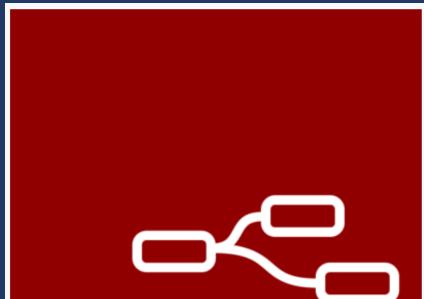


<https://github.com/sworkz-GMI2021/GMI-mar2021>



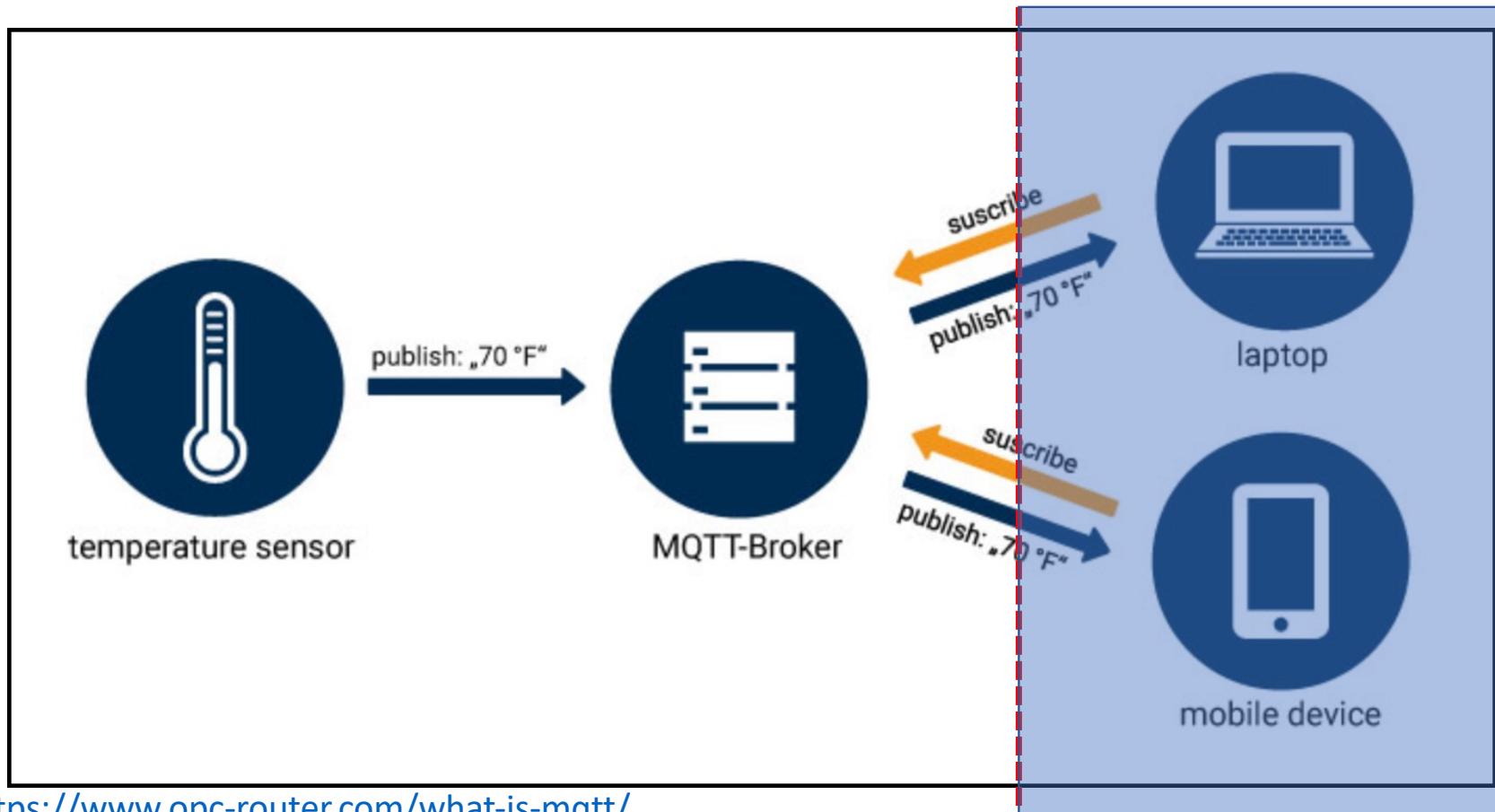
IoT Case Study: Monitoring Server Room Temperature & Humidity Level

“Configuring the MQTT Publish & Subscribe nodes in Node-RED [using ESP32 + I2C LCD + DHT11]”

v1 mar2021

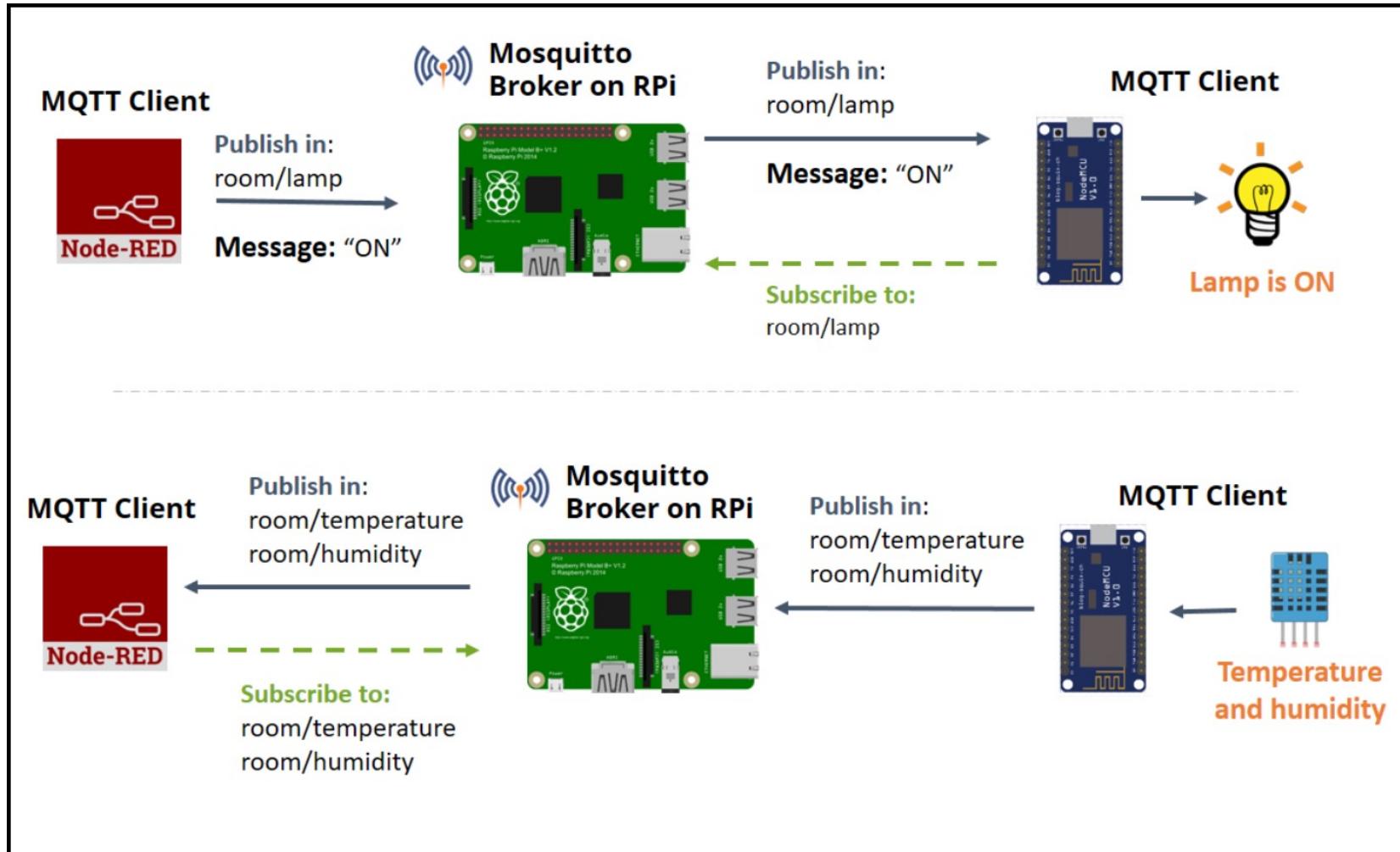
iezan74@gmail.com

Scenario: Display temperature & humidity data on Node-RED using MQTT communication protocol.



IoT Case Study: Monitoring Server Room Temperature & Humidity Level

MQTT



Requirement:

- i. Experience with Node-RED, DHT11 & ESP32 would be an advantage.
- ii. DHT 11 & ESP32 (with its accessories).
- ii. PC/Laptop/Raspberry Pi.

To Do:

A. Hardware Section

- i. Download & Install Arduino IDE
- ii. Install ESP32 board
- iii. Download DHT11, I2C LCD libraries & client library for MQTT messaging – [PubSubClient](#)
- iv. Connect/assemble & test hardware: DHT11 & I2C 16x2 LCD
- v. Download sketch from xxxx & test.

B. Node-RED Section

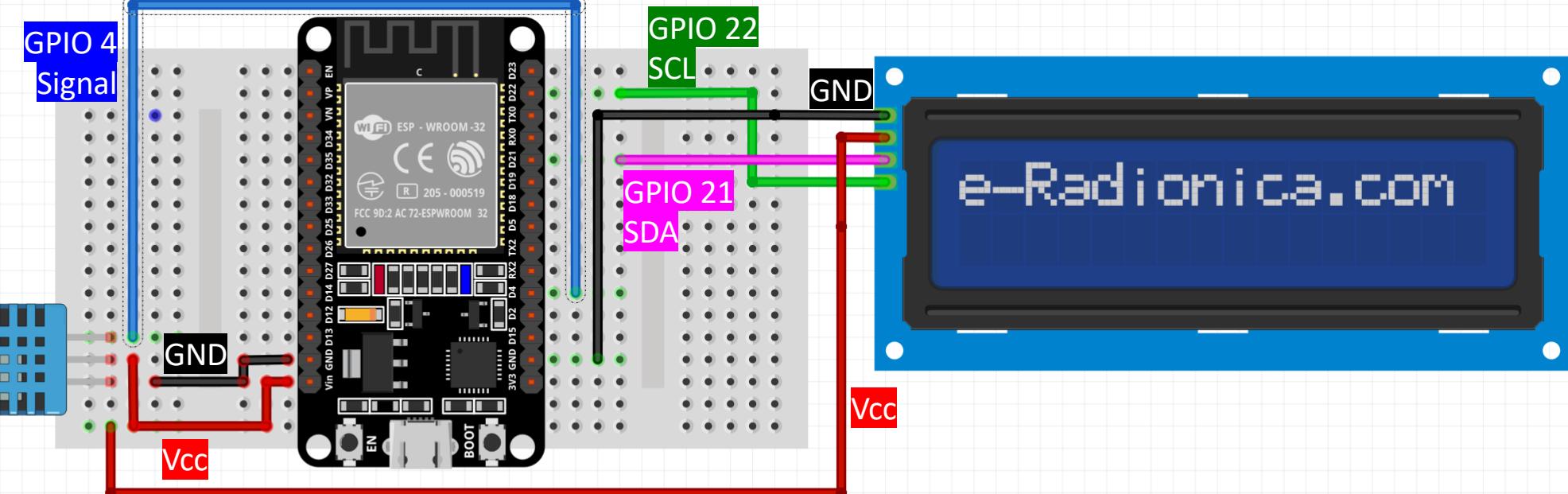
- i. Install Aedes Broker.

C. System Deployment

- i. Part A + Part B.

D. MySQL (if time permits)

Connection



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



Installation.

> Click the following link to download:
www.apachefriends.org/download.html

> Run the .exe file & follow the steps you are prompted with.

B. STARTING XAMPP SERVICE

Starting XAMPP Service.

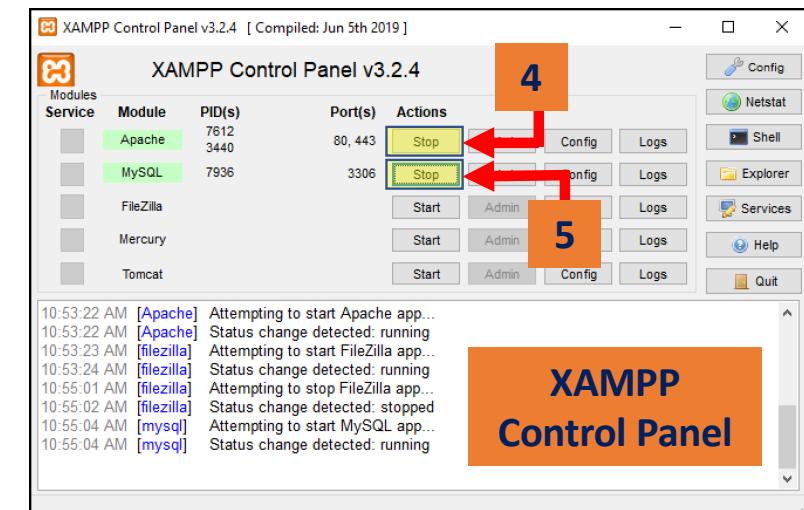
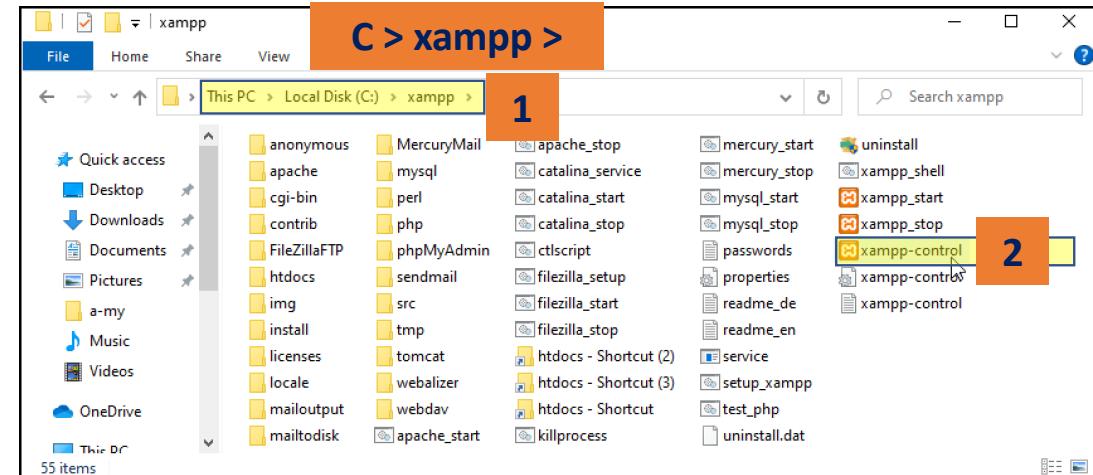
> Default XAMPP location is at

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**.

The screenshot shows a browser window with the URL `http://localhost/phpmyadmin/` in the address bar. A red arrow labeled '7' points to the address bar. The page content is the phpMyAdmin control panel, showing the "General settings" and "Database server" sections. The "Database server" section lists the connection details: 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, and Server charset: UTF-8 Unicode (utf8mb4). A large red arrow points from the bottom right of this panel towards the XAMPP main page.

MPP for Windows 7.3.26

IP on this system! Now you can start using Apache, MariaDB, PHP and other in the [FAQs](#) section or check the [HOW-TO Guides](#) for getting started with PHP

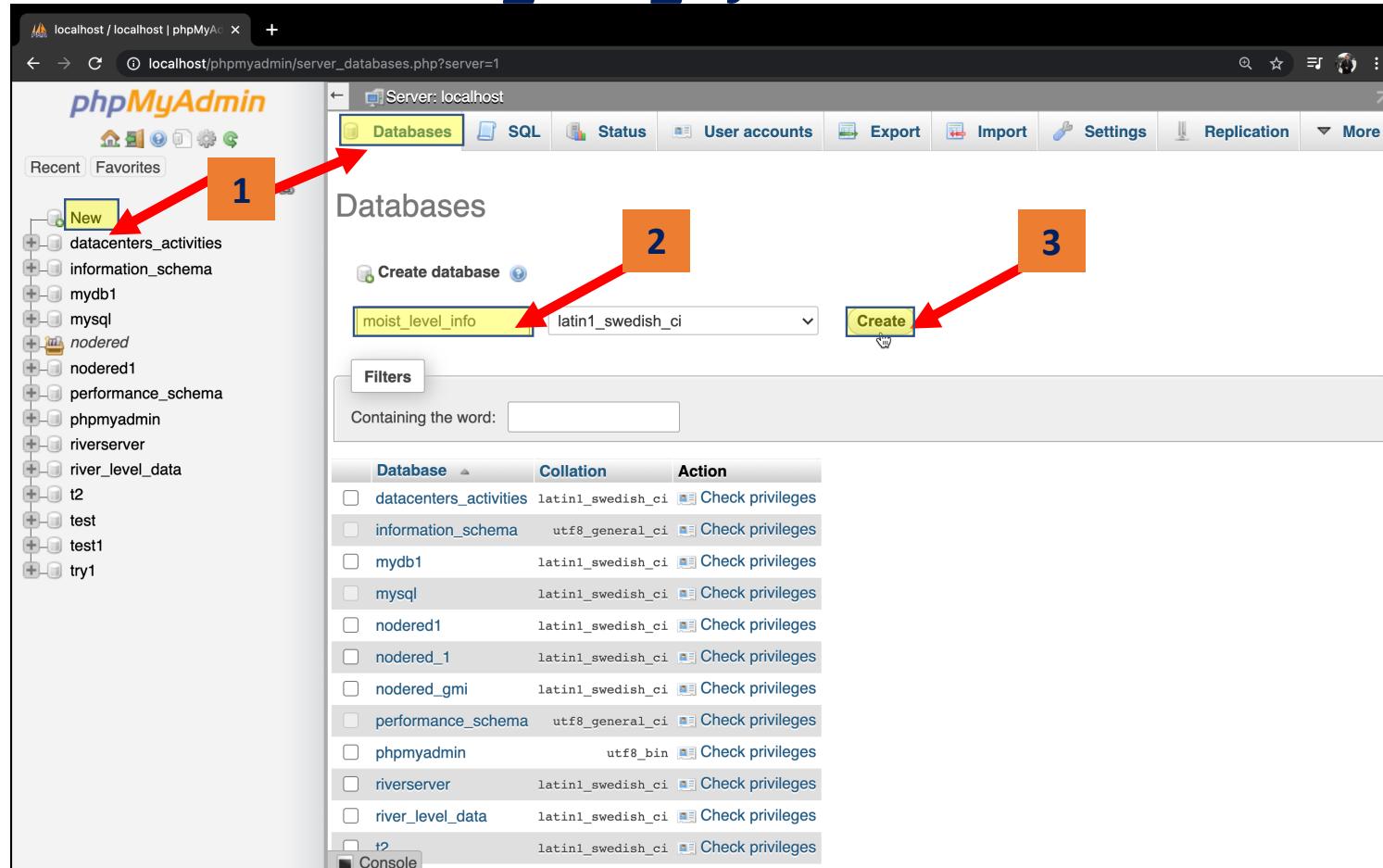
Creating Your Database

> 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: PRIMARY
			A.I.: v **A.I. = Auto Increment
logged	DATETIME	6	Default: CURRENT_TIMESTAMP
moisture_level	VARCHAR	100	

Creating Your Database: Set Database Name

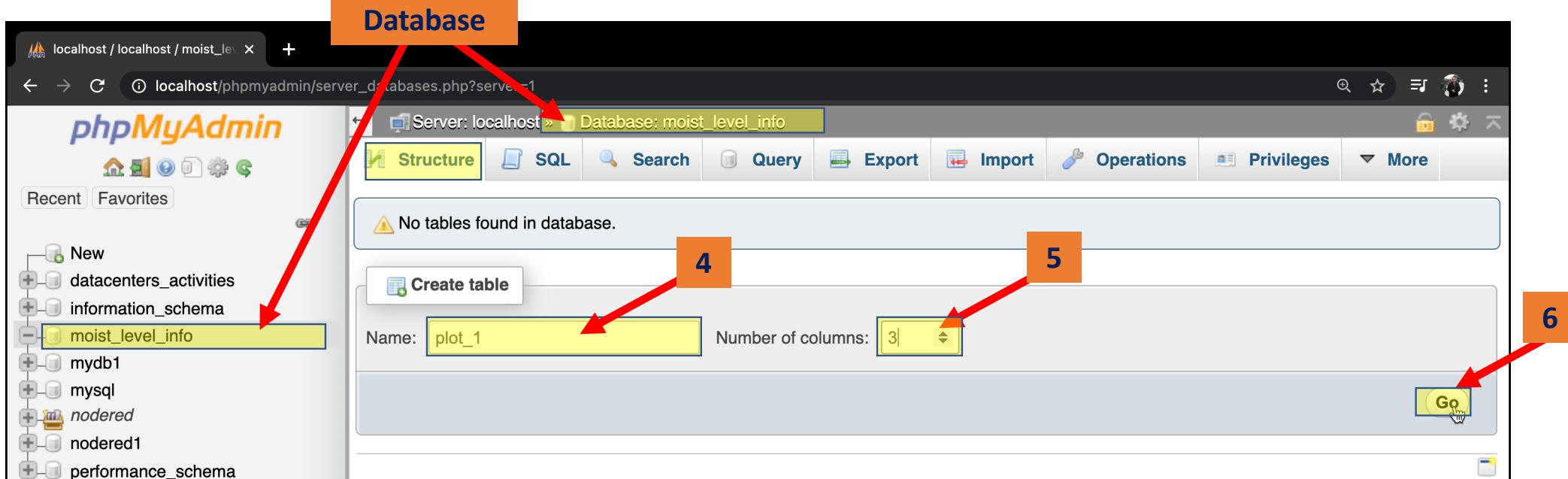
Step 1 >> Database Name: *moist_level_info*



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**

**A.I. = Auto Increment

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
7 id	8 INT	9 15	None				10 PRIMARY
13 logs	14 DATETIME		CURRENT_TIME	15			
16 reading	17 VARCHAR	18 100					

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

11 Go 12 Save 13 Preview SQL 14 15 16 17 18 19

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

Add 1 column(s) after reading Go

Print Propose table structure Track table Move columns Normalize

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# VI-Mai-Z1

Creating Your Database: Seeing Your Data

Step 5 >> Browse tab

The screenshot shows the phpMyAdmin interface with the 'Browse' tab selected. The left sidebar lists databases and tables, with 'moist_level_info' expanded to show 'plot_1'. The main area displays the results of the SQL query `SELECT * FROM `plot_1``. The results table has columns `id`, `logs`, and `reading`. A red arrow points from an orange box containing the text 'Empty since no data stored' to the results table.

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0021 seconds.)

```
SELECT * FROM `plot_1`
```

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

<code>id</code>	<code>logs</code>	<code>reading</code>
-----------------	-------------------	----------------------

Query results operations

Create view

Bookmark this SQL query

Label: Let every user access this bookmark

Bookmark this SQL query

Console

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 a MySQL database named 'moist_level_info'. The 'plot_1' table is selected. The 'Insert' tab is active. The 'reading' column is highlighted with a yellow background and a red arrow pointing to it. The value '999' is typed into this field. A blue box with the text 'Just type any value at this field only' is overlaid on the interface. The 'logs' column has a function set to 'CURRENT_TIMESTAMP'. The 'id' column is of type int(15) and has a function set to 'NULL'. The 'Value' column for 'reading' is empty. The 'logs' column has a function set to 'CURRENT_TIMESTAMP'. The 'id' column is of type int(15) and has a function set to 'NULL'. The 'Value' column for 'logs' is empty. The 'reading' column has a function set to 'NULL'. The 'Value' column for 'reading' is empty. At the bottom, there are buttons for 'Insert as new row' and 'and then', along with links to 'Go back to previous page', 'Console', 'Preview SQL', and 'Reset'.

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 main area, a green message box displays: '1 row inserted.' and 'Inserted row id: 1'. Below this, the SQL query is shown: `INSERT INTO `plot_1` (`id`, `logs`, `reading`) VALUES (NULL, CURRENT_TIMESTAMP, '999');`. To the right, a 'Columns' panel lists the table structure: `id`, `logs`, and `reading`. At the bottom, there are several buttons: `SELECT *`, `SELECT`, `INSERT`, `UPDATE`, `DELETE`, `Clear`, `Format`, and `Get auto-saved query`. There are also checkboxes for `Bind parameters` and `Bookmark this SQL 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.

Your first data

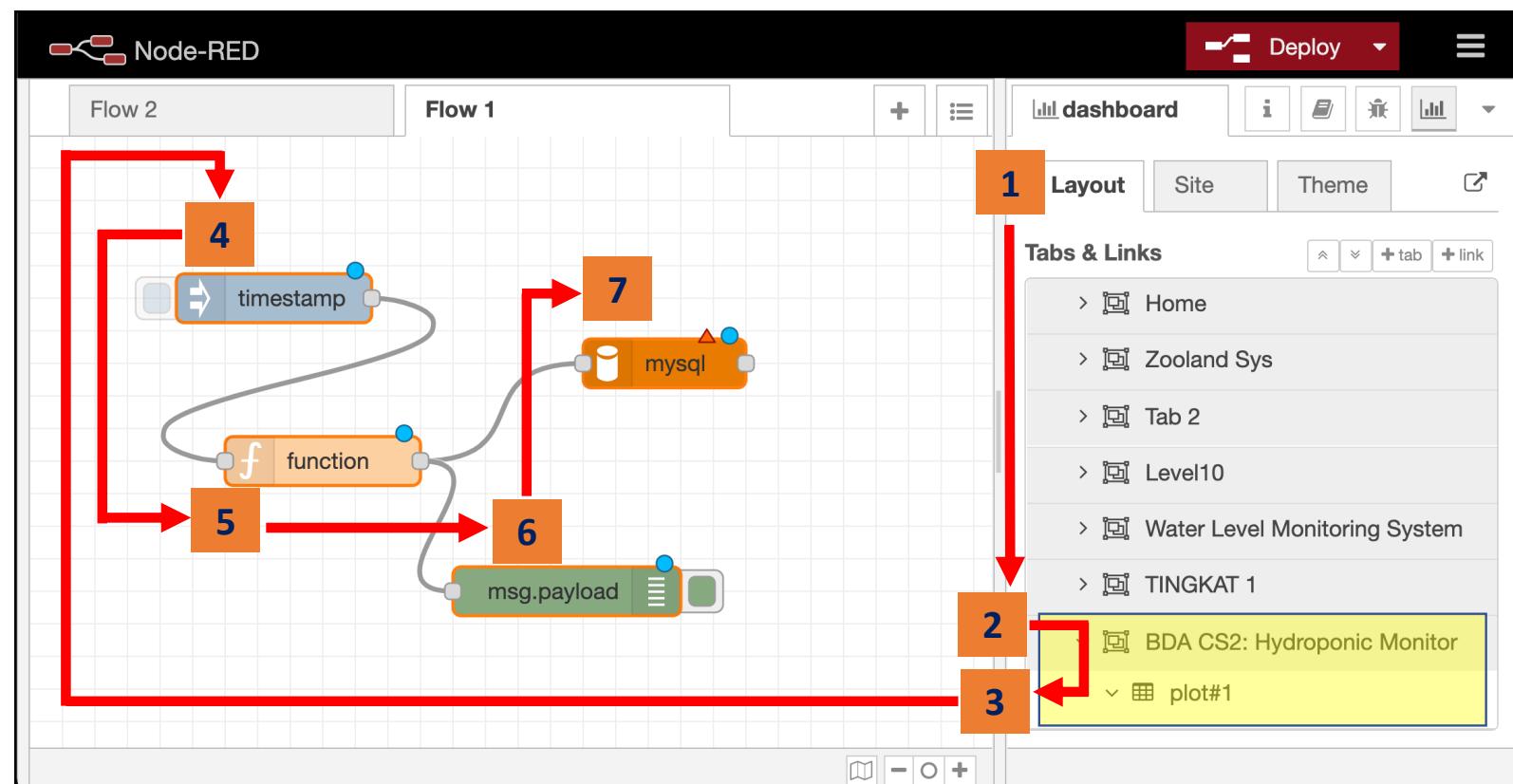
	id	logs	reading
<input type="checkbox"/>	1	2021-03-11 00:02:54	999

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

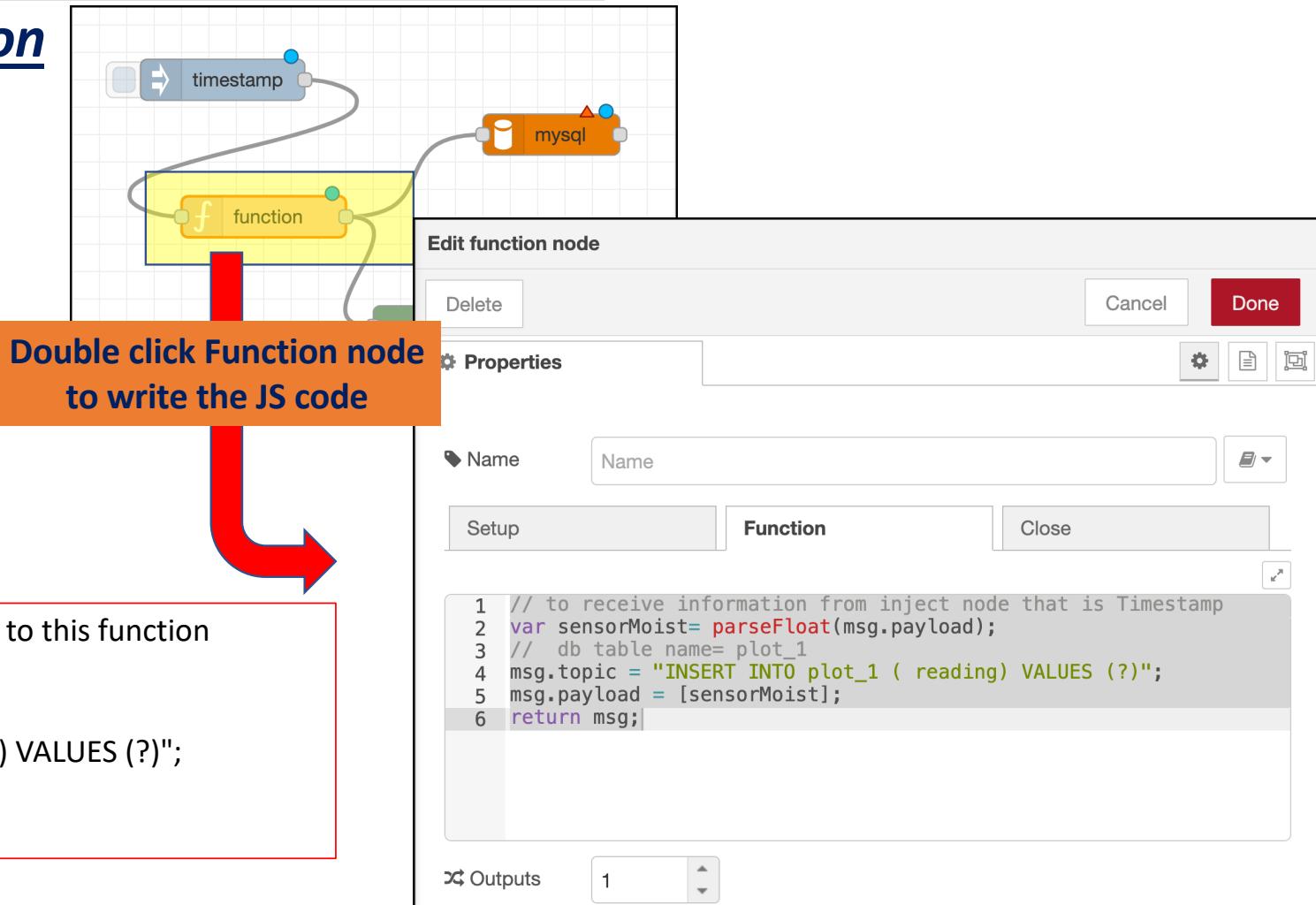
Step 2 >> Writing Function

> 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;
```



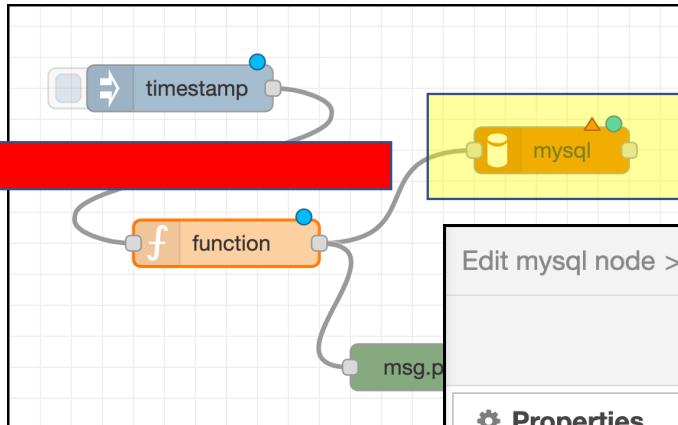
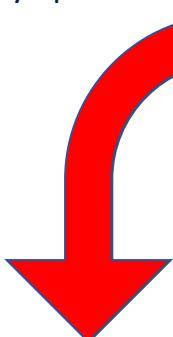
nodeRED + MySQL: Manually Inject into Database

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

Click Add – to go back to previous page

Edit mysql node

Delete

Cancel Done

Properties

Database: Add new MySQLdatabase...

Name: Name

v1-mar-21

Edit mysql node > Add new MySQLdatabase config node

Cancel Add

Properties

Host: 127.0.0.1

Port: 3306

User: root **User: root**

Password:

Database: moist_level_info **Database name: moist_level_info**

Timezone:

Enabled: 0 nodes use this config On all flows

nodeRED + MySQL: Manually Inject into Database

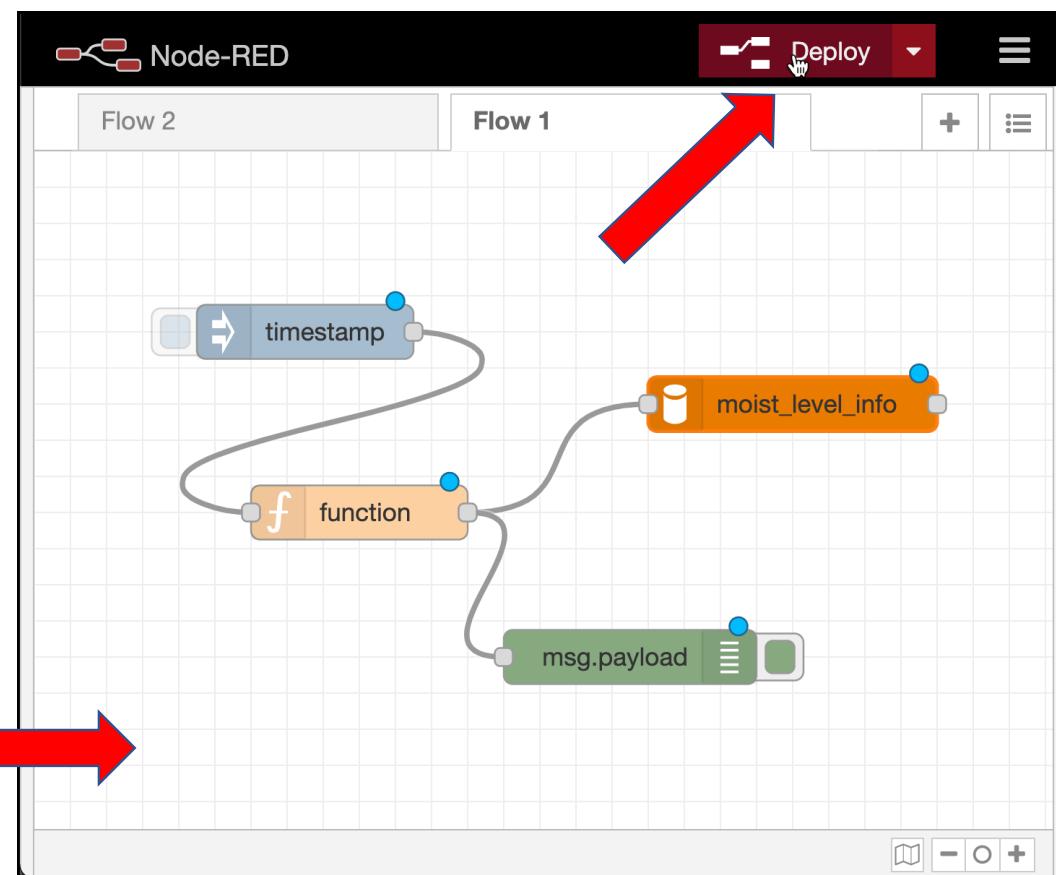
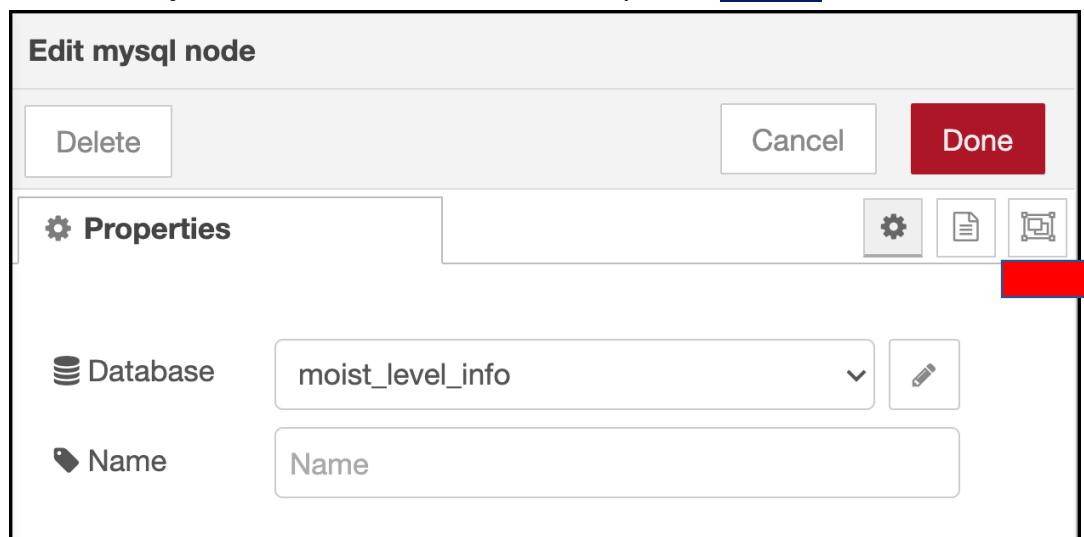
Step 4 >> Time to Test

> 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.

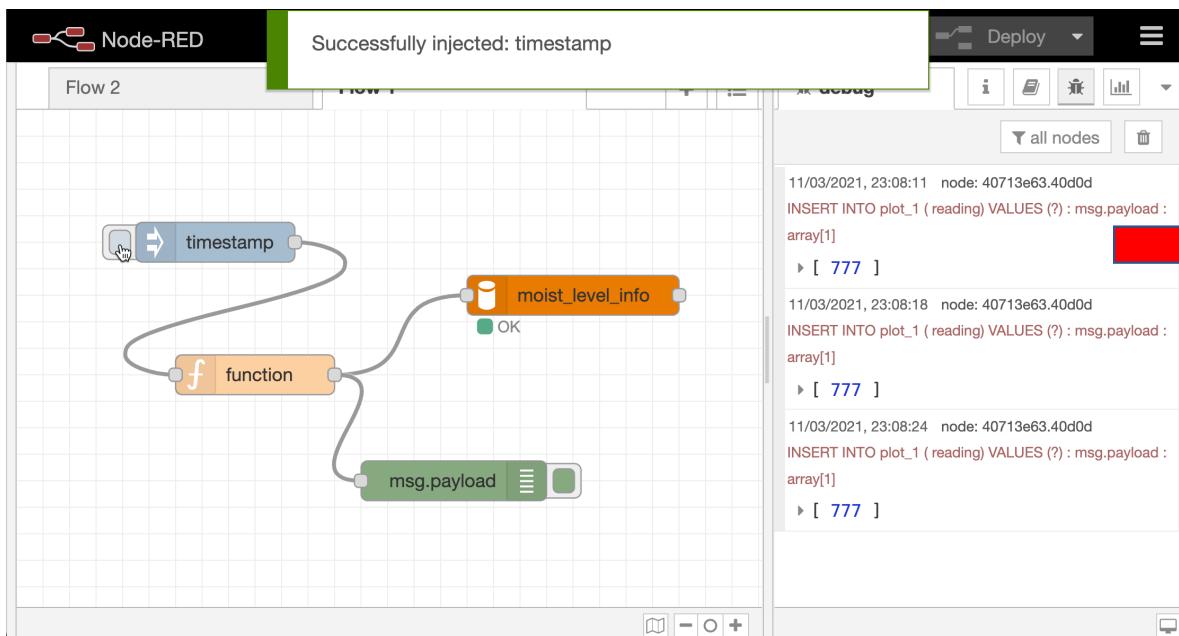
> Click the inject button & observe the output at Debug section.



>>DONE
>>Deploy

nodeRED + MySQL: a. Manually Inject into Database Step 4 >> The Results

<http://localhost:1880/>



<http://localhost/phpmyadmin>

The screenshot shows the phpMyAdmin interface connected to the "moist_level_info" database. The "plot_1" table is selected. A red arrow points from the Node-RED interface to this table.

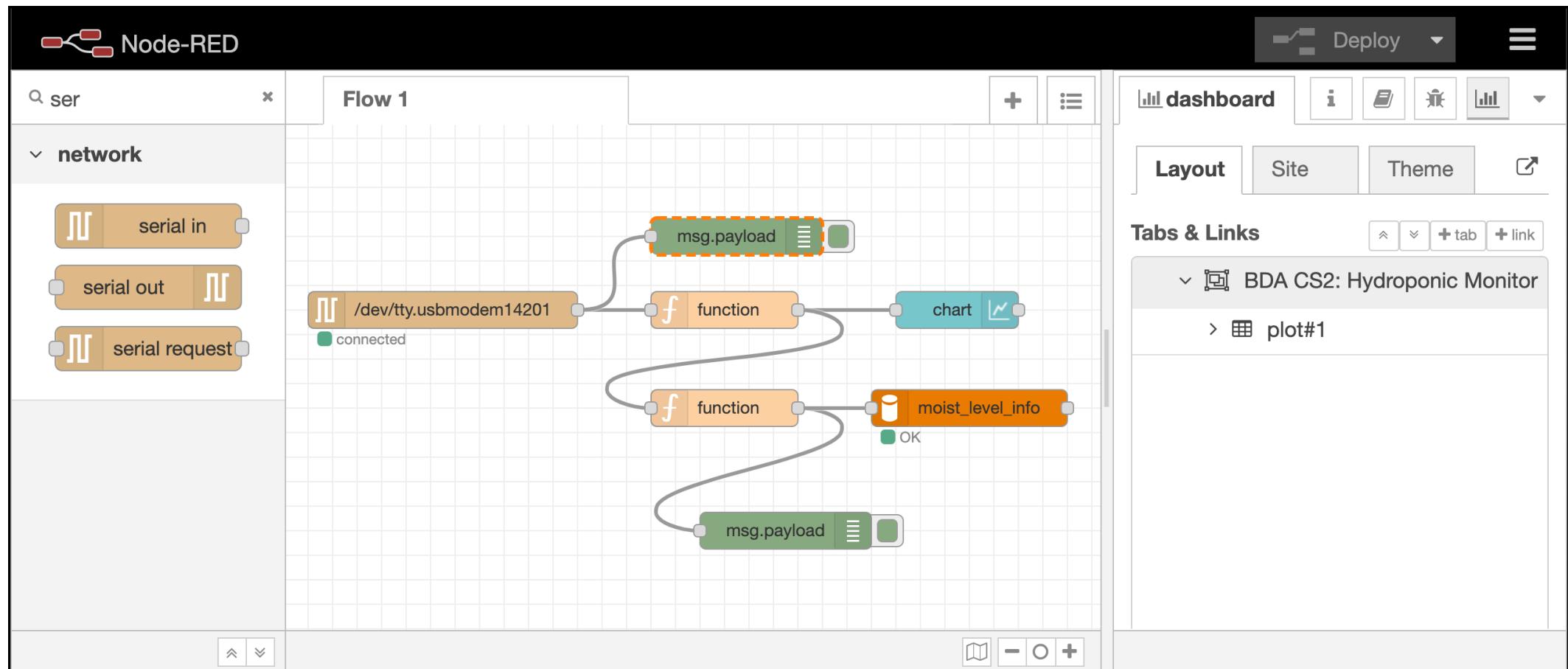
The table structure is:

	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

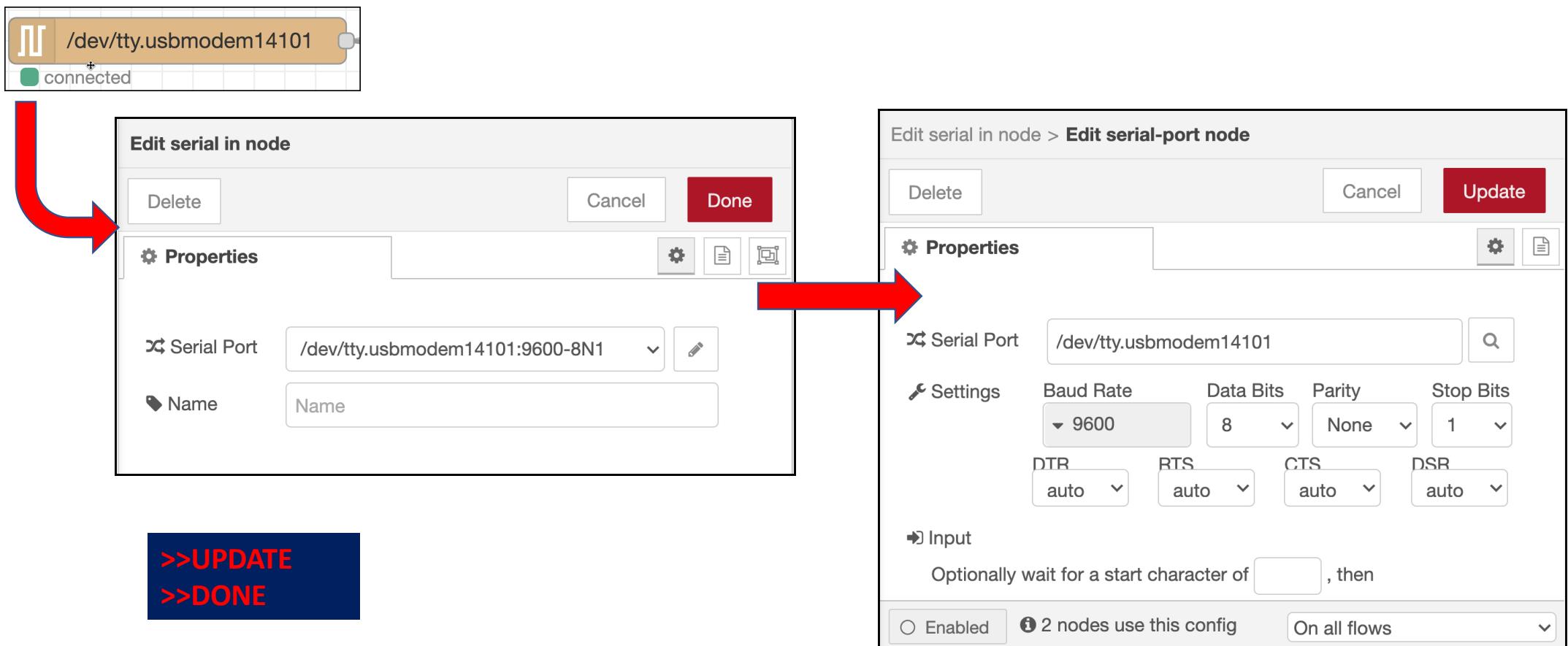
nodeRED + MySQL: From Sensor to nodeRED to Database

Step 1 >> Hook up your workspace as follows:



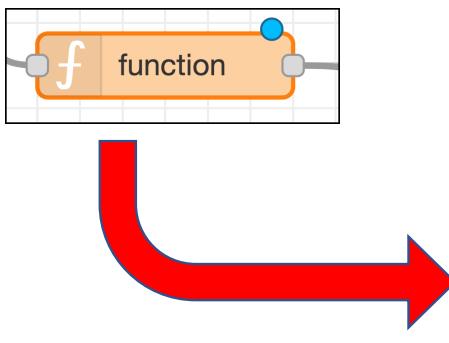
nodeRED + MySQL: From Sensor to nodeRED to Database

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

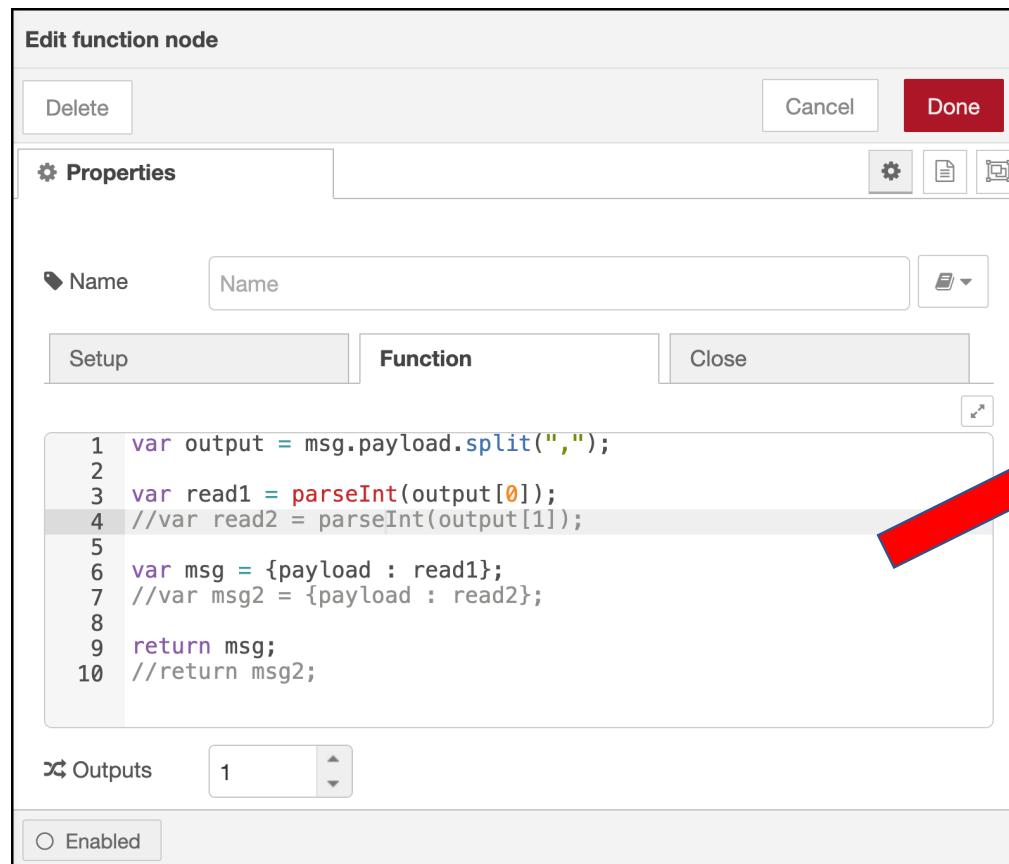


nodeRED + MySQL: From Sensor to nodeRED to Database

Step 2 >> Recap: Configuring Function node.



>>DONE



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

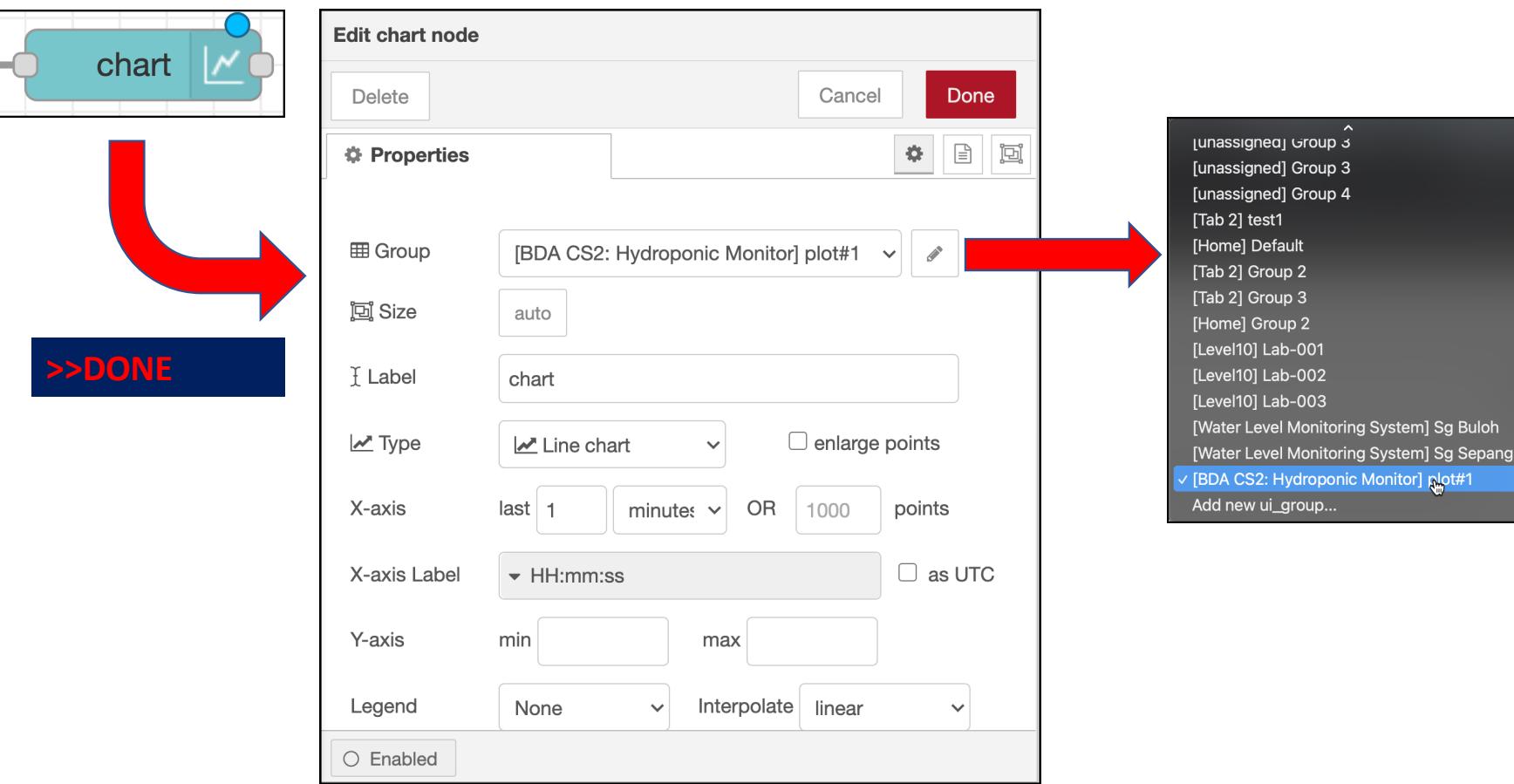
var msg = {payload : read1};
//var msg2 = {payload : read2};

return msg;
//return msg2;
```

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

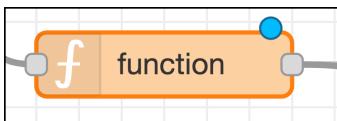
nodeRED + MySQL: From Sensor to nodeRED to Database

Step 3 >> Recap: Configuring Chart node.

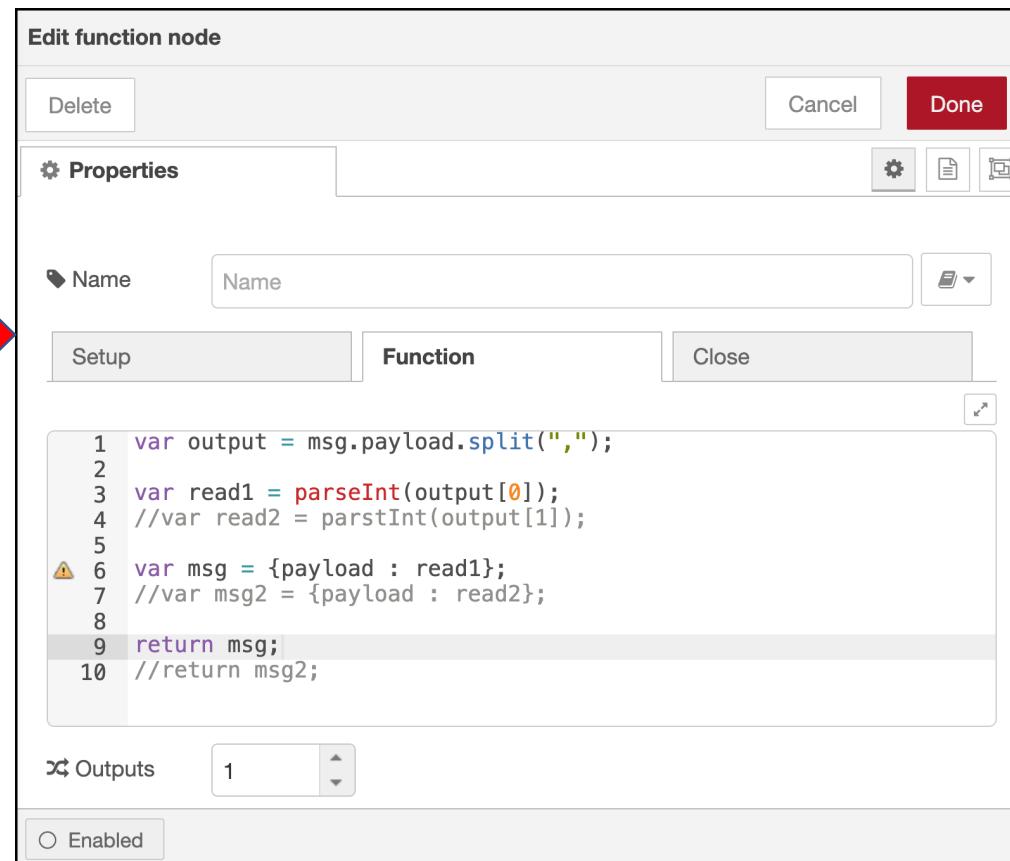


nodeRED + MySQL: From Sensor to nodeRED to Database

Step 4 >> Recap: Setting on Function node.



>>DONE



```

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

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

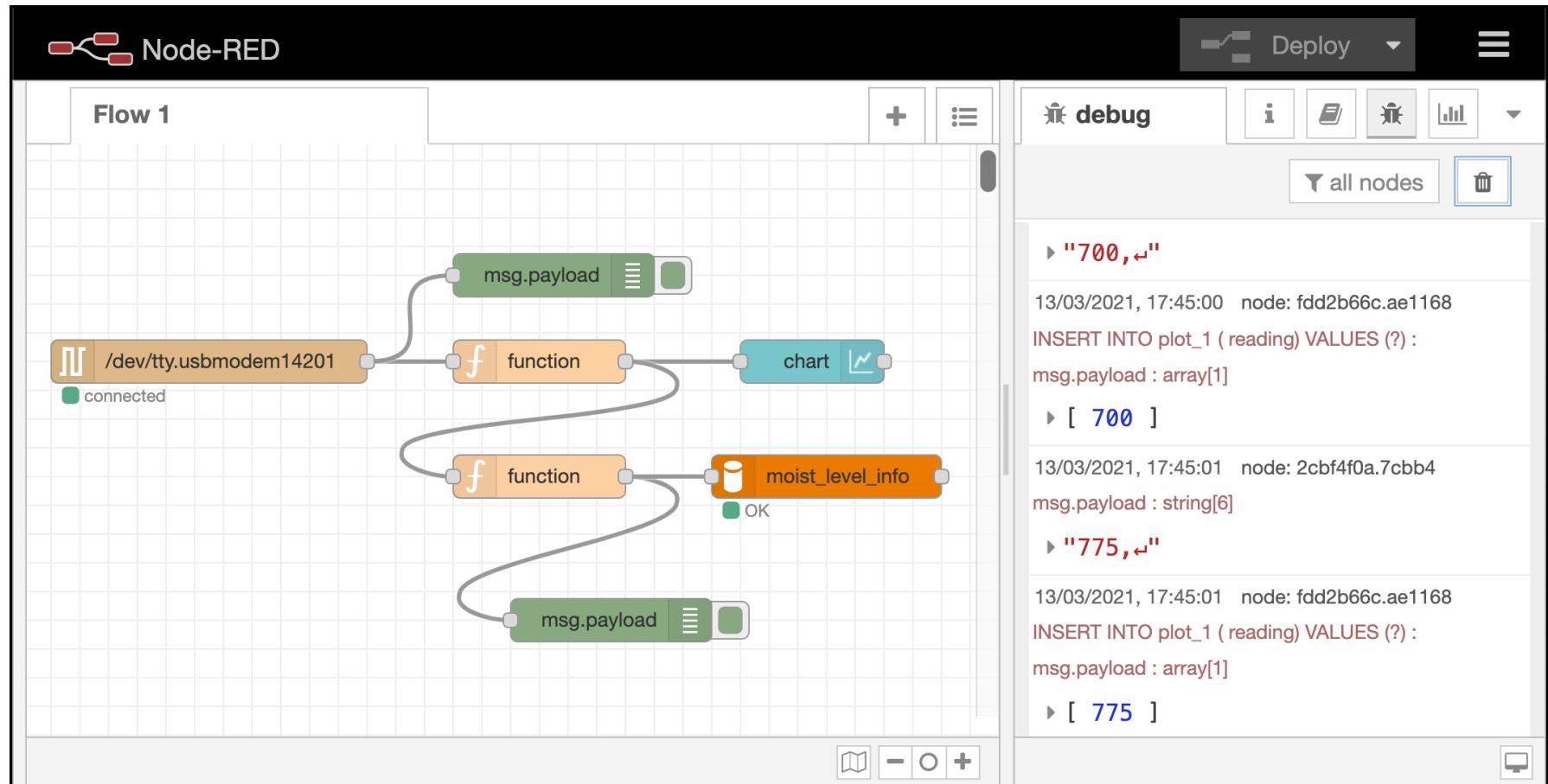
return msg;
//return msg2;

```

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

nodeRED + MySQL: From Sensor to nodeRED to Database

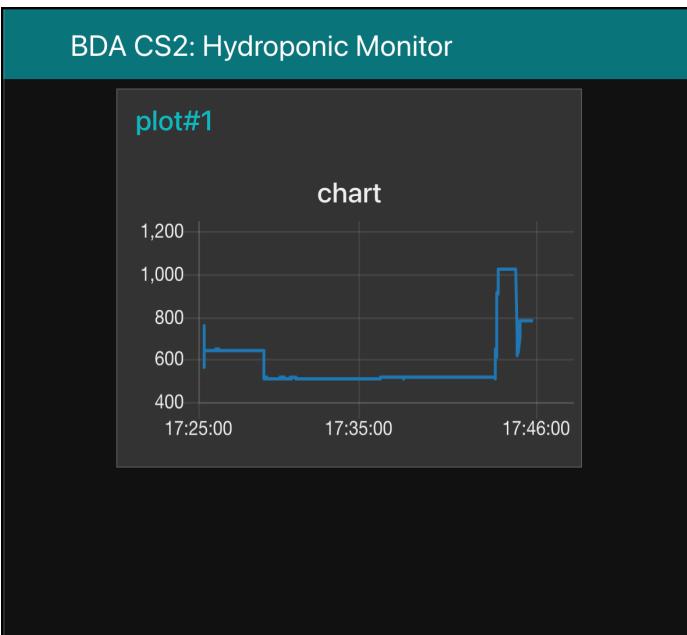
Step 5 >> Click Deploy & observe the output at Debug window.



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

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

+ Options ↪ ↮ id logs 1 reading

<input type="checkbox"/>	Edit	Copy	Delete	835	2021-03-13 17:46:07 779
<input type="checkbox"/>	Edit	Copy	Delete	834	2021-03-13 17:46:06 779
<input type="checkbox"/>	Edit	Copy	Delete	833	2021-03-13 17:46:05 779
<input type="checkbox"/>	Edit	Copy	Delete	832	2021-03-13 17:46:04 779
<input type="checkbox"/>	Edit	Copy	Delete	831	2021-03-13 17:46:03 779
<input type="checkbox"/>	Edit	Copy	Delete	830	2021-03-13 17:46:02 779
<input type="checkbox"/>	Console	Copy	Delete	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":"fd2b66c.ae1168","type":"debug","z":"7f9ea9bc.c4f338","name":"","active":true,"tostidebar":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;"}, {"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","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};\nreturn msg;\nreturn msg2;"}, {"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};\nreturn msg;\nreturn msg2;"}, {"id": "2cbf4f0a.7cbb4","type": "debug","z": "7f9ea9bc.c4f338","name": "", "active": true,"tostidebar": 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}]]
```

END