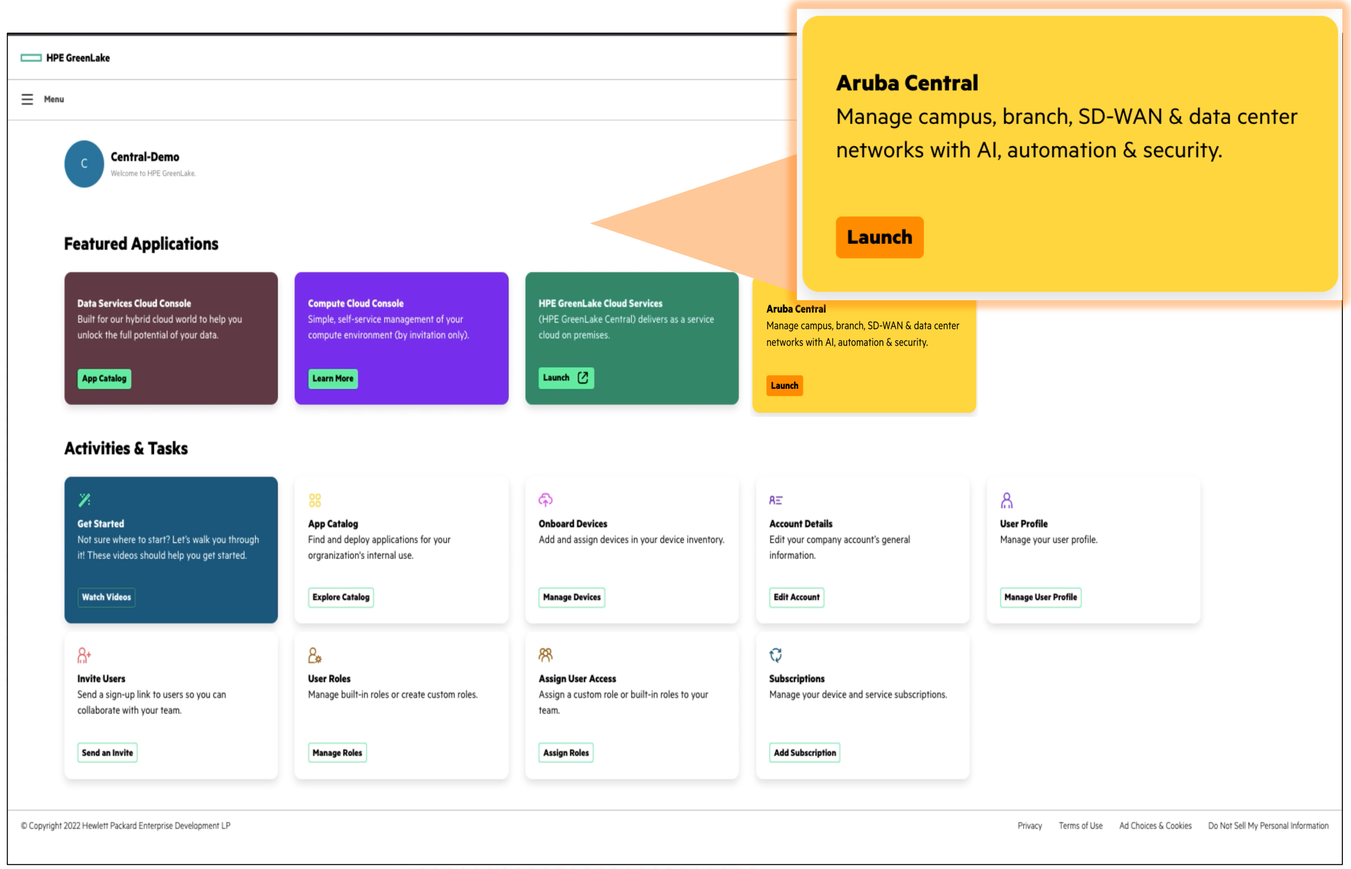
扩展性

Microservices APIs Webhooks

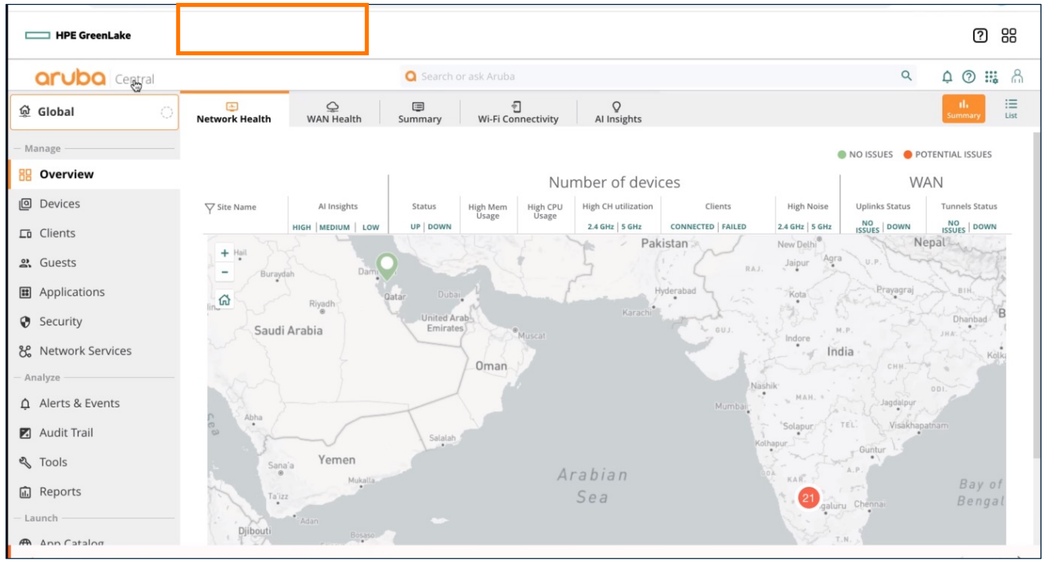
HPE Green Lake for Aruba

Aruba Central: HPE GreenLake 大解决方案的重要组成部分!

整合了HPE 计算、存储、网络服务的统一平台



New “HPE GreenLake” common header



覆盖有线、无线、广域网的AI运维能力

- 从被动故障排除转变为主动解决问题

AI 洞察力

基于动态基线、机器学习的异常自动探测、数据分析、自动化的修复建议

AI 搜索

基于数据湖的自然语言查询，简化搜索并大大减少问题平均解决时间

AI 协助

基于事件驱动的AI自动化，动态收集诊断信息、通过告警提醒管理员，并自动创建CASE

设备指纹识别

基于众包智能技术的精准设备类型识别提升IOT接入安全性

终端异常探测

自动化的终端应用访问性能监测

Aruba Central

(网络管理)

ClearPass Device Insight

(物联网安全)

User Experience Insight

(应用程序监控)

端到端 “自动化”和编排

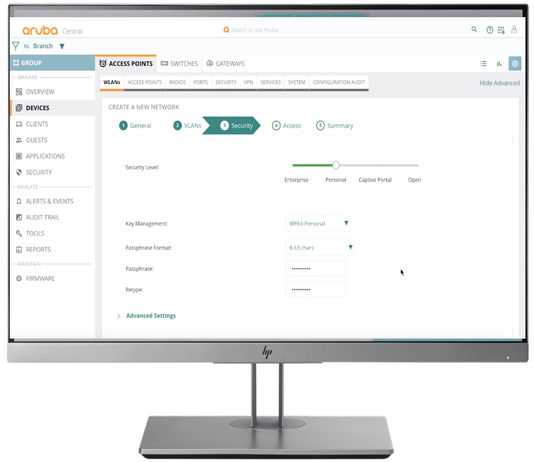
- 减少手动操作以提高整个网络生命周期的 IT 敏捷性

简化的设备注册、上线配置流程

移动app和设置向导用于设备快速 安装以及新网络设备的跟踪

真正的零接触配置，在新设备启动时自动将配置和站点特定策略推送到新设备

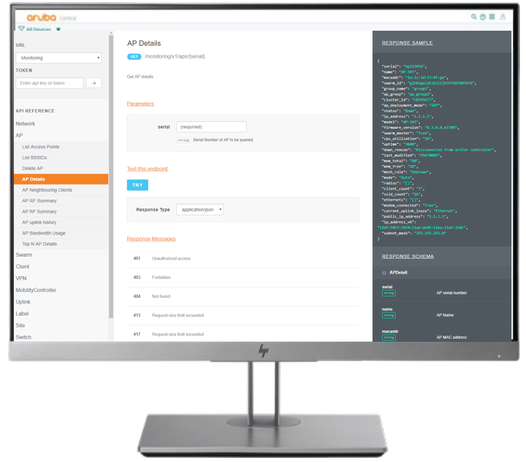
自动化、集中式的工作流程



配置或策略更改后，推送到所有目标设备，立即验证，确保配置更新不出错

live upgrades功能取消了传统设备升级时所需的维护时间窗口

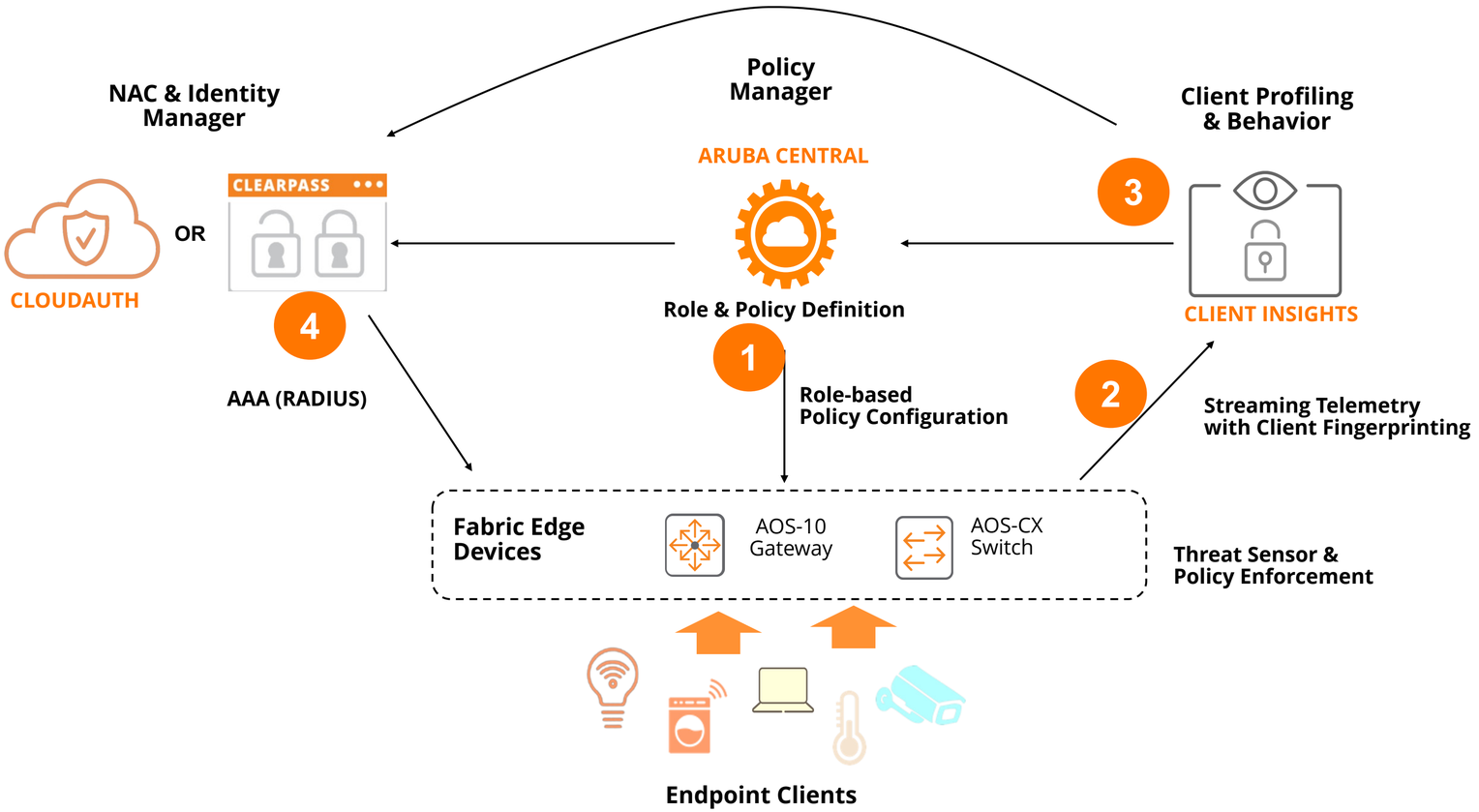
与第三方IT管理平台集成提供灵活的扩展性



Webhooks 和 API功能使Central与第三方工具集成更简单便捷

从配置到日志记录、报表的完整生命周期覆盖

Aruba Central NetConductor（策略中心）



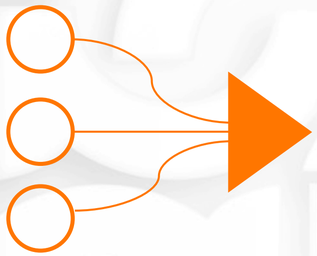
Aruba Central NetConductor Benefits

- ✓ Streaming telemetry automates all client discovery & profiling in network
- ✓ ML algorithms automate client classifications to automate policy assignments
- ✓ NAC services integration automates client admission control and service entitlements
- ✓ Continuous monitoring ensures client behavior conforms to the security policies

Aruba Central NetConductor 提供具有AI能力的零信任安全

自动化所带来的好处

简化操作



工作流程自动化

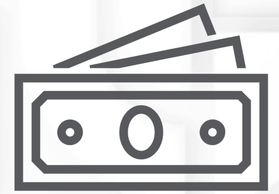
改进的可视化体验

网络设备可见性/终端可见性/应用访问可见性

增强业务敏捷性

无论任何场景，任何接入方式，部署都更简单、省时、有效率

成本节约



节省维护的人力及时间

系统可用性提升，影响业务的downtime大大减少

自动化可大大提升IT团队工作效率

ARUBA CENTRAL 所带来的网络自动化

1. 大规模的网络部署

2. 业务工作流的标准化集成

3. 实时监控及可视化

4. 常规运维操作

ARUBA CENTRAL 可扩展性

REST API

可用于监控、配置和故障排除功能的 API

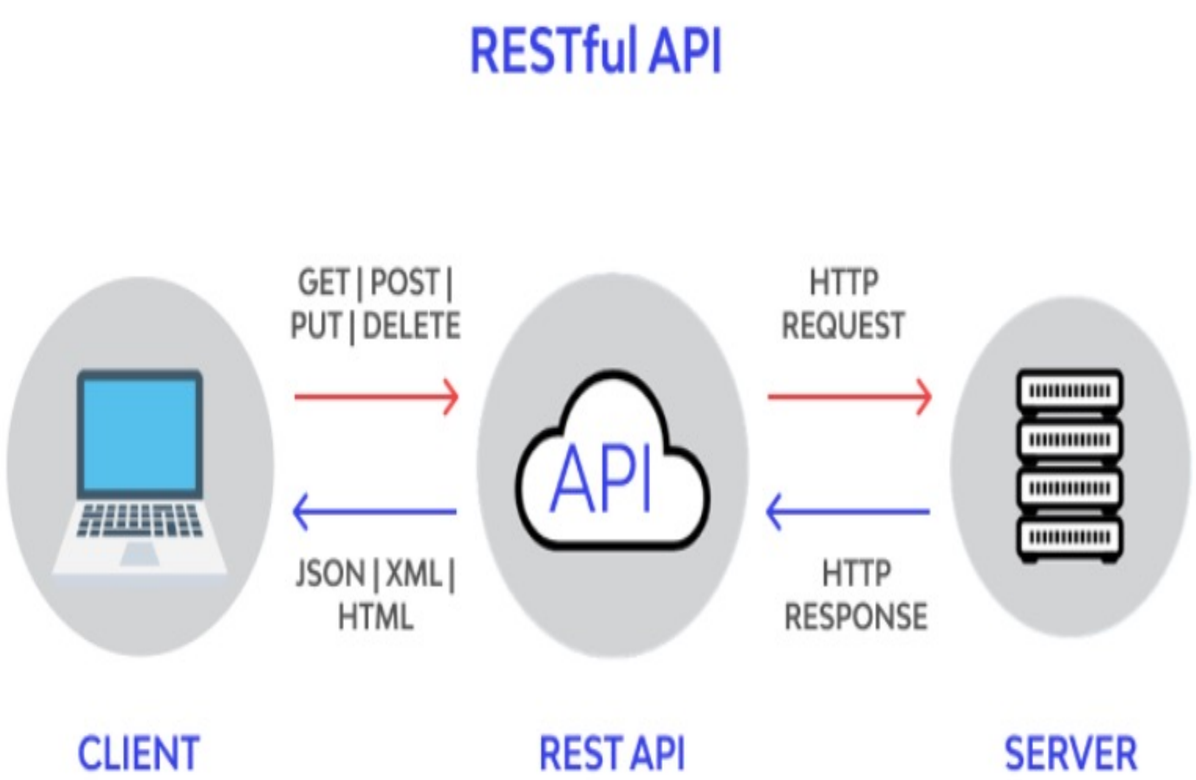
Webhooks

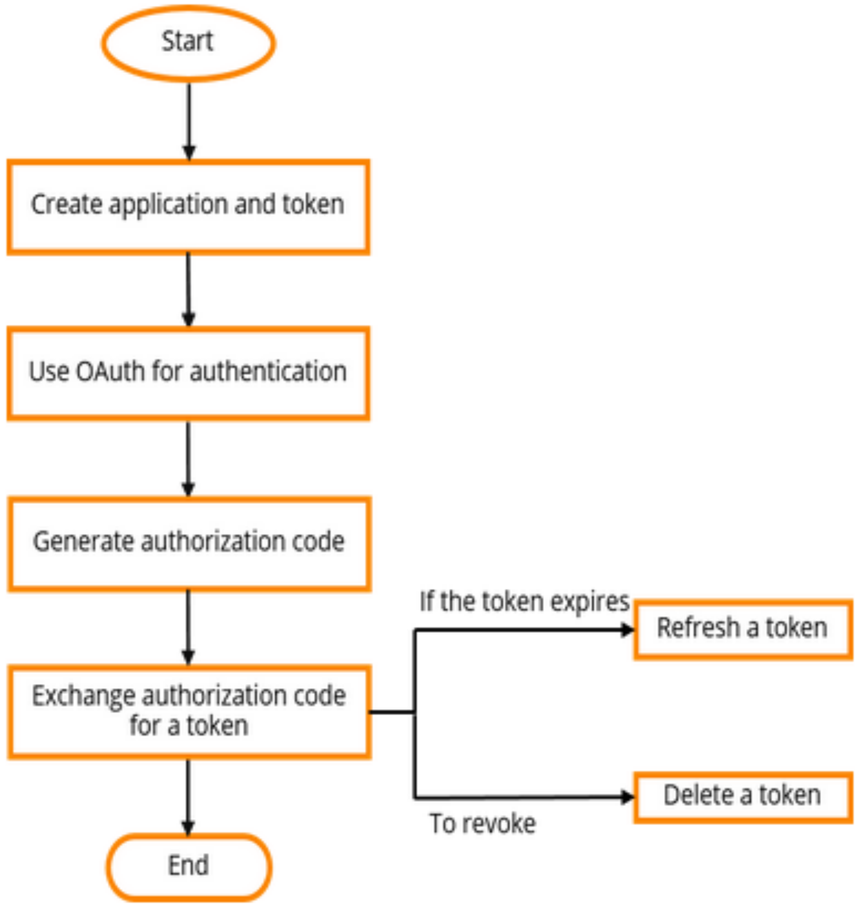
系统中发生的消息事件。对于事件驱动型的自动化很有用，如告警

API Streaming

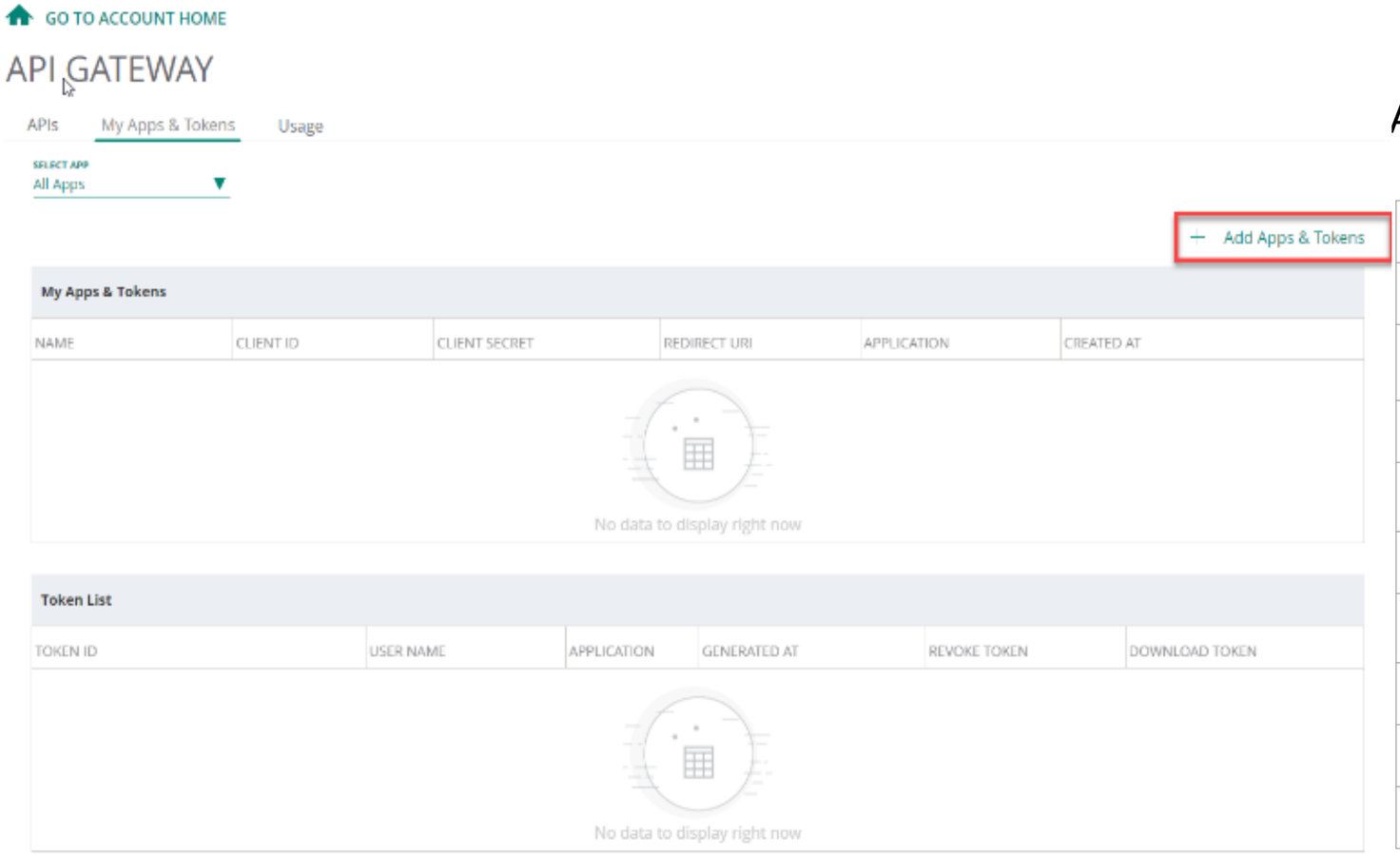
将整个网络的所有设备、用户状态和统计数据流式传输到第三方工具以进行实时监控.

Rest API Workflow





REST APIs配置

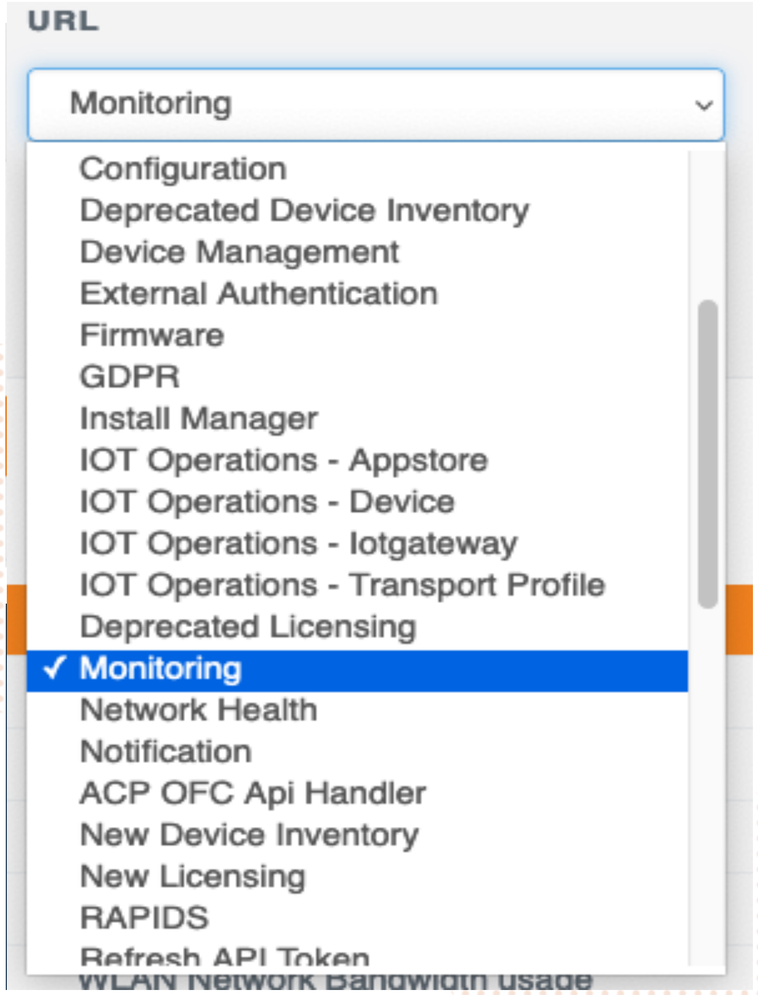


API Gateway

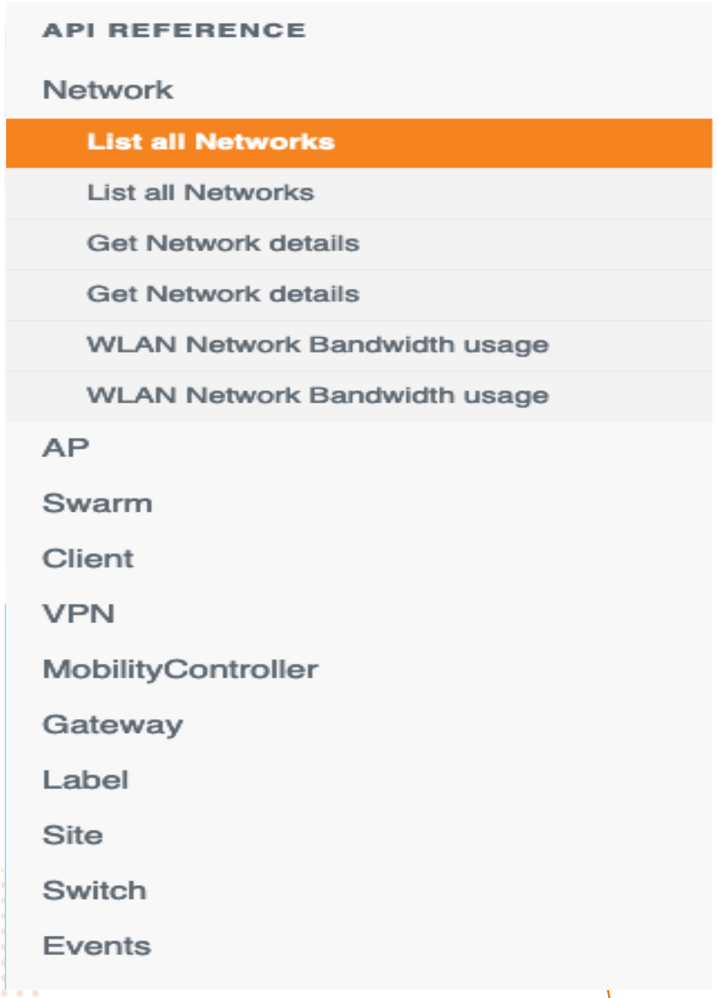
access_token	
appname	"nms"
authenticated_userid	
created_at	
credential_id	
expires_in	7200
id	
refresh_token	
scope	"all"
token_type	"bearer"

REST APIs 列表

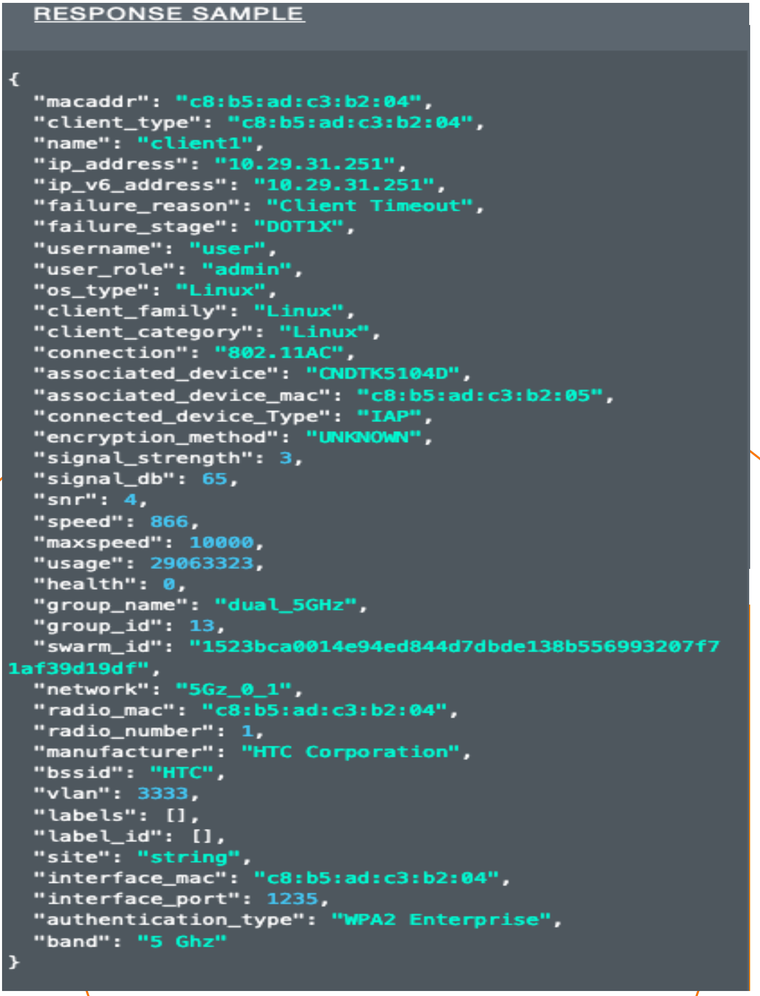
All Topics list



Monitoring



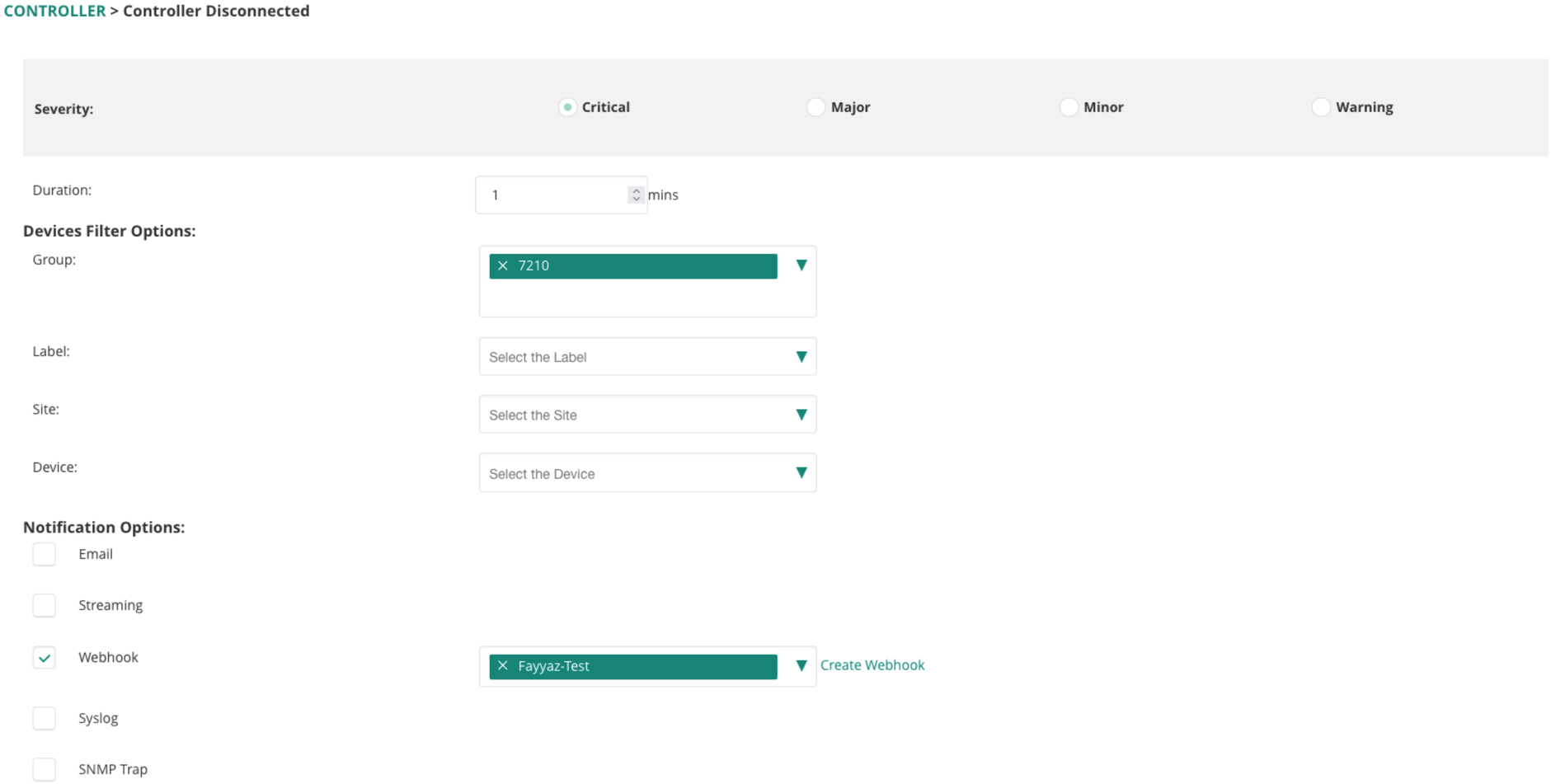
Sample output



ARUBA CENTRAL WEBHOOKS

1. User Alerts
2. Connectivity Alerts
3. Gateway Alerts
4. Switch Alerts
5. Access Point Alerts
6. Audit Alerts

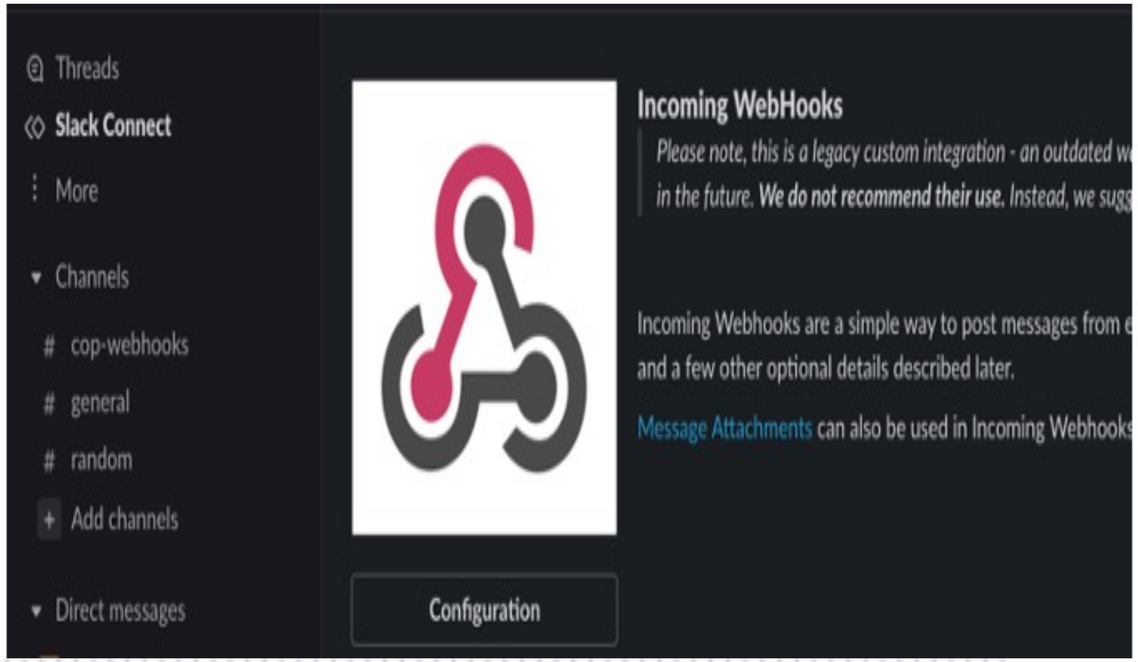
Webhook 集成 – 配置生成告警

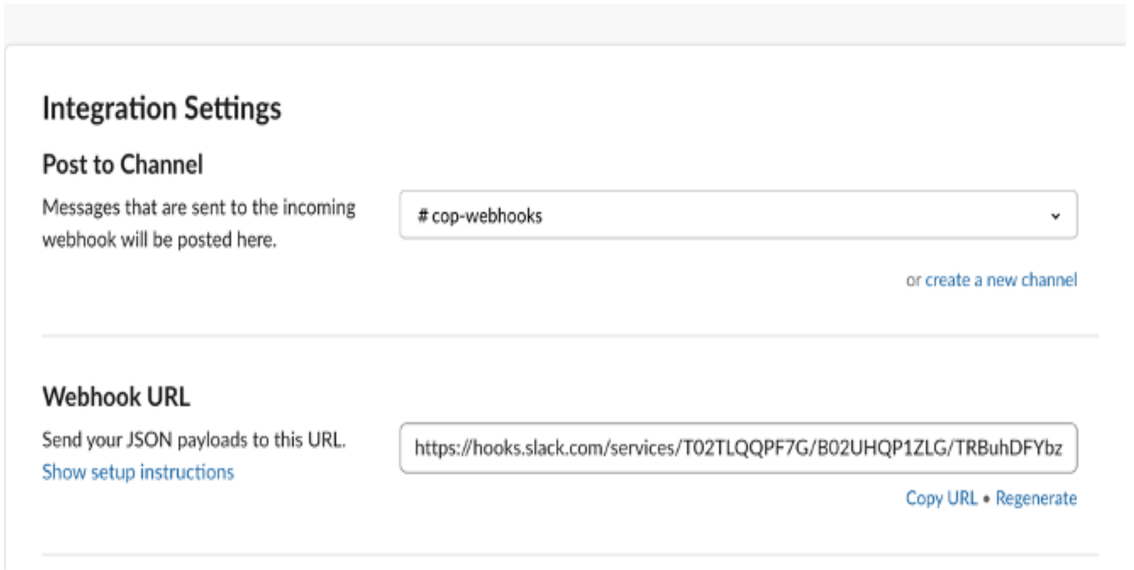


Webhook 第三方工具集成

- Central can integrate with any Webhooks supported Applications like Slack/Service now/Remedy etc.
- Which can be helpful in alerting the system admins, on time to take an action accordingly.

Webhook 集成 – Slack 配置示例





Install and add a Webhook application on your slack

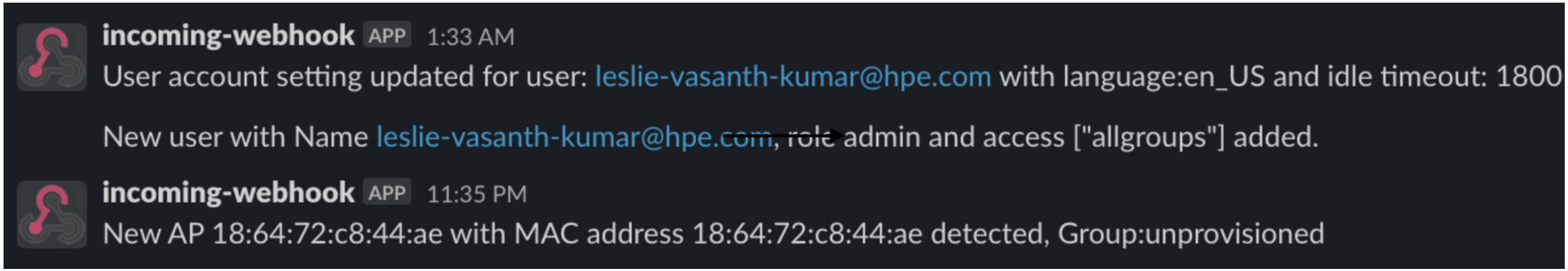
Create a slack channel

Configure the app to post the content to channel

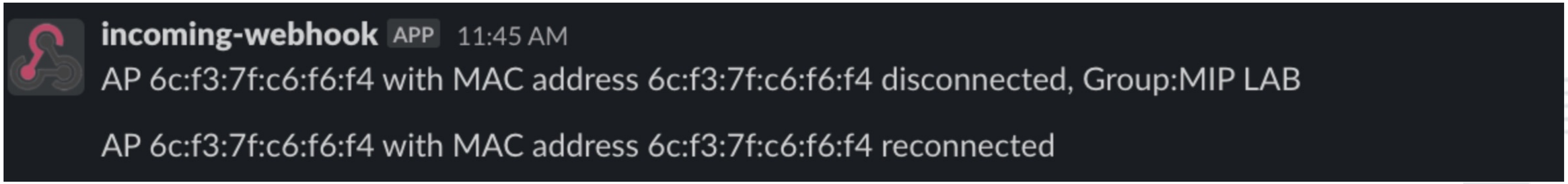
Copy the URL to use it on COP

Webhook 集成 – Slack消息输出示例

User Account edited and New AP Added Notification Alert



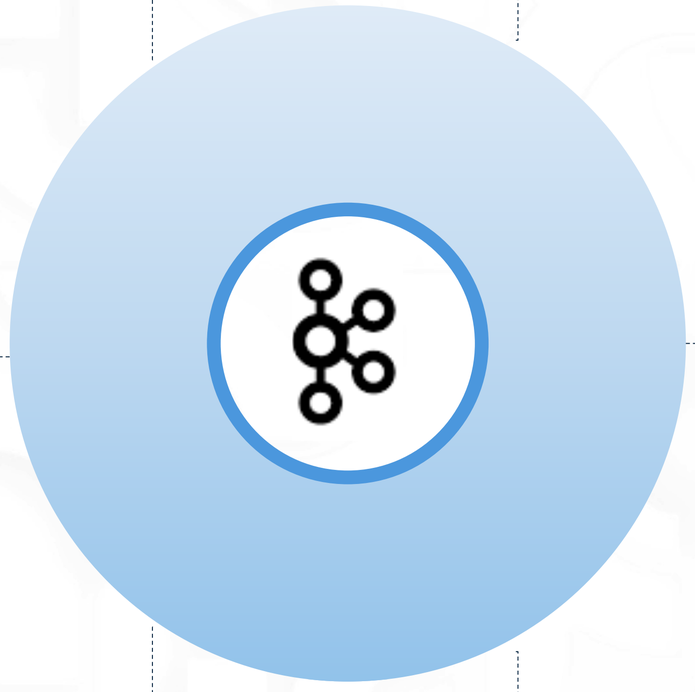
AP UP/Down Notification Alert



API STREAMING

Monitoring
State and Stats message of all devices

AppRF
Flow of all the client sessions



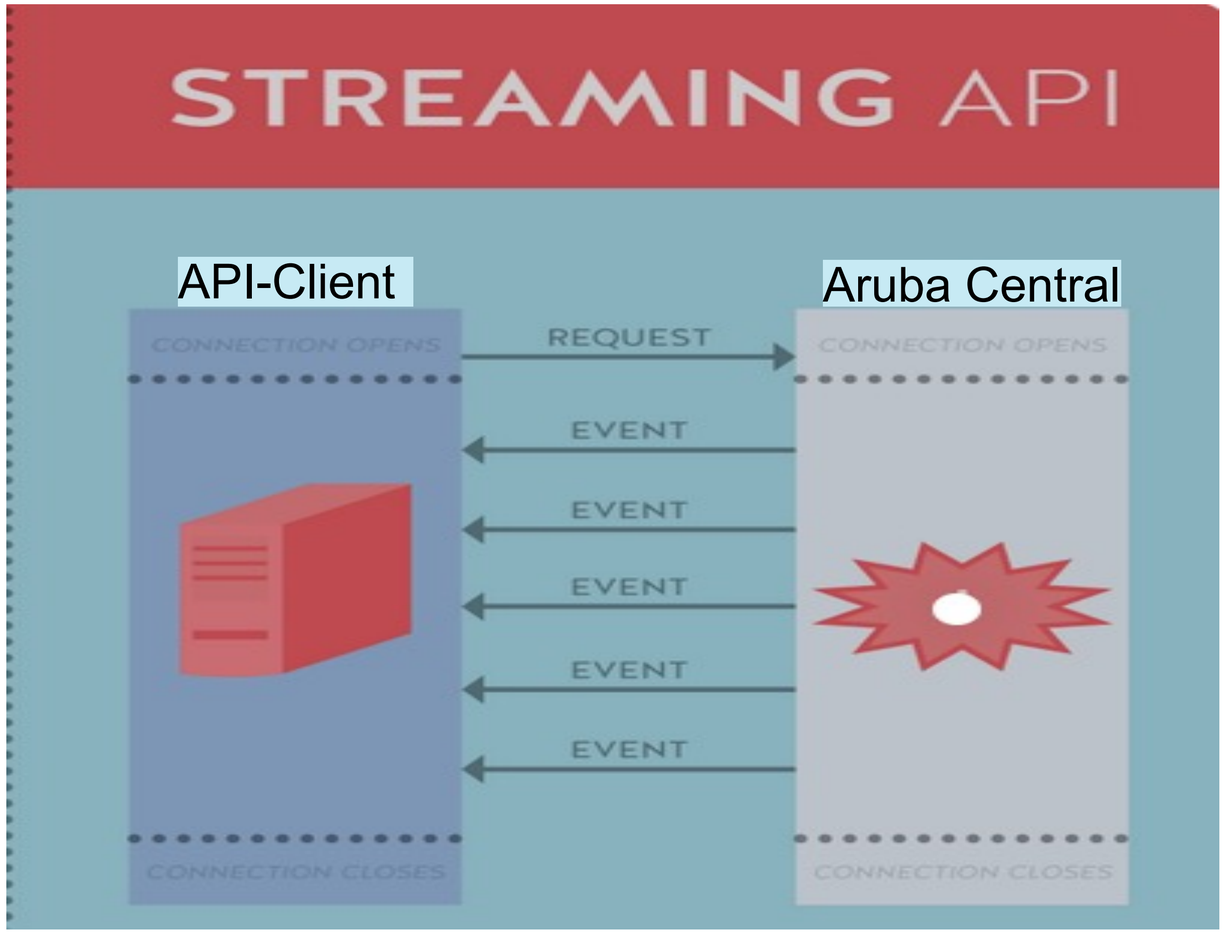
Security
Rogue AP & IDS detections

Location
Location of all wireless client computed using RSSI values

Audit
Device connectivity, config status, & firmware status

Presence
Setails of all associated & unassociated clients

Streaming APIs



ANY Streaming API client can establish WebSocket connection with Central and Stream the data.

We have a Streaming API python client that could be installed on any Linux machine, if you want to test and get familiarize with streaming APIs on COP.

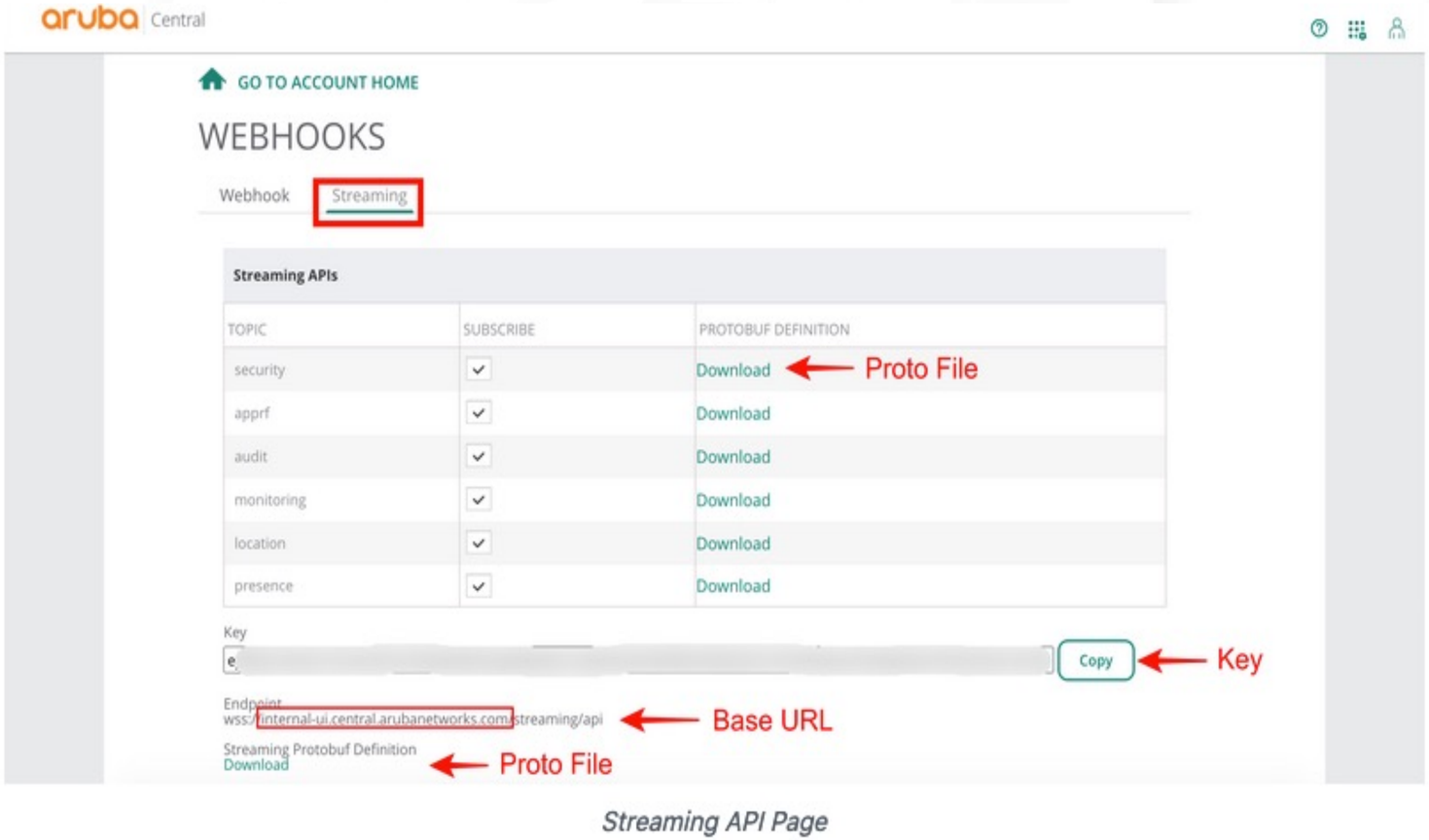
https://developer.arubanetworks.com

Streaming APIs 集成

In this example, we will be using simple_app.py script.

Step # 3 Gathering Requirements

- WebSocket URL Endpoint
- WebSocket Authorization key
- Subscription topic
- user email
- ProtoBuf files



Streaming API Page

Streaming APIs集成

Step # 4: We could edit the simple_app.py to update the server information

- WebSocket URL Endpoint
- WebSocket Authorization key
- Subscription topic
- user email
- ProtoBuf files

```
if __name__ == "__main__":
    # URL for WebSocket Connection from Streaming API page
    hostname = "cop-vip.nstaclab.com"
    url = "wss://{}/streaming/api".format(hostname)
    # Construct Header for WebSocket Connection
    header = {}
    # Central User email
    header["UserName"] = "admin@cop-vip.nstaclab.com"
    # WebSocket Key from Streaming API Page
    header["Authorization"] = "eyJhbGciOiJIUzI1NiIsInR5cGUiOiJqd3QifQ.eyJjdXN0b21]
fZGF0ZSI6MTY0Njk4MzUzNH0.5UwAxbpZufsCOUeuGFS5EI2X-vjLChooiRNRtKV-Oxg"
    # Subscription TOPIC for Streaming API
    # (audit|apprf|location|monitoring|presence|security)
    header["Topic"] = "monitoring"
    # Create WebSocket connection
    websocket.enableTrace(True)
    ws = websocket.WebSocketApp(url=url,
```

Streaming APIs 集成

Step # 5: We could now run the script from the same folder to stream the data:

python3 simple_app.py

example output:

```
----------------------
--- response header ---
HTTP/1.1 101 Switching Protocols
Date: Mon, 14 Mar 2022 22:19:25 GMT
Connection: upgrade
Upgrade: websocket
Sec-WebSocket-Accept: r9sVqj5AcwIbDTwQ6tEXqZ57RDw=
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-Frame-Options: SAMEORIGIN
X-Request-Start: t=1647296365.940
X-Using-Nginx-Controller: true
X-XSS-Protection: 1; mode=block
----------------------
Start Streaming Data!
++Rcv raw: b'\x82c\x12\nmonitoring\x1a)\n c1489b3b8486488984bc88035c1fdc83\xb8\x02\xb6\xfe\xbe\x91\x06 \x9a\xbc\x89\xf3\xc3\xe8\x97\xee\x16* c1489b3b
8486488984bc88035c1fdc83'
++Rcv decoded: fin=1 opcode=2 data=b'\x12\nmonitoring\x1a)\n c1489b3b8486488984bc88035c1fdc83\xb8\x02\xb6\xfe\xbe\x91\x06 \x9a\xbc\x89\xf3\xc3\xe8\x9
7\xee\x16* c1489b3b8486488984bc88035c1fdc83'
Timestamp in Epoch: 1647296310419807770
Customer_ID: c1489b3b8486488984bc88035c1fdc83
customer_id: "c1489b3b8486488984bc88035c1fdc83"
timestamp: 1647296310

++Rcv raw: b'\x82~\x00\x8a\x12\nmonitoring\x1aP\n c1489b3b8486488984bc88035c1fdc83\x10\rj#\n\tBA0000128\x10\x90\xff\xbe\x91\x06\x18\xe0\x837 \x03(\x8
0\x80\xa8\xdb\x130\x80\x80\x98\xf7\x0c\xb8\x02\xf2\xfe\xbe\x91\x06 \x86\x89\xe0\xcb\xa3\xea\x97\xee\x16* c1489b3b8486488984bc88035c1fdc83'
++Rcv decoded: fin=1 opcode=2 data=b'\x12\nmonitoring\x1aP\n c1489b3b8486488984bc88035c1fdc83\x10\rj#\n\tBA0000128\x10\x90\xff\xbe\x91\x06\x18\xe0\x8
37 \x03(\x80\x80\xa8\xdb\x130\x80\x80\x98\xf7\x0c\xb8\x02\xf2\xfe\xbe\x91\x06 \x86\x89\xe0\xcb\xa3\xea\x97\xee\x16* c1489b3b8486488984bc88035c1fdc83'
Timestamp in Epoch: 1647296370466882694
Customer_ID: c1489b3b8486488984bc88035c1fdc83
customer_id: "c1489b3b8486488984bc88035c1fdc83"
data_elements: STAT_DEVICE
device_stats {
  device_id: "BA0000128"
  timestamp: 1647296400
  uptime: 901600
  cpu_utilization: 3
  mem_total: 5291769856
  mem_free: 3471179776
```

Aruba devHub

devhub.arubanetworks.com

atmosphere'22

MAKING CONNECTIONS, ANYWHERE

Thank you

April 2022

aruba
a Hewlett Packard
Enterprise company