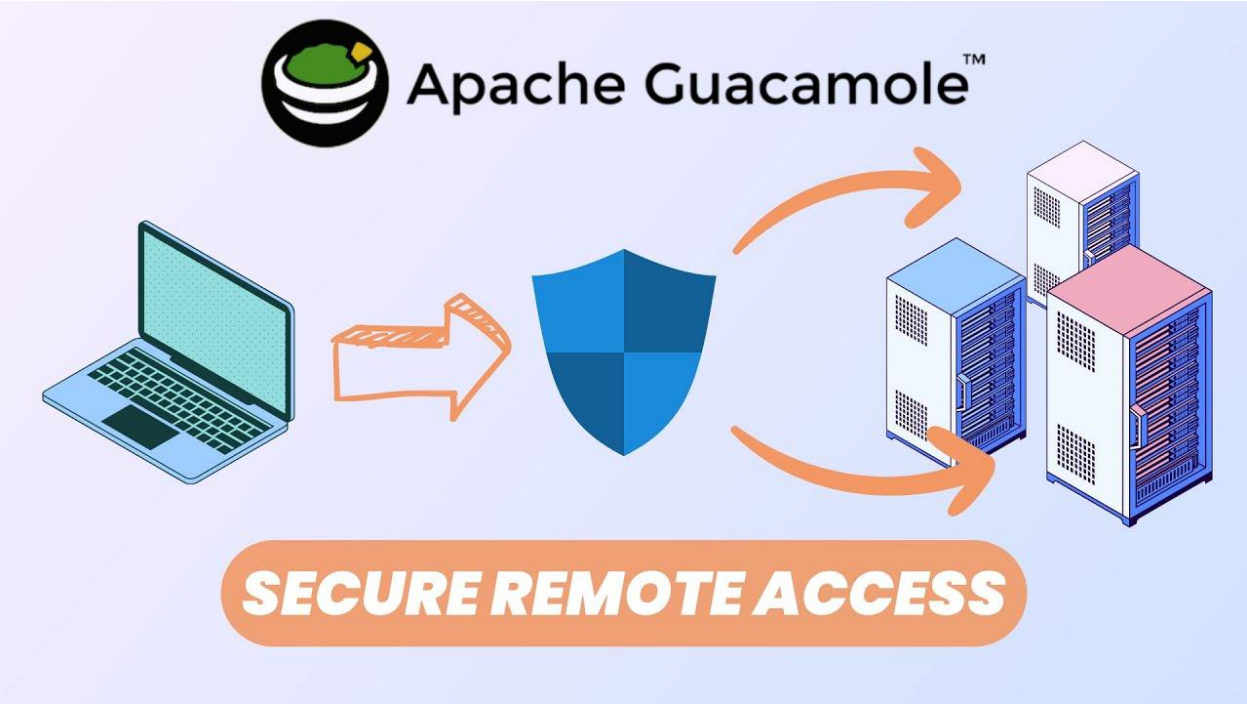
DOC TECHNIQUE



Mamadou CAMARA

Table des matières

Table des matières 2
1. Présentation d'Apache Guacamole 4
1.1 Protocoles supportés 4
1.2 Fonctions clés 4
2. Installation sur Debian 12 4
2.1 Prérequis système 4
2.2 Compilation de Guacamole Server 4
2.3 Installation de Tomcat 9 et Client Web 5
3. Configuration de la base de données 6
3.1 Installation de MariaDB 6
Création de la base de données 6
3.2 Extension MySQL et connecteur 6
Import de la structure 6
3.3 Fichiers de configuration 6
4. Configuration et premiers pas 8
4.1 Accès initial 8
4.2 Créer un nouveau compte admin 8
4.3 Création d'une connexion RDP 11
5. Sécurisation avancée 16
5.1 Authentification multi-facteurs (TOTP) 16
5.2 Connexion avec l'AD (LDAP) 17
5.2 Enregistrement vidéo des sessions 19
6. Résolution de problèmes 20
6.1 Erreur de certificat RDP 20
6.2 Erreur de négociation de sécurité 20
7. Bonnes pratiques 21
7.1 Sécurité 21

7.2 Haute disponibilité ____ **21**

7.3 Gestion des utilisateurs ____ **21**

8. Conclusion et ressources ____ ***22***

8.1 Ressources officielles ____ **22**

8.2 Versions utilisées ____ **22**

1. Présentation d'Apache Guacamole

Apache Guacamole est une solution open source permettant de mettre en place un bastion d'administration (passerelle d'accès ou serveur de rebond). Cette solution offre une interface web HTML5 pour accéder aux serveurs sans nécessiter l'installation de clients lourds.

1.1 Protocoles supportés

- **RDP** (Remote Desktop Protocol)
- **SSH** (Secure Shell)
- **VNC** (Virtual Network Computing)
- **Telnet**
- **Kubernetes**

1.2 Fonctions clés

- **Centralisation des connexions** : traçabilité complète (qui, quand, où, durée)
- **Interface web HTML5** : aucun client lourd requis
- **Authentification MFA** : support TOTP
- **SSO** : compatible SAML, OpenID Connect, LDAP
- **Enregistrements vidéo** : replay complet des sessions
- **Gestion granulaire** : autorisations par groupes ou utilisateurs

2. Installation sur Debian 12

2.1 Prérequis système

```
apt-get update

apt-get install build-essential libcairo2-dev libjpeg62-turbo-dev libpng-dev libtool-bin uuid-dev libossp-uuid-dev libavcodec-dev libavformat-dev libavutil-dev libswscale-dev freerdp2-dev libpango1.0-dev libssh2-1-dev libtelnet-dev libvncserver-dev libwebsockets-dev libpulse-dev libssl-dev libvorbis-dev libwebp-dev
```

2.2 Compilation de Guacamole Server

```
cd /tmp

wget https://downloads.apache.org/guacamole/1.6.0/source/guacamole-server-1.6.0.tar.gz

tar -xzf guacamole-server-1.6.0.tar.gz

cd guacamole-server-1.6.0/

sudo ./configure --with-systemd-dir=/etc/systemd/system/

sudo make
```

```
sudo make install
sudo ldconfig
sudo systemctl daemon-reload
```

```
root@debian:/home/mamadou# sudo systemctl status guacd
● guacd.service - Guacamole Server
     Loaded: loaded (/etc/systemd/system/guacd.service; enabled; preset: enabled)
     Active: active (running) since Wed 2026-02-04 11:45:32 CET; 6min ago
       Docs: man:guacd(8)
   Main PID: 691 (guacd)
      Tasks: 1 (limit: 4552)
     Memory: 10.0M
        CPU: 29ms
     CGroup: /system.slice/guacd.service
             └─691 /usr/local/sbin/guacd -f

févr. 04 11:45:32 debian systemd[1]: Started guacd.service - Guacamole Server.
févr. 04 11:45:32 debian guacd[691]: Guacamole proxy daemon (guacd) version 1.6.0 started
févr. 04 11:45:32 debian guacd[691]: guacd[691]: INFO:        Guacamole proxy daemon (guacd) version 1.6.0 started
févr. 04 11:45:32 debian guacd[691]: Listening on host 0.0.0.0, port 4822
févr. 04 11:45:32 debian guacd[691]: guacd[691]: INFO:        Listening on host 0.0.0.0, port 4822
root@debian:/home/mamadou# sudo systemctl restart tomcat9 guacd
```

```
sudo systemctl enable --now guacd
```

2.3 Installation de Tomcat 9 et Client Web

Tomcat 10 n'est pas supporté. Ajoutez le dépôt Debian 12 :

```
sudo nano /etc/apt/sources.list.d/bullseye.list
```

Ajoutez : deb http://deb.debian.org/debian/ bullseye main

```
sudo apt-get update
sudo apt-get install tomcat9 tomcat9-admin tomcat9-common tomcat9-user
cd /tmp
wget https://downloads.apache.org/guacamole/1.6.0/binary/guacamole-1.6.0.war
sudo mv guacamole-1.6.0.war /var/lib/tomcat9/webapps/guacamole.war
sudo systemctl restart tomcat9 guacd
```

3. Configuration de la base de données

3.1 Installation de MariaDB

```
sudo apt-get install mariadb-server
sudo mysql_secure_installation
```

Création de la base de données

```
mysql -u root -p
CREATE DATABASE guacadb; CREATE USER 'guaca_nachos'@'localhost' IDENTIFIED BY 'P@ssword!'; GRANT SELECT,INSERT,UPDATE,DELETE ON guacadb.* TO 'guaca_nachos'@'localhost'; FLUSH PRIVILEGES; EXIT;
```

> ⚠️ **Important :** Adaptez le nom de la base, l'utilisateur et le mot de passe selon votre politique de sécurité.

3.2 Extension MySQL et connecteur

```
cd /tmp
wget https://downloads.apache.org/guacamole/1.6.0/binary/guacamole-auth-jdbc-1.6.0.tar.gz
tar -xzf guacamole-auth-jdbc-1.6.0.tar.gz
sudo mv guacamole-auth-jdbc-1.6.0/mysql/guacamole-auth-jdbc-mysql-1.6.0.jar /etc/guacamole/extensions/
```

Téléchargez le connecteur MySQL :

```
wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-j-9.1.0.tar.gz
tar -xzf mysql-connector-j-9.1.0.tar.gz
sudo cp mysql-connector-j-9.1.0/mysql-connector-j-9.1.0.jar /etc/guacamole/lib/
```

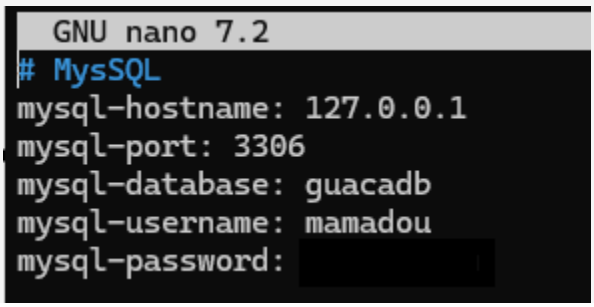
Import de la structure

```
cd guacamole-auth-jdbc-1.6.0/mysql/schema/
cat *.sql | mysql -u root -p guacadb
```

3.3 Fichiers de configuration

Créez /etc/guacamole/guacamole.properties :

```
# MySQL
mysql-hostname: 127.0.0.1
mysql-port: 3306
mysql-database: guacadb
mysql-username: mamadou
 mysql-password: P@ssword!
```



Créez /etc/guacamole/guacd.conf :

```
[server]
bind_host = 0.0.0.0
bind_port = 4822
```

Redémarre les services

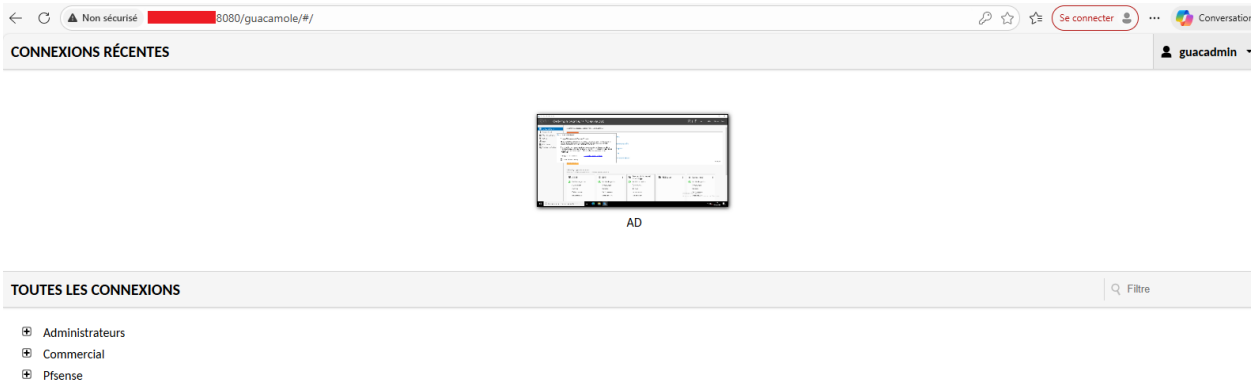
```
sudo systemctl restart tomcat9 guacd mariadb
```

4. Configuration et premiers pas

4.1 Accès initial

Accédez à l'interface : http://<IP>:8080/guacamole/

Identifiants par défaut : guacadmin / guacadmin



🔒 **Sécurité :** Créez immédiatement un nouveau compte administrateur puis supprimez le compte 'guacadmin'.

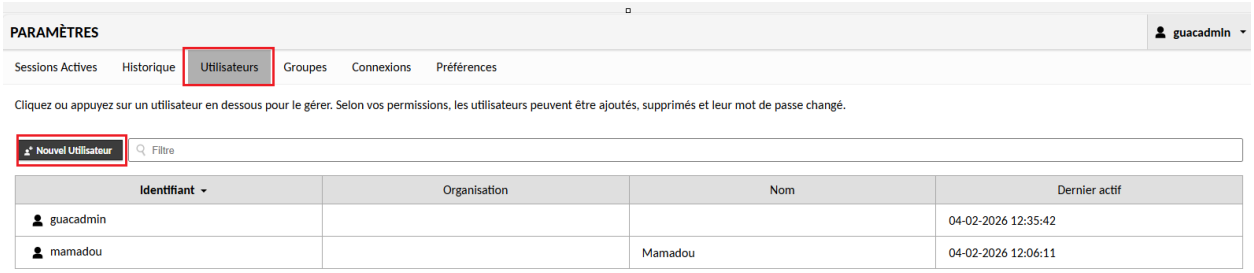
4.2 Créer un nouveau compte admin

Tout d'abord, nous allons **créer un nouveau compte d'administration** et supprimer le compte par défaut, pour des raisons de sécurité.

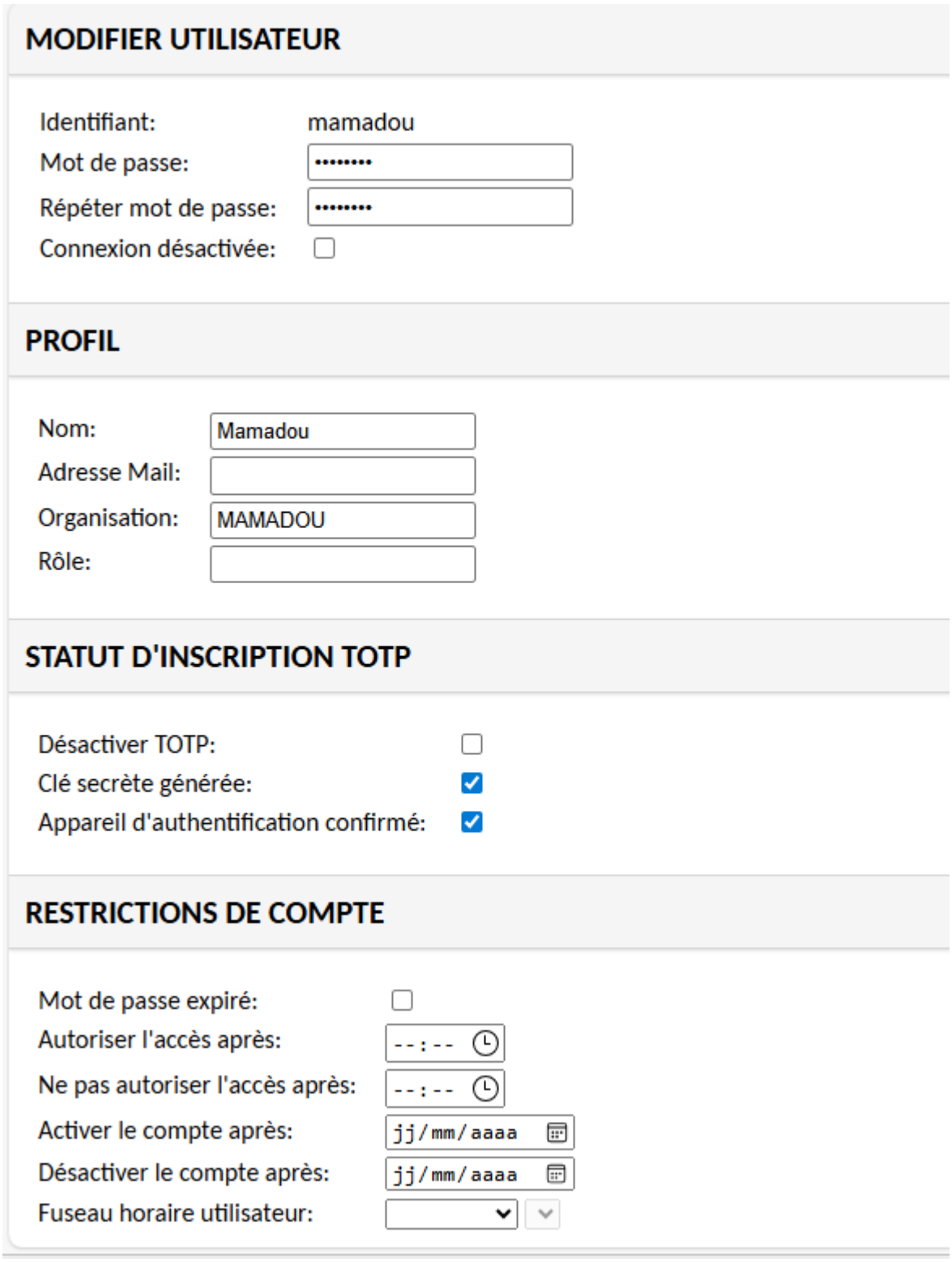
Notre objectif est le suivant :

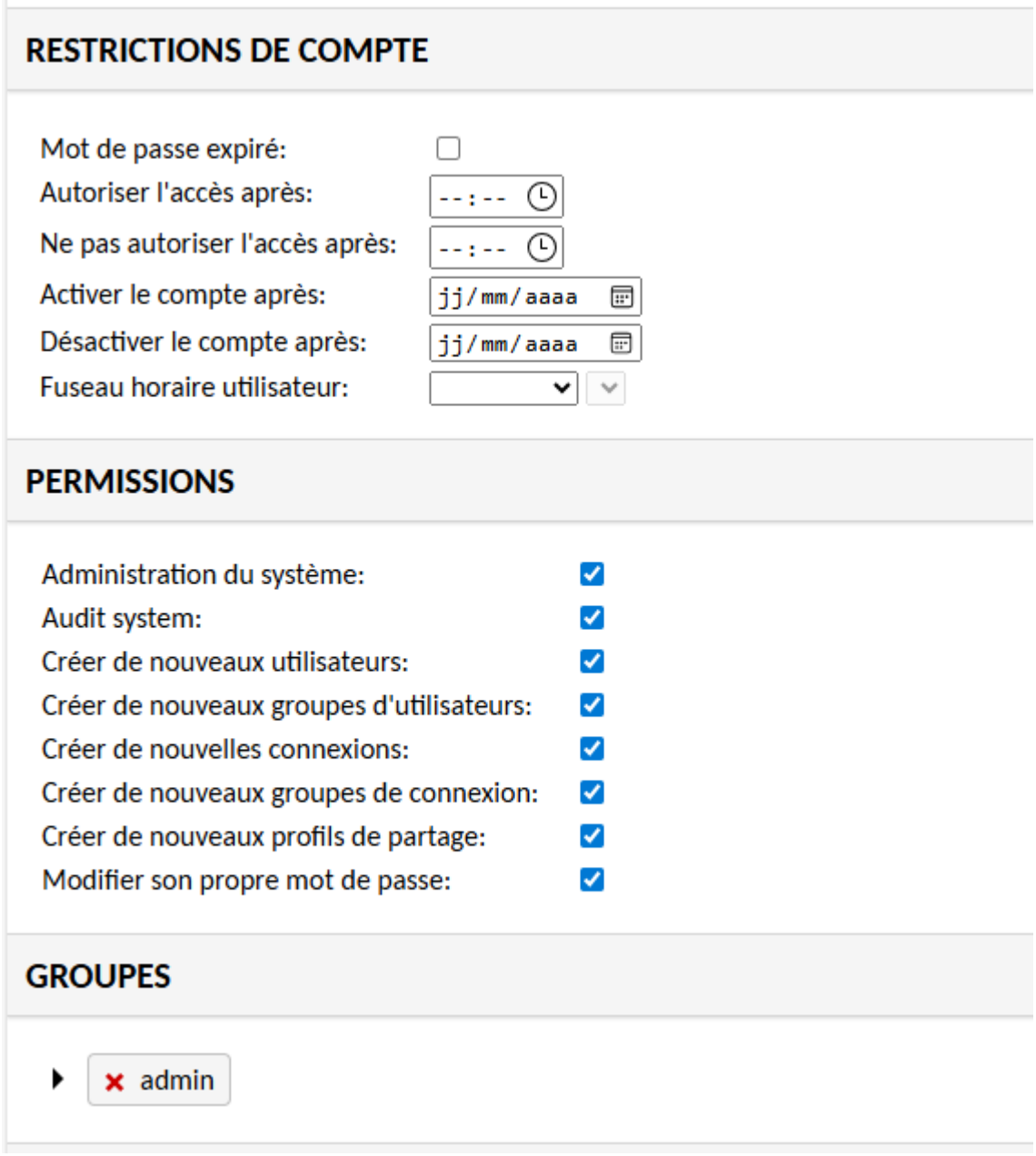
- Créer un nouveau compte administrateur (avec un nom personnalisé)
- Se déconnecter du compte "**guacadmin**"
- Se reconnecter avec le nouveau compte administrateur
- Supprimer le compte "**guacadmin**" par défaut (ou à minima changer son mot de passe et le désactiver)

Pour accéder aux paramètres, il faut cliquer sur le nom d'utilisateur en haut à droite puis sur "**Paramètres**".



Un formulaire s'affiche. **Indiquez un nom d'utilisateur**, en évitant les traditionnels "Administrateur", "Admin", etc.... Et choisissez **un mot de passe robuste**. Cochez l'ensemble des permissions pour que cet utilisateur soit administrateur de la plateforme Guacamole.


MODIFIER UTILISATEUR
Identifiant:
mamadou
Mot de passe:
Répéter mot de passe:
Connexion désactivée:
PROFIL
Nom:
Mamadou
Adresse Mail:
Organisation:
MAMADOU
Rôle:
STATUT D'INSCRIPTION TOTP
Désactiver TOTP:
Clé secrète générée:
Appareil d'authentification confirmé:
RESTRICTIONS DE COMPTE
Mot de passe expiré:
Autoriser l'accès après:
--:--
Ne pas autoriser l'accès après:
--:--
Activer le compte après:
jj/mm/aaaa
Désactiver le compte après:
jj/mm/aaaa
Fuseau horaire utilisateur:



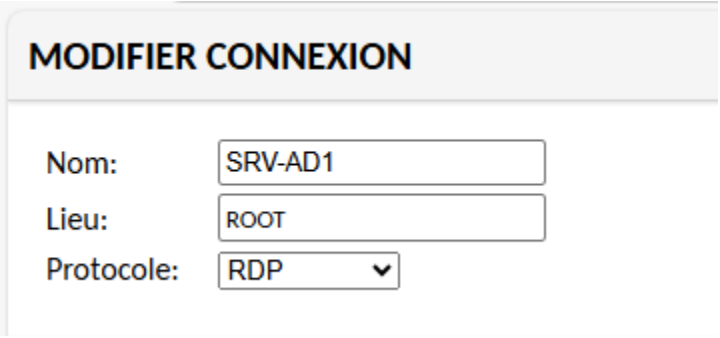
4.3 Création d'une connexion RDP

Pour créer une nouvelle connexion : **Paramètres > Connexion > Nouvelle connexion**

Mais avant cela, on va **créer un nouveau groupe**, car ces groupes vont permettre d'organiser les connexions : **Paramètres > Connexion > Nouveau groupe**

Dans cet exemple, je crée un groupe nommé "**Serveurs applicatifs**". Il sera positionné sous le lieu "**ROOT**" qui est la racine de l'arborescence. Le type de groupe "**Organizationel**" doit être sélectionné pour tous les groupes qui ont pour vocation à organiser les connexions.

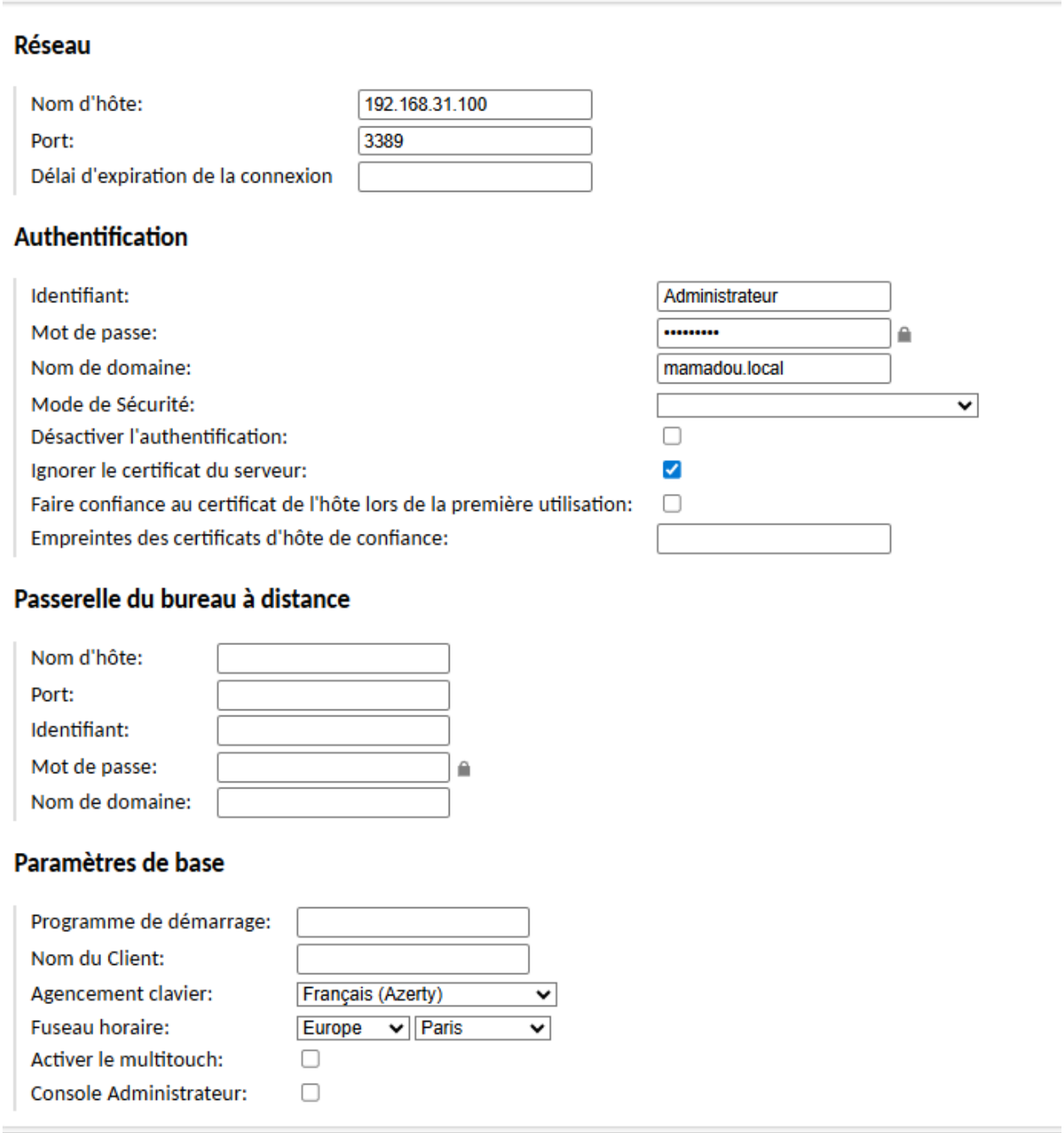
On enregistre et on clique sur le bouton "**Nouvelle connexion**". On commence par nommer la connexion, choisir le groupe et le protocole. Ici, c'est sur le serveur "**SRV-AD1**" que je souhaite me connecter, associé à l'adresse IP "**192.168.31.100**"



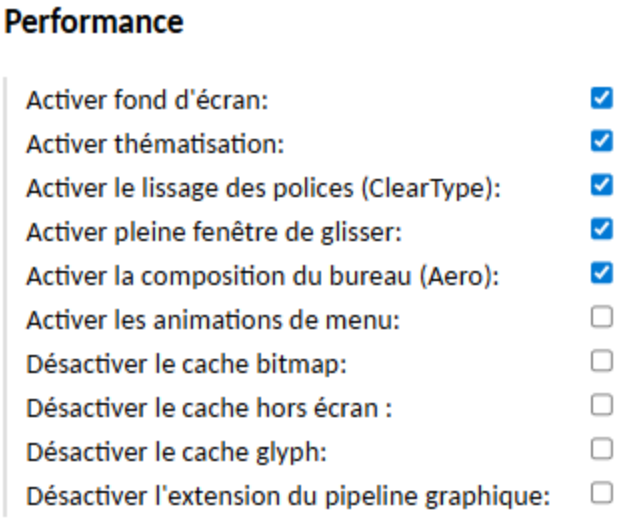
Ensuite, il y a un ensemble de paramètres à renseigner :

- **Nom d'hôte** : le nom DNS du serveur (si le serveur Apache Guacamole est capable de résoudre le nom), sinon l'adresse IP
- **Port** : le numéro de port du RDP, par défaut 3389 (pas utile de le préciser si c'est le port par défaut)
- **Identifiant** : compte avec lequel s'authentifier sur le serveur
- **Mot de passe** : mot de passe du compte spécifié ci-dessus
- **Nom de domaine** : nom du domaine Active Directory, si besoin
- **Mode de sécurité** : par défaut, c'est en détection automatique (vous pouvez également choisir le NLA)
- **Ignorer le certificat du serveur** : cochez cette option si vous n'avez pas déployé de certificat pour vos connexions RDP et si vous utilisez une adresse IP pour la connexion
- **Agencement clavier** : choisissez "**Français (Azerty)**", ou adaptez selon votre configuration

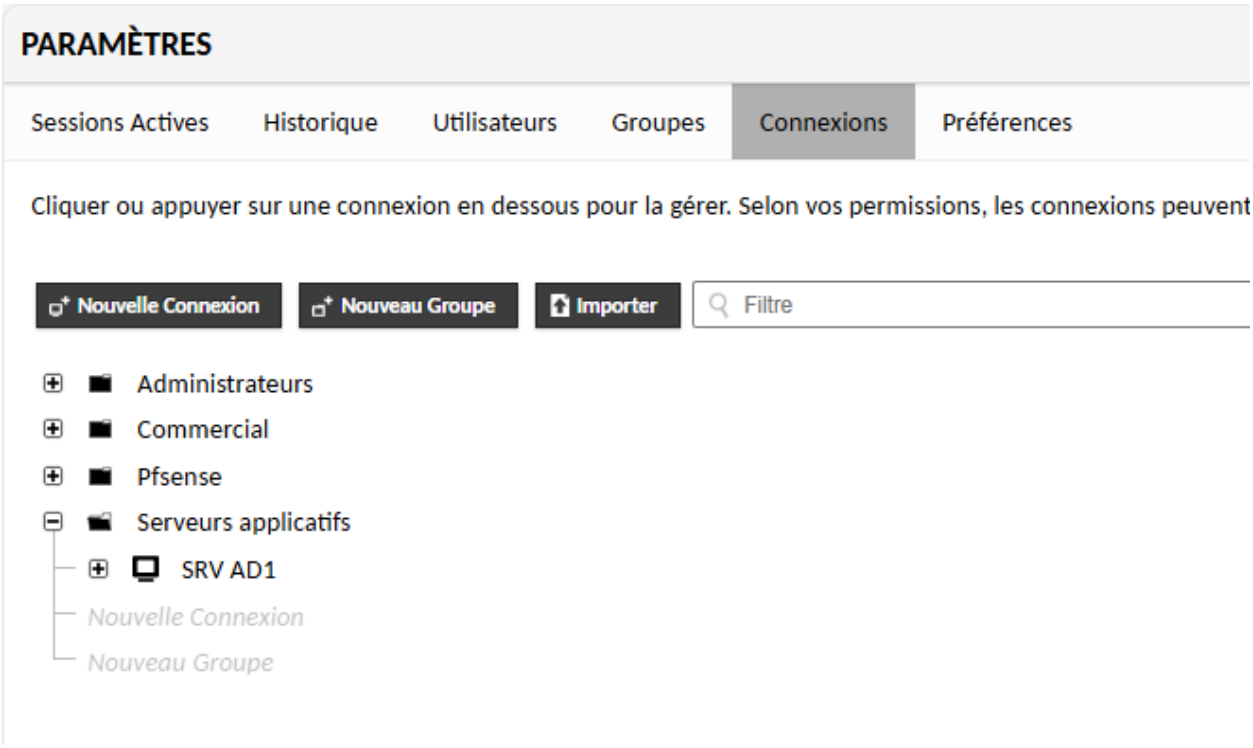
- **Fuseau horaire** : choisissez "**Europe / Paris**", ou adaptez selon votre configuration



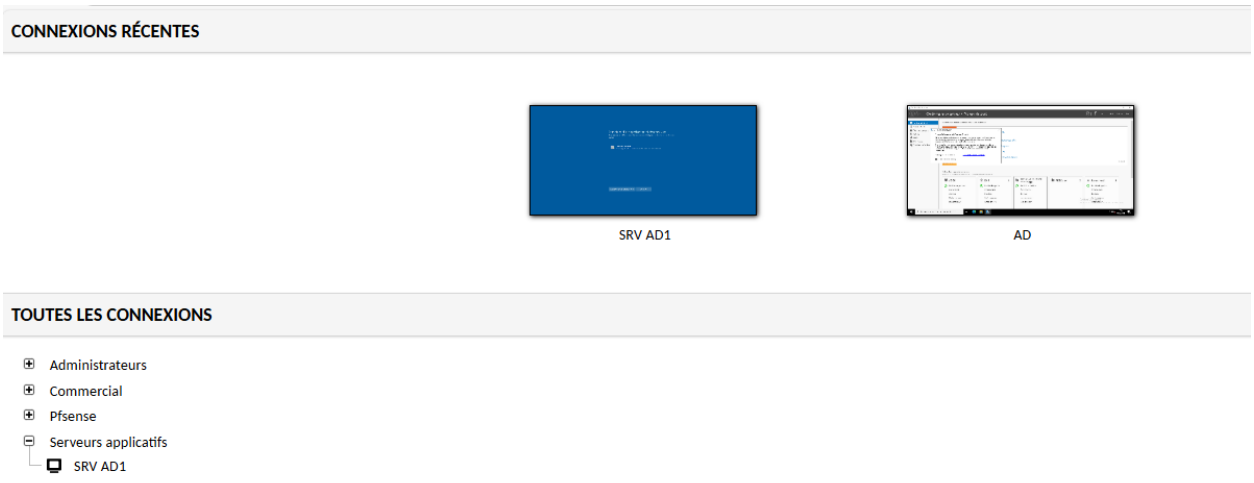
Il y a de **nombreuses options disponibles**, notamment pour faire remonter les périphériques locaux, ou passer par une passerelle de bureau à distance. Au début, il faut passer du temps à trouver la bonne formule pour que la connexion RDP intègre toutes les fonctions dont on a besoin. Toutefois, si l'on veut simplement se connecter et avoir le contrôle à distance, ce n'est pas utile de modifier profondément la configuration.



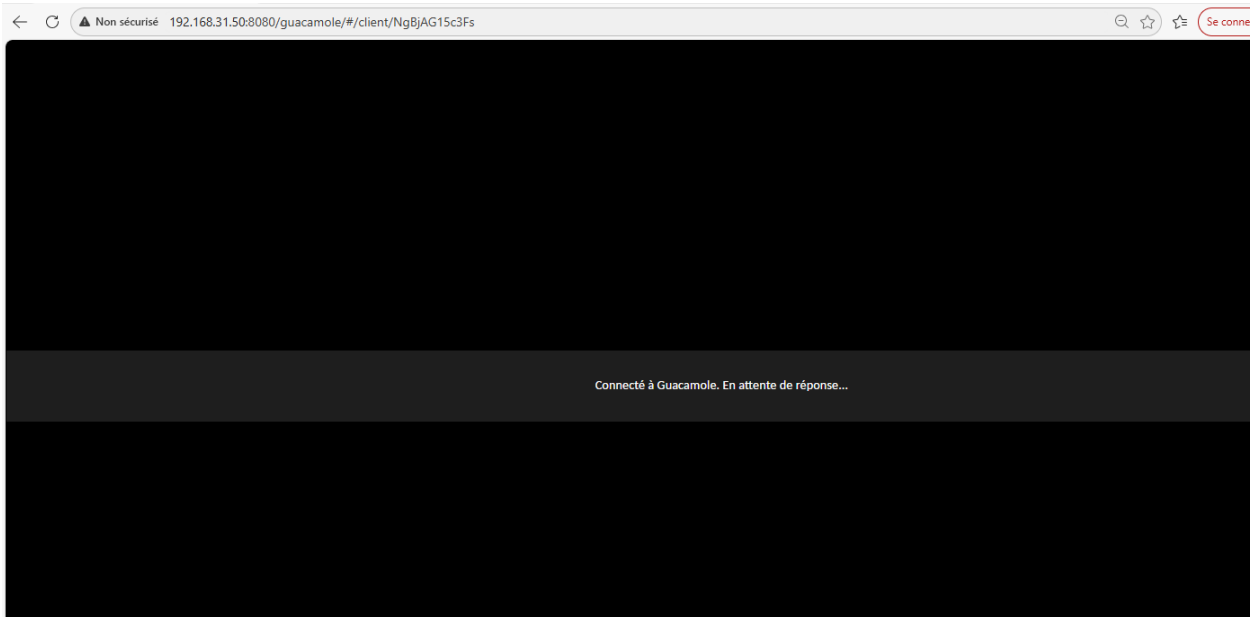
Enregistrez. La nouvelle connexion apparaît sous "**Serveurs applications**". Pour tester cette connexion, il faut basculer sur "**Accueil**" en cliquant sur son identifiant en haut à droite.

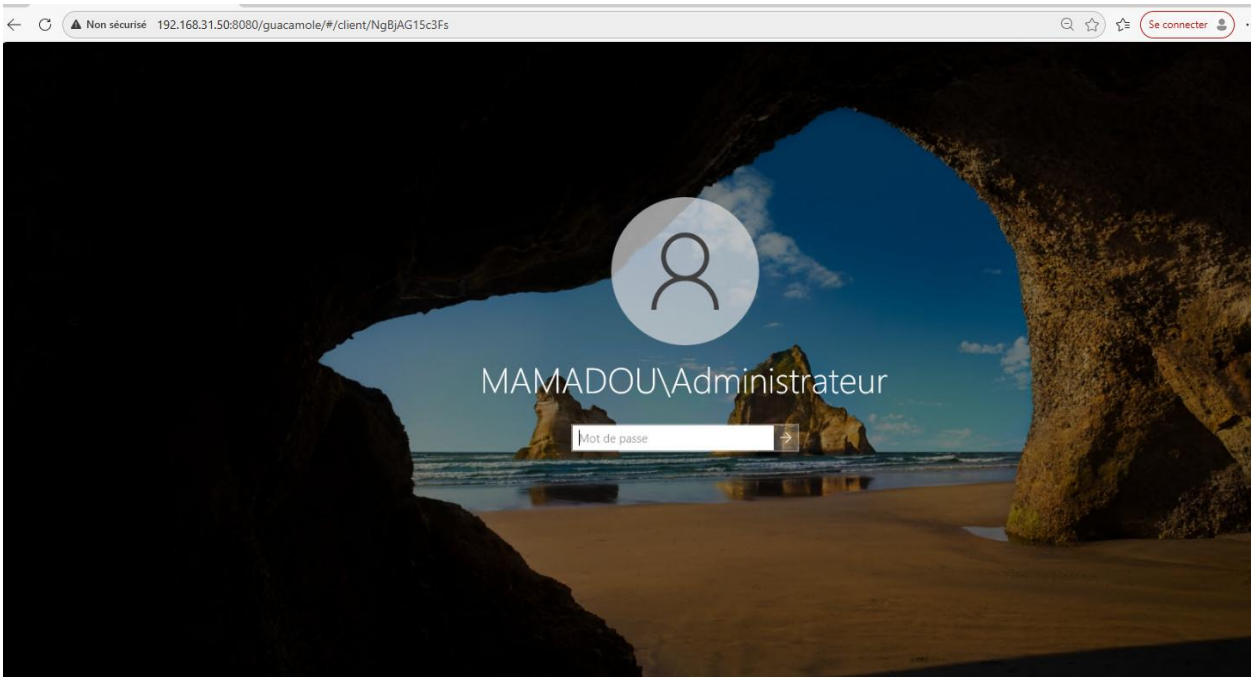


Dans l'accueil, l'utilisateur peut visualiser toutes les connexions qu'il a le droit d'utiliser.



Il suffit de cliquer sur le serveur et la connexion va se lancer...





5. Sécurisation avancée

5.1 Authentification multi-facteurs (TOTP)

```
cd /tmp

wget https://downloads.apache.org/guacamole/1.6.0/binary/guacamole-auth-totp-1.6.0.tar.gz

tar -xzf guacamole-auth-totp-1.6.0.tar.gz

sudo mv guacamole-auth-totp-1.6.0/guacamole-auth-totp-1.6.0.jar /etc/guacamole/extensions/
```

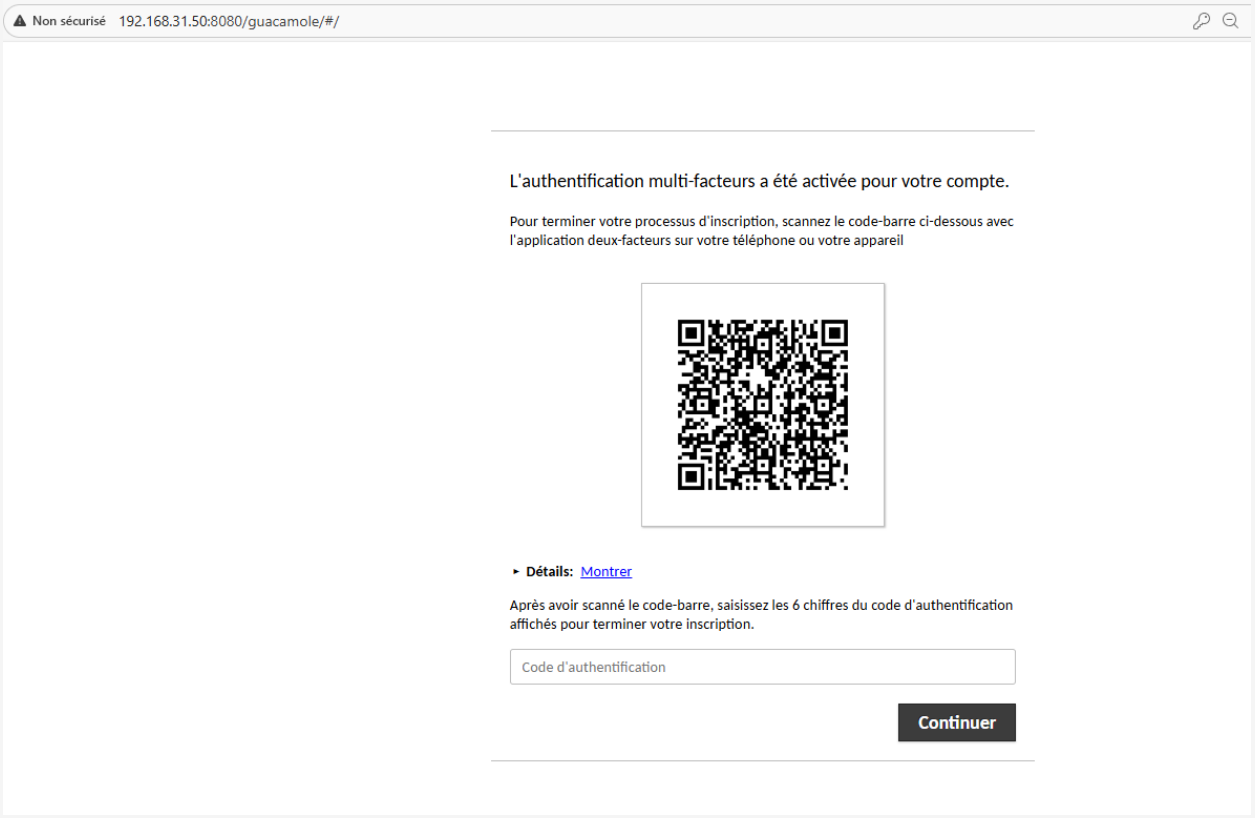
Ajoutez dans guacamole.properties :

```
# TOTP totp-issuer: Guacamole MAMADOU totp-digits: 6 totp-period: 30 totp-mode: sha1
```

```
# TOTP
totp-issuer: Guacamole mamadou
totp-digits: 6
totp-period: 30
totp-mode: sha1
```

```
sudo systemctl restart tomcat9
```

Crée un utilisateur dans guacamole pour tester le TOTP :



Désormais, un code TOTP devra être indiqué à chaque nouvelle connexion sur Guacamole. Dans les paramètres de chaque utilisateur, il y a une section "Configure TOTP" qui donne l'état du MFA sur le compte, avec la possibilité de réinitialiser le secret TOTP sur le compte en question.

5.2 Connexion avec l'AD (LDAP)

Pour qu'Apache Guacamole s'appuie sur l'Active Directory, il y a plusieurs prérequis :

- Installer l'extension LDAP pour Guacamole
- Disposer d'un compte utilisateur dans l'Active Directory (pour interroger l'annuaire) - *Pas besoin d'être admin !*
- Identifier la partie de l'arborescence Active Directory qui sera utilisée comme base de recherche pour les objets

Ensuite, positionnez-vous dans "/tmp" et téléchargez l'archive tar.gz de l'extension :

```
cd /tmp
```

```
wget https://downloads.apache.org/guacamole/1.6.0/binary/guacamole-auth-ldap-1.6.0.tar.gz
```

Une fois que c'est fait, décompressez cette archive tar.gz :

```
tar -xzf guacamole-auth-ldap-1.6.0.tar.gz
```

Puis, déplacer le fichier JAR de l'extension vers le répertoire "extensions" de Guacamole.

```
sudo mv guacamole-auth-ldap-1.6.0/guacamole-auth-ldap-1.6.0.jar /etc/guacamole/extensions
```

Pour que cette extension soit prise en charge, il convient de redémarrer Tomcat9 :

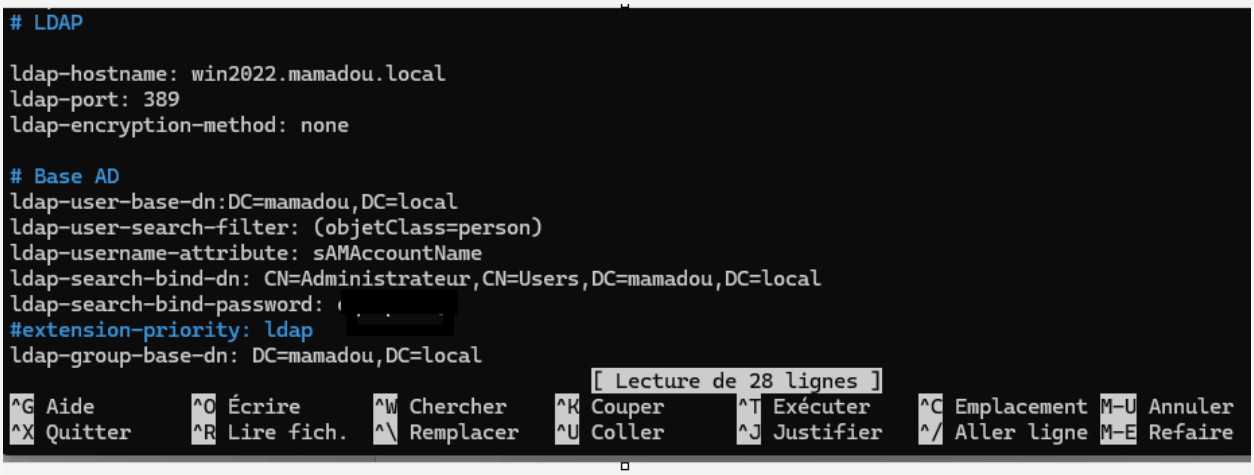
```
sudo systemctl restart tomcat9
```

Toujours sur le serveur Apache Guacamole, ouvrez le fichier de configuration :

```
sudo nano /etc/guacamole/guacamole.properties
```

Dans ce fichier, il y a déjà d'autres propriétés définies, notamment les informations de connexion à la base de données MySQL (MariaDB). Vous devez configurer l'extension LDAP.

Premièrement, vous devez indiquer le nom du contrôleur de domaine, le port LDAP (*389* par défaut) et la méthode de chiffrement (*none* avec du LDAP classique). Ce qui donne les lignes suivantes à ajouter dans le fichier :



Deuxièmement, à la suite, vous devez ajouter les informations sur le compte à utiliser pour s'authentifier auprès de l'annuaire Active Directory. Ce compte sert uniquement à

lire le contenu de l'annuaire (utilisateurs, groupes, et membre des groupes). Dans l'exemple ci-dessous, le compte "*Sync_Guacamole@it-connect.local*" est utilisé.

Le fichier de configuration est prêt ! Sauvegardez et fermez le fichier. Il ne reste plus qu'à relancer Tomcat9 :

```
sudo systemctl restart tomcat9
```

5.2 Enregistrement vidéo des sessions

```
cd /tmp
wget https://downloads.apache.org/guacamole/1.6.0/binary/guacamole-history-recording-storage-1.6.0.tar.gz
tar -xzf guacamole-history-recording-storage-1.6.0.tar.gz
sudo mv guacamole-history-recording-storage-1.6.0/guacamole-history-recording-storage-1.6.0.jar /etc/guacamole/extensions/
sudo mkdir -p /var/lib/guacamole/recordings
sudo chown root:tomcat /var/lib/guacamole/recordings
sudo chmod 2750 /var/lib/guacamole/recordings
sudo systemctl restart tomcat9
```

Dans les paramètres de chaque connexion, configurez la section 'Enregistrement écran' :

- Chemin : ${HISTORY_PATH}/${HISTORY_UUID}
- Nom : ${GUAC_DATE}-${GUAC_TIME} - RDP - ${GUAC_USERNAME}
- Créer automatiquement le chemin : ✓

6. Résolution de problèmes

6.1 Erreur de certificat RDP

Vérifiez les logs avec :

```
sudo systemctl status guacd
```

Si vous voyez 'Certificate validation failed', cochez l'option 'Ignorer le certificat du serveur' dans les paramètres de la connexion.

6.2 Erreur de négociation de sécurité

Pour 'Security negotiation failed', créez un utilisateur dédié pour guacd :

```
sudo useradd -M -d /var/lib/guacd/ -r -s /sbin/nologin -c "Guacd User" guacd
sudo mkdir /var/lib/guacd
sudo chown -R guacd: /var/lib/guacd
sudo sed -i 's/daemon/guacd/' /etc/systemd/system/guacd.service
sudo systemctl daemon-reload
sudo systemctl restart guacd
```

7. Bonnes pratiques

7.1 Sécurité

- Activez systématiquement le MFA pour tous les utilisateurs
- Placez un reverse proxy HTTPS avec certificat valide devant Guacamole
- Configurez les règles de pare-feu pour que seul Guacamole puisse accéder aux serveurs
- Exigez une connexion VPN pour l'accès externe

7.2 Haute disponibilité

- Déployez plusieurs instances derrière un load balancer
- Utilisez une base de données en cluster ou réplication
- Sauvegardez régulièrement la base de données et les enregistrements

7.3 Gestion des utilisateurs

- Organisez les connexions avec des groupes logiques
- Intégrez avec Active Directory via LDAP pour centraliser les identités
- Configurez des dates d'expiration pour les comptes temporaires

8. Conclusion et ressources

Apache Guacamole représente une solution complète pour centraliser et sécuriser les accès aux serveurs. Ses fonctionnalités d'authentification multi-facteurs, d'enregistrement des sessions et de gestion granulaire des autorisations permettent de répondre aux exigences de conformité les plus strictes.

8.1 Ressources officielles

- **Site officiel :** https://guacamole.apache.org/
- **Documentation :** https://guacamole.apache.org/doc/gug/
- **Téléchargements :** https://downloads.apache.org/guacamole/

8.2 Versions utilisées

Composant	Version
Système d'exploitation	Debian 12 (Bookworm)
Apache Guacamole	1.6.0
Apache Tomcat	9.x
MariaDB Server	10.11.x

FIN