

LoRa IoT Google Cloud Platform

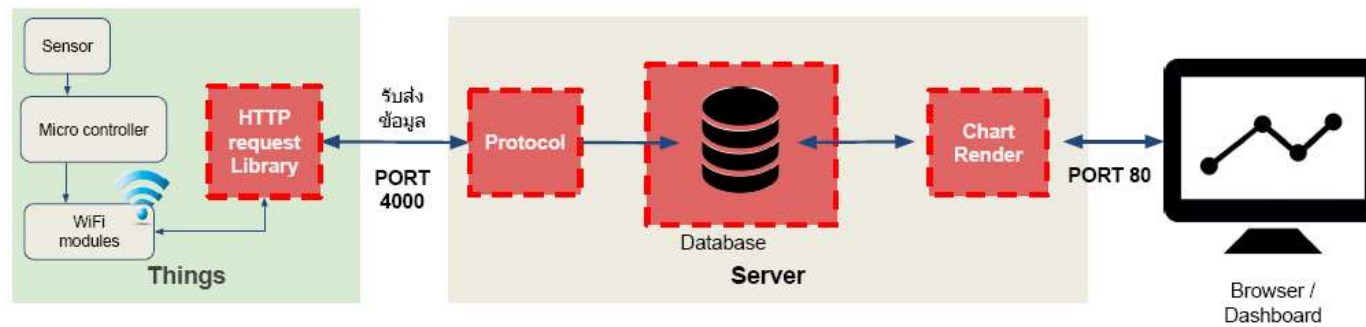
Akarawat Panwilai (Tommy)

เตรียมความพร้อมก่อนเริ่มต้น

1. **Google account** (gmail) : <https://www.google.co.th/>
2. **Chrome browser** : <https://www.google.com/chrome/index.html>
3. **ATOM (code editor)** : <https://atom.io/>
4. **Arduino IDE software** : <https://www.arduino.cc/en/Main/Software>
5. **POSTMAN** : <https://www.getpostman.com/>
6. **Code ที่ใช้ในการอบรม** :
<https://github.com/akarawat/GCPLoRaTraining>
7. **LoRa Module : TTGO OLED LoRa Module**
8. **Sensor (DHT22 - temperature & humidity sensor)**

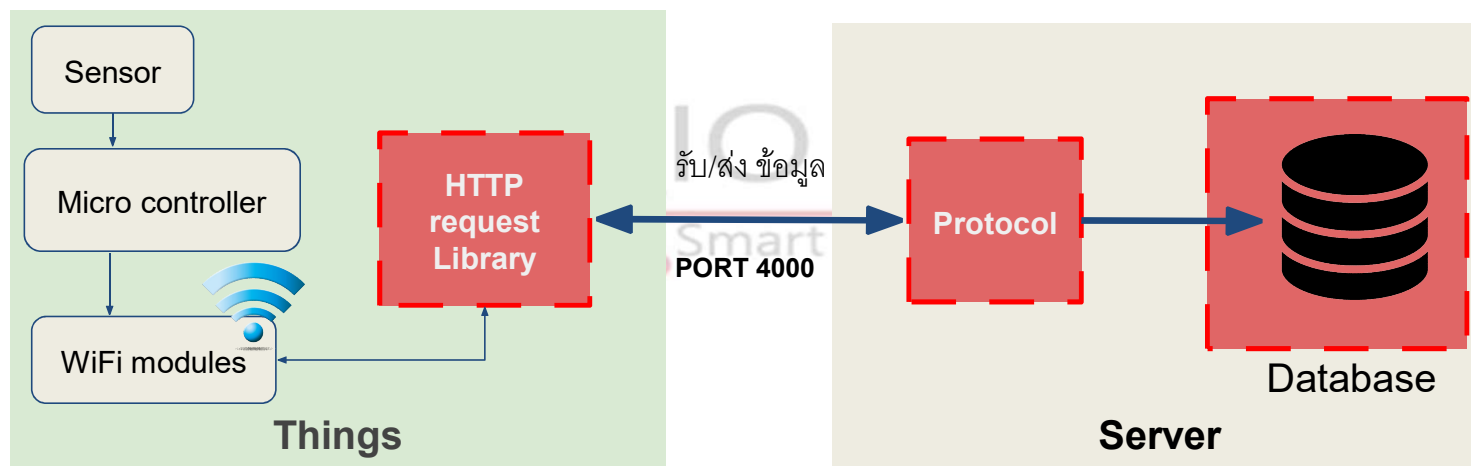


SYSTEM ARCHITECTURE DIAGRAM

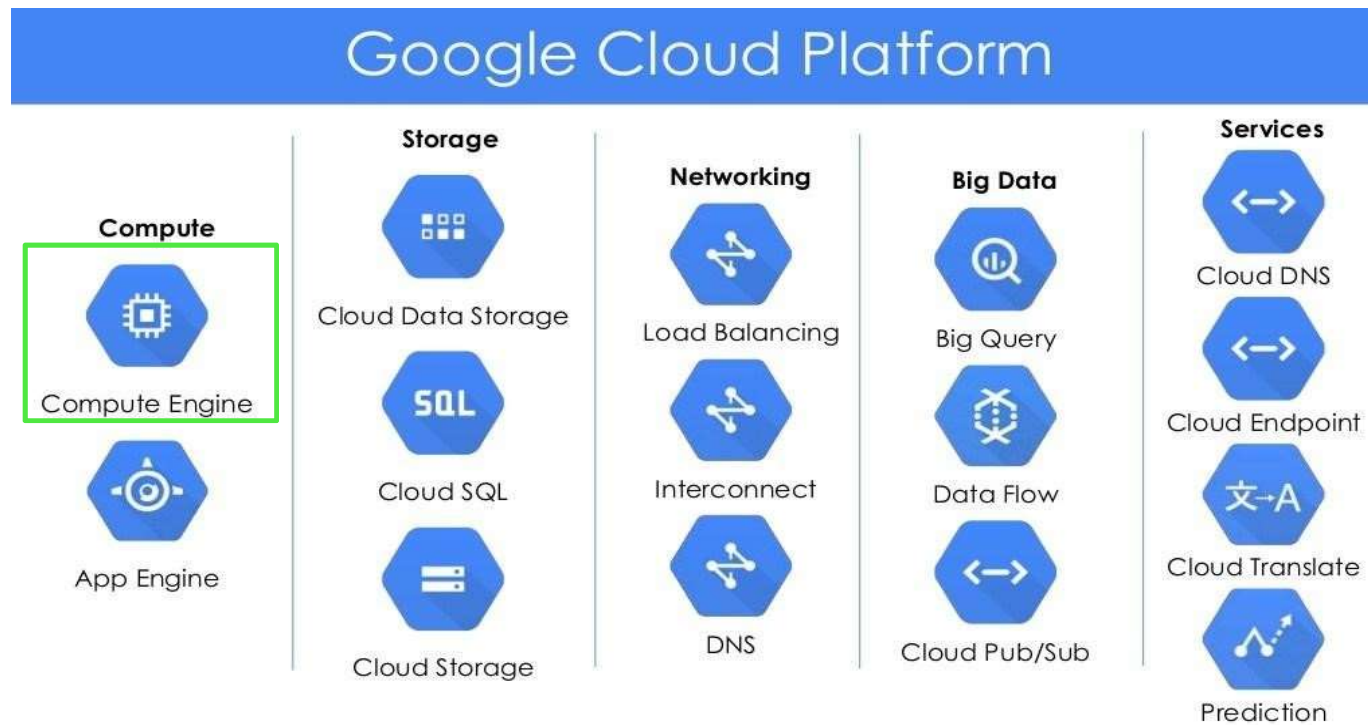


IoT protocol คือช่องทางการสื่อสารระหว่าง IoT device - Server - Database

- เพื่อเก็บข้อมูลจาก IoT device ลง Database.
- เพื่อการควบคุมจาก Server (Dashboard) ไปยัง IoT device.



การให้บริการ ของ Google Cloud Platform



เริ่มต้นใช้งาน Google Cloud Platform

1. <https://cloud.google.com/>



2. ลงทะเบียน

ขั้นตอนที่ 1 จาก 2

ประเทศ

ไทย

ข้อกำหนดในการให้บริการ

- ☒ ฉันยอมรับ ข้อกำหนดในการให้บริการของ Google Cloud Platform และข้อกำหนดในการให้บริการของ บริการและ API ทั้งหมดที่เกี่ยวข้อง.ฉันได้อ่านและยอมรับ ข้อกำหนดในการให้บริการสำหรับการทดลองใช้ Google Cloud Platform ฟรีแล้ว

จำเป็นต้องเลือกเพื่อดำเนินการต่อ

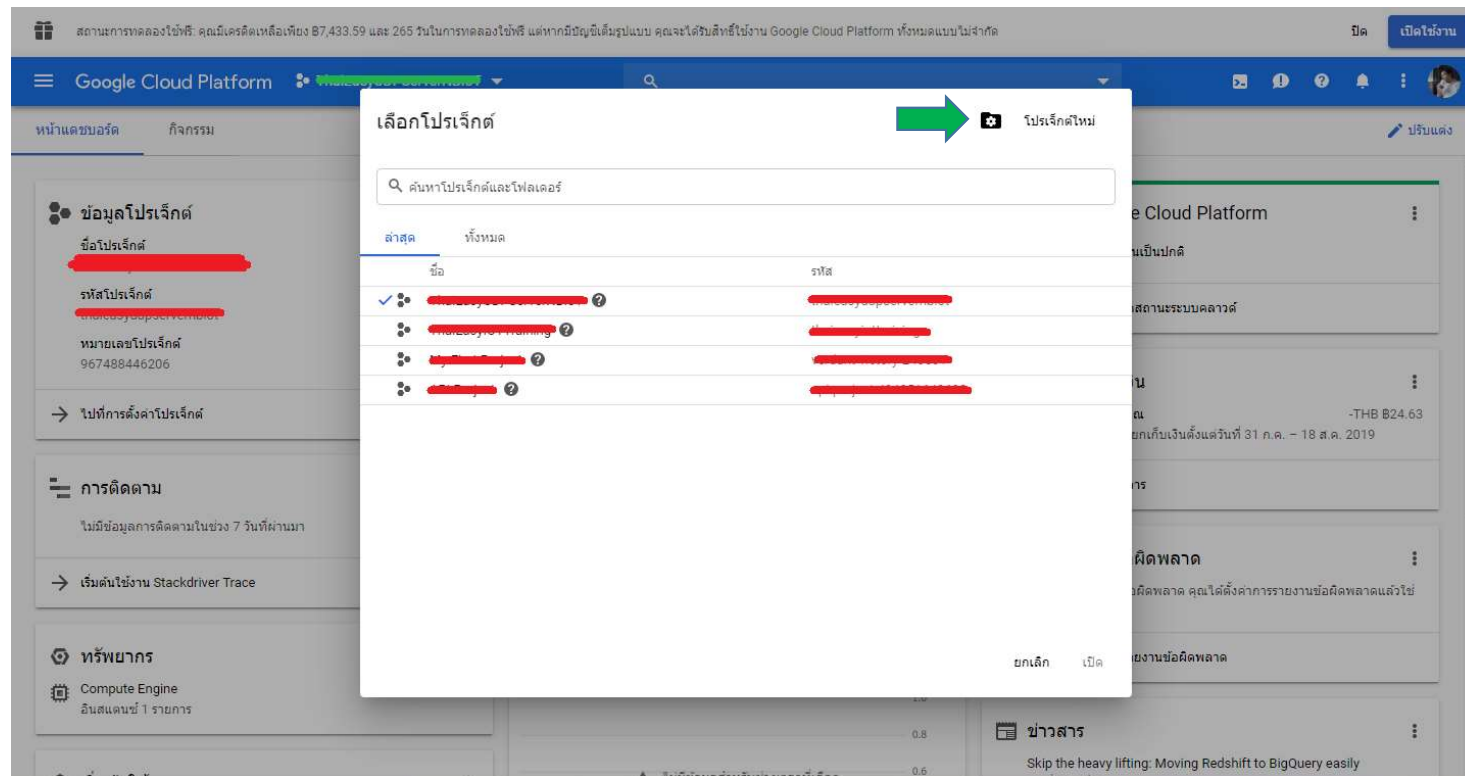
การอัปเดตทางอีเมล

- ☒ ฉันต้องการรับอีเมลข่าวสาร การอัปเดตผลิตภัณฑ์ และข้อเสนอพิเศษจาก Google Cloud และพาร์ทเนอร์ของ Google Cloud เป็นครั้งคราว

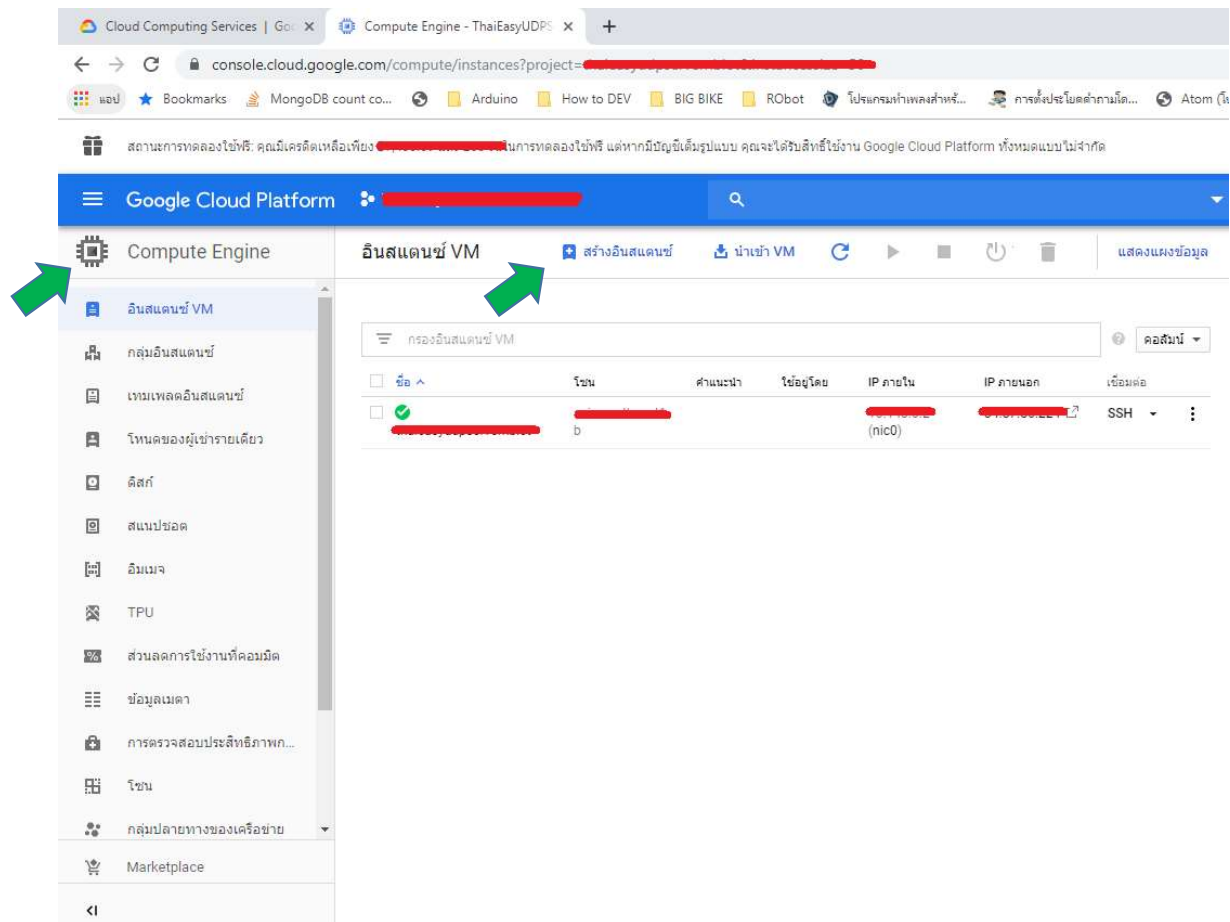
ต่อไป



เริ่มต้นสร้าง Projects



เริ่มต้นสร้าง Instances



Cloud Computing Services | Go: X | Compute Engine - ThaiEasyUDPS X

console.cloud.google.com/compute/instances?project=

Google Cloud Platform

Compute Engine

Instances VM

สร้างอินสแตนซ์ นำเข้า VM แสดงแผงข้อมูล

กรองอินสแตนซ์ VM

ชื่อ	โซน	คำนวณค่า	ใช้ดูโดย	IP ภายใน	IP ภายนอก	เชื่อมต่อ
✓	b			(nic0)		SSH



กำหนดค่า - ปรับแต่ง Virtual Machine

สถานะการทดลองใช้ฟรี: คุณมีเครดิตเหลือเพียง ฿7,433.59 และ 265 วันในการทดลองใช้ฟรี แต่หากมีปัญหาก็มีรูปแบบ คุณจะได้รับสิทธิ์ใช้งาน Google Cloud Platform

Google Cloud Platform

← สร้างอินสแตนซ์

ในการสร้างอินสแตนซ์ VM ให้เลือกหนึ่งในตัวเลือกต่อไปนี้

- อินสแตนซ์ VM ใหม่**
สร้างอินสแตนซ์ VM เดียวตั้งแต่เริ่มต้น
- อินสแตนซ์ VM ใหม่จากเทมเพลต**
สร้างอินสแตนซ์ VM เดียวจากเทมเพลตที่มีอยู่
- Marketplace**
ทำให้โซลูชันแบบพร้อมใช้ใช้งานได้
ในอินสแตนซ์ VM

ชื่อ **loragcpotommy**

ภูมิภาค **asia-southeast1 (สิงคโปร์)** โซน **asia-southeast1-b**

การกำหนดค่าเครื่อง

กลุ่มของเครื่อง

วัดประสิทธิภาพทั่วไป เพิ่มประสิทธิภาพหน่วยความจำ

ประเภทเครื่องสำหรับการใช้งานทั่วไป ซึ่งเพิ่มประสิทธิภาพเพื่อประหยัดต้นทุนและเพิ่มความยืดหยุ่น

รุ่น

ขับเคลื่อนโดยแพลตฟอร์ม CPU ของ Skylake หรือรุ่นก่อนหน้า

ประเภทเครื่อง

g1-small (vCPU 1 รายการ หน่วยความจำ 1.7 GB)

vCPU	หน่วย
แกนประมวลผลรวม 1 แกน	1.7 GB

☒ GPU และแพลตฟอร์ม CPU

คอนเทนเนอร์ ☐ ทำให้โฮสต์คอนเทนเนอร์ใช้งานร่วมกับอินสแตนซ์ VM นี้ [ดูข้อมูล](#)

ดิสก์เปิดเครื่อง ☐ ดิสก์ถาวรแบบมาตรฐาน 10 GB ใหม่

ไฟร์วอลล์

เพิ่มแท็กและกฎไฟร์วอลล์เพื่ออนุญาตให้มีการจราจรของข้อมูลในเครือข่ายที่ระบุจากอินเทอร์เน็ต

- ☒ อนุญาตการจราจรของข้อมูลจาก HTTP
- ☐ อนุญาตการจราจรของข้อมูลจาก HTTPS

☒ การจัดการ การรักษาความปลอดภัย ดิสก์ เครือข่าย ผู้เช่ารายเดียว

ระบบจะใช้เครดิตการทดลองใช้ฟรีของคุณสำหรับอินสแตนซ์ VM นี้ GCP ระดับฟรี

สร้าง ยกเลิก

Equivalent REST or command line

สามารถปรับแต่ง Core CPU หรือ RAM ได้ และ Project นี้

- แนะนำ 1 vCPU
- แนะนำ 1.7 GB



Virtual Machine พร้อมใช้งาน

Google Cloud Platform

Compute Engine

อินสแตนซ์ VM

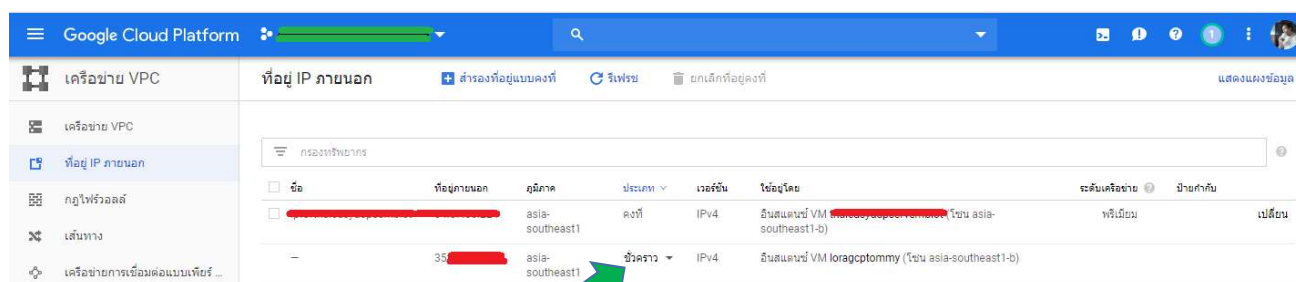
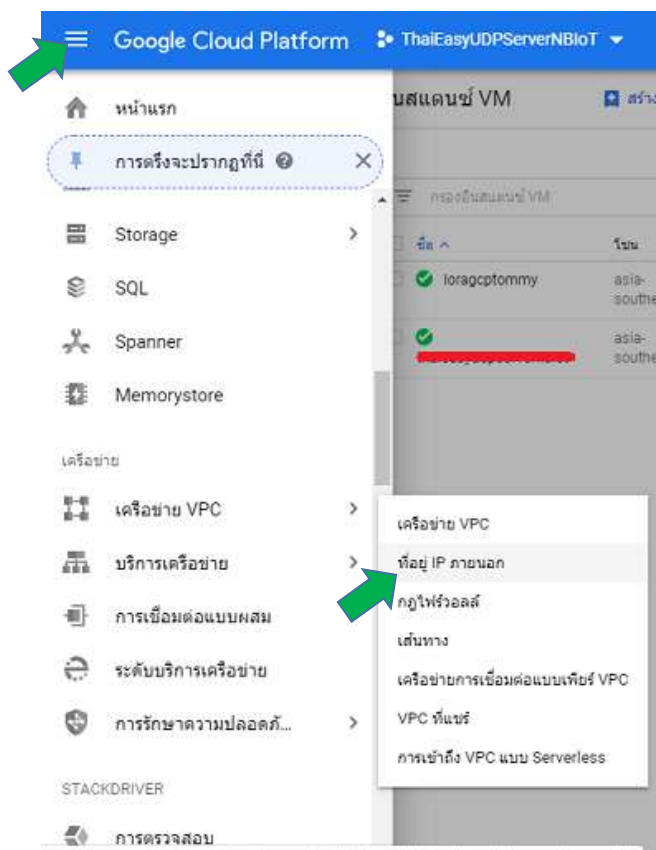
สร้างอินสแตนซ์ นำเข้า VM แสดงแผงข้อมูล

ชื่อ	โซน	สำเนา	ใช้อยู่โดย	IP ภายใน	IP ภายนอก	เชื่อมต่อ
<input type="checkbox"/> loragcptommy	asia-southeast1-b			10.0.0.1 (nic0)	35.240.1.1	SSH
<input type="checkbox"/> [redacted]	asia-southeast1-b			[redacted] (nic0)	[redacted]	SSH

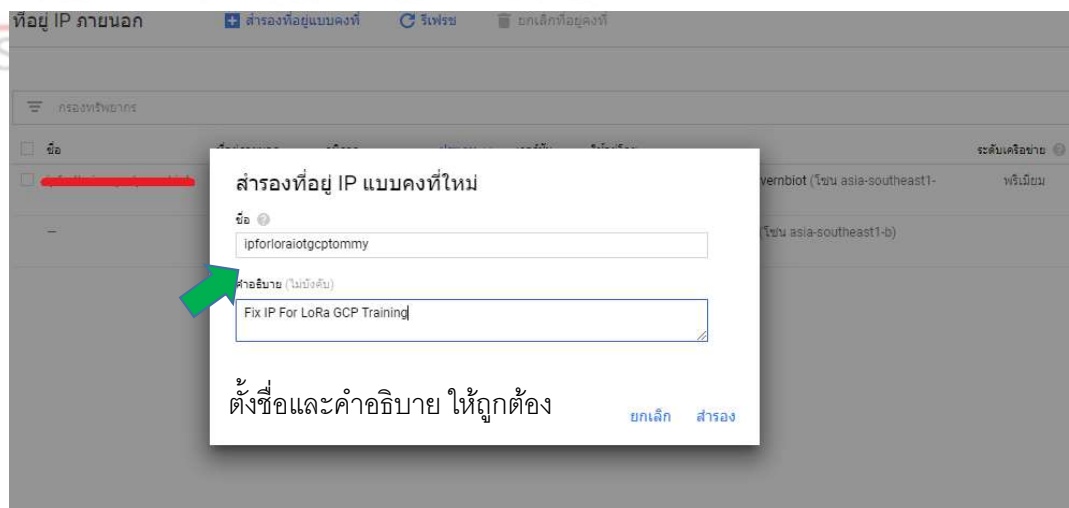
- Virtual สถานะ ออนไลน์
จะได้ IP มา และควรทำการ Fix IP ก่อนเสมอ



การ Fix IP



เปลี่ยนจาก “ชั่วคราว” เป็น “คงที่”



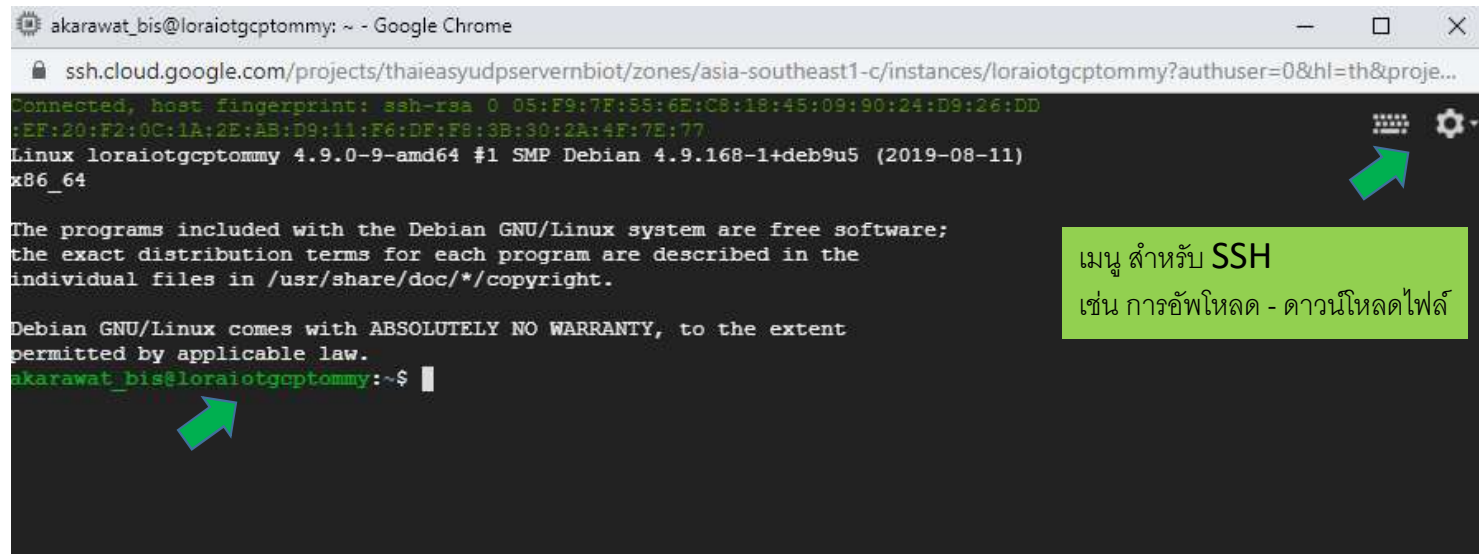
Virtual พร้อมใช้งาน

อินสแตนซ์ VM						
<div>สร้างอินสแตนซ์</div> <div>นำเข้า VM</div> <div></div> <div></div> <div></div> <div></div> <div>แสดงแผงข้อมูล</div>						
<div>กรองอินสแตนซ์ VM</div> <div>คอมส์</div>						
<input type="checkbox"/>	ชื่อ	โซน	สำเนา	ใช้โดย	IP ภายใน	IP ภายนอก
<input type="checkbox"/>	loraiotgcptommy	asia-southeast1-c			10.148.0.4 (nic0)	35.240.177.29
<input type="checkbox"/>		asia-southeast1-b				

คลิกที่ SSH เพื่อเปิด Terminal และเริ่มใช้งาน



Virtual พร้อมใช้งาน



```
akarawat_bis@loraiotgcptommy: ~ - Google Chrome
ssh.cloud.google.com/projects/thaieasyudpserversnbiot/zones/asia-southeast1-c/instances/loraiotgcptommy?authuser=0&hl=th&proje...
Connected, host fingerprint: ssh-rsa 0 05:F9:7F:55:6E:C8:18:45:09:90:24:D9:26:DD
:EF:20:F2:0C:1A:2E:AB:D9:11:F6:DF:F8:3B:30:2A:4F:7E:77
Linux loraiotgcptommy 4.9.0-9-amd64 #1 SMP Debian 4.9.168-1+deb9u5 (2019-08-11)
x86_64

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.
akarawat_bis@loraiotgcptommy:~$
```

เมนู สำหรับ SSH
เช่น การอัปโหลด - ดาวน์โหลดไฟล์



ติดตั้ง Apache2

```
$ sudo apt-get update
```

```
$ sudo apt-get install apache2
```

```
$ sudo systemctl start apache2
```

```

akarawat_bis@loraiotgcptommy:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libicu57
  liblua5.2-0 libperl5.24 libxml2 perl perl-modules-5.24 rename sgml-base ssl-cert xml-core
Suggested packages:
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom perl-doc libterm-readline-gnu-perl
  | libterm-readline-perl-perl make sgml-base-doc openssl-blacklist debhelper
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  libicu57 liblua5.2-0 libperl5.24 libxml2 perl perl-modules-5.24 rename sgml-base ssl-cert xml-core
0 upgraded, 18 newly installed, 0 to remove and 1 not upgraded.
Need to get 17.3 MB of archives.
After this operation, 80.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y

```

```
akarawat_bis@loraiotgcptommy:~$ sudo systemctl start apache2
akarawat_bis@loraiotgcptommy:~$
```

เสร็จแล้ว ทำการ คลิก เปิด หน้าเพจหลักของ **Apache2**

จากหน้า Instance หลัก

อินสแตนซ์ VM

+ สร้างอินสแตนซ์ ⬇️ นำเข้า VM ↻ ▶ ■ 🔌 🗑 แสดงแผงข้อมูล

cache2

ชื่อ	โซน	คำแนะนา	ใช้อยู่โดย	IP ภายใน	IP ภายนอก	เชื่อมต่อ
loraiotgcptommy	asia-southeast1-c			[REDACTED] (nic0)	35.240.177.29	SSH
[REDACTED]	asia-southeast1-b			[REDACTED] (nic0)	[REDACTED]	SSH



หน้าหลักของ Apache2 และตำแหน่งการเก็บไฟล์

เสร็จแล้ว ทำการ คลิก เปิด หน้าเพจหลักของ Apache2
จากหน้า Instance หลัก

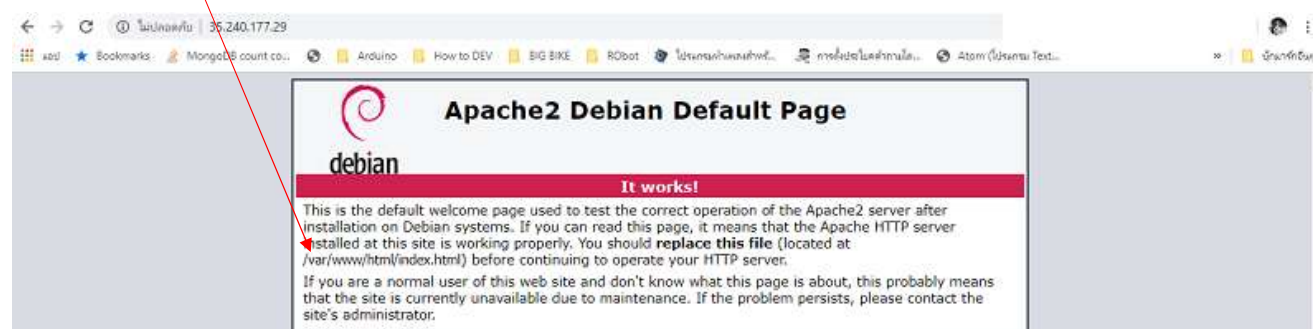
อินสแตนซ์ VM

สร้างอินสแตนซ์ | นำเข้า VM | แสดงแผงข้อมูล

กรองอินสแตนซ์ VM

<input type="checkbox"/>	ชื่อ	โซน	สำเนา	ใช้อยู่โดย	IP ภายใน	IP ภายนอก	เชื่อมต่อ
<input checked="" type="checkbox"/>	loraiotgcptommy	asia-southeast1-c			[REDACTED] (nic0)	35.240.177.29	SSH
<input checked="" type="checkbox"/>	[REDACTED]	asia-southeast1-b			[REDACTED] (nic0)	[REDACTED]	SSH

ตำแหน่งไฟล์จะอยู่ที่ `/var/www/html/index.html`



คำสั่งพื้นฐาน ที่ได้ใช้บ่อย

ls	แสดงชื่อไฟล์ใน directory นั้นๆ
ls -l	แสดงชื่อไฟล์ใน directory นั้นๆ โดยแสดงรายละเอียดอื่นๆด้วย
ls -al	แสดงชื่อไฟล์ใน directory นั้นๆ ทั้งหมด โดยแสดงรายละเอียดอื่นๆด้วย
ls -lt	แสดงชื่อไฟล์ใน directory นั้นๆ โดยเรียงลำดับตามเวลา
cd /foo/bar	Change directory ไปยัง /foo/bar
mv foo bar	ย้ายไฟล์ชื่อ 'foo' ไปยังตำแหน่ง 'bar'
sudo mkdir foo	สร้าง folder ชื่อ 'foo'
sudo rm foo	ลบไฟล์ชื่อ foo
sudo rm -rf foo	ลบโฟลเดอร์ชื่อ foo
sudo unzip foo.zip	Extract file ชื่อ foo.zip (ต้องติดตั้ง zip, unzip ก่อน)



โปรแกรมและโมดูลที่จำเป็นที่ต้องใช้ในระบบ IOP GCP Platform

- Node.js
- npm
- zip unzip
- MongoDB
- Forever
- MongoJS
- Promise
- split-string
- express
- moment



ติดตั้ง Node.js

```
$ curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -  
$ sudo apt-get install -y nodejs  
$ node -v
```

```
akarawat_bis@loraiotgcptommy:~$ node -v  
v8.16.1  
akarawat_bis@loraiotgcptommy:~$
```

ติดตั้ง NPM

```
$ sudo apt-get install npm  
$ npm -v
```

```
akarawat_bis@loraiotgcptommy:~$ npm -v  
6.4.1  
akarawat_bis@loraiotgcptommy:~$
```



ติดตั้ง MongoDB

\$ sudo apt-get install mongodb

```
akarawat_bis@loraiotgcptommy:~$ node -v  
v8.16.1  
akarawat_bis@loraiotgcptommy:~$
```

เสร็จแล้ว เปิดใช้งาน Service ของ MongoDB

\$ sudo systemctl start mongodb

```
akarawat_bis@loraiotgcptommy:~$ sudo systemctl start mongodb  
akarawat_bis@loraiotgcptommy:~$ mongo  
MongoDB shell version: 3.2.11  
connecting to: test  
Welcome to the MongoDB shell.  
For interactive help, type "help".  
For more comprehensive documentation, see  
  http://docs.mongodb.org/  
Questions? Try the support group  
  http://groups.google.com/group/mongodb-user  
>
```

\$ ทดสอบ mongodb ด้วยคำสั่ง db.show

```
> db.show  
test.show  
>
```



พื้นฐานคำสั่ง MongoDB

\$ mongo

- db.show แสดงฐานข้อมูลทั้งหมด
- use myiotdb เลือกใช้ ฐานข้อมูลชื่อ myiotdb
- show collections แสดง Collection ทั้งหมดในฐานข้อมูล

ตัวอย่างการ เขียนข้อมูลใน MongoDB

- db.lora01.insert({"device_id" : "lora01", "t" : 0, "h" : 0, "rssi" : "-82", "ts" : 1562655730 })

ตัวอย่างการ แสดงผล แบบทั้งหมด

- db.lora01.find({})

ตัวอย่างการ แสดงผล แบบแถวล่าสุด และจำกัดการแสดงผล

- db.lora01.find({}).limit(10).sort({"ts":-1})



สร้าง **Folder** โปรเจค และติดตั้งโมดูลที่เกี่ยวข้องติดตั้ง โปรแกรมอื่นๆ

```
$ sudo apt-get install zip  
$ sudo apt-get install unzip  
$ sudo npm install forever -g
```

```
$ sudo mkdir myprotocol  
$ cd myprotocol  
$ sudo npm init
```

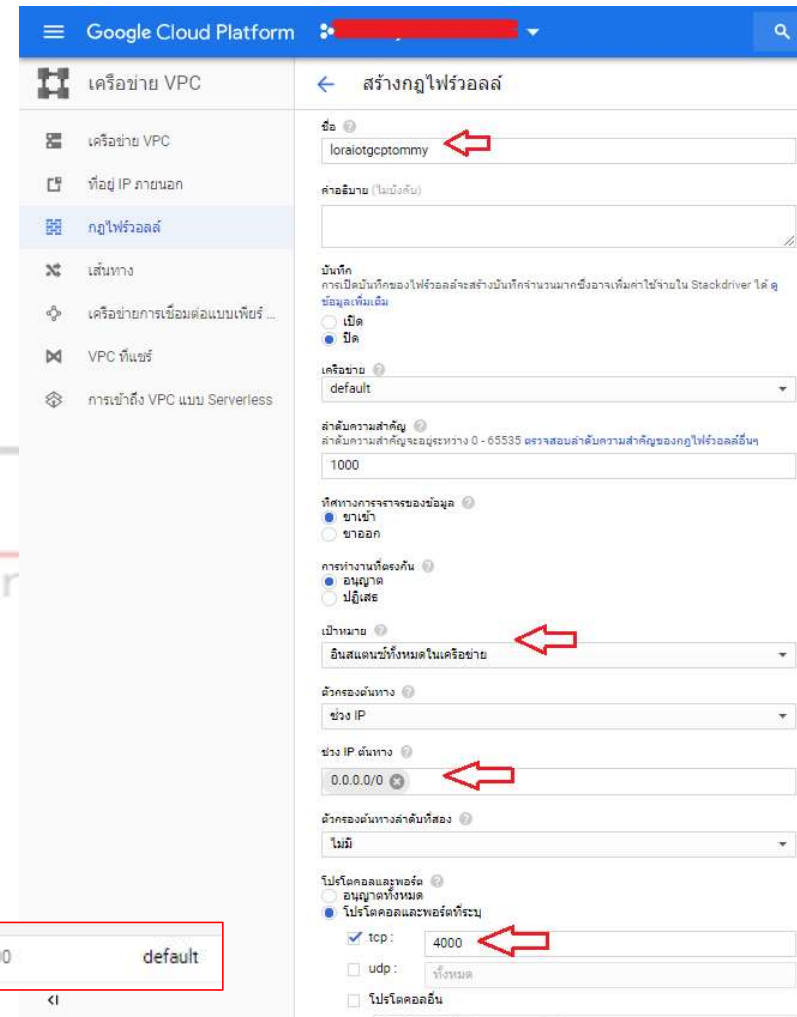
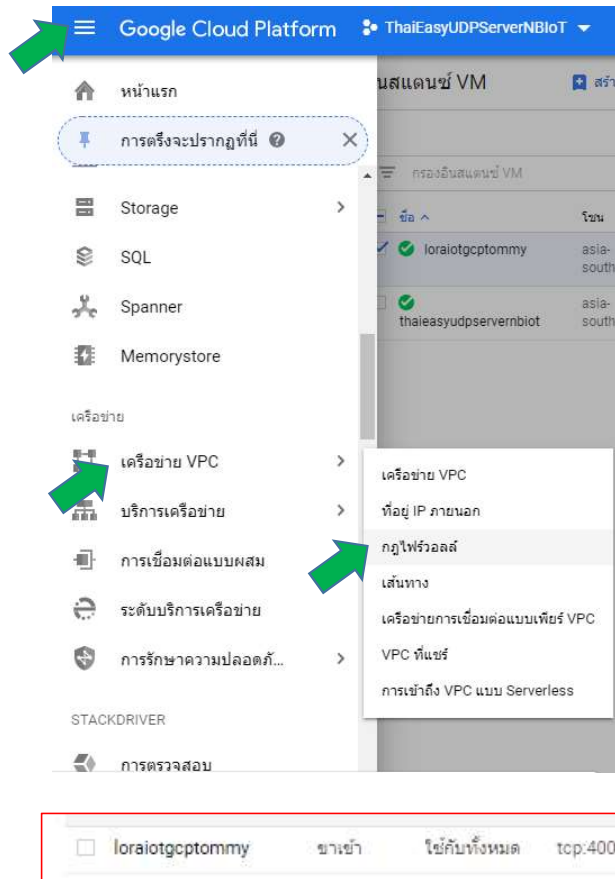
```
$ sudo npm install express --save  
$ sudo npm install promise --save  
$ sudo npm install mongojs --save  
$ sudo npm install split-string --save  
$ sudo npm install moment
```

ขั้นตอน การ **init** ให้ระบุข้อมูล Project ในไฟล์ package.json ให้สมบูรณ์

```
{  
  "name": "loraiotgcbytommy",  
  "version": "1.0.0",  
  "description": "",  
  "main": "app.js",  
  "dependencies": {  
    "express": "^4.17.1",  
    "mongojs": "^2.6.0",  
    "promise": "^8.0.3",  
    "split-string": "^6.1.0"  
  },  
  "devDependencies": {},  
  "scripts": {  
    "test": "echo \\\"Error: no test specified\\\" && exit 1"  
  },  
  "author": "akarawat",  
  "license": "ISC"  
}
```

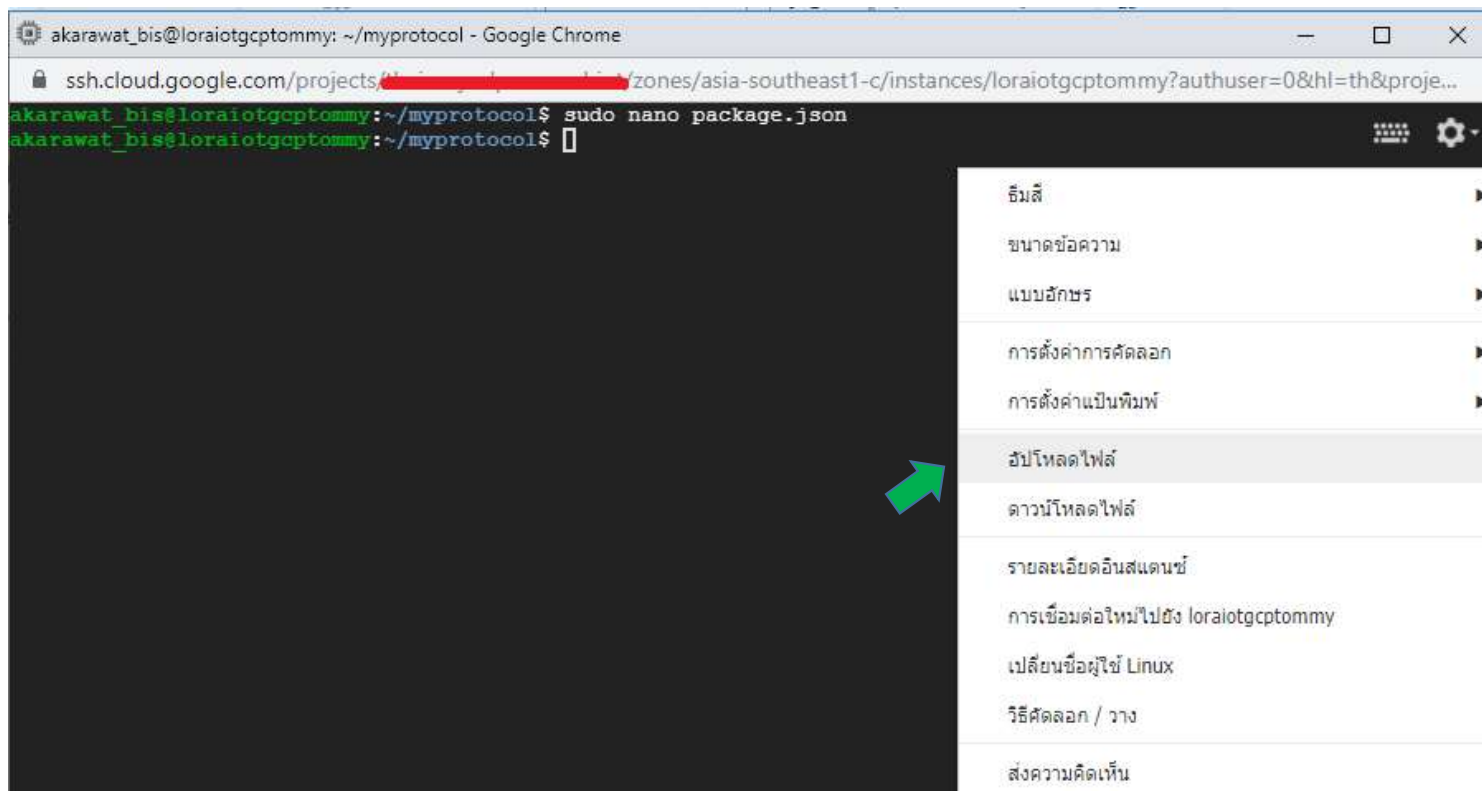


เริ่มต้น Projects แล้ว ต้องกำหนด Firewall ด้วย




Upload Script app.js ขึ้นระบบ ในตำแหน่ง Path ของ Project ที่เรา

สร้าง



ไฟล์ที่ ถูกอัปโหลด จะอยู่ตำแหน่ง **root** เสมอ *จะต้องทำการย้ายมาที่ **Project** ด้วยตัวเอง



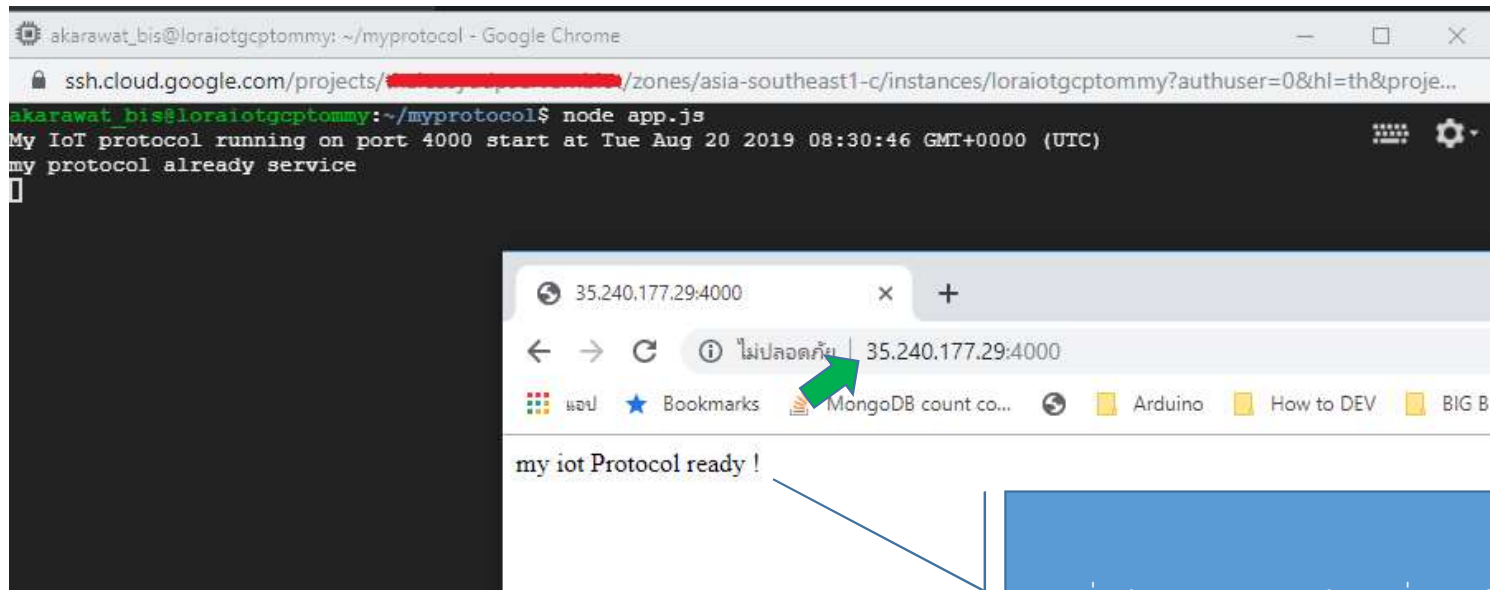
```
akarawat_bis@loraiotgcptommy:~/myprotocol$ cd
akarawat_bis@loraiotgcptommy:~$ ls
app.js  myprotocol
akarawat_bis@loraiotgcptommy:~$ sudo mv app.js myprotocol/
```

- ออกมาที่ตำแหน่ง **root** ด้วย **cd**
- ดูไฟล์ด้วยคำสั่ง **ls**
- ใช้คำสั่งย้ายไฟล์ด้วย **sudo mv app.js myprotocol**
- * จะเห็นว่า ไฟล์ถูกย้ายมา ตามด้านล่าง

```
akarawat_bis@loraiotgcptommy:~$ sudo mv app.js myprotocol/
akarawat_bis@loraiotgcptommy:~$ ls
myprotocol
akarawat_bis@loraiotgcptommy:~$ cd myprotocol/
akarawat_bis@loraiotgcptommy:~/myprotocol$ ls
app.js  node_modules  package.json  package-lock.json
akarawat_bis@loraiotgcptommy:~/myprotocol$
```



สั่ง Run Script และทดสอบ Run Script ที่เราอัปโหลด



The image shows a terminal window and a web browser. The terminal window displays the command `node app.js` and the output: `My IoT protocol running on port 4000 start at Tue Aug 20 2019 08:30:46 GMT+0000 (UTC)` and `my protocol already service`. The web browser shows the URL `35.240.177.29:4000` and the message `my iot Protocol ready !`. A green arrow points from the terminal output to the browser message.

*หมายเหตุ ถ้าต้องการ ให้Script ทำงานตลอดเวลา แม้จะปิด Console
จะต้องสั่งด้วย `sudo forever start app.js`

1. สั่งให้ Script ทำงานด้วยคำสั่ง `node app.js`
2. เปิด Browser ระบุเลขไอพี และ port
3. สังเกตว่า จะเห็น คำสั่ง ที่เรากำหนดใน Script เมื่อมีการเรียกใช้ Script ผ่าน URL ที่ Console จะมีการแสดงผลตาม Script



คำสั่ง forever พื้นฐาน

```
akarawat_bis@loraiotgcptommy:~/myprotocol$ sudo forever list
info:    Forever processes running
data:    uid  command      script  forever pid  id  logfile      uptime
data:    [0]  3z0H  /usr/bin/node app.js  7309    7315  /root/.forever/3z0H.log  0:0:1:45.745000000000005
akarawat_bis@loraiotgcptommy:~/myprotocol$
```

เรียกดู forever service ที่กำลังทำงาน

\$ sudo forever list

สั่ง start forever

\$ sudo forever start (ชื่อscript)

สั่ง stop forever

\$ sudo forever stop (ชื่อscript)

* ข้อควรระวัง ถ้ามีการแก้ไข script ต้องสั่ง stop script ก่อนเสมอ

แก้ไขเสร็จ จึงสั่ง start ใหม่

** การสั่งงาน forever ถ้าใช้ sudo กับไม่ใช้มีผลต่างกัน เนื่องจาก ระบบ จะมองว่าเป็นคนละ Transaction ดังนั้นควรระวังข้อนี้ด้วย

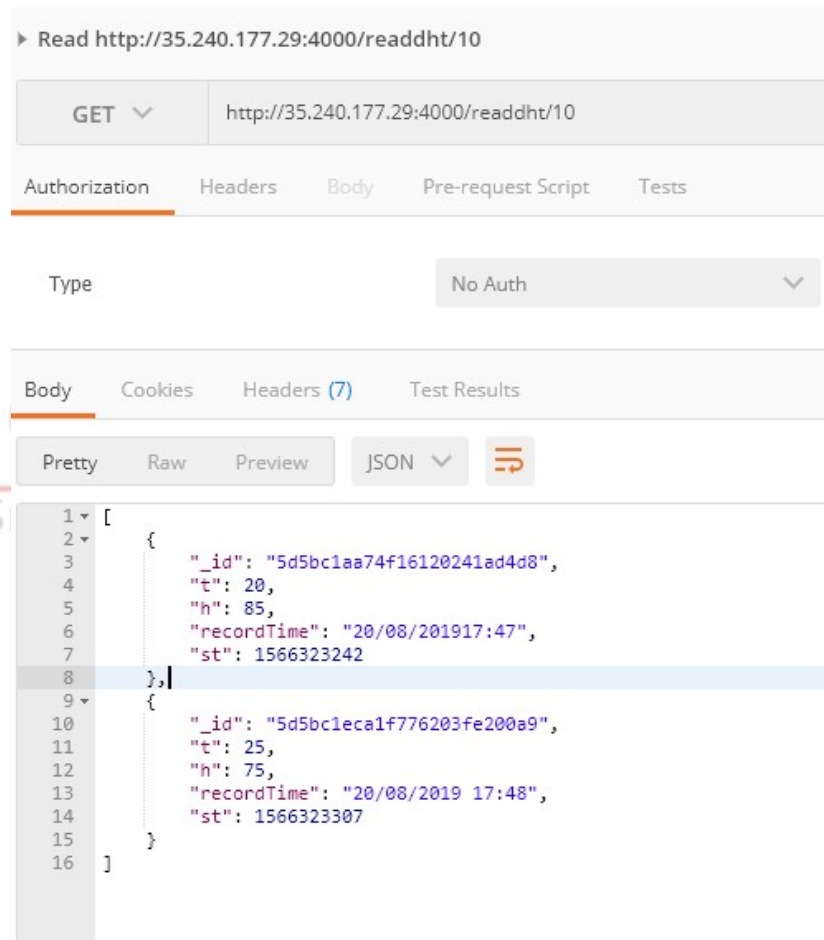


ทดสอบการติดต่อกับ Api ด้วย Postman

1. ทดสอบ เรียกข้อมูล ด้วย วิธีการ GET

`http://35.240.177.29:4000/readdht/10`

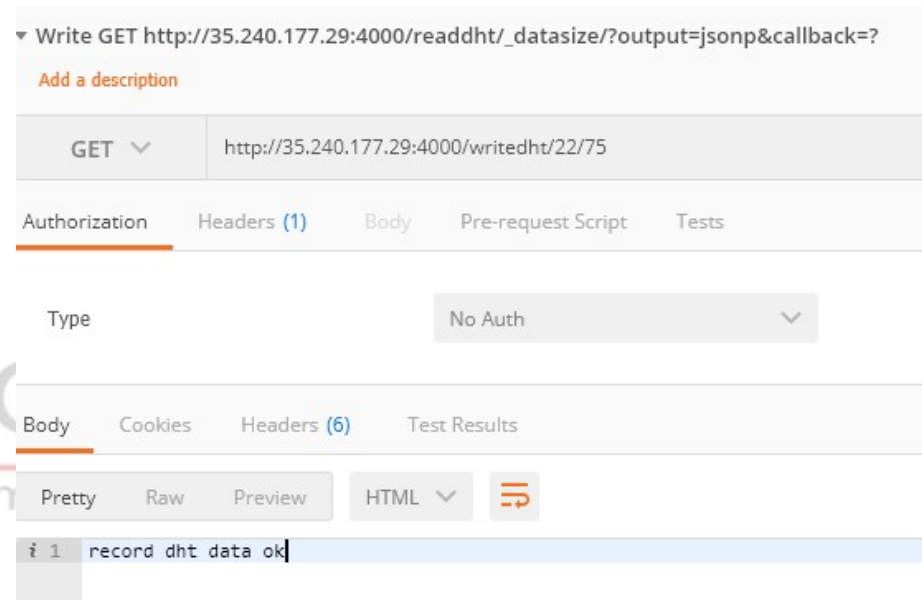
1. ระบุ IP และ Port ตามที่เราได้มา ให้ถูกต้อง
2. ระบุ function ใน script ให้ถูกต้อง



ทดสอบการติดต่อกับ Api ด้วย Postman

2. ทดสอบ เขียนข้อมูล ด้วย วิธีการ ส่งข้อมูล GET ไปให้ Api
`http://35.240.177.29:4000/writedht/22/75`

1. ระบุ IP และ Port ตามที่เราได้มา ให้ถูกต้อง
2. ระบุ function ใน script ให้ถูกต้อง
3. สังเกตว่า จะมี Parameters ตาม function ที่ได้เขียนไว้ใน Script



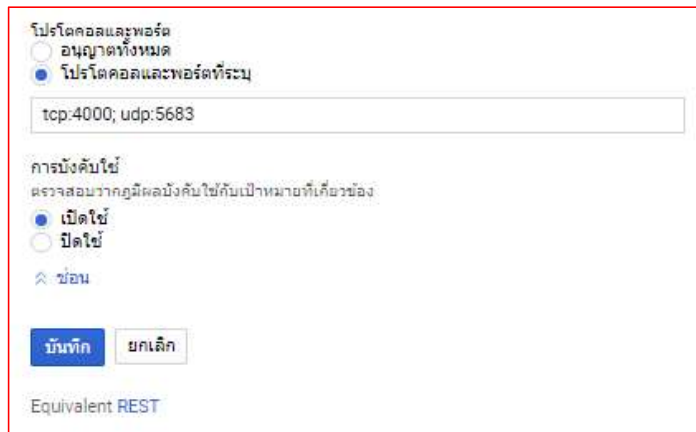
3. เมื่อ เรียกดูใน mongo จะเห็นว่า มีข้อมูลถูกเพิ่มเข้ามาตามที่เราระบุ

```
> db.lora01.find({}).limit(10).sort({"st":-1})
{ "_id" : ObjectId("5d5bc3ef3e39f720c4bf4e08"), "t" : 22, "h" : 75, "recordTime" : "20/08/2019 17:57", "st" : 1566323823 }
{ "_id" : ObjectId("5d5bc1ecaf776203fe200a9"), "t" : 25, "h" : 75, "recordTime" : "20/08/2019 17:48", "st" : 1566323307 }
{ "_id" : ObjectId("5d5bc1aa74f16120241ad4d8"), "t" : 20, "h" : 85, "recordTime" : "20/08/2019 17:47", "st" : 1566323242 }
>
```



*** สร้าง การเชื่อมต่อกับ Api ด้วย UDP Service

1. กลับไปที่เมนู “กฎ Firewall” เพื่อแก้ไข ให้เพิ่ม UDP Port เป็น 5683 ตามรูปด้านล่าง แล้วบันทึก



โปรโตคอลและพอร์ต

☐ อนุญาตทั้งหมด

☒ โปรโตคอลและพอร์ตที่ระบุ

tcp:4000; udp:5683

การบังคับใช้

ตรวจสอบว่ากฎมีผลบังคับใช้กับเป้าหมายที่เกี่ยวข้อง

☒ เปิดใช้

☐ ปิดใช้

[ช่วยเหลือ](#)

[บันทึก](#) [ยกเลิก](#)

Equivalent REST

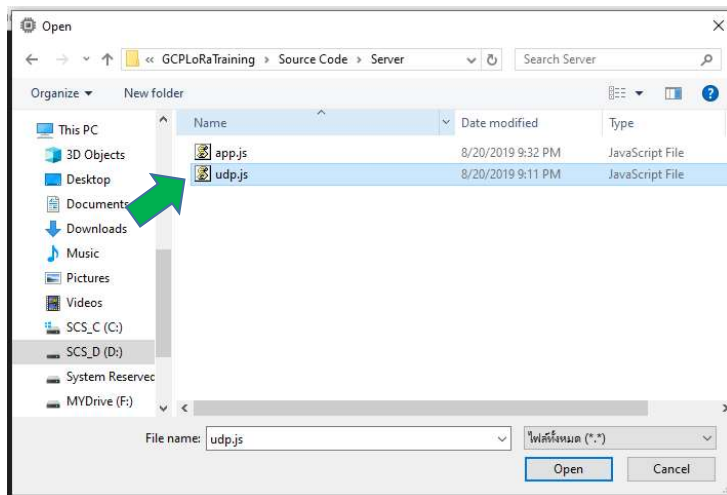
2. จะเห็นว่า มีการเพิ่ม UDP Port เป็น 5683 ตามรูปด้านล่าง

loraiotgcptommy	ขาเข้า	ใช้กับทั้งหมด	ช่วง IP: 0.0.0.0/0	tcp:4000 udp:5683	อนุญาต	1000
-----------------	--------	---------------	--------------------	----------------------	--------	------



*** สร้าง การเชื่อมต่อกับ Api ด้วย UDP Service

1. Upload file “udp.js” ขึ้นระบบ และย้ายไปไว้ที่ Folder myprotocol



```
akarawat_bis@loraiotgcptommy:~$ sudo mv udp.js myprotocol
```

2. สั่ง run script ด้วยคำสั่ง

```
$ node udp.js
```

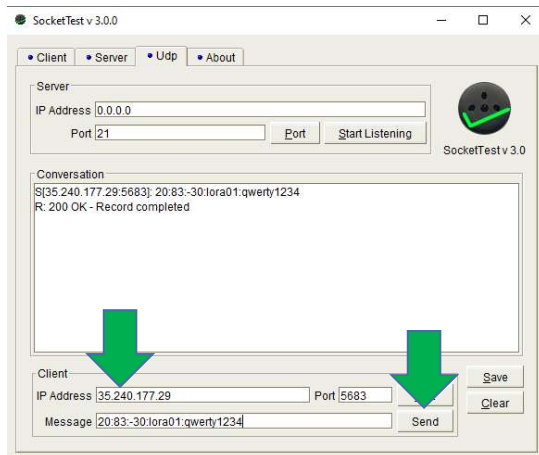
3. จะพบ คำสั่ง ที่เรากำหนดไว้ใน script

```
akarawat_bis@loraiotgcptommy:~/myprotocol$ node udp.js  
My udp protocol start listen on port 5683
```



*** สร้าง การเชื่อมต่อกับ Api ด้วย UDP Service

1. ทดสอบ การนำข้อมูลขึ้น ระบบ ด้วยโปรแกรม เช่น Socket Tester.



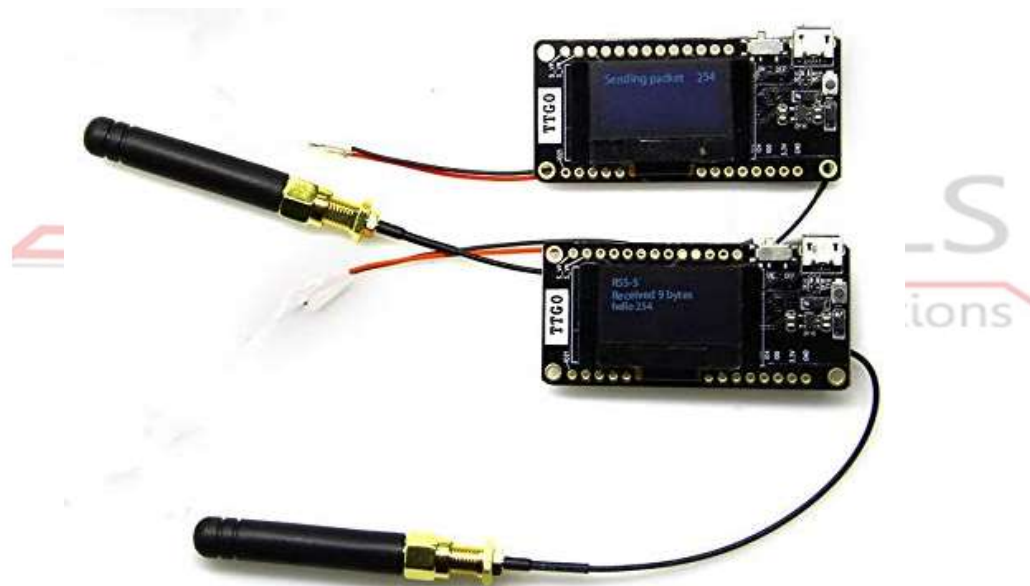
* ตัวอย่าง การติดต่อ Server และตัวอย่างข้อมูลที่ทดสอบส่งเข้า UDP
Server IP : 35.240.177.29:5683
Data : 20:83:-30:lora01:qwerty1234

เมื่อส่งข้อมูลแล้ว ทำการเรียกดูข้อมูลที่ถูกเพิ่ม ใน mongo ตามด้านล่าง

```
akarawat_bis@loraiotgcp.commy:~$ mongo
MongoDB shell version: 3.2.11
connecting to: test
> use myiotdb
switched to db myiotdb
> db.lora01.find({}).limit(10).sort({"ts":-1})
{ "_id" : ObjectId("5d5ca39a4240bc321bf05158"), "t" : 20, "h" : 83, "rssi" : -30, "recordTime" : "21/08/2019 09:51", "ts" : 1566381082 }
{ "_id" : ObjectId("5d5c0874afb7342f57c92d37"), "t" : 25, "h" : 85, "rssi" : -35, "recordTime" : "20/08/2019 22:49", "ts" : 1566341364 }
{ "_id" : ObjectId("5d5c079dafb7342f57c92d36"), "t" : 15, "h" : 85, "rssi" : -35, "recordTime" : "20/08/2019 22:45", "ts" : 1566341149 }
{ "_id" : ObjectId("5d5c0794afb7342f57c92d35"), "t" : 25, "h" : 85, "rssi" : -35, "recordTime" : "20/08/2019 22:45", "ts" : 1566341140 }
{ "_id" : ObjectId("5d5c0782afb7342f57c92d34"), "t" : 21, "h" : 91, "rssi" : -35, "recordTime" : "20/08/2019 22:45", "ts" : 1566341122 }
{ "_id" : ObjectId("5d5c06cbafb7342f57c92d33"), "t" : 35, "h" : 95, "rssi" : -45, "recordTime" : "20/08/2019 22:42", "ts" : 1566340939 }
{ "_id" : ObjectId("5d5c019bafb7342f57c92d31"), "t" : 25, "h" : 85, "rssi" : -35, "recordTime" : "20/08/2019 22:20", "ts" : 1566339611 }
{ "_id" : ObjectId("5d5c018f3b6ad62f41c18db1"), "t" : 23, "h" : 73, "rssi" : 0, "recordTime" : "20/08/2019 22:19", "ts" : 1566339599 }
{ "_id" : ObjectId("5d5c011d787eaf2cfb78c2e0"), "t" : 23, "h" : 73, "rssi" : 0, "recordTime" : "20/08/2019 22:18", "ts" : 1566339485 }
```



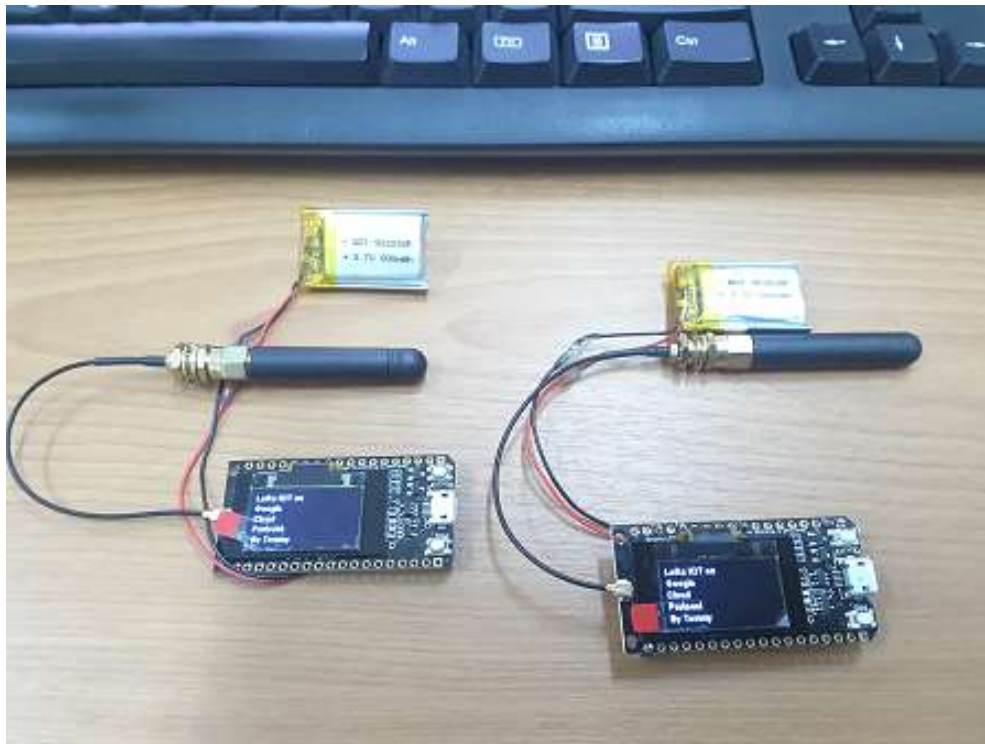
การส่งข้อมูล ขึ้นระบบ ด้วย LoRa Esp32 OLED



การส่งข้อมูล ขึ้นระบบ ด้วย LoRa Esp32 OLED

Link Reference

<https://github.com/osresearch/esp32-ttgo>



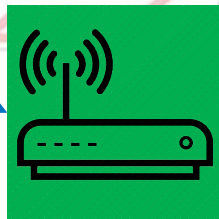
TOOLS
Connect Solutions



การส่งข้อมูล ขึ้นระบบ ด้วย LoRa Esp32 OLED

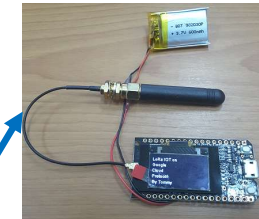
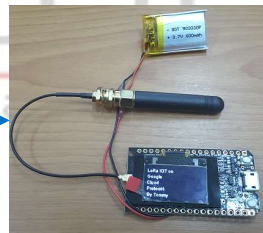
LoRa to GCP Infrastructure

Google Cloud Platform



Access point Or Cells Site

LoRa Master



LoRa Node 1



LoRa Node 3



LoRa Node 2



การส่งข้อมูล ขึ้นระบบ ด้วย LoRa Esp32 OLED

LoRa Master



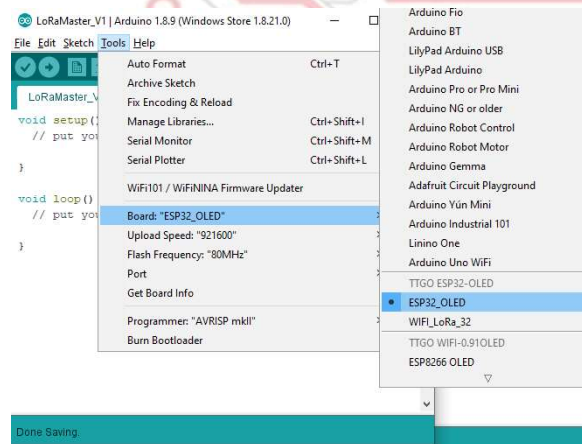
1. File โปรแกรมของ LoRa ตัวที่จะเป็น Master LoRaMaster_V1.ino

 LoRaMaster_V1.ino

เสียบบอร์ด และ Download ตัวอย่างโปรแกรมจาก

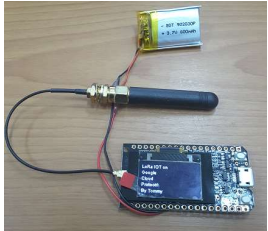
https://github.com/akarawat/GCPLoRaTraining/blob/master/Source%20Code/IOT%20Device/LoRaMaster_V1/LoRaMaster_V1.ino

ทำการเลือกบอร์ด และ Upload โปรแกรมที่ได้ดาวน์โหลดมาลงไป



การส่งข้อมูล ขึ้นระบบ ด้วย LoRa Esp32 OLED

LoRa Node



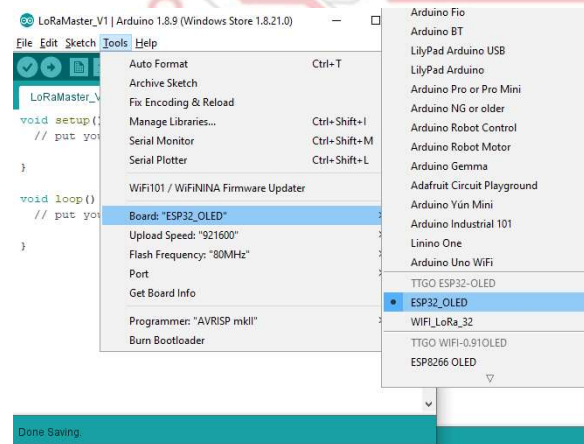
2. File โปรแกรมของ LoRa ตัวที่จะเป็น Node LoRaNode_V1.ino

LoRaNode_V1.ino

เสียบบอร์ด และ Download ตัวอย่างโปรแกรมจาก

https://github.com/akarawat/GCPLoRaTraining/blob/master/Source%20Code/IOT%20Device/LoRaNode_V1/LoRaNode_V1.ino

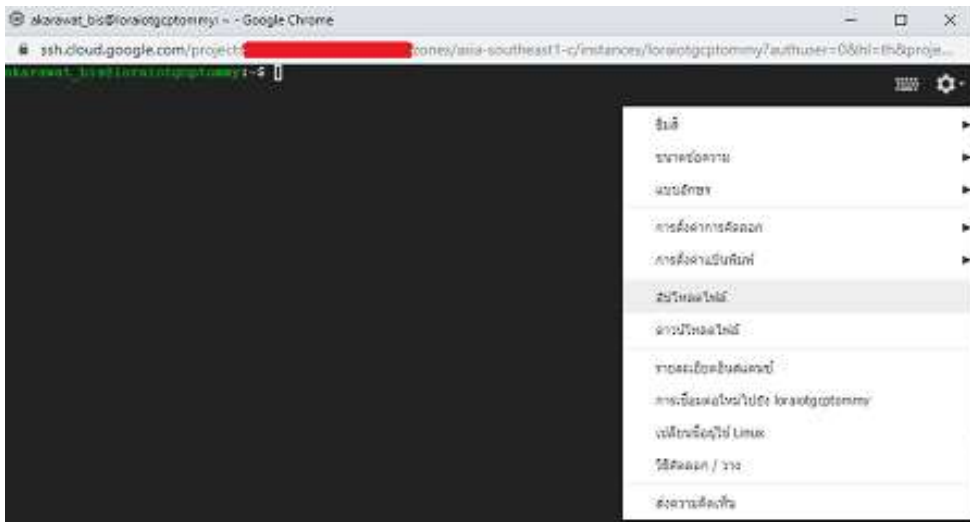
ทำการเลือกบอร์ด และ Upload โปรแกรมที่ได้ดาวน์โหลดมาลงไป



นำข้อมูลที่ได้ไปแสดงบน Dashboard

1. การแสดงผล **Dashboard** จะต้องทำระบบให้เสร็จเรียบร้อยแล้วให้ **zip file** เพื่ออัปโหลดขึ้นไปบน **Google Cloud Platform**

Download : <https://github.com/akarawat/GCPLoRaTraining/blob/master/Source%20Code/Dashboard.zip>



2. เมื่อ **upload file** แล้ว จะต้องทำการย้ายไฟล์ไปตำแหน่งหลักของ **Apache webserver** เนื่องจากไฟล์ เมื่ออัปโหลดแล้วมันจะไปอยู่ใน **root** เราต้องทำการย้ายเอง

โดยตำแหน่งไฟล์ที่จะย้ายไป คือ **/var/www/html**

```
akarawat_bis@loraiotgcptommy:~$ ls
Dashboard.zip  myprotocol  tmp
akarawat_bis@loraiotgcptommy:~$ sudo mv Dashboard.zip /var/www/html/
```



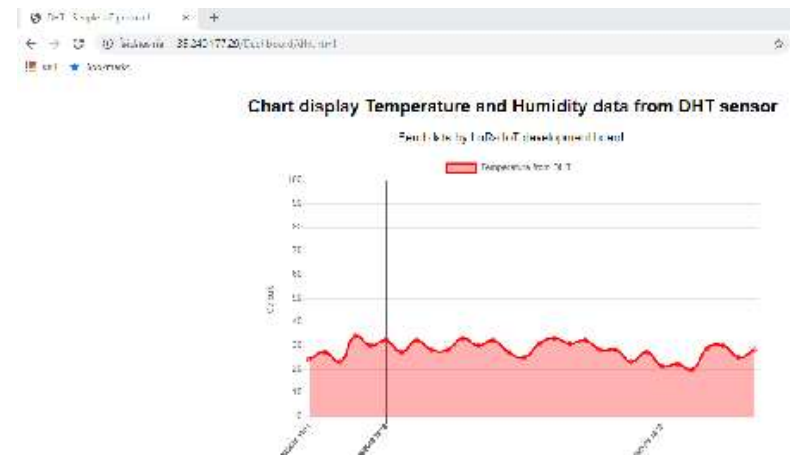
นำข้อมูลที่ได้ไปแสดงบน Dashboard

3. เสร็จแล้วทำการแตกไฟล์ด้วยคำสั่ง `$sudo unzip filexxx.zip`

```
akarawat_bis@loraiotgcptommy:/var/www/html$ ls
Dashboard.zip  index.html
akarawat_bis@loraiotgcptommy:/var/www/html$ sudo unzip Dashboard.zip
Archive:  Dashboard.zip
  creating: Dashboard/
  inflating: Dashboard/dht.html
  inflating: Dashboard/readme.md
  creating: Dashboard/src/
  creating: Dashboard/src/css/
  inflating: Dashboard/src/css/style.css
  creating: Dashboard/src/js/
  inflating: Dashboard/src/js/Chart.min.js
  inflating: Dashboard/src/js/moment.js
  inflating: Dashboard/src/js/readme.md
  inflating: Dashboard/src/readme.md
akarawat_bis@loraiotgcptommy:/var/www/html$
```

4. ทดสอบเข้า Dashboard ผ่าน web browser โดยระบุ IP และ path ของ Server ให้ถูกต้อง

เช่น <http://35.240.177.29/Dashboard/dht.html>



อ้างอิงข้อมูล

<https://cloud.google.com/gcp/>

<http://code.isaranu.com/>

