**지능형 정보감시 플랫폼4차년도**

**MQTT 설치/테스트**

|  |
| --- |
| **작성일 : 2020.06.08** |

**목 차**

[**1. MQTT 선정 3**](#_30j0zll)

[**2. 설치 3**](#_1fob9te)

[**2.1 host 디렉토리에 로그 생성하도록 실행. 3**](#_3znysh7)

[**2.2 기본 실행 3**](#_2et92p0)

[**3. MQTT explorer 4**](#_tyjcwt)

[**3.1 설치 4**](#_3dy6vkm)

[**3.2 모니터링 4**](#_1t3h5sf)

[**4. 파이썬 Publish / Subscribe 예제 5**](#_4d34og8)

[**4.1 publish.py 5**](#_2s8eyo1)

[**4.2 subscribe.py 6**](#_17dp8vu)

[**5. 바이너리 데이터 송/수신 검토(테스트) 8**](#_3rdcrjn)

# MQTT 선정

Mosquitto MQTT Broker 를 사용하여 구축.

# 설치

- 리눅스에서, docker 를 이용한 설치.

## host 디렉토리에 로그 생성하도록 실행.

# host 에 디렉토리 생성.

mkdir config data log

chmod 777 config data log

$ cat config/mosquitto.conf

persistence true

persistence\_location /mosquitto/data/

log\_dest file /mosquitto/log/mosquitto.log

# 도커 실행. ( 로그를, host에 남기기)

docker run -it --name mos -p 1883:1883 -p 9001:9001 -v $PWD/config:/mosquitto/config -v $PWD/data:/mosquitto/data -v $PWD/log:/mosquitto/log eclipse-mosquitto

#위의 9001 포트는, MQTT over Websockets이 사용하는 포트이다.

## docker-compose 사용

docker-compose.yaml 파일

|  |
| --- |
| version: '3'  services:  mqtt:  container\_name: mqtt  image: eclipse-mosquitto  volumes:  - ./config:/mosquitto/config  - ./data:/mosquitto/data  - ./log:/mosquitto/log  ports:  - 1883:1883 |

시작 : docker-compose up

종료 : docker-compose down

## 기본 실행

docker run -it --name mos -p 1883:1883 eclipse-mosquitto

# MQTT explorer

## 설치

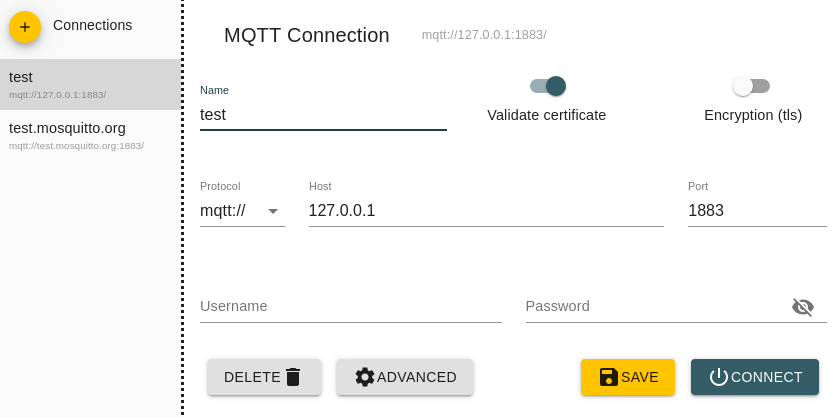
sudo apt update

sudo apt install snapd

sudo snap install mqtt-explorer

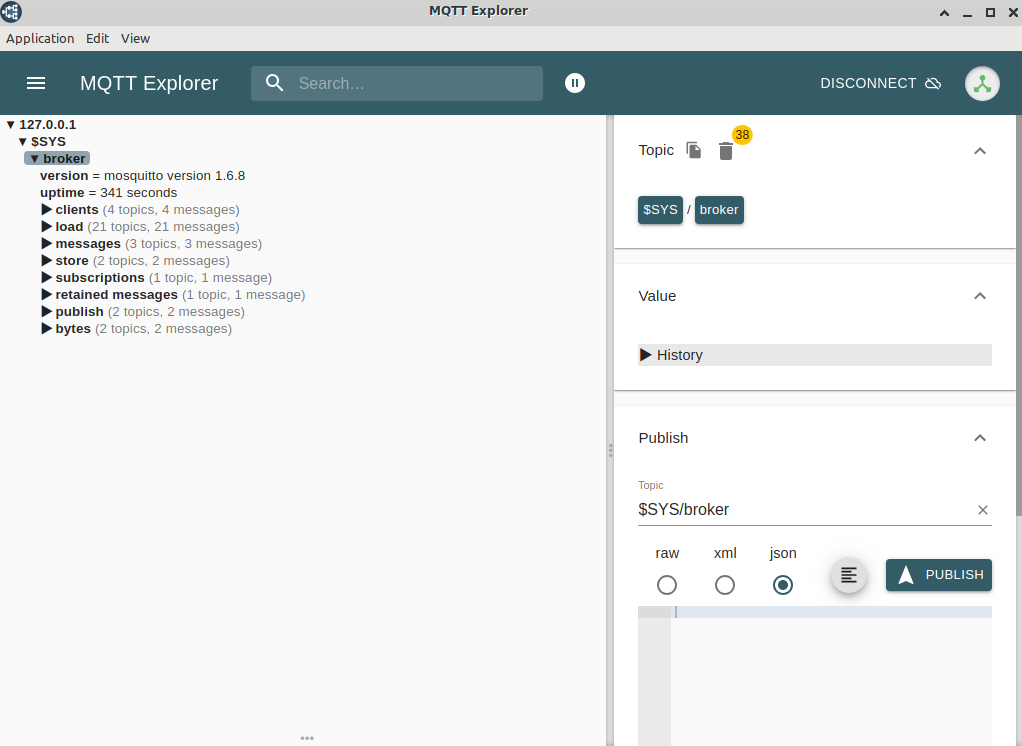
## 모니터링

MQTT Explorer 실행.



127.0.0.1을 입력하고, connect 버튼을 클릭한다.

MQTT 모니터링이 가능.



# 파이썬 Publish / Subscribe 예제

## publish.py

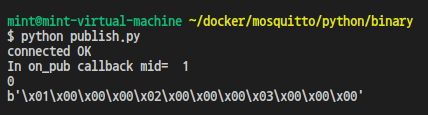
|  |
| --- |
| import paho.mqtt.client as mqtt  import json  def on\_connect(client, userdata, flags, rc):  if rc == 0:  print("connected OK")  else:  print("Bad connection Returned code=", rc)  def on\_disconnect(client, userdata, flags, rc=0):  print(str(rc))  def on\_publish(client, userdata, mid):  print("In on\_pub callback mid= ", mid)  # 새로운 클라이언트 생성  client = mqtt.Client()  # 콜백 함수 설정 on\_connect(브로커에 접속), on\_disconnect(브로커에 접속중료), on\_publish(메세지 발행)  client.on\_connect = on\_connect  client.on\_disconnect = on\_disconnect  client.on\_publish = on\_publish  # address : localhost, port: 1883 에 연결  client.connect('localhost', 1883)  client.loop\_start()  # common topic 으로 메세지 발행  client.publish('common', json.dumps({"success": "ok"}), 1)  client.loop\_stop()  # 연결 종료  client.disconnect() |

## subscribe.py

|  |
| --- |
| import paho.mqtt.client as mqtt  def on\_connect(client, userdata, flags, rc):  if rc == 0:  print("connected OK")  else:  print("Bad connection Returned code=", rc)  def on\_disconnect(client, userdata, flags, rc=0):  print(str(rc))  def on\_subscribe(client, userdata, mid, granted\_qos):  print("subscribed: " + str(mid) + " " + str(granted\_qos))  def on\_message(client, userdata, msg):  print(str(msg.payload.decode("utf-8")))  # 새로운 클라이언트 생성  client = mqtt.Client()  # 콜백 함수 설정 on\_connect(브로커에 접속), on\_disconnect(브로커에 접속중료), on\_subscribe(topic 구독),  # on\_message(발행된 메세지가 들어왔을 때)  client.on\_connect = on\_connect  client.on\_disconnect = on\_disconnect  client.on\_subscribe = on\_subscribe  client.on\_message = on\_message  # address : localhost, port: 1883 에 연결  client.connect('localhost', 1883)  # common topic 으로 메세지 발행  client.subscribe('common', 1)  client.loop\_forever() |

## 바이너리 데이터 송/수신 검토(테스트)

전송



수신

