

## Table of Contents

Table of Contents .....	1
Internet of Things (IoT) .....	3
1. Windows Subsystem for Linux 2 (WSL2) .....	4
1.1 Enable the Windows Subsystem for Linux .....	4
1.2 Check requirements for running WSL 2 .....	4
1.3 Enable Virtual Machine Feature .....	5
1.4 Download the Linux kernel update package .....	5
1.5 Set WSL 2 as your default version .....	5
1.6 Install your Linux distribution of choice .....	6
1.6.1 Open the Microsoft Store ( <a href="https://aka.ms/wslstore">https://aka.ms/wslstore</a> ) and select your favorite Linux distribution .....	6
1.6.2 From the distribution's page, select "Get" .....	6
2. Docker .....	7
2.1 Install and Use Docker on Ubuntu .....	7
2.1.1 Installing Docker .....	7
2.1.2 Executing the Docker Command Without Sudo (Optional) .....	9
2.2 How to use Docker and Docker compose to Start service (php, Apache, Node-Red, Influxdb, MySQL, PHPMysqladmin, Adminer, Grafana and MQTT) .....	9
2.2.1 การ Clone Project จาก GitHub เพื่อใช้งาน .....	9
2.2.2 การตั้งค่า Username และ Password ให้กับ Service .....	10

3. How to use .....	12
3.1 Node-Red.....	12
วิธีการใช้งาน Node-Red:.....	14

## Internet of Things (IoT)

IoT ย่อมาจาก Internet of Things (อินเทอร์เน็ตของสิ่งของ) ซึ่งเป็นแนวคิดที่เกี่ยวข้องกับการเชื่อมต่ออุปกรณ์ต่าง ๆ กันทางอินเทอร์เน็ต เพื่อให้สามารถรับส่งข้อมูลและทำงานร่วมกันได้อย่างอัตโนมัติ และมีประสิทธิภาพมากขึ้น

ในระบบ IoT อุปกรณ์ต่าง ๆ เช่น เซ็นเซอร์ เครื่องมือวัด หรือ อุปกรณ์ไฟฟ้า เชื่อมต่อกับเครือข่ายอินเทอร์เน็ตและสื่อสารกันได้ ซึ่งทำให้สามารถเก็บรวบรวมข้อมูลจากอุปกรณ์เหล่านั้นและใช้ข้อมูลเหล่านั้นในการวิเคราะห์ ประมวลผล หรือใช้ในการควบคุมอุปกรณ์ต่าง ๆ ได้อย่างสะดวกและรวดเร็ว

ระบบ IoT มีการนำเอาเทคโนโลยีเครือข่าย การสื่อสารไร้สาย การรวมระบบคอมพิวเตอร์ การวิเคราะห์ข้อมูลและปัญญาประดิษฐ์ เพื่อให้เกิดความสามารถในการติดต่อสื่อสารและการทำงานอัตโนมัติระหว่างอุปกรณ์ต่าง ๆ

ตัวอย่างการนำ IoT มาใช้ประโยชน์ได้แก่ การติดตามและจัดการการส่งสินค้าในธุรกิจขนส่ง การควบคุมและจัดการอุปกรณ์ในระบบอัตโนมัติในอาคารสำนักงานหรือบ้าน หรือ ระบบควบคุมแสงสว่างและอุณหภูมิในสวนพักผ่อนอัตโนมัติ เป็นต้น



## 1. Windows Subsystem for Linux 2 (WSL2)

WSL2 เป็นเทคโนโลยีที่ออกแบบมาเพื่อเป็นช่วงส่วนกลางที่ทำให้สามารถรันระบบปฏิบัติการ Linux บนระบบปฏิบัติการ Windows ได้

WSL2 เป็นการพัฒนาของ Microsoft ที่มีเป้าหมายในการเพิ่มประสิทธิภาพและความเข้ากันได้ระดับสูงของ WSL ซึ่งเป็นเวอร์ชันก่อนหน้านี้ โดย WSL2 ใช้เทคโนโลยีการจำลองสถาปัตยกรรมหนึ่งเพื่อสร้างเคอร์เนล Linux แยกออกมาตัวเองที่ทำงานบน Windows ในลักษณะเป็นระบบปฏิบัติการเสมือน (virtualized) นั่นหมายความว่า WSL2 ทำงานบนระบบปฏิบัติการ Windows แต่ให้บริการและรันโปรแกรม Linux ในพื้นที่เโมเลชันของตัวเอง

การใช้งาน WSL2 ช่วยให้ผู้ใช้สามารถเรียกใช้คำสั่ง Linux และรันแอปพลิเคชัน Linux ได้โดยตรงในเครื่อง Windows โดยไม่ต้องติดตั้งเครื่องมือสำหรับจำลองหรือตั้งค่าเซิร์ฟเวอร์ Linux แยกต่างหาก ผู้ใช้งานสามารถเรียกใช้คำสั่งที่รู้จักของ Linux, ติดตั้งและใช้งานซอฟต์แวร์ Linux, และทำงานกับไฟล์และไดเรกทอรีของ Linux ได้ในระบบไฟล์ของ Windows

สามารถศึกษาเพิ่มเติมได้จาก [Windows Subsystem for Linux Documentation | Microsoft Learn](#)

ในการเริ่มใช้งาน WSL2 มีขั้นตอนการติดตั้งดังต่อไปนี้

### 1.1 Enable the Windows Subsystem for Linux

เปิดใช้พีเออร์เสริม "Windows Subsystem for Linux" ก่อนติดตั้ง Linux รุ่นต่าง ๆ บน Windows ซึ่งจะต้องใช้ PowerShell ในการดำเนินงาน โดยการดังนี้ Start menu > PowerShell > right-click > Run as Administrator

จากนั้นพิมพ์คำสั่งใน PowerShell ดังนี้

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

### 1.2 Check requirements for running WSL 2

หากต้องการปรับเป็น WSL 2 คุณต้องใช้

Windows 10

- สำหรับระบบ x64: รุ่น 1903 หรือใหม่กว่า พร้อม Build 18362 หรือใหม่กว่า
- สำหรับระบบ ARM64: รุ่น 2004 หรือใหม่กว่า พร้อม Build 19041 หรือใหม่กว่า

หรือ Windows 11

หากต้องการตรวจสอบรุ่นและหมายเลขบิลด์ของคุณ ให้เลือก Windows logo key + R พิมพ์ winver เลือกตกลง ปรับเป็น Windows รุ่นล่าสุดในเมนูการตั้งค่า

#### หมายเหตุ

- 1) รุ่นต่ำกว่า 18362 ไม่รองรับ WSL 2 ใช้ Windows Update Assistant เพื่อปรับรุ่นของ Windows
- 2) หากคุณใช้ Windows 10 รุ่น 1903 หรือ 1909 ให้เปิด "การตั้งค่า" จากเมนู Windows ไปที่ "Update & Security" แล้วเลือก "Check for Updates" Build number ต้องเป็น 18362.1049+ หรือ 18363.1049+

### 1.3 Enable Virtual Machine Feature

ก่อนการติดตั้ง WSL 2 ต้องเปิดใช้งานคุณสมบัติเสริมของแพลตฟอร์มเครื่องเสมือน ตรวจสอบคุณสมบัติ [Troubleshooting Windows Subsystem for Linux | Microsoft Learn](#) และพิมพ์คำสั่งใน PowerShell ดังนี้

```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

จากนั้นรีสตาร์ทเครื่อง เพื่อทำการติดตั้ง WSL และอัปเดตเป็น WSL 2

### 1.4 Download the Linux kernel update package

- 1) ดาวน์โหลดแพ็คเกจล่าสุด:
  - [https://wslstorestorage.blob.core.windows.net/wslblob/wsl\\_update\\_x64.msi](https://wslstorestorage.blob.core.windows.net/wslblob/wsl_update_x64.msi)
- 2) เรียกใช้แพ็คเกจการปรับปรุงที่ดาวน์โหลดในขั้นตอนก่อนหน้านี้

#### หมายเหตุ

หากคุณใช้เครื่อง ARM64 โปรดดาวน์โหลดแพ็คเกจ ARM64 แทน หาก你不แน่ใจว่าคุณมีเครื่องประเภทใด ให้เปิด Command Prompt หรือ PowerShell แล้วพิมพ์: `systeminfo | find "System Type"`

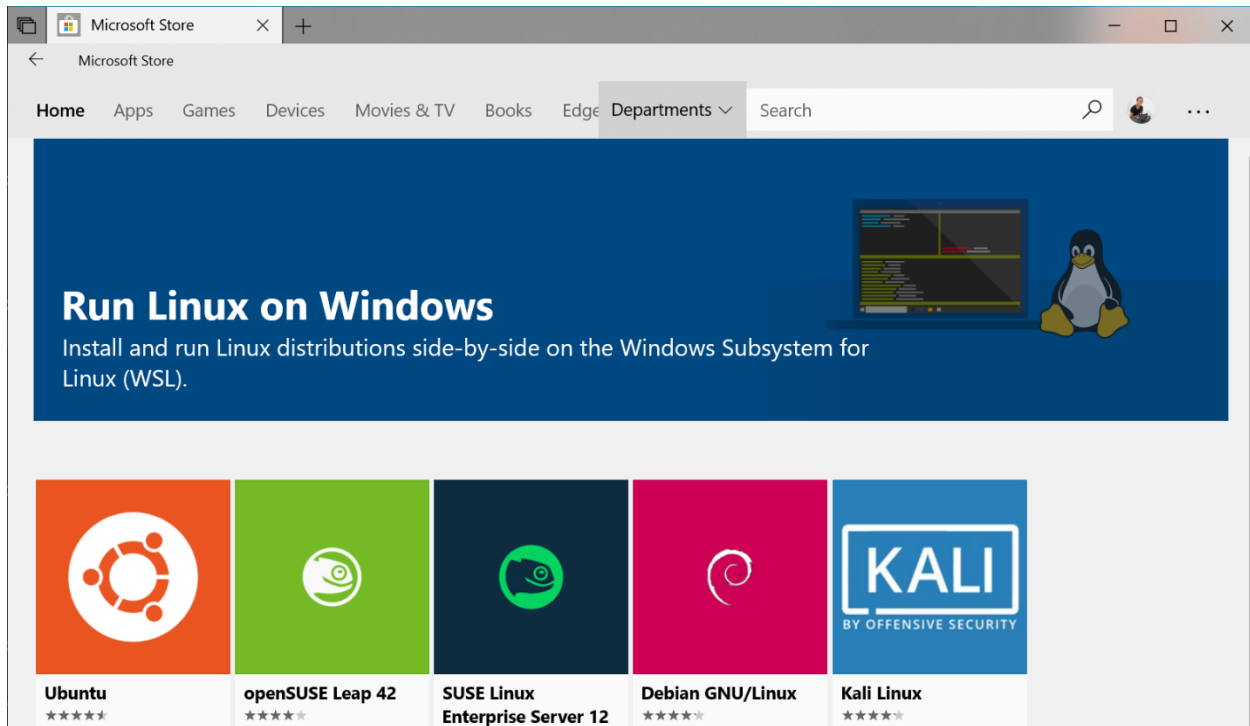
### 1.5 Set WSL 2 as your default version

เปิด PowerShell และเรียกใช้คำสั่งนี้เพื่อตั้งค่า WSL 2 เป็นรุ่นเริ่มต้น เมื่อทำการติดตั้ง Linux ใหม่

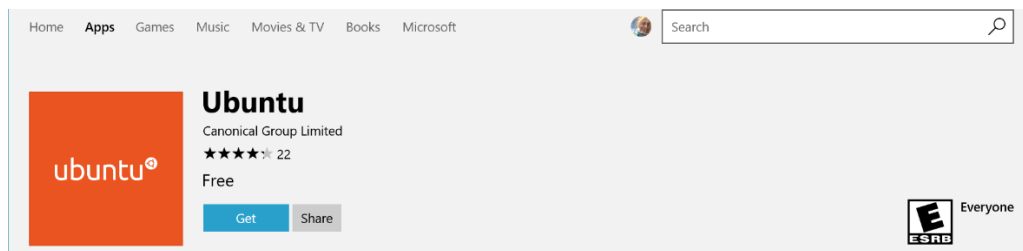
```
wsl --set-default-version 2
```

## 1.6 Install your Linux distribution of choice

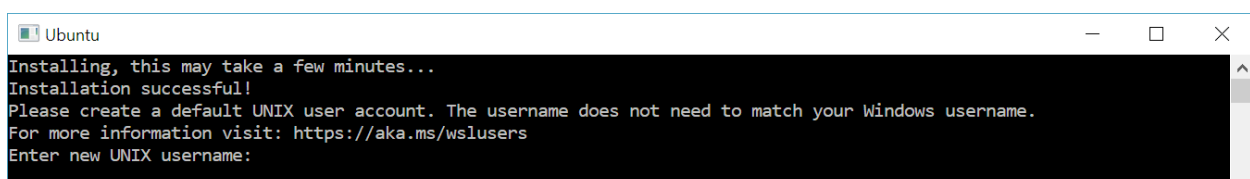
1.6.1 Open the Microsoft Store (<https://aka.ms/wslstore>) and select your favorite Linux distribution.



1.6.2 From the distribution's page, select "Get".



ครั้งแรกที่เปิด Linux ที่ติดตั้งใหม่ หน้าต่างคอนโซลจะเปิดขึ้น รอหนึ่งหรือสองนาที่เพื่อให้ไฟล์คลายการบีบอัดและจัดเก็บไว้ในเครื่องคอมพิวเตอร์ จากนั้นต้องสร้างบัญชีผู้ใช้และรหัสผ่าน [Set up a WSL development environment | Microsoft Learn](https://aka.ms/wslusers) เสร็จขั้นตอนการติดตั้ง Linux OS บน Windows



## 2. Docker

Docker เป็นแพลตฟอร์มสำหรับการจัดการและปรับใช้แอปพลิเคชันที่อยู่ในสภาพแวดล้อมที่เรียกว่า "คอนเทนเนอร์" (Containers) ซึ่งเป็นเทคโนโลยีที่ช่วยให้สามารถแพ็คเกจแอปพลิเคชันและส่วนประกอบต่าง ๆ ที่เกี่ยวข้องไว้ในคอนเทนเนอร์เดียวกันได้ ทำให้ง่ายต่อการโอนย้ายและเปิดใช้งานแอปพลิเคชันในสภาพแวดล้อมที่ต่างกันได้อย่างยืดหยุ่นและมีประสิทธิภาพมากขึ้น นอกจากนี้ Docker ยังมีเครื่องมือที่ช่วยในการจัดการคอนเทนเนอร์ เช่น Docker Compose เพื่อการจัดการและกำหนดคอนเทนเนอร์และบริการที่เกี่ยวข้องอื่น ๆ

### 2.1 Install and Use Docker on Ubuntu

#### 2.1.1 Installing Docker

การติดตั้ง Docker เวอร์ชันล่าสุดใน Ubuntu นั้นไม่ควรใช้เวอร์ชันที่มีอยู่ในที่เก็บ Ubuntu เพราะอาจไม่เป็นเวอร์ชันล่าสุด แทนที่นั้นควรติดตั้ง Docker จากที่เก็บ Docker อย่างเป็นทางการโดยการเพิ่มแหล่งที่มาของแพ็คเกจใหม่และคีย์ GPG จาก Docker เพื่อให้มั่นใจว่าการดาวน์โหลดและติดตั้งแพ็คเกจถูกต้องตามมาตรฐาน นั่นคือขั้นตอนที่แนะนำในการติดตั้ง Docker ให้ถูกต้องและปลอดภัย รายละเอียดดังต่อไปนี้

1) Update your existing list of packages:

```
sudo apt update
```

2) Install a few prerequisite packages which let apt use packages over HTTPS:

```
sudo apt install apt-transport-https ca-certificates curl software-properties-common
```

3) Add the GPG key for the official Docker repository to your system:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

4) Add the Docker repository to APT sources:

```
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"
```

5) Make sure you are about to install from the Docker repo instead of the default Ubuntu repo:

```
apt-cache policy docker-ce
```

You'll see output like this, although the version number for Docker may be different:

```
docker-ce:
  Installed: (none)
  Candidate: 5:19.03.9~3-0~ubuntu-focal
  Version table:
     5:19.03.9~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
```

6) Finally, install Docker:

```
sudo apt install docker-ce
```

Docker should now be installed, the daemon started, and the process enabled to start on boot.

Check that it's running:

```
sudo systemctl status docker
```

The output should be similar to the following, showing that the service is active and running:

```
Output
• docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2020-05-19 17:00:41 UTC; 17s ago
 TriggeredBy: • docker.socket
    Docs: https://docs.docker.com
 Main PID: 24321 (dockerd)
    Tasks: 8
   Memory: 46.4M
    CGroup: /system.slice/docker.service
            └─24321 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

### 2.1.2 Executing the Docker Command Without Sudo (Optional)

1) If you want to avoid typing `sudo` whenever you run the `docker` command, add your username to the `docker` group:

```
sudo usermod -aG docker ${USER}
```



2) To apply the new group membership, log out of the server and back in, or type the following:

```
su - ${USER}
```

3) If you need to add a user to the docker group that you're not logged in as, declare that username explicitly using:

```
sudo usermod -aG docker username
```

## 2.2 How to use Docker and Docker compose to Start service (php, Apache, Node-Red, Influxdb, MySQL, PHPMysqladmin, Adminer, Grafana and MQTT)

ในขั้นตอนนี้จะต้องเข้าไปใน Command ของ Ubuntu ซึ่งสามารถใช้คีย์ลัดในการค้นหาคือ **Windows key + R** จากนั้นพิมพ์ **ubuntu** แล้ว Enter เพื่อเปิดหน้า Ubuntu



### 2.2.1 การ Clone Project จาก GitHub เพื่อใช้งาน

ในหน่วยนี้ผู้จัดทำได้สร้าง Project ไว้แล้ว โดยนำไปไว้ใน GitHub ซึ่งเป็นแพลตฟอร์มเว็บเซอร์วิสที่ให้บริการในการพัฒนาซอฟต์แวร์ร่วมกันแบบระบบควบคุมเวอร์ชัน (Version Control System) ที่ใช้งานได้อย่างทั่วไพบ โดย GitHub ให้ผู้ใช้สร้างและจัดการโปรเจกต์ซอฟต์แวร์ของตน และทำให้ผู้ใช้สามารถทำงานร่วมกันบนโครงการเดียวกันได้ง่ายขึ้น

GitHub ให้บริการการจัดเก็บโค้ดและการเปิดเผยโค้ดที่เป็นระบบควบคุมเวอร์ชัน Git ซึ่งเป็นระบบควบคุมเวอร์ชันที่ใช้กันอย่างแพร่หลายในการพัฒนาซอฟต์แวร์ ผู้ใช้สามารถสร้างรีพอสิตอรี (repository) เพื่อจัดเก็บโค้ดและเทรนด์การเปลี่ยนแปลงต่างๆ ในโปรเจกต์ของตนได้ รวมถึงสามารถทำงานร่วมกับผู้อื่น และดูประวัติการเปลี่ยนแปลง และแก้ไขข้อบกพร่องของโค้ดร่วมกันได้

นอกจากนี้ GitHub ยังมีคุณสมบัติและเครื่องมือต่าง ๆ เพื่อช่วยในกระบวนการพัฒนาซอฟต์แวร์ เช่น การตรวจสอบการเปลี่ยนแปลงในโค้ด (code review), การจัดการงาน (issue tracking), การบริหารจัดการโค้ด (project management), การแบ่งงาน (task management), การทดสอบ (testing) เป็นต้น

GitHub เป็นแพลตฟอร์มที่ได้รับความนิยมอย่างมากในชุมชนนักพัฒนาซอฟต์แวร์ และมีการใช้งานกว้างขวางทั่วโลก นอกจากนี้ยังมีการบริการ GitHub Enterprise ที่ให้การเปิดเผยโค้ดภายในองค์กรเป็นไปอย่างปลอดภัย

### การใช้งาน Git เพื่อ Clone Project มาจาก GitHub แล้วเริ่มใช้งาน

ในการเริ่มต้นจะต้อง Clone Project มาจาก GitHub ก่อน (1) จากนั้น เข้าไปยังโฟลเดอร์ docker ที่ Clone มา (2) และใช้คำสั่งเพื่อเริ่มการทำงานของ Docker (3)

```
(1) $ git clone https://github.com/woeis-me/docker.git
(2) $ cd docker
(3) $ sudo docker compose up -d
```

```
nattawat@nattawat:~/download$ git clone https://github.com/woeis-me/docker.git
Cloning into 'docker'...
remote: Enumerating objects: 112, done.
remote: Counting objects: 100% (112/112), done.
remote: Compressing objects: 100% (64/64), done.
remote: Total 112 (delta 46), reused 94 (delta 32), pack-reused 0
Receiving objects: 100% (112/112), 61.60 KiB | 1.26 MiB/s, done.
Resolving deltas: 100% (46/46), done.
nattawat@nattawat:~/download$ cd docker
nattawat@nattawat:~/download/docker$ sudo docker compose up -d
[+] Building 0.0s (0/0)
[+] Running 9/9
✓ Network docker_networkName Created
✓ Container user-adminer Started
✓ Container user-mosquitto Started
✓ Container user-influxdb Started
✓ Container user-db Started
✓ Container user-myadmin Started
✓ Container user-php-apache Started
✓ Container user-nodered Started
✓ Container user-grafana Started
```

## 2.2.2 การตั้งค่า Username และ Password ให้กับ Service

### 1) Node-Red Config

ตั้งค่า Node-Red ใน Command โดยพิมพ์คำสั่งตามลำดับ (1) – (4)

```
(1) $ sudo apt-get install npm
(2) $ sudo npm install -g node-red-admin
(3) $ sudo docker exec -it user-nodered sh
(4) $ npx node-red admin hash-pw
```

ในข้อที่ (4) Command จะให้ใส่รหัสผ่าน (แต่จะไม่แสดงบน Command) จากนั้นจะได้รหัสผ่านจากการ  
เข้ารหัส ตัวอย่างเช่น

```
$2b$08$X8stDRyPOvBU6KCSEi5j8uWeETKA50KLvegXRnHf3hRUCV7MU2P72
```

จากนั้นคัดลอกไปวางไว้ในไฟล์ /nodered/setting.js (สามารถใช้คำสั่งใน Command `sudo nano /nodered/setting.js`)

```
adminAuth: {  
  type: "credentials",  
  users: [{  
    username: "admin",  
    password: "$2b$08$X8stDRyPOvBU6KCSEi5j8uWeETKA50KLvegXRnHf3hRUCV7MU2P72",  
    permissions: "*"   
  }]  
},
```

## 2) Influxdb Config

เริ่มต้นทำการคอมเมนต์โค้ด ซึ่งอยู่ในไดเรกทอรี /influxdb/config/influxdb.conf ในบรรทัดที่ 263

```
#Determines whether user authentication is enabled over HTTP/HTTPS.  
#auth-enabled = true
```

จากนั้นพิมพ์คำสั่งดังต่อไปนี้ โดยแก้ไข ชื่อผู้ใช้และรหัสผ่าน ตามความเหมาะสม

```
$ sudo docker exec -it user-influxdb sh  
$ influx  
$ create database mydb  
$ CREATE USER "admin" WITH PASSWORD 'admin_passwd' WITH ALL PRIVILEGES  
$ exit  
$ exit
```

จากนั้นทำการปิดคอมเมนต์โค้ด ซึ่งอยู่ในไดเรกทอรี /influxdb/config/influxdb.conf ในบรรทัดที่ 263

```
#Determines whether user authentication is enabled over HTTP/HTTPS.  
auth-enabled = true
```

## 3) MQTT Config

การตั้งค่า MQTT ให้มีชื่อผู้ใช้และรหัสผ่าน จะต้องใช้คำสั่งในการดำเนินงาน โดยสามารถพิมพ์คำสั่ง  
ดังต่อไปนี้

```
$ sudo docker exec -it user-mosquitto sh
$ mosquitto_passwd -b /mosquitto/config/password_file user pass
$ exit
```

ตัวอย่างการตั้งค่ารหัสผ่าน

```
$ mosquitto_passwd -b /mosquitto/config/password_file admin 12345678
```

การตั้งค่าการเข้ารหัสของโปรแกรมทั้งหมดครบทุกขั้นตอนแล้ว จะต้องทำการเริ่มระบบใหม่อีกครั้ง โดยใช้คำสั่ง

```
$ docker compose restart
```

### 3. How to use ...

การใช้งานโปรแกรม โปรเจค หรือ เซอร์วิสที่ได้ดำเนินการในหัวข้อที่ 2 สามารถดำเนินการได้ดังรายละเอียดต่อไปนี้

#### 3.1 Node-Red

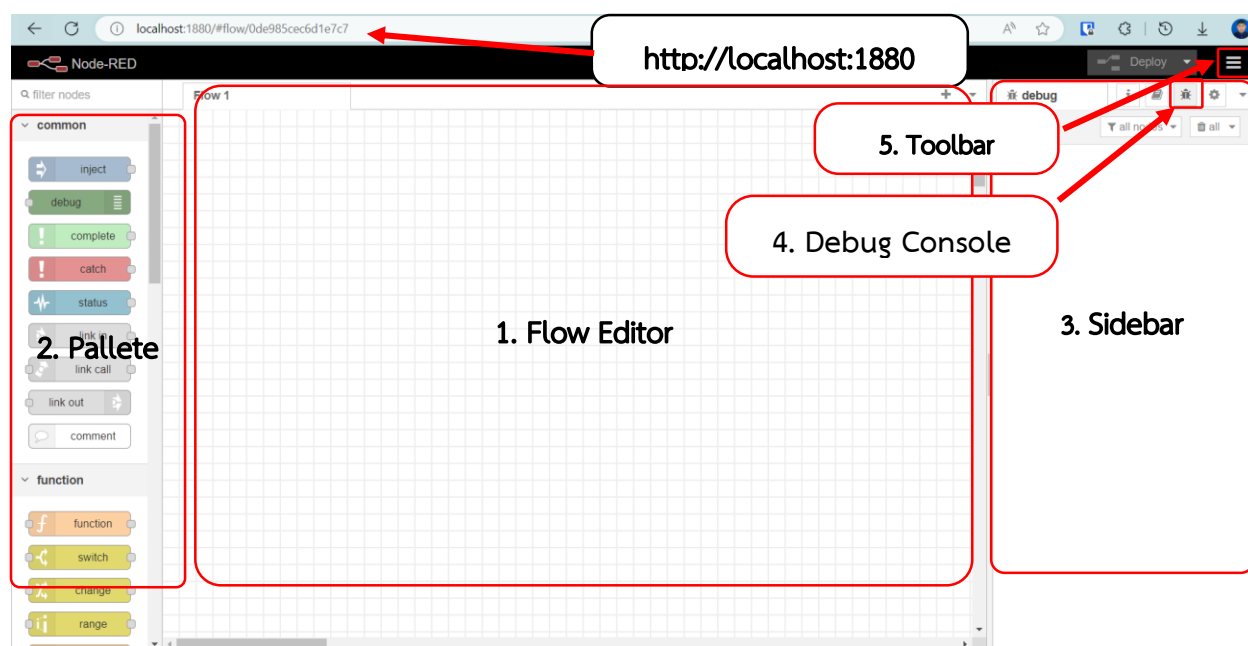
Node-Red เป็นแพลตฟอร์มแบบเบสออนไลน์เปิดต้นฉบับที่ใช้สำหรับสร้างและจัดการกระบวนการอัตโนมัติที่เชื่อมต่อกับอุปกรณ์และบริการต่างๆ ส่วนใหญ่ใช้งานในการสร้างและควบคุมอินเทอร์เน็ตของสร้างของอุปกรณ์อิเล็กทรอนิกส์ (IoT) โดยใช้กราฟสร้างเส้นทางที่เรียกว่า "flow" ซึ่งประกอบด้วยโหนด (nodes) และการเชื่อมต่อกันเพื่อให้ข้อมูลไหลผ่านกัน

หน้าของ Node-Red ประกอบด้วยส่วนหลัก ๆ ดังนี้:

1. Flow Editor: เป็นส่วนที่ให้ผู้ใช้งานสร้างและแก้ไขกราฟฟลว์ (flow) ซึ่งเป็นการเชื่อมต่อและประมวลผลข้อมูลระหว่างโหนด (node) ต่าง ๆ ภายในโครงการ Node-Red
2. Palette: เป็นเครื่องมือที่ใช้ในการเลือกและเพิ่มโหนด (node) เข้าสู่ Flow Editor เพื่อให้สามารถใช้งานและปรับแต่งกราฟฟลว์ได้ตามต้องการ
3. Sidebar: เป็นส่วนที่มีเครื่องมือและการตั้งค่าต่าง ๆ ที่เกี่ยวข้องกับโปรเจกต์ Node-Red รวมถึงการจัดการกับโหนด (node) ที่อยู่ใน Flow Editor ในรูปแบบของตารางหรือรายการ
4. Debug Console: เป็นส่วนที่แสดงผลลัพธ์หรือข้อมูลที่ผ่านการประมวลผลในโหนด (node) ต่าง ๆ ภายใน Flow Editor ให้ผู้ใช้งานสามารถติดตามข้อมูลที่ถูกส่งผ่านได้

5. Toolbar: เป็นแถบเครื่องมือที่อยู่ด้านบนของหน้าจอ ใช้ในการบันทึกโปรเจกต์ การเริ่มต้น/หยุดการทำงาน ของโปรเจกต์ และฟังก์ชันอื่น ๆ เช่น Import/Export, Deploy, และการเปิด/ปิด Sidebar

หน้าที่หลักของ Node-Red คือให้ผู้ใช้สร้างและจัดการกราฟแฟลว์การประมวลผลข้อมูลต่าง ๆ ในรูปแบบของ โหนด โดยทำหน้าที่เชื่อมต่อและประมวลผลข้อมูลในรูปแบบของแพลตฟอร์มสตรีมข้อมูล (streaming platform) โดยรองรับการสร้างและปรับแต่งโหนดต่าง ๆ เพื่อให้เหมาะสมกับงานที่ต้องการทำในโครงการแต่ละรายการ



วิธีการใช้งาน Node-Red:

1. ติดตั้งและเปิดใช้งาน Node-Red: ให้ติดตั้ง Node-Red บนเครื่องคอมพิวเตอร์หรือเซิร์ฟเวอร์ของคุณ โดยใช้คำสั่งที่เหมาะสมสำหรับระบบปฏิบัติการที่คุณใช้ หลังจากติดตั้งเสร็จสิ้น คุณสามารถเรียกใช้งาน Node-Red ได้ผ่านทางเบราว์เซอร์โดยพิมพ์ URL ต่อท้ายด้วยพอร์ตที่กำหนด (เช่น <http://localhost:1880>)

2. สร้างเส้นทาง (flow): เมื่อเริ่มต้น Node-Red คุณจะเห็นหน้าต่างของตัวจัดการเส้นทาง ในการสร้างเส้นทางใหม่ ให้ลากโหนดจากแถบเครื่องมือและวางลงในพื้นที่ทำงาน (workspace) ตามต้องการ คุณสามารถเพิ่มและปรับเปลี่ยนโหนดต่างๆ และเชื่อมต่อกันเพื่อกำหนดกระบวนการที่คุณต้องการให้เกิดขึ้น

3. กำหนดค่าและการตั้งค่า: คุณสามารถกำหนดค่าและการตั้งค่าของโหนดแต่ละตัวได้ตามความต้องการ โดยคลิกขวาที่โหนดและเลือก "Edit" เพื่อเข้าสู่หน้าต่างการตั้งค่า ตรวจสอบค่าต่างๆ และปรับเปลี่ยนตามต้องการ

4. เชื่อมต่อกับอุปกรณ์หรือบริการ: Node-Red มีความสามารถในการเชื่อมต่อกับอุปกรณ์หรือบริการต่างๆ เช่น อุปกรณ์ IoT, ฐานข้อมูล, บริการเว็บ, และอื่นๆ ให้คลิกขวาที่โหนดที่ต้องการเชื่อมต่อและเลือก "Edit" เพื่อกำหนดการเชื่อมต่อให้เหมาะสม

5. บันทึกและประมวลผล: เมื่อคุณได้กำหนดเส้นทางที่ต้องการและกำหนดการตั้งค่าให้เรียบร้อยแล้ว คุณสามารถบันทึกและประมวลผลเส้นทางนั้นๆ โดยกดปุ่ม "Deploy" ที่มุมบนด้านขวาของหน้าต่าง Node-Red นี้จะทำให้กระบวนการที่คุณสร้างขึ้นเริ่มทำงาน

6. ตรวจสอบผลลัพธ์: Node-Red จะแสดงผลลัพธ์ของกระบวนการในหน้าต่างของเบราว์เซอร์ คุณสามารถตรวจสอบผลลัพธ์จากการทำงานของโหนดแต่ละตัว และดูข้อมูลที่ได้รับจากอุปกรณ์หรือบริการที่คุณเชื่อมต่อ

## การใช้งานโหนดต่าง ๆ ใน Node-Red

### 1. Inject Node



Inject ใน Node-Red เป็นโหนดที่ใช้เป็นจุดเริ่มต้นของกระบวนการ โดยเป็นตัวแทนของเหตุการณ์ (event) หรือเหตุการณ์ที่เริ่มต้นกระบวนการทำงาน

องค์ประกอบหลักของโหนด Inject ประกอบด้วย:

1. Timestamp (ประเภท: timestamp): เป็นข้อมูลเวลาที่เริ่มต้นกระบวนการ ค่านี้สามารถกำหนดได้ว่าเป็นเวลาปัจจุบันหรือเวลาที่กำหนดเอง

2. Payload (ประเภท: ข้อมูล): เป็นข้อมูลที่ส่งผ่านไปยังโหนดถัดไปในกราฟ ข้อมูลใน Payload สามารถเป็นประเภทต่างๆ เช่น ข้อความ (string), ตัวเลข (number), ออบเจกต์ (object) หรืออาร์เรย์ (array) ซึ่งขึ้นอยู่กับข้อกำหนดค่าในการใช้งาน

โหนด Inject สามารถใช้ในการเริ่มต้นกระบวนการในหลายวิธี เช่น:

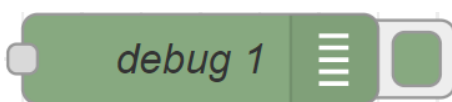
1. เริ่มต้นกระบวนการที่กำหนดเวลา: โหนด Inject สามารถกำหนดให้เริ่มต้นกระบวนการในเวลาที่กำหนด โดยการเลือกค่า Timestamp เป็นเวลาที่คุณต้องการให้เกิดเหตุการณ์

2. เริ่มต้นกระบวนการด้วยการกดปุ่ม: โหนด Inject ยังสามารถใช้เป็นปุ่มเริ่มต้นกระบวนการ คุณสามารถกำหนดให้เหตุการณ์เกิดขึ้นเมื่อคุณคลิกที่ปุ่ม Inject ในหน้าต่าง Node-Red

3. เริ่มต้นกระบวนการจากเหตุการณ์ภายนอก: คุณสามารถใช้โหนด Inject เพื่อรับเหตุการณ์จากแหล่งข้อมูลภายนอก เช่น การรับข้อมูลจากเซ็นเซอร์หรือระบบอื่นๆ และนำข้อมูลที่รับมาเป็น Payload เพื่อส่งต่อไปยังโหนดอื่นๆ ในกราฟ

โหนด Inject เป็นโหนดที่มีความสำคัญในการสร้างกราฟ Node-Red เนื่องจากมันเป็นจุดเริ่มต้นของกระบวนการและเหตุการณ์ที่เกิดขึ้นในโหนดอื่นๆ สามารถตั้งค่าและใช้งานโหนด Inject ได้ตามความต้องการของโปรเจกต์

## 2. Debug Node



Debug เป็นกระบวนการการตรวจสอบและแก้ไขข้อผิดพลาดหรือปัญหาที่เกิดขึ้นในโปรแกรมหรือระบบที่กำลังทำงาน การ debug ช่วยให้นักพัฒนาซอฟต์แวร์สามารถวิเคราะห์และตรวจสอบข้อมูลที่เกี่ยวข้องกับข้อผิดพลาดเพื่อหาสาเหตุและแก้ไขได้

การ debug มักใช้เครื่องมือหรือโปรแกรมช่วย เช่น:

1. ตัวแปรการแสดงผล (print statement): การแสดงผลข้อมูลต่างๆ ในตำแหน่งที่สำคัญของโค้ด เช่น ค่าตัวแปร ข้อความสถานะ เพื่อตรวจสอบค่าและสถานะของโปรแกรมในขณะที่ทำงาน

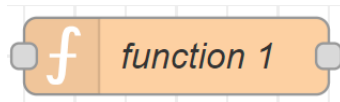
2. เครื่องมือ Debugging: เป็นโปรแกรมหรืออุปกรณ์ที่ให้ความสามารถในการตรวจสอบและแก้ไขข้อผิดพลาด มีฟังก์ชันต่างๆ เช่น การหยุดทำงานที่จุดหนึ่ง (breakpoint) เพื่อตรวจสอบค่าข้อมูล การดูค่าตัวแปร การติดตามการเข้าถึงฟังก์ชัน และอื่นๆ

3. การบันทึกข้อมูล (logging): การบันทึกข้อมูลเกี่ยวกับการทำงานของโปรแกรม รวมถึงข้อผิดพลาดที่เกิดขึ้น ไว้ในไฟล์หรือระบบบันทึกเพื่อวิเคราะห์หรือตรวจสอบภายหลัง

4. เครื่องมือตรวจสอบการทำงาน (debugger): เครื่องมือที่ช่วยให้นักพัฒนาสามารถติดตามกระบวนการทำงานของโปรแกรมได้ละเอียด รวมถึงเข้าถึงค่าตัวแปร การเรียกฟังก์ชัน และกระบวนการอื่นๆ ในระหว่างการทำงาน

การ debug เป็นกระบวนการที่สำคัญในการพัฒนาซอฟต์แวร์เนื่องจากช่วยให้สามารถตรวจสอบและแก้ไขข้อผิดพลาดได้อย่างรวดเร็ว นอกจากนี้ยังช่วยให้เข้าใจการทำงานของโค้ดและระบบในระดับลึกขึ้น

### 3. Function Node



Function ใน Node-Red เป็นโหนดที่ใช้ในการเขียนโค้ดหรือสคริปต์ที่กำหนดเองเพื่อประมวลผลข้อมูลหรือกระบวนการที่ซับซ้อนกว่าที่โหนดอื่นๆ ใน Node-Red สามารถทำได้ โดยโหนดนี้ให้คุณเขียนโค้ด JavaScript ภายในตัวเองเพื่อประมวลผลข้อมูลที่เข้าสู่โหนดและส่งผลลัพธ์ออกไปยังโหนดถัดไปในกราฟ Node-Red

โหนด Function มีลักษณะเด่นต่อไปนี้:

1. ภาษาโปรแกรม: สามารถเขียนโค้ด JavaScript ภายในโหนด Function ซึ่งเป็นภาษาโปรแกรมที่กว้างขวางและมีความสามารถมากมาย เพื่อประมวลผลข้อมูลตามต้องการ
2. การประมวลผลเส้นทางที่ซับซ้อน: โหนด Function ช่วยให้คุณสามารถดำเนินการประมวลผลที่ซับซ้อนและยืดหยุ่นขึ้นได้ โดยคุณสามารถเขียนโค้ดที่ต้องการในการประมวลผลข้อมูล การควบคุมการไหลของข้อมูล หรือการเรียกใช้งานบริการหรืออุปกรณ์ภายนอกได้
3. การเข้าถึงข้อมูล: โหนด Function สามารถเข้าถึงข้อมูลที่เข้าสู่โหนดได้ โดยข้อมูลนั้นจะถูกส่งผ่านเป็นพารามิเตอร์ในฟังก์ชัน JavaScript ภายในโหนด เช่น ข้อมูลจากโหนดที่เชื่อมต่อมา หรือข้อมูลที่ได้รับผ่านการสื่อสารกับอุปกรณ์หรือบริการอื่นๆ

โหนด Function เป็นอีกหนึ่งเครื่องมือที่ทรงพลังในการทำงาน โดยมีความยืดหยุ่นและความสามารถในการปรับแต่งการประมวลผลข้อมูลและการควบคุมไหลข้อมูลให้ตรงตามความต้องการของโปรเจกต์