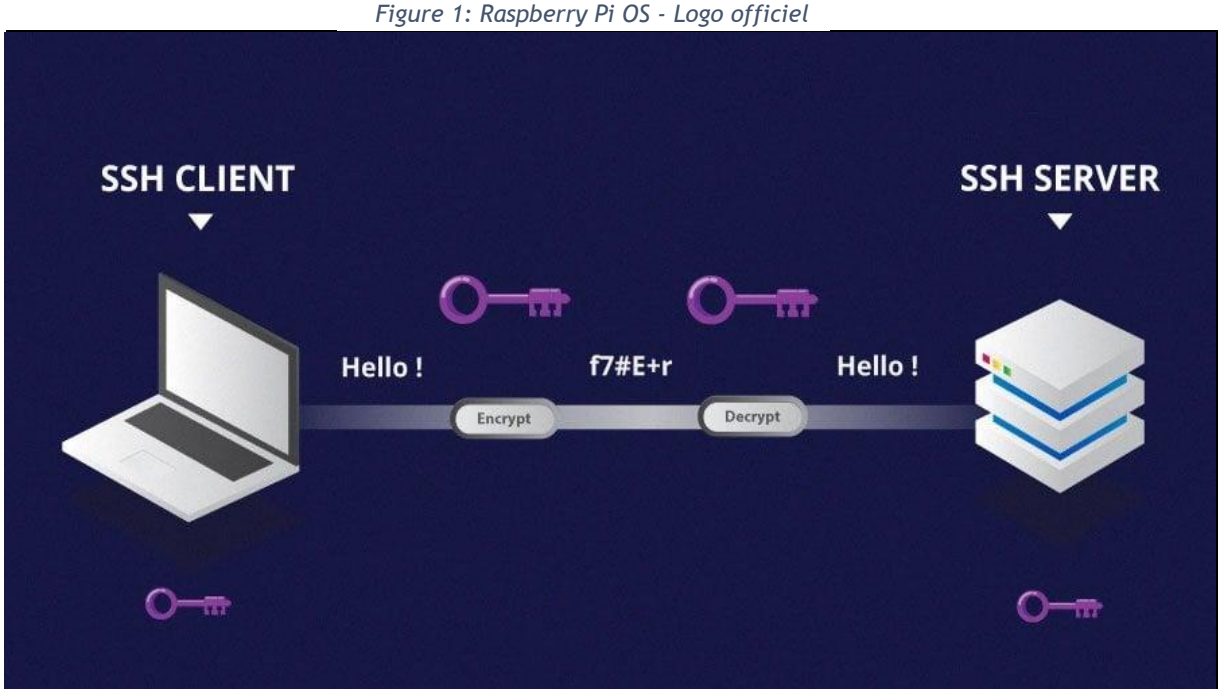
Hes·so VALAIS WALLIS
Haute Ecole de Gestion
Hochschule für Wirtschaft

Filière Informatique de gestion
Studiengang Wirtschaftsinformatik

Projet intermédiaire Raspberry Pi

Configurations réseaux et accès à distance

Figure 1: Raspberry Pi OS - Logo officiel



Source : (https://www.hostinger.com/, s.d.)

Etudiant : Dasek Joiakim

Professeurs : Barmaz Xavier, Russo David

Date de rendu : 13.12.2022

Résumé

Dans ce document nous allons effectuer des manipulations axées en ligne de commande de préférence, bien que la partie configuration et manipulation en mode graphique est tout de même démontrée. Nous allons apprendre à configurer le réseau ainsi que les services qui y sont associés pour le bon fonctionnement des étapes.

Nous découvrons d'une part le contrôle à distance du terminal, ses différentes méthodes d'authentification comme la génération de clés.

Ensuite, nous allons nous connecter au Raspberry Pi pour obtenir l'interface graphique et contrôler celui-ci comme s'il était présent localement.

Enfin nous terminons par installer un système de fenêtrage appelé « X11 » afin d'apprendre une alternative complémentaire.

Ce document contient un exercice final qui permet de mettre en pratique les connaissances acquises.

Mots clés : SSH, VNC, X11, Deamon, Bash, Batch, Xming, PuTTY

Table des matières

Liste des figures

Liste des abréviations

- GUI — Graphical User Interface, interface utilisateur graphique
- Wi-Fi — Wireless Fidelity pour fidélité sans fil
- GNU-Linux — C'est une famille de système d'exploitation Open Source de type Unix
- VNC — Virtual Network Computing, permet de virtualiser l'interface graphique d'un système d'exploitation
- X11 — Système de fenêtrage d'application
- CLI — Command Line Interface, Interface en ligne de commande
- GUI — Graphical User Interface, Interface graphique pour utilisateur
- SSH — Secure Shell, c'est un protocole de communication au terminal sécurisé
- SCP — Secure Copy, permet de copier des fichier source à une destination utilisant « SSH »
- Daemon — Le nom donné pour les services, tâches ou processus en arrière-plan
- Prompt — Etat du terminal, en attente d'une entrée de l'utilisateur
- Shell — C'est un synonyme du terminal ou invité de commande
- HTTPS — Hyper-Text Transport Protocol
- SSL/TLS — Protocol de cryptage
- Passphrase — A la différence d'un mot de passe, il s'agit d'une phrase de passe
- Seed — Littéralement « graine », utilisé pour la génération algorithmique

- XMing — C’est un programme de serveur « X » pour Microsoft Windows
- VLC — C’est un lecteur multimédia et open-source
- Bash — C’est un interpréteur de commande sous « Unix » et dérivés
- Batch — C’est un langage de script sur Windows

1. Introduction

Tout d'abord pour configurer le réseau du Raspberry Pi nous allons partir du principe que le nano-ordinateur a bien le système d'exploitation « Raspberry Pi OS » installé, qu'il est bien alimenté, connecté au réseau « Wi-Fi » avec les dernières mises à jour selon le document « Projet intermédiaire Raspberry Pi - Installation et configuration du Raspberry Pi OS » ainsi qu'un moniteur, un clavier et une souris.

Deux façons s'offrent à nous pour la suite du document, soit nous allons effectuer toutes les étapes de manière graphique, soit en ligne de commande. L'interface graphique est parfois limitante en termes de personnalisation, configuration d'un système informatique, d'une application ou d'un service, on ne pourra pas exploiter au plein potentiel ceux-ci.

Etant donné que tout ce qui se fait en interface graphique peut varier entre version, système d'exploitation ou autres facteurs, en ligne de commande on utilisera souvent une syntaxe similaire et tout en profitant d'un maximum de liberté d'interaction avec les outils.

Le but du document est de pouvoir se connecter à distance au terminal du Raspberry Pi, de contrôler son interface graphique de deux manières différentes l'une par le biais d'un système de virtualisation appelé « Virtual Network Computing », l'autre via le protocole « X11 ». Pour arriver à ces buts nous allons donc commencer par configurer le réseau du Raspberry Pi afin que nous puissions communiquer avec celui-ci.

2. Configurations réseau du Raspberry Pi

Premièrement, pour implémenter une configuration réseau adapté, en ligne de commande, quelques conditions seraient agréables à intégrer. Nous voulons :

1. Forcer à ce que le Raspberry Pi démarre uniquement s'il est connecté à un réseau. C'est particulièrement important qu'il soit strictement dans une situation dont il est atteignable dans le cas où aucun réseau n'est disponible ou que nous l'avons éteint et qu'il n'est pas accessible en mode graphique (absence de moniteur ou de clavier/souris) ce qui le rendrait de toute manière inutile.
2. Configurer un nom d'hôte, donc un alias de l'adresse IP du Raspberry Pi afin de nous faciliter une résolution d'hôte en réseau local, en l'occurrence pour une connexion à distance.

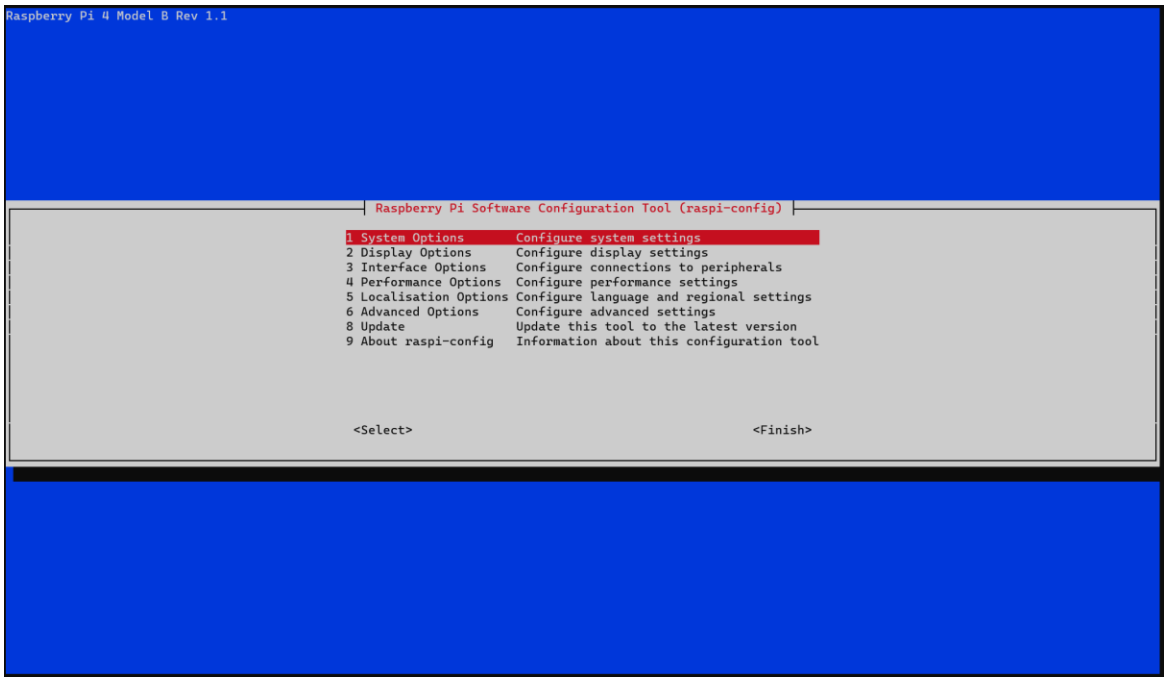
Nous allons commencer par ouvrir le terminal du Raspberry Pi et lancer la commande « raspi-config », il s'agit d'un utilitaire de configuration qui va nous permettre de définir des paramètres de base du système d'exploitation comme l'heure, la langue ainsi que des paramètres avancés. Pour lancer cet outil et que les modifications prennent effet il va falloir l'exécuter en mode administrateur avec le préfix « sudo » donc « sudo raspi-config » et appuyer sur la touche « Enter » :

Figure 2 - Outil raspi-config

Source : Auteur

Voici à quoi l'interface ressemble, un menu contextuel se présente à nous, subdivisant plusieurs domaines de configurations :

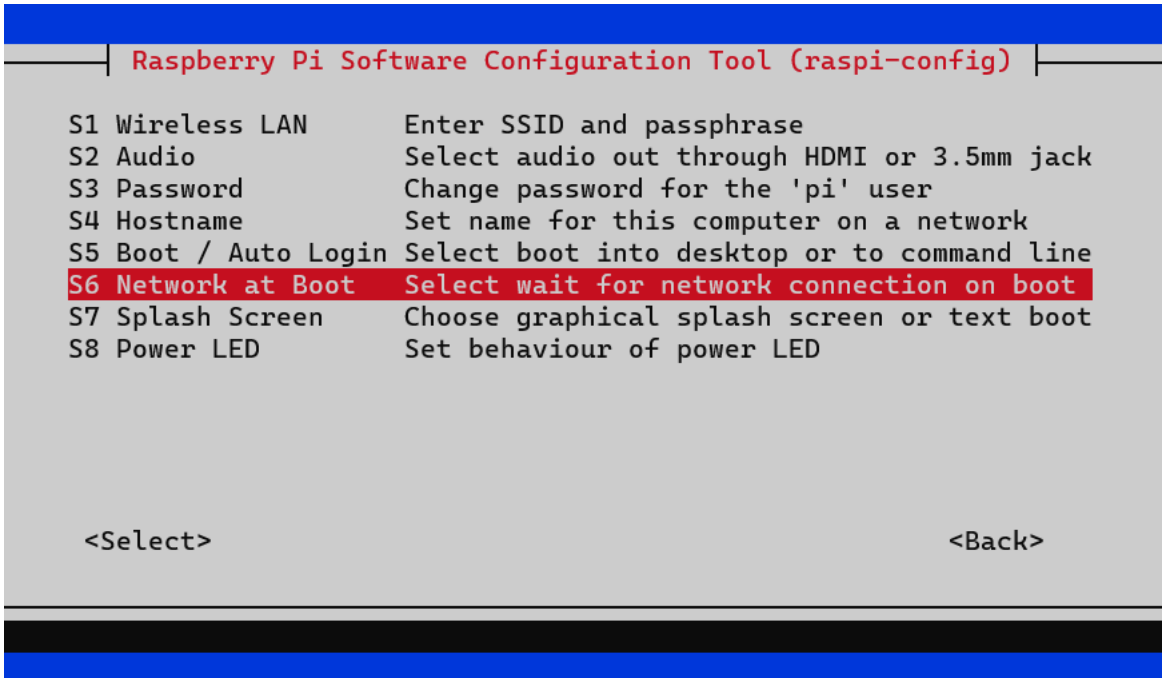
Figure 3 - Interface raspi-config



SOURCE : AUTEUR

Pour définir que le Raspberry Pi doive se lancer uniquement si une connexion réseau est active, nous allons devoir procéder comme suit. Tout d'abord nous devons sélectionner la première option du menu, ce qui nous mènera à ceci :

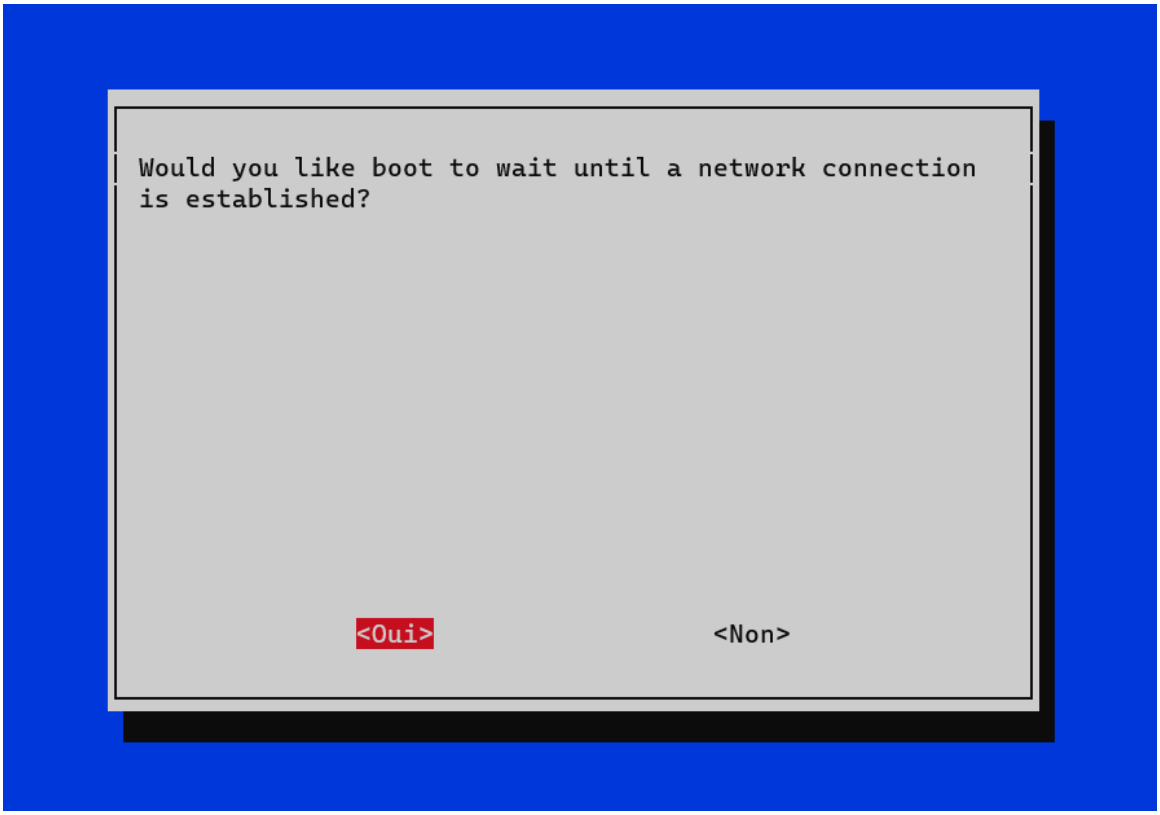
Figure 4 - Réseau au démarrage



SOURCE : AUTEUR

Ensuite sélectionner la sixième option « S6 Network at Boot » et appuyer sur la touche « Enter » puis nous devons confirmer notre choix, nous pouvons déplacer notre sélection avec la touche « Tab » puis la touche « Enter » pour confirmer la sélection.

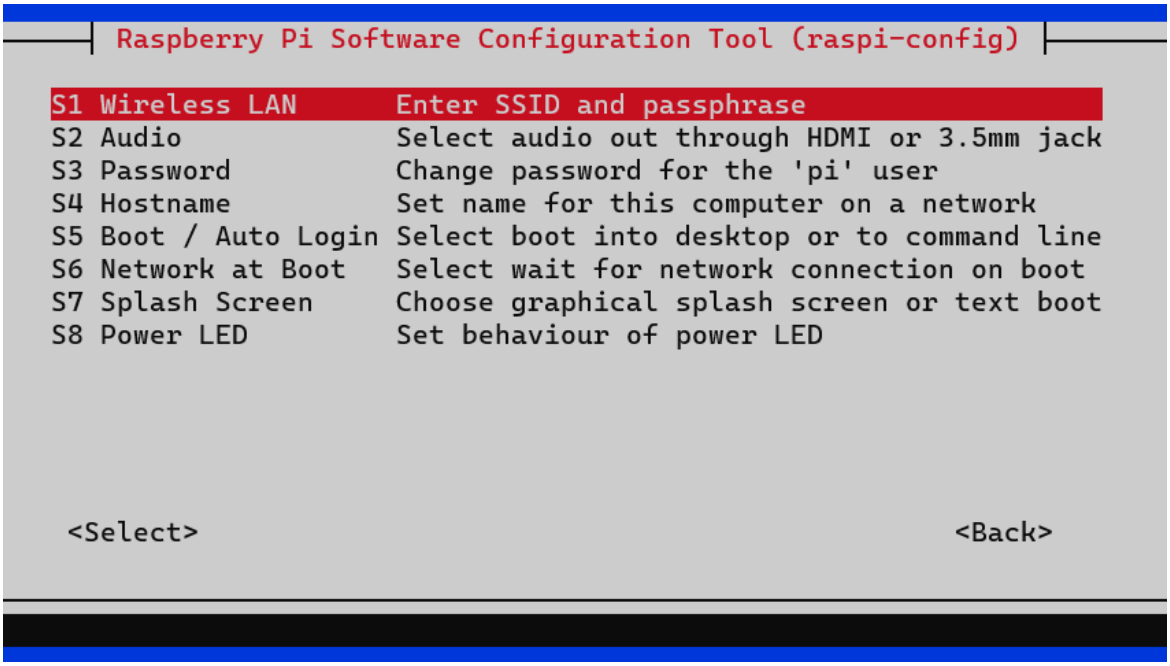
Figure 5 - Confirmation pour pour le réseau au démarrage



SOURCE : AUTEUR

Nous pouvons passer à la suivante étape, définir un alias, un nom d'hôte au Raspberry Pi. Pour ce faire il faut aller dans la même première option que précédemment par rapport au menu contextuel. Nous devons sélectionner cette fois-ci l'option « S4 Hostname » et appuyer sur la touche « Enter » :

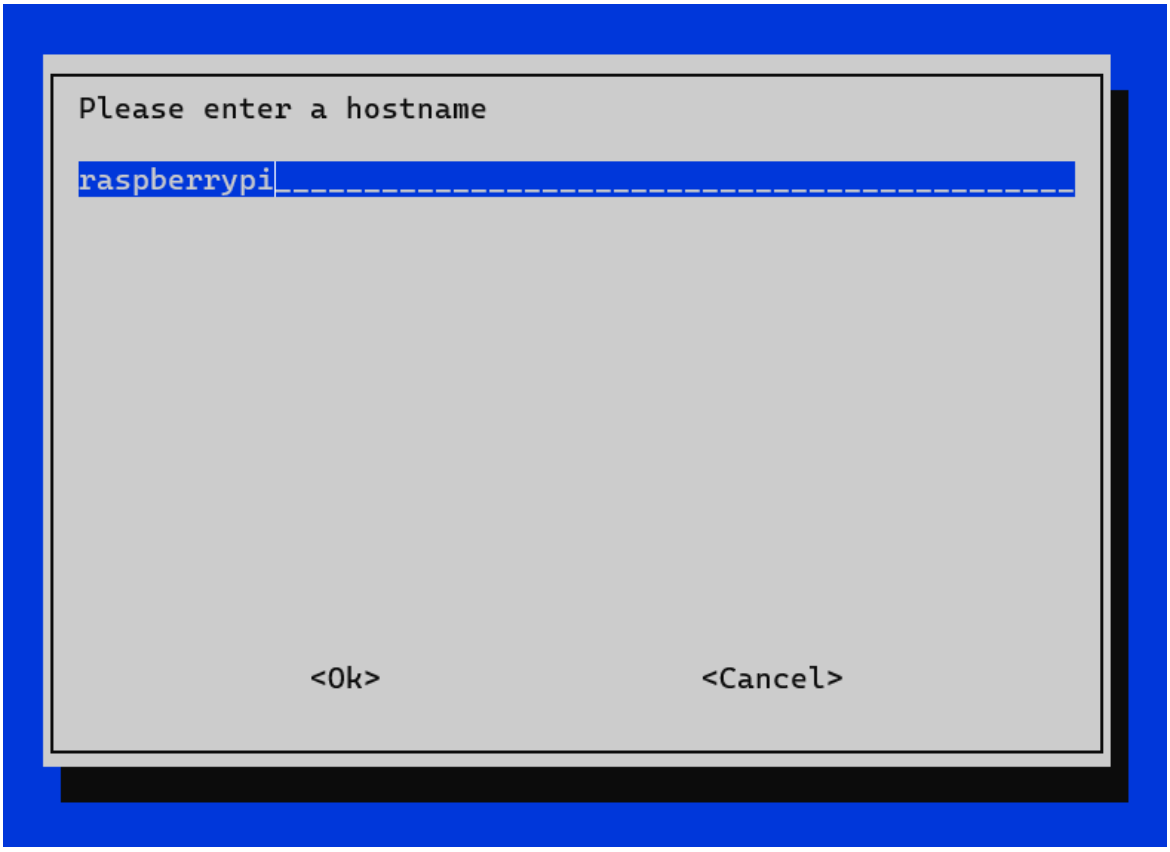
Figure 6 - Configuration de l'alias



SOURCE : AUTEUR

Un champ nous propose de définir le nom d'hôte, nous allons pour la suite de ce document, utiliser le nom par défaut qui décrit bien l'hôte en question mais nous pouvons le modifier à souhait. Appuyer sur la touche « Enter » après avoir écrit :

Figure 7 - Confirmation du nom d'hôte



SOURCE : AUTEUR

Nous devons bien entendu avoir une connexion à un réseau et qu'après le redémarrage, ce même réseau soit atteignable sinon le Raspberry Pi ne démarrera pas ! Donc pour que les modifications prennent effets nous devons redémarrer le système d'exploitation via cette commande :

Figure 8 . Commande de redémarrage

SOURCE : AUTEUR

3. Méthodes de gestion du Raspberry Pi à distance

1. Connexion à distance via le Secure Shell

Le Secure Shell, appelé aussi « SSH » est un protocole réseau utilisé pour établir une connexion sécurisée entre un ordinateur local et un ordinateur à distance. Il permet de se connecter au terminal du serveur « SSH » à distance en tant qu'utilisateur spécifique. Dans notre cas il s'agirait de se connecter en tant qu'utilisateur « pi » pour exécuter des commandes sur le Raspberry Pi. Il permet aussi de transférer des fichiers entre deux serveurs « SSH » utilisant le programme « Secure Copy », de son acronyme « SCP ».

Pour effectuer une connexion à distance quatre éléments sont indispensables :

1. Un ordinateur local disposant d'un client « SSH ».
2. Un ordinateur distant avec un serveur « SSH » dans notre cas le Raspberry Pi.
3. Une adresse IP ou d'un nom d'hôte en l'occurrence « raspberrypi.local ».
4. Un nom d'utilisateur et un mot de passe ou d'une clé publique et sa clé privée associée.

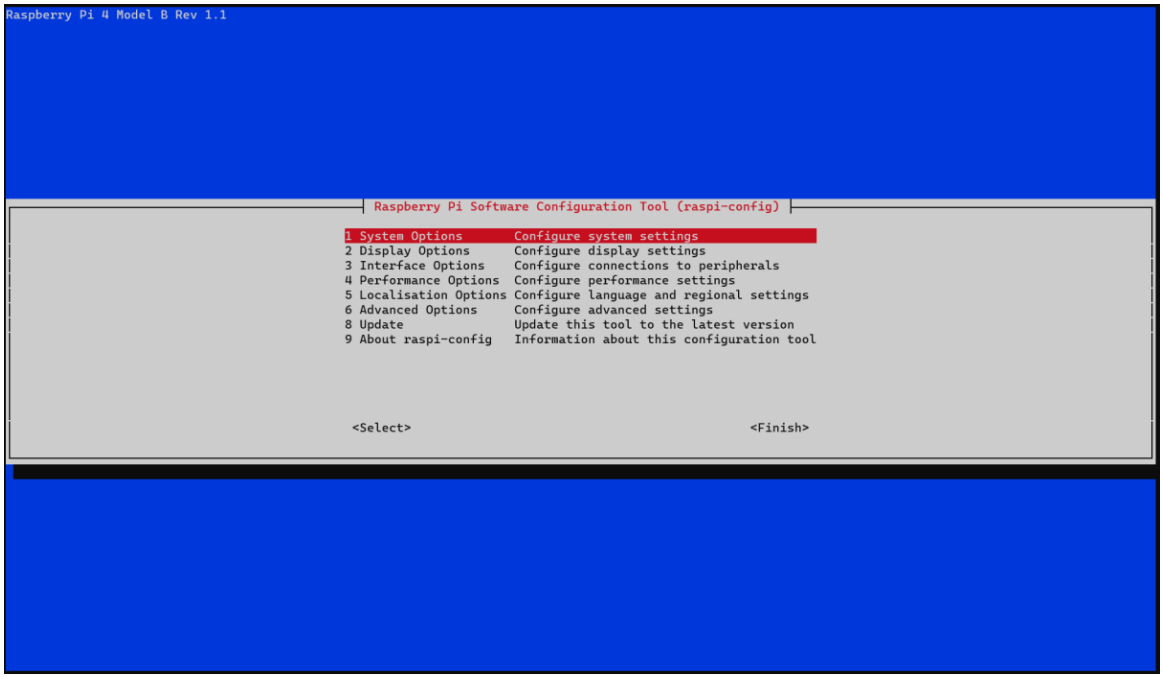
Nous allons commencer par configurer le serveur « SSH » sur le Raspberry Pi puis effectuer une connexion à distance depuis un ordinateur local. Nous allons utiliser le même outil que précédemment pour effectuer une connexion à distance. Un point important, le serveur « SSH » est une dépendance déjà préinstallée sur le système d'exploitation du Raspberry Pi. Il suffit de l'activer selon les étapes suivantes :

Figure 9 - Commande pour lancer raspi-config

SOURCE : AUTEUR

Dans l'image suivante, nous devons sélectionner les options d'interface, « Interface options » :

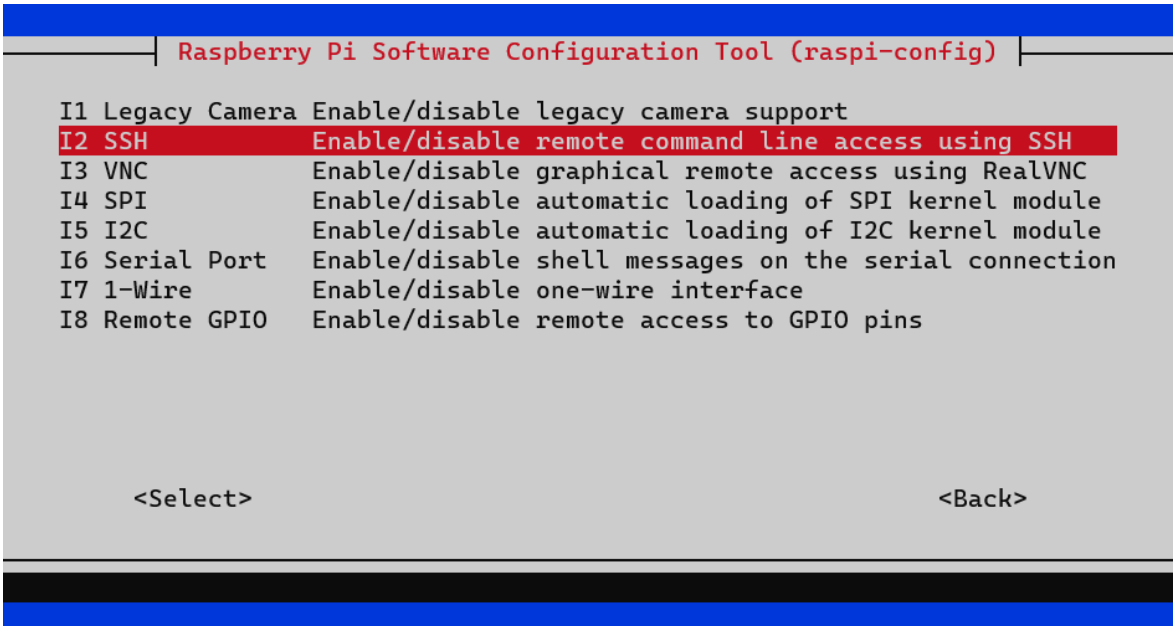
Figure 10 - Interface du raspi-config



SOURCE : AUTEUR

Ensuite nous devons activer la deuxième option qui permet d'activer ou désactiver le contrôle d'accès à la ligne de commande utilisant « SSH » :

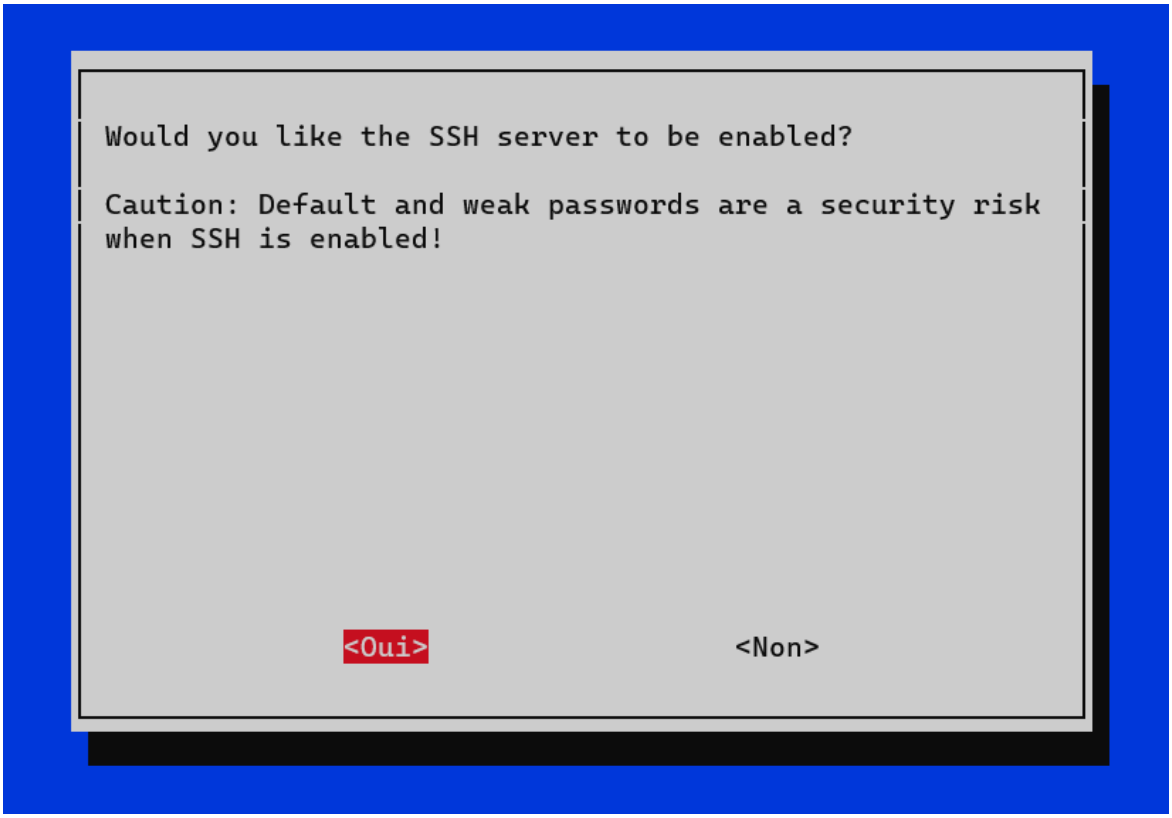
Figure 11 - Activation du SSH



SOURCE : AUTEUR

L'outil nous demande si nous sommes sûr de vouloir activer le serveur « SSH » dès le démarrage du système. Il prévient avec une grande attention parce que de base si vous n'avez pas modifier le mot de passe de l'utilisateur par défaut, toute personne ayant un minimum de connaissance sur ce système d'exploitation connaît le nom d'utilisateur ainsi que le mot de passe ! Il est donc important d'avoir changé au préalable le mot de passe. Nous confirmons en sélectionnant « Oui » :

Figure 12 - Confirmation de l'activation du service SSH

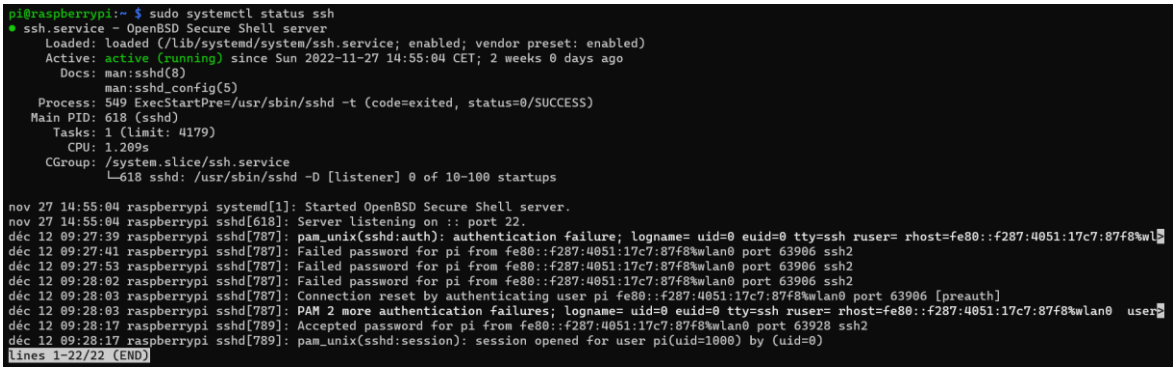


SOURCE : AUTEUR

Nous allons tout d'abord apprendre ce qu'est un service pour comprendre comment se lance le serveur « SSH » dès le démarrage :

Un service dans un système d'exploitation est un programme en cours d'exécution en arrière-plan. Il fournit une fonctionnalité spécifique à un autre programme ou un utilisateur. Un service est généralement démarré automatiquement lors du démarrage du système d'exploitation et peut être redémarré ou arrêté manuellement par ligne de commande ou en mode graphique. Dans les systèmes « GNU/Linux », on les appelle des « Daemon » alors que sous « Windows » on appelle cela des services. Le gestionnaire de « Daemon » de Linux est « systemctl ». Dans l'exemple précédent avec « raspi-config », l'outil exécute cette commande : « sudo systemctl enable ssh » pour activer le serveur « SSH » dès le démarrage du Raspberry Pi et ainsi vous pouvez vérifier le statut actuel du service SSH avec cette commande « sudo systemctl status ssh » :

Figure 13 - Systctl, gestionnaire de service



SOURCE : AUTEUR

Dans le cas où nous souhaitons modifier la configuration de manière plus avancée nous pouvons accéder à celle-ci via le terminal en utilisant cette commande : « sudo nano /etc/ssh/sshd_config », nous allons y revenir dessus dans la suite du document.

Pour faire une rétrospective des éléments précédents, nous avons un serveur « SSH » préinstallé sur le Raspberry Pi, nous l'avons activé en utilisant « raspi-config » ou en ligne de commande et vu où configurer de manière plus spécifique le fichier de configuration. Si le serveur « SSH » n'était pas existant sur le système d'exploitation, il aurait fallu exécuter ces commandes : « sudo apt update » permettant de mettre à jour le fichier « liste de dépendance système » puis « sudo apt install openssh-server » pour installer le serveur « SSH » sur le système.

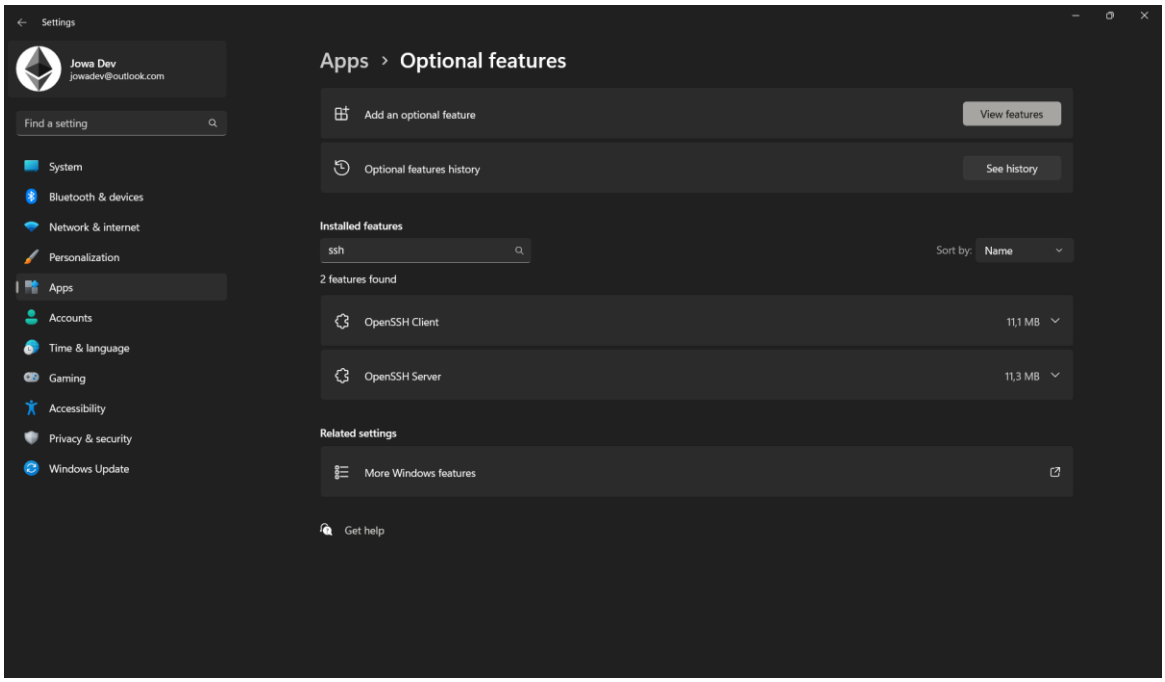
Nous avons donc en ce moment, la possibilité de nous connecter au serveur « SSH » par le biais d'un client « SSH » en utilisant la méthode « User – Password » pour l'authentification et non la méthode clé publique et clé privée que nous allons voir plus tard.

Passons du côté client, il existe plusieurs clients pour se connecter au terminal du Raspberry Pi :

1. « PuTTY » est un logiciel client qui intègre plusieurs protocoles comme « File Transfert Protocol » ou encore « Telnet » anciennement « SSH » mais sans le côté sécurisé. Ce qui nous intéresse c'est qu'il intègre le protocole Secure Shell.
2. Nous pouvons aussi intégrer le client « SSH » dans les fonctionnalités Windows pour qu'il soit disponible dans l'invité de commande Windows mais aussi dans le PowerShell. L'invité de commande et le PowerShell sont tous deux des terminaux à la différence que le PowerShell est le successeur de l'invité de commande et donc intègre bien des avantages.

Nous allons utiliser la deuxième option parce que nous utiliserons « PuTTY » plus tard dans le document. Pour installer le client « SSH » sous Windows nous allons devoir occuper notre ordinateur local avec lequel nous aurons un contrôle à distance sur le Raspberry Pi et aller sur notre système d'exploitation Windows. Il nous suffira d'appuyer sur le menu « Démarrer » et écrire « Fonctionnalités facultatives » ce qui nous mènera à cette fenêtre :

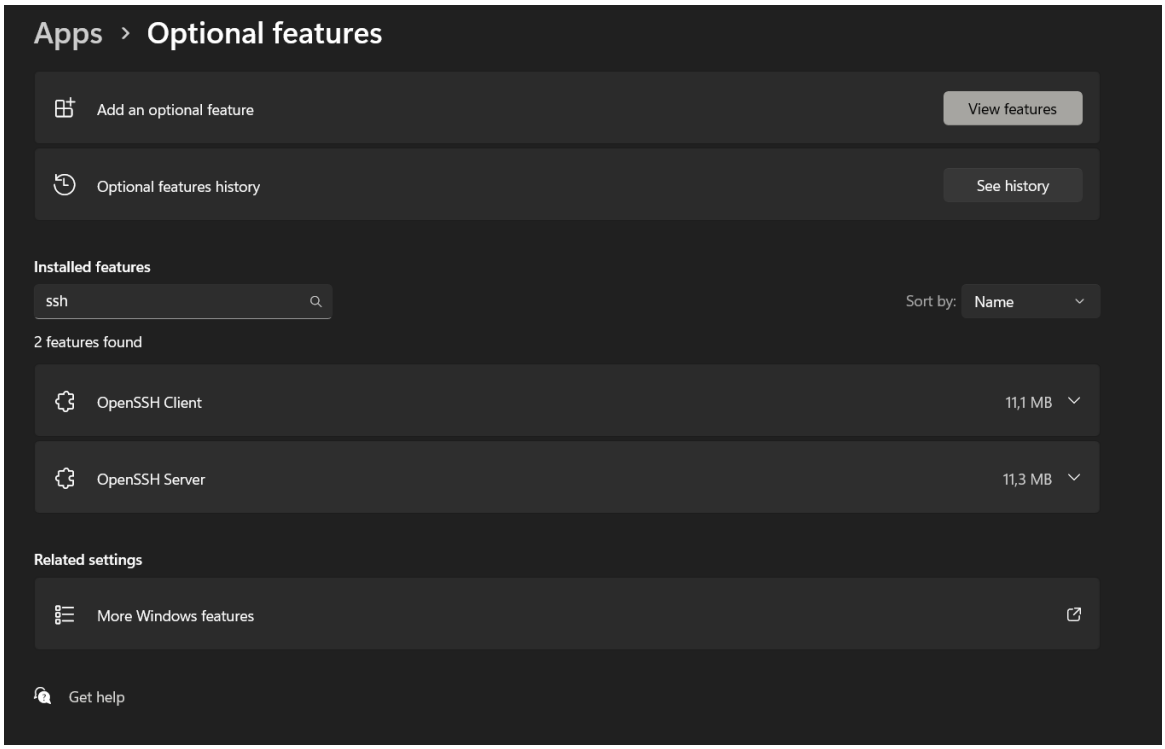
Figure 14 - Paramètre Windows, fonctionnalités optionnelles



SOURCE : AUTEUR

Ensuite nous devons appuyer sur « View features » pour « Voir les fonctionnalités » et une nouvelle petite fenêtre va apparaître avec une barre de recherche. Dans le champe nous allons entrer « ssh client » et la liste va se mettre à jour, il suffira de sélectionner « OpenSSH Client » et appuyer sur « Next » pour installer. Une fois installer il devrait être présent dans les fonctionnalités installées :

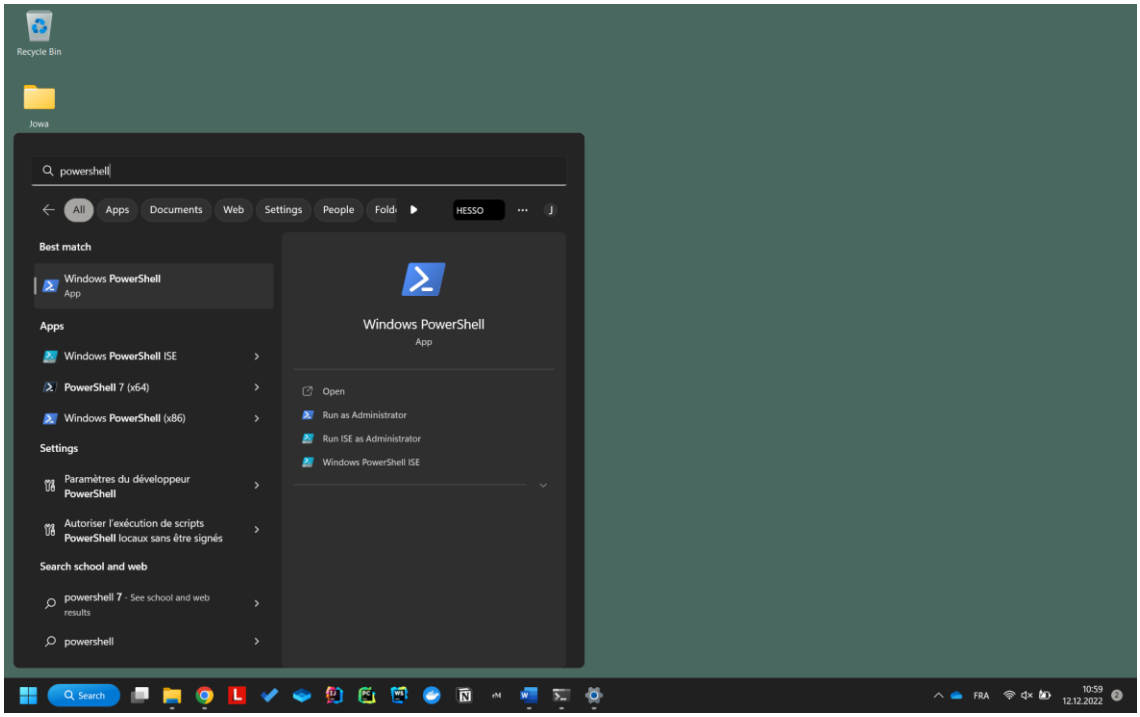
Figure 15 - Ajout de fonctionnalité Windows



SOURCE : AUTEUR

Une fois installé nous pouvons lancer le terminal « PowerShell » pour nous connecter à distance au Raspberry Pi. Voici comment procéder, premièrement ouvrons le menu démarrer et recherchons « PowerShell » comme suit :

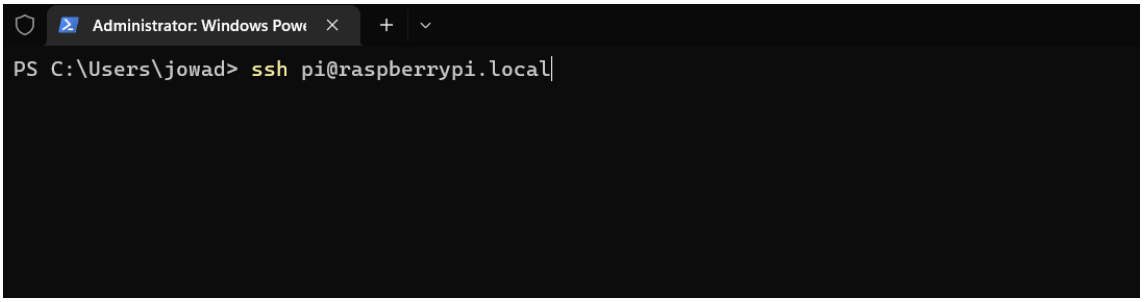
Figure 16 - Lancement du powerShell



SOURCE : AUTEUR

Nous pouvons le lancer, l'interface est similaire à l'invité de commande par défaut de Windows. Nous allons pouvoir entrer la commande suivante pour utiliser le client « SSH » comme suit dans le « prompt », pour information le « prompt » est un état du terminal qui signifie qu'il est prêt à recevoir une commande : « ssh pi@raspberrypi.local » :

Figure 17 - Connexion en SSH



SOURCE : AUTEUR

La commande est structurée de la manière suivante, « ssh » permet d'utiliser le client, « pi » permet de spécifier que l'on prend le contrôle de l'utilisateur « pi » du Raspberry Pi donc son rôle, droit et accès. Le « @ » signifie littéralement « chez » pour spécifier, je veux prendre le contrôle de l'hôte « raspberrypi.local » en tant qu'utilisateur « pi ». Traduisible par « Je suis « pi » chez « raspberrypi.local ». Nous pouvons appuyer sur la touche « Enter ». La réponse du serveur « SSH » est l'authentification par mot de passe :

Figure 18 - Demande de mot de passe par le serveur

SOURCE : AUTEUR

Nous avons donc ainsi accès au « Shell », terminal, du Raspberry Pi à distance avec quelques informations sur le serveur auquel nous avons accès :

Figure 19 - Connexion SSH établie

SOURCE : AUTEUR

Nous avons réussi à nous connecter au Raspberry Pi avec un contrôle du terminal à distance avec la méthode « Utilisateur / Mot de passe ».

Pour effectuer la méthode de la paire, clé publique et clé privée, il faut tout d'abord comprendre ce que sont ces clés et la différence avec la méthode d'authentification classique.

Le système de clé publique et clé privée est d'ailleurs utilisés dans beaucoup d'autres domaines comme les portefeuilles électronique du domaine des chaînes de blocks « Blockchain » ou encore pour le protocole « HTTPS » venant du « SSL/TLS ». Elles permettent par exemple de s'authentifier, sécuriser et même gérer des connexions.

« Ce sont deux parties d'un système de cryptage asymétrique utilisé en l'occurrence pour authentifier les utilisateurs sur un serveur « SSH ». La clé publique est partagée publiquement et peut donc être diffusée de manière large, avec le serveur SSH par exemple, tandis que la clé privée est gardée secrète par l'utilisateur, le client.

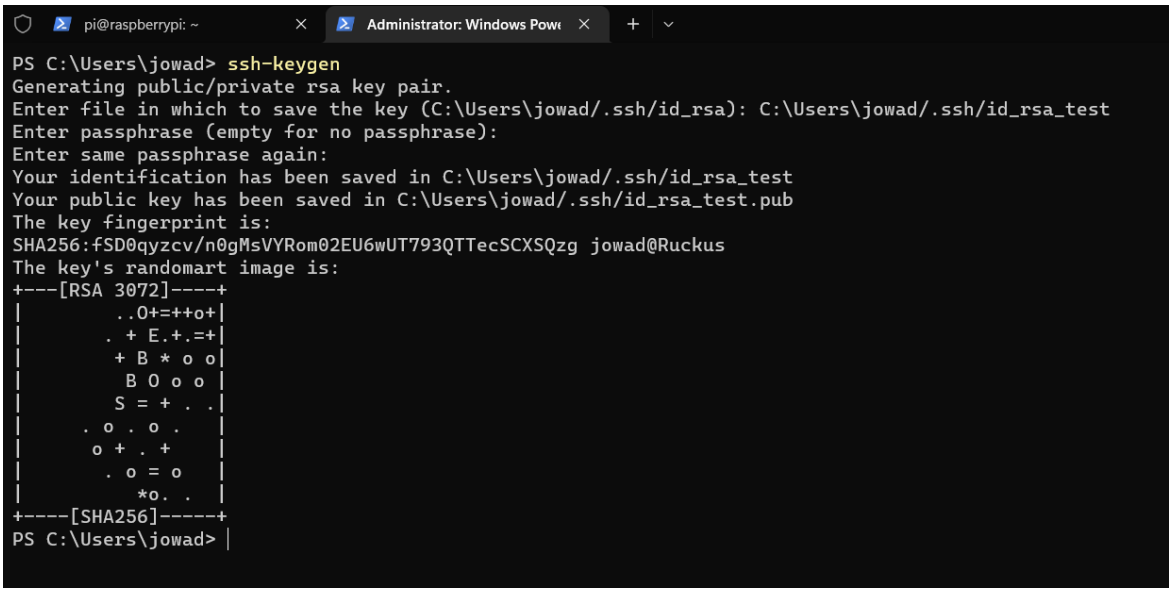
Lorsqu'un utilisateur souhaite se connecter au serveur, il utilise sa clé privée pour authentifier son identité au serveur, qui vérifie alors l'authenticité de la clé en utilisant la clé publique correspondante. Cette méthode est considérée comme plus sécurisée qu'une simple authentification par mot de passe, car elle empêche les attaques de type "brute force" où un utilisateur malveillant essaie de deviner le mot de passe. Cette méthode permet aussi d'éviter la redondance du mot de passe. » (Authentification-ssh-par-paire-de-cles-priee-publique, s.d.)

Nous allons comme décrit précédemment devoir partager la clé publique venant du client au serveur « SSH » pour qu'il puisse nous authentifier et vérifier que nous somme bien légitime pour effectuer une connexion. Pour ce faire, nous allons devoir générer les clés sur Windows, voici comment procéder à cela :

Nous devons utiliser l'outil « ssh-keygen » dans le terminal, disponible depuis l'installation que l'on a fait de « Open-SSH Client », ensuite il va nous demander sous quel nom nous voudrons l'enregistrer vous pouvez laisser le nommage par défaut et enfin une « passphrase » est demandé.

A la différence d'un mot de passe, une phrase de passe à un niveau plus « élevé » de sécurité parce qu'il peut inclure des espaces comme caractère et peut avoir une plus grande longueur. Cette « passphrase » est utile pour protéger la clé privée, si quelqu'un vole votre clé privée, il devra aussi connaître la « passphrase » pour pouvoir l'utiliser ! A noter qu'elle n'est utile qu'à rajouter une couche supérieure de sécurité et elle est non pas utilisé comme « graine » aussi appelé « seed » pour la génération des clés ! Voici à quoi toutes ces étapes ressemblent :

Figure 20 - Génération de clé SSH

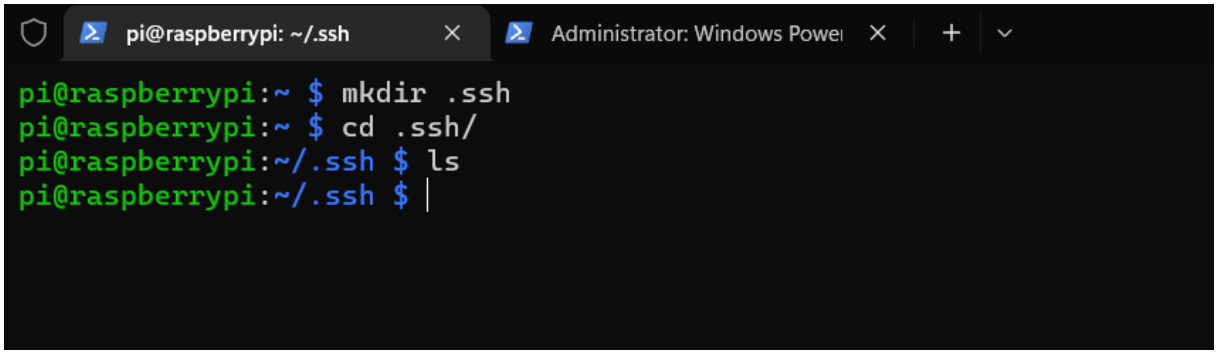


Source : Auteur

Les deux clés ont été générées dans notre dossier utilisateur, dans un dossier caché « .ssh », les deux fichiers, l'un la clé publique avec extension « .pub » et la clé privée sans extension. Il ne faut pas les confondre, la clé publique sera le seul fichier pouvant être partagé !

Nous allons devoir accueillir la clé publique que le client va par la suite envoyer au serveur « SSH ». Pour cela nous pouvons aller sur le Raspberry Pi et ouvrir le terminal afin de créer un dossier caché, de même type que sur Windows, dans le dossier de l'utilisateur. Voici les commande, « mkdir » littéralement « Make Directory » donc créer un répertoire, « cd » pour « Change Directory » afin de changer de répertoire et « ls » pour lister les fichiers contenus dans le répertoire courant :

Figure 21 - Création de dossier sous Linux



SOURCE : AUTEUR

Nous allons devoir utiliser deux commande « Unix » tel que « chmod » et « chown ». Ce sont des commandes qui permettent respectivement de modifier les permissions d'un fichier ou d'un répertoire et de changer le propriétaire d'un fichier ou d'un répertoire. Elles sont souvent utilisées par des administrateurs systèmes. (configure-ssh-key-based-authentication-on-raspberry-pi, s.d.)

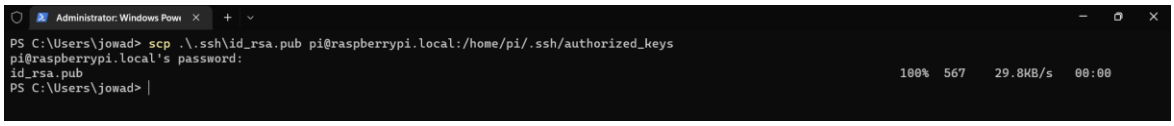
Tout d'abord « chown » littéralement « change owner » s'utilise de la manière suivante : « chown [OPTIONS] NOUVEAU_PROPRIETAIRE FICHIER_OU_REPERTOIRE ». La commande « chmod » littéralement « change mode » s'utilise de la manière suivante : « chmod [OPTIONS] PERMISSIONS FICHIER_OU_REPERTOIRE ». Il faut noter que l'utilisation de ces deux commandes nécessite des droits administrateurs. (linux-chmod-chown-change-file-permissions, s.d.)

Nous avons besoin de ces deux commandes pour que le serveur « SSH » installé sur le Raspberry Pi puisse accéder en tant qu'utilisateur « pi » et avec des droits d'accès permettant de lire le fichier « authorized_keys » qui va être présent dans le dossier « .ssh » du répertoire utilisateur.

Cela va se passer en deux temps. Premièrement nous allons copier la clé publique générée depuis l'ordinateur local Windows puis effectuer nos commandes :

Nous pouvons retourner sur la machine locale, sur Windows, pour effectuer le transfert de la clé publique utilisant « SSH ». La commande est « scp » pour « Secure Copy » suivit du « chemin absolue ou relatif de la source » et enfin suivi comme fin d'argument à la commande par « pi@raspberrypi.local:/home/pi/.ssh/authorized_keys » ce qui suit les deux points mentionne le chemin de destination du fichier source, on doit d'ailleurs le renommer comme sur l'exemple, au nom de « authorized_keys » pour qu'il soit reconnu par le serveur « SSH ». Le serveur nous répond « 100% » cela veut dire que le transfert s'est bien passé.

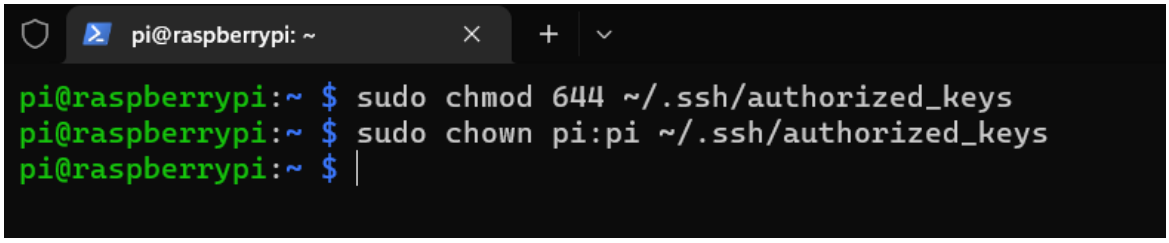
Figure 22 - Copie de la clé publique



SOURCE : AUTEUR

Nous devons retourner sur le Raspberry Pi et s'assurer que le propriétaire du dossier soit bien « pi » et qu'il ait les droits de lecture :

Figure 23 - Changement des droits et propriétaire



SOURCE : AUTEUR

La première commande sur le fichier « authorized_keys » définie que le propriétaire obtient les droits de lecture et d'écriture et que les autres utilisateurs ont uniquement un droit de lecture. La deuxième commande définie l'utilisateur en tant que propriétaire du fichier.

Nous pouvons dès à présent nous connecter sans devoir utiliser un mot de passe sauf pour prouver que la clé privée nous appartient bien ! Attention à noter que ce n'est pas réellement le serveur qui demande la « passphrase » mais bel et bien notre propre ordinateur local. En réalité quand la commande est exécutée dans l'image qui suit, le serveur va envoyer un message, une chaîne de caractère aléatoire que l'on appelle un « défi ».

Pour déchiffrer ce message on va devoir utiliser la « passphrase » pour déverrouiller la clé privée, si elle est correcte, on déchiffre le « défi » et on envoie une réponse correcte au serveur « SSH » qui lui sait à ce moment que l'on est le propriétaire de la clé privée.

Donc deux mesures de sécurité sont en place. La première est que l'on doit déposer sur le serveur « SSH », la clé publique et la deuxième est que l'on doit prouver que l'on est le propriétaire de la clé privée ! (what-are-ssh-keys, s.d.)

Figure 24 - Demande de connexion SSH avec clé

SOURCE : AUTEUR

Voilà, nous avons effectué les deux méthodes de connexion à distance !

2. Connexion à distance avec VNCViewer

« VNC » pour « Virtual Network Computing » est un logiciel nous permet de nous connecteur à un ordinateur à distance et d'interagir avec lui comme si celui-ci était devant nous. Nous aurons donc une interface graphique face à nous et pourrons utiliser tous les périphériques externes comme la souris, le clavier, les haut-parleurs etc… Nous aurons la possibilité d'exécuter des applications comme s'ils étaient présent localement. Les cas d'utilisations seraient, de la maintenance à distance ou encore du support technique. (Qu'est ce qu'un VNCviewer, s.d.)

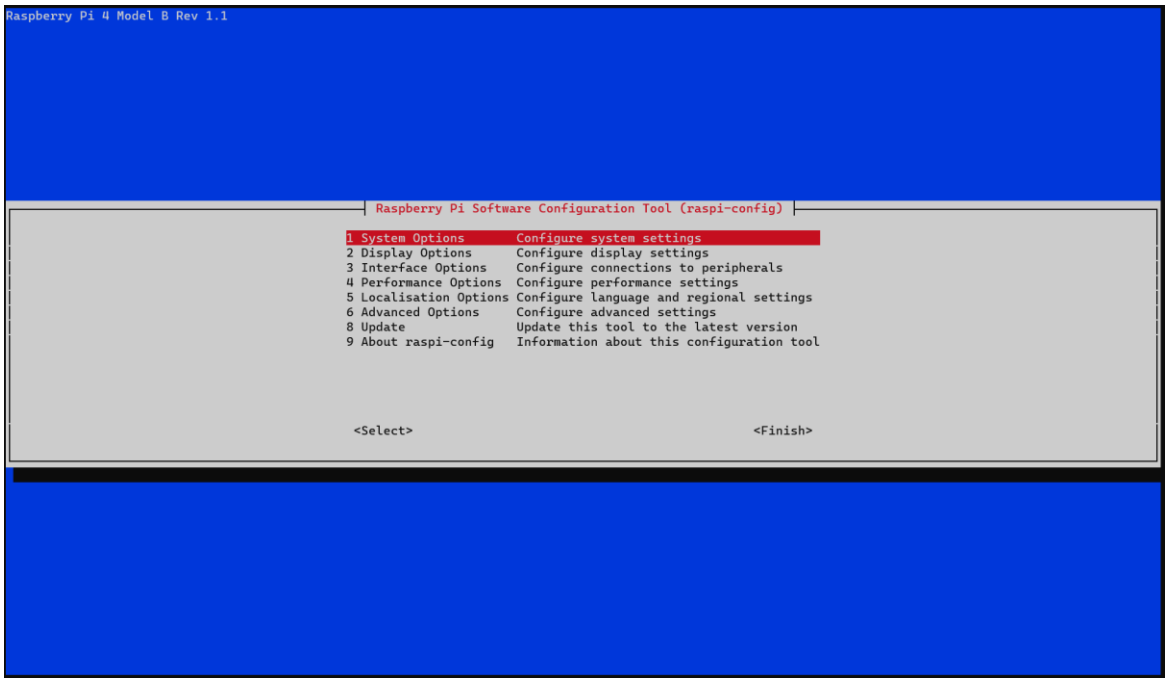
Nous allons pouvoir nous connecter avec une interface graphique à distance avec « VNCViewer » il s'agit d'un client « VNC » qui permet de faire tout ce qui est décrit précédemment. Le serveur « VNC » est déjà préinstallé sur le système d'exploitation du Raspberry Pi mais nous allons tout de même, voir comment installer le serveur en ligne de commande.

Comme d'habitude pour installer un paquet la commande est la suivante : « sudo apt install realvnc-vnc-server » puis pour configurer le serveur il suffit d'entrer la commande « vncserver » dans le terminal. Il suffit de suivre la procédure à l'écran pour configurer un mot de passe, si rien n'est entré le mot de passe de l'utilisateur « pi » sera utilisé.

Pour lancer le service « VNC Server » dès le démarrage il suffit d'entrer la commande suivante : « sudo systemctl enable vncserver-virtuald.service ».

Pour effectuer cette étape avec l'outil « raspi-config », voici la procédure, exécuter la commande « sudo raspi-config » et aller dans l'option numéro trois du menu contextuel :

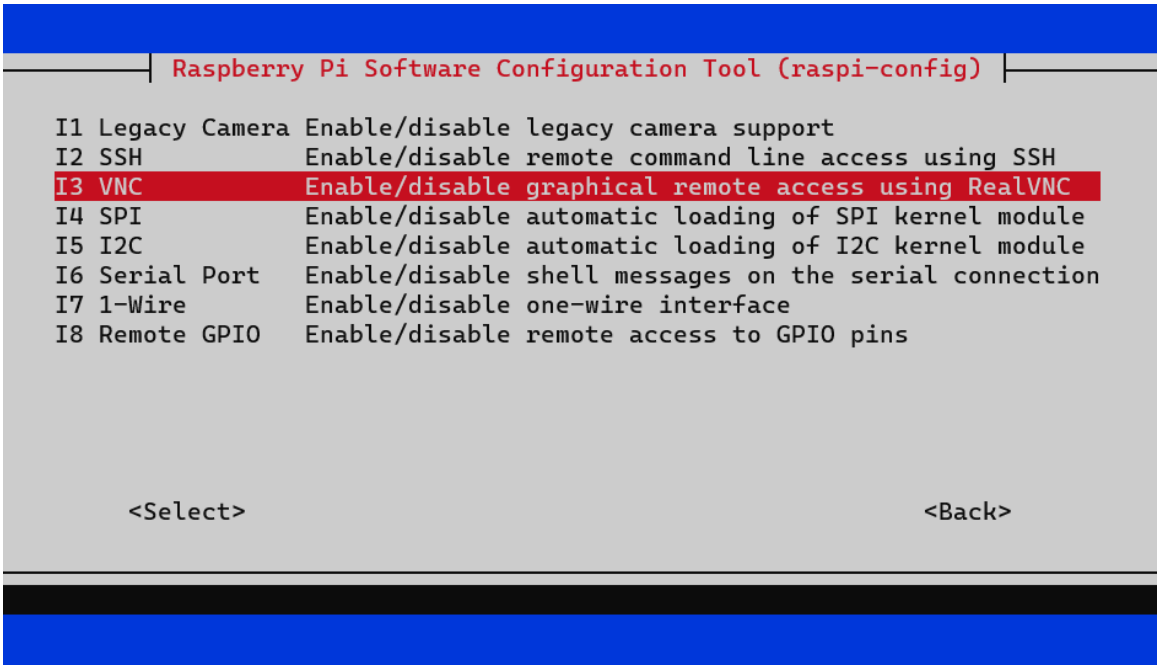
Figure 25 - Interface raspi-config



SOURCE : AUTEUR

Puis nous sélectionnons à nouveau la troisième option « activer/désactiver l'accès au contrôle graphique utilisant RealVNC » afin d'activer le « daemon » dès le démarrage du Raspberry Pi :

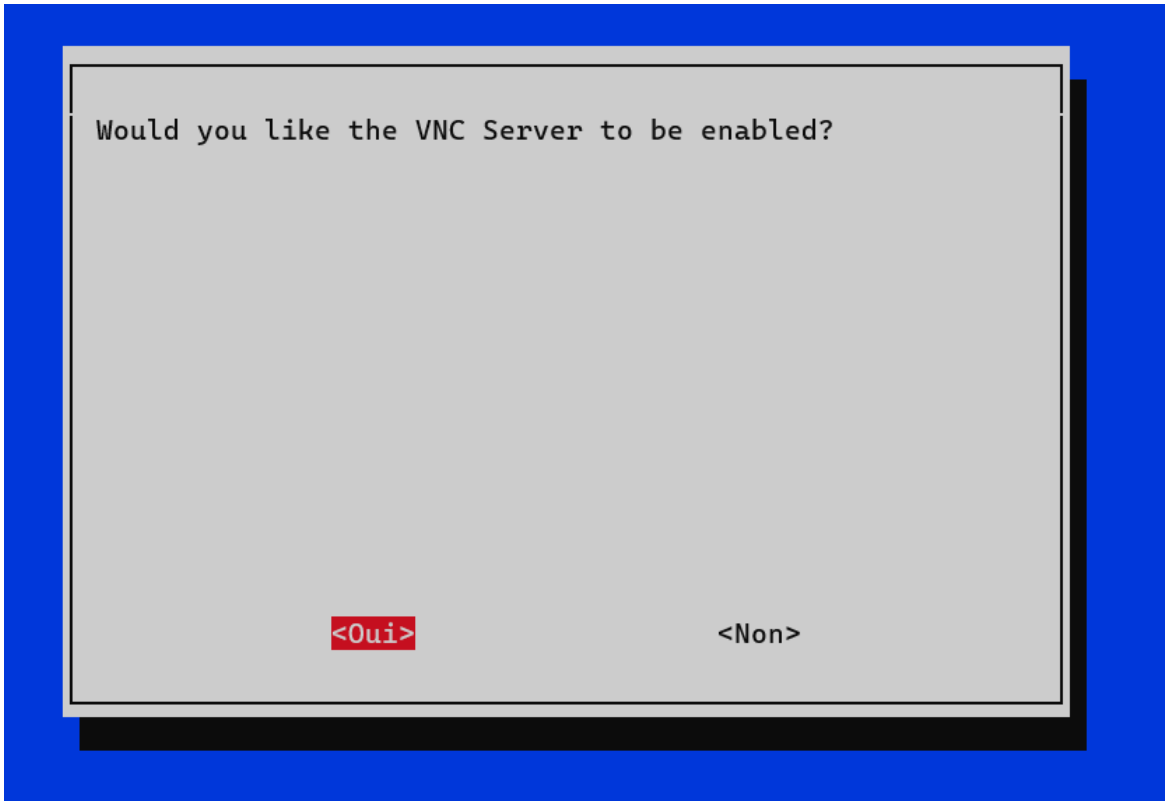
Figure 26 - Activation du service VNC au démarrage



SOURCE : AUTEUR

Il suffit de confirmer l'activation :

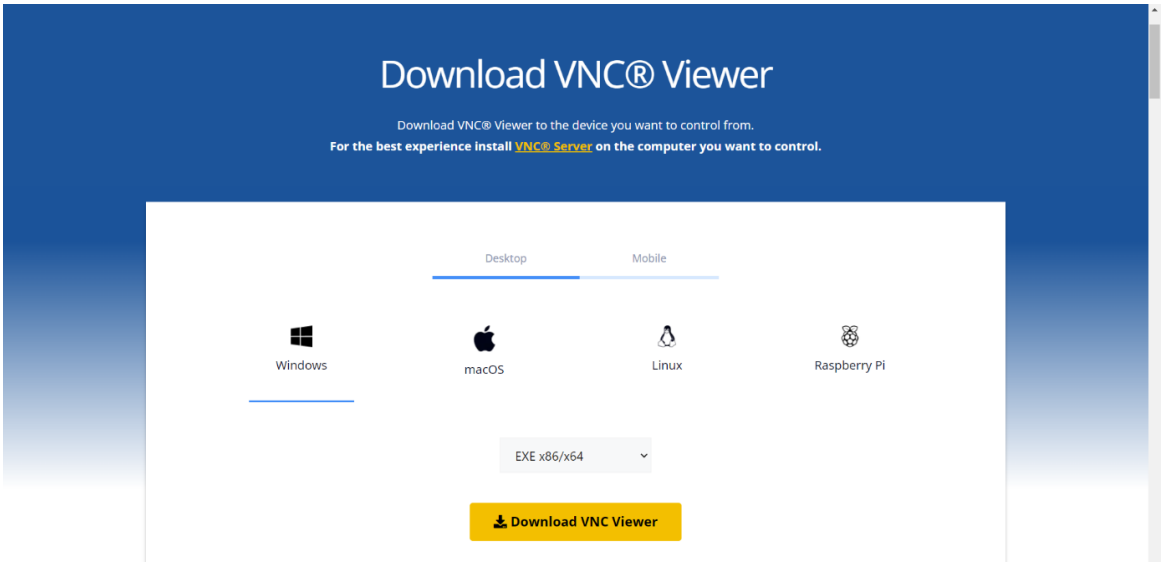
Figure 27 - Confirmation de l'activation du service



SOURCE : AUTEUR

Par la suite nous retournons vers notre ordinateur local et installons le logiciel client VNCViewer sur ce lien : https://www.realvnc.com/. Puis il suffit de suivre l'installation et enfin exécuter le programme.

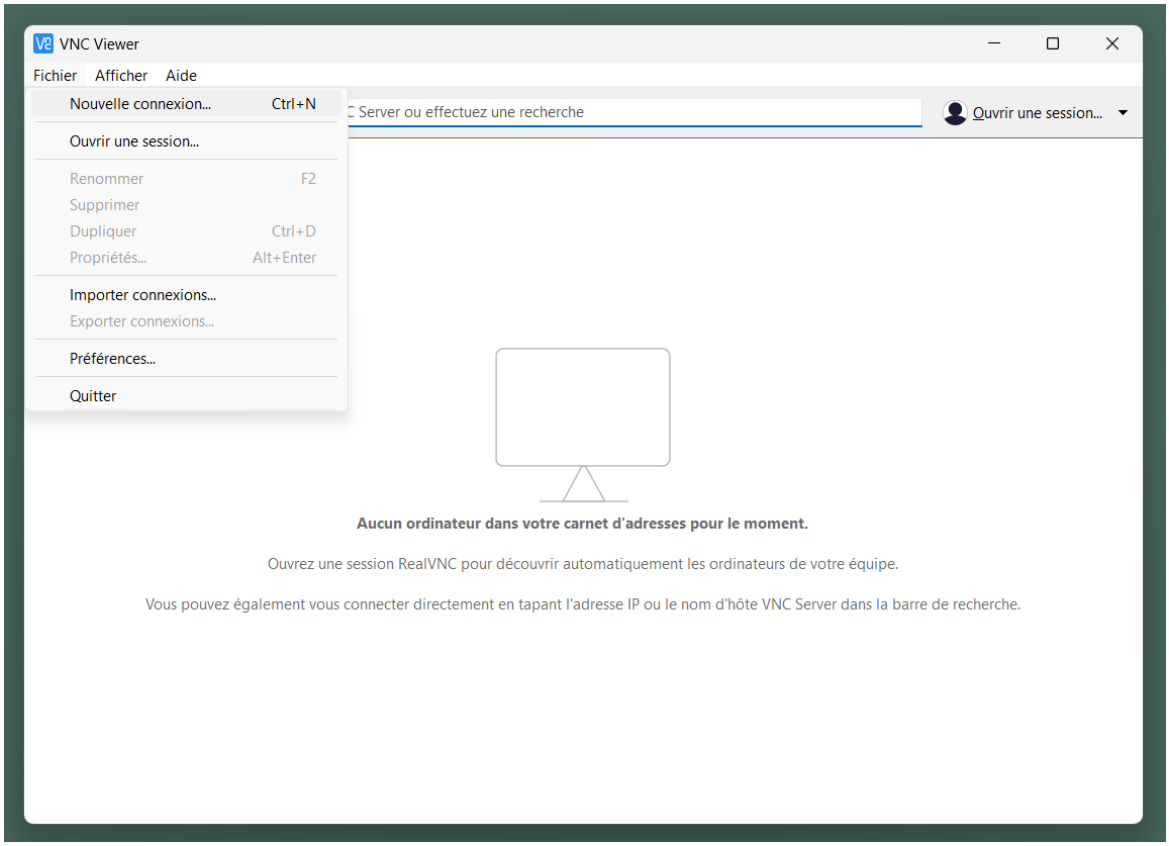
Figure 28 - Page de téléchargement de l'outil VNC Viewer



(https://www.realvnc.com/, s.d.)

L'interface est simpliste, aucune inscription n'est nécessaire pour utiliser le logiciel. Nous devons cliquer sur l'onglet fichier puis nouvelle connexions ou tout simplement le raccourci clavier « Ctrl + N » :

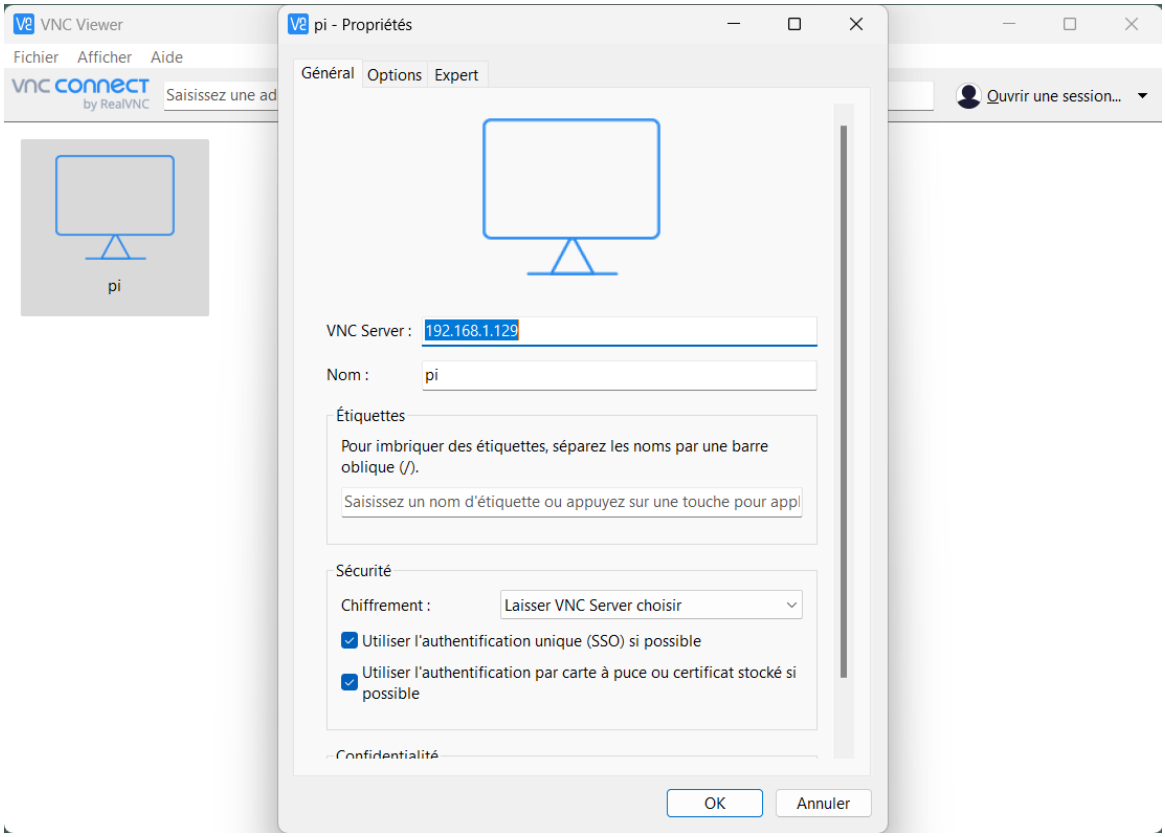
Figure 29 - Interface de VNC Viewer



SOURCE : AUTEUR

Une nouvelle fenêtre apparaît, nous allons configurer les propriétés d'un nouveau profil de connexion à distance. Le premier champ est l'adresse IP du serveur « VNC », nous pouvons aussi utiliser l'alias de l'hôte, en l'occurrence « raspberrypi.local ». Ensuite le nom de l'hôte pour le reconnaître parmi les différentes connexions que l'on pourrait avoir dans la liste des connexions sur l'interface principale. Nous pouvons donc le remplir comme bon nous semble et laisser le reste par défaut :

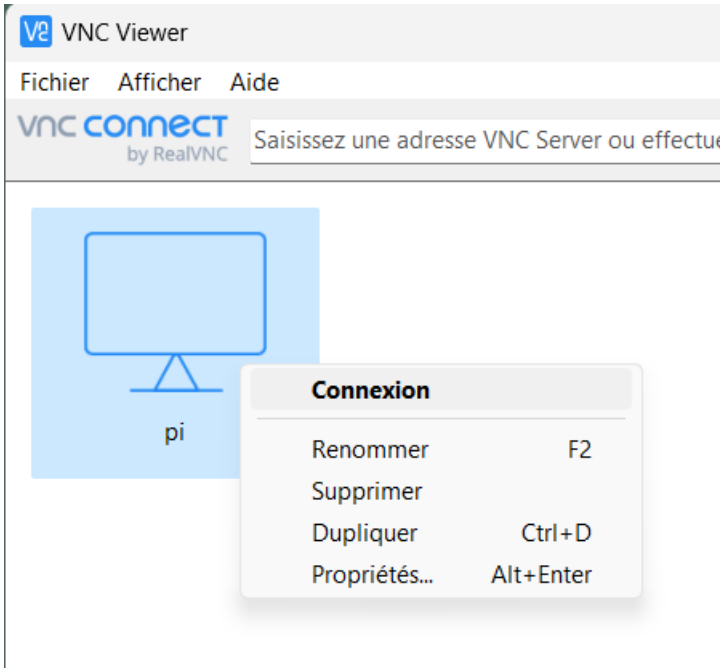
Figure 30 - Propriété d'une connexion VNC



Source : Auteur

Nous sommes prêts à effectuer une connexion à distance avec un double-clique ou clique-droit et « Connexion » :

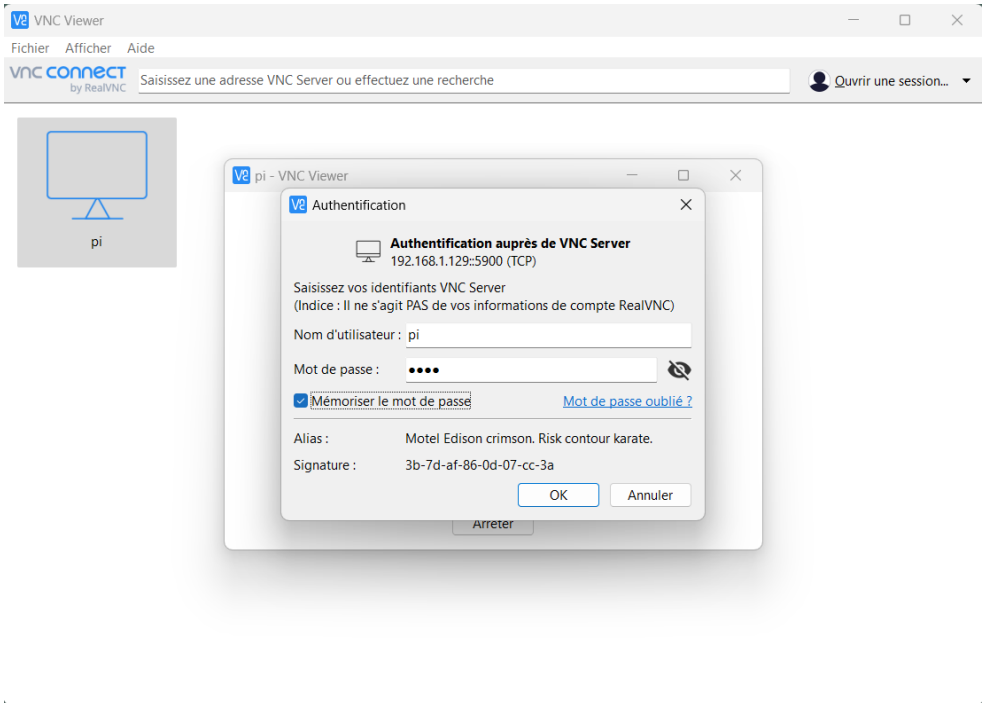
Figure 31 - Connexion au profil



Source : Auteur

Le serveur nous demande de nous authentifier, avec l'utilisateur « pi » et le mot de passe qui lui est associé :

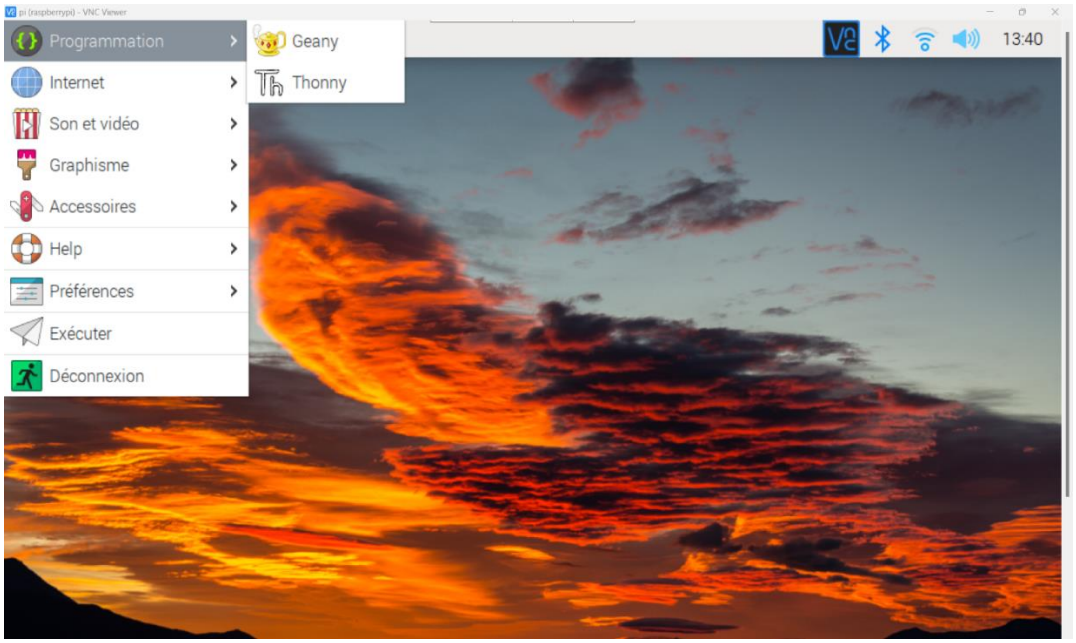
Figure 32 - Demande de mot de passe de l'utilisateur "pi"



SOURCE : AUTEUR

Nous avons ainsi accès à l'interface graphique du Raspberry Pi avec le bureau comme si tout été présent en local :

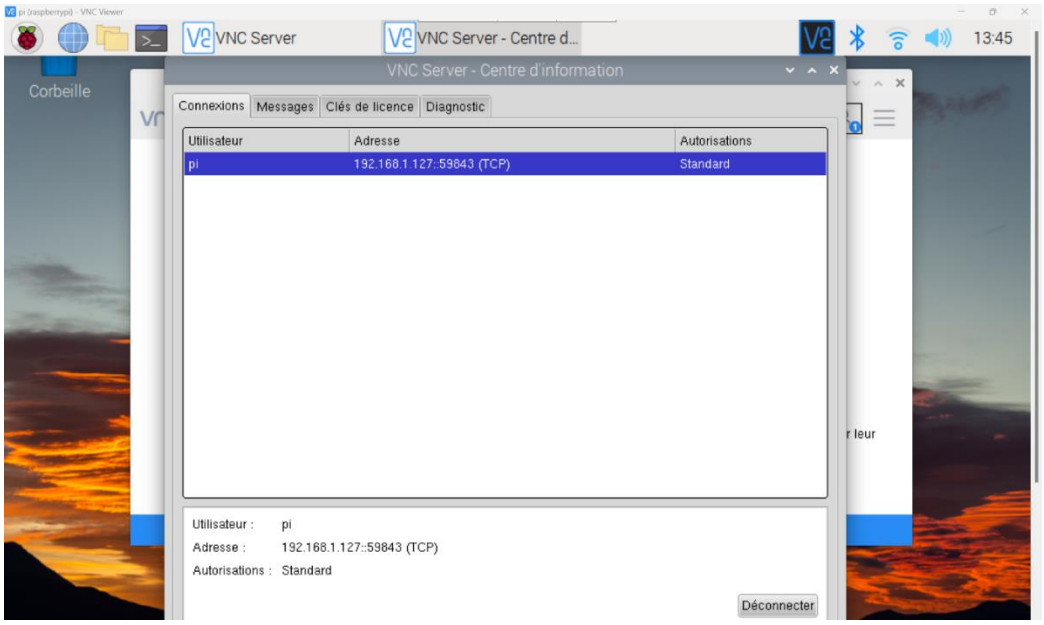
Figure 33 - Bureau du raspberry Pi



SOURCE : AUTEUR

Nous accédons au panneau de « Monitoring », de surveillance et vue d'ensemble des connexions sur le serveur « VNC » par le biais l'icône en haut à droite de l'écran, à gauche de l'icône du « Bluetooth » :

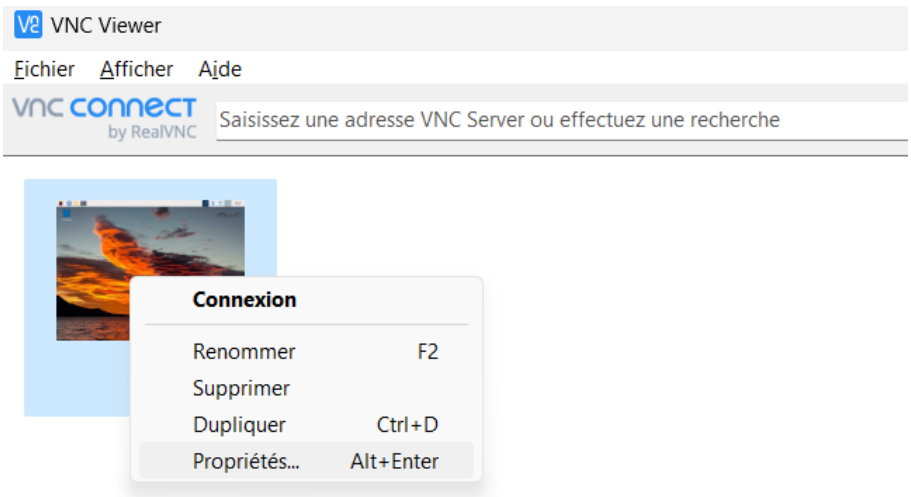
Figure 34 - Liste de connexion active VNC



SOURCE : AUTEUR

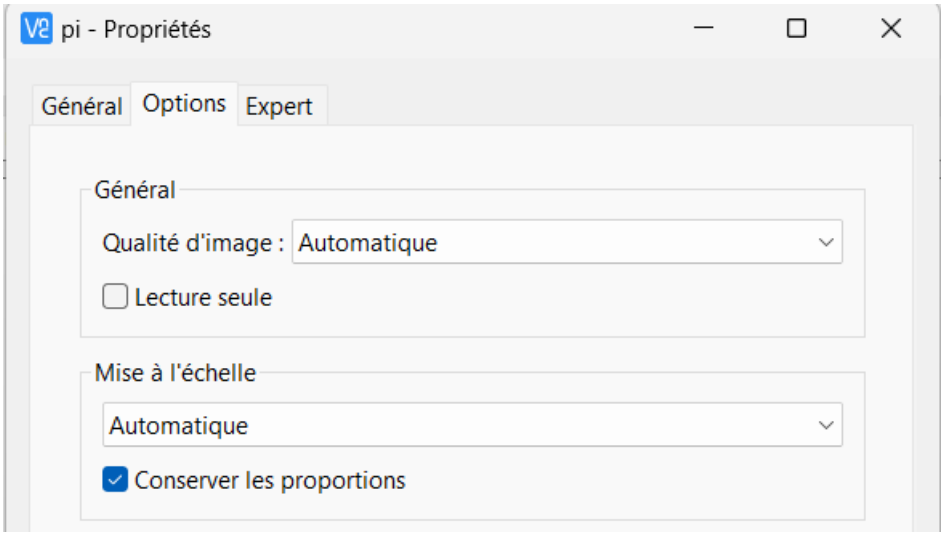
Nous pouvons configurer le client de manière plus avancée en effectuant un clique-droit et propriété :

Figure 35 - Accès au propriétés du profil



Dans l'onglet « options », nous avons plusieurs choix mais la plus intéressante reste la « qualité de l'image », c'est un choix qui peut être personnalisé pour mettre en priorité la réactivité sur la qualité d'image si la connexion est lente, la valeur automatique analyse la qualité du réseau pour faire varier cette option.

Figure 36 - Option du profil VNC



Source : Auteur

De plus lors du contrôle à distance l'outil offre la possibilité de transférer des fichiers depuis le périphérique local au serveur « VNC » à distance et inversement, il a aussi un chat intégré pour communiquer en cas d'assistance.

Il faut comprendre que lors d'une connexion à distance, cela créer une session. Il est important de comprendre que nous pouvons avoir une session pour chaque utilisateur existant du serveur « VNC » et des sessions partageant le même bureau ainsi que des sessions plus sécurisées dont les tunnels de communication ont un niveau plus élevé de chiffrage. Nous avons réussi à effectuer un contrôle avec accès graphique à distance. (Can-I-share-my-remote-control-session-screen-, s.d.)

3. Connexion à distance via « X11 »

« X11 » est un système de fenêtrage pour les systèmes d'exploitation « Unix » et « Linux ». Il permet d'afficher des fenêtres, une interface graphique sur l'écran d'un ordinateur local ainsi qu'interpréter les entrées provenant du clavier ou de la souris.

La grande différence entre « VNC Viewer » et « X11 » est que « VNC » permet de se connecter à un ordinateur distant et « X11 » est un système de fenêtrage, ils sont tout à fait complémentaires.

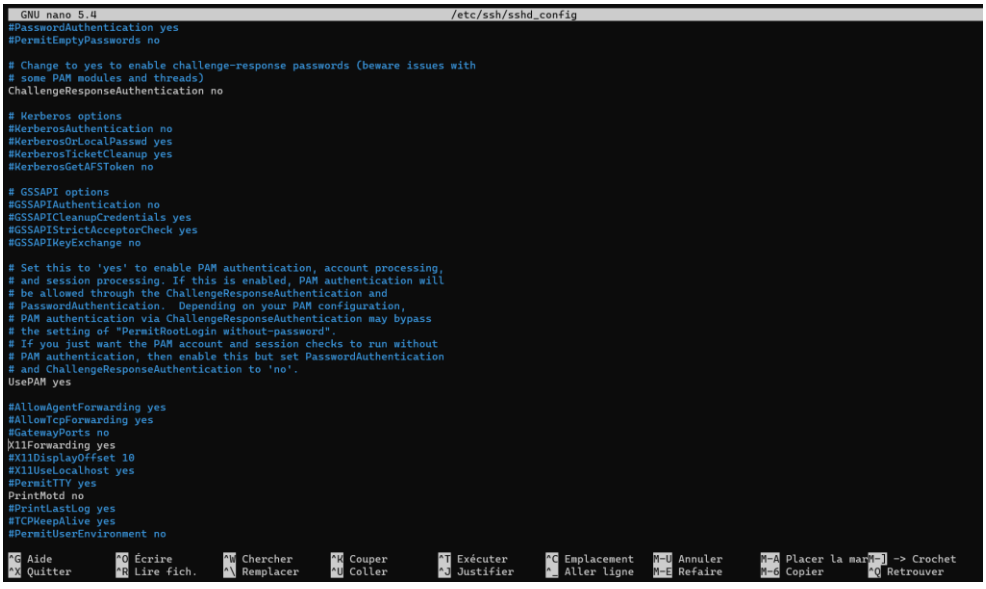
Tout d'abord pour utiliser « X11 » nous allons devoir modifier le fichier de configuration du serveur « SSH » pour autoriser la redirection « X11 » :

Figure 37 - Outil d'édition de texte

SOURCE : AUTEUR

Ensuit nous devons décommenter la ligne suivante « X11Forwarding yes » :

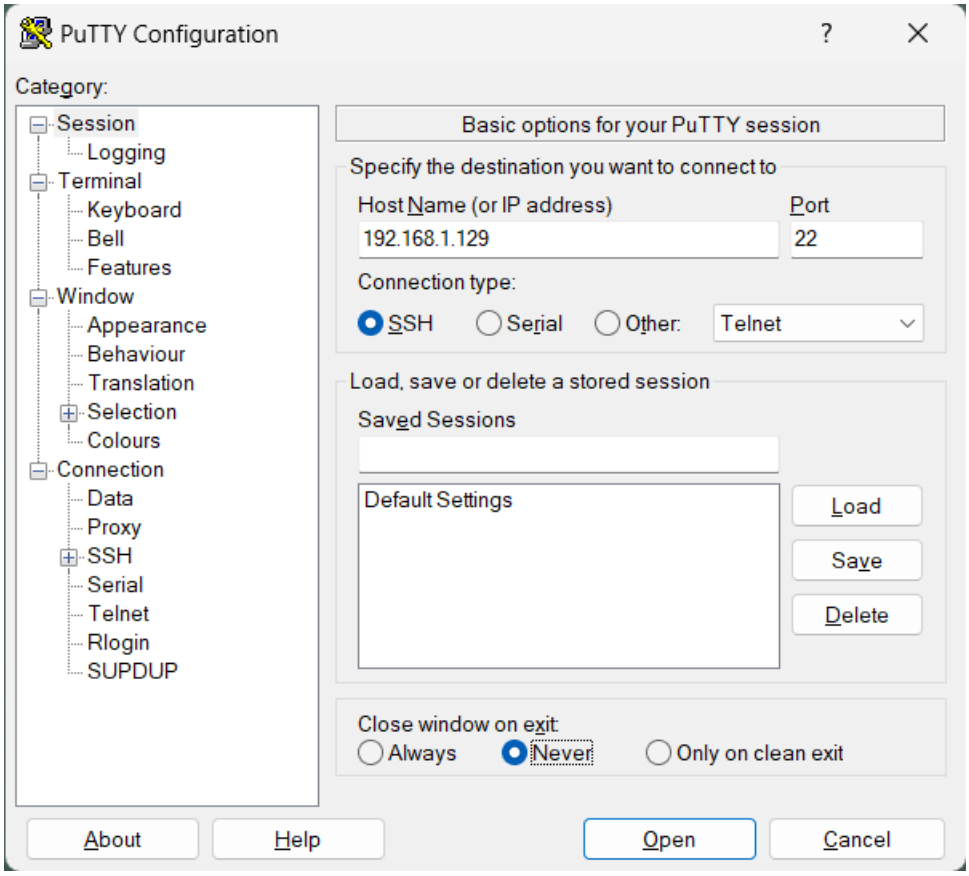
Figure 38 - Configuration du service SSH



SOURCE : AUTEUR

Nous allons relancer le service pour que les modifications prennent effet « sudo systemctl reload sshd » et ainsi ouvrir le programme « PuTTY » qui intègre la redirection « X11 » ainsi nous permettre de lancer des fenêtres via l'émulateur de terminal « PuTTY ». Nous entrons l'adresse IP ou le nom d'hôte et utilisons le port 22 destiné au service « SSH » :

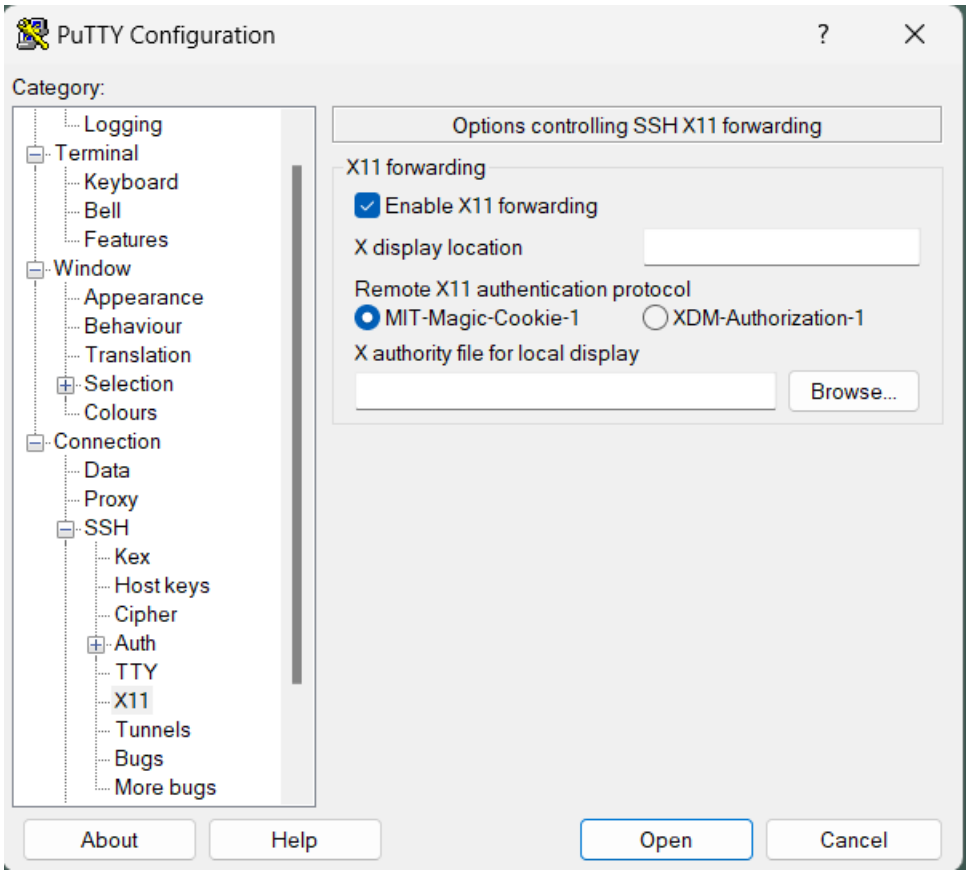
Figure 39 - Interface PuTTY



SOURCE : AUTEUR

Nous allons dans le menu latéral gauche dans Connexion > SSH > X11 puis checkboxer « Enable X11 forwarding » :

Figure 40 - Activation du module X11



SOURCE : AUTEUR

Pour pouvoir profiter pleinement de « X11 » nous devons installer « Xming » qui est programme qui implémente X11 pour les systèmes Windows et donc nous pouvons exécuter des programmes à distance via « PuTTY » et les lancer en local via « X11 » et « Xming ».

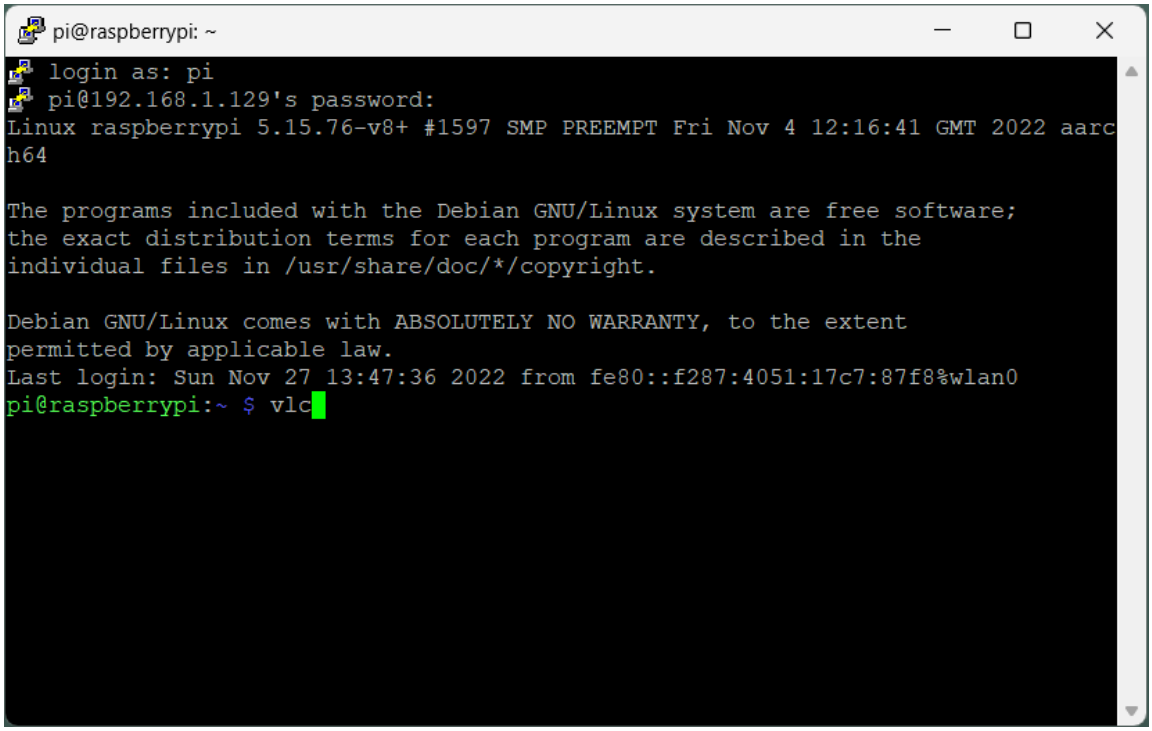
Pour que cela fonctionne « Xming doit impérativement être en cours d'exécution lors du fenêtrage d'application ! Nous devons donc télécharger « Xming » sur ce site https://sourceforge.net/ puis l'installer. Aucune configuration n'est requise, il faut tout simplement le lancer et nous pouvons passer à l'étape suivante :

Figure 41 - Site de téléchargement XMing

(xming)

Après l'avoir installé puis démarré nous pouvons retourner sur « PuTTY ». Lancer la connexion, s'authentifier et essayer de lancer la commande « vlc » qui est un lecteur multimédia :

Figure 42 - Lancement du fenêtrage de VLC



SOURCE : AUTEUR

Le fenêtrage apparaît sur l'ordinateur local, tout s'est bien passé !

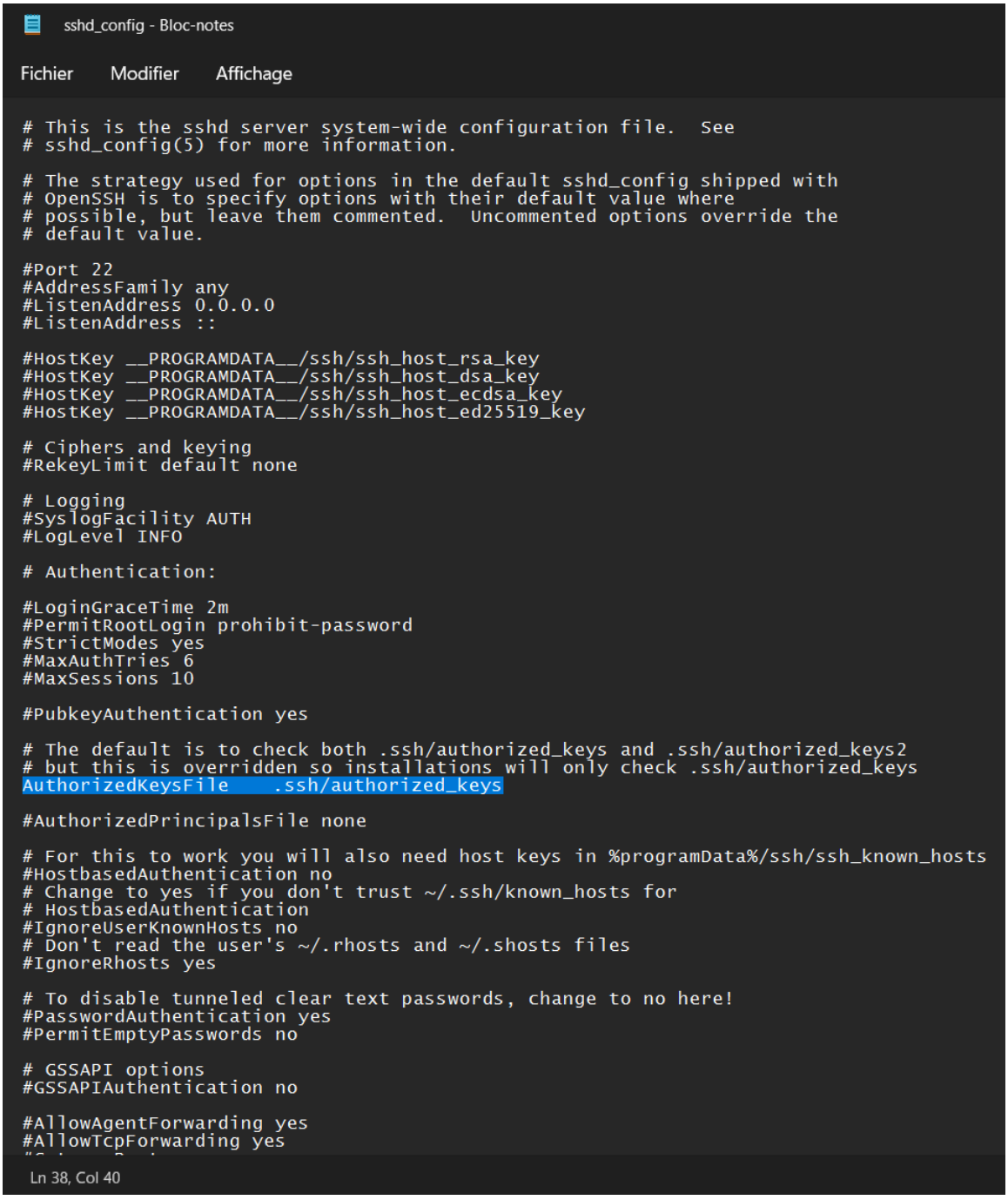
3. Exemple pratique

Nous allons mettre en pratique différents éléments vu dans ce document. Il s'agira à ce que le Raspberry Pi lance, dès le démarrage, un service ou « daemon » qui exécutera un script « bash » afin d'automatiser la suite d'instruction suivante :

Le Raspberry Pi, se connectera en « SSH » à l'ordinateur local par la méthode de clé, créer un fichier de type « batch », l'exécuter pour notifier l'utilisateur que le nano-ordinateur est bien lancé et prêt à l'emploi.

Tout d'abord il faudra installer sur Windows, le serveur « SSH » pour que le Raspberry Pi puisse se connecter. Comme démontré précédemment, il faut rechercher dans le menu démarré « Fonctionnalités facultative » puis installer « Open-SSH Server ». Une fois installer nous allons devoir configurer le serveur au chemin suivant « C:\ProgramData\ssh\sshd_config » de manière à activer l'authentification par clé :

Figure 43 - Configuration du service SSH sur WIndows



SOURCE : AUTEUR

Après que cela est fait, nous allons poser les clés générées du Raspberry Pi sur Windows. Nous allons donc effectuer un « Secure Copy » pour déposer la clé publique depuis le Raspberry pi :

Figure 44 - Connexion au Pi puis transfert de la clé publique généré sur Windows

SOURCE : AUTEUR

La méthode d'authentification par clé est judicieusement choisie parce qu'elle permet de ne pas divulguer de mot de passe dans le script « Bash » !

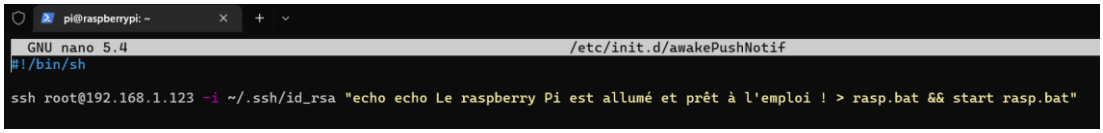
Nous allons pouvoir construire le fichier « bash » pour automatiser le processus décrit plus haut et nous devons l'activer dès le démarrage du Raspberry Pi comme service :

Figure 45 - Edition avec l'outil nano

Source : Auteur

Nous allons pouvoir éditer le fichier « bash », et automatiser la connexion. Ce qui est entre guillemet permet d'exécuter des commandes au terminal sécurisé. La commande « echo » permet d'afficher et l'opérateur « > » permet d'écrire dans le fichier « rasp.bat » qui sera enregistré dans le répertoire de l'utilisateur « jowad » :

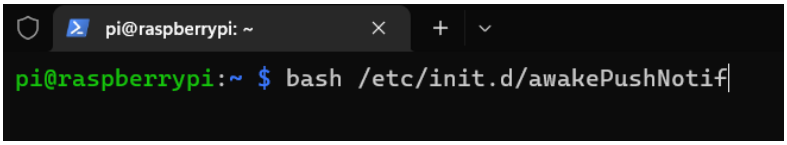
Figure 46 - Ajout des lignes pour se connecter et créer un fichier batch



Source : Auteur

Nous pouvons tester le code pour voir s'il effectue bien ce qui est demandé :

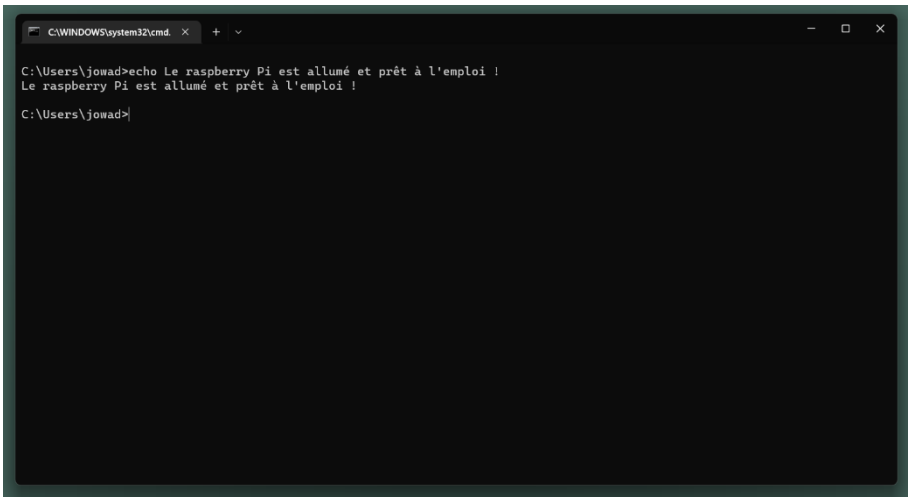
Figure 47 - Lancement du script bash



Source : Auteur

Tout se passe comme prévu, le script a bien écrit dans le dossier de l'utilisateur et à exécuter le fichier pour afficher cette fenêtre ci-dessous :

Figure 48 - Constat de l'exécution du script



Source : Auteur

4. Conclusion générale

Nous avons effectué plusieurs variantes de connexion, proposant, d'une part une connexion au terminal utilisant une méthode plus sécurisée avec clés d'authentification. Ainsi qu'une compréhension et manipulation des « Daemon » sous Linux mais aussi des services « Windows ».

Ensuite nous avons pu obtenir l'interface graphique du Raspberry Pi et comprendre l'outil « VNC Viewer » afin d'effectuer des actions à distance, utile pour le l'assistance à distance ou la maintenance.

Enfin un dernier point sur une option complémentaire de système de fenêtrage appelé « X11 » permettant d'apporter une approche graphique tout en étant en ligne de commande. Un système différent de « VNC » mais avec tous ses avantages.

5. Conclusion personnelle

Ce document contient des domaines majeurs, largement utilisé dans le monde informatique. Il est important de les comprendre et savoir utiliser ces différents outils.

L'informatique est un vaste domaine et pratiquer un métier « IT » oblige d'aborder et de prendre le rôle dans plusieurs secteurs. Finalement chacun devra se spécialiser dans une partie de ce qu'est l'informatique. Ce document présente infime partie des immenses possibilités que propose l'informatique.

Personnellement la partie la plus intéressante est sur le « Shell sécurisé » qui aborde des notions de cryptographie et comprendre comment cela fonctionne montre à quel point ce système est ingénieux ! Le « SSH » est énormément utilisé, le connaître est une évidence pour la suite des travaux.

6. Références

Authentification-ssh-par-paire-de-cles-priee-publique. (s.d.). Récupéré sur https://www.oceanet-technology.com/publications/technique/authentification-ssh-par-paire-de-cles-priee-publique/: https://www.oceanet-technology.com/publications/technique/authentification-ssh-par-paire-de-cles-priee-publique/#:~:text=Une%20cl%C3%A9%20publique%20%3A%20elle%20sera,subtilit%C3%A9%2C%20passphrase%20et%20pas%20password%E2%80%A6

Can-I-share-my-remote-control-session-screen-. (s.d.). Récupéré sur https://help.realvnc.com/: https://help.realvnc.com/hc/en-us/articles/360002325398-Can-I-share-my-remote-control-session-screen-

configure-ssh-key-based-authentication-on-raspberry-pi. (s.d.). Récupéré sur https://www.geekyhacker.com: https://www.geekyhacker.com/2021/02/15/configure-ssh-key-based-authentication-on-raspberry-pi/

https://www.hostinger.com/. (s.d.). Récupéré sur https://www.hostinger.com/: https://www.hostinger.com/tutorials/wp-content/uploads/sites/2/2017/07/symmetric-encryption-ssh-tutorial.jpg

https://www.raspberrypi.com/software/. (s.d.). *https://www.raspberrypi.com/software/*. Récupéré sur https://www.raspberrypi.com/software/: https://www.raspberrypi.com/software/

https://www.realvnc.com/. (s.d.). Récupéré sur https://www.realvnc.com/: https://www.realvnc.com/

Joiakim, D. (s.d.).

linux-chmod-chown-change-file-permissions. (s.d.). Récupéré sur https://www.freecodecamp.org/: https://www.freecodecamp.org/news/linux-chmod-chown-change-file-permissions/

Qu'est ce qu'un VNCviewer. (s.d.). Récupéré sur http://www.ordinateur.cc/: http://www.ordinateur.cc/r%C3%A9seaux/Autre-R%C3%A9seaux-informatiques/79284.html

what-are-ssh-keys. (s.d.). Récupéré sur https://jumpcloud.com/: https://jumpcloud.com/fr/blog/what-are-ssh-keys

xming. (s.d.). *https://sourceforge.net/*. https://sourceforge.net/projects/xming/.