# NODERED+POSTGRESQL WORKFLOW IMPLEMENTATION FOR DIMO DIESEL CONSUMPTION ANALYSIS

MALAVIKA K.V

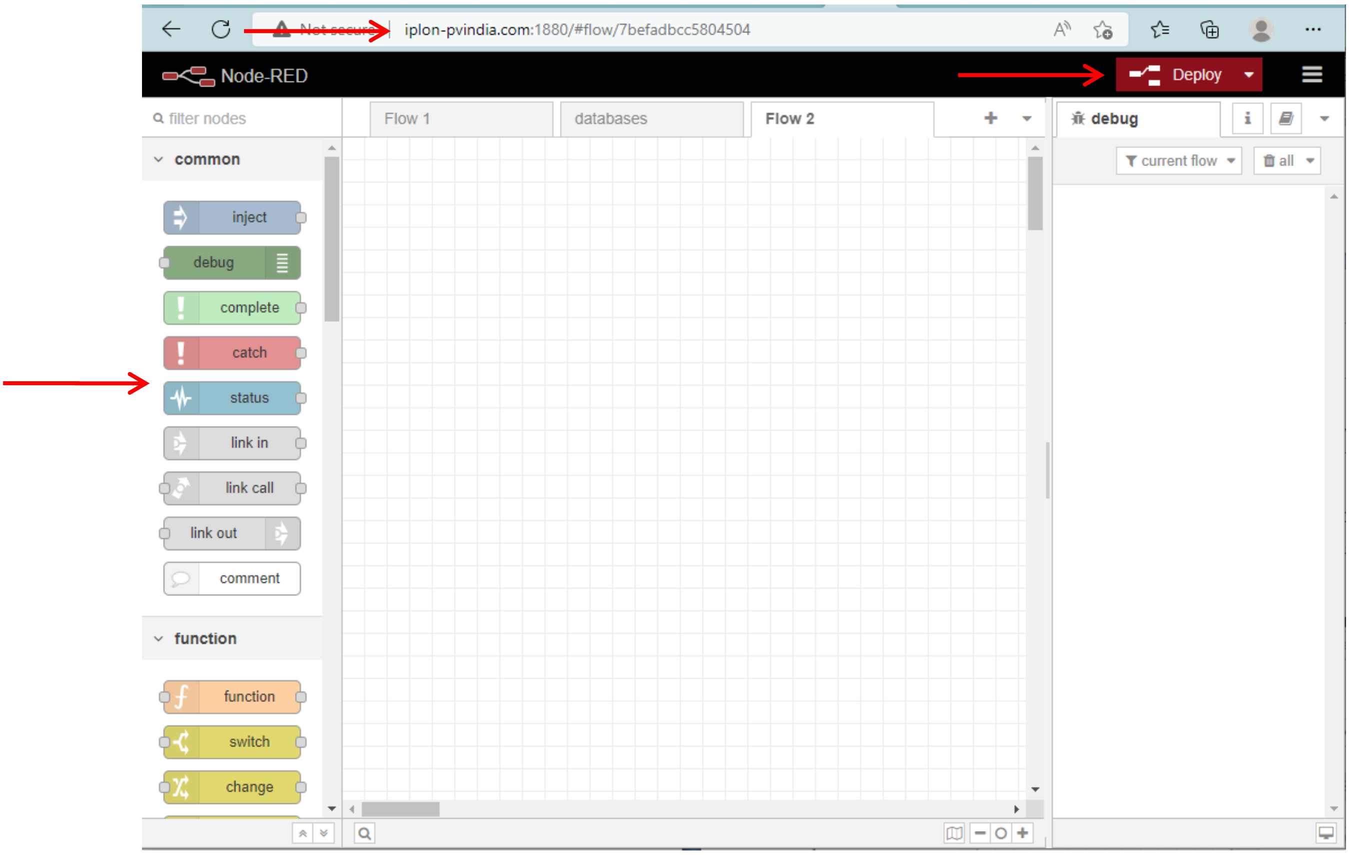
# NODE-RED INSTALLATION SETUP

1. Node-red installed in a docker inside a server iplon-pvindia.com in the port 1880 by using the command inside iplon-pvindia.com server

```
docker run -itd --name nodered_dimo_test --network host nodered/node-red
```

once the node-red docker installation finished we can access node-red in the url http://iplon-pvindia.com:1880/

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.



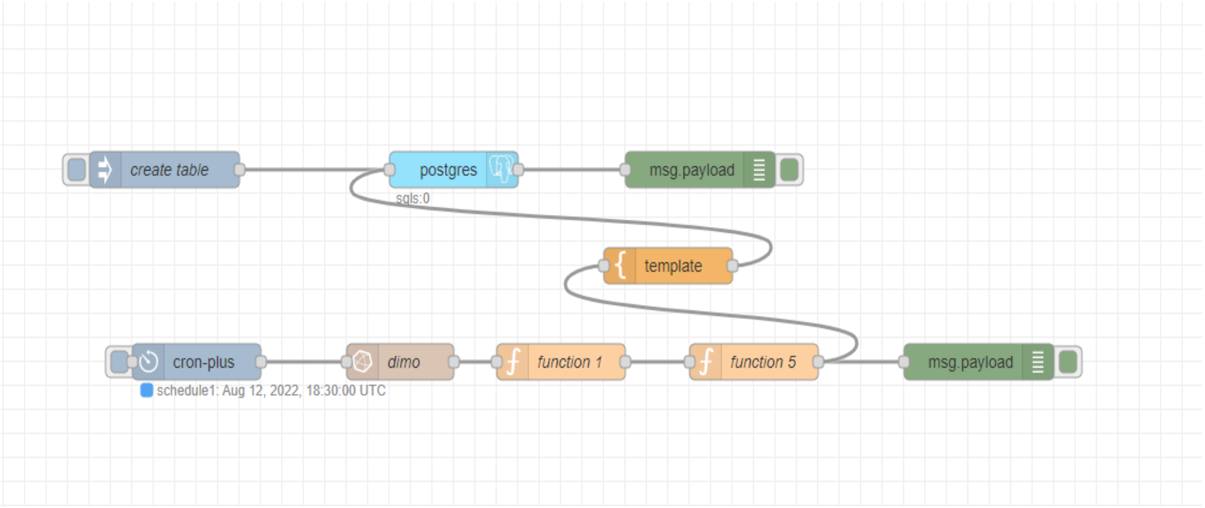
# DIMO DIESEL CONSUMPTION ANALYSIS PROJECT REQUIREMETS

- Node-red flow for creating table inside the postgres which running in iplon-pvindia.com server and one more flow for get data from influxdb dimo database for last 24h and take start time, end time , duration and number of power cuts,DG1,DG2, SOLAR, Total load at that particular time period.
- Tags Needed in the DG Analysis Postgres Table

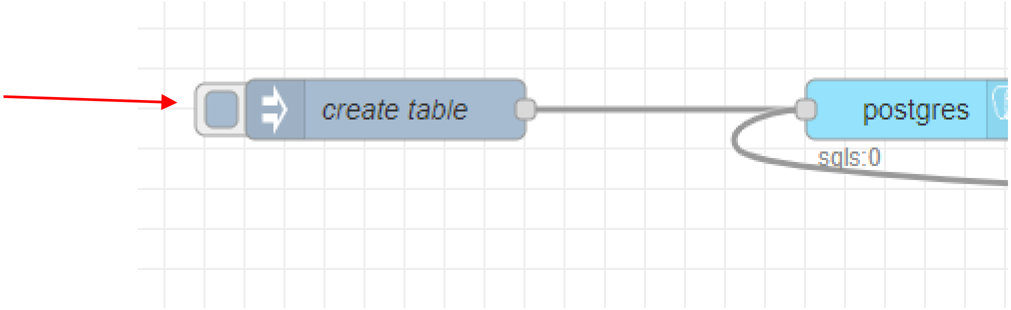
  Date in hr:mm:ss<br>
  2. Grid Down count in No
  3. Grid Down Start Time in Hr:mm:ss
  4. Grid Down End Time in Hr:mm:ss
  5. Grid Down Duration in Hrs
  6. DG1 Generation in kWhr
  7. DG2 to DGn Generation in kWhr
  8. Solar PV generation in kWhr
  9. Total Demand in kWhr (Sum of DG + Solar)

- Visualization of postgres table in Grafana

## NODE-RED FLOW:

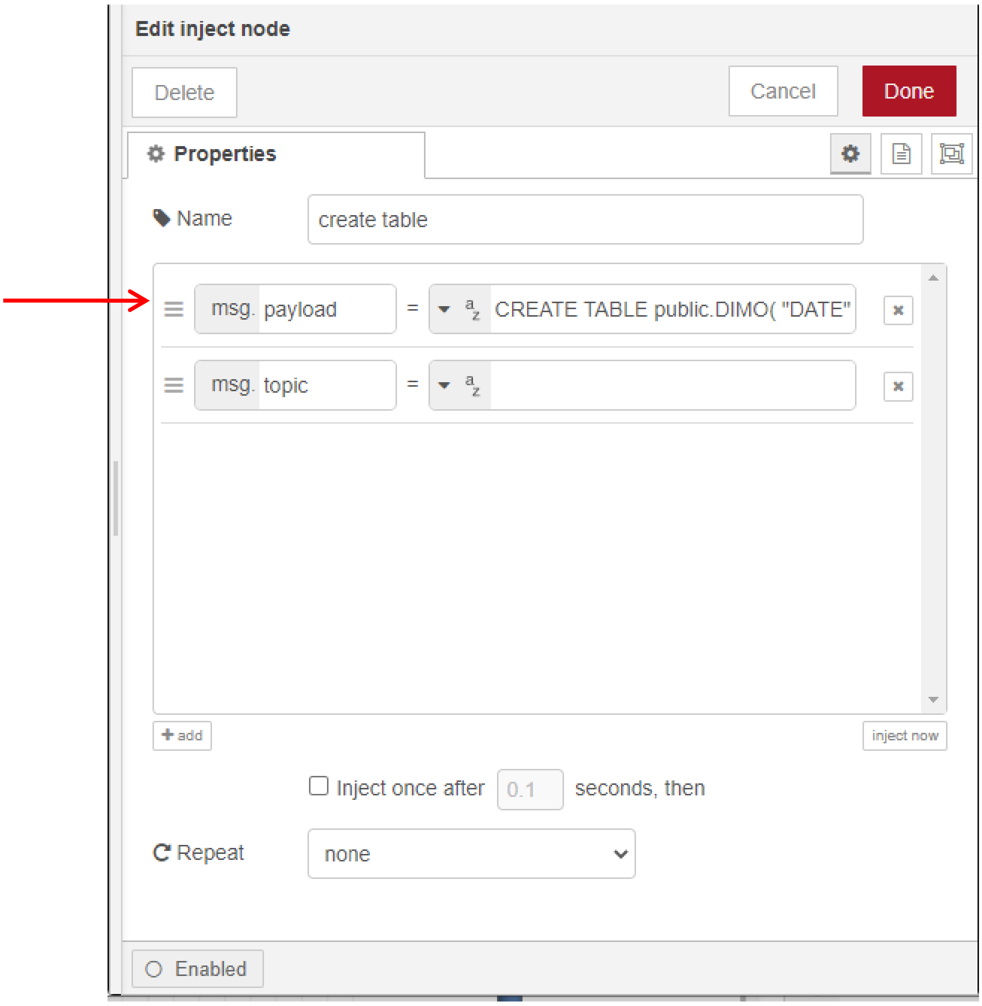


## Inject Node:



1. The Inject 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. Drag one onto the workspace from the palette.
3. Set msg.payload as SQL query to create table inside postgres databse SQL Query: CREATE TABLE public.DIMO( "DATE" numeric PRIMARY KEY, "POWER_CUT" numeric,"START_TIME" text,"END_TIME" text,"Duration (hr)" numeric,"Gen1(kwh)" numeric,"Gen2(kwh)" numeric,"Total_solar(kwh)" numeric,"Total_Load(kwh)" numeric) WITH ( OIDS=FALSE )

   When the inject node trigged it will create a table inside datase



Name: create table

msg. payload = CREATE TABLE public.DIMO( "DATE"

msg. topic =

□ Inject once after 0.1 seconds, then

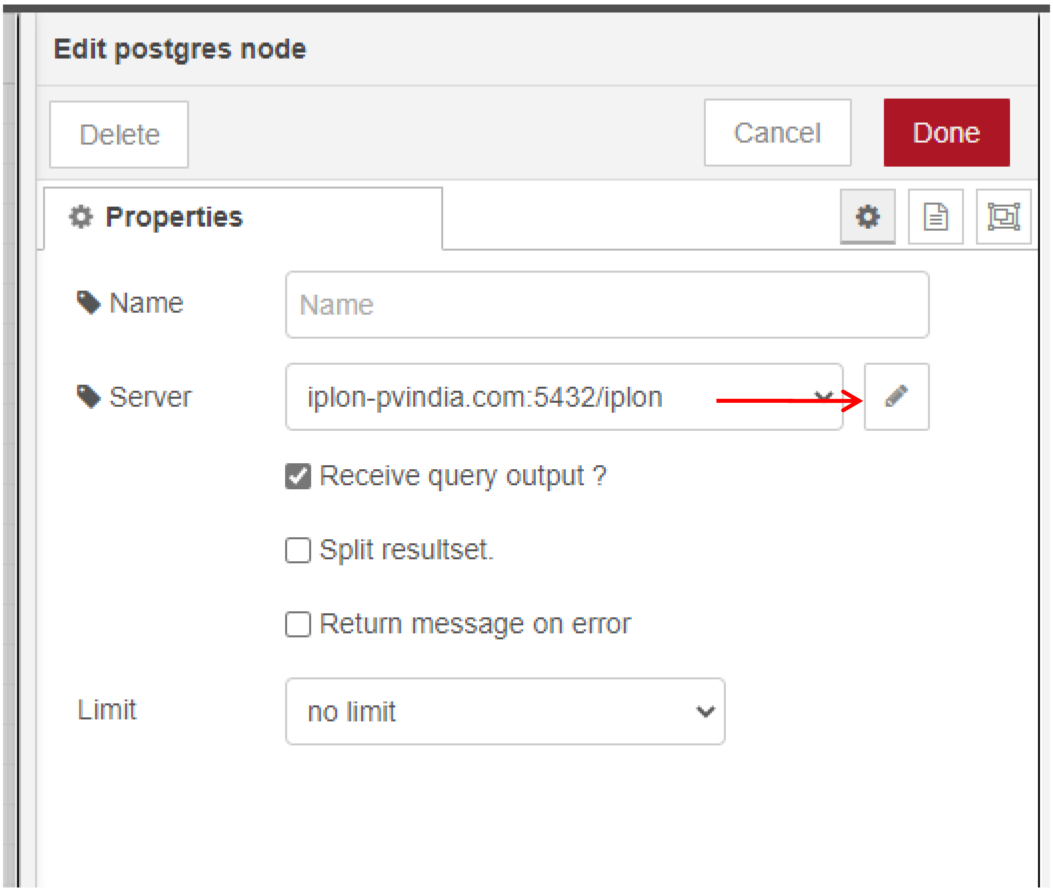
Repeat: none

**Postgres Node** (node-red-contrib-re-postgres (node) - Node-RED (nodered.org)):

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

Node to access postgres database from node-red

double click on the postgres node



Name: Name

Server: iplon-pvindia.com:5432/iplon

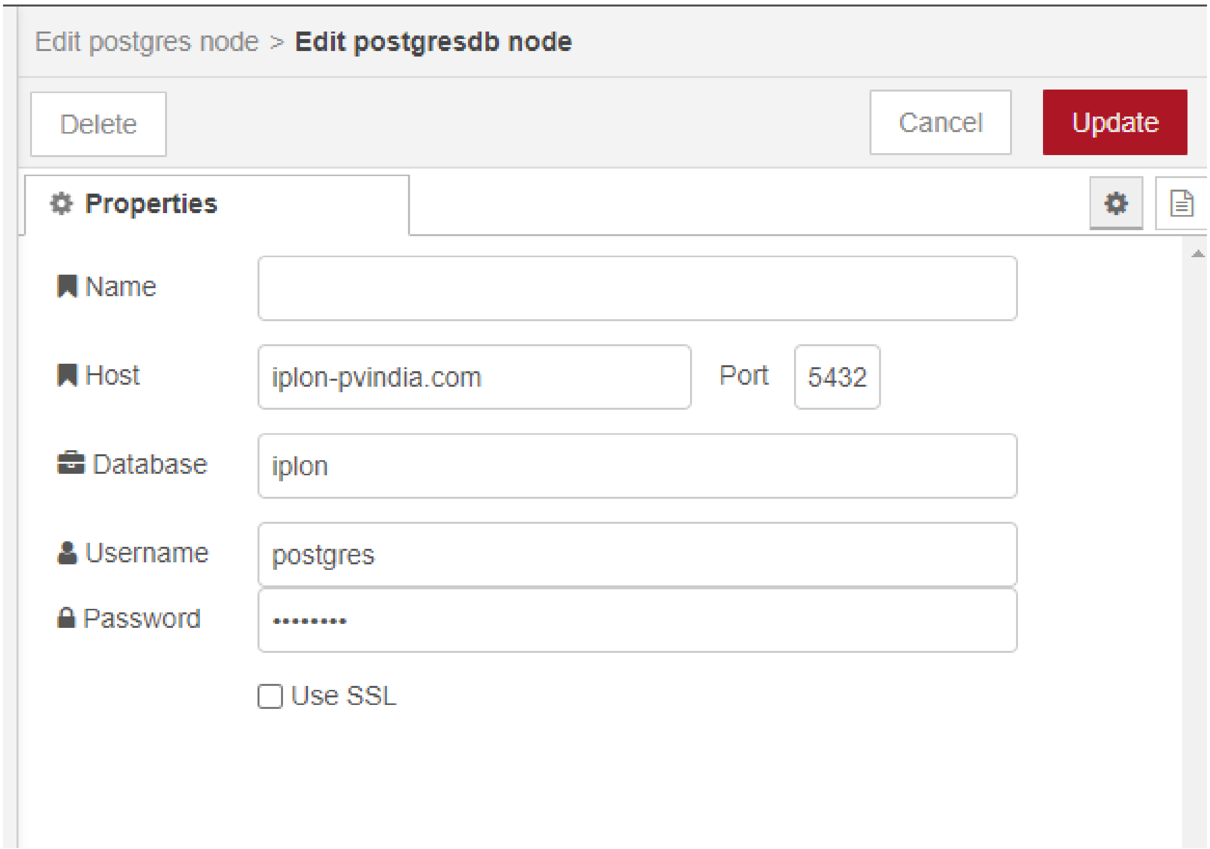
☑ Receive query output ?

☐ Split resultset.

☐ Return message on error

Limit: no limit

Click on the pencil icon to give server configation



Edit postgres node > Edit postgresdb node

Name:

Host: iplon-pvindia.com

Port: 5432

Database: iplon

Username: postgres

Password: ••••••••

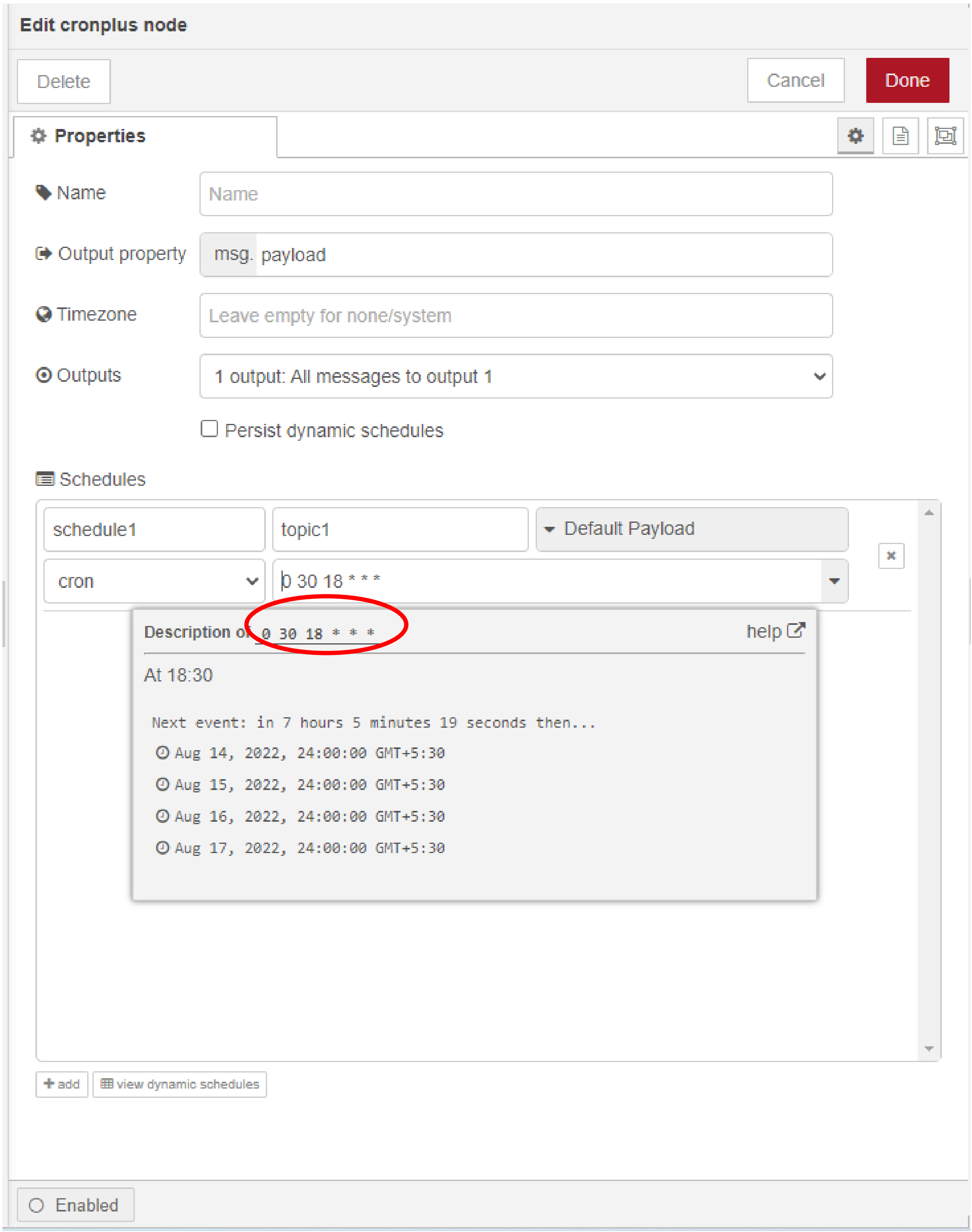
□ Use SSL

Feed host,port,database,username,password in the corresponding boxes to connect the postgres server from node-red

**Cron Plus node:(**[node-red-contrib-cron-plus (node) - Node-RED (nodered.org)](https://flows.nodered.org/node/node-red-contrib-cron-plus))

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

Set cron as 0 30 18 * * * so this will trigger the flow everyday 18.30 (UTC Time)ie, 12AM(IST)



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 | Default Payload

cron | 0 30 18 * * *

Description of 0 30 18 * * * help

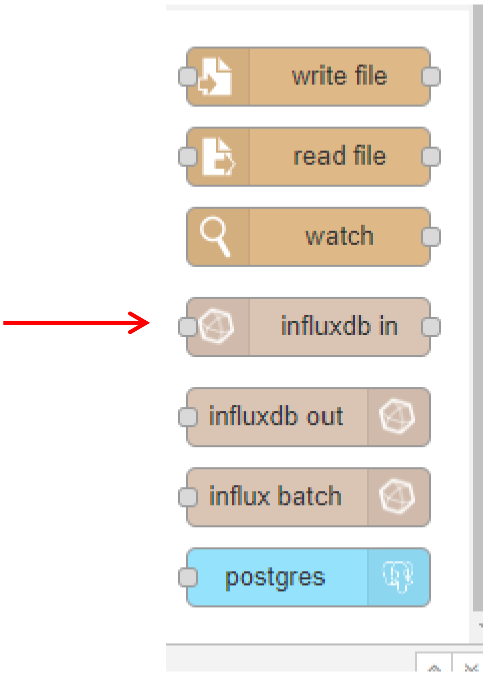
At 18:30

```
Next event: in 7 hours 5 minutes 19 seconds then...
Aug 14, 2022, 24:00:00 GMT+5:30
Aug 15, 2022, 24:00:00 GMT+5:30
Aug 16, 2022, 24:00:00 GMT+5:30
Aug 17, 2022, 24:00:00 GMT+5:30
```

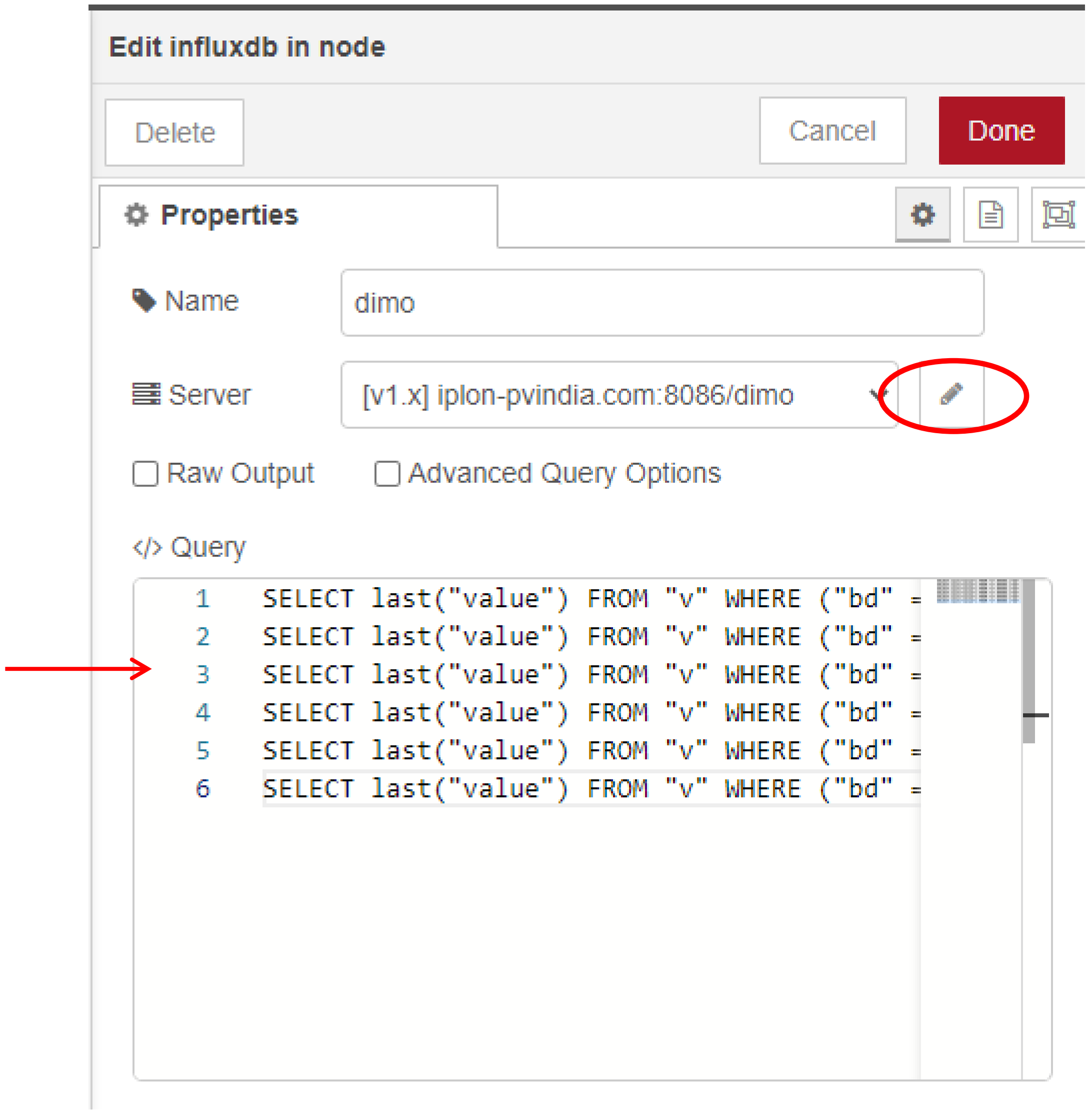
**Influxdb in node** (node-red-contrib-influxdb (node) - Node-RED (nodered.org)):

To access this node we have to install the node package(node-red-contrib-influxdb)from manage pallette 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.



Drag&Drop this node to workspace double click on the influx db node



Edit influxdb in node

Name: dimo

Server: [v1.x] iplon-pvindia.com:8086/dimo

□ Raw Output □ Advanced Query Options

Query

```
SELECT last("value") FROM "v" WHERE ("bd" =
SELECT last("value") FROM "v" WHERE ("bd" =
SELECT last("value") FROM "v" WHERE ("bd" =
SELECT last("value") FROM "v" WHERE ("bd" =
SELECT last("value") FROM "v" WHERE ("bd" =
SELECT last("value") FROM "v" WHERE ("bd" =
```

Write the influx query in the query box given

SELECT last("value") FROM "v" WHERE ("bd" = 'dimo_700W' AND "d" = 'GRID_EM' AND "f" = 'UAC12') AND time >= now() - 1d GROUP BY time(1m), "d", "f" fill(none);

SELECT last("value") FROM "v" WHERE ("bd" = 'dimo_700W' AND "d" = 'GRID_EM' AND "f" = 'UAC23') AND time >= now() - 1d GROUP BY time(1m), "d", "f" fill(none);

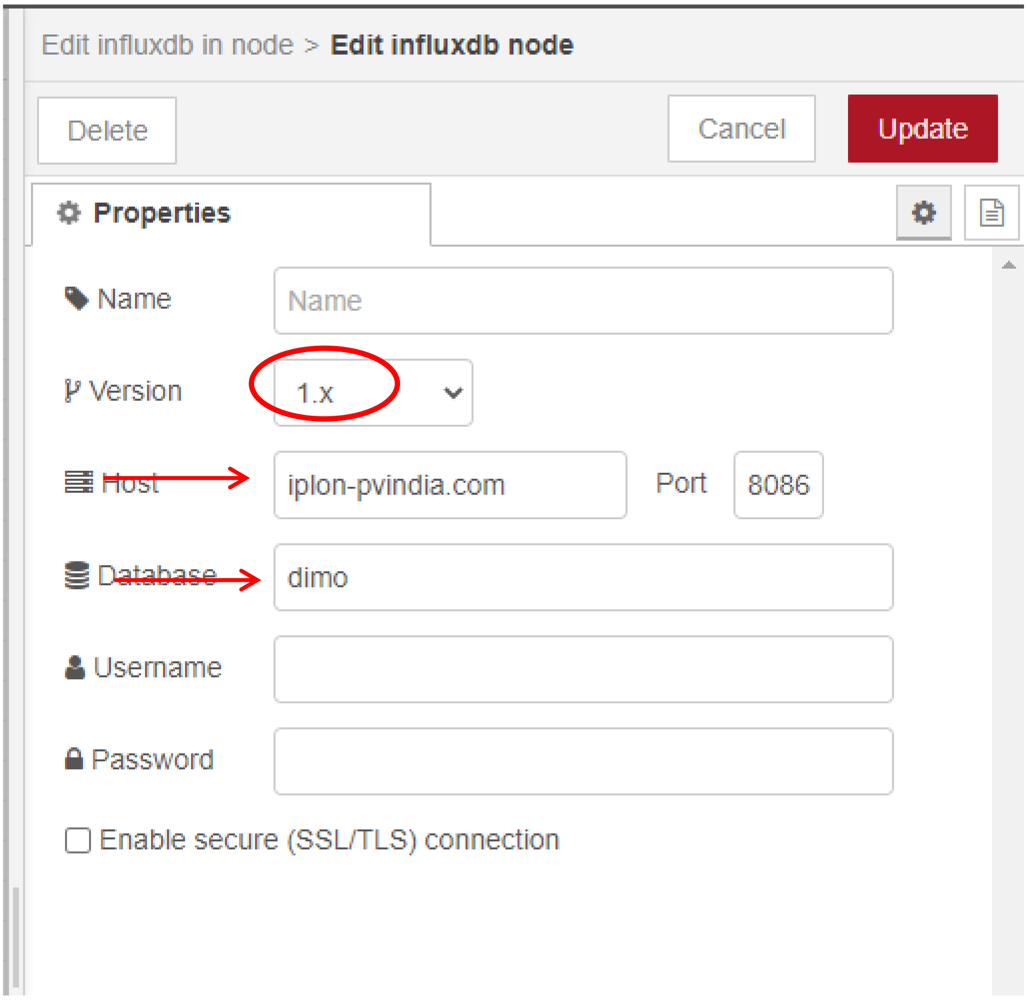
SELECT last("value") FROM "v" WHERE ("bd" = 'dimo_700W' AND "d" = 'GRID_EM' AND "f" = 'UAC31') AND time >= now() - 1d GROUP BY time(1m), "d", "f" fill(none);

SELECT last("value") FROM "v" WHERE ("bd" = 'dimo_700W' AND "d" = 'DG2' AND "f" = 'EAE') AND time >= now() - 1d GROUP BY time(1m), "d", "f" fill(none);

SELECT last("value") FROM "v" WHERE ("bd" = 'dimo_700W' AND "d" = 'DG1' AND "f" = 'EAE') AND time >= now() - 1d GROUP BY time(1m), "d", "f" fill(none);

SELECT last("value") FROM "v" WHERE ("bd" = 'dimo_700W' AND "d" = 'SOLAR_EM' AND "f" = 'EAE') AND time >= now() - 1d GROUP BY time(1m), "d", "f" fill(none)

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



Name: Name

Version: 1.x

Host: iplon-pvindia.com Port: 8086

Database: dimo

Username:

Password:

☐ Enable secure (SSL/TLS) connection

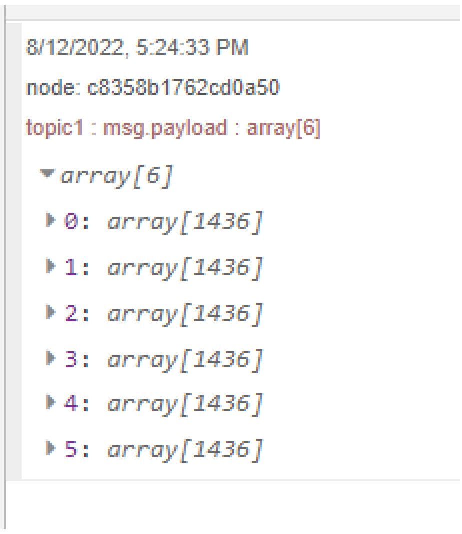
Version =1.x

Host = iplon-pvindia.com

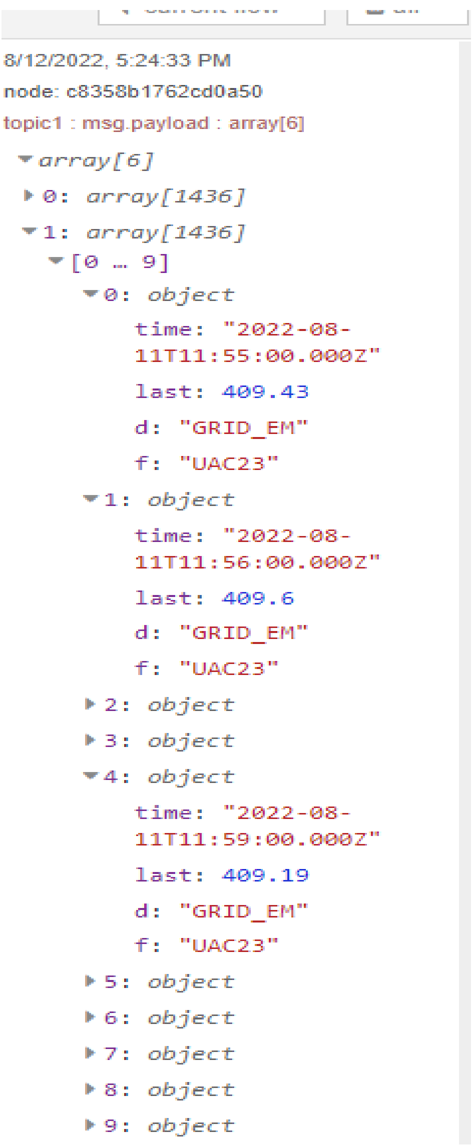
Port = 8086

Database = dimo(use the same database name used in influx db)

## Influx db output:

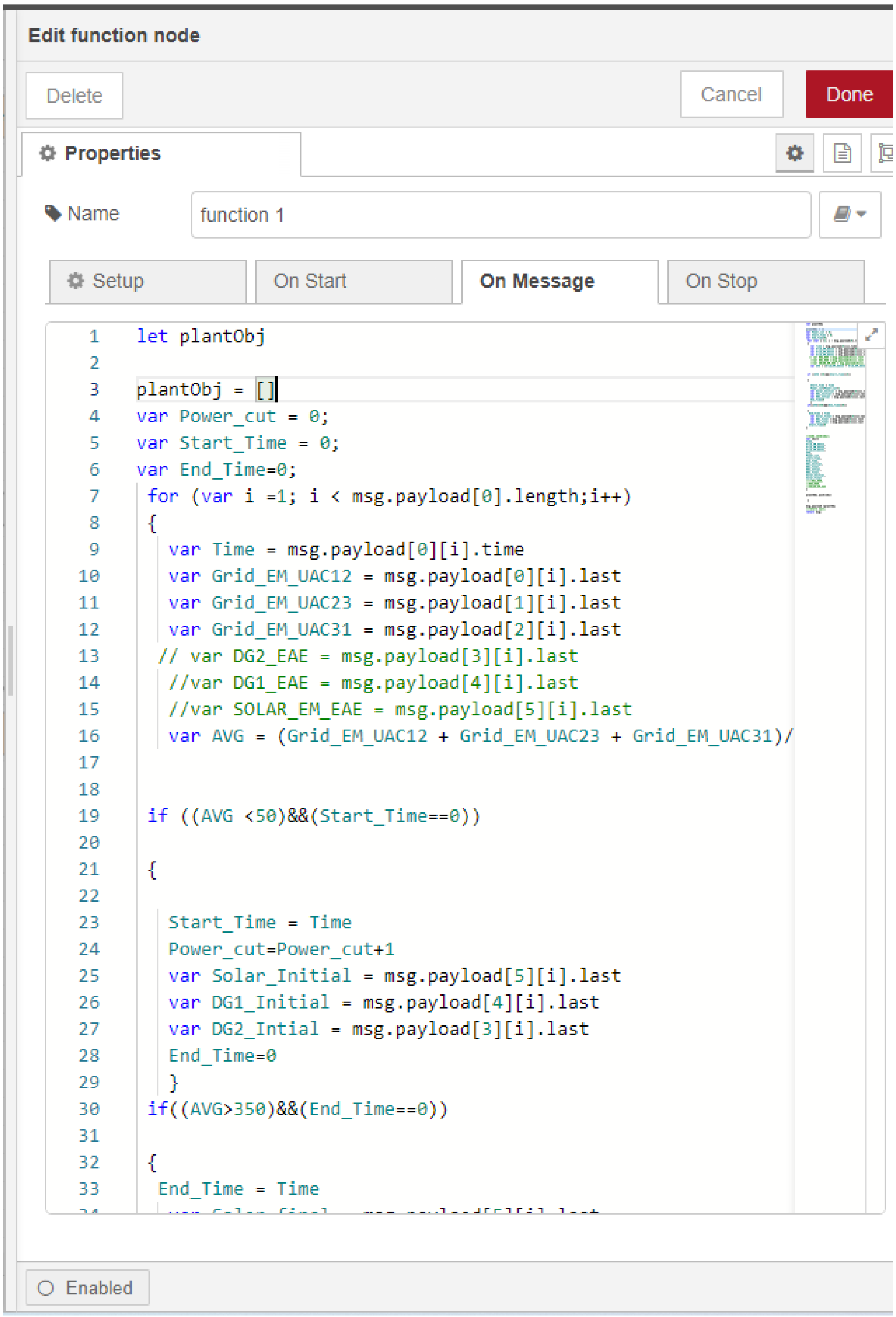


```
8/12/2022, 5:24:33 PM
node: c8358b1762cd0a50
topic1 : msg.payload : array[6]
 ▾array[6]
  ▸0: array[1436]
  ▸1: array[1436]
  ▸2: array[1436]
  ▸3: array[1436]
  ▸4: array[1436]
  ▸5: array[1436]
```



```
8/12/2022, 5:24:33 PM
node: c8358b1762cd0a50
topic1 : msg.payload : array[6]
 ▾array[6]
  ▸0: array[1436]
  ▾1: array[1436]
    ▾[0 … 9]
      ▾0: object
         time: "2022-08-
         11T11:55:00.000Z"
         last: 409.43
         d: "GRID_EM"
         f: "UAC23"
      ▾1: object
         time: "2022-08-
         11T11:56:00.000Z"
         last: 409.6
         d: "GRID_EM"
         f: "UAC23"
      ▸2: object
      ▸3: object
      ▾4: object
         time: "2022-08-
         11T11:59:00.000Z"
         last: 409.19
         d: "GRID_EM"
         f: "UAC23"
      ▸5: object
      ▸6: object
      ▸7: object
      ▸8: object
      ▸9: object
```

## Function node:



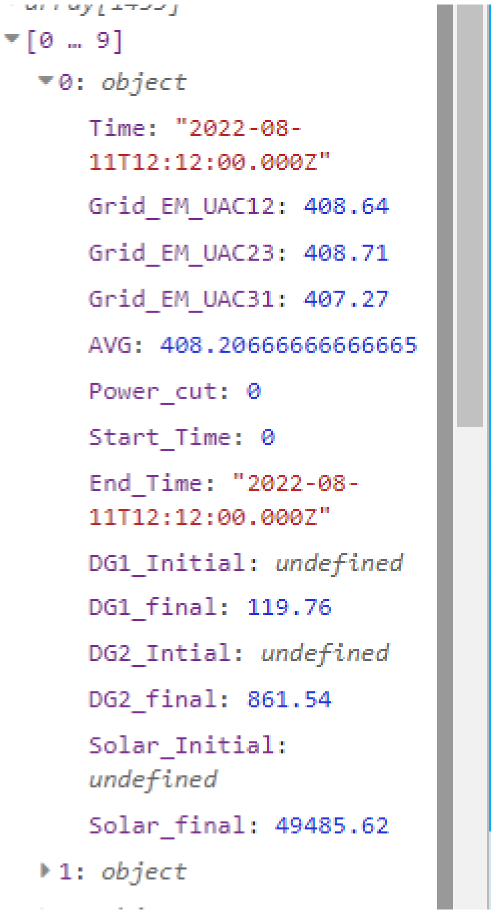
Name: function 1

```
let plantObj

plantObj = []
var Power_cut = 0;
var Start_Time = 0;
var End_Time=0;
 for (var i =1; i < msg.payload[0].length;i++)
 {
   var Time = msg.payload[0][i].time
   var Grid_EM_UAC12 = msg.payload[0][i].last
   var Grid_EM_UAC23 = msg.payload[1][i].last
   var Grid_EM_UAC31 = msg.payload[2][i].last
  // var DG2_EAE = msg.payload[3][i].last
   //var DG1_EAE = msg.payload[4][i].last
   //var SOLAR_EM_EAE = msg.payload[5][i].last
   var AVG = (Grid_EM_UAC12 + Grid_EM_UAC23 + Grid_EM_UAC31)/


 if ((AVG <50)&&(Start_Time==0))

 {

   Start_Time = Time
   Power_cut=Power_cut+1
   var Solar_Initial = msg.payload[5][i].last
   var DG1_Initial = msg.payload[4][i].last
   var DG2_Intial = msg.payload[3][i].last
   End_Time=0
   }
 if((AVG>350)&&(End_Time==0))

 {
  End_Time = Time
   var Solar_final = msg.payload[5][i].last
```

In this function node i am taking input from influx db defining array and creating for loop for iterate each array values from input  defining msg paths and calculating average of grid diving the sum of grid values by 3

After that setting if condition to get start and end timing of DG's operation if  avg<50 it will start count the start timing intial DG values if avg>350 it will start count end timing and final DG value

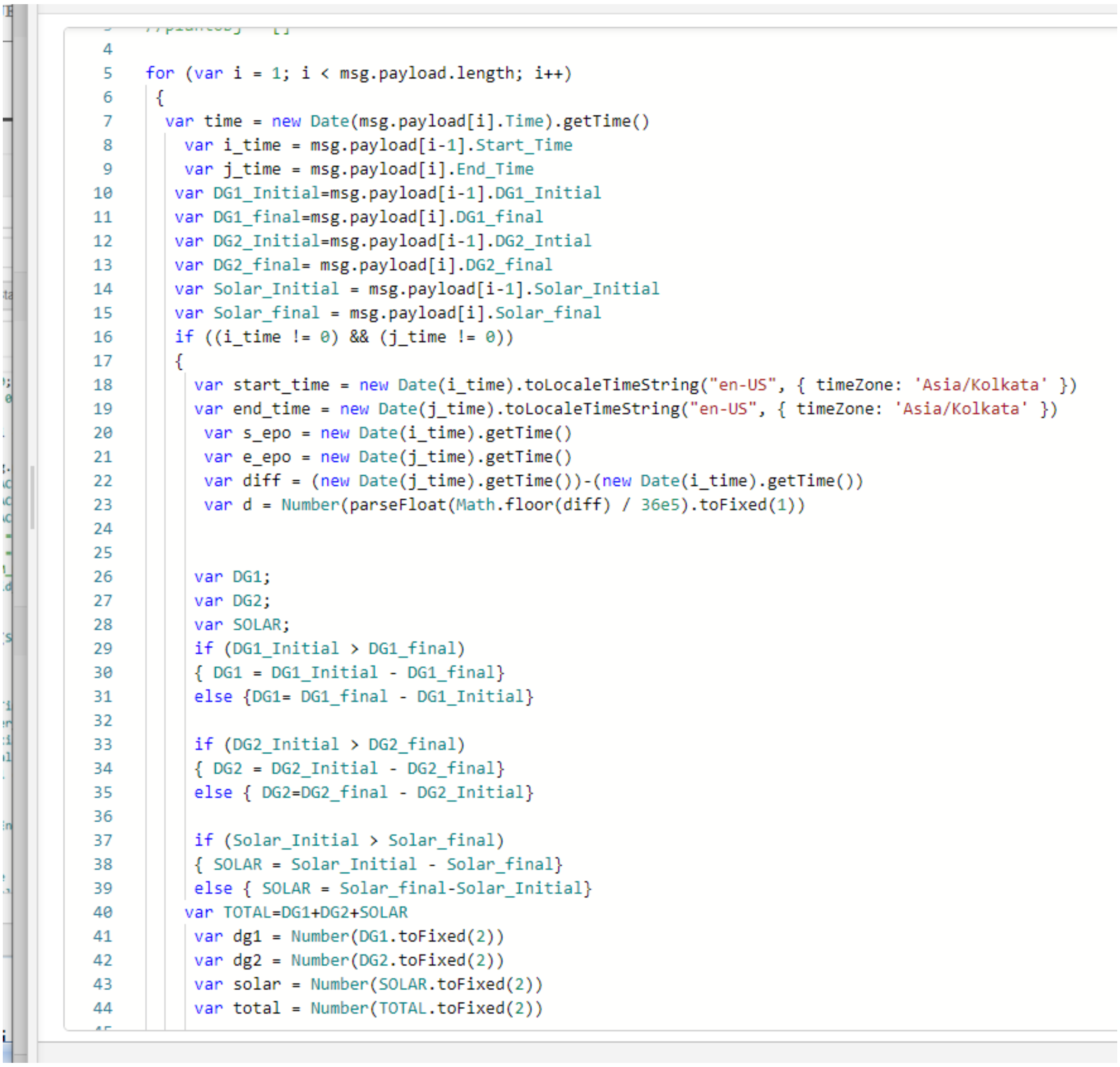
Function output:



```
array[1435]
▾[0 … 9]
  ▾0: object
      Time: "2022-08-
      11T12:12:00.000Z"
      Grid_EM_UAC12: 408.64
      Grid_EM_UAC23: 408.71
      Grid_EM_UAC31: 407.27
      AVG: 408.20666666666665
      Power_cut: 0
      Start_Time: 0
      End_Time: "2022-08-
      11T12:12:00.000Z"
      DG1_Initial: undefined
      DG1_final: 119.76
      DG2_Intial: undefined
      DG2_final: 861.54
      Solar_Initial:
      undefined
      Solar_final: 49485.62
  ▸1: object
```

**Function Node:**

First function output i am giving to another function to filter out start time and end times and calculate duration time DG Loads,solar loads, total loads during the powercut period.

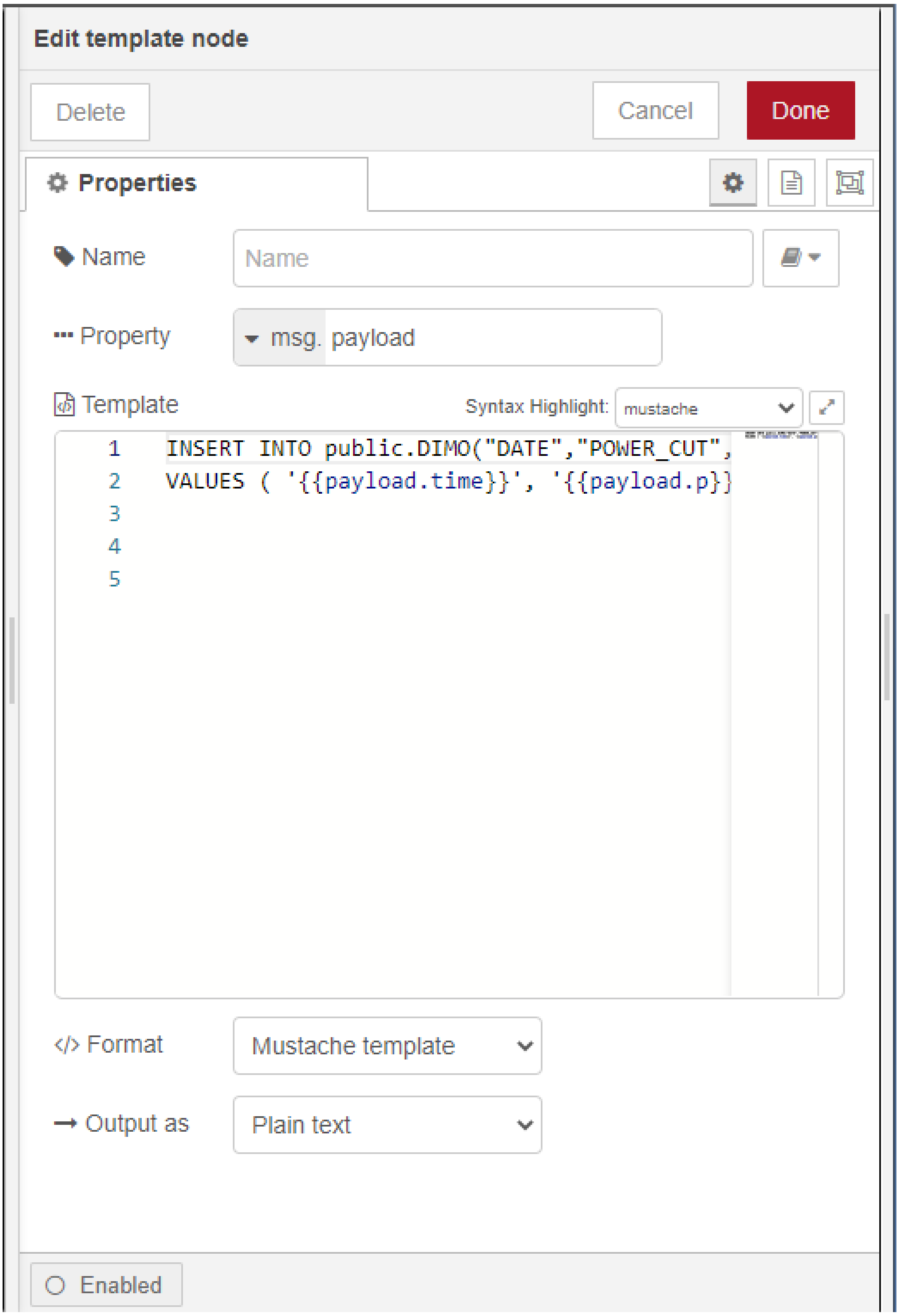


```
//plantobj = []

for (var i = 1; i < msg.payload.length; i++)
 {
  var time = new Date(msg.payload[i].Time).getTime()
    var i_time = msg.payload[i-1].Start_Time
    var j_time = msg.payload[i].End_Time
   var DG1_Initial=msg.payload[i-1].DG1_Initial
   var DG1_final=msg.payload[i].DG1_final
   var DG2_Initial=msg.payload[i-1].DG2_Intial
   var DG2_final= msg.payload[i].DG2_final
   var Solar_Initial = msg.payload[i-1].Solar_Initial
   var Solar_final = msg.payload[i].Solar_final
   if ((i_time != 0) && (j_time != 0))
   {
     var start_time = new Date(i_time).toLocaleTimeString("en-US", { timeZone: 'Asia/Kolkata' })
     var end_time = new Date(j_time).toLocaleTimeString("en-US", { timeZone: 'Asia/Kolkata' })
      var s_epo = new Date(i_time).getTime()
      var e_epo = new Date(j_time).getTime()
      var diff = (new Date(j_time).getTime())-(new Date(i_time).getTime())
      var d = Number(parseFloat(Math.floor(diff) / 36e5).toFixed(1))


     var DG1;
     var DG2;
     var SOLAR;
     if (DG1_Initial > DG1_final)
     { DG1 = DG1_Initial - DG1_final}
     else {DG1= DG1_final - DG1_Initial}

     if (DG2_Initial > DG2_final)
     { DG2 = DG2_Initial - DG2_final}
     else { DG2=DG2_final - DG2_Initial}

     if (Solar_Initial > Solar_final)
     { SOLAR = Solar_Initial - Solar_final}
     else { SOLAR = Solar_final-Solar_Initial}
    var TOTAL=DG1+DG2+SOLAR
     var dg1 = Number(DG1.toFixed(2))
     var dg2 = Number(DG2.toFixed(2))
     var solar = Number(SOLAR.toFixed(2))
     var total = Number(TOTAL.toFixed(2))
```

## Template Node:

Node connected between last function node and postgres node

Template node used to insert the values to postgres table here using the mustache syntax here we have to mention the table name and column names and insert the value message path in the corresponding order of column name given in INSERT INTO query



Name: Name

Property: msg. payload

Template — Syntax Highlight: mustache

```
INSERT INTO public.DIMO("DATE","POWER_CUT",
VALUES ( '{{payload.time}}', '{{payload.p}}
```

Format: Mustache template

Output as: Plain text

○ Enabled

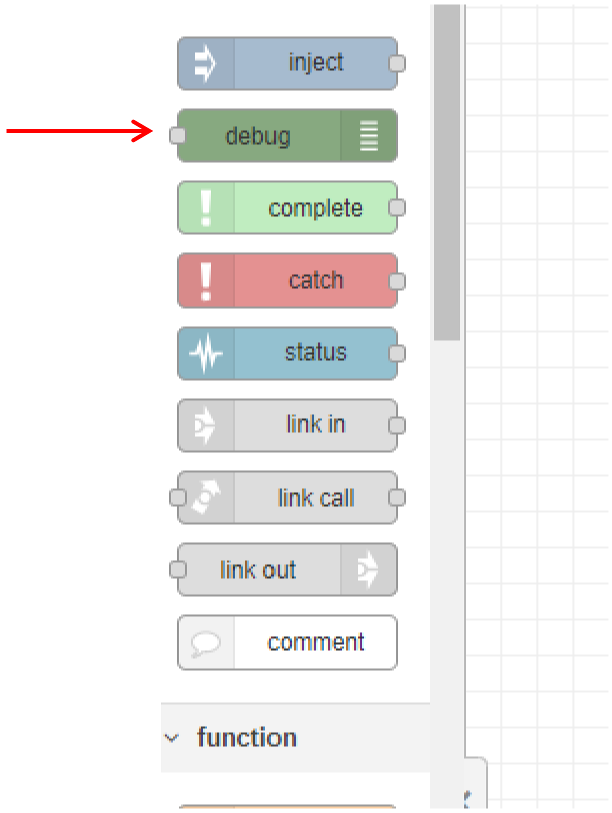
**INSERT VALUES QUERY**

Mention column name same as how you gave at the time table creation

Give the value message path in the corresponding order of column name

```
INSERT INTO public.DIMO("DATE","POWER_CUT","START_TIME", "END_TIME",,
"Duration (hr)", "Gen1(kwh)", "Gen2(kwh)", "Total_solar(kwh)",
"Total_Load(kwh)")

VALUES ( '{{payload.time}}', '{{payload.p}}', '{{payload.start_time}}',
'{{payload.end_time}}', '{{payload.d}}', '{{payload.dg1}}', '{{payload.dg2}}',
'{{payload.solar}}', '{{payload.total}}')
```

## Debug node:



The Debug node causes any message to be displayed in the 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 triggerng time set.
3. Check the destination of file what we give as path to check whether the csv file created or not