



*Cisco
Developers
Connect
SuperCAT*

Nov 29th 2023, Seoul Korea

Ansible과 Meraki API로 클라우드 네트워킹 자동화하기

최혜연 프로, Meraki Technical Solutions Specialist



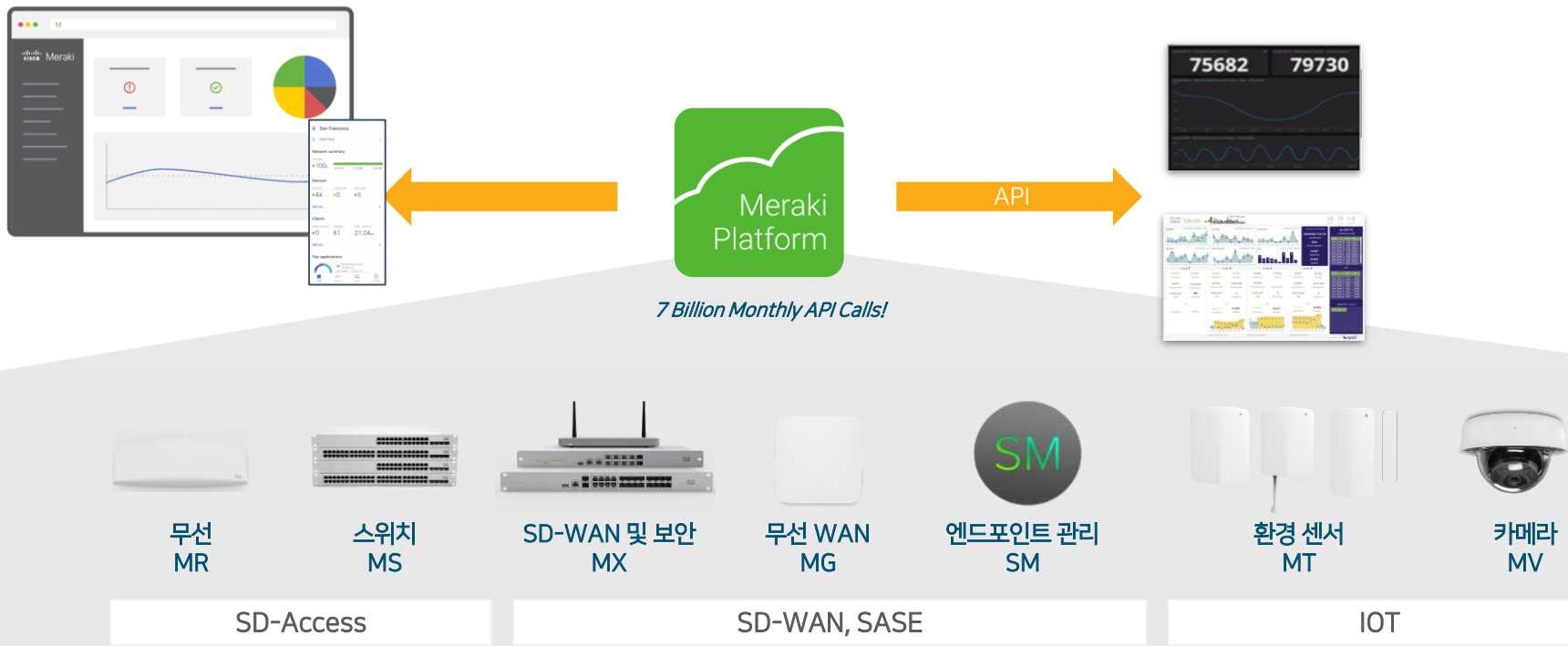
순서

- Meraki 소개
- Meraki API를 사용하면?
- Ansible for Meraki 실습
- 미션



Meraki 소개

클라우드 플랫폼 & API 솔루션



Meraki API 서비스



{APIs}

DASHBOARD API

- Programmability
- 자동화
- 모니터링
- 리포팅
- Data Insights

WEBHOOK API

- 이벤트 스트림
- 자동화 트리거

LOCATION STREAMING API

- Wayfinding (길찾기)
- 자산 추적
- 위치 분석

CAPTIVE PORTAL API

- 게스트 Wi-Fi
- 안전한 온보딩

MV SENSE API

- 실시간(4Hz) 데이터 스트림
- Historical time-series via REST
- 현재 snapshot

Meraki API를 사용하면?



비즈니스 성과



신규 수익 창출

비즈니스 인텔리전스, 매장 최적화, 근접 마케팅,
온/오프라인 마케팅



비용 절감

손실 방지, 지속 가능성



Risk

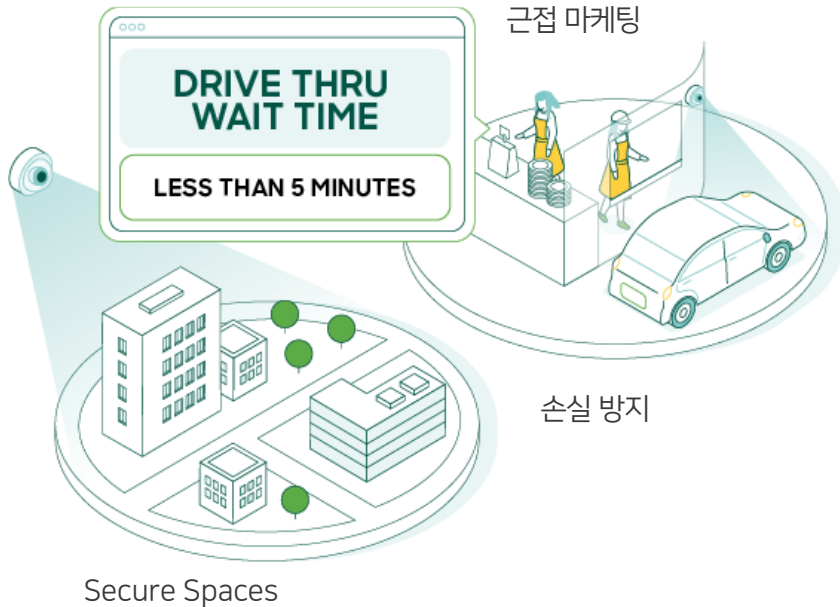
물리적 보안, 네트워크 보안, 안전



Workforce Experience 개선

하이브리드 워크, 스마트 스페이스

물리적 보안, 네트워크 보안, 안전



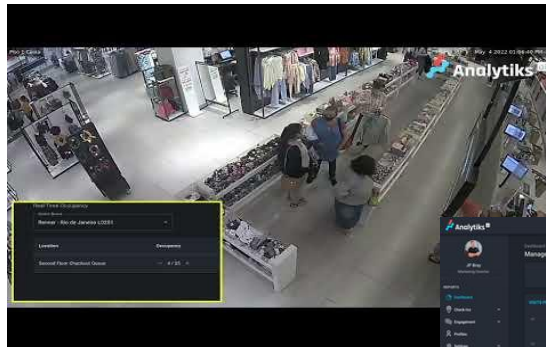
신규 수익 창출

Business Intelligence



물리적 공간의 디지털화

- 매장에 입장하는 방문자 수
- 매장에서 보낸 시간
- 방문 빈도
- 시간별/일별 트래픽
- 방문자 활동 분류
- 방문 고객 제품 구매율, 직원 대 방문자 비율
- 게스트 WiFi를 활용한 체크인



WiFi 및 카메라 분석 AI 플랫폼

Meraki API 서비스



{APIs}

DASHBOARD API

- Programmability
- 자동화
- 모니터링
- 리포팅
- Data Insights

WEBHOOK API

- 이벤트 스트림
- 자동화 트리거

LOCATION STREAMING API

- Wayfinding (길찾기)
- 자산 추적
- 위치 분석

CAPTIVE PORTAL API

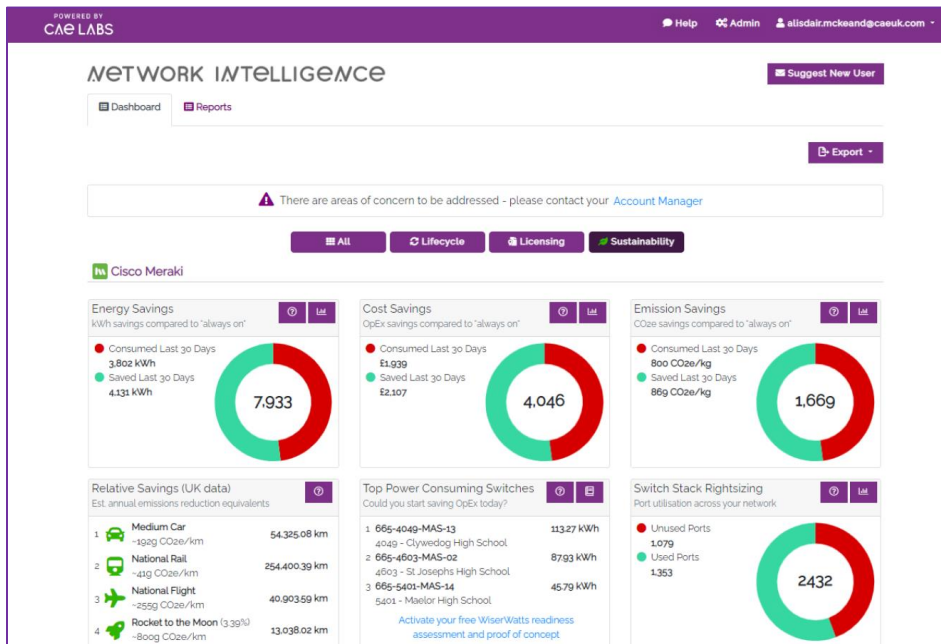
- 게스트 Wi-Fi
- 안전한 온보딩

MV SENSE API

- 실시간(4Hz) 데이터 스트림
- Historical time-series via REST
- 현재 snapshot

지속가능성 (ESG)

Cisco 기술 탈탄소화



#1

지속 가능성 추진의 장애물은
환경 영향을 측정하는 데
어려움이 있다는 것이다.

(Deloitte, 2022)



Discover



Act



Report

Meraki API 서비스



{APIs}

DASHBOARD API

- Programmability
- 자동화
- 모니터링
- 리포팅
- Data Insights

WEBHOOK API

- 이벤트 스트림
- 자동화 트리거

LOCATION STREAMING API

- Wayfinding (길찾기)
- 자산 추적
- 위치 분석

CAPTIVE PORTAL API

- 게스트 Wi-Fi
- 안전한 온보딩

MV SENSE API

- 실시간(4Hz) 데이터 스트림
- Historical time-series via REST
- 현재 snapshot

Meraki API 문서

<https://developer.cisco.com/meraki/api-v1/>

The screenshot shows the Meraki API v1 documentation page. The page layout includes a top navigation bar with links like 'Docs', 'Explore', 'What's new', 'Resources', and 'Marketplace'. The main content area is titled 'Meraki Dashboard API' and describes it as a RESTful API for managing Meraki networks. A green cloud icon with a code symbol is present. The page is divided into several sections: 'Introduction', 'Authentication', 'Getting Started', 'API Changelog', 'Guides', 'API Reference', 'What can you do with it?', 'What's New in v1', and 'Responses'.

Annotations include:

- A yellow box highlighting the 'API Reference' section in the left sidebar.
- A yellow box highlighting the 'Explore' section in the top navigation bar.
- A yellow box highlighting the 'API' section in the left sidebar.
- A yellow box highlighting the 'Responses' section in the right sidebar.

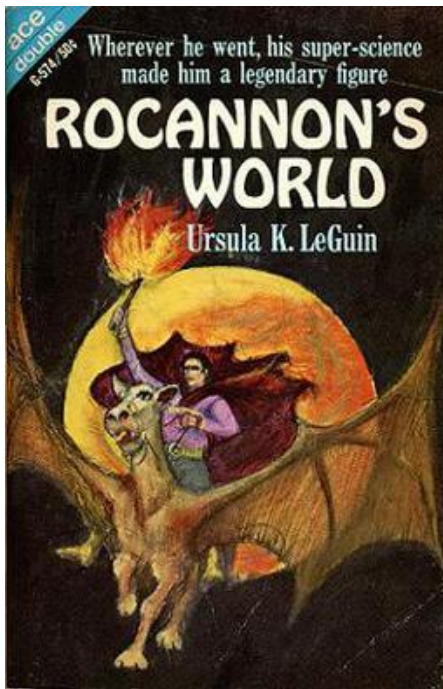
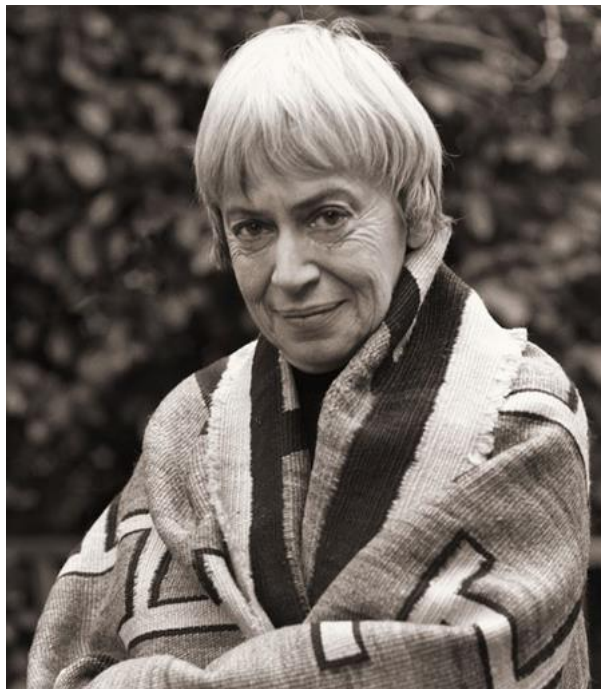
The 'API Reference' section lists various endpoints under categories like Platform, Products, and Monitor. The 'Responses' section shows a sample JSON response for a 'switch' endpoint, indicating a successful operation with a status of 200.

```
portDescription : "PORT 20",
"systemCapabilities": "switch"
},
"clientCount": 10,
"powerUsageInW": 55.9,
"trafficInKbps": {
  "total": 2.2,
  "sent": 1.2,
  "recv": 1
},
"securePort": {
  "enabled": true,
  "active": true
}
```

DAILY
KOREAN

실전에서 적용해 봅시다

Ansible?? Ursula K. Le Guin!!



- 로캐논의 세계 (1966)
빛의 속도보다 빨리 정보를 전송할 수 있는 허구의 통신 장치, 성간 거리에서도 메시지에 대한 답변을 받을 수 있음 (*answerable*)
- 엔더의 게임 (1977, 오슨 스콧 카드)
먼 거리에서 많은 원격 우주선을 동시에 통제하는데 사용

Using Ansible with Cisco Meraki 실습

<https://developer.cisco.com/learning/labs/meraki-dashboard-ansible/using-ansible-with-cisco-meraki/>

The screenshot shows the Cisco DevNet Learning Labs Center interface. The header includes the Cisco DevNet logo and 'Learning Labs Center'. The main title is 'Using Ansible with Cisco Meraki'. Below the title, there is a visual representation of the integration: a green square with the Meraki logo, a plus sign, and a black circle with the Ansible logo. A 'Talk to us' button is visible. A paragraph describes Ansible as an open-source software provisioning, configuration management, and application-deployment tool. A 'Start Learning' button is highlighted with an orange border and the Korean text '클릭!' (Click!). On the right side of the interface, there is a dark-themed code editor showing Ansible playbooks and a terminal window with the prompt 'Code as you learn!' and a message to 'Continue the learning lab to unlock the full interactive experience.'

이 실습에서는 Meraki Dashboard API와 Ansible을 함께 활용하는 방법을 배워봅니다.

위 링크로 로그인하세요!

Using Ansible with Cisco Meraki 실습

Using Ansible with Cisco Meraki

In this Lab, you learn how to use Ansible with the Cisco Meraki's Dashboard API.

Objectives

In this Lab you will:

- Learn how to install Cisco Meraki Ansible Collection.
- Connect to Meraki's Dashboard API using API keys.
- Perform queries against Meraki's Dashboard API using Ansible.
- Change settings in Meraki with Ansible.

Prerequisites

Before you begin, you should:

- Be familiar with Ansible and Ansible Galaxy.
- Have access to a Cisco Meraki account. Cisco [DevNet Sandbox](#) can be used if you do not have an account.

Search ...

Editor

< >

.cache + -

src + -

.bash_logout -

.bashrc -

profile -

이번 실습을 통해 할 일

1. Meraki Ansible Collection 설치

2. API Key 사용하여 Meraki Dashboard API에 연결

3. Ansible 사용하여 Meraki Dashboard API 정보 query

4. Ansible 사용하여 Meraki 설정 변경

Terminal x +

developer:src >

Using Ansible with Cisco Meraki 실습

Meraki Ansible Collection 설치

Using Ansible with Cisco Meraki

Ansible and Meraki Collection Installation

The Learning Labs environment is Linux-based. Install Ansible using Python's pip utility.

1

`pip install ansible`

Ansible 설치

2

The output is similar to the following example:

```
Collecting ansible
  Downloading ansible-8.2.0-py3-none-any.whl (45.1 MB)
    45.1/45.1 MB 28.8 MB/s eta 0:00:00
Collecting ansible-core<=2.15.2
  Downloading ansible_core-2.15.3-py3-none-any.whl (2.2 MB)
    2.2/2.2 MB 35.7 MB/s eta 0:00:00
...
Successfully installed MarkupSafe-2.1.3 ansible-8.2.0 ansible-core-2.15.3 jinja2-3.1.2 resolvelib-1.0.1
```

Terminal

```
developer:src > pip install ansible
Defaulting to user installation because normal site-packages is not writeable
Collecting ansible
  Downloading ansible-9.0.1-py3-none-any.whl (48.0 MB)
    48.0/48.0 MB 34.0 MB/s eta 0:00:00
Collecting ansible-core==2.16.0
  Downloading ansible_core-2.16.0-py3-none-any.whl (2.2 MB)
    2.2/2.2 MB 138.3 MB/s eta 0:00:00
Requirement already satisfied: cryptography in /usr/local/lib/python3.10/site-packages (from ansible-core==2.16.0->ansible) (2.8)
Requirement already satisfied: PyYAML>=5.1 in /usr/local/lib/python3.10/site-packages (from ansible-core==2.16.0->ansible) (5.1.2)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/site-packages (from ansible-core==2.16.0->ansible) (21.3)
Collecting Jinja2>=3.0.0
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
    133.1/133.1 kB 12.9 MB/s eta 0:00:00
Collecting resolvelib<1.1.0,>=0.5.3
  Downloading resolvelib-1.0.1-py2.py3-none-any.whl (17 kB)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.3-cp310-cp310-manylinux_2_17_x86_64_manylinux2014_x86_64.whl (25 kB)
Requirement already satisfied: cffi!=1.11.3,>=1.8 in /usr/local/lib/python3.10/site-packages (from cryptography->ansible-core==2.16.0->ansible) (1.15.0)
Requirement already satisfied: six>=1.4.1 in /usr/local/lib/python3.10/site-packages (from cryptography->ansible-core==2.16.0->ansible) (1.13.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.10/site-packages (from packaging->ansible-core==2.16.0->ansible) (3.0.9)
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/site-packages (from cffi!=1.11.3,>=1.8->cryptography->ansible-core==2.16.0->ansible) (2.19)
Installing collected packages: resolvelib, MarkupSafe, Jinja2, ansible-core, ansible
Successfully installed MarkupSafe-2.1.3 ansible-9.0.1 ansible-core-2.16.0 Jinja2-3.1.2 resolvelib-1.0.1
WARNING: There was an error checking the latest version of pip.
developer:src > []
```

Using Ansible with Cisco Meraki 실습

Meraki Ansible Collection 설치

Learning Labs Center

Using Ansible with Cisco Meraki

You can now check the Ansible version and Python in use.

`ansible --version`

Ansible 버전 확인

클릭

The output is similar to the following example:

```
ansible [core 2.15.3]
  config file = None
  configured module search path = ['/home/developer/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /home/developer/.local/lib/python3.10/site-packages/ansible
  ansible collection location = /home/developer/.ansible/collections:/usr/share/ansible/collections
  executable location = /home/developer/.local/bin/ansible
  python version = 3.10.4 (main, May 11 2022, 07:26:18) [GCC 10.2.1 20210110]
  jinja version = 3.1.2
  libyaml = False
```

`pip install meraki`

Search ...

Editor

Ansible 버전: 2.16.0

Python 버전: 3.10.4

Terminal

```
developer:src > ansible --version
ansible [core 2.16.0]
  config file = None
  configured module search path = ['/home/developer/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /home/developer/.local/lib/python3.10/site-packages/ansible
  ansible collection location = /home/developer/.ansible/collections:/usr/share/ansible/collections
  executable location = /home/developer/.local/bin/ansible
  python version = 3.10.4 (main, May 11 2022, 07:26:18) [GCC 10.2.1 20210110] (/usr/local/bin/python)
  jinja version = 3.1.2
  libyaml = False
developer:src > pip install meraki
Defaulting to user installation because normal site-packages is not writeable
Collecting meraki
  Downloading meraki-1.39.0-py3-none-any.whl (262 kB)
    262.4/262.4 kB 22.2 MB/s eta 0:00:00
Collecting aiohttp
  Downloading aiohttp-3.9.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.2 MB)
    1.2/1.2 MB 64.2 MB/s eta 0:00:00
Requirement already satisfied: requests in /usr/local/lib/python3.10/site-packages (from meraki) (2.22.0)
Collecting multidict<7.0,>=4.5
  Downloading multidict-6.0.4-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (114 kB)
    114.5/114.5 kB 31.0 MB/s eta 0:00:00
Collecting frozenlist<=1.1.1
```

SUPERCAT
Cisco Developers Connect - SuperCAT

Presentation ID

19

Using Ansible with Cisco Meraki 실습

Meraki Ansible Collection 설치

Using Ansible with Cisco Meraki

```
ansible python module location = /home/developer/.local/lib/python3.10/site-p
ansible collection location = /home/developer/.ansible/collections:/usr/share
executable location = /home/developer/.local/bin/ansible
python version = 3.10.4 (main, May 11 2022, 07:26:18) [GCC 10.2.1 20210110] (
jinja version = 3.1.2
libyaml = False
```

1

`pip install meraki`

Meraki 설치

클릭

The output is similar to the following example:

```
Defaulting to user installation because normal site-packages is not writeable
Collecting meraki
  Downloading meraki-1.36.0-py3-none-any.whl (255 kB)
    255.2/255.2 kB 25.5 MB/s eta 0:00
...
Installing collected packages: multidict, frozenlist, charset-normalizer, async
Successfully installed aiohttp-3.8.5 aiosignal-1.3.1 async-timeout-4.0.3 charse
```

2

Terminal 화면 출력이 동일한지 확인

```
developer@src ~$ pip install meraki
Defaulting to user installation because normal site-packages is not writeable
Collecting meraki
  Downloading meraki-1.39.0-py3-none-any.whl (262 kB)
    262.4/262.4 kB 22.2 MB/s eta 0:00:00
Collecting aiohttp
  Downloading aiohttp-3.9.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.2 MB)
    1.2/1.2 MB 64.2 MB/s eta 0:00:00
Requirement already satisfied: requests in /usr/local/lib/python3.10/site-packages (from meraki) (2.22.0)
Collecting multidict<7.0,>=4.5
  Downloading multidict-6.0.4-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (114 kB)
    114.5/114.5 kB 31.0 MB/s eta 0:00:00
Collecting frozenlist<=1.1.1
  Downloading frozenlist-1.4.0-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux20
4_x86_64.whl (225 kB)
    225.7/225.7 kB 59.2 MB/s eta 0:00:00
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/site-packages (from aiohttp->meraki) (19.
.0)
Collecting aiosignal<=1.1.2
  Downloading aiosignal-1.3.1-py3-none-any.whl (7.6 kB)
Collecting async-timeout<5.0,>=4.0
  Downloading async_timeout-4.0.3-py3-none-any.whl (5.7 kB)
Collecting yarl<2.0,>=1.9
  Downloading yarl-1.9.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (300 kB)
    300.7/300.7 kB 66.0 MB/s eta 0:00:00
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.10/site-packages (
rom requests->meraki) (1.25.6)
Requirement already satisfied: charset<3.1.0,>=3.0.2 in /usr/local/lib/python3.10/site-packages (from requests->mer
ki) (3.0.4)
Requirement already satisfied: certifi<=2017.4.17 in /usr/local/lib/python3.10/site-packages (from requests->meraki
(2019.9.11))
Requirement already satisfied: idna<2.9,>=2.5 in /usr/local/lib/python3.10/site-packages (from requests->meraki) (2
8)
Installing collected packages: multidict, frozenlist, async-timeout, yarl, aiosignal, aiohttp, meraki
Successfully installed aiohttp-3.9.0 aiosignal-1.3.1 async-timeout-4.0.3 frozenlist-1.4.0 meraki-1.39.0 multidict-6
0.4 yarl-1.9.3
WARNING: There was an error checking the latest version of pip.
```

Meraki Ansible Collection 설치



Using Ansible with Cisco Meraki 실습

Meraki Dashboard API Key 값 입력

Configure Meraki Dashboard API Settings

Now that you installed the Meraki Ansible Collection, you can use it to configure Meraki's Dashboard API settings. First you need API key credentials from your Meraki account. If you don't have an account, you can use [Cisco DevNet Sandbox](#).

Meraki API Keys

To generate a Meraki API Key, use [the following guide](#).

Meraki's Ansible Collection can use environment variables for API Key. Key data can also be set in playbooks, inventory files, or other variable sources from Ansible. In this Lab, the example playbooks will lookup data from environment variables.

Note: Do not save sensitive information (for example, credentials and API keys) in an unencrypted file that can accidentally be uploaded with the code to a public repository.

The following commands set the Ansible environment variables with data from the files that are created in the previous steps.

1

```
export MERAKI_DASHBOARD_API_KEY=<your API key>
```

Next >

<your API key>를 아래 값으로 변경
4c15c13572a2567dfdc16ccd35d0f8fe780ebb9d

Search ...

Editor

```
> .cache
> src
bash_logout
bashrc
profile
```

Terminal x +


```
its-packages (from requests>meraki) (1.25.6)
Requirement already satisfied: certifi>2017.4.17 in /usr/local/lib/python3.10/site-packages (from re
quests>meraki) (2019.9.11)
Installing collected packages: multidict, frozenlist, async-timeout, yarl, aiohttp, aiohttp, meraki
Successfully installed aiohttp-3.9.0 aiohttp-1.3.1 async-timeout-4.0.3 frozenlist-1.4.0 meraki-1.39
0 multidict-6.0.4 yarl-1.9.3
WARNING: There was an error checking the latest version of pip.
developer:src> ansible-galaxy collection install cisco.meraki --force
Starting galaxy collection install process
Process install dependency map
Starting collection install process
Downloading https://galaxy.ansible.com/api/v3/plugin/ansible/content/published/collections/artifacts/
cisco.meraki-2.16.15.tar.gz to /home/developer/.ansible/tmp/ansible-local-35xshide2y/tmpm1er0aed/cisc
o.meraki-2.16.15
Installing https://galaxy.ansible.com/api/v3/plugin/ansible/content/published/collections/ansible_collections/cisco/
meraki/
cisco.meraki-2.16.15 was installed successfully.
WARNING: cisco.meraki-2.16.15 is already installed, skipping.
developer:src> export MERAKI_DASHBOARD_API_KEY=6bec40cf957de430a6f1f2baa056b99a4fac9ea0
developer:src>
```


Using Ansible with Cisco Meraki 실습

Ansible Playbook 사용

The screenshot displays the Cisco DevNet Learning Labs Center interface for the 'Using Ansible with Cisco Meraki' lab. The interface is divided into two main sections: a left-hand sidebar with instructions and a right-hand workspace for file editing and terminal execution.



Left Panel (Instructions):

- Step 1:** Use the built-in editor on the right hand side (you can also use your favorite editor) to create and save a file named `hosts` (no extension):
Src 폴더에 hosts 파일 생성 (1)  **클릭** (클릭)
- Step 2:** Copy the following contents in the file named `'hosts'`. (You will find the newly created file under `src` folder on the right hand side in the built-in editor)

```
[meraki_servers]
meraki_server
```
- Step 3:** Create and save a playbook file named `who_am_i.yml` with the following contents.
touch ~/src/named_who_am_i.yml 
- Copy the following contents in the file named `'who_am_i.yml'`. (You will find the newly created file under `src` folder on the right hand side in the built-in editor)**

```
---
- hosts: meraki_servers
  gather_facts: false
  tasks:
    - name: Get my information and identify it
```

Right Panel (Workspace):

- File Explorer:** Shows the `src` folder containing `hosts` and `meraki_server`. **hosts 파일 생성된 것 확인** (2) 
- Terminal:** Shows the command `touch ~/src/hosts` being executed. **hosts 파일 내에 해당 내용 입력** (3) 

Using Ansible with Cisco Meraki 실습

Ansible Playbook 사용

Using Ansible with Cisco Meraki

102 mins

2. Copy the following contents in the file named 'hosts' (You will find the newly created file under 'src' folder on the right hand side in the built-in editor)

```
[meraki_servers]
meraki_server
```

3. Create and save a playbook file named who_am_i.yaml with the following contents.

1 클릭

touch ~/src/named who_am_i.yaml Src 폴더에 who_am_i 파일 생성

2 파일 생성된 것 확인

3 Who_am_i.yaml 파일 내에 내용 입력

```
1 ---
2 - hosts: meraki_servers
3   gather_facts: false
4   tasks:
5     - name: Get my administered identities
6       cisco.meraki.administered_identities_me_info:
7         register: result
8     - name: Show result
9       ansible.builtin.debug:
10        msg: "{{ result }}"
```

Copy the following contents in the file named 'who_am_i.yaml' (You will find the newly created file under 'src' folder on the right hand side in the built-in editor)

⚠ Be careful with the spacing and alignment of text in YAML files. If you deviate from this strict formatting, you get the errors. JavaScript uses {} to create structured blocks of data whereas YAML uses spaces. That is why it looks so clean - no curly braces - but you must be diligent with the spaces.

```
---
- hosts: meraki_servers
  gather_facts: false
  tasks:
    - name: Get my administered identities
      cisco.meraki.administered_identities_me_info:
        register: result
    - name: Show result
      ansible.builtin.debug:
        msg: "{{ result }}"
```

terminal

```
cisco-meraki-2.16.15.tar.gz to /home/developer/.ansible/tmp/ansible-local-35xshtde2y/tmpmier0aed/cisco-meraki-2.16.15-bzahvkat
Installing 'cisco.meraki:2.16.15' to '/home/developer/.ansible/collections/ansible_collections/cisco/meraki'
cisco.meraki:2.16.15 was installed successfully
'ansible.utils:2.11.0' is already installed, skipping.
developer:src > export MERAKI_DASHBOARD_API_KEY=6be
c40cf957de430a6f1f2baa056b99a4fac9ea0
developer:src > touch ~/src/named who_am_i.yaml
developer:src >
```


Using Ansible with Cisco Meraki 실습

Ansible Playbook 사용

Using Ansible with Cisco Meraki

119 mins

```
tasks:
- name: Get my administered identities
  cisco.meraki.administered_identities_me_info:
  register: result

- name: Show result
  ansible.builtin.debug:
  msg: "{{ result }}"
```

4. Run the playbook using the command:

```
ansible-playbook -i hosts who_am_i.yml
```

Click here to see the expected output

Finding the right Ansible module

With more than 400 available modules in the Meraki Ansible collection, it's important to understand how to find the right module for the task. The modules are named based on their path, which can be found on the [Dashboard API documentation site](#). For example, the operation for [Network Switch Settings](#) has the following URL: `/networks/{networkId}/switch/settings`, therefore its module is called `cisco.meraki.networks_switch_settings`.

The operation for [Device Camera Video Settings](#) has the following URL: `/devices/{serial}/camera/video/settings`, therefore its module is called `cisco.meraki.devices_camera_video_settings`.

The operation for [Get Organization Admins](#) has the following URL: `/organizations/{organizationId}/admins`, therefore its module is called `cisco.meraki.organizations_admins_info` and the module for updating admins is called `cisco.meraki.organizations_admins`.

Next >

who_am_i.yml

```
1 ---
2 - hosts: meraki_servers
3   gather_facts: false
4   tasks:
5     - name: Get my administered identities
6       cisco.meraki.administered_identities_me_info:
7       register: result
8
9     - name: Show result
10      ansible.builtin.debug:
11      msg: "{{ result }}"
```

실행 결과 확인

```
PLAY RECAP *****
meraki_server : ok=2  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

1 클릭

2 실행 결과 확인

Using Ansible with Cisco Meraki 실습

Ansible Playbook 사용

module > organizations_admins

Resource module for organizations_admins

- [Synopsis](#)
- [Parameters](#)
- [Notes](#)
- [Examples](#)
- [Return Values](#)

Synopsis

- Manage operations create, update and delete of the resource organizations_admins.
- Create a new dashboard administrator.
- Revoke all access for a dashboard administrator within this organization.
- Update an administrator.

400개 이상의 사용 가능한 모듈 존재

모듈명은 Dashboard API 문서 내의 경로에 따라 지정됨

예) 스위치 설정을 위한 Ansible 모듈명

Dashboard API 상 경로
/networks/{networkId}/switch/settings

Meraki ansible 모듈명
cisco.meraki.networks_switch_settings

Quiz!!!

cisco.meraki.organizations_admins

Using Ansible with Cisco Meraki 실습

Ansible Collection 예시

Using Ansible with Cisco Meraki

76 mins

Ansible Collection Help and Query Examples

Now that you have configured API authentication, you can run example playbooks that query Meraki's Dashboard API. To view examples, change directory to the playbooks subdirectory that was installed with the Collection.

cd /home/developer/.ansible/collections/ansible_collections/cisco/meraki/playbooks

1 클릭

Query the API for data

Meraki's Dashboard API can be used to query various information. For example, list all of the organizations the API key has access to (equivalent to [this API endpoint](#) and):

```
---
- hosts: meraki_servers
  gather_facts: false
  tasks:
    - name: Get all Organizations
      cisco.meraki.organizations_info:
        register: result

    - name: Show result
      ansible.builtin.debug:
        msg: "{{ result }}"
```

hosts

[meraki_servers]

meraki_server

2 Is 실행하여 해당 디렉토리 내의 playbook 목록 확인

Terminal x +

developer:playbooks > cd /home/developer/.ansible/collections/ansible_collections/cisco/meraki/playbooks

developer:playbooks > ls

administered_identities_me_info.yml networks_devices_remove.yml

credentials.template networks_wireless_ssids_identityPsk.yml

credentials.yml old_collection_test.yml

device_blink_leds.yml organization.yml

device_idp.cdp.yml organization_info.yml

device_statuses.yml organization_summary.yml

devices.yml organizations_adaptivePolicy_acs.yml

devices_info.yml organizations_adaptive_policy_groups.yml

devices_live_tools_ping.yml 'organizations_admin_copy.yml'

devices_switch_routing_interfaces.yml organizations_admin.yml

files organizations_admin_info.yml

hosts organizations_camera_custom_analytics_artifacts.yml

meraki_api_log_2023-11-23_13-29-09.log organizations_clients_search_info.yml

meraki_api_log_2023-11-23_13-30-10.log organizations_config_templates.yml

meraki_api_log_2023-11-23_13-31-44.log organizations_config_templates_info.yml

meraki_api_log_2023-11-23_13-33-05.log organizations_devices_info.yml

meraki_api_log_2023-11-23_13-33-50.log organizations_login_security.yml

meraki_api_log_2023-11-23_13-35-47.log organizations_saml_idps.yml

ms_playbook.yml switch_port_config.yml

mv_playbook.yml test.yml

mx_deployment.yml who_am_i.yml

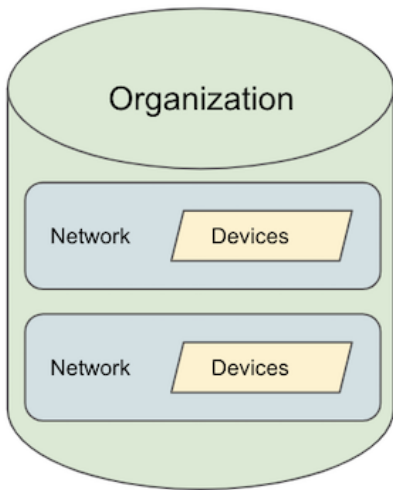
network_devices_claim.yml wifi_deploy_one.yml

networks.yml wifi_deployment.yml

networks_appliance_vlans.yml

developer:playbooks > []

Meraki Dashboard API 기본



<Meraki API 구조>

1. Base URI = <https://api.meraki.com/api/v1>
2. Meraki Dashboard에 로그인
3. *Organization > Settings*에서 Dashboard API access 활성화

Dashboard API access

API Access ⓘ

☒ Enable access to the Cisco Meraki Dashboard API

After enabling the API here, go to your [profile](#) to generate an API key. The API will return 404 for requests with a missing or incorrect API key.

4. 대시보드 오른쪽 상단 *My profile* 클릭
5. API 사용 권한 등을 위해 API Keys 사용

API access

API keys

Key

Generate new API key

Using Ansible with Cisco Meraki 실습

퀴즈! Organization 리스트를 확인할 수 있는 플레이북은 무엇일까요?

```
Terminal x + A- A+
developer:playbooks >
developer:playbooks > cd /home/developer/.ansible/collections/ansible_collections/cisco/meraki/playbooks
developer:playbooks > ls
administered_identities_me_info.yml      networks_devices_remove.yml
credentials.template                     networks_wireless_ssids_identityPsks.yml
credentials.yml                           old_collection_test.yml
device_blink_leds.yml                    organization.yml
device_lldp_cdp.yml                       organization_info.yml
device_statuses.yml                       organization_summary.yml
devices.yml                               organizations_adaptivePolicy_acls.yml
devices_info.yml                          organizations_adaptive_policy_groups.yml
devices_live_tools_ping.yml               'organizations_admin_copy.yml'
devices_switch_routing_interfaces.yml      organizations_admin.yml
files                                     organizations_admin_info.yml
hosts                                    organizations_camera_custom_analytics_artifacts.yml
meraki_api_log_2023-11-23_13-29-09.log     organizations_clients_search_info.yml
meraki_api_log_2023-11-23_13-30-10.log     organizations_config_templates.yml
meraki_api_log_2023-11-23_13-31-44.log     organizations_config_templates_info.yml
meraki_api_log_2023-11-23_13-33-05.log     organizations_devices_info.yml
meraki_api_log_2023-11-23_13-33-50.log     organizations_login_security.yml
meraki_api_log_2023-11-23_13-35-47.log     organizations_saml_idps.yml
ms_playbook.yml                           switch_port_config.yml
mv_playbook.yml                           test.yml
mx_deployment.yml                         who_am_i.yml
network_devices_claim.yml                  wifi_deploy_one.yml
networks.yml                               wifi_deployment.yml
networks_appliance_vlans.yml
developer:playbooks > |
```

Using Ansible with Cisco Meraki 실습

Organization 정보 불러오기

```
developer:playbooks > ansible-playbook -i hosts organization_info.yml

PLAY [localhost] *****

TASK [Get all Organizations] *****
An exception occurred during task execution. To see the full traceback, use -vvv. The error was: NoneType: None
fatal: [localhost]: FAILED! => [{"changed": false, "msg": ["missing required arguments: meraki_api_key"]}

PLAY RECAP *****
localhost : ok=0    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
```

1. organization_info playbook 실행 시
에러가 나는 이유는?

```
developer:playbooks > export MERAKI_DASHBOARD_API_KEY=6bec40cf957de430a6f1f2baa056b99a4fac9ea0
developer:playbooks > ansible-playbook -i hosts organization_info.yml

PLAY [localhost] *****

TASK [Get all Organizations] *****
2023-11-23 13:54:52 meraki: INFO > Meraki dashboard API session initialized with these parameters: {'version': '1.39.0', 'api_key': '*****9ea0', 'base_url': 'https://api.meraki.com/api/v1', 'single_request_timeout': 60, 'certificate_path': '', 'requests_proxy': '', 'wait_on_rate_limit': True, 'nginx_429_retry_wait_time': 60, 'action_batch_retry_wait_time': 60, 'network_delete_retry_wait_time': 240, 'retry_4xx_error': False, 'retry_4xx_error_wait_time': 60, 'maximum_retries': 2, 'simulate': False, 'be_geo_id': None, 'caller': 'MerakiAnsibleCollection/1.0.0 Cisco', 'use_iterator_for_get_pages': False}
2023-11-23 13:54:52 meraki: INFO > GET https://api.meraki.com/api/v1/organizations/828899381482762766
2023-11-23 13:54:52 meraki: ERROR > organizations - getOrganization - 404 Not Found b''
An exception occurred during task execution. To see the full traceback, use -vvv. The error was: meraki.exceptions.APIError: organizations, getOrganization - 404 Not Found, please wait a minute if the key or org was just newly created
fatal: [localhost]: FAILED! => [{"changed": false, "msg": "An error occurred when executing operation.The error was: organizations, getOrganization - 404 Not Found, please wait a minute if the key or org was just newly created."}]

PLAY RECAP *****
localhost : ok=0    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
```

2. API key 값 복사 후에도 에러 발생

Using Ansible with Cisco Meraki 실습

Organization 정보 불러오기

```
developer:playbooks > cat organization_info.yml
---
- hosts: localhost
  gather_facts: false
  tasks:
    - name: Get all Organizations
      cisco.meraki.organizations_info:
        meraki_suppress_logging: false
        # meraki_username: "{{meraki_username}}"
        # meraki_password: "{{meraki_password}}"
        # meraki_verify: "{{meraki_verify}}"
        # meraki_port: "{{meraki_port}}"
        # meraki_version: "{{meraki_version}}"
        # meraki_debug: "{{meraki_debug}}"
        # headers: "{{my headers | from_iso}}"
        organizationId: "828099381482762766"
      register: result

    - name: Show result
      ansible.builtin.debug:
        msg: "{{ result }}"
developer:playbooks > []
```

1. Cat organization_info.yml로 플레이북 파일 확인
2. organizationId 라인 주석처리 필요

Organization 정보 불러오기

```

Terminal x +
-- hosts: meraki_servers
gather_facts: false
tasks:
  - name: Get all Organizations
    cisco.meraki.organizations.info:
      meraki_suppress_logging: false
      # meraki_username: "{{meraki_username}}"
      # meraki_password: "{{(meraki_password)}}"
      # meraki_verify: "{{(meraki_verify)}}"
      # meraki_port: "{{(meraki_port)}}"
      # meraki_version: "{{(meraki_version)}}"
      # meraki_debug: "{{(meraki_debug)}}"

      #organizationId: "828099381482762766"
register: result

```

- vi 파일 저장 및 종료하기

모드	명령키	설명
마지막 행 모드	:q	vi에서 작업한것이 없을때 vi 종료합니다.
	:q!	작업한 내용을 저장하지 않고 종료합니다.
	:w[파일명]	작업한 내용을 저장만 한다. 파일명을 지정하면 새 파일로 저장합니다.
	:wq, :wq!	작업한 내용을 저장하고 vi를 종료합니다.
명령 모드	ZZ (대문자)	작업한 내용을 저장하고 vi를 종료합니다.

내용을 편집한 후 vi를 종료하려면 명령모드나 마지막 행 모드로서 명령을 입력해야 합니다. 마지막 행 모드로 가려면 :을 누르면 됩니다. 명령모드에서 파일의 저장과 종료를 동시에 수행하려면 대문자ZZ를 누르시면 파일을 저장한 후 vi를 바로 종료합니다.

vi 입력모드로 전환하기

명령키	설명
I	현재 커서 앞에 입력합니다.
a	현재 커서 뒤에 입력합니다.
o	커서가 위치한 행의 다음 행에 입력합니다.
I(대문자 i)	커서가 위치한 행의 첫 컬럼으로 이동하여 입력합니다.
A	커서가 위치한 행의 마지막 컬럼으로 이동하여 입력합니다.
O	커서가 위치한 행의 이전 행에 입력합니다.



Using Ansible with Cisco Meraki 실습

Organization 정보 불러오기

```
developer:playbooks > cat organization_info.yml ①
---
- hosts: meraki_servers
  gather_facts: false
  tasks:
    - name: Get all Organizations
      cisco.meraki.organizations_info:
        meraki_suppress_logging: false
        # meraki_username: "{{meraki_username}}"
        # meraki_password: "{{meraki_password}}"
        # meraki_verify: "{{meraki_verify}}"
        # meraki_port: "{{meraki_port}}"
        # meraki_version: "{{meraki_version}}"
        # meraki_debug: "{{meraki_debug}}"
        # headers: "{{mv headers | from_json}}"
        #organizationId: "828099381482762766"
      register: result

    - name: Show result
      ansible.builtin.debug:
        msg: "{{ result }}"
developer:playbooks > ansible-playbook -i hosts organization_info.yml ②
```

1. `cat organization_info.yml`로 파일 내용 수정 잘 되었는지 확인
2. Playbook 재실행
3. Organization 정보 불러온 것 확인

```
③ TASK [Show result] *****
ok: [localhost] => {
  "msg": {
    "changed": false,
    "failed": false,
    "meraki_response": [
      {
        "api": {
          "enabled": true
        },
        "cloud": {
          "region": {
            "name": "Asia"
          }
        },
        "id": "1518011",
        "licensing": {
          "model": "co-term"
        },
        "management": {
          "details": []
        },
        "name": "hyechoi demo",
        "url": "https://n483.meraki.com/o/kXwqaa/manage/organization/overview"
      }
    ],
    "result": ""
  }
}

PLAY RECAP *****
localhost : ok=2  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

Using Ansible with Cisco Meraki 실습

특정 organization의 network 정보 불러오기

```
Show result] *****
*****
localhost] => {
  "sg": {
    "changed": false,
    "failed": false,
    "meraki_response": [
      {
        "api": {
          "enabled": true
        },
        "cloud": {
          "region": {
            "name": "Asia"
          }
        },
        "id": "1518011",
        "licensing": {
          "model": "co-term"
        },
        "management": {
          "details": []
        },
        "name": "hyechoi's lab",

```

1. 이번 랩에서는 hyechoi's lab organization 사용 예정
orgId = 1518011
2. hyechoi's lab 에 등록되어 있는 network 정보를 불러오기 위한
playbook은 networks.yml
3. **vi networks.yml**로 플레이북 파일 편집
organizationId를 1518011 로 변경 후 :wq로 저장

```
Terminal x +
[1]
[2]
[3]
- hosts: localhost
  gather_facts: false
  tasks:
    - name: Get all networks
      cisco.meraki.networks_info:
        # configTemplateId: string
        # isBoundToConfigTemplate: True
        # tags: []
        # tagsFilterType: string
        # perPage: 0
        # startingAfter: string
        # endingBefore: string
        organizationId: "828099381482762270"
    - name: Get all networks2
      cisco.meraki.networks_clients_info:
        # configTemplateId: string
        # isBoundToConfigTemplate: True
        # tags: []
        # tagsFilterType: string
        # perPage: 0
        # startingAfter: string
        # endingBefore: string
        # organizationId: "828099381482762270"
        networkId: "{{(item.id)}}"
        loop: "{{(result.meraki_response)}}"
        register: result2

"networks.yml" [noel] 50L, 1369B 1,1
```

Using Ansible with Cisco Meraki 실습

특정 organization의 network 정보 불러오기

1. developer:playbooks > ansible-playbook -i hosts networks.yml

PLAY [localhost] *****

TASK [Get all networks] *****

ok: [localhost]

2. TASK [Get all networks2] *****

ok: [localhost] => (item={'id': 'L_834854780923806877', 'organizationId': '1518011', 'name': 'superCAT_MR28', 'productTypes': ['appliance', 'camera', 'cellularGateway', 'sensor', 'switch', 'wireless'], 'timeZone': 'America/Los_Angeles', 'tags': [], 'enrollmentString': None, 'url': 'https://n483.meraki.com/superCAT_MR28-ap/n/bY5LobJh/manage/usage/list', 'notes': '', 'isBoundToConfigTemplate': False})

ok: [localhost] => (item={'id': 'L_834854780923806883', 'organizationId': '1518011', 'name': 'superCAT_MR36', 'productTypes': ['appliance', 'camera', 'cellularGateway', 'sensor', 'switch', 'wireless'], 'timeZone': 'America/Los_Angeles', 'tags': [], 'enrollmentString': None, 'url': 'https://n483.meraki.com/superCAT_MR36-ca/n/U-vgXdJh/manage/usage/list', 'notes': '', 'isBoundToConfigTemplate': False})

ok: [localhost] => (item={'id': 'L_834854780923806893', 'organizationId': '1518011', 'name': 'superCAT_CW9166I', 'productTypes': ['appliance', 'camera', 'cellularGateway', 'sensor', 'switch', 'wireless'], 'timeZone': 'America/Los_Angeles', 'tags': [], 'enrollmentString': None, 'url': 'https://n483.meraki.com/superCAT_CW9166I/n/U2xRdaJh/manage/usage/list', 'notes': None, 'isBoundToConfigTemplate': False})

1. Networks.yml 플레이북 실행
2. hyechoi's lab organization에 등록되어 있는 network 정보가 print 된 것 확인

Using Ansible with Cisco Meraki 실습

특정 organization에 등록된 장비 정보 확인하기

1

```
Terminal x +
--
- hosts: localhost
  gather_facts: false
  tasks:
    - name: Get Device
      cisco.meraki.devices_info:
        meraki suppress_logging: true
        organizationId: "828099381482762270"
        # serial: QBSD-36C3-473D
        # tags: ["recently-added"]
        # productTypes: ["appliance"]
        perPage: 3
        total_pages: -1
        register: result

    - name: Show result
      ansible.builtin.debug:
        msg: "{{ result }}"

~
~
```

organizationId: "1518011"

1. vi devices_info.yml로 플레이북 파일 편집
organizationId를 1518011 로 변경 후 :wq로 저장
2. ansible-playbook -i hosts devices_info.yml로 플레이북 실행

2

```
Terminal x + A- A+
),
{
  "address": "",
  "configurationUpdatedAt": "2023-11-28T06:08:47Z",
  "details": [],
  "firmware": "mt-1-5",
  "lanIp": null,
  "lat": 37.4180951010362,
  "lng": -122.098531723022,
  "mac": "e8:d3:b4:fd:5e:9f",
  "model": "MT15",
  "name": "",
  "networkId": "L_834854780923806883",
  "notes": "",
  "productType": "sensor",
  "sensor": {
    "alertProfileIds": [],
    "alertingOn": [],
    "metrics": [
      "temperature",
      "humidity",
      "eco2",
      "co2",
      "tvoc",
      "pm25",
      "noise",
      "indoorAirQuality"
    ]
  },
  "serial": "Q3CQ-6A5D-8SMR",
  "tags": [],
  "url": "https://n483.meraki.com/superCAT_MR36-en/n/TPqV4aJh/manage/nodes/n
ex_list/247199879224991"
},
```

미션!



이름	네트워크명	Pod #	이름	네트워크명	Pod #	이름	네트워크명	Pod #
A	superCAT_MR28	1	A	superCAT_MR36	1	A	superCAT_CW9166I	1
B	superCAT_MR28	2	B	superCAT_MR36	2	B	superCAT_CW9166I	2
C	superCAT_MR28	3	C	superCAT_MR36	3	C	superCAT_CW9166I	3
D	superCAT_MR28	4	D	superCAT_MR36	4	D	superCAT_CW9166I	4
E	superCAT_MR28	5	E	superCAT_MR36	5	E	superCAT_CW9166I	5
F	superCAT_MR28	6	F	superCAT_MR36	6	F	superCAT_CW9166I	6
G	superCAT_MR28	7	G	superCAT_MR36	7	G	superCAT_CW9166I	7
H	superCAT_MR28	8	H	superCAT_MR36	8	H	superCAT_CW9166I	8
I	superCAT_MR28	9	I	superCAT_MR36	9	I	superCAT_CW9166I	9
J	superCAT_MR28	10	J	superCAT_MR36	10	J	superCAT_CW9166I	10
K	superCAT_MR28	11	K	superCAT_MR36	11	K	superCAT_CW9166I	11

미션#1



목표

- 'hyechoi's lab' organization의 top devices by usage 정보를 확인하여 어느 model의 장비가 가장 사용량이 높은지 확인

주의할 점

- Organization_summary.yml 파일 편집
- Org_id 수정 필요

미션#1 정답



```
Terminal x +
developer:playbooks > cat organization_summary.yml
---
- hosts: localhost
  vars:
    org_id: "1518011"
  gather_facts: false
  tasks:
    - name: Get all organizations summary top appliances by utilization
      cisco.meraki.organizations_summary_top_appliances_by_utilization_info:
        organizationId: "{{ org_id }}"
      register: result

    - name: Show result
      ansible.builtin.debug:
        msg: "{{ result }}"

    - name: Get all organizations summary top clients by usage
      cisco.meraki.organizations_summary_top_clients_by_usage_info:
        organizationId: "{{ org_id }}"
      register: result

    - name: Show result
      ansible.builtin.debug:
        msg: "{{ result }}"
```

Org id 수정

```
TASK [Get all organizations summary top devices by usage] *****
ok: [localhost]

TASK [Show result] *****
ok: [localhost] => {
  "msg": {
    "changed": false,
    "failed": false,
    "meraki_response": [
      {
        "clients": {
          "counts": {
            "total": 1
          }
        },
        "model": "MS130-8P",
        "name": "superCAT_MR36",
        "network": {
          "id": "L_834854788923806883",
          "name": "superCAT_MR36"
        },
        "productType": "switch",
        "serial": "Q4CB-DH9K-N2X2",
        "usage": {
          "percentage": 100.0,
          "total": 387.0546875
        }
      }
    ]
  }
}
```

정답: MS130-8P

정상적으로 플레이북 실행됐을 시 결과 화면

- ansible-playbook -i hosts organization_summary.yml 로 파일 실행
- 정답은 MS130-8P

미션#2



목표

- wifi_deploy_one.yml 파일을 편집
(라인 4줄 수정 필요)
- 'hyechoi's lab' organization의 개인별 할당받은 네트워크에 (예:superCAT_MR28) SSID 생성
- SSID 명은 DevNet_#_영문이름
- #는 본인이 할당받은 pod 번호 사용
(예:내 pod 번호가 3번일 경우, SSID 이름은 DevNet_3_Hyeyeon)
-

주의할 점

- Org ID, Network ID 확인
- ★ yml 파일 내의 Create corporate SSID 아래 number를 할당받은 pod 번호와 동일하게 설정하기
예: number: "3"
- corp_name이 SSID 명

미션#2 정답



```
developer:playbooks > cat wifi_deploy_one.yml
```

```
---
```

```
- hosts: localhost
```

```
vars:
```

```
org_id: "1518011"
```

```
corp_name: "DevNet_15_hyeyeon"
```

```
network_id: "L_834854780923806893"
```

```
gather_facts: false
```

```
tasks:
```

```
- name: Create corporate SSID
```

```
cisco.meraki.networks_wireless_ssids:
```

```
meraki_suppress_logging: true
```

```
state: present
```

```
enabled: true
```

```
name: "{{corp_name}}"
```

```
networkId: "{{ network_id }}"
```

```
number: "0"
```

- wifi_deploy_one.yml 파일 수정 후 저장 필요
- 실행 명령어
Ansible-playbook -i hosts wifi_deploy_one.yml

```
developer:playbooks > ansible-playbook -i hosts wifi_deploy_one.yml
```

```
PLAY [localhost] *****
```

```
TASK [Create corporate SSID] *****
```

```
meraki_compare_equality superCAT_CW9166I - wireless WiFi DevNet_15_hyeyeon
```

```
changed: [localhost]
```

정상적으로 플레이북 실행됐을 시 결과 화면

```
PLAY RECAP *****
```

```
localhost : ok=1 changed=1 unreachable=0 failed=0 skipped=0
```

```
rescued=0 ignored=0
```



Thank you

