

NODERED及MQTT 應用

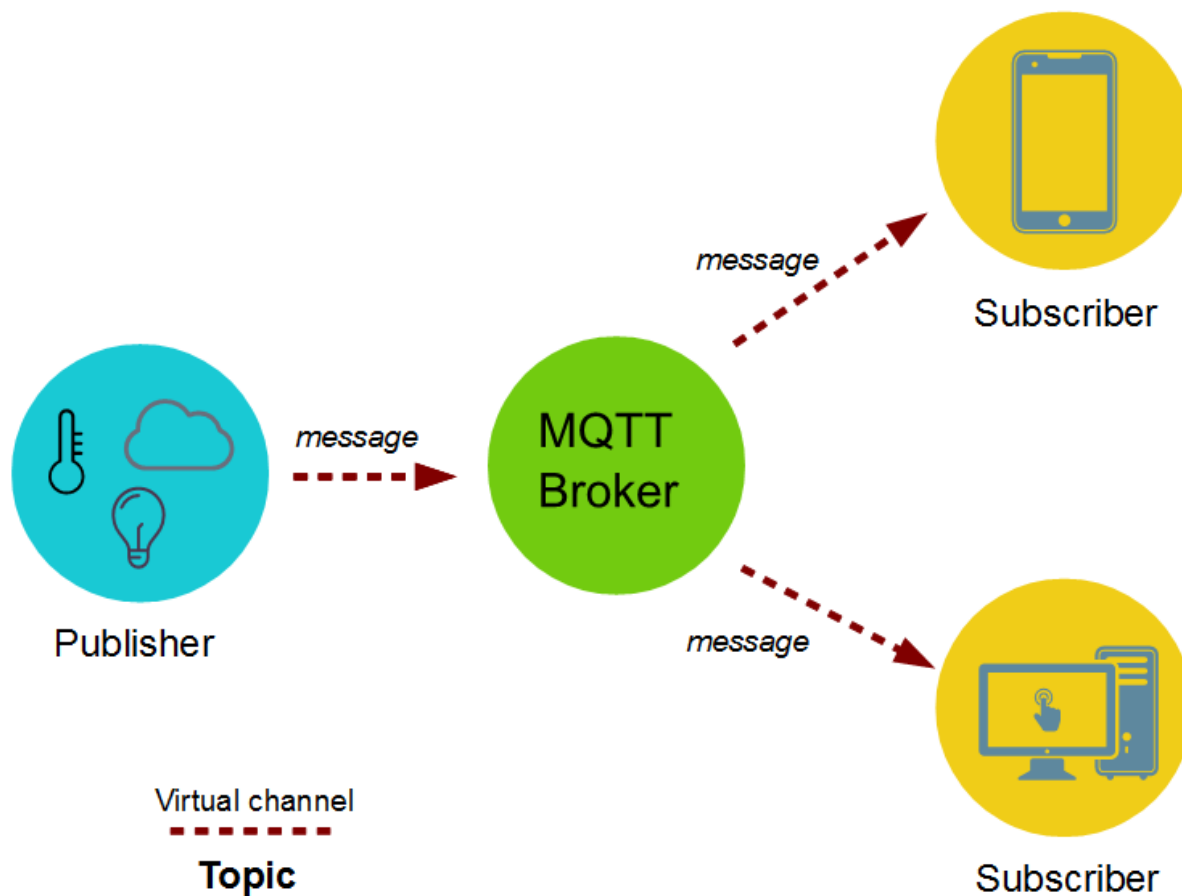
Speaker: Malo – 楊穎煜



MQTT入門應用

MQTT入門

□ 一種輕量化的物聯網協定



MQTT入門-以Linux工具為例

□ MQTT的測試工具:

- ▣ 安裝：`sudo apt install mosquitto-clients`

- ▣ 接收訊息:

 - ▣ `mosquitto_sub -h broker.hivemq.com -t /malo/hello`

- ▣ 傳送訊息:

 - ▣ `mosquitto_pub -h broker.hivemq.com -t /malo/hello -m "hello world"`

MQTT入門

- 發佈MQTT訊息 → 控制燈:

- Topic: /malo/light1

- 傳送訊息:

- mosquitto_pub -broker.hivemq.com -t /malo/light1 -m "1"

- 接收端的設備：hub-5168+

MQTT入門

- 沒有Linux工具!
- 沒關係我們自己用圖像化工具進行MQTT操作

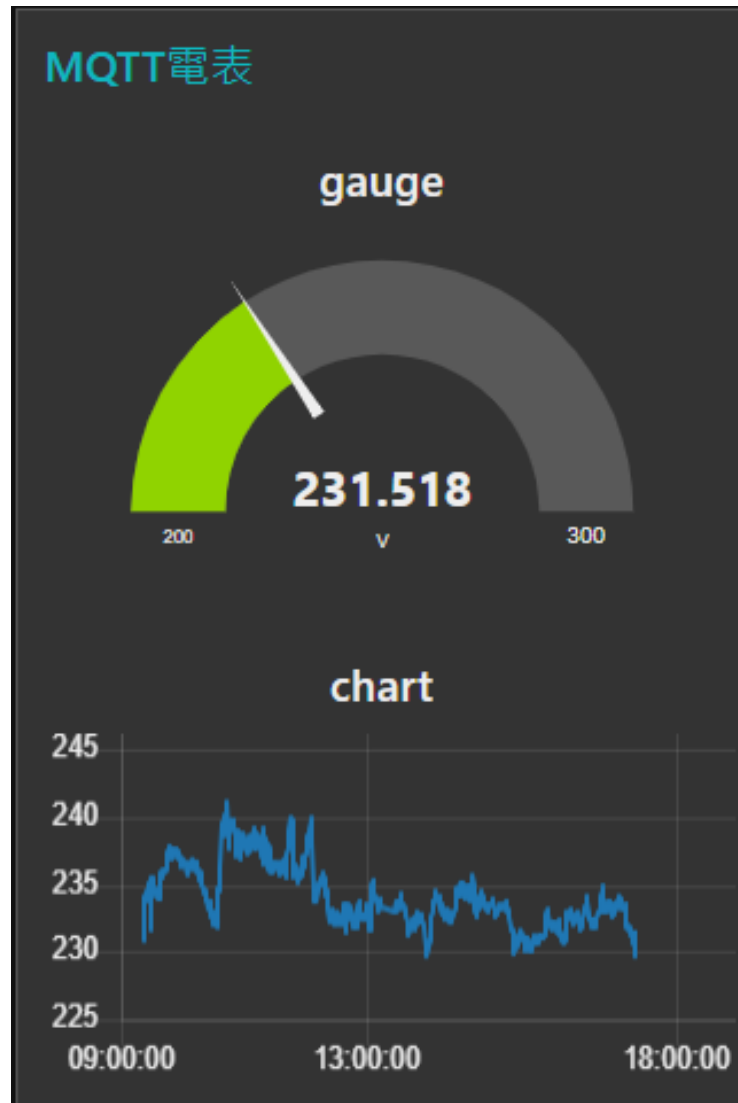


NodeRED應用

NodeRED應用-何謂NodeRED

- ❑ 參考網站：<https://nodered.org/>
- ❑ 以網頁型式，提供流程圖圖像化的程式規畫
- ❑ 組合各種功能的Node(節點)，讓使用者使用很少的程式碼，就可以進行IoT整合
- ❑ 底層以Node.js開發

NodeRED應用-資訊視覺化好幫手



安裝NodeRED (Windows)

NodeRED應用-安裝NodeRED

- 使用這一個版本最為方便，因此先教這個方式
- 到這邊有portable的安裝：
(<https://github.com/mdkrieg/nodered-portable>)
- 或是到這邊直接下載安裝 (我目前使的是v1.1.1的版本)：
(<https://sourceforge.net/projects/nodered-portable/>)
- 因沒有node.js所以會發現少了「節點管理」的功能
- 需要先安裝 Node.js ([link](#))才能進行節點的安裝管理

NodeRED應用-安裝NodeRED

□ 安裝程式

▣ 先安裝Node.js環境 (NodeRED底層使用Node.js)

▣ <https://nodejs.org/en/>

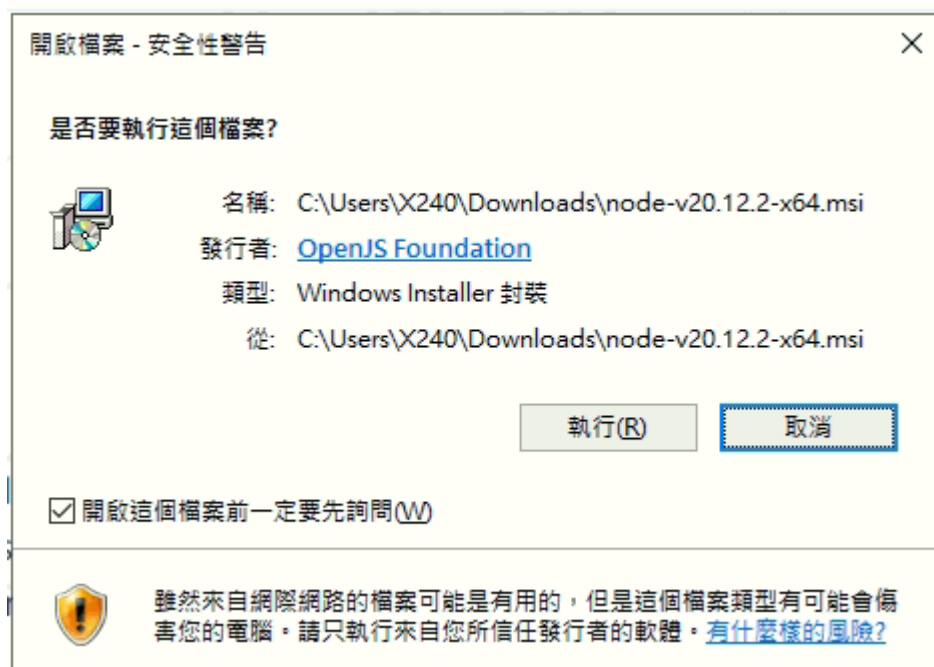


NodeRED應用-安裝NodeRED

□ 安裝程式

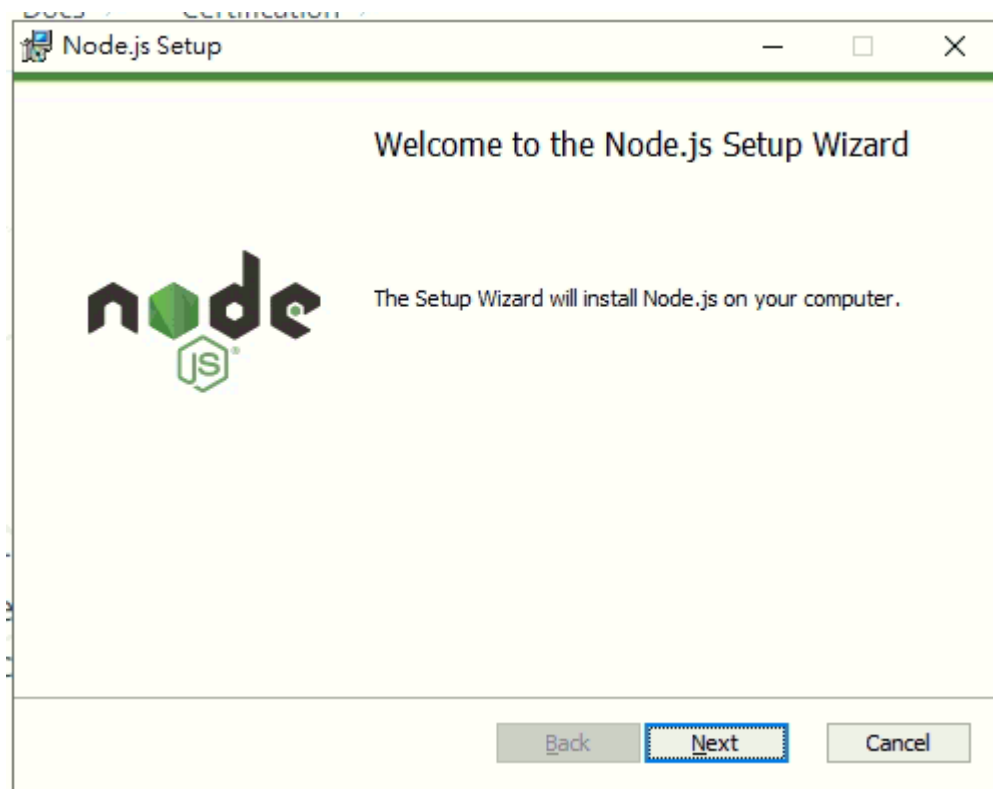
□ 先安裝Node.js環境 (NodeRED底層使用Node.js)

□ <https://nodejs.org/en/>



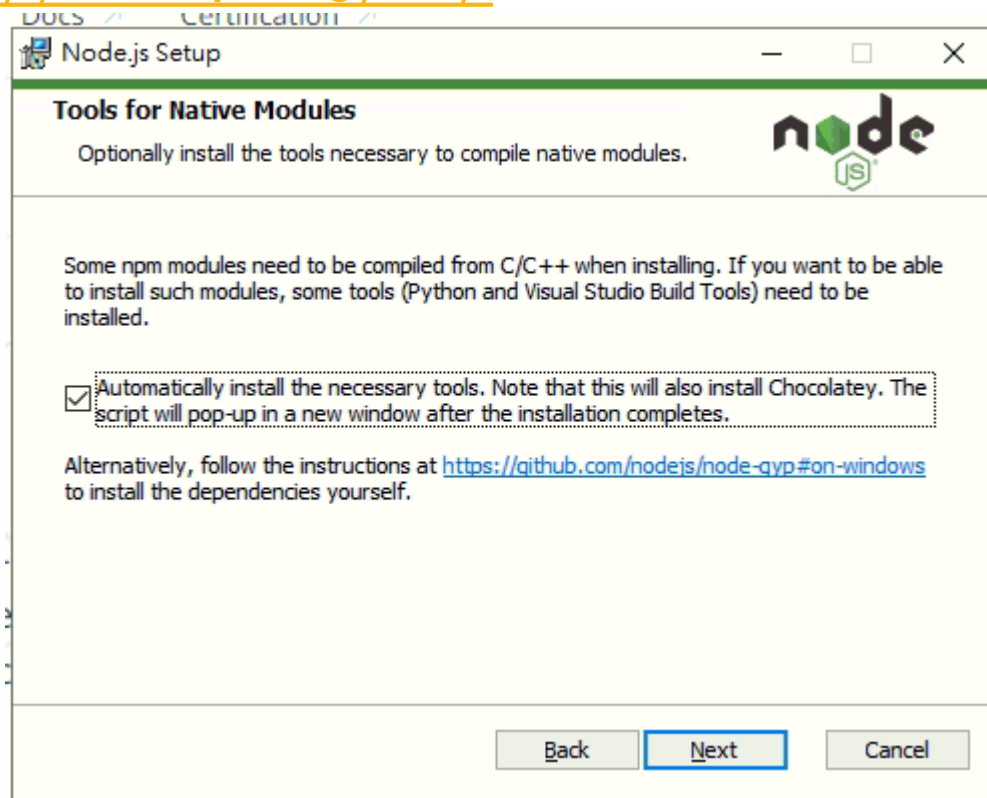
NodeRED應用-安裝NodeRED

- 安裝程式
 - 先安裝Node.js環境 (NodeRED底層使用Node.js)
 - <https://nodejs.org/en/>



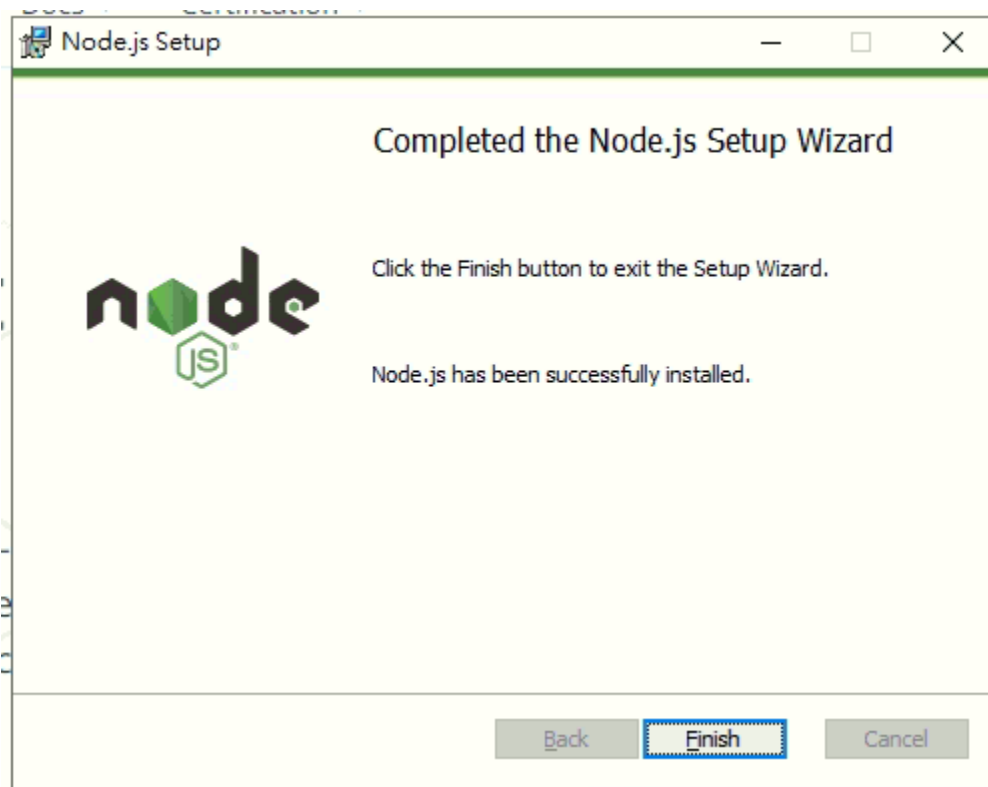
NodeRED應用-安裝NodeRED

- 安裝程式
 - 先安裝Node.js環境 (NodeRED底層使用Node.js)
 - <https://nodejs.org/en/>



NodeRED應用-安裝NodeRED

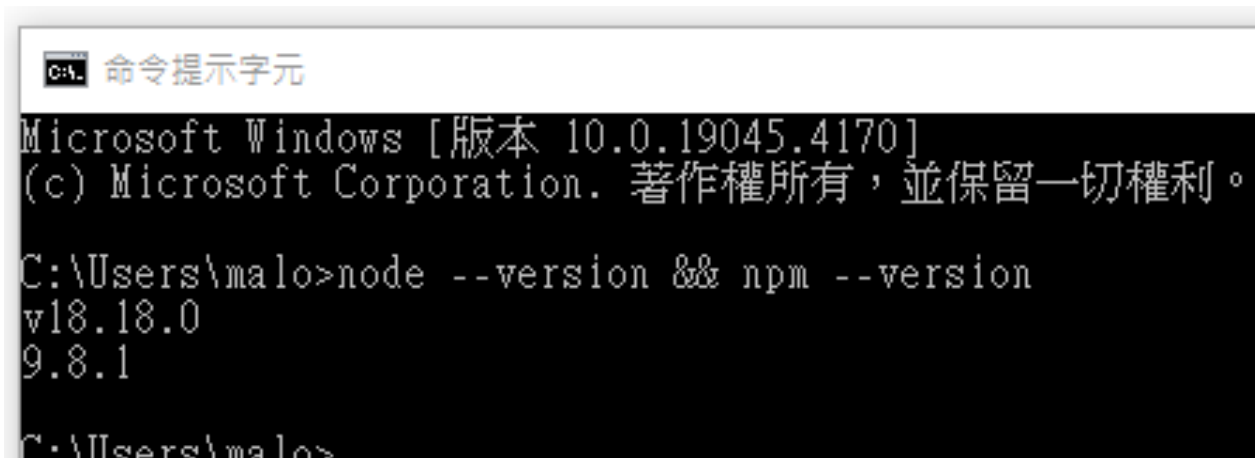
- 安裝程式
 - 先安裝Node.js環境 (NodeRED底層使用Node.js)
 - <https://nodejs.org/en/>



NodeRED應用-安裝NodeRED

□ 安裝程式

- ▣ 安裝完成後在console下指令【`node --version && npm --version`】，可以得到以下的回應



```
C:\> 命令提示字元

Microsoft Windows [版本 10.0.19045.4170]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

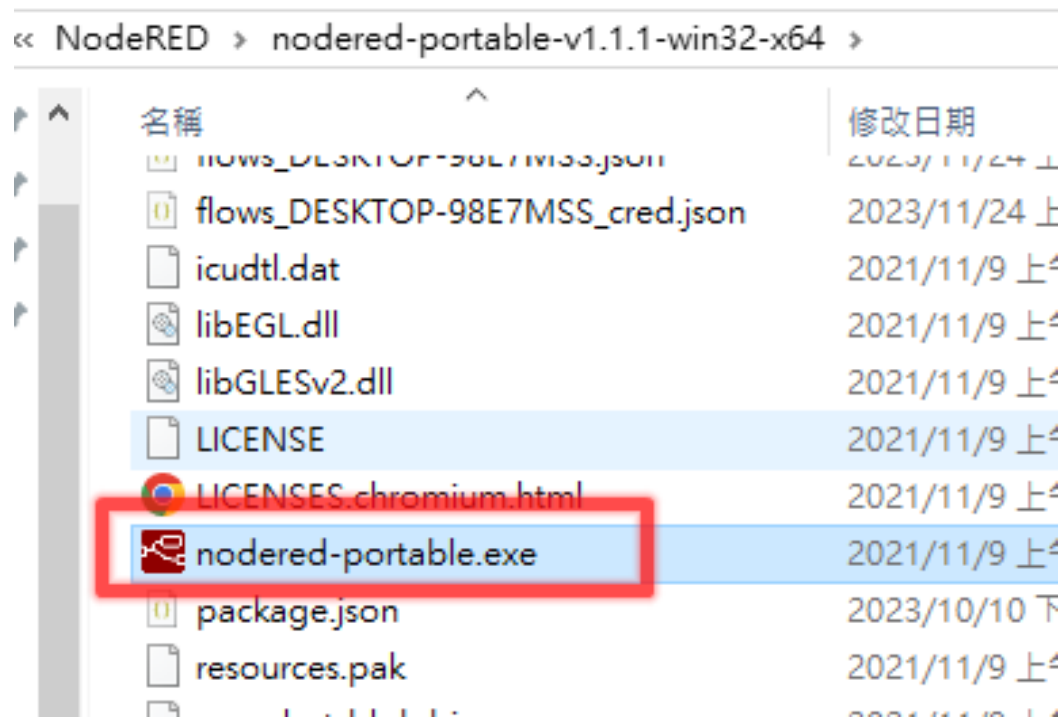
C:\Users\malo>node --version && npm --version
v18.18.0
9.8.1

C:\Users\malo>
```

執行NodeRED

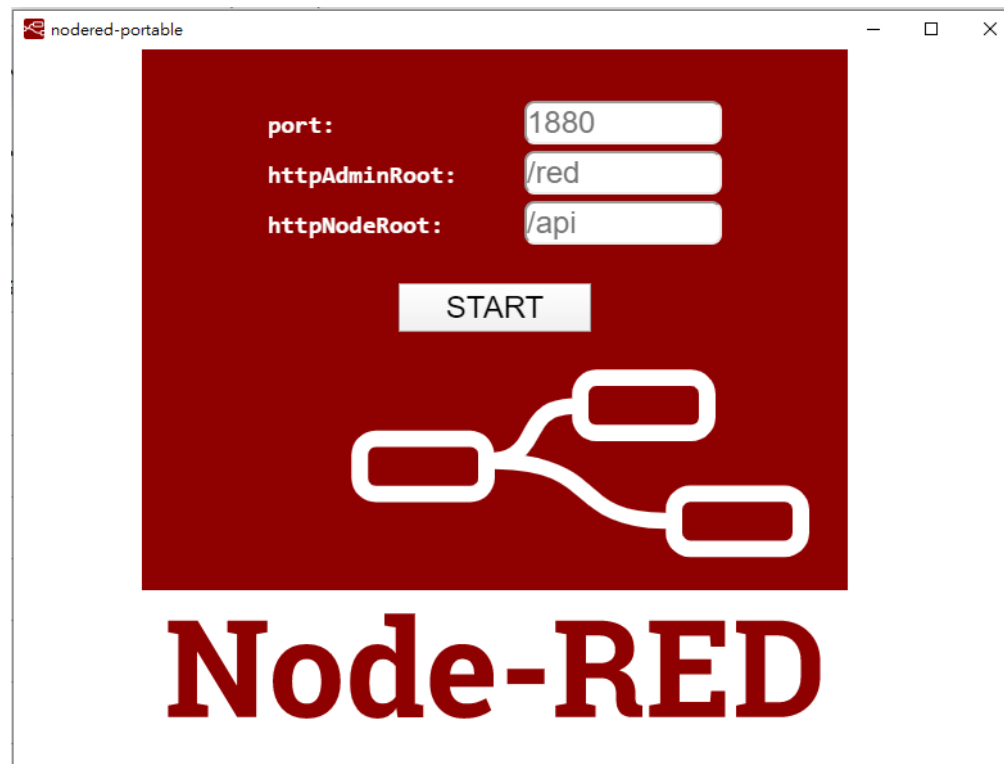
NodeRED應用 - 執行NodeRED

- 解壓縮nodered-portable打開程式nodered-portable.exe



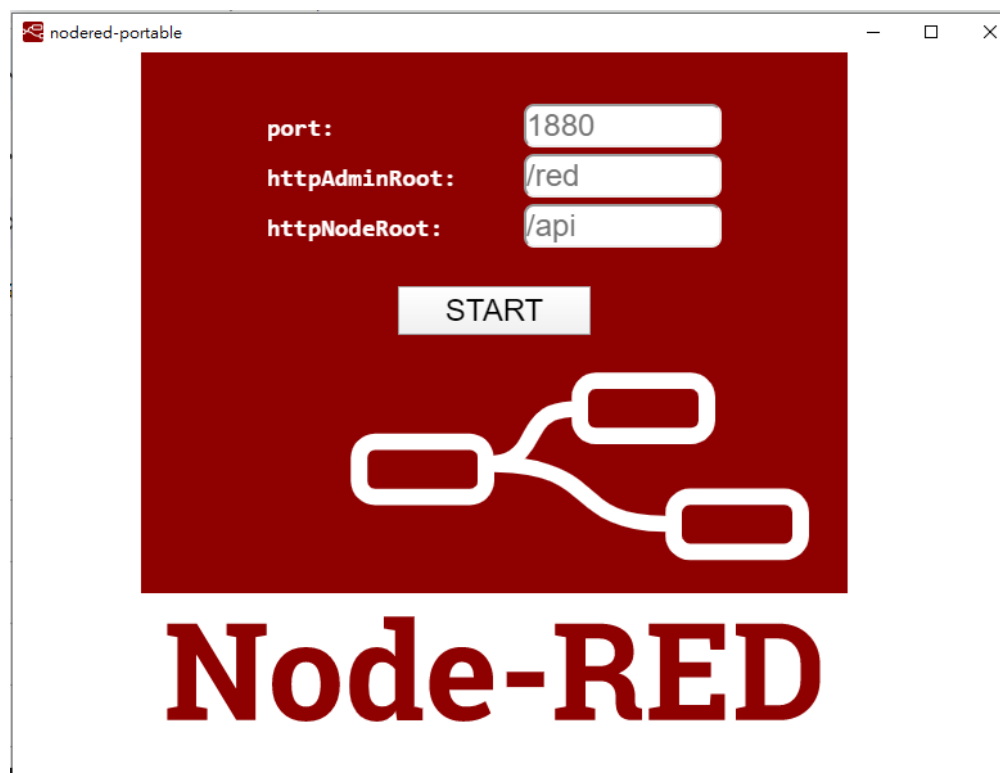
NodeRED應用 - 執行NodeRED

- 剛執行的時候會看到這樣的畫面, 點START就可以開始用



NodeRED應用 - 執行NodeRED

- 其中port代表目前Node-RED的Server是使用1880
- 開發用的編輯畫面在 <http://localhost:1880/red> 這邊



NodeRED應用 - 執行NodeRED

- 由右邊的menu可以看到這是V2.1.3版的 Node-RED





基礎操作

基礎操作

The image shows the Node-RED web interface. On the left, a vertical panel labeled "功能節點" (Function Nodes) contains various nodes like inject, debug, complete, catch, status, link in, link call, link out, comment, function, and switch. The central workspace, labeled "流程圖繪製區" (Flow Diagram Canvas), is a large grid area for building flows. On the right, the "屬性設定區" (Property Settings Panel) displays settings for the selected flow, "流程1" (Flow 1), including its ID and a search bar. A tip at the bottom of the right panel states: "按住 ctrl 的同時點擊工作介面可以在節點的對話欄中快速添加節點。" (Pressing ctrl while clicking the workspace can quickly add nodes to the node's dialog bar).

功能節點

流程圖繪製區

屬性設定區

Node-RED

部署

信息

搜索流程

流程

流程1

子

全

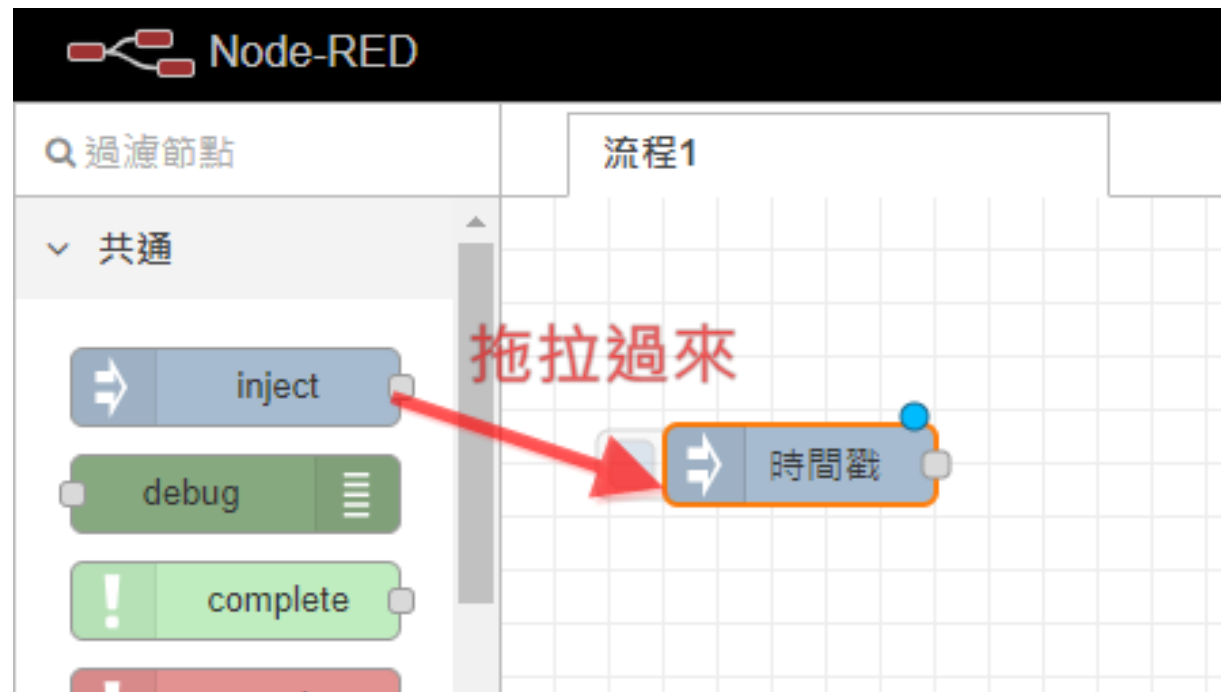
流程1

流程 "1a7255f0b04cc81d"

按住 ctrl 的同時點擊工作介面可以在節點的對話欄中快速添加節點。

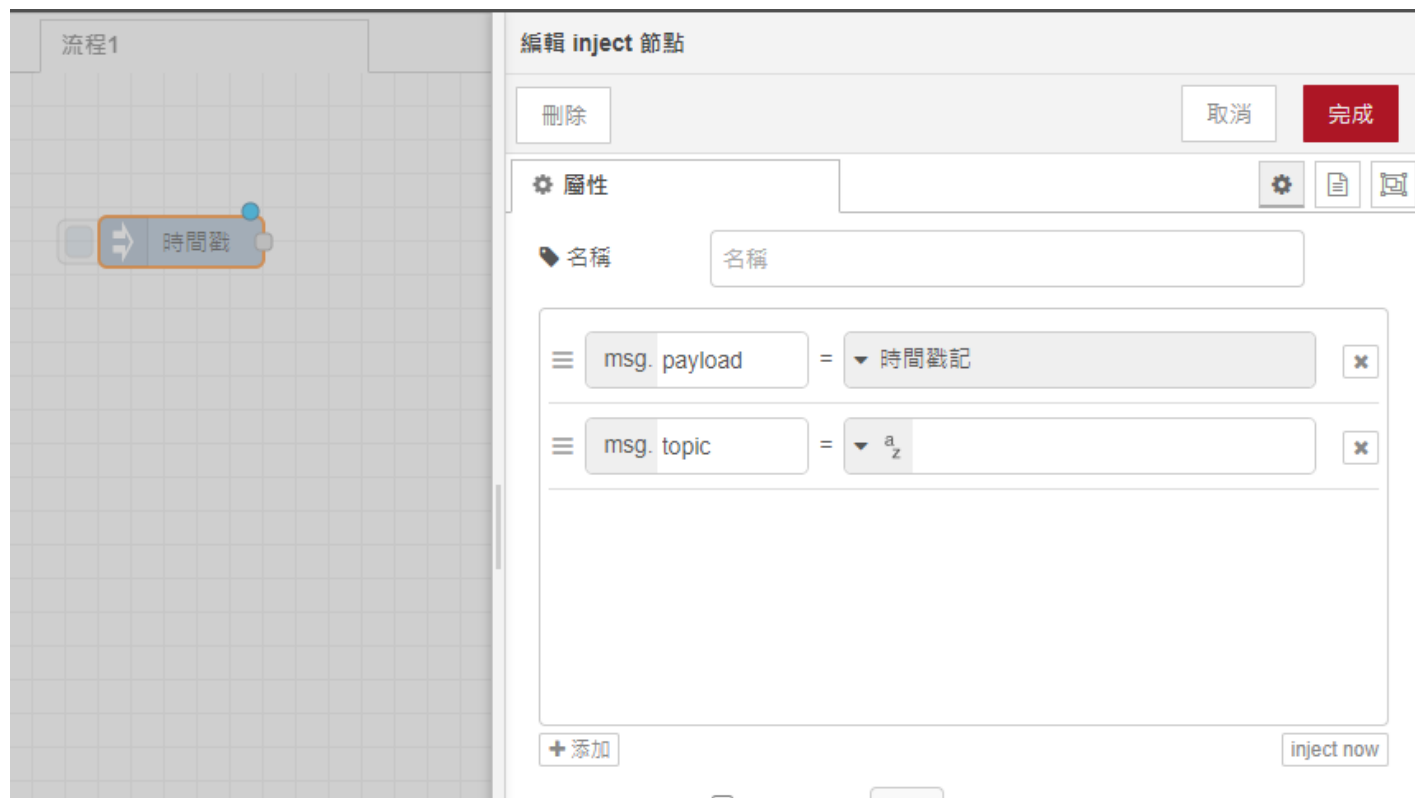
基礎操作

- 首先從左邊拖拉一個node進入中間的「流程」區塊



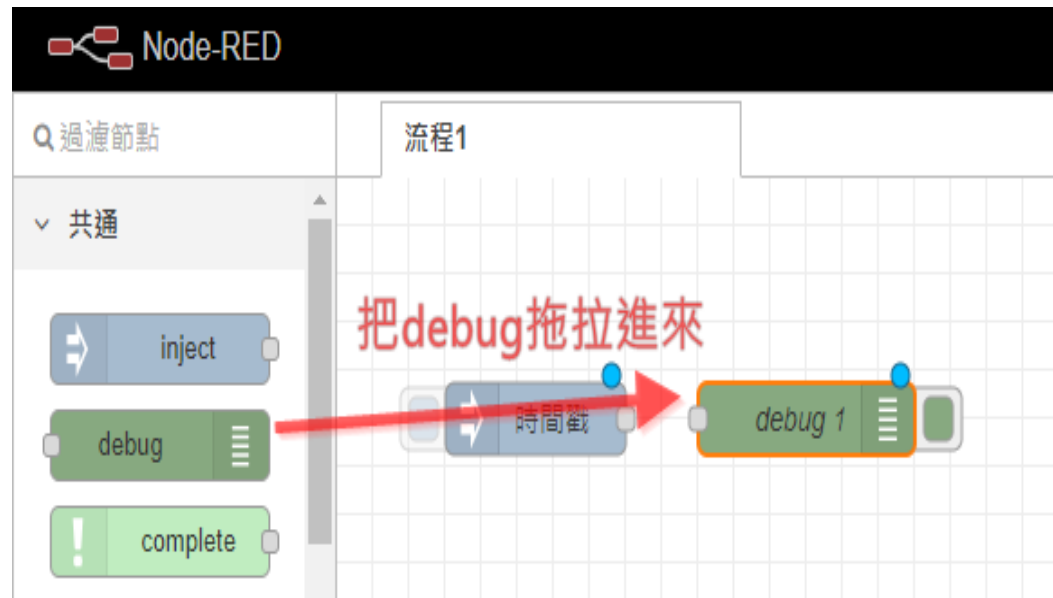
基礎操作

- 首在剛剛的node上點二下
- 可以看到編輯node的設定頁
- 預設這個node會輸出一個目前時間的資訊



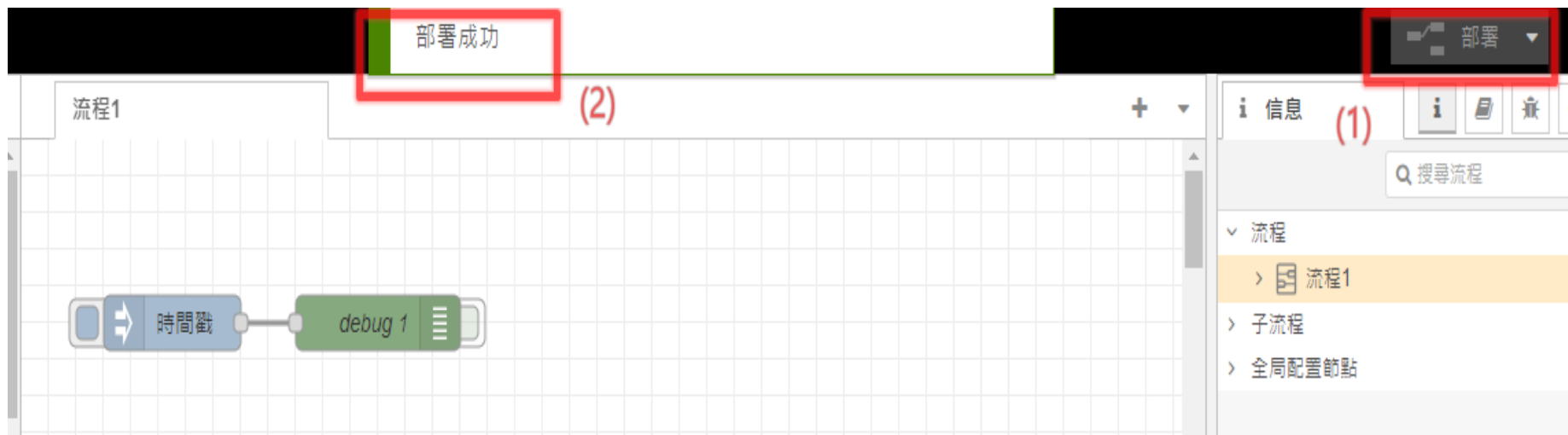
基礎操作

- 拖拉debug node進來
- 再把第一個node的右邊小點點接到debug node的左邊小點點



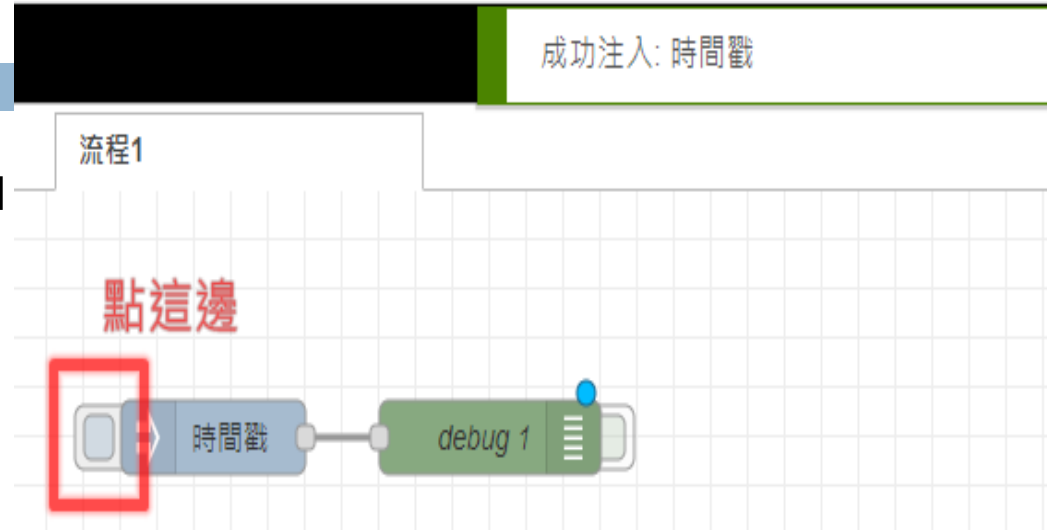
基礎操作

- ✚ 點一下右上角的「部署」，英文是「**deploy**」
- ▣ 看到部署成功，代表我們目前的流程程式沒有問題

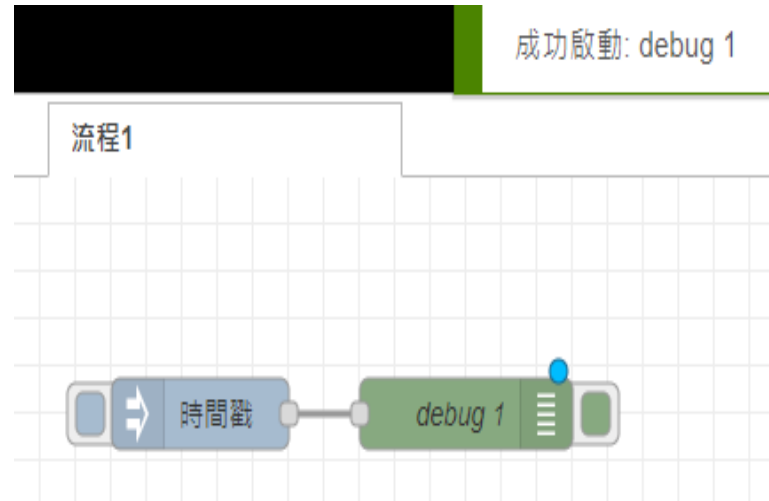


基礎操作

- 點一下「時間戳」的輸入觸發點，如右圖，會發現除了有訊息「成功注入：時間戳」以外，沒發生什麼事

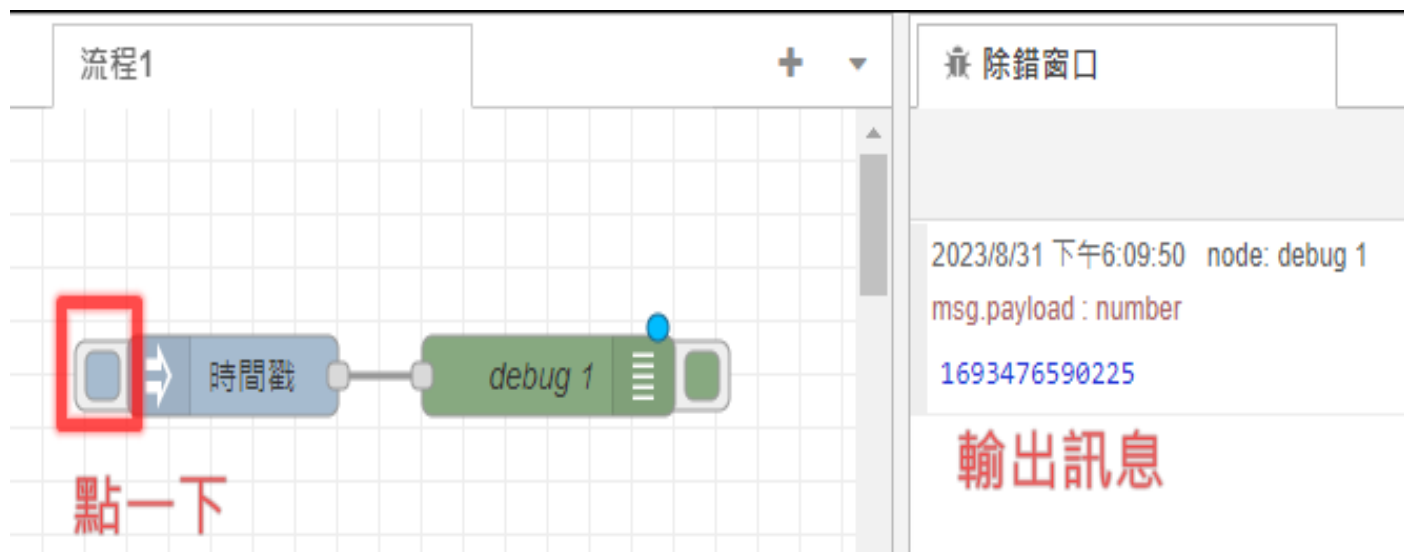


- 再點一下 debug node 的右邊的節點，啟用輸出功能



基礎操作

- 點一下最左邊的節點，此為輸入訊息的「觸發」點，點一下就會輸出訊息到debug1去
- 因為debug1已啟用，所以可以在除錯窗口看到訊息



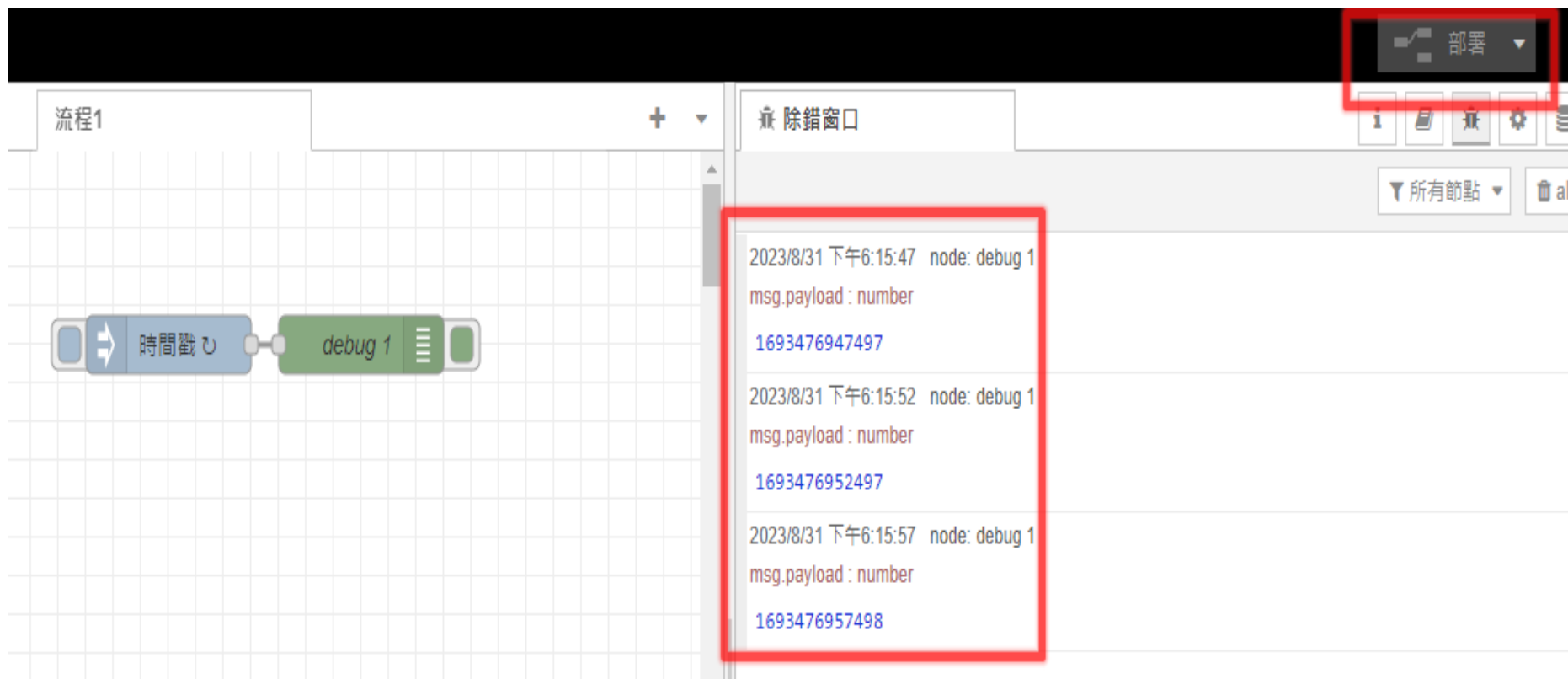
基礎操作

- 如果我們是想要定時觸發，自動一直輸出訊息，可以點二下「時間戳」
- 設定此node訊息觸發為「週期性執行」，5秒一次
- 再點一下完成



基礎操作

- ✚ 因為有改變設定，所以再點一下「部署」，就可以見到如下的訊息每5秒出現一個(這邊的數字是ms，所以每一個訊息增加5000ms)





Dashboard

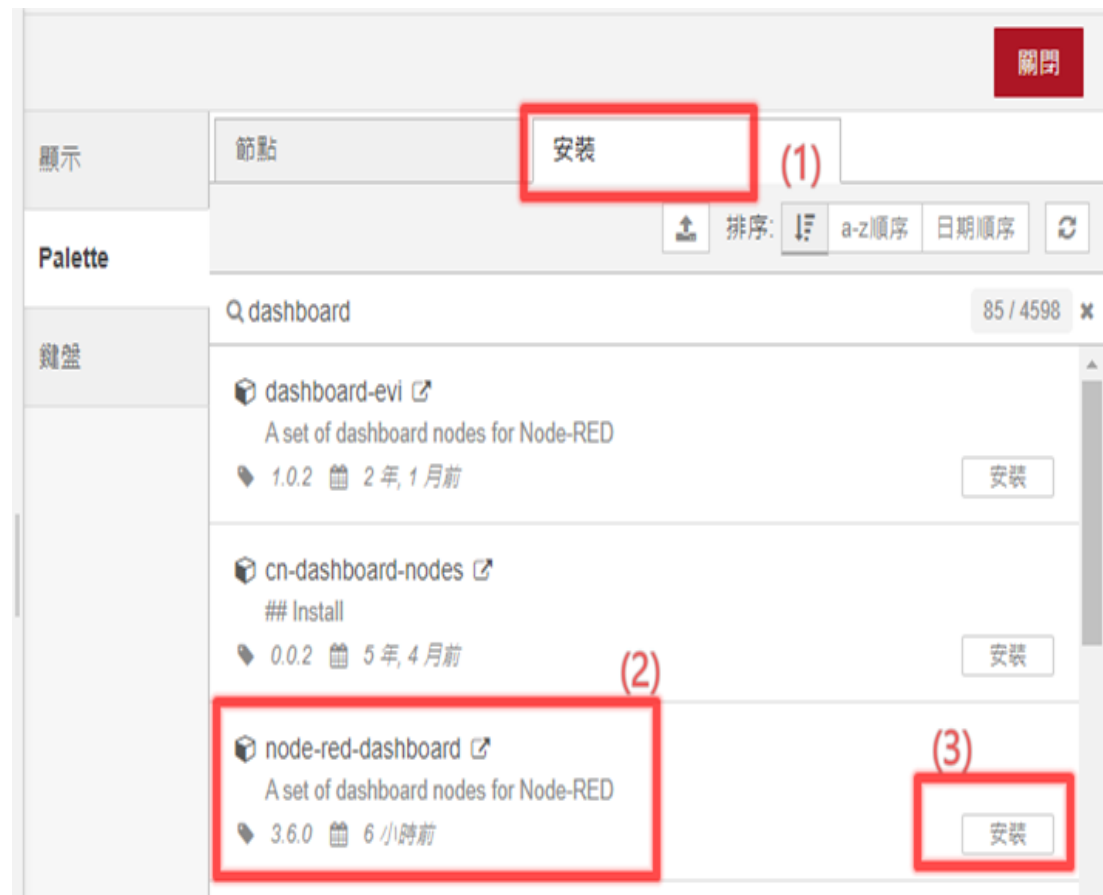
Dashboard

- 點右上角的三條線，可以看到一個選單
- 點選「節點管理」



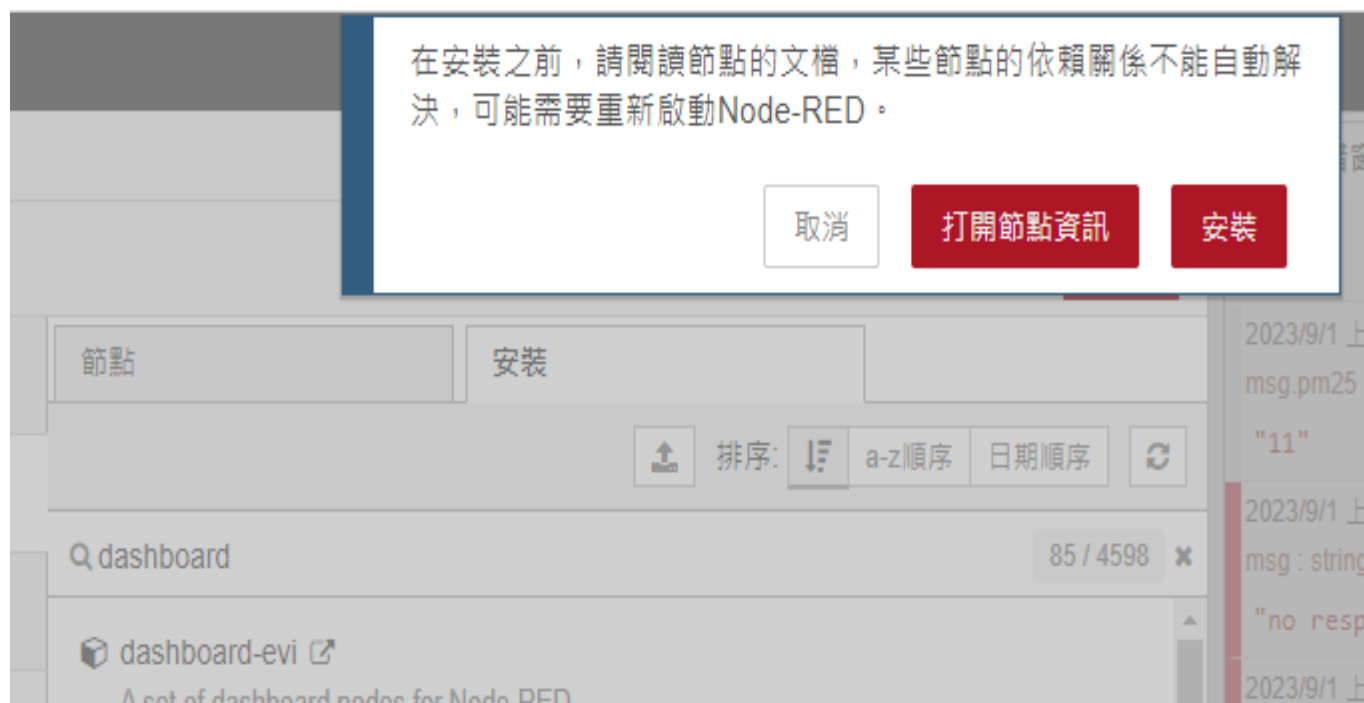
Dashboard

- 選擇「安裝」分頁
- 點選安裝 node-red-dashboard



Dashboard

□ 選擇「安裝」



Dashboard

- ❑ 安裝後的訊息
- ❑ 及新增的節點

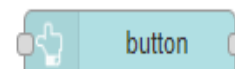
添加到面板中的節點:

- ui_base
- ui_button
- ui_dropdown
- ui_switch
- ui_slider
- ui_numeric
- ui_text_input
- ui_date_picker
- ui_colour_picker
- ui_form
- ui_text
- ui_gauge
- ui_chart
- ui_audio
- ui_toast
- ui_ui_control
- ui_template
- ui_link
- ui_tab
- ui_group
- ui_spacer

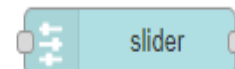
Node-RED

Q 過濾節點

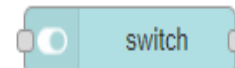
▼ dashboard



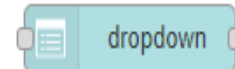
button



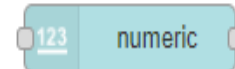
slider



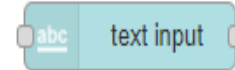
switch



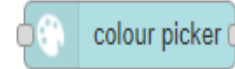
dropdown



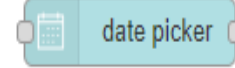
numeric



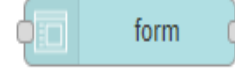
text input



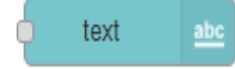
colour picker



date picker

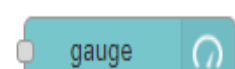


form



text

chart



gauge

Dashboard

- 我們點一下右上同的「新增流程」，來增加一個流程分頁



Dashboard

- 在「流程3」的分頁上連點二下
- 會出現名稱設定的頁面
- 我們把它修改為「儀表板測試」

編輯流程: 流程3

刪除 取消 完成

⚙ 屬性

📌 名稱 儀表板測試

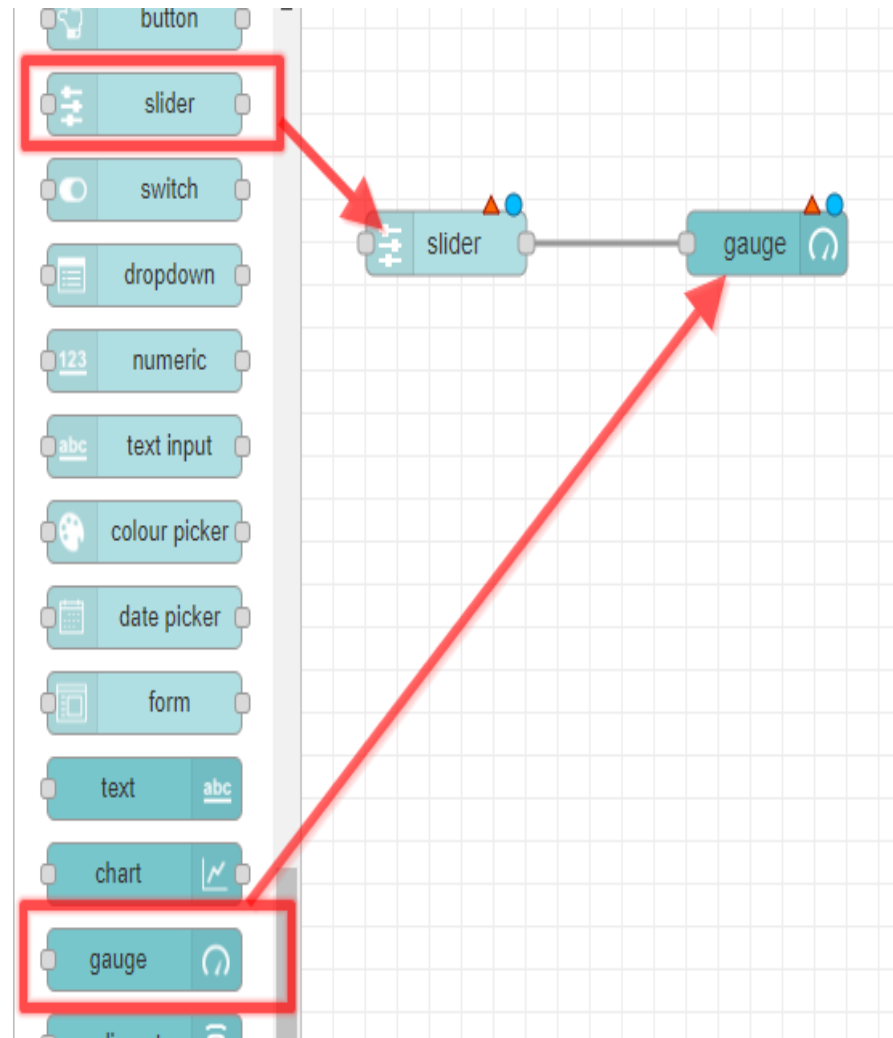
詳細描述

h1 h2 h3 B I </> [List Icon] [List Icon] [Quote Icon] [Link Icon] [Link Icon]

1

Dashboard

- ▣ 拖拉一組slider，
gauge 作為輸入，
輸出的介面



Dashboard

- ▣ 接下來要對dashboard的畫面作配置
- ▣ 要記得在dashboard中是以這三個分層次的：
 - ▣ Tab
 - ▣ Group
 - ▣ 元件
- ▣ Tab下有Group，Group下有元件

Dashboard

- 第一次進來設定slider
需要先設定group

編輯 slider 節點

刪除 取消 完成

屬性

Group [TEST] group1

Size auto

Label slider

Tooltip optional tooltip

Range min 0 max 10 step 1

Output continuously while sliding

→ If msg arrives on input, pass through to output: ☒

Dashboard

- 設定Group前要先設定Tab (新增一個)

編輯 slider 節點 > 編輯 dashboard group 節點

刪除 取消 更新

屬性

Name group1

Tab TEST

Class Optional CSS class name(s) for widget

Width 6

☒ Display group name

☐ Allow group to be collapsed

Dashboard

- 如圖，新增一個Tab名稱

編輯 slider 節點 > 編輯 dashboard group 節點 > 添加新的 dashboard tab 配置

取消 添加

屬性

Name TEST

Icon dashboard

State ☒ Enabled

Nav. Menu ☒ Visible

The **Icon** field can be either a Material Design icon (e.g. 'check', 'close')

Dashboard

- 接下來設定 Gauge 就比較輕鬆，直接選擇 [TEST]group1 就可以

編輯 gauge 節點

刪除 取消 完成

屬性

Group 添加新的 dashboard group 節點 [TEST] group1 添加新的 dashboard group 節點

Size

Type Gauge

Label gauge

Value format {{value}}

Units units

Range min 0 max 10

Colour gradient

Sectors 0 ... optional ... optional ... 10

Fill gauge from centre. ☐

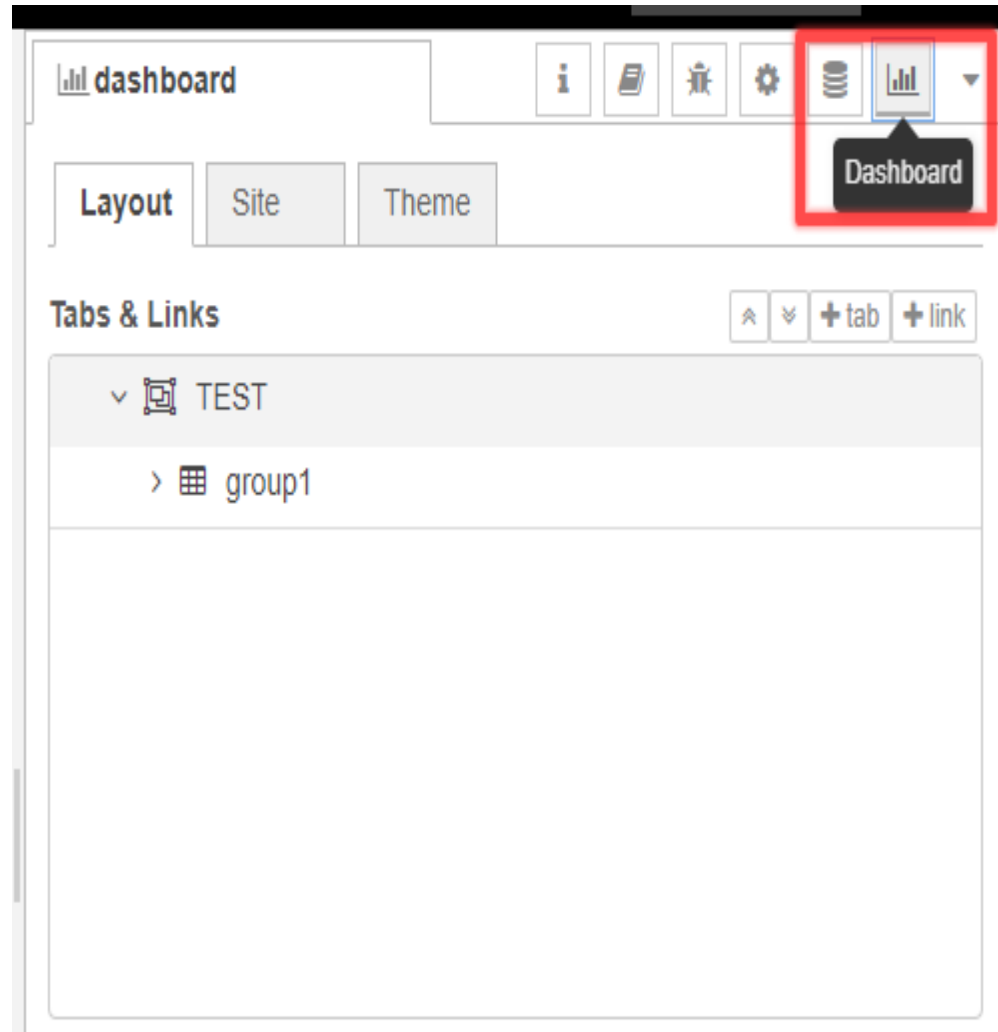
Dashboard

- 完成後，點選「部署」



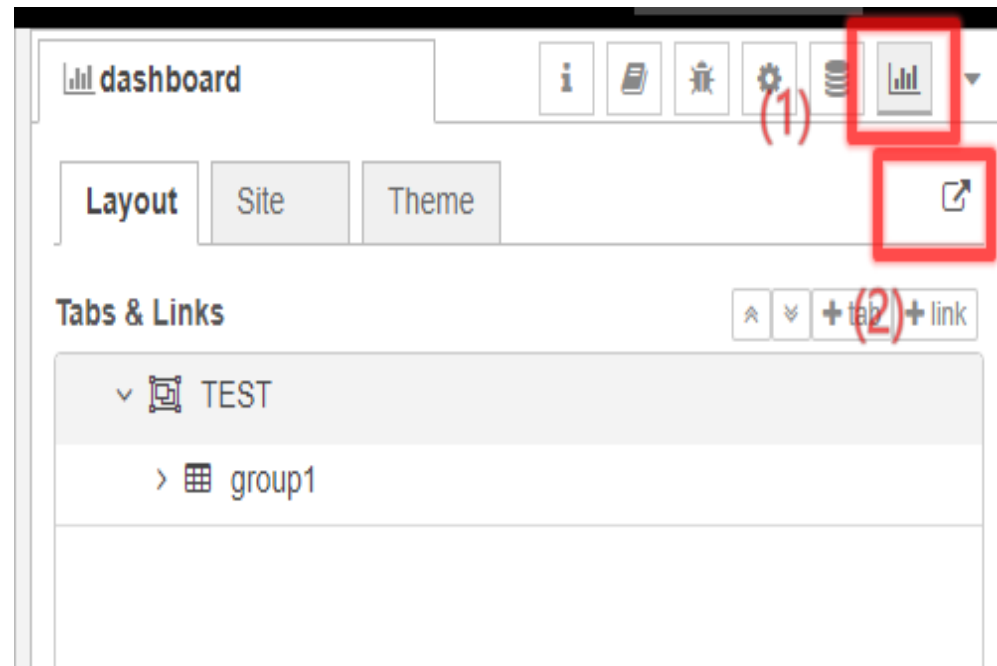
Dashboard

- 再點選右上角的「Dashboard」icon



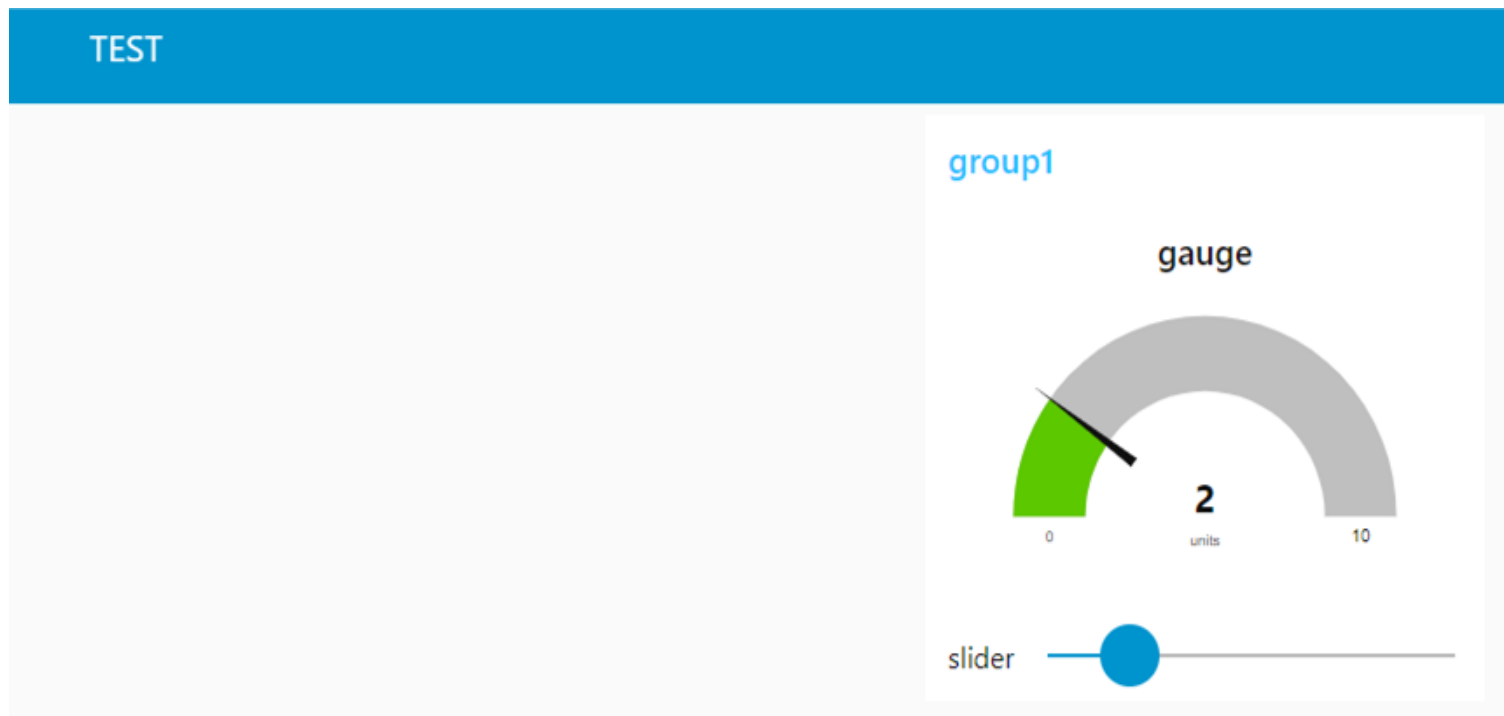
Dashboard

- 點選右上角的「Dashboard」icon
- 再點選下方的彈出按鈕，就會跳出dashboard的畫面



Dashboard

- 可以拖拉slider去影響上面的gauge的數值



Dashboard的排版

Dashboard

- 如右圖，先拖拉出如右的項目
- 三個輸入 slider
- 三個Gauge
- 三個chart

The screenshot shows the Node-RED web interface. On the left, the 'palette' contains various widgets. Three sliders are being dragged from the palette to the workspace. In the workspace, three input nodes (labeled '輸入1', '輸入2', '輸入3') are connected to three gauge nodes (labeled '表頭一', '表頭二', '表頭三'). The '表頭二' node is highlighted with a dashed orange border. On the right, the 'dashboard' sidebar is visible, showing a 'TEST' tab with a list of widgets. The list includes '輸入控制區' (containing '輸入1', '輸入2', '輸入3') and '輸出儀表區一' (containing '表頭一', '表頭二', '表頭三'). The '表頭二' widget is highlighted in the list. Below the list, the text '上下拖拉決定順序' (Order determined by up and down dragging) is displayed.

Node-RED

流程1 儀表板測試

dashboard

Layout Site Theme

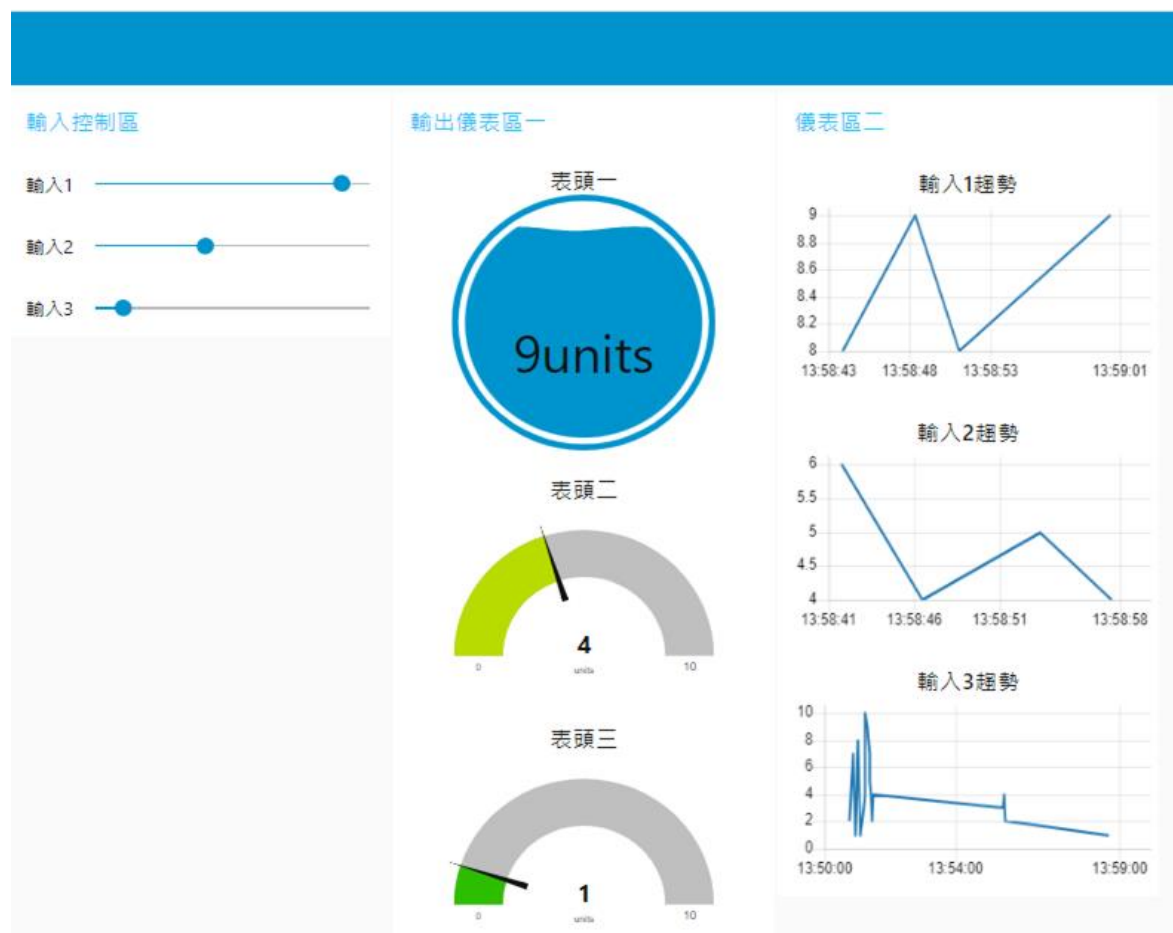
Tabs & Links

- TEST
 - 輸入控制區
 - 輸入1
 - 輸入2
 - 輸入3
 - 輸出儀表區一
 - 表頭一
 - 表頭二
 - 表頭三
 - 儀表區二
 - 輸入1趨勢
 - 輸入2趨勢
 - 輸入3趨勢

上下拖拉決定順序

Dashboard

- 輸出成果如右圖



Dashboard

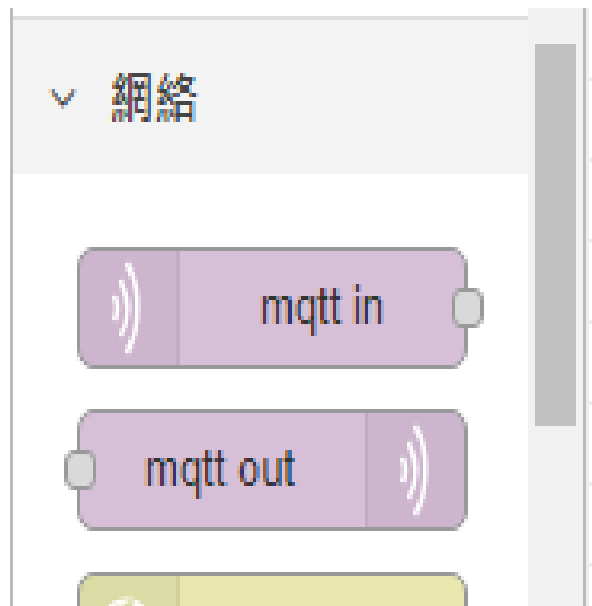
✚ Layout的排版介面如右



MQTT的應用

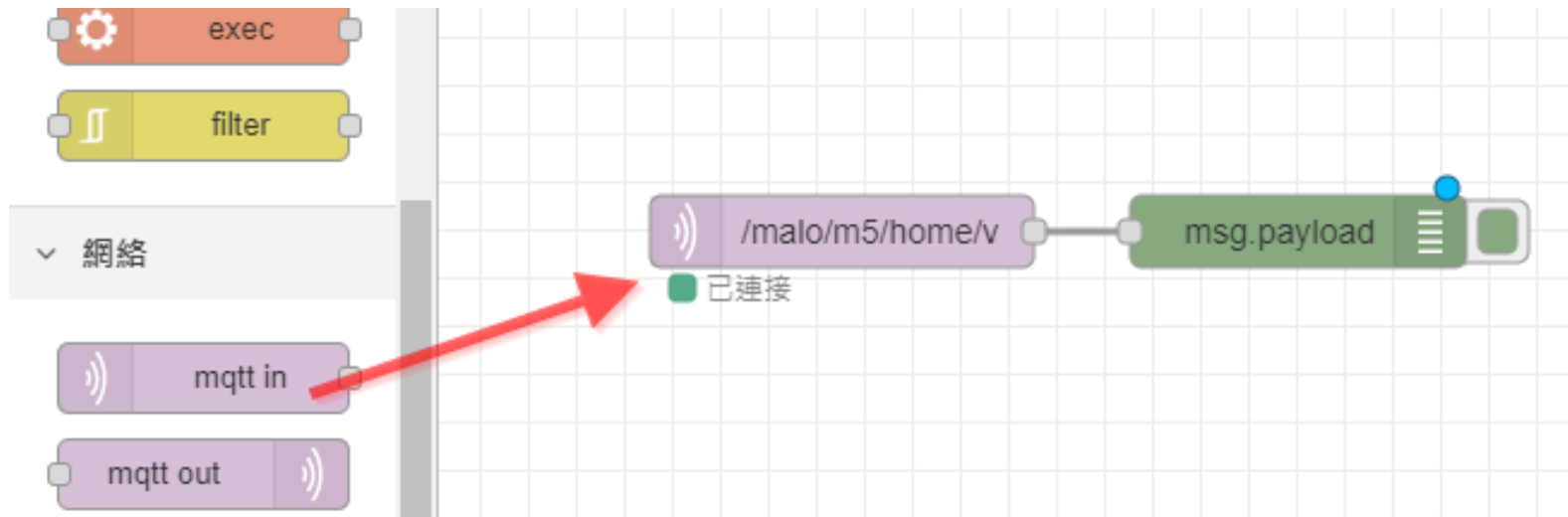
MQTT應用

- MQTT為Node-RED內建就有的節點服務



MQTT應用

□ 加入節點



MQTT應用

- ❑ MQTT的設定如右
- ❑ 主題為 topic
- ❑ QoS一般用1即可
- ❑ 名稱是節點要顯示的文字
- ❑ 服務端為Broker的位置這個需要按一下紅色框選處進行新增

The screenshot shows the '編輯 mqtt in 節點' (Edit MQTT in node) configuration window. Red boxes and arrows highlight key settings: the '完成' (Done) button, the '服務端' (Broker) field with value 'broker.hivemq.com:1883', the '主題' (Topic) field with value '/malo/m5/home/v', and the 'QoS' field with value '1'. The 'Action' is set to 'Subscribe to single topic' and the '輸出' (Output) is set to '自動檢測 (字符串或buffer)'.

屬性	值
刪除	取消 完成
服務端	broker.hivemq.com:1883
Action	Subscribe to single topic
主題	/malo/m5/home/v
QoS	1
輸出	自動檢測 (字符串或buffer)
名稱	名稱

MQTT應用

□ 設定如圖所示

編輯 mqtt in 節點 > 編輯 mqtt-broker 節點

刪除 取消 更新

屬性

名稱 名稱

連接 安全 消息

服務端 broker.hivemq.com 埠 1883

☒ Connect automatically
☐ 使用 TLS

Protocol MQTT V3.1.1

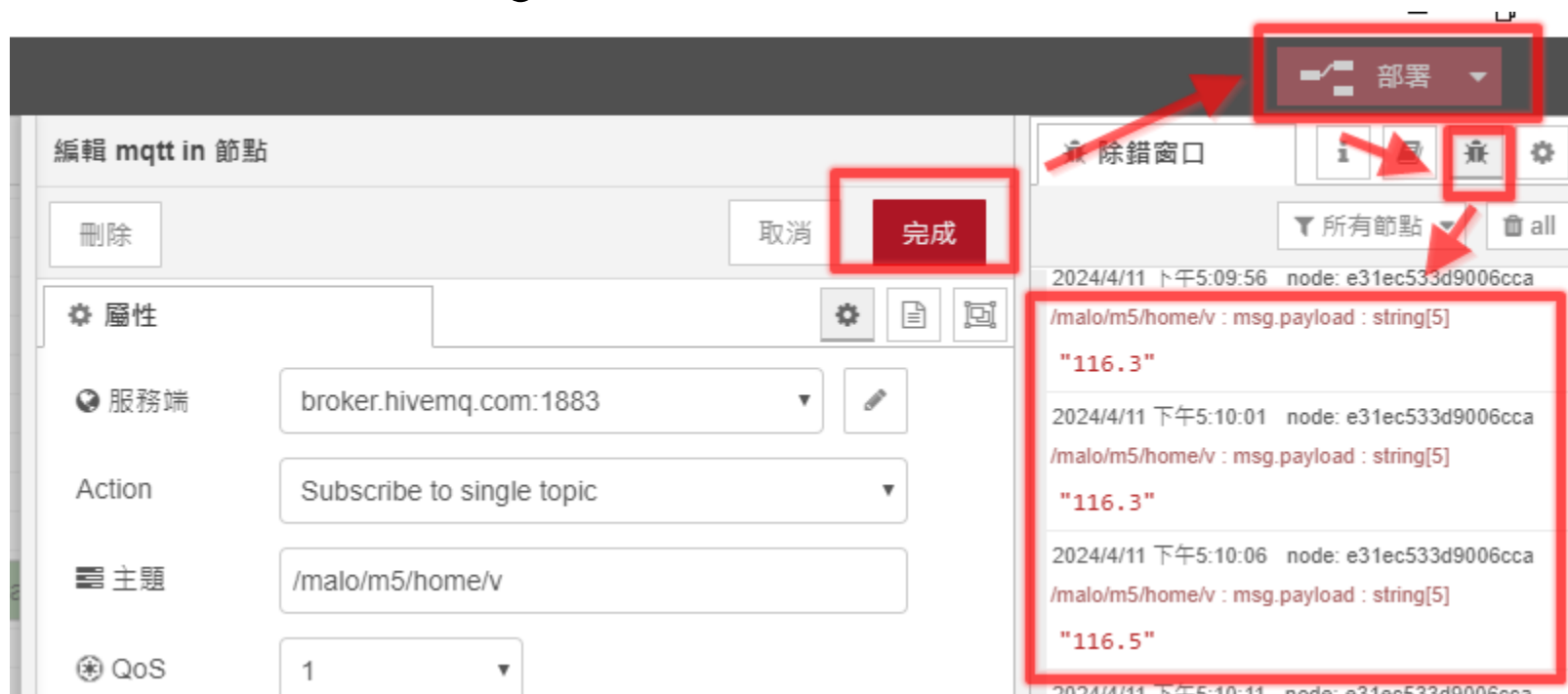
使用者端ID 留白則自動隨機生成

Keepalive計時(秒) 60

Session ☒ 使用新的會話

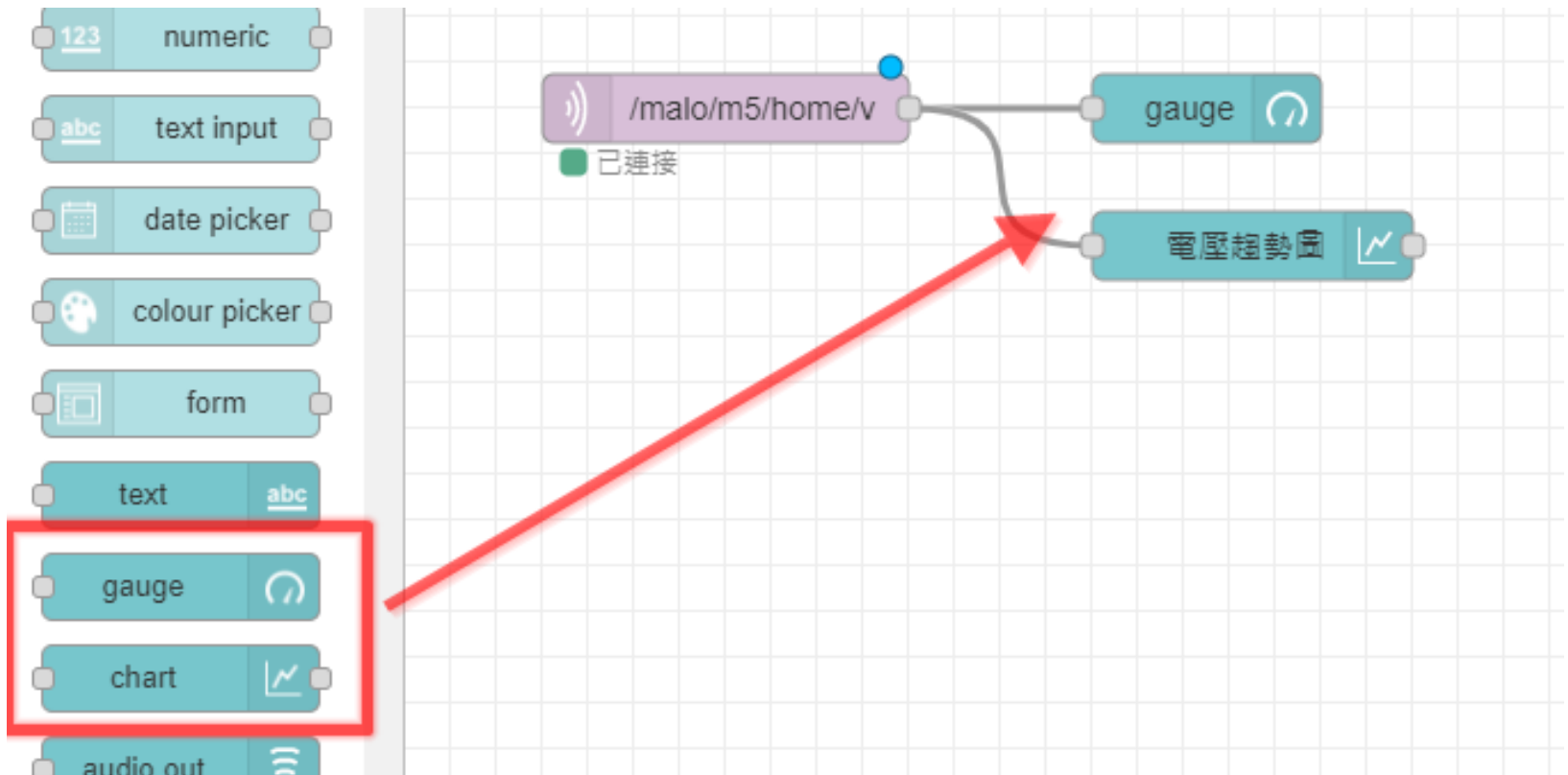
MQTT應用

- 點一下【完成】
- 部屬專案
- 可以在debug訊息看到電壓資訊



MQTT應用

- 加入Gauge和Chart，如下圖



MQTT應用

- 進入Gauge的編輯畫面
- Range填入100，200, 因為我們的電壓是110V的
- 點一下新增Group

編輯 gauge 節點

刪除 取消 完成

⚙ 屬性

Group [Home] 電力

Size auto

Type Gauge

Label gauge

Value format {{value}}

Units units

Range min 100 max 200

Colour gradient

Sectors 100 ... optional ... optional ... 200

MQTT應用

- 新增的Group設定如右

編輯 gauge 節點 > 編輯 dashboard group 節點

刪除 取消 更新

⚙ 屬性 ⚙ 📄

📁 Name 電力

📁 Tab Home ▼ ✎

</> Class Optional CSS class name(s) for widget

↔ Width 6

☒ Display group name

MQTT應用

- 回到Gauge按「完成」

編輯 gauge 節點

刪除 取消 完成

⚙ 屬性

Group [Home] 電力

Size auto

Type Gauge

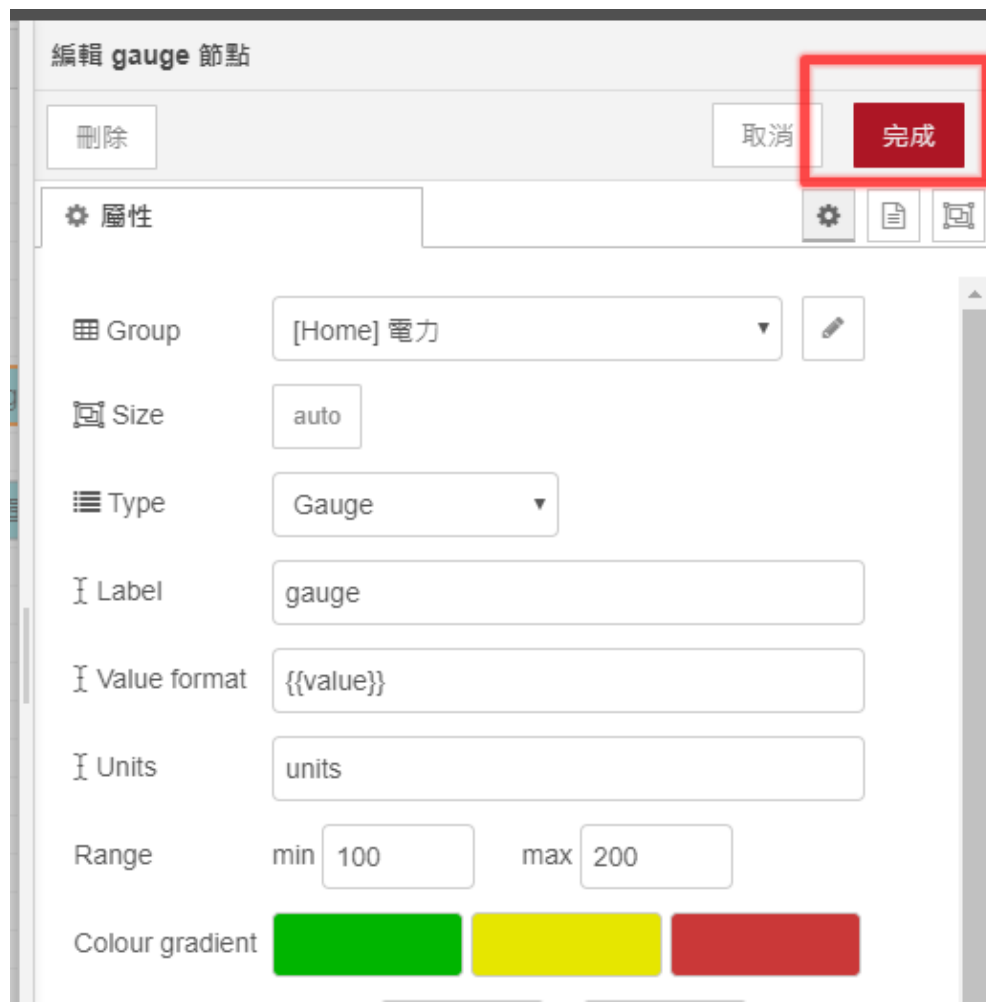
Label gauge

Value format {{value}}

Units units

Range min 100 max 200

Colour gradient



MQTT應用

- 進入chart設定頁面
- 按完成即可

編輯 chart 節點

刪除 取消 完成

⚙️ 屬性

Group [Home] 電力

Size auto

Label 電壓趨勢圖

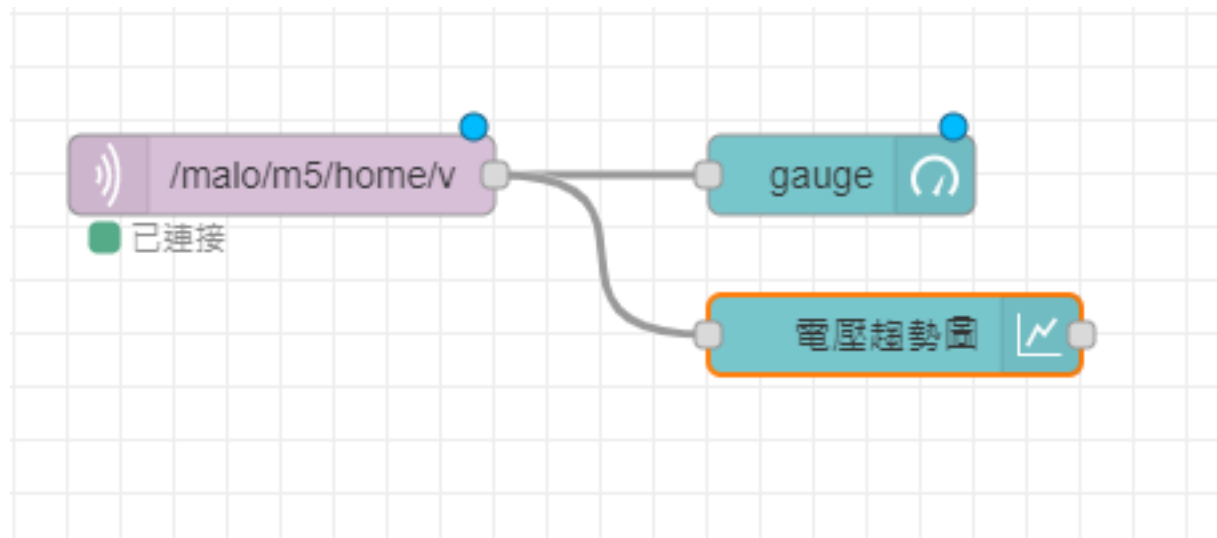
Type Line chart ☐ enlarge points

X-axis last 1 hours OR 1000 points

X-axis Label HH:mm:ss ☐ as UTC

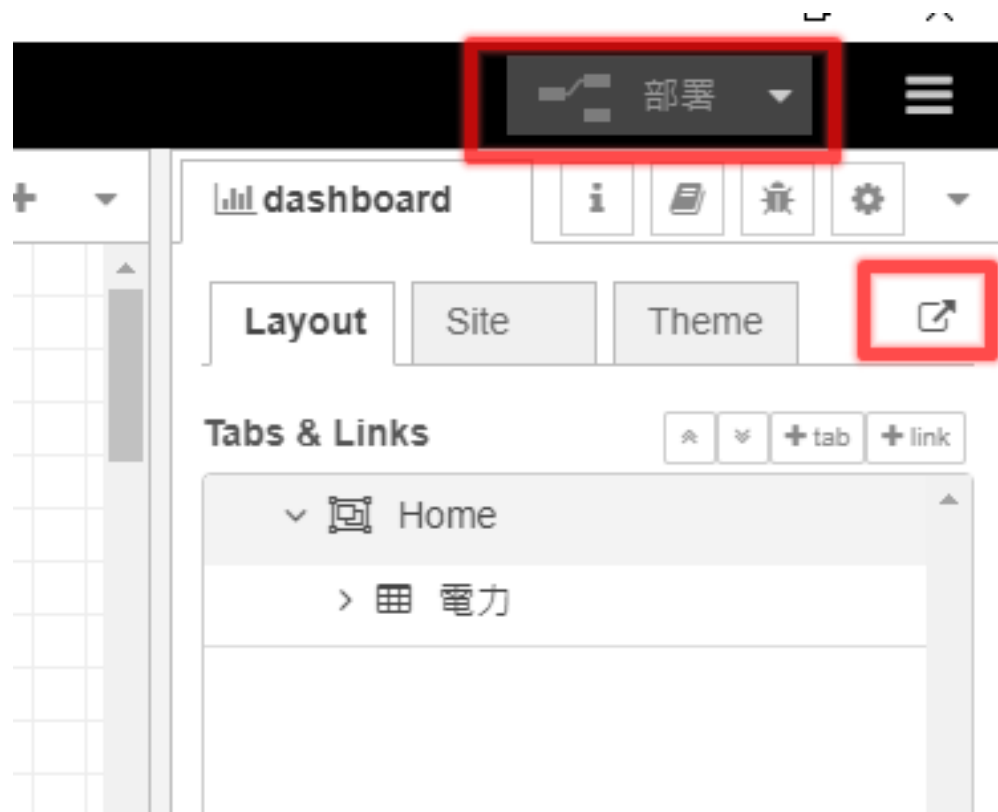
MQTT應用

- 把Node連結如下



MQTT應用

- 點「部署」
- 於dashboard分頁點選彈出鈕，進入dashboard的網頁



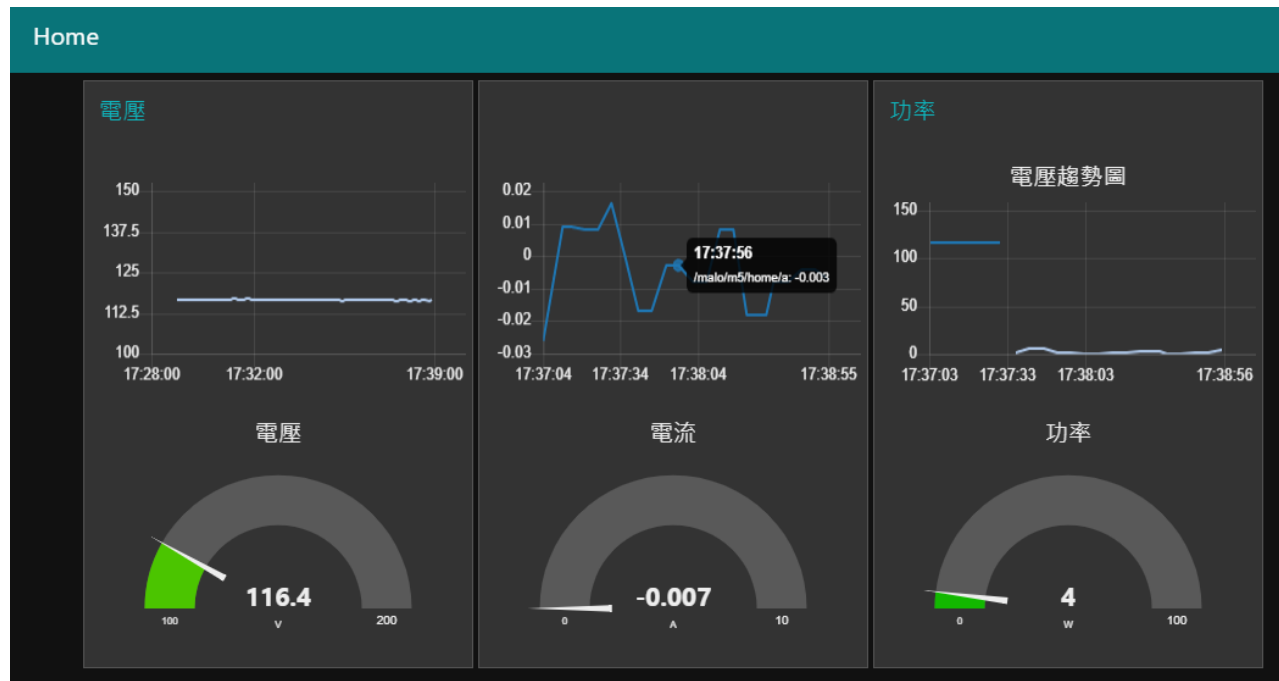
MQTT應用

- 整個設定完，我們可以得到這樣的畫面



MQTT應用

- 試著把電壓、電流、功率都加進來
- `/malo/m5/home/v`
- `/malo/m5/home/a`
- `/malo/m5/home/w`



MQTT應用

- 試著把電壓、電流、功率都加進來，並調整順序



MQTT應用2: 牧場電表資訊

- ❑ 牧場電表資訊的Topic如下，請試著收集資訊，並將其視覺化:
- ❑ /malo/farm/v1
- ❑ /malo/farm/a1
- ❑ /malo/farm/v2
- ❑ /malo/farm/a2
- ❑ /malo/farm/v3
- ❑ /malo/farm/a3
- ❑ /malo/farm/total_e

