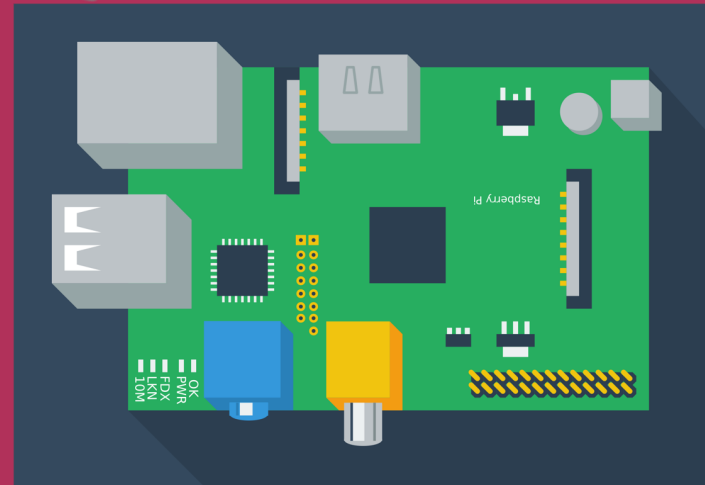
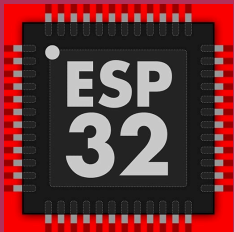
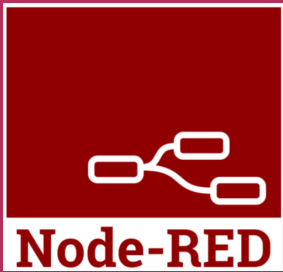


MQTT on Pi

NOV 2021



Safyzan Salim
scriptworkz ent

Scenario

You have been assigned to monitor the temperature and moisture of a server room. The results can be monitored anywhere. Your system must be able to capture 2 server racks' temperature and humidity, and also the server room's temperature and humidity too.

Future Project >> *A notification alarm will be activated if the any of the sensors passes the certain condition, e.g., rack temperature greater than 25°C, or server room's temp too high... etc.*

Prerequisites

Hardware:

- Raspberry Pi board with:
 - Raspberry Pi OS
 - 16 GB class 10 microSD
 - 5V 2.5A power supply
- ESP32 with Temperature & Humidity Sensor, DHT11**

Software:

- Node-RED
- Arduino IDE
- Arduino Client for MQTT library
- Aedes MQTT

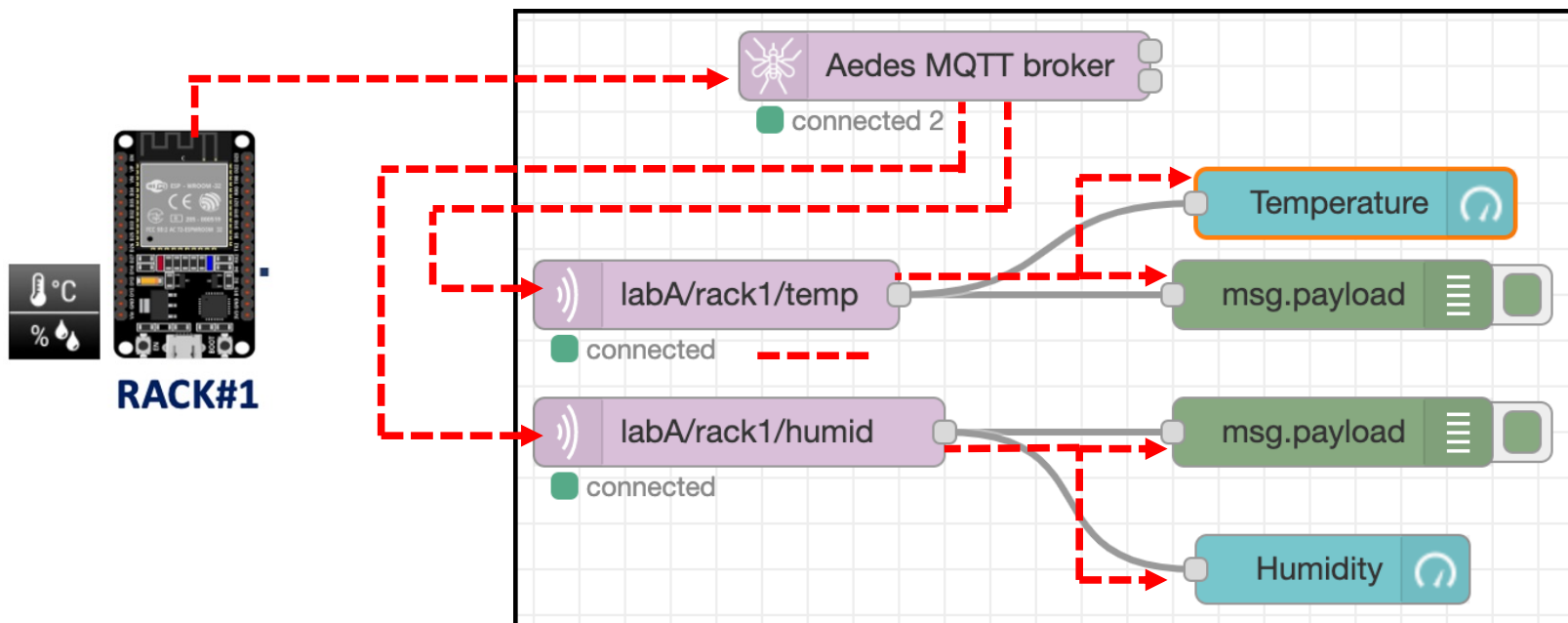
WiFi:

- A 2.4G AP

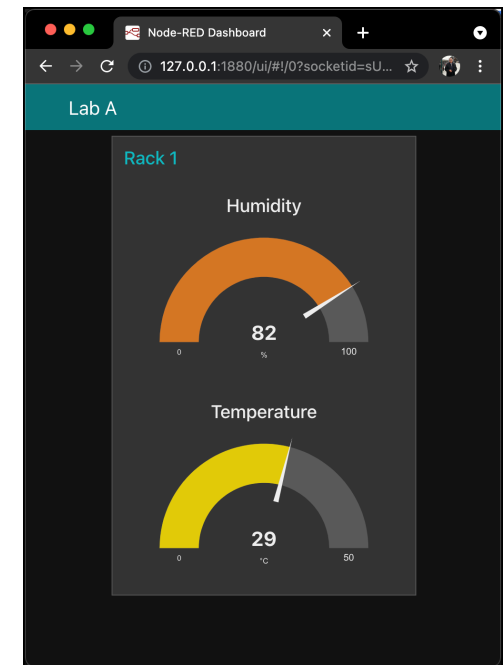
**** This project uses Grove - Temperature & Humidity Sensor (DHT11)**

How It Works

- ESP32 & DHT11 will be fixed at a rack, named Rack#1 at Lab A. Aedes broker is running on a Raspberry Pi (PC/Laptop/Mac).
- After connected to WiFi, data from ESP32 will be sent to broker.
- Since the Node-RED has subscribed topic **labA/rack1/temp** & **labA/rack1/humid** from ESP32, both information (temp & humid) will be displayed at debug & also gauges – in real time.



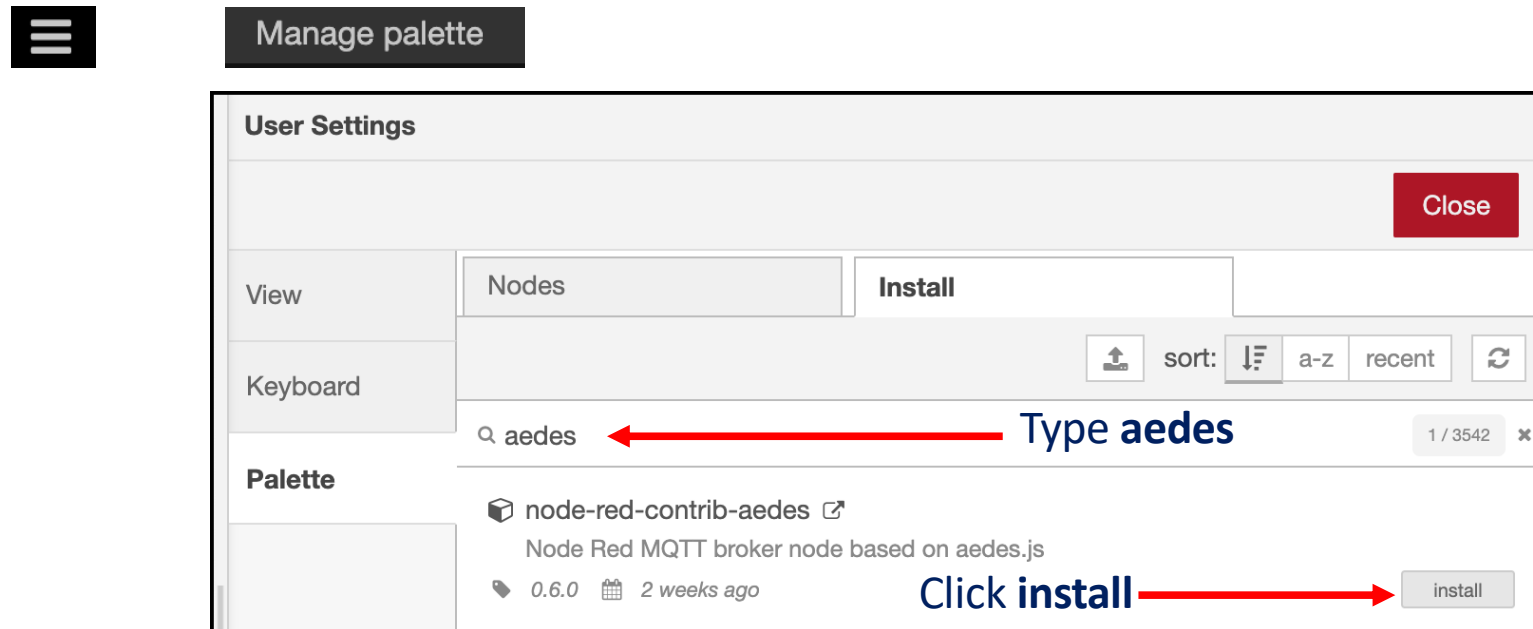
Raspberry Pi / Laptop / Desktop – Cross Platform



i. Install & Run Aedes Broker Service

- Execute Node-RED service.
- Install **Aedes** broker by clicking:

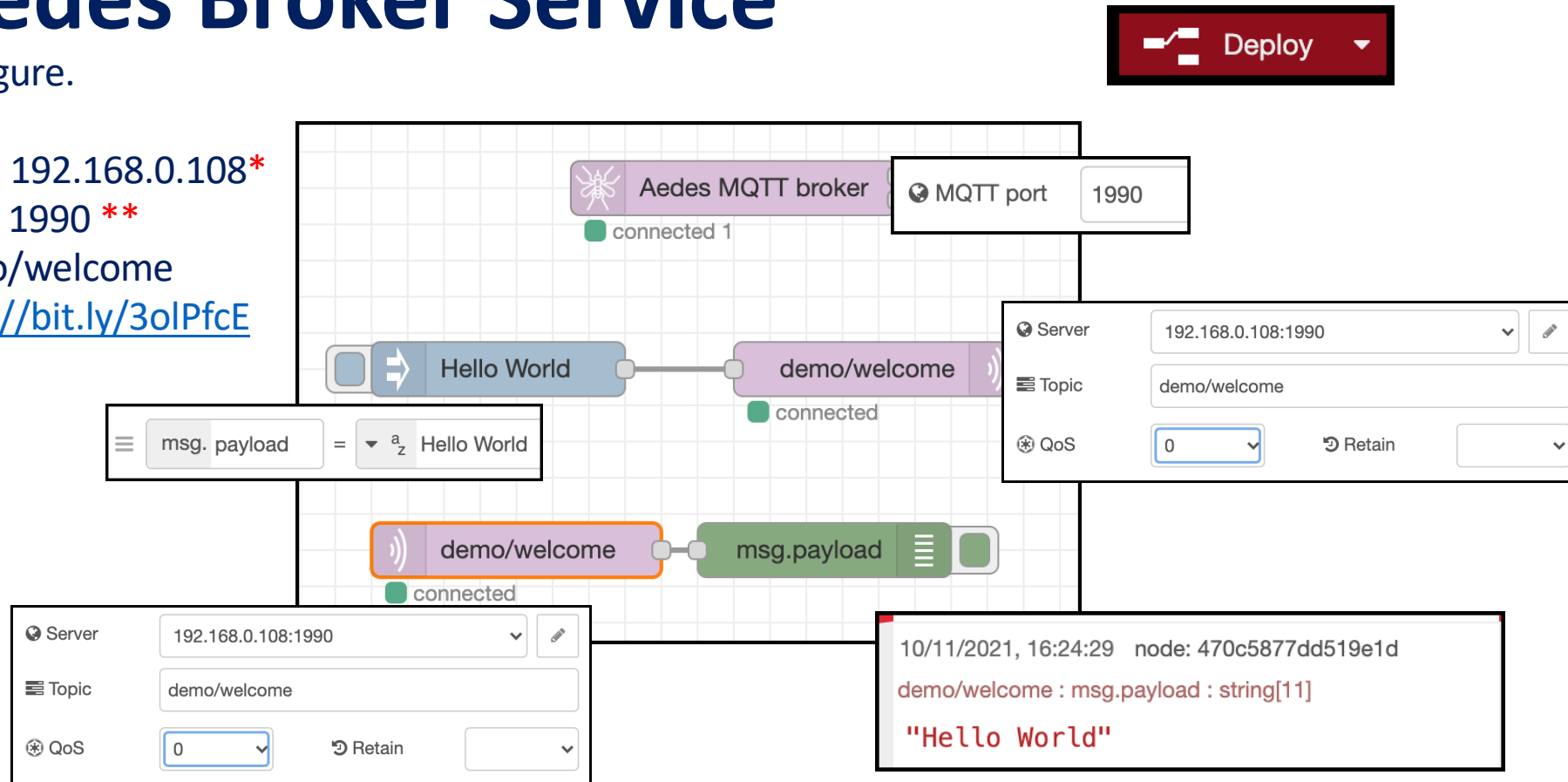
Node-RED Menu > Manage palette > Install > type **aedes** at search module > **node-red-contrib-aedes** > install



- Wait & make sure the install button changed to installed.

ii. Test Aedes Broker Service

- Setup as in the figure.
- Configurations:
 - BROKER IP = 192.168.0.108*
 - MQTT port = 1990 **
 - Topic = demo/welcome
- Download: <https://bit.ly/3oIPfcE>

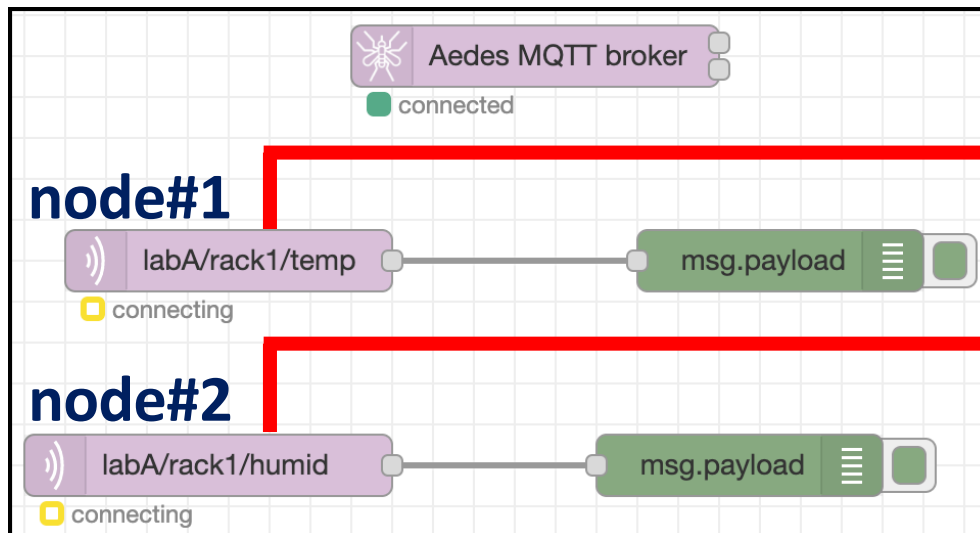


* Please check your device IP address.

** Do not use 1880 – reserved to localhost:1880

iii. Deploying Aedes Broker

- Configurations:
 - Broker IP = 192.168.0.108*
 - MQTT port = 1889**
 - MQTT_IN = labA/rack1/temp ← node#1
 - MQTT_IN = labA/rack1/humid ← node#2
- Download palette: <https://bit.ly/3qGwyTE>.



Properties

Server: 192.168.0.108:1889

Topic: labA/rack1/temp

QoS: 0

Output: auto-detect (string or buffer)

Name: Name

Properties

Server: 192.168.0.108:1889

Topic: labA/rack1/humid

QoS: 0

Output: auto-detect (string or buffer)

Name: Name

* Please check your device IP address.

** do not use 1880 – reserved to localhost:1880

SZS - Nov 2021

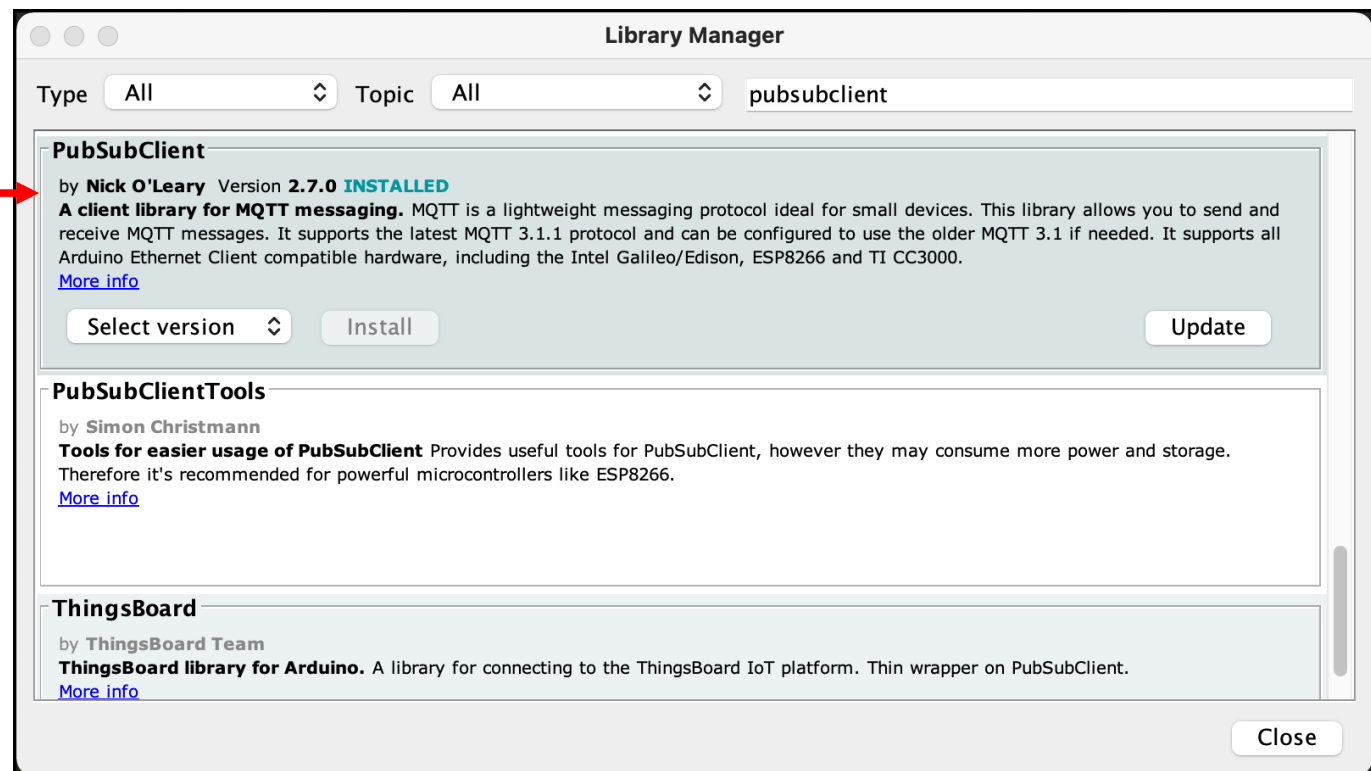


iv. Arduino Client MQTT

- Install New Library by opening **Library Manager** and search for **PubSubClient** by **Nick O'Leary** – latest version.

Sketch > Include Library > Manage Libraries

PubSubClient
Nick O'Leary



v. Arduino PubSub Sketch

- Download the file working file from <https://bit.ly/30dwDnb>.

sketch_nov11a-nov21-aedes-esp32-groove-dht11

```
15
16 // #include <ESP8266WiFi.h> //if you are using nodemcu
17 #include <WiFi.h> //if you are using esp32
18 #include <PubSubClient.h>
19
20 //////////GROVE DHT11//////////
21 #include "DHT.h"
22 #define DHTTYPE DHT11
23
24 #define DHTPIN 26 // connect signal to pin 26
25 DHT dht(DHTPIN, DHTTYPE);
26 ////////////////////////////
```

#16 Remove // and put // at
#17 Line#15 if you are using
NodeMCU.

#24 Place a jumper at signal pin to
pin 26 of ESP32.

v. Arduino PubSub Sketch

- Download the file working file from <https://bit.ly/30dwDnb>.

```
30
31 // Update these with values suitable for your network.
32 const char* ssid = "YOUR SSID";
33 const char* password = "YOUR SSID PASSWORD";
34 const char* mqtt_server = "NODE-RED IP ADDRESS"; /// NodeRED IP Address
35 int mqtt_port = 1889; //Must match with broker's MQTT port - dont use 1880
36
```

#32
#33

Set your SSID & password.

#34

IP address of your Node-RED device.

v. Arduino PubSub Sketch

- Download the file working file from <https://bit.ly/30dwDnb>.

```

44 void setup() {
45   Serial.begin(115200);
46   // Start up the library
47   client.setServer(mqtt_server, mqtt_port);
48   client.setCallback(callback);
49
50   setup_wifi(); //function call
51
52   Serial.println("WiFi Connected...");
53   Serial.print("MQTT Server IP Address:");
54   Serial.print(mqtt_server);
55   Serial.print(":");
56   Serial.println(String(mqtt_port));
57   Serial.print("ESP32 IP Address: ");
58   Serial.println(WiFi.localIP());
59   Serial.println("Modbus RTU Master Online");
60
61   pinMode(LED_BUILTIN, OUTPUT);
62 }

```

From Serial Monitor

```

#50 Connecting to raspberrypi 2.4G
#52 .WiFi Connected...
#53 MQTT Server IP Address:192.168.0.108:1884
#57 ESP32 IP Address: 192.168.0.110
#59 Modbus RTU Master Online
77.00 28.90 Reading from DHT11 sensor
Attempting MQTT connection...Broker connected to ESP32

```

v. Arduino PubSub Sketch

- Download the file working file from <https://bit.ly/30dwDnb>.

```
63
64 void setup_wifi() {
65     delay(10);
66     // Start by connecting to a WiFi network
67     Serial.println();
68     Serial.print("Connecting to ");
69     Serial.println(ssid);
70
71     WiFi.begin(ssid, password);
72
73     while (WiFi.status() != WL_CONNECTED) {
74         delay(500);
75         Serial.print(".");
76     }
77 }
78
```

A function to connect the ESP32 / NodeMCU to existing access point (WiFi).

Check your SSID & password in case non-stop dotted line appears.

v. Arduino PubSub Sketch

- Download the file working file from <https://bit.ly/30dwDnb>.

```
103 void reconnect() {
104
105 // Loop until we're reconnected
106 while (!client.connected()) {
107     Serial.print("Attempting MQTT connection...");
108 // Attempt to connect
109     if (client.connect("ESP32Client")) {
110         Serial.println("Broker connected to ESP32");
111         client.subscribe("event"); // Topic at ESP32
112     } else {
113         Serial.print("failed, rc=");
114         Serial.print(client.state());
115         Serial.println(" try again in 5 seconds");
116         // Wait 5 seconds before retrying
117         delay(5000);
118     }
119 }
120 }
```

ESP32 @ NodeMCU ensure persistent connection between MCU and the broker.

v. Arduino PubSub Sketch

- Download the file working file from <https://bit.ly/30dwDnb>.

```

123 ///////////////////////////////////////////////////
124 float temp_hum_val[2] = {0}; // grove_dht set nilai 2 array = 0
125 dht.readTempAndHumidity(temp_hum_val); // baca temp & humid
126 ///////////////////////////////////////////////////
127 Serial.print(temp_hum_val[0]);
128 Serial.print(" ");
129 Serial.println(temp_hum_val[1]);
130 char tempValue[15], humidValue[15];
131 dtostrf(temp_hum_val[0],4, 0, humidValue); // convert float to char
132 dtostrf(temp_hum_val[1],4, 0, tempValue); // convert float to char
133 client.publish("labA/rack1/humid", humidValue); /// send char
134 client.publish("labA/rack1/temp", tempValue); /// send char
135
136 if (!client.connected()) {
137     reconnect(); //function call
138 }
```

- L#124** Set 2 array elements; temp_hum_val[0] & temp_hum_val[1]. Set both value = 0.
- L#125** Start to read temp & humid from sensor.
- L#127** Print both values; temp & humid at Serial Monitor.
- L#127** Convert temp & humid from float to char.
- L#133** Publish the temp & humid value accordingly. Topic name must match with Node-RED's topic.
- L#136** Will reconnect to broker if disconnected.

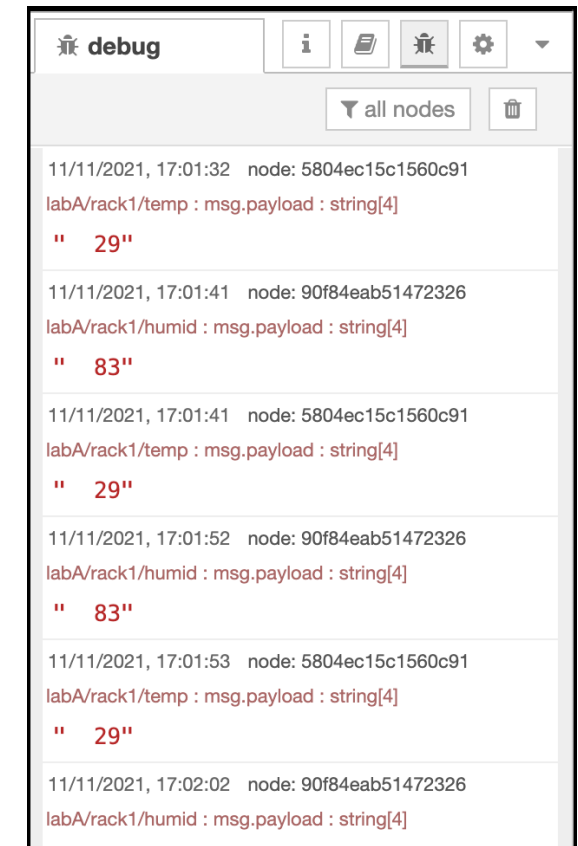
v. Arduino PubSub Sketch – Expected Output

▪ Arduino's Serial Monitor



```
Connecting to raspberrypi 2.4G
.WiFi Connected...
MQTT Server IP Address:192.168.0.108:1884
ESP32 IP Address: 192.168.0.110
Modbus RTU Master Online
77.00 28.90
Attempting MQTT connection...Broker connected to ESP32
77.00 28.90
78.00 29.00
76.00 28.90
77.00 29.10
77.00 29.00
Attempting MQTT connection...Broker connected to ESP32
76.00 28.90
76.00 28.90
77.00 29.00
76.00 28.90
76.00 28.90
```

▪ Node-RED Debug Window



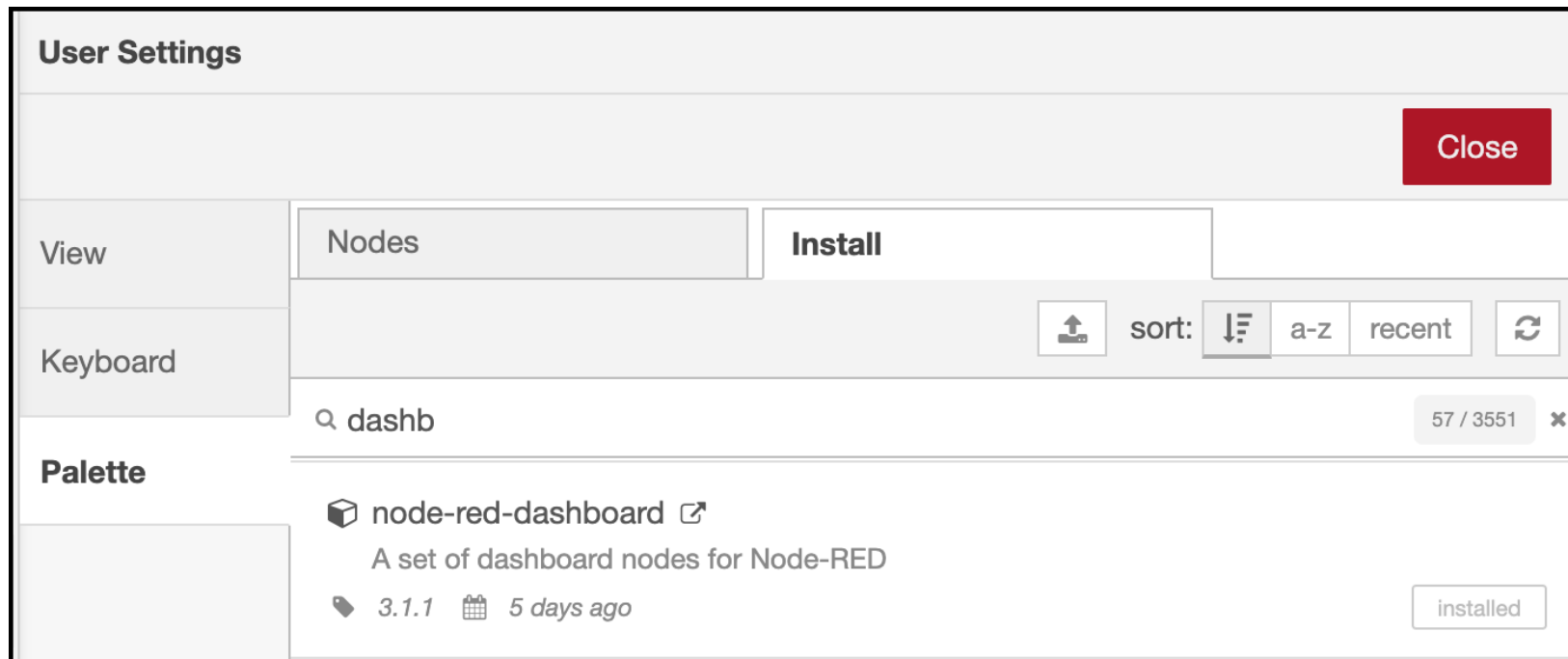
Time	Node	Topic	Payload
11/11/2021, 17:01:32	node: 5804ec15c1560c91	labA/rack1/temp	29
11/11/2021, 17:01:41	node: 90f84eab51472326	labA/rack1/humid	83
11/11/2021, 17:01:41	node: 5804ec15c1560c91	labA/rack1/temp	29
11/11/2021, 17:01:52	node: 90f84eab51472326	labA/rack1/humid	83
11/11/2021, 17:01:53	node: 5804ec15c1560c91	labA/rack1/temp	29
11/11/2021, 17:02:02	node: 90f84eab51472326	labA/rack1/humid	83

* Do check your IP address, Topic at your sketch if you did not get as expected output.

vi. Create Visual Dashboard

- Go to Node-RED **editor**.
- Install **Dashboard** broker by clicking:

Node-RED Menu > Manage palette > Install > type dashboard at search module > node-red-dashboard > install



vi. Create Visual Dashboard

- In order to run a Node-RED **dashboard**, you need to create a **tab** & a **group**.
- Decide what information that you want to display.
- In our case, you should consider:
 - ✓ Classroom : **Lab A**
 - ✓ Rack ID: **1**
 - ✓ Item to visual/display: **Temperature & Humidity**
- Thus,
 - **Tab = Main Title = Lab A**
 - **Group = Sub Title = Rack1**
- FYI, more than 1 group can be hosted by 1 tab.
- Your tab can also host group from other tab.



vi. Create Visual Dashboard

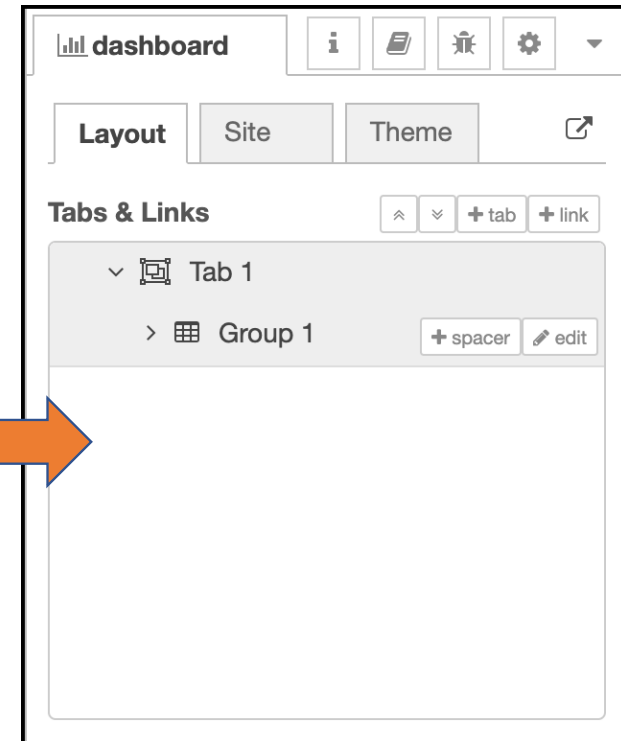
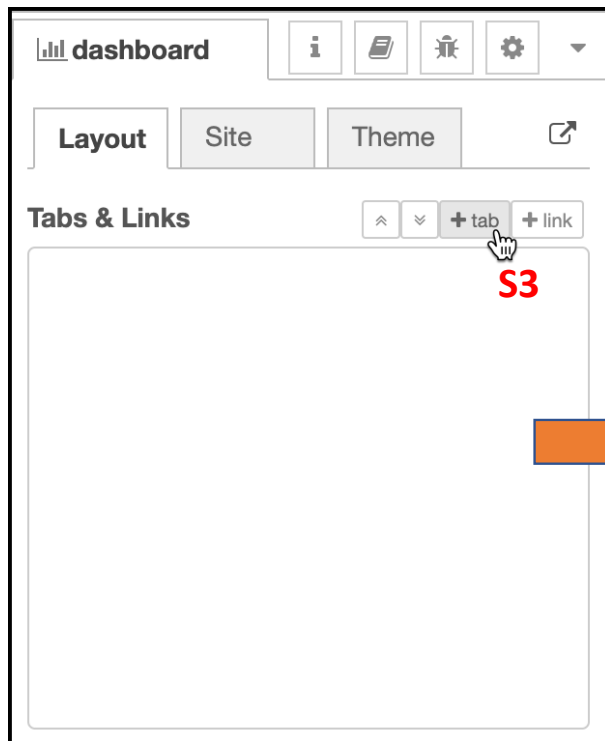
- We need to setup dashboard properties before connecting the gauge/meter node to the flow.
- **Step 1:** Click the dropdown **Menu** of Node-RED.
- **Step 2:** Click **Dashboard** link.

The screenshot shows the Node-RED web interface. On the left, the 'common' node palette is visible. The central workspace contains a flow with an 'Aedes MQTT broker' node connected to two 'labA/rack1/temp' and 'labA/rack1/humid' nodes, each followed by a 'msg.payload' node. On the right, the 'debug' console is open, displaying a list of messages. A dropdown menu is open over the 'debug' console, showing options: 'Information', 'Help', 'Debug messages', 'Configuration nodes', 'Context Data', and 'Dashboard'. The 'Dashboard' option is highlighted with a red 'S2' label. A red 'S1' label points to the 'debug' console header.

vi. Create Visual Dashboard

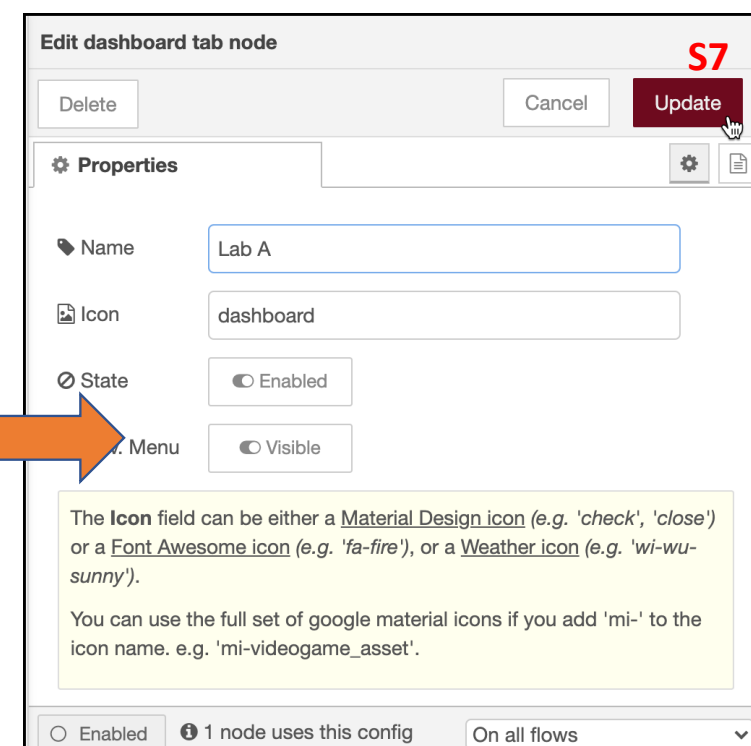
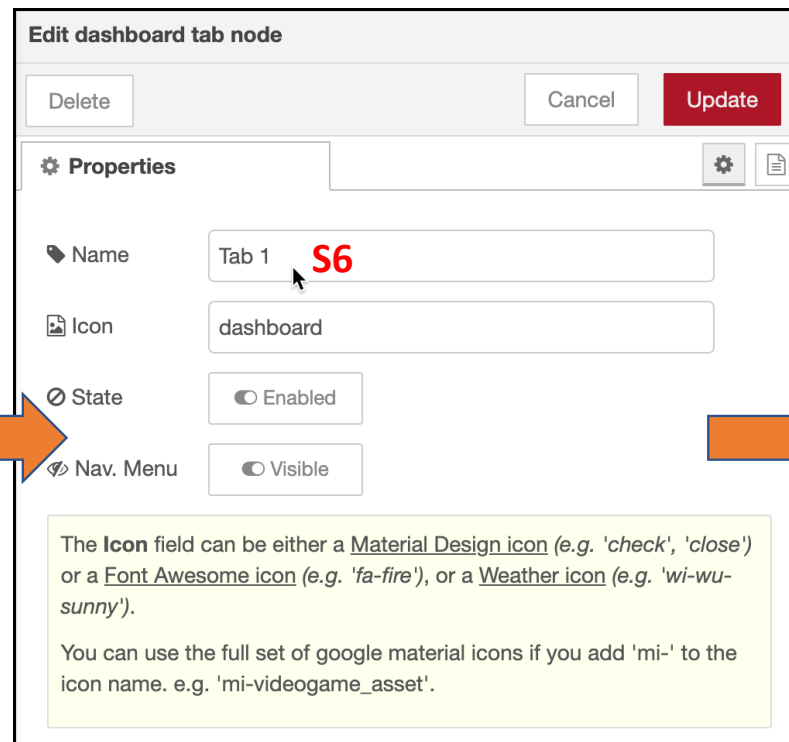
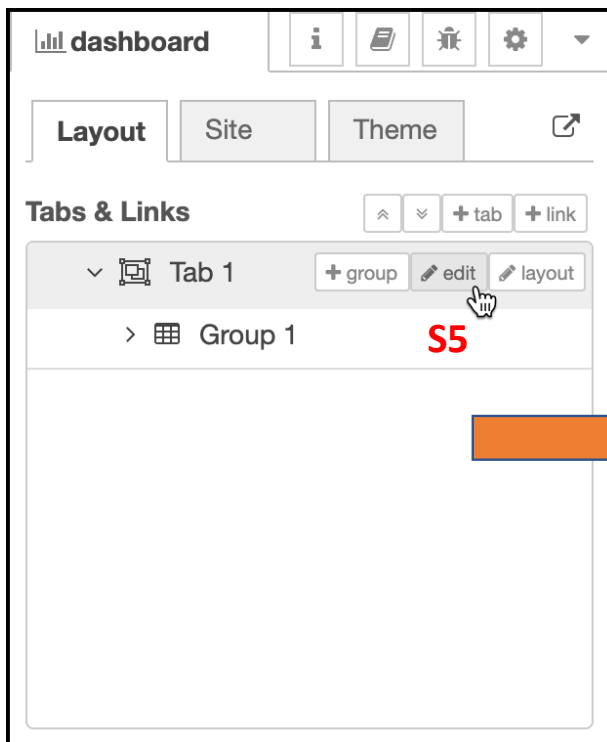
+ tab

- **Step 3:** Create a project by clicking **+ tab** .
- **Step 4:** Add element in your project by clicking at **+ group** .
- You may edit/update/delete **tab** properties or **group** properties by clicking at **edit** .



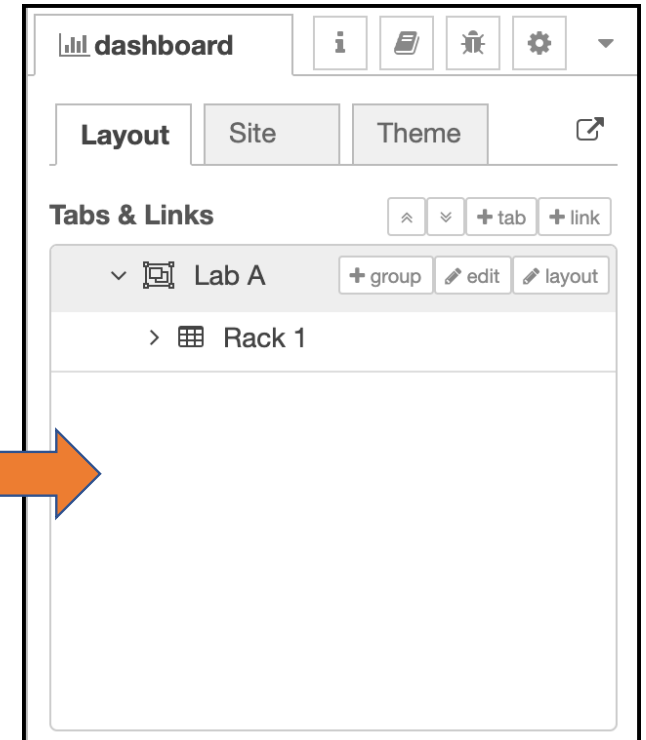
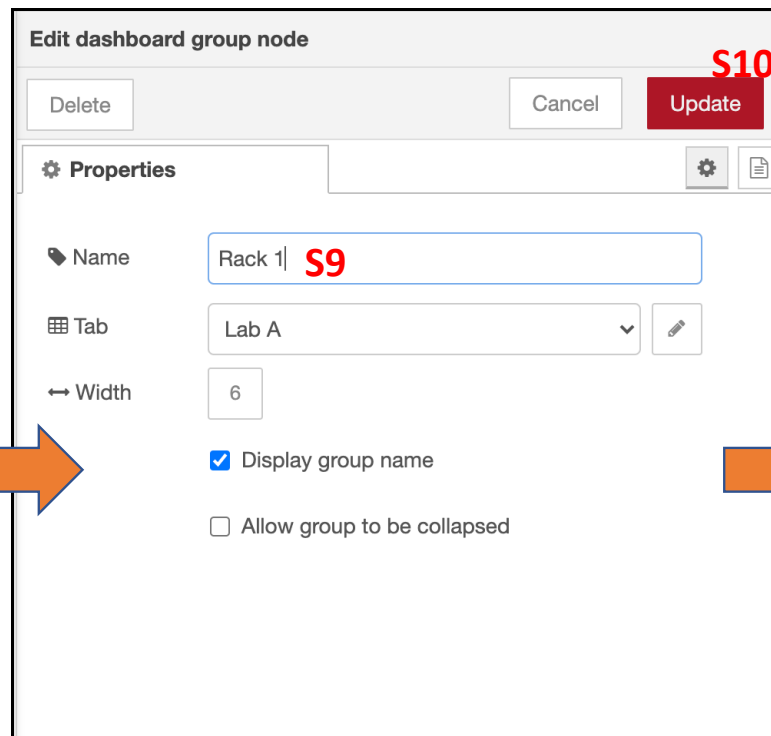
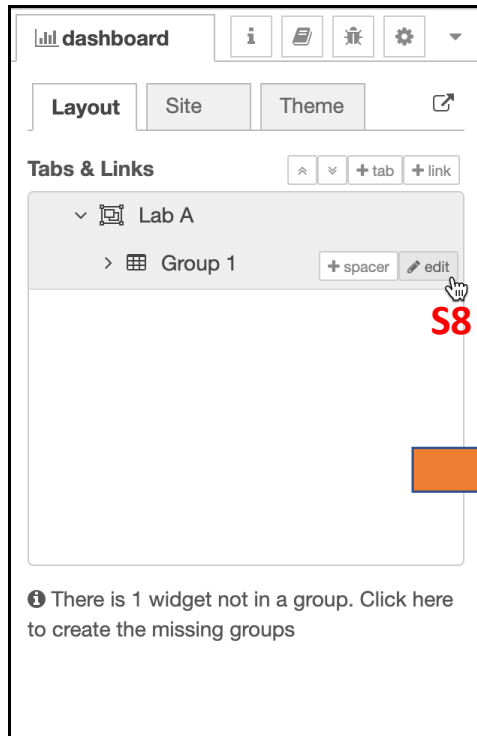
vi. Create Visual Dashboard

- **Step 5:** Click  to edit the **Tab 1** name .
- **Step 6:** Change **Tab 1** with **Lab A**.
- **Step 7:** Next, click  to save the setting.



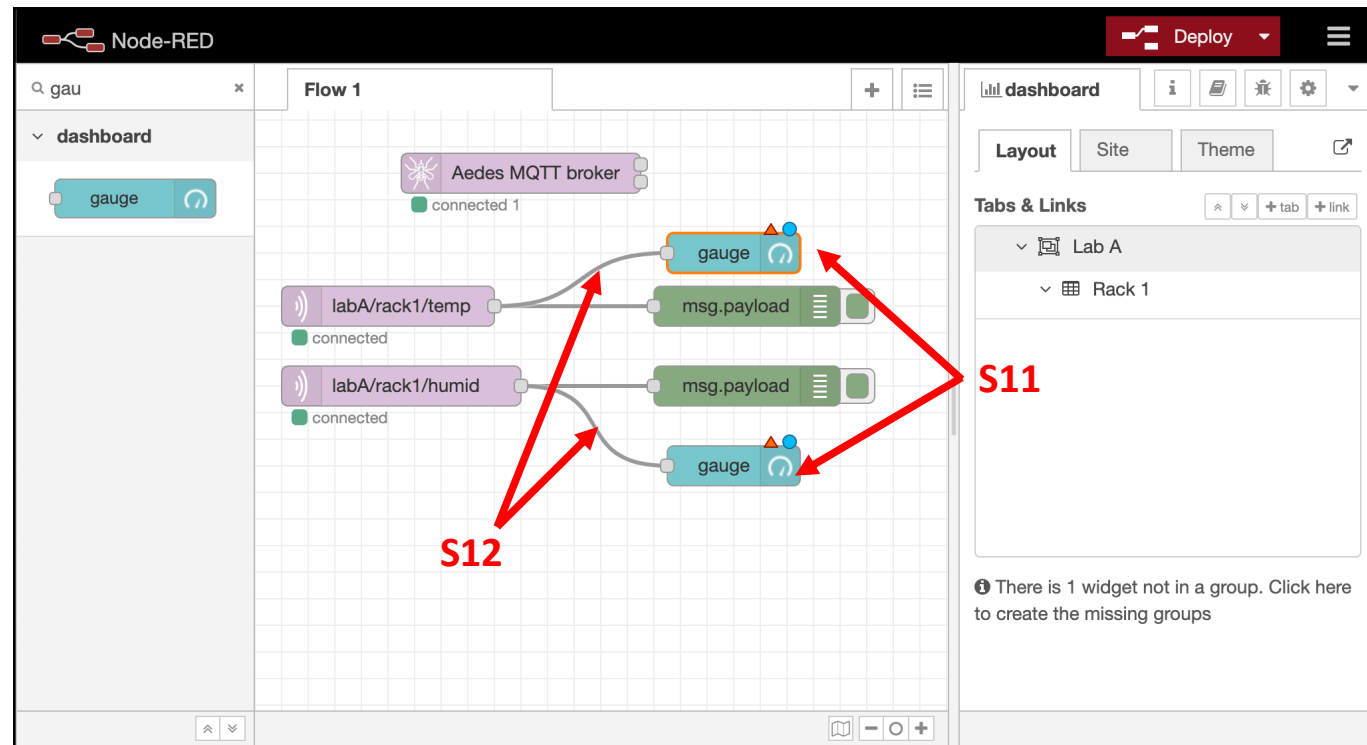
vi. Create Visual Dashboard

- **Step 8:** Click **edit** tab at **Group 1**.
- **Step 9:** Change **Group 1** to **Rack 1**.
- **Step 10:** Click **Update** tab to save the setting..



vi. Create Visual Dashboard

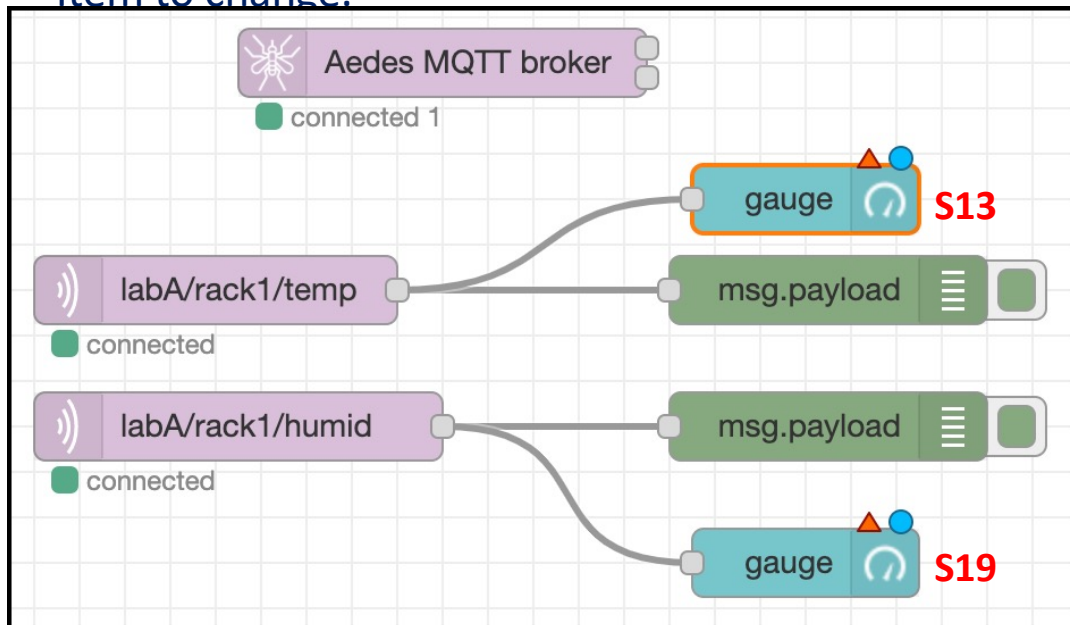
- **Step 11:** Now, add 2 gauges from dashboard palette.
- **Step 12:** Wire each gauge to **labA/rack1/temp** & **labA/rack1/humid** respectively.
- Any of dashboard palette must be configured into its specific tab & group.
- The chances of your node appears in different tab will happen when you mistakenly mapped the groups & tabs.



vi. Create Visual Dashboard

Configuration of Gauges.

- **Step 13:** Double click gauge connected to **labA/rack1/temp**.
- **Step 19:** Double click gauge connected to **labA/rack1/humid**.
- ******Item to change.



Edit gauge node

Delete Cancel Done

⚙ Properties

****** Group [unassigned] Group 1

Size auto




Type Gauge

****** Label gauge

Value format {{value}}

****** Units units

****** Range min 0 max 10

Colour gradient   

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

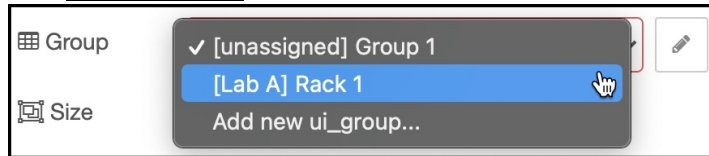
Name

Enabled

vi. Create Visual Dashboard

Update Temperature Gauge Properties.

- Step 14:** Click textbox at **Group**. Change **[unassigned] Group 1** to select **[Lab A] Rack 1**.



- Step 15:** Change text **gauge** to **Temperature** at .
- Step 16:** Replace **units** with **° C** (you may copy ° here).
- Step 17:** Replace maximum range **10** with **50**
- DHT11's range is between **0° C to 50° C**.
- Step 18:** Click **Done** upon completion. ******Item to change.

Edit gauge node

Delete Cancel **S18** Done

Properties

Group [Lab A] Rack 1 **S14**

Size auto




Type Gauge

Label Temperature **S15**

Value format {{value}}

Units ° C **S16**

Range min 0 max 50 **S17**

Colour gradient   

Sectors 0 ... optional ... optional ... 50

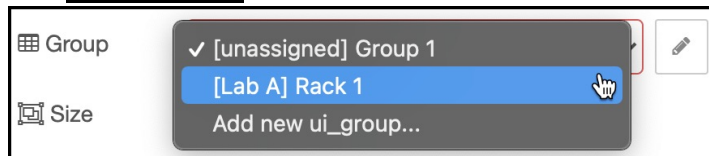
Name

☐ Enabled

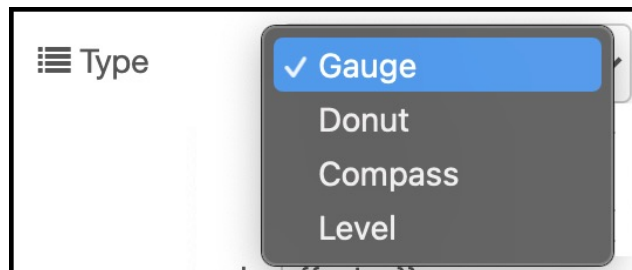
vi. Create Visual Dashboard

Update Humidity Gauge Properties.

- Step 20:** Click textbox at **Group**. Change **[unassigned] Group 1** to select **[Lab A] Rack 1**.



- Step 21:** Change text **gauge** to **Temperature** at .
- Step 22:** Replace **units** with **° C** (you may copy ° here).
- Step 23:** Replace maximum range **10** with **50**
- Humidity range is between **0° C to 100° C**.
- Step 24:** Click **Done** upon completion.
- Step 25:** Node-RED offers a few interactive visual for metering. You may go to **Type** & select the output that suit with your needs. Do not forget to click **Deploy** button.



Delete
Cancel
Done

Properties

Group
[Lab A] Rack 1

Size
auto

Type
Gauge

Label
Humidity

Value format
{{value}}

Units
%

Range
min 0 max 100

Colour gradient

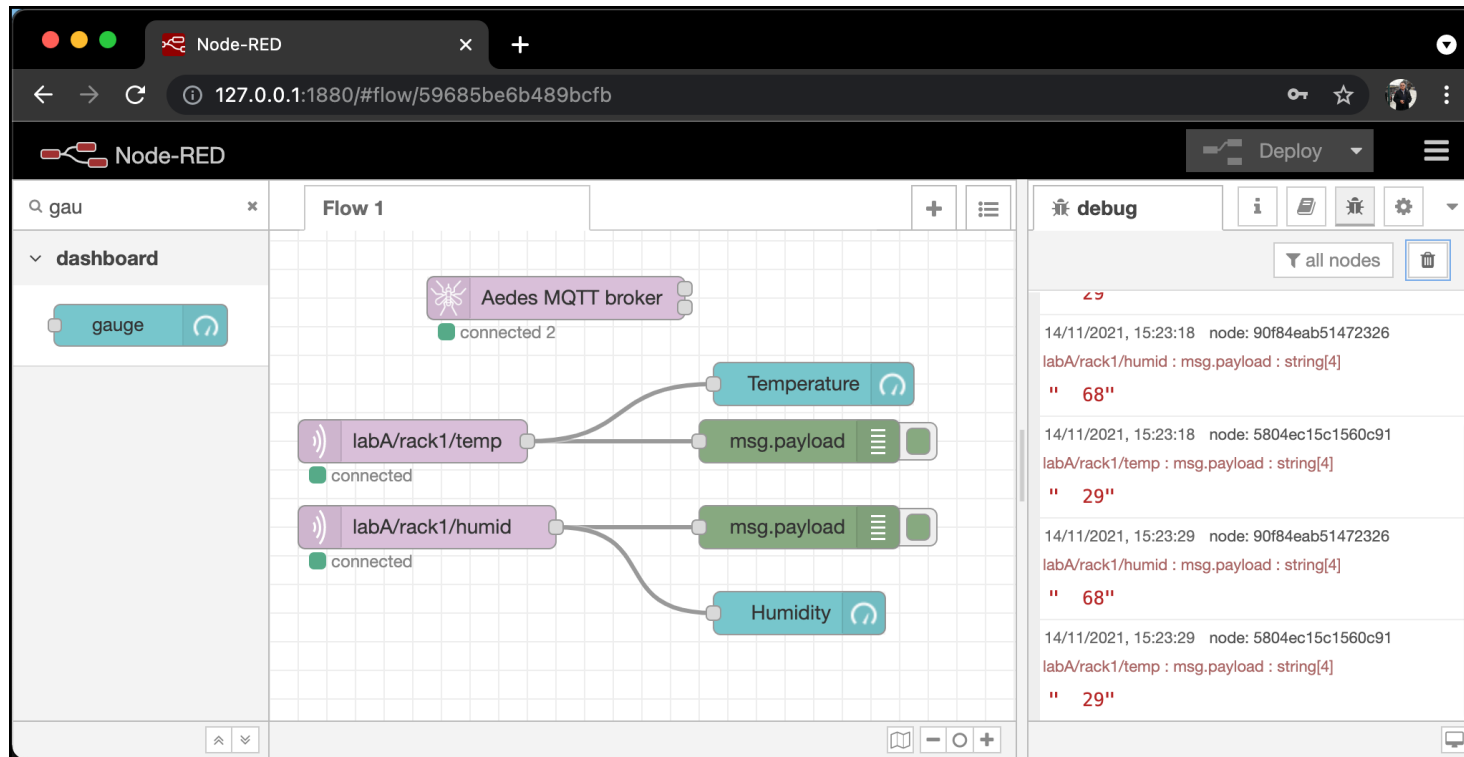
Sectors
0 ... optional ... optional ... 100

Name

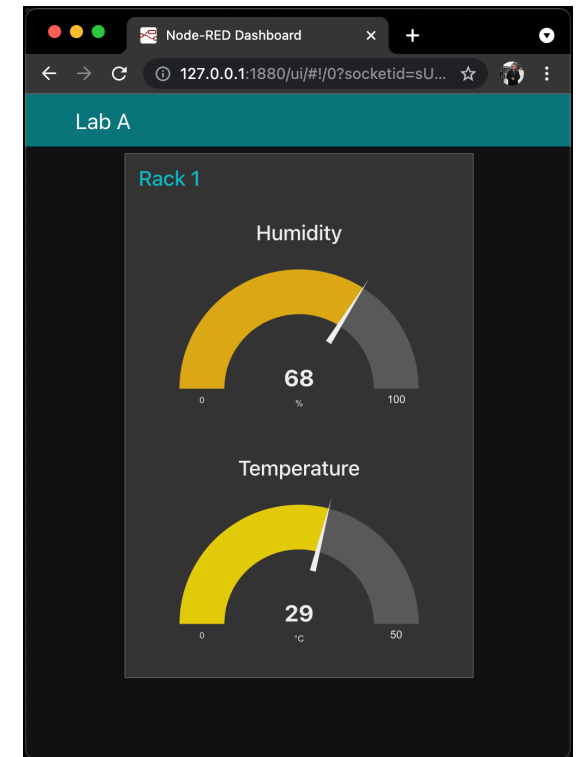
Enabled

vi. Create Visual Dashboard

- Download the file working file from <https://bit.ly/3DfNfsl>.



Node-RED Flow Editor



Node-RED Dashboard

vi. Create Visual Dashboard

Adding More Racks.

- In order to add more rack, means you have to add more sensors.
- You have to consider the topic at ESP32 and also at broker side.
- Assume you want to add Rack 2, Lab A. Open arduino sketch (<https://bit.ly/31WQUhn>) and change:
labA/rack1/humid at line number **133** to **labA/rack2/humid**, and also ,
labA/rack1/temp at line number **134** to **labA/rack2/temp**.

```
133 client.publish("labA/rack1/humid", humidValue); /// send char  
134 client.publish("labA/rack1/temp", tempValue); /// send char
```

- Next, at Node-RED flow editor, add 2 **mqtt in**, set IP address similar with steps, label the topic as **labA/rack2/humid** and **labA/rack2/temp**.

QUESTIONS

END