

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO MÔN DỰ ÁN CNTT 1
KHAI THÁC API TỪ NETWORK CONTROLLER APIC-EM
BẰNG PHẦN MỀM POSTMAN VÀ
THƯ VIỆN REQUESTS CỦA NGÔN NGỮ PYTHON**

Người hướng dẫn tại doanh nghiệp: THÀY ĐẶNG QUANG MINH

Người phụ trách môn học: GV. DƯƠNG HỮU PHÚC

Người thực hiện: LÊ THANH HẢI – 51800279

Lớp : 18050401

Khoá : 22

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO MÔN DỰ ÁN CNTT 1
KHAI THÁC API TỪ NETWORK CONTROLLER APIC-EM
BẰNG PHẦN MỀM POSTMAN VÀ
THU VIỆN REQUESTS CỦA NGÔN NGỮ PYTHON

Người hướng dẫn tại doanh nghiệp: THÀY ĐẶNG QUANG MINH

Người phụ trách môn học: GV. DƯƠNG HỮU PHÚC

Người thực hiện: LÊ THANH HẢI – 51800279

Lớp : 18050401

Khoa : 22

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

LỜI CẢM ƠN

Trong quá trình học tập và rèn luyện tại Trung tâm tin học VnPro, em xin bày tỏ lòng biết ơn đến thầy Đặng Quang Minh và các anh chị ở trung tâm đã tạo điều kiện để em có thể hoàn thành môn học cũng như là có được nhiều kinh nghiệm thực tế.

Đặc biệt, chúng em xin gửi lời cảm ơn chân thành đến thầy Đặng Quang Minh đã giảng dạy chúng em trong suốt quá trình thực tập tại Trung tâm. Thầy đã truyền đạt cho chúng em nhiều kiến thức. Không những dạy những kiến thức trong chuyên ngành mà còn những kiến thức về kỹ năng mềm, cách làm việc trong một tổ chức. Cảm ơn thầy đã hướng dẫn, góp ý để chúng em có thể hoàn thành bài báo cáo. Chúng em xin chúc thầy có thật nhiều sức khỏe, luôn thành công trong công việc. Em xin cảm ơn thầy.

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của Thầy Đặng Quang Minh giám đốc của Trung tâm tin học VnPro. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày tháng năm

Tác giả

(ký tên và ghi rõ họ tên)



Lê Thanh Hải

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

TÓM TẮT

Khi kỹ nguyên mới của khả năng lập trình mạng, tự động hóa và SDN đã đến, các kỹ sư mạng phải học cách viết code đã trở nên cần thiết. Mã hóa cho phép phát triển các công cụ phần mềm. Kết quả là, các kỹ sư mạng có thể tự động hóa các nhiệm vụ mạng phức tạp.

Giao tiếp giữa các công cụ phần mềm và nền tảng mạng xảy ra do các nhà sản xuất đã kích hoạt các giao diện lập trình ứng dụng hoặc API tốt hơn. Trong số các nền tảng mạng, API REST là một trong những nền tảng phổ biến nhất. Vì lý do này, điều quan trọng là phải biết cơ sở lý thuyết và thực tiễn của API này.

Trong bài báo cáo này em sẽ giới thiệu về các khái niệm, định nghĩa, các kiến thức nền tảng về API. Sau đó là sử dụng Northbound REST API vào kiến trúc SDN (Software-Defined Networking) gọi REST API để truy xuất các thiết bị được quản lý bởi Network Controller APIC-EM do VnPro cung cấp thông qua phần mềm Postman và thư viện request của ngôn ngữ Python.

MỤC LỤC

TÓM TẮT	iv
MỤC LỤC	1
DANH MỤC HÌNH	3
DANH MỤC BẢNG	5
DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT	6
CHƯƠNG 1 – MỞ ĐẦU.....	7
1.1 Giới thiệu tổng quan về doanh nghiệp	7
1.2 Lý do chọn đề tài	8
1.3 Mục tiêu đề tài.....	9
CHƯƠNG 2 – TỔNG QUAN ĐỀ TÀI.....	10
2.1 Tìm hiểu về API	10
2.1.1 Khái niệm API	10
2.1.2 Ví dụ về API	10
2.1.3 Phân loại API	12
2.1.4 Ưu điểm và nhược điểm của API	16
2.2 Tìm hiểu về REST API	17
2.3 Tìm hiểu về SDN (Software-Defined Network)	23
2.3.1 Định nghĩa SDN.....	23
2.3.2 Phân loại SDN	24
2.3.3 API ứng dụng vào SDN	25
CHƯƠNG 3 – THỰC NGHIỆM.....	27
3.1 Thiết kế sơ đồ mạng cấu hình cho thiết bị	27
3.2 Thêm thiết bị vào network controller APIC-EM	27
3.2.1 Tổng quan về network controller APIC-EM.....	27
3.2.2 Giao diện APIC-EM.....	28

3.2.3 Thêm thiết bị vào APIC-EM	31
3.3 Sử dụng Postman lấy ticket, số lượng, danh sách thiết bị, thông tin cấu hình..	38
3.3.1 Tổng quan về Postman	38
3.3.2 Sử dụng Postman lấy ticket, số lượng, danh sách thiết bị, thông tin cấu hình.....	40
3.4 Sử dụng thư viện requests lấy ticket, danh sách thiết bị	51
CHƯƠNG 4 – TỔNG KẾT	56
TÀI LIỆU THAM KHẢO.....	57
Bibliography	57

DANH MỤC HÌNH

Hình 1.1.1 Lộ trình đào tạo 1	7
Hình 1.1.2 Thiết bị tại VnPro 1	8
Hình 2.1.1 Minh họa API 1	10
Hình 2.1.2.1 Ví dụ API 1.....	11
Hình 2.1.2.2 Ví dụ đặt vé máy bay ứng dụng API 1	12
Hình 2.1.3.1 Sử dụng API cung cấp dịch vụ 1	14
Hình 2.1.3.2 Sử dụng API cung cấp thông tin 1	14
Hình 2.1.3.3 Sử dụng API ứng dụng phần cứng 1	15
Hình 2.2.1 Cách hoạt động của REST API 1	18
Hình 2.2.2 Các phương thức client sử dụng khi gọi API 1	19
Hình 2.3.1.1 Thành phần hạ tầng ứng dụng công nghệ SDN 1	24
Hình 3.1.1 Sơ đồ thiết bị mạng 1	27
Hình 3.2.2.1 Giao diện đăng nhập 1	28
Hình 3.2.2.2 Giao diện Dashboard 1	29
Hình 3.2.2.3 Giao diện Discovery 1	29
Hình 3.2.2.4 Giao diện Device Inventory 1	30
Hình 3.2.2.5 Giao diện host inventory 1	30
Hình 3.2.2.6 Giao diện Topology 1	31
Hình 3.2.3.1 Giao diện Dashboard 1	31
Hình 3.2.3.2 Giao diện thêm thiết bị vào APIC-EM 1	32

Hình 3.2.3.3 Kiểm tra IP thiết bị cấu hình SSH 1	33
Hình 3.2.3.4 Điền thông tin để thêm thiết bị 1	33
Hình 3.2.3.5 Điền username và password SSH 1.....	34
Hình 3.2.3.6 Nhập thông tin SSH và SNMP 1	35
Hình 3.2.3.7 Điền thông tin cấu hình SNMP 1	36
Hình 3.2.3.8 Bấm Start để bắt đầu tiến trình 1	36
Hình 3.2.3.9 Tiến trình đang thực hiện 1	37
Hình 3.2.3.10 Thực hiện thêm thông tin thiết bị vừa quét đc 1	38
Hình 3.2.3.11 Thiết bị đã được thêm vào 1	38
Hình 3.3.2.1 Tạo WorkSpaces 1.....	40
Hình 3.3.2.2 WorkSpace được tạo 1	41
Hình 3.3.2.3 Tạo Collection 1	41
Hình 3.3.2.4 Collection được tạo 1	41
Hình 3.3.2.5 Tạo Request xin ticket 1	42
Hình 3.3.2.6 Tài liệu API của network controller APIC-EM 1	42
Hình 3.3.2.7 Tạo môi trường để lưu các biến. 1	43
Hình 3.3.2.8 Tạo biến môi trường để gán các giá trị. 1.....	43
Hình 3.3.2.9 Điền URL 1	44
Hình 3.3.2.10 Điền kiểu dữ liệu trả về. 1	44
Hình 3.3.2.11 Khai báo thông tin username và password đăng nhập vào APIC-EM 1	45

Hình 3.3.2.12 Thông tin ticket được trả về. 1	46
Hình 3.3.2.13 Tạo biến môi trường để lưu giá trị Ticket. 1	46
Hình 3.3.2.14 Gán giá trị ticket cho biến vừa tạo 1	47
Hình 3.3.2.15 Điền thông tin URL để lấy thông tin thiết bị. 1.....	48
Hình 3.3.2.17 Gọi lại biến có giá trị ticket vừa lưu 1.....	49
Hình 3.3.2.18 Thực hiện gửi yêu cầu để lấy thông tin thiết bị trong APIC-EM. 1.....	49
Hình 3.3.2.19 Tạo requests yêu cầu đếm số lượng thiết bị. 1	49
Hình 3.3.2.20 Thay đổi đường dẫn để đếm số lượng thiết bị 1	50
Hình 3.3.2.21 Kết quả trả về số lượng thiết bị hiện có trong APIC-EM 1	50
Hình 3.3.2.22 Thay đổi đường dẫn để lấy thông tin cấu hình. 1	51
Hình 3.4.1 Kết quả Ticket được trả về 1	52
Hình 3.4.2 Danh sách thông tin thiết bị được trả về 1	54
Hình 3.4.3 Thông tin cấu hình của thiết bị 1	54
Hình 3.4.4 Danh sách Id được truy xuất. 1	55

DANH MỤC BẢNG

Bảng 2.2.1 Các phương thức của REST API 1	20
Bảng 2.2.2 Dữ liệu của Headers 1	21
Bảng 2.2.3 Một số mã trạng thái phổ biến 1	22
Bảng 2.2.4 Dữ liệu Headers Request 1	23
Bảng 2.2.5 Data format của dữ liệu trả về 1.....	23

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

CCNA: Cisco Certification Network Assosiation

CCNP: Cisco Certification Network Professional

CCIE: Cisco Certification Internet Expert

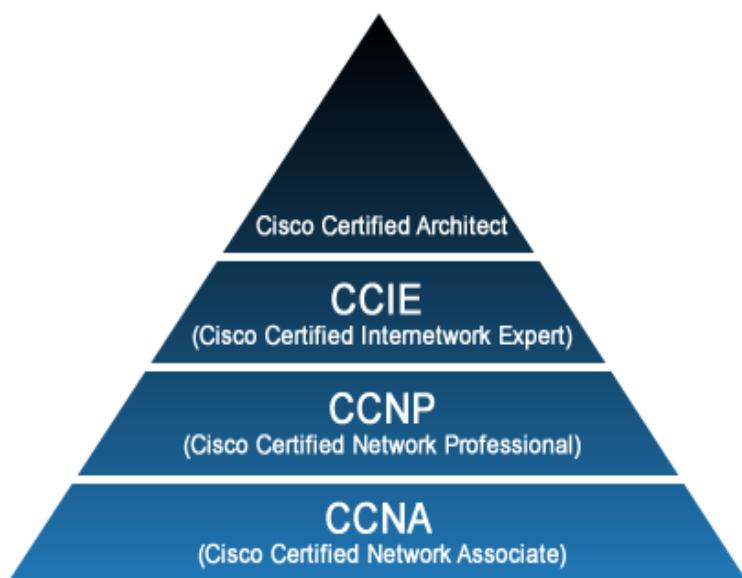
API: Application Programming Interfaces

SDN : Software –Defined Networking

CHƯƠNG 1 – MỞ ĐẦU

1.1 Giới thiệu tổng quan về doanh nghiệp

VnPro được thành lập vào tháng 3 năm 2003. VnPro Trung Tâm Đào Tạo Chuyên Gia Quản Trị Mạng. Với hơn 19 năm kinh nghiệm trong việc đào tạo và cung cấp nguồn nhân lực chất lượng cao trong lĩnh vực mạng. Chuyên tư vấn, thiết kế khóa học cho mọi đối tượng từ Sinh viên đến Doanh nghiệp. Đào tạo được hơn 30.000 học viên ở mọi cấp độ. Là trung tâm đào tạo tất cả các chứng chỉ Cisco (CCNA – CCNP – CCIE).



Hình 1.1.1 Lộ trình đào tạo 1

Hàng trăm thiết bị chính hãng hoạt động mỗi ngày phục vụ cho công việc nghiên cứu và giảng dạy tại VnPro.



Hình 1.1.2 Thiết bị tại VnPro 1

Giám đốc trung tâm người sáng lập ra VnPro là ông Đặng Quang Minh, một trong các CCSI và CCIE đầu tiên của Việt Nam với hoài bão chia sẻ, truyền đạt những kiến thức và kinh nghiệm của mình cho cộng đồng IT Việt Nam.

Lĩnh vực hoạt động của VnPro:

- Chuyên đào tạo quản trị mạng và hạ tầng Internet.
- Phát hành sách chuyên môn.
- Tư vấn và tuyển dụng nhân sự IT.
- Tư vấn và thiết kế hỗ trợ kỹ thuật hệ thống mạng

VnPro thường xuyên tổ chức các chương trình kết nối doanh nghiệp, kết nối nhà tuyển dụng, chuyển tiếp hồ sơ trực tiếp cho nhà tuyển dụng. Tư vấn và hỗ trợ nguồn nhân lực chất lượng cao cho doanh nghiệp trong và ngoài nước. Các chương trình kết nối đều được VnPro tài trợ và thực hiện suốt quá trình hoạt động.

1.2 Lý do chọn đề tài

Khi kỷ nguyên mới của khả năng lập trình mạng, tự động hóa và SDN đã đến, các kỹ sư mạng phải học cách viết code đã trở nên cần thiết. Mã hóa cho phép phát triển các

công cụ phần mềm. Kết quả là, các kỹ sư mạng có thể tự động hóa các nhiệm vụ mạng phức tạp.

Giao tiếp giữa các công cụ phần mềm và nền tảng mạng xảy ra do các nhà sản xuất đã kích hoạt các giao diện lập trình ứng dụng hoặc API tốt hơn. Trong số các nền tảng mạng, API REST là một trong những nền tảng phổ biến nhất. Vì lý do này, điều quan trọng là phải biết cơ sở lý thuyết và thực tiễn của API này.

VnPro là một trung tâm chuyên đào tạo kỹ sư mạng luôn thích ứng tốt với sự thay đổi, tiến hóa của công nghệ trong lĩnh vực network nên khi thực tập ở đây được hỗ trợ rất nhiều tài nguyên để học tập và nghiên cứu và APIC-EM là một trong những network controller kinh điển cho việc sử dụng và tương tác API cho người mới sử dụng cũng như là một trong những nền tảng quan trọng trong việc tương tác và làm việc với API.

1.3 Mục tiêu đề tài

Khai thác API documentation của Network Controller APIC-EM được VnPro dựng để tạo Ticket xác thực và GET các thông tin về vị trí, số seri, thông tin cấu hình, interfaces,của các con thiết bị có trong APIC-EM. Đây là một trong những kiến thức quan trọng để làm quen và thực hành tương tác với API khi thực tập tại trung tâm được giới thiệu và API là xu hướng của công nghệ mạng trong tương lai.

CHƯƠNG 2 – TỔNG QUAN ĐỀ TÀI

2.1 Tìm hiểu về API

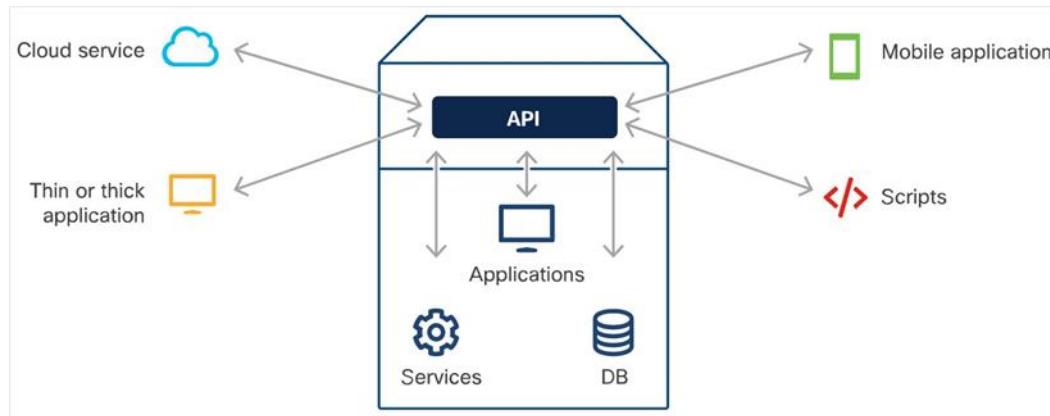
2.1.1 Khái niệm API

API viết tắt của Application Programming Interface.

API là một phần mềm cho phép truy cập dữ liệu hoặc dịch vụ giữa các ứng dụng khác nhau. Đó là một tập hợp các quy tắc mô tả cách thức một ứng dụng tương tác với một ứng dụng khác và các hướng dẫn để cho phép tương tác này xảy ra.

Người sử dụng gửi yêu cầu truy cập API request đến máy chủ để yêu cầu thông tin và nhận được trả lời API response từ server cùng với thông tin được yêu cầu.

API xác định loại dữ liệu, dịch vụ và chức năng mà ứng dụng hiển thị cho các bên thứ ba.



Hình 2.1.1 Minh họa API 1

Các trường hợp sử dụng API:

- Các tác vụ tự động hóa - Xây dựng một tập lệnh thực hiện các tác vụ thủ công một cách tự động và theo chương trình.
- Tích hợp dữ liệu - Một ứng dụng có thể tiêu thụ hoặc phản ứng với dữ liệu được cung cấp bởi một ứng dụng khác.
- Thực hiện chức năng - Một ứng dụng có thể tích hợp chức năng của ứng dụng khác vào sản phẩm của nó.

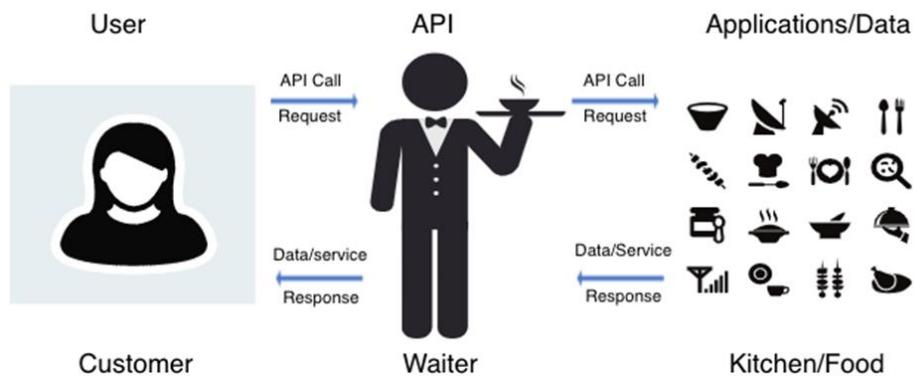
2.1.2 Ví dụ về API

Khi sử dụng một ứng dụng trên điện thoại di động, ứng dụng đó sẽ kết nối với Internet và gửi dữ liệu đến máy chủ. Sau đó, máy chủ lấy dữ liệu đó, diễn giải nó, thực hiện các hành động cần thiết và gửi lại cho điện thoại.

Sau đó, ứng dụng sẽ diễn giải dữ liệu đó và trình bày cho bạn thông tin bạn muốn theo cách có thể đọc được tóm tắt cả điều này xảy ra thông qua API.

Để giải thích điều này rõ hơn, chúng ta hãy lấy một ví dụ quen thuộc:

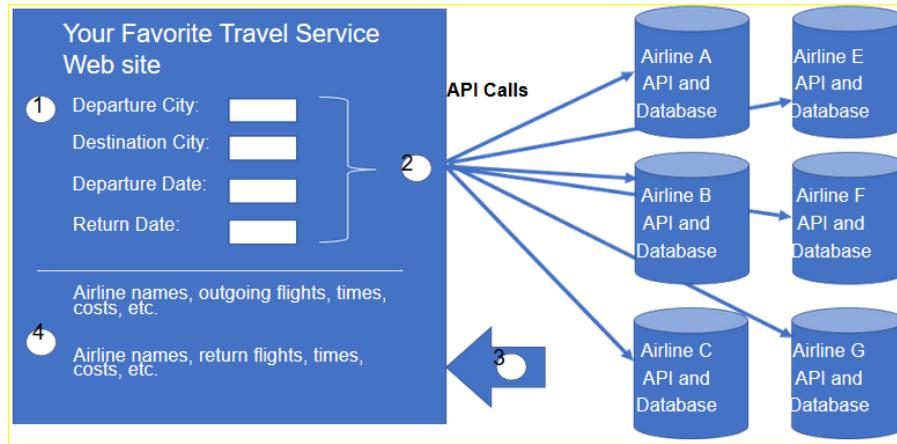
- API tương tự như người phục vụ trong nhà hàng: Hãy tưởng tượng ta đang ngồi trên bàn trong một nhà hàng với thực đơn có nhiều lựa chọn để gọi món. Nhà bếp là một phần của “hệ thống” sẽ chuẩn bị món ăn của ta. Điều còn thiếu là liên kết quan trọng để thông báo đơn đặt hàng của ta đến nhà bếp và giao thức ăn trở lại bàn của ta. Và người thực hiện việc liên kết giữa ta và nhà bếp đó chính là người phục vụ hay API. Người phục vụ là người đưa tin - hay API - nhận yêu cầu hoặc đơn đặt hàng của ta và cho nhà bếp - hệ thống - phải làm gì. Sau đó, người phục vụ gửi phản hồi lại cho ta; trong trường hợp này, nó là thức ăn.



Hình 2.1.2.1 Ví dụ API 1

- Một trang web du lịch có thể truy cập cùng thông tin về các chuyến bay, không chỉ từ một hãng hàng không cụ thể mà nhiều loại hãng hàng không. Trong trường hợp này, người dùng nhập thông tin đặt phòng tương tự. Trang web dịch vụ du lịch tương tác với các cơ sở dữ liệu hàng không khác nhau bằng các API được cung cấp bởi mỗi hãng hàng không. Dịch vụ du lịch sử dụng từng API của hãng hàng không để yêu cầu thông tin từ hãng hàng không cụ thể đó và sau đó nó sẽ

hiển thị thông tin từ tất cả các hãng hàng không trên trang web của hãng. API hoạt động như một loại tin nhắn giữa ứng dụng yêu cầu và ứng dụng trên máy chủ cung cấp dữ liệu hoặc dịch vụ. Thông báo từ ứng dụng yêu cầu đến máy chủ nơi lưu trữ dữ liệu được gọi là lệnh gọi API.



Hình 2.1.2.2 Ví dụ đặt vé máy bay ứng dụng API 1

2.1.3 Phân loại API

a. Phân loại API dựa trên quyền truy cập

- API mở (Open APIs/Public APIs): Các API này có sẵn công khai và có thể được sử dụng không bị hạn chế. Vì các API này công khai nên nhiều nhà cung cấp API yêu cầu người sử dụng nhận khóa hoặc mã thông báo trước khi sử dụng nhằm giúp kiểm soát số lượng yêu cầu API mà họ nhận được và xử lý.
- API nội bộ (Internal APIs): Đây là các API được sử dụng bởi một tổ chức hoặc công ty để truy cập dữ liệu và dịch vụ nội bộ.
- API cho đối tác (Partner APIs): Đây là các API được sử dụng giữa một tổ chức hoặc công ty với đối tác để tạo thuận lợi cho việc kinh doanh giữa họ. Đối tác kinh doanh phải có giấy phép hoặc được sự cho phép để sử dụng API này.

b. Phân loại theo kiểu truy cập và sử dụng

Ta có ba kiểu truy cập và sử dụng API

- Kiểu riêng (private): API dạng này chỉ được sử dụng nội bộ. Kiểu truy cập này sẽ cho phép doanh nghiệp kiểm soát toàn bộ API của họ.

- Kiểu đối tác (partner): Kiểu API này được dùng để chia sẻ với các đối tác kinh doanh. Kiểu này có thể tạo ra các dòng giá trị lợi nhuận mà không làm giảm chất lượng.
- Kiểu công cộng (public): Kiểu API công cộng thì sẵn sàng cho mọi người sử dụng. Loại API này cho phép các bên thứ ba phát triển các ứng dụng giao tiếp với API và có thể là một nguồn để tăng tính sáng tạo, đổi mới.

Bất chấp các API được truy cập như thế nào, các API được thiết kế để tương tác thông qua một hạ tầng mạng. Và bởi vì phần lớn các giao tiếp mạng ngày nay dùng Internet, phần lớn các cuộc gọi API được thiết kế dựa trên chuẩn web.

Do đặc điểm độc đáo của HTTP trên web, phần lớn các nhà phát triển ứng dụng đã dùng HTTP như là giao thức bên dưới cho các API của họ. Lợi ích lớn nhất của việc sử dụng HTTP là nó giảm thiểu việc học hỏi thêm về HTTP do HTTP quá phổ biến. HTTP có vài đặc điểm rất hữu ích khi chúng ta xây dựng API.

c. Phân loại API dựa trên ba loại công việc mà API thực hiện

- API dịch vụ (Service API): Trong loại API này, một ứng dụng có thể gọi một ứng dụng khác để giải quyết một vấn đề cụ thể. Thường thì các hệ thống này có thể tồn tại độc lập với nhau. Ví dụ, trong một hệ thống thanh toán, một ứng dụng có thể gọi API để chấp nhận một thanh toán dùng thẻ tín dụng. Một ví dụ khác là với hệ thống quản lý người dùng, một ứng dụng có thể gọi một API để kiểm tra và xác thực người dùng. Hình vẽ bên dưới minh họa cách App1 sử dụng API trong các chức năng thanh toán, quản lý người dùng, quản lý các mạng xã hội.



Hình 2.1.3.1 Sử dụng API cung cấp dịch vụ 1

- API cung cấp thông tin (Information API): Cho phép một ứng dụng hỏi một ứng dụng khác về một thông tin nào đó. Thông tin trong ngữ cảnh này có thể muôn ám chỉ đến các dữ liệu được thu thập trong một thời gian dài, các dữ liệu từ xa telemetry hay danh sách của một số thiết bị đang kết nối vào hệ thống mạng.



Hình 2.1.3.2 Sử dụng API cung cấp thông tin 1

- API phần cứng (Hardware API): Một số nhà phát triển ứng dụng có thể dùng các API phần cứng này để truy cập đến một số tính năng của thiết bị phần cứng. Thông thường các API bao gồm một vài thiết bị phần cứng hay các cảm biến sensor. Một ứng dụng có thể gọi kiểu API này để truy xuất vị trí GPS hay các dữ liệu thời gian thực khác như nhiệt độ hay độ ẩm.



Hình 2.1.3.3 Sử dụng API ứng dụng phần cứng 1

d. Phân loại API theo kiến trúc: Có một số tiêu chuẩn, giao thức và phong cách kiến trúc cụ thể giúp người tiêu dùng API tìm hiểu và hiểu API dễ dàng hơn. Ba kiểu kiến trúc API phổ biến nhất là: RPC (Remote Procedure Call – Gọi thủ tục từ xa), SOAP (Simple Object Access Protocol – Giao thức truy cập đối tượng đơn giản), REST (Representational State Transfer)

- RPC (Remote Procedure Call – Gọi thủ tục từ xa): là một mô hình phản hồi yêu cầu cho phép một ứng dụng thực hiện một cuộc gọi thủ tục tới một ứng dụng khác. Khi RPC được gọi đến một máy khách, phương thức sẽ được thực thi và kết quả được trả về. RPC là một kiểu API có thể được áp dụng cho các giao thức truyền tải khác nhau như: XML-RPC, JSON-RPC, NFS (Hệ thống tệp mạng), SOAP
- SOAP (Simple Object Access Protocol – Giao thức truy cập đối tượng đơn giản): Giao thức truy cập đối tượng đơn giản (SOAP) là một giao thức nhắn tin dựa trên XML. Nó được sử dụng để giao tiếp giữa các ứng dụng trên các nền tảng khác nhau hoặc được xây dựng bằng các ngôn ngữ lập trình khác nhau.
- REST (Representational State Transfer): à một phong cách kiến trúc do Roy Thomas Fielding tạo ra. Điều kiện của một mô hình REST là:

- client-server: máy khách và máy chủ phải độc lập với nhau cho phép máy khách được xây dựng cho nhiều nền tảng sẽ đơn giản hóa các thành phần phía máy chủ.
- Không trạng thái (Stateless): Yêu cầu từ máy khách đến máy chủ phải chứa mô hình máy khách-máy chủ REST và tất cả thông tin mà máy chủ cần để thực hiện yêu cầu. Máy chủ không thể chứa các trạng thái phiên.
- Mô hình bộ nhớ đệm: Phản hồi từ máy chủ phải cho biết phản hồi có thể lưu vào bộ nhớ cache hay không thể lưu vào bộ nhớ cache. Nếu nó có thể lưu vào bộ nhớ cache, máy khách có thể sử dụng dữ liệu từ phản hồi cho các yêu cầu sau này.
- Giao diện thống nhất: Giao diện giữa máy khách và máy chủ tuân thủ bốn nguyên tắc:
 - Xác định các nguồn lực
 - Thao tác tài nguyên thông qua các đại diện
 - Thông điệp tự mô tả
 - Hypermedia làm động cơ của trạng thái ứng dụng
- Hệ thống phân lớp: Hệ thống phân lớp bao gồm các lớp phân cấp khác nhau, trong đó mỗi lớp chỉ cung cấp các dịch vụ cho lớp phía trên nó. Kết quả là, nó tiêu thụ các dịch vụ từ lớp bên dưới.
- Mã theo yêu cầu: Thông tin do dịch vụ REST trả về có thể bao gồm mã thực thi (ví dụ: javascript) hoặc các liên kết nhằm mục đích mở rộng chức năng của máy khách một cách hữu ích. Ràng buộc này là tùy chọn vì việc thực thi mã của bên thứ ba dẫn đến các rủi ro bảo mật tiềm ẩn.

2.1.4 Ưu điểm và nhược điểm của API

a. Ưu điểm

- Tính tự động hóa rất cao: API có thể thay thế chúng ta quản lý công việc cực kì hiệu quả. API giúp các cơ quan có thể cập nhật, xử lý hoàn thiện công việc nhanh và chất lượng hơn.
- API có thể truy cập vào các thành phần ứng dụng giúp việc cung cấp dịch vụ và thông tin linh hoạt hơn nhiều.
- API có chức năng thay đổi cũng như dự đoán thay đổi theo thời gian cho nên dữ liệu được truyền tốt hơn, thông tin được chọn lọc kỹ hơn, dịch vụ tốt hơn.
- Người dùng có thể tinh chỉnh API cho phù hợp nhu cầu sử dụng.
- API cho phép mọi thông tin được tạo ở dạng chính chủ luôn có sẵn cho mọi người xem được.
- Ngoài dữ liệu mới có sẵn được chia sẻ rộng rãi, người dùng còn có thể hiệu chỉnh web API để cung cấp dịch vụ, thông tin cá nhân hóa.
- API có thể thực hiện các thao tác CRUD (Create Read Update Delete) điển hình thông qua các động từ HTTP GET, PUT, POST và DELETE.
- Các ngôn ngữ mã hóa đơn giản hóa như Python đã giúp các kỹ sư không chuyên về phần mềm có thể xây dựng ứng dụng và sử dụng API.

b. Nhược điểm

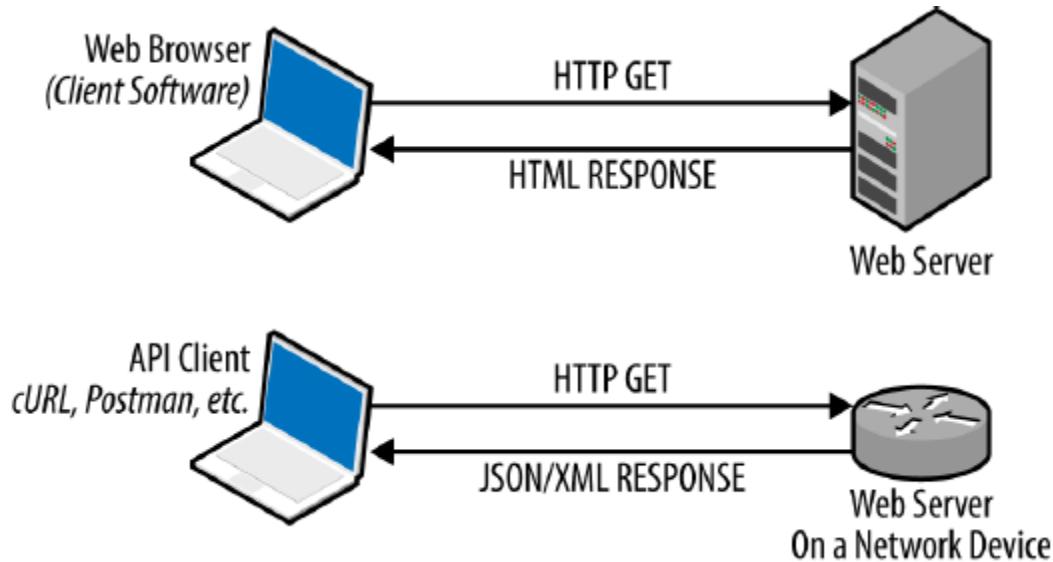
- Chi phí: Cung cấp một API tốn kém về thời gian phát triển, bảo trì liên tục, tài liệu cho API trên trang web của bạn và hỗ trợ cho người dùng API.
- Bảo mật: bởi tính chất kết hợp API, ta vô tình mở rộng phạm vi tấn công cho các hacker.
- Phát triển các API là một thủ tục dài đòi hỏi một kích thước đã định. Kỹ năng lập trình là bắt buộc để xây dựng các API. Nó có thể bị lỗi khi kiểm tra API. Chi phí bảo trì là đáng kể.

2.2 Tìm hiểu về REST API

REST API ngày càng trở nên phổ biến và được sử dụng nhiều trong ngành công nghiệp mạng, mặc dù kiểu kiến trúc REST đã có từ những năm 2000. Hầu hết các kiểu API hiện có trong hạ tầng mạng là kiểu REST API. Điều này có nghĩa là nếu bạn nói đến một thiết bị hay một SDN controller có hỗ trợ API, thì loại API đó là một giao tiếp theo mô hình client-server. Phía client có thể là một phần mềm ứng dụng chẵng hạn như một chương trình Python hoặc một giao diện web. Phía máy chủ server có thể là một thiết bị mạng hoặc là một network controller.

Một API sử dụng kiến trúc REST thường được gọi là RESTful API. Kiến trúc REST thường được mô tả chung như là một kiến trúc theo mô hình client-server, phi trạng thái (stateless). Các Restful API thường dùng giao thức HTTP và các phương thức của HTTP để thu thập và thao tác các dữ liệu.

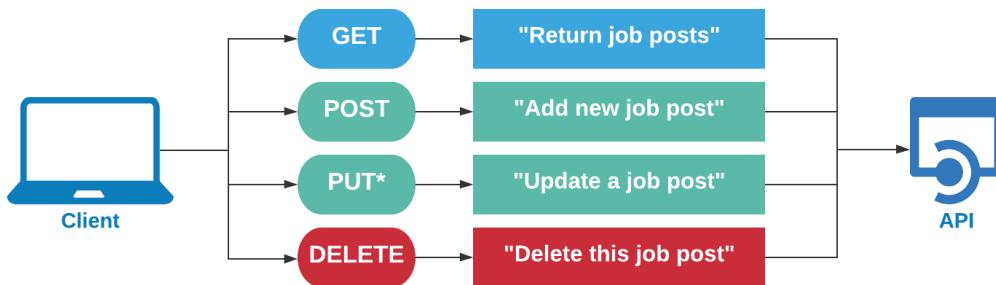
Trong hình minh họa bên dưới, sự khác nhau nằm ở định dạng dữ liệu được gửi vào và ra của máy chủ web. Khi chúng ta duyệt Internet, chúng ta sẽ nhận được các dữ liệu HTML, sau đó các trình duyệt sẽ hiển thị các nội dung này. Còn khi chúng ta thực thi một hàm gọi HTTP GET đến một máy chủ đang cung cấp chức năng của nó thông qua RESTful API, chúng ta sẽ nhận lại được kết quả trả về ở dạng JSON hay XML. Lúc này, các máy client phải hiểu và diễn dịch được các định dạng dữ liệu JSON hay XML.



Hình 2.2.1 Cách hoạt động của REST API 1

Vì HTTP được dùng như một giao thức truyền vận, khi chúng ta gọi API, chúng ta cũng thực hiện các tác vụ sử dụng URL cũng giống như khi chúng ta duyệt World Wide Web. Khi chúng ta truy cập đến một website, phương thức HTTP GET được thực hiện. Nếu chúng ta điền vào một biểu mẫu trên web và nhấn nút Submit, một phương thức HTTP POST đang được thực thi. RESTful API cũng hoạt động cơ bản giống như vậy.

Chúng ta ít nhiều đều đã nghe nói đến giao thức HTTP hoặc đã biết về cách thức HTTP hoạt động. Hoạt động của HTTP thường theo cấu trúc được định nghĩa chặt chẽ, nhất quán. Chính vì vậy HTTP cung cấp một phương thức nhất quán để gọi API từ các thiết bị khác nhau của các nhà cung cấp khác nhau.



Hình 2.2.2 Các phương thức client sử dụng khi gọi API 1

Các chức năng của HTTP tương tự như các chức năng mà phần lớn các ứng dụng và các cơ sở dữ liệu dùng để lưu trữ và thay đổi dữ liệu. Các chức năng này được gọi là CRUD, viết tắt của CREATE, READ, UPDATE, DELETE.

Kiến trúc REST sử dụng các chức năng có sẵn của HTTP để tương tác với dữ liệu.

Để gửi API yêu cầu đến máy chủ (REST API Request) bằng giao thức HTTP ta cần có bốn thành phần: Bộ danh tài nguyên (Uniform Resource Identifier - URI), phương thức HTTP, Header – Tiêu đề, Body – Nội dung

- a. Bộ danh tài nguyên (Uniform Resource Identifier - URI) hay còn được gọi là URL xác định tài nguyên nào mà khách hàng muốn thao tác. Các thành phần của URI là Lược đồ: chỉ định giao thức HTTP nào nên được sử dụng, http hoặc https. Quyền hạn: bao gồm hai phần, đó là máy chủ và cổng. Đường dẫn: đại

diện cho vị trí của tài nguyên, dữ liệu hoặc đối tượng, được thao tác trên máy chủ. Truy vấn: cung cấp chi tiết bổ sung cho phạm vi, lọc hoặc để làm rõ một yêu cầu.

- b. Phương thức HTTP: Các API REST sử dụng các phương thức HTTP tiêu chuẩn để giao tiếp với các dịch vụ web mà hành động đang được yêu cầu đối với tài nguyên đã cho.

Phương thức HTTP	Tác dụng	Mô tả
POST	Tạo	Tạo một đối tượng mới hay một tài nguyên mới
GET	Đọc	Liệt kê chi tiết các nguồn tài nguyên có trong hệ thống
PUT	Cập nhật	Thay thế hoặc cập nhật tài nguyên có trong hệ thống
PATCH	Cập nhật một phần	Cập nhật một số chi tiết từ một tài nguyên có sẵn trong hệ thống
DELETE	Xóa	Xóa một tài nguyên khỏi hệ thống.

Bảng 2.2.1 Các phương thức của REST API 1

- c. Tiêu đề: Tiêu đề HTTP được định dạng dưới dạng các cặp tên-giá trị được phân tách bằng dấu hai chấm (:), [tên]: [giá trị]. Có 2 loại Headers:
- Tiêu đề requests: Cung cấp thêm thông tin không liên quan đến nội dung tin nhắn.

Khóa	Giá trị ví dụ	Mô tả
Authorization	Basic dmFncmFudDp2YWdyYW	Cung cấp thông tin chứng chỉ xác thực request.

- Tiêu đề thực thể: Thông tin bổ sung mô tả nội dung của phần nội dung thư.

Khóa	Giá trị ví dụ	Mô tả
Content-Type	application/json	Chỉ định định dạng của dữ liệu của nội dung trả về.

Bảng 2.2.2 Dữ liệu của Headers 1

d. Nội dung: Nội dung của yêu cầu API REST chứa dữ liệu liên quan đến tài nguyên mà máy khách muốn thao tác. Các yêu cầu API REST sử dụng phương thức HTTP POST, PUT và PATCH thường bao gồm một phần thân. Phần thân là tùy chọn tùy thuộc vào phương thức HTTP. Nếu dữ liệu được cung cấp trong phần nội dung, thì kiểu dữ liệu phải được chỉ định trong tiêu đề bằng cách sử dụng khóa Loại-Nội dung (Content-Type).

Sau khi gửi REST API request ta sẽ nhận lại các phản hồi API REST là các phản hồi HTTP thông báo kết quả của một yêu cầu HTTP của mình.

Phản hồi API REST được tạo thành từ ba thành phần chính: Trạng thái HTTP, Tiêu đề, Nội dung:

- a. Trạng thái HTTP: Mã trạng thái HTTP giúp khách hàng xác định lý do gây ra lỗi và đôi khi có thể cung cấp các đề xuất để khắc phục sự cố. Mã trạng thái HTTP bao gồm ba chữ số, trong đó chữ số đầu tiên là danh mục phản hồi và hai chữ số khác được gán theo thứ tự số. Có năm loại mã trạng thái HTTP khác nhau: 1xx - Thông tin - vì mục đích thông tin, câu trả lời không chứa nội dung; 2xx - Thành công - máy chủ đã nhận và chấp nhận yêu cầu; 3xx - Chuyển hướng - khách hàng phải thực hiện thêm một hành động để yêu cầu được hoàn thành; 4xx - Lỗi máy khách - yêu cầu có lỗi như cú pháp sai hoặc đầu vào không hợp lệ; 5xx - Lỗi Máy chủ - không thể thực hiện các yêu cầu hợp lệ. Các trạng thái HTTP phổ biến:

Mã trạng thái HTTP	Tình trạng thông điệp	Mô tả
200	Ok	Yêu cầu đã thành công và trả về nội dung được yêu cầu
201	Created (Tạo)	Yêu cầu đã được thực hiện và tài nguyên được yêu cầu đã được tạo
202	Accepted (Chấp nhận)	Yêu cầu đã được chấp nhận để xử lý và đang trong quá trình xử lý
400	Bad Request	Yêu cầu sẽ không được xử lý do lỗi với yêu cầu.
401	Unauthorized	Yêu cầu không có thông tin xác thực hợp lệ để thực hiện yêu cầu
403	Forbidden	Yêu cầu đã được gửi nhưng đã bị máy chủ từ chối
404	Not Found	Không thể thực hiện yêu cầu vì đường dẫn tài nguyên của yêu cầu không được tìm thấy trên máy chủ
500	Internal Server Error	Không thể thực hiện yêu cầu do lỗi máy chủ
503	Service Unavailable	Không thể thực hiện yêu cầu vì hiện tại máy chủ không thể xử lý yêu cầu

Bảng 2.2.3 Một số mã trạng thái phổ biến 1

b. Tiêu đề: Tiêu đề trong phản hồi là cung cấp thông tin bổ sung giữa máy chủ và máy khách ở định dạng cặp tên-giá trị được phân tách bằng dấu hai chấm (:), [name]: [value]. Có hai loại tiêu đề: tiêu đề phản hồi (response headers): Nó chứa thông tin bổ sung không liên quan đến nội dung của tin nhắn. Các tiêu đề phản hồi điển hình cho một yêu cầu API REST bao gồm: Key, Value

Key	Ví dụ Value	Mô tả
-----	-------------	-------

Set-Cookie	JSESSIONID=30A9DN810FQ428P; Path=/	Được sử dụng để gửi cookie từ máy chủ
Cache-Control	Cache-Control: max-age=3600, public	Chỉ định các lệnh phải tuân theo tất cả các cơ chế lưu vào bộ nhớ đệm

Bảng 2.2.4 Dữ liệu Headers Request 1

và tiêu đề thực thể (entity headers): Chúng là thông tin bổ sung mô tả nội dung của nội dung thông điệp. Một tiêu đề thực thể chung chỉ định loại dữ liệu được trả về:

Key	Example Value	Description
Content-Type	application/json	Chỉ định định dạng của dữ liệu trong nội dung.

Bảng 2.2.5 Data format của dữ liệu trả về 1

2.3 Tìm hiểu về SDN (Software-Defined Network)

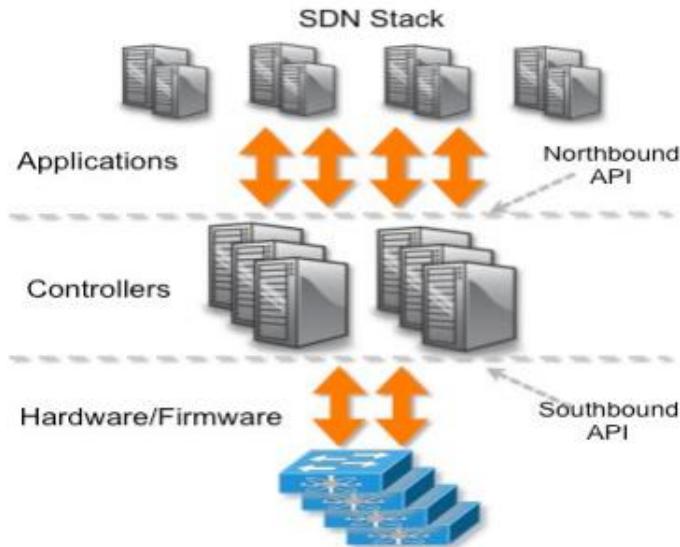
2.3.1 Định nghĩa SDN

Mạng do phần mềm xác định (SDN) là một kiến trúc mới năng động, dễ quản lý, tiết kiệm chi phí và có thể thích ứng, lý tưởng cho tính chất tự động hóa, băng thông cao của các ứng dụng ngày nay. Kiến trúc này tách rời các chức năng điều khiển mạng và chuyển tiếp cho phép điều khiển mạng trở nên có thể lập trình trực tiếp và cơ sở hạ tầng cơ bản được trùu tượng hóa cho các ứng dụng và dịch vụ mạng cải tiến hơn so với việc quản lý truyền thống. Giao thức OpenFlow là yếu tố nền tảng để xây dựng các giải pháp SDN. SD-WAN áp dụng công nghệ SDN cho mạng diện rộng (WAN).

Công nghệ SDN hứa hẹn sẽ phá vỡ ranh giới vật lý trên thiết bị mạng thông qua Giao diện lập trình ứng dụng (API). Vì vậy, trong SDN, toàn bộ quá trình kết nối mạng là do phần mềm điều khiển và API là nhân tố quan trọng của mô hình kiến trúc mới này.

API là nền tảng quan trọng cho một thế hệ mới các phần mềm trong lĩnh vực mạng, điện toán đám mây (cloud), lĩnh vực di động mobile và Internet của vạn vật IoT. API kết nối các phần mềm lại với nhau, cho phép hai phần mềm giao tiếp, tương tác với nhau. Các

nhà phát triển phần mềm cũng sử API để cho phép các phần mềm giao tiếp với hạ tầng mạng. Chúng ta có thể dùng API để cấu hình và giám sát các thành phần cụ thể của một hạ tầng mạng.



Hình 2.3.1.1 Thành phần hạ tầng ứng dụng công nghệ SDN 1

2.3.2 Phân loại SDN

Thường có hai bộ điều khiển SDN

Bộ điều khiển SDN cho ảo hóa chức năng mạng (NFV) của một trung tâm dữ liệu là một công nghệ ảo hóa CNTT để ảo hóa toàn bộ các lớp chức năng của nút mạng thành các khối xây dựng có thể kết nối hoặc chuỗi với nhau để tạo và cung cấp các dịch vụ truyền thông.

NFV dựa trên các kỹ thuật ảo hóa máy chủ truyền thống như các kỹ thuật được sử dụng trong CNTT doanh nghiệp. Một chức năng mạng ảo hóa, hay còn gọi là VNF, được triển khai trong một hoặc nhiều máy ảo hoặc bộ chia sẻ chạy phần mềm và quy trình khác nhau, bên cạnh các máy chủ, thiết bị chuyển mạch và thiết bị lưu trữ khối lượng lớn đã được thương mại hóa (COTS), hoặc thậm chí là cơ sở hạ tầng điện toán đám mây, thay vì có các thiết bị phần cứng tùy chỉnh cho từng chức năng mạng do đó tránh bị khóa nhà cung cấp.

Ví dụ, một bộ điều khiển biên phiến ảo có thể được triển khai để bảo vệ mạng mà không tốn kém chi phí và sự phức tạp điển hình của việc lấy và cài đặt các đơn vị bảo vệ mạng vật lý. Một số ví dụ khác về NFV bao gồm bộ cân bằng tải được ảo hóa, tường lửa, thiết bị phát hiện xâm nhập và bộ tăng tốc WAN.

Việc tách phần mềm chức năng mạng khỏi nền tảng phần cứng tùy chỉnh tạo ra một kiến trúc mạng linh hoạt cho phép quản lý mạng linh hoạt, triển khai dịch vụ mới nhanh chóng với giảm đáng kể CAPEX và OPEX.

Bộ điều khiển SDN để quản lý các công tắc lập trình của mạng.

2.3.3 API ứng dụng vào SDN

API được sử dụng phổ biến nhất trong kiến trúc SDN có 2 loại là northbound API và southbound API dưới góc nhìn của lĩnh vực tự động trong hạ tầng mạng network automation.

Northbound API là một giao tiếp cho phép một thành phần của hạ tầng mạng giao tiếp với các thành phần ở lớp cao hơn. Một cách tương ứng, một giao tiếp southbound cho phép một thành phần mạng giao tiếp với một thành phần ở lớp thấp hơn.

Từ controller, nếu một bạn quản trị đang thay đổi cấu hình của một switch, các thay đổi này sẽ được đẩy từ controller xuống switch. Dạng API này gọi là southbound API do chiều gọi API là từ controller đi xuống thiết bị. Trong hạ tầng mạng, chúng ta có nhiều loại thiết bị như routers, switch, wireless access point AP.

API có thể được dùng để thay đổi nhiều chức năng của thiết bị, chứ không chỉ là thay đổi các chức năng dữ liệu data plane. (Chức năng luân chuyển dữ liệu giữa các cổng của thiết bị và tất cả các hoạt động khác liên quan đến định dạng dữ liệu, nén, mã hóa, kiểm soát lỗi...)

Northbound API tạo thành lớp ứng dụng cho bộ điều khiển SDN và cung cấp hỗ trợ chính cho các ứng dụng bên thứ 3 như các công cụ tương tác được với REST API như Postman, Python, Swagger UI,... kết nối đến các Network Controller có thể kể đến như APIC-EM, Cisco Meraki, DNA Center, SDWAN-Controller,....

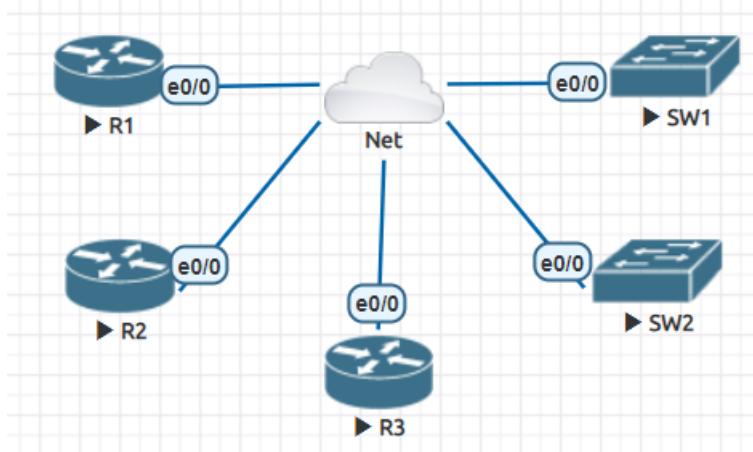
Mỗi một network controller sẽ cung cấp cho ta một API document để người dùng có thể dựa vào đó và gọi REST API đến network controller.

Ví dụ: Phần mềm Cisco DNA Center có một giao diện đồ họa GUI cho phép chúng ta quản lý chính nó. Khi một nhân viên quản trị mạng login vào DNA để quản lý hạ tầng mạng, các thông tin sẽ được gửi từ phần mềm quản lý đến controller thông qua northbound API. Chúng ta còn gọi kiểu API này như là REST API. Các kinh nghiệm thực tế cho thấy các lưu lượng mạng loại này nên được mã hóa bằng giao thức TLS giữa phần mềm và controller. Phần lớn các API đều có khả năng dùng các thuật toán mã hóa để bảo vệ dữ liệu đang luân chuyển.

CHƯƠNG 3 – THỰC NGHIỆM

3.1 Thiết kế sơ đồ mạng cấu hình cho thiết bị

- Đăng nhập vào lab EVE đăng nhập với username và password do VnPro cung cấp do VnPro cung cấp để tiến hành thiết kế sơ đồ mạng gồm 3 Router và 2 Switch.



Hình 3.1.1 Sơ đồ thiết bị mạng 1

- Cấu hình SSH cho các thiết bị. Cấu hình cho các thiết bị với domain-name là vnpro.vn username là vnpro, password là vnpro@123, enable password là vnpro@321.

3.2 Thêm thiết bị vào network controller APIC-EM

3.2.1 Tổng quan về network controller APIC-EM

Application Policy Infrastructure Controller Enterprise Module (APIC-EM) là một phần trung tâm của Kiến trúc mạng kỹ thuật số của Cisco. Là bộ điều khiển mạng, nó cung cấp kết nối mạng được xác định bằng phần mềm cho chi nhánh doanh nghiệp, khuôn viên và mạng WAN. Giao diện người dùng đơn giản của nó cho phép bạn tự động hóa hồ sơ ứng dụng dựa trên chính sách. Với mô-đun này, CNTT có thể đáp ứng nhanh chóng các cơ hội kinh doanh mới.

- ✓ Lợi ích của APIC-EM

Tạo một mạng thông minh, mở, có thể lập trình với các API (Application Programming Interface) mở.

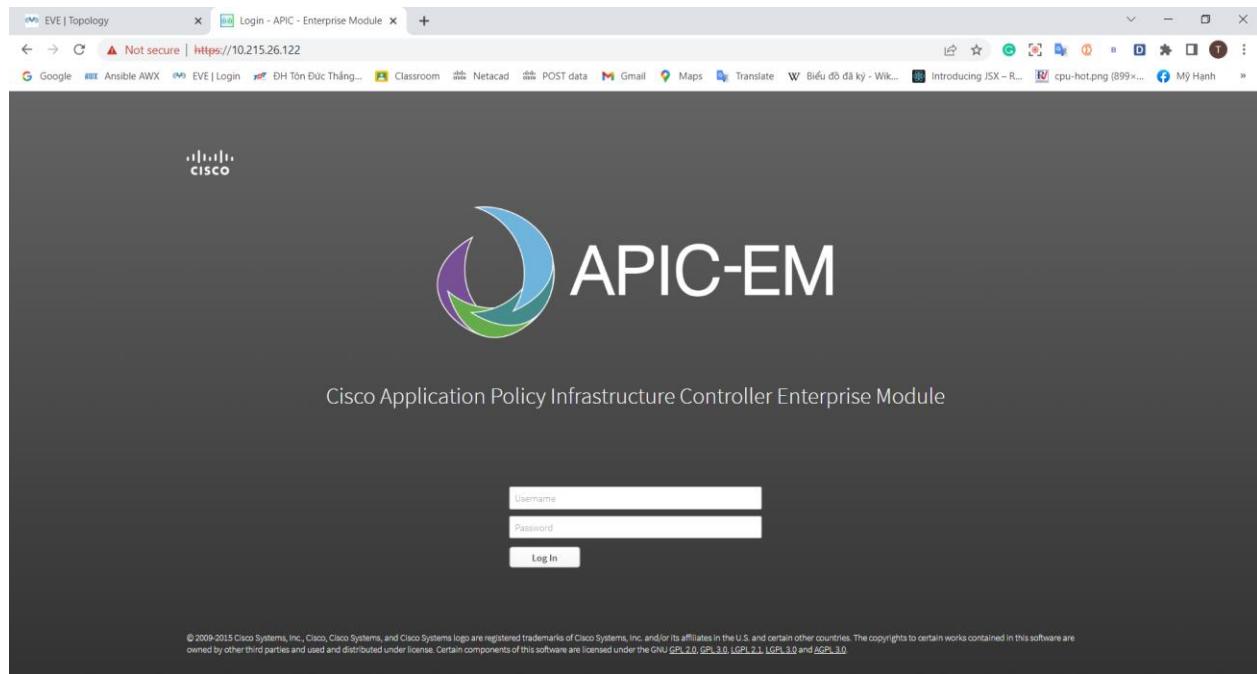
Có thể giúp khách hàng tiết kiệm thời gian, tài nguyên và chi phí thông qua các dịch vụ tự động hóa tiên tiến.

Có thể chuyển đổi chính sách mục đích kinh doanh thành cấu hình mạng động.

Cung cấp một điểm duy nhất cho tự động hóa và điều khiển trên toàn mạng.

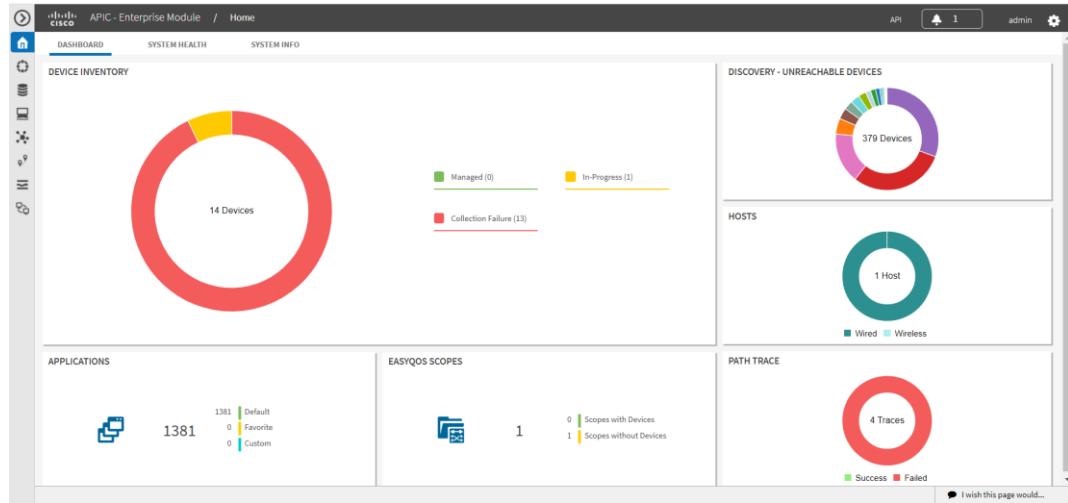
3.2.2 Giao diện APIC-EM

Giao diện đăng nhập



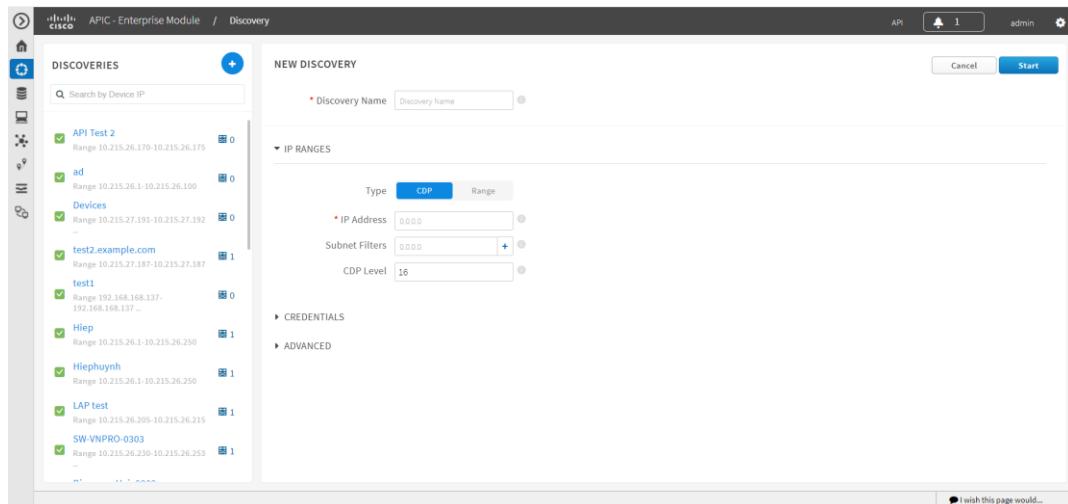
Hình 3.2.2.1 Giao diện đăng nhập 1

Dashboard APIC-EM



Hình 3.2.2.2 Giao diện Dashboard 1

Discoveries



Hình 3.2.2.3 Giao diện Discovery 1

Device Inventory

Device Name	IP Address	Reachability Status	Up Time	Last Updated Time	Poller Time	Last Inventory Collection Status
R_vnpro	10.215.26.200	Unreachable	4 days, 7:33:40.12	22 minutes ago	00:25:00	DEV-UNREACHED ⓘ
R1.example.com	10.215.27.187	Unreachable	2 days, 15:44:13.04	5 minutes ago	00:25:00	DEV-UNREACHED ⓘ
R1.vnpro.org	10.215.26.245	Unreachable	0:21:47.81	22 minutes ago	00:25:00	DEV-UNREACHED ⓘ
R2.vnpro	10.215.26.210	Unreachable	1 day, 1:40:28.66	3 minutes ago	00:25:00	DEV-UNREACHED ⓘ
R2.vnpro.org	10.215.26.225	Unreachable	20:19:32.98	25 minutes ago	00:25:00	In Progress
Ridemo1-vnpro.vnpro.vn	10.215.26.233	Unreachable	11:13:45.69	25 minutes ago	00:25:00	DEV-UNREACHED ⓘ
Ridemo2-demo.vnpro.vn	10.215.26.234	Unreachable	1:36:48.74	a few seconds ago	00:25:00	DEV-UNREACHED ⓘ
Ridemo3-vnpro.vnpro.vn	10.215.26.235	Unreachable	11:31:23.86	24 minutes ago	00:25:00	DEV-UNREACHED ⓘ
Ridemo5-vnpro.vnpro.vn	10.215.26.237	Unreachable	11:22:58.33	24 minutes ago	00:25:00	DEV-UNREACHED ⓘ
RouterHan.vnpro.vn	10.215.26.254	Unreachable	20:24:41.10	22 minutes ago	00:25:00	DEV-UNREACHED ⓘ
RouterLai.vnpro.vn	10.215.27.154	Unreachable	20:23:05.75	11 minutes ago	00:25:00	DEV-UNREACHED ⓘ
RouterQuy.vnpro.vn	10.215.27.151	Unreachable	20:23:05.30	14 minutes ago	00:25:00	DEV-UNREACHED ⓘ

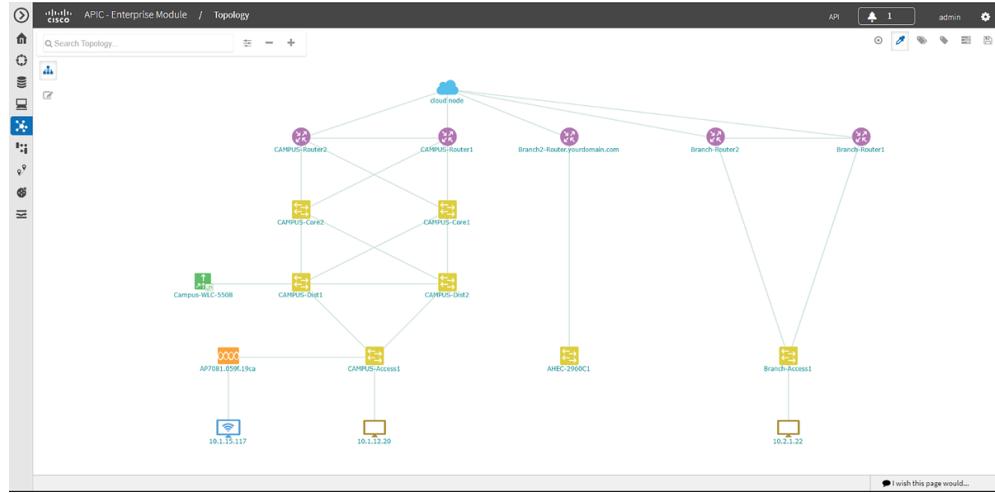
Hình 3.2.2.4 Giao diện Device Inventory 1

Host Inventory

Host MAC Address	Host IP Address	Host Type	Connected Device IP Address	Connected Interface Name	Host Name
undefined	10.215.26.77	WIRED	10.215.26.235	Ethernet0/0	Cisco-ISE-8

Hình 3.2.2.5 Giao diện host inventory 1

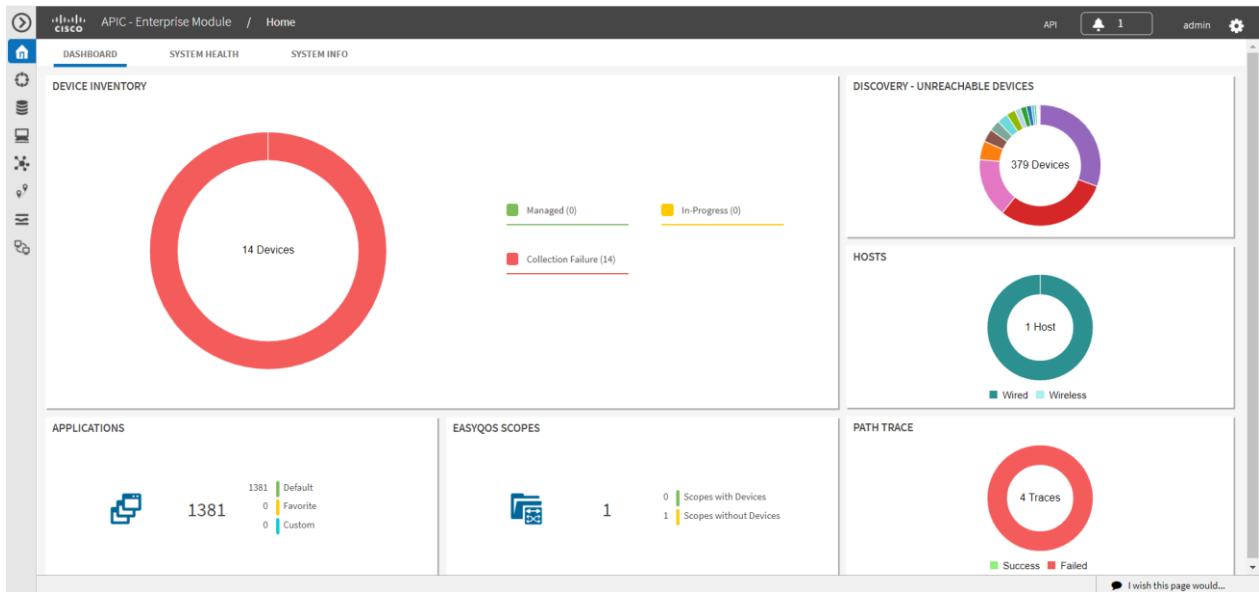
Topology



Hình 3.2.2.6 Giao diện Topology 1

3.2.3 Thêm thiết bị vào APIC-EM

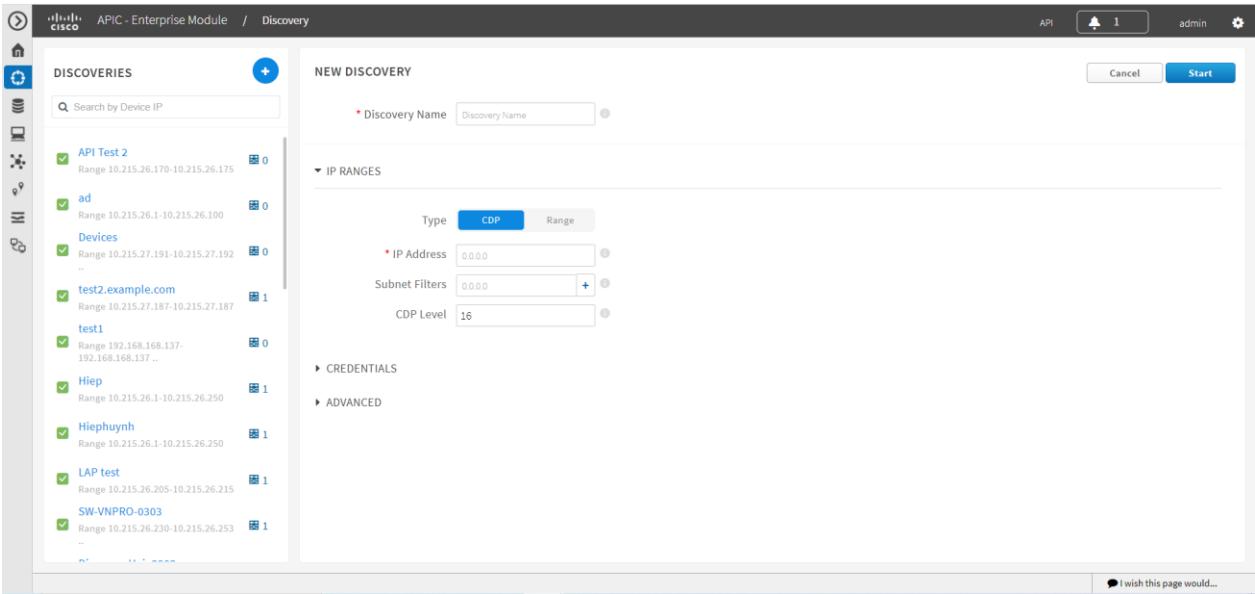
Bước 1: Đăng nhập vào Network Controller



Hình 3.2.3.1 Giao diện Dashboard 1

Bước 2: Thêm thiết bị

Vào phần Discovery



Hình 3.2.3.2 Giao diện thêm thiết bị vào APIC-EM 1

Nhập thông tin Discovery Name, Chọn Range ở phần IP Ranges sau đó nhập dãy IP muốn Controller scan để tìm thiết bị. Khi đó ta lần lượt điền các IP các thiết bị mà ta đã cấu hình SSH vào theo sơ đồ mạng thì em sẽ điền dãy IP từ 10.215.26.181 đến 10.215.26.228

```
R1>
R1>
R1>en
Password:
Password:
R1#sh ip int br
Interface          IP-Address      OK? Method Status      Prot
ocul
Ethernet0/0        10.215.27.160   YES  DHCP    up           up
Ethernet0/1        unassigned     YES  unset   administratively down down
Ethernet0/2        unassigned     YES  unset   administratively down down
Ethernet0/3        unassigned     YES  unset   administratively down down
R1#
```

Hình 3.2.3.3 Kiểm tra IP thiết bị cấu hình SSH 1

NEW DISCOVERY

* Discovery Name i

▼ IP RANGES

Type	CDP	Range
* IP Address	<input type="text" value="10.215.27.160"/> i	
Subnet Filters	<input type="text" value="0.0.0.0"/> + i	
	<input type="text" value="255.255.254.0"/> x	
CDP Level	<input type="text" value="1"/> i	

Hình 3.2.3.4 Điền thông tin để thêm thiết bị 1

Sau đó tiếp tục vào mục Credentials chọn Add Credentials

NEW DISCOVERY

* Discovery Name: DACNTT1_Hai

Type: Range

* IP Ranges: 0.0.0 - 0.0.0
10.215.26.181 to 10.215.26.228

ADD CREDENTIALS

CLI

GLOBAL: admin, vnpro, vnpro

SNMPv2c Read: public

* Username:

* Password:

* Confirm Password:

Enable Password:

Confirm Enable Password:

Save as global settings
Settings will be used for this specific Discovery job only

Save

Hình 3.2.3.5 Điền username và password SSH 1

Ta tiếp tục điền username, password SSH và enable password vào mục CLI sau đó nhấn Save. Khi đó bên mục CLI sẽ xuất hiện username mà ta mới điền.

*** Username**
tdtu

*** Password**
.....

*** Confirm Password**
.....

Enable Password
.....

Confirm Enable Password
.....

Save as global settings
Settings will be used for this specific Discovery job **only**

Save

CLI		SNMPv2c Read	
admin	✓	public	✓
vnpro	✓		
vnpro	✓		
tdtu	✓		

SNMPv2c Write		SNMP v3	
private	✓	admin1	✓
		admin	✓

Hình 3.2.3.6 Nhập thông tin SSH và SNMP 1

Tiếp theo ta sẽ điền vào mục SNMP v2c, ở mục read ta điền tên và read community là public nếu điền vào mục write ta sẽ điền là private. Sau đó nhấn Save để lưu thì bên mục SNMP v2c sẽ xuất hiện mục tương ứng.

SNMP v2c

Read **Write**

* Name/Description
public

* Read Community
xxxx

* Confirm Read Community
xxxx

Save as global settings
Settings will be used for this specific Discovery job **only**

Save

CLI		SNMPv2c Read	
admin	✓	public	✓
vnpro	✓	SNMP v2c Read	✓
vnpro	✓		
tdtu	✓		

SNMPv2c Write		SNMP v3	
private	✓	admin1	✓
		admin	✓

Hình 3.2.3.7 ĐIỀN THÔNG TIN CẤU HÌNH SNMP 1

Sau khi hoàn tất quá trình điền các thông tin thì ta sẽ nhấn Start để bắt đầu cho Controller quét các con thiết bị mình muốn thêm vào

NEW DISCOVERY

* Discovery Name **DACNTT1_Hai**

+ Add Credentials

GLOBAL JOB SPECIFIC

CLI **SNMPv2c Read**

Hình 3.2.3.8 BẤM START ĐỂ BẮT ĐẦU TIẾN TRÌNH 1

Sau khi nhấn Start sẽ xuất hiện cửa sổ với tên Discovery và tác vụ đang thực hiện

CDP LEVEL	PROTOCOL ORDER	RETRY COUNT	TIMEOUT
None	ssh	3	5

IP RANGE: 10.215.26.181-10.215.26.228 IP FILTER LIST: None

► CREDENTIALS

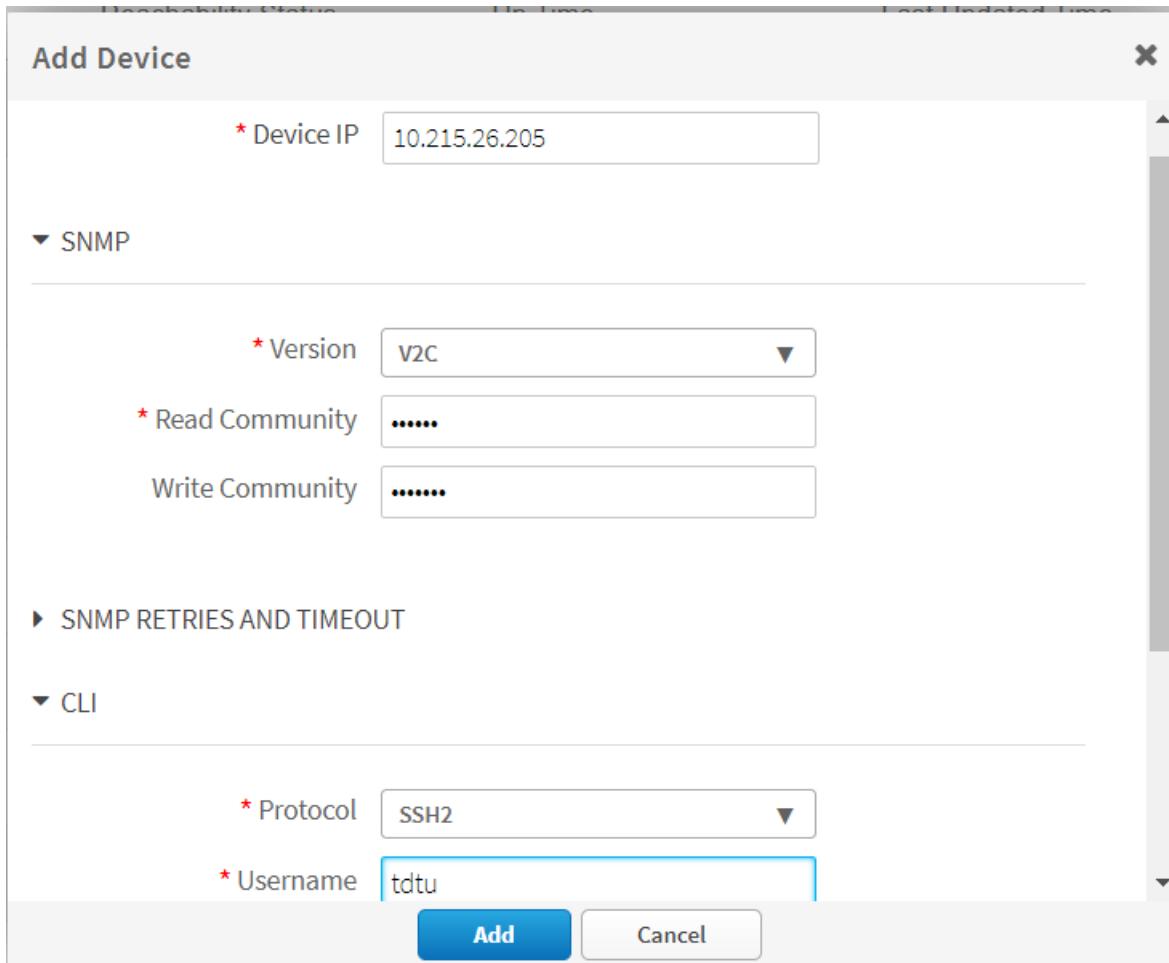
▼ HISTORY

Showing last 0 runs

#	STATUS	DEVICES	START TIME	DURATION
No results to display				

Hình 3.2.3.9 Tiết trình đang thực hiện 1

Sau khi quét xong ta vào Host Inventory để add thiết bị đã được quét vào Host Inventory



Hình 3.2.3.10 Thực hiện thêm thông tin thiết bị vừa quét đc 1

Các thiết bị còn lại cũng thêm tương tự vào.

<input type="checkbox"/>	SW1.tdtu.vn	10.215.26.205	Reachable	3:40:39.31	a few seconds ago	00:25:00	Managed
--------------------------	-------------	---------------	-----------	------------	-------------------	----------	---------

Hình 3.2.3.11 Thiết bị đã được thêm vào 1

3.3 Sử dụng Postman lấy ticket, số lượng, danh sách thiết bị, thông tin cấu hình.

3.3.1 Tổng quan về Postman

Postman là một ứng dụng cho phép kiểm tra các API web. Phần mềm được tạo ra vào năm 2012 bởi Abhinav Asthana, Ankit Sobe và Abhijit Kane ở Bangalore nhằm giải quyết vấn đề chia sẻ các bài kiểm tra API. Ban đầu, nó là một plugin cho Google Chrome,

sau đó là một ứng dụng khách phong phú và cuối cùng là một ứng dụng khách mỏng. Nó hiện được sử dụng bởi hơn 500.000 công ty trên toàn thế giới

Postman API Network cung cấp một dịch vụ cho cả người sử dụng API và người tạo API để dễ dàng khám phá, khám phá và chia sẻ các API.

Postman API Network có hai phần: Postman API network public và Postman API network private. Postman API network public của Postman mở cho toàn bộ cộng đồng API, trong khi Postman API network private chỉ dành cho nhóm của ta hoặc của riêng ta.

Postman có 2 phiên bản: giao diện Web và phiên bản phần mềm. Giao diện chính của phần mềm Postman

Tính năng:

Postman nhóm lại các bài kiểm tra API trong các bộ sưu tập, để đồng nhất hóa các URL và xác thực của chúng. Nó bao gồm:

- Một phiên bản miễn phí với tính năng chia sẻ không gian làm việc cho tối đa ba người dùng.
- Các biến tùy thuộc vào môi trường đã chọn.
- Một hệ thống kiểm soát phiên bản của các thử nghiệm và môi trường.
- Quyền truy cập cho mỗi vai trò (người dùng hoặc người chỉnh sửa).
- Điểm chuẩn.
- Nhập và xuất trong JSON.
- Kiểm tra xuất sang các định dạng máy khách HTTP khác nhau (cURL, PHP, Python, Java, Node.js ...).
- Xác thực bằng Mã thông báo web JSON.
- REST, SOAP, GraphQL và gRPC [6]
- Một ứng dụng người dùng Guest cho phép tải tệp lên để gửi tới các API.
- Một bảng điều khiển gỡ lỗi lưu giữ các yêu cầu và phản hồi trước đó trong bộ nhớ.
- Tập lệnh để tự động hóa các bài kiểm tra.

3.3.2 Sử dụng Postman lấy ticket, số lượng, danh sách thiết bị, thông tin cấu hình.

a. Tạo WorkSpaces

Tạo WorkSpaces: tại mục WorkSpaces ta chọn Create workspace

The screenshot shows the Postman application interface. At the top, there is a navigation bar with 'Postman' logo, 'File', 'Edit', 'View', 'Help', 'Home' (with a pencil icon), 'Workspaces' (highlighted with a red box), 'API Network', 'Reports', 'Explore', and a search bar 'Search Postman'. Below the navigation bar, the main content area has a heading 'APIs don't sleep either' and a sub-heading 'Pick up where you left off, catch up with your team's'. On the left, there is a sidebar with a 'G' logo and the title 'galactic-station-364346'. Below it are three buttons: 'Workspaces', 'Private API Network', and 'Integrations'. On the right, under 'Recently visited workspaces', there is a list of workspace names: 'API_APIC-EM_DEMO', 'API_Demo', 'API', 'LAB_DEVNET', and 'Cisco DNA Center'.

Hình 3.3.2.1 Tạo WorkSpaces 1

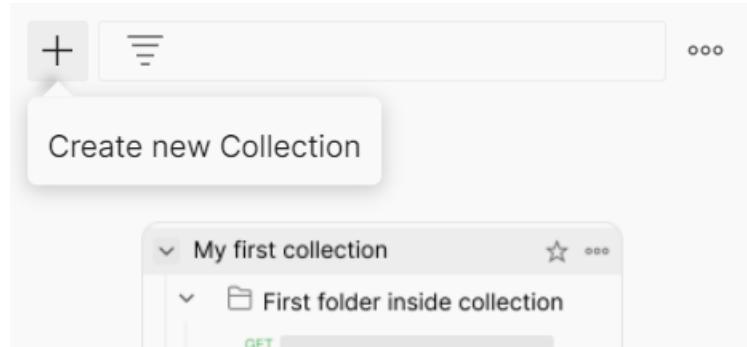
Sau đó ta điền tên và chọn chế độ truy cập. Sau khi hoàn thành ta sẽ được giao diện như hình bên dưới

The screenshot shows the 'DACNTT1_DEMO' workspace overview in Postman. The left sidebar includes tabs for 'Collections' (selected), 'APIs', 'Environments', 'Mock Servers', 'Monitors', 'Flows', and 'History'. The main content area shows an 'Overview' tab with a summary of the workspace: 'Demo Dự án CNTT1', 'Add a description to explain all about this workspace', 'Activity' (showing 'Hai created this team workspace just now'), and sections for 'In this workspace' (Requests, Collections, APIs, Environments, Mock Servers, Monitors, Flows) and 'Recent contributors' ('You'). At the bottom, there are links for 'Find and Replace', 'Console', and various system icons.

Hình 3.3.2.2 WorkSpace được tạo 1

b. Tạo Collection

Tiếp theo ta chọn vào dấu + ở mục Collection để thêm Collection mới. Collection để chứa các request của ta đến APIC-EM gồm thông tin của ticket, thông tin thiết bị,....

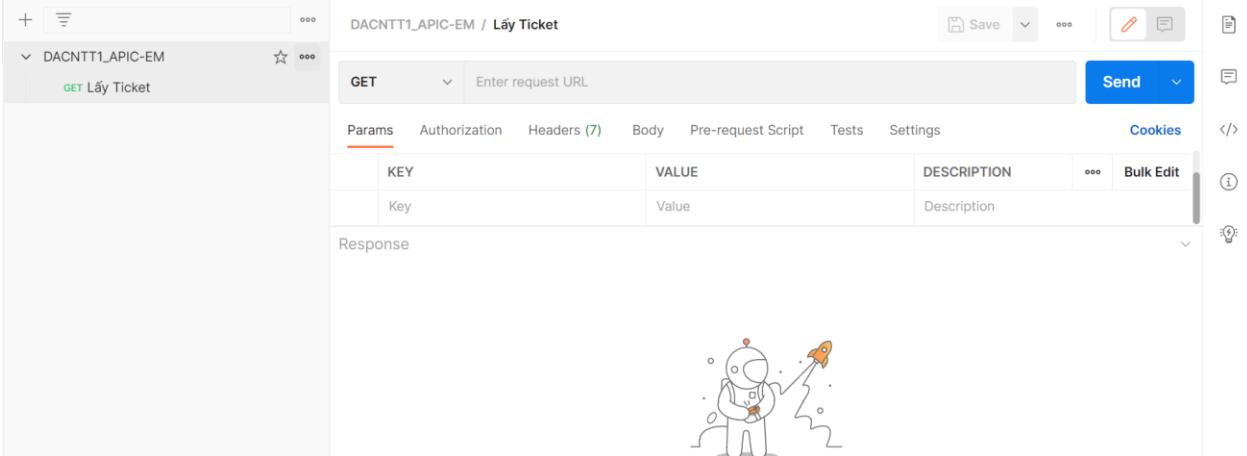


Hình 3.3.2.3 Tạo Collection 1

Hình 3.3.2.4 Collection được tạo 1

Tiếp theo ở mục DACNTT1_APIC-EM ta tạo các Request đến APIC-EM

c. Tạo Request lấy Ticket



Hình 3.3.2.5 Tạo Request xin ticket 1

Sau đó truy cập API Document do APIC-EM cung cấp để thực hiện xác định phương thức và URL cơ sở dẫn đến tài nguyên của APIC-EM

Hình 3.3.2.6 Tài liệu API của network controller APIC-EM 1

Để Postman có thể truy cập vào Network Controller thì Postman phải gửi đến Network Controller ID và Password. Nếu xác thực thành công thì Network Controller sẽ gửi về Postman một dãy số gọi là Ticket. Với Ticket này Postman có thể thực hiện các thao tác Get, Post, ... và Ticket

d. Tạo biến môi trường trong Postman để lưu các thông tin về Base URL

- Trong Postman ta chọn vào mục Environments để tạo thông tin biến môi trường:

The screenshot shows the Postman application interface. On the left, there's a sidebar with options like Collections, APIs, Environments (which is selected), Mock Servers, Monitors, Flows, and History. The main area shows a collection named 'DACNTT1_DEMO' with a single request named 'Lấy Ticket'. The request is set to 'POST' and has a placeholder URL 'Enter request URL'. Below the request, there's a table for 'Params' with one entry: 'Key' and 'Value'. A note says 'An environment is a set of variables that allows you to switch the context of your requests.' and a button 'Create Environment'. At the bottom, there are buttons for 'Cookies', 'Capture requests', 'Bootcamp', 'Runner', 'Trash', and help.

Hình 3.3.2.7 Tạo môi trường để lưu các biến. 1

Ta nhấn dấu + để tạo nơi chứa các biến môi trường mới. Sau đó chúng ta tạo lần lượt là baseURL, user_name, pass_word để có thể gọi lại và tăng tính bảo mật cho các yêu cầu

This screenshot shows the 'Environments' section of Postman. It lists an environment named 'API_APIC-EM'. Inside this environment, there are three variables defined: 'baseURL' (set to 'https://10.215.26.122/api/v1'), 'user_name' (set to 'admin'), and 'pass_word' (set to 'vnpro@149'). There are buttons for 'Fork', 'Save', 'Share', and 'Persist All' or 'Reset All'. The sidebar on the left remains the same as in the previous screenshot.

Hình 3.3.2.8 Tạo biến môi trường để gán các giá trị. 1

Sau đó ta trở lại mục Collection để bắt đầu lấy ticket bằng phương thức POST. Để có thể sử dụng các biến mình vừa tạo thì ở góc trên bên phải có mục No Environment ta chọn lại thành API_APIC-EM để sử dụng các biến.

e. Lấy Ticket

Khi đó ở mục điền URL ta gọi lại các biến bằng cách sử dụng cú pháp {{<tên _ biến>}} như hình bên dưới

The screenshot shows the Postman interface with a POST request to {{baseURL}}/ticket. The 'Params' tab is active, displaying a single parameter 'Key' with 'Value' 'Value'. The 'Headers' tab is also visible. A cartoon illustration of an astronaut launching a rocket is displayed in the center of the interface.

Hình 3.3.2.9 Điền URL 1

Tiếp tục đến phần Header ta điền vào Key: Content-Type và Value: application/json

The screenshot shows the Postman interface with a POST request to {{baseURL}}/ticket. The 'Headers' tab is active, showing a header 'Content-Type' with value 'application/json'. Other tabs like 'Params' and 'Body' are visible.

Hình 3.3.2.10 Điền kiểu dữ liệu trả về. 1

Tiếp tục qua phần Body ta chọn raw để khai báo username, password đăng nhập vào APIC-EM để xác thực yêu cầu Controller lấy ticket

The screenshot shows the Postman application interface. On the left, there's a sidebar with options like Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The main area displays a collection named 'DACNTT1_DEMO' with a single API endpoint named 'Lấy Ticket'. The request method is set to 'POST' and the URL is {{baseURL}}/ticket. The 'Body' tab is active, showing a JSON structure with two fields: 'username' and 'password', each containing a placeholder {{user_name}} and {{pass_word}} respectively. Below the body, there's a 'Response' section with a cartoon character holding a rocket. At the bottom, there are various toolbars and buttons for managing requests and responses.

Hình 3.3.2.11 Khai báo thông tin username và password đăng nhập vào APIC-EM 1

Sau khi hoàn tất điền các thành phần như trên ta sẽ lưu lại Request và nhấn Send để gửi yêu cầu lấy user ticket. Thành công thì postman sẽ trả về các thông tin như hình bên dưới.

The screenshot shows the Postman application interface. On the left, there's a sidebar with icons for Home, Workspaces, API Network, Reports, and Explore. The main area displays a collection named 'DACNTT1_DEMO'. Inside this collection, there's a POST request named 'Lấy Ticket' under the 'DACNTT1_APIC-EM' environment. The request URL is {{baseURL}}/ticket. The 'Body' tab is selected, showing a JSON payload with 'username' and 'password' fields. The response tab shows a 200 OK status with a response body containing a ticket object with serviceTicket, idleTimeout, sessionTimeout, and version fields. Below the main interface, there are buttons for Find and Replace, Console, Cookies, Capture requests, Bootcamp, Runner, and Trash.

Hình 3.3.2.12 Thông tin ticket được trả về. 1

Sau đó ta vào biến môi trường để tạo thêm biến chứa ticket vừa lấy để sử dụng lại

The screenshot shows the Postman application interface. The sidebar on the left has icons for Home, Workspaces, API Network, Reports, and Explore. The main area shows the 'API_APIC-EM' environment under the 'DACNTT1_DEMO' collection. The 'Globals' section is selected, displaying a table of environment variables. There are four variables: 'baseURL' (default value: https://10.215.26.122/api/v1), 'user_name' (default value: admin), 'pass_word' (default value: vnpro@149), and 'Ticket' (default value: empty). Below the table, there's a button to add a new variable. At the bottom, there are buttons for Find and Replace, Console, Cookies, Capture requests, Bootcamp, Runner, and Trash.

Hình 3.3.2.13 Tạo biến môi trường để lưu giá trị Ticket. 1

Sau đó ta trở lại kết quả vừa lấy về quét khôi nội dung serviceTicket sau đó nhấp chuột phải chọn Set API_APIC-EM sau đó chọn biến vừa tạo để gán giá trị ticket vào biến.

Hình 3.3.2.14 Gán giá trị ticket cho biến vừa tạo 1

f. Lấy danh sách thiết bị có trong Controller APIC-EM

Để lấy được danh sách thiết bị ta cần tạo Request mới sau đó đổi phương thức thành GET, phần API query từ “/ticket” thành “/network-device” theo hướng dẫn của API Document của APIC-EM cung cấp

Hình 3.3.2.15 Thông tin API truy cập lấy thông tin thiết bị trong APIC-EM 1

The screenshot shows the Postman application interface. On the left, there's a sidebar with icons for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The main area displays a collection named 'DACNTT1_DEMO'. Inside the collection, there are two items: a POST request for 'Lấy Ticket' and a GET request for 'Lấy danh sách thiết bị'. The GET request is currently selected. The 'Headers' tab is open, showing 'Content-Type: application/json' and 'X-Auth-Token: {{Ticket}}'. Other tabs include Params, Authorization, Body, Pre-request Script, Tests, Settings, and Cookies. A 'Send' button is visible at the top right of the request configuration area. Below the request, there's a 'Response' section with a placeholder message: 'Click Send to get a response' and an illustration of an astronaut launching a rocket.

Hình 3.3.2.16 Điền thông tin URL để lấy thông tin thiết bị. 1

Sau đó qua phần Headers ta thêm trường Key: “X-Auth-Token” và Value: “{{Ticket}}” để truyền user ticket vừa tạo vào xác thực để có thể lấy thông tin danh sách thiết bị

This screenshot shows the same Postman interface as the previous one, but with the 'Headers' tab selected in the configuration bar. The 'X-Auth-Token' header is now explicitly defined with the value '{{Ticket}}'. The other headers listed are 'Content-Type: application/json'. The rest of the interface, including the collection structure and the 'Send' button, remains the same.

Hình 3.3.2.17 Gọi lại biến có giá trị ticket vừa lưu 1

Sau đó Save và nhấn Send để lấy danh sách thiết bị

The screenshot shows the Postman interface with the following details:

- Header Bar:** Postman, File, Edit, View, Help.
- Toolbar:** Home, Workspaces, API Network, Reports, Explore, Search Postman, Invite, Upgrade.
- Request Panel:**
 - Method: GET, URL: {{baseUrl}}/network-device
 - Headers: (11)
 - Body: (green dot)
 - Pre-request Script, Tests, Settings.
- Response Panel:**
 - Status: 200 OK, Time: 194 ms, Size: 22.09 KB.
 - Save Response.
 - Pretty, Raw, Preview, Visualize, JSON.
 - Code block (highlighted):

```

1  {
2      "response": [
3          {
4              "serialNumber": null,
5              "errorCode": "DEV-UNREACHED",
6              "family": "Routers",
7              "type": "Cisco GatewayServer",
8              "location": null,
9              "role": "CORE",
10             "lastUpdateTime": 1650567762557,
11             "softwareVersion": "15.7(3)M2",
12             "inventoryStatusDetail": "<status><general code=\"DEV_UNREACHED\"/></status>",
13             "collectionInterval": "Global Default",
14             "macAddress": "aa:bb:cc:00:3c:00",
15             "errorDescription": "SNMP timeouts are occurring with this device. Either the SNMP credentials are not correctly provided to controller or the device is not responding.",
16             "upTime": "0:21:47.81"
17         }
18     ]
19 }
```
- Bottom Navigation:** Find and Replace, Console, Cookies, Capture requests, Bootcamp, Runner, Trash.

Hình 3.3.2.18 Thực hiện gửi yêu cầu để lấy thông tin thiết bị trong APIC-EM. 1

Lấy danh sách thiết bị thành công. Sẽ trả về kết quả như hình bên trên.

g. Đếm số lượng thiết bị

The screenshot shows the Postman interface with the following details:

- Header Bar:** Postman, File, Edit, View, Help.
- Toolbar:** Home, Workspaces, API Network, Reports, Explore, Search Postman, Invite, Upgrade.
- Request Panel:**
 - Method: GET, URL: {{baseUrl}}/network-device
 - Headers: (11)
 - Body: (green dot)
 - Pre-request Script, Tests, Settings.
 - Cookies.
- Query Params:**

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Hình 3.3.2.19 Tạo requests yêu cầu đếm số lượng thiết bị. 1

DACNTT1_APIC-EM / Đếm số lượng thiết bị

GET {{baseUrl}}/network-device/count

Params Authorization Headers (11) Body ● Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Response

Hình 3.3.2.20 Thay đổi đường dẫn để đếm số lượng thiết bị 1

DACNTT1_APIC-EM / Đếm số lượng thiết bị

GET {{baseUrl}}/network-device/count

Params Authorization Headers (11) Body ● Pre-request Script Tests Settings

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON ↻

```

1
2     "response": 17,
3     "version": "1.0"
4

```

Hình 3.3.2.21 Kết quả trả về số lượng thiết bị hiện có trong APIC-EM 1

h. Lấy thông tin cấu hình

DACNTT1_APIC-EM / Lấy thông tin cấu hình

GET {{baseURL}}/network-device/config

Params Authorization Headers (11) Body ● Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Body		Status: 200 OK Time: 70 ms Size: 130.08 KB		Save Response
Pretty	Raw Preview Visualize JSON			

```

1  {
2    "response": [
3      {
4        "runningConfig": "\nBuilding configuration...\n\n Current configuration : 1325 bytes\n!\n! Last configuration change at 12:37:32
5        "id": "4b3747ba-9bab-41f0-872b-bf32d85873ab"
6      },
7      {
8        "runningConfig": "\nBuilding configuration...\n\n Current configuration : 1258 bytes\n!\n! Last configuration change at 05:08:42
9        "id": "c24c2f0d-fb1b-4551-86ba-3d7ffe666938"
10     },
11     {
12       "runningConfig": "\nBuilding configuration...\n\n Current configuration : 1325 bytes\n!\n! Last configuration change at 12:39:01
13       "id": "a64ff059-96d2-4d53-bd59-1a192649e1ed"
14     },
15     {
16       "runningConfig": "\nBuilding configuration...\n\n Current configuration : 1325 bytes\n!\n! Last configuration change at 12:32:22
17       "id": "b0e78a34-02b8-47ea-9af4-446c17bab542"
18     },
19     {
20       "runningConfig": "\nBuilding configuration...\n\n Current configuration : 1348 bytes\n!\n! Last configuration change at 11:49:59
21       "id": "50fd3c87-b95c-449d-8791-13f319b06b00"
22     }
23   ]
24 }
```

Hình 3.3.2.22 Thay đổi đường dẫn để lấy thông tin cấu hình. 1

3.4 Sử dụng thư viện requests

Ngoài cách sử dụng công cụ Postman để truy xuất thông tin thiết bị thì ta cũng có thể sử dụng một công cụ mạnh khác để truy xuất thông tin thiết bị bằng REST API là sử dụng thư viện requests của ngôn ngữ Python.

a. Lấy ticket

Đầu tiên ta vào Command Promt để cài đặt thư viện requests.

```
pip install requests
```

Tiếp theo ta cần import thư viện requests, json và file chứa thông tin để sử dụng.

```
import requests
import json
import info_account
```

Tiếp theo viết hàm khai báo đường dẫn URL đến tài nguyên API của APIC-EM

```
def get_ticket():
    url_ticket = info_account.base_url + "/ticket"
```

Tiếp theo khai báo headers và body.

```
header = {
    "Content-Type": "application/json"
}
body = json.dumps({
    # Ép
    # kiểu dữ liệu về kiểu JSON
    "username": info_account.username,
    "password": info_account.password
})
```

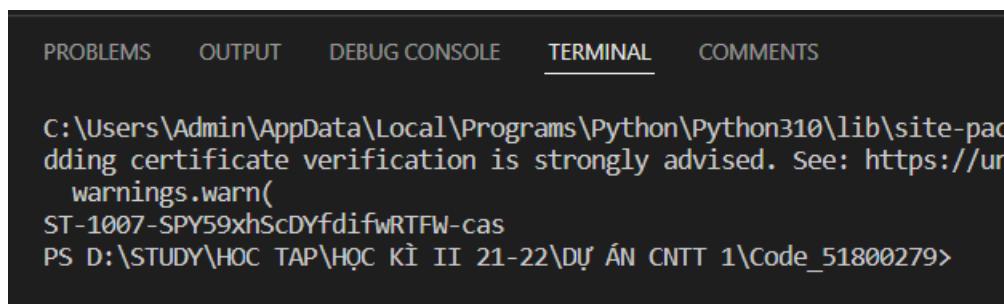
Tiếp theo khai báo phương thức kết nối là POST và truyền vào các tham số đã được khai báo bên trên để gửi yêu cầu tạo Ticket đến APIC-EM.

```
send_request = requests.post(url_ticket, headers = header,  
data=body, verify=False)
```

Sau đó ép kiểu từ json về dạng chuỗi để python có thể hiểu là một danh sách chứa các cặp key và value bao gồm ticket và sau đó truy xuất đến các phần tử serviceTicket để lấy ticket được trả về.

```
response = send_request.json()
ticket = response["response"]["serviceTicket"]
return ticket
```

Kết quả ticket được trả về:



Hình 3.4.1 Kết quả Ticket được trả về 1

b. Lấy danh sách thiết bị

Đầu tiên khai báo URL để truy cập lấy danh sách thiết bị trong APIC-EM

```
| def get list networkdevice():
```

```
url_networkdevice = info_account.base_url + "/network-device"
```

Khai báo headers xác thực ticket vừa được trả về:

```
headers = {
    "x-auth-token": get_ticket()
}
```

Tiếp theo khai báo phương thức kết nối là GET và truyền vào các tham số đã được khai báo bên trên để gửi yêu cầu lấy danh sách thiết bị đến APIC-EM.

```
devices = requests.get(url_networkdevice, headers = headers,
verify=False)
```

Sau đó ép kiểu từ json về dạng chuỗi để python có thể hiểu là một danh sách chứa thông tin của các con thiết bị bao gồm thông tin về vị trí, cấu hình, ip, serial number,.....

```
response = json.dumps(devices.json(), indent=4)
data = devices.json()
return data
```

Hàm sẽ trả về thông tin của các thiết bị để phục vụ truy xuất các thông tin bên dưới như cấu hình thiết bị và id của các thiết bị có trong APIC-EM.

Kết quả:

```

{
    "macAddress": "aa:bb:cc:00:3c:00",
    "errorDescription": "SNMP timeouts are occurring with this device. Either the SNMP credentials are not correctly provided to controller or the device is responding slow and snmp timeout is low. If its a timeout issue, controller will attempt to progressively adjust the timeout in subsequent collection cycles to get device to managed state. User can also run discovery again only for this device using the range discovery feature after adjusting the timeout and snmp credentials as required.",
    "uptime": "0:21:47.81",
    "series": "Other Routers",
    "locationName": null,
    "tagCount": "0",
    "hostname": "R1.vnpro.org",
    "roleSource": "AUTO",
    "apManagerInterfaceIp": "",
    "bootDateTime": "2022-03-03 06:51:30",
    "collectionStatus": "Partial Collection Failure",
    "interfaceCount": "4",
    "lineCardCount": null,
    "managementIpAddress": "10.215.26.245",
    "memorySize": "883975308",
    "platformId": null,
    "reachabilityFailureReason": "SNMP Connectivity Failed",
    "reachabilityStatus": "Unreachable",
    "snmpContact": ".",
    "snmpLocation": ".",
    "tunnelLdpPort": null,
    "lastUpdated": "2022-05-09 14:08:05",
    "instanceUuid": "f097659f-d035-4d94-b3a7-9c937d72917b",
    "id": "f097659f-d035-4d94-b3a7-9c937d72917b"
},
{
    "serialNumber": null,
    "errorCode": "DEV-UNREACHED",
    "family": "Routers",
    "type": "Cisco GatewayServer",
    "location": null,
    "role": "CORE",
    "lastUpdateTime": "1652105640905",
    "softwareVersion": "15.7(3)M2",
    "InventoryStatusDetail": "<status><general code='DEV_UNREACHED'></status>",
    "collectionInterval": "Global Default",
    "macAddress": "aa:bb:cc:00:1c:00",
    "errorDescription": "SNMP timeouts are occurring with this device. Either the SNMP credentials are not correctly provided to controller or the device is responding slow and snmp timeout is low. If its a timeout issue, controller will attempt to progressively adjust the timeout in subsequent collection cycles to get device to managed state. User can also run discovery again only for this device using the range discovery feature after adjusting the timeout and snmp credentials as required."
}

```

Hình 3.4.2 Danh sách thông tin thiết bị được trả về 1

c. Truy xuất thông tin cấu hình của các thiết bị trong APIC-EM

Viết hàm và khai báo url truy xuất thông tin cấu hình thiết bị APIC-EM

```
def get_device_config():
    url_networkdevice_config = info_account.base_url +
"/network-device/config"
```

Khai báo header với nội dung xác thực dựa vào giá trị trả về của hàm get ticket()

```
headers = {
    "x-auth-token": get_ticket()
}
```

Sử dụng phương thức get để yêu cầu APIC-EM trả về thông tin cấu hình thiết bị dựa vào các thông số đã khai báo.

```
    devices = requests.get(url_networkdevice_config, headers =  
headers, verify=False)  
    response = json.dumps(devices.json(), indent=4)  
    return response
```

Kết quả

Hình 3.4.3 Thông tin cấu hình của thiết bị 1

d. Truy xuất lấy Id của các thiết bị

Đặt tên hàm

```
def get_id_device():
```

Gọi lại hàm trả về thông tin danh sách thiết bị để phục vụ việc truy xuất

```
    data = get_list_networkdevice()
```

Viết lệnh thực hiện truy xuất thông tin Id

```
device_num = len(data["response"])
for i in range(device_num):
    print(data["response"][i]["id"])
```

Kết quả

```
warnings.warn(
C:\Users\Admin\AppData\Local\Programs\Python\Python310\lib\site-packages\urllib3\connectionpool.py:1043: InsecureRequestWarning: Unverified HTTPS request is being made to host '10.215.26.122'. Adding c
ertificate verification is strongly advised. See: https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
    warnings.warn(
f097659f-d035-4d94-b3a7-9c937d72917b
af28cb3d-89e6-439c-9ae7-50db7e6aa4bf
c5a018e1-de6e-473d-ab99-7cf7cd3210fe
adcc0ca3-03d1-4170-a11c-51ba50c7f964
78379db-d054-4dd9-90e3-88e77d0e94a6
4caf0135-a26f-4c8c-926b-9d0d0eb5deca0
ac011a54-73ac-43fa-9aa5-b84f6fd613600
07dea39f-80df-4878-aef5-ca6f6f97d1a12
3d6f593e-adb2-4ee2-358a-a2abbd95711
33df7956-b7b6-48c9-9a6b-6746187e7673
d0780180-e12a-4150-a3d0-079597a7
92e80806-ec54-4a05-9d55-218791079388
(jb93547-f8a7-44ec-a2f1-a1f1e1b95c0b)
c9106151-15f9-4277-9580-b8e1721d801b
f0fe4a8c-1542-4070-a912-3c72314d1927
a098a954-2513-4699-a0bf-3bc034706be8
1d1cc969-a77c-4edf-a8a9-be1b744e15df
PS D:\STUDY\HỌC TẬP\HỌC KÌ II 21-22\ĐỀ ÁN CNTT 1\Code_51800279>
```

Hình 3.4.4 Danh sách Id được truy xuất. 1

CHƯƠNG 4 – TỔNG KẾT

Trong bài báo cáo này em đã hoàn thành được mục tiêu là tìm hiểu về API một trong những xu hướng mới của ngành mạng trong tương lai sau này và khai thác API và sử dụng network controller do VnPro cung cấp để thực hành tương tác với APIC-EM thông qua REST API bằng công cụ là Postman và Python để có thể truy xuất được các thông tin của các thiết bị ở hạ tầng mạng một cách nhanh chóng và hiệu quả hơn sao với các cách truyền thống.

TÀI LIỆU THAM KHẢO

Bibliography

- [1] VnPro, "VnPro," [Online]. Available: <https://vnpro.vn/thu-vien/tong-quan-ve-api-application-programming-interface--3473.html>.
- [2] VnPro, "VnPro," 08 09 2021. [Online]. Available: <https://vnpro.vn/thu-vien/softwaredefined-networking-4121.html>.
- [3] E. Tittel, "SDN vs. NFV: What's the difference?," 22 04 2022. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/software-defined-networking/sdn-vs-nfv.html>.
- [4] C. O. V. D. Peterson, Software-Defined Networks: A Systems Approach, Sphinx, 2022.
- [5] ONF, "Software-Defined Networking (SDN) Definition," [Online]. Available: <https://opennetworking.org/sdn-definition/>.
- [6] Wikipedia, "Wikipedia," 22 04 2022. [Online]. Available: [https://en.wikipedia.org/wiki/Software-defined_networking#SDN_flow_forwarding_\(sdn\)](https://en.wikipedia.org/wiki/Software-defined_networking#SDN_flow_forwarding_(sdn)).
- [7] M. Alvarez, "What is a REST API? Fundamentals for Network Engineers," 2020. [Online]. Available: <https://codingnetworks.blog/what-is-a-rest-api-fundamentals-for-network-engineers/#:~:text=An%20API%20is%20an%20interface,built%20using%20slightly%20coupled%20components..>

PHỤ LỤC

Phần này bao gồm những nội dung cần thiết nhằm minh họa hoặc hỗ trợ cho nội dung luận văn như số liệu, biểu mẫu, tranh ảnh. . . nếu sử dụng những câu trả lời cho một bảng câu hỏi thì bảng câu hỏi mẫu này phải được đưa vào phần Phụ lục ở dạng nguyên bản đã dùng để điều tra, thăm dò ý kiến; **không được tóm tắt hoặc sửa đổi**. Các tính toán mẫu trình bày tóm tắt trong các biểu mẫu cũng cần nêu trong Phụ lục của luận văn. Phụ lục không được dày hơn phần chính của luận văn