

[2018 종합설계1]

<졸업작품 최종보고서>

## 미세포(미세먼지 INFORMATION)

지도교수

한선영 교수님

Project Team

201011348 윤장웅

201311269 김제헌

201311274 박기원

## 내용

1. 개발 동기 .....	3
2. 프로젝트 목표 .....	4
3. 전체 흐름도 .....	5
4. 서버 아키텍처 .....	7
5. 모바일 아키텍처 .....	8
6. 미세먼지 기준 및 단계별 대응요령 .....	9
7. 프로젝트 실행 화면 .....	11
8. 추후 개선 사항 .....	17

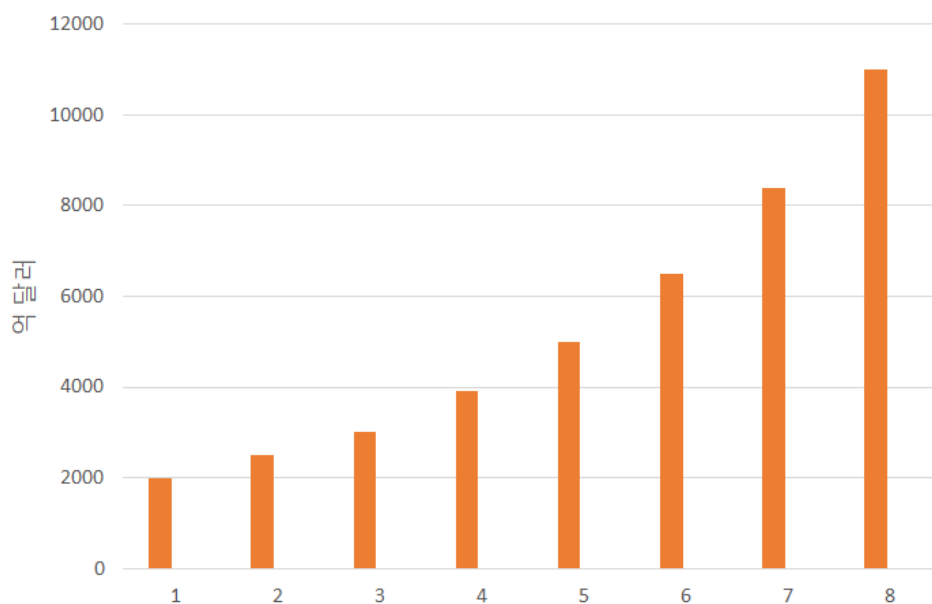
# 1. 개발 동기

## 1.1 올해 대표 이슈는 '미세먼지'

사회적 관심도를 알 수 있는 구글 트렌드에 따르면 3월 중순 이후 미세먼지에 대한 관심도가 크게 증가했다고 한다. 3월초 구글 내 미세먼지 검색은 10포인트 수준이었지만 27일 현재 83포인트로 8배 이상 관심도가 높아졌다. 게다가 미세먼지 주의보가 상시화 되면서 최근 미세먼지 관련 제품들이 불티나게 팔리고 있다.

## 1.2 점점 커지고 있는 세계 IoT 시장 전망

다음 차트는 세계 IoT의 시장 전망이 점점 더 커지고 있다는 그래프이다. 세계적으로 IoT 시장이 크게 성장하고 있는 상황에서 어떤 식으로 IoT가 이루어지고 개발되는지 한번 경험해보고자 해서 주제를 IoT로 정해보았다. IoT 중에서도 최근 폭발적으로 각광받고 있는 Home IoT에 주목했다.



<세계 IoT의 시장 전망>

### 1.3 시중에 나와있는 App으로는 알 수 없는 우리 집의 미세먼지 농도

현재 수많은 미세먼지 알람 어플리케이션이 존재하지만, 대체적으로 반경 수백 미터 당 하나씩 대기오염 측정기가 설치된 정보만 받을 수 있다. 심지어 지방의 경우 대기오염 측정기는 지역 당 몇 개 되지 않는다. 즉, 시중에 제공되고 있는 어플리케이션으로는 현재 자신의 집의 미세먼지 농도는 정확하게 알 수 없다.

## **2. 프로젝트 목표**

### **Hot Issue Keyword의 결합. '미세먼지' + 'Home IoT' + OpenStack**

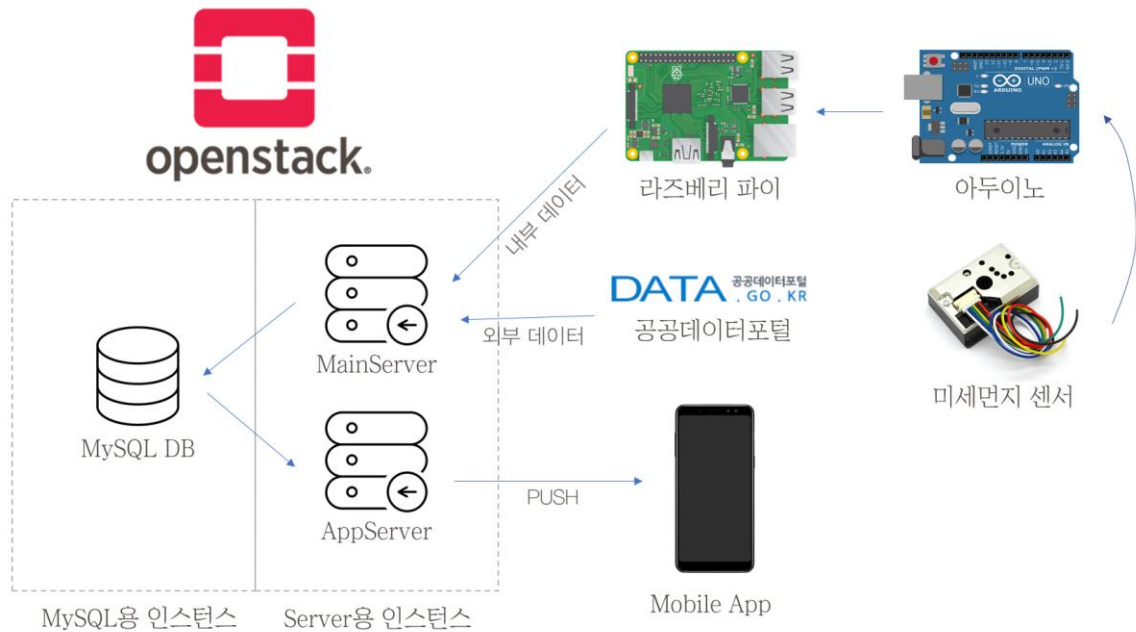
#### 2.1 '미세포'의 의의

현재 자신의 집의 미세먼지 농도는 정확하게 알 수 없는 시중에 제공되고 있는 어플리케이션과는 달리, 미세먼지를 측정해주는 Home IoT '미세포'를 통해 어플리케이션으로 현재 우리 집의 미세먼지를 확인할 수 있다. 서비스적인 측면 외에도, 요즘 대규모 데이터를 처리하는 대부분의 IoT 서비스들이 클라우드 환경에서 돌아간다는 점에 감안해서 기술적으로 OpenStack을 사용해 보기로 하였다.

#### 2.2 '미세포'의 기능적인 목표

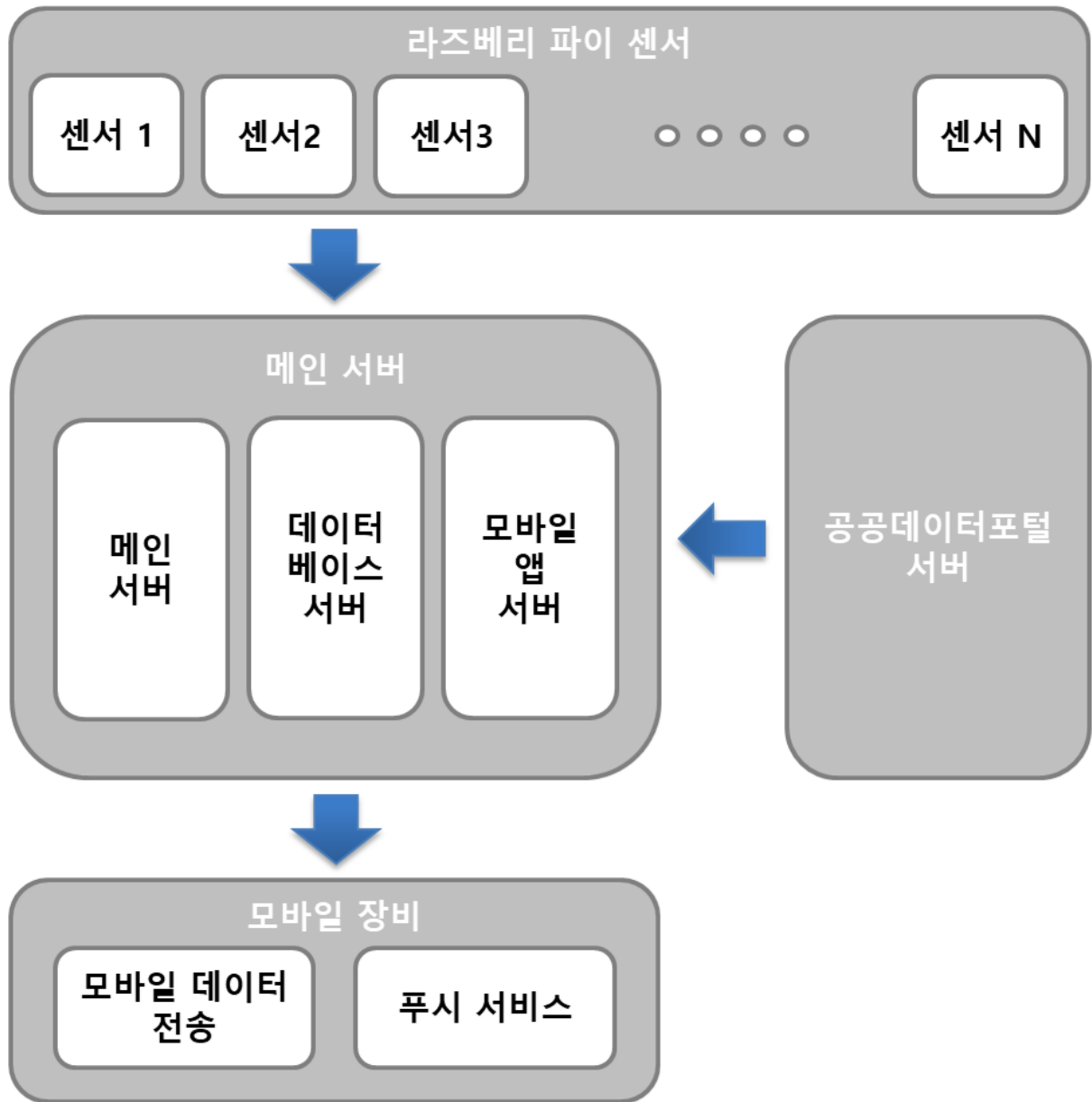
미세먼지 센서를 통해 미세먼지 농도를 측정하고 데이터를 수집한다. 그와는 별개로 공공데이터포털에서 현재 미세먼지 농도도 받아온다. 이 둘의 값을 어플리케이션을 통해 사용자에게 제공한다. 이 때 수집된 데이터는 OpenStack의 내부 DB에 저장된다.

### 3. 전체 흐름도



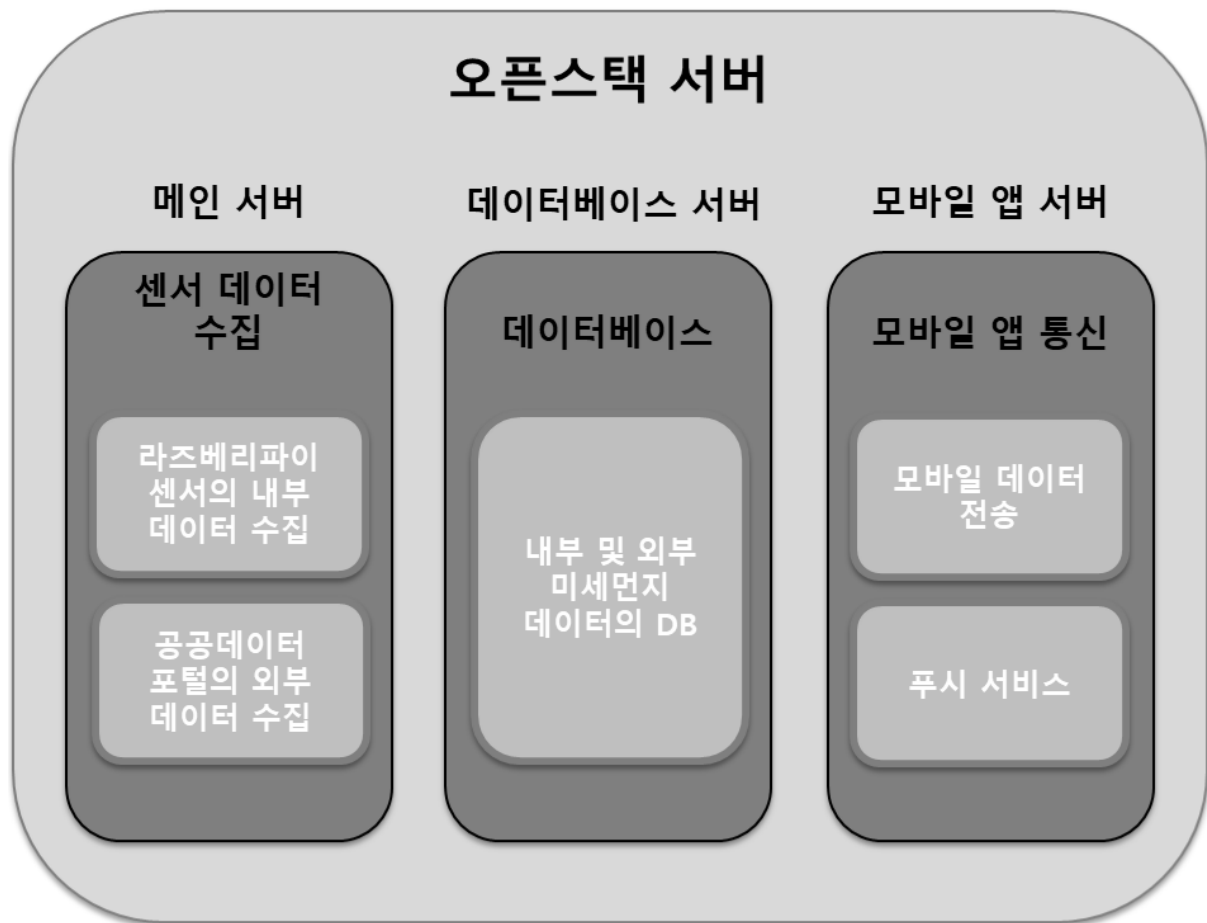
- 오픈 스택의 경우, Server 용 인스턴스와 MySQL 용 인스턴스 2 개를 생성하였다.
- 기본적으로 OpenStack 에서 Server 용 인스턴스의 MainServer 및 AppServer, MySQL 용 인스턴스의 MySQL DB 가 돌아가고 있다.
- 미세먼지 센서의 경우 라즈베리 파이용 센서가 존재하지 않아서 아두이노용 센서를 구매하였고, 따라서 아두이노도 이용할 수 밖에 없었다.
- 라즈베리파이에서 'Arduino IDE'를 통해서 아두이노를 제어하여, 미세먼지 센서로부터 집안 내부의 미세먼지 측정 값을 읽어오면, RaspberryPi 가 이를 OpenStack 에서 Server 용 인스턴스의 MainServer 로 보내게 된다.
- 이와는 별개로 MainServer 에서 공공데이터포털로부터 외부 미세먼지 측정 값을 읽어온다.
- MainServer 는 1 분마다 이 둘의 값을 읽어오게 되며, 이 값들을 MySQL 용 인스턴스의 MySQL DB 에 저장하게 된다.
- Mobile App 을 실행 시, Server 용 인스턴스의 AppServer 에 접속해서 데이터 요청을 받고, MySQL 용 인스턴스의 MySQL DB 에 저장된 가장 최근 측정값을 받아서 반환하게 된다.

좀 더 큰 관점에서 보다 간략하게 서술해본 흐름도이다.



1. N 개수의 라즈베리 파이 장비로부터 미세먼지 데이터를 실시간으로 메인 서버로 전송한다. 동시에 공공데이터포털 서버로부터 외부 미세먼지 데이터도 수집한다.
2. 메인서버에서 수집된 데이터는 데이터베이스 서버로 재전송하여 저장한다.
3. 모바일 장비에서 데이터 요청을 받으면 모바일 앱 서버에서 데이터베이스의 데이터를 앱으로 전송한다.
4. 제안된 조건이 성립이 되면 모바일 앱 서버의 푸시 서비스를 요청하여 모바일 장비로 푸시 알람 메시지가 전송된다.

## 4. 서버 아키텍처



오픈스택을 활용하여 총 3 개의 서버를 구현한다.

### 1. 메인서버

- 운영체제 : 우분투
- 프로그래밍 언어 : 파이썬
- 역할 :
  1. 라즈베리 파이 센서의 데이터를 실시간[1 분 간격]으로 수집하여 메인서버에 전송한다.
  2. 기상청 데이터를 실시간[1 분 간격]으로 수집한다.
  3. 메인서버에서 수집된 데이터는 데이터베이스 서버로 재전송하여 저장한다.
  4. 조건 성립시 모바일 푸시 서버로 푸시 요청을 보낸다.

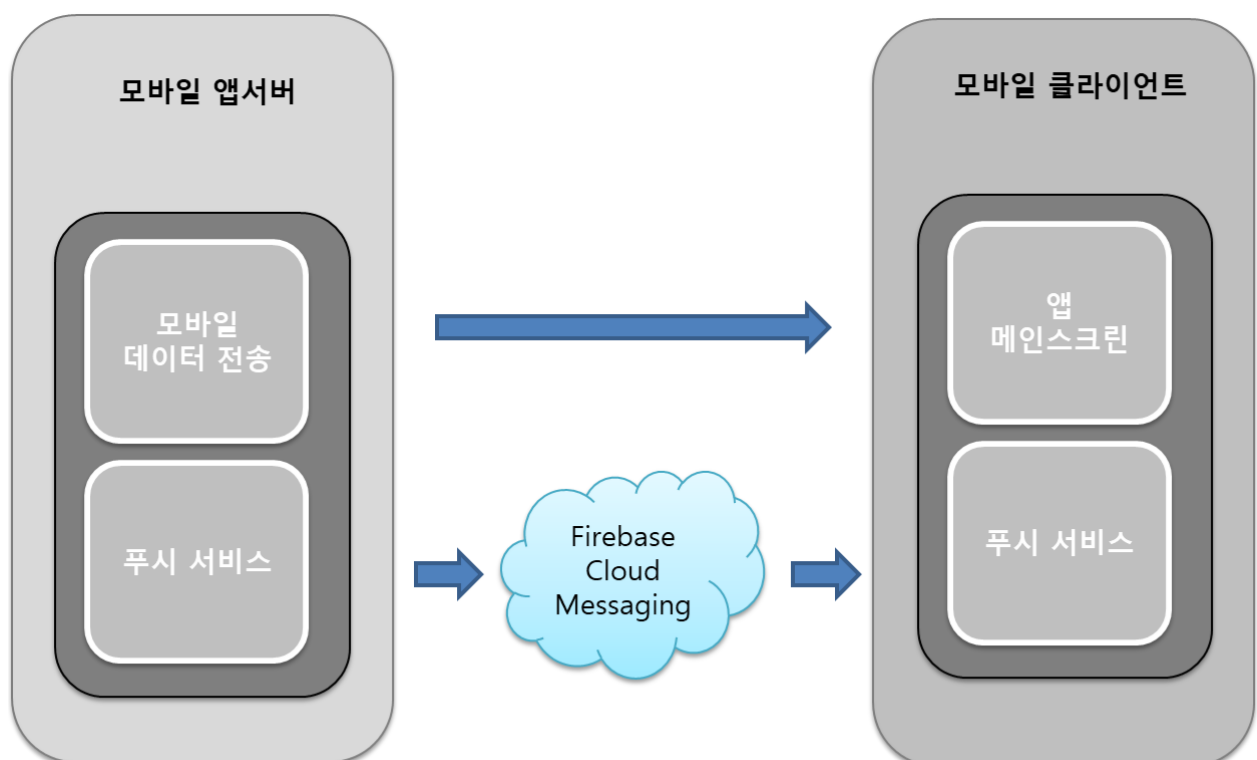
## 2. 데이터베이스 서버

- 운영체제 : 우분투
- 데이터베이스 : MySQL
- 역할 :
  1. 메인서버에서 전송된 데이터를 데이터베이스 형식으로 저장한다.
  2. 라즈베리 센서 데이터와 공공데이터 포털의 데이터를 동일한 데이터베이스에 저장한다.
  3. 모바일 장비에서 데이터 요청을 받으면 모바일 앱 서버에서 데이터베이스의 데이터를 앱으로 전송한다.

## 3. 모바일 앱 서버

- 운영체제 : 우분투
- 프로그래밍언어 : 파이썬
- 역할 :
  1. 데이터베이스에서 최신 측정 값을 모바일 앱으로 전송한다.
  2. 메인 서버에서 푸시 요청시 모바일 장비로 푸시 메시지를 전송한다.

# 5. 모바일 아키텍처





파이썬으로 앱 서버를 구축하고 안드로이드 모바일 애플리케이션과 통신한다.

1. 안드로이드 애플리케이션 작동시 모바일 앱 서버로 정보를 요청한다.
2. 모바일 앱서버에서는 데이터베이스에서 데이터를 가져와 모바일 클라이언트로 전송해준다.
3. 조건 성립시 메인서버에서 모바일 앱서버로 푸시 서비스 요청을 하면, Google Firebase Cloud Messaging(FCM) 을 통해 모바일 앱으로 푸시 알람 메시지를 보낸다.

## 6. 미세먼지 기준 및 단계별 대응요령

### 초미세먼지(PM2.5)란 (환경부 지침)?

지름 2.5 $\mu$ m 이하의 우리 눈에 보이지 않을 정도로 가늘고 작은 입자이다.

지수단계	좋음	보통	나쁨	매우 나쁨
기준값( $\mu$ g/ $m^3$ )	0 ~ 15	16 ~ 35	36 ~ 75	76 ~

### 미세먼지 단계별 대응요령 (환경부 지침) :

#### 1 단계 : 고농도 발생

- 가급적 외출 자제하기
- 외출시 보건용 마스크 착용하기
- 외출시 대기오염이 심한 도로변, 공사장은 피하고 활동량 줄이기
- 폐기물 태우는 행위 등 대기오염 유발 자제하기

#### 2 단계 : 비상저감조치 발령

- 에어코리아, 우리동네대기질 모바일 앱 활용 미세먼지 농도 수시 확인
- TV 방송(기상예보) 미세먼지 확인
- 차량 2부제 대비 교통수단 점검하기
- 보건용 마스크(KF80, KF94, KF99) 준비하기

### **3 단계 : 비상저감조치 시행**

- 홀수날에는 홀수 차량이, 짝수날에는 짝수 차량이 운행
- 서울시 공공기관 주차장 폐쇄, 체육 문화 의료시설 주차장은 차량 2부제(인천, 경기 자율참여)
- 서울시 출퇴근 서울 대중교통 무료

### **4 단계 : 주의보 발령**

- 영유아, 학생, 어르신: 실외수업(활동) 시간 단축 또는 금지 / 이용시설 내 기계, 기구류 세척 등 식당 위생관리 강화
- 일반국민 : 가급적 외출 자제하기 / 외출시 보건용 마스크 착용하기 / 외출시 대기오염이 심한 도로변, 공사장은 피하고 활동량 줄이기 / 폐기물 태우는 행위 등 대기오염 유발 자제하기

### **5 단계 : 경보 발령**

- 영유아, 학생, 어르신: 등,하교(원) 시간 조정, 휴업 권고 / 질환자 파악 및 특별 관리 (진료,조기귀가 등)
- 일반국민 : 가급적 외출 자제하기 / 외출시 보건용 마스크 착용하기 / 외출시 대기오염이 심한 도로변, 공사장은 피하고 활동량 줄이기 / 폐기물 태우는 행위 등 대기오염 유발 자제하기

### **6 단계 : 주의보 경보 해제**

- 외출 후 깨끗이 씻기
- 물관 비타민 C가 풍부한 과일, 야채 섭취하기
- 실내 공기질 관리하기 : 실내,외 공기 오염도를 고려하여 적절한 환기 실시하기 / 실내 물걸레질 등 물청소하기

## 7. 프로젝트 실행 화면

- 메인 서버의 실행 화면 : OpenStack에서 Server용 인스턴스에 ssh 접속한 후, MainServer.py를 실행시킨 화면이다. MainServer가 시작되면, MySQL용 인스턴스의 DB에 접속해서, DB가 있는지 확인한다. 만약 DB가 없으면 DB 및 테이블을 새로 만든다.

```
kimjh@ubuntu:~/졸업작품/Ver0.5$ ssh ubuntu@117.16.137.226
ubuntu@117.16.137.226's password:
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-116-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

16 packages can be updated.
13 updates are security updates.

Last login: Wed Jun  6 16:05:23 2018 from 175.125.195.69
ubuntu@server:~$ python3 MainServer.py
PI Bind
이미 testDB가 만들어져 있음
```

- 라즈베리파이의 메인 서버 접속 : 라즈베리파이에서 RaspberryPi.py 를 실행시키면, OpenStack Server 용 인스턴스의 MainServer 에 접속하게 된다. 다음은 이때의 RaspberryPi 및 MainServer 의 화면이다.

```
kimjh@ubuntu:~/졸업작품/Ver0.5$ ssh pi@10.10.10.14
pi@10.10.10.14's password:
Linux raspberrypi 4.14.30-v7+ #1102 SMP Mon Mar 26 16:45:49 BST 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jun  6 09:23:01 2018 from 10.10.10.12
pi@raspberrypi:~$ python3 RaspberryPi.py
Arduino Serial is open : True
hello piTest
Sensing [60 sec]
PI Bind
이미 testDB가 만들어져 있음
1번째 New Connected RaspberryPi : ('175.125.195.69', 58130)
hello mainServer
pi로부터 메시지를 받기위해 대기중
```

- 파이어에서의 내부 미세먼지 데이터를 메인 서버로 전송 : RaspberryPi 에서 시리얼 통신을 통해서 Arduino 와 연결된 미세먼지 센서로부터 집안 내부의 미세먼지 측정 값을 읽어오면, 이를 OpenStack 에서 Server 용 인스턴스의 MainServer 로 보내게 된다. 다음은 이때의 RaspberryPi 및 MainServer 의 화면이다.

```

Sensing [60 sec]
11.5714
12
[0] fine dust data : 12 Sending
Sensing [60 sec]
7.1429
7
[1] fine dust data : 7 Sending
Sensing [60 sec]
9.4286
9
[2] fine dust data : 9 Sending
Sensing [60 sec]
11.0714
11
[3] fine dust data : 11 Sending

```

[측정값은 반올림해서 전송된다]

```

pi로부터 메시지를 받기위해 대기중
pi로부터 받은 메시지 : data/12
pi로부터 메시지를 받기위해 대기중
pi로부터 받은 메시지 : data/7
pi로부터 메시지를 받기위해 대기중
pi로부터 받은 메시지 : data/9
pi로부터 메시지를 받기위해 대기중
pi로부터 받은 메시지 : data/11

```

- 공공데이터포털 : 이와는 별개로 MainServer 에서 공공데이터포털로부터 외부 미세먼지 측정 값을 읽어온다.

```

pi로부터 메시지를 받기위해 대기중
pi로부터 받은 메시지 : data/12

외부에서 읽어들인 값
['2018-06-06 18:00', '광진구', '0.005', '0.8', '0.047', '0.020', '50', '27']
이중, 필요로 하고 있는 데이터는 미세먼지[PM10] 데이터 : 50 µg/m³ 와, 초미세먼지[PM2.5] 데이터 : 27 µg/m³ 입니다.

```

- DB 저장 : MainServer 는 1 분마다 이 둘의 값을 읽어오게 되며, 이 값들을 MySQL 용 인스턴스의 MySQL DB 에 저장하게 된다.

```

가장 최근에 삽입된 데이터. in_value=12, out_value=27

```

- DB 목록 : MySQL 용 인스턴스에 ssh 로 접속 후, MySQL DB 에 접속해서 현재 테이블 목록을 확인해본 화면이다. 1 분 간격으로 내부 및 외부 데이터가 쌓이고 있음을 확인 할 수 있다.

48	8	27	2018-06-06 18:31:23
49	9	27	2018-06-06 18:32:23
50	11	27	2018-06-06 18:33:23
51	14	27	2018-06-06 18:34:23
52	12	27	2018-06-06 18:35:53
53	7	27	2018-06-06 18:36:53
54	9	27	2018-06-06 18:37:53
55	11	27	2018-06-06 18:38:53
56	12	27	2018-06-06 18:39:53
57	7	27	2018-06-06 18:40:53
58	11	27	2018-06-06 18:41:53
59	10	27	2018-06-06 18:42:53
60	11	27	2018-06-06 18:43:53
61	9	27	2018-06-06 18:44:53
62	11	27	2018-06-06 18:45:54
63	9	27	2018-06-06 18:46:54

- Mobile App 을 실행 시, Server 용 인스턴스의 AppServer 에 접속해서 데이터 요청을 받고, MySQL 용 인스턴스의 MySQL DB 에 저장된 가장 최근 측정값을 받아서 반환하게 된다.
- 다음은 app\_server.py 의 수행화면이다.

```
ubuntu@server:~/misefor/venv$ source bin/activate
(venv) ubuntu@server:~/misefor/venv$ cd ../../
(venv) ubuntu@server:~$ cd misefor/venv
(venv) ubuntu@server:~/misefor/venv$ python app_server.py
* Serving Flask app "app_server" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
{'u'out_value': 12, u'id': 88, u'in_value': 1, u'register_time': datetime.datetime(2018, 6, 14, 17, 24, 48)}
fine_dust table [in_value : 1, out_value : 12]
221.146.87.39 - - [20/Jun/2018 15:41:45] "POST /api/post_some_data HTTP/1.1" 200
-
{'u'out_value': 12, u'id': 88, u'in_value': 1, u'register_time': datetime.datetime(2018, 6, 14, 17, 24, 48)}
fine_dust table [in_value : 1, out_value : 12]
221.146.87.39 - - [20/Jun/2018 15:41:45] "POST /api/post_some_data HTTP/1.1" 200
-
{'u'out_value': 12, u'id': 88, u'in_value': 1, u'register_time': datetime.datetime(2018, 6, 14, 17, 24, 48)}
fine_dust table [in_value : 1, out_value : 12]
221.146.87.39 - - [20/Jun/2018 15:41:53] "POST /api/post_some_data HTTP/1.1" 200
-
{'u'out_value': 12, u'id': 88, u'in_value': 1, u'register_time': datetime.datetime(2018, 6, 14, 17, 24, 48)}
fine_dust table [in_value : 1, out_value : 12]
221.146.87.39 - - [20/Jun/2018 15:41:54] "POST /api/post_some_data HTTP/1.1" 200
-
```

휴대폰 앱[misefor\_v2.0.apk]의 실행화면은 다음과 같으며, 이를 설치 후 실행시켰을 때의 화면은 다음과 같다.



앞서, 6. 미세먼지 기준 및 단계별 대응요령에서 언급한 내용으로, 초미세먼지(PM2.5)의 환경부 지침에 따라 다음과 같은 이미지를 표시하였다. 앱에서 값을 요청하고, 받아온 값이, 해당 기준에 따라서 그에 맞는 이미지가 나오게 된다.

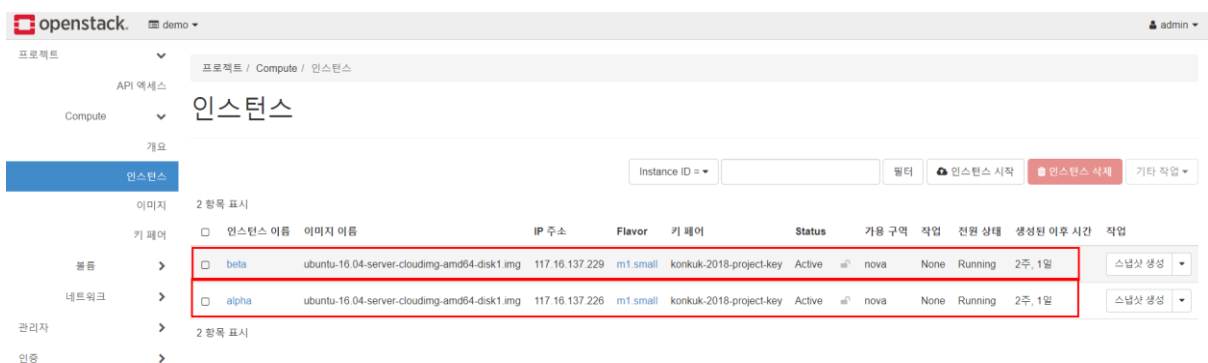
지수단계	좋음	보통	나쁨	매우 나쁨
기준값( $\mu\text{g}/\text{m}^3$ )	0 ~ 15	16 ~ 35	36 ~ 75	76 ~
이미지				

오픈스택은 한선영 교수님의 연구실(cclab)에 설치되어 있는 오픈스택을 이용하였다. 다음은 <http://117.16.137.186/> 을 입력했을 때, 오픈스택의 데시보드의 로그인 화면이다.



The image shows the OpenStack login interface. At the top is the OpenStack logo, which consists of a red square with a white 'O' inside, followed by the word 'openstack' in a dark grey sans-serif font. Below the logo is the Korean word '로그인' (Login). Underneath, there are two input fields: '사용자 이름' (Username) with the text 'admin' entered, and '암호' (Password) with a masked password '.....'. To the right of the password field is an eye icon for toggling visibility. At the bottom right is a blue button labeled '연결' (Connect).

우분투 16.04 의 운영체제로, 서버용 인스턴스인 alpha[117.16.137.226]와  
유용 인스턴스인 beta[117.16.137.229]를 생성하였다.



The image is a screenshot of the OpenStack dashboard. The top navigation bar shows 'openstack.' and 'demo'. The left sidebar has a menu with '인스턴스' (Instances) highlighted. The main content area is titled '인스턴스' and shows a table of instances. Two instances, 'beta' and 'alpha', are highlighted with a red box. The table has columns for Instance ID, Name, Image, Flavor, Key Pair, Status, and other details. The 'beta' instance has ID 'beta', name 'beta', image 'ubuntu-16.04-server-cloudimg-amd64-disk1.img', flavor 'm1.small', key pair 'konkuk-2018-project-key', and status 'Active'. The 'alpha' instance has ID 'alpha', name 'alpha', image 'ubuntu-16.04-server-cloudimg-amd64-disk1.img', flavor 'm1.small', key pair 'konkuk-2018-project-key', and status 'Active'.

Instance ID	Name	Image	Flavor	Key Pair	Status	Other
beta	beta	ubuntu-16.04-server-cloudimg-amd64-disk1.img	m1.small	konkuk-2018-project-key	Active	nova, None, Running, 2주, 1일
alpha	alpha	ubuntu-16.04-server-cloudimg-amd64-disk1.img	m1.small	konkuk-2018-project-key	Active	nova, None, Running, 2주, 1일

서버용 인스턴스인 alpha[117.16.137.226]의 사양은 다음과 같다.

The screenshot shows the OpenStack dashboard for a project named 'demo'. The breadcrumb trail is '프로젝트 / Compute / 인스턴스 / alpha'. The instance name 'alpha' is highlighted with a red box. The instance is in the 'Active' state. The '사양' (Specifications) tab is selected, showing the following details:

이름	alpha
ID	74cc61a2-ac5b-446b-8aca-981ccba3df3f
Status	Active
잠김	False
가용 구역	nova
생성됨	2018년 6월 4일 7:15 오전
생성된 이후 시간	2주, 1일
호스트	ubuntu

The '사양' tab is also highlighted with a red box. Below the specifications, the 'IP 주소' (IP Address) is shown as 'External 117.16.137.226', with the IP address highlighted by a red box.

db 용 인스턴스인 alpha[117.16.137.226]의 사양은 다음과 같다.

The screenshot shows the OpenStack dashboard for a project named 'demo'. The breadcrumb trail is '프로젝트 / Compute / 인스턴스 / beta'. The instance name 'beta' is highlighted with a red box. The instance is in the 'Active' state. The '사양' (Specifications) tab is selected, showing the following details:

이름	beta
ID	a1e7fb32-d069-4068-a413-a39343599f3b
Status	Active
잠김	False
가용 구역	nova
생성됨	2018년 6월 5일 6:04 오전
생성된 이후 시간	2주, 1일
호스트	ubuntu

The '사양' tab is also highlighted with a red box. Below the specifications, the 'IP 주소' (IP Address) is shown as 'External 117.16.137.229', with the IP address highlighted by a red box.



## 8. 추후 개선 사항

- 오토 스케일링 및 로드 밸런싱 기능 추가

이번 프로젝트에서는 서비스 자체는 단순하게 잡았고, 오픈스택을 중심으로 잡았었다. 그래서 초반에 오픈 스택의 환경 구축에 많은 시간을 소모하였고, 설치부터 오픈스택 구축까지의 과정을 익힐 수 있었다. 추후에 시간이 되서 이를 더 보완한다면, 이러한 클라우드의 장점을 활용하여 서버의 부하가 폭주할 때, 서버 인스턴스를 여러 개 늘리고 부하가 줄어들면 인스턴스를 줄이는 기능인 오토 스케일링 기능 및, 이러한 서버가 여러 대일 때 접속 부하를 적절하게 분배하는 로드 밸런싱 기능을 구현하면 좋을 것이다.