

# AWS 활용 IoT

## [4] AWS IoT : 플릿 인덱싱



강사 : 고병화

# Fleet Indexing

# Fleet Indexing : 플릿 인덱싱 소개

Fleet Indexing은 클라우드에서 레지스트리, 새도우 및 연결 데이터를 인덱싱하고 검색할 수 있는 관리형 서비스 이다  
플릿 인덱스가 설정되면 서비스에서 모든 레지스트리 및 새도우 업데이트의 인덱싱을 관리한다.

인기 있는 오픈 소스 검색 엔진인 Apache Lucene을 기반으로 하는 간단한 쿼리 언어를 사용하여 이 데이터를 검색할 수 있다

# Fleet Indexing : 실습 시나리오

방마다 온도 센서가 있는 건물을 관리한다고 가정해 보자  
어떤 방의 온도가  $20^{\circ}$  이상인지 확인하려고 한다  
이는 공기 상태가 제대로 작동하지 않는다는 표시일 수 있기 때문이다

보고된 온도를 장치 그림자에 추가하고 방 번호 속성도 장치에 추가하여  
온도 센서를 시뮬레이션 한다.

실습은 앞의 대량 프로비저닝 실습에서 생성한 장치를 사용한다



# Device Registry/Shadow/Connectivity Indexing

## Lucene-index queries

Get me all the widgets prior to version 6 that support more than 3 colors

```
"attributes.version < 6 AND attributes.colors > 3"
```

Find all rooms where the temperature is above 20°

```
"shadow.reported.temperature>20 AND attributes.room_number:*"
```

Find all my lightbulbs that are currently on and no one present in the warehouse

```
"shadow.reported.illumination_state = on AND shadow.reported.occupant present  
= no AND attributes.building_type = warehouse"
```

Find all connected devices

```
"connectivity.connected:true"
```

# Fleet Indexing : 플릿 인덱싱 준비

Cloud9의 터미널에서 아래 명령을 차례로 수행한다

# 레지스트리 인덱싱을 활성화

```
aws iot update-indexing-configuration --thing-indexing-configuration  
thingIndexingMode=REGISTRY_AND_SHADOW,thingConnectivityIndexingM  
ode=STATUS
```

# 대량 프로비저닝에서 만들어진 경로로 이동한다

```
cd ~/provisioning/bulky-2022-09-12_05-17-59
```

# 파이썬 코드를 실행시킨다

```
fleet-indexing.py -b bulky
```

# Fleet Indexing : fleet-indexing.py 소스 동작

```
temp = random.randint(15,30)
```

```
return {
```

```
    "state": {
```

```
        "reported" : {
```

```
            "temperature" : temp
```

```
        }
```

```
    }
```

```
...
```

```
attributePayload={
```

```
    'attributes': {
```

```
        'building': building_names[random.randint(0, len(building_names) - 1)],
```

```
        'room_number': str(ROOM_NUMBER)    },
```

“bulky” 로 시작하는 등록된 모든 디바이스의 디바이스 새도우에 "temperature" 를 추가하고 속성에 “building”과 “room\_number”를 추가한다

# Fleet Indexing : 인덱스 쿼리

```
aws iot search-index --query-string "thingName:bulky*"
```

# 방 번호가 있고 온도가 20° 이상인 모든 장치를 레지스트리에서 찾는다

```
aws iot search-index \  
--query-string "shadow.reported.temperature>20 AND  
attributes.room_number:*
```

# XX represents the number for the device that you used, e.g. 10

```
aws iot search-index --index-name "AWS_Things" --query-string  
"thingName:bulky10"
```



# Fleet Indexing : 인덱스 쿼리

AWS IoT 콘솔의 [사물]에서도 bulky1의 속성을 볼 수 있다

속성	인증서	사물 그룹	디바이스 새도우	상호 작용	활동	작업	경보	Defender 지표
<b>속성 (4) 정보</b>								
속성은 검색 가능하거나 검색 불가능할 수 있는 키-값 페어입니다. 검색 가능한 속성은 플릿 인덱싱을 사용하지 않고 사물 목록을 필터링하는 데 사용할 수 있습니다. 검색 불가능한 속성은 사물을 찾는 데 사용할 수 있지만 플릿 인덱싱이 활성화된 경우에만 사용할 수 있습니다.								
키			값					
building			Doppler					
room_number			100					
serialNumber			1					
version			v1					

# Fleet Indexing : 인덱스 쿼리

AWS IoT 콘솔의 [사물]의  
[디바이스 새도우 문서]에서도  
bulky1의 새도우 상태 값을 볼 수  
있다

디바이스 새도우 문서

MQTT 주제

디바이스 새도우 문서 정보

디바이스 새도우 문서에는 디바이스 상태의 보고된 값

디바이스 새도우 상태

```
{
  "state": {
    "reported": {
      "temperature": 28
    }
  }
}
```

디바이스 새도우 메타데이터

```
{
  "metadata": {
    "reported": {
      "temperature": {
        "timestamp": 1663046414
      }
    }
  }
}
```

# Fleet Indexing : 연결(Connectivity) 인덱싱

# Cloud9에서 다음을 실행하여 bulky1 을 연결 상태로 만든다

cd ~/provisioning/*bulky-YYYY-mm-dd\_H-M-S* (→생성된 경로로 이동)

```
THING_NAME=bulky1
```

```
mosquitto_sub --cafile ~/root.ca.bundle.pem \  
--cert $THING_NAME.crt --key $THING_NAME.key \  
-h $IOT_ENDPOINT -p 8883 -q 0 -t iot/ws \  
-i $THING_NAME --tls-version tlsv1.2 -d
```



# Fleet Indexing : 연결(Connectivity) 인덱싱

아래 명령을 실행하면 “connected” 값 “true”인 bulky1 만 출력된다

**aws iot search-index --index-name "AWS\_Things" --query-string  
"connectivity.connected:true" → 연결된 모든 장치 찾기 쿼리**

```
aicore0427:~ $ aws iot search-index --index-name "AWS_Things" --query-string "connectivity.connected:true"
{
  "things": [
    {
      "thingName": "bulky1",
      "thingId": "098db395-dbc4-45ae-8302-aff684727a86",
      "thingTypeName": "bulk-type",
      "thingGroupNames": [
        "bulk-group"
      ],
      "attributes": {
        "building": "Doppler",
        "room_number": "100",
        "serialNumber": "1",
        "version": "v1"
      },
      "shadow": "{\"reported\":{\"temperature\":28},\"metadata\":{\"reported\":{\"temperature\":{\"timestamp\"":
      "connectivity": {
        "connected": true,
        "timestamp": 1663083054757
      }
    }
  ]
}
```

# Fleet Indexing : 연결(Connectivity) 인덱싱

아래 명령을 실행하면 “connected” 값 “false”인 장치가 모두 출력된다

**aws iot search-index --index-name "AWS\_Things" --query-string  
"connectivity.connected:false" → 연결이 끊긴 모든 장치 찾기 쿼리**

```
{
  "thingName": "bulky3",
  "thingId": "24f86b13-56e1-425d-86c3-3e961a5450cf",
  "thingTypeName": "bulk-type",
  "thingGroupNames": [
    "bulk-group"
  ],
  "attributes": {
    "building": "Kumo",
    "room_number": "101",
    "serialNumber": "3",
    "version": "v1"
  },
  "shadow": "{\"reported\":{\"temperature\":15},\"metadata\":{\"reported\":{\"temperature\":{\"timestamp\":1663046414}}},\"hasDelta\":false,\"version\":2}",
  "connectivity": {
    "connected": false,
    "timestamp": 1662961434509,
    "disconnectReason": "CLIENT_INITIATED_DISCONNECT"
  }
},
{
  "thingName": "bulky10",
  "thingId": "39219851-ae7d-453d-b51d-f870ce6768ea",
  "thingTypeName": "bulk-type",
  "thingGroupNames": [
    "bulk-group"
  ],
  "attributes": {
    "building": "Kumo",
    "room_number": "101",
    "serialNumber": "3",
    "version": "v1"
  },
  "shadow": "{\"reported\":{\"temperature\":15},\"metadata\":{\"reported\":{\"temperature\":{\"timestamp\":1663046414}}},\"hasDelta\":false,\"version\":2}",
  "connectivity": {
    "connected": false,
    "timestamp": 1662961434509,
    "disconnectReason": "CLIENT_INITIATED_DISCONNECT"
  }
}
```

# Fleet Indexing : 통계(Statistics) 얻기

get-statistics (GetStatistics API)를 사용하면 통계(aggregate) 데이터에 대한 인덱스를 검색할 수 있다 (count, average, sum, minimum, maximum 값)

# AWS IoT에 연결된 디바이스 수 가져오기

```
aws iot get-statistics --query-string "connectivity.connected:true"
```

# 건물이름이 “Doppler”인 디바이스 수 가져오기

```
aws iot get-statistics --query-string "attributes.building:Doppler"
```

# Fleet Indexing : 통계(Statistics) 얻기

[ 연결된 디바이스 수 ]

```
{
  "statistics": {
    "count": 1,
    "average": 4.0,
    "sum": 4.0,
    "minimum": 4.0,
    "maximum": 4.0,
    "sumOfSquares": 16.0,
    "variance": 0.0,
    "stdDeviation": 0.0
  }
}
```

[ 건물명이 "Doppler"인 디바이스 수 ]

```
{
  "statistics": {
    "count": 5,
    "average": 4.0,
    "sum": 20.0,
    "minimum": 4.0,
    "maximum": 4.0,
    "sumOfSquares": 80.0,
    "variance": 0.0,
    "stdDeviation": 0.0
  }
}
```



# Fleet Indexing : Pi3 보드의 연결 상태 얻기

AWS IoT 클라우드에서 Pi3 보드의 연결/해제 상태를 플릿 인덱싱으로 검색하여 보자

먼저 Pi3 보드에서 접속 프로그램을 실행시킨다

`./run_shadow_led.sh`

(연결 이벤트를 확인을 위해 다음 페이지의 주제로 구독을 먼저 수행해 놓는다)

```
tea@planx ~/aws$ ./run_shadow_led.sh
Connecting to a1mp2lfc29w0b5-ats.iot.us-east-1.amazonaws.com with client ID 'Raspi_Shadow_Led'...
Connected!
Subscribing to Update responses...
shadow_thing_name RaspberryPi
Subscribing to Get responses...
Subscribing to Delta events...
Requesting current shadow state...
Launching thread to read user input...
Finished getting initial shadow state.
Shadow contains reported value 'off'.
Enter desired value:
```

# Fleet Indexing : Pi3 보드의 연결 상태 얻기

\$aws/events/presence/connected/Raspi\_Shadow\_Led를 주제로 구독한다

**주제 구독** | 주제 게시

주제 필터 | 정보  
주제 필터는 구독할 주제를 설명합니다. 주제 필터에는 MQTT 와일드카드 문자

▶ 추가 구성

**구독**

**\$aws/events/presence/connected/Raspi\_Shadow\_Led**

▼ \$aws/events/presence/connected/Raspi\_Shadow\_Led

```
{
  "clientId": "Raspi_Shadow_Led",
  "timestamp": 1663081052973,
  "eventType": "connected",
  "sessionIdentifier": "7dc2b8a5-b975-40e4-838e-50c94ba897f0",
  "principalIdentifier": "20758280db74ba7012ad509cde47043e24e0daaa1",
  "ipAddress": "123.213.208.180",
  "versionNumber": 2
}
```

# Fleet Indexing : Pi3 보드의 연결 상태 얻기

# Cloud9 에서 Raspberry Pi3 device shadow 와 인덱스 검색

```
aws iot search-index --query-string "thingName:RaspberryPi"
```

```
aws iot search-index --query-string "shadow.reported.led:on"
```

```
aws iot search-index --query-string "shadow.reported.led:*"
```

```
aws iot search-index --index-name "AWS_Things" --query-string  
"connectivity.connected:true"
```

```
aws iot search-index --query-string "thingName:RaspberryPi"
```

```
"thingName": "RaspberryPi",  
"thingId": "b27d6d3a-a1fb-4be5-b79f-3f25a077816f",  
"shadow": "{\n  \"desired\": {\n    \"welcome\": \"aws-iot\",  
    \"led\": {\n      \"timestamp\": 1663085900\n    }\n  },  
  \"reported\": {\n    \"led\": {\n      \"timestamp\": 1663085924\n    }\n  }\n}",  
"connectivity": {\n  "connected": false,
```

Pi3보드가 접속 되었는데도  
connected는 false로 나온다

Why?

# Fleet Indexing : Pi3 보드의 연결 상태 얻기

[Answer] : connectivity status data 는 **client ID**와 **thing name**이 일치 할 때만 인덱스 되어진다 <https://stackoverflow.com/questions/65875369/how-does-an-aws-iot-device-show-as-connected>

보드의 실행 스크립트를 아래와 같이 수정하여 새로 저장한다

[run\_shadow\_led.sh : 변경 전]

```
python3 aws-iot-device-sdk-python-v2/samples/shadow_led.py --endpoint  
a1mp21fc29w0b5-ats.iot.us-east-1.amazonaws.com --ca_file root-CA.crt --cert  
RaspberryPi.cert.pem --key RaspberryPi.private.key --client_id Raspi_Shadow_Led  
--thing_name RaspberryPi --shadow_property led
```

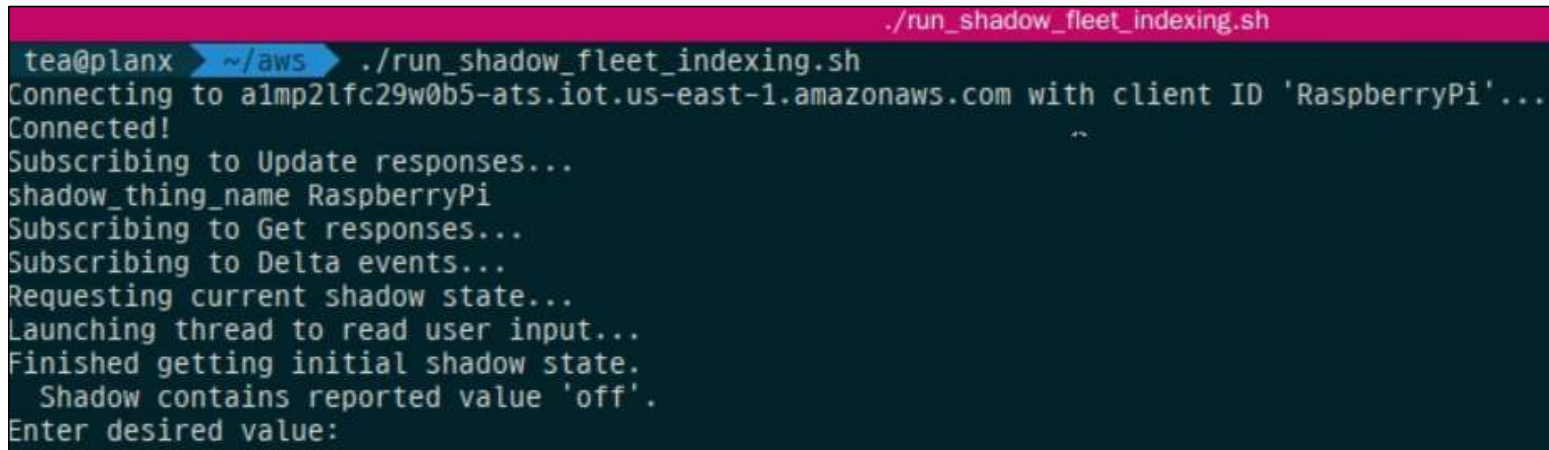
[run\_shadow\_fleet\_indexing.sh : 변경 후]

```
python3 aws-iot-device-sdk-python-v2/samples/shadow_led.py --endpoint  
a1mp21fc29w0b5-ats.iot.us-east-1.amazonaws.com --ca_file root-CA.crt --cert  
RaspberryPi.cert.pem --key RaspberryPi.private.key --client_id RaspberryPi --  
thing_name RaspberryPi --shadow_property led
```

# Fleet Indexing : Pi3 보드의 연결 상태 얻기

Pi3 보드에서 수정된 스크립트 파일로 접속 프로그램을 실행시킨다

**`./run_shadow_fleet_indexing .sh`**



```
./run_shadow_fleet_indexing.sh
tea@planx ~/aws$ ./run_shadow_fleet_indexing.sh
Connecting to a1mp2lfc29w0b5-ats.iot.us-east-1.amazonaws.com with client ID 'RaspberryPi'...
Connected!
Subscribing to Update responses...
shadow_thing_name RaspberryPi
Subscribing to Get responses...
Subscribing to Delta events...
Requesting current shadow state...
Launching thread to read user input...
Finished getting initial shadow state.
Shadow contains reported value 'off'.
Enter desired value:
```

Cloud9에서 검색 명령을 수행한다

**`aws iot search-index --index-name "AWS_Things" --query-string "connectivity.connected:true"`**

# Fleet Indexing : Pi3 보드의 연결 상태 얻기

Cloud9에서 다시 검색 명령을 실행하면 connected가 true로 변경 된 걸 볼 수 있다

```
aicore0427:~ $ aws iot search-index --query-string "thingName:RaspberryPi"
{
  "things": [
    {
      "thingName": "RaspberryPi",
      "thingId": "b27d6d3a-a1fb-4be5-b79f-3f25a077816f",
      "shadow": "{\"desired\":{\"welcome\":\"aws-iot\",\"led\":\"off\"},\"reported\":{\"welcome\":\"aws-iot\",\"led\":\"off\"},\"metadata\":{\"timestamp\":1663085900},\"led\":{\"timestamp\":1663085924}},\"reported\":{\"welcome\":{\"timestamp\":1663085900},\"led\":{\"timestamp\":1663085924}},\"version\":196}\",
      "connectivity": {
        "connected": true,
        "timestamp": 1663087494776
      }
    }
  ]
}
```





# Fleet Indexing : API 사용 쿼리

Pi3보드와 bulky1을 모두 연결 상태로 하고 Cloud9에서 배포된 fleet\_indexing\_api\_call.py를 실행하면 접속된 기기를 알 수 있다

|\_]

```
Connected Devices : 2
connected device [ 0 ] : bulky1
connected device [ 1 ] : RaspberryPi
```

```
-----
response:
```

```
{'ResponseMetadata': {'RequestId': 'b0c03aac-a55c-4771-84ec-6798e3f7c036', 'HTTPStatusCode': 200, 'HTTPHeaders': {'date': 'Wed, 14 Nov 2018 12:12:12 GMT', 'application/json', 'content-length': '1124', 'connection': 'keep-alive', 'x-amzn-requestid': 'b0c03aac-a55c-4771-84ec-6798e3f7c036'}, 'RetryAttempts': 0}, 'ingName': 'bulky1', 'thingId': '098db395-dbc4-45ae-8302-aff684727a86', 'thingTypeName': 'bulk-type', 'thingGroupNames': ['bulk-type', 'room_number': '100', 'serialNumber': '1', 'version': 'v1'}, 'shadow': '{"reported":{"temperature":28},"metadata":{"reported":{"timestamp":1663123964745},"hasDelta":false,"version":2}}', 'connectivity': {'connected': True, 'timestamp': 1663123964745}}, {'thingName': 'RaspberryPi', 'thingId': '816f', 'shadow': '{"desired":{"welcome":"aws-iot","led":"off"},"reported":{"welcome":"aws-iot","led":"off"},"metadata":{"desired":{"timestamp":1663085924},"reported":{"welcome":{"timestamp":1663085900},"led":{"timestamp":1663085924}}},"hasDelta":false,"version":1}}', 'connectivity': {'connected': True, 'timestamp': 1663111606480}}}]}
```



The End