

스마트 센서와 클라우드 연동1

한백전자 기술연구소



HANBACK ELECTRONICS CO.,LTD



학습6

스마트 센서와 클라우드 연동1

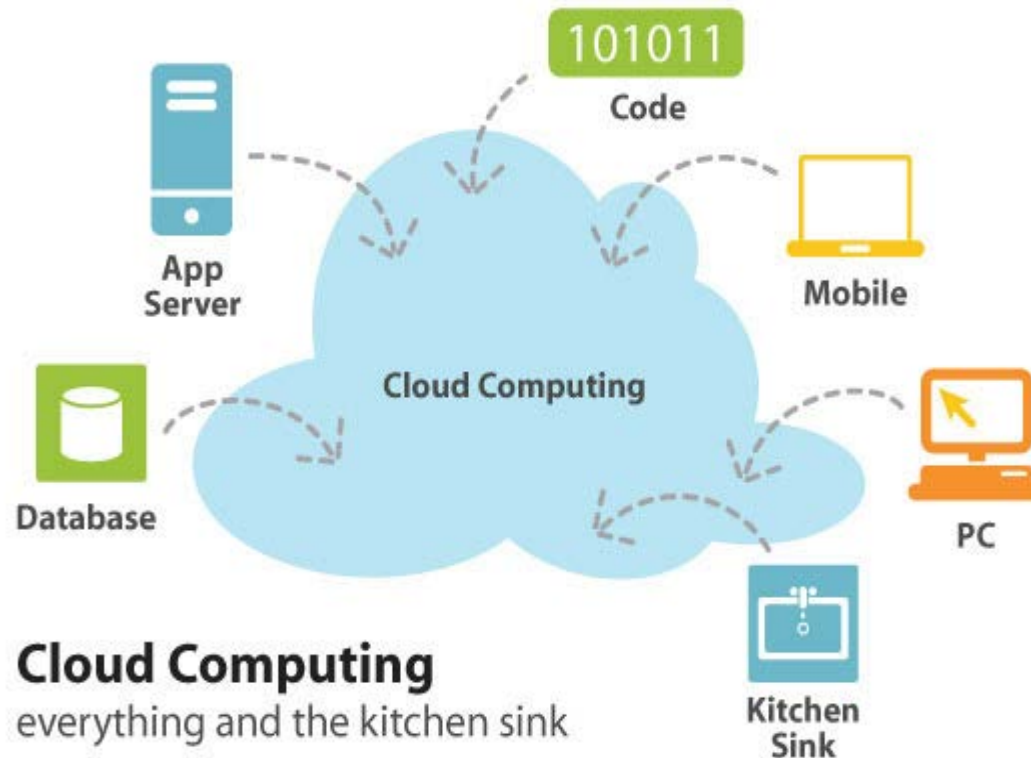


- Cloud & IoT-Smart-Pioneer 개발 환경 구성
- Minutes Cloud Platform 사용 환경 구성하기



1. 클라우드 컴퓨팅 개요

클라우드의 의미는 컴퓨터 통신망이 구름과 같은 것에 싸여 안이 보이지 않고, 일반 사용자는 이 복잡한 내부를 굳이 알 필요도 없이 어디에서나 구름 속으로 손을 집어넣어 자기가 원하는 작업을 할 수 있다는 것이다.





2. 클라우드 컴퓨팅의 3모델

- 클라우드 컴퓨팅에는 지원되는 IT자원의 성격에 따라 IaaS, PaaS, SaaS등으로 분류 된다.

1) IaaS

- IaaS 모델은 Infrastructure as a Service 의 약자로 서비스로서의 인프라는 클라우드 IT의 기본 빌딩 블록을 포함하고 일반적으로 네트워킹 기능, 컴퓨터 및 데이터 스토리지 공간을 제공합니다.

2) PaaS

- PaaS 모델은 Platform as a Service 의 약자로 서비스로서의 플랫폼을 사용하면 조직은 기본 인프라를 관리할 필요가 없어 애플리케이션 개발과 관리에 집중할 수 있습니다

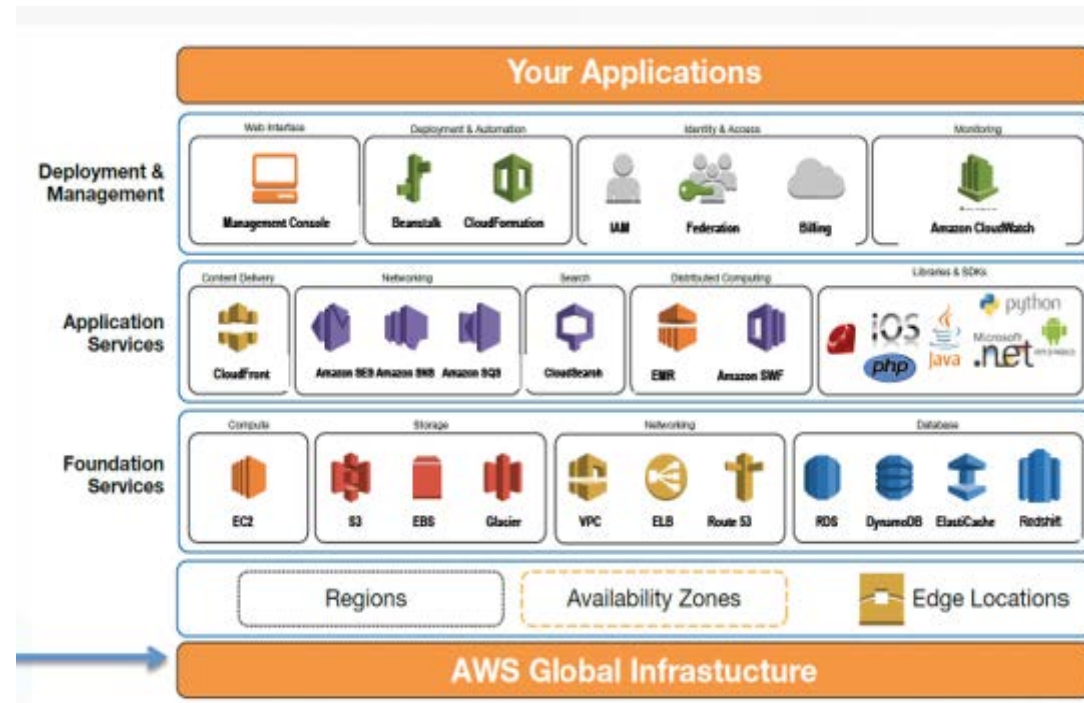
3) SaaS

- SaaS 모델은 Software as a Service 의 약자로 서비스로서의 소프트웨어를 사용하면 조직은 기본 인프라를 관리할 필요가 없어 애플리케이션 개발과 관리에 집중할 수 있습니다



3. AWS : Amazon Web Service

- 서버 호스팅 산업에 ‘자유롭게 실시간으로 서버를 임대하고 반환한다.’ 는 개념을 도입한 AWS 클라우드의 주요 이점 중 하나는 초기 기본 인프라 비용을 비즈니스에 맞춰 조정되는 저렴한 가변 비용으로 대체할 수 있는 기회가 된다는 점이다.





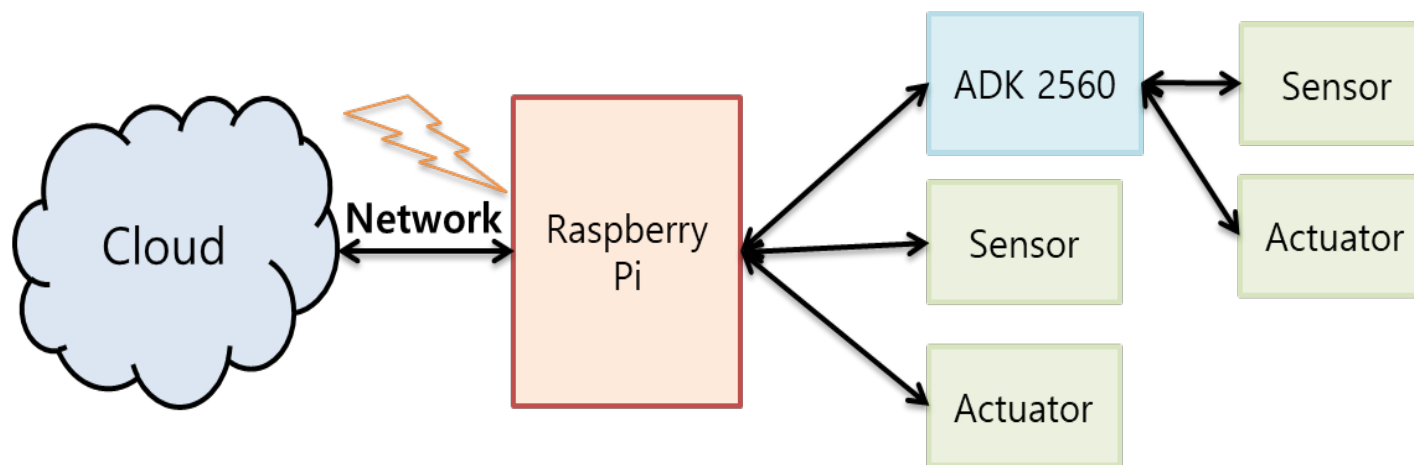
4. Minutes Cloud 플랫폼

Minutes Cloud 플랫폼은 IoT 서비스의 일종이다.

다양한 기능을 제공하는데 Dash Board, Data Buckets, End Points 크게 세가지 기능을 제공한다.

- Dash Board 는 센서 또는 액추에이터를 한 눈에 확인이 가능한 UI 기능을 제공한다.
- Data Buckets 는 등록되어 있는 특정 센서의 로그 정보를 확인해 볼 수 있는 일종의 가상 스토리지 이다.
- End Points 는 이벤트가 발생하였을 때 이메일을 보내거나, SMS 메시지를 보내거나, 다른 서비스 등을 호출하는 역할을 한다.

5. 시스템 구조





1. 패키지 설치

- 라즈베리파이에서 Minutes Cloud 에 접속하기 위해서는 소스 코드 작성과 컴파일 하는 과정이 필요하다. 컴파일 하는 과정에서 필요한 패키지를 네트워크를 통해 설치한다. 우선적으로 패키지 다운로드 받을 수 있는 경로를 확인한다.

```
pi@raspberrypi:~ $ sudo apt-get update
```

- 특별한 문제가 없다면 apt-get install 명령을 통해 cmake 패키지와 libssl-dev 패키지를 설치한다. 이 패키지들은 이후 소스코드 컴파일 하는 과정에 사용된다.

```
pi@raspberrypi:~ $ sudo apt-get install cmake libssl-dev
```



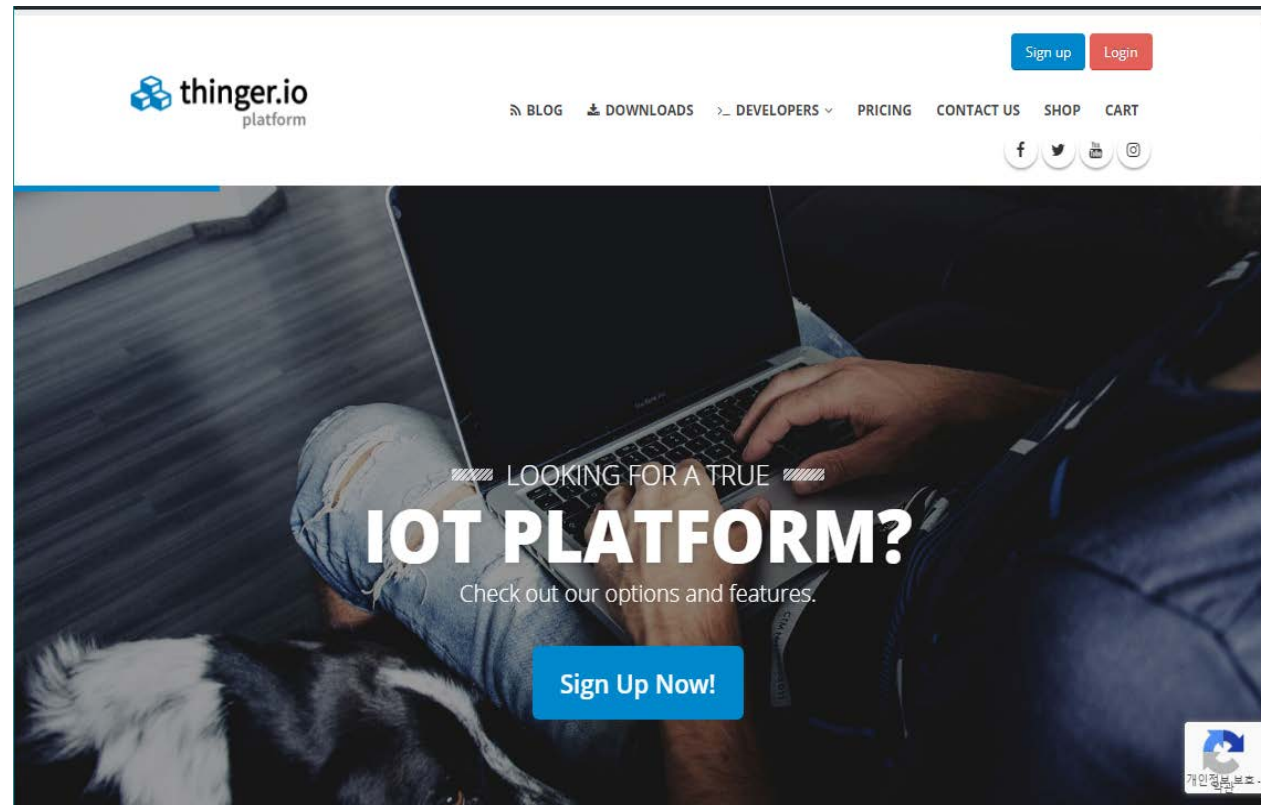

- 패키지 설치가 완료되면 git 명령을 통해 Cloud 에 접속하기 위한 소스코드 예제 파일을 다운로드 받는다. 명령 실행이 완료되면 Linux-Client 라는 폴더가 생성되며 폴더 내부에 소스코드들이 함께 다운로드 된다.

```
pi@raspberrypi:~ $ git clone https://github.com/thinger-io/Linux-Client.git
Cloning into 'Linux-Client'...
remote: Enumerating objects: 133, done.
remote: Total 133 (delta 0), reused 0 (delta 0), pack-reused 133
Receiving objects: 100% (133/133), 41.60 KiB | 0 bytes/s, done.
Resolving deltas: 100% (69/69), done.
Checking connectivity... done.
pi@raspberrypi:~ $
```



2. Cloud 회원 가입 및 설정

- Cloud 서비스에서 개인의 디바이스는 개인 계정과 연동된다.
연동 정보를 생성하여 소스코드에 적용하기 위해 우선 계정 생성이 필요하다.
www.thinger.io 로 접속 하여 회원가입 절차를 수행 한다.





- 접속 시 가운데 나타나는 Sign Up을 누르거나 우측 상단에 있는 버튼을 눌러 회원 가입 창으로 넘어간다. 회원 가입 시 필요한 정보는 아래 그림과 같다. 개인 e-mail을 이용하여 인증을 받는 형태로 되어 있다.



- 회원가입 정보를 입력하고 하단에 Sign up 버튼을 누르면 입력한 e-mail 로 인증 메일이 전달 된다. 인증 메일은 2일간 효력을 가지고 있으므로 바로 접속하여 인증 절차를 진행한다.



no-reply@thinger.io 도메인: amazonses.com

오후 1:50 (1)

나에게 ▼

文A 영어 ▼ > 한국어 ▼ [메일 번역](#)



thinger.io
platform

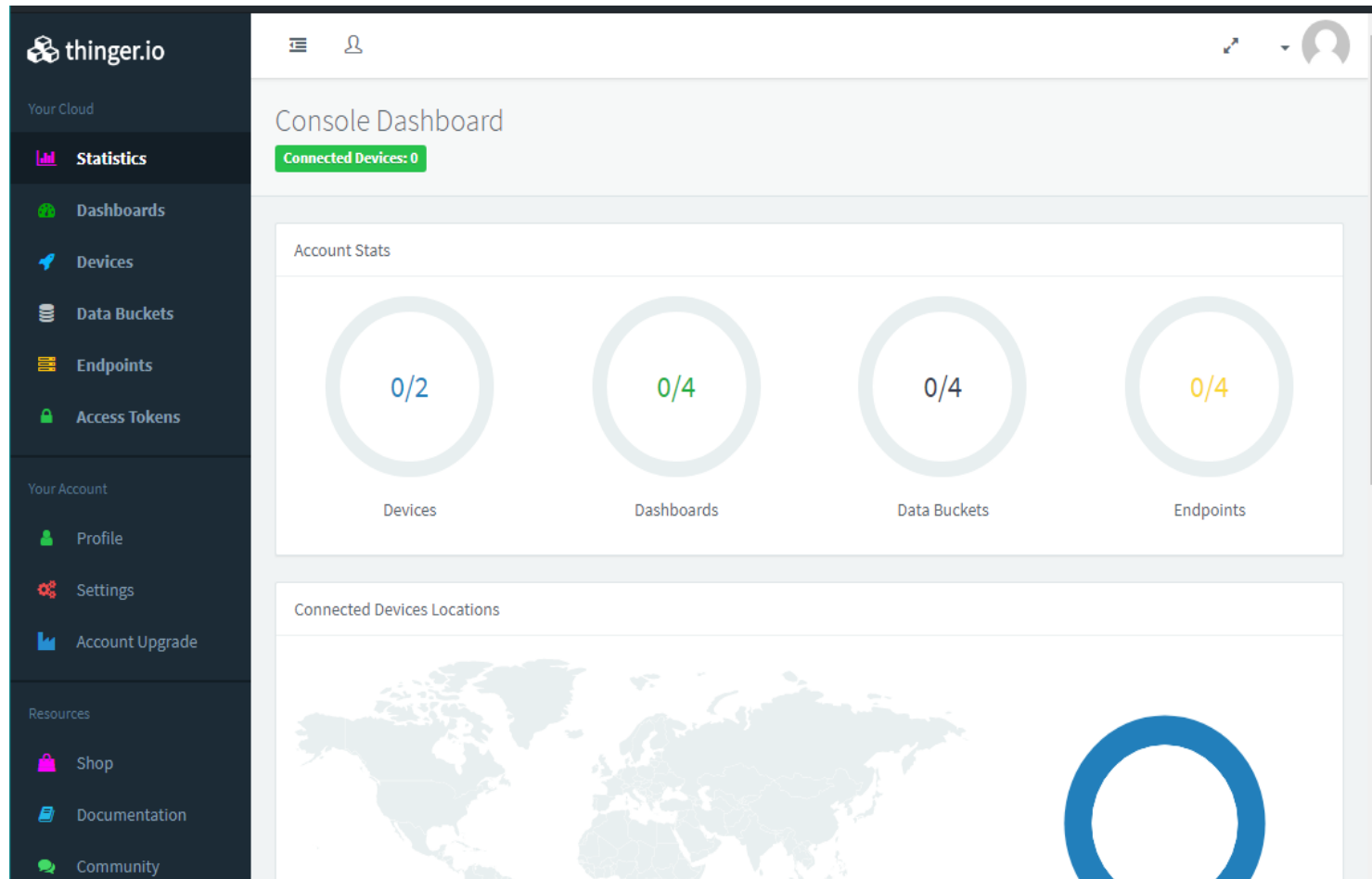
Please confirm your email address

You have created a Thinger.io Account for [shivan8709@gmail.com](#)

[Confirm Your Email](#)



- 회원 인증 절차까지 모두 마무리가 되고 나면 접속 하였을 때 첫 화면은 아래 그림과 같은 화면을 볼 수 있다.





- 디바이스 등록을 위한 정보를 생성하기 위해 좌측 메뉴 중 Devices 버튼을 눌러 화면을 전환 한다.
Add Device 버튼을 누르면 디바이스 추가 화면으로 넘어가게 된다.

The screenshot displays the 'Devices' management interface. On the left sidebar, the 'Devices' menu item is highlighted with a red box. The main content area, titled 'Devices', features a 'Device List' section. Within this section, there is a green '+ Add Device' button, which is also highlighted with a red box, and a search input field labeled 'Search'. Below the 'Add Device' button is a table with the following columns: 'Device', 'Description', 'Last Connection', and 'State'. The table currently displays the message 'No devices to show'.



- Device ID 와 마지막에 나오는 Device Credential 이다.
이 두 정보는 사용자 ID 와 함께 Cloud 접속 시 실행할 프로그램 내에 저장되어야 하는 내용이다.



- Add Device 버튼을 눌러 등록을 완료 하고 다시 좌측의 Devices 버튼을 눌러보면 새로 생성한 디바이스가 나타나는 것을 확인할 수 있다.

thinger.io

Your Cloud

- Statistics
- Dashboards
- Devices**
- Data Buckets
- Endpoints
- Access Tokens

Your Account

- Profile
- Settings
- Account Upgrade

Resources

- Shop
- Documentation
- Community

Devices

Device List ⓘ

+ Add Device

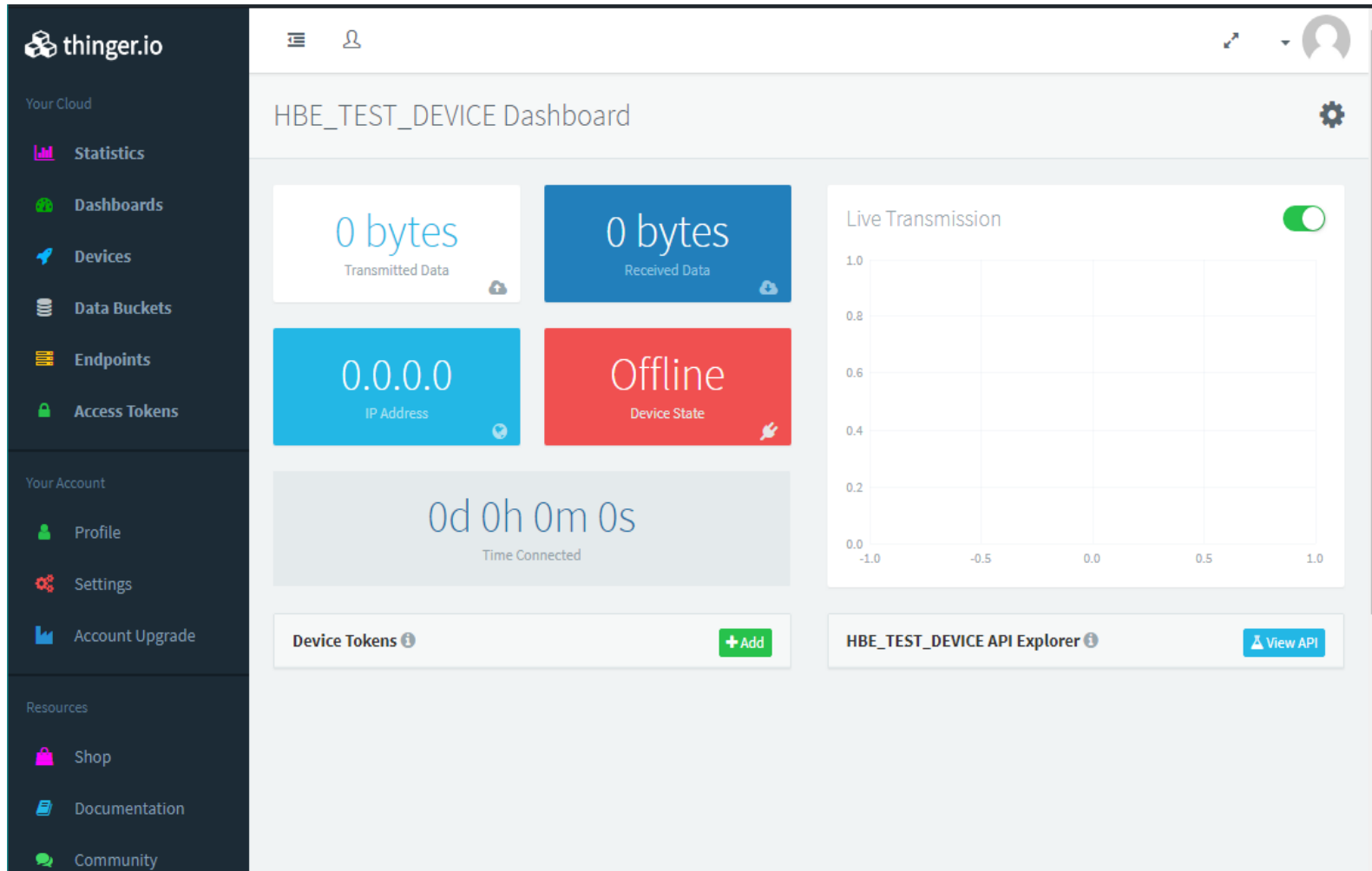
Search

Device	Description	Last Connection	State
<input type="checkbox"/> hbe_test_device	hanback iot test device	Never	Disconnected

Showing 1 device



- 생성된 디바이스의 상세정보를 확인할 때는 목록에서 디바이스 명칭을 누르면 해당 디바이스에 대한 상세 정보들이 출력되게 된다.





- 센서나 액추에이터 정보를 표기 하거나 제어할 수 있는 DashBoard를 생성한다.

The screenshot shows the 'Add Dashboard' interface on the Thinger.io platform. The left sidebar contains navigation links for 'Your Cloud' (Statistics, Dashboards, Devices, Data Buckets, Endpoints, Access Tokens), 'Your Account' (Profile, Settings, Account Upgrade), and 'Resources' (Shop, Documentation, Community). The main content area is titled 'Add Dashboard' and includes a 'Dashboard details' section with three input fields: 'Dashboard id' (placeholder: 'Enter a dashboard id'), 'Dashboard name' (placeholder: 'Enter dashboard description'), and 'Dashboard description' (placeholder: 'Enter dashboard description'). A green 'Add Dashboard' button with a checkmark icon is located at the bottom of the form.



3. 디바이스 등록

- 생성한 디바이스 정보와 회원 가입 시 입력한 ID 정보를 기반으로 디바이스 등록을 진행한다.
우선 라즈베리파이에서 Linux-Client 폴더로 이동 한다. src/main.cpp 파일을 수정한다.

```
pi@raspberrypi:~ $ cd Linux-Client/  
pi@raspberrypi:~/Linux-Client $ vi src/main.cpp
```

- USER_ID, DEVICE_ID, DEVICE_CREDENTIAL 부분의 정보를 입력한다.

```
23  
24 #include "thinger/thinger.h"  
25  
26 #define USER_ID          "YOUR_USER_ID"  
27 #define DEVICE_ID        "YOUR_DEVICE_ID"  
28 #define DEVICE_CREDENTIAL "YOUR_DEVICE_CREDENTIAL"  
29  
30 int main(int argc, char *argv[])  
31 {  
32     thinger_device thing(USER_ID, DEVICE_ID, DEVICE_CREDENTIAL);  
33  
34     // define thing resources here. i.e, this is a sum example  
35     thing["sum"] = [] (pson& in, pson& out){  
36         out["result"] = (int) in["value1"] + (int) in["value2"];  
37     };  
38  
39     thing.start();  
40     return 0;  
41 }  
1 change; before #1 3 seconds ago
```



Minutes Cloud Platform 사용 환경 구성하기



HANBACK ELECTRONICS.,LTD

- build 폴더로 이동하여 cmake 명령을 이용하여 컴파일을 위한 설정을 수행한다.
build 폴더가 없다면 mkdir 명령을 통해 폴더를 생성한다.
이 과정은 최초 1회 수행한다.

```
pi@raspberrypi:~/Linux-Client $ cd build/  
pi@raspberrypi:~/Linux-Client/build $ cmake -DCMAKE_BUILD_TYPE=Release -DDAEMON=OFF -DRASPBerry=ON .  
./
```

- 컴파일과 실행을 수월하게 진행하기 위해 run.sh 파일을 수정한다.

```
pi@raspberrypi:~/Linux-Client $ vi run.sh
```

```
1 #!/usr/bin/env bash  
2 mkdir -p build  
3 cd build  
4 cmake -DCMAKE_BUILD_TYPE=Release -DDAEMON=OFF -DRASPBerry=ON ../  
5 cd ..  
6 make thingier  
7 ./thingier
```



Minutes Cloud Platform 사용 환경 구성하기



HANBACK ELECTRONICS.,LTD

- run.sh 파일의 수정이 완료되면 “./run.sh” 명령을 통해 실행한다.
run.sh 파일을 실행하면 소스코드 컴파일을 진행한다.
컴파일이 에러가 없이 완료가 되면 thinger 라는 실행파일이 생성되고
이 파일을 자동으로 실행하여 Cloud 에 접속을 시도 한다.

```
pi@raspberrypi:~/Linux-Client $ ./run.sh
-- OpenSSL Version: 1.0.1t
-- Configuring done
-- Generating done
-- Build files have been written to: /home/pi/Linux-Client
Scanning dependencies of target thinger
[ 50%] Building CXX object CMakeFiles/thinger.dir/src/main.cpp.o
[100%] Linking CXX executable thinger
[100%] Built target thinger
[3733714.903000]: Not connected!
[3733715.074000]: Connecting to iot.thinger.io:25202 ...
[3733716.274000]: Connected!
[3733716.274000]: Authenticating...
[3733716.558000]: Authenticated!
```



Minutes Cloud Platform 사용 환경 구성하기

- 웹페이지에서 디바이스 정보를 확인해보면 디바이스 상태가 Connected 로 변경된 것을 확인 할 수 있다.

Device	Description	Last Connection	State
<input type="checkbox"/> hbe_test_device	hanback iot test device	2019-05-22 16:57:57 +0900	Connected

- 디바이스 상세 정보를 확인해 보기 위해 디바이스 명칭을 클릭하여 상세 정보 화면으로 넘어간다. 앞서 등록과정과 같은 화면이지만 상세 정보들이 변경된 것을 확인할 수 있다.




Minutes Cloud Platform 사용 환경 구성하기



HANBACK ELECTRONICS.,LTD

- API Explorer를 확인해 보면 현재 등록되어 있는 센서나 액추에이터들의 목록과 제어가 가능하다.

HBE_TEST_DEVICE API 

sum - Private

Resource Input

value1 Number

1

value2 Number


3

Resource Output

result Number

4

Options

>_ Run  Show query