Gateway Meshtastic eJebberd – Mise en marche

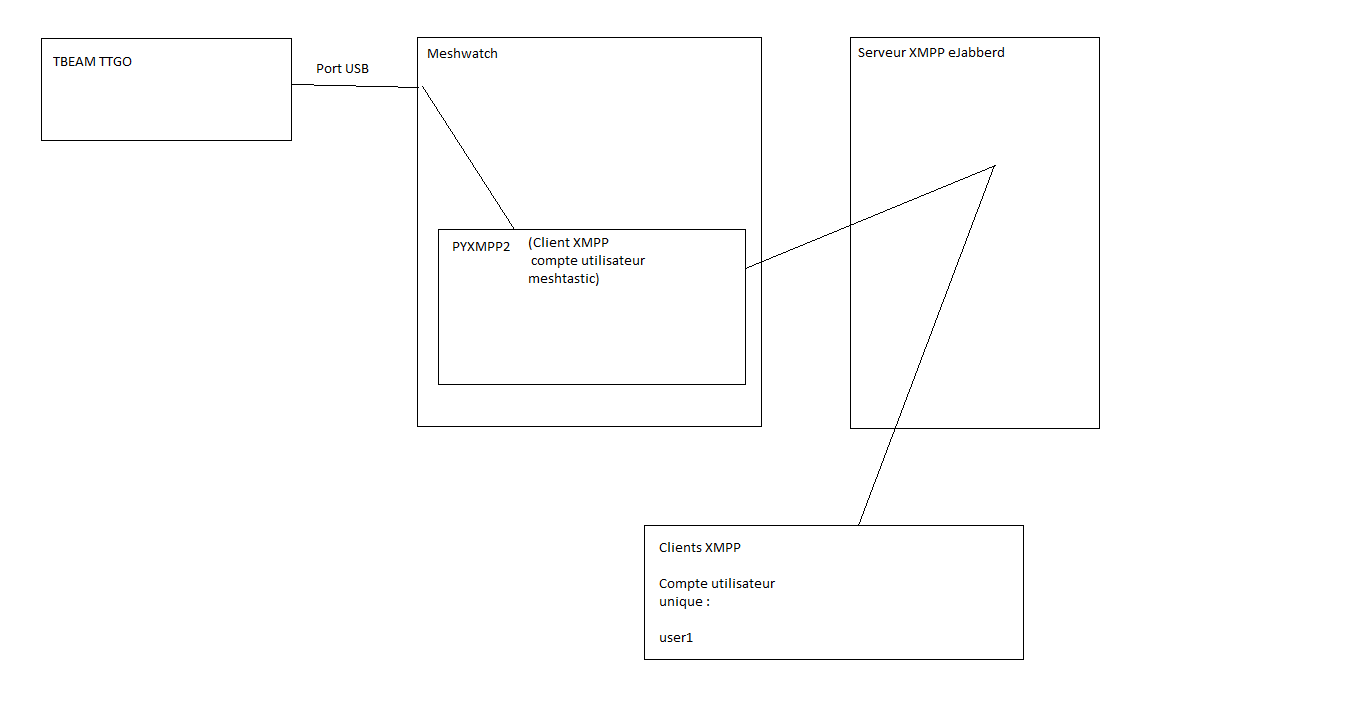
Adapté par ON3MEE

**Introduction**

Voici le projet d’un gateway Meshtastic eJabberd. Le gateway permet de se connecter à un réseau Meshtastic via un client de messagerie de tchat instantané XMPP sur un LAN. Cela permet de pouvoir se connecter à plusieurs stations (PC, smartphone, …) sur un seul nœud Meshtastic à une interface de tchat web ou via des applications de tchat clientes XMPP. Ces applications clientes ressemblent à un tchat Messenger et sont compatible Windows, Android, IOS, …

L’objectif initial du projet est de pouvoir mettre quelques PC (ou smartphones) sur le même nœud meshtastic et afficher un seul tchat pour un poste de commandement de services de secours. Un nœud Meshtastic n’accepte qu’un seul utilisateur !

Coté conception, je me suis basé sur la distribution Freedom Box pour le serveur XMPP et son client web et sur le projet Meshwatch (<https://github.com/datagod/meshwatch>) qui fait l’interface avec le nœud Meshtastic. J’ai adapté le code source de Meshwatch pour y intégrer le client PYXMPP2 (<https://github.com/Jajcus/pyxmpp2> ).



**AVERTISSEMENT !!!**

Je vous préviens que je ne suis pas un codeur et que je déteste PYTHON. Epargnez-moi des remarques du genre : « Vous codez mal, … » . Je suis très conscient que le projet ici n’est pas du tout bien pensé et que c’est du bricolage. Mais ça fonctionne.

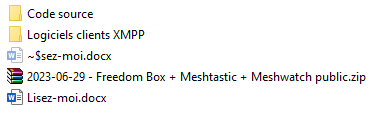
Je vous mets le code à disposition. N’hésitez pas à l’améliorer et à même reprendre le projet. Je suis 100 % ouvert à toute aide comme le veut le monde du logiciel libre.

**Remerciements**

Je remercie ON3CED (<http://www.on3ced.com/>) pour leur aide apportée pour les différents essais du réseau Meshtastic et du gateway Freedom Box.

On peut aussi remercier [datagod](https://github.com/datagod/meshwatch/commits?author=datagod) (<https://github.com/datagod/meshwatch/commits?author=datagod>) qui a créé l’outil Meshtalk qui a permis le développement de ce projet.

**Contenu du package**

****

* *Code source*

Ce sont les listings des programmes modifiés et utilisé.

* *Logiciels clients XMPP*

Les installs de différentes applications clientes XMPP pour Windows, Linux et Android.

* *2023-06-09 … .zip*

C’est l’image Freedom Box modifiée à installer sur la carte SD.

**Schéma de câblage**

Internet



Console via SSH

LAN

Ecran HDMI

Câble USB

Module Meshtastic

LAN

Freedom box Meshtastic

Raspberry Pi

WAN

ROUTEUR

WIFI + DHCP

**Prérequis**

Il faut câbler en respectant le schéma de la page précédente. On peut y ajouter des points d’accès WIFI, etc …

Concernant le routeur, il est indispensable de configurer le nom de domaine correctement : localdomain.

Les comptes XMPP n’accepte que des adresses de ce genre : user@machinex.localdomain . Pas d’adresse IP !!! [user@192.168.1.1](mailto:user@192.168.1.1) ne fonctionnera pas !!! Si la résolution du domaine ne fonctionne pas, le système ne pourra pas fonctionner !

On règle ces paramètres dans le web CLI du routeur.

Dans ce projet, le nom de domaine est localdomain. Le nom de machine est meshtastic. Les comptes utilisateurs sont user1 ; user2 ; … Cela donne une adresse du style :

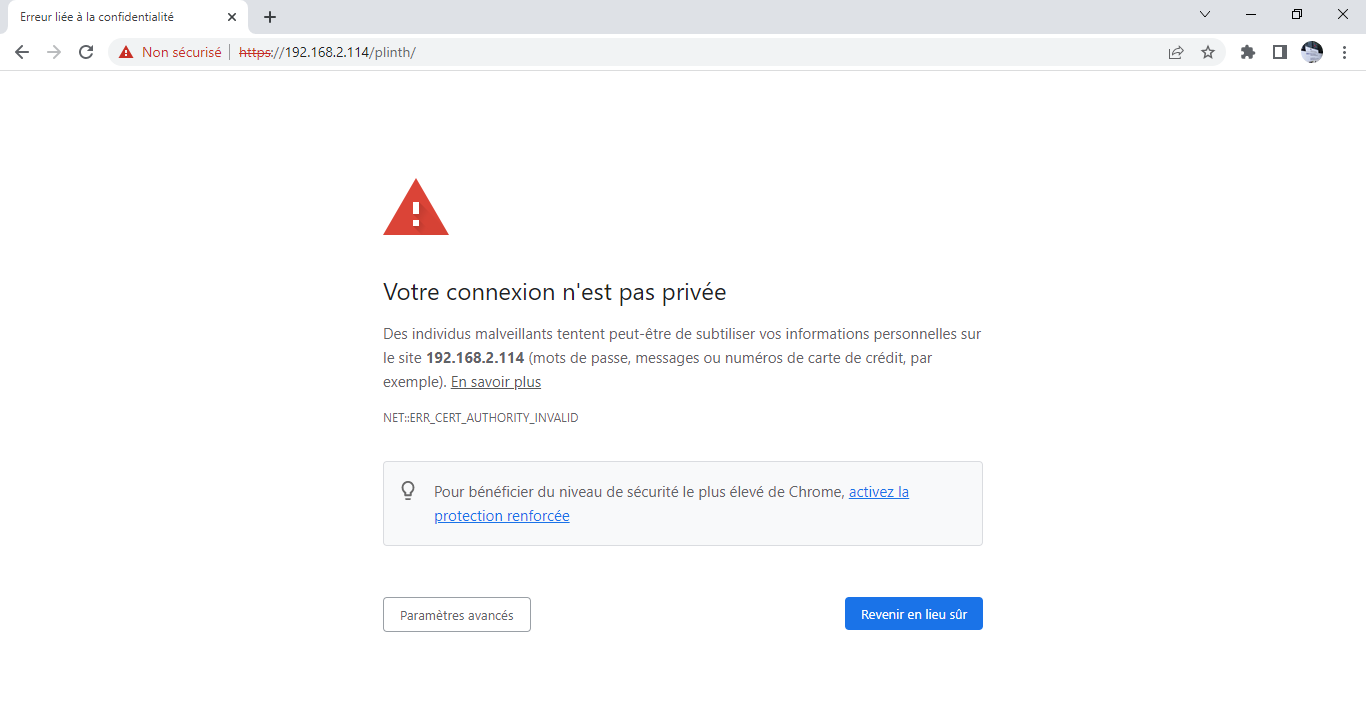
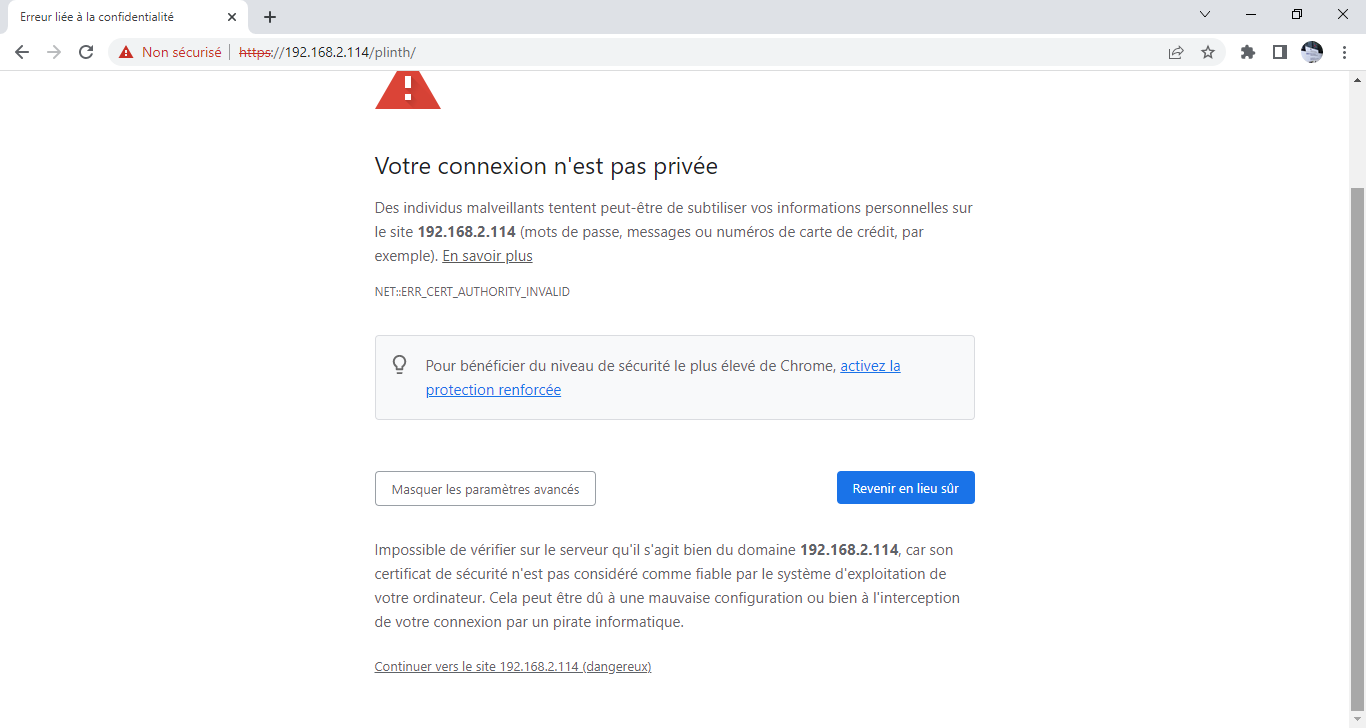
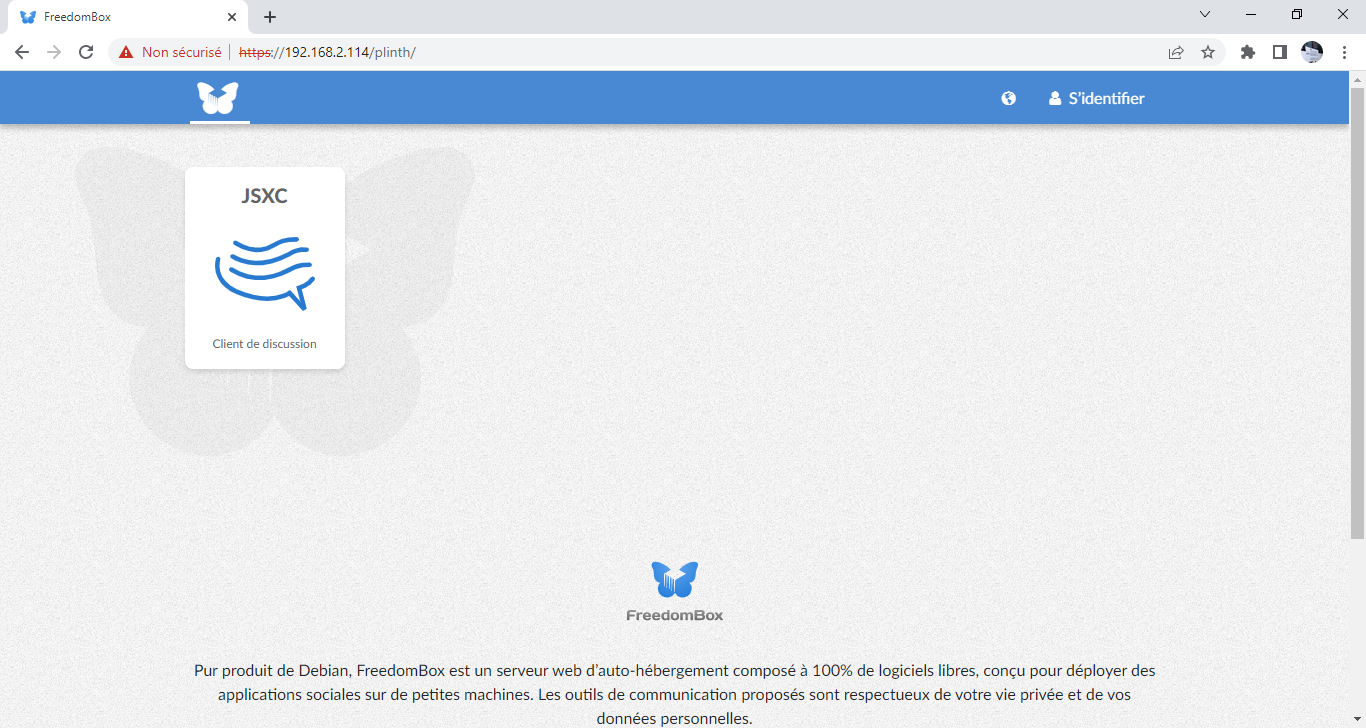
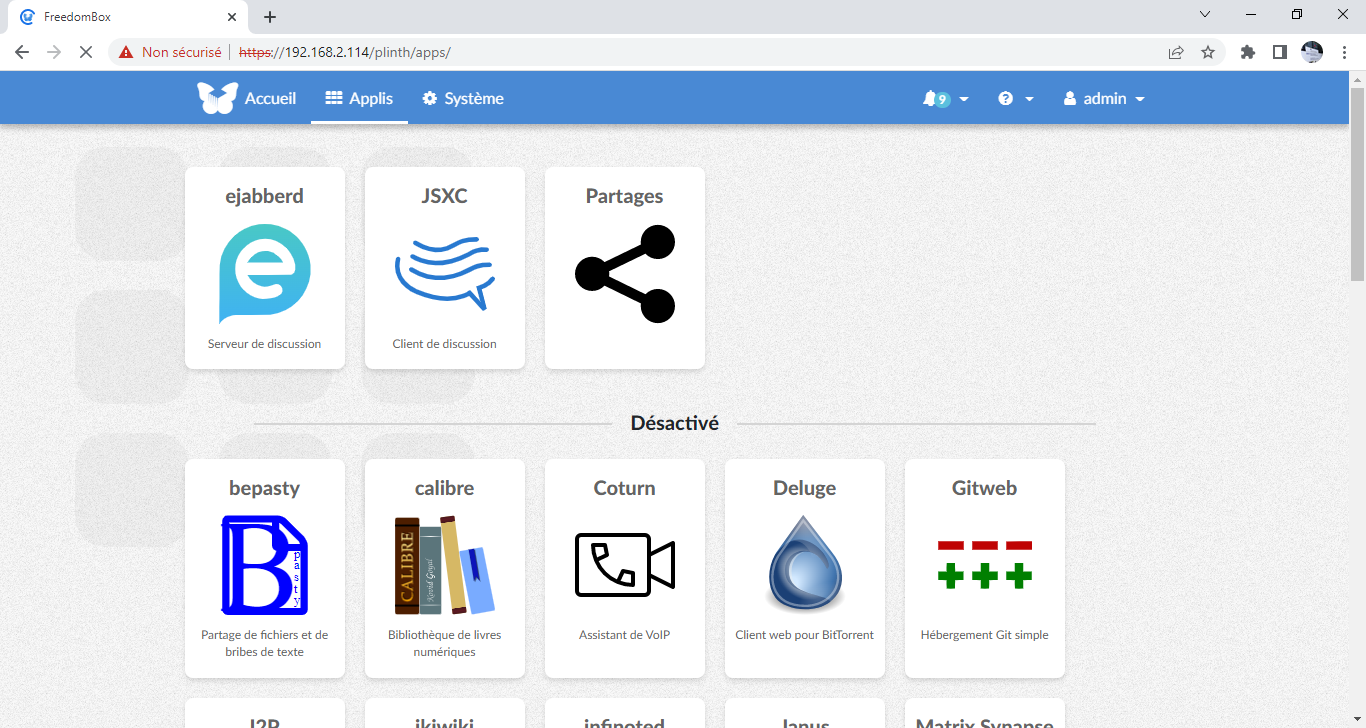
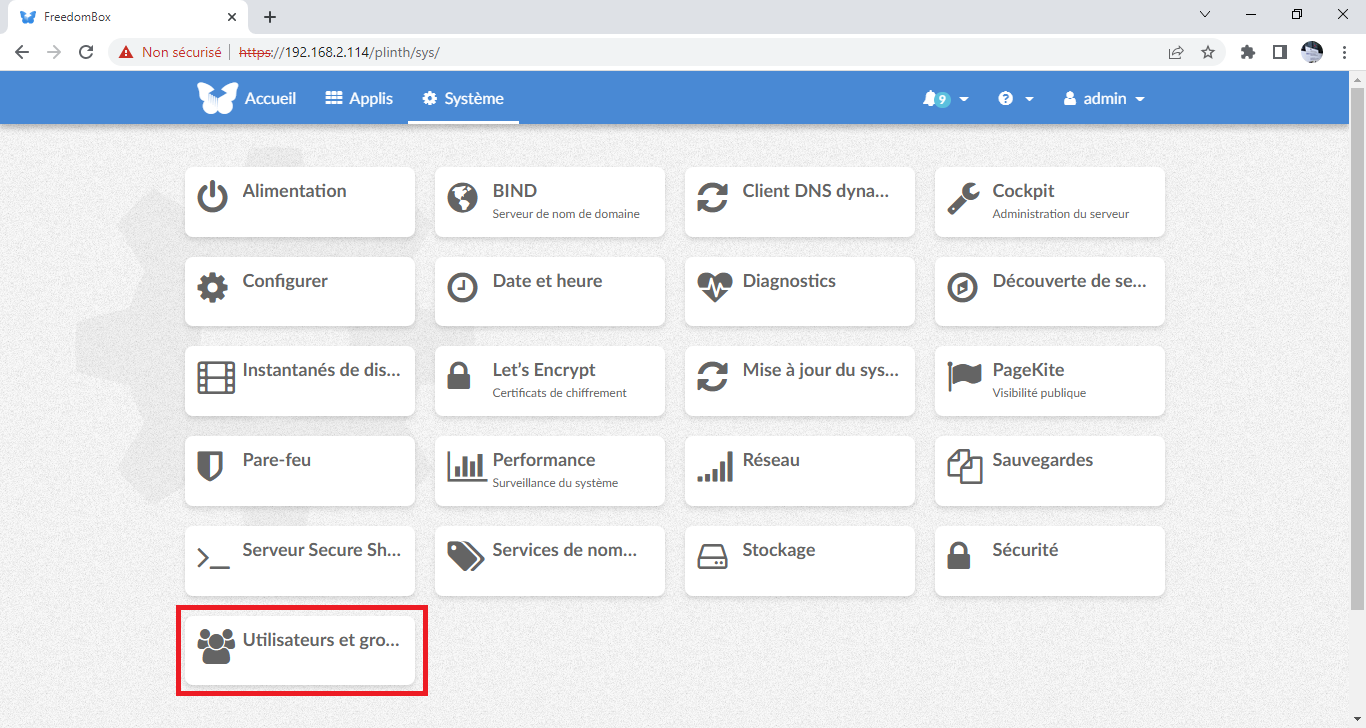
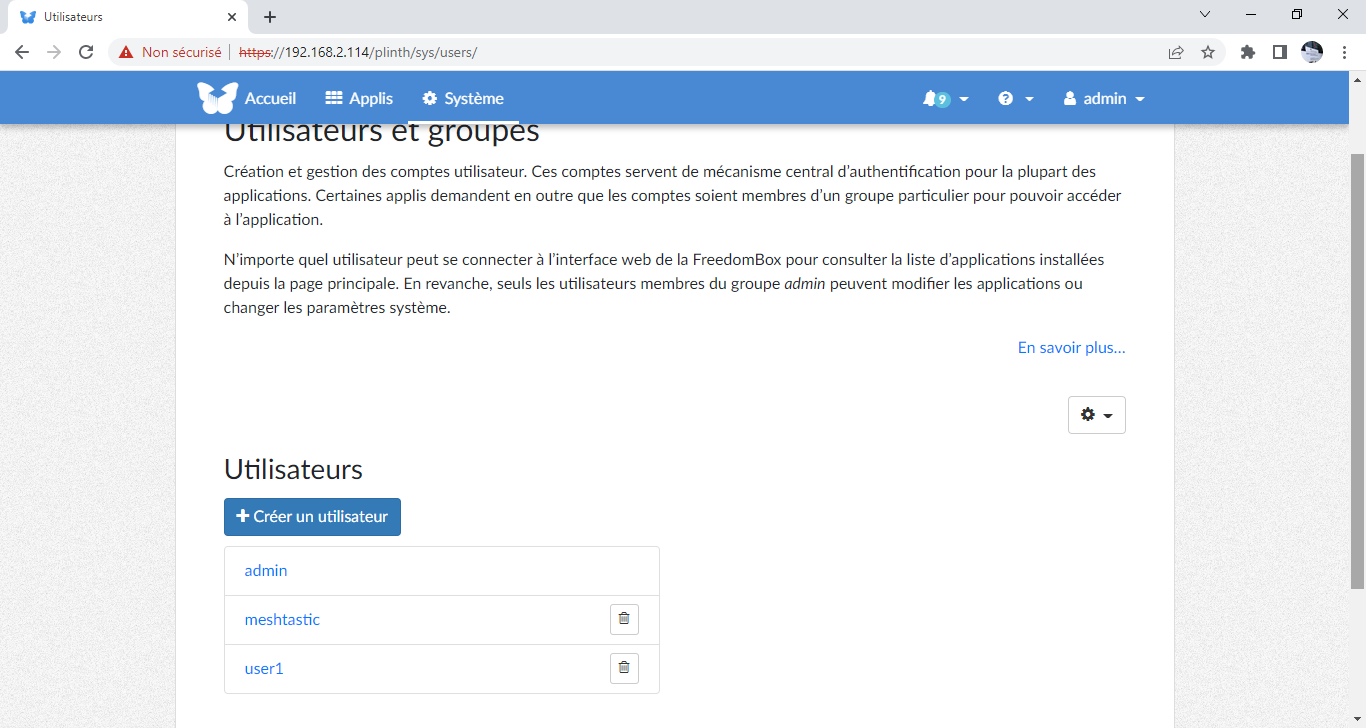
[user1@meshtastic.localdomain](mailto:user1@meshtastic.localdomain).

Dans certains cas, il faudra une adresse du style : user1@meshtastic et non user1@meshtastic.localdomain. Cela dépendra de votre routeur. Il faudra donc faire des tests pour déterminer le bon suffixe. Dans ce cas, il faudra modifier les adresses dans le code source de la freedom box (voir annexe de ce document).

Concernant les adresses IP, c’est à vous de décider. Il n’y a pas d’IP à respecter. Le rpi est configuré en IP dynamique. Il est cependant conseillé de fixer l’adresse IP de la freedom box.

Concernant le WIFI, c’est aussi à vous de décider.

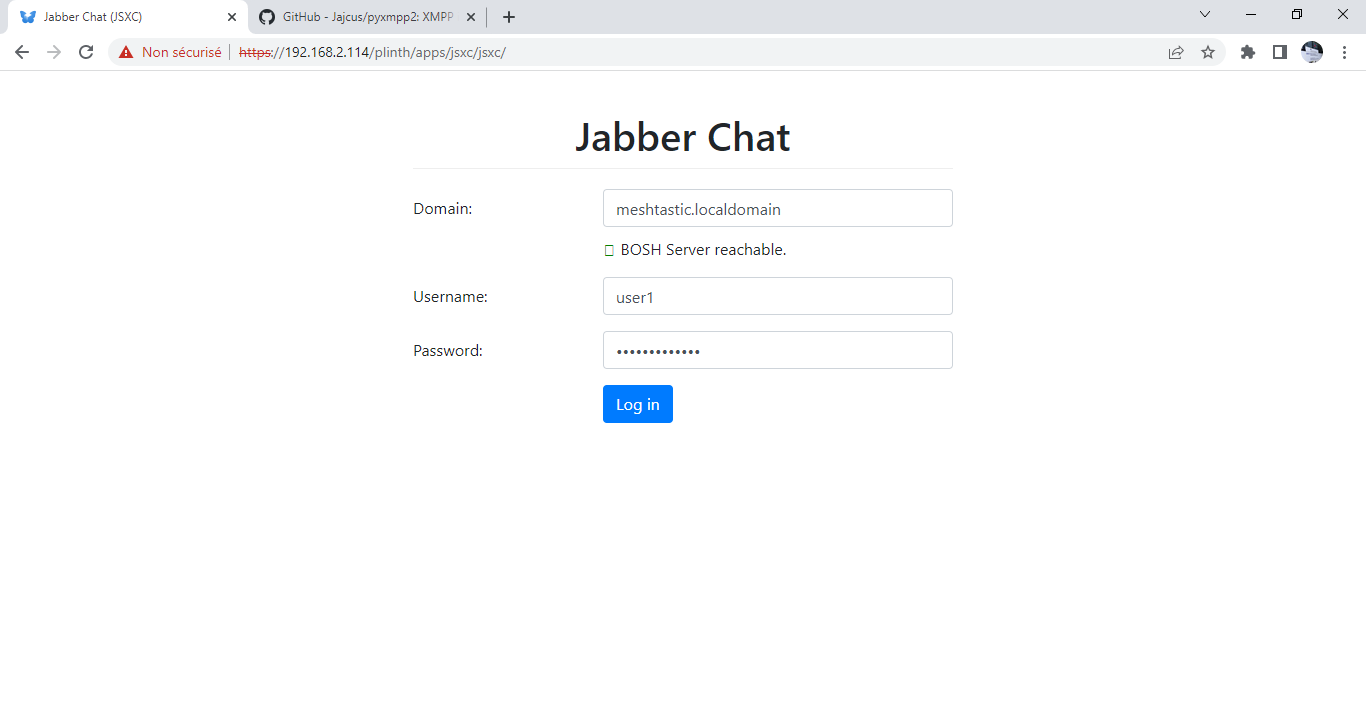
**Freedom Box et le serveur XMPP eJabberd**

1. Mettre l’image « … .img » sur une carte SD de 16 GB dans un rpi 3 au minimum.
2. Connecter sur l’adresse IP sur rpi via un navigateur web :  
     
     
   
3. Cliquer sur « Continuer vers le site xxx (dangereux). Ce n’est pas dangereux. Le certificats est autosigné !
4. Le web CLI de freedom box apparait :  
     
     
     
   Login : admin – Mot de passe : adminadmin
5. Les applications clientes sont ici :  
     
     
     
   Celle qui nous intéresse est eJabberd pour le serveur de messagerie XMPP et JSXC pour le client web de messagerie.  
     
   Laisser les paramètres configurés.
6. Les comptes utilisateurs se règle ici :  
     
     
     
     
     
   Les comptes déjà créés ne doivent pas être changés. Ils sont liés au programme meshwatch modifié. Libre à vous de changer, mais il faudra adapter le code source en conséquence !

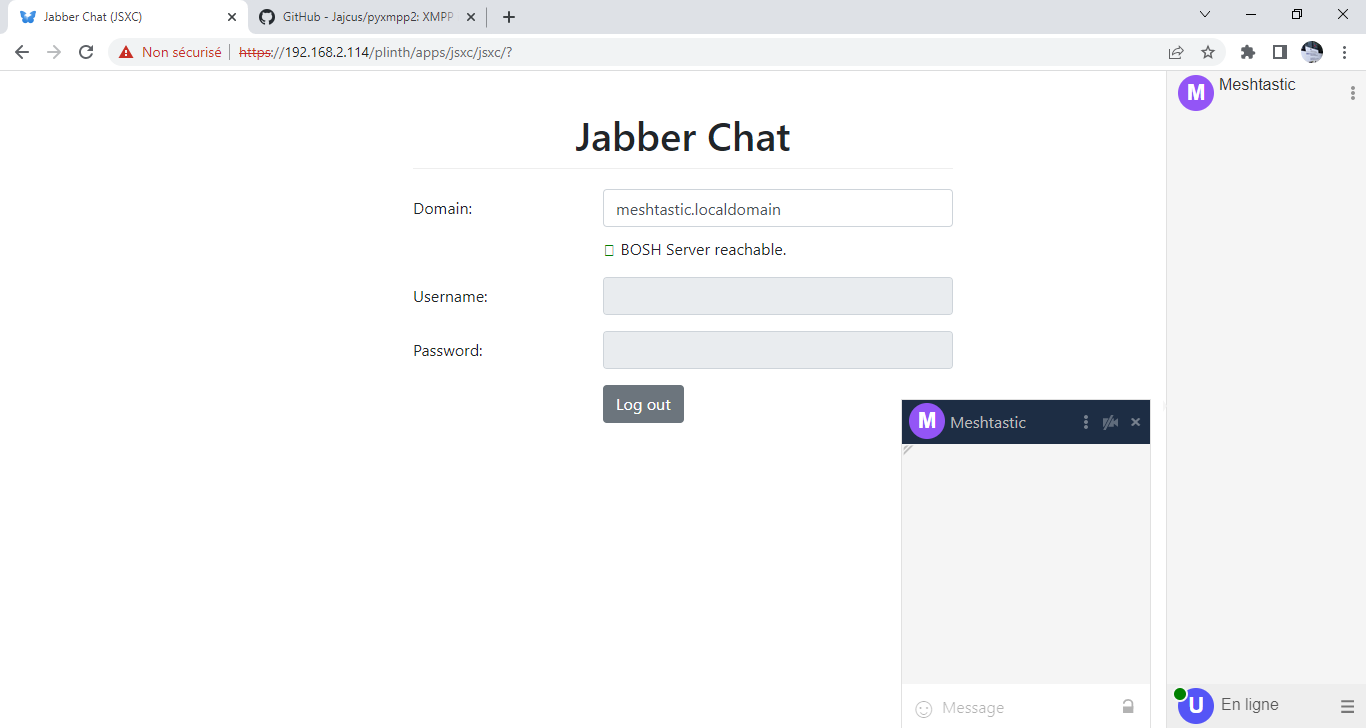
Voici les credentials :  
  
user : meshtastic / password : 2022user12022 -> C’est le compte qui fait la passerelle entre Meshwatch et eJabberd

User : user1 / password : 2022user12022 -> C’est le compte client avec lequel vous allez vous connecter sur le client JSXC.

1. Voici le client JSXC



L’adresse XMPP est du genre [user1@meshtastic.localdomain](mailto:user1@meshtastic.localdomain) . Il est TRES IMPORTANT de respecter la syntaxe [user@meshtastic.localdomain](mailto:user@meshtastic.localdomain) . Une IP à la place du domaine ne marche pas. Il faudra bien programmer votre routeur avec le suffixe [userx@meshtactic.localdomain](mailto:userx@meshtactic.localdomain) ou userx@meshtastic.

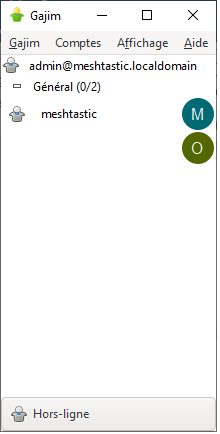
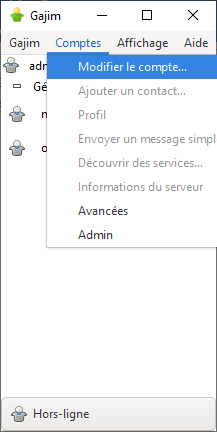


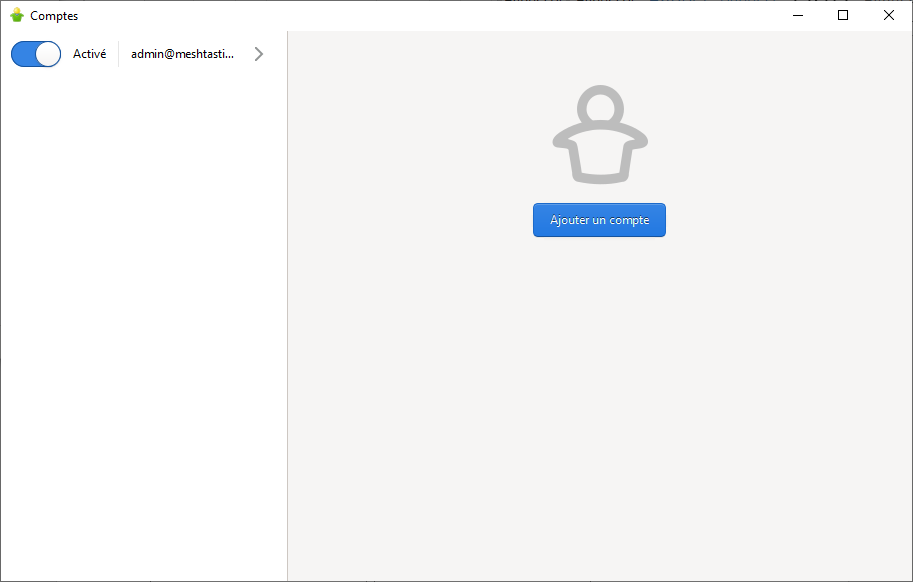
Le contact Meshtastic est le gateway.

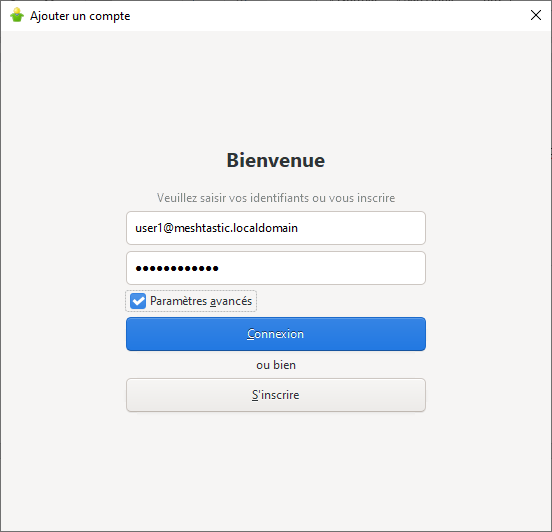
1. Gajim

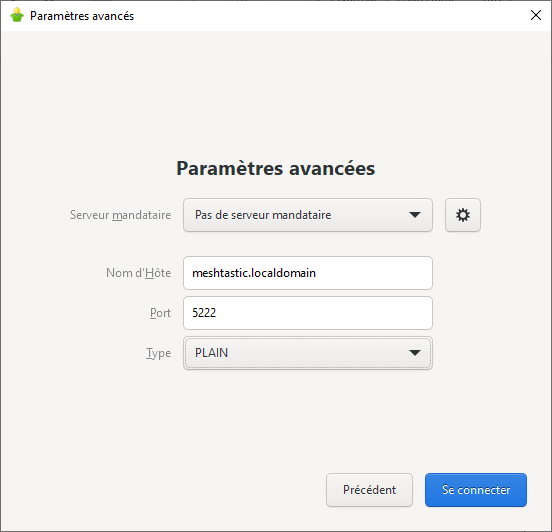
Gajim (<https://gajim.org/>) est une application XMPP cliente qui tourne sur Windows et Linux.

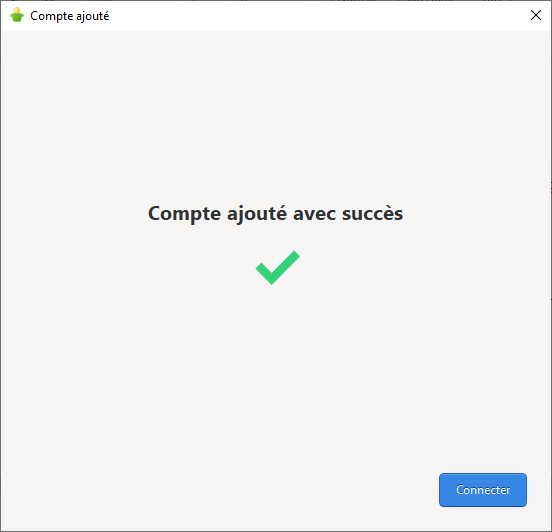
Voici comment on la configure :

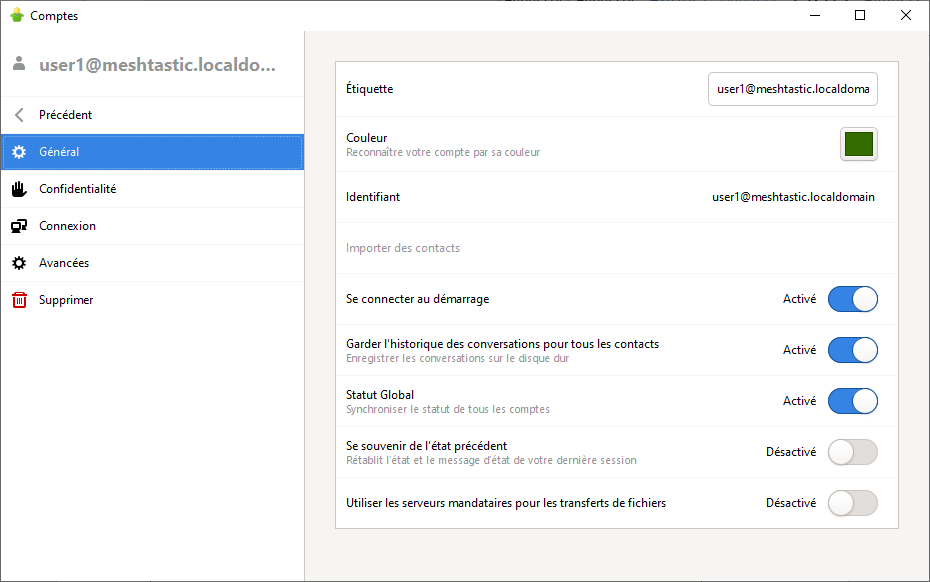
 



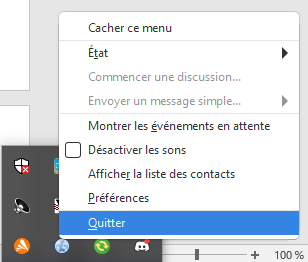




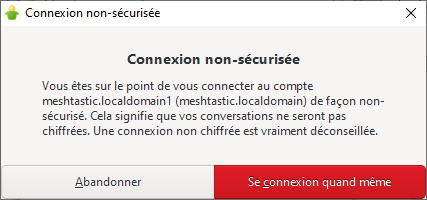




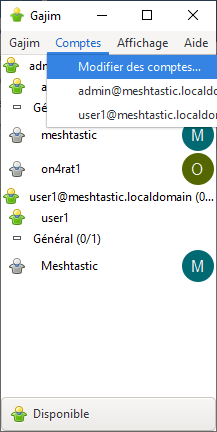
Fermer Gajim :

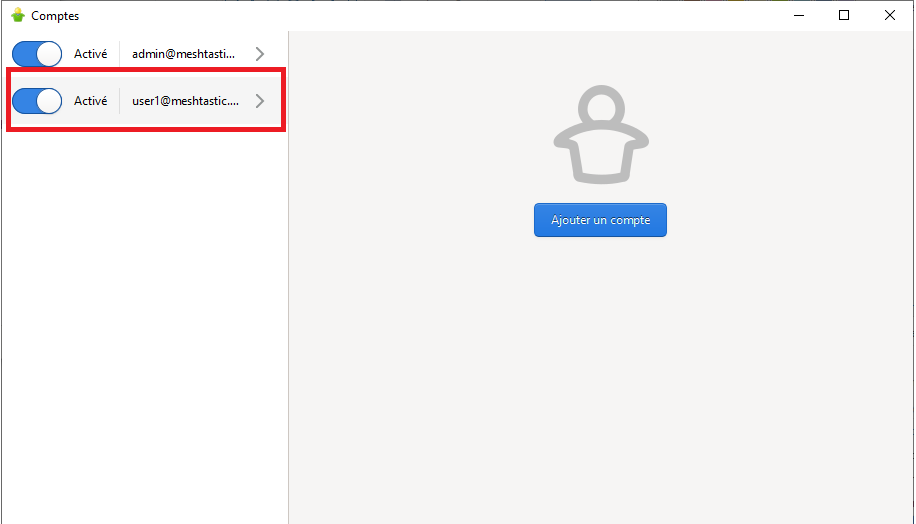


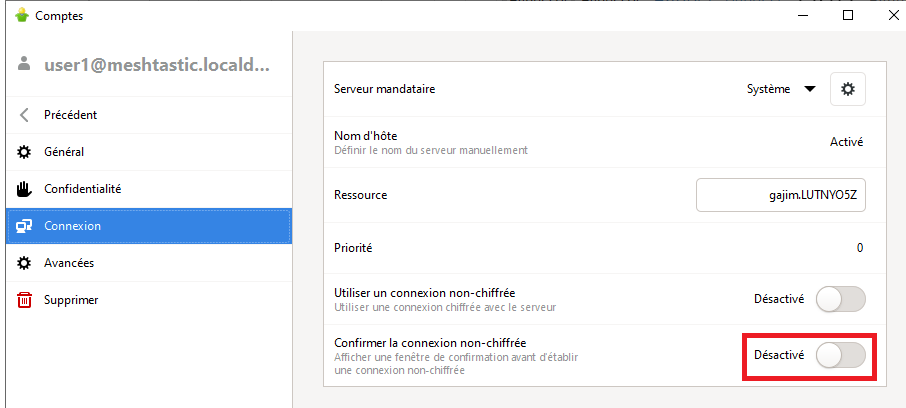
Rouvrir Gajim.

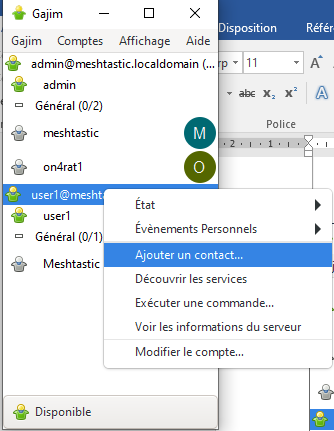


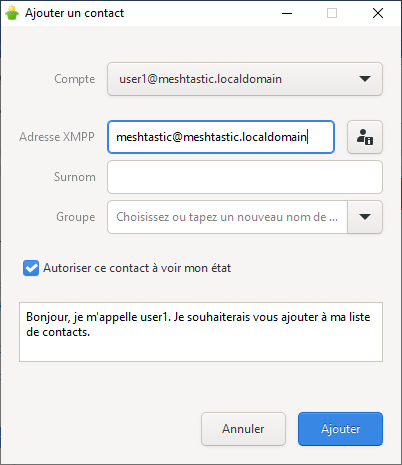
Se connecter quand même.





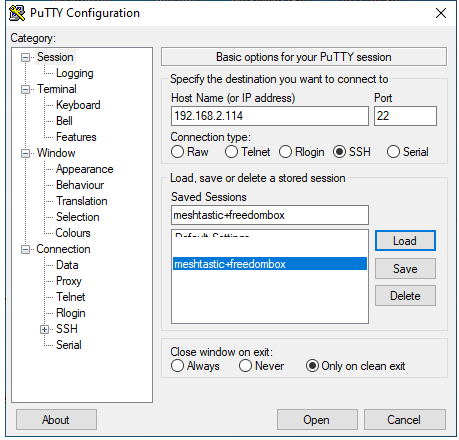






**Meshwatch**

1. Se connecter par SSH sur le gateway



Login : root / password : admin

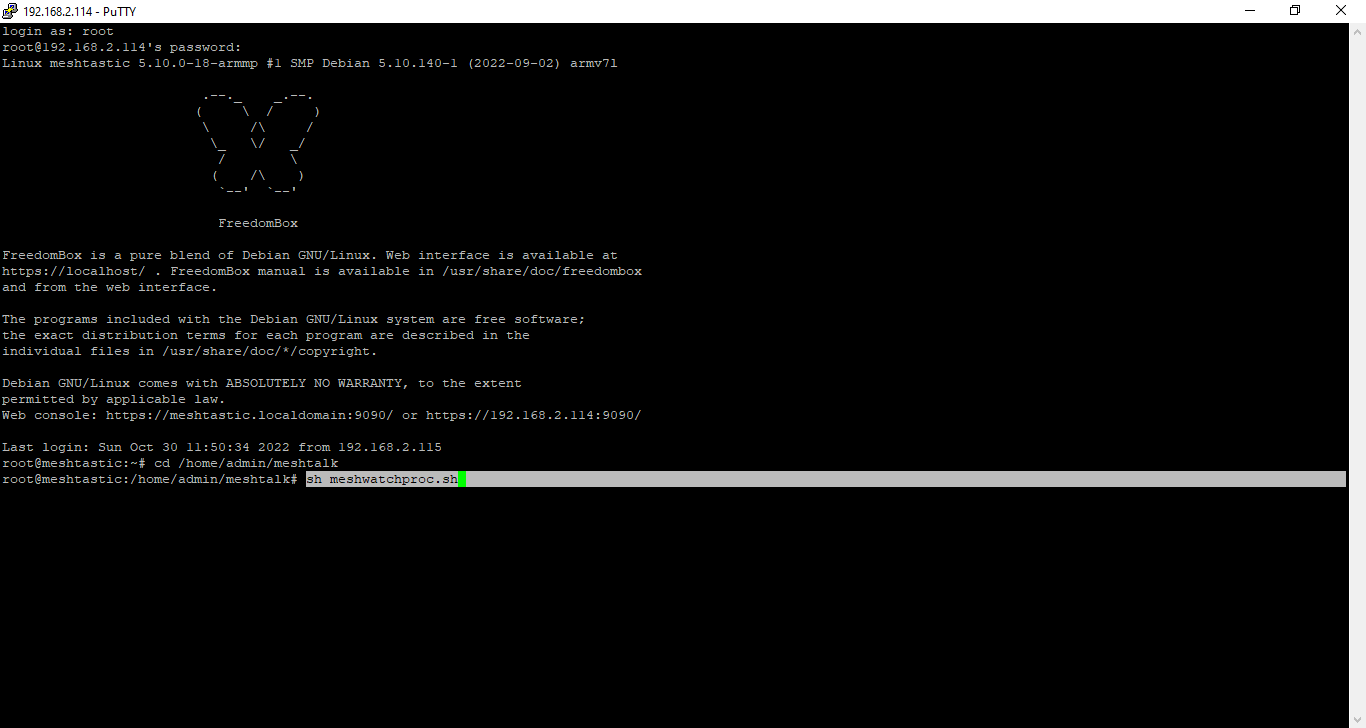


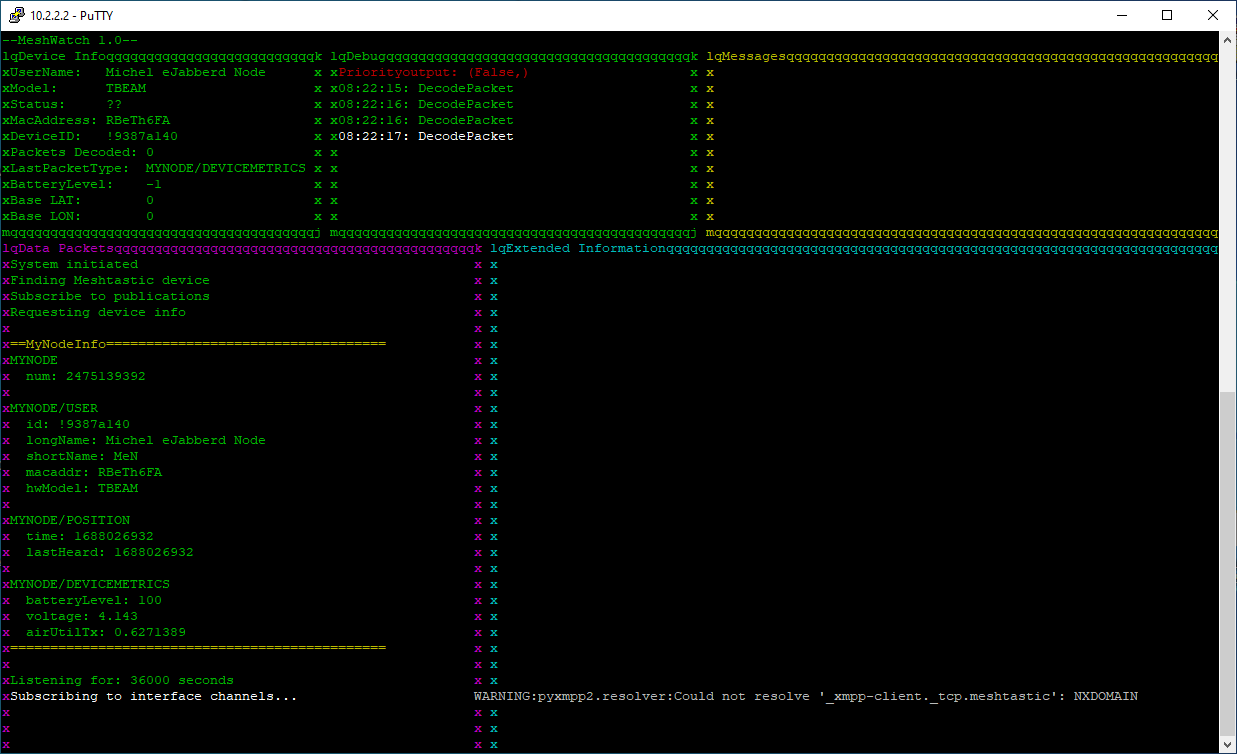
1. Mettre en plein écran.
2. Brancher le node Meshtastic sur le port USB (TTGO par exemple).
3. Aller dans le répertoire :

/home/admin/meshtalk

Commande : *cd /home/admin/meshtalk*

Pour lancer le script gateway commande*: sh meshwatchproc.sh*





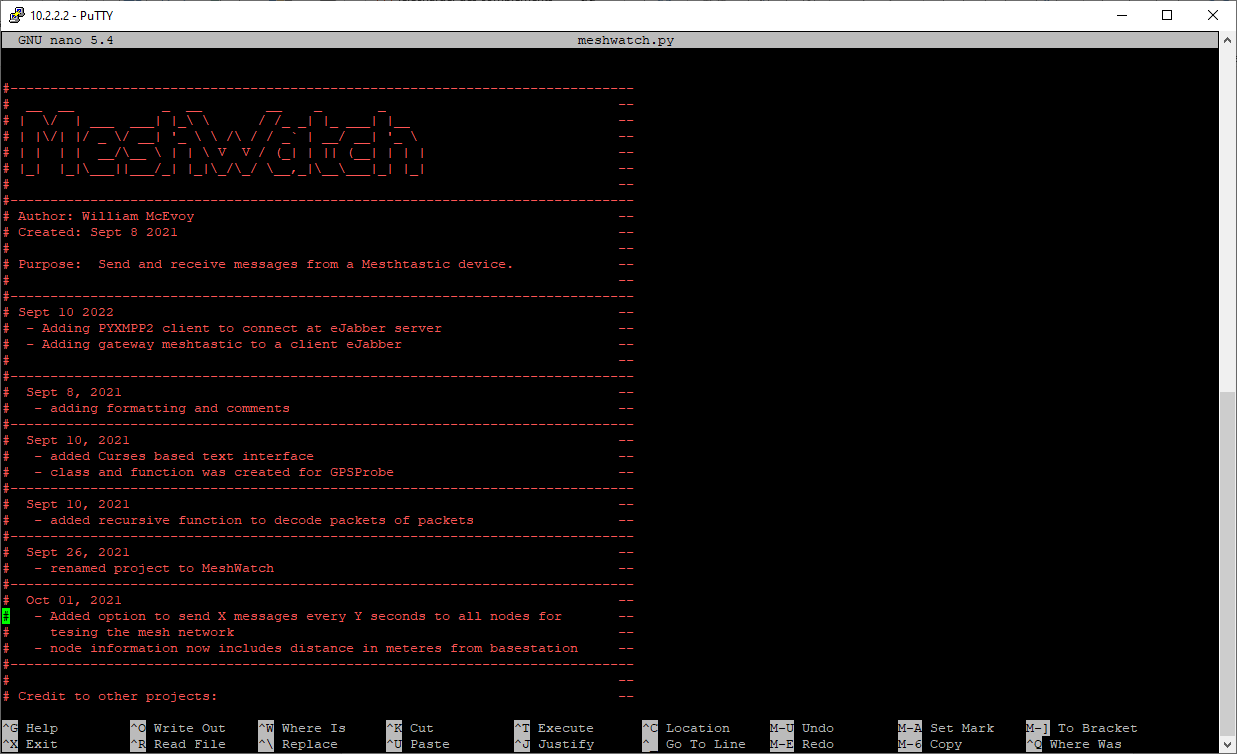
Tout tourne.

**Attention, Meshwatch ne tourne que s’il est affiché sur un écran ! Cela peut être une console SSH ou un écran HDMI.**

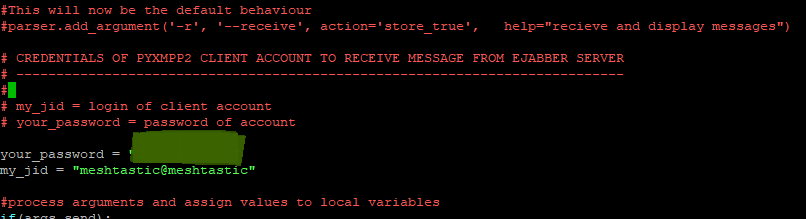
1. **Pour changer les adresses des comptes users dans meshwatch si localdomain ne fonctionne pas :**

Dans le répertoire « /home/admin/meshtalk », taper la commande :

nano meshwatch.py



Modifier le suffixe à la variable « my\_jid » :



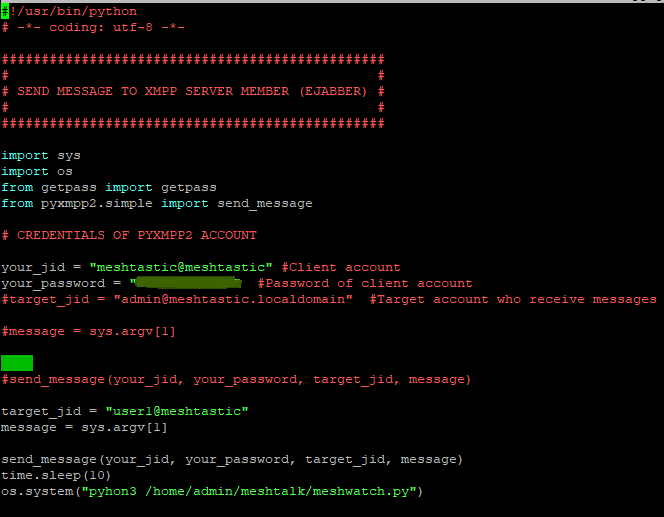
Sauver : CTRL + O.

Quitter : CTRL + X.

Ensuite, taper la commande :

nano sendxmpp.py

Modifier « your\_jid » et « target\_jid » :

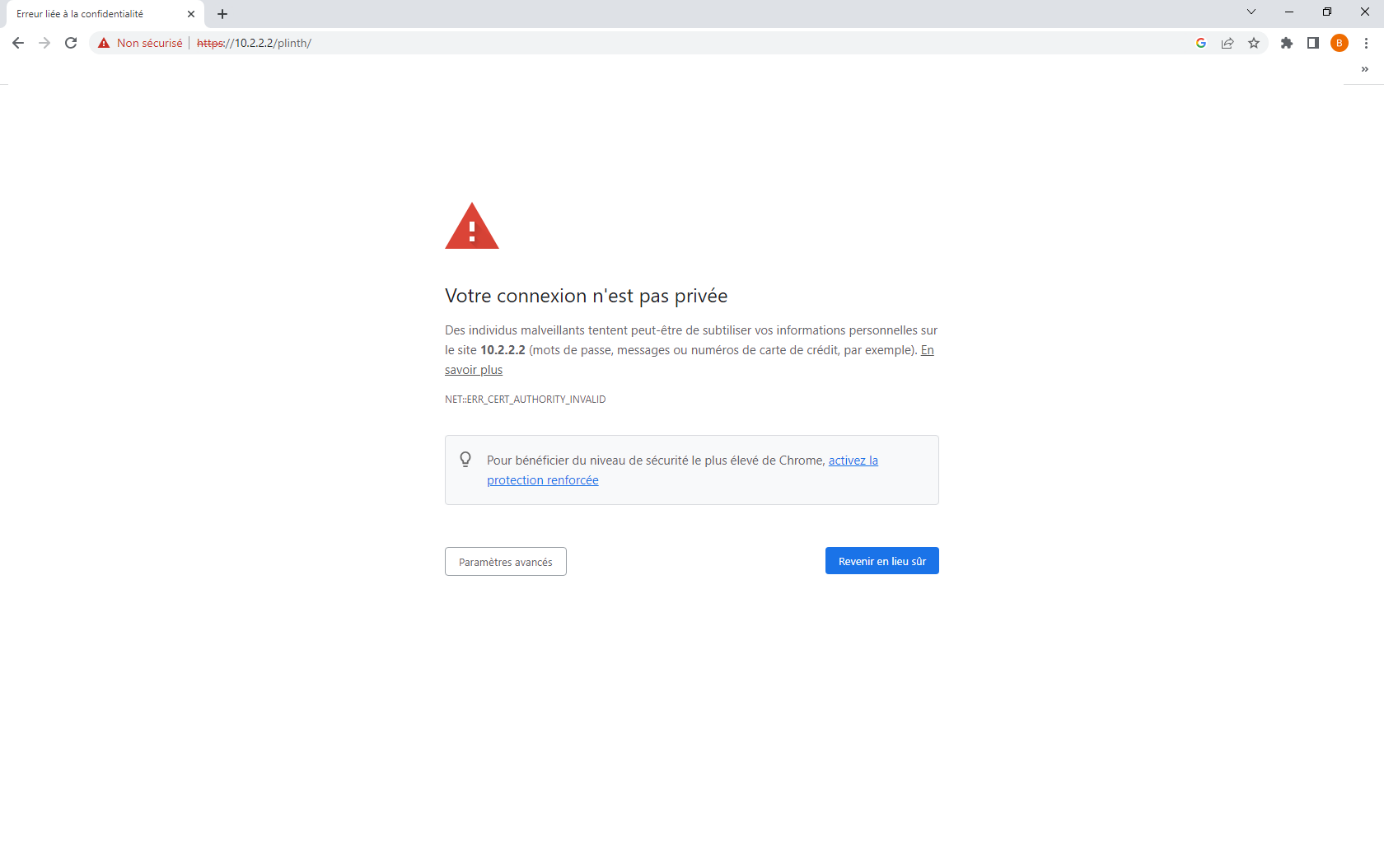


Sauver : CTRL + O.

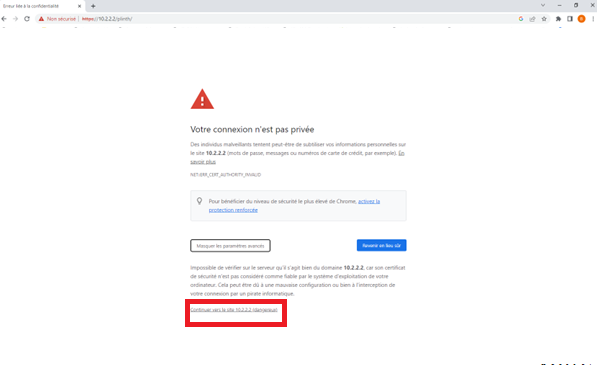
Quitter : CTRL + X.

**Utiliser le client web JSXC**

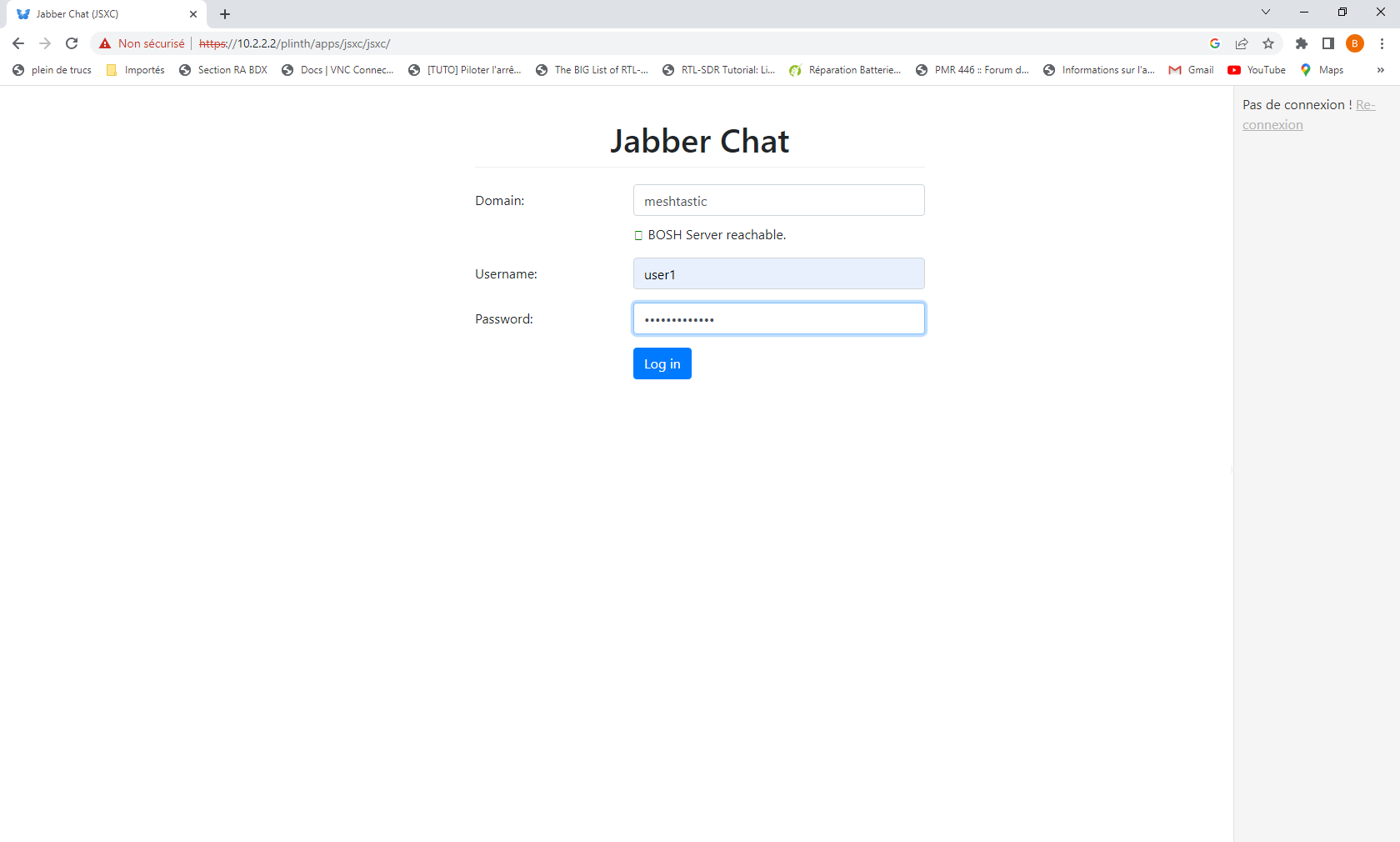
1. Entrer l’adresse de la freedom box



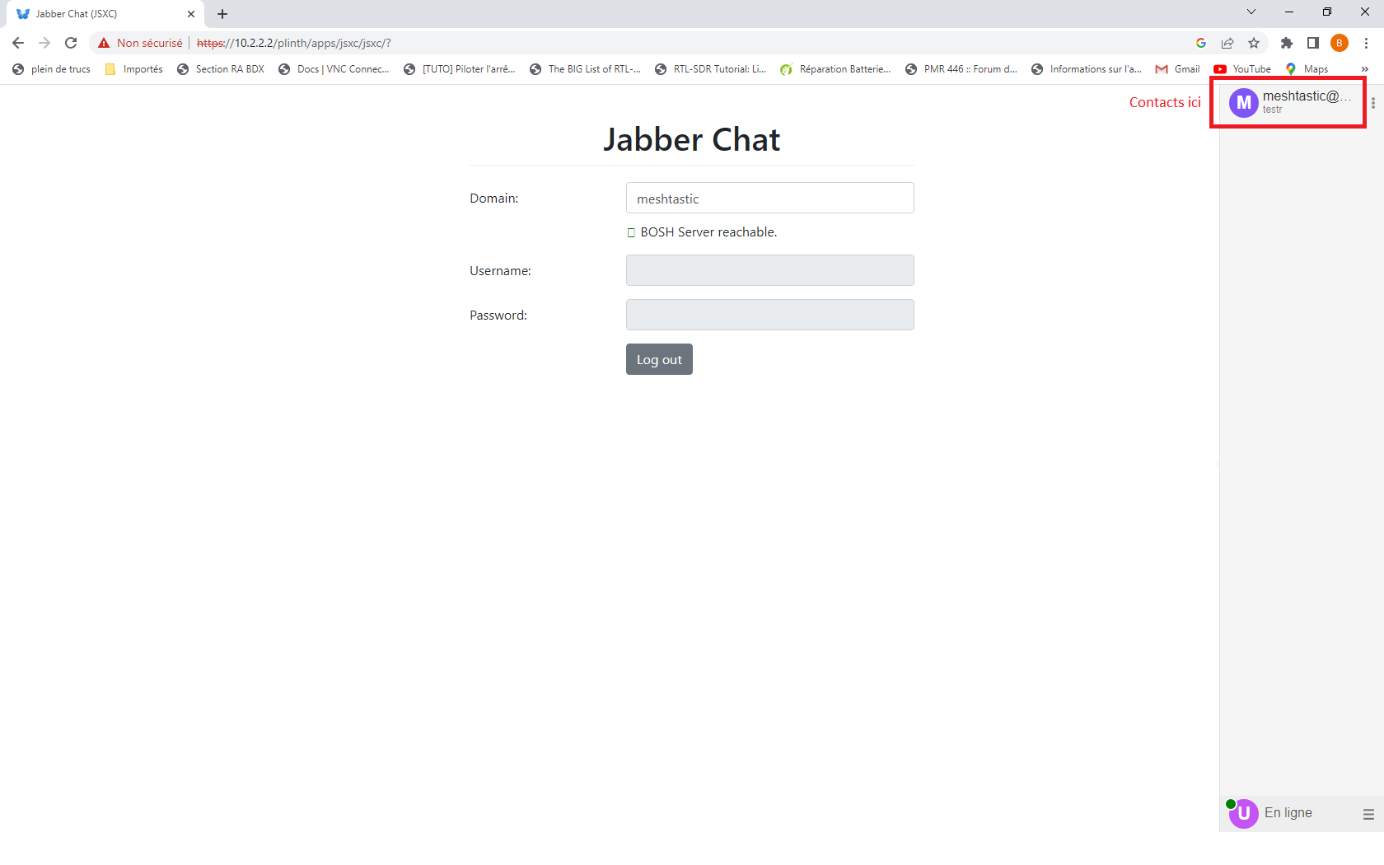
1. Cliquer sur « Continuer sur le site » :



Cliquer sur JSXC :



Se loguer.

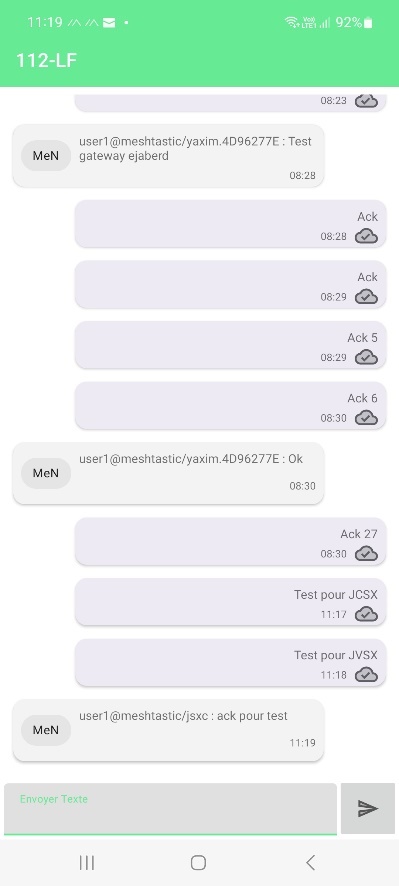


Voici la fenêtre de tchat :

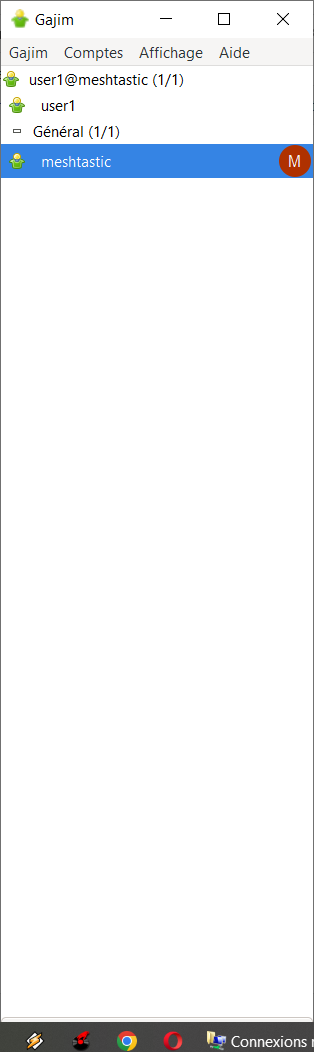


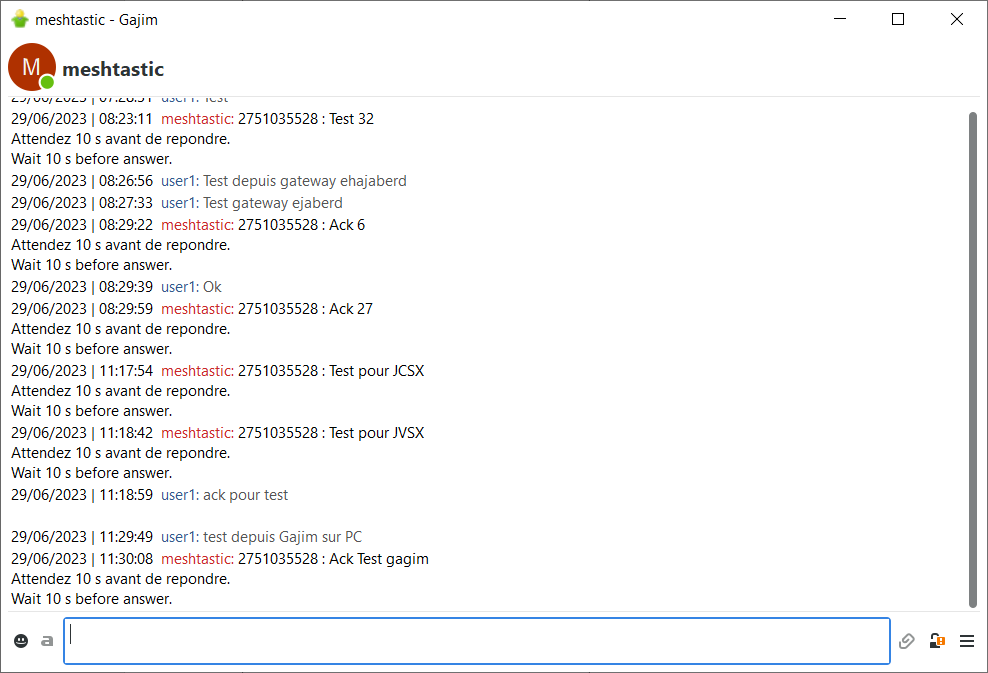
Comme il est bien indiqué, il faut attendre 10 secondes avant de répondre !!!

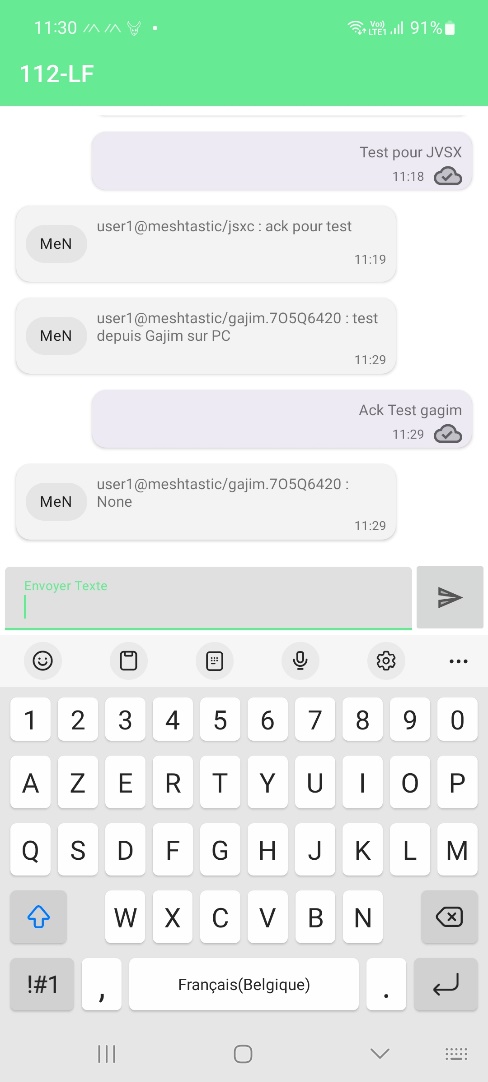
Message reçu dans Meshtastic :



**Utiliser le client Gajim sur PC**





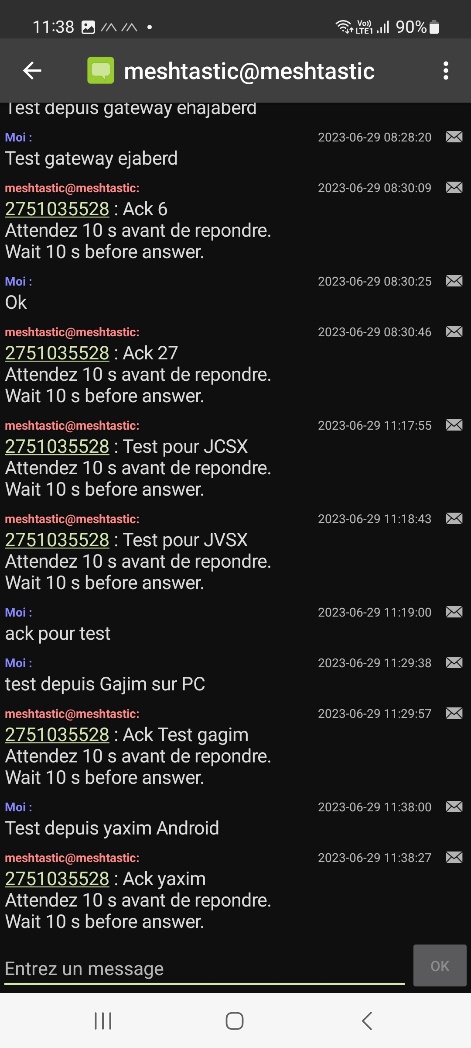
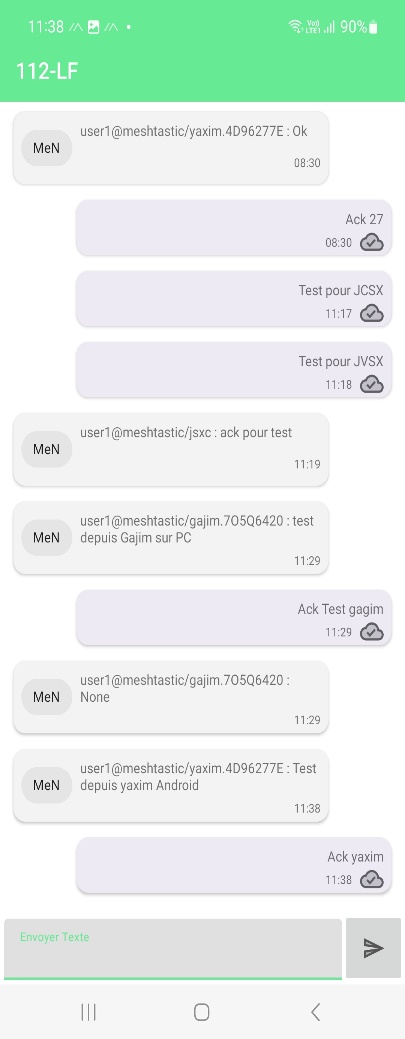


**Utiliser le client Android yaxim**

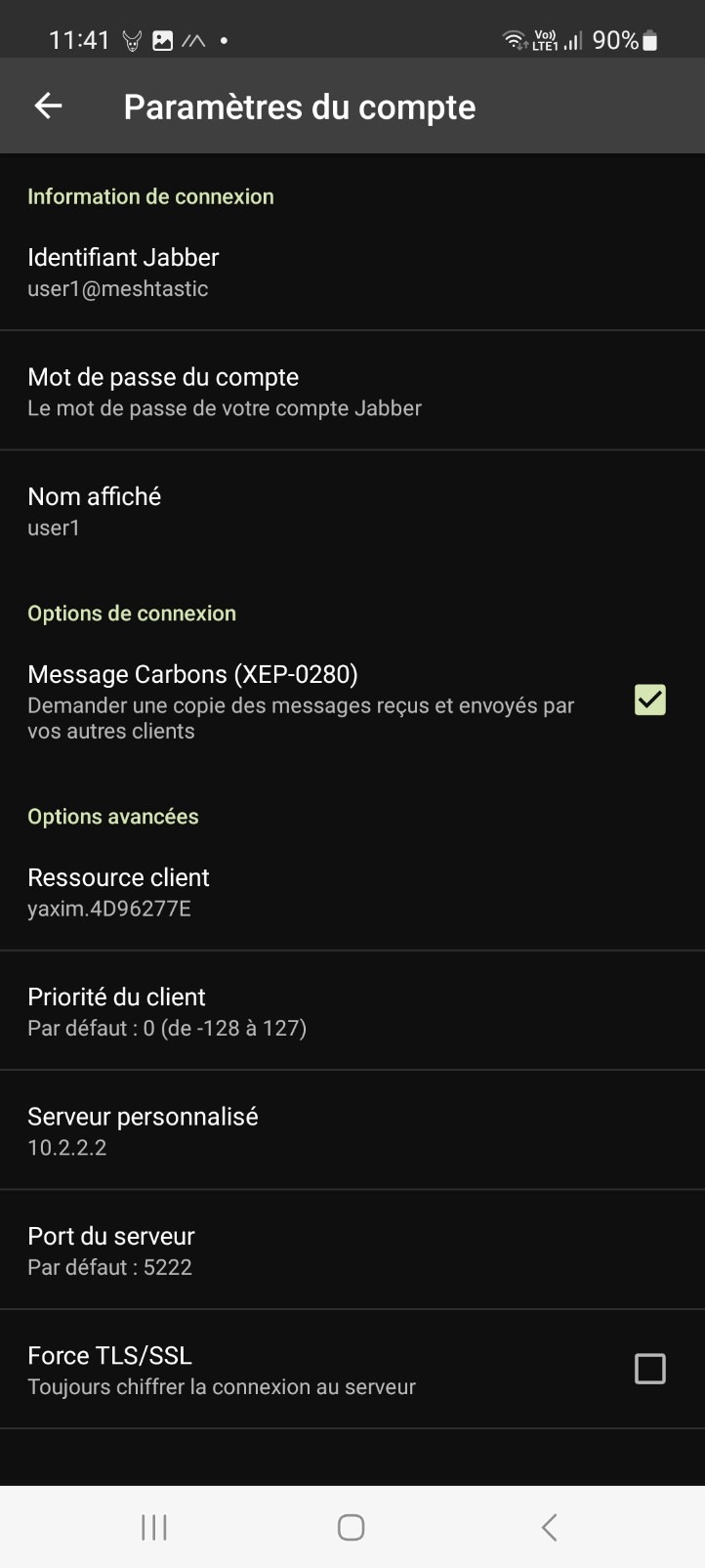
Client android à télécharger sur <https://yaxim.org/> .

Se connecter au WIFI de la Freedom Box.



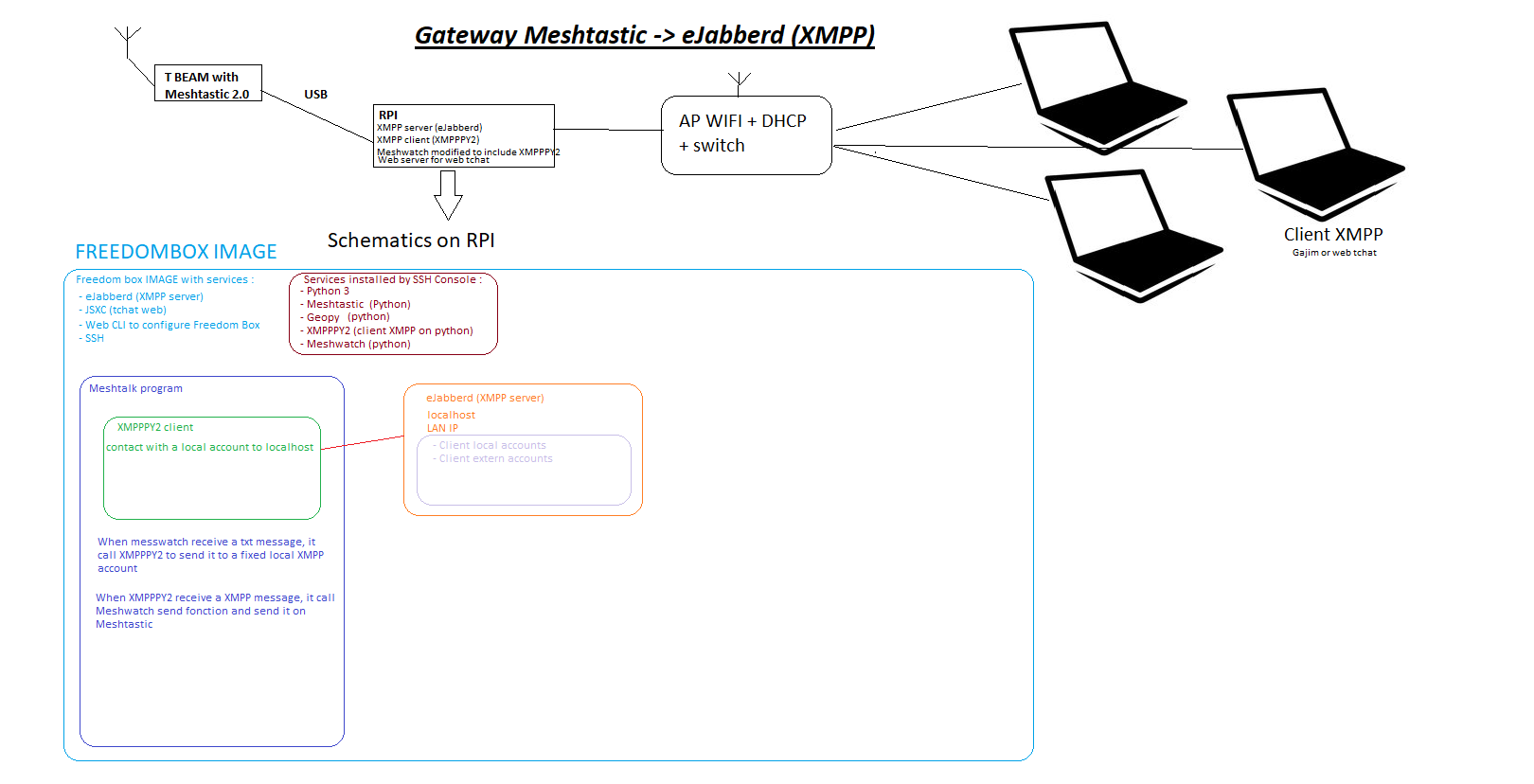
  

Paramètres du client :

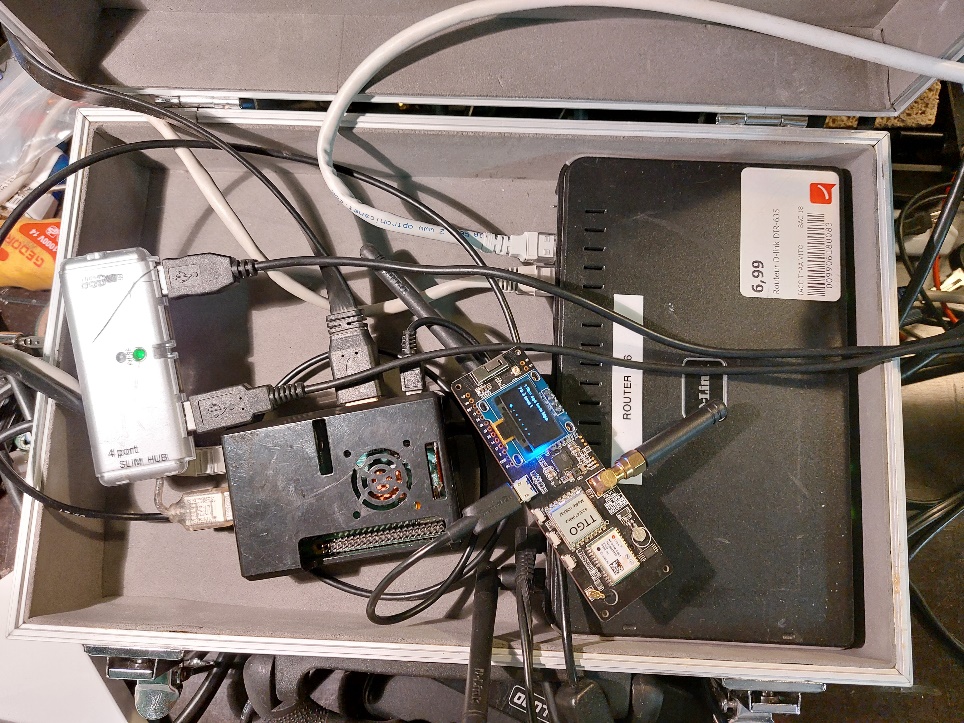
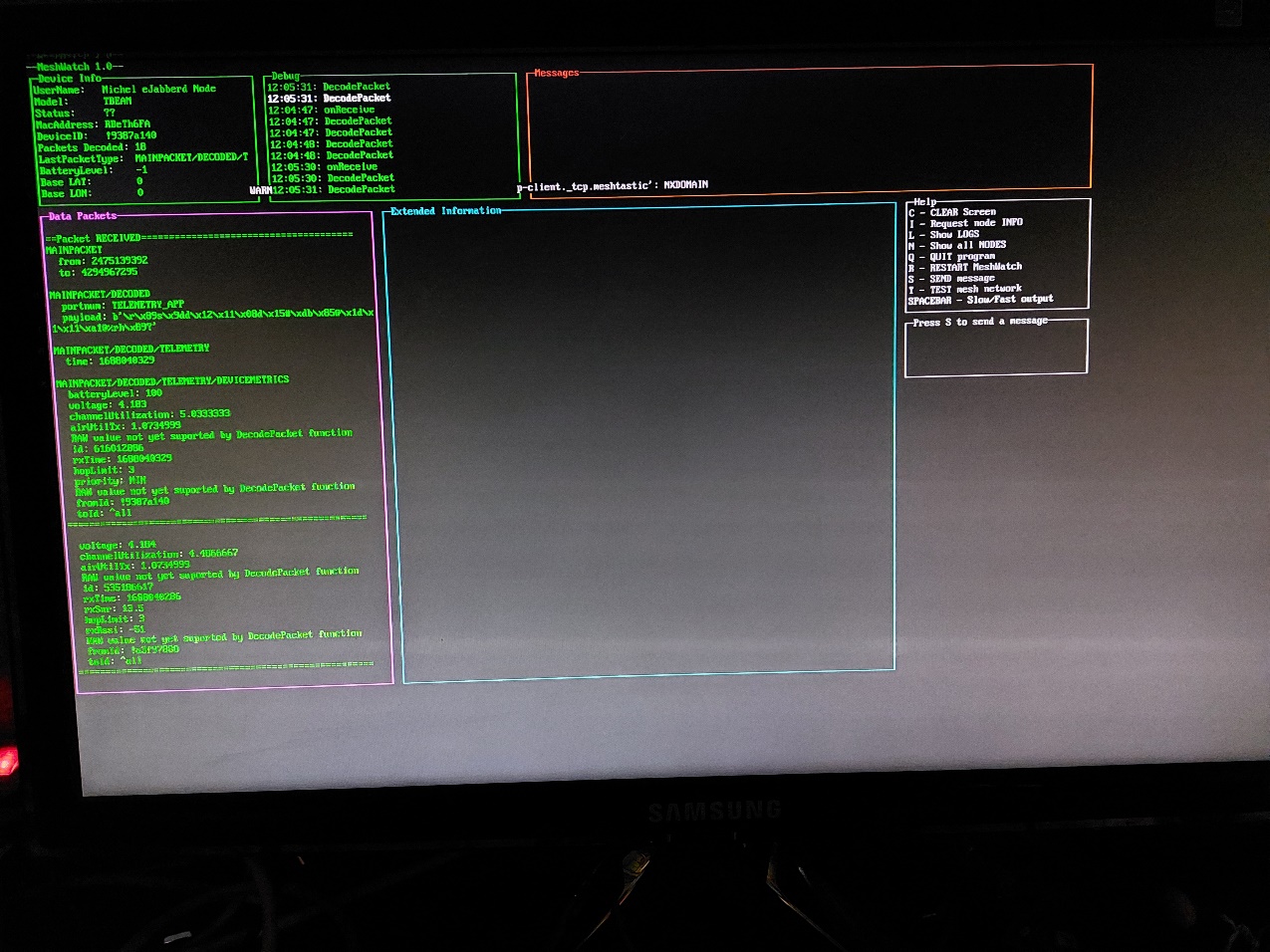


Bien mettre l’IP de la freedom box à « Serveur personnalisé » !!!

**Fonctionnement du programme Meshwatch adapté par ON3MEE**



**Vue de Meshwatch**

A gauche, le projet d’une gobox Meshtastic pour le terrain. On peut y voir un module T BEAM relié en USB à un RPI 3B+ et un routeur WIFI D-LINK. A droite, la vue de Meshwatch sur l’écran.

* **Freedom box**

<https://freedombox.org/fr/>

La distribution Freedom Box pour raspberry pi est l’OS de tout le système. Elle inclut nativement un serveur XMPP et client web de tchat.

* **Meshtastic**

Il faut installer tous les outils Meshtastic ensuite selon la procédure indiquée sur Meshwatch.

* **Meshwatch**

<https://github.com/datagod/meshwatch>

Meshwatch (anciennement Meshtalk) est un outil développé en python pour afficher et envoyer des messages avec un rasperry pi.

Il faut suivre la procédure indiquée sur le github.

Bien mettre à jour les dépendances Meshtastic si vous utilisez la version Meshtastic V2.x !

* **PYXMPP2**

<https://github.com/Jajcus/pyxmpp2>

PYXMPP2 est en client xmpp python. Il a été intégré au code source de Meshwatch afin de faire le lien entre le monde Meshtastic et le serveur XMPP.

**Brève description du programme**

Tous les programmes se trouvent dans « */home/admin/meshtalk* ».

« *meshwatchproc.sh* » est le programme principal. C’est un watchdog.

« *meshwatch.py* » est le code source de Meshwatch.

« *sendxmpp.py* » est le sous-programme qui sert à envoyer des messages de Meshwatch vers XMPPPY2.

Les codes sources sont disponibles avec les fichiers du projet.

**CONCLUSION**

Voilà le projet Gateway Meshtastic Freedom Box.

Il peut être utilisé en l’état sur un LAN pour permettre à plusieurs stations de se connecteur sur un seul node.

Des évolutions vont être apportées pour enregistrer un log des messages par exemple.

N’hésitez pas à consulter mes autres tutoriels sur [www.on3mee.be](http://www.on3mee.be) .

N’hésitez pas aussi à utiliser mon projet parallèle Freedom Box Off Grid : <https://www.on3mee.be/informatique/freedom_box_off_grid.htm>