

AWS 활용 IoT

[8] AWS IoT : SNS Notification



강사 : 고병화

AWS SNS Notification

SNS Notification : AWS IoT SNS

이 실습에서는 온도가 규칙에 설정된 값을 초과할 때마다 기후 센서에서 Amazon SNS 주제의 모든 구독자에게 메시지 데이터를 전송하는 규칙을 생성합니다. 이 규칙은 보고된 온도가 규칙에 의해 설정된 값을 초과할 때를 감지하고 디바이스 ID, 보고된 온도 및 초과된 온도 제한만 포함하는 새 메시지 페이로드를 생성합니다. 이 규칙은 새 메시지 페이로드를 JSON 문서로 SNS 주제에 전송하여 모든 구독자에게 SNS 주제를 알립니다.

https://docs.aws.amazon.com/ko_kr/iot/latest/developerguide/iot-sns-rule.html

SNS Notification : AWS IoT SNS

1단계: SMS 문자 메시지를 전송하는 Amazon SNS 주제 생성
SMS 문자 메시지를 전송하는 Amazon SNS 주제를 생성하려면

1.Amazon SNS 주제를 생성합니다.

- 1.[Amazon SNS 콘솔](#)에 로그인합니다.
- 2.왼쪽 탐색 창에서 **주제**를 선택합니다.
- 3.**주제(Topics)** 페이지에서 **새 주제 생성(Create new topic)**을 선택합니다.
- 4.**세부 정보**에서 **표준** 유형을 선택합니다. 기본적으로 콘솔은 FIFO 주제를 만듭니다.
- 5.**이름**에 SNS 주제 이름을 입력합니다. 이 자습서에서는 **high_temp_notice**을 입력합니다.
- 6.페이지 끝으로 스크롤하고 **주제 생성**을 선택합니다.
콘솔에 새 주제의 **세부 정보** 페이지가 열립니다.

SNS Notification : AWS IoT SNS

2. Amazon SNS 구독을 생성합니다.

1. **high_temp_notice** 주제 세부 정보 페이지에서 **구독 생성(Create subscription)**을 선택합니다.

2. 구독 생성에 있는 세부 정보 섹션의 **프로토콜** 목록에서 **SMS**를 선택합니다.

3. **엔드포인트**에 문자 메시지를 받을 수 있는 전화 번호를 입력합니다. +로 시작하고 국가 및 지역 코드를 포함하고 다른 구두점은 포함하지 않도록 입력해야 합니다.

4. 구독 생성을 선택합니다.

SNS Notification : AWS IoT SNS

3. Amazon SNS 알림을 테스트합니다.

1. [Amazon SNS 콘솔](#)의 왼쪽 탐색 창에서 **주제**를 선택합니다.
2. 주제의 세부 정보 페이지를 열려면 **주제**의 주제 목록에서 **high_temp_notice**를 선택한다.
3. **주제에 메시지 게시(Publish message to topic)** 페이지를 열려면 **high_temp_notice** 세부 정보 페이지에서 **메시지 게시(Publish message)**를 선택합니다.
4. **주제에 메시지 게시(Publish message to topic)**에 있는 **메시지 본문** 섹션에서 **엔드포인트로 전송할 메시지 본문(Message body to send to the endpoint)**에 짧은 메시지를 입력한다.
5. 페이지의 하단으로 스크롤하고 **메시지 게시(Publish message)**를 선택한다.
6. 구독을 만들 때 이전에 사용한 번호가 있는 전화에서 메시지가 수신되었는지 확인하세요.

SNS Notification : AWS IoT SNS

2단계: 텍스트 메시지를 전송하는 AWS IoT 규칙 생성

이 실습에서 만들 AWS IoT 규칙은 device/*device_id*/data MQTT 주제를 구독합니다. 여기서 *device_id*는 메시지를 전송한 디바이스의 ID입니다. 이러한 주제는 주제 필터에서 device/+ /data로 설명됩니다. 여기서 +는 두 개의 슬래시 문자 사이의 모든 문자열과 일치하는 와일드카드 문자입니다. 이 규칙은 메시지 페이로드의 temperature 필드 값도 테스트합니다.

규칙이 일치하는 주제로부터 메시지를 수신하면 주제 이름에서 *device_id*, 메시지 페이로드에서 temperature 값을 가져와 테스트 중인 제한에 대한 상수 값을 추가하고 이러한 값을 JSON 문서로 Amazon SNS 알림 주제에 전송합니다. 예를 들어, 기상 센서 디바이스 번호 32의 MQTT 메시지는 device/32/data 주제를 사용하고 다음과 같은 메시지 페이로드가 있습니다.

SNS Notification : AWS IoT SNS

```
{  
  "temperature": 38,  
  "humidity": 80,  
  "barometer": 1013,  
  "wind": {  
    "velocity": 22,  
    "bearing": 255  
  }  
}
```


SNS Notification : AWS IoT SNS

1. [AWS IoT 콘솔의 규칙 허브](#)를 엽니다.
2. 이것이 첫 번째 규칙인 경우 **생성** 또는 **규칙 생성**을 선택합니다.
3. **규칙 생성(Create a rule)**에서:
 1. **이름(Name)**에 **temp_limit_notify**를 입력합니다.
규칙 이름은 AWS 계정 및 리전 내에서 고유해야 하며 공백을 포함할 수 없다. 이 이름에 밑줄 문자를 사용하여 규칙 이름의 단어를 구분했습니다.
 2. **설명**에서 규칙을 설명합니다.
4. **규칙 생성의 규칙 쿼리 문**에서:
 1. **SQL 버전 사용**에서 **2016-03-23**을 선택합니다.
 2. **규칙 쿼리 문** 편집 상자에 쿼리 문을 입력합니다.

SNS Notification : AWS IoT SNS

```
SELECT topic(2) as device_id,  
       temperature as reported_temperature,  
       28 as max_temperature  
FROM 'device/+/data'  
WHERE temperature > 28
```

SNS Notification : AWS IoT SNS

- 1.이 규칙에 대한 규칙 작업 목록을 열려면 **하나 이상의 작업 설정(Set one or more actions)**에서 **작업 추가(Add action)**를 선택합니다.
- 2.**Select an action(작업 선택)**에서 **SNS 푸시 알림으로 메시지 전송(Send a message as an SNS push notification)**을 선택합니다.
- 3.작업 목록 하단에서 선택한 작업의 구성 페이지를 열려면 **구성 작업을 선택합니다.**
- 4.구성 작업에서:
 - 1.**SNS 대상에서 선택(Select)**을 클릭하고 **high_temp_notice**라는 이름의 SNS 주제를 찾은 다음 **선택(Select)**을 클릭합니다.
 - 2.**메시지 형식(Message format)**에서 **RAW**를 선택합니다.
 - 3.**AWS IoT에 이 작업을 수행할 액세스 권한을 부여하는 역할 선택 또는 생성에서 역할 생성**을 선택합니다.
 - 4.**새 역할 생성**에서 이름에 새 역할의 고유 이름을 입력합니다. 본 자습서에서는 **sns_rule_role**를 사용합니다.
 - 5.**역할 생성(Create role)**을 선택합니다.

SNS Notification : AWS IoT SNS

이 실습을 반복하거나 기존 역할을 재사용하는 경우 계속 진행하기 전에 **역할 업데이트**를 선택합니다. 그러면 SNS 대상에서 작동하도록 역할의 정책 문서가 업데이트됩니다.

5.작업 추가를 선택하여 **규칙 생성** 페이지로 이동합니다.

새 작업의 타일에서 **SNS 푸시 알림으로 메시지 전송하기(Send a message as an SNS push notification)** 아래에서 규칙이 호출할 SNS 주제를 볼 수 있습니다.

이 규칙 작업은 이 규칙에 추가할 유일한 규칙 작업입니다.

6.규칙을 생성하고 이 단계를 완료하려면 **규칙 생성(Create a rule)**에서 하단으로 스크롤하여 **규칙 생성(Create rule)**을 선택합니다.

SNS Notification : AWS IoT SNS

3단계: AWS IoT 규칙 및 Amazon SNS 알림 테스트

새 규칙을 테스트하려면 MQTT 클라이언트를 사용하여 이 규칙에서 사용하는 MQTT 메시지를 게시하고 구독합니다.

새 창에서 [AWS IoT 콘솔의 MQTT 클라이언트](#)를 엽니다. 이렇게 하면 MQTT 클라이언트의 구성을 그대로 유지하면서 규칙을 편집할 수 있습니다. 콘솔의 다른 페이지로 이동하기 위해 MQTT 클라이언트를 나가면 구독이나 메시지 로그가 유지되지 않습니다.

MQTT 클라이언트를 사용하여 규칙을 테스트하려면

1. [AWS IoT 콘솔의 MQTT 클라이언트](#)에서 입력 주제(이 경우 device/+data)를 구독합니다.

1. MQTT 클라이언트에서 **구독** 아래에서 **주제 구독**을 선택합니다.
2. **구독 주제**에 입력 주제 필터 **device/+data**의 주제를 입력합니다.
3. 나머지 필드는 기본 설정을 유지합니다.
4. **주제 구독**을 선택합니다.

구독 열의 **주제 게시** 아래에 **device/+data**가 나타납니다.

SNS Notification : AWS IoT SNS

2. 특정 디바이스 ID **device/32/data**로 입력 주제에 메시지를 게시합니다.
와일드카드 문자가 포함된 MQTT 항목에는 게시할 수 없습니다.

1. MQTT 클라이언트의 **구독** 아래에서 **주제 게시**를 선택합니다.

2. 게시 필드에 입력 주제 이름 **device/32/data**를 입력합니다.

3. 여기에 표시된 샘플 데이터를 복사하고 주제 이름 아래의 편집 상자에 샘플 데이터를 붙여 넣습니다.

```
{  
  "temperature": 38,  
  "humidity": 80,  
  "barometer": 1013,  
  "wind": {  
    "velocity": 22,  
    "bearing": 255  
  }  
}
```

주제에 게시를 선택하여 MQTT 메시지를 게시합니다.

SNS Notification : AWS IoT SNS

3. 문자 메시지가 전송되었는지 확인합니다.

1.MQTT 클라이언트의 **구독** 아래에서 이전에 구독한 주제 옆에 녹색 점이 있습니다.녹색 점은 마지막으로 메시지를 보았을 때 하나 이상의 새 메시지가 수신되었음을 나타냅니다.

2.**구독** 아래에서 **device/+data**를 선택하여 메시지 페이로드가 방금 게시한 것과 일치하고 다음과 같이 표시되는지 확인하세요.

```
{  
  "temperature": 38,  
  "humidity": 80,  
  "barometer": 1013,  
  "wind": {  
    "velocity": 22,  
    "bearing": 255  
  }  
}
```

SNS Notification : AWS IoT SNS

SNS 주제를 구독하는 데 사용한 전화를 확인하고 메시지 페이로드의 내용이 다음과 같이 표시되는지 확인합니다.

```
{"device_id":"32","reported_temperature":38,"max_temperature":30}
```

device_id 값은 따옴표로 묶인 문자열이고 temperature 값은 숫자여야 합니다. 이는 [topic\(\)](#) 함수가 입력 메시지의 주제 이름에서 문자열을 추출하는 반면 temperature 값은 입력 메시지의 페이로드에서 숫자 값을 사용하기 때문입니다.

device_id 값을 숫자 값으로 만들려면 규칙 쿼리 문에서 topic(2)를 다음으로 바꿉니다.

```
cast(topic(2) AS DECIMAL)
```

주제의 해당 부분에 숫자만 포함된 경우에만 topic(2) 값을 숫자 DECIMAL 값으로 캐스팅할 수 있다는 점을 유의하세요.

SNS Notification : AWS IoT SNS

4. 온도가 한계를 초과하지 않는 MQTT 메시지를 전송해보세요.

1. MQTT 클라이언트의 **구독** 아래에서 **주제 게시**를 선택합니다.

2. 게시 필드에 입력 주제 이름 **device/33/data**를 입력합니다.

3. 여기에 표시된 샘플 데이터를 복사하고 주제 이름 아래의 편집 상자에 샘플 데이터를 붙여 넣습니다.

```
{  
  "temperature": 28,  
  "humidity": 80,  
  "barometer": 1013,  
  "wind": {  
    "velocity": 22,  
    "bearing": 255  
  }  
}
```

SNS Notification : AWS IoT SNS

4.MQTT 메시지를 전송하려면 주제 게시를 선택합니다.

device/+/data 구독에서 전송한 메시지가 표시되어야 합니다. 그러나 온도 값이 규칙 쿼리 문에서 최대 온도보다 낮기 때문에 문자 메시지를 받지 않아야 합니다.

올바른 동작이 표시되지 않으면 문제 해결 팁을 확인합니다.

SNS Notification : AWS IoT SNS

IoT Core 콘솔의 [보안]→[정책]에 가서 RaspberryPi-Policy의 정책을 아래와 같이 두 줄을 17번,30번에 각각 추가한다

"arn:aws:iot:us-east-1:844311781633:topic/device/1/data"

"arn:aws:iot:us-east-1:844311781633:topicfilter/device/1/data"

```
15      "arn:aws:iot:us-east-1:844311781633:topic/$aws/things/RaspberryPi/jobs/*",  
16      "arn:aws:iot:us-east-1:844311781633:topic/sys/info",  
17      "arn:aws:iot:us-east-1:844311781633:topic/device/1/data"  
18  ]
```

```
28      "arn:aws:iot:us-east-1:844311781633:topicfilter/$aws/things/RaspberryPi/jobs/*",  
29      "arn:aws:iot:us-east-1:844311781633:topicfilter/sys/info",  
30      "arn:aws:iot:us-east-1:844311781633:topicfilter/device/1/data"  
31  ]
```

SNS Notification : AWS IoT SNS

IoT Core 콘솔의 [MQTT 테스트 클라이언트]에 가서 "device/1/data"를 주제로 구독한다 다음페이지의 pi3보드에서 메시지를 보내오면 아래와 같은 메시지를 볼 수 있을 것이다

주제 구독

주제

주제 필터



정보

주제 필터는 구독할 주제를

device/1/data

▶ 추가 구성

구독

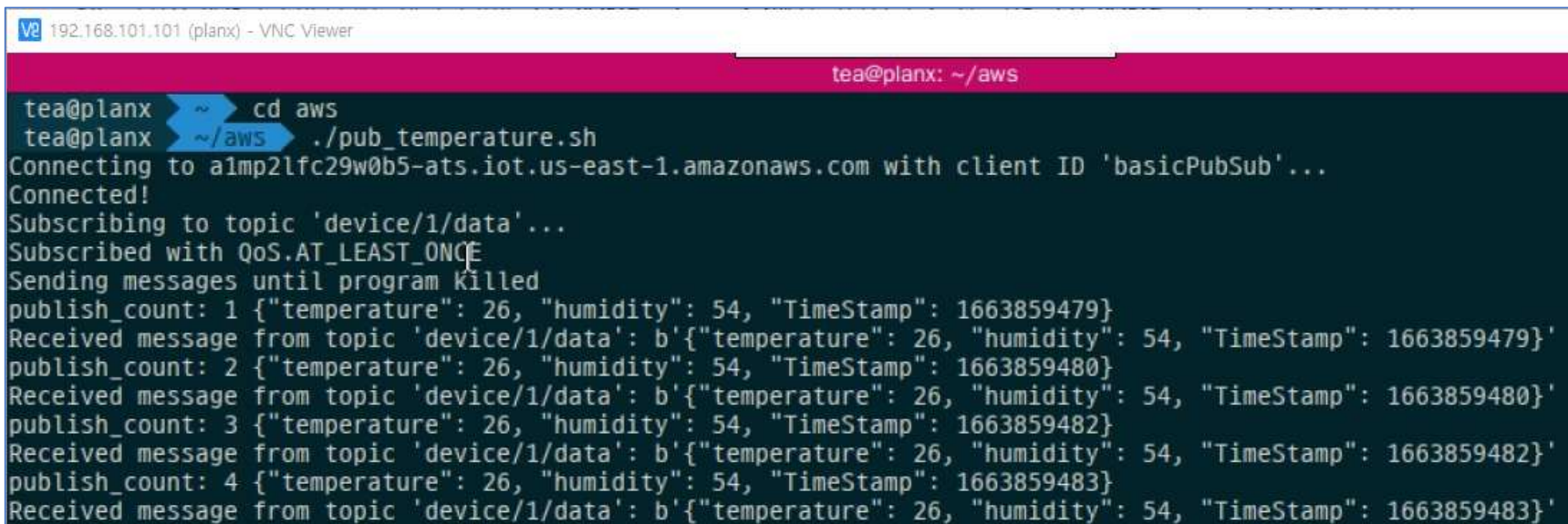
구독	device/1/data
device/1/data  	<div>▼ device/1/data</div> <div><pre>{ "temperature": 27, "humidity": 53, "TimeStamp": 1663860099 }</pre></div>

SNS Notification : Pi3 실습 보드 사용

Pi3 보드에 배포된 파일 두개를 다운 로드하고 아래 명령을 수행한다(pub_temperature.sh 와 pub_temperature.py 파일)

```
cd aws
```

```
./pub_temperature.sh
```



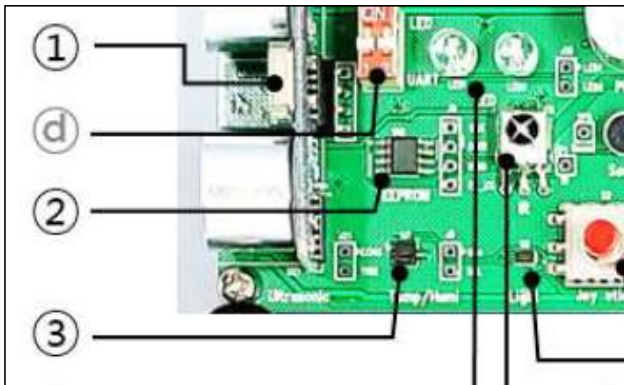
```
V2 192.168.101.101 (planx) - VNC Viewer
tea@planx: ~/aws
tea@planx ~ cd aws
tea@planx ~/aws ./pub_temperature.sh
Connecting to a1mp2lfc29w0b5-ats.iot.us-east-1.amazonaws.com with client ID 'basicPubSub'...
Connected!
Subscribing to topic 'device/1/data'...
Subscribed with QoS.AT_LEAST_ONCE
Sending messages until program Killed
publish_count: 1 {"temperature": 26, "humidity": 54, "TimeStamp": 1663859479}
Received message from topic 'device/1/data': b'{"temperature": 26, "humidity": 54, "TimeStamp": 1663859479}'
publish_count: 2 {"temperature": 26, "humidity": 54, "TimeStamp": 1663859480}
Received message from topic 'device/1/data': b'{"temperature": 26, "humidity": 54, "TimeStamp": 1663859480}'
publish_count: 3 {"temperature": 26, "humidity": 54, "TimeStamp": 1663859482}
Received message from topic 'device/1/data': b'{"temperature": 26, "humidity": 54, "TimeStamp": 1663859482}'
publish_count: 4 {"temperature": 26, "humidity": 54, "TimeStamp": 1663859483}
Received message from topic 'device/1/data': b'{"temperature": 26, "humidity": 54, "TimeStamp": 1663859483}'
```

SNS Notification : Pi3 실습 보드 사용

Pi3 보드의 온/습도 센서에 따뜻한 손가락을 가져다 살짝 누르면 (물기가 없도록 주의) 체온으로 온도가 상승하여 28도를 넘게 할 수 있을 것이다

28도를 넘었기 때문에 "temp_limit_notify" 규칙은 AWS SNS에 통지를 하고 SNS는 등록된 휴대폰으로 SMS문자를 발송한다 (이메일 사용도가능)

아래에서 ③번이 온/습도 센서



device/1/data

▼ device/1/data

```
{
  "temperature": 31,
  "humidity": 77,
  "TimeStamp": 1663860455
}
```

SNS Notification : Pi3 실습 보드 사용

CloudWatch의 로그 그룹 [AWSIoTLogsV2](#)에 로그 스트림중에 아래와 같은 내용이 들어 있는 것을 확인 할 수 있다

```
2022-09-23T00:27:36.092+09:00      {"timestamp":"2022-09-22 15:27:36.092","logLevel":"INFO"}
{
  "timestamp": "2022-09-22 15:27:36.092",
  "logLevel": "INFO",
  "traceId": "71ae82e6-aec6-4b78-218c-7f0d6d104c69",
  "accountId": "844311781633",
  "status": "Success",
  "eventType": "RuleExecution",
  "clientId": "basicPubSub",
  "topicName": "device/1/data",
  "ruleName": "temp_limit_notify",
  "ruleAction": "SNSAction",
  "resources": {
    "TargetArn": "arn:aws:sns:us-east-1:844311781633:high_temp_notice",
    "SNSMessageId": "c6522d1c-30e1-502f-bf7b-76bab1c96729"
  },
  "principalId": "20758280db74ba7012ad509cde47043e24e0daaa1d84aec9b8ad2ac5e03ad766"
}
```


SNS Notification : AWS IoT SNS

AWS SNS의 SMS 최대 할당량으로 문자 발송이 안 되는 경우 콘솔의 Amazon SNS의 [Mobile]의 [문자 메시지(SMS)]의 [전송 상태 로그]에서 아래와 같은 오류 메시지를 확인 해볼 수 있다. 미리 [문자 메시지(SMS)]의 [문자 메시지 기본 설정 편집]으로 들어가서 [전송 상태 로깅]을 설정해 주어야만 볼 수 있다.

SMS 전송 로그

```
@message {"notification":{"messageId":"41335490-26d0-5c38-86ae-0f36a7f69a6e","timestamp":"2022-09-22 15:27:34.59"},"deli
@timestamp 1663860476433
delivery.destination +8201041572936
delivery.dwellTimeMs 69
delivery.providerResponse No quota left for account
delivery.smsType Promotional
notification.messageId 41335490-26d0-5c38-86ae-0f36a7f69a6e
notification.timestamp 2022-09-22 15:27:34.59
status FAILURE
```


▶ 2	2022-09-22T15:27:54.070Z	FAILURE	c6522d1c-30e1-502f-bf7b-76bab1c96729	+8201041572936	No quota left for account
-----	--------------------------	---------	--------------------------------------	----------------	---------------------------

SNS Notification : AWS IoT SNS

<https://support.console.aws.amazon.com/support/home#/case/create?issueType=service-limit-increase&limitType=service-code-sns-text-messaging>

에서 Quota 증가 요청을 작성하여 보낼 수 있다

Requests

 To request additional limit increases for the same limit type, choose **Add another request**.

Request 1

Region

US East (Northern Virginia) ▼

Resource Type

General Limits ▼

Limit

Select or search ▼

Case description

Use case description

Do not share any sensitive information in case correspondences, such as account IDs or other identifying information.

Please increase my quota...

The End