

# Un serveur pour le FabLab - YunoHost ?

Installer un serveur yunohost.org sur un Raspberry pour les besoins du FabLab : Projets d'IoT  
Projets de robotique Groupe de travail IA

- [Installer un serveur avec la distribution Linux YunoHost](#)
- [Serveur d'impression 3D - Octoprint](#)
- [Moyens informatiques](#)
- [Serveur de colocation d'Alsace Réseau Neutre](#)
- [Serveur IoT - MQTT, SQL, Node-RED, InfluxDB](#)

# Installer un serveur avec la distribution Linux Yunohost

- Flasher [Yunohost pour Raspberry](#) avec Etcher (sous Linux) ou Rufus (sous Windows) sur une carte SD
- Connecter la Raspberry à un routeur via un câble Ethernet
- **Si** le routeur autorise la Raspberry à récupérer une adresse IP via DHCP et à émettre un nom de domaine via mDNS
  - Se connecter à [yunohost.local](#) depuis un ordinateur connecté au routeur
- **Sinon**
  - [Se connecter avec écran+clavier](#) pour récupérer l'adresse IP
    - user : root
    - mdp : yunohost
    - attention par défaut le clavier est anglais
  - Se connecter à l'adresse IP récupérée depuis un ordinateur connecté au routeur
- Lancer la post-installation pour créer le premier utilisateur Yunohost qui en sera administrateur
  - nécessite une bonne connexion internet
- Mettre un mot de passe fort car il s'agit d'un serveur qui pourra potentiellement être exposé sur internet
- Ajouter le lien entre IP et adresses [https://yunohost.org/oc/dns\\_local\\_network#configure-hosts-file-on-c](https://yunohost.org/oc/dns_local_network#configure-hosts-file-on-c)

## Connecter la Raspberry en wifi plutôt qu'ethernet

On ne pourra plus configurer de HotSpot Wifi, à moins d'ajouter une carte wifi supplémentaire (dongle USB par exemple)

- Pour que le Raspberry (Yunohost 11, debian bullseye) se connecte en Wifi (par défaut il ne se connecte qu'en filaire)
  - Écrire un fichier `wpa_supplicant.conf` dans la partition boot
  - voir : [https://www.raspberryme.com/configurer-le-wifi-sur-un-pi-manuellement-a-laide-de-wpa\\_supplicant-conf/](https://www.raspberryme.com/configurer-le-wifi-sur-un-pi-manuellement-a-laide-de-wpa_supplicant-conf/)
  - Bien penser à modifier :

```
country=fr
ssid="MyNetworkSSID"
psk="Pa55w0rd1234"
```

- A partir de Yunohost 12 (Debian bookworm), il faudra suivre [la doc' officielle](#)

## Configurer un HotSpot Wifi

- Configurer un point d'accès wifi qui partage la connexion ethernet de la brique avec les appareils connectés (bridge).
- Installer l'application <https://apps.yunohost.org/app/hotspot>

Alternative : <https://www.raspberrypi.com/documentation/computers/configuration.html#use-your-raspberry-pi-as-a-network-bridge>

## Ajouter du SWAP

- Pour les Raspberry 3B+ qui n'ont que 1G de RAM, il est préférable d'ajouter un fichier de SWAP d'au moins 1G
- Cela permettra d'installer les applications YunoHost même si la RAM demandée est importante
- <https://www.tartarefr.eu/fr/blog/modifier-swap-rpi>

```
sudo dphys-swapfile swapoff
sudo nano /etc/dphys-swapfile
```

- Modifier le fichier de configuration

```
CONF_SWAPSIZE=4096
...
CONF_MAXSWAP=4096
```

- Recréer le fichier de SWAP et redémarrer le SWAP

```
sudo dphys-swapfile setup
sudo dphys-swapfile swapon
```

Impérativement régler la swappiness à 1 sinon la carte SD va mourir prématurément

# Raspberry 5 avec NVME

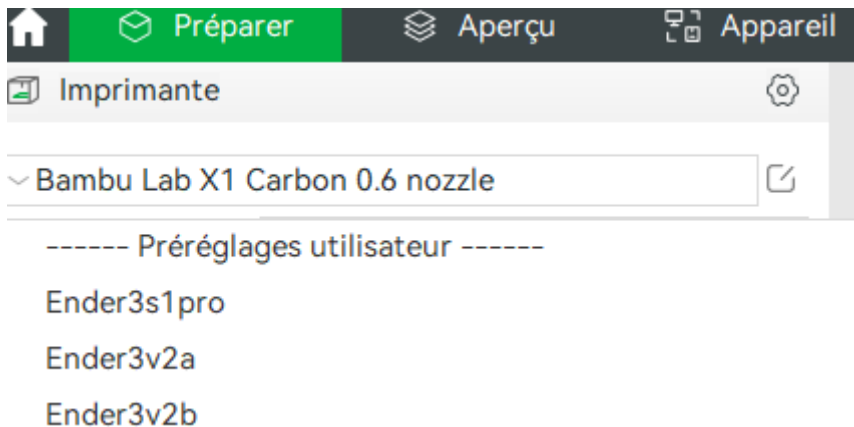
- suivre cette documentation : <https://www.framboise314.fr/nvme-base-de-pimoroni-ajoutez-un-ssd-m2-a-votre-raspberry-pi-5/>
- Sans interface graphique et en installant d'abord [YunoHost sur la carte SD](#)
- Flasher l'intégralité de la carte SD vers le disque NVME
- `dd /dev/sda /dev/nvme0n1`
- Monter la partition qui contient `/boot/firmware`
- `mount /dev/nvme0n1p1 /mnt/nvme0n1p1`
- bien vérifier que la ligne apparait à la fin de `/mnt/nvme0n1p1/config.txt`
  - `dtparam=pciex1_gen3`
- vérifier que le UUID apparaissant dans le fichier `/etc/fstab` est bien celui du disque NVME et non celui de la carte SD.

# Serveur d'impression 3D - Octoprint

## Lancement d'une impression

### Depuis Bambu Studio

- Sélectionner l'imprimante :



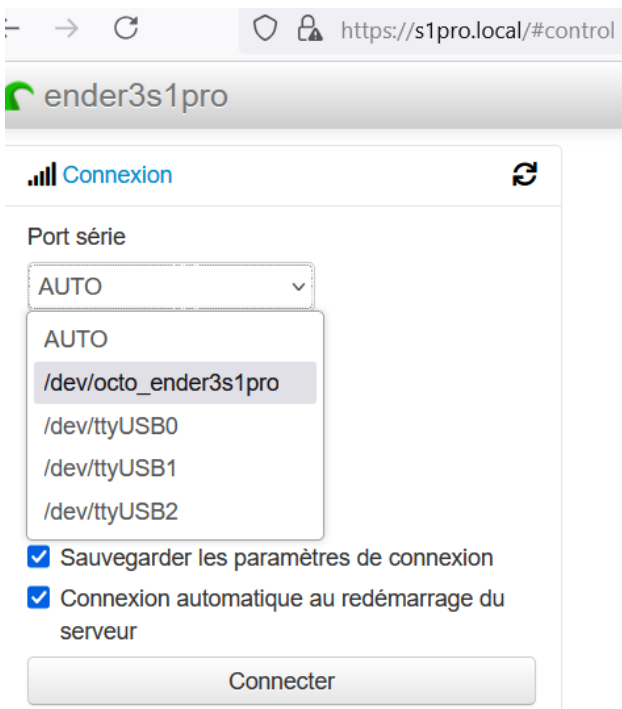
- Lancer l'impression

Pour voir comment ajouter une imprimante à BambuStudio via Octoprint :

<https://innovation.iha.unistra.fr/books/3-fablab-machines-techniques/page/bambu-studio#bkmrk-autre-imprimante-via>

### Depuis un navigateur

- Se connecter au serveur d'impression Octoprint correspondant à l'imprimante souhaitée, cf . <https://innovation.iha.unistra.fr/books/1-fablab-vos-projets/page/adressage-ip-du-reseau-du-fablab#bkmrk-configuration-r%C3%A9seau>
- Démarrer l'imprimante
- Se connecter à l'imprimante en sélectionnant le port série correspondant, par ex.  
`/dev/octo_ender3s1pro`



# Installation de Octoprint

Nous avons vu comment [installer un serveur YunoHost sur une Raspberry Pi](#).

On souhaite maintenant installer plusieurs instances de Octoprint pour pouvoir piloter à distance plusieurs imprimantes 3D

- connexion via ssh au serveur
- `mkdir -p /opt/octoprint_deploy`
- `cd /opt/octoprint_deploy`
- `git clone https://github.com/paukstelis/octoprint\_deploy.git`
- débrancher l'imprimante à ajouter
- lancer le script de déploiement et d'upgrade : `./octoprint_deploy.sh`
- taper `y` ou `n` en fonction des questions
- rebrancher l'imprimante lorsque demandé pour la détection du port USB

Vérification que le système tourne :

- `sudo systemctl status ender3v2a`

```
● ender3v2a.service - The snappy web interface for your 3D printer
   Loaded: loaded (/etc/systemd/system/ender3v2a.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2024-06-20 09:17:48 CEST; 25min ago
     Main PID: 51985 (octoprint)
        Tasks: 14 (limit: 779)
           CPU: 1min 33ms
```

```
CGroup: /system.slice/ender3v2a.service
```

```
└─51985 /home/fan_admin/OctoPrint/bin/python3
```

```
/home/fan_admin/OctoPrint/bin/octoprint serve --config=/home/fan_admin/.ender3v2a/config.yaml  
--basedir=/home/fan_admin/.ender3v2a --port=XXXXXX
```

- le système est bien `active (running)`
- le port `XXXXXX` est indiqué
- on se connecte donc à l'interface web d'octoprint via [http://adresse\\_ip:XXXXXX](http://adresse_ip:XXXXXX)

A condition d'avoir bien ouvert le port `XXXXXX` dans le pare-feu YunoHost

Démarrage automatique du service systemd au démarrage de la Raspberry :

- `sudo systemctl enable ender3s1pro`

Redémarrage du service au bout de 5s lorsqu'il échoue :

- `nano /etc/systemd/system/ender3s1pro.service`

```
[Service]
```

```
Restart=on-failure
```

```
RestartSec=5s
```

## Exposition d'Octoprint sur le réseau local via YunoHost :

- ajout d'une URL locale type `imprimante.local`



[🏠](#) / [Domaines](#) / Ajouter un domaine

## 🌐 Ajouter un domaine

- Je veux ajouter un domaine que je possède, ou un sous-domaine

❗ Vous devrez configurer manuellement les enregistrements DNS et la configuration de ce domaine. Le diagnostic de YunoHost vous guide

Nom de domaine

v2b.local|

- Installation d'une application redirect sur l'URL créée et pointant sur l'adresse locale et le port de l'imprimante



## Paramètres d'installation

Libellé pour Redirect

Ender3 V2 A

Il s'agit du nom affiché dans le portail d'u

Choisissez le domaine sur lequel vous souhaitez installer cette application

v2a.local

Choisissez le chemin d'URL (après le domaine) où cette application doit être installée

/

Be careful when using this app in reverse-meaning the path is just '/'!

Type de redirection

Reverse-proxy (nginx proxy\_pass).

Cible

http://192.168.5.2:5001

This may be something like <https://some.>

Qui doit avoir accès à cette application ? (Ceci peut être modifié ultérieurement)

Visiteurs

## Exposition d'Octoprint sur le net via YunoHost :

Vous vous exposez à des attaques, sécurisez les comptes et vérifiez les droits d'accès avant de réaliser cela

- Ouverture du port dans le Firewall YunoHost  
<https://fan.ynh.fr/yunohost/admin/#/tools/firewall>

|        |          |      |        |           |        |           |            |
|--------|----------|------|--------|-----------|--------|-----------|------------|
| Action | Ouvrir ▾ | Port | 5001 ▾ | Connexion | IPv4 ▾ | Protocole | Les deux ▾ |
|--------|----------|------|--------|-----------|--------|-----------|------------|

- ajout d'une URL publique



[🏠](#) / [Domaines](#) / Ajouter un domaine

## 🌐 Ajouter un domaine

- Je veux ajouter un domaine que je possède, ou un sous-domaine

📘 Vous devrez configurer manuellement les enregistrements DNS et la configuration de ce domaine. Le diagnostic de YunoHost vous guide

Nom de domaine

v2b.local|

- Installation d'une application redirect

## ⚙ Paramètres d'installation

Libellé pour Redirect

Ender3 V2 A

Il s'agit du nom affiché dans le portail d'u

Choisissez le domaine sur lequel vous souhaitez installer cette application

v2a.local

Choisissez le chemin d'URL (après le domaine) où cette application doit être installée

/

Be careful when using this app in reverse-meaning the path is just '/'!

Type de redirection

Reverse-proxy (nginx proxy\_pass).

Cible

http://192.168.5.2:5001

This may be something like <https://some.>

Qui doit avoir accès à cette application ? (Ceci peut être modifié ultérieurement)

Visiteurs

## Installation de Klipper

- Ce nouveau firmware à installer sur le contrôleur de l'imprimante permet de permettre d'améliorer les performances. Les algorithmes de planification de trajectoire fonctionnent sur un ordinateur déporté, par exemple notre serveur d'impression 3D
- Installation managée via KIAUH <https://github.com/dw-0/kiauh>

- ```
sudo apt-get update && sudo apt-get install git -y
cd ~ && git clone https://github.com/dw-0/kiauh.git
./kiauh/kiauh.sh
```

- Installation des dépendances

```
sudo apt-get install avr-libc avrdude binutils-arm-none-eabi binutils-avr dfu-util
gcc-arm-none-eabi gcc-avr libnewlib-arm-none-eabi libusb-1.0-0-dev libusb-dev
stm32flash
```

- ```
cd ~ && git clone https://github.com/Klipper3d/klipper.git
```

- on installe 4 instances en respectant les noms des imprimantes

<https://innovation.iha.unistra.fr/books/1-fablab-vos-projets/page/adressage-ip-du-reseau-du-fablab#bkmrk-configuration-r%C3%A9seau>

- (Méthode d'installation "manuelle" : <https://www.klipper3d.org/OctoPrint.html> )
- Configuration du printer.cfg via ssh

<https://www.klipper3d.org/Installation.html#configuring-klipper>

```
cp klipper/config/printer-creality-ender3-s1-2021.cfg
```

```
/home/fan_admin/printer_ender3s1pro_data/config
```

```
nano /home/fan_admin/printer_ender3s1pro_data/config/printer-creality-ender3-s1-2021.cfg
```

```
mv /home/fan_admin/printer_ender3s1pro_data/config/printer-creality-ender3-s1-2021.cfg
```

```
/home/fan_admin/printer_ender3s1pro_data/config/printer.cfg
```

- Compiler et flasher le driver <https://www.klipper3d.org/Installation.html#building-and-flashing-the-micro-controller>
- Intégration avec Octoprint (qui est déjà installé)

<https://www.klipper3d.org/OctoPrint.html#configuring-octoprint-to-use-klipper>

- Installation de l'interface Web Mainsail requiert d'abord d'installer Moonraker

```
kliauh/kliauh.sh
```

# Moyens informatiques

## Synthèse

cf. <https://innovation.iha.unistra.fr/books/1-fablab-vos-projets/page/adressage-ip-du-reseau-du-fablab#bkmrk-synth%C3%A8se-des-moyens->

## PC

- 3 PC tours Ubuntu 22.04 administrés par le FabLab
  - Pour les AppImage sous Ubuntu 24.04 qui a FUSE3 par défaut --> [installer FUSE2](#)  
`sudo apt install libfuse2`
  - Eviter les m2j sur le réseau du FabLab car limité à 20Go, plutôt faire un partage de connexion smartphone via USB ou se connecter à Osiris
  - Arduino IDE (v1.8 dans les dépôts) --> Télécharger [AppImage](#) (2.3.2)  
`chmod +x arduino-ide...AppImage`
  - Télécharger [BambuStudio.AppImage](#)  
`chmod +x BambuStudio...AppImage`
  - FreeCAD 0.21 dans le [PPA stable](#)  
`sudo add-apt-repository ppa:freecad-maintainers/freecad-stable`
  - Inkscape 1.3.2 dans le [PPA stable](#)  
`sudo add-apt-repository ppa:inkscape.dev/stable`
    - [Extension Inkcute pour traceur de découpe](#)
    - Extension Lasercut tabbed box
    - Extension Lasercut Jigsaw
  - Firefox LTS [via les dépôts plutôt que le snap](#)
    - cf. <https://innovation.iha.unistra.fr/books/robotique-open-source/page/installation-pc-ros2#bkmrk-installer-firefox-da>
  - PrusaSlicer depuis [snap](#)  
`sudo snap install prusa-slicer`
  - Ultimaker Cura [via les dépôts](#)  
`sudo apt install cura`
- [Logiciels utiles pour la Robotique Open Source ROS](#)
  - QTcreator-ROS depuis snap (pour la robotique Open Source)
  - Visual Studio Code sans tracker [VSCodium](#)

- Profils et Droits Ubuntu
  - Un profil public
    - Pour tout utilisateur du FabLab
    - Français
    - Automatic Login
    - Même Password qu'étudiant
  - un profil étudiant en robotique
    - Pour les étudiants GEII ou toute personne voulant découvrir ou travailler sur la robotique Open Source avec ROS
    - administrateur
    - Anglais (pour ROS)
  - en profil enseignant/administrateur
    - Pour les enseignants de robotique, quelques workspace ROS propres, que les étudiants ne devraient pas avoir modifiés (rien ne les en empêche)
    - administrateur
    - Anglais (pour ROS)
- 3 PC tours Windows administrés par l'IUT

## Double-écran en salle A1-TP06

L'idée c'est de brancher un all-in-one comme écran secondaire du all-in-one voisin pour avoir 2 écrans. C'est pratique d'avoir 2 écrans quand on fait du VirtualBox et qu'on veut suivre de la doc' à côté. Voyons comment faire ça avec 2 Dell Optiplex 7470 en A1TP06 ?

Sur le all-in-one qui a besoin d'un écran externe : brancher le câble sur le port HDMI "normal"

Sur le all-in-one qui doit servir d'écran externe :

- installer ce driver Dell "On screen display" [https://dl.dell.com/FOLDER05872073M/5/Dell-On-Screen-Display-Application\\_PRGT0\\_WIN\\_1.0.4.0\\_A02\\_03.EXE](https://dl.dell.com/FOLDER05872073M/5/Dell-On-Screen-Display-Application_PRGT0_WIN_1.0.4.0_A02_03.EXE)
- brancher le câble HDMI sur le port qui a un logo "input"
- appuyer sur le bouton qui est sous l'écran à droite
- le bouton est désactivable dans le BIOS mais activé par défaut
- déploiement du driver en ligne de commande :

```
PS C:\Users\install> .\Dell-On-Screen-Display-Application_PRGT0_WIN_1.0.4.0_A02_03.EXE /help
PS C:\Users\install> A02
Dell On-Screen Display Application, 1.0.4.0, A02
```

General Usage:

```
Dell-On-Screen-Display-Application_PRGT0_WIN_1.0.4.0_A02_03.exe [ /<option1>[=<value1>]]
[ /<option2>[=<value2>]]...
```

See listing below for <option> and <value> information.

NOTE: If the folder name contains space, double quotes should be given for folder name

EXAMPLE: /s /e="c:\Folder Name"

Option - Description:

-----

(none) - Display of graphical user interface for guided installation/update or extraction.

/? or /h - Displays this Update Package usage information.

/s - Suppresses all graphical user interfaces of the Update Package.

/e=<path> - Extraction of update contents to a folder.

(NOTE: Requires /s option)

/passthrough ... - (Advanced) Sends all text following the /passthrough option directly to the vendor install software of the Update Package. This mode suppresses any provided graphical user interfaces, but not necessarily those of the vendor software.

/factoryinstall /passthrough (vendor command line parameter) - (Advanced) (command line parameters) will be passed to vendor installer directly without parsing MUP.xml. Post installation, If the vendor installer returns REBOOT\_REQUIRED or REBOOT\_UPDATE\_PENDING, it will be turned to SUCCESS.

/l=<path> - Define a specific path for the Update Package log file.

(NOTE: This option can NOT be used in combination with /passthrough)

Example(s):

-----

Update the system silently

Dell-On-Screen-Display-Application\_PRGT0\_WIN\_1.0.4.0\_A02\_03.exe /s

Extract the update contents to the folder C:\mydir\

Dell-On-Screen-Display-Application\_PRGT0\_WIN\_1.0.4.0\_A02\_03.exe /s /e=C:\mydir

Pass command line arguments directly to vendor installer.

Turn the return code to success if required

Dell-On-Screen-Display-Application\_PRGT0\_WIN\_1.0.4.0\_A02\_03.exe /factoryinstall /passthrough

D: \Sample.xml C: \log\FI.log

Change from the default log location to C:\my path with spaces\log.txt

Dell-On-Screen-Display-Application\_PRGT0\_WIN\_1.0.4.0\_A02\_03.exe /l="C:\my path with spaces\log.txt"

# Réseau

- Routeur wifi 5G CradlePoint
  - Émet un réseau wifi isolé de l'infrastructure réseau de l'IUT dans le FabLab (A1-01), le bureau du Fab-Manager (A1-05) et les deux salles adjacentes (A1-TP01 et A1-TP06)
  - Connexion internet via SIM 5G SFR 20Go de quota 2€/mois
  - Extension possible à un forfait illimité pour 10€/mois
  - Passer en navigation privée si la connexion échoue
- Permet de créer un réseau et de fournir internet pour des PC, serveurs, et robots Linux/Ubuntu
- Permet de réaliser des projets de robotique avec ROS en connectant des ordinateurs sous Linux/Ubuntu avec des IP fixes
- Permet de réaliser des projets d'IoT avec un serveur YunoHost exposé à internet
- [VPN Wireguard arn-fai.net](#) 4€/mois

# Serveurs

- Un raspberry 3B+ installé sous Yunohost
  - Usages locaux : serveur d'impression 3D octoprint
  - Usages internet : expérimentations serveur autour de la collecte de données (IoT)
  - OS Yunohost 11
  - Connecté en filaire au Routeur 4G CradlePoint
  - [VPN Wireguard arn-fai.net](#) 4€/mois
  - Accès local via `yunohost.local/yunohost/admin`
  - Exposition sur internet via [VPN Wireguard arn-fai.net](#) 4€/mois
  - Nom de domaine public fourni par Yunohost : [fan.ynh.fr](#)
  - Application serveur d'impression 3D [Octoprint](#) pour Ender3 S1Pro, Ender3 V2 A, Ender3 V2 B
  - Applications YunoHost installées :
    - WireGuard Client pour la connexion au serveur VPN wireguard
    - Home Assistant
      - pour faire de l'IoT
      - pour synthétiser les serveurs Octoprint



- InfluxDB pour le stockage de données temporelles issues des capteur IoT
- Mosquitto comme serveur MQTT
- Node-RED pour programmer la récupération de données depuis les capteurs, les afficher sur un dashboard et les stocker dans une base de données MySQL ou InfluxDB
- phpMyAdmin pour gérer les bases de données MySQL
- Un [serveur virtuel \(VPS\) en colocation chez arn-fai.net](#)
  - Administré par le FabLab avec l'aide des administrateur bénévoles de l'association Alsace Réseau Neutre
  - Un nom de domaine fablab-alsacenord.fr payé chez ovh
  - Usage IA
    - Un compte OpenAI du FabLab payé par la caisse de l'association
    - L'[application ChatGPT web](#) pour se connecter à l'API OpenAI
  - Usage FabLab
    - Le [site web de gestion du FabLab](#) d'Alsace-Nord : fab-manager.com
    - Installé via le paquet YunoHost
- Un Raspberry 5 installé sous HomeAssistant OS
  - 8G RAM
  - [HomeAssistant OS](#) installé
  - Pour la [gestion de flotte de capteurs connectés ESP32](#)
  - Pour la [collecte, l'affichage et le traitement des données](#)

## Gestion parc informatique avec GLPI

Pour gérer les PC Ubuntu du FAN et les PC Windows de CréaBot, on installe le logiciel GLPI avec Fusion Inventory :

- Installer le [paquet GLPI](#) sur le raspberry 3B+ `fan. ynh. fr`
- Installation sur un domaine local, on ne veut pas de piratage de l'extérieur : `parcinfo.local`
- Login par défaut : glpi:glpi

Source : <https://openclassrooms.com/fr/courses/1730516-gerez-votre-parc-informatique-avec-glpi/5994176-installez-le-plugin-et-l-agent-fusioninventory>

## Autre approche avec Ansible

<https://github.com/richlamdev/ansible-desktop-ubuntu>

# Serveur de colocation d'Alsace Réseau Neutre

## Fab-Manager

### Configuration

- <https://manager.fablab-alsacenord.fr>
- déploiement via [app YunoHost](#)
- installé dans `cd /var/www/fab-manager`
- fichier de config : `nano .env`
- recharger le service systemd : `sudo systemctl restart fab-manager-worker fab-manager-app`
- renouvellement du certificat lets'Encrypt géré par YunoHost
- Configuration du paiement en ligne via Stripe

The screenshot displays the 'Paramètres de paiement' (Payment Settings) page in the Fab-Manager application. The interface features a red sidebar menu on the left with various navigation options. The main content area has a top navigation bar with tabs for 'Liste des factures', 'Échéanciers de paiement', 'Paramètres de facturation', 'Codes comptables', and 'Paramètres de paiement'. The 'Paramètres de paiement' tab is selected, showing a 'Paieement en ligne' (Online Payment) section. This section includes a toggle for 'Activer les paiements en ligne' (Activate online payments), which is currently turned on, and an 'enregistrer' (Save) button. Below this is the 'Clefs Stripe' (Stripe Keys) section, which contains a 'Clef publique' (Public key) field with the value 'pk\_test\_51NGjJBVN1N9qWGOL5OPIVAQAgveOespD4PmM...' and a 'Clef secrète' (Secret key) field with masked characters. A 'Modifier les clefs' (Modify keys) button is also present. The 'Devise' (Currency) section includes a message about the currency and a 'Devise Stripe' dropdown menu set to 'EUR'.

- Test paiement en ligne :
  - Utiliser une CB de test de Stripe <https://docs.stripe.com/testing#cards>

[Automatisation de la publication des projets de Fab-Manager vers Facebook](#)

<https://fabmanager.csc49.fr/#!/projects/publication-des-projets-fabmanager-sur-le-site-du-bocal-et-la-page-facebook>

## Ancien serveur hébergé Unistra

- <https://fan-manager.di.unistra.fr/>
- déploiement via docker comme dans la [doc' officielle](#)
- installé dans `cd /apps/fabmanager`
- fichier de config : `nano config/env`
- recharger l'environnement :

```
docker compose down
docker compose up -d
```

- script de renouvellement du certificat lets'Encrypt (à programmer avec crontab)

```
root@fan-manager: /apps/fabmanager# cat /usr/local/sbin/renew.sh
#!/bin/bash
certbot renew --dry-run -n > /tmp/renewornot 2>&1
if grep -Fxq 'Cert not due for renewal, but simulating renewal for dry run' "/tmp/renewornot"
then
    echo "Certificat non arrivé à échéance...pas d'action"
else
    echo "Renouvellement du certicat..."
    cd /apps/fabmanager/
    docker-compose down
    cd /apps/fabmanager/letsencrypt/etc/live/fan-manager.di.unistra.fr
    #certbot -n --renew-by-default --domain fan-manager.di.unistra.fr
    certbot renew --post-hook "systemctl stop nginx"
    cd /etc/letsencrypt/archive/fan-manager.di.unistra.fr
    ls -Art | tail -n 4 | sort > /tmp/letslist
    declare -a lets_array
    lets_array=(`cat "/tmp/letslist"`)
    cp -f /etc/letsencrypt/archive/fan-manager.di.unistra.fr/${lets_array[0]}
    /apps/fabmanager/letsencrypt/etc/archive/fan-manager.di.unistra.fr/cert198.pem
    cp -f /etc/letsencrypt/archive/fan-manager.di.unistra.fr/${lets_array[1]}
    /apps/fabmanager/letsencrypt/etc/archive/fan-manager.di.unistra.fr/chain198.pem
    cp -f /etc/letsencrypt/archive/fan-manager.di.unistra.fr/${lets_array[2]}
    /apps/fabmanager/letsencrypt/etc/archive/fan-manager.di.unistra.fr/fullchain198.pem
    cp -f /etc/letsencrypt/archive/fan-manager.di.unistra.fr/${lets_array[3]}
```

```
/apps/fabmanager/letsencrypt/etc/archive/fan-manager.di.unistra.fr/privkey198.pem
```

```
cd /apps/fabmanager/
```

```
killall nginx
```

```
docker-compose up -d
```

```
fi
```

```
exit
```

# Serveur IoT - MQTT, SQL, Node-RED, InfluxDB

On a besoin d'un broker (serveur) MQTT public (exposé sur internet via une IP publique et éventuellement un nom de domaine) pour pouvoir envoyer des données openSenseMap.

Ce serveur public peut recevoir et centraliser les données provenant de :

- stations LoraWan via le serveur Chirpstack Unistra/EMS
- d'un autre Raspberry (non public) configuré pour récupérer les données des stations via wifi ou filaire. Par exemple avec HomeAssistant

## Exposition du serveur IoT sur internet

fan.ynh.fr (Raspberry 3) sur le net via un VPN wireguard

<https://innovation.iha.unistra.fr/books/1-fablab-vos-projets/page/installer-un-serveur-avec-la-distribution-linux-yunohost>

## Installation des services publics

- Création des domaines et d'un certificat let'sEncrypt
- Installer sur le serveur fan.ynh.fr les applications YunoHost :
  - mosquitto\_ynh
  - phpmyadmin
  - nodered
  - influxdb\_v2