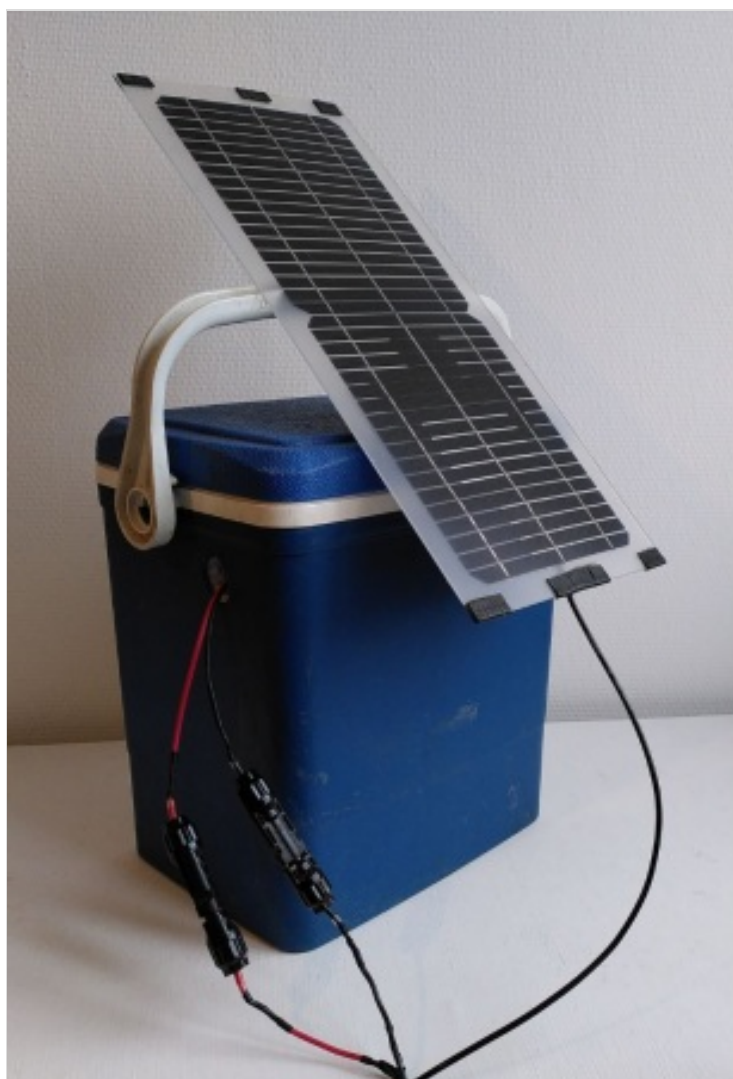


Dominik - module électronique central


 Girard15



MODULE ÉLECTRONIQUE CENTRAL

https://wiki.lowtechlab.org/wiki/Dominik_-_module_%C3%A9lectronique_central

Dernière modification le 23/01/2023

 Difficulty **Hard**

 Duration **4 hour(s)**

 Cost **0 EUR (€)**

Description

Ce tutoriel vient compléter le tutoriel "Dominik - serveur modulable autoalimenté en réemploi". Il a pour but de présenter la configuration et gestion d'un système électronique pour satisfaire plusieurs fonctionnalités essentielles du numérique.

Summary

Contents

Description

Summary

Introduction

Step 1 - Écrire un article html sur le site

HTML, c'est quoi ?

Des templates tout fait

Bonnes pratiques quand on crée un site

Step 2 - Installer un serveur local sur smartphone - Méthode simple

Prérequis

Déroulé

Rendre le site accessible depuis internet

Annexe :

Step 3 - Installation d'un serveur web local apache sur un smartphone pour afficher un site html

Prérequis

Installation de Termux sur le téléphone

Installation de Apache2

Se connecter au serveur

Remplacer la page par défaut par son site ou le mettre à jour

Step 4 - Rendre son site accessible par n'importe qui sur internet

Configuration de la redirection de port sur le routeur (box wifi) de votre maison

Introduction

Prérequis

Théorie de la redirection de port

Trouver son routeur

Se connecter à sa box pour administrer son réseau local

Connecte ton serveur maison

Redirection de port

Paramétrages bonus :

Conclusion

Obtenir un nom de site grâce à un service de nom de domaine (DNS)

Adresse ip dynamique et script cron

Step 5 - Installation et paramétrage d'une connexion SSH (bonus)

Prérequis

Installation du serveur SSH sur Termux et du client SSH sur l'ordinateur

Créer un mot de passe sur la session du téléphone pour sécuriser les connexions

Mode d'utilisation pour écrire des commandes à distance

Quelques commandes utiles

Mode d'utilisation pour copier un fichier et mettre à jour le site

Step 6 - Utiliser Manyverse pour interagir dans le réseau local

Rappel : Comment mettre en place notre réseau local ?

Comment mettre en place Manyverse dans notre réseau local ?


Avertissement sur la différence entre la solution Termux et la solution Manyverse :

Comments

Introduction

Ce module s'inscrit dans un tutoriel plus général : Dominik - serveur modulable autoalimenté en réemploi


Il sert donc à proposer des méthodes par thèmes pour concevoir votre propre Dominik, mais peu aussi être pris à part pour l'aspect technique abordé.

 Difficulté : Moyen

 Durée : 2 à 4h

 Coût : 0€

L'objectif de ce module central est d'obtenir un smartphone fonctionnel permettant l'hébergement de fichiers sur un serveur local qu'il contient, l'hébergement d'un site web ainsi que sa mise en ligne par une WIFI extérieure et le partage de connexion pour permettre l'accès à internet. Il permettra aussi de créer autour de lui un réseau local auquel d'autres appareils électroniques environnants pourront se connecter.

 Dominik - serveur modulable autoalimenté en réemploi

Step 1 - Écrire un article html sur le site

Dans cette étape, nous verrons pas à pas comment écrire et ajouter un article sur le site, grâce aux modèles templates. Ce guide prends pour exemple la publication d'article sur notre site, tu n'auras probablement pas le même site donc pas le même code !

HTML, c'est quoi ?

HTML (HyperText Markup Language) c'est un langage utilisé pour créer des pages web. Il permet d'indiquer à un navigateur (Chrome, firefox, etc.). Ce que l'on souhaite qu'il affiche. Par exemple, pour écrire

J'ai trouvé des infos intéressantes sur le Kosovo, regarde c'est ici : Kosovo.

On écrit

```
<h4>HEY !</h4>
<p>J'ai trouvé des infos intéressantes sur le Kosovo, regarde c'est ici : <a href="https://fr.wikipedia.org/wiki/Kosovo">Kosovo</a>.</p>
```

Les balises `<h4>` permettent ici d'indiquer un titre de niveau. `<p>` indique un paragraphe et `<a>` permet de créer un lien. À chaque fois, il faut fermer la balise, avec une balise similaire avec un `/` comme `</p>`.

Une fois qu'on a écrit tout notre code dans un fichier texte dans un éditeur de texte normal ou dans notepad++ ou Sublime Texte (pour avoir des balises colorées, c'est plus joli et pratique), on l'enregistre en '*NomDuFichier.html*', et on l'ouvre depuis n'importe quel navigateur.

Derrière chaque site il y a ce code html. On utilise aussi des fichiers écrit en `.css`, un langage assez proche créé uniquement pour mettre en forme et rendre joli ton site html. Enfin, beaucoup de gens utilisent des scripts javascript, qui s'exécutent au lancement de la page, mais on a décidé de ne pas en utiliser : cela complexifie les sites et les rend moins sobres selon nous.

Plein de tutoriel plus ou moins complet sont disponibles sur internet pour apprendre rapidement :

- Les bases de html et les bases de css, pour apprendre juste l'essentiel
- Un cours assez complet sur OpenClassroom

Il existe aussi des générateur de site statiques comme Pelican, mais c'est un peu technique à prendre en main.

Des templates tout fait

Pour te simplifier la vie, on a créé des templates (des modèles prêts à remplir) de site tout prêt. Tu peux les retrouver dans l'espace de stockage.

Grâce à eux, pas besoin de comprendre html en détail, tu as juste à copier les fichiers, les ouvrir avec un éditeur de texte et copier les ensembles de balises en commentaires (ceux entre ces balises) et remplacer les textes en majuscules par ce que tu veux.

Dans ces templates, il te suffit de mettre les bonnes infos au bon endroit !

- Ouvre d'un côté le fichier code `index.html` dans un éditeur de code html (notepad++), et ouvre le même fichier dans un navigateur internet (chrome, firefox).

- Cherche par exemple le titre du document (main title) dans le code et remplace le. Il doit être entre des balises `<h1>Main Title</h1>`

Puis remplace ce que tu veux !

Quelques indices pour trouver ce que tu cherches :

- Les **titres** sont entre des balises `<h2> Titre </h2>` le numéro peut varier en fonction de la taille du titre.
- Les **paragraphes** de textes sont entre les balises `<p> ... </p>`
- Les **images** sont entre les balises ``
 - `src="images/IMG_6045.jpg"` indique le chemin et le nom de l'image. Ici, PAR RAPPORT à la page où se trouve la page, l'image est dans le dossier images.
 - Pour indiquer un chemin qui part du dossier supérieur à celui dans lequel est ta page, tu peux écrire « `../nomdelimage.png` »
- Les **liens** :
 - `Crowdfunding` → Ici href permet de préciser l'emplacement de la page de notre site que l'on veut mettre en lien sur la page d'accueil. « Crowdfunding » c'est le texte qui va s'afficher.
 - `Tripalium` → Ici on fait un lien vers un site.
- Les **listes** :
 - ` Point 1 Point 2 Point 3 `

Le fichier CSS

On va pas trop s'attarder dessus, mais tu remarqueras peut-être qu'il y a des fichiers *layout.css* et *framework.css*. Ces fichiers, c'est les fichiers du `~ style` ~.

html c'est la structure et le fond du site, css c'est un langage qui sert à pimper ton site. En soit, tu peux mettre des couleurs, changer les tailles des images, changer les polices directement dans les fichiers html mais le fichier css permet de centraliser ces changements. Par exemple pour tout notre site lowtechnumerique.mooo.com, on utilise qu'une seule feuille de style css.

C'est vraiment pas très compliqué, en gros tu dis à quoi tu veux appliquer un changement et quels paramètres tu change, puis quel valeurs.

Genre :

```
img{
  max-width:100%; ← la taille maximale de l'image ne doit pas dépasser celle de l'écran

  height:auto; ← la hauteur est réglée automatiquement en face de la classe

  border: 1px solid grey; ← on ajoute une bordure grise d'un pixel
}
```

Bref, celui des templates est vachement fourni donc pas facile à changer, mais tu peux essayer quand même ! Et si tu réécris ton site de A à Z, en vrai c'est plus simple et il sera peut-être moins énorme.

Bonnes pratiques quand on crée un site

- Limiter la taille du site :
 - Limiter la quantité et la taille des vidéos, des sons, et des images.
 - Limiter le contenu inutiles, fichiers, etc
- Eviter d'utiliser des scripts (javascript) et du contenu dynamique (certe ça permet d'avoir des sites super stylés et internet, mais ça fait plusieurs connexion au lieu d'une seule, donc ça multiplie le trafic internet)
- Eviter les cookies, les pubs, les services tiers et services d'analyse. Tu peux retrouver plus de détail dans le guide dédié

Si tu as réalisé ces manipulations sur des copies des fichiers html de ton site sur ton ordinateur, il faut maintenant mettre à jour les fichiers html de ton site sur ton serveur, grâce à la partie *Remplacer la page par défaut par son site ou le mettre à jour* du guide *Installation d'un serveur local sur smartphone*

Step 2 - Installer un serveur local sur smartphone - Méthode simple

L'objectif de cette étape est d'installer un serveur sur son téléphone qui pourra afficher un site web aux personnes connectées au même réseau (même wifi par exemple). Ici, on utilisera une méthode plus simple que la méthode complète, qui ne demande pas d'exécuter de ligne de commande, grâce à des applications. C'est plus facile, mais cela permet moins de choses qu'avec un vrai serveur http comme décrit dans la méthode complète.

Prérequis

- Avoir créé un site html basique sur son ordinateur. Tu peux par exemple utiliser les templates mis à disposition ou aller voir le guide *Écrire et publier un article grâce à html*
- Avoir un smartphone qui fonctionne encore plus ou moins !

- Etre connecté à une box wifi ou un partage de connexion

Déroulé

Tout d'abord, transférez les fichiers HTML sur votre téléphone (en le connectant à l'aide d'un câble par exemple). Tu peux créer un dossier spéciale, par exemple *MonSite* pour les stocker en un même endroit.

Ensuite, installe une application de gestion de serveur http. Il en existe beaucoup, comme par exemple *http server*. Si tu n'as pas accès au playstore ou que cela ne fonctionne pas car le portable est trop vieux, tu peux télécharger directement depuis internet un fichier .apk, un type de fichier qui permet d'installer une application. Par exemple celui-ci.

Les smartphones n'aiment pas trop ce type de fichier car ils ont peur que cela soit des virus, mais tu peux normalement l'autoriser à le télécharger, puis cliquer dessus pour l'exécuter et ainsi installer l'application. Il te demandera d'activer un paramètre permettant d'exécuter des applications tierces. Tu peux l'activer sans danger.

Clic sur l'application puis sur "Start". C'est bon ! Les fichiers de ton téléphone sont accessibles par n'importe qui connecté au même wifi ! Tu peux tester en rentrant l'adresse écrite sur l'application, quelque chose comme `http://192.168.1.137:8080/`. C'est ton adresse IP locale. Bon c'est sympa, mais on aimerait bien que cela affiche directement le site web !

Dans l'onglet "réglage" tu peux modifier l'adresse du "document root" pour mettre celle du dossier dans lequel tu as mis les fichiers http. Maintenant, les gens qui se connecteront à l'adresse indiquée (ton adresse ip) arriveront directement dans ce dossier, et ne pourront pas remonter plus haut et avoir accès à tous tes fichiers. C'est quand même plus sécurisé !

Pour que les gens arrivent directement sur ta page d'accueil, il faut qu'il rentre l'adresse en ajoutant le nom du fichier de ta page d'accueil : `http://192.168.1.137:8080/index.html`

Et voilà ! Ton site s'affiche !

Rendre le site accessible depuis internet

Bon c'est bien sympa, mais si on veut que quelqu'un qui n'est pas connecté à la même wifi accède au site, comment on fait ? Et bien tu peux suivre le guide *Mettre en ligne le site sur internet*. mais je t'explique rapidement ici :

Il faut que notre box laisse passer les gens qui cherchent à accéder au site depuis internet. Sur l'app on voit que le site se trouve sur le port 8080 et à l'IP 192.168.1.137 (chez moi). On se connecte sur notre box qui a la même adresse mais en .1 ou .255 à la fin.

On se connecte à la plateforme de gestion de la box (souvent il y a un mot de passe admin par défaut : ton mot de passe pour te connecter au wifi et admin en identifiant, ou appuyer 5s sur le bouton wps de la box chez moi, ou parfois c'est marqué sous la box)

Une fois sur la box, dans la catégorie sécurité, on paramètre une "redirection d'un port" vers l'ip locale de notre réseau (ip notée sur l'app 192.168.1.137). On choisit le protocole TCP. On met le port 8080 partout.

C'est bon on peut accéder au serveur depuis l'extérieur, en entrant notre IP publique suivie de '/index.html'. Tu peux connaître ton ip en cliquant sur ce lien <https://www.myip.com/>. Chez moi cela donne : `http://79.86.49.168:8080/index.html`

Bonus : on peut donner un nom à notre serveur pour fixer l'ip et faire moins brouillon, car les gens retiendront mieux un nom de site qu'une suite de chiffres. On utilise pour ça un service DNS. On peut aussi rendre l'adresse du locale du tel stable. Pour cela, on te laisse regarder le guide *Mettre en ligne le site sur internet*

Annexe :

Voici un site html simple si tu veux tester sans en avoir vraiment créé un. Tu peux le copier dans un fichier texte (.txt) et l'enregistrer au format .html

index.html

```
<!DOCTYPE html>

<html lang="fr">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title> Lowtech Numerique </title> </head>

<body style="font-family: sans-serif; color:#5d5f65; text-align:center">

<h1>Mon site Low tech</h1>

<p>Petit test d'hebergement sur telephone</p>

<p><a href="next.html">Lien vers une page</a></p>

</body>

</html>
```

Step 3 - Installation d'un serveur web local apache sur un smartphone pour afficher un site html

L'objectif de cette étape est d'installer un serveur sur son téléphone qui pourra fournir aux personnes connectées au même réseau (même wifi par exemple) le contenu d'un site web. Apache2 est un logiciel qui permet de gérer les pages de ce site web sur le serveur. Pour l'installer, il faut pouvoir exécuter des lignes de commandes sur le téléphone à l'aide de l'application "Termux".

Prérequis

Ce tutoriel peut être utilisé par n'importe qui et ne nécessite pas de connaissances informatiques particulières. Normalement, la plupart des étapes peuvent être réalisées sur n'importe quel téléphone, et sur n'importe quel ordinateur, du moment que Termux arrive à s'installer sur le téléphone.

- Un smartphone à peu près fonctionnel
- Une connexion internet pour tout installer
- Pour tester si tout fonctionne, être connecté avec d'autres appareils au même réseau local. Par exemple, dans un appartement, quand un smartphone et un ordinateur se connectent en Wifi à une box internet, ou le même partage de connexion, alors, ils sont sur le même réseau local (c'est possible que ça ne fonctionne pas sur les wifis d'école ou d'entreprise).
- Une ou des pages html prête à être hébergée sur le serveur : c'est donc bien d'avoir déjà un site html tout prêt !

Installation de Termux sur le téléphone

Le plus simple, c'est de télécharger Termux depuis le playstore.

Tu peux cliquer sur l'application et voilà, tu as ce qu'on appelle un terminal qui te permet d'écrire et d'exécuter des "commandes". Les commandes c'est des instructions que tu donnes à ton appareil, par exemple "créer un fichier", "emmène-moi dans tel dossier", "installe ce logiciel". Il y a des gens qui connaissent toutes les commandes par cœur, et d'autres (comme nous) qui copient celles qu'on leur dit d'écrire ! Fait bien attention à les copier à la lettre (mêmes majuscules, mêmes espaces, ...).

Si tu as envi d'apprendre et d'approfondir, il existe pas mal de listes de commandes et de tutoriels sur internet pour apprendre. Windows a ses propres lignes de commandes, ce fichier donne des correspondances, mais la plupart des commandes ici sont exécutées sur votre téléphone, pas sur Windows.

Grâce à termux, on va pouvoir installer le logiciel qui permet de gérer le serveur (Apache2), mais on va aussi pouvoir se connecter à distance au téléphone pour uploader des fichiers dessus par exemple (SSH).

Installation de Apache2

Ici on s'est inspiré de ce tutoriel.

On peut maintenant installer Apache2, le logiciel qui permet de gérer un serveur (ou plutôt de gérer le protocole http et https, qui permet de se faire passer des pages web. Techniquement parlant Apache2 EST le serveur, c'est lui qui "sert" les pages quand on les demande).

Pour cela rentrer les commandes suivantes. La première permet d'être sûr que tout est à jour sur notre téléphone et la deuxième permet de chercher et d'installer Apache2.

```
pkg update && pkg upgrade
pkg install apache2
```

En cas de soucis, si Termux dit ne pas trouver Apache2 dans la bibliothèque d'adresse de package, tu peux essayer de changer le dossier dans lequel il fouille pour le trouver.

```
termux-change-repo
```

Une fenêtre s'affiche : tu peux entrer dans l'option "main repo" et changer de dossier en en cochant un autre (un miroir ou un repo main ou principale) . Nous c'est le deuxième main repo qui a marché.

Une fois le serveur http installé, on peut le lancer à l'aide d'une des deux commandes suivantes :

```
apachectl
httpd
```

Termux affiche un avertissement lié au ServerName, mais cela fonctionne quand même.
À chaque fois que tu voudras relancer le serveur, il faut exécuter cette commande httpd.

Se connecter au serveur

Pour se connecter au serveur il faut connaître son IP.

L'adresse IP est un numéro d'identification attribué à un appareil connecté sur un réseau. Tous les appareils sur ton réseau local (les appareils connectés au même wifi) ont une adresse IP locale. Par contre, ils ont la même IP publique : sur internet, on te retrouve avec l'IP de ta box internet, et c'est elle qui redirige ensuite vers le bon appareil. Tu peux voir ça comme une coloc où chacun à un numéro de chambre. Quand on t'envoie une lettre, on l'envoie à l'adresse de la coloc, qui est la même pour tout le monde à l'intérieur. Le courrier est ensuite réparti entre vous dans la coloc, sans que les gens à l'extérieur aient besoin de connaître vos numéros de chambre.

Pour trouver ton IP local, sur termux, écrire :

```
ifconfig
```

La commande affiche les adresses IP du téléphone dans les différents réseaux auxquels il est connecté. L'adresse IP qui nous intéresse est celle située juste après inet dans la catégorie wlan0. Bon courage pour fouiller, c'est un peu long à lire !

Ensuite, sur un ordinateur ou un téléphone connecté sur la même wifi, écrire dans la barre de recherche d'un navigateur l'IP, puis le port sur lequel est le site (par défaut, ce sera 8080) :

```
http://192.168.1.50:8080
```

Normalement une page créée par défaut par apache s'affiche, avec un 'it works!', tout fonctionne !! C'est l'heure de prendre une pause goûter bien méritée !

Remplacer la page par défaut par son site ou le mettre à jour

Manuellement

Bon elle est sympa cette page mais on aimerait bien la remplacer par notre propre page !! Dans cette partie, il faut donc **déjà** avoir créé un site html. Tu peux aller voir le tutoriel expliquant comment écrire un article en HTML, qui t'expliquera comment créer une page html simplement. Par défaut, sur les versions d'Android supérieur à Android 6, Termux n'enregistre pas de fichier accessible par qui que ce soit d'autre que lui, et il n'a pas accès au reste... Pas très pratique si on veut lui remplacer le fichier qui sert de page d'exemple d'apache par les fichiers de notre site web.

On va donc activer cette fonctionnalité en exécutant :

```
termux-setup-storage
```

puis on peut cliquer sur "autoriser" dans la fenêtre qui s'affiche.

On peut ensuite **brancher le téléphone sur notre ordinateur**, autoriser l'accès aux fichiers si c'est demandé, et aller chercher tous nos fichiers html sur l'ordinateur et les copier sur le portable dans un dossier site, créé dans le dossier downloads par exemple (storage/downloads/site). Tu peux maintenant débrancher ton téléphone.

Depuis termux, on va ensuite déplacer tous les fichiers contenus dans le dossier site vers le dossier contenant le fichier index.html par défaut d'apache. Pour cela, on va d'abord revenir au dossier home à l'aide de la commande cd (change directory), puis exécuter la commande mv, qui sert à déplacer (move) les fichiers contenus :


```
cd ~
mv storage/downloads/site/* ../usr/share/apache2/default-site/htdocs
```

Depuis un navigateur privé sur ton ordinateur (qui doit être sur le même wifi que ton téléphone), rentre l'ip de ton téléphone et le numéro de port, comme plus haut.

```
http://192.168.1.50:8080
```

Normalement ton site s'affiche !

A terme pour mettre à jour le site sans toucher les fichiers qui n'ont pas changé, il peut être pratique d'utiliser la commande **rsync**, et parfois la commande **cp**, qui permet de copier

En SSH

C'est aussi possible d'utiliser une connexion ssh pour copier un fichier de son ordinateur vers le smartphone, comme détaillé dans le guide *Paramétrer une connexion SSH*.

Il faut que SSH soit installé sur le portable à l'aide de :

```
pkg update && pkg upgrade
pkg install openssh
```

puis l'activer avec

```
sshd
```

Maintenant, par exemple, on peut essayer de rajouter un fichier dans le dossier contenant tous les fichiers de notre site. On ouvre un terminal sur notre ordinateur **là où se trouve notre fichier** (clic droit puis "ouvrir dans un terminal"), puis on exécute la commande suivante :

```
scp -P 8022 nomDuFichier nomUtilisateur@AdresseServeur:~/chemin/nouveauNomDuFichier
```

Par exemple, la commande suivante envoie le fichier diapo.pdf sur mon serveur, dans le dossier home de termux, sans le renommer :

```
scp -P 8022 diapo.pdf u0_a208@192.168.1.50:~
```

Si on veut mettre à jour tous les fichiers du site, on peut, sur notre ordinateur, ouvrir un terminal dans le dossier contenant tous les fichiers du site (html, css, image...), et exécuter

```
scp -r -P 8022 * u0_a208@192.168.1.50:~/usr/share/apache2/default-site/htdocs
```

Puis rentrer le mot de passe. Après quelques longues secondes d'attente, le site se met à jour ! L'option -r permet de copier des dossiers et pas que des fichiers.

Step 4 - Rendre son site accessible par n'importe qui sur internet

Ici on va apprendre à rendre son site accessible par n'importe qui, en manipulant la redirection de port et le DNS. Attention, nous ne recommandons pas de mettre en ligne un site qui sert à des usages autres que purement informatifs (formulaires, système de comptes, paiement, données sensibles, etc.) à moins d'avoir paramétré le protocole https (<https://homebrewserver.club/fundamentals-webserver-website.html#https>)

Ok, c'est super, dans l'étape précédente on a créé un site qui peu être vu par n'importe qui connecté au réseau local, mais ça serait encore mieux si n'importe qui sur internet pouvait accéder au site non ? Pour ça il faut faire deux choses : obtenir un nom de site internet qui redirigera les gens qui rentreront ce nom sur internet vers notre box internet, et demander à notre box internet de laisser rentrer les gens sur son réseau local et de les redirigeant vers le serveur hébergeant le réseau local.

Configuration de la redirection de port sur le routeur (box wifi) de votre maison

Traduit et adapté depuis <https://homebrewserver.club/fundamentals-port-forwarding.html>

Introduction

En théorie internet est un espace bidirectionnel. N'importe qui possédant une connexion internet peut non seulement consulter des choses sur internet mais peut aussi en héberger !

Au temps des cloud et des serveurs virtuels privés, on l'oublie facilement, et les fournisseurs d'accès internet ne cherchent pas à nous faciliter la tâche, mais un serveur auto-hébergé peut-être aussi simple qu'un vieille ordinateur portable connecté directement à notre box wifi. Néanmoins, on a besoin de modifier quelques réglages sur le routeur pour que cela arrive !

Prérequis

Pour pouvoir héberger un serveur chez soit il faut :

- Être sur d'avoir physiquement accès à sa box wifi
- Connaître le mot de passe d'administration (c'est généralement fourni avec la box ou écrit sur l'étiquette sous la box. Parfois tu peux aussi appuyer plusieurs seconde sur le bouton "WPS" de la box pour te connecter dessus en tant qu'administrateur)
- Avoir un support d'hébergement avec un serveur local déjà paramétré et si possible un serveur open SSH qui tourne dessus (pour pouvoir le contrôler à distance, mais tu peux aussi le paramétrer directement depuis le serveur)
- Un câble Ethernet pour se connecter au routeur (si tu utilise un smartphone comme serveur, pas besoin, connecte toi juste en wifi !)

Théorie de la redirection de port

Par défaut, les box internet ont un pare-feu configuré pour que les appareils derrière elles soient inaccessible depuis internet. Cela permet d'éviter que ton réseau privé (les appareils de ta maison : ordinateur, téléphone, imprimantes, etc.) soient accessible publiquement par n'importe qui.

Les machines derrière la box wifi (appelée le réseau local ou "local area network" (LAN)) peuvent se connecter au gros réseau internet (Wide area network, WAN), mais la connexion ne peut pas se faire dans l'autre sens.

Néanmoins, quand on héberge un serveur à la maison, on veut que notre serveur soit accessible depuis internet. Pour cela, on doit ouvrir ce qu'on appelle les "ports réseaux".

Les ports sont des portes logiques qui sont ouvertes ou fermées aux connexions. Ces ports ont des nombres et sont standardisés pour des protocoles spécifiques ou des applications particulières. Par exemple le trafic http passe par le port 80 par défaut. Https sur le port 443 et SSH sur le port 22.

Pour rendre notre serveur accessible depuis internet on doit ouvrir les ports sur la box wifi et les rediriger vers notre serveur. Cela s'appelle la redirection de port (port forwarding en anglais).

La méthode exacte de la redirection de port est différente pour chaque routeur (ou box wifi). Néanmoins, cela suit toujours le même schéma où le trafic entrant par un certain port sera redirigé vers l'adresse ip de notre serveur et un port sur notre réseau local.

C'est pour cette raison qu'on a besoin d'un accès à la page d'administration de notre retour.

Trouver son routeur

Pour accéder au panneau d'administration de son routeur (box wifi), il faut trouver son adresse IP sur le réseau local (pas celle publique). Tu peux faire cela en te connectant à ton routeur en Ethernet ou en wifi, et en trouvant ensuite quelle est ta propre adresse IP locale.

Sur les systèmes basés sur Debian (Linux), ou sur Termux sur ton portable, tu peux écrire dans un terminal (un terminal s'ouvre en faisant ctrl + alt + T ou en cherchant le programme Terminal) :

```
$ ifconfig
```

Cela renvoie les informations de connexion. Cherche la ligne où est écrit

```
inet
```

(mais il y en a plusieurs, plutôt celle après le 3: wl** et pas celle 1: lo)

Dans mon cas, l'adresse IP est 192.168.1.11

Sur windows tu peux chercher le programme "invite de commande" et écrire :

```
ipconfig
```

Si tu es en wifi, l'adresse IP local sera affiché à la ligne "IPV4" dans la catégorie réseau sans fil.

L'adresse IP locale commence souvent par 192.168.x.x tu peux essayer de changer le dernier numéro soit en 1 soit en 254 pour trouver ton routeur : 192.168.1.1 pour moi.

Se connecter à sa box pour administrer son réseau local

Sur un navigateur internet (Firefox, chrome, etc.), écrit l'adresse ip dans la bar de recherche (donc ici 192.168.1 ou 192.168.254). Le navigateur devrait afficher une page de connexion, depuis laquelle tu pourras te connecter à la page d'administration de la box grâce à un mot de passe et un identifiant (souvent cela va être admin et le mot de passe d'origine, ou un mot de passe indiqué dans un livret fournit avec la box ou indiqué sous l'étiquette. Tu peux aussi appuyer plusieurs longues secondes sur le bouton "WPS" de la box si il y en a un, pour que la connexion soit autorisée).

Ici tu auras donc accès à beaucoup de réglages. Les options "LAN" "DHCP Leases" ou "Network" ou "Réseau" peuvent permettre d'avoir une vue d'ensemble sur tous les appareils.

Connecte ton serveur maison

Si ton serveur est hébergé sur un smartphone, connecte le au wifi. Si tu utilises un ordinateur ou une raspberry pi tu peux utiliser un câble Ethernet que tu peux brancher sur la box (celui marqué LAN ou INET si il y en a plusieurs).

Une fois notre serveur connecté au réseau local, on peut regarder sur le site d'administration de la box quelle IP locale a été attribuée au

serveur. Si c'est un serveur sur smartphone et que tu as déjà installé une application de gestion de serveur http sur ton téléphone, tu peux voir ton ip sur cette application. On supposera dans ce tutoriel que l'adresse est 192.168.1.10.

Ensuite, essaye de trouver l'option appelée "static (DHCP) Lease" ou "DHCP Binding" ou "adresse IP statique" ou "baux statiques" chez moi dans la catégorie LAN à gauche, ou quelque chose de similaire sur l'interface d'affichage de ton réseau local. Assigne à ton serveur une static DHCP lease (une adresse IP statique). Cela permettra d'être sûr que le serveur est toujours joignable à la même IP locale.

Redirection de port

Une fois que tu as paramétré une ip statique à ton serveur local, tu peux commencer la redirection de port. Selon la marque de la box, cela peut aussi s'appeler "partage de port", "redirection de trafic", "traffic forwarding", "port sharing", "NAT/PAT", etc. Cela peut se trouver dans différents onglets : "sécurité", "contrôle des accès", "internet", "configuration réseau"...

On te conseille de chercher sur internet "redirection de port [le nom de la marque de ta box]" comme ça tu auras les conseils qui correspondent clairement à ta box.

Le principe du processus est de déterminer quel port externe ouvrir et vers quelle adresse IP sur le réseau local et quel port le rediriger. Quelques informations supplémentaires seront peut-être demandées, comme le protocole (TCP, UDP, ou les deux). Parfois on peut aussi paramétrer un intervalle de port plutôt qu'un unique port.

Pour ouvrir les ports pour le serveur web, on ouvrira deux ports séparés : un pour le protocole HTTP et un pour le protocole sécurisé HTTPS. Le port externe 80 sera donc ouvert pour le serveur et directement redirigé à l'adresse locale du serveur maison.

Name: "HTTP Homeserver"

Protocol: TCP

Device: 192.168.1.10

External Port: 80

Port to device: 8080

Sur smartphone, Apache fonctionne sur le port 8080.

Paramétrages bonus :

De la même manière, ouvre le port 443 pour HTTPS et redirige le vers l'adresse IP local du serveur :

Name: "HTTPS Homeserver"

Protocol: TCP

Device: 192.168.1.10

External Port: 443

Port to device: 443

Enfin, en bonus on peut ouvrir un port pour le SSH (un protocole qui permet de contrôler ton serveur à distance). Ici la différence est qu'on ouvre le port externe 9999 et qu'on le redirige vers le port interne 22. Régler le SSH sur un port différent est une (très) légère barrière de sécurité, qui permet d'éviter des scripts automatique d'avoir accès à ton serveur local en testant de rentrer par un port connu pour se connecter en SSH (le 22).

Si tu ne comptes pas utiliser de SSH depuis ailleurs que ton réseau local, ne paramètre pas cette redirection, cela évitera des risques de sécurité !

Name: "SSH Homeserver"

Protocol: TCP

Device: 192.168.1.10

External Port: 9999

Port to device: 8022

Conclusion

Maintenant que tu as ouvert les ports correspondants, tu devrais être capable de taper ton adresse IP externe dans ton navigateur et tu devrais être automatiquement redirigé vers le site web de ton serveur maison. Ton IP externe peut être trouvée très facilement en demandant à un site internet de te la donner, par exemple sur <http://www.mon-ip.com/>

Obtenir un nom de site grâce à un service de nom de domaine (DNS)

Dans l'étape précédente, on a réussi à obtenir un site internet accessible depuis internet, mais cela passait encore par une adresse IP compliquée qu'on devait rentrer sur le navigateur. Nous on aimerait bien avoir une vraie adresse de site internet !

Pour ça on utilise des DNS. Les DNS (Domain Name System), c'est en quelque sorte les annuaires téléphoniques d'internet : ce sont des grands registres qui relient le nom d'un site à son adresse IP. Ces annuaires sont stockés sur des serveurs que notre navigateur internet va consulter quand on lui demande d'accéder à <https://lowtechlab.org/fr>, pour obtenir l'adresse IP du site et s'y connecter. Il faut donc qu'on demande à un de ces serveurs d'associer notre IP à un nom. Quelques services gratuits existent, mais souvent il nous impose une partie de l'adresse web. Il est parfois aussi possible d'obtenir un serveur DNS directement dans les paramètres de sa box, chez Orange par exemple.

Nous avons utilisé le site suivant :

<https://freedns.afraid.org/>

On peut obtenir un nom de sous-domaine dans la catégorie subdomains à gauche. Il faut se créer un compte.

Image 1

On peut ensuite cliquer sur "créer un nouveau domaine".

On choisit un type A, et on rentre le nom de notre site, puis on choisit le nom de domaine qui sera en suffixe de l'adresse de notre site. Ceux proposer immédiatement en public fonctionne directement, mais il est possible d'en choisir d'autre dans la liste "many more available". Seulement, ceux en privé ne vont pas forcément vous autoriser à utiliser leur suffixe sans vous connaître, donc c'est plus simple d'en choisir un public. Dans "Destination" il faut rentrer votre adresse IP. Il suffit ensuite de remplir le captcha et de valider.

Image 2

Normalement c'est bon ! Si vous rentrez l'URL dans un navigateur, depuis une connexion extérieur au réseau local (utilise un téléphone en 4G par exemple) vous tomber sur votre site internet, si toutes les étapes précédente ont fonctionné !

Adresse ip dynamique et script cron

Le dernier petit problème que l'on pourrait avoir, c'est que cela arrive que l'IP de notre box change, on dit qu'elle est dynamique, ou qu'on déplace la box sur une autre wifi (et qu'on reparamètre la wifi). C'est le cas chez Orange, chez SFR, et parfois chez Bouygue et Free même si c'est pas toujours le cas.

On peut changer manuellement l'adresse IP sur le DNS à chaque fois que cela arrive mais c'est une manipulation à penser en plus.

[Nous ne sommes pas 100% certains que la suite fonctionne parfaitement, mais nous la laissons pour donner une base d'explication. A tester !]

Pour contourner ça, freeDNS propose un service "Dynamic DNS" dans les onglets à gauche (il faut ensuite cliquer sur sa V2) . Ce service consiste en avoir un petit script sur son serveur qui tourne en arrière plan et qui va régulièrement dire à notre service de DNS "Hey ! je suis ici". Comme ça si le site change d'adresse IP, le DNS se synchronise automatiquement.

Le système pour avoir ce petit script, c'est un "cron". Il faut installer un petit logiciel spécifique via termux :

```
pkg install cronie termux-services
```

Il faut ensuite éteindre puis redémarrer termux (tu peux relancer la commande httpd quand tu fais ça pour être sur que le serveur tourne toujours). Puis :

```
sv-enable crond
```

```
crontab -e
```

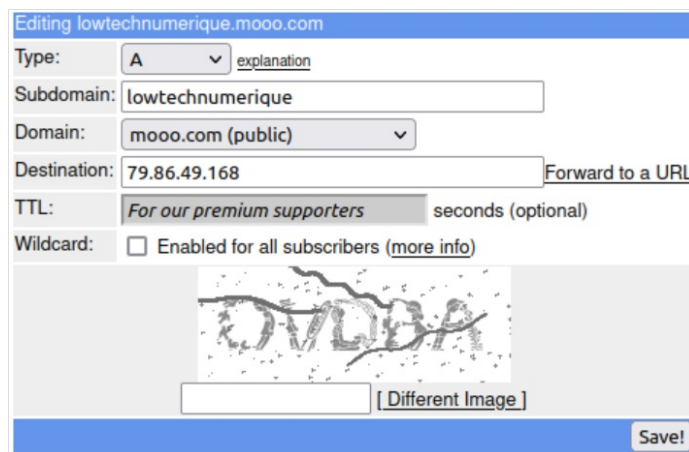
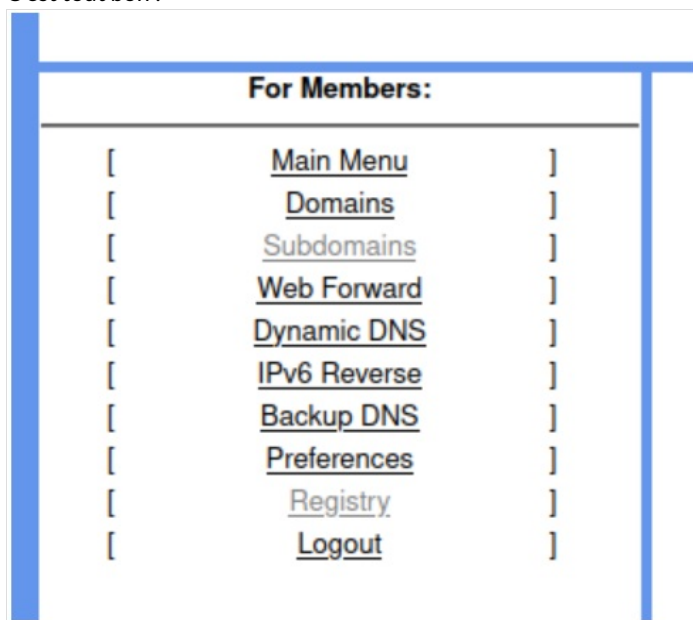
La fenêtre qui s'ouvre est un éditeur de texte. On va écrire la commande cron ici. Et cette commande, c'est freeDNS qui nous la donne :

<https://freedns.afraid.org/dynamic/v2/>

On coche la case du nom de site qu'on a obtenue avant, on clic sur enable, et un tableau apparaît avec un petit lien "Generate a : cron script".

Quand on clic dessus, le script apparaît. C'est la dernière ligne, celle qui ne commence pas par un #, qu'il faut copier dans le fichier cron sur termux (ici forcément, c'est pratique d'avoir une connection ssh pour faire copier collé mais c'est faisable sans, avec un peu de patience). On ferme le fichier après avoir sauvegarder (ctrl X).

C'est tout bon !



Step 5 - Installation et paramétrage d'une connexion SSH (bonus)

Ce tutoriel explique comment créer une connexion SSH entre son ordinateur et un téléphone. SSH est une connexion sécurisée entre deux appareils.

Concrètement cela permet d'exécuter des instructions à distance sur son portable depuis son ordinateur, ou de copier des fichiers depuis l'ordinateur vers le portable.

Prérequis

- Avoir installé Termux sur son téléphone

- Une connexion internet pour l'installation

Pour tout installer on peut soit tout écrire à la main depuis le téléphone, directement sur Termux, soit paramétrer une connexion pour écrire à distance dans Termux depuis son ordinateur. On appelle ça une connexion SSH, et ça peut être pratique pour gérer le serveur à distance, mais c'est du bonus !

Installation du serveur SSH sur Termux et du client SSH sur l'ordinateur

D'abord il faut avoir le bon "logiciel" sur son ordi et sur son téléphone : un client SSH et un serveur SSH. Sur termux sur le téléphone, rentrer les

instructions suivantes :

```
pkg update && pkg upgrade
pkg install openssh
```

Termux va ensuite charger pour installer openssh. Il demandera sûrement des confirmations, d'installation, à chaque fois il suffit d'écrire 'y' puis

d'appuyer sur entrée, ou de rentrer la réponse par défaut. Ensuite, il faut qu'openSSH soit installé sur ton ordinateur. Il y a de grande chance que ce soit déjà le cas par défaut !! Tu peux vérifier qu'il est déjà installé en exécutant dans un terminal (ctr + alt + T ou chercher "Terminal" ou "Invite de commande" dans les programmes) :

```
ssh
```

Si un texte expliquant comment marche la commande (usage : blablablalbla), c'est que SSH est déjà installé ! Si ce n'est pas le cas, sur Linux, tu peux ouvrir un terminal (ctrl + alt + T sur linux). Dedans tu peux

écrire :

```
sudo apt install openssh-server
```

puis rentrer le mot de passe de ton ordinateur. Et confirmer l'installation (yes). Sur Windows, peut-être que ce tutoriel peut t'aider.

Créer un mot de passe sur la session du téléphone pour sécuriser les connexions

Maintenant qu'il est possible d'accéder à distance à notre portable, c'est important de créer un mot de passe pour que n'importe qui ne puisse pas

faire n'importe quoi. Pour cela, sur Termux il faut rentrer la commande :

```
passwd
```

Puis rentrer son nouveau mot de passe deux fois (c'est normal qu'il ne s'affiche pas). Il nous faudra aussi le nom d'utilisateur et l'adresse ip de la

session utilisée par termux, obtenu grâce à la commande sur termux :

```
whoami
ifconfig
```

whoami renvoie le nom d'utilisateur et ifconfig renvoie les ip local que vous avez sur différents réseaux. Souvent celle qui nous intéresse est située juste après inet, dans la catégorie wlan0.

On peut maintenant rendre accessible le portable en ssh en ouvrant la possibilité de s'y connecter en écrivant sur termux

```
sshd
```

Maintenant on peut soit utiliser SSH pour contrôler à distance le portable (en écrivant des commandes depuis notre ordinateur), soit l'utiliser pour copier des fichiers depuis l'ordinateur vers le portable.

Mode d'utilisation pour écrire des commandes à distance

Il ne nous reste donc plus qu'à nous connecter ! Sur l'ordi écrire la commande suivante, en remplaçant les parties entre crochets par les informations que vous avez chez vous :

```
ssh -p 8022 [nom d'utilisateur]@[ip locale du téléphone]
```

Par exemple pour moi

```
ssh -p 8022 u0_a208@192.168.1.50
```

Puis écrire "yes" et entrer le mot de passe paramétré plus haut pour le portable. C'est bon vous êtes connecté (tel le ou la meilleur-e des hackers).

Quelques commandes utiles

pour voir où on est

```
pwd
```

Pour se déplacer, puis pour revenir en arrière

```
cd [Un dossier]/[un dossier dans le dossier]
```

```
cd ..
```

Pour lister les fichiers contenus dans le dossier où

on est :

```
ls
```

Pour trouver un fichier particulier dans tous les dossiers suivants (ici le fichier index.html), puis se déplacer à cet endroit.

```
find . -name index.html
```

```
cd /data/data/com.termux/files/usr/share/doc/pcre2/html
```

Mode d'utilisation pour copier un fichier et mettre à jour le site

C'est aussi possible d'utiliser la connexion ssh pour copier un fichier. Par exemple on peut essayer de rajouter un fichier dans le dossier fichier de notre site. **Sur notre ordinateur**, on ouvre un terminal sur

notre ordinateur **dans le dossier où se trouve notre fichier** (clic droit, ouvrir l'invite de commande ou le terminal), puis on écrit :

```
scp -P 8022 nomDuFichier
```

```
nomUtilisateur@AdresseServeur:~/chemin/Ou/On/Le/Copie
```

Par exemple, la commande suivante envoie le fichier diapo.pdf sur mon serveur, dans le dossier home de termux :

```
scp -P 8022 diapo.pdf u0_a208@192.168.1.50:~
```

Si on veut mettre à jour tous les fichiers du site, on peut, sur notre ordinateur, ouvrir un terminal dans le dossier contenant tous les fichiers du site (html, css, image...), et exécuter

```
scp -r -P 8022 *u0_a208@192.168.1.50:~/usr/share
```

```
/apache2/default-site/htdocs
```

Puis rentrer le mot de passe. Après quelques secondes d'attente, tous les fichiers du site sur notre ordinateur sont copiés dans le dossier d'apache du serveur. Le site se met à jour !

L'option -r permet de copier les dossiers.

On notera que cette commande permet juste de copier des fichiers, et de remplacer ceux qui ont le même nom par leur nouvelle version, mais elle ne

supprime pas les fichiers en trop. Pour cela sur termux ou sur ton ordinateur via une connexion ssh entre l'ordinateur et termux, dans le dossier

contenant le fichier, tu peux utiliser la commande suivante :

```
rm nomDuFichier.html
```

Ressources externes :

<https://doc.ubuntu-fr.org/ssh>

https://wiki.termux.com/wiki/Remote_Access

<https://homebrewserver.club/demystifyingssh.html>

Step 6 - Utiliser Manyverse pour interagir dans le réseau local

Il existe deux solutions pour interagir en utilisant le réseau local, pour échanger des fichiers et des données. La première solution consiste à utiliser

Termux, le terminal de commandes. La seconde a recours à Manyverse.

Si dans le guide d'installation d'un serveur local sur smartphone & dans le guide sur les connexions ssh, on a vu comment on pouvait se connecter au serveur, y trouver des fichiers ou en ajouter, on peut déplorer l'aspect rustique que représente l'interface de Termux, basé sur des

terminaux de commande.

Ainsi, on peut utiliser l'application Manyverse pour avoir un moyen plus "user-friendly" d'échanger des données. Manyverse est un réseau social gratuit et open-source, disponible sur téléphone et ordinateur. Il est basé sur le protocole peer-to-peer SSB. Son intérêt principal est qu'il peut

fonctionner en réseau local et sans connexion internet.

Pour plus d'informations sur Manyverse :

<https://www.manyver.se/>

Pour plus d'informations sur le protocole :

<https://ssbc.github.io/scuttlebutt-protocol-guide/>

Rappel : Comment mettre en place notre réseau local ?

La solution la plus développée et la plus simple à mettre en place est celle de la Wifi, à travers la création d'un Hotspot (point d'accès) Wifi. Ce point d'accès peut être créé par la fonctionnalité "Partage de connexion" disponible sur tous les smartphones.

Pour activer ce "Partage de connexion", il faut généralement se rendre dans les paramètres du téléphone. Chaque smartphone est différent et le plus simple pour trouver comment activer le partage de connexion sur ton smartphone est de rechercher sur un moteur de recherche "activer partage de connexion [nom et modèle du smartphone]".

Remarque : il est possible que la fonctionnalité de "Partage de connexion" ne soit pas disponible sur ton smartphone. Egalement, sur certains smartphones, il est obligatoire d'activer les données mobiles (connexion à la 3G, 4G) pour pouvoir activer le partage de connexion. Cela peut poser des problèmes si tu n'as pas de carte SIM et/ou si tu ne souhaites pas te connecter/consommer d'Internet. Il existe une alternative grâce à des applications qui outrepassent cette obligation. Globalement, il faut rechercher sur ces magasins d'applications "Hotspot WiFi" et en sélectionner une. Mais il faut être vigilant car la plupart propose (voir oblige au bout d'une durée d'utilisation) des versions payantes, certaines ne sont pas très souvent mise à jour et certaines autres limitent l'accès à certains sites lorsqu'on les utilise. On n'a pas trouvé de solution particulièrement fiable, gratuite et open source.

On peut citer notamment Portable WiFi Hotspot Tether disponible sur les magasins d'application classiques.

Une fois la fonctionnalité trouvée ou l'application téléchargée, le partage de connexion peut être activé. Les appareils alentours peuvent repérer le smartphone central en tant que borne WiFi et se connecter à lui. Par exemple, sur l'image, le FAIRPHONE_3_AP1162 est le smartphone qui est en partage de connexion, repéré par un autre smartphone.

Voilà, le réseau local a été techniquement créé !

Comment mettre en place Manyverse dans notre réseau local ?

Nous prendrons une situation avec un smartphone central (qu'on appellera par la suite smartphone 0) dans un Dominik, ainsi qu'un autre smartphone (smartphone 1) et un ordinateur (ordinateur 2), tous les deux extérieurs et à proximité du smartphone 0. Ces 3 appareils sont reliés en réseau local par le point d'accès WiFi créé par le smartphone central.

Il est nécessaire d'installer l'application sur les appareils qui souhaitent échanger des informations. Cela peut être entre un appareil externe et le smartphone central (par exemple si l'appareil externe souhaite lui envoyer des données/fichiers pour les stocker), et cela peut aussi être entre les appareils externes s'ils souhaitent se partager des données directement entre eux grâce au réseau local créé par le smartphone central.

Une fois installé, Manyverse propose une interface simple et claire :

On distingue 4 parties différentes :

- **Connexions** : Cette partie te sert à activer les modes de communication par lequel tu souhaites repérer les autres utilisateurs de Manyverse environnant. Tu peux ensuite les ajouter en tant qu'ami pour suivre leur contenu et interagir avec eux.
- **Partage public** : Cette partie sert à suivre les posts partagés en public par les amis que tu suis. Il existe sur ces posts un système de like, de commentaires ou de partage.
- **Partage privé** : Cette partie sert à interagir en privé avec tes amis. C'est un système de conversation dans lequel tu peux envoyer du texte, des audios, des images et des vidéos.
- **Activités** : Cette partie sert à te notifier les différentes activités de tes amis (like, commentaires, partage, message privé).

Prenons l'exemple précédent de nos 3 appareils : *smartphone 0, smartphone 1 et ordinateur 2*. Le smartphone 0 est en partage de connexion afin de créer un réseau local. Les trois appareils se connectent en réseau local.

Ils peuvent voir dans la fenêtre Public les personnes avec qui ils peuvent devenir ami ("se connecter").

Une fois connecté, il faut que la personne accepte dans la fenêtre Activités. s'ajoute en ami puis publie et s'envoie un message.

Maintenant, à vous de jouer !

Avertissement sur la différence entre la solution Termux et la solution Manyverse :

La première solution permet d'échanger des fichiers grâce à des manipulations par ligne de commande dans le terminal TERMUX. Cet aspect peut sembler assez rétrograde par rapport à l'interface claire et ergonomique de Manyverse. Néanmoins, Manyverse, bien qu'il soit assez basique, représente quand même la remise en place d'un réseau social, dans ses bons comme dans ses mauvais aspects. D'autant plus dans le cadre d'un réseau local, il est intéressant d'interroger l'utilité de certaines

fonctionnalités proposés par Manyverse. Effectivement, l'ensemble des personnes connectés au réseau local créé par le smartphone central de Dominik se trouve dans le rayon de portée du point d'accès WiFi, qui se situe dans la dizaine de mètres autour de Dominik. Ainsi, est-ce que si j'ai quelque chose à dire à une personne qui se trouve à une dizaine de mètres de moi, est-ce que je ne peux pas tout simplement lui dire directement ? Idem, si je souhaite montrer une photo, une vidéo ou une musique que j'apprécie, est-ce que je ne peux pas simplement organiser un temps convivial de partage pour en discuter ? Si la solution Manyverse est séduisante, il est important de ne pas reproduire inutilement une dépendance à une identité virtuelle, et replacer les moments de partages et d'interactions dans

la réalité autant que possible, dans l'optique de recréer des Communs. C'est pourquoi il est intéressant de considérer la première solution, car dans ce cadre, il peut être enviable d'avoir un numérique limité à son utilisation minimale et légèrement difficile d'accès.

