

## *Rapport*

# **Projet d'administration système**

Filière:L2 TELECOMS

THÈME:

Installation et configuration des serveurs:  
SAMBA,ASTERISK et DHCP

Présenté par:

MALIKA AMINE SALEH

Sous la direction de:

M.Moustapha(chargé du cours)

Année universitaire : 2023-2024

# Contents

0.1	INTRODUCTION . . . . .	2
0.2	Installation et configuration de serveur de partage des fichiers: SAMBA . . . . .	3
0.2.1	Installation de serveur SAMBA . . . . .	3
0.2.2	Configuration de serveur SAMBA . . . . .	3
0.3	Installation et configuration de serveur de téléphonie ip Asterisk . . . . .	9
0.3.1	Installation . . . . .	9
0.3.2	Configuration . . . . .	9
0.4	Installation et configuration de serveur DHCP . . . . .	15
0.4.1	Installation . . . . .	15
0.4.2	Configuration . . . . .	16
0.4.3	Configuration des options spécifiques . . . . .	17
0.5	Conclusion . . . . .	18

## 0.1 INTRODUCTION

Les systèmes de téléphonie, de partage de fichiers et de gestion des adresses IP sont des éléments essentiels pour le bon fonctionnement d'un réseau informatique. Dans ce rapport, nous aborderons l'installation et la configuration de trois services clés sur un serveur Ubuntu : Asterisk, Samba et DHCP.

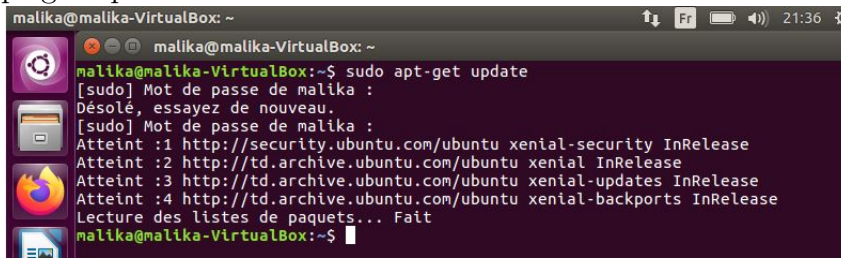
Asterisk est une plateforme de téléphonie open source qui permet la mise en place d'un système de VoIP (Voice over IP) flexible et personnalisable. Samba, quant à lui, est un logiciel libre qui facilite le partage de fichiers et d'imprimantes entre systèmes Windows et Linux. Enfin, le protocole DHCP (Dynamic Host Configuration Protocol) est utilisé pour attribuer automatiquement des adresses IP aux périphériques connectés à un réseau.

Ce rapport détaillera les étapes nécessaires à l'installation de ces services sur un serveur Ubuntu, ainsi que les configurations requises pour assurer leur bon fonctionnement. Nous examinerons également les avantages et les cas d'utilisation de chaque service, ainsi que les bonnes pratiques de sécurité à mettre en place pour protéger ces services contre les menaces potentielles.

L'objectif de ce rapport est de fournir un guide complet et pratique pour mettre en place un serveur Ubuntu polyvalent, capable de répondre aux besoins de téléphonie, de partage de fichiers et de gestion des adresses IP au sein d'un réseau informatique.

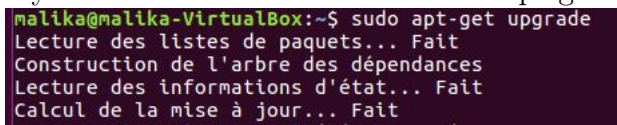
## 0.2 Installation et configuration de serveur de partage des fichiers: SAMBA

Avant l'installation des 3 serveurs, nous avons mis à jour le système en utilisant la commande "apt-get update":



```
malika@malika-VirtualBox: ~  
malika@malika-VirtualBox:~$ sudo apt-get update  
[sudo] Mot de passe de malika :  
Désolé, essayez de nouveau.  
[sudo] Mot de passe de malika :  
Atteint :1 http://security.ubuntu.com/ubuntu xenial-security InRelease  
Atteint :2 http://td.archive.ubuntu.com/ubuntu xenial InRelease  
Atteint :3 http://td.archive.ubuntu.com/ubuntu xenial-updates InRelease  
Atteint :4 http://td.archive.ubuntu.com/ubuntu xenial-backports InRelease  
Lecture des listes de paquets... Fait  
malika@malika-VirtualBox:~$
```

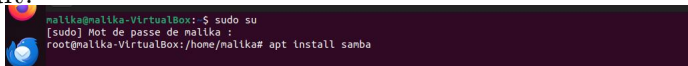
cette ligne permet de faire la lecture des paquets. l'étape suivante est celle de la mise à jour du système en utilisant la commande "apt-get upgrade":



```
malika@malika-VirtualBox:~$ sudo apt-get upgrade  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances  
Lecture des informations d'état... Fait  
Calcul de la mise à jour... Fait
```

### 0.2.1 Installation de serveur SAMBA

Samba est une implémentation open source du protocole SMB/CIFS qui permet à des systèmes Unix-like de partager des fichiers et des imprimantes avec des systèmes Windows. Pour l'installer, il faut généralement installer le paquet logiciel samba à l'aide du gestionnaire de paquets de la distribution Linux utilisée. Une fois installé, le serveur Samba peut être activé pour permettre le partage de fichiers et d'imprimantes entre les machines du réseau, facilitant ainsi l'interopérabilité entre les systèmes Windows et les systèmes Unix-like. le processus d'installation se fait comme suit:

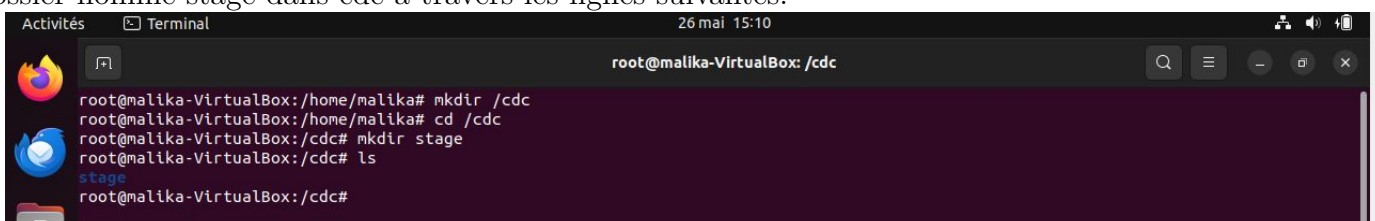


```
malika@malika-VirtualBox:~$ sudo su  
[sudo] Mot de passe de malika :  
root@malika-VirtualBox:/home/malika# apt install samba
```

### 0.2.2 Configuration de serveur SAMBA

La configuration d'un serveur Samba implique généralement de définir les partages de fichiers et d'imprimantes dans le fichier de configuration principal, smb.conf. Les paramètres tels que les permissions d'accès, les utilisateurs autorisés, les chemins des répertoires partagés et d'autres options de partage peuvent être spécifiés dans ce fichier. En outre, l'authentification des utilisateurs peut être gérée en utilisant un backend d'authentification tel que les utilisateurs locaux, un serveur LDAP ou un contrôleur de domaine Active Directory. Une fois la configuration terminée, le service Samba doit être redémarré pour appliquer les changements. La configuration de Samba peut être complexe en fonction des besoins et des exigences de sécurité du réseau, mais une fois correctement configuré, le serveur Samba permet un partage de fichiers et d'imprimantes fiable et sécurisé entre des systèmes Windows et des systèmes Unix-like.

pour pouvoir accéder à la configuration, nous avons tout d'abord créer un répertoire cdc et un dossier nommé stage dans cdc à travers les lignes suivantes:



```
root@malika-VirtualBox: /cdc  
root@malika-VirtualBox:/home/malika# mkdir /cdc  
root@malika-VirtualBox:/home/malika# cd /cdc  
root@malika-VirtualBox:/cdc# mkdir stage  
root@malika-VirtualBox:/cdc# ls  
stage  
root@malika-VirtualBox:/cdc#
```

Après la création, nous avons ouvrir le fichier de configuration da SAMBA, généralement situé à

/etc/samba/smb.conf, en utilisant l'éditeur de texte nano. Nous avons configuré les paramètres de partage en définissant le répertoire à partager, les autorisations d'accès et les utilisateurs:

```
[stage]
comment = partage
writable = yes
browseable = yes
path = cdc/stage
```

Après cette étape, nous passerons à la création des utilisateurs. Nous avons créé des utilisateurs SAMBA en utilisant la commande, "adduser utilisateur", "smbpasswd -a nom d'utilisateur" et en définissant le mot de passe de l'utilisateur. Dans notre projet nous avons pris "tachefine" comme nom d'utilisateur et 5 espaces comme mot de passe.

```
Terminal
root@malika-VirtualBox: /cdc
root@malika-VirtualBox:/cdc# adduser tachefine
Ajout de l'utilisateur « tachefine » ...
Ajout du nouveau groupe « tachefine » (1001) ...
Ajout du nouvel utilisateur « tachefine » (1001) avec le groupe « tachefine » ..
.
Création du répertoire personnel « /home/tachefine »...
Copie des fichiers depuis « /etc/skel »...
Entrez le nouveau mot de passe UNIX :
Retapez le nouveau mot de passe UNIX :
passwd : le mot de passe a été mis à jour avec succès
Modification des informations relatives à l'utilisateur tachefine
Entrez la nouvelle valeur ou « Entrée » pour conserver la valeur proposée
Nom complet []:

root@malika-VirtualBox:/cdc# smbpasswd -a tachefine
New SMB password:
Retype new SMB password:
Added user tachefine.
root@malika-VirtualBox:/cdc#
```

Une fois créés les utilisateurs, nous avons redémarré le service SAMBA pour appliquer les modifications en utilisant la commande:

```
root@malika-VirtualBox: /cdc
root@malika-VirtualBox:/cdc# systemctl restart smb
root@malika-VirtualBox:/cdc#
```

Après le redémarrage, nous avons vérifié l'état du serveur en utilisant la commande "systemctl status smbd"

```
root@malika-VirtualBox: /cdc
root@malika-VirtualBox:/cdc# systemctl status smbd
● smbd.service - Samba SMB Daemon
   Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2024-05-26 15:23:15 WAT; 38s ago
     Docs: man:smbd(8)
            man:samba(7)
            man:smb.conf(5)
   Process: 33339 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile (code=exited, status=0/SUCCESS)
   Main PID: 33350 (smbd)
    Status: "smbd: ready to serve connections..."
     Tasks: 4 (limit: 2261)
    Memory: 8.8M
       CPU: 160ms
    CGroup: /system.slice/smbd.service
            └─33350 /usr/sbin/smbd --foreground --no-process-group
              33352 /usr/sbin/smbd --foreground --no-process-group
              33353 /usr/sbin/smbd --foreground --no-process-group
              33354 /usr/lib/x86_64-linux-gnu/samba/samba-bgqd --ready-signal-fd=45 --parent-watch-fd=11 --debuglevel=0 -F

mai 26 15:23:15 malika-VirtualBox systemd[1]: Starting Samba SMB Daemon...
mai 26 15:23:15 malika-VirtualBox systemd[1]: Started Samba SMB Daemon.
root@malika-VirtualBox:/cdc#
```

Après avoir vérifié le status du serveur, nous avons accédé depuis notre machine réelle (possédant le système windows) à la machine virtuelle à travers l'adresse ip de la machine virtuelle.

```
Activités Terminal 26 mai 15:26 root@malika-VirtualBox: /home/malika
root@malika-VirtualBox:/cdc# exit
exit
malika@malika-VirtualBox:~$ sudo su
root@malika-VirtualBox:/home/malika# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.43.94 netmask 255.255.255.0 broadcast 192.168.43.255
    inet6 fe80::ebe3:62a3:96ee:939d prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:ea:b1:5e txqueuelen 1000 (Ethernet)
    RX packets 187399 bytes 249840723 (249.8 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 95103 bytes 11207060 (11.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Boucle locale)
    RX packets 657 bytes 56787 (56.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 657 bytes 56787 (56.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@malika-VirtualBox:/home/malika#
```

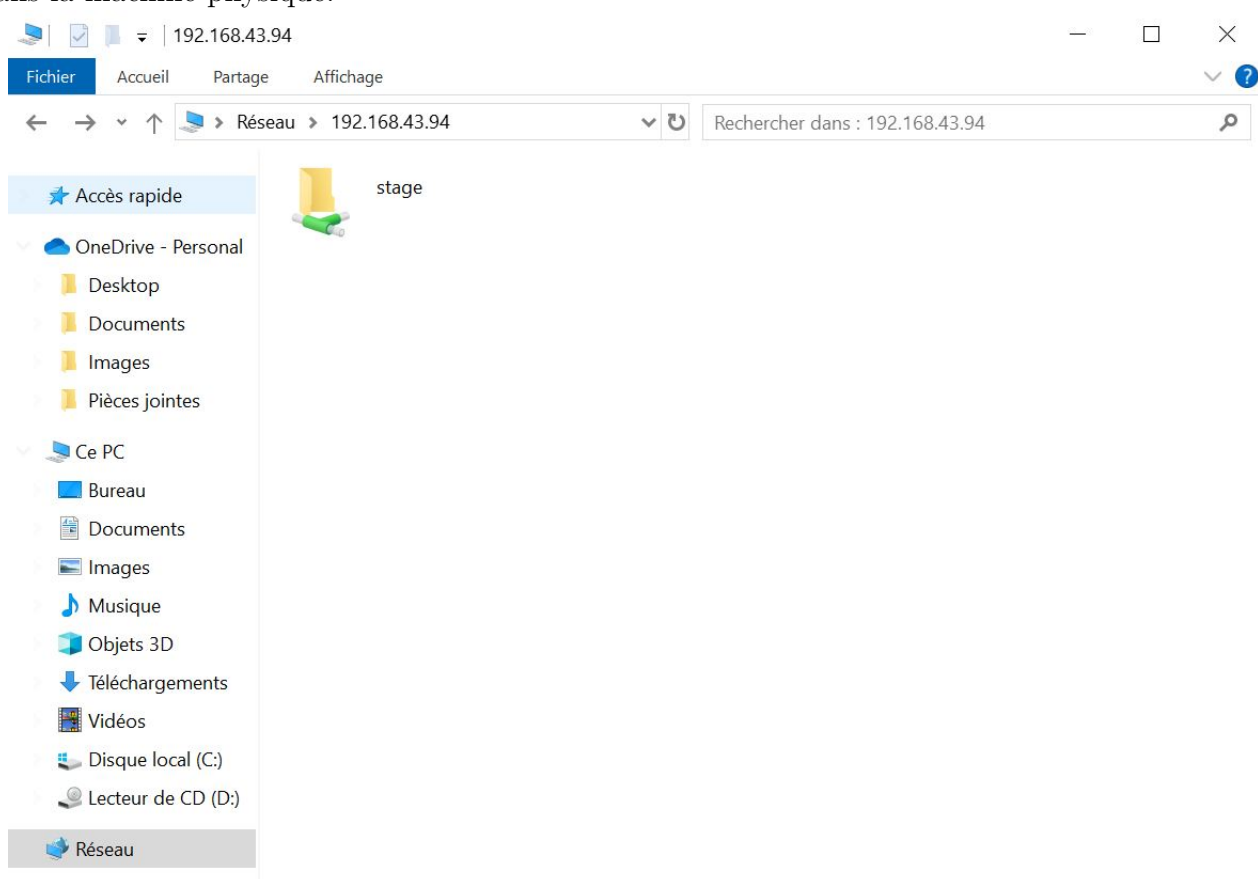
l'image ci-dessus nous montre l'adresse ip de la machine qui nous permet d'accéder à cette dernière. L'accès à la machine virtuelle depuis la machine physique se fait comme suit:

- \*ouvrir l'explorateur des fichiers

- \*aller dans réseau

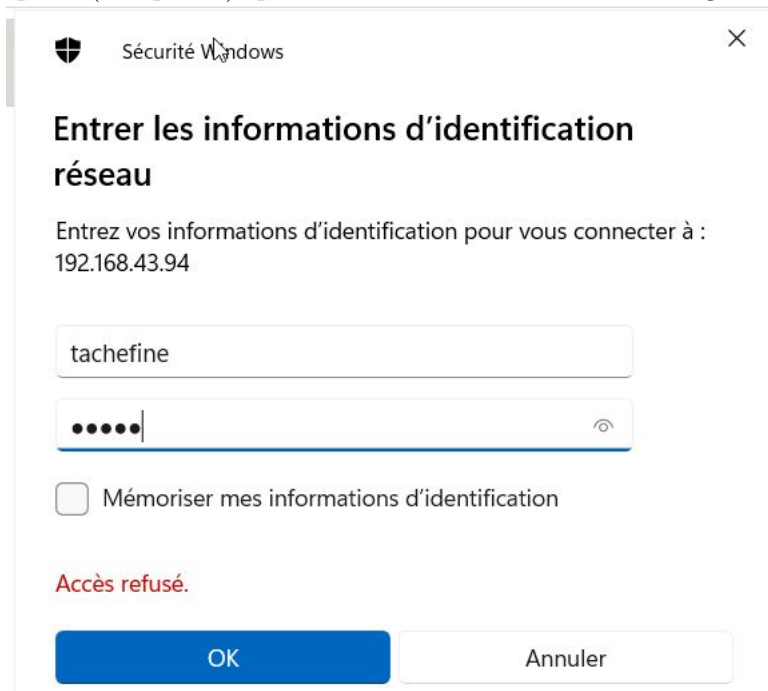
- \*écrire l'adresse de serveur au dessus dans la partie notée réseau.

Suivant ces étapes,nous avons accédé à la machine virtuelle et notre dossier "stage" créé se figure dans la machine physique.





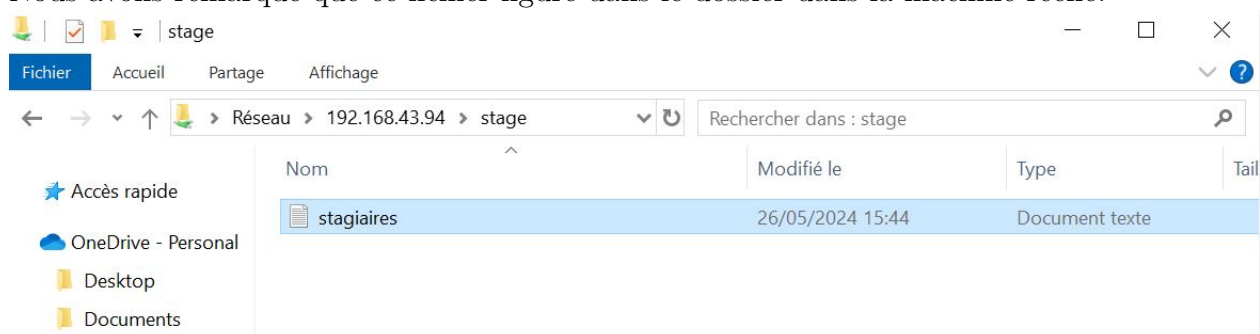
Pour accéder au dossier stage, il faut d'abord identifier le nom d'utilisateur (tachefine) et le mot de passe (5 espaces) qui ont été définis lors de la configuration de serveur:



Pour plus de précision, nous avons créé un fichier "stagiaires.txt" dans le dossier stage:

```
root@malika-VirtualBox: /cdc/stage
root@malika-VirtualBox: /cdc# cd stage/
root@malika-VirtualBox: /cdc/stage# touch stagiaires.txt
root@malika-VirtualBox: /cdc/stage# ls
stagiaires.txt
root@malika-VirtualBox: /cdc/stage#
```

Nous avons remarqué que ce fichier figure dans le dossier dans la machine réelle:



Nous avons écrit à l'intérieur de ce fichier les noms de quelques stagiaires en utilisant l'éditeur nano:

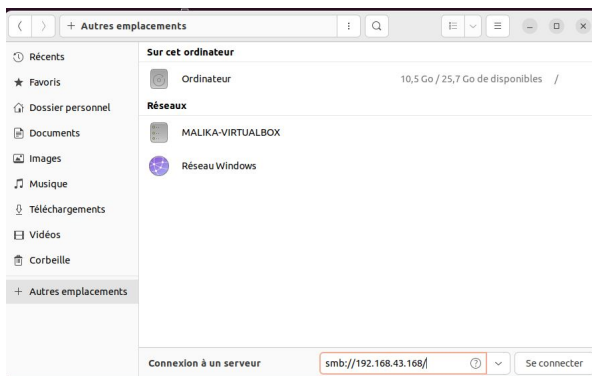
```
root@malika-VirtualBox: /cdc/stage
root@malika-VirtualBox: /cdc/stage# nano stagiaires.txt
```

le contenu du fichier dans la machine réelle:

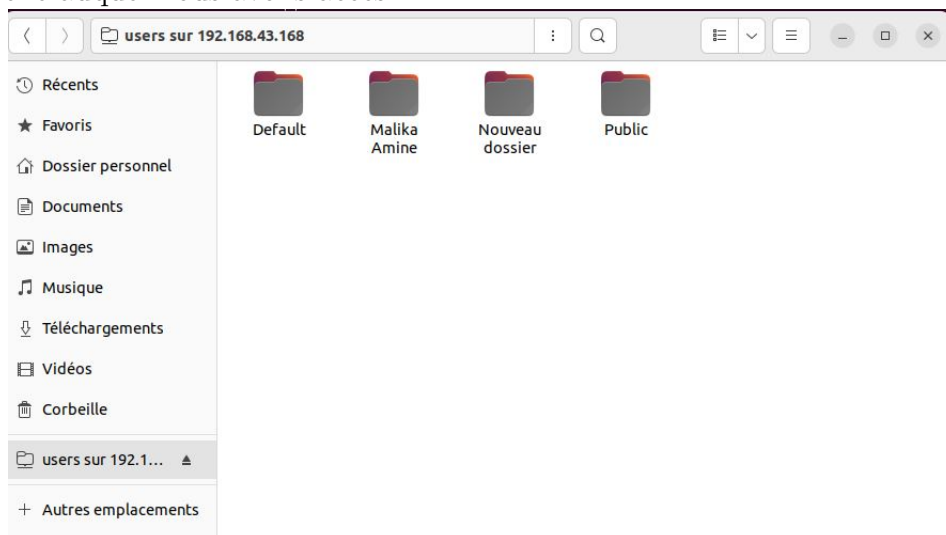


Nous pouvons aussi accéder au contenu de la machine réelle depuis notre machine virtuelle et le processus est comme suit:

- \*ouvrir le gestionnaire des fichiers de la machine virtuelle
- \*choisir autre emplacement
- \*écrire l'adresse de la machine réelle sous cette forme:smb://adresse/

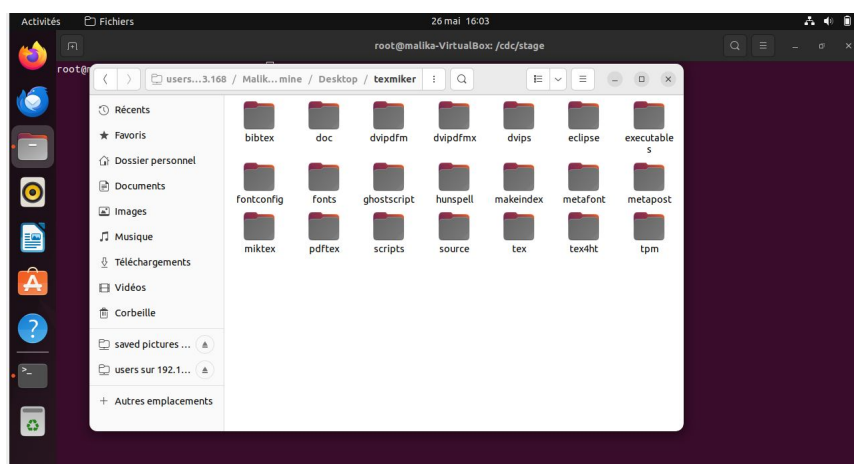
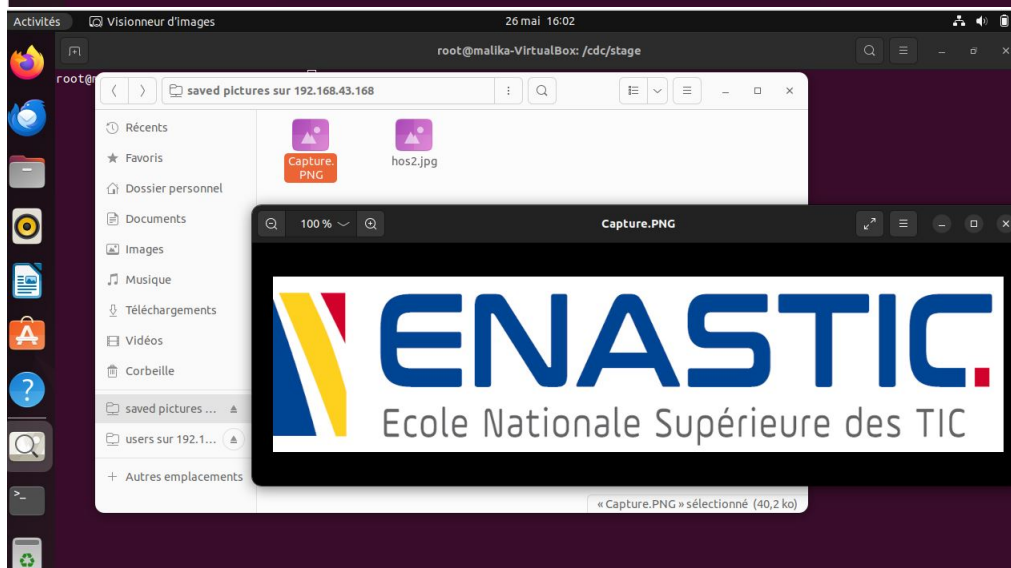
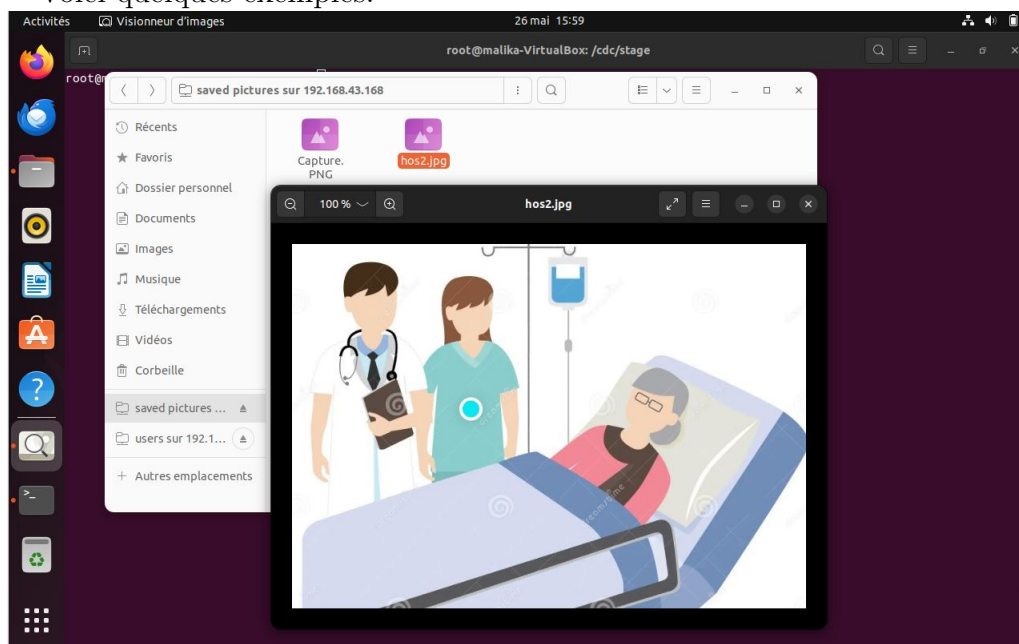


Après avoir cliqué sur le bouton se connecter,nous pouvons accéder au contenu de la machine réelle auquel nous avons accès:





Voici quelques exemples:



Le serveur Samba est un outil essentiel pour faciliter le partage de fichiers et d'imprimantes entre des systèmes Windows et des systèmes Unix-like au sein d'un réseau. En utilisant le protocole SMB/CIFS, Samba permet une interopérabilité efficace et transparente, offrant aux utilisateurs la possibilité d'accéder et de partager des ressources de manière sécurisée. La configuration de Samba peut être personnalisée pour répondre aux besoins spécifiques de chaque environnement, offrant des options avancées de contrôle d'accès et d'authentification. Grâce à sa nature open

source, Samba est largement utilisé dans les environnements professionnels et domestiques, offrant une solution fiable et flexible pour le partage de fichiers dans un réseau hétérogène.

## 0.3 Installation et configuration de serveur de téléphonie ip Asterisk

Asterisk est un logiciel open source de type PBX (Private Branch Exchange) qui permet de créer un système de téléphonie sur IP (VoIP) complet. Il a été développé par Mark Spencer de la société Digium (maintenant partie d'un groupe appelé Sangoma Technologies) et est largement utilisé dans le monde entier pour mettre en place des systèmes téléphoniques flexibles et évolutifs.

Voici quelques points clés concernant Asterisk :

1. Fonctionnalités: Asterisk offre de nombreuses fonctionnalités de téléphonie, y compris la messagerie vocale, les files d'attente, les conférences, les menus interactifs vocaux, la distribution d'appels, l'enregistrement d'appels, etc.

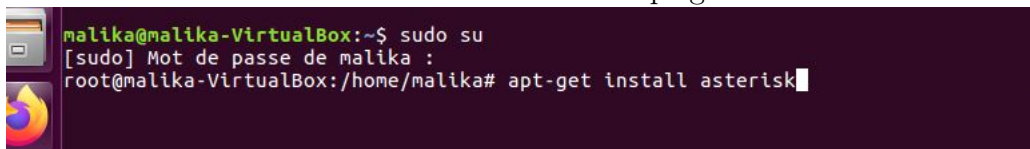
2. Compatibilité : Il est compatible avec de nombreux protocoles de communication, tels que SIP (Session Initiation Protocol), IAX (Inter-Asterisk eXchange), H.323, etc. Cela permet d'interconnecter Asterisk avec différents équipements et services de téléphonie.

3. Personnalisation: Asterisk est très flexible et personnalisable. Les utilisateurs avancés peuvent écrire des scripts et des extensions personnalisées en utilisant le langage de script Asterisk (Asterisk Dialplan) pour répondre à des besoins spécifiques.

4. Intégration : Asterisk peut être intégré avec d'autres logiciels et services, tels que des bases de données, des CRM (Customer Relationship Management), des applications web, etc., pour créer des solutions téléphoniques plus avancées et intégrées.

### 0.3.1 Installation

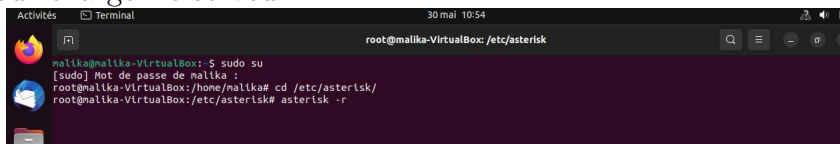
La première étape consiste à installer le serveur de téléphonie IP sur notre système Linux. Nous avons utilisé Asterisk, un logiciel open source populaire pour la téléphonie IP. Pour l'installer, nous avons utilisé la commande suivante : `sudo apt-get install asterisk`



```
malika@malika-VirtualBox:~$ sudo su
[sudo] Mot de passe de malika :
root@malika-VirtualBox:/home/malika# apt-get install asterisk
```

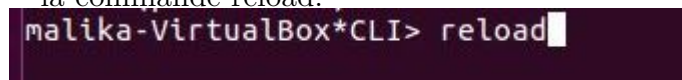
### 0.3.2 Configuration

Après l'installation, nous allons démarrer asterisk en utilisant la commande "asterisk -r" et reload pour charger le serveur :



```
root@malika-VirtualBox:/etc/asterisk
malika@malika-VirtualBox:~$ sudo su
[sudo] Mot de passe de malika :
root@malika-VirtualBox:/home/malika# cd /etc/asterisk/
root@malika-VirtualBox:/etc/asterisk# asterisk -r
```

la commande reload:

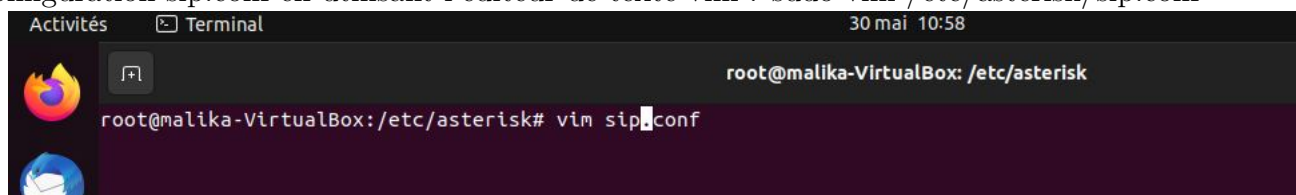


```
malika-VirtualBox*CLI> reload
```

Le statut de serveur:

```
root@malika-VirtualBox:/etc/asterisk# systemctl status asterisk
● asterisk.service - Asterisk PBX
   Loaded: loaded (/lib/systemd/system/asterisk.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2024-05-30 09:38:11 CEST; 1h 41min ago
     Docs: man:asterisk(8)
   Main PID: 705 (asterisk)
    Tasks: 70 (limit: 2254)
   Memory: 65.0M
      CPU: 1min 30.138s
   CGroup: /system.slice/asterisk.service
           └─705 /usr/sbin/asterisk -g -f -p -U asterisk
             714 astcanary /var/run/asterisk/alt.asterisk.canary.tweet.tweet.tweet 705
```

Une fois Asterisk redémarré, nous devons configurer ses paramètres pour définir les extensions, les files d'attente, les règles de routage d'appels, etc. Pour cela, nous avons édité le fichier de configuration sip.conf en utilisant l'éditeur de texte vim : `sudo vim /etc/asterisk/sip.conf`



Voici la configuration faite:


```
[sanar]
type=friend
host=dynamic
secret=123
user=sanar
context=internal

[lina]
type=friend
host=dynamic
secret=123
user=lina
context=internal

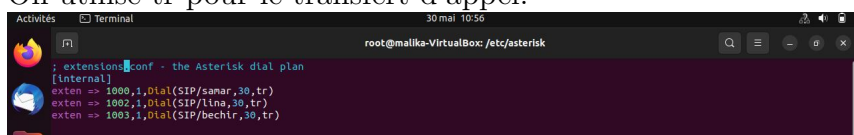
[bechir]
type=friend
host=dynamic
secret=123
user=bechir
context=internal
```

## Définition des extensions

Dans le fichier de configuration extensions.conf, nous avons défini les extensions qui permettent aux utilisateurs d'accéder au serveur de téléphonie IP et de passer des appels. Voici un exemple de configuration pour une extension utilisateur :



On utilise tr pour le transfert d'appel.



Sans aucune autre configuration, seuls les clients d'un même contexte pourront communiquer entre eux.

affichage des utilisateurs:

```
malika-VirtualBox*CLI> sip show peers
Name/Username      Host              Dyn Forcerport Comedia ACL Port  Status  Description
-----
bechir/bechir      (Unspecified)    D Auto (No) No      0        Unmonitored
lina/lina          (Unspecified)    D Auto (No) No      0        Unmonitored
sanar/sanar        (Unspecified)    D Auto (No) No      0        Unmonitored

3 sip peers [Monitored: 0 online, 0 offline Unmonitored: 0 online, 3 offline]
malika-VirtualBox*CLI>
```



## configurations des appareils

Après avoir téléchargé et installé un softphone nommé "PortSip" sur le téléphone et la machine réelle comme étant un client, nous les avons connectés sur un même réseau à travers le wifi. L'adresse de wi-fi était du classe C de même pour la machine. Ensuite, sur le téléphone, nous avons entré le nom d'utilisateur ou le user ensuite son mot de passe et l'adresse du serveur asterisk. Quant à la machine réelle, nous avons d'abord entré le domaine SIP qui est l'adresse de la machine ensuite le nom de l'extension (nom de user), mot de passe et la langue.



Domaine SIP  
192.168.170.30

SUIVANT



Numéro d'extension  
samar

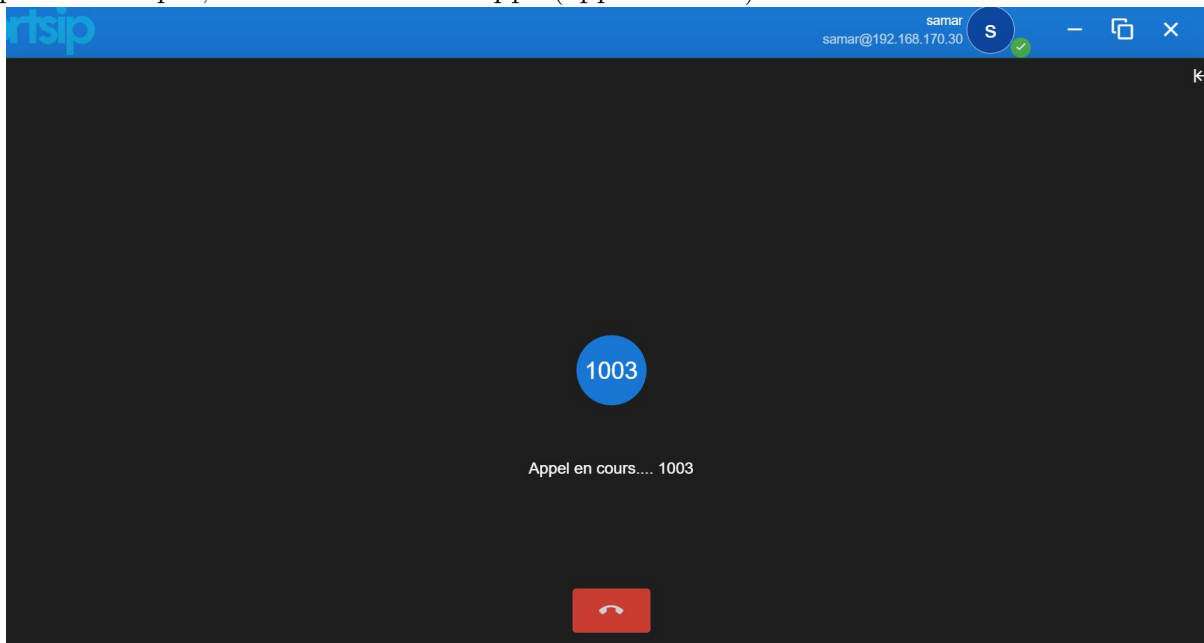
Mot de passe  
...

Langue  
Français

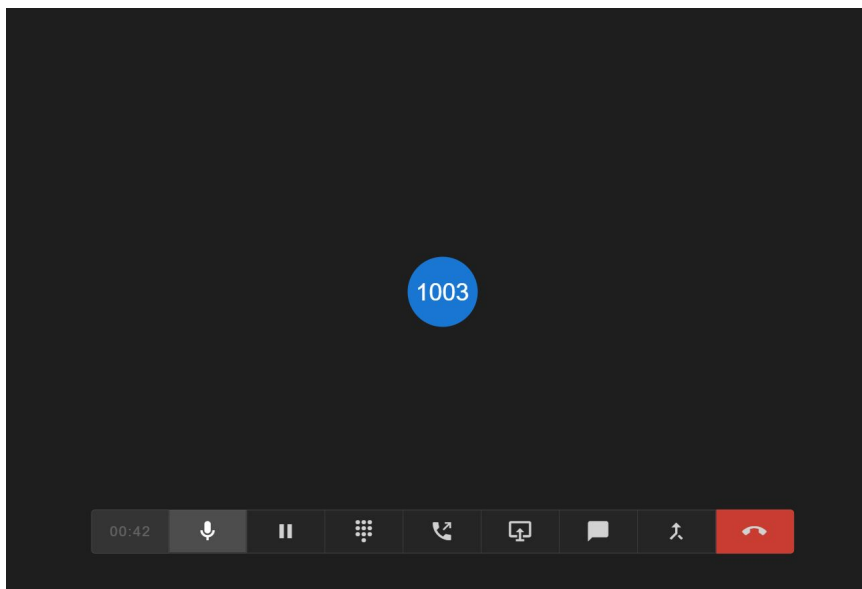
☐ Se souvenir du mot de passe  
☐ Connexion automatique

← CONNEXION

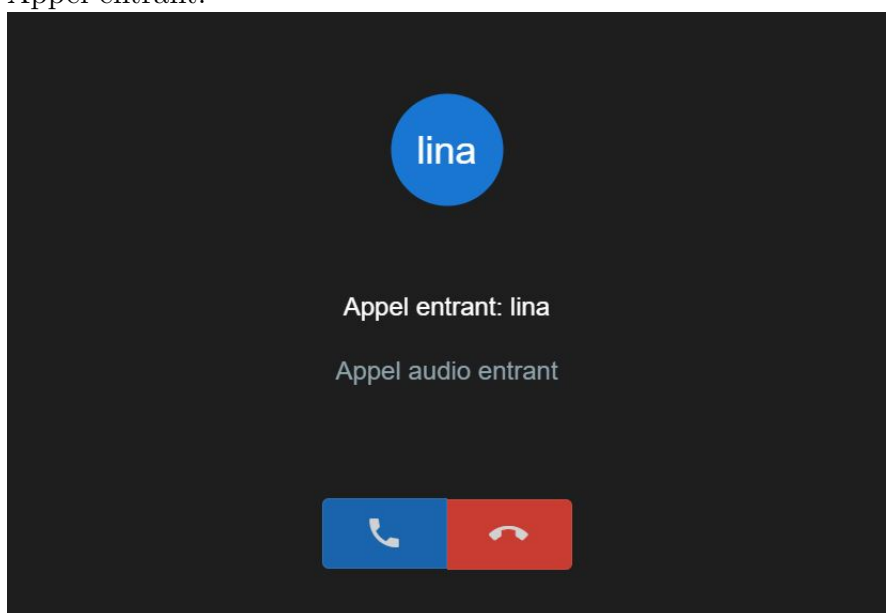
Après ces étapes,nous avons lancer l'appel(appel sortant) :



décrochage de l'appel:



Appel entrant:



historique d'appel:

portsip

samar  
samar@192.168.170.30

S

—

📄

✕

Chat

Appel

Reunion

Fonctionnal

Appel

Contacts

Historique

Lanceur d'appels

Entrer le numéro

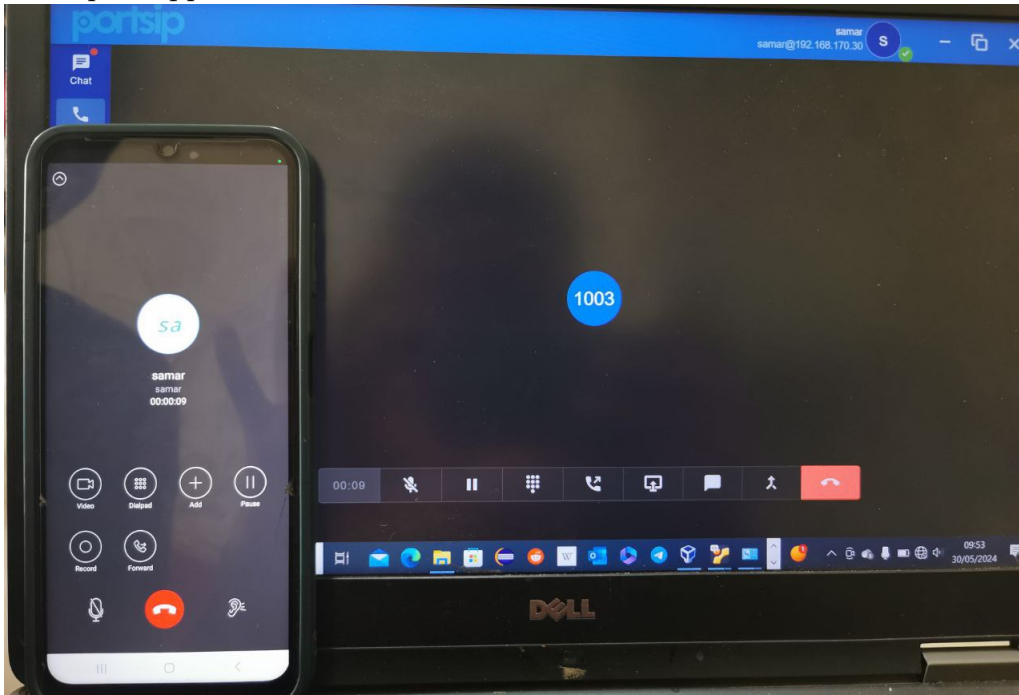
123456789\*#📞📺📠

Historique

TOUSMANQUE

Utilisateur	Nom	Extension	Type	Durée	Heure de fin	
1003		1003	En cours d'appel...	02:17	30/05/2024 09:55:56	📞📺📠
lina	lina		Nouvel Appel	03:27	30/05/2024 09:53:15	📞📺📠
1003		1003	En cours d'appel...	00:27	30/05/2024 09:48:56	📞📺📠
1003		1003	En cours d'appel...	00:54	30/05/2024 09:48:18	📞📺📠
1003		1003	En cours d'appel...	--:--	30/05/2024 09:46:50	📞📺📠

temps d'appel:



Asterisk est un logiciel de serveur de communication open source qui offre une plateforme flexible pour la mise en place de systèmes de téléphonie, de messagerie vocale, de conférence et de messagerie instantanée. Il est conçu pour être hautement personnalisable et extensible, offrant aux utilisateurs la possibilité de créer des solutions de communication sur mesure en fonction de leurs besoins spécifiques. Asterisk est largement utilisé dans les entreprises, les centres d'appels et les fournisseurs de services de télécommunications pour sa fiabilité, sa polyvalence et sa capacité à s'intégrer avec d'autres technologies.



## 0.4 Installation et configuration de serveur DHCP

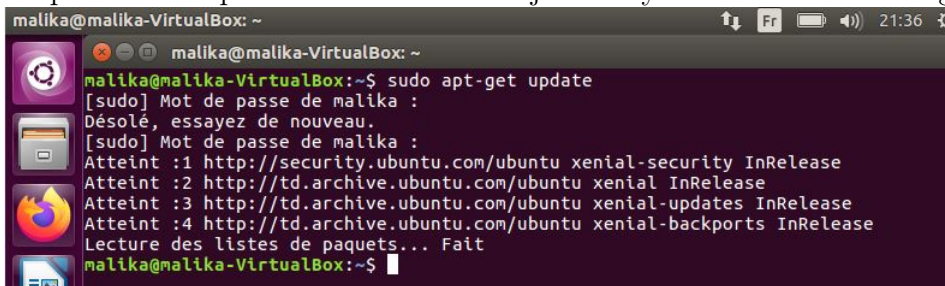
Un serveur DHCP (Dynamic Host Configuration Protocol) est un serveur utilisé dans les réseaux informatiques pour attribuer de manière dynamique des adresses IP et d'autres informations de configuration réseau aux dispositifs qui se connectent au réseau, tels que des ordinateurs, des smartphones, des imprimantes, etc.

Voici quelques points importants à connaître sur les serveurs DHCP :

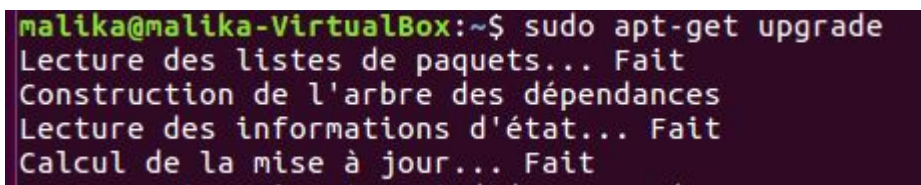
1. Attribution d'adresses IP : Le serveur DHCP attribue automatiquement des adresses IP aux clients qui se connectent au réseau, évitant ainsi les conflits d'adresses IP.
2. Gestion des adresses : Le serveur DHCP gère un pool d'adresses IP disponibles qu'il peut attribuer aux clients. Il peut également réserver des adresses IP spécifiques pour certains clients.
3. Autres informations de configuration : En plus des adresses IP, le serveur DHCP peut également fournir d'autres informations de configuration réseau aux clients, telles que l'adresse du serveur DNS, la passerelle par défaut, etc.

### 0.4.1 Installation

La première étape consiste à mettre à jour le système à travers les lignes qui suivent:

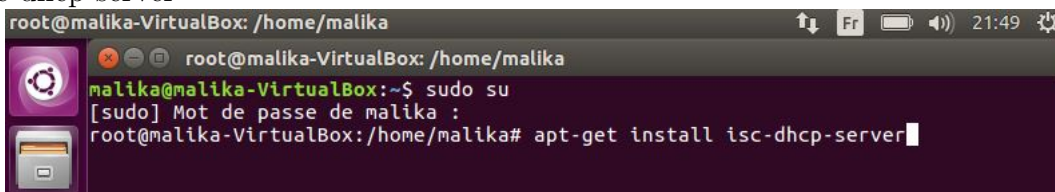


```
malika@malika-VirtualBox: ~  
malika@malika-VirtualBox:~$ sudo apt-get update  
[sudo] Mot de passe de malika :  
Désolé, essayez de nouveau.  
[sudo] Mot de passe de malika :  
Atteint :1 http://security.ubuntu.com/ubuntu xenial-security InRelease  
Atteint :2 http://td.archive.ubuntu.com/ubuntu xenial InRelease  
Atteint :3 http://td.archive.ubuntu.com/ubuntu xenial-updates InRelease  
Atteint :4 http://td.archive.ubuntu.com/ubuntu xenial-backports InRelease  
Lecture des listes de paquets... Fait  
malika@malika-VirtualBox:~$
```



```
malika@malika-VirtualBox:~$ sudo apt-get upgrade  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances  
Lecture des informations d'état... Fait  
Calcul de la mise à jour... Fait
```

La seconde consiste à mettre à jour le sye étape consiste à installer le serveur DHCP sur notre système Linux. Pour cela, nous avons utilisé la commande suivante : `sudo apt-get install isc-dhcp-server`

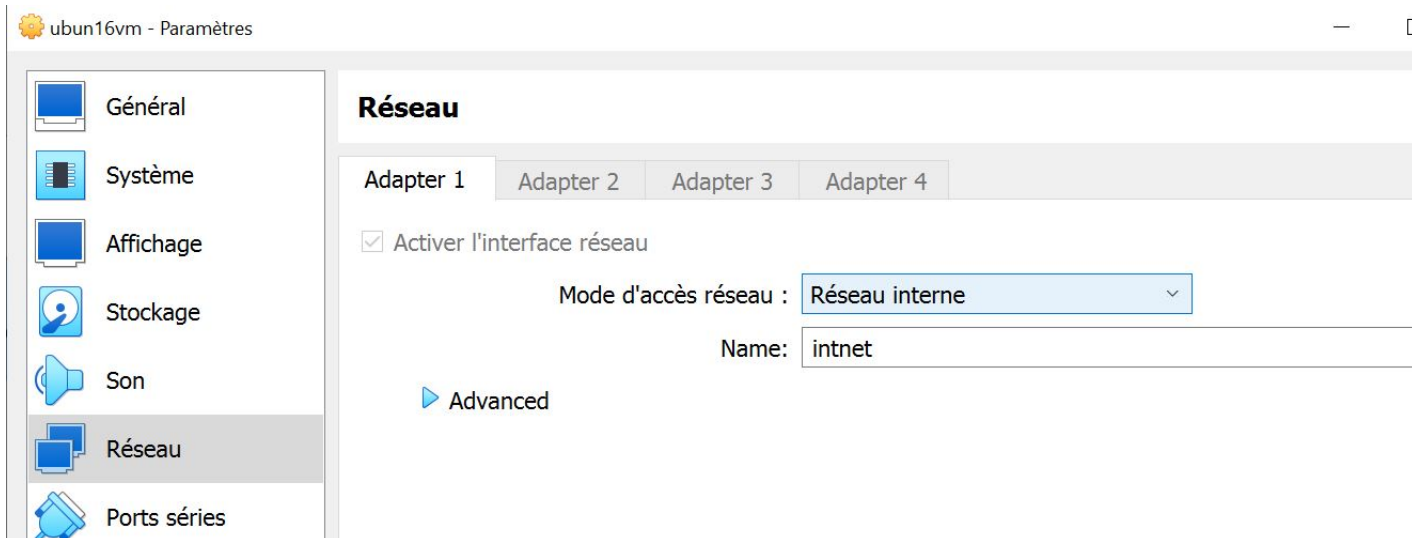


```
root@malika-VirtualBox: /home/malika  
malika@malika-VirtualBox:~$ sudo su  
[sudo] Mot de passe de malika :  
root@malika-VirtualBox:/home/malika# apt-get install isc-dhcp-server
```

Pour pouvoir finir l'installation, un message sous forme d'interrogation s'affiche pour finaliser ce processus et dont on doit répondre par O ou Y et cliquer sur la touche entrée:

```
root@malika-VirtualBox: /home/malika
root@malika-VirtualBox: /home/malika
malika@malika-VirtualBox:~$ sudo su
[sudo] Mot de passe de malika :
root@malika-VirtualBox:/home/malika# apt-get install isc-dhcp-server
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  libirs-export141 libiscfg-export140
Paquets suggérés :
  isc-dhcp-server-ldap policycoreutils
Les NOUVEAUX paquets suivants seront installés :
  isc-dhcp-server libirs-export141 libiscfg-export140
0 mis à jour, 3 nouvellement installés, 0 à enlever et 10 non mis à jour.
Il est nécessaire de prendre 470 ko dans les archives.
Après cette opération, 1 587 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] o
```

Après ces étapes, nous avons mis le réseau de la machine sur "réseau interne" dans sa partie configuration:



Ensuite, définir l'interface à travers: "nano /etc/default/isc-dhcp-server):

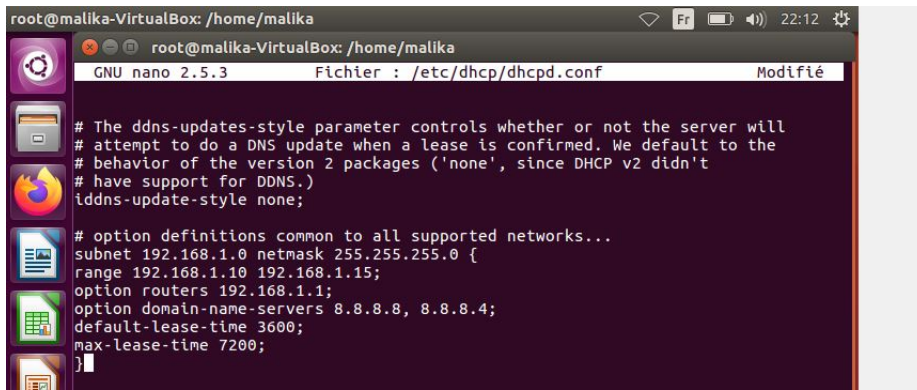
```
Activités Terminal 26 mai 18:21
root@malika-VirtualBox: /home/malika
GNU nano 6.2 /etc/default/isc-dhcp-server
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)
#
# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPD_CONF=/etc/dhcp/dhcpd.conf
#DHCPD_O_CONF=/etc/dhcp/dhcpd.conf
#
# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPD_PID=/var/run/dhcpd.pid
#DHCPD_O_PID=/var/run/dhcpd.pid
#
# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""
#
# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="enp0s3"
INTERFACESv6=""
```

## 0.4.2 Configuration

Une fois le serveur DHCP installé, nous devons configurer ses paramètres pour définir les plages d'adresses IP disponibles et les options de configuration réseau. Pour cela, nous avons édité le fichier de configuration dhcpd.conf en utilisant l'éditeur de texte Nano : nano /etc/dhcp/dhcpd.conf

```
root@malika-VirtualBox: /home/malika
root@malika-VirtualBox: /home/malika
malika@malika-VirtualBox:~$ sudo su
[sudo] Mot de passe de malika :
root@malika-VirtualBox:/home/malika# nano /etc/dhcp/dhcpd.conf
```

le fichier de configuration est:



```
root@malika-VirtualBox: /home/malika
GNU nano 2.5.3 Fichier : /etc/dhcp/dhcpd.conf Modifié

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
# have support for DDNS.)
ddns-update-style none;

# option definitions common to all supported networks...
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.10 192.168.1.15;
  option routers 192.168.1.1;
  option domain-name-servers 8.8.8.8, 8.8.4.4;
  default-lease-time 3600;
  max-lease-time 7200;
}
```

Après avoir édité, nous avons fait un "CTRL+X" puis cliqué O et ensuite sur la touche entrée pour enregistrer et sortir du fichier.

\* 'subnet 192.168.1.0 netmask 255.255.255.0' : Définit le réseau et le masque de sous-réseau.

\* 'range 192.168.1.100 192.168.1.200' : Définit la plage d'adresses IP à attribuer.

\* 'option routers 192.168.1.1' : Définit le routeur par défaut pour les clients.

\* 'option domain-name-servers 8.8.8.8, 8.8.4.4' : Définit les serveurs DNS pour les clients.

### 0.4.3 Configuration des options spécifiques

Nous pouvons configurer des options spécifiques pour les clients, telles que des options de réservation d'adresse IP. Par exemple, pour réserver l'adresse IP '192.168.1.10' pour le client avec l'adresse MAC '00:11:22:33:44:55' :

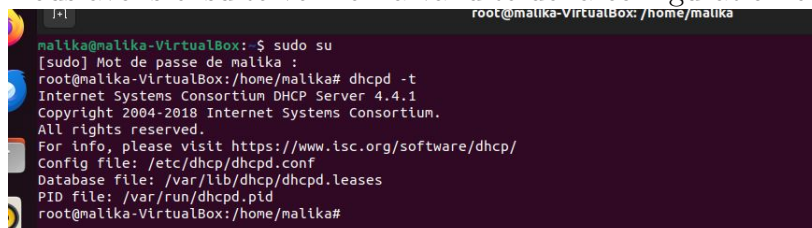
host client-name

hardware

ethernet 00:11:22:33:44:55;

fixed-address 192.168.1.10;

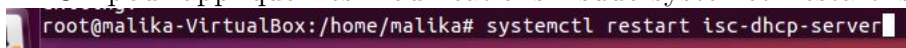
Nous avons ensuite vérifié la validité de la configuration en utilisant la commande : dhcp-t



```
root@malika-VirtualBox: /home/malika
malika@malika-VirtualBox:~$ sudo su
[sudo] Mot de passe de malika :
root@malika-VirtualBox: /home/malika# dhcpd -t
Internet Systems Consortium DHCP Server 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Config file: /etc/dhcp/dhcpd.conf
Database file: /var/lib/dhcp/dhcpd.leases
PID file: /var/run/dhcpd.pid
root@malika-VirtualBox: /home/malika#
```

### redémarrage de serveur DHCP

Après avoir enregistré et vérifié la validité de la configuration, nous avons redémarré le service DHCP pour appliquer les modifications : `sudo systemctl restart isc-dhcp-server`



```
root@malika-VirtualBox: /home/malika# systemctl restart isc-dhcp-server
```

Ensuite, nous avons activé le service DHCP pour qu'il démarre automatiquement au démarrage du système : `sudo systemctl enable isc-dhcp-server`

nous pouvons aussi vérifier le journal des événements : Vérifiez le journal des événements pour toute erreur : `sudo journalctl -u isc-dhcp-server`

En somme le serveur DHCP (Dynamic Host Configuration Protocol) joue un rôle essentiel dans la gestion efficace des adresses IP au sein d'un réseau informatique en attribuant automatiquement des adresses IP aux appareils connectés, simplifiant ainsi la configuration et la maintenance du réseau. Grâce à sa capacité à allouer dynamiquement des adresses IP, des passerelles et d'autres paramètres réseau aux périphériques, le serveur DHCP permet d'optimiser l'utilisation des ressources du réseau, d'améliorer la connectivité et de réduire les tâches administratives manuelles, offrant ainsi une solution pratique et scalable pour les administrateurs réseau.

## 0.5 Conclusion

En conclusion, l'installation et la configuration des serveurs de téléphonie IP, Samba et DHCP ont été menées à bien avec succès. Ce projet a permis d'établir une infrastructure informatique robuste et fiable pour répondre aux besoins de communication et de partage de ressources au sein de l'organisation.

La mise en place du serveur de téléphonie IP a amélioré l'efficacité des communications internes et externes, offrant des fonctionnalités avancées telles que la gestion des appels, la messagerie vocale et la mobilité des utilisateurs.

Le déploiement du serveur Samba a facilité le partage de fichiers et d'imprimantes au sein du réseau, permettant une collaboration plus efficace entre les différents départements de l'entreprise.

Enfin, la configuration du serveur DHCP a simplifié la gestion des adresses IP au sein du réseau, garantissant une allocation efficace des ressources réseau et facilitant l'ajout de nouveaux périphériques.

Dans l'ensemble, ce projet a permis d'améliorer l'efficacité opérationnelle de l'organisation et de renforcer sa capacité à répondre aux besoins de communication et de partage de ressources de manière sécurisée et évolutive.