

---

# UNIVERSITÉ CHEIKH ANTA DIOP DE DAKAR

---

## École Supérieure Polytechnique

---



ECOLE SUPERIEURE  
POLYTECHNIQUE

---

## Rapport Unifié : Solutions Réseaux Avancées

---

Présenté par :

Salif BIAYE

Ndeye Astou DIAGOURAGA

Sous la direction de :

Dr Keba

*Enseignant*

---

Année universitaire 2024-2025

---



## Table des Matières

---

- [\*\*1. Rapport Kerberos, LDAP et DNS\*\*](#)

---
- [\*\*2. Protocoles d'Accès et Services de Partage\*\*](#)

---
- [\*\*3. Rapport Asterisk et LDAP \(BONUS VIDEO\)\*\*](#)

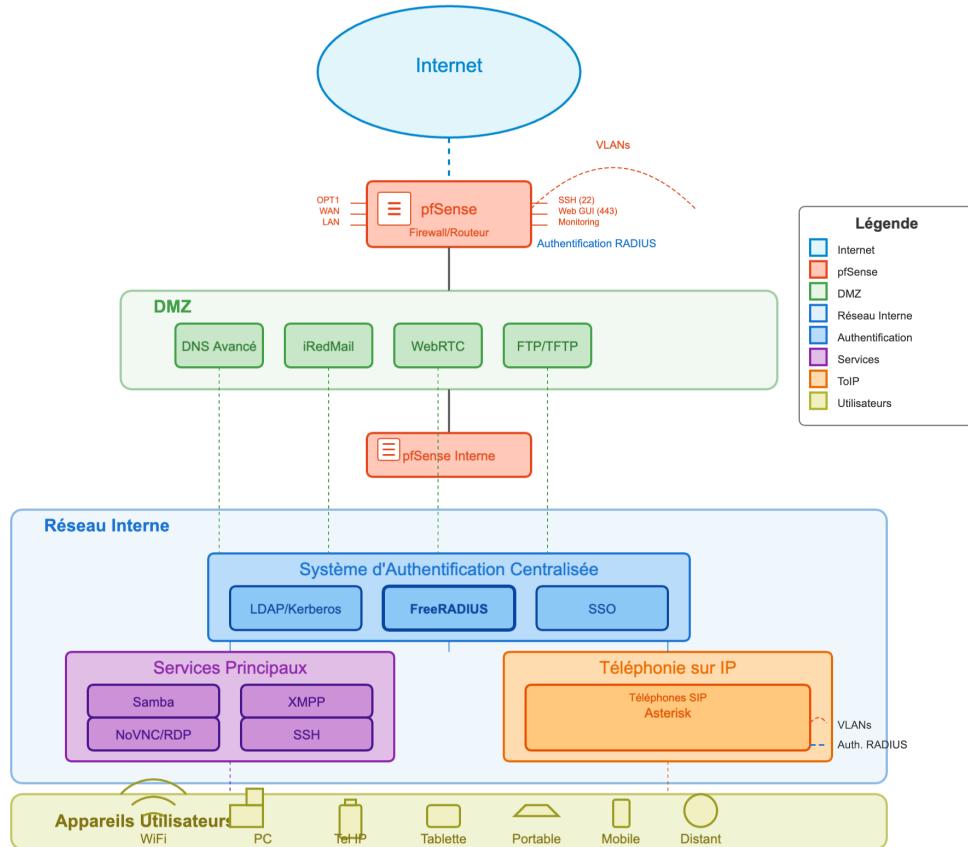
---
- [\*\*4. Rapport iRedMail et LDAP \(BONUS VIDEO\)\*\*](#)

---
- [\*\*5. Rapport WebRTC avec Jitsi Meet \(BONUS VIDEO\)\*\*](#)

---
- [\*\*6. Rapport PfSense et RADIUS \(BONUS VIDEO\)\*\*](#)

---

## Architecture Réseau Centralisée avec ToIP - smarttech



# UNIVERSITÉ CHEIKH ANTA DIOP DE DAKAR

École Supérieure Polytechnique



ECOLE SUPERIEURE  
POLYTECHNIQUE

## Configuration DNS et Couplage Kerberos/LDAP

Présenté par :

Salif BIAYE

Ndeye Astou DIAGOURAGA

Sous la direction de :

Dr Keba

*Enseignant*

Année universitaire 2024-2025

\*\*

# Table des Matières

---

## I. Configuration DNS

---

### I.1. Introduction

---

### I.2. Configuration Serveur

---

### I.3. Tests et Validation

---

## II. Couplage Kerberos/LDAP

---

### II.1. Configuration LDAP

---

### II.2. Intégration Kerberos

---

### II.3. Tests d'Intégration

---

## III. Conclusion

---

# I. Configuration du Serveur DNS

## I.1. Introduction

Ce rapport présente les étapes pour configurer un serveur DNS (Domain Name System) pour le domaine [smarttech.sn](#). La configuration du DNS est cruciale pour permettre la résolution des noms de domaine vers les adresses IP, facilitant ainsi l'accès aux services associés à ce domaine.



Avant de commencer la configuration du serveur DNS, il est essentiel de remplir les pré-requis suivants :

- Disposer d'un serveur DNS fonctionnel (par exemple, [BIND](#) sous Linux).
- Un domaine enregistré ([smarttech.sn](#)).
- Des droits administratifs sur le serveur DNS.
- Un fichier de zone pour le domaine [smarttech.sn](#).

## I.2. Configuration du Serveur

💡 Pour installer bind (le serveur dns le plus couramment utilisé), on exécute les commandes suivantes sur un système ubuntu :

```
sudo apt update  
sudo apt install bind9 bind9utils bind9-doc
```

Le fichier [named.conf](#) est utilisé pour configurer BIND et définir les zones de DNS. Ce fichier se trouve généralement dans le répertoire [/etc/bind/](#). Ajoutez la configuration suivante pour inclure la zone [smarttech.sn](#) :

```
zone "smarttech.sn" {  
    type master;  
    file "/etc/bind/db.smarttech.sn";  
};
```

Le fichier de zone contient les enregistrements DNS pour le domaine. On crée un fichier [/etc/bind/db.smarttech.sn](#) avec le contenu suivant :

```

TTL    86400
@      **IN**      **SOA**      kdc.smarttech.sn. admin.smarttech.sn. (
                           20250303 ; Serial           3600
@      **IN**      **NS**       kdc.smarttech.sn.kdc      **IN**      **A**
                                        

www    **IN**      **A**        192.168.1.200
mail   **IN**      **A**        192.168.1.201 ; Enregistrement A pour mail

@      IN      MX      10 mail.smarttech.sn. ; Enregistrement MX pour le

```

- **SOA (Start of Authority)** : Indique les informations de base sur la zone.
- **NS (Name Server)** : Définit les serveurs DNS autoritaires pour le domaine.
- **A (Address)** : Associe des noms d'hôtes à des adresses IP.
- **MX (Mail Exchanger)** : Pour la gestion des emails.

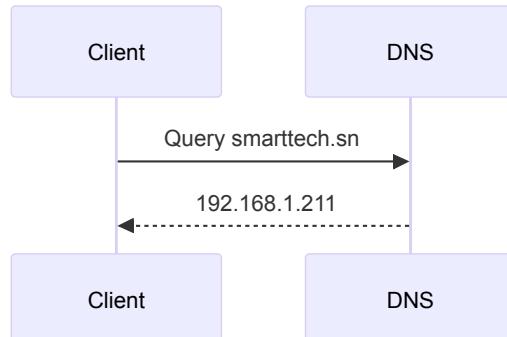
### I.3. Validation DNS

Une fois les configurations effectuées, redémarrez le service BIND pour appliquer les modifications :

```
sudo systemctl restart bind9
```

Vérifiez que le serveur DNS fonctionne correctement en utilisant des outils comme [dig](#) ou [nslookup](#) :

```
dig @localhost smarttech.sn
nslookup mail.smarttech.sn
```



 On obtient ce qui suit:

```
root@server:/home/server# dig @localhost smarttech.sn

; <>> DiG 9.18.30-0ubuntu0.22.04.2-Ubuntu <>> @localhost smarttech.sn
; (1 server found)

;; global options: +cmd
;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1894
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: udp: 1232
; COOKIE: 2fcc35a089dbda54010000067c5b8ca05c24d5c69af507 (good)

;; QUESTION SECTION:
;smarttech.sn.           IN      A

;; AUTHORITY SECTION:
smarttech.sn.       604800  IN      SOA      kdc.smarttech.sn. admin.sma

;; Query time: 274 msec
;; SERVER: 127.0.0.1#53(localhost) (UDP)
;; WHEN: Mon Mar 03 14:12:26 GMT 2025
```

 On peut aussi faire un nslookup pour vérifier

```
root@server:/home/server# nslookup kdc.smarttech.sn
Server:          127.0.0.53
Address:        127.0.0.53#53

Name:  kdc.smarttech.sn
Address: 192.168.1.211

Name:  kdc.smarttech.sn
Address: fd00::ae77:529b:c27d:59db

Name:  kdc.smarttech.sn
Address: fd00::eb47:b1a2:2d20:e082

Name:  kdc.smarttech.sn
Address: fe80::989c:7743:7e28:7163

root@server:/home/server#
```

```
root@server:/home/server# named-checkzone smarttech.sn /etc/bind/db.smarttech
zone smarttech.sn/IN: loaded serial 20250303
OK
```

La configuration du serveur DNS pour le domaine **smarttech.sn** assure une résolution correcte des noms de domaine et une gestion efficace des services réseau. Cependant, pour renforcer la sécurité, il est essentiel d'intégrer **Kerberos**, un protocole d'authentification centralisée. Tandis que le DNS garantit l'accès aux ressources, **Kerberos** assure que seules les entités authentifiées peuvent y accéder. Cette transition permet de sécuriser les communications et d'assurer un contrôle d'accès solide au sein de notre réseau.

## II. Intégration Kerberos/LDAP

### 💡 Introduction

Kerberos et LDAP sont deux technologies essentielles dans les environnements d'entreprise pour assurer une authentification centralisée et sécurisée. LDAP (Lightweight Directory Access Protocol) est un protocole utilisé pour accéder et gérer un annuaire d'utilisateurs, tandis que Kerberos est un protocole d'authentification sécurisé basé sur un système de tickets.

Le couplage de Kerberos et LDAP permet d'utiliser LDAP comme base d'annuaire centralisée et d'exploiter Kerberos pour authentifier nos utilisateurs de manière sécurisée.

### 💡 Objectif du couplage

L'intégration de Kerberos avec LDAP vise à :

- Centraliser la gestion des utilisateurs et des mots de passe.
- Sécuriser l'authentification avec Kerberos.
- Faciliter l'administration des accès réseau.
- Permettre l'authentification unique (SSO - Single Sign-On).

### 💡 Prerequisites

Avant de procéder à l'installation et à la configuration, il est nécessaire de disposer :

- D'un serveur Linux (Ubuntu dans notre cas).
- Des paquets krb5-kdc, krb5-admin-server, krb5-user pour Kerberos.
- D'un serveur LDAP fonctionnel (ex : OpenLDAP).
- Des paquets libnss-ldap, libpam-krb5, krb5-config, krb5-user pour l'intégration LDAP-Kerberos.

### 💡 Remarque:

- notre nom de domaine est smarttech.sn
- l'adresse IP de notre serveur est 192.168.1.211
- le hostname de notre machine kdc.smarttech.sn

### II.1. Configuration LDAP

Nous allons installer le serveur OpenLDAP sur le même hôte que le KDC, afin de simplifier la communication entre eux.

#### 💡 Installation des paquets nécessaires

```
sudo apt install krb5-kdc-ldap krb5-admin-server
```

#### 💡 Extraction du fichier kerberos.schema.gz

kerberos.schema.gz contient la définition des objets et attributs nécessaires pour stocker des informations Kerberos dans un annuaire LDAP.

```
sudo cp /usr/share/doc/krb5-kdc-ldap/kerberos.schema.gz /etc/ldap/schema/
sudo gunzip /etc/ldap/schema/kerberos.schema.gz
```

#### 🔗 Ajout du schema kerberos dans l'arborescence

Le fichier schema doit etre converti au format ldif avant de pouvoir etre ajoute. Pour cela on installe:

```
sudo apt install schema2ldif
```

#### 🔗 Pour importer le schéma kerberos, on execute:

```
$ sudo ldap-schema-manager -i kerberos.schema
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0executing 'ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0**adding** new entry "cn=kerberos,cn=schema,cn=config"
```

#### 🔗 Indexons un attribut souvent utilisé dans les recherches

```
$ sudo ldapmodify -Q -Y EXTERNAL -H ldapi:/// <<EOF
dn: olcDatabase={1}mdb,cn=config
add: olcDbIndex
olcDbIndex: krbPrincipalName eq,pres,sub
EOF
modifying entry "olcDatabase={1}mdb,cn=config"
```

#### 🔗 Creation des entrees ldap pour les entrees administratives kerberos

```
***$ ldapadd -x -D cn=admin,dc=smarttech,dc=sn -W <<EOF
dn: uid=kdc-service,dc=smarttech,dc=sn
uid: kdc-service
objectClass: account
objectClass: simpleSecurityObject
userPassword: {CRYPT}x
description: Account used for the Kerberos KDC

dn: uid=kadmin-service,dc=smarttech,dc=sn
uid: kadmin-service
objectClass: account
objectClass: simpleSecurityObject
userPassword: {CRYPT}x
description: Account used for the Kerberos Admin server
EOF
Enter LDAP Password:
adding new entry "uid=kdc-service,dc=smarttech,dc=sn"

adding new entry "uid=kadmin-service,dc=smarttech,dc=sn"**
```

#### 🔗 On va ensuite devenir un mot de passe pour chaque entree: kdc service et kadmin service

```
$ ldappasswd -x -D cn=admin,dc=smarttech,dc=sn -W -S uid=kdc-service,dc=sm  
**New** password:*****Re-enter **new** password:*****  
Enter LDAP Password: *****
```

#### 🔗 Mise à jour des listes de contrôle d'accès acl

```
**$ sudo ldapmodify -Q -Y EXTERNAL -H ldapi:/// <<EOF  
dn: olcDatabase={1}mdb,cn=config  
add: olcAccess  
olcAccess: {2}to attrs=krbPrincipalKey by anonymous auth by dn.exact="ui  
-  
add: olcAccess  
olcAccess: {3}to dn.subtree="cn=krbContainer,dc=smarttech,dc=sn"  
by dn.exact="uid=kdc-service,dc=smarttech,dc=sn" read by dn.exact="uid=  
EOF  
  
modifying entry "olcDatabase={1}mdb,cn=config"**
```

Notre annuaire LDAP est maintenant prêt à servir de base de données principale Kerberos.

#### LDAP

dc=smarttech  
dc=sn  
ou=Users  
krbPrincipalName

## II.2. Configuration Kerberos

#### 🔗 Editons le fichier /etc/krb5.conf

```
**[realms]  
EXAMPLE.COM = {  
    kdc = kdc.smarttech.sn  
    admin_server = kdc.smarttech.sn  
    default_domain = smarttech.sn  
    database_module = openldap_ldapconf  
}
```

```
[dbdefaults]
ldap_kerberos_container_dn = cn=krbContainer,dc=smarttech,dc=sn
[dbmodules]
openldap_ldapconf = {
    db_library = kldap
    disable_last_success = true
    disable_lockout = true
    ldap_kdc_dn = "uid=kdc-service,dc=smarttech,dc=sn"
    ldap_kadmind_dn = "uid=kadmin-service,dc=smarttech,dc=sn"
    ldap_service_password_file = /etc/krb5kdc/service.keyfile
    ldap_servers = ldapi:///
    ldap_conns_per_server = 5
}
```

#### 🔗 Creer le domaine avec kdb5\_ldap\_util

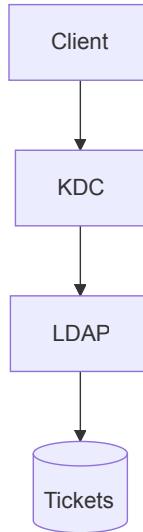
```
$ sudo ldap-schema-manager -i kerberos.schema
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0executing 'ldapadd -Y EXTERNAL -H ldapi:///
-f /etc/ldap/schema/kerberos.ldif'
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0**adding** new entry "cn=kerberos,cn=schema,cn=config"
```

#### 🔗 Creons les mots de passe pour chacun

```
sudo kdb5_ldap_util -D cn=admin,dc=smarttech,dc=sn stashsrwpw
-f /etc/krb5kdc/service.keyfile uid=kdc-
service,dc=smarttech,dc=sn
sudo kdb5_ldap_util -D cn=admin,dc=smarttech,dc=sn stashsrwpw
-f /etc/krb5kdc/service.keyfile uid=kadmin-
service,dc=smarttech,dc=sn**
```

#### 🔗 Redémarrage des services

```
sudo systemctl start krb5-kdc.service
sudo systemctl start krb5-admin-server.service
```



### II.3. Tests d'Intégration

Testons l'authentification kerberos LDAP avec Kerberos  
Créons un utilisateur dans kerberos avec

```
sudo kadmin.local -q "addprinc salif@SMARTTECH.SN"
```

Vérifions si le principal est dans ldap avec la commande:

```
ldapsearch -x -D "cn=admin,dc=smarttech,dc=sn" -W
-b "dc=smarttech,dc=sn"
```

On peut voir avec la capture suivante que l'utilisateur salif a été ajouté dans ldap

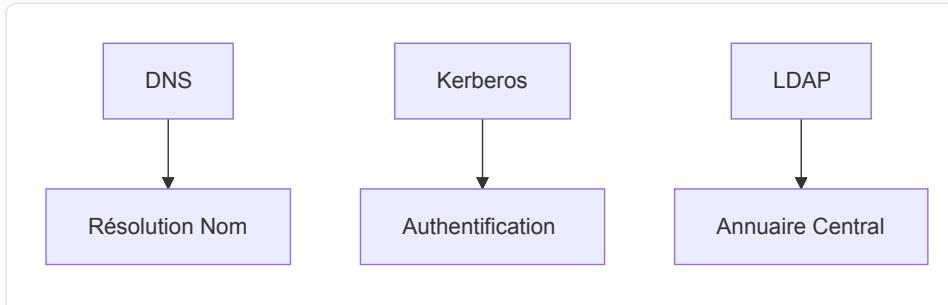
```
# salif@SMARTTECH.SN, SMARTTECH.SN, krbContainer, smarttech.sn
dn: krbPrincipalName=salif@SMARTTECH.SN,cn=SMARTTECH.SN,cn=krbContainer,dc
krbLoginFailedCount: 0
krbPrincipalName: salif@SMARTTECH.SN
krbPrincipalKey:: MIG20AMCAQGhAwIBAaIDAgEBowMCAQQkgZ8wgZwwVKAHMAwGAwIBAKFJ
krbLastPwdChange: 20250303084201Z
krbExtraData:: AAJZa8Vncm9vdC9hZG1pbkBTUFSVFRFQ0gUU04A
krbExtraData:: AAgBAA==
objectClass: krbPrincipal
objectClass: krbPrincipalAux
objectClass: krbTicketPolicyAux

# host/kdc.smarttech.sn@SMARTTECH.SN, SMARTTECH.SN, krbContainer, smarttec
```

### III. Conclusion

L'intégration de **Kerberos** avec **LDAP** offre une solution robuste pour la gestion centralisée des identités et l'authentification sécurisée des utilisateurs. En exploitant **LDAP** comme annuaire de stockage des identités et **Kerberos** comme mécanisme d'authentification, cette architecture permet de garantir une **sécurité renforcée**, une **administration simplifiée** et une **expérience utilisateur améliorée** grâce au Single Sign-On (**SSO**).

Cette architecture combinant DNS, Kerberos et LDAP permet :



- Gestion centralisée des identités
- Sécurité renforcée avec tickets Kerberos
- Single Sign-On (SSO)

# UNIVERSITÉ CHEIKH ANTA DIOP DE DAKAR

École Supérieure Polytechnique



ECOLE SUPERIEURE  
POLYTECHNIQUE

## Protocoles d'Accès et Services de Partage

Présenté par :

Salif BIAYE

Ndeye Astou DIAGOURAGA

Sous la direction de :

Dr Keba

Enseignant

Année universitaire 2024-2025

\*\*

# Table des Matières

---

## Introduction

---

## Kerberos et SSH

---

### Introduction à Kerberos

---

### Intégration avec SSH

---

### Préparation du serveur et du client

---

### Configuration du serveur SSH

---

### Configuration du client SSH

---

### Test de la connexion

---

## Protocoles d'Accès Bureau à Distance

---

### IRDP (Remote Desktop Protocol)

---

### INoVNC

---

## Samba avec LDAP et Kerberos

---

### Introduction

---

### Installation des paquets

---

### Configuration de LDAP

---

### Configuration de Kerberos pour Samba

---

### Configuration de Samba

---

### Ajout des utilisateurs Samba

---

### Tests et validation

---

## Services de Transfert de Fichiers

---

## Conclusion

---

# Introduction

Ce rapport présente l'implémentation de plusieurs protocoles d'accès et services de partage sécurisés dans un environnement d'entreprise. Dans un contexte de sécurité informatique en constante évolution, la mise en place de solutions d'authentification centralisées et de connexions distantes sécurisées est devenue indispensable.

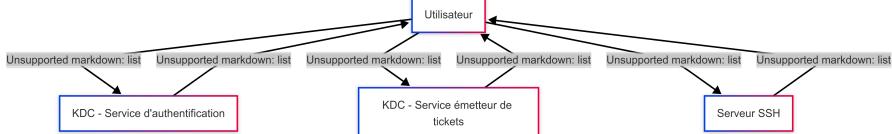
Nous aborderons l'intégration de Kerberos avec SSH pour l'authentification centralisée, les solutions d'accès bureau à distance avec RDP et NoVNC, la configuration de Samba avec LDAP et Kerberos, ainsi que les services de transfert de fichiers TFTP et SFTP.

L'objectif principal est de mettre en place une infrastructure sécurisée et efficace pour la gestion à distance des systèmes, en combinant des technologies complémentaires pour répondre aux besoins spécifiques de l'entreprise SmartTech.SN.

## Kerberos et SSH

### Introduction à Kerberos

L'intégration de Kerberos avec SSH permet d'optimiser la sécurité des connexions à distance sur des serveurs Linux/Unix en centralisant l'authentification des utilisateurs.



### Intégration avec SSH

SSH (Secure Shell) est un protocole utilisé pour accéder à distance à des systèmes Unix/Linux de manière sécurisée. Lorsqu'il est couplé avec Kerberos, SSH bénéficie d'une authentification renforcée, permettant une gestion centralisée et simplifiée des identités et des accès.

### Avantages du couplage Kerberos avec SSH

- Authentification centralisée** : Les utilisateurs sont authentifiés via un serveur Kerberos, tel qu'un KDC (Key Distribution Center), permettant une gestion centralisée des identités.
- Sécurité améliorée** : Kerberos utilise des tickets cryptographiques pour l'authentification, ce qui protège contre les attaques par interception de mot de passe.
- Accès sans mot de passe** : Une fois l'utilisateur authentifié par Kerberos, il n'a pas à saisir son mot de passe à chaque connexion SSH, ce qui facilite l'accès sécurisé à plusieurs serveurs.

```
sudo apt-get install krb5-user libpam-krb5
```

## Préparation du serveur et du client avec Kerberos

---

Pour configurer l'authentification Kerberos avec SSH, nous devons d'abord créer les principaux pour le serveur et le client, puis générer les clés Kerberos et ajouter les utilisateurs nécessaires dans la base de données Kerberos.

### Création des principaux Kerberos

---

```
kadmin.local  
addprinc -randkey host/kdc.smarttech.sn  
addprinc -randkey host/client1.smarttech.sn
```

### Génération des clés

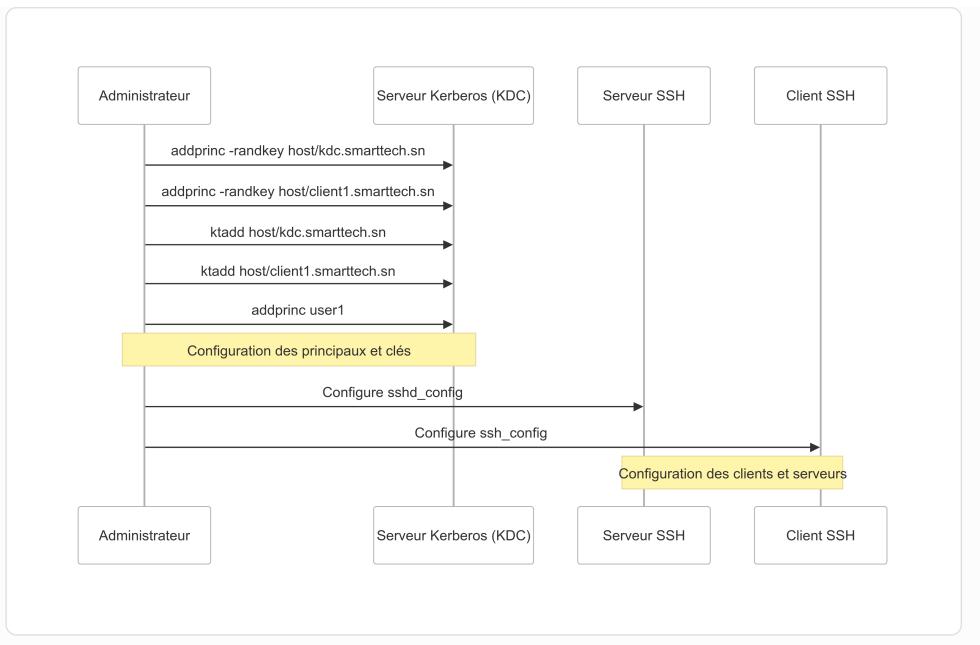
---

```
ktadd host/kdc.smarttech.sn  
ktadd host/client1.smarttech.sn
```

### Ajout d'un utilisateur

---

```
addprinc user1
```



## Configuration du serveur SSH

Sur le serveur où SSH sera activé, nous devons configurer le fichier `sshd_config` pour activer l'authentification Kerberos.

```

# Éditer le fichier /etc/ssh/sshd_config
KerberosAuthentication yes
KerberosOrLocalPasswd yes
GSSAPIAuthentication yes
GSSAPICleanupCredentials yes
# Redémarrer le service SSH
systemctl restart sshd
  
```

## Configuration du client SSH

Sur le client qui se connectera au serveur via SSH, nous devons configurer le fichier `ssh_config` pour activer l'authentification Kerberos.

```

# Éditer le fichier /etc/ssh/ssh_config
GSSAPIAuthentication yes
GSSAPIDelegateCredentials yes
  
```

## Test de la connexion

Pour tester la connexion, nous devons d'abord obtenir un ticket Kerberos, puis utiliser ce ticket pour nous connecter au serveur SSH sans mot de passe.

```
# Obtenir un ticket Kerberos
kinit user1@SMARTTECH.SN
# Vérifier le ticket
klist
# Se connecter au serveur SSH
ssh kdc.smarttech.sn
```

Exemple de sortie réussie :

```
user1@client:~$ ssh kdc.smarttech.sn
The authenticity of host 'kdc.smarttech.sn (192.168.1.211)' can't be estab
ECDSA key fingerprint is SHA256:xCY1GIIrNHrF7DrDCg9gleHB/GenH3PqyGwtm5WiVx
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'kdc.smarttech.sn,192.168.1.211' (ECDSA) to th
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-130-generic x86_64)
[... Informations système ...]
user1@kdc:~$ pwd
/home/user1
```

## Protocoles d'Accès Bureau à Distance

### RDP (Remote Desktop Protocol)

RDP est un protocole développé par Microsoft pour accéder à un bureau Windows à distance. Contrairement à SSH, qui est généralement utilisé pour une interface en ligne de commande, RDP permet une interaction graphique avec l'interface de l'ordinateur distant.

### Caractéristiques principales de RDP

- **Accessibilité graphique** : Permet l'accès complet à l'interface graphique du système distant.
- **Contrôle à distance sécurisé** : RDP utilise des mécanismes de chiffrement pour sécuriser la communication entre le client et le serveur.
- **Utilisation dans des environnements Windows** : RDP est principalement utilisé dans des environnements Windows, mais des clients RDP existent pour d'autres systèmes d'exploitation.

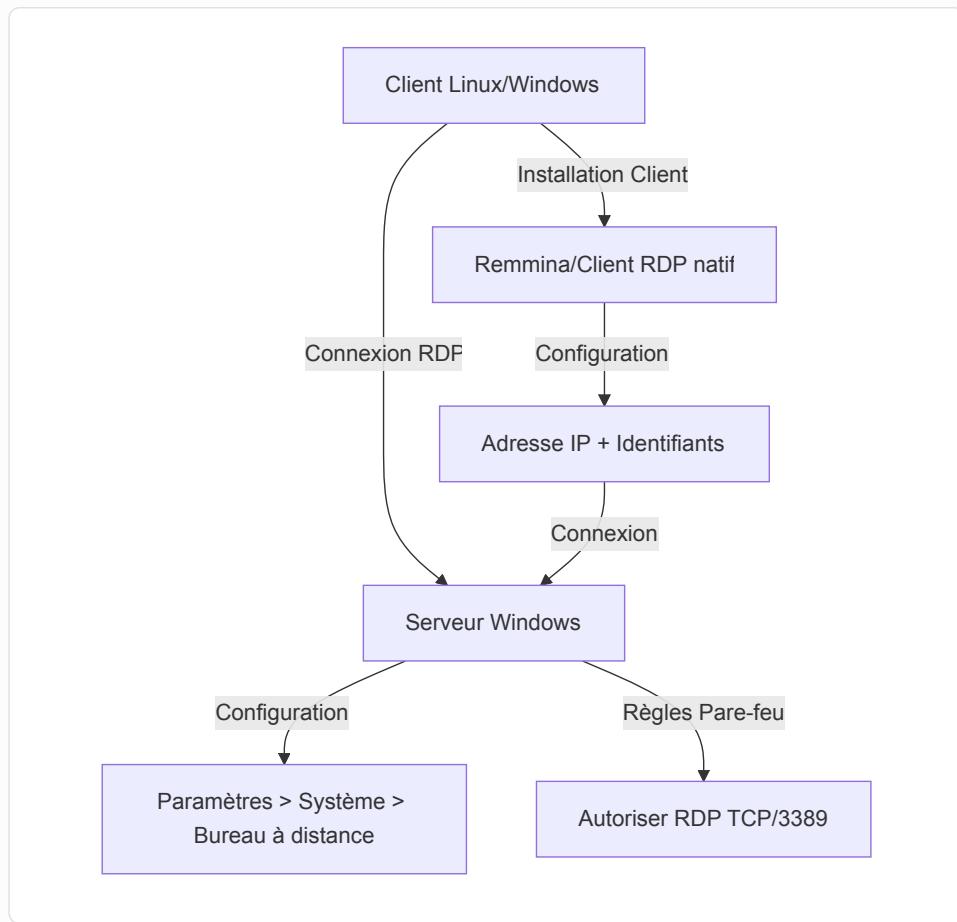
### Configuration de RDP

### Prérequis :

- Une machine Windows (version Pro, Enterprise, ou Education) pour servir de serveur RDP.
- Une machine Windows ou Linux pour agir comme client RDP.
- Accès administrateur sur la machine serveur.

### Activer le Bureau à Distance sur le Serveur Windows

Paramètres > Système > Bureau à distance

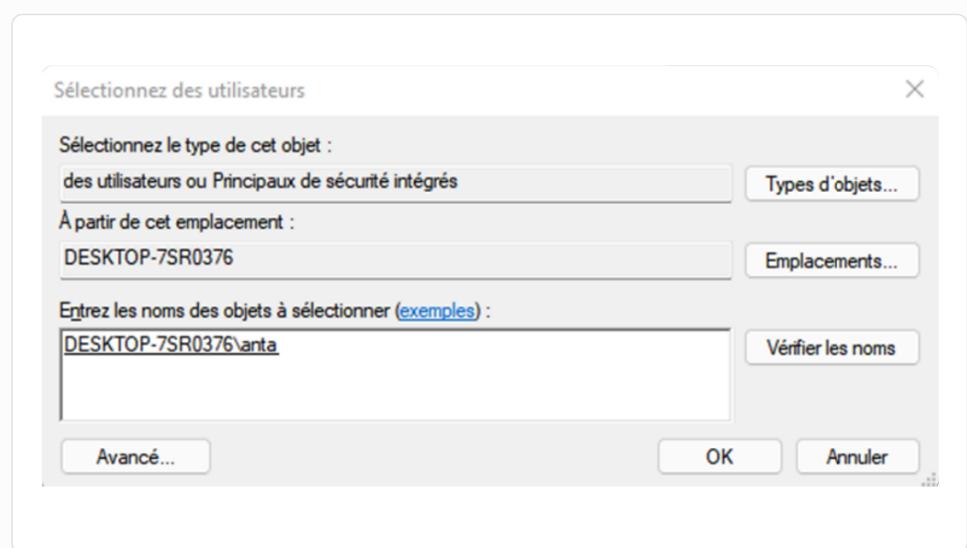
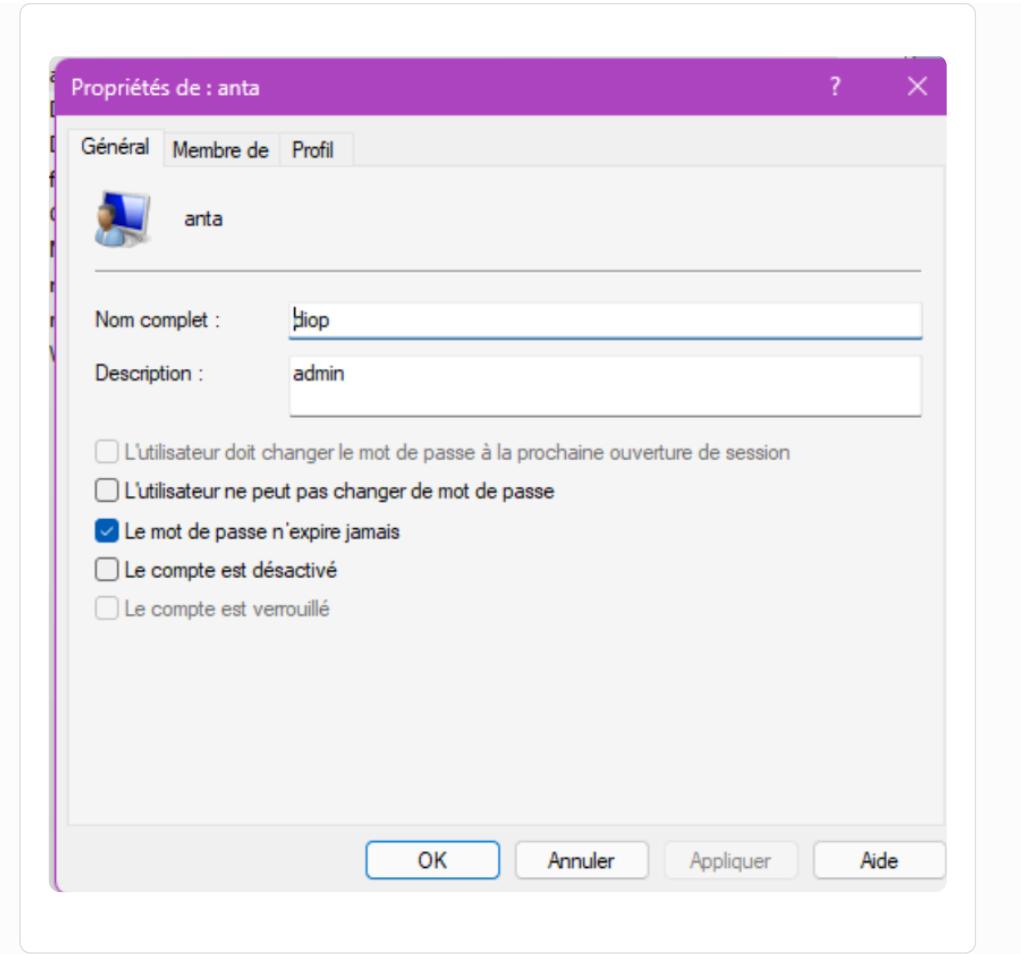


### Système > Bureau à distance

Bureau à distance  
Connectez-vous à cet ordinateur et utilisez-le à partir d'un autre appareil à l'aide de l'application Bureau à distance

Activé

on crée d'abord un utilisateur sur notre serveur Windows



On va autoriser le bureau à distance au niveau des règles de parefeu.

Panneau de configuration > Système et sécurité > Pare-feu Windows Defender > Applications autorisées

## Autoriser les applications à communiquer à travers le Pare-feu Windows Defender

Pour ajouter, modifier ou supprimer des applications et des ports autorisés, cliquez sur Modifier les paramètres.

[Quels sont les risques si une application est autorisée à communiquer ?](#)

 [Modifier les paramètres](#)

### Applications et fonctionnalités autorisées :

Nom	Privé	Public
<input type="checkbox"/> BranchCache - Découverte d'homologue (utilise WSD)	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> BranchCache - Extraction du contenu (utilise HTTP)	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> BranchCache - Serveur de cache hébergé (utilise HTTPS)	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> Bureau à distance	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Bureau à distance (WebSocket)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Calculatrice Windows	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Caméra Windows	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Cartes Windows	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Centre de configuration des graphiques Intel®	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> ceup.exe	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Compte professionnel ou scolaire	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Contacts Microsoft	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[Détails...](#)

[Supprimer](#)

Une fois ceci fait, les clients VNC sur Windows ou Linux peuvent se connecter. on va tester avec un client linux

### Configuration du client Linux

```
# Installation de Remmina
sudo apt update
sudo apt install remmina remmina-plugin-rdp
```

Lancer Remmina, créer une nouvelle connexion avec l'adresse du serveur et les identifiants d'un utilisateur autorisé.

### Remote Connection Profile

Name	Quick Connect
Group	
Protocol	<input checked="" type="radio"/> RDP - Remote Desktop Protocol
<b>Basic</b> <a href="#">Advanced</a> <a href="#">Behavior</a> <a href="#">SSH Tunnel</a> <a href="#">Notes</a>	
Server	192.168.1.200
Username	anta
Password	*****
Domain	
Share Folder	<input type="checkbox"/> (None)
<input type="checkbox"/> Restricted admin mode	
Password hash	
<input type="checkbox"/> Left-handed mouse support <input type="checkbox"/> Disable smooth scrolling	

Remmina ▾    4 Mar 15:44

### Quick Connect

**Quick Connect** ×

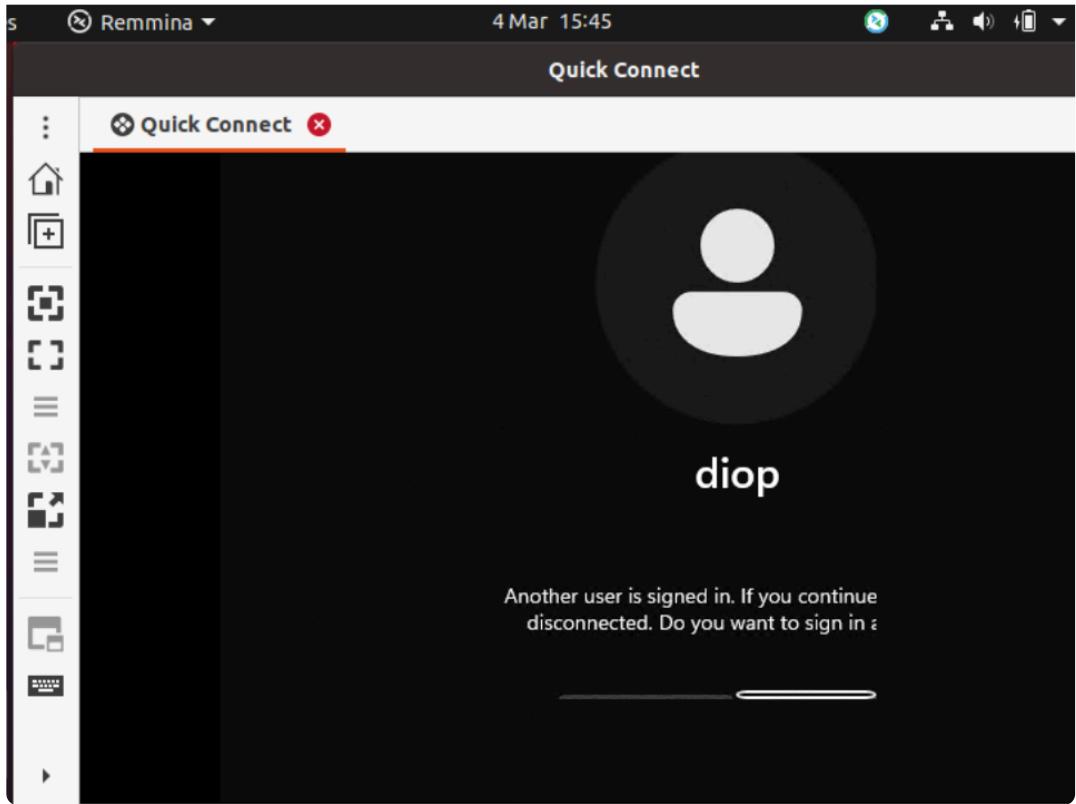
Certificate details:

Subject: CN = Astou  
Issuer: CN = Astou  
Fingerprint: 71:5e:b2:43:f8:eb:ad:af:78:10:1d:9f:5f:32:b8:3c:bc:2f:47:71:ba:83:e6:11:cf:35:cc:32:3

Accept certificate?

On renseigne l'adresse du serveur distant ainsi que l'utilisateur qui est autorisé à accéder au bureau à distance et son mdp

 [Testons la connexion](#)



On peut ainsi accéder à distance à la machine serveuse via RDP.

D'un autre côté, [RDP](#) et [NoVNC](#) sont des solutions très utiles pour accéder à des bureaux distants dans des environnements Windows ou via un navigateur. RDP est adapté aux environnements Windows, tandis que NoVNC fournit une solution multiplateforme et accessible via un simple navigateur web.

## NoVNC

NoVNC est une implémentation du protocole VNC (Virtual Network Computing) qui permet d'accéder à un bureau distant via un navigateur web. Il fonctionne en utilisant le protocole WebSockets et permet de contrôler un ordinateur à distance en utilisant un client HTML5.

### Caractéristiques principales de NoVNC

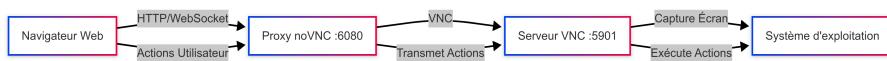
- [Accès via navigateur](#) : Permet d'accéder à un bureau distant sans installer de logiciel client, simplement via un navigateur web.
- [Compatibilité multiplateforme](#) : Étant basé sur HTML5, NoVNC fonctionne sur tous les systèmes d'exploitation modernes sans nécessiter de plugins.
- [Utilisation pour la gestion des serveurs virtuels](#) : NoVNC est souvent utilisé pour gérer des environnements de serveurs virtuels, notamment dans les infrastructures cloud.

### Installation et configuration

```

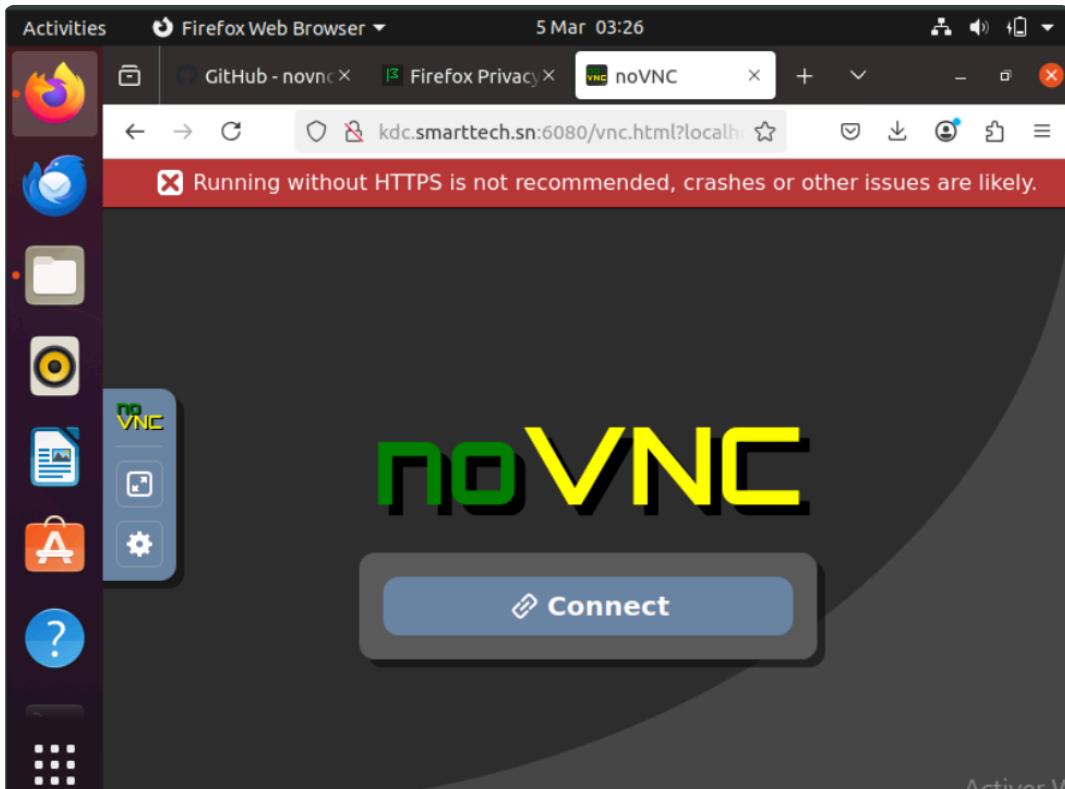
# Installation du serveur VNC
apt install tigervnc-server-standalone -y
# Clonage de noVNC
git clone https://github.com/novnc/noVNC.git ~/novnc
# Création d'un lien symbolique
ln -s ~/noVNC/utils/novnc_proxy /usr/local/bin/novnc_proxy
# Démarrage du proxy WebSocket
novnc_proxy --vnc localhost:5901 --listen 6080

```



## Interface graphique

Lorsqu'on tape l'url au niveau de notre machine cliente on obtient:



noVNC va permettre à notre entreprise d'accéder aux interfaces graphiques des serveurs à distance via un simple navigateur, sans nécessiter de client VNC dédié. Cette solution a amélioré la flexibilité et la sécurité des connexions tout en simplifiant la gestion des accès pour les utilisateurs. Son intégration a donc optimisé notre infrastructure en facilitant l'administration à distance et en réduisant les contraintes techniques.

En combinant **Kerberos avec SSH** pour l'authentification centralisée et en utilisant **RDP** ou

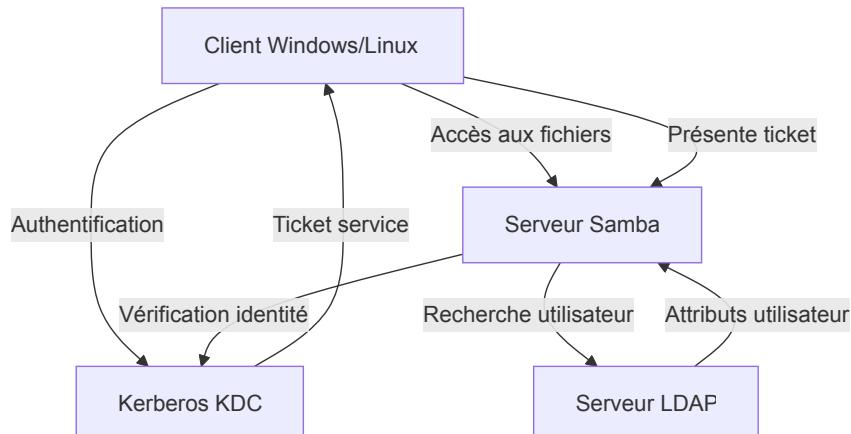
NoVNC pour l'accès graphique, on peut créer une infrastructure sécurisée et efficace pour la gestion à distance des systèmes.

## Samba avec LDAP et Kerberos

### Introduction

L'objectif de cette section est d'intégrer Samba avec LDAP et Kerberos sur SMART TECH.SN. Cette configuration permet :

- D'utiliser LDAP pour centraliser les comptes utilisateurs.
- D'utiliser Kerberos pour l'authentification sécurisée.
- D'autoriser l'accès aux partages Samba sans entrer de mot de passe grâce à Kerberos.



### Installation des paquets

```
sudo apt install samba smbclient krb5-user winbind  
libnss-winbind libpam-winbind smbldap-tools schema2ldif -y
```

### Configuration de LDAP

Ajout du schéma Samba à LDAP

```
# Localiser le schéma Samba
locate samba.schema
# Copier et convertir en LDIF
cp /usr/share/doc/samba/examples/LDAP/samba.schema /etc/ldap/schema/
schema2ldif /etc/ldap/schema/samba.schema > samba.ldif
# Ajout du fichier schema.ldif
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f samba.ldif
# Vérification
ldapsearch -LLL -Y EXTERNAL -H ldapi:/// -b cn=schema,cn=config | grep sam
```

## Configuration de Kerberos pour Samba

### Fichier de configuration

```
# Éditer /etc/krb5.conf
[libdefaults]
default_realm = SMARTTECH.SN
[realms]
SMARTTECH.SN = {
    kdc = kdc.smarttech.sn
    admin_server = kdc.smarttech.sn
}
```

### Création des utilisateurs

```
# Ajouter le principal salif
sudo kadmin.local addprinc salif@SMARTTECH.SN
# Ajouter le service Samba
addprinc -randkey cifs/server.smarttech.sn@SMARTTECH.SN
ktadd -k /etc/krb5.keytab cifs/server.smarttech.sn@SMARTTECH.SN
```

## Configuration de Samba

### Modification de /etc/samba/smb.conf

```
[global]
workgroup = SMARTTECH
security = user
realm = SMARTTECH.SN
encrypt passwords = yes
passdb backend = ldapsam:ldap://server.smarttech.sn
ldap admin dn = cn=admin,dc=smarttech,dc=sn
ldap suffix = dc=smarttech,dc=sn
ldap user suffix = ou=users
ldap group suffix = ou=groups
```

```
ldap machine suffix = ou=computers
ldap ssl = no
kerberos method = system keytab
dedicated keytab file = /etc/krb5.keytab
[partage]
path = /srv/samba/share
read only = no
browseable = yes
guest ok = no
```

### Création du répertoire de partage

```
sudo mkdir -p /srv/samba/share
sudo addgroup smbusers
sudo chown -R root:smbusers /srv/samba/share
sudo chmod -R 2770 /srv/samba/share
```

## Ajout des utilisateurs Samba

```
# Ajout d'un utilisateur Samba
sudo smbpasswd -a salif
# Vérification des utilisateurs Samba
pdbeedit -Lv
```

**Remarque :** Avant ces commandes, l'utilisateur salif doit obligatoirement exister dans l'annuaire LDAP et dans le système (adduser). La commande smbpasswd ajoute seulement les attributs Samba dans l'annuaire.

The screenshot shows a terminal window with four tabs open. The active tab, highlighted with a red bar at the top, displays detailed user information for the user 'salif'. The information includes:

- Unix username: salif
- NT username:
- Account Flags: [U ]
- User SID: S-1-5-21-1320290922-3887964779-207989209-1001
- Primary Group SID: S-1-5-21-1320290922-3887964779-207989209-513
- Full Name:
- Home Directory: \\KDC\\salif
- HomeDir Drive:
- Logon Script:
- Profile Path: \\KDC\\salif\\profile
- Domain: KDC
- Account desc:
- Workstations:
- Munged dial:
- Logon time: 0
- Logoff time: Wed, 06 Feb 2036 15:06:39 GMT
- Kickoff time: Wed, 06 Feb 2036 15:06:39 GMT
- Password last set: Wed, 05 Mar 2025 01:30:48 GMT
- Password can change: Wed, 05 Mar 2025 01:30:48 GMT
- Password must change: never
- Last bad password : 0
- Bad password count : 0
- Logon hours : FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

At the bottom of the terminal window, the prompt 'root@kdc:/home/server#' is visible.

## Tests et validation

### Redémarrage des services

```
sudo systemctl restart smbd nmbd krb5-kdc krb5-admin-server slapd winbind
```

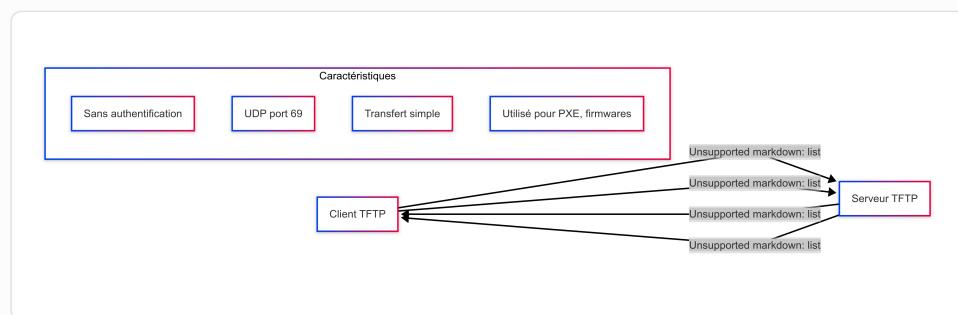
### Connexion sans mot de passe via Kerberos

```
kinit salif@SMARTTECH.SN
smbclient -k -L //kdc.smarttech.sn
smbclient -k //kdc.smarttech.sn/partage
# Vérification du partage Samba
smb: > ls
```

## Services de Transfert de Fichiers

## TFTP (Trivial File Transfer Protocol)

Le Trivial File Transfer Protocol ([TFTP](#)) est un protocole de transfert de fichiers léger, principalement utilisé pour des opérations où l'authentification et les fonctionnalités avancées de gestion de fichiers ne sont pas nécessaires. Il est couramment utilisé pour le déploiement de systèmes d'exploitation via [PXE](#), la mise à jour de firmwares, ainsi que le transfert de configurations d'équipements réseau (routeurs, switches, etc.). Ce rapport présente l'installation, la configuration et l'utilisation d'un serveur TFTP dans un environnement Linux et Windows.



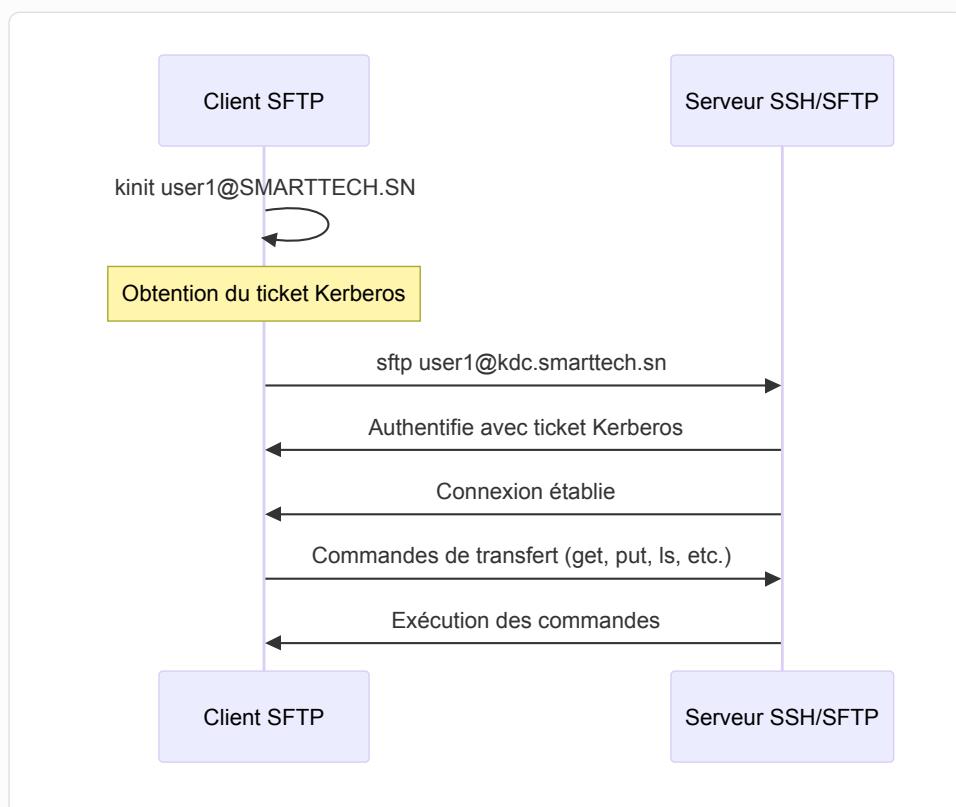
## Installation d'un serveur TFTP sous Linux

```
# Installation du service TFTP
sudo apt update && sudo apt install tftpd-hpa -y
# Configuration du serveur TFTP
sudo nano /etc/default/tftpd-hpa
# Contenu:
TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/srv/tftp"
TFTP_ADDRESS="0.0.0.0:69"
TFTP_OPTIONS="--secure"
# Création du répertoire de stockage
sudo mkdir -p /srv/tftp
sudo chmod -R 777 /srv/tftp
sudo chown -R tftp:tftp /srv/tftp
# Démarrage et activation du serveur
sudo systemctl restart tftpd-hpa
sudo systemctl enable tftpd-hpa
```

Pour des raisons de sécurité, nous préférons utiliser SFTP pour assurer un transfert de fichiers sécurisé.

## SFTP (SSH File Transfer Protocol)

SFTP est un protocole de transfert de fichiers qui utilise SSH pour sécuriser les communications. Il est déjà configuré si SSH est configuré sur le système.



## Test de connexion SFTP

```
# Obtenir un ticket Kerberos
kinit user1
# Se connecter via SFTP
sftp user1@kdc.smarttech.sn
# Exemple de sortie:
user1@kdc:/home/server$ sftp user1@kdc.smarttech.sn
Connected to kdc.smarttech.sn.
sftp> ls -l
drwx----- 3 user1 user1 4096 Mar 3 13:04 snap
sftp>
```

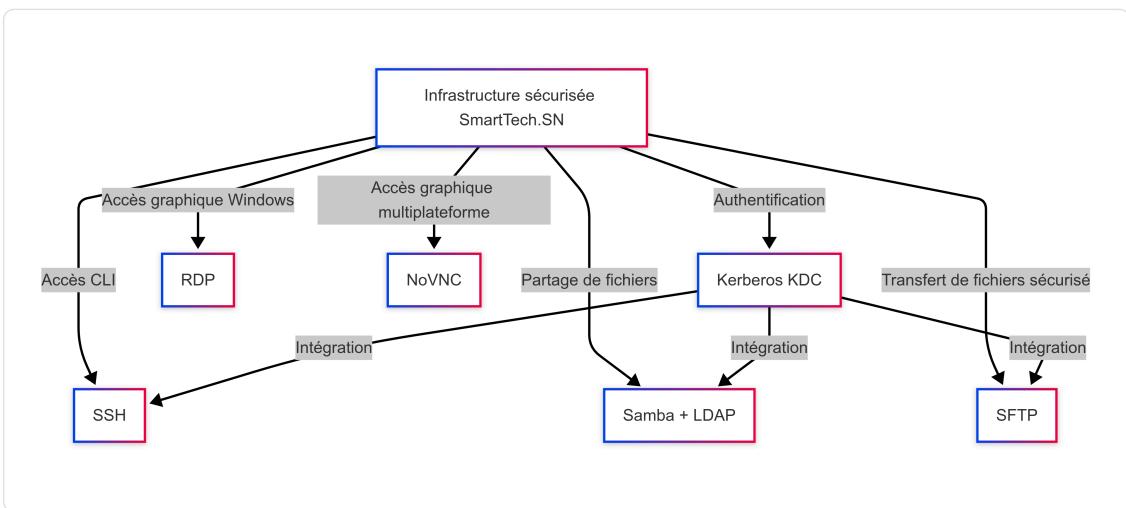
En activant SFTP, nous garantissons des transferts de fichiers sécurisés grâce à l'utilisation du chiffrement SSH, ainsi qu'un meilleur contrôle d'accès et une traçabilité des actions des utilisateurs. Cette solution est bien plus adaptée aux environnements nécessitant une protection renforcée des données.

## Conclusion

Ce rapport a présenté l'implémentation de plusieurs technologies d'accès et de partage sécurisés au sein de l'environnement SmartTech.SN :

- **Kerberos avec SSH** : Cette intégration permet une authentification centralisée et une connexion sans mot de passe, améliorant à la fois la sécurité et l'expérience utilisateur.
- **RDP et NoVNC** : Ces protocoles offrent un accès graphique à distance adapté à différents scénarios, avec RDP pour les environnements Windows et NoVNC pour une solution multiplateforme accessible via navigateur.
- **Samba avec LDAP et Kerberos** : Cette configuration permet de centraliser la gestion des utilisateurs et de sécuriser les partages de fichiers, tout en offrant une intégration transparente avec les environnements Windows.
- **SFTP** : Ce protocole sécurisé remplace avantageusement TFTP pour les transferts de fichiers, grâce à l'utilisation du chiffrement SSH.

En combinant ces technologies, nous avons créé une infrastructure sécurisée et efficace pour la gestion à distance des systèmes. Cette architecture répond aux besoins de sécurité, de simplicité d'administration et d'expérience utilisateur, tout en assurant une traçabilité des actions et une protection des données.



# UNIVERSITÉ CHEIKH ANTA DIOP DE DAKAR

École Supérieure Polytechnique



ECOLE SUPERIEURE  
POLYTECHNIQUE

## Rapport Asterisk et Ldap

Présenté par :

Salif BIAYE

Ndeye Astou DIAGOURAGA

Sous la direction de :

Dr Keba

Enseignant

Année universitaire 2024-2025

\*\*

## Table des Matières

---

### I- Installation des Services

---

#### I-a Installation d'Asterisk

---

#### I-b Installation de L'annuaire LDAP

---

### II- Couplage Asterisk et LDAP

---

#### Configuration de l'annuaire LDAP

---

#### II-a Connexion au serveur LDAP

---

#### II-b Configuration générale des comptes SIP

---

#### II-c Configuration générale du dialplan

---

#### Vérification de la connexion au LDAP

---

#### Ajout des utilisateurs

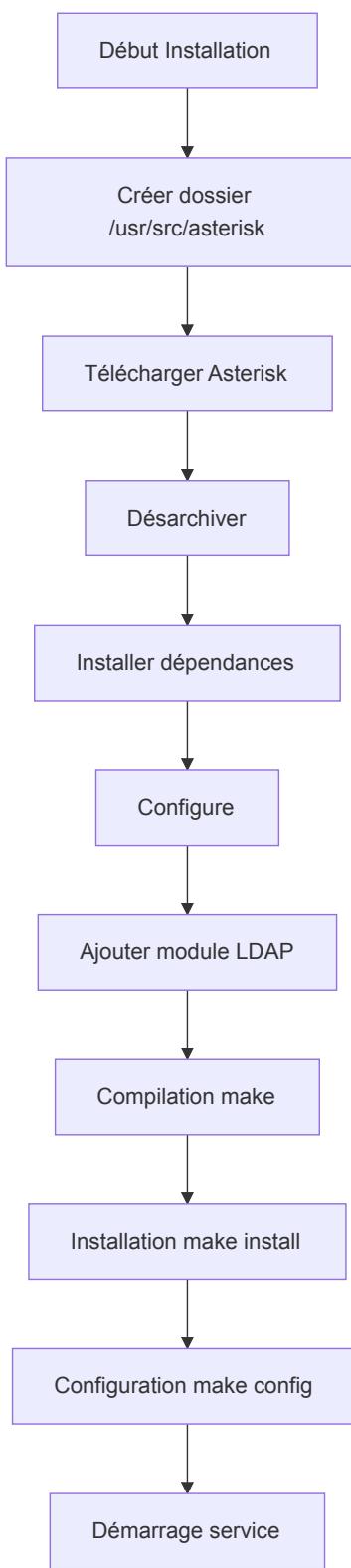
---

#### Test de la configuration

---

# I- Installation des Services

## I-a Installation d'Asterisk



Pour installer Asterisk, nous allons suivre une série d'étapes précises. Commençons par créer le dossier qui contiendra notre paquet Asterisk.

```
root@asterisk-ldap:/usr/src/asterisk# mkdir /usr/src/asterisk
```

On se déplace dans le dossier cree et on telecharge le paquet asterisk comme suit

```
root@asterisk-ldap:/usr/src/asterisk# cd /usr/src/asterisk
root@asterisk-ldap:/usr/src/asterisk# wget http://downloads.asterisk.org/p
--2020-08-25 17:16:34--  http://downloads.asterisk.org/pub/telephony/aster
Résolution de downloads.asterisk.org (downloads.asterisk.org)... 76.164.171.
Connexion à downloads.asterisk.org (downloads.asterisk.org)|76.164.171.238
requête HTTP transmise, en attente de la réponse.. 200 OK
Taille : 43455574 (41M) [application/x-gzip]
Enregistre : «asterisk-17-current.tar.gz»

-17-current.tar.gz      98%[=====]  40,82M  49,5KB/s    tps
```

Maintenant on peut désarchiver le fichier tar comme suit

```
root@asterisk-ldap:/usr/src/asterisk# tar -xvzf asterisk-17-current.tar.gz
asterisk-17.6.0/
asterisk-17.6.0/.cleancount
asterisk-17.6.0/.gitignore
asterisk-17.6.0/.gitreview
asterisk-17.6.0/.lastclean
asterisk-17.6.0/.version
asterisk-17.6.0/BSDBuildfile
asterisk-17.6.0/BUGS
```

Apres avoir désarchivé on se déplace dans le repertoire de base de asterisk selon la version

```
root@asterisk-ldap: /usr/src/asterisk# ls
asterisk-17.6.0    asterisk-17-current.tar.gz

root@asterisk-ldap: /usr/src/asterisk# cd asterisk-17.6.0/
```

Il faut ensuite installer les dépendances requises pour la compilation d'asterisk:

```
root@asterisk-ldap: /usr/src/asterisk/asterisk-17.6.0# apt-get install gcc
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  g++-7 gcc-7 libasan4 libatomic1 libc-dev-bin libc6-dev libcilkrt5
  libgcc-7-dev libitm1 liblsan0 libmpx2 libquadmath0 libstdc++-7-dev
  libtinfo-dev libtsan0 libubsan0 linux-libc-dev manpages-dev
```

on télécharge le deuxième

```
root@asterisk-ldap: /usr/src/asterisk/asterisk-17.6.0# ./contrib/scripts/i
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
```

```
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  aptitude-common libcwidget3v5
Paquets suggérés :
```

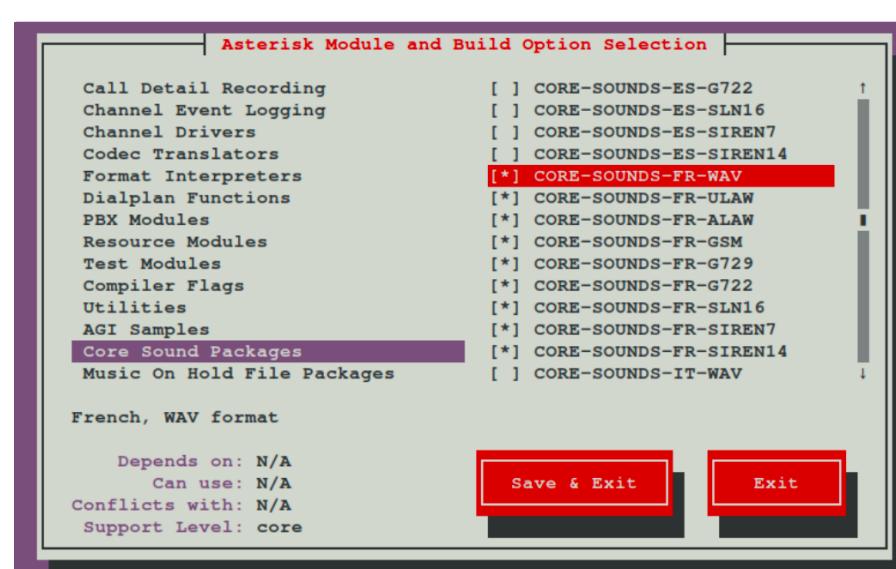
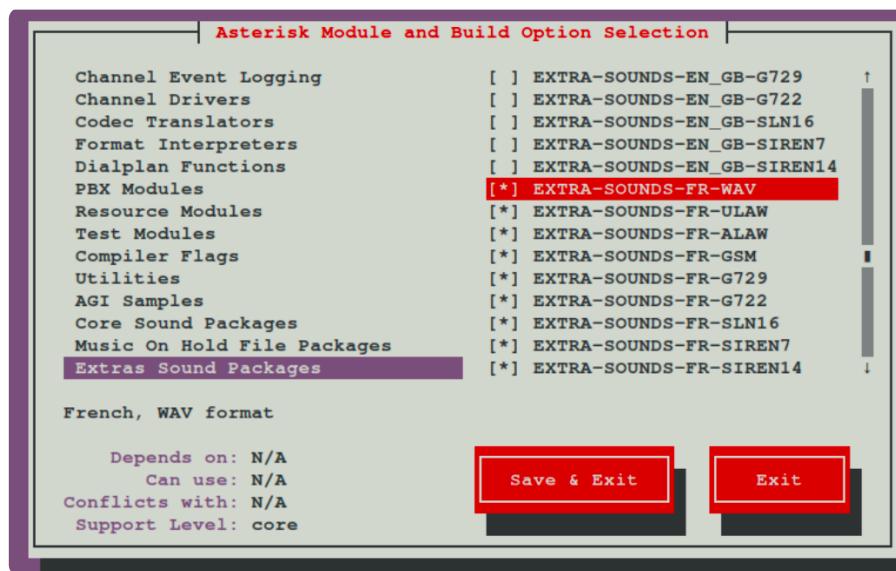
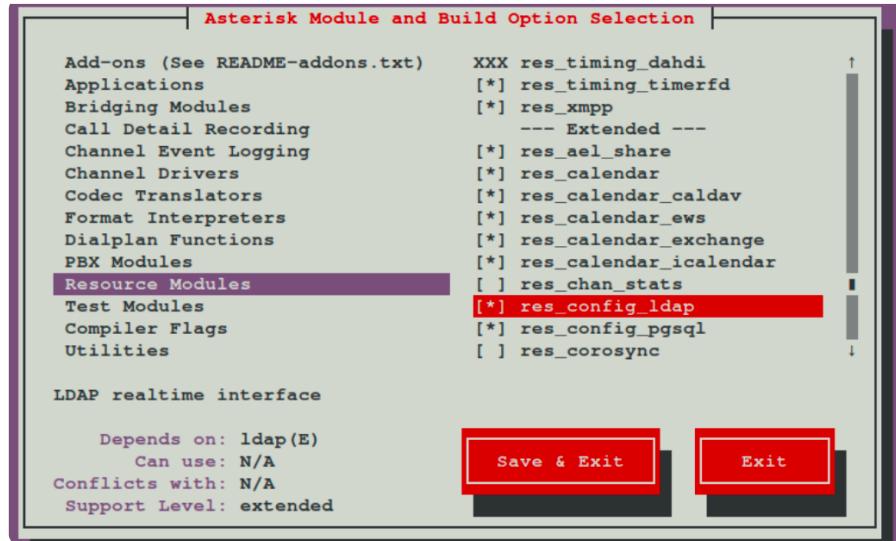
Nous allons ensuite compiler asterisk

```
root@asterisk-ldap: /usr/src/asterisk/asterisk-17.6.0# ./configure  
checking build system type... x86_64-pc-linux-gnu  
checking host system type... x86_64-pc-linux-gnu  
checking for gcc... gcc  
checking whether the C compiler works... yes  
checking for C compiler default output file name... a.out
```

On aura comme résultat

On ajoute le module LDAP pour Asterisk et sélectionnons des voix francaises:

```
root@asterisk-ldap: /usr/src/asterisk/asterisk-17.6.0# make menuconfig
```



Apres tabulation on Save & Exit pour sauvegardé

## Compilation

---

```
root@asterisk-ldap:/usr/src/asterisk/asterisk-17.6.0# make
[CC] astcanary.c -> astcanary.o
[LD] astcanary.o -> astcanary
[CC] astdb2sqlite3.c -> astdb2sqlite3.o
[CC] hash/hash.c -> hash/hash.o
[CC] hash/hash_bigkey.c -> hash/hash_bigkey.o
[CC] hash/hash_buf.c -> hash/hash_buf.o
[CC] hash/hash_func.c -> hash/hash_func.o
```

## Installation

---

```
root@asterisk-ldap:/usr/src/asterisk/asterisk-17.6.0# make install
Installing modules from channels...
Installing modules from pbx...
Installing modules from apps...
Installing modules from codecs...
Installing modules from formats...
Installing modules from cdr...
```

```
root@asterisk-ldap:/usr/src/asterisk/asterisk-17.6.0# make samples
Installing adsi config files...
/usr/bin/install -c -d "/etc/asterisk"
Installing configs/samples/asterisk.adsi
Installing configs/samples/telcordia-1.adsi
Installing other config files...
Installing file configs/samples/acl.conf.sample
Installing file configs/samples/adsi.conf.sample
Installing file configs/samples/agents.conf.sample
Installing file configs/samples/alarmreceiver.conf.sample
Installing file configs/samples/alsa.conf.sample
Installing file configs/samples/amd.conf.sample
```

```
root@asterisk-ldap:/usr/src/asterisk/asterisk-17.6.0# make config
```

on démarre le service asterisk comme suit

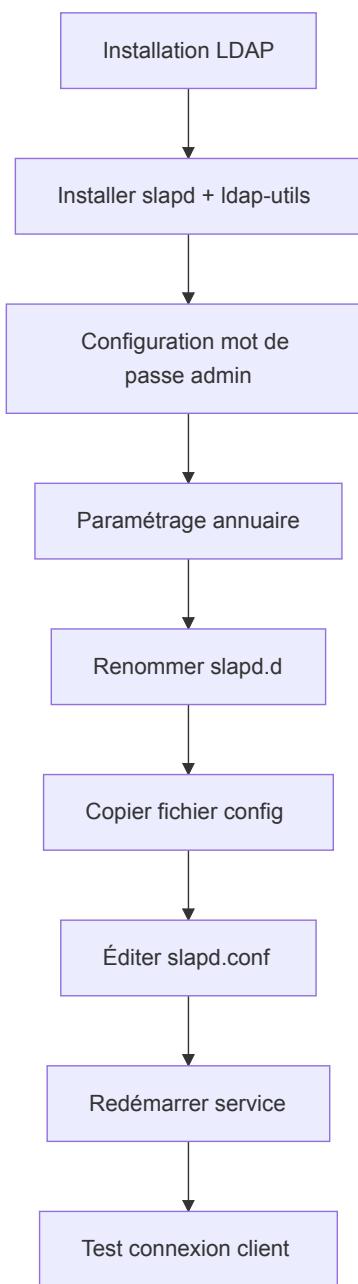
```
root@asterisk-ldap:/usr/src/asterisk/asterisk-17.6.0# /etc/init.d/asterisk
[ ok ] Starting asterisk (via systemctl): asterisk.service.
root@asterisk-ldap:/usr/src/asterisk/asterisk-17.6.0#
```

Pour vérifier le fonctionnement du serveur Asterisk, on peut lancer la console :

```
root@asterisk-ldap:/usr/src/asterisk/asterisk-17.6.0# asterisk -rvvvvv
Asterisk 17.6.0, Copyright (C) 1999 - 2018, Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
```

```
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for  
This is free software, with components licensed under the GNU General Publ  
=====  
Connected to Asterisk 17.6.0 currently running on asterisk-ldap (pid = 164  
asterisk-ldap:*CLI>
```

## I-b Installation de L'annuaire LDAP



Un annuaire LDAP est une base de donnée non \*\*Relationnelle .\*\*

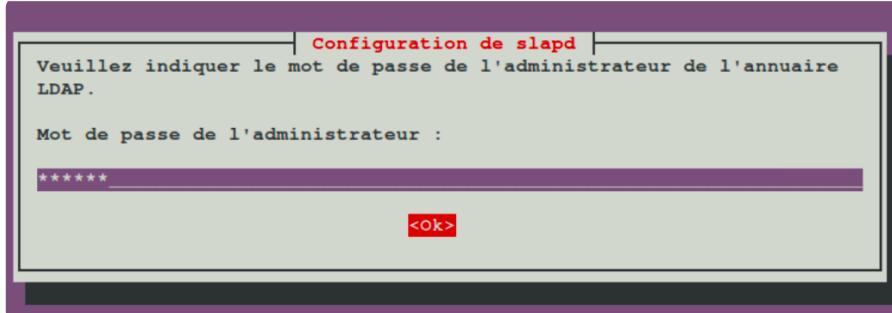
Elle peut être associé à un système de stockage de données permettant de rendre accessible un ensemble d'informations à tous les utilisateurs de ce système.

Installation des paquets nécessaires : slapd le serveur LDAP ldap-utils pour les commandes côté client

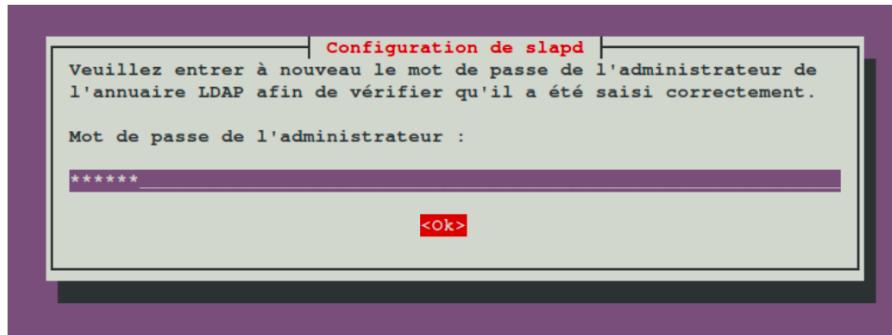
```
root@asterisk-ldap:~# apt install slapd ldap-utils
Lecture des listes de paquets... Fait
```

```
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Paquets suggérés :
  libsasl2-modules-gssapi-mit | libsasl2-modules-gssapi-heimdal
Les NOUVEAUX paquets suivants seront installés :
```

on nous demande de choisir un mot de passe pour l'administrateur



on retape le mot de passe



## Paramètres principaux du fichier de configuration :

Ici nous utiliserons la méthode de config dans un fichier . Pour ça il faut :

-Impérativement renommer le répertoire `/etc/ldap/slapd.d/` en `/etc/ldap/slapd.d.old/` comme ceci :

```
root@asterisk-ldap:~# mv /etc/ldap/slapd.d/ /etc/ldap/slapd.d.old/
root@asterisk-ldap:~#
```

\*\*-Et\*\* copier un fichier d'exemple de configuration d'un serveur LDAP qui se trouve dans

`/usr/share/slapd/slapd.conf` comme ceci :

```
root@asterisk-ldap:~# cp /usr/share/slapd/slapd.conf /etc/ldap/
root@asterisk-ldap:~#
```

```
root@asterisk-ldap:/etc/ldap# ls
```

## ldap.conf sasl2 schema slapd.conf slapd.d.old

Maintenant éditons le fichier \*\*slapd.conf\*\* copier dans /\*\*etc/ldap/\*\* et y mettre la configuration qui se

trouve dans le tableau suivant :

Nom paramètre	Signification	Valeur possible
moduleload	Pilote de base de donnée à charger	<a href="#">back_hdb</a>
backend	Le backend est en fait le « moteur » permettant le stockage ou la récupération de donnée en réponse à une requête LDAP dans un annuaire	hdb
database	Type de base de donnée	hdb
suffixe	Le nom de la racine de l'annuaire	'dc=smarttech,dc=sn '
rootdn	Le dn de l'administrateur	'cn=admin,dc=smarttech,dc=sn '
rootpw(il faut ajouter en bas de rootdn)	Mot de passe de l'administrateur	passer
access	Liste d'accès à une entrée	o hange attrs=userPassword,shadowLastC by dn='cn=admin,dc=smarttech,dc=sn ' write by anonymous by self write by * none
access		to by dn='cn=admin,dc=smarttech,dc=sn ' write by read

voici quelques exemples:

```

GNU nano 2.9.3                               slapd.conf                                Modifié

loglevel     none

# Where the dynamically loaded modules are stored
modulepath   /usr/lib/ldap
moduleload   back_hdb

# The maximum number of entries that is returned for a search operation
sizelimit 500

# The tool-threads parameter sets the actual amount of cpu's that is used
# for indexing.
tool-threads 1

#####
# Specific Backend Directives for @BACKEND@:
# Backend specific directives apply to this backend until another
# 'backend' directive occurs
backend      hdb

```

```

# 'database' directive occurs
database      hdb

# The base of your directory in database #1
suffix        "dc=smarttech,dc=sn"

# rootdn directive for specifying a superuser on the database. This is nee
# for syncrepl.
rootdn        "cn=admin,dc=smarttech,dc=sn"
rootpw        passer

```

```

# These access lines apply to database #1 only
access to attrs=userPassword,shadowLastChange
    by dn="cn=admin,dc=smarttech,dc=sn" write
    by anonymous auth
    by self write
    by * none

```

```

GNU nano 2.9.3                               slapd.conf                                Modifié

# want SASL (and possible other things) to work
# happily.
access to dn.base="" by * read

# The admin dn has full write access, everyone else
# can read everything.
access to *
    by dn="cn=admin,dc=smarttech,dc=sn" write
    by * read

```

Enregistrons et redémarrer le serveur comme suit :

```

root@asterisk-ldap:/etc/ldap# service slapd start

```

```
root@asterisk-ldap:/etc/ldap#
```

Vérifions si le serveur écoute sur le port \*\*389\*\* comme suit :

```
root@asterisk-ldap:/etc/ldap# netstat -anp | grep -w 389
tcp        0      0 0.0.0.0:389          0.0.0.0:*                  LISTEN
tcp6       0      0 ::1:389           ::*:*                   LISTEN
root@asterisk-ldap:/etc/ldap#
```

Nous Constatons que le serveur est t'en Bon état et que l'algorithme a été claire ....

Pour le teste nous allons paramétrer le client ldap...

comme je suis en local j'utilise la même machine comme serveur et client ...

Le fichier de configuration du client est dans </etc/ldap/ldap.conf> nous allons lui renseigner deux

(2) paramètres qui sont :

- Le nom de la racine de l'annuaire
- L'adresse Ip du serveur LDAP

```
GNU nano 2.9.3                               ldap.conf                                Modifié

# LDAP Defaults
#
# See ldap.conf(5) for details
# This file should be world readable but not world writable.

BASE    dc=smarttech,dc=sn
URI    ldap://192.168.158.128

#SIZELIMIT 12
#TIMELIMIT 15
#DEREF never

# TLS certificates (needed for GnuTLS)
TLS_CACERT      /etc/ssl/certs/ca-certificates.crt
```

Test de la connexion client-serveur : Pour tester si la connexion entre le client et le serveur fonctionne c'est simple

Il faut ouvrir une console chez le client et tapez la commande suivante :

```
root@asterisk-ldap:/etc/ldap# ldapsearch -x
# extended LDIF
#
# LDAPv3
# base <dc=smarttech,dc=sn> (default) with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# search result
search: 2
result: 32 No such object

# numResponses: 1
root@asterisk-ldap:/etc/ldap#
```

## II- Couplage Asterisk et LDAP

Avant toute chose, nous allons indiquer à notre \*\*LDAP\*\* les variables à prendre en compte pour

[Asterisk](#). Ceci se passe dans un schéma disponible Cliquez-ici. Copier l'ensemble de ce fichier dans un fichier [asterisk.schema](#) que vous placerez dans le répertoire [/etc/ldap/schema](#).

Il faut ensuite indiquer à LDAP de prendre en compte ce schéma. Pour cela, il faut ajouter ceci dans le fichier de configuration [/etc/ldap/slapd.conf](#) comme ceci :

```
GNU nano 2.9.3                               slapd.conf

# This is the main slapd configuration file. See slapd.conf(5) for more
# info on the configuration options.

#####
# Global Directives:

# Schema and objectClass definitions
include      /etc/ldap/schema/core.schema
include      /etc/ldap/schema/cosine.schema
include      /etc/ldap/schema/nis.schema
include      /etc/ldap/schema/inetorgperson.schema
include      /etc/ldap/schema/asterisk.schema

# Where the pid file is put. The init.d script
# will not stop the server if you change this.
pidfile     /var/run/slapd/slapd.pid
```

Pour que cette modification soit prises en compte, il faut relancer le serveur LDAP :

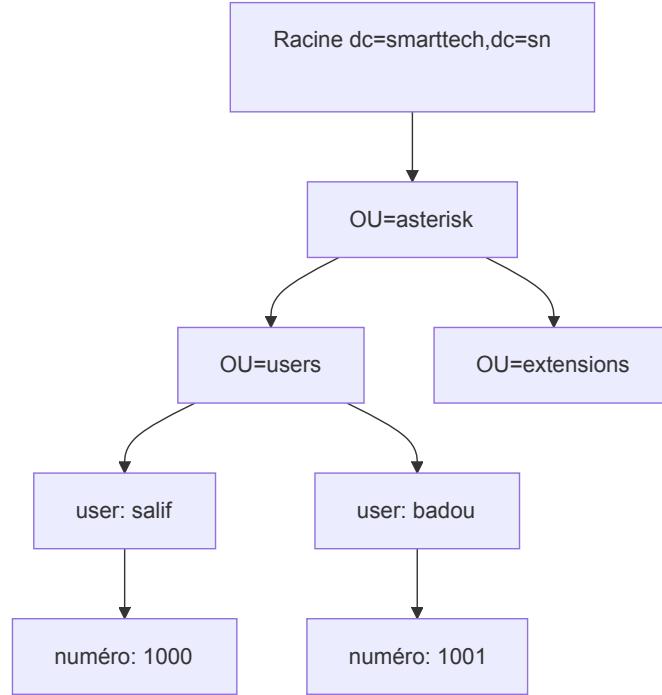
```
root@asterisk-ldap:/etc/ldap# /etc/init.d/asterisk start
[ ok ] Starting slapd (via systemctl): slapd.service.
root@asterisk-ldap:/etc/ldap#
```

### Configuration de l'annuaire LDAP

Afin de stocker les paramètres des comptes SIP de l'Asterisk, il faut que nous ayons un endroit où les stockés.

j'ai décidé de séparer ces deux informations par soucis de lisibilité. J'ai donc choisi de créer une OU (Organizational Unit) dédiée à Asterisk. J'ai donc créer une OU Asterisk et avec deux "sous-OU" users et extensions. Vous pouvez ajouter cette configuration comme cela :

Tout d'abord nous allons créer un fichier [LDIF](#) nommé [racine.ldif](#) dans [/etc/ldap/](#) contenant la racine de notre annuaire LDAP comme ceci :



```

GNU nano 2.9.3
# racine.ldif
dn: dc=smarttech,dc=sn
objectClass: dcObject
objectClass: organization
dc: smarttech
o: smarttech.sn

```

On alimente avec la commande suivante

```

root@asterisk-ldap:/etc/ldap# ldapadd -x -D "cn=admin,dc=smarttech,dc=sn" -W -f racine.ldif
Enter LDAP Password:
adding new entry "dc=smarttech,dc=sn"

root@asterisk-ldap:/etc/ldap#

```

Nous allons afficher les entrés de l'annuaire comme ceci:

```

root@asterisk-ldap:/etc/ldap# ldapsearch -x
# extended LDIF
#
# LDAPv3
# base <dc=smarttech,dc=sn> (default) with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# smarttech.sn
dn: dc=smarttech,dc=sn
objectClass: organization
objectClass: dcObject
o: smarttech
dc: smarttech

# search result
search: 2
result: 0 Success

```

```
# numResponses: 2
# numEntries: 1
root@asterisk-ldap:/etc/ldap#
```

Ensuite nous allons créez aussi un fichier \*\*LDIF\*\* nommé \*\*info.ldif\*\* dans \*\*/etc/ldap/\*\* contenant la information d'asterisk et celle de l'annuaire comme ceci :

```
GNU nano 2.9.3                                         info.ldif
# OU asterisk
dn: ou=asterisk,dc=smarttech,dc=sn
objectClass: top
objectClass: organizationalUnit
ou: asterisk

# OU users
dn: ou=users,ou=asterisk,dc=smarttech,dc=sn
objectClass: top
objectClass: organizationalUnit
ou: users

# OU extensions
dn: ou=extensions,ou=asterisk,dc=smarttech,dc=sn
objectClass: top
objectClass: organizationalUnit
ou: extensions
```

Il faut ensuite ajouter ce fichier \*\*LDIF\*\* à notre arborescence \*\*LDAP\*\* , pour cela il faut utiliser la commande suivante comme suit dans un terminal :

```
root@asterisk-ldap:/etc/ldap# ldapadd -x -D "cn=admin,dc=smarttech,dc=sn" -W -f info.ldif
Enter LDAP Password:
adding new entry "ou=asterisk,dc=smarttech,dc=sn"
adding new entry "ou=users,ou=asterisk,dc=smarttech,dc=sn"
adding new entry "ou=extensions,ou=asterisk,dc=smarttech,dc=sn"
```

Nous allons afficher les entrés de l'annuaire encore:

```
# OU asterisk
dn: ou=asterisk,dc=smarttech,dc=sn
objectClass: top
objectClass: organizationalUnit
ou: asterisk

# OU users
dn: ou=users,ou=asterisk,dc=smarttech,dc=sn
objectClass: top
objectClass: organizationalUnit
ou: users

# OU extensions
dn: ou=extensions,ou=asterisk,dc=smarttech,dc=sn
objectClass: top
objectClass: organizationalUnit
ou: extensions

# search result
search: 2
```

```
result: 0 Success
```

## II-a Connexion au serveur LDAP

Configuration de la connexion dans /etc/asterisk/res\_ldap.conf :

```
[_general]
host=192.168.158.128
port=389
protocol=3
basedn=ou=asterisk,dc=smarttech,dc=sn
pass=passer
user=cn=admin,dc=smarttech,dc=sn

[sip]
name = uid
callerid = AstAccountCallerID
canreinvite = AstAccountCanReinvite
context = AstAccountContext
host = AstAccountHost
type = AstAccountType
mailbox = AstAccountMailbox
md5secret = AstAccountRealmedPassword
fullcontact = AstAccountFullContact
nat = AstAccountNAT
qualify = AstAccountQualify
allow = AstAccountAllowedCodec
useragent = AstAccountUserAgent
lastms = AstAccountLastQualifyMilliseconds
additionalFilter=(objectClass=AsteriskSIPUser)

[extensions]
context = AstContext
exten = AstExtension
priority = AstPriority
app = AstApplication
appdata = AstApplicationData
additionalFilter =(objectClass=AsteriskExtension)
```

```
GNU nano 2.9.3                                res_ldap.conf                               Mon Mar 20 10:45:23 2017

; Note that you can configure an ldaps: url here to get TLS support.
; Detailed configuration of certificates and supported CAs is done in your
; ldap.conf file for OpenLDAP clients on your system.
; This requires that you have OpenLDAP libraries compiled with TLS support

; ****
; NOTE: res_ldap.conf should be chmod 600 because it contains the plain-text LDAP
; password to an account with WRITE access to the asterisk configuration.
; ****

[_general]
;

; Specify one of either host and port OR url. URL is preferred, as you can
; use more options.
host=192.168.158.128                         ; LDAP host
port=389
```

```
;url=ldap://ldap3.mydomain.com:3890
protocol=3 ; Version of the LDAP protocol to use; default
basedn=ou=asterisk,dc=smarttech,dc=sn ; Base DN
user=cn=admin,dc=smarttech,dc=sn ; Bind DN
pass=passer ; Bind password
```

```
[extensions]
context = AstExtensionContext
exten = AstExtensionExten
priority = AstExtensionPriority
app = AstExtensionApplication
appdata = AstExtensionApplicationData
additionalFilter=(objectClass=AstExtension)
```

```
[sip]
name = uid ; We use the "cn" as the default value for name
          ; because objectClass=AsteriskSIPUser does not allow
          ; If your entry combines other objectClasses and you
          ; prefer to change the line to be name = uid, es
          ; contain spaces in the cn field.
          ; You may also find it appropriate to use someth
          ; This is possible by changing the line above to
          ; prefer).
          ;
amaflags = AstAccountAMAFlags
callgroup = AstAccountCallGroup
callerid = AstAccountCallerID
canreinvite = AstAccountCanReinvite
directmedia = AstAccountDirectMedia
context = AstAccountContext
dtmfmode = AstAccountDTMFMode
fromuser = AstAccountFromUser
fromdomain = AstAccountFromDomain
fullcontact = AstAccountFullContact
```

```
fullcontact = gecos
host = AstAccountHost
insecure = AstAccountInsecure
mailbox = AstAccountMailbox
md5secret = AstAccountRealmedPassword ; Must be an MD5 hash
          ; {md5} but it
          ; Generate the hash
          ; echo "my_pass"

nat = AstAccountNAT
deny = AstAccountDeny
permit = AstAccountPermit
pickupgroup = AstAccountPickupGroup
port = AstAccountPort
qualify = AstAccountQualify
restrictcid = AstAccountRestrictCID
rtptimeout = AstAccountRTPTimeout
```

```
type = AstAccountType
useragent = AstAccountUserAgent
```

```

Disallow = AstAccountDisallowedCodec
Allow = AstAccountAllowedCodec
MusicOnHold = AstAccountMusicOnHold
RegSeconds = AstAccountExpirationTimestamp
RegContext = AstAccountRegistrationContext
RegExten = AstAccountRegistrationExten
CanCallForward = AstAccountCanCallForward
IPAddr = AstAccountIPAddress
DefaultUser = AstAccountDefaultUser
RegServer = AstAccountRegistrationServer
LastMS = AstAccountLastQualifyMilliseconds
SupportPath = AstAccountPathSupport
AdditionalFilter=(objectClass=AsteriskSIPUser)

```

Comme vous pouvez le voir la section \*\*[sip]\*\* permet de faire la translation entre les variables Asterisk et les variables LDAP afin que les deux serveur puissent se "comprendre".

Il en est de même pour la partie [\[extensions\]](#).

Une fois que nous avons définit la connexion entre le serveur [Asterisk](#) et le serveur LDAP, il faut dire à [Asterisk](#) où il doit aller chercher les paramètres des utilisateurs SIP. Pour cela, il faut modifier le fichier [/etc/asterisk/extconfig.conf](#) comme ceci :

```

; The only option available currently is the 'p' option, which disallows
; extension pattern queries to the database. If you have no patterns defined
; in a particular context, this will save quite a bit of CPU time. However,
; note that using dynamic realtime extensions is not recommended anymore as a
; best practice: instead, you should consider writing a static dialplan with
; proper data abstraction via a tool like func_odbc.

sipusers => ldap,"ou=users,ou=asterisk,dc=barry,dc=sn",sip
sippeers => ldap,"ou=users,ou=asterisk,dc=barry,dc=sn",sip
extensions => ldap,"ou=extensions,ou=asterisk,dc=barry,dc=sn",extensions

```

## II-b Configuration générale des comptes SIP

Configuration dans [/etc/asterisk/sip.conf](#) :

```

[general]
rtcachefriends=yes
callevents=yes
realm=smarttech.sn

```

L'authentification des utilisateur SIP. Si vous utilisez déjà un Asterisk, veuillez rajouter ces

Paramètres sans écraser les anciens contenus dans votre fichier.

[rtcachefriends=yes](#) //permet de mettre en cache les infos des utilisateurs (obligatoire car

elle permet de garder en mémoire l'adresse IP avec laquelle l'utilisateur s'est connecté. Sans ce paramètre l'appel n'aboutira pas car l'Asterisk ne saura pas trouver les utilisateurs.

- callevents=yes** //permet de remonter les informations concernant un appel
  - realm=smarttech.sn** //nom de domaine géré par l'annuaire LDAP

## II-c Configuration générale du dialplan

## Configuration dans /etc/asterisk/extensions.conf :

```
GNU nano 2.9.3                                         extensions.conf

exten => _X.,n,Wait(1.25)
exten => _X.,n,SayDigits(${CALLERID(ani)})           ; playback again in case of mis$
exten => _X.,n,Return()

; For more information on applications, just type "core show applications" at y$ 
; friendly Asterisk CLI prompt.
;
; "core show application <command>" will show details of how you
; use that particular application in this file, the dial plan.
; "core show functions" will list all dialplan functions
; "core show function <COMMAND>" will show you more information about
; one function. Remember that function names are UPPER CASE.

[internal]
switch => Realtime@
```

## Vérification de la connexion au LDAP

Avant toute chose, il faut recharger la configuration afin que les modifications que l'on a apporté soient prises en compte. Pour cela, il faut se connecter au CLI d'Asterisk :

```
root@asterisk-ldap:/etc/asterisk# asterisk -vvvvvvvvvvvvvvvvvvvvvvvvvvvvv  
  
Asterisk 17.6.0, Copyright (C) 1999 - 2018, Digium, Inc. and others.  
Created by Mark Spencer <markster@digium.com>  
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.  
This is free software, with components licensed under the GNU General Public License  
=====  
Connected to Asterisk 17.6.0 currently running on bhpro (pid = 1258)  
asterisk*CLI> reload  
-- Reloading module 'extconfig' (Configuration)
```

Mettre plusieurs \*\*v\*\* dans la commande permet d'augmenter le niveau de debug. Il faut ensuite taper les commandes suivantes dans le CLI:

```
asterisk-ldap*CLI> module reload
```

cette commande permet de recharger tous les modules qu'asterisk a charger notamment le module LDAP puis chargeons le fichier pjsip.conf

```
asterisk-ldap*CLI> pjsip reload
```

Nous rechargeons ensuite le fichier extensions.conf

```
asterisk-ldap*CLI> dialplan reload
Dialplan reloaded.
== Setting global variable 'CONSOLE' to 'Console/dsp'
== Setting global variable 'IAXINFO' to 'guest'
== Setting global variable 'TRUNK' to 'DAHDI/G2'
== Setting global variable 'TRUNKMSD' to '1'
-- Including switch 'DUNDi/e164' in context 'dundi-e164-switch'
-- Including switch 'Realtime/@' in context 'internal'
-- Including switch 'Lua//' in context 'public'
-- Including switch 'Lua//' in context 'demo'
-- Including switch 'Lua//' in context 'local'
-- Including switch 'Lua//' in context 'default'
-- Including switch 'DUNDi/e164' in context 'ael-dundi-e164-switch'
-- Time to scan old dialplan and merge leftovers back into the new: 0.000605 sec
-- Time to restore hints and swap in new dialplan: 0.000016 sec
-- Time to delete the old dialplan: 0.000537 sec
-- Total time merge_contexts_delete: 0.001158 sec
-- pbx_config successfully loaded 51 contexts (enable debug for details).
asterisk-ldap*CLI>
```

La commande \*\*realtime show ldap status\*\* permet de connaître l'état de la connexion entre \*\*le\*\*

[serveur Asterisk](#) et [le serveur LDAP](#).

```
asterisk-ldap*CLI> realtime show ldap status
Connected to 'ldap://192.168.158.1278:389', baseDN ou=asterisk,dc=barry,dc=sn with 1
asterisk-ldap*CLI>
```

## Ajout des utilisateurs

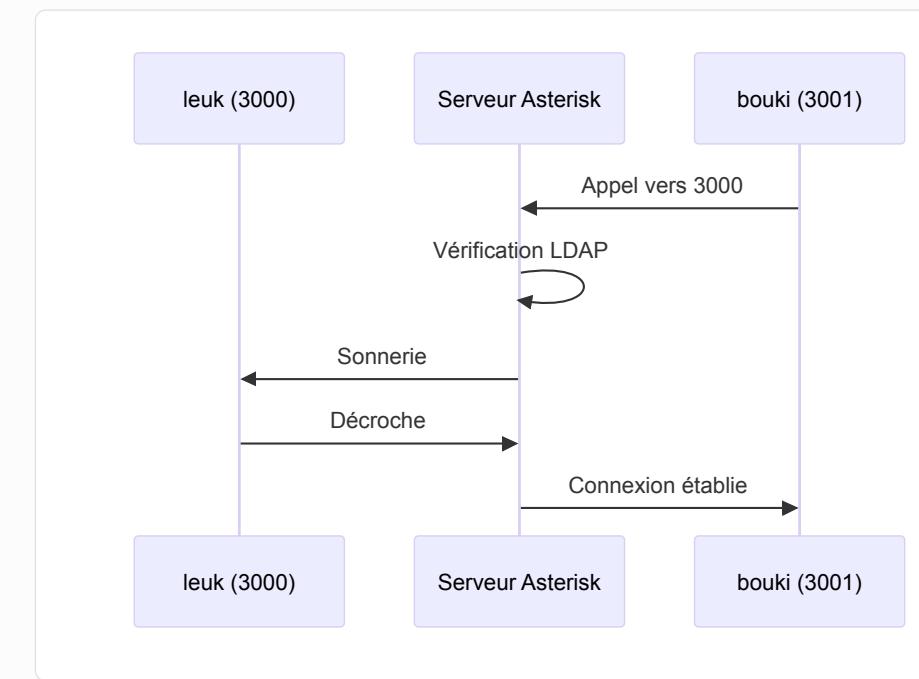
Maintenant que nos deux serveurs communiquent correctement ensemble, nous pouvons ajouter les utilisateurs dans notre annuaire LDAP. Pour ça on a un script python pour ajouter des users ldap une version via la console et une autre avec tkinter tous les deux doivent être accompagné d'un troisième fichier bash adduser.sh Les voici : [users-gui.py](#) , [users.py](#) [addusers.sh](#)

## Test de la configuration

voici une [video](#) montrant comment executer le script les resultat et on va utiliser un outil pour appeler zoiper

Création de deux utilisateurs pour le test :

- leuk (extension: 3000) - bouki(extension: 3001)



# UNIVERSITÉ CHEIKH ANTA DIOP DE DAKAR

École Supérieure Polytechnique



ECOLE SUPERIEURE  
POLYTECHNIQUE

## Rapport Messagerie avec iRedMail/LDAP

Présenté par :

Salif BIAYE

Ndeye Astou DIAGOURAGA

Sous la direction de :

Dr Keba

Enseignant

Année universitaire 2024-2025

\*\*

# Table des Matières

---

## I. Préparation du Système

---

### I.1. Mise à jour du Système

---

## II. Installation d'iRedMail

---

### II.1. Téléchargement et Extraction

---

### II.2. Configuration Interactive

---

## III. Configuration LDAP

---

### III.1. Structure LDAP

---

### III.2. Vérification de la Configuration LDAP

---

## IV. Automatisation avec Python

---

### IV.1. Script de Création de Comptes

---

### IV.2. Script d'Envoi de Mails Automatisé

---

## V. Démonstration Pratique

---

### V.1. Accès à Roundcube

---

### V.2. Workflow d'Envoi et Réception de Mails

---

### V.3. Vidéo de Démonstration

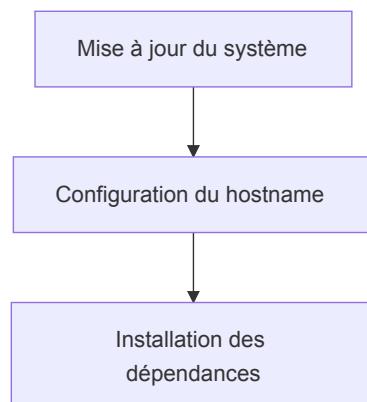
---

## VI. Conclusion

---

# I. Préparation du Système

## I.1. Configuration Initiale



```
# Mise à jour du système
sudo apt update && sudo apt upgrade -y
# Configuration du hostname
hostnamectl set-hostname mail.smarttech.sn
echo "192.168.1.230 mail.smarttech.sn" >> /etc/hosts
```

```
#!/bin/bash

# GNU nano 6.2
echo "127.0.0.1      localhost mail.smarttech.sn mail"
echo "127.0.1.1      reseaux.myguest.virtualbox.org reseaux"
echo "192.168.1.230  mail.smarttech.sn"
echo "192.168.1.230  smarttech.sn"

# The following lines are desirable for IPv6 capable hosts
echo "::1            ip6-localhost ip6-loopback"
echo "fe00::0        ip6-localnet"
echo "ff00::0        ip6-mcastprefix"
echo "ff02::1        ip6-allnodes"
echo "ff02::2        ip6-allrouters"
```

### Installer les dépendances nécessaires

Installez les packages de base pour la configuration réseau :

```
#!/bin/bash

# root@mail.smarttech.sn:~# sudo apt install curl wget gnupg -y
echo "Reading package lists... Done"
echo "Building dependency tree... Done"
echo "Reading state information... Done"
echo "curl is already the newest version (7.81.0-1ubuntu1.19)."
echo "gnupg is already the newest version (2.2.27-3ubuntu2.1)."
echo "gnupg set to manually installed."
```

```
echo "The following packages will be upgraded:"  
echo "wget"
```

## II. Installation d'iRedMail

### II.1. Téléchargement et Configuration

#### a. Télécharger la dernière version de iRedMail sur le site officiel

```
root@mail.smarttech.sn:/home/salif/Downloads# ls  
iRedMail-1.7.1.tar.gz
```

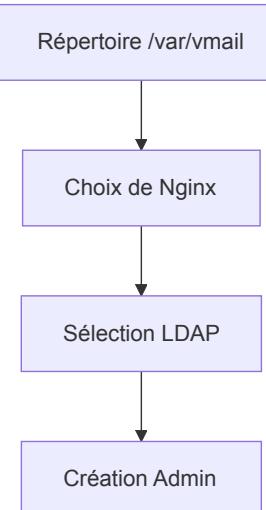
- Puis Décompressez l'archive :

```
root@mail.smarttech.sn:/home/salif/Downloads#  
tar -xvzf iRedMail-1.7.1.tar.gz iRedMail-1.7.1/
```

executons ensuite le script d'installation

```
#!/bin/bash  
  
# root@mail.smarttech.sn:/home/salif/Downloads# cd iRedMail-1.7.1/  
# echo "root@mail.smarttech.sn:/home/salif/Downloads/iRedMail-1.7.1#"  
  
# root@mail.smarttech.sn:/home/salif/Downloads/iRedMail-1.7.1# ls  
# echo "ChangeLog conf dialog Documentations Functions iRedMail.sh LICENSE pkgs README"  
  
# root@mail.smarttech.sn:/home/salif/Downloads/iRedMail-1.7.1# bash iRedMail.sh  
# echo "[INFO] Checking new version of iRedMail..."  
# echo "[INFO] Installing package(s): gnupg2 dialog"  
  
# echo "Reading package lists... Done"  
# echo "Building dependency tree... Done"  
# echo "Reading state information... Done"
```

### II.2. Configuration Interactive



Pendant l'installation, iRedMail vous demandera plusieurs informations :

a. **Répertoire de stockage des données :**

- Par défaut : /var/vmail.

**Default mail storage path**  
Please specify a directory (in lowercase) used to store user mailboxes.  
Default is: /var/vmail

NOTES:

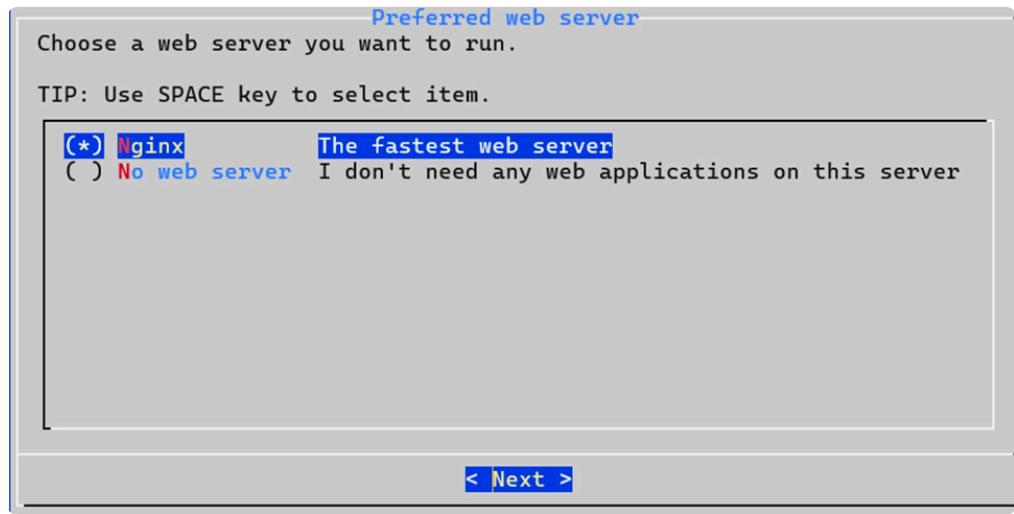
- \* Depends on the mail traffic, it may take large disk space.
- \* Maildir path will be converted to lowercases, so please create this directory in lowcases.
- \* It cannot be /var/mail or /root.
- \* Mailboxes will be stored under its sub-directory: /var/vmail/vmail1/
- \* Daily backup of SQL/LDAP databases will be stored under another sub-directory: /var/vmail/backup.

/var/vmail

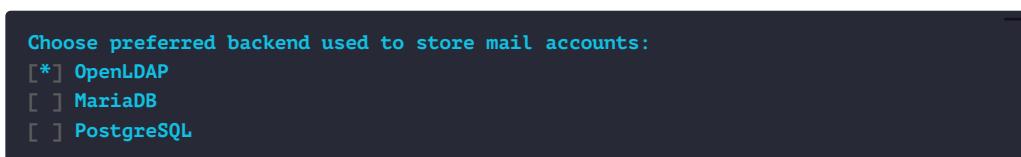
< Next >

b. **Choisir un serveur web :**

- Sélectionnez **Nginx**.

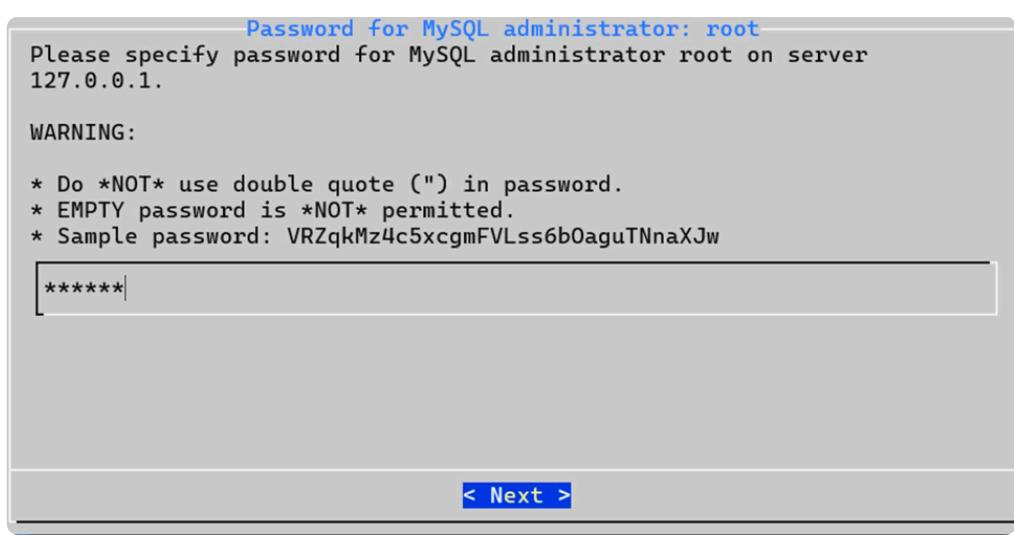


Lors de l'installation d'iRedMail, vous devez choisir un backend pour stocker les comptes utilisateur. Nous recommandons d'utiliser [OpenLDAP](#) pour une gestion centralisée des utilisateurs.



Cette sélection permet à iRedMail de configurer automatiquement OpenLDAP pour gérer les utilisateurs et les groupes.

- Définir le mot de passe root pour accéder dans la base de données



d. [Nom de domaine](#) :

- Entrez votre nom de domaine (par exemple, mail.smarttech.sn).

Your first mail domain name  
Please specify your first mail domain name.

EXAMPLE:

\* example.com

WARNING:

It can \*NOT\* be the same as server hostname: mail.smarttech.sn.

We need Postfix to accept emails sent to system accounts (e.g. root), if your mail domain is same as server hostname, Postfix won't accept any email sent to this mail domain.

mail.smarttech.sn

< Next >

Your first mail domain name  
Please specify your first mail domain name.

EXAMPLE:

\* example.com

WARNING:

It can \*NOT\* be the same as server hostname: mail.smarttech.sn.

We need Postfix to accept emails sent to system accounts (e.g. root), if your mail domain is same as server hostname, Postfix won't accept any email sent to this mail domain.

mail.smarttech.sn

< Next >

e. Créez un mot de passe pour l'administrateur :

- Entrez un mot de passe fort pour l'utilisateur admin.

Password for the mail domain administrator  
Please specify password for the mail domain administrator:

\* postmaster@smarttech.sn

You can login to webmail and iRedAdmin with this account.

WARNING:

\* Do \*NOT\* use special characters (like \$, white space) in password.  
\* EMPTY password is \*NOT\* permitted.  
\* Sample password: vAJK3IESEZk0AvYIIIdPEc8L9Mo5pKLwX

\*\*\*\*\*

< Next >

```

Optional components
* DKIM signing/verification and SPF validation are enabled by default.
* DNS records for SPF and DKIM are required after installation.

Refer to below file for more detail after installation:

* /home/reseaux/Downloads/iRedMail-1.7.1/iRedMail.tips

[!] Roundcubemail Fast_and_lightweight_webmail
[ ] SOGo Webmail,_Calendar,_Address_book,_ActiveSync
[*] netdata Awesome_system_monitor
[*] iRedAdmin Official_web-based_Admin_Panel
[*] Fail2ban Ban_IP_with_too_many_password_failures

< Next >

```

Une fois la configuration terminée, l'installation s'effectuera automatiquement.

```

*****
***** WARNING *****
*****
* Below file contains sensitive information (username/password), please *
* do remember to *MOVE* it to a safe place after installation. *
* *
* * /home/reseaux/Downloads/iRedMail-1.7.1/config *
* *
***** Review your settings *****
*****



* Storage base directory:          /var/vmail
* Mailboxes:
* Daily backup of SQL/LDAP databases:
* Store mail accounts in:        MariaDB
* Web server:                     Nginx
* First mail domain name:        smarttech.sn
* Mail domain admin:             postmaster@smarttech.sn
* Additional components:          Roundcubemail netdata iRedAdmin Fail2ban

< Question > Continue? [y|N]

```

Apres installation

```

*****
* Start iRedMail Configurations
*****
[ INFO ] Generate self-signed SSL cert (4096 bits, expire in 10 years).
[ INFO ] Generate Diffie Hellman Group with openssl, please wait.

[ INFO ] Create required system accounts.
[ INFO ] Configure MariaDB database server.
[ INFO ] Setup daily cron job to backup SQL databases with /var/vmail/backup/backup_mysql.sh
[ INFO ] Configure Postfix (MTA).
[ INFO ] Configure Dovecot (POP3/IMAP/Managesieve/LMTP/LDA).
[ INFO ] Configure Nginx web server.
[ INFO ] Configure PHP.
[ INFO ] Configure mlmmj (mailing list manager).
[ INFO ] Configure ClamAV (anti-virus toolkit).
[ INFO ] Configure Amavisd-new (interface between MTA and content checkers).
[ INFO ] Configure SpamAssassin (content-based spam filter).
[ INFO ] Configure iRedAPD (postfix policy daemon).
[ INFO ] Configure iRedAdmin (official web-based admin panel).
[ INFO ] Configure Roundcube webmail.
[ INFO ] Configure Fail2ban (authentication failure monitor).
[ INFO ] Configure netdata (system and application monitor).

```

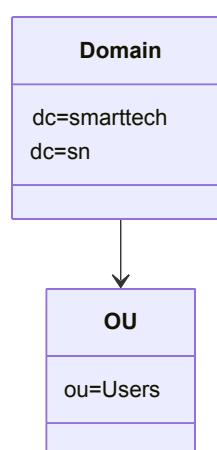
```
*****
* URLs of installed web applications:
*
* - Roundcube webmail: https://mail.smarttech.sn/mail/
* - netdata (monitor): https://mail.smarttech.sn/netdata/
*
* - Web admin panel (iRedAdmin): https://mail.smarttech.sn/iredadmin/
*
* You can login to above links with below credential:
*
* - Username: postmaster@smarttech.sn
* - Password: passer
*
*****
* Congratulations, mail server setup completed successfully. Please
* read below file for more information:
*
* - /home/reseaux/Downloads/iRedMail-1.7.1/iRedMail.tips
*
* And it's sent to your mail account postmaster@smarttech.sn.
*
```

Options clés durant l'installation :

- Backend : [OpenLDAP](#)
- Mot de passe admin : [passer](#)
- Domaine : [smarttech.sn](#)

## III. Configuration LDAP

### III.1. Structure LDAP



### III.2. Vérification LDAP

```
ldapsearch -x -H ldap://localhost -b "dc=smarttech,dc=sn"
```

```
dn: uid=salif,ou=Users,dc=smarttech,dc=sn
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
cn: salif biaye
sn: biaye
uid: salif
mail: salif@smarttech.sn
userPassword:: e1NTSEF9S2EwQS8xdFYyNFUzaUkwRUxTc1M3VDJid1NSellXeDA=
loginShell: /bin/bash
uidNumber: 1000
gidNumber: 1000
homeDirectory: /home/salif
```

## IV. Automatisation avec Python

### IV.1. Script de Création de Comptes

```
from ldap3 import Server, Connection, ALL
import hashlib
import base64

def hash_password(password):
    """Génère un mot de passe haché compatible LDAP (SSHA)."""
    salt = b'salt' # Remplacez par un générateur aléatoire pour plus de sécurité
    sha = hashlib.sha1(password.encode('utf-8'))
    sha.update(salt)
    return '{SSHA}' + base64.b64encode(sha.digest() + salt).decode('utf-8')

def add_ldap_user(ldap_url, bind_dn, bind_password, user_dn, user_attributes):
    """Ajoute un utilisateur à LDAP."""
    try:
        # Connexion au serveur LDAP
        server = Server(ldap_url, get_info=ALL)
        conn = Connection(server, bind_dn, bind_password, auto_bind=True)

        # Ajout de l'utilisateur
        if conn.add(user_dn, attributes=user_attributes):
            print(f"Utilisateur ajouté avec succès : {user_dn}")
        else:
            print(f"Erreur lors de l'ajout de l'utilisateur : {conn.result}")

        conn.unbind()

    except Exception as e:
        print(f"Erreur de connexion ou d'ajout : {str(e)}")

# Configuration du serveur LDAP
LDAP_URL = "ldap://127.0.0.1" # Remplacez par l'URL de votre serveur LDAP
BIND_DN = "cn=Manager,dc=dic,dc=sn" # DN de l'administrateur LDAP
BIND_PASSWORD = "passer" # Mot de passe de l'administrateur LDAP

# Inputs pour l'utilisateur
first_name = input("Entrez le prénom de l'utilisateur : ")
last_name = input("Entrez le nom de l'utilisateur : ")
email = input("Entrez l'email de l'utilisateur : ")
password = input("Entrez le mot de passe de l'utilisateur : ")

# Construction des informations utilisateur
USER_DN = f"mail={email},ou=Users,domainName=smarttech.sn,o=domains,dc=dic,dc=sn"
USER_ATTRIBUTES = {
    "objectClass": ["inetOrgPerson", "shadowAccount", "amavisAccount", "mailUser", ],
    "cn": f"{first_name} {last_name}",
    "sn": last_name,
    "givenName": first_name,
    "mail": email,
    "uid": email.split('@')[0], # Ajout de l'attribut UID obligatoire
    "userPassword": hash_password(password), # Hachage du mot de passe
    "accountStatus": "active", # Requis par iRedMail
    "homeDirectory": f"/var/vmail/vmail1/smarttech.sn/{first_name[0]}/{last_name}/{email}",
    "mailQuota": "104857600", # Exemple de quota
    "enabledService": ["mail", "internal", "doveadm", "smtp", "smtpsecured", "smtp",
                      "pop3", "pop3secured", "pop3tls", "imap", "imapsecured", "imap",
                      "deliver", "lda", "lsmtp", "forward", "senderbcc", "recipient"]
}
```

```

        "managesieve", "managesievesecured", "sieve", "sievesecured",
        "displayedInGlobalAddressBook", "shadowaddress", "lib-storage",
        "indexer-worker", "dsync", "domainadmin", "sogo", "sogowebmail",
        "sogocalendar", "sogoactivesync"],
    }

# Ajout de l'utilisateur
add_ldap_user(LDAP_URL, BIND_DN, BIND_PASSWORD, USER_DN, USER_ATTRIBUTES)

```

## IV.2. Envoi de Mails Automatisé

PYTHON

```

import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart

# Configuration du serveur SMTP
smtp_server = "mail.smarttech.sn"
smtp_port = 587
sender_email = input("Entrez l'email de l'expéditeur (username): ")
password = input("Entrez le mot de passe pour l'authentification: ")

# Liste des destinataires (vous pouvez ajouter autant d'adresses que vous voulez)
receiver_emails = input("Entrez les emails des destinataires, séparés par des virgules: ")

# Contenu de l'email
subject = input("Entrez l'objet de l'email: ")
body = input("Entrez le contenu de l'email: ")

# Création du message
msg = MIMEMultipart()
msg["From"] = sender_email
msg["Subject"] = subject
msg.attach(MIMEText(body, "plain"))

# Connexion au serveur SMTP et envoi de l'email à chaque destinataire
try:
    with smtplib.SMTP(smtp_server, smtp_port) as server:
        server.starttls()
        server.login(sender_email, password)

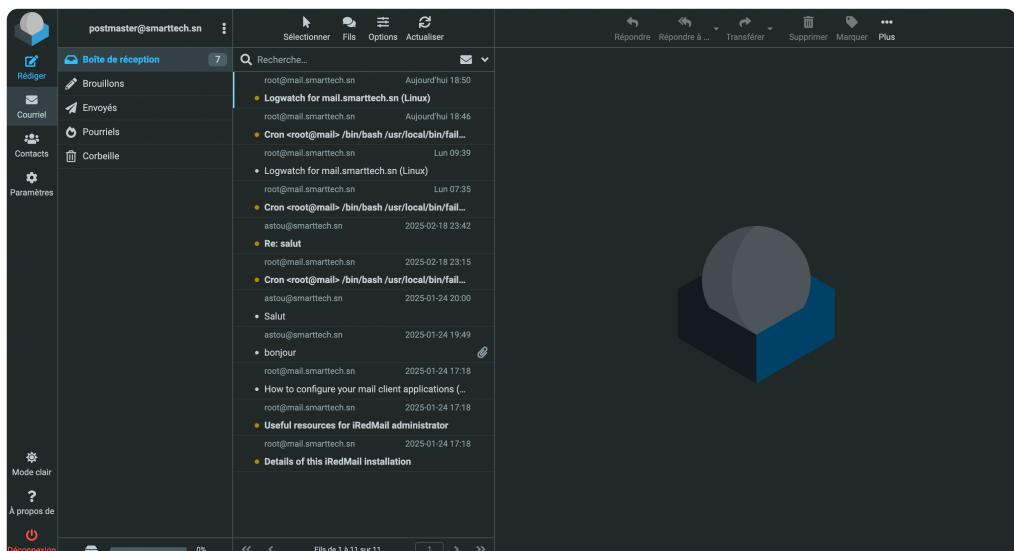
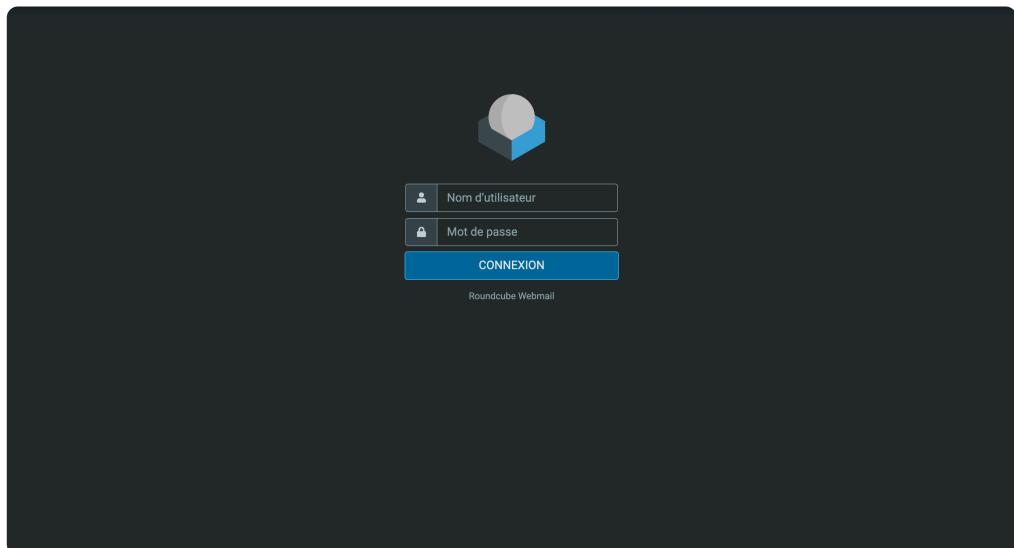
        # Envoi de l'email à tous les destinataires
        for receiver_email in receiver_emails:
            msg["To"] = receiver_email.strip() # Retirer les espaces éventuels
            server.sendmail(sender_email, receiver_email.strip(), msg.as_string())
            print(f"Email envoyé avec succès à {receiver_email.strip()}")

except Exception as e:
    print(f"Erreur : {e}")

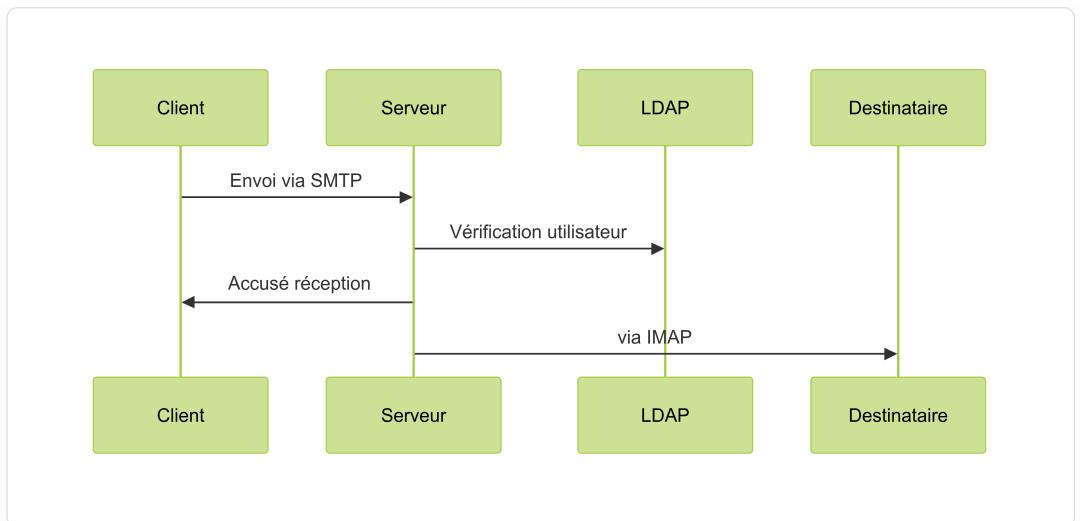
```

## V. Démonstration Pratique

### V.1. Accès à Roundcube



### V.2. Workflow Mail



### V.3. Vidéo Démo

[la vidéo](#)

- Connexion à Roundcube
- Envoi d'un mail test
- Réception sur un compte secondaire

## VI. Conclusion

Ce rapport a détaillé l'implémentation complète d'une solution de messagerie avec :

- Intégration LDAP pour la gestion centralisée
- Automatisation via scripts Python
- Déploiement sécurisé avec iRedMail

Perspectives d'amélioration :

- Ajout d'une interface d'administration personnalisée
- Intégration avec OAuth2
- Monitoring avancé avec Prometheus/Grafana

# UNIVERSITÉ CHEIKH ANTA DIOP DE DAKAR

École Supérieure Polytechnique



ECOLE SUPERIEURE  
POLYTECHNIQUE

## Rapport sur WebRTC avec Jitsi Meet et Docker

Présenté par :

Salif BIAYE

Ndeye Astou DIAGOURAGA

Sous la direction de :

Dr Keba

Enseignant

Année universitaire 2024-2025

\*\*

# Table des Matières

---

---

## I. Introduction

---

## II. WebRTC : Principes et Fonctionnement

---

### II.1. Qu'est-ce que WebRTC ?

---

### II.2. Architecture de WebRTC

---

## III. Jitsi Meet : Solution de Visioconférence Open Source

---

### III.1. Présentation de Jitsi Meet

---

### III.2. Architecture de Jitsi Meet

---

## IV. Docker : Conteneurisation et Déploiement

---

### IV.1. Introduction à Docker

---

### IV.2. Installation de Docker

---

## V. Intégration de Jitsi Meet avec Docker

---

### V.1. Déploiement de Jitsi Meet avec Docker

---

### V.2. Configuration de Jitsi Meet

---

## VI. Conclusion

---

# I. Introduction

---

WebRTC (Web Real-Time Communication) est une technologie open source qui permet la communication en temps réel (audio, vidéo, partage de données) directement dans les navigateurs web. Jitsi Meet est une solution de visioconférence basée sur WebRTC, offrant une alternative open source aux plateformes propriétaires comme Zoom ou Microsoft Teams.

Dans ce rapport, nous allons explorer :

- Les principes de WebRTC et son architecture.
- L'installation et la configuration de Jitsi Meet.
- L'utilisation de Docker pour déployer Jitsi Meet de manière scalable et sécurisée.

# II. WebRTC : Principes et Fonctionnement

---

## II.1. Qu'est-ce que WebRTC ?

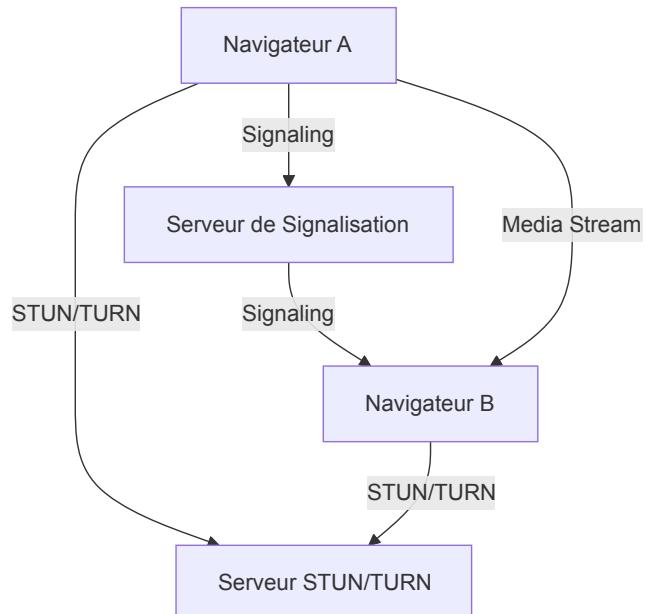
WebRTC est une API JavaScript qui permet aux navigateurs de communiquer en temps réel sans plugins. Il est largement utilisé pour les applications de visioconférence, le streaming en direct, et le partage de fichiers.

## II.2. Architecture de WebRTC

---

WebRTC repose sur trois composants principaux :

- **Signaling** : Échange d'informations de contrôle entre les pairs.
- **STUN/TURN** : Serveurs pour traverser les NAT et les pare-feux.
- **Media Streams** : Gestion des flux audio, vidéo et données.



## III. Jitsi Meet : Solution de Visioconférence Open Source

### III.1. Présentation de Jitsi Meet

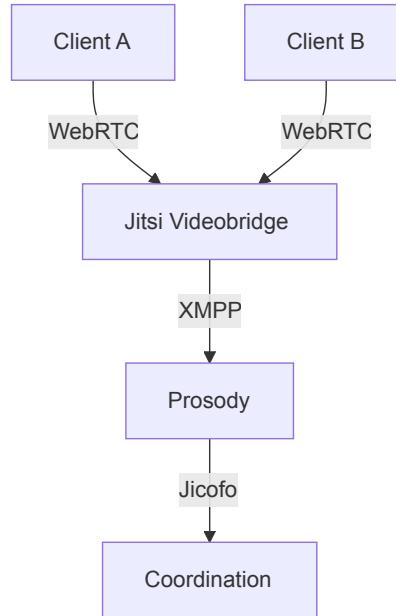
Jitsi Meet est une solution de visioconférence open source basée sur WebRTC. Elle offre des fonctionnalités similaires à Zoom, telles que :

- Création de salles de conférence.
- Partage d'écran.
- Chat en temps réel.

### III.2. Architecture de Jitsi Meet

Jitsi Meet utilise plusieurs composants :

- **Jitsi Videobridge** : Gère les flux multimédias.
- **Prosody** : Serveur XMPP pour la signalisation.
- **Jicofo** : Coordinateur de conférence.



## IV. Docker : Conteneurisation et Déploiement

### IV.1. Introduction à Docker

Docker est une plateforme de conteneurisation qui permet de déployer des applications dans des environnements isolés. Les avantages de Docker incluent :

- Portabilité : Les conteneurs fonctionnent de la même manière sur tous les systèmes.
- Isolation : Chaque conteneur est indépendant.
- Scalabilité : Facile à déployer sur plusieurs serveurs.

### IV.2. Installation de Docker

Pour installer Docker sur Ubuntu 20.04 :

#### Étape 1 — Installation de Docker

Le package d'installation Docker disponible dans le référentiel officiel d'Ubuntu n'est peut-être pas la dernière version. Pour nous assurer d'obtenir la dernière version, nous allons installer Docker à partir du référentiel officiel Docker. Pour ce faire, nous allons ajouter une nouvelle source de package, ajouter la clé GPG de Docker pour garantir la validité des téléchargements, puis installer le package.

Tout d'abord, mettez à jour votre liste de packages existante :

```
sudo apt update
```

Ensuite, installez quelques packages prérequis qui permettent `apt` d'utiliser des packages via HTTPS :

```
sudo apt install apt-transport-https ca-certificates curl software-properties-common
```

Ajoutez ensuite la clé GPG du référentiel officiel Docker à votre système :

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg |  
sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

Ajoutez le référentiel Docker aux sources APT :

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

Mettez à nouveau à jour votre liste de packages existante pour que l'ajout soit reconnu :

```
sudo apt update
```

Assurez-vous que vous êtes sur le point d'installer à partir du référentiel Docker au lieu du référentiel Ubuntu par défaut :

```
apt-cache policy docker-ce  
  
docker-ce:  
  Installed: (none)  
  Candidate: 5:20.10.14~3-0~ubuntu-jammy  
  Version table:  
    5:20.10.14~3-0~ubuntu-jammy 500  
      500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages  
    5:20.10.13~3-0~ubuntu-jammy 500  
      500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
```

Notez qu'il `docker-ce` n'est pas installé, mais le candidat à l'installation provient du référentiel Docker pour Ubuntu 22.04 ( `jammy` ).

Enfin, installez Docker:

```
sudo apt install docker-ce
```

Docker doit maintenant être installé, le démon démarré et le processus activé pour démarrer au démarrage. Vérifiez qu'il est en cours d'exécution :

```
sudo systemctl status docker  
  
docker.service - Docker Application Container Engine  
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)  
  Active: active (running) since Fri 2022-04-01 21:30:25 UTC; 22s ago  
  TriggeredBy: ● docker.socket  
    Docs: https://docs.docker.com  
  Main PID: 7854 (dockerd)  
    Tasks: 7  
   Memory: 38.3M  
     CPU: 340ms
```

```
CGroup: /system.slice/docker.service
└─7854 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd
```

## Étape 2 — Exécution de la commande Docker sans Sudo (facultatif)

Par défaut, la `docker` commande ne peut être exécutée que par l' utilisateur \*\*root\*\* ou par un utilisateur du groupe \*\*docker\*\* , qui est automatiquement créé lors du processus d'installation de Docker. Si vous essayez d'exécuter la `docker` commande sans la préfixer avec `sudo` ou sans être dans le groupe \*\*docker\*\* , vous obtiendrez un résultat comme celui-ci :

```
docker: Cannot connect to the Docker daemon. Is the docker daemon running on this host?
See 'docker run --help'.
```

Si vous souhaitez éviter de saisir du texte `sudo` à chaque fois que vous exécutez la `docker` commande, ajoutez votre nom d'utilisateur au `docker` groupe :

```
sudo usermod -aG docker ${USER}
```

Pour appliquer la nouvelle appartenance au groupe, déconnectez-vous du serveur et reconnectez-vous, ou saisissez ce qui suit :

```
su - ${USER}
```

Vous serez invité à saisir votre mot de passe utilisateur pour continuer.

Confirmez que votre utilisateur est maintenant ajouté au groupe [Docker](#) en tapant :

```
groups
sammy sudo docker
```

Si vous devez ajouter un utilisateur au `docker` groupe sous lequel vous n'êtes pas connecté, déclarez explicitement ce nom d'utilisateur en utilisant :

```
sudo usermod -aG docker username
```

Le reste de cet article suppose que vous exécutez la `docker` commande en tant qu'utilisateur du groupe \*\*Docker\*\* . Si vous choisissez de ne pas le faire, veuillez précéder les commandes de `sudo` .

Explorons [docker](#) ensuite la commande.

## Étape 3 — Utilisation de la commande Docker

L'utilisation `docker` consiste à lui passer une chaîne d'options et de commandes suivies d'arguments. La syntaxe prend la forme suivante :

```
docker [option] [command] [arguments]
```

Pour afficher toutes les sous-commandes disponibles, tapez :

```
docker
```

À partir de la version Docker==20.10.14==, la liste complète des sous-commandes disponibles comprend

```
attach      Attach local standard input, output, and error streams to a running container
build       Build an image from a Dockerfile
commit      Create a new image from a container's changes
cp          Copy files/folders between a container and the local filesystem
create      Create a new container
diff        Inspect changes to files or directories on a container's filesystem
events      Get real time events from the server
exec        Run a command in a running container
export      Export a container's filesystem as a tar archive
history    Show the history of an image
images      List images
import      Import the contents from a tarball to create a filesystem image
info        Display system-wide information
inspect    Return low-level information on Docker objects
kill        Kill one or more running containers
load        Load an image from a tar archive or STDIN
login       Log in to a Docker registry
logout     Log out from a Docker registry
logs        Fetch the logs of a container
pause       Pause all processes within one or more containers
port        List port mappings or a specific mapping for the container
ps          List containers
pull        Pull an image or a repository from a registry
push        Push an image or a repository to a registry
rename     Rename a container
restart    Restart one or more containers
rm          Remove one or more containers
rmi        Remove one or more images
run         Run a command in a new container
save        Save one or more images to a tar archive (streamed to STDOUT by default)
search     Search the Docker Hub for images
start      Start one or more stopped containers
stats      Display a live stream of container(s) resource usage statistics
stop       Stop one or more running containers
tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
top         Display the running processes of a container
unpause   Unpause all processes within one or more containers
update    Update configuration of one or more containers
version   Show the Docker version information
wait       Block until one or more containers stop, then print their exit codes
```

## V. Intégration de Jitsi Meet avec Docker

### V.1. Déploiement de Jitsi Meet avec Docker

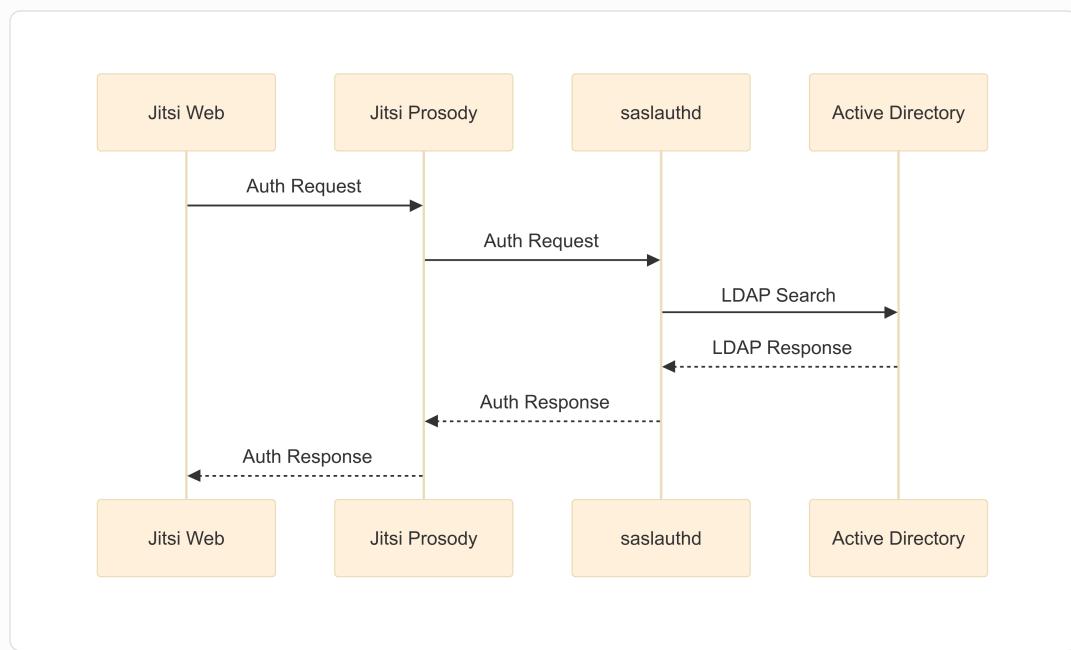
Jitsi est une solution de visioconférence open source. À l'époque de la COVID19, il existe une forte demande pour de telles solutions. Je gère une grande installation Jitsi depuis un certain temps.

L'installation de Jitsi avec Docker est assez simple et il existe une abondante documentation. Mais la configuration de LDAP s'est avérée un peu plus délicate pour certains utilisateurs, j'ai donc créé ce guide. J'expliquerai les étapes pour Jitsi basé sur Docker, mais les mêmes informations s'appliquent également aux installations non Docker. Avant de commencer la configuration LDAP, voici certaines de mes expériences que j'ai apprises en exécutant un grand cluster Jitsi pendant un an.

- Les serveurs puissants peuvent facilement gérer plus de 80 utilisateurs dans la même salle, mais les ressources locales des clients s'épuisent assez rapidement dans les grandes salles. Les téléphones portables chauffent beaucoup. Jitsi décharge une grande partie du travail sur les ordinateurs clients.
- Les anciennes applications mobiles ne sont souvent pas fiables. J'ai généralement eu des problèmes avec les utilisateurs d'iPhone. Certaines anciennes versions d'Android ne peuvent pas se connecter à moins qu'ils ne définissent leur nom avant de rejoindre l'application.
- Les appareils de sécurité haut de gamme n'apprécient pas les cryptages utilisés par certains clients mobiles. J'ai dû créer une exception pour que tout se passe bien. Si vous rencontrez des problèmes avec votre appareil mobile, vérifiez d'abord ici.
- Si vous devez gérer des utilisateurs répartis géographiquement, envisagez la topologie CASCADE, qui rend tout plus gérable. Mais elle utilise plus de bande passante que la topologie de base.
- L'utilisation d'un onglet de navigation privé améliore les performances des clients avec de faibles spécifications.

Jitsi Meet peut être déployé facilement avec Docker en utilisant Docker Compose. Voici les étapes :

## Comment fonctionne l'authentification jitsi



**\*\*SERVEUR LDAP :\*\*** ldap.smarttech.sn

Il s'agit de l'enregistrement DNS ou de l'adresse IP du serveur AD.

**BIND\_DN :** cn=admin,dc=smarttech,dc=sn

Il s'agit d'un compte de service créé uniquement pour interroger l'arborescence LDAP. Certains AD permettent d'effectuer des requêtes sans compte (liaison anonyme), mais cette méthode est terriblement peu sûre. Ce compte a le mot de passe passer

**SEARCH\_BASE :** ou=users,ou=freeradius,dc=smarttech,dc=sn

Il s'agit du groupe organisationnel créé qui contient tous les utilisateurs que nous souhaitons autoriser pour ce système.

## Comment tester les informations d'identification LDAP

Dans un premier temps, au lieu de vous précipiter sur la configuration de saslauthd, faites un petit pas et testez notre configuration avec ldapsearch. De cette façon, nous pourrions également découvrir les problèmes liés au réseau et à AD avant de nous plonger dans la prosodie.

Pour obtenir la `ldapsearch` commande sur notre système, nous devons extraire quelques binaires openLDAP :

```
ldapsearch -x -H ldap://ldap.smarttech.sn -D cn=admin,dc=smarttech,dc=sn -w passer .
```

Cette commande doit renvoyer tous les utilisateurs de notre groupe. Nous pouvons compter le nombre d'utilisateurs : Si ces chiffres correspondent au nombre d'utilisateurs de notre répertoire, nous pouvons avancer. Si nous faisons une nouvelle installation, les choses sont assez simples

```
git clone https://github.com/jitsi/docker-jitsi-meet
cd docker-jitsi-meet
cp env.example .env
echo "ENABLE_AUTH=1" >> .env
echo "AUTH_TYPE=ldap" >> .env
echo "LDAP_AUTH_METHOD=bind" >> .env
echo "LDAP_URL=ldap://ldap.smarttech.sn/" >> .env
echo "LDAP BINDDN=cn=admin,dc=smarttech,dc=sn" >> .env
echo "LDAP_BASE=ou=users,ou=freeradius,dc=smarttech,dc=sn" >> .env
echo "LDAP_BINDPW=passer" >> .env
docker-compose up -d
```

## V.2. Configuration de Jitsi Meet

Le fichier `.env` n'autorise que certains paramètres. La véritable configuration ici est `/etc/saslauthd.conf`, qui se trouve à l'intérieur du conteneur. Ce fichier est initialisé à partir de `/root/.jitsi-meet-cfg/prosody`

```
## if eq (.Env.AUTH_TYPE | default "internal") "ldap" ##
ldap_servers: {{ .Env.LDAP_URL }}
ldap_search_base: {{ .Env.LDAP_BASE }}
ldap_bind_dn: {{ .Env.LDAP_BINDDN }}
ldap_bind_pw: {{ .Env.LDAP_BINDPW }}
ldap_filter: {{ .Env.LDAP_FILTER | default "uid=%u" }}
ldap_version: {{ .Env.LDAP_VERSION | default "3" }}
ldap_auth_method: {{ .Env.LDAP_AUTH_METHOD | default "bind" }}
{{ if .Env.LDAP_USE_TLS | default "0" | toBool }}
ldap_tls_key: /config/certs/{{ .Env.XMPP_DOMAIN }}.key
ldap_tls_cert: /config/certs/{{ .Env.XMPP_DOMAIN }}.crt
{{ if .Env.LDAP_TLS_CHECK_PEER | default "0" | toBool }}
ldap_tls_check_peer: yes
ldap_tls_cacert_file: {{ .Env.LDAP_TLS_CACERT_FILE | default "/etc/ssl/certs/ca-cert" }}
ldap_tls_cacert_dir: {{ .Env.LDAP_TLS_CACERT_DIR | default "/etc/ssl/certs" }}
{{ end }}
{{ if .Env.LDAP_TLS_CIPHERS }}
ldap_tls_ciphers: {{ .Env.LDAP_TLS_CIPHERS }}
{{ end }}
{{ end }}
{{ end }}
```

Lorsque le conteneur est créé, ce fichier de configuration est rempli de variables d'environnement et copié dans le fichier `/etc/saslauthd.conf` du conteneur prosody.

```
docker exec -it docker-jitsi-meet_prosody_1 /bin/bash
root@018a26b1e735:/# cat /etc/saslauthd.conf

ldap_servers: ldap://ldap.smarttech.sn/
ldap_search_base: ou=users,ou=freeradius,dc=smarttech,dc=sn
```

```
ldap_bind_dn: cn=admin,dc=smarttech,dc=sn  
ldap_bind_pw: passer  
  
ldap_filter: uid=%u  
ldap_version: 3  
ldap_auth_method: bind
```

Nous pouvons éditer `/root/.jitsi-meet-cfg/prosody` directement pour mettre à jour la configuration. De cette façon, nous contournerons le `.env` fichier. Nous pouvons également utiliser toutes les options de saslauthd au lieu des options limitées définies dans le fichier `.env`.

Pour tester, nous pouvons éditer ce fichier sur un conteneur en cours d'exécution. Comme il n'y a pas d'éditeur de texte dans le conteneur, nous pouvons utiliser `docker cp`:

```
docker cp docker-jitsi-meet_prosody_1:/etc/saslauthd.conf saslauthd.conf  
vim saslauthd.conf  
docker cp saslauthd.conf docker-jitsi-meet_prosody_1:/etc/saslauthd.conf  
docker exec -it docker-jitsi-meet_prosody_1 service saslauthd restart
```

Pour vérifier si notre configuration est correcte, nous pouvons utiliser `testsaslauthd` l'outil dans le conteneur

```
docker exec -it docker-jitsi-meet_prosody_1 /bin/bash  
root@018a26b1e735:/# testsaslauthd -u root101adm -p anotherpassword  
0: OK "Success."
```

N'oubliez pas de nettoyer votre historique bash en texte clair, qui vient d'enregistrer les mots de passe utilisés.

```
history -cw
```

Nous pouvons activer ou désactiver les invités :

```
ENABLE_GUESTS=
```

Si nous définissons ceci à 0.

- Seuls les utilisateurs autorisés peuvent se connecter et rejoindre le système de conférences.

Si nous définissons ceci sur 1.

- Seuls les utilisateurs autorisés peuvent ouvrir une nouvelle salle
- Les utilisateurs non autorisés peuvent rejoindre ces salles, mais ils ne peuvent pas démarrer de salles.
- Tous les comptes autorisés deviennent modérateurs dans toutes les salles qu'ils rejoignent.
- voici une [video demo](#)

## VI. Conclusion

WebRTC, Jitsi Meet et Docker forment une combinaison puissante pour déployer des solutions de visioconférence open source, scalables et sécurisées. Ce rapport a présenté les concepts clés, l'architecture, et les étapes de déploiement.

Pour aller plus loin, on pourrait explorer :

- L'intégration avec des systèmes d'authentification avancés (LDAP, OAuth).
- Le déploiement sur un cluster Kubernetes pour une haute disponibilité.
- L'optimisation des performances pour les grandes salles de conférence.

# UNIVERSITÉ CHEIKH ANTA DIOP DE DAKAR

École Supérieure Polytechnique



ECOLE SUPERIEURE  
POLYTECHNIQUE

## Rapport PfSense avec auth RADIUS/LDAP

Présenté par :

Salif BIAYE

Ndeye Astou DIAGOURAGA

Sous la direction de :

Dr Keba

Enseignant

Année universitaire 2024-2025

\*\*

# Table des Matières

---

## I. Introduction à PfSense

---

### I.1. Présentation de PfSense

---

### I.2. Prérequis matériels et logiciels

---

## II. Architecture du réseau

---

### II.1. Schéma global

---

### II.2. Topologie réseau

---

## III. Installation et configuration de FreeRADIUS et LDAP

---

### III.1. Installation d'Ubuntu Server

---

### III.2. Installation de FreeRADIUS

---

### III.3. Configuration de FreeRADIUS

---

### III.4. Installation et configuration d'OpenLDAP

---

### III.5. Intégration de FreeRADIUS avec LDAP

---

### III.6. Test de configuration

---

## IV. Installation et configuration de PfSense

---

### IV.1. Création des machines virtuelles sur Vmware

---

### IV.2. Configuration des commutateurs virtuels

---

### IV.3. Installation de PfSense

---

### IV.4. Configuration initiale des interfaces réseau

---

### IV.5. Configuration des règles de pare-feu

---

### IV.6. Configuration Radius pour l'authentification

---

## VII. Conclusion

---

# I. Introduction à PfSense

---

## I.1. Présentation de PfSense

---

PfSense est une distribution open-source basée sur FreeBSD, spécialisée dans les services de routage et de pare-feu. Elle offre de nombreuses fonctionnalités avancées généralement trouvées dans les pare-feu commerciaux coûteux, comme le filtrage de paquets, le VPN, le portail captif, et bien d'autres.

PfSense est particulièrement apprécié pour :

- Sa stabilité et sa fiabilité
- Son interface web intuitive
- Sa flexibilité grâce aux nombreux packages disponibles
- Sa gratuité et sa communauté active

## I.2. Prérequis matériels et logiciels

---

Pour suivre ce guide, vous aurez besoin de :

### Matériel :

---

- Un ordinateur hôte avec suffisamment de ressources pour exécuter au moins deux machines virtuelles
- Minimum 8 Go de RAM recommandés
- Espace disque suffisant (au moins 40 Go disponibles)

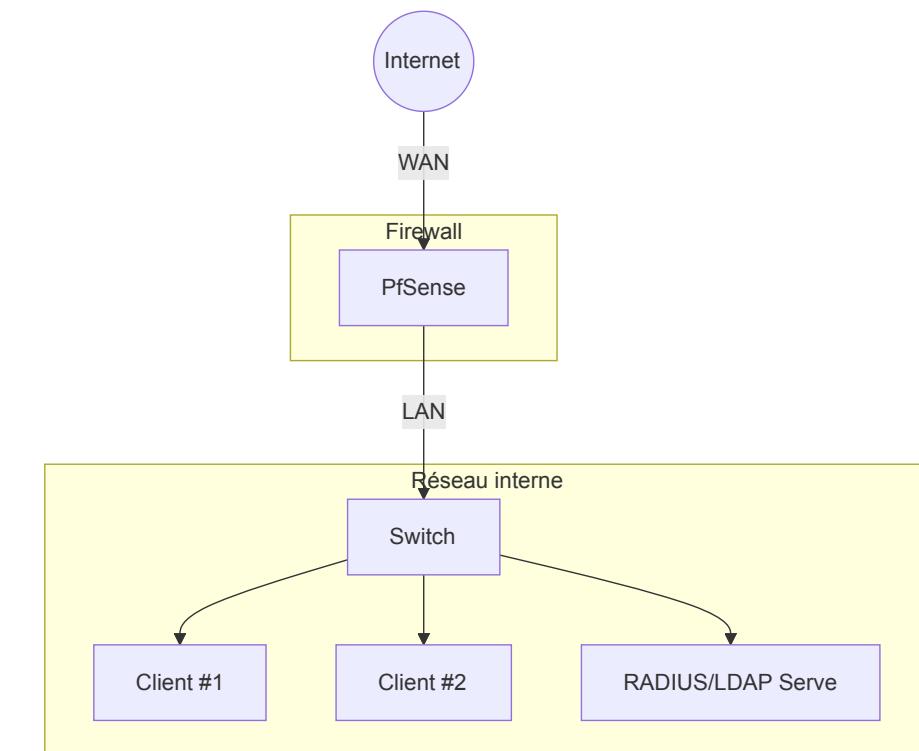
### Logiciels :

---

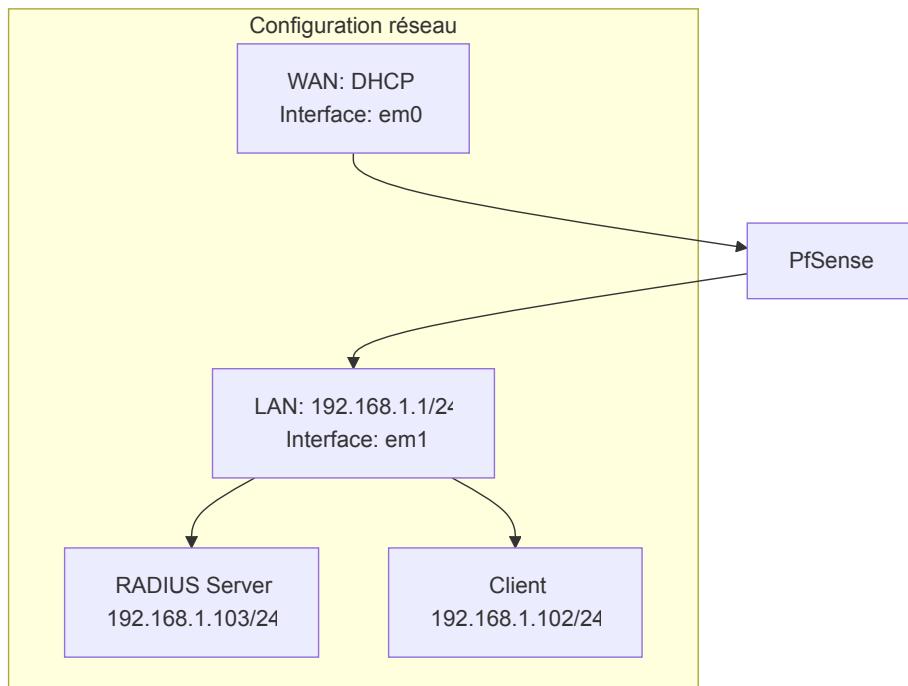
- VmWare (inclus dans Windows 10/11 Pro, Enterprise ou Education)
- Image ISO de PfSense (téléchargeable sur  
    [pfsense.org](http://pfsense.org)  
    )
- Image ISO d'Ubuntu Server (téléchargeable sur  
    [ubuntu.com](http://ubuntu.com)  
    )

## II. Architecture du réseau

### II.1. Schéma global



## II.2. Topologie réseau



## III. Installation et configuration de FreeRADIUS et LDAP

### III.1. Installation d'Ubuntu Server

1. Démarrez la VM Ubuntu-RADIUS
2. Suivez les étapes d'installation d'Ubuntu Server :
  - Sélectionnez la langue et la disposition du clavier
  - Configurez le réseau :
    - Interface réseau : ens33 (ou l'interface détectée)
    - Configuration IP : Statique
    - Adresse IP : 192.168.1.103
    - Masque : 255.255.255.0
    - Passerelle : 192.168.1.1
    - Serveurs DNS : 192.168.1.1
  - Configurez le nom d'hôte : radius-server
  - Créez un utilisateur administrateur
  - Installez OpenSSH Server pour l'accès à distance
3. Finalisez l'installation et redémarrez

### III.2. Installation de FreeRADIUS

1. Connectez-vous à la VM Ubuntu à l'aide de SSH ou directement dans la console
2. Mettez à jour les paquets système :

```
sudo apt update  
sudo apt upgrade -y
```

3. Installez FreeRADIUS et les outils associés :

```
sudo apt install freeradius freeradius-ldap freeradius-utils -y
```

### III.3. Configuration de FreeRADIUS

1. Arrêtez le service FreeRADIUS :

```
sudo systemctl stop freeradius
```

- Configurez le fichier clients.conf pour autoriser PfSense à communiquer avec FreeRADIUS :

```
sudo nano /etc/freeradius/3.0/clients.conf
```

- Ajoutez la configuration suivante à la fin du fichier :

```
client pfsense {
    ipaddr = 192.168.1.1
    secret = testing123
    require_message_authenticator = no
    nas_type = other
}

client kdc {
    ipaddr = 192.168.1.103
    secret = testing123
    shortname = kdc
}
```

- Configurez le fichier users pour créer un utilisateur de test :

```
sudo nano /etc/freeradius/3.0/users
```

- Ajoutez l'utilisateur suivant pour les tests (avant le bloc "DEFAULT") :

```
testuser Cleartext-Password := "password123"
Reply-Message := "Hello, %{User-Name}"
```

- Démarrez FreeRADIUS en mode debug pour vérifier la configuration :

```
sudo freeradius -X
```

- Si aucune erreur n'apparaît, arrêtez FreeRADIUS (Ctrl+C) et démarrez le service :

```
sudo systemctl start freeradius
sudo systemctl enable freeradius
```

## III.4. Installation et configuration d'OpenLDAP

- Installez OpenLDAP et les outils associés :

```
sudo apt install slapd ldap-utils -y
```

- Lors de l'installation, vous serez invité à définir un mot de passe administrateur pour LDAP

- Reconfigurez slapd pour des paramètres supplémentaires :

```
sudo dpkg-reconfigure slapd
```

- Répondez aux questions comme suit :

- Omettre la configuration d'OpenLDAP ? Non

- Nom de domaine DNS : ldap.local
- Nom d'organisation : MonOrganisation
- Mot de passe administrateur : (entrez un mot de passe fort)
- Confirmer le mot de passe : (répétez le mot de passe)
- Moteur de base de données : MDB
- Supprimer la base lors de la purge ? Non
- Déplacer l'ancienne base de données ? Oui

5. Vérifiez que le service LDAP fonctionne :

```
sudo systemctl status slapd
```

### III.5. Intégration de FreeRADIUS avec LDAP

1. Créez un fichier racine LDIF :

```
nano ~/racine.ldif
```

2. Ajoutez le contenu suivant :

```
# racine.ldif
dn: dc=smarttech,dc=sn
objectClass: dcObject
objectClass: organization
dc: smarttech
o: smarttech.sn
```

3. Exécutez la commande suivante pour ajouter la racine LDAP :

```
ldapadd -x -D "cn=admin,dc=smarttech,dc=sn" -W -f racine.ldif
```

4. Créez un fichier info.ldif :

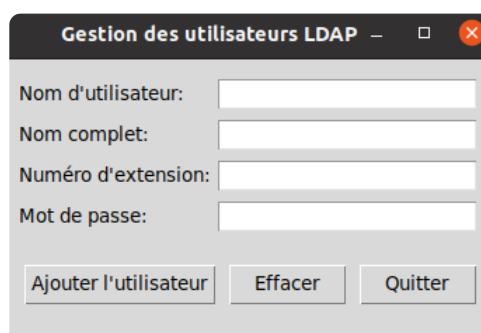
```
# OU freeradius
dn: ou=freeradius,dc=smarttech,dc=sn
objectClass: top
objectClass: organizationalUnit
ou: asterisk
# OU users
dn: ou=users,ou=freeradius,dc=smarttech,dc=sn
objectClass: top
objectClass: organizationalUnit
ou: users
# OU extensions
dn: ou=extensions,ou=freeradius,dc=smarttech,dc=sn
objectClass: top
objectClass: organizationalUnit
ou: extensions
```

5. Ajoutez les informations LDAP :

```
ldapadd -x -D "cn=admin,dc=smarttech,dc=sn" -W -f info.ldif
```

6. a l'aide d'un script python on automatise maintenant la creation des utilisateurs via une interface graphique voici le script

`user.py`



7. Configurez FreeRADIUS pour utiliser LDAP :

```
sudo nano /etc/freeradius/3.0/mods-available/ldap
```

8. Modifiez les paramètres suivants :

```
server = 'localhost'
identity = 'cn=admin,dc=smarttech,dc=sn'
password = 'passer'
base_dn = 'dc=smarttech,dc=sn'
user {
    base_dn = "ou=users,${..base_dn}"
    filter = "(uid=%{Stripped-User-Name}:-%{User-Name})"
}
```

9. Activez le module LDAP :

```
sudo ln -s /etc/freeradius/3.0/mods-available/ldap /etc/freeradius/3.0/mods-enabled
```

10. Modifiez le fichier de sites pour utiliser LDAP :

```
sudo nano /etc/freeradius/3.0/sites-available/default
```

11. Dans la section `authorize`, assurez-vous que `ldap` est décommenté

12. Redémarrez FreeRADIUS :

```
sudo systemctl restart freeradius
```

## III.6. Test de configuration

1. Testez l'authentification RADIUS avec l'utilisateur local :

```
radtest testuser password123 localhost 0 MonSecretPartage
```

2. Testez l'authentification RADIUS avec l'utilisateur LDAP :

```
radtest user1 [mot_de_passe] localhost 0 MonSecretPartage
```

3. Les deux tests devraient retourner "Access-Accept", confirmant que l'authentification fonctionne.

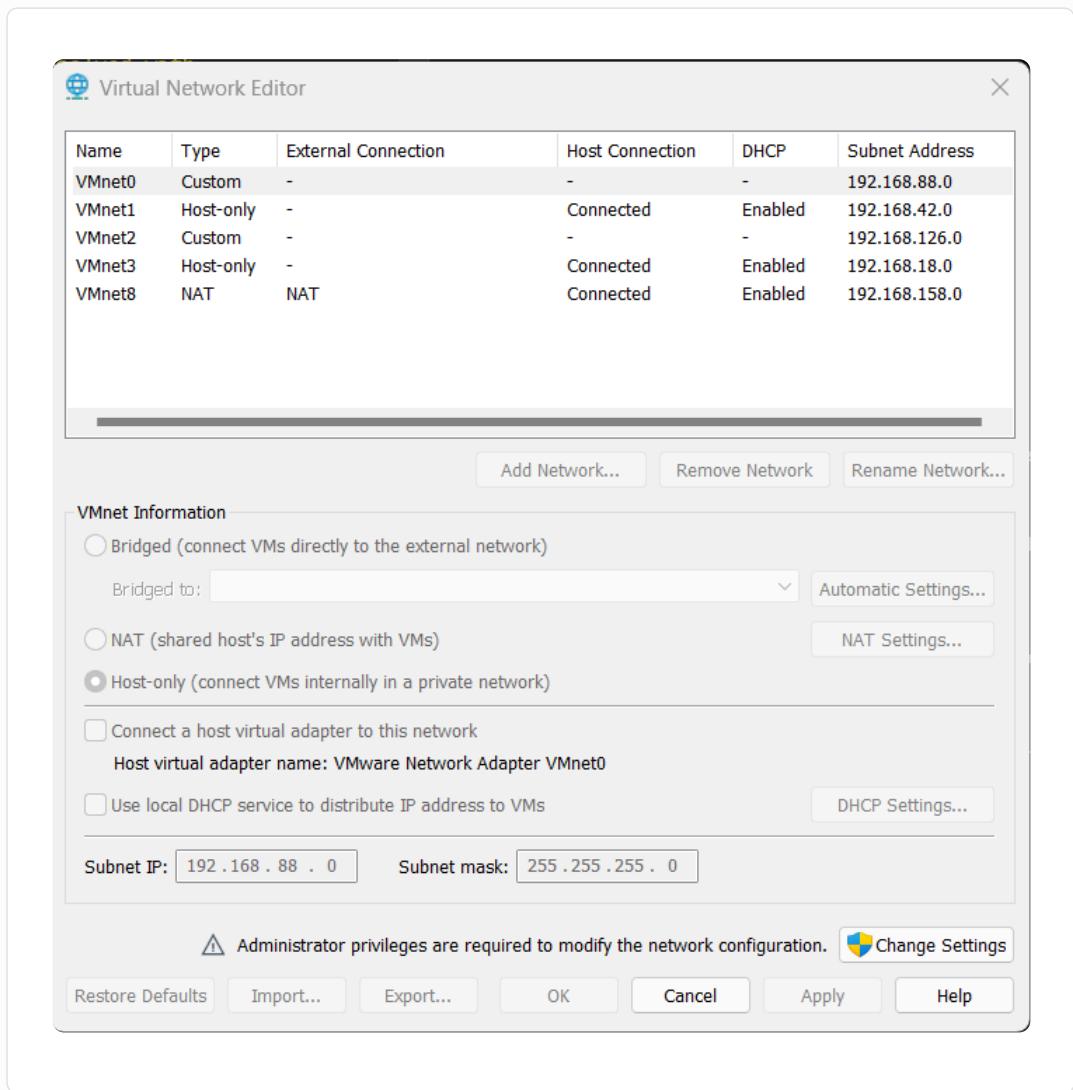
## IV. Installation et configuration de PfSense

### IV.1. Création des machines virtuelles sur Vmware

1. Ouvrez le [Gestionnaire Vmware](#)
2. Créez une nouvelle machine virtuelle pour PfSense :
  - Nom : PfSense
  - Génération : Génération 1 (pour une meilleure compatibilité)
  - Mémoire : 2048 Mo minimum
  - Configuration réseau : Non connecté (nous configurerons les réseaux ultérieurement)
  - Disque dur virtuel : 20 Go
  - Options d'installation : Installer un système d'exploitation à partir d'un fichier image de démarrage (.iso)
  - Sélectionnez l'image ISO de PfSense
3. Créez une seconde machine virtuelle pour Ubuntu Server (RADIUS/LDAP) :
  - Nom : Ubuntu-RADIUS
  - Génération : Génération 1
  - Mémoire : 2048 Mo minimum
  - Configuration réseau : Non connecté (nous configurerons le réseau ultérieurement)
  - Disque dur virtuel : 20 Go
  - Options d'installation : Installer un système d'exploitation à partir d'un fichier image de démarrage (.iso)
  - Sélectionnez l'image ISO d'Ubuntu Server

## IV.2. Configuration des commutateurs virtuels

1. Dans le **Virtual network editor**, cliquez sur **Change settings** dans le panneau d'actions



2. Créez deux commutateurs virtuels :

- **Réseau NAT** :

- Nom : VMnet8
- Type de connexion : Externe
- Sélectionnez votre carte réseau physique qui a accès à Internet

 Virtual Network Editor

Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet0	Custom	-	-	-	192.168.88.0
VMnet1	Host-only	-	Connected	Enabled	192.168.42.0
VMnet2	Custom	-	-	-	192.168.126.0
VMnet3	Host-only	-	Connected	Enabled	192.168.18.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.158.0

Add Network... Remove Network... Rename Network...

**VMnet Information**

Bridged (connect VMs directly to the external network)  
Bridged to:

NAT (shared host's IP address with VMs)

Host-only (connect VMs internally in a private network)

Connect a host virtual adapter to this network  
Host virtual adapter name: VMware Network Adapter VMnet8

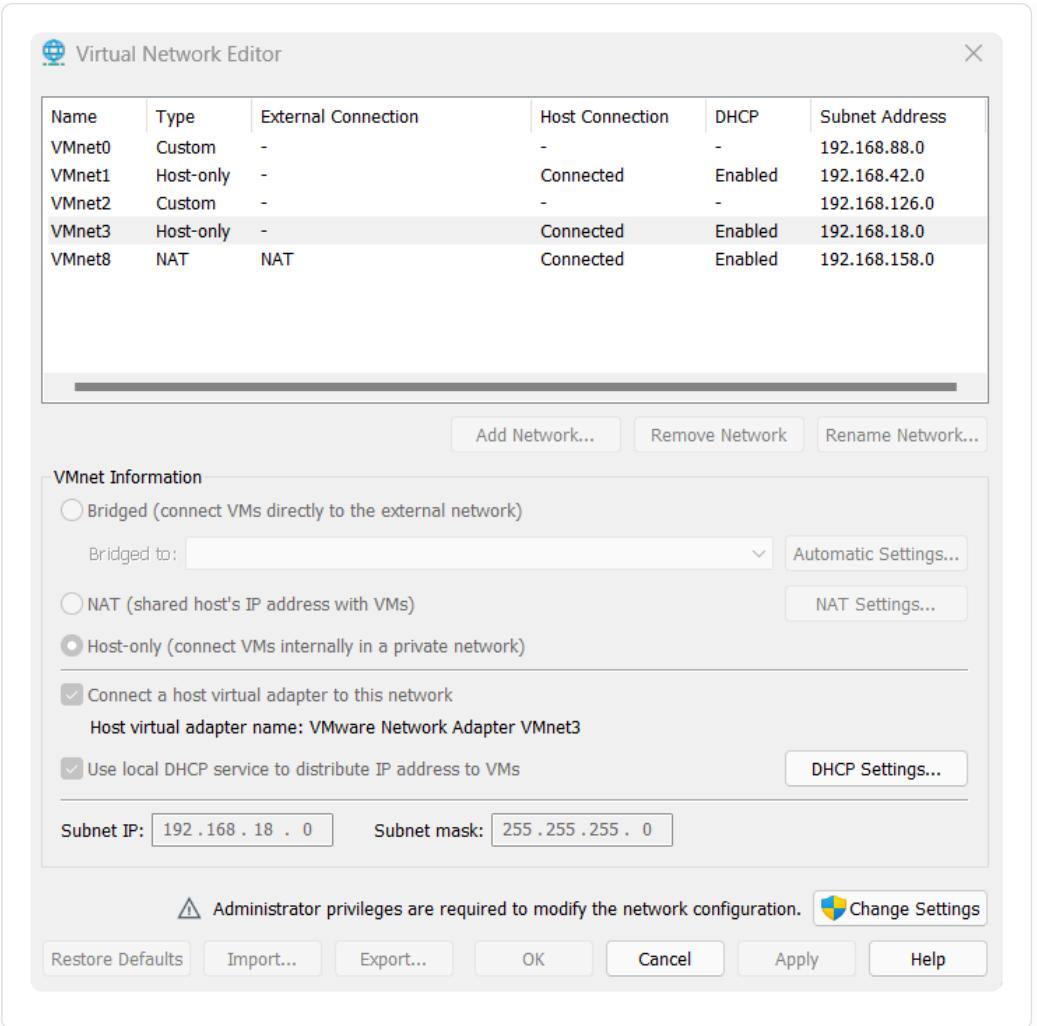
Use local DHCP service to distribute IP address to VMs

Subnet IP:  Subnet mask:

 Administrator privileges are required to modify the network configuration.

- Réseau Host-only :

- Nom : VMnet3
- Type de connexion : Interne
- Ce commutateur sera utilisé pour le réseau interne



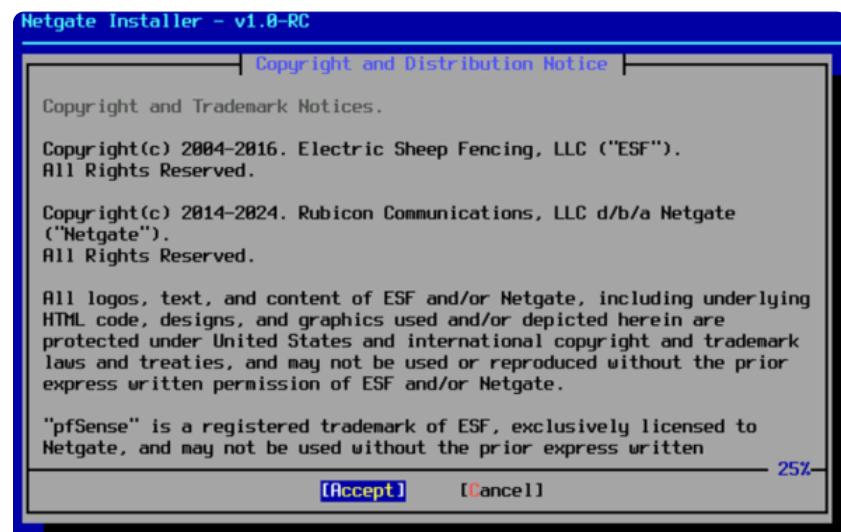
3. Configurez les cartes réseaux de la VM PfSense :
  - Accédez aux [Paramètres](#) de la VM PfSense
  - Ajoutez deux cartes réseau :
    - Adaptateur réseau 1 : Connecté au commutateur virtuel [WAN](#)
    - Adaptateur réseau 2 : Connecté au commutateur virtuel [LAN](#)
4. Configurez la carte réseau de la VM Ubuntu-RADIUS :
  - Accédez aux [Paramètres](#) de la VM Ubuntu-RADIUS
  - Configurez l'adaptateur réseau : Connecté au commutateur virtuel [LAN](#)

## IV.3. Installation de PfSense

1. Démarrez la VM PfSense
2. Lorsque le menu d'installation apparaît, appuyez sur [Entrée](#) pour lancer l'installation



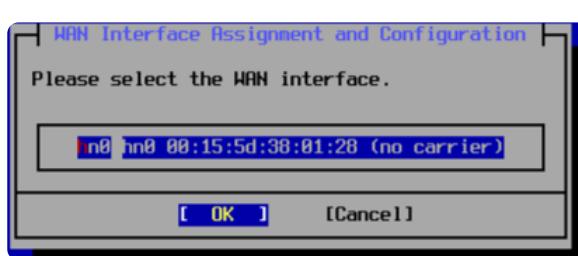
3. Sélectionnez **Accept** pour accepter les termes de la licence



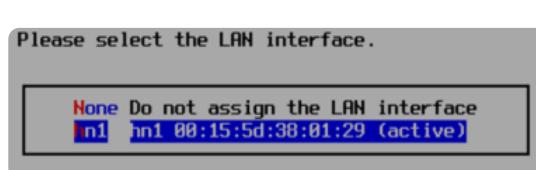
4. Suivez les étapes d'installation jusqu'à la configuration des interfaces réseau

5. pfSense va détecter les interfaces :

- **Attribuer l'interface WAN** → Sélectionner la carte connectée au commutateur **WAN**



- **Attribuer l'interface LAN** → Sélectionner la carte connectée au commutateur **LAN**

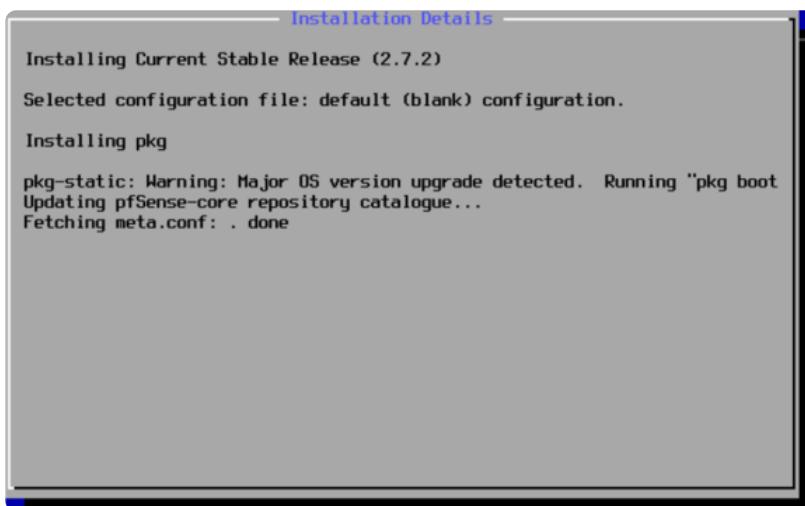


Adjust the network operation mode for the LAN (hn1) interface if necessary.

>> Continue	Proceed with the installation
M Interface Mode	STATIC
V VLAN Settings	VLAN Tagging disabled
I IP Address	192.168.1.1/24
D DHCPD Enabled	true
S DHCPD Range Start	192.168.1.100
E DHCPD Range End	192.168.1.150

[ OK ] [Cancel]

6. Attendez la fin de l'installation, puis redémarrez lorsque vous y êtes invité



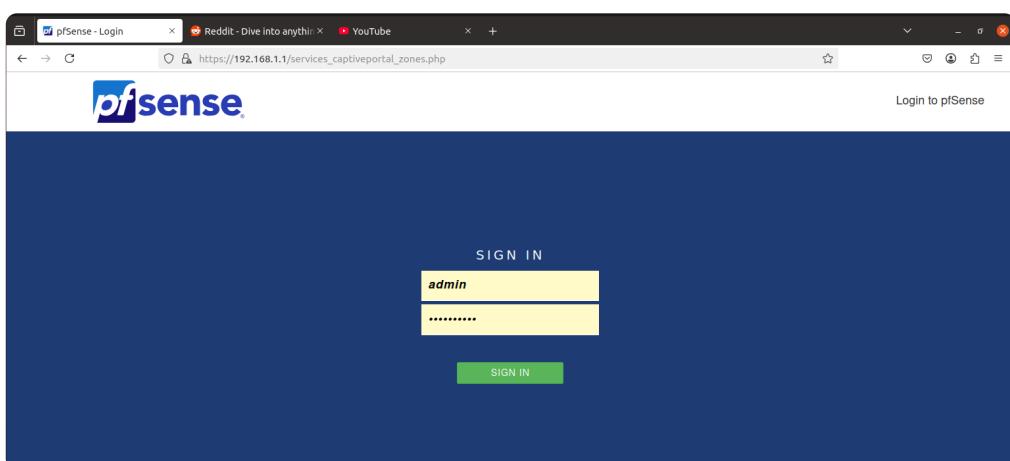
7. Une fois pfSense installé et redémarré, vous verrez un menu avec :

- WAN (par défaut en [DHCP](#))
- LAN (par défaut en [192.168.1.1/24](#))



## IV.4. Configuration initiale des interfaces réseau

1. Sur votre ordinateur hôte, configurez une adresse IP statique dans le même sous-réseau que l'interface LAN de PfSense :
  - Adresse IP : 192.168.1.102
  - Masque de sous-réseau : 255.255.255.0
  - Passerelle par défaut : 192.168.1.1
2. Ouvrez un navigateur web et accédez à l'adresse <https://192.168.1.1>
3. Ignorez les avertissements de sécurité du navigateur concernant le certificat
4. Connectez-vous avec les identifiants par défaut :
  - Nom d'utilisateur : **admin**
  - Mot de passe : **pfsense**



## IV.5. Configuration des règles de pare-feu

1. Dans l'interface web de PfSense, allez dans **Firewall > Rules**
2. Sélectionnez l'onglet **LAN**
3. Par défaut, une règle permettant tout le trafic sortant depuis LAN devrait exister
4. Si ce n'est pas le cas, ajoutez une règle :
  - Action : **Pass**
  - Interface : **LAN**
  - Adresse source : **LAN net**
  - Adresse de destination : **Any**
  - Description : "Allow LAN to Internet"
5. Cliquez sur **Save** puis sur **Apply Changes**

## IV.6. Configuration Radius pour l'authentification

### 1. Activer et Configurer le Captive Portal sur pfSense

1. Connectez-vous à [pfSense](#)
2. Va dans [Services](#) → [Captive Portal](#)
3. Clique sur [Ajouter une Zone](#) et donne-lui un nom (ex: Portail\_Reseau)
4. Active la zone et choisis l'interface sur laquelle tu veux appliquer le portail captif (ex: [LAN](#) ou [WIFI](#))
5. Clique sur [Save & Continue](#)

**Captive Portal Configuration**

**Enable**:  Enable Captive Portal

**Description**: veuillez mettre vos infos  
A description may be entered here for administrative reference (not parsed).

**Interfaces**: WAN LAN  
Select the interface(s) to enable for captive portal.

**Maximum concurrent connections**:   
Limits the number of concurrent connections to the captive portal HTTP(S) server. This does not set how many users can be logged in to the captive portal, but rather how many connections a single IP can establish to the portal web server.

**Idle timeout (Minutes)**:   
Clients will be disconnected after this amount of inactivity. This means a session immediately times out. Leave this field blank for no idle timeout.

**Authentication**

**Authentication Method**: Use an Authentication backend  
Select an Authentication Method to use for this zone. One method must be selected.  
- "Authentication backend" will force the login page to be displayed and will authenticate users using their login and password, or using vouchers.  
- "None" method will force the login page to be displayed but will accept any visitor that clicks the "submit" button.  
- "RADeUS MAC Authentication" method will try to authenticate devices automatically with their MAC address without displaying any login page.

**Authentication Server**: RADIUS Local Database  
You can add a remote authentication server in the User Manager. Vouchers could also be used, please go to the Vouchers Page to enable them.

**Secondary authentication Server**: RADIUS Local Database  
You can optionally select a second set of servers to authenticate users. Users will then be able to login using separated HTML inputs. This setting is useful if you want to provide multiple authentication method to your users. If you don't need multiple authentication method, then leave this setting empty.

**NAS Identifier**:   
Specify a NAS identifier to override the default value (CaptivePortal-smarttechsn)

**Reauthenticate Users**:  Reauthenticate connected users every minute

## 2. Configurer l'Authentification via FreeRADIUS

1. Dans **pfSense**, va dans **System** → **User Manager** → **Authentication Servers**

2. Clique sur **Add** et remplis :

- **Descriptive Name** : FreeRADIUS
- **Type** : RADIUS
- **Hostname or IP Address** : 192.168.1.103 (ton serveur FreeRADIUS)
- **Shared Secret** : (mets le même secret que dans FreeRADIUS)
- **Services Offered** : coche Authentication and Accounting
- **Authentication Port** : 1812
- **Accounting Port** : 1813

3. Clique sur **Save & Test** pour voir si la connexion est OK

https://192.168.1.1/system\_authservers.php?act=edit&id=

Users	Groups	Settings	Authentication Servers
<b>Server Settings</b>			
<u>Descriptive name</u>	RADIUS		
<u>Type</u>	RADIUS		
<b>RADIUS Server Settings</b>			
<u>Protocol</u>	MS-CHAPv2		
<u>Hostname or IP address</u>	192.168.1.103		
<u>Shared Secret</u>	*****		
<u>Services offered</u>	Authentication and Accounting		
<u>Authentication port</u>	1812		
<u>Accounting port</u>	1813		
<u>Authentication Timeout</u>	5		
This value controls how long, in seconds, that the RADIUS server may take to respond to an authentication request. NOTE: If using an interactive two-factor authentication system, increase this timeout to account for the delay in entering a token.			
<b>RADIUS NAS IP Attribute</b>			
LAN - 192.168.1.1			
Enter the IP to use for the "NAS-IP-Address" attribute during RADIUS Access-Requests. Please note that this choice won't change the interface used for contacting the RADIUS server.			

#### 4. Vous pouvez voir le nombre d'utilisateurs connectés dans l'interface

https://192.168.1.1/services\_captiveportal\_zones.php

Captive Portal Zones				
Zone	Interfaces	Number of users	Description	Actions
smartechsn	LAN	2	veuillez mettre vos infos	

voici une **video demo**

## VII. Conclusion

Félicitations ! Vous avez maintenant un système complet avec PfSense, un portail captif, et une authentification via RADIUS et LDAP. Cette configuration vous permet de :

- Gérer votre réseau avec un pare-feu robuste
- Sécuriser l'accès à Internet via un portail d'authentification
- Centraliser la gestion des utilisateurs via LDAP
- Auditer les connexions grâce aux journaux RADIUS

Cette configuration est adaptée à de nombreux environnements, notamment :

- Les petites et moyennes entreprises
- Les établissements éducatifs
- Les hôtels et espaces publics offrant un accès Wi-Fi
- Les environnements de test et de développement

Pour aller plus loin, vous pourriez explorer :

- La mise en place d'un VPN pour l'accès à distance
- La configuration de VLAN pour segmenter davantage votre réseau
- L'implémentation de règles de filtrage par utilisateur
- La supervision du réseau avec des outils comme Nagios ou Zabbix