Table des matières

Sadek Adel 09/10/2025

Introduction

Aujourdhui je vais m'initier à l'installation et l'utilisation d'un serveur bastion qui va nous permettre de centraliser les accès à nos serveurs et ressources au lieu d'uaotirser tout un réseau à avoir accès en SSH vers le VLAN SRV on autorisera seulement le serveur bastion

Nous installerons la solution apache-guacamole

Installation

Apache Guacamole devient un élément central de l'infrastructure puisqu'il sert de **passerelle pour administrer les machines**. Rassurez-vous, il est possible d'avoir plusieurs hôtes Apache Guacamole pour **répartir la charge et assurer la haute disponibilité**.

Apache Guacamole intègre plusieurs fonctions séduisantes qui vont nous permettre de mieux suivre les accès aux serveurs de notre infrastructure.

- **Centralisation et suivi des connexions** : qui, quand, où, combien de temps, depuis où
- **Aucun client lourd à installer**, l'accès s'effectue en mode web grâce au HTML5
- **Authentification multi-facteurs pour l'accès aux connexions**, via un code TOTP
- **Authentification SSO**, compatible avec SAML, OpenID Connect, CAS ou encore LDAP
- **Enregistrements vidéos des sessions**, c'est-à-dire quand une connexion est en cours d'utilisation
- **Gestion des autorisations pour l'accès aux connexions**, par groupes ou par utilisateurs

Installer les prerequis

Il faut installer tout les prérequis avant

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

Installer tomcat

Apache-guacamole utilise un serveur tomcat

Il faut ce rendre dans le dossier tmp et installer les sources

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

Sadek Adel 09/10/2025


```
root@srv-ollama:/tmp# wget https://downloads.apache.org/guacamole/1.6.0/source/guacamole-server-1.6.0.tar.gz
--2025-10-09 23:48:28--  https://downloads.apache.org/guacamole/1.6.0/source/guacamole-server-1.6.0.tar.gz
Résolution de downloads.apache.org (downloads.apache.org)… 135.181.214.104, 88.99.208.237, 2a01:4f8:10a:39da::2, ...
Connexion à downloads.apache.org (downloads.apache.org)|135.181.214.104|:443… connecté.
requête HTTP transmise, en attente de la réponse… 200 OK
Taille : 1252749 (1,2M) [application/x-gzip]
Sauvegarde en : « guacamole-server-1.6.0.tar.gz »

guacamole-server-1.6.0.tar.gz      100%[===================================================================>]   1,19M  4,48MB/s    ds 0,3s

2025-10-09 23:48:29 (4,48 MB/s) – « guacamole-server-1.6.0.tar.gz » sauvegardé [1252749/1252749]

root@srv-ollama:/tmp#
```

Ensuite on fait un tar

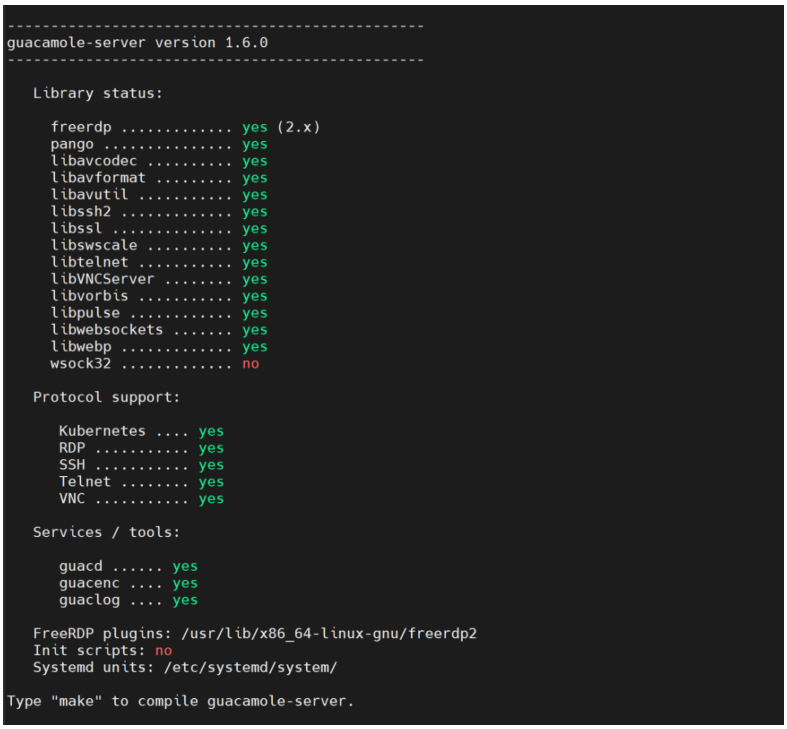
```
root@srv-ollama:/tmp# tar -xzf guacamole-server-1.6.0.tar.gz
root@srv-ollama:/tmp# cd guacamole-server-1.6.0/
root@srv-ollama:/tmp/guacamole-server-1.6.0# ls
aclocal.m4  build-aux    configure     CONTRIBUTING  Dockerfile  m4          Makefile.in  README  util
bin         config.h.in  configure.ac  doc           LICENSE     Makefile.am NOTICE       src
root@srv-ollama:/tmp/guacamole-server-1.6.0#
```

On exécute la commande ci-dessous pour se préparer à la compilation, ce qui va permettre de vérifier la présence des dépendances :

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

Si on a une erreur aprceque freerdp n’est pas reconnue il faut contourner avec cette commande

./configure --with-systemd-dir=/etc/systemd/system/ --enable-allow-freerdp-snapshots



Ensuite lancer commande “make”

Problème rencontré lors de la compilation de FreeRDP (pré-requis Guacamole Server)

Contexte

Lors de la compilation de **FreeRDP** (version 2.9.0, nécessaire pour la compatibilité avec Guacamole Server 1.6.0), une erreur s’est produite au moment de l’exécution de la commande :

```
cmake -DWITH_PULSE=ON -DWITH_SERVER=OFF -DCMAKE_BUILD_TYPE=Release .
```

Message d’erreur obtenu

```
CMake Error: The following variables are used in this project, but they are
set to NOTFOUND.
Please set them or make sure they are set and tested correctly in the CMake
files:
LIBUSB_1_INCLUDE_DIR
   used as include directory in directory /usr/src/FreeRDP/channels/urbdrc
LIBUSB_1_LIBRARY
    linked by target "urbdrc-client-libusb" in directory
/usr/src/FreeRDP/channels/urbdrc/client/libusb
```

Explication

Le message indique que **les dépendances liées à la bibliothèque `libusb-1.0`** ne sont pas trouvées sur le système.
Ces dépendances sont nécessaires pour la compilation du **module URBDRC** (*USB*

redirection channel) de FreeRDP.
Ce module permet la redirection des périphériques USB dans une session RDP, mais il n'est pas obligatoire pour le fonctionnement de Guacamole.

Solutions possibles

Solution 1 – Installer la bibliothèque manquante (recommandée)

Si la redirection USB est souhaitée, il faut installer le paquet de développement `libusb-1.0-0-dev` :

```
sudo apt update
sudo apt install -y libusb-1.0-0-dev
```

Ensuite, relancer la configuration et la compilation :

```
cd /usr/src/FreeRDP
sudo rm -rf CMakeCache.txt CMakeFiles
sudo cmake -DWITH_PULSE=ON -DWITH_SERVER=OFF -DCMAKE_BUILD_TYPE=Release .
sudo make -j$(nproc)
sudo make install
sudo ldconfig
```

Solution 2 – Désactiver le module USB (alternative rapide)

Si la redirection USB n'est pas nécessaire (cas le plus fréquent pour Guacamole),
on peut désactiver complètement la compilation du module **URBDRC** :

```
cd /usr/src/FreeRDP
sudo rm -rf CMakeCache.txt CMakeFiles
sudo cmake -DWITH_PULSE=ON -DWITH_SERVER=OFF -DWITH_URBDRC=OFF -
DCMAKE_BUILD_TYPE=Release .
sudo make -j$(nproc)
sudo make install
sudo ldconfig
```

Cette méthode contourne l'erreur tout en conservant la compatibilité avec Guacamole.

Résultat attendu

Après l'application de l'une des deux solutions ci-dessus, la compilation de FreeRDP se termine avec succès.
On peut alors procéder à la compilation du **Guacamole Server** sans erreur :

```
cd /tmp/guacamole-server-1.6.0
make clean
./configure --with-systemd-dir=/etc/systemd/system/
make -j$(nproc)
sudo make install
sudo ldconfig
```

🏁 Conclusion

Le problème provenait d'une **dépendance manquante (`libusb-1.0`)** utilisée par le module de redirection USB de FreeRDP.
Deux solutions sont possibles :

- **Installer le paquet `libusb-1.0-0-dev`** pour conserver la redirection USB.
- **Désactiver `URBDRC`** pour simplifier la compilation.

Après correction, FreeRDP s'est compilé avec succès et Guacamole Server a pu être installé normalement.

Et ensuite make install

```
 /usr/bin/install -c -m 644 man/guacd.conf.5 '/usr/local/share/man/man5'
 /usr/bin/mkdir -p '/usr/local/share/man/man8'
 /usr/bin/install -c -m 644 man/guacd.8 '/usr/local/share/man/man8'
 /usr/bin/mkdir -p '/etc/systemd/system/'
 /usr/bin/install -c -m 644 systemd/guacd.service '/etc/systemd/system/'
make[2] : on quitte le répertoire « /tmp/guacamole-server-1.6.0/src/guacd »
make[1] : on quitte le répertoire « /tmp/guacamole-server-1.6.0/src/guacd »
Making install in src/guacenc
make[1] : on entre dans le répertoire « /tmp/guacamole-server-1.6.0/src/guacenc »
make[2] : on entre dans le répertoire « /tmp/guacamole-server-1.6.0/src/guacenc »
 /usr/bin/mkdir -p '/usr/local/bin'
  /bin/bash ../../libtool   --mode=install /usr/bin/install -c guacenc '/usr/local/bin'
libtool: install: /usr/bin/install -c .libs/guacenc /usr/local/bin/guacenc
 /usr/bin/mkdir -p '/usr/local/share/man/man1'
 /usr/bin/install -c -m 644 man/guacenc.1 '/usr/local/share/man/man1'
make[2] : on quitte le répertoire « /tmp/guacamole-server-1.6.0/src/guacenc »
make[1] : on quitte le répertoire « /tmp/guacamole-server-1.6.0/src/guacenc »
Making install in src/guaclog
make[1] : on entre dans le répertoire « /tmp/guacamole-server-1.6.0/src/guaclog »
make[2] : on entre dans le répertoire « /tmp/guacamole-server-1.6.0/src/guaclog »
 /usr/bin/mkdir -p '/usr/local/bin'
  /bin/bash ../../libtool   --mode=install /usr/bin/install -c guaclog '/usr/local/bin'
libtool: install: /usr/bin/install -c .libs/guaclog /usr/local/bin/guaclog
 /usr/bin/mkdir -p '/usr/local/share/man/man1'
 /usr/bin/install -c -m 644 man/guaclog.1 '/usr/local/share/man/man1'
make[2] : on quitte le répertoire « /tmp/guacamole-server-1.6.0/src/guaclog »
make[1] : on quitte le répertoire « /tmp/guacamole-server-1.6.0/src/guaclog »
make[1] : on entre dans le répertoire « /tmp/guacamole-server-1.6.0 »
make[2] : on entre dans le répertoire « /tmp/guacamole-server-1.6.0 »
make[2]: rien à faire pour « install-exec-am ».
make[2]: rien à faire pour « install-data-am ».
make[2] : on quitte le répertoire « /tmp/guacamole-server-1.6.0 »
make[1] : on quitte le répertoire « /tmp/guacamole-server-1.6.0 »
root@srv-ollama:/tmp/guacamole-server-1.6.0#
```

Ensuite il faut executer la commande « ldconfig « qui mettra à jour les liens entre quacamole-server et ses librairires

```
Ensuite
sudo systemctl daemon-reload
sudo systemctl enable --now guacd
```

Ensuite on regarde le status pour voir si le serveur tourne correctement

Sadek Adel 09/10/2025

```
root@srv-ollama:/tmp/guacamole-server-1.6.0# sudo systemctl status guacd
● guacd.service - Guacamole Server
     Loaded: loaded (/etc/systemd/system/guacd.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2025-10-10 00:01:36 CEST; 6s ago
       Docs: man:guacd(8)
   Main PID: 32084 (guacd)
      Tasks: 1 (limit: 4700)
     Memory: 9.9M
     CGroup: /system.slice/guacd.service
             └─32084 /usr/local/sbin/guacd -f

oct. 10 00:01:36 srv-ollama systemd[1]: Started Guacamole Server.
oct. 10 00:01:36 srv-ollama guacd[32084]: Guacamole proxy daemon (guacd) version 1.6.0 started
oct. 10 00:01:36 srv-ollama guacd[32084]: guacd[32084]: INFO:        Guacamole proxy daemon (guacd) version 1.6.0 starte
oct. 10 00:01:36 srv-ollama guacd[32084]: guacd[32084]: INFO:        Listening on host ::1, port 4822
oct. 10 00:01:36 srv-ollama guacd[32084]: Listening on host ::1, port 4822
lines 1-15/15 (END)
```

Avant de passer à la partie cliente

Création de l'arborescence

Dernière étape avant de passer à la partie client d'Apache Guacamole, **on crée l'arborescence pour la configuration d'Apache Guacamole**. Cela va donner le répertoire "**/etc/guacamole**" avec les sous-répertoires "**extensions**" et "**lib**". Nous en aurons besoin par la suite pour mettre en place le stockage des données dans une base de données MariaDB / MySQL.

```
mkdir -p /etc/guacamole/{extensions,lib}
```

Installation de la web app

Pour la **Web App** correspondante à Apache Guacamole, et donc à la partie cliente, nous avons besoin d'un serveur **Tomcat 9**. J'insiste sur le fait que **Tomcat 10**, **distribué par défaut via les dépôts de Debian 12**, n'est **pas pris en charge par Apache Guacamole**. Nous devons **ajouter le dépôt de Debian 11** sur notre machine Debian 12 afin de pouvoir **télécharger les paquets correspondants à Tomcat 9**.

Ensuite apt-get update

Et

```
apt-get install tomcat9 tomcat9-admin tomcat9-common tomcat9-user
```

Puis, nous allons **télécharger la dernière version de la Web App d'Apache Guacamole** depuis le dépôt officiel (même endroit que pour la partie serveur). On se positionne dans "**/tmp**" et on télécharge la Web App, ce qui revient à télécharger un fichier avec l'extension "**.war**". Ici, la **version 1.5.5** est téléchargée.

Sadek Adel 09/10/2025

Le client à installer

```
cd /tmp
wget https://downloads.apache.org/guacamole/1.6.0/binary/guacamole-1.6.0.war
```

Ensuite il faut deplacer le fichier dans librairie Web App de Tomcat9

```
/var/lib/tomcat9/webapps/guacamole.war
```

Dans ce dossier

```
root@srv-ollama:/tmp# mv guacamole-1.6.0.war /var/lib/tomcat9/webapps/
root@srv-ollama:/tmp# cd /var/lib/tomcat9/webapps/
root@srv-ollama:/var/lib/tomcat9/webapps# mv guacamole-1.6.0 guacamole.war
root@srv-ollama:/var/lib/tomcat9/webapps#
```

Ensuite on relance les services tomcat9 et guacamole

```
root@srv-ollama:/var/lib/tomcat9/webapps# service tomcat9 restart
root@srv-ollama:/var/lib/tomcat9/webapps# service guacd restart
root@srv-ollama:/var/lib/tomcat9/webapps#
```

Installation mariadb

Ensuite il faut installer mariadb-server

```
MariaDB [(none)]> CREATE DATABASE guacadb;
Query OK, 1 row affected (0,000 sec)

MariaDB [(none)]> CREATE USER 'guaca_nachos'@'localhost' IDENTIFIED BY 'P@ssword!';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> GRANT SELECT,INSERT,UPDATE,DELETE ON guacadb.* TO 'guaca_nachos'@'localhost';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,000 sec)

MariaDB [(none)]> EXIT;
Bye
```

Il faut maintenant installer l'extension Mysql d'apache guacamole

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

Ensuite on decompresse avec tar

Sadek Adel 09/10/2025

```
root@srv-ollama:/tmp# tar -xzf guacamole-auth-jdbc-1.6.0.tar.gz
root@srv-ollama:/tmp#
```

On obtient un fichier jar il faut le déplacer ici

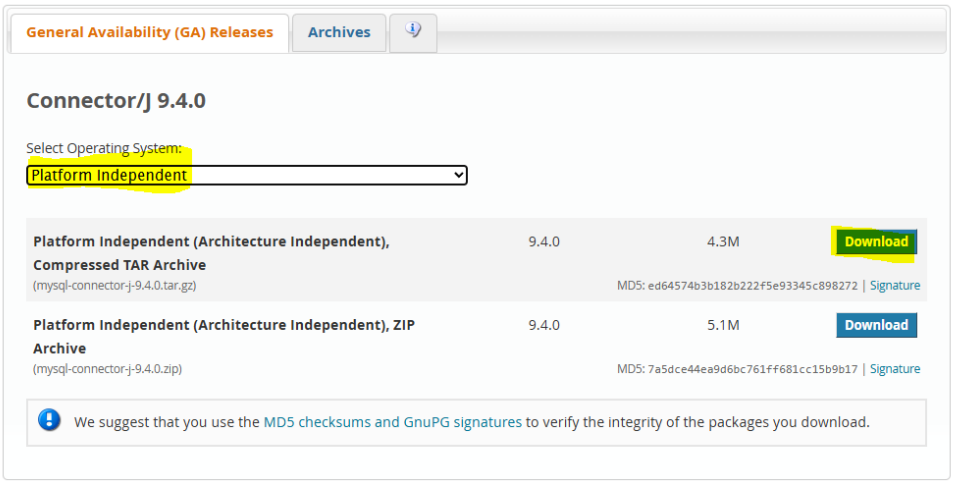
```
/etc/guacamole/extensions/
```

Il faut ensuite installer le connector mysql un peu comme quand on fait du sql sur python

https://dev.mysql.com/downloads/connector/j/

MySQL Community Downloads

Connector/J



Wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-j-9.4.0.tar.gz

```
root@srv-ollama:/tmp# wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-j-9.4.0.tar.gz
--2025-10-10 00:26:08--  https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-j-9.4.0.tar.gz
Résolution de dev.mysql.com (dev.mysql.com)… 23.204.228.220, 2a02:26f0:9100:595::2e31, 2a02:26f0:9100:58a::2e31
Connexion à dev.mysql.com (dev.mysql.com)|23.204.228.220|:443… connecté.
requête HTTP transmise, en attente de la réponse… 302 Moved Temporarily
Emplacement : https://cdn.mysql.com//Downloads/Connector-J/mysql-connector-j-9.4.0.tar.gz [suivant]
--2025-10-10 00:26:08--  https://cdn.mysql.com//Downloads/Connector-J/mysql-connector-j-9.4.0.tar.gz
Résolution de cdn.mysql.com (cdn.mysql.com)… 23.64.36.51, 2a02:26f0:9100:594::1d68, 2a02:26f0:9100:58c::1d68
Connexion à cdn.mysql.com (cdn.mysql.com)|23.64.36.51|:443… connecté.
requête HTTP transmise, en attente de la réponse… 200 OK
Taille : 4493403 (4,3M) [application/x-tar-gz]
Sauvegarde en : « mysql-connector-j-9.4.0.tar.gz »

mysql-connector-j-9.4.0.tar.g 100%[=============================================>]   4,29M  7,23MB/s    ds 0,6s

2025-10-10 00:26:09 (7,23 MB/s) – « mysql-connector-j-9.4.0.tar.gz » sauvegardé [4493403/4493403]

root@srv-ollama:/tmp# tar -xzf mysql-connector-j-9.4.0.tar.gz
root@srv-ollama:/tmp#
```

Il faut copier le fichier jar encore une fois dans ce dossier

/etc/guacamole/lib/

```
root@srv-ollama:/tmp# cp mysql-connector-j-9.4.0/mysql-connector-j-9.4.0.jar /etc/guacamole/lib/
root@srv-ollama:/tmp#
```

Les dépendances sont déployées, mais nous n'avons pas encore fini cette intégration avec MariaDB.

En effet, il faut **importer la structure de la base de données Apache Guacamole dans notre base de données "guacadb"**. Pour cela, on va importer tous les fichiers SQL situés dans le répertoire "**guacamole-auth-jdbc-1.6.0/mysql/schema/**". Le mot de passe root de MariaDB doit être saisit pour effectuer l'import.

root@srv-ollama:/tmp# tar -xzf mysql-connector-j-9.4.0.tar.gz

root@srv-ollama:/tmp# cp mysql-connector-j-9.4.0/mysql-connector-j-9.4.0.jar /etc/guacamole/lib/

root@srv-ollama:/tmp# cd ./guacamole-auth-jdbc-1.6.0/mysql/schema/

root@srv-ollama:/tmp/guacamole-auth-jdbc-1.6.0/mysql/schema# cat *.sql | mysql -u root -p guacadb

```
root@srv-ollama:/tmp# tar -xzf mysql-connector-j-9.4.0.tar.gz
root@srv-ollama:/tmp# cp mysql-connector-j-9.4.0/mysql-connector-j-9.4.0.jar /etc/guacamole/lib/
root@srv-ollama:/tmp# cd ./guacamole-auth-jdbc-1.6.0/mysql/schema/
root@srv-ollama:/tmp/guacamole-auth-jdbc-1.6.0/mysql/schema# cat *.sql | mysql -u root -p guacadb
Enter password:
root@srv-ollama:/tmp/guacamole-auth-jdbc-1.6.0/mysql/schema#
```

Une fois que c'est fait, on va **créer et éditer le fichier "guacamole.properties"** pour déclarer la connexion à MariaDB. Ce fichier peut être utilisé pour d'autres paramètres, selon vos besoins.

```
  GNU nano 3.2                          /etc/guacamole/guacamole.properties

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

Sadek Adel 09/10/2025

Declaration du serveur guacamole

Tant que l'on est dans la configuration, **éditez le fichier "guacd.conf" pour déclarer le serveur Guacamole** (ici, on déclare une connexion locale sur le port par défaut, à savoir 4822).

Les deux fichiers sont /etc/guacamole/guacd.conf

Ensuite on redemarre tout les services et si tout va bien l'instllation de base sera terminer

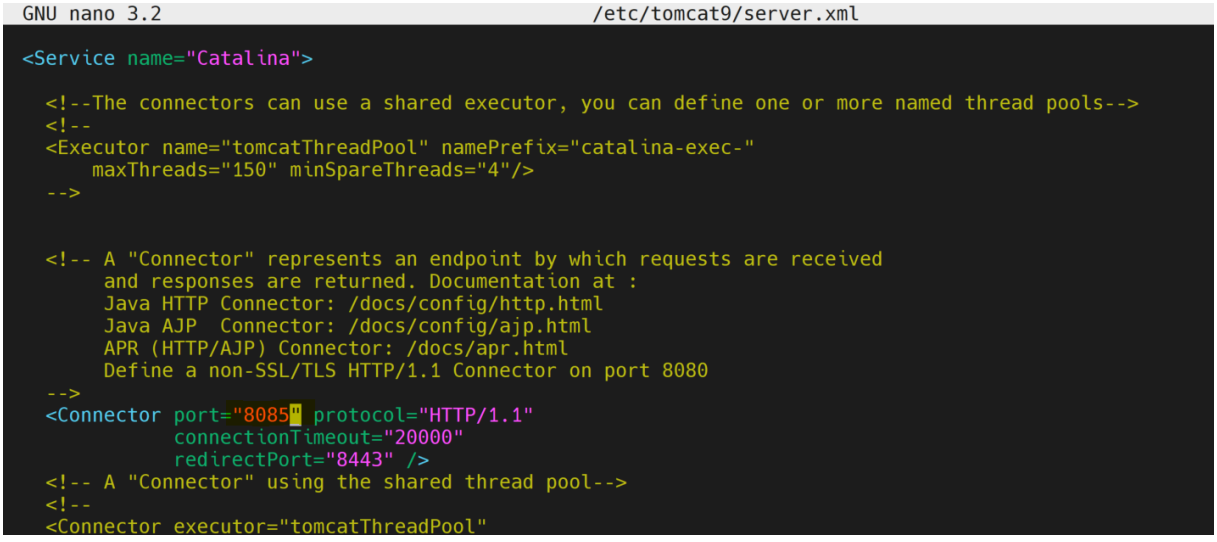
```
systemctl restart tomcat9 guacd mariadb
```

Accès web

```
http://<Adresse IP>:8080/guacamole/
```

Mon port étant déjà occuper j'utiliserai le 8085

sudo nano /etc/tomcat9/server.xml



Sadek Adel 09/10/2025

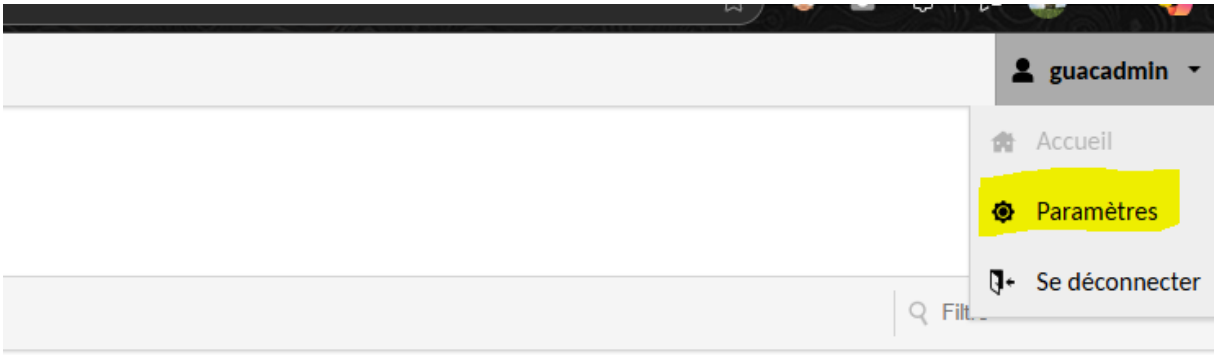
Donc le port est 8085

Connexion à la WEBUI

Pour se connecter, on va utiliser les identifiants par défaut :

- Utilisateur : **guacadmin**
- Mot de passe : **guacadmin**

Pour opérer des modifications il faut ce rendre dans parametre ici

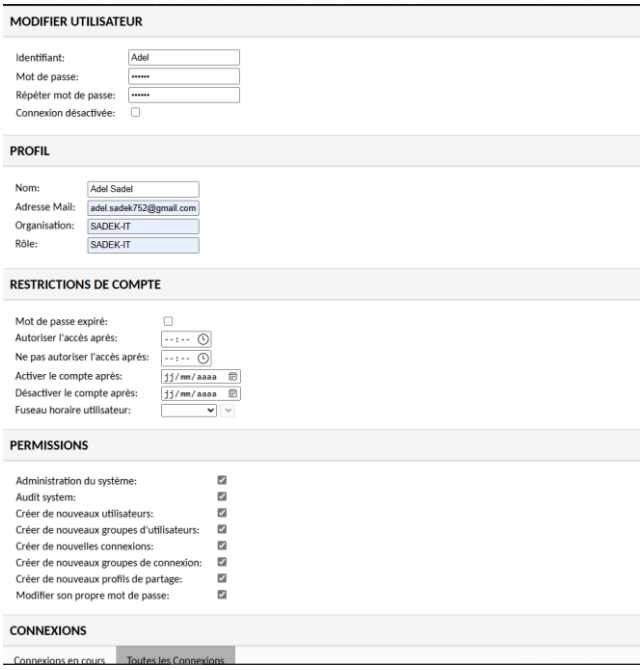


Création user

On va dans utilisateur > Créer nouvel utilisateur

Sadek Adel 09/10/2025

Je créer mon utilisateur ce qui est intéressant ce que l'on peut faire des restrictions de compte accès à une certaine plage horaire etc



Voila nos deux comptes

Il faut supprimer ou descativer guacadmin maintenant

Je suis rentrer dedans et j'ai cliquer sur connexion désactiver

Je me reconencte avec mon nouveau admin

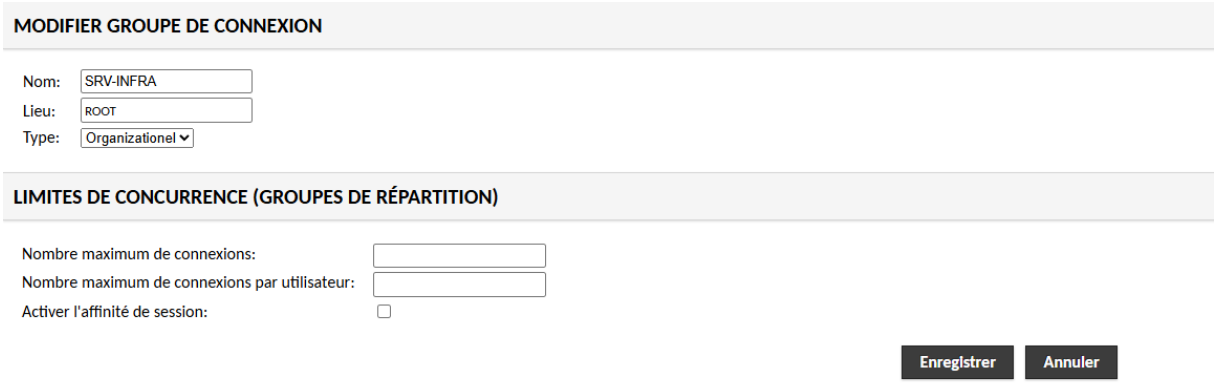
Nous allons maintenant ajouter une connection RDP

Sadek Adel 09/10/2025

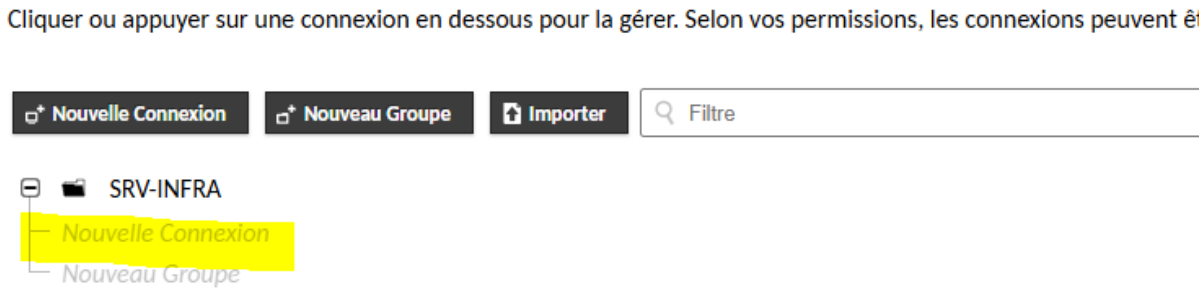
Ajout d’une machine

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 INFRA** ". 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.

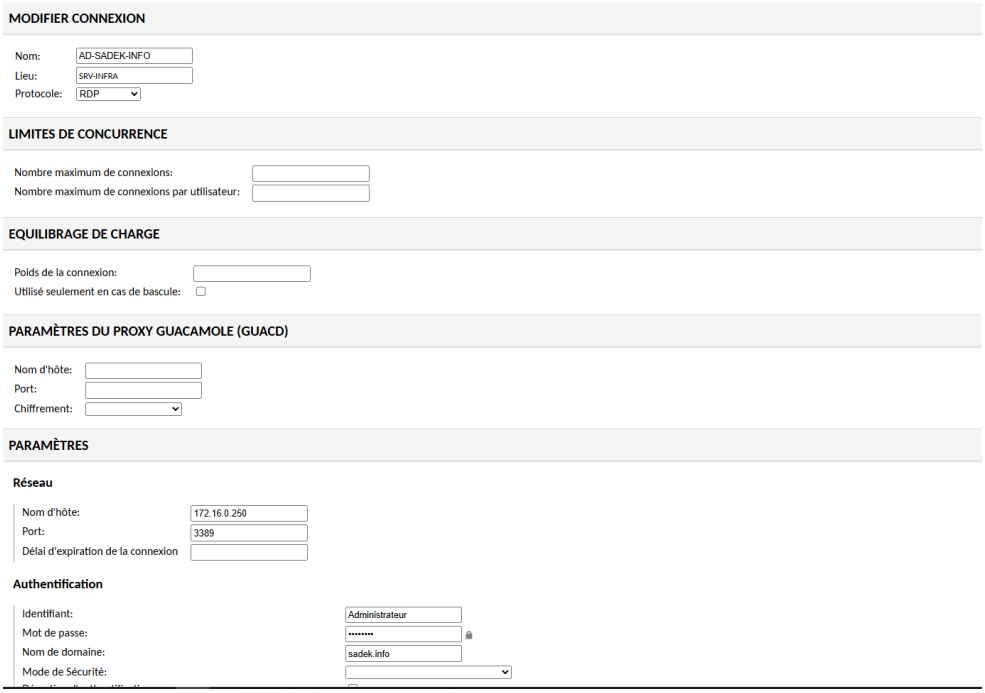


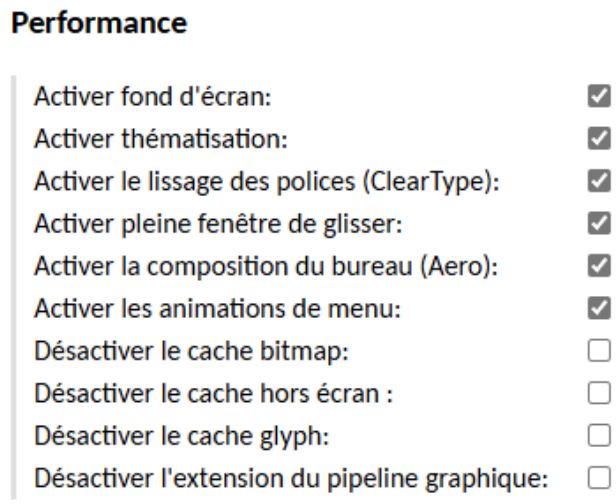
Ensuite je clique ici



Nous avons enroememnt d’options qui peuvent être ajouté je ne vais mettre que les plus importantes

Sadek Adel 09/10/2025



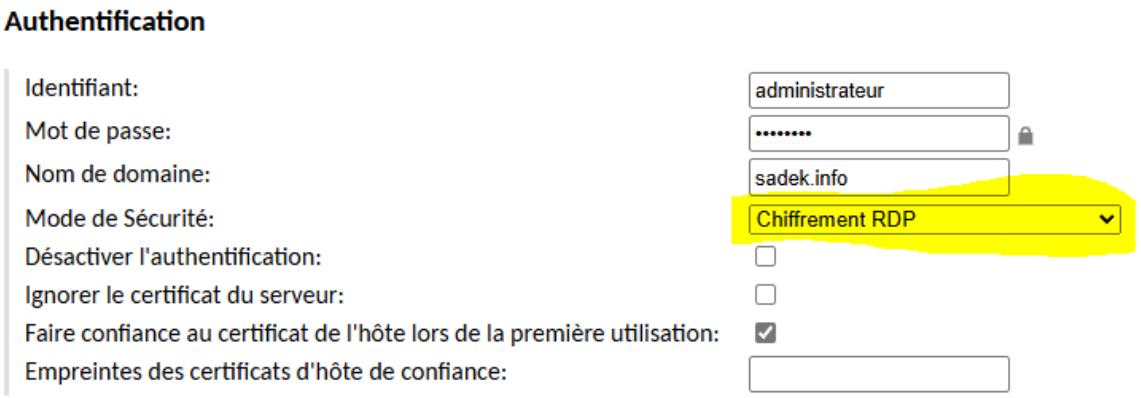


On a meme des options pour enregistrer l’ecran lors des sessions

Ensuite j’enregistre

Ensuite il faut revenir en mode accueil et appuyer sur le serveur et ça va se lancer

Sadek Adel 09/10/2025



Problème de connexion RDP entre Guacamole et le serveur Windows

Lors de la mise en place de Guacamole avec le protocole RDP, la connexion échouait systématiquement avec l'erreur suivante dans les logs :

RDP server closed/refused connection: Security negotiation failed (wrong security type?)

Ce message indiquait un désaccord entre le mode de sécurité utilisé par le serveur Windows et celui attendu par Guacamole. Par défaut, Windows applique un niveau de sécurité élevé (SecurityLayer=2), correspondant à **NLA (Network Level Authentication)**. Ce mode exige une authentification réseau préalable, basée sur NTLM ou Kerberos, avant d'établir la session graphique.

Or, Guacamole ne prend pas en charge NLA sans configuration supplémentaire (intégration LDAP/Kerberos). Le serveur refusait donc la négociation TLS avant même l'étape d'authentification.

Diagnostic et cause identifiée

L'analyse des journaux guacd a confirmé l'origine du problème :

RDP server closed/refused connection: Security negotiation failed (wrong security type?)

Le client Guacamole tentait une négociation TLS classique, mais le serveur imposait NLA.
Le paramètre responsable de ce comportement se situe dans le registre Windows à l'emplacement suivant :

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp

La clé SecurityLayer détermine le protocole de chiffrement utilisé par le service Bureau à distance :

Valeur Signification **Niveau de sécurité Compatibilité Guacamole**

Sadek Adel 09/10/2025

Valeur	Signification	Niveau de sécurité	Compatibilité Guacamole
0	RDP Security (RC4)	Faible	Compatible
1	TLS Security	Moyen à bon	Compatible
2	NLA (TLS + Kerberos/NTLM)	Élevé	Non compatible sans AD

Solution appliquée

La solution immédiate a consisté à modifier la clé SecurityLayer pour autoriser les connexions TLS standard sans authentification NLA préalable :

Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp" -Name "SecurityLayer" -Value 1

Un redémarrage du serveur (ou du service Bureau à distance) a ensuite permis de valider le changement.

Après cette modification, la connexion RDP via Guacamole a pu être établie correctement, le serveur acceptant désormais une négociation TLS classique.

Considérations de sécurité

Cette solution de contournement permet la compatibilité avec Guacamole mais réduit légèrement le niveau de sécurité.
Le chiffrement RC4 utilisé dans le mode RDP (valeur 0) est obsolète et vulnérable. Même si le mode TLS (1) assure un chiffrement AES, l'absence d'authentification réseau (NLA) laisse la surface d'attaque légèrement plus exposée, notamment aux tentatives d'accès par force brute.

Ce mode reste acceptable dans un environnement interne ou isolé (LAN/VPN), à condition de :

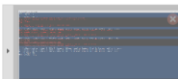
- restreindre l'accès au port TCP 3389 uniquement au serveur Guacamole,
- désactiver l'exposition du service RDP sur Internet,
- et surveiller les journaux d'authentification.

Actions futures

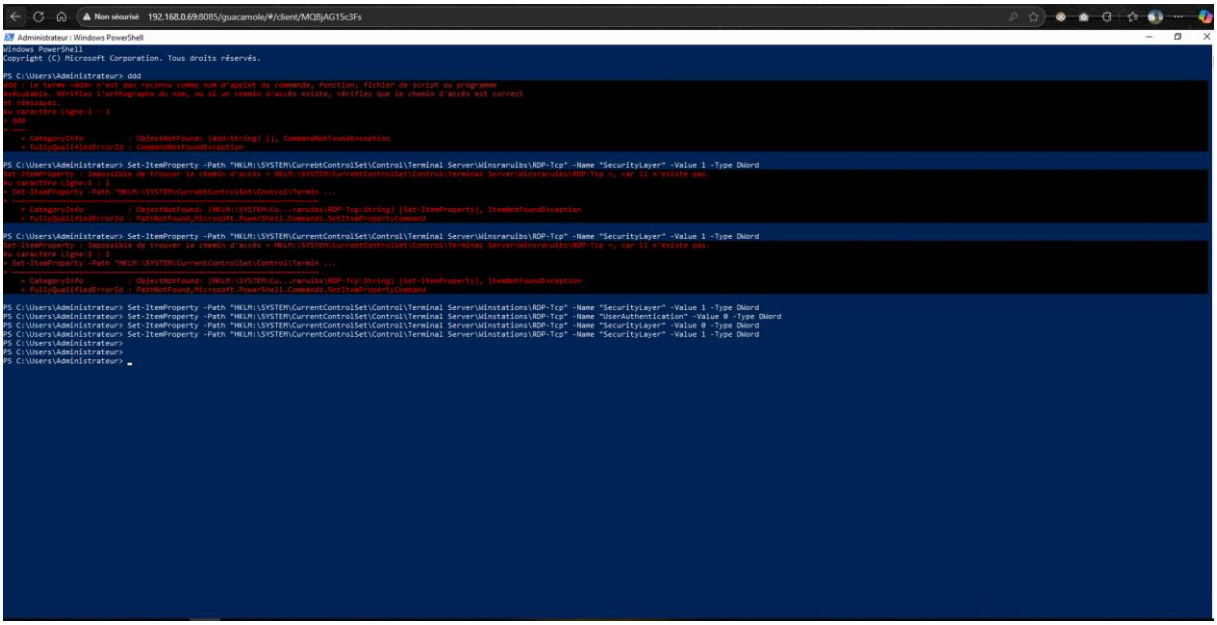
À moyen terme, il sera nécessaire d'étudier l'intégration de Guacamole avec **Active Directory et Kerberos** afin de réactiver le mode SecurityLayer=2 (NLA).
Cette configuration permettrait de bénéficier d'un chiffrement TLS complet couplé à une authentification préalable côté domaine, garantissant un niveau de sécurité conforme aux bonnes pratiques Microsoft.

Sadek Adel 09/10/2025

Voila il suffit juste d’appuyer ici



Voila nous sommes connecter via une fenetre web en RDP à notre AD



Sadek Adel 09/10/2025

Tentative de passage à une connexion sécurisée (TLS) via intégration au domaine Active Directory

Dans le but de sécuriser les connexions RDP utilisées par **Guacamole**, une tentative a été menée pour remplacer l'authentification non chiffrée (RDP Security Layer = 0) par une connexion basée sur **Kerberos** via **TLS**.
L'objectif était d'intégrer la machine hébergeant Guacamole (srv-ollama) au **domaine Active Directory sadek.info**, afin de permettre une authentification mutuelle sécurisée et la négociation automatique de certificats.

Étapes réalisées

Préparation DNS et résolution du domaine

- Le DNS local de la machine Linux a été configuré pour pointer vers le contrôleur de domaine :
- nameserver 172.16.0.250
- search sadek.info
- Vérification de la visibilité LDAP du domaine :
- dig _ldap._tcp.sadek.info SRV
- realm discover sadek.info

Ouverture du pare-feu Active Directory

- Les ports LDAP (389), LDAPS (636), Kerberos (88, 464) et RPC (135) ont été ouverts sur le DC adsadek.sadek.info.

Diagnostic du contrôleur de domaine

- Un test du rôle **RID Master** a révélé une corruption dans le service d'allocation des identifiants :
- dcdiag /test:ridmanager /v
- Le rôle RID a été réparé via **NTDSUTIL** et synchronisé à l'aide de :
- repadmin /syncall /AdeP

Intégration de la machine Linux au domaine

- Une fois le domaine fonctionnel, la machine srv-ollama a été jointe avec succès :
- realm join -v -U Administrateur sadek.info
- Le processus a créé un compte machine dans l'unité **Computers** de l'AD :
- CN=SRV-OLLAMA,CN=Computers,DC=sadek,DC=info
- Le keytab Kerberos a été généré :

- FILE:/etc/krb5.keytab
- Confirmation de l'appartenance au domaine :
- realm list

Résultat :

configured: kerberos-member

server-software: active-directory

client-software: sssd

login-formats: %U@sadek.info

Résultat

L'intégration a été effectuée avec succès : la machine est désormais enregistrée dans le domaine et dispose de son propre principal Kerberos (host/srv-ollama@SADEK.INFO).
Toutefois, **Guacamole n'exploite pas encore ce ticket Kerberos** pour initier une session RDP via TLS — la configuration du backend FreeRDP reste encore à adapter.

Limites actuelles

- La couche de sécurité **RDP native** (niveau 0) reste temporairement activée, faute d'un support TLS/Kerberos complet du côté serveur.
- Le chiffrement RDP standard utilise **RC4 128 bits**, aujourd'hui considéré comme faible.
- Les identifiants transitent donc encore dans une session protégée uniquement par le chiffrement RDP, sans authentification mutuelle.

Suite prévue

1. Configurer le **chiffrement TLS** sur le serveur Windows (niveau de sécurité = 1).
2. Tester l'authentification Kerberos en session RDP depuis Guacamole.
3. Forcer Guacamole à utiliser /sec:tls ou /sec:nla une fois la négociation Kerberos fonctionnelle.
4. Vérifier la validité du ticket avec :
5. klist

On genere maintenant un ticket kerberos avec kinit

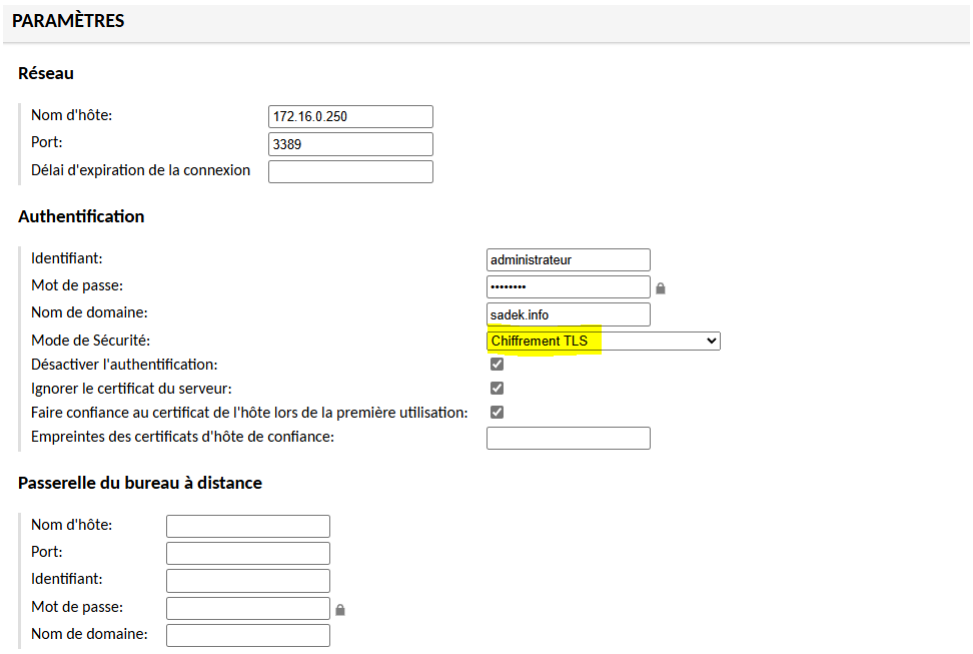
```
root@srv-ollama:/var/lib/tomcat9/webapps# kinit Administrateur@SADEK.INFO
Password for Administrateur@SADEK.INFO:
root@srv-ollama:/var/lib/tomcat9/webapps# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: Administrateur@SADEK.INFO

Valid starting       Expires              Service principal
10/10/2025 00:18:33  10/10/2025 10:18:33  krbtgt/SADEK.INFO@SADEK.INFO
        renew until 11/10/2025 00:18:29
root@srv-ollama:/var/lib/tomcat9/webapps#
```

La solution RDP

Le probleme viendrai du compte guacd qui n'est pas créer sur notre machine linux il est censé lancer le service guacd sa créer donc un conflit il fautr le créer et lui donner l'autorisation sur le dossier guacd et effectuer une modif dans le fichier guacd.service

```
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
```

Ensuite je mets chiffrement TLS



Sadek Adel 09/10/2025

Ensuite au niveau des logs

```
root@srv-ollama:/var/lib/tomcat9/webapps# sudo systemctl restart guacd
root@srv-ollama:/var/lib/tomcat9/webapps# tail -f /var/log/syslog
Oct 10 02:35:22 srv-ollama guacd[11642]: guacd[11670]: INFO:#011Loading keymap "base"
Oct 10 02:35:22 srv-ollama guacd[11642]: guacd[11670]: INFO:#011Loading keymap "base_altgr"
Oct 10 02:35:22 srv-ollama guacd[11642]: guacd[11670]: INFO:#011Loading keymap "fr-fr-azerty"
Oct 10 02:35:22 srv-ollama guacd[11670]: Loading keymap "base_altgr"
Oct 10 02:35:22 srv-ollama guacd[11670]: Loading keymap "fr-fr-azerty"
Oct 10 02:35:24 srv-ollama dhclient[20493]: DHCPDISCOVER on br-e39b5b6ffdb9 to 255.255.255.255 port 67 interval 18
Oct 10 02:35:27 srv-ollama guacd[11670]: Connected to RDPDR 1.13 as client 0x0002
Oct 10 02:35:27 srv-ollama guacd[11642]: guacd[11670]: INFO:#011Connected to RDPDR 1.13 as client 0x0002
Oct 10 02:35:27 srv-ollama guacd[11670]: RDPDR user logged on
Oct 10 02:35:27 srv-ollama guacd[11642]: guacd[11670]: INFO:#011RDPDR user logged on
Oct 10 02:35:28 srv-ollama dhclient[993]: DHCPDISCOVER on br-e39b5b6ffdb9 to 255.255.255.255 port 67 interval 11
Oct 10 02:35:39 srv-ollama dhclient[993]: DHCPDISCOVER on br-e39b5b6ffdb9 to 255.255.255.255 port 67 interval 13
```

On voit bien que c’est l’user guacd qui cherche à exécuter les commandes or qu’avant il n’existait pas

Me voila connecter enfin

Le **raccourci clavier CTRL + ALT + MAJ** donne accès au presse-papiers et à d'autres options pour gérer la connexion distante. D'ailleurs, en étant ici, si l'on clique sur le nom d'utilisateur et sur "**Accueil**", la **session reste active, mais elle se réduit en bas à droite de l'écran**.

Mise en place de la double authentification

Il faut rajouter un module TOTP à guacamole comme ceci

Sadek Adel 09/10/2025

cd /tmp

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

ensuite tar -xvf

Puis :

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

Maintenant, on doit configurer l'extension à partir du fichier "**guacamole.properties**" que l'on va éditer sans plus attendre :

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

Dans ce fichier, il y a 4 paramètres que l'on peut déclarer pour configurer l'extension TOTP. Même s'ils ne sont pas obligatoires, ils permettent de personnaliser le déploiement. Ils sont expliqués dans la documentation officielle :

- Apache Guacamole - Paramètres TOTP

Dans le fichier, on va déclarer 4 paramètres :

- **totp-issuer** : le nom avec lequel apparaîtra Apache Guacamole dans votre application TOTP.
- **totp-digits** : nombre de chiffres pour le code à usage unique - entre 6 et 8, par défaut c'est 6.
- **totp-period** : durée pendant laquelle est valide chaque code, par défaut 30.
- **totp-mode** : l'algorithme de hachage utilisé, entre sha1, sha256 et sha512 - par défaut c'est sha1.

Ce qui donne cela

Sadek Adel 09/10/2025

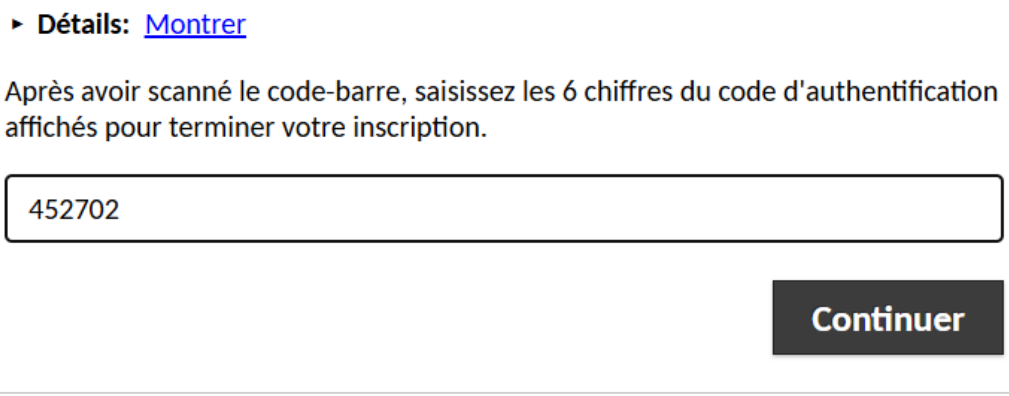
```
#TOTP

totp-issuer: Guacamole SADEK-IT
totp-digits: 6
totp-period: 30
totp-mode: sha1
```

Ensuite on redemarre le service

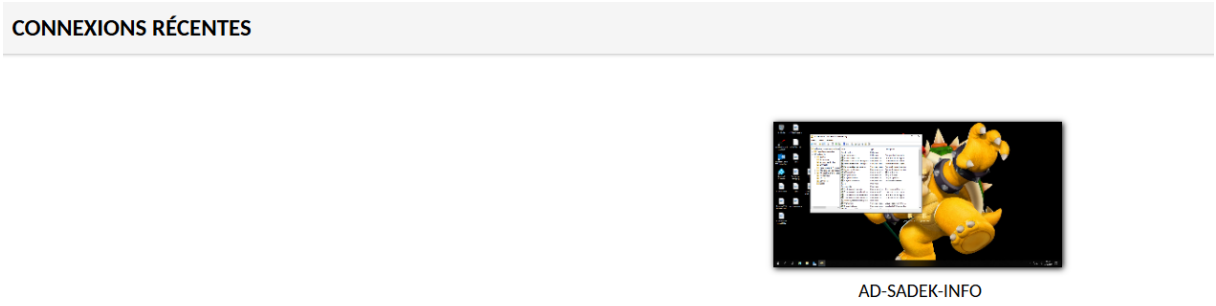
```
systemctl restart tomcat9
```

Ensuite tout ce passe au niveau de l'interface web des qu'on se connectera avec notre utilisateur une fenetre s'affichera et nous demandera de configurer le MFA

Sadek Adel 09/10/2025

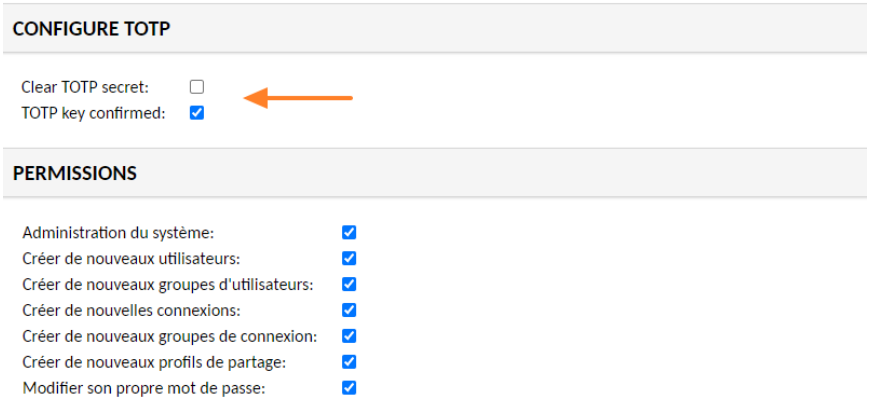


Me voila connecter maintenant



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 questio

Sadek Adel 09/10/2025



Problème rencontré :
Guacamole ne parvenait pas à établir une connexion SSH, affichant dans les logs :
SSH handshake failed.
Le handshake échouait avant l'authentification.

Cause probable :
Incompatibilité entre les algorithmes cryptographiques utilisés par libssh2 (employé par Guacamole) et ceux autorisés par OpenSSH sur le serveur distant.

Solution temporaire :
Ajout de compatibilité rétroactive dans /etc/ssh/sshd_config :

KexAlgorithms +diffie-hellman-group1-sha1,diffie-hellman-group14-sha1

HostKeyAlgorithms +ssh-rsa

PubkeyAcceptedKeyTypes +ssh-rsa

Puis redémarrage du service SSH.

Solution pérenne :
Mettre à jour libssh2 (⩾ 1.11) ou Guacamole pour bénéficier du support des algorithmes modernes (rsa-sha2-256, curve25519-sha256).

Symptômes observés :

- Message SSH handshake failed dans /var/log/syslog.
- Aucune tentative d'authentification visible côté serveur.
- Connexion SSH manuelle possible depuis ssh, mais pas depuis Guacamole.

Sadek Adel 09/10/2025

Configurer enregistrement de session

Même principe que les autres extensionson installe on decompresse et on envoie dans le repertoire /etc/guacamole/extensions

Cd /tmp

```
wget https://downloads.apache.org/guacamole/1.6.0/binary/guacamole-history-recording-storage-1.6.0.tar.gz
```

Ensuite

mv ./guacamole-history-recording-storage-1.6.0/guacamole-history-recording-storage-1.6.0.jar /etc/guacamole/extensions/

On redemarre ensuite le serveur tomcat 9

L'extension est intégrée à Apache Guacamole.

Ensuite, il faut configurer l'espace de stockage. Ici, ce sera un dossier sur notre serveur, mais il doit être possible d'utiliser un espace de stockage distant que l'on monte en local sur le serveur. On commence par créer le dossier pour accueillir les enregistrements :

sudo mkdir -p /var/lib/guacamole/recordings

Puis, on définit les autorisations sur ce répertoire :

sudo chown root:tomcat /var/lib/guacamole/recordings

sudo chmod 2750 /var/lib/guacamole/recordings

Ici, on détermine "root" comme utilisateur propriétaire, car le service "**guacd**" tourne par défaut avec cet utilisateur. Quant au groupe propriétaire, il s'agit de "tomcat" pour que notre serveur Tomcat9 soit en mesure de lire les enregistrements vidéos.

Désormais, il reste à **configurer l'enregistrement vidéo sur une connexion** à partir de l'interface web de Guacamole.

On va éditer une connexion existante et s'intéresser à la section "**Enregistrement écran**". Il y a trois paramètres à configurer :

- **Chemin de l'enregistrement** :

Sadek Adel 09/10/2025

${HISTORY_PATH}/${HISTORY_UUID}

Chaque enregistrement sera stocké dans un sous-dossier de "**/var/lib/guacamole/recordings**" qui aura un UUID de session comme nom. Grâce à l'extension installée précédemment, **Apache Guacamole peut faire la correspondance entre les sessions et les enregistrements afin de nous proposer la lecture depuis le Web**. Cette correspondance est effectuée par le nom du répertoire qui intègre l'UUID. L'alternative consiste à utiliser "**${HISTORY_UUID}**" comme nom d'enregistrement pour faire la correspondance.

- **Nom de l'enregistrement** :

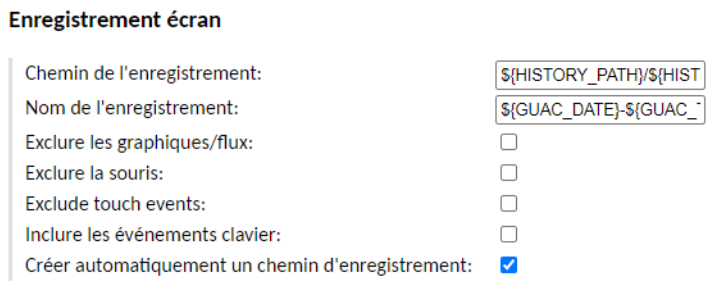
${GUAC_DATE}-${GUAC_TIME} - RDP - ${GUAC_USERNAME}

Ceci va permettre de nommer l'enregistrement avec la date, l'heure, le terme "RDP" et l'utilisateur qui s'est connecté.

- **Créer automatiquement un chemin d'enregistrement** :

À cocher, pour que le répertoire avec le nom de l'UUID soit créé.

Ce qui donne :



Ensuite je lance une connexion j'ai configuré sur machine AD

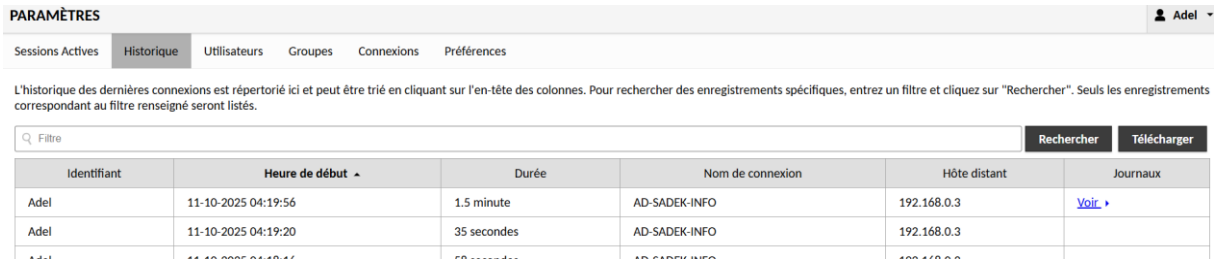
Probleme enregistrement permission denied

- Il faut exécuter cette commande chmod 777 ../recordings/
- Il faut ensuite cocher autoriser l'ecriture dans un fichier existant sinon sa ne marchera pas

Ensuite si l'on va dans parametre > historique

On peut voir un « view » à coter de la derniere session cela nous permettra de voir l'enregistrement video

Sadek Adel 09/10/2025



Je clique dessus

Je vois maintenant tout ce qui se passe sur le serveur

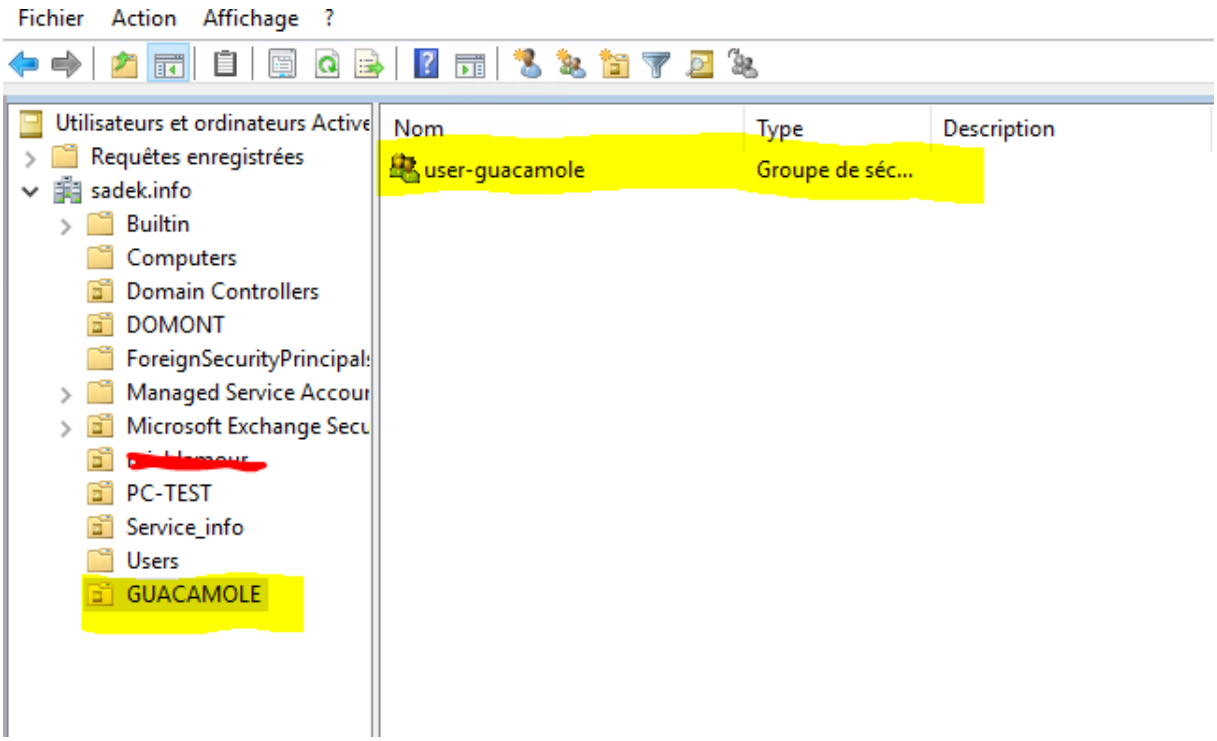
Si l'on souhaite **exporter un enregistrement**, il faut le **convertir en ligne de commande** au préalable. En effet, le format de base n'est pas lisible directement.
Apache Guacamole intègre l'outil "**guacenc**" prévu à cet effet pour **créer un fichier vidéo au format M4V**. Pour convertir un enregistrement en fichier de sortie de qualité HD, on utilisera cette commande :

```
sudo guacenc -s <résolution> -f <fichier à convertir>
sudo guacenc -s 1280x720 -f "/var/lib/guacamole/recordings/fdf244e0-cdd9-3fa7-ab2d-03773b22ba5c/20230616-100730 - RDP - admin.fb"
```

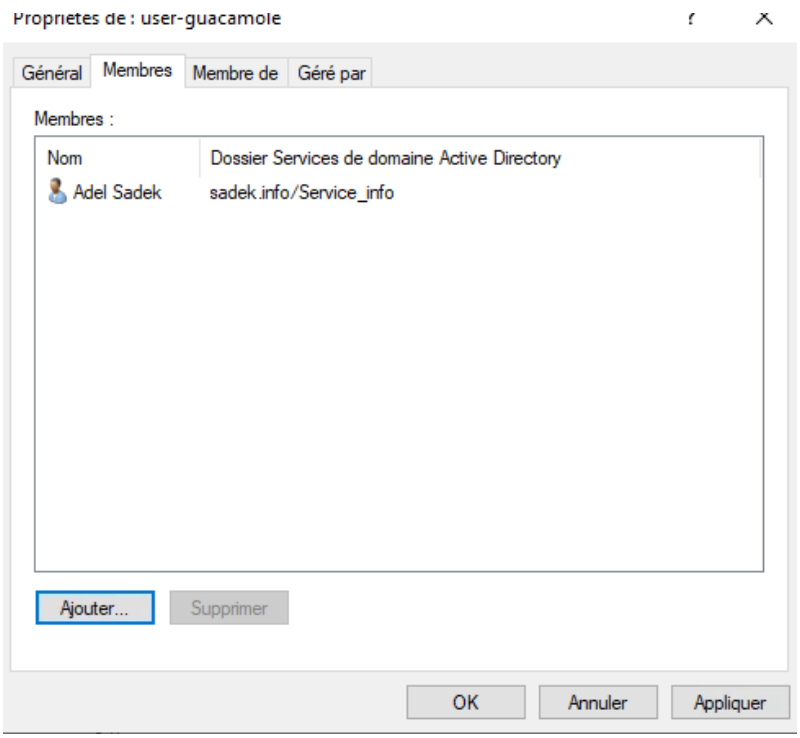
Authentication LDAP Active directory

D’abord je créer une OU guacamole avec un groupe user-guacamole

Sadek Adel 09/10/2025



Je rajoute l'user asadek comme membre du groupe



Il faut installer l'extension pour l'authentification LDAP

Sadek Adel 09/10/2025

Comme d’habitude

Cd /tmp

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

Ensuite on desarchive avec la commande tar et on transfere dans le dossier extension

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

Ensuite on restart le serveur tomcat9

Ensuite on doit conencter guacamole à l’active directory

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

```
###ACTIVE DIRECTORY
#Controleur de domaine
ldap-hostname: adsadek.sadek.info
ldap-port: 389
ldap-encryption-method: none
# Utilisateur pour connexion AD
ldap-search-bind-dn: administrateur@sadek.info
ldap-search-bind-password: [redacted]
```

Troisièmement, vous devez indiquer **comment rechercher les utilisateurs dans l'Active Directory**. Comme base DN, c'est-à-dire comme **base de recherche**, on évite de mettre la racine de l'annuaire Active Directory.
Ici, l'OU "*OU=Tiering,OU=IT,DC=it-connect,DC=local*" qui contient d'autres sous-OU ainsi que les comptes d'administration sera ciblée. Tout ce qui est en dehors de cette racine "*ne sera pas vu*" par Guacamole. Ceci correspond au paramètre "**ldap-user-base-dn**". Ensuite, le paramètre "**ldap-username-attribute**" sert à spécifier l'attribut AD utilisé pour les noms d'utilisateurs.
Enfin, le paramètre "**ldap-user-search-filter**" permet de déclarer le filtre de recherche. Ici, on prend tous les utilisateurs qui sont membres du groupe de sécurité Active Directory "**GDL-Guacamole-Access**".

```
# Recherche des utilisateurs
ldap-user-base-dn:OU=GUACAMOLE,DC=sadek,DC=info
ldap-username-attribute: sAMAccountName
ldap-user-search-filter: (&(objectClass=User)(sAMAccountName=*)(memberOf:1.2.840.113556.1.4.1941:=CN=user-guacamole,OU=GUACAMOLE,DC=sadek,DC=info))
```

Ensuite on redemarre le serveur

Attention erreur élargir champs mieu que restreindre

Il faut mieux directement mettre la recherche d'user à la base directement et ensuite via le filtre restreindre à une OU

```
# Recherche des utilisateurs
ldap-user-base-dn: DC=sadek,DC=info
ldap-username-attribute: sAMAccountName
ldap-user-search-filter: (&(objectClass=User)(sAMAccountName=*)(memberOf:1.2.840.113556.1.4.1941:=CN=user-guacamole,OU=GUACAMOLE,DC=sadek,DC=info))
```

Ensuite il faut se conencter avec l'user sans domaine et mettre le mdp AD

On verra que nous n'avons accès à aucune connexion il faut octroyer ce droit via l'interface admin donc il faut se deconncetr et se reconencter

Que faire ?

Il faut que l'on ait un compte administrateur de Guacamole qui existe aussi dans l'Active Directory et qui soit dans le périmètre de nos filtres LDAP. Par exemple, vous pouvez créer

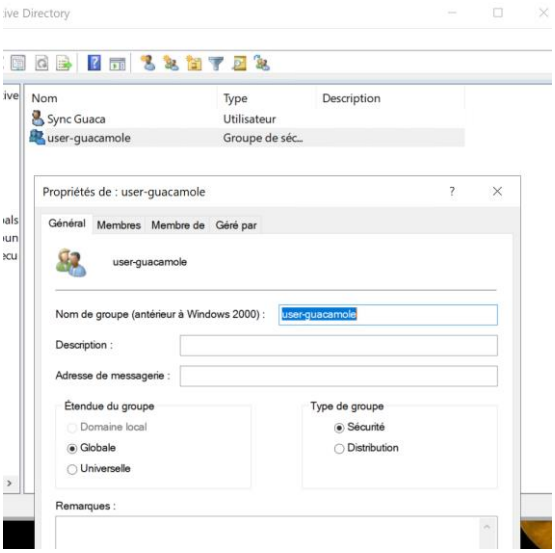
le compte "**Sync_Guacamole**" dans Apache Guacamole en tant que nouvel administrateur (attribuez-lui tous les droits).

Attention, si vous mettez le même identifiant que dans l'AD ainsi que le même mot de passe, vous allez avoir des surprises. En effet, Guacamole recherche en priorité dans la base MySQL donc s'il parvient à vous authentifier via la base MySQL, il n'ira pas chercher dans l'AD donc les objets utilisateurs et groupes n'apparaîtront pas dans Guacamole.

Vous avez deux solutions :

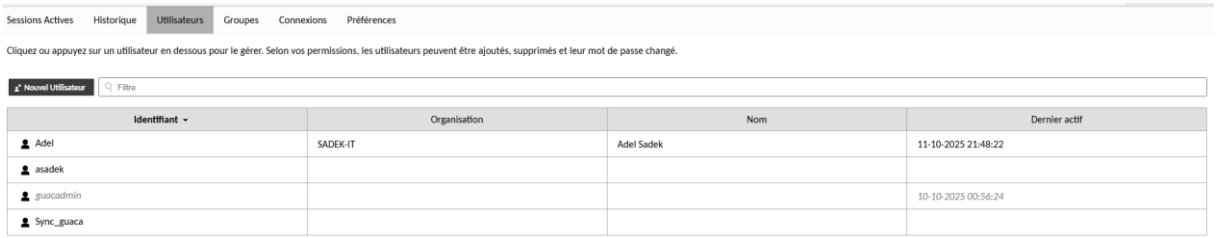
- Utiliser un mot de passe robuste différent pour le compte dans Guacamole (comme ça, la connexion MySQL échouera donc il fera une tentative LDAP et là, ça fonctionnera)
- Modifier le fichier "*guacamole.properties*" pour prioriser l'utilisation de LDAP comme source d'authentification en ajoutant cette ligne :

```
extension-priority: ldap
```



Désormais, quand on se connecte avec **Sync_Guacamole** **(EN METTANT BIEN LE MDP DE L'USER DANS L'AD),** on voit bien **les utilisateurs de l'Active Directory** :

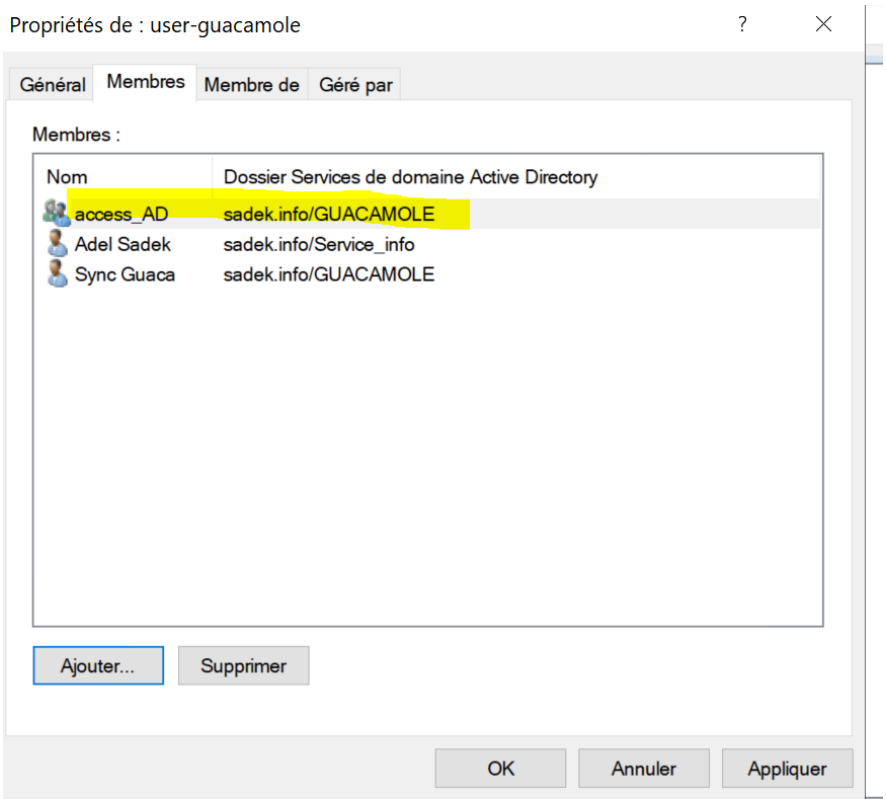
Sadek Adel 09/10/2025



Ensuite on peut enfin accorder à chaque user dans l'AD des machines auxquelles il peut accéder

Par exemple je vais créer un groupe d'accès pour l'AD

Je vais juste créer un groupe dans l'OU qui s'appel access_AD et mettre dans ce groupe un user



Ensuite dans guacamole dans le fichier de conf rajouter ces lignes

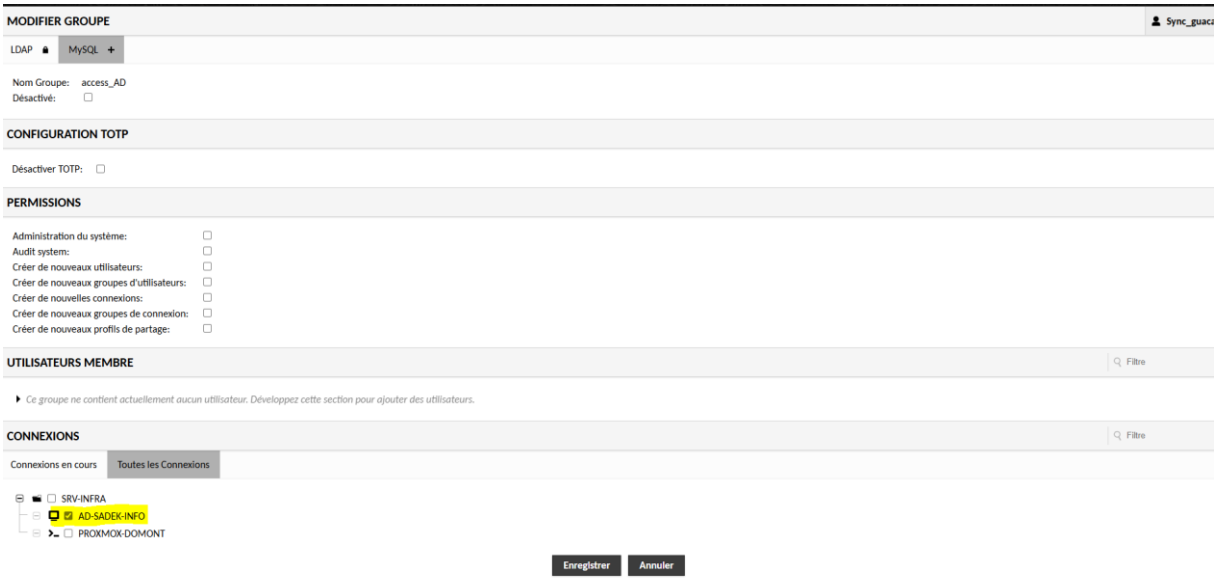
Sadek Adel 09/10/2025

```
# Recherche des groupes
ldap-group-base-dn: OU=GUACAMOLE,DC=sadek,DC=info
ldap-group-search-filter: (objectClass=group)
```

Sa permet de définir le scope des groupes et quel type d'objet sont ce que nous recherchons en l'occurrence ici des groupes

Et la je vois mon nouveau groupe

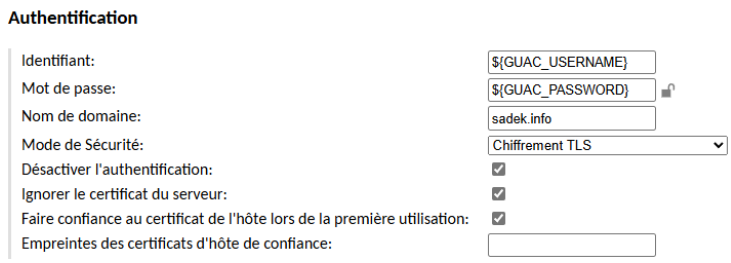
On ne voit pas les users membres mais il faut pas s'inquieter sa appliquera quand meme sur les users



Je me co avec l'user asadek qui n'est censé que pouvoir se connecter à l'AD

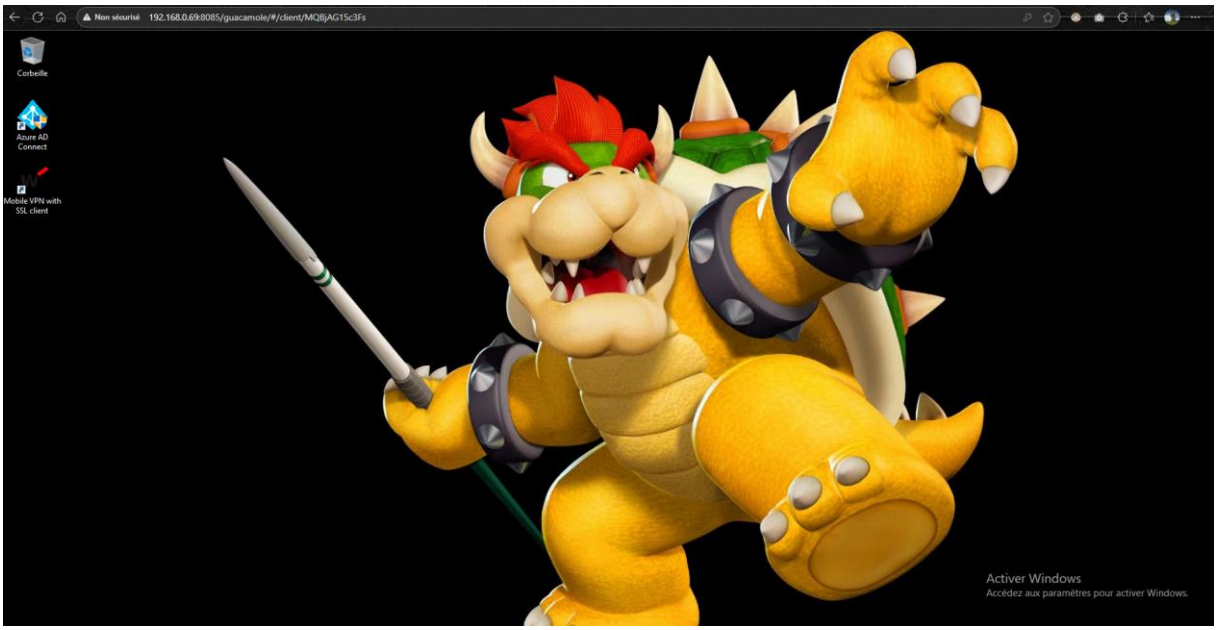
Sadek Adel 09/10/2025

Il faut aussi au préalable maintenant qu'on utilise des users de l'active diectory faire en sorte que la connexion RDP s'effectue avec l'user/mdp de celui qui se connecter il faut donc aller sur la confgiuration de la amchine à qui on soiuhaite donner l'accès et rajouter **${GUAC_USERNAME}** et **${GUAC_PASSWORD}** pour la connexion



Ensuite lorsque je me connecte avec l'utilisateur ASADEK si je n'ai accès qu'a une seule machine sa me connecte automatiquement à cette machine

Me voila connecter



Sadek Adel 09/10/2025