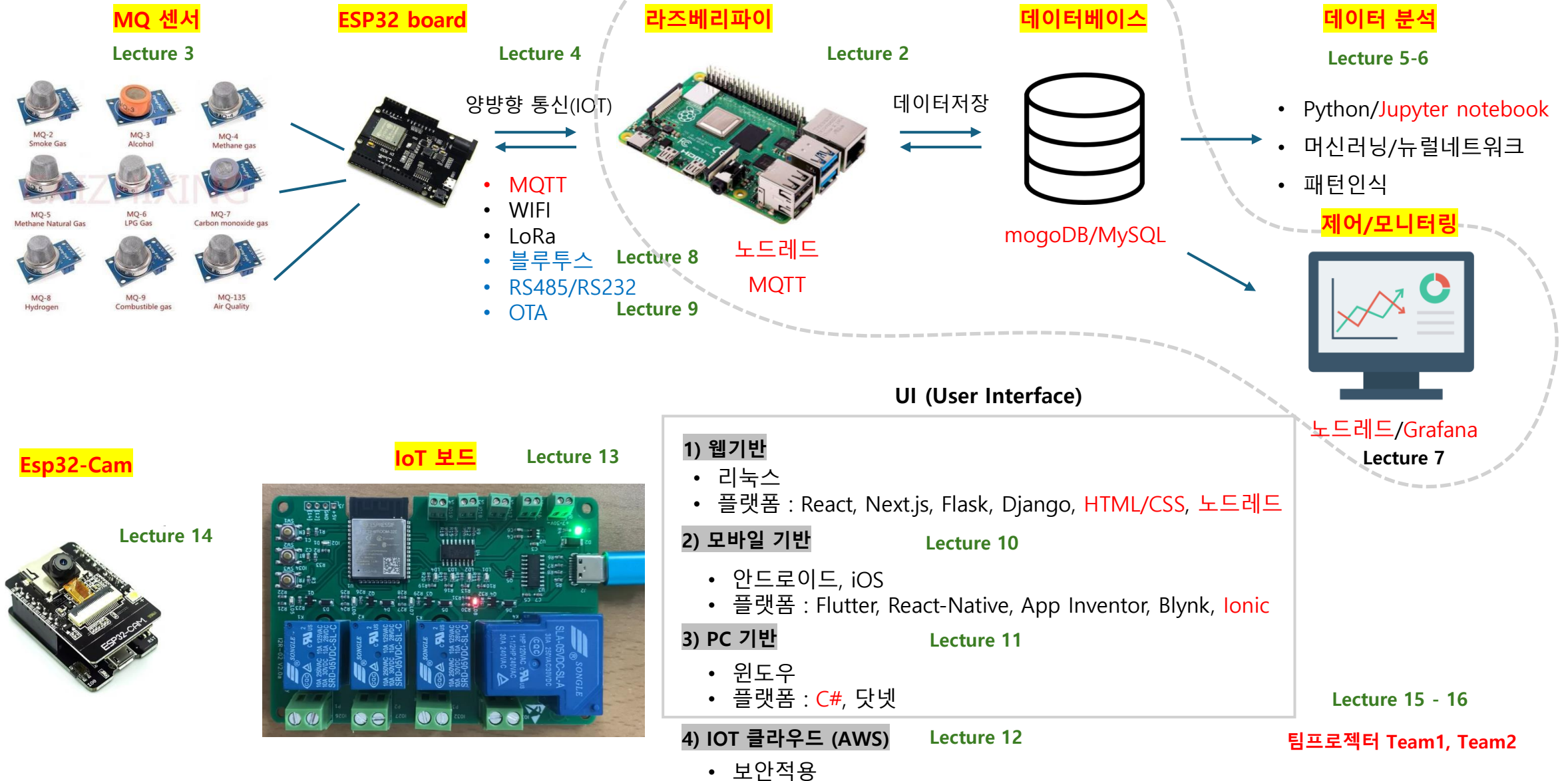


강의 Flow & Plan



Docker 설치

System update

```
sudo apt-get update  
sudo apt-get upgrade
```

docker setup 파일 다운로드

```
curl -fsSL https://get.docker.com -o get-docker.sh
```

Shell 실행

```
sudo sh get-docker.sh
```

권한 부여

```
sudo usermod -aG docker ${USER}
```

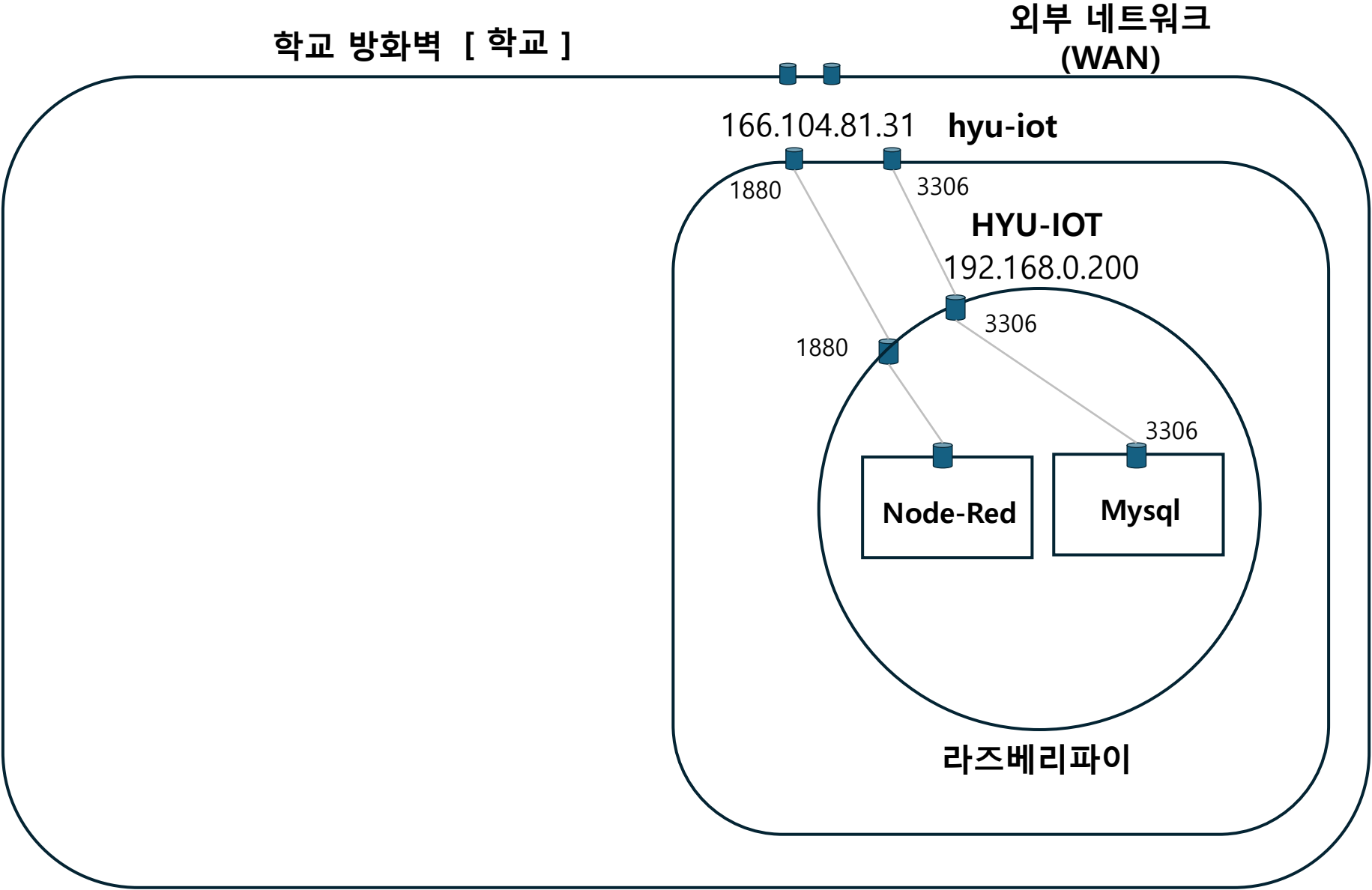
권한 확인

```
groups ${user}
```

부팅시 docker 자동 실행

```
sudo systemctl enable docker
```

네트워크 구조



Container Application in Docker

<https://www.docker.com/products/docker-hub/>



Node-Red 서버

```
$ docker run -itd -p 1880:1880 -v /home/iot/nodered_1880:/data --name  
nodered_1880 nodered/node-red:latest
```



Mysql 서버

<https://dev.mysql.com/downloads/installer/>

```
$ docker run -itd -p 3306:3306 --name mysql_3306 -e  
MYSQL_ROOT_PASSWORD=iot1004 mysql:latest
```



mongoDB®

mongoDB 서버

```
$ docker run -itd -p 27017:27017 --name mongodb mongodb/mongodb-  
community-server:latest
```



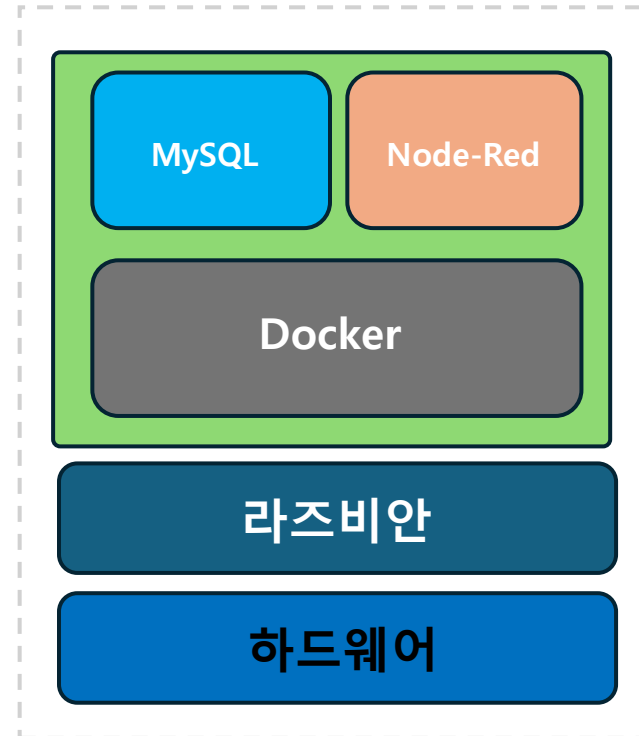
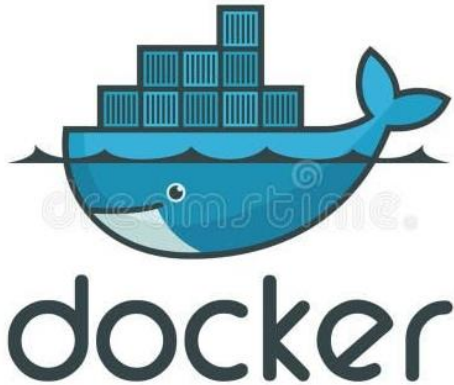
MQTT 서버

```
$ docker run -it -p 1883:1883 -p 9001:9001 -v  
mosquitto.conf:/mosquitto/config/mosquitto.conf eclipse-mosquitto
```

```

iot@HYU-IOT: ~
iot@HYU-IOT:~ $ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS
0734fcb762e0   mysql:latest   "docker-entrypoint.s..." 10 hours ago   Up 7 hours         0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp
mysql_3306
7f0c0f92c3f4   nodered/node-red  "./entrypoint.sh"       10 hours ago   Up 7 hours (healthy) 0.0.0.0:1880->1880/tcp, :::1880->1880/tcp
nodered_1880
iot@HYU-IOT:~ $

```

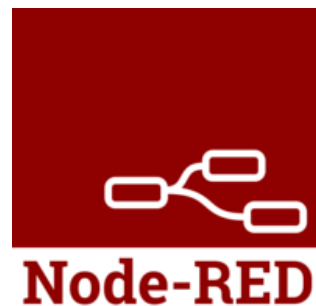


데이터 입력/출력



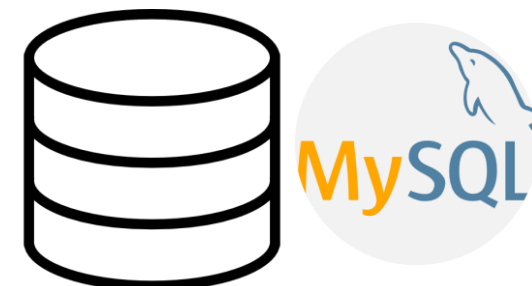
라즈베리파이

제어 컨트롤러

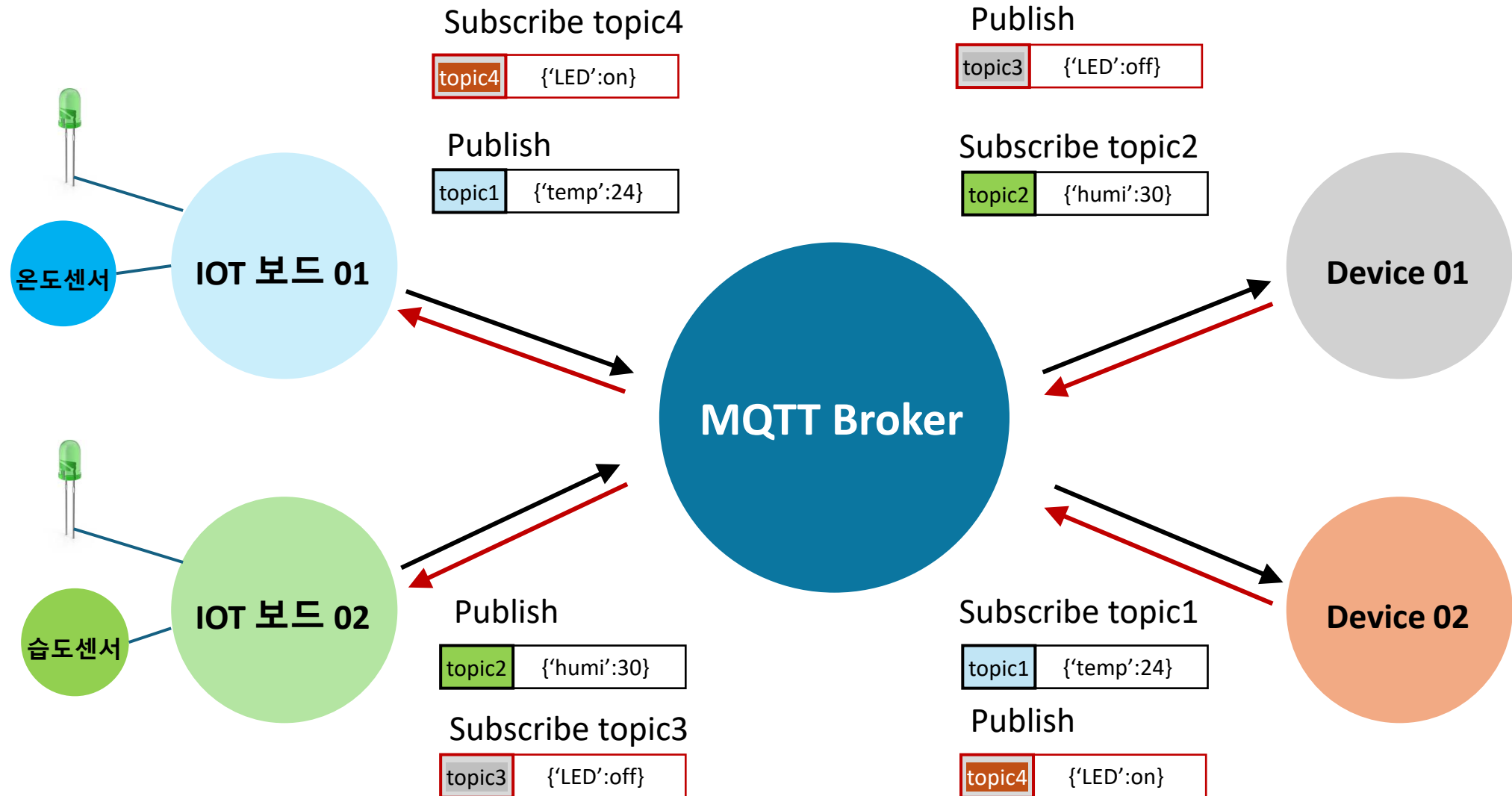


데이터저장

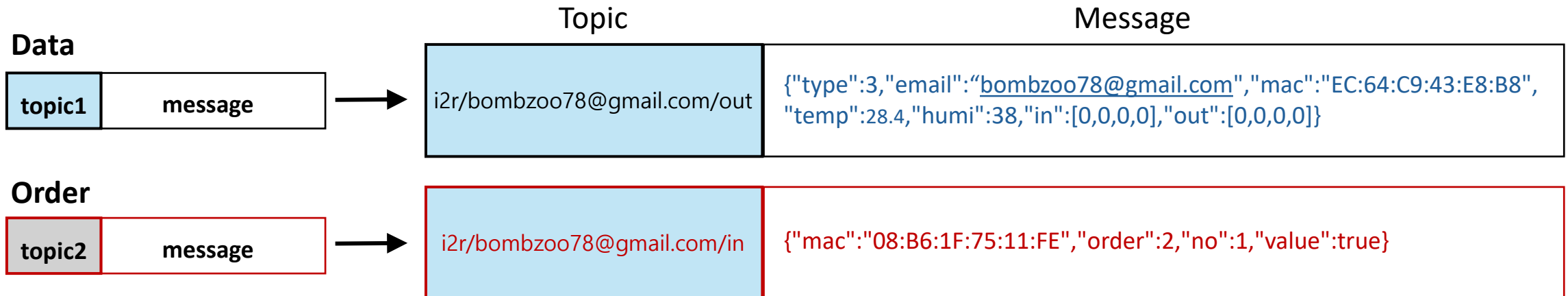
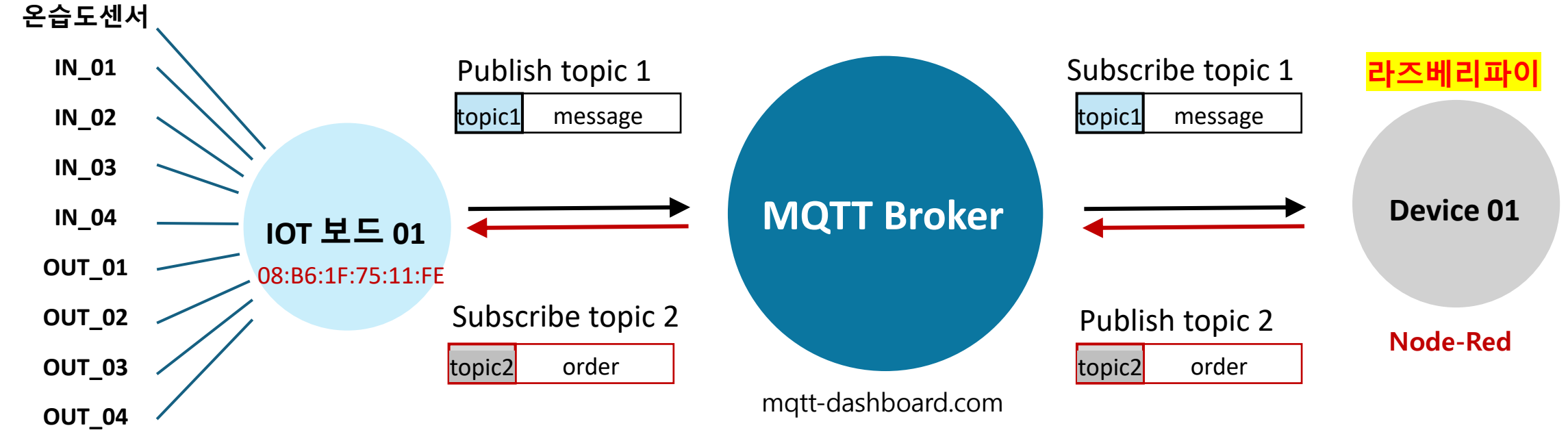
데이터저장



MQTT 통신



IOT 보드 MQTT 통신 모식도



IOT 보드



Broker

- mqtt-dashboard.com

Topic

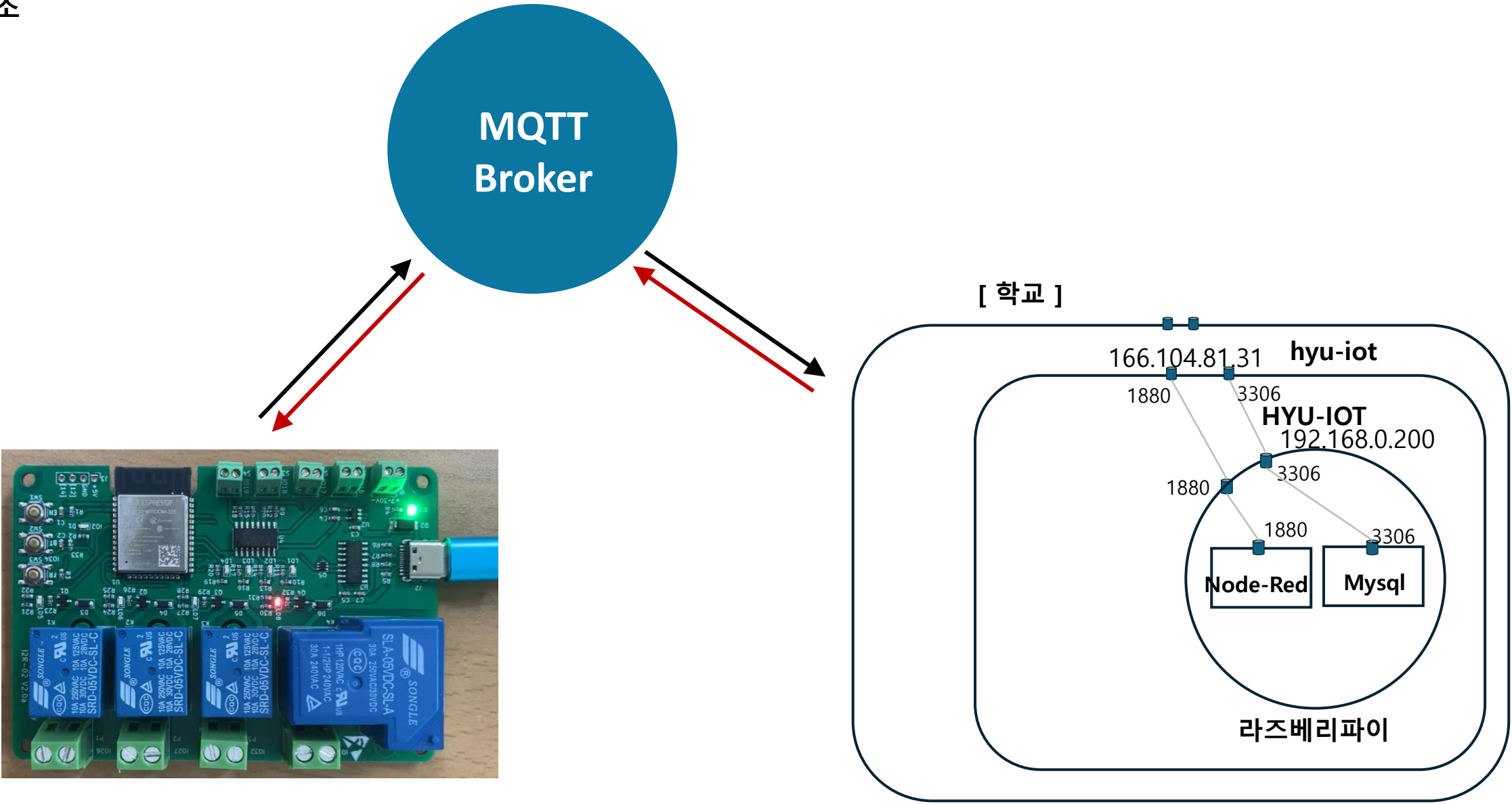
- Out : i2r/bombzoo78@gmail.com/out
- In : i2r/bombzoo78@gmail.com/in

Mac Address

"08:B6:1F:75:11:FE"

order	기능	설명 및 프로토콜
0	펌웨어 다운로드	인터넷에서 통신으로 펌웨어를 보드로 내려 받는다 {'order':0, 'fileName'='i2r-03.ino.bin'}
1	정보입력	와이파이 및 통신에 필요한 정보를 보내 보드에 기록하여 기기는 여기 정보로 통신을 연결한다. {"order":1, 'ssid':', 'password'=', 'email':'bombzoo78@gmail.com', 'mqttBroker':' mqtt-dashboard.com'}
2	핀출력	IoT PLC의 핀번호(0,1,2,3 4개)와 true/false를 보내면 릴레이가 동작한다. {"mac":"08:B6:1F:75:11:FE","order":2,"no":1,"value":true}
3	상태전송요청	요청 : {"mac":"EC:64:C9:43:E8:B8",'order':3} 응답예시 {"type":3,"email":'bombzoo78@gmail.com',"mac":"EC:64:C9:43:E8:B8", "temp":28.4,"humi":38,"in":[0,0,0,0],"out":[0,0,0,0]}

네트워크 구조



데이터 입력/출력



양방향 통신(IOT)



MQTT

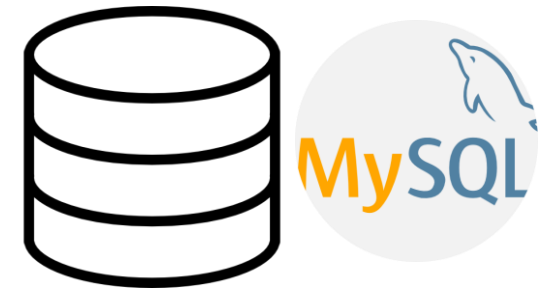
제어 컨트롤러



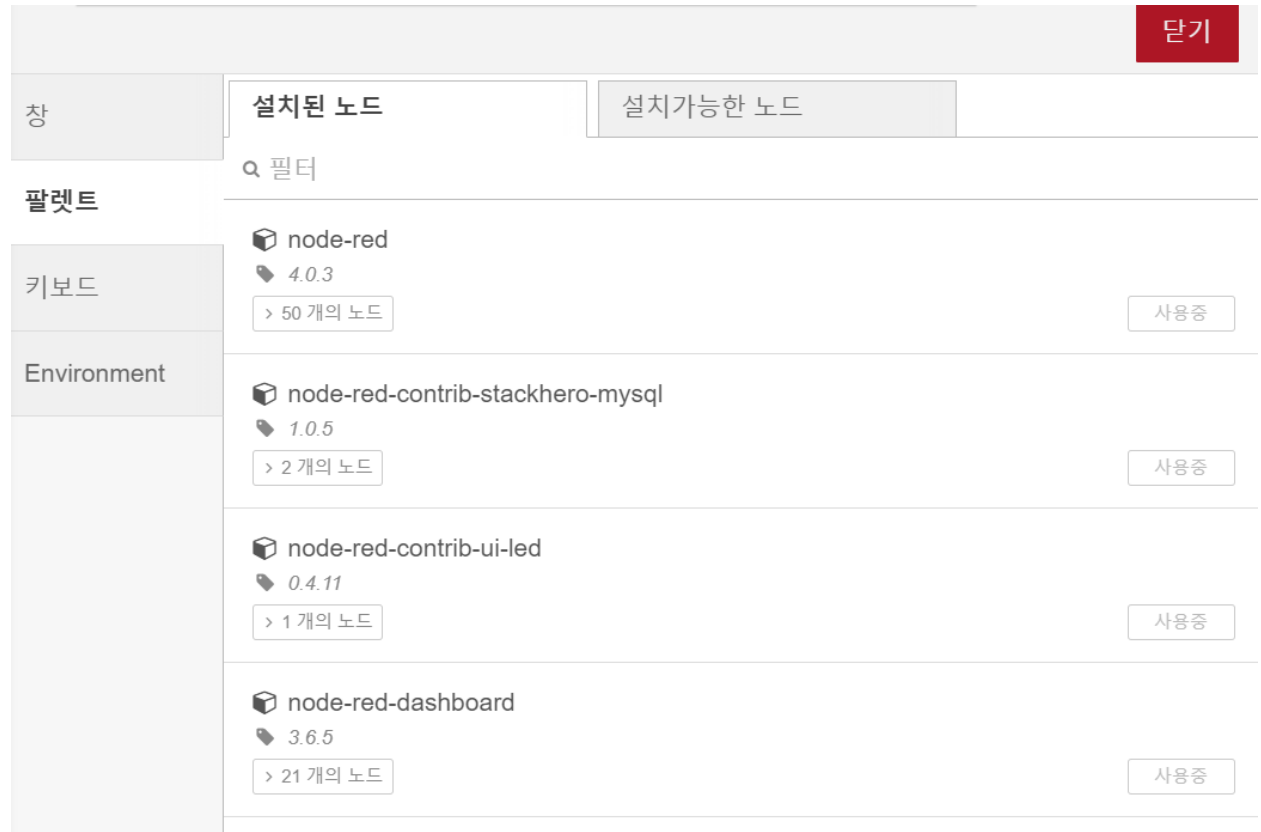
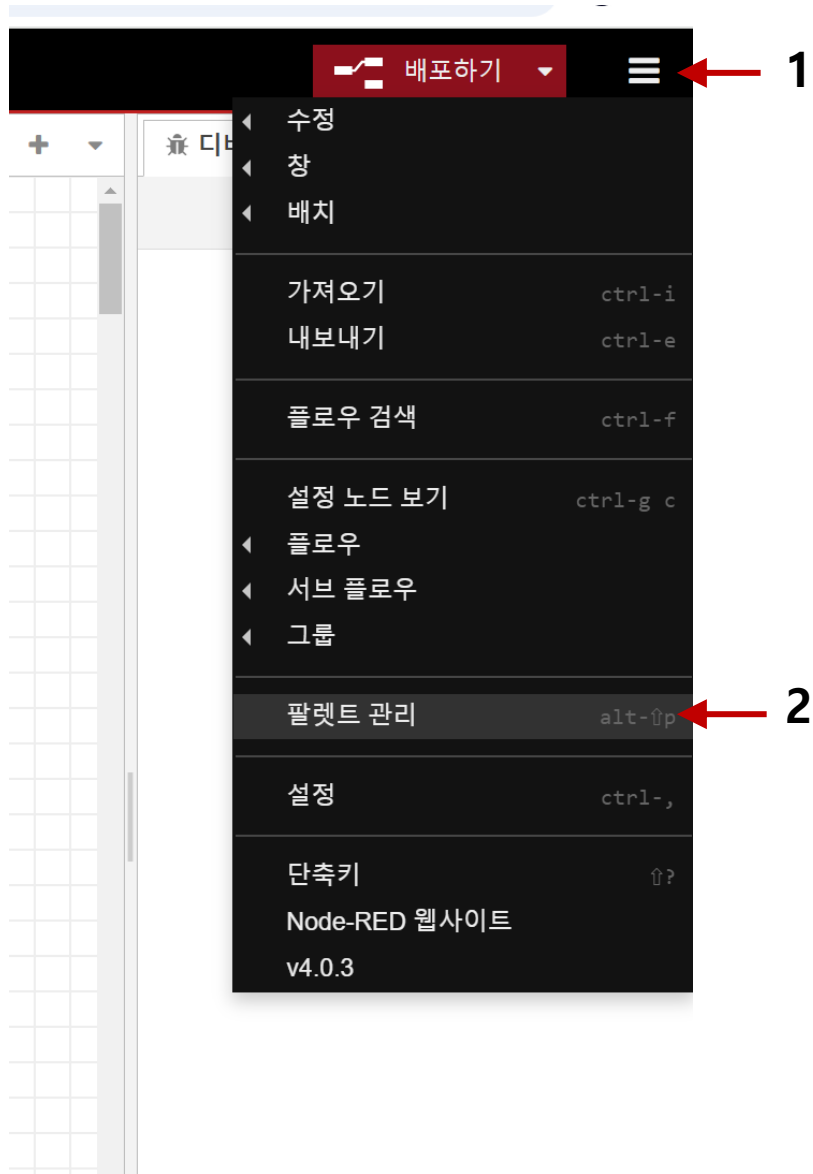
라즈베리파이

데이터저장

데이터저장



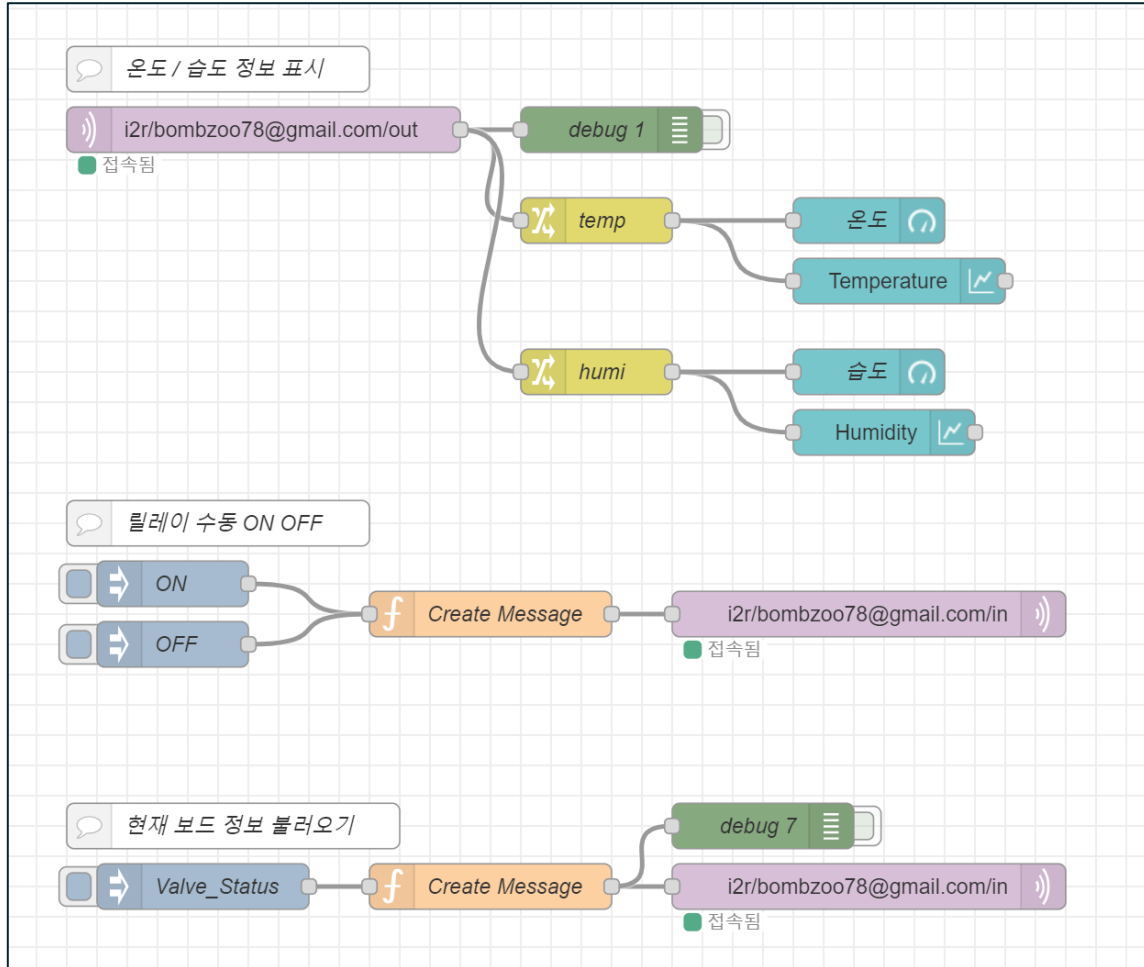
Node-Red 팔레트 설치



1. node-red-dashboard
2. node-red-contrib-ui-led
3. node-red-contrib-stackhero-mysql

Node-Red 사용법

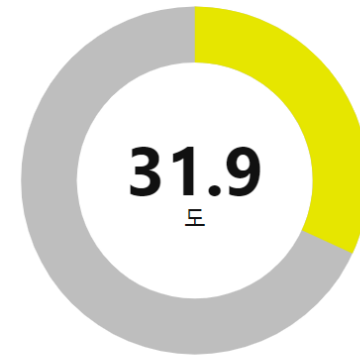
온도/습도 표시



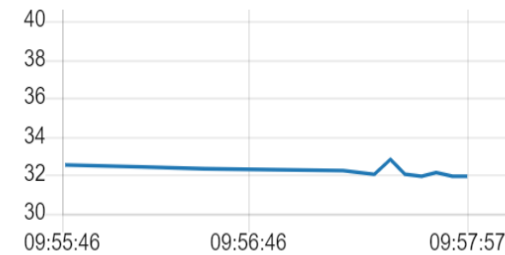
Dashboard

≡ 온도/습도

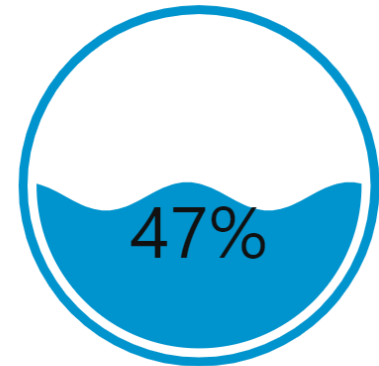
온도



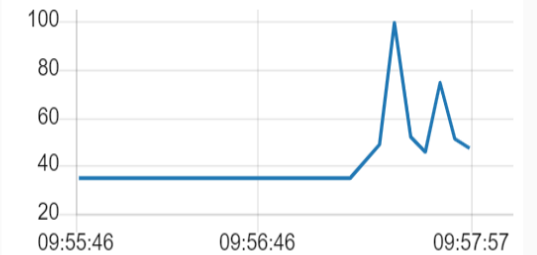
Temperature



습도

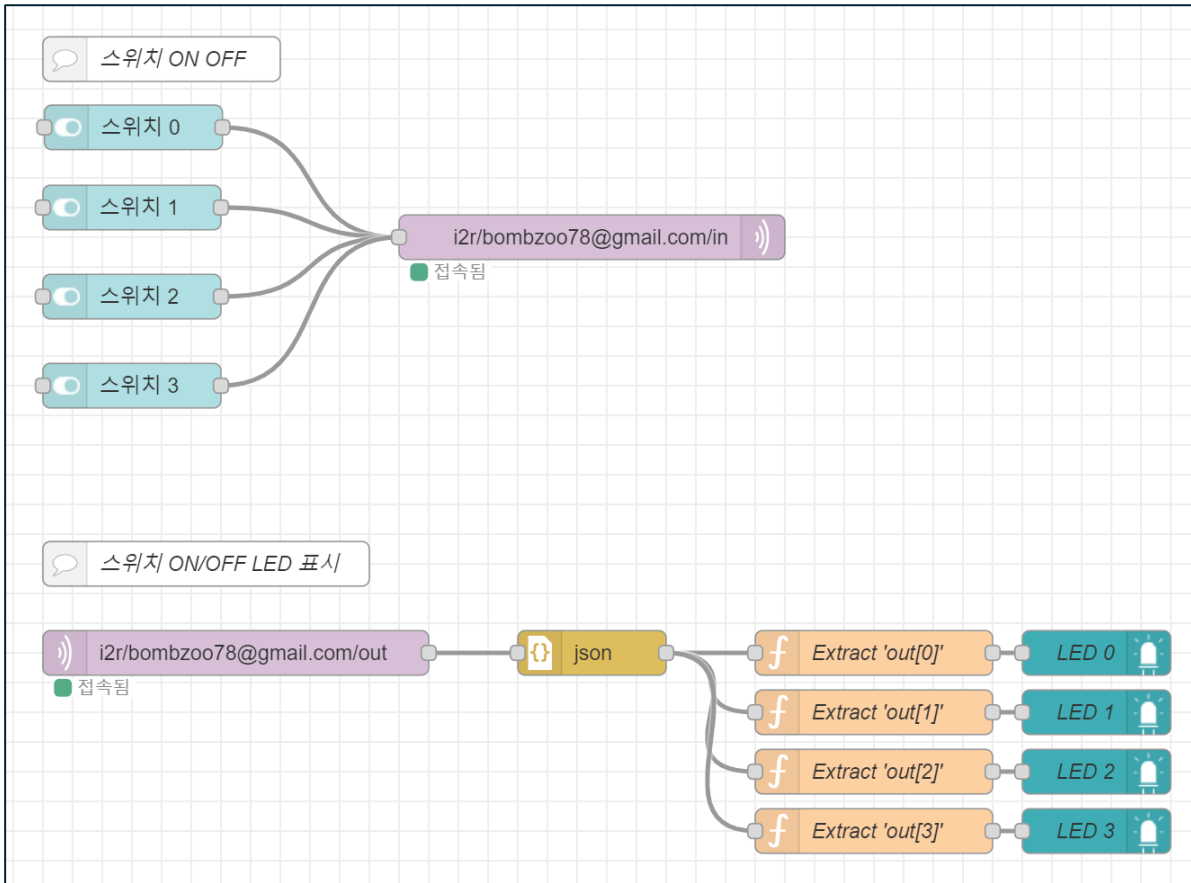


Humidity

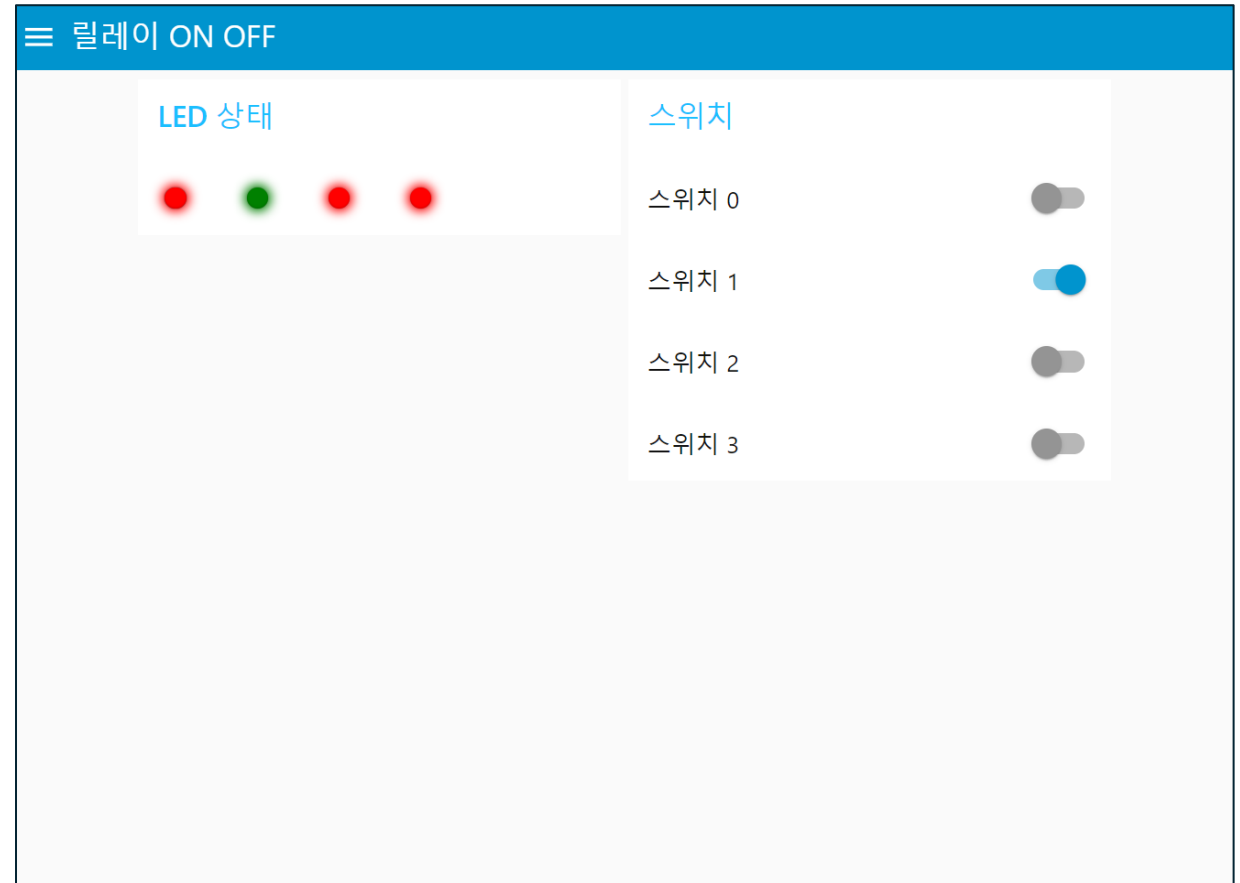


Node-Red 사용법

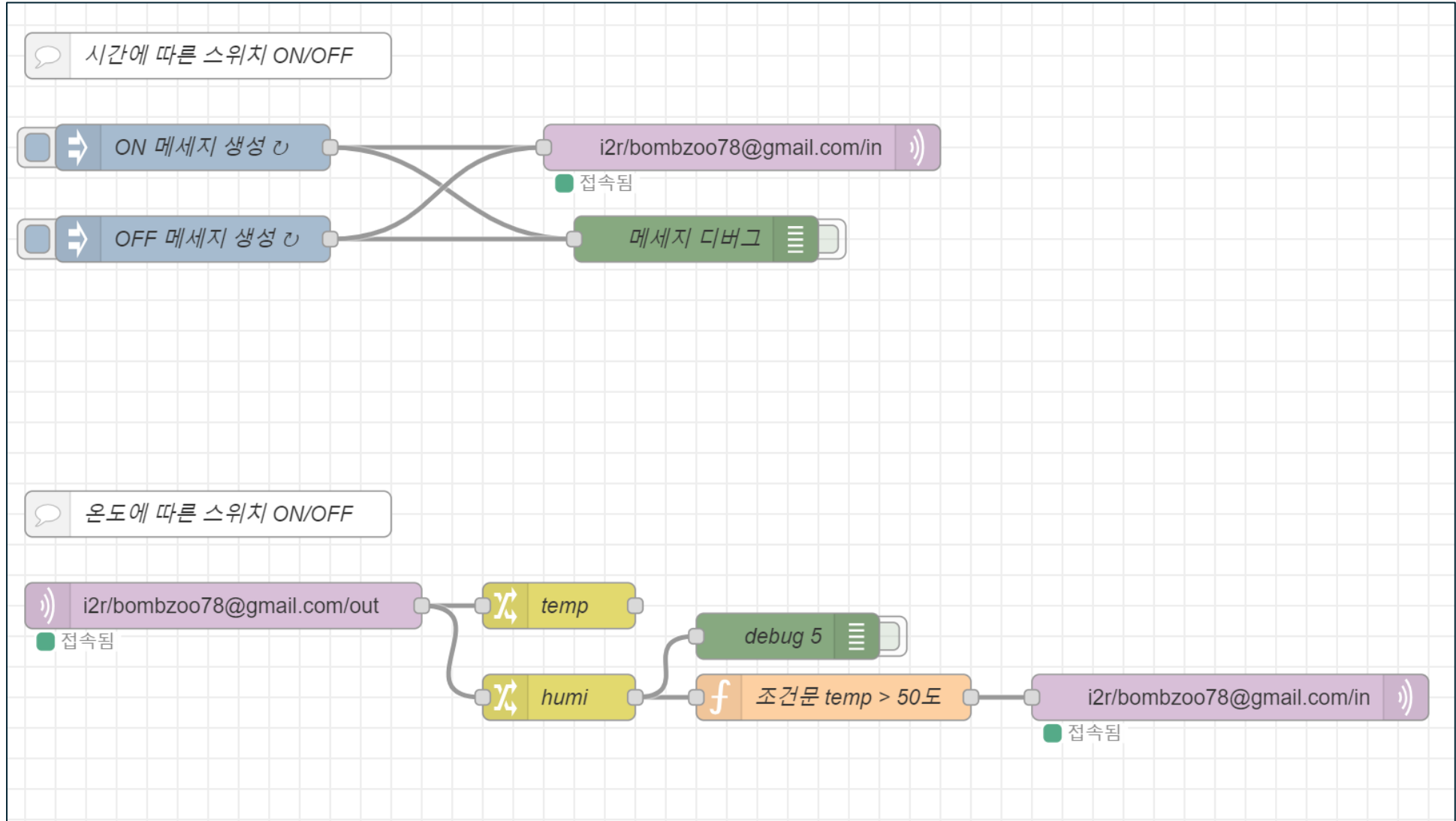
출력 스위치 on/off



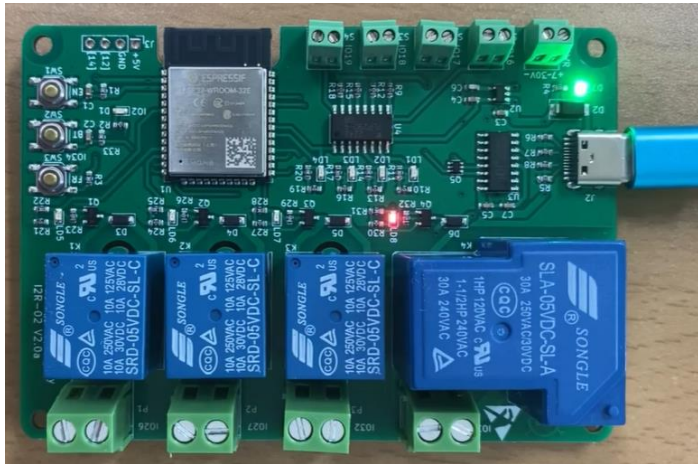
Dashboard



스케줄러



데이터 입력/출력



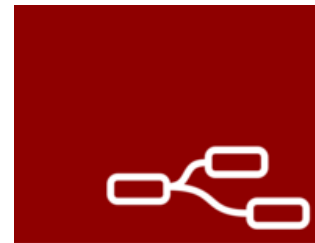
양방향 통신(IOT)



MQTT

라즈베리파이

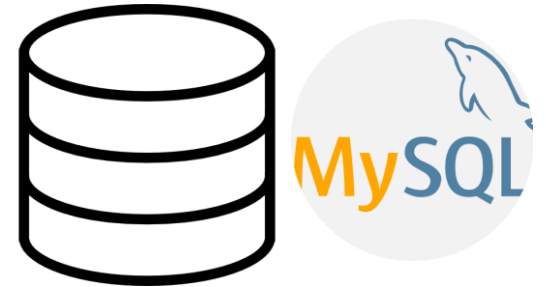
제어 컨트롤러



Node-RED

데이터저장

데이터저장



MySQL 컨테이너 접속

```
iot@HYU-IOT:~ $ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
NAMES
0734fcb762e0   mysql:latest   "docker-entrypoint.s..." 10 hours ago   Up 7 hours
mysql_3306
7f0c0f92c3f4   nodered/node-red  "./entrypoint.sh"        10 hours ago   Up 7 hours (healt
nodered_1880
iot@HYU-IOT:~ $ docker exec -it mysql_3306 /bin/bash
bash-5.1# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 151
Server version: 9.0.1 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

SQL 기본 사용법

1. 데이터 정의 언어

데이터베이스 리스트 보기

```
SHOW DATABASES;
```

데이터베이스 만들기

```
CREATE DATABASE my_database;
```

데이터베이스 지우기

```
DROP DATABASE my_database;
```

데이터베이스 사용하기

```
USE my_database;
```

테이블 만들기

```
CREATE TABLE users (  
  id INT PRIMARY KEY,  
  name VARCHAR(100),  
  age INT  
);
```

```
mysql> select * from users;  
+----+-----+-----+-----+  
| id | name  | age  | email |  
+----+-----+-----+-----+  
|  1 | Alice |   30 | NULL  |  
+----+-----+-----+-----+  
1 row in set (0.00 sec)
```

테이블 수정하기

```
ALTER TABLE users ADD COLUMN email VARCHAR(100);
```

테이블 지우기

```
DROP TABLE users;
```

SQL 사용법

2. 데이터 조작 언어

테이블에 데이터 첨가하기

```
INSERT INTO users (id, name, age) VALUES (1, 'Alice', 30);
```

```
INSERT INTO users (id, name, email, age) VALUES (2, 'John',  
'john@example.com', 32), (3, 'Bob', 'bob@example.com', 25),  
(4, 'Charlie', 'charlie@example.com', 35);
```

데이터 수정하기

```
UPDATE users SET age = 31 WHERE id = 1;
```

```
UPDATE users SET age = 26 WHERE name = 'Bob';
```

데이터 삭제하기

```
DELETE FROM users WHERE name = 'Charlie';
```

SQL 사용법

3. 데이터 조회 언어

데이터 조회 (SELECT)

```
SELECT name, age FROM users;
```

```
SELECT * FROM users;
```

조건 데이터 조회 (WHERE)

```
SELECT name FROM users WHERE age > 25;
```

조건 결과 정렬 (ORDER BY)

```
SELECT name, age FROM users ORDER BY age DESC;
```

```
SELECT name, age FROM users ORDER BY age DESC LIMIT 2;
```

IOT 보드 실시간 정보

"type":3,
"email":"bombzoo78@gmail.com",
"mac":"EC:64:C9:43:E8:B8",
"temp":28.4,
"humi":38,
"in":[0,0,0,0],
"out":[0,0,0,0]



"timestamp":1727170478,
"email":"bombzoo78@gmail.com",
"mac":"EC:64:C9:43:E8:B8",
"temp":28.4,
"humi":38,
"in":[0,0,0,0],
"out":[0,0,0,0]

```
mysql> select * from status;
```

timestamp	email	mac	temp	humi	in00	in01	in02	in03	out00	out01	out02	out03
1727174982	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.2	40	0	0	0	0	0	0	0	0
1727174984	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.2	39	0	0	0	0	0	0	0	0
1727174990	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.1	39	0	0	0	0	0	0	0	0
1727175003	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.2	39	0	0	0	0	0	0	0	0
1727175006	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.2	40	0	0	0	0	0	0	0	0
1727175015	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.3	40	0	0	0	0	0	0	0	0
1727175020	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.8	99	0	0	0	0	0	0	0	0
1727175025	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.4	51	0	0	0	0	0	0	1	0
1727175030	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.4	47	0	0	0	0	0	0	1	0
1727175035	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.4	45	0	0	0	0	0	0	0	0
1727175040	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.4	44	0	0	0	0	0	0	0	0
1727175047	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.4	43	0	0	0	0	0	0	0	0
1727175050	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.4	42	0	0	0	0	0	0	0	0

IOT DB 만들기

데이터베이스 만들기

```
CREATE DATABASE iot;
```

데이터베이스 사용하기

```
USE iot;
```

테이블 보기

```
SHOW TABLES;
```

테이블 만들기

```
CREATE TABLE status (  
    timestamp VARCHAR(45) PRIMARY KEY,  
    email VARCHAR(45) NOT NULL,  
    mac VARCHAR(45) NOT NULL,  
    temp FLOAT NOT NULL,  
    humi FLOAT NOT NULL,  
    in00 INT NOT NULL,  
    in01 INT NOT NULL,  
    in02 INT NOT NULL,  
    in03 INT NOT NULL,  
    out00 INT NOT NULL,  
    out01 INT NOT NULL,  
    out02 INT NOT NULL,  
    out03 INT NOT NULL  
);
```

IOT DB 만들기

데이터 보기

```
SELECT * FROM status;
```

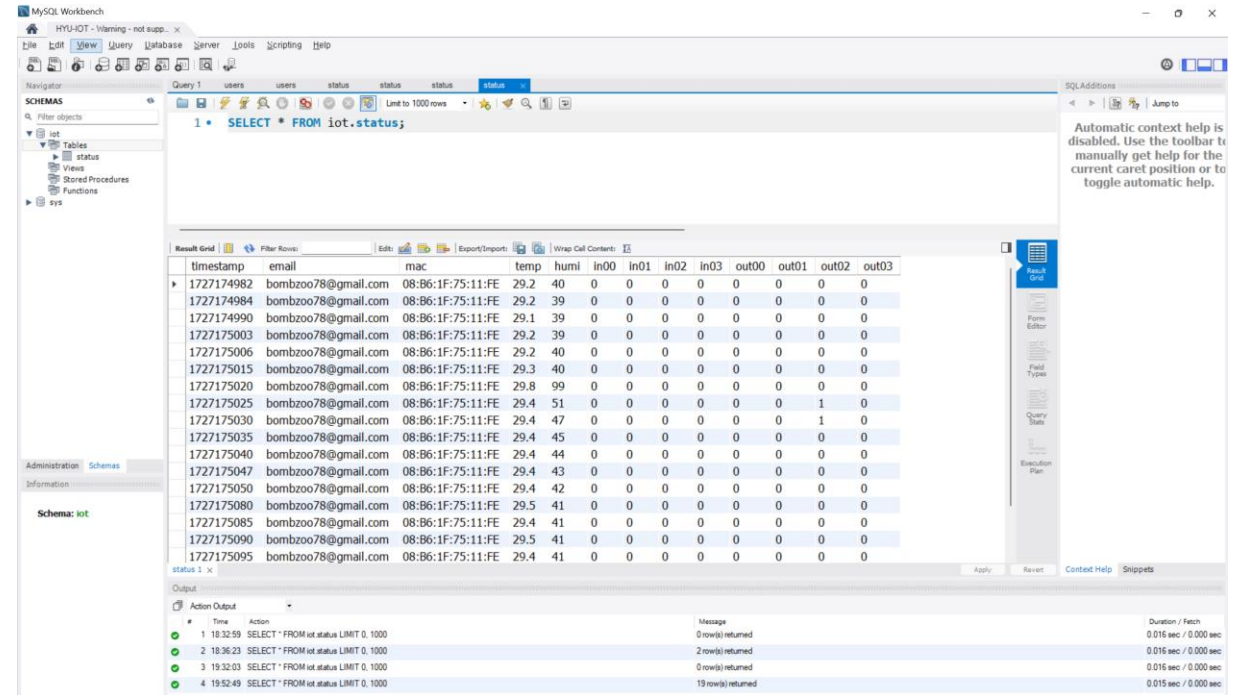
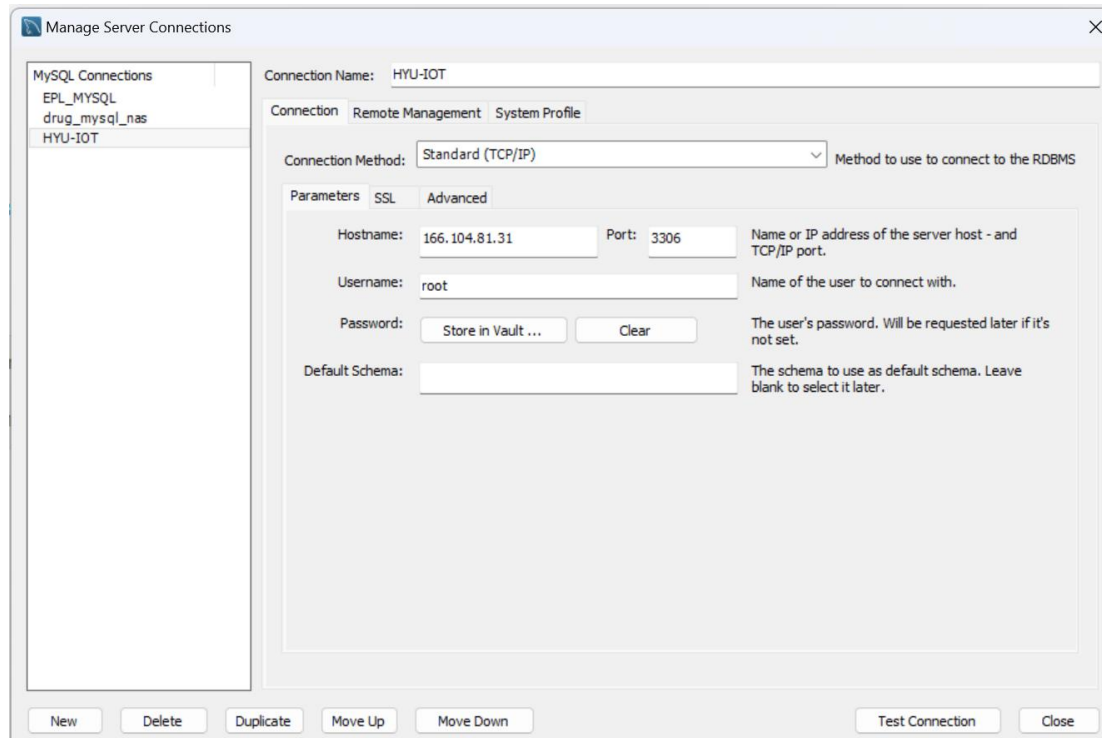
```
mysql> select * from status;
```

timestamp	email	mac	temp	humi	in00	in01	in02	in03	out00	out01	out02	out03
1727174982	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.2	40	0	0	0	0	0	0	0	0
1727174984	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.2	39	0	0	0	0	0	0	0	0
1727174990	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.1	39	0	0	0	0	0	0	0	0
1727175003	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.2	39	0	0	0	0	0	0	0	0
1727175006	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.2	40	0	0	0	0	0	0	0	0
1727175015	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.3	40	0	0	0	0	0	0	0	0
1727175020	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.8	99	0	0	0	0	0	0	0	0
1727175025	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.4	51	0	0	0	0	0	0	1	0
1727175030	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.4	47	0	0	0	0	0	0	1	0
1727175035	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.4	45	0	0	0	0	0	0	0	0
1727175040	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.4	44	0	0	0	0	0	0	0	0
1727175047	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.4	43	0	0	0	0	0	0	0	0
1727175050	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.4	42	0	0	0	0	0	0	0	0
1727175080	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.5	41	0	0	0	0	0	0	0	0
1727175085	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.4	41	0	0	0	0	0	0	0	0
1727175090	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.5	41	0	0	0	0	0	0	0	0
1727175095	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.4	41	0	0	0	0	0	0	0	0
1727175140	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.5	41	0	0	0	0	0	0	0	0
1727175160	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.5	40	0	0	0	0	0	0	0	0
1727175180	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.5	41	0	0	0	0	0	0	0	0
1727175185	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.5	40	0	0	0	0	0	0	0	0
1727175195	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.6	40	0	0	0	0	0	0	0	0
1727175215	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.5	40	0	0	0	0	0	0	0	0
1727175220	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.6	40	0	0	0	0	0	0	0	0

MySQL Workbench

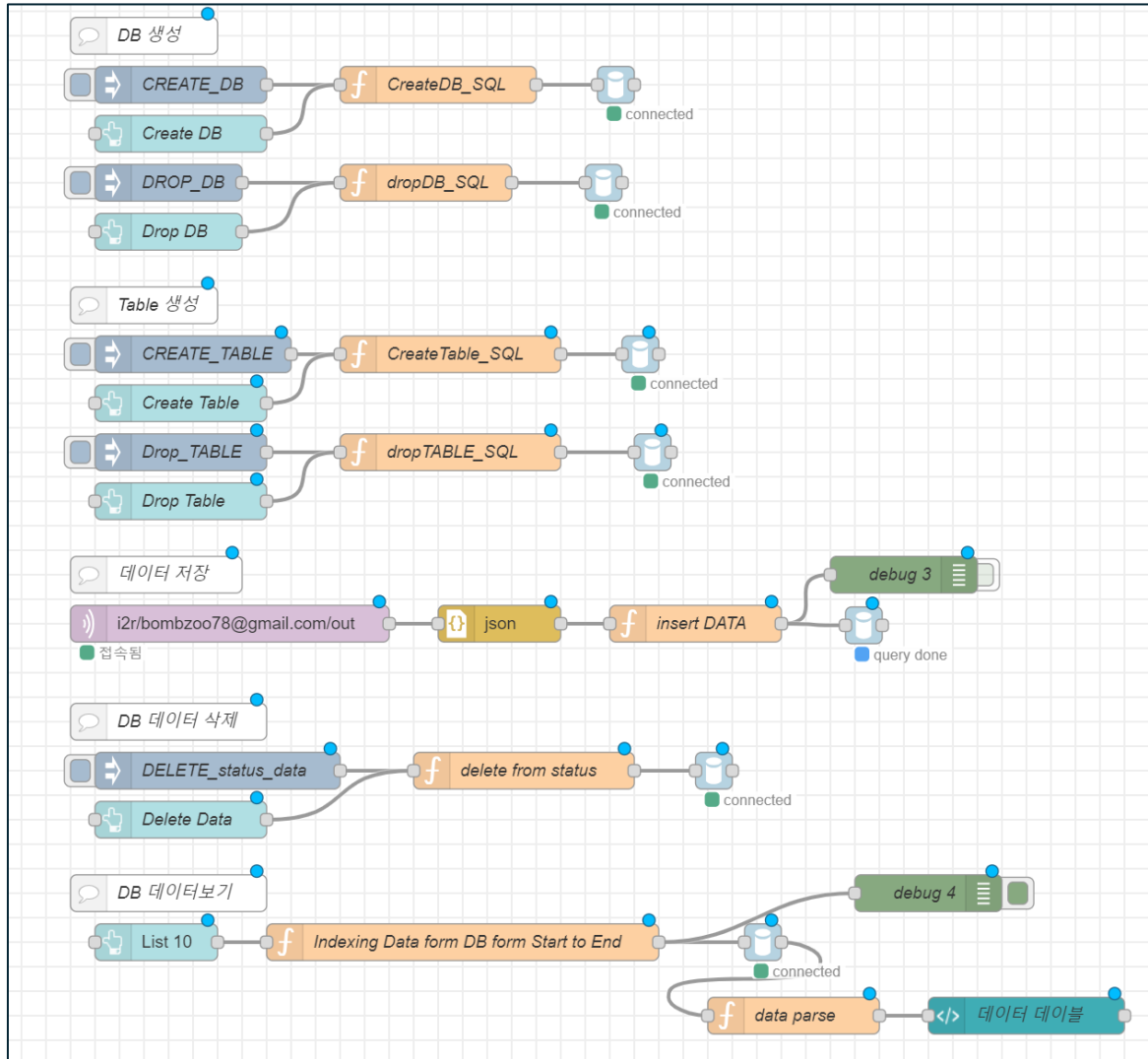
홈페이지

<https://dev.mysql.com/downloads/workbench/>



Node-Red 사용법

DB 관리



Dashboard

DB 관리

DB 관리		데이터 표시	
CREATE DB	DROP DB	LIST 10	
CREATE TABLE	DROP TABLE	Timestamp	email
DELETE DATA		9/26/2024, 10:11:36 AM	bombzoo78@gmail.com 08:
		9/26/2024, 10:11:22 AM	bombzoo78@gmail.com 08:
		9/26/2024, 10:11:11 AM	bombzoo78@gmail.com 08:
		9/26/2024, 10:10:41 AM	bombzoo78@gmail.com 08:
		9/26/2024, 10:10:36 AM	bombzoo78@gmail.com 08:
		9/26/2024, 10:10:31 AM	bombzoo78@gmail.com 08:
		9/26/2024, 10:10:26 AM	bombzoo78@gmail.com 08:
		9/26/2024, 10:10:11 AM	bombzoo78@gmail.com 08:
		9/26/2024, 10:09:41 AM	bombzoo78@gmail.com 08:
		9/26/2024, 10:09:06 AM	bombzoo78@gmail.com 08:

