

Week12A - Visualize Sensor data using Grafana and InfluxDB

กลุ่มคำนวณ 5 รหัส B6210533 ชื่อ-สกุล รินรดา วัฒนชนสรณ์

1/4 การทำรายงาน

- เป็นการเก็บคะแนน 10 คะแนนตัดเกรด ให้ทำสุดความสามารถในเวลาที่มี
- ให้รับส่งโดยถือว่าโครงสร้างก่อนถือว่าเป็นต้นฉบับส่งที่หลังถือว่าลอกเพื่อนมา
- ให้แก้ไขหัวกระดาษ, เพิ่มเติมเอกสารให้สมบูรณ์ รูปภาพการทำงานทั้งวงจร และโค้ดโปรแกรม
- เพิ่มเติมเนื้อหาด้านท้าย ด้วยการคัดลอกและจัดเรียงใหม่
- รูปที่เป็นการทดสอบ ESP32 ควรเป็นรูปของตัวเองที่ทดสอบ
- รูปถ่ายต้องเป็นของตัวเองและมีกระดาษรองอุปกรณ์ที่เขียน รหัส ชื่อ-สกุล ของตัวเอง
- บันทึกไฟล์ในรูป pdf, กำหนดชื่อไฟล์ Wk12A-B3701234-Wichai-Srisuruk.pdf (แก้ไขตามรหัส ชื่อตัวเอง)
- ส่งงานก่อน 06:00น วันพุธสุดที่ 24 มิย 64 ที่ Link >> <https://shorturl.at/efuM2>

2/4. Read More

- <https://grafana.com/blog/2021/03/08/how-i-built-a-monitoring-system-for-my-avocado-plant-with-arduino-and-grafana-cloud/>
- <https://www.metricfire.com/blog/iot-dashboards-with-grafana-and-prometheus/>
- <https://gabrieltanner.org/blog/grafana-sensor-visualization>

3/4. ให้จัดเรียบเรียงข้อมูล

- <https://www.techtalkthai.com/arm-pelion-full-stack-iot-platform/>

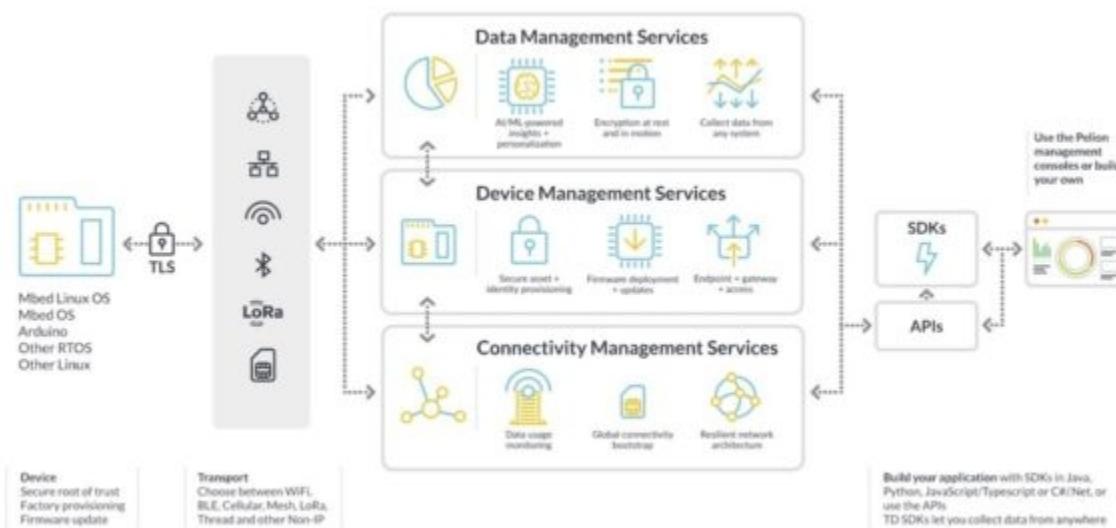
Arm-Pelion Full Stack IoT Platform

รู้จัก ARM Pelion แพลตฟอร์ม IoT จาก ARM จัดการทุกอย่างครบจบในที่เดียว

เมื่อวันที่ 30 กรกฎาคมที่ผ่านมา ทาง ARM ได้จัด IoT Workshop ขึ้นเพื่อนำเสนอและมุ่งด้านของการนำเทคโนโลยี IoT ไปใช้ในธุรกิจ ที่มีงาน TechTalkThai ได้เข้าไปร่วมงาน และทำความรู้จักกับ ARM Pelion IoT Platform ซึ่งอย่างจะมาเล่าให้ผู้อ่านฟังกันคร่าวๆว่าเจ้าแพลตฟอร์ม IoT นี้มีความสามารถอย่างไร และเหมาะสมกับธุรกิจแบบไหนบ้าง

IoT นั้นเป็นหนึ่งในเทคโนโลยีที่หลายองค์กรยกให้เป็นยุทธศาสตร์ในปี 2019 จากความสามารถในการรวบรวมข้อมูล ซึ่งนับว่าเป็นวัตถุดินในการทำธุรกิจที่ขาดไม่ได้เลยในยุคปัจจุบัน จนถึงตอนนี้ หลายองค์กรอาจเริ่มต้นกับ IoT กันบ้างแล้ว แต่ความท้าทายใหม่ที่ธุรกิจมักจะเผชิญกับคือการจัดการและสเกลระบบ IoT ให้ใช้งานเก็บข้อมูลได้จริงเต็มประสิทธิภาพ ปลอดภัย และมีระบบจัดการที่ดี ดังนั้นจึงมีการพัฒนาแพลตฟอร์ม IoT ขึ้นเพื่อช่วยธุรกิจในการแก้ปัญหานี้

ARM Pelion IoT Platform ก็เป็นหนึ่งในแพลตฟอร์ม IoT ที่จะเข้ามาช่วยลดความซับซ้อนของการนำ IoT ไปใช้งานในธุรกิจ แพลตฟอร์ม Pelion นี้แบ่งออกเป็น 3 ส่วน ตามการใช้งาน คือ Connectivity Management Services, Device Management Services, และ Data Management Services โดยทั้ง 3 ส่วนจะทำงานร่วมกันภายใต้ระบบบริการความปลอดภัย ซึ่งเป็นหลักสำคัญที่สุดในการพัฒนาผลิตภัณฑ์ทุกๆตัวของ Arm



ภาพรวมของแพลตฟอร์ม Pelion ที่แบ่งการทำงานออกเป็น 3 ส่วน โดยธุรกิจสามารถเลือกใช้เพียงส่วนใดส่วนหนึ่งหรือทั้ง 3 ส่วนร่วมกันได้ (ภาพ: ARM)

Pelion จะช่วยให้ธุรกิจจัดการกับเครือข่าย อุปกรณ์ในเครือข่าย และข้อมูลที่เก็บมาได้่ายั่งชื่น โดยสามารถทำงานร่วมกับอุปกรณ์ระบบเครือข่าย คลาวด์ และข้อมูลได้หลากหลายรูปแบบ อีกทั้งยังมีความปลอดภัย และสามารถช่วยในการนำข้อมูลไปวิเคราะห์ และแสดงผลเบื้องต้นได้ด้วย

รู้จักแพลตฟอร์มนี้ไปคร่าวๆแล้ว ลองมาเจาะลึกกันว่าส่วนประกอบทั้ง 3 ส่วน คืออะไร Connectivity Management, Device Management, และ Data Management นั้นประกอบไปด้วยอย่างไร และมีจุดเด่นอย่างไรบ้าง

Connectivity Management

การเชื่อมต่อในเครือข่าย IoT นั้นมีอยู่หลายรูปแบบ และมีรายละเอียดปลีกย่อยที่ธุรกิจจะต้องจัดการอยู่พสมควร Pelion จะช่วยให้องค์กรสามารถจัดการการเชื่อมต่อได้อย่างมีประสิทธิภาพ ปลอดภัย และพร้อมต่อการสเกลเครือข่ายขึ้นไปอีกระดับโลก โดย Connectivity Management ของ Pelion มีความสามารถที่น่าสนใจ ดังนี้

- **Global Cellular**

Pelion จะช่วยให้อุปกรณ์ IoT สามารถเชื่อมต่อผ่านเครือข่ายได้ไม่ว่าอุปกรณ์นั้นจะอยู่ที่ใดในโลก ผ่านเวนเดอร์เพียงเจ้าเดียว โดยกลไกของ Pelion จะช่วยเชื่อมต่อสัญญาณจากซิมของอุปกรณ์ไปยังเครือข่ายท้องถิ่นที่ Pelion ได้ทำข้อตกลงไว้ ลดภาระความปวดหัวในการติดต่อกับผู้ให้บริการเครือข่ายในแต่ละประเทศ

- **Protocol เชื่อมต่อทั้ง IP และ Non-IP**

นอกจากรสั่งข้อมูลผ่าน IP Network แล้ว Pelion ยังรองรับ Non-IP Network เช่น NB-IoT ด้วย โปรโตคอลที่ Pelion รองรับนั้นมีได้แก่ MQTT(s), HTTPS, และ Sockets

- **ใช้ได้ทั้ง eSIM และซิมแบบปกติ**

ธุรกิจสามารถสั่งผลิตอุปกรณ์ที่มีระบบ eSIM ผ่าน ARM ได้ตามต้องการ โดย eSIM ที่ติดมากับอุปกรณ์นั้นจะรองรับการเชื่อมต่อกับเครือข่ายกว่า 600 เครือข่ายทั่วโลก และหากต้องการเปลี่ยนเครือข่าย ก็สามารถตั้งค่าใหม่ได้ภายในตัว Pelion ที่ให้บริการซิมการ์ดในทุกขนาด อีกทั้งยังมีแผนที่จะพัฒนาไปจนถึง iSIM ที่มีขนาดเล็กกว่า eSIM มากด้วย

- **Network Infrastructure**

Pelion ได้พัฒนาโครงสร้างพื้นฐานของเครือข่ายให้สามารถทำงานร่วมกับผู้ให้บริการเครือข่ายทั่วโลกได้อย่างมีประสิทธิภาพสูง โดยมีทั้งความเสถียร ยืดหยุ่น และเป็นไปตามกฎข้อบังคับด้านข้อมูลของแต่ละประเทศ ในกรณีที่ Pelion ผู้ใช้งานสามารถเลือกได้ว่าจะส่งข้อมูลจากอุปกรณ์ไปยังแอปพลิเคชันผ่านเทคโนโลยีใด เช่น IPSEC, Open VPN, ผู้ให้บริการ Cloud, หรือทางเชื่อมที่ธุรกิจเข้ามาใช้โดยเฉพาะ (Leased Line)

- **Device Management**

Device Management ของ Pelion นั้นจะช่วยให้องค์กรสามารถจัดการกับอุปกรณ์และการเชื่อมต่อกับอุปกรณ์ผ่านซอฟต์แวร์ได้โดยสะดวก ไม่ว่าจะเป็นการเขียนซอฟต์แวร์แบบ Embedded หรือว่าการเขียนแอปพลิเคชันด้านบนอย่าง Web App ก็ตาม



(ภาษาไทย)

พ: ARM)

โดยภายในโซลูชัน Device Management ก็จะมีโมดูลในการจัดการเรื่องต่างๆให้อย่างครบถ้วน ตั้งแต่เรื่องการอัปเดท Firmware ซึ่งไม่ง่ายเลยหากมีอุปกรณ์ที่หลากหลายและมีจำนวนที่มากในเครือข่าย, Access Management ซึ่งจะช่วยจำกัดการเข้าถึง อุปกรณ์และการควบคุมแต่ละส่วน, Connector ซึ่งจัดการการเชื่อมต่อกับอุปกรณ์หลากหลายประเภท การออก Certificate เข้า ใช้ระบบ การขึ้นทะเบียนอุปกรณ์แต่ละตัว และการเก็บสถิติ, Device Directory ซึ่งช่วยในการแบ่งกลุ่ม ค้นหา เรียกดู และเช็ค สถานะของอุปกรณ์แต่ละตัว, เป็นต้นในการรักษาความปลอดภัยในการส่งต่อรหัสผ่านเครือข่าย WiFi

โดยทั้งหมดนี้จะเห็นได้ว่าเป็นการจัดการ Lifecycle ของอุปกรณ์ทั้งหมด ตั้งแต่การ Onboard เข้าระบบ ไปจนถึงการใช้งานและ บำรุงรักษา

Device Management นั้นสามารถพูดคุยกับอุปกรณ์ IoT ผ่านโปรโตคอลหลากหลาย โดยเฉพาะ LwM2M ซึ่งช่วยให้กับแพทฟอร์มฯ เชื่อมต่อกับอุปกรณ์ได้ผ่านไมโครเดลที่มีลักษณะคล้ายๆ REST Model และ CoAP ซึ่งจะช่วยประหยัดแบตเตอรี่ของอุปกรณ์ได้มากกว่า HTTP ราว 8-10 เท่า และในการเชื่อมต่อกับแอปพลิเคชันซึ่งเป็นปลายทางอีกด้านหนึ่ง Pelion ก็ได้เตรียม REST API และ SDK ในภาษา Java, Python, JavaScript และ .NET ไว้ให้พัฒนาแอปพลิเคชันกันได้โดยง่าย

Device Management ของ Pelion นี้รองรับการทำงานร่วมกับชาร์ดแวร์ที่หลากหลาย ไม่ว่าจะเป็นอุปกรณ์แบบ Bare metal (มีระบบเชื่อมต่อที่เรียกว่า Edge รองรับ) และการทำงานร่วมกับระบบปฏิบัติการทั้ง Mbed OS และ Linux

Data Management

เป้าประสงค์หลักของการจัดตั้งระบบ IoT นั้นคือการสร้างระบบจัดเก็บข้อมูลที่จะช่วยให้องค์กรสามารถเรียกข้อมูลเหล่านั้นขึ้นมาวิเคราะห์เป็นความรู้ที่นำไปใช้ประโยชน์ต่อธุรกิจได้ แน่นอนว่า Pelion ย่อมไม่เลี่มความสำคัญของส่วนนี้ จึงได้พัฒนาระบบจัดการข้อมูลครบวงจรที่จะช่วยตั้งแต่การจัดเก็บ นำข้อมูลมาใช้ตัดสินใจแบบ Real-time และรักษาความปลอดภัยและความเป็นส่วนตัวของข้อมูล โดยมีกลไกรองรับการสเกลเต็มที่ ทำให้องค์กรไม่ต้องกังวลว่าระบบจะทำงานได้แบ่งหากมีข้อมูลหรืออุปกรณ์ในเครือข่าย IoT เพิ่มมากขึ้นเมื่อเวลาผ่านไป

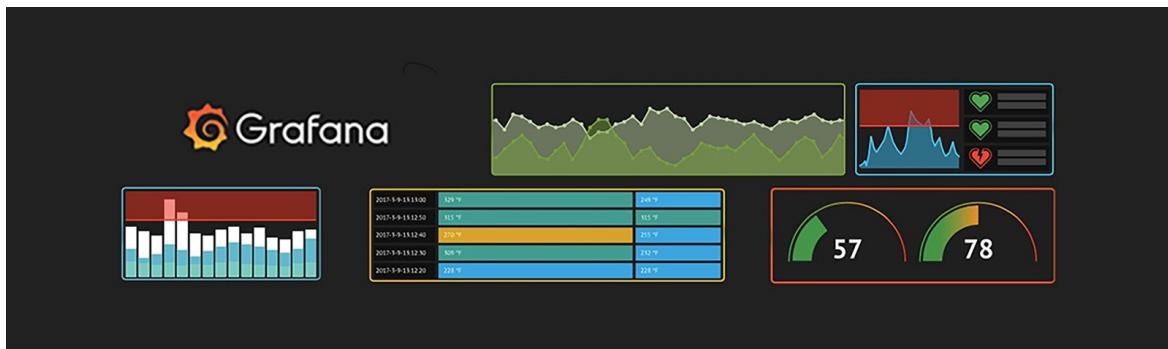
โซลูชันหลักของส่วนนี้ คือ [ARM Treasure Data](#) ซึ่งเป็นซอฟต์แวร์จัดการและวิเคราะห์ข้อมูลที่เข้ามาร่วมกับ Pelion ได้จำกัดในตัวเดียว โดยมีเครื่องมือต่างๆพร้อมให้เลือกใช้งาน เช่น ระบบ Predictive Analytics การสร้าง Customer View 360 ของค่าจากข้อมูลการใช้งาน การวิเคราะห์ข้อมูลเพื่อ Cross-sell และ Upsell และการสร้างระบบ Recommendation เป็นต้น ซึ่งโซลูชันนี้หลายๆองค์กรก็ได้นำไปใช้งานเพิ่มประสิทธิภาพให้กับการทำงานในอุตสาหกรรมมากมาย เช่น อุตสาหกรรมค้าปลีก อุตสาหกรรมพลังงาน อุตสาหกรรมการผลิต และอุตสาหกรรมอื่นๆอีกมาก

แพลตฟอร์ม Pelion นั้นปัจจุบันได้มีการนำไปใช้งานกับระบบ IoT ทั้งในโปรเจกต์ขนาดใหญ่และขนาดเล็ก เช่น ระบบ IoT ในการดูแลสัตว์น้ำผ่านเชื่อต่อรับข้อมูลจากเสียง ระบบตรวจสอบสถานะการทำงานของเครื่องจักรในโรงงาน ระบบจัดการคลังสินค้า และระบบซึ่งเป็นส่วนประกอบของ Smart City เช่น ที่จอดรถอัจฉริยะ และเสาไฟฟ้าที่เปิดปิดตามความเคลื่อนไหวของคน และสามารถควบคุมได้จากระบบส่วนกลาง เป็นต้น

ท่านใดที่สนใจอยากศึกษาเกี่ยวกับ Pelion เพิ่มเติม สามารถเข้าไปอ่านเกี่ยวกับกรณีศึกษาการใช้งานในอุตสาหกรรมได้ที่ <https://www.arm.com/products/iot/pelion-iot-platform> และหากต้องการข้อมูลเชิงเทคนิคโดยละเอียด สามารถอ่าน Document เต็มๆตามลิงก์นี้ <https://www.pelion.com/docs/>

สำหรับในประเทศไทย ARM ได้จับมือเป็นพาร์ทเนอร์กับ Advantech ในการให้บริการด้านต่างๆ ท่านที่สนใจสามารถติดต่อเพื่อพูดคุยถึงโซลูชันและผลิตภัณฑ์เกี่ยวกับ IoT ของ ARM ได้ที่อีเมล์ chanchai.p@advantech.com

- <https://developers.ascendcorp.com/ทำความรู้จักกับ-grafana-dashboard-1a5efe6d170a>
Grafana Dashboard
ทำความรู้จักกับ Grafana Dashboard



Grafana คือ open source Dashboard tool เนี่ยกง่าย ๆ ก็คือเครื่องมือในการสร้าง Dashboard พรี นั่นเอง โดย Grafana จะทำงานร่วมกับ Datasource ต่าง ๆ เช่น Graphite, InfluxDB, OpenTSDB หรือ Elasticsearch ฯลฯ ช่วยให้ users สามารถสร้างและแก้ไข Dashboard ได้อย่างง่ายๆ ครอบคลุมรูปแบบกราฟหลากหลายประเภท

จุดเด่นของ Grafana

- เน้นการนำเสนอ Metrics ที่เฉพาะเจาะจง เช่น CPU, Memory หรือ I/O ในรูปแบบของกราฟ Time series
- มี Role-based access ในการจัดการ user ในการเข้าใช้งานให้ในตัว
- ความยืดหยุ่นในการใช้งาน มี option ให้เลือกใช้จำนวนมาก
- รองรับ datasource ที่หลากหลายและมี query editor ที่สำหรับ datasource นั้นๆ

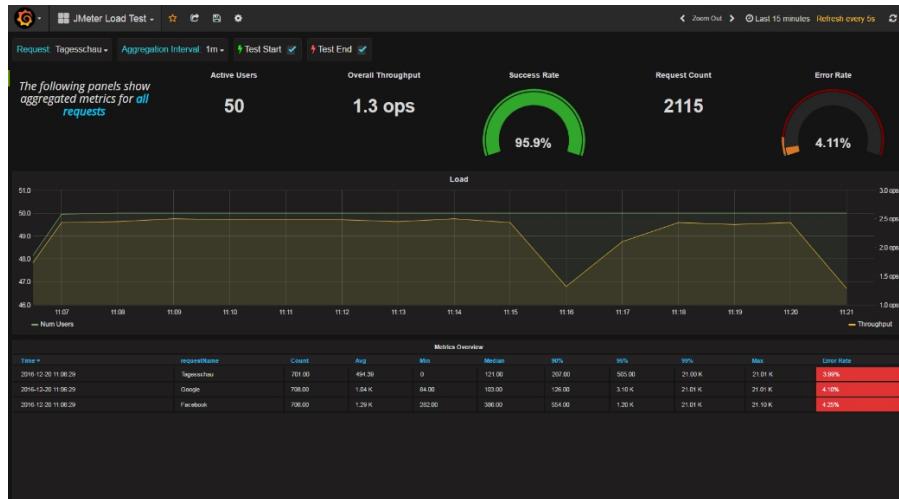
ตัวอย่างการใช้งาน Grafana Dashboard

- Monitoring Server ใช้งานร่วมกับ Influxdb และ Telegraf



credit : <https://grafana.com/dashboards/1443>

- Monitoring Realtime result สำหรับ Jmeter ใน non-gui mode



credit : <https://grafana.com/dashboards/1152>

การติดตั้ง Grafana Dashboard

- ดาวน์โหลด [ที่นี่](#)

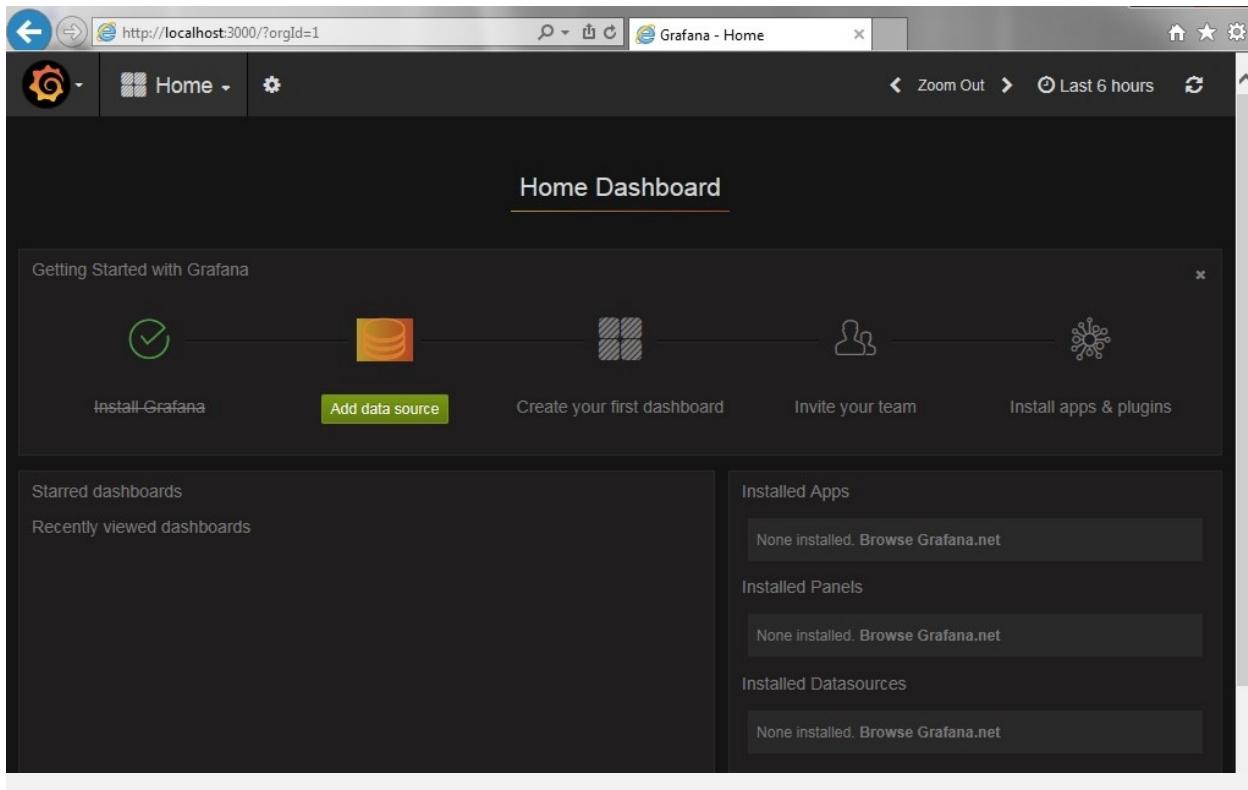
สำหรับ Windows (x64)

- สามารถใช้ grafana-server.exe เพื่อเริ่มใช้งานได้ทันที
- กรณีต้องการระบุ custom config ดูรายละเอียด [ที่นี่](#)
สำหรับ Mac (Via [Homebrew](#))

```
C:\grafana-4.4.1>bin\grafana-server.exe --config conf\custom.ini
C:\[32nINFO+[@m[07-28:22:33:40] Starting Grafana           +[32nlogg
er+[@m=main +[32mversion+[0n=4.4.1 +[32mcommit+[0m=6a9f8caa4 +[32mcompiled+[0m=2
017-07-05T14:15:04+0000
*[32nINFO+[@m[07-28:22:33:40] Config loaded from      +[32nlogg
er+[@m=settings +[32mfile+[0m=C:\\grafana-4.4.1\\conf\\defaults.ini   +[32nlogg
*[32nINFO+[@m[07-28:22:33:40] Config loaded from      +[32nlogg
er+[@m=settings +[32mfile+[0m=conf\\custom.ini       +[32nlogg
*[32nINFO+[@m[07-28:22:33:40] Path Home             +[32nlogg
er+[@m=settings +[32mpath+[0m=C:\\grafana-4.4.1      +[32nlogg
*[32nINFO+[@m[07-28:22:33:40] Path Data            +[32nlogg
er+[@m=settings +[32mpath+[0m=C:\\grafana-4.4.1\\data  +[32nlogg
*[32nINFO+[@m[07-28:22:33:40] Path Log             +[32nlogg
er+[@m=settings +[32mpath+[0m=C:\\grafana-4.4.1\\data\\log +[32nlogg
*[32nINFO+[@m[07-28:22:33:40] Path Plugins        +[32nlogg
er+[@m=settings +[32mpath+[0m=C:\\grafana-4.4.1\\data\\plugins +[32nlogg
*[32nINFO+[@m[07-28:22:33:40] Initializing DB         +[32nlogg
er+[@m=sqldstore +[32mdbtype+[0m.sqlite3          +[32nlogg
*[32nINFO+[@m[07-28:22:33:40] Starting DB migration +[32nlogg
er+[@m=migrator
*[32nINFO+[@m[07-28:22:33:40] Executing migration    +[32nlogg
er+[@m=migrator +[32mid+[0m="copy data account to org"
*[32nINFO+[@m[07-28:22:33:40] Skipping migration condition not fulfilled +[32nlogg
er+[@m=migrator +[32mid+[0m="copy data account to org"
*[32nINFO+[@m[07-28:22:33:40] Executing migration    +[32nlogg
er+[@m=migrator +[32mid+[0m="copy data account_user to org_user"
*[32nINFO+[@m[07-28:22:33:40] Skipping migration condition not fulfilled +[32nlogg
er+[@m=migrator +[32mid+[0m="copy data account_user to org_user"
*[32nINFO+[@m[07-28:22:33:40] Starting plugin search     +[32nlogg
er+[@m=plugins
*[32nINFO+[@m[07-28:22:33:41] Initializing Alerting      +[32nlogg
er+[@m=alerting_engine
*[32nINFO+[@m[07-28:22:33:41] Initializing CleanUpService +[32nlogg
er+[@m=cleanup
*[32nINFO+[@m[07-28:22:33:41] Initializing Stream Manager +[32nlogg
*[32nINFO+[@m[07-28:22:33:41] Initializing HTTP Server    +[32nlogg
er+[@m=http.server +[32maddress+[0m=0.0.0.0:3000 +[32mprotocol+[0n=http +[32msub
Url+[0m= +[32msocket+[0m=
```

```
:~ brew update
:~ brew install grafana
:~ brew services start grafana
```

- เมื่อทำการติดตั้งและ start service เรียบร้อยแล้ว เริ่มต้นใช้งานโดย default port ของ grafana คือ 3000
- เข้าใช้งานโดย <http://localhost:3000> และ user/password เริ่มต้นคือ admin/admin



เท่านี้ก็สามารถเริ่มต้นใช้งาน Grafana Dashboard ได้แล้ว ครั้งหน้าจะมาแนะนำการใช้งานร่วมกับ Influxdb ในการ Monitoring Server และ Monitoring realtime jmeter ~~~*

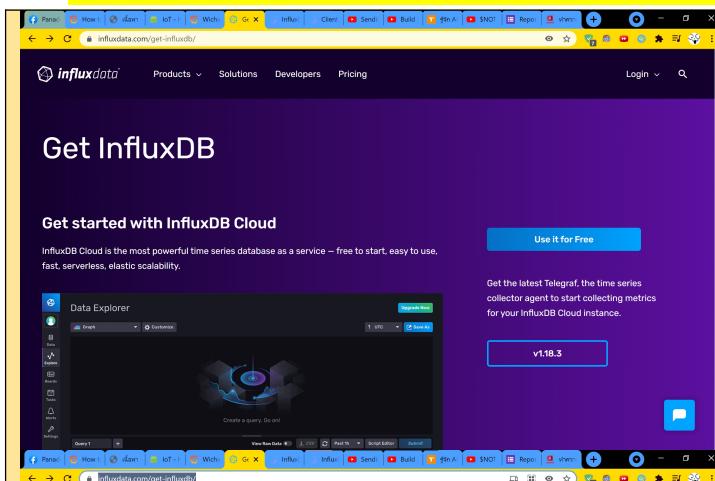
credit : <https://logz.io/blog/grafana-vs-kibana/>

4/4. การทดสอบ

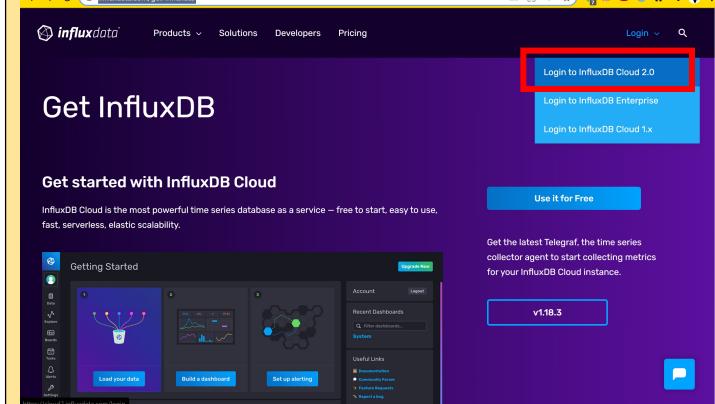
- ให้ทำการทดสอบและเขียนขั้นตอนการทดสอบ โดยใช้ ESP32 ส่งข้อมูลไปยัง MQTT Broker และใช้ Grafana .ในการมองเห็นข้อมูล โดยปรับแก้การทดสอบจาก <https://gabrieltanner.org/blog/grafana-sensor-visualization>

Visualize Sensor data using Grafana and InfluxDB

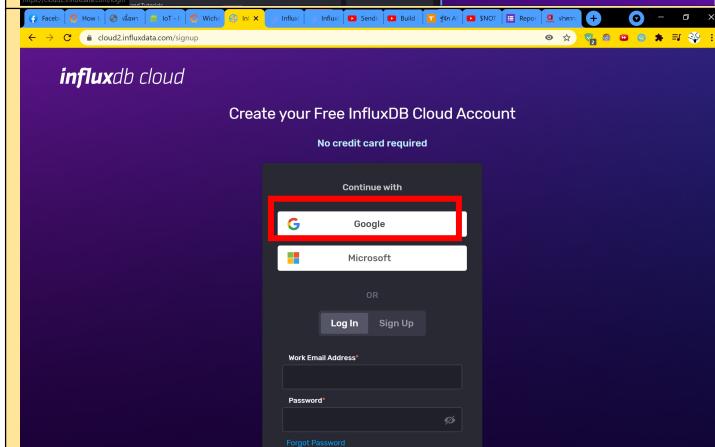
1. การสมัครเข้าใช้งาน ติดตั้ง InfluxDB และสร้าง key สำคัญ

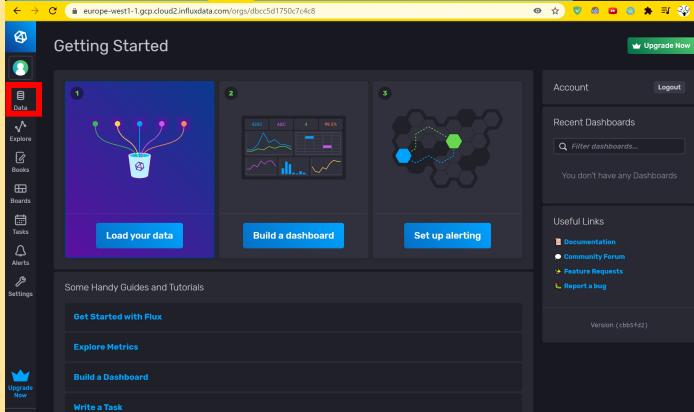
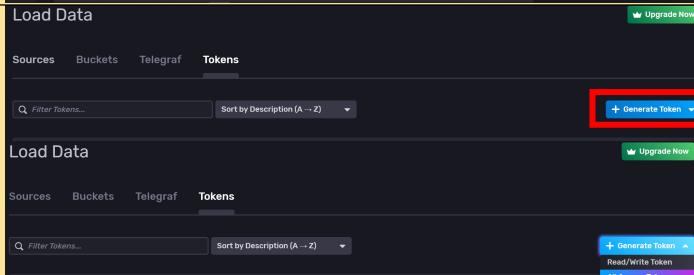
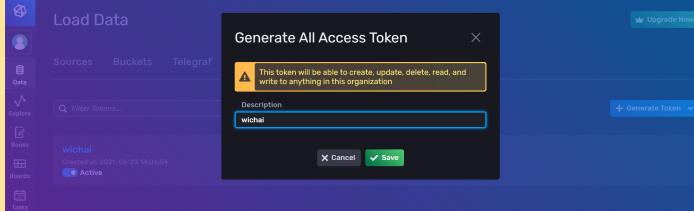
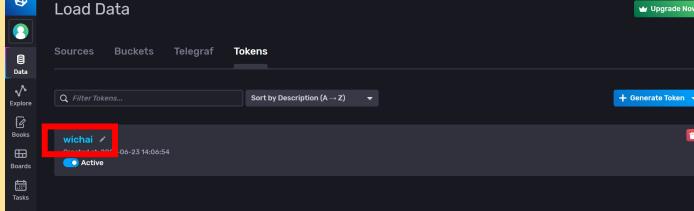
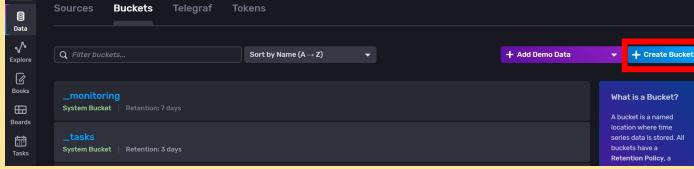


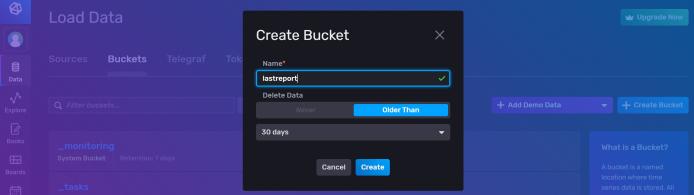
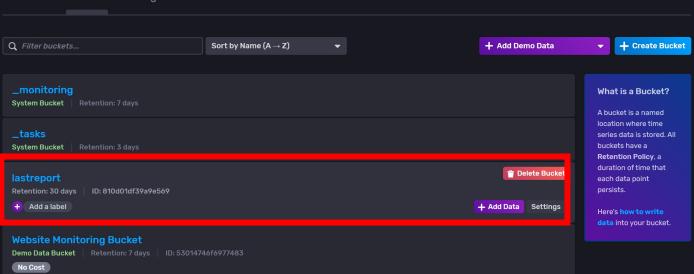
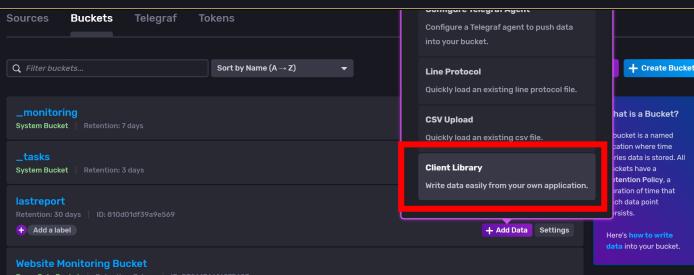
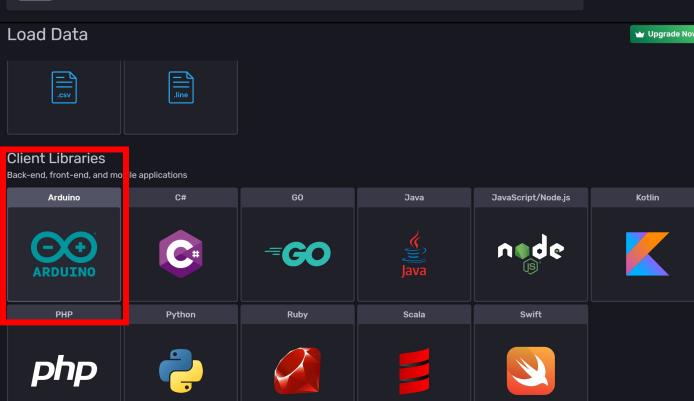
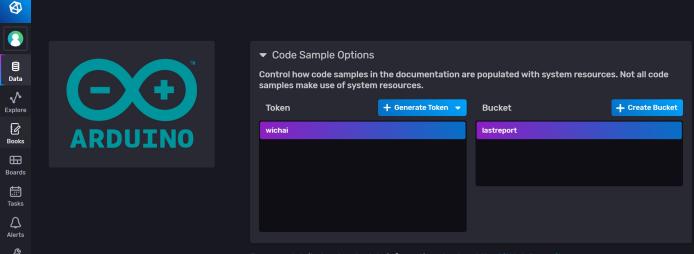
เข้าไปที่ <https://www.influxdata.com/get-influxdb/> เพื่อสมัครเข้าใช้งาน โดยกดเลือกอินมูมขวาบนและเลือกตัวเลือกตามภาพ



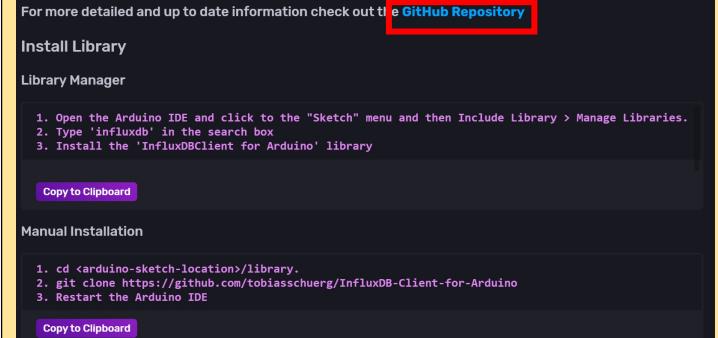
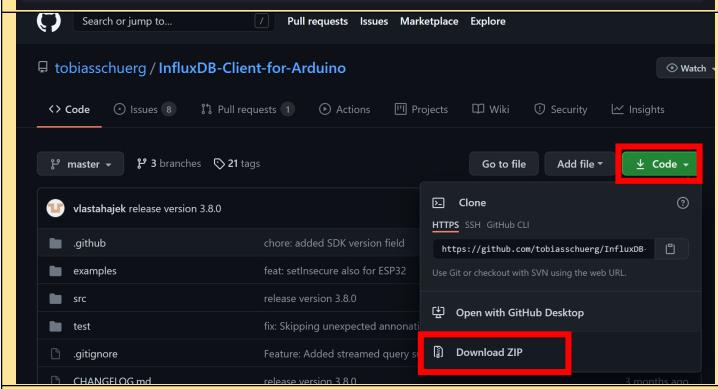
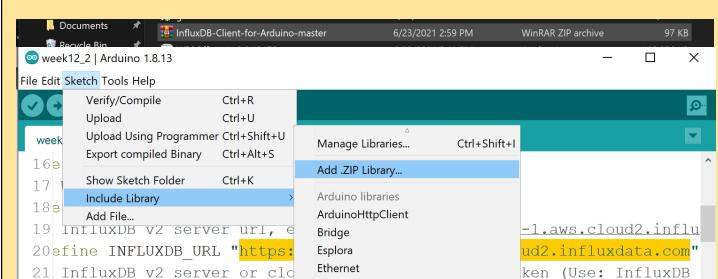
ระบบจะเข้ามาสู่หน้าสมัครสมาชิก โดยในที่นี่จะเลือกการเข้าระบบด้วยบัญชี Google



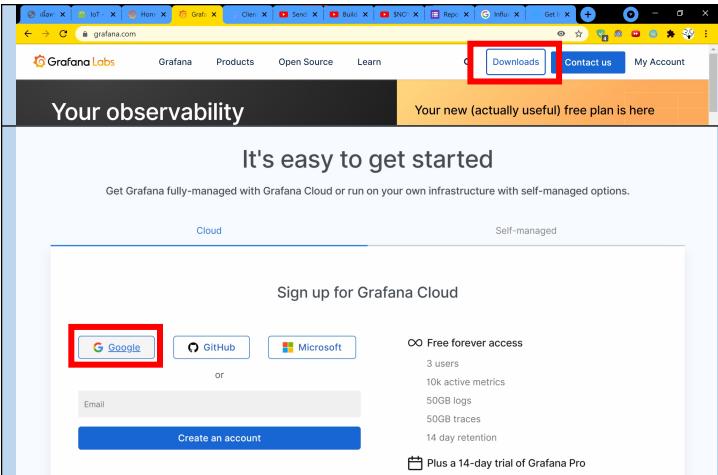
	<p>เมื่อเข้าสู่ระบบเรียบร้อยแล้วจะมาที่หน้าต่างหลักของโปรแกรม akan นี้ให้ไปที่ tab Data</p>
	<p>คลิกเลือก tab tokens และเลือก create token เลือกประเภทเป็น All access token ตามรูป</p>
	<p>ตั้งชื่อของ token ในช่อง description เสร็จแล้วกด save</p>
	<p>Token ที่สร้างจะขึ้นเป็นลิสต์ด้านล่าง สามารถกดที่ชื่อเพื่อดู token และทำการคัดลอกเก็บไว้</p>
	<p>จากนี้ไปที่หน้า Bucket เพื่อสร้างสัดส่วนการเก็บข้อมูล akan นี้เลือก create bucket</p>
	

	<p>ตั้งชื่อตามที่ต้องการ สามารถเปลี่ยนเวลาการเคลียร์ข้อมูลได้ จากนั้นกด create</p>
	<p>เมื่อสร้างเสร็จสิ้นจะขึ้น bucket ตามในลิสต์</p>
	<p>จากนั้นให้กด Add data เพื่อเพิ่มข้อมูลแบบ client library เพื่อรับข้อมูลจาก ESP 32</p>
	<p>เลื่อนหา และคลิกเลือก Arduino</p>
	<p>ระบบจะปรากฏหน้าต่างดังภาพ ให้เลือก token และ bucket ตามที่เราได้สร้างมาในข้างต้นให้ถูกต้อง</p>

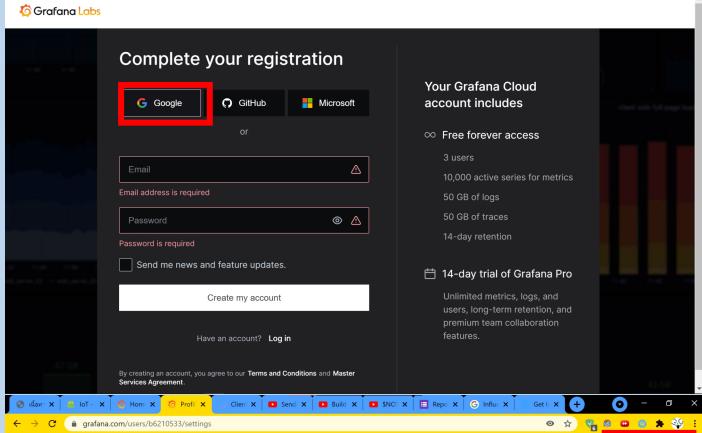
2. การติดตั้ง library ใน arduino เพื่อส่งข้อมูลเข้าใน InfluxDB

 <p>For more detailed and up to date information check out the GitHub Repository</p> <p>Install Library</p> <p>Library Manager</p> <ol style="list-style-type: none"> 1. Open the Arduino IDE and click to the "Sketch" menu and then Include Library > Manage Libraries. 2. Type 'influxdb' in the search box 3. Install the 'InfluxDBClient for Arduino' library <p>Copy to Clipboard</p> <p>Manual Installation</p> <ol style="list-style-type: none"> 1. cd <arduino-sketch-location>/library. 2. git clone https://github.com/tobiasschuerig/InfluxDB-Client-for-Arduino 3. Restart the Arduino IDE <p>Copy to Clipboard</p>	<p>จากเว็บไซต์ https://europe-west1-1.gcp.cloud2.influxdata.com/orgs/dbcc5d1750c7c4c8/load-data/client-libraries/arduino</p> <p>เมื่อเลื่อนลงมาจะเจอกับ tutorial สำหรับการส่งค่าเข้า InfluxDB โดยในขั้นแรกจะทำการลง library ภายใน arduino แบบ manual โดยให้คลิกตามลิ้งค์ของ Github Repository</p>
 <p>Search or jump to... Pull requests Issues Marketplace Explore</p> <p>tobiasschuerig / InfluxDB-Client-for-Arduino</p> <p><> Code Issues 8 Pull requests 1 Actions Projects Wiki Security Insights</p> <p>master 3 branches 21 tags Go to file Add file Code</p> <p>vlastahajek release version 3.8.0</p> <p>.github chore: added SDK version field</p> <p>examples feat: setInsecure also for ESP32</p> <p>src release version 3.8.0</p> <p>test fix: Skipping unexpected annotation</p> <p>.gitignore Feature: Added streamed query support</p> <p>CHANGELOG.md release version 3.8.0 3 months ago</p> <p>Clone HTTPS SSH GitHub CLI https://github.com/tobiasschuerig/InfluxDB-Client-for-Arduino</p> <p>Use Git or checkout with SVN using the web URL</p> <p>Open with GitHub Desktop</p> <p>Download ZIP</p>	<p>ลิ้งค์จะนำเข้าสู่เว็บໄว์เต็มๆ ของ Influx Client for Arduino ให้กดที่ code แล้วเลือก download ZIP เพื่อทำการดาวน์โหลด library</p>
 <p>File Edit Sketch Tools Help</p> <p>week12_2 Arduino 1.8.13</p> <p>Verify/Compile Ctrl+R</p> <p>Upload Ctrl+U</p> <p>Upload Using Programmer Ctrl+Shift+U</p> <p>Export compiled Binary Ctrl+Alt+S</p> <p>Show Sketch Folder Ctrl+K</p> <p>Include Library</p> <p>Add File...</p> <p>19 INFLUXDB_V2 server url, e</p> <p>20 define INFLUXDB_URL "https://</p> <p>21 InfluxDB v2 server or clc</p> <p>Manage Libraries... Ctrl+Shift+I</p> <p>Add ZIP Library...</p> <p>Arduino libraries</p> <p>ArduinoHttpclient</p> <p>Bridge</p> <p>Espola</p> <p>Ethernet</p> <p>-1.aws.cloud2.influxdb2.influxdata.com"</p> <p>ken (Use: InfluxDB)</p>	<p>ให้ทำการติดตั้ง library โดยเปิด Arduino และเลือก Add ZIP library จากนั้นให้กดเลือกไฟล์ที่โหลดได้จาก GitHub และ Arduino จะลง library ให้อัตโนมัติ โดยสามารถเช็คไฟล์ได้ที่ Include Library</p>

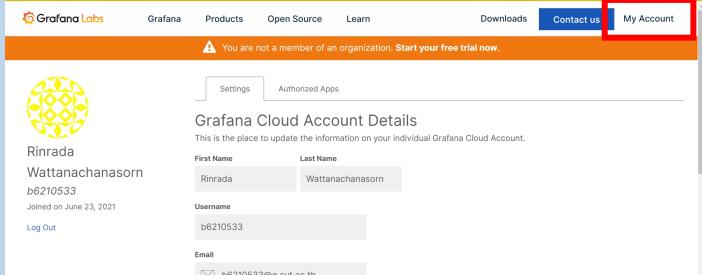
3. การสมัครเข้าใช้งาน และติดตั้ง Grafana

 <p>Your observability</p> <p>It's easy to get started</p> <p>Get Grafana fully-managed with Grafana Cloud or run on your own infrastructure with self-managed options.</p> <p>Cloud Self-managed</p> <p>Sign up for Grafana Cloud</p> <p>Google GitHub Microsoft</p> <p>or</p> <p>Email</p> <p>Create an account</p> <p>Free forever access</p> <p>3 users 10k active metrics 50GB logs 50GB traces 14 day retention</p> <p>Plus a 14-day trial of Grafana Pro</p>	<p>เข้าเว็บไซต์ https://grafana.com/ จากนั้นกด download ที่มุมขวาบน</p> <p>เว็บไซต์จะนำมาที่หน้าการสมัครสมาชิก โดยในนี้จะใช้ account google เดียวกันกับที่สมัคร InflexDB</p>
---	---

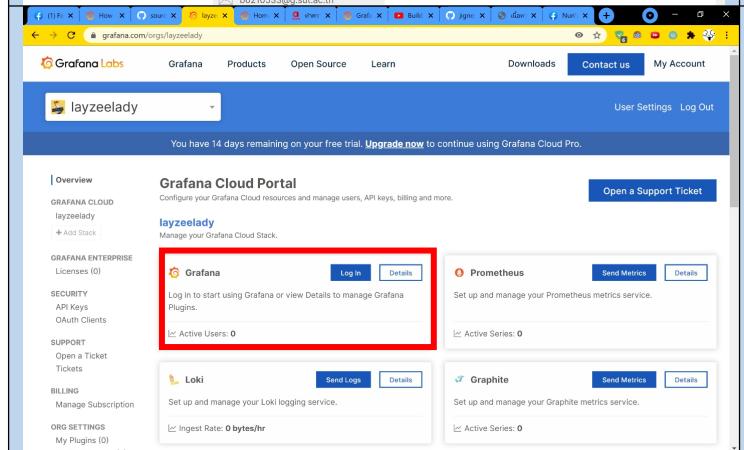
จากนั้นจะให้กรอกข้อมูลเพิ่ม หากเป็นบัญชี google ให้คลิกที่ google ซึ่ง เมื่อสมัครเสร็จเรียบร้อยจะขึ้นหน้าต่างสืขาระดับล่าง ว่าสามารถใช้ cloud ของ Grafana ได้ คลิกที่ My Account หวาน



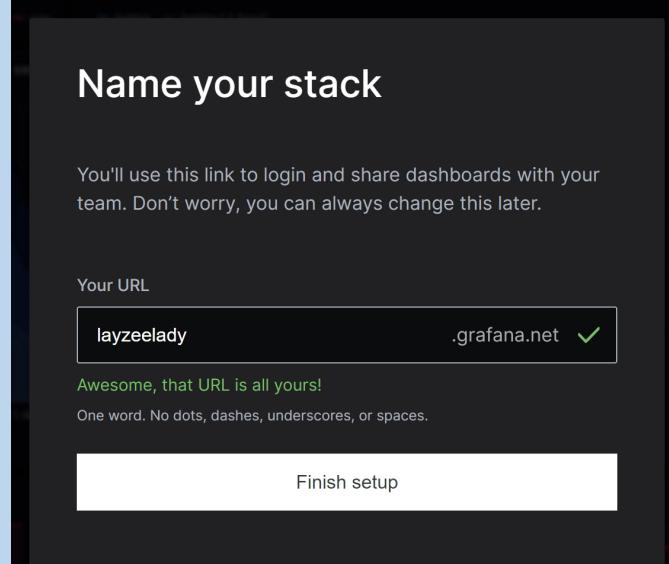
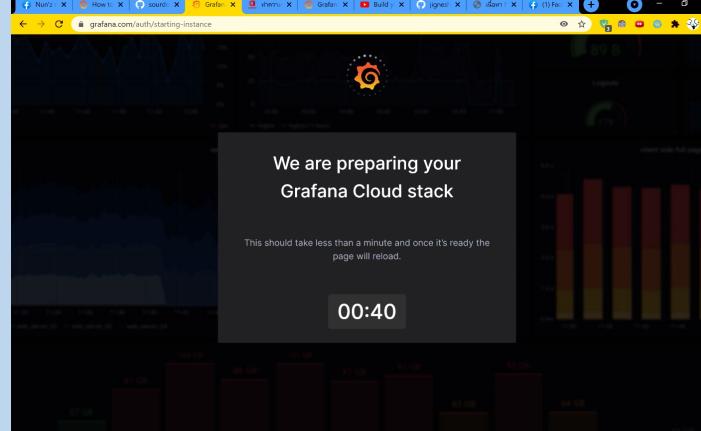
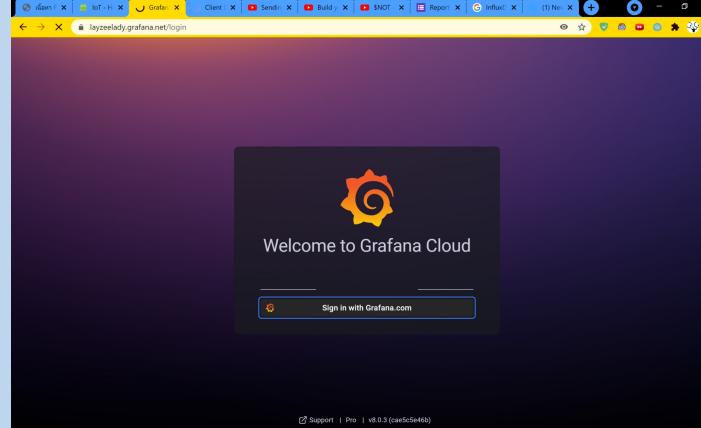
After registration, the user is directed to their account settings. A message indicates they are not part of an organization and encourages them to start a free trial.

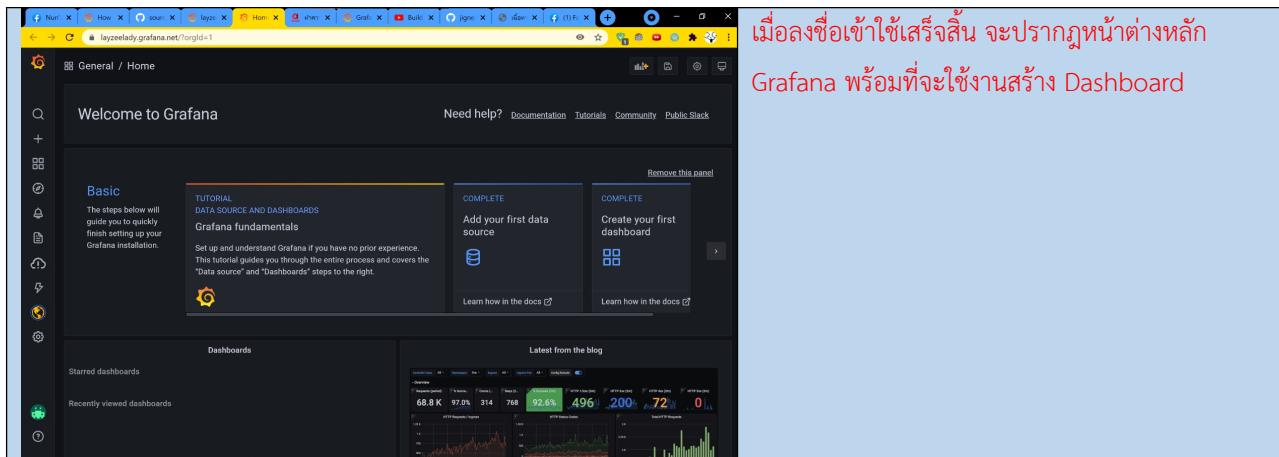


The user then logs in to the Grafana Cloud Portal. They have 14 days remaining on their free trial.



เลือก log-in เข้า Grafana

 <p>Name your stack</p> <p>You'll use this link to login and share dashboards with your team. Don't worry, you can always change this later.</p> <p>Your URL</p> <p>layzeelady .grafana.net ✓</p> <p>Awesome, that URL is all yours!</p> <p>One word. No dots, dashes, underscores, or spaces.</p> <p>Finish setup</p>	<p>ในครั้งแรก เว็บไซต์จะให้เราต้องชื่อลิ้งค์ เพื่อไปที่หน้าหลัก Grafana ประจำ account โดยในเบื้องต้นจะมีชื่อคล้ายกับ e-mail ของเรา สามารถเปลี่ยนได้ตามชอบ จากนั้นคลิก Finish setup</p>
 <p>We are preparing your Grafana Cloud stack</p> <p>This should take less than a minute and once it's ready the page will reload.</p> <p>00:40</p>	<p>เว็บไซต์จะทำการประมวลผล และจัดเตรียม cloud ของ Grafana ให้พร้อมใช้งาน</p>
 <p>Welcome to Grafana Cloud</p> <p>Sign in with Grafana.com</p>	<p>ใช้ลิ้งค์เข้าใช้งานที่กรอกลงช่องเข้าใช้กับเว็บในข้างต้น คัดลอกลงมาใส่ใน address bar ของ browser โดยจะมีชื่อเข้าใช้ และรหัสผ่านเริมต้นเป็น admin และ admin</p>



เมื่อลงชื่อเข้าใช้เสร็จล้วน จะปรากฏหน้าต่างหลัก

Grafana พร้อมที่จะใช้งานสร้าง Dashboard

4. การเขียนโปรแกรมใน ESP32 เพื่อส่งค่าเข้าไปยัง InfluxDB

<p>Initialize the Client</p> <pre>Serial.println(); // Add tags sensor.addTag("device", DEVICE); sensor.addTag("SSID", WiFi.SSID()); // Accurate time sync required for certificate validation and writing in batches // For the fastest time sync find NTP servers in your area: https://www.pool.ntp.org/zone/ // Syncing progress and the time will be printed to Serial. timeSync(TZ_INFO, "pool.ntp.org", "time.nis.gov"); // Check server connection if (client.connect("InfluxDB")) { Serial.print("Connected to InfluxDB: "); Serial.println(client.getServerUrl()); } else { Serial.print("InfluxDB connection failed: "); Serial.println(client.getLastErrorMessage()); }</pre> <p>Copy to Clipboard</p>	<p>จากเว็บไซต์ https://europe-west1-1.gcp.cloud2.influxdata.com/orgs/dbcc5d175c7c48/load-data/client-libraries/arduino จะมีตัวอย่างการเขียนโปรแกรมทั้งหมด 3 ส่วน แต่ที่จำเป็นในการทดลองจะมีสองส่วนคือ ส่วนเข้าเชื่อมต่อ กับ Database โดยในที่นี้จะต้องใช้คีย์สำคัญที่สมัคร และสร้างไว้กับ InfluxDB</p>
<p>Write Data</p> <pre>void loop() { // Clear fields for reusing the point. Tags will remain untouched sensor.clearFields(); // Store measured value into point // Report RSSI of currently connected network sensor.addField("rssi", WiFi.RSSI()); // Print what we are exactly writing Serial.print("Writing: "); Serial.println(sensor.toLineProtocol()); // If no WiFi signal, try to reconnect it if ((WiFi.RSSI() == 0) && (wifiMulti.run() != WL_CONNECTED)) { Serial.println("Wifi connection lost"); } // Write point if (!client.writePoint(sensor)) { Serial.print("InfluxDB write failed: "); Serial.println(client.getLastErrorMessage()); } }</pre> <p>Copy to Clipboard</p>	<p>ส่วนที่สองจะเป็นโปรแกรมในส่วน loop ทำการส่งค่าแบบ real time จาก ESP32 board เข้าสู่ InfluxDB</p>
<p>Organization</p> <p>The screenshot shows the 'Organization' settings page. It has tabs for 'Users' and 'About'. Under 'About', there's a 'Danger Zone!' button for renaming the organization. Below that are 'Common IDs' for a user and an organization, each with a 'Copy to Clipboard' button.</p>	<p>คีย์สำคัญที่ต้องใช้งานได้แก่ Organization, token, buckets หาได้จากเว็บ InfluxDB > Profile และ URL ดูจากเว็บไซต์ที่เราใช้เข้าไปใน InfluxDB โดยในที่นี้คือ https://europe-west1-1.gcp.cloud2.influxdata.com</p>

โค้ดโปรแกรมส่งค่าสุ่มอุณหภูมิ และความชื้นใน ESP32

```
#if defined(ESP32)
#include <WiFiMulti.h>
WiFiMulti wifiMulti;
#define DEVICE "ESP32"
#elif defined(ESP8266)
#include <ESP8266WiFiMulti.h>
ESP8266WiFiMulti wifiMulti;
#define DEVICE "ESP8266"
#endif

float temp;
float humd;

#include <InfluxDbClient.h>
#include <InfluxDbCloud.h>

// WiFi AP SSID
#define WIFI_SSID "U"
// WiFi password
#define WIFI_PASSWORD "11111111"

#define INFLUXDB_URL "https://europe-west1-1.gcp.cloud2.influxdata.com"
#define INFLUXDB_TOKEN
"oPkcdVx_wbkPWXQYz0FOLYgIm8nDFgv6fB33hOZDrd0zUV1Z_7Pvokp4N6Fd1KoHjsPOO3XESC3IKZZB1Orm6A=="
#define INFLUXDB_ORG "layzeelady@gmail.com"
#define INFLUXDB_BUCKET "lastreport"

// Set timezone string according to https://www.gnu.org/software/libc/manual/html\_node/TZ-Variable.html

#define TZ_INFO "UTC+7"

// InfluxDB client instance with preconfigured InfluxCloud certificate
InfluxDBClient client(INFLUXDB_URL, INFLUXDB_ORG, INFLUXDB_BUCKET, INFLUXDB_TOKEN, InfluxDbCloud2CACert);

// Data point
Point sensor("temp&humid");

void setup() {
  Serial.begin(115200);

  // Setup wifi
  WiFi.mode(WIFI_STA);
  wifiMulti.addAP(WIFI_SSID, WIFI_PASSWORD);

  Serial.print("Connecting to wifi");
}

// Task to send data to InfluxDB
void loop() {
  if (client.connected()) {
    Point p;
    p.add("temp", temp);
    p.add("humd", humd);
    client.sendPoint(p);
  }
}
```

สร้างตัวแปรเพื่อกีบค่าสุ่มอุณหภูมิ และความชื้น แบบ float

ทำการกรอกคีย์สำคัญที่ได้จากข้างต้น บนเว็บ InfluxDB

กำหนด_timezone ให้มีชื่อประจำประเทศไทย คือ UTC+7

ทำการเชื่อมต่อเข้าสู่ InfluxDB ตามคีย์สำคัญ

ส่งค่าที่ได้จาก ESP32 นี้ไปยัง sensor ทั่วโลกที่ชื่อ temp&humid

ทำการเชื่อมต่อไฟฟ้าตามค่าที่ตั้งไว้ด้านบน

```

while (wifiMulti.run() != WL_CONNECTED) {
    Serial.print(".");
    delay(100);
}
Serial.println();

// Add tags
sensor.addTag("device", DEVICE);
sensor.addTag("ssid", WiFi.SSID());

```

addTag เพื่อบุข้อมูลว่าส่งจาก device และ network ใด


```

// Accurate time is necessary for certificate validation and writing in batches
// For the fastest time sync find NTP servers in your area: https://www.pool.ntp.org/zone/
// Syncing progress and the time will be printed to Serial.
timeSync(TZ_INFO, "pool.ntp.org", "time.nis.gov");

```

ทำการเพิ่มเว็บไซต์เพื่อ
ทำ timesync


```

// Check server connection
if (client.validateConnection()) {
    Serial.print("Connected to InfluxDB: ");
    Serial.println(client.getServerUrl());
} else {
    Serial.print("InfluxDB connection failed: ");
    Serial.println(client.getLastErrorMessage());
}
}

void loop() {
    temp = (random(280,350)/10.0);
    humd = (random(650,780)/10.0);

```

ทำการเช็คการเข้าถึง InfluxDB และแจ้งหากมีปัญหา

ทำการสุ่มค่า random ระหว่าง temp และ humd เป็น float


```

// Clear fields for reusing the point. Tags will remain untouched
sensor.clearFields();

// Store measured value into point
// Report RSSI of currently connected network
sensor.addField("rssi", WiFi.RSSI());
sensor.addField("temp", temp);
sensor.addField("humid", humd);

```

เพิ่ม filed ข้อมูลเพื่อแยกประเภทข้อมูลแต่ละชนิด


```

// Print what are we exactly writing
Serial.print("Writing: ");
Serial.print("temp = "); Serial.println(temp);
Serial.print("humid = "); Serial.println(humd);
Serial.println(sensor.toLineProtocol());

// If no Wifi signal, try to reconnect it
if ((WiFi.RSSI() == 0) && (wifiMulti.run() != WL_CONNECTED)) {
    Serial.println("Wifi connection lost");
}

// Write point
if (!client.writePoint(sensor)) {
    Serial.print("InfluxDB write failed: ");

```

ทำการแจ้งเตือนสิ่งที่เขียนออกที่จอ serial monitor

ทำการเขียนลง InfluxDB หากมีปัญหาให้แจ้งเตือน

```

Serial.println(client.getLastErrorMessage());
}

//Wait 10s
Serial.println("Wait 1s");
delay(1000);
}

```

ทำการส่งค่าใหม่ทุกๆ 1 วินาทีซึ่ง

humid = ,
70.40
temp&humid,device=ESP32,ssid=U rssi=-35i,temp=32.10,humid=70.40
Wait 1s
Writing: temp = ,33.40
humid = ,
65.70
temp&humid,device=ESP32,ssid=U rssi=-27i,temp=33.40,humid=65.70
Wait 1s
Writing: temp = ,34.90
humid = ,
72.30
temp&humid,device=ESP32,ssid=U rssi=-26i,temp=34.90,humid=72.30

Autoscroll Show timestamp

Newline 115200 baud Clear output

time zone thailand - Bing

time zone thailand

ALL WORK IMAGES VIDEOS NEWS

2,010,000 Results Open links in new tab

Thailand Timezone Visa Climate Official language Economy Culture Geography History Political system

Current time in Bangkok, Thailand (UTC+7)

4:01 PM

Wednesday, 23 June 2021

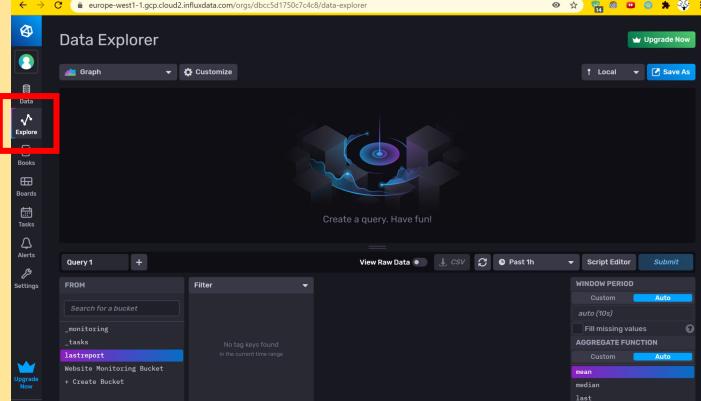
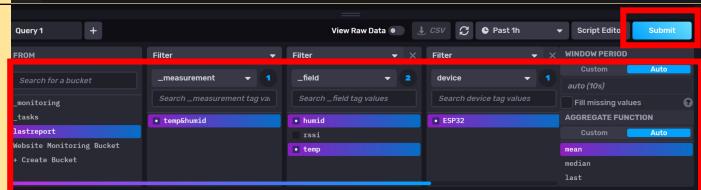
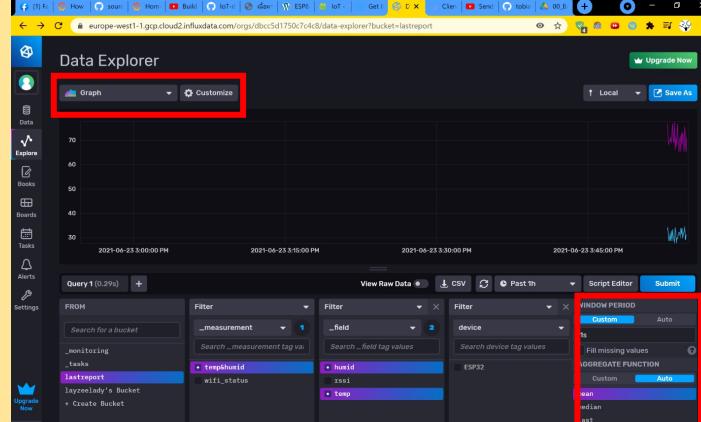
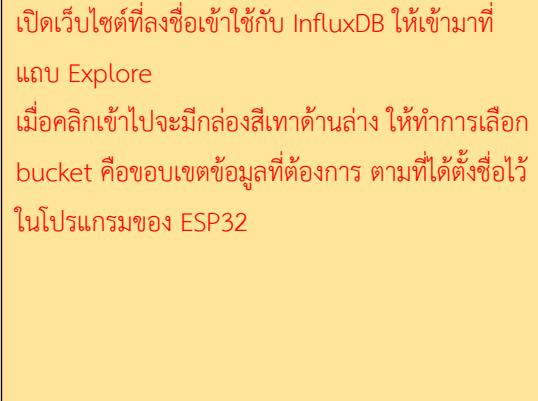
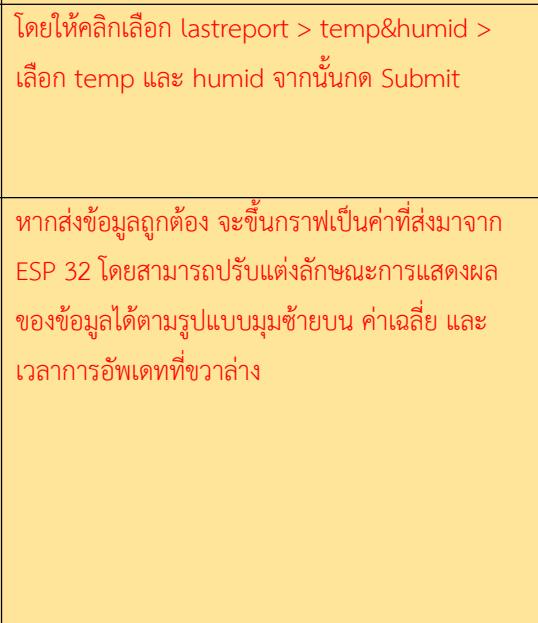
Time zone. Indochina Time (ICT), **UTC +7**. No daylight saving time, same UTC offset all year. The IANA time zone identifier for Thailand is

Thailand

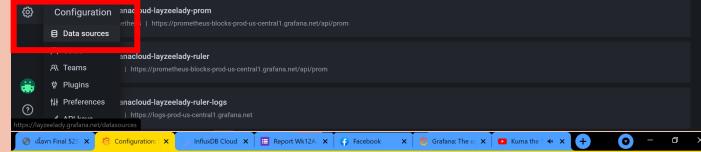
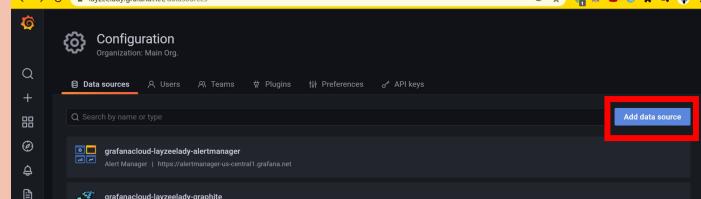
All images

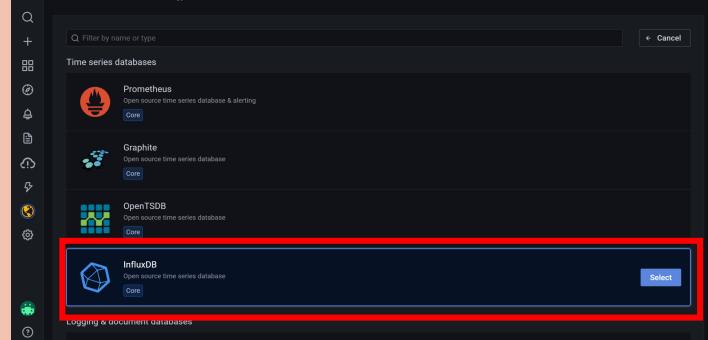
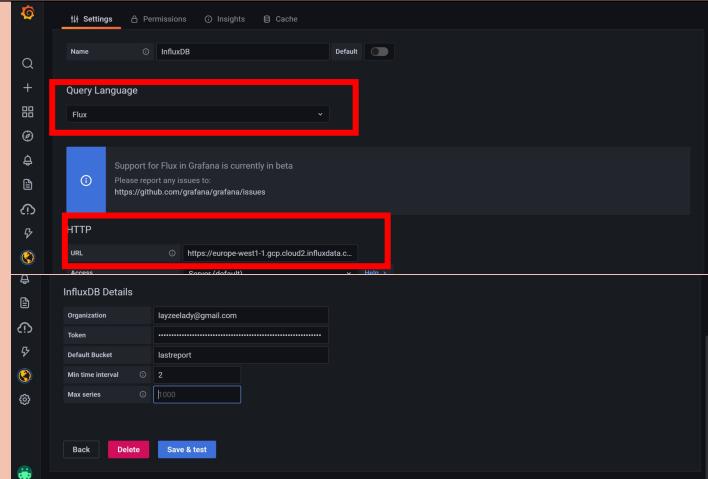
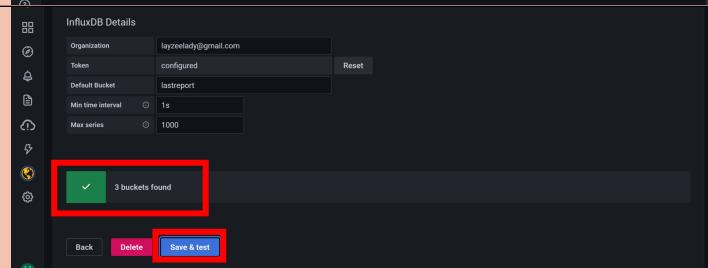
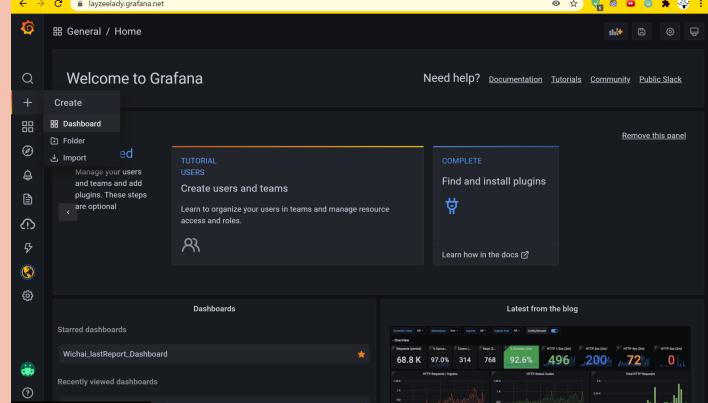
COUNTRY IN SOUTHEAST ASIA

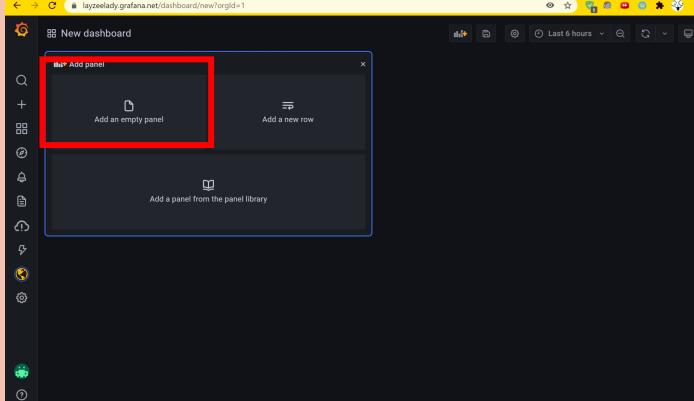
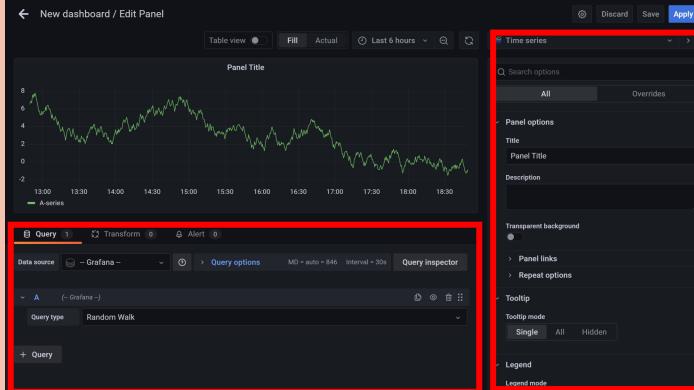
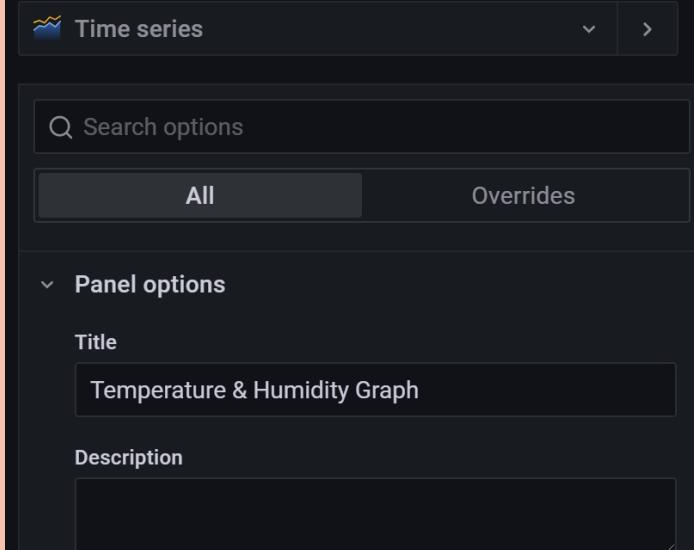
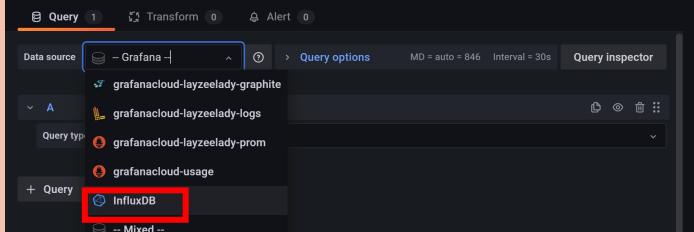
5. เช็คค่าข้อมูลใน InfluxDB

  	<p>เปิดเว็บไซต์ที่ลงชื่อเข้าใช้กับ InfluxDB ให้เข้ามาที่ แดบ Explore</p> <p>เมื่อคลิกเข้าไปจะมีกล่องสีเทาด้านล่าง ให้ทำการเลือก bucket คือขอบเขตข้อมูลที่ต้องการ ตามที่ได้ตั้งชื่อไว้ ในโปรแกรมของ ESP32</p>
	<p>โดยให้คลิกเลือก lastreport > temp&humid > เลือก temp และ humid จากนั้นกด Submit</p>
	<p>หากส่งข้อมูลถูกต้อง จะขึ้นกราฟเป็นค่าที่ส่งมาจาก ESP 32 โดยสามารถปรับแต่งลักษณะการแสดงผล ของข้อมูลได้ตามรูปแบบบุณฑ์ข้างบน ค่าเฉลี่ย และ เวลาการอัพเดทที่ข่าวล่าง</p>

6. สร้าง Grafana Dashboard จากข้อมูลใน InfluxDB และการปรับแต่งค่า

 	<p>มาที่เว็บไซต์ Dashboard ของ Grafana แล้วไปที่ แดบเครื่องมือ เลือก Data sources เพื่อเพิ่ม ฐานข้อมูล แล้วเลือก Add data source</p>
--	---

	<p>คลิกเลือก InfluxDB</p>
	<p>ตั้งค่า กรอกข้อมูลตาม key สำคัญที่สร้างขึ้นใน InfluxDB โดยเลือก Query Language เป็น Flux กรอก URL ใน HTTP ตาม URL ของเว็บ Influx ที่ใช้ กรอก InfluxDB Details เป็น ORG, Token, Default Bucket ตามที่ได้สมัครไว้ และกรอก Min time interval = 2</p>
	<p>หากเข้ามายังหน้า database สำเร็จ เมื่อคลิก Save and test จะขึ้นจำนวน buckets ทั้งหมดใน database ที่เจอ</p>
	<p>กลับมาที่เว็บไซต์ Dashboard ของ Grafana ที่ทำ การตั้งค่าไว้ในข้างต้น แล้วเลือกแท็บ + > Create Dashboard</p>

	<p>เว็บไซต์จะปรากฏให้เพิ่ม panel สำหรับการแสดงผลข้อมูล คลิกเลือก Add an empty panel</p>
	<p>หน้าต่างจะปรากฏหน้าเริ่มต้นเป็นการสร้าง graph แบบ time series ด้านล่างเป็นส่วนการกำหนดการแสดงผลของข้อมูล ด้านขวาเป็นการกำหนดการตั้งค่า panel</p>
	<p>โดยสำหรับการกำหนดรูปแบบของ panel จะเป็นการตั้งค่าภายในต่างๆ เกี่ยวกับ panel เช่น สี ลักษณะเส้นกราฟ ชื่อกราฟและแกน โดยในที่นี้จะทำการสร้าง Time series ของ temp และ humd ก่อน</p>
	<p>เลื่อนลงมาที่ Query ให้ทำการเลือก Data Source ที่ต้องการ โดยในที่นี้จะเลือก InfluxDB ตามที่ทำการ Add data source ไว้ในข้างต้น</p>

หน้าต่างจะปรากฏเป็นหน้าโล่งๆ ให้ทำการโปรแกรม
การแสดงผลในส่วน Query และปรับแต่งให้เหมาะสม

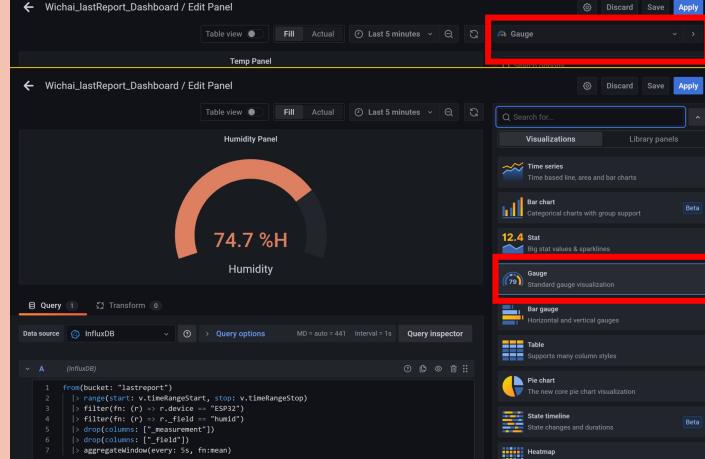
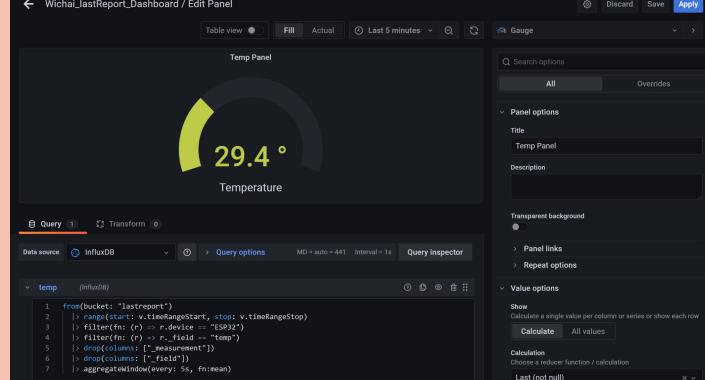
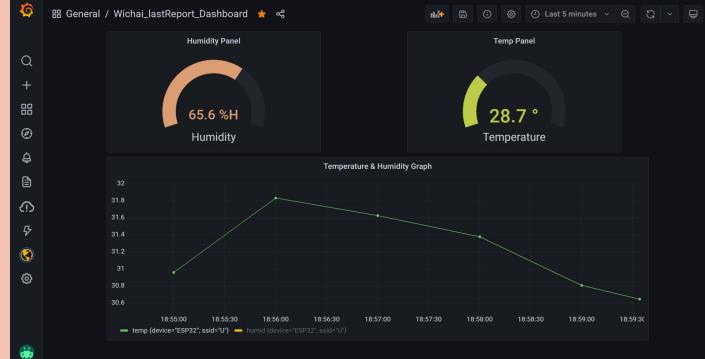
กรอกโค้ดเพื่อเลือกการแสดงผลข้อมูลในส่วน Query
โดยในที่นี่จะมีห้องหมวด 2 Query กดเพิ่มโดยการกด^{ปุ่ม}ช้ายล่าง สามารถถอดอ้างอิงໄวยากรณ์การเขียนได้ที่
flux language syntax

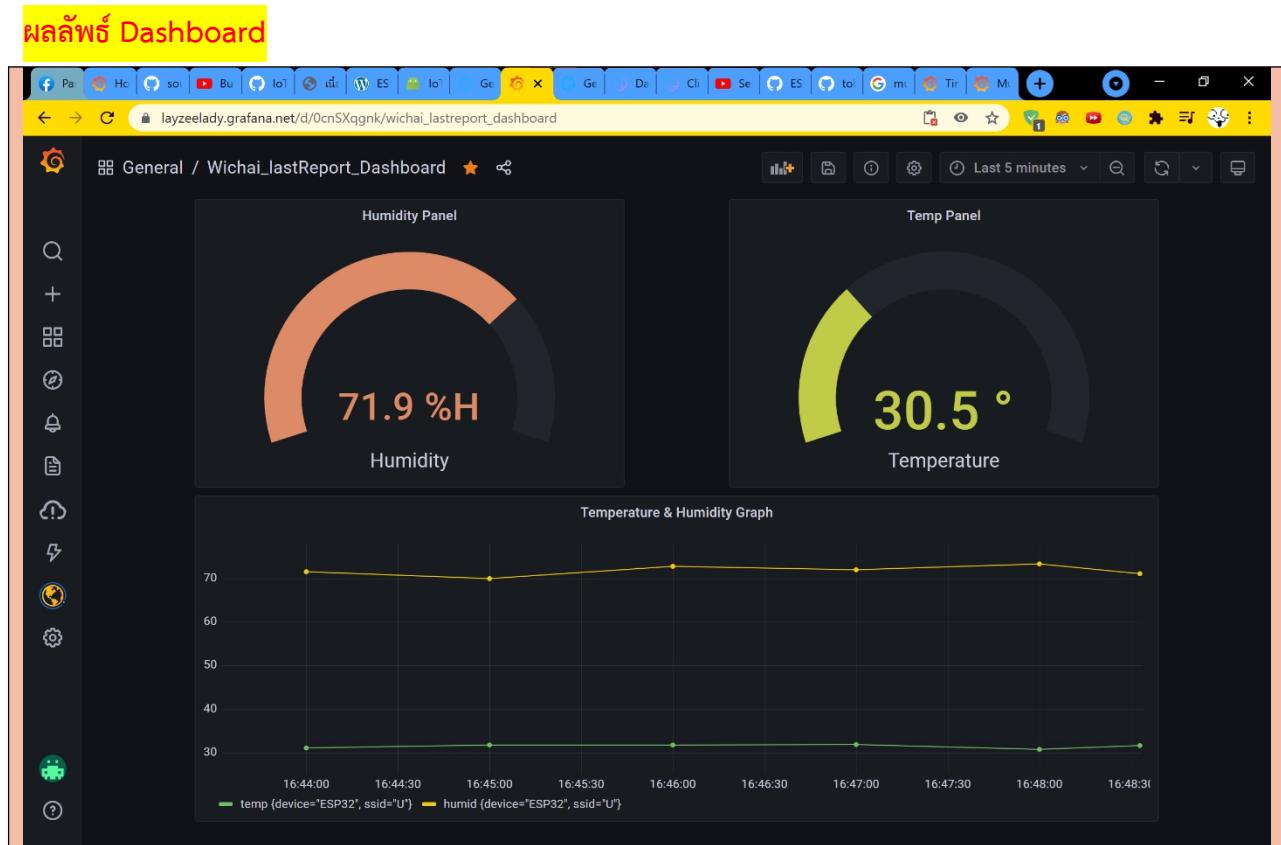
```
//Coding for temperature
from(bucket: "lastreport")
|> range(start: v.timeRangeStart, stop: v.timeRangeStop)
|> filter(fn: (r) => r.device == "ESP32")
|> filter(fn: (r) => r._field == "temp")
|> drop(columns: ["_measurement"])
|> drop(columns: ["_field"])
|> aggregateWindow(every: 60s, fn:mean)

//Coding for humidity
from(bucket: "lastreport")
|> range(start: v.timeRangeStart, stop: v.timeRangeStop)
|> filter(fn: (r) => r.device == "ESP32")
|> filter(fn: (r) => r._field == "humid")
|> drop(columns: ["_measurement"])
|> drop(columns: ["_field"])
|> aggregateWindow(every: 60s, fn:mean)
```

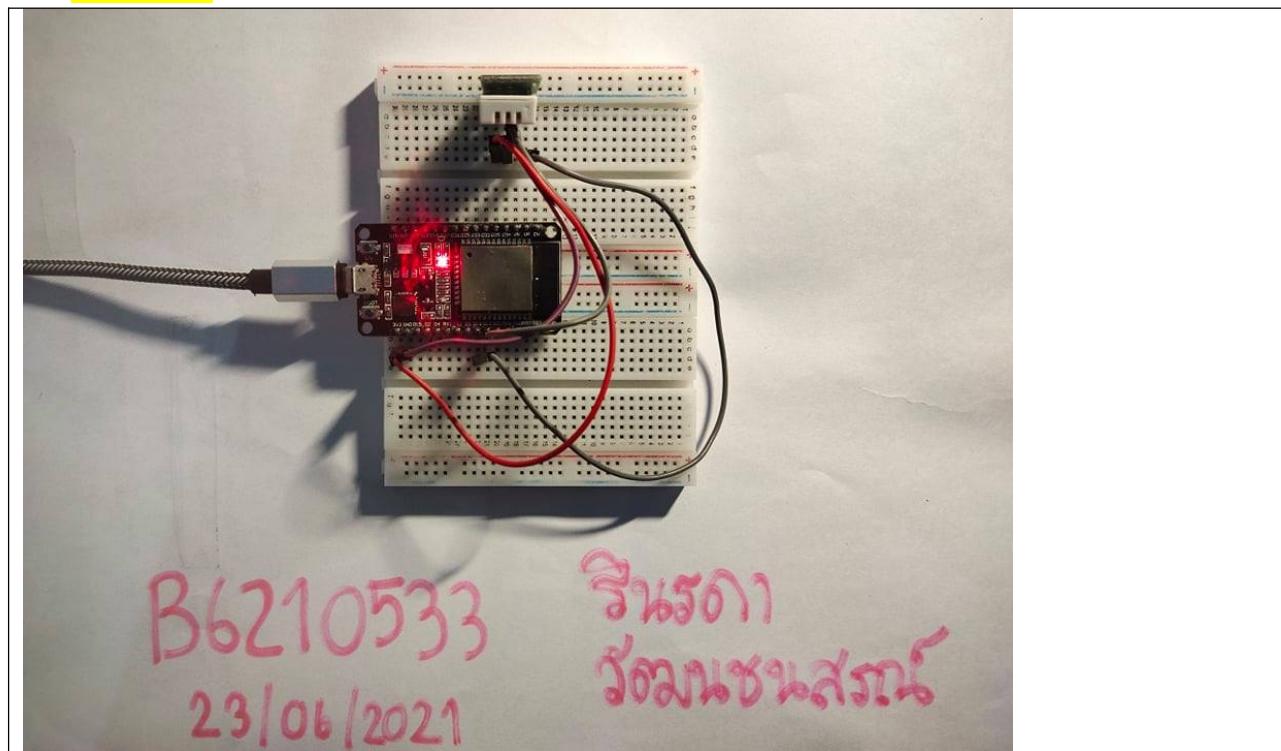
เมื่อทำการเลือกส่วนข้อมูลที่จะแสดงผลแล้ว จะ^{ปุ่ม}ปรากฏผลลัพธ์ดังภาพ ให้กด Apply or Save เพื่อบันทึก

ย้อนกลับมาที่หน้า Dashboard หลัก ทำการเพิ่ม panel อีกสองตัวคือ temp Gauge และ humd Gauge

 <pre data-bbox="182 663 887 671">from(bucket: "lastreport") .> range(start: v.timeRangeStart, stop: v.timeRangeStop) .> filter(fn: (r) => r.device == "ESP32") .> filter(fn: (r) => r.field == "humidity") .> drop(columns: ["_field"]) .> aggregateWindow(every: 5s, fn:mean)</pre>	<p>สามารถเปลี่ยนประเภทของ panel ได้โดยการคลิก drop down เมนูแล้วเลือก panel ชนิดที่ต้องการ ส่วนการโปรแกรมเลือกข้อมูลให้ใช้โค้ดเดียวกันกับข้างต้น ตั้งชื่อ Humidity Gauge</p>
 <pre data-bbox="182 1068 887 1077">from(bucket: "lastreport") .> range(start: v.timeRangeStart, stop: v.timeRangeStop) .> filter(fn: (r) => r.device == "ESP32") .> filter(fn: (r) => r.field == "temp") .> drop(columns: ["_measurement"]) .> drop(columns: ["_field"]) .> aggregateWindow(every: 5s, fn:mean)</pre>	<p>ทำซ้ำกับ Temperature Gauge ปรับแต่งชื่อ หน่วยข้อมูล สีเก่าตามใจชอบ</p>
	<p>จะได้ Dashboard แสดงผล 3 panel ที่สามารถปรับขยาย และเคลื่อนย้าย panel ต่างๆได้</p>
	<p>กราฟจาก time series ของ temp และ humd สามารถกดเลือกแสดงผลเพียงข้อมูลอย่างใดอย่างหนึ่งได้ และสามารถกดคลิกเพื่อแสดงผลข้อมูลเทียบกันได้</p>



7. รูปวงจร



8. อ้างอิง

<https://www.youtube.com/watch?v=Jr3KbuwHPgw>

<https://www.youtube.com/watch?v=N0vHWkigySY>

<https://europe-west1-1.gcp.cloud2.influxdata.com/orgs/dbcc5d1750c7c4c8/load-data/client-libraries/arduino>