

# The Vinoblock

## Connect and run the costaflores global vineyard node

First of all we'll create a new directory in the /home/pi/ path to allocate all the files related to the node:

```
pi$ mkdir node && cd node
```

Once we are in the node directory we create an ethereum account using the geth client and following the instructions:

```
pi$ geth account new --datadir .  
  
...  
WARN [07-26|12:46:35.692] Sanitizing cache to Go's GC limits      provided=1024 updated=308  
INFO [07-26|12:46:35.698] Maximum peer count      ETH=25 LES=0 total=25  
Your new account is locked with a password. Please give a password. Do not forget this password.  
Passphrase: ****  
Repeat passphrase: ****  
Address: {5f68298dd10e8ea45d64d973c8d90c3b60215e64}  
  
...
```

In our case, the public address of the account generated is 5f68298dd10e8ea45d64d973c8d90c3b60215e64, it is important to copy this address somewhere because it will be used in the future to set up the running node. Once we have our account created we can initialise our node, to do so we will need the genesis file which will be provided by the openvino admin. Once we have it:

```
computer$ scp genesis.json pi@"ip of your raspberry":/home/pi/node/
```

To initialise the geth node we have to:

```
pi$ geth init /home/pi/node/genesis.json --datadir /home/pi/node/
```

Once the node is initialised have to create a file with the description of the nodes it is going to be paired with. In our case, this file will be, once again, provided by the admins, so:

```
computer$ scp static-nodes.json pi@"ip of your raspberry":/home/pi/node/
```

Now that we have all the necessary files, we can proceed to create and script that runs the node.

```
pi$ nano run.sh
```

### Run.sh

```
nohup /usr/local/bin/geth --identity cgV-test --networkid ****[1] --datadir  
/home/pi/node --rpc --rpcaddr 127.0.0.1 --rpcport "7547" --rpccorsdomain  
"*" --nat "any" --rpcapi db,eth,web3,personal,net,miner,admin --port 30303  
--verbosity 3 --syncmode fast --ethstats  
cgV-test:****[2]?@10.112.48.25:3000 >> /home/pi/logs/geth.log &
```

*(The missing information (1 and 2) will be given by the admins as they are private for our network)*

```
pi$ chmod +x run.sh
```

Finally when this file is created we can check it by using the following command, I personally recommend to run this command during the first full synchronisation to check if it is working properly:

```
pi$ ./run.sh
```

Once the node has already synchronised we can set up a systemctl service that runs on boot creating a ethereum-node.service file in /etc/systemd/system/ with this content:

```
pi$ nano /etc/systemd/system/ethereum-node.service
```

```
ethereum-node.service

[Unit]

Description=Costaflores Global Vineyard Node

Wants=network-online.target

After=network.target

[Service]

Type=forking

ExecStart=/home/pi/node/run.sh

ExecStop=/usr/bin/killall -9 geth

[Install]

WantedBy=multi-user.target
```

Finally now we just have to reload the daemon list and to test and enable it to run on boot using this commands:

```
pi$ sudo systemctl daemon-reload
```

```
pi$ sudo systemctl start ethereum-node
```

```
pi$ sudo systemctl enable ethereum-node
```

## Download Vinoblock package

To ease the installation we've packed some files in a github repository for you to download all the source files. To download and place it all in a proper hierarchy you can:

```
pi$ git clone https://github.com/jestape/vinoblock-package
```

```
pi$ mv /home/pi/vinoblock/db /home/pi/  
pi$ mv /home/pi/vinoblock/app /home/pi/  
pi$ rm -rf /home/pi/vinoblock
```

## Create database tables and events

We'll now create the local database that will store a 1 day copy of the database so that it can be used for displaying purposes. This database will be formed by a basic table containing information gathered frequently and a table containing it's daily information. To create this tables we will:

```
pi$ sudo mysql -u root -p  
  
pi$ MariaDB [(none)] >> CREATE DATABASE db;  
pi$ MariaDB [(none)] >> USE db;  
pi$ MariaDB [(db)] >> CREATE USER 'system' IDENTIFIED BY 'raspberrypi';  
pi$ MariaDB [(db)] >> GRANT ALL PRIVILEGES ON db.* TO 'system'@'%' WITH GRANT OPTION;  
pi$ MariaDB [(db)] >> source /home/pi/db/tables.sql
```

Once this commands have been executed the tables will be created and a user named system too. Remember that the user is being created with the password 'raspberrypi' for testing purposes, in case you want to change it you can modify user creation command to fit your password.

## Downloading and executing the app

With the database and the blockchain node working we have accomplished all the prerequisites to run the background app that will display data through the lcd 16x02 display. The python code packages for this purpose has been uploaded to: <https://github.com/jestape/vinoblock-package>. So we'll just have to:

```
pi$ python3 app/main.py
```

*(In case you have changed the system database user password you must modify the password parameter of line 10 in main.py)*

Great! If you have connected everything properly your lcd 16x02 display should giving you some cool data to play with. It will be also sending this data to the costaflores private blockchain for auditing purposes and storing it in the database so that you can monitor it. So, now let's make it work on boot:

```
pi$ nano /etc/systemd/system/background-app.service
```

## app.service

[Unit]

Description=Background App

Wants=network-online.target

After=network.target

[Service]

Type=simple

ExecStart=/usr/bin/python3 /home/pi/app/main.py &

[Install]

WantedBy=multi-user.target

Finally now we just have to reload the daemon list and to test and enable it to run on boot using this commands:

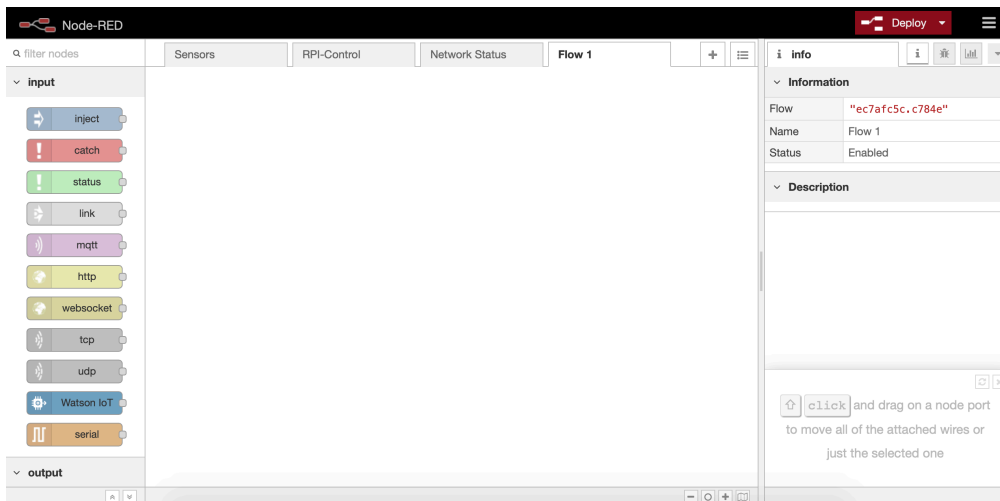
```
pi$ sudo systemctl daemon-reload
```

```
pi$ sudo systemctl start background-app
```

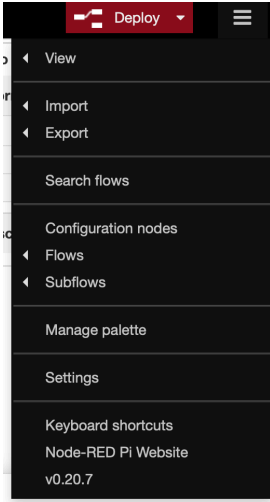
```
pi$ sudo systemctl enable background-app
```

## Initialise and implement nodered front-end

Finally we have to set up our front-end web page that will allow us to see the dashboard of the data collected and control the raspberry pi via UI. This will be done via web browser so firstly you'll have to open a web browser and search: "http://ip of your raspberry":1880. The following screen will appear:



Firstly we'll install some node-red extra packages that our platform is going to use, to do so we'll have to open the right top menu button and click onto the manage palette option:

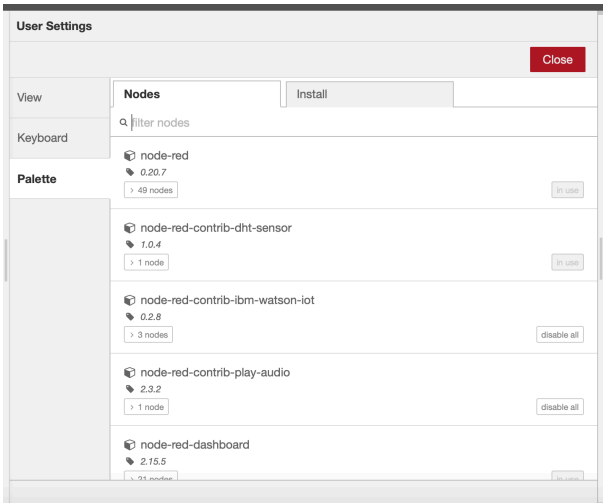


A pop up screen will appear, there we'll have to go to the tab install and search and install the following packages:

`node-red-dashboard`

`node-red-contrib-dht-sensor`

`node-red-node-mysql`



We recommend you to wait for several minutes for packages to get installed. Once this is done we have to open the menu again, the import option and inside it the clipboard one. This will display the following screens/popup:

**Import nodes**

Paste flow json or

```
[
  {
    "wires": [
      []
    ]
  }
]
```

Import to

In this screen we have to copy paste the following code:

```
flow.json
[
  {
    "id": "d360b8b7.05523",
    "type": "tab",
    "label": "Sensors",
    "disabled": false,
    "info": ""
  },
  {
    "id": "67d6f99e.a13658",
    "type": "tab",
    "label": "RPI-Control",
    "disabled": false,
    "info": ""
  },
  {
    "id": "4acb4e01.568548",
    "type": "tab",
    "label": "Network Status",
    "disabled": false,
    "info": ""
  },
  {
    "id": "a90eb261.8b188",
    "type": "ui_group",
    "z": "",
    "name": "Current data",
    "tab": "a2bde7d.6a63f18",
    "order": 1,
    "disp": true,
    "width": "14",
    "collapse": false
  },
],
```

```

{
  "id": "alaad443.4d5388",
  "type": "ui_group",
  "z": "",
  "name": "Today data",
  "tab": "a2bde7d.6a63f18",
  "order": 4,
  "disp": true,
  "width": "22",
  "collapse": false
},
{
  "id": "a2bde7d.6a63f18",
  "type": "ui_tab",
  "z": "",
  "name": "Costaflores Global Vineyard",
  "icon": "dashboard",
  "disabled": false,
  "hidden": false
},
{
  "id": "afa334c1.2b7cc",
  "type": "ui_base",
  "theme": {
    "name": "theme-dark",
    "lightTheme": {
      "default": "#0094CE",
      "baseColor": "#0094CE",
      "baseFont": "-apple-system,BlinkMacSystemFont,Segoe
UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif",
      "edited": true,
      "reset": false
    },
    "darkTheme": {
      "default": "#097479",
      "baseColor": "#8912a8",
      "baseFont": "-apple-system,BlinkMacSystemFont,Segoe
UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif",
      "edited": true,
      "reset": false
    },
    "customTheme": {
      "name": "Untitled Theme 1",
      "default": "#4B7930",
      "baseColor": "#4B7930",
      "baseFont": "-apple-system,BlinkMacSystemFont,Segoe
UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"
    },
    "themeState": {
      "base-color": {
        "default": "#097479",
        "value": "#8912a8",
        "edited": true
      }
    }
  }
}

```

```

    },
    "page-titlebar-backgroundColor": {
      "value": "#8912a8",
      "edited": false
    },
    "page-backgroundColor": {
      "value": "#111111",
      "edited": false
    },
    "page-sidebar-backgroundColor": {
      "value": "#000000",
      "edited": false
    },
    "group-textColor": {
      "value": "#belfe7",
      "edited": false
    },
    "group-borderColor": {
      "value": "#555555",
      "edited": false
    },
    "group-backgroundColor": {
      "value": "#333333",
      "edited": false
    },
    "widget-textColor": {
      "value": "#eeeeee",
      "edited": false
    },
    "widget-backgroundColor": {
      "value": "#8912a8",
      "edited": false
    },
    "widget-borderColor": {
      "value": "#333333",
      "edited": false
    },
    "base-font": {
      "value": "-apple-system,BlinkMacSystemFont,Segoe
UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"
    }
  },
  "angularTheme": {
    "primary": "indigo",
    "accents": "blue",
    "warn": "red",
    "background": "grey"
  }
},
"site": {
  "name": "Costaflores Global Vineyard",
  "hideToolbar": "false",
  "allowSwipe": "false",

```



```
        "lockMenu": "false",
        "allowTempTheme": "true",
        "dateFormat": "DD/MM/YYYY",
        "sizes": {
            "sx": 48,
            "sy": 48,
            "gx": 6,
            "gy": 6,
            "cx": 6,
            "cy": 6,
            "px": 2,
            "py": 2
        }
    },
    {
        "id": "b7df6311.fca09",
        "type": "MySQLdatabase",
        "z": "",
        "host": "127.0.0.1",
        "port": "3306",
        "db": "db",
        "tz": ""
    },
    {
        "id": "f96f18e5.0f1a1",
        "type": "mqtt-broker",
        "broker": "50.23.106.210",
        "port": "1883",
        "clientid": ""
    },
    {
        "id": "bb9db436.f6ce",
        "type": "ui_group",
        "z": "",
        "name": "Charts",
        "tab": "",
        "order": 2,
        "disp": false,
        "width": "6"
    },
    {
        "id": "dcc1457f.4c68a",
        "type": "ui_group",
        "z": "",
        "name": "Inputs",
        "tab": "",
        "order": 1,
        "disp": false,
        "width": "6"
    },
    {
        "id": "df7c8056.e67bb8",
```

```
"type": "ui_group",
"z": "",
"name": "Group 3",
"tab": "",
"order": 3,
"disp": false,
"width": "6"
},
{
  "id": "4726c93.5b6f238",
  "type": "ui_group",
  "z": "",
  "name": "Col1",
  "tab": "2c0a67a9.03e9c",
  "order": 1,
  "disp": false,
  "width": "6"
},
{
  "id": "872347db.c261",
  "type": "ui_group",
  "z": "",
  "name": "Col2",
  "tab": "2c0a67a9.03e9c",
  "order": 2,
  "disp": false,
  "width": "6"
},
{
  "id": "2c0a67a9.03e9c",
  "type": "ui_tab",
  "z": "",
  "name": "RPi Control",
  "icon": "dashboard",
  "order": 1
},
{
  "id": "c082fe66.73da9",
  "type": "ui_group",
  "z": "",
  "name": "Default",
  "tab": "",
  "order": 1,
  "disp": true,
  "width": "12",
  "collapse": false
},
{
  "id": "db10f6cb.28bae8",
  "type": "ui_tab",
  "z": "",
  "name": "Costaflores Network Status",
  "icon": "dashboard",
```

```
    "order": 4,
    "disabled": false,
    "hidden": false
  },
  {
    "id": "da4107cd.9719b8",
    "type": "ui_group",
    "z": "",
    "name": "Iframe",
    "tab": "db10f6cb.28bae8",
    "order": 1,
    "disp": false,
    "width": "26",
    "collapse": false
  },
  {
    "id": "2ad24e2c.c51302",
    "type": "ui_group",
    "z": "",
    "name": "Daily data",
    "tab": "a2bde7d.6a63f18",
    "order": 5,
    "disp": true,
    "width": "22",
    "collapse": false
  },
  {
    "id": "7319f621.40f18",
    "type": "ui_group",
    "z": "",
    "name": "Charts",
    "tab": "",
    "order": 2,
    "disp": false,
    "width": "6"
  },
  {
    "id": "f51b69c4.7ec4",
    "type": "ui_group",
    "z": "",
    "name": "Inputs",
    "tab": "",
    "order": 1,
    "disp": false,
    "width": "6"
  },
  {
    "id": "9297ba18.8577e",
    "type": "ui_group",
    "z": "",
    "name": "Group 3",
    "tab": "",
    "order": 3,
```

```
"disp": false,
"width": "6"
},
{
  "id": "6b7935fb.d63d04",
  "type": "ui_spacer",
  "name": "spacer",
  "group": "alaad443.4d5388",
  "order": 3,
  "width": 1,
  "height": 1
},
{
  "id": "d34a8674.b3833",
  "type": "ui_spacer",
  "name": "spacer",
  "group": "a90eb261.8b188",
  "order": 3,
  "width": 1,
  "height": 1
},
{
  "id": "b8bed28c.af375",
  "type": "ui_group",
  "z": "",
  "name": "Maximum",
  "tab": "a2bde7d.6a63f18",
  "order": 2,
  "disp": true,
  "width": "8",
  "collapse": false
},
{
  "id": "c7362e41.0b3ff8",
  "type": "ui_group",
  "z": "",
  "name": "Minimum",
  "tab": "a2bde7d.6a63f18",
  "order": 3,
  "disp": true,
  "width": "8",
  "collapse": false
},
{
  "id": "a2b846e7.fdc77",
  "type": "ui_spacer",
  "name": "spacer",
  "group": "2ad24e2c.c51302",
  "order": 2,
  "width": 1,
  "height": 1
},
{
```

```
    "id": "90e54760.5d0588",
    "type": "ui_group",
    "z": "",
    "name": "Montly data",
    "tab": "a2bde7d.6a63f18",
    "order": 6,
    "disp": true,
    "width": "22",
    "collapse": false
  },
  {
    "id": "c523b47c.bcbbc",
    "type": "ui_spacer",
    "name": "spacer",
    "group": "90e54760.5d0588",
    "order": 2,
    "width": 1,
    "height": 1
  },
  {
    "id": "885ec80a.ac0fa",
    "type": "inject",
    "z": "d360b8b7.05523",
    "name": "30s",
    "topic": "30s",
    "payload": "",
    "payloadType": "date",
    "repeat": "30",
    "crontab": "",
    "once": true,
    "onceDelay": "3",
    "x": 210,
    "y": 80,
    "wires": [
      [
        "91db56d3.9b2358"
      ]
    ]
  },
  {
    "id": "91db56d3.9b2358",
    "type": "rpi-dht22",
    "z": "d360b8b7.05523",
    "name": "DHT22",
    "topic": "dht22",
    "dht": 22,
    "pintype": "0",
    "pin": "18",
    "x": 400,
    "y": 80,
    "wires": [
      [
        "51148cdc.ed43bc",

```

```
        "3456a4b4.8cede4",
        "bcf468e0.518ca",
        "89700a2e.94fe88"
    ]
}
},
{
    "id": "51148cdc.ed43bc",
    "type": "function",
    "z": "d360b8b7.05523",
    "name": "Temperature",
    "func": "msg.topic = \"temperature\";\nmsg.payload =
msg.payload;\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 650,
    "y": 20,
    "wires": [
        [
            "cbebce5e.4525c8"
        ]
    ]
},
{
    "id": "3456a4b4.8cede4",
    "type": "function",
    "z": "d360b8b7.05523",
    "name": "Humidity",
    "func": "msg.topic = \"humidity\";\nmsg.payload =
msg.humidity;\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 640,
    "y": 140,
    "wires": [
        [
            "610c24d2.70d354"
        ]
    ]
},
{
    "id": "cbebce5e.4525c8",
    "type": "ui_gauge",
    "z": "d360b8b7.05523",
    "name": "Temperature",
    "group": "a90eb261.8b188",
    "order": 2,
    "width": "6",
    "height": "5",
    "gtype": "gage",
    "title": "Temperature",
    "label": "°C",
    "format": "{{value}}",
```

```
"min": "-30",
"max": "60",
"colors": [
  "#2328c3",
  "#17a609",
  "#ca3838"
],
"seg1": "19",
"seg2": "28",
"x": 1050,
"y": 20,
"wires": []
},
{
  "id": "610c24d2.70d354",
  "type": "ui_gauge",
  "z": "d360b8b7.05523",
  "name": "Humidity",
  "group": "a90eb261.8b188",
  "order": 4,
  "width": "6",
  "height": "5",
  "gtype": "gage",
  "title": "Humidity",
  "label": "%",
  "format": "{{value}}",
  "min": 0,
  "max": "100",
  "colors": [
    "#00b500",
    "#e6e600",
    "#ca3838"
  ],
  "seg1": "33",
  "seg2": "66",
  "x": 1040,
  "y": 140,
  "wires": []
},
{
  "id": "fd017272.637dd",
  "type": "ui_chart",
  "z": "d360b8b7.05523",
  "name": "",
  "group": "alaad443.4d5388",
  "order": 2,
  "width": "10",
  "height": "7",
  "label": "Temperature",
  "chartType": "line",
  "legend": "false",
  "xformat": "HH:mm",
  "interpolate": "linear",
```

```
"nodata": "",
"dot": true,
"ymin": "15",
"ymax": "30",
"removeOlder": "24",
"removeOlderPoints": "",
"removeOlderUnit": "3600",
"cutout": 0,
"useOneColor": false,
"colors": [
  "#b41013",
  "#b40d17",
  "#ff7f0e",
  "#2ca02c",
  "#98df8a",
  "#d62728",
  "#ff9896",
  "#9467bd",
  "#c5b0d5"
],
"useOldStyle": false,
"outputs": 1,
"x": 1050,
"y": 520,
"wires": [
  []
]
},
{
  "id": "fe061bd0.f99e18",
  "type": "ui_chart",
  "z": "d360b8b7.05523",
  "name": "Humidity",
  "group": "alaad443.4d5388",
  "order": 4,
  "width": "10",
  "height": "7",
  "label": "Humidity",
  "chartType": "line",
  "legend": "false",
  "xformat": "HH:mm",
  "interpolate": "linear",
  "nodata": "",
  "dot": true,
  "ymin": "0",
  "ymax": "100",
  "removeOlder": "24",
  "removeOlderPoints": "",
  "removeOlderUnit": "3600",
  "cutout": 0,
  "useOneColor": false,
  "colors": [
    "#25b432",
```



```

        "#26bd1e",
        "#ff7f0e",
        "#2ca02c",
        "#98df8a",
        "#d62728",
        "#ff9896",
        "#9467bd",
        "#c5b0d5"
    ],
    "useOldStyle": false,
    "outputs": 1,
    "x": 1056,
    "y": 615,
    "wires": [
        []
    ]
},
{
    "id": "a5a1d80b.4e347",
    "type": "inject",
    "z": "d360b8b7.05523",
    "name": "",
    "topic": "15 min + Startup",
    "payload": "",
    "payloadType": "date",
    "repeat": "900",
    "crontab": "",
    "once": true,
    "onceDelay": "10",
    "x": 259,
    "y": 569,
    "wires": [
        [
            "46a665df.81ablC"
        ]
    ]
},
{
    "id": "46a665df.81ablC",
    "type": "function",
    "z": "d360b8b7.05523",
    "name": "SQL Query",
    "func": "msg.topic = \"SELECT FLOOR( UNIX_TIMESTAMP(_timestamp)/900
) * 900 AS t, AVG(temperature) as temp, AVG(humidity) FROM sensor_data
GROUP BY t\"\\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 519,
    "y": 570,
    "wires": [
        [
            "891c363b.8d4"
        ]
    ]
}

```

```

    ],
    {
        "id": "891c363b.8d4",
        "type": "mysql",
        "z": "d360b8b7.05523",
        "mydb": "b7df6311.fca09",
        "name": "",
        "x": 679,
        "y": 570,
        "wires": [
            [
                "59ff6de0.1475dc",
                "634a5a67.26a0dc"
            ]
        ]
    },
    {
        "id": "59ff6de0.1475dc",
        "type": "function",
        "z": "d360b8b7.05523",
        "name": "Temperature",
        "func": "\nvar array = [];\n\nfor (var i=0;
i<msg.payload.length;i++) {\n    \narray.push({    x: msg.payload[i][\"t\"]
* 1000,\n        y: msg.payload[i][\"temp\"]\n    });\n
\n}\n\nmsg.payload = [{ \"series\": [\"temperature\"],\n    \"data\":
[array],\n    \"labels\": [\"\"] }]\n    \nreturn msg;\n\n",
        "outputs": 1,
        "noerr": 0,
        "x": 859,
        "y": 520,
        "wires": [
            [
                "fd017272.637dd"
            ]
        ]
    },
    {
        "id": "634a5a67.26a0dc",
        "type": "function",
        "z": "d360b8b7.05523",
        "name": "Humidity",
        "func": "\nvar array = [];\n\nfor (var i=0;
i<msg.payload.length;i++) {\n    \narray.push({    x: msg.payload[i][\"t\"]
* 1000,\n        y: msg.payload[i][\"AVG(humidity)\"]\n    });\n
\n}\n\nmsg.payload = [{ \"series\": [\"humidity\"],\n    \"data\": [array],\n    \"labels\": [\"\"] }]\n    \nreturn msg;\n\n",
        "outputs": 1,
        "noerr": 0,
        "x": 850,
        "y": 615,
        "wires": [
            [

```

```

        "fe061bd0.f99e18"
    ]
}
{
    "id": "bc1fb4d5.27e318",
    "type": "ui_gauge",
    "z": "67d6f99e.a13658",
    "name": "",
    "group": "4726c93.5b6f238",
    "order": 2,
    "width": 0,
    "height": 0,
    "gtype": "gage",
    "title": "CPU Temperature",
    "label": "C",
    "format": "{{value}}",
    "min": 0,
    "max": "100",
    "colors": [
        "#00b500",
        "#e6e600",
        "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
    "x": 670,
    "y": 340,
    "wires": []
},
{
    "id": "34c3b599.4a6012",
    "type": "exec",
    "z": "67d6f99e.a13658",
    "command": "vcgencmd measure_temp",
    "addpay": false,
    "append": "",
    "useSpawn": "false",
    "timer": "",
    "oldrc": false,
    "name": "RPi Temp.",
    "x": 310,
    "y": 340,
    "wires": [
        [
            "5ae8ad90.ec6214"
        ],
        [],
        []
    ]
},
{
    "id": "3e51365a.76e4ba",

```

```

    "type": "inject",
    "z": "67d6f99e.a13658",
    "name": "",
    "topic": "",
    "payload": "",
    "payloadType": "date",
    "repeat": "10",
    "crontab": "",
    "once": false,
    "x": 110,
    "y": 420,
    "wires": [
      [
        "34c3b599.4a6012",
        "3a3c0c01.c4482c",
        "889bf2dc.610058"
      ]
    ]
  },
  {
    "id": "5ae8ad90.ec6214",
    "type": "function",
    "z": "67d6f99e.a13658",
    "name": "",
    "func": "str = msg.payload\nmsg.payload =
str.substring(5,9);\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 490,
    "y": 340,
    "wires": [
      [
        "bc1fb4d5.27e318",
        "5af8453.3ef31bc"
      ]
    ]
  },
  {
    "id": "51613dfb.850f84",
    "type": "ui_button",
    "z": "67d6f99e.a13658",
    "name": "",
    "group": "872347db.c261",
    "order": 5,
    "width": "6",
    "height": "1",
    "passthru": false,
    "label": "Reboot",
    "tooltip": "",
    "color": "",
    "bgcolor": "orange",
    "icon": "",
    "payload": ""
  }

```

```
"payloadType": "str",
"topic": "",
"x": 80,
"y": 800,
"wires": [
  [
    "cdac5ba1.60b0e8"
  ]
]
},
{
  "id": "cdac5ba1.60b0e8",
  "type": "exec",
  "z": "67d6f99e.a13658",
  "command": "sudo reboot",
  "addpay": false,
  "append": "",
  "useSpawn": "false",
  "timer": "",
  "oldrc": false,
  "name": "Reboot",
  "x": 300,
  "y": 800,
  "wires": [
    [],
    [],
    []
  ]
},
{
  "id": "5af8453.3ef31bc",
  "type": "ui_chart",
  "z": "67d6f99e.a13658",
  "name": "",
  "group": "4726c93.5b6f238",
  "order": 3,
  "width": 0,
  "height": 0,
  "label": "",
  "chartType": "line",
  "legend": "false",
  "xformat": "HH:mm:ss",
  "interpolate": "linear",
  "nodata": "",
  "dot": false,
  "ymin": "",
  "ymax": "",
  "removeOlder": "24",
  "removeOlderPoints": "",
  "removeOlderUnit": "3600",
  "cutout": 0,
  "useOneColor": false,
  "colors": [
```

```

        "#b51a00",
        "#aec7e8",
        "#ff7f0e",
        "#2ca02c",
        "#98df8a",
        "#b51a00",
        "#ff9896",
        "#9467bd",
        "#c5b0d5"
    ],
    "useOldStyle": true,
    "outputs": 1,
    "x": 670,
    "y": 380,
    "wires": [
        []
    ]
},
{
    "id": "3a3c0c01.c4482c",
    "type": "exec",
    "z": "67d6f99e.a13658",
    "command": "top -d 0.5 -b -n2 | grep \"Cpu(s)\"|tail -n 1 | awk
' {print $2 + $4} '",
    "addpay": false,
    "append": "",
    "useSpawn": "false",
    "timer": "",
    "oldrc": false,
    "name": "CPU Load",
    "x": 310,
    "y": 420,
    "wires": [
        [
            "1871c9e5.0758ee"
        ],
        [],
        []
    ]
},
{
    "id": "889bf2dc.610058",
    "type": "exec",
    "z": "67d6f99e.a13658",
    "command": "free | grep Mem | awk '{print 100*($4+$6)/$2} '",
    "addpay": false,
    "append": "",
    "useSpawn": "false",
    "timer": "",
    "oldrc": false,
    "name": "Free Memory",
    "x": 310,
    "y": 500,

```

```
    "wires": [
      [
        "abe6a638.7b7e58"
      ],
      [],
      []
    ]
  },
  {
    "id": "1871c9e5.0758ee",
    "type": "ui_gauge",
    "z": "67d6f99e.a13658",
    "name": "",
    "group": "4726c93.5b6f238",
    "order": 1,
    "width": 0,
    "height": 0,
    "gtype": "gage",
    "title": "Processor",
    "label": "CPU",
    "format": "{{value}}",
    "min": 0,
    "max": "100",
    "colors": [
      "#00b500",
      "#e6e600",
      "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
    "x": 670,
    "y": 420,
    "wires": []
  },
  {
    "id": "abe6a638.7b7e58",
    "type": "ui_gauge",
    "z": "67d6f99e.a13658",
    "name": "",
    "group": "872347db.c261",
    "order": 2,
    "width": 0,
    "height": 0,
    "gtype": "gage",
    "title": "Memory",
    "label": "RAM",
    "format": "{{value}}",
    "min": 0,
    "max": "100",
    "colors": [
      "#b51a00",
      "#e6e600",
      "#4cbb47"
    ]
  }
}
```

```
],
"seg1": "",
"seg2": "",
"x": 670,
"y": 500,
"wires": []
},
{
  "id": "9be27828.91a62",
  "type": "exec",
  "z": "67d6f99e.a13658",
  "command": "df -h",
  "addpay": false,
  "append": "",
  "useSpawn": "false",
  "timer": "",
  "oldrc": false,
  "name": "Disk Usage",
  "x": 310,
  "y": 580,
  "wires": [
    [
      "2971090b.492b26"
    ],
    [],
    []
  ]
},
{
  "id": "d22a3729.a53978",
  "type": "ui_gauge",
  "z": "67d6f99e.a13658",
  "name": "",
  "group": "872347db.c261",
  "order": 1,
  "width": 0,
  "height": 0,
  "gtype": "gage",
  "title": "Disk",
  "label": "Usage",
  "format": "{{value}}",
  "min": 0,
  "max": "100",
  "colors": [
    "#00b500",
    "#e6e600",
    "#ca3838"
  ],
  "seg1": "",
  "seg2": "",
  "x": 670,
  "y": 580,
  "wires": []
}
```



```
},
{
  "id": "2971090b.492b26",
  "type": "function",
  "z": "67d6f99e.a13658",
  "name": "",
  "func": "var re = /([0-9]{2})%/\nvar idx =
msg.payload.search(re);\nvar str = msg.payload;\nif (idx >=0) {\n  str =
msg.payload.substring(idx, idx + 2);\n}\nmsg.payload = str;\nreturn msg;",
  "outputs": 1,
  "noerr": 0,
  "x": 490,
  "y": 580,
  "wires": [
    [
      "d22a3729.a53978"
    ]
  ]
},
{
  "id": "7304e063.77be3",
  "type": "inject",
  "z": "67d6f99e.a13658",
  "name": "",
  "topic": "",
  "payload": "",
  "payloadType": "date",
  "repeat": "60",
  "crontab": "",
  "once": false,
  "x": 110,
  "y": 580,
  "wires": [
    [
      "9be27828.91a62"
    ]
  ]
},
{
  "id": "79c74c62.f1e554",
  "type": "exec",
  "z": "67d6f99e.a13658",
  "command": "sudo systemctl restart app",
  "addpay": false,
  "append": "",
  "useSpawn": "false",
  "timer": "",
  "oldrc": false,
  "name": "Restart app",
  "x": 310,
  "y": 740,
  "wires": [
    [],
  ]
}
```

```
        [],
        []
    ]
},
{
    "id": "f67d505b.90fa28",
    "type": "ui_button",
    "z": "67d6f99e.a13658",
    "name": "",
    "group": "872347db.c261",
    "order": 4,
    "width": 0,
    "height": 0,
    "passthru": false,
    "label": "Reset app",
    "tooltip": "",
    "color": "white",
    "bgcolor": "",
    "icon": "",
    "payload": "",
    "payloadType": "str",
    "topic": "",
    "x": 90,
    "y": 740,
    "wires": [
        [
            "79c74c62.f1e554"
        ]
    ]
},
{
    "id": "eac7a940.5772d8",
    "type": "ui_button",
    "z": "67d6f99e.a13658",
    "name": "",
    "group": "872347db.c261",
    "order": 6,
    "width": 0,
    "height": 0,
    "passthru": false,
    "label": "Shutdown",
    "tooltip": "",
    "color": "",
    "bgcolor": "red",
    "icon": "",
    "payload": "",
    "payloadType": "str",
    "topic": "",
    "x": 90,
    "y": 860,
    "wires": [
        [
            "4fc07dc8.a7befc"
        ]
    ]
}
```

```
    ]
  ],
  {
    "id": "4fc07dc8.a7befc",
    "type": "exec",
    "z": "67d6f99e.a13658",
    "command": "sudo shutdown -h now",
    "addpay": false,
    "append": "",
    "useSpawn": "false",
    "timer": "",
    "oldrc": false,
    "name": "Shutdown",
    "x": 310,
    "y": 860,
    "wires": [
      [],
      [],
      []
    ]
  },
  {
    "id": "4aea8fe0.3126f8",
    "type": "exec",
    "z": "67d6f99e.a13658",
    "command": "sudo systemctl restart ethereum-node",
    "addpay": false,
    "append": "",
    "useSpawn": "false",
    "timer": "",
    "oldrc": false,
    "name": "Restart node",
    "x": 310,
    "y": 680,
    "wires": [
      [],
      [],
      []
    ]
  },
  {
    "id": "f1782408.65d8d",
    "type": "ui_button",
    "z": "67d6f99e.a13658",
    "name": "",
    "group": "872347db.c261",
    "order": 3,
    "width": 0,
    "height": 0,
    "passthru": false,
    "label": "Reset node",
    "tooltip": ""
  }
}
```

```
"color": "white",
"bgcolor": "",
"icon": "",
"payload": "",
"payloadType": "str",
"topic": "",
"x": 90,
"y": 680,
"wires": [
  [
    "4aea8fe0.3126f8"
  ]
]
},
{
  "id": "63275d50.899cc4",
  "type": "ui_template",
  "z": "4acb4e01.568548",
  "group": "da4107cd.9719b8",
  "name": "Ethereum Network Status",
  "order": 1,
  "width": "26",
  "height": "12",
  "format": "\n<iframe \n          width=\"100%\" \n
height=\"100%\" \n \n          src=\"\n          http://10.112.48.25:3000\n\n
\n\n          >\n</iframe>\n",
  "storeOutMessages": true,
  "fwdInMessages": true,
  "templateScope": "local",
  "x": 170,
  "y": 100,
  "wires": [
    []
  ]
},
{
  "id": "99570ccf.d73bb8",
  "type": "inject",
  "z": "d360b8b7.05523",
  "name": "",
  "topic": "6 h + Startup",
  "payload": "",
  "payloadType": "date",
  "repeat": "21600",
  "crontab": "",
  "once": true,
  "onceDelay": "10",
  "x": 230,
  "y": 800,
  "wires": [
    [
      "988a2ea1.0daa28"
    ]
  ]
}
```

```

    ]
  },
  {
    "id": "988a2ea1.0daa28",
    "type": "function",
    "z": "d360b8b7.05523",
    "name": "SQL Query",
    "func": "msg.topic = \"SELECT UNIX_TIMESTAMP(_day), temperature,
humidity FROM sensor_data_daily;\";\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 510,
    "y": 800,
    "wires": [
      [
        "a1f74cd5.7257a"
      ]
    ]
  },
  {
    "id": "a1f74cd5.7257a",
    "type": "mysql",
    "z": "d360b8b7.05523",
    "mydb": "b7df6311.fca09",
    "name": "",
    "x": 670,
    "y": 800,
    "wires": [
      [
        "faed0bfa.54f16",
        "42a83924.7292f"
      ]
    ]
  },
  {
    "id": "faed0bfa.54f16",
    "type": "function",
    "z": "d360b8b7.05523",
    "name": "Temperature",
    "func": "\nvar array = [];\n\nfor (var i=0;
i<msg.payload.length;i++) {\n  \narray.push({  x:
parseInt(msg.payload[i][\"UNIX_TIMESTAMP(_day)\"]) * 1000,\n
y: msg.payload[i][\"temperature\"]\n  });\n
\n}\n\nmsg.payload = [{ \"series\": [\"temperature\"],\n\n  \"data\":
[array],\n\n  \"labels\": [\"\"] }]\n  \nreturn msg;\n\n",
    "outputs": 1,
    "noerr": 0,
    "x": 850,
    "y": 740,
    "wires": [
      [
        "5cf2f494.0fc47c"
      ]
    ]
  }
}

```

```

    ],
    {
      "id": "5cf2f494.0fc47c",
      "type": "ui_chart",
      "z": "d360b8b7.05523",
      "name": "",
      "group": "2ad24e2c.c51302",
      "order": 1,
      "width": "10",
      "height": "7",
      "label": "Daily temperature",
      "chartType": "line",
      "legend": "false",
      "xformat": "D/M",
      "interpolate": "bezier",
      "nodata": "",
      "dot": true,
      "ymin": "0",
      "ymax": "31",
      "removeOlder": "31",
      "removeOlderPoints": "",
      "removeOlderUnit": "604800",
      "cutout": 0,
      "useOneColor": true,
      "colors": [
        "#b51a00",
        "#aec7e8",
        "#ff7f0e",
        "#2ca02c",
        "#98df8a",
        "#b51a00",
        "#ff9896",
        "#9467bd",
        "#c5b0d5"
      ],
      "useOldStyle": false,
      "outputs": 1,
      "x": 1080,
      "y": 740,
      "wires": [
        []
      ]
    },
    {
      "id": "42a83924.7292f",
      "type": "function",
      "z": "d360b8b7.05523",
      "name": "Humidity",
      "func": "\nvar array = [];\n\nfor (var i=0;
i<msg.payload.length;i++) {\n  \narray.push({    x:
parseInt(msg.payload[i][\"UNIX_TIMESTAMP(_day)\"]) * 1000,\n
y: msg.payload[i][\"humidity\"]\n    });\n

```



```
},
{
  "id": "a03c19d.8a04768",
  "type": "ui_text",
  "z": "d360b8b7.05523",
  "group": "b8bed28c.af375",
  "order": 1,
  "width": "0",
  "height": "0",
  "name": "",
  "label": "Max. temperature",
  "format": "{{msg.max_temp}}",
  "layout": "row-spread",
  "x": 1070,
  "y": 220,
  "wires": []
},
{
  "id": "2da5719.7833a0e",
  "type": "ui_text",
  "z": "d360b8b7.05523",
  "group": "c7362e41.0b3ff8",
  "order": 5,
  "width": "0",
  "height": "0",
  "name": "",
  "label": "Min. temperature",
  "format": "{{msg.min_temp}}",
  "layout": "row-spread",
  "x": 1070,
  "y": 260,
  "wires": []
},
{
  "id": "f1dc7e83.ee8ea8",
  "type": "ui_text",
  "z": "d360b8b7.05523",
  "group": "b8bed28c.af375",
  "order": 2,
  "width": "0",
  "height": "0",
  "name": "",
  "label": "Max. humidity",
  "format": "{{msg.max_hum}}",
  "layout": "row-spread",
  "x": 1060,
  "y": 300,
  "wires": []
},
{
  "id": "c78993c6.afc99",
  "type": "ui_text",
  "z": "d360b8b7.05523",
```



```
"group": "c7362e41.0b3ff8",
"order": 8,
"width": "0",
"height": "0",
"name": "",
"label": "Min. humidity",
"format": "{{msg.min_hum}}",
"layout": "row-spread",
"x": 1060,
"y": 340,
"wires": []
},
{
  "id": "6dd844eb.0b8bd4",
  "type": "file in",
  "z": "d360b8b7.05523",
  "name": "read maxmin file",
  "filename": "/home/pi/db/maxmin.txt",
  "format": "utf8",
  "chunk": false,
  "sendError": false,
  "encoding": "none",
  "x": 420,
  "y": 340,
  "wires": [
    [
      "bcf468e0.518ca"
    ]
  ]
},
{
  "id": "2aaade0e.7dc94a",
  "type": "inject",
  "z": "d360b8b7.05523",
  "name": "",
  "topic": "",
  "payload": "",
  "payloadType": "date",
  "repeat": "",
  "crontab": "",
  "once": true,
  "onceDelay": 0.1,
  "x": 190,
  "y": 340,
  "wires": [
    [
      "6dd844eb.0b8bd4"
    ]
  ]
},
{
  "id": "bcf468e0.518ca",
  "type": "function",
```

```

    "z": "d360b8b7.05523",
    "name": "max_min",
    "func": "context.max_temp = context.max_temp ||
0.0;\ncontext.min_temp = context.min_temp || 100.0;\ncontext.max_humidity =
context.max_humidity || 0.0;\ncontext.min_humidity = context.min_humidity
|| 100.0;\n\nif (msg.topic != \"dht22\") {\n    \n    var array =
msg.payload.split(\"\\\",\\\")\n    context.max_temp = parseFloat(array[0])\n
context.min_temp = parseFloat(array[1])\n    context.max_humidity =
parseFloat(array[2])\n    context.min_humidity = parseFloat(array[3])\n
\n} else {\n    \n    if (msg.payload > context.max_temp) {\n
context.max_temp = msg.payload;\n    }\n    \n    if (msg.payload <
context.min_temp) {\n        context.min_temp = msg.payload;\n    }\n    \n
if(msg.humidity > context.max_humidity) {\n        context.max_humidity =
msg.humidity;\n    }\n    \n    if(msg.humidity < context.min_humidity) {\n
context.min_humidity = msg.humidity;\n    }\n}\n\nvar db = context.max_temp
+ \"\\\",\" +\n        context.min_temp + \"\\\",\" +\n
context.max_humidity + \"\\\",\" +\n
context.min_humidity;\nmsg.payload = db;\nmsg.max_temp =
context.max_temp;\nmsg.min_temp = context.min_temp;\nmsg.max_hum =
context.max_humidity;\nmsg.min_hum = context.min_humidity;\n\nreturn
msg;\n",
    "outputs": 1,
    "noerr": 0,
    "x": 640,
    "y": 260,
    "wires": [
        [
            "a03c19d.8a04768",
            "2da5719.7833a0e",
            "f1dc7e83.ee8ea8",
            "c78993c6.afc99",
            "5d79c235.0cb13c"
        ]
    ]
},
{
    "id": "5d79c235.0cb13c",
    "type": "file",
    "z": "d360b8b7.05523",
    "name": "write maxmin file",
    "filename": "/home/pi/db/maxmin.txt",
    "appendNewline": true,
    "createDir": false,
    "overwriteFile": "true",
    "encoding": "none",
    "x": 1010,
    "y": 420,
    "wires": [
        []
    ]
},
{
    "id": "3de9fca0.b67044",

```

```
    "type": "mysql",
    "z": "d360b8b7.05523",
    "mydb": "b7df6311.fca09",
    "name": "",
    "x": 1030,
    "y": 80,
    "wires": [
      []
    ]
  },
  {
    "id": "89700a2e.94fe88",
    "type": "function",
    "z": "d360b8b7.05523",
    "name": "SQL Query",
    "func": "msg.topic =\n\"INSERT INTO sensor_data VALUES\n(NOW(),?,?,?);\n\";\nmsg.payload = ['DHT22', msg.payload,\nmsg.humidity];\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 650,
    "y": 80,
    "wires": [
      [
        "3de9fca0.b67044"
      ]
    ]
  },
  {
    "id": "74cad1b6.39628",
    "type": "inject",
    "z": "d360b8b7.05523",
    "name": "",
    "topic": "12 h + Startup",
    "payload": "",
    "payloadType": "date",
    "repeat": "43200",
    "crontab": "",
    "once": true,
    "onceDelay": "10",
    "x": 240,
    "y": 1020,
    "wires": [
      [
        "6c0140a1.d94268"
      ]
    ]
  },
  {
    "id": "6c0140a1.d94268",
    "type": "function",
    "z": "d360b8b7.05523",
    "name": "SQL Query",
```

```
    "func": "msg.topic = \"SELECT MONTH(_day), MAX(temperature),
AVG(temperature), MIN(temperature), MAX(humidity), AVG(humidity),
MIN(humidity) FROM sensor_data_daily GROUP BY MONTH(_day);\"\\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 510,
    "y": 1020,
    "wires": [
      [
        "5d40a2f7.ca996c"
      ]
    ]
  },
  {
    "id": "5d40a2f7.ca996c",
    "type": "mysql",
    "z": "d360b8b7.05523",
    "mydb": "b7df6311.fca09",
    "name": "",
    "x": 670,
    "y": 1020,
    "wires": [
      [
        "d6bf0368.13ee",
        "9daa9977.721b1"
      ]
    ]
  },
  {
    "id": "9daa9977.721b1",
    "type": "function",
    "z": "d360b8b7.05523",
    "name": "Temperature",
    "func": "\\nvar months = [\"Jan\", \"Feb\", \"Mar\", \"Apr\",
\\\"May\", \"Jun\", \"Jul\", \"Aug\", \"Sep\", \"Oct\", \"Nov\",
\\\"Dec\"]\\nvar series = [\"Max. temp\", \"Avg. temp\", \"Min. temp\"]\\nvar
labels = [];\\nvar maxs = [];\\nvar avgs = [];\\nvar mins = [];\\n\\nfor (var
i=0; i<msg.payload.length;i++) {\\n
maxs.push(msg.payload[i][\"MAX(temperature)\"]);\\n
avgs.push(msg.payload[i][\"AVG(temperature)\"]);\\n
mins.push(msg.payload[i][\"MIN(temperature)\"]);\\n
labels.push(months[msg.payload[i][\"MONTH(_day)\"]-1]);
\\n}\\n\\nmsg.payload = [{\\n  \"series\": series,\\n  \"data\": [ maxs,
avgs, mins ],\\n  \"labels\": labels\\n}]\\n  \\nreturn msg;\\n\\n",
    "outputs": 1,
    "noerr": 0,
    "x": 850,
    "y": 960,
    "wires": [
      [
        "9b208a9a.588738"
      ]
    ]
  }
]
```

```

    },
    {
      "id": "d6bf0368.13ee",
      "type": "function",
      "z": "d360b8b7.05523",
      "name": "Humidity",
      "func": "\nvar months = [\"Jan\", \"Feb\", \"Mar\", \"Apr\",
      \"May\", \"Jun\", \"Jul\", \"Aug\", \"Sep\", \"Oct\", \"Nov\",
      \"Dec\"]\nvar series = [\"Max. hum\", \"Avg. hum\", \"Min. hum\"]\nvar
      labels = []\nvar maxs = []\nvar avgs = []\nvar mins = []\n\nfor (var
      i=0; i<msg.payload.length;i++) {\n
      maxs.push(msg.payload[i][\"MAX(humidity)\"]); \n
      avgs.push(msg.payload[i][\"AVG(humidity)\"]); \n
      mins.push(msg.payload[i][\"MIN(humidity)\"]); \n
      labels.push(months[msg.payload[i][\"MONTH(_day)\"]-1]);
      \n}\n\nmsg.payload = [{\n  \"series\": series,\n  \"data\": [ maxs,
      avgs, mins ],\n  \"labels\": labels\n}]\n  \nreturn msg;\n\n",
      "outputs": 1,
      "noerr": 0,
      "x": 840,
      "y": 1080,
      "wires": [
        [
          "4845257.78cd75c"
        ]
      ]
    }
  ],
  {
    "id": "4845257.78cd75c",
    "type": "ui_chart",
    "z": "d360b8b7.05523",
    "name": "",
    "group": "90e54760.5d0588",
    "order": 3,
    "width": "10",
    "height": "7",
    "label": "Monthly humidity",
    "chartType": "bar",
    "legend": "true",
    "xformat": "auto",
    "interpolate": "bezier",
    "nodata": "",
    "dot": true,
    "ymin": "0",
    "ymax": "31",
    "removeOlder": "31",
    "removeOlderPoints": "",
    "removeOlderUnit": "86400",
    "cutout": 0,
    "useOneColor": true,
    "colors": [
      "#b51a00",
      "#77bb40",
    ]
  }

```

```
        "#0056d6",
        "#2ca02c",
        "#98df8a",
        "#b51a00",
        "#ff9896",
        "#9467bd",
        "#c5b0d5"
    ],
    "useOldStyle": false,
    "outputs": 1,
    "x": 1090,
    "y": 1080,
    "wires": [
        []
    ]
},
{
    "id": "9b208a9a.588738",
    "type": "ui_chart",
    "z": "d360b8b7.05523",
    "name": "",
    "group": "90e54760.5d0588",
    "order": 1,
    "width": "10",
    "height": "7",
    "label": "Monthly temperature",
    "chartType": "bar",
    "legend": "true",
    "xformat": "auto",
    "interpolate": "bezier",
    "nodata": "",
    "dot": true,
    "ymin": "0",
    "ymax": "31",
    "removeOlder": "31",
    "removeOlderPoints": "",
    "removeOlderUnit": "86400",
    "cutout": 0,
    "useOneColor": true,
    "colors": [
        "#b51a00",
        "#77bb40",
        "#0056d6",
        "#2ca02c",
        "#98df8a",
        "#b51a00",
        "#ff9896",
        "#9467bd",
        "#c5b0d5"
    ],
    "useOldStyle": false,
    "outputs": 1,
    "x": 1100,
```

```
    "y": 960,  
    "wires": [  
      []  
    ]  
  }  
]
```



Once we've added the code we can click onto import and it will automatically generate 3 flows for us. Now we just have to click onto deploy (top right corner) and check the result at: <http://your raspberry ip:1880/ui>

