

강원혁신플랫폼

리눅스프로그래밍

MQTT 개요





낮은 대역폭, 높은 대기 시간 또는 불안정한 네트워크 조건에서 장치 간의 효율적인 통신을 위해 설계된 경량 메시징 프로토콜로 실시간 데이터 교환을 위해 사물 인터넷(IoT) 애플리케이션에 널리 사용되는 프로토콜은 무엇인가요?

MQTT
(Message Queuing Telemetry Transport)





학습 내용

- 1 MQTT란?
- 2 Mosquitto 브로커 설치

학습 목표

- MQTT 특징을 설명할 수 있다.
- Mosquitto 브로커를 설치할 수 있다.

강원혁신플랫폼

리눅스프로그래밍



MQTT란?





MQTT

Message Queuing Telemetry Transport

낮은 대역폭, 높은 대기 시간 또는 불안정한 네트워크 조건에서 장치
간의 효율적인 통신을 위해 설계된 경량 메시징 프로토콜
실시간 데이터 교환을 위해
사물 인터넷(IoT) 애플리케이션에 널리 사용



MQTT

주요 기능

게시
(Publish)

구독
(Subscribe)

표준화된 프로토콜

- ◆ MQTT는 개방적이고 표준화된 프로토콜이므로 다양한 플랫폼과 장치에서 상호 운용 가능

MQTT 브로커

- ◆ Mosquitto, Eclipse Mosquitto, HiveMQ 및 AWS IoT Core와 같은 자체 호스팅 및 클라우드 기반의 다양한 MQTT 브로커를 사용



MQTT 프로토콜 비교

MQTT vs. HTTP

MQTT

- ◆ 리소스가 제한된 장치 및 저대역폭 네트워크에 적합

HTTP

- ◆ 헤더 및 상태 비저장으로 인해 더 많은 오버헤드가 수반되는 요청-응답 프로토콜
- ◆ 리소스가 제한된 IoT 장치에는 적합하지 않음



MQTT 프로토콜 비교

MQTT vs. CoAP (Constrained Application Protocol)

MQTT

- ◆ IoT 애플리케이션용으로 설계
- ◆ QoS 수준, 보존된 메시지 및 영구 세션을 포함하여 보다 광범위한 기능 집합을 제공

CoAP

- ◆ IoT 애플리케이션용으로 설계
- ◆ 전력이 매우 낮고 처리 능력이 제한된 제한된 장치에 최적화
- ◆ 단순성과 리소스 효율성에 중점



MQTT 프로토콜 비교

MQTT vs. WebSocket

MQTT

- ◆ IoT 통신을 위한 메시징 프로토콜
- ◆ 낮은 오버헤드로 효율적인 발행/구독 메시징을 위해 설계

WebSocket

- ◆ 단일 장기 연결을 통해 클라이언트와 서버 간의 전이중 통신을 위한 프로토콜
- ◆ 웹 브라우저와 서버 간의 실시간 양방향 통신

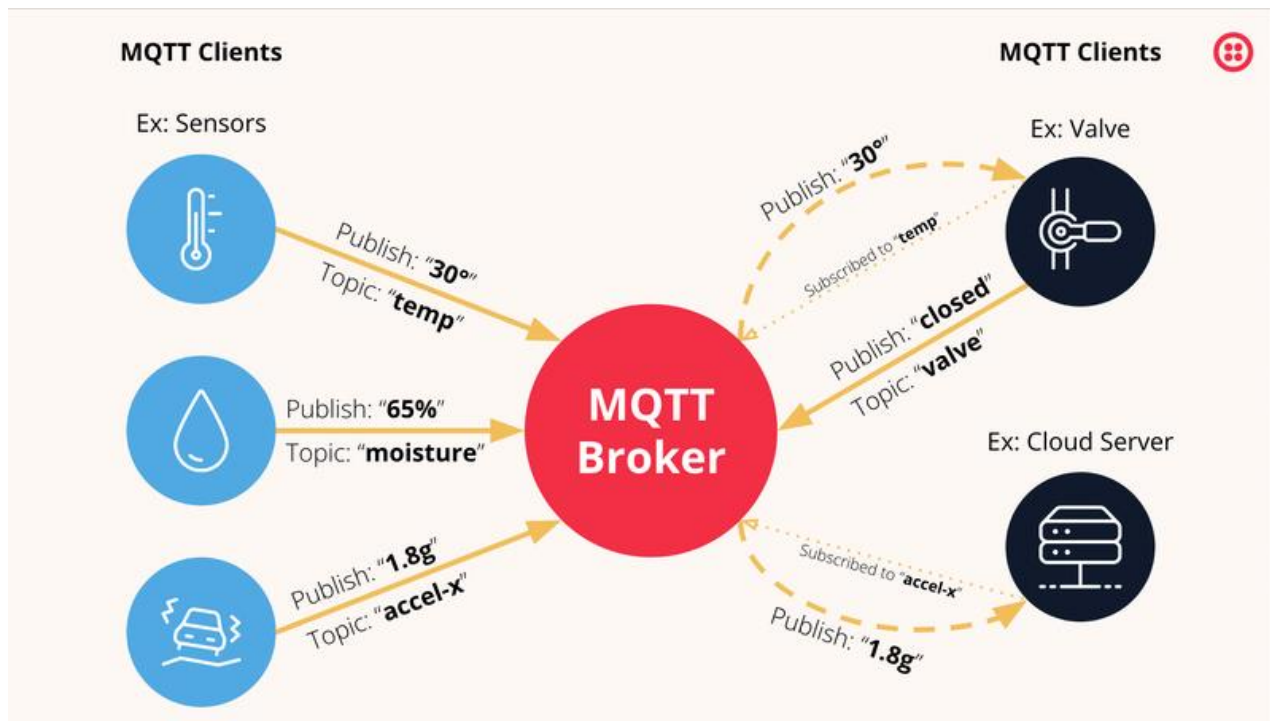


MQTT 프로토콜 비교

- ❏ AMQP (Advanced Message Queuing Protocol)
- ❏ DDS (Data Distribution Service)



MQTT publish/subscribe



강원혁신플랫폼

리눅스프로그래밍



Mosquitto 브로커 설치





설치

```
$ sudo apt-get install mosquitto mosquitto-clients
```

서비스 시작

```
$ sudo systemctl start mosquitto.service
```

PS 확인

```
$ ps -ef | grep mosq
```

```
mosquit+ 1139    1 0 20:25 ?        00:00:00 /usr/sbin/mosquito -c  
/etc/mosquitto/mosquitto.conf
```



테스트

```
$ mosquitto_sub -h localhost -t test (Hello, world! 응답 대기)
```

```
Hello, world!
```

```
$ mosquitto_pub -h localhost -t test -m "Hello, world!"(다른 터미널 창에서)
```



📦 myuser 사용자 생성, 비밀번호 설정

```
$ sudo mosquitto_passwd -c /etc/mosquitto/passwd myuser
```

📦 설정 파일에 추가

```
$ sudo nano /etc/mosquitto/mosquitto.conf  
allow_anonymous false  
password_file /etc/mosquitto/passwd
```

📦 서비스 재시작

```
$ sudo systemctl restart mosquitto.service
```




📦 패스워드 사용 연결

```
$ mosquitto_sub -h localhost -t test -u myuser -P mypassword  
Hello, world!
```

```
$ mosquitto_pub -h localhost -t test -m "Hello, world!"(다른 터미널 창에서)
```



MQTT connection settings

Host: broker.hivemq.com

TCP Port: 1883

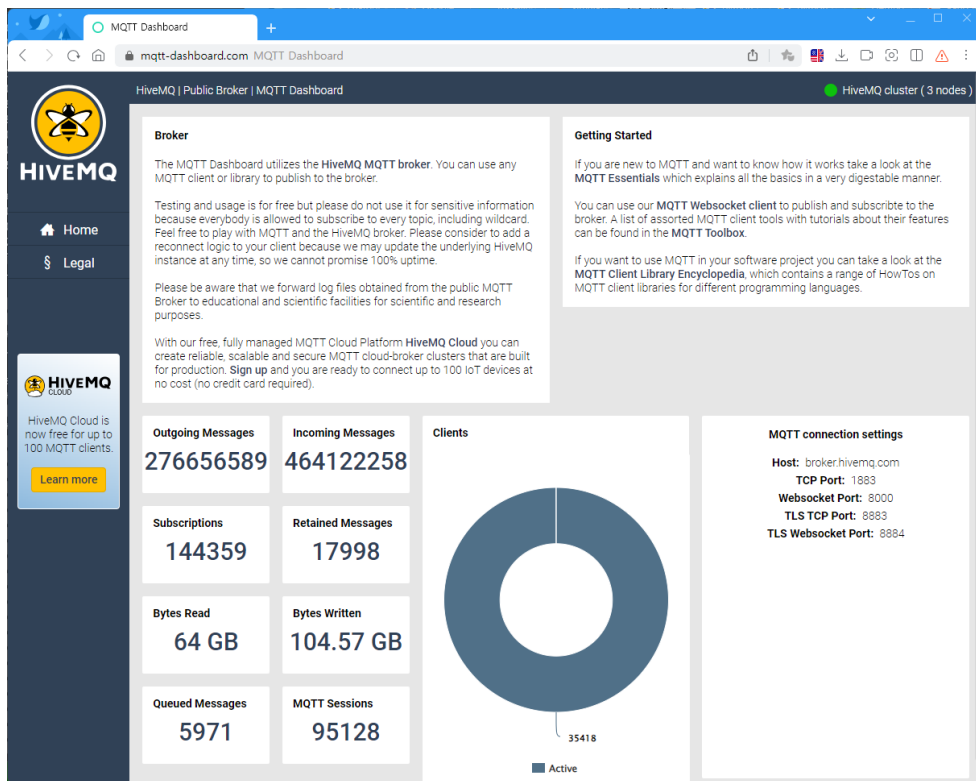
Websocket Port: 8000

TLS TCP Port: 8883

TLS Websocket Port: 8884



클라우드 기반 MQTT 브로커





```
(.venv) $ cat mqtt_pub.py  
import paho.mqtt.client as mqtt  
import time
```

```
# MQTT Broker configuration
```

```
broker_address = '127.0.0.1'
```

```
broker_port = 1883
```

```
topic = 'my/topic'
```

```
# Create an MQTT client instance
```

```
client = mqtt.Client()
```



```
# Connect to the MQTT broker
```

```
client.connect(broker_address, broker_port)
```

```
# Publish a msg to the topic every 1 second
```

```
while True:
```

```
    message = f"Hello, MQTT! Time: {time.strftime('%H:%M:%S')}}"
```

```
    client.publish(topic, message)
```

```
    print(f"Published: {message}")
```

```
    time.sleep(1)
```



```
(.venv) $ cat mqtt_sub.py  
import paho.mqtt.client as mqtt
```

```
# MQTT Broker configuration /* 생략 */
```

```
def on_connect(client, userdata, flags, rc):  
    print("Connected to MQTT broker")
```

```
# Subscribe to the topic  
client.subscribe(topic)
```



```
def on_message(client, userdata, msg):  
    print(f"Received: {msg.payload.decode()}")
```

```
client = mqtt.Client()
```

```
client.on_connect = on_connect
```

```
client.on_message = on_message
```

```
client.connect(broker_address, broker_port)
```

```
client.loop_forever()
```



실행결과

```
(.venv) $ python mqtt_pub.py
```

Published: Hello, MQTT! Time: 11:05:43

Published: Hello, MQTT! Time: 11:05:44

Published: Hello, MQTT! Time: 11:05:45

Published: Hello, MQTT! Time: 11:05:46

Published: Hello, MQTT! Time: 11:05:47

Published: Hello, MQTT! Time: 11:05:48

Published: Hello, MQTT! Time: 11:05:49

Published: Hello, MQTT! Time: 11:05:50

Published: Hello, MQTT! Time: 11:05:51

Published: Hello, MQTT! Time: 11:05:52



실행결과

```
(.venv) $ python mqtt_sub.py
```

```
Connected to MQTT broker
```

```
Received: Hello, MQTT! Time: 11:05:44
```

```
Received: Hello, MQTT! Time: 11:05:45
```

```
Received: Hello, MQTT! Time: 11:05:46
```

```
Received: Hello, MQTT! Time: 11:05:47
```

```
Received: Hello, MQTT! Time: 11:05:48
```

```
Received: Hello, MQTT! Time: 11:05:49
```

```
Received: Hello, MQTT! Time: 11:05:50
```

```
Received: Hello, MQTT! Time: 11:05:51
```



01 • MQTT란?

02 • Mosquitto 브로커 설치