

Node-RED on Docker

- [Installing Node-RED as a Docker container](#)
- [Adding Node-RED To Home Assistant](#)
 - [Node-RED](#)
 - [Home Assistant](#)
- [Automations](#)
 - [Lights](#)
 - [Stream Deck Support](#)
 - [Messaging](#)

After seeing some of the stuff my colleague Martin has done with Node-RED, I wanted to switch to it, despite my KISS strategy, and Node-RED just introduces another device to manage.

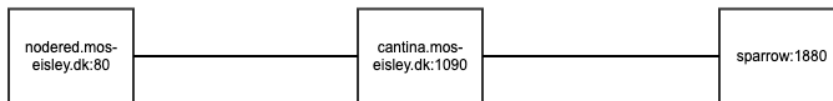
Installing Node-RED as a Docker container

But, starting it on Docker on my Home Laptop (also used for [Home Assistant on Docker](#) and [Plex Media Server](#)):

```
mkdir /opt/node-red
docker run -d --restart unless-stopped --network=host -v /opt/node-red:/data --name nodered nodered/node-red
```

And it was running on `http://sparrow:1880`. I saw some weird stuff in the start, until I pressed "Deploy" the first time, after that everything has just been working.

Node-RED has no (as far as I know currently, any users or admin/management in front of it by default, and I wanted to be able to use it from home and work/work-vpn... so I created a dual stack of Reverse Proxies:



Setting the browser to `http://nodered.mos-eisley.dk` it hits my Confluence Server (in a Datacenter) and the Apache2 on it proxies it to <http://cantina.mos-eisley.dk:1890> (Fiber Router with port forward to Home laptop), and the Apache2 on the Laptop proxies it to `sparrow:1880` (the Docker container on the Home Laptop).

On both Apache2 I have IP Restrictions, and on the first, username/password - otherwise all people at work could change my Node-RED setup:

```

<VirtualHost nodered.mos-eisley.dk:80>

    ServerName nodered.mos-eisley.dk

    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"" combined
    ErrorLog ${APACHE_LOG_DIR}/nodered.mos-eisley.dk-error.log
    CustomLog ${APACHE_LOG_DIR}/nodered.mos-eisley.dk-access.log combined

    RewriteEngine on
    RewriteCond %{HTTP:Upgrade} ^WebSocket$ [NC]
    RewriteCond %{HTTP:Connection} Upgrade$ [NC]
    RewriteRule .*/(.*) "ws://cantina.mos-eisley.dk:1890/$1" [P,L]

    ProxyPreserveHost On
    ProxyRequests Off
    ProxyPass / http://cantina.mos-eisley.dk:1890/
    ProxyPassReverse / http://cantina.mos-eisley.dk:1890/

    <Proxy *>
    Order deny,allow
    Deny from all
    Allow from xxx.xxx.xxx.xxx
    Allow from yyy.yyy.yyy.yyy

    AuthType Basic
    Authname "Password Required"
    AuthUserFile /etc/apache2/.htpasswd
    Require valid-user
    </Proxy>

</VirtualHost>

```

Creating password:

```
sudo htpasswd -c /etc/apache2/.htpasswd <username>
```

cantina.mos-eisley.dk

```

<VirtualHost *:1890>
    ServerAdmin webmaster@localhost

    RewriteEngine on
    RewriteCond %{HTTP:Upgrade} ^WebSocket$ [NC]
    RewriteCond %{HTTP:Connection} Upgrade$ [NC]
    RewriteRule .*/(.*) "ws://localhost:1880/$1" [P,L]

    ProxyPreserveHost On
    ProxyRequests Off
    ProxyPass / http://localhost:1880/
    ProxyPassReverse / http://localhost:1880/

    <Proxy *>
    Order deny,allow
    Deny from all
    Allow from 77.243.52.144 # www.mos-eisley.dk
    Allow from 10          # Local Lan
    </Proxy>

</VirtualHost>

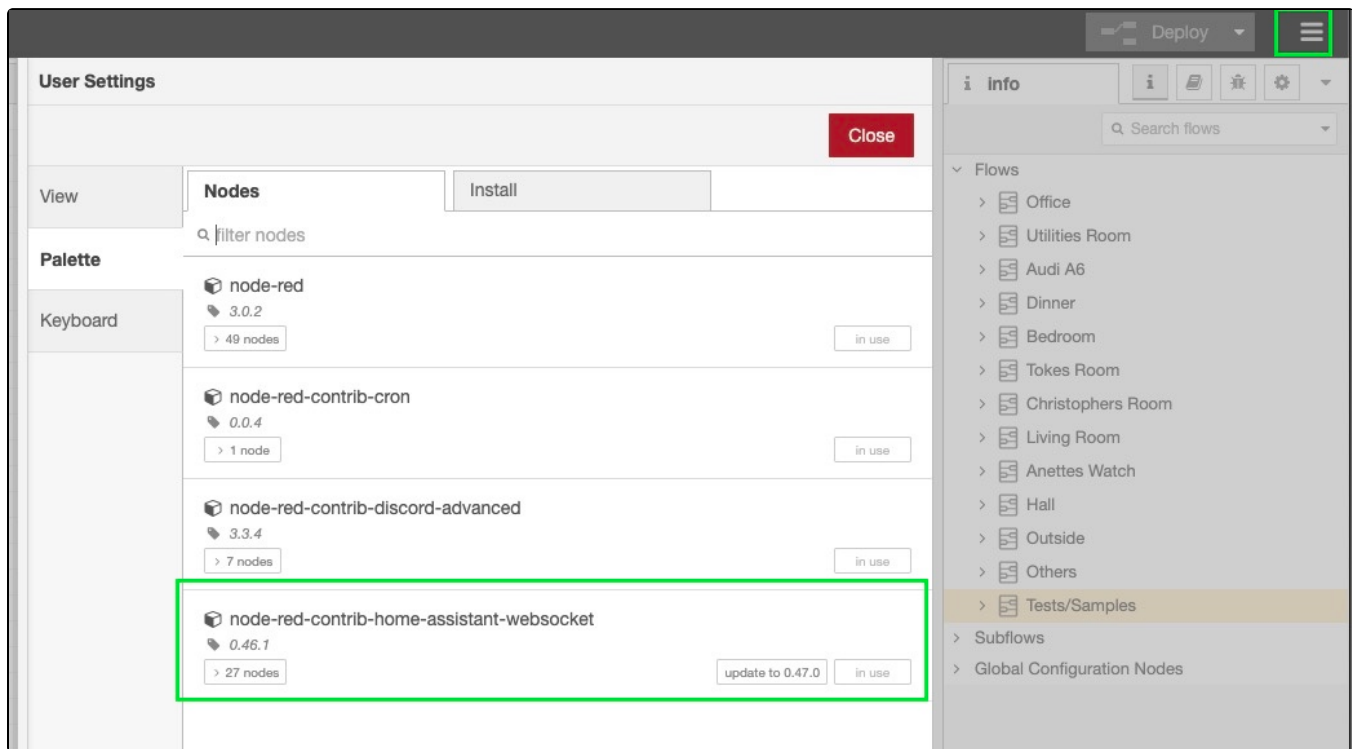
```

With this setup I can use it from xxx.xxx.xxx.xxx and yyy.yyy.yyy.yyy that represents some well known addresses like work and fiber.

Adding Node-RED To Home Assistant

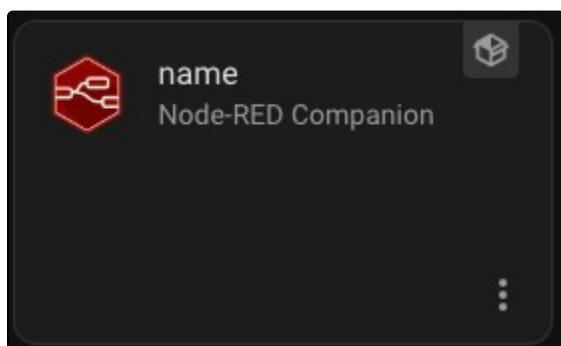
Node-RED

In Node-RED, under Settings Palette - Add "node-red-contrib-home-assistant-websocket":



Home Assistant

In Home Assistant, Add "Node-RED Companion" as an Integration:



Check out https://zachowj.github.io/node-red-contrib-home-assistant-websocket/guide/custom_integration/ for some tips also.

And You are ready to go.

Automations

Lights

Now, Home Assistant is my "heart" of the Installation, but starting to use it, I realize that it has Integrations for Philips Hue and MQTT directly, so Home Assistant could be bypassed for several automations.... I guess that more religion than tech...

My first Automations in Home Assistant was to bridge my proprietary [ZenseHome](#) system to Philips Hue:

Triggers ?

▼ ↔ When an MQTT message has been received ⋮

+ ADD TRIGGER

Conditions ?

+ ADD CONDITION

Actions ?

▼ 🏠 Turn on 10598 ⋮

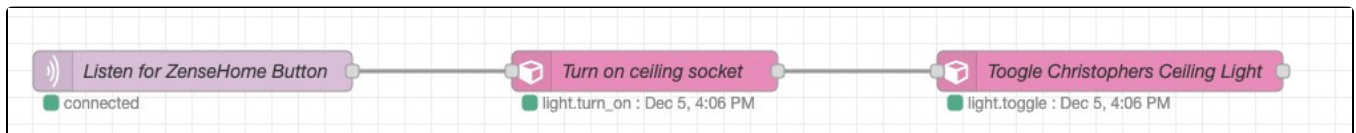
▼ 🏠 Toggle Christopher Ceiling 1 ⋮

▼ 🏠 Toggle Christopher Ceiling 2 ⋮

▼ 🏠 Toggle Christopher Ceiling 3 ⋮

Later on, [ZenseHome](#) actually got native Hue support... but I'm still using Home Assistant.

So, I moved this to Node-Red:



The "Listen" is listening on [MQTT](#) Directly and turns on the ZenseHome ceiling outlet (just to be sure) and turns on the Philips Hue lightbulbs through Home Assistant.

I have also used the Philips Hue Zone "Christophers Ceiling Light" instead of turning on all three bulbs individually.

So, I moved all these similar automations from Home Assistant to Node-Red in no time!

Stream Deck Support

My youngest son got a [Stream Deck](#) and we wanted to be able to trigger stuff from the buttons, so I looked into the Home Assistant API - and well, You need authentications and all sorts of "complicated" stuff.

So I turned to Node-Red, and found the "http in" node, which setup and url that Node-RED listens on:

Edit http in node

Delete

Cancel

Done

⚙️ Properties

⚙️

📄

🖨️

☰ Method

GET

▼

🌐 URL

switch-vr

🏷️ Name

Http Req Toggle VR

Added the toggle for the Home Assistant entity (A [Philips Hue Plug](#)):

Edit device node

Delete

Cancel

Done

⚙️ Properties

⚙️

📄

🖨️

Alpha version: At this point anything could change or not work.
Found an issue? Post it in [issues](#) . Have questions or comments? Post them [here](#) .

Name

Toogle VR

Server

Home Assistant

▼

✎

Type

Action

▼

Device

VR Base Stations

▼

Trigger

toggle

▼

And wired them together:

```
graph LR; A[Http Req Toggle VR] --> B[Toogle VR]; B --> C[http (200)];
```

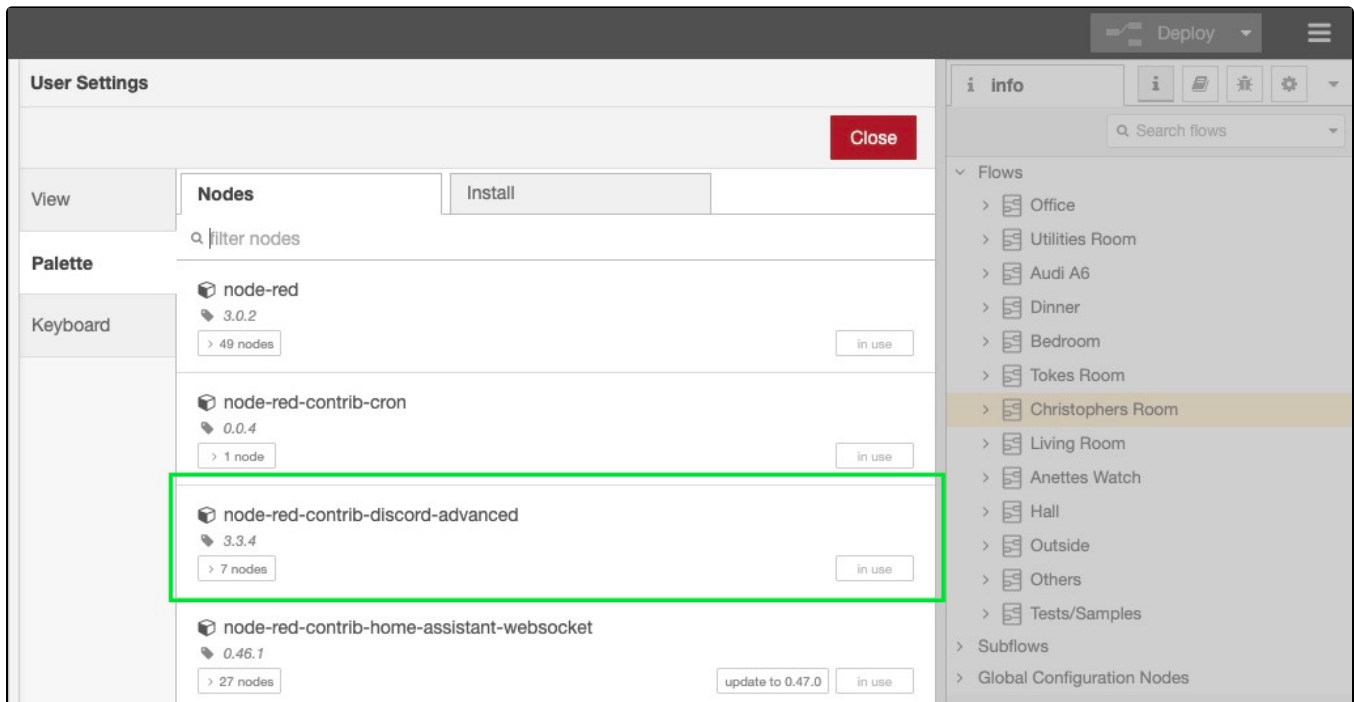
The diagram illustrates a workflow on a grid background. It starts with an orange node labeled 'Http Req Toggle VR' with a globe icon. A line connects it to a pink node labeled 'Toogle VR' with a cube icon. Below the 'Toogle VR' node is a green status indicator showing 'light.toggle : Dec 2, 4:49 PM'. Another line connects the 'Toogle VR' node to a yellow node labeled 'http (200)' with a globe icon.

Now, going to <http://sparrow:1880/switch-vr> simply toggles the VR on/off - no authentication or similar needed, the URL can be put directly into the Stream Deck software and assigned to a Button.

Messaging

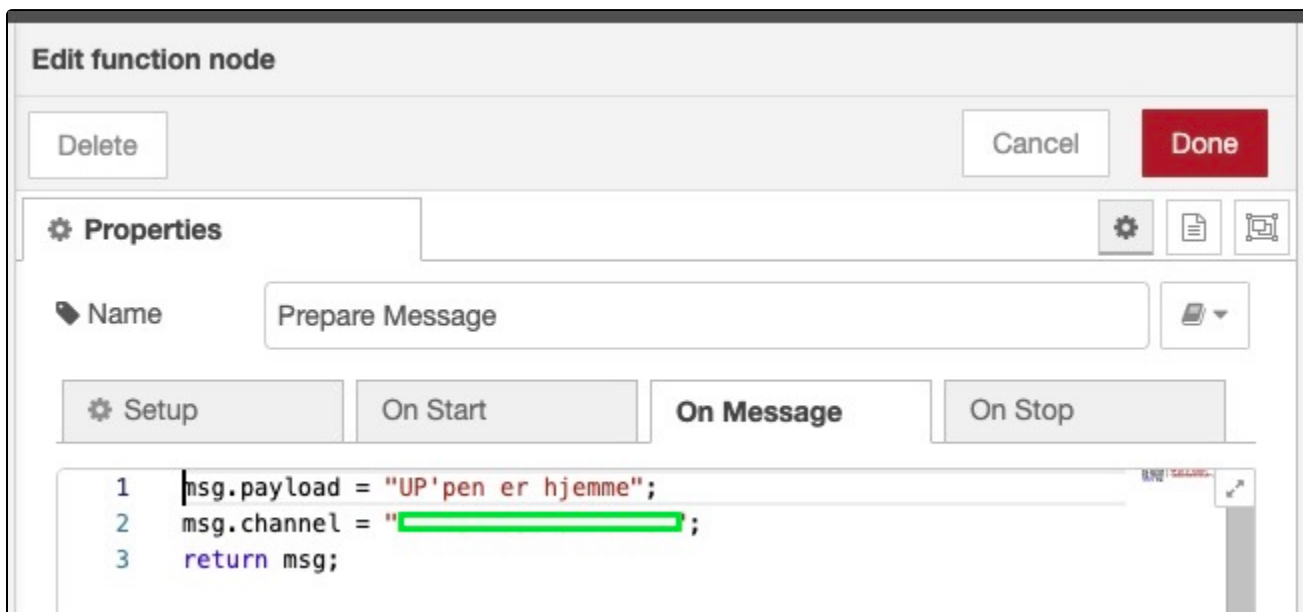
I looked into messaging (Home Assistant already setup for twillo) and found that Facebook Messenger are possible, but the setup is complicated.

So I turned (as and SMS alternative) to Discord, from the Settings Palette - Add "node-red-contrib-discord-advanced":



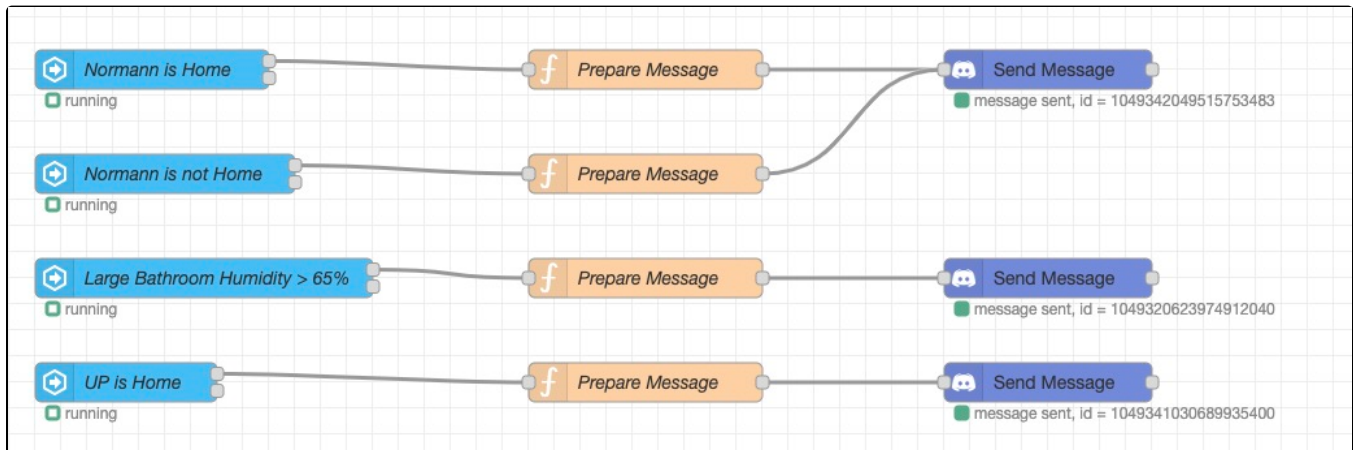
The setup is a bit complicated - follow the links on <https://github.com/Markoudstaal/node-red-contrib-discord-advanced>

Before sending the Discord Message, a "function" node will set the message (msg) - that is the Yellow/orange "Prepare Message":



The "msg.channel" is an ID and can be found in the URL for the Channel when running Discord in a Browser, it's NOT a string like "#home-assistant". Thank for Support from the coders.

And so, a few messages to Discord:



Result:

