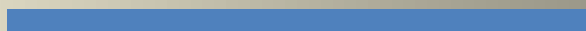





**FTP-PUSH
NODE-RED PROJECT
MANUAL**

MALAVIKA K.V



NODE-RED INSTALLATION SETUP

1. First need to install node-red using [Running Node-RED locally : Node-RED \(nodered.org\)](#)
2. Once installed as a global module you can use the node-red command to start Node-RED in your terminal. You can use Ctrl-C or close the terminal window to stop Node-RED.



```
node-red
at processTicksAndRejections (node:internal/process/task_queues:96:5)
C:\Users\NODE-RED>node-red
10 May 11:26:23 - [info]

Welcome to Node-RED
=====

10 May 11:26:23 - [info] Node-RED version: v2.2.2
10 May 11:26:23 - [info] Node.js version: v16.14.2
10 May 11:26:23 - [info] Windows_NT 10.0.19044 x64 LE
10 May 11:26:30 - [info] Loading palette nodes
10 May 11:26:45 - [info] Settings file : C:\Users\NODE-RED\.node-red\settings.js
10 May 11:26:45 - [info] Context store : 'default' [module=memory]
10 May 11:26:45 - [info] User directory : \Users\NODE-RED\.node-red
10 May 11:26:45 - [warn] Projects disabled : editorTheme.projects.enabled=false
10 May 11:26:45 - [info] Flows file : \Users\NODE-RED\.node-red\flows.json
10 May 11:26:45 - [info] Server now running at http://127.0.0.1:1880/
10 May 11:26:45 - [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.

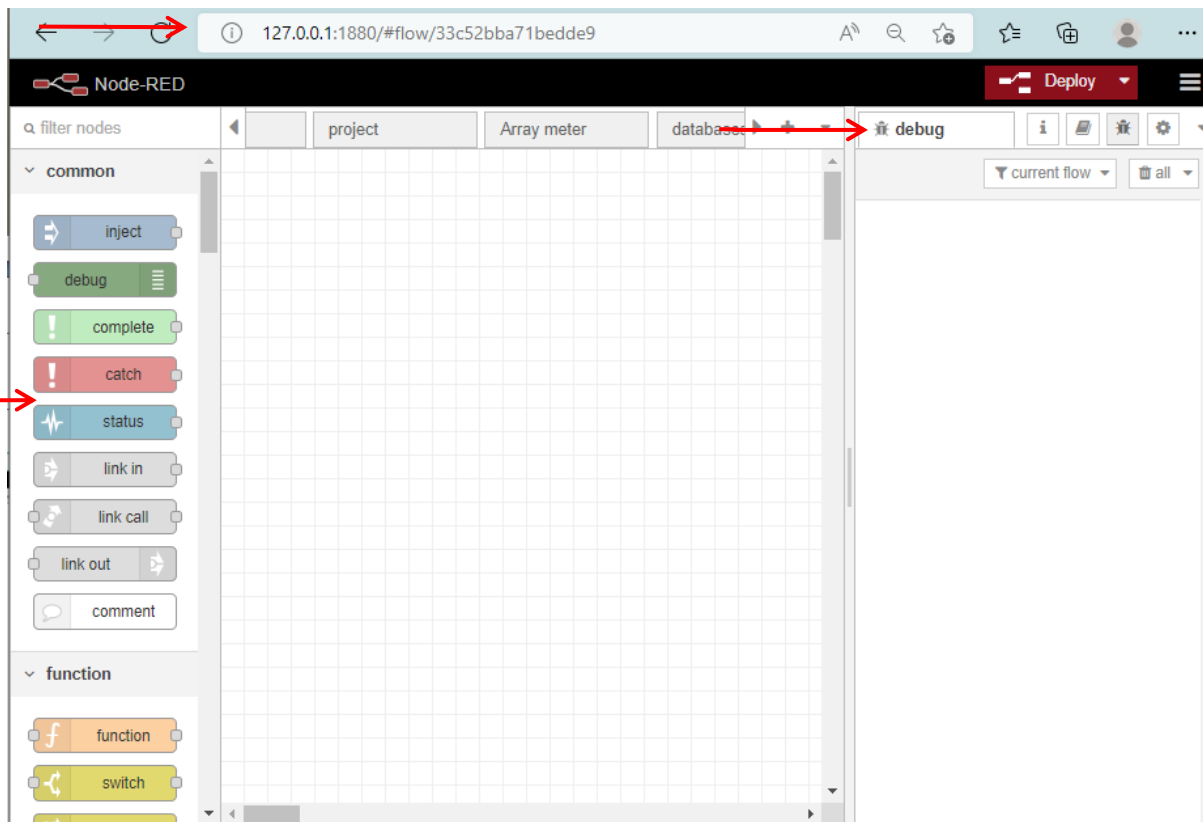
You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
```

3. Installation using docker command:

```
# docker run -it -p 1880:1880 -v node_red_data:/data --name mynodered
nodered/node-red
```

4. Access the editor with Node-RED [running](#), open the editor in a web browser. If you are using a browser on the same computer that is running Node-RED, you can access it with the url: <http://localhost:1880>. If you are using a browser on another computer, you will need to use the ip address of the computer running Node-RED: <http://<ip-address>:1880>.

5. A node-red workspace will open on the left side of workspace there will nodes to work on and in right side debug window to show outputs.



FTP PUSH PROJECT REQUIREMENTS

STEP 1: Collect every 15 min data of every block device, control room PAC, PDC, CUF, SOLAR_RADIATION, SOLAR_RADIATION_TILT, AIR AMP, MODULE_TEMP, WIND_SPEED, WIND_DIRECTION, HUMIDITY tags from medak and sircilla site and save this as csv file in local system for one full day with a file name of “[plant name]_[today's date].csv [eg: Medak_2022_12_22.csv].

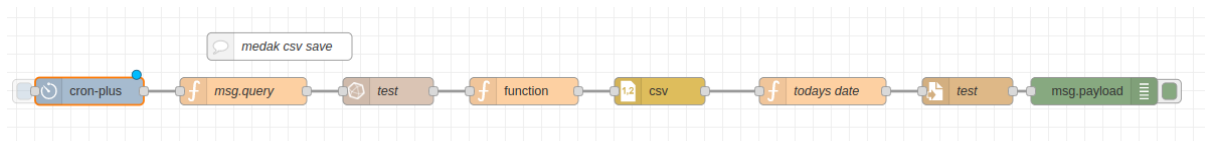
STEP 2: Sent this csv file to a server in every 15 min through ftp push.

STEP 3: Compare local file path and ftp directory path to check any unsent files due to internet connection loss if there is any sent that to ftp server.

NODE-RED FLOW FOR STEP 1:

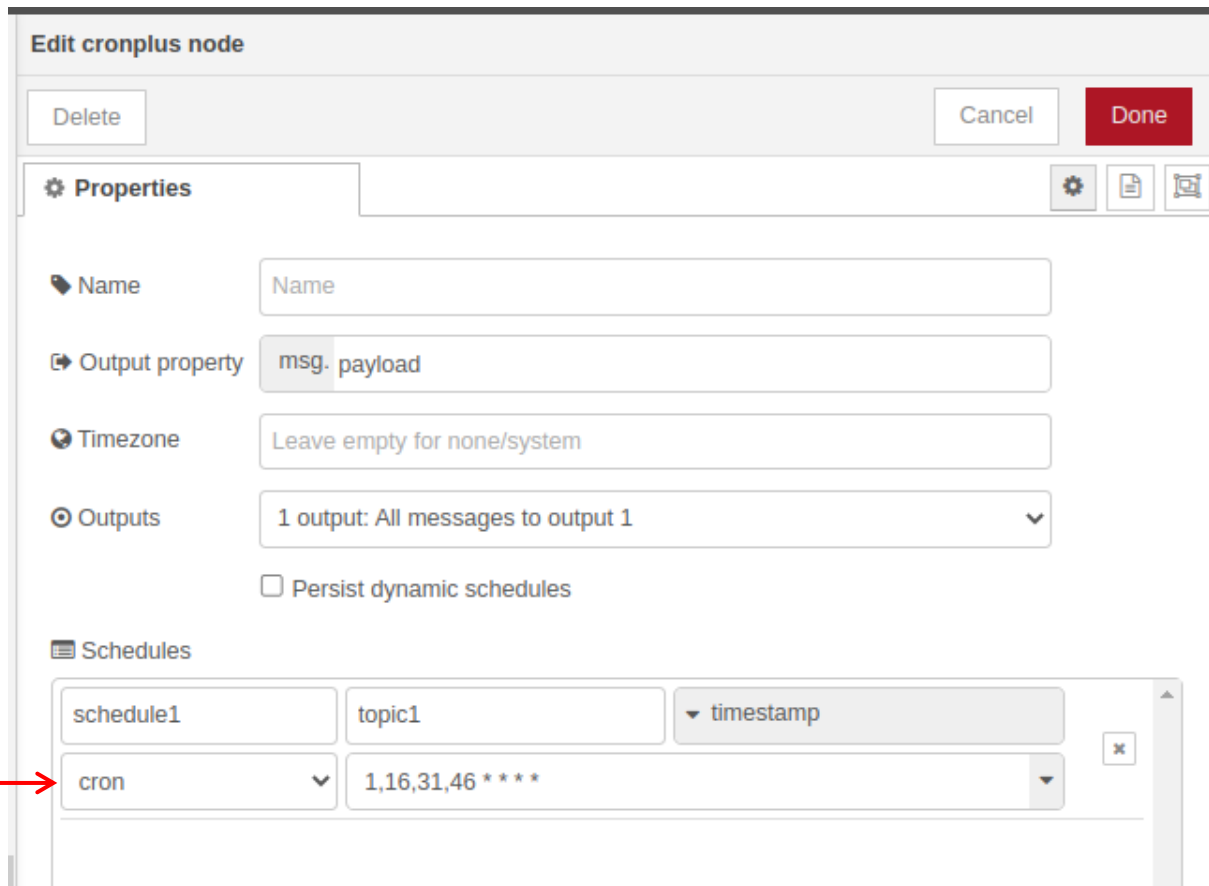
Collect every 15 min data of every block device, control room PAC, PDC, CUF, SOLAR_RADIATION, SOLAR_RADIATION_TILT, AIR_AMP, MODULE_TEMP, WIND_SPEED, WIND_DIRECTION, HUMDITIY tags from medak and sircilla site and save this as csv file in local system for one full day with a file name of “[plant name]_[todays date]”.csv [eg: Medak_2022_12_22.csv].

NODE-RED FLOW FOR STEP 1:



Cron plus Node:

- 1 The cron plus node allows you to inject messages into a flow, either by clicking the button on the node, or setting a time interval between injects.
- 2 You have to install this node package externally (node-red-contrib-cron-plus)from manage palette feature(click 3 lines symbol in the top right corner in the node-red workspace)
- 3 Select the newly added Inject node to see information about its properties and a description of what it does in the [Information sidebar pane](#).



Here we are setting up the triggering time to every 1st, 16th, 31st, 46th min repeat that will trigger the corresponding flow in the pre-set time everyday.

Function node : Here we used function node to give msg.query output to influxdb in node. If we need to query for a variable start and stop time we can use javascript logic to set up start and stop time here we need midnight timestamp[12.00AM] of that day as start time and triggering time as end timestamp

```
var $today = new Date();
```

```
var d = new Date($today);
```

```
d.setDate($today.getDate() - 1);
```

```
d.setHours(18,30,0,0);
```

```
var tstart=d.getTime();
```

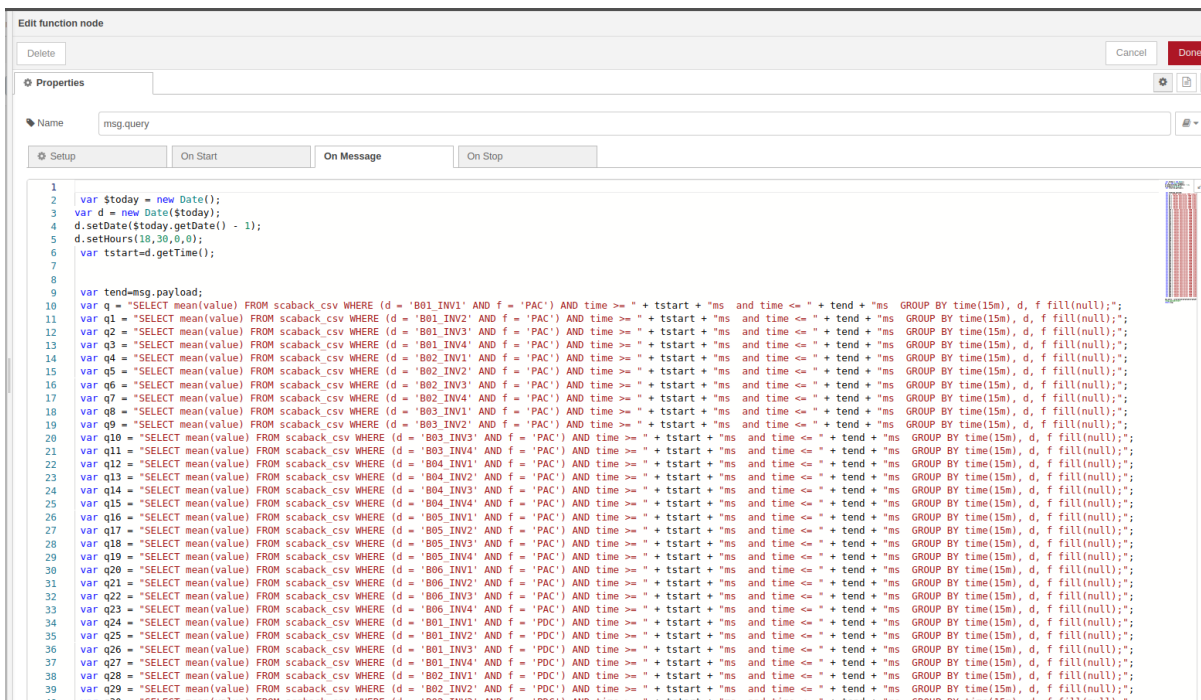
```
var tend=msg.payload;
```

Then add query of different devices we want one by one in double quotes

Eg: `var q = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B01_INV1' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";`

after add all devices like this add all the query to msg.query

`msg.query=q1+q2+.....+q81 like that`



```

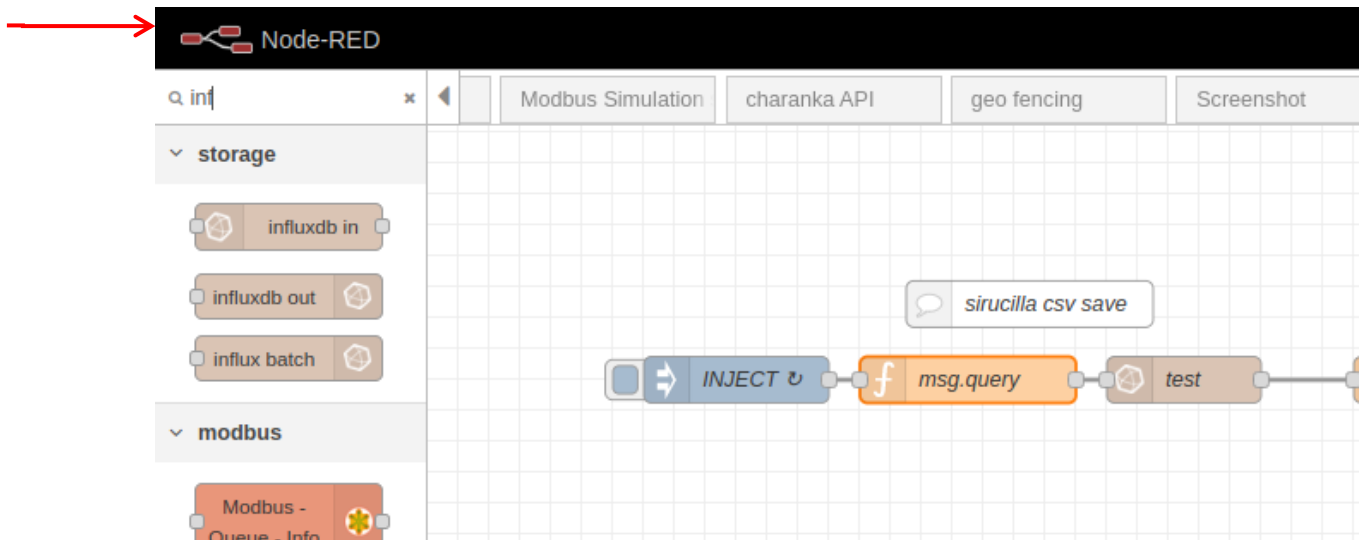
1  var $today = new Date();
2  var d = new Date($today);
3  d.setDate($today.getDate() - 1);
4  d.setHours(18,30,0,0);
5  var tstart=d.getTime();
6
7
8
9  var tend=msg.payload;
10 var q = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B01_INV1' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
11 var q1 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B01_INV2' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
12 var q2 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B01_INV3' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
13 var q3 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B01_INV4' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
14 var q4 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B02_INV1' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
15 var q5 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B02_INV2' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
16 var q6 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B02_INV3' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
17 var q7 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B02_INV4' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
18 var q8 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B03_INV1' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
19 var q9 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B03_INV2' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
20 var q10 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B03_INV3' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
21 var q11 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B03_INV4' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
22 var q12 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B04_INV1' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
23 var q13 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B04_INV2' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
24 var q14 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B04_INV3' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
25 var q15 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B04_INV4' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
26 var q16 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B05_INV1' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
27 var q17 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B05_INV2' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
28 var q18 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B05_INV3' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
29 var q19 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B05_INV4' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
30 var q20 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B06_INV1' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
31 var q21 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B06_INV2' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
32 var q22 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B06_INV3' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
33 var q23 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B06_INV4' AND f = 'PAC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
34 var q24 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B01_INV1' AND f = 'PDC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
35 var q25 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B01_INV2' AND f = 'PDC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
36 var q26 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B01_INV3' AND f = 'PDC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
37 var q27 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B01_INV4' AND f = 'PDC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
38 var q28 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B02_INV1' AND f = 'PDC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
39 var q29 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B02_INV2' AND f = 'PDC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";
40 var q30 = "SELECT mean(value) FROM scaback_csv WHERE (d = 'B02_INV3' AND f = 'PDC') AND time >= " + tstart + "ms and time <= " + tend + "ms GROUP BY time(15m), d, f fill(null);";

```

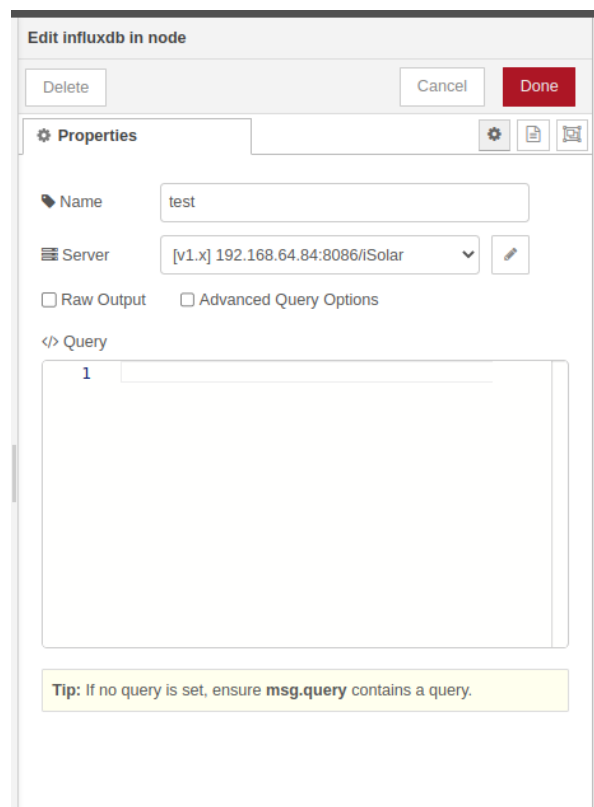
Influxdb in node ([node-red-contrib-influxdb \(node\) - Node-RED \(nodered.org\)](#)):

To access this node we have to install the node package(node-red-contrib-influxdb)from manage palette feature(click 3 lines symbol in the top right corner in the node-red workspace)

Nodes to query data from an influxdb time series database. Supports InfluxDb versions 1.x to 2.0.



double click on the influx db node



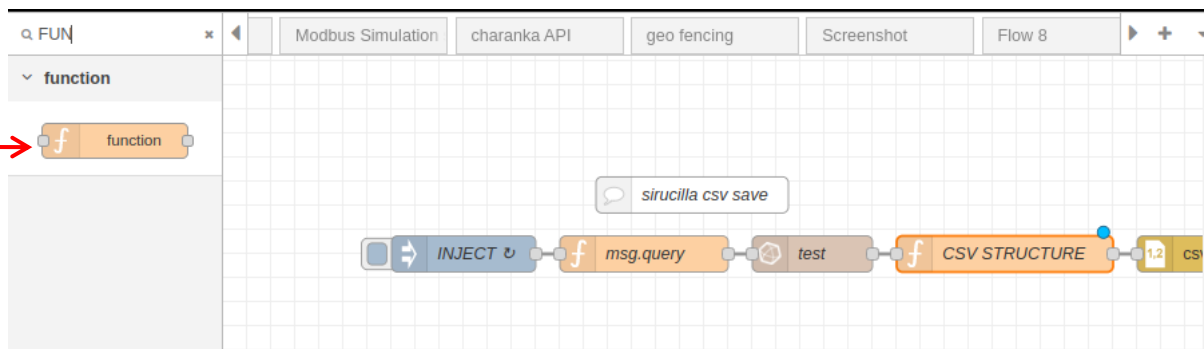
keep the query box blank if you are giving msg.query through a function node

Then click on the pencil icon to add details of the new server

The screenshot shows a configuration window titled "Edit influxdb in node > Edit influxdb node". At the top, there are three buttons: "Delete", "Cancel", and "Update". Below these is a tabbed interface with the "Properties" tab selected. The "Properties" tab contains several input fields: "Name" (text input), "Version" (dropdown menu showing "1.x"), "Host" (text input showing "192.168.64.84") and "Port" (text input showing "8086"), "Database" (text input showing "iSolar"), "Username" (text input), and "Password" (text input). At the bottom, there is a checkbox labeled "Enable secure (SSL/TLS) connection" which is currently unchecked.

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

The message is passed in as an object called **msg**. By convention it will have a **msg.payload** property containing the body of the message. Other nodes may attach their own properties to the message, and they should be described in their documentation.



Feed the javascript code on the message box

Edit function node

Delete

Cancel

Done

Properties

Name

CSV STRUCTURE

Setup

On Start

On Message

On Stop

```

1  let array = [];
2  let array1 = [];
3  var value = msg.payload.length;
4  let plantObj
5
6  plantObj = []
7
8  var j=0;
9
10
11  for(var i=0;i<msg.payload.length;i++)
12  {
13      if(msg.payload[i].length == 0)
14      {
15          for(var k=0;k<msg.payload[0].length;k++)
16          {
17              msg.payload[i][k] = 0;
18          }
19      }
20  }
21  }
22  }
23
24  for (i = 0; i < msg.payload[0].length;i++)
25  {
26      var dateIST =new Date(msg.payload[0][i].time).toLocaleString("en-US", {timeZone: 'Asia/
27
28      var obj=
29      {
30          TIME:dateIST,
31          B01_INV1_PAC_KW:msg.payload[0][i].mean,
32          B01_INV2_PAC_KW:msg.payload[1][i].mean,
33          B01_INV3_PAC_KW:msg.payload[2][i].mean,
34          B01_INV4_PAC_KW:msg.payload[3][i].mean,
35          B02_INV1_PAC_KW:msg.payload[4][i].mean,
36          B02_INV2_PAC_KW:msg.payload[5][i].mean,
37          B02_INV3_PAC_KW:msg.payload[6][i].mean,
38          B02_INV4_PAC_KW:msg.payload[7][i].mean,
39          B03_INV1_PAC_KW:msg.payload[8][i].mean,
40          B03_INV2_PAC_KW:msg.payload[9][i].mean

```

```
let array = [];

let array1 = [];
var value = msg.payload.length;
let plantObj

plantObj = []

var j=0;
for(var i=0;i<msg.payload.length;i++)
{
if(msg.payload[i].length == 0)
{
for(var k=0;k<msg.payload[0].length;k++)
{
msg.payload[i][k] = 0;
}
}
}
for (i = 0; i < msg.payload[0].length;i++)
{
var dateIST =new Date(msg.payload[0][i].time).toLocaleString("en-US", {timeZone: 'Asia/Kolkata'});

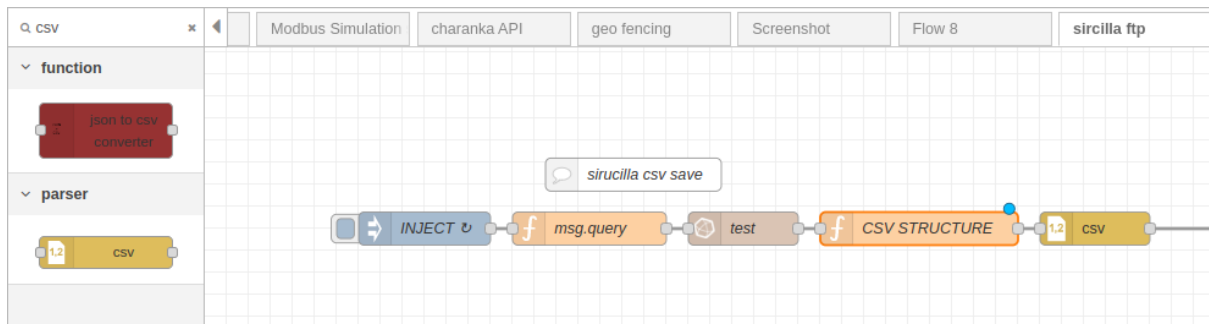
var obj=
{
TIME:dateIST,
B01_INV1_PAC_KW:msg.payload[0][i].mean,
B01_INV2_PAC_KW:msg.payload[1][i].mean,
B01_INV3_PAC_KW:msg.payload[2][i].mean,
B01_INV4_PAC_KW:msg.payload[3][i].mean,
B02_INV1_PAC_KW:msg.payload[4][i].mean,
B02_INV2_PAC_KW:msg.payload[5][i].mean,
B02_INV3_PAC_KW:msg.payload[6][i].mean,
B02_INV4_PAC_KW:msg.payload[7][i].mean,
B03_INV1_PAC_KW:msg.payload[8][i].mean,
B03_INV2_PAC_KW:msg.payload[9][i].mean,
B03_INV3_PAC_KW:msg.payload[10][i].mean,
B03_INV4_PAC_KW:msg.payload[11][i].mean,
B04_INV1_PAC_KW:msg.payload[12][i].mean,
B04_INV2_PAC_KW:msg.payload[13][i].mean,
B04_INV3_PAC_KW:msg.payload[14][i].mean,
B04_INV4_PAC_KW:msg.payload[15][i].mean,
B05_INV1_PAC_KW:msg.payload[16][i].mean,
B05_INV2_PAC_KW:msg.payload[17][i].mean,
B05_INV3_PAC_KW:msg.payload[18][i].mean,
B05_INV4_PAC_KW:msg.payload[19][i].mean,
B06_INV1_PAC_KW:msg.payload[20][i].mean,
```

```
B06_INV2_PAC_KW:msg.payload[21][i].mean,
B06_INV3_PAC_KW:msg.payload[22][i].mean,
B06_INV4_PAC_KW:msg.payload[23][i].mean,
B01_INV1_PDC_KW:msg.payload[24][i].mean,
B01_INV2_PDC_KW:msg.payload[25][i].mean,
B01_INV3_PDC_KW:msg.payload[26][i].mean,
B01_INV4_PDC_KW:msg.payload[27][i].mean,
B02_INV1_PDC_KW:msg.payload[28][i].mean,
B02_INV2_PDC_KW:msg.payload[29][i].mean,
B02_INV3_PDC_KW:msg.payload[30][i].mean,
B02_INV4_PDC_KW:msg.payload[31][i].mean,
B03_INV1_PDC_KW:msg.payload[32][i].mean,
B03_INV2_PDC_KW:msg.payload[33][i].mean,
B03_INV3_PDC_KW:msg.payload[34][i].mean,
B03_INV4_PDC_KW:msg.payload[35][i].mean,
B04_INV1_PDC_KW:msg.payload[36][i].mean,
B04_INV2_PDC_KW:msg.payload[37][i].mean,
B04_INV3_PDC_KW:msg.payload[38][i].mean,
B04_INV4_PDC_KW:msg.payload[39][i].mean,
B05_INV1_PDC_KW:msg.payload[40][i].mean,
B05_INV2_PDC_KW:msg.payload[41][i].mean,
B05_INV3_PDC_KW:msg.payload[42][i].mean,
B05_INV4_PDC_KW:msg.payload[43][i].mean,
B06_INV1_PDC_KW:msg.payload[44][i].mean,
B06_INV2_PDC_KW:msg.payload[45][i].mean,
B06_INV3_PDC_KW:msg.payload[46][i].mean,
B06_INV4_PDC_KW:msg.payload[47][i].mean,
PLANT_ACTIVEPOWER_KW:(msg.payload[48][i].mean)*1000,
PR:msg.payload[49][i].mean,
CUF:msg.payload[50][i].mean,
"GHI_Wm2":msg.payload[60][i].mean,
"GTI_Wm2":msg.payload[61][i].mean,
AIR_TEMP_°C:msg.payload[62][i].mean,
MODULE_TEMP_°C:msg.payload[63][i].mean,
WIND_SPEED_Kmph:msg.payload[64][i].mean,
WIND_DIRECTION_°:msg.payload[65][i].mean,
"HUMITIDY_%":msg.payload[66][i].mean
}
plantObj.push(obj)
}
msg.payload = plantObj

return msg
```

This javascript code is used to check every message in the array of output getting from influx output using a for loop function and to take make a csv struture as per requirement

csv node:



Tick the boxes like this and take the output as a message per row

The 'Edit csv node' dialog box is shown. It has a title bar 'Edit csv node' and buttons for 'Delete', 'Cancel', and 'Done'. The 'Properties' tab is active. Under 'Columns', the 'Columns' field is set to 'comma-separated column names'. The 'Separator' is set to 'comma'. The 'Name' field is set to 'Name'. Under 'CSV to Object options', the 'Input' section has 'Skip first' set to '0' lines. The checkboxes for 'first row contains column names', 'parse numerical values', and 'include null values' are all checked. The 'include empty strings' checkbox is unchecked. The 'Output' section has a dropdown set to 'a message per row'. Under 'Object to CSV options', the 'Output' section has a dropdown set to 'always send column headers'. The 'Newline' section has a dropdown set to 'Linux (\n)'. At the bottom, there is a checkbox labeled 'Enabled' which is currently unchecked.

While writing the code in other devices the file path will be different that has to change accordingly.

Edit function node

Delete Cancel Done

Properties

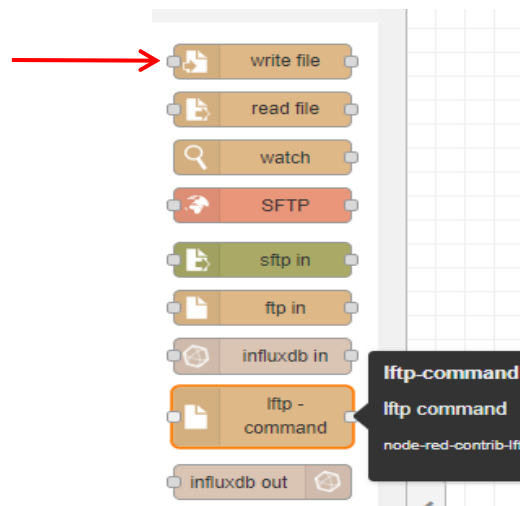
Name: todays date

Setup On Start **On Message** On Stop

```
1 var currentTime = new Date();
2
3 var currentOffset = currentTime.getTimezoneOffset();
4
5 var ISTOffset = 330; // IST offset UTC +5:30
6
7 var today = new Date(currentTime.getTime() + (ISTOffset + currentOffset)*60000);
8 var dd = String(today.getDate()).padStart(2, '0');
9 var mm = String(today.getMonth() + 1).padStart(2, '0'); //January is 0!
10 var yyyy = today.getFullYear();
11 var today1 = dd + '_' + mm + '_' + yyyy;
12
13
14 var c =today1;
15 context.set('c',c)
16
17 var path = "/usr/src/node-red/ftp/sircilla_"+c+".csv"
18 msg.filename = path
19 return msg;
```

☒ Enabled

Write file node:



Edit write file node

Delete Cancel Done

Properties

Filename

Action

☒ Add newline (\n) to each payload?

☒ Create directory if it doesn't exist?

Encoding

Name

Tip: The filename should be an absolute path, otherwise it will be relative to the working directory of the Node-RED process.

☐ Enabled

No need to fill the file name box because we already giving it through function node

Debug node:

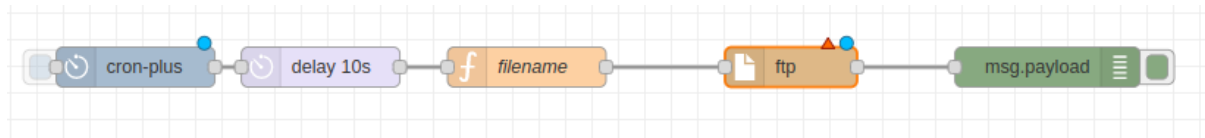


The Debug node causes any message to be displayed in the [Debug sidebar](#). By default, it just displays the payload of the message, but it is possible to display the entire message object.

- 1 Click the Deploy button. With the Debug sidebar tab selected,
- 2 Click the Inject button to get output if there is no triggering time set.
- 3 Check the destination of file what we give as path to check whether the csv file created or not

STEP 2: Sent this csv file to a server in every 15 min through ftp push.

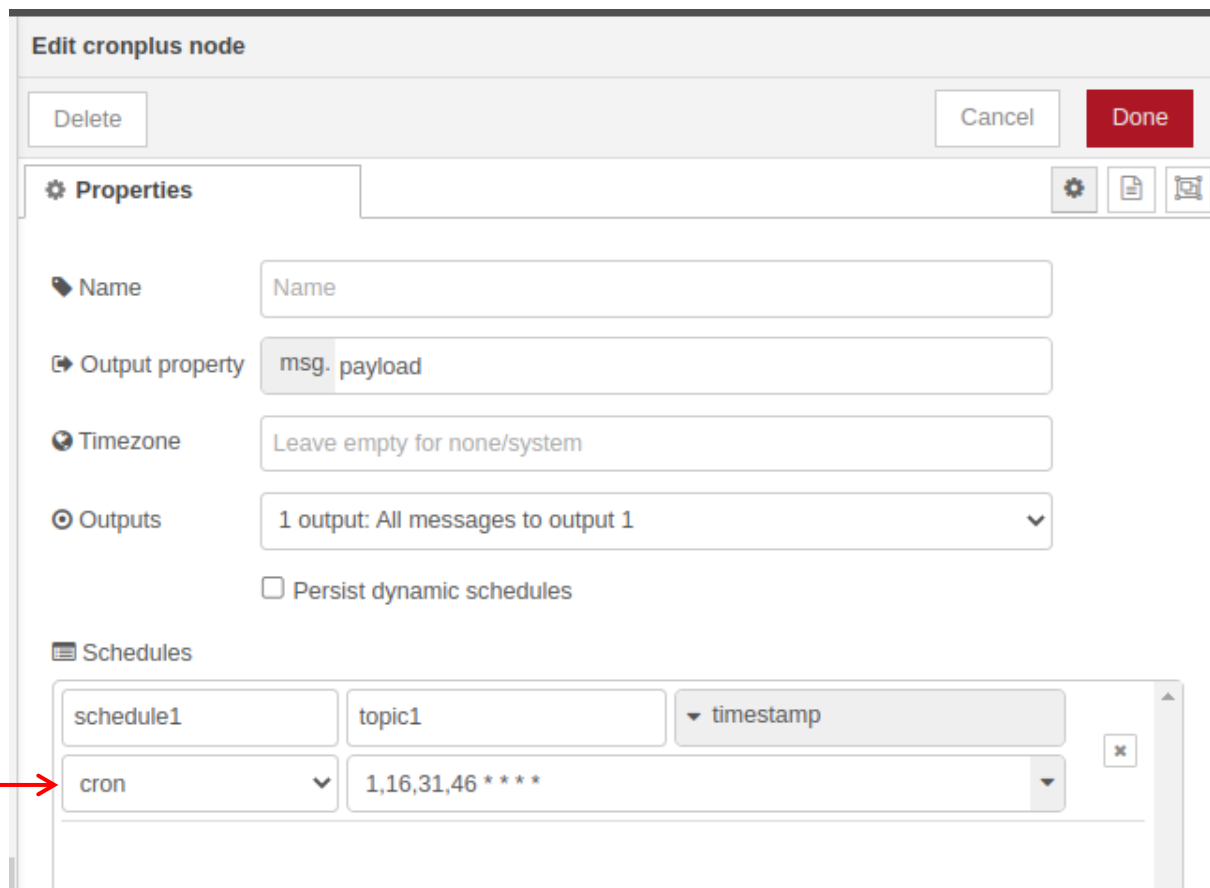
NODE-RED FLOW FOR STEP 2:



Here we are sending the csv file that we saved using the first flow through ftp node

Cron plus Node:

- 4 The cron plus node allows you to inject messages into a flow, either by clicking the button on the node, or setting a time interval between injects.
- 5 You have to install this node package externally (node-red-contrib-cron-plus)from manage palette feature(click 3 lines symbol in the top right corner in the node-red workspace)
- 6 Select the newly added Inject node to see information about its properties and a description of what it does in the [Information sidebar pane](#).



Edit cronplus node

Delete Cancel Done

Properties

Name: Name

Output property: msg. payload

Timezone: Leave empty for none/system

Outputs: 1 output: All messages to output 1

☐ Persist dynamic schedules

Schedules

schedule1	topic1	timestamp
cron	1,16,31,46 ****	timestamp

Here we are setting up the triggering time to every 1st, 16th, 31st, 46th min repeat that will trigger the corresponding flow in the pre-set time everyday.

Delay Node:

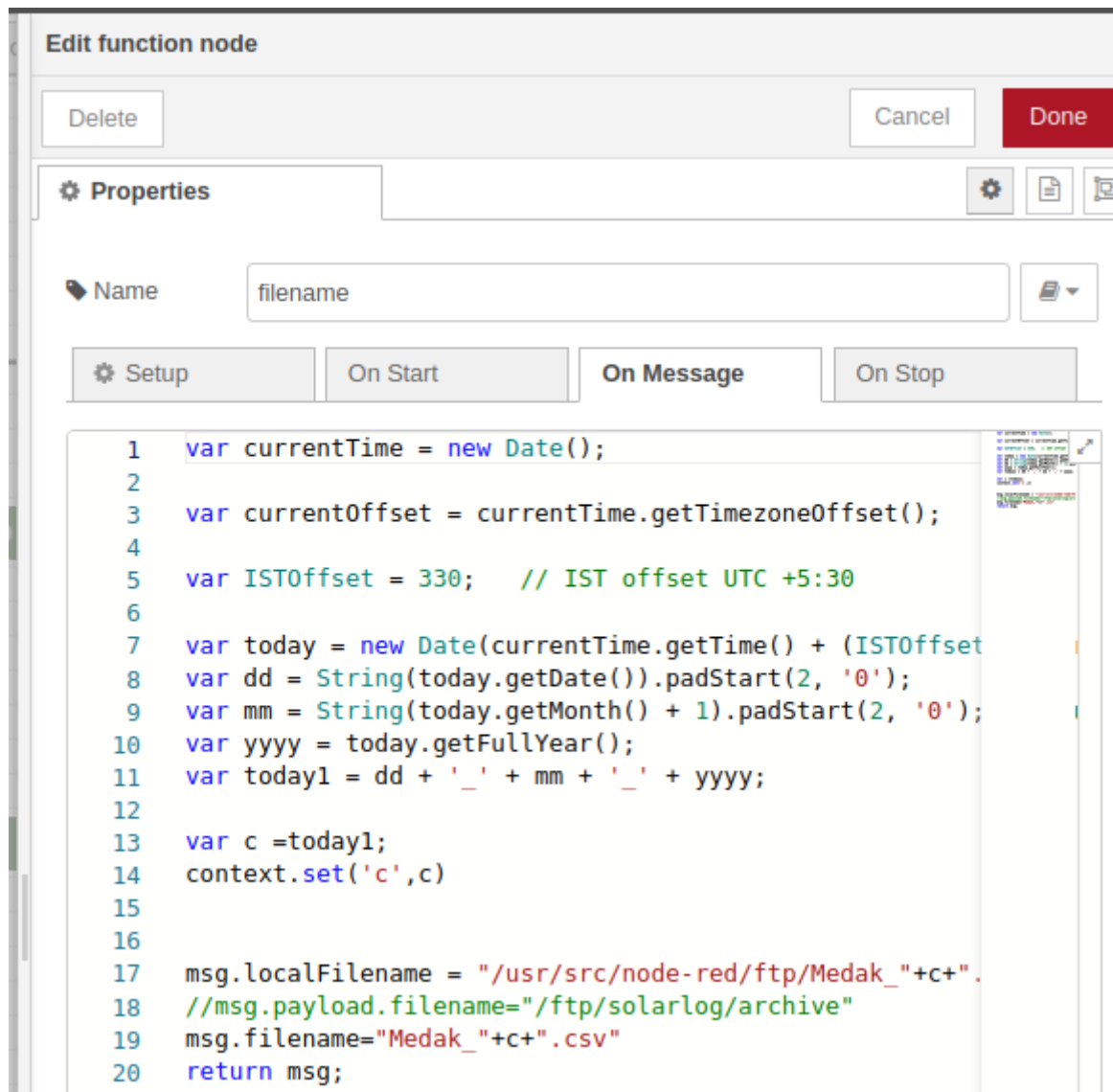
Cron plus node Then connected to delay node avoid triggering conflict because we after used same triggering in first flow [delay node help us to sent csv file only after it updated in timing]from first flow

The screenshot shows the 'Edit delay node' configuration window. At the top, there are three buttons: 'Delete', 'Cancel', and 'Done'. Below these is a 'Properties' tab with a gear icon. The configuration options are as follows:

- Action:** A dropdown menu set to 'Delay each message'.
- Fixed delay:** A dropdown menu set to 'Fixed delay'.
- For:** A numeric input field set to '10' and a unit dropdown menu set to 'Seconds'.
- Name:** A text input field with the placeholder text 'Name'.

Function node:

Delay node connected to a function node (today's date) for giving file path as today's date or read the same csv file we saved in today's date.



ftp in Node:

Install the node by installing node package(node-red-contrib-ftp)drag and drop the node

Edit ftp in node

Delete Cancel Done

Properties

Add new FTP Server Add new ftp...

Operation put

Filename Filename

Local Filename Local Filename

Name Name

Set Operation –put

Filename - leave this box blank so it will take filename from the function node connected behind it.

Local Filename-leave this box blank so it will take filename from the function node connected behind it.

Setup the server configuration as per below figure (click the pencil icon)enter the username and password to authorize the server.

Set port always 21 for FTP

Edit ftp in node > Add new ftp config node

Cancel Add

Properties

Host iplon-pvindia.com Port 21

User iplon

Password

Connection Timeout 10000

PASV Timeout 10000

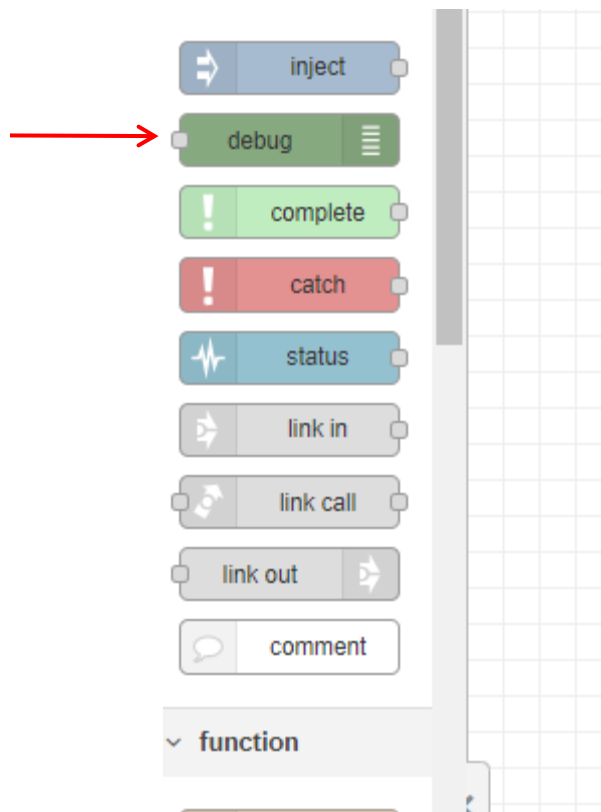
Keepalive 10000

Data connection encryption ☐ Secure Options

Debug node:

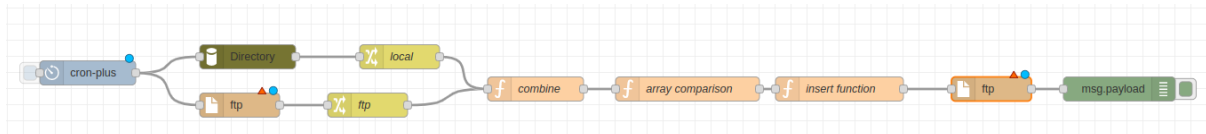
The Debug node causes any message to be displayed in the [Debug sidebar](#). By default, it just displays the payload of the message, but it is possible to display the entire message object.

- 1 Click the Deploy button. With the Debug sidebar tab selected,
- 2 Click the Inject button to get output if there is no triggering time set.



NODE-RED FLOW FOE STEP-3

STEP 3: Compare local file path and ftp directory path to check any unsent files due to internet connection loss if there is any sent that to ftp server.



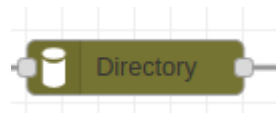
Cron plus Node:

1. The cron plus node allows you to inject messages into a flow, either by clicking the button on the node, or setting a time interval between injects.
2. You have to install this node package externally (node-red-contrib-cron-plus) from manage palette feature (click 3 lines symbol in the top right corner in the node-red workspace)
3. Select the newly added Inject node to see information about its properties and a description of what it does in the [Information sidebar pane](#).

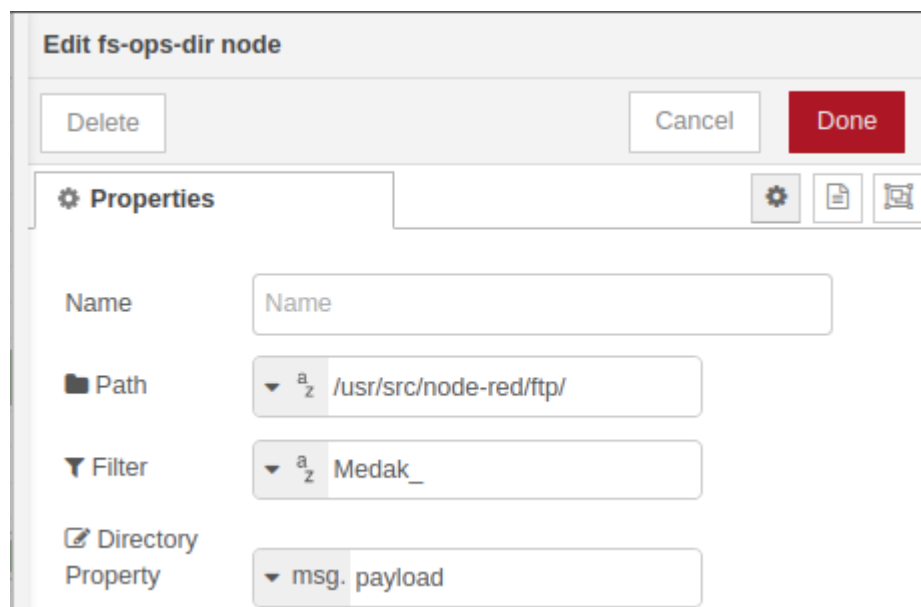
schedule1	topic1	Default Payload
cron	0 1 ***	

Here we are setting up the triggering time to every 1min it will trigger the corresponding flow in the pre-set time everyday. Connecting this cron plus node to both nodes (fs-ops-dir and ftp in) so that both will trigger at same time

Fs-ops-dir Node:



Here we are going to compare 2 directories one directory is in local server where the file are saved from flow 1 next directory is in ftp push server so first we have to list out files in local server directory for that we are using fs-ops-dir node we can install this using external node package (node-red-contrib-fs-ops)

The image shows the 'Edit fs-ops-dir node' dialog box. It has a title bar 'Edit fs-ops-dir node' and three buttons: 'Delete', 'Cancel', and 'Done'. Below the buttons is a 'Properties' section with a gear icon and three sub-icons. The properties are: 'Name' with a text input field; 'Path' with a dropdown menu showing 'a_z' and the path '/usr/src/node-red/ftp/'; 'Filter' with a dropdown menu showing 'a_z' and the filter 'Medak_'; and 'Directory Property' with a checked checkbox and a dropdown menu showing 'msg.payload'.

inside this node we have to give path of the directory from where we have to list about the files if there is any filter we can give that also for better listing in this our file format is like Medak_20_12_2011.csv so that I am filtering it with “Medak_” then set directory property =msg.payload

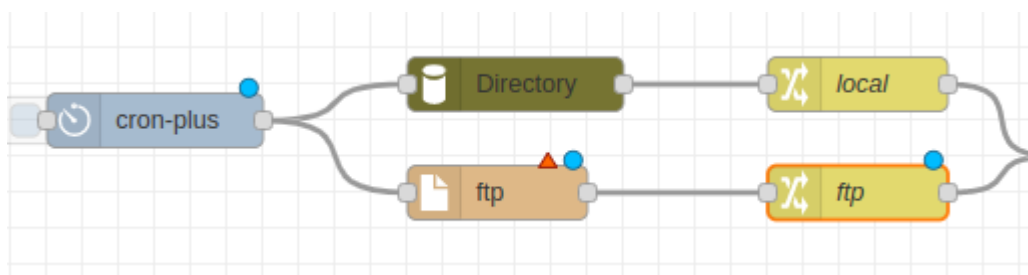
FTP in node:

Parellel to fs-ops-dir node we are setting ftp in node also for listing files in the ftp push server for that we have to give ftp server configurations and set operation to list

The screenshot shows the 'Edit ftp in node' dialog box. At the top, there are buttons for 'Delete', 'Cancel', and 'Done'. Below this is a 'Properties' section with a gear icon and three sub-sections: 'Add new FTP Server' with a dropdown menu showing 'Add new ftp...' and an edit icon; 'Operation' with a dropdown menu showing 'list'; and 'Name' with a text input field containing 'Name'.

Change Node:

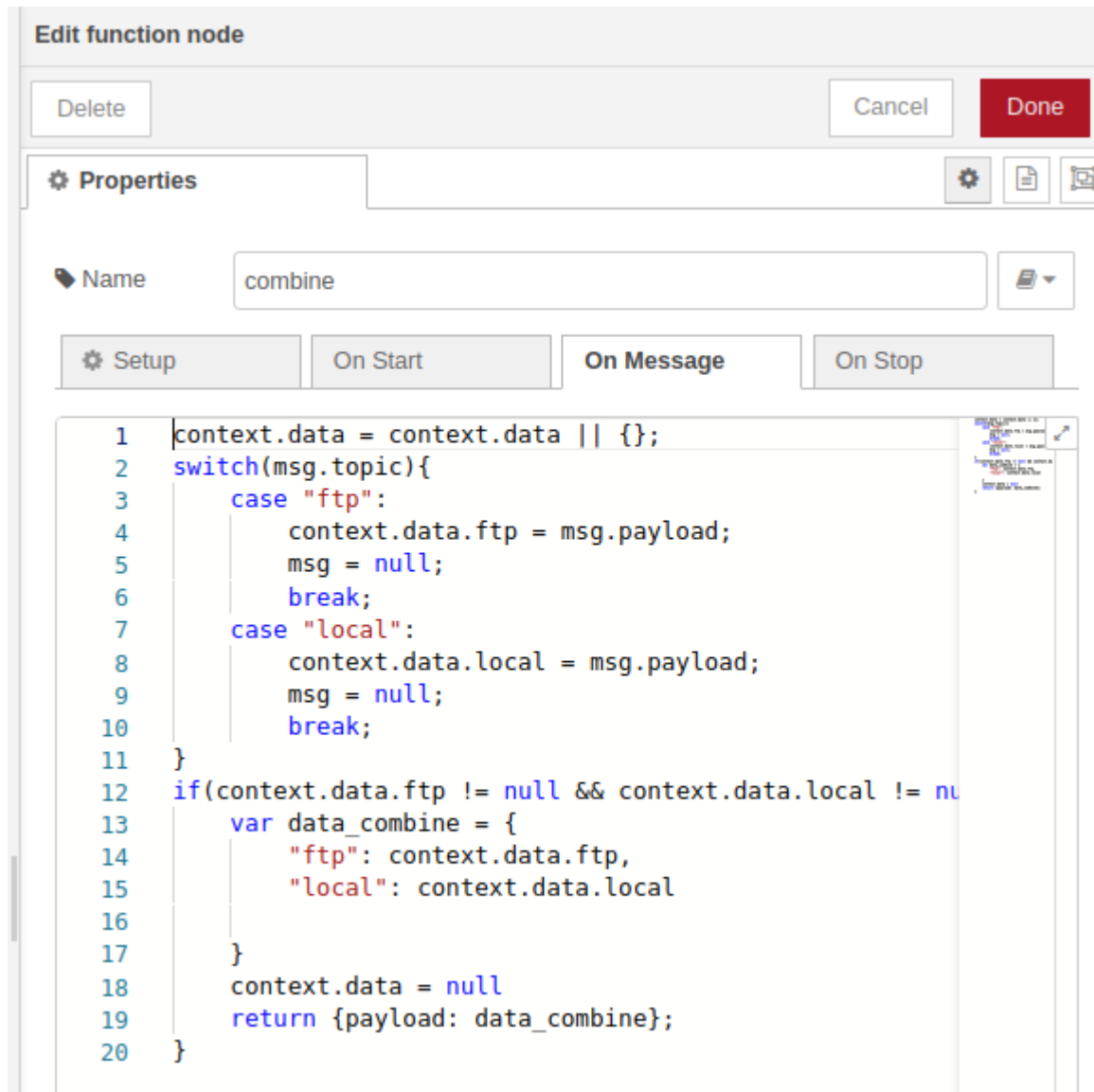
we are connecting both fs-ops-dir and ftp in node to change node to set topic for both outputs



The screenshot shows the 'Edit change node' dialog box. It has 'Delete', 'Cancel', and 'Done' buttons at the top. The 'Properties' section includes a 'Name' field with the value 'local'. The 'Rules' section shows a 'Set' operation on the 'msg. topic' property, with a rule 'to the value' set to 'a_z local'.

There you have to set msg.topic to any name according to your flow [use the same in combine logic]

Function node for combine outputs from both flows:



Inside function we are writing logic to combine outputs from both flows and merge it to one array of output

combine output will be look like this



Function node to compare both arrays:

- In this logic we are first defining length of both array
- Defining 2 empty array to push files names from the array of outputs and one more empty array to save filtered file names
- Normally if there is any unsent files are there, The array length won't be same (local files number will be more than files in ftp server) in that case only we have to send the unsent file to ftp server so that we are setting a if condition to check both array length if that is not equal to only it will go inside to take out the unsent file
- If the array lengths are not equal it will go inside a for loop to read array one by one and push the file names to new array we created.
- Then it compares filenames in both arrays by using if condition if it is equal it goes inside a condition object and sets flag = 1
- so that flag = 0 means filename which is not equal or not included so we can push this filtered filename to the 3rd array we create and take out this array only to payload.

Edit function node

Delete
Cancel
Done

Properties

Name
array comparison

Setup
On Start
On Message
On Stop

```

1  var local_LENGTH = msg.payload.local.length
2
3  var ftp_LENGTH = msg.payload.ftp.length
4
5  let local = [];
6  let ftp = [];
7
8  let BALANCE = [];
9  if (local_LENGTH != ftp_LENGTH) {
10     for (var i = 0; i < local_LENGTH; i++) {
11         local.push(msg.payload.local[i]);
12     }
13
14     for (var j = 0; j < ftp_LENGTH; j++) {
15
16         ftp.push(msg.payload.ftp[j].name);
17     }
18
19     for (var k = 0; k < local_LENGTH; k++) {
20         var flag = 0;
21         for (var l = 0; l < ftp_LENGTH; l++) {
22             if (local[k] === ftp[l]) {
23
24                 ftp.splice(l, 1);
25                 l--;
26                 flag = 1;
27             }
28         }
29     }
30
31     if (flag == 0) {
32         var device2 = {};

```

Function node for insert filename and path for ftp in

The output from array comparison function node is again giving to one more function node where we will mention the directory path and filename as a inputs for ftp in node.

```
1 msg.payload.data = "/usr/src/node-red/ftp/" + msg.payload.files_retrived
2 //flow.set("path_test", msg.payload.path);
3 msg.payload.filename = msg.payload.files_retrived
4 return msg;
5
```

ftp in node:

Edit ftp in node

Delete Cancel Done

Properties

Add new FTP Server Add new ftp...

Operation put

Filename Filename

Local Filename Local Filename

Name Name

Set Operation –put

Filename - leave this box blank so it will take filename from the function node connected behind it.

Local Filename-leave this box blank so it will take filename from the function node connected behind it.

Setup the server configuration as per below figure (click the pencil icon)enter the username and password to authorize the server.

Set port always 21 for FTP

Debug node:

The Debug node causes any message to be displayed in the [Debug sidebar](#). By default, it just displays the payload of the message, but it is possible to display the entire message object.

- 3 Click the Deploy button. With the Debug sidebar tab selected,
- 4 Click the Inject button to get output if there is no triggering time set.



