

## Week12A - Visualize Sensor data using Grafana and InfluxDB

กลุ่มคำนวณ 2 รหัส B6224196 ชื่อ-สกุล อนันต์ กระเช็น

### 1/4 การทำรายงาน

- เป็นการเก็บคะแนน 10 คะแนนตัดเกรด ให้ทำสุดความสามารถในเวลาที่มี
- ให้รับส่งโดยถือว่าใครส่งก่อนถือว่าเป็นต้นฉบับส่งที่หลังถือว่าลอกเพื่อนมา
- 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

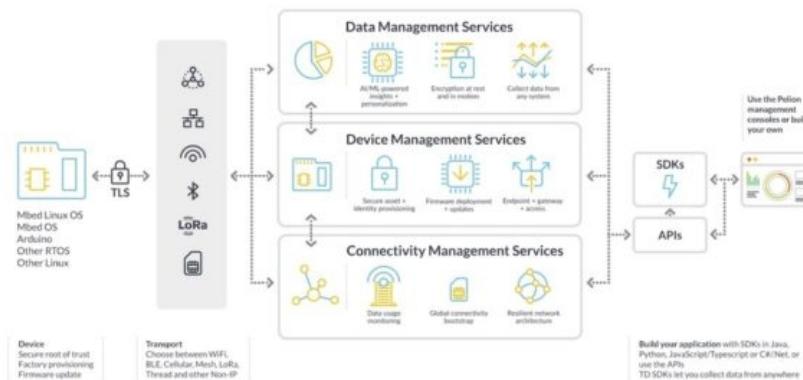
รู้จัก ARM Pelion แพลตฟอร์ม IoT จาก ARM จัดการทุกอย่างครบจบในที่เดียว

October 15, 2019 Applications, ARM, Big Data and Data Science, Business Software, Cloud and Systems, Internet of Things, Products, Software Development & DevOps, Uncategorized

เมื่อวันที่ 30 กรกฎาคมที่ผ่านมา ทาง ARM ได้จัด IoT Workshop ขึ้นเพื่อนำเสนอเรื่องมุ่งต่างๆของการนำเทคโนโลยี IoT ไปใช้ในธุรกิจ ทีมงาน TechTalkThai ได้เข้าไปร่วมงาน และทำความรู้จักกับ ARM Pelion IoT Platform จึงอยากจะมาเล่าให้ผู้อ่านฟัง กันคร่าวๆว่าเจ้าแพลตฟอร์ม IoT นี้มีความสามารถอย่างไร และเหมาะสมกับธุรกิจแบบไหนบ้าง

IoT นั้นเป็นหนึ่งในเทคโนโลยีที่หลายองค์กรยกให้เป็นยุทธศาสตร์ในปี 2019 จากความสามารถในการรวบรวมข้อมูล ซึ่งนับว่าเป็นวัตถุดีในการทำธุรกิจที่ขาดไม่ได้เลยในยุคปัจจุบัน จนถึงตอนนี้ หลายองค์กรอาจเริ่มต้นกับ IoT กันบ้างแล้ว แต่ความท้าทายใหม่ ที่ธุรกิจมักจะเผชิญกับก็คือการจัดการและสเกลระบบ 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 และข้อมูลที่เก็บมาได้ง่ายขึ้น โดยสามารถทำงานร่วมกับอุปกรณ์ระบบเครือข่าย คลาวด์ และข้อมูลได้หลากหลายรูปแบบ อีกทั้งยังมีความปลอดภัย และสามารถช่วยในการนำข้อมูลไปวิเคราะห์ และแสดงผลเบื้องต้นได้ด้วย

รู้จักแพลตฟอร์มนี้ไปคร่าวๆแล้ว ลองมาเจาะลึกกันว่าส่วนประกอบทั้ง 3 ส่วน ลักษณะเด่นๆ คืออะไร และมีจุดเด่นอย่างไรบ้าง

## 1.) Connectivity Management

การเชื่อมต่อในเครือข่าย IoT นั้นมีอยู่หลายรูปแบบ และมีรายละเอียดปลีกย่อยที่ธุรกิจจะต้องจัดการอยู่พอสมควร Pelion จะช่วยให้องค์กรสามารถจัดการการเชื่อมต่อได้อย่างมีประสิทธิภาพ ปลอดภัย และพร้อมต่อการสเกลเครือข่ายขึ้นไปถึงระดับโลก โดย Connectivity Management ของ Pelion มีความสามารถที่น่าสนใจ ดังนี้

### 1.1) Global Cellular

Pelion จะช่วยให้อุปกรณ์ IoT สามารถเชื่อมต่อผ่านเครือข่ายได้ไม่ว่าอุปกรณ์นั้นจะอยู่ที่ใดในโลก ผ่านเวนเดอร์เพียงเจ้าเดียว โดยกลไกของ Pelion จะช่วยเชื่อมต่อสัญญาณจากชิมของอุปกรณ์ไปยังเครือข่ายท้องถิ่นที่ Pelion ได้ทำข้อตกลงไว้ ลดภาระความปวดหัวในการติดต่อกับผู้ให้บริการเครือข่ายในแต่ละประเทศ

Protocol เชื่อมต่อทั้ง IP และ Non-IP

นอกจากส่งข้อมูลผ่าน IP Network แล้ว Pelion ยังรองรับ Non-IP Network เช่น NB-IoT ด้วย โดยโปรโตคอลที่ Pelion รองรับนั้นได้แก่ MQTT(s), HTTPS, และ Sockets ใช้ได้ทั้ง eSIM และชิมแบบปกติ

ธุรกิจสามารถสั่งผลิตอุปกรณ์ที่มีระบบ eSIM ผ่าน ARM ได้ตามต้องการ โดย eSIM ที่ติดมากับอุปกรณ์นั้นจะรองรับการเชื่อมต่อ กับเครือข่ายกว่า 600 เครือข่ายทั่วโลก และหากต้องการเปลี่ยนเครือข่าย ก็สามารถตั้งค่าใหม่ได้ภายในหลัง และในส่วนของชิมแบบปกติของ Pelion ก็ให้บริการซิมการ์ดในทุกขนาด อีกทั้งยังมีแพนที่จะพัฒนาไปจนถึง iSIM ที่มีขนาดเล็กกว่า eSIM มากด้วย

### 1.2) Network Infrastructure

Pelion ได้พัฒนาโครงสร้างพื้นฐานของเครือข่ายให้สามารถทำงานร่วมกับผู้ให้บริการเครือข่ายทั่วโลกได้อย่างมีประสิทธิภาพสูงสุด โดยมีทั้งความเสถียร ยืดหยุ่น และเป็นไปตามกฎข้อบังคับด้านข้อมูลของแต่ละประเทศ ในกรณีใช้ Pelion ผู้ใช้สามารถเลือกได้ว่า จะส่งข้อมูลจากอุปกรณ์ไปยังแอปพลิเคชันผ่านเทคโนโลยีใด เช่น IPSEC, Open VPN, ผู้ให้บริการ Cloud, หรือทางเชื่อมที่ธุรกิจเข้ามาใช้โดยเฉพาะ (Leased Line)

## 2.) Device Management

Device Management ของ Pelion นั้นจะช่วยให้องค์กรสามารถจัดการกับอุปกรณ์และการเชื่อมต่อกับอุปกรณ์ผ่านซอฟต์แวร์ได้โดยสะดวก ไม่ว่าจะเป็นการเขียนซอฟต์แวร์แบบ Embedded หรือว่าการเขียนแอปพลิเคชันด้านบนอย่าง Web App ก็ตาม



(ภาพ: ARM)

โดยภายในโซลูชัน Device Management ก็จะมีโมดูลในการจัดการเรื่องต่างๆให้อย่างครบถ้วน ทั้งแต่เรื่องการอัพเดท Firmware ซึ่งไม่ง่ายเลยหากมีอุปกรณ์ที่หลากหลายและมีจำนวนที่มากในเครือข่าย, Access Management ซึ่งจะช่วยจำกัดการ

เข้าถึงอุปกรณ์และการควบคุมแต่ละส่วน, Connector ซึ่งจัดการการเชื่อมต่อกับอุปกรณ์หลากหลายประเภท การออก Certificate เข้าใช้ระบบ การขึ้นทะเบียนอุปกรณ์แต่ละตัว และการเก็บสถิติ, Device Directory

ซึ่งช่วยในการแบ่งกลุ่ม ค้นหา เรียกดู และเช็คสถานะของอุปกรณ์แต่ละตัว, ไปจนถึงการรักษาความปลอดภัยในการส่งต่อรหัสผ่านเครือข่าย WiFi

โดยทั้งหมดนี้จะเห็นได้ว่าเป็นการจัดการ Lifecycle ของอุปกรณ์ทั้งหมด ตั้งแต่การ Onboard เข้าระบบ ไปจนถึงการใช้งานและบำรุงรักษา

Device Management นั้นสามารถพูดคุยกับอุปกรณ์ IoT ผ่านโปรโตคอลหลากหลาย โดยเฉพาะ LwM2M ซึ่งช่วยให้นักพัฒนาเชื่อมต่อ กับอุปกรณ์ได้ผ่านโนมinalg ที่มีลักษณะคล้ายๆ REST Model และ CoAP ซึ่งจะช่วยประหยัดแบตเตอรี่ของอุปกรณ์ ได้มากกว่า HTTP ราว 8-10 เท่า และในการเชื่อมต่อ กับแอปพลิเคชันซึ่งเป็นปลายทางอีกด้านหนึ่ง Pelion ได้เตรียม REST API และ SDK ในภาษา Java, Python, JavaScript และ .NET ไว้ให้พัฒนาแอปพลิเคชันกันได้โดยง่าย

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

### 3.) Data Management

เป้าประสงค์หลักของการจัดตั้งระบบ IoT นั้นคือการสร้างระบบจัดเก็บข้อมูลที่จะช่วยให้องค์กรสามารถเรียกข้อมูลเหล่านั้น ขึ้นมาวิเคราะห์เป็นความรู้ที่ปรับโยธาต์ต่อธุรกิจได้ แนะนำว่า Pelion ย่อมไม่เลี่มความสำคัญของส่วนนี้ จึงได้พัฒนาระบบจัดการ ข้อมูลคร่าวๆ ที่จะช่วยตั้งแต่การจัดเก็บ นำข้อมูลมาใช้ตัดสินใจแบบ Real-time และรักษาความปลอดภัยและความเป็นส่วนตัว ของข้อมูล โดยมีกลไกการองรับการสเกลเต็มที่ ทำให้องค์กรไม่ต้องกังวลว่าระบบจะทำงานได้แย่ลงหากมีข้อมูลหรืออุปกรณ์ในเครือข่าย IoT เพิ่มมากขึ้นเมื่อเวลาผ่านไป

โซลูชันหลักของส่วนนี้ คือ ARM Treasure Data ซึ่งเป็นซอฟต์แวร์จัดการและวิเคราะห์ข้อมูลที่เชื่อมต่อกับ Pelion ได้ครบครัน ในตัวเดียว โดยมีเครื่องมือต่างๆ พร้อมให้เลือกใช้งาน เช่น ระบบ Predictive Analytics การสร้าง Customer View 360 ของค่า จากข้อมูลการใช้งาน การวิเคราะห์ข้อมูลเพื่อ Cross-sell และ Upsell และการสร้างระบบ Recommendation เป็นต้น ซึ่งโซลูชันนี้หลายอย่างที่กรอกได้นำไปใช้งานเพิ่มประสิทธิภาพให้กับการทำงานในอุตสาหกรรมมากมาย เช่น อุตสาหกรรมค้าปลีก อุตสาหกรรมพลังงาน อุตสาหกรรมการผลิต และอุตสาหกรรมอื่นๆ อีกมาก

แพลตฟอร์ม Pelion นั้นปัจจุบันได้มีการนำไปใช้งานกับระบบ IoT ทั้งในไปรษณีย์และขนาดเล็ก เช่น ระบบ IoT ในการดูแลสัตว์น้ำฝ่ายเซ็นเซอร์รับข้อมูลจากเสียง ระบบตรวจสอบสถานะการทำงานของเครื่องจักรในโรงงาน ระบบจัดการคลังสินค้า และระบบซึ่งเป็นส่วนประกอบของ Smart City เช่น ที่จอดรถอัจฉริยะ และเสาไฟฟ้าที่เปิดปิดตามความเคลื่อนไหวของคน และสามารถควบคุมได้จากระบบส่วนกลาง เป็นต้น

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

## Grafana Dashboard

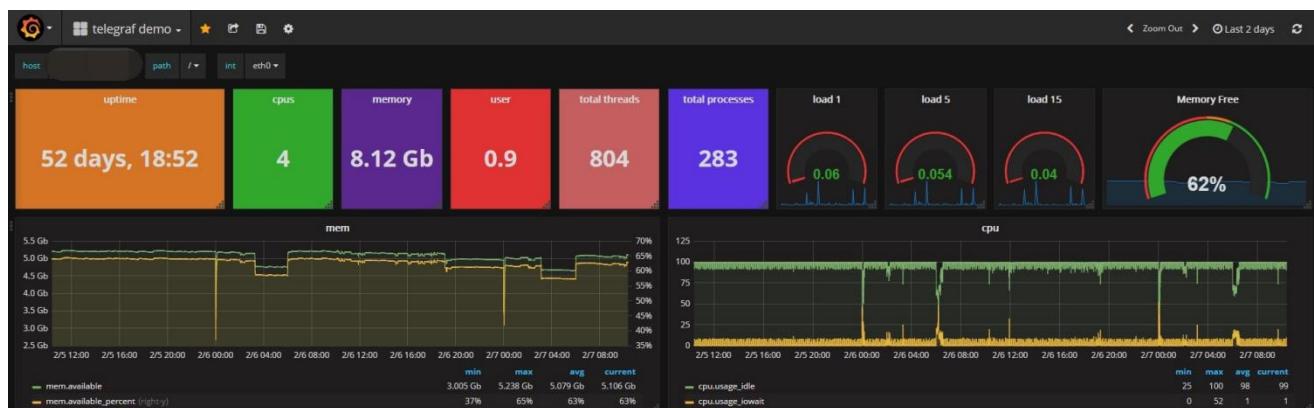
ทำความรู้จักกับ Grafana Dashboard



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

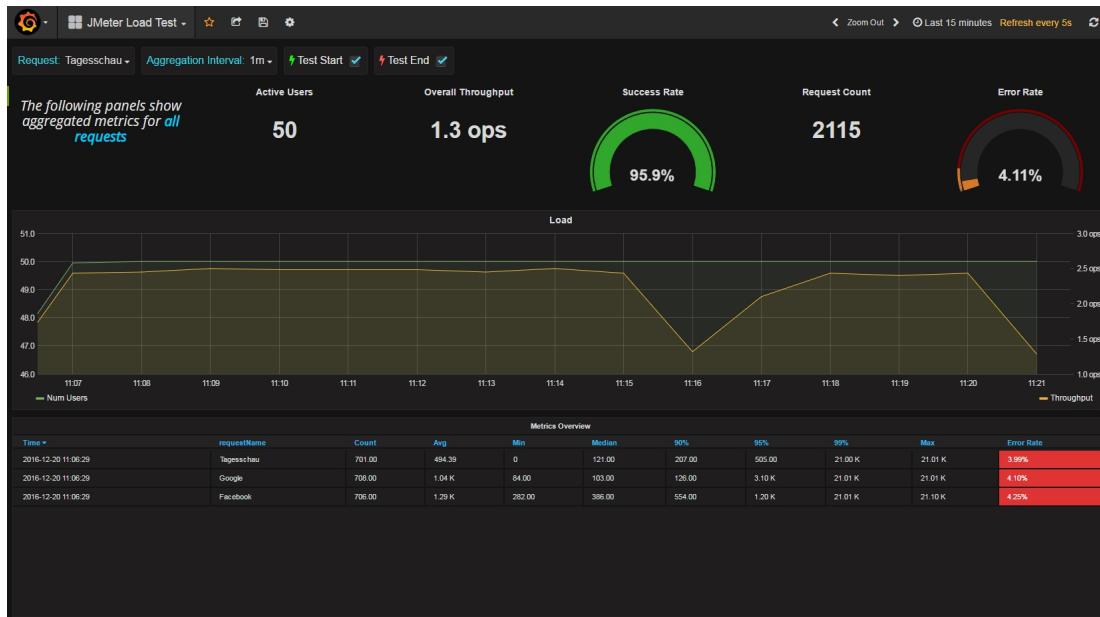
### จุดเด่นของ Grafana

- 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

สำหรับ Windows (x64)

- สามารถใช้ grafana-server.exe เพื่อเริ่มใช้งานได้ทันที
  - กรณีต้องการระบุ custom config ดูรายละเอียด ที่นี่

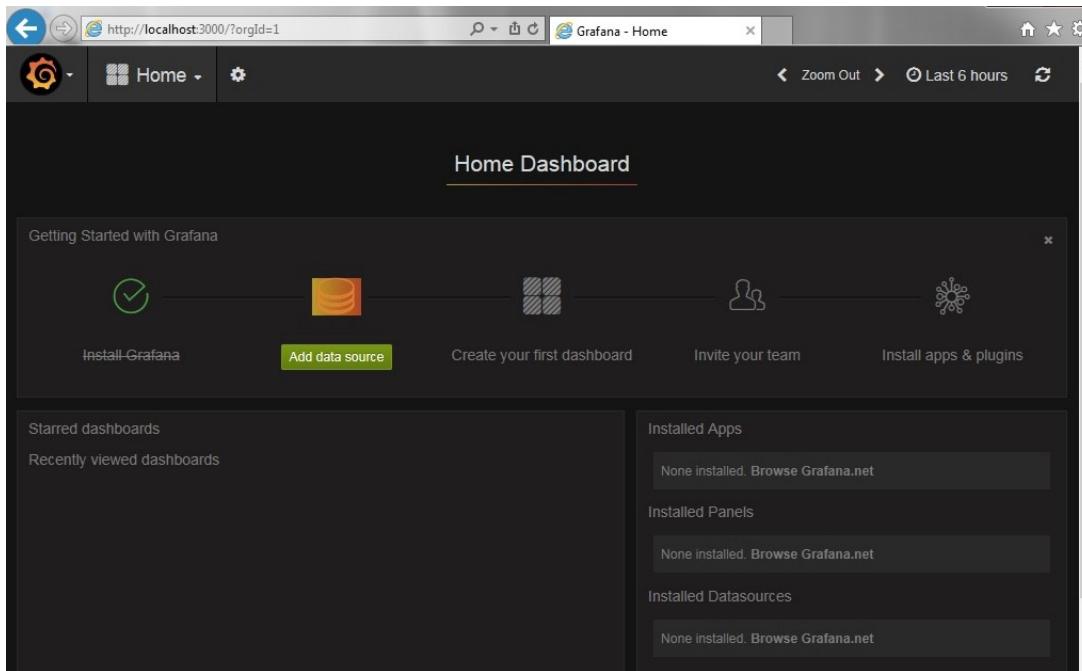
```
C:\Windows\system32\cmd.exe - bin\grafana-server.exe --config config\custom.ini

C:\>grafana-4.4.1\bin\grafana-server.exe --config conf\custom.ini
+ [32]INFO+[0m07-28 20:22:33:40] Starting Grafana +[32]log
+ [32]INFO+[0m07-28 20:22:33:40] Grafana version: 4.4.1 +[32]log
+ [32]INFO+[0m07-28 20:22:33:40] +[32]compiled+[0m=2
017-02-05T14:05+0400
+ [32]INFO+[0m07-28 20:22:33:40] Config loaded from +[32]log
+ [32]INFO+[0m07-28 20:22:33:40] Config loaded from +[32]log
+ [32]INFO+[0m07-28 20:22:33:40] config\custom.ini +[32]log
+ [32]INFO+[0m07-28 20:22:33:40] Path Home +[32]log
+ [32]INFO+[0m07-28 20:22:33:40] Path Data +[32]log
+ [32]INFO+[0m07-28 20:22:33:40] Path Data +[32]log
+ [32]INFO+[0m07-28 20:22:33:40] Path Log +[32]log
+ [32]INFO+[0m07-28 20:22:33:40] Path Plugins +[32]log
+ [32]INFO+[0m07-28 20:22:33:40] Initializing DB +[32]log
+ [32]INFO+[0m07-28 20:22:33:40] +[32]dbtype+[0m.sqlite3 +[32]log
+ [32]INFO+[0m07-28 20:22:33:40] Starting DB migration +[32]log
+ [32]INFO+[0m07-28 20:22:33:40] Executing migration +[32]log
+ [32]INFO+[0m07-28 20:22:33:40] +[32]mid+[0m"copy data account to org" +[32]log
+ [32]INFO+[0m07-28 20:22:33:40] Skipping migration condition not fulfilled +[32]log
+ [32]INFO+[0m07-28 20:22:33:40] +[32]mid+[0m"copy data account to org" +[32]log
+ [32]INFO+[0m07-28 20:22:33:40] Executing migration +[32]log
+ [32]INFO+[0m07-28 20:22:33:40] +[32]mid+[0m"copy data account_user to org_user" +[32]log
+ [32]INFO+[0m07-28 20:22:33:40] Skipping migration condition not fulfilled +[32]log
+ [32]INFO+[0m07-28 20:22:33:40] Starting plugin search +[32]log
+ [32]INFO+[0m07-28 20:22:33:41] Initializing Alerting +[32]log
+ [32]INFO+[0m07-28 20:22:33:41] Initializing CleanUpService +[32]log
+ [32]INFO+[0m07-28 20:22:33:41] Initializing Stream Manager +[32]log
+ [32]INFO+[0m07-28 20:22:33:41] Initializing HTTP Server +[32]log
+ [32]INFO+[0m07-28 20:22:33:41] http_server +[32]address+[0m=0.0.0.0:3000]+[32]protocol+[0m=http +[32]sub
Url+[0m +[32]socket+[0m=
```

## สำหรับ Mac (Via Homebrew)

```
:~ 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 .

#### 4/4. การทดสอบ

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

#### Visualize Sensor data using Grafana and InfluxDB

- <https://www.influxdata.com/> >> login to InfluxDB Cloud2.0 >> Continue with Google account

The screenshot shows the InfluxDB Cloud 2.0 homepage. At the top right, there are three login options: 'Login to InfluxDB Cloud 2.0' (highlighted in a red box), 'Login to InfluxDB Enterprise', and 'Login to InfluxDB Cloud 1.x'. Below these options is a large purple banner with the text 'Act in Time. Build on InfluxDB.' and 'The platform for building and operating time series applications.' A 'Start Building' button is located at the bottom left of the banner. The main content area features a 3D illustration of people working with data and servers.

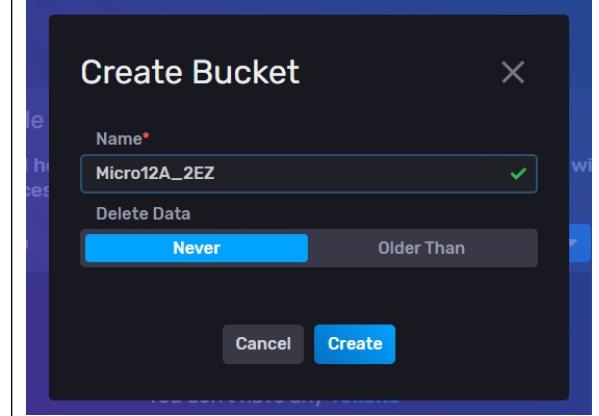
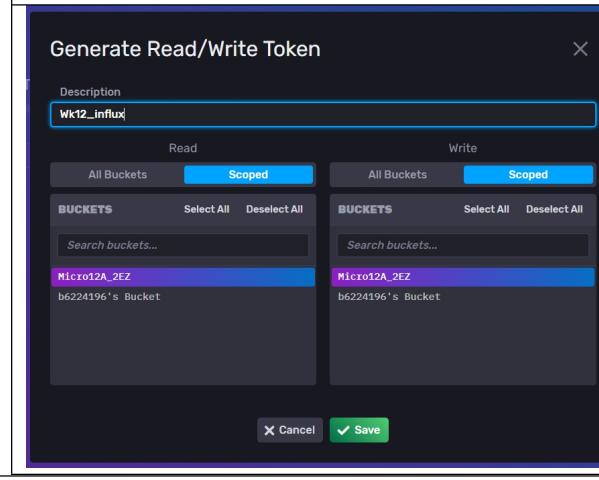
The screenshot shows the 'Create your Free InfluxDB Cloud Account' sign-up page. It includes a 'No credit card required' note. The 'Continue with' section features a 'Google' button (highlighted in a red box) and a 'Microsoft' button. Below this are 'Log In' and 'Sign Up' buttons. There are fields for 'Work Email Address\*' and 'Password\*', along with a 'Forgot Password' link and a 'Log In' button at the bottom.

## 2. Data >> Arduino

The screenshot shows the InfluxDB UI with the 'Data' section selected. The 'Load Data' screen is displayed, featuring a search bar and sections for 'File Upload' and 'Client Libraries'. Under 'Client Libraries', there is a grid of icons for various programming languages. The 'Arduino' icon, which includes the Arduino logo and the word 'ARDUINO' below it, is highlighted with a red box.

## 3. Create Bucket and Token key

The screenshot shows the InfluxDB UI with the 'Client Library' section selected. The 'Sources | Load Data' screen for the Arduino client is displayed. It features a large Arduino logo and sections for 'Code Sample Options', 'Install Library', and 'Manual Installation'. Two buttons are highlighted with red boxes: '+ Generate Token' and '+ Create Bucket'. The '+ Create Bucket' button is specifically highlighted.

	<p>Create Bucket &gt;&gt; Name “Micro12A_2EZ”  &gt;&gt; Create</p>
	<p>Generate Token &gt;&gt; Read/Write Token  &gt;&gt; Description “Wk12_influx” &gt;&gt; Save</p>
<p>4. Copy Initialize the Client to Arduino</p> <p>Initialize the Client</p> <pre>#if defined(ESP32) #include &lt;WiFiMulti.h&gt; WiFiMulti wifiMulti; #define DEVICE "ESP32" #elif defined(ESP8266) #include &lt;ESP8266WiFiMulti.h&gt; ESP8266WiFiMulti wifiMulti; #define DEVICE "ESP8266" #endif  #include &lt;InfluxDbClient.h&gt; #include &lt;InfluxDbCloud.h&gt;  // WiFi AP SSID #define WIFI_SSID "SSID" // WiFi password #define WIFI_PASSWORD "PASSWORD" // InfluxDB V2 server url, e.g. https://eu-central-1-1.aws.cloud2.influxdata.com (Use: InfluxDB UI -&gt; Load Data -&gt; Client Libraries) #define INFLUXDB_URL "https://europe-west1-1.gcp.cloud2.influxdata.com" // InfluxDB v2 server or cloud API authentication token (Use: InfluxDB UI -&gt; Data -&gt; Tokens -&gt; &lt;select token&gt;) </pre> <p><a href="#">Copy to Clipboard</a></p>	

## 5. Copy Write Data to Arduino

### Write Data

```
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("rss", WiFi.RSSI());

    // Print what are we 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());
    }
}
```

[Copy to Clipboard](#)

## 6. Edit Code

```
#if defined(ESP32)
#include <WiFiMulti.h>

WiFiMulti wifiMulti;
#define DEVICE "ESP32"

#elif defined(ESP8266)
#include <ESP8266WiFiMulti.h>

ESP8266WiFiMulti wifiMulti;
#define DEVICE "ESP8266"
#endif

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

// WiFi AP SSID
#define WIFI_SSID "lalala"
// WiFi password
#define WIFI_PASSWORD "anan110900"

// InfluxDB v2 server url, e.g. https://eu-central-1-1.aws.cloud2.influxdata.com (Use: InfluxDB UI -> Load Data -> Client Libraries)
#define INFLUXDB_URL "https://europe-west1-1.gcp.cloud2.influxdata.com"
// InfluxDB v2 server or cloud API authentication token (Use: InfluxDB UI -> Data -> Tokens -> <select token>)
#define INFLUXDB_TOKEN
"RwzdtZFwyTSiG1W5yL3hKqZAEBh4AAmEzU3QYyqykDduFRFzaOjIV_9_DNKs9SVzPU25eigmnYCEszGWjcTbsw=="
// InfluxDB v2 organization id (Use: InfluxDB UI -> User -> About -> Common Ids )
```

```

#define INFLUXDB_ORG "b6224196@g.sut.ac.th"
// InfluxDB v2 bucket name (Use: InfluxDB UI -> Data -> Buckets)
#define INFLUXDB_BUCKET "Micro12A_2EZ"

// Set timezone string according to https://www.gnu.org/software/libc/manual/html_node/TZ-Variable.html
// Examples:
// Pacific Time: "PST8PDT"
// Eastern: "EST5EDT"
// Japanesse: "JST-9"
// Central Europe: "CET-1CEST,M3.5.0,M10.5.0/3"
#define TZ_INFO "CET-1CEST,M3.5.0,M10.5.0/3"

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

// Data point
Point sensor("wifi_status");

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

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

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

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

    // 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");

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

    float temp = random(0,6500)*0.01 ;
    float humid = random(2500,9000)*0.01 ;
    // Store measured value into point
    sensor.addField("temperature",temp);
    sensor.addField("humidity",humid);

    // Print what are we 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());
    }
}

```

```
//Wait 10s
Serial.println("Wait 10s");
delay(10000);
}
```

COM5

```
14:28:52.717 ->
14:28:56.309 -> Connected to InfluxDB: https://europe-west1-1.gcp.cloud2.influxdata.com
14:28:56.309 -> Writing: wifi_status,device=ESP32,SSID=lalala temperature=26.67,humidity=67.43
14:28:59.136 -> InfluxDB write failed: connection lost
14:28:59.136 -> Wait 10s
14:29:09.145 -> Writing: wifi_status,device=ESP32,SSID=lalala temperature=15.12,humidity=88.58
14:29:25.344 -> Wait 10s
14:29:35.319 -> Writing: wifi_status,device=ESP32,SSID=lalala temperature=37.39,humidity=86.49
14:29:38.072 -> Wait 10s
14:29:48.057 -> Writing: wifi_status,device=ESP32,SSID=lalala temperature=64.23,humidity=51.73
14:29:51.542 -> Wait 10s
14:30:01.523 -> Writing: wifi_status,device=ESP32,SSID=lalala temperature=11.66,humidity=44.94
14:30:04.643 -> Wait 10s
14:30:14.630 -> Writing: wifi_status,device=ESP32,SSID=lalala temperature=63.24,humidity=67.76
14:30:17.624 -> Wait 10s
14:30:27.594 -> Writing: wifi_status,device=ESP32,SSID=lalala temperature=48.78,humidity=30.75
14:30:31.091 -> Wait 10s
14:30:41.116 -> Writing: wifi_status,device=ESP32,SSID=lalala temperature=29.76,humidity=62.80
14:30:44.239 -> Wait 10s
14:30:54.244 -> Writing: wifi_status,device=ESP32,SSID=lalala temperature=14.18,humidity=27.49
14:30:57.122 -> Wait 10s
14:31:07.118 -> Writing: wifi_status,device=ESP32,SSID=lalala temperature=25.21,humidity=48.36
14:31:10.575 -> Wait 10s
14:31:20.594 -> Writing: wifi_status,device=ESP32,SSID=lalala temperature=61.71,humidity=82.61
14:31:23.525 -> Wait 10s
14:31:33.517 -> Writing: wifi_status,device=ESP32,SSID=lalala temperature=62.17,humidity=89.45
14:31:36.229 -> Wait 10s
14:31:46.214 -> Writing: wifi_status,device=ESP32,SSID=lalala temperature=25.61,humidity=34.55
14:31:49.715 -> Wait 10s
14:31:59.709 -> Writing: wifi_status,device=ESP32,SSID=lalala temperature=9.55,humidity=26.11
14:32:02.849 -> Wait 10s
14:32:12.844 -> Writing: wifi_status,device=ESP32,SSID=lalala temperature=64.63,humidity=77.98
14:32:16.569 -> Wait 10s
14:32:26.587 -> Writing: wifi_status,device=ESP32,SSID=lalala temperature=50.47,humidity=52.61
14:32:29.478 -> Wait 10s
14:32:39.459 -> Writing: wifi_status,device=ESP32,SSID=lalala temperature=59.03,humidity=73.98
14:32:42.577 -> Wait 10s
14:32:52.554 -> Writing: wifi_status,device=ESP32,SSID=lalala temperature=53.63,humidity=44.37
14:32:55.487 -> Wait 10s
14:33:05.478 -> Writing: wifi_status,device=ESP32,SSID=lalala temperature=51.65,humidity=72.79
14:33:08.606 -> Wait 10s
```

<  Autoscroll  Show timestamp >

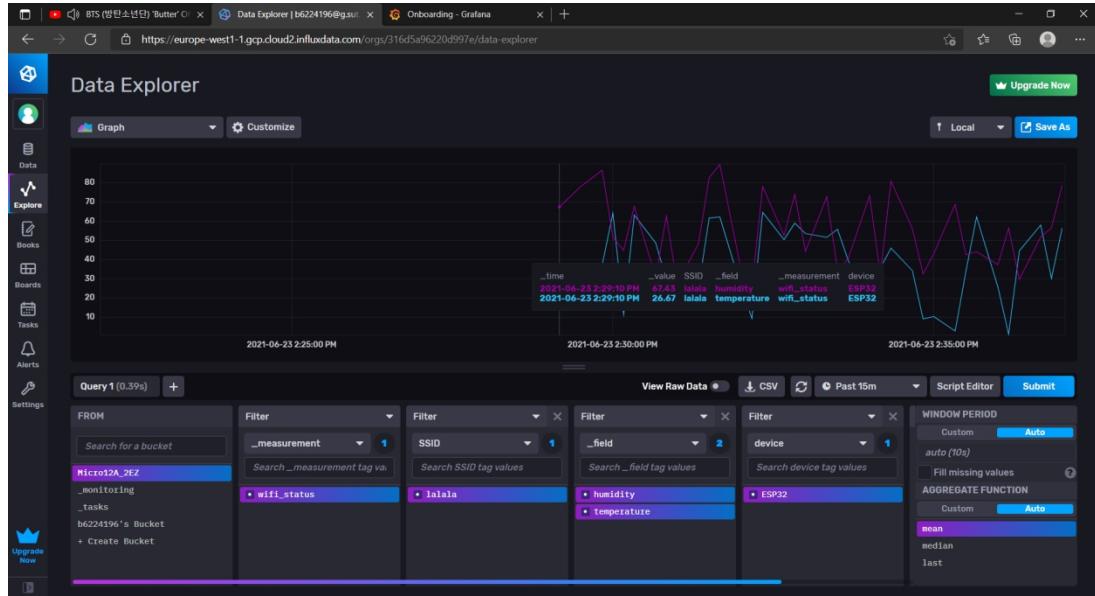
Newline 115200 baud Clear output

B622 4116 05/06/2021

23 / 6 / 64

## 7. Influx >> Explore >> From “Micro12A \_2EZ”

Past >> Past 15m



### View Raw Data

- Temperature

_start	_stop	_time	_value	SSID	_field
2021-06-23T07:22:04.015Z	2021-06-23T07:37:04.015Z	2021-06-23T07:29:10.000Z	26.67	lalala	temperature
2021-06-23T07:22:04.015Z	2021-06-23T07:37:04.015Z	2021-06-23T07:29:30.000Z	29.895	lalala	temperature
2021-06-23T07:22:04.015Z	2021-06-23T07:37:04.015Z	2021-06-23T07:29:50.000Z	37.39	lalala	temperature
2021-06-23T07:22:04.015Z	2021-06-23T07:37:04.015Z	2021-06-23T07:30:00.000Z	64.23	lalala	temperature
2021-06-23T07:22:04.015Z	2021-06-23T07:37:04.015Z	2021-06-23T07:30:10.000Z	11.66	lalala	temperature
2021-06-23T07:22:04.015Z	2021-06-23T07:37:04.015Z	2021-06-23T07:30:20.000Z	63.24	lalala	temperature
2021-06-23T07:22:04.015Z	2021-06-23T07:37:04.015Z	2021-06-23T07:30:40.000Z	48.78	lalala	temperature
2021-06-23T07:22:04.015Z	2021-06-23T07:37:04.015Z	2021-06-23T07:30:50.000Z	29.76	lalala	temperature
2021-06-23T07:22:04.015Z	2021-06-23T07:37:04.015Z	2021-06-23T07:31:00.000Z	14.18	lalala	temperature
2021-06-23T07:22:04.015Z	2021-06-23T07:37:04.015Z	2021-06-23T07:31:20.000Z	26.21	lalala	temperature
2021-06-23T07:22:04.015Z	2021-06-23T07:37:04.015Z	2021-06-23T07:31:30.000Z	61.71	lalala	temperature
2021-06-23T07:22:04.015Z	2021-06-23T07:37:04.015Z	2021-06-23T07:31:40.000Z	62.37	lalala	temperature
2021-06-23T07:22:04.015Z	2021-06-23T07:37:04.015Z	2021-06-23T07:32:00.000Z	25.61	lalala	temperature
2021-06-23T07:22:04.015Z	2021-06-23T07:37:04.015Z	2021-06-23T07:32:10.000Z	9.55	lalala	temperature
2021-06-23T07:22:04.015Z	2021-06-23T07:37:04.015Z	2021-06-23T07:32:20.000Z	64.63	lalala	temperature
2021-06-23T07:22:04.015Z	2021-06-23T07:37:04.015Z	2021-06-23T07:32:40.000Z	50.47	lalala	temperature
2021-06-23T07:22:04.015Z	2021-06-23T07:37:04.015Z	2021-06-23T07:32:50.000Z	59.83	lalala	temperature
2021-06-23T07:22:04.015Z	2021-06-23T07:37:04.015Z	2021-06-23T07:33:00.000Z	53.63	lalala	temperature
2021-06-23T07:22:04.015Z	2021-06-23T07:37:04.015Z	2021-06-23T07:33:20.000Z	51.65	lalala	temperature

### - Humidity

	_start	_stop	_time	_value	SSID	_field
2921-06-23T07:22:04.015Z	2921-06-23T07:29:04.015Z	2021-06-23T07:29:10.000Z	67.43	lalala	humidity	
2921-06-23T07:22:04.015Z	2921-06-23T07:29:30.000Z	2021-06-23T07:29:38.000Z	78.006	lalala	humidity	
2921-06-23T07:22:04.015Z	2921-06-23T07:29:50.000Z	2021-06-23T07:30:08.000Z	66.49	lalala	humidity	
2921-06-23T07:22:04.015Z	2921-06-23T07:30:10.000Z	2021-06-23T07:30:18.000Z	51.73	lalala	humidity	
2921-06-23T07:22:04.015Z	2921-06-23T07:30:18.000Z	2021-06-23T07:30:26.000Z	44.94	lalala	humidity	
2921-06-23T07:22:04.015Z	2921-06-23T07:30:28.000Z	2021-06-23T07:30:29.000Z	67.76	lalala	humidity	
2921-06-23T07:22:04.015Z	2921-06-23T07:30:40.000Z	2021-06-23T07:30:48.000Z	30.75	lalala	humidity	
2921-06-23T07:22:04.015Z	2921-06-23T07:30:58.000Z	2021-06-23T07:31:06.000Z	62.8	lalala	humidity	
2921-06-23T07:22:04.015Z	2921-06-23T07:31:06.000Z	2021-06-23T07:31:14.000Z	27.49	lalala	humidity	
2921-06-23T07:22:04.015Z	2921-06-23T07:31:20.000Z	2021-06-23T07:31:28.000Z	48.36	lalala	humidity	
2921-06-23T07:22:04.015Z	2921-06-23T07:31:30.000Z	2021-06-23T07:31:39.000Z	82.61	lalala	humidity	
2921-06-23T07:22:04.015Z	2921-06-23T07:31:40.000Z	2021-06-23T07:31:48.000Z	89.45	lalala	humidity	
2921-06-23T07:22:04.015Z	2921-06-23T07:32:00.000Z	2021-06-23T07:32:08.000Z	34.55	lalala	humidity	
2921-06-23T07:22:04.015Z	2921-06-23T07:32:10.000Z	2021-06-23T07:32:18.000Z	26.11	lalala	humidity	
2921-06-23T07:22:04.015Z	2921-06-23T07:32:20.000Z	2021-06-23T07:32:28.000Z	77.98	lalala	humidity	
2921-06-23T07:22:04.015Z	2921-06-23T07:32:40.000Z	2021-06-23T07:32:48.000Z	52.61	lalala	humidity	
2921-06-23T07:22:04.015Z	2921-06-23T07:32:50.000Z	2021-06-23T07:32:58.000Z	73.98	lalala	humidity	
2921-06-23T07:22:04.015Z	2921-06-23T07:33:00.000Z	2021-06-23T07:33:08.000Z	44.37	lalala	humidity	
2921-06-23T07:22:04.015Z	2921-06-23T07:33:20.000Z	2021-06-23T07:33:28.000Z	72.79	lalala	humidity	

### 8. <https://grafana.com/> >> login >> Log in with Google Account

Your new (actually useful) free plan is here  
Get 10,000 series for Prometheus or Graphite metrics, 50 GB of logs, 50 GB of traces, and much more with Grafana Cloud.  
[Create account](#) → [Learn more](#) →

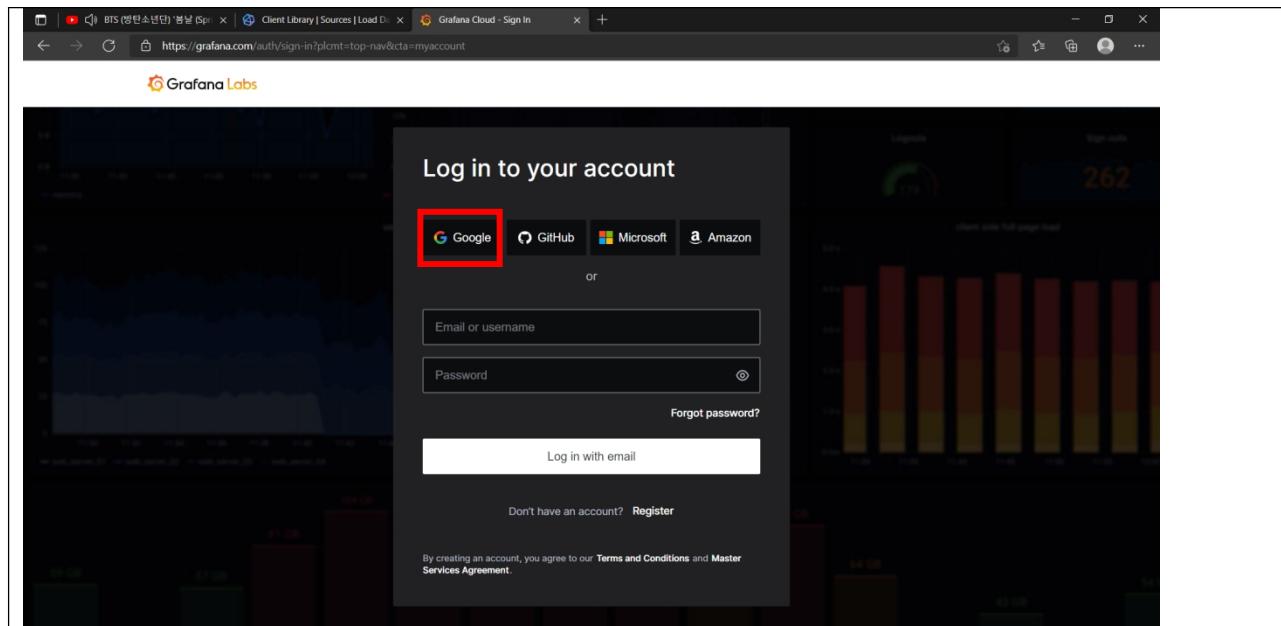
Your observability wherever you need it  
From dashboards to metrics, logs and traces. Choose what works best for you and your team.

Robust free tier      Self-managed  
[Grafana Cloud sign up](#) →      [Grafana Enterprise Stack](#) →

Complete your observability picture  
Enterprise features & plugins like Snowflake, ServiceNow, Splunk, MongoDB, Datadog, Dynatrace and more  
[Enterprise plugins](#) → [Enterprise features](#) →

Success stories →

<https://grafana.com/auth/sign-in?picmt=top-nav&cta=myaccount>



## 9. Configuration >> Data Source >> Add Data Source >> InfluxDB

The screenshot shows the 'Configuration' section of Grafana. On the left, there's a sidebar with icons for Configuration, Data sources, Users, Teams, Plugins, Preferences, and API keys. The main area is titled 'Data sources' and shows three existing data sources: 'grafanacloud-b6224196-alertmanager', 'grafanacloud-b6224196-graphite', and 'grafanacloud-b6224196-logs'. Below this, under 'Configuration', there are sections for 'anacloud-b6224196-prom' (with Prometheus), 'anacloud-b6224196-ruler' (with Prometheus), and 'anacloud-b6224196-ruler-logs' (with Loki). At the top right of the 'Data sources' list, there is a blue 'Add data source' button, which is highlighted with a red box.

The screenshot shows the 'Add data source' interface in Grafana. Under the 'Time series databases' section, the 'InfluxDB' option is highlighted with a blue border. To its right is a 'Select' button. A sidebar on the left contains various icons for different data sources.

## 10. Setting

The screenshot shows the 'InfluxDB' configuration page in Grafana. It includes fields for 'Name' (set to 'InfluxDB'), 'Query Language' (set to 'Flux'), 'HTTP' settings (URL: 'https://europe-west1-1.gcp.cloud2.influxdata.com'), and 'Auth' settings (Basic auth enabled). To the right, there is a note about Flux support being beta, followed by the text 'Query Language >> Flux' and 'URL >> https://europe-west1-1.gcp.cloud2.influxdata.com/'.

## 11. Find Token >> Influx >> Data >> Token

The screenshot shows the 'Tokens' page in Grafana. It lists a single token named 'Wk12\_influx'. The token details show it was created at '2021-06-23 14:06:50' and has the status 'Active'. A red arrow points to the 'Active' status indicator.

The screenshot shows a modal window titled "Wk12\_influx". Inside, there is a text input field containing a long token string: "RwzdtZFwyTSiG1W5yL3hKqZAEbh4AAmEzU3QYyqykDduFRFzaOjiV\_9\_DNKs9SVzPU25eigmnYCEszGwjcTbsw==". This field is highlighted with a red rectangle. Below it is a "Copy to Clipboard" button. The background of the modal shows "buckets-Micro12A\_2EZ" with "read" and "write" permissions listed.

## 12. Gafana >> Save and Test

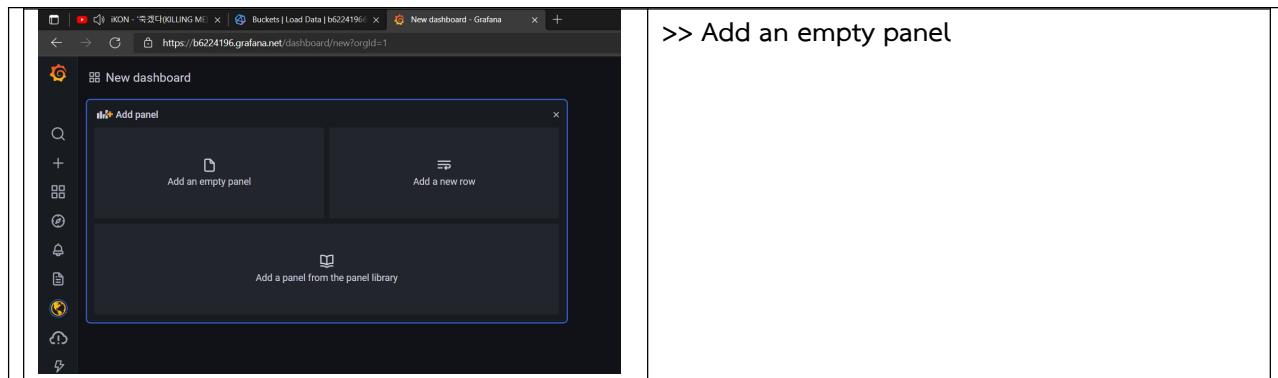
The screenshot shows the "InfluxDB Details" section of the Gafana configuration. It includes fields for Organization (set to "b6224196@g.sut.ac.th"), Token (redacted), Default Bucket ("Micro12A\_2EZ"), Min time interval ("10s"), and Max series ("1000"). To the right is a sidebar menu with a user icon, the email "b6224196@g.sut.ac.th", and links for Usage, Billing, Users, About, and Logout. A red arrow points from the "Organization" field in the main form to the user icon in the sidebar.

- If Gafana can Connect to Influx

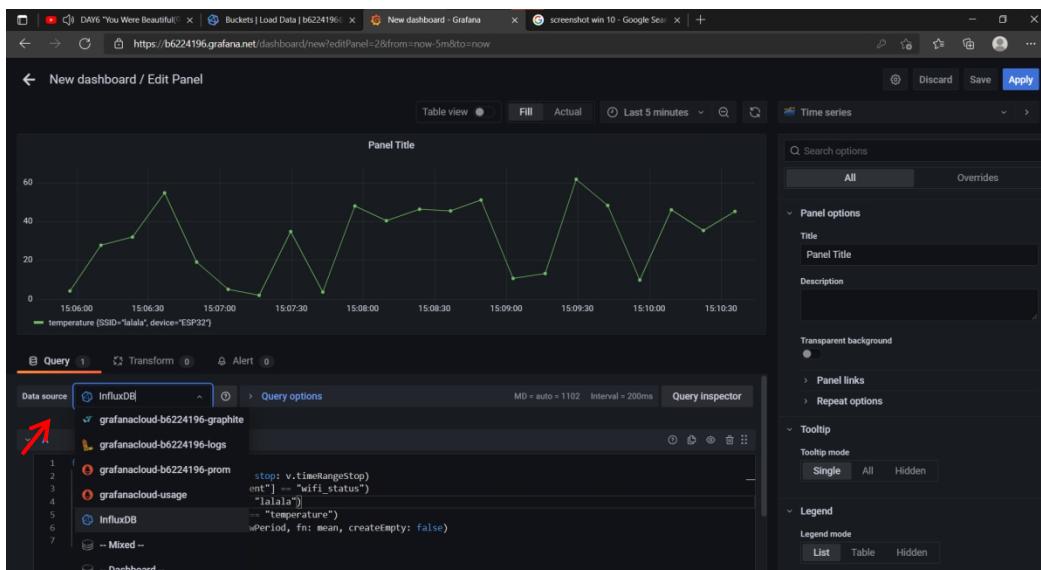
The screenshot shows a confirmation message: "1 buckets found". Below it are three buttons: "Back", "Delete", and "Save & test".

## 13. Gafana

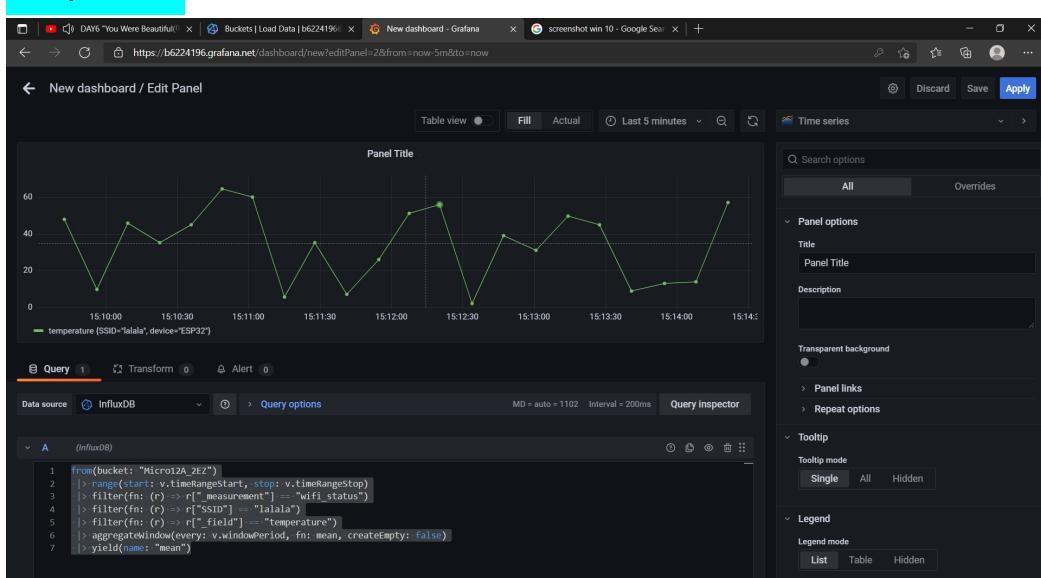
The screenshot shows the "Data Sources" section of the Gafana interface. On the left, there's a sidebar with "Create", "Dashboard", "Folder", "Import", and "Flux". The main area shows a "Data Source" card for "Type: InfluxDB". On the right, a large white box contains the text "Create >> Dashboard".



#### 14. Data source >> InfluxDB



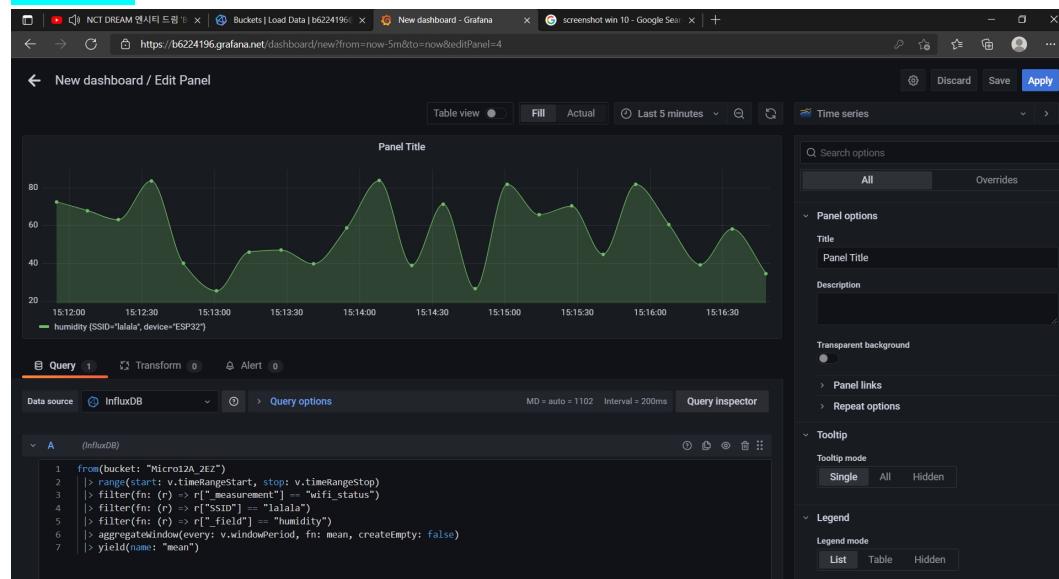
#### Temperature



### Code Temperature :

```
from(bucket: "Micro12A_2EZ")
|> range(start: v.timeRangeStart, stop: v.timeRangeStop)
|> filter(fn: (r) => r["_measurement"] == "wifi_status")
|> filter(fn: (r) => r["SSID"] == "lalala")
|> filter(fn: (r) => r["_field"] == "temperature")
|> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)
|> yield(name: "mean")
```

### Humidity



### Code Humidity :

```
from(bucket: "Micro12A_2EZ")
|> range(start: v.timeRangeStart, stop: v.timeRangeStop)
|> filter(fn: (r) => r["_measurement"] == "wifi_status")
|> filter(fn: (r) => r["SSID"] == "lalala")
|> filter(fn: (r) => r["_field"] == "humidity")
|> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)
|> yield(name: "mean")
```

## 15. Dashboard

