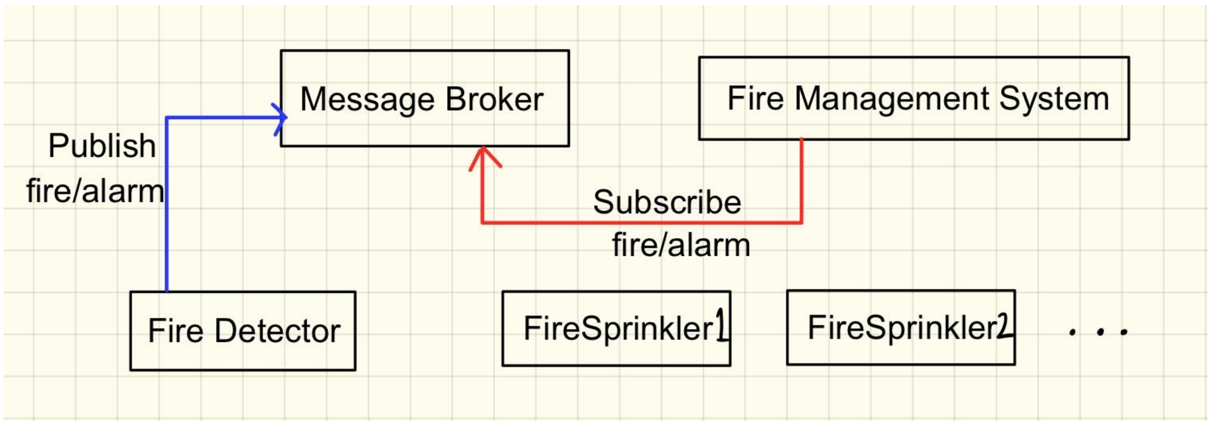
# 2021 Cloud IoT Services Assignment #4

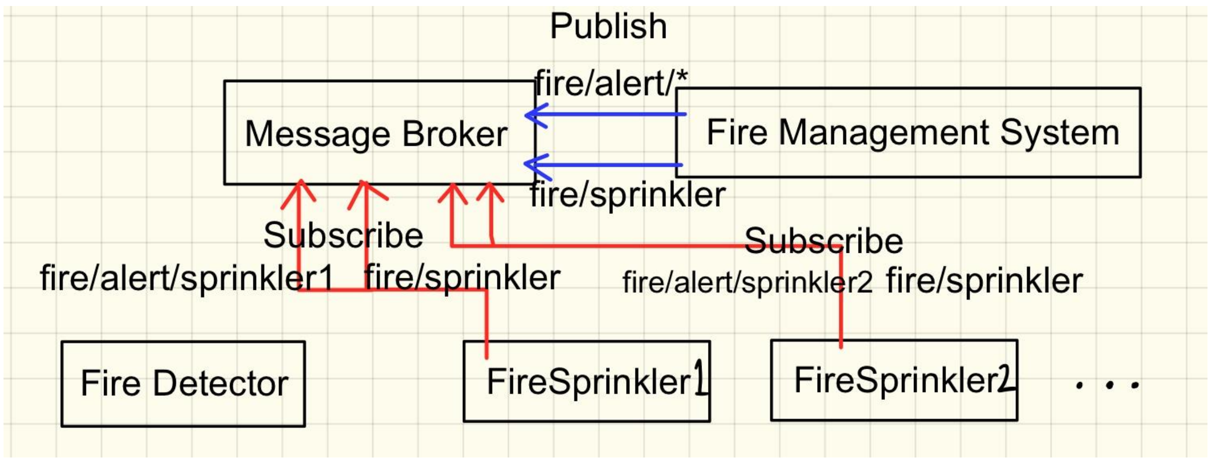
[ 이름 : 유지원 / 학번 : 201812627 ]

**Task : Create two applications for the virtual IoT devices**

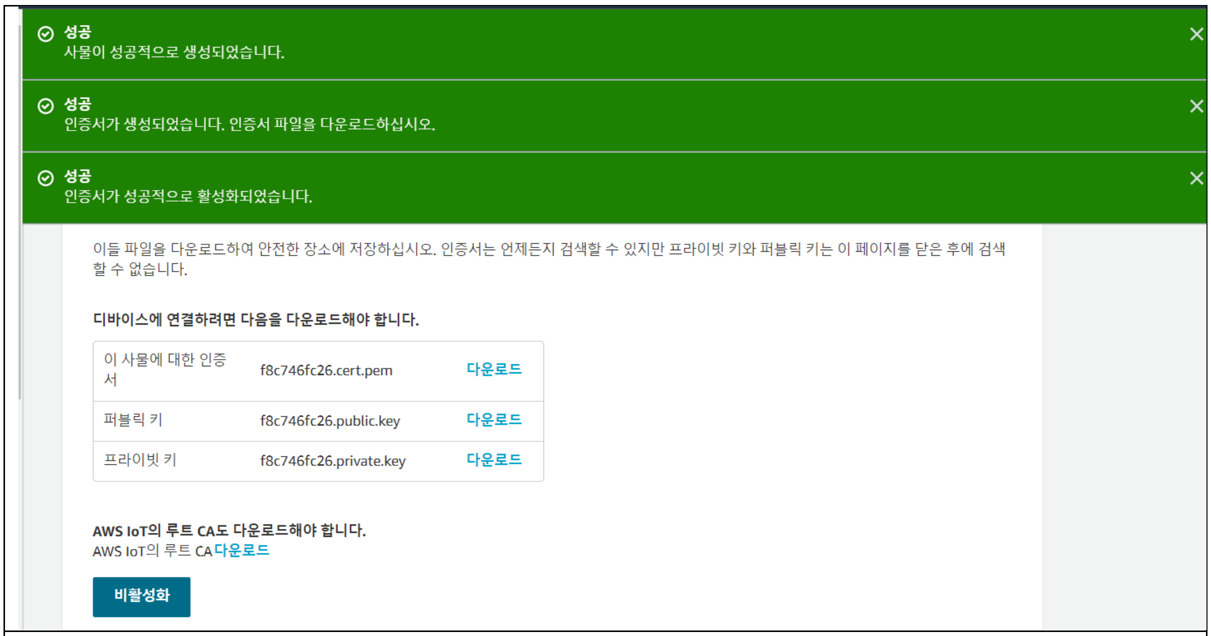
**<Fire Detector & Fire Sprinkler & Fire Management System>**

## 0. Fire System





## 1. 사물 생성



성공
사물이 성공적으로 생성되었습니다.

성공
인증서가 생성되었습니다. 인증서 파일을 다운로드하십시오.

성공
인증서가 성공적으로 활성화되었습니다.

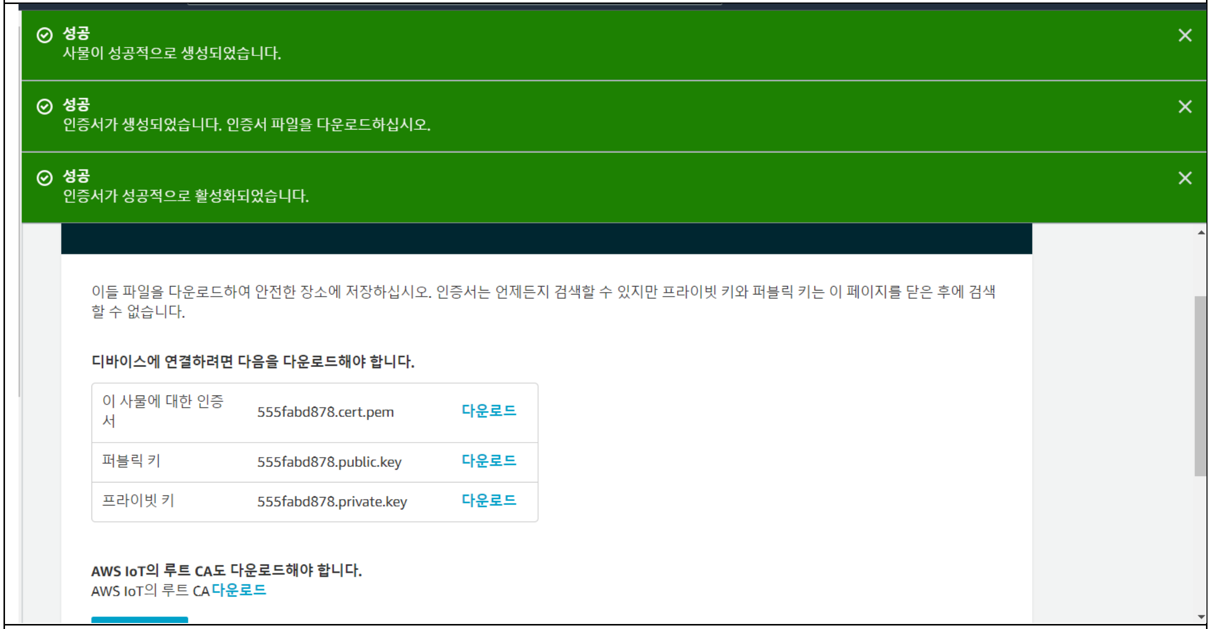
이들 파일을 다운로드하여 안전한 장소에 저장하십시오. 인증서는 언제든지 검색할 수 있지만 프라이빗 키와 퍼블릭 키는 이 페이지를 닫은 후에 검색할 수 없습니다.

**디바이스에 연결하려면 다음을 다운로드해야 합니다.**

| | | |
|---|---|---|
| 이 사물에 대한 인증서 | f8c746fc26.cert.pem | 다운로드 |
| 퍼블릭 키 | f8c746fc26.public.key | 다운로드 |
| 프라이빗 키 | f8c746fc26.private.key | 다운로드 |

**AWS IoT의 루트 CA도 다운로드해야 합니다.**
AWS IoT의 루트 CA 다운로드

**fireDetector 사물 생성**



성공
사물이 성공적으로 생성되었습니다.

성공
인증서가 생성되었습니다. 인증서 파일을 다운로드하십시오.

성공
인증서가 성공적으로 활성화되었습니다.

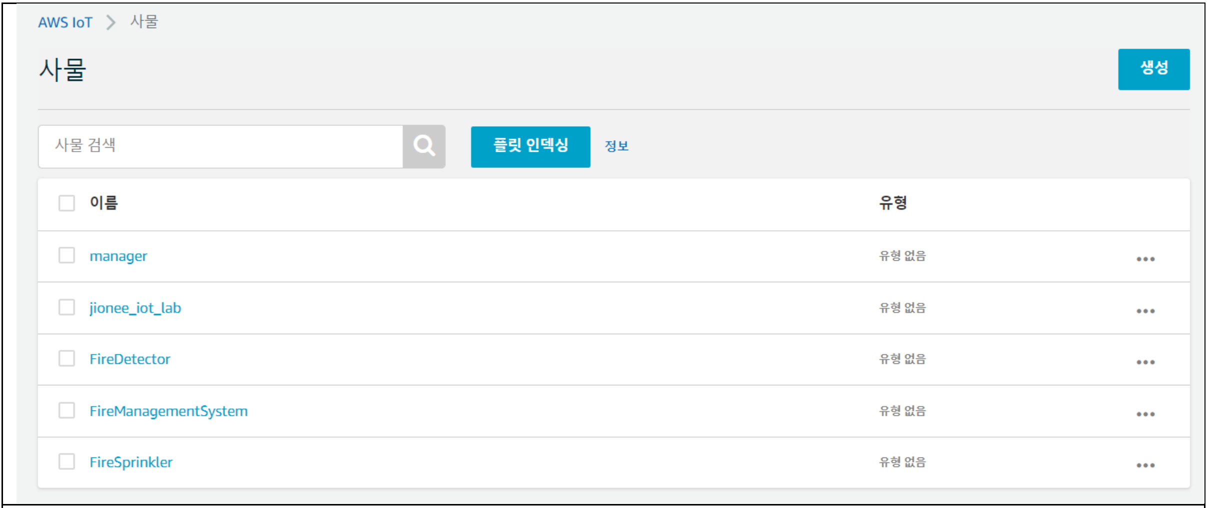
이들 파일을 다운로드하여 안전한 장소에 저장하십시오. 인증서는 언제든지 검색할 수 있지만 프라이빗 키와 퍼블릭 키는 이 페이지를 닫은 후에 검색할 수 없습니다.

**디바이스에 연결하려면 다음을 다운로드해야 합니다.**

| | | |
|---|---|---|
| 이 사물에 대한 인증서 | 555fabd878.cert.pem | 다운로드 |
| 퍼블릭 키 | 555fabd878.public.key | 다운로드 |
| 프라이빗 키 | 555fabd878.private.key | 다운로드 |

**AWS IoT의 루트 CA도 다운로드해야 합니다.**
AWS IoT의 루트 CA 다운로드

**FireManagement 사물 생성**

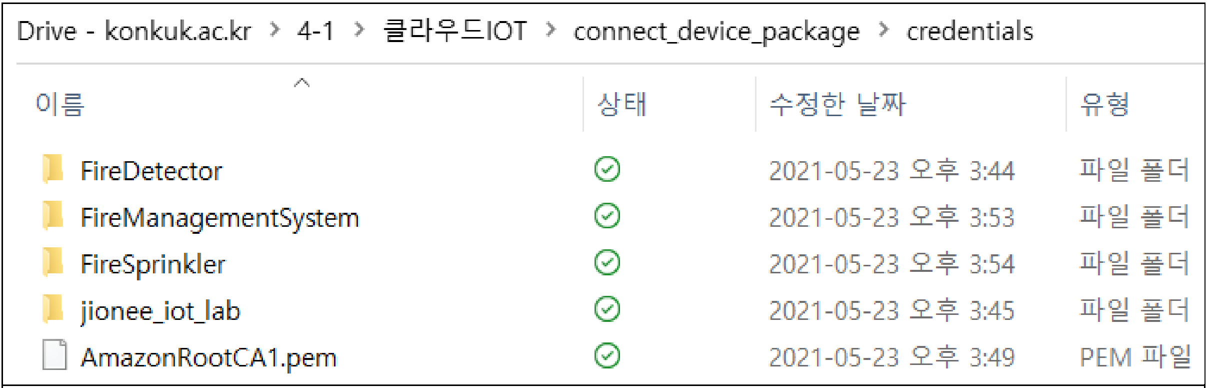


AWS IoT > 사물

사물

| 이름 | 유형 |
|---|---|
| manager | 유형 없음 |
| jionee_iot_lab | 유형 없음 |
| FireDetector | 유형 없음 |
| FireManagementSystem | 유형 없음 |
| FireSprinkler | 유형 없음 |

**Fire detector, fire sprinkler, fire management system 사물 생성완료**

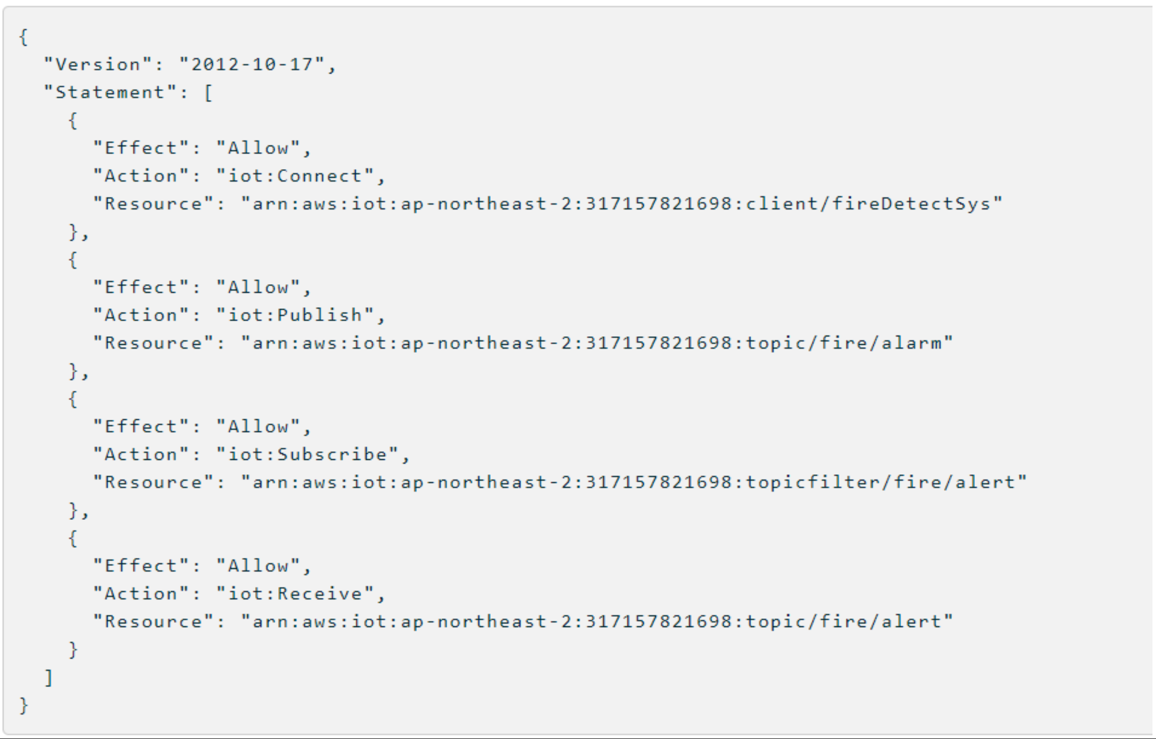


Drive - konkuk.ac.kr > 4-1 > 클라우드IOT > connect_device_package > credentials

| 이름 | 상태 | 수정한 날짜 | 유형 |
|---|---|---|---|
| FireDetector | ✓ | 2021-05-23 오후 3:44 | 파일 폴더 |
| FireManagementSystem | ✓ | 2021-05-23 오후 3:53 | 파일 폴더 |
| FireSprinkler | ✓ | 2021-05-23 오후 3:54 | 파일 폴더 |
| jionee_iot_lab | ✓ | 2021-05-23 오후 3:45 | 파일 폴더 |
| AmazonRootCA1.pem | ✓ | 2021-05-23 오후 3:49 | PEM 파일 |

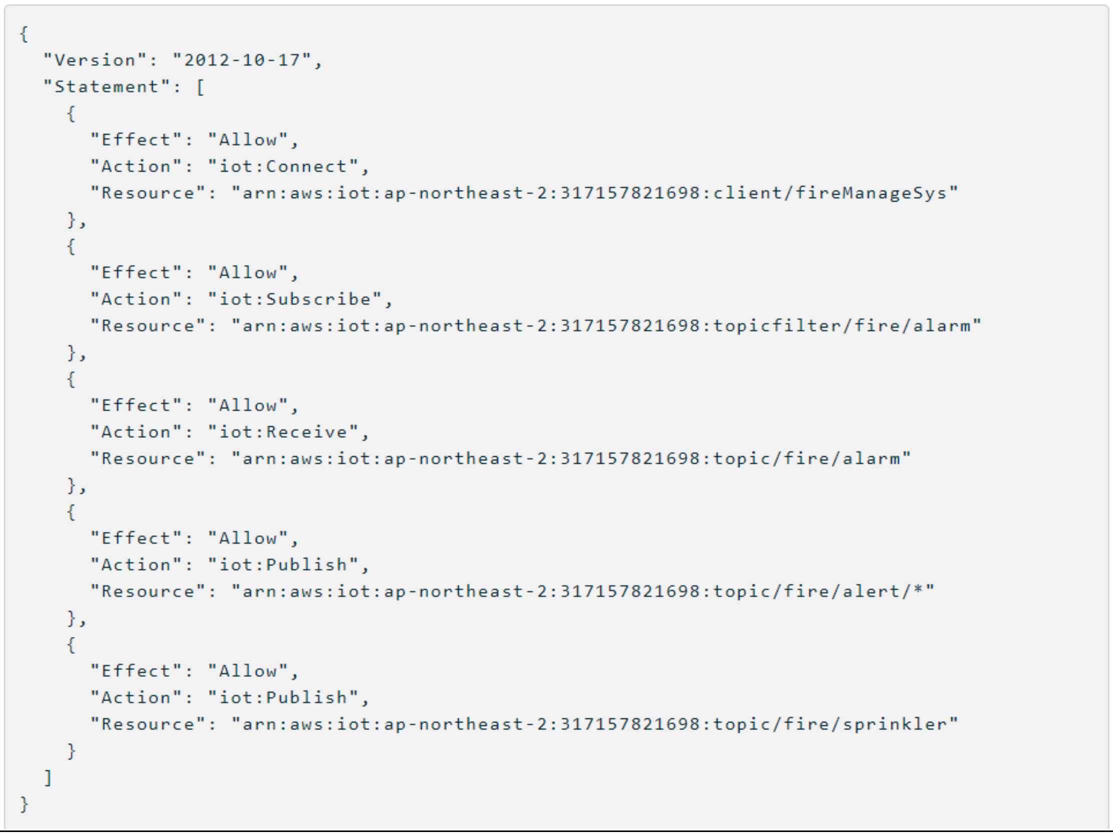
**Credential 관리**

## 2. 정책 생성 및 연결



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:Connect",
      "Resource": "arn:aws:iot:ap-northeast-2:317157821698:client/fireDetectSys"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Publish",
      "Resource": "arn:aws:iot:ap-northeast-2:317157821698:topic/fire/alarm"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:ap-northeast-2:317157821698:topicfilter/fire/alert"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": "arn:aws:iot:ap-northeast-2:317157821698:topic/fire/alert"
    }
  ]
}
```

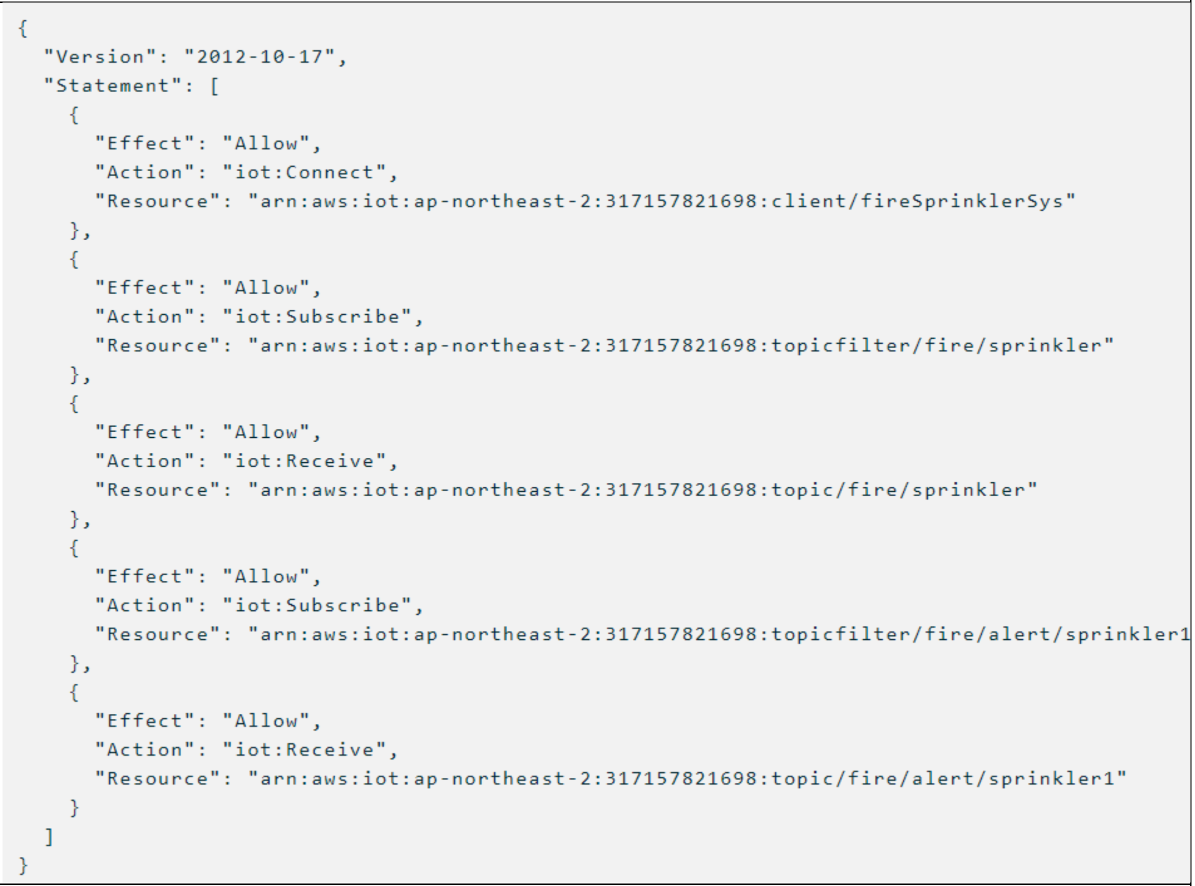
**Policy4fireDectector 작성**
**(connect-fireDetectSys, publish-fire/alarm, subscribe-fire/alert, receive-fire/alert)**



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:Connect",
      "Resource": "arn:aws:iot:ap-northeast-2:317157821698:client/fireManageSys"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:ap-northeast-2:317157821698:topicfilter/fire/alarm"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": "arn:aws:iot:ap-northeast-2:317157821698:topic/fire/alarm"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Publish",
      "Resource": "arn:aws:iot:ap-northeast-2:317157821698:topic/fire/alert/*"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Publish",
      "Resource": "arn:aws:iot:ap-northeast-2:317157821698:topic/fire/sprinkler"
    }
  ]
}
```

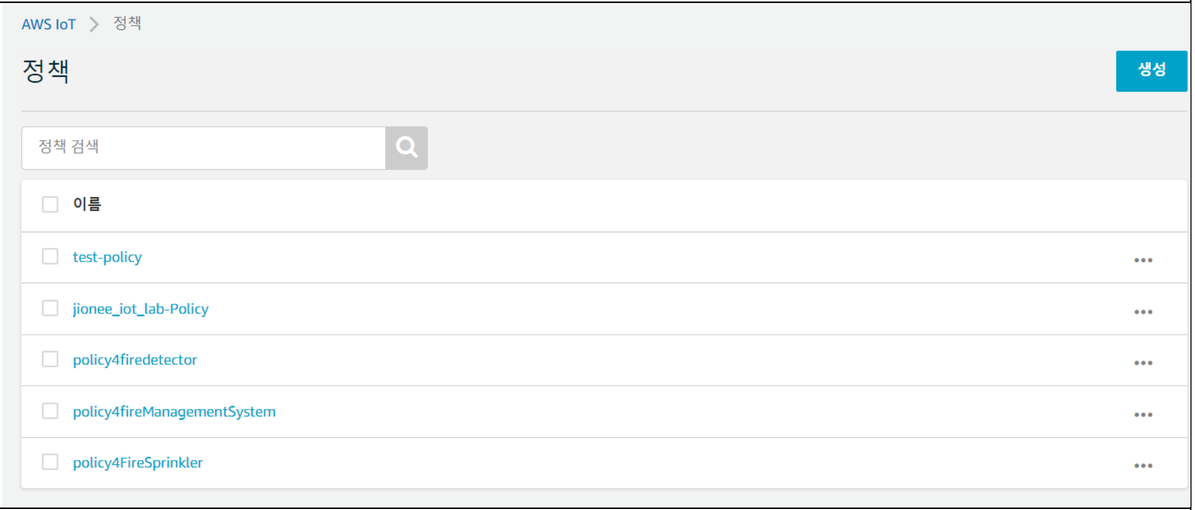
**Policy4fireManagementSystem 작성**

(connect-fireManagaSys, subscribe-fire/alarm, receive-fire/alarm, publish-fire/alert/*, publish-fire/sprinkler)



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:Connect",
      "Resource": "arn:aws:iot:ap-northeast-2:317157821698:client/fireSprinklerSys"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:ap-northeast-2:317157821698:topicfilter/fire/sprinkler"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": "arn:aws:iot:ap-northeast-2:317157821698:topic/fire/sprinkler"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:ap-northeast-2:317157821698:topicfilter/fire/alert/sprinkler1
    },
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": "arn:aws:iot:ap-northeast-2:317157821698:topic/fire/alert/sprinkler1"
    }
  ]
}
```

Policy4fireSprinkler 작성
(connect-fireSprinklerSys, subscribe-fire/sprinkler, receive-fire/sprinkler, subscribe-fire/alert/sprinkler1, receive-fire/alert/sprinkler1)



AWS IoT > 정책

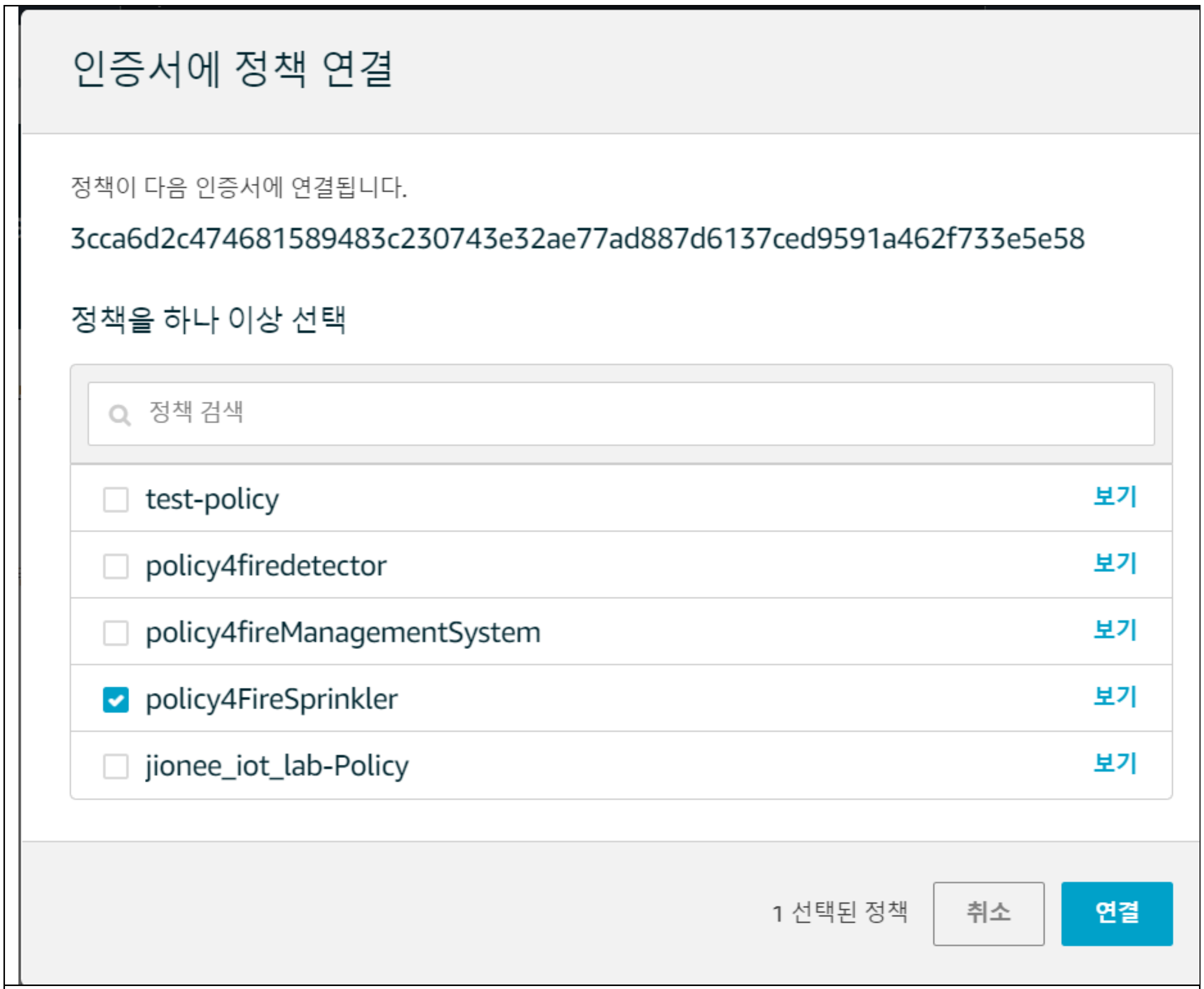
정책

생성

정책 검색

| 이름 |
|---|
| test-policy |
| jionee_iot_lab-Policy |
| policy4firedetector |
| policy4fireManagementSystem |
| policy4FireSprinkler |

정책 작성 완료



인증서에 정책 연결

정책이 다음 인증서에 연결됩니다.

3cca6d2c474681589483c230743e32ae77ad887d6137ced9591a462f733e5e58

정책을 하나 이상 선택

정책 검색

- [ ] test-policy 보기
- [ ] policy4firedetector 보기
- [ ] policy4fireManagementSystem 보기
- [x] policy4FireSprinkler 보기
- [ ] jionee_iot_lab-Policy 보기

1 선택된 정책 취소 연결

각 사물에 정책 연결 완료

## 3. node js 코드 작성

### 3-1) Fire Detector

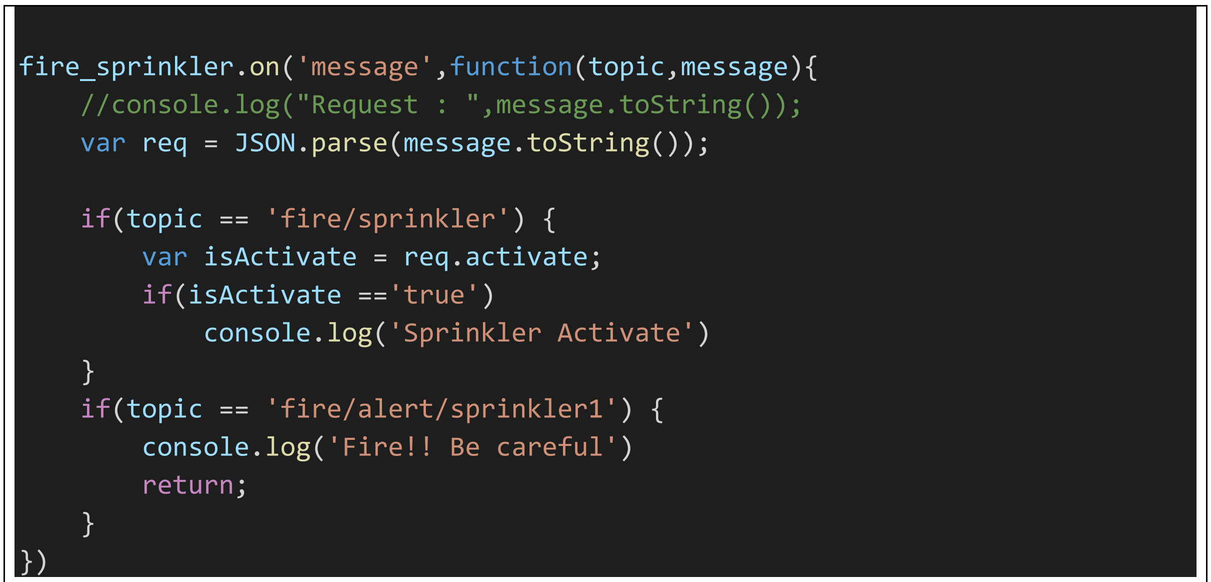
```
var awsIot = require('aws-iot-device-sdk');

var fire_detector=awsIot.device({
    keyPath: "./credentials/FireDetector/f8c746fc26-private.pem.key",
    certPath: "./credentials/FireDetector/f8c746fc26-certificate.pem.crt",
    caPath: "./credentials/AmazonRootCA1.pem",
    clientId: "fireDetectSys",
    host: "a3323sqgsmclbb-ats.iot.ap-northeast-2.amazonaws.com"
});

fire_detector.on('connect', function(){
    console.log('Fire Detector connected');

    setInterval(function(){
        var message = {'notify' : 'fire/alarm', 'msg' : 'fire!'};
        console.log('publish to FMS' + JSON.stringify(message));
        fire_detector.publish('fire/alarm', JSON.stringify(message));
    }, 3000);
});
```

### 3-2) Fire Management System

```
var awsIot = require('aws-iot-device-sdk');

var fire_management=awsIot.device({
    keyPath: "./credentials/FireManagementSystem/555fabd878-
private.pem.key",
    certPath: "./credentials/FireManagementSystem/555fabd878-
certificate.pem.crt",
    caPath: "./credentials/AmazonRootCA1.pem",
    clientId: "fireManageSys",
    host: "a3323sqgsmclbb-ats.iot.ap-northeast-2.amazonaws.com"
});

fire_management.subscribe('fire/alarm',function(){
    console.log('subscribing to the topic fire/alarm !');
});

fire_management.on('message',function(topic,message){
    console.log("Request : ",message.toString());
    if(topic != 'fire/alarm') {
        console.log('no fire alarm')
        return;
    }

    var req = JSON.parse(message.toString());
    var isFire = req.msg;
    if(isFire == "fire!"){
        //publish alert to any device
        var message = {'notify' : 'fire/alert/sprinkler1'};
        console.log('publish to Any Device' + JSON.stringify(message));
        fire_management.publish('fire/alert/sprinkler1', JSON.stringify(mess
age));

        // var message = {'notify' : 'fire/alert/sprinkler2'};
        // console.log('publish to Any Device' + JSON.stringify(message));
        // fire_management.publish('fire/alert/sprinkler2', JSON.stringify(m
essage));

        //publish Activation Command to sprinkler
        var message = {'notify' : 'fire/sprinkler', 'activate' : 'true'};
        console.log('publish to Sprinkler' + JSON.stringify(message));
        fire_management.publish('fire/sprinkler', JSON.stringify(message));
    }
    else{
        console.log('It is not fire :)')
    }
})
```

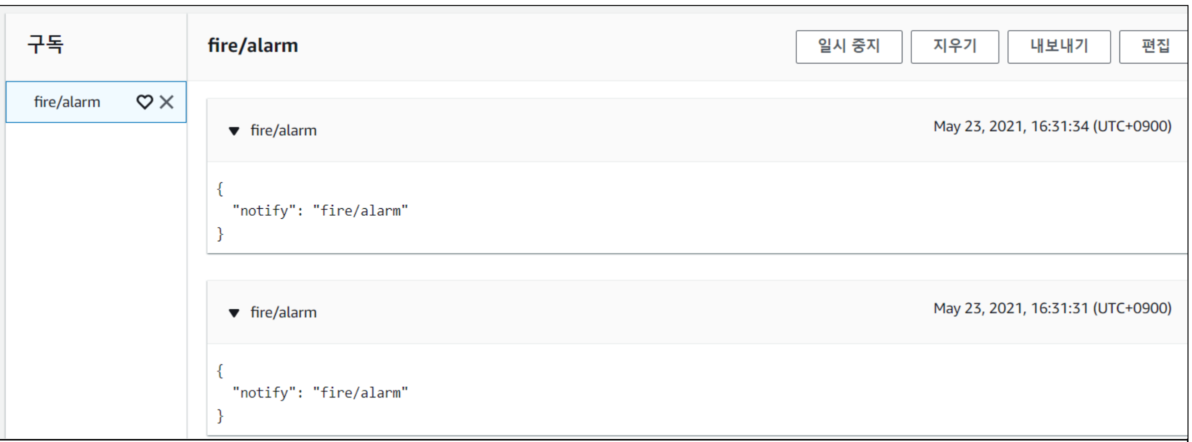
### 3-3) Fire Sprinkler

```
var awsIot = require('aws-iot-device-sdk');

var fire_sprinkler=awsIot.device({
    keyPath: "./credentials/FireSprinkler/3cca6d2c47-private.pem.key",
    certPath: "./credentials/FireSprinkler/3cca6d2c47-certificate.pem.crt",
    caPath: "./credentials/AmazonRootCA1.pem",
    clientId: "fireSprinklerSys",
    host: "a3323sqgsmclbb-ats.iot.ap-northeast-2.amazonaws.com"
});

fire_sprinkler.subscribe('fire/sprinkler',function(){
    console.log('subscribing to the topic fire/sprinkler !');
});
fire_sprinkler.subscribe('fire/alert/sprinkler1',function(){
    console.log('subscribing to the topic fire/alert/sprinkler1 !');
});
```



```
fire_sprinkler.on('message',function(topic,message){
    //console.log("Request : ",message.toString());
    var req = JSON.parse(message.toString());

    if(topic == 'fire/sprinkler') {
        var isActivate = req.activate;
        if(isActivate =='true')
            console.log('Sprinkler Activate')
    }
    if(topic == 'fire/alert/sprinkler1') {
        console.log('Fire!! Be careful')
        return;
    }
})
```

## 4. 중간 테스트



fire/alarm

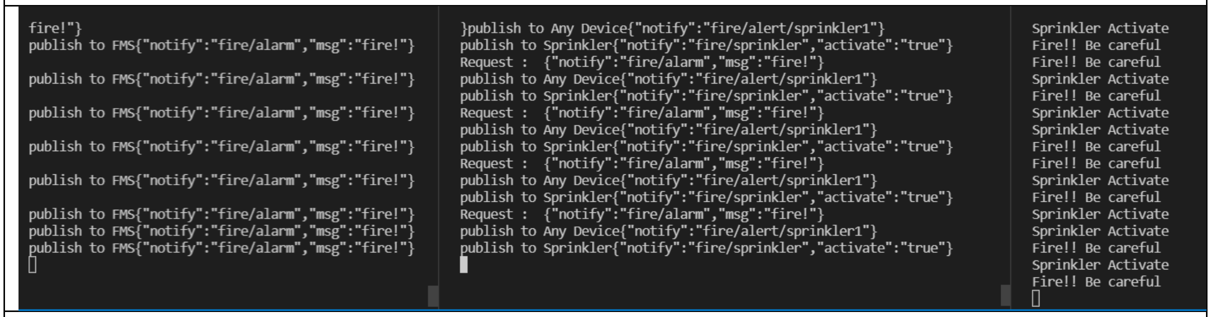
fire/alarm — May 23, 2021, 16:31:34 (UTC+0900)

```
{
  "notify": "fire/alarm"
}
```

fire/alarm — May 23, 2021, 16:31:31 (UTC+0900)

```
{
  "notify": "fire/alarm"
}
```

**fireDetector.js 작성 후 fire/alarm publish 정상 작동 테스트**
**(aws 내의 test 에서 subscribe 사용)**



```
fire!"}
publish to FMS{"notify":"fire/alarm","msg":"fire!"}

publish to FMS{"notify":"fire/alarm","msg":"fire!"}

publish to FMS{"notify":"fire/alarm","msg":"fire!"}

publish to FMS{"notify":"fire/alarm","msg":"fire!"}

publish to FMS{"notify":"fire/alarm","msg":"fire!"}

publish to FMS{"notify":"fire/alarm","msg":"fire!"}
publish to FMS{"notify":"fire/alarm","msg":"fire!"}
publish to FMS{"notify":"fire/alarm","msg":"fire!"}
```

```
}publish to Any Device{"notify":"fire/alert/sprinkler1"}
publish to Sprinkler{"notify":"fire/sprinkler","activate":"true"}
Request :  {"notify":"fire/alarm","msg":"fire!"}
publish to Any Device{"notify":"fire/alert/sprinkler1"}
publish to Sprinkler{"notify":"fire/sprinkler","activate":"true"}
Request :  {"notify":"fire/alarm","msg":"fire!"}
publish to Any Device{"notify":"fire/alert/sprinkler1"}
publish to Sprinkler{"notify":"fire/sprinkler","activate":"true"}
Request :  {"notify":"fire/alarm","msg":"fire!"}
publish to Any Device{"notify":"fire/alert/sprinkler1"}
publish to Sprinkler{"notify":"fire/sprinkler","activate":"true"}
Request :  {"notify":"fire/alarm","msg":"fire!"}
publish to Any Device{"notify":"fire/alert/sprinkler1"}
publish to Sprinkler{"notify":"fire/sprinkler","activate":"true"}
```

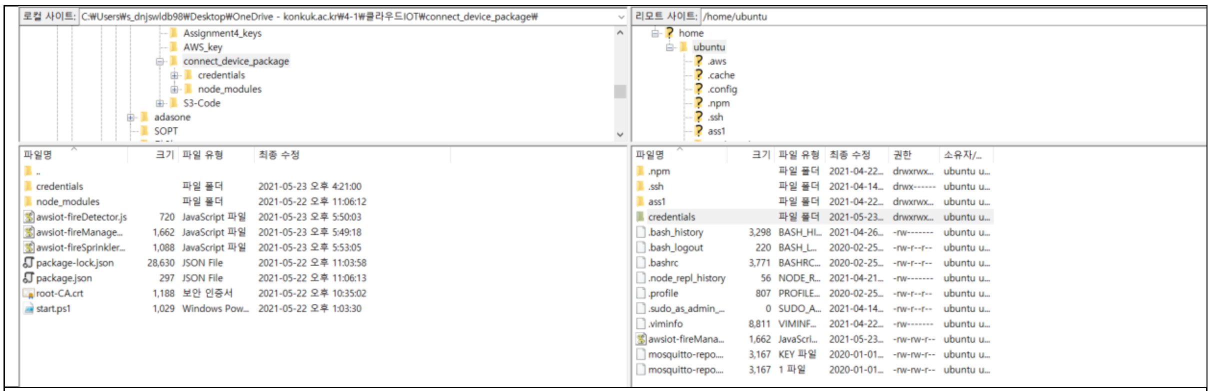
```
Sprinkler Activate
Fire!! Be careful
Fire!! Be careful
Sprinkler Activate
Fire!! Be careful
Sprinkler Activate
Sprinkler Activate
Fire!! Be careful
Fire!! Be careful
Sprinkler Activate
Fire!! Be careful
Sprinkler Activate
Sprinkler Activate
Fire!! Be careful
Sprinkler Activate
Fire!! Be careful
```

**좌 : node fireDetector 중간 : node fireManagementSystem 우 : node fireSprinkler 실행 결과**
**(한 로컬 내에서 세 노드 파일 모두 실행)**
**>> fireDetector : 정상적으로 fire/alarm 을 보낸다.**

**>> fireManagementSystem : 정상적으로 fire/alarm 을 구독하여 신호를 받고,**
**Fire/alert/sprinkler1, fire/sprinkler 신호를 publish 한다.**

**>> fireSprinkler : 정상적으로 fire/alert/sprinkler1, fire/sprinkler 를 구독하여 신호를 받고 각각 “Fire!! Be careful” 과 “Sprinkler Activate” 동작을 실행한다.**

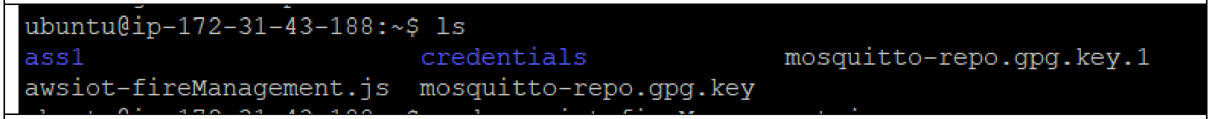
## 4. fire management system EC2 서버에 올리기



```
로컬 사이트: C:₩Users₩s_dnjswldb98₩Desktop₩OneDrive - konkuk.ac.kr₩4-1₩클라우드IOT₩connect_device_package₩
    Assignment4_keys
    AWS_key
    connect_device_package
        credentials
        node_modules
    S3-Code
  adasone
  SOPT

파일명                  크기    파일 유형         최종 수정
..
credentials                     파일 폴더         2021-05-23 오후 4:21:00
node_modules                    파일 폴더         2021-05-22 오후 11:06:12
awsiot-fireDetector.js   720    JavaScript 파일   2021-05-23 오후 5:50:03
awsiot-fireManage...     1,662  JavaScript 파일   2021-05-23 오후 5:49:18
awsiot-fireSprinkler...  1,088  JavaScript 파일   2021-05-23 오후 5:53:05
package-lock.json        28,630 JSON File         2021-05-22 오후 11:03:58
package.json             297    JSON File         2021-05-22 오후 11:06:13
root-CA.crt              1,188  보안 인증서       2021-05-22 오후 10:35:02
start.ps1                1,029  Windows Pow...    2021-05-22 오후 1:03:30
```

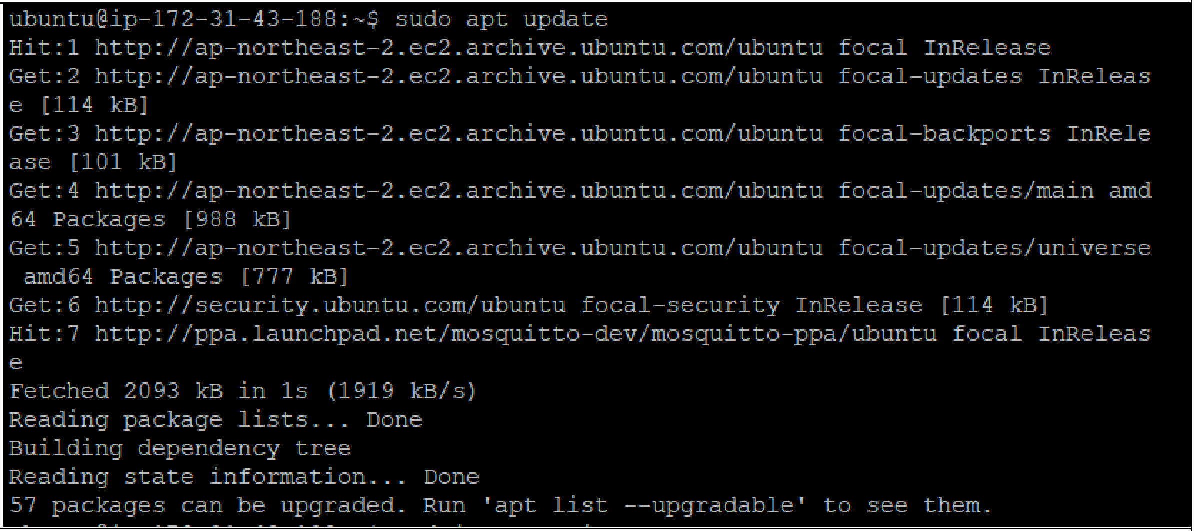
```
리모트 사이트: /home/ubuntu
home
  ubuntu
    .aws
    .cache
    .config
    .npm
    .ssh
    ass1

파일명               크기   파일 유형    최종 수정      권한         소유자/...
.npm                        파일 폴더    2021-04-22...  drwxrwx...   ubuntu u...
.ssh                        파일 폴더    2021-04-14...  drwx------   ubuntu u...
ass1                        파일 폴더    2021-04-22...  drwxrwx...   ubuntu u...
credentials                 파일 폴더    2021-05-23...  drwxrwx...   ubuntu u...
.bash_history        3,298  BASH_HI...   2021-04-26...  -rw-------   ubuntu u...
.bash_logout         220    BASH_L...    2020-02-25...  -rw-r--r--   ubuntu u...
.bashrc              3,771  BASHRC...    2020-02-25...  -rw-r--r--   ubuntu u...
.node_repl_history   56     NODE_R...    2021-04-21...  -rw-------   ubuntu u...
.profile             807    PROFILE...   2020-02-25...  -rw-r--r--   ubuntu u...
.sudo_as_admin_...   0      SUDO_A...    2021-04-14...  -rw-r--r--   ubuntu u...
.viminfo             8,811  VIMINF...    2021-04-22...  -rw-------   ubuntu u...
awsiot-fireMana...   1,662  JavaScri...  2021-05-23...  -rw-rw-r--   ubuntu u...
mosquitto-repo....   3,167  KEY 파일     2020-01-01...  -rw-rw-r--   ubuntu u...
mosquitto-repo....   3,167  1 파일       2020-01-01...  -rw-rw-r--   ubuntu u...
```

**파일질라 이용해서 ec2 서버로 파일 옮기기**



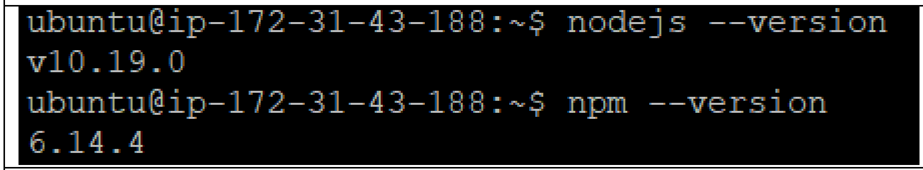
```
ubuntu@ip-172-31-43-188:~$ ls
ass1                           credentials                mosquitto-repo.gpg.key.1
awsiot-fireManagement.js  mosquitto-repo.gpg.key
```

**Putty 로 접속하여 파일 전송 확인 완료**



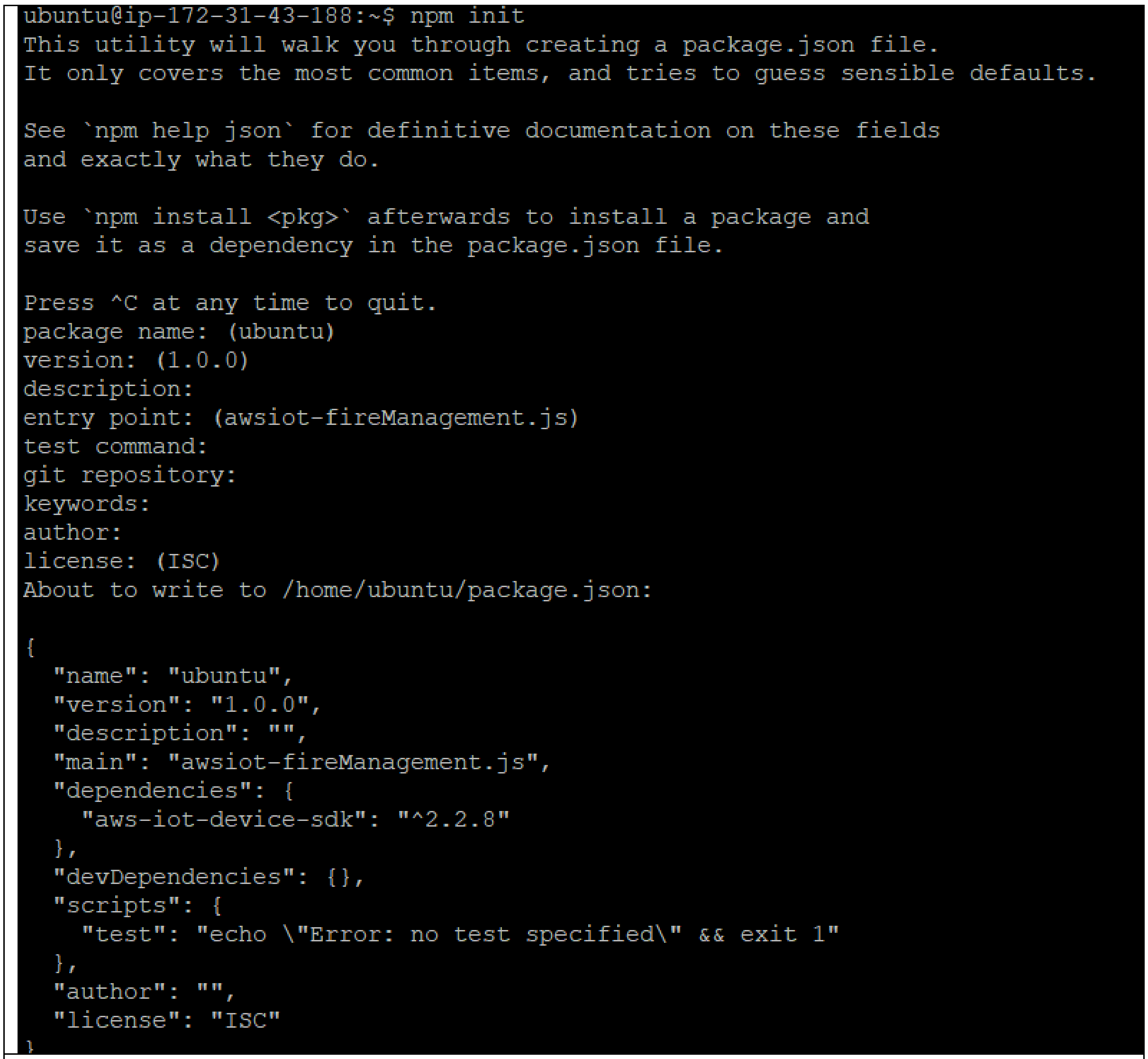
```
ubuntu@ip-172-31-43-188:~$ sudo apt update
Hit:1 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu focal-updates InReleas
e [114 kB]
Get:3 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu focal-backports InRele
ase [101 kB]
Get:4 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd
64 Packages [988 kB]
Get:5 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe
 amd64 Packages [777 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:7 http://ppa.launchpad.net/mosquitto-dev/mosquitto-ppa/ubuntu focal InReleas
e
Fetched 2093 kB in 1s (1919 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
57 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

**Sudo apt update**



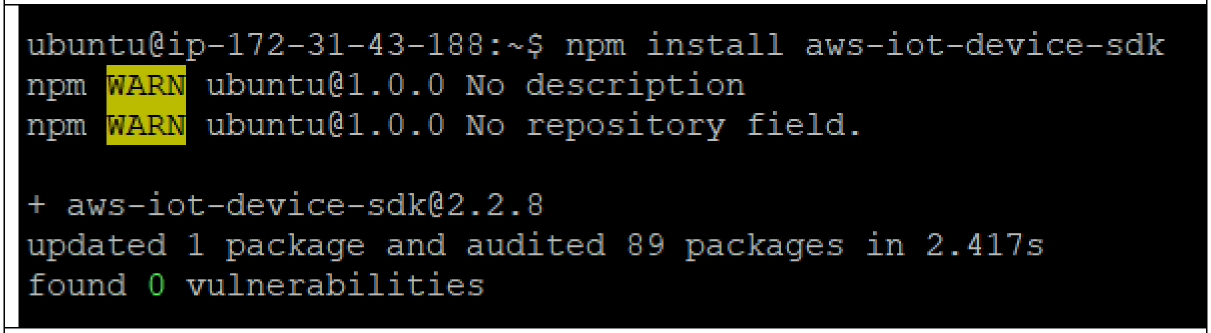
```
ubuntu@ip-172-31-43-188:~$ nodejs --version
v10.19.0
ubuntu@ip-172-31-43-188:~$ npm --version
6.14.4
```

**Nodejs, npm 버전 확인**



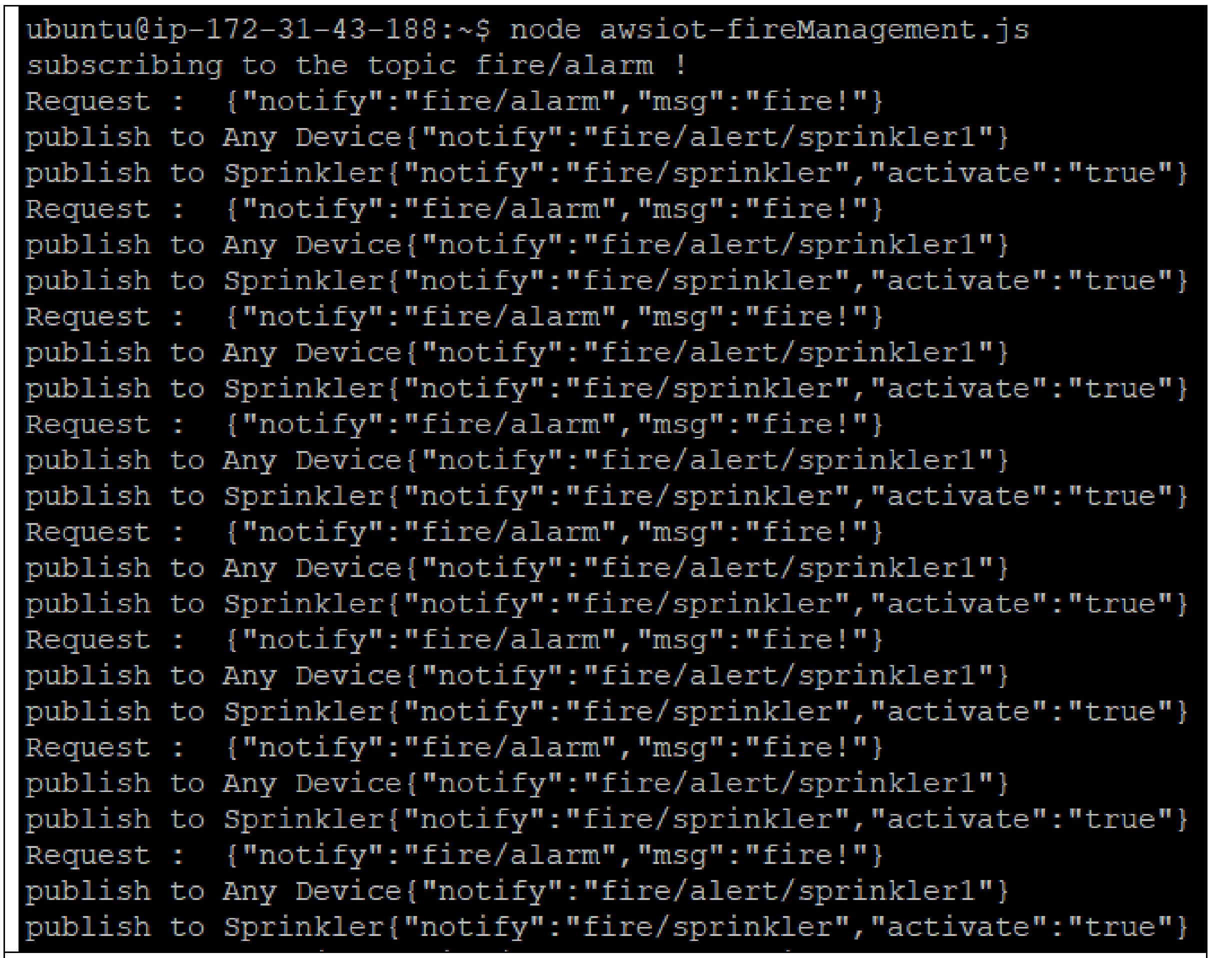
```
ubuntu@ip-172-31-43-188:~$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (ubuntu)
version: (1.0.0)
description:
entry point: (awsiot-fireManagement.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /home/ubuntu/package.json:

{
  "name": "ubuntu",
  "version": "1.0.0",
  "description": "",
  "main": "awsiot-fireManagement.js",
  "dependencies": {
    "aws-iot-device-sdk": "^2.2.8"
  },
  "devDependencies": {},
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}
```

**Npm init**



```
ubuntu@ip-172-31-43-188:~$ npm install aws-iot-device-sdk
npm WARN ubuntu@1.0.0 No description
npm WARN ubuntu@1.0.0 No repository field.

+ aws-iot-device-sdk@2.2.8
updated 1 package and audited 89 packages in 2.417s
found 0 vulnerabilities
```

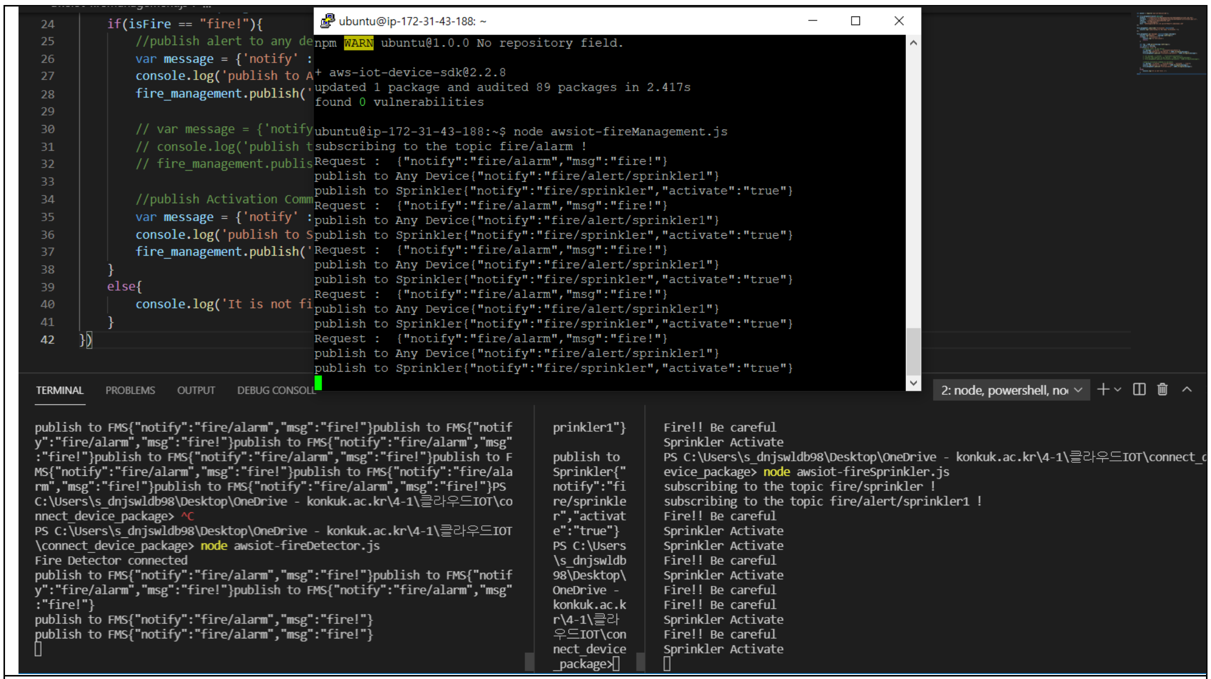
**Npm install aws-iot-device-sdk 설치**



```
ubuntu@ip-172-31-43-188:~$ node awsiot-fireManagement.js
subscribing to the topic fire/alarm !
Request :  {"notify":"fire/alarm","msg":"fire!"}
publish to Any Device{"notify":"fire/alert/sprinkler1"}
publish to Sprinkler{"notify":"fire/sprinkler","activate":"true"}
Request :  {"notify":"fire/alarm","msg":"fire!"}
publish to Any Device{"notify":"fire/alert/sprinkler1"}
publish to Sprinkler{"notify":"fire/sprinkler","activate":"true"}
Request :  {"notify":"fire/alarm","msg":"fire!"}
publish to Any Device{"notify":"fire/alert/sprinkler1"}
publish to Sprinkler{"notify":"fire/sprinkler","activate":"true"}
Request :  {"notify":"fire/alarm","msg":"fire!"}
publish to Any Device{"notify":"fire/alert/sprinkler1"}
publish to Sprinkler{"notify":"fire/sprinkler","activate":"true"}
Request :  {"notify":"fire/alarm","msg":"fire!"}
publish to Any Device{"notify":"fire/alert/sprinkler1"}
publish to Sprinkler{"notify":"fire/sprinkler","activate":"true"}
Request :  {"notify":"fire/alarm","msg":"fire!"}
publish to Any Device{"notify":"fire/alert/sprinkler1"}
publish to Sprinkler{"notify":"fire/sprinkler","activate":"true"}
Request :  {"notify":"fire/alarm","msg":"fire!"}
publish to Any Device{"notify":"fire/alert/sprinkler1"}
publish to Sprinkler{"notify":"fire/sprinkler","activate":"true"}
Request :  {"notify":"fire/alarm","msg":"fire!"}
publish to Any Device{"notify":"fire/alert/sprinkler1"}
publish to Sprinkler{"notify":"fire/sprinkler","activate":"true"}
```

**Node awsiot-fireManagement.js 파일 실행, 정상 작동 확인**

## 5. 최종 실행 결과



```
        if(isFire == "fire!"){
            //publish alert to any de
            var message = {'notify' :
            console.log('publish to A
            fire_management.publish('

            // var message = {'notify
            // console.log('publish t
            // fire_management.publis

            //publish Activation Comm
            var message = {'notify' :
            console.log('publish to S
            fire_management.publish('
        }
        else{
            console.log('It is not fi
        }
})
```

ubuntu@ip-172-31-43-188: ~

```
npm WARN ubuntu@1.0.0 No repository field.

+ aws-iot-device-sdk@2.2.8
updated 1 package and audited 89 packages in 2.417s
found 0 vulnerabilities

ubuntu@ip-172-31-43-188:~$ node awsiot-fireManagement.js
subscribing to the topic fire/alarm !
Request :  {"notify":"fire/alarm","msg":"fire!"}
publish to Any Device{"notify":"fire/alert/sprinkler1"}
publish to Sprinkler{"notify":"fire/sprinkler","activate":"true"}
Request :  {"notify":"fire/alarm","msg":"fire!"}
publish to Any Device{"notify":"fire/alert/sprinkler1"}
publish to Sprinkler{"notify":"fire/sprinkler","activate":"true"}
Request :  {"notify":"fire/alarm","msg":"fire!"}
publish to Any Device{"notify":"fire/alert/sprinkler1"}
publish to Sprinkler{"notify":"fire/sprinkler","activate":"true"}
Request :  {"notify":"fire/alarm","msg":"fire!"}
publish to Any Device{"notify":"fire/alert/sprinkler1"}
publish to Sprinkler{"notify":"fire/sprinkler","activate":"true"}
Request :  {"notify":"fire/alarm","msg":"fire!"}
publish to Any Device{"notify":"fire/alert/sprinkler1"}
publish to Sprinkler{"notify":"fire/sprinkler","activate":"true"}
```

```
publish to FMS{"notify":"fire/alarm","msg":"fire!"}publish to FMS{"notif
y":"fire/alarm","msg":"fire!"}publish to FMS{"notify":"fire/alarm","msg"
:"fire!"}publish to FMS{"notify":"fire/alarm","msg":"fire!"}publish to F
MS{"notify":"fire/alarm","msg":"fire!"}publish to FMS{"notify":"fire/ala
rm","msg":"fire!"}publish to FMS{"notify":"fire/alarm","msg":"fire!"}PS
C:\Users\s_dnjswldb98\Desktop\OneDrive - konkuk.ac.kr\4-1\클라우드IOT\co
nnect_device_package> ^C
PS C:\Users\s_dnjswldb98\Desktop\OneDrive - konkuk.ac.kr\4-1\클라우드IOT
\connect_device_package> node awsiot-fireDetector.js
Fire Detector connected
publish to FMS{"notify":"fire/alarm","msg":"fire!"}publish to FMS{"notif
y":"fire/alarm","msg":"fire!"}publish to FMS{"notify":"fire/alarm","msg"
:"fire!"}
publish to FMS{"notify":"fire/alarm","msg":"fire!"}
publish to FMS{"notify":"fire/alarm","msg":"fire!"}
```

```
prinkler1"}

publish to
Sprinkler{"
notify":"fi
re/sprinkle
r","activat
e":"true"}
PS C:\Users
\s_dnjswldb
98\Desktop\
OneDrive -
konkuk.ac.k
r\4-1\클라
우드IOT\con
nect_device
_package>
```

```
Fire!! Be careful
Sprinkler Activate
PS C:\Users\s_dnjswldb98\Desktop\OneDrive - konkuk.ac.kr\4-1\클라우드IOT\connect_d
evice_package> node awsiot-fireSprinkler.js
subscribing to the topic fire/sprinkler !
subscribing to the topic fire/alert/sprinkler1 !
Fire!! Be careful
Sprinkler Activate
Sprinkler Activate
Fire!! Be careful
Sprinkler Activate
Fire!! Be careful
Fire!! Be careful
Sprinkler Activate
Fire!! Be careful
Sprinkler Activate
```

**좌측하단 : fireDetector 실행,**
**중앙상단 : ec2 서버에서 fireManagementSystem 실행,**
**우측하단 : fireSprinkler 실행**
**(중앙하단은 로컬 실행 멈춰놓은 것)**

**>> fireDetector : 정상적으로 fire/alarm 을 보낸다.**

**>> fireManagementSystem : 정상적으로 fire/alarm 을 구독하여 신호를 받고,**
**Fire/alert/sprinkler1, fire/sprinkler 신호를 publish 한다.**

**>> fireSprinkler : 정상적으로 fire/alert/sprinkler1, fire/sprinkler 를 구독하여 신호를 받고 각각 “Fire!! Be careful” 과 “Sprinkler Activate” 동작을 실행한다.**

**>> 정상적으로 모두 작동함을 알 수 있다.**