# Lecture 7. NodeRED

2143399 ICE CAPSTONE

03.18.2019

# Class Schedule after the Mid-term

| | |
|---|---|
| March 18 | Lecture 7 – Open IoT platform |
| March 25 | Lecture 8 – Open IoT platform and database |
| April 1 | Lecture 9 – Data analytics |
| **April 10 (Wednesday)** | **MS3 presentation** |
| April 15 | No class (Songkran day) |
| April 22 | Advanced data analytics and machine learning (guess lecture) |
| **May 1 (Wednesday)** | **MS4 presentation** |
| May 13 | TBD* |

# What is ‘NodeRED’

- Powerful tool for building IoT applications
- Simplify the task by ‘wiring together’ code blocks (or **‘nodes’**)
- By wiring input nodes, processing nodes and output nodes, this creates a ‘**flows**’
- Developed by IBM as an **open source** project (in late 2013)

# Today’s Objectives

- High-level understanding of Node-RED
- Usage scenarios of Node-RED
- Create a dashboard that displays data from multiple sensors/NodeMCU
- Understand MQTT
- Send data from Arduino to NodeRED via MQTT

# Hands-on:

**Workshop 1:** Install NodeRED

**Workshop 2:** Build your first flow

**Workshop 3:** Get weather data and send alert

**Workshop 4:** UI nodes from node-red-dashboard

**Workshop 5:** MQTT

**Workshop 6:** Arduino, MQTT and NodeRED***

# **Workshop 1.** Install NodeRED

# NodeRED Installation
# Step 1: Install Node.js

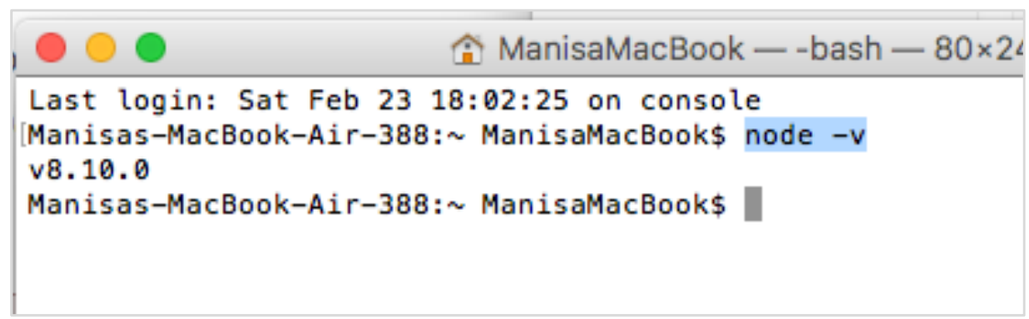
- Please have a working version of Node.js (Node.js **LTS 8.x** is recommended)
- If you need to install Node.js….we will install Node.js and NPM using Homebrew

Go to terminal and type:

- **brew install node**

Check node and npm version:

- **node -v**
- **npm -v**



# NodeRED Installation
# Step 2: Install NodeRED

- To install NodeRED, use the following command:

```
sudo npm install -g --unsafe-perm node-red
```

- Here is the installation instruction for Windows, Raspberry Pi, BeagleBone black

https://nodered.org/docs/getting-started/installation

# Running Node-RED

- Use command: node-red

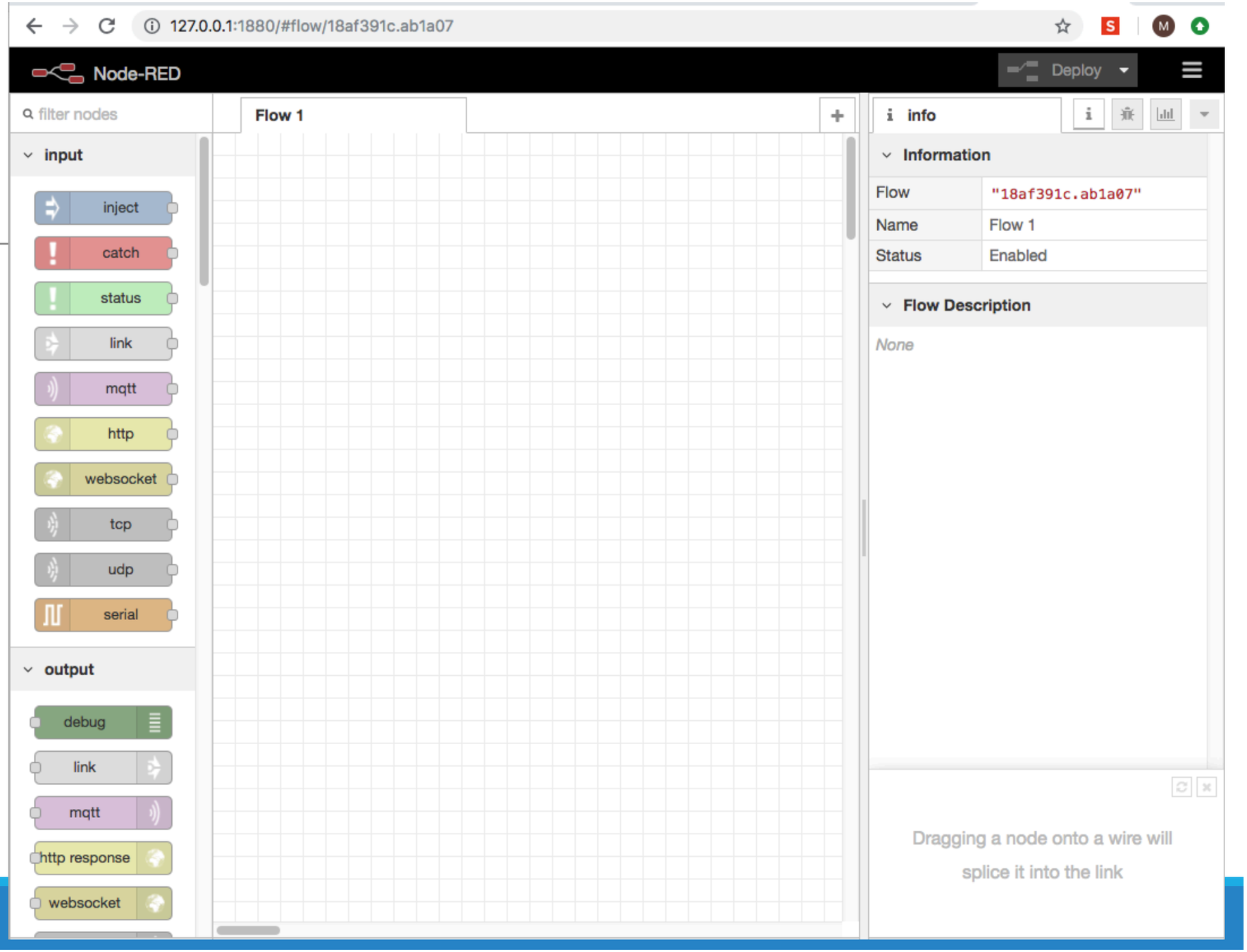
- Visit your server IP:

http://127.0.0.1:1880

ManisaMacBook — node-red NVM_RC_VERSION= — 80×45

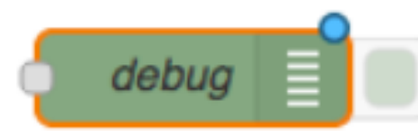
```
Last login: Sat Feb 23 18:02:25 on console
Manisas-MacBook-Air-388:~ ManisaMacBook$ node -v
v8.10.0
Manisas-MacBook-Air-388:~ ManisaMacBook$ node-red
1 Mar 06:02:11 - [info]

Welcome to Node-RED
===================

1 Mar 06:02:11 - [info] Node-RED version: v0.19.5
1 Mar 06:02:11 - [info] Node.js  version: v8.10.0
1 Mar 06:02:11 - [info] Darwin 16.7.0 x64 LE
1 Mar 06:02:11 - [info] Loading palette nodes
1 Mar 06:02:12 - [warn] rpi-gpio : Raspberry Pi specific node set inactive
1 Mar 06:02:13 - [info] Dashboard version 2.12.2 started at /ui
1 Mar 06:02:14 - [info] Settings file  : /Users/ManisaMacBook/.node-red/settings
.js
1 Mar 06:02:14 - [info] Context store  : 'default' [module=memory]
1 Mar 06:02:14 - [info] User directory : /Users/ManisaMacBook/.node-red
1 Mar 06:02:14 - [warn] Projects disabled : editorTheme.projects.enabled=false
1 Mar 06:02:14 - [info] Flows file     : /Users/ManisaMacBook/.node-red/flows_Ma
nisas-MacBook-Air-388.local.json
1 Mar 06:02:14 - [info] Creating new flow file
1 Mar 06:02:14 - [warn]

---------------------------------------------------------------------
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
---------------------------------------------------------------------

1 Mar 06:02:14 - [info] Starting flows
1 Mar 06:02:14 - [info] Started flows
1 Mar 06:02:14 - [info] Server now running at http://127.0.0.1:1880/
```

# NodeRED GUI


127.0.0.1:1880/#flow/18af391c.ab1a07
Node-RED
Deploy
filter nodes
Flow 1
info
input
inject
catch
status
link
mqtt
http
websocket
tcp
udp
serial
output
debug
link
mqtt
http response
websocket
Information
Flow
"18af391c.ab1a07"
Name
Flow 1
Status
Enabled
Flow Description
None
Dragging a node onto a wire will splice it into the link

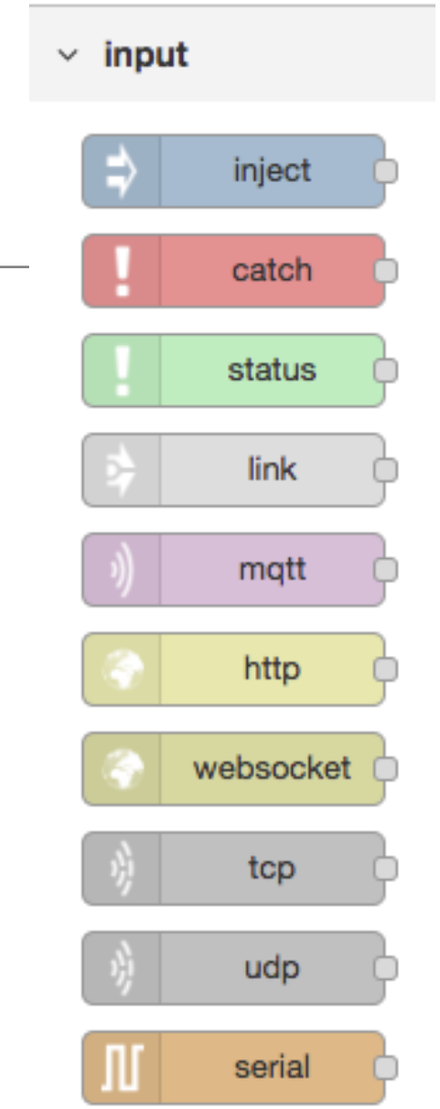
# Three Main Types of Nodes

- Input
- Output
- Processing



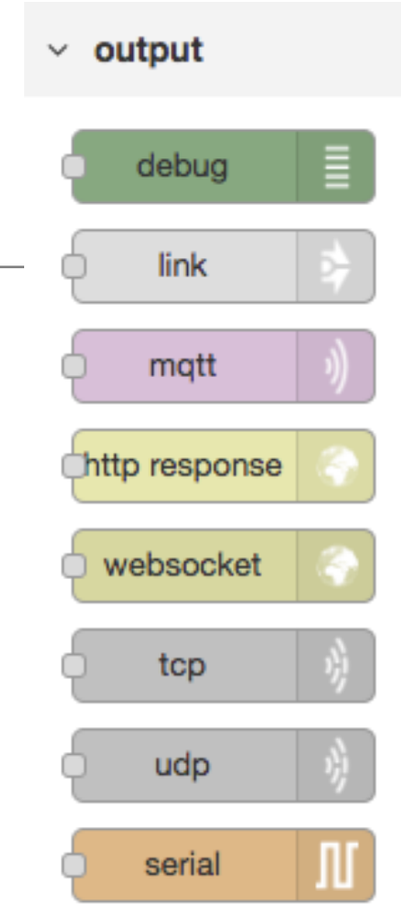
# Input Nodes

- Input nodes allow you to input data into a Node-RED application or “flow”
- They have at least one output endpoint represented by the small grey square only on their right side.
- Input nodes allow you to connect data from other services, like: Twitter, Google, serial, websockets or tcp nodes, or to manually input data into a flow using the inject node.



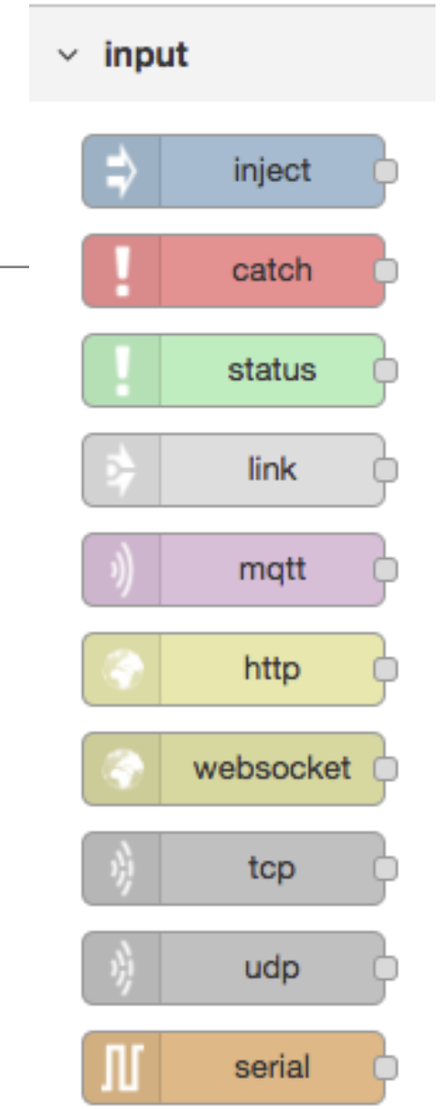
# Output Nodes

- Input nodes allow you to send data outside of a Node-RED flow.
- They have a single input endpoint on their left side.
- Input nodes allow you to send data to other services, for example via Twitter, tcp, serial or email nodes, or to use the debug node to output to the debug pane.



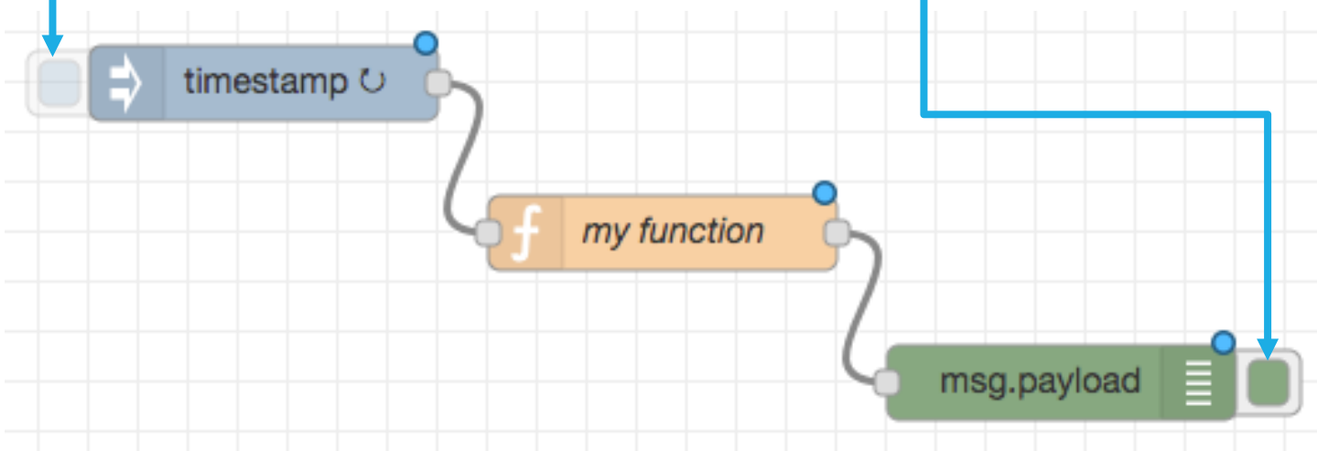
# Processing Nodes

- Input nodes allow you to process data.
- They have an input endpoint and one or more output endpoints.
- They allow you to transform the data type (e.g., json, csv, xml) nodes, use the data to trigger a message (e.g., trigger, delay) nodes and to write custom code that uses the data received (e.g. function node).



# Node with Button

- Some nodes have a button that allows you to … actuate a node or enable/disable a node.

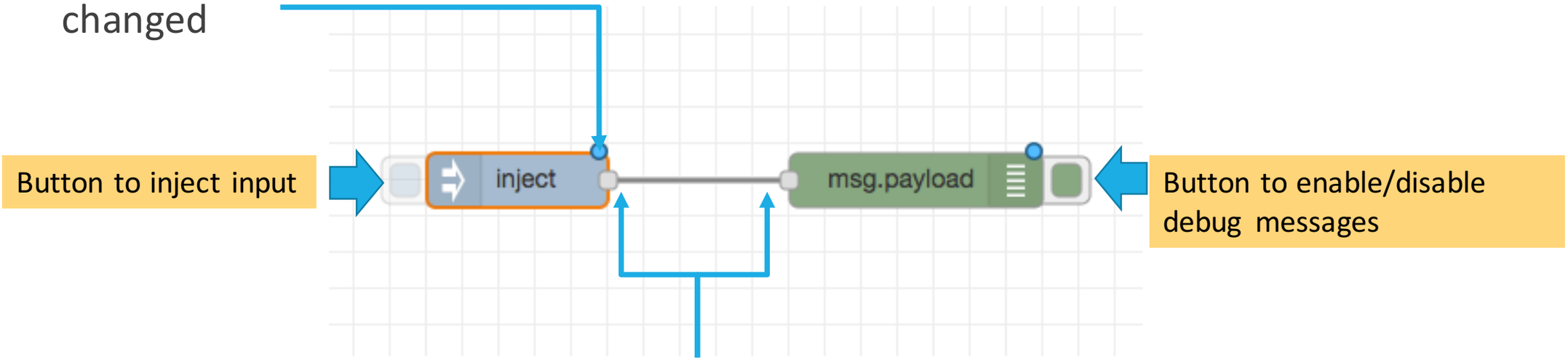



# Flow

- Flows consist of multiple nodes wired together
- Messages flow along the nodes carrying data from node to node.

# Symbols

- The blue dot indicates that the node hasn't been deployed since it was last changed




- Gray dots indicate output/input for wiring

# NodeRED message

- Node-RED nodes consume input messages and produce output messages.
- Messages are JavaScript objects that contain at least a “payload” parameter, like this:

```
msg = {
  payload:"message payload"
};
```

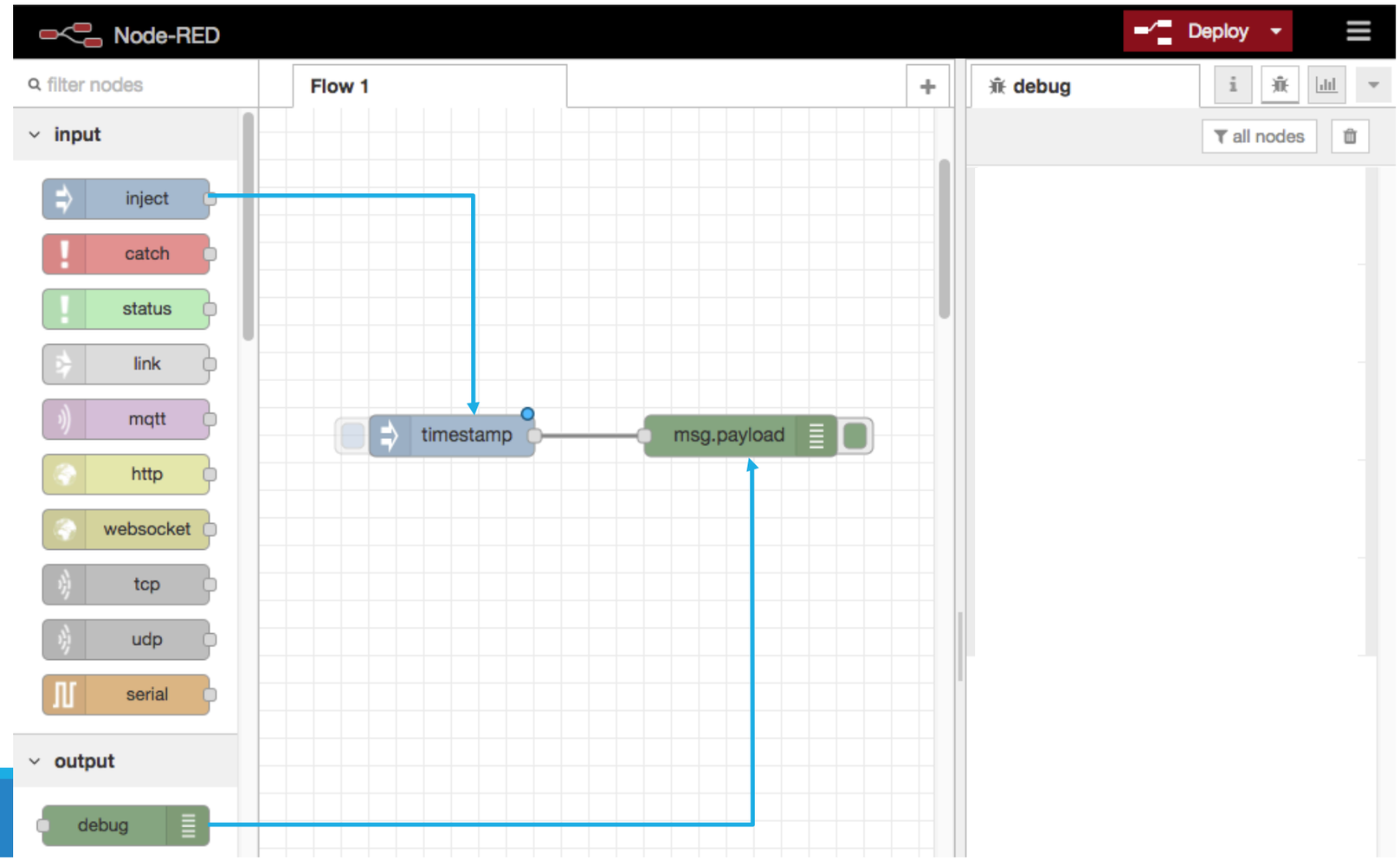
```
msg = {
  payload:"message payload",
  topic:"error",
  location:"somewhere in space and time"
};
```

# **Workshop 2.** Building your First Flow “Hello, Beautiful!”

# Let’s Get Started

Let’s start with the easiest flow possible:

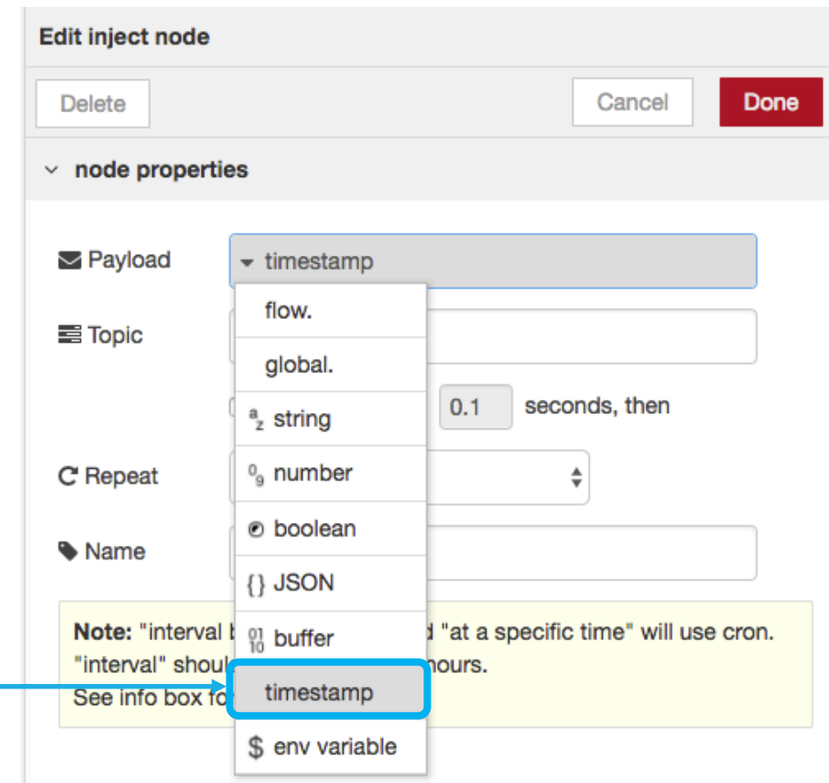
- Drag and drop the ’inject’ and ‘debug’ nodes
- Wire them together



# Explore Options of the Inject Node

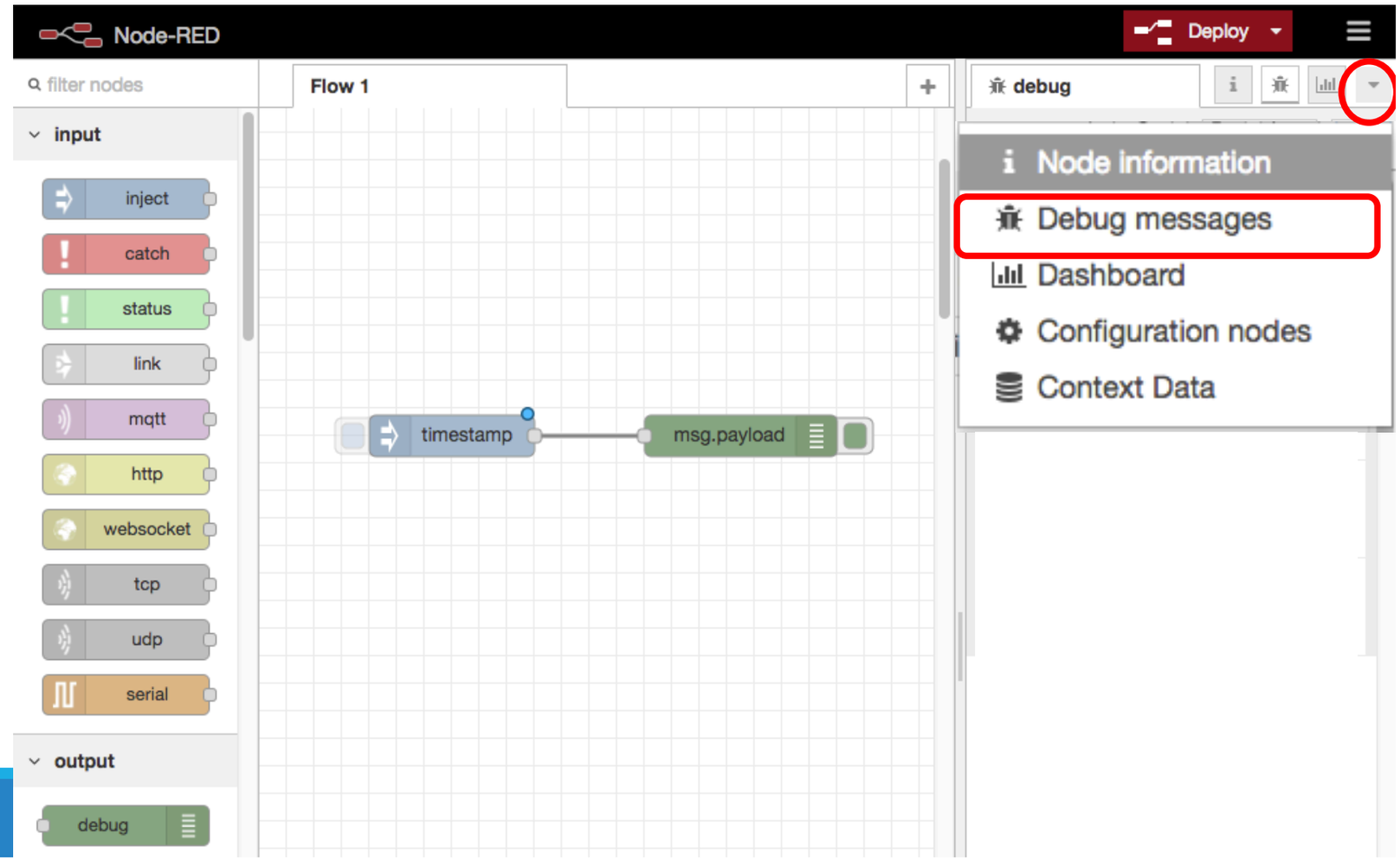
Let us try:

- timestamp (default)
- Hit ‘Done’



# Open the Debug Window

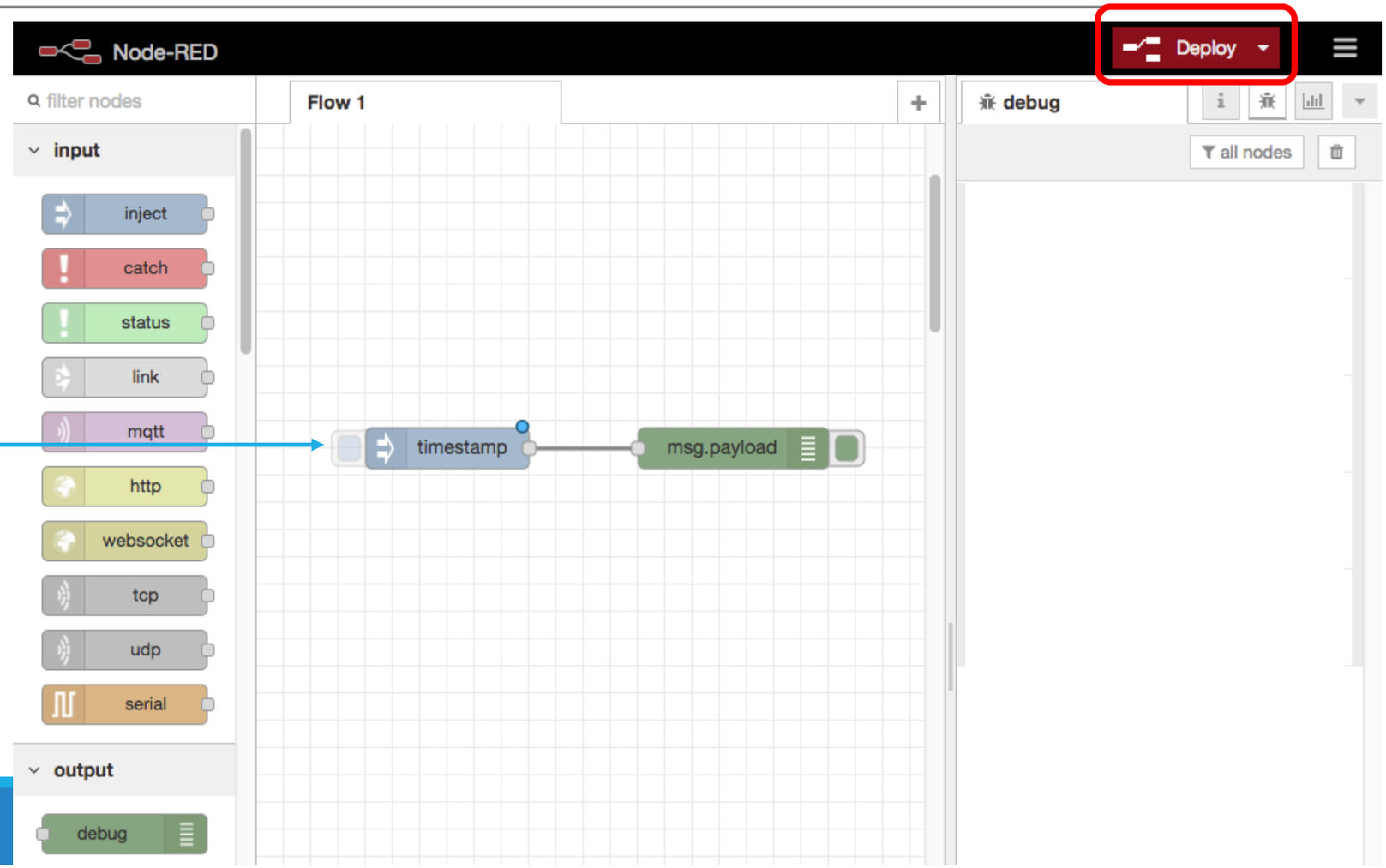
- Select the drop-down arrow on the top right of the screen
- Choose ‘Debug messages’



# Deploy

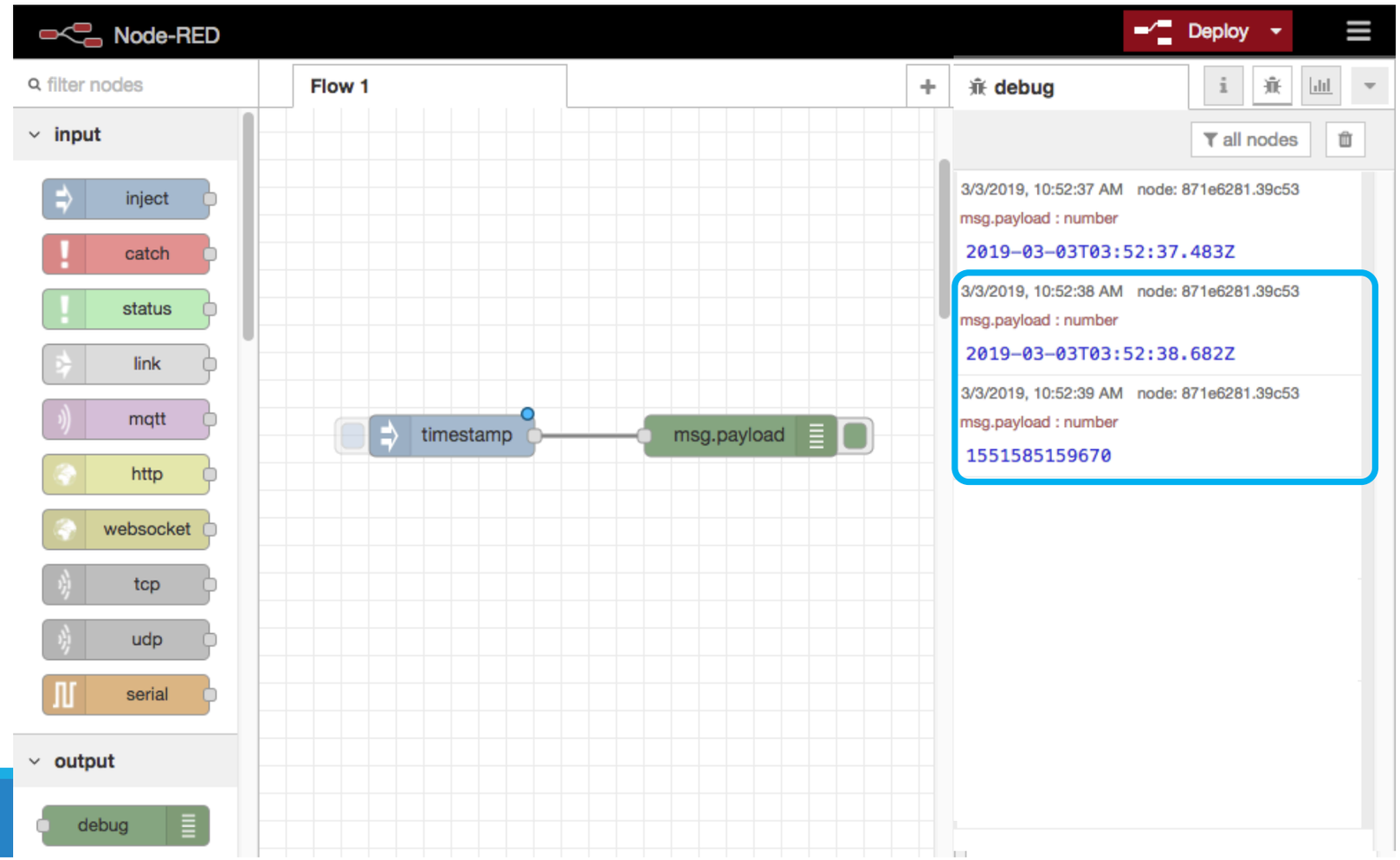
Red = have not been deployed
Gray = already deployed

- Hit the 'Deploy' button on the top right
- Hit the 'button' of the inject node, examine the output debug messages!



# Timestamp from the Inject Node

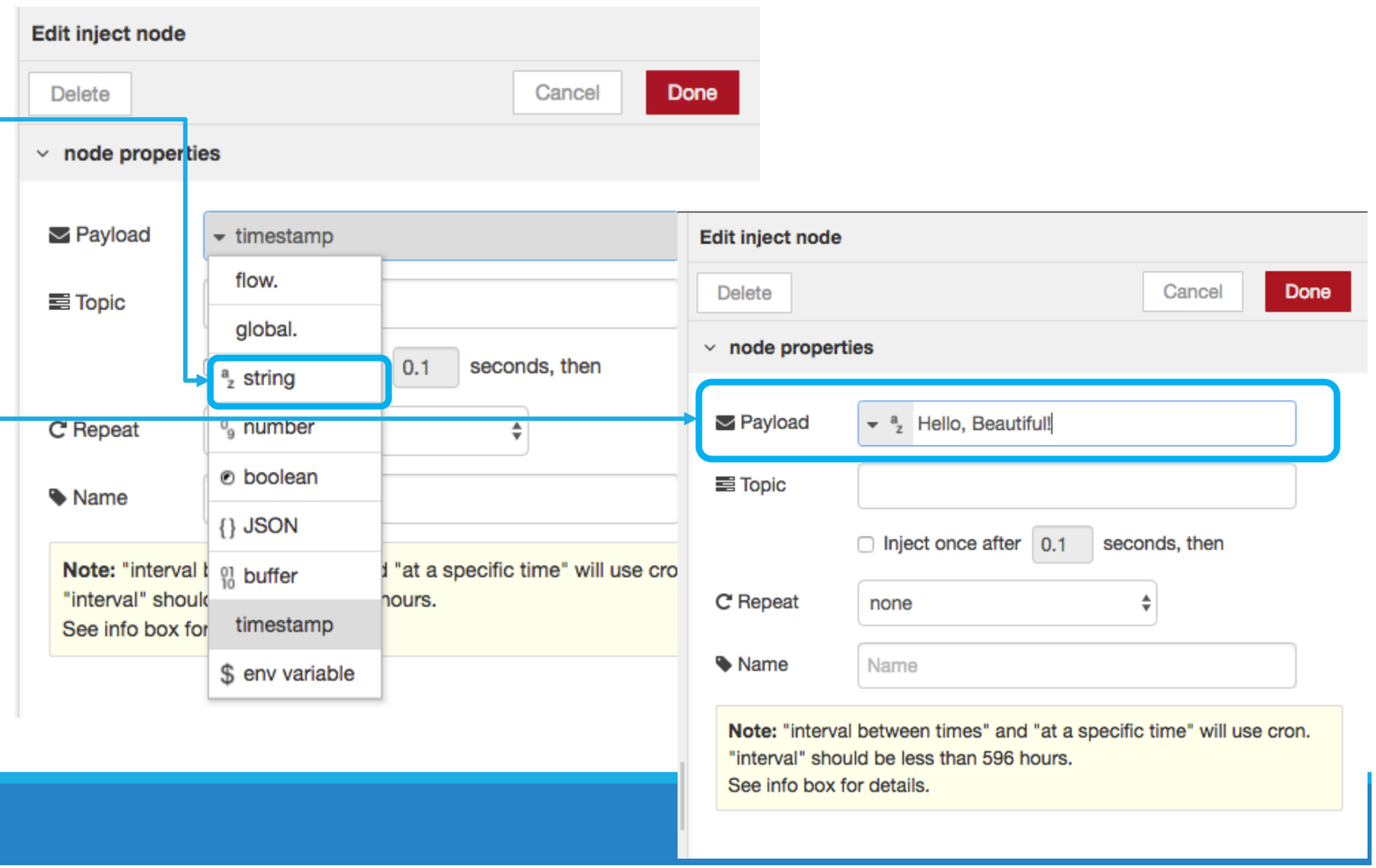
- Timestamp = the current time in milliseconds since January 1, 1970.
- Click at the message to see different time formats.



# Explore Options of the Inject Node

Let us try:

- String
- In the Payload field, type: “Hello, Beautiful”
- Hit ‘Done’



# Deploy

- Hit the 'Deploy' button on the top right
- Hit the 'button' of the inject node, examine the output debug messages!

# Debug Node Options

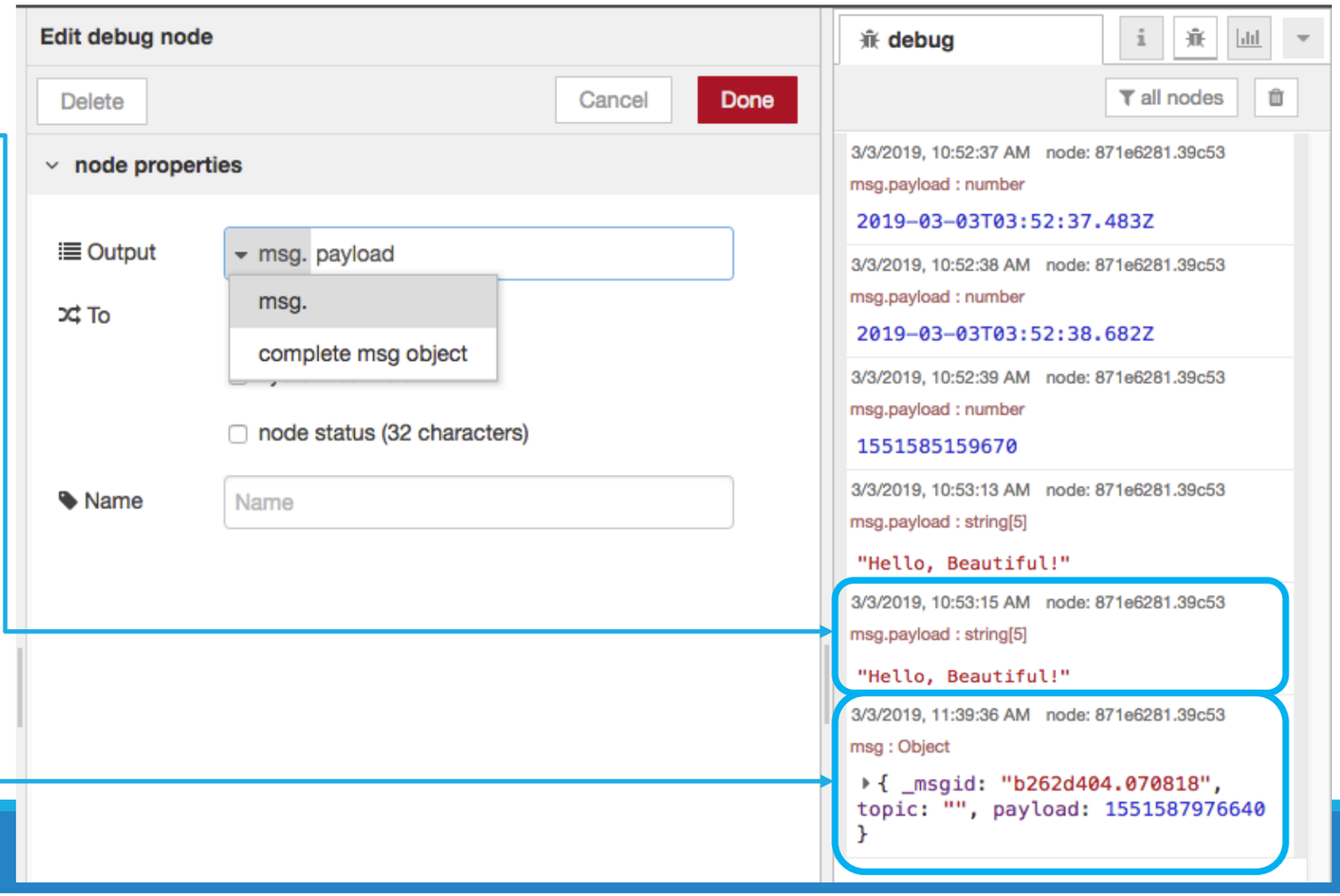
Two options:

- ‘msg.’
- ‘complete msg object’

All messages in Node-RED have three default properties:

1) the payload, which we are seeing above;
2) a message topic, which is a user-defined string describing what the message is about; and
3) an internal identifier.

These fields are extensible!

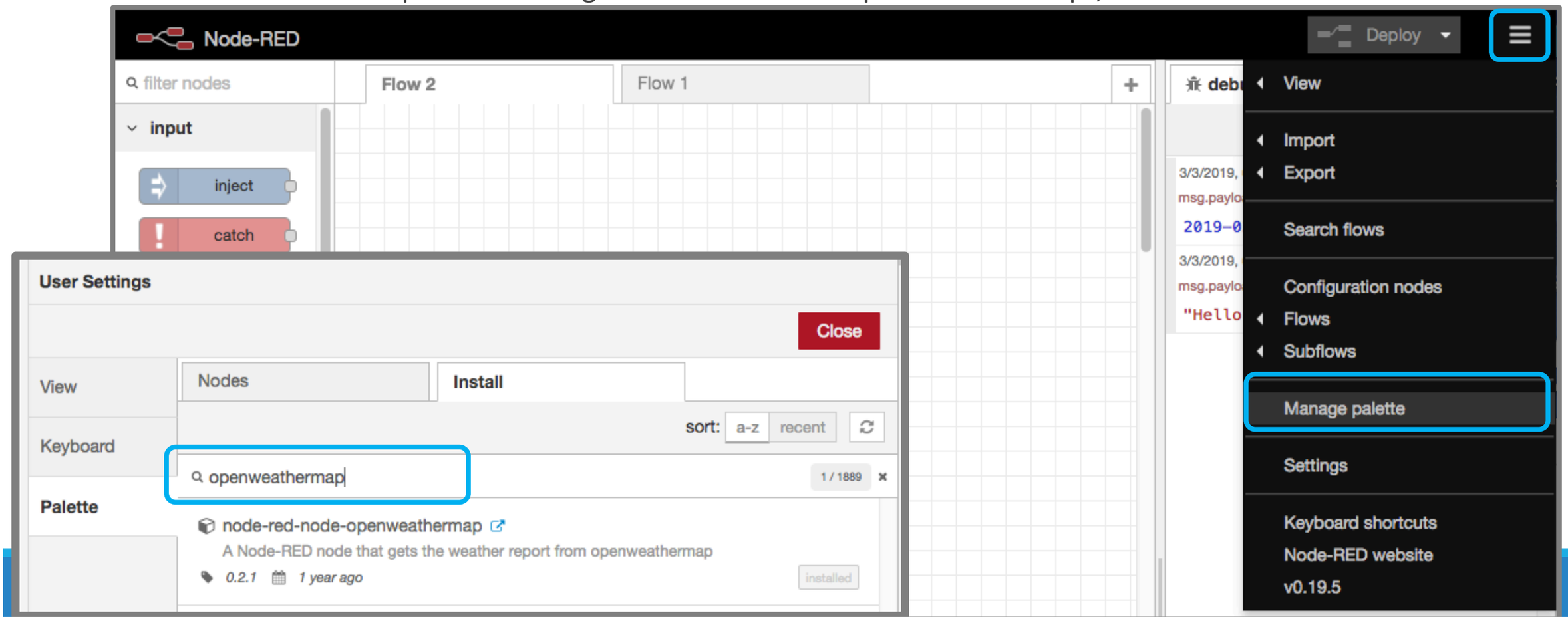


# **Workshop 3.** Get weather data and send alert

We will get weather data and send an email alert.

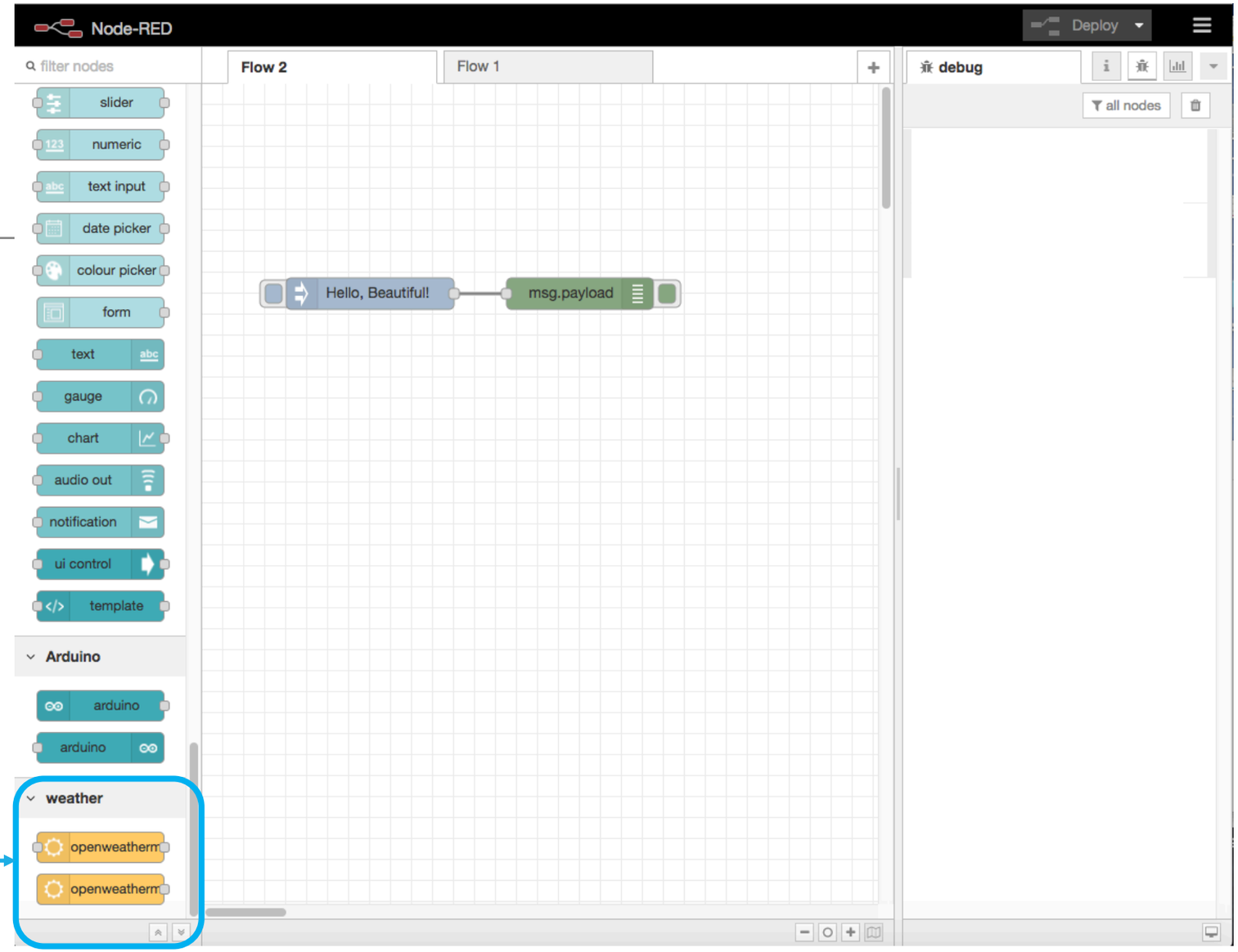
# Download “openweathermap” Node

- Go to Node-RED palette manager and search for “openweathermap”, and install it.



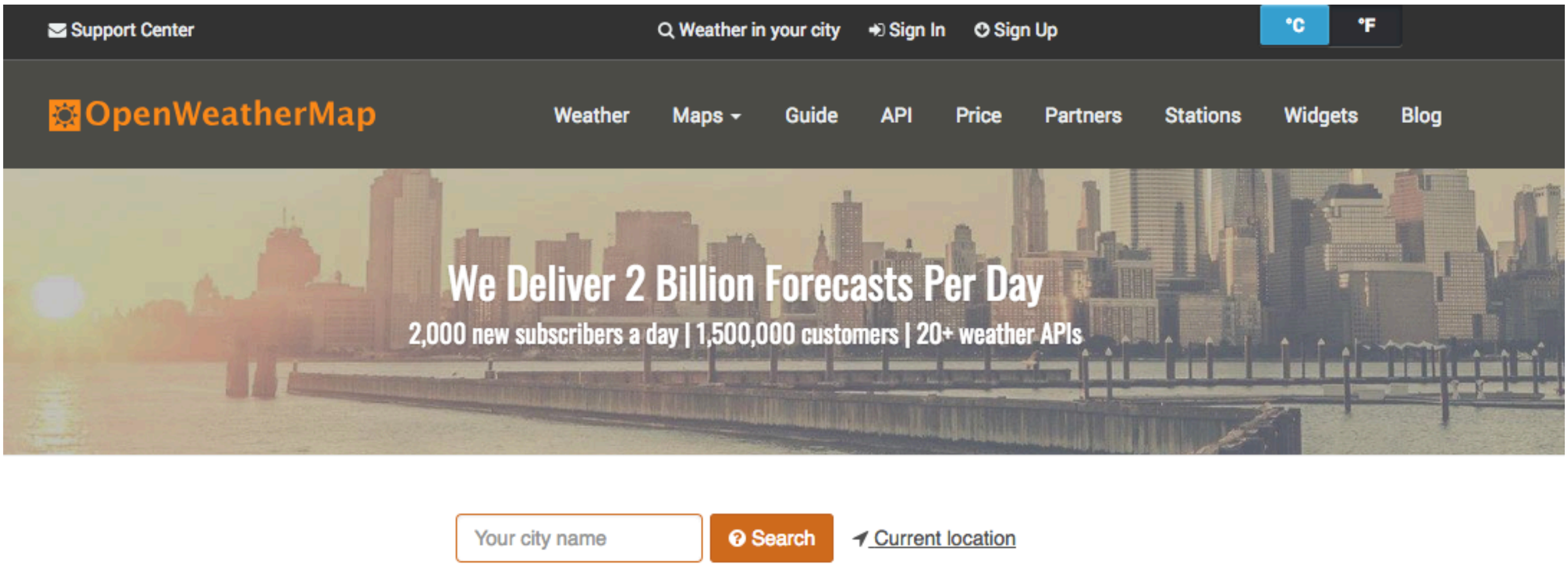
# openweathermap node

You will see openweathermap node on your left navigation panel.

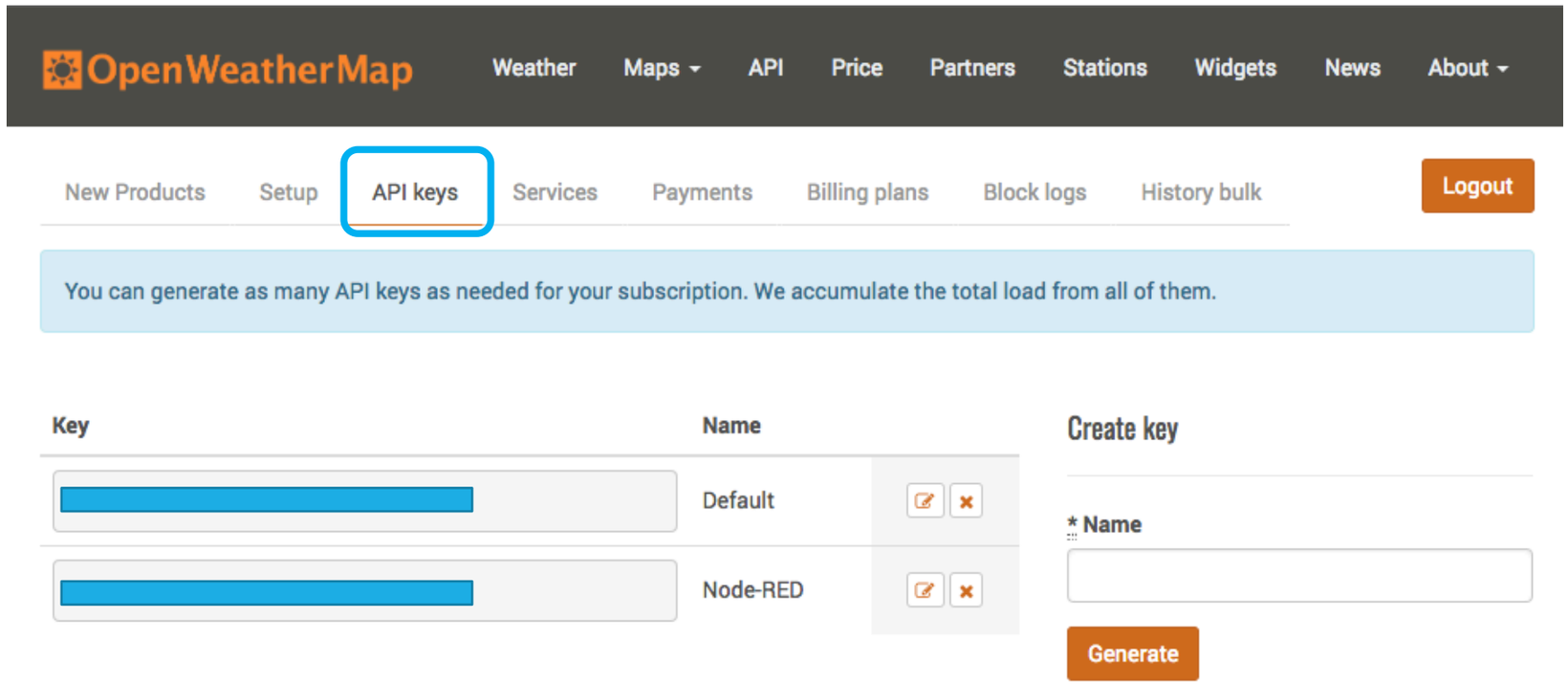


# Sign up for openweathermap

Sign up for openweathermap at https://openweathermap.org/

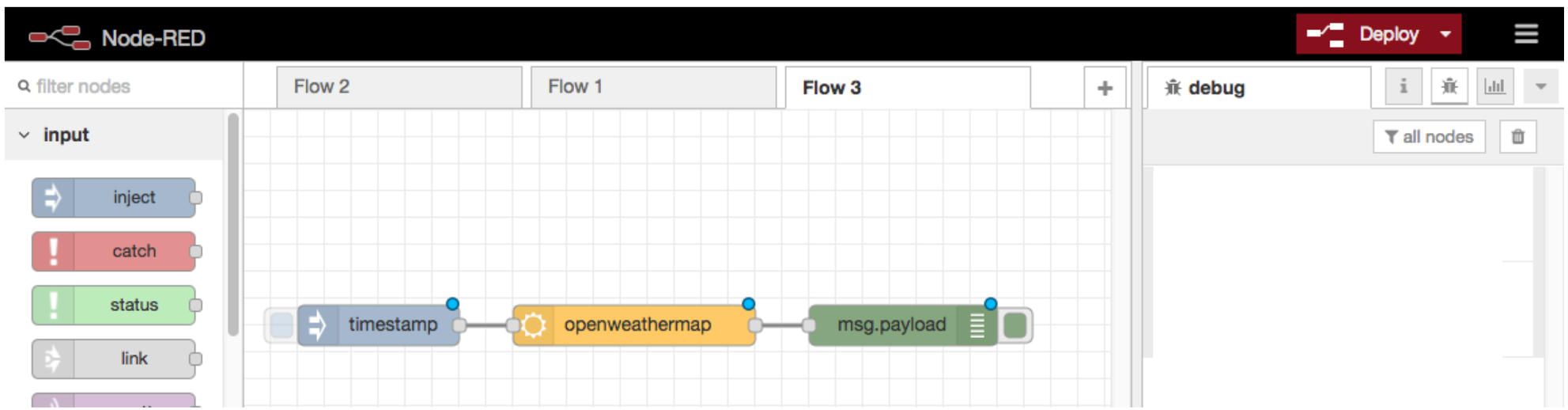


# Go to “API keys” Tab, and Get/Generate Your Key



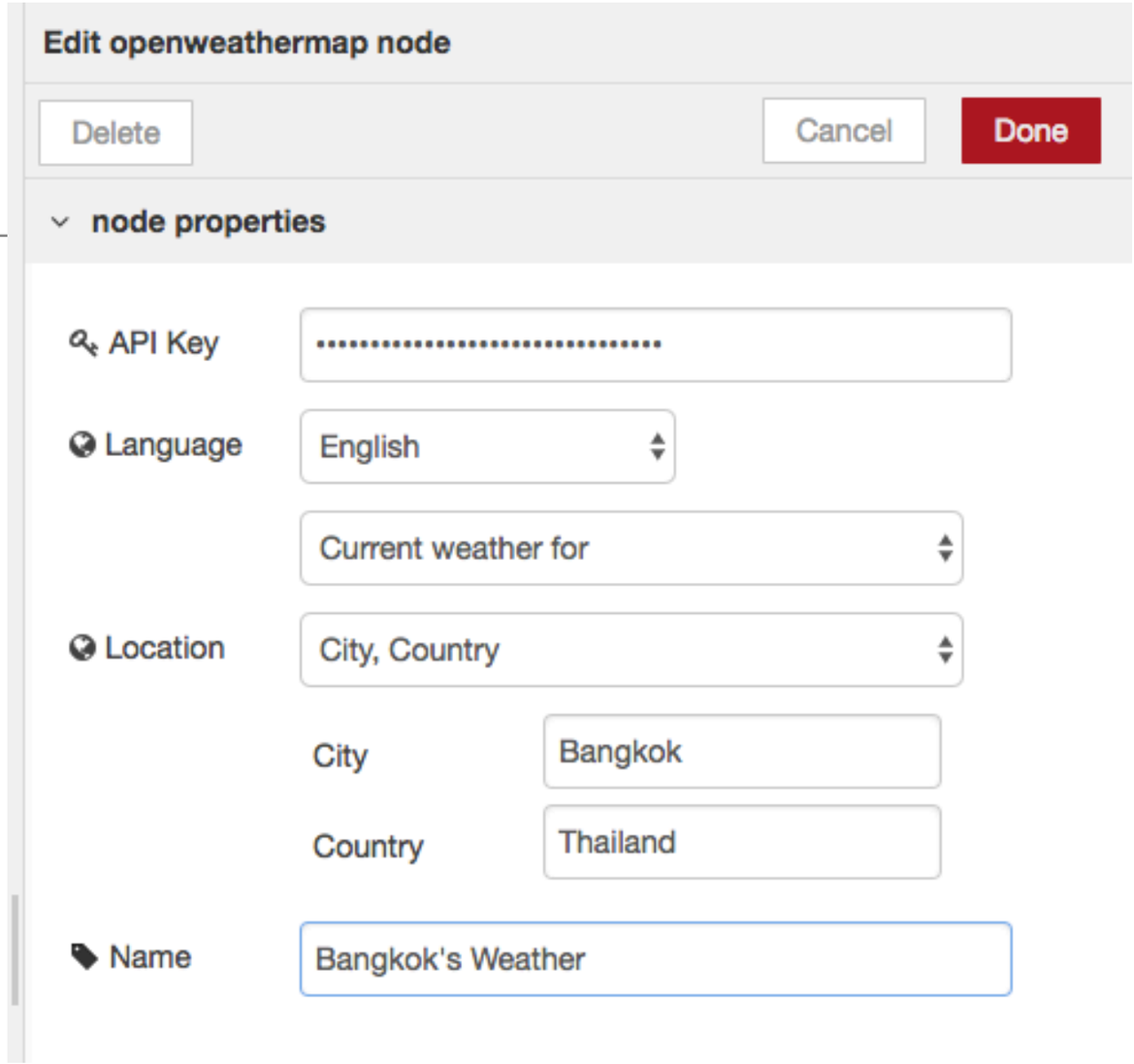
# Create the Flow

- Drag & drop: the inject node, the openweathermap  node, and the debug node.
- Wire them together



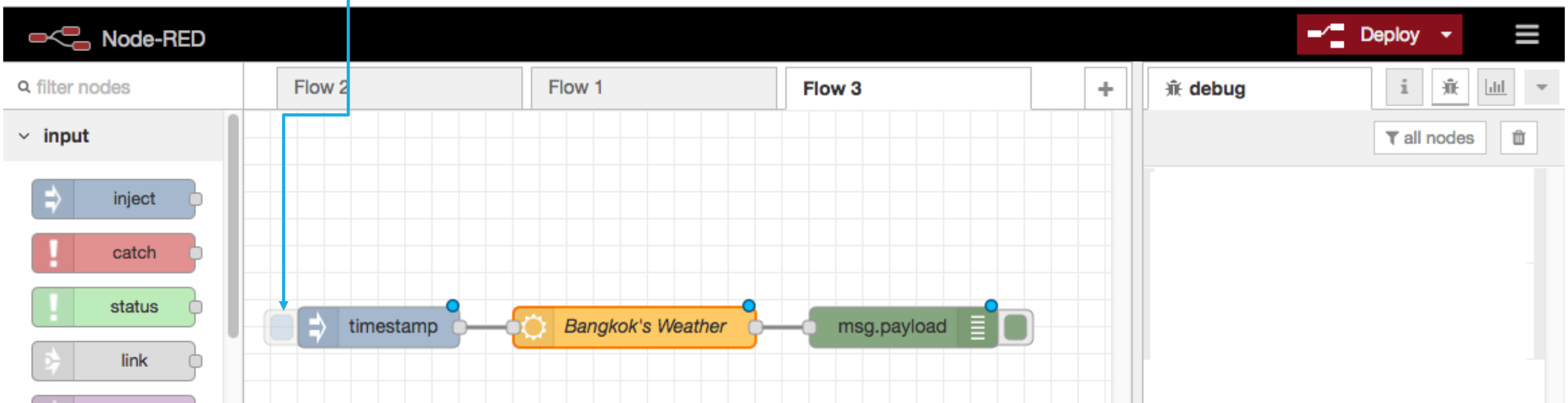
# Configure the openweathermap node

- Enter the API Key (from openweathermap)
- Enter the “City” and ”Country”
- Enter the node name (optional)
- Hit ”Done”

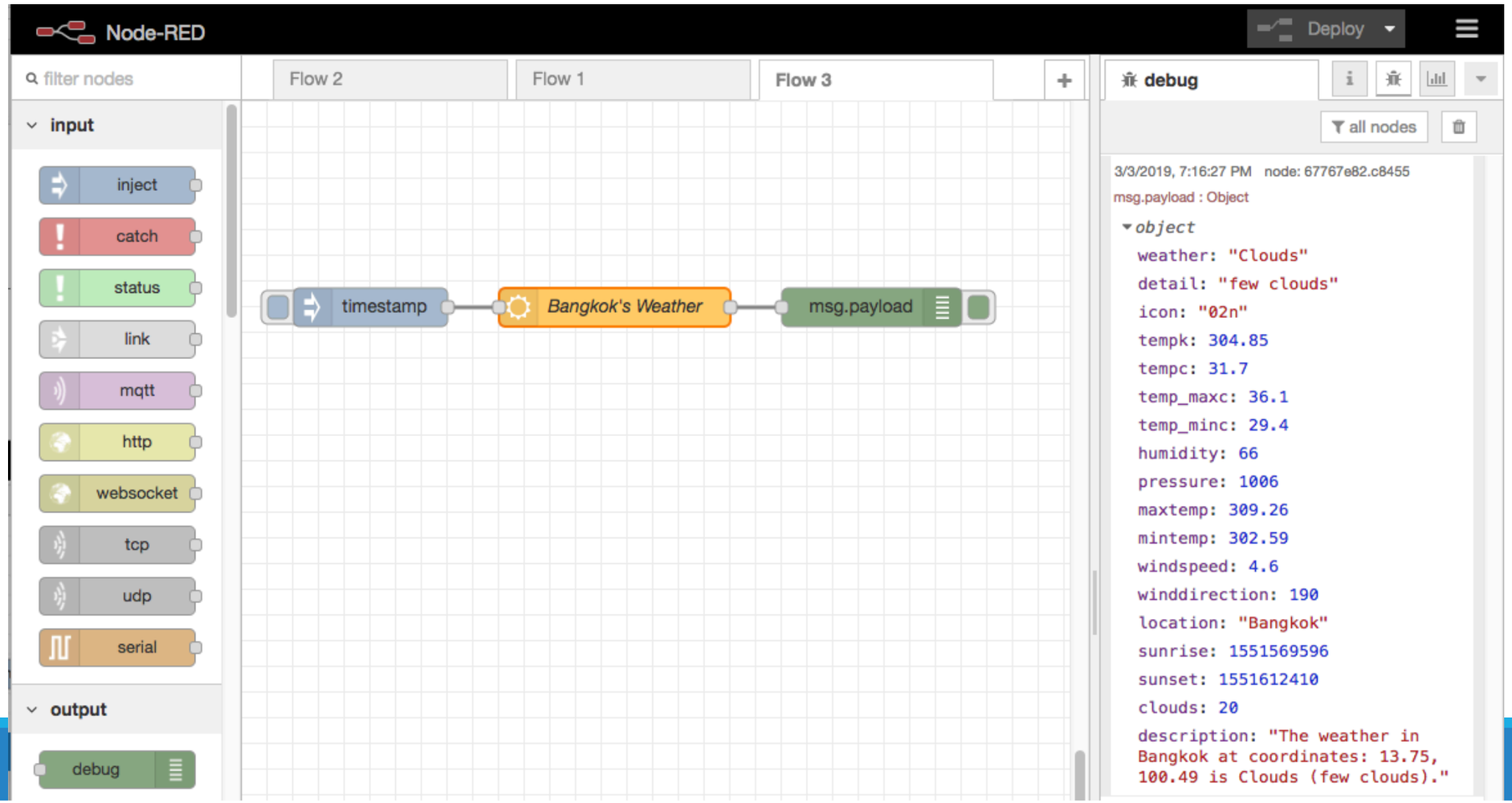


# Deploy

- Hit the ‘Deploy’ button on the top right
- Hit the ‘button’ of the inject node & examine the output debug messages!

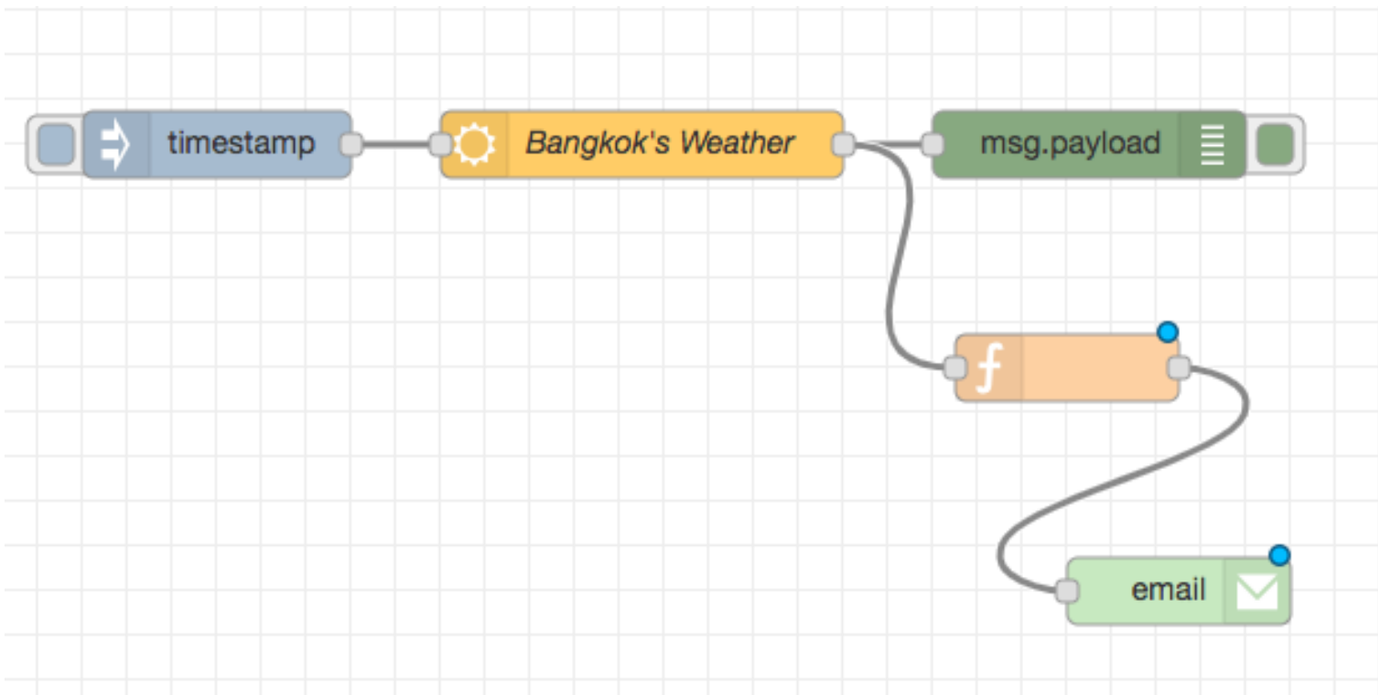


# Examine the Results



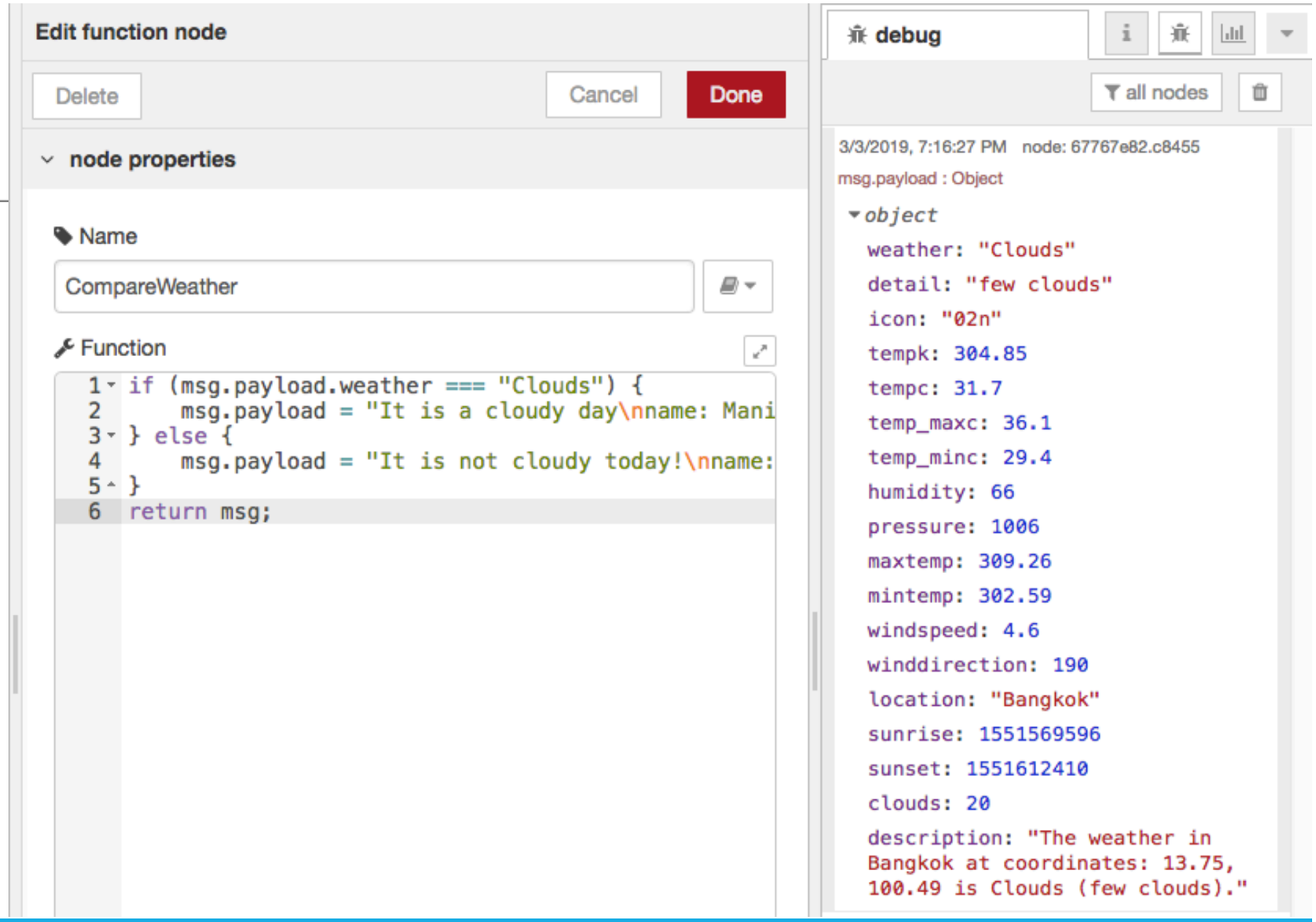
# Let us React on a Weather Condition!

- Drag and drop the ”function” node and the “email” node



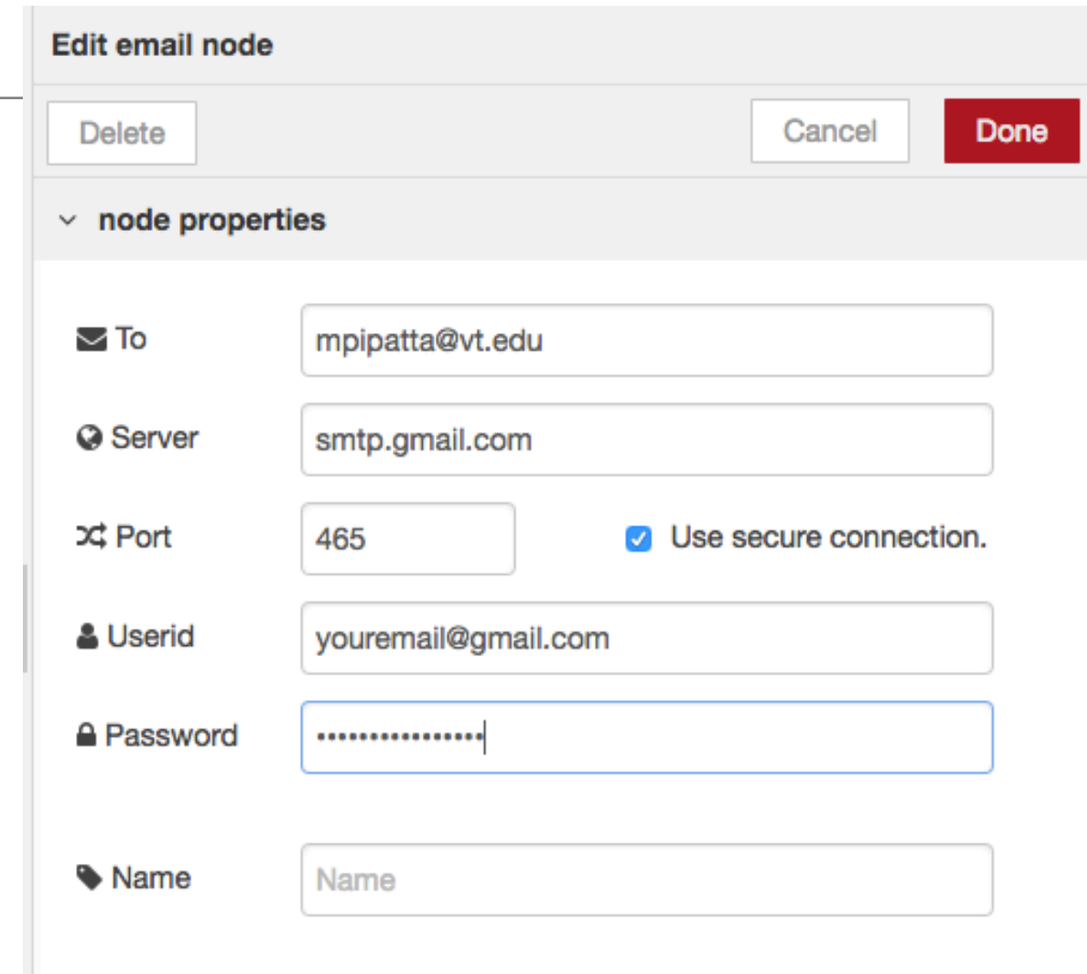
# Function Node

- Add a function to compare weather data, and set **msg.payload**
- In the **msg.payload**, please add your name and ID number.
- Use **\n** to start new line of the message.



# Configure Email Node

- Send email to **your friend's email address**
- Enter your **userID** (gmail email address)
- Enter your **App password***



# Set up your Google App Password

Please follow the instruction available here: [https://support.google.com/accounts/answer/185833?hl=en](https://support.google.com/accounts/answer/185833?hl=en)

## How to generate an App password

1. Go to your Google Account .
2. On the left navigation panel, click **Security**.
3. On the *Signing in to Google* panel, click **App passwords**.
   **Note:** If you can't get to the page, 2-Step Verification is:
   - Not set up for your account
   - Set up for security keys only
4. At the bottom, click **Select app** and choose the app you're using.
5. Click **Select device** and choose the device you're using.
6. Click **Generate**.
7. Follow the instructions to enter the App password (the 16 character code in the yellow bar) on your device.
8. Click **Done**.

Once you are finished, you won't see that App password code again. However, you will see a list of apps and devices you've created App passwords for.

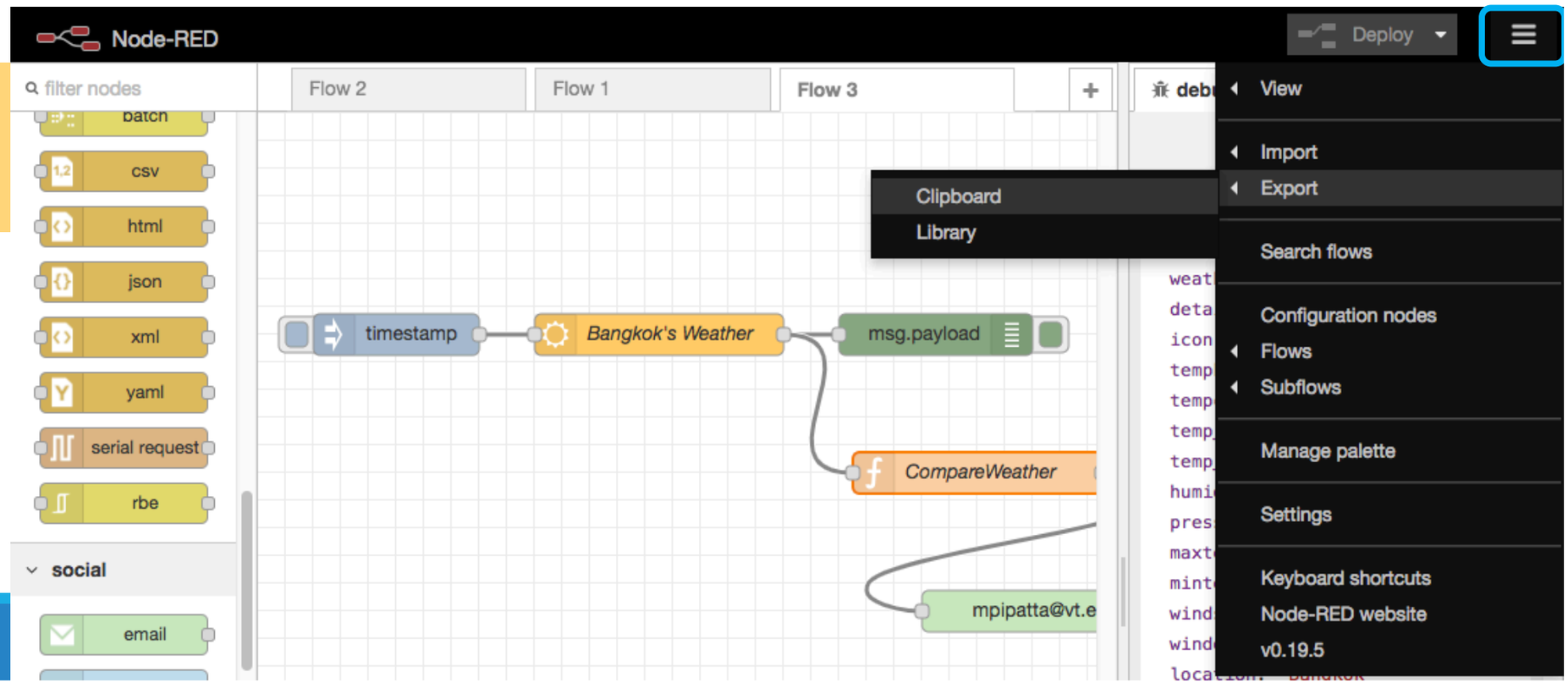
**Note:** You may not be able to create an App password for less secure apps. Learn more about allowing less secure apps.

# Saving Your Flow: Import/Export NodeRED Workspace

- NodeRED flows are coded as JSON strings
- You can copy and paste nodes and flows using the standard cut&paste clipboard.

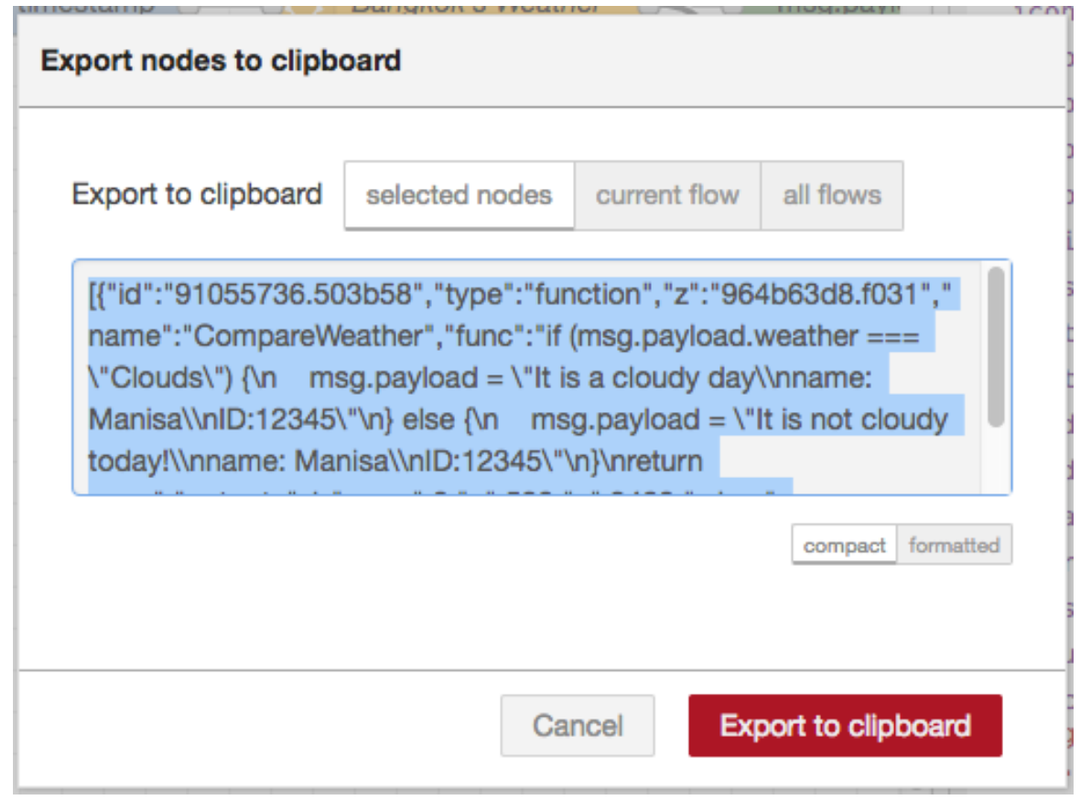
To export your flow:

- Click the hamburger > Export > Clipboard



# Export the Flow

- Copy your flow to clipboard.
- You can keep the code for later use by importing it to your NodeRED application.

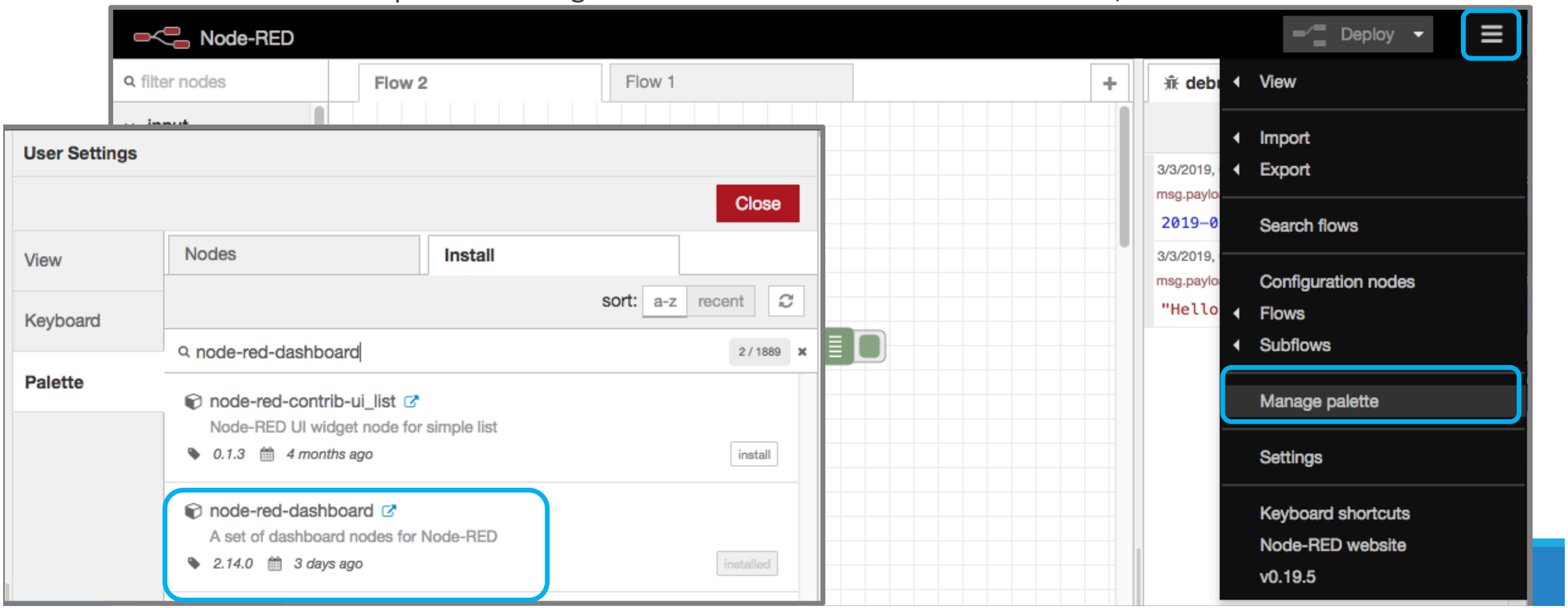


# **Workshop 4.** UI nodes from node-red-dashboard

We will learn how to create dashboard using NodeRED

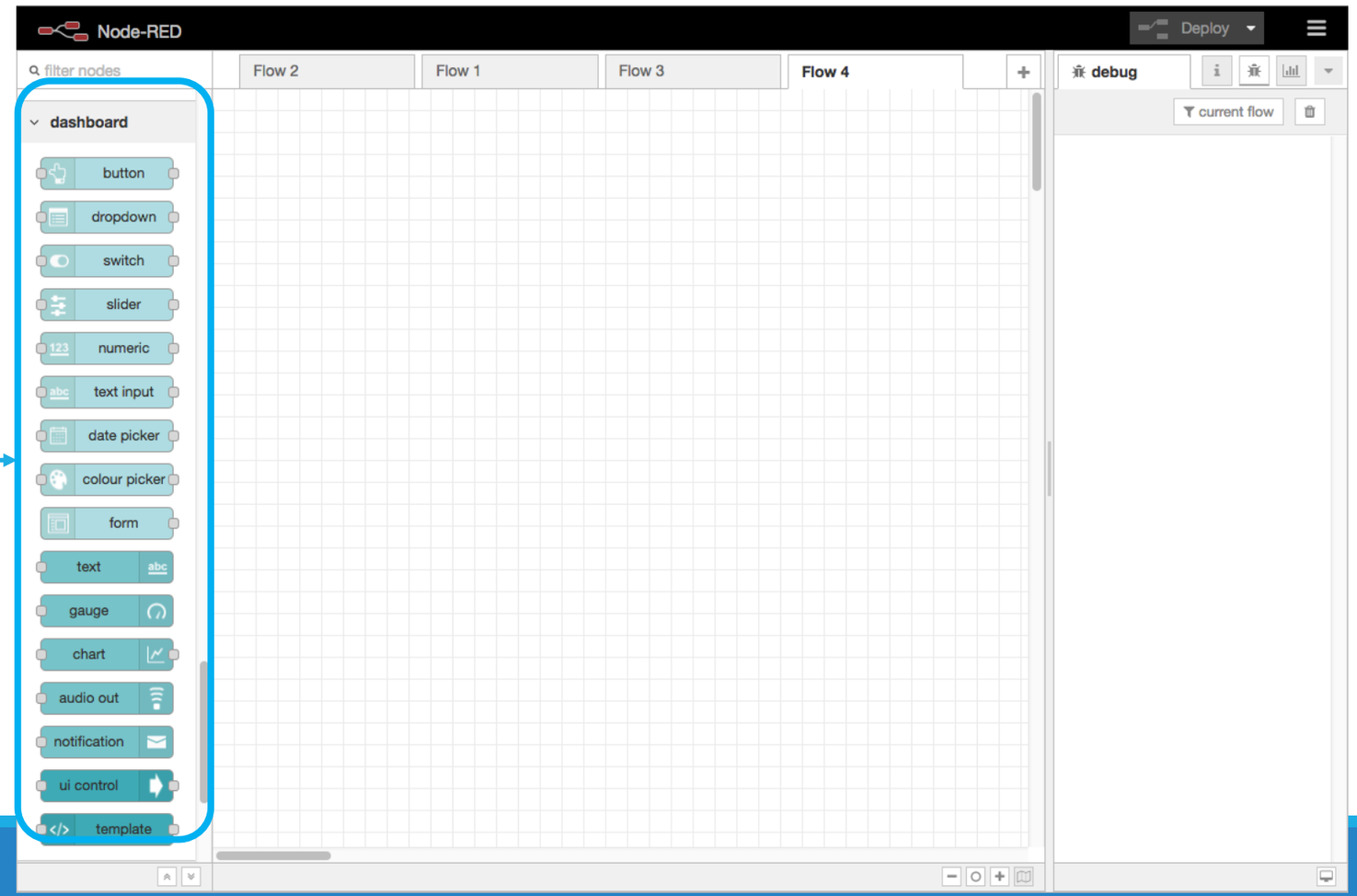
# Download “node-red-dashboard” Node

- Go to Node-RED palette manager and search for “node-red-dashboard”, and install it.



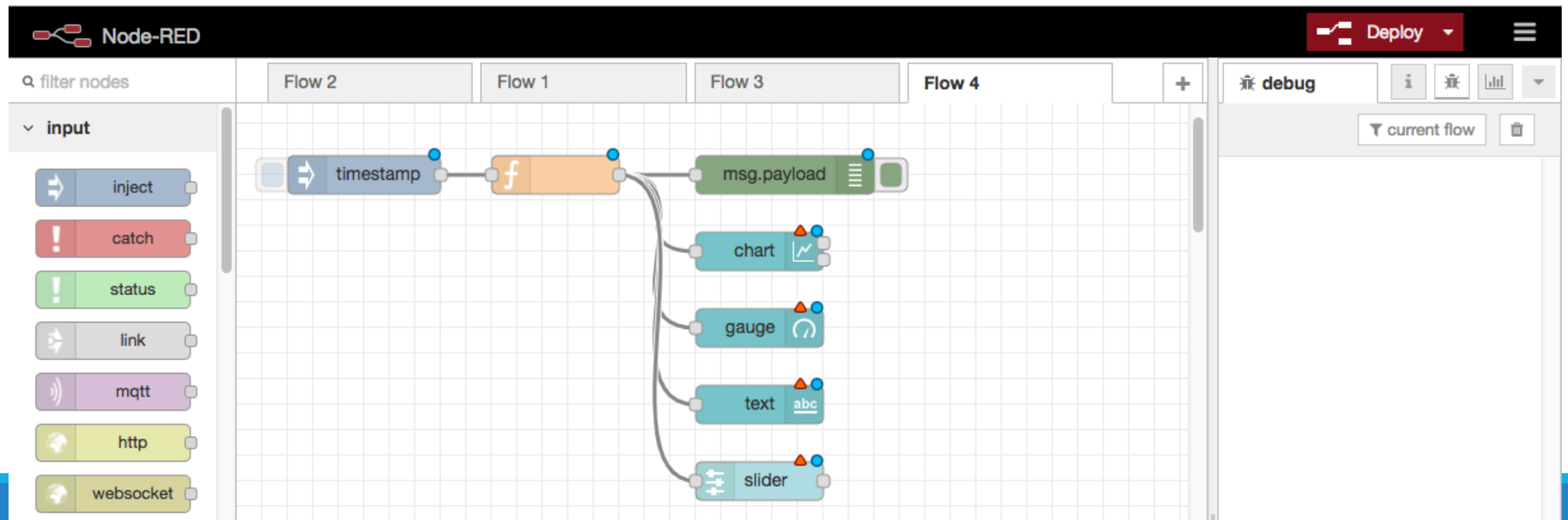
# ‘node-red-dashboard’ node

You will see the dashboard nodes on your left navigation panel.



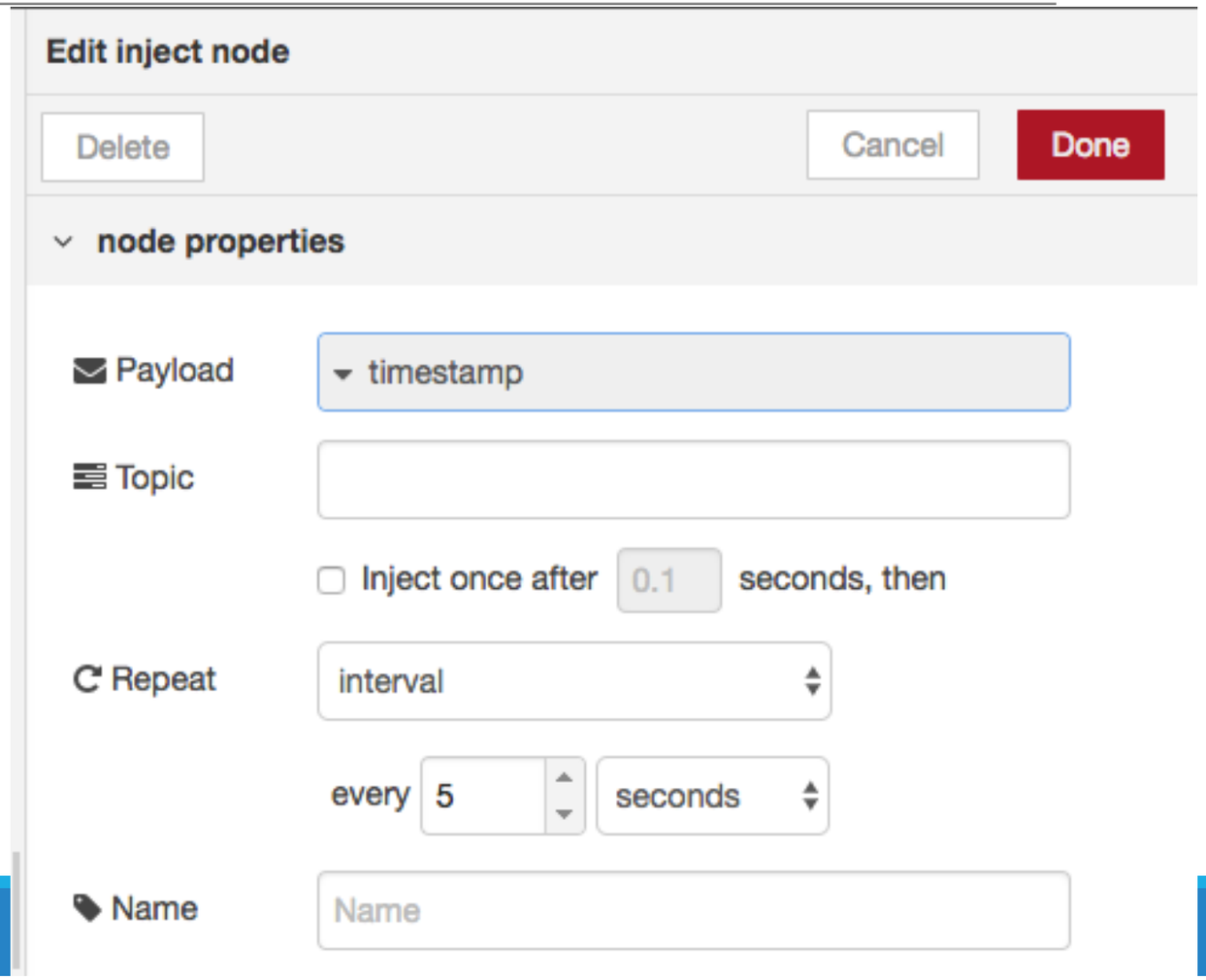
# Create the Flow

- Drag & drop: the **inject** node, the **function** node, the **debug** node, and the **chart, gauge, text and slider** nodes.
- Wire them together as shown.



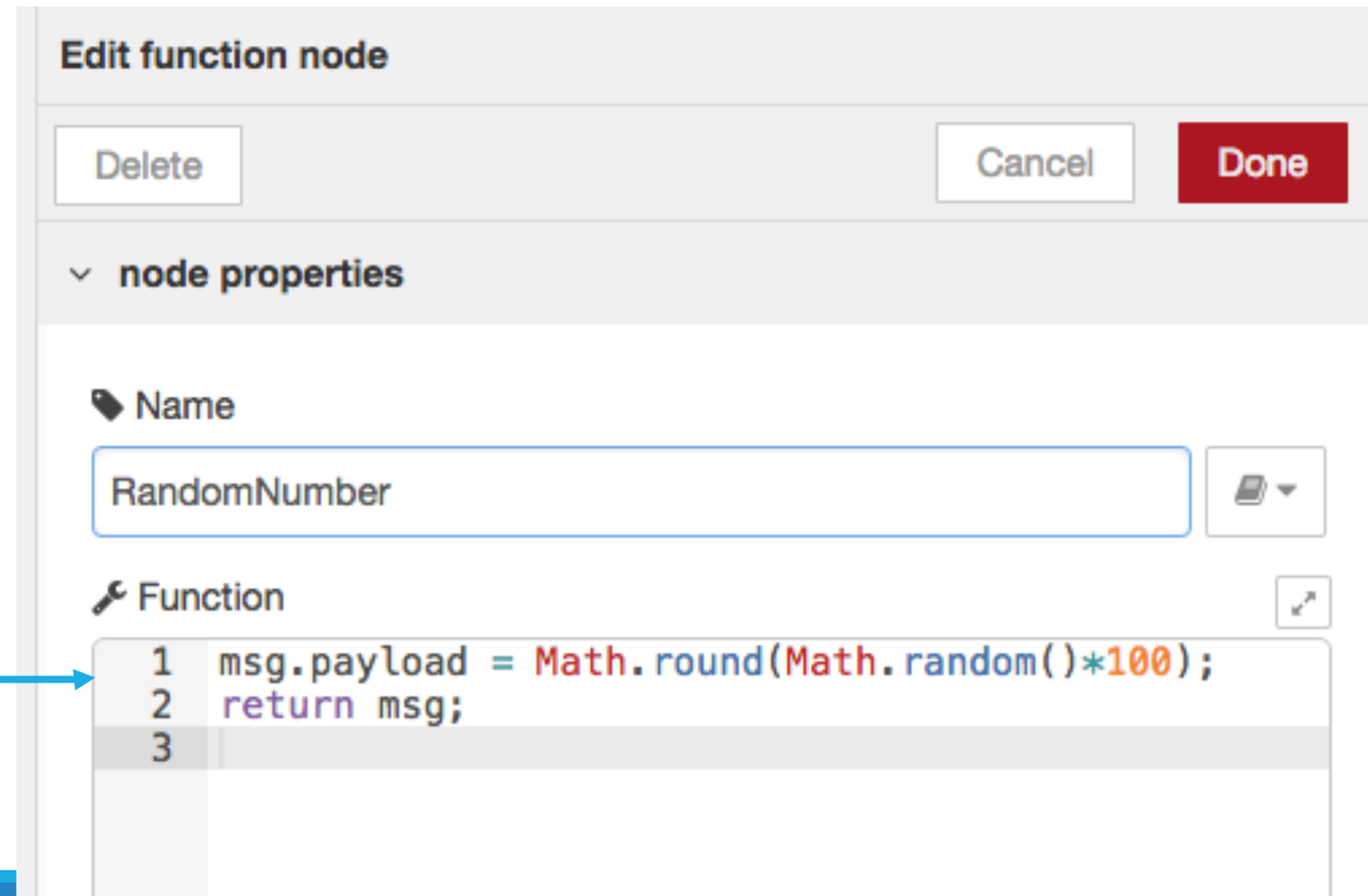
# Edit the Inject Node

- Payload: timestamp
- Repeat: interval, every 5 seconds
- Hit “Done”



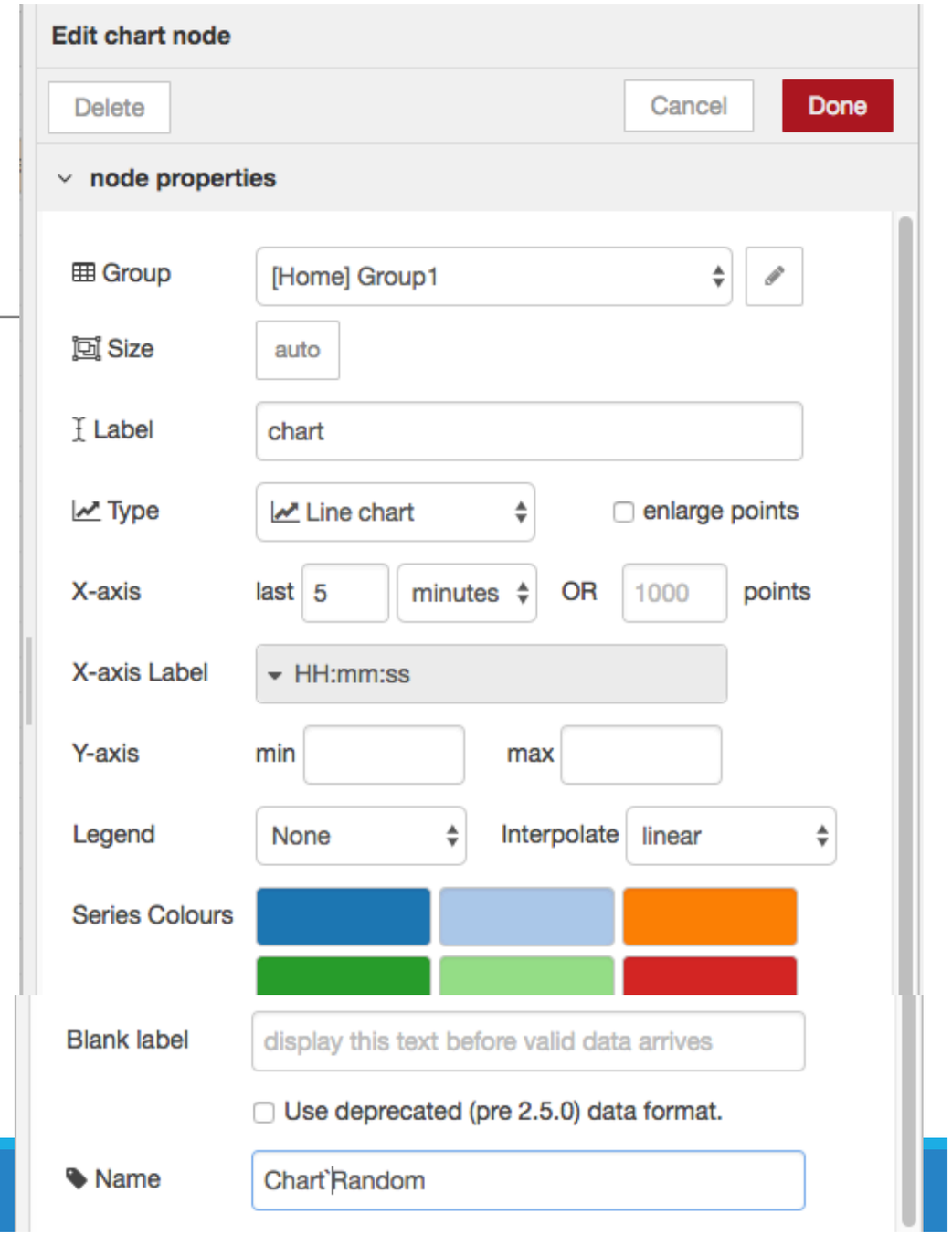
# Edit the Function Node

- Name: RandomNumber
- Function: this function will generate a random number between 0-99.
- Hit “Done”



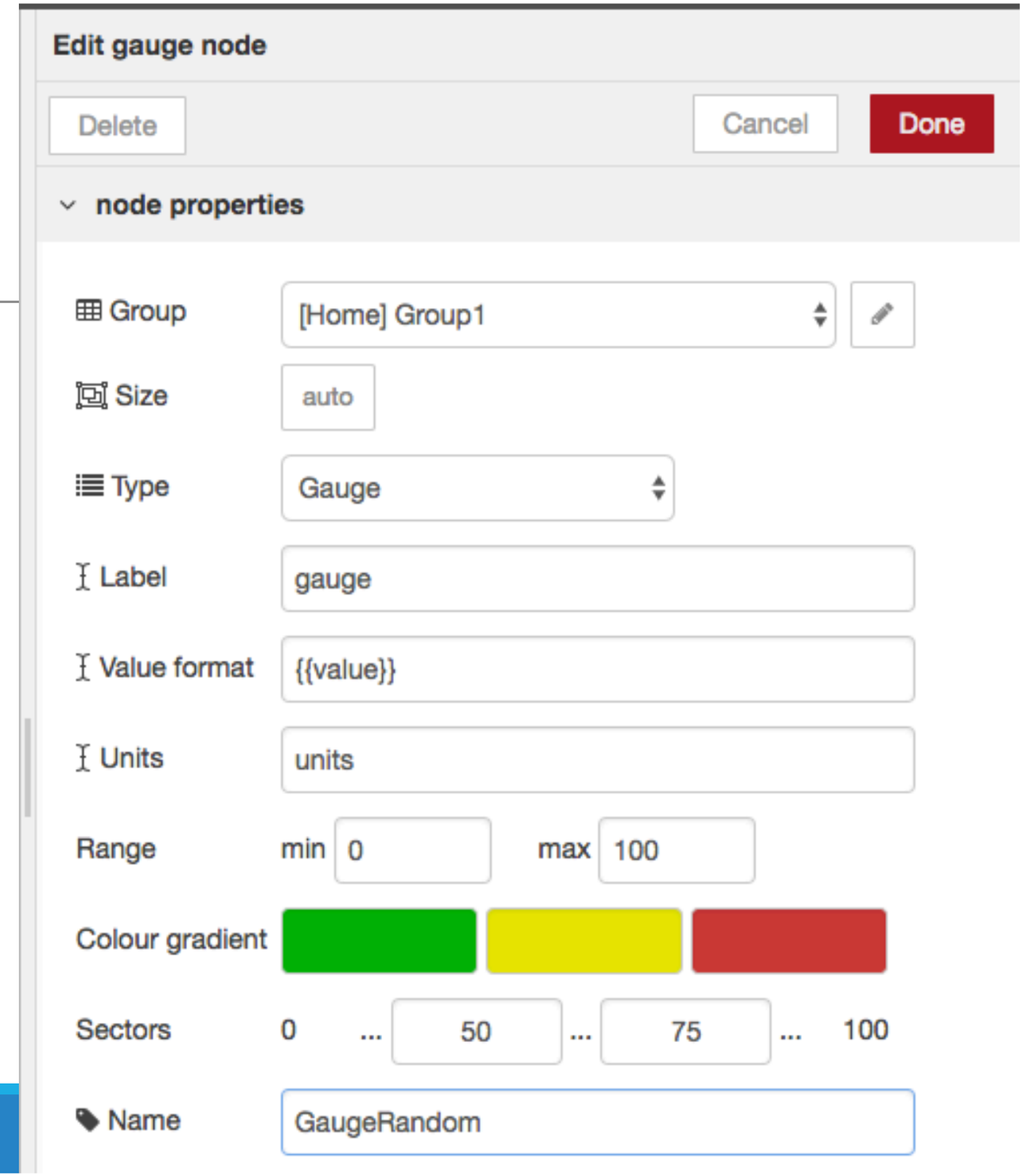
# Edit the Chart Node

- Group: select “Add new ui_group”, click the pen icon to edit, enter…
  - Name: Group 1
  - Tab: Home (need to edit)
  - Click “Add” button
- X-axis: last 5 minutes
- Name: ChartRandom
- Hit “Done”



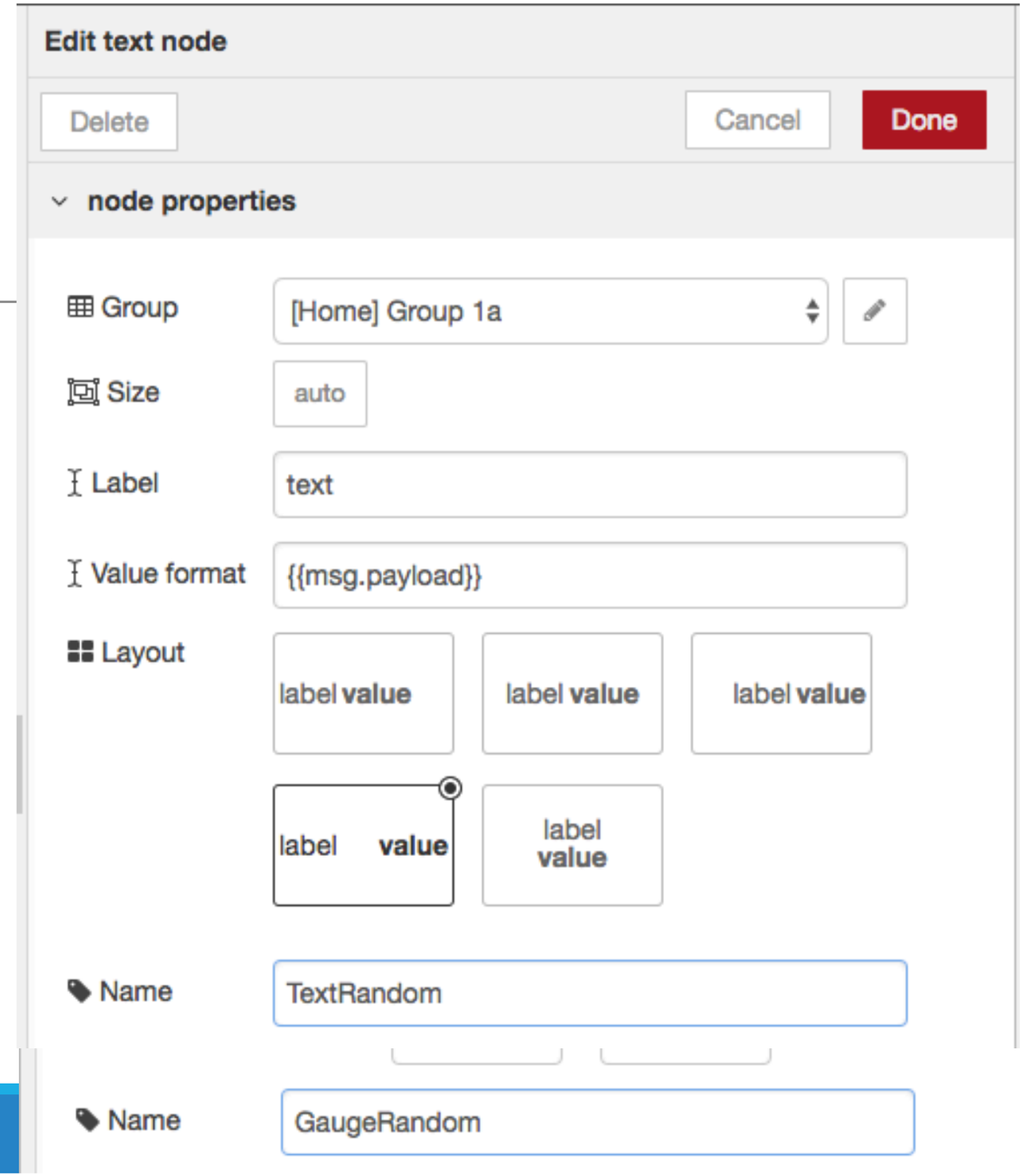
# Edit the Gauge Node

- Group: select [Home] Group1
- Range: 0 - 100
- Name: GaugeRandom
- Sectors: 0…50…75…100
- Hit “Done”



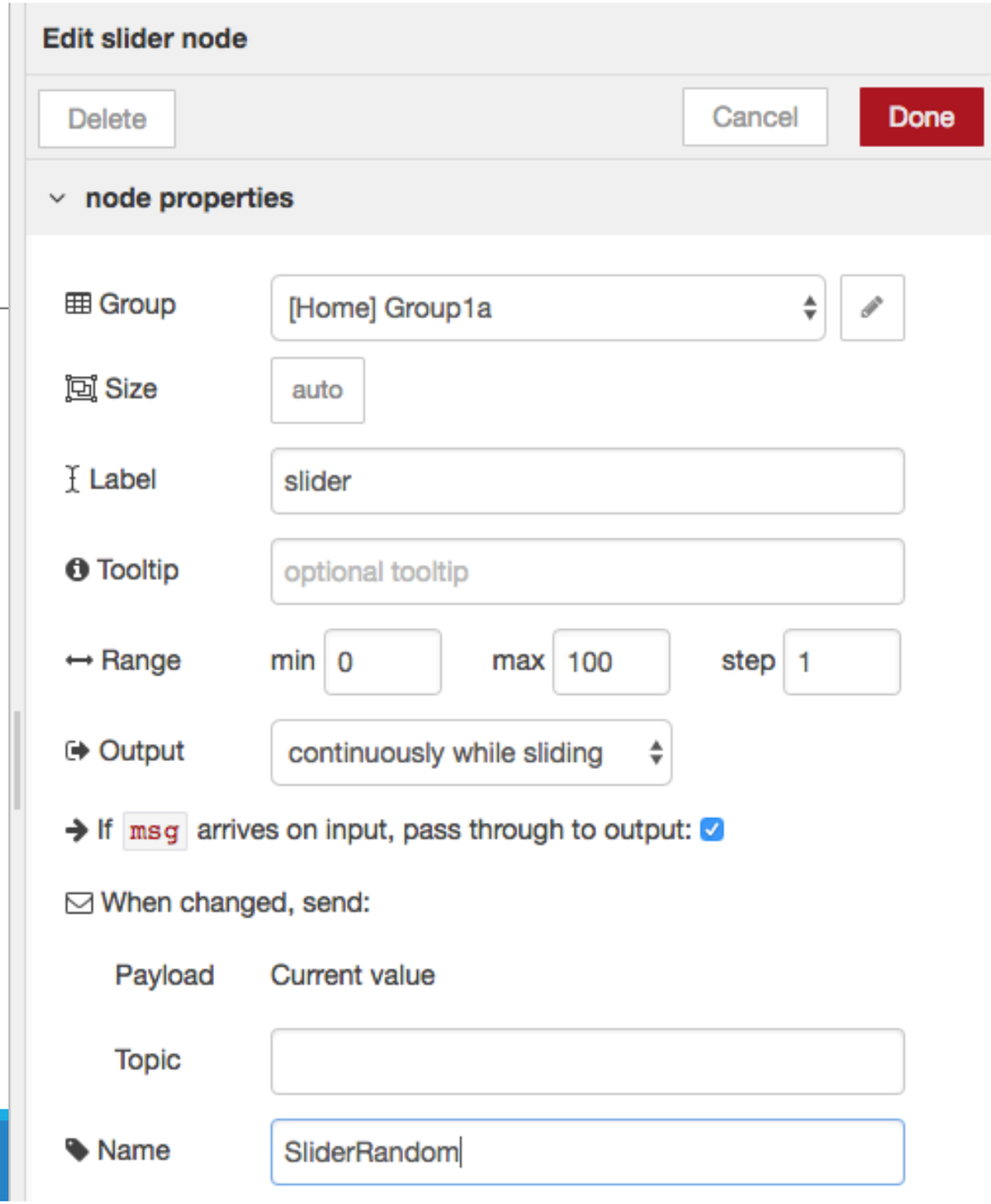
# Edit the Text Node

- Group: select “Add new ui_group”, click the pen icon to edit, enter…
  - Name: Group 1a
  - Tab: Home
  - Click “Add” button
- Name: TextRandom
- Hit “Done”



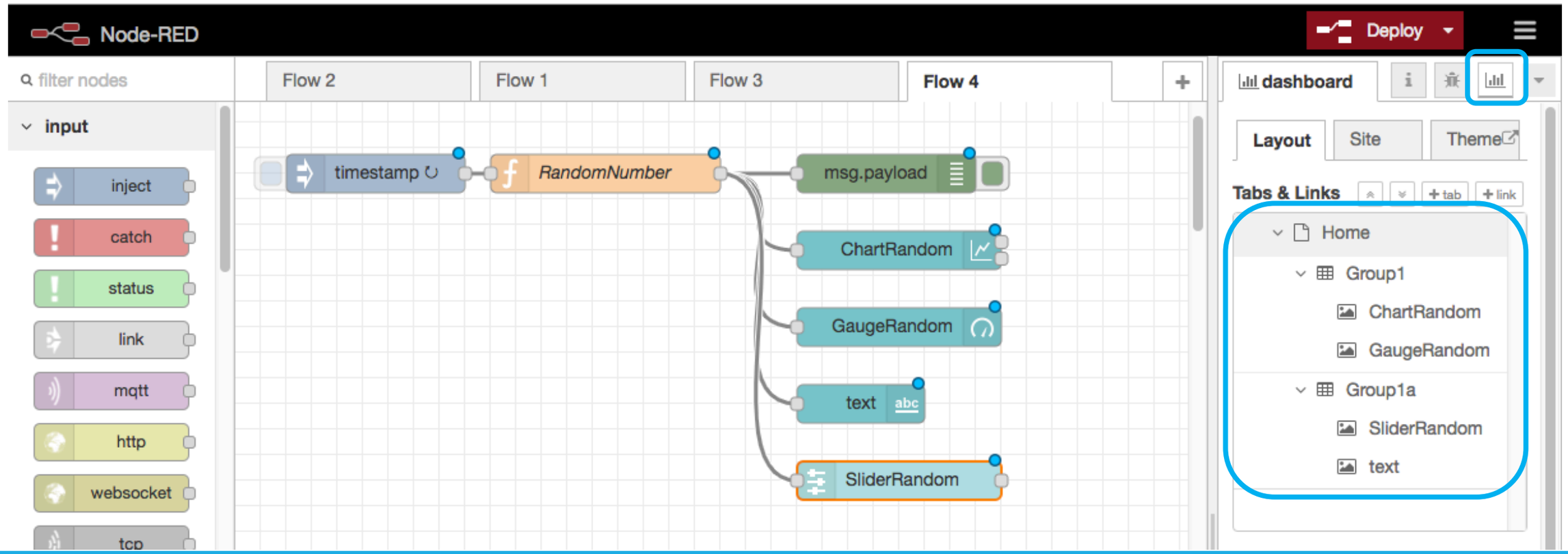
# Edit the Slider Node

- Group: select [Home] Group1a
- Range: 0 - 100
- Name: SliderRandom
- Hit “Done”



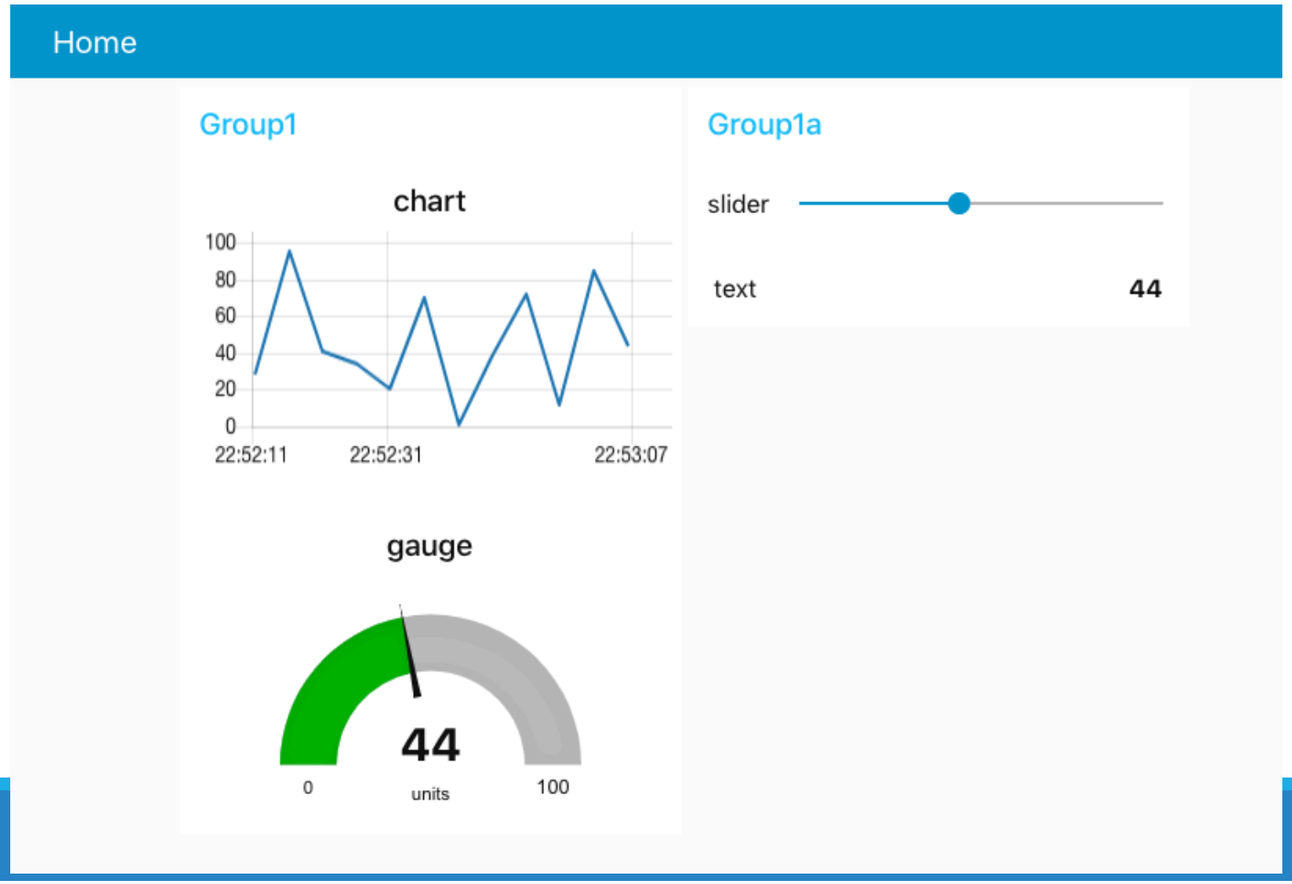
# See the Finished Product

- The dashboard icon (top right) reveals the order of our grouping on the 'Home' page



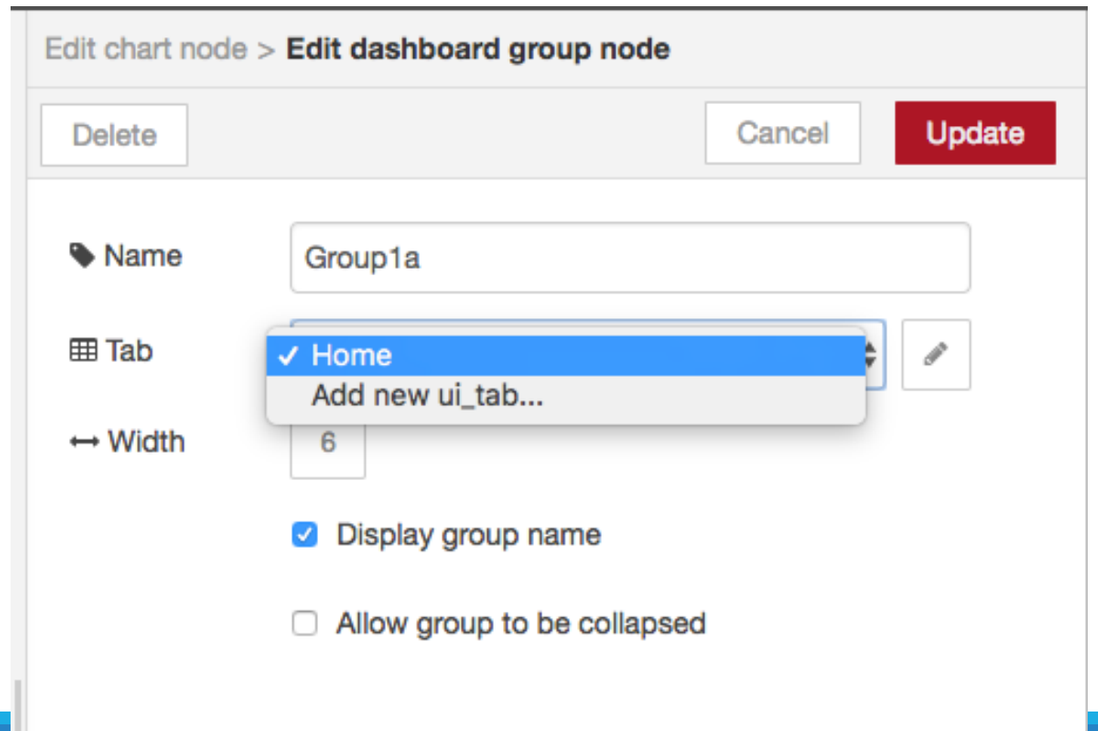
# Visit the UI Page

- Click “Deploy” and visit http://127.0.0.1:1880/ui/

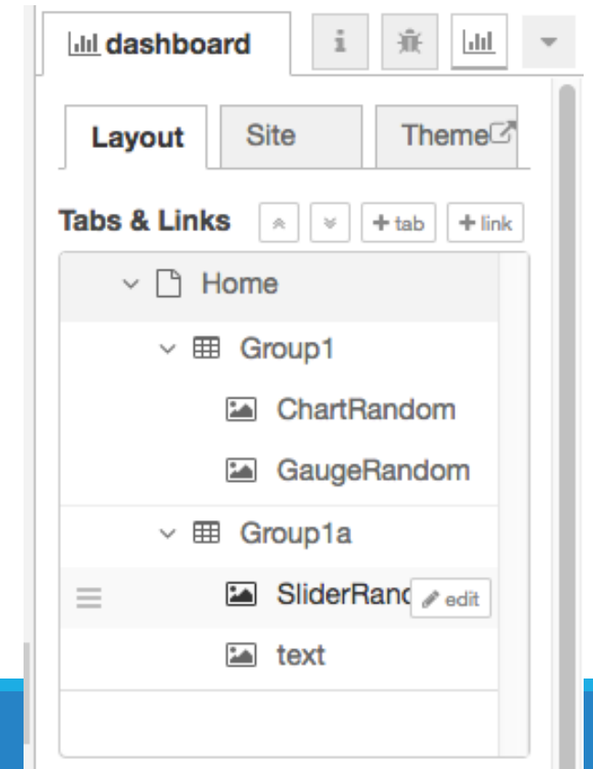


# Tips

- You can add as many UI pages as you like, by adding more “Tab”

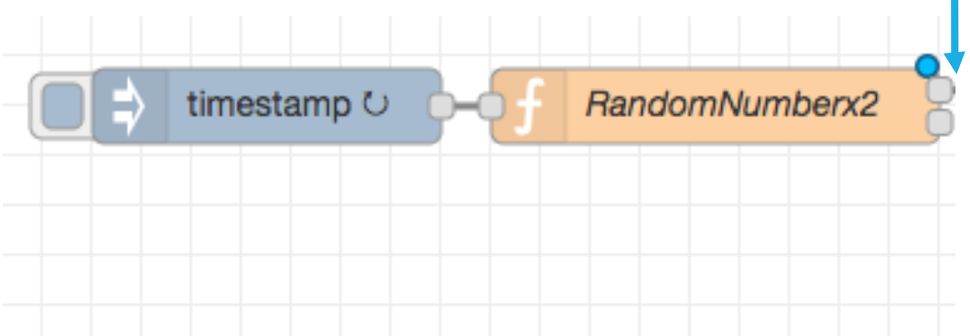


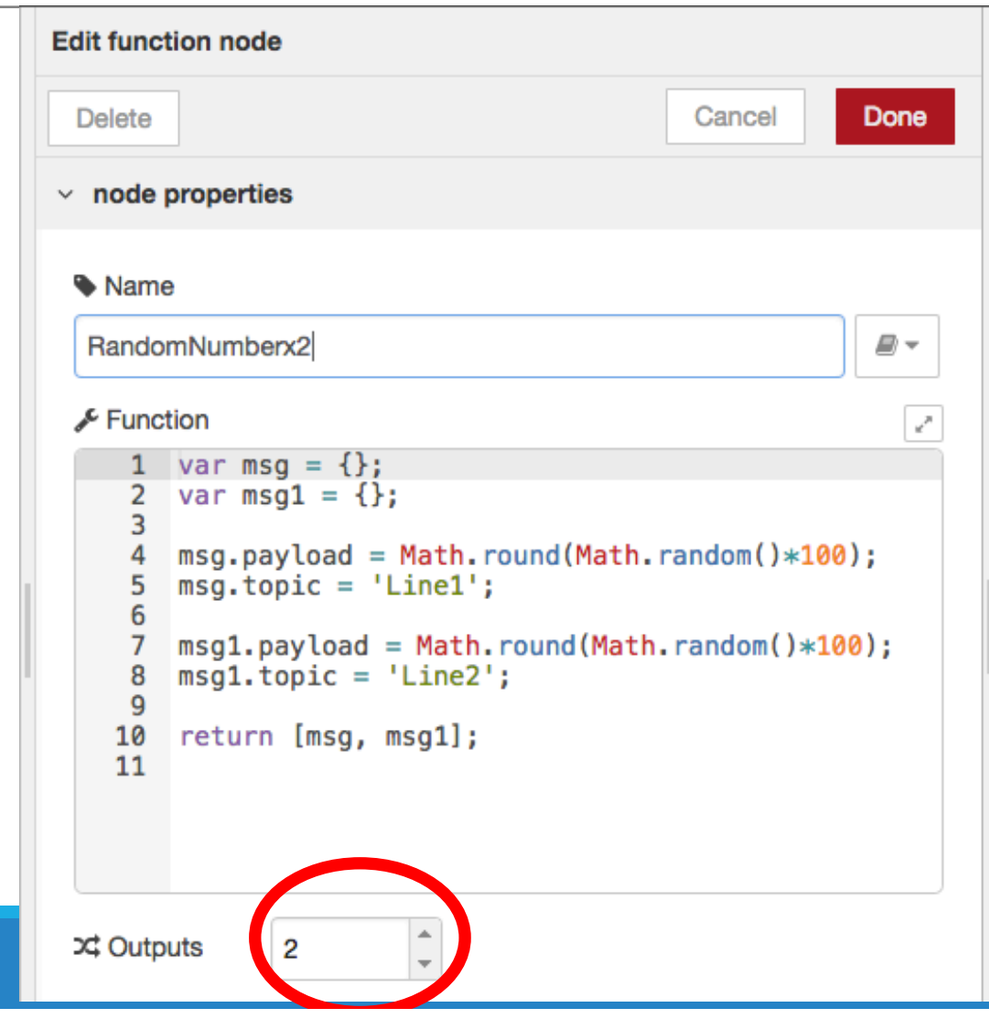

- You can re-order your chart elements under the dashboard icon by moving an element up or down



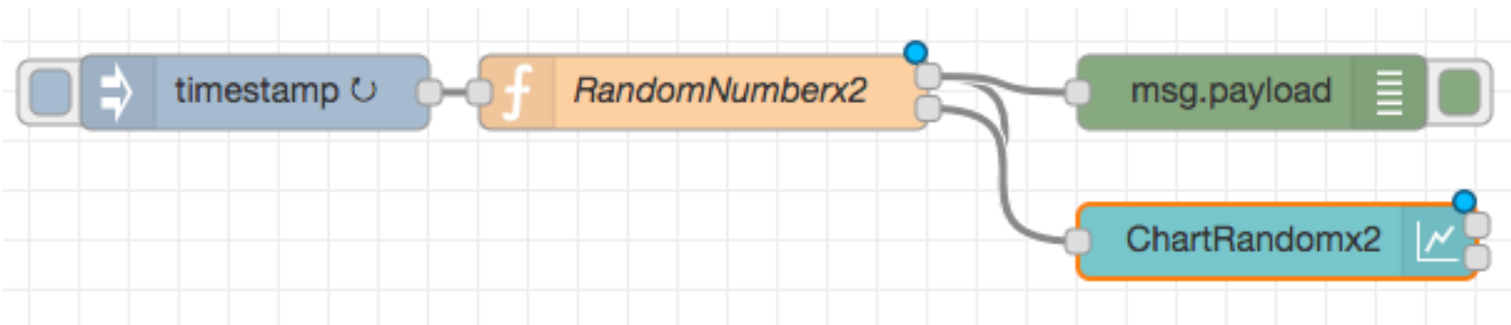
# UI Part 2: Plotting a Chart with Two Lines

- Edit your function node to generate two messages: **msg** and **msg1**
- Change '**Outputs**' to 2
- The function node then has two outputs



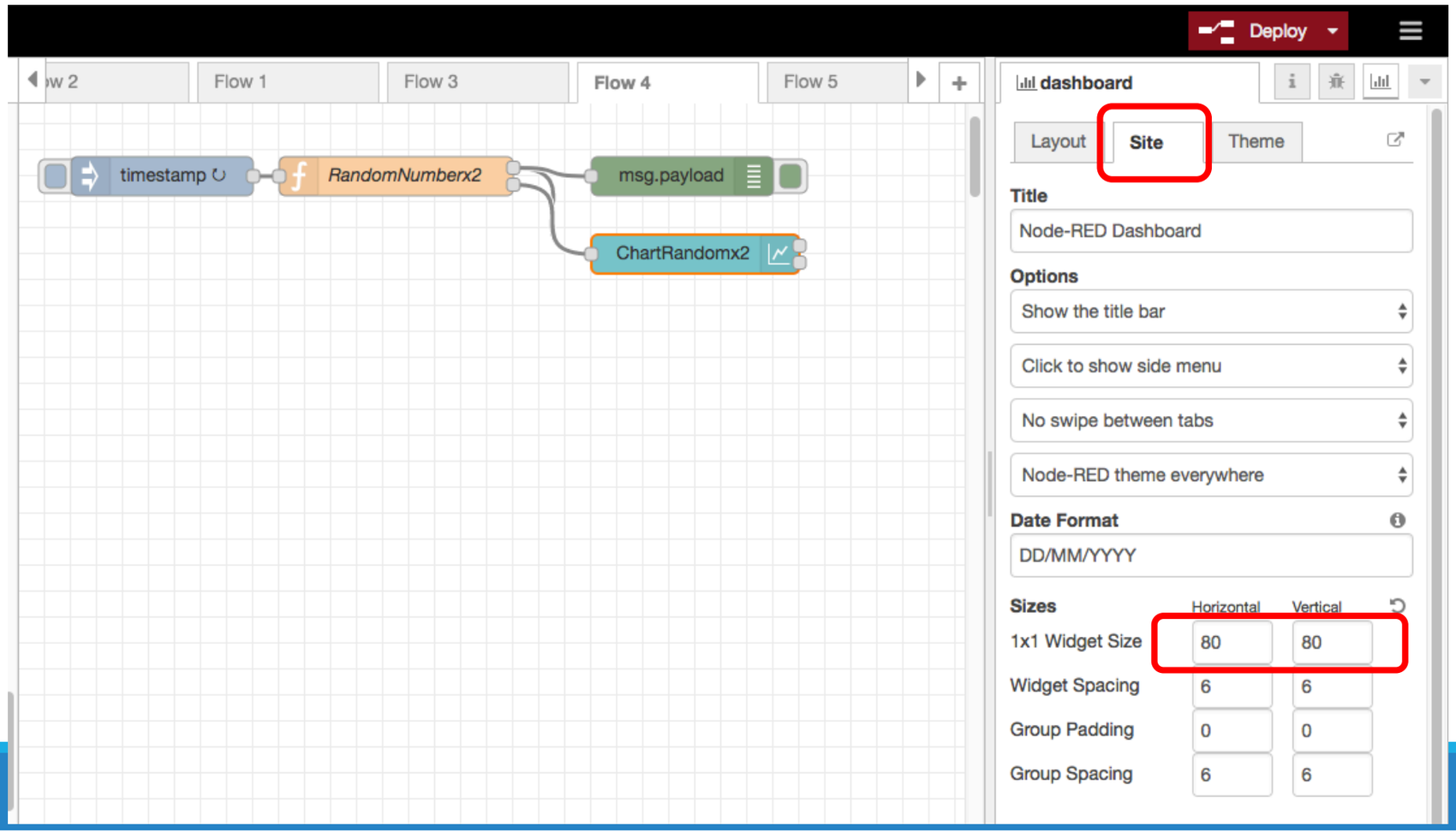



# Wire Outputs of the Function Node to the Input of the Chart Node

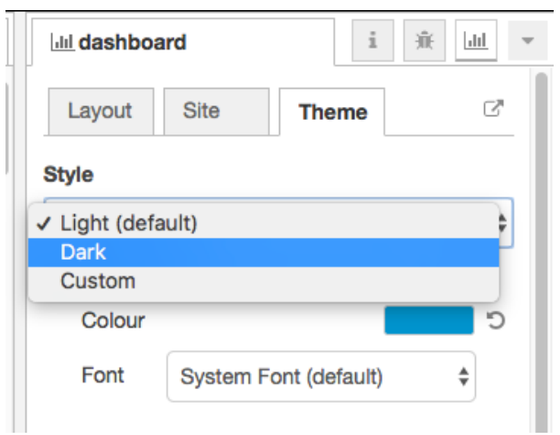



- Configure the Chart Node:
  - Group: [Home] Group1
  - Name: ChartRandomx2

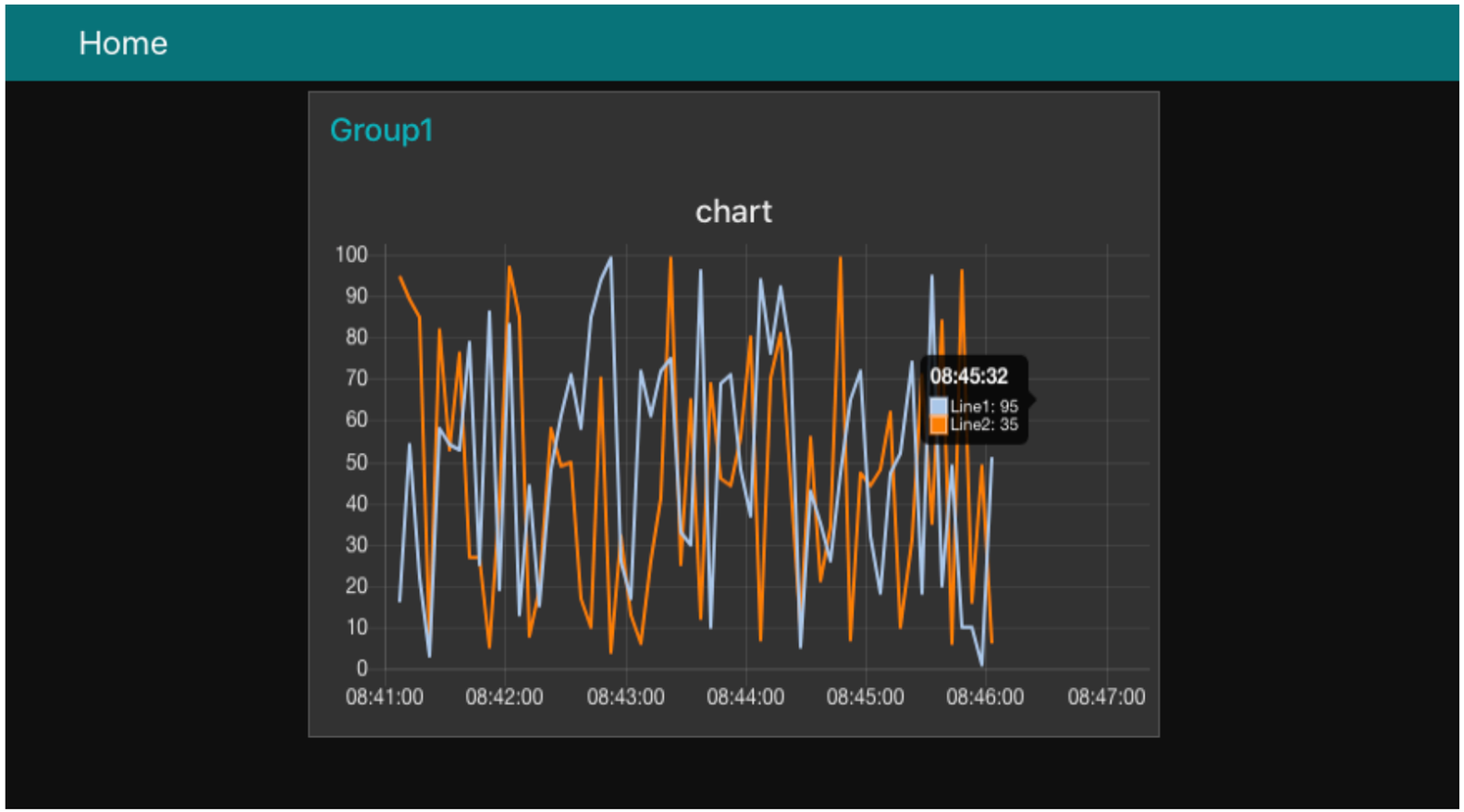
# Configure Chart Parameters




- Site tab, increase the Widget size
- Theme tab, try “Dark” theme

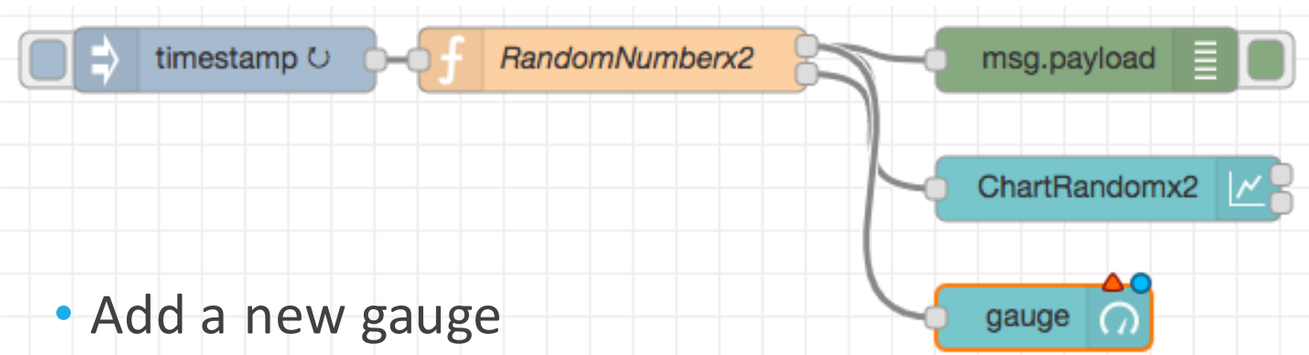


# Deploy and See the UI

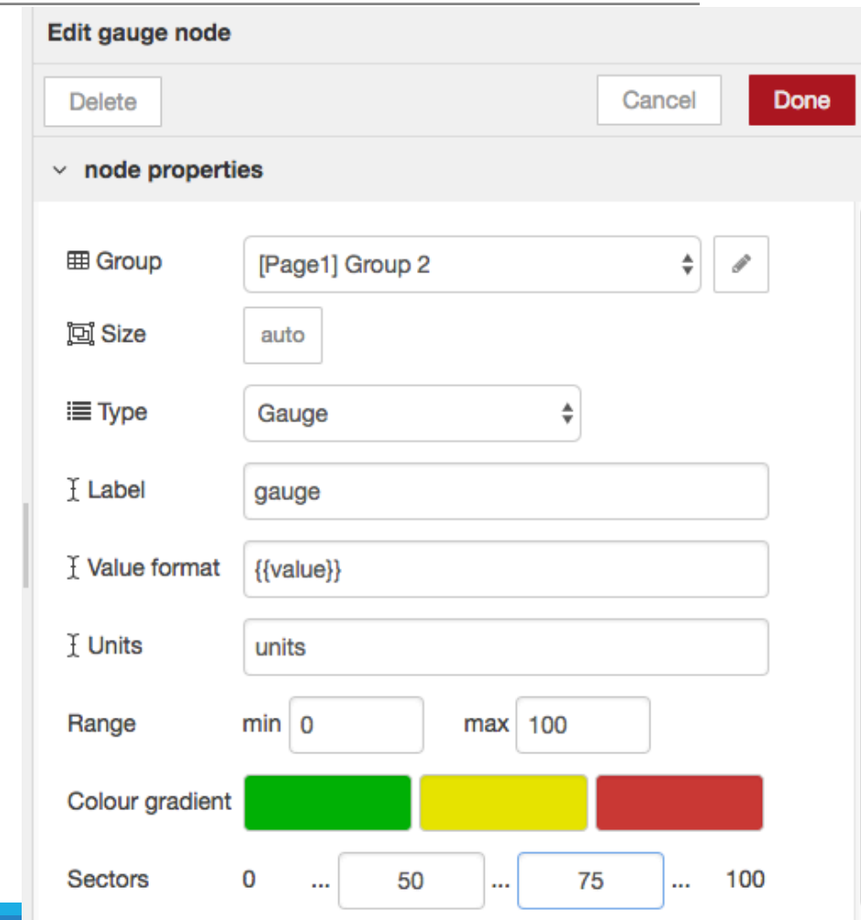



http://127.0.0.1:1880/ui/

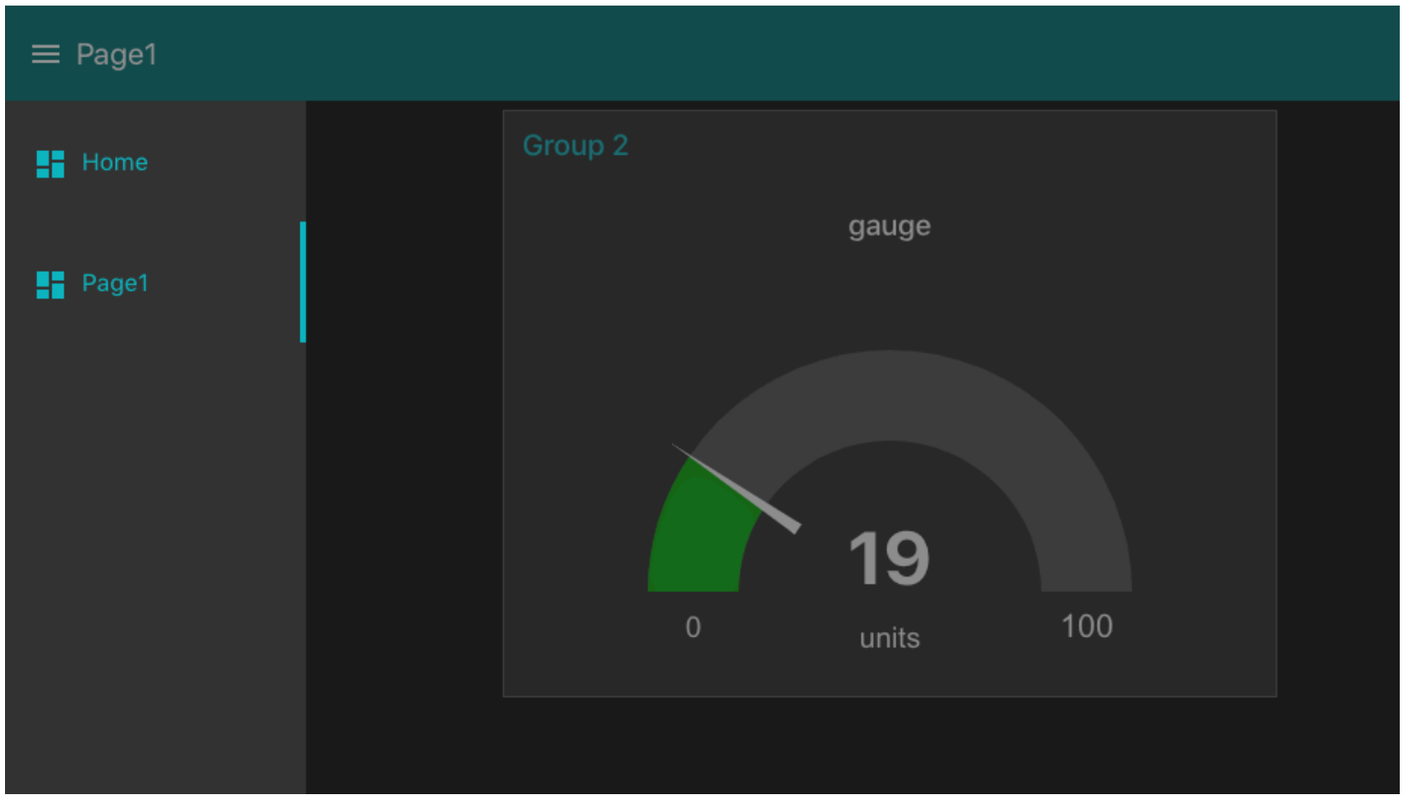
# UI Part 3: Adding the Second UI Page




- Add a new gauge
- Configure the gauge by selecting “Add new ui_group”, click the pen icon to edit, enter…
  - Name: Group 2
  - Tab: Page1 (need to edit)
  - Update range: 0-100
  - Update sectors: 0…50…75
  - Click “Add” button



# Deploy and See the UI




http://127.0.0.1:1880/ui/

Notice the hamburger that allows you to switch between UI pages

# Workshop 5. MQTT

We will learn how to publish data to MQTT broker; and subscribe to a topic from MQTT broker

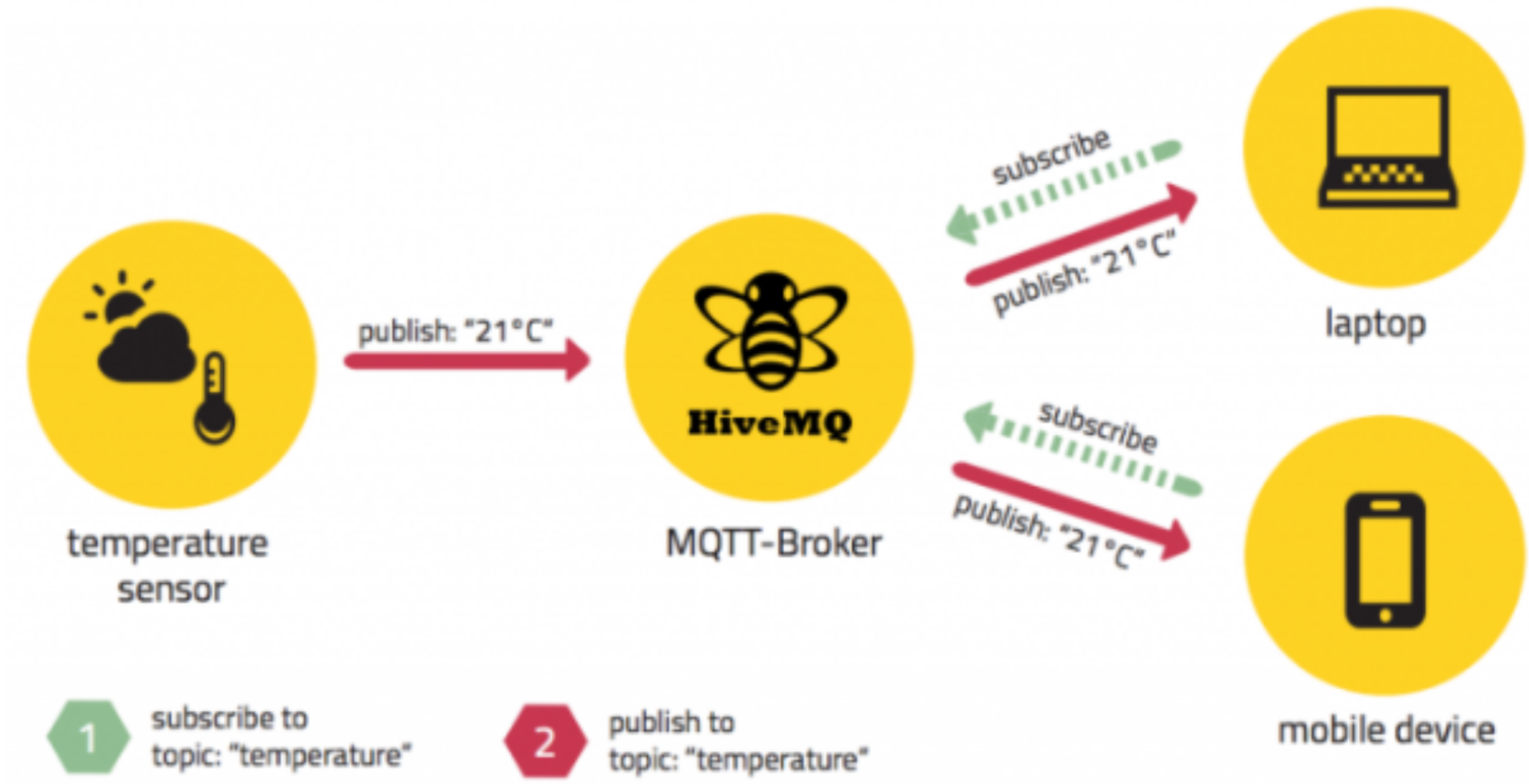
# What is MQTT?

MQTT stands for **MQ Telemetry Transport** and it is a nice lightweight publish and subscribe system where you can publish and receive messages as a client.

It is a simple messaging protocol, designed for constrained devices and with low-bandwidth.

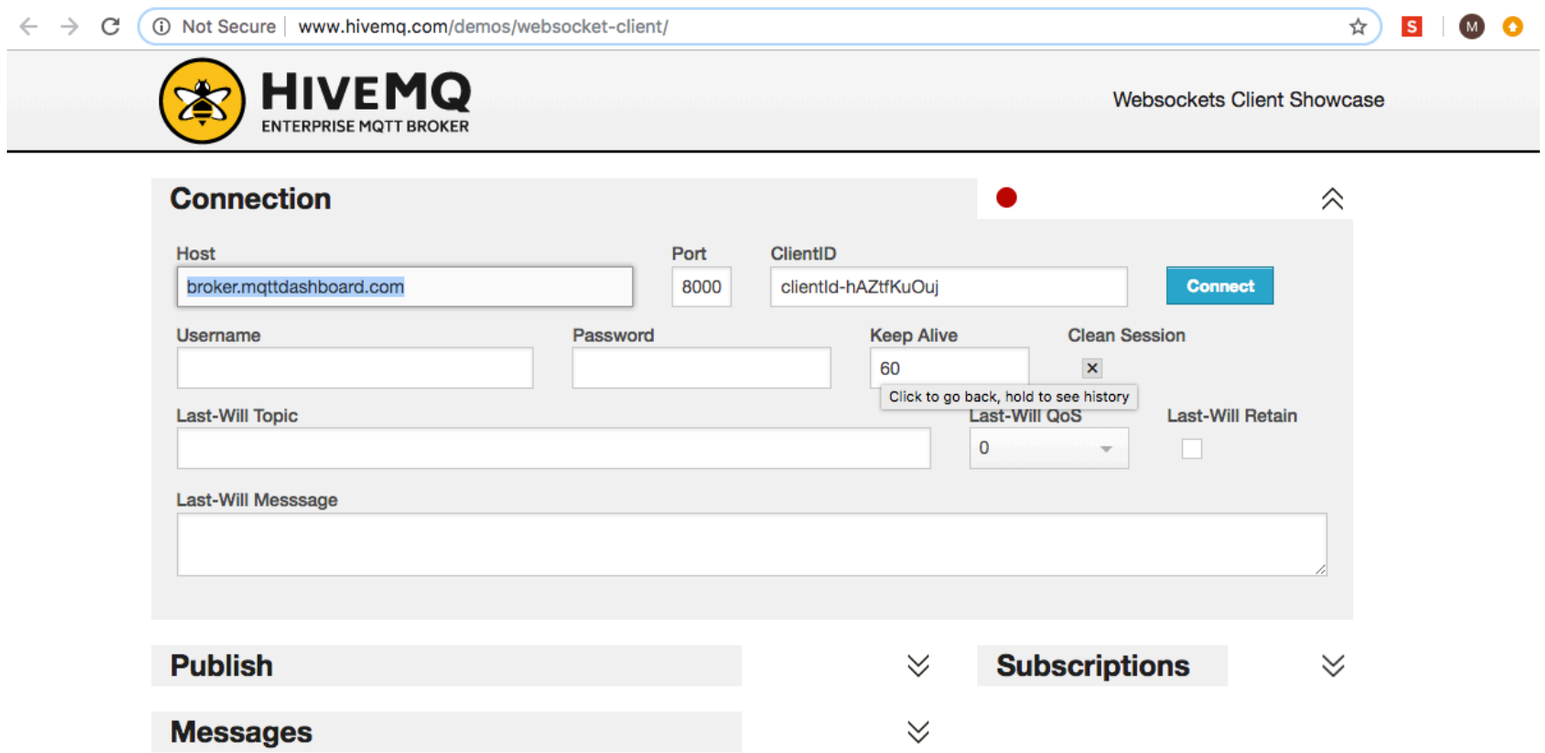
So, it's the perfect solution for Internet of Things applications.

# Publish/Subscribe Concept



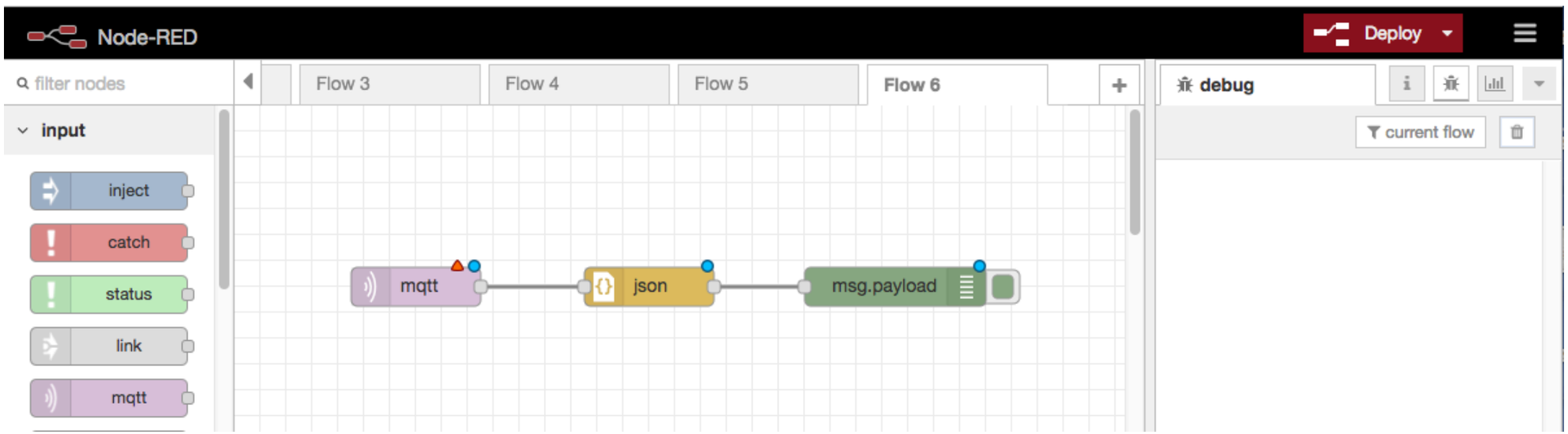
# We will Use Free MQTT Broker

http://www.hivemq.com/demos/websocket-client/



# Create the Flow

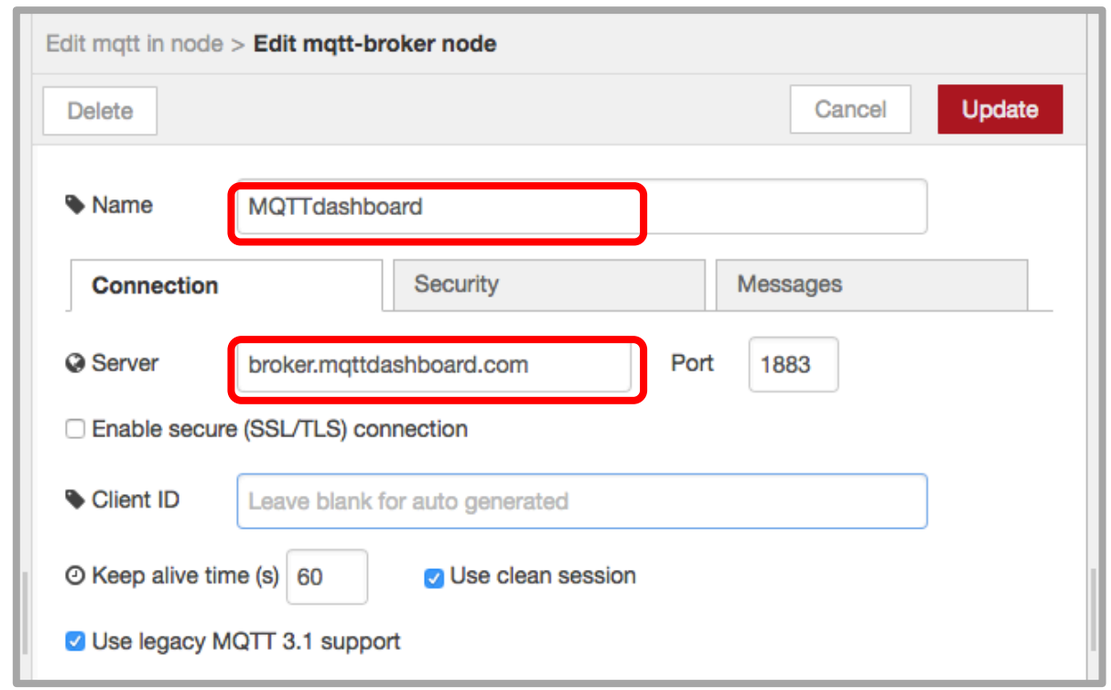
- Drag & drop: the **mqtt** node (input), the **json** node (function), the **debug** node (output).
- Wire them together as shown.



# MQTT node – Configure the Server

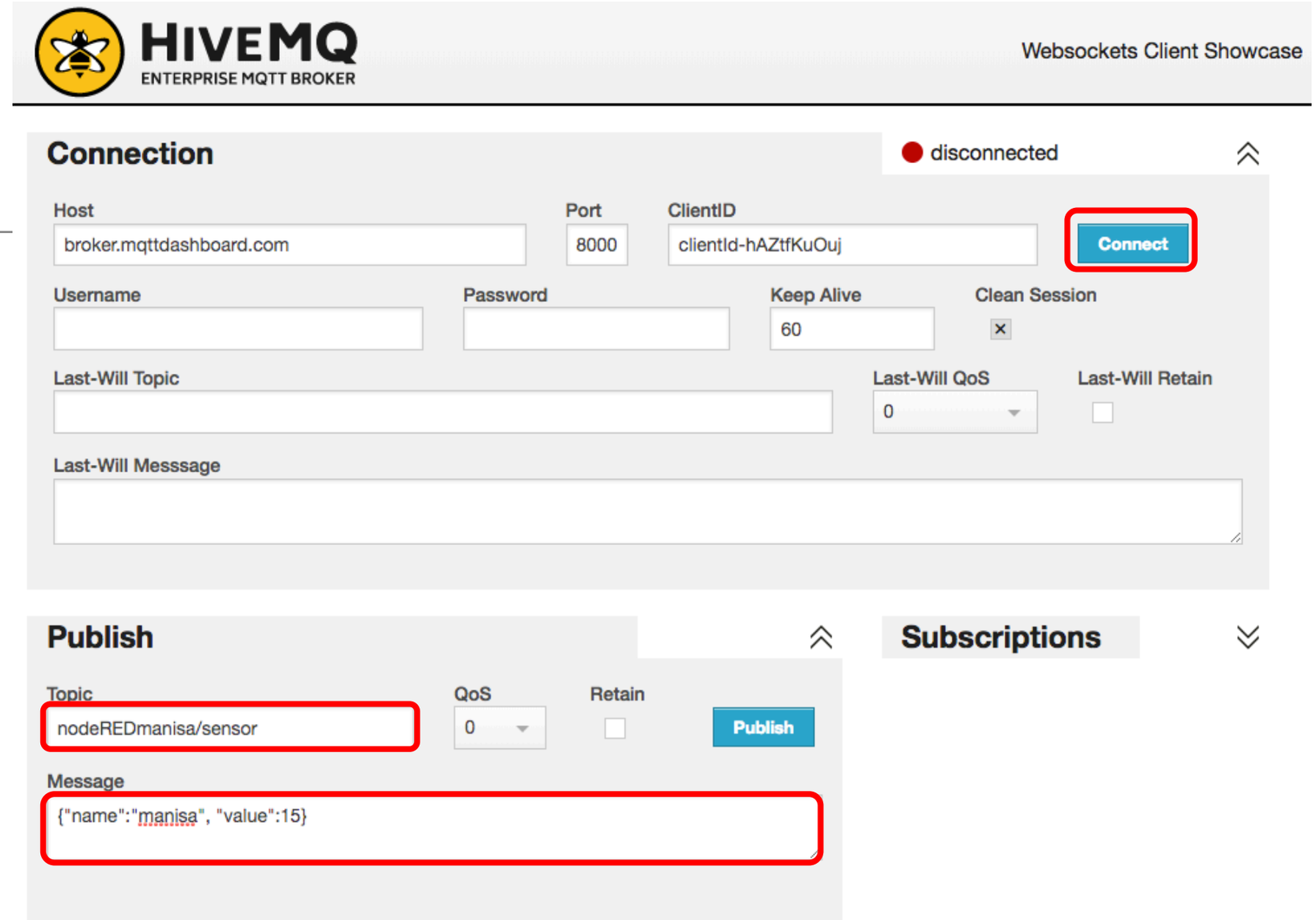
- Configure the **mqtt** node
- Under Server, edit "Add new mqtt-broker…"
  - Add Name: <u>any name</u>
  - Add Server: <u>from HiveMQ/Host</u>
  - Add Client ID: <u>any name or leave blank</u>
  - Click "Add"
- Topic: **<u>any unique topic name</u>**
- Name: MQTTdashboard
- Click "Done"

Add MQTT broker connection

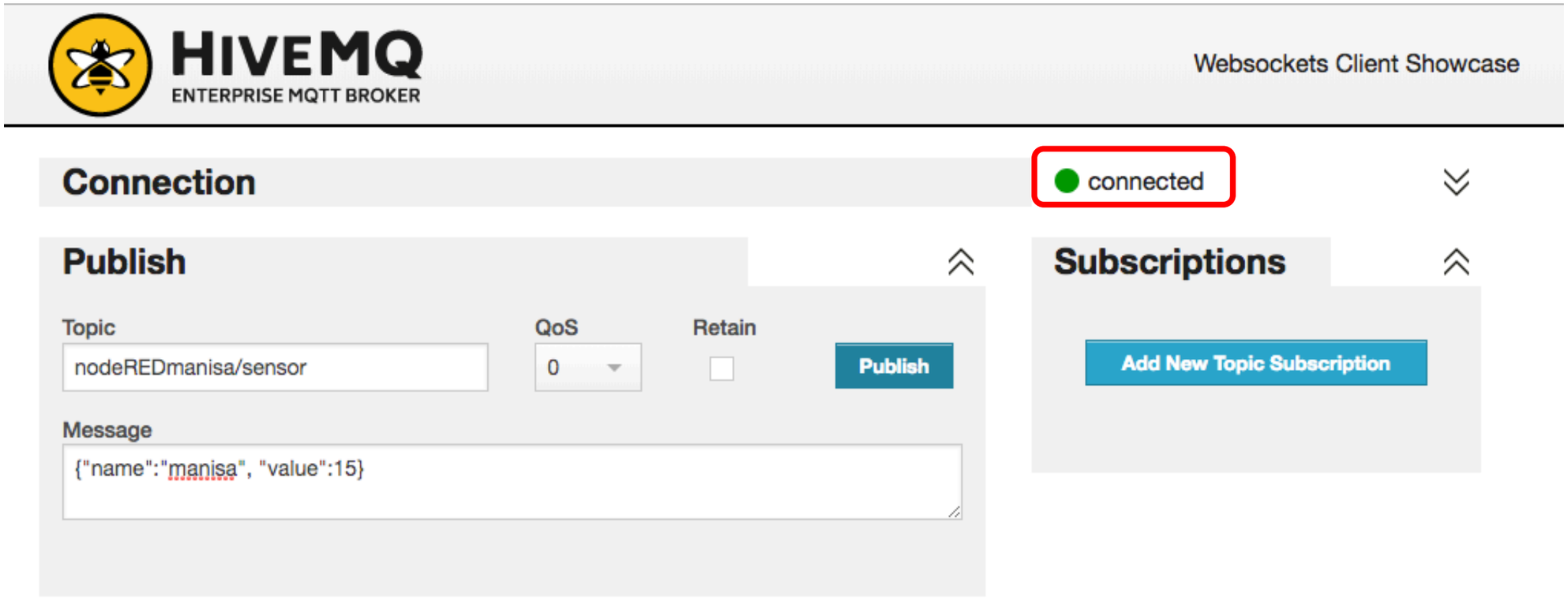


# Configure HiveMQ

- Under Publish:
  - Topic: Enter the same unique topic
  - Message: Enter a json message
- Under Connection:
  - Click “Connect” button

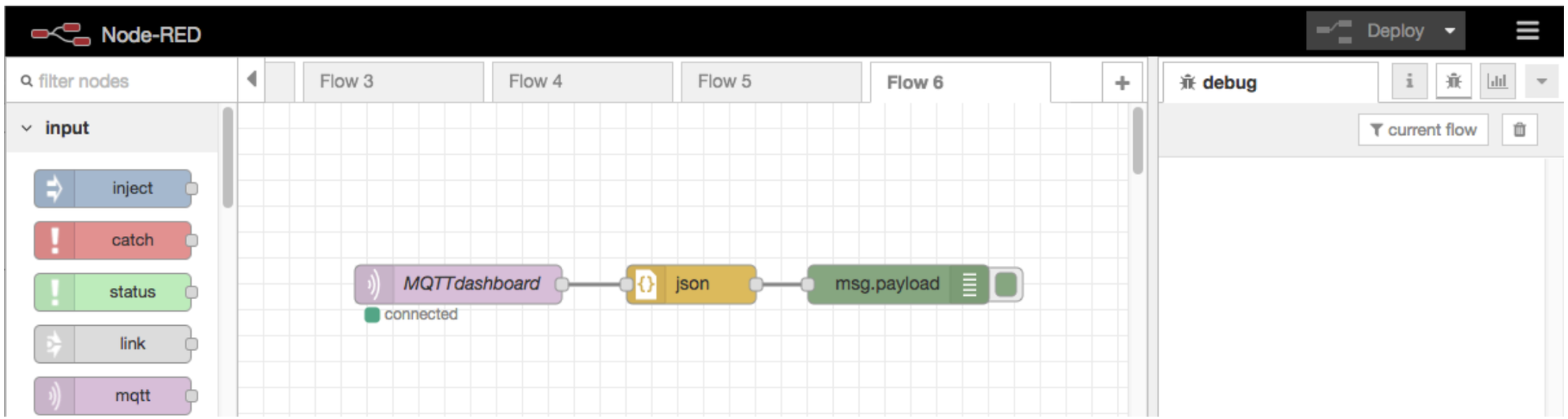


# HiveMQ is now Connected

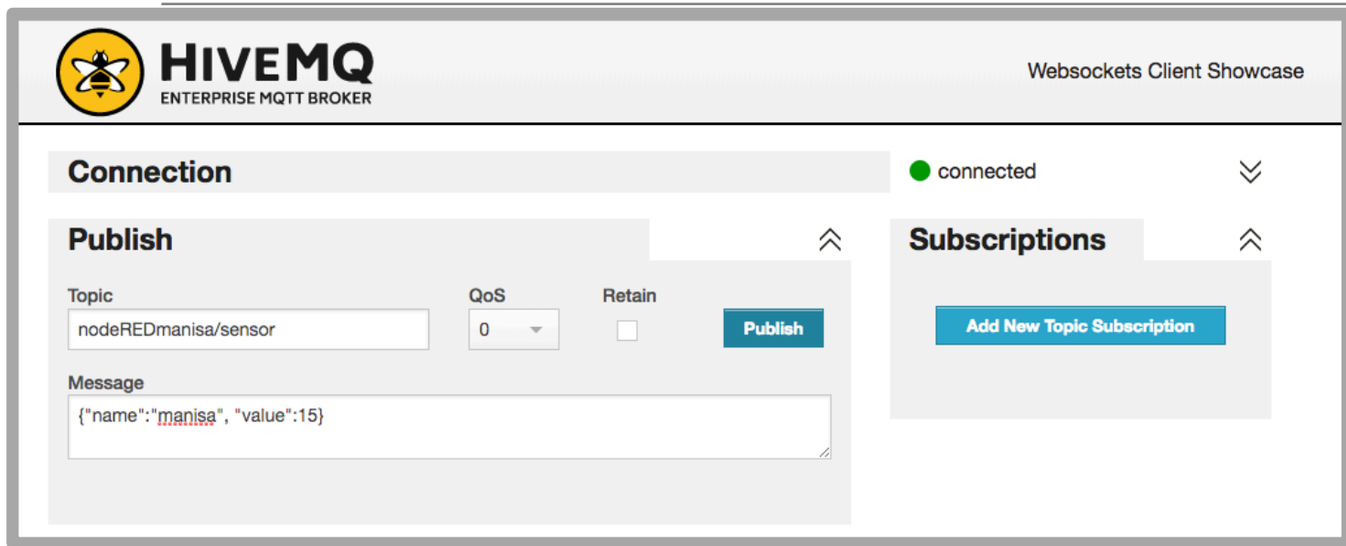


# Deploy the Flow in NodeRED

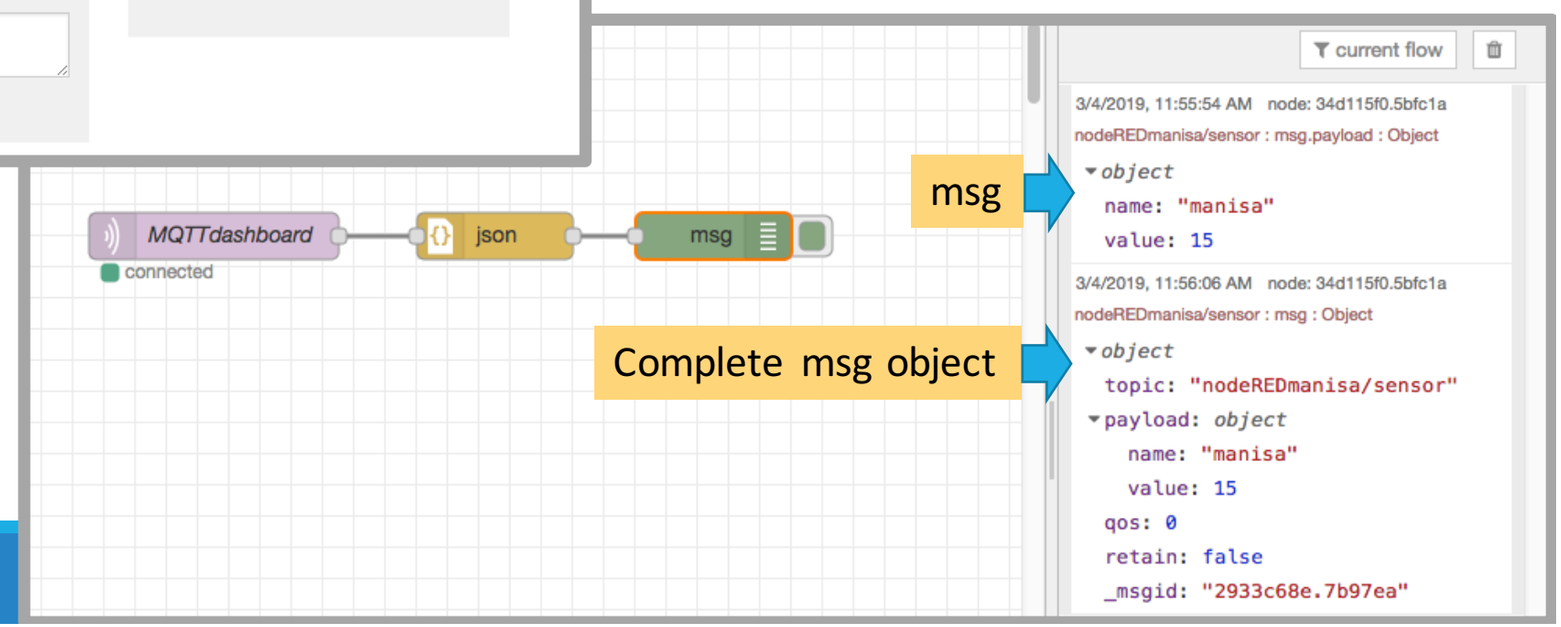
- In NodeRED, hit 'Deploy', MQTTdashboard is now connected.



# Click “**Publish**” in HiveMQ, See the Debug Message




Examine the debug node options: ‘msg’ vs ‘complete msg object’



# Congratulations!

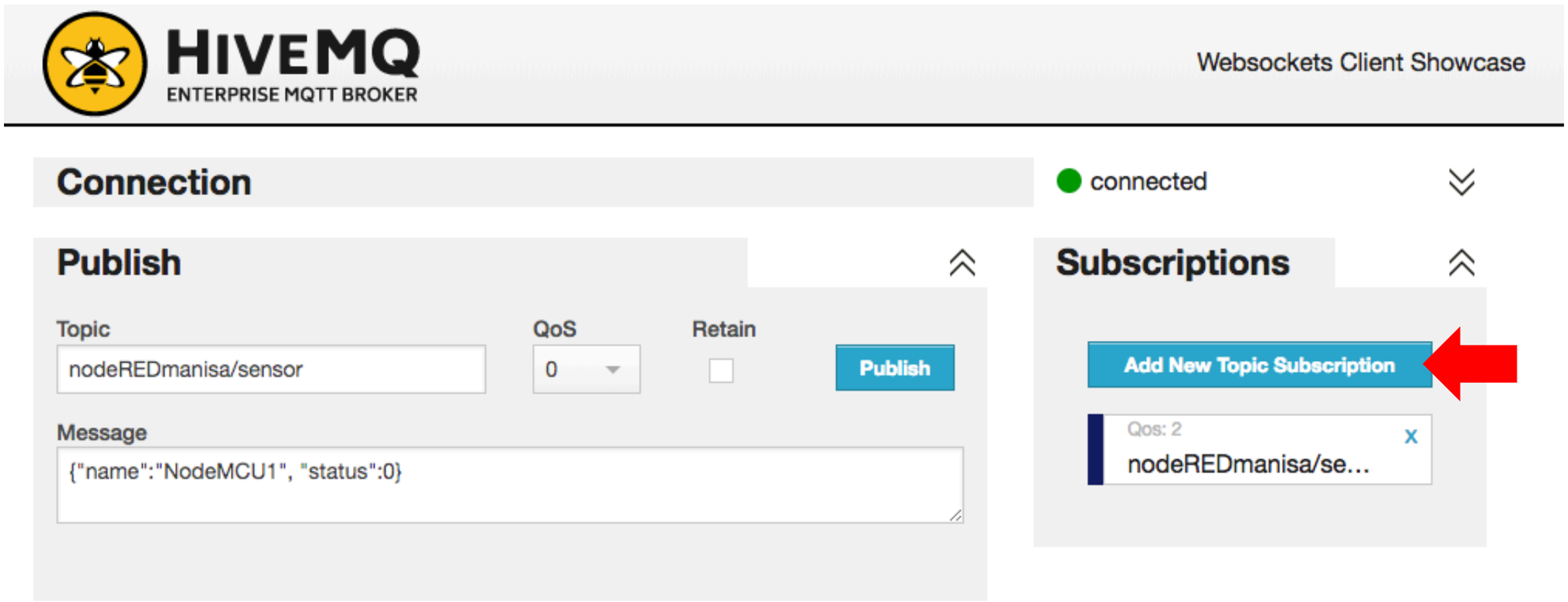
You can now subscribe to your topic from an MQTT broker.

**Applications:**

- You can subscribe to any topic and get information from anywhere
- You can subscribe to your NodeMCU that reads temperature/humidity data
- You can subscribe to as many NodeMCU as you like
- These data can be saved to a database, plot on a dashboard and can be used to control anything!

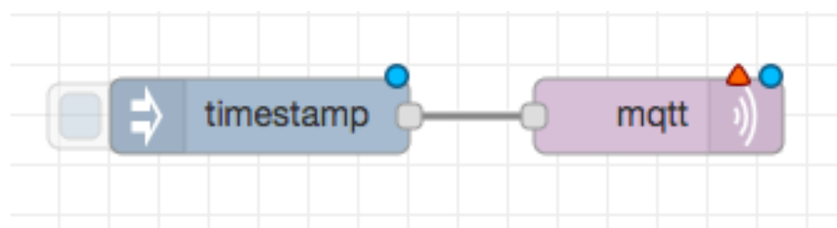
# Part 2: Send Data to MQTT Broker

In HiveMQ, click “Add New Topic Subscription”



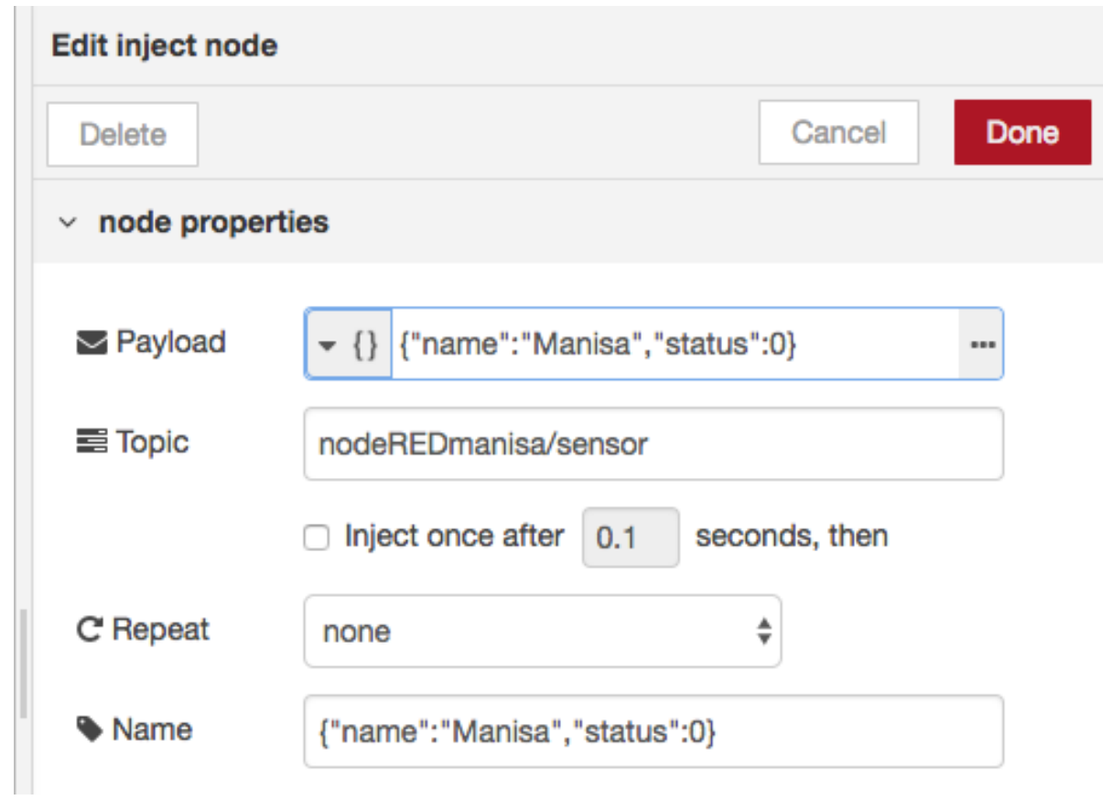
# Create the Flow

- Drag & drop: the **inject** node (input) and the **mqtt** node (output).
- Wire them together as shown.



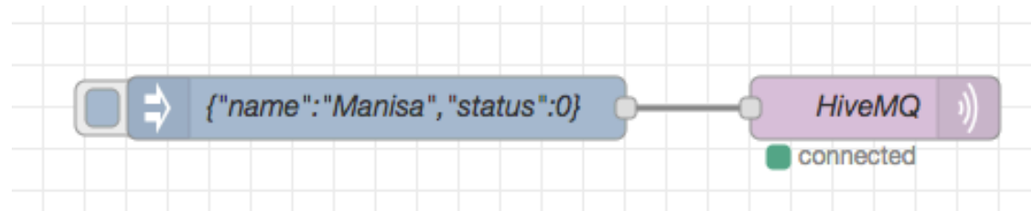
# Configure the Inject Node

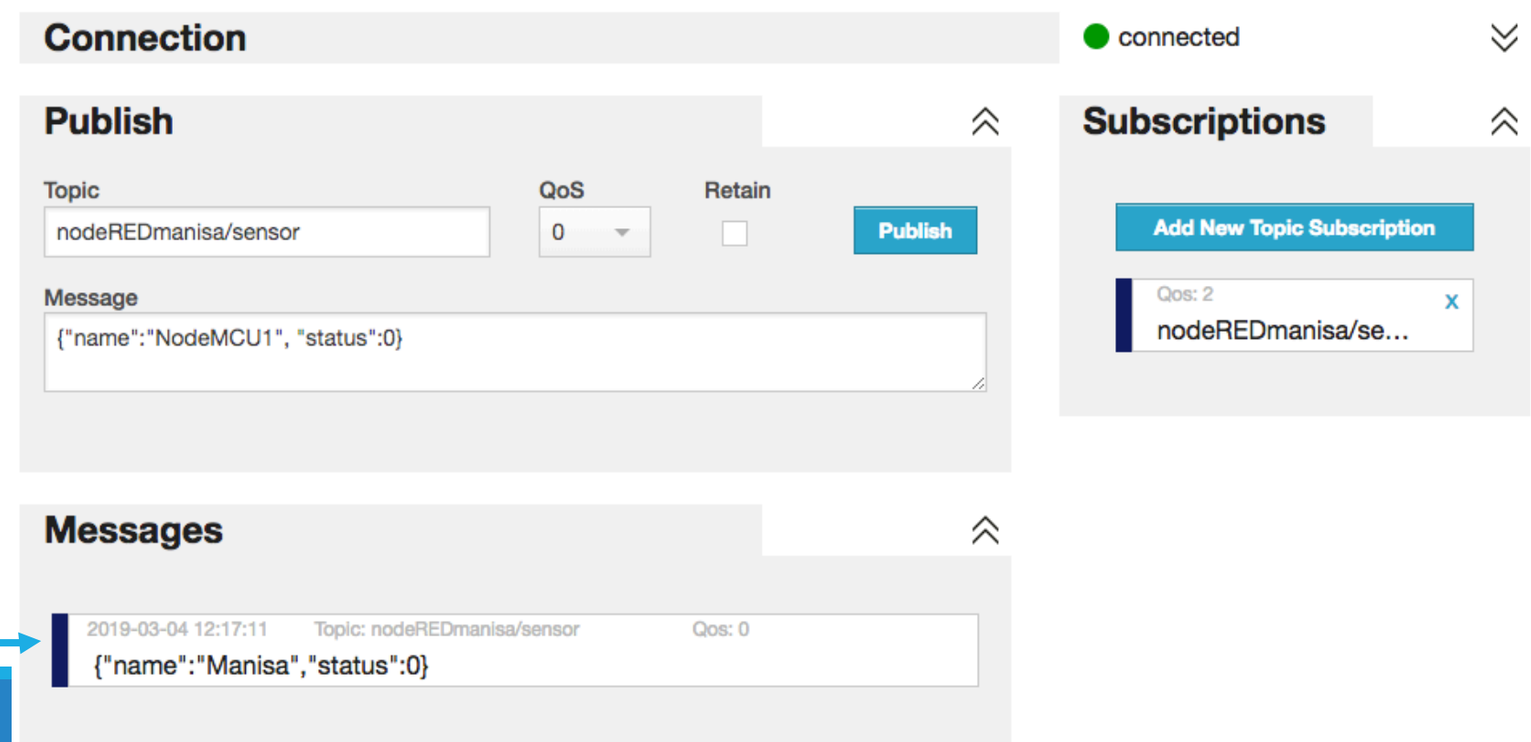
- Payload: json, and type any json message
- Topic: the same topic as you specify in HiveMQ (subscription)
- Name: any name



# Configure the MQTT Node

- Hit deploy, “HiveMQ” shows connected
- Click the inject button
- Check HiveMQ for the received message



# Congratulations!

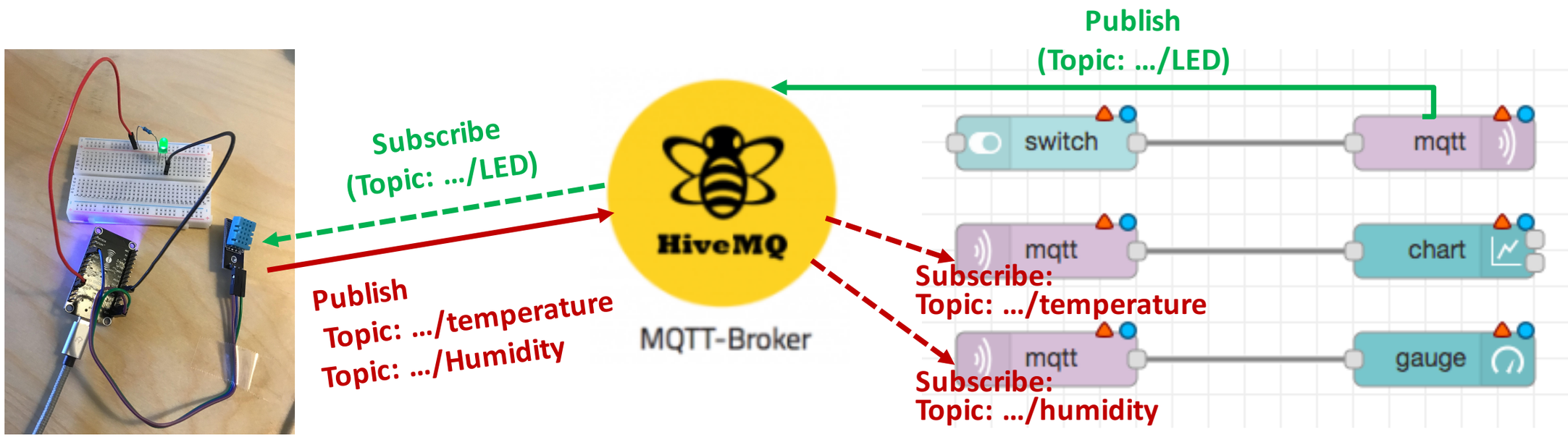
You can now send a message to your MQTT broker.

# **Workshop 6.** Arduino, MQTT and NodeRED

We will learn how to publish data from Arduino (NodeMCU-DHT11) and subscribe to the data we publish, using MQTT and NodeRED. We will also use MQTT to control an LED.

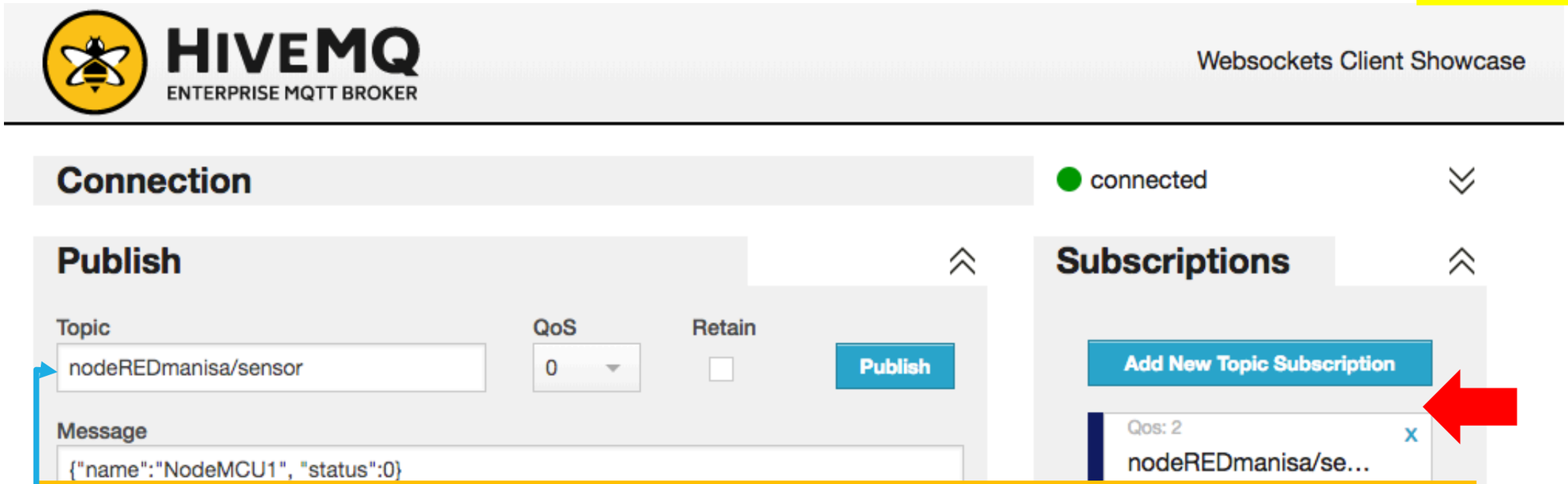
Github: https://github.com/ICECapstone/MQTT_PM25_DHT11

# Concept


Publish
(Topic: …/LED)
Subscribe
(Topic: …/LED)
HiveMQ
MQTT-Broker
Publish
Topic: …/temperature
Topic: …/Humidity
switch
mqtt
mqtt
chart
Subscribe:
Topic: …/temperature
mqtt
gauge
Subscribe:
Topic: …/humidity

# Set up Pub/Sub Topics

GroupName:
- FlyingBird2
- InkzandFriends
- LockerSwarm
- MonkolTeam
- Parking
- vCanteen



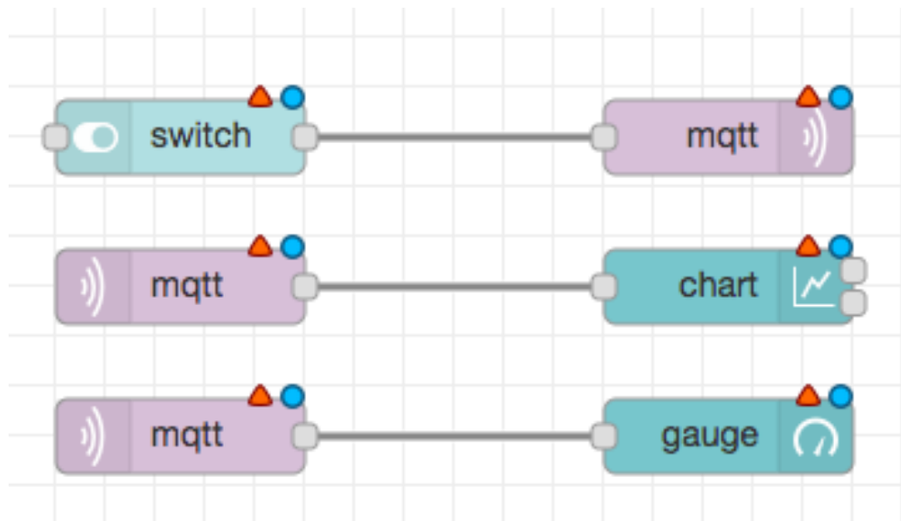

Specify **two subscribe topics**: one for getting temperature data & one for getting humidity data. Please use unique topic name. Suggestion: use your group name.
- GroupName/temperature
- GroupName/humidity

Also, specify **one publish topic** for controlling an LED:
- GroupName/LED

# Create the Flow

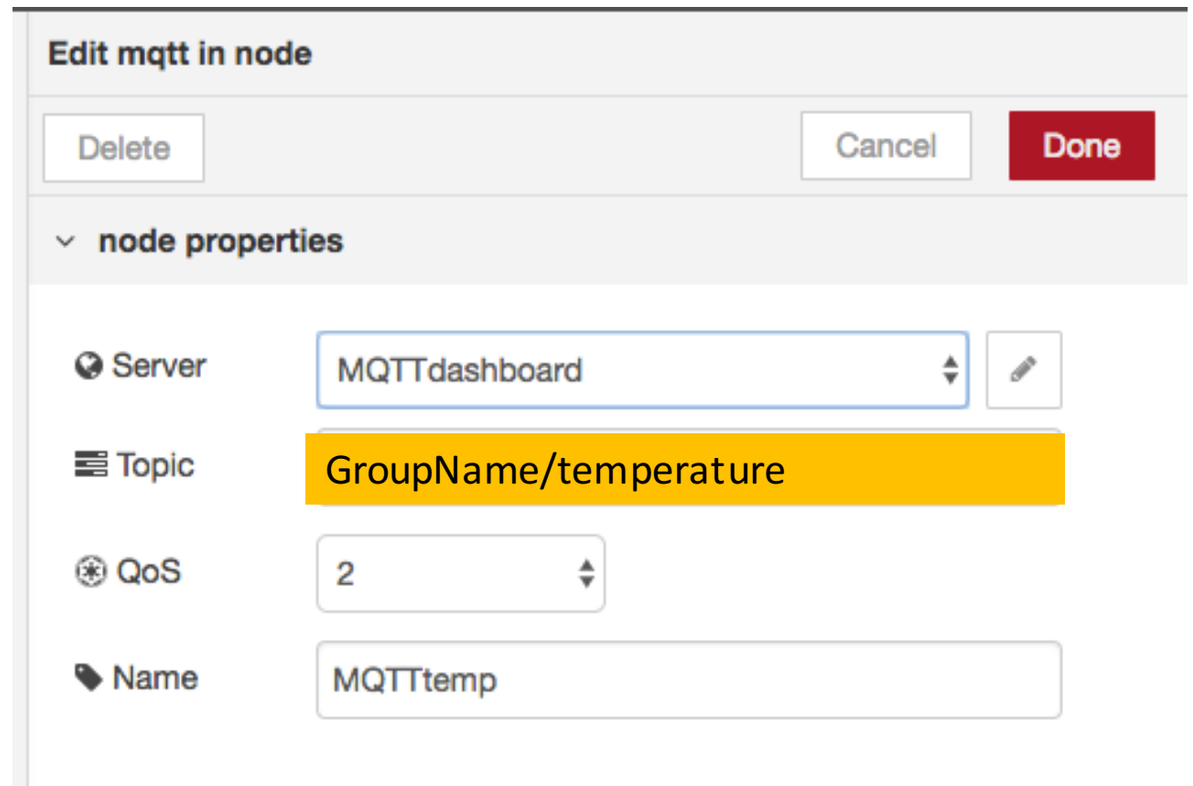
•Drag & drop: two **mqtt** nodes (input), **switch, chart, gauge** nodes (dashboard), one **mqtt** node (output).

• Wire them together as shown.



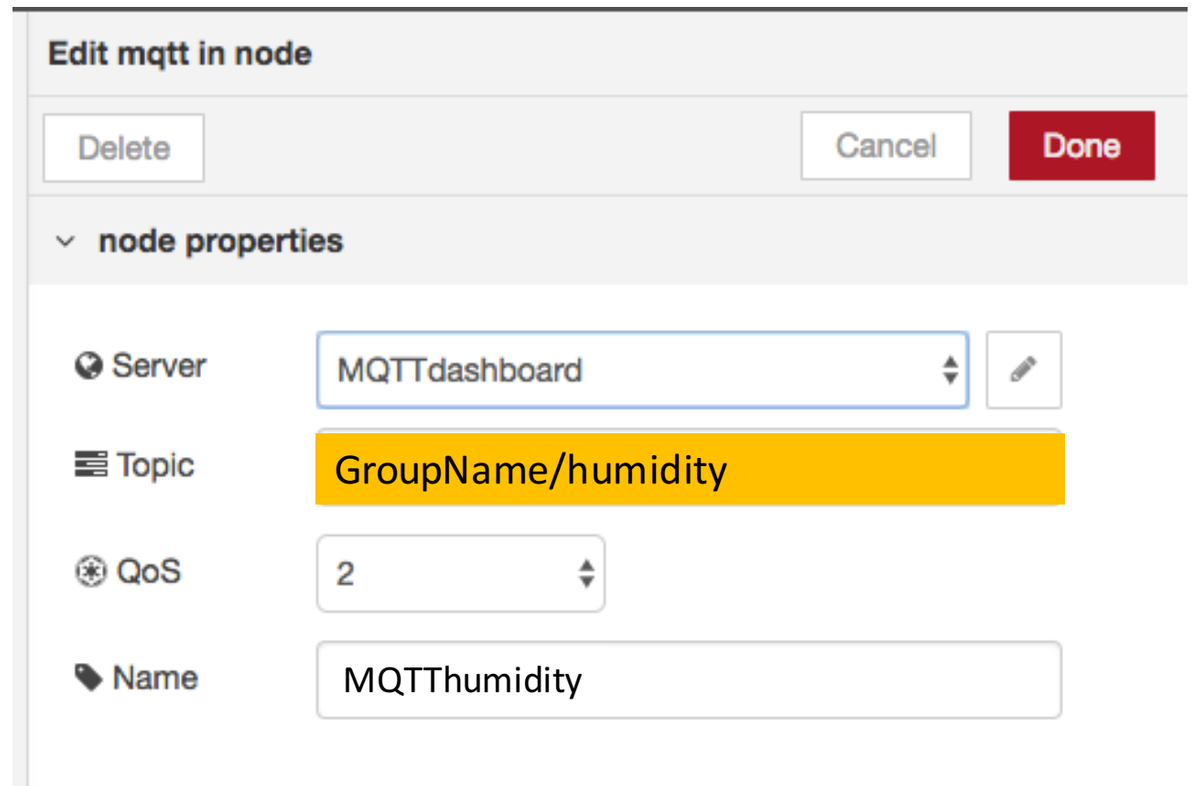
# Configure the 1st mqtt Input Node

- Configure the **mqtt** node to subscribe to temperature data
- Under Server,
  - user the same server as the previous exercise
- Topic: **unique topic name (to subscribe to temperature topic)**
- Name: MQTTtemp
- Click "Done"



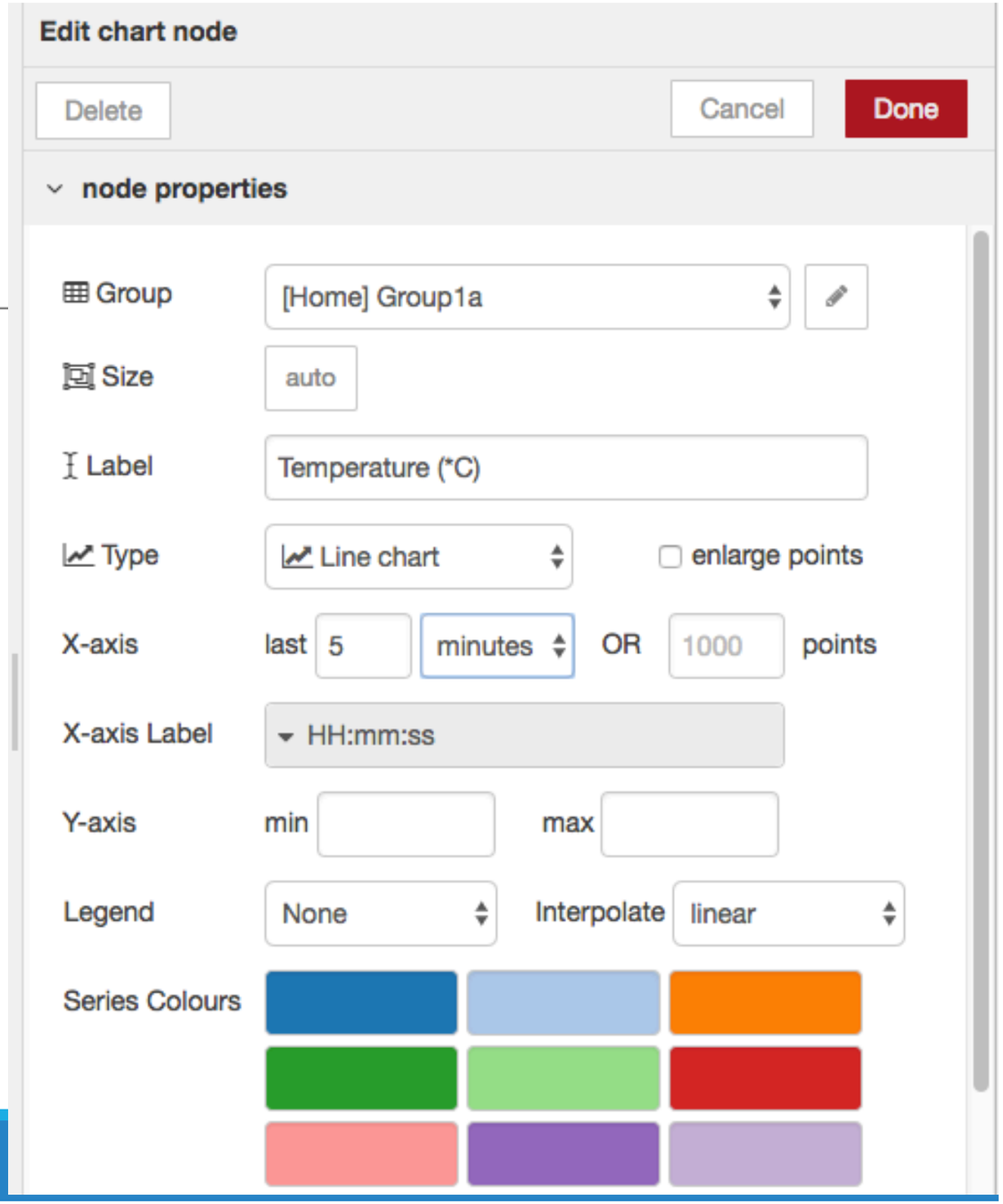
# Configure the 2nd mqtt Input Node

- Configure the **mqtt** node to subscribe to humidity data
- Under Server,
  - use same server
- Topic: **any unique topic name (to subscribe to humidity topic)**
- Name: MQTThumidity
- Click "Done"



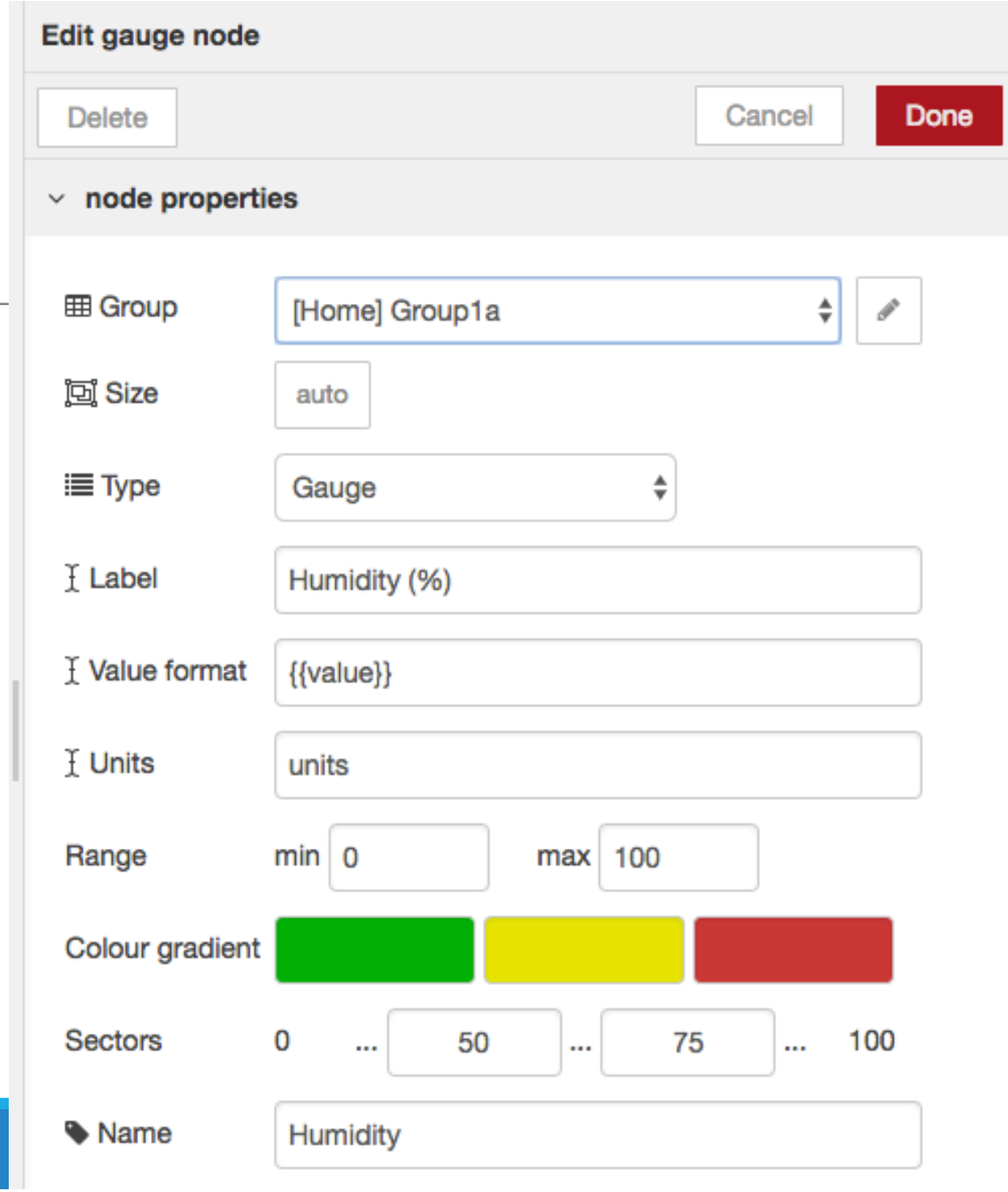
# Edit the Chart Node

- Group:
  - Please feel free to use the existing or create a new UI group
- X-axis: last 5 minutes
- Name: Temperature
- Hit “Done”



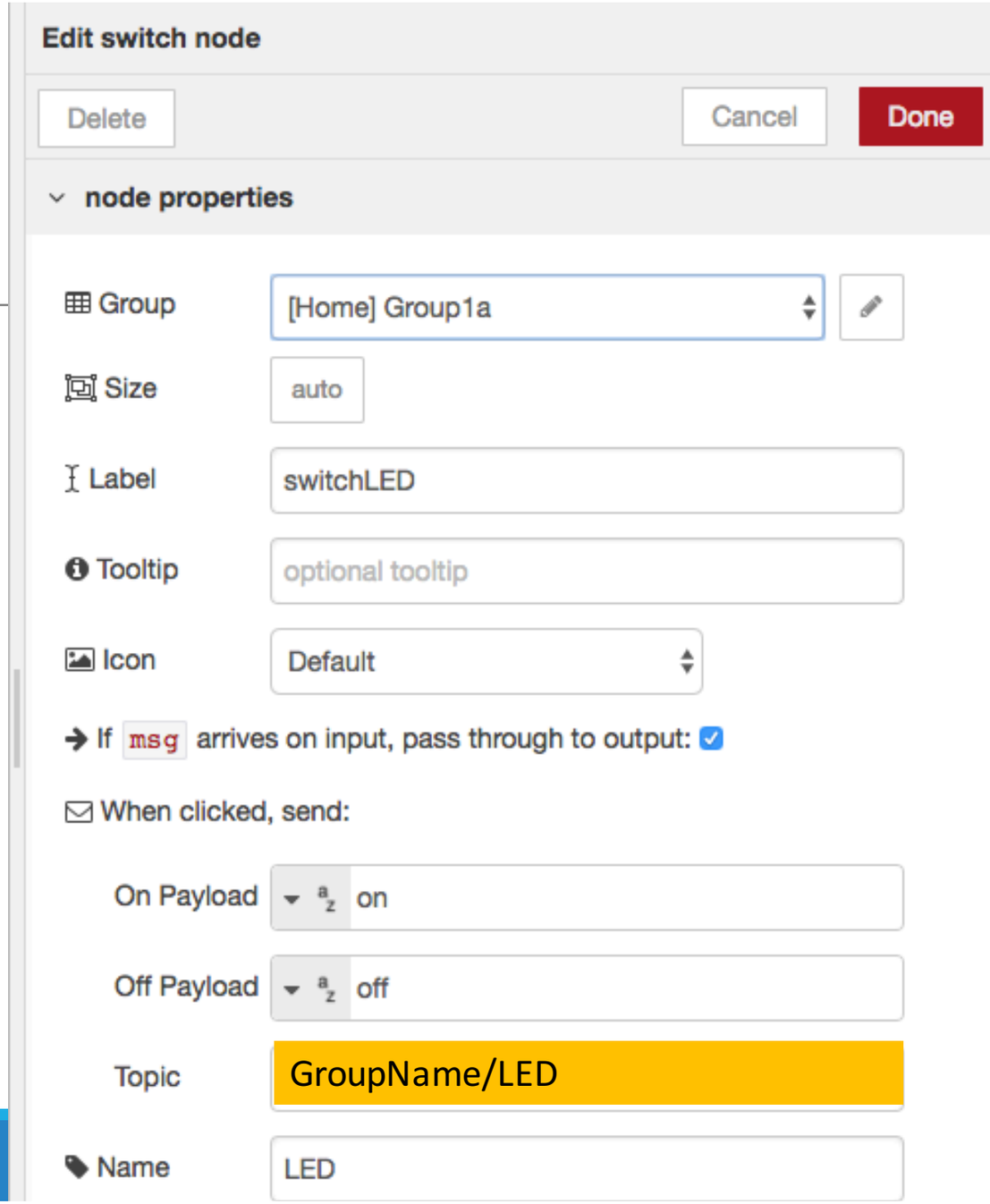
# Edit the Gauge Node

- Group:
  - Please feel free to use the existing or create a new UI group
- Range: 0 - 100
- Name: Humidity
- Hit “Done”



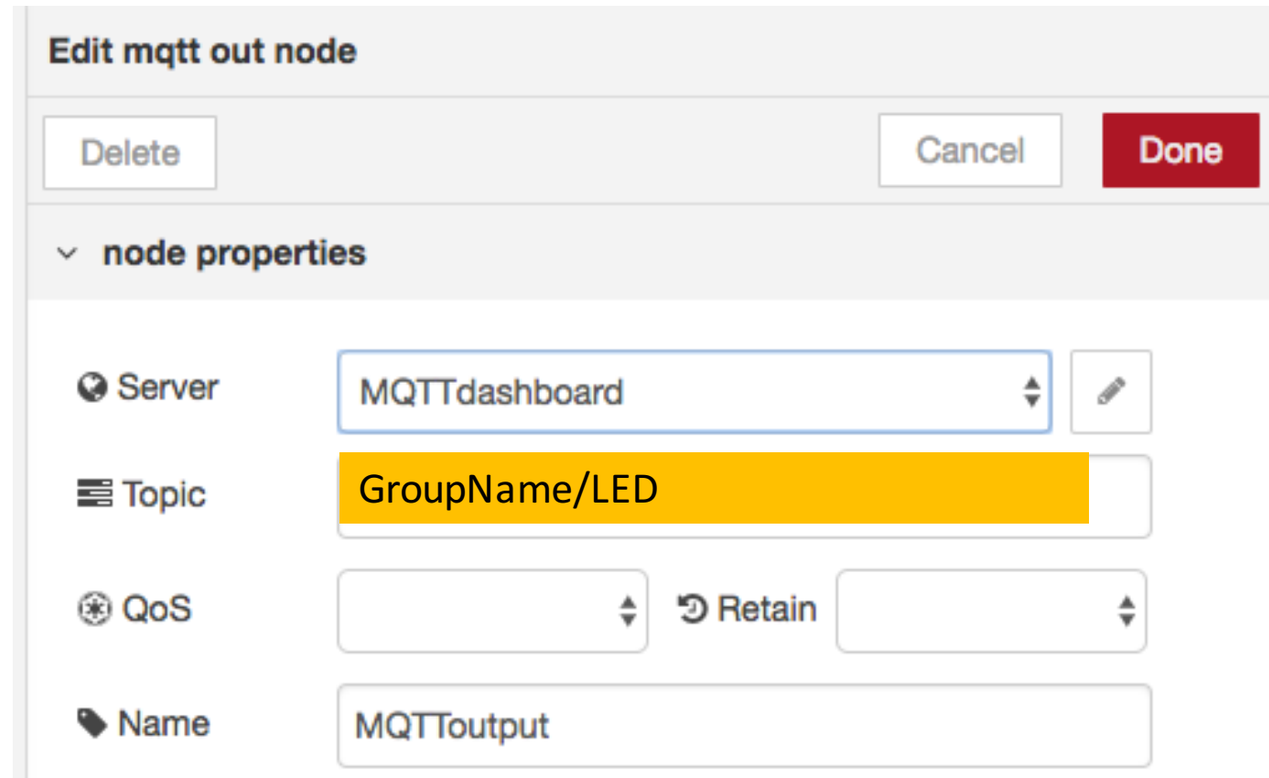
# Edit the Switch Node

- Group:
  - Please feel free to use the existing or create a new UI group
- Label: switchLED
- When clicked, send:
  - On Payload (String) on
  - Off Payload (String) off
  - Topic: **unique topic name (to publish for LED control)**
- Hit "Done"



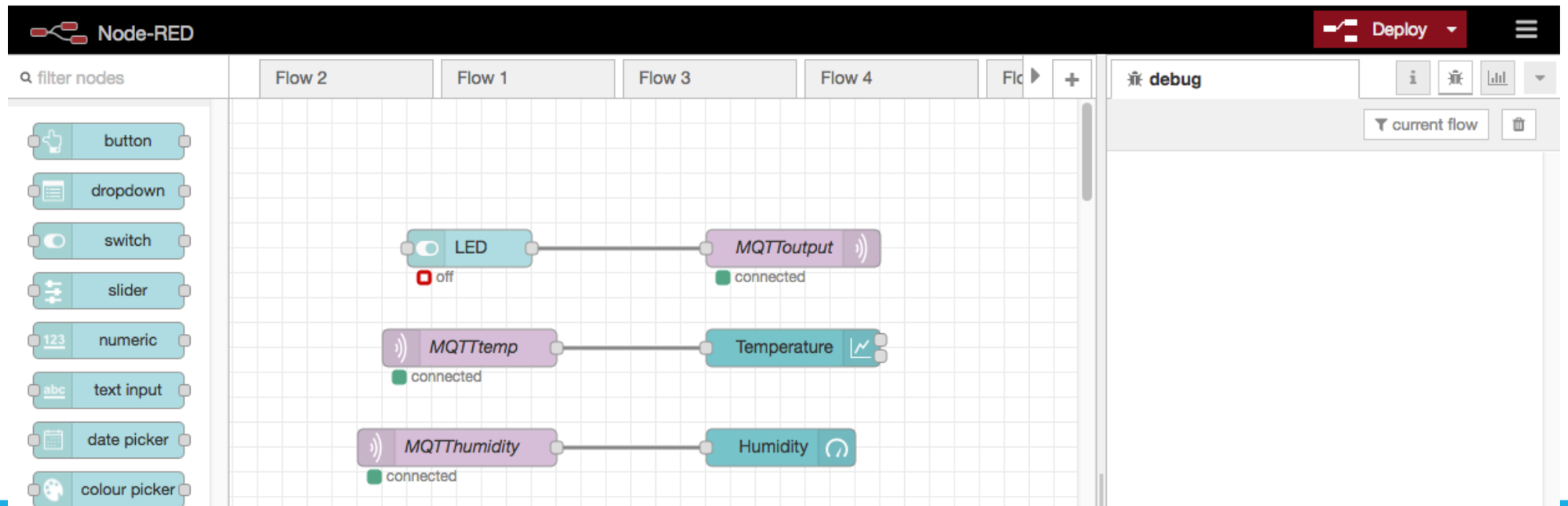
# Configure the mqtt Output Node

- Configure the **mqtt** output node to act on the message received from the switch
- Under Server,
  - use same server
- Topic: **unique topic name (to publish for LED control)**
- Name: MQTToutput
- Click “Done”



# Deploy the Flow

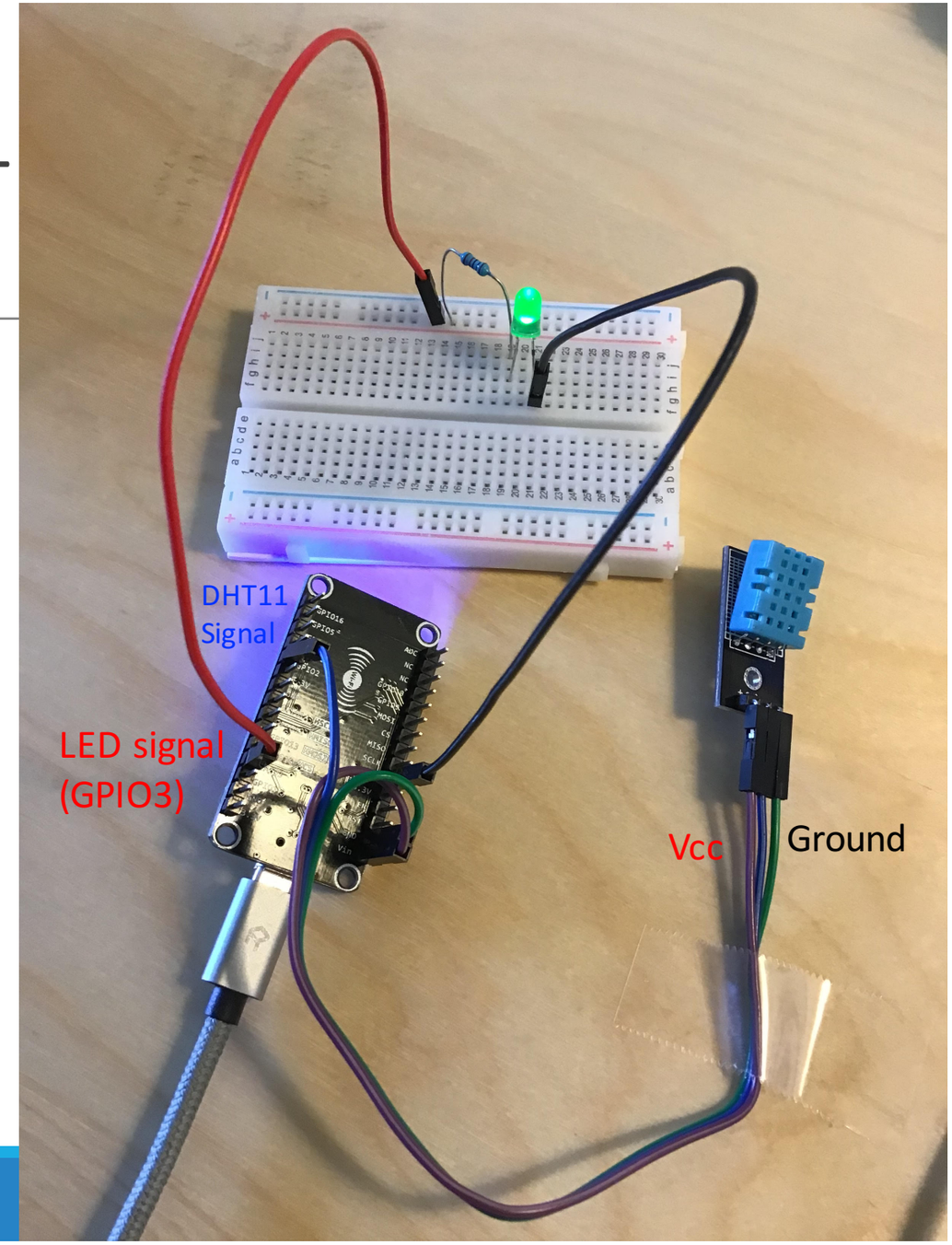
- Your nodeRED flow is now ready for deployment.
- We can deploy it now, but there will be no data coming in.



# Connect LED and DHT11 to NodeMCU

DHT11 signal (GPIO2)

LED signal (GPIO3)



# Edit & Upload the Code

- Replace ssid with your WiFi ssid
- Replace password with your WiFi password
- mqtt_server, please use "broker.mqttdashboard.com”

- Replace topic (in the callback function) with your LED topic
  - if(topic== “…….”)

GroupName/LED

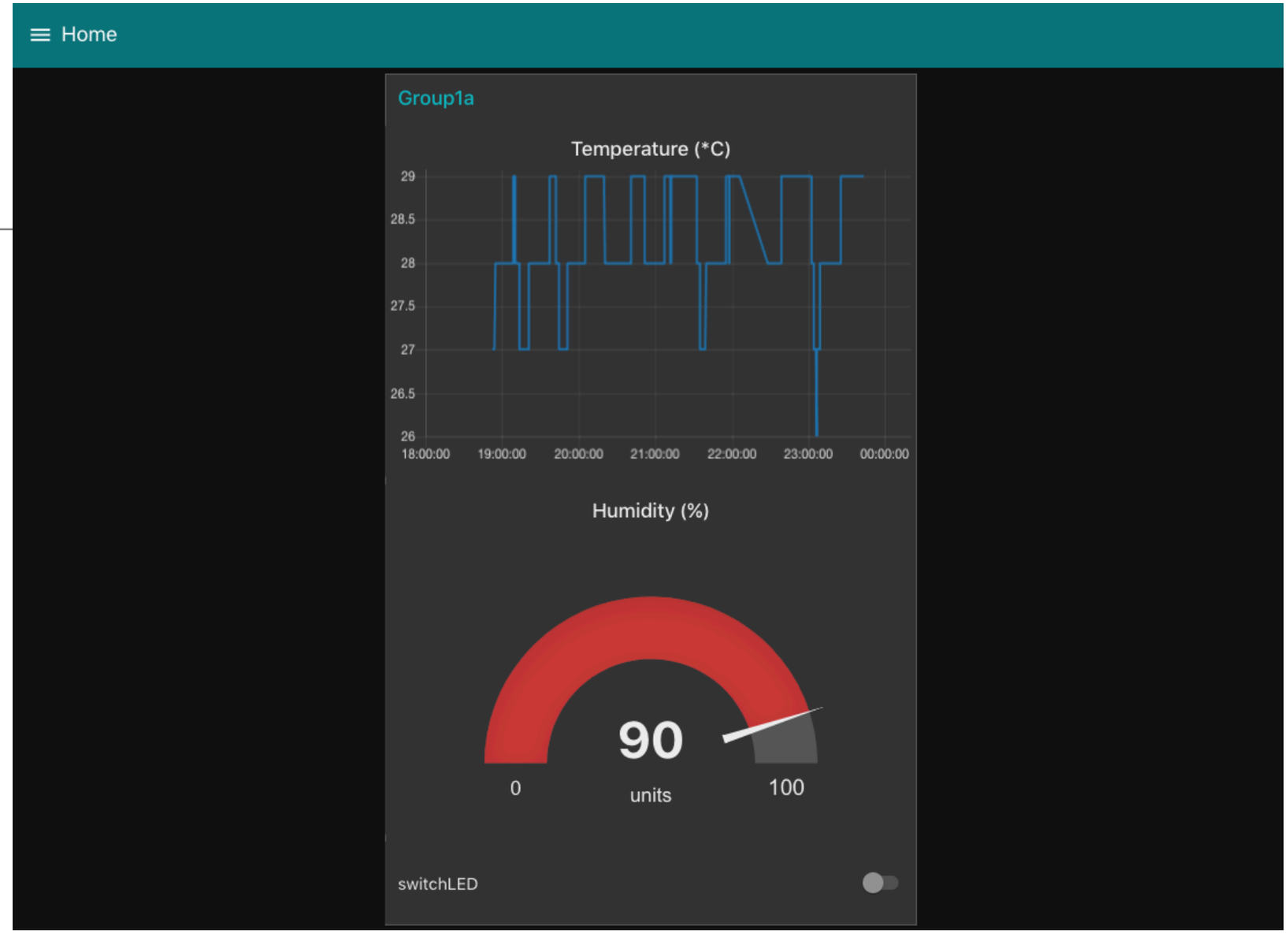
- Replace topic (in the reconnect function) with your LED topic
  - Client.subscribe(“…….”)

GroupName/LED

- Replace the two topics (in the loop) with your temperature and humidity topics
  - Client.publish(”……”, temperatureTemp)
  - Client.publish(“……”, humidityTemp)

GroupName/temperature
GroupName/humidity

GroupName:
- FlyingBird2
- InkzandFriends
- LockerSwarm
- MonkolTeam
- Parking
- vCanteen

# Done!

- You will now see your live plots
- Please switch the button ON/OFF and see if you can control the LED!