

## SMARTFARM

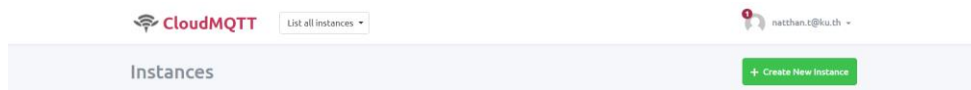
### อุปกรณ์

- 1.Raspberry Pi
- 2.switch
- 3.relay 6 channel
- 4.smartphone
- 5.Account Heroku
- 6.Account Cloudmqtt
- 7.Account mongoDB Atlas

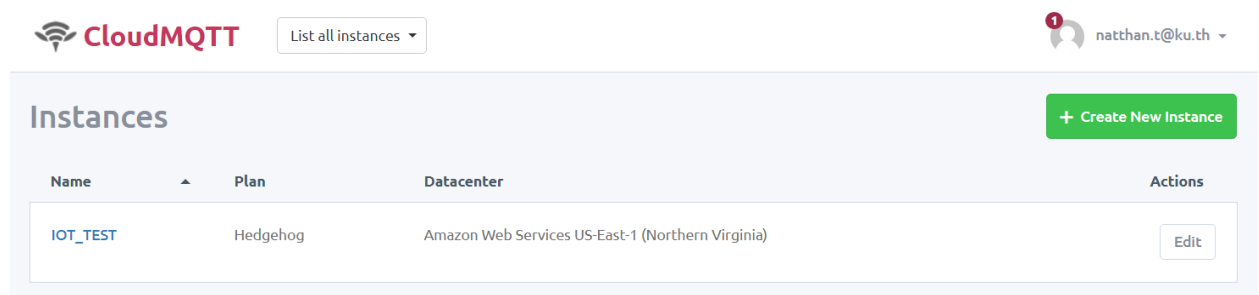
### ขั้นตอนการติดตั้ง

#### -Cloudmqtt

- 1.ทำการSign up ที่ <https://www.cloudmqtt.com/>
- 2.ทำการLogin เข้าสู่เว็บ <https://www.cloudmqtt.com/>
- 3.click ที่  เพื่อสร้างInstanceใหม่



- 4.ทำการกรอกข้อมูลจนเสร็จจะมี Instance ที่ได้จากการ Create ตามรูปด้านล่าง



5.ทำการclick ที่ Instance ที่ได้สร้างจากขั้นตอนที่ผ่านมา จะได้ตามรูปด้านล่าง

## Details

### Instance info

Server tailor.cloudmqtt.com

User mvmmwqiq

Restart

Password i9wpB79zENJg



Port 12106

SSL Port 22106

Websockets Port (TLS only) 32106

Connection limit 25

### Active Plan



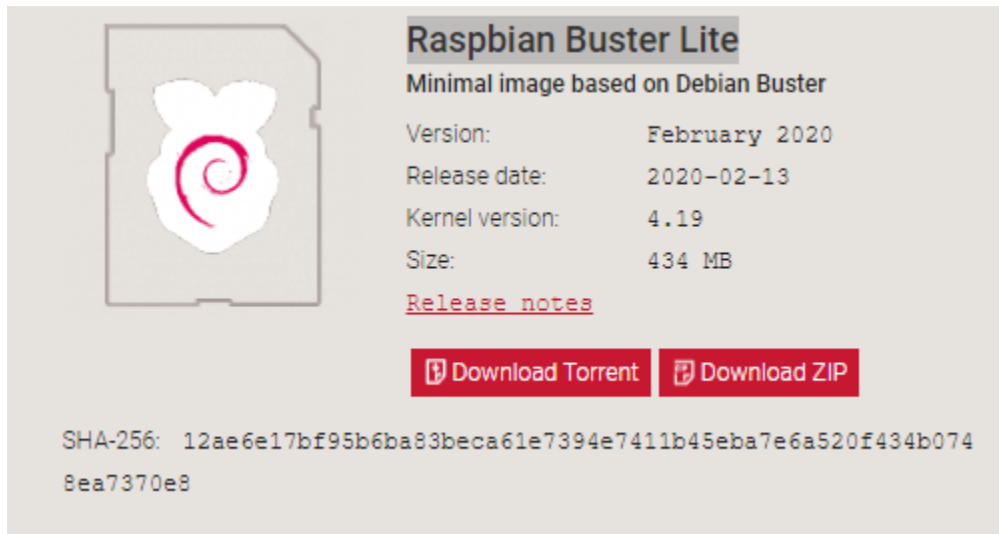
Upgrade Instance

ในขั้นตอนนี้ผู้ใช้ จะได้รับถึง Server,User,Password,Port,SSL Port,Websockets Port,Connection limit ซึ่งต้องนำไปใช้ในการติดตั้งส่วนอื่นๆต่อไป

## -Raspberry Pi4

1.ติดตั้ง OS (Raspbian Buster Lite) ให้กับ Raspberry Pi

Download : <https://www.raspberrypi.org/downloads/raspbian/>



2.ตั้งค่าการเชื่อมต่อ wifi ให้กับ Raspberry Pi

2.1 ใช้ command ด้านล่างเพื่อแก้ไขไฟล์ wpa\_supplicant.conf

```
pi@raspberrypi:~ $ sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

2.2 ไปที่ด้านล่างของไฟล์และทำการเพิ่มตามรูปด้านล่าง

ssid ใส่เป็นชื่อ access pointที่เราต้องการให้Raspberry Pi เชื่อมต่อ

psk ใส่เป็น Password ของ access point ที่เราจะเชื่อมต่อ

```
network={
    ssid="testing"
    psk="testingPassword"
}
```

3.ทำการติดตั้ง pip3 ให้กับ Raspberry Pi

```
pi@raspberrypi:~ $ sudo apt-get install python-pip3
```

4.ทำการติดตั้ง library ที่จำเป็นต้องใช้ โดยใช้ command ด้านล่าง

```
pi@raspberrypi:~ $ pip3 install RPi.GPIO
```

```
pi@raspberrypi:~ $ pip3 install paho.mqtt
```

```
pi@raspberrypi:~ $ pip3 install Adafruit_Python_ADS1x15/
```

5.เปิดfile python ชื่อ 6030300300.py ที่อยู่ใน folder pi

5.1 ทำการเปลี่ยนcode ส่วน

```
-client.username_pw_set(username="your username",password="yourpassword")
```

```
-client.connect("your server",your port,60)
```

ให้เป็นของตนเอง ตามที่สมัครและสร้างInstanceไว้จาก <https://www.cloudmqtt.com/>

ตัวอย่างตามรูปด้านล่าง

```
client = mqtt.Client()|
client.username_pw_set(username="mvmmwqig",password="i9wpB79zENJg")
client.connect("tailor.cloudmqtt.com",12106,60)
```

ในส่วนของclient.subscribeนั้น จะต้องตั้งให้ตรงกับtopicที่เราต้องการจะให้ตัวpiนั้นรอ

subscribeจากcloudmqtt

```
client.subscribe([("valveM1",0), ("valveM2",0), ("valveM3",0), ("valveM4",0), ("valveM5",0), ("motor",0), ("mode",0), ("volt0",0), ("volt1",0)])
```

5.2 ทำการsetup pin GPIO ตามที่ต้องการว่าต้องให้Pinไหนเป็นตัวรับINPUT จากswitch หรือ  
จ่าย OUTPUT เพื่อควบคุมการทำงานของRelayต่างๆ

```
GPIO.setup(37,GPIO.IN,pull_up_down=GPIO.PUD_UP) # switch
GPIO.setup(22,GPIO.OUT)
GPIO.setup(29,GPIO.OUT)
GPIO.setup(31,GPIO.OUT)
GPIO.setup(33,GPIO.OUT)
GPIO.setup(35,GPIO.OUT)
GPIO.setup(36,GPIO.OUT)
```

และทำการเปลี่ยนแปลงpin GPIO ทั้งหมดของcode ตามที่ได้setup ไว้

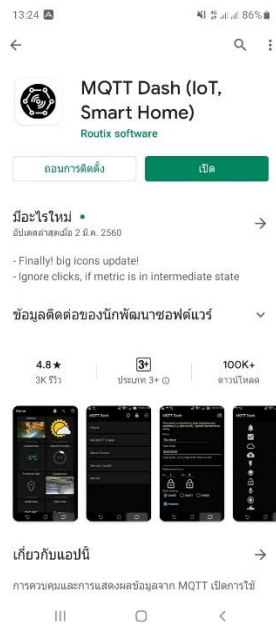
6.ทำการต่อขาPin เชื่อมต่อกับ Relay และ Switch ตามที่กำหนดไว้บน code

7.นำไฟไปทำการแก้ไขเรียบร้อยแล้วไปเก็บไว้ในRaspberry PI และทำการRun code ผ่าน Command  
ด้านล่าง

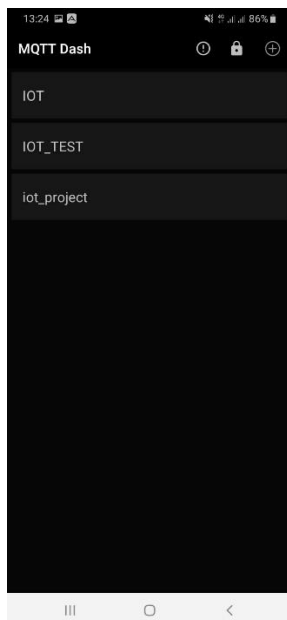
```
pi@raspberrypi:~/natthan $ python3 run.py
```

## -SmartPhone

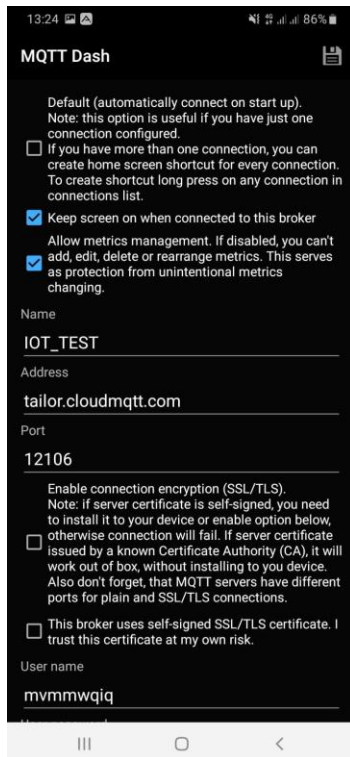
### 1.ทำการติดตั้งApplication MQTT Dash



### 2.เปิดApplication MQTT Dash แล้วกดที่ปุ่มเครื่องหมาย + มุมบนขวาของApplication



### 3. ทำการกรอกข้อมูลของcloudmqttที่ได้สมัครไว้ เพื่อทำการเชื่อมต่อระหว่างApplicationและcloudmqtt



13:24 86%

**MQTT Dash**

Default (automatically connect on start up).  
Note: this option is useful if you have just one connection configured.

☐ If you have more than one connection, you can create home screen shortcut for every connection. To create shortcut long press on any connection in connections list.

☒ Keep screen on when connected to this broker

Allow metrics management. If disabled, you can't add, edit, delete or rearrange metrics. This serves as protection from unintentional metrics changing.

Name  
**IOT\_TEST**

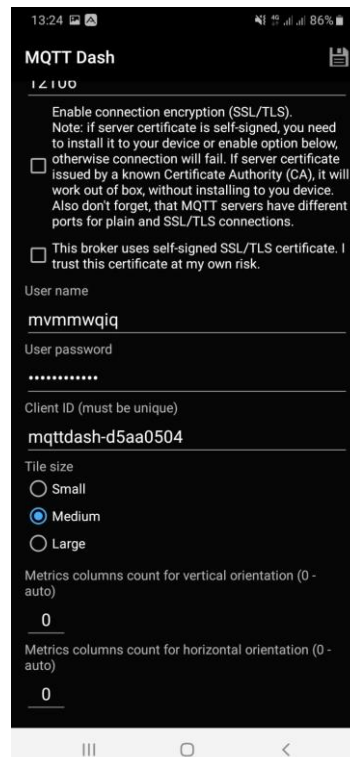
Address  
**tailor.cloudmqtt.com**

Port  
**12106**

Enable connection encryption (SSL/TLS).  
Note: if server certificate is self-signed, you need to install it to your device or enable option below, otherwise connection will fail. If server certificate issued by a known Certificate Authority (CA), it will work out of box, without installing to you device. Also don't forget, that MQTT servers have different ports for plain and SSL/TLS connections.

☐ This broker uses self-signed SSL/TLS certificate. I trust this certificate at my own risk.

User name  
**mvmmwqiq**



13:24 86%

**MQTT Dash**

Enable connection encryption (SSL/TLS).  
Note: if server certificate is self-signed, you need to install it to your device or enable option below, otherwise connection will fail. If server certificate issued by a known Certificate Authority (CA), it will work out of box, without installing to you device. Also don't forget, that MQTT servers have different ports for plain and SSL/TLS connections.

☐ This broker uses self-signed SSL/TLS certificate. I trust this certificate at my own risk.

User name  
**mvmmwqiq**

User password  
\*\*\*\*\*

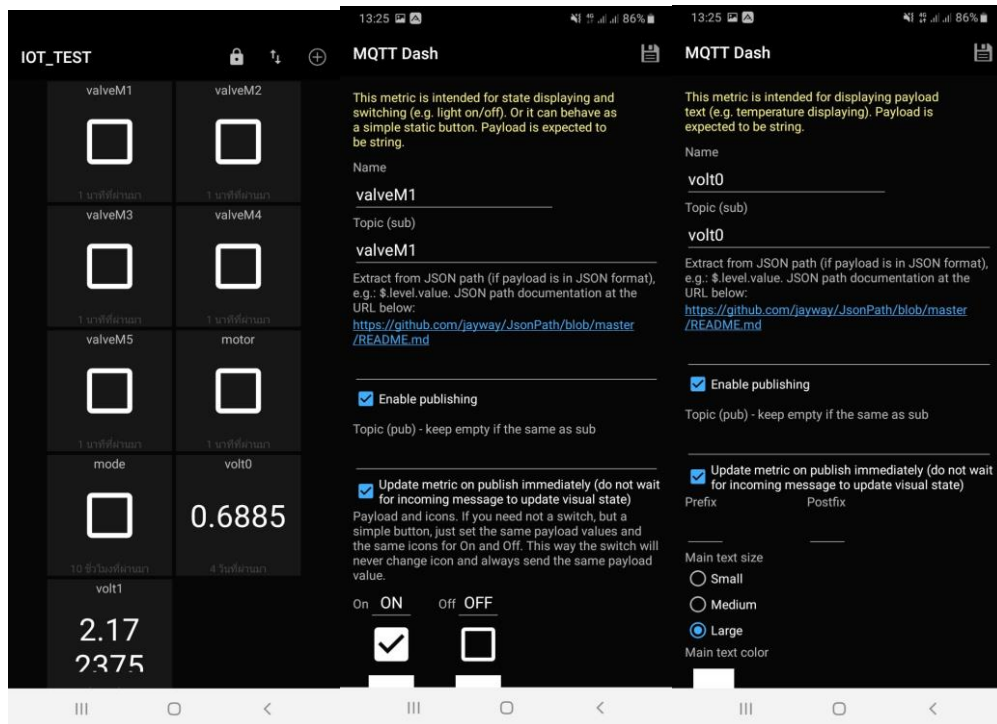
Client ID (must be unique)  
**mqttdash-d5aa0504**

Tile size  
☐ Small  
☒ Medium  
☐ Large

Metrics columns count for vertical orientation (0 - auto)  
**0**

Metrics columns count for horizontal orientation (0 - auto)  
**0**

4.หลังจากนั้นทำการสร้างปุ่มต่างๆที่ผู้ใช้งานต้องการ และทำการกำหนดtopicให้ตรงกับcode pythonที่สร้างไว้ด้านบน โดยการclick ที่เครื่องหมาย + มุมขวบนของ Application

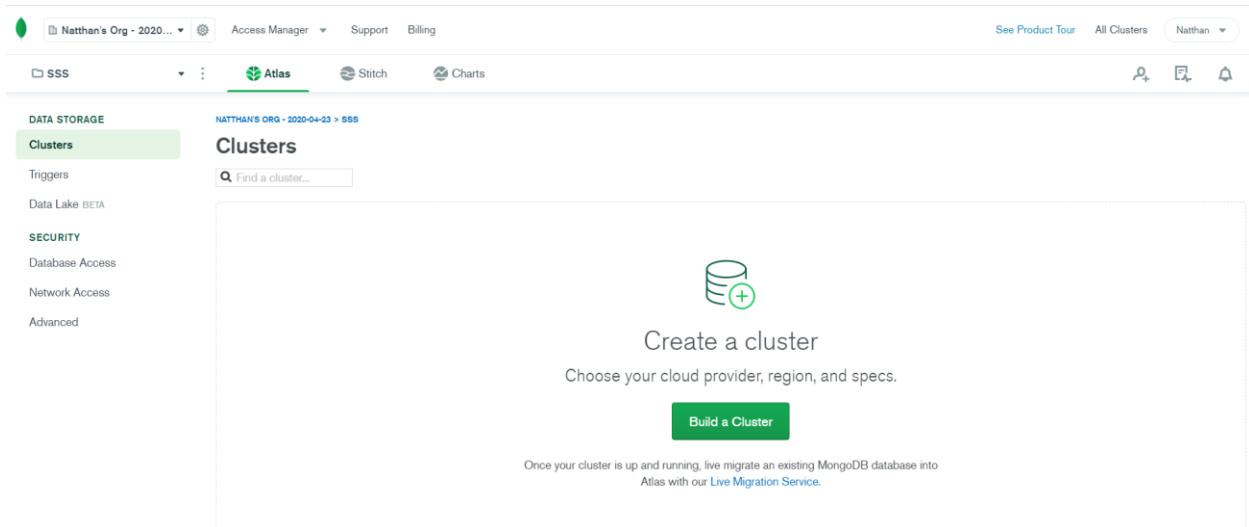


ทำแบบนี้กับทุกๆปุ่มที่เราต้องการสร้าง

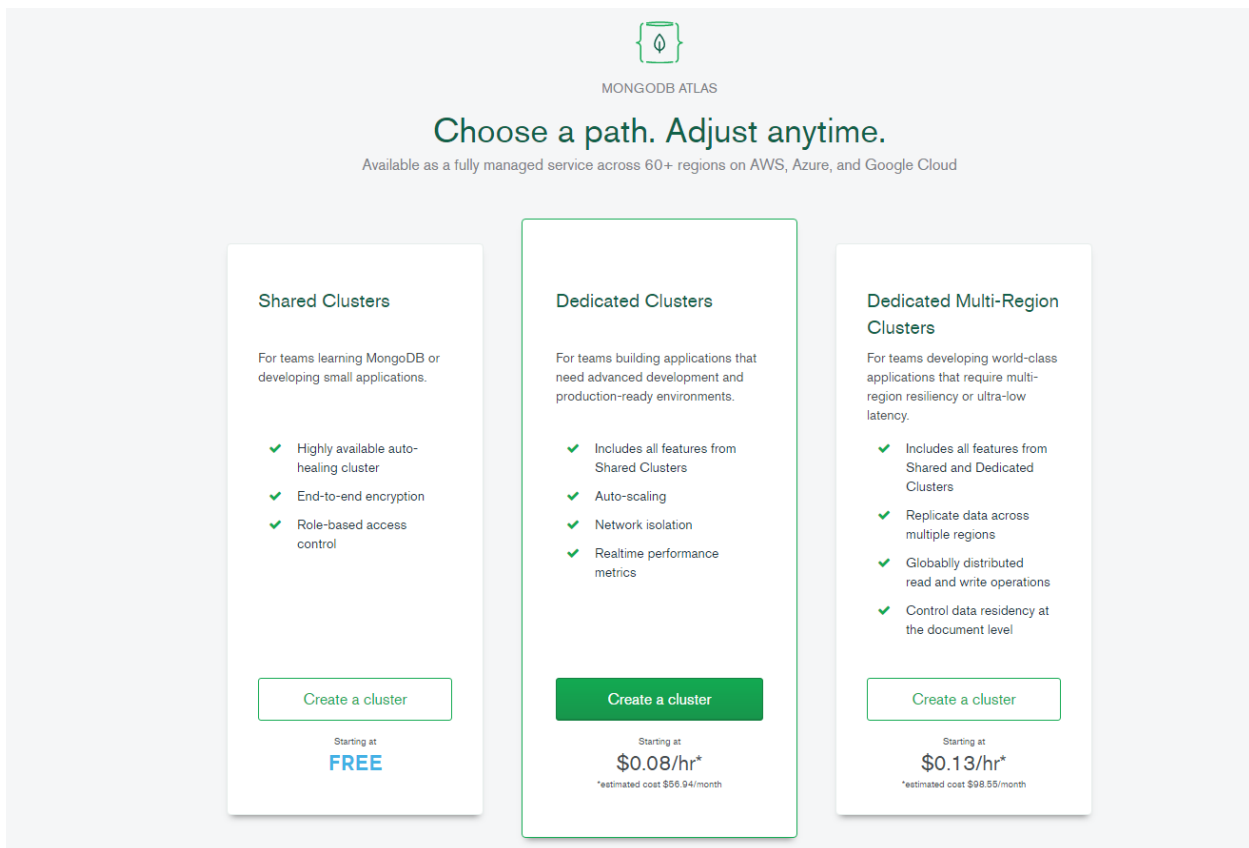


## -mongoDB

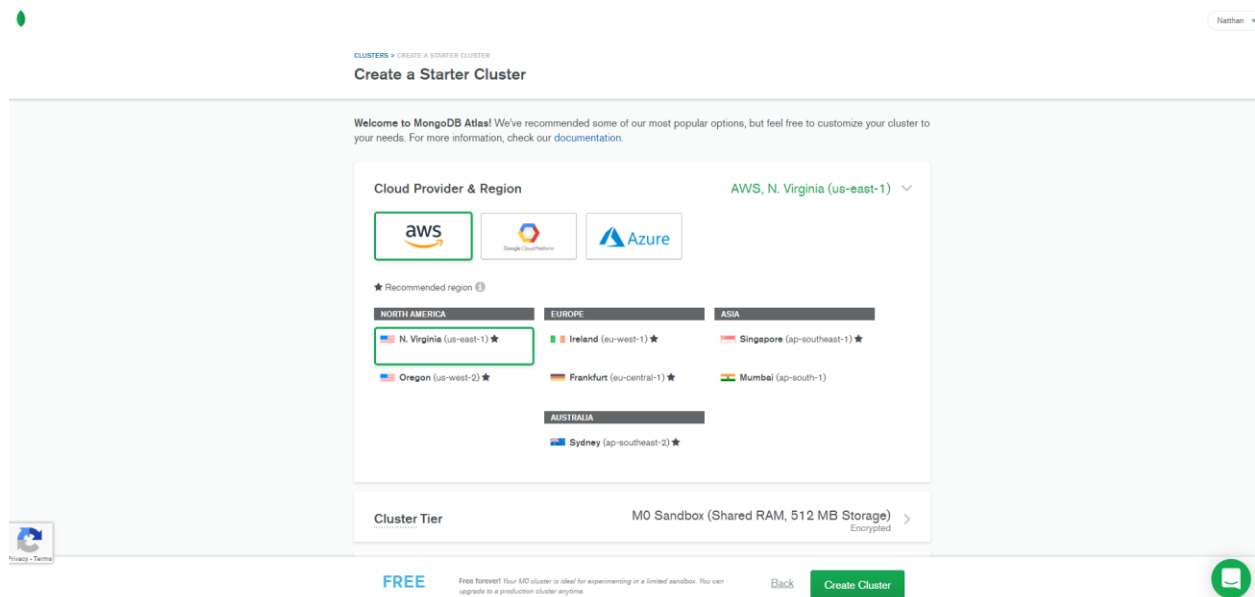
1.ทำการSign in ที่ <https://www.mongodb.com/cloud/atlas> แล้วทำการกด Build a Cluster



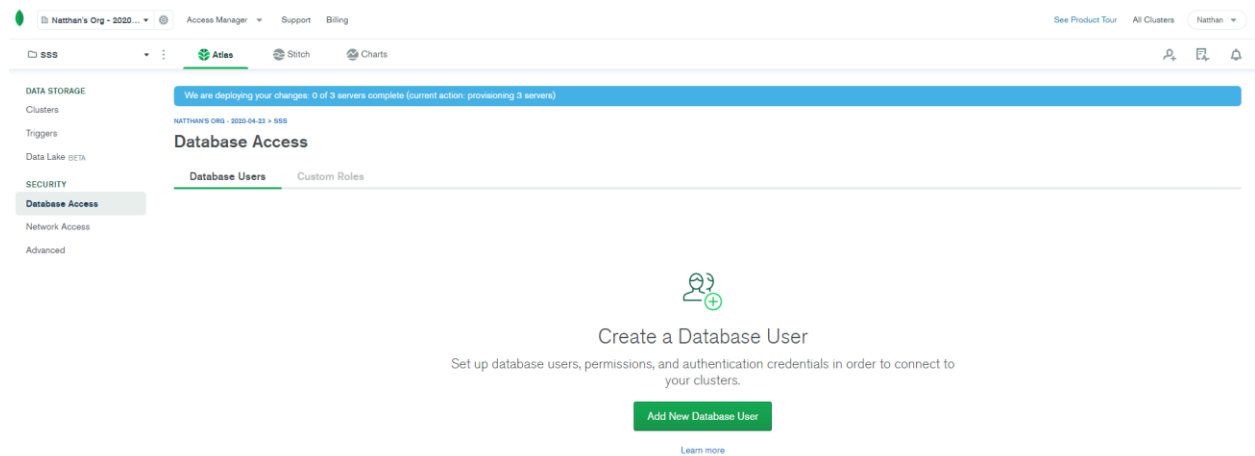
2.เลือก Create a cluster แบบ FREE



### 3. กด Create Cluster



### 4. กด Database Access แล้วกด ADD NEW DATABASE USER



5. ตั้ง username และ password แล้วเลือก Atlas admin แล้วกด Add User

×

## Add New Database User

Choose Authentication Method

PASSWORD

CERTIFICATE

**Password Authentication**  
MongoDB uses [SCRAM](#) as its default authentication method.

admin

e.g. new-user\_31

admin

HIDE

🔑 Autogenerate Secure Password

📋 Copy

Database User Privileges

Atlas admin

Read and write to any database

Only read any database

Select Custom Role ▾

[Add Default Privileges](#)

☐ This user is temporary and will be deleted in

6 hours ▾

Cancel

Add User

6. หลังจากนั้นไปที่ Network Access แล้ว กด ADD IP ADDRESS แล้ว เลือก ALLOW ACCESS FROM ANYWHERE แล้ว กด Confirm

Nathan's Org - 3250

Access Manager

Support

Billing

SSS

Atlas

Slack

Charts

DATA STORAGE

Clusters

Triggers

Data Lake (ETA)

SECURITY

Database Access

Network Access

Advanced

We are deploying your changes (current action: configuring)

NATHAN'S ORG - 3250-04-02 • SSS

Network Access

IP Whitelist

Peering

Private Endpoints

Add IP Address

Learn more

Add IP Whitelist Entry

Atlas only allows client connections to a cluster from entries in the project's whitelist. Each entry should either be a single IP address or a CIDR-notated range of addresses. [Learn more.](#)

ADD CURRENT IP ADDRESS

ALLOW ACCESS FROM ANYWHERE

Whitelist Entry:

0.0.0.0/0

Comment:

Optional comment describing this entry

☐ This entry is temporary and will be deleted in

6 hours ▾

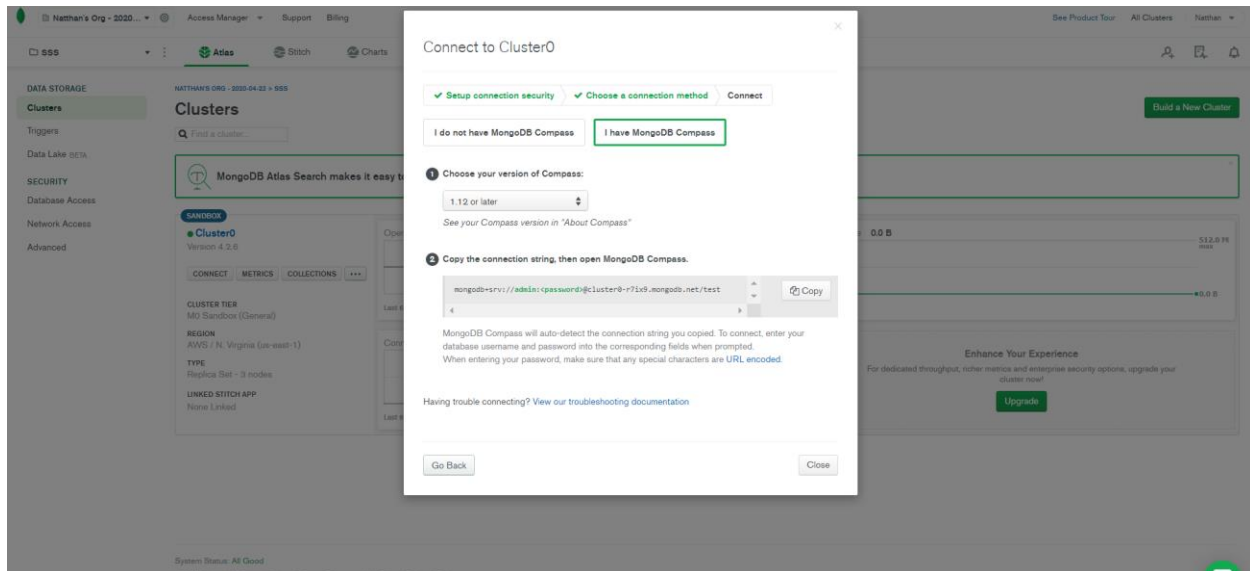
Cancel

Confirm

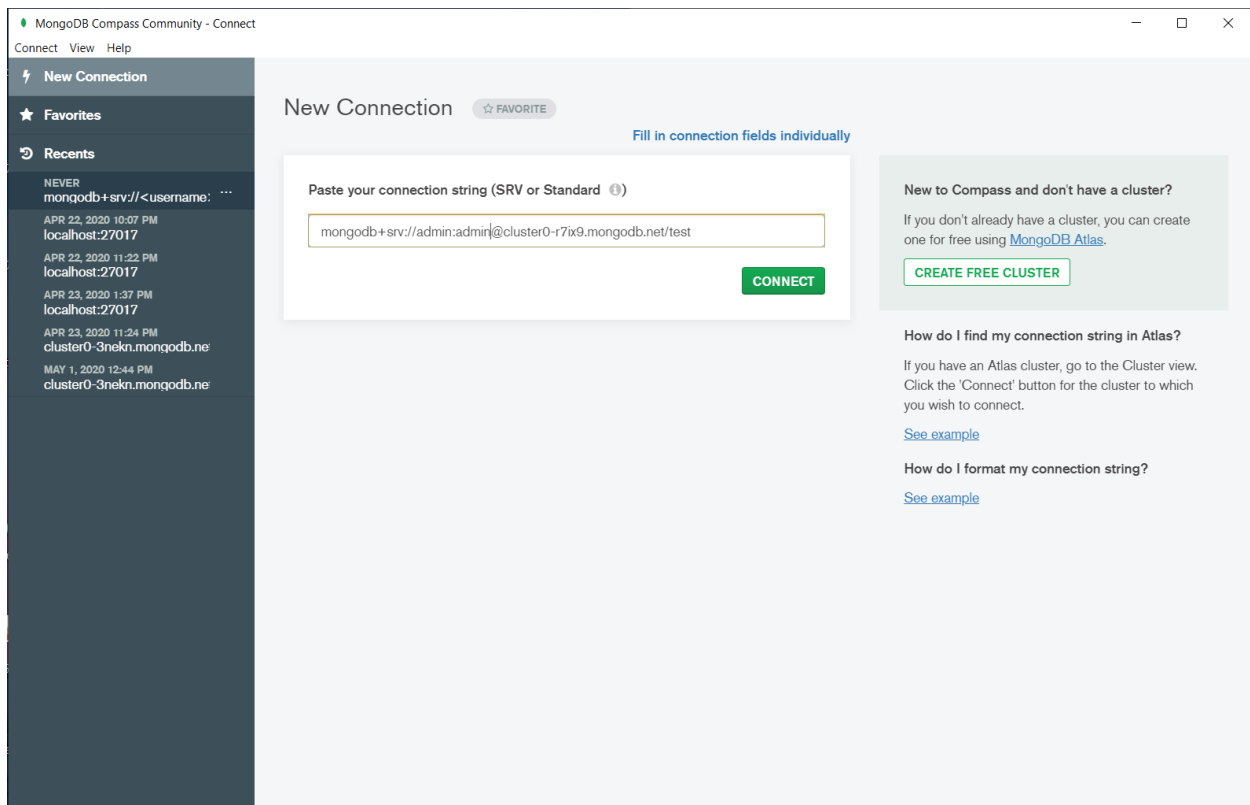
7.หลังจากนั้นกด Clusters แล้ว กดที่ Connect แล้วกด Connect using MongoDB Compass >>

Download Compass

8. copy ส่วนนี้เอาไว้ connect ใน mongoDB Compass

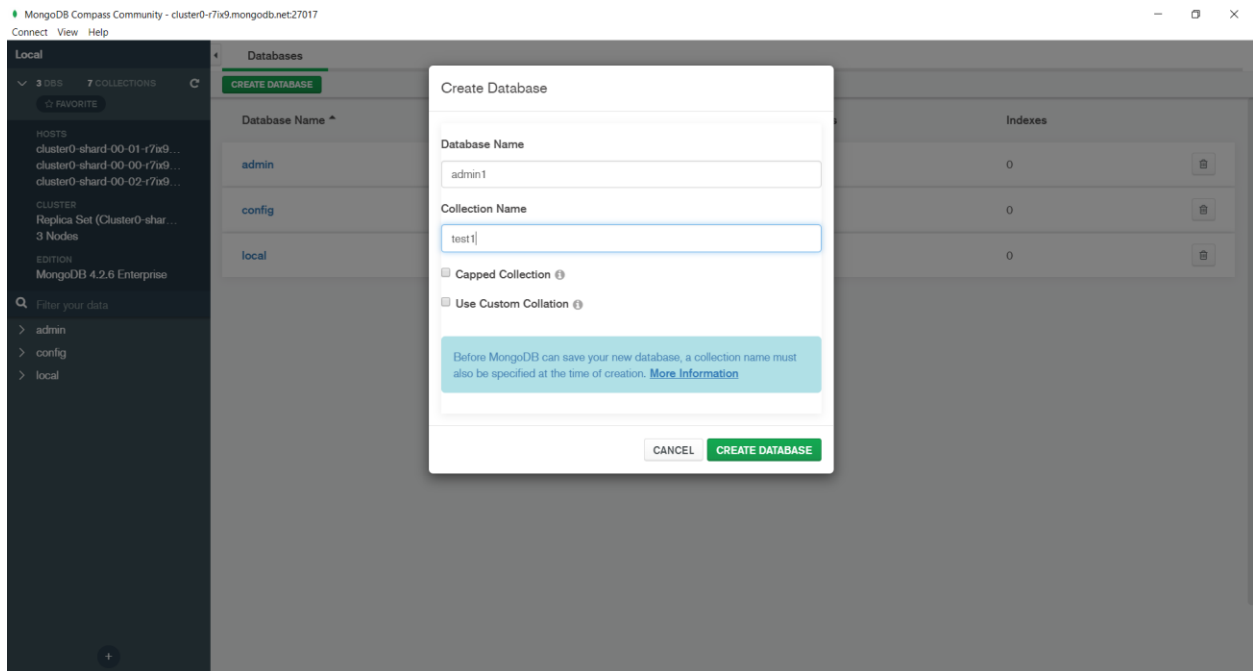


9. นำสิ่งที่ copy ในขั้นตอนก่อนหน้าไปวางตามรูปด้านล่าง



10.เมื่อconnect ได้ ให้กด CREATE DATABASE เพื่อสร้าง DATABASE และ collection เสร็จแล้วกด

## CREATE DATABASE



หลังจากขั้นตอนนี้จะได้database และ collection ที่พร้อมสำหรับเก็บข้อมูลแล้ว

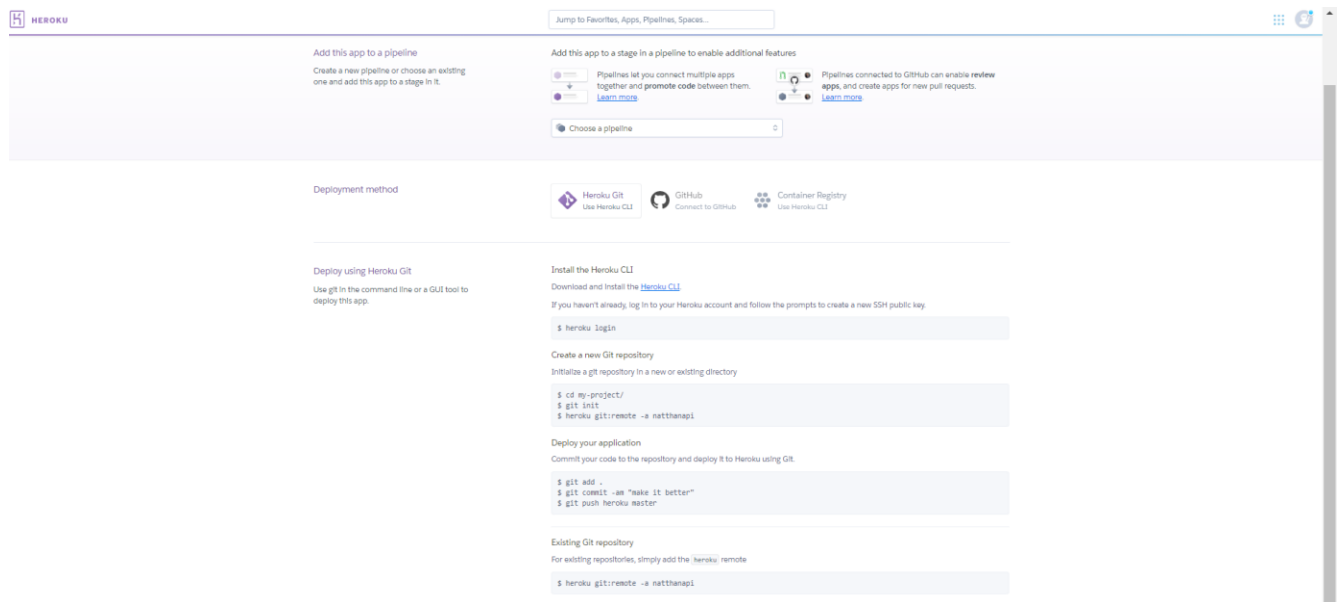
## -Heroku

1.เข้าเว็บ <https://herokuapp.com/>

2.สมัครและทำการlogin

3.กด New แล้วกด Create new app และทำการCreate appจะได้หน้านี้

โดยทำการCreate 2 app สำหรับอัปโหลดไฟล์frontendและbackend




4. ใน folder backend ต้องแก้ไขไฟล์ server.js ส่วนต่างๆ ดังนี้

-const url = 'mongodb+srv://admin:admin@cluster0-

3nekn.mongodb.net/test?retryWrites=true&w=majority'; โดยเปลี่ยนurlให้ตรงกับที่สร้างdbไว้ในขั้นตอนที่ผ่านมา


สามารถหาurlได้ตามรูปด้านล่าง โดยกดที่connect



MongoDB Atlas Search makes it easy to build simple, integrated search capabilities on top of your data in the cloud.

Try the Beta Now

SANDBOX


**Cluster0**  
Version 4.2.6

CONNECT METRICS COLLECTIONS ...

**CLUSTER TIER**  
M0 Sandbox (General)

**REGION**  
AWS / N. Virginia (us-east-1)

**TYPE**  
Replica Set - 3 nodes

**LINKED STITCH APP**  
None Linked

Operations R: 0 W: 0 100.0/s

Last 6 Hours

Logical Size 0.0 B 512.0 Mi max

Last 6 Hours

Connections 1 500 max


Last 6 Hours

**Enhance Your Experience**

For dedicated throughput, richer metrics and enterprise security options, upgrade your cluster now!

Upgrade

จากนั้นเลือก Connect your application



Setup connection security

Choose a connection method

Connect


**Choose a connection method** [View documentation](#)

Get your pre-formatted connection string by selecting your tool below.




**Connect with the mongo shell**

Interact with your cluster using MongoDB's interactive Javascript interface



**Connect your application**

Connect your application to your cluster using MongoDB's native drivers



**Connect using MongoDB Compass**

Explore, modify, and visualize your data with MongoDB's GUI

Go Back

Close

เลือกDRIVER ให้ตรงกับภาษาที่เราเขียน และกด **copy** ตามรูปด้านล่าง เราก็จะได้urlมา

×

## Connect to Cluster0

✓ Setup connection security

✓ Choose a connection method

Connect

1

Select your driver and version

DRIVER

Node.js

VERSION

3.0 or later

2

Add your connection string into your application code

Connection String Only

Full Driver Example

mongodb+srv://admin:<password>@cluster0-r7ix9.mongodb.net/test?retryl

Copy

Replace **<password>** with the password for the user, **admin**, and ensure all special characters are URL encoded.

Having trouble connecting? [View our troubleshooting documentation](#)

Go Back

Close



-แก้ส่วนนี้ให้เป็น cloud mqtt ของเรา

```
var options = {
  port: 12106,
  host: 'mqtt://tailor.cloudmqtt.com',
  clientId: 'mqttjs_' + Math.random().toString(16).substr(2, 8),
  username: 'mvmmwqiq',
  password: 'i9wpB79zENJg',
  keepalive: 60,
  reconnectPeriod: 1000,
  protocolId: 'MQIsdp',
  protocolVersion: 3,
  clean: true,
  encoding: 'utf8'
};
```

-แก้ส่วนนี้ให้เป็นชื่อdatabaseที่สร้างไว้

```
const dbName = "admin1";
```

-แก้ส่วนนี้ให้เป็นชื่อcollectionที่สร้างไว้ในทุกๆที่บนcode

```
db.collection('test1').
```

-แก้ไขส่วนsubscribe ให้ตรงกับtopic ที่ต้องการรอรับข้อมูลเพื่อส่งต่อไปเก็บที่mongoDB

```
client.subscribe(['valveM1', 'valveM2', 'valveM3', 'valveM4', 'valveM5', 'motor', 'mode', 'volt0', 'volt1'], ()=>{
  client.on('message', function(topic, message, packet) {
    if(topic== 'mode'){
      mode = String(message);
      var data = {
        device : topic,
        payload : String(message),
        date : String(Moment(new Date()).tz('Asia/Bangkok').format())
      }
      addValue(data);
      payload=String(message);
      console.log(topic+" "+payload);
    }

    if(mode=="OFF" && (topic == 'valveM1' || topic == 'valveM2' || topic == 'valveM3' || topic == 'valveM4' || topic == 'valveM5' || topic == 'motor')){
      var data = {
        device : topic,
        payload : String(message),
        date : String(Moment(new Date()).tz('Asia/Bangkok').format())
      }
      addValue(data);
      payload=String(message);
      console.log(topic+" "+payload);
    }

    if(topic=='volt0' || topic=='volt1'){
      var data = {
        device : topic,
        payload : String(message),
        date : String(Moment(new Date()).tz('Asia/Bangkok').format())
      }
      addValue(data);
    }
  })
});
```

5. ใน folder frontend จะต้องทำการแก้ไขดังนี้

-ที่ไฟล์ a.php จะเป็นไฟล์ที่ทำหน้าที่ดึงค่าจากmongoDB Atlasมาแสดงจะต้องแก้ไขในส่วนที่อยู่mongoDB ให้เป็นของตนเอง

```
$client = new MongoClient(
    'mongodb+srv://admin:admin@cluster0-3nekn.mongodb.net/test?retryWrites=true&w=majority');
```

-แก้ไข admin1 ที่ขีดเส้นสีแดง ให้เป็นชื่อDBที่สร้างไว้

-แก้ไข test1 ที่ขีดเส้นสีแดง ให้เป็น collection ที่สร้างไว้

```
$admin1 = $client->admin1;
$test1 = $admin1->test1;
```

```
$cursor = $test1->find();
```

-ที่ไฟล์ index.php ให้ทำการแก้ไขส่วนต่างๆดังนี้

-แก้ไขให้เป็นcloud mqttของตนเอง

```
client = new Paho.MQTT.Client("tailor.cloudmqtt.com",32106,"client_id");
var options = {
    useSSL: true,
    userName: "mvmmwqiq",
    password: "i9wpB79zENJg",
    onSuccess:onConnect,
}
```

-ให้message.destinationNameของทุกๆfunctionให้เป็นtopic ที่เราต้องการจะส่งไปถึง

```
function send_mqtt_Valve1(){
    var checkBox = document.getElementById("valveM1");

    if(checkBox.checked == true){
        message = new Paho.MQTT.Message("ON");
        message.destinationName = "valveM1";
        client.send(message);
    }
    else if(checkBox.checked == false){
        message = new Paho.MQTT.Message("OFF");
        message.destinationName = "valveM1";
        client.send(message);
    }
}

function send_mqtt_Valve2(){
    var checkBox = document.getElementById("valveM2");

    if(checkBox.checked == true){
        message = new Paho.MQTT.Message("ON");
        message.destinationName = "valveM2";
        client.send(message);
    }
    else if(checkBox.checked == false){
        message = new Paho.MQTT.Message("OFF");
        message.destinationName = "valveM2";
        client.send(message);
    }
}
```

6.หลังจากแก้ไขไฟล์backendและfrontendแล้วให้ทำการอัปโหลดไฟล์ขึ้นHeroku ตามขั้นตอนด้านล่าง เป็นอันเสร็จ

Install the Heroku CLI

Download and Install the [Heroku CLI](#).

If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

```
$ heroku login
```

Create a new Git repository

Initialize a git repository in a new or existing directory

```
$ cd my-project/  
$ git init  
$ heroku git:remote -a natthanapi
```

Deploy your application

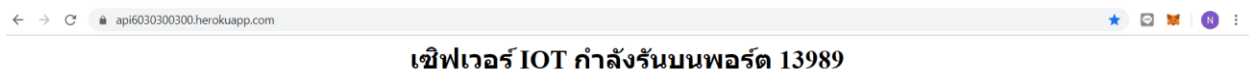
Commit your code to the repository and deploy it to Heroku using Git.

```
$ git add .  
$ git commit -am "make it better"  
$ git push heroku master
```

วิธีการใช้งานและผลการทดลอง

1.Run ไฟล์ 6030300300.py ในRaspberry PI

2.กดopen app ในเว็บ Heroku ทั้ง backend และ frontend



← → ↻ log6030300300.herokuapp.com

LOG IOT						
ValveM1	ValveM2	ValveM3	ValveM4	ValveM5	Motor	Mode
น้ำออก	เติมกรด(Acid)	สาร A	สาร B	ปั้มน้ำวน	ปั้มน้ำเข้า	mode
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Number	DEVICE	STATUS	DATE			
1	mode	OFF	2020-05-02T11:14:12+07:00			
2	valveM1	ON	2020-05-02T11:14:14+07:00			
3	valveM2	ON	2020-05-02T11:14:15+07:00			
4	valveM3	ON	2020-05-02T11:14:15+07:00			
5	valveM4	ON	2020-05-02T11:14:16+07:00			
6	valveM5	ON	2020-05-02T11:14:17+07:00			
7	motor	ON	2020-05-02T11:14:17+07:00			
8	valveM1	OFF	2020-05-02T11:14:22+07:00			
9	valveM2	OFF	2020-05-02T11:14:22+07:00			

3.หลังทำขั้นตอน1,2แล้ว ก็จะสามารถสั่งควบคุมRelayผ่านApplication MQTT Dash บนโทรศัพท์ได้ หรือสั่งผ่านหน้าเว็บfrontendได้เลย

