

## BDA Case Study 2: Storing Data in MySQL



v1 mar2021

iezan74@gmail.com



**Scenario:** The result from the sensor need to be stored into database and at same time, need to display at dashboard too.

## **Solution & Requirement:**

- i. Continuation of BDA Case Study 1 (Smart Agricultural System).
- ii. Latest version XAMPP software .

## **Methods:**

NodeRED + MySQL deployment:

- i. Install XAMPP.
- ii. Manually inject data & inject live data.
- iii. Combination of IoT & BDA

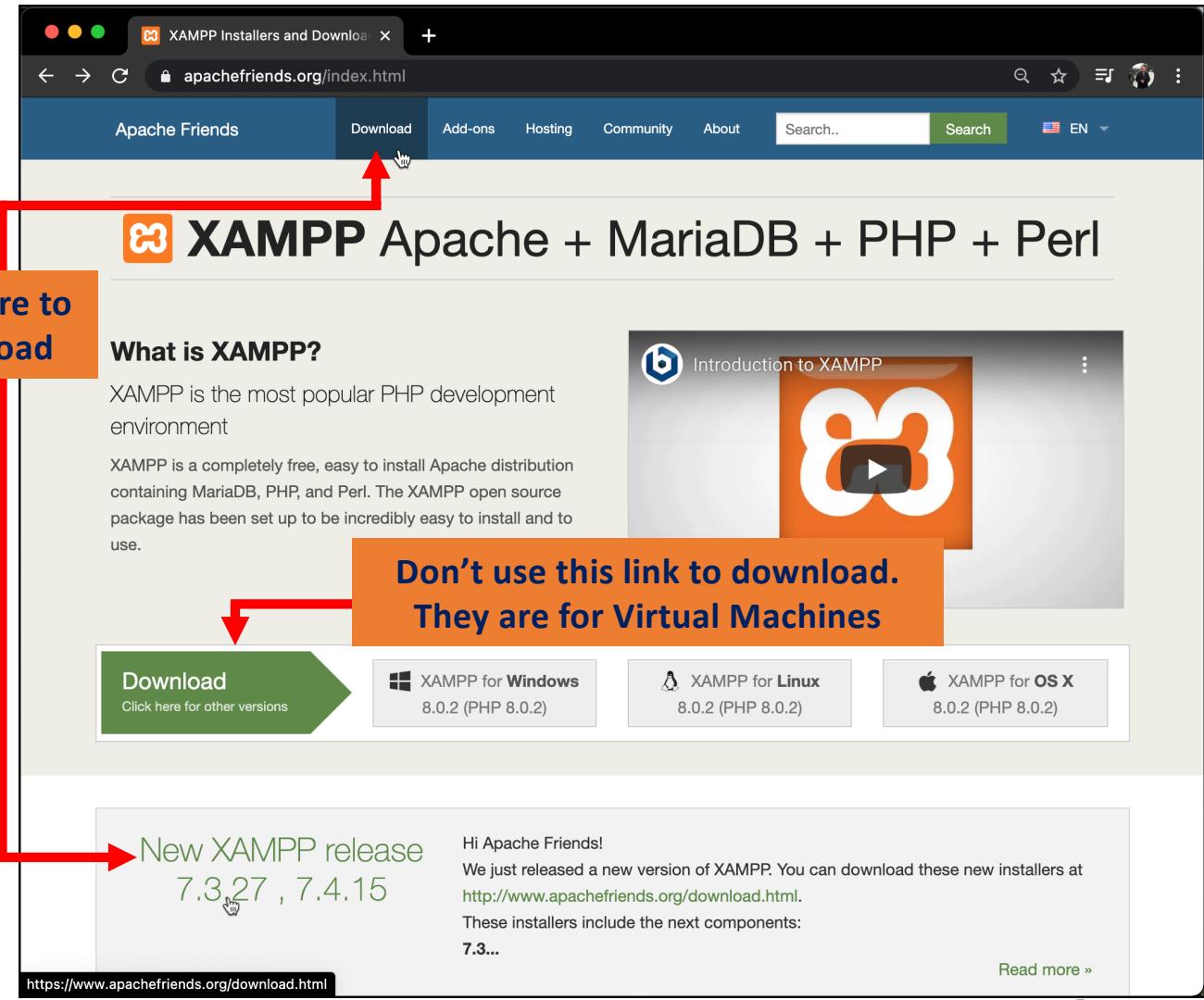
## A. WHY XAMPP??

## Why XAMPP?

> XAMPP provides server services such as web hosting, database ( MySQL), file transfer protocol (FileZilla), mail (Mercury) & also PHP & Perl programming.

> It can be used for educational, production, training purposes, etc. & is locally install in your PC or laptop.

Click here to download



## B. STARTING XAMPP SERVICE

## Starting XAMPP Service.

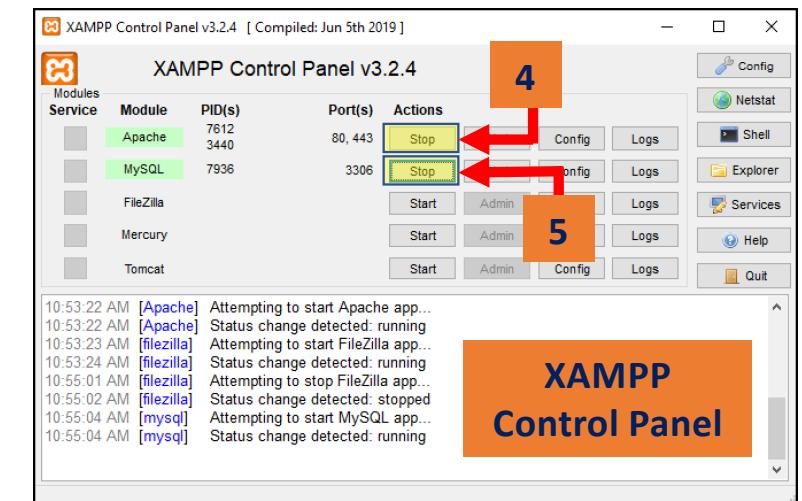
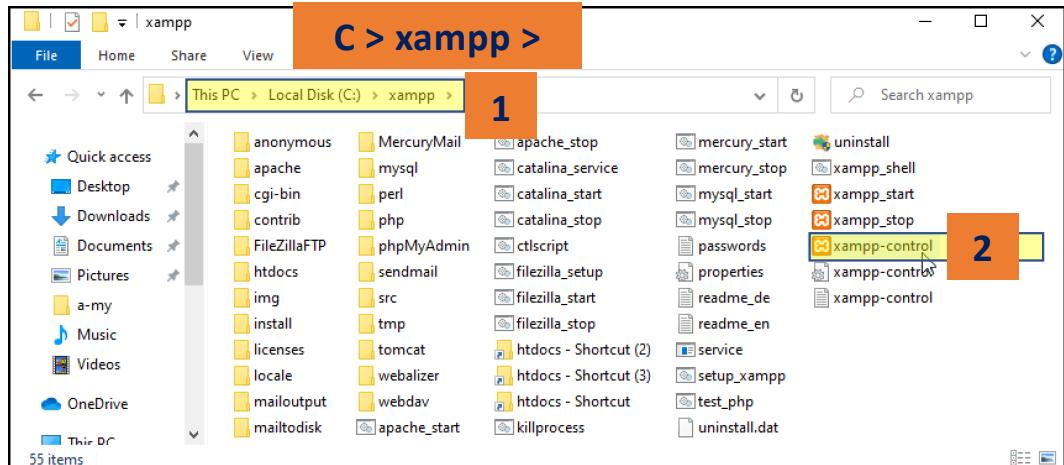
> Default XAMPP location is at

**1 C://xampp.**

> Click **2 xampp-control.exe** to launch the its **Control Panel** or just click **3 XAMPP's shortcut** at desktop.

> At **XAMPP's Control Panel**, click only at **2 start buttons** labelled **4 Apache** and **5 MySQL**. They should be lit up in green.

**NOTE:** If any or both of the services are not deployed, check your PC/laptop's **firewall, VM, antivirus** etc. Perhaps you might need to change the port number. Do Google the error message.



## Connecting to localhost.

- 6 Open your preferred browser & type **localhost** at address bar. Press enter and you will see the XAMPP's main page.

## Connecting to phpMyAdmin.

- 7 Click at **phpMyAdmin** link in order to open **MySQL Control Panel**.

<http://localhost/>

<http://localhost/phpmyadmin/>

XAMPP Apache + MariaDB + PHP + Perl

localhost / 127.0.0.1 | phpMyAdmin

phpMyAdmin

Recent Favorites

- New
- information\_schema
- mysql
- performance\_schema
- phpmyadmin
- test

Databases SQL Status User accounts Export Import More

General settings

Server connection collation: utf8mb4\_unicode\_ci

Appearance settings

Language English

Theme: pmahomme

Database server

- Server: 127.0.0.1 via TCP/IP
- Server type: MariaDB
- Server connection: SSL is not being used
- Server version: 10.4.17-MariaDB - mariadb.org binary distribution
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8mb4)

v1-mar-21 Web server

MPP for Windows 7.3.26

XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components in the system. Check the FAQs section or check the HOW-TO Guides for getting started with PHP

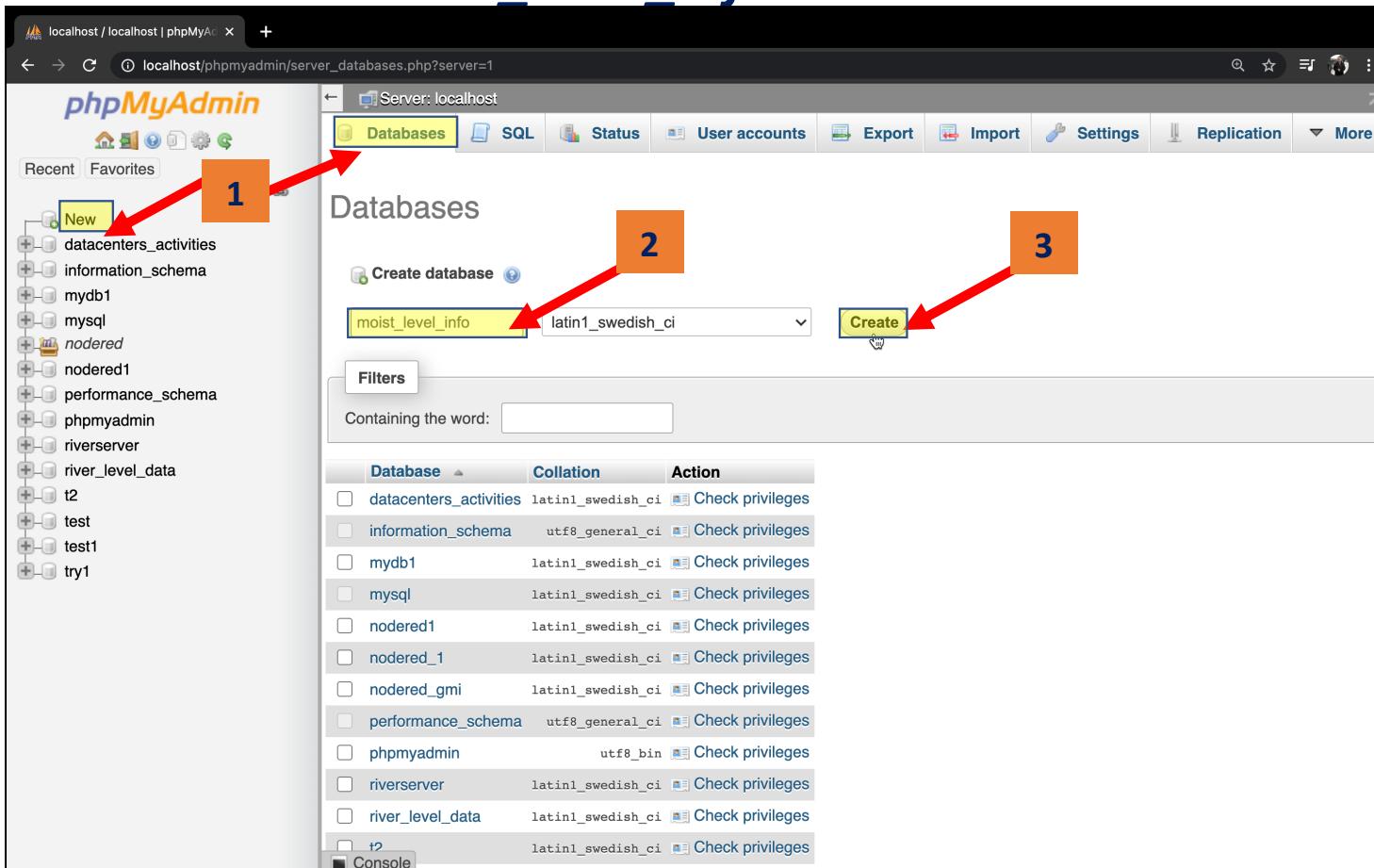
## Creating Your Database

> Referring to **BDA Case Study 1** exercise, create a database that will store data from Soil Moisture Sensor. The database must also store log time & log date. Use the setup as shown in the table.

Database Name	moist_level_info		
Table Name	plot1		
3 Columns			
NAME	TYPE	LENGTH	ADDITIONAL SETTING
dataID	INT	11	Index: <b>PRIMARY</b>
			A.I.: ✓ **A.I. = Auto Increment
logged	DATETIME	6	Default: <b>CURRENT_TIMESTAMP</b>
moisture_level	VARCHAR	100	

## Creating Your Database: Set Database Name

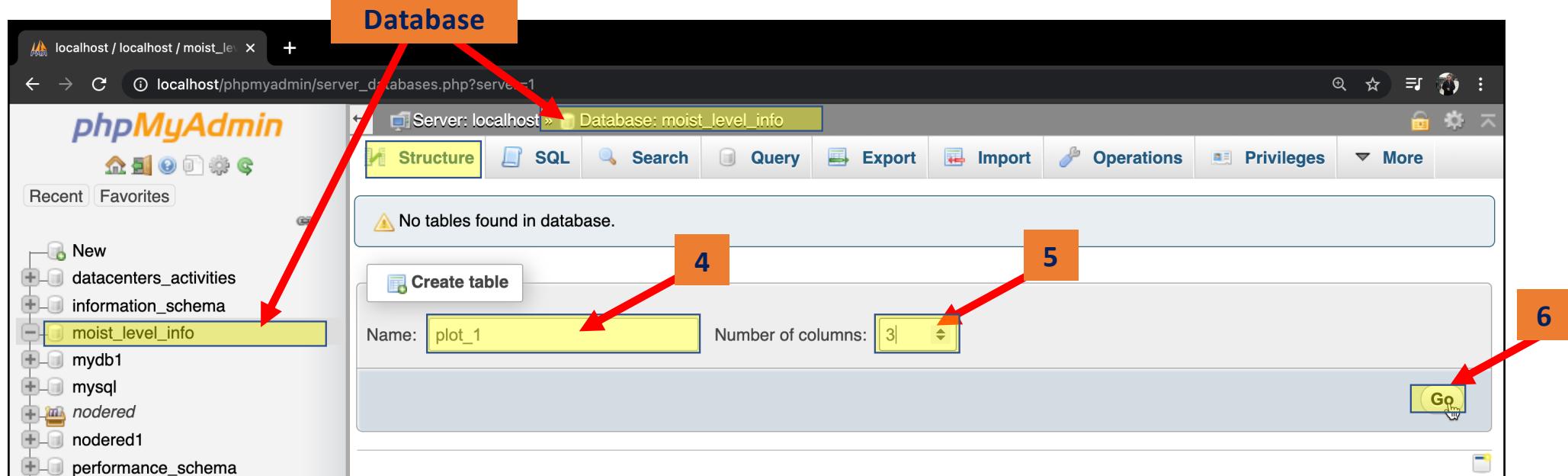
### Step 1 >> Database Name: *moist\_level\_info*



## Creating Your Database: Set Table Name

Step 2 >> Table Name: *plot\_1*

>> Number of columns: 3



## Creating Your Database: Columns Properties

### Step 3 >> Table Name: *plot\_1* ; columns: 3

> Database name:  
**moist\_level\_info**

> Table name:  
**plot\_1**

> 3 Columns:  
Name: **id**  
Type: **INT**  
Length: **15**  
Index: **PRIMARY**  
Auto Increment

Name: **logs**  
Type: **DATETIME**  
Default: **CURRENT\_TIME**

Name: **reading**  
Type: **VARCHAR**  
Length: **100**

Database name & table name

\*\* A.I. = Auto Increment

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
id	INT	15	None			PRIMARY	A.I. C
logs	DATETIME		CURRENT_TIME				
reading	VARCHAR	100					

Add index

Index name: PRIMARY  
Index choice: PRIMARY  
Column: id [int]

Save

## Creating Your Database: Your Database Structure

### Step 4 >> Keyword: Change - edit / Drop - delete

localhost / localhost / moist\_level\_info / plot\_1

Server: localhost » Database: moist\_level\_info » Table: plot\_1

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(15)			No	None		AUTO_INCREMENT	Change  Drop  More
2	logs	datetime			No	CURRENT_TIMESTAMP			Change  Drop  More
3	reading	varchar(100)	latin1_swedish_ci		No	None			Change  Drop  More

Check all With selected: Browse Change Drop Primary Unique Index Fulltext Add to central columns Remove from central columns

Add 1 column(s) after reading Go

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit  Drop	PRIMARY	BTREE	Yes	No	id	0	A	No	

Create an index on 1 columns Go

Partitions

No partitioning defined!

Partition table

localhost/phpmyadmin/db\_structure.php?server=1&db=moist\_level\_info&table=plot\_1#

**Go to Structure if you want to add, edit or delete column(s)**

**You may add new column(s) here, whether in between , first or end of column**

## Creating Your Database: Seeing Your Data

### Step 5 >> Browse tab

The screenshot shows the phpMyAdmin interface. On the left, the database tree is visible with the 'moist\_level\_info' database selected. Inside it, the 'plot\_1' table is selected. The main area is the 'Browse' tab, which displays the results of the SQL query: 'SELECT \* FROM `plot\_1`'. A message at the top states: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0021 seconds.)'. Below the message, the SQL query is shown again. The results grid has columns labeled 'id', 'logs', and 'reading'. An orange box with the text 'Empty since no data stored' and a red arrow points to the 'reading' column header. At the bottom of the results grid, there are 'Query results operations' buttons: 'Create view' and 'Bookmark this SQL query'. The 'Label:' field is empty, and there is a checkbox for 'Let every user access this bookmark'. A 'Bookmark this SQL query' button is also present. At the very bottom of the page is a 'Console' section.

## Creating Your Database: Time to Test

Step 6 >> Go to Insert tab & type a value at reading portion & press GO.

The screenshot shows the phpMyAdmin interface for the 'moist\_level\_info' database. The 'plot\_1' table is selected. The 'Insert' tab is active. In the 'Value' column for the 'reading' field, the value '999' is typed. A red arrow points to this input field. A callout box with the text 'Just type any value at this field only' is overlaid on the interface. The 'Go' button is highlighted with a yellow box.

Column	Type	Function	Null	Value
id	int(15)			
logs	datetime			CURRENT_TIMESTAMP
reading	varchar(100)			999

**Just type any value at this field only**

**Go**

## Creating Your Database: Seeing Your Data

### Step 7 >> Success saved data into *plot\_1* table

> If error message pop up, go back to structure and check the columns configuration.

> Check also the data type of the information entered, whether the information fits with the data type or not.

The screenshot shows the phpMyAdmin interface for the 'moist\_level\_info' database. The 'Structure' tab is selected. In the SQL tab, the following query was run:

```
INSERT INTO `plot_1`(`id`, `logs`, `reading`) VALUES (NULL, CURRENT_TIMESTAMP, '999');
```

A green success message box displays: "1 row inserted. Inserted row id: 1". The 'Columns' panel on the right lists 'id', 'logs', and 'reading'. At the bottom, there are buttons for SELECT\*, SELECT, INSERT, UPDATE, DELETE, Clear, Format, and Get auto-saved query.

## Creating Your Database: Seeing Your Data Again

### Step 8 >> Click *Browse* & what did you see?

- > Id is auto increment & if you delete, the new data will get new running Id instead of the Id of the deleted data.
- > Check also the data type of the information entered, whether the information fits with the data type or not.

phpMyAdmin

Recent Favorites

- New
- datacenters\_activities
- information\_schema
- moist\_level\_info
  - New
  - plot\_1
- mydb1
- mysql
- nodered
- nodered1
- performance\_schema
- phpmyadmin
- riverserver
- river\_level\_data
- t2
  - New
  - t3
- test
- test1
- try1

Server: localhost » Database: moist\_level\_info » Table: plot\_1

Browse Structure SQL Search Insert Export Import More

Showing rows 0 - 0 (1 total, Query took 0.0015 seconds.)

SELECT \* FROM `plot\_1`

Profiling [Edit inline] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all Number of rows: 25 Filter rows: Search this table

+ Options id logs reading

	Edit	Copy	Delete
1	2021-03-11 00:02:54	999	

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

Label: Let every user access this bookmark

localhost/phpmyadmin/sql.php?db=moist\_level\_info&table=plot\_1

v1-mar-21

Your first data

## C. TESTING NODERED AND MYSQL - MANUAL PROCESS

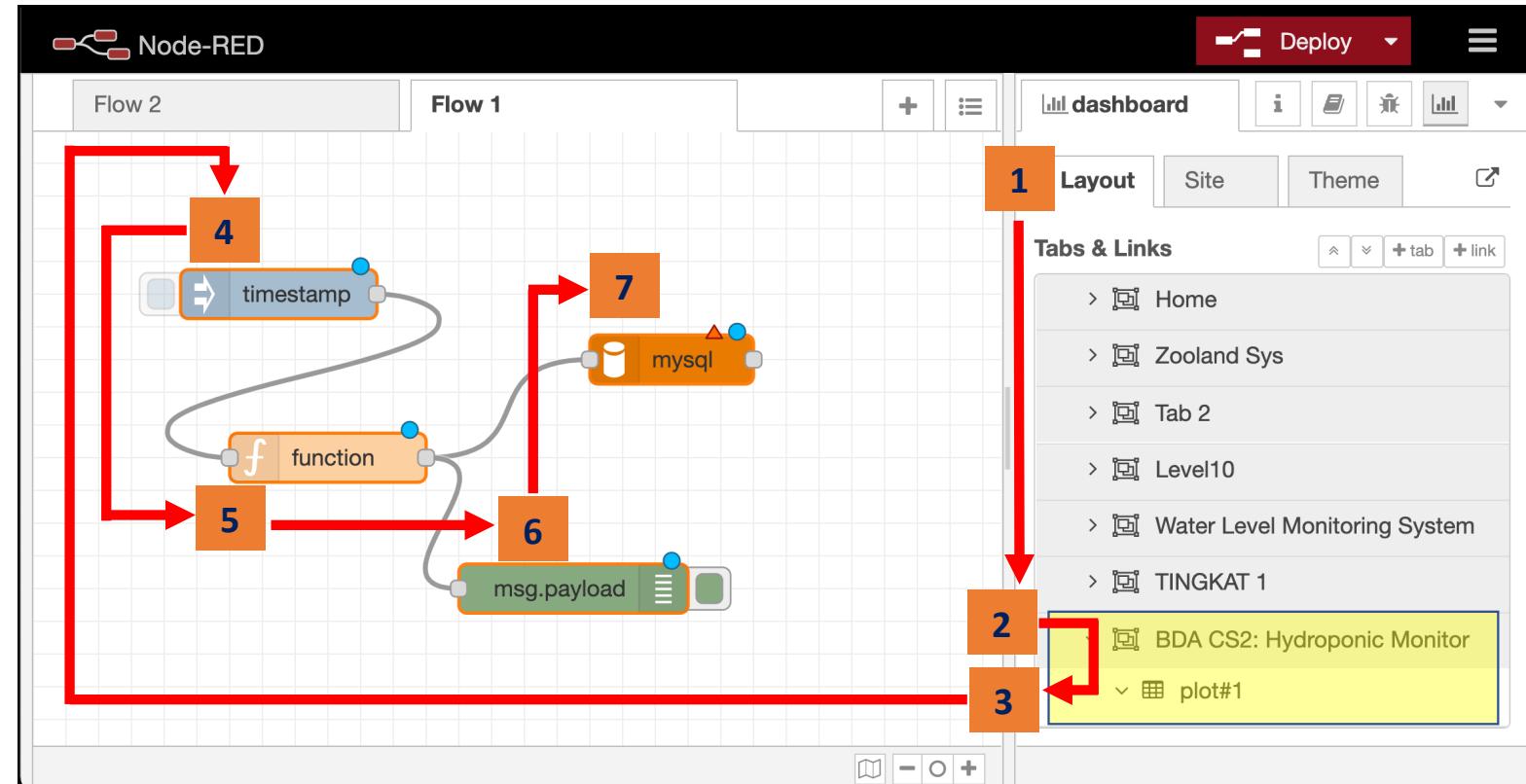
## nodeRED + MySQL: Manually Inject into Database

### Step 1 >> Storyboard

> The dashboard must show a chart that display the reading of soil moisture value of individual plot before the value stored in database.

> Go to **Layout**, add **Tab**, name it as **BDA2 CS: Hydroponic Monitor & Group**, name it as **plot#1**.

> Drag & drop nodes & connect them with wires as in the figure.



## nodeRED + MySQL: Manually Inject into Database

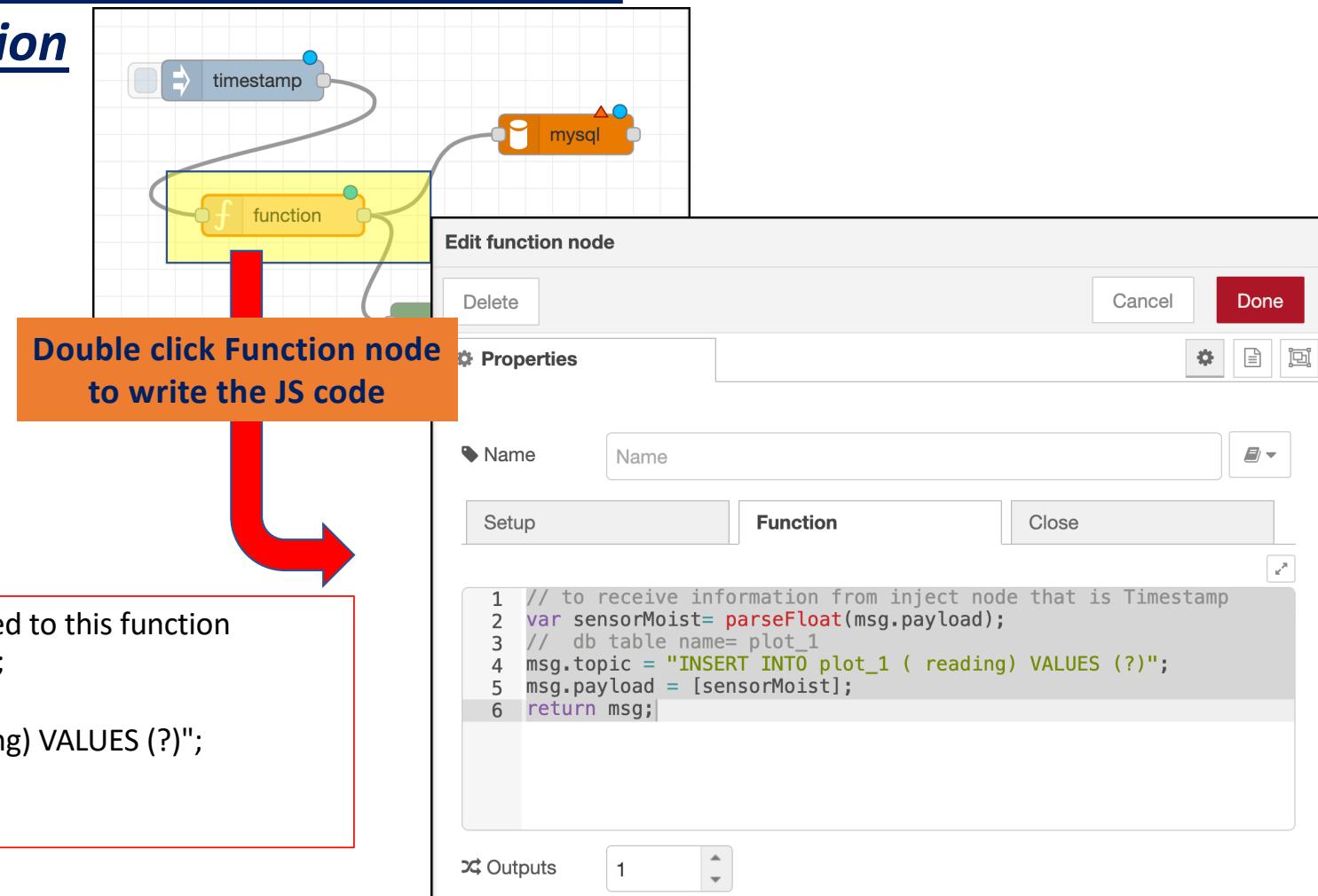
### Step 2 >> Writing Function

> The Function node allows JavaScript code to be run against the messages that are passed through it.

> The task over this code is to **insert the dummy data into plot\_1 at reading column.**

**>>DONE**

```
// to receive information from node wired to this function
var sensorMoist= parseInt(msg.payload);
// db table name= plot_1
msg.topic = "INSERT INTO plot_1 ( reading) VALUES (?)";
msg.payload = [sensorMoist];
return msg;
```



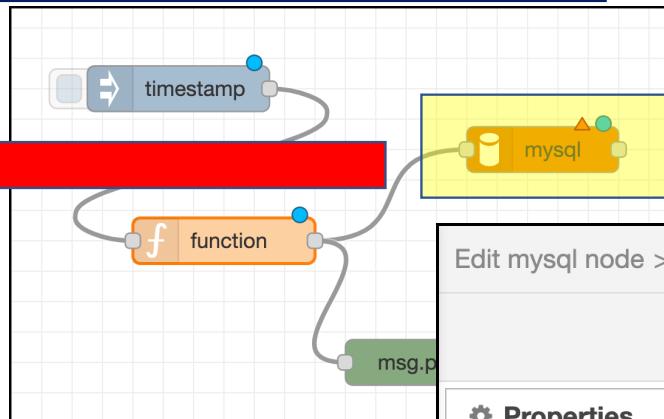
## nodeRED + MySQL: Manually Inject into Database

### Step 3 >> The Database

> Need to configure the mysql node.

> Refer to [slide 9](#).

>>ADD  
>>DONE



Double click mysql node to configure the properties

Edit mysql node > Add new MySQLdatabase config node

Cancel Add

Properties

Host: 127.0.0.1  
Port: 3306  
User: root  
Password: (empty)  
Database: moist\_level\_info  
Timezone: (empty)

Enabled: 0 nodes use this config  
On all flows

Click [Add](#) – to go back to previous page

User: root

Database name: moist\_level\_info

Edit mysql node

Delete Cancel Done

Properties

Database: Add new MySQLdatabase...  
Name: Name

## nodeRED + MySQL: Manually Inject into Database

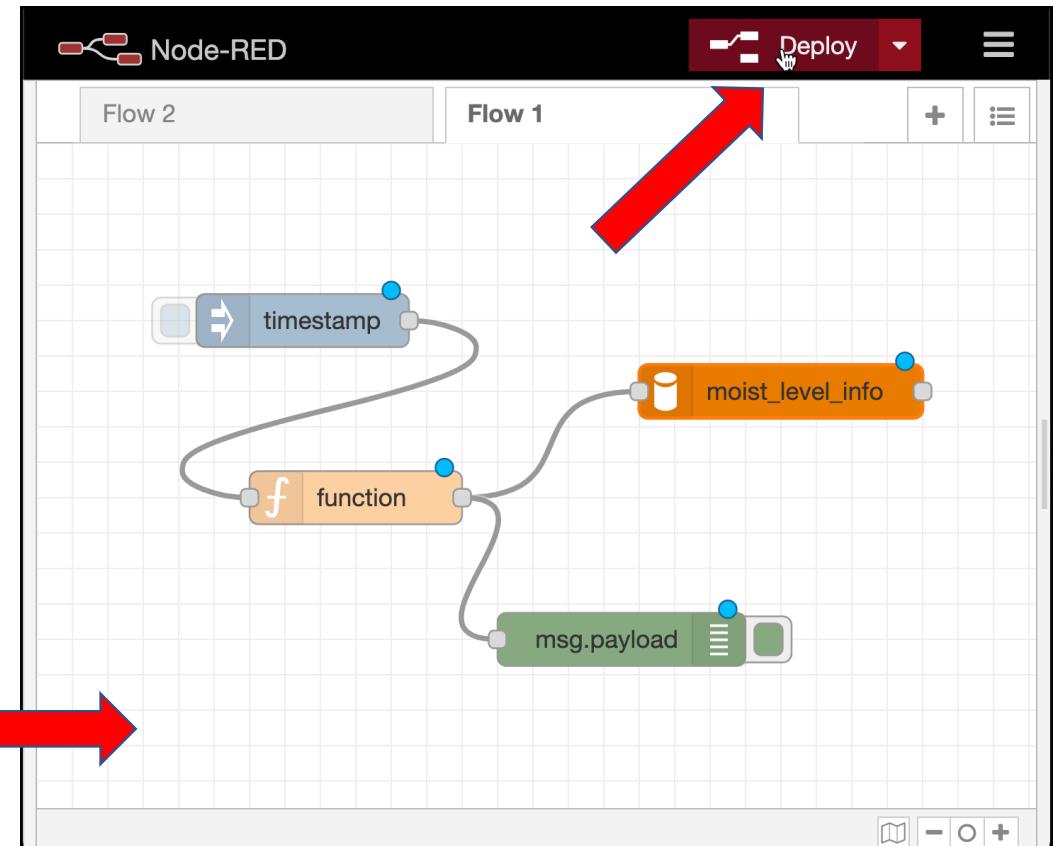
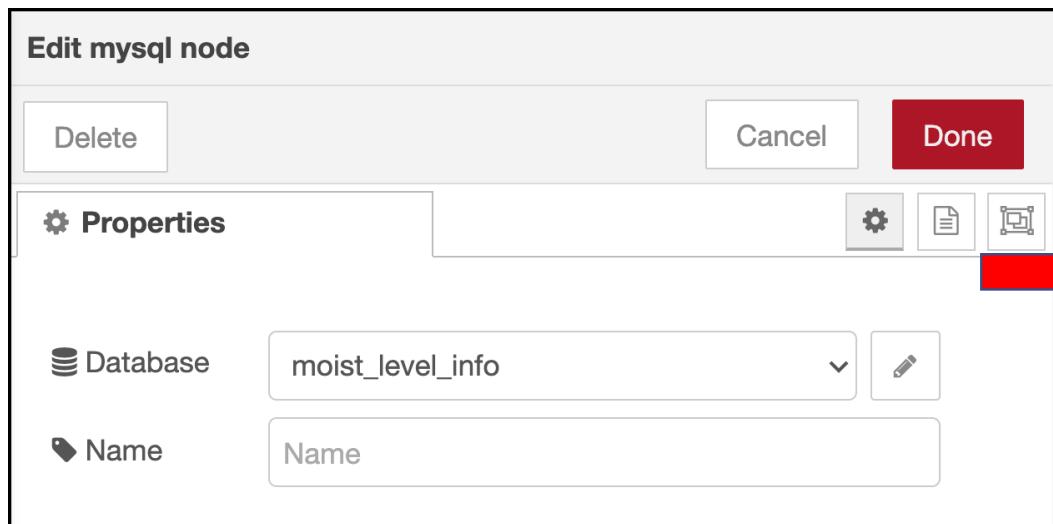
### Step 4 >> Time to Test

> Make sure that you are using the correct database, which may avoid rejected data from stored in database.

> The blue dot  indicates that the system has been configured properly.

> Don't forget to click Deploy upon completion.

> Click the inject button & observe the output at Debug section.



>>DONE  
>>Deploy

## nodeRED + MySQL: a. Manually Inject into Database

### Step 4 >> The Results

<http://localhost:1880/>

```

graph LR
    timestamp((timestamp)) --> function(function)
    function --> payload[msg.payload]
    payload --> moist_moist[moist_level_info]
    
```

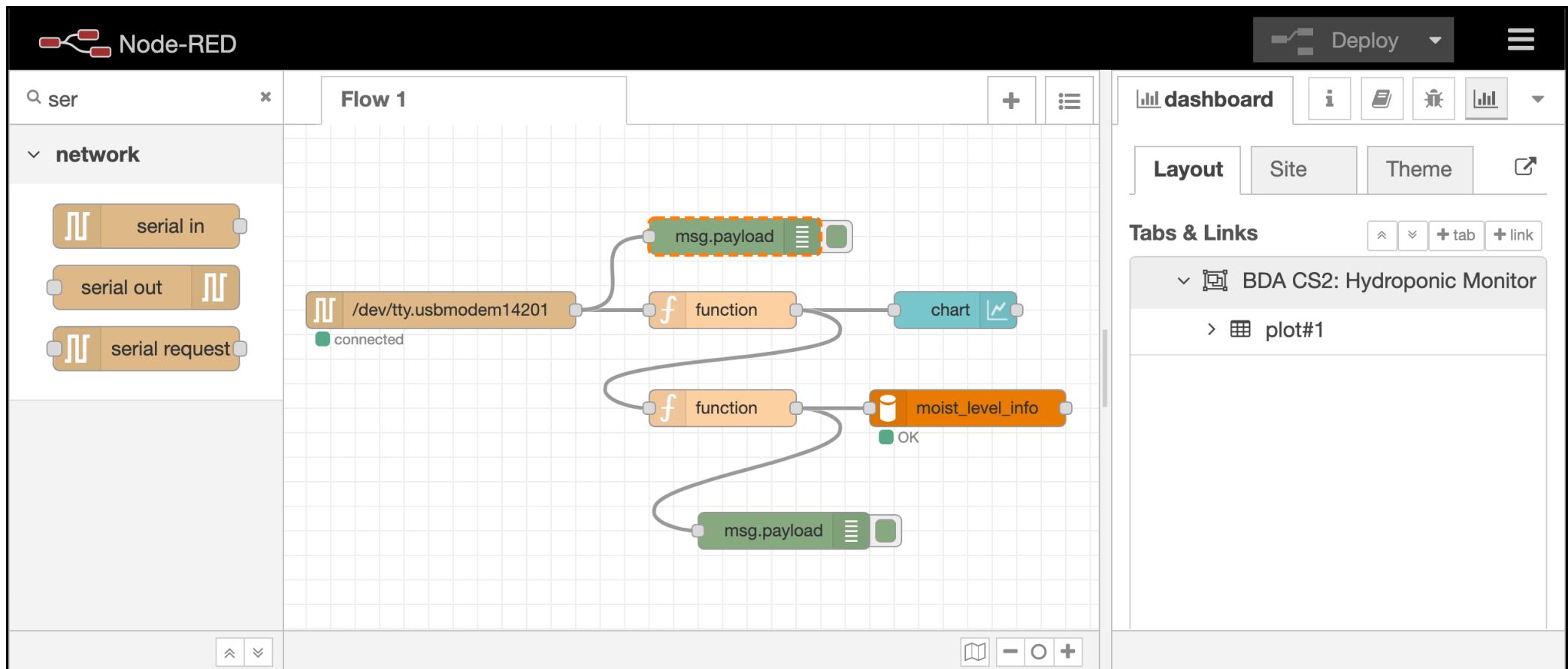
<http://localhost/phpmyadmin>

	id	logs	reading
1	2021-03-11 00:02:54	999	
2	2021-03-11 23:08:11	777	
3	2021-03-11 23:08:18	777	
4	2021-03-11 23:08:24	777	
5	2021-03-11 23:11:05	1615475465384	

## D. MICROCONTROLLER + NODERED + MYSQL

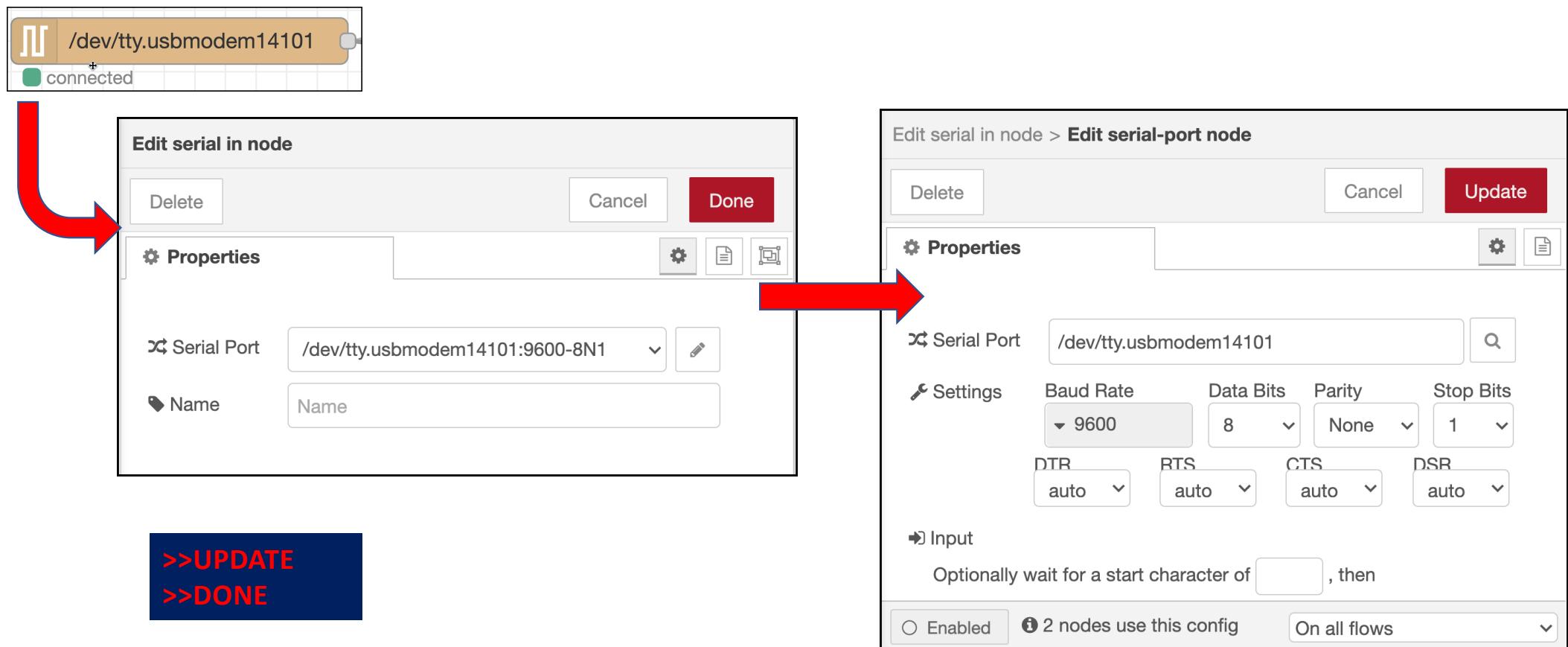
## nodeRED + MySQL: From Sensor to nodeRED to Database

Step 1 >> Hook up your workspace as follows:



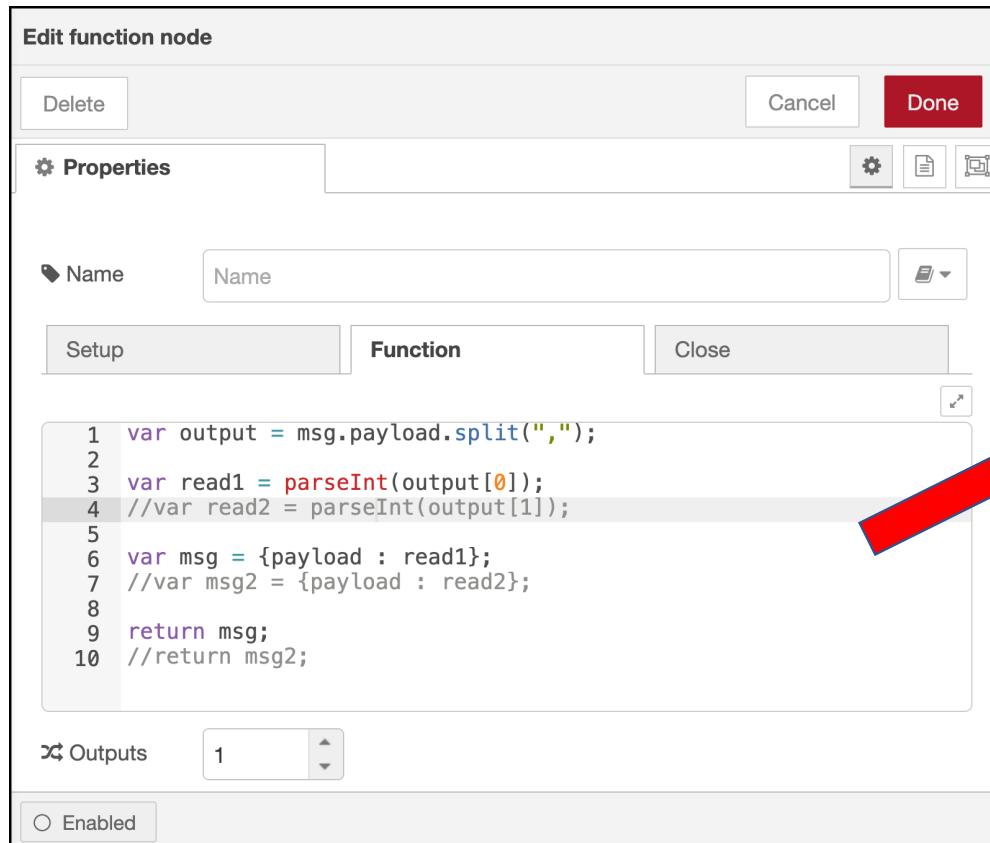
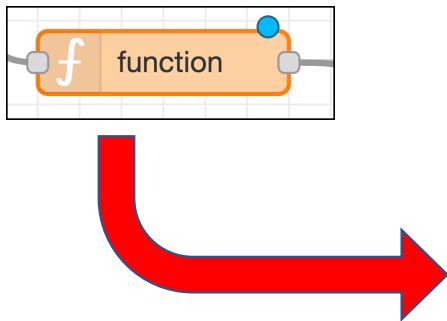
## nodeRED + MySQL: From Sensor to nodeRED to Database

### Step 1 >> Recap: Configuring *Serial In* node.



## nodeRED + MySQL: From Sensor to nodeRED to Database

### Step 2 >> Recap: Configuring *Function* node.

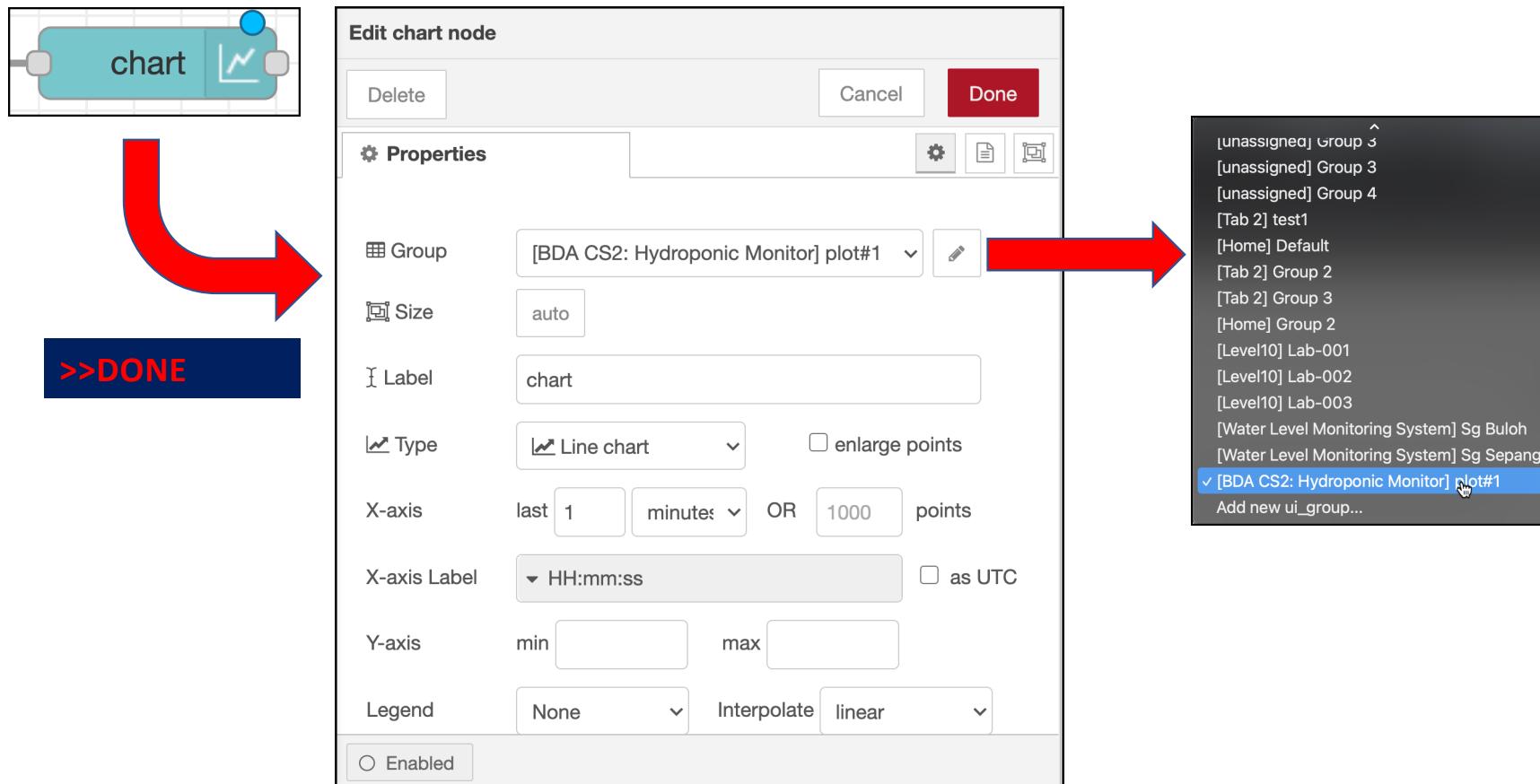


```
var output =  
msg.payload.split(",");  
  
var read1 = parseInt(output[0]);  
//var read2 = parseInt(output[1]);  
  
var msg = {payload : read1};  
//var msg2 = {payload : read2};  
  
return msg;  
//return msg2;
```

**When js sees “,” , it will split into  
2: as output[0] & output[1]**

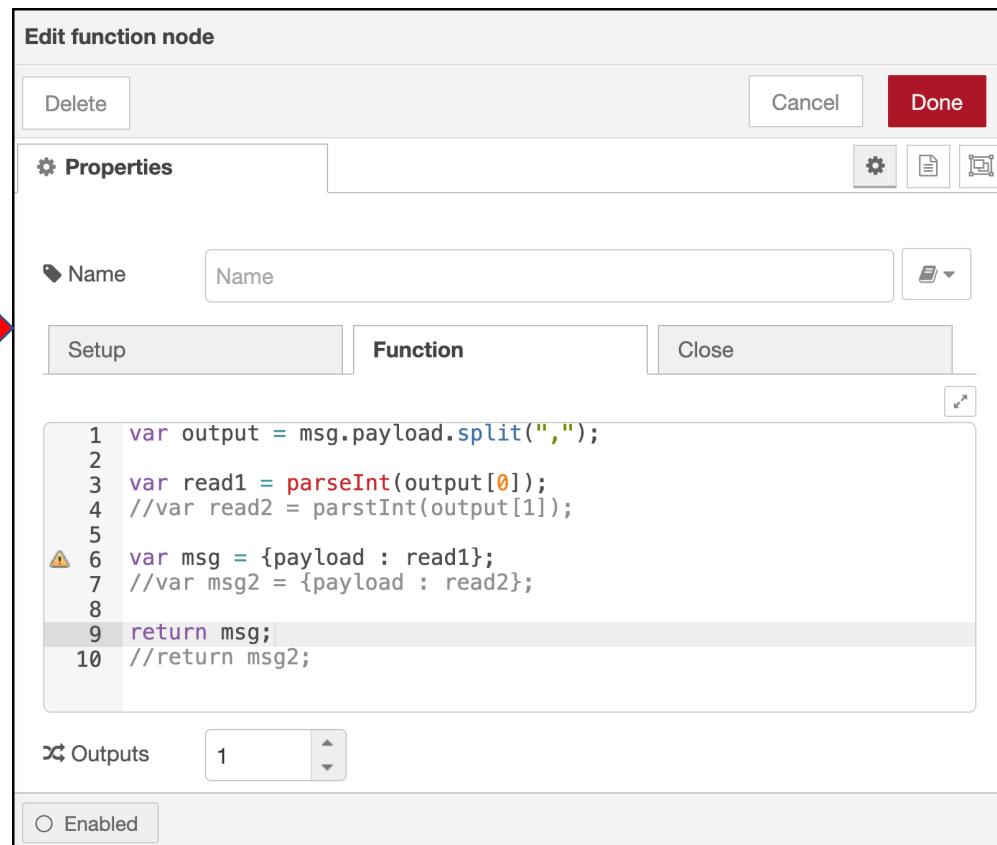
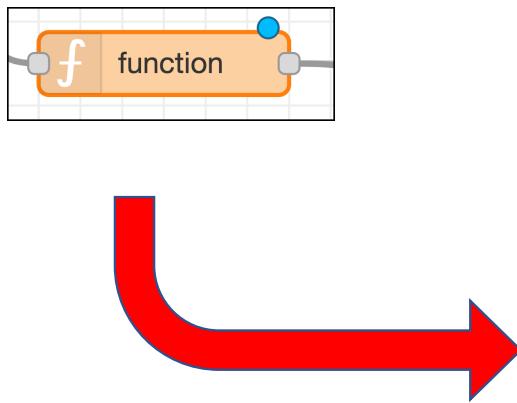
## nodeRED + MySQL: From Sensor to nodeRED to Database

### Step 3 >> Recap: Configuring *Chart* node.



## nodeRED + MySQL: From Sensor to nodeRED to Database

### Step 4 >> Recap: Setting on Function node.



```
var output =
msg.payload.split(",");
var read1 = parseInt(output[0]);
//var read2 = parstInt(output[1]);

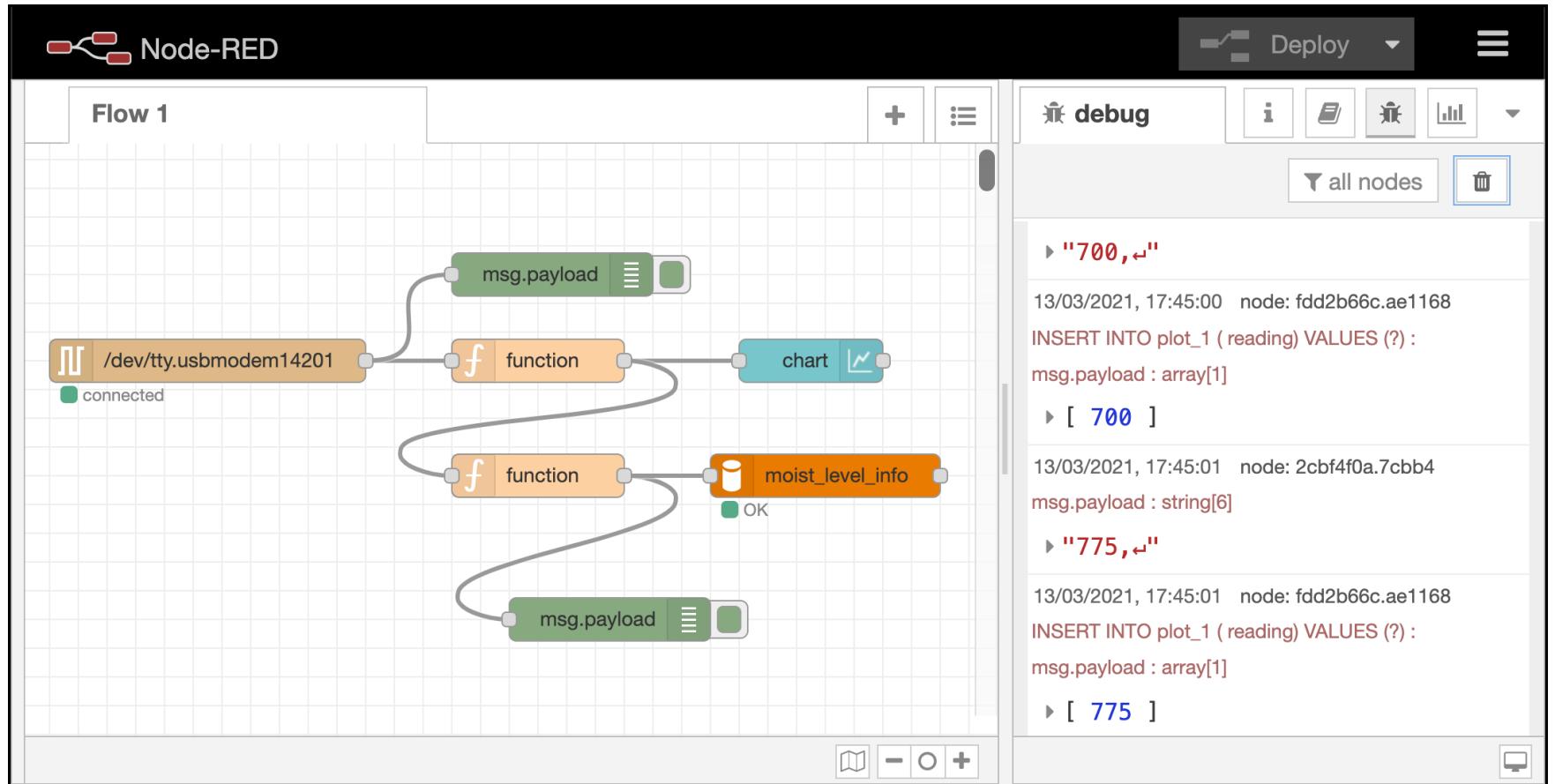
var msg = {payload : read1};
//var msg2 = {payload : read2};

return msg;
//return msg2;
```

When js sees “,” , it will  
split into 2: i.e., output[0]  
& output[1]

## nodeRED + MySQL: From Sensor to nodeRED to Database

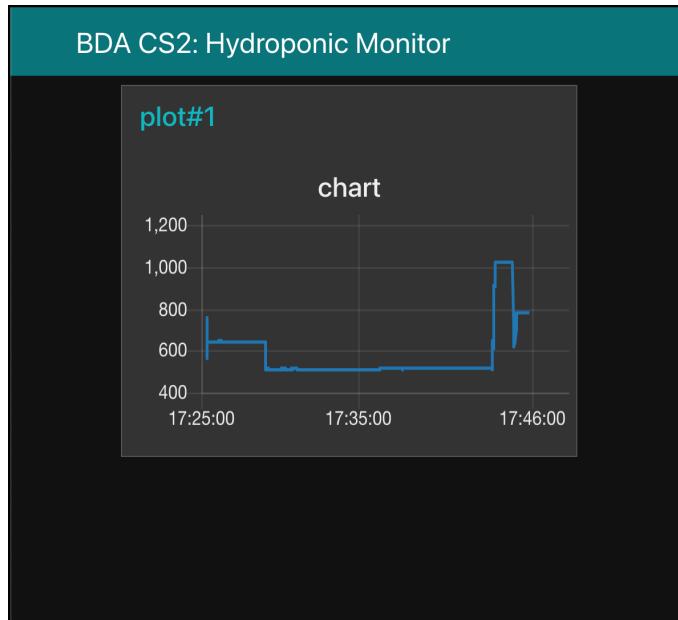
### Step 5 >> Click Deploy & observe the output at Debug window.



## nodeRED + MySQL: From Sensor to nodeRED to Database

### Step 6 >> Browse Dashboard & phpMyAdmin – observe both results.

<http://localhost:1880/ui>



<http://localhost/phpmyadmin>

Server: localhost » Database: moist\_level\_info » Table: plot\_1

Browse Structure SQL Search Insert Export Import Privileges More

Showing rows 0 - 24 (835 total, Query took 0.0016 seconds.) [logs: 2021-03-13 17:46:07... - 2021-03-13 17:45:43...]

SELECT \* FROM `plot\_1` ORDER BY `logs` DESC

Profiling [Edit inline] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options	id	logs	reading
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	835	2021-03-13 17:46:07	779
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	834	2021-03-13 17:46:06	779
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	833	2021-03-13 17:46:05	779
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	832	2021-03-13 17:46:04	779
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	831	2021-03-13 17:46:03	779
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	830	2021-03-13 17:46:02	779
<input checked="" type="checkbox"/> <a href="#">Console</a> <a href="#">Copy</a> <a href="#">Delete</a>	829	2021-03-13 17:46:01	779

BDA CASE STUDY 2: nodeRED with Database - MySQL

## Solution:

```
[{"id": "7f9ea9bc.c4f338", "type": "tab", "label": "Flow 1", "disabled": false, "info": ""}, {"id": "7c46fdbcb810c4", "type": "mysql", "z": "7f9ea9bc.c4f338", "mydb": "e2373d64.93b03", "name": "", "x": 620, "y": 260, "wires": [[], {"id": "fdd2b66c.ae1168", "type": "debug", "z": "7f9ea9bc.c4f338", "name": "", "active": true, "tosidebar": true, "console": false, "tostatus": false, "complete": "false", "statusVal": "", "statusType": "auto", "x": 470, "y": 360, "wires": []}], {"id": "a67e8bd9.350ac8", "type": "function", "z": "7f9ea9bc.c4f338", "name": "", "func": "// to receive data pass from node wired to this function\nvar sensorMoist= parseInt(msg.payload);\n// db table name= plot_1\nmsg.topic = \"INSERT INTO plot_1 ( reading) VALUES (?)\";\nmsg.payload = [sensorMoist];\nreturn msg;", "outputs": 1, "noerr": 0, "initialize": "", "finalize": "", "x": 420, "y": 260, "wires": [{"id": "fdd2b66c.ae1168", "z": "7c46fdbcb810c4"}]}, {"id": "e1186575.87d208", "type": "ui_chart", "z": "7f9ea9bc.c4f338", "name": "", "group": "7fb64bc4.bf6d94", "order": 0, "width": 0, "height": 0, "label": "chart", "chartType": "line", "legend": "false", "xformat": "HH:mm:ss", "interpolate": "linear", "nodata": "", "dot": false, "ymin": "", "ymax": "", "removeOlder": 1, "removeOlderPoints": "", "removeOlderUnit": "3600", "cutout": 0, "useOneColor": false, "useUTC": false, "colors": ["#1f77b4", "#aec7e8", "#ff7f0e", "#2ca02c", "#98df8a", "#d62728", "#ff9896", "#9467bd", "#c5b0d5"], "useOldStyle": false, "outputs": 1, "x": 610, "y": 180, "wires": [[], {"id": "aca61c55.a3c9", "type": "serial_in", "z": "7f9ea9bc.c4f338", "name": "", "serial": "f1779b4e.c964f8", "x": 190, "y": 180, "wires": [{"id": "4cdba24.968c75c", "z": "2bf4f0a.7ccb4"}]}], {"id": "4cdba24.968c75c", "type": "function", "z": "7f9ea9bc.c4f338", "name": "", "func": "var output = msg.payload.split('\\');\nvar read1 = parseInt(output[0]);\nvar read2 = parseInt(output[1]);\nvar msg = {payload : read1};\nvar msg2 = {payload : read2};\n\nreturn msg;\n\nmsg2;\n", "outputs": 1, "noerr": 0, "initialize": "", "finalize": "", "x": 420, "y": 180, "wires": [{"id": "e1186575.87d208", "z": "a67e8bd9.350ac8"}]}, {"id": "2cbf4f0a.7ccb4", "type": "debug", "z": "7f9ea9bc.c4f338", "name": "", "active": true, "tosidebar": true, "console": false, "tostatus": false, "complete": "false", "statusVal": "", "statusType": "auto", "x": 430, "y": 120, "wires": []}, {"id": "e2373d64.93b03", "type": "MySQLdatabase", "z": "", "name": "", "host": "127.0.0.1", "port": "3306", "db": "moist_level_info", "tz": "", "charset": "UTF8"}, {"id": "7fb64bc4.bf6d94", "type": "ui_group", "z": "", "name": "plot#1", "tab": "deacc80.f9ff2", "order": 1, "disp": true, "width": 6, "collapse": false}, {"id": "f1779b4e.c964f8", "type": "serial-port", "z": "", "serialport": "/dev/tty.usbmodem14201", "serialbaud": "9600", ".databits": "8", "parity": "none", "stopbits": "1", "waitfor": "", "dtr": "none", "rts": "none", "cts": "none", "dsr": "none", "newline": "\\n", "bin": "false", "out": "char", "addchar": "", "responsetimeout": "10000"}, {"id": "ddeacc80.f9ff2", "type": "ui_tab", "z": "", "name": "BDA CS2: Hydroponic Monitor", "icon": "dashboard", "order": 7, "disabled": false, "hidden": false}]]
```



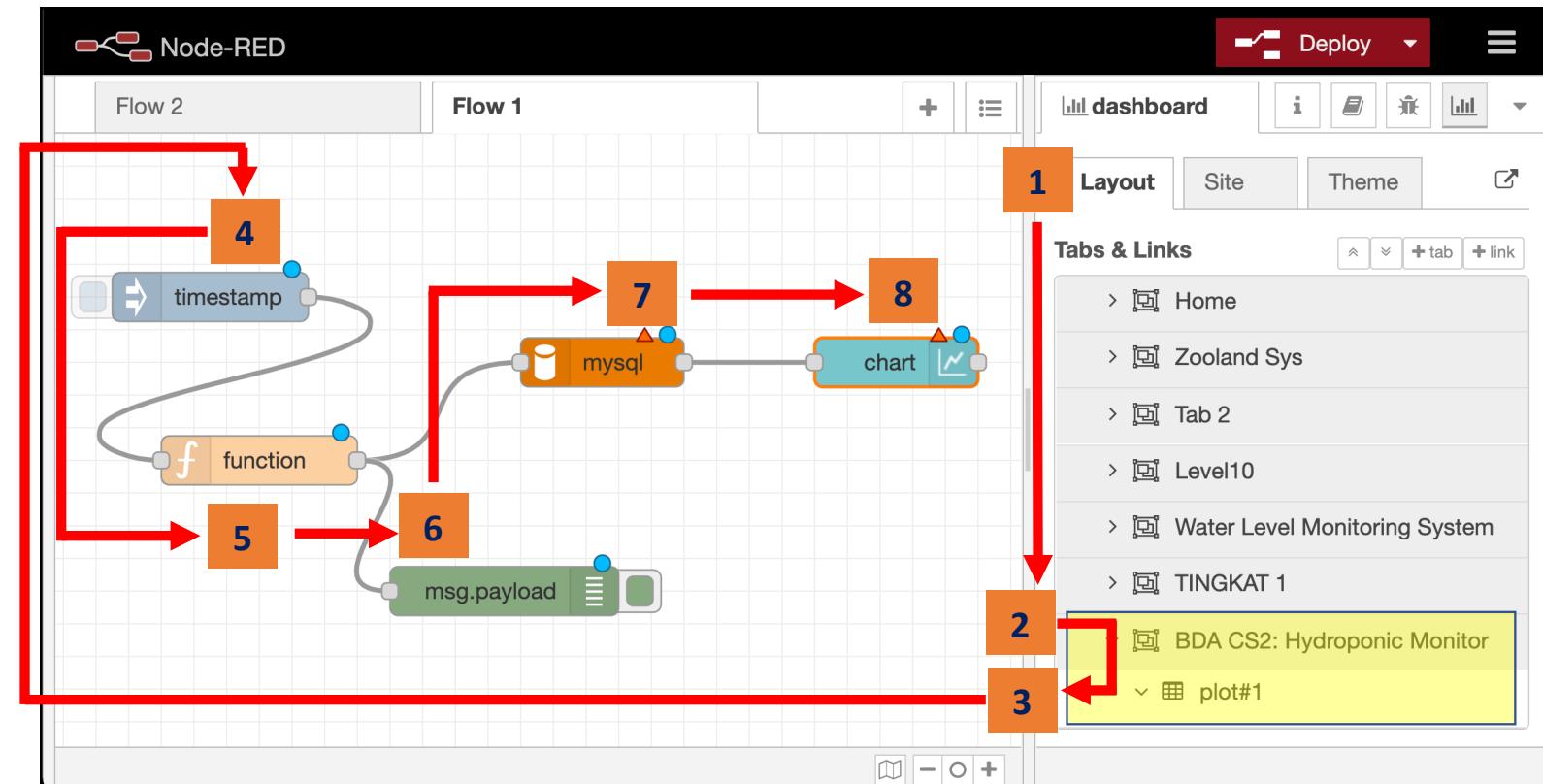
## nodeRED + MySQL: Manually Inject into Database

### Step 1 >> Storyboard

> The dashboard must show a chart that display the reading of soil moisture value of individual plot before the value stored in database.

> Go to **Layout**, add **Tab**, name it as **BDA2 CS: Hydroponic Monitor & Group**, name it as **plot#1**.

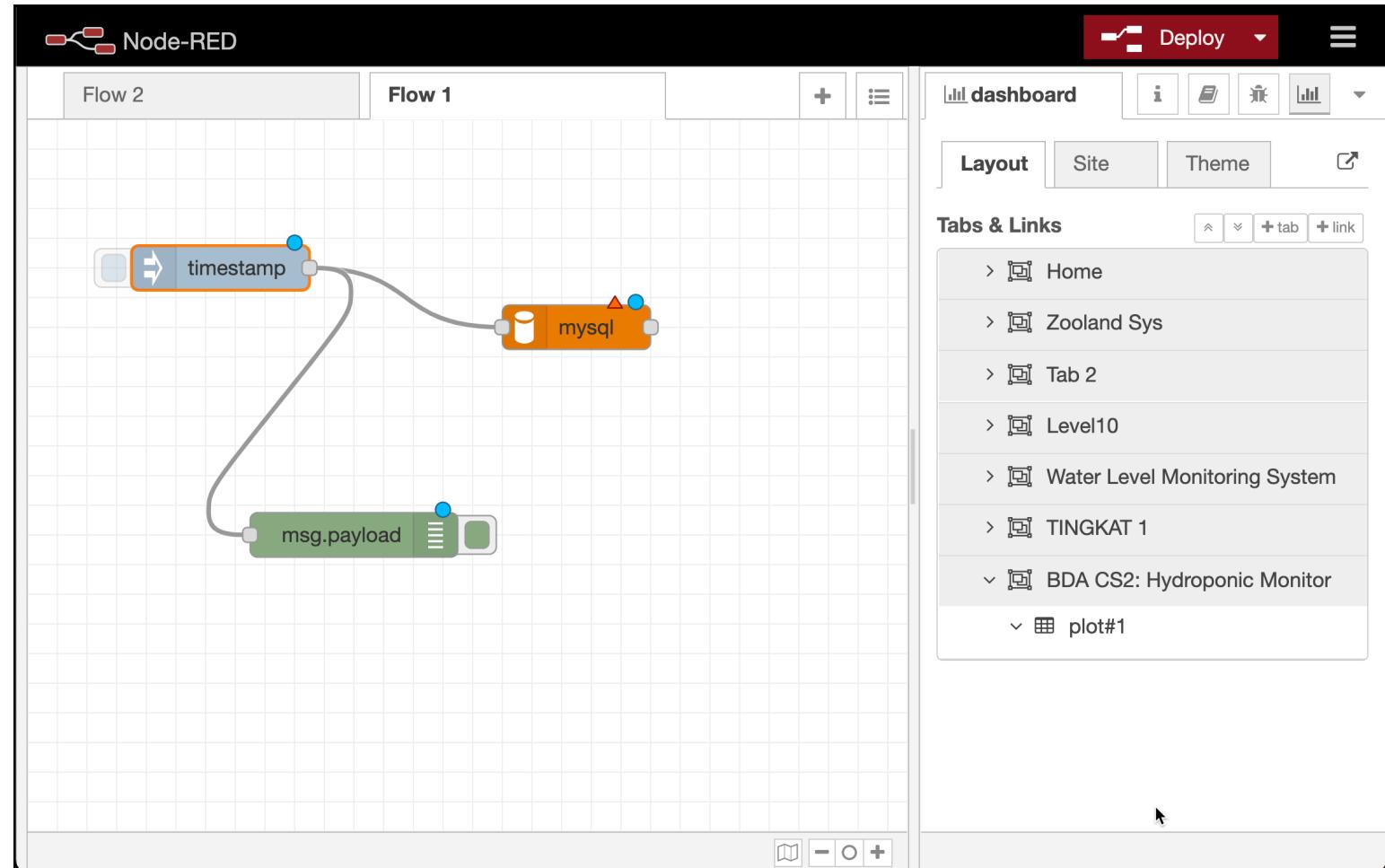
> Drag & drop nodes & connect them with wires as in the figure.



## nodeRED + MySQL: Manually Inject into Database

### Step 1 >> xxx

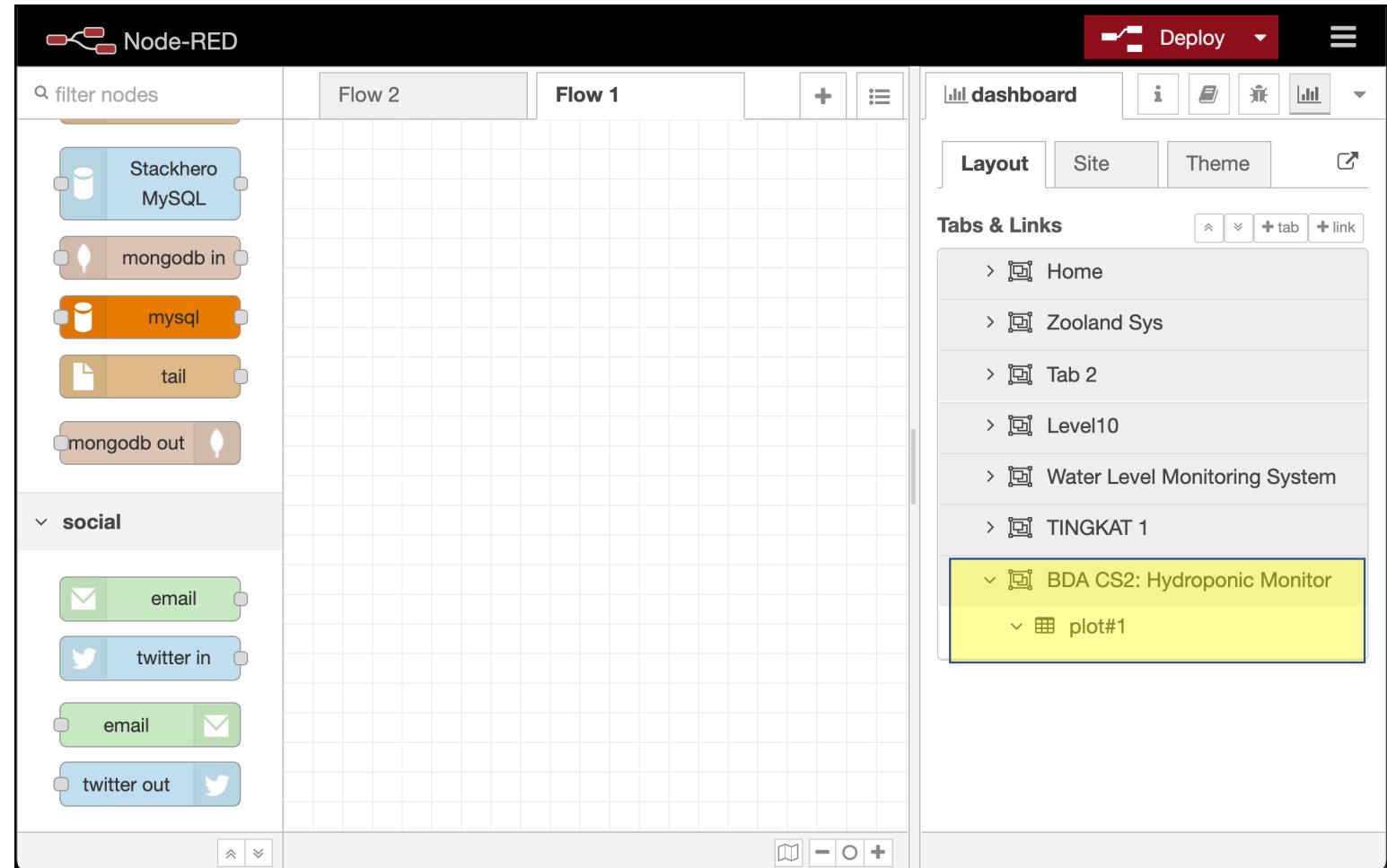
- > Id is auto increment,
- > Check also the data type of the information entered, whether the information fits with the data type or not.



## nodeRED + MySQL: Manually Inject into Database

### Step 1 >> xxx

- > Id is auto increment,
- > Check also the data type of the information entered, whether the information fits with the data type or not.



## Microcontroller: b. The sketch.

```
1 //IoT Based Smart Agriculture with Remote Monitoring System
2 //v1-mac2021
3 void setup() {
4     // initialize serial communication at 9600 bits per second:
5     Serial.begin(9600);
6 }
7
8 void loop() {
9     String moist; //set moist as string
10    int sensorValue = analogRead(A0); //read incoming value from analog pin 1 & put at variable named sensorValue
11    moist = String(sensorValue); //convert sensorValue from int to string -> only to display at nodeRED dashboard
12
13    Serial.print("Moisture Level: "); //remark this line when displaying the value at nodeRED dashboard
14    Serial.print(moist); //will print at Serial Monitor & nodeRED: debug node & serial in node
15    Serial.println(","); //delimiter -> for nodeRED -> to differentiate new data
16    delay(1000);      //pause for 1 sec
17 }
```

source: nodeRed-02mac21-agriculture-v1.ino

### Note:

Put a remark at line number 9 when using nodeRED dashboard. The chart node cannot read the serial value.

Refer to next slide.

### Microcontroller: c. Expected Output.

>Upload the sketch.

>**Error are expected** when you copy the sketch from previous page. Check the **double quotes** symbols. (""). Delete and replace new **double quotes**. (Always happens when CnP text form net or pdf)

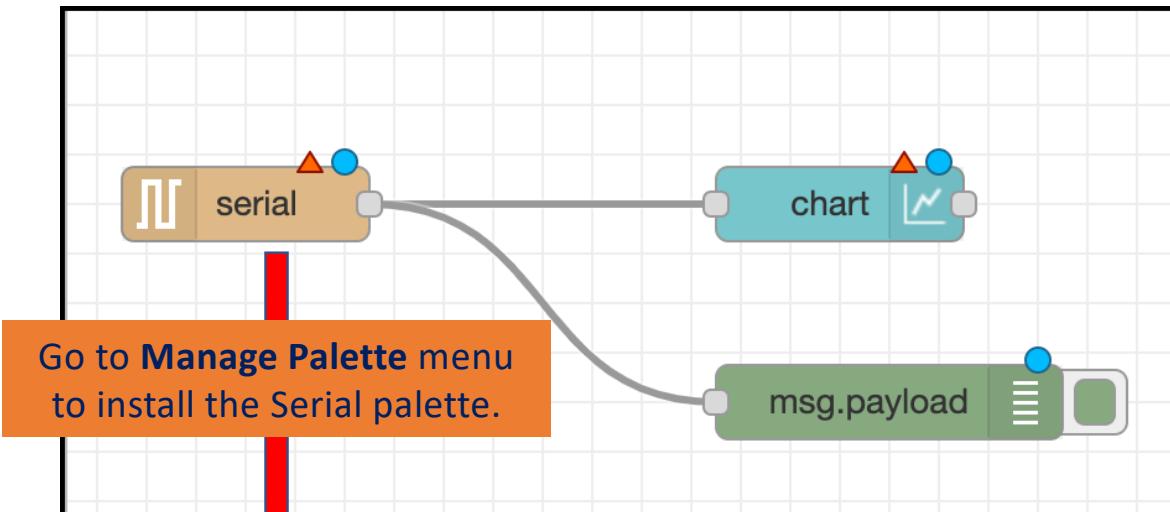
Moisture Level: 707,  
Moisture Level: 707,  
Moisture Level: 706,  
Moisture Level: 707,  
Moisture Level: 706,  
Moisture Level: 707,  
Moisture Level: 706,

Moisture Level: 706,

Autoscroll Show timestamp No line ending 9600 baud Clear output

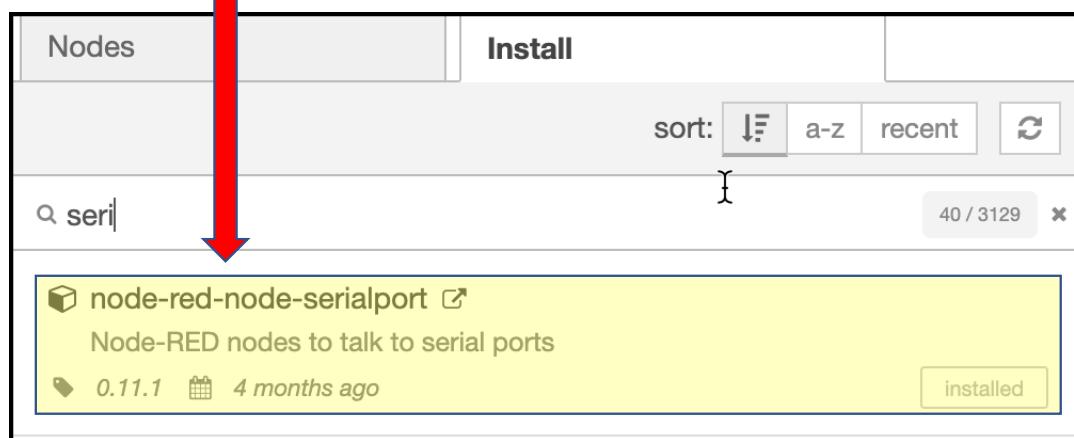
>**Delimiter** in line 15 is an indication for nodeRED to split data into array ([0],[1]...[n<sup>th</sup>]).

## Node-RED: d. Layout & Installation.



> The main idea is to display the moisture value over nodeRED's chart dashboard.

>**Serial in** node is used in for the microcontroller to talk (communicate) with the computer (with nodeRED installed).



>**Serial widget** is not installed by default. Go to Manage Palette to install **node-red-node-serialport**.

>  indicate nodes not properly setup.

## Node-RED: e. Setting the Layout.

> Since the project requires dashboard UI, it is advisable to start the task by configuring the **Layout** properties located at right side panel.

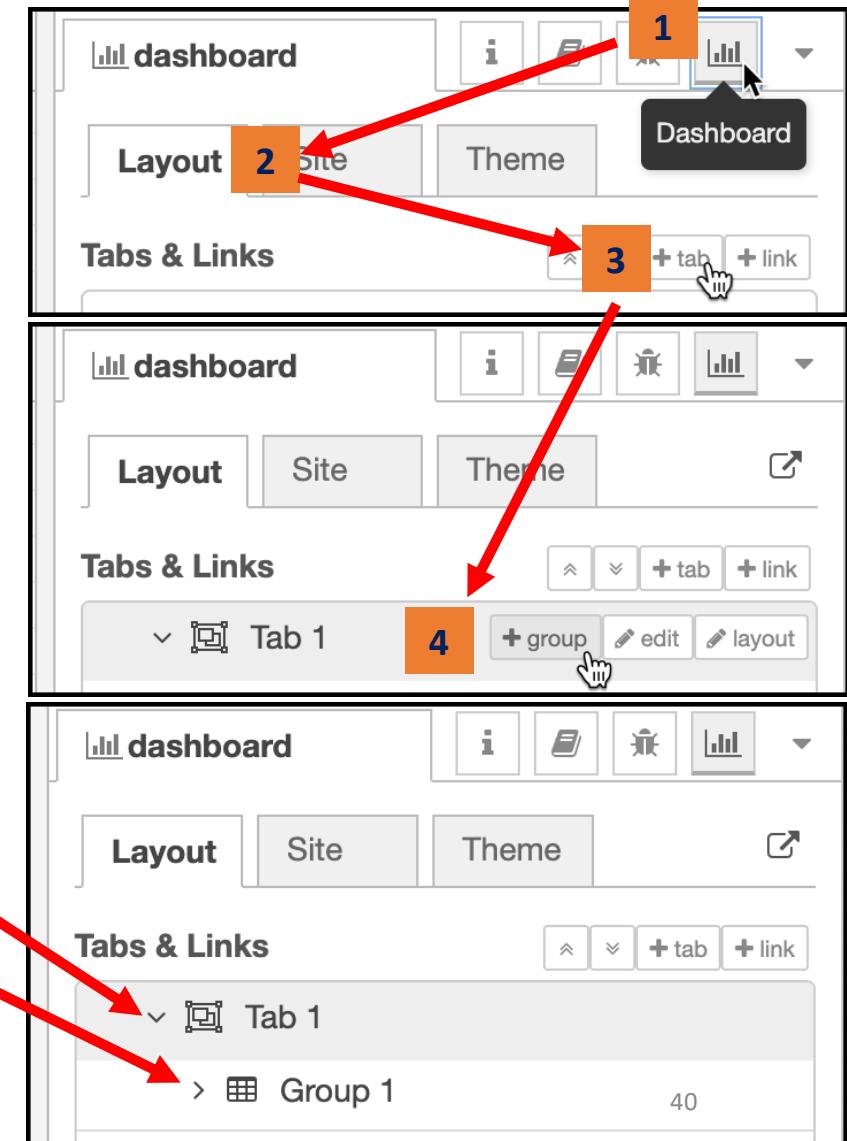
> Every dashboard UI must be in a **tab** field & **group** field.

3 Click **+tab** once to create **Tab 1** field.

4 Click **+group** once to create **Group 1** field.

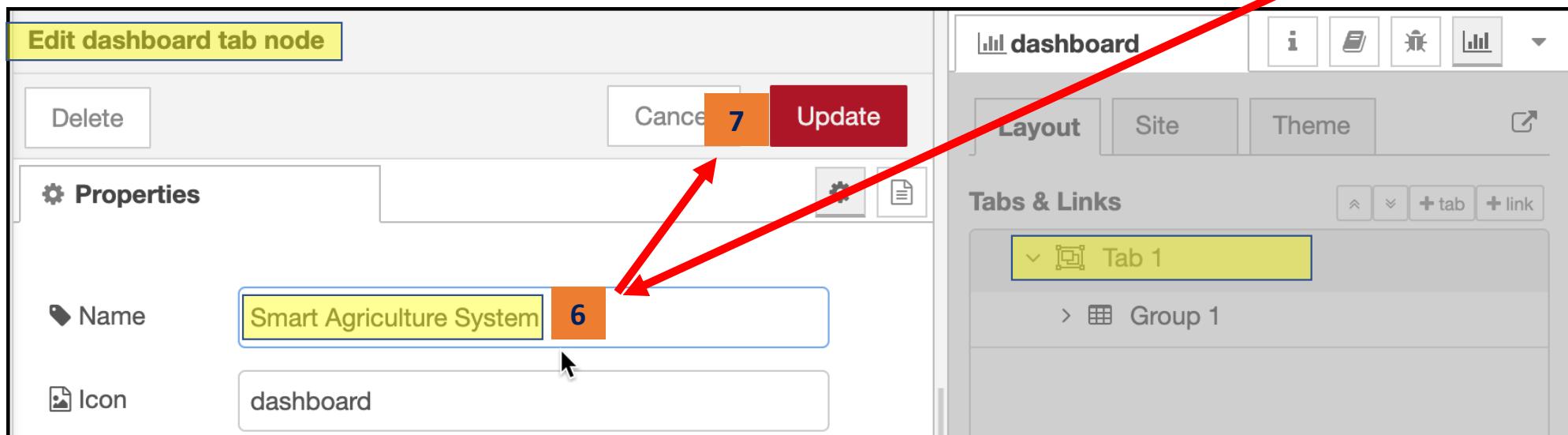
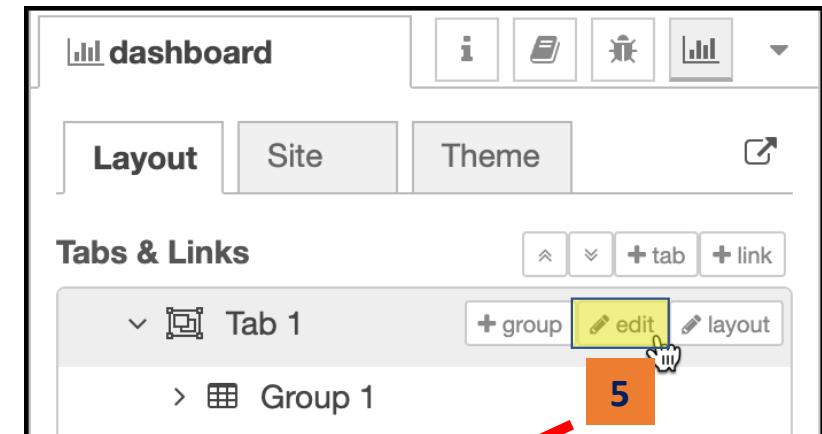
> Next step is to rename **Tab 1** & **Group 2** that reflect the project.

v1(2)-mar-21



## Node-RED: e. Setting the Layout.

- 5 Click **edit** link at **Tab 1** field to rename the **Tab 1**.
- 6 Change **Tab 1** to **Smart Agriculture System**.
- 7 Click **Update** upon completion.

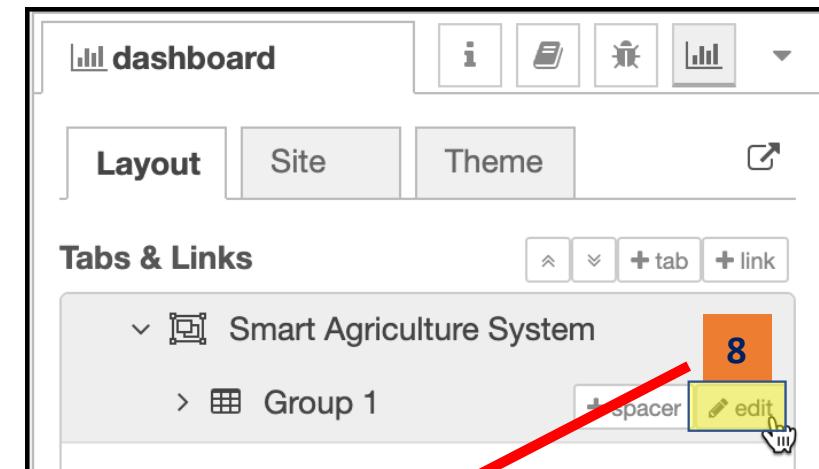


## Node-RED: e. Setting the Layout

8 Click **edit** link at **Group 1** field.

9 Change **Group 1** to **Plot #1**.

10 Click **Update** upon completion.



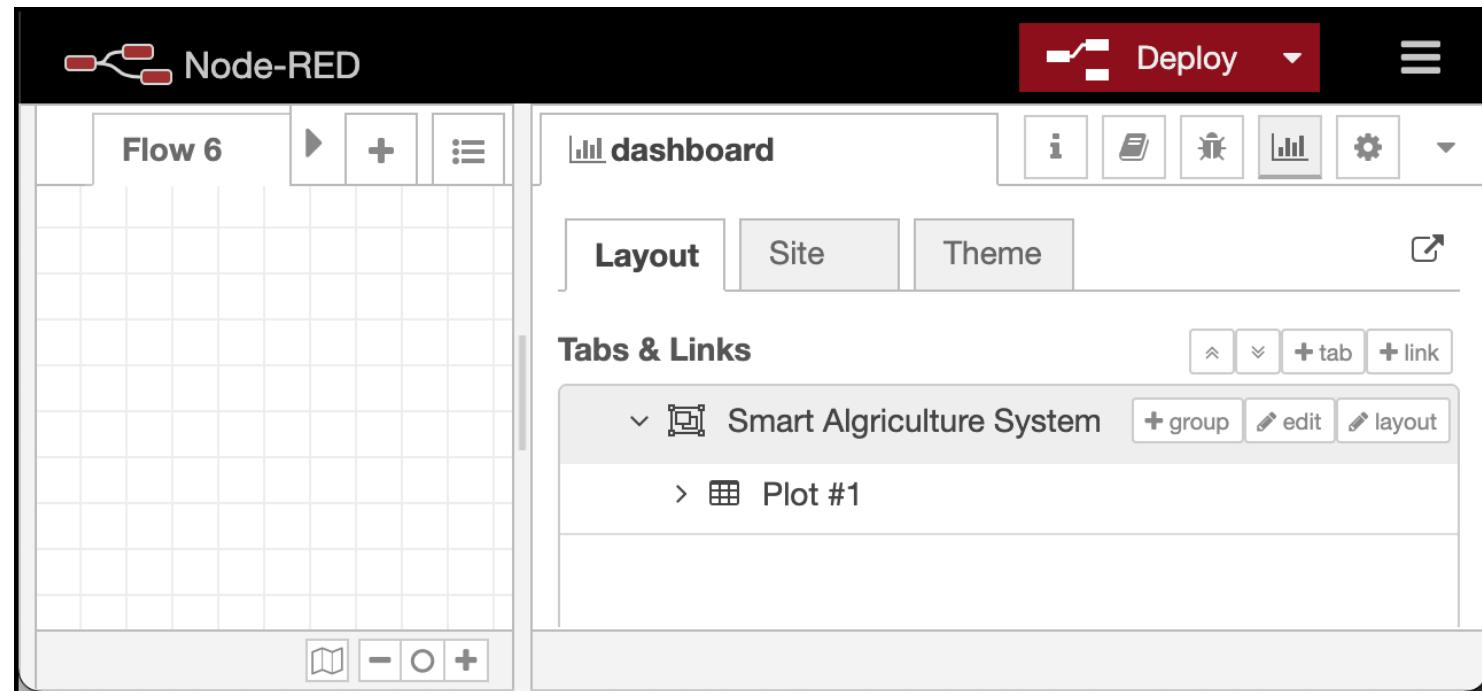
The screenshot shows the 'Edit dashboard group node' dialog box. It has a 'Delete' button, a 'Cancel' button, and a large red 'Update' button. On the left, there's a 'Properties' section with a gear icon. Below it, there are fields for 'Name' (containing 'Plot #1') and 'Tab' (set to 'Smart Agriculture System'). A red arrow labeled '9' points to the 'Name' field, and another red arrow labeled '10' points to the 'Update' button.

## Node-RED: e. Setting the Layout.

> If you have multiple sensors attached at different **Plot**, just add **+group** and name it according the **Plot** number.

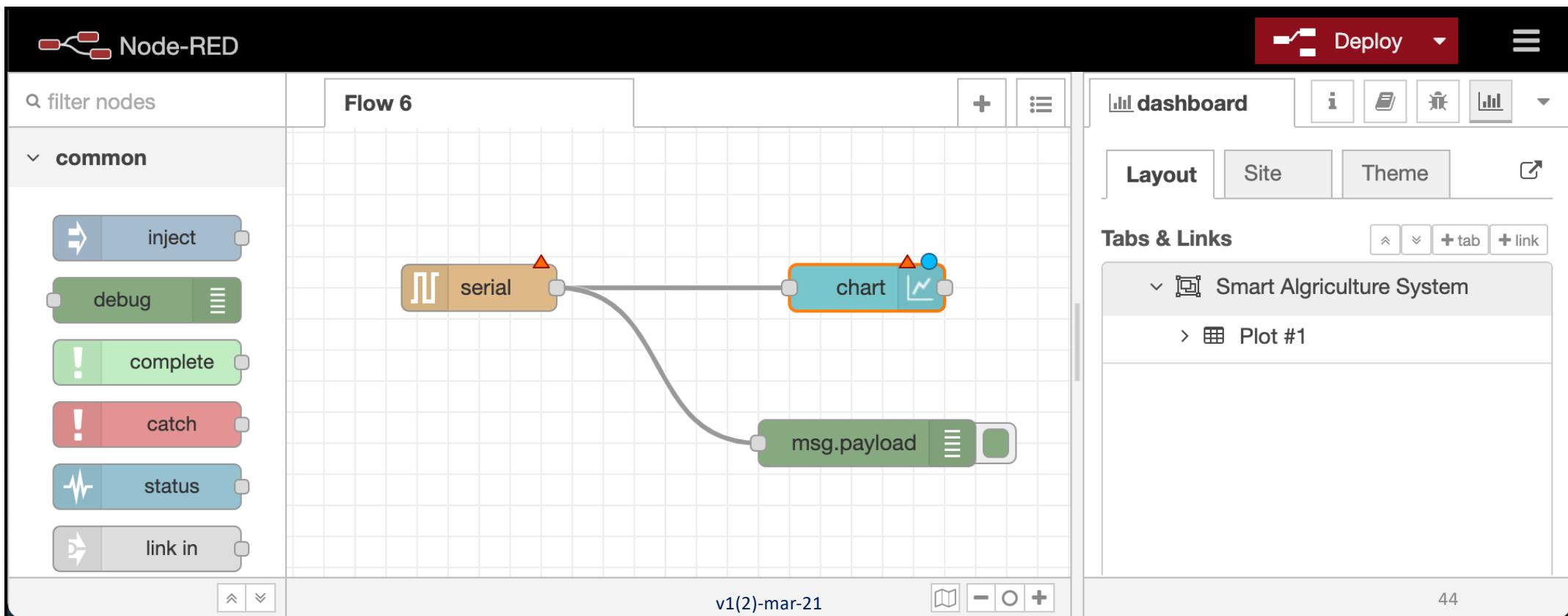
> To delete the layout, go to **edit** section and click **Delete**.

> Don't forget to click **Deploy** after every activity, otherwise, your will lost your work.



## Node-RED: f. Workspace Setup.

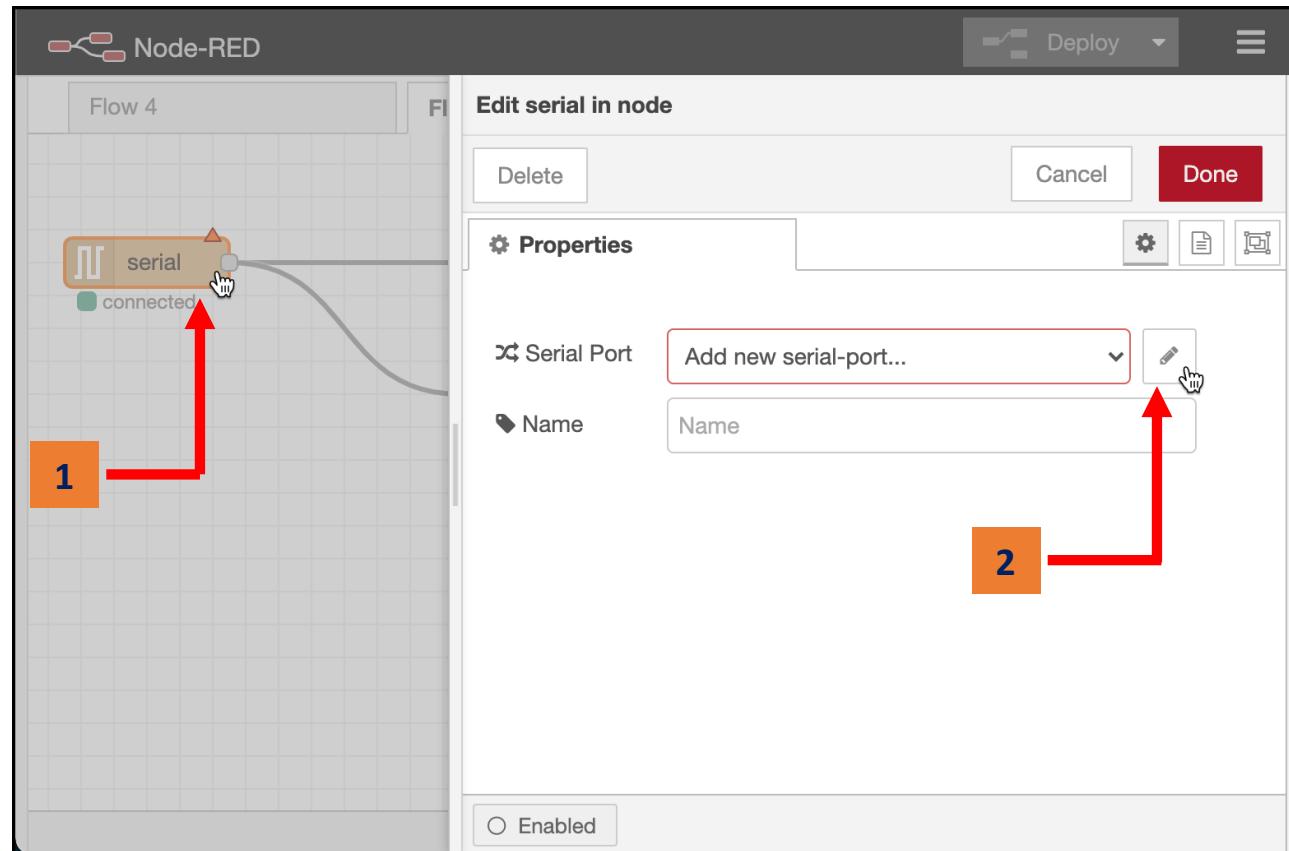
> Next process is to setup the **Serial in** node & assign the properties to **Chart** node.



## Node-RED: f. Serial in Config.

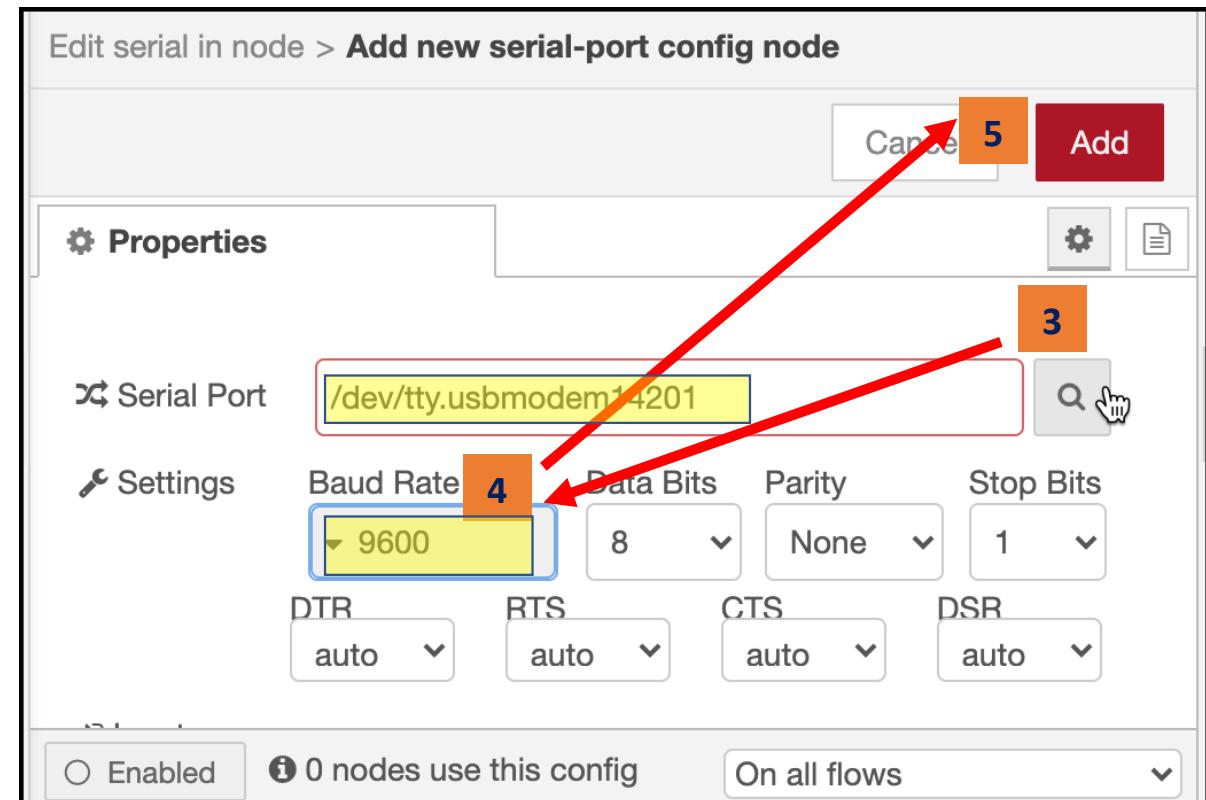
- 1 Double click serial in node.
- 2 Click the pencil icon to add new serial-port config node.

>The port number is the same with the microcontroller's (refer to **Arduino IDE or Device Manager>Port**)



## Node-RED: f. Serial in Config.

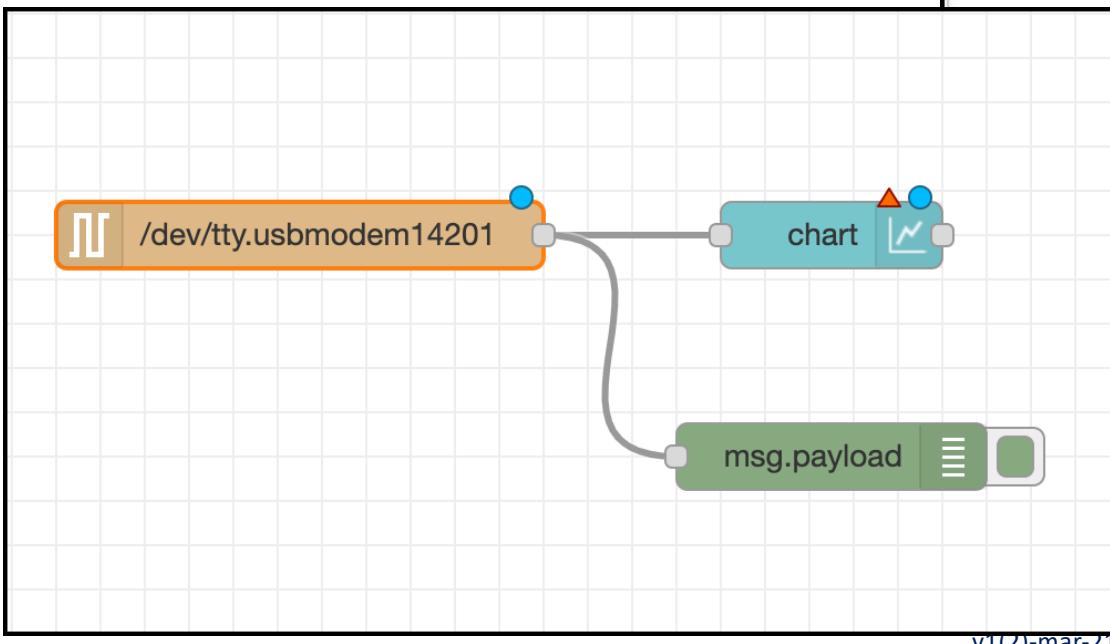
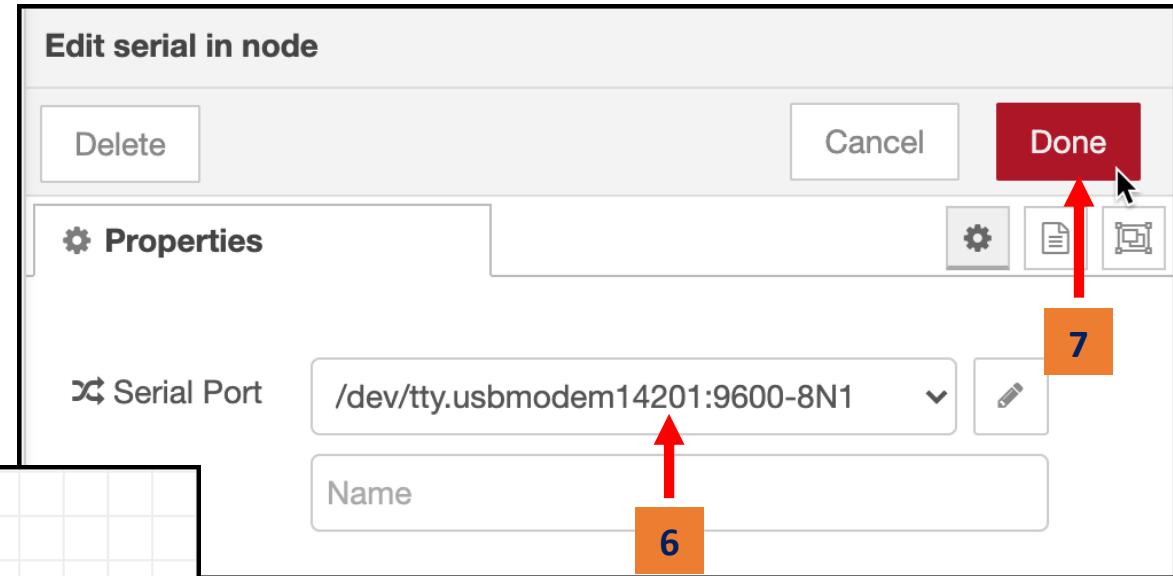
- 3 Click the browse port icon & select the correct serial port. Make sure the microcontroller is connected to your system.
- 4 Change the **Baud Rate** accordingly. Refer to **Slide 5 line#5** **Serial.begin(9600)**.
- 5 Click **Add** & you will be diverted to previous page



## Node-RED: f. Serial in Config.

6 Confirm the setting? If not, click the **pencil icon** to edit.

7 Click **Done** upon completion.



## Node-RED: g. Chart Config.

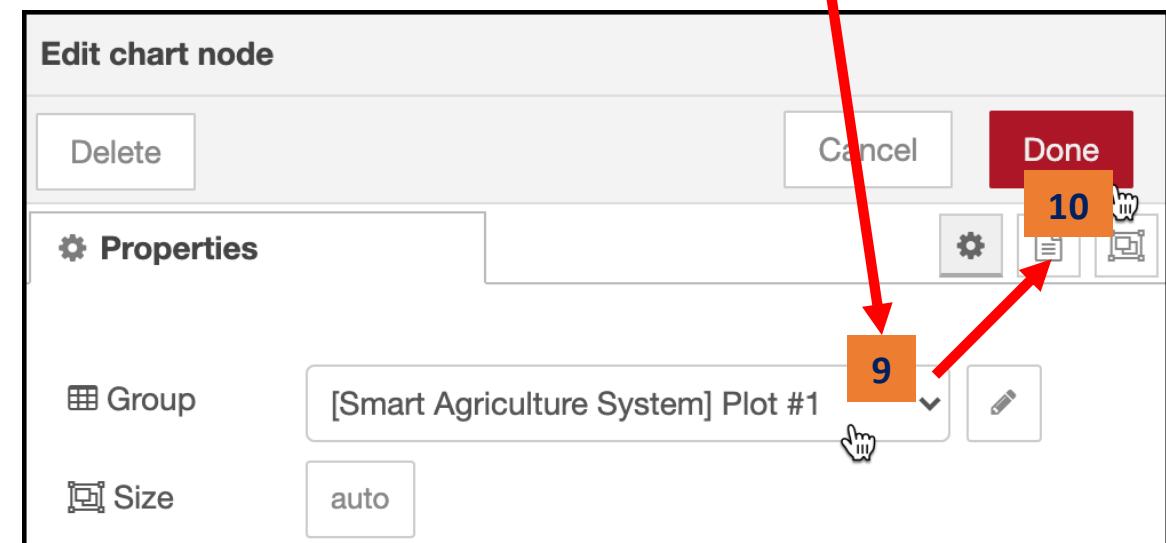
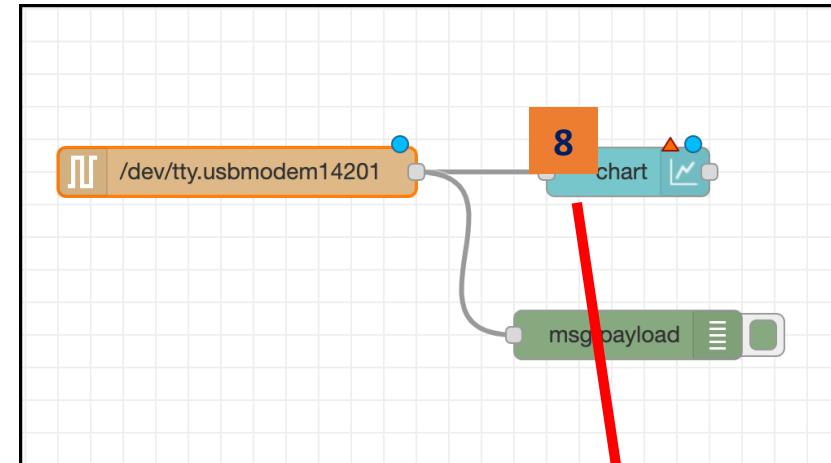
> Next step is to set up Chart's properties.

8 Double click Chart node.

9 Select [Smart Agriculture System] Plot #1.

10 Click Done upon completion.

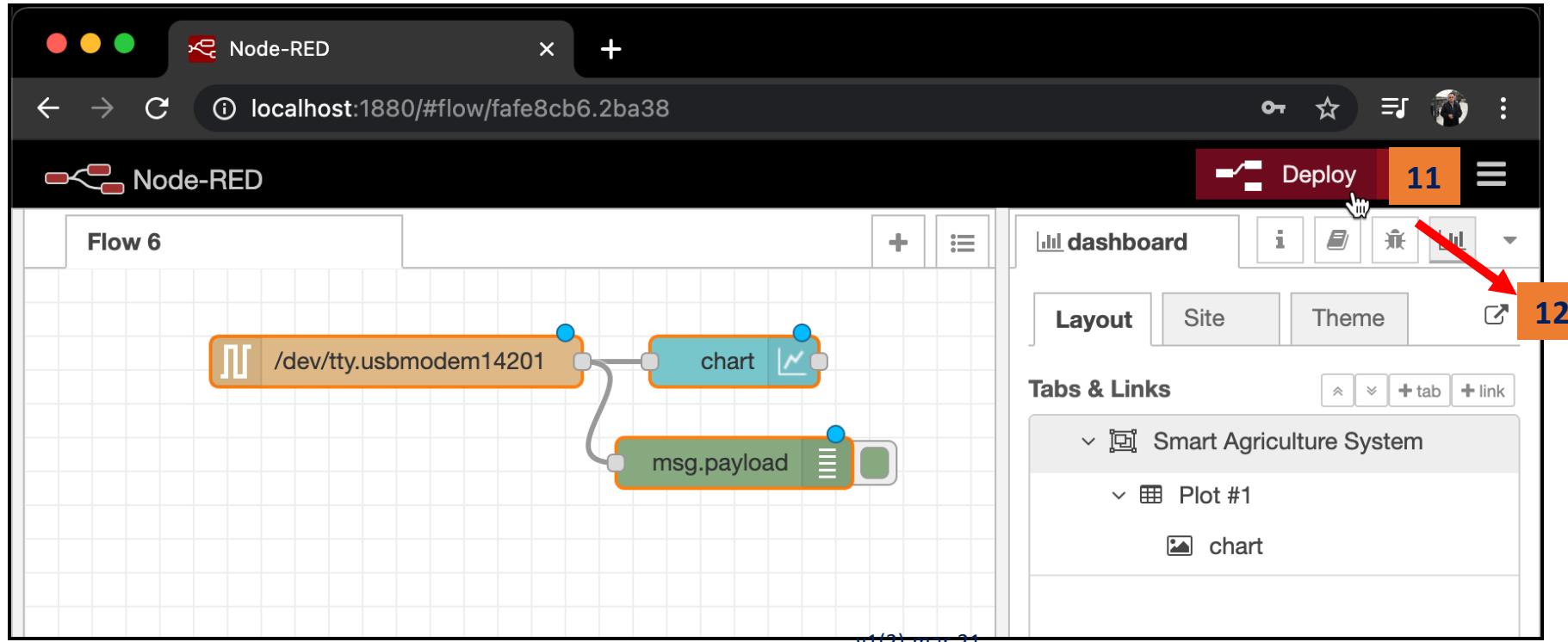
**Note:** Total element in group will increase if you have created series of dashboard tasks.



## Node-RED: h. Prepare to Execute.

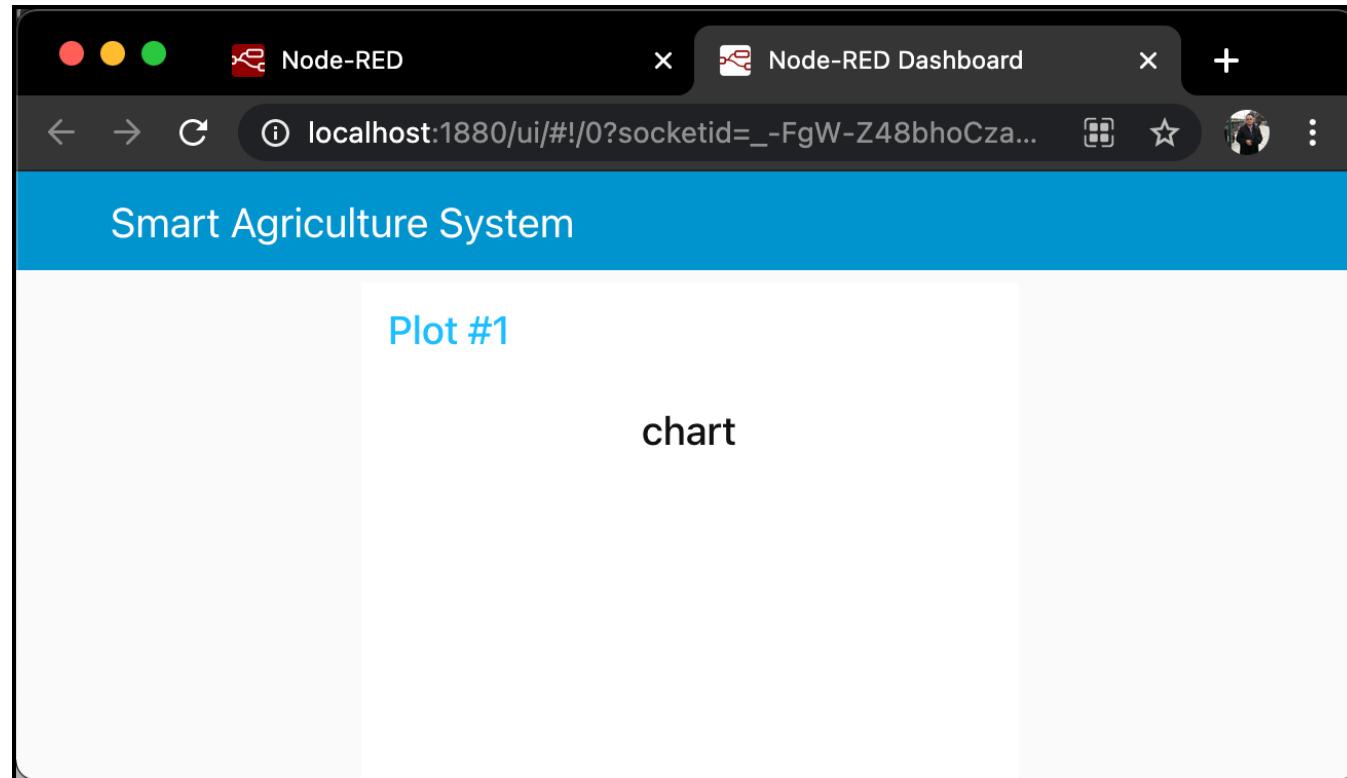
11 Click Deploy to compile the flow's update, setup and config.

12 Click  to view the UI interface aka dashboard. (<http://localhost:1880/ui/>)



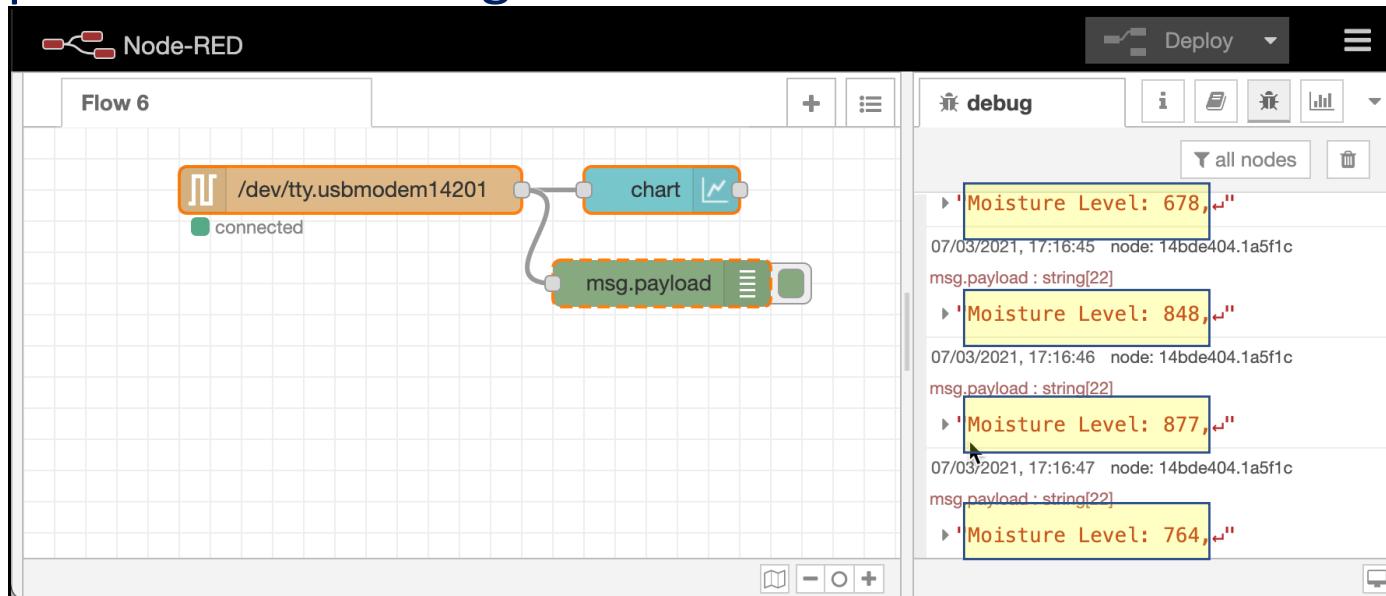
## **Node-RED: i. The Output & Troubleshoot.**

> No output produced. Go back to **nodeRED workspace editor** & click debug  message to see the output tapped at **debug node**.



## Node-RED: i. The Output & Troubleshoot.

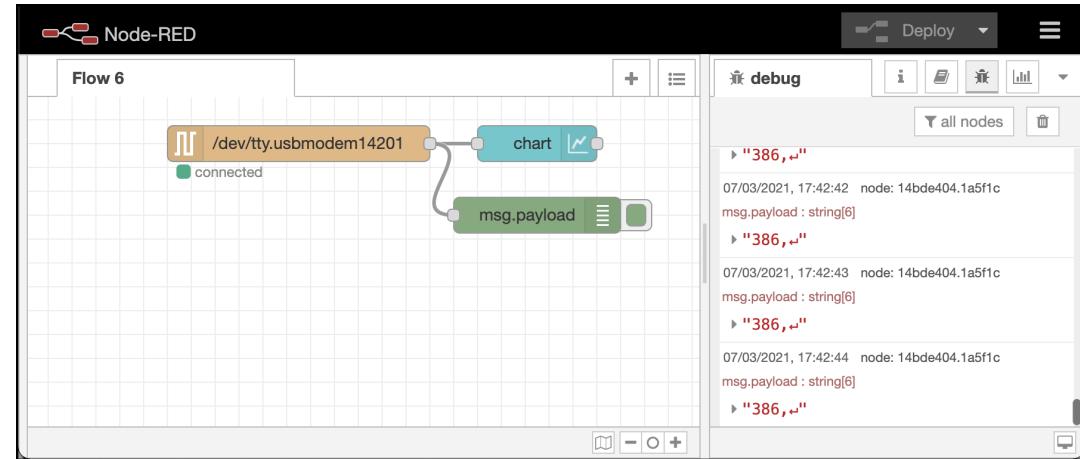
> Need to omit the '**Moisture Level:**' text. Remark line number 13 (refer slide 5).  
Upload the sketch again.



**Note:** You need to disconnect the board from nodeRED first, by double click the **serial** in node text & change port parameter  & click **Deploy**. When connect to nodeRED, close the **Serial Monitor**.

## **Node-RED: i. The Output & Troubleshoot.**

- > Output at Serial Monitor shows the reading from the sensor.
  - > Similar with at nodeRED, don't forget to exit Serial Monitor before triggering the nodeRED serial port again. Only one application per serial port.

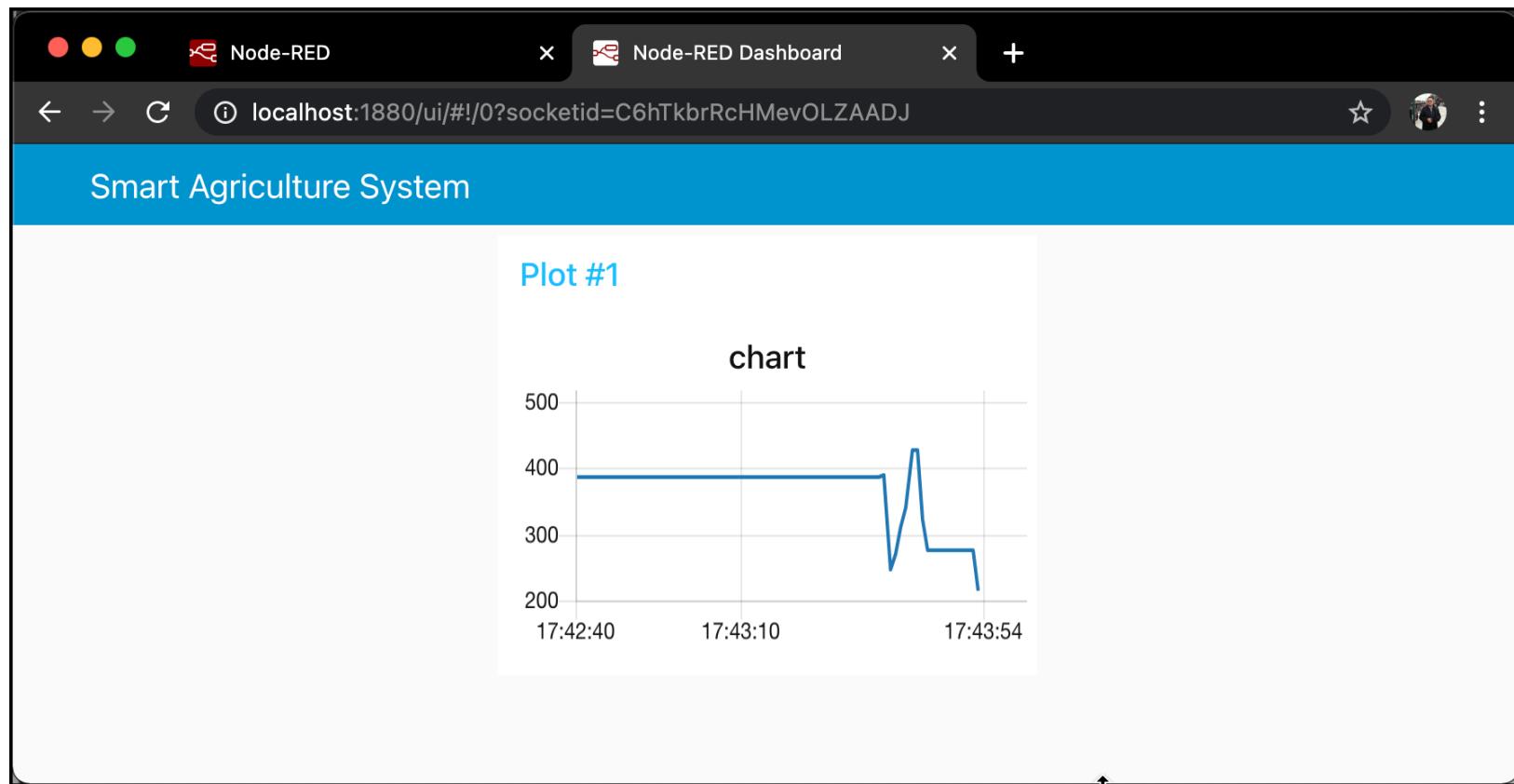


**Note:** You need to disconnect the board from nodeRED by double click the serial in node text & change port parameter  & click Deploy. When connect to nodeRED, close the **Serial Monitor**.

 Serial Port Add new serial-port... 

## Node-RED: i. The Output & Troubleshoot.

> The final output.



## EXERCISE:

Add one more sensor, modify the function node and also the sketch.

Answer:

```
[{"id":"bac65c23.74b21","type":"tab","label":"Flow
2","disabled":false,"info":""}, {"id":"1c057498.f326ab","type":"debug","z":"bac65c23.74b21","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","statusVal":"","statusType":"auto","x":470,"y":100,"wires":[]}, {"id":"70291b26.ad9e44","type":"serial
in","z":"bac65c23.74b21","name":"","serial": "a674939a.de6d1","x":190,"y":220,"wires": [{"8289256.014bfd8","5285227e.00726c"}]}, {"id":"5285227e.00726c","type":"function","z":"bac65c23.74b21","name":"Humid","func":"var output = msg.payload.split(\",\");\nvar velimp = parseInt(output[0]);\nvar angle = parseInt(output[1]);\nvar msg = {payload : velimp};\nvar msg2 = {payload : angle};\nreturn msg;\n", "outputs":1,"noerr":0,"initialize":"","finalize":"","x":330,"y":140,"wires": [{"1c057498.f326ab","9d0bbc0b.d35ba"}]}, {"id":"8289256.014bfd8","type":"debug","z":"bac65c23.74b21","name":"Temp","func":"var output = msg.payload.split(\",\");\nvar velimp = parseInt(output[0]);\nvar angle = parseInt(output[1]);\nvar msg2 = {payload : velimp};\nvar msg = {payload : angle};\nreturn msg;\n", "outputs":1,"noerr":0,"initialize":"","finalize":"","x":310,"y":300,"wires": [{"909b5194.dcf8e","dfe43fa5.09c56","88d47994.eee278"}]}, {"id":"909b5194.dcf8e","type":"debug","z":"bac65c23.74b21","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","statusVal":"","statusType":"auto","x":490,"y":280,"wires":[]}, {"id":"9d0bbc0b.d35ba","type":"ui_gauge","z":"bac65c23.74b21","name": "Humidity at CL412","group": "9f246d07.8e352","order":1,"width":0,"height":0,"gtype": "gage","title": "", "label": "units", "format": "{{value}}%", "min": 0, "max": 100, "colors": ["#00b500", "#e6e600", "#ca3838"], "seg1": "", "seg2": "", "x": 490, "y": 200, "wires": []}, {"id": "dfe43fa5.09c56", "type": "ui_chart", "z": ".bac65c23.74b21", "name": "Temperature at CL412", "group": "af20372d.328a38", "order": 1, "width": 0, "height": 0, "label": "Temp (C)", "chartType": "line", "legend": false, "format": "HH:mm:ss", "interpolate": "linear", "nodata": "", "dot": false, "ymin": "", "ymax": "", "removeOlder": 1, "removeOlderPoints": "", "removeOlderUnit": "3600", "cutout": 0, "useOneColor": false, "useUTC": false, "colors": ["#1f77b4", "#aec7e8", "#ff7f0e", "#2ca02c", "#98df8a", "#d62728", "#ff9896", "#9467bd", "#5b9d54"], "useOldStyle": false, "outputs": 1, "x": 520, "y": 320, "wires": []}, {"id": "88d47994.eee278", "type": "ui_text", "z": ".bac65c23.74b21", "group": "af20372d.328a38", "order": 1, "width": 0, "height": 0, "name": "", "label": "text", "format": "{msg.payload}", "layout": "row-spread", "x": 480, "y": 440, "wires": []}, {"id": "a674939a.de6d1", "type": "serial-port", "z": "", "serialport": "/dev/tty.usbmodem14201", "serialbaud": "9600", "databits": "8", "parity": "none", "stopbits": "1", "waitfor": "", "dtr": "none", "rts": "none", "cts": "none", "dsr": "none", "newline": "\n", "bin": "false", "out": "char", "addchar": "", "responsetimeout": "10000"}, {"id": "9f246d07.8e352", "type": "ui_group", "z": "", "name": "Humidity", "tab": "ff22fd9e.4e5c3", "order": 1, "disp": true, "width": 4, "collapse": false}, {"id": "af20372d.328a38", "type": "ui_group", "z": "", "name": "Temperature", "tab": "ff22fd9e.4e5c3", "order": 2, "disp": true, "width": 4, "collapse": false}, {"id": "ff22fd9e.4e5c3", "type": "ui_tab", "z": "", "name": "lab2", "icon": "dashboard", "disabled": false, "hidden": false}]
```

nodeRED

```
/Variables
String data1, data2;
int d1, d2;

void setup()
{
    Serial.begin(9600);
}

void loop()
{
    //Read data from port 0 & 2, and store it to variables d1, d2 in integer mode
    d1 = analogRead(0);
    d2 = analogRead(2);
    data1 = String(d1);
    data2 = String(d2);
    //Print d1 and d2 values to serial monitor
    Serial.print(d1);
    Serial.print(" ");
    Serial.print(d2);

    delay(2000); //Delay 2 sec.
}
```

arduino

**END**