

1. Rapport de projet

2. Sommaire

- [1. Rapport de projet](#)
- [2. Sommaire](#)
- [3. Introduction](#)
- [4. Installation physique](#)
 - [4.1. Nidus](#)
 - [4.2. Volt](#)
- [5. Schéma de principe](#)
- [6. OS](#)
 - [6.1. Ubuntu](#)
 - [6.2. Raspbian](#)
 - [6.3. Première installation](#)
 - [6.4. Seconde installation Ubuntu Server](#)
 - [6.4.1. Configuration post installation](#)
 - [6.4.2. Installation Apache](#)
 - [6.4.3. Script MQTT](#)
- [7. Node-RED](#)
 - [7.1. Installation](#)
 - [7.2. Configuration](#)
 - [7.2.1. Installation des plugins](#)
 - [7.2.2. Sécurisation de Node-Red](#)
 - [7.2.3. Suivi Git](#)
- [8. Gatling](#)
 - [8.1. Installation](#)
 - [8.1.1. Prerequis](#)
 - [8.1.2. Download](#)
 - [8.2. Vérification de l'installation](#)
 - [8.3. Configuration](#)
- [9. Apache et Site Web](#)
 - [9.1. Installation](#)
 - [9.2. Mise en place d'un site Web](#)
- [10. MQTT](#)
 - [10.1. Instalaion de Mosquitto sur Nidus](#)
 - [10.2. Ouverture des port sur Nidus](#)
 - [10.3. Script MQTT](#)
 - [10.4. Installation](#)
 - [10.5. Utilisation du script](#)
 - [10.5.1. Vérification](#)
- [11. INA219](#)
 - [11.1. Installation physique](#)
 - [11.1.1. Branchement SANS VOLT](#)

- 11.1.2. Branchement AVEC VOLT
 - 11.2. Vérification de la présence du INA219
 - 11.3. Obtention des données
 - 11.3.1. Test avec le script python A vide
- 12. Noeud Node-Red
 - 12.1. INA219
 - 12.2. Monitoring
 - 12.3. Dashboard
 - 12.4. PDF
 - 12.4.1. Base
 - 12.5. Images de graphiques et de tableaux
- 13. Stress Test V1.0
 - 13.1. Écran d'Accueil
 - 13.2. En Exécution
 - 13.3. Résultat
 - 13.4. Purge
- 14. Sources

3. Introduction

Le système sera conçu pour simuler des requêtes HTTP réalistes à l'aide de Gatling, mesurer la consommation électrique en utilisant l'INA219 connecté via le bus I2C, et collecter les mesures de performance à l'aide de Node-RED. Les rapports générés fourniront des informations détaillées sur les performances du système testé, y compris le temps de réponse, la consommation d'énergie par requête, l'utilisation du processeur, etc.

4. Instalation physique

4.1. Nidus



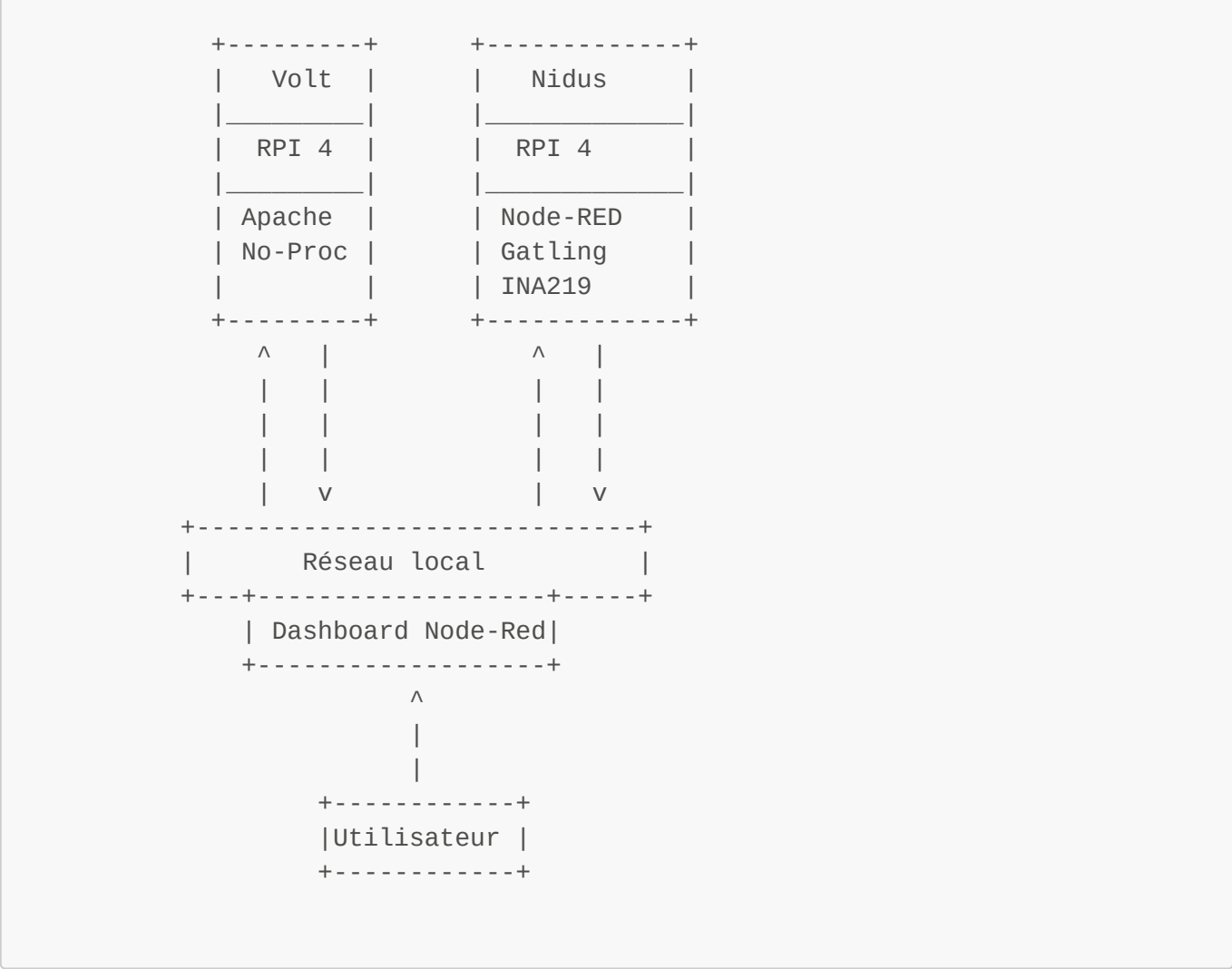
{width=100%}

4.2. Volt



{width=100%}

5. Shéma de principe



6. OS

Dans le cadre de ce projet il y auras plusieurs OS utilisée mais pour débiter nous allons utiliser Ubuntu.

6.1. Ubuntu

Ubuntu est un OS largement utilisé pour les serveurs et les ordinateurs de bureau. Ubuntu est livré avec un ensemble d'outils de développement et de productivité, y compris un navigateur Web, un éditeur de texte, des logiciels de programmation, des outils de calcul, des jeux et des logiciels de productivité. Ubuntu est un environnement de bureau léger et réactif conçu pour les ordinateurs de bureau et les serveurs.

6.2. Raspbian

Raspbian est un système d'exploitation libre basé sur Debian optimisé pour le Raspberry Pi. Depuis 2015, Raspbian est fourni avec un ensemble d'outils appelé Pixel. Pixel est un environnement de bureau qui comprend un navigateur Web, un éditeur de texte, des logiciels de programmation, des outils de calcul, des jeux et des logiciels de productivité. Pixel est un environnement de bureau léger et réactif conçu pour les ordinateurs monocarte Raspberry Pi.

6.3. Première instalation

Dans un premier temps nous allons installer Ubuntu en version desktop sur Volt. La raison derrière ce choix est que pour tester au plus vite tous les concepts du projet il est plus simple de travailler sur un environnement de bureau.

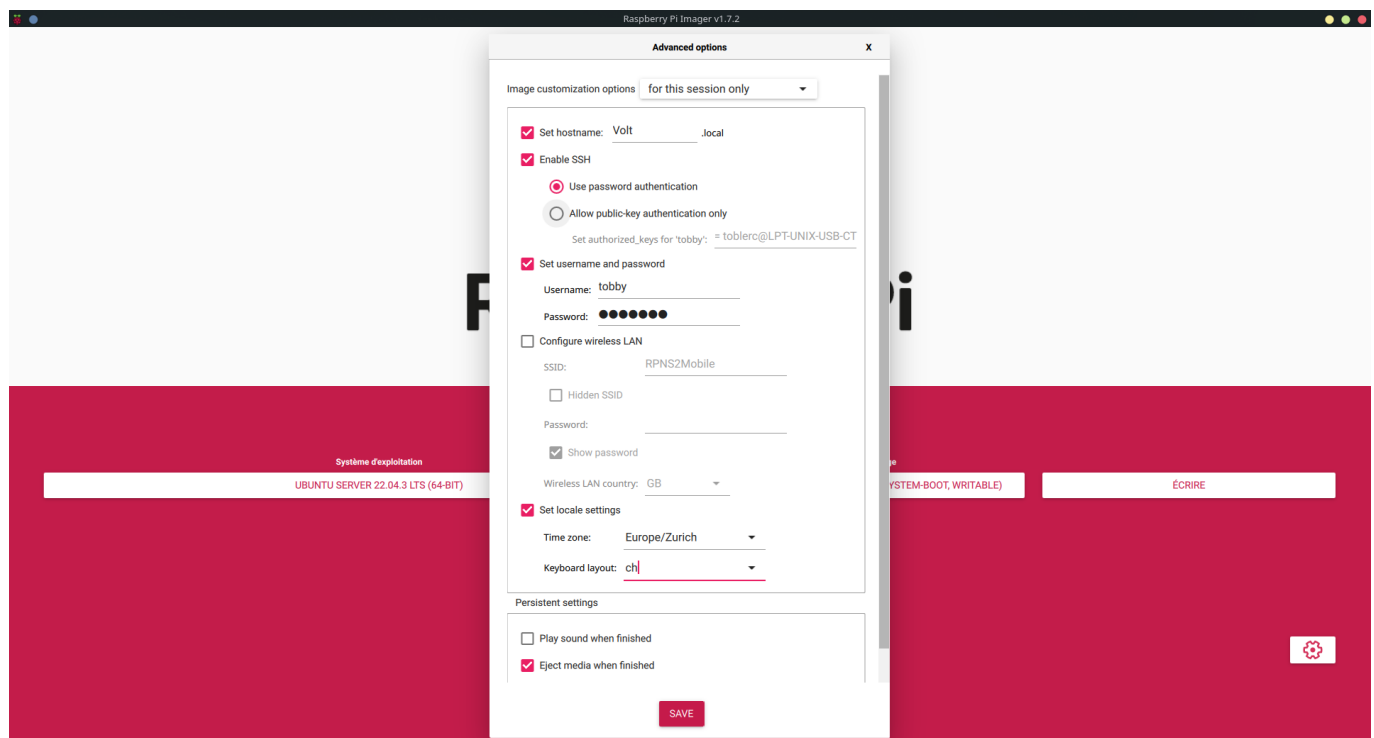
Sur Nidus c'est Raspbian en version desktop qui sera installé pour les mêmes raisons que pour Volt.

La raison derrière ce choix est que Ubuntu core est plus léger que raspbian et plus utilisée pour les serveur web et que Raspbian est plus performant sur les Raspberry pi que Ubuntu core. Dernier point important, étant donné que l'INA219 sera branché à Nidus, il est plus simple de mettre raspbian sur Nidus pour avoir accès au GPIO.

Dans un second temps pour avoir des mesures plus précises nous allons installer des versions core de Ubuntu et de raspbian.

Adresse IP de Volt : 157.26.228.130 Adresse IP de Nidus : 157.26.251.158

6.4. Seconde installation Ubuntu Server



{width=100%}

6.4.1. Configuration post installation

```
toblerc@LPT-UNIX-USB-CT:~$ ssh toby@157.26.228.77
The authenticity of host '157.26.228.77 (157.26.228.77)' can't be
established.
ED25519 key fingerprint is
SHA256:/5raLLKqk0A4AnFWnLP9bagNS3zKE9rFPqn5vA5pc+M.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '157.26.228.77' (ED25519) to the list of known
hosts.
tobby@157.26.228.77's password:
```

```
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-1034-raspi aarch64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Wed Aug 23 09:29:06 CEST 2023

System load:  2.43994140625      Temperature:           39.4 C
Usage of /:   4.0% of 58.36GB    Processes:             158
Memory usage: 7%                Users logged in:       0
Swap usage:   0%                IPv4 address for eth0: 157.26.228.77

Expanded Security Maintenance for Applications is not enabled.

25 updates can be applied immediately.
12 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Wed Aug 23 09:29:10 2023
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

tobby@Volt:~$ ls -la
total 28> 2022 .bash_logout
-rw-r--r-- 1 toby toby 3771 Jan  6  2022 .bashrc
drwx----- 2 toby toby 4096 Aug 23 09:29 .cache
-rw-r--r-- 1 toby toby  807 Jan  6  2022 .profile
drwx----- 2 toby toby 4096 Aug 23 09:30 .ssh
tobby@Volt:~$ cd .ssh/
tobby@Volt:~/.ssh$ ls -la
total 8
drwx----- 2 toby toby 4096 Aug 23 09:30 .
drwxr-x--- 4 toby toby 4096 Aug 23 09:30 ..
-rw----- 1 toby toby    0 Aug 23 09:30 authorized_keys
tobby@Volt:~/.ssh$ sudo vi authorized_keys
[sudo] password for toby:
tobby@Volt:~/.ssh$ exit
logout
Connection to 157.26.228.77 closed.
toblerc@LPT-UNIX-USB-CT:~$ ssh toby@157.26.228.77
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-1034-raspi aarch64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Wed Aug 23 09:31:44 CEST 2023
```



```
System load: 0.490234375    Temperature: 40.9 C
Usage of /: 4.1% of 58.36GB Processes: 153
Memory usage: 6%           Users logged in: 1
Swap usage: 0%             IPv4 address for eth0: 157.26.228.77
```

Expanded Security Maintenance **for** Applications is not enabled.

25 updates can be applied immediately.

12 of these updates are standard security updates.

To see these additional updates run: `apt list --upgradable`

Enable ESM Apps to receive additional future security updates.

See <https://ubuntu.com/esm> or run: `sudo pro status`

Last login: Wed Aug 23 09:30:02 2023 from 157.26.215.31

```
tobby@Volt:~$ sudo apt update && sudo apt upgrade -y && sudo apt dist-
upgrade -y && sudo apt auto-remove -y
```

```
[sudo] password for toby:
```

```
[...]
```

```
tobby@Volt:~$ sudo apt update && sudo apt upgrade -y && sudo apt dist-
upgrade -y && sudo apt auto-remove -y
```

```
Hit:1 http://ports.ubuntu.com/ubuntu-ports jammy InRelease
```

```
Get:2 http://ports.ubuntu.com/ubuntu-ports jammy-updates InRelease [119 kB]
```

```
Hit:3 http://ports.ubuntu.com/ubuntu-ports jammy-backports InRelease
```

```
Get:4 http://ports.ubuntu.com/ubuntu-ports jammy-security InRelease [110
kB]
```

```
Fetch 229 kB in 2s (100 kB/s)
```

```
Reading package lists... Done
```

```
Building dependency tree... Done
```

```
Reading state information... Done
```

```
All packages are up to date.
```

```
Reading package lists... Done
```

```
Building dependency tree... Done
```

```
Reading state information... Done
```

```
Calculating upgrade... Done
```

```
#
```

```
# You can verify the status of security fixes using the `pro fix` command.
```

```
# E.g., a recent Ruby vulnerability can be checked with: `pro fix USN-6219-
1`
```

```
# For more detail see: https://ubuntu.com/security/notices/USN-6219-1
```

```
#
```

```
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

```
Reading package lists... Done
```

```
Building dependency tree... Done
```

```
Reading state information... Done
```

```
Calculating upgrade... Done
```

```
#
```

```
# You can verify the status of security fixes using the `pro fix` command.
```

```
# E.g., a recent Ruby vulnerability can be checked with: `pro fix USN-6219-
1`
```

```
# For more detail see: https://ubuntu.com/security/notices/USN-6219-1
```

```
#
```



```
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

6.4.2. Instalation Apache

```
tobby@Volt:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils bzip2 libapr1 lib
[...]
```

```
toblerc@LPT-UNIX-USB-CT:~$ scp -r /home/toblerc/Documents/ES_2024/banc-de-
mesures-de-la-consommation-electrique/siteWeb/www/html
tobby@157.26.228.77:~/home/tobby
[...]
```

```
toblerc@LPT-UNIX-USB-CT:~$
tobby@Volt:~$ sudo cp -r /home/tobby/html /var/www/
```

6.4.3. Script MQTT

```
toblerc@LPT-UNIX-USB-CT:~/Documents/ES_2024/banc-de-mesures-de-la-
consommation-electrique$ scp ./mqtt.sh toby@157.26.228.77:/home/tobby
mqtt.sh
100% 2522      1.7MB/s   00:00
toblerc@LPT-UNIX-USB-CT:~/Documents/ES_2024/banc-de-mesures-de-la-
consommation-electrique$
tobby@Volt:~$ sudo cp ./mqtt.sh /usr/local/bin/
tobby@Volt:~$ ls -la /usr/local/bin/
total 12
drwxr-xr-x  2 root root 4096 Aug 23 10:26 .
drwxr-xr-x 10 root root 4096 Aug  7 17:23 ..
-rw-r--r--  1 root root 2522 Aug 23 10:26 mqtt.sh
```

```
tobby@Volt:/usr/local/bin$ sudo ./mqtt.sh
Installation de mosquitto-clients...
Hit:1 http://ports.ubuntu.com/ubuntu-ports jammy InRelease
Get:2 http://ports.ubuntu.com/ubuntu-ports jammy-updates InRelease [119 kB]
```

```
Hit:3 http://ports.ubuntu.com/ubuntu-ports jammy-backports InRelease
Get:4 http://ports.ubuntu.com/ubuntu-ports jammy-security InRelease [110
kB]
Fetched 229 kB in 2s (133 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
[...]
```

7. Node-RED

Node-RED est un outil de programmation visuelle open source conçu pour connecter des périphériques, des API et des services en ligne. Il fournit un éditeur de flux basé sur un navigateur qui facilite la connexion de nœuds en utilisant des liens glisser-déposer qui peuvent être exécutés dans un environnement Node.js. Les nœuds peuvent être des fonctions JavaScript ou des modules npm, tels que node-red-contrib-gpio, node-red-contrib-sqlite, node-red-contrib-modbustcp, etc. Node-RED est livré avec un ensemble de nœuds de base prêts à l'emploi, mais il existe maintenant plus de 2000 nœuds de la communauté qui sont disponibles pour une utilisation.

7.1. Installation

```
tobby@Nidus:~ $ bash <(curl -sL https://raw.githubusercontent.com/node-
red/linux-installers/master/deb/update-nodejs-and-nodered)
Running Node-RED install for user toby at /home/tobby on debian
```

This can take 20-30 minutes on the slower Pi versions - please wait.

Stop Node-RED	✓	
Remove old version of Node-RED	✓	
Remove old version of Node.js	✓	
Install Node.js 18 LTS	✓	v18.17.1 Npm 9.6.7
Clean npm cache	✓	
Install Node-RED core	✓	3.0.2
Move global nodes to local	-	
Npm rebuild existing nodes	✓	
Install extra Pi nodes	✓	
Add shortcut commands	✓	
Update systemd script	✓	

Any errors will be logged to /var/log/nodered-install.log

All done.

You can now start Node-RED with the command node-red-start
or using the icon under Menu / Programming / Node-RED

Then point your browser to localhost:1880 or http://{your_pi_ip-
address}:1880

Started : mer 16 août 2023 14:12:19 CEST

Finished: mer 16 août 2023 14:16:01 CEST

```
*****
*****
```

WARNING

DO NOT EXPOSE NODE-RED TO THE OPEN INTERNET WITHOUT SECURING IT FIRST

Even if your Node-RED doesn't have anything valuable, (automated) attacks will happen and could provide a foothold in your local network

Follow the guide at <https://nodered.org/docs/user-guide/runtime/securing-node-red> to setup security.

ADDITIONAL RECOMMENDATIONS

- Remove the /etc/sudoers.d/010_pi-nopasswd file to require entering your password when performing any sudo/root commands:

```
sudo rm -f /etc/sudoers.d/010_pi-nopasswd
```

- You can customise the initial settings by running:

```
node-red admin init
```

- After running Node-RED for the first time, change the ownership of the settings file to 'root' to prevent unauthorised changes:

```
sudo chown root:root ~/.node-red/settings.js
```

```
*****
*****
```

Would you like to customise the settings now (y/N) ? y

Node-RED Settings File initialisation

```
=====
```

This tool will help you create a Node-RED settings file.

✓ Settings file · /home/tobby/.node-red/settings.js

User Security

```
=====
```

✓ Do you want to setup user security? · Yes

✓ Username · Tobby

✓ Password · *****

✓ User permissions · full access

✓ Add another user? · Yes

✓ Username · FMA

✓ Password · ***** (Pa\$\$w.rd)

✓ User permissions · read-only access

- ✓ Add another user? · Yes
- ✓ Username · BVI
- ✓ Password · ***** (Pa\$\$w.rd)
- ✓ User permissions · read-only access
- ✓ Add another user? · No

Projects

=====

The Projects feature allows you to version control your flow using a local git repository.

- ✓ Do you want to enable the Projects feature? · No

Flow File settings

=====

- ✓ Enter a name for your flows file · flows.json
- ✓ Provide a passphrase to encrypt your credentials file ·

Editor settings

=====

- ✓ Select a theme for the editor. To use any theme other than "default", you will need to install @node-red-contrib-themes/theme-collection in your Node-RED user directory. · dark
- ✓ Select the text editor component to use in the Node-RED Editor · monaco (default)

Node settings

=====

- ✓ Allow Function nodes to load external modules? (functionExternalModules) · Yes

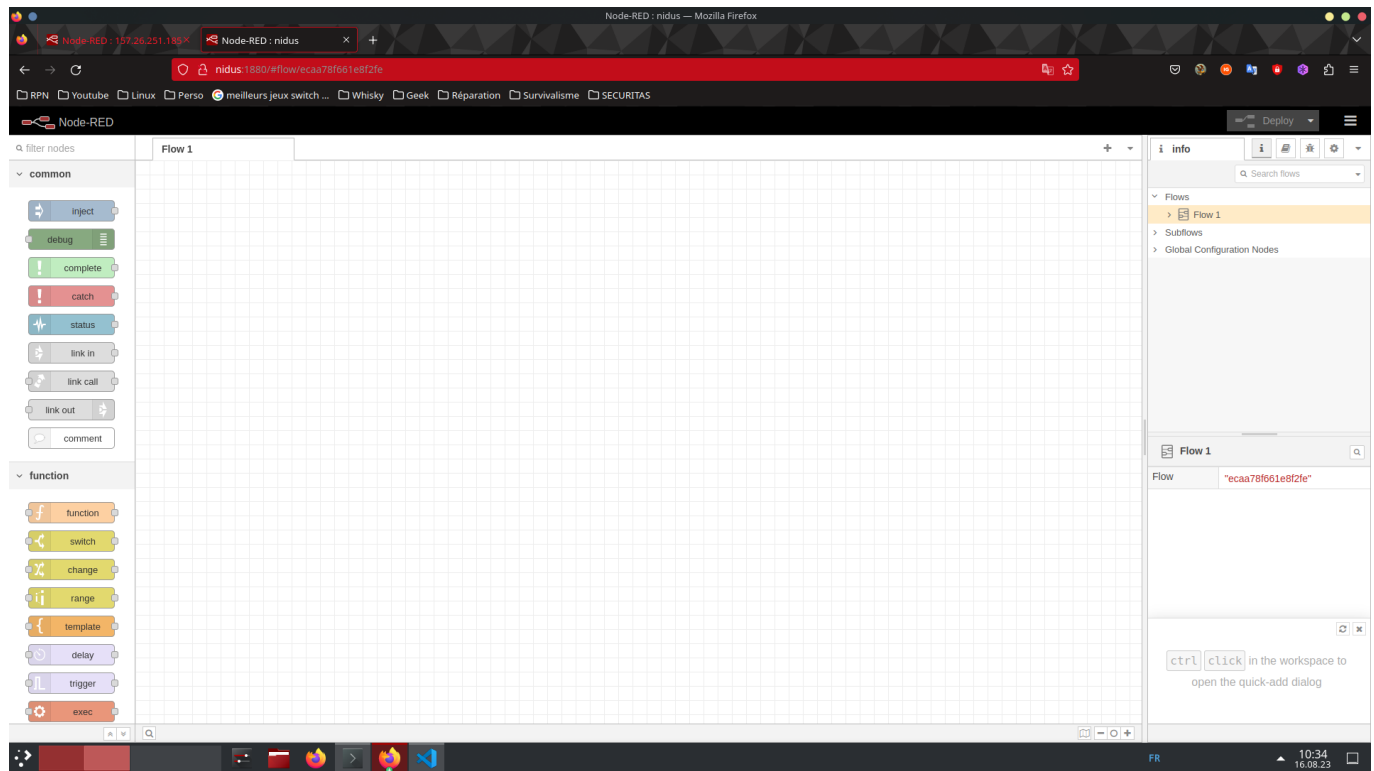
Settings file written to /home/tobby/.node-red/settings.js

To use the 'dark' editor theme, remember to install @node-red-contrib-themes/theme-collection in your Node-RED user directory

```
tobby@Nidus:~ $ sudo systemctl enable nodered.service
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/nodered.service
```

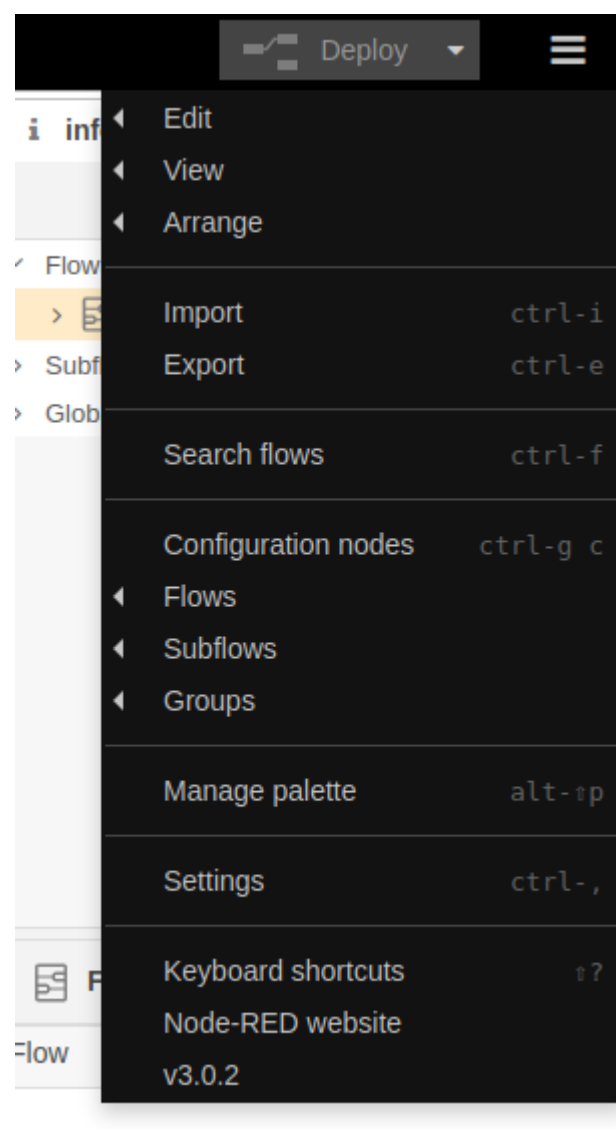
```
→ /lib/systemd/system/nodered.service.
```



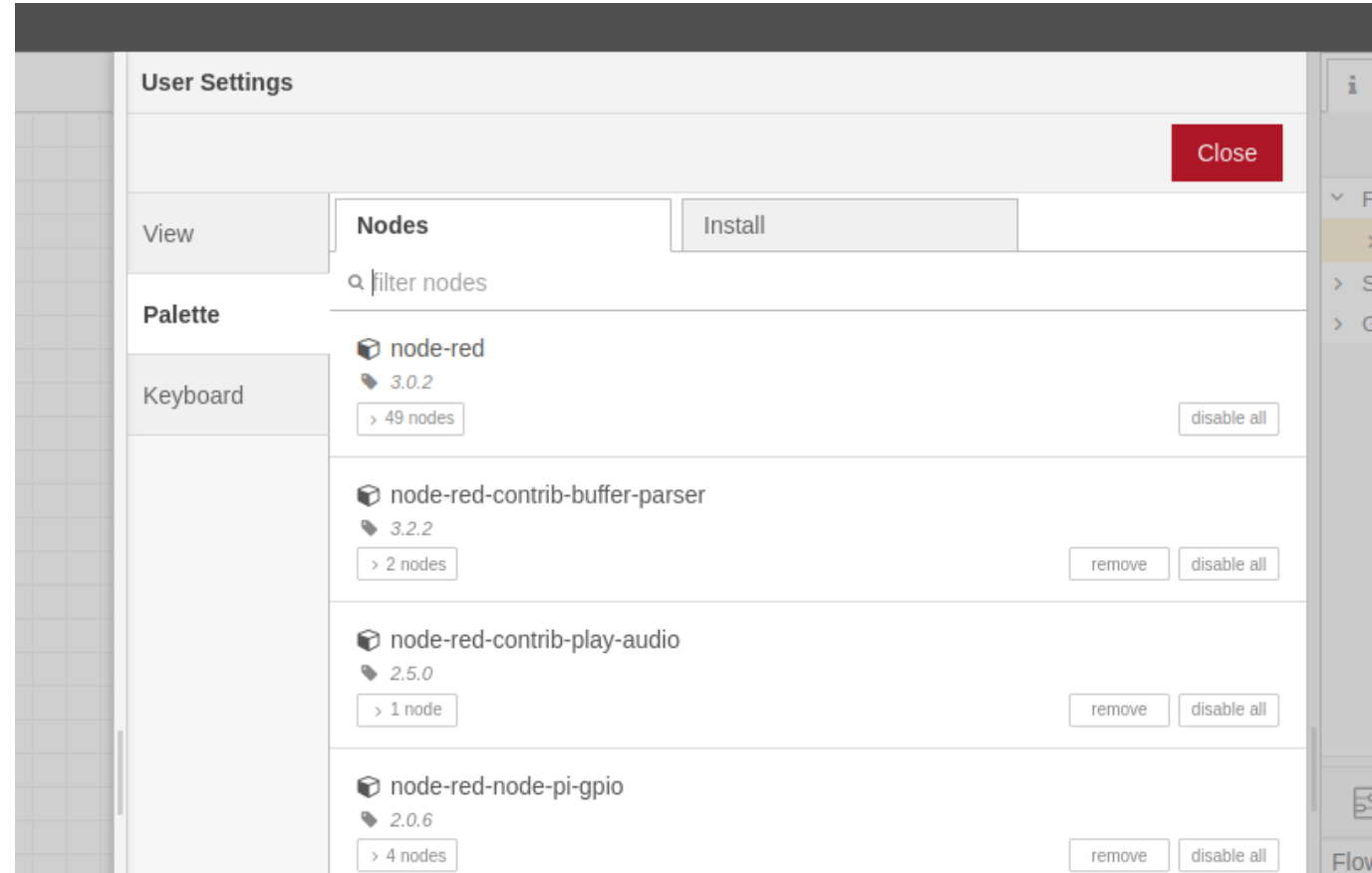
{width=100%}

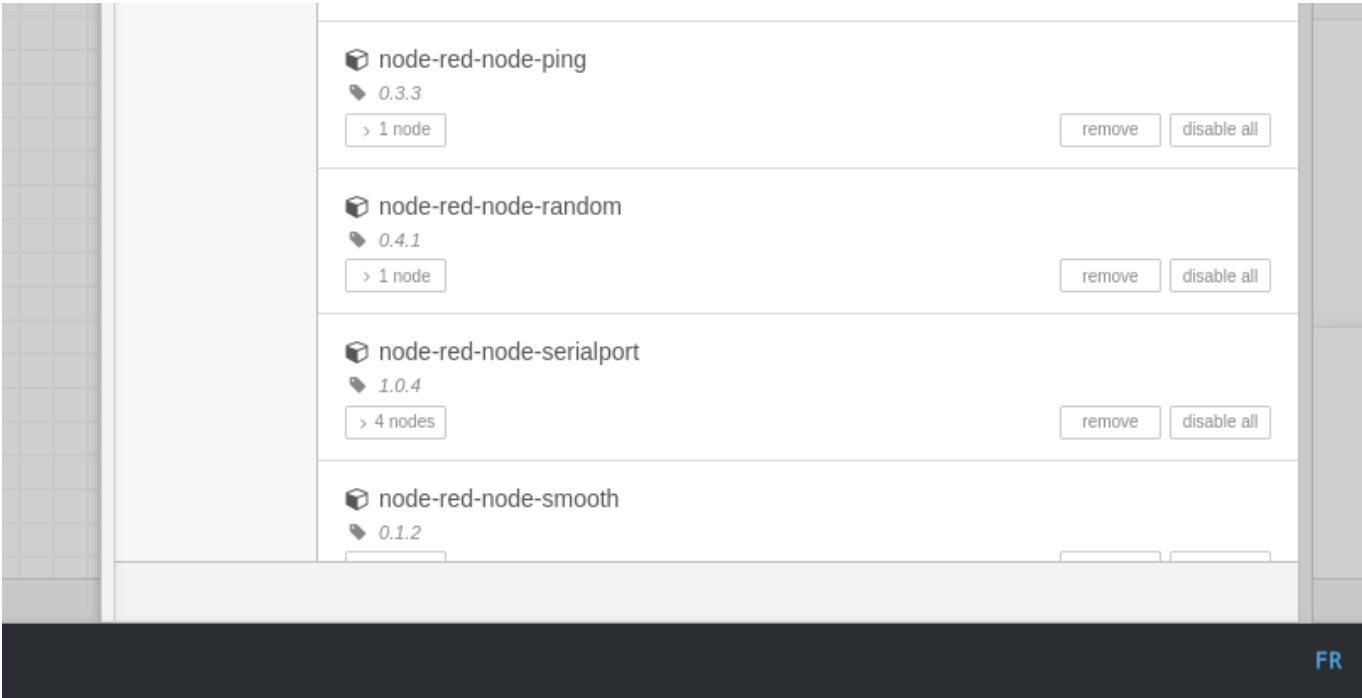
7.2. Configuration

7.2.1. Instalation des plugins

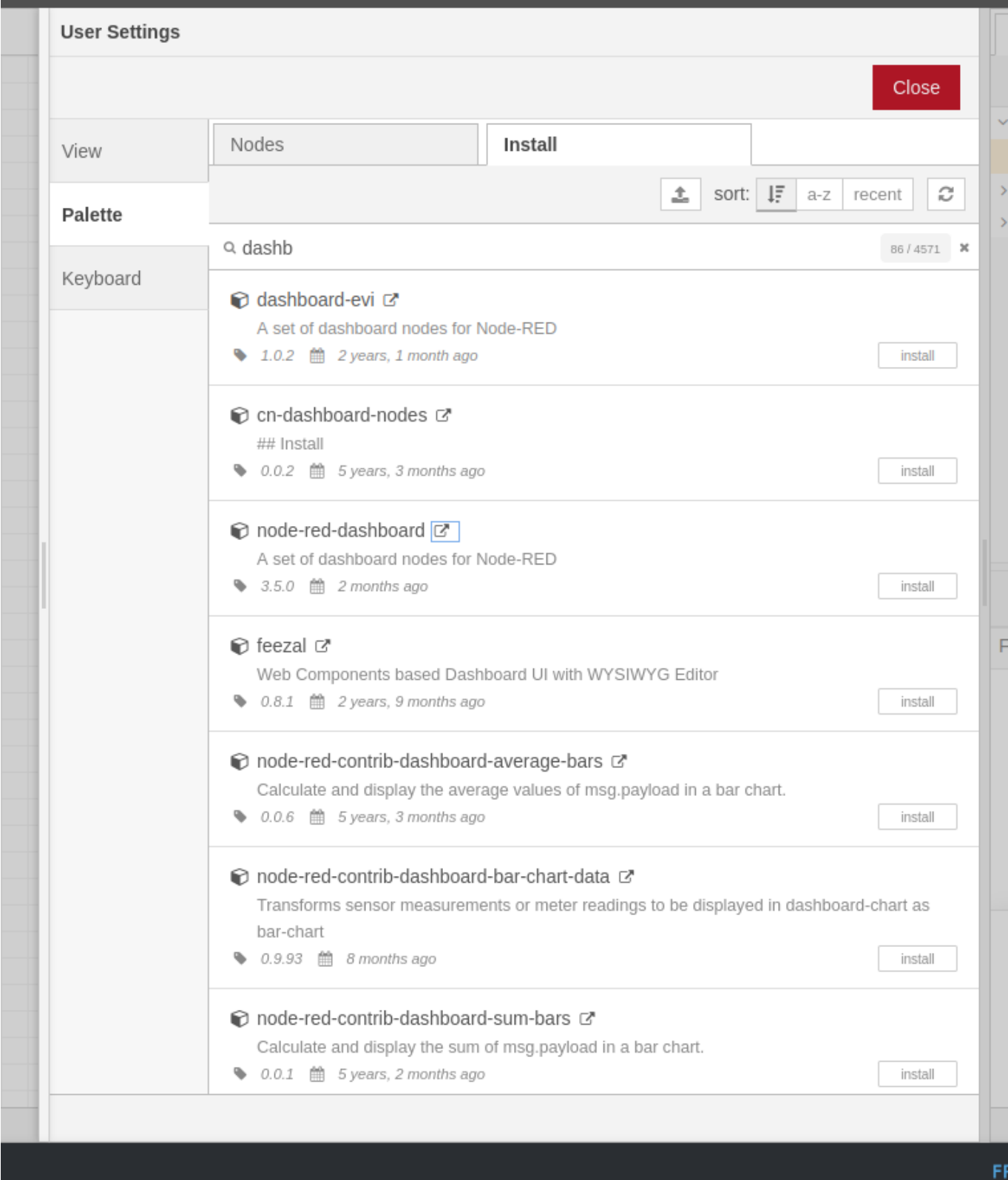


{width=33%}





{width=33%}



Installing 'node-red-dashboard'

Before installing, please read the node's documentation. Some nodes have dependencies that cannot be automatically resolved and can require a restart of Node-RED.

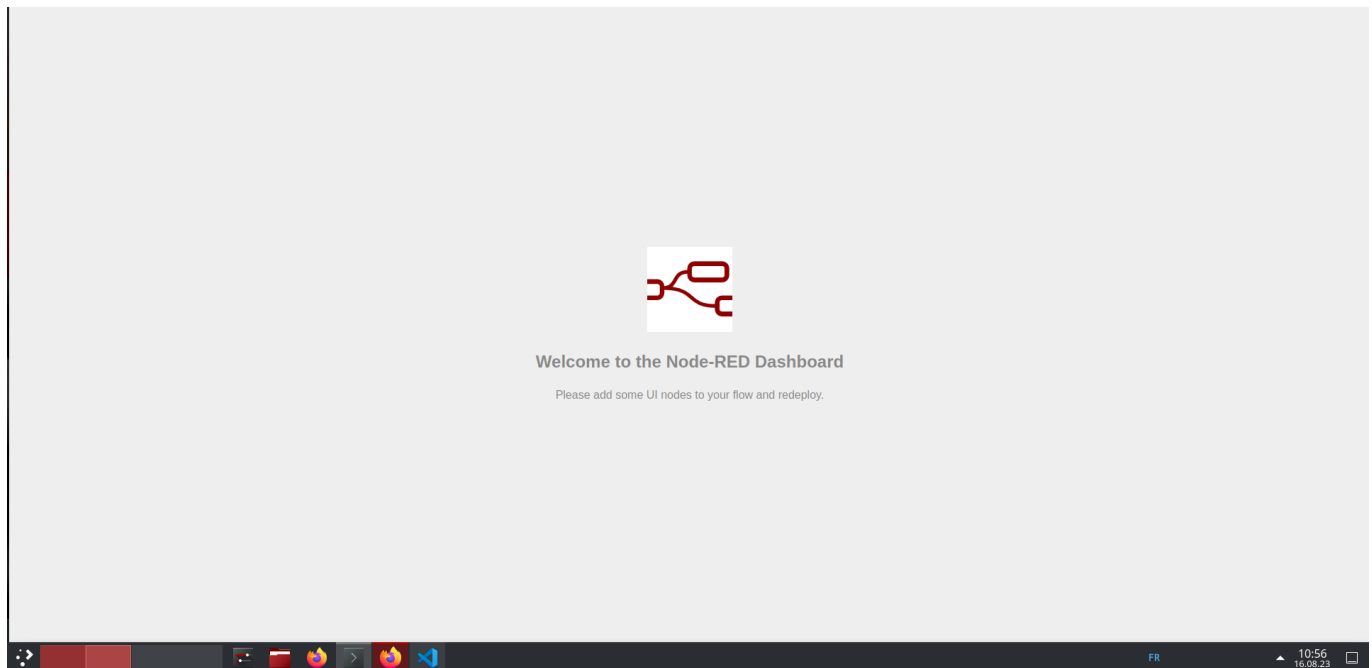
Cancel

Open node information

Install

{width=33%}

{width=100%}



{width=100%}

7.2.2. Sécurisation de Node-Red

La sécurisation de Node-Red se fait en modifiant le fichier settings.js, ou dans notre cas en utilisant la commande `node-red admin init` qui permet par exemple de créer le couple Utilisateur mot de passe.

Suite à cela il faudra également si cela s'avère utile ajouter un login au Dashboard.

7.2.3. Suivi Git

Pour suivre le projet sur git, il faut configurer un utilisateur avec la clé SSH puis faire un clone du projet.

User Settings

Close

View

Committer Details

Leave blank to use system default

Username

ToblerC

Email

cyril.tobler@rpn.ch

Git config

Keyboard

Workflow

Choose your preferred git workflow

☐ Manual

All changes must be manually committed under the 'history' sidebar


☒ Automatic


Changes are committed automatically with every deploy

SSH Keys

Allows you to create secure connections to remote git repositories.


add key

 Node-Red




{width=30%}


Projects



Open Project



Create Project



Clone Repository

Project name


Must contain only A-Z 0-9 _ -


Git repository URL

https://git.example.com/path/my-project.git

https://, ssh:// or file://

Credentials encryption key

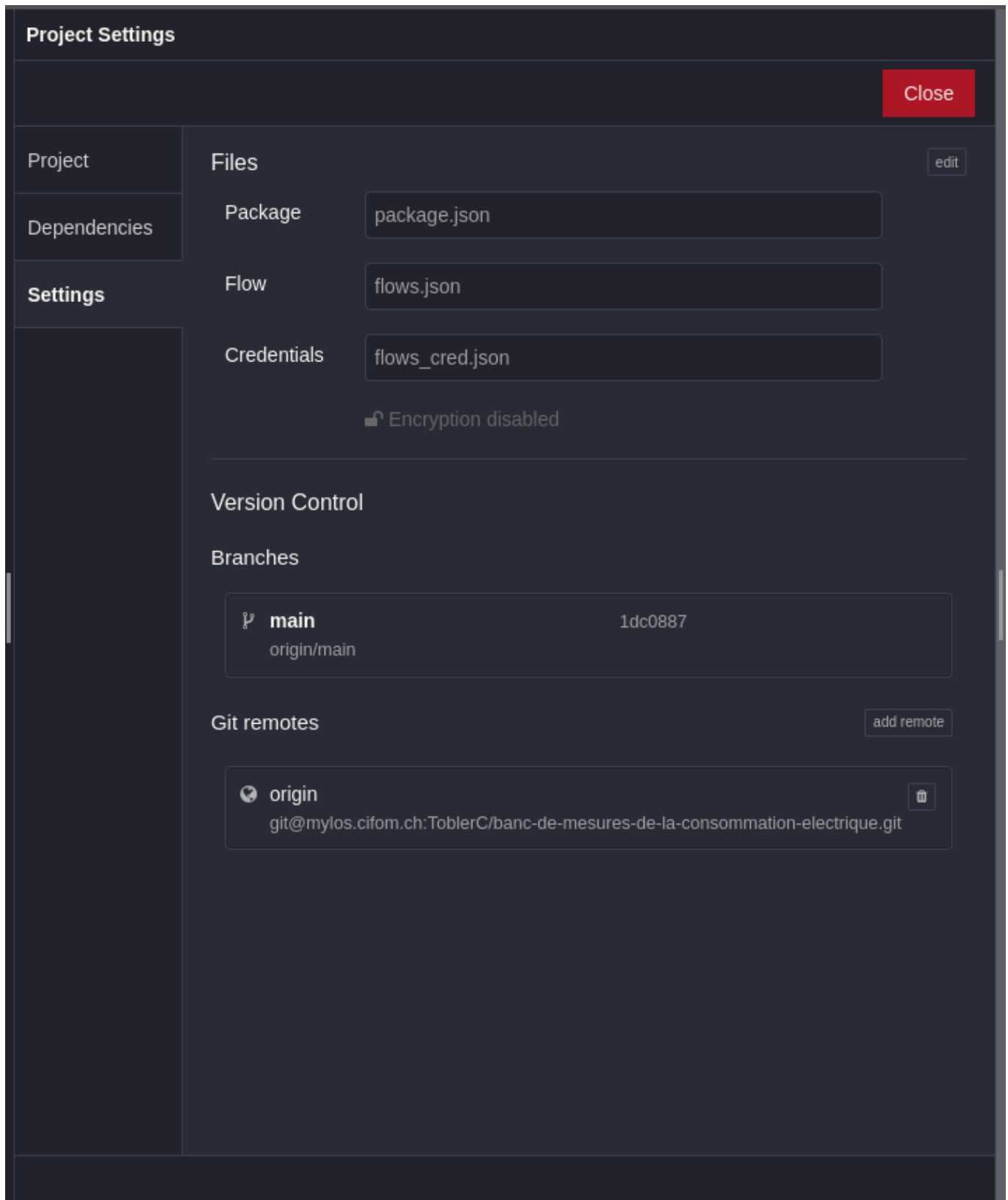




Cancel

Clone Repository

{width=30%}



{width=30%}

Comme c'est un clone il faut ajouter un fichier qui manque et modifier les droits d'accès.

```
tobby@Nidus:~/node-red/projects/banc-de-mesures-de-la-consommation-  
electrique $ touch ~/.node-red/projects/banc-de-mesures-de-la-consommation-  
electrique/flows_cred.json  
tobby@Nidus:~/node-red/projects/banc-de-mesures-de-la-consommation-  
electrique $ chmod 600 ~/.node-red/projects/banc-de-mesures-de-la-
```

```
consommation-electrique/flows_cred.json
```

8. Gatling

Gatling est un outil de test de charge open source basé sur Scala, conçu pour tester les performances des applications et des sites Web. Gatling simule des utilisateurs virtuels qui envoient des requêtes HTTP vers le système cible. Il enregistre les temps de réponse des requêtes et les présente sous forme de graphiques. Gatling est livré avec un éditeur de scénario basé sur un navigateur qui permet aux utilisateurs de créer des scénarios de test de charge en utilisant un langage de domaine spécifique (DSL) appelé Gatling DSL. Gatling DSL est un langage de programmation basé sur Scala qui permet aux utilisateurs de définir des scénarios de test de charge en utilisant des mots-clés tels que `exec`, `pause`, `feed`, etc.

La dernière version de Gatling est la version 3.9.5 qui est compatible avec Java 8 et Java 11. Dans ce projet, nous utiliserons Java 11 pour exécuter Gatling.

8.1. Instalation

8.1.1. Prerequis

```
tobby@Nidus:~ $ sudo apt install default-jdk
tobby@Nidus:~/.node-red $ java -version
openjdk version "11.0.18" 2023-01-17
OpenJDK Runtime Environment (build 11.0.18+10-post-Debian-1deb11u1)
OpenJDK 64-Bit Server VM (build 11.0.18+10-post-Debian-1deb11u1, mixed
mode)
tobby@Nidus:~/.node-red $
```

8.1.2. Download

```
tobby@Nidus:~ $ mkdir .gatling
tobby@Nidus:~ $ ls -la
total 104
drwxr-xr-x 18 toby toby 4096 16 août 15:10 .
drwxr-xr-x  3 root  root 4096 16 août 13:58 ..
-rw-----  1 toby toby  453 16 août 14:26 .bash_history
-rw-r--r--  1 toby toby   220  3 mai 04:53 .bash_logout
-rw-r--r--  1 toby toby 3523  3 mai 04:53 .bashrc
drwxr-xr-x  2 toby toby 4096  3 mai 05:02 Bookshelf
drwxr-xr-x  4 toby toby 4096 16 août 13:58 .cache
drwx-----  5 toby toby 4096 16 août 13:58 .config
drwxr-xr-x  2 toby toby 4096 16 août 13:58 Desktop
drwxr-xr-x  2 toby toby 4096 16 août 13:58 Documents
drwxr-xr-x  2 toby toby 4096 16 août 15:10 .gatling
```

```
drwxr-xr-x  2 toby toby 4096 16 août 13:58 Images
drwxr-xr-x  3 toby toby 4096  3 mai 05:02 .local
drwxr-xr-x  2 toby toby 4096 16 août 13:58 Modèles
drwxr-xr-x  2 toby toby 4096 16 août 13:58 Musique
drwxr-xr-x  4 toby toby 4096 16 août 15:05 .node-red
drwxr-xr-x  4 toby toby 4096 16 août 14:15 .npm
-rw-r--r--  1 toby toby  22 16 août 14:15 .npmrc
-rw-r--r--  1 toby toby  807  3 mai 04:53 .profile
drwxr-xr-x  2 toby toby 4096 16 août 13:58 Public
drwxr-xr-x  2 toby toby 4096 16 août 14:06 .ssh
drwxr-xr-x  2 toby toby 4096 16 août 13:58 Téléchargements
drwxr-xr-x  2 toby toby 4096 16 août 13:58 Vidéos
-rw-r--r--  1 toby toby  106 16 août 14:27 .Xauthority
-rw-r--r--  1 toby toby 2521 16 août 14:27 .xsession-errors
-rw-r--r--  1 toby toby 2521 16 août 14:02 .xsession-errors.old
tobby@Nidus:~ $ wget -O ~/.gatling/gatling-charts-highcharts-bundle-3.9.5-
bundle.zip https://repo1.maven.org/maven2/io/gatling/highcharts/gatling-
charts-highcharts-bundle/3.9.5/gatling-charts-highcharts-bundle-3.9.5-
bundle.zip
--2023-08-16 15:12:41--
https://repo1.maven.org/maven2/io/gatling/highcharts/gatling-charts-
highcharts-bundle/3.9.5/gatling-charts-highcharts-bundle-3.9.5-bundle.zip
Résolution de repo1.maven.org (repo1.maven.org)... 146.75.116.209,
2a04:4e42:8d::209
Connexion à repo1.maven.org (repo1.maven.org)|146.75.116.209|:443...
connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 77080673 (74M) [application/zip]
Sauvegarde en : « /home/tobby/.gatling/gatling-charts-highcharts-bundle-
3.9.5-bundle.zip »

/home/tobby/.gatling/gatling-charts-highcharts-bundl 100%
[=====
=====>] 73.51M 11.0MB/s ds 5.8s

2023-08-16 15:12:47 (12.8 MB/s) – « /home/tobby/.gatling/gatling-charts-
highcharts-bundle-3.9.5-bundle.zip » sauvegardé [77080673/77080673]

tobby@Nidus:~ $ unzip ~/.gatling/gatling-charts-highcharts-bundle-3.9.5-
bundle.zip -d ~/.gatling/
Archive: /home/tobby/.gatling/gatling-charts-highcharts-bundle-3.9.5-
bundle.zip

[...]

inflating: /home/tobby/.gatling/gatling-charts-highcharts-bundle-
3.9.5/LICENSE
tobby@Nidus:~ $ cd .gatling/
tobby@Nidus:~/.gatling $ ls -la
total 75288
drwxr-xr-x  3 toby toby      4096 16 août 15:12 .
drwxr-xr-x 18 toby toby      4096 16 août 15:10 ..
drwxr-xr-x  7 toby toby      4096 10 mai 11:19 gatling-charts-highcharts-
bundle-3.9.5
```



```
-rw-r--r-- 1 toby toby 77080673 10 mai 11:19 gatling-charts-highcharts-
bundle-3.9.5-bundle.zip
tobby@Nidus:~/.gatling $ cd gatling-charts-highcharts-bundle-3.9.5/
tobby@Nidus:~/.gatling/gatling-charts-highcharts-bundle-3.9.5 $ ls -la
total 48
drwxr-xr-x 7 toby toby 4096 10 mai 11:19 .
drwxr-xr-x 3 toby toby 4096 16 août 15:12 ..
drwxr-xr-x 2 toby toby 4096 10 mai 11:19 bin
drwxr-xr-x 2 toby toby 4096 10 mai 11:19 conf
drwxr-xr-x 2 toby toby 12288 10 mai 11:19 lib
-rw-r--r-- 1 toby toby 11367 10 mai 11:19 LICENSE
drwxr-xr-x 2 toby toby 4096 10 mai 11:19 results
drwxr-xr-x 5 toby toby 4096 10 mai 11:19 user-files
```

8.2. Vérification de l'installation

```
tobby@Nidus:~/.gatling/gatling-charts-highcharts-bundle-3.9.5/bin $
./gatling.sh
GATLING_HOME is set to /home/tobby/.gatling/gatling-charts-highcharts-
bundle-3.9.5
Do you want to run the simulation locally, on Gatling Enterprise, or just
package it?
Type the number corresponding to your choice and press enter
[0] <Quit>
[1] Run the Simulation locally
[2] Package and upload the Simulation to Gatling Enterprise Cloud, and run
it there
[3] Package the Simulation for Gatling Enterprise
[4] Show help and exit
1
août 16, 2023 4:28:28 PM java.util.prefs.FileSystemPreferences$1 run
INFO: Created user preferences directory.
computerdatabase.ComputerDatabaseSimulation is the only simulation,
executing it.
Select run description (optional)
InstallVerif
Simulation computerdatabase.ComputerDatabaseSimulation started...

=====
=====
2023-08-16 16:29:14                                     5s elapsed
---- Requests -----
-----
> Global (OK=23 KO=0
)
> Home (OK=6 KO=0
)
> Home Redirect 1 (OK=6 KO=0
)
> Search (OK=5 KO=0
)
)
```

```
> Select (OK=3 K0=0
)
> Page 0 (OK=3 K0=0
)

---- Users -----
-----
[-----
] 0%
      waiting: 5      / active: 5      / done: 0
---- Admins -----
-----
[-----
] 0%
      waiting: 1      / active: 1      / done: 0
=====
=====

=====
=====
2023-08-16 16:29:19 10s elapsed
---- Requests -----
-----
> Global (OK=71 K0=0
)
> Home (OK=12 K0=0
)
> Home Redirect 1 (OK=12 K0=0
)
> Search (OK=11 K0=0
)
> Select (OK=10 K0=0
)
> Page 0 (OK=9 K0=0
)
> Page 1 (OK=8 K0=0
)
> Page 2 (OK=5 K0=0
)
> Page 3 (OK=3 K0=0
)
> Form (OK=1 K0=0
)

---- Users -----
-----
[#####-----
-] 20%
      waiting: 0      / active: 8      / done: 2
---- Admins -----
-----
[-----
-] 0%
```

```

waiting: 0      / active: 2      / done: 0

=====
=====

=====
=====
2023-08-16 16:29:24                                15s elapsed
---- Requests -----
-----
> Global                                           (OK=101    K0=2
)
> Home                                           (OK=12     K0=0
)
> Home Redirect 1                               (OK=12     K0=0
)
> Search                                           (OK=12     K0=0
)
> Select                                           (OK=12     K0=0
)
> Page 0                                           (OK=12     K0=0
)
> Page 1                                           (OK=12     K0=0
)
> Page 2                                           (OK=11     K0=0
)
> Page 3                                           (OK=10     K0=0
)
> Form                                           (OK=4      K0=0
)
> Post                                           (OK=3      K0=0
)
> Post Redirect 1                               (OK=1      K0=2
)
---- Errors -----
-----
> status.find.is(201), but actually found 200                2
(100,0%)

---- Users -----
-----
[#####-----
-] 70%
      waiting: 0      / active: 3      / done: 7
---- Admins -----
-----
[#####-----
-] 50%
      waiting: 0      / active: 1      / done: 1

=====
=====

=====
```

```
=====
2023-08-16 16:29:26                                     17s elapsed
---- Requests -----
----
> Global (OK=105 K0=3
)
> Home (OK=12 K0=0
)
> Home Redirect 1 (OK=12 K0=0
)
> Search (OK=12 K0=0
)
> Select (OK=12 K0=0
)
> Page 0 (OK=12 K0=0
)
> Page 1 (OK=12 K0=0
)
> Page 2 (OK=12 K0=0
)
> Page 3 (OK=12 K0=0
)
> Form (OK=4 K0=0
)
> Post (OK=4 K0=0
)
> Post Redirect 1 (OK=1 K0=3
)
---- Errors -----
----
> status.find.is(201), but actually found 200 3
(100,0%)

---- Users -----
[#####]100%
      waiting: 0 / active: 0 / done: 10
---- Admins -----
[#####]100%
      waiting: 0 / active: 0 / done: 2
=====

Simulation computerdatabase.ComputerDatabaseSimulation completed in 17
seconds
Parsing log file(s)...
Parsing log file(s) done
Generating reports...

=====
```

```

---- Global Information -----
-----
> request count                                108 (OK=105    K0=3
)
> min response time                            108 (OK=108
K0=111    )
> max response time                            1563 (OK=1563
K0=114    )
> mean response time                           162 (OK=163
K0=112    )
> std deviation                                168 (OK=170    K0=1
)
> response time 50th percentile                 115 (OK=115
K0=112    )
> response time 75th percentile                 120 (OK=121
K0=113    )
> response time 95th percentile                 351 (OK=352
K0=114    )
> response time 99th percentile                 620 (OK=620
K0=114    )
> mean requests/sec                            6.353 (OK=6.176
K0=0.176  )
---- Response Time Distribution -----
-----
> t < 800 ms                                  104 ( 96%)
> 800 ms <= t < 1200 ms                       0 (  0%)
> t >= 1200 ms                                 1 (  1%)
> failed                                       3 (  3%)
---- Errors -----
-----
> status.find.is(201), but actually found 200                                     3
(100,0%)
=====
=====

Reports generated in 0s.
Please open the following file: file:///home/tobby/.gatling/gatling-charts-
highcharts-bundle-3.9.5/results/computerdatabasesimulation-
20230816142907884/index.html

```

8.3. Configuration

9. Apache et Site Web

9.1. Installation

```

sudo apt install apache2
sudo systemctl status apache2
sudo systemctl enable apache2

```

9.2. Mise en place d'un site Web

J'ai crée un site web très simple reprenant le readme du projet. Et il comporte trois pages ainsi que du CSS.

```
scp -r /home/toblerc/Documents/ES_2024/banc-de-mesures-de-la-consommation-electrique/siteWeb/www/html toby@VOLT:/var/www/html/
```

10. MQTT

Dans notre cas je souhaite utiliser le MQTT pour envoyer les données de consommation à Node-Red. En passant outre le transfert de requête via le SSH et l'utilisation de clé SSH, le MQTT permet de gagner en performance et en sécurité. Niveau performance, le MQTT est plus léger que le SSH de l'ordre de 10 fois plus léger.

10.1. Instalation de Mosquitto sur Nidus

```
tobby@Nidus:~/.ssh $ sudo apt install mosquitto
Lecture des listes de paquets... Fait
[...]
tobby@Nidus:~/.ssh $ sudo systemctl status mosquitto
• mosquitto.service - Mosquitto MQTT Broker
   Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-08-22 16:01:58 CEST; 7s ago
     Docs: man:mosquitto.conf(5)
           man:mosquitto(8)
   Process: 22571 ExecStartPre=/bin/mkdir -m 740 -p /var/log/mosquitto (code=exited, status=0/SUCCESS)
   Process: 22572 ExecStartPre=/bin/chown mosquitto /var/log/mosquitto (code=exited, status=0/SUCCESS)
   Process: 22573 ExecStartPre=/bin/mkdir -m 740 -p /run/mosquitto (code=exited, status=0/SUCCESS)
   Process: 22574 ExecStartPre=/bin/chown mosquitto /run/mosquitto (code=exited, status=0/SUCCESS)
   Main PID: 22575 (mosquitto)
     Tasks: 1 (limit: 3933)
        CPU: 42ms
   CGroup: /system.slice/mosquitto.service
           └─22575 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf

aoû 22 16:01:58 Nidus systemd[1]: Starting Mosquitto MQTT Broker...
aoû 22 16:01:58 Nidus systemd[1]: Started Mosquitto MQTT Broker.
```

10.2. Ouverture des port sur Nidus

Modifier le fichier de conf comme suit :

```
tobby@Nidus:~ $ sudo vim /etc/mosquitto/mosquitto.conf
tobby@Nidus:~ $ sudo cat /etc/mosquitto/mosquitto.conf
# Place your local configuration in /etc/mosquitto/conf.d/
#
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

pid_file /run/mosquitto/mosquitto.pid

persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

include_dir /etc/mosquitto/conf.d

listener 1883
allow_anonymous true
```

10.3. Script MQTT

J'ai créé un script MQTT qui permet de publier les données de consommation sur le broker MQTT. Le script est lancé au démarrage de la machine et tourne en boucle. Et vérifie si les dépendances sont installées, si ce n'est pas le cas il les installe. Et finalement il vérifie si le lien symbolique vers init.d existe, si ce n'est pas le cas il le crée.

```
#!/bin/bash
#### BEGIN INIT INFO
# Provides:          mqtt
# Required-Start:    $remote_fs $syslog
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Script MQTT de collecte de données
# Description:       Ce script collecte la charge CPU, la charge RAM
#                   et le nombre de processus, puis publie ces données
#                   sur un broker MQTT.
#### END INIT INFO

# Pour ajouter les droits d'exécution :
# chmod +x mqtt.sh
# Pour le copier depuis Nidus vers Volt :
# scp ./mqtt.sh toby@volt:/usr/local/bin/mqtt.sh
# Emplacement du script (doit être dans /usr/local/bin)
INSTALL_DIR="/usr/local/bin"
# Nom du script
SCRIPT_NAME="mqtt.sh"
# Adresse du broker MQTT
MQTT_BROKER="nidus"
# Sujets MQTT pour les différentes données
```



```
MQTT_TOPIC_CPU="benchmark/cpu"
MQTT_TOPIC_RAM="benchmark/ram"
MQTT_TOPIC_PROCESSES="benchmark/processes"

# Vérification si le script est dans le bon dossier d'installation
if [ "$(dirname "$(readlink -f "$0")")" != "$INSTALL_DIR" ]; then
    echo "Erreur : Le script doit être installé dans $INSTALL_DIR"
    exit 1
fi

# Vérification et installation des dépendances (mosquitto-clients)
if ! command -v mosquitto_pub &> /dev/null; then
    echo "Installation de mosquitto-clients..."
    sudo apt-get update
    sudo apt-get install mosquitto-clients
    echo "Installation terminée."
fi

# Vérification si le lien symbolique vers init.d existe
if [ ! -e "/etc/init.d/$SCRIPT_NAME" ]; then
    echo "Création du lien symbolique dans /etc/init.d..."
    sudo ln -s "$INSTALL_DIR/$SCRIPT_NAME" "/etc/init.d/$SCRIPT_NAME"
    echo "Lien symbolique créé."
fi

# Vérification et activation du service init.d
if ! sudo service "$SCRIPT_NAME" status &> /dev/null; then
    echo "Activation du service..."
    sudo update-rc.d "$SCRIPT_NAME" defaults
    echo "Service activé."
fi

# Boucle principale pour la collecte et la publication des données
while true; do
    # Collecte des données
    CPU_LOAD=$(top -bn1 | grep "Cpu(s)" | awk '{print $2 + $4}')
    RAM_LOAD=$(free | awk '/Mem/{printf("%.2f\n", $3/$2*100)}')
    PROCESS_COUNT=$(ps aux | wc -l)

    # Publication des données sur MQTT
    mosquitto_pub -h $MQTT_BROKER -t $MQTT_TOPIC_CPU -m "$CPU_LOAD"
    mosquitto_pub -h $MQTT_BROKER -t $MQTT_TOPIC_RAM -m "$RAM_LOAD"
    mosquitto_pub -h $MQTT_BROKER -t $MQTT_TOPIC_PROCESSES -m
"$PROCESS_COUNT"

    echo "Données publiées sur MQTT"

    sleep 1 # Attente d'une seconde
done
```

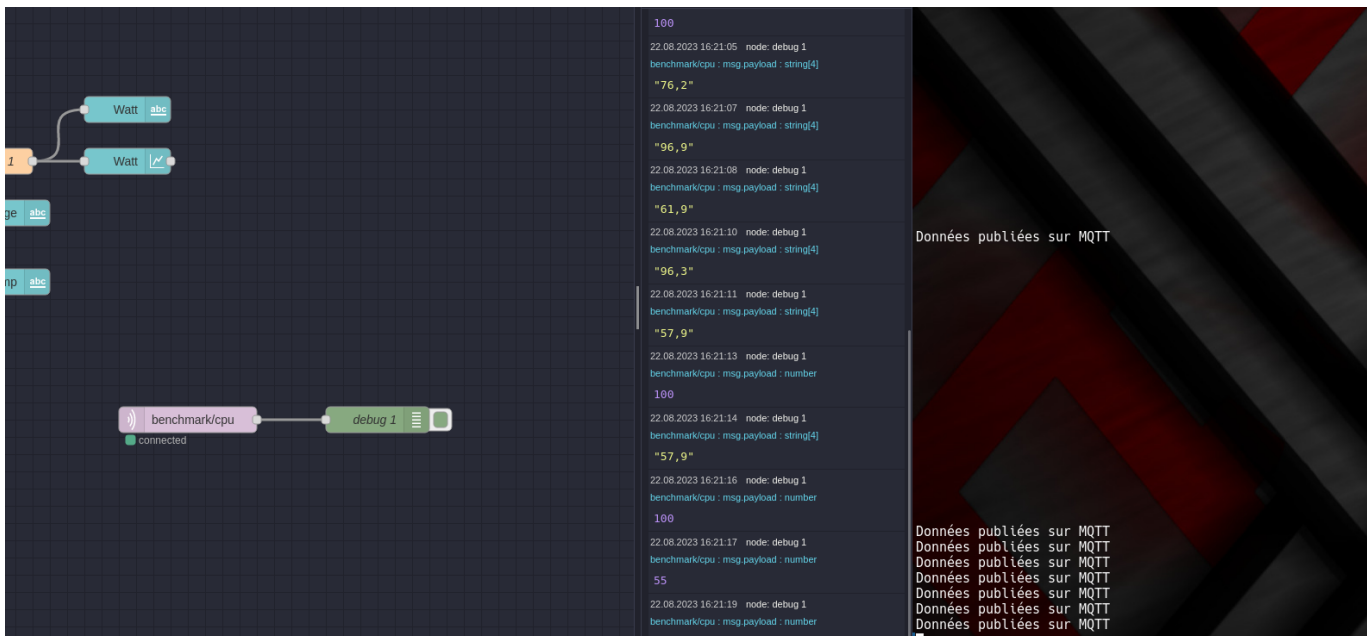
10.4. Installation

```
toblerc@LPT-UNIX-USB-CT:~/Documents/ES_2024/banc-de-mesures-de-la-
consommation-electrique$ scp ./mqtt.sh toby@volt:/usr/local/bin/mqtt.sh
mqtt.sh
100% 2526      2.1MB/s   00:00
```

10.5. Utilisation du script

```
tobby@Volt:/usr/local/bin$ sudo ./mqtt.sh
Installation de mosquitto-clients...
[...]
Il est nécessaire de prendre 136 ko dans les archives.
Après cette opération, 568 ko d'espace disque supplémentaires seront
utilisés.
Souhaitez-vous continuer ? [0/n] 0
[...]
Installation terminée.
Création du lien symbolique dans /etc/init.d...
Lien symbolique créé.
Activation du service...
Service activé.
```

10.5.1. Vérification



{width=100%}

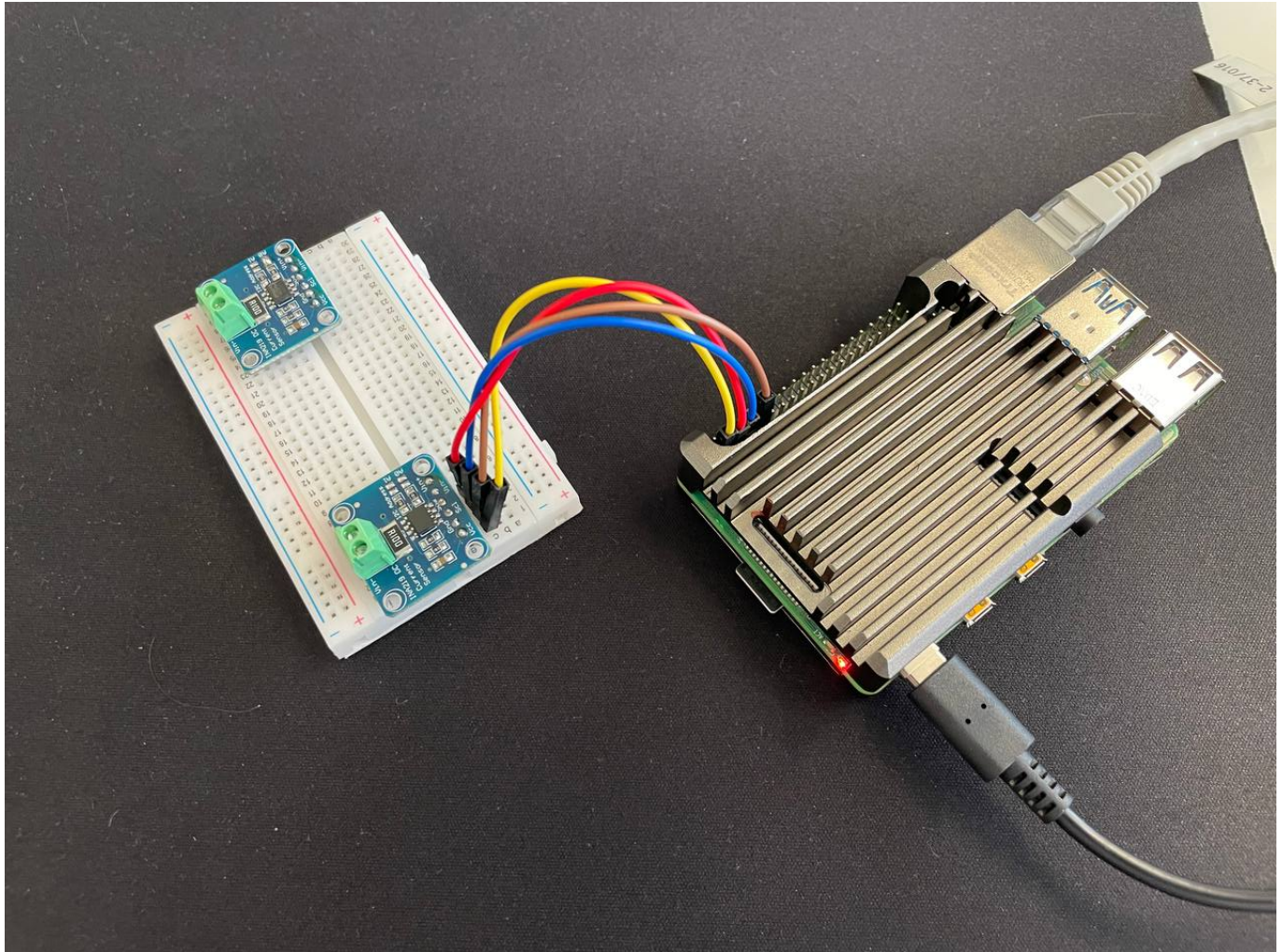
11. INA219

Dans notre cas, il y a deux puce INA219, une en remplacement en cas de problème et l'autre pour la mesure de la consommation. Pour les différencier, j'ai soudé l'adresse I2C de la puce de mesure sur 0x40 et celle de

la puce de remplacement sur 0x41. Sur le photos, la puce branchée en E1 à E6 est celle de mesure et celle branchée en E25 à E30 est celle de remplacement.

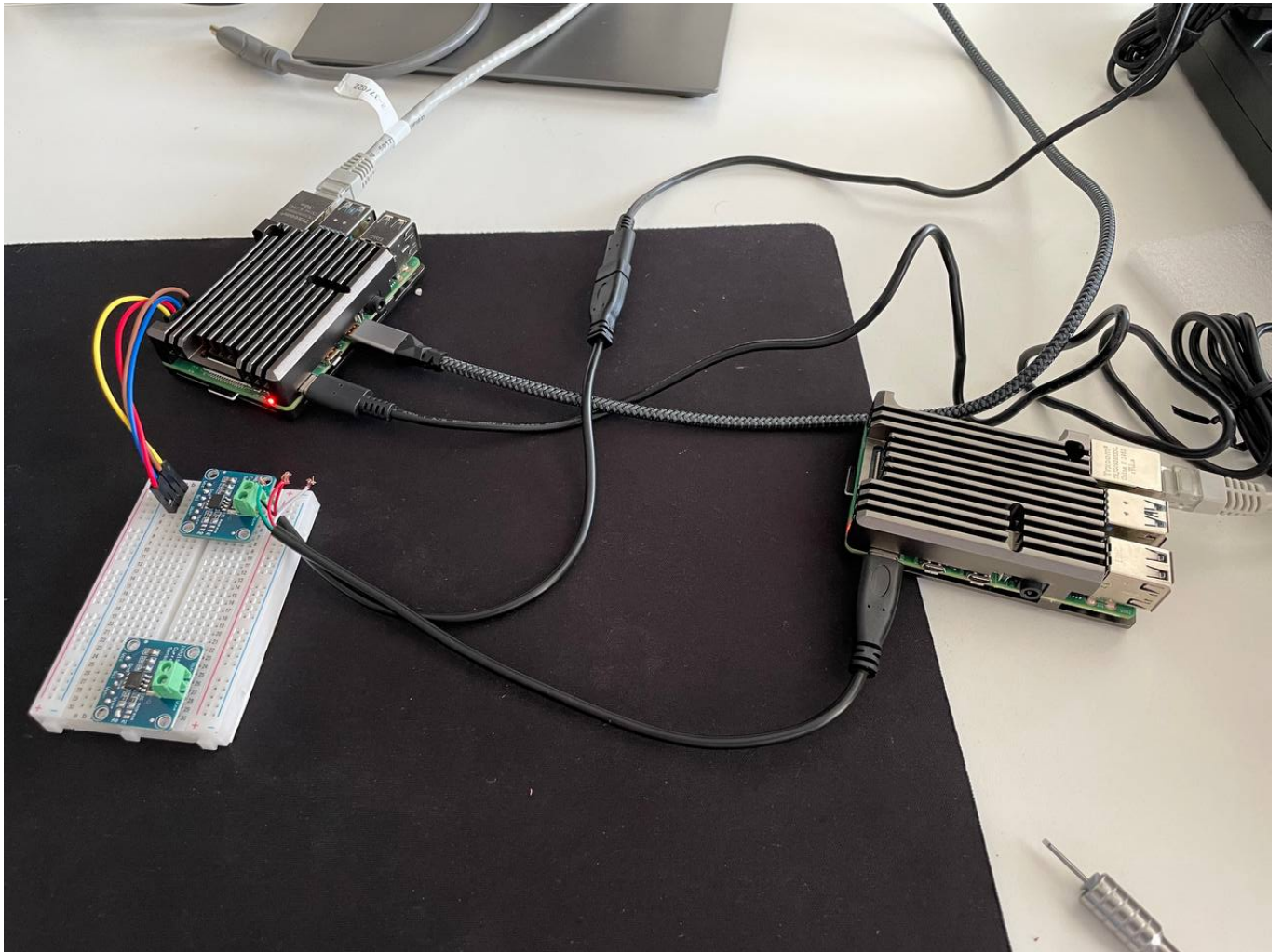
11.1. Instalation physique

11.1.1. Branchement SANS VOLT

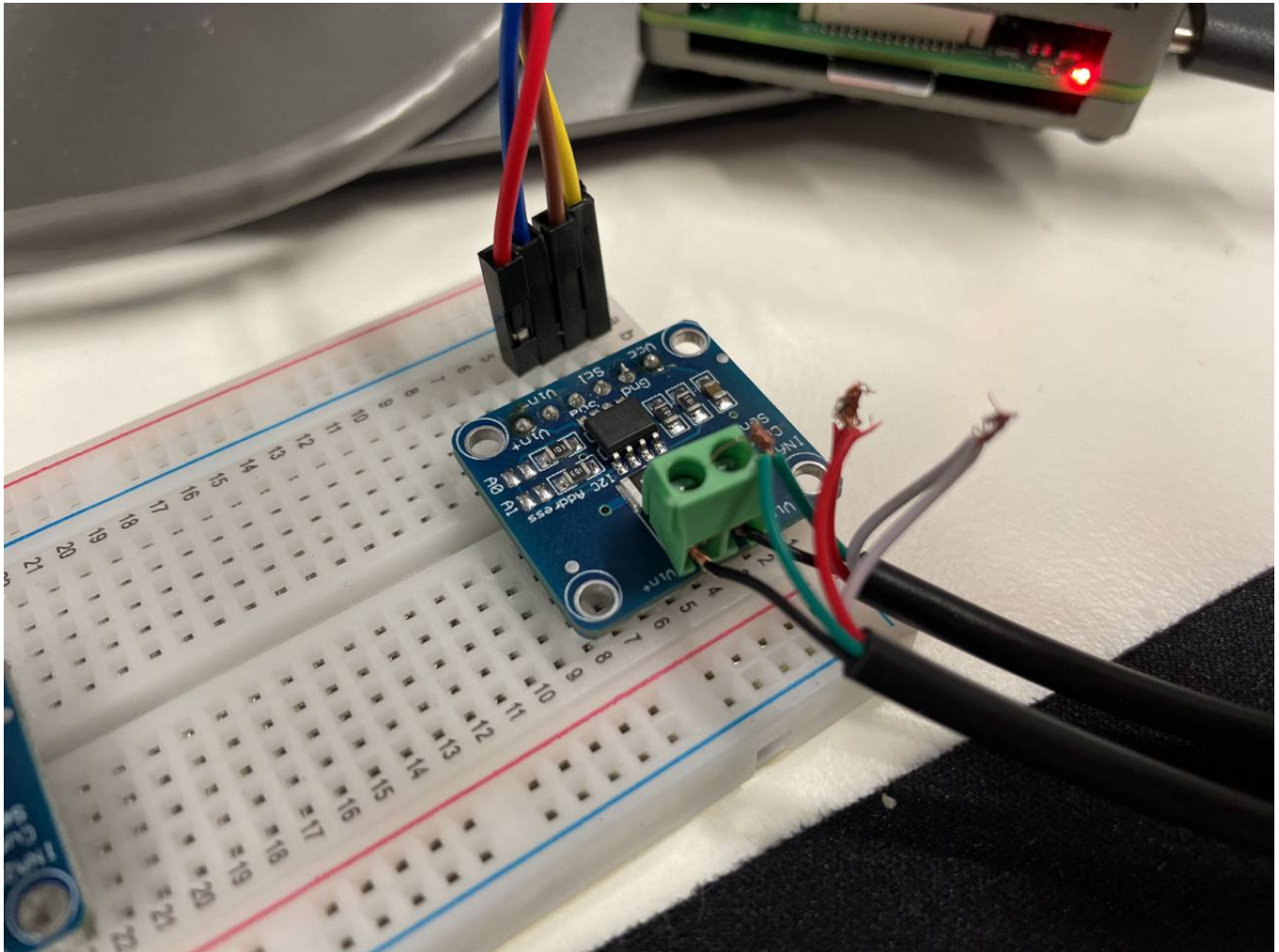


{width=100%}

11.1.2. Branchement AVEC VOLT



```
{width=100%}
```



{width=100%}

11.2. Vérification de la présence du INA219

```
tobby@Nidus:~ $ sudo i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                                     -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- --
40: 40 -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- --
```

```
tobby@Nidus:~ $ sudo i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                                     -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- 41 -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- --
```

11.3. Obtention des données

11.3.1. Test avec le script python A vide

Installation de la bibliothèque python

```
tobby@Nidus:~ $ sudo pip3 install pi-ina219
Looking in indexes: https://pypi.org/simple,
https://www.piwheels.org/simple
Collecting pi-ina219
  Downloading pi_ina219-1.4.1-py2.py3-none-any.whl (10 kB)
Collecting Adafruit-GPIO
  Downloading https://www.piwheels.org/simple/adafruit-gpio/Adafruit_GPIO-1.0.3-py3-none-any.whl (38 kB)
Collecting mock
  Downloading https://www.piwheels.org/simple/mock/mock-5.1.0-py3-none-any.whl (30 kB)
Collecting adafruit-pureio
  Downloading https://www.piwheels.org/simple/adafruit-pureio/Adafruit_PureIO-1.1.11-py3-none-any.whl (10 kB)
Requirement already satisfied: spidev in /usr/lib/python3/dist-packages (from Adafruit-GPIO->pi-ina219) (3.5)
Installing collected packages: adafruit-pureio, mock, Adafruit-GPIO, pi-ina219
Successfully installed Adafruit-GPIO-1.0.3 adafruit-pureio-1.1.11 mock-5.1.0 pi-ina219-1.4.1
```

Vérification de la présence de l'INA219

```
tobby@Nidus:~ $ i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40: 40  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

Création du script python

```
tobby@Nidus:~/Documents $ mkdir py
tobby@Nidus:~/Documents $ cd py
tobby@Nidus:~/Documents/py $ touch my_ina219.py
tobby@Nidus:~/Documents/py $ ls -la
```

```
total 8
drwxr-xr-x 2 toby toby 4096 22 août 10:19 .
drwxr-xr-x 3 toby toby 4096 22 août 10:18 ..
-rw-r--r-- 1 toby toby    0 22 août 10:19 my_ina219.py
tobby@Nidus:~/Documents/py $ sudo vi ./my_ina219.py
```

```
#!/usr/bin/env python
from ina219 import INA219
from ina219 import DeviceRangeError

SHUNT_OHMS = 0.1

def read():
    ina = INA219(SHUNT_OHMS)
    ina.configure()

    print("Bus Voltage: %.3f V" % ina.voltage())
    try:
        print("Bus Current: %.3f mA" % ina.current())
        print("Power: %.3f mW" % ina.power())
        print("Shunt voltage: %.3f mV" % ina.shunt_voltage())
    except DeviceRangeError as e:
        # Current out of device range with specified shunt resistor
        print(e)

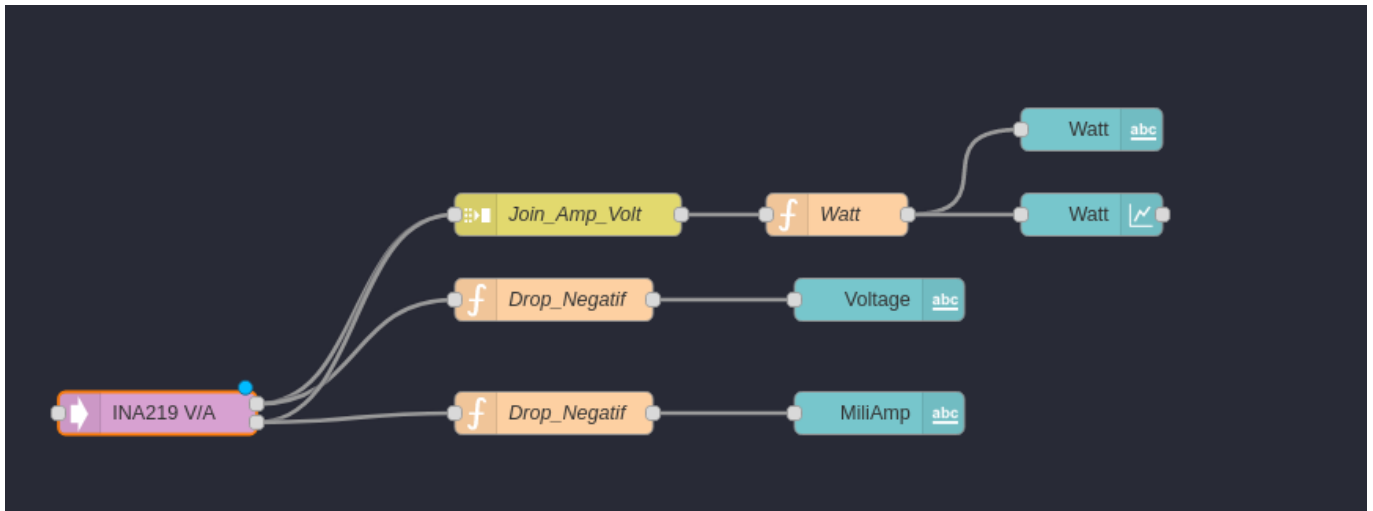
if __name__ == "__main__":
    read()
```

Execution du script

```
tobby@Nidus:~/Documents/py $ python ./my_ina219.py
Bus Voltage: 0.888 V
Bus Current: -0.195 mA
Power: 0.000 mW
Shunt voltage: -0.010 mV
```

12. Noeud Node-Red

12.1. INA219



L'INA219 possède deux sorties qui fournissent des valeurs en milliampères et en volts. J'ai configuré des nœuds de fonctions pour exclure les valeurs négatives, qui sont des erreurs de lecture, et pour ensuite traiter ces valeurs en les affichant dans un libellé.

À côté, un nœud "join" combine les deux valeurs en un seul message, qui est ensuite envoyé à un nœud de fonction pour calculer les watts et les afficher à la fois dans un label et dans un graphique.

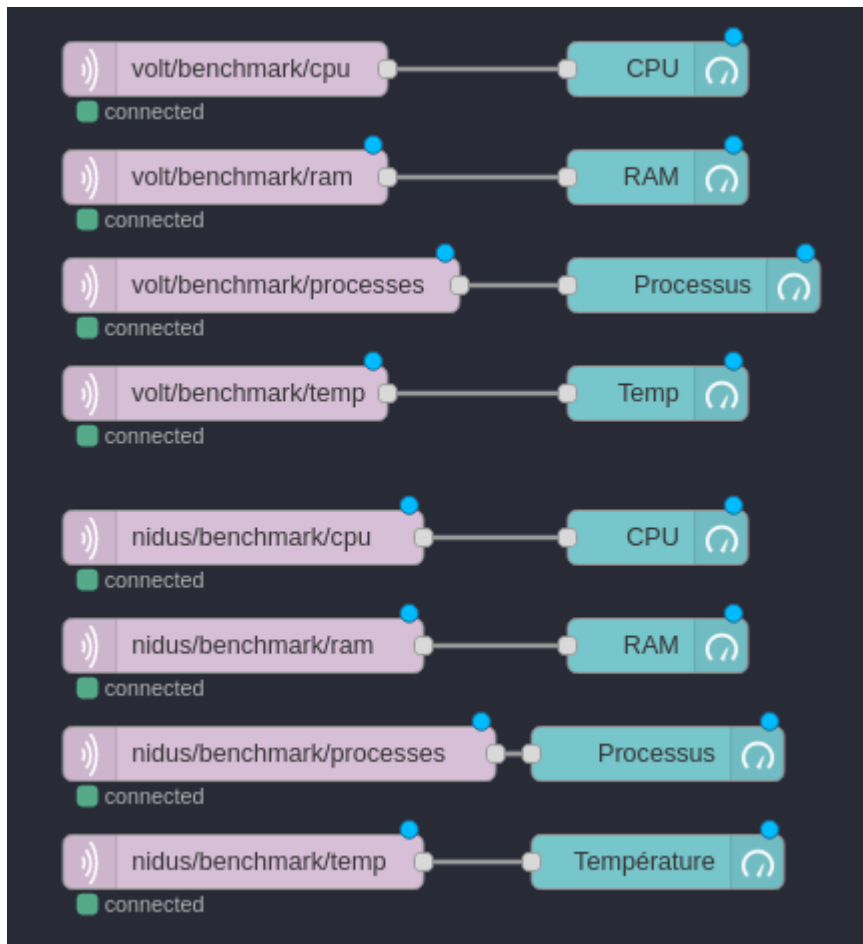
```
// Récupérer les valeurs de courant (mA) et de tension (V) depuis les
propriétés msg.payload
var current_mA = msg.payload.miliamps;
var voltage_V = msg.payload.voltage;

// Calculer la puissance en watts (W)
var power_W = (current_mA / 1000) * voltage_V; // Convertir le courant en
ampères

// Vérifier si la tension est négative
if (voltage_V < 0.5) {
    // Si la tension est négative, ne rien faire et retourner le message
    inchangé
    return null;
}

// Créer un nouvel objet msg avec la puissance en watts comme payload
msg.payload = power_W;
msg.topic = "Watt";
// Renvoyer le message modifié
return msg;
```

12.2. Monitoring



Pour le monitoring, j'ai utilisé le MQTT pour échanger les données plus rapidement et plus légèrement que le SSH. Depuis le nœud de réception MQTT, les données sont envoyées dans un nœud de gauge qui affiche la valeur en temps réel.

12.3. Dashboard

Le dashboard est le regroupement de tous les nœuds qui vont permettre de visualiser les données. Les nœuds qui permettent de visualiser les données sont les suivants :

- bouton
- dropdown
- switch
- slider
- numeric
- text input
- date picker
- colour picker
- form## 12.3. Dashboard Le dashboard, en tant que point central de regroupement, rassemble tous les nœuds nécessaires pour une visualisation efficace des données générées.

Il met à disposition une multitude de nœuds spécifiques qui permettent de créer une interface utilisateur intuitive et interactive. Ces nœuds offrent des fonctionnalités variées pour présenter, ajuster et transmettre les données. Voici quelques-uns des nœuds qui facilitent cette expérience :

- **Bouton (Button)** : Permet aux utilisateurs d'interagir en déclenchant des actions spécifiques d'une manière simple et directe.

- **Liste déroulante (*Dropdown*)** : Offre un moyen de sélectionner une option parmi plusieurs, ce qui permet de contrôler les paramètres ou les valeurs de manière structurée.
- **Interrupteur (*Switch*)** : Fournit un moyen clair et immédiat de basculer entre deux états, souvent utilisé pour activer ou désactiver des fonctionnalités.
- **Curseur (*Slider*)** : Permet un réglage précis d'une valeur numérique en glissant un curseur. Utile pour ajuster des paramètres continus.
- **Champ numérique (*Numeric*)** : Offre une interface pour entrer des valeurs numériques avec précision.
- **Champ de texte (*Text input*)** : Permet aux utilisateurs d'entrer du texte, généralement utilisé pour des commentaires, des descriptions ou des valeurs personnalisées.
- **Sélecteur de date (*Date picker*)** : Facilite la sélection de dates et d'heures, souvent utilisé pour des enregistrements horodatés.
- **Sélecteur de couleur (*Colour picker*)** : Permet de choisir une couleur avec précision pour des éléments visuels ou des codes couleur.
- **Formulaire (*Form*)** : Permet de regrouper plusieurs champs de saisie et de contrôle en une seule entité logique, simplifiant ainsi la collecte de données.
- **Texte (*Text*)** : Affiche du texte ou des instructions pour guider l'utilisateur dans l'interprétation des données ou l'utilisation de l'interface.
- **Jauge (*Gauge*)** : Affiche une valeur numérique sous forme graphique, permettant de visualiser rapidement un état ou une mesure.
- **Graphique (*Chart*)** : Permet de créer divers types de graphiques pour afficher visuellement les tendances et les relations entre les données.
- **Sortie audio (*Audio out*)** : Peut être utilisé pour fournir des commentaires auditifs ou des alertes sonores.
- **Notification (*Notification*)** : Permet d'afficher des messages d'information ou d'alerte à l'utilisateur pour des événements spécifiques.
- **Contrôle d'interface utilisateur (*UI control*)** : Offre des éléments interactifs personnalisables pour répondre aux besoins spécifiques de l'application.
- **Modèle (*Template*)** : Fournit une flexibilité avancée en permettant d'intégrer du contenu HTML personnalisé, ce qui peut inclure des graphiques, des widgets et bien plus encore.

Ces nœuds rassemblent un ensemble puissant d'outils pour créer des interfaces visuelles riches, ce qui élimine la nécessité de coder manuellement chaque élément. Cela permet aux développeurs et aux utilisateurs non techniques de collaborer efficacement dans la création d'interfaces utilisateur conviviales et informatives.

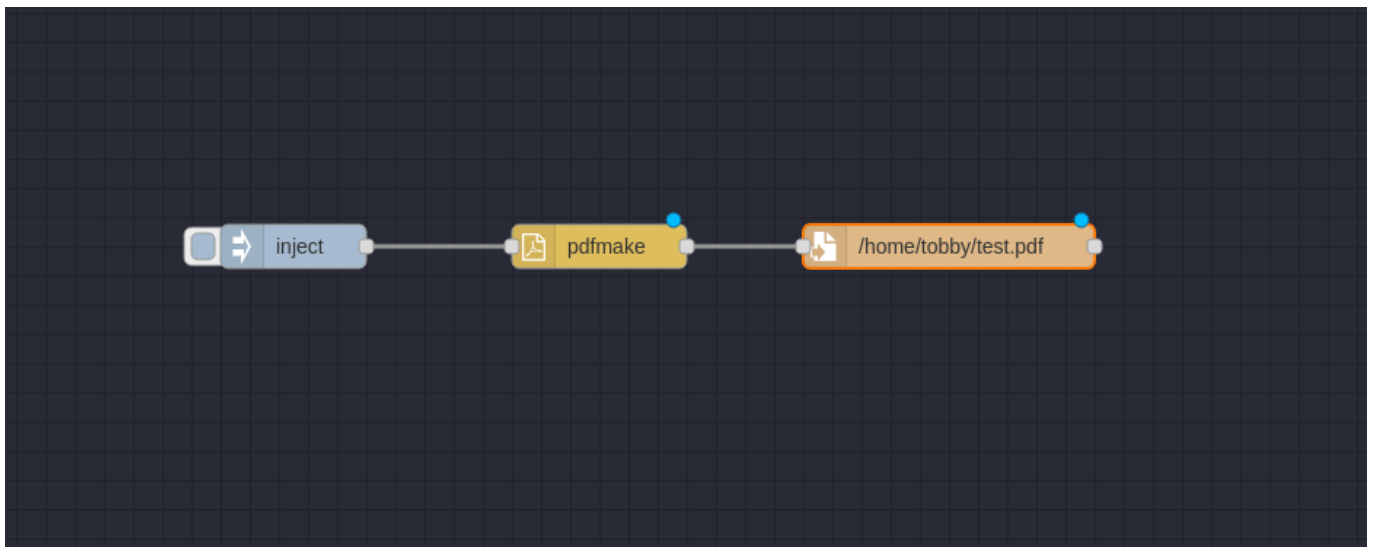
12.4. PDF

12.4.1. Base

Pour generer un PDF, il faut passer un Json dans le payload du message :

```
{
  "_msgid": "b63574aa110e9d9b"
, "payload":
  {
    "content":
      [
        "First paragraph",
        "Another paragraph, this time a little bit longer to make sure,
this line will be divided into at least two lines"
      ]
    },
    "topic": ""
  }
```

Qui est reçu dans le noeud pdfmake qui le passe en Base64 qui est ensuite reçu dans le noeud write file qui l'ecrie dans un fichier PDF.



12.5. Images de graphiques et de tableaux

Une fois que la génération de PDF est maîtrisée, il est temps de valoriser davantage les informations en y ajoutant des images.

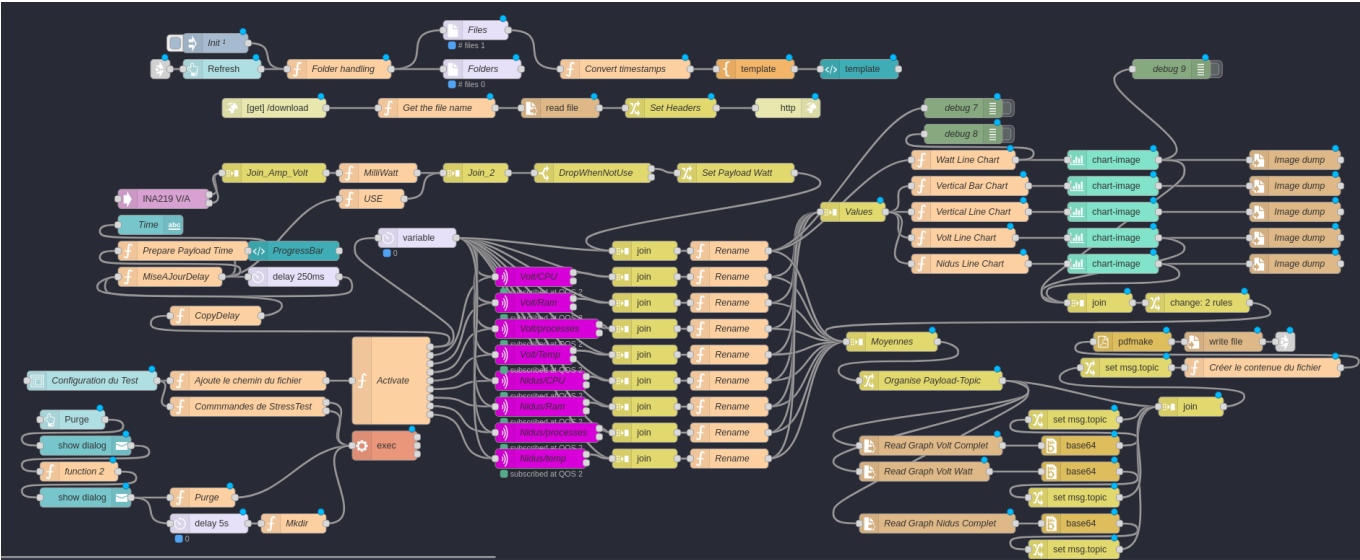
En effet, bien que disposer des valeurs à un instant donné soit utile, pouvoir visualiser ces valeurs sous forme de graphique est encore plus puissant. Pour réaliser cela, nous utiliserons le nœud **node-red-contrib-chart-image**, qui nous permettra de générer des graphiques. Ce nœud repose sur le module **Chart.js**, qui permet de créer des graphiques en utilisant du code JavaScript.

En plus du nœud de graphique, nous aurons besoin du nœud **node-red-node-base64**, qui facilitera la conversion d'images en base64 et vice versa. Cette conversion est essentielle pour intégrer les images dans le document PDF.

Cette combinaison de nœuds nous permettra de créer des représentations visuelles attrayantes et informatives des données, offrant ainsi une compréhension plus approfondie et une présentation visuellement engageante.

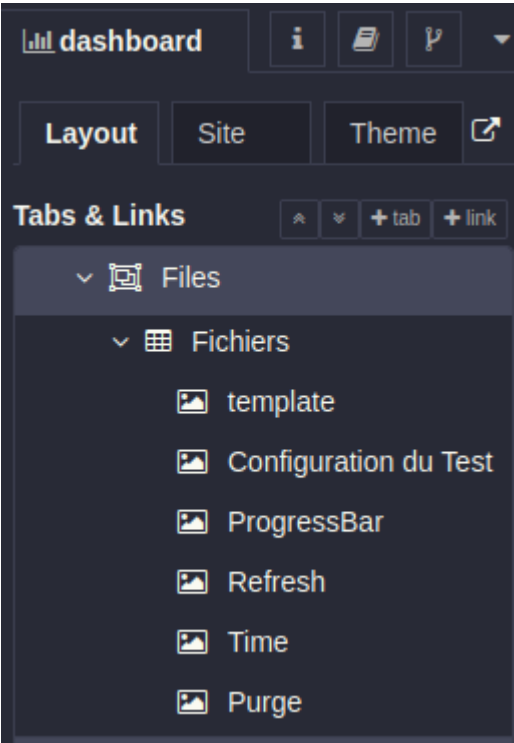
13. Stress Test V1.0

J'ai créé une page qui permet de générer un rapport en fonction de la durée et de l'exécution d'un stress test sur Nidus et/ou sur Volt. Voici le flux complet pour la génération du rapport:



Pour être honnête, il faut admettre que la lisibilité initiale n'est pas optimale. Par conséquent, j'ai décidé de décomposer le processus en plusieurs étapes afin d'obtenir une meilleure compréhension globale.

13.1. Écran d'Accueil



Au premier abord, vous serez accueilli par un navigateur de fichiers et un formulaire. Ce formulaire vous permet de spécifier la durée du test et de décider si vous souhaitez exécuter un test de stress sur Nidus et/ou sur Volt. Voici le contenu de la page "file" qui contient le formulaire:

The screenshot shows a web application interface with a dark theme. At the top, there is a red header bar with a hamburger menu icon and the text "Files". Below this, a dark gray panel contains the "Fichiers" form. The form has the following elements:

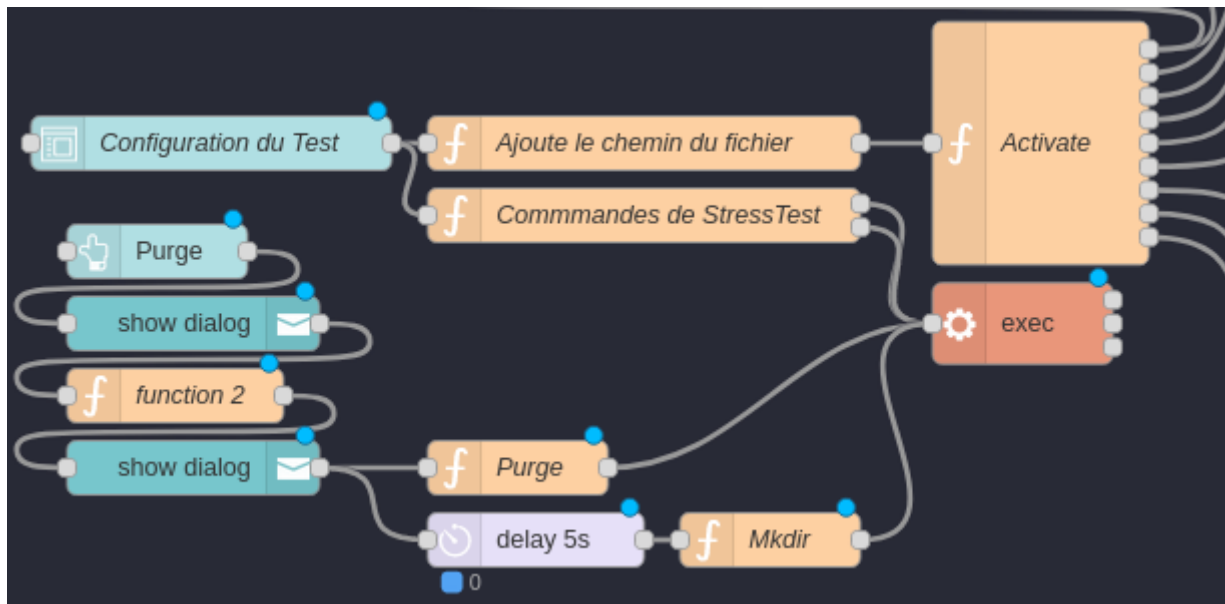
- Nom du Fichier**: A text input field.
- Temps de mesure en minutes**: A text input field with a small icon to its right.
- Stress test Nidus**: A checkbox.
- Stress test Volt**: A checkbox.
- Temps de Stress test**: A text input field with a small icon to its right.
- SUBMIT**: A red button.
- CANCEL**: A red button.
- REFRESH**: A red button with a circular arrow icon, located at the bottom left of the form.
- 00:00**: A red timer display in the center of the form.
- PURGE**: A red button located at the bottom right of the form.

Après avoir rempli le formulaire:

The screenshot shows the same "Fichiers" form, but now it is filled out with the following data:

- Nom du Fichier**: Nidus_Complet
- Temps de mesure en minutes**: 20
- Stress test Nidus**: ☒
- Stress test Volt**: ☒
- Temps de Stress test**: 5
- SUBMIT**: A red button.
- CANCEL**: A red button.
- REFRESH**: A red button with a circular arrow icon, located at the bottom left of the form.

Les nœuds responsables de cette section sont les suivants:



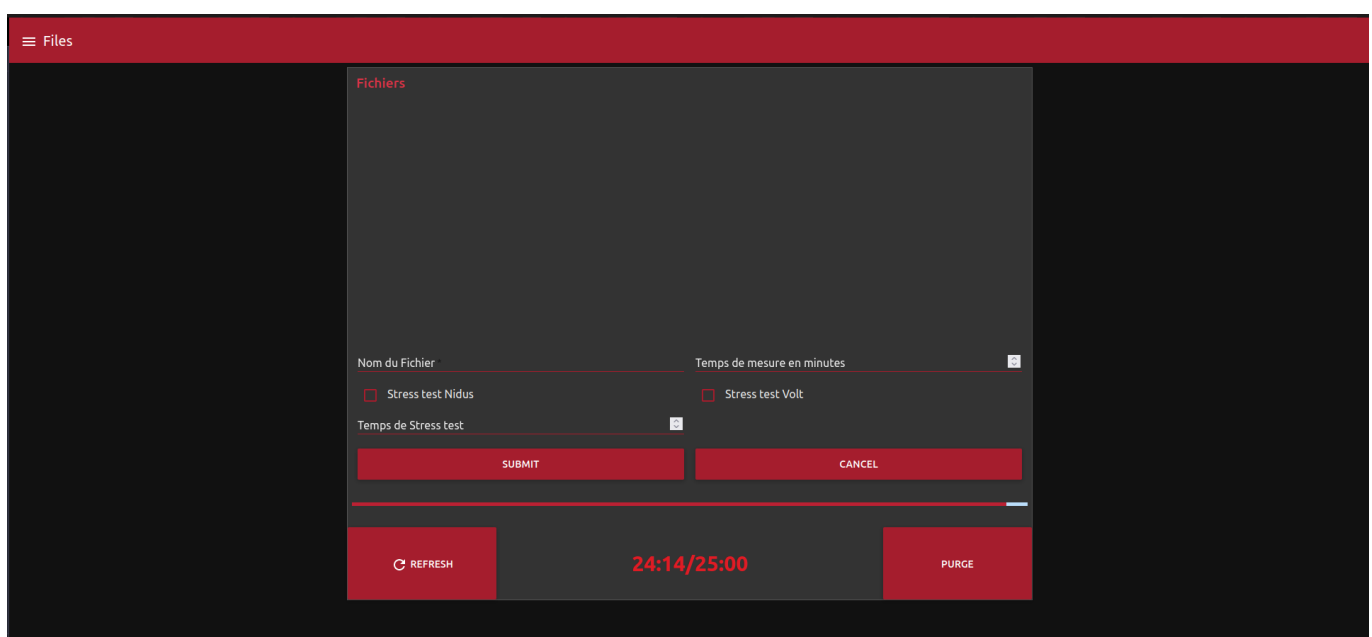
Deux éléments se distinguent ici:

- Un formulaire de "Configuration du Test"
- Un bouton "Purge" dont nous discuterons ultérieurement

Le formulaire recueille les données saisies par l'utilisateur. Ensuite, il transmet ces données en sortie. Deux fonctions sont connectées à cette sortie. La première fonction ajoute les chemins des fichiers, tels que "chart.png" et "report.pdf", à un tableau. La seconde fonction gère l'exécution des tests de stress en fonction des entrées de l'utilisateur, et les envoie ensuite à un nœud "exec" qui exécute les commandes sur Nidus et/ou Volt.

La première fonction transmet ensuite les données à une fonction à sorties multiples, ce qui permet d'envoyer différents messages distincts.

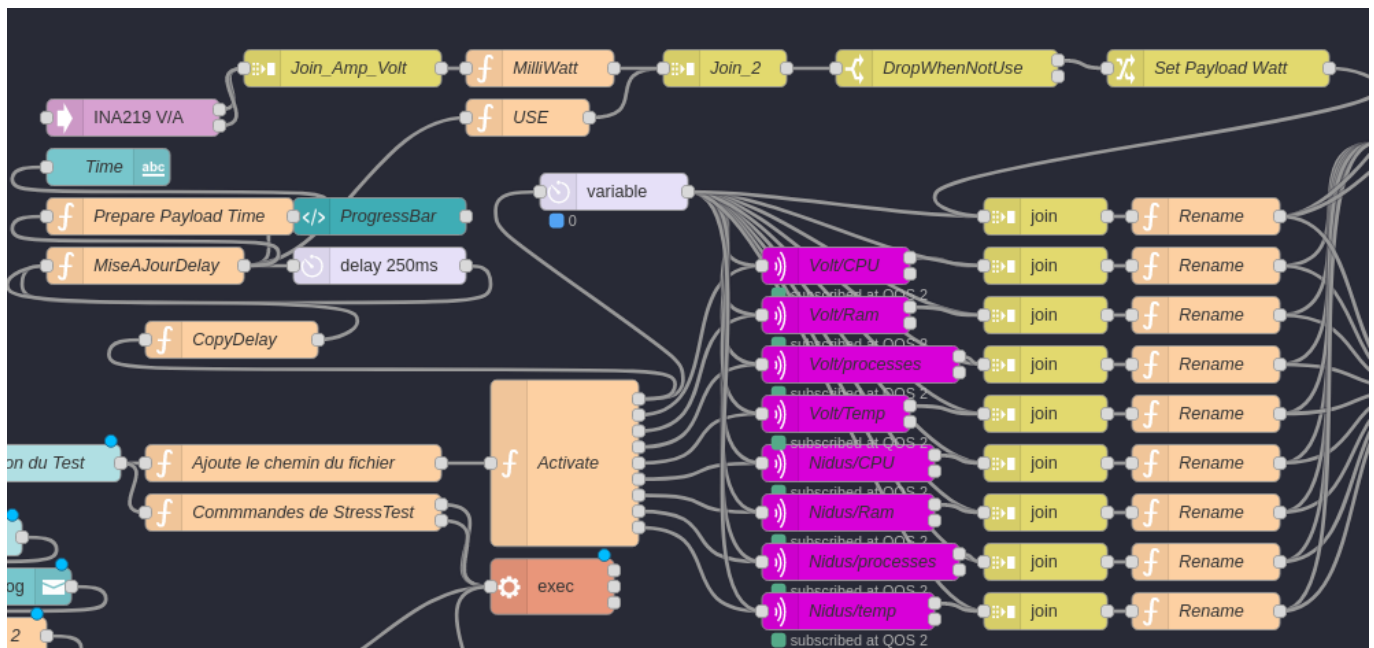
13.2. En Exécution



Pendant l'exécution, une **barre de progression** est affichée pour montrer l'avancement du test, accompagnée d'une **étiquette** en dessous pour indiquer le pourcentage d'avancement. Cela permet

d'obtenir une meilleure visualisation de l'état d'avancement.

En arrière-plan, un certain nombre de tâches se déroulent :



Pour en donner plus de détails :

- La première sortie du nœud **Activate** est connectée à un nœud **delay** qui ajuste la durée du test, ainsi qu'à une série d'autres nœuds qui gèrent la barre de progression.
- La deuxième sortie du nœud **Activate** est reliée à un nœud MQTT amélioré. Celui-ci permet de souscrire aux **topics** appropriés. Au début du test, il souscrit au topic `#/benchmark/#`, puis à la fin du test, il reçoit le topic `/` pour se désinscrire. Cela permet de filtrer uniquement les informations nécessaires et d'éviter d'être submergé par les messages superflus envoyés sur le broker MQTT.
- Les six autres sorties du nœud **Activate** ont la même fonction. Elles envoient toutes des messages pour modifier le topic MQTT.

Ensuite, ces messages sont acheminés vers un nœud **join**, qui les combine en un tableau de messages. Ce tableau est ensuite transmis à un nœud **function** chargé de traiter les données. Parmi les tâches effectuées par ce nœud figurent la personnalisation des **topics** pour chaque ensemble de données et le calcul de la moyenne des valeurs reçues :

```
// Définir le sujet du message
msg.topic = "volt/benchmark/cpu";

// Vérifier si le tableau payload existe et n'est pas vide
if (msg.payload && Array.isArray(msg.payload) && msg.payload.length > 0) {
    // Convertir les valeurs en chaînes de caractères en nombres entiers
    var numericValues = msg.payload.map(function (value) {
        return parseInt(value, 10); // 10 indique la base décimale
    }).filter(function (value) {
        return !isNaN(value); // Filtrer les valeurs non numériques
    });

    // Vérifier si des valeurs numériques ont été trouvées
```



```
if (numericValues.length > 0) {
    // Calculer la somme des valeurs numériques dans le tableau
    var sum = numericValues.reduce(function (acc, value) {
        return acc + value;
    }, 0);

    // Calculer la moyenne en divisant la somme par le nombre
d'éléments
    var moyenne = sum / numericValues.length;

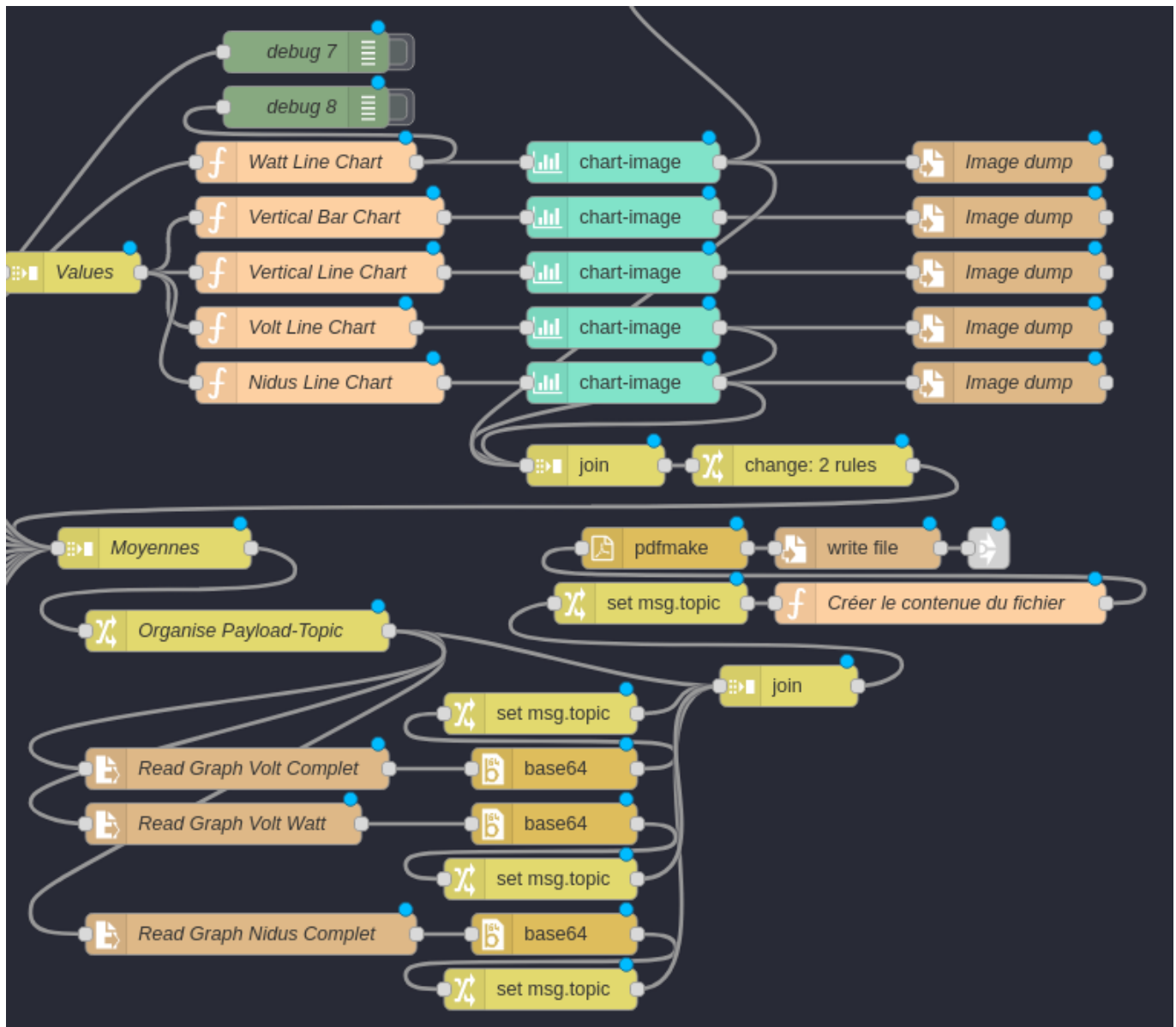
    // Arrondir la moyenne à deux chiffres après la virgule et au
multiple de 0.05 le plus proche
    moyenne = Math.round(moyenne * 20) / 20;

    // Ajouter la moyenne au message
    msg.moyenne = moyenne.toFixed(2);
} else {
    // Si aucune valeur numérique n'a été trouvée, définir la moyenne à
0
    msg.moyenne = "0.00";
}
} else {
    // Si le tableau est vide ou n'existe pas, définir la moyenne à 0
    msg.moyenne.volt.benchmark.cpu = "0.00";
}

// Renvoyer le message modifié
return msg;
```

La partie supérieure permet d'atteindre le même résultat à l'aide de l'INA219. Cependant, puisque je ne peux pas choisir le moment où je veux récupérer les valeurs et qu'elles sont envoyées de manière continue, j'ai utilisé une astuce consistant à détourner les messages de mise à jour de la **barre de progression**. Je les ai synchronisés avec les messages de l'INA219, puis les ai dirigés vers un nœud **join** qui les regroupe. Ensuite, ces messages sont envoyés dans un nœud **switch** qui rejette les messages ne provenant pas de la barre de progression. Cela a pour effet de ne conserver que les messages de l'INA219 pendant le test.

Une fois les ensembles de données collectés, il est temps de les utiliser :



Après l'application des fonctions **rename**, deux nœuds **join** sont utilisés pour regrouper les données. L'un regroupe les tableaux de données, tandis que l'autre regroupe les moyennes calculées.

Intéressons-nous d'abord au nœud **Values**, car c'est le premier à être utilisé. Il permet de créer un tableau de données qui est ensuite transmis à une série de nœuds de fonctions. Ces nœuds de fonctions traitent les données et les formatent pour créer des **graphiques linéaires** sous forme d'images PNG :

```
// Données reçues du flux précédent
var rawData = msg.payload;
var delayInSeconds = msg.delay / 1000; // Conversion en secondes

// Extraction des données nécessaires
var voltWatt = rawData;
// Création du graphique
var chartData = {
  type: 'line', // Changement du type de graphique en "line"
  options: {
    title: {
      display: true,
      text: 'Comparaison des performances'
```

```

    },
    legend: {
        display: true
    },
    chartArea: {
        backgroundColor: '#d3d7dd'
    },
    plugins: {
        datalabels: {
            display: false // Désactiver l'affichage des étiquettes de
données
        }
    },
    data: {
        labels: Array.from({ length: voltWatt.length }, (_, i) => (i *
delayInSeconds).toFixed(1)), // Temps en secondes
        datasets: [
            {
                label: "Volt Watt",
                borderColor: 'rgba(0, 255, 255, 1)',
                fill: false,
                data: voltWatt,
                pointRadius: 0,
            },
        ]
    }
};

msg.payload = chartData;

return msg;

```

L'exemple ci-dessus est volontairement plus simple, car il ne contient qu'un seul ensemble de données, à savoir les watts de Volt. À la sortie de cette fonction, un nœud utilise ce qui a été créé pour générer un **tampon PNG**. Ce tampon est ensuite transmis à un nœud **write file** qui écrit le fichier dans le dossier spécifié par le nœud **Ajoute le nom du fichier**, situé après le formulaire. Simultanément, le nœud envoie également le tampon à un nœud **join** qui attend que tous les graphiques soient créés pour qu'ils puissent être réutilisés.

Une fois que le signal indiquant que les fichiers ont été créés est reçu, le nœud **join** appelé **Moyenne** peut transmettre ses données. Ces données sont réorganisées par un nœud **change**, puis envoyées à plusieurs autres nœuds pour récupérer les images en base64 des graphiques. Ces images sont envoyées en même temps que les moyennes à la fonction **Créer le contenu du fichier** :

```

msg.payload = {
    header: function (currentPage, pageCount, pageSize) {
        return [
            {

```

```
        text: "Tobler Cyril",
        alignment: "left",
        fontSize: 10,
        margin: [15, 10, 0, 0]
    },
    {
        text: "Nom du projet : Confuse T-Rex",
        alignment: "center",
        fontSize: 10,
        margin: [0, 0, 0, 0]
    }
];
},
footer: function (currentPage, pageCount) {
    return {
        columns: [
            {
                text: currentPage.toString() + " / " + pageCount,
                alignment: "left",
                fontSize: 10,
                margin: [15, 0, 0, 10]
            },
            {
                text: new Date().toLocaleDateString("fr-FR"),
                alignment: "right",
                fontSize: 10,
                margin: [0, 0, 15, 10]
            }
        ],
        margin: [0, 0, 0, 10]
    };
},
content: [
    {
        text: "Rapport d'utilisation",
        style: "header",
        margin: [0, 10, 0, 0]
    },
    {
        text: "Les valeurs sont des moyennes sur les " +
Math.floor(msg.delay / (1000 * 60)) + " dernière minutes"
    },
    {
        text: "Nidus :",
        style: "header2"
    },
    {
        text: "CPU : " +
msg.payload.moyenne["nidus/benchmark/cpu"]
    },
    {
        text: "RAM : " +
msg.payload.moyenne["nidus/benchmark/ram"]
    },
]
```

```
{
  text: "Nombre de processus : " +
msg.payload.moyenne["nidus/benchmark/processes"]
},
{
  text: "Température CPU :      " +
msg.payload.moyenne["nidus/benchmark/temp"]
},
{
  image: 'nidusImage',
  width: 500,
  pageBreak: 'after',
},
{
  text: "Volt",
  style: "header2"
},
{
  text: "CPU :                      " +
msg.payload.moyenne["volt/benchmark/cpu"]
},
{
  text: "RAM :                      " +
msg.payload.moyenne["volt/benchmark/ram"]
},
{
  text: "Nombre de processus : " +
msg.payload.moyenne["volt/benchmark/processes"]
},
{
  text: "Température CPU :      " +
msg.payload.moyenne["volt/benchmark/temp"]
},
{
  text: "MilliWatt :                " +
msg.payload.moyenne["volt/benchmark/watt"]
},
{
  image: 'voltImage',
  width: 500
},
{
  image: 'wattImage',
  width: 500
},
],
images: {
  voltImage: 'data:image/png;base64,' +
msg.payload.voltGraph.toString('base64'), // Utilisation du buffer pour
l'image Volt
  wattImage: 'data:image/png;base64,' +
msg.payload.wattGraph.toString('base64'), // Utilisation du buffer pour
l'image Watt de volt
  nidusImage: 'data:image/png;base64,' +
```

```
msg.payload.nidusGraph.toString('base64'), // Utilisation du buffer pour
l'image Nidus
},
styles: {
  header: {
    fontSize: 22,
    bold: true,
    margin: [0, 30, 0, 0]
  },
  header2: {
    fontSize: 18,
    bold: true,
    margin: [0, 20, 0, 0]
  }
}
};

return msg;
```

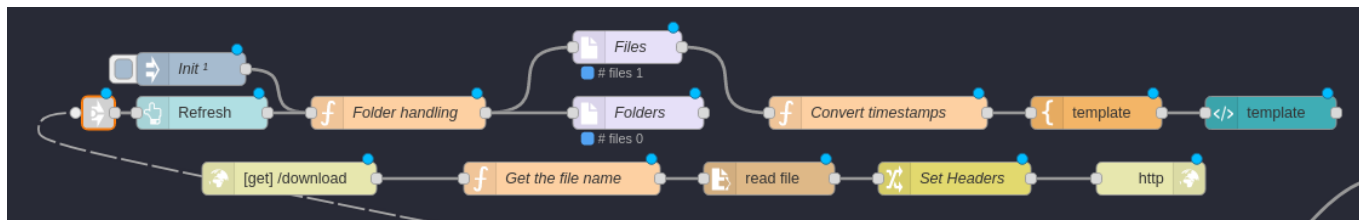
Cette fonction va créer, de manière similaire aux graphiques, une structure utilisée par PDFMake pour générer un fichier PDF. Cette structure est ensuite transmise à un nœud **pdfmake**, qui la convertit en base64 et l'envoie à un nœud **write file**. Ce dernier écrit le fichier PDF dans le dossier spécifié par le nœud **Ajoute le nom du fichier**, situé après le formulaire.

Le nœud final permet de mettre à jour le modèle HTML qui répertorie les fichiers PDF et PNG dans le dossier défini par le nœud **Ajoute le nom du fichier**. Ce modèle HTML permet de les télécharger en un seul clic.

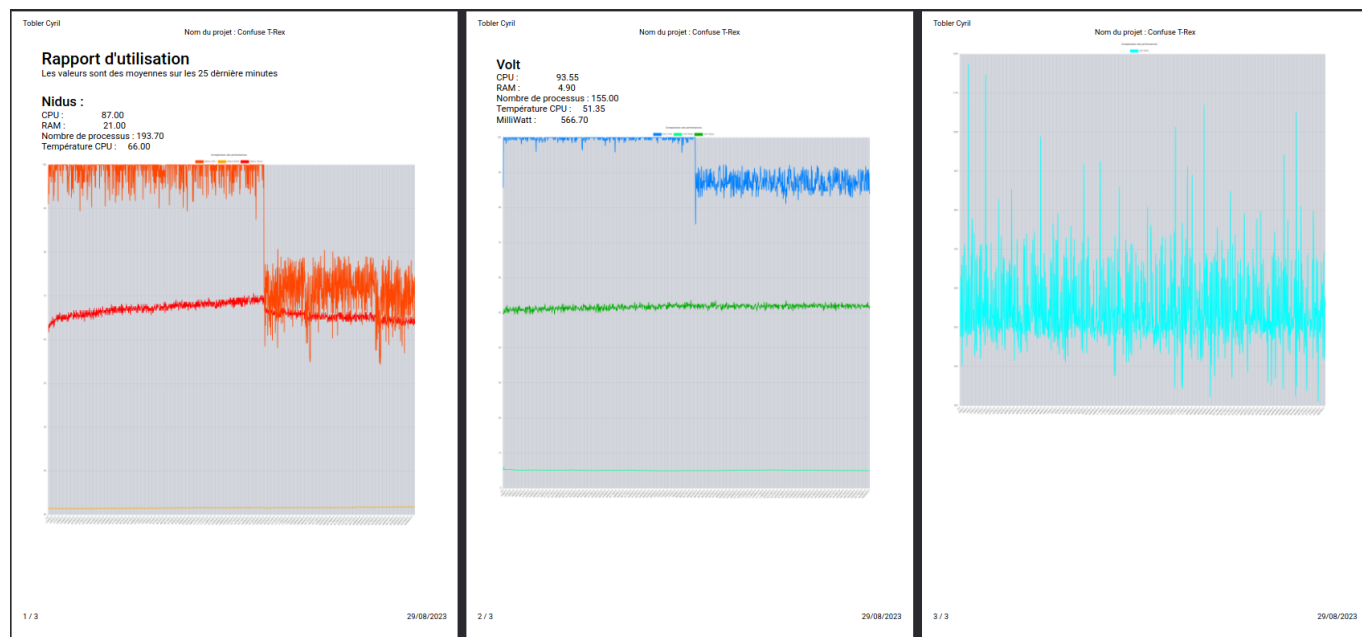
13.3. Résultat

The screenshot shows a web application interface with a dark theme. At the top, there is a red header bar with a hamburger menu icon and the text "Files". Below the header, there is a table titled "Fichiers" with the following columns: "Nom de Fichier", "Taille", "Créer", and "Changer". The table contains one row with the file "Nidus_Volt_25Min_2908.pdf", a size of "1499942", and creation/modification dates of "2023-08-29 12:47:24". Below the table, there is a form with two input fields: "Nom du Fichier" and "Temps de mesure en minutes". There are two checkboxes: "Stress test Nidus" and "Stress test Volt". Below these, there is a "Temps de Stress test" input field. At the bottom of the form, there are two buttons: "SUBMIT" and "CANCEL". Below the form, there is a red bar with a "REFRESH" button, a timer showing "24:39/25:00", and a "PURGE" button.

Pour obtenir les résultats, il suffit de cliquer sur le nom du fichier, qui sera automatiquement téléchargé. Ce processus est géré par ces nœuds :

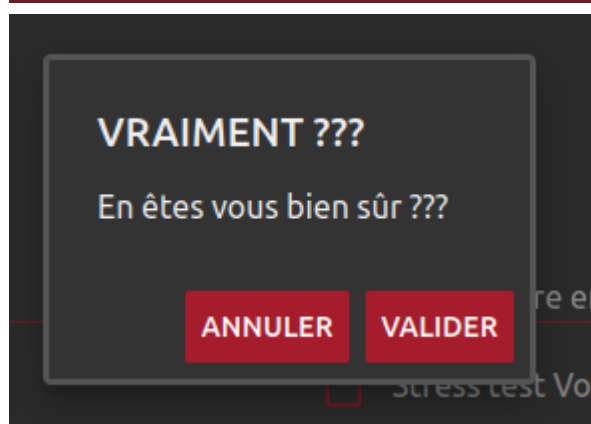


La partie supérieure gère l'affichage des fichiers dans un modèle et ajoute aux noms de fichier des requêtes GET qui permettent de télécharger les fichiers en un seul clic. La partie inférieure gère la réception des requêtes GET et envoie le fichier demandé à un nœud **read file**, qui le lit et l'envoie ensuite à un nœud **http response**. Ce dernier envoie le fichier au client ayant effectué la requête.

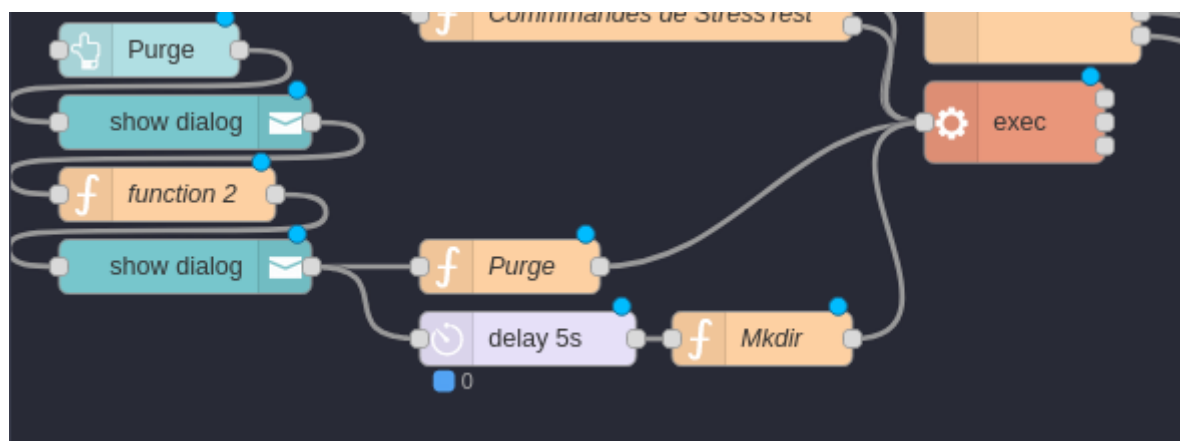


13.4. Purge

Au cours de mes tests, j'ai réalisé qu'un problème survient lorsque l'on génère un certain nombre de rapports, le dossier devient rapidement surchargé. Par conséquent, j'ai décidé de mettre en place un bouton permettant de purger le dossier de tous les fichiers .pdf et .png qui s'y trouvent. Cependant, pour éviter toute suppression accidentelle de fichiers importants, j'ai mis en place un système de confirmation demandant à l'utilisateur s'il est sûr de vouloir supprimer les fichiers.



Voici les nœuds qui gèrent cette partie :



Ce que l'on peut observer, c'est qu'après avoir appuyé sur le bouton de purge, un message est envoyé dans un nœud `show dialog` qui affiche une fenêtre de confirmation. Si l'utilisateur appuie sur le bouton "Oui", un message est transmis à un nœud de fonction qui vérifie le contenu du message et redemande une confirmation s'il est à nouveau validé. À ce stade, deux flux sont créés :

- Le premier effectue la purge totale de tous les fichiers dans `/home/NodeRed/`.
- Le second commence par un délai de quelques secondes avant de recréer les dossiers de structuration.

14. Sources

1. **Guide d'Installation Node-Red**
[Installer Node-Red](#)
2. **Guide de Sécurisation de Node-Red**
[Sécurisation de Node-Red](#)
3. **Tutoriel de Base Rototron**
[Tutoriel Rototron](#)
4. **Documentation Technique de l'INA219**
[Documentation INA219](#)
5. **Recherche de M. Lamber**
[Profil de Consommation par M. Lamber](#)
6. **Recherche de M. Pol J. Planas Pulido**
[Profil de Consommation par M. Pol J. Planas Pulido](#)
7. **Bibliothèque Python pour l'INA219**
[Bibliothèque pi-ina219](#)
8. **Forum Problème de Détection I2C**
[Forum Raspberry Pi](#)
9. **Tutoriel Mise en Place INA219**
[Tutoriel INA219](#)
10. **Tutoriel Création d'un Enregistreur de Consommation**
[Tutoriel Enregistreur de Consommation](#)
11. **Tutoriel Complet avec Arduino**
[Tutoriel Complet avec Arduino](#)
12. **Téléchargement Gatling**
[Téléchargement Gatling](#)
13. **Tutoriel Avancé Gatling**
[Tutoriel Avancé Gatling](#)
14. **Tutoriel de Démarrage Rapide Gatling**
[Tutoriel de Démarrage Rapide Gatling](#)
15. **Tutoriel sur l'utilisation de S1seven**
[Tutoriel S1seven](#)