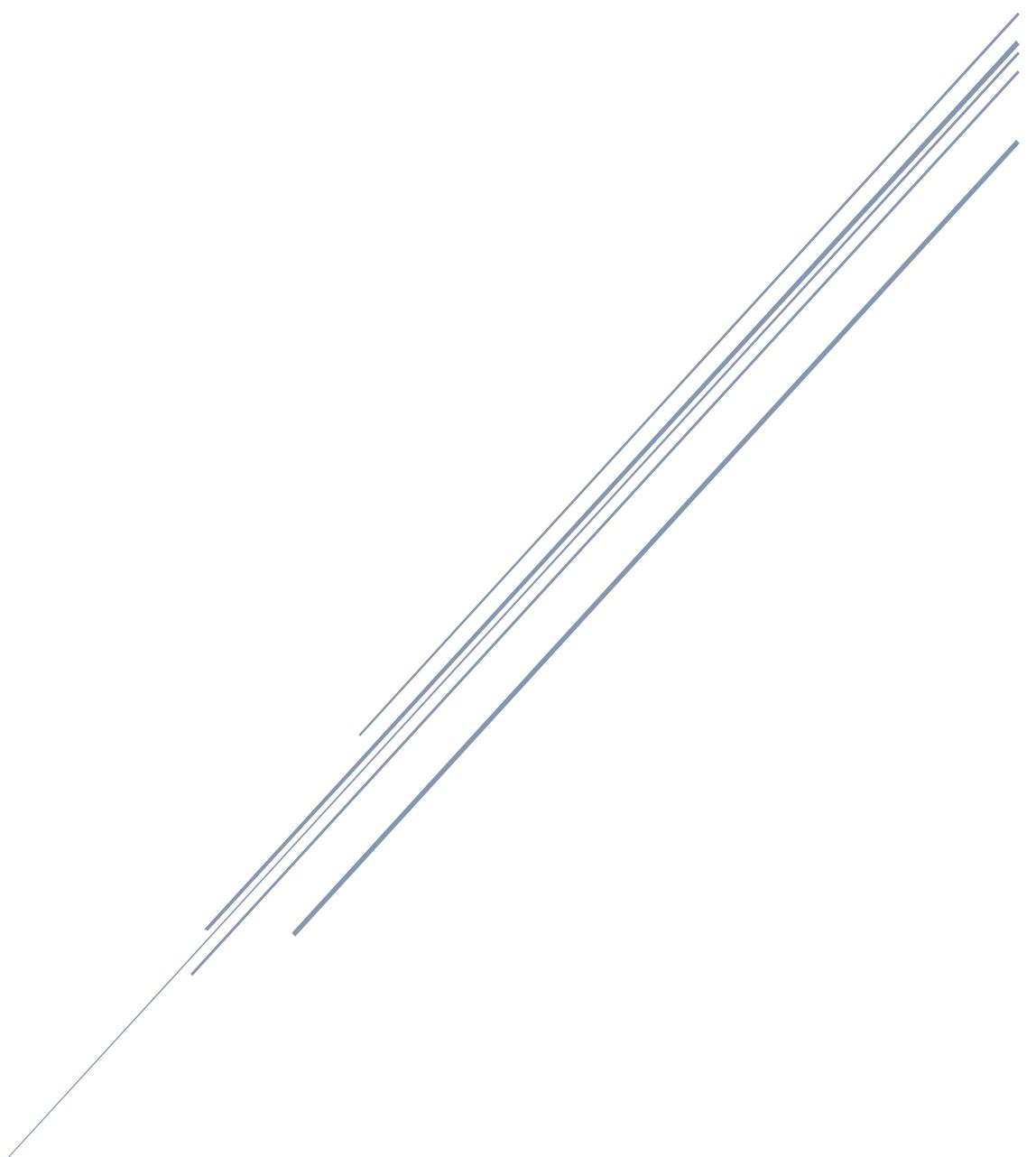


# RAPPORT DE PROJET

Raspberry Pi : Utilisation à domicile et services avancés



EPFL  
Rocco Ronzano

## Table des matières

<b>1</b>	<b>Analyse préliminaire .....</b>	<b>3</b>
1.1	Introduction.....	3
1.2	Objectifs .....	3
1.3	Planification initiale .....	3
<b>2</b>	<b>Analyse / Conception .....</b>	<b>5</b>
2.1	Concept .....	5
2.2	Stratégie de test.....	5
2.3	Risques techniques .....	6
2.4	Planification.....	7
2.5	Dossier de conception.....	8
2.6	Gestion de projet .....	9
<b>3</b>	<b>Réalisation.....</b>	<b>11</b>
3.1	Dossier de réalisation.....	11
3.1.1	Installation de Raspberry Pi OS.....	12
3.1.2	Configuration de base des Raspberry Pi .....	12
3.1.3	Mise en place de la sauvegarde sur les laptops.....	17
3.1.4	Mise en place de la réPLICATION des données avec Rsync .....	20
3.1.5	Mise en place des containers Docker .....	21
3.1.6	Création page web pour la documentation avec certificat SSL .....	23
3.1.7	Mise en place d'Ansible .....	29
3.2	Description des tests effectués.....	30
3.2.1	Tests de configuration de base .....	30
3.2.2	Tests de réPLICATION des données.....	34
3.2.3	Tests d'accès au site web .....	34
3.2.4	Tests Ansible .....	35
3.3	Erreurs restantes.....	35
3.4	Liste des documents fournis .....	35
<b>4</b>	<b>Conclusions.....</b>	<b>35</b>
4.1	Objectifs atteints / non-atteints .....	35
4.2	Points positifs / négatifs.....	36
4.3	Difficultés particulières .....	36
4.4	Suites possibles pour le projet.....	36
4.5	Bilan personnel .....	37
<b>5</b>	<b>Annexes .....</b>	<b>38</b>
5.1	Résumé du rapport du TPI / version succincte de la documentation .....	38
5.2	Sources – Bibliographie .....	39
5.3	Journal de travail .....	40
5.4	Glossaire .....	41

5.5	<b>Cahier des charges .....</b>	<b>43</b>
5.6	<b>Code source .....</b>	<b>47</b>

# 1 Analyse préliminaire

## 1.1 Introduction

Dans le cadre de ce second TPI mandaté par l'EPFL, l'objectif est de créer un environnement Raspberry Pi complet et polyvalent, destiné à des usages tant domestiques que professionnels. Le projet implique l'utilisation de trois Raspberry Pi (Pi1 à Pi3) pour deux étapes distinctes : une étape concernant un cas d'utilisation à domicile et une autre portant sur des services avancés et flexibles. Les objectifs principaux comprennent la configuration, la sécurisation et la documentation des services, en mettant l'accent sur l'automatisation via Ansible. Des contraintes techniques telles que l'utilisation de Raspberry Pi OS, Docker, Apache, Webmin, Rsync, CUPS, Git, et l'accès au NAS via SMB/CIFS définissent le cadre du projet.

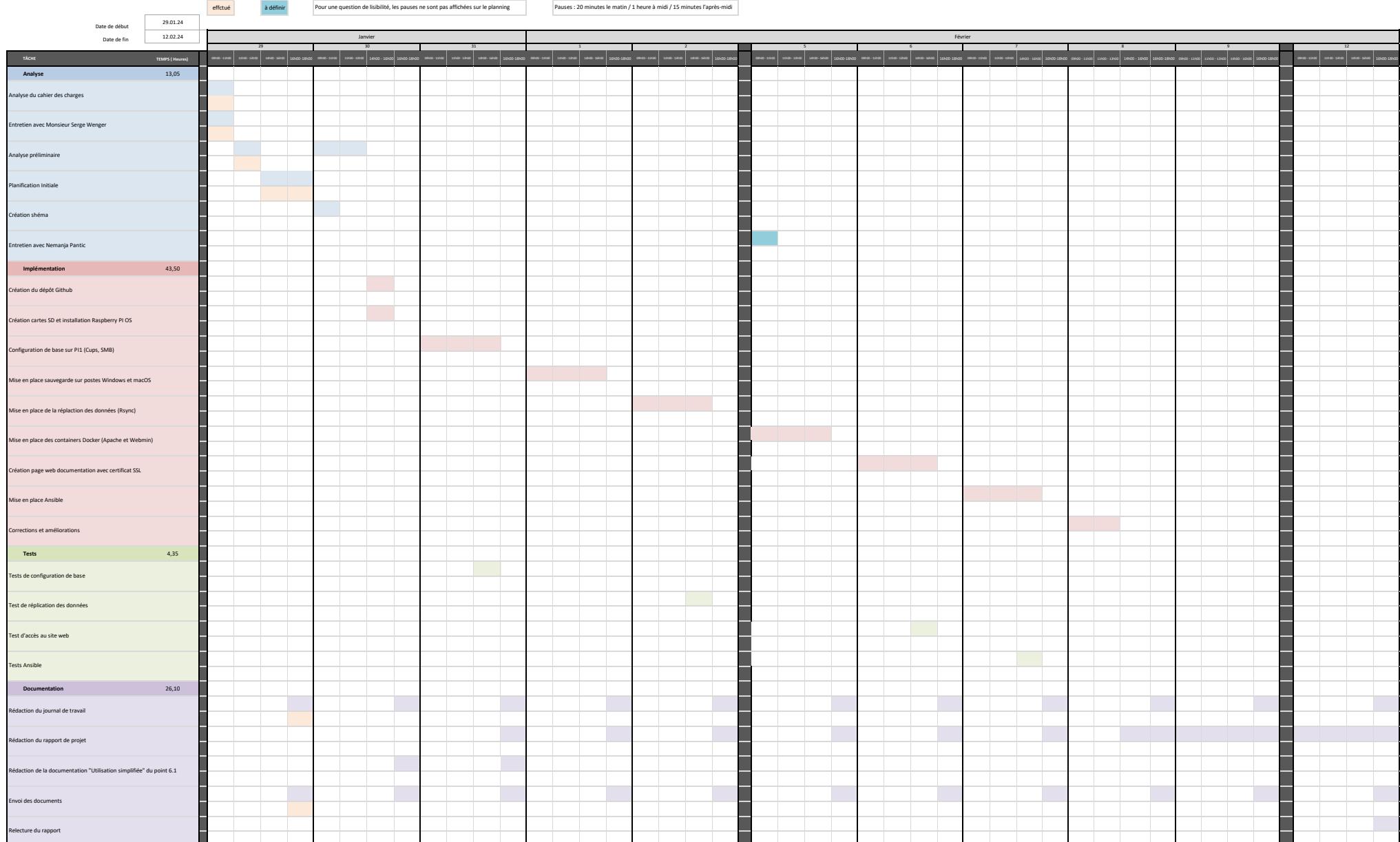
## 1.2 Objectifs

Les objectifs spécifiques comprennent l'élaboration d'une procédure "Utilisation simplifiée d'un Raspberry Pi à domicile" adaptée aux utilisateurs non techniques, ainsi que la mise en place de services avancés avec une gestion automatisée. Cela inclut la création d'un serveur web (Apache) hébergeant la documentation responsive "Utilisation simplifiée d'un Raspberry Pi à domicile" avec un certificat SSL, l'intégration de Webmin, et la mise en place d'une sauvegarde unidirectionnelle du NAS. De plus, une répartition cohérente des services sur les trois Raspberry Pi (Pi1 à Pi3) sera effectuée et une automatisation des tâches minimisant ainsi les actions manuelles grâce à l'utilisation d'Ansible.

## 1.3 Planification initiale

La planification initiale du projet sera structurée en deux grandes étapes (points 6.1 et 6.2) et présentée sous forme de diagramme. Cette planification initiale sera sujet à révision après une analyse approfondie du dossier et des besoins spécifiques du projet, afin de s'assurer de son adéquation avec les objectifs fixés.

## Planification initiale



*Figure 1 - Planification initiale*

## 2 Analyse / Conception

### 2.1 Concept

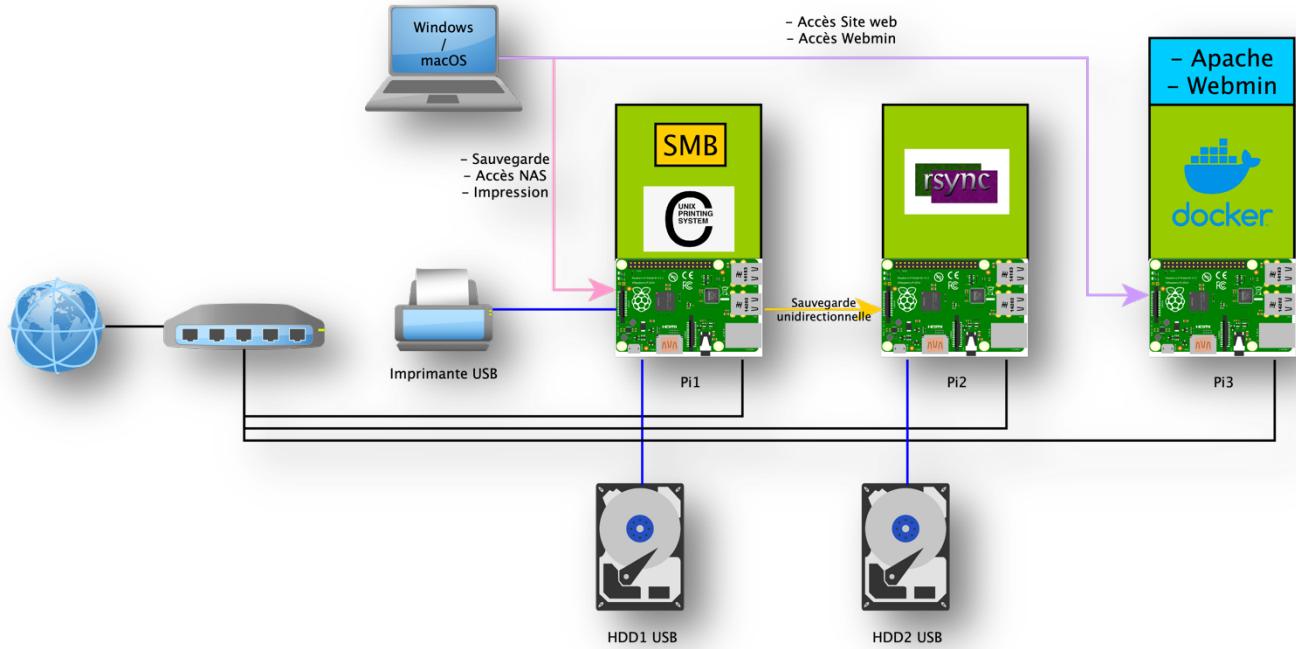


Figure 2 - Schéma de conception

Voici un concept schématisé pour répartir les charges entre les différents services demandés sur les 3 Raspberry Pi à disposition.

### 2.2 Stratégie de test

La stratégie de test sera soigneusement élaborée pour garantir la fiabilité et la stabilité du projet. Les tests seront organisés en plusieurs étapes, couvrant différentes dimensions du système.

Types de tests :

**Tests unitaires** : Chaque composant du projet sera soumis à des tests unitaires pour s'assurer de son bon fonctionnement individuel. Cela inclut les playbook Ansible, les configurations Docker, les fonctionnalités de sauvegarde avec Rsync et les différents services tels qu'Apache et Webmin.

**Tests d'intégration** : Les composants seront ensuite intégrés pour s'assurer qu'ils fonctionnent de manière cohérente ensemble. En vérifiant particulièrement l'interaction entre Raspberry Pi OS, Docker, et les services déployés.

**Tests de système** : Ces tests garantiront que l'ensemble du système fonctionne comme prévu dans son environnement complet. Cela inclut l'interaction entre les Raspberry Pi (Pi1 à Pi3), la gestion centralisée des services avec Webmin et la disponibilité des services même en cas de défaillance matérielle.

**Environnement de test dédié** : Un environnement de test séparé sera mis en place pour reproduire fidèlement le scénario d'utilisation réelle. Cela permettra d'identifier les problèmes potentiels avant la mise en production.

**Couverture exhaustive** : La stratégie de test visera à couvrir toutes les fonctionnalités du projet, des procédures d'installation aux services avancés, en passant par la sauvegarde et le partage de périphériques.

**Scénarios critiques** : Des scénarios critiques, tels que la défaillance d'un Raspberry Pi ou la perte de connectivité réseau, seront spécifiquement testés pour garantir la résilience du système.

**Données simulées** : Des données simulées seront utilisées pour tester la sauvegarde avec Rsync en s'assurant que les données sont correctement synchronisées entre les périphériques.

**Scénarios réels** : Des scénarios d'utilisation réelle seront reproduits autant que possible pour garantir la pertinence des tests.

### 2.3 Risques techniques

Le projet présente des défis techniques potentiels qui nécessitent une identification proactive et une gestion appropriée. Voici une énumération des principaux risques techniques, accompagnée des solutions envisagées pour les atténuer :

#### Risques :

**Complexité de la configuration** : La mise en place d'un environnement Raspberry Pi avec plusieurs services avancés peut devenir complexe, surtout pour des utilisateurs novices.

**Compétences techniques limitées** : Les utilisateurs finaux pour la procédure "Utilisation simplifiée d'un Raspberry Pi à domicile" peuvent manquer de compétences informatiques avancées.

**Interactions entre services** : L'intégration des différents services, notamment Docker, Apache, Webmin et Rsync, peuvent poser des défis d'interactions inattendues.

**Automatisation avec Ansible** : La configuration automatisée avec Ansible peut rencontrer des problèmes liés à la diversité des configurations matérielles et logicielles.

#### Solutions et atténuation :

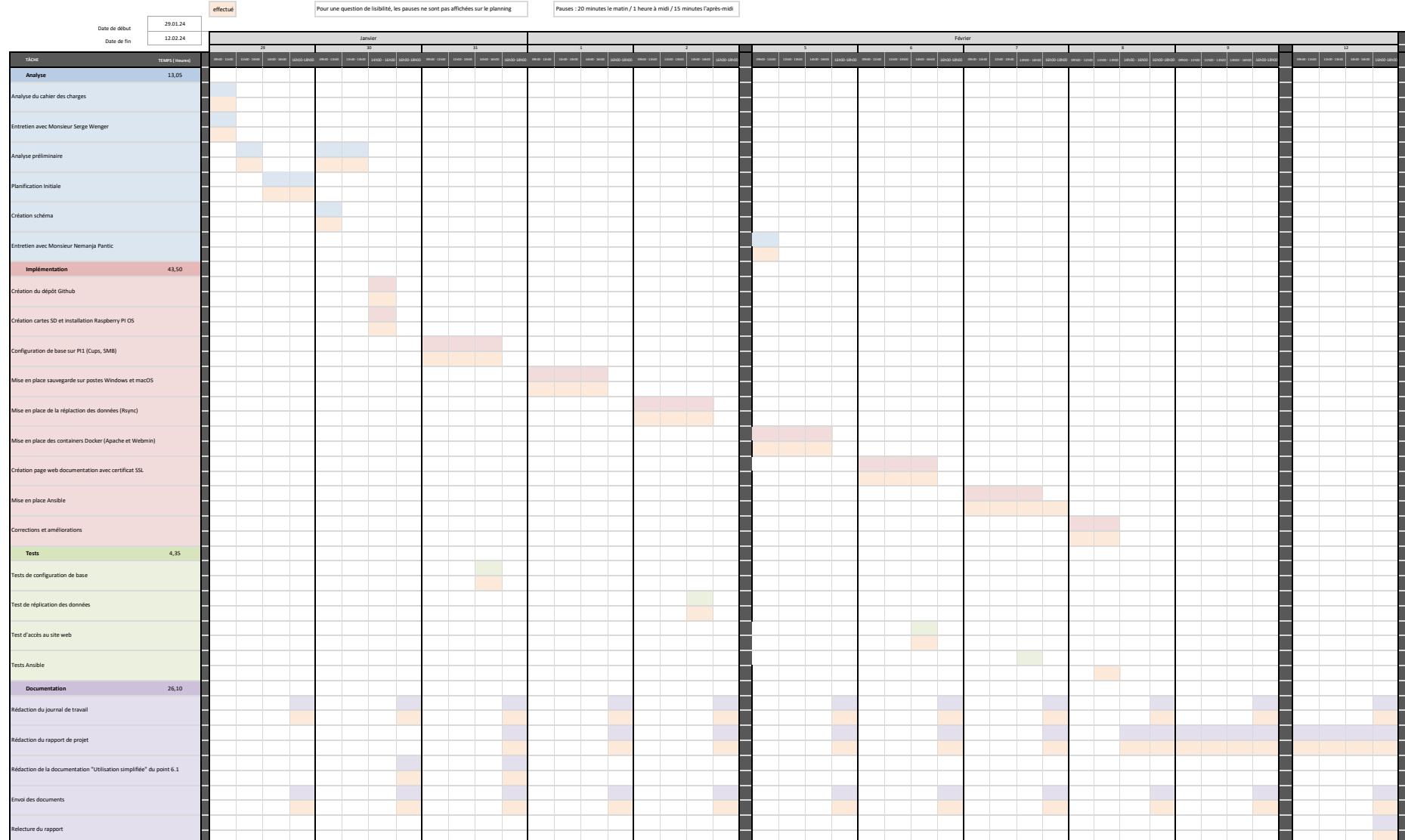
**Documentation détaillée** : Une documentation approfondie, incluant des tutoriels visuels et des explications simples, accompagnera la procédure pour les utilisateurs novices.

**Tests approfondis** : Des tests approfondis seront effectués sur l'interaction entre les différents services pour identifier et résoudre les problèmes potentiels avant la mise en production.

**Tests de compatibilité** : Des tests de compatibilité approfondis seront réalisés pour s'assurer qu'Ansible fonctionne de manière cohérente sur les différentes configurations matérielles.

## 2.4 Planification

Planification finale



## 2.5 Dossier de conception

### Raspberry Pi (Pi1 à Pi3)

#### 1. Installation de Raspberry Pi OS :

Objectif : Configurer un environnement Raspberry Pi OS 64 bits fonctionnel.

Étapes prévues :

- Installation de Raspberry Pi OS 64 bits.
- Mise à jour de l'OS pour partir d'une base sûre.

#### 2. Configuration individuelle des Raspberry Pi :

Objectif : Attribuer des rôles spécifiques à chaque Raspberry Pi en répartissant les charges selon les exigences du projet.

Étapes prévues :

- Identification des rôles spécifiques pour chaque Raspberry Pi.
- Installation des dépendances logicielles nécessaires.

#### 3. Automatisation avec Ansible :

Objectif : Automatiser la configuration individuelle de chaque Raspberry Pi.

Étapes prévues :

- Écriture de scripts Ansible pour l'automatisation.
- Tests approfondis des scripts pour garantir la cohérence des configurations.

### Disques durs USB

#### 1. Configuration des disques durs :

Objectif : Configurer les disques durs pour le partage de données via le protocole SMB/CIFS.

Étapes prévues :

- Formatage des disques durs selon les besoins du projet.
- Mise en place du partage de données via SMB/CIFS.

#### 2. Configuration de la sauvegarde avec Rsync :

Objectif : Écrire des scripts Rsync pour la sauvegarde unidirectionnelle depuis le NAS.

Étapes prévues :

- Écriture de scripts Ansible pour la sauvegarde unidirectionnelle Rsync.
- Tests de sauvegarde avec des données simulées pour valider le processus.

### Imprimante USB (HP Color LaserJet P3015)

#### 1. Configuration de l'imprimante :

Objectif : Installer les pilotes nécessaires sur Raspberry Pi et configurer le partage d'imprimante avec Cups.

Étapes prévues :

- Installation des pilotes sur Raspberry Pi.
- Configuration du partage d'imprimante via le réseau grâce à Cups.

#### 2. Tests d'impression :

Objectif : Réaliser des tests d'impression depuis les postes clients Windows et macOS.

Étapes prévues : Effectuer des tests d'impression pour garantir la fonctionnalité.

## Laptops macOS et Windows (MacBook Air 13" de 2014 avec macOS Big Sur et Dell Latitude 7490 avec Windows 11)

### 1. Configuration des postes clients macOS et Windows :

Objectif : Accéder au partage de fichier (NAS), sauvegarder la machine et imprimer

Étapes prévues :

- Création de raccourci pour l'accès au NAS.
- Implémenter et mettre en place la sauvegarde.
- Mise en place de l'imprimante en réseau

### 2. Tests d'utilisation simplifiée à domicile :

Objectif : Exécuter la procédure "Utilisation simplifiée d'un Raspberry Pi à domicile" pour évaluer la facilité d'utilisation depuis les laptops.

Étapes prévues : Suivre la procédure pour évaluer l'accessibilité et l'efficacité.

## 2.6 Gestion de projet

La gestion de projet joue un rôle crucial dans la réussite et la cohésion de toutes les étapes de ce projet. Pour assurer une approche méthodique et efficace, une combinaison des méthodologies (cascade et Agile) dite « hybride » sera utilisée, tirant parti des avantages de chaque méthode pour répondre aux besoins spécifiques du projet.

### Méthode Cascade

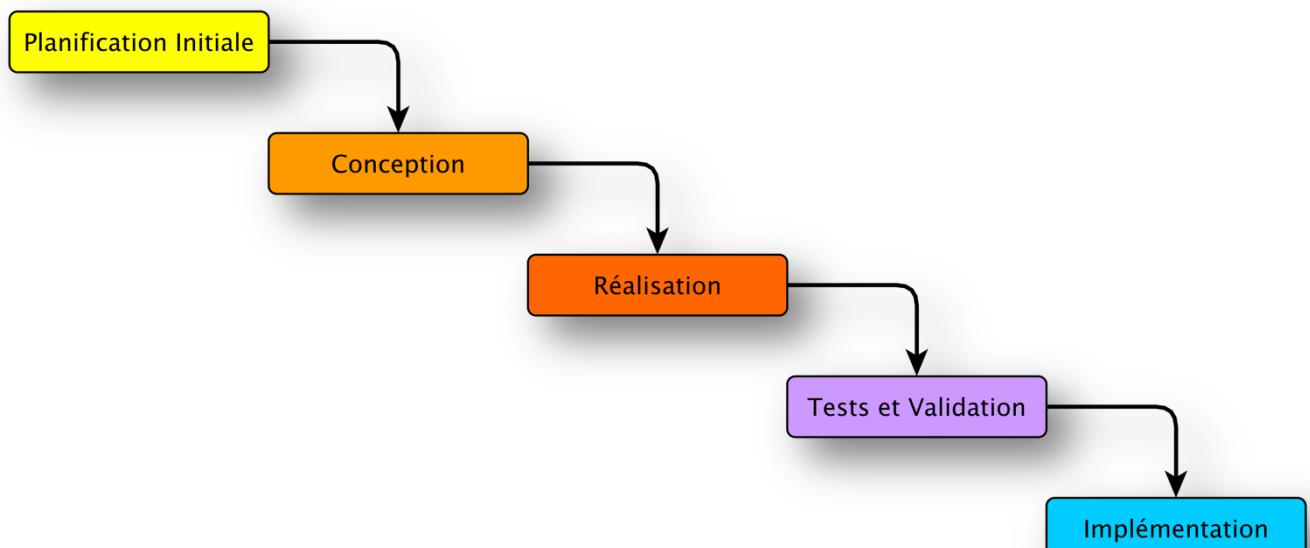


Figure 3 - Schéma méthode cascade

La méthode en cascade, également connue sous le nom de modèle en cascade, est une approche linéaire et séquentielle qui divise le projet en phases distinctes, chaque phase devant être complétée avant de passer à la suivante. Dans le contexte de ce projet, la méthode cascade sera utilisée pour établir une structure claire et définie, avec des jalons clés à chaque étape du processus. Les principales phases de la méthode en cascade incluront :

#### Planification Initiale :

- Définition des objectifs du projet, des livrables attendus et des contraintes techniques.

- Élaboration d'une première planification détaillée des tâches et des ressources nécessaires.

#### Conception :

- Analyse approfondie des besoins et des spécifications du projet.
- Élaboration d'un dossier de conception détaillé, décrivant les choix matériels et logiciels, ainsi que l'architecture globale du projet.

#### Réalisation :

- Mise en œuvre concrète des décisions prises lors de la phase de conception.
- Installation, configuration et tests des composants matériels et logiciels du projet.

#### Tests et Validation :

- Réalisation de tests approfondis pour vérifier le bon fonctionnement de tous les éléments du système.
- Validation des résultats par rapport aux spécifications initiales du projet.

#### Implémentation :

- Déploiement final du système dans l'environnement de production.
- Formation des utilisateurs finaux et transition vers l'utilisation opérationnelle du système.

### Approche Agile

L'approche Agile est une méthodologie itérative et incrémentale qui met l'accent sur la collaboration continue entre les équipes et les parties prenantes, ainsi que sur la flexibilité et l'adaptabilité aux changements. Dans le cadre de ce projet, des éléments de l'approche Agile seront intégrés pour favoriser une gestion plus dynamique et réactive du processus de développement. Quelques principes de l'approche Agile inclus dans ce projet sont :

#### Itérations Courtes :

- Adoption de cycles de développement courts pour livrer des résultats tangibles à intervalles réguliers, permettant des ajustements rapides en fonction des retours d'expérience.

#### Adaptabilité :

- Capacité à réagir rapidement aux changements et aux évolutions des exigences du projet, en ajustant les priorités et les plans en conséquence.

En combinant les avantages de la méthode en cascade pour une planification rigoureuse et structurée avec les principes de l'approche Agile pour une gestion flexible et adaptative, ce projet vise à maximiser les chances de succès et à répondre efficacement aux besoins de toutes les parties prenantes.

### 3 Réalisation

#### 3.1 Dossier de réalisation

Le processus de réalisation du projet consiste en la mise en œuvre effective des décisions prises lors de la phase de conception. Cette section détaille des différentes étapes de mise en place du projet, depuis l'installation initiale jusqu'aux tests finaux.

Dans le cadre de ce projet, plusieurs logiciels jouent un rôle crucial dans la gestion et le développement. Parmi ces outils :



**GitHub Desktop** est utilisé comme plateforme de gestion de version pour le contrôle de code source. En fournissant un environnement convivial pour la gestion des dépôts Git, GitHub Desktop facilite le suivi des modifications apportées au code et permet une gestion efficace des versions.



Pour la conception et la visualisation des diagrammes, **yEd Graph Editor** est un choix judicieux. Grâce à ses fonctionnalités avancées de création de graphiques et de diagrammes, yEd permet de représenter visuellement l'architecture du projet, les flux de données et les interactions système, offrant ainsi une compréhension claire et intuitive de la structure globale du projet.



**Visual Studio Code** demeure un choix idéal en tant qu'éditeur de code puissant et polyvalent. Il offre une gamme étendue d'extensions et de fonctionnalités pour le développement logiciel, ce qui facilite l'écriture, la modification et le débogage du code. Son interface intuitive et ses fonctionnalités avancées contribuent à optimiser mon flux de travail de développement.



Enfin, **OneDrive** est utilisé comme solution de stockage et de partage de fichiers, offrant une accessibilité accrue aux documents et aux ressources du projet. Sa capacité à synchroniser les fichiers sur plusieurs appareils et à fournir un espace de stockage sécurisé dans le cloud facilite la gestion des fichiers. La synchronisation automatique des fichiers garantit que toutes les versions sont toujours à jour, ce qui est essentiel pour maintenir la cohérence et l'intégrité du projet, assurant ainsi une progression fluide et efficace.

L'utilisation de ces logiciels offre une combinaison harmonieuse de fonctionnalités essentielles pour la gestion, le développement et la productivité dans le cadre de ce TPI, contribuant ainsi à son efficacité globale et à son succès.

### 3.1.1 Installation de Raspberry Pi OS

Pour installer Raspberry Pi OS j'ai choisi d'utiliser l'utilitaire Raspberry Pi Imager sur le site officiel de Raspberry car c'est un des outils les plus connus pour créer des médias bootable facilement. J'ai ensuite créé 3 cartes SD qui contiennent chacune une copie de Raspberry Pi OS 64bits avec des pré-paramètres pour faciliter la connexion en SSH.



Figure 4 – Interface Raspberry Pi Imager

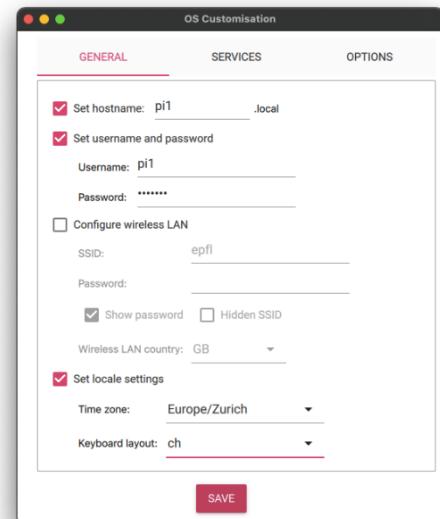


Figure 5 - Interface customisation de l'OS

J'ai ainsi configuré pour chaque Pi :

Nom machine	Nom localhost	Nom utilisateur	Mot de passe
Pi1	pi1.local	pi1	EPFLpi1
Pi2	pi2.local	pi2	EPFLpi2
Pi3	pi3.local	pi3	EPFLpi3

### 3.1.2 Configuration de base des Raspberry Pi

Après avoir inséré les cartes dans les Pi, je me suis connecté en SSH sur chaque Pi grâce à leur nom locaux (pi1.local, pi2.local et pi3.local) que j'ai configuré au préalable lors de la création des cartes SD et j'ai lancé une mise à jour du système et du firmware sur tous les Pi pour partir sur une base saine.

```
ssh pi1@pi1.local  
sudo apt -y update && sudo apt -y full-upgrade
```

Ensuite, pour éviter tout problèmes de connexion SSH, j'ai généré des clés SSH sur chaque compte root de chaque Raspberry Pi avec les paramètres par défaut en utilisant cette commande :

```
ssh-keygen -t rsa
```

Après cela j'ai échangé les clés publiques entre tous les comptes avec cette commande :

```
ssh-copy-id user@ipAddress
```

Pour la configuration de base du Pi1 j'ai exécuté ces commandes qui permettent de mettre en place rapidement Samba et Cups.

```
sudo apt -y install cups samba samba-common-bin
```

Je rajoute l'utilisateur actuel dans le groupe lpadmin.

```
sudo usermod -aG lpadmin whoami
```

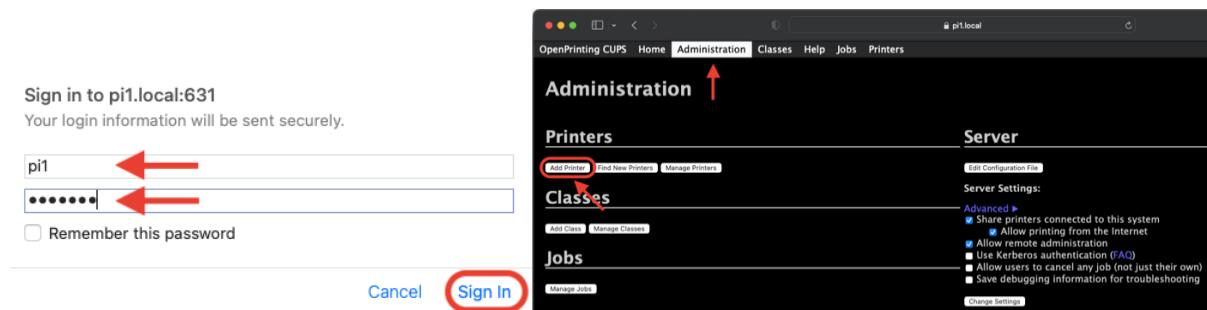
Après avoir installé les paquets et rajouté l'utilisateur actuel dans le groupe lpadmin, j'édite le fichier de configuration /etc/cups/cupsd.conf pour permettre d'imprimer et afficher la page d'administration depuis n'importe quelle machine sur le réseau.

```
Listen *:631
<Location />
    Order allow,deny
    Allow all
</Location>
<Location /admin>
    Order allow,deny
    Allow all
</Location>
```

Ensuite je redémarre le service CUPS pour prendre en charge le fichier édité

```
sudo service cups restart
```

Finalement, j'ouvre un navigateur et entre l'url « pi1.local:631 » pour accéder à la page d'administration de CUPS en rentrant l'utilisateur et son mot passe qui a été ajouté au groupe « lpadmin » précédemment et une fois connecté je vais sur « Ajouter une imprimante ».



Ensute je choisi quelle imprimante à ajouter à CUPS, pour ce TPI j'ai une imprimante HP LaserJet P3015 donc je choisis la ligne « HP LaserJet P3010 Series (HP LaserJet P3010 Series) » et je valide avec « suivant ».

**Add Printer**

### Add Printer

**Local Printers:**

- CUPS-BRF (Virtual Braille BRF Printer)
- VNC Remote Printer (VNC Printer)
- HP LaserJet P3010 Series USB VNCQF980V8 HPLIP (HP LaserJet P3010 Series)
- HP LaserJet P3010 Series (HP LaserJet P3010 Series)** (selected)
- HP Fax (HPLIP)

**Discovered Network Printers:**

- EPSON ET-8550 Series (EPSON ET-8550 Series)
- HP Color LaserJet M553 [25154E] (Hewlett-Packard HP Color LaserJet M553)
- HP LaserJet 400 color M451dn (EAF42A) (Hewlett-Packard HP LaserJet 400 color M451dn)
- Xerox Phaser 6600DN (04:34:8D) (Xerox Phaser 6600DN)
- Xerox VersaLink C500 (8a:c8:a3) (Xerox VersaLink C500)
- Hewlett-Packard HP LaserJet 400 color M451dn (driverless) (Hewlett-Packard HP LaserJet 400 color M451dn)
- Xerox Phaser 6600DN (driverless) (Xerox Phaser 6600DN)
- EPSON ET-8550 Series (driverless) (EPSON ET-8550 Series)
- Hewlett-Packard HP Color LaserJet M553 (driverless) (Hewlett-Packard HP Color LaserJet M553)
- Xerox VersaLink C500 DN Printer (Xerox VersaLink C500)
- Xerox VersaLink C500 (driverless) (Xerox VersaLink C500)
- HP Color LaserJet M553 (HP Color LaserJet M553)
- Xerox Phaser 6600DN: Net 18.01,ESS 201207031042,IOT 02.00.01 (Xerox Phaser 6600DN)
- HP Color LaserJet M750 (HP Color LaserJet M750)
- Canon PRO-4000S (Canon PRO-4000S)
- EPSON AL-M320DN (EPSON AL-M320DN)
- HP LaserJet 400 color M451dn (HP LaserJet 400 color M451dn)
- EPSON ET-8550 Series (EPSON ET-8550 Series)

**Other Network Printers:**

- Backend Error Handler
- Internet Printing Protocol (https)
- LPD/LPR Host or Printer
- AppSocket/HP JetDirect
- Internet Printing Protocol (http)
- Internet Printing Protocol (ipp)
- Internet Printing Protocol (ipp)

**Continue**

Copyright © 2021-2022 OpenPrinting. All rights reserved.

Ensuite j'ajoute son nom pour y accéder en ligne, une description de l'imprimante, sa localisation, je clique sur la case à cocher pour partager l'imprimante sur le réseau et je valide avec « suivant ».

**Add Printer**

### Add Printer

**Name:** hp-tpi  
(May contain any printable characters except "/", "#", and space)

**Description:** hp-tpi  
(Human-readable description such as "HP LaserJet with Duplexer")

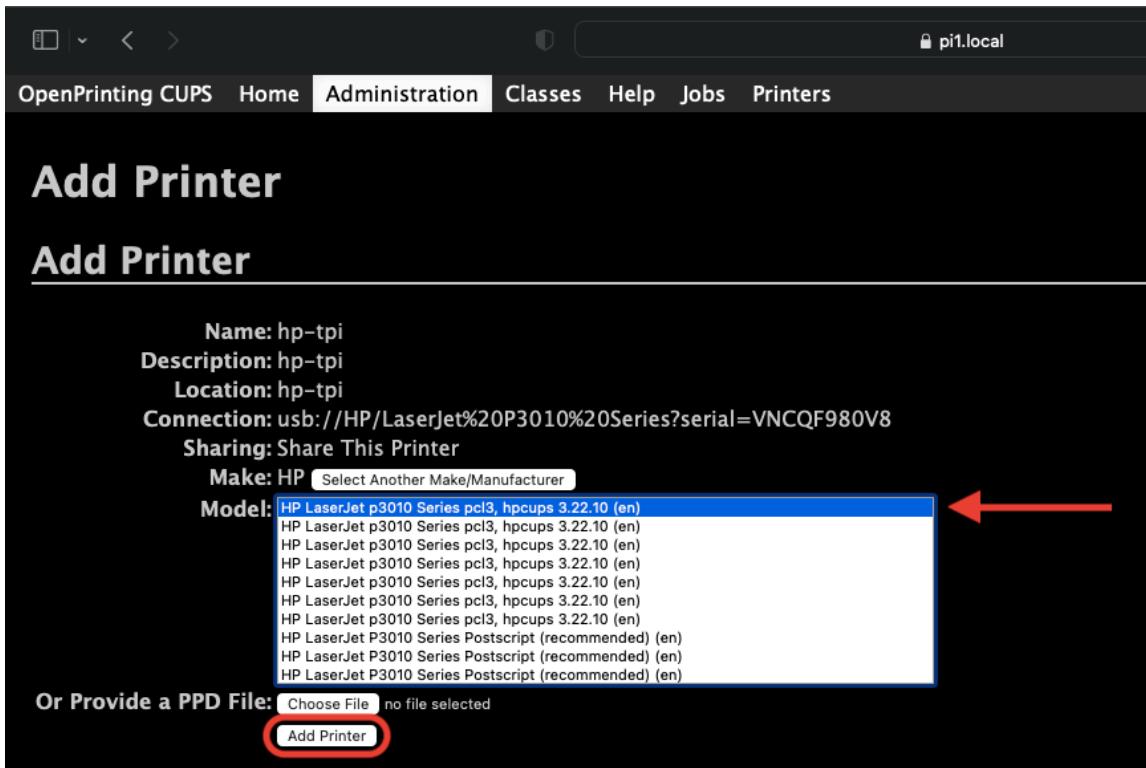
**Location:** hp-tpi  
(Human-readable location such as "Lab 1")

**Connection:** usb://HP/LaserJet%20P3010%20Series?serial=VNCQF980V8

**Sharing:**  Share This Printer

**Continue**

Ensuite je sélectionne le modèle le plus proche de l'imprimante actuelle, pour mon cas « HP LaserJet p3010 Series pcl3, hpcups 3.22.10 (en) » et je valide avec « suivant ».



Une fois validé, l'imprimante est prête à être utilisée.

Description	Location	Driver	Connection	Defaults
hp-tpi	hp-tpi	LaserJet p3010 Series pcl3, hpcups 3.22.10 (grayscale, 2-sided printing)	usb://HP/LaserJet%20P3010%20Series?serial=VNCQF980V8	job-sheets:none, none media=iso_a4_210x297mm sides=one-sided

Pour la partie Samba, je commence par monter le disque dur au système.

```
sudo mkdir /mnt/sda1 && sudo mount /dev/sda1
```

Pour m'éviter d'écrire cette commande à chaque fois que je redémarre le Pi, j'ajoute une ligne au fichier /etc/fstab pour monter directement le disque au démarrage.

```
/dev/sda1 /mnt/sda1/ ext4 defaults,noatime 0 1
```

Une fois monté, je crée les dossiers qui vont être utilisé pour le partage de fichiers.

```
sudo mkdir /mnt/sda1/backups
sudo mkdir /mnt/sda1/shared
sudo mkdir /mnt/sda1/shared/nas1-tpi
sudo mkdir /mnt/sda1/shared/nas2-tpi
sudo mkdir /mnt/sda1/TimeMachine
```

J'écris le fichier de configuration Samba pour permettre de partager les dossiers créés précédemment sur le réseau.

```
sudo tee -a /etc/samba/smb.conf > /dev/null <<EOL
[shared]
path=/mnt/sda1/shared
writeable = yes
create mask = 0770
directory mask = 0770
public=no

[backups]
Path = /mnt/sda1/backups
Writeable = yes
create mask = 0770
directory mask = 0770
public = no

[TimeMachine]
comment = Time Machine Backup
path = /mnt/sda1/TimeMachine
browseable = yes
writeable = yes
guest ok = no
create mask = 0770
directory mask = 0770
vfs objects = catia fruit streams_xattr
fruit:aapl = yes
fruit:time machine = yes
EOL
```

Je redémarre le service Samba pour appliquer les changements effectués dans le fichier de configuration.

```
sudo systemctl restart smbd
```

Je crée les utilisateurs qui accéderont aux partages de fichiers ainsi que les groupes nécessaires.

```
sudo adduser nas1-tpi
sudo adduser nas2-tpi
sudo adduser mac-tpi
sudo adduser win-tpi
sudo adduser homepi
sudo usermod -aG sudo homepi
sudo groupadd backups
sudo groupadd shared
sudo groupadd timemachine
```

Je crée un mot de passe aux utilisateurs Samba.

```
sudo smbpasswd -a nas1-tpi  
sudo smbpasswd -a nas2-tpi  
sudo smbpasswd -a mac-tpi  
sudo smbpasswd -a win-tpi
```

J'ajoute les utilisateurs aux bons groupes.

```
sudo adduser homepi nas1-tpi  
sudo adduser homepi nas2-tpi  
sudo adduser homepi mac-tpi  
sudo adduser homepi win-tpi  
sudo adduser homepi backups  
sudo adduser homepi shared  
sudo adduser homepi timemachine  
sudo adduser nas1-tpi homepi  
sudo adduser nas2-tpi homepi  
sudo adduser mac-tpi homepi  
sudo adduser win-tpi homepi  
sudo adduser nas1-tpi shared  
sudo adduser nas2-tpi shared  
sudo adduser mac-tpi timemachine  
sudo adduser win-tpi backups  
sudo adduser "$(whoami)" homepi
```

Je définit les droits et les permissions aux dossiers.

```
sudo chmod -R 770 /mnt/sda1  
sudo chown -R homepi /mnt/sda1  
sudo chown -R :homepi /mnt/sda1  
sudo chown -R :backups /mnt/sda1/backups  
sudo chown -R :shared /mnt/sda1/shared  
sudo chown -R :nas1-tpi /mnt/sda1/shared/nas1-tpi  
sudo chown -R :nas2-tpi /mnt/sda1/shared/nas2-tpi  
sudo chown -R :timemachine /mnt/sda1/TimeMachine
```

Je redémarre le service Samba pour appliquer les changements effectués.

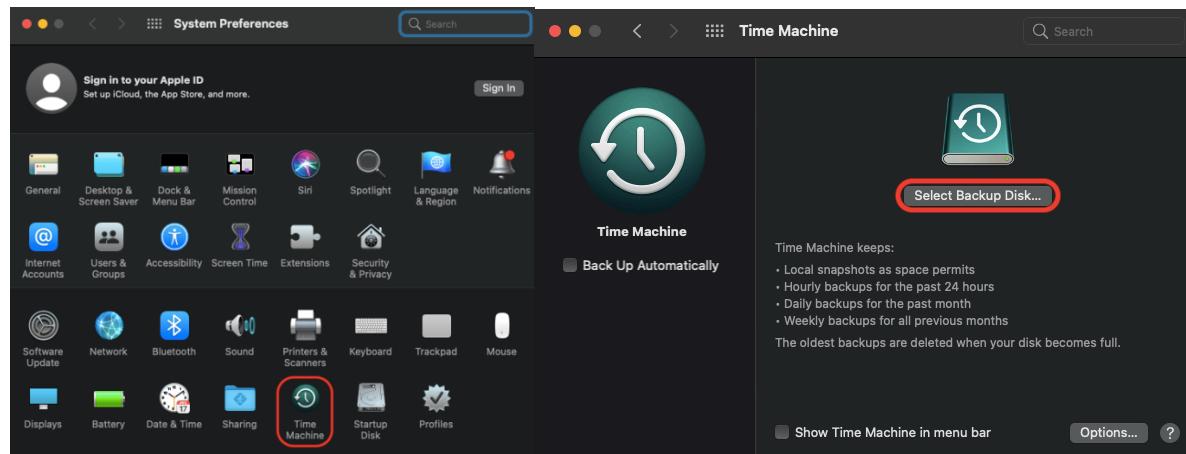
```
sudo systemctl restart smbd
```

### 3.1.3 Mise en place de la sauvegarde sur les laptops

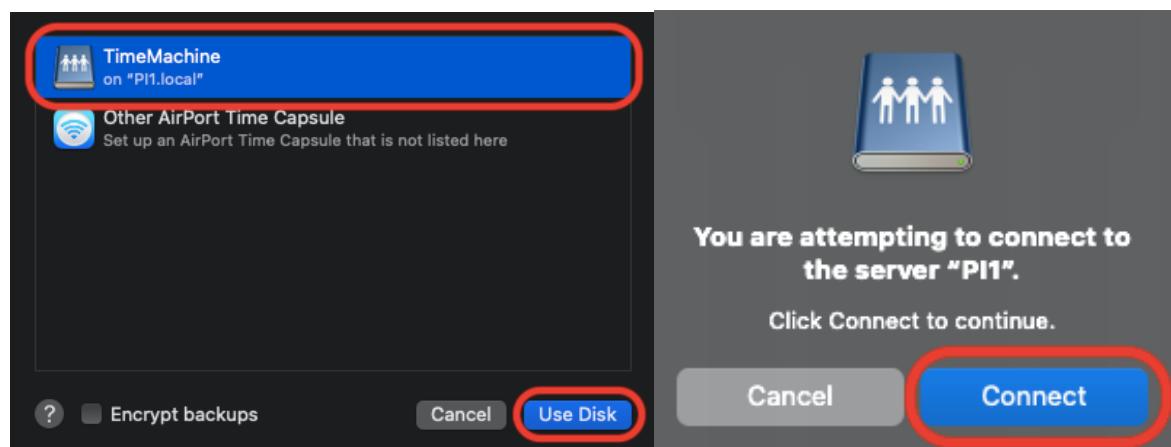
Sur le laptop Windows j'ai décidé d'utiliser l'utilitaire « Sauvegarder et restaurer (Windows 7) » et pour le laptop Apple j'ai décidé d'utiliser l'utilitaire « Time Machine » car il est déjà installé par défaut sur macOS.

Ces programmes ont l'avantage d'être gratuit, déjà présent dans le système donc aucune installation nécessaire, sauvegarde le système et les fichiers personnels de l'utilisateur, permet de réutiliser la sauvegarde comme point d'image système et permet donc de revenir à un point antérieur de son ordinateur très facilement.

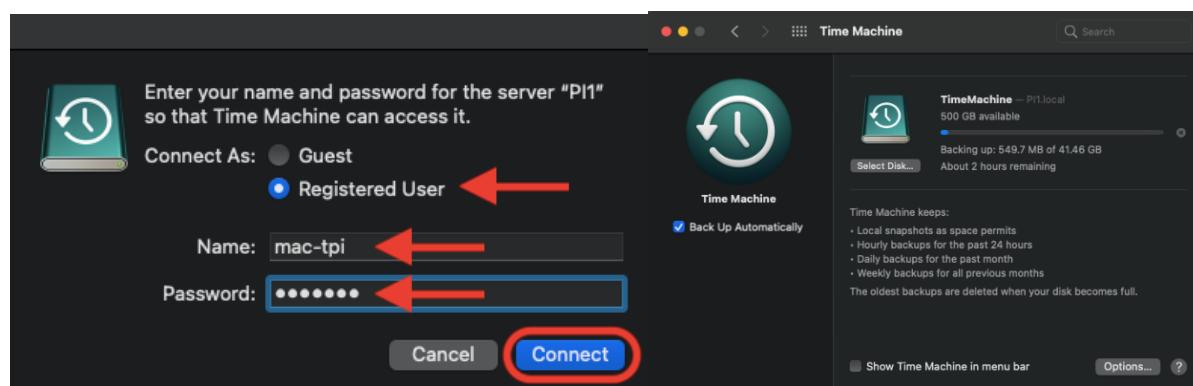
Pour macOS, j'ouvre les paramètres et cliqué sur Time Machine, ensuite j'ai cliqué sur « sélectionner un disque de sauvegarde ».



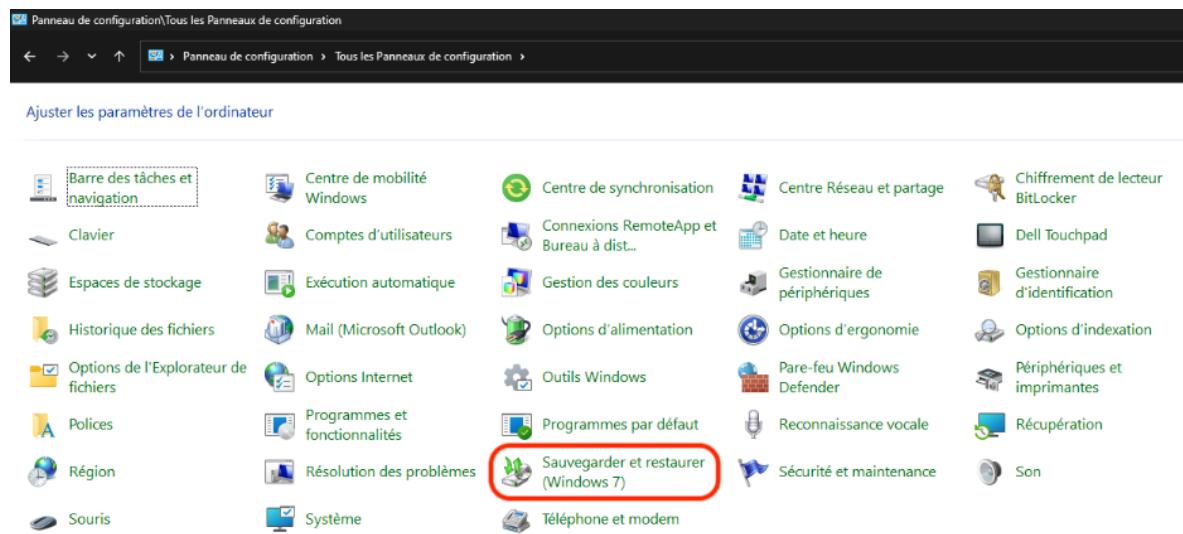
Un pop-up me demande alors de sélectionner un disque de sauvegarde USB ou en réseau, je sélectionne mon partage fraîchement créé et je clique sur « utiliser ce disque », un nouveau pop-up me demande alors si je veux me connecter sur partage et je clique sur « connecter ».



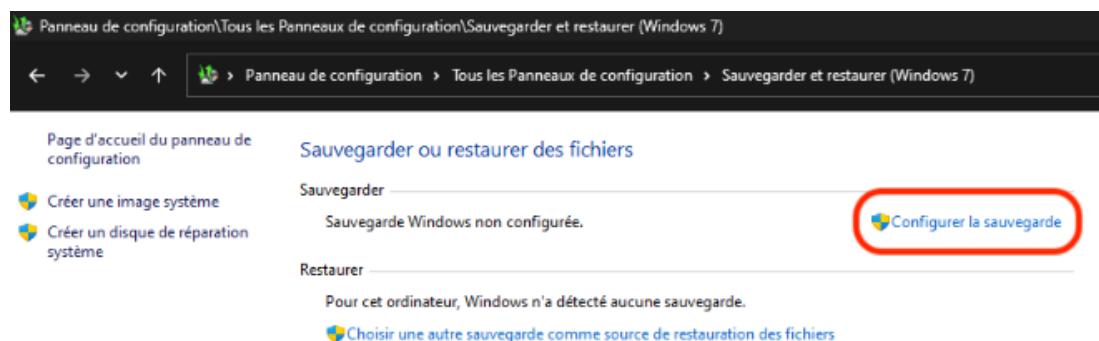
Dans la fenêtre de connexion, je sélectionne « utilisateur enregistré », j'entre l'utilisateur créé précédemment avec son mot de passe et je clique sur « connecter », Time Machine commence alors la sauvegarde.



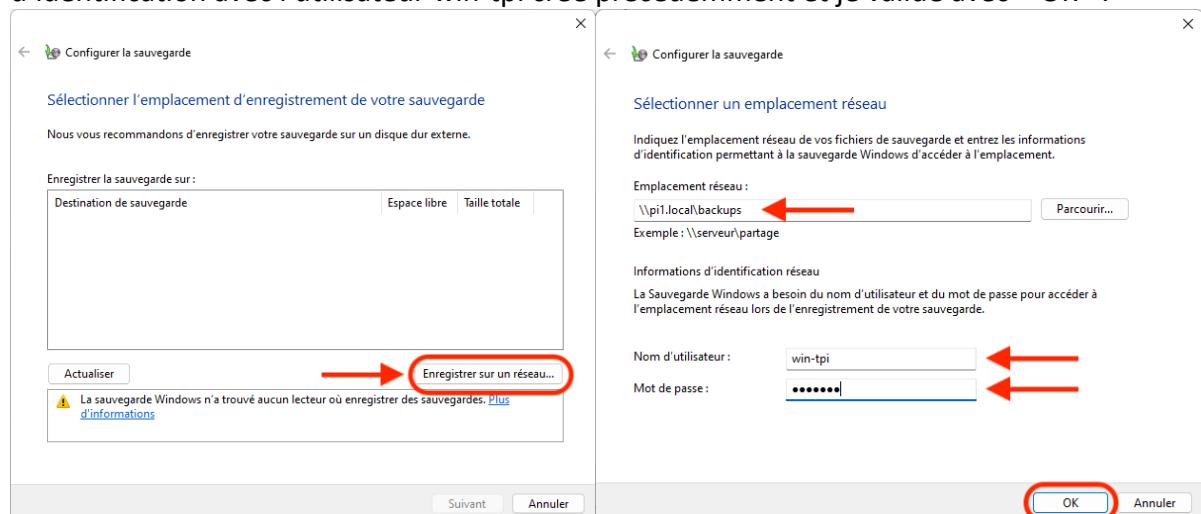
Pour Windows, j'ai ouvert l'ancien panneau de configuration et cliqué sur « Sauvegarder et restaurer (Windows 7) ».



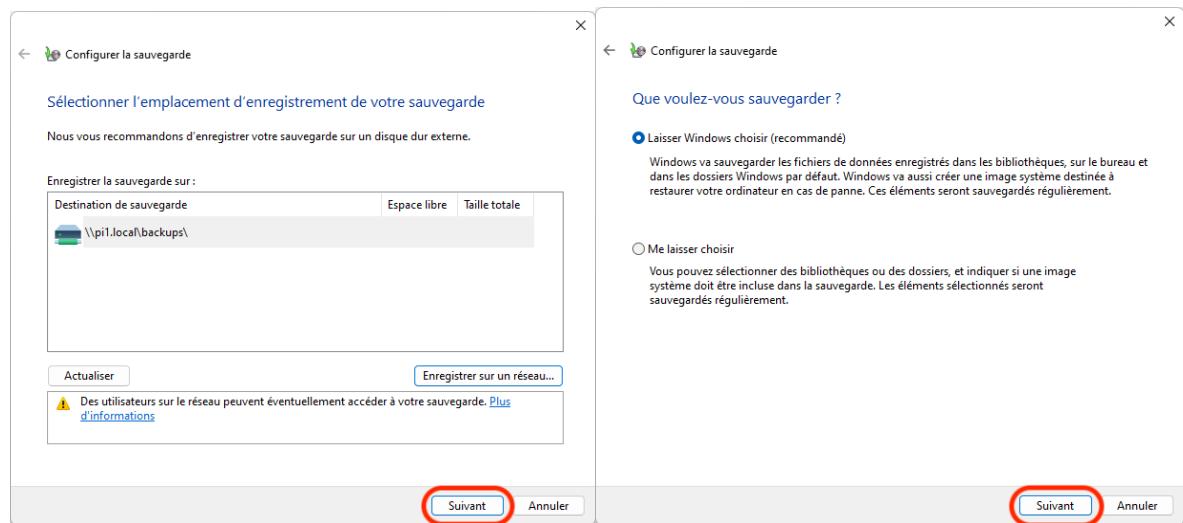
Ensuite j'ai cliqué sur « Configurer la sauvegarde ».



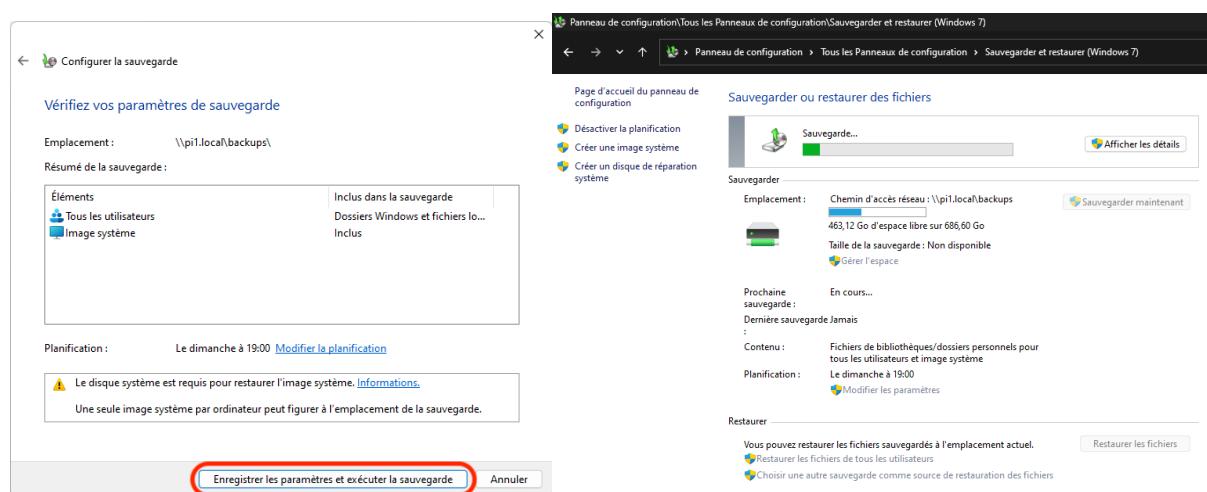
Un pop-up me demande alors où sauvegarder, je sélectionne « Enregistrer sur un réseau », j'indique l'emplacement réseau \\\pi1.local\backups et je remplis les informations d'identification avec l'utilisateur win-tpi créé précédemment et je valide avec « OK ».



L'emplacement réseau s'est bien rajouté, je clique sur « suivant », je laisse les paramètres par défaut et reclique sur « suivant ».



J'enregistre les paramètres et la sauvegarde se lance.



### 3.1.4 Mise en place de la réPLICATION des données avec Rsync

Sur le Raspberry Pi2, j'ai installé Rsync directement avec apt.

```
sudo apt -y install rsync
```

Ensuite j'ai formaté le disque dur en exfat car j'ai trouvé plus pratique de le formater dans ce système de fichiers pour plus de praticité en cas de panne majeur du disque du Raspberry Pi1 où l'on devrait accéder au disque rapidement depuis n'importe quelle OS étant donnée qu'exfat est supporté par la majorité des systèmes d'exploitation actuel.

Pour éviter de devoir monter le disque à chaque démarrage j'ai ajouté une ligne au fstab en indiquant quel disque avec quel format et où monter dans le disque dès le démarrage du système.

```
/dev/sda1 /mnt/sda1/ exfat defaults,noatime 0 1
```

Ensuite, j'ai ajouté une ligne au fichier crontab qui va s'exécuter toutes les heures pour copier les fichiers du disque du Pi1 sur le disque du Pi2 avec Rsync.

```
0 * * * *sudo rsync -rvzu homepi@pi1.local:/mnt/sda1/ /mnt/sda1/
```

### 3.1.5 Mise en place des containers Docker

Sur le Pi3, j'ai ajouté des nouvelles clés GPG et installé Docker avec apt en suivant la documentation de Docker.

Mise à jour et ajout de la clé gpg.

```
sudo apt -y update
sudo apt -y install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

Configuration du repository Docker.

```
echo "deb      [arch=$(dpkg --print-architecture)      signed-
by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/debian  $(lsb_release -cs)
stable" | sudo tee /etc/apt/sources.list.d/docker.list >
/dev/null
sudo apt -y update
```

Installation des derniers paquets pour que Docker fonctionne.

```
sudo apt -y install docker-ce docker-ce-cli containerd.io
docker-buildx-plugin docker-compose-plugin
```

Vérification de l'installation de Docker en lançant le container « hello-world » pour démontrer que Docker est opérationnel.

```
sudo docker run hello-world
```

Ensuite j'ai créé un fichier docker-compose.yml qui va ajouter et configurer tous les containers automatiquement.

```
version: '3'
services:
  apache:
    image: httpd:latest
    container_name: apache
    ports:
```

```

        - "80:80"
volumes:
  - ./apache:/usr/local/apache2/htdocs
  - ./apache/httpd.conf:/usr/local/apache2/conf/httpd.conf
restart: unless-stopped

webmin:
  container_name: webmin
  build:
    context: .
    dockerfile: Dockerfile
  restart: unless-stopped
  ports:
    - "10000:10000"
    - "9631:631"

# --- Optionnel ---

watchtower:
  image: containrrr/watchtower
  container_name: watchtower
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
  restart: unless-stopped

portainer:
  image: portainer/portainer-ce:latest
  container_name: portainer
  ports:
    - "8000:8000"
    - "9443:9443"
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
    - ./portainer_data:/data
  restart: unless-stopped

```

Pour Apache j'ai pu prendre l'image officielle du Docker Hub et préconfiguré qu'à la création du container qu'il mappe une page web de test et qu'il importe le fichier de configuration de base d'Apache. Ainsi, en allant sur <http://pi3.local/> je peux accéder à ma page web de test.

J'ai choisi une approche optimisée pour utiliser Webmin sur Raspberry Pi (arm64). Après avoir examiné les différentes options, j'ai opté pour une image Debian slim, jugée plus adaptée à l'installation de Webmin. Cette décision a été motivée par le souci de garantir une intégration fluide, conforme aux demandes du projet. Bien que l'utilisation de Webmin via Docker ne soit pas la solution la plus intuitive, cette configuration offre une base solide pour tirer le meilleur parti de cette application, même dans un environnement conteneurisé.

En plus de l'installation d'Apache et Webmin, j'ai également ajouté deux conteneurs supplémentaires pour faciliter la gestion et la maintenance de l'environnement. Le premier, Portainer, est un outil de gestion de conteneurs Docker offrant une interface utilisateur

conviviale pour surveiller et gérer les conteneurs, les images et les réseaux. Son interface graphique intuitive permet une gestion simplifiée, notamment la création, la modification et la suppression de conteneurs, ainsi que la surveillance des performances et des ressources. Le second conteneur ajouté est Watchtower, un outil pratique pour la mise à jour automatique des conteneurs Docker. Watchtower surveille les images Docker en cours d'exécution et met à jour automatiquement les conteneurs lorsque de nouvelles versions des images sont disponibles, garantissant ainsi la sécurité et la stabilité du système sans nécessiter d'intervention manuelle. Ces ajouts enrichissent l'infrastructure en offrant des fonctionnalités essentielles pour une gestion efficace et simplifiée des applications conteneurisées.

### 3.1.6 Création page web pour la documentation avec certificat SSL

Dans le cadre du TPI, le cahier des charges requiert que le site contenant la documentation utilisateur fonctionnant sur Apache soit sécurisé avec un certificat SSL Let's Encrypt. Cependant, pour des raisons de sécurité et de politique interne à l'EPFL, cette configuration n'est actuellement pas possible, car l'EPFL détient sa propre autorité de certification. Néanmoins, voici une méthode alternative pour parvenir à sécuriser le site.

#### Certificat auto-signé :

Tout d'abord, je configure Apache avec un certificat auto-signé utilisant le nom de domaine "pi3.local". Bien que ce certificat ne soit pas émis par une autorité de certification reconnue, il fournit une couche de chiffrement pour sécuriser les communications. Voici la commande utilisée :

```
COUNTRY="CH"
STATE="Vaud"
CITY="Lausanne"
ORGANIZATION="EPFL"
ORG_UNIT="ENAC-IT"
COMMON_NAME="pi3.local"
EMAIL="rocco.ronzano@epfl.ch"

sudo openssl req -x509 -nodes -days 365 -newkey rsa:4096 -keyout
/home/pi3/HOMEpi/docker/apache/server.key -out
/home/pi3/HOMEpi/docker/apache/server.crt -subj
"/C=$COUNTRY/ST=$STATE/L=$CITY/O=$ORGANIZATION/OU=$ORG_UNIT/CN
=$COMMON_NAME/emailAddress=$EMAIL"
```

Ensuite j'édite le fichier httpd-ssl.conf pour activer le support du site en SSL.

```
# General setup for the virtual host
DocumentRoot "/usr/local/apache2/htdocs"
ServerName pi3.local:443
ServerAdmin rocco.ronzano@epfl.ch
```

Et le fichier httpd.conf pour prendre en charge le fichier de configuration httpd-ssl.conf en décommentant ces lignes :

```
LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
LoadModule ssl_module modules/mod_ssl.so
```

```
LoadModule rewrite_module modules/mod_rewrite.so
Include conf/extra/httpd-ssl.conf
```

Pour éviter de devoir rentrer manuellement « https » dans la barre de navigation du navigateur, j'ai activé la redirection automatique en https :

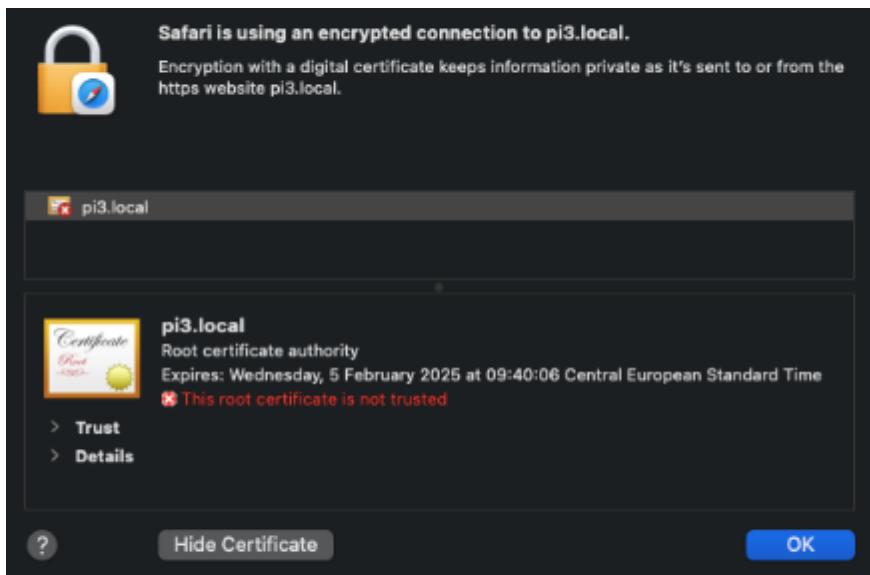
```
<Directory "/usr/local/apache2/htdocs">
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted

    # Redirection HTTP vers HTTPS
    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule ^ https:// %{HTTP_HOST}%{REQUEST_URI} [L,R=301]
</Directory>
```

Après avoir enregistré les modifications des fichiers de configuration d'Apache je vais éditer le docker-compose.yml pour que tout s'importe directement dès le déploiement du container.

```
services:
  apache:
    image: httpd:latest
    container_name: apache
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - ./apache:/usr/local/apache2/htdocs
      - ./apache/httpd.conf:/usr/local/apache2/conf/httpd.conf
      - ./apache/httpd-ssl.conf:/usr/local/apache2/conf/extra/httpd-ssl.conf
      - ./apache/server.crt:/usr/local/apache2/conf/server.crt
      - ./apache/server.key:/usr/local/apache2/conf/server.key
    restart: unless-stopped
```

Si je déploie le site je peux voir que le certificat a bien été importé.



### Certificat Let's Encrypt ! :

Pour la mise en place d'un certificat Let's Encrypt, j'ai utilisé l'un de mes propres domaines personnels, "homepi.ronzano.ch". Cette approche demande d'avoir accès à un nom de domaine personnel pour effectuer des modifications dans le panneau de ressources du domaine.

Pour commencer, j'ai installé au préalable certbot via apt qui est l'outil pour créer des certificats avec let's encrypt, j'ai ensuite lancé la commande « sudo certbot -d homepi.ronzano.ch certonly –manual –preferred-challenge dns ». certbot demande alors de renseigner mon adresse mail pour des conditions de sécurité et de renouvellement, il demande ensuite d'accepter les termes de service ce à quoi je réponds « y », il demande si je veux partager mon mail à une mailing liste pour Electronic Frontier Foundation ce à quoi je réponds « n » et finalement certbot me demande de déployer un enregistrement TXT DNS

« \_acme-challenge.homepi.ronzano.ch » avec la valeur en dessous.

```
pi3@pi3:~ $ sudo certbot -d homepi.ronzano.ch certonly --manual --preferred-challenge dns ←
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security notices)
(Enter 'c' to cancel): rocco.ronzano@epfl.ch ←

-----
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.3-September-21-2022.pdf. You must
agree in order to register with the ACME server. Do you agree?
-----
(Y)es/(N)o: y ←

-----
Would you be willing, once your first certificate is successfully issued, to
share your email address with the Electronic Frontier Foundation, a founding
partner of the Let's Encrypt project and the non-profit organization that
develops Certbot? We'd like to send you email about our work encrypting the web,
EFF news, campaigns, and ways to support digital freedom.
-----
(Y)es/(N)o: n ←
Account registered.
Requesting a certificate for homepi.ronzano.ch

-----
Please deploy a DNS TXT record under the name:
_acme-challenge.homepi.ronzano.ch. ←

with the following value:
oigogAqFZofOWHIU3fjXYIq15zMyLewAl4Ip9UrQLz4 ←

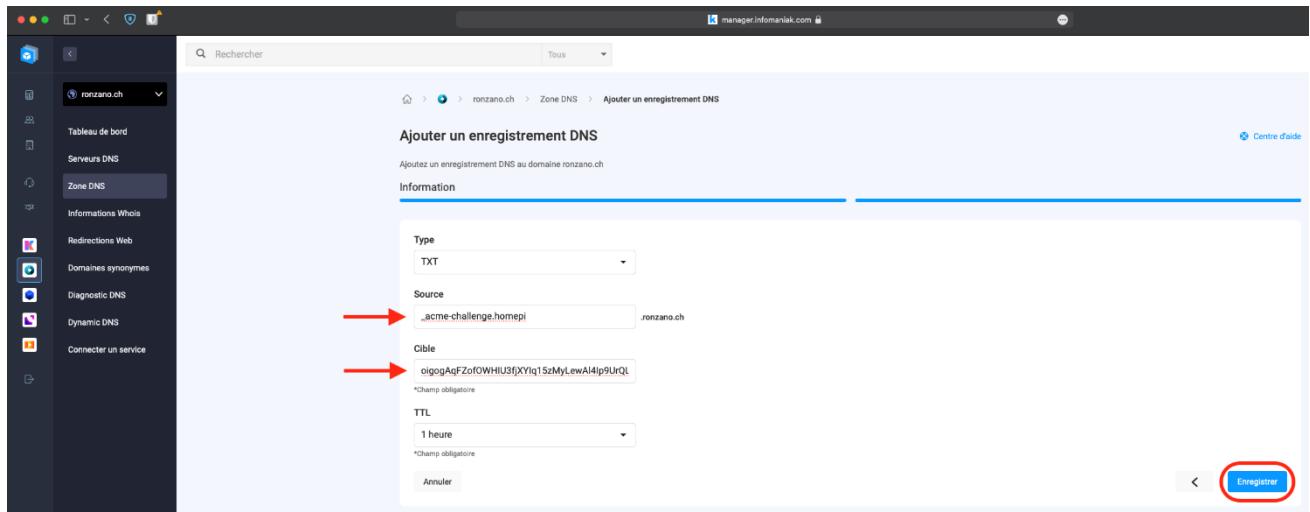
Before continuing, verify the TXT record has been deployed. Depending on the DNS
provider, this may take some time, from a few seconds to multiple minutes. You can
check if it has finished deploying with aid of online tools, such as the Google
Admin Toolbox: https://toolbox.googleapps.com/apps/dig/#TXT/_acme-challenge.homepi.ronzano.ch.
Look for one or more bolded line(s) below the line 'ANSWER'. It should show the
value(s) you've just added.

-----
Press Enter to Continue
```

Maintenant je peux me rendre sur l'interface de mon fournisseur de noms de domaine. Pour mon cas c'est Infomaniak, je me rends alors dans mes domaines et je choisi mon domaine ronzano.ch. Dans la section « Zone DNS » je clique sur « Ajouter un enregistrement ».

The screenshot shows the Infomaniak DNS management interface. On the left, a sidebar lists domain management options: Tableau de bord, Serveurs DNS, Zone DNS (which is highlighted with a red circle), Informations Whois, Redirections Web, Domaines synonymes, Diagnostic DNS, Dynamic DNS, and Connecter un service. The main area is titled "Gérer la zone DNS de votre domaine" and shows a message about the risks of manipulating DNS records. Below this, there are tabs for "Vue simplifiée" and "Vue avancée". A table displays 22 existing DNS records. At the bottom of the table, a blue button labeled "Ajouter un enregistrement" is circled in red.

Je sélectionne « TXT » dans la liste et je rajoute dans le champ « source » l'adresse « \_acme-challenge.homepi.ronzano.ch » que certbot m'a donné et la valeur donnée plus bas et je valide mon entrée avec « Enregistrer ».



Je peux voir que l'entrée « TXT » a été validée. Il ne manque plus qu'à attendre que l'entrée ait été mise à jour sur les serveurs du fournisseur de domaine ou vérifier avec Google Admin Toolbox :

[https://toolbox.googleapps.com/apps/dig/#TXT/\\_acme-challenge.homepi.ronzano.ch](https://toolbox.googleapps.com/apps/dig/#TXT/_acme-challenge.homepi.ronzano.ch).

Où il devrait avoir la valeur enregistrée précédemment.



Sur Google Admin Toolbox je peux voir que l'entrée a été mise à jour avec succès sur les serveurs d'Infomaniak.

A	AAAA	ANY	CAA	CNAME	DNSKEY	DS	MX	NS	PTR	SOA	SRV	TLSA	TSIG	TXT
														TXT

**TXT**

**Nom:** \_acme-challenge.homepi.ronzano.ch

**TTL:** 58 minutes 43 seconds

**VALUE:** "oigogAqFZofOWHIU3fjXYIq15zMyLewAl4Ip9UrQLz4"

Je peux désormais appuyer sur « entrée » pour valider la suite des opérations avec certbot. Je vois que certbot à réussi trouver mon champ TXT et a réussi à créer le certificat et la clé liée.

```
Press Enter to Continue

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/homepi.ronzano.ch/fullchain.pem ←
Key is saved at:          /etc/letsencrypt/live/homepi.ronzano.ch/privkey.pem ←
This certificate expires on 2024-05-06.
These files will be updated when the certificate renews.

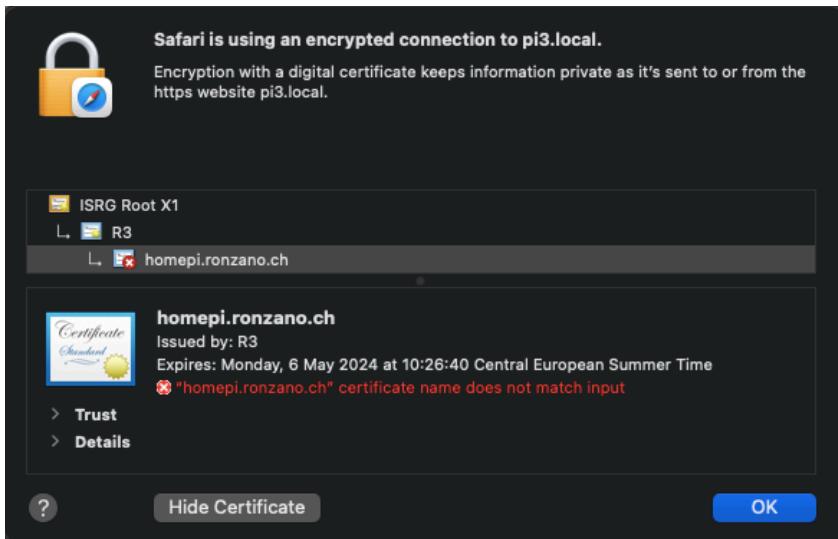
NEXT STEPS:
- This certificate will not be renewed automatically. Autorenewal of --manual certificates requires the use of an authentication hook script (--manual-auth-hook) but one was not provided. To renew this certificate, repeat this same certbot command before the certificate's expiry date.

-----  
If you like Certbot, please consider supporting our work by:  
* Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate  
* Donating to EFF: https://eff.org/donate-le  
-----  
pi3@pi3:~ $
```

Je peux donc adapter mon docker-compose pour importer le certificat et la clé.

```
services:
  apache:
    image: httpd:latest
    container_name: apache
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - ./apache:/usr/local/apache2/htdocs
      - ./apache/httpd.conf:/usr/local/apache2/conf/httpd.conf
      - ./apache/httdp-ssl.conf:/usr/local/apache2/conf/extra/httpd-ssl.conf
      - /etc/letsencrypt/live/homepi.ronzano.ch/fullchain.pem:/usr/local/apache2/conf/server.crt
      - /etc/letsencrypt/live/homepi.ronzano.ch/privkey.pem:/usr/local/apache2/conf/server.key
    restart: unless-stopped
```

Si je déploie le site je peux voir que le certificat a bien été importé.



Cette approche permet de démontrer la mise en œuvre réussie d'un certificat SSL Let's Encrypt dans un environnement externe à l'EPFL, offrant ainsi une alternative viable pour assurer la sécurité des communications sur le site Apache.

### 3.1.7 Mise en place d'Ansible

Pour Ansible j'ai commencé par l'installer sur tous les Pi via apt.

```
sudo apt -y install ansible
```

Ensuite j'ai créé plusieurs playbook pour différents cas :

- Master\_playbook\_pi : Regroupement de playbook, permet d'installer tous les Pi à la suite.
- Playbook\_install\_pi1 : Permet d'installer Pi1 dans son état d'origine avec le service d'impression et le nas.
- Playbook\_install\_pi2 : Permet d'installer Pi2 dans son état d'origine avec Rsync.
- Playbook\_install\_pi3 : Permet d'installer Pi3 dans son état d'origine avec Docker.
- Playbook\_update : Permet de mettre à jour tous les Pi.
- Playbook\_cups : Permet de mettre en place le service d'impression avec CUPS.
- Playbook\_nas : Permet de mettre en place le nas.
- Playbook\_rsync : Permet de mettre en place Rsync.
- Playbook\_docker : Permet de mettre en place Docker.

Tous ces playbook sont réunis dans un script HOMEPi.sh qui permet de les lancer facilement depuis n'importe quelle machine connectée au même réseau avec Ansible installé.

### 3.2 Description des tests effectués

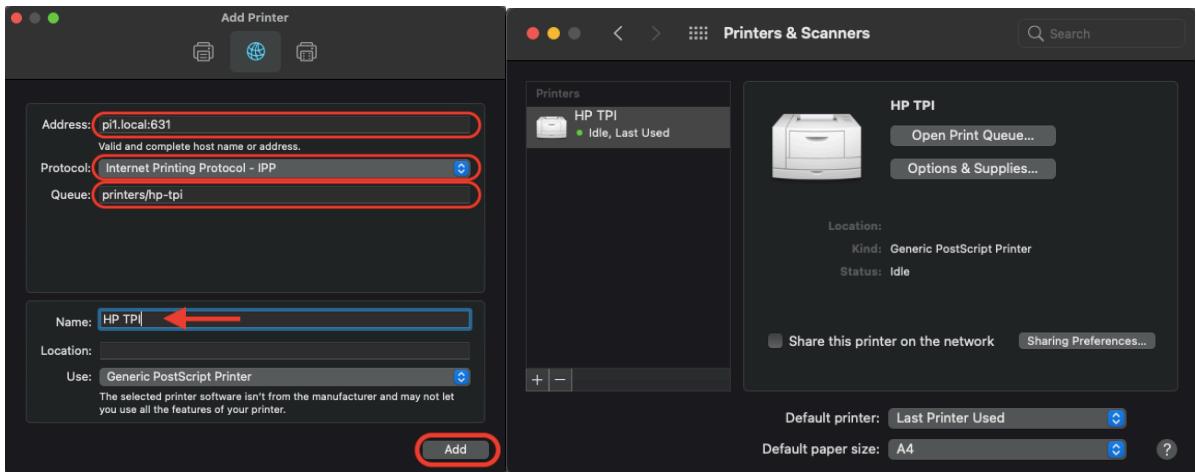
Chaque composant du projet sera soumis à des tests rigoureux pour garantir son bon fonctionnement et sa conformité aux spécifications.

Test effectué	Description du test	Résultat attendu	Résultat obtenu
Impression depuis les laptops	Impression d'une page de test depuis chaque laptop	Les laptops envoient une page de test et s'imprime	Les laptops ont envoyé les pages de tests et sont imprimées
Accès au NAS	Les laptops peuvent se connecter avec leur utilisateur et accéder au partage de fichiers lié à leur utilisateur	Les laptops arrivent à se connecter avec leur utilisateur et accéder au partage de fichiers lié à leur utilisateur	Les laptops arrivent à se connecter avec leur utilisateur et accéder au partage de fichiers lié à leur utilisateur
Accès au partage pour sauvegarder les laptops	Les laptops peuvent se connecter avec leur utilisateur et accéder au partage de fichiers pour sauvegarder la machine	Les laptops arrivent à se connecter avec leur utilisateur et accéder au partage de fichiers pour sauvegarder la machine	Les laptops arrivent à se connecter avec leur utilisateur et accéder au partage de fichiers pour sauvegarder la machine
RéPLICATION DES DONNÉES	Le Pi2 peut se connecter au Pi1 et répliquer les données à partir d'une tâche cron	Le Pi2 arrive à se connecter au Pi1 et répliquer les données à partir d'une tâche cron	Le Pi2 arrive à se connecter au Pi1 et répliquer les données à partir d'une tâche cron
ACCÈS AU SITE WEB	Les clients du réseau local peuvent accéder au site web en https	Les clients du réseau local arrivent à accéder au site web en https	Les clients du réseau local arrivent à accéder au site web en https
AUTOMATISATION AVEC ANSIBLE	Les playbooks Ansible peuvent accéder aux Pi et exécuter les différentes tâches	Les playbooks Ansible arrivent à accéder aux Pi et exécuter les différentes tâches	Certains playbooks Ansible arrivent à accéder aux Pi mais n'arrivent pas à exécuter les différentes tâches correctement

#### 3.2.1 Tests de configuration de base

Sur les laptops j'ai pu ajouter l'imprimante en Bonjour et par IP ainsi qu'imprimer sans problème.

Sur le laptop macOS

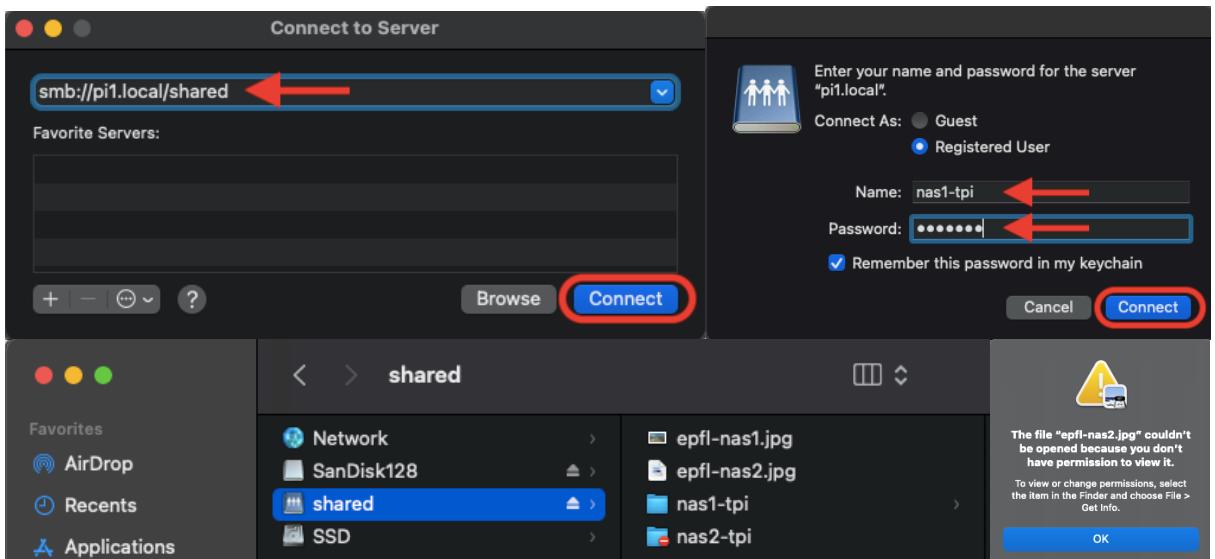


Sur le laptop Windows

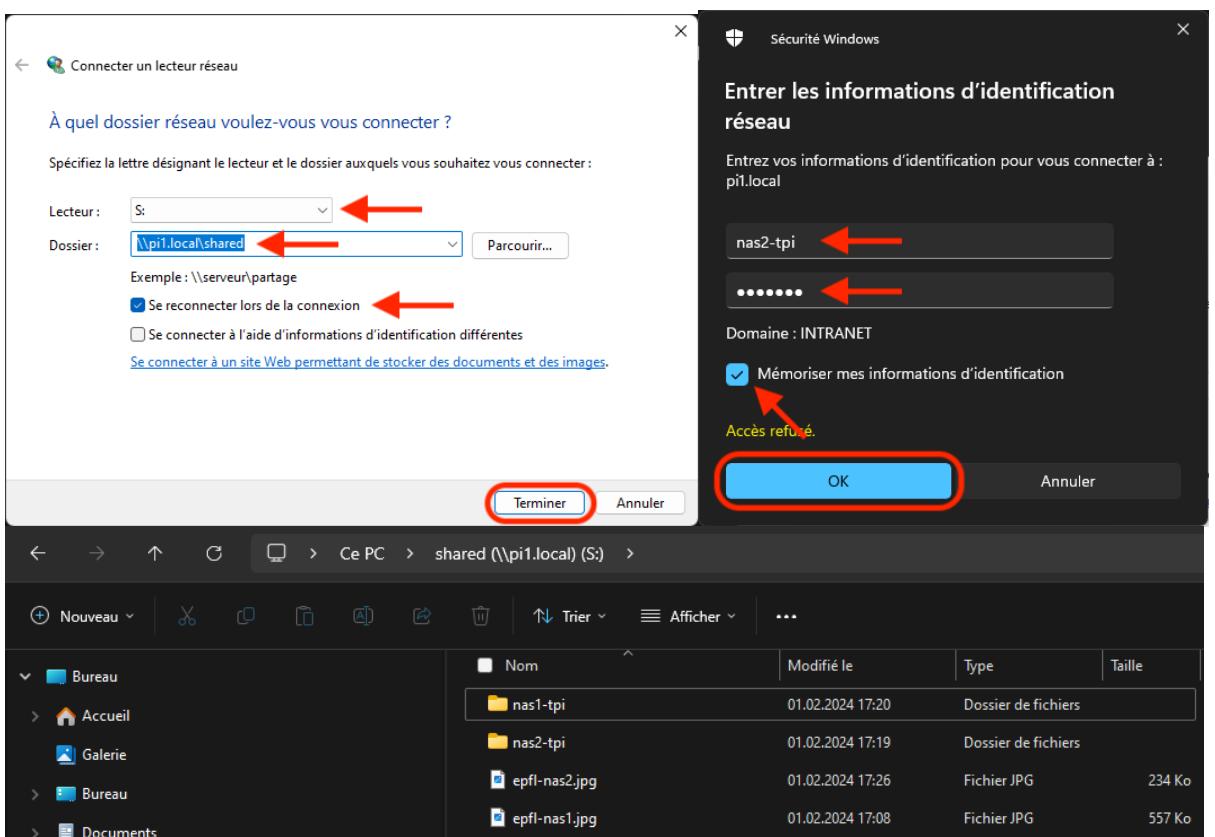
Pour le NAS j'ai pu me connecter avec différents utilisateurs sur les laptops et ainsi créer des dossiers et des fichiers qui n'étaient lisibles qu'avec l'utilisateur qui avait les droits.

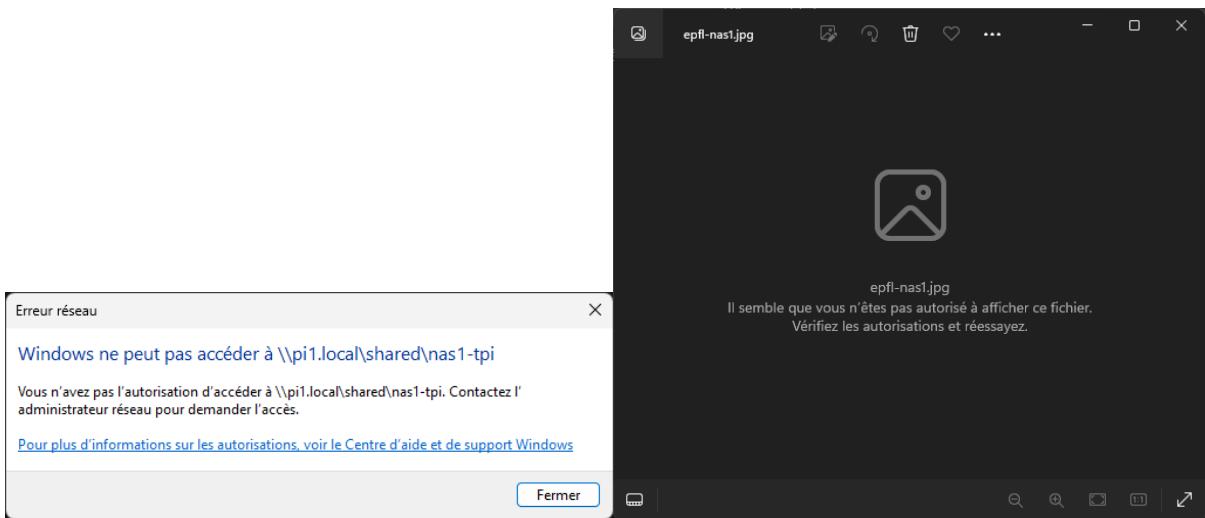
Nom utilisateur	Accès unique aux dossiers	Faisant parti du groupe
homepi	Propriétaire du disque	homepi, backups, shared, timemachine, nas1-tpi, nas2-tpi, mac-tpi et win-tpi
nas1-tpi	shared, nas1-tpi	nas1-tpi, shared
nas2-tpi	shared, nas2-tpi	nas2-tpi, shared
mac-tpi	TimeMachine	mac-tpi, timemachine
win-tpi	backups	win-tpi, backups

Sur le laptop macOS j'étais connecté avec l'utilisateur nas1-tpi, je n'avais accès qu'au dossier nas1-tpi et à l'image nas1-tpi.jpg.



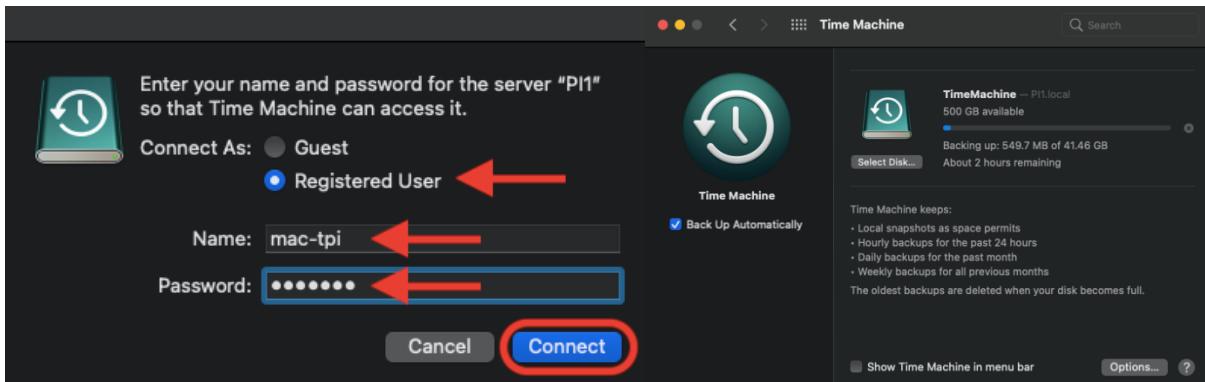
Sur le laptop Windows j'étais connecté avec l'utilisateur nas2-tpi, je n'avais accès qu'au dossier nas2-tpi et à l'image nas2-tpi.jpg.



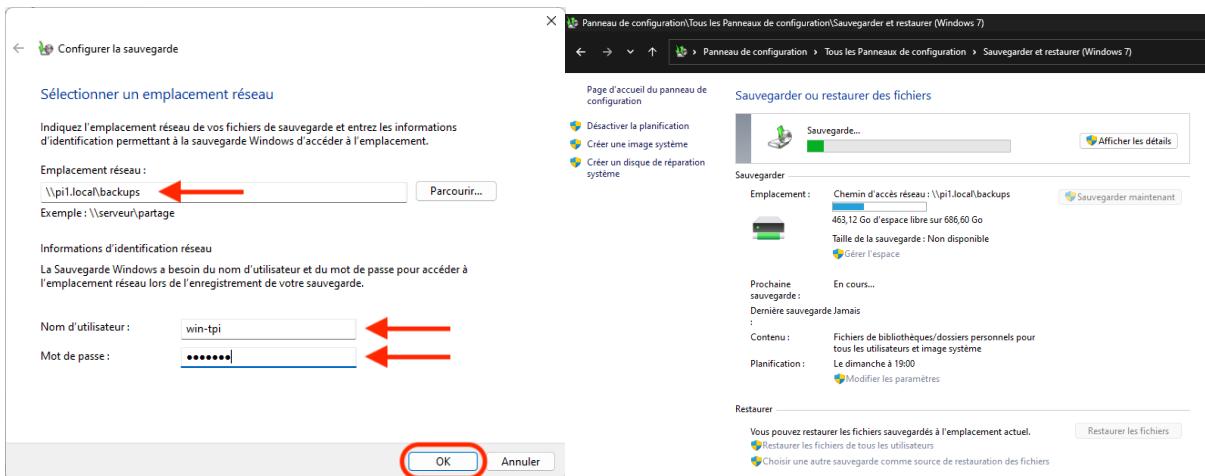


Sur chaque laptop j'ai pu configurer une sauvegarde dans les dossiers backups et TimeMachine du NAS.

Sur le laptop macOS j'ai pu ajouter un disque de sauvegarde réseau via Time Machine et me connecter avec l'utilisateur mac-tpi.

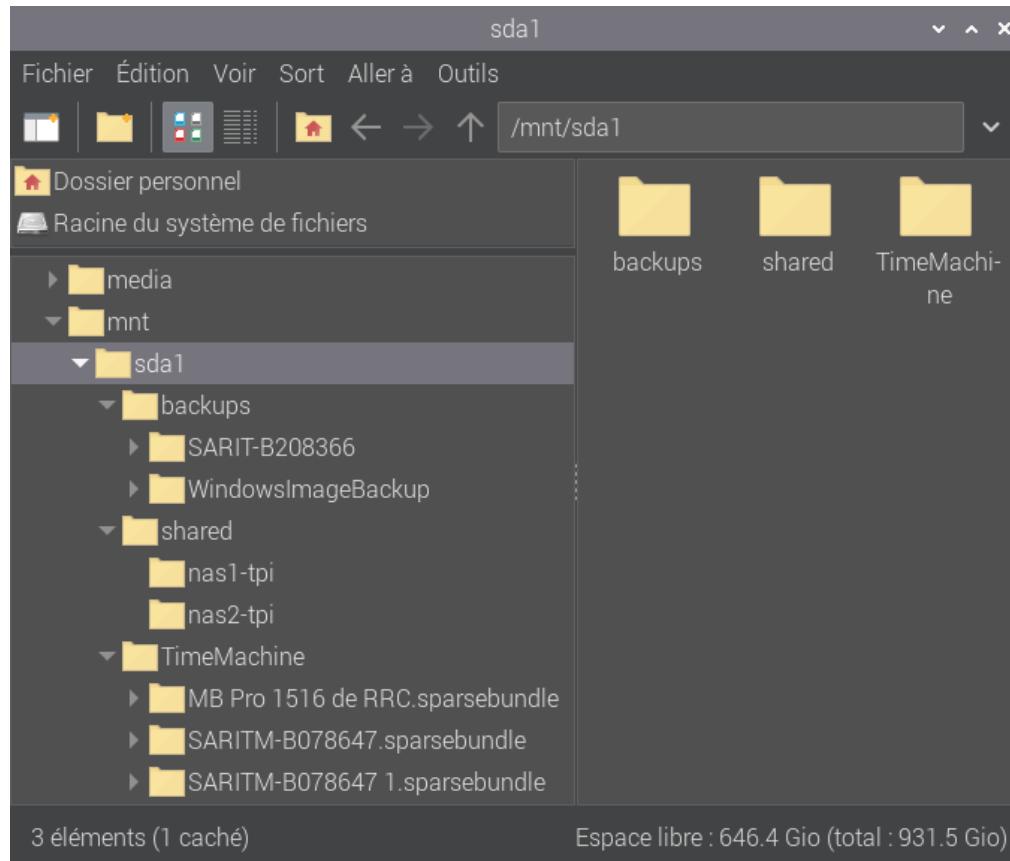


Sur le laptop Windows j'ai pu ajouter un disque de sauvegarde réseau via l'utilitaire « Sauvegarder et restaurer (Windows 7) » et me connecter avec l'utilisateur win-tpi.



### 3.2.2 Tests de réPLICATION DES DONNÉES

Sur le Raspberry Pi2 j'ai actuellement pu lancer la commande Rsync qui va copier toutes les données qui sont sur le disque du Pi1 sur le disque du Pi2.



### 3.2.3 Tests d'accès au site web

Depuis n'importe quel client du réseau local j'arrive à accéder la page web.

### 3.2.4 Tests Ansible

Sur un des Raspberry Pi j'ai recréé une carte SD avec Raspberry Pi OS avec les paramètres par défaut et j'ai pu lancer le playbook `playbook_update` et mettre à jour tout le système du Pi sans soucis.

Sur le même Pi j'ai pu lancer le playbook `playbook_docker` et Ansible a installé Docker, générés un certificat SSL et lancé un Docker compose.

## 3.3 Erreurs restantes

Bien que la majeure partie du projet ait été réalisée avec succès, quelques erreurs et ajustements restent à prendre en compte :

**Vérification des Playbooks Ansible** : Certains Playbooks Ansible nécessitent une vérification supplémentaire pour s'assurer que toutes les configurations sont correctement appliquées. Des tests approfondis doivent être effectués pour garantir le bon fonctionnement de l'automatisation des tâches système.

**Correction dans l'entrypoint.sh du docker-compose** : Une erreur mineure a été identifiée dans l'entrypoint.sh du docker-compose, où l'import du fichier `cupsd.conf` dans le container Webmin ne se déroule pas correctement. Cette erreur doit être corrigée pour garantir que la configuration du serveur d'impression soit correctement appliquée et fonctionnelle.

Ces erreurs restantes nécessitent une attention immédiate afin d'assurer le bon fonctionnement et la stabilité globale du projet. Une fois ces ajustements effectués, le projet sera prêt à être livré dans sa version finale.

## 3.4 Liste des documents fournis

Une liste complète des documents fournis avec le produit final sera établie, notamment :

- Planification finale
- Rapport de Projet
- Journal de travail
- Ensemble des scripts et fichiers de configuration
- Documentation pour l'utilisateur « Utilisation simplifiée d'un Raspberry Pi à domicile ».

## 4 Conclusions

### 4.1 Objectifs atteints / non-atteints

Les objectifs fixés initialement ont été évalués pour déterminer dans quelle mesure ils ont été atteints.

Objectifs atteints :

- L'élaboration de la procédure "Utilisation simplifiée d'un Raspberry Pi à domicile" a été réalisée avec succès, répondant aux besoins des utilisateurs non techniques.
- Les services avancés ont été mis en place conformément aux spécifications, avec une gestion automatisée et une répartition cohérente sur les trois Raspberry Pi.

- La documentation des services, la configuration du serveur web avec un certificat SSL, et l'intégration de Webmin ont été achevées selon les exigences.

Objectifs non-atteints :

- Certaines vérifications des Playbooks Ansible nécessitent encore d'être effectuées pour s'assurer que toutes les configurations sont correctement appliquées.
- Une erreur mineure dans l'entrypoint.sh du docker-compose doit être corrigée pour garantir une importation correcte du fichier cupsd.conf dans le container Webmin.

## 4.2 Points positifs / négatifs

Points positifs :

- La procédure "Utilisation simplifiée d'un Raspberry Pi à domicile" répond aux besoins des utilisateurs ciblés, offrant une solution conviviale et fonctionnelle.
- L'automatisation des tâches avec Ansible a permis de minimiser les actions manuelles et d'assurer une gestion efficace des services.
- La documentation complète et la configuration du serveur web avec certificat SSL ont été réalisées avec succès.

Points négatifs :

- Des erreurs mineures subsistent dans certains scripts et configurations, nécessitant une correction pour garantir le bon fonctionnement du système.
- La vérification complète des Playbooks Ansible et la correction de l'erreur dans l'entrypoint.sh du docker-compose restent des tâches à accomplir.

## 4.3 Difficultés particulières

Une des principales difficultés rencontrées durant le projet a été liée à l'utilisation d'Ansible pour l'automatisation des tâches. La courbe d'apprentissage des modules spécifiques aux tâches s'est avérée plus longue que prévu, ce qui a entraîné un retard dans la mise en œuvre de certaines configurations. Comprendre le fonctionnement et la syntaxe des modules a demandé plus de temps que prévu, ce qui a impacté la planification initiale du projet.

La complexité de certaines configurations, en particulier celles impliquant des interactions entre plusieurs services et composants, a également posé des défis. La nécessité de déboguer et d'optimiser ces configurations pour assurer un fonctionnement harmonieux des playbooks a demandé des efforts supplémentaires et a contribué à allonger la durée de la mise en place d'Ansible au projet.

Malgré ces difficultés, des progrès significatifs ont été réalisés dans la compréhension et la maîtrise d'Ansible, et les défis rencontrés ont permis d'acquérir de nouvelles compétences et de consolider les connaissances existantes en administration système et en automatisation des tâches.

## 4.4 Suites possibles pour le projet

Pour assurer la finalisation complète du projet, il est nécessaire de procéder comme suit :

- Effectuer une vérification approfondie des Playbooks Ansible pour s'assurer que toutes les configurations sont correctement appliquées.

- Corriger l'erreur dans l'entrypoint.sh du docker-compose pour garantir une importation correcte du fichier cupsd.conf dans le container Webmin.
- Effectuer des tests exhaustifs pour garantir le bon fonctionnement de l'ensemble du système.
- Envisager des améliorations potentielles, telles que l'ajout de fonctionnalités supplémentaires ou l'optimisation des performances, pour répondre aux besoins futurs.
- À titre d'évolution possible, la mise en place d'un service tel que Nagios pourrait être mis en place pour afin d'assurer une reprise de failover.

#### 4.5 Bilan personnel

Dans le cadre de ce projet, la plupart des objectifs principaux du cahier des charges ont été atteints et la planification a été quasiment tenue jusqu'au bout à l'exception de l'automatisation via Ansible où il manque encore de vérifier certains playbooks et valider leurs bons fonctionnements. La compréhension des modules m'a pris plus de temps que prévu et je me suis volontairement coupé pour rester dans les temps de la planification initiale et ne pas déborder sur le temps mis à disposition pour avancer la documentation.

Malgré ces défis, ce projet m'a offert une opportunité précieuse d'acquérir de nouvelles compétences et de consolider mes connaissances existantes. En particulier ma compréhension de la certification SSL, en apprenant à configurer et à gérer un certificat SSL pour sécuriser les communications sur un serveur web. De plus, j'ai aimé en apprendre plus sur l'utilisation d'Ansible pour l'automatisation des tâches système, ce qui m'a permis de mieux comprendre la gestion de configuration et d'orchestration des infrastructures. Ces apprentissages ont été essentiels pour surmonter les défis rencontrés tout au long du projet et ont renforcé ma confiance dans ma capacité à relever des défis techniques complexes à l'avenir.

J'aimerais souligner que j'ai eu l'occasion d'utiliser des outils tels que Webmin, Ansible et Docker au cours de mon projet. Ces outils offrent des fonctionnalités puissantes et sont largement employés dans le domaine de l'administration système et du déploiement d'applications. Cependant, j'ai réalisé que chacun de ces outils est si vaste et complexe qu'un TPI pourrait facilement être consacré à un seul de ces outils. Cette expérience m'a permis de comprendre l'étendue des technologies disponibles et la nécessité de se concentrer sur des domaines spécifiques pour acquérir une expertise approfondie.

## 5 Annexes

### 5.1 Résumé du rapport du TPI / version succincte de la documentation

#### Situation de départ

Le projet "Raspberry Pi : Utilisation à domicile et services avancés" a été initié en réponse à la demande croissante d'une solution polyvalente et performante basée sur Raspberry Pi. La situation initiale comprenait l'identification des besoins spécifiques des utilisateurs domestiques et professionnels, ainsi que la définition des exigences techniques telles que la sécurité, la gestion automatisée et la flexibilité de configuration.

#### Mise en œuvre

Installation et configuration de Raspberry Pi avec Raspberry Pi Imager :

- Pré-paramétrage des images avec le nom local, nom d'utilisateur, mot de passe, activation SSH et les locales.

Configuration du service d'impression avec CUPS :

- Édition du fichier de configuration cupsd.conf pour autoriser l'impression depuis tous les clients sur le réseau et l'accès à la page administrateur.

Configuration du partage de fichier SMB/CIFS avec Samba :

- Utilisation de Samba pour partager des dossiers sur le réseau via SMB/CIFS.

RéPLICATION DES DONNÉES :

- Utilisation de Rsync pour la réPLICATION DES DONNÉES UNIDIRECTIONNELLE.

Configuration de l'infrastructure de conteneurs avec Docker :

- Utilisation de Docker-Compose pour décrire et gérer l'infrastructure de conteneurs.
- Définition des services dans un fichier docker-compose.yml, spécifiant les dépendances entre les conteneurs et les paramètres de configuration.

Gestion des certificats SSL auto signé et avec Let's Encrypt :

- Intégration de certificat OpenSSL auto signé dans les playbooks Ansible pour automatiser la génération et le renouvellement des certificats SSL.

Automatisation des tâches avec Ansible :

- Utilisation d'Ansible pour orchestrer la configuration des Raspberry Pi et le déploiement des services.
- Utilisation de playbooks définis pour organiser et réutiliser des tâches communes, améliorant ainsi la modularité et la maintenabilité du système.

#### Résultats

Malgré les défis techniques rencontrés, le projet a abouti à la création d'une infrastructure fonctionnelle pour le cas d'utilisation à domicile. Les résultats comprennent la mise en place réussie de tous les services requis pour cette phase, notamment l'installation et la configuration de Raspberry Pi OS, du partage de fichiers et d'imprimantes, ainsi que de la solution de sauvegarde des postes clients. Cependant, l'automatisation des tâches avec Ansible n'a pas encore été finalisée. En résumé, la phase initiale du projet a été réalisée avec succès, répondant aux besoins domestiques avec une solution robuste et polyvalente basée sur Raspberry Pi.

## 5.2 Sources – Bibliographie

### Méthode de gestion de projet

- [https://fr.wikipedia.org/wiki/Gestion\\_de\\_projet](https://fr.wikipedia.org/wiki/Gestion_de_projet)
- <https://www.atlassian.com/fr/agile/project-management/project-management-intro>
- <https://blog-gestion-de-projet.com/gestion-de-projet/outils-methodes-gestion-projet/>
- <https://blog-gestion-de-projet.com/modele-en-cascade/>

### Logiciels utilisés

- <https://desktop.github.com/>
- <https://www.yworks.com/products/yed>
- <https://code.visualstudio.com/>

### Dépôt Github

- <https://github.com/>
- <https://github.com/roccom14/HOMEpi>

### Installation des Raspberry Pi

- <https://www.raspberrypi.com/>

### Configuration des Raspberry Pi

- <https://www.cups.org/>
- <https://www.samba.org/>
- <https://www.raspberrypi.com/tutorials/nas-box-raspberry-pi-tutorial/>

### Mise en place de Rsync

- <https://rsync.samba.org/>

### Mise en place de Docker

- <https://docker.com/>
- [https://hub.docker.com/\\_/httpd](https://hub.docker.com/_/httpd)
- <https://hub.docker.com/r/slab42/bind9.18>
- <https://hub.docker.com/r/containrrr/watchtower>
- <https://hub.docker.com/r/portainer/portainer-ce>
- <https://httpd.apache.org/>
- <https://webmin.com/>
- <https://containrrr.dev/watchtower/>
- <https://www.portainer.io/>

### Mise en place du certificat SSL :

- <https://www.openssl.org/>
- <https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-apache-in-ubuntu-20-04-fr>
- <https://certbot.eff.org/>
- <https://manager.infomaniak.com/>
- <https://toolbox.googleapps.com/apps/main/>

### Mise en place d'Ansible

- <https://docs.ansible.com/>
- [https://docs.ansible.com/ansible/latest/getting\\_started/get\\_started\\_playbook.html](https://docs.ansible.com/ansible/latest/getting_started/get_started_playbook.html)
- [https://docs.ansible.com/ansible/latest/inventory\\_guide/intro\\_inventory.html](https://docs.ansible.com/ansible/latest/inventory_guide/intro_inventory.html)
- <https://docs.ansible.com/ansible/latest/cli/ansible-playbook.html>

### Aide au développement

- <https://chat.openai.com>

### 5.3 Journal de travail

Journal de travail - Rocco Ronzano		
Heures	Nom	Liens, références, commentaires
29.01.24		
09:00 - 09:45	Analyse du cahier des charges	
09:45 - 10:20	Visite de Monsieur Serge Wenger	
10:20 - 12:30	Création planification, journal de travail, rapport de projet et analyse préliminaire	
12:30 - 13:30	Pause	
13:30 - 18:30	Planification initiale	
30.01.24		
09:00 - 09:45	Création schéma de conception	
09:45 - 12:00	Analyse préliminaire, analyse et conception	
12:00 - 13:00	Pause	
13:00 - 13:20	Création des cartes SD avec Raspberry Pi OS	Cartes créées avec succès
13:20 - 15:50	Mise en place du matériel et mise à jour des Raspberry Pi, prise de screenshots pour documentation utilisateur	
15:50 - 16:00	Création du repo git	<a href="https://github.com/Roccom14/HOMEpi">https://github.com/Roccom14/HOMEpi</a>
16:00 - 18:20	Édition du rapport de projet	
18:20 - 18:30	Édition du journal de travail et de la planification	
31.01.24		
09:00 - 11:20	Mise en place de Cups avec prise de screenshots	
11:20 - 11:30	Tests d'impression depuis les laptops	Impression fonctionnelle
11:30 - 12:00	Mise en place des disques pour les Raspberry Pi	
12:00 - 13:00	Pause	
13:00 - 15:40	Mise en place de Samba avec prise de screenshots	
15:40 - 16:00	Test d'accès de lecture et écriture de fichiers depuis les laptops	Accès aux partages fonctionnel
16:00 - 18:20	Édition du rapport de projet	
18:20 - 18:30	Édition du journal de travail et de la planification	
01.02.24		
09:00 - 10:00	Mise en place du montage automatique du disque dans le Pi1	
10:00 - 11:30	Création des dossiers de sauvegarde, édition du fichier de configuration Samba et création des différents utilisateurs	
11:30 - 12:00	Mise en place du dossier de sauvegarde visible pour TimeMachine	
12:00 - 13:00	Pause	
13:00 - 15:00	Tests de sauvegarde sur les laptops macOS et Windows	Sauvegarde fonctionnelle sur les laptops
15:00 - 15:30	Mise en place de la sauvegarde sur le laptop macOS avec prise de screenshot	
15:30 - 16:00	Mise en place de la sauvegarde sur le laptop Windows avec prise de screenshot	
16:00 - 18:20	Édition du rapport de projet	
18:20 - 18:30	Édition du journal de travail et de la planification	
02.02.24		
09:00 - 09:30	Formatage du disque et mise en place du montage automatique du disque dans le Pi2	
09:30 - 11:30	Mise en place de Rsync	
11:30 - 12:00	Tests de réplication des données sur le Pi2	
12:00 - 13:00	Pause	Erreurs de droits
13:00 - 15:30	Résolution des erreurs de droits des fichiers	
15:30 - 16:00	Tests de réplication des données sur le Pi2	Réplication des données fonctionnelle
16:00 - 18:20	Édition du rapport de projet	
18:20 - 18:30	Édition du journal de travail et de la planification	
05.02.24		
09:00 - 09:45	Installation de Docker sur pi3	
09:45 - 10:20	Recherche d'un container utilisable pour Webmin	
10:20 - 12:30	Création et configuration du docker-compose.yml pour tous les containers	
12:30 - 13:30	Pause	
13:30 - 15:00	Configuration d'Apache	Container Apache fonctionnel
15:00 - 16:00	Tentative de configuration de Webmin	Problèmes de configuration Webmin
16:00 - 18:20	Édition du rapport de projet	
18:20 - 18:30	Édition du journal de travail et de la planification	
06.02.24		
09:00 - 10:30	Installation et configuration d'Apache avec le certificat SSL auto signé + Screenshots	SSL auto-signé fonctionnel
10:30 - 11:20	Viste de Monsieur Nemanja Pantic pour le point de situation	
11:20 - 12:20	Installation et configuration d'apache avec le certificat Let's Encrypt + Screenshots	SSL Let's Encrypt fonctionnel
12:20 - 13:20	Pause	
13:20 - 16:00	Écriture de la documentation utilisateur "Utilisation simplifiée d'un Raspberry Pi à domicile"	
16:00 - 18:20	Édition du rapport de projet	
18:20 - 18:30	Édition du journal de travail et de la planification	

Journal de travail - Rocco Ronzano		
Heures	Nom	Liens, références, commentaires
		07.02.24
09:00 - 10:30	Création de playbook pour pi1	Finalisé, à tester
10:30 - 11:20	Création de playbook pour pi2	Finalisé, à tester
11:20 - 12:40	Création de playbook pour pi3	Finalisé, à tester
12:40 - 13:40	Pause	
13:40 - 18:10	Continuation des playbook et création des autres playbook pour les tâches uniques	Finalisé, à tester
18:10 - 18:20	Édition du rapport de projet	
18:20 - 18:30	Édition du journal de travail et de la planification	
		08.02.24
09:00 - 11:50	Continuation des playbook et correction des erreurs	
11:50 - 12:50	Test des playbook	Erreurs restantes dans les playbooks
12:50 - 13:50	Pause	
13:50 - 18:20	Édition du rapport de projet	
18:20 - 18:30	Édition du journal de travail et de la planification	
		09.02.24
09:00 - 12:20	Édition du rapport de projet	
12:20 - 13:20	Pause	
13:20 - 18:20	Édition du rapport de projet	
18:20 - 18:30	Édition du journal de travail et de la planification	
		12.02.24
09:00 - 12:15	Édition du rapport de projet	
12:15 - 13:15	Pause	
13:15 - 18:20	Édition du rapport de projet	
18:20 - 18:30	Édition du journal de travail, de la planification finale et envoi des documents finaux	

## 5.4 Glossaire

**64 bits** : Type d'architecture de processeur qui peut gérer des données et des instructions en blocs de 64 bits.

**Ansible** : Outil de gestion de configuration open-source utilisé pour automatiser le déploiement, la configuration et la gestion des logiciels.

**Apache** : Serveur web open-source populaire largement utilisé pour servir des sites web statiques et dynamiques.

**Arm64** : Architecture de processeur développée par ARM Holdings, utilisée principalement dans les appareils mobiles et les serveurs.

**Bonjour** : Protocole de réseau local développé par Apple permettant la découverte automatique de services et de périphériques sur un réseau local.

**Certbot** : Outil open-source permettant d'automatiser l'installation et le renouvellement des certificats SSL/TLS.

**Certificat SSL/TLS** : Fichier électronique utilisé pour sécuriser les communications sur Internet en chiffrant les données échangées entre un serveur web et un navigateur web.

**Clés SSH** : Paires de clés cryptographiques utilisées pour l'authentification dans le cadre d'une connexion SSH.

**Cron job** : Tâche planifiée sur les systèmes Unix-like pour exécuter des scripts ou des commandes à des intervalles spécifiés.

**CUPS** : Common Unix Printing System, un système de gestion d'impression en réseau utilisé pour configurer et gérer des imprimantes sur des systèmes d'exploitation de type UNIX.

**Docker** : Plateforme logicielle open-source permettant de créer, de tester et de déployer des applications dans des conteneurs.

**Failover** : Transition automatique vers un système de secours ou de reprise en cas de défaillance, assurant ainsi la continuité des services et des opérations.

**GitHub** : Plateforme de développement logiciel basée sur Git, utilisée pour l'hébergement et la gestion de projets informatiques.

**Groupadd** : Commande utilisée pour créer de nouveaux groupes d'utilisateurs dans un système UNIX.

**GPG** : GNU Privacy Guard, un logiciel de chiffrement et de signature numérique qui fournit une implémentation open-source du standard OpenPGP.

**HTTPS** : Hypertext Transfer Protocol Secure, une version sécurisée du protocole HTTP utilisé pour le transfert de données sur le web.

**NAS** : Network Attached Storage, dispositif de stockage de données connecté à un réseau qui permet le stockage et le partage de fichiers entre différents appareils.

**OpenSSL** : Bibliothèque open source offrant des implémentations de qualité des protocoles SSL/TLS et des outils de cryptographie, largement utilisée pour sécuriser les communications sur Internet.

**Permissions** : Droits d'accès attribués aux fichiers et répertoires dans un système de fichiers UNIX, déterminant qui peut lire, écrire ou exécuter ces fichiers.

**Playbook** : Fichier YAML utilisé par Ansible pour décrire les tâches à exécuter sur un ensemble de machines. Il contient les instructions nécessaires pour automatiser des opérations telles que l'installation de logiciels, la configuration de services, etc.

**Portainer** : Plateforme de gestion des conteneurs open-source, offrant une interface conviviale pour gérer, surveiller et déployer des conteneurs Docker dans des environnements de développement et de production.

**Raspberry Pi** : Série d'ordinateurs monocarte de petite taille développée au Royaume-Uni par la Fondation Raspberry Pi.

**Raspberry Pi Imager** : Utilitaire officiel de la Fondation Raspberry Pi permettant de créer des supports bootables pour les cartes SD utilisées avec les Raspberry Pi.

**Raspberry Pi OS** : Système d'exploitation basé sur Debian spécialement conçu pour les Raspberry Pi.

**Rsync** : Utilitaire de synchronisation de fichiers et de réPLICATION de données souvent utilisé pour effectuer des sauvegardes unidirectionnelles.

**Samba** : Suite logicielle qui implémente le protocole SMB/CIFS permettant le partage de fichiers et d'imprimantes entre systèmes d'exploitation différents.

**SMB/CIFS** : Protocoles de partage de fichiers réseau utilisés pour permettre le partage de fichiers et d'imprimantes entre ordinateurs.

**SSL** : Secure Socket Layer, protocole de sécurité utilisé pour établir une connexion sécurisée entre un serveur web et un navigateur web, assurant le chiffrement des données.

**SSH** : Secure Shell, protocole de communication sécurisé utilisé pour accéder à distance à des ordinateurs et exécuter des commandes.

**Sudo** : Commande utilisée dans les systèmes d'exploitation de type UNIX pour exécuter des commandes avec les priviléges de sécurité d'un autre utilisateur, généralement l'utilisateur root.

**Systemctl** : Commande utilisée pour contrôler les services système dans les systèmes d'exploitation basés sur Linux utilisant systemd comme gestionnaire de services.

**Time Machine** : Logiciel de sauvegarde intégrée à macOS permettant de sauvegarder automatiquement les données sur un disque externe local ou en réseau.

**Watchtower** : Outil open-source qui surveille les conteneurs Docker en cours d'exécution et met automatiquement à jour les images des conteneurs lorsqu'une nouvelle version est disponible.

**Webmin** : Interface web basée sur l'interface utilisateur graphique traditionnelle Unix pour la gestion des systèmes Unix, y compris la configuration du système, la gestion des utilisateurs, la gestion des paquets, etc.

**Yed Graph Editor** : Logiciel de création de graphiques et de diagrammes utilisé pour visualiser l'architecture des projets et les interactions système.

## 5.5 Cahier des charges

### 1 INFORMATIONS GENERALES

Candidat	Nom : <b>RONZANO</b>	Prénom : Rocco
	 rocco.ronzano@epfl.ch	 +41216934249
Lieu de travail :	EPFL ENAC-IT, BP 3 220, Station 16, 1015 Lausanne	
Orientation :	<input type="checkbox"/> 88601 Développement d'applications <input checked="" type="checkbox"/> 88602 Informatique d'entreprise <input type="checkbox"/> 88603 Technique des systèmes	
Chef de projet	Nom : <b>CHEVALLEY</b>	Prénom : Nicolas
	 nicolas.chevalley@epfl.ch	 +41216936278
Expert 1	Nom : <b>WENGER</b>	Prénom : Serge
	 serge.wenger@matisa.ch Bd de l'Arc-en-Ciel 25 1023 Crissier	 +41216312213

<b>Expert 2</b>	Nom : <b>PANTIC</b>	Prénom : Nemanja
	 nemo.pantic@gmail.com	 +41782304767
<b>Période de réalisation :</b>	Du lundi 29 janvier au lundi 12 février 2024	
<b>Horaire de travail :</b>	Lundi, mardi, mercredi, jeudi, vendredi : 9h-12h / 13h-18h30 Pauses : 20 minutes le matin / 15 minutes l'après-midi	
<b>Nombre d'heures :</b>	87 heures (11 x 7h55)	
<b>Planning (en H ou %)</b>	Analyse : 15%	
	Implémentation : 50%	
	Tests : 5%	
	Documentations : 30%	

## 2 PROCÉDURE

- Le candidat réalise un travail personnel sur la base d'un cahier des charges reçu le 1er jour.
- Le cahier des charges est approuvé par les deux experts. Il est en outre présenté, commenté et discuté avec le candidat. Par sa signature, le candidat accepte le travail proposé.
- Le candidat a connaissance de la feuille d'évaluation avant de débuter le travail.
- Le candidat est entièrement responsable de la sécurité de ses données.
- En cas de problèmes graves, le candidat avertit au plus vite les deux experts et son CdP.
- Le candidat a la possibilité d'obtenir de l'aide, mais doit le mentionner dans son dossier.
- A la fin du délai imparti pour la réalisation du TPI, le candidat doit transmettre par courrier électronique le dossier de projet aux deux experts et au chef de projet. En parallèle, une copie papier du rapport doit être fournie sans délai en trois exemplaires (L'un des deux experts peut demander à ne recevoir que la version électronique du dossier). Cette dernière doit être en tout point identique à la version électronique.

## 3 TITRE

Raspberry Pi : Utilisation à domicile et services avancés.

## 4 MATÉRIEL ET LOGICIEL À DISPOSITION

- 3x Raspberry Pi 4
- 3x carte microSD
- 2x Disque dur USB
- 1x Clé USB
- 1x Imprimante USB
- Switch, câbles et adaptateurs divers
- Postes clients Windows et macOS
- Raspberry Pi OS
- Accès Internet illimité

## 5 PRÉREQUIS

- Bonnes connaissances en administration système Linux, CLI, réseau, sécurité
- Connaissances Ansible, Docker, Apache, SSL, Webmin, SMB, CUPS, Git

## 6 DESCRIPTIF DU PROJET

Le projet consiste à mettre en place un environnement Raspberry Pi complet et polyvalent, basé sur Raspberry Pi OS, intégrant une gamme de services populaires pour répondre à des besoins domestiques et professionnels.

L'objectif est de créer une infrastructure, capable de fournir divers services essentiels, avec une attention particulière à la configuration, à la sécurité et à la documentation de chaque service selon les meilleures pratiques.

Pour ce faire, le candidat dispose de 3 Raspberry Pi (nommés Pi1 à Pi3).

Le projet se décompose en 2 étapes :

- Point 6.1 : Cas concret d'utilisation, intitulé "Utilisation simplifiée d'un Raspberry Pi à domicile" (concerne l'utilisation du Pi1 uniquement).
- Point 6.2 : Services avancés et flexibilité.

### 1.1 Cas concret d'utilisation :

Plusieurs utilisateurs nous ont récemment sollicité en raison de la suppression de fonctionnalités essentielles de la part de leurs fournisseurs de services Internet à domicile.

Ces fonctionnalités incluent notamment la possibilité de partager des imprimantes et des disques USB via le réseau domestique.

Dans le cadre de ce projet, je demande au candidat d'élaborer une procédure d'installation simplifiée, extraite du "rapport de projet", et à l'attention d'utilisateurs ne possédant pas de compétences informatiques avancées.

Cette procédure, intitulée "Utilisation simplifiée d'un Raspberry Pi à domicile", aura pour objectif de guider les utilisateurs au travers des étapes suivantes :

- Installation de Raspberry Pi OS.
- Configuration du partage d'un disque dur USB (NAS).
- Mise en place d'une solution de sauvegarde des postes clients.
- Partage d'une imprimante USB.

En suivant cette procédure, les utilisateurs seront en mesure de configurer un Raspberry Pi (unique), d'accéder à leurs données, de les sauvegarder, et d'imprimer depuis n'importe quel poste client Windows ou macOS au sein de leur réseau domestique.

### 1.2 Services avancés et flexibilité :

Sur la base du travail effectué lors de l'étape 6.1, le candidat est chargé d'implémenter une stratégie de configuration et de gestion automatisée en utilisant tous les Raspberry Pi disponibles (Pi1 à Pi3).

Mise en place d'un service de conteneurisation (CaaS) exécutant :

- Un serveur Web sous Apache, hébergeant la procédure "Utilisation simplifiée d'un Raspberry Pi à domicile" précédemment créée.
- Webmin pour la gestion centralisée des services.

La sauvegarde du NAS doit être effectuée de manière unidirectionnelle.

Par soucis de complexité et de temps à disposition, la mise en place d'une reprise de Failover n'est pas demandée.

Cependant, il est attendu que toutes tâches susceptibles d'être automatisées au maximum avec Ansible le soient, permettant ainsi une diminution de la charge opérationnelle et une minimalisation des risques d'erreurs humaines.

Une répartition cohérente des services est attendue.

Cette approche vise à garantir une expérience utilisateur fluide, même en cas de défaillance matérielle ou système, grâce à une répartition stratégique des services et une gestion simplifiée de la configuration.

### 1.3 Contraintes techniques :

1. Le projet doit être réalisé sur la distribution Raspberry Pi OS.
2. La configuration de type "Cluster" n'est pas autorisée.
3. Le NAS doit être accessible via le protocole SMB/CIFS.
4. Utilisation de Docker pour la partie CaaS.
5. La procédure "Utilisation simplifiée d'un Raspberry Pi à domicile" du point 6.1 doit être accessible en tant que site Web. Il doit être responsive (Google Chrome, Firefox et Safari), et utiliser un certificat SSL (Let's Encrypt).
6. Utilisation de Webmin pour la gestion centralisée des services.
7. Utilisation de Rsync pour la sauvegarde du NAS.
8. Pour le point 6.2, l'ensemble des Raspberry Pi (Pi1 à Pi3) doit être utilisé avec une répartition cohérente des services. Automatisation des tâches au maximum avec Ansible.
9. L'intégralité des scripts et fichiers de configuration doivent se trouver sur un système de contrôle de version (Git).

## 7 LIVRABLES

Le candidat est responsable de livrer à son chef de projet et aux deux experts, par mail :

- Une planification initiale (livrée à la fin du premier jour)
- Un rapport de projet (livré 3 fois par semaine, lundi / mercredi / vendredi)
- Un journal de travail (livré chaque jour)
- L'ensemble des scripts et fichiers de configuration sur un repo Git (livré le dernier jour)
- La procédure "Utilisation simplifiée d'un Raspberry Pi à domicile" (livré le dernier jour)

## 8 POINTS TECHNIQUES ÉVALUÉS SPÉCIFIQUES AU PROJET

La grille d'évaluation définit les critères généraux selon lesquels le travail du candidat sera évalué (documentation, journal de travail, respect des normes, qualité, ...).

En plus de cela, le travail sera évalué sur les 7 points spécifiques suivants (Point A14 à A20) :

1. La procédure "Utilisation simplifiée d'un Raspberry Pi à domicile" est adaptée aux utilisateurs.
2. Responsivité et certificat SSL du site Web.
3. Les services Apache et Webmin fonctionnent dans des conteneurs.
4. La configuration de la sauvegarde fonctionne et est adaptée au cas d'utilisation.
5. Les installations et configurations ont été automatisées.
6. Répartition des services et tolérance aux pannes (test de l'environnement en fonction).
7. Évaluation du repo Git (contenu / fréquence : pull, commit, push / descriptifs des commits).

## 9 VALIDATION

	Lu et approuvé le :	Signature :
Candidat :		
Expert n°1 :		
Expert n° 2 :		
Chef de projet :		

### 5.6 Code source

Repo Github : <https://github.com/roccom14/HOMEpi>

README.md

```
# HOMEpi
```

```
Repository de mon second TPI à l'EPFL
```

```
Il contiendra tout les scripts nécessaire au bon fonctionnement du projet ainsi que la documentation pour installer et configurer les Raspberry Pi.
```

### ansible.cfg

```
# ansible.cfg

[defaults]
inventory = ./inventory.yml

[ssh_connection]
ssh_args = -o ControlPersist=60s -o StrictHostKeyChecking=no
```

### inventory.yml

```
# inventory.yml

raspberries:
  hosts:
    pi1:
      ansible_host: pi1.local
      ansible_user: pi1
    pi2:
      ansible_host: pi2.local
      ansible_user: pi2
    pi3:
      ansible_host: pi3.local
      ansible_user: pi3
```

### master\_playbook\_install\_pi.yml

```
---
- name: Master Playbook – Installation des Raspberry Pi
  hosts: raspberries
  become: yes
  tasks:
    - name: Installation Pi1
      include_playbook: playbook_install_pi1.yml

    - name: Installation Pi2
      include_playbook: playbook_install_pi2.yml

    - name: Installation Pi3
      include_playbook: playbook_install_pi3.yml
```

### playbook\_cups.yml

```
---
- name: Playbook – Mise en place de CUPS
  hosts: all
  become: yes
  tasks:
    - name: Mise à jour des paquets
      apt:
        update_cache: yes

    - name: Mise à niveau du système
      apt:
```

```

upgrade: full
force_apt_get: yes
autoremove: yes

- name: Installation de CUPS
  apt:
    name: cups
    state: present

- name: Ajout de l'utilisateur actuel au groupe lpadmin
  user:
    name: "{{ ansible_user }}"
    groups: lpadmin
    append: yes

- name: Importaton du fichier cupsd.conf
  copy:
    src: ../conf/cupsd.conf
    dest: /etc/cups/cupsd.conf
    owner: root
    group: lp
    mode: '0644'
  notify: Redémarrage du service CUPS

handlers:
- name: Redémarrage du service CUPS
  service:
    name: cups
    state: restarted

```

## playbook\_docker.yml

```

---
- name: Playbook - Mise en place de Docker
  hosts: all
  become: yes
  tasks:
    - name: Mise à jour des paquets
      apt:
        update_cache: yes

    - name: Mise à niveau du système
      apt:
        upgrade: full
        force_apt_get: yes
        autoremove: yes

    - name: Ajout de la clé GPG officielle de Docker
      apt_key:
        url: https://download.docker.com/linux/debian/gpg
        state: present

```

```

- name: Installation des paquets requis
  apt:
    name: "{{ item }}"
    state: present
  loop:
    - ca-certificates
    - curl

- name: Création du répertoire /etc/apt/keyrings
  file:
    path: /etc/apt/keyrings
    state: directory
    mode: '0755'

- name: Téléchargement de la clé GPG de Docker
  get_url:
    url: https://download.docker.com/linux/debian/gpg
    dest: /etc/apt/keyrings/docker.asc

- name: Définition du repository de Docker
  blockinfile:
    path: /etc/apt/sources.list.d/docker.list
    block: |
      deb [arch={{ ansible_architecture }} signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/debian {{ ansible_distribution_release }} stable
    state: present

- name: Mise à jour du cache Apt
  apt:
    update_cache: yes

- name: Installation de Docker et des composants associés
  apt:
    name: "{{ item }}"
    state: present
  loop:
    - docker-ce
    - docker-ce-cli
    - containerd.io
    - docker-buildx-plugin
    - docker-compose-plugin

- name: Vérification de l'installation de Docker
  command: docker run hello-world
  register: docker_hello
  ignore_errors: yes

- name: Affichage du résultat de la vérification
  debug:
    msg: "Docker est correctement installé et fonctionnel."
  when: docker_hello is success

```

```

- name: Cloner le dépôt Git
  git:
    repo: https://github.com/roccom14/HOMEpi
    dest: "/home/{{ ansible_user }}/HOMEpi"
    version: main
    force: yes

- name: Création des clés SSL auto-signées
  shell: |
    COUNTRY="CH"
    STATE="Vaud"
    CITY="Lausanne"
    ORGANIZATION="EPFL"
    ORG_UNIT="ENAC-IT"
    COMMON_NAME="{{ ansible_user }}.local"
    EMAIL="rocco.ronzano@epfl.ch"

    sudo openssl req -x509 -nodes 365 -newkey rsa:4096 -keyout /home/{{ ansible_user }}/HOMEpi/docker/apache/server.key -out /home/{{ ansible_user }}/HOMEpi/docker/apache/server.crt -subj "/C=$COUNTRY/ST=$STATE/L=$CITY/O=$ORGANIZATION/OU=$ORG_UNIT/CN=$COMMON_NAME/emailAddress=$EMAIL"

- name: Lancement du docker compose
  command: "docker compose -f /home/{{ ansible_user }}/HOMEpi/docker/docker-compose.yml up -d"

```

### playbook\_install\_pi1.yml

```

---
- name: Playbook - Installation et configuration Raspberry Pi1
  hosts: pi1
  become: yes
  tasks:
    - name: Générer les locales en français suisse
      command: locale-gen fr_CH.UTF-8

    - name: Définition des paramètres régionaux
      command: localectl set-locale LANG=fr_CH.UTF-8

    - name: Définition des variables d'environnement
      lineinfile:
        dest: /etc/environment
        line: "{{ item }}"
      with_items:
        - "LANGUAGE=fr_CH.UTF-8"
        - "LC_ALL=fr_CH.UTF-8"
        - "LANG=fr_CH.UTF-8"
        - "LC_CTYPE=fr_CH.UTF-8"

    - name: Mise à jour des paquets

```

```

apt:
  update_cache: yes

- name: Mise à niveau du système
  apt:
    upgrade: full
    force_apt_get: yes
    autoremove: yes

- name: Installation des programmes spécifiques au Pi1
  apt:
    name: "{{ item }}"
    state: present
  with_items:
    - cups
    - samba
    - samba-common-bin
    - ansible

- name: Ajout de l'utilisateur actuel au groupe lpadmin
  user:
    name: "{{ ansible_user }}"
    groups: lpadmin
    append: yes

- name: Importaton du fichier cupsd.conf
  copy:
    src: ./conf/cupsd.conf
    dest: /etc/cups/cupsd.conf
    owner: root
    group: lp
    mode: '0644'
  notify: Redémarrage du service CUPS

handlers:
- name: Redémarrage du service CUPS
  service:
    name: cups
    state: restarted

- name: Montage du disque
  mount:
    src: /dev/sda1
    path: /mnt/sda1
    state: mounted

- name: Montage automatique du disque au démarrage
  lineinfile:
    dest: /etc/fstab
    line: "/dev/sda1 /mnt/sda1/ ext4 defaults,noatime 0 1"
    insertafter: EOF

```

```

state: present

- name: Création des dossiers
  file:
    path: "{{ item }}"
    state: directory
  loop:
    - /mnt/sda1/backups
    - /mnt/sda1/shared
    - /mnt/sda1/shared/nas1-tpi
    - /mnt/sda1/shared/nas2-tpi
    - /mnt/sda1/TimeMachine

- name: Ajouter des partages Samba au fichier smb.conf
  blockinfile:
    path: /etc/samba/smb.conf
    block: |
      [shared]
      path = /mnt/sda1/shared
      writeable = yes
      create mask = 0770
      directory mask = 0770
      public = no

      [backups]
      path = /mnt/sda1/backups
      writeable = yes
      create mask = 0770
      directory mask = 0770
      public = no

      [TimeMachine]
      comment = Time Machine
      path = /mnt/sda1/TimeMachine
      browseable = yes
      writeable = yes
      guest ok = no
      create mask = 0770
      directory mask = 0770
      vfs objects = catia fruit streams_xattr
      fruit:aapl = yes
      fruit:time machine = yes
      state: present

- name: Réglage des permissions sur le disque
  file:
    path: /mnt/sda1
    mode: "0770"
    state: directory
    recursive: yes

```

```

- name: Ajout d'une tâche cron pour la mise à jour automatique du système
  cron:
    minute: "30"
    hour: "3"
    job: "ansible-playbook -i inventory.yml playbook_update.yml"
    user: "root"
    state: present
    when: inventory_hostname == ansible_hostname

- name: Redémarrage du système pour appliquer les paramètres
  command: reboot now

```

### playbook\_install\_pi2.yml

```

---
- name: Playbook – Installation et configuration Raspberry Pi2
  hosts: pi2
  become: yes
  tasks:
    - name: Générer les locales en français suisse
      command: locale-gen fr_CH.UTF-8

    - name: Définition des paramètres régionaux
      command: localectl set-locale LANG=fr_CH.UTF-8

    - name: Définition des variables d'environnement
      lineinfile:
        dest: /etc/environment
        line: "{{ item }}"
      with_items:
        - "LANGUAGE=fr_CH.UTF-8"
        - "LC_ALL=fr_CH.UTF-8"
        - "LANG=fr_CH.UTF-8"
        - "LC_CTYPE=fr_CH.UTF-8"

    - name: Mise à jour des paquets
      apt:
        update_cache: yes

    - name: Mise à niveau du système
      apt:
        upgrade: full
        force_apt_get: yes
        autoremove: yes

    - name: Installation des programmes spécifiques au Pi2
      apt:
        name: "{{ item }}"
        state: present
      with_items:
        - rsync
        - ansible

```

```

- name: Montage automatique du disque au démarrage
  lineinfile:
    dest: /etc/fstab
    line: "/dev/sda1 /mnt/sda1/ exfat defaults,noatime 0 1"
    insertafter: EOF
    state: present

- name: Ajout d'une tâche cron rsync
  cron:
    minute: "0"
    hour: "*"
    user: "root"
    job: "sudo rsync -rvzu homepi@pi1.local:/mnt/sda1/ /mnt/sda1/"

- name: Ajout d'une tâche cron pour la mise à jour automatique du système
  cron:
    minute: "30"
    hour: "3"
    job: "ansible-playbook -i inventory.yml playbook_update.yml"
    user: "root"
    state: present
  when: inventory_hostname == ansible_hostname

- name: Redémarrage du système pour appliquer les paramètres
  command: reboot now

```

### playbook\_install\_pi3.yml

```

---
- name: Playbook - Installation et configuration Raspberry Pi3
  hosts: pi3
  become: yes
  tasks:
    - name: Générer les locales en français suisse
      command: locale-gen fr_CH.UTF-8

    - name: Définition des paramètres régionaux
      command: localectl set-locale LANG=fr_CH.UTF-8

    - name: Définition des variables d'environnement
      lineinfile:
        dest: /etc/environment
        line: "{{ item }}"
      with_items:
        - "LANGUAGE=fr_CH.UTF-8"
        - "LC_ALL=fr_CH.UTF-8"
        - "LANG=fr_CH.UTF-8"
        - "LC_CTYPE=fr_CH.UTF-8"

    - name: Mise à jour des paquets
      apt:

```

```

update_cache: yes

- name: Mise à niveau du système
  apt:
    upgrade: full
    force_apt_get: yes
    autoremove: yes

- name: Ajout de la clé GPG officielle de Docker
  apt_key:
    url: https://download.docker.com/linux/debian/gpg
    state: present

- name: Installation des paquets requis
  apt:
    name: "{{ item }}"
    state: present
  loop:
    - ca-certificates
    - curl

- name: Création du répertoire /etc/apt/keyrings
  file:
    path: /etc/apt/keyrings
    state: directory
    mode: '0755'

- name: Téléchargement de la clé GPG de Docker
  get_url:
    url: https://download.docker.com/linux/debian/gpg
    dest: /etc/apt/keyrings/docker.asc

- name: Définition du repository de Docker
  blockinfile:
    path: /etc/apt/sources.list.d/docker.list
    block: |
      deb [arch={{ ansible_architecture }} signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/debian {{ ansible_distribution_release }} stable
    state: present

- name: Mise à jour du cache Apt
  apt:
    update_cache: yes

- name: Installation de Docker et des composants associés
  apt:
    name: "{{ item }}"
    state: present
  loop:
    - docker-ce
    - docker-ce-cli

```

```

      - containerd.io
      - docker-buildx-plugin
      - docker-compose-plugin

- name: Vérification de l'installation de Docker
  command: docker run hello-world
  register: docker_hello
  ignore_errors: yes

- name: Affichage du résultat de la vérification
  debug:
    msg: "Docker est correctement installé et fonctionnel."
  when: docker_hello is success

- name: Cloner le dépôt Git
  git:
    repo: https://github.com/roccom14/HOMEpi
    dest: "/home/{{ ansible_user }}/HOMEpi"
    version: main
    force: yes

- name: Création des clés SSL auto-signées
  shell: |
    COUNTRY="CH"
    STATE="Vaud"
    CITY="Lausanne"
    ORGANIZATION="EPFL"
    ORG_UNIT="ENAC-IT"
    COMMON_NAME="{{ ansible_user }}.local"
    EMAIL="rocco.ronzano@epfl.ch"

    sudo openssl req -x509 -nodes -days 365 -newkey rsa:4096 -keyout /home/{{ ansible_user }}/HOMEpi/docker/apache/server.key -out /home/{{ ansible_user }}/HOMEpi/docker/apache/server.crt -subj "/C=$COUNTRY/ST=$STATE/L=$CITY/O=$ORGANIZATION/OU=$ORG_UNIT/CN=$COMMON_NAME/emailAddress=$EMAIL"

- name: Lancement du docker compose
  command: "docker compose -f /home/{{ ansible_user }}/HOMEpi/docker/docker-compose.yml up -d"

- name: Ajout d'une tâche cron pour la mise à jour automatique du système
  cron:
    minute: "30"
    hour: "3"
    job: "ansible-playbook -i /home/{{ ansible_user }}/HOMEpi/ansible/inventory.yml /home/{{ ansible_user }}/HOMEpi/playbook_update.yml -l {{ ansible_user }}"
    user: "root"
    state: present
    when: inventory_hostname == ansible_hostname

```

```
- name: Redémarrage du système pour appliquer les paramètres
  command: reboot now
```

## playbook\_nas.yml

```
---
- name: Playbook - Mise en place du NAS
  hosts: all
  become: yes
  tasks:
    - name: Mise à jour des paquets
      apt:
        update_cache: yes

    - name: Mise à niveau du système
      apt:
        upgrade: full
        force_apt_get: yes
        autoremove: yes

    - name: Installation des programmes spécifiques au NAS
      apt:
        name: "{{ item }}"
        state: present
      with_items:
        - samba
        - samba-common-bin

    - name: Montage du disque
      mount:
        src: /dev/sda1
        path: /mnt/sda1
        state: mounted

    - name: Montage automatique du disque au démarrage
      lineinfile:
        dest: /etc/fstab
        line: "/dev/sda1 /mnt/sda1 ext4 defaults,noatime 0 1"
        insertafter: EOF
        state: present

    - name: Création des dossiers
      file:
        path: "{{ item }}"
        state: directory
      loop:
        - /mnt/sda1/backups
        - /mnt/sda1/shared
        - /mnt/sda1/shared/nas1-tpi
        - /mnt/sda1/shared/nas2-tpi
        - /mnt/sda1/TimeMachine
```

```

- name: Ajouter des partages Samba au fichier smb.conf
  blockinfile:
    path: /etc/samba/smb.conf
    block: |
      [shared]
      path = /mnt/sda1/shared
      writeable = yes
      create mask = 0770
      directory mask = 0770
      public = no

      [backups]
      path = /mnt/sda1/backups
      writeable = yes
      create mask = 0770
      directory mask = 0770
      public = no

      [TimeMachine]
      comment = Time Machine
      path = /mnt/sda1/TimeMachine
      browsable = yes
      writeable = yes
      guest ok = no
      create mask = 0770
      directory mask = 0770
      vfs objects = catia fruit streams_xattr
      fruit:aapl = yes
      fruit:time machine = yes
      state: present

- name: Réglage des permissions sur le disque
  file:
    path: /mnt/sda1
    mode: "0770"
    state: directory
    recursive: yes

handlers:
- name: Redémarrage du service Samba
  service:
    name: smbd
    state: restarted

```

## playbook\_rsync.yml

```

---
- name: Playbook - Mise en place de Rsync
  hosts: all
  become: yes
  tasks:
    - name: Mise à jour des paquets

```

```

apt:
  update_cache: yes

- name: Mise à niveau du système
  apt:
    upgrade: full
    force_apt_get: yes
    autoremove: yes

- name: Installation de Rsync
  apt:
    name: rsync
    state: present

- name: Montage automatique du disque au démarrage
  lineinfile:
    dest: /etc/fstab
    line: "/dev/sda1 /mnt/sda1/ exfat defaults,noatime 0 1"
    insertafter: EOF
    state: present

- name: Ajout d'une tâche cron Rsync
  cron:
    minute: "0"
    hour: "*"
    user: "root"
    job: "sudo rsync -rvzu homepi@pi1.local:/mnt/sda1/ /mnt/sda1/"

```

### playbook\_update.yml

```

---
- name: Playbook – Mise à jour des Raspberry Pi
  hosts: raspberries
  become: yes
  tasks:
    - name: Mise à jour des paquets
      apt:
        update_cache: yes

    - name: Mise à niveau du système
      apt:
        upgrade: full
        force_apt_get: yes
        autoremove: yes

```

### cupsd.conf

```

LogLevel warn
PageLogFormat
MaxLogSize 0
ErrorPolicy retry-job
Listen *:631
Port 631

```

```

Listen /run/cups/cups.sock
Browsing On
BrowseLocalProtocols dnssd
DefaultAuthType Basic
WebInterface Yes
IdleExitTimeout 60
<Location />
  Order allow,deny
  Allow all
</Location>
<Location /admin>
  Order allow,deny
  Allow all
</Location>
<Location /admin/conf>
  AuthType Default
  Require user @SYSTEM
</Location>
<Location /admin/log>
  AuthType Default
  Require user @SYSTEM
  Order allow,deny
  Allow all
</Location>
<Policy default>
  JobPrivateAccess default
  JobPrivateValues default
  SubscriptionPrivateAccess default
  SubscriptionPrivateValues default
  <Limit Create-Job Print-Job Print-URI Validate-Job>
    Order deny,allow
  </Limit>
  <Limit Send-Document Send-URI Hold-Job Release-Job Restart-Job Purge-Jobs Set-Job-
Attributes Create-Job-Subscription Renew-Subscription Cancel-Subscription Get-
Notifications Reprocess-Job Cancel-Current-Job Suspend-Current-Job Resume-Job Cancel-My-
Jobs Close-Job CUPS-Move-Job>
    Require user @OWNER @SYSTEM
    Order deny,allow
  </Limit>
  <Limit CUPS-Get-Document>
    AuthType Default
    Require user @OWNER @SYSTEM
    Order deny,allow
  </Limit>
  <Limit CUPS-Add-Modify-Printer CUPS-Delete-Printer CUPS-Add-Modify-Class CUPS-Delete-
Class CUPS-Set-Default CUPS-Get-Devices>
    AuthType Default
    Require user @SYSTEM
    Order deny,allow
  </Limit>

```

```

<Limit Pause-Printer Resume-Printer Enable-Printer Disable-Printer Pause-Printer-After-
Current-Job Hold-New-Jobs Release-Held-New-Jobs Deactivate-Printer Activate-Printer
Restart-Printer Shutdown-Printer Startup-Printer Promote-Job Schedule-Job-After Cancel-
Jobs CUPS-Accept-Jobs CUPS-Reject-Jobs>
  AuthType Default
  Require user @SYSTEM
  Order deny,allow
</Limit>
<Limit Cancel-Job CUPS-Authenticate-Job>
  Require user @OWNER @SYSTEM
  Order deny,allow
</Limit>
<Limit All>
  Order deny,allow
</Limit>
</Policy>
<Policy authenticated>
  JobPrivateAccess default
  JobPrivateValues default
  SubscriptionPrivateAccess default
  SubscriptionPrivateValues default
  <Limit Create-Job Print-Job Print-URI Validate-Job>
    AuthType Default
    Order deny,allow
  </Limit>
  <Limit Send-Document Send-URI Hold-Job Release-Job Restart-Job Purge-Jobs Set-Job-
Attributes Create-Job-Subscription Renew-Subscription Cancel-Subscription Get-
Notifications Reprocess-Job Cancel-Current-Job Suspend-Current-Job Resume-Job Cancel-My-
Jobs Close-Job CUPS-Move-Job CUPS-Get-Document>
    AuthType Default
    Require user @OWNER @SYSTEM
    Order deny,allow
  </Limit>
  <Limit CUPS-Add-Modify-Printer CUPS-Delete-Printer CUPS-Add-Modify-Class CUPS-Delete-
Class CUPS-Set-Default>
    AuthType Default
    Require user @SYSTEM
    Order deny,allow
  </Limit>
  <Limit Pause-Printer Resume-Printer Enable-Printer Disable-Printer Pause-Printer-After-
Current-Job Hold-New-Jobs Release-Held-New-Jobs Deactivate-Printer Activate-Printer
Restart-Printer Shutdown-Printer Startup-Printer Promote-Job Schedule-Job-After Cancel-
Jobs CUPS-Accept-Jobs CUPS-Reject-Jobs>
    AuthType Default
    Require user @SYSTEM
    Order deny,allow
  </Limit>
  <Limit Cancel-Job CUPS-Authenticate-Job>
    AuthType Default
    Require user @OWNER @SYSTEM
    Order deny,allow
  </Limit>

```

```

</Limit>
<Limit All>
  Order deny,allow
</Limit>
</Policy>
<Policy kerberos>
  JobPrivateAccess default
  JobPrivateValues default
  SubscriptionPrivateAccess default
  SubscriptionPrivateValues default
  <Limit Create-Job Print-Job Print-URI Validate-Job>
    AuthType Negotiate
    Order deny,allow
  </Limit>
  <Limit Send-Document Send-URI Hold-Job Release-Job Restart-Job Purge-Jobs Set-Job-
Attributes Create-Job-Subscription Renew-Subscription Cancel-Subscription Get-
Notifications Reprocess-Job Cancel-Current-Job Suspend-Current-Job Resume-Job Cancel-My-
Jobs Close-Job CUPS-Move-Job CUPS-Get-Document>
    AuthType Negotiate
    Require user @OWNER @SYSTEM
    Order deny,allow
  </Limit>
  <Limit CUPS-Add-Modify-Printer CUPS-Delete-Printer CUPS-Add-Modify-Class CUPS-Delete-
Class CUPS-Set-Default>
    AuthType Default
    Require user @SYSTEM
    Order deny,allow
  </Limit>
  <Limit Pause-Printer Resume-Printer Enable-Printer Disable-Printer Pause-Printer-After-
Current-Job Hold-New-Jobs Release-Held-New-Jobs Deactivate-Printer Activate-Printer
Restart-Printer Shutdown-Printer Startup-Printer Promote-Job Schedule-Job-After Cancel-
Jobs CUPS-Accept-Jobs CUPS-Reject-Jobs>
    AuthType Default
    Require user @SYSTEM
    Order deny,allow
  </Limit>
  <Limit Cancel-Job CUPS-Authenticate-Job>
    AuthType Negotiate
    Require user @OWNER @SYSTEM
    Order deny,allow
  </Limit>
  <Limit All>
    Order deny,allow
  </Limit>
</Policy>

```

httpd-ssl.conf

```

#
# This is the Apache server configuration file providing SSL support.
# It contains the configuration directives to instruct the server how to
# serve pages over an https connection. For detailed information about these

```

```

# directives see <URL:http://httpd.apache.org/docs/2.4/mod/mod\_ssl.html>
#
# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.
#
# Required modules: mod_log_config, mod_setenvif, mod_ssl,
#                   socache_shmcb_module (for default value of SSLSessionCache)

#
# Pseudo Random Number Generator (PRNG):
# Configure one or more sources to seed the PRNG of the SSL library.
# The seed data should be of good random quality.
# WARNING! On some platforms /dev/random blocks if not enough entropy
# is available. This means you then cannot use the /dev/random device
# because it would lead to very long connection times (as long as
# it requires to make more entropy available). But usually those
# platforms additionally provide a /dev/urandom device which doesn't
# block. So, if available, use this one instead. Read the mod_ssl User
# Manual for more details.
#
#SSLRandomSeed startup file:/dev/random 512
#SSLRandomSeed startup file:/dev/urandom 512
#SSLRandomSeed connect file:/dev/random 512
#SSLRandomSeed connect file:/dev/urandom 512

#
# When we also provide SSL we have to listen to the
# standard HTTP port (see above) and to the HTTPS port
#
Listen 443

##
## SSL Global Context
##
## All SSL configuration in this context applies both to
## the main server and all SSL-enabled virtual hosts.
##


# SSL Cipher Suite:
# List the ciphers that the client is permitted to negotiate,
# and that httpd will negotiate as the client of a proxied server.
# See the OpenSSL documentation for a complete list of ciphers, and
# ensure these follow appropriate best practices for this deployment.
# httpd 2.2.30, 2.4.13 and later force-disable aNULL, eNULL and EXP ciphers,
# while OpenSSL disabled these by default in 0.9.8zf/1.0.0r/1.0.1m/1.0.2a.
SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4:!3DES
SSLProxyCipherSuite HIGH:MEDIUM:!MD5:!RC4:!3DES

# By the end of 2016, only TLSv1.2 ciphers should remain in use.

```

```

# Older ciphers should be disallowed as soon as possible, while the
# kRSA ciphers do not offer forward secrecy. These changes inhibit
# older clients (such as IE6 SP2 or IE8 on Windows XP, or other legacy
# non-browser tooling) from successfully connecting.
#
# To restrict mod_ssl to use only TLSv1.2 ciphers, and disable
# those protocols which do not support forward secrecy, replace
# the SSLCipherSuite and SSLProxyCipherSuite directives above with
# the following two directives, as soon as practical.
# SSLCipherSuite HIGH:MEDIUM:!SSLv3:!kRSA
# SSLProxyCipherSuite HIGH:MEDIUM:!SSLv3:!kRSA

# User agents such as web browsers are not configured for the user's
# own preference of either security or performance, therefore this
# must be the prerogative of the web server administrator who manages
# cpu load versus confidentiality, so enforce the server's cipher order.
SSLHonorCipherOrder on

# SSL Protocol support:
# List the protocol versions which clients are allowed to connect with.
# Disable SSLv3 by default (cf. RFC 7525 3.1.1). TLSv1 (1.0) should be
# disabled as quickly as practical. By the end of 2016, only the TLSv1.2
# protocol or later should remain in use.
SSLProtocol all -SSLv3
SSLProxyProtocol all -SSLv3

# Pass Phrase Dialog:
# Configure the pass phrase gathering process.
# The filtering dialog program (`builtin' is an internal
# terminal dialog) has to provide the pass phrase on stdout.
SSLPassPhraseDialog builtin

# Inter-Process Session Cache:
# Configure the SSL Session Cache: First the mechanism
# to use and second the expiring timeout (in seconds).
#SSLSessionCache      "dbm:/usr/local/apache2/logs/ssl_scache"
SSLSessionCache      "shmcb:/usr/local/apache2/logs/ssl_scache(512000)"
SSLSessionCacheTimeout 300

# OCSP Stapling (requires OpenSSL 0.9.8h or later)
#
# This feature is disabled by default and requires at least
# the two directives SSLUseStapling and SSLStaplingCache.
# Refer to the documentation on OCSP Stapling in the SSL/TLS
# How-To for more information.
#
# Enable stapling for all SSL-enabled servers:
#SSLUseStapling On

# Define a relatively small cache for OCSP Stapling using
# the same mechanism that is used for the SSL session cache

```

```

# above. If stapling is used with more than a few certificates,
# the size may need to be increased. (AH01929 will be logged.)
#SSLStaplingCache "shmcb:/usr/local/apache2/logs/ssl_stapling(32768)"

# Seconds before valid OCSP responses are expired from the cache
#SSLStaplingStandardCacheTimeout 3600

# Seconds before invalid OCSP responses are expired from the cache
#SSLStaplingErrorCacheTimeout 600

## 
## SSL Virtual Host Context
## 

<VirtualHost _default_:443>

# General setup for the virtual host
DocumentRoot "/usr/local/apache2/htdocs"
ServerName pi3.local:443
ServerAdmin rocco.ronzano@epfl.ch
ErrorLog /proc/self/fd/2
TransferLog /proc/self/fd/1

# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

# Server Certificate:
# Point SSLCertificateFile at a PEM encoded certificate. If
# the certificate is encrypted, then you will be prompted for a
# pass phrase. Note that a kill -HUP will prompt again. Keep
# in mind that if you have both an RSA and a DSA certificate you
# can configure both in parallel (to also allow the use of DSA
# ciphers, etc.)
# Some ECC cipher suites (http://www.ietf.org/rfc/rfc4492.txt)
# require an ECC certificate which can also be configured in
# parallel.
SSLCertificateFile "/usr/local/apache2/conf/server.crt"
#SSLCertificateFile "/usr/local/apache2/conf/server-dsa.crt"
#SSLCertificateFile "/usr/local/apache2/conf/server-ecc.crt"

# Server Private Key:
# If the key is not combined with the certificate, use this
# directive to point at the key file. Keep in mind that if
# you've both a RSA and a DSA private key you can configure
# both in parallel (to also allow the use of DSA ciphers, etc.)
# ECC keys, when in use, can also be configured in parallel
SSLCertificateKeyFile "/usr/local/apache2/conf/server.key"
#SSLCertificateKeyFile "/usr/local/apache2/conf/server-dsa.key"
#SSLCertificateKeyFile "/usr/local/apache2/conf/server-ecc.key"

```

```

# Server Certificate Chain:
# Point SSLCertificateChainFile at a file containing the
# concatenation of PEM encoded CA certificates which form the
# certificate chain for the server certificate. Alternatively
# the referenced file can be the same as SSLCertificateFile
# when the CA certificates are directly appended to the server
# certificate for convenience.
#SSLCertificateChainFile "/usr/local/apache2/conf/server-ca.crt"

# Certificate Authority (CA):
# Set the CA certificate verification path where to find CA
# certificates for client authentication or alternatively one
# huge file containing all of them (file must be PEM encoded)
# Note: Inside SSLCACertificatePath you need hash symlinks
#       to point to the certificate files. Use the provided
#       Makefile to update the hash symlinks after changes.
#SSLCACertificatePath "/usr/local/apache2/conf/ssl.crt"
#SSLCACertificateFile "/usr/local/apache2/conf/ssl.crt/ca-bundle.crt"

# Certificate Revocation Lists (CRL):
# Set the CA revocation path where to find CA CRLs for client
# authentication or alternatively one huge file containing all
# of them (file must be PEM encoded).
# The CRL checking mode needs to be configured explicitly
# through SSLCARevocationCheck (defaults to "none" otherwise).
# Note: Inside SSLCARevocationPath you need hash symlinks
#       to point to the certificate files. Use the provided
#       Makefile to update the hash symlinks after changes.
#SSLCARevocationPath "/usr/local/apache2/conf/ssl.crl"
#SSLCARevocationFile "/usr/local/apache2/conf/ssl.crl/ca-bundle.crl"
#SSLCARevocationCheck chain

# Client Authentication (Type):
# Client certificate verification type and depth. Types are
# none, optional, require and optional_no_ca. Depth is a
# number which specifies how deeply to verify the certificate
# issuer chain before deciding the certificate is not valid.
#SSLVerifyClient require
#SSLVerifyDepth 10

# TLS-SRP mutual authentication:
# Enable TLS-SRP and set the path to the OpenSSL SRP verifier
# file (containing login information for SRP user accounts).
# Requires OpenSSL 1.0.1 or newer. See the mod_ssl FAQ for
# detailed instructions on creating this file. Example:
# "openssl srp -srpvfile /usr/local/apache2/conf/passwd.srpv -add username"
#SSLSRPVerifierFile "/usr/local/apache2/conf/passwd.srpv"

# Access Control:
# With SSLRequire you can do per-directory access control based
# on arbitrary complex boolean expressions containing server

```

```

#   variable checks and other lookup directives.  The syntax is a
#   mixture between C and Perl.  See the mod_ssl documentation
#   for more details.

#<Location />
#SSLRequire (    %{SSL_CIPHER} !~ m/^(EXP|NULL)/ \
#            and %{SSL_CLIENT_S_DN_0} eq "Snake Oil, Ltd." \
#            and %{SSL_CLIENT_S_DN_OU} in {"Staff", "CA", "Dev"} \
#            and %{TIME_WDAY} >= 1 and %{TIME_WDAY} <= 5 \
#            and %{TIME_HOUR} >= 8 and %{TIME_HOUR} <= 20      ) \
#            or %{REMOTE_ADDR} =~ m/^192\.76\.162\.[0-9]+$/
```

#</Location>

# SSL Engine Options:

# Set various options for the SSL engine.

- # o **FakeBasicAuth:**

# Translate the client X.509 into a Basic Authorisation. This means that

# the standard Auth/DBMAuth methods can be used for access control. The

# user name is the 'one line' version of the client's X.509 certificate.

# Note that no password is obtained from the user. Every entry in the user

# file needs this password: `xxj31ZMTZzkVA`.

- # o **ExportCertData:**

# This exports two additional environment variables: SSL\_CLIENT\_CERT and

# SSL\_SERVER\_CERT. These contain the PEM-encoded certificates of the

# server (always existing) and the client (only existing when client

# authentication is used). This can be used to import the certificates

# into CGI scripts.

- # o **StdEnvVars:**

# This exports the standard SSL/TLS related `SSL\_\*` environment variables.

# Per default this exportation is switched off for performance reasons,

# because the extraction step is an expensive operation and is usually

# useless for serving static content. So one usually enables the

# exportation for CGI and SSI requests only.

- # o **StrictRequire:**

# This denies access when "SSLRequireSSL" or "SSLRequire" applied even

# under a "Satisfy any" situation, i.e. when it applies access is denied

# and no other module can change it.

- # o **OptRenegotiate:**

# This enables optimized SSL connection renegotiation handling when SSL

# directives are used in per-directory context.

```
#SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
<FilesMatch "\.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
</FilesMatch>
<Directory "/usr/local/apache2/cgi-bin">
    SSLOptions +StdEnvVars
</Directory>

#   SSL Protocol Adjustments:
#   The safe and default but still SSL/TLS standard compliant shutdown
#   approach is that mod_ssl sends the close notify alert but doesn't wait for
#   the close notify alert from client. When you need a different shutdown
```

```

# approach you can use one of the following variables:
# o ssl-unclean-shutdown:
#   This forces an unclean shutdown when the connection is closed, i.e. no
#   SSL close notify alert is sent or allowed to be received. This violates
#   the SSL/TLS standard but is needed for some brain-dead browsers. Use
#   this when you receive I/O errors because of the standard approach where
#   mod_ssl sends the close notify alert.
# o ssl-accurate-shutdown:
#   This forces an accurate shutdown when the connection is closed, i.e. a
#   SSL close notify alert is send and mod_ssl waits for the close notify
#   alert of the client. This is 100% SSL/TLS standard compliant, but in
#   practice often causes hanging connections with brain-dead browsers. Use
#   this only for browsers where you know that their SSL implementation
#   works correctly.
# Notice: Most problems of broken clients are also related to the HTTP
# keep-alive facility, so you usually additionally want to disable
# keep-alive for those clients, too. Use variable "nokeepalive" for this.
# Similarly, one has to force some clients to use HTTP/1.0 to workaround
# their broken HTTP/1.1 implementation. Use variables "downgrade-1.0" and
# "force-response-1.0" for this.

BrowserMatch "MSIE [2-5]" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0

# Per-Server Logging:
# The home of a custom SSL log file. Use this when you want a
# compact non-error SSL logfile on a virtual host basis.
CustomLog /proc/self/fd/1 \
    "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

</VirtualHost>

```

httpd.conf

```

#
# This is the main Apache HTTP server configuration file. It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.4/> for detailed information.
# In particular, see
# <URL:http://httpd.apache.org/docs/2.4/mod/directives.html>
# for a discussion of each configuration directive.
#
# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path. If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so "logs/access_log"
# with ServerRoot set to "/usr/local/apache2" will be interpreted by the
# server as "/usr/local/apache2/logs/access_log", whereas "/logs/access_log"

```

```
# will be interpreted as '/logs/access_log'.
```

```
#
```

```
# ServerRoot: The top of the directory tree under which the server's
```

```
# configuration, error, and log files are kept.
```

```
#
```

```
# Do not add a slash at the end of the directory path. If you point
```

```
# ServerRoot at a non-local disk, be sure to specify a local disk on the
```

```
# Mutex directive, if file-based mutexes are used. If you wish to share the
```

```
# same ServerRoot for multiple httpd daemons, you will need to change at
```

```
# least PidFile.
```

```
#
```

```
ServerRoot "/usr/local/apache2"
```

```
#
```

```
# Mutex: Allows you to set the mutex mechanism and mutex file directory
```

```
# for individual mutexes, or change the global defaults
```

```
#
```

```
# Uncomment and change the directory if mutexes are file-based and the default
```

```
# mutex file directory is not on a local disk or is not appropriate for some
```

```
# other reason.
```

```
#
```

```
# Mutex default:logs
```

```
#
```

```
# Listen: Allows you to bind Apache to specific IP addresses and/or
```

```
# ports, instead of the default. See also the <VirtualHost>
```

```
# directive.
```

```
#
```

```
# Change this to Listen on specific IP addresses as shown below to
```

```
# prevent Apache from glomming onto all bound IP addresses.
```

```
#
```

```
#Listen 12.34.56.78:80
```

```
Listen 80
```

```
#
```

```
# Dynamic Shared Object (DSO) Support
```

```
#
```

```
# To be able to use the functionality of a module which was built as a DSO you
```

```
# have to place corresponding `LoadModule` lines at this location so the
```

```
# directives contained in it are actually available _before_ they are used.
```

```
# Statically compiled modules (those listed by `httpd -l`) do not need
```

```
# to be loaded here.
```

```
#
```

```
# Example:
```

```
# LoadModule foo_module modules/mod_foo.so
```

```
#
```

```
LoadModule mpm_event_module modules/mod_mpm_event.so
```

```
#LoadModule mpm_prefork_module modules/mod_mpm_prefork.so
```

```
#LoadModule mpm_worker_module modules/mod_mpm_worker.so
```

```
LoadModule authn_file_module modules/mod_authn_file.so
```

```
#LoadModule authn_dbm_module modules/mod_authn_dbm.so
#LoadModule authn_anon_module modules/mod_authn_anon.so
#LoadModule authn_dbd_module modules/mod_authn_dbd.so
#LoadModule authn_socache_module modules/mod_authn_socache.so
LoadModule authn_core_module modules/mod_authn_core.so
LoadModule authz_host_module modules/mod_authz_host.so
LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
LoadModule authz_user_module modules/mod_authz_user.so
#LoadModule authz_dbm_module modules/mod_authz_dbm.so
#LoadModule authz_owner_module modules/mod_authz_owner.so
#LoadModule authz_dbd_module modules/mod_authz_dbd.so
LoadModule authz_core_module modules/mod_authz_core.so
#LoadModule authnz_ldap_module modules/mod_authnz_ldap.so
#LoadModule authnz_fcg_module modules/mod_authnz_fcg.so
LoadModule access_compat_module modules/mod_access_compat.so
LoadModule auth_basic_module modules/mod_auth_basic.so
#LoadModule auth_form_module modules/mod_auth_form.so
#LoadModule auth_digest_module modules/mod_auth_digest.so
#LoadModule allowmethods_module modules/mod_allowmethods.so
#LoadModule isapi_module modules/mod_isapi.so
#LoadModule file_cache_module modules/mod_file_cache.so
#LoadModule cache_module modules/mod_cache.so
#LoadModule cache_disk_module modules/mod_cache_disk.so
#LoadModule cache_socache_module modules/mod_cache_socache.so
LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
#LoadModule socache_dbm_module modules/mod_socache_dbm.so
#LoadModule socache_memcache_module modules/mod_socache_memcache.so
#LoadModule socache_redis_module modules/mod_socache_redis.so
#LoadModule watchdog_module modules/mod_watchdog.so
#LoadModule macro_module modules/mod_macro.so
#LoadModule dbd_module modules/mod_dbd.so
#LoadModule bucketeer_module modules/mod_bucketeer.so
#LoadModule dumpio_module modules/mod_dumpio.so
#LoadModule echo_module modules/mod_echo.so
#LoadModule example_hooks_module modules/mod_example_hooks.so
#LoadModule case_filter_module modules/mod_case_filter.so
#LoadModule case_filter_in_module modules/mod_case_filter_in.so
#LoadModule example_ipc_module modules/mod_example_ipc.so
#LoadModule buffer_module modules/mod_buffer.so
#LoadModule data_module modules/mod_data.so
#LoadModule ratelimit_module modules/mod_ratelimit.so
LoadModule reqtimeout_module modules/mod_reqtimeout.so
#LoadModule ext_filter_module modules/mod_ext_filter.so
#LoadModule request_module modules/mod_request.so
#LoadModule include_module modules/mod_include.so
LoadModule filter_module modules/mod_filter.so
#LoadModule reflector_module modules/mod_reflector.so
#LoadModule substitute_module modules/mod_substitute.so
#LoadModule sed_module modules/mod_sed.so
#LoadModule charset_lite_module modules/mod_charset_lite.so
#LoadModule deflate_module modules/mod_deflate.so
```

```

#LoadModule xml2enc_module modules/mod_xml2enc.so
#LoadModule proxy_html_module modules/mod_proxy_html.so
#LoadModule brotli_module modules/mod_brotli.so
LoadModule mime_module modules/mod_mime.so
#LoadModule ldap_module modules/mod_ldap.so
LoadModule log_config_module modules/mod_log_config.so
#LoadModule log_debug_module modules/mod_log_debug.so
#LoadModule log_forensic_module modules/mod_log_forensic.so
#LoadModule logio_module modules/mod_logio.so
#LoadModule lua_module modules/mod_lua.so
LoadModule env_module modules/mod_env.so
#LoadModule mime_magic_module modules/mod_mime_magic.so
#LoadModule cern_meta_module modules/mod_cern_meta.so
#LoadModule expires_module modules/mod_expires.so
LoadModule headers_module modules/mod_headers.so
#LoadModule ident_module modules/mod_ident.so
#LoadModule usertrack_module modules/mod_usertrack.so
#LoadModule unique_id_module modules/mod_unique_id.so
LoadModule setenvif_module modules/mod_setenvif.so
LoadModule version_module modules/mod_version.so
#LoadModule remoteip_module modules/mod_remoteip.so
#LoadModule proxy_module modules/mod_proxy.so
#LoadModule proxy_connect_module modules/mod_proxy_connect.so
#LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
#LoadModule proxy_http_module modules/mod_proxy_http.so
#LoadModule proxy_fcgi_module modules/mod_proxy_fcgi.so
#LoadModule proxy_scgi_module modules/mod_proxy_scgi.so
#LoadModule proxy_uwsgi_module modules/mod_proxy_uwsgi.so
#LoadModule proxy_fdpass_module modules/mod_proxy_fdpass.so
#LoadModule proxy_wstunnel_module modules/mod_proxy_wstunnel.so
#LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
#LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
#LoadModule proxy_express_module modules/mod_proxy_express.so
#LoadModule proxy_hcheck_module modules/mod_proxy_hcheck.so
#LoadModule session_module modules/mod_session.so
#LoadModule session_cookie_module modules/mod_session_cookie.so
#LoadModule session_crypto_module modules/mod_session_crypto.so
#LoadModule session_dbd_module modules/mod_session_dbd.so
#LoadModule slotmem_shm_module modules/mod_slotmem_shm.so
#LoadModule slotmem_plain_module modules/mod_slotmem_plain.so
LoadModule ssl_module modules/mod_ssl.so
#LoadModule optional_hook_export_module modules/mod_optional_hook_export.so
#LoadModule optional_hook_import_module modules/mod_optional_hook_import.so
#LoadModule optional_fn_import_module modules/mod_optional_fn_import.so
#LoadModule optional_fn_export_module modules/mod_optional_fn_export.so
#LoadModule dialup_module modules/mod_dialup.so
#LoadModule http2_module modules/mod_http2.so
#LoadModule proxy_http2_module modules/mod_proxy_http2.so
#LoadModule md_module modules/mod_md.so
#LoadModule lbmethod_byrequests_module modules/mod_lbmethod_byrequests.so
#LoadModule lbmethod_bytraffic_module modules/mod_lbmethod_bytraffic.so

```

```

#LoadModule lbmethod_bybusyness_module modules/mod_lbmethod_bybusyness.so
#LoadModule lbmethod_heartbeat_module modules/mod_lbmethod_heartbeat.so
LoadModule unixd_module modules/mod_unixd.so
#LoadModule heartbeat_module modules/mod_heartbeat.so
#LoadModule heartmonitor_module modules/mod_heartmonitor.so
#LoadModule dav_module modules/mod_dav.so
LoadModule status_module modules/mod_status.so
LoadModule autoindex_module modules/mod_autoindex.so
#LoadModule asis_module modules/mod_asis.so
#LoadModule info_module modules/mod_info.so
#LoadModule suexec_module modules/mod_suexec.so
<IfModule !mpm_prefork_module>
    #LoadModule cgid_module modules/mod_cgid.so
</IfModule>
<IfModule mpm_prefork_module>
    #LoadModule cgi_module modules/mod_cgi.so
</IfModule>
#LoadModule dav_fs_module modules/mod_dav_fs.so
#LoadModule dav_lock_module modules/mod_dav_lock.so
#LoadModule vhost_alias_module modules/mod_vhost_alias.so
#LoadModule negotiation_module modules/mod_negotiation.so
LoadModule dir_module modules/mod_dir.so
#LoadModule imagemap_module modules/mod_imagemap.so
#LoadModule actions_module modules/mod_actions.so
#LoadModule speling_module modules/mod_speling.so
#LoadModule userdir_module modules/mod_userdir.so
LoadModule alias_module modules/mod_alias.so
LoadModule rewrite_module modules/mod_rewrite.so

<IfModule unixd_module>
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# It is usually good practice to create a dedicated user and group for
# running httpd, as with most system services.
#
User www-data
Group www-data

</IfModule>

# 'Main' server configuration
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition. These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <VirtualHost> containers,

```

```

# in which case these default settings will be overridden for the
# virtual host being defined.
#
#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed. This address appears on some server-generated pages, such
# as error documents. e.g. admin@your-domain.com
#
ServerAdmin rocco.ronzano@epfl.ch

#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName pi3.local:80

#
# Deny access to the entirety of your server's filesystem. You must
# explicitly permit access to web content directories in other
# <Directory> blocks below.
#
<Directory />
    AllowOverride none
    Require all denied
</Directory>

#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/usr/local/apache2/htdocs"
<Directory "/usr/local/apache2/htdocs">
    #
    # Possible values for the Options directive are "None", "All",
    # or any combination of:
    #   Indexes Includes FollowSymLinks SymLinksIfOwnerMatch ExecCGI MultiViews
    #
    # Note that "MultiViews" must be named *explicitly* --- "Options All"
    # doesn't give it to you.

```

```

#
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.4/mod/core.html#options
# for more information.
#
Options Indexes FollowSymLinks

#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   AllowOverride FileInfo AuthConfig Limit
#
AllowOverride None

#
# Controls who can get stuff from this server.
#
Require all granted

# Redirection HTTP vers HTTPS
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule ^ https:// %{HTTP_HOST}%{REQUEST_URI} [L,R=301]
</Directory>

#
# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
#
<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>

#
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
<Files ".ht*">
    Require all denied
</Files>

#
# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here. If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.
#
ErrorLog /proc/self/fd/2

#

```

```

# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
#
LogLevel warn

<IfModule log_config_module>
#
# The following directives define some format nicknames for use with
# a CustomLog directive (see below).
#
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common

<IfModule logio_module>
# You need to enable mod_logio.c to use %I and %O
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %I %O"
combinedio
</IfModule>

#
# The location and format of the access logfile (Common Logfile Format).
# If you do not define any access logfiles within a <VirtualHost>
# container, they will be logged here. Contrariwise, if you *do*
# define per-<VirtualHost> access logfiles, transactions will be
# logged therein and *not* in this file.
#
CustomLog /proc/self/fd/1 common

#
# If you prefer a logfile with access, agent, and referer information
# (Combined Logfile Format) you can use the following directive.
#
#CustomLog "logs/access_log" combined
</IfModule>

<IfModule alias_module>
#
# Redirect: Allows you to tell clients about documents that used to
# exist in your server's namespace, but do not anymore. The client
# will make a new request for the document at its new location.
# Example:
# Redirect permanent /foo http://www.example.com/bar

#
# Alias: Maps web paths into filesystem paths and is used to
# access content that does not live under the DocumentRoot.
# Example:
# Alias /webpath /full/filesystem/path
#
# If you include a trailing / on /webpath then the server will

```

```

# require it to be present in the URL. You will also likely
# need to provide a <Directory> section to allow access to
# the filesystem path.

#
# ScriptAlias: This controls which directories contain server scripts.
# ScriptAliases are essentially the same as Aliases, except that
# documents in the target directory are treated as applications and
# run by the server when requested rather than as documents sent to the
# client. The same rules about trailing "/" apply to ScriptAlias
# directives as to Alias.
#
ScriptAlias /cgi-bin/ "/usr/local/apache2/cgi-bin/"

</IfModule>

<IfModule cgid_module>
#
# ScriptSock: On threaded servers, designate the path to the UNIX
# socket used to communicate with the CGI daemon of mod_cgid.
#
#Scriptsock cgisock
</IfModule>

#
# "/usr/local/apache2/cgi-bin" should be changed to whatever your ScriptAliased
# CGI directory exists, if you have that configured.
#
<Directory "/usr/local/apache2/cgi-bin">
    AllowOverride None
    Options None
    Require all granted
</Directory>

<IfModule headers_module>
#
# Avoid passing HTTP_PROXY environment to CGI's on this or any proxied
# backend servers which have lingering "httpoxy" defects.
# 'Proxy' request header is undefined by the IETF, not listed by IANA
#
    RequestHeader unset Proxy early
</IfModule>

<IfModule mime_module>
#
# TypesConfig points to the file containing the list of mappings from
# filename extension to MIME-type.
#
#TypesConfig conf/mime.types

#

```

```

# AddType allows you to add to or override the MIME configuration
# file specified in TypesConfig for specific file types.
#
#AddType application/x-gzip .tgz
#
# AddEncoding allows you to have certain browsers uncompress
# information on the fly. Note: Not all browsers support this.
#
#AddEncoding x-compress .Z
#AddEncoding x-gzip .gz .tgz
#
# If the AddEncoding directives above are commented-out, then you
# probably should define those extensions to indicate media types:
#
AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz

#
# AddHandler allows you to map certain file extensions to "handlers":
# actions unrelated to filetype. These can be either built into the server
# or added with the Action directive (see below)
#
# To use CGI scripts outside of ScriptAliased directories:
# (You will also need to add "ExecCGI" to the "Options" directive.)
#
#AddHandler cgi-script .cgi

# For type maps (negotiated resources):
#AddHandler type-map var

#
# Filters allow you to process content before it is sent to the client.
#
# To parse .shtml files for server-side includes (SSI):
# (You will also need to add "Includes" to the "Options" directive.)
#
#AddType text/html .shtml
#AddOutputFilter INCLUDES .shtml
</IfModule>

#
# The mod_mime_magic module allows the server to use various hints from the
# contents of the file itself to determine its type. The MIMEMagicFile
# directive tells the module where the hint definitions are located.
#
#MIMEMagicFile conf/magic

#
# Customizable error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects
#

```

```

# Some examples:
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
#
#
# MaxRanges: Maximum number of Ranges in a request before
# returning the entire resource, or one of the special
# values 'default', 'none' or 'unlimited'.
# Default setting is to accept 200 Ranges.
#MaxRanges unlimited

#
# EnableMMAP and EnableSendfile: On systems that support it,
# memory-mapping or the sendfile syscall may be used to deliver
# files. This usually improves server performance, but must
# be turned off when serving from networked-mounted
# filesystems or if support for these functions is otherwise
# broken on your system.
# Defaults: EnableMMAP On, EnableSendfile Off
#
#EnableMMAP off
#EnableSendfile on

# Supplemental configuration
#
# The configuration files in the conf/extra/ directory can be
# included to add extra features or to modify the default configuration of
# the server, or you may simply copy their contents here and change as
# necessary.

# Server-pool management (MPM specific)
#Include conf/extra/httpd-mpm.conf

# Multi-language error messages
#Include conf/extra/httpd-multilang-errordoc.conf

# Fancy directory listings
#Include conf/extra/httpd-autoindex.conf

# Language settings
#Include conf/extra/httpd-languages.conf

# User home directories
#Include conf/extra/httpd-userdir.conf

# Real-time info on requests and configuration
#Include conf/extra/httpd-info.conf

```

```

# Virtual hosts
#Include conf/extra/httpd-vhosts.conf

# Local access to the Apache HTTP Server Manual
#Include conf/extra/httpd-manual.conf

# Distributed authoring and versioning (WebDAV)
#Include conf/extra/httpd-dav.conf

# Various default settings
#Include conf/extra/httpd-default.conf

# Configure mod_proxy_html to understand HTML4/XHTML1
<IfModule proxy_html_module>
Include conf/extra/proxy-html.conf
</IfModule>

# Secure (SSL/TLS) connections
Include conf/extra/httpd-ssl.conf
#
# Note: The following must be present to support
#       starting without SSL on platforms with no /dev/random equivalent
#       but a statically compiled-in mod_ssl.
#
<IfModule ssl_module>
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
</IfModule>

```

index.html

```

<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Documentation Raspberry Pi</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f2f2f2;
        }
        .container {
            max-width: 1200px;
            margin: 0 auto;
            padding: 20px;
        }
        header {
            background-color: #333;
            color: #fff;

```

```

        padding: 10px 0;
        text-align: center;
    }
    nav {
        background-color: #444;
        color: #fff;
        padding: 20px;
        text-align: left;
        height: 100%;
        position: fixed;
        left: 0;
        top: 0;
    }
    nav ul {
        list-style-type: none;
        margin: 0;
        padding: 0;
        height: 100%;
        display: flex;
        flex-direction: column;
    }
    nav ul li {
        display: inline;
        margin: 10px 0;
    }
    nav ul li a {
        color: #fff;
        text-decoration: none;
        display: block;
    }
    section {
        padding: 20px 0;
    }
    footer {
        background-color: #333;
        color: #fff;
        text-align: center;
        padding: 10px 0;
        position: static;
        bottom: 0;
        width: 100%;
    }
    section:last-child {
        margin-bottom: 100px;
    }

```

</style>

</head>

<body>

    <header>

        <h1>Documentation utilisateur "Utilisation simplifiée d'un Raspberry Pi à domicile"</h1>

```

</header>
<nav>
    <ul>
        <li><a href="#Prérequis">Prérequis</a></li>
        <li><a href="#Installation de Raspberry Pi OS">Installation de Raspberry Pi OS</a></li>
            <li><a href="#Configuration de base du Raspberry Pi">Configuration de base du Raspberry Pi</a></li>
                <li><a href="#Configuration du service d'impression (CUPS)">Configuration du service d'impression (CUPS)</a></li>
                <li><a href="#Configuration du NAS et de la sauvegarde">Configuration du NAS et de la sauvegarde</a></li>
                <li><a href="#Mise en place de l'impression sur macOS">Mise en place de l'impression sur macOS</a></li>
                <li><a href="#Mise en place de l'impression sur Windows">Mise en place de l'impression sur Windows</a></li>
                <li><a href="#Mise en place de la sauvegarde sur macOS">Mise en place de la sauvegarde sur macOS</a></li>
                <li><a href="#Mise en place de la sauvegarde sur Windows">Mise en place de la sauvegarde sur Windows</a></li>
                <li><a href="#Mise en place du NAS sur macOS">Mise en place du NAS sur macOS</a></li>
                <li><a href="#Mise en place du NAS sur Windows">Mise en place du NAS sur Windows</a></li>
            </ul>
        </nav>
        <div class="container">
            <section id="Prérequis">
                <h2>Prérequis</h2>
                <ul>
                    <li>Connexion internet</li>
                    <li>Connaissances minimales en informatique</li>
                    <li>Un ordinateur</li>
                    <li>Un Raspberry Pi</li>
                    <li>Une carte micro SD</li>
                    <li>Une imprimante</li>
                    <li>Un stockage externe USB</li>
                </ul>
            </section>
            <section id="Installation de Raspberry Pi OS">
                <h2>Installation de Raspberry Pi OS</h2>
                <ol>
                    <li>Pour commencer, télécharger l'outil de création d'image officielle de Raspberry PI depuis le site web de Raspberry.</li>
                    <li>Dans la première colonne, choisir « aucun filtre » pour afficher toutes les images disponibles.</li>
                    <li>Dans la seconde colonne, choisir « Raspberry PI OS (64-bit) ».</li>
                    <li>Dans troisième colonne, choisir la carte SD sur laquelle l'image sera installée.</li>
                    <li>Cliquer sur « suivant », un pop-up apparaît, cliquer sur « modifier réglages »</li>
                </ol>
            </section>
        </div>
    </body>
</html>

```

```

<li>Dans le panneau général, définir le nom d'hôte, par exemple « pi1 », définir le nom d'utilisateur, par exemple « pi1 » et un mot de passe, définir le fuseau horaire sur « Europe/Zurich » et définir le clavier sur « ch ». Dans le panneau services, cocher « Activer SSH ». Dans le panneau options, décocher « Activer la télémétrie ».</li>
<li>Cliquer sur « Enregistrer » et valider les paramètres en cliquant sur « Oui ». Valider le formatage de la carte SD en cliquant sur « Oui » et valider toutes les demandes du système.</li>
<li>Attendre la création de la carte SD lorsque la carte est prête un pop-up apparaît pour indiquer s'il a réussi la création de la carte SD.</li>
</ol>
</section>
<section id="Configuration de base du Raspberry Pi">
    <h2>Configuration de base du Raspberry Pi</h2>
    <ol>
        <li>Insérer la carte SD préconfigurée dans le Pi et le brancher au courant ainsi que lui brancher un câble Ethernet.</li>
        <li>Attendre que le Pi démarre et sur un ordinateur ouvrir un terminal et lancer cette commande (en remplaçant user par l'utilisateur et host par le nom d'hôte créé précédemment) :</li>
            <p>ssh user@host.local</p>
            <p>Valider avec « yes » et entrer le mot de passe de l'utilisateur préconfiguré et nous voilà dans le terminal du Pi.</p>
            <li>Mettre à jour le Raspberry Pi avec cette commande :</li>
            <p>sudo apt -y update && sudo apt -y full-upgrade</p>
            <li>Installer les programmes nécessaires au bon fonctionnement du Raspberry Pi :</li>
                <p>sudo apt -y install cups samba samba-common-bin</p>
            </ol>
        </section>
        <section id="Configuration du service d'impression (CUPS)">
            <h2>Configuration du service d'impression (CUPS)</h2>
            <ol>
                <li>Rajout de l'utilisateur actuel dans le groupe lpadmin</li>
                <p>sudo usermod -aG lpadmin whoami</p>
                <li>Éditer le fichier de configuration du serveur d'impression :</li>
                <p>Sudo nano /etc/cups/cupsd.conf</p>
                <p>Dans le fichier, rajouter les lignes manquantes :</p>
                <p>
                    Listen *:631
                    <Location />
                    Order allow,deny
                    Allow all
                    </Location>
                    <Location /admin>
                        Order allow,deny
                        Allow all
                    </Location>
                </p>
                <li>Après avoir édité le fichier de configuration du serveur d'impression, redémarrer le service pour appliquer les paramètres.</li>
                <p>sudo service cups restart</p>
            </ol>
        </section>
    </ol>
</section>

```

```

<li>Ouvrir un navigateur et entrer l'url « host.local:631 » pour accéder à la page d'administration de CUPS en rentrant les informations de connexion de l'utilisateur qui a été ajouté au groupe « lpadmin » précédemment et une fois connecté, aller sur « Ajouter une imprimante ».</li>
    <li>Choisir l'imprimante dans la liste et cliquer sur « suivant ».</li>
    <li>Définir le nom de l'imprimante pour y accéder en ligne, une description de l'imprimante et sa localisation, cliquer sur la case à cocher pour partager l'imprimante sur le réseau et valider avec « suivant ».</li>
        <li>Sélectionner le modèle le plus proche de l'imprimante actuelle dans la liste et valider la sélection en cliquant sur « Ajouter l'imprimante ».</li>
            <li>L'imprimante est désormais installée.</li>
        </ol>
    </section>
<section id="Configuration du NAS et de la sauvegarde">
    <h2>Configuration du NAS et de la sauvegarde</h2>
    <ol>
        <li>Pour la partie Samba, commencer par monter le disque dur au système.</li>
            <p>sudo mkdir /mnt/sda1 && sudo mount /dev/sda1</p>
            <li>Pour éviter d'écrire cette commande à chaque fois que le Pi redémarre, ajouter une ligne au fichier /etc/fstab pour monter directement le disque au démarrage.</li>
                <p>echo '/dev/sda1 /mnt/sda1/ ext4 defaults,noatime 0 1' | sudo tee -a /etc/fstab</p>
                <li>Une fois monté, créer les dossiers qui vont être utilisé pour le partage de fichiers. Voici des exemples :</li>
                    <p>sudo mkdir /mnt/sda1/nas</p>
                    <li>Éditer le fichier de configuration Samba pour permettre de partager les dossiers créés précédemment sur le réseau. Voici un exemple :</li>
                        <p>sudo tee -a /etc/samba/smb.conf > /dev/null <EOL
                            [nas]
                            path=/mnt/sda1/shared
                            writeable = yes
                            create mask = 0770
                            directory mask = 0770
                            public=no

                            [TimeMachine]
                            comment = Time Machine Backup
                            path = /mnt/sda1/TimeMachine
                            browseable = yes
                            writeable = yes
                            guest ok = no
                            create mask = 0770
                            directory mask = 0770
                            vfs objects = catia fruit streams_xattr
                            fruit:aapl = yes
                            fruit:time machine = yes
                            EOL
                        </p>
                    </li>
                </ol>
            </section>
        <li>Ajouter une sauvegarde pour la machine sous forme de plan de sauvegarde. Pour cela, se rendre dans la section « Sauvegarde » et cliquer sur « Ajouter une sauvegarde ».</li>
            <li>Choisir le type de sauvegarde (Time Machine) et spécifier les paramètres nécessaires (chemin de sauvegarde, intervalle de temps, etc.).</li>
            <li>Valider la configuration et lancer la sauvegarde.</li>
        </ol>
    </section>
</body>

```

```

<li>Redémarrer le service Samba pour appliquer les changements effectués dans le fichier de configuration.</li>
<p>sudo systemctl restart smbd</p>
<li>Créer les utilisateurs qui accéderont aux partages de fichiers ainsi que les groupes nécessaires. Voici des exemples :</li>
<p> sudo adduser user_nas
      sudo groupadd group_nas
</p>
<li>Créer un mot de passe aux utilisateurs Samba.</li>
<p>sudo smbpasswd -a user_nas</p>
<li>Ajouter les utilisateurs aux bons groupes.</li>
<p>sudo adduser user_nas group_nas</p>
<li>Définir les droits et les permissions aux dossiers, voici des exemples :</li>
<p> sudo chmod -R 770 /mnt/sda1
      sudo chown -R :group_nas /mnt/sda1/nas
      sudo chown -R :timemachine /mnt/sda1/TimeMachine
</p>
<li>Redémarrer le service Samba pour appliquer les changements effectués.</li>
<p>sudo systemctl restart smbd</p>
</ol>
</section>
<section id="Mise en place de l'impression sur macOS">
    <h2>Mise en place de l'impression sur macOS</h2>
    <ol>
        <li>Aller dans « Réglages systèmes » sous « Imprimantes et scanners ». Cliquer sur le bouton « + ».</li>
        <li>Dans la nouvelle fenêtre, chercher l'imprimante ajoutée dans CUPS. Valider les paramètres de l'imprimante.</li>
        <li>L'imprimante est ajoutée.</li>
    </ol>
</section>
<section id="Mise en place de l'impression sur Windows">
    <h2>Mise en place de l'impression sur Windows</h2>
    <ol>
        <li>Ouvrir « Paramètres ».</li>
        <li>Dans « Bluetooth et appareils » sous « Imprimantes et scanners ».</li>
        <li>Cliquer sur « Ajouter un appareil ».</li>
        <li>Dans la liste, chercher l'imprimante ajoutée dans CUPS et cliquer sur « Ajouter l'appareil ».</li>
        <li>L'imprimante est ajoutée.</li>
    </ol>
</section>
<section id="Mise en place de la sauvegarde sur macOS">
    <h2>Mise en place de la sauvegarde sur macOS</h2>
    <ol>
        <li>Aller dans « Réglages systèmes » sous « Time Machine ».</li>
        <li>Cliquer sur « Sélectionner un disque de sauvegarde ».</li>

```

```

        <li>Sélectionner le partage Samba pour Time Machine. Valider la connexion.</li>
            <li>Introduire les informations de connexion du compte configuré précédemment.</li>
                <li>La sauvegarde se lance.</li>
            </ol>
        </section>
        <section id="Mise en place de la sauvegarde sur Windows">
            <h2>Mise en place de la sauvegarde sur Windows</h2>
            <ol>
                <li>Ouvrir l'ancien panneau de configuration avec Win + R et enter « control ».</li>
                    <li>Cliquer sur « Sauvegarder et restaurer (Windows 7) ».</li>
                    <li>Cliquer sur « Configurer la sauvegarde ».</li>
                    <li>Cliquer sur « Enregistrer sur un réseau ».</li>
                    <li>Introduire l'adresse du Raspberry Pi et les informations de connexion du compte configuré précédemment et valider avec « OK ». Cliquer sur « suivant ».</li>
                    <li>Laisser les paramètres par défaut et valider avec « OK ». Vérifier les paramètres de sauvegarde et valider avec « Enregistrer les paramètres et exécuter la sauvegarde ».</li>
                    <li>La sauvegarde se lance.</li>
                </ol>
            </section>
            <section id="Mise en place du NAS sur macOS">
                <h2>Mise en place du NAS sur macOS</h2>
                <ol>
                    <li>Ouvrir la fenêtre de connexion avec CMD + K et introduire l'adresse du Raspberry Pi et valider avec « Connecter ».</li>
                    <li>Introduire les informations de connexion du compte configuré précédemment et valider avec « Connecter ».</li>
                    <li>Le partage est connecté.</li>
                </ol>
            </section>
            <section id="Mise en place du NAS sur Windows">
                <h2>Mise en place du NAS sur Windows</h2>
                <ol>
                    <li>Ouvrir une fenêtre dans l'explorateur de fichier et cliquer sur « Ce PC », dans le menu burger, sélectionner « Connecter un lecteur réseau ». Sélectionner la lettre de chemin d'accès, introduire l'adresse du Raspberry Pi, cocher « Se reconnecter lors de la connexion » et valider avec « Terminer ».</li>
                    <li>Introduire les informations de connexion du compte configuré précédemment, cocher « Mémoriser mes informations d'identification » et valider avec « OK ». Le partage est connecté.</li>
                </ol>
            </section>
        </div>
        <footer>
            <p>&copy; Février 2024 – TPI Homepi – Raspberry Pi : Utilisation à domicile et services avancés – Documentation utilisateur "Utilisation simplifiée d'un Raspberry Pi à domicile"</p>
        </footer>

```

```
</body>
</html>
```

## Dockerfile

```
FROM debian:stable-slim

# Installation des mises à jour, des dépendances de Webmin, des outils réseau et de CUPS
RUN apt -y update && apt -y full-upgrade
RUN apt -y install perl libnet-ssleay-perl openssl libauthen-pam-perl libpam-runtime
libio-pty-perl apt-show-versions python-is-python3 wget unzip shared-mime-info net-tools
cups apache2-utils curl

# Suppression des listes de paquets
RUN rm -rf /var/lib/apt/lists/*

# Installation du module Perl DateTime
RUN cpan DateTime

# Téléchargement et installation de Webmin
RUN curl -o setup-repos.sh https://raw.githubusercontent.com/webmin/webmin/master/setup-
repos.sh && yes | sh setup-repos.sh && apt -y update && apt -y install webmin --install-
recommends

# Exposer les ports nécessaires
EXPOSE 631 10000

# Copier le fichier cupsd.conf personnalisé dans le conteneur
COPY ../conf/cupsd.conf /etc/cups/cupsd.conf

# Copier le fichier d'entrée personnalisé dans le conteneur
COPY entrypoint.sh /entrypoint.sh

# Donner les permissions d'exécution au script d'entrée
RUN chmod +x /entrypoint.sh

# Lancer les services lors du démarrage du conteneur
CMD ["/bin/bash", "-c", "/entrypoint.sh && tail -f /dev/null"]
```

## docker-compose.yml

```
version: '3'
services:
  apache:
    image: httpd:latest
    container_name: apache
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - ./apache:/usr/local/apache2/htdocs
      - ./apache/httpd.conf:/usr/local/apache2/conf/httpd.conf
      - ./apache/httpd-ssl.conf:/usr/local/apache2/conf/extrahttpd-ssl.conf
```

```

- ./apache/server.crt:/usr/local/apache2/conf/server.crt
- ./apache/server.key:/usr/local/apache2/conf/server.key
restart: unless-stopped

webmin:
  container_name: webmin
  build:
    context: .
    dockerfile: Dockerfile
  restart: unless-stopped
  ports:
    - "10000:10000"
    - "9631:631"

# --- Optionnel ---

watchtower:
  image: containrrr/watchtower
  container_name: watchtower
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
  restart: unless-stopped

portainer:
  image: portainer/portainer-ce:latest
  container_name: portainer
  ports:
    - "8000:8000"
    - "9443:9443"
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
    - ./portainer_data:/data
  restart: unless-stopped

```

### entrypoint.sh

```

#!/bin/bash

# Démarrer les services CUPS et Webmin
service cups start
service webmin start

# Installer le client Apache
apt-get update
apt-get install -y apache2 apache2-utils

# Changer le mot de passe de Webmin
/usr/share/webmin/changepass.pl /etc/webmin root password

# Créer le répertoire htdocs s'il n'existe pas déjà
mkdir -p /var/www/html

```

```

# Modifier le fichier de configuration d'Apache pour inclure le répertoire racine correct
sed -i 's|DocumentRoot /var/www/html|DocumentRoot /var/www/html|'
/etc/apache2/apache2.conf

# Ajouter la directive ServerName dans le fichier de configuration d'Apache
echo "ServerName localhost" >> /etc/apache2/apache2.conf

# Redémarrer Apache pour appliquer les changements
service apache2 restart

# Garder le conteneur en cours d'exécution
tail -f /dev/null

```

### HOMEpI.sh

```

#!/bin/bash

# Couleur et style de police
CO='\033[0m'
RED='\033[0;31m'
ITALIC='\033[3m'
ITALIC_RED='\033[3;31m'

# Vérification admin
if [ "$EUID" -ne 0 ]; then
    echo
    echo -e "${ITALIC_RED}Ce script nécessite des droits administrateurs, exécutez-le avec"
    sudo !$CO"
    echo
    exit
fi

# Fonction pour effectuer l'action 1
action1() {
    ansible-playbook -i ./ansible/inventory.yml ./ansible/master_playbook_install_pi.yml
}

# Fonction pour effectuer l'action 2
action2() {
    ansible-playbook -i ./ansible/inventory.yml ./ansible/playbook_install_pi1.yml
}

# Fonction pour effectuer l'action 3
action3() {
    ansible-playbook -i ./ansible/inventory.yml ./ansible/playbook_install_pi2.yml
}

# Fonction pour effectuer l'action 4
action4() {
    ansible-playbook -i ./ansible/inventory.yml ./ansible/playbook_install_pi3.yml
}

```

```

# Fonction pour effectuer l'action 5
action5() {
    ansible-playbook -i ./ansible/inventory.yml ./ansible/playbook_update.yml
}

# Fonction pour effectuer l'action 6
action6() {
    ansible-playbook -i ./ansible/inventory.yml ./ansible/playbook_cups.yml
}

# Fonction pour effectuer l'action 7
action7() {
    ansible-playbook -i ./ansible/inventory.yml ./ansible/playbook_nas.yml
}

# Fonction pour effectuer l'action 8
action8() {
    ansible-playbook -i ./ansible/inventory.yml ./ansible/playbook_rsync.yml
}

# Fonction pour effectuer l'action 9
action9() {
    ansible-playbook -i ./ansible/inventory.yml ./ansible/playbook_docker.yml
}

# Fonction pour afficher le menu
afficher_menu() {
    echo "Menu :"
    echo
    echo "1. Installer tous les Pi"
    echo "2. Installer Pi1"
    echo "3. Installer Pi2"
    echo "4. Installer Pi3"
    echo "5. Mettre à jour tout les Raspberry Pi"
    echo "6. Mettre en place CUPS"
    echo "7. Mettre en place le NAS"
    echo "8. Mettre en place Rsync"
    echo "9. Mettre en place Docker"
    echo -e "${RED}10. Quitter${CO}"
}
}

# Boucle principale
while true; do
    echo
    afficher_menu
    echo
    read -p "Choisissez une option (1-10) : " choix
    echo

    case $choix in
        1) action1 ;;

```

```
2) action2 ;;
3) action3 ;;
4) action4 ;;
5) action5 ;;
6) action6 ;;
7) action7 ;;
8) action8 ;;
9) action9 ;;
10) echo -e "${ITALIC}Au revoir !${CO}"; echo ; exit ;;
*) echo -e "${RED}Option invalide, veuillez choisir une option valide (1-
10).${CO}";;
esac
done
```