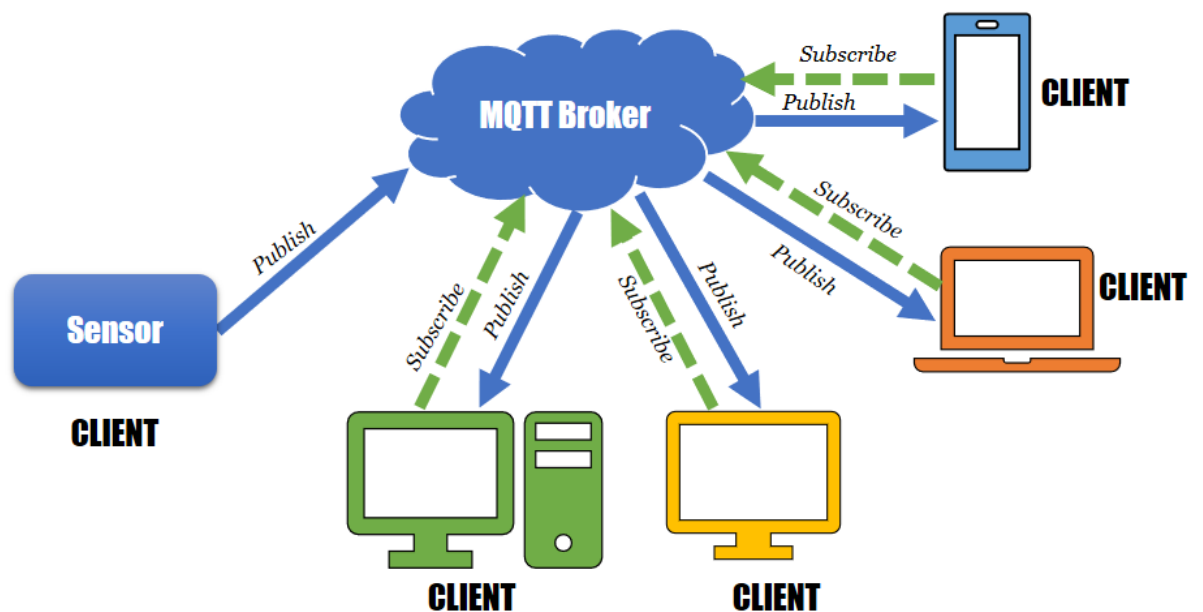


2020 - 2021

MQTT

Implémentation du MQTT pour
une utilisation uplink/downlink

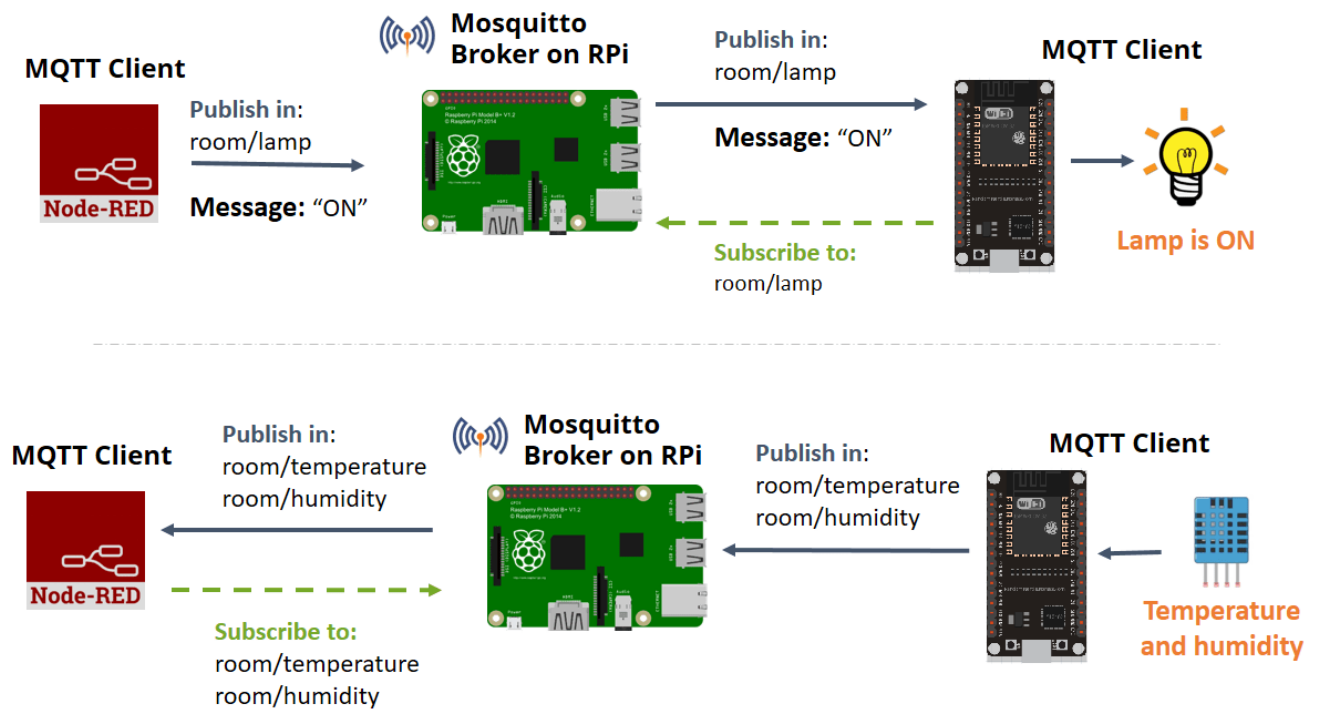


MQTT

Table des matières

Schéma synopsis.....	2
Liste du matériel.....	2
Branchement des différents modules du projet	3
Installation de NODE-RED.....	4
Installation de Mosquitto Broker (Raspberry Pi).....	6
Configuration de NODE-RED après l'installation afin d'établir une communication MQTT	8
Préparation du Arduino IDE	14

Schéma synopsis

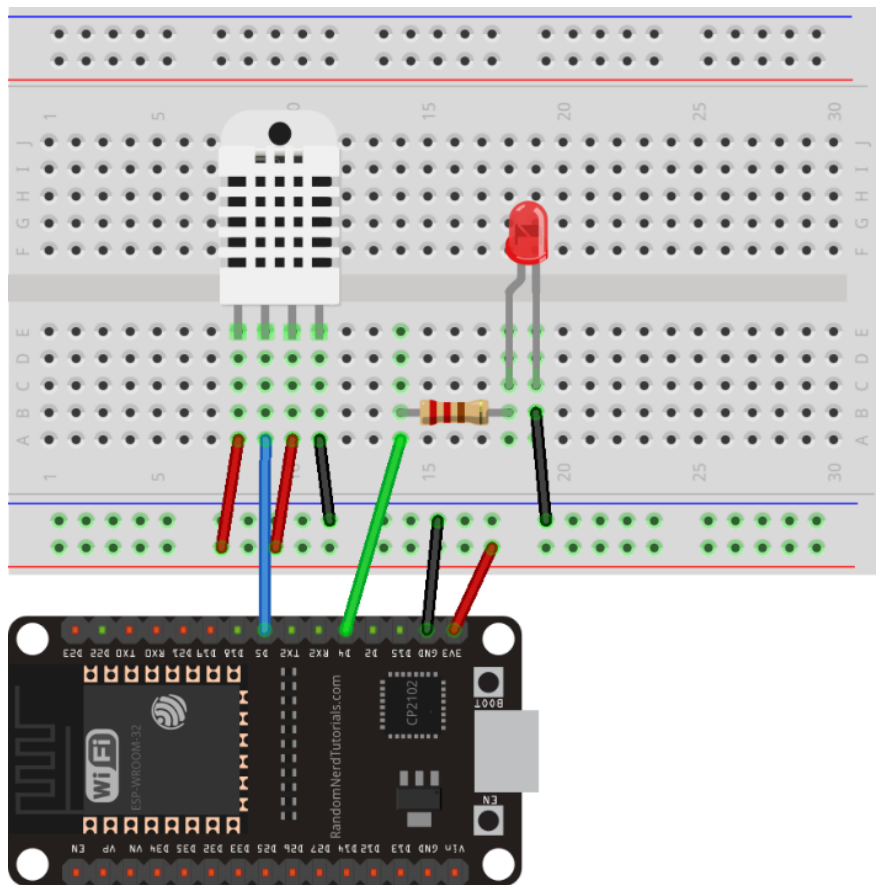


Liste du matériel

Voici la liste du matériel nécessaire pour réaliser le projet :

- Raspberry Pi
- ESP32
- LED
- DHT22

Branchement des différents modules du projet

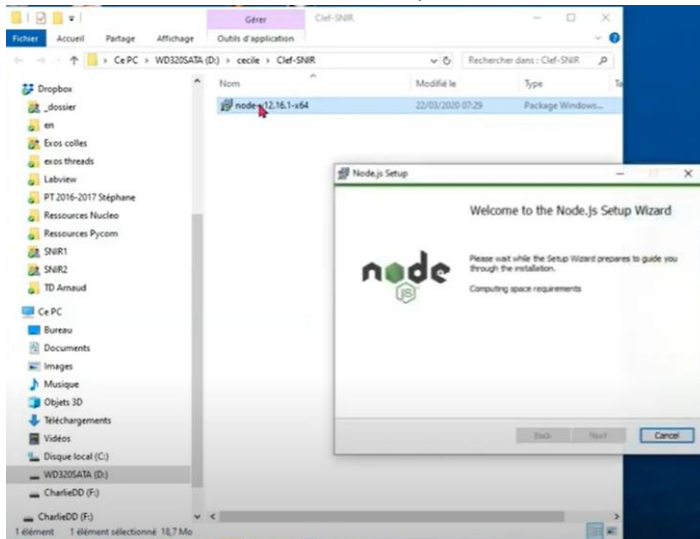


fritzing

Installation de NODE-RED

1) Installation de Node.js

- Cliquer sur ce lien <https://nodejs.org/en/> et installer la version LTS (Long-Term support)
- Exécuter l'installateur (le fichier qui vient d'être téléchargé)



- Suivre les différentes étapes en maintenant une configuration par défaut
 - Vérifier l'installation :
 - Ouvrir un terminal
 - Taper `node -v` et vérifier la version installée
 - Faire de même pour vérifier l'installation du gestionnaire de paquet npm
- Taper `npm -v`

```
Invite de commandes
Microsoft Windows [version 10.0.18362.720]
(c) 2019 Microsoft Corporation. Tous droits réservés.

C:\Users\cecile>node -v
v12.16.1

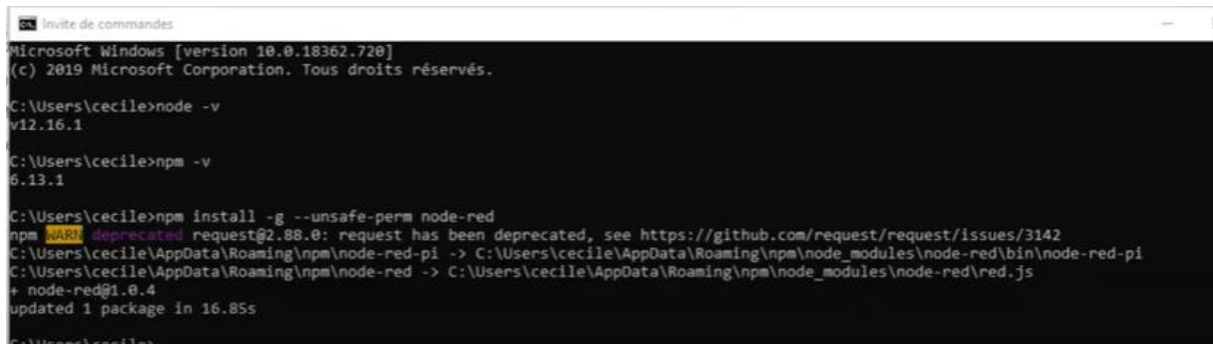
C:\Users\cecile>npm -v
6.13.1

C:\Users\cecile>
```

2) Installation de Node-Red via npm (gestionnaire de paquet)

- Taper dans le terminal la commande suivante :

npm install -g --unsafe-perm node-red



```
Invite de commandes
Microsoft Windows [version 10.0.18362.720]
(c) 2019 Microsoft Corporation. Tous droits réservés.

C:\Users\cecile>node -v
v12.16.1

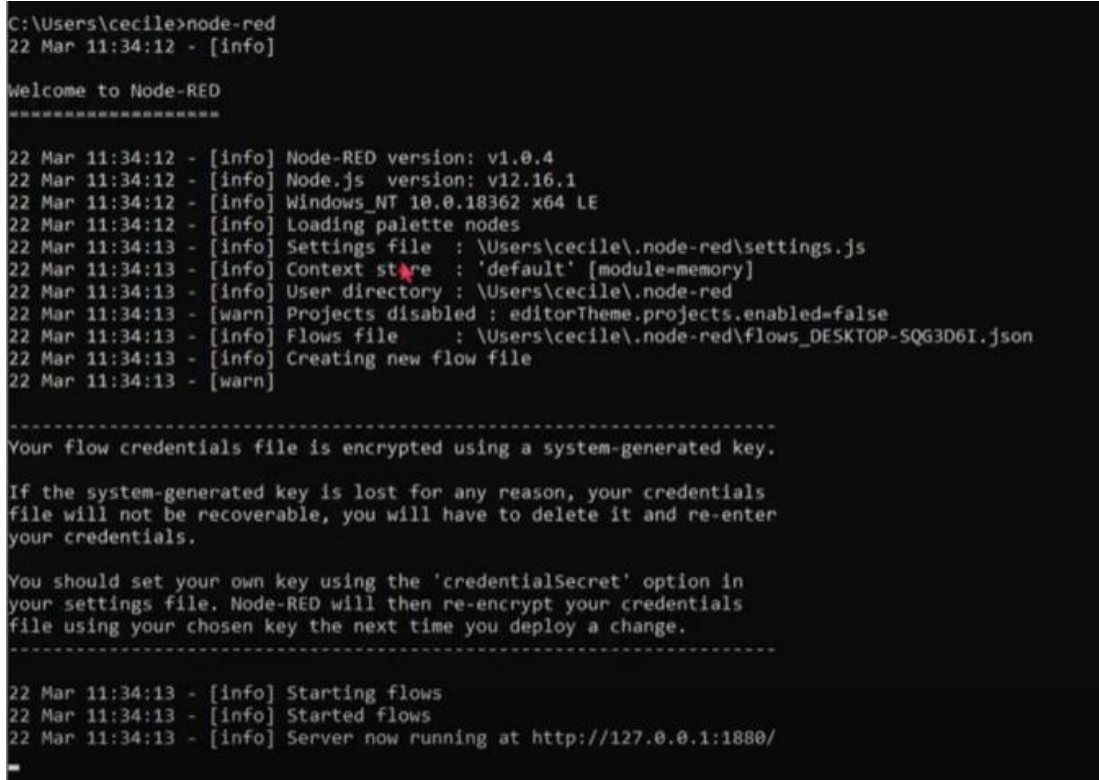
C:\Users\cecile>npm -v
6.13.1

C:\Users\cecile>npm install -g --unsafe-perm node-red
npm WARN deprecated request@2.88.0: request has been deprecated, see https://github.com/request/request/issues/3142
C:\Users\cecile\AppData\Roaming\npm\node-red -> C:\Users\cecile\AppData\Roaming\npm\node_modules\node-red\bin\node-red-pi
C:\Users\cecile\AppData\Roaming\npm\node-red -> C:\Users\cecile\AppData\Roaming\npm\node_modules\node-red\red.js
+ node-red@1.0.4
updated 1 package in 16.85s

C:\Users\cecile>
```

- Exécuter Node-Red en tapant la commande suivante :

node-red



```
C:\Users\cecile>node-red
22 Mar 11:34:12 - [info]

Welcome to Node-RED
=====

22 Mar 11:34:12 - [info] Node-RED version: v1.0.4
22 Mar 11:34:12 - [info] Node.js version: v12.16.1
22 Mar 11:34:12 - [info] Windows_NT 10.0.18362 x64 LE
22 Mar 11:34:12 - [info] Loading palette nodes
22 Mar 11:34:13 - [info] Settings file : \Users\cecile\.node-red\settings.js
22 Mar 11:34:13 - [info] Context store : 'default' [module=memory]
22 Mar 11:34:13 - [info] User directory : \Users\cecile\.node-red
22 Mar 11:34:13 - [warn] Projects disabled : editorTheme.projects.enabled=false
22 Mar 11:34:13 - [info] Flows file : \Users\cecile\.node-red\flows_DESKTOP-SQG3D6I.json
22 Mar 11:34:13 - [info] Creating new flow file
22 Mar 11:34:13 - [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.
-----

22 Mar 11:34:13 - [info] Starting flows
22 Mar 11:34:13 - [info] Started flows
22 Mar 11:34:13 - [info] Server now running at http://127.0.0.1:1880/
=
```

- Accéder à l'interface graphique à l'aide d'un navigateur en accédant à l'adresse indiquée à la dernière ligne.

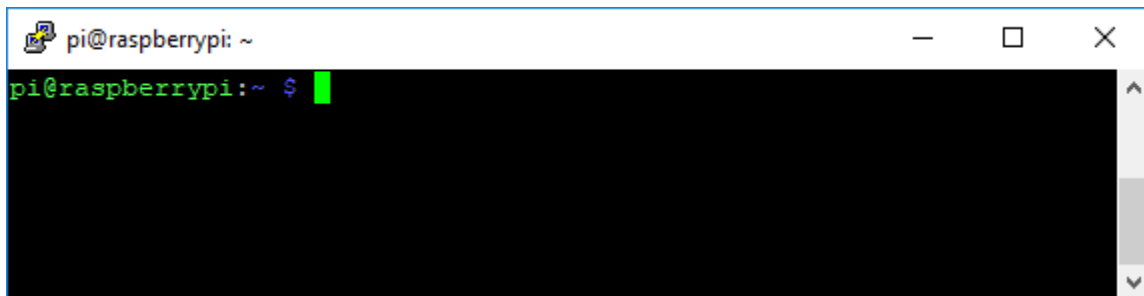
Comme vous pouvez le voir notre serveur tourne en local sur le port 1880.

On peut y accéder à l'adresse <http://127.0.0.1:1880/>

Installation de Mosquitto Broker (Raspberry Pi)

1. Installation

- Ouvrez une nouvelle fenêtre de terminal Raspberry Pi :



- Pour installer Mosquitto Broker, entrez les commandes suivantes :
pi @ raspberry : ~ \$ *sudo apt update*
pi @ raspberry : ~ \$ *sudo apt install -y mosquitto mosquitto-clients*

Vous devrez taper **Y** et appuyer sur **Entrée** pour confirmer l'installation.

- Pour que Mosquitto démarre automatiquement au démarrage, entrez :
pi @ raspberry : ~ \$ *sudo systemctl enable mosquitto.service*

2. Tester l'installation

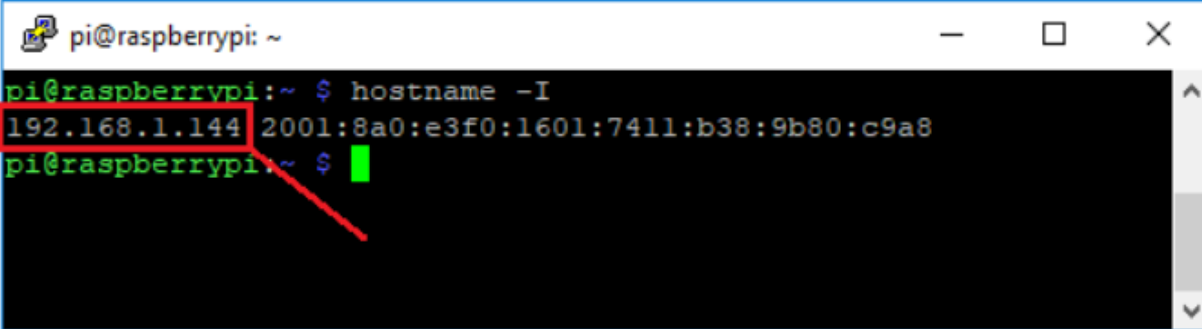
- Taper la commande suivante :
pi@raspberrypi: ~ \$ *mosquitto -v*
- Vérifier que la version de Mosquitto qui est cours d'exécution dans votre raspberry est supérieur à 1.4.X.

Remarque : parfois, la commande *mosquitto -v* affiche un message d'avertissement indiquant « *Erreur : Adresse déjà utilisée* ». Ce message d'avertissement signifie que votre Mosquitto Broker est déjà en cours d'exécution, alors ne vous inquiétez pas pour cela.

3. Récupération de l'adresse IP de la Raspberry Pi

- Pour utiliser le courtier Mosquitto plus tard sur le projet, vous aurez besoin de votre adresse IP Raspberry Pi.
Pour récupérer votre adresse IP Raspberry Pi, tapez la commande suivante dans la fenêtre de votre terminal :

pi@rasberry : ~ \$ `hostname -I`



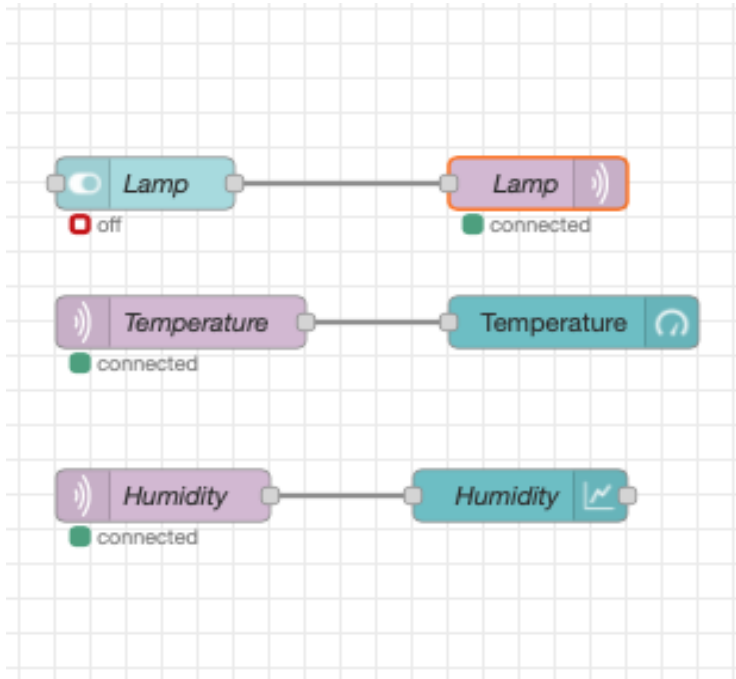
```
pi@rasberry: ~  
pi@rasberry:~$ hostname -I  
192.168.1.144 2001:8a0:e3f0:1601:7411:b38:9b80:c9a8  
pi@rasberry:~$
```

- Dans notre cas, l'adresse IP du Raspberry Pi est **192.168.1.144** .
Enregistrez votre adresse IP Raspberry Pi car vous en aurez besoin dans la suite du projet.

Configuration de NODE-RED après l'installation afin d'établir une communication MQTT

Dans cette partie nous allons configurer et établir une communication MQTT à partir de node-red.

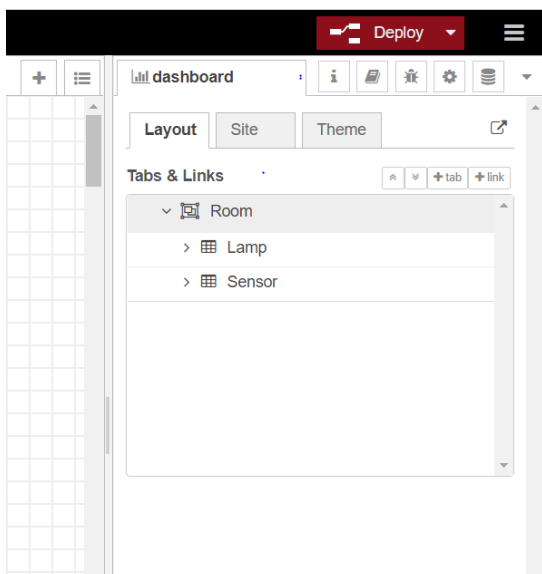
Voici le résultat final attendu :



1. Création des Topics

- Sur node-red local on sélectionne **dashboard** puis **Layout** pour créer une tab en cliquant sur **+tab**, la tab constituée de groupe de capteur

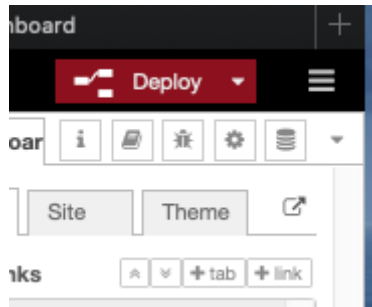
En positionnant la souris sur la tab créée des options vont apparaître, cliquez sur **+group** et vous pouvez ajouter vos périphériques externes (actionneurs, capteurs...).



Nommer vos tabs avec les mêmes noms que sur l'image ci-dessus afin de pouvoir utiliser notre projet.

2. Créer les Nodes sur node-red

- Télécharger la palette **node-red-dashboard**

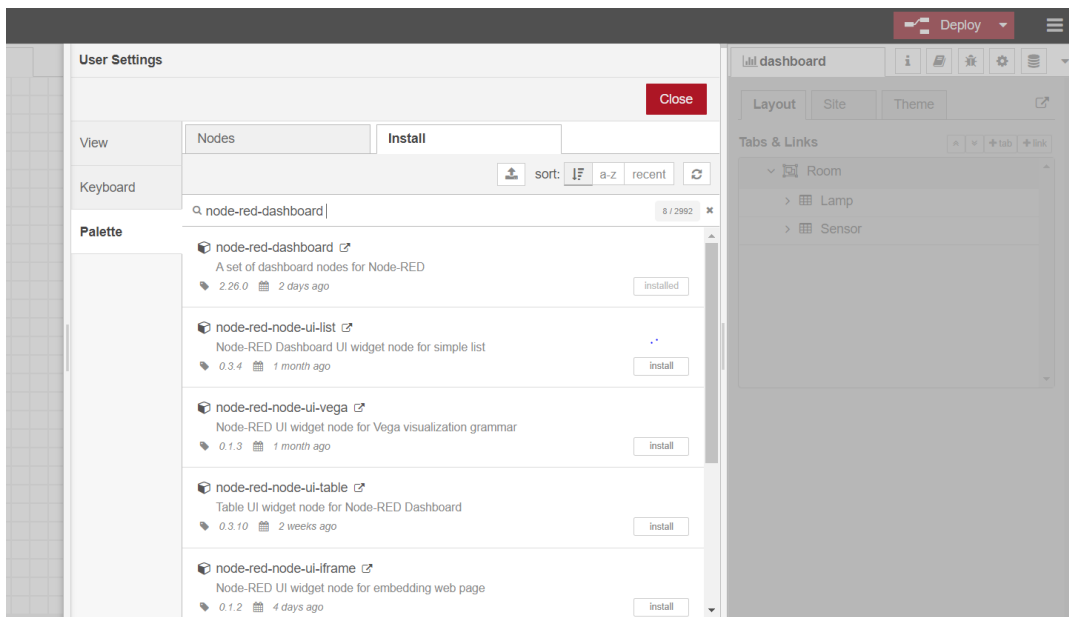


- Cliquez sur le bouton à droite de Deploy

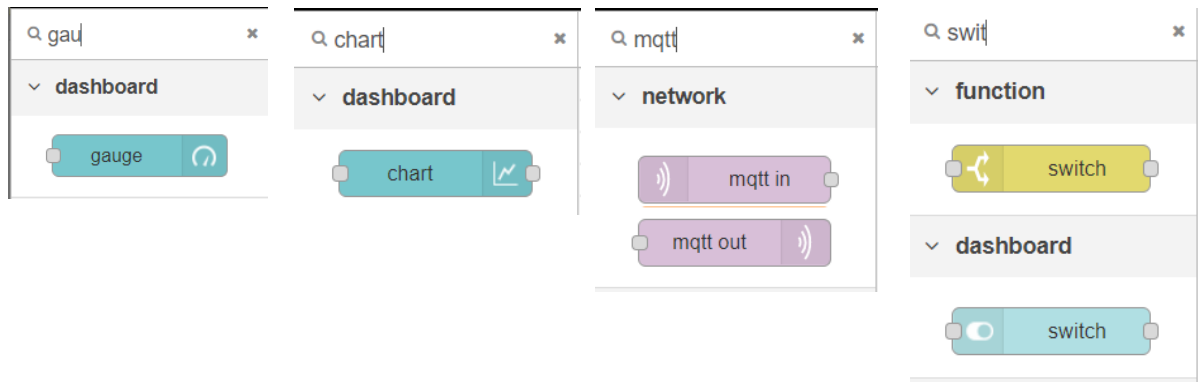


→ manage palette → install → node-red-dashboard

- Installer le paquet



- Maintenant vous pouvez importer les nodes à l'aide de la barre de recherche en haut à gauche sur Node-red et les placer.



- Configuration des nodes qui devront faire appel aux tab et group créer précédemment.
 - Cliquez sur le **switch** et configurez-le comme ci-dessous :

The screenshot shows the 'Edit switch node' configuration window. It has tabs for 'Properties', 'Code', and 'Help'. The 'Properties' tab is active.
 Fields include:
 - Group: '[Room] Lamp' (with a dropdown arrow and edit icon)
 - Size: 'auto'
 - Label: 'Lamp'
 - Tooltip: 'optional tooltip'
 - Icon: 'Default' (with a dropdown arrow)
 - A checked checkbox: 'Pass through msg if payload matches new state:'
 - 'When clicked, send:' section with:
 - 'On Payload': 'on' (with a dropdown arrow)
 - 'Off Payload': 'off' (with a dropdown arrow)
 - Topic: (empty text field)
 - Name: 'Lamp'
 At the top right are 'Delete', 'Cancel', and 'Done' buttons.

- Cliquez sur le **MQTT OUT** et configurez-le comme ci-dessous :

The screenshot shows the 'Edit mqtt out node' configuration window. It has tabs for 'Properties', 'Code', and 'Help'. The 'Properties' tab is active.
 Fields include:
 - Server: '192.168.1.14:1883' (with a dropdown arrow and edit icon)
 - Topic: 'room/lamp'
 - QoS: (dropdown arrow)
 - Retain: (checkbox)
 - Name: 'Lamp'
 A yellow tip box at the bottom says: 'Tip: Leave topic, qos or retain blank if you want to set them via msg properties.'
 At the top right are 'Delete', 'Cancel', and 'Done' buttons.

Au niveau du serveur utilisez l'adresse IP de votre Raspberry sur laquelle vous avez installé mosquitto. Le topic à une liaison avec tab et group et le code source.

- Cliquez sur le **Gauge** et configurez-le comme ci-dessous :

Edit gauge node

Delete Cancel Done

Properties

Group [Room] Sensor

Size auto

Type Gauge

Label Temperature

Value format {{value}}

Units units

Range min 0 max 40

Colour gradient

Sectors 0 ... optional ... optional ... 40

Name

- Cliquez sur le **MQTT IN** et configurez-le comme ci-dessous :
Le serveur représente toujours l'adresse IP de la Raspberry.

Edit mqtt in node

Delete Cancel Done

Properties

Server 192.168.1.14:1883

Topic room/humidity

QoS 2

Output auto-detect (string or buffer)

Name Humidity

- Cliquez sur le **Chart** et configurez-le comme ci-dessous :

Edit chart node

Delete Cancel Done

Properties

Group [Room] Sensor

Size auto

Label Humidity

Type Line chart ☐ enlarge points

X-axis last 1 hours OR 1000 points

X-axis Label HH:mm:ss ☐ as UTC

Y-axis min 0 max 100

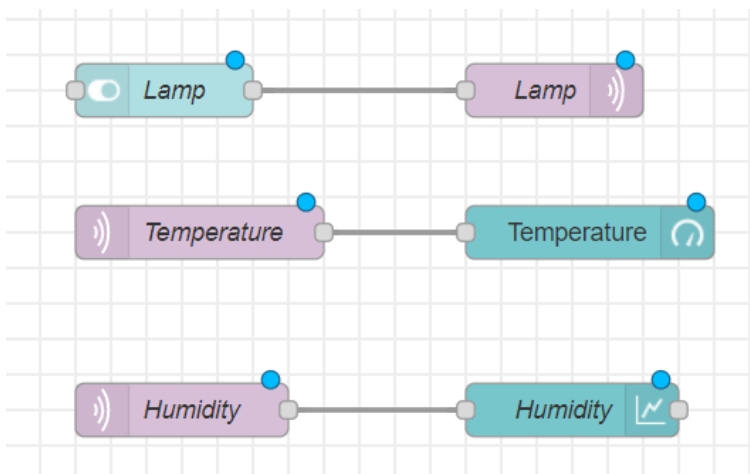
Legend None Interpolate linear

Series Colours

Blank label display this text before valid data arrives

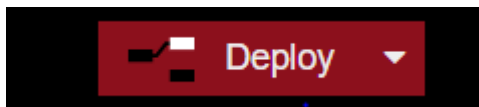
Name Humidity


- Voici le résultat attendu en prenant soin de connecter les nodes :

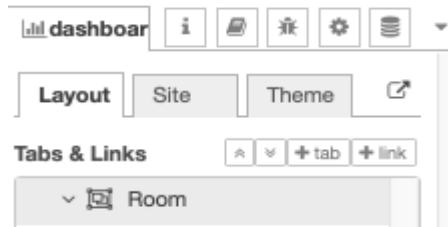


Une fois terminé, votre node-red sera prêt

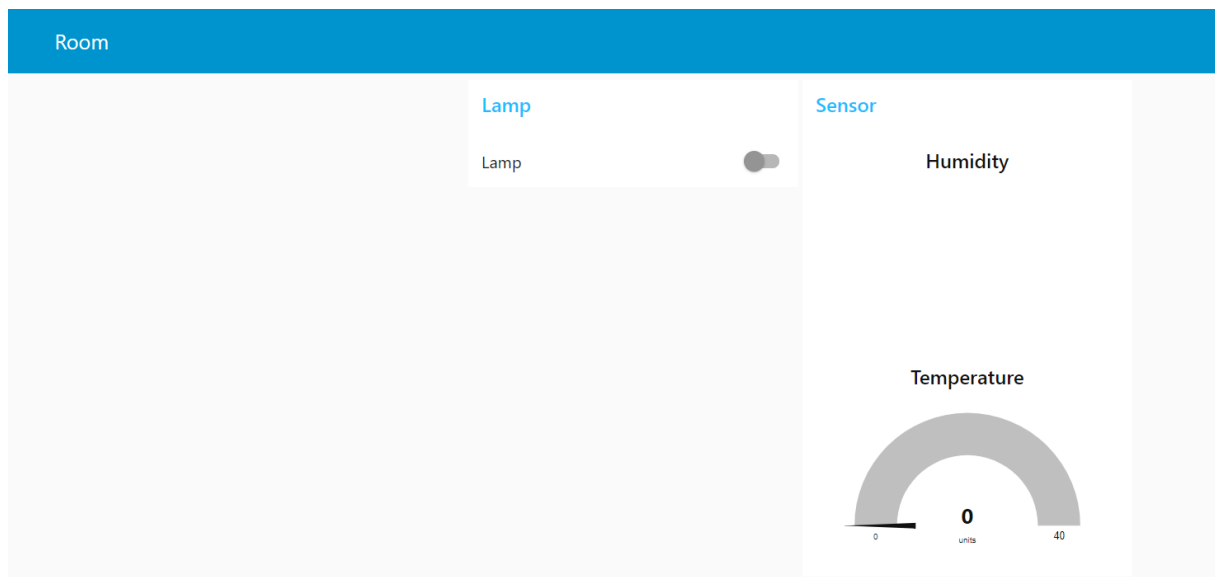
- Cliquer sur « Deploy » afin de déployer l'application



- Visualiser l'aperçu du Dashboard en cliquant sur ce bouton dans le menu **dashboard** : 



Voici l'aperçu attendu de notre dashboard :



Préparation du Arduino IDE

Nous programmerons l'ESP32 en utilisant l'IDE Arduino. Pour télécharger du code sur votre ESP32 à l'aide de l'IDE Arduino, [vous devez installer l'ESP32](#) . Vous devrez également installer deux bibliothèques supplémentaires pour que tout soit prêt pour votre ESP32.

1/ La bibliothèque PubSubClient fournit un client pour faire des messages simples de publication/abonnement avec un serveur qui prend en charge MQTT (permet essentiellement à votre ESP32 de parler avec Node-RED).

- [Installer la librairie PubSubClient](#)
- Dézipper le fichier et modifier son nom en pubsubclient
- Déplacer le au dossier des librairies arduino

2/ La bibliothèque de capteurs DHT permet d'utiliser facilement n'importe quel capteur DHT pour lire la température et l'humidité avec vos cartes ESP32 ou Arduino.

- [Installer la librairie DHT](#)
- Dézipper le fichier et modifier son nom en DHT
- Déplacer le au dossier des librairies arduino

Vous pouvez télécharger le code source pour l'utilisation du MQTT, LED, Mosquitto et d'un DHT11 avec un ESP32 sur le lien drive suivant [sur le lien drive suivant](#).

Au moment de téléverser le code vérifiez que vous ciblez la bonne carte sur Arduino.

Sur le drive vous trouverez aussi un [fichier JSON](#) que vous pouvez importer directement sur node-red et avoir une configuration identique à celle décrite ci-dessus dans la partie configuration de node-red.