

## iHost extension

There are several extensions that can be installed on iHost that allow you to expand its functionality. Below are some extensions that we recommend.

- [Node-RED](#)
- [eWeLink Smart Home](#)

## Node-RED

It is an extension that allows you to graphically program some functions not provided by iHost such as sending messages or calls to a cell phone in the event of a certain event.

### ATTENTION:

After installing Node-RED into iHost, into Node-RED you will also need to install:  
**node-red-contrib-ewelink-cube**

to access the various devices used by iHost.

[Here](#) is an explanatory video.

If you want to send **email** you will also need to install:

**node-red-node-email**

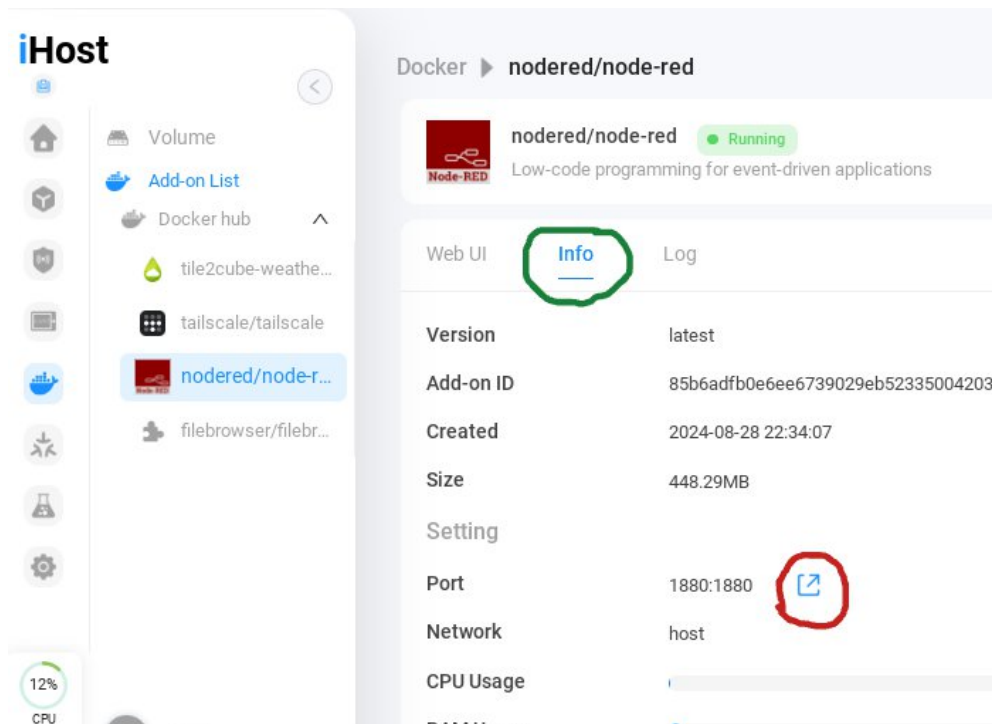
[Here](#) is an explanatory video.

To install Node-RED you need to select the **Docker icon** from iHost.



and from the page that appears, install Node-RED using the default parameters.  
Once installed, it is convenient, due to the size of the screen, to use Node-RED from a window external to iHost.

To access it, enter Node-RED, select INFO (the icon circled in **green** below) and then click on the icon circled in **red**, highlighted in the image below.



The screenshot shows the iHost interface. On the left, there is a sidebar with various icons, including a Docker icon. The main area displays the Docker container 'nodered/node-red' which is in a 'Running' state. Below the container name, there is a table of details:

Property	Value
Version	latest
Add-on ID	85b6adfb0e6ee6739029eb523350042034
Created	2024-08-28 22:34:07
Size	448.29MB
Setting	
Port	1880:1880
Network	host
CPU Usage	
RAM Usage	

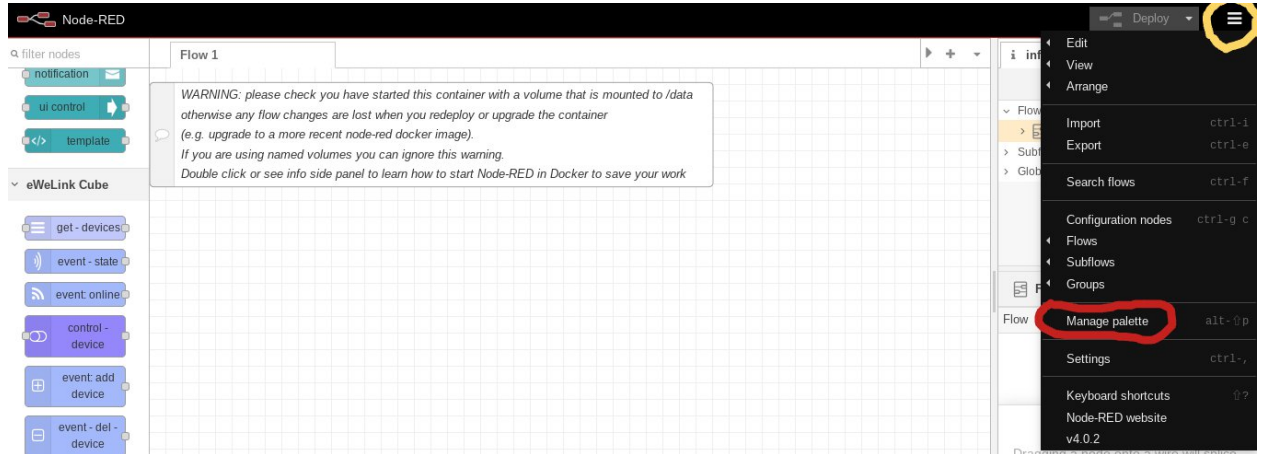
The 'Info' button in the 'Web UI' row is circled in green. The external link icon in the 'Port' row is circled in red.

Now open Node-RED and install:

### **node-red-contrib-ewelink-cube**

This is used to see the objects connected to iHost.

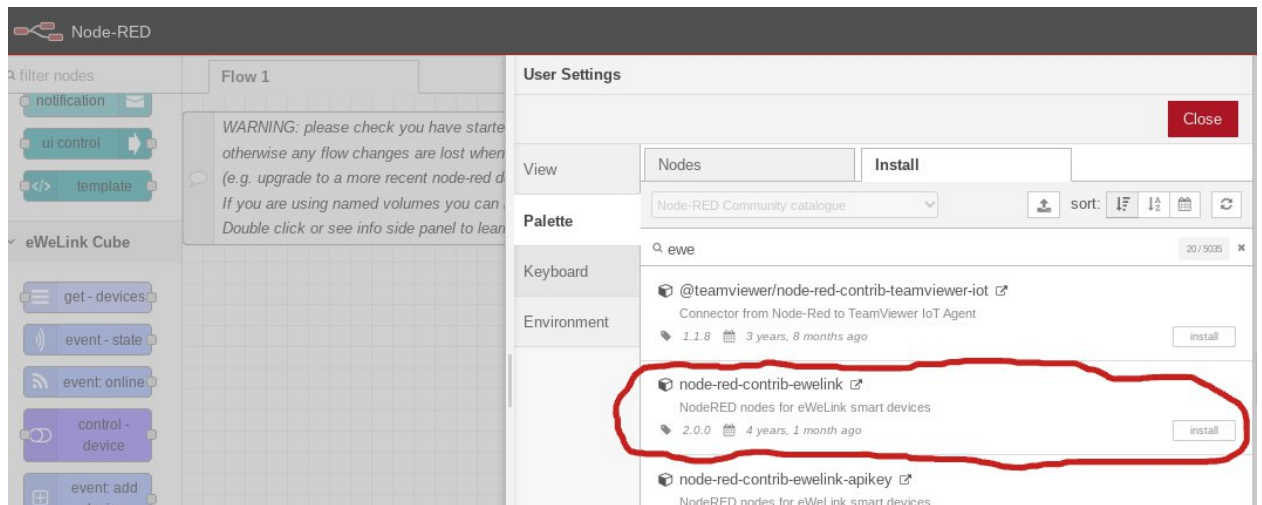
To do this, first click on the icon highlighted in **yellow** below and then, from the menu that appears, click on **MANAGE PALETTE** (icon circled in **red** below).



From the window that appears go to INSTALL and search for **eWe**.

Several possible choices will appear, you must choose and install:

### **node-red-contrib-ewelink-cube**



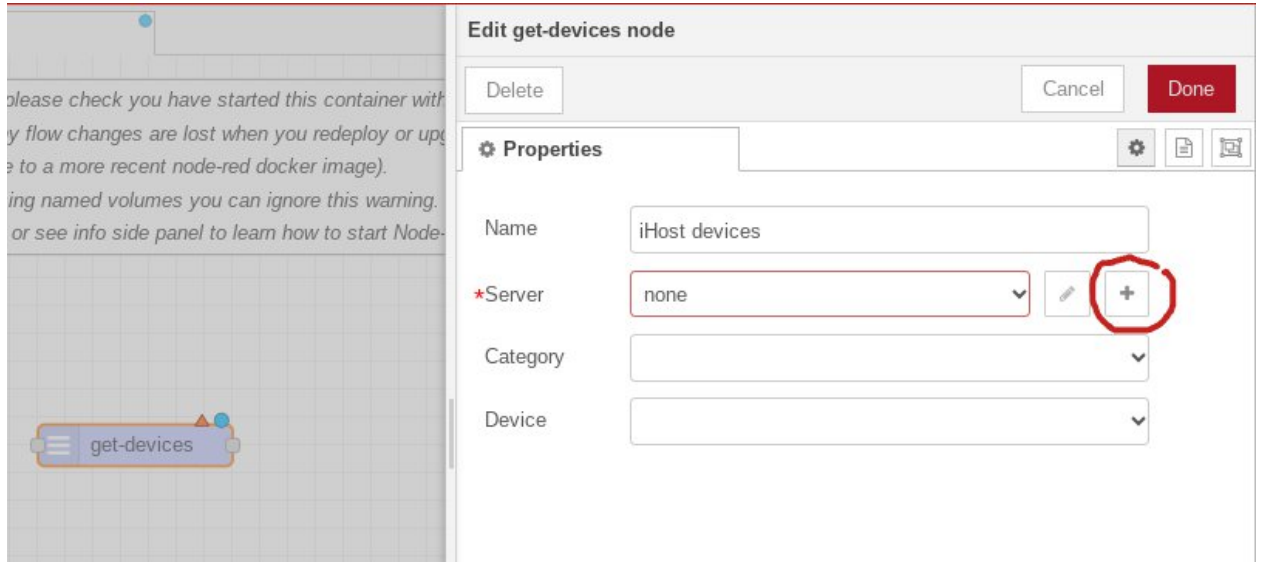
Once this is done, you can upload the devices connected to iHost to Note-RED.

To load the objects connected to iHost you have to drag the icon:

### GET-DEVICES

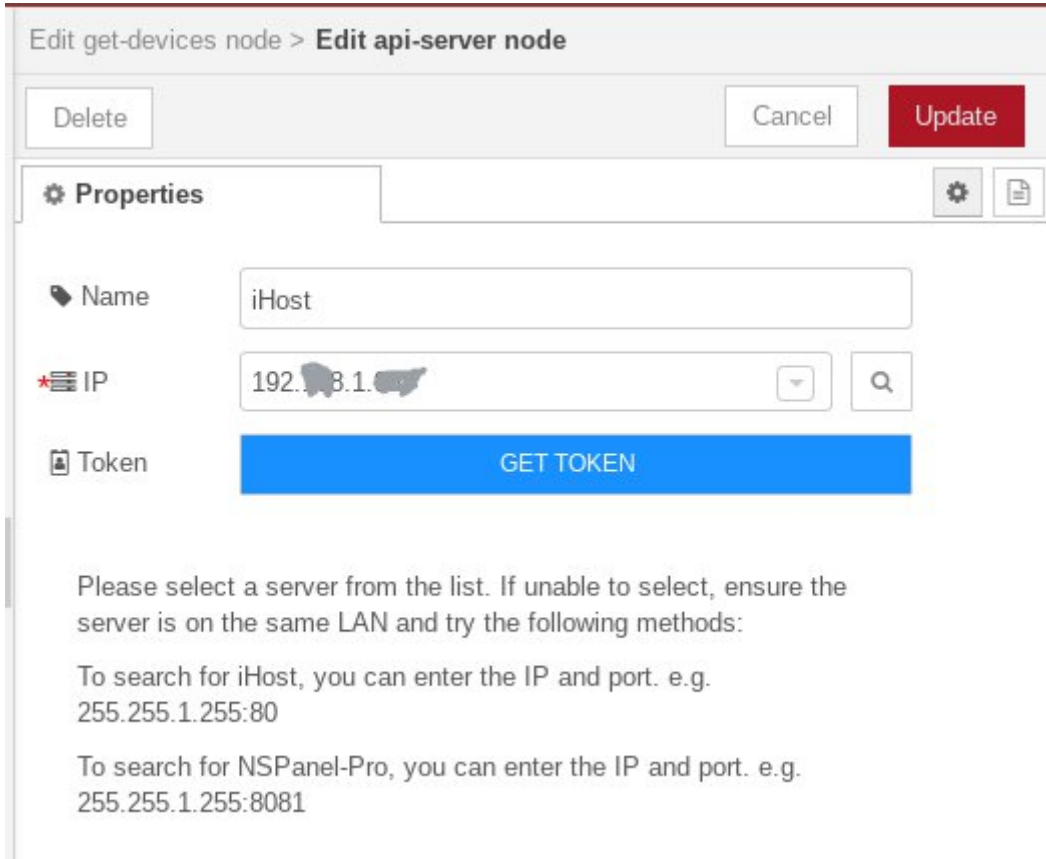
into the squared part of the screen and then double-click on it.

From the window that appears you will have to configure the **SERVER** from which to acquire the devices, to do this click on the **+** that is circled in **red** below.



From the page that appears, of which you can see an example below, you will have to fill in the fields:

- **NAME** (this will be the name you gave to your iHost)
- **IP** (your iHost IP)



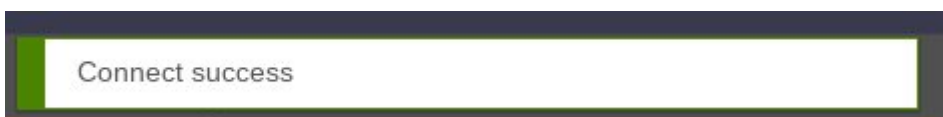
The screenshot shows a web interface for editing an API server node. At the top, there is a breadcrumb trail: "Edit get-devices node > Edit api-server node". Below this are three buttons: "Delete", "Cancel", and "Update". The main section is titled "Properties" and contains three fields:

- Name:** A text input field containing "iHost".
- IP:** A text input field containing "192.168.1.1" with a magnifying glass icon to its right.
- Token:** A blue button labeled "GET TOKEN".

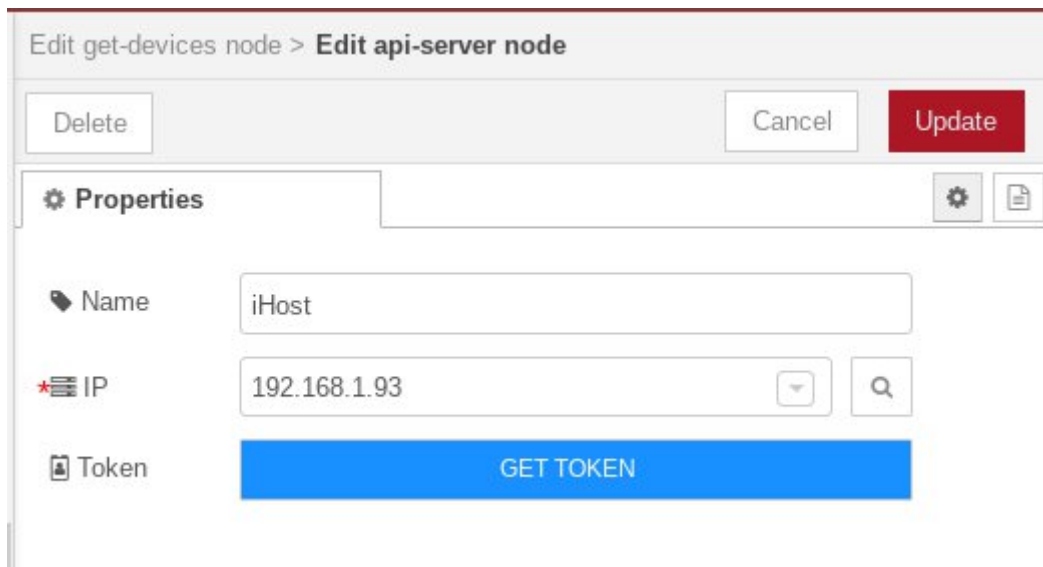
Below the fields, there is instructional text: "Please select a server from the list. If unable to select, ensure the server is on the same LAN and try the following methods:"

- To search for iHost, you can enter the IP and port. e.g. 255.255.1.255:80
- To search for NSPanel-Pro, you can enter the IP and port. e.g. 255.255.1.255:8081

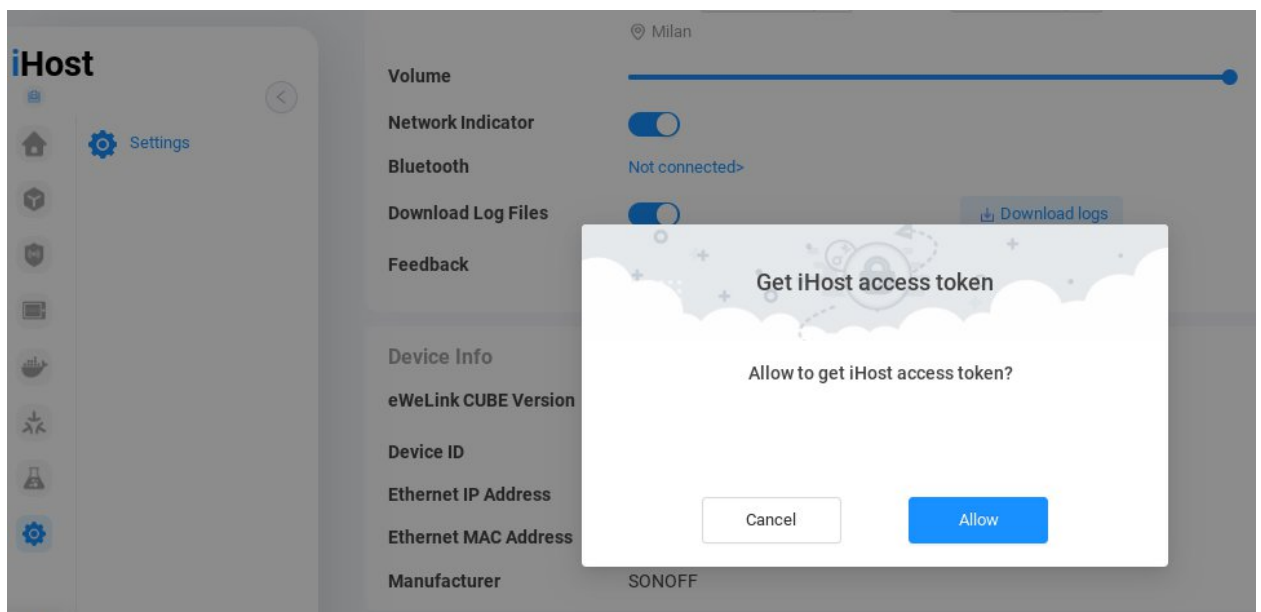
Then press on the magnifying glass icon (on the IP field line) and the writing below should appear.



At this point you will have the word **GET TOKEN** highlighted (see image below), **click on it**.



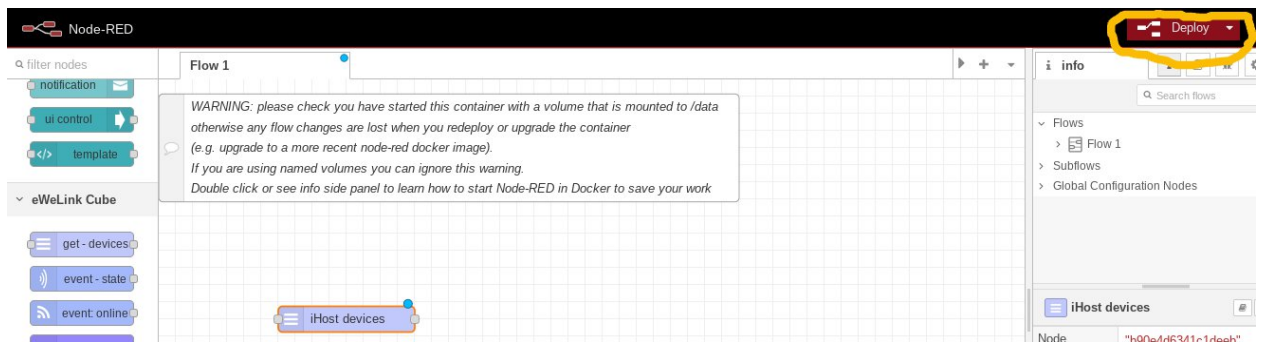
Go **back to your iHost window** and the TOKEN request window will appear (see image below), press **ALLOW**.



Go **back to the Node-RED window** and press **UPDATE**.

Now **finish filling out the window as shown below**, then press **DONE**.

Once you return to the main window, remember to press **DEPLOY**, see image below.



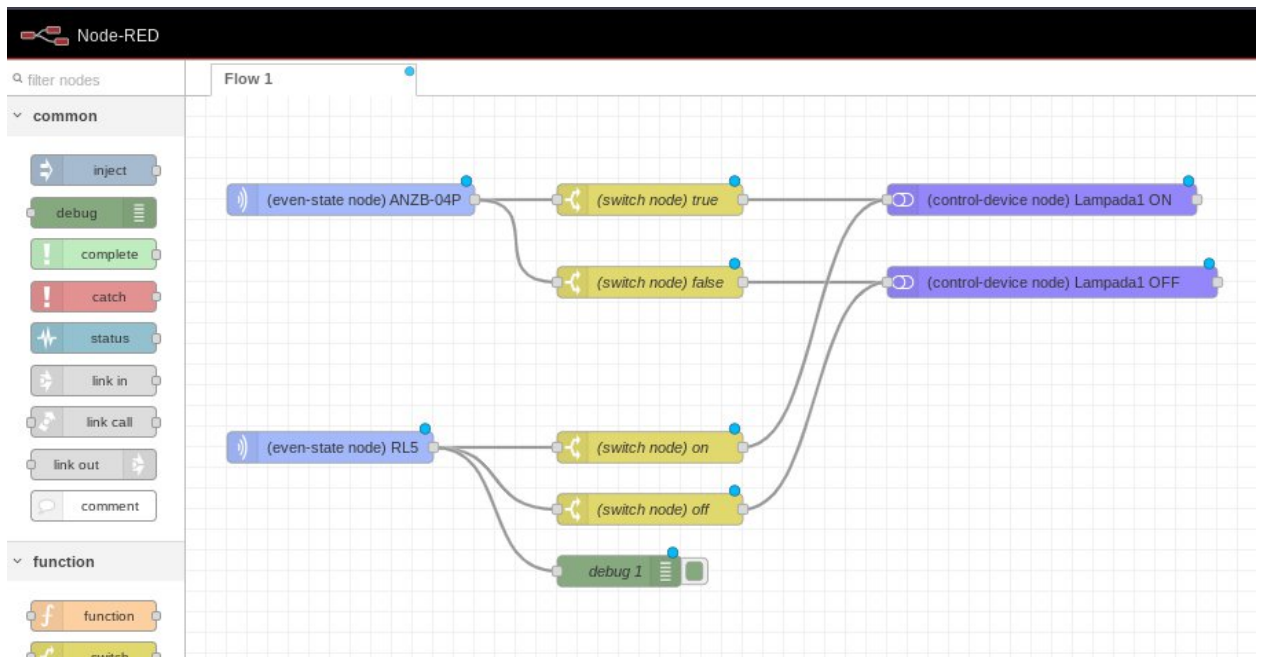
***If you have come this far, it means that your Node-RED is working perfectly on iHost and so now let's create a concrete automation with the kit below:***

- **Door/Window Contact SonOff [SNZB-04P](#)**
- **Relay SonOff [MINI](#)**
- **Bulb [Osram Smart+](#)** (that are compatible with iHost)

What we will do is turn on/off the light bulb when the Door/Window Contact opens/closes and similarly when the switch that we will connect to the Relay opens/closes (for details see [here](#)).

Below is the image of the finished automation, between the brackets (...) there is the name of the function used which I summarize:

- **even-state node**
- **switch node**
- **control-device node**
- **debug** It is optional and depending on where it has been positioned it shows the messages of the switch connected to the relay



Below are the various parts compiled according to the names we used.

The screenshot shows the 'Edit event-state node' dialog box. The dialog has a 'Delete' button, a 'Cancel' button, and a 'Done' button. The 'Properties' section contains the following fields:

- Name: (even-state node) ANZB-04P
- \*Server: iHost
- Device: eWeLink SNZB-04P
- State: Contact



### Edit event-state node

Delete Cancel Done

**Properties** [Settings] [Save] [Cancel]

Name (even-state node) RL5

\*Server iHost [v] [edit] [add]

Device RL5\_iHost [v]

State All ON/OFF [v] [add]

### Edit switch node

Delete Cancel Done

**Properties** [Settings] [Save] [Cancel]

Name (switch node) true

Property msg. payload

contains [v]  $\frac{a}{z}$  true - 1 [x]

+ add

checking all rules [v]

recreate message sequences

[Save]  Enabled

**Edit switch node**

Delete Cancel Done

**Properties** [Settings] [Copy] [Paste]

Name (switch node) false

Property msg.payload

contains false → 1 ×

+ add

checking all rules ▼

recreate message sequences

Enabled

**Edit switch node**

Delete Cancel Done

**Properties** [Settings] [Copy] [Paste]

Name (switch node) on

Property msg. payload

contains [msg. payload] on → 1 [x]

+ add

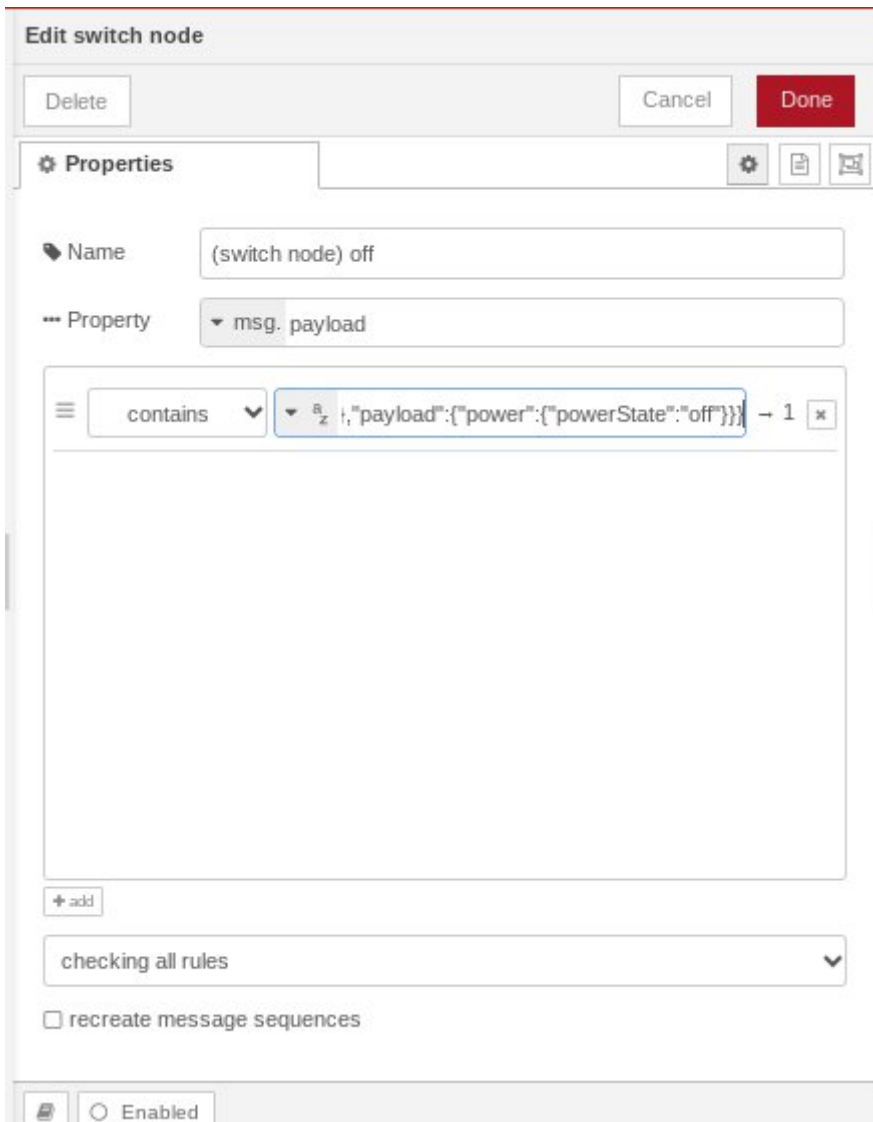
checking all rules [v]

recreate message sequences

[Icon]  Enabled

The window below is similar to the previous ones but, here we show how to insert the entire message that the relay returns when it is OFF.

We see this message in the **debug window**.



**Debug window:**



By going to the message, the icon appears that allows us to copy it.

### Edit control-device node

Delete Cancel Done

**Properties** [Settings] [Print] [Help]

Name (control-device node) Lampada1 ON

\*Server iHost [Edit] [Add]

Category light [Dropdown]

\*Device Lampada1 [Dropdown]

**\*Action**

Power  [Dropdown: On]

Brightness  [Slider: 100%]

[Print]  Enabled

**Edit control-device node**

Delete Cancel Done

**Properties**

Name (control-device node) Lampada1 OFF

\*Server iHost

Category light

\*Device Lampada1

\*Action

Power  Off

Brightness

Enabled

[Here](#) and [here](#) are some tutorials in English.  
[Qui](#) e [qui](#) vi sono dei tutorial in Inglese.

Click [here](#) to go to the beginning of the document.

## eWeLink Smart Home

If it is not already installed, you must install it to have access to the SonOff objects functionality. The installation procedure is the same as described for Node-RED.

Click [here](#) to go to the beginning of the document.