

# 2021 Cloud IoT Services Assignment #1

[ 이름 : 유지원 / 학번 : 201812627 ]

## Task #1 : Create a MQTT Message Broker

### 1. Configuration of the MQTT message broker

- Public IP : 3.36.217.73

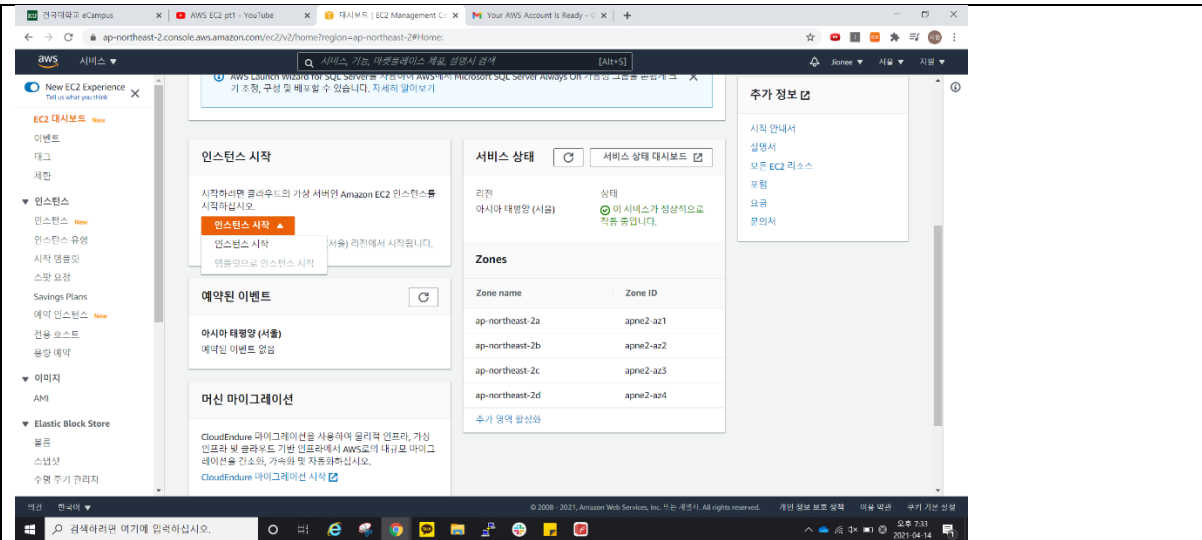
- Private IP : 172.31.43.188

- Topic names : test, assignment1

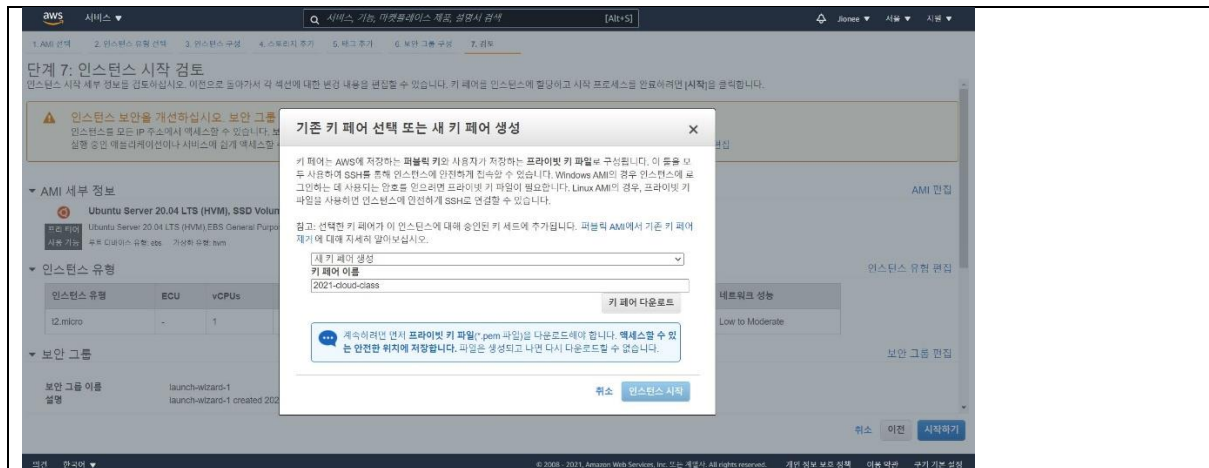
++ 추가적인 configuration에 대한 설명, reconfigure of security group

### 2. Steps

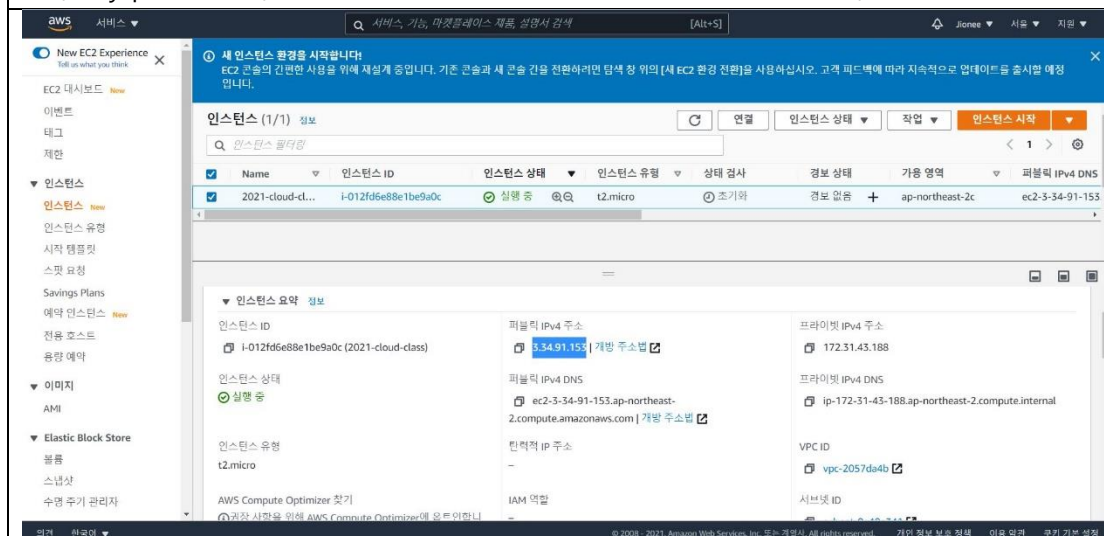
1) aws ec2 instance 생성, putty를 이용한 서버 연결



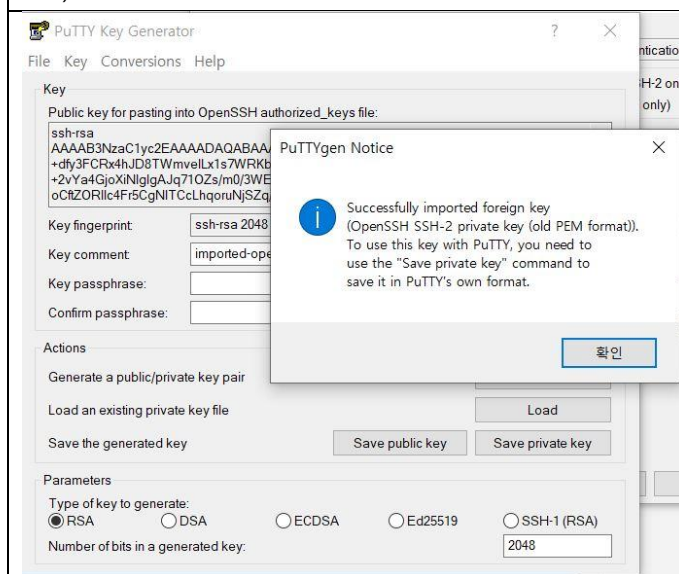
1-1) Instance 생성



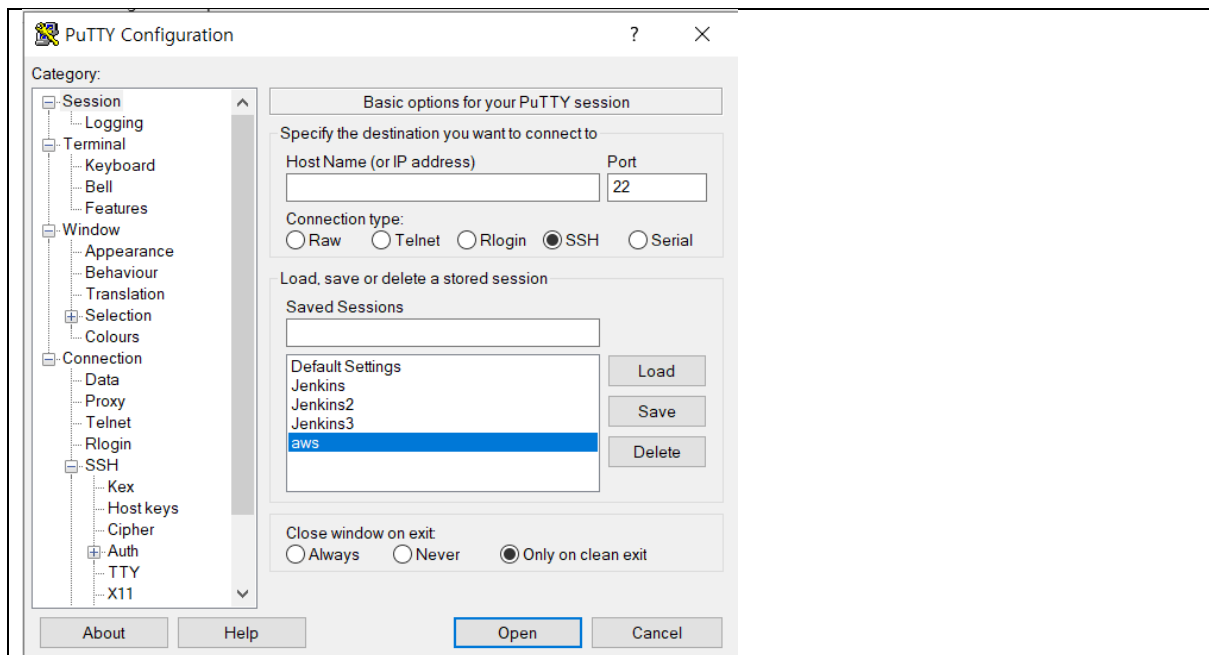
## 1-2) Key pair 생성 (잘 저장 해 두어야 함, 잃어버리면 찾을 수 없음)



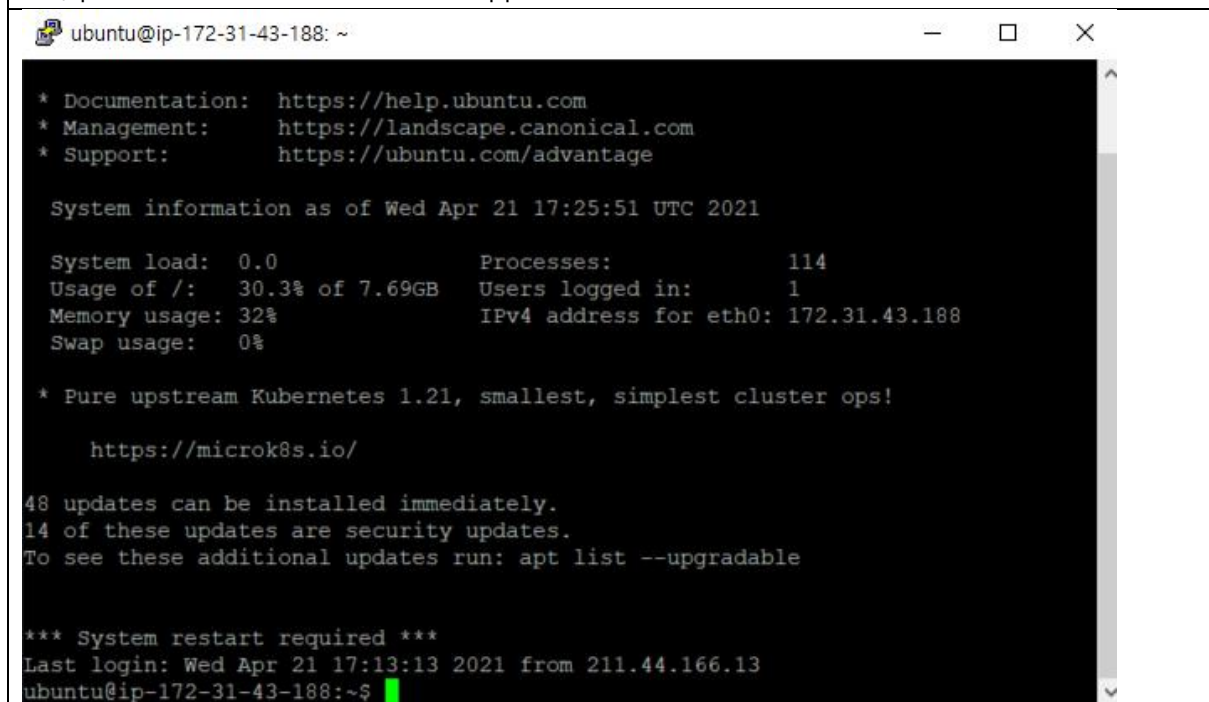
## 1-3) 생성 완료 모습



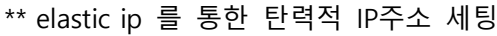
## 1-4) puttygen으로 ppk만들기



1-5) public IP입력과 SSH-Auth에서의 ppk파일로 ec2서버 연결



1-6) 연결 완료



```

npm
The following NEW packages will be installed:
  libc-ares2 libnode64 nodejs nodejs-doc
0 upgraded, 4 newly installed, 0 to remove and 59 not upgraded.
Need to get 6807 kB of archives.
After this operation, 30.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libc-ares2 amd64 1.15.0-1build1 [37.8 kB]
Get:2 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 libnode64 amd64 10.19.0-dfsg-3ubuntu1 [5765 kB]
Get:3 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 nodejs-doc all 10.19.0-dfsg-3ubuntu1 [942 kB]
Get:4 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 nodejs amd64 10.19.0-dfsg-3ubuntu1 [61.1 kB]
Fetched 6807 kB in 0s (22.0 MB/s)
Selecting previously unselected package libc-ares2:amd64.
(Reading database ... 60092 files and directories currently installed.)
Preparing to unpack .../libc-ares2_1.15.0-1build1_amd64.deb ...
Unpacking libc-ares2:amd64 (1.15.0-1build1) ...
Selecting previously unselected package libnode64:amd64.
Preparing to unpack .../libnode64_10.19.0-dfsg-3ubuntu1_amd64.deb ...
Unpacking libnode64:amd64 (10.19.0-dfsg-3ubuntu1) ...
Selecting previously unselected package nodejs-doc.
Preparing to unpack .../nodejs-doc_10.19.0-dfsg-3ubuntu1_all.deb ...
Unpacking nodejs-doc (10.19.0-dfsg-3ubuntu1) ...
Selecting previously unselected package nodejs.
Preparing to unpack .../nodejs_10.19.0-dfsg-3ubuntu1_amd64.deb ...
Unpacking nodejs (10.19.0-dfsg-3ubuntu1) ...
Setting up libc-ares2:amd64 (1.15.0-1build1) ...
Setting up libnode64:amd64 (10.19.0-dfsg-3ubuntu1) ...
Setting up nodejs-doc (10.19.0-dfsg-3ubuntu1) ...
Setting up nodejs (10.19.0-dfsg-3ubuntu1) ...
update-alternatives: using /usr/bin/nodejs to provide /usr/bin/js (js) in auto mode
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
Processing triggers for man-db (2.9.1-1) ...

Progress: [ 94%] [#####.....]

```

```

Fetched 394 kB in 0s (11.4 MB/s)
Selecting previously unselected package libdlt2:amd64.
(Reading database ... 74833 files and directories currently installed.)
Preparing to unpack .../libdlt2_2.18.4-0.1_amd64.deb ...
Unpacking libdlt2:amd64 (2.18.4-0.1) ...
Selecting previously unselected package libev4:amd64.
Preparing to unpack .../libev4_1%3a4.31-1_amd64.deb ...
Unpacking libev4:amd64 (1:4.31-1) ...
Selecting previously unselected package libwebsockets15:amd64.
Preparing to unpack .../libwebsockets15_3.2.1-3_amd64.deb ...
Unpacking libwebsockets15:amd64 (3.2.1-3) ...
Selecting previously unselected package mosquitto.
Preparing to unpack .../mosquitto_1.6.9-1_amd64.deb ...
Unpacking mosquitto (1.6.9-1) ...
Setting up libev4:amd64 (1:4.31-1) ...
Setting up libdlt2:amd64 (2.18.4-0.1) ...
Setting up libwebsockets15:amd64 (3.2.1-3) ...
Setting up mosquitto (1.6.9-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/mosquitto.service → /lib/systemd/system/mosquitto.service.
Processing triggers for systemd (245.4-4ubuntu3.4) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
ubuntu@ip-172-31-43-188:~$ sudo apt install mosquitto-clients
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libmosquitto1
The following NEW packages will be installed:
  libmosquitto1 mosquitto-clients
0 upgraded, 2 newly installed, 0 to remove and 58 not upgraded.
Need to get 105 kB of archives.
After this operation, 442 kB of additional disk space will be used.
Do you want to continue? [Y/n]

```

## 2-2) mosquitto 설치

```

ubuntu@ip-172-31-43-188:~$ sudo apt install mosquitto-clients
Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor preset: Active)
Active: activating (auto-restart) (Result: exit-code) since Thu 2021-04-15 01:24:24 UTC; 1min 10s ago
Docs: man:mosquitto.conf(5)
       man:mosquitto(8)
Process: 32598 ExecStartPre=/bin/mkdir -m 740 -p /var/log/mosquitto (code=exited, status=0/SUCCESS)
Process: 32599 ExecStartPre=/bin/chown mosquitto: /var/log/mosquitto (code=exited, status=0/SUCCESS)
Process: 32600 ExecStartPre=/bin/mkdir -m 740 -p /var/run/mosquitto (code=exited, status=0/SUCCESS)
Process: 32601 ExecStartPre=/bin/chown mosquitto: /var/run/mosquitto (code=exited, status=0/SUCCESS)
Process: 32602 ExecStart=/usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf (code=exited, status=1/FAILURE)
Main PID: 32602 (code=exited, status=1/FAILURE)

Apr 15 01:24:24 ip-172-31-43-188 systemd[1]: mosquitto.service: Main process exited, code=exited, status=1/FAILURE
Apr 15 01:24:24 ip-172-31-43-188 systemd[1]: mosquitto.service: Failed with result 'exited'.
Apr 15 01:24:24 ip-172-31-43-188 systemd[1]: Failed to start Mosquitto MQTT Broker.
dpkg: error processing package mosquitto (--configure):
 installed mosquitto package post-installation script subprocess returned error
Errors were encountered while processing:
 mosquitto
E: Sub-process /usr/bin/dpkg returned an error code (1)
ubuntu@ip-172-31-43-188:~$ ^C
ubuntu@ip-172-31-43-188:~$ mosquitto_sub -h localhost -t "test"
hello
hello world

```

```

ubuntu@ip-172-31-43-188:~$ mosquitto_pub -h localhost -t "test" -m "hello"
ubuntu@ip-172-31-43-188:~$ mosquitto_pub -h localhost -t "test" -m "hello world"

```

## 2-3) mqtt 메시지 주고 받기 성공 (Left : Subscriber, Right : Publisher)

### ▼ 인바운드 규칙

Q 필터 규칙			
포트 범위	프로토콜	원본	보안 그룹
전체	전체	0.0.0.0/0	launch-wizard-1
전체	전체	::/0	launch-wizard-1
22	TCP	0.0.0.0/0	launch-wizard-1
1883	TCP	0.0.0.0/0	launch-wizard-1
1883	TCP	175.125.148.75/32	launch-wizard-1

### ▼ 아웃바운드 규칙

Q 필터 규칙			
포트 범위	프로토콜	대상	보안 그룹
전체	전체	0.0.0.0/0	launch-wizard-1
1883	TCP	0.0.0.0/0	launch-wizard-1
1883	TCP	::/0	launch-wizard-1

\*\* 보안 그룹 inbound, outbound 설정

Inbound

- 전체 : 누구나 접속 가능해서 test할 수 있게 하기 위함
- 22 : putty연결 위함
- 1883 : mosquito 연결 (175.XX -> 친구 컴퓨터에서도 작동하는지 보기 위함)

## Task #2 : Build an application to copy a file from one program to another

### 1. 설명

- fileTransfer.js에서 같은 폴더에 존재하는 testTxt.txt파일을 읽어 assignment1을 topic으로 파일을 publish한다.
- fileReceiver.js에서는 assignment1을 topic으로 subscribe하여 pulish된 파일을 message로 받아 같은 폴더 상에 'newTxt'라는 파일 이름으로 파일을 작성한다.

=> 이를 통해 한 프로그램으로부터 다른 프로그램으로 파일이 복사되는 application을 만들 수 있다.

### 2. 코드

#### fileTransfer.js

```
console.log("FILESENDER START");
var mqtt = require('mqtt');
var client = mqtt.connect('mqtt://3.36.217.73'); //mqtt 와 ec2 서버 연결

var fs = require('fs');
// 방법 1. 파일 통째로 읽기
var file = fs.readFileSync('testTxt.txt');

// 방법 2. 파일 buffer 이용해서 읽기
// file = 'testTxt.txt'
// data = fs.readFileSync(file);
// buf = {
//     "name": file,
//     "data": data,
// };

client.on('connect',function(){
    console.log("CONNECTION");
```

```

    //방법 1
    client.publish('assignment1', file); //publish, file transfer
    //방법 2
    //console.log('buf==>'+JSON.stringify(buf));
    //client.publish('assignment1', JSON.stringify(buf));
  });

  client.on('message', function(topic,message){ //when got message from Topic'
  assignment1'
    console.log("MESSAGE");
    console.log(topic + ' : send');
    client.end(); //close Client
  })

```

### fileReceiver.js

```

console.log("RECEIVER START");

var mqtt = require('mqtt');
var client = mqtt.connect('mqtt://3.36.217.73');
var fs = require('fs');

client.on('connect', function(){
  console.log("CONNECTION");
  client.subscribe('assignment1');
});

client.on('message', function(topic,message){
  console.log("MESSAGE!!"+message);
  //방법 1
  fs.writeFileSync('newTxt',message);

  //방법 2
  //data = JSON.parse(message);
  //fs.writeFileSync('newTxt',data.data);

  client.end();
});

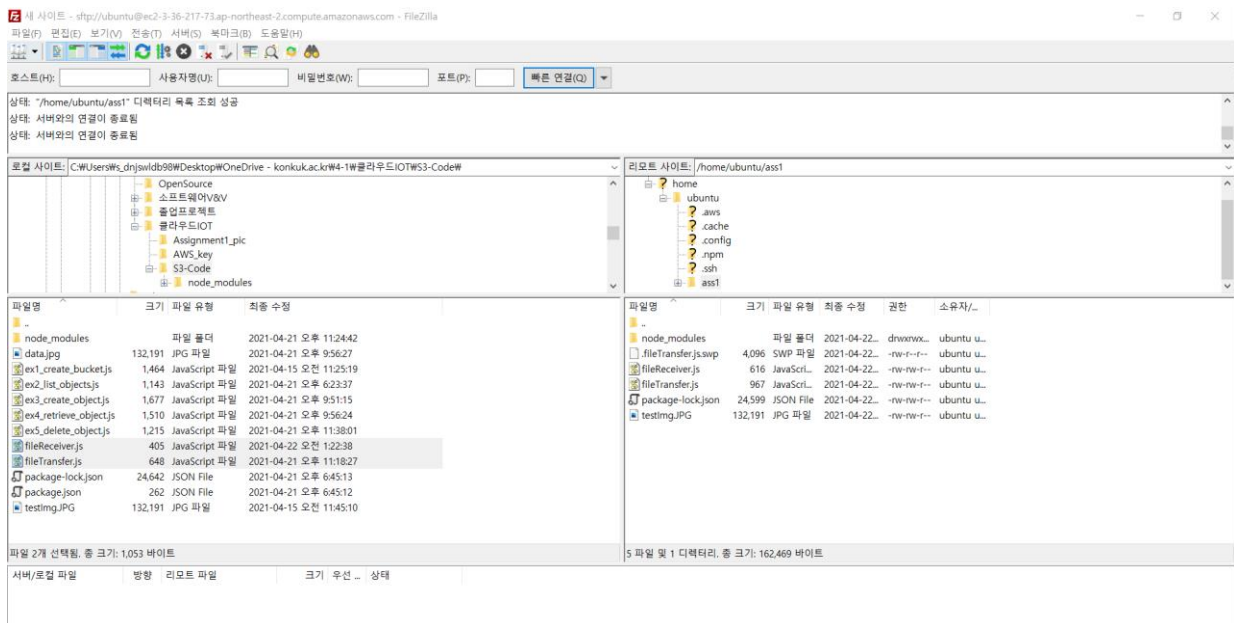
```

### 3. steps

- 1) ec2 인스턴스와 연결한 우분투 서버에 ass1 폴더를 만든 후 fileTransfer.js 와 fileReceiver.js를 vi 에디터를 활용하여 작성한다.
- 2) 같은 폴더 안에 testTxt.txt 또한 vi를 이용하여 작성한다.



(\*\* 혹은 파일질라를 사용하여 visual studio code 등의 ide에서 작성한 코드를 서버로 옮길 수도 있다. )



3) 작성한 두 파일 fileTransfer.js와 fileReceiver.js를 실행하고 결과를 확인한다.

```

ubuntu@ip-172-31-43-188: ~/ass1
System information as of Wed Apr 21 17:13:12 UTC 2021

System load:  0.0          Processes:           116
Usage of /:   30.3% of 7.69GB Users logged in:       1
Memory usage: 30%          IPv4 address for eth0: 172.31.43.188
Swap usage:   0%

* Pure upstream Kubernetes 1.21, smallest, simplest cluster ops!

https://microk8s.io/

8 updates can be installed immediately.
4 of these updates are security updates.
To see these additional updates run: apt list --upgradable

*** System restart required ***
last login: Wed Apr 21 13:58:53 2021 from 211.44.166.13
ubuntu@ip-172-31-43-188:~$ cd ass1/
ubuntu@ip-172-31-43-188:~/ass1$ node fileTransfer.js
FILESENDER START
client - [object Object]
CONNECTION
  
```



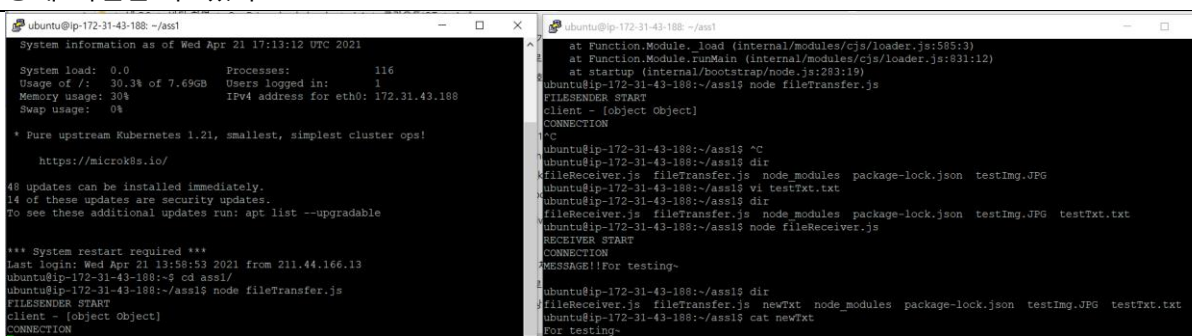
3-1) node fileTransfer.js를 통해 실행시켜서 Topic assignment1으로 파일 testTxt.txt를 보낸다.

```
ubuntu@ip-172-31-43-188:~/ass1$ vi testTxt.txt
ubuntu@ip-172-31-43-188:~/ass1$ dir
fileReceiver.js fileTransfer.js node_modules package-lock.json testImg.JPG testTxt.txt
ubuntu@ip-172-31-43-188:~/ass1$ node fileReceiver.js
RECEIVER START
CONNECTION
MESSAGE!!For testing~

ubuntu@ip-172-31-43-188:~/ass1$ dir
fileReceiver.js fileTransfer.js newTxt node_modules package-lock.json testImg.JPG testTxt.txt
ubuntu@ip-172-31-43-188:~/ass1$ cat newTxt
For testing~
ubuntu@ip-172-31-43-188:~/ass1$
```

3-2) node fileReceiver.js를 통해 실행시켜서 Topic assignment1을 구독하고 기다린다.

fileTransfer.js 가 실행되면 메시지가 넘어오게 되고, newTxt가 옳게 전송되었음을 cat newTxt를 통해 확인할 수 있다.



```
ubuntu@ip-172-31-43-188:~/ass1$ node fileTransfer.js
FILESENDER START
client - [object Object]
CONNECTION
MESSAGE!!For testing~

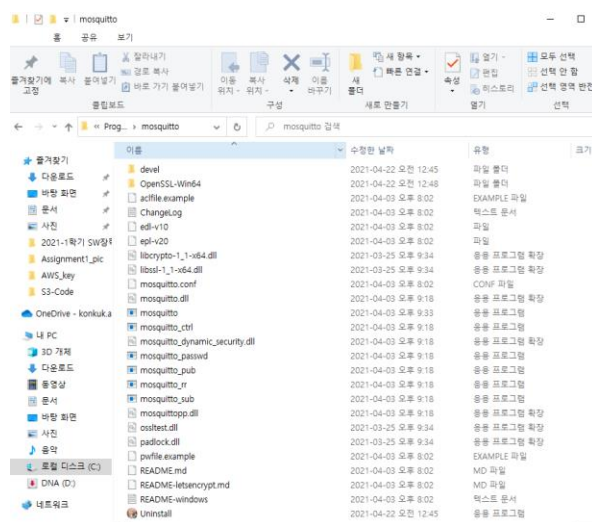
ubuntu@ip-172-31-43-188:~/ass1$ node fileReceiver.js
RECEIVER START
CONNECTION
MESSAGE!!For testing~

ubuntu@ip-172-31-43-188:~/ass1$ cat newTxt
For testing~
```

3-3) 최종결과 (Left : fileTransfer.js, Right : fileReceiver.js)

\*\*windows에서도 정상 동작

(windows에 mosquitto 깔아서 정상 동작 확인함)



(윈도우에 mosquitto 까는 과정 중 하나)

```
C:\Users\#s_dnjswldb98\Desktop\#OneDrive - konkuk.ac.kr\#4-1\클라우드IoT\#S3-Code>node fileTransfer.js
FILESENDER START
client - [object Object]
CONNECTION
MESSAGE!!For testing~
assignment1 : send
```

## node fileTransfer.js 실행

```
C:\Users\ws_dnjsw\OneDrive - konkuk.ac.kr\4-1\클라우드IoT\S3-Code>node fileReceiver.js
RECEIVER START
CONNECTION
MESSAGE!!For testing...
```

## node fileReceiver.js 실행

내 PC > 바탕 화면 > OneDrive - konkuk.ac.kr > 4-1 > 클라우드IoT > S3-Code

이름	상태	수정한 날짜	유형	크기
node_modules	✓	2021-04-21 오후 11:24	파일 폴더	
data	✓	2021-04-21 오후 9:56	JPG 파일	130KB
ex1_create_bucket	✓	2021-04-15 오전 11:25	JavaScript 파일	2KB
ex2_list_objects	✓	2021-04-21 오후 6:23	JavaScript 파일	2KB
ex3_create_object	✓	2021-04-21 오후 9:51	JavaScript 파일	2KB
ex4_retrieve_object	✓	2021-04-21 오후 9:56	JavaScript 파일	2KB
ex5_delete_object	✓	2021-04-21 오후 11:38	JavaScript 파일	2KB
fileReceiver	✓	2021-04-22 오전 2:05	JavaScript 파일	1KB
fileTransfer	✓	2021-04-22 오전 2:07	JavaScript 파일	1KB
newImg	✓	2021-04-22 오전 1:47	파일	1KB
newTxt	✓	2021-04-22 오전 2:07	파일	1KB
package	✓	2021-04-21 오후 6:45	JSON File	1KB
package-lock	✓	2021-04-21 오후 6:45	JSON File	25KB
testimg	✓	2021-04-21 오후 9:56	JPG 파일	130KB
testTxt	✓	2021-04-22 오전 1:48	텍스트 문서	1KB

## 실행 결과 : newTxt 생성

newTxt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

For testing...

newTxt 파일을 열어보면 testTxt와 같은 파일임을 알 수 있다.