# Cartographie complète de votre infrastructure

## Les machines:

### Ubuntu

#### Config:

* RAM → 2 Go
* CPU → 2 coeurs
* HDD → 20 Go

#### Cartes Réseaux:

* Typologies →
* Type de cartes → Bridge
* Numéro d’interface → GUI

#### L’adressage:

* IP → 192.168.10.7
* DNS → 192.168.10.1
* Gateway → 192.168.10.1

### RainLoop:

#### Introduction

RainLoop est un client de messagerie web open-source léger, rapide et facile à utiliser. Il prend en charge plusieurs fournisseurs de messagerie, tels que Gmail, Yahoo, Outlook, iCloud, Zimbra, et d'autres encore, ainsi que les protocoles IMAP, POP3 et SMTP. Avec RainLoop, vous pouvez lire et envoyer des e-mails, gérer des dossiers, créer des filtres, gérer des signatures et bien plus encore. Il est également compatible avec les appareils mobiles.

#### Prérequis

* Lamp

#### Installation de RainLoop

* [Téléchargement](https://www.rainloop.net/)
  + Décompression de l’archive
* Copie des fichiers sur le serveur web

#### Configuration de RainLoop

* Configuration des autorisations de fichiers.

cd /chemin/vers/le/repertoire/rainloop

chown -R www-data:www-data .

chmod -R 775 .

cd /chemin/vers/le/repertoire/rainloop chown -R www-data:www-data . chmod -R 775 .

* Configuration du serveur web
  + Création d’un fichier rainloop.conf dans le répertoire ‘/etc/apache2/sites-available/’
  + Configuration du fichier rainloop.conf

<VirtualHost \*:80>

ServerName webmail.example.com

DocumentationRoot /chemin/vers/le/répertoire/rainloop

<Directory /chemin/vers/le/répertoire/rainloop>

AllowOverride All

Require all granted

</Directory>

ErrorLog ${APACHE\_LOG\_DIR}/rainloop\_error.log

CustomLog ${APACHE\_LOG\_DIR}/rainloop\_access.log combined

</VirtualHost>

Remplacer webmail.exemple.com → par le dns qu’on souhaite utiliser pour accéder à Rainloop.

* Acti

### Ubuntu Server

#### Config:

* RAM → 2 Go
* CPU → 2 coeurs
* HDD → 20 Go

#### Cartes Réseaux:

* Type de carte → Bridged
* Numéro d’interface → CLI

#### L’adressage:

* IP → 192.168.10.5
* DNS → 192.168.10.1
* Gateway → 192.168.10.1

#### Services installés et configurés

#### UFW - Uncomplicated Firewall

+

#### SAMBA Server

Après avoir réalisé des recherches afin de pouvoir utiliser le service SAMBA sur notre serveur Ubuntu, nous avons pu l’installer et le configurer avec succès. Voici comment installer un serveur SAMBA :

* Source  
    
   [Ubuntu Samba Shares with AD Permissions](https://www.youtube.com/watch?v=dPM-NS9bqb8)

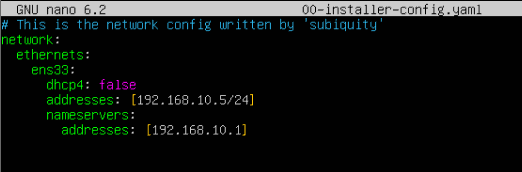
**Connecter la machine au DNS et au Réseau :**

cd /etc/netplan

sudo nano 00-installer-config.yaml



Le modifier comme celà :



Appliquer les changements :

sudo netplan apply



**L’installation du service Samba :**

Mettre à jour les paquets :

sudo apt-get update && upgrade

Installer les paquets nécessaires :

sudo apt -y install winbind libpam-winbind libnss-winbind krb5-config

Installer Samba :

sudo apt-get install samba

Nous avons installé le service Samba et ses dépendances … Place à la configuration !

Après avoir Installer le service Samba et ses dépendances, nous allons maintenant le configurer.

* Source  
    
   [Ubuntu Samba Shares with AD Permissions](https://www.youtube.com/watch?v=dPM-NS9bqb8)

**Il faut premièrement rejoindre l’AD :**

Commandes pour join l’AD :

sudo kinit admin-user



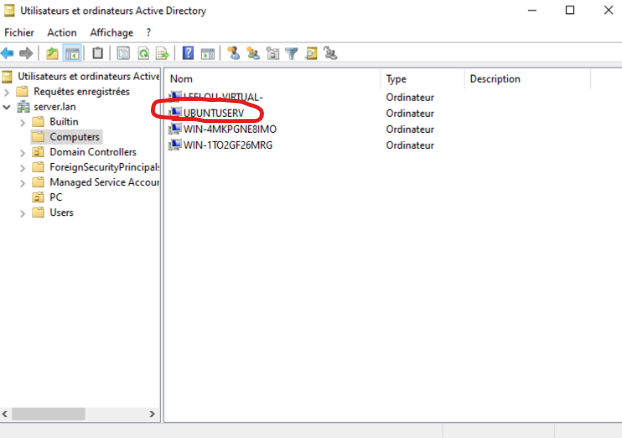
sudo apt -y install winbind libpam-winbind libnss-winbind krb5-config



sudo net ads keytab create -U ubuntuserv



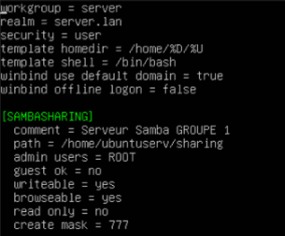
sudo net ads join -U ubuntuserv



**Configurer le serveur SAMBA :**

****nano /etc/samba/smb.conf

Nous allons éditer notre fichier .conf de cette manière :



workgroup → domaine du serveur

realm → correspond au nom de domaine

security → user-level = configuration par défaut, si le serveur accepte le mdp et utilisateurs du client, alors il n’aura plus besoin de le rentrer à nouveau son mdp et son username pour plusieurs autres shares.

template homedir → le répertoire personnel de chaque utilisateur sera créé sous le répertoire /home/ du serveur, avec le nom du domaine (%D) et le nom de l'utilisateur (%U) utilisés pour former le nom complet du répertoire.

template shell → Définit le shell par défaut pour les users quand ils se connectent au serveur via Samba.

winbind use default domain = true → Samba prendra en compte que par défaut l’utilisateur fait parti du domaine.

winbind offline logon = false → n’autorise pas la connexion en hors ligne grâce au cache des identifiants.

[SAMBASHARING] → Nom de notre share

comment → description du share affichée au client

path → chemin du dossier

admin users → détermine l’user ROOT administrateur

guest ok → n'autorise pas les connexions sans utilisateurs, demande une authentification avant l’accès au fichier.

writeable → Ce sont les permissions, de modifications de fichier (ajout, suppression ou modification)

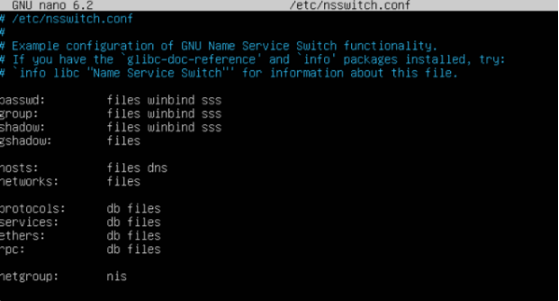
browseable → visible pour les utilisateurs du réseau

read only →Les utilisateurs peuvent modifier les fichiers, les supprimer ou en ajouter.

create mask → 777 niveaux de privilèges pour lire, modifier, et exécuter les fichiers

* Configurer la méthode d’authentification

nano etc/nsswitch.conf

  


* Créer les répertoires “home” automatiquement à l’authentification de l’utilisateur

nano /etc/pam.d/common-session

  
  
  
 Ajouter cette ligne ^^

* Rechercher le domaine et pointer le dns

nano /etc/network/interfaces

  


* Créer des utilisateurs

smbpasswd -A utilisateur

smbpasswd -e utilisateur

smbpasswd -A ROOT

smbpasswd -e ROOT



* Relancer les services:

systemctl restart networking

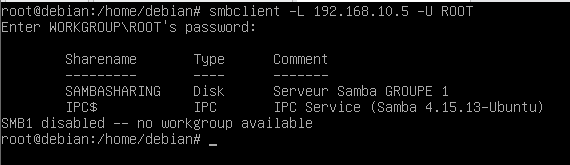
systemctl restart winbind

systemctl restart smbd

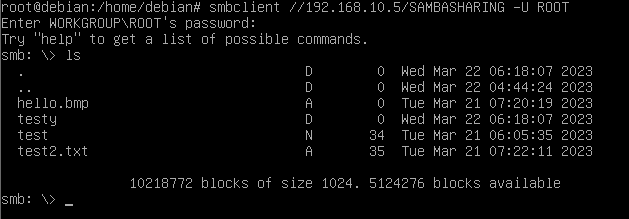
systemctl restart nmbd

* Tester :

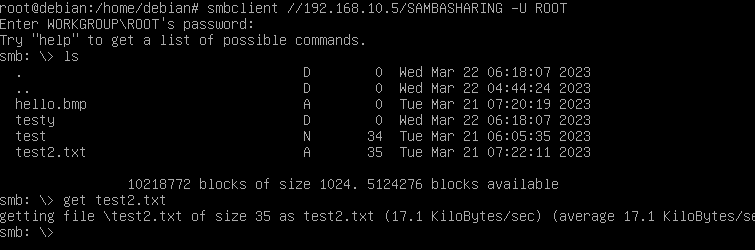
Notre share s’affiche bien.

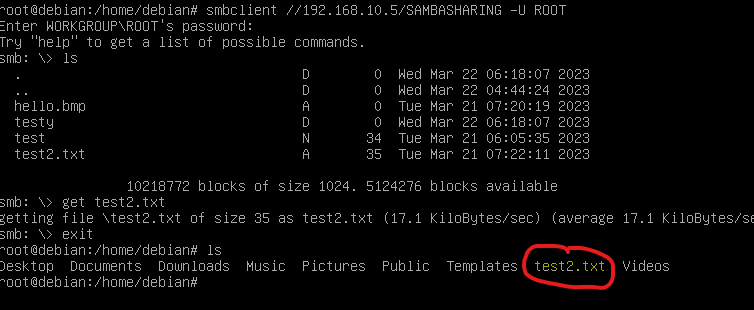


Nous avons accès au serveur SAMBA.

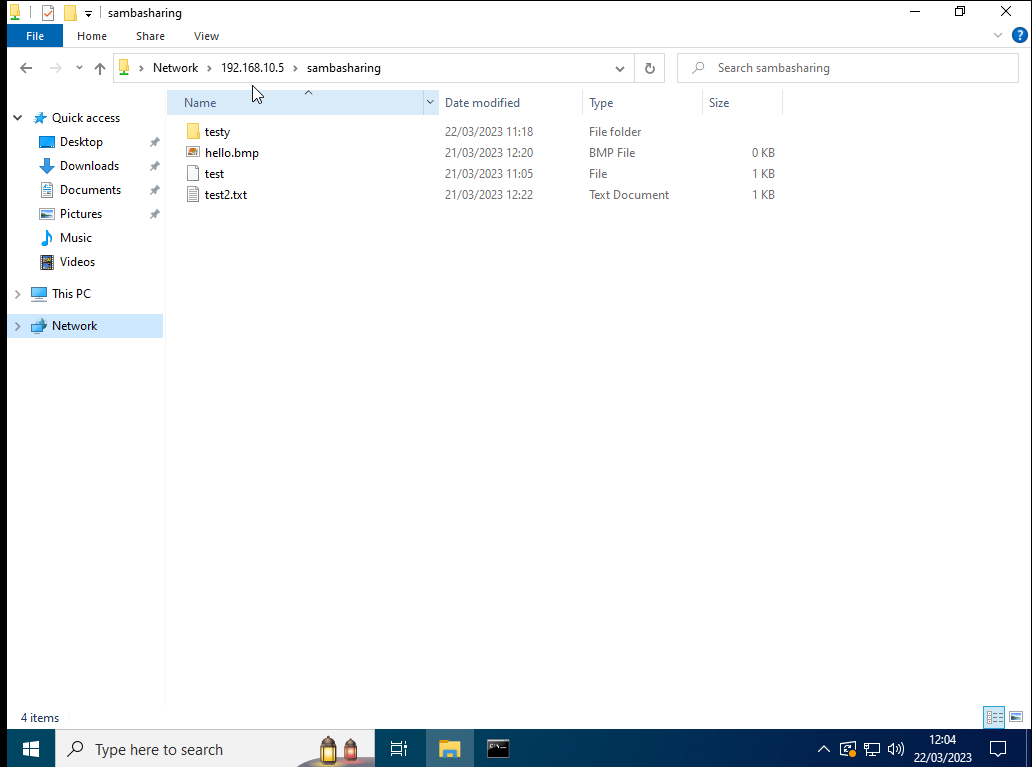


Nous pouvons bien récupérer le fichier test2.txt

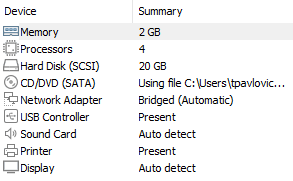




Les utilisateurs windows aussi :



CONFIG DE LA VM UBUNTU SERVEUR



### Fedora

#### Config:

* RAM → 2 Go
* CPU → 2 coeurs
* HDD → 20 Go

#### Cartes Réseaux:

* Typologies →
* Type de cartes → Bridge
* Numéro d’interface → GUI

#### L’adressage:

* IP → 192.168.10.4
* DNS → 192.168.10.1
* Gateway → 192.168.10.1

## 

## 

## 

## Configuration d’une machine cliente ADDS

Après avoir effectué quelques recherches, nous nous sommes basés sur un site afin de mettre en place notre machine cliente.

Source:

Tout d’abord, nous devons installer les prérequis ! Pour celà nous devons commencer par vérifier si la machine est à jour.

sudo apt update && apt upgrade –y



Puis nous devons installer les paquets nécessaires :

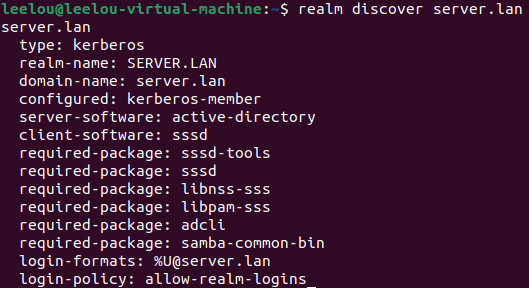
sudo apt install realmd sssd sssd-tools libnss-sss libpam-sss adcli samba-common-bin oddjob oddjob-mkhomedir packagekit –y



Ensuite, nous devons vérifier que le domaine est bien joignable. Je rappelle que notre DNS est : server.lan

realm discover server.lan

Voici le résultat:



Aucun problème ne s’affiche, nous pouvons donc passer à la prochaine étape qui est la connexion :

realm join --user=Administrateur server.lan



Nous pouvons vérifier que nous sommes bien connecté à la base de donnée :

id userv@server.lan



Nous sommes donc bien connectés…



sudo nano /etc/pam.d/common-session



Il nous suffit de rajouter cette ligne :



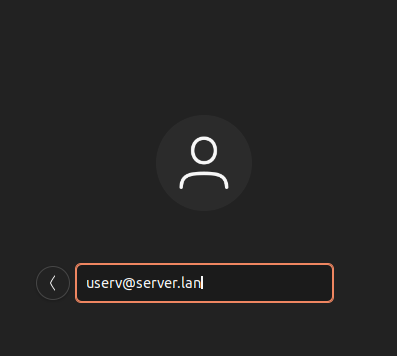
Et voilà ! Plus qu’à tester :

En CLI :

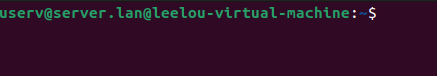


En GUI:



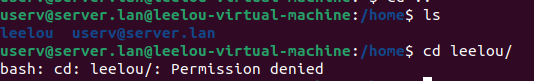


Et voila tout est fonctionnel:



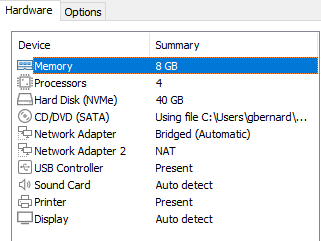


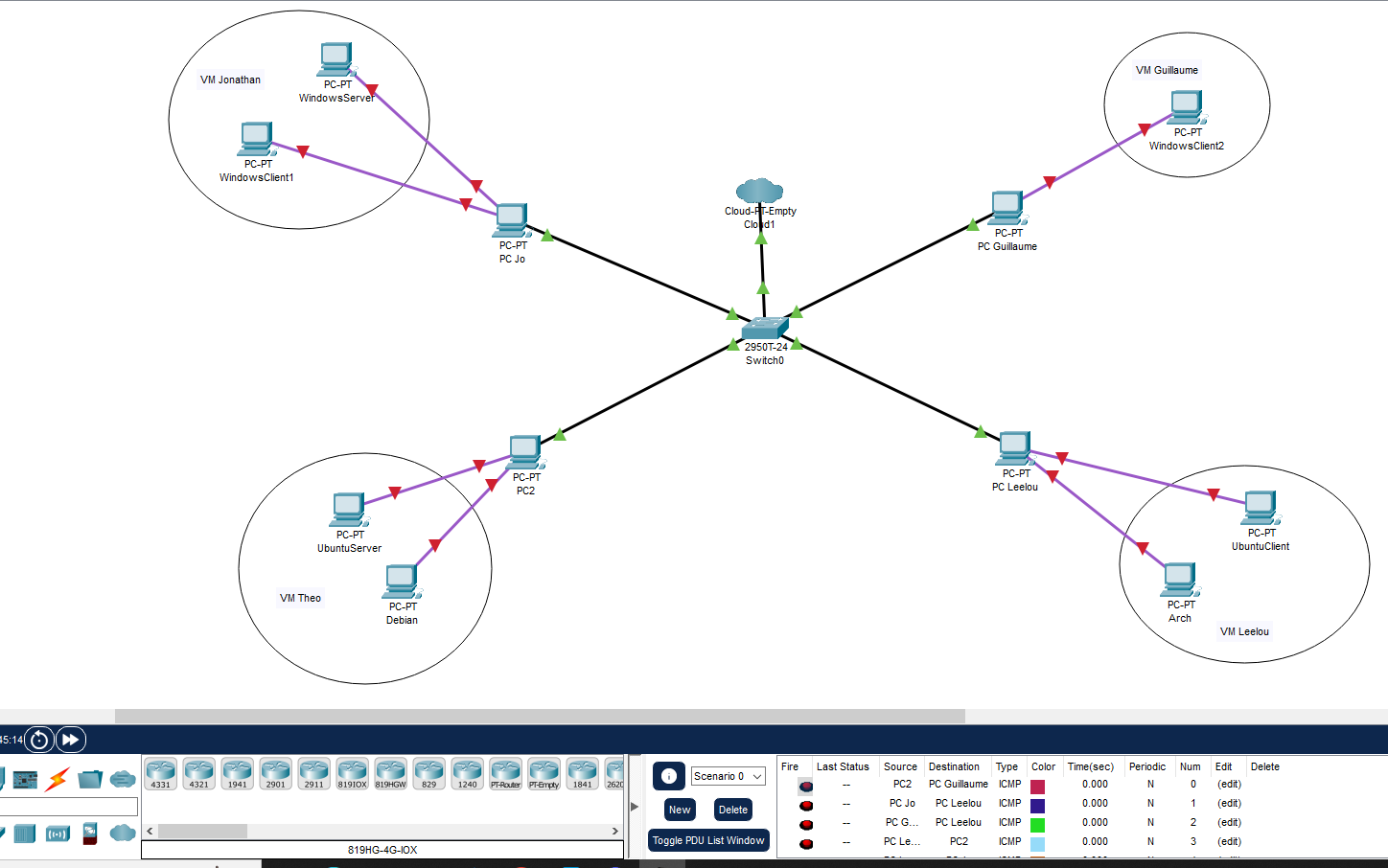
Et nous avons seulement accès à notre répertoire:



Windows client

* + CPU  
    - 8 go
    - 4 cœurs
  + HDD  
    - 40 go HDD
  + Cartes Réseaux  
    - Bridge
    - NAT
  + Numéro d’interface  
    - GUI
  + Adressage  
    - IP
      * 192.168.10.3
      * 192.168.10.2
    - DNS
      * 192.168.10.1
    - Gateway
      * 192.168.10.1



CISCO : 

## Template

### Config:

* RAM
* CPU
* HDD

### Cartes Réseaux:

* Typologies
* Type de cartes
* Numéro d’interface

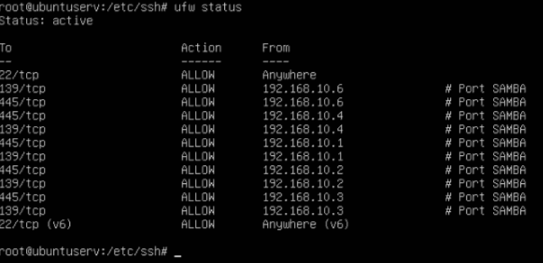
### L’adressage

IP

DNS

Gateway

# - Politique établie sur les Firewall

* UFW - Uncomplicated Firewall  
    
   La politique que l’on a établie sur le pare-feu UFW consiste à autoriser seulement le trafic nécessaire à l'opération des services et des applications utilisés par le système. Tous les autres ports et protocoles sont bloqués par défaut.  
    
   Voici la configuration du Firewall UFW établi sur la VM Ubuntu Serveur.  
    
     
    
   Nous avons autorisé le trafic sortant mais bloqué le trafic entrant en acceptant seulement les interactions nécessaires au bon fonctionnement du service SAMBA installé préalablement (PORTS: **445** et **139** en ciblant les IP de notre réseau).  
    
   Pour bloquer tout le trafic entrant, la commande est la suivante :  
    
     
    
   Pour accepter seulement une adresse IP en ciblant un port définis, on utilise la commande suivante :  
    
    
    
   Par défaut, les connections SSH sont bloquées alors, nous allons permettre les connections SSH avec la commande suivante :



# - La configuration de votre serveur SMTP avec preuves de réception de mails.

# 

# 

# - La méthodologie de gestion du projet

### - Les méthodes de travail du référent projet

- Théo :

### - Méthodes de travail de chaque profil projet

- Leelou

- Jonathan

- Guillaume : pas là

# - Les implications

### - Qui à fait quoi

* Leelou
* Théo
* Jonathan
* Guillaume : 1 Windows Client, Cisco Packet Tracer, Recherche UFW et WSUS

### - Niveau d’investissement

### - Commentaires globalisé sur chaque personne du projet

# 

# 

# 

# - Conclusion sur le projet, et sur la viabilité de votre infrastructure

Calcul viabilité :

De 1 à 10 - A combien pourriez-vous déployer cette infrastructure au sein d’une école. A destination de tous.