Zafar sio2

05/01/2025

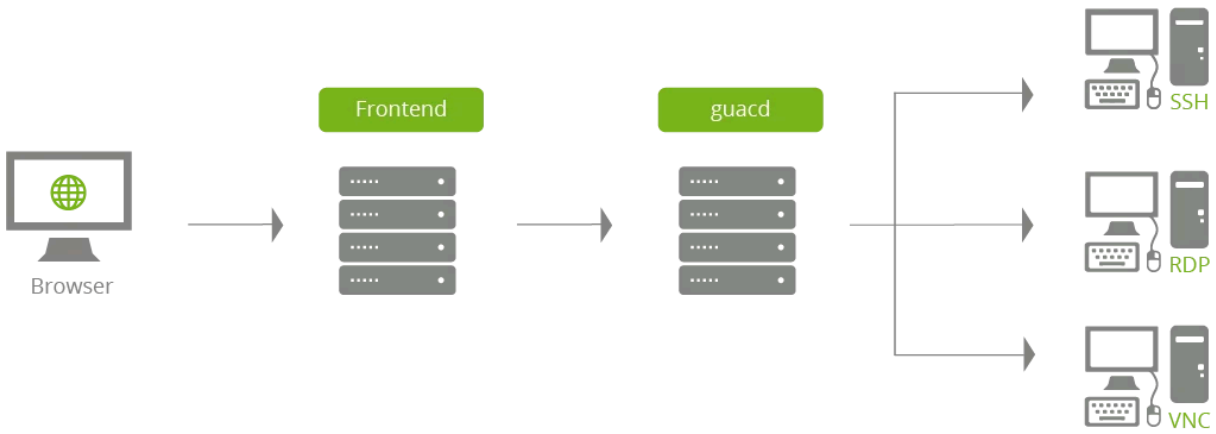
Projet Apache-Guacamole

Contexte et Objectif du Projet

Dans ce rapport nous allons installer et configurer Apache Guacamole, une solution open source et gratuite que l'on peut mettre en place en tant que bastion d'administration, passerelle d'accès ou encore serveur de rebond.
Une machine sous Ubuntu 24.05 sera utilisée pour héberger l'application.

Configuration
Apache Guacamole

Le serveur Apache Guacamole sera utilisé comme point d'entrée unique la zone DMZ de Daudruy pour accéder aux serveurs et équipements de l'infrastructure que ce soit via les protocoles RDP, SSH, VNC et Telnet, et même Kubernetes. Que l'on soit en externe ou en interne, les connexions aux serveurs vont obligatoirement passer par l'hôte Apache Guacamole.



Apache Guacamole devient un élément central de l'infrastructure puisqu'il sert de passerelle pour administrer les machines. il est possible d'avoir plusieurs hôtes Apache Guacamole pour répartir la charge et assurer la haute disponibilité mais dans notre cas on as pas trop de charge 10 utilisateur sur 8 serveur.

Enfin, les règles de pare-feu doivent aussi être adaptées : l'hôte Apache Guacamole doit être le seul à pouvoir se connecter en RDP/SSH/VNC/Etc. sur les machines de l'infrastructure.

II. Les fonctions clés d'Apache Guacamole

- 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

III. Installer Apache Guacamole sur Debian

A. Installer les prérequis d'Apache Guacamole

Tout d'abord, nous devons installer un ensemble de paquets indispensables au bon fonctionnement d'Apache Guacamole. Certains paquets sont spécifiques à certaines fonctionnalités, comme les connexions RDP par exemple. Cette liste de dépendance est consultable dans la documentation.

[Installing Guacamole natively — Apache Guacamole Manual v1.5.5]

Sur la machine Ubuntu, on commence par installer ces fameuses dépendances avec les bonne version de 2025 avec la commande suivante :

Créer un super utilisateur pour travailler la dessu :

```
zafar@apache-guaca:~$
```

B. Compiler et installer Apache Guacamole "Server"

La partie "Serveur Apache Guacamole" doit être téléchargée et compilée en local pour s'installer. La dernière version sera utilisée, à savoir la version 1.5.5. Pour identifier la dernière version, nous pouvons nous appuyer sur ces deux liens :

- Historique des versions d'Apache Guacamole
- Télécharger les sources d'installation d'Apache Guacamole

On va se positionner dans le répertoire "/tmp" et télécharger l'archive tar.gz :

```
zafar@apache-guaca:/tmp$ wget https://downloads.apache.org/guacamole/1.5.5/source/guacamole-server-1.5.5.tar.gz
z
--2025-01-17 12:53:41--  https://downloads.apache.org/guacamole/1.5.5/source/guacamole-server-1.5.5.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 135.181.214.104, 88.99.208.237, 2a01:4f9:3a:2c57::2, .
..
Connecting to downloads.apache.org (downloads.apache.org)|135.181.214.104|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1136892 (1,1M) [application/x-gzip]
Saving to: ‘guacamole-server-1.5.5.tar.gz’

guacamole-server-1.5.5.tar. 100%[=========================================>]   1,08M  3,39MB/s    in 0,3s

2025-01-17 12:53:42 (3,39 MB/s) - ‘guacamole-server-1.5.5.tar.gz’ saved [1136892/1136892]
```

Une fois le téléchargement terminé, on décompresse l'archive tar.gz et on se positionne dans le répertoire obtenu :

```
zafar@apache-guaca:/tmp$ tar -xzf guacamole-server-1.5.5.tar.gz
zafar@apache-guaca:/tmp$ cd guacamole-server-1.5.5/
zafar@apache-guaca:/tmp/guacamole-server-1.5.5$
```

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

```
zafar@apache-guaca:/tmp/guacamole-server-1.5.5$ sudo ./configure --with-systemd-dir=/etc/systemd/system/
```

```
  Services / tools:

     guacd ..... yes
     guacenc .... yes
     guaclog .... yes

  FreeRDP plugins: /usr/lib/x86_64-linux-gnu/freerdp2
  Init scripts: no
  Systemd units: /etc/systemd/system/

Type "make" to compile guacamole-server.

zafar@apache-guaca:/tmp/guacamole-server-1.5.5$
```

Enfin, on termine par installer le composant Guacamole Server :

```
Type "make" to compile guacamole-server.

zafar@apache-guaca:/tmp/guacamole-server-1.5.5$ sudo make
make  all-recursive
make[1]: Entering directory '/tmp/guacamole-server-1.5.5'
Making all in src/libguac
```

```
make[2]: Leaving directory '/tmp/guacamole-server-1.5.5/src/guaclog'
make[2]: Entering directory '/tmp/guacamole-server-1.5.5'
make[2]: Leaving directory '/tmp/guacamole-server-1.5.5'
make[1]: Leaving directory '/tmp/guacamole-server-1.5.5'
zafar@apache-guaca:/tmp/guacamole-server-1.5.5$ sudo make install
Making install in src/libguac
make[1]: Entering directory '/tmp/guacamole-server-1.5.5/src/libguac'
Making install in .
```

Voilà, la partie serveur d'Apache Guacamole est installée ! 👍

La commande ci-dessous sert à mettre à jour les liens entre guacamole-server et les librairies (cette commande ne retourne aucun résultat) :

```
zafar@apache-guaca:/tmp/guacamole-server-1.5.5$ sudo ldconfig
zafar@apache-guaca:/tmp/guacamole-server-1.5.5$
```

Ensuite, on va démarrer le service "guacd" correspondant à Guacamole et activer son démarrage automatique. La première commande sert à prendre en compte le nouveau service.

```
zafar@apache-guaca:/tmp/guacamole-server-1.5.5$ sudo systemctl daemon-reload
zafar@apache-guaca:/tmp/guacamole-server-1.5.5$ sudo systemctl enable --now guacd
Created symlink /etc/systemd/system/multi-user.target.wants/guacd.service → /etc/systemd/system/guacd.service.
zafar@apache-guaca:/tmp/guacamole-server-1.5.5$
```

Enfin, on vérifie le statut d'Apache Guacamole Server :

```
zafar@apache-guaca:/tmp/guacamole-server-1.5.5$ sudo systemctl status guacd
● guacd.service - Guacamole Server
     Loaded: loaded (/etc/systemd/system/guacd.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2025-01-17 12:59:01 UTC; 25s ago
       Docs: man:guacd(8)
   Main PID: 23804 (guacd)
      Tasks: 1 (limit: 9394)
```

C. Créer le répertoire de configuration

Dernière étape avant de passer à la partie cliente 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.

```
zafar@apache-guaca:/tmp/guacamole-server-1.5.5$ sudo mkdir -p /etc/guacamole/{extensions,lib}
zafar@apache-guaca:/tmp/guacamole-server-1.5.5$
```

D. Installer Guacamole Client (Web App)

Pour exécuter **Guacamole Web App**, un **serveur Tomcat 9** est nécessaire. Il permet d'héberger l'application Java et de gérer les connexions utilisateurs via un navigateur.

On install le paquet tomcat9

```
zafar@apache-guaca:/tmp/guacamole-server-1.5.5$ sudo apt-get install tomcat9 tomcat9-admin tomcat9-common tomca
t9-user
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  ca-certificates-java default-jre-headless java-common libapr1 libeclipse-jdt-core-java liblcms2-2
  libpcsclite1 libtcnative-1 libtomcat9-java openjdk-11-jre-headless
Paquets suggérés :
```

Puis, nous allons télécharger la dernière version de la Web App d'Apache Guacamole depuis le dépôt officiel:

```
zafar@apache-guaca:/tmp/guacamole-server-1.5.5$ cd /tmp
zafar@apache-guaca:/tmp$ wget https://downloads.apache.org/guacamole/1.5.5/binary/guacamole-1.5.5.war
--2025-01-17 13:03:02--  https://downloads.apache.org/guacamole/1.5.5/binary/guacamole-1.5.5.war
Resolving downloads.apache.org (downloads.apache.org)... 135.181.214.104, 88.99.208.237, 2a01:4f9:3a:2c57::2, .
..
Connecting to downloads.apache.org (downloads.apache.org)|135.181.214.104|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 17401039 (17M)
Saving to: ‘guacamole-1.5.5.war’

guacamole-1.5.5.war         100%[=======================================>]  16,59M  25,1MB/s    in 0,7s

2025-01-17 13:03:03 (25,1 MB/s) - ‘guacamole-1.5.5.war’ saved [17401039/17401039]

zafar@apache-guaca:/tmp$
```

Une fois que le fichier est téléchargé, on le déplace dans la librairie de Web App de Tomcat9 avec cette commande :

```
zafar@apache-guaca:/tmp$ sudo mv guacamole-1.5.5.war /var/lib/tomcat9/webapps/guacamole.war
zafar@apache-guaca:/tmp$ sudo systemctl restart tomcat9 guacd
zafar@apache-guaca:/tmp$
```

05/01/2025

Voilà, Apache Guacamole Client est installé ! 👍

Base de données MariaDB pour l'authentification

Guacamole utilise **MariaDB Server** sur **Ubuntu** pour stocker les informations des utilisateurs, les connexions et les configurations.

Exemple : Lorsqu'un utilisateur se connecte, Guacamole récupère ses droits et paramètres depuis MariaDB.

✅ Pourquoi MariaDB ?

- Compatible avec MySQL et Guacamole
- Rapide et sécurisé
- Facile à gérer sur Ubuntu

```
zafar@apache-guaca:/tmp$ sudo apt-get install mariadb-server
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  galera-4 libcgi-fast-perl libcgi-pm-perl libclone-perl libconfig-inifi
  libdbi-perl libencode-locale-perl libfcgi-bin libfcgi-perl libfcgi0ldb
  libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-me
  liblwp-mediatypes-perl libmariadb3 libmysqlclient21 libtimedate-perl l
```

Création de base et utilisateur :

```
zafar@apache-guaca:/tmp$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 46
Server version: 10.6.18-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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

MariaDB [(none)]> CREATE USER 'userdb'@'localhost' IDENTIFIED BY 'zafar';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> GRANT SELECT, INSERT, UPDATE, DELETE ON guacadb.* TO 'userdb'@'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
zafar@apache-guaca:/tmp$
```

05/01/2025

La suite va consister à ajouter l'extension MySQL à Apache Guacamole ainsi que le connecteur correspondant.Toujours depuis le dépôt officiel, on télécharge cette extension :

```
zafar@apache-guaca:/tmp$ wget https://downloads.apache.org/guacamole/1.5.5/binary/guacamole-auth-jdbc-1.5.5.tar
.gz
--2025-01-17 13:11:32--  https://downloads.apache.org/guacamole/1.5.5/binary/guacamole-auth-jdbc-1.5.5.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 88.99.208.237, 135.181.214.104, 2a01:4f9:3a:2c57::2, .
..
Connecting to downloads.apache.org (downloads.apache.org)|88.99.208.237|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 33099128 (32M) [application/x-gzip]
Saving to: ‘guacamole-auth-jdbc-1.5.5.tar.gz’

guacamole-auth-jdbc-1.5.5.t 100%[=========================================>]  31,57M  37,2MB/s    in 0,8s

2025-01-17 13:11:33 (37,2 MB/s) - ‘guacamole-auth-jdbc-1.5.5.tar.gz’ saved [33099128/33099128]

zafar@apache-guaca:/tmp$
```

On décompresser le fichier puis on déplace le fichier ".jar" de l'extension dans le répertoire "/etc/guacamole/extensions/" créé précédemment :

```
zafar@apache-guaca:/tmp$ tar -xzf guacamole-auth-jdbc-1.5.5.tar.gz
zafar@apache-guaca:/tmp$ sudo mv guacamole-auth-jdbc-1.5.5/mysql/guacamole-auth-jdbc-mysql-1.5.5.jar /etc/guaca
mole/extensions/
zafar@apache-guaca:/tmp$
```

Ensuite, le connecteur MySQL doit être téléchargé depuis le site de MySQL (peu importe si vous utilisez MariaDB ou MySQL).
On copie (ou déplace) le fichier .jar du connecteur vers le répertoire "lib" d'Apache Guacamole :

```
zafar@apache-guaca:/tmp$ wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-j-9.1.0.tar.gz
--2025-01-17 13:13:57--  https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-j-9.1.0.tar.gz
Resolving dev.mysql.com (dev.mysql.com)... 23.54.143.15, 2a02:26f0:2b00:3a2::2e31, 2a02:26f0:2b00:387::2e31
Connecting to dev.mysql.com (dev.mysql.com)|23.54.143.15|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://cdn.mysql.com//Downloads/Connector-J/mysql-connector-j-9.1.0.tar.gz [following]
--2025-01-17 13:13:58--  https://cdn.mysql.com//Downloads/Connector-J/mysql-connector-j-9.1.0.tar.gz
Resolving cdn.mysql.com (cdn.mysql.com)... 2.18.132.71, 2a02:26f0:2b00:382::1d68, 2a02:26f0:2b00:386::1d68
Connecting to cdn.mysql.com (cdn.mysql.com)|2.18.132.71|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4485702 (4,3M) [application/x-tar-gz]
Saving to: ‘mysql-connector-j-9.1.0.tar.gz’

mysql-connector-j-9.1.0.tar 100%[=========================================>]   4,28M  --.-KB/s    in 0,09s

2025-01-17 13:13:58 (48,9 MB/s) - ‘mysql-connector-j-9.1.0.tar.gz’ saved [4485702/4485702]

zafar@apache-guaca:/tmp$ tar -xzf mysql-connector-j-9.1.0.tar.gz
zafar@apache-guaca:/tmp$ sudo cp mysql-connector-j-9.1.0/mysql-connector-j-9.1.0.jar /etc/guacamole/lib/
zafar@apache-guaca:/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-1.5.5/mysql/schema/". Le mot de passe root de MariaDB doit être saisi pour effectuer l'import.

```
zafar@apache-guaca:/tmp$ sudo cp mysql-connector-j-9.1.0/mysql-connector-j-9.1.0.jar /etc/guacamole/lib/
zafar@apache-guaca:/tmp$ cd guacamole-auth-jdbc-1.5.5/mysql/schema/
zafar@apache-guaca:/tmp/guacamole-auth-jdbc-1.5.5/mysql/schema$ cat *.sql | mysql -u root -p guacadb
Enter password:
zafar@apache-guaca:/tmp/guacamole-auth-jdbc-1.5.5/mysql/schema$ ls -l
total 28
-rw-r--r-- 1 zafar zafar 20174 juil. 21  2021 001-create-schema.sql
-rw-r--r-- 1 zafar zafar  2876 juil. 21  2021 002-create-admin-user.sql
drwxr-xr-x 2 zafar zafar  4096 juil. 21  2021 upgrade
zafar@apache-guaca:/tmp/guacamole-auth-jdbc-1.5.5/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 6.2                          /etc/guacamole/guacamole.properties *
#declaration de de la connexion a Mariadb
#ce fichier est utile aussi pour d'autre parametres

# MySQL ------------------------
mysql-hostname: 127.0.0.1
mysql-port: 3306
mysql-database: guacadb
mysql-username: userdb
mysql-password: zafar
#------------------------------
```

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).
Communication interne : La Web App Guacamole (sur Tomcat) se connecte à **Guacd** via ce port.

```
 GNU nano 6.2                                  /etc/guacamole/guacd.conf *
#Declaration de une connexion local par defaut sur le port 4822
[server]
bind_host = 0.0.0.0
bind_port = 4822
#-----------------------------------------------------------------------
```

On redemarre tous les service

```
zafar@apache-guaca:/tmp/guacamole-auth-jdbc-1.5.5/mysql/schema$ sudo systemctl restart tomcat9 guacd mariadb
zafar@apache-guaca:/tmp/guacamole-auth-jdbc-1.5.5/mysql/schema$
```

Voilà, l'installation de base est terminée ! 👍

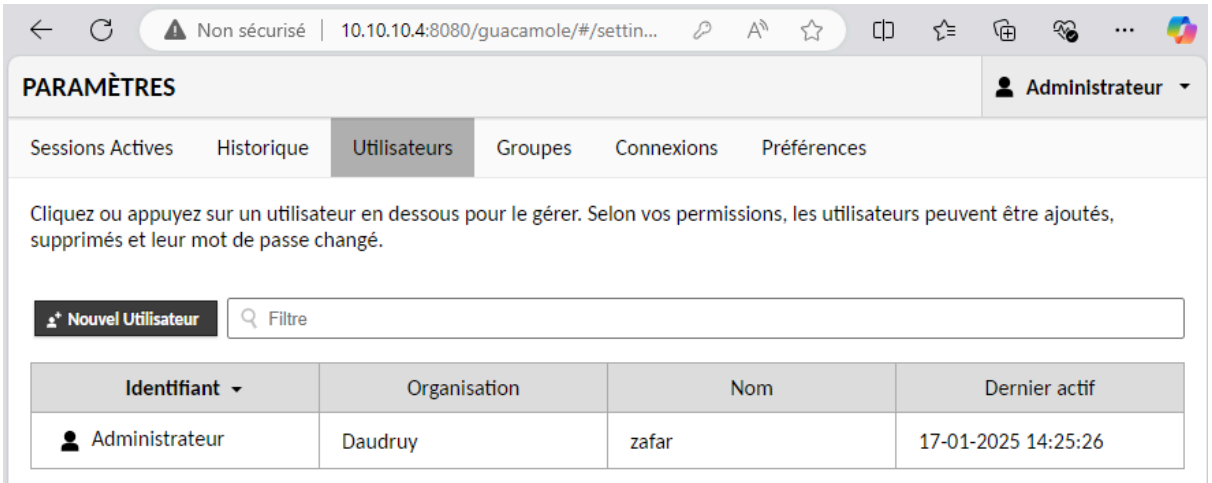
Zafar sio2

05/01/2025

Premiers pas avec Apache Guacamole

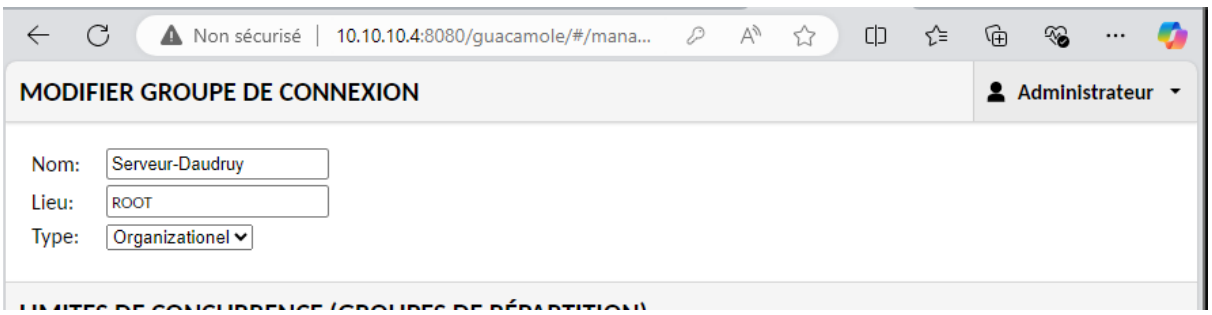
Pour se connecter, on va utiliser les identifiants par défaut : Utilisateur : guacadmin Mot de passe : guacadmin

Créer un nouveau compte admin



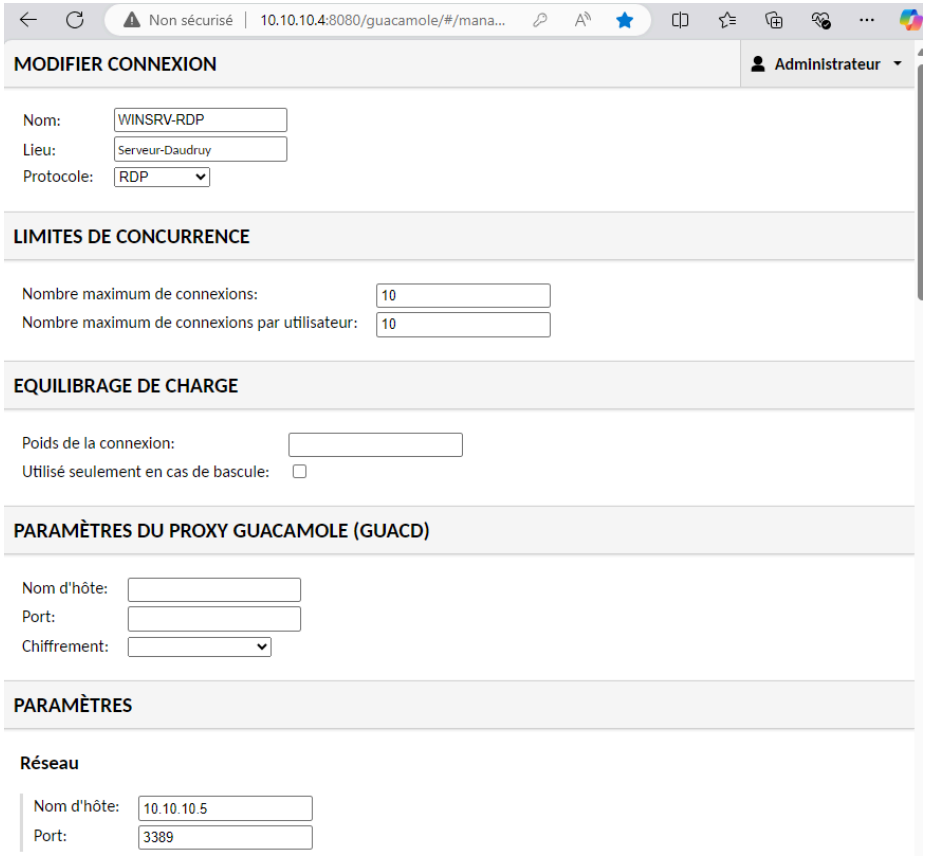
Ajouter une connexion RDP

on va créer un nouveau groupe pour organiser les machine 👍

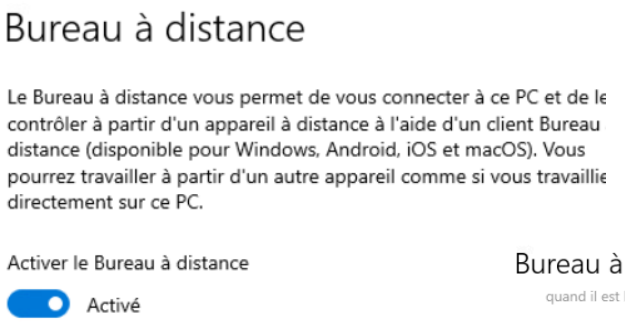


Zafar sio2

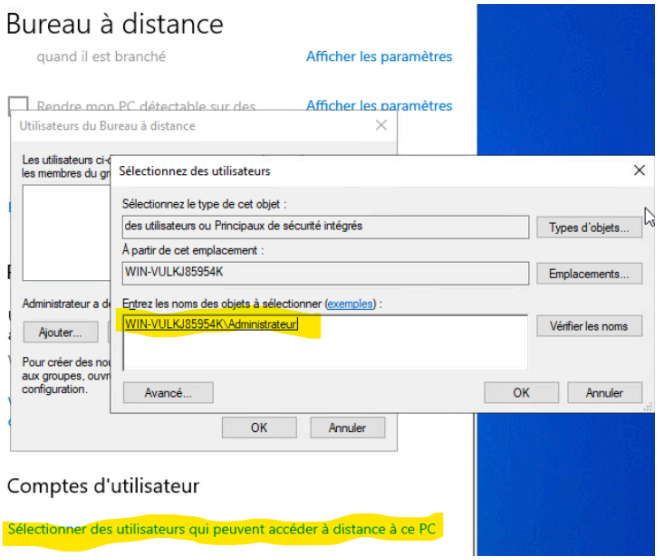
05/01/2025



Activer le Bureau à Distance (RDP) sur Windows :



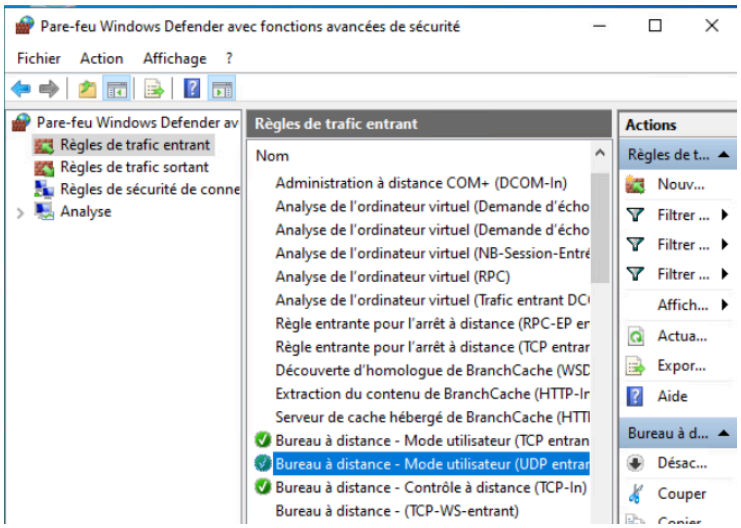
Autorise les utilisateurs RDP :



Zafar sio2

05/01/2025

Configurer le pare-feu Windows :

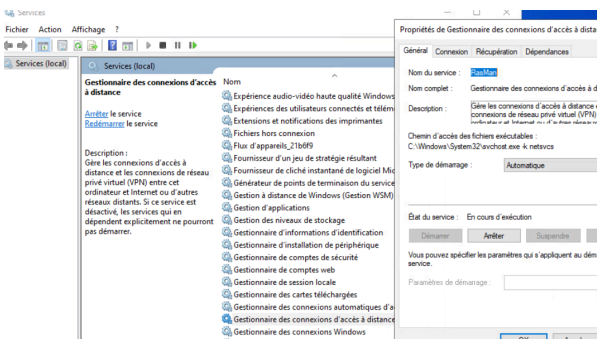


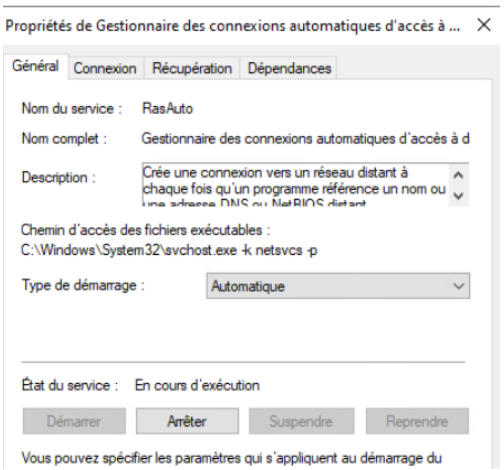

```
zafar@apache-guaca:~$ ping 10.10.10.5
PING 10.10.10.5 (10.10.10.5) 56(84) bytes of data.
64 bytes from 10.10.10.5: icmp_seq=1 ttl=128 time=0.196 ms
64 bytes from 10.10.10.5: icmp_seq=2 ttl=128 time=0.195 ms
64 bytes from 10.10.10.5: icmp_seq=3 ttl=128 time=0.212 ms
```

```
C:\Users\Administrateur>ping 10.10.10.4

Envoi d'une requête 'Ping'  10.10.10.4 avec 32 octets de données :
Réponse de 10.10.10.4 : octets=32 temps<1ms TTL=64
Réponse de 10.10.10.4 : octets=32 temps<1ms TTL=64
```

```
zafar@apache-guaca:~$ sudo systemctl restart guacd
zafar@apache-guaca:~$ sudo ufw allow 3389
Rules updated
Rules updated (v6)
zafar@apache-guaca:~$ sudo ufw allow 4822
Rules updated
Rules updated (v6)
zafar@apache-guaca:~$ sudo ufw allow 8080
Rules updated
Rules updated (v6)
zafar@apache-guaca:~$
```

On vérifie le statut du service RDP:
Services.msc puis on démarre le RDP en mode **Automatique**.



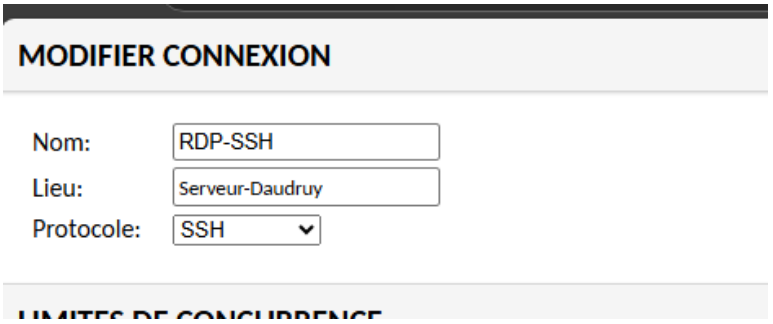


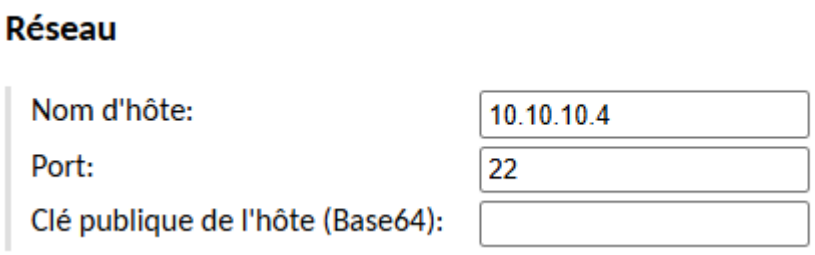
Zafar sio2

05/01/2025

Voilà nous avons une connexion en RDP sur un machine win 👍

Nous allons maintenant configurer la connexion en SSH





Jusqu' ici nous avons des configuré par défaut et des connexion et ssh et rdp

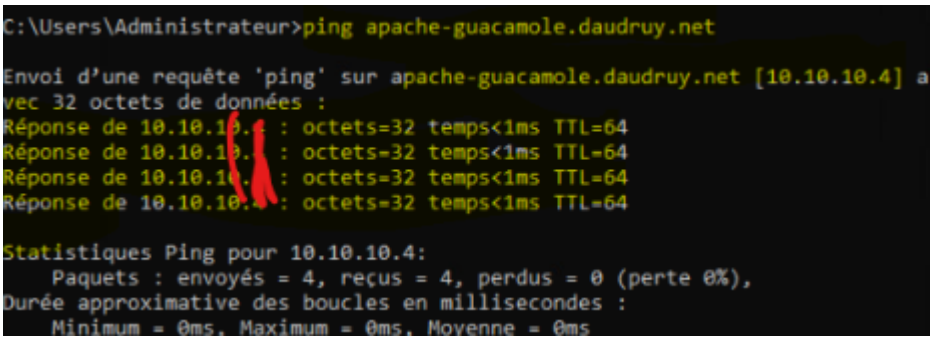
Maintenant nous allons configurer les configue avancé comme DNS Https Certificate SSL

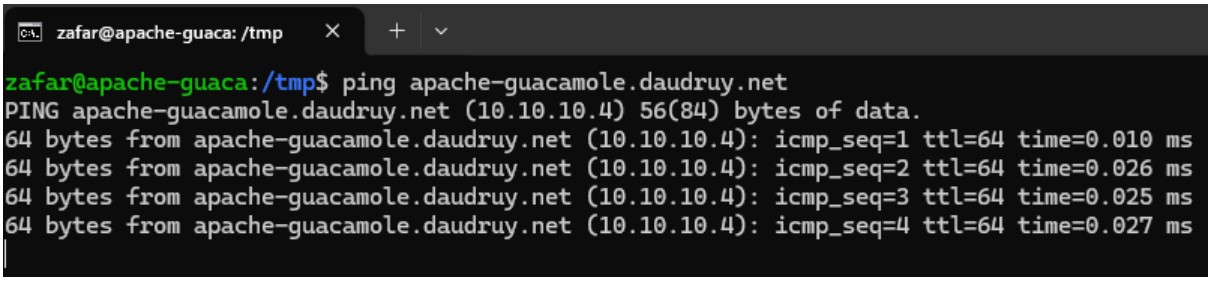
Zafar sio2

05/01/2025

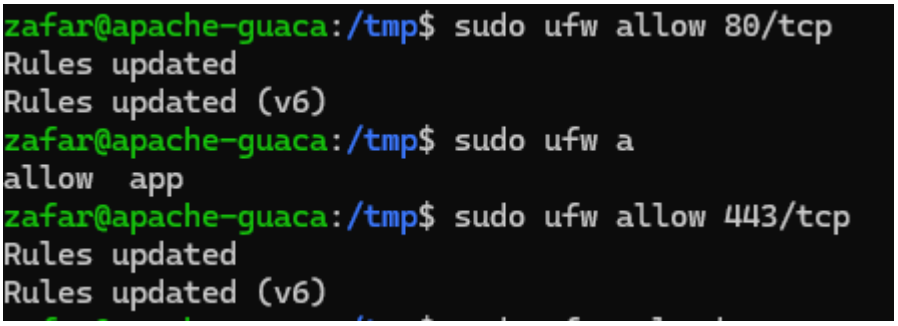
Configuration Avancé Guacamole

Le DNS est déjà configuré sur le serveur DNS, Nous allons configurer le dsn sur apache pour qu' il réponde en local

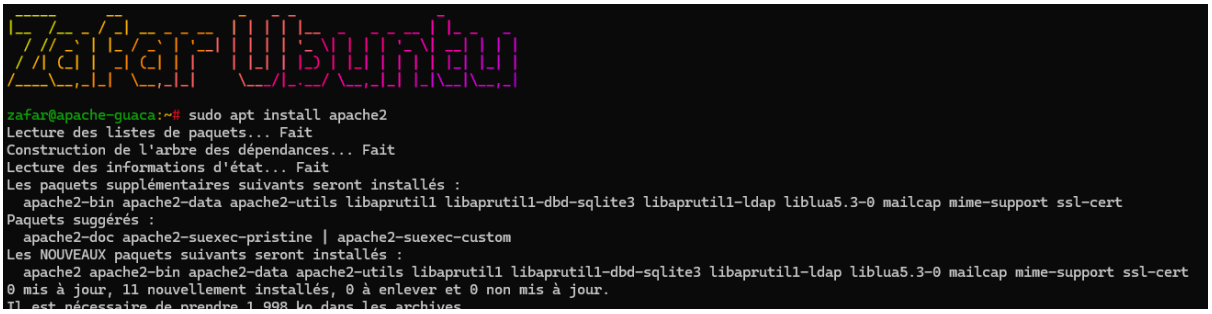




Configure firewall :



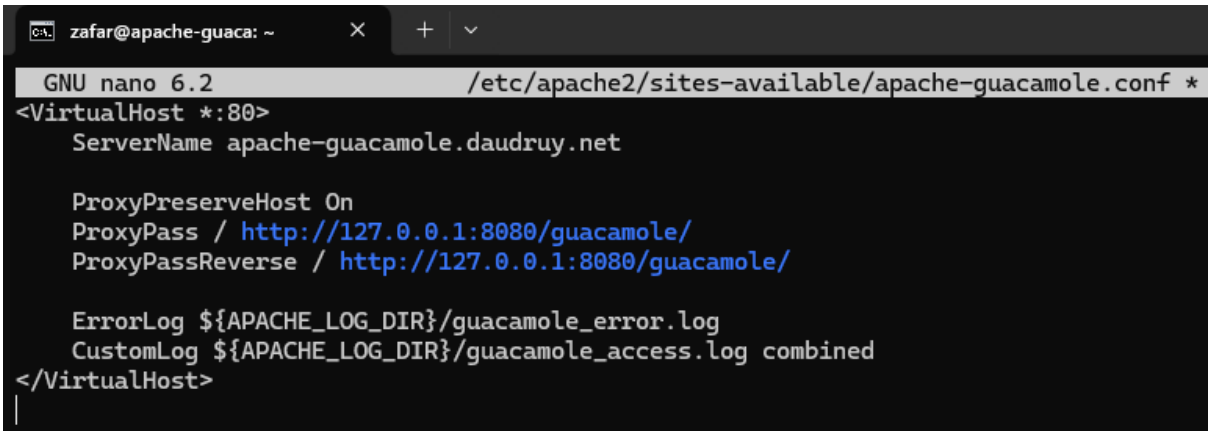
Pour la configure DNS on a besoin d'un Apache2

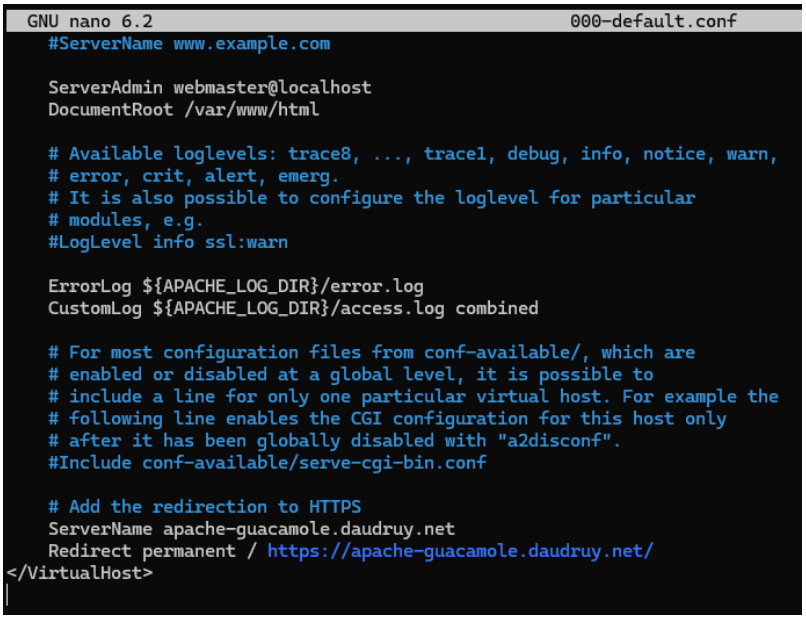


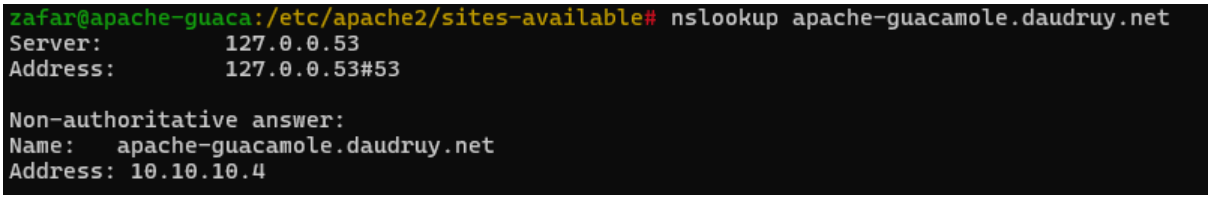
Zafar sio2

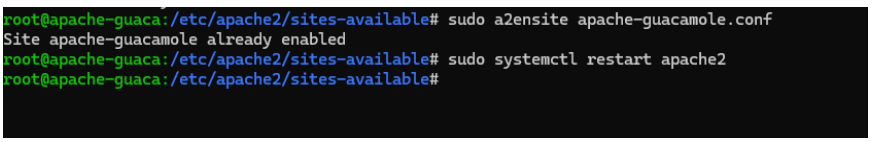
05/01/2025

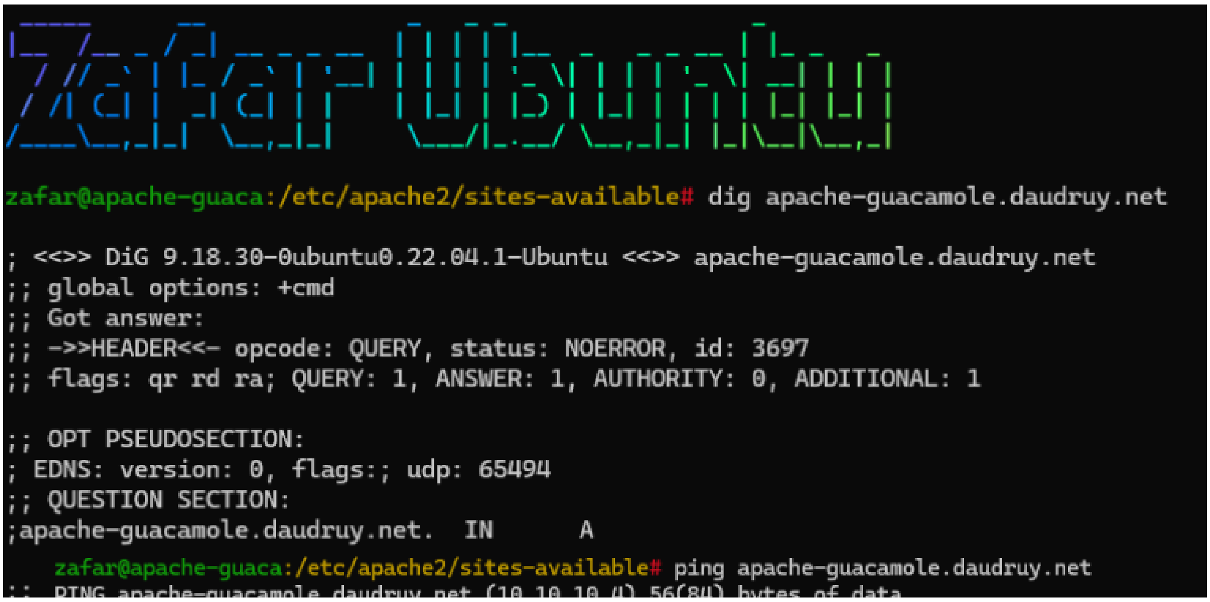
Sur Apache, la configuration des DNS se fait généralement dans les fichiers de configuration des hôtes virtuels, plus précisément dans les fichiers **`000-default.conf`** ou dans des fichiers personnalisés dans le répertoire **`/etc/apache2/sites-available/`**.
Ces fichiers sont utilisés pour définir des hôtes virtuels (Virtual Hosts) et peuvent inclure des directives qui spécifient les noms de domaine pour lesquels le serveur doit répondre.

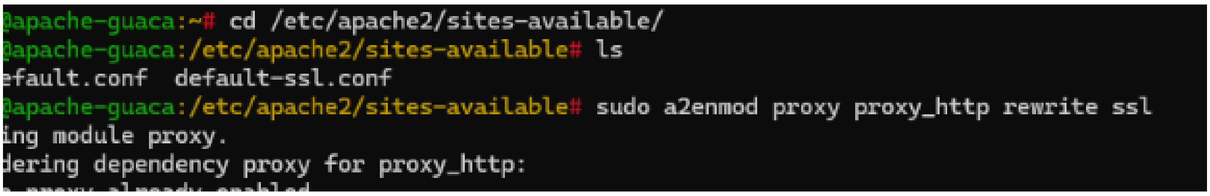




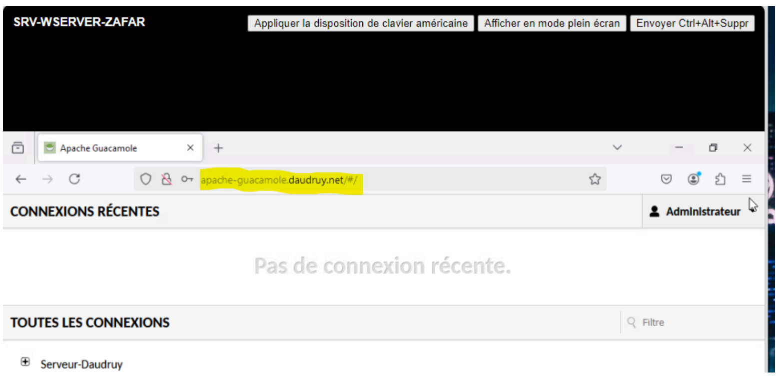








Nous allons tester la connexion DNS sur machine virtuelle en Local et la ca marche 👍

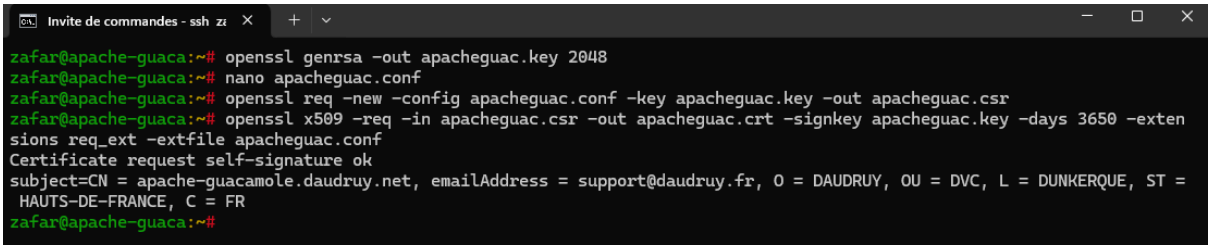


Le domaine `apache-guacamole.daudruy.net` est correctement résolu vers l'adresse IP `10.10.10.4`, confirmant que la configuration DNS est fonctionnelle.

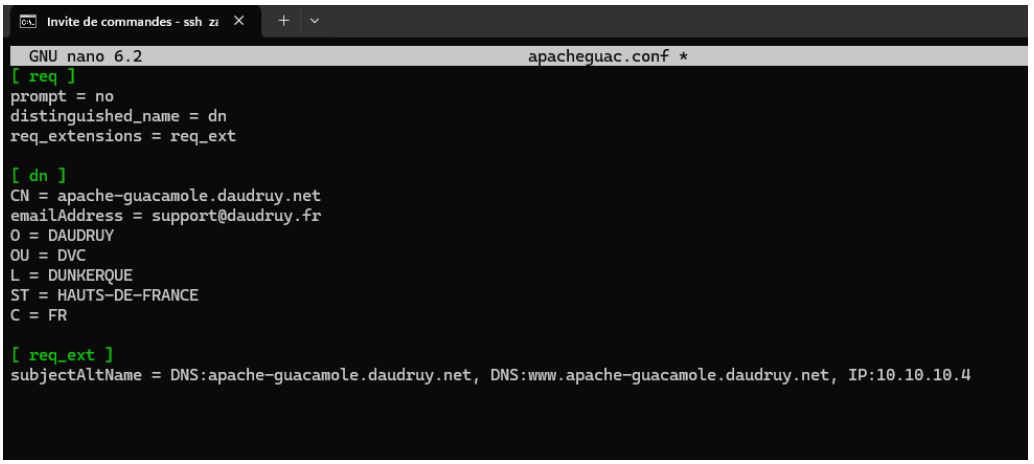
Certification SSL-HTTPS - HTTP

Solution pour avoir une connexion https sans le message de erreur de connexion https

On exécute cette commande pour générer une clé privée de 2048 bits :



On crée le fichier apache.conf Il contient la configuration **OpenSSL** pour créer un certificat auto-signé.

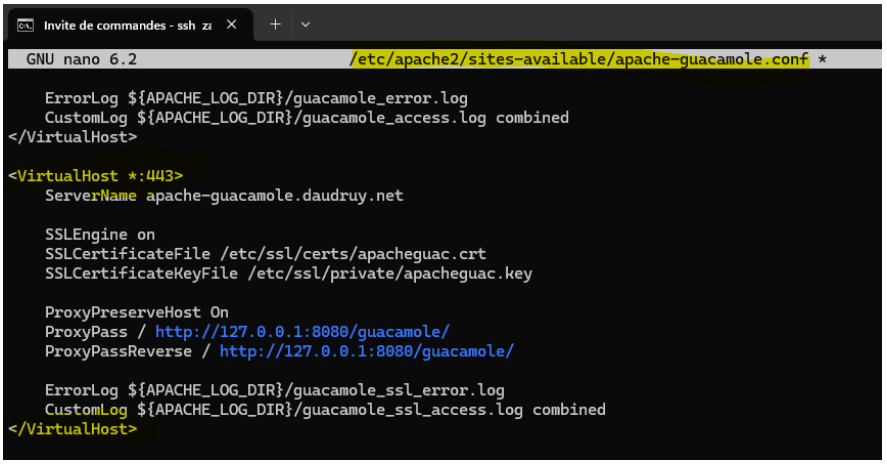


Configurer Apache pour utiliser le certificat

On déplace les fichiers générés dans les dossiers SSL d'Apache :

```
zafar@apache-guaca:~# sudo cp apacheguac.key /etc/ssl/private/
[sudo] password for zafar:
zafar@apache-guaca:~# sudo cp apacheguac.crt /etc/ssl/certs/
zafar@apache-guaca:~# sudo nano /etc/apache2/sites-available/apache-guacamole.conf
zafar@apache-guaca:~# sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
zafar@apache-guaca:~# sudo systemctl restart apache2
```

On modifié le fichier de configuration Apache et on ajoute les lignes suivantes :



Active le module SSL et redémarre Apache :

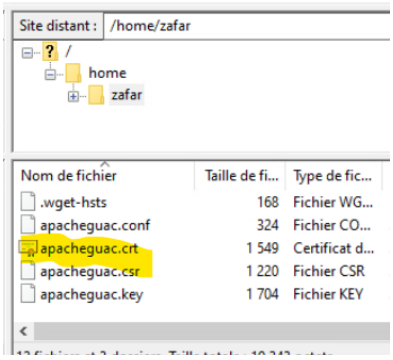
```
zafar@apache-guaca:~# sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
zafar@apache-guaca:~# sudo systemctl restart apache2
```

Zafar sio2

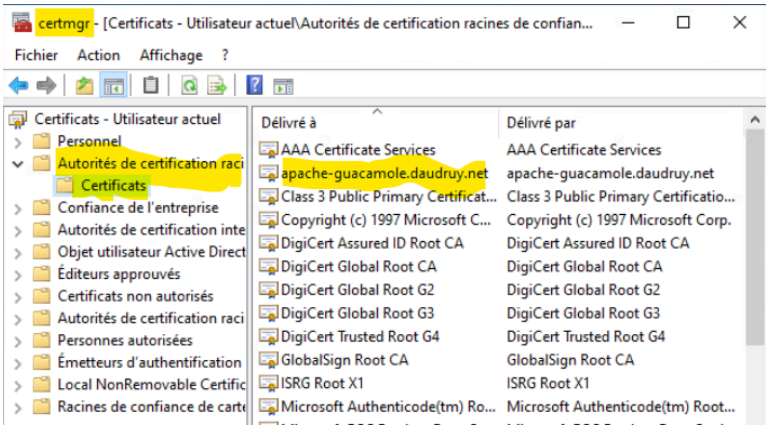
05/01/2025

Exporter le certificat et l'importer dans Windows

On récupère le certificate depuis filezilla:



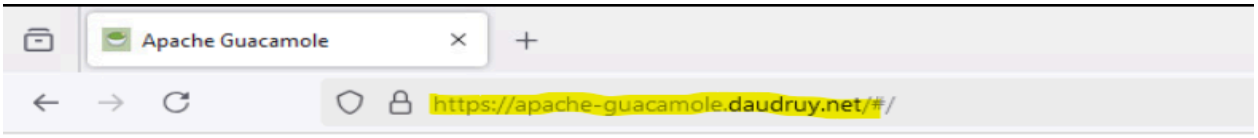
Importer le certificate dans Windows :



Zafar sio2

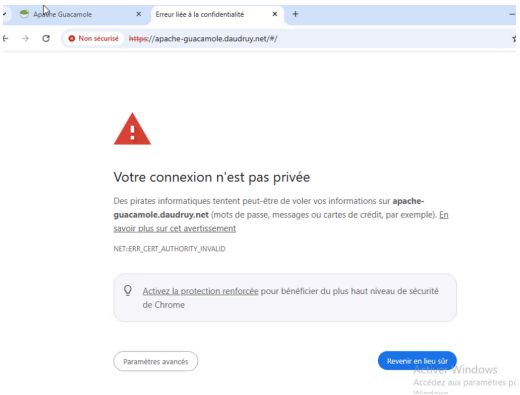
05/01/2025

On teste l'accès 👍



En résumé : On crée une clé privée, on génère un certificat SSL avec SAN via un fichier de configuration (`apacheguac.conf`), on configure Apache pour utiliser le certificat, on redémarre Apache, puis on exporte et importe le certificat dans Windows.

C'est un bon methode si on veut avoir des accès https en local sans avoir le massage sur la photo



Zafar sio2

05/01/2025

Mettre en place la double authentification TOTP

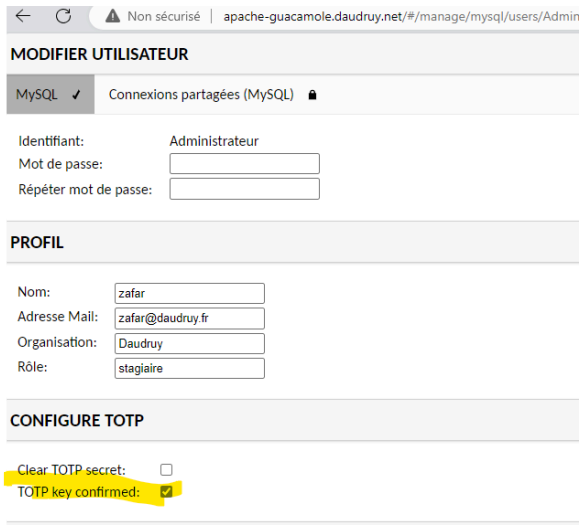
Pour bénéficier de la double authentification avec un code TOTP comme second facteur, une extension doit être ajoutée à Apache Guacamole.

```
zafar@apache-guaca:~# cd /tmp
zafar@apache-guaca:/tmp# wget https://downloads.apache.org/guacamole/1.5.5/binary/guacamole-auth-totp-1.5.5.tar.gz
--2025-01-20 15:11:12--  https://downloads.apache.org/guacamole/1.5.5/binary/guacamole-auth-totp-1.5.5.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 88.99.208.237, 135.181.214.104, 2a01:4f8:10a:39da::2, ...
Connecting to downloads.apache.org (downloads.apache.org)|88.99.208.237|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4923857 (4,7M) [application/x-gzip]
Saving to: 'guacamole-auth-totp-1.5.5.tar.gz'

guacamole-auth-totp-1.5.5.tar 100%[=============================================>]   4,70M  19,6MB/s    in 0,2s

2025-01-20 15:11:13 (19,6 MB/s) - 'guacamole-auth-totp-1.5.5.tar.gz' saved [4923857/4923857]

zafar@apache-guaca:/tmp#
```

```
zafar@apache-guaca:/tmp# tar -xzf guacamole-auth-totp-1.5.5.tar.gz
zafar@apache-guaca:/tmp# sudo mv guacamole-auth-totp-1.5.5/guacamole-auth-totp-1.5.5.jar /etc/guacamole/extensions/
[sudo] password for zafar:
zafar@apache-guaca:/tmp# sudo nano /etc/guacamole/guacamole.properties
zafar@apache-guaca:/tmp# sudo systemctl restart tomcat9
zafar@apache-guaca:/tmp# sudo nano /etc/guacamole/guacamole.properties
zafar@apache-guaca:/tmp# sudo systemctl restart tomcat9
zafar@apache-guaca:/tmp# sudo nano /etc/guacamole/guacamole.properties
zafar@apache-guaca:/tmp#
```

Invite de commandes - ssh zz

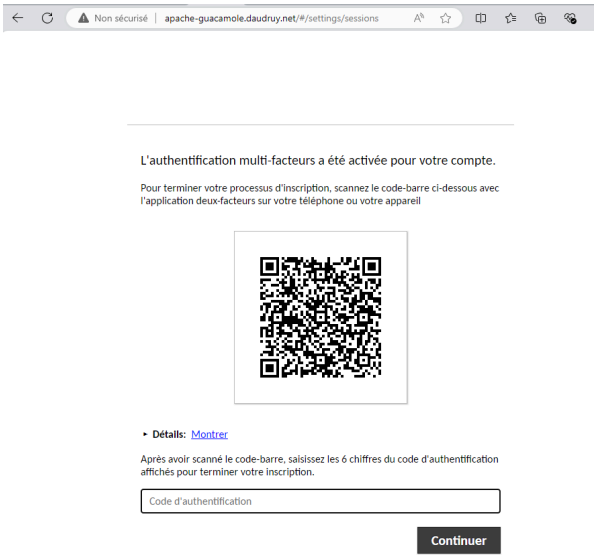
```
 GNU nano 6.2                                /etc/guacamole/guacamole.properties *
#declaration de de la connexion a Mariadb
#ce fichier est utile aussi pour d'autre parametres

# MySQL ------------------------
mysql-hostname: 127.0.0.1
mysql-port: 3306
mysql-database: guacadb
mysql-username: userdb
mysql-password: zafar
#-------------------------------



# TOTP
#ici on ajoute le nome de chaque utilisateur déja cree sur apache-guacamole
#pou le moment j'ai un compte admin
totp-issuer: Administrateur
totp-digits: 6
totp-period: 30
totp-mode: sha1
```

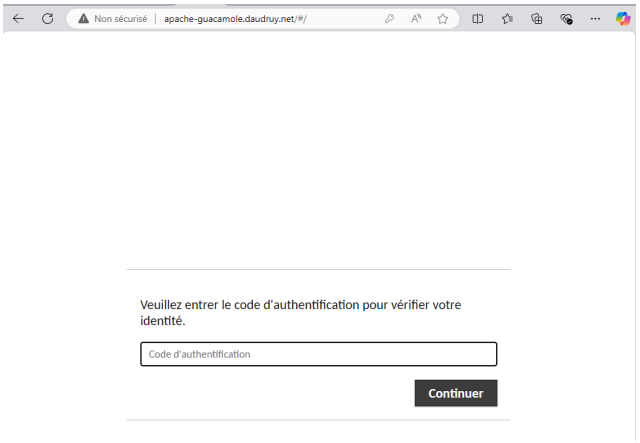


Zafar sio2

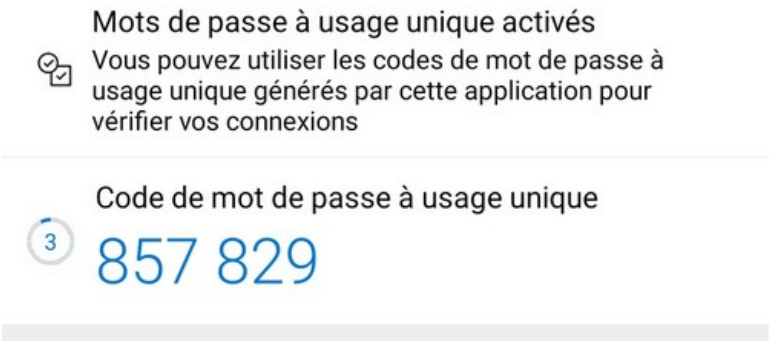
05/01/2025

On scan le code bar puis à chaque connexion il nous demande le code qui est dans le Microsoft authentificateur





Code reçu sur téléphone



05/01/2025

Configure avancé Apache-GUACAMOLE

Redirection de HTTP vers HTTPS

Objectif : Configurer Apache pour rediriger automatiquement les requêtes HTTP vers HTTPS pour le domaine `apache-guacamole.daudruy.net`.

Ici on redirige tout le trafic HTTP vers HTTPS
Ce fichier **gère les requêtes HTTP non sécurisées (port 80)** et contient :

```
zafar@apache-guaca:~# sudo nano /etc/apache2/sites-available/000-default.conf
zafar@apache-guaca:~#
```

```
  GNU nano 6.2                                          000-default.conf *
<VirtualHost *:80>

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html


    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined


    # Add the redirection to HTTPS
    ServerName apache-guacamole.daudruy.net
    Redirect permanent / https://apache-guacamole.daudruy.net/
</VirtualHost>
```

➡ Cela signifie que toutes les requêtes HTTP sont automatiquement redirigées vers HTTPS.

Zafar sio2

05/01/2025

Configuration du Proxy et SSL pour Guacamole (`apache-guacamole.conf`)

📌 But

Active le proxy vers Guacamole (port 8080)
Active **HTTPS avec un certificat SSL**
Redirige **HTTP → HTTPS**

```
 GNU nano 6.2                                    apache-guacamole.conf
<VirtualHost *:80>
    ServerName apache-guacamole.daudruy.net

    ProxyPreserveHost On
    ProxyPass / http://127.0.0.1:8080/guacamole/
    ProxyPassReverse / http://127.0.0.1:8080/guacamole/

    ErrorLog ${APACHE_LOG_DIR}/guacamole_error.log
    CustomLog ${APACHE_LOG_DIR}/guacamole_access.log combined
</VirtualHost>

<VirtualHost *:443>
    ServerName apache-guacamole.daudruy.net

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/apacheguac.crt
    SSLCertificateKeyFile /etc/ssl/private/apacheguac.key

    ProxyPreserveHost On
    ProxyPass / http://127.0.0.1:8080/guacamole/
    ProxyPassReverse / http://127.0.0.1:8080/guacamole/

    ErrorLog ${APACHE_LOG_DIR}/guacamole_ssl_error.log
    CustomLog ${APACHE_LOG_DIR}/guacamole_ssl_access.log combined
</VirtualHost>


<VirtualHost *:80>
    ServerName apache-guacamole.daudruy.net
    Redirect permanent / https://apache-guacamole.daudruy.net/
</VirtualHost>
```

Activation ssl configure

```
zafar@apache-guaca:~# sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
zafar@apache-guaca:~#   systemctl reload apache2
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to reload 'apache2.service'.
Authenticating as: admin (zafar)
Password:
==== AUTHENTICATION COMPLETE ===
zafar@apache-guaca:~#
```

Zafar sio2

05/01/2025

Configuration du Proxy pour l'IP locale (`guacamole.conf`)

📌 **But :** Accès à Guacamole via internet avec son IP et le port

```
 GNU nano 6.2                                    guacamole.conf *
<VirtualHost *:443>
    ServerName 10.10.10.4  # Remplace par ton IP publique ou nom de domaine
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/ton-certificat.crt  # Remplace avec certificat autorite le jours de mise en produ>
    SSLCertificateKeyFile /etc/ssl/private/ton-certificat.key

    ProxyPreserveHost On
    ProxyPass / http://127.0.0.1:8080/  # Guacamole tourne sur Tomcat sur le port 8080
    ProxyPassReverse / http://127.0.0.1:8080/

    ErrorLog ${APACHE_LOG_DIR}/guacamole_error.log
    CustomLog ${APACHE_LOG_DIR}/guacamole_access.log combined
</VirtualHost>
```

Pare-feu :

```
zafar@apache-guaca:~# sudo systemctl start apache2
zafar@apache-guaca:~# sudo systemctl start tomcat9
zafar@apache-guaca:~# sudo netstat -tuln | grep 8080
sudo: netstat: command not found
zafar@apache-guaca:~# sudo ufw allow 80
Rule added
Rule added (v6)
zafar@apache-guaca:~# sudo ufw allow 443
Rule added
Rule added (v6)
zafar@apache-guaca:~# sudo ufw allow 8080
Skipping adding existing rule
Skipping adding existing rule (v6)
zafar@apache-guaca:~# sudo ufw reload
Firewall reloaded
```

Résultat Final

✅ **HTTP (port 80) → HTTPS (port 443)** automatique.
✅ **Guacamole accessible via https://apache-guacamole.daudruy.net**. En local
✅ **Sécurisation avec un certificat SSL**.
✅ **Reverse Proxy fonctionnel avec Apache vers Tomcat (Guacamole sur port 8080)**.

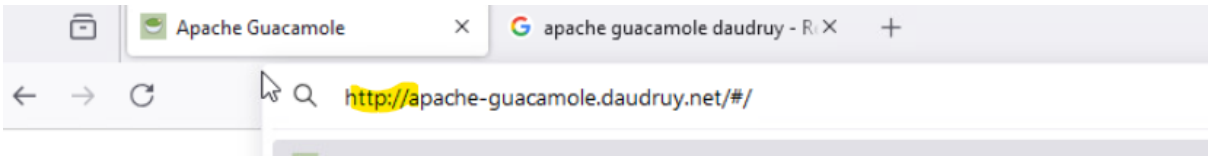
Tests réalisés

La redirection HTTP vers HTTPS est opérationnelle.
Le site est accessible uniquement via une connexion sécurisée HTTPS en local , sur internet il faut avoir un certificate

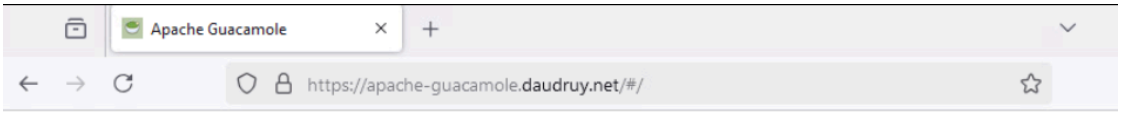
Zafar sio2

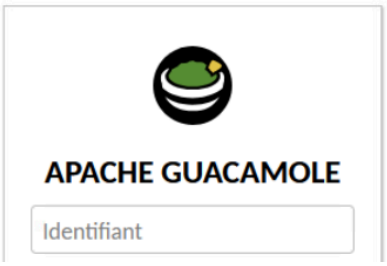
05/01/2025

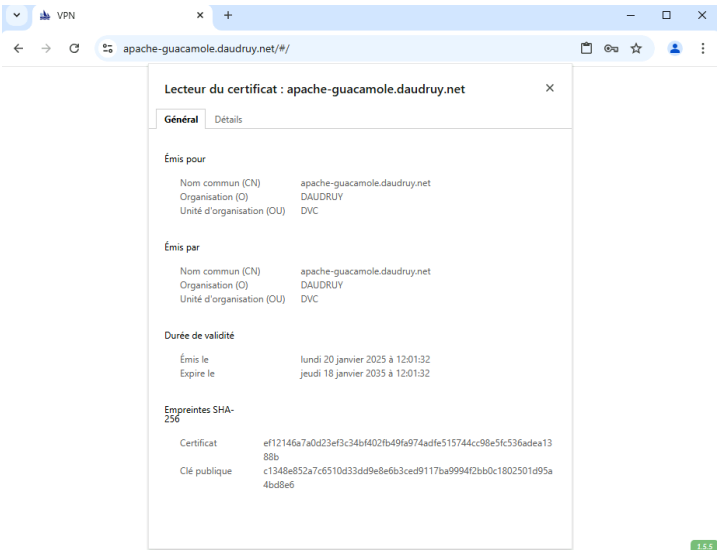
On tape http://



Et on est dirigé vers https://







05/01/2025

Personalisation Guacamole

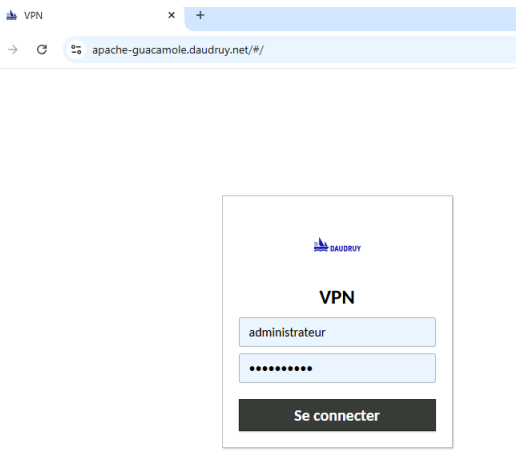
```
root@apache-guaca:/var/lib# su zafar
 _____     __                  _ _                 _
|__  /__ _ / _| __ _ _ __     | | | |__  _   _ _ __ | |_ _   _
  / // _` | |_ / _` | '__|    | | | '_ \| | | | '_ \| __| | | |
 / /| (_| |  _| (_| | |       | |_| | | | |_| | | | | |_| |_| |
/____\__,_|_|  \__,_|_|       \___/|_.__/ \__,_|_| |_|\__|\__,_|

zafar@apache-guaca:/var/lib# su -
Password:
root@apache-guaca:~# cd /var/lib/tomcat9/webapps/guacamole/translations/
root@apache-guaca:/var/lib/tomcat9/webapps/guacamole/translations# ls
ca.json  de.json  es.json  it.json  ko.json  no.json  ru.json
cs.json  en.json  fr.json  ja.json  nl.json  pt.json  zh.json
```

```
 GNU nano 6.2
{

   "NAME" : "English",

   "APP" : {

       "NAME"    : "VPN-GETWAY",
       "VERSION" : "1.5.5",
```

Changer le logo

```
root@apache-guaca:/var/lib/tomcat9/webapps/guacamole/images# chown tomcat logo-64.svg
root@apache-guaca:/var/lib/tomcat9/webapps/guacamole/images# chgrp tomcat logo-64.svg
root@apache-guaca:/var/lib/tomcat9/webapps/guacamole/images# systemctl restart guacd
root@apache-guaca:/var/lib/tomcat9/webapps/guacamole/images# systemctl restart tomcat9
root@apache-guaca:/var/lib/tomcat9/webapps/guacamole/images# ls -l
total 124
drwxr-x--- 2 tomcat tomcat 4096 janv. 17 13:03 action-icons
drwxr-x--- 2 tomcat tomcat 4096 janv. 17 13:03 arrows
-rw-r----- 1 tomcat tomcat  359 mars  29  2024 checker.svg
-rw-r----- 1 tomcat tomcat  369 mars  29  2024 checkmark.svg
-rw-r----- 1 tomcat tomcat 1408 mars  29  2024 circle-arrows.svg
-rw-r----- 1 tomcat tomcat  924 mars  29  2024 cog.svg
-rw-r----- 1 tomcat tomcat  994 mars  29  2024 drive.svg
-rw-r----- 1 tomcat tomcat  609 mars  29  2024 file.svg
-rw-r----- 1 tomcat tomcat  689 mars  29  2024 folder-closed.svg
-rw-r----- 1 tomcat tomcat  691 mars  29  2024 folder-open.svg
-rw-r----- 1 tomcat tomcat  984 mars  29  2024 folder-up.svg
drwxr-x--- 2 tomcat tomcat 4096 janv. 17 13:03 group-icons
-rw-rw-r-- 1 tomcat tomcat 2782 janv. 23 13:40 guac-tricolor.svg
-rw-r----- 1 tomcat tomcat 1180 mars  29  2024 lock.svg
-rw-r----- 1 tomcat tomcat 9167 mars  29  2024 logo-144.png
-rw-rw-r-- 1 zafar  zafar  2782 janv. 23 13:48 logo-64.svg
-rw-r----- 1 tomcat tomcat  647 mars  29  2024 magnifier.svg
```

On peut aller plus loin dans la configuration des interfaces personnalisées, mais comme ce n'est pas ma spécialité et que Guacamole utilise des langages comme JSON, JavaScript, etc., que je ne maîtrise pas totalement, cela complique les modifications. D'ailleurs, rien que la mise en place des logos m'a pris tout un après-midi à chercher dans les fichiers.



Zafar sio2

05/01/2025

Mise en place du Fail2Ban

Pour empêcher les attaques par force brute deviennent une menace sérieuse sur guacamole je me en place fail2ban

Commencez par installer Fail2Ban

```
zafar@apache-guaca:~# sudo apt update
[sudo] password for zafar:
Atteint :1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Atteint :2 http://archive.ubuntu.com/ubuntu jammy InRelease
Atteint :3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Atteint :4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
1 paquet peut être mis à jour. Exécutez « apt list --upgradable » pour le voir.
zafar@apache-guaca:~# sudo apt install fail2ban -y
```

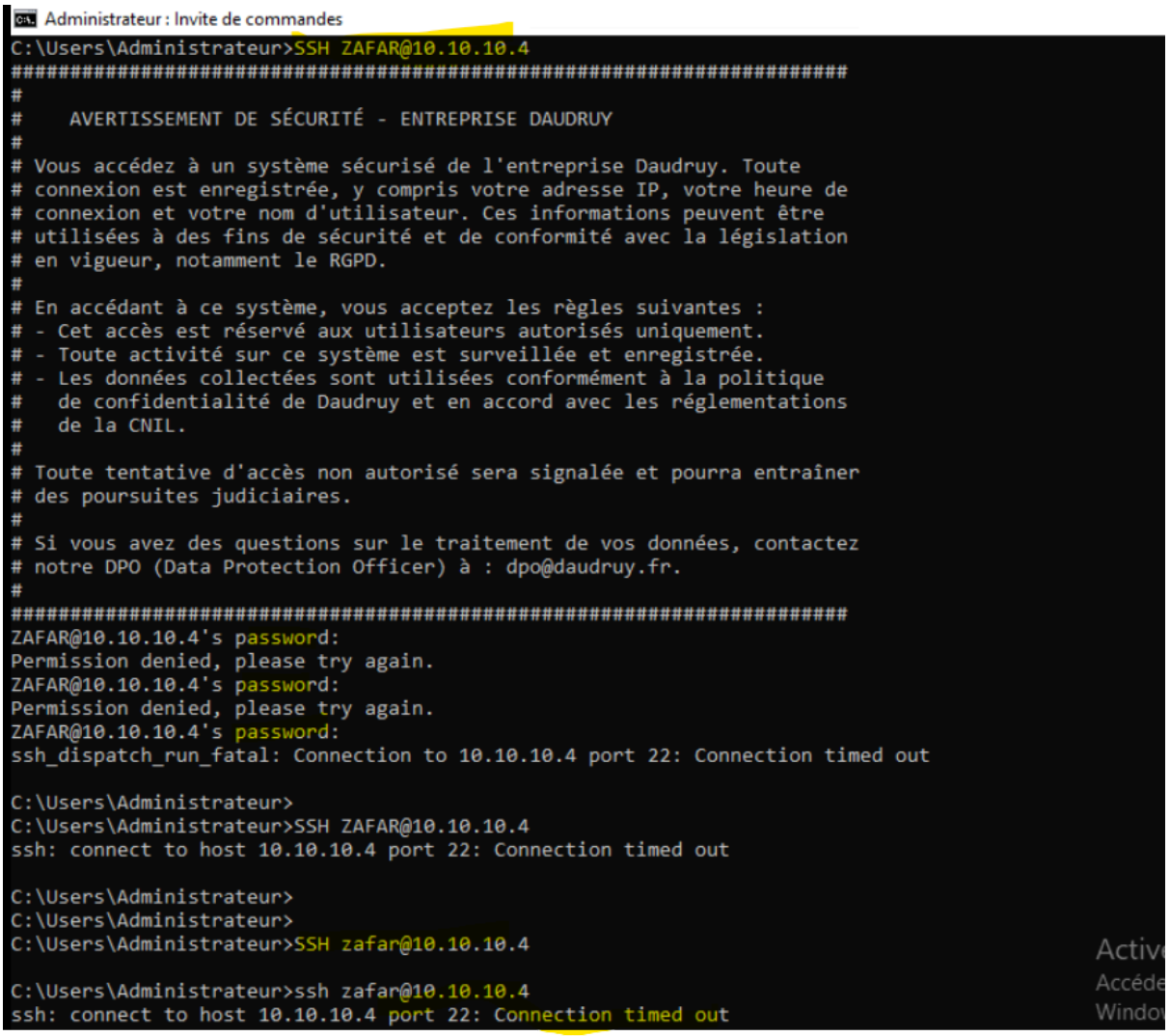
```
zafar@apache-guaca:~# cd /etc/fail2ban/
zafar@apache-guaca:/etc/fail2ban# ls
action.d  fail2ban.conf  fail2ban.d  filter.d  jail.conf  jail.d  paths-arch.conf  paths-common.conf  paths-debian.conf  paths-opensuse.conf
zafar@apache-guaca:/etc/fail2ban# sudo cp jail.conf jail.local
zafar@apache-guaca:/etc/fail2ban# sudo nano jail.local
```

```
  GNU nano 6.2                                                                jail.local
#
# SSH servers
#

[sshd]
# To use more aggressive sshd modes set filter parameter "mode" in jail.local:
# normal (default), ddos, extra or aggressive (combines all).
# See "tests/files/logs/sshd" or "filter.d/sshd.conf" for usage example and details.
#mode   = normal
enabled = true
port = ssh
filter = sshd
logpath = /var/log/auth.log
maxretry = 3
bantime = 3600
findtime = 600
```

```
zafar@apache-guaca:/etc/fail2ban# sudo systemctl enable fail2ban.service
Synchronizing state of fail2ban.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable fail2ban
zafar@apache-guaca:/etc/fail2ban# sudo systemctl restart fail2ban
zafar@apache-guaca:/etc/fail2ban# sudo systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
     Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2025-01-23 10:49:35 UTC; 2s ago
       Docs: man:fail2ban(1)
   Main PID: 48431 (fail2ban-server)
      Tasks: 5 (limit: 9394)
```

Test

05/01/2025



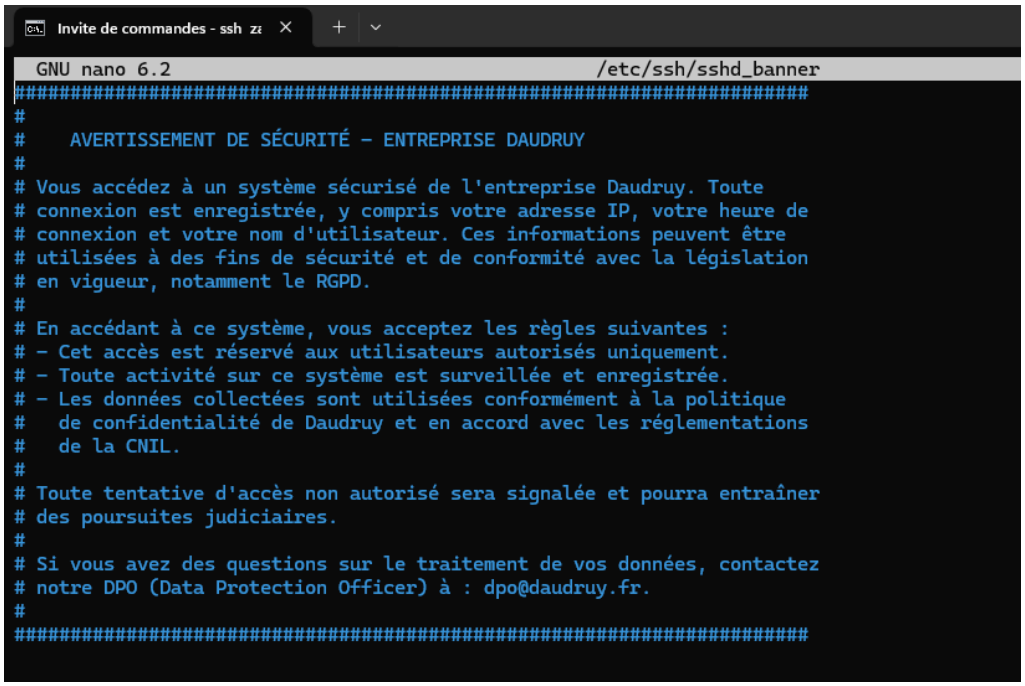

```
zafar@apache-guaca:/etc/fail2ban# sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 0
|  |- Total failed:     3
|  `- File list:        /var/log/auth.log
`- Actions
   |- Currently banned: 1
   |- Total banned:     2
   `- Banned IP list:   10.10.10.5
zafar@apache-guaca:/etc/fail2ban#
```

```
zafar@apache-guaca:/etc/fail2ban# sudo tail -f /var/log/fail2ban.log
2025-01-23 11:21:39,076 fail2ban.filter         [48752]: INFO    Added logfile: '/var/log/auth.log' (pos = 94190, ha
2025-01-23 11:21:39,079 fail2ban.jail           [48752]: INFO    Jail 'sshd' started
2025-01-23 11:21:39,277 fail2ban.actions        [48752]: NOTICE  [sshd] Restore Ban 10.10.10.5
2025-01-23 11:27:08,249 fail2ban.actions        [48752]: NOTICE  [sshd] Unban 10.10.10.5
2025-01-23 11:31:31,230 fail2ban.filter         [48752]: INFO    [sshd] Found 10.10.10.5 - 2025-01-23 11:31:31
2025-01-23 11:31:42,318 fail2ban.filter         [48752]: INFO    [sshd] Found 10.10.10.5 - 2025-01-23 11:31:41
2025-01-23 11:31:47,885 fail2ban.filter         [48752]: INFO    [sshd] Found 10.10.10.5 - 2025-01-23 11:31:47
2025-01-23 11:31:48,581 fail2ban.actions        [48752]: NOTICE  [sshd] Ban 10.10.10.5
```

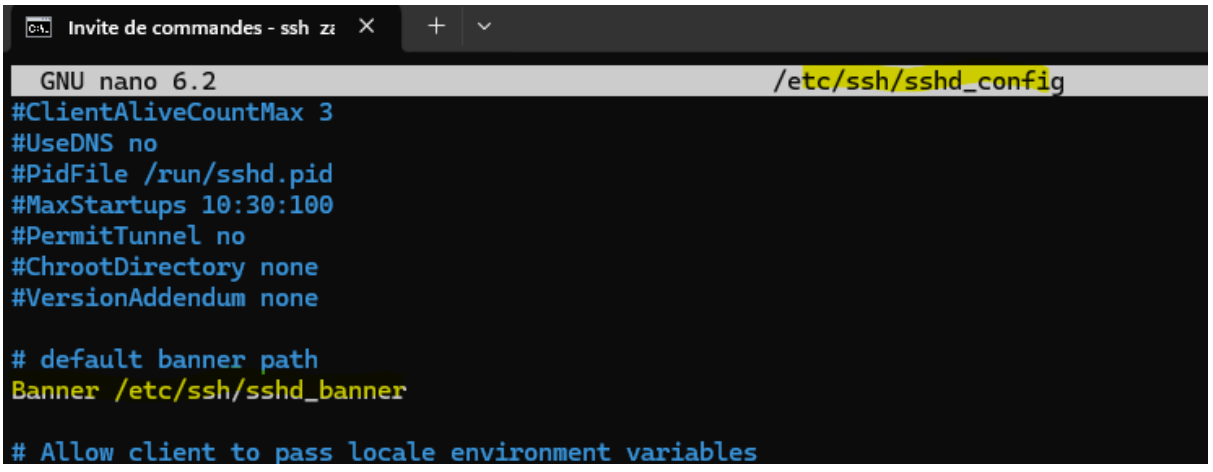
Bannière de connexion SSH avec conformité RGPD

05/01/2025

```
zafar@apache-guaca:/etc/fail2ban# sudo nano /etc/ssh/sshd_banner
```




```
zafar@apache-guaca:/etc/fail2ban# sudo nano /etc/ssh/sshd_config
```



05/01/2025

B. Créer un enregistrement vidéo des sessions

Téléchargement de l'extension

```
root@apache-guaca:/tmp# wget https://downloads.apache.org/guacamole/1.5.5/binary/guacamole-history-recording-storage-1.5.5.tar.gz
--2025-01-15 09:08:00--  https://downloads.apache.org/guacamole/1.5.5/binary/guacamole-history-recording-storage-1.5.5.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 88.99.208.237, 135.181.214.104, 2a01:4f9:3a:2c57::2, ...
Connecting to downloads.apache.org (downloads.apache.org)|88.99.208.237|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 15894 (16K) [application/x-gzip]
Saving to: 'guacamole-history-recording-storage-1.5.5.tar.gz'

guacamole-history-recording-storage-1.5.5.tar.gz     100%[=============================================================================================================>]  15,52K  --.-KB/s    in 0,02s

2025-01-15 09:08:00 (661 KB/s) - 'guacamole-history-recording-storage-1.5.5.tar.gz' saved [15894/15894]

root@apache-guaca:/tmp#
```

Puis, on décompresse l'archive tar.gz , On déplace le fichier .jar de l'extension vers le répertoire "extensions" de Guacamole :

```
root@apache-guaca:/tmp# sudo mv guacamole-history-recording-storage-1.5.5/guacamole-history-recording-storage-1.5.5.jar /etc/guacamole/extensions/
```

On redémarre le service puis crée un répertoire pour l' enregistrement

```
root@apache-guaca:/etc/guacamole/extensions# ls
guacamole-auth-jdbc-mysql-1.5.5.jar  guacamole-history-recording-storage-1.5.5.jar
```

```
root@apache-guaca:/etc/guacamole/extensions# ls -ld /var/lib/guacamole/recordings
drwxrws--- 2 tomcat tomcat 4096 janv. 15 09:10 /var/lib/guacamole/recordings
root@apache-guaca:/etc/guacamole/extensions# sudo chown -R root:tomcat /var/lib/guacamole/recordings
root@apache-guaca:/etc/guacamole/extensions# sudo chmod -R 770 /var/lib/guacamole/recordings
root@apache-guaca:/etc/guacamole/extensions# sudo usermod -aG tomcat $(ps -o user= -p $(pgrep guacd))
root@apache-guaca:/etc/guacamole/extensions# sudo systemctl restart guacd
```

```
root@apache-guaca: /etc/gua
  GNU nano 6.2                                  guacamole.properties
# MySQL
mysql-hostname=127.0.0.1
mysql-port=3306
mysql-database=guacadb
mysql-username=db-user
mysql-password=zafar

#activer les logs en mode débogage
guacd-hostname: localhost
guacd-port: 4822
log-level: debug



recording-path: /var/lib/guacamole/recordings
recording-name: ${GUAC_DATE}-${GUAC_TIME}-${GUAC_USERNAME}
create-recording-path: true
```

Il faut juste faire attention au droit de différent fichier et utilisateur

guacadmin	15-01-2025 12:45:22	24 secondes	ssh-apache	192.168.40.65	
guacadmin	15-01-2025 12:32:51	28 secondes	win-rdp-test	192.168.40.65	View
guacadmin	15-01-2025 12:32:29	3 secondes	ssh-apache	192.168.40.65	

Pour ssh : il vas utiliser la meme dossier que rdp et la meme chemins

Créer automatiquement le chemin typescript : ☐

Enregistrement Ecran

Chemin de l'enregistrement:	${HISTORY_PATH}/${HIST
Nom de l'enregistrement:	${GUAC_DATE}-${GUAC_
Exclure les graphiques/flux:	☐
Exclure la souris:	☐
Inclure les événements clavier:	☐
Créer automatiquement le chemin d'enregistrement:	☑

guacadmin	15-01-2025 14:22:25	22 secondes	ssh-apache	192.168.40.65	View
guacadmin	15-01-2025 14:21:14	15 secondes	ssh-apache	192.168.40.65	View

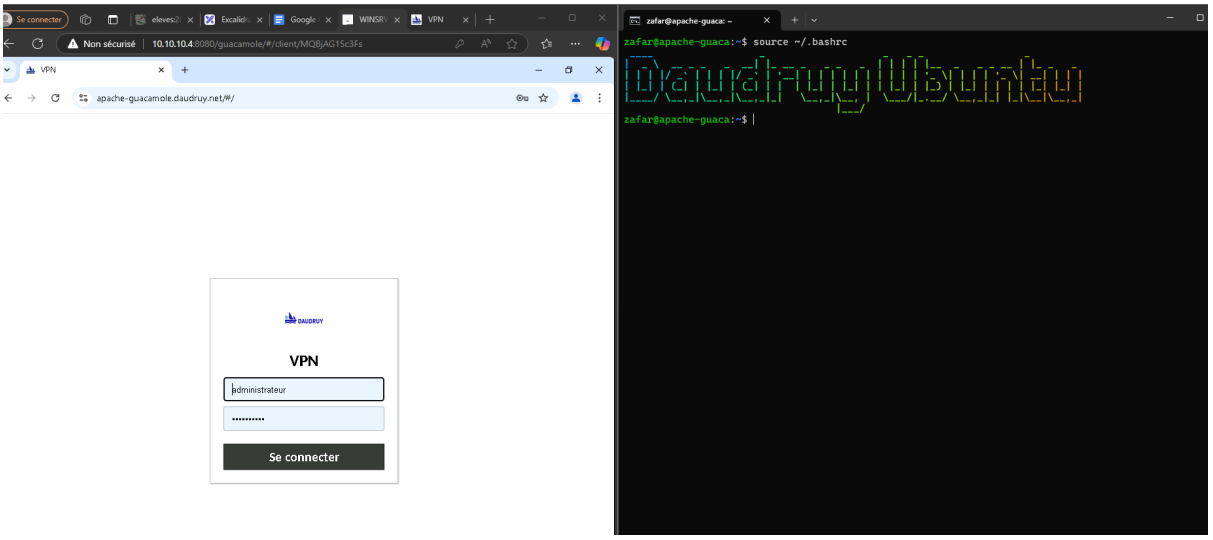
Les enregistrements des video de session utilisateur ont été sauvegardées le PC

Conclusion

Durant ces trois semaines de stage, j'ai pu rechercher et configurer **Guacamole**, en passant par :

✅ **Installation et base de données**
✅ **Mise en place de l'authentification TOTP**
✅ **Configuration DNS et certificat SSL auto-généré**
✅ **Déploiement du certificat sur PC**
✅ **Activation de HTTPS et redirection HTTP → HTTPS**
✅ **Enregistrement des sessions**
✅ **Sécurisation avec Fail2Ban**

Tout au long de cette configuration, j'ai rencontré plusieurs problématiques, mais j'ai pu les résoudre en m'appuyant sur la documentation et des ressources adaptées. Ce stage m'a permis de renforcer mes compétences en administration système linux et en sécurisation des services.



Créer un enregistrement vidéo des sessions

Enregistrement sur Serveur en local :

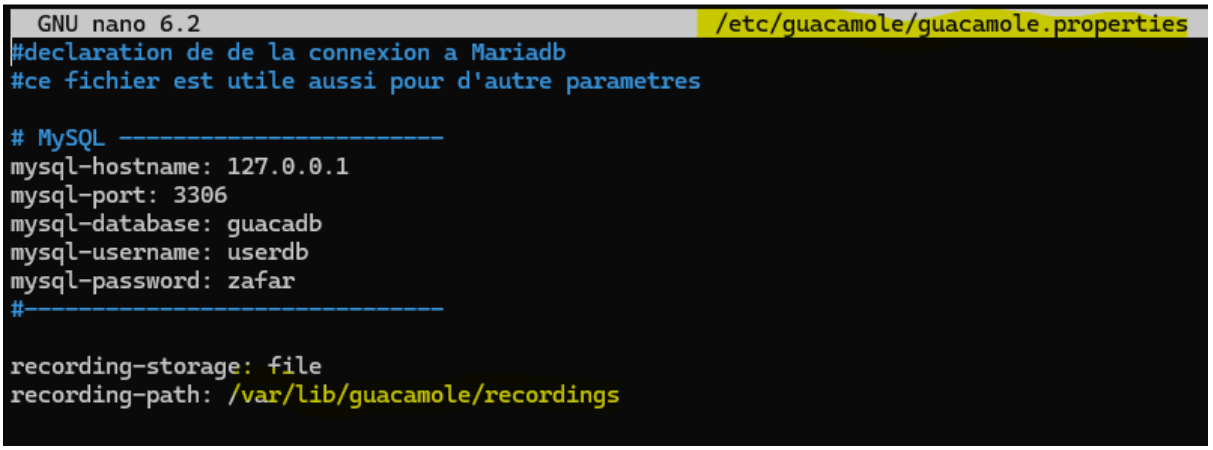
1. Installation et configuration de Guacamole

On commence par télécharger l'archive tar.gz d'Apache Guacamole , avec la bonne version 1.5.5.
On a téléchargé, extrait, déplacé l'extension d'enregistrement de sessions Guacamole dans le répertoire des extensions et redémarré le service Tomcat pour qu'elle soit prise en compte.
L'extension est intégrée à Apache Guacamole. 👍

2. Configuration de l'enregistrement des sessions dans Guacamole

Objectif : Configurer Guacamole pour enregistrer les sessions d'accès.

sudo nano /etc/guacamole/guacamole.properties



Créer le répertoire des enregistrements et donner droit à tomcat :

```
root@apache-guaca:/tmp# sudo nano /etc/guacamole/guacamole.properties
root@apache-guaca:/tmp# sudo mkdir -p /var/lib/guacamole/recordings
root@apache-guaca:/tmp# sudo chown -R tomcat:tomcat /var/lib/guacamole/recordings
root@apache-guaca:/tmp# sudo systemctl restart tomcat9
root@apache-guaca:/tmp#
```

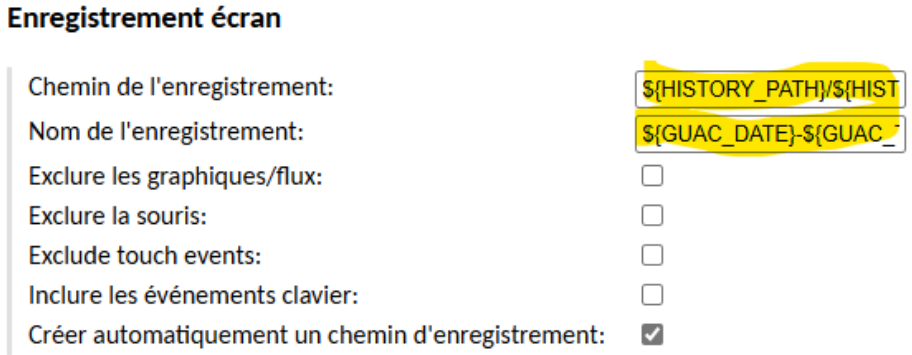
Création de fichier et l' endroit guacamole vas déposer les video :

```
  GNU nano 6.2                                  /etc/guacamole/guacamole.properties
#declaration de de la connexion a Mariadb
#ce fichier est utile aussi pour d'autre parametres

# MySQL -------------------------
mysql-hostname: 127.0.0.1
mysql-port: 3306
mysql-database: guacadb
mysql-username: userdb
mysql-password: zafar
#------------------------------

recording-storage: file
recording-path: /var/lib/guacamole/recordings
```

Ensuite, il faut configurer l'espace de stockage.

```
root@apache-guaca:~# sudo mkdir -p /var/lib/guacamole/recordings
root@apache-guaca:~# sudo chown -R tomcat:tomcat /var/lib/guacamole/recordings
root@apache-guaca:~# sudo systemctl restart tomcat9
root@apache-guaca:~#
```

Sur guacamole on met le variable
Chemin de l'enregistrement : `${HISTORY_PATH}/${HISTORY_UUID}` (Cela définit où les enregistrements sont stockés sur le serveur).
Nom de l'enregistrement : `${GUAC_DATE}-${GUAC_TIME} - RDP - ${GUAC_USERNAME}`.

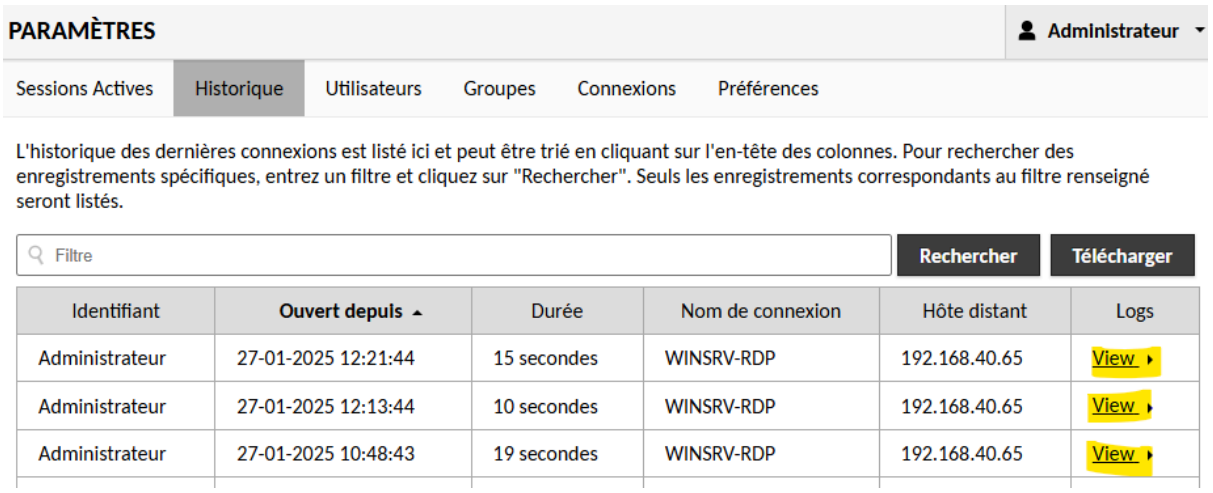


Maintenant on peut ouvrire une session et faire le test pour voir si le vidéo est enregistré dans le fichier :/var/lib/guacamole/recordings

Test a été fait et il enregistre les vidéo dans le répertoire sur serveur 👏

```
root@apache-guaca:/tmp# ls /var/lib/guacamole/recordings
72ae31ab-63fa-3145-a954-88210bbd3651  938794f2-cd5f-368b-8148-bc11a9cea29c  fdf244e0-cdd9-3fa7-ab2d-03773b22ba5c
root@apache-guaca:/tmp#
```

Par contre, on peut ouvrir les vidéos à l'aide de l'interface Guacamole, mais on ne peut les visualiser avec les formats MP4 ou M4V. Guacamole enregistre les vidéos sous un format brut .flv, donc il faut les convertir si on souhaite les voir sur Windows .



Jusqu'ici, j'ai pu suivre les étapes grâce à un tutoriel, mais pour aller plus loin et convertir le fichier vidéo puis l'envoyer vers le NAS, j'ai passé beaucoup de temps à rechercher le bon format, les outils à installer, etc. J'ai consulté plusieurs tutoriels, posé des questions à ChatGPT, et étudié la documentation d'Apache, mais certains aspects sont restés flous.

Conversion les vidéo et l' envoyer sur NAS

Methode 1

Le guacd et tomcat doit avoir le droit d'écrire :

```
root@apache-guaca:/var/lib/guacamole/recordings# sudo chown -R guacd:tomcat /var/lib/guacamole/recordings
root@apache-guaca:/var/lib/guacamole/recordings# sudo chmod -R 770 /var/lib/guacamole/recordings
root@apache-guaca:/var/lib/guacamole/recordings# sudo systemctl restart tomcat9
root@apache-guaca:/var/lib/guacamole/recordings# sudo systemctl restart guacd
root@apache-guaca:/var/lib/guacamole/recordings# ls
e5adf067-6a3a-3c92-ac8c-dd954360d6dd
root@apache-guaca:/var/lib/guacamole/recordings#
```

Schéma pour déposer le vidéo

```
 GNU nano 6.2                                  /etc/guacamole/guacamole.properties
#declaration de de la connexion a Mariadb
#ce fichier est utile aussi pour d'autre parametres

# MySQL ------------------------
mysql-hostname: 127.0.0.1
mysql-port: 3306
mysql-database: guacadb
mysql-username: userdb
mysql-password: zafar
#-------------------------------

history-recording-enabled: true
history-recording-storage-dir: /var/lib/guacamole/recordings
```

Conversion en m4v

On installe l'utile de conversion :

```
root@apache-guaca:/var/lib/guacamole/recordings# sudo apt install ffmpeg
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
```

On ouvre une session en rdp pour le test de conversion puis on se dirige vers le dossier d'enregistrement :

```
root@apache-guaca:/var/lib/guacamole/recordings# ls
d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9
root@apache-guaca:/var/lib/guacamole/recordings# cd d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9/
root@apache-guaca:/var/lib/guacamole/recordings/d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9# ls
'20250127-145853 - RDP - Administrateur'
root@apache-guaca:/var/lib/guacamole/recordings/d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9#
```

Maintenant on convertir le fichier vidéo en .m4v à l'aide ffmpeg

```
root@apache-guaca:/var/lib/guacamole/recordings/d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9# sudo ffmpeg -f rawvideo -pix_fmt yuv420p -s 1280x720 -i "/var/lib/guacamole/recordings/d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9/20250127-145853 - RDP - Administrateur" "/var/lib/guacamole/recordings/d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9/20250127-145853 - RDP - Administrateur.m4v"
```

Il a bien converti le fichier

```
root@apache-guaca:/var/lib/guacamole/recordings/d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9# sudo ffmpeg -f rawvideo -pix_fmt yuv420p -s 1280x720 -i "/var/lib/guacamole/recordings/d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9/20250127-145853 - RDP - Administrateur" "/var/lib/guacamole/recordings/d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9/20250127-145853 - RDP - Administrateur.m4v"
ffmpeg version 4.4.2-0ubuntu0.22.04.1 Copyright (c) 2000-2021 the FFmpeg developers
  built with gcc 11 (Ubuntu 11.2.0-19ubuntu1)
  configuration: --prefix=/usr --extra-version=0ubuntu0.22.04.1 --toolchain=hardened --libdir=/usr/lib/x86_64-linux-gnu --incdir=/usr/include/x86_64-linux-gnu --arch=amd64 --enable-gpl --disable-stripping --enable-gnutls --enable-ladspa --enable-libaom --enable-libass --enable-libbluray --enable-libbs2b --enable-libcaca --enable-libcdio --enable-libcodec2 --enable-libdav1d --enable-libflite --enable-libfontconfig --enable-libfreetype --enable-libfribidi --enable-libgme --enable-libgsm --enable-libjack --enable-libmp3lame --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --enable-libpulse --enable-librabbitmq --enable-librubberband --enable-libshine --enable-libsnappy --enable-libsoxr --enable-libspeex --enable-libsrt --enable-libssh --enable-libtheora --enable-libtwolame --enable-libvidstab --enable-libvorbis --enable-libvpx --enable-libwebp --enable-libx265 --enable-libxml2 --enable-libxvid --enable-libzimg --enable-libzmq --enable-libzvbi --enable-lv2 --enable-omx --enable-openal --enable-opencl --enable-opengl --enable-sdl2 --enable-pocketsphinx --enable-librsvg --enable-libmfx --enable-libdc1394 --enable-libdrm --enable-libiec61883 --enable-chromaprint --enable-frei0r --enable-libx264 --enable-shared
  libavutil      56. 70.100 / 56. 70.100
  libavcodec     58.134.100 / 58.134.100
  libavformat    58. 76.100 / 58. 76.100
  libavdevice    58. 13.100 / 58. 13.100
  libavfilter     7.110.100 /  7.110.100
  libswscale      5.  9.100 /  5.  9.100
  libswresample   3.  9.100 /  3.  9.100
  libpostproc    55.  9.100 / 55.  9.100
[rawvideo @ 0x562c85a8f6c0] Packet corrupt (stream = 0, dts = 0).
[rawvideo @ 0x562c85a8f6c0] Estimating duration from bitrate, this may be inaccurate
Input #0, rawvideo, from '/var/lib/guacamole/recordings/d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9/20250127-145853 - RDP - Administrateur':
  Duration: 00:00:00.04, start: 0.000000, bitrate: 236346 kb/s
  Stream #0:0: Video: rawvideo (I420 / 0x30323449), yuv420p, 1280x720, 276480 kb/s, 25 tbr, 25 tbn, 25 tbc
Stream mapping:
  Stream #0:0 -> #0:0 (rawvideo (native) -> h264 (libx264))
Press [q] to stop, [?] for help
/var/lib/guacamole/recordings/d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9/20250127-145853 - RDP - Administrateur: corrupt input packet in stream 0
[rawvideo @ 0x562c85a9bcc0] Invalid buffer size, packet size 1181734 < expected frame_size 1382400
Error while decoding stream #0:0: Invalid argument
[libx264 @ 0x562c85a9d500] using cpu capabilities: MMX2 SSE2Fast SSSE3 SSE4.2 AVX FMA3 BMI2 AVX2 AVX512
[libx264 @ 0x562c85a9d500] profile High, level 3.1, 4:2:0, 8-bit
[libx264 @ 0x562c85a9d500] 264 - core 163 r3060 5db6aa6 - H.264/MPEG-4 AVC codec - Copyleft 2003-2021 - http://www.videolan.org/x264.html - options: cabac=1 ref=3 deblock=1:0:0 analyse=0x3:0x113 me=hex subme=7 psy=1 psy_rd=1.00:0.00 mixed_ref=1 me_range=16 chroma_me=1 trellis=1 8x8dct=1 cqm=0 deadzone=21,11 fast_pskip=1 chroma_qp_offset=-2 threads=3 lookahead_threads=1 sliced_threads=0 nr=0 decimate=1 interlaced=0 bluray_compat=0 constrained_intra=0 bframes=3 b_pyramid=2 b_adapt=1 b_bias=0 direct=1 weightb=1 open_gop=0 weightp=2 keyint=250 keyint_min=25 scenecut=40 intra_refresh=0 rc_lookahead=40 rc=crf mbtree=1 crf=23.0 qcomp=0.60 qpmin=0 qpmax=69 qpstep=4 ip_ratio=1.40 aq=1:1.00
Output #0, ipod, to '/var/lib/guacamole/recordings/d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9/20250127-145853 - RDP - Administrateur.m4v':
  Metadata:
    encoder         : Lavf58.76.100
  Stream #0:0: Video: h264 (avc1 / 0x31637661), yuv420p, 1280x720, q=2-31, 25 fps, 12800 tbn
    Metadata:
      encoder         : Lavc58.134.100 libx264
    Side data:
      cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv_delay: N/A
frame=    0 fps=0.0 q=0.0 Lsize=       0kB time=00:00:00.00 bitrate=N/A speed=   0x
video:0kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: unknown
Conversion failed!
root@apache-guaca:/var/lib/guacamole/recordings/d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9# ls
'20250127-145853 - RDP - Administrateur'  '20250127-145853 - RDP - Administrateur.m4v'
root@apache-guaca:/var/lib/guacamole/recordings/d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9#
```

- **FFmpeg** : Utilisé pour la conversion vidéo.

bash — Copier

```
sudo apt update
sudo apt install ffmpeg
```

- **rsync** : Utilisé pour envoyer les fichiers sur le NAS.

bash — Copier

```
sudo apt install rsync
```

Montage sur le NAS :

Installer les outils nécessaires pour le montage du NAS, Pour SMB/CIFS :

```
root@apache-guaca:/var/lib/guacamole/recordings# sudo apt-get install cifs-utils
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  keyutils libtalloc2 libtevent0 libwbclient0
```

Créer un point de montage

```
root@apache-guaca:/var/lib/guacamole/recordings# sudo mkdir -p /mnt/nas
root@apache-guaca:/var/lib/guacamole/recordings#
```

Monter le NAS Pour SMB/CIFS :

On cree un dossier puis on exécute la commande de montage

```
root@apache-guaca:/var/lib/guacamole/recordings# sudo mkdir -p /mnt/nas
root@apache-guaca:/var/lib/guacamole/recordings# sudo mount -t cifs //172.16.[redacted]/Guacamole /mnt/nas -o username=apache-guacamole,password=[redacted]loGF65I
root@apache-guaca:/var/lib/guacamole/recordings#
```

Verification : Ok 👍

```
root@apache-guaca:/var/lib/guacamole/recordings# ls /mnt/nas
@Recycle
root@apache-guaca:/var/lib/guacamole/recordings#
```

envoyer le fichier vers le NAS avec `rsync` :

```
root@apache-guaca:/var/lib/guacamole/recordings# ls
d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9
root@apache-guaca:/var/lib/guacamole/recordings# cd d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9/
root@apache-guaca:/var/lib/guacamole/recordings/d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9# ls
'20250127-145853 - RDP - Administrateur'  '20250127-145853 - RDP - Administrateur.m4v'
root@apache-guaca:/var/lib/guacamole/recordings/d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9# sudo rsync -av "
/var/lib/guacamole/recordings/d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9/20250127-145853 - RDP - Administrat
eur.m4v" /mnt/nas/
sending incremental file list
20250127-145853 - RDP - Administrateur.m4v

sent 404 bytes  received 35 bytes  878,00 bytes/sec
total size is 262  speedup is 0,60
root@apache-guaca:/var/lib/guacamole/recordings/d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9# ls /mnt/nas/
'20250127-145853 - RDP - Administrateur.m4v'   @Recycle
root@apache-guaca:/var/lib/guacamole/recordings/d655a5fc-b13d-3d9f-b46a-0b874b0a3ef9#
```

Méthode 2 conseiller par apache

L'utilisation de **guacenc** pour convertir les enregistrements Guacamole en fichiers vidéo **.m4v** permet de rendre ces enregistrements lisibles et exploitables, offrant ainsi une meilleure accessibilité et gestion des sessions.

```
root@apache-guaca:/var/lib/guacamole/recordings/d9666d3f-5d6a-335d-bf49-e052d6130967# sudo guacenc -s 1280x720 -f "/var
lib/guacamole/recordings/d9666d3f-5d6a-335d-bf49-e052d6130967/20250128-125117 - RDP - Administrateur"
guacenc: INFO: Guacamole video encoder (guacenc) version 1.5.5
guacenc: INFO: 1 input file(s) provided.
guacenc: INFO: Video will be encoded at 1280x720 and 2000000 bps.
guacenc: INFO: Encoding "/var/lib/guacamole/recordings/d9666d3f-5d6a-335d-bf49-e052d6130967/20250128-125117 - RDP - Adm
nistrateur" to "/var/lib/guacamole/recordings/d9666d3f-5d6a-335d-bf49-e052d6130967/20250128-125117 - RDP - Administrate
r.m4v" ...
guacenc: INFO: All files encoded successfully.
root@apache-guaca:/var/lib/guacamole/recordings/d9666d3f-5d6a-335d-bf49-e052d6130967# ls /var/lib/guacamole/recordings/
9666d3f-5d6a-335d-bf49-e052d6130967/
'20250128-125117 - RDP - Administrateur'  '20250128-125117 - RDP - Administrateur.m4v'
```

Automatiser la conversion et l'envoi vers le NAS avec BASH

Script 1

Le format d'enregistrement de base n'est pas directement lisible. Apache Guacamole propose l'outil **guacenc** pour convertir ces enregistrements en vidéos au format **M4V**.

Objectif :

1. transférer les enregistrements vidéo depuis **var/lib/guacamole/recordings** vers le NAS avec le dossier montage **/mnt/nas/guacamole_recordings**
2. les convertir en format vidéo m4v
3. supprimer les fichiers locaux

Resultat

```
root@apache-guaca:/opt/scripts# ls
envoie_et_nettoie.sh
root@apache-guaca:/opt/scripts#
```

```
root@apache-guaca:/var/lib/guacamole/recordings# sudo /opt/scripts/envoie_et_nettoie.sh
Étape 3 : Vérification du répertoire des enregistrements...
Répertoire trouvé : /var/lib/guacamole/recordings/9b348969-92d2-3f63-9fbe-86ebffe634c2
Fichier trouvé : /var/lib/guacamole/recordings/9b348969-92d2-3f63-9fbe-86ebffe634c2/20250128-114900 - RDP - Administrateur
Étape 1.1 : Conversion du fichier 20250128-114900 - RDP - Administrateur en .m4v
guacenc: INFO: Guacamole video encoder (guacenc) version 1.5.5
guacenc: INFO: 1 input file(s) provided.
guacenc: INFO: Video will be encoded at 1280x720 and 2000000 bps.
guacenc: INFO: Encoding "/var/lib/guacamole/recordings/9b348969-92d2-3f63-9fbe-86ebffe634c2/20250128-114900 - RDP - Administrateur" to "/var/lib/guacamole/recordings/9b348969-92d2-3f63-9fbe-86ebffe634c2/20250128-114900 - RDP - Administrateur.m4v" ...
guacenc: INFO: All files encoded successfully.
Conversion réussie avec guacenc : /var/lib/guacamole/recordings/9b348969-92d2-3f63-9fbe-86ebffe634c2/20250128-114900 - RDP - Administrateur -> /var/lib/guacamole/recordings/9b348969-92d2-3f63-9fbe-86ebffe634c2/20250128-114900 - RDP - Administrateur.m4v
Fichier 20250128-114900 - RDP - Administrateur converti en .m4v avec succès ==========.
========== Étape 1.2 : Transfert du fichier 20250128-114900 - RDP - Administrateur sur le NAS==========
sending incremental file list
20250128-114900 - RDP - Administrateur.m4v

sent 1.563.247 bytes  received 35 bytes  3.126.564,00 bytes/sec
total size is 1.562.734  speedup is 1,00
Fichier 20250128-114900 - RDP - Administrateur transféré avec succès vers le NAS.==========
Étape 1.3 : Suppression du fichier local /var/lib/guacamole/recordings/9b348969-92d2-3f63-9fbe-86ebffe634c2/20250128-114900 - RDP - Administrateur et fichier converti
Fichier local 20250128-114900 - RDP - Administrateur supprimé avec succès.
Étape 1.4 : Vérification et suppression du répertoire /var/lib/guacamole/recordings/9b348969-92d2-3f63-9fbe-86ebffe634c2 si vide
Répertoire /var/lib/guacamole/recordings/9b348969-92d2-3f63-9fbe-86ebffe634c2 supprimé car il est vide.
Le script Guacamole a été exécuté avec succès.
root@apache-guaca:/var/lib/guacamole/recordings#
```

```
root@apache-guaca:/var/lib/guacamole/recordings# ls
root@apache-guaca:/var/lib/guacamole/recordings#
```

NAS

```
root@apache-guaca:/var/lib/guacamole/recordings# ls /mnt/nas/guacamole_recordings/
'20250128-114129 - RDP - Administrateur.m4v'  '20250128-114900 - RDP - Administrateur.m4v'
'20250128-114727 - RDP - Administrateur.m4v'
root@apache-guaca:/var/lib/guacamole/recordings#
```

Script1 complete

```
#!/bin/bash

# Dossier des enregistrements Guacamole
recordings_dir="/var/lib/guacamole/recordings"
# Dossier du NAS monté
nas_dir="/mnt/nas/guacamole_recordings"

# Étape 1 : Transfert et conversion des fichiers
transfer_and_convert() {
    local dir="$1"
    local file="$2"
    # Extraire le nom du fichier
    filename=$(basename "$file")

    # Étape 1.1 : Conversion du fichier avec guacenc en .m4v
    echo "Étape 1.1 : Conversion du fichier $filename en .m4v"
    convert_with_guacenc "$file"

    # Vérifier si la conversion a réussi
    if [ $? -eq 0 ]; then
        echo "Fichier $filename converti en .m4v avec succès."

        # Étape 1.2 : Transfert du fichier converti sur le NAS
        echo "Étape 1.2 : Transfert du fichier $filename sur le NAS"
        rsync -av "$file.m4v" "$nas_dir/"

        # Vérifier si le fichier a été transféré correctement
        if [ $? -eq 0 ]; then
            echo "Fichier $filename transféré avec succès vers le NAS."

            # Étape 1.3 : Suppression du fichier local après conversion et transfert
            echo "Étape 1.3 : Suppression du fichier local $file et fichier converti"
            rm -f "$file" "$file.m4v"
            if [ $? -eq 0 ]; then
                echo "Fichier local $filename supprimé avec succès."
            else
                echo "Erreur lors de la suppression du fichier local $filename."
            fi

            # Étape 1.4 : Vérification et suppression du répertoire si vide
            echo "Étape 1.4 : Vérification et suppression du répertoire $dir si vide"
            if [ ! "$(ls -A "$dir")" ]; then
                rmdir "$dir"
                echo "Répertoire $dir supprimé car il est vide."
            else
                echo "Répertoire $dir non vide, il n'a pas été supprimé."
            fi
```

```
        else
            echo "Erreur lors du transfert du fichier $filename vers le NAS."
        fi
    else
        echo "Erreur lors de la conversion du fichier $filename."
    fi
}

# Étape 2 : Conversion avec guacenc pour créer un fichier .m4v
convert_with_guacenc() {
    local input_file="$1"
    # Appeler guacenc pour convertir en .m4v (résolution 1280x720)
    sudo guacenc -s 1280x720 -f "$input_file"

    # Vérifier si la conversion s'est bien déroulée
    if [ $? -eq 0 ]; then
        echo "Conversion réussie avec guacenc : $input_file -> $input_file.m4v"
        return 0
    else
        echo "Erreur lors de la conversion avec guacenc pour le fichier $input_file"
        return 1
    fi
}

# Étape 3 : Vérification du répertoire des enregistrements
echo "Étape 3 : Vérification du répertoire des enregistrements..."
if [ -d "$recordings_dir" ]; then
    # Parcourir tous les sous-répertoires dans /recordings
    for dir in "$recordings_dir"/*; do
        if [ -d "$dir" ]; then
            echo "Répertoire trouvé : $dir"

            # Parcourir les fichiers à l'intérieur de chaque sous-répertoire
            for file in "$dir"/*; do
                if [ -f "$file" ]; then
                    echo "Fichier trouvé : $file"
                    # Appeler la fonction pour transférer ce fichier vers le NAS et le convertir
                    transfer_and_convert "$dir" "$file"
                fi
            done
        fi
    done
else
    echo "Le répertoire des enregistrements Guacamole n'existe pas."
fi

# Fin du script, sans notification par email
echo "Le script Guacamole a été exécuté avec succès."
```

Automatiser script de envoie_et_nettoie

```
zafar@apache-guaca:~# crontab -e
no crontab for zafar - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.basic
  3. /usr/bin/vim.tiny
  4. /bin/ed

Choose 1-4 [1]: 1
```

```
# Crontab pour exécuter le script à 12h et 17h chaque jour
0 12,17 * * * /opt/scripts/envoie_et_nettoie.sh
```

Droit des utilisateur sur les fichier

```
root@apache-guaca:~# ls -l /var/lib/guacamole/recordings/
total 4
drwxr-s--- 2 guacd tomcat 4096 janv. 28 12:54 d9666d3f-5d6a-335d-bf49-e052d6130967
root@apache-guaca:~#
```

```
root@apache-guaca:~# ls -l /mnt/nas/guacamole_recordings/
total 9168
-rwxr-xr-x 1 root root 3627082 janv. 28 11:41 '20250128-114129 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root 2380950 janv. 28 11:47 '20250128-114727 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root 1562734 janv. 28 11:49 '20250128-114900 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root 1792829 janv. 28 12:43 '20250128-124234 - RDP - Administrateur.m4v'
root@apache-guaca:~#
```

```
root@apache-guaca:~# ls -l /opt/scripts/
total 4
-rwxr-xr-x 1 zafar root 3618 janv. 28 11:48 envoie_et_nettoie.sh
root@apache-guaca:~#
```

Commande diagnostique :

sudo tail -f /var/log/tomcat9/catalina.out
`ls -ld /var/lib/guacamole/recordings`

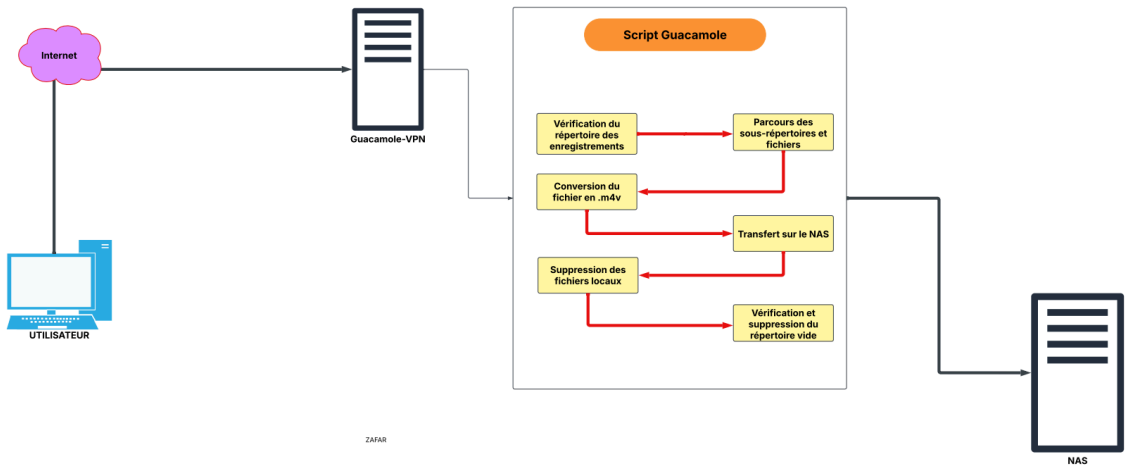
Nas ==========
apache-guacamole

2ZwfzZQloGF65l

172.16.171.13

Guacamole

J'ai passé quatre jours à développer ce script pour automatiser la gestion des enregistrements Guacamole. Chaque étape m'a demandé plusieurs tentatives : d'abord pour vérifier les répertoires et convertir les fichiers avec `guacenc`, puis pour transférer les vidéos sur le NAS en utilisant `rsync`, et enfin, pour supprimer les fichiers locaux uniquement après un transfert réussi. J'ai appris à tester plusieurs options pour chaque commande, et à gérer les erreurs de manière robuste. Ce projet m'a aussi appris l'importance de valider chaque étape avant de passer à la suivante, et à optimiser les outils pour garantir des transferts fiables et efficaces.



Scripte de suppression des fichier sur NAS

Objective :

Ce script supprime tous les fichiers dans le répertoire spécifié sur le NAS, où le script 1 envoie les vidéos converties. Il est exécuté automatiquement via cron pour une suppression périodique des fichiers, tous les 10 jours à midi (12h).

```
root@apache-guaca:/opt/scripts# nano nas_supprime.sh
root@apache-guaca:/opt/scripts# chmod +x /opt/scripts/clean_nas.sh
chmod: cannot access '/opt/scripts/clean_nas.sh': No such file or directory
root@apache-guaca:/opt/scripts# chmod +x /opt/scripts/nas_supprime.sh
```

```
root@apache-guaca:/opt/scripts# ls /mnt/nas/guacamole_recordings/
'20250128-114129 - RDP - Administrateur.m4v'  '20250128-114900 - RDP - Administrateur.m4v'
'20250128-114727 - RDP - Administrateur.m4v'  '20250128-124234 - RDP - Administrateur.m4v'
root@apache-guaca:/opt/scripts# sudo /opt/scripts/nas_supprime.sh
Suppression de tous les fichiers dans le répertoire /mnt/nas/guacamole_recordings/
Tous les fichiers ont été supprimés avec succès.
root@apache-guaca:/opt/scripts# ls /mnt/nas/guacamole_recordings/
```

```
root@apache-guaca:/opt/scripts# crontab -e
no crontab for root - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.basic
  3. /usr/bin/vim.tiny
  4. /bin/ed

Choose 1-4 [1]:
```

```
# script s'exécute tous les 10 jours à 13h (1 PM) pour supprimer les videos de NAS
0 13 */10 * * /opt/scripts/nas_supprime.sh
```

Test :

Aujourd'hui, nous sommes le 30/01 et mon script, avec l'aide de cron, a bien converti les vidéos brutes en M4V et sauvegardé les vidéos sur le NAS le 29/01. Il y a un autre script qui supprime les vidéos du NAS au bout de 10 jours.

Ce travail m'a permis, en tant que stagiaire, d'acquérir de nouvelles compétences en automatisation et en gestion du stockage.

Hier à 14h, il a envoyé les premières vidéos et a tenté également à 17h (aucune vidéo envoyée car je n'avais pas ouvert de session).

J'ai mis deux horaires pour l'exécution : 14h et 17h.

```
zafar@apache-guaca:~# sudo ls -l /mnt/nas/guacamole_recordings/
total 12940
-rwxr-xr-x 1 root root 5684234 janv. 29 14:43 '20250128-125117 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root   10761 janv. 29 14:43 '20250129-143650 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root 5814962 janv. 29 14:43 '20250129-143654 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root 1719985 janv. 29 14:43 '20250129-143727 - RDP - Administrateur.m4v'
zafar@apache-guaca:~#
```

Ligne de commande cron exécutée :

- Par exemple :
  `Jan 29 17:00:01 apache-guaca CRON[83361]: (zafar) CMD (/opt/scripts/envoie_et_nettoie.sh)` Cela indique que le script `envoie_et_nettoie.sh` a été exécuté par l'utilisateur `zafar` à 17:00 le 29 janvier.

```
zafar@apache-guaca:~# grep CRON /var/log/syslog
Jan 27 15:17:01 apache-guaca CRON[56206]: (root) CMD (   cd / && run-parts --report /etc/cron.hourly)
Jan 28 10:17:01 apache-guaca CRON[73874]: (root) CMD (   cd / && run-parts --report /etc/cron.hourly)
Jan 28 11:17:01 apache-guaca CRON[79438]: (root) CMD (   cd / && run-parts --report /etc/cron.hourly)
Jan 28 12:00:01 apache-guaca CRON[81723]: (root) CMD (test -x /usr/bin/certbot -a \! -d /run/systemd/s
Jan 28 12:17:01 apache-guaca CRON[81729]: (root) CMD (   cd / && run-parts --report /etc/cron.hourly)
Jan 28 13:17:01 apache-guaca CRON[82330]: (root) CMD (   cd / && run-parts --report /etc/cron.hourly)
Jan 29 15:17:01 apache-guaca CRON[83361]: (root) CMD (   cd / && run-parts --report /etc/cron.hourly)
Jan 29 16:17:01 apache-guaca CRON[83382]: (root) CMD (   cd / && run-parts --report /etc/cron.hourly)
Jan 29 17:00:01 apache-guaca CRON[83396]: (zafar) CMD (/opt/scripts/envoie_et_nettoie.sh)
```

Script alert SSH

C’est script en **Python** qui envoie une notification par **e-mail** chaque fois qu'un utilisateur se connecte à **Guacamole** ou tente une connexion SSH. Il utilise **Postfix** pour intercepter les connexions et **SMTP** pour envoyer l'alerte

```
zafar@apache-guaca:~# sudo apt update && sudo apt install postfix mailutils -y
[sudo] password for zafar:
Sorry, try again.
[sudo] password for zafar:
Sorry, try again.
[sudo] password for zafar:
Atteint :1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Atteint :2 http://archive.ubuntu.com/ubuntu jammy InRelease
Atteint :3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Atteint :4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Lecture des listes de paquets... Fait
```

```
zafar@apache-guaca:~# sudo apt install python3 python3-pip -y
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
python3 est déjà la version la plus récente (3.10.6-1~22.04.1).
python3 passé en « installé manuellement ».
```

```
zafar@apache-guaca:/opt/scripts# sudo nano monitor_guacamole_ssh.py
```

```
zafar@apache-guaca:/opt/scripts# sudo chmod +x monitor_guacamole_ssh.py
```

```
 GNU nano 6.2                                                monitor_guacamole_ssh.py
import smtplib
import time
import re
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart

# Configuration SMTP
SMTP_SERVER = "smtp-mibc-fr-07.mailinblack.com"
SMTP_PORT = 25
MAIL_TO = "stagiaire-it@daudruy.fr"
MAIL_FROM = "stagiaire-it@daudruy.fr"  # À remplacer par un mail valide

# Fonction d'envoi d'alerte
def send_alert(subject, message):
    try:
        msg = MIMEMultipart()
        msg['From'] = MAIL_FROM
        msg['To'] = MAIL_TO
        msg['Subject'] = subject

        msg.attach(MIMEText(message, 'plain'))

        with smtplib.SMTP(SMTP_SERVER, SMTP_PORT) as server:
            server.sendmail(MAIL_FROM, MAIL_TO, msg.as_string())
        print("[+] Notification envoyée avec succès !")
    except Exception as e:
        print(f"[-] Erreur d'envoi de l'email: {e}")

# Surveillance des logs SSH et Guacamole
def monitor_logs():
    auth_log = "/var/log/auth.log"
    guac_log = "/var/log/tomcat9/catalina.out"

    with open(auth_log, "r") as ssh_log, open(guac_log, "r") as guac:
        ssh_log.seek(0, 2)
        guac.seek(0, 2)

        while True:
            ssh_line = ssh_log.readline()
            guac_line = guac.readline()
```

Test :

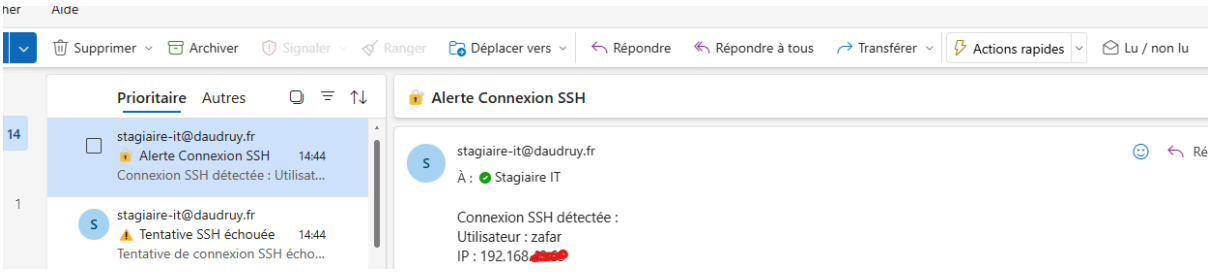
```
zafar@apache-guaca:/opt/scripts# nohup: ignoring input and appending output to '/home/zafar/nohup.out'
```

Je me connecte en ssh sur serveur

```
*** System restart required ***
Last login: Tue Feb  4 13:35:49 2025 from 192.168.40.65
 _____       __                  _   _ _
|__  /__ _ / _| __ _ _ __       | | | | |__   _   _ _ __ | |_ _   _
  / // _` | |_ / _` | '__|      | |_| | '_ \| | | | '_ \| __| | | |
 / /| (_| |  _| (_| | |         |  _  | |_) | |_| | | | | |_| |_| |
/____\__,_|_|  \__,_|_|         |_| |_|_.__/ \__,_|_| |_|\__|\__,_|

zafar@apache-guaca:~#
```

Mail si je tape mal le mot de passe

La je me suis connecté en ssh



Automatiser ce script avec systemd

```
 GNU nano 6.2                                  /etc/systemd/system/monitor.service
[Unit]
Description=Surveillance des connexions SSH et Guacamole
After=network.target

[Service]
ExecStart=/usr/bin/python3 /opt/scripts/monitor_guacamole_ssh.py
Restart=always
User=root

[Install]
WantedBy=multi-user.target
```

```
zafar@apache-guaca:/etc/systemd/system# sudo nano /etc/systemd/system/monitor.service
zafar@apache-guaca:/etc/systemd/system# sudo systemctl daemon-reload
zafar@apache-guaca:/etc/systemd/system# sudo systemctl enable monitor.service
zafar@apache-guaca:/etc/systemd/system# sudo systemctl start monitor.service
zafar@apache-guaca:/etc/systemd/system# sudo systemctl status monitor.service
● monitor.service - Surveillance des connexions SSH et Guacamole
     Loaded: loaded (/etc/systemd/system/monitor.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2025-02-04 14:09:14 UTC; 3s ago
   Main PID: 102321 (python3)
      Tasks: 1 (limit: 9394)
```

✅ **Parfait !** Mon service **monitor.service** fonctionne maintenant et tourne bien en arrière-plan. 🎉

le scripte complet :

```
import smtplib
import time
import re
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart

# Configuration SMTP
SMTP_SERVER = "smtp-mibc-fr-07.mailinblack.com"
SMTP_PORT = 25
MAIL_TO = "stagiaire-it@daudruy.fr"
MAIL_FROM = "alert@yourdomain.com"  # À remplacer par un mail valide

# Fonction d'envoi d'alerte
def send_alert(subject, message):
    try:
        msg = MIMEMultipart()
        msg['From'] = MAIL_FROM
        msg['To'] = MAIL_TO
        msg['Subject'] = subject

        msg.attach(MIMEText(message, 'plain'))

        with smtplib.SMTP(SMTP_SERVER, SMTP_PORT) as server:
            server.sendmail(MAIL_FROM, MAIL_TO, msg.as_string())
        print("[+] Notification envoyée avec succès !")
    except Exception as e:
        print(f"[-] Erreur d'envoi de l'email: {e}")

# Surveillance des logs SSH et Guacamole
def monitor_logs():
    auth_log = "/var/log/auth.log"
    guac_log = "/var/log/tomcat9/catalina.out"

    with open(auth_log, "r") as ssh_log, open(guac_log, "r") as guac:
        ssh_log.seek(0, 2)
        guac.seek(0, 2)

        while True:
            ssh_line = ssh_log.readline()
            guac_line = guac.readline()

            # Détection des connexions SSH réussies
            if ssh_line and "Accepted password" in ssh_line:
                user = re.search(r'Accepted password for (\w+)', ssh_line)
                ip = re.search(r'from ([\d\.]+)', ssh_line)
                if user and ip:
                    msg = f"Connexion SSH détectée :\nUtilisateur : {user.group(1)}\nIP : {ip.group(1)}"
                    send_alert("🔐 Alerte Connexion SSH", msg)

            # Détection des échecs SSH
            if ssh_line and "Failed password" in ssh_line:
                ip = re.search(r'from ([\d\.]+)', ssh_line)
                if ip:
                    msg = f"Tentative de connexion SSH échouée depuis {ip.group(1)}"
                    send_alert("⚠️ Tentative SSH échouée", msg)

            # Détection des connexions Guacamole
            if guac_line and "User \" " in guac_line and "connected from" in guac_line:
                user = re.search(r'User \"(.*?)\"', guac_line)
                ip = re.search(r'from ([\d\.]+)', guac_line)
                if user and ip:
                    msg = f"Connexion Guacamole détectée :\nUtilisateur : {user.group(1)}\nIP : {ip.group(1)}"
                    send_alert("🖥️ Connexion Guacamole", msg)

            time.sleep(1)

if __name__ == "__main__":
    monitor_logs()
```

Script de notification

1. Installer msmtp

```
root@apache-guaca:~# sudo apt-get install msmtp
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  libsecret-1-0 libsecret-common
Paquets suggérés :
  msmtp-mta
Les NOUVEAUX paquets suivants seront installés :
```

```
 GNU nano 6.2                                    /home/zafar/.msmtprc
# Paramètres par défaut
defaults
auth            off
tls             off
logfile         ~/.msmtp.log

# Configuration du compte par défaut
account default
host            smtp-mibc-fr-07.mailinblack.com
port            25
from            stagiaire-it@daudruy.fr
```

```
root@apache-guaca:/opt/scripts# chmod 600 /home/zafar/.msmtprc
```

Puisque le script tourne sous `root`, la config a été copiée dans `/root/.msmtprc` :
sudo cp /home/zafar/.msmtprc /root/
sudo chown root:root /root/.msmtprc
sudo chmod 600 /root/.msmtprc

zafar

Mise en place du script de surveillance des logs

```
  GNU nano 6.2                                  /opt/scripts/watch_guac_log.sh
#!/bin/bash
#
# Script qui surveille les logs de Guacamole et envoie une notification lors d'une connexion.

LOGFILE="/var/log/tomcat9/catalina.out"
KEYWORD="User .* connected to connection"

# Suivi en temps réel des logs
tail -n0 -F "$LOGFILE" | while read -r LINE; do
    # Vérifie si la ligne contient le mot-clé indiquant une connexion
    echo "$LINE" | grep -E "$KEYWORD" > /dev/null
    if [ $? -eq 0 ]; then
        # Extraction du nom de l'utilisateur et de la connexion
        USER_NAME=$(echo "$LINE" | awk -F'"' '{print $2}')
        CONNECTION_ID=$(echo "$LINE" | awk -F'connected to connection "' '{print $2}' | awk -F'"' '{print $1}')
        DATE_CONNEXION=$(date '+%Y-%m-%d %H:%M:%S')

        # Log local (pour debug)
        echo "$(date '+%Y-%m-%d %H:%M:%S') - Connexion détectée : $USER_NAME sur connexion $CONNECTION_ID" >> /tmp/guac_notify_watch.log

        # Appel du script de notification
        /opt/scripts/guac_notify.sh "$USER_NAME" "$CONNECTION_ID"
    fi
done
```

On le rende executable : chmod +x /opt/scripts/watch_guac_log.sh

```
root@apache-guaca:/opt/scripts# ls -l watch_guac_log.sh
-rwxr-xr-x 1 root root 1001 févr.  3 14:48 watch_guac_log.sh
root@apache-guaca:/opt/scripts#
```

Mise en place du script de notification (`guac_notify.sh`)
Ce script est déclenché par `watch_guac_log.sh` pour envoyer un email.

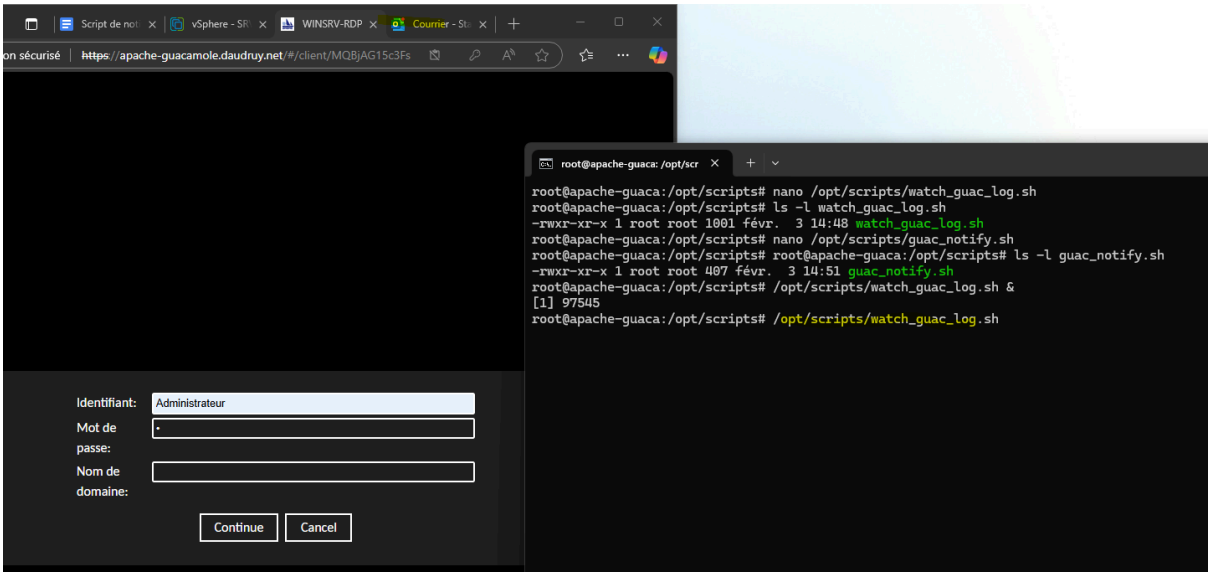
```
  GNU nano 6.2                                  /opt/scripts/guac_notify.sh
#!/bin/bash

USER_NAME="$1"
CONNECTION_ID="$2"
DATE_CONNEXION=$(date '+%Y-%m-%d %H:%M:%S')

SUBJECT="Connexion Guacamole - Utilisateur: $USER_NAME"
BODY="Salut,\n\nL'utilisateur $USER_NAME s'est connecté à Guacamole.\nDate et heure: $DATE_CONNEXION\nConnexion ID: $CONNECTION_ID\n\nÀ plus !"

echo -e "Subject: $SUBJECT\n\n$BODY" | /usr/bin/msmtp --file=/root/.msmtprc -a default stagiaire-it@daudruy.fr
```

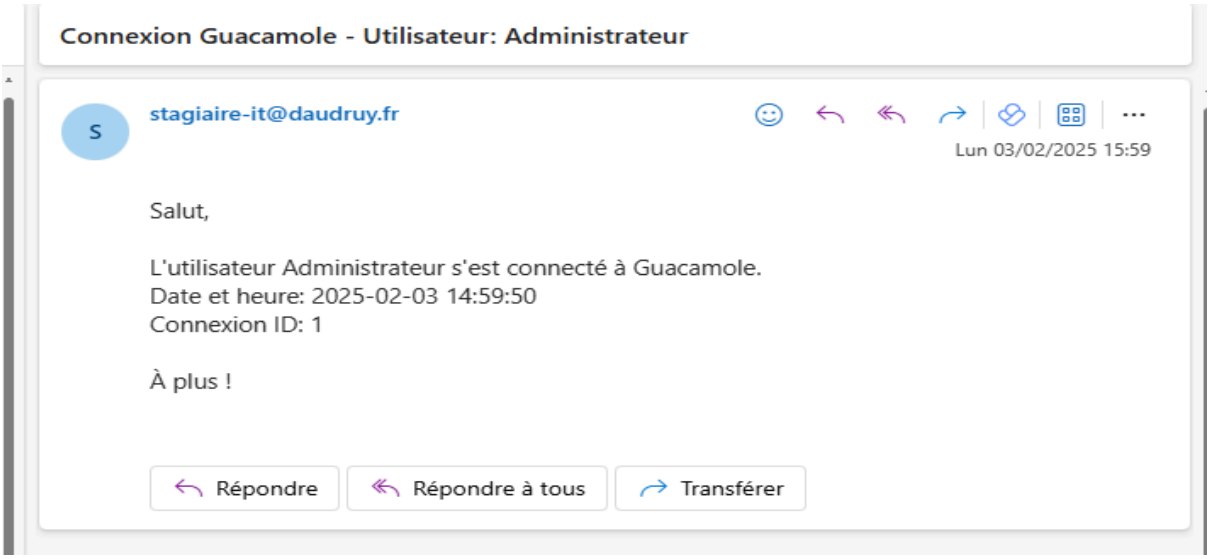
Rendre le script exécutable: `chmod +x /opt/scripts/guac_notify.sh`

```
root@apache-guaca:/opt/scripts# root@apache-guaca:/opt/scripts# ls -l guac_notify.sh
-rwxr-xr-x 1 root root 407 févr.  3 14:51 guac_notify.sh
root@apache-guaca:/opt/scripts#
```

zafar

Lancer manuellement pour le teste :





Pour s'assurer que le script démarre automatiquement au redémarrage.

```
  GNU nano 6.2          /etc/systemd/system/watch_guac.service *
[Unit]
Description=Surveillance des connexions Guacamole et envoi de notifications
After=network.target

[Service]
ExecStart=/opt/scripts/watch_guac_log.sh
Restart=always
User=root

[Install]
WantedBy=multi-user.target
```

Activer et démarrer le service

```
root@apache-guaca:/opt/scripts# sudo systemctl daemon-reload
root@apache-guaca:/opt/scripts# sudo systemctl enable watch_guac.service
root@apache-guaca:/opt/scripts# sudo systemctl start watch_guac.service
root@apache-guaca:/opt/scripts# sudo systemctl status watch_guac.service
● watch_guac.service - Surveillance des connexions Guacamole et envoi de notifications
     Loaded: loaded (/etc/systemd/system/watch_guac.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2025-02-03 15:02:51 UTC; 55s ago
   Main PID: 98240 (watch_guac_log.)
      Tasks: 3 (limit: 9394)
     Memory: 740.0K
        CPU: 2ms
     CGroup: /system.slice/watch_guac.service
             ├─98240 /bin/bash /opt/scripts/watch_guac_log.sh
             ├─98241 tail -n0 -F /var/log/tomcat9/catalina.out
             └─98242 /bin/bash /opt/scripts/watch_guac_log.sh
```

Vérifier les logs du script

```
root@apache-guaca:/opt/scripts# cat /tmp/guac_notify_watch.log
2025-02-03 14:49:25 - Connexion détectée : Administrateur sur connexion 1
2025-02-03 14:49:36 - Connexion détectée : Administrateur sur connexion 1
2025-02-03 14:51:58 - Connexion détectée : Administrateur sur connexion 1
2025-02-03 14:52:32 - Connexion détectée : Administrateur sur connexion 1
2025-02-03 14:59:50 - Connexion détectée : Administrateur sur connexion 1
2025-02-03 14:59:50 - Connexion détectée : Administrateur sur connexion 1
root@apache-guaca:/opt/scripts#
```

Vérifier les logs systemd

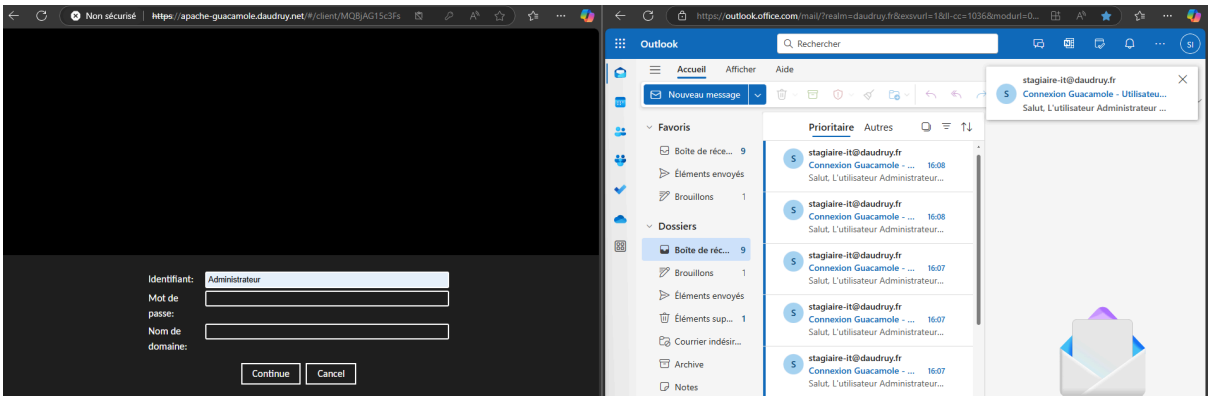
```
root@apache-guaca:/opt/scripts# sudo journalctl -u watch_guac.service --no-pager | tail -n 5
0
févr. 03 15:02:51 apache-guaca systemd[1]: Started Surveillance des connexions Guacamole et
envoi de notifications.
root@apache-guaca:/opt/scripts#
```

zafar

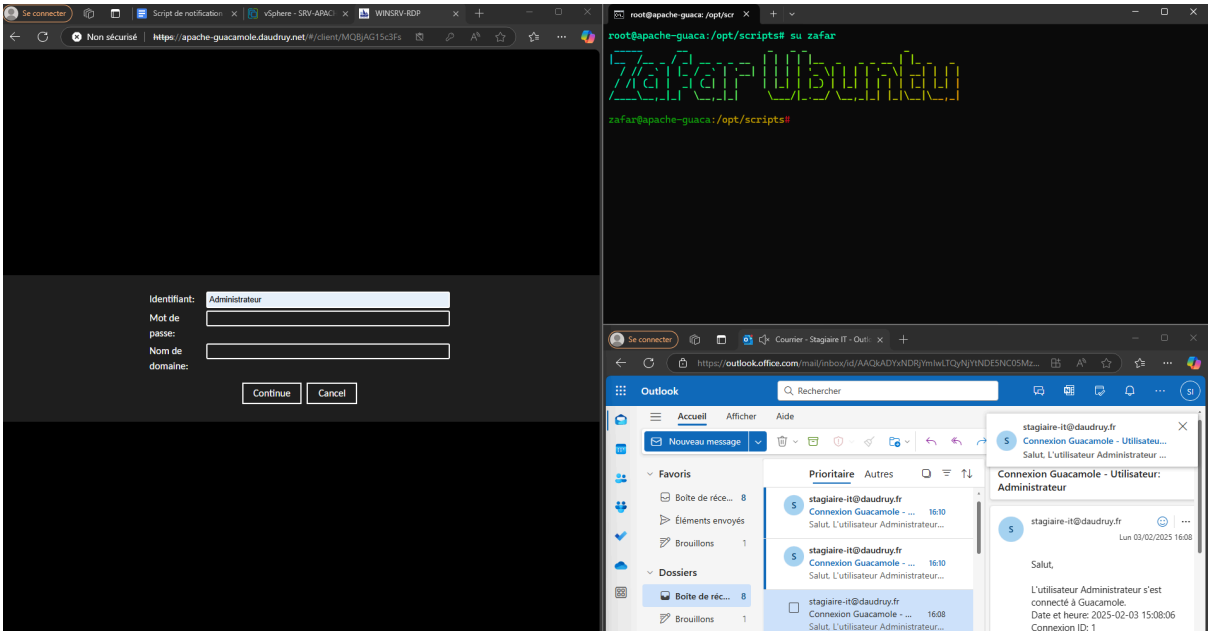
```
root@apache-guaca:/opt/scripts# /opt/scripts/guac_notify.sh testuser 1234.
root@apache-guaca:/opt/scripts# cat ~/.msmtp.log
févr. 03 14:51:58 host=smtp-mibc-fr-07.mailinblack.com tls=off auth=off from=stagiaire-it@daudruy.fr r
ecipients=stagiaire-it@daudruy.fr mailsize=257 smtpstatus=250 smtpmsg='250 2.0.0 Ok: queued as 39E1412
007F' exitcode=EX_OK
févr. 03 14:52:32 host=smtp-mibc-fr-07.mailinblack.com tls=off auth=off from=stagiaire-it@daudruy.fr r
ecipients=stagiaire-it@daudruy.fr mailsize=257 smtpstatus=250 smtpmsg='250 2.0.0 Ok: queued as 6AB9D12
0083' exitcode=EX_OK
```

🎉 Tout est maintenant automatisé.

Dèse que le utilisateur met son identifiant les scrippte les detecte et envoite un notification



Et ca marche en étant user normal :

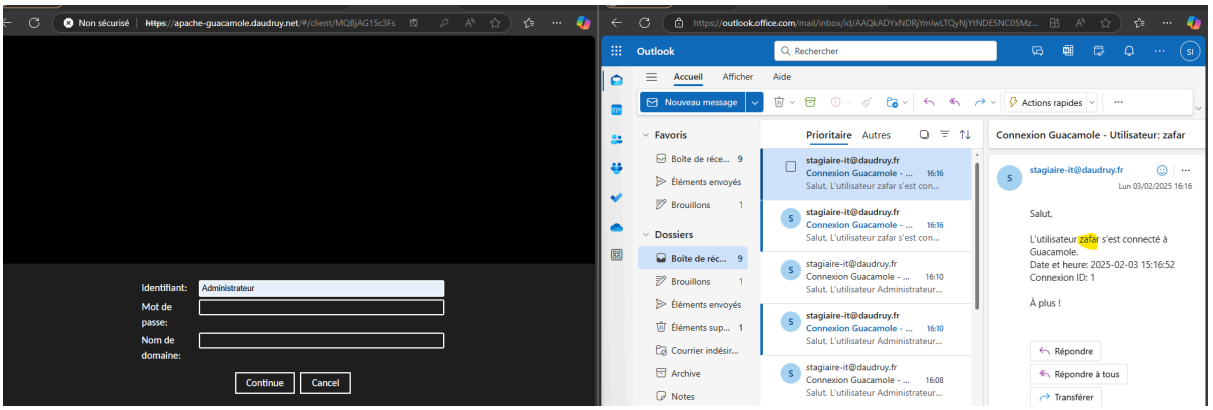


zafar

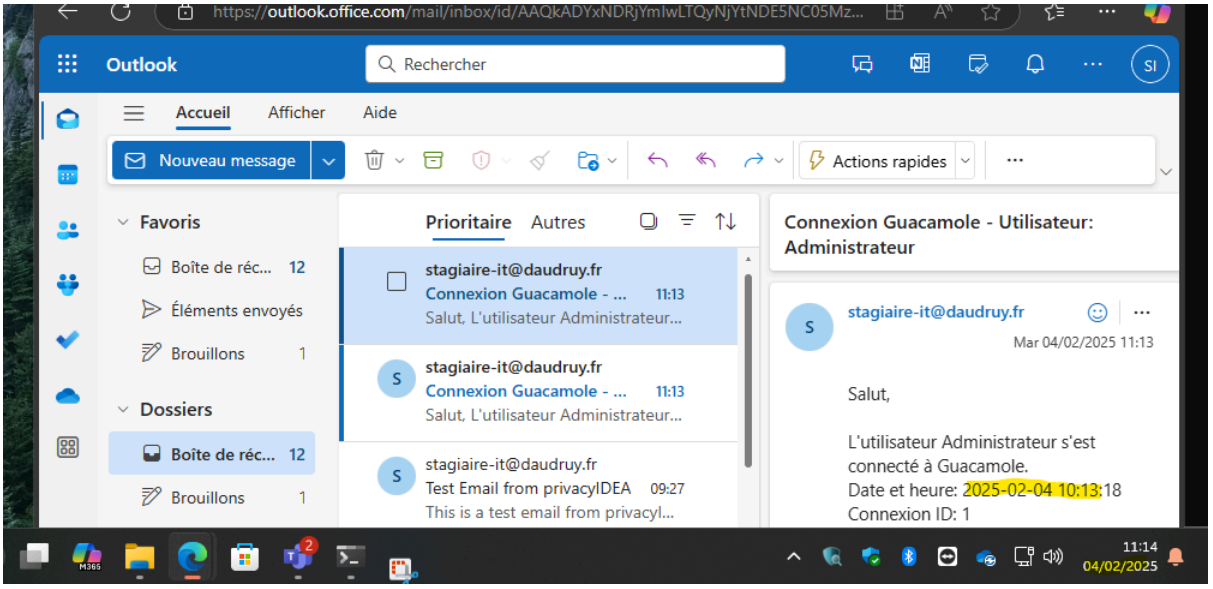
Les droit :

```
zafar@apache-guaca:/opt/scripts# ls -l /opt/scripts/
total 16
-rwxr-xr-x 1 zafar root 3618 janv. 28 11:48 envoie_et_nettoie.sh
-rwxr-xr-x 1 root  root  407 févr.  3 14:51 guac_notify.sh
-rwxr-xr-x 1 root  root  417 janv. 28 13:54 nas_supprime.sh
-rwxr-xr-x 1 root  root 1001 févr.  3 14:48 watch_guac_log.sh
```

Teste avec autre utilisateur :



Test script un jours après config :

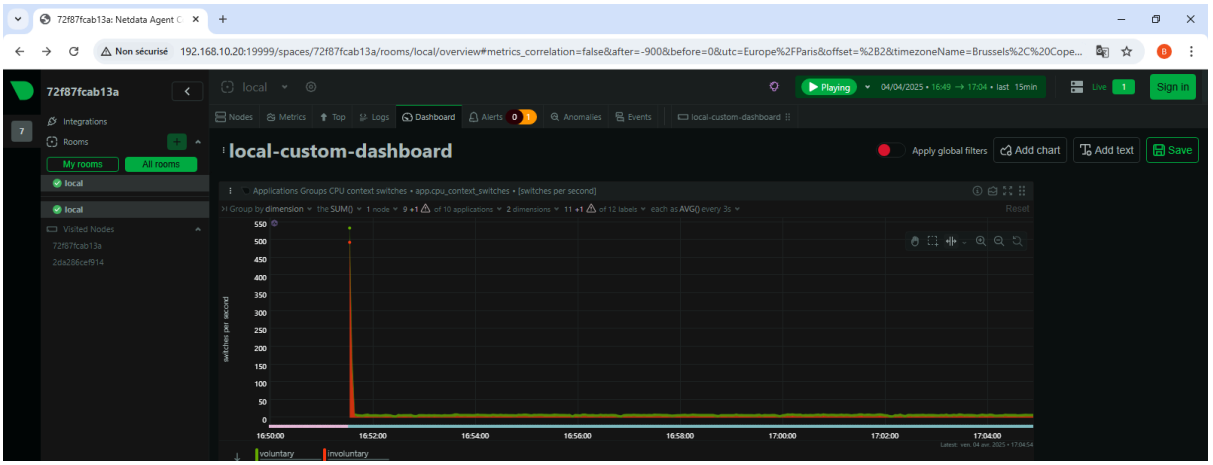


Supervision avec Netdata sur Docker

```
root@deb12:~# sudo apt install docker.io -y
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets suivants ont été installés automatiquement et ne sont
```

```
root@deb12:~#
root@deb12:~# sudo mkdir -p /etc/systemd/system/docker.service.d
root@deb12:~# sudo nano /etc/systemd/system/docker.service.d/http-proxy.conf
root@deb12:~# sudo systemctl daemon-reload
root@deb12:~# sudo systemctl restart docker
root@deb12:~# docker run -d --name=netdata \
>   -p 19999:19999 \
>   --cap-add=SYS_PTRACE \
>   --security-opt apparmor=unconfined \
>   netdata/netdata

Unable to find image 'netdata/netdata:latest' locally
latest: Pulling from netdata/netdata
7cd785773db4: Pull complete
74a6e2c79650: Pull complete
08dcfea3bbfd: Pull complete
68d8042082b7: Pull complete
079073c61649: Pull complete
Digest: sha256:809e61a27c6b79c12a8faeaf3495fb7edb4a87bdea24e1b7fef6de9b1d6c2245
Status: Downloaded newer image for netdata/netdata:latest
72f87fcab13a0b8341a2588cd6723329b991527835bcad1b32c7465f416937ed
root@deb12:~#
```

Puis acces intefaceip et le port 19999



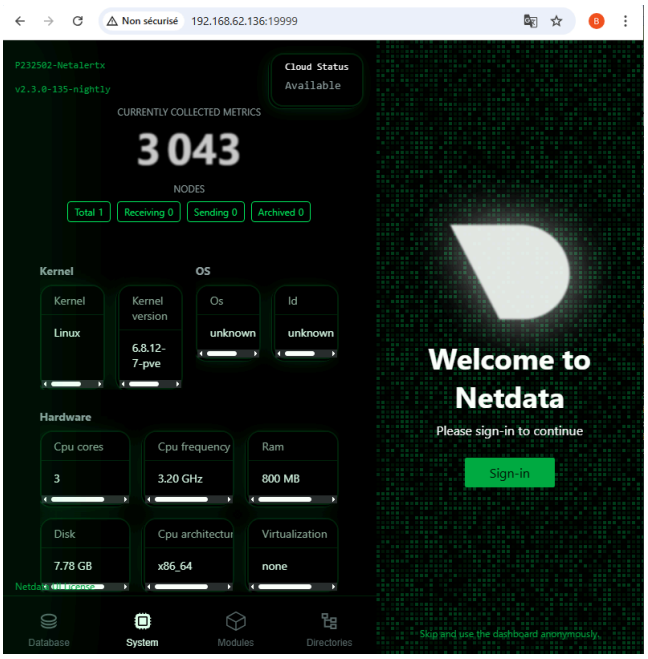
Étape 1 : Installer Netdata sur la machine cible

Pour commencer, tu dois mettre Netdata sur la machine que tu veux superviser.

Sur cette machine cible, lance simplement :

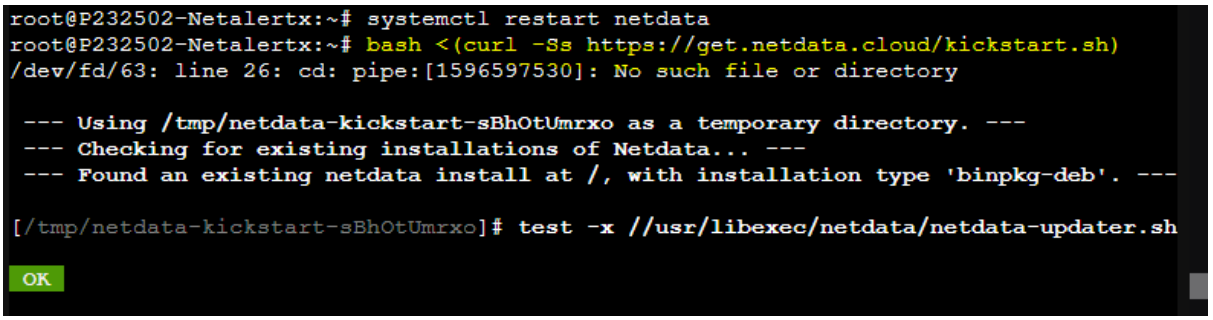
```
root@P232502-Netalertx:~# bash <(curl -Ss https://get.netdata.cloud/kickstart.sh)
/dev/fd/63: line 26: cd: pipe:[1596597530]: No such file or directory

 --- Using /tmp/netdata-kickstart-sBhOtUmrxo as a temporary directory. ---
 --- Checking for existing installations of Netdata... ---
 --- Found an existing netdata install at /, with installation type 'binpkg-deb'. ---

[/tmp/netdata-kickstart-sBhOtUmrxo]# test -x //usr/libexec/netdata/netdata-updater.sh

OK

[/tmp/netdata-kickstart-sBhOtUmrxo]# //usr/libexec/netdata/netdata-updater.sh --inter
```

tu accèdes directement à chaque machine par son IP

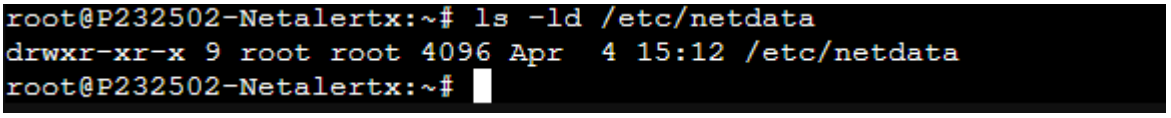


Objectif : Centraliser les données sur un serveur principal avec Netdata.

🚀 Étape 1 : Vérifie l'installation de Netdata sur machine a supervisie



bash <(curl -Ss https://get.netdata.cloud/kickstart.sh)

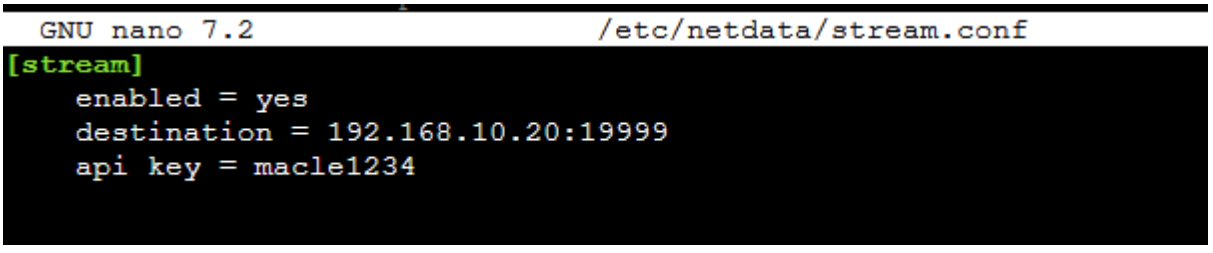


🌟 Étape 2 : cree le dossier sur machine a supervisi :

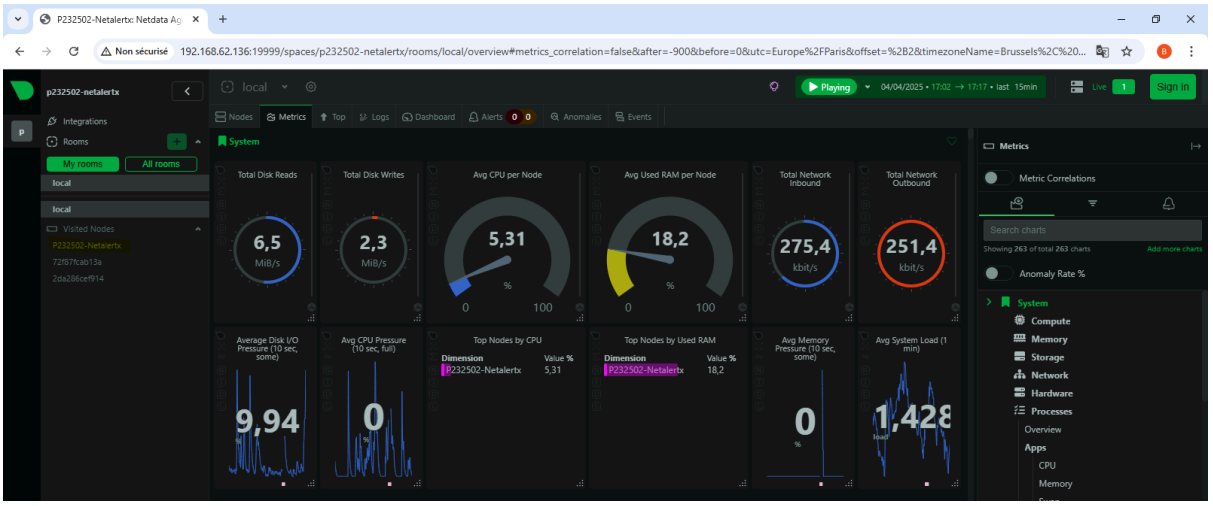
sudo mkdir /etc/netdata
sudo nano /etc/netdata/stream.conf
Puis ajouute ca :

[stream]
enabled = yes
destination = <IP_du_serveur_principla:19999
api key = macle1234

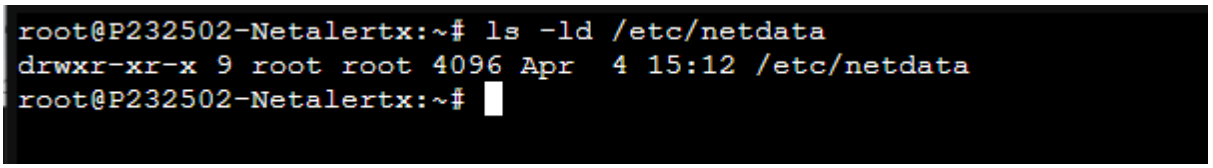
sudo systemctl restart netdata



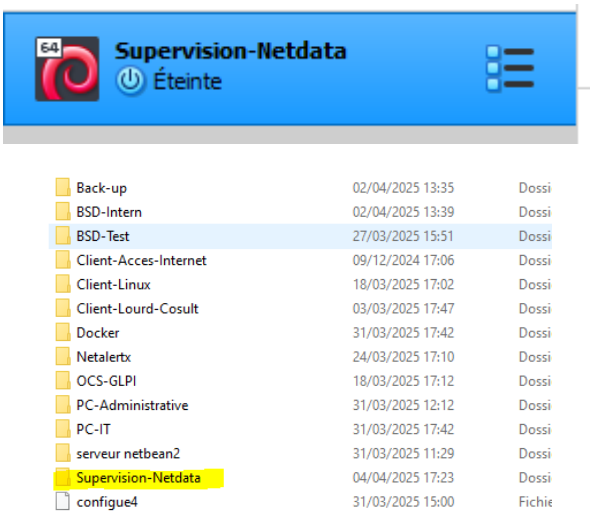
Test on vas sur serveur principle :



Il survaillle cett machine:



Le serveur netdata je l'ai dans le local :



Pour apres vacance Il faut aller sur chaque machine et installee le pacque et cree le fichier puis c'est bon tu supervise soit ca ou bien sur docker que tu as dan le lan tu reinstalle et recommnace des le zero

SIDE PROJECT

AI server for high-performance, local

Mistral

Running the Docker Container

First, ensure you have Docker installed and your GPU drivers are up to date. Use the following command to run the Mistral AI LLM Inference image:

```
zafar@auth-srv:~$ docker run --gpus all \
    -e HF_TOKEN=$HF_TOKEN -p 8000:8000 \
    ghcr.io/mistralai/mistral-src/vllm:latest \
    --host 0.0.0.0 \
    --model mistralai/Mistral-7B-Instruct-v0.2
Unable to find image 'ghcr.io/mistralai/mistral-src/vllm:latest' locally
latest: Pulling from mistralai/mistral-src/vllm
43f89b94cd7d: Pull complete
45f7ea5367fe: Pull complete
3d97a47c3c73: Pull complete
12cd4d19752f: Pull complete
da5a484f9d74: Pull complete
5e5846364eee: Downloading [=======>                                           ]  163.4MB/1.291GB
fd355de1d1f2: Download complete
3480bb79c638: Download complete
e7016935dd60: Download complete
99541166a133: Downloading [>                                                  ]  35.63MB/2.509GB
8999112df5b0: Download complete
e969c5eb17ee: Download complete
174617b6ae76: Download complete
7fcb0eeb3246: Waiting
8546325b89a2: Waiting
fd3e44b6510f: Waiting
1ad8795b31a4: Waiting
962181193532: Waiting
ccb00ad5abb9: Waiting
fa4989232485: Waiting
```

Authentication via Privacyidea

1. Mise à jour du système

Avant tout, mets à jour les paquets existants :

```
zafar@auth-srv:~$ sudo apt update && sudo apt upgrade -y
```

les outils nécessaires sont installés :

```
zafar@auth-srv:~$ sudo apt install -y wget gnupg software-properties-common
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
wget est déjà la version la plus récente (1.21.4-1ubuntu4.1).
wget passé en « installé manuellement ».
gnupg est déjà la version la plus récente (2.4.4-2ubuntu17).
gnupg passé en « installé manuellement ».
software-properties-common est déjà la version la plus récente (0.99.49.1).
software-properties-common passé en « installé manuellement ».
Le paquet suivant a été installé automatiquement et n'est plus nécessaire :
  libllvm17t64
Veuillez utiliser « sudo apt autoremove » pour le supprimer.
0 mis à jour, 0 nouvellement installés, 0 à enlever et 2 non mis à jour.
zafar@auth-srv:~$
```

Objectif

1️⃣ Un utilisateur tente de se connecter à Guacamole
2️⃣ Guacamole redirige la connexion vers PrivacyIDEA
3️⃣ PrivacyIDEA envoie un OTP par e-mail
4️⃣ L'utilisateur entre son OTP et accède à Guacamole

2. Ajouter le dépôt de privacyIDEA

Comme il n'existe pas encore de dépôt officiel pour **Ubuntu 24.04 (Noble)**, on va utiliser le dépôt de **Ubuntu 22.04 (Jammy)** :

Télécharger la clé GPG du dépôt :

Puis on déplace la clé dans le bon dossier avec les privilèges root

```
zafar@auth-srv:~$
wget https://lancelot.netknights.it/NetKnights-Release.asc
--2025-02-03 08:34:41--  https://lancelot.netknights.it/NetKnights-Release.asc
Resolving lancelot.netknights.it (lancelot.netknights.it)... 46.4.108.34
Connecting to lancelot.netknights.it (lancelot.netknights.it)|46.4.108.34|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3096 (3,0K) [application/octet-stream]
Saving to: 'NetKnights-Release.asc'

NetKnights-Release.asc        100%[=============================================>]   3,02K  --.-KB/s    in 0s

2025-02-03 08:34:41 (908 MB/s) - 'NetKnights-Release.asc' saved [3096/3096]

zafar@auth-srv:~$ sudo mv NetKnights-Release.asc /etc/apt/trusted.gpg.d/
zafar@auth-srv:~$ ls /etc/apt/trusted.gpg.d/
NetKnights-Release.asc  ubuntu-keyring-2012-cdimage.gpg  ubuntu-keyring-2018-archive.gpg
zafar@auth-srv:~$
```

On vois **NetKnights-Release.asc**, c'est bon 👍

Maintenant que la clé est bien ajoutée, on peut passer à l'étape suivante en ajoutant le dépôt et en mettant à jour :

```
zafar@auth-srv:~$ echo "deb http://lancelot.netknights.it/community/jammy/stable jammy main" | sudo tee /etc/apt/sources
.list.d/privacyidea.list
deb http://lancelot.netknights.it/community/jammy/stable jammy main
zafar@auth-srv:~$ sudo apt update
Atteint :1 http://security.ubuntu.com/ubuntu noble-security InRelease
Atteint :2 http://archive.ubuntu.com/ubuntu noble InRelease
```

Installer privacyIDEA avec Apache

```
zafar@auth-srv:~$ sudo apt install -y privacyidea-apache2
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Le paquet suivant a été installé automatiquement et n'est
```

Vérifier que privacyIDEA est bien installé :

```
zafar@auth-srv:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Mon 2025-02-03 08:38:54 UTC; 47s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 32144 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 32149 (apache2)
      Tasks: 55 (limit: 4613)
```

```
zafar@auth-srv:~$ dpkg -l | grep privacyidea
ii  privacyidea                          3.10.2-1noble                           amd64        two-fact
or authentication system e.g. for OTP devices
ii  privacyidea-apache2                  3.10.2-1noble                           all          2FA syst
em. This is a meta package to install privacyidea with apache2
zafar@auth-srv:~$
```

```
             _                    _______  _______
   ___  ____(_)  _____ _______ __/  _/ _ \/ __/ _ |
  / _ \/ __/ / |/ / _ `/ __/ // // // // / _// __ |
 / .__/_/ /_/|___/\_,_/\__/\_, /___/____/___/_/ |_|
/_/                       /___/
                                               v3.10.2

Encryption key written to /etc/privacyidea/enckey
The file permission of /etc/privacyidea/enckey was set to 400!
Please ensure, that it is owned by the right user.
```

Créer un administrateur pour privacyIDEA :

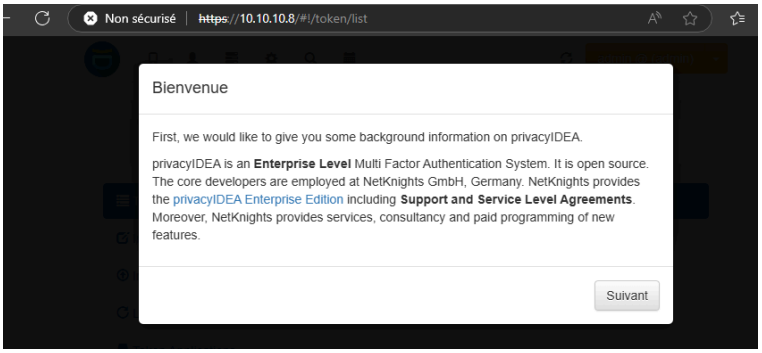
```
zafar@auth-srv:~$ sudo pi-manage admin add admin -e admin@localhost -p "TonMotDePasse"

             _                    _______  _______
   ___  ____(_)  _____ _______ __/  _/ _ \/ __/ _ |
  / _ \/ __/ / |/ / _ `/ __/ // // // // / _// __ |
 / .__/_/ /_/|___/\_,_/\__/\_, /___/____/___/_/ |_|
/_/                       /___/
                                               v3.10.2

Admin admin was registered successfully.
zafar@auth-srv:~$ |
```

```
zafar@auth-srv:~$ sudo pi-manage admin reset admin -p "admin"

             _                    _______  _______
   ___  ____(_)  _____ _______ __/  _/ _ \/ __/ _ |
  / _ \/ __/ / |/ / _ `/ __/ // // // // / _// __ |
 / .__/_/ /_/|___/\_,_/\__/\_, /___/____/___/_/ |_|
/_/                       /___/
                                               v3.10.2

Usage: pi-manage admin [OPTIONS] COMMAND [ARGS]...
Try 'pi-manage admin -h' for help.

Error: No such command 'reset'.
zafar@auth-srv:~$ |
```

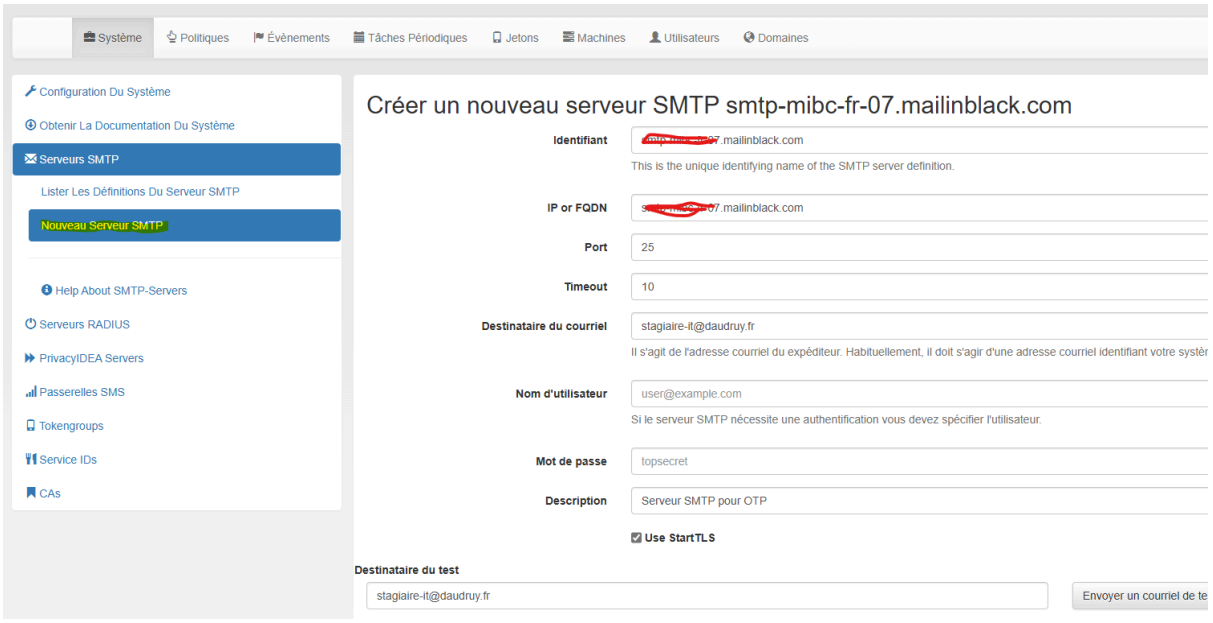
```
zafar@auth-srv:~$ dpkg -l | grep privacyidea
ii  privacyidea                    3.10.2-1jammy                   amd64        two-factor authentication
system e.g. for OTP devices
iF  privacyidea-apache2           3.10.2-1jammy                   all          2FA system. This is a meta
 package to install privacyidea with apache2
zafar@auth-srv:~$
```

```
zafar@auth-srv:~$ systemctl restart apache2
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to restart 'apache2.service'.
Authenticating as: zafar
Password:
==== AUTHENTICATION COMPLETE ====
zafar@auth-srv:~$
```

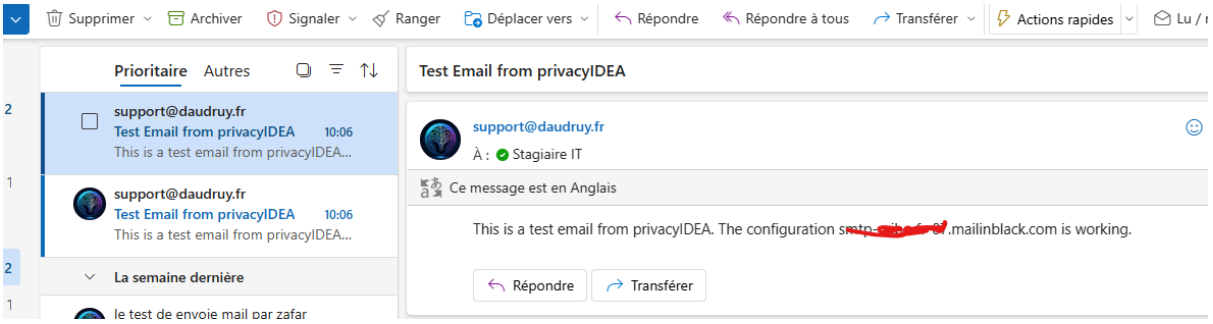
Accéder à l'interface web 👍



Configurer le serveur SMTP dans privacyIDEA :



Test smtp : 👍



Privacyidea a beson une base et il peut aller chercher les user sur la bas de guacamole

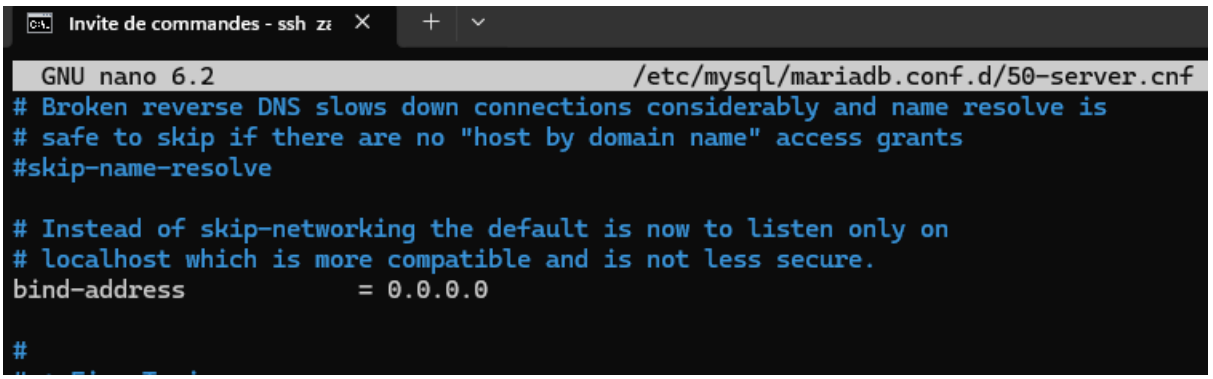
Trouver la structure de la table des utilisateurs dans MySQL/MariaDB

Pour que PrivacyIDEA puisse récupérer les utilisateurs, on doit voir comment Guacamole stocke ses comptes.

Sur guacamole on cree un user :

```
MariaDB [guacadb]> CREATE USER 'privacyidea'@'10.10.10.8' IDENTIFIED BY 'zafar';
Query OK, 0 rows affected (0,002 sec)

MariaDB [guacadb]> GRANT ALL PRIVILEGES ON guacadb.* TO 'privacyidea'@'10.10.10.8';
Query OK, 0 rows affected (0,002 sec)

MariaDB [guacadb]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,000 sec)

MariaDB [guacadb]> SHOW GRANTS FOR 'privacyidea'@'10.10.10.8';
+-----------------------------------------------------------------------------------------------------------------+
| Grants for privacyidea@10.10.10.8                                                                               |
+-----------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `privacyidea`@`10.10.10.8` IDENTIFIED BY PASSWORD '*EE159D4B82B52E5C9F27A79925E9E29BB7B840A6' |
| GRANT ALL PRIVILEGES ON `guacadb`.* TO `privacyidea`@`10.10.10.8`                                               |
+-----------------------------------------------------------------------------------------------------------------+
2 rows in set (0,000 sec)

MariaDB [guacadb]>
```

Sur guacamole Si MySQL n'écoute que sur `127.0.0.1`, il faut modifier la config.



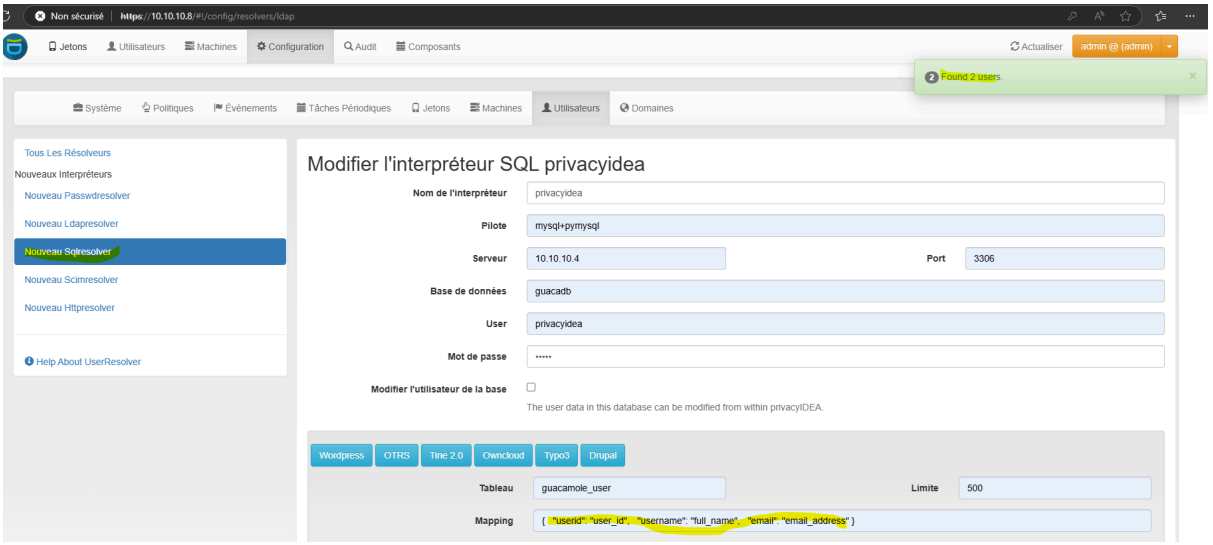
Ouvrir le port 3306 dans le pare-feu Autorise PrivacyIDEA (`10.10.10.8`) à accéder à MySQL :

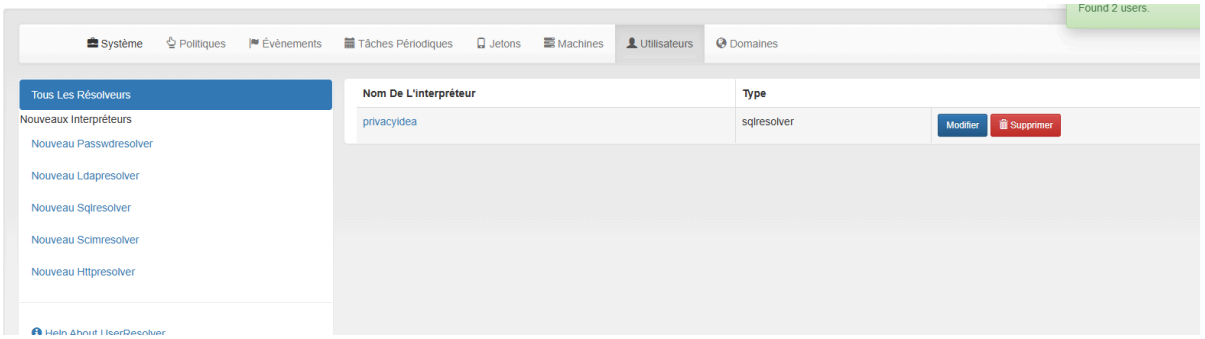
```
zafar@apache-guaca:~# sudo ufw allow from 10.10.10.8 to any port 3306 proto tcp
Rule added
zafar@apache-guaca:~# sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
3389                       ALLOW       Anywhere
4822                       ALLOW       Anywhere
8080                       ALLOW       Anywhere
22                         ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere
443/tcp                    ALLOW       Anywhere
80                         ALLOW       Anywhere
443                        ALLOW       Anywhere
3306/tcp                   ALLOW       10.10.10.8
```

On test la connexion a la base depuis Privacyidea :

```
zafar@auth-srv:~$ mysql -u privacyidea -p -h 10.10.10.4 -D guacadb
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 31
Server version: 5.5.5-10.6.18-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> exit
Bye
```

Configuration du **résolveur SQL** dans PrivacyIDEA pour récupérer les utilisateurs depuis `guacamole_user`.

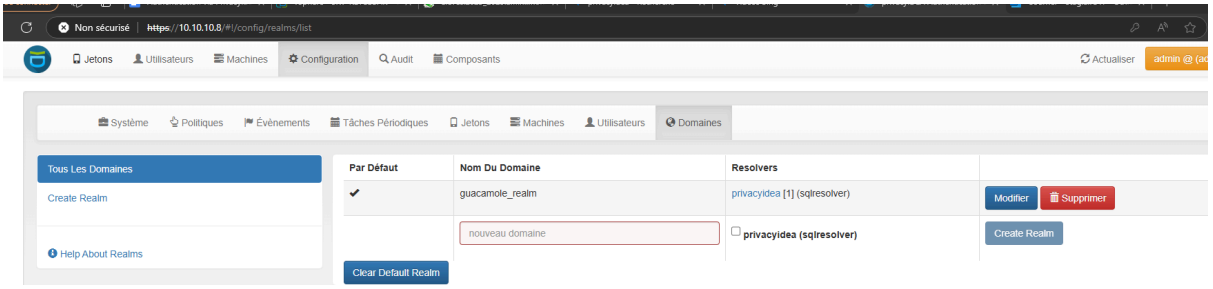




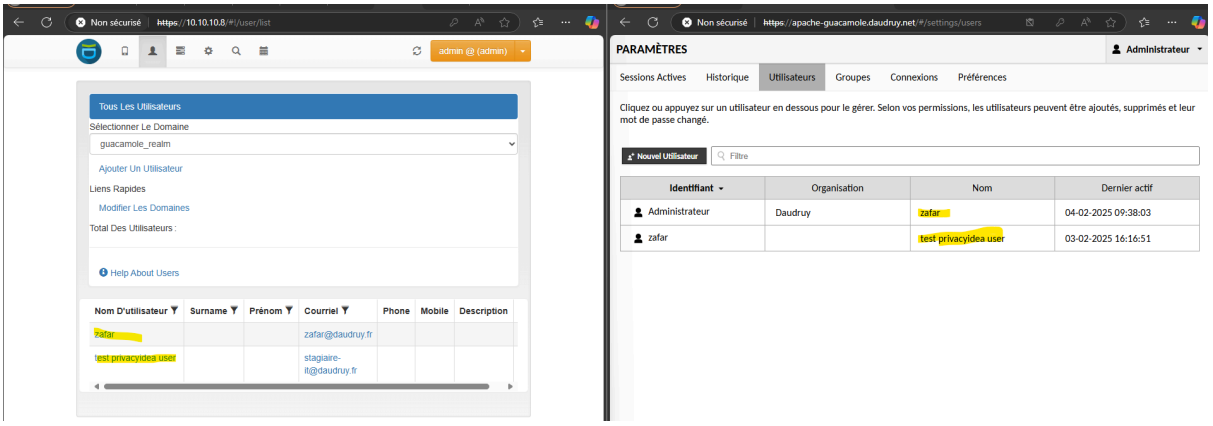
Créer un domaine (Realm) dans PrivacyIDEA

Un **realm** est un groupe d'utilisateurs géré par PrivacyIDEA. On doit lier notre résolveur SQL à un domaine.

📌 **Objectif** : Quand un utilisateur essaie de se connecter, PrivacyIDEA va chercher les comptes dans `guacamole_mysql`.



Vérifier que PrivacyIDEA trouve bien les utilisateurs du Realm

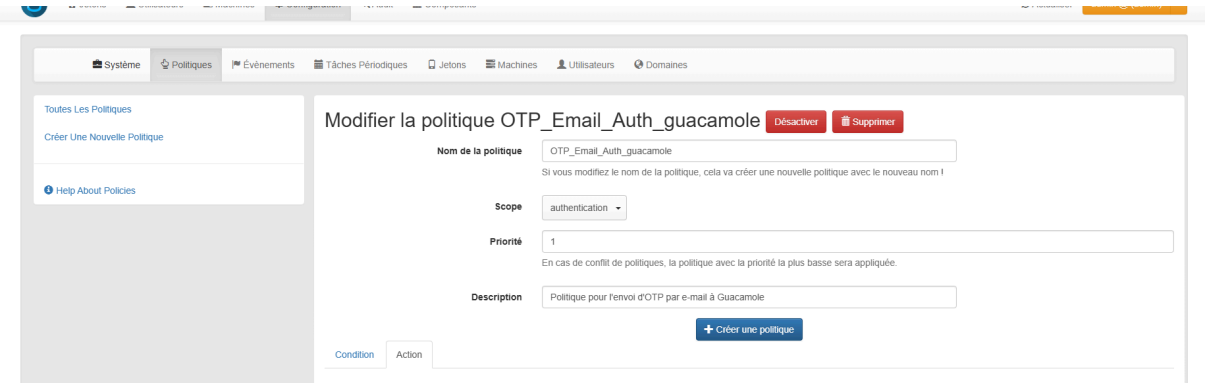


Modifier Guacamole pour pointer vers le bon Realm

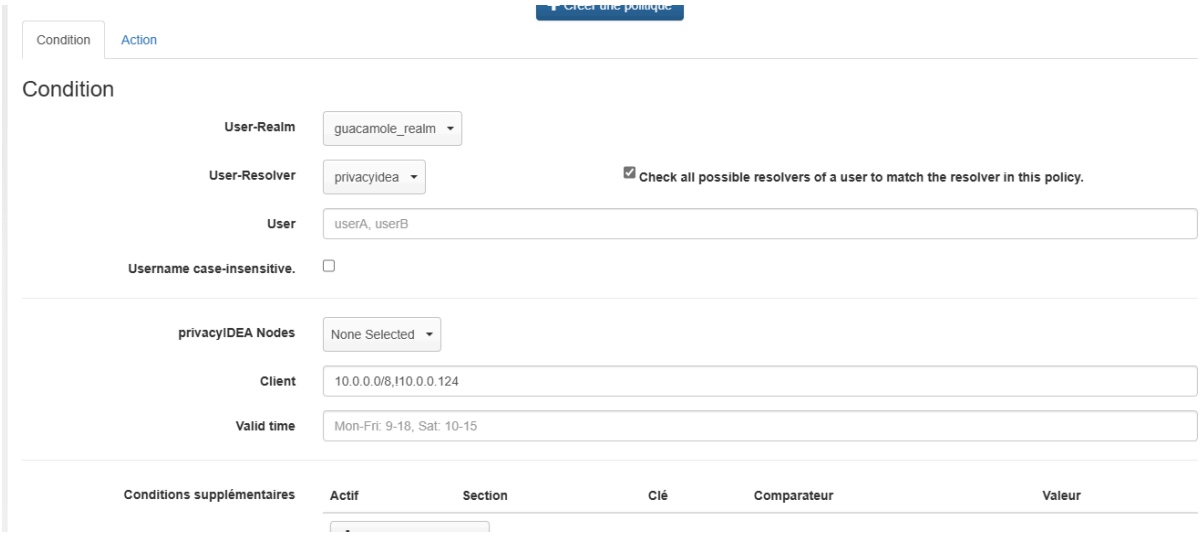
Super ! 🎉 Maintenant que **PrivacyIDEA récupère bien les utilisateurs de Guacamole via le domaine `guacamole_realm`**, voici les **dernières étapes** pour finaliser l'authentification OTP.

Activer l’authentification OTP pour les utilisateurs

On crée un politique de autentification



Puis on intègre notre domaine et le user qui vas chercher le user dans guacamole



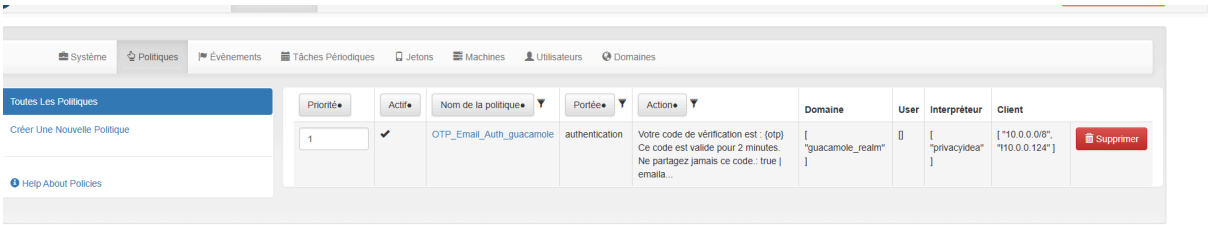
Dans la section **Action**, on cherche ces paramètres et configurer-les :

Clé : `emailautosend`
📌 Cela permet d'envoyer un OTP automatiquement à chaque tentative de connexion.

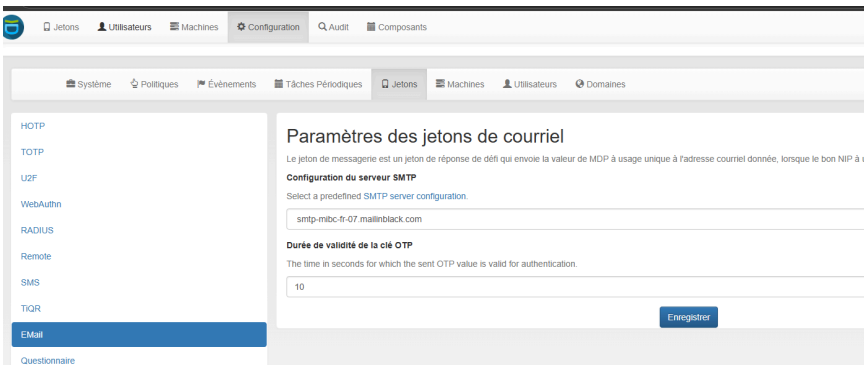
Clé : `emailsubject`
Valeur : `Votre code OTP pour la connexion à Guacamole`

Clé : emailtext
Valeur : Bonjour,
Votre code de vérification est : {otp}
Ce code est valide pour 2 minutes.
Ne partagez jamais ce code.

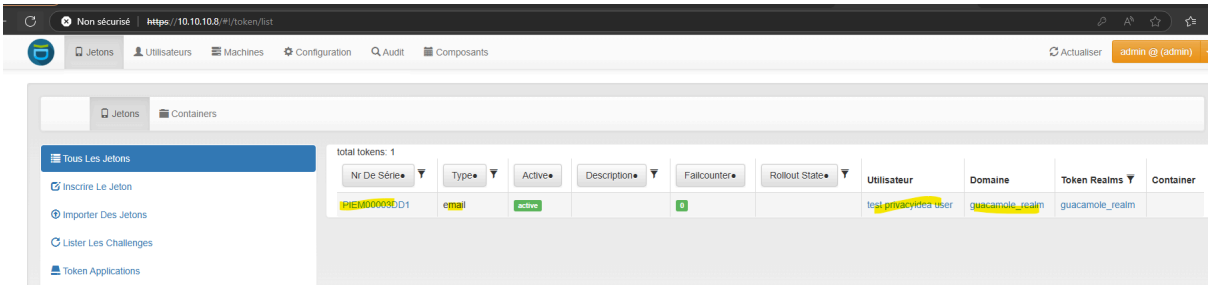
📌 {otp} est un paramètre dynamique qui sera remplacé par le code OTP.



Vérifier que l'OTP est bien actif pour chaque utilisateur

Le smtp  :

l’utilisateur a bien un "Jeton OTP Email" actif



PrivacyIDEA est bien configuré (SMTP, SQL Resolver, Politique OTP), mais il faut maintenant l’intégrer avec Guacamole pour que l’authentification OTP fonctionne.

Télécharger et installer l’extension OpenID sur guacamole

```
zafar@apache-guaca:~# wget https://dlcdn.apache.org/guacamole/1.5.5/binary/guacamole-auth-sso-1.5.5.ta
r.gz
--2025-02-04 09:19:40--  https://dlcdn.apache.org/guacamole/1.5.5/binary/guacamole-auth-sso-1.5.5.tar.
gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 38286364 (37M) [application/x-gzip]
Saving to: ‘guacamole-auth-sso-1.5.5.tar.gz’

guacamole-auth-sso-1.5.5. 100%[===================================>]  36,51M  87,5MB/s    in 0,4s

2025-02-04 09:19:41 (87,5 MB/s) - ‘guacamole-auth-sso-1.5.5.tar.gz’ saved [38286364/38286364]

zafar@apache-guaca:~# ls
apacheguac.conf  apacheguac.csr  dead.letter                    guac_notify.sh
apacheguac.crt   apacheguac.key  guacamole-auth-sso-1.5.5.tar.gz
zafar@apache-guaca:~# tar -xvzf guacamole-auth-sso-1.5.5.tar.gz
```

```
zafar@apache-guaca:~# sudo cp guacamole-auth-sso-1.5.5/openid/guacamole-auth-sso-openid-1.5.5.jar /etc
/guacamole/extensions/
zafar@apache-guaca:~# ls
apacheguac.conf  apacheguac.csr  dead.letter                guacamole-auth-sso-1.5.5.tar.gz
apacheguac.crt   apacheguac.key  guacamole-auth-sso-1.5.5  guac_notify.sh
zafar@apache-guaca:~# ls /etc/guacamole/extensions/
guacamole-auth-jdbc-mysql-1.5.5.jar  guacamole-history-recording-storage-1.5.5.jar
guacamole-auth-sso-openid-1.5.5.jar
zafar@apache-guaca:~#
```

```
zafar@apache-guaca:~# sudo chown zafar:zafar /etc/guacamole/extensions/guacamole-auth-sso-openid-1.5.5
.jar
zafar@apache-guaca:~# sudo chmod 644 /etc/guacamole/extensions/guacamole-auth-sso-openid-1.5.5.jar
zafar@apache-guaca:~#
zafar@apache-guaca:~# sudo systemctl restart guacd
zafar@apache-guaca:~# sudo systemctl restart tomcat9
```

```
#declaration de de la connexion a Mariadb
#ce fichier est utile aussi pour d'autre parametres

# MySQL -------------------------
#mysql-hostname: 127.0.0.1
#mysql-port: 3306
#mysql-database: guacadb
#mysql-username: userdb
#mysql-password: zafar
#-------------------------------


history-recording-enabled: true
history-recording-storage-dir: /var/lib/guacamole/recordings

auth-provider: net.sourceforge.guacamole.net.auth.openid.OpenIDAuthenticationProvider
openid-issuer: http://10.10.10.8
openid-authentication-uri: http://10.10.10.8/validate/check
openid-client-id: admin
openid-client-secret: admin
openid-redirect-uri: https://10.10.10.4/guacamole/
openid-scope: openid email profile
openid-username-claim-type: sub
openid-realm: guacamole_realm
openid-authorization-endpoint: http://10.10.10.8/validate/check
openid-userinfo-endpoint: http://10.10.10.8/validate/check
openid-authorization-endpoint: https://10.10.10.8/validate/check
openid-userinfo-endpoint: http://10.10.10.8/validate/check?user={USERNAME}
openid-login-form: true
```

Compte Rendu – Configuration de l'authentification OpenID avec PrivacyIDEA

Date : 04/02/2025

Dans le cadre de mon stage, j'ai entrepris la mise en place d'une authentification OpenID avec PrivacyIDEA pour Guacamole. Après configuration, la redirection depuis Guacamole vers PrivacyIDEA fonctionne correctement. Cependant, un problème persiste : PrivacyIDEA ne reçoit pas correctement l'utilisateur et retourne l'erreur `ERR905`, empêchant l'authentification finale.

L'une des exigences était **l'utilisation obligatoire du serveur SMTP et des adresses e-mail de l'entreprise** pour l'envoi des OTP. Cette contrainte a complexifié la configuration et nécessité plus de temps pour la recherche et l'adaptation du système.

Étant dans ma dernière semaine de stage, je ne peux pas poursuivre cette tâche, car d'autres priorités restent à traiter, notamment :

- **Mettre Guacamole sur Internet**
- **Configurer le NAT et le Proxy**

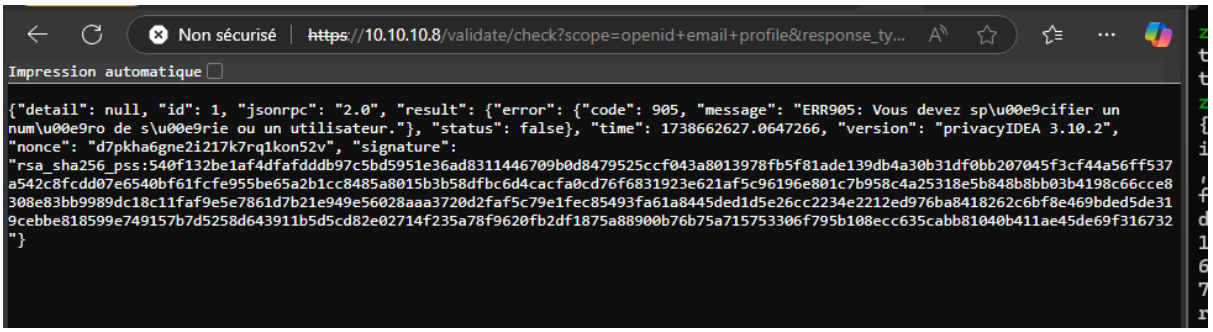
Comme cette implémentation d'OpenID ne faisait pas partie du cahier des charges initial, je vais proposer une alternative plus simple et mieux adaptée : **l'authentification TOTP** recommandée par Guacamole.

Bilan des apprentissages

Malgré les difficultés rencontrées, cette configuration m'a permis d'acquérir des connaissances approfondies sur les différentes méthodes d'authentification et leur intégration, notamment :

- **Les protocoles d'authentification OpenID, TOTP et LDAP**
- **L'intégration de PrivacyIDEA avec Guacamole**
- **Les contraintes liées à l'authentification en entreprise (SMTP, sécurité, gestion des identités)**

Cette expérience m'a permis de mieux comprendre les défis de l'authentification avancée et la gestion des accès en entreprise.



Sources :
[3. First Steps — privacyIDEA 3.10dev1 documentation]

[privacyidea/doc/installation/ubuntu.rst at master · privacyidea/privacyidea · GitHub]

Authentification SSO via SAML entre Guacamole et Keycloak

Cette solution de sécurisation de Guacamole VPN est très fiable, que ce soit pour un usage en local ou lorsqu'on expose Guacamole Daudruy sur Internet. Keycloak gère les utilisateurs et leur authentification, offrant ainsi une gestion centralisée et sécurisée des accès.

1. Préparer l'environnement

- Serveur Keycloak : sur Ubuntu

Téléchargement Keycloak :

Dernier version

```
zafar@auth-srv:/opt$ sudo wget https://github.com/keycloak/keycloak/releases/download/26.1.0/keycloak-26.1.0.tar.gz
--2025-01-31 09:09:40--  https://github.com/keycloak/keycloak/releases/download/26.1.0/keycloak-26.1.0.tar.gz
Resolving github.com (github.com)... 140.82.121.3
Connecting to github.com (github.com)|140.82.121.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/11125589/ff752aee-0a36-473d-9168-
7fa9355643c6?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250131%2Fus-east-1%2Fs3%2Faws4
_request&X-Amz-Date=20250131T090931Z&X-Amz-Expires=300&X-Amz-Signature=ac5346f53a8f90dd419b7cf3e1b8701be73a221e61aada89a
98dd053fede2b57&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dkeycloak-26.1.0.tar.gz&
response-content-type=application%2Foctet-stream [following]
--2025-01-31 09:09:40--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/11125589/ff752aee-
0a36-473d-9168-7fa9355643c6?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250131%2Fus-eas
t-1%2Fs3%2Faws4_request&X-Amz-Date=20250131T090931Z&X-Amz-Expires=300&X-Amz-Signature=ac5346f53a8f90dd419b7cf3e1b8701be7
3a221e61aada89a98dd053fede2b57&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dkeycloak
-26.1.0.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.109.133, 185.199.110.133, 185.199.108
.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 148227819 (141M) [application/octet-stream]
Saving to: ‘keycloak-26.1.0.tar.gz’

keycloak-26.1.0.tar.gz    100%[===================================>] 141,36M  93,2MB/s    in 1,5s
```

Installation de java :

Keycloak est basé sur Quarkus, qui fonctionne sur la JVM (Java Virtual Machine). Il a besoin de Java 17 ou 21 pour s'exécuter

```
zafar@auth-srv:/opt$ sudo apt install -y openjdk-21-jre
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Le paquet suivant a été installé automatiquement et n'est plus nécessaire :
  libllvm17t64
```

décompression des fichier tar keycloack

```
zafar@auth-srv:/opt$ sudo tar -xzvf keycloak-26.1.0.tar.gz
keycloak-26.1.0/version.txt
keycloak-26.1.0/conf/cache-ispn.xml
keycloak-26.1.0/README.md
keycloak-26.1.0/themes/README.md
```

```
zafar@auth-srv:/opt$ ls
keycloak-26.1.0  keycloak-26.1.0.tar.gz
zafar@auth-srv:/opt$
```

Les dossier de configuration et exécution :

```
zafar@auth-srv:/opt/keycloak-26.1.0$ tree bin/
bin/
├── client
│   ├── keycloak-admin-cli-26.1.0.jar
│   └── lib
│       ├── bcprov-jdk18on-1.78.1.jar
│       ├── keycloak-crypto-default-26.1.0.jar
│       └── keycloak-crypto-fips1402-26.1.0.jar
├── federation-sssd-setup.sh
├── kcadm.bat
├── kcadm.sh
├── kc.bat
├── kcreg.bat
├── kcreg.sh
└── kc.sh

3 directories, 11 files
zafar@auth-srv:/opt/keycloak-26.1.0$
```

```
root@auth-srv:/opt/keycloak# cd keycloak-26.1.0/
root@auth-srv:/opt/keycloak/keycloak-26.1.0# ls
bin  conf  lib  LICENSE.txt  providers  README.md  themes  version.txt
root@auth-srv:/opt/keycloak/keycloak-26.1.0# cd bin/
root@auth-srv:/opt/keycloak/keycloak-26.1.0/bin# ls
client  federation-sssd-setup.sh  kcadm.bat  kcadm.sh  kc.bat  kcreg.bat  kcreg.sh  kc.sh
root@auth-srv:/opt/keycloak/keycloak-26.1.0/bin#
root@auth-srv:/opt/keycloak/keycloak-26.1.0/bin# ls -l ./kc.sh
-rwxr-xr-x 1 1001 118 6286 janv. 15 10:25 ./kc.sh
root@auth-srv:/opt/keycloak/keycloak-26.1.0/bin# chmod +x kc.sh
root@auth-srv:/opt/keycloak/keycloak-26.1.0/bin#
```

Maintenant qu' on installé il faut créer un user Admin pour se connecter à l' interface Keycloack :

Premiere solution :

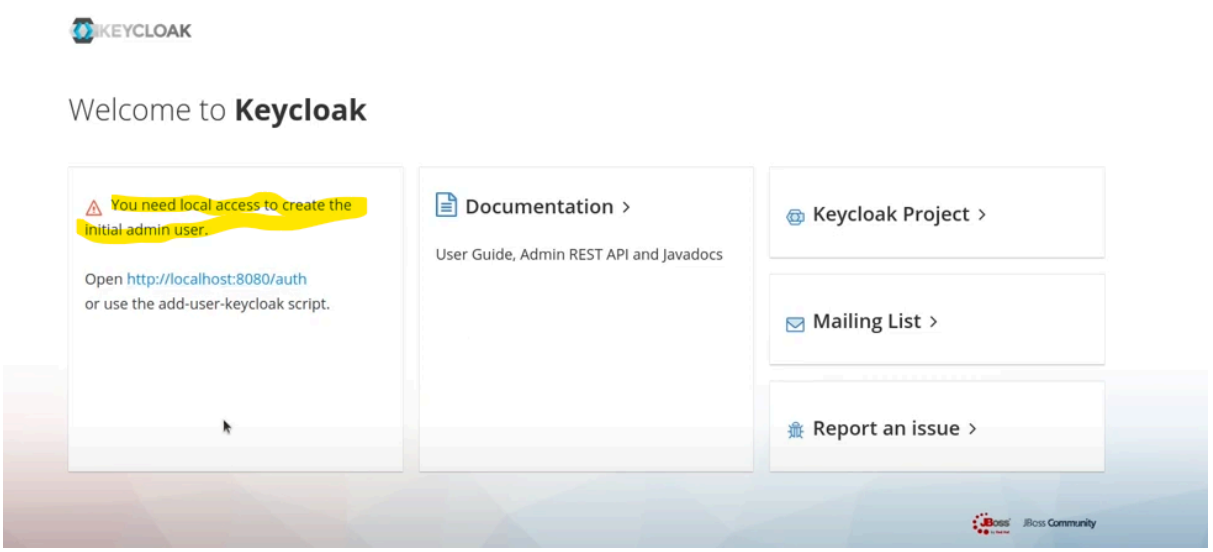
```
zafar@auth-srv:/opt/keycloak-26.1.0/bin$ export KEYCLOAK_ADMIN=admin
zafar@auth-srv:/opt/keycloak-26.1.0/bin$ export KEYCLOAK_ADMIN_PASSWORD=admin
```

Puis on lance le scripte de keycloack :

```
zafar@auth-srv:/opt/keycloak-26.1.0/bin$ sudo ./kc.sh start-dev
Updating the configuration and installing your custom providers, if any. Please wait.
2025-01-31 09:16:48,065 WARN  [io.qua.config] (build-9) Unrecognized configuration key "quarkus.smallrye-health.extensio
ns.enabled" was provided; it will be ignored; verify that the dependency extension for this configuration is set or that
 you did not make a typo
2025-01-31 09:16:49,629 INFO  [io.qua.hib.orm.dep.HibernateOrmProcessor] (build-13) Persistence unit 'keycloak-default':
 Enforcing Quarkus defaults for dialect 'org.hibernate.dialect.H2Dialect' by automatically setting 'jakarta.persistence.
database-product-version=2.3.230'.
2025-01-31 09:16:49,632 INFO  [io.qua.hib.orm.dep.HibernateOrmProcessor] (build-13) A legacy persistence.xml file is pre
```

Puis on tapes sur interface l' ip machine et le port :

http://10.10.10.x:80xx



Si cette méthode ne fonctionne pas on vas forcer Keycloack à configuier le user :

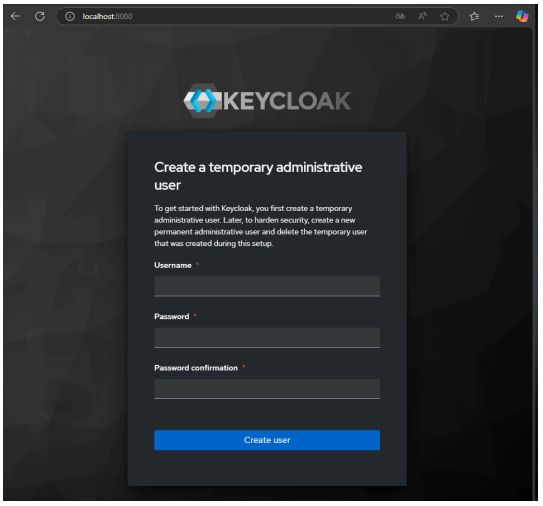
```
 GNU nano 7.2                          /opt/keycloak-26.1.0/conf/keycloak.conf
#metrics-enabled=true

# HTTP

# The file path to a server certificate or certificate chain in PEM format.
#https-certificate-file=${kc.home.dir}conf/server.crt.pem

# The file path to a private key in PEM format.
#https-certificate-key-file=${kc.home.dir}conf/server.key.pem

# The proxy address forwarding mode if the server is behind a reverse proxy.
#proxy=reencrypt

# Do not attach route to cookies and rely on the session affinity capabilities from reverse proxy
#spi-sticky-session-encoder-infinispan-should-attach-route=false

# Hostname for the Keycloak server.
#hostname=myhostname

http-enabled=true
http-host=0.0.0.0
hostname=10.10.10.8
http-port=8080
admin=admin
admin-password=admin
```

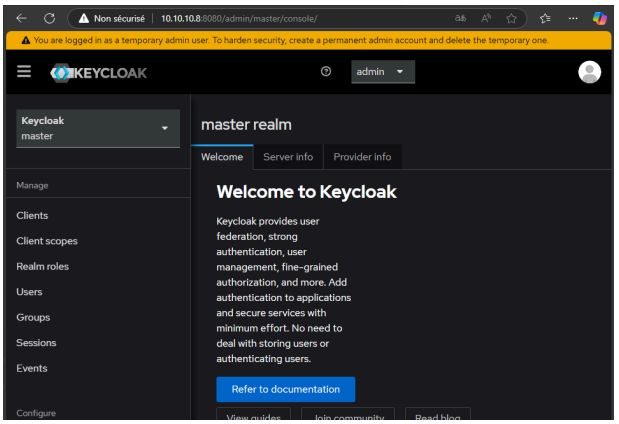
```
zafar@auth-srv:/opt/keycloak-26.1.0/bin$ sudo ./kc.sh build
WARNING: The following run time options were found, but will be ignored

Updating the configuration and installing your custom providers, if any
```

Puis on exécute à nouveau le script sur le serveur

```
zafar@auth-srv:/opt/keycloak-26.1.0$ sudo ./bin/kc.sh start-dev
Running the server in development mode. DO NOT use this configuration in production.
2025-01-31 10:06:29,469 WARN  [io.quarkus.config] (main) Unrecognized configuration key "quarkus.small
rye-health.extensions.enabled" was provided; it will be ignored; verify that the dependency extension
for this configuration is set or that you did not make a typo
2025-01-31 10:06:29,903 INFO  [org.keycloak.url.HostnameV2ProviderFactory] (main) If hostname is speci
fied, hostname-strict is effectively ignored
2025-01-31 10:06:31,763 INFO  [org.keycloak.quarkus.runtime.storage.infinispan.CacheManagerFactory] (T
hread-5) Starting Infinispan embedded cache manager
2025-01-31 10:06:31,845 INFO  [io.agroal.pool] (JPA Startup Thread) Datasource '<default>': Initial si
ze smaller than min. Connections will be created when necessary
2025-01-31 10:06:31,937 INFO  [org.infinispan.CONTAINER] (Thread-5) Virtual threads support enabled
2025-01-31 10:06:32,245 INFO  [org.infinispan.CONTAINER] (Thread-5) ISPN000556: Starting user marshall
```

On vérifie sur interface avec ip et port





Pour le moment le firewall est désactivé :

```
zafar@auth-srv:/opt/keycloak-26.1.0/bin$ sudo ufw status
Status: inactive
zafar@auth-srv:/opt/keycloak-26.1.0/bin$
```

Par défaut, on doit exécuter la commande suivante pour démarrer Keycloak à chaque fois que tu veux accéder à l'interface : /opt/keycloak-26.1.0/bin$ sudo ./kc.sh start-dev

Dans le cas ou on veut automatiser le démarrage

1 Créer un service systemd pour Keycloak

1. Ouvre un fichier de service :

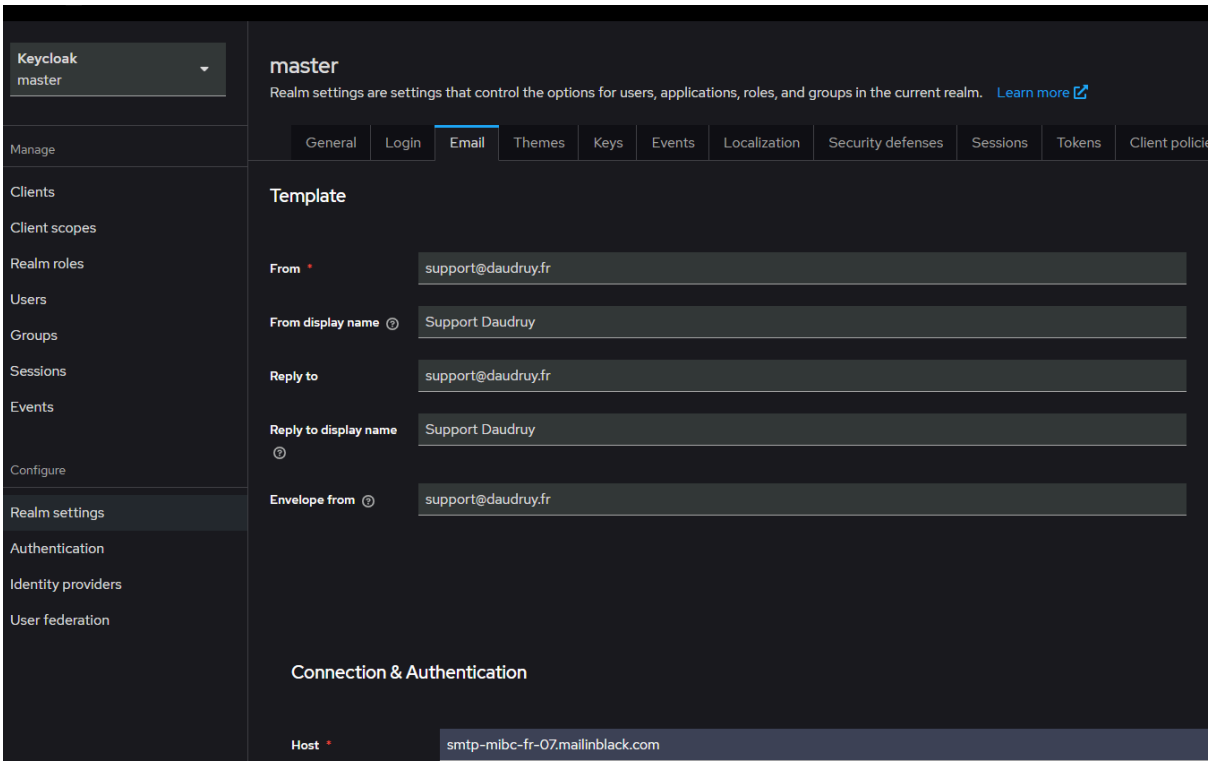
bash — Copier — Modifier

```
sudo nano /etc/systemd/system/keycloak.service
```

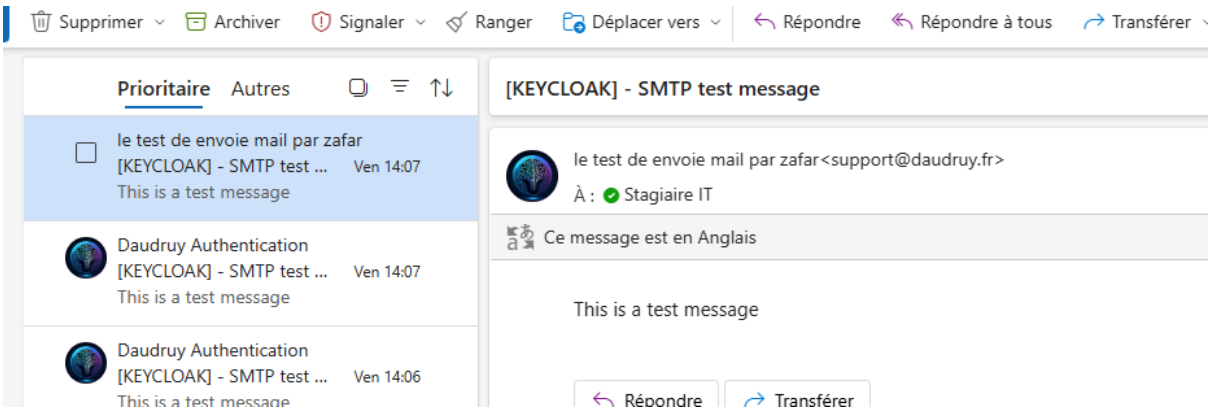
2. Ajoute cette configuration :

ini — Copier — Modifier

```
[Unit]
Description=Keycloak Server
After=network.target

[Service]
User=root
WorkingDirectory=/opt/keycloak
ExecStart=/opt/keycloak/bin/kc.sh start-dev
Restart=always
StandardOutput=journal
StandardError=journal
LimitNOFILE=1024

[Install]
WantedBy=multi-user.target
```

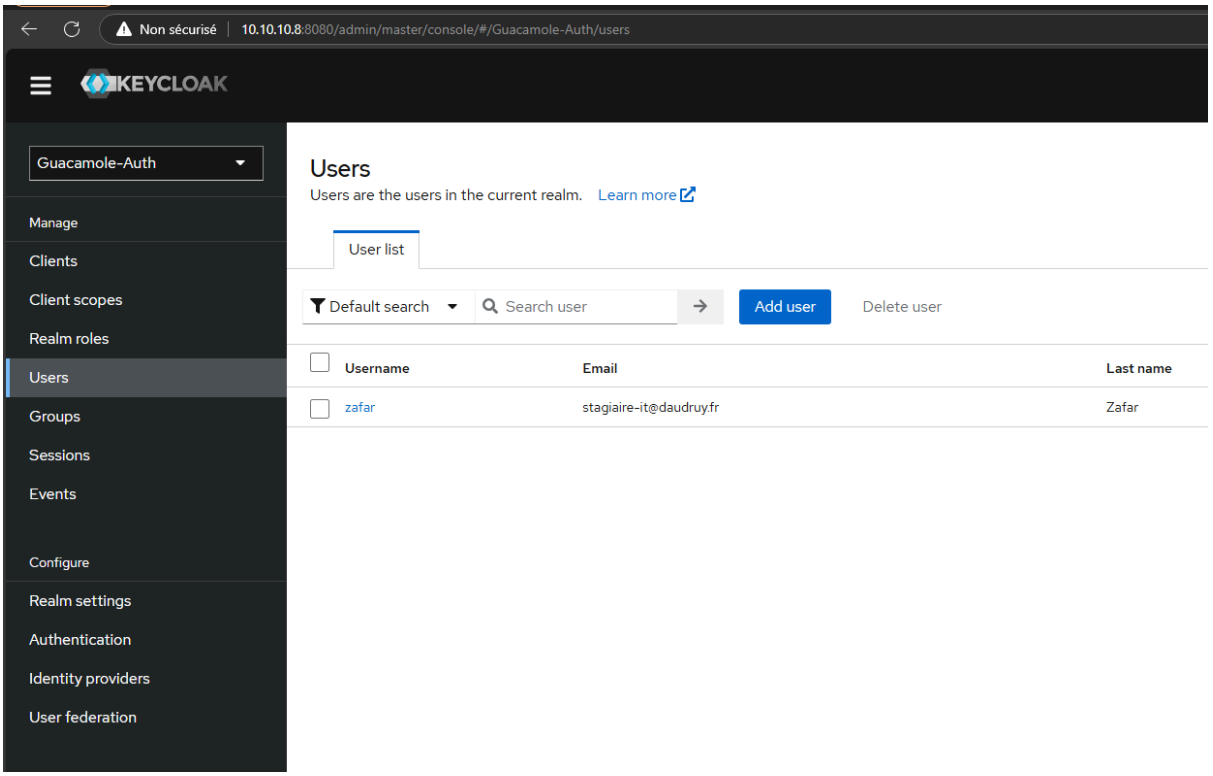


Test envoi mail :



Mode recovery si jamis on a un sousci de connexion

```
zafar@auth-srv:/opt/keycloak-26.1.0/bin$ cd /opt/keycloak-26.1.0/bin
zafar@auth-srv:/opt/keycloak-26.1.0/bin$ sudo ./kc.sh start-dev --spi-authenticator-required-action-verify-email-enabled=false
Updating the configuration and installing your custom providers, if any. Please wait.
2025-01-31 11:35:07,451 WARN  [io.qua.config] (build-40) Unrecognized configuration key "quarkus.smallrye-health.extensions.enabled" was provide
his configuration is set or that you did not make a typo
2025-01-31 11:35:09,089 INFO  [io.qua.hib.orm.dep.HibernateOrmProcessor] (build-19) Persistence unit 'keycloak-default': Enforcing Quarkus defau
y setting 'jakarta.persistence.database-product-version=2.3.230'.
2025-01-31 11:35:09,091 INFO  [io.qua.hib.orm.dep.HibernateOrmProcessor] (build-19) A legacy persistence.xml file is present in the classpath. T
units, and any configuration of the Hibernate ORM extension will be ignored. To ignore persistence.xml files instead, set the configuration prop
2025-01-31 11:35:12,892 INFO  [io.qua.dep.QuarkusAugmentor] (main) Quarkus augmentation completed in 6679ms
Running the server in development mode. DO NOT use this configuration in production.
```

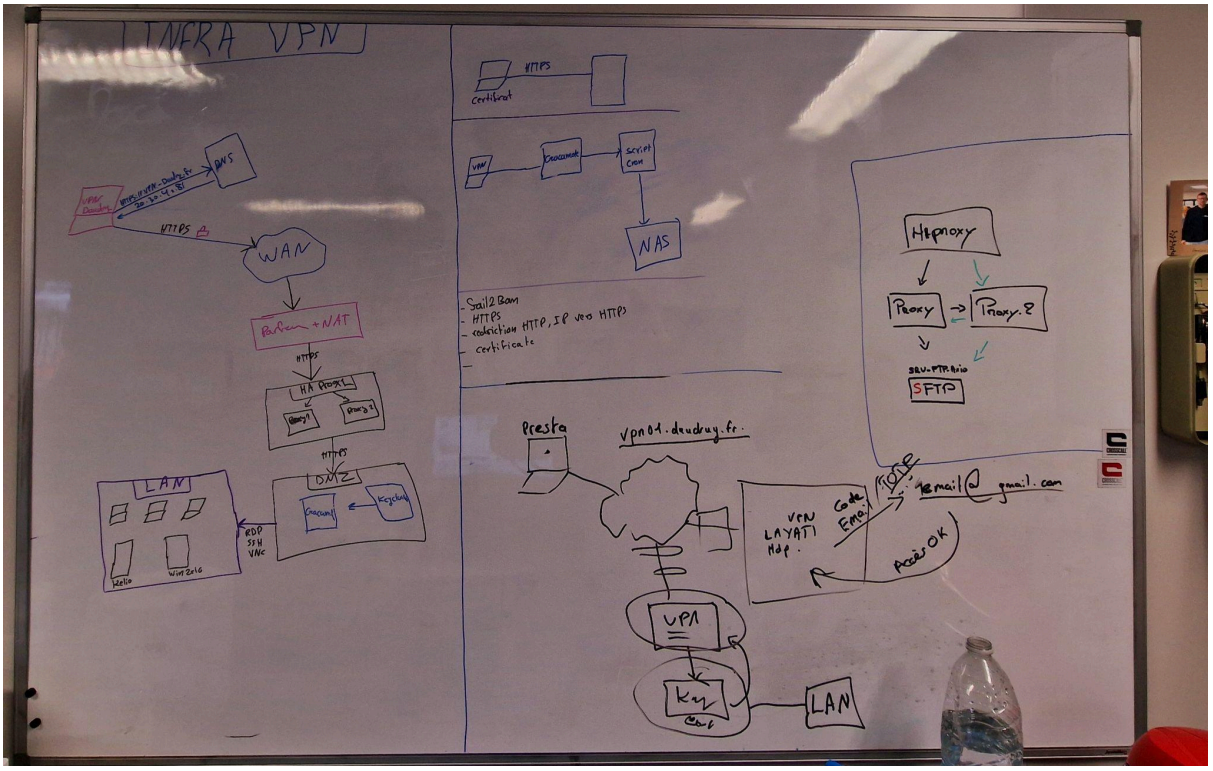


Compte rendu

Keycloak n’était pas une solution adaptée dans ce contexte, car il aurait nécessité une infrastructure plus lourde à mettre en place et à maintenir, avec une gestion des identités centralisée qui dépasse le cadre du besoin initial.

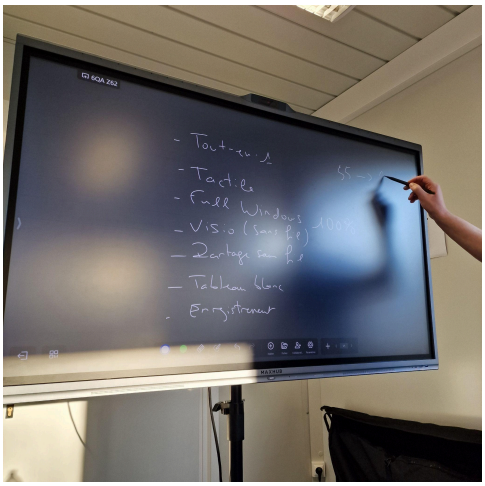
En effet, avec Keycloak, il faudrait créer un utilisateur à chaque fois sur Keycloak, puis créer manuellement le même utilisateur sur Guacamole, tout en effectuant des configurations supplémentaires. Cela ajouterait des tâches supplémentaires aux équipes IT, ce qui n’est pas souhaitable. L'équipe préfère une solution simple et facile à gérer, sans complexité inutile car ils ont l'habitude de sous-traiter la plupart de leurs services à des prestataires externes et, en cas de problème, ils créent des tickets pour obtenir une assistance.

Cependant, cette recherche, mise en place et configuration m’ont permis d’approfondir ma compréhension des protocoles d’authentification ainsi que de la configuration de Keycloak, en explorant son intégration avec d’autres services comme ldap , Microsoft authentication , authentification par AD etc et ses mécanismes de gestion des identités.



Autre tâche effectuée pendant mon stage :

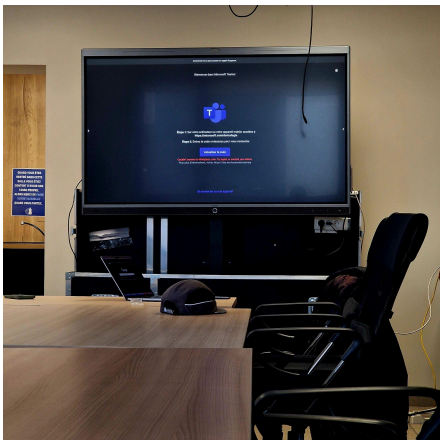
Assiste un une reunion de presentation d'écran tactile :



Faire l'inventaire des stock et pc :

Intégration des nouveau pc dans le AD existence :

Aide a la mis en place de nouveau ecran tactile :

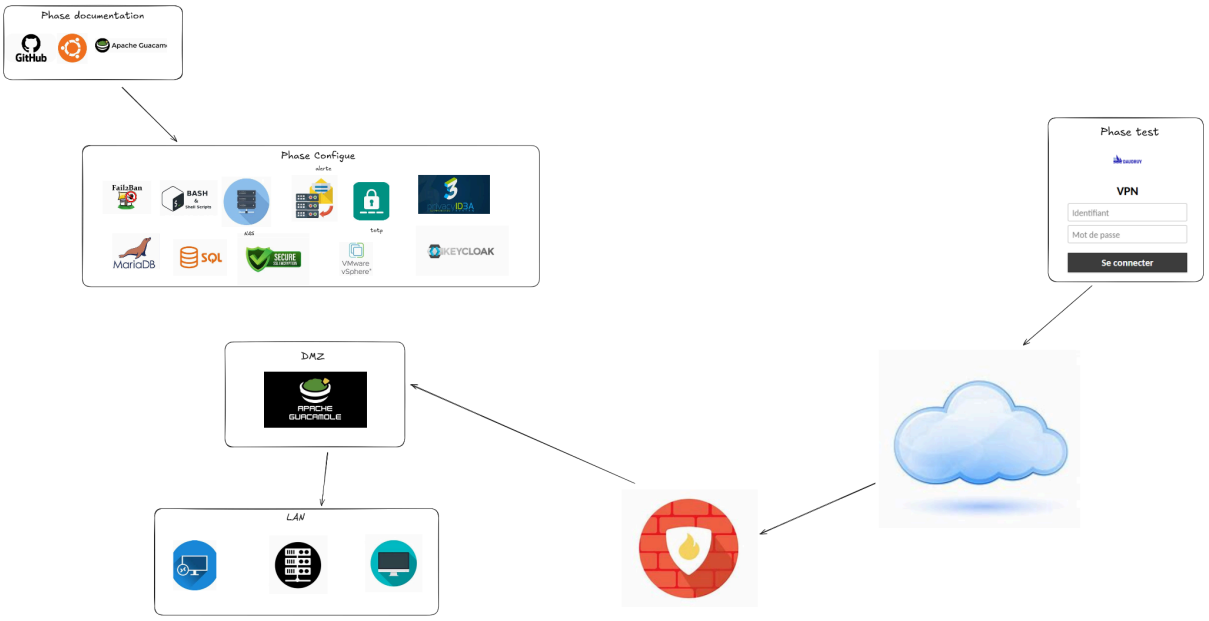


Changement des encore des impriment des defrenet bureau

Créer un schéma Visio de l'infrastructure réseau, représentant les différents switches présents dans les différents bureaux.

Guacamole :

1. Installation et configuration sur ubuntu
2. Création base de données
3. Configure win-serveur pour RDP
4. Creation des connexion rdp ssh
5. DNS
6. HTTPS
7. Certification autosigneé
8. TOTP
9. Conversion des enregist=rement m4v en mp4 enregistrement
10. Envoi enregistrement vers NAS
11. Script bash pour nettoyer le vedio enregistrement apres 10 jours
12. Script envoie alert de connexion ssh et rdp
13. Fail2ban
14. Nagios sur docker
15.
16.



Mission et projet réalisé

Dans le cadre de son stage, **[Nom du stagiaire]** a travaillé sur la **mise en place d'une solution d'accès distant sécurisée** pour les serveurs hébergés chez Daudruy. Ses missions ont inclus :

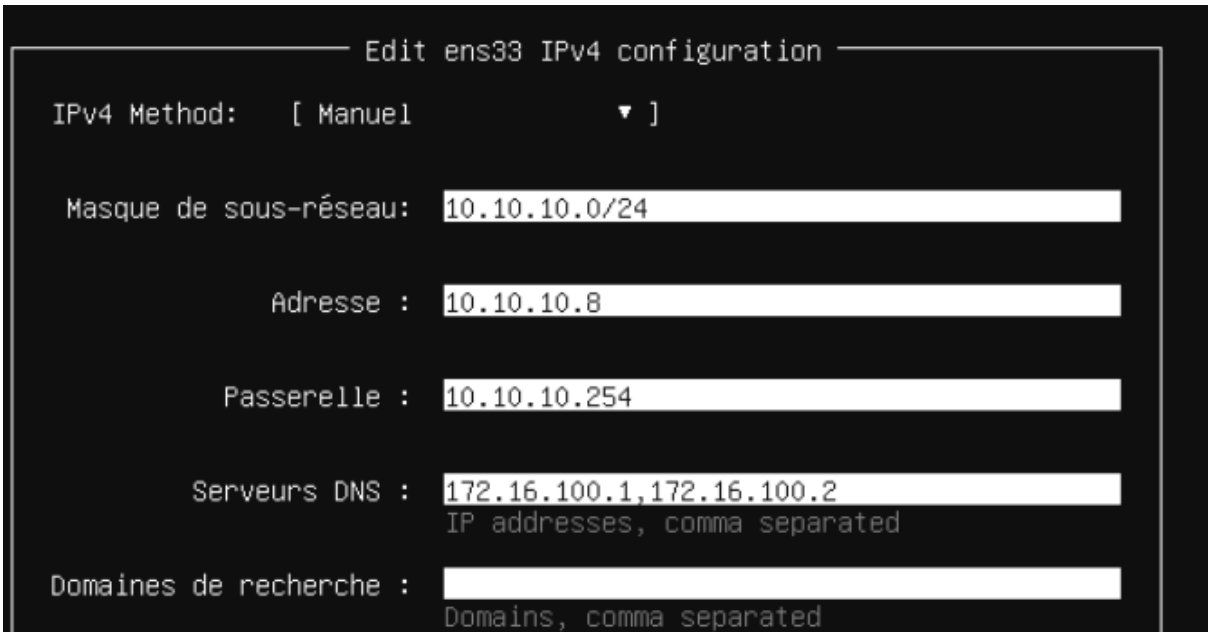
- **Installation et configuration d'Apache Guacamole** sur un serveur Ubuntu pour les accès RDP et SSH.
- **Sécurisation des accès** avec Fail2Ban, pare-feu et certificat SSL.
- **Automatisation des alertes de connexion**, avec un script envoyant des notifications en cas d'accès via Guacamole.
- **Mise en place de l'enregistrement des sessions utilisateurs**, transfert automatique des fichiers vers un NAS et conversion des vidéos via FFmpeg.
- **Rédaction de la documentation technique et formation** de l'équipe IT.

Compétences développées

- **Administration système** (gestion des serveurs Linux et Windows).
- **Sécurité réseau** (protection contre les attaques, gestion des accès).
- **Automatisation** (scripts Bash pour monitoring et gestion des accès).
- **Supervision et stockage** (gestion des enregistrements et transfert vers NAS).
- **Gestion de projet IT** (documentation, tests, présentation aux équipes).

Authentication Guacamole-VPN

Solution recommandée : **Utiliser OpenLDAP avec Guacamole**



Methode 1 créer guacamole extensions : pas marcher car c’est plus poussé et besoin d'un développeur

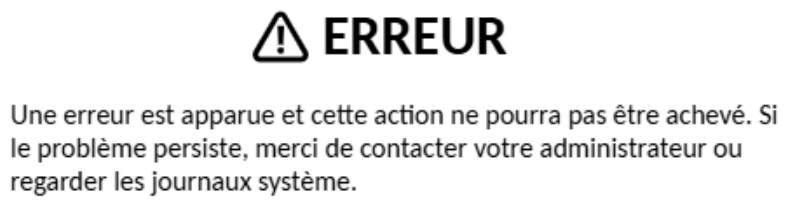
```
afar@apache-guaca:~# ls
pacheguac.conf  apacheguac.csr  dead.letter
pacheguac.crt   apacheguac.key  guacamole-auth-extension
afar@apache-guaca:~# cd guacamole-auth-extension/
afar@apache-guaca:~/guacamole-auth-extension# ls
om.xml  src
afar@apache-guaca:~/guacamole-auth-extension# cd src/
afar@apache-guaca:~/guacamole-auth-extension/src# ls
ain
afar@apache-guaca:~/guacamole-auth-extension/src# cd main/
afar@apache-guaca:~/guacamole-auth-extension/src/main# ls
ava  resources
afar@apache-guaca:~/guacamole-auth-extension/src/main# cd resources/
afar@apache-guaca:~/guacamole-auth-extension/src/main/resources# ls
afar@apache-guaca:~/guacamole-auth-extension/src/main/resources# nano guac-manifest.json
afar@apache-guaca:~/guacamole-auth-extension/src/main/resources# cd
afar@apache-guaca:~# cd guacamole-auth-extension/
afar@apache-guaca:~/guacamole-auth-extension# ls
om.xml  src
afar@apache-guaca:~/guacamole-auth-extension# mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -------------< org.apache.guacamole:guacamole-auth-custom >-------------
[INFO] Building guacamole-auth-custom 1.5.5
[INFO] --------------------------------[ jar ]---------------------------------
ownloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-resource
-plugin/2.6/maven-resources-plugin-2.6.pom
ownloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-resources
```

```
ng-2.1.jar (208 kB at 3.6 MB/s)
[INFO] Building jar: /home/zafar/guacamole-auth-extension/target/guacamole-auth-custom-1.5.5.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  8.061 s
[INFO] Finished at: 2025-01-29T08:12:39Z
[INFO] ------------------------------------------------------------------------
zafar@apache-guaca:~/guacamole-auth-extension# cp target/guacamole-auth-custom-1.5.5.jar /etc/guacamol
```

```
zafar@apache-guaca:~/guacamole-auth-extension# tree
.
├── pom.xml
├── src
│   └── main
│       ├── java
│       │   └── org
│       │       └── apache
│       │           └── guacamole
│       │               └── auth
│       │                   └── TutorialAuthenticationProvider.java
│       └── resources
│           └── guac-manifest.json
└── target
    ├── classes
    │   ├── guac-manifest.json
    │   └── org
    │       └── apache
    │           └── guacamole
    │               └── auth
    │                   └── TutorialAuthenticationProvider.class
    ├── generated-sources
    │   └── annotations
    ├── guacamole-auth-custom-1.5.5.jar
    ├── maven-archiver
    │   └── pom.properties
    └── maven-status
        └── maven-compiler-plugin
            └── compile
                └── default-compile
                    ├── createdFiles.lst
                    └── inputFiles.lst

21 directories, 9 files
```

Sources

[Custom authentication — Apache Guacamole Manual v1.5.5]



Cette méthode ne répond pas à ma demande d'authentification, car elle est trop complexe. En effet, il est difficile pour les équipes informatiques de gérer manuellement les utilisateurs en modifiant les fichiers à chaque ajout ou modification. Créer chaque utilisateur à la main dans les fichiers de configuration devient rapidement ingérable et peu évolutif, surtout quand le nombre d'utilisateurs augmente.

Cette rechache et mis en place m’a appris de comment on part de une besoin puis rechache puis mis en place test etc j’aimm beaucoup je vais me speciale en cyber et infrastra

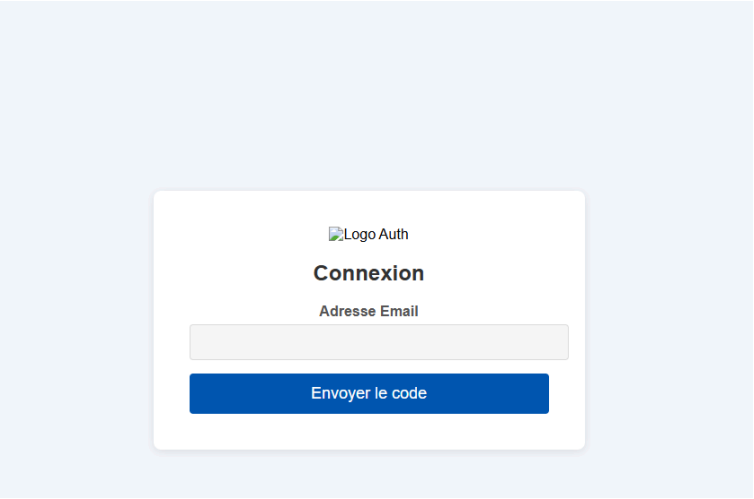
Methode 2

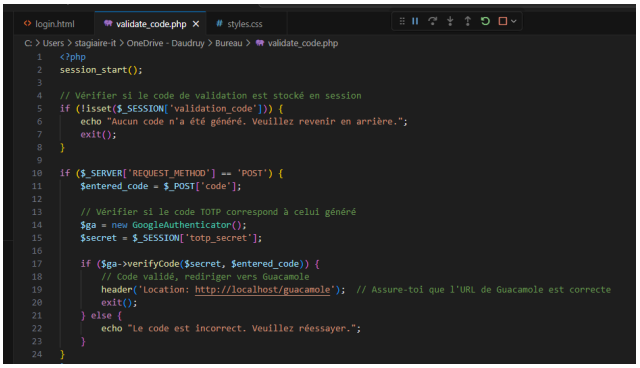
Architecture proposée :

1. **Utilisateur sur Guacamole :** L'utilisateur entre ses identifiants dans l'interface de connexion de **Guacamole**.
2. **Redirection vers un serveur Apache d'authentification externe :** Après la saisie des identifiants, **Guacamole** redirige l'utilisateur vers un serveur Apache dédié à l'authentification.
3. **Authentification par email :** Sur ce serveur, l'utilisateur saisit un code envoyé par email (authentification à deux facteurs).
4. **Redirection vers Guacamole :** Une fois le code validé, le serveur Apache redirige l'utilisateur vers **Guacamole** pour accéder à ses sessions.

En effet, je crée un serveur **Apache séparé** qui gère l'authentification à deux facteurs et redirige l'utilisateur vers **Guacamole** une fois le code validé.

```
zafar@srvauth:~$ sudo apt install apache2
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
```

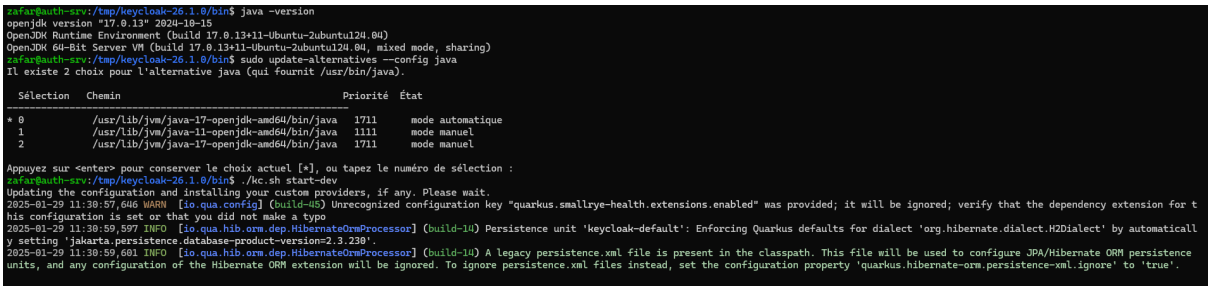




Cette solution ets tres pousé c’est
fessable avec html css et php et redrictaion des page mais cela risque de prendre le temps et je suis eau 4em semain de stage donc je tente un autre solution de ath

Methode 3

Installer Keycloak : je sui avance masi il y certain endroit ou il faut avoir ds compétence en développement



Methode 4

mettre en place l'authentification SSO via SAML entre Guacamole et Keycloak

---

1. Préparer l'environnement

- **Serveur Guacamole : Assure-toi que Guacamole est installé et fonctionne correctement.**
- **Serveur Keycloak : Assure-toi que Keycloak est installé et accessible.**

```
root@auth-srv:~# cd /tmp/
root@auth-srv:/tmp# ls
hsperfdata_root
hsperfdata_zafar
keycloak-26.1.0.tar.gz
```

```
root@auth-srv:/tmp# sudo mv /tmp/keycloak-26.1.0 /opt/keycloak
root@auth-srv:/tmp#
```

```
root@auth-srv:/opt/keycloak# cd keycloak-26.1.0/
root@auth-srv:/opt/keycloak/keycloak-26.1.0# ls
bin  conf  lib  LICENSE.txt  providers  README.md  themes  version.txt
root@auth-srv:/opt/keycloak/keycloak-26.1.0# cd bin/
root@auth-srv:/opt/keycloak/keycloak-26.1.0/bin# ls
client  federation-sssd-setup.sh  kcadm.bat  kcadm.sh  kc.bat  kcreg.bat  kcreg.sh  kc.sh
root@auth-srv:/opt/keycloak/keycloak-26.1.0/bin#
root@auth-srv:/opt/keycloak/keycloak-26.1.0/bin# ls -l ./kc.sh
-rwxr-xr-x 1 1001 118 6286 janv. 15 10:25 ./kc.sh
root@auth-srv:/opt/keycloak/keycloak-26.1.0/bin# chmod +x kc.sh
root@auth-srv:/opt/keycloak/keycloak-26.1.0/bin#
```

```
root@auth-srv:/opt/keycloak/keycloak-26.1.0/bin# ./kc.sh start-dev
Updating the configuration and installing your custom providers, if any. Please wait.
2025-01-30 09:11:07,416 WARN  [io.qua.config] (build-17) Unrecognized configuration key "quarkus.small
rye-health.extensions.enabled" was provided; it will be ignored; verify that the dependency extension
for this configuration is set or that you did not make a typo
2025-01-30 09:11:09,300 INFO  [io.qua.hib.orm.dep.HibernateOrmProcessor] (build-4) Persistence unit 'k
eycloak-default': Enforcing Quarkus defaults for dialect 'org.hibernate.dialect.H2Dialect' by automati
cally setting 'jakarta.persistence.database-product-version=2.3.230'.
2025-01-30 09:11:09,303 INFO  [io.qua.hib.orm.dep.HibernateOrmProcessor] (build-4) A legacy persistenc
e.xml file is present in the classpath. This file will be used to configure JPA/Hibernate ORM persiste
nce units, and any configuration of the Hibernate ORM extension will be ignored. To ignore persistence
.xml files instead, set the configuration property 'quarkus.hibernate-orm.persistence-xml.ignore' to '
true'.
```

✅ 1. Modifier Keycloak pour autoriser les connexions distantes

Ouvre le fichier **Keycloak.conf** :

bash — Copier — Modifier

```
sudo nano /opt/keycloak-26.1.0/conf/keycloak.conf
```

Ajoute ces lignes à la fin :

makefile — Copier — Modifier

```
http-enabled=true
http-host=0.0.0.0
hostname=10.10.10.8
http-port=8080
admin=admin
admin-password=admin
```

💡 **Explication :**

- `http-enabled=true` → Active HTTP (désactive l'obligation d'HTTPS).
- `http-host=0.0.0.0` → Accepte les connexions depuis toutes les IPs.
- `hostname=10.10.10.8` → Définit l'IP du serveur comme l'host.
- `http-port=8080` → Définit le port d'écoute.
- `admin=admin` → Force la création d'un compte admin.
- `admin-password=admin` → Définit le mot (↓)asse.

✅ 2. Contourner la restriction avec SSH (si accès distant impossible)

Si Keycloak bloque encore l'accès **depuis une autre machine**, utilise **un tunnel SSH** depuis Windows :

1️⃣ **Ouvre une invite de commande sur Windows et entre :**

powershell — Copier — Modifier

```
ssh -L 8080:localhost:8080 zafar@10.10.10.8
```

2️⃣ **Sur ton navigateur Windows**, ouvre :

arduino — Copier — Modifier

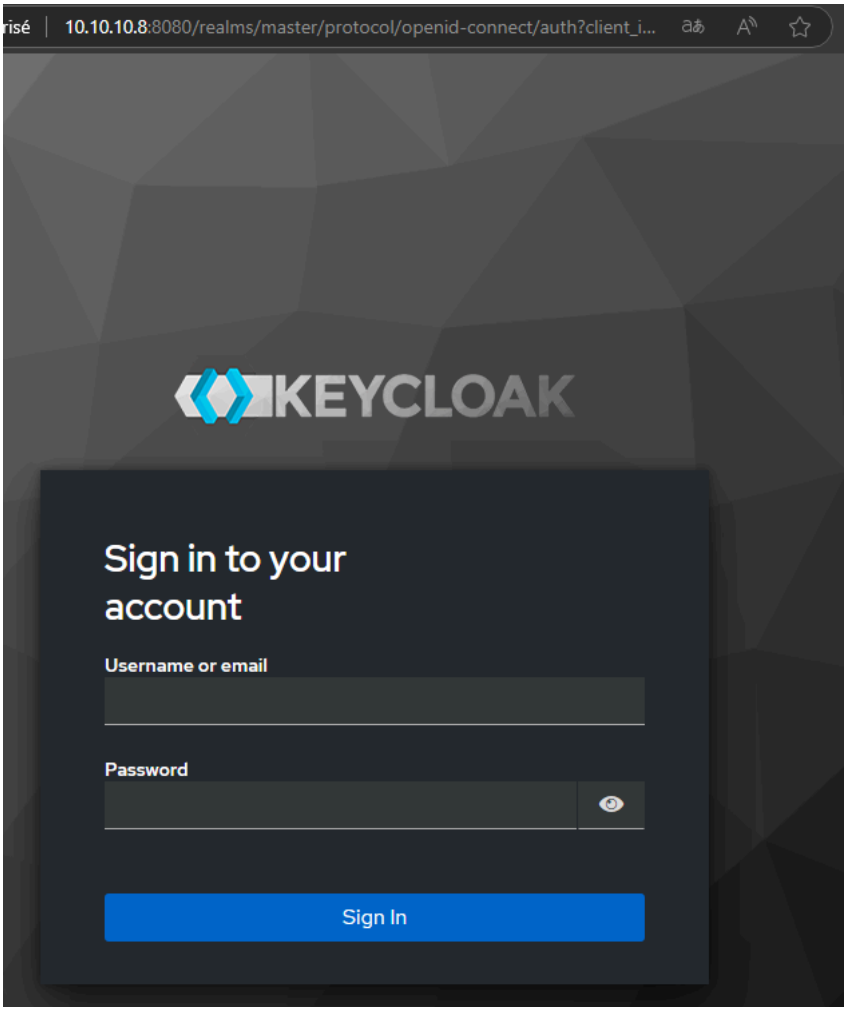
```
http://localhost:8080
```

👉 Tu seras **redirigé vers Keycloak sur le serveur distant.**

On lance notre serveur :

```
zafar@auth-srv:/opt/keycloak-26.1.0/bin$ sudo ./kc.sh start-dev
Running the server in development mode. DO NOT use this configuration in production.
2025-01-31 09:38:56,411 WARN  [io.quarkus.config] (main) Unrecognized configuration key "quarkus.small
rye-health.extensions.enabled" was provided; it will be ignored; verify that the dependency extension
for this configuration is set or that you did not make a typo
2025-01-31 09:38:56,827 INFO  [org.keycloak.url.HostnameV2ProviderFactory] (main) If hostname is speci
fied, hostname-strict is effectively ignored
2025-01-31 09:38:58,730 INFO  [org.keycloak.quarkus.runtime.storage.infinispan.CacheManagerFactory] (T
hread-5) Starting Infinispan embedded cache manager
2025-01-31 09:38:58,789 INFO  [io.agroal.pool] (JPA Startup Thread) Datasource '<default>': Initial si
ze smaller than min. Connections will be created when necessary
2025-01-31 09:38:58,884 INFO  [org.infinispan.CONTAINER] (Thread-5) Virtual threads support enabled
```

Test la Connexion à Distance



Pour le moment le firewall est désactivé :

```
zafar@auth-srv:/opt/keycloak-26.1.0/bin$ sudo ufw status
Status: inactive
zafar@auth-srv:/opt/keycloak-26.1.0/bin$
```

Résumé

1️⃣ **Modifier `keycloak.conf` pour autoriser l'accès distant.**
2️⃣ **Exécuter `kc.sh build` puis redémarrer Keycloak.**
3️⃣ **Utiliser un tunnel SSH si Windows ne peut pas accéder à Keycloak.**

Comparaison des méthodes d'authentification

Méthode	Avantages	Inconvénients	Exemples d'utilisation
OAuth2/OpenID Connect	SSO, intégration facile avec de nombreux services	Nécessite un serveur d'identité (ex. Keycloak)	Intégration avec Google, Facebook, etc.
LDAP	Intégration avec Active Directory, gestion centralisée des utilisateurs	Nécessite un serveur LDAP/Active Directory	Entreprises utilisant LDAP/Active Directory
JWT (JSON Web Token)	Authentification stateless, sécurisé, aucune session côté serveur	Gestion des jetons et des clés secrètes	Applications modernes, API sécurisées
SMS (2FA)	Sécurisé, utilisé largement	Coût des messages SMS, dépend d'une passerelle SMS	Authentification via SMS (Twilio, Nexmo)
Google Authenticator/Authy	Sécurisé, ne dépend pas d'une connexion internet	L'utilisateur doit installer une application	Utilisé avec 2FA, Authentification mobile

Methode 5

```
zafar@auth-srv:~$ apt-cache search jre
default-jre - environnement d'exécution Java standard ou compatible
default-jre-headless - environnement d'exécution standard Java ou compatible - non graphique
openjdk-17-jre - Environnement d'exécution Java OpenJDK qui utilise Hotspot JIT
openjdk-17-jre-headless - environnement d'exécution Java OpenJDK utilisant Hotspot JIT (sans affichage
)
openjdk-21-jre - Environnement d'exécution Java OpenJDK qui utilise Hotspot JIT
openjdk-21-jre-headless - environnement d'exécution Java OpenJDK utilisant Hotspot JIT (sans affichage
)
android-platform-tools-base - outils de base pour développer des applications pour le système Android
docbook-xsl - feuilles de style de conversion de fichiers DocBook XML vers différents formats
```

```
zafar@auth-srv:~$ sudo apt install -y openjdk-21-jre
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
```

```
zafar@auth-srv:/opt$ sudo wget https://github.com/keycloak/keycloak/releases/download/26.1.0/keycloak-
26.1.0.tar.gz
--2025-01-31 08:08:31--  https://github.com/keycloak/keycloak/releases/download/26.1.0/keycloak-26.1.0
.tar.gz
Resolving github.com (github.com)... 140.82.121.4
Connecting to github.com (github.com)|140.82.121.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/11125589/ff752a
ee-0a36-473d-9168-7fa9355643c6?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproductio
n%2F20250131%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250131T080831Z&X-Amz-Expires=300&X-Amz-Signa
ture=e2b32e3a69c86b391491392afe42c26620b128bb01fb1546681ad306e38dde5a&X-Amz-SignedHeaders=host&respons
e-content-disposition=attachment%3B%20filename%3Dkeycloak-26.1.0.tar.gz&response-content-type=applicat
ion%2Foctet-stream [following]
--2025-01-31 08:08:32--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/
11125589/ff752aee-0a36-473d-9168-7fa9355643c6?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releas
eassetproduction%2F20250131%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250131T080831Z&X-Amz-Expires=
300&X-Amz-Signature=e2b32e3a69c86b391491392afe42c26620b128bb01fb1546681ad306e38dde5a&X-Amz-SignedHeade
rs=host&response-content-disposition=attachment%3B%20filename%3Dkeycloak-26.1.0.tar.gz&response-conten
t-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.108.133, 185.199.11
1.133, 185.199.109.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.108.133|:443... co
nnected.
HTTP request sent, awaiting response... 200 OK
Length: 148227819 (141M) [application/octet-stream]
Saving to: ‘keycloak-26.1.0.tar.gz’

keycloak-26.1.0.tar.gz    100%[===================================>] 141,36M  74,5MB/s    in 1,9s

2025-01-31 08:08:34 (74,5 MB/s) - ‘keycloak-26.1.0.tar.gz’ saved [148227819/148227819]
```

```
zafar@auth-srv:/opt$ sudo tar xzvf keycloak-26.1.0.tar.gz
keycloak-26.1.0/version.txt
keycloak-26.1.0/conf/cache-ispn.xml
keycloak-26.1.0/README.md
```

```
zafar@auth-srv:/opt$ ls
keycloak-26.1.0  keycloak-26.1.0.tar.gz
zafar@auth-srv:/opt$
```

Line symbolique ou cas ou

```
zafar@auth-srv:/opt$ sudo ln -s keycloak-26.1.0/ keycloak
zafar@auth-srv:/opt$
```

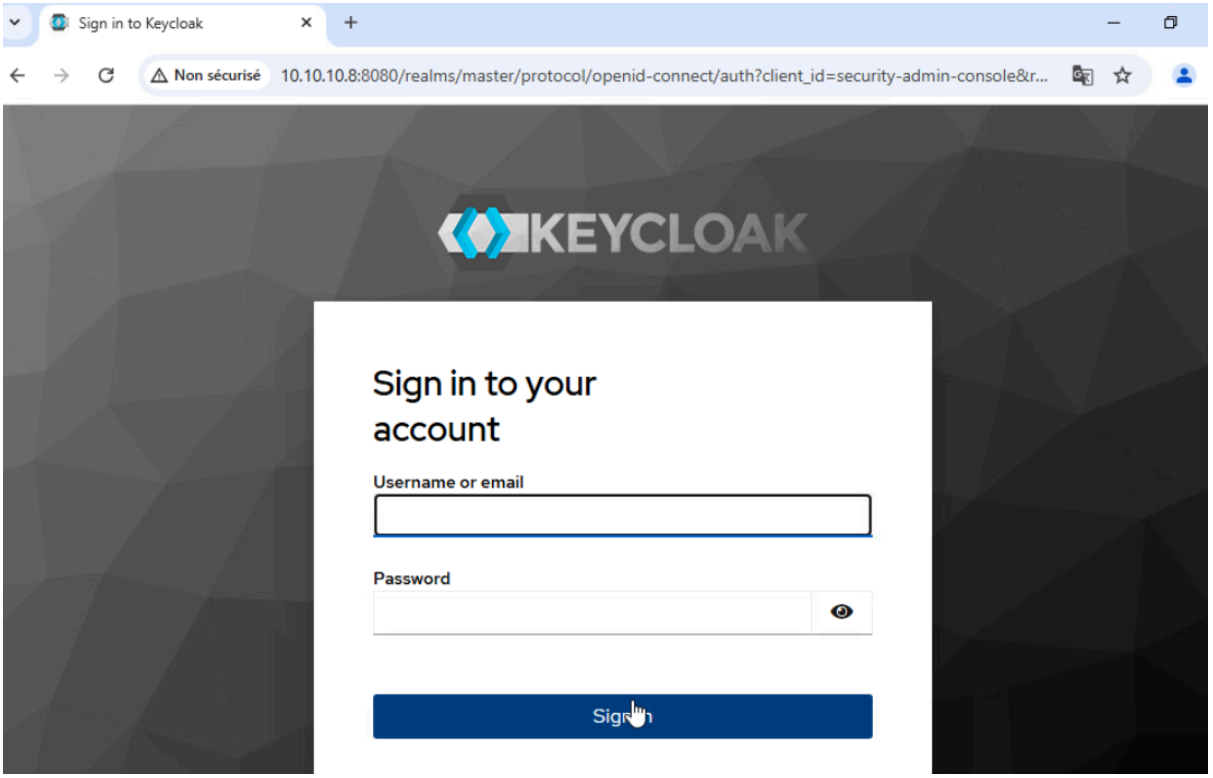
```
zafar@auth-srv:/opt$ cd keycloak
zafar@auth-srv:/opt/keycloak$ ls
bin  conf  lib  LICENSE.txt  providers  README.md  themes  version.txt
zafar@auth-srv:/opt/keycloak$ ls
bin  conf  lib  LICENSE.txt  providers  README.md  themes  version.txt
zafar@auth-srv:/opt/keycloak$ ls bin/
client  federation-sssd-setup.sh  kcadm.bat  kcadm.sh  kc.bat  kcreg.bat  kcreg.sh  kc.sh
zafar@auth-srv:/opt/keycloak$
```

Ajoute des utilisateur :

```
 GNU nano 7.2                          /opt/keycloak-26.1.0/conf/keycloak.conf
#metrics-enabled=true

# HTTP

# The file path to a server certificate or certificate chain in PEM format.
#https-certificate-file=${kc.home.dir}conf/server.crt.pem

# The file path to a private key in PEM format.
#https-certificate-key-file=${kc.home.dir}conf/server.key.pem

# The proxy address forwarding mode if the server is behind a reverse proxy.
#proxy=reencrypt

# Do not attach route to cookies and rely on the session affinity capabilities from reverse proxy
#spi-sticky-session-encoder-infinispan-should-attach-route=false

# Hostname for the Keycloak server.
#hostname=myhostname

http-enabled=true
http-host=0.0.0.0
hostname=10.10.10.8
http-port=8080
admin=admin
admin-password=admin
```

On execute build :

```
zafar@auth-srv:/opt/keycloak-26.1.0/bin$ sudo ./kc.sh build
WARNING: The following run time options were found, but will be ignored during build time: kc.hostname
, kc.http-enabled, kc.http-host, kc.http-port

Updating the configuration and installing your custom providers, if any. Please wait.
2025-01-31 09:27:45,987 WARN  [io.qua.config] (build-5) Unrecognized configuration key "quarkus.smallr
ye-health.extensions.enabled" was provided; it will be ignored; verify that the dependency extension f
```


Sign in to Keycloak
Non sécurisé
10.10.10.8:8080/realms/master/protocol/openid-connect/auth?client_id=security-admin-console&r...
KEYCLOAK
Sign in to your account
Username or email
Password
Sign

Étapes pour mettre en place l'authentification SSO via SAML entre Guacamole et Keycloak

---

1. Préparer l'environnement

- **Serveur Guacamole : Assure-toi que Guacamole est installé et fonctionne correctement.**
- **Serveur Keycloak : Assure-toi que Keycloak est installé et accessible.**

---

2. Installer l'extension SSO pour Guacamole

a. Télécharger l'extension SSO/SAML pour Guacamole :

- **Télécharge le fichier `guacamole-auth-sso-saml-1.4.0.jar` à partir du dépôt Apache ou de la source officielle.**

b. Installer l'extension dans Guacamole :

- **Copie le fichier JAR dans le répertoire extensions de Guacamole, généralement situé dans `/etc/guacamole/extensions/`.**

bash

Copier

```
sudo cp guacamole-auth-sso-saml-1.4.0.jar /etc/guacamole/extensions/
```

- **Assure-toi que le serveur Guacamole est arrêté avant d'ajouter l'extension.**

c. Redémarrer Guacamole :

bash

Copier

```
sudo systemctl restart guacamole
```

---

3. Configurer Keycloak comme fournisseur d'identité (IdP)

a. Créer un Realm dans Keycloak :

- **Connecte-toi à l'interface d'administration de Keycloak.**
- **Crée un Realm spécifique pour Guacamole (par exemple, `Guacamole-Realm`).**

b. Créer un client dans Keycloak pour Guacamole :

- **Dans Keycloak, va dans Clients et crée un client pour Guacamole.**
- **Le type de client doit être OpenID Connect ou SAML.**

Exemple de configuration pour Keycloak :

- **Client ID : `guacamole-client`**
- **Client Secret : Génére un secret pour ce client.**
- **Root URL : L'URL de ton serveur Guacamole (par exemple `http://guacamole-server/guacamole`).**

c. Configurer les paramètres SAML dans Keycloak :

- **Active le protocole SAML pour ce client.**
- **Assure-toi que les métadonnées SAML (URL d'IdP, entité IdP, etc.) sont bien définies.**
- **Obtiens l'URL du SSO (Single Sign-On) pour intégrer dans Guacamole.**

---

4. Configurer Guacamole pour utiliser l'authentification SSO via SAML

a. Modifier le fichier `guacamole.properties` :

Ouvre le fichier `guacamole.properties` et ajoute les configurations pour utiliser SAML comme mécanisme d'authentification.

Exemple de configuration pour Guacamole dans `guacamole.properties` :

ini

Copier

```
# Activer l'authentification SSO via SAML
```

```
auth-provider:
net.sourceforge.guacamole.auth.sso.SSOAuthenticationProvider


# URL du fournisseur d'identité SAML (Keycloak)

sso-saml-idp-url:
https://<keycloak-server>/realms/Guacamole-Realm/protocol/saml


# ID de l'entité du fournisseur SAML

sso-saml-idp-entity-id: Guacamole-Realm


# Configuration du client SAML dans Keycloak

sso-saml-client-id: guacamole-client


# Secret du client SAML dans Keycloak

sso-saml-client-secret: <client-secret>
```

b. Redémarrer Guacamole pour appliquer les changements :

bash

Copier

```
sudo systemctl restart guacamole
```

---

5. Tester l'authentification SSO

1. **Accéder à Guacamole : Ouvre un navigateur et accède à Guacamole via l'URL configurée (par exemple `http://guacamole-server/guacamole`).**
2. **Redirection vers Keycloak : L'utilisateur sera automatiquement redirigé vers Keycloak pour s'authentifier (s'il n'est pas déjà authentifié).**
    - **L'utilisateur saisit ses identifiants dans Keycloak.**
3. **Authentification réussie : Une fois l'utilisateur authentifié, Keycloak génère une assertion SAML et redirige l'utilisateur vers Guacamole avec une session active.**
4. **Accès à Guacamole : Une fois l'utilisateur authentifié, il peut accéder aux différentes connexions distantes (RDP, SSH, etc.) configurées dans Guacamole sans avoir à saisir de nouveau ses identifiants.**

---

6. Vérification des logs et du bon fonctionnement

a. Vérifier les logs de Guacamole :

Si l'authentification échoue, vérifie les logs de Guacamole pour détecter toute erreur :

bash

Copier

```
sudo tail -f /var/log/guacamole/guacamole.log
```

b. Vérifier les logs de Keycloak :

Si l'authentification échoue côté Keycloak, vérifie les logs de Keycloak pour toute erreur dans l'authentification SSO.

Résumé des étapes

1. **Installer Guacamole et l'extension SSO via SAML.**
2. **Configurer Keycloak comme fournisseur d'identité SAML.**
3. **Configurer Guacamole pour utiliser Keycloak via SAML pour l'authentification SSO.**
4. **Tester l'intégration et vérifier que l'utilisateur est redirigé vers Keycloak pour se connecter et est ensuite redirigé vers Guacamole avec une session active.**

Amini zafar 07/01/2025

Compte rendu : Configuration de deux switches

Cisco série 1200

1. Configuration de base :

- Initialisation et configuration des deux switches Cisco.
- Attribution des noms d'hôtes pour distinguer les switches dans le réseau.

---

2. Création des VLANs :

- **VLAN 170** créé avec le nom **170_MGMT_RZO**.
- Association du port **GigabitEthernet 10** en mode **access** au VLAN 170.

---

3. Configuration IP et routage :

- Configuration des adresses IP de management pour chaque switch :
  - **Switch 1** : IP **172.16.170.4/24**, Gateway **172.16.170.x**.
  - **Switch 2** : IP **172.16.170.5/24**, Gateway **172.16.170.x**.
- Mise en place d'une route par défaut avec : ip route 0.0.0.0/0 172.16.170.x

---

4. Préparation pour l'installation en baie :

- Installation des kits de rack pour les deux switches.
- Vérification des connexions réseau et alimentation avant mise en place en baie.

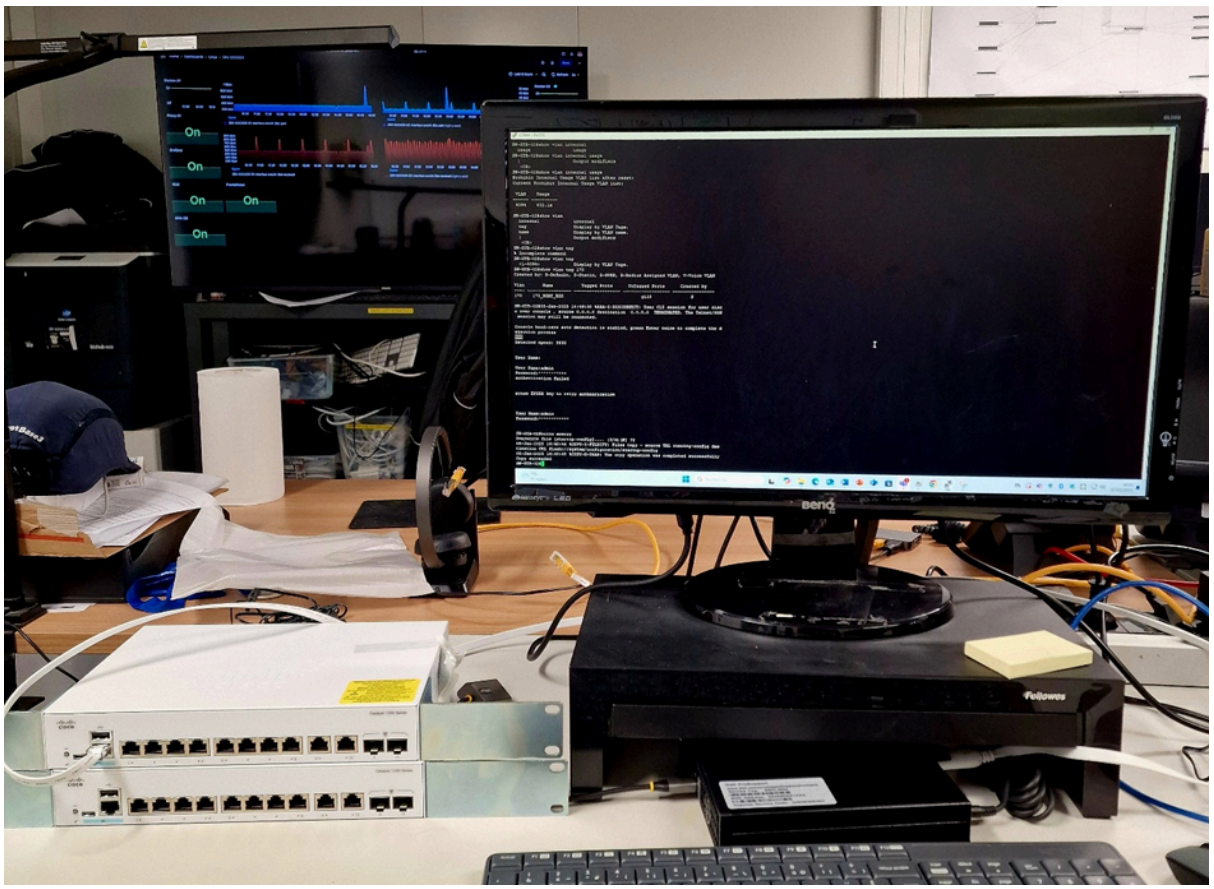
Conclusion : Les deux switches sont configurés conformément aux besoins de l'infrastructure, avec des VLANs, des IP de management, et un routage fonctionnel. Ils sont prêts pour être montés en baie et intégrés au réseau existant. 😊

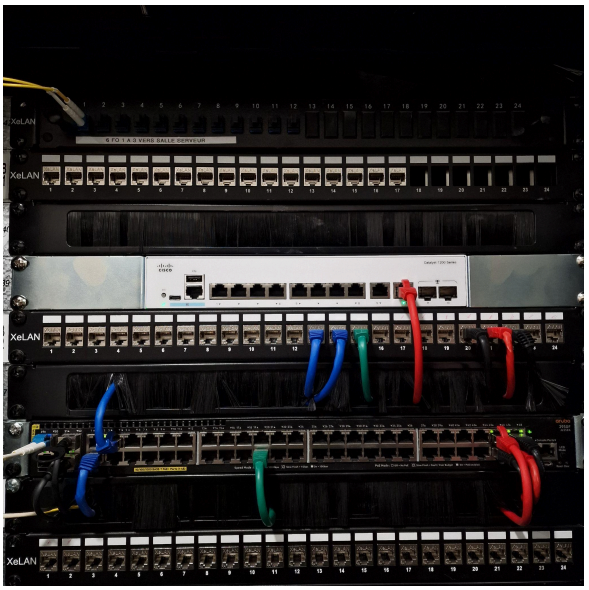
```
interface vlan 170
 name 170_MGMT_RZO
 ip address [redacted] 255.255.255.0
!
interface GigabitEthernet10
 switchport access vlan 170
!
exit
ip default-gateway [redacted]
SW-GTB-02#
```

```
Enter new username: admin
Enter new password: ************
Confirm new password: ************
Username and password were successfully updated.
switch45dblf#configure terminal
switch45dblf(config)#hostname SW-GTB-02
SW-GTB-02(config)#interface vlan 170
SW-GTB-02(config-if)#exit
SW-GTB-02(config)#vlan 170
SW-GTB-02(config)#interface vlan 170
SW-GTB-02(config-if)#name 170_MGMT_RZO
SW-GTB-02(config-if)#ip address [redacted]
SW-GTB-02(config-if)#no shutdown
SW-GTB-02(config-if)#exi
SW-GTB-02(config)#ip default-gateway [redacted]
SW-GTB-02(config)#interface gigabitethernet 10
SW-GTB-02(config-if)#switchport mode access
SW-GTB-02(config-if)#switchport access vlan 170
SW-GTB-02(config-if)#no shutdown
```

```
SW-GTB-02#show ip route static
Maximum Parallel Paths: 1 (1 after reset)
IP Forwarding: enabled

Codes: A - active, I - inactive

I  0.0.0.0/0 [1/4] via 172.16.[redacted]

SW-GTB-02#
```

```
SW-GTB-02#show vlan
Created by: D-Default, S-Static, G-GVRP, R-Radius Assigned VLAN, V-Voice VLAN

Vlan       Name            Tagged Ports      UnTagged Ports      Created by
---- ---------------- ----------------- ------------------ ----------------
 1          1                                gi1-9,Po1-4            D
170    170_MGMT_RZO                             gi10                S
```


On
On
On
On
On
BenQ
Fellowes


XeLAN
XeLAN
XeLAN
XeLAN


XeLAN
XeLAN
XeLAN
XeLAN

Fichiers principaux de Guacamole

1.1. `guacamole.properties`

- **Emplacement :** `/etc/guacamole/guacamole.properties`
- **Rôle :** Fichier de configuration principal de Guacamole.
    - Définit les paramètres globaux comme :
        - Connexion à la base de données (MariaDB/MySQL).
        - Chemin vers les extensions.
        - Paramètres réseau.

```
zafar@apache-guaca:~# cd /etc/guacamole/
zafar@apache-guaca:/etc/guacamole# ls
extensions  guacamole.properties  guacd.conf  lib
zafar@apache-guaca:/etc/guacamole#
```

1.2. Extensions (Fichiers `.jar`)

- **Emplacement :** `/etc/guacamole/extensions/`
- **Rôle :** Fournir des fonctionnalités supplémentaires comme :
    - Authentification via MySQL, LDAP ou autre.
    - Enregistrement des sessions.
- Exemples :
    - `guacamole-auth-jdbc-mysql-1.5.5.jar` : Extension pour l'authentification via MySQL/MariaDB.
    - `guacamole-history-recording-storage.jar` : Extension pour l'enregistrement des connexions.
    - Ou bien autre extension que on peut télécharger et déployer sur serveur

```
zafar@apache-guaca:/etc/guacamole/extensions# ls
guacamole-auth-jdbc-mysql-1.5.5.jar  guacamole-history-recording-storage-1.5.5.jar
zafar@apache-guaca:/etc/guacamole/extensions#
```

Fichiers de schéma SQL

- **Emplacement :** Décompressé dans un dossier temporaire, comme `/tmp/guacamole-auth-jdbc-1.5.5/mysql/schema/`.
- **Rôle :** Contient les commandes SQL pour créer les tables nécessaires à la base de données.
    - Exemple : `001-create-schema.sql`.

```
zafar@apache-guaca:/tmp/guacamole-auth-jdbc-1.5.5/mysql/schema# ls
001-create-schema.sql  002-create-admin-user.sql  upgrade
zafar@apache-guaca:/tmp/guacamole-auth-jdbc-1.5.5/mysql/schema#
```

Fichiers de log

- **Emplacement :** `/var/log/tomcat9/`
- **Rôle :** Enregistrer les activités, erreurs et autres informations pour diagnostiquer les problèmes.

```
zafar@apache-guaca:/var/log/tomcat9# ls
catalina.2025-01-17.log.gz  catalina.out.1              localhost_access_log.2025-01-17.txt.gz
catalina.2025-01-20.log     localhost.2025-01-17.log.gz  localhost_access_log.2025-01-20.txt
catalina.out                localhost.2025-01-20.log     localhost_access_log.2025-01-22.txt
```

Répertoires associés à Apache Tomcat

Fichier WAR

- **Emplacement :** `/var/lib/tomcat9/webapps/guacamole.war`
- **Rôle :** Contient l'application Guacamole prête à être déployée par Tomcat.

```
zafar@apache-guaca:/var/log/tomcat9# cd /var/lib/tomcat9/webapps
zafar@apache-guaca:/var/lib/tomcat9/webapps# ls
guacamole  guacamole.war  ROOT
zafar@apache-guaca:/var/lib/tomcat9/webapps#
```

Répertoire de déploiement ultra sécurisé

- **Emplacement :** `/var/lib/tomcat9/webapps/guacamole/`
- **Rôle :** Dossier généré lorsque le fichier `.war` est déployé. Contient tous les fichiers décompressés nécessaires à l'exécution de Guacamole.

```
zafar@apache-guaca:/var/lib/tomcat9/webapps# su -
Password:
root@apache-guaca:~# cd /var/lib/tomcat9/webapps/guacamole/
root@apache-guaca:/var/lib/tomcat9/webapps/guacamole# ls
1.guacamole.6f7b293d2dba5a891aa5.css      Blob.js                              images          npm-dependencies.txt
1.guacamole.6f7b293d2dba5a891aa5.css.map  datalist-polyfill.min.js             index.html      templates.js
1.guacamole.7935cf403412cd79c600.js       fonts                                jquery.min.js   translations
1.guacamole.7935cf403412cd79c600.js.map   guacamole.4baaa9df5aed3303a40f.js    layouts         verifyCachedVersion.js
angular.min.js                            guacamole.4baaa9df5aed3303a40f.js.map lodash.min.js  WEB-INF
app                                       guacamole-common-js                  META-INF
root@apache-guaca:/var/lib/tomcat9/webapps/guacamole#
```

Répertoires Guacd (Guacamole proxy daemon)

3.1. Fichier de configuration Guacd

- **Emplacement :** `/etc/guacamole/guacd.conf`
- **Rôle :** Configure Guacd (daemon de Guacamole) :
    - Hôte et port d'écoute (par défaut : `0.0.0.0:4822`).

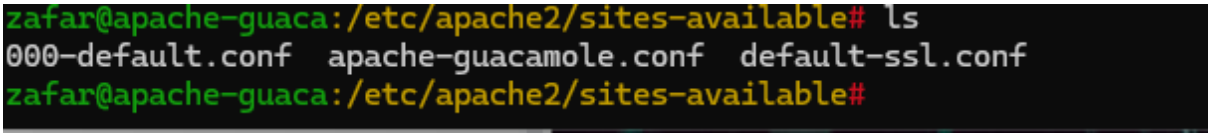
```
zafar@apache-guaca:/var/lib/tomcat9/webapps/guacamole# cd /etc/guacamole/
zafar@apache-guaca:/etc/guacamole# ls
extensions  guacamole.properties  guacd.conf  lib
zafar@apache-guaca:/etc/guacamole# nano guacamole.properties
zafar@apache-guaca:/etc/guacamole#
```

```
 GNU nano 6.2                                    guacamole.properties
#declaration de de la connexion a Mariadb
#ce fichier est utile aussi pour d'autre parametres

# MySQL ------------------------
mysql-hostname: 127.0.0.1
mysql-port: 3306
mysql-database: guacadb
mysql-username: userdb
mysql-password: zafar
#-------------------------------
```

Fichiers Apache (reverse proxy utilisé)

4.1. Configuration du proxy

- **Emplacement :** `/etc/apache2/sites-available/000-default.conf` (Apache)
- **Rôle :** Permet de configurer le proxy pour rediriger le trafic vers Tomcat sur le port 8080.



Emplacement des fichiers SSL

Clé privée (private key)

- **Chemin par défaut :** `/etc/ssl/private/`
- **Exemple :**
  - `/etc/ssl/private/guacamole.key`
- **Rôle :**
  - Utilisée pour déchiffrer les connexions entrantes.
  - Doit être gardée secrète avec des permissions strictes

Certificat public (public certificate)

- **Chemin par défaut :** `/etc/ssl/certs/`
- **Exemple :**
  - `/etc/ssl/certs/guacamole.crt`
- **Rôle :**
  - Utilisé pour authentifier le serveur auprès des clients.

2. Configuration du serveur web

Fichier Apache pour HTTPS

- **Chemin par défaut :** `/etc/apache2/sites-available/`
- **Exemple :**
  - `/etc/apache2/sites-available/guacamole-ssl.conf`
- **Rôle :**
  - Définit les paramètres SSL, y compris les chemins vers les fichiers de certificats et clés.
  -

Configuration Apache globale

- **Chemin :**
  - `/etc/apache2/apache2.conf`
  - `/etc/apache2/mods-available/ssl.conf`
- **Rôle :**
  - Paramètres globaux SSL (comme les protocoles TLS autorisés).

Certificat intermédiaire	`/etc/ssl/certs/chain.crt`	Relie le certificat à une autorité racine.
Certificat public	`/etc/ssl/certs/guacamole.crt`	Authentifie le serveur.
Certificats racines système	`/usr/local/share/ca-certificates/`	Certificats de confiance.
Clé privée	`/etc/ssl/private/guacamole.key`	Décrypte les connexions entrantes.
Configuration Apache SSL	`/etc/apache2/sites-available/`	Paramètres SSL et proxy.
Logs Apache	`/var/log/apache2/`	Vérifie les erreurs et activités.

Déploiement sur internet

Lors du déploiement d'Apache Guacamole sur Internet, plusieurs fichiers de configuration doivent être ajustés pour s'assurer que tout fonctionne correctement. Voici les fichiers essentiels à vérifier et à configurer pour une phase de déploiement :

---

1. Fichiers de configuration Apache Guacamole

a. `guacamole.properties`

- **Emplacement** : `/etc/guacamole/guacamole.properties`
- **Contenu clé à vérifier** :

Assure-toi que le backend MySQL et guacd sont correctement configurés :
properties
CopierModifier

```
mysql-hostname: 127.0.0.1
mysql-port: 3306
mysql-database: guacadb
mysql-username: guac_user
mysql-password: strongpassword
guacd-hostname: 127.0.0.1
guacd-port: 4822
```

  -
  - Ajoute un certificat SSL personnalisé pour sécuriser la connexion si ce n'est pas déjà fait.

---

2. Fichiers de configuration du proxy Apache2

Si tu utilises un proxy inversé avec **Apache2**, configure ces fichiers :

a. Configuration des sites disponibles (`/etc/apache2/sites-available/*.conf`)

Exemple de configuration pour un proxy SSL :
apache
CopierModifier

```
<VirtualHost *:443>
    ServerName guacamole.example.com

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/guacamole.crt
    SSLCertificateKeyFile /etc/ssl/private/guacamole.key
```

```
    ProxyPreserveHost On
    ProxyRequests Off
    ProxyPass / http://localhost:8080/guacamole/ retry=0 timeout=5
    ProxyPassReverse / http://localhost:8080/guacamole/
</VirtualHost>
```

-

Active les modules nécessaires :
bash
CopierModifier
```
sudo a2enmod ssl proxy proxy_http rewrite headers
```

-

b. Redirection HTTP vers HTTPS

- Fichier : `/etc/apache2/sites-available/000-default.conf`

Ajoute une redirection :
apache
CopierModifier
```
<VirtualHost *:80>
    ServerName guacamole.example.com
    Redirect permanent / https://guacamole.example.com/
</VirtualHost>
```

-

---

3. Fichiers de configuration de `guacd`

- **Emplacement** : `/etc/guacamole/guacd.conf`
- **À vérifier** :

Si guacd écoute bien sur **0.0.0.0** et le port correct (par défaut : 4822) :
properties
CopierModifier
```
[server]
bind_host = 0.0.0.0
bind_port = 4822
```

    -

---

4. Configuration réseau

a. Paramètres IP

- Assure-toi que l'adresse IP publique du serveur est bien configurée pour NAT ou DMZ (si nécessaire).
- Ajoute des règles NAT dans ton pare-feu pour rediriger les ports externes :
    - Port 443 (HTTPS) vers le serveur Apache.
    - Port 4822 vers guacd (si nécessaire).

b. Pare-feu

Ouvre les ports nécessaires :
bash
CopierModifier

```
sudo ufw allow 443
sudo ufw allow 4822
sudo ufw reload
```

-

---

5. Paramètres DNS

- Configure ton nom de domaine pour pointer vers l'adresse IP publique de ton serveur.

Exemple pour une entrée DNS A :
css
CopierModifier

```
guacamole.example.com. IN A 203.0.113.10
```

-

---

6. Tomcat9 (si nécessaire)

Fichier `server.xml`

- Emplacement : `/etc/tomcat9/server.xml`

Si tu veux que Tomcat serve directement HTTPS (peu recommandé, mais possible), configure un connecteur SSL :
xml
CopierModifier

```
<Connector port="8443"
protocol="org.apache.coyote.http11.Http11NioProtocol"
```

```
           maxThreads="200" SSLEnabled="true">
    <SSLHostConfig>
        <Certificate certificateKeyFile="/path/to/your/private.key"
                     certificateFile="/path/to/your/certificate.crt"
                     type="RSA" />
    </SSLHostConfig>
</Connector>
```

-

---

7. Autres ajustements à considérer

a. Sécurité

- Désactive les ports inutilisés sur le serveur pour éviter des failles.
- Installe et configure un certificat SSL valide (via Let's Encrypt ou un CA tiers) pour sécuriser l'accès.

b. Monitoring

- Configure des outils comme **fail2ban** pour limiter les tentatives de connexion non autorisées.
- Utilise les journaux de `/var/log/tomcat9/catalina.out` et `/var/log/syslog` pour surveiller les erreurs.

Mes script

```
#!/bin/bash

# Dossier des enregistrements Guacamole
recordings_dir="/var/lib/guacamole/recordings"
# Dossier du NAS monté
nas_dir="/mnt/nas/guacamole_recordings"

# Étape 1 : Transfert et conversion des fichiers
transfer_and_convert() {
    local dir="$1"
    local file="$2"
    # Extraire le nom du fichier
    filename=$(basename "$file")

    # Étape 1.1 : Conversion du fichier avec guacenc en .m4v
    echo "Étape 1.1 : Conversion du fichier $filename en .m4v"
    convert_with_guacenc "$file"

    # Vérifier si la conversion a réussi
    if [ $? -eq 0 ]; then
        echo "Fichier $filename converti en .m4v avec succès
============================."

        # Étape 1.2 : Transfert du fichier converti sur le NAS
        echo "==================== Étape 1.2 : Transfert du fichier $filename sur le
NAS==================="
        rsync -av "$file.m4v" "$nas_dir/"

        # Vérifier si le fichier a été transféré correctement
        if [ $? -eq 0 ]; then
            echo "Fichier $filename transféré avec succès vers le
NAS.============================"

            # Étape 1.3 : Suppression du fichier local après conversion et transfert
            echo "Étape 1.3 : Suppression du fichier local $file et fichier converti"
            rm -f "$file" "$file.m4v"
            if [ $? -eq 0 ]; then
                echo "Fichier local $filename supprimé avec succès."
            else
                echo "Erreur lors de la suppression du fichier local $filename."
            fi

            # Étape 1.4 : Vérification et suppression du répertoire si vide
            echo "Étape 1.4 : Vérification et suppression du répertoire $dir si vide"
            if [ ! "$(ls -A "$dir")" ]; then
                rmdir "$dir"
```

```
            echo "Répertoire $dir supprimé car il est vide."
        else
            echo "Répertoire $dir non vide, il n'a pas été supprimé."
        fi
    else
        echo "Erreur lors du transfert du fichier $filename vers le NAS."
    fi
  else
    echo "Erreur lors de la conversion du fichier $filename."
  fi
}

# Étape 2 : Conversion avec guacenc pour créer un fichier .m4v
convert_with_guacenc() {
  local input_file="$1"
  # Appeler guacenc pour convertir en .m4v (résolution 1280x720)
  sudo guacenc -s 1280x720 -f "$input_file"

  # Vérifier si la conversion s'est bien déroulée
  if [ $? -eq 0 ]; then
    echo "Conversion réussie avec guacenc : $input_file -> $input_file.m4v"
    return 0
  else
    echo "Erreur lors de la conversion avec guacenc pour le fichier $input_file"
    return 1
  fi
}

# Étape 3 : Vérification du répertoire des enregistrements
echo "Étape 3 : Vérification du répertoire des enregistrements..."
if [ -d "$recordings_dir" ]; then
  # Parcourir tous les sous-répertoires dans /recordings
  for dir in "$recordings_dir"/*; do
    if [ -d "$dir" ]; then
      echo "Répertoire trouvé : $dir"

      # Parcourir les fichiers à l'intérieur de chaque sous-répertoire
      for file in "$dir"/*; do
        if [ -f "$file" ]; then
          echo "Fichier trouvé : $file"
          # Appeler la fonction pour transférer ce fichier vers le NAS et le convertir
          transfer_and_convert "$dir" "$file"
        fi
      done
    fi
  done
else
  echo "Le répertoire des enregistrements Guacamole n'existe pas."
```

```
fi

# Fin du script, sans notification par email
echo "Le script Guacamole a été exécuté avec succès."
=====================================================================

#!/bin/bash

USER_NAME="$1"
CONNECTION_ID="$2"
DATE_CONNEXION=$(date '+%Y-%m-%d %H:%M:%S')

SUBJECT="Connexion Guacamole - Utilisateur: $USER_NAME"
BODY="Salut,\n\nL'utilisateur $USER_NAME s'est connecté à Guacamole.\nDate et heure:
$DATE_CONNEXION\nConnexion ID: $CONNECTION_ID\n\nÀ plus !"

echo -e "Subject: $SUBJECT\n\n$BODY" | /usr/bin/msmtp --file=/root/.msmtprc -a default
stagiaire-it@daudruy.fr

=================================================================


zafar@apache-guaca:/opt/scripts$ cat monitor_guacamole_ssh.py
import smtplib
import time
import re
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart

# Configuration SMTP
SMTP_SERVER = "smtp-mibc-fr-07.mailinblack.com"
SMTP_PORT = 25
MAIL_TO = "stagiaire-it@daudruy.fr"
MAIL_FROM = "stagiaire-it@daudruy.fr"  # À remplacer par un mail valide

# Fonction d'envoi d'alerte
def send_alert(subject, message):
    try:
        msg = MIMEMultipart()
        msg['From'] = MAIL_FROM
        msg['To'] = MAIL_TO
        msg['Subject'] = subject

        msg.attach(MIMEText(message, 'plain'))

        with smtplib.SMTP(SMTP_SERVER, SMTP_PORT) as server:
            server.sendmail(MAIL_FROM, MAIL_TO, msg.as_string())
        print("[+] Notification envoyée avec succès !")
```

```
    except Exception as e:
        print(f"[-] Erreur d'envoi de l'email: {e}")

# Surveillance des logs SSH et Guacamole
def monitor_logs():
    auth_log = "/var/log/auth.log"
    guac_log = "/var/log/tomcat9/catalina.out"

    with open(auth_log, "r") as ssh_log, open(guac_log, "r") as guac:
        ssh_log.seek(0, 2)
        guac.seek(0, 2)

        while True:
            ssh_line = ssh_log.readline()
            guac_line = guac.readline()

            # Détection des connexions SSH réussies
            if ssh_line and "Accepted password" in ssh_line:
                user = re.search(r'Accepted password for (\w+)', ssh_line)
                ip = re.search(r'from ([\d\.]+)', ssh_line)
                if user and ip:
                    msg = f"Connexion SSH détectée sur le serveur guacamole :\nUtilisateur :
{user.group(1)}\nIP : {ip.group(1)}"
                    send_alert("🔐 Alerte Connexion SSH sur le serveur Guacamole", msg)

            # Détection des échecs SSH
            if ssh_line and "Failed password" in ssh_line:
                ip = re.search(r'from ([\d\.]+)', ssh_line)
                if ip:
                    msg = f"Tentative de connexion SSH échouée depuis {ip.group(1)}"
                    send_alert("⚠️ Tentative SSH échouée", msg)

            # Détection des connexions Guacamole
            if guac_line and "User \" " in guac_line and "connected from" in guac_line:
                user = re.search(r'User \"(.*?)\"', guac_line)
                ip = re.search(r'from ([\d\.]+)', guac_line)
                if user and ip:
                    msg = f"Connexion Guacamole détectée en ssh  :\nUtilisateur :
{user.group(1)}\nIP : {ip.group(1)}"
                    send_alert("🖥️ Connexion Guacamole", msg)

            time.sleep(1)

if __name__ == "__main__":
    monitor_logs()
```

```
==============================================================
zafar@apache-guaca:/opt/scripts$ cat nas_supprime.sh
#!/bin/bash

# Répertoire cible
nas_dir="/mnt/nas/guacamole_recordings/"

# Vérifier si le répertoire existe
if [ -d "$nas_dir" ]; then
    echo "Suppression de tous les fichiers dans le répertoire $nas_dir"

    # Supprimer tous les fichiers dans le répertoire
    rm -rf "$nas_dir"/*

    echo "Tous les fichiers ont été supprimés avec succès."
else
    echo "Le répertoire $nas_dir n'existe pas."
fi

================================================
zafar@apache-guaca:/opt/scripts$ cat watch_guac_log.sh
#!/bin/bash
#
# Script qui surveille les logs de Guacamole et envoie une notification lors d'une connexion.

LOGFILE="/var/log/tomcat9/catalina.out"
KEYWORD="User .* connected to connection"

# Suivi en temps réel des logs
tail -n0 -F "$LOGFILE" | while read -r LINE; do
    # Vérifie si la ligne contient le mot-clé indiquant une connexion
    echo "$LINE" | grep -E "$KEYWORD" > /dev/null
    if [ $? -eq 0 ]; then
        # Extraction du nom de l'utilisateur et de la connexion
        USER_NAME=$(echo "$LINE" | awk -F"" '{print $2}')
        CONNECTION_ID=$(echo "$LINE" | awk -F'connected to connection "' '{print $2}' | awk
-F"" '{print $1}')
        DATE_CONNEXION=$(date '+%Y-%m-%d %H:%M:%S')

        # Log local (pour debug)
        echo "$(date '+%Y-%m-%d %H:%M:%S') - Connexion détectée : $USER_NAME sur
connexion $CONNECTION_ID" >> /tmp/guac_notify_watch.log

        # Appel du script de notification
        /opt/scripts/guac_notify.sh "$USER_NAME" "$CONNECTION_ID"
    fi
done
```

Mes Tache déjà effectué sur le projet VPN

1. Installation et configuration de Guacamole

- Déploiement et configuration de **Guacamole** sur le serveur.
- Téléchargement des dépendance version 2024
- Mis en place une base de données Mariadb -SQL

2. Enregistrement des sessions sur le NAS

- Montage du NAS (`/mnt/nas/guacamole-recordings`).
- Un script convertit les enregistrements **.fb** en **.m4v**,
- Un script les transfère vers un NAS
- Une fois les vidéos transférées, un autre script les supprime localement et vide le répertoire NAS tous les 10 jours à midi via une tâche **cron**.
- Script d'envoi le notif de connexion
- SCRIPT ALERT CONEXION SSH

3. Sécurisation de Guacamole

- **Installation et configuration de Fail2Ban** pour bloquer les IP après **3 tentatives de connexion échouées** sur SSH et Guacamole.
- **Renforcer la sécurité** en activant le **HTTPS obligatoire** et en **désactivant HTTP**.
- Redirection http et ip vers https
- Configure ssl certificate

Filtrage des ports ouverts

- **Scan des ports avec Nmap** (`nmap -p- apache-guacamole.daudruy.net`).
- Fermeture des **ports inutiles** dans les règles de pare-feu (`ufw deny 80`, `ufw allow 443`…)
- Message d'alerte de connexion (Bannière SSH)
- Authentification TOTP

4. Personnalisation de l'interface Guacamole

- Ajout d'un **logo et personnalisation des couleurs**.

5. DNS

Supervision de Guacamole avec Nagios dans un Conteneur Docker sur Ubuntu

On va mettre en place un serveur de supervision Nagios dans un conteneur Docker sur Ubuntu pour surveiller Guacamole.

Plan d'action

1. Introduction à Nagios et Docker
2. Installation de Docker et Docker Compose sur Ubuntu
3. Déploiement de Nagios dans un conteneur
4. Configuration de Nagios pour superviser Guacamole
5. Ajout de services et vérifications

Introduction à Nagios et Docker

Nagios est une solution open-source de supervision permettant de surveiller les serveurs, services et équipements réseau. Il peut alerter en cas de panne et fournir des métriques sur l'état des machines.

Docker permet de déployer des applications dans des conteneurs légers, facilitant l'installation et la maintenance des services comme Nagios.

L'objectif ici est de superviser Guacamole en utilisant Nagios exécuté dans un conteneur Docker.

Installation de Docker et Docker Compose sur Ubuntu

```
zafar@auth-srv:~$ sudo apt install -y docker.io
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Le paquet suivant a été installé automatiquement et n'est plus nécessaire :
  libllvm17t64
```

```
zafar@auth-srv:~$ docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
zafar@auth-srv:~$
```

On a activé et démarré Docker puis on ajoute un utilisateur au group Docker qui exécute commande suo san mot de passe puis on installe docker compose et enfin on vérifie la version de composer

```
zafar@auth-srv:~$ sudo systemctl enable --now docker
zafar@auth-srv:~$ sudo usermod -aG docker $USER
zafar@auth-srv:~$ newgrp docker
zafar@auth-srv:~$ sudo apt install -y docker-compose
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Le paquet suivant a été installé automatiquement et n'est plus nécessaire :
```

```
zafar@auth-srv:~$ docker-compose --version
docker-compose version 1.29.2, build unknown
zafar@auth-srv:~$
```

Déploiement de Nagios dans un Conteneur Docker

On crée un répertoire pour Nagios et un fichier `docker-compose.yml` pour Nagios

```
zafar@auth-srv:~$ mkdir -p ~/nagios && cd ~/nagios
zafar@auth-srv:~/nagios$ nano docker-compose.yml
zafar@auth-srv:~/nagios$
```

```
  GNU nano 7.2                                  docker-compose.yml *
version: '3'
services:
  nagios:
    image: jasonrivers/nagios
    container_name: nagios
    restart: always
    ports:
      - "8080:80"
    volumes:
      - ./nagios/etc:/opt/nagios/etc
      - ./nagios/var:/opt/nagios/var
      - ./nagios/logs:/var/log/nagios
      - ./nagios/plugins:/opt/Custom-Nagios-Plugins
    environment:
      - NAGIOSADMIN_USER=admin
      - NAGIOSADMIN_PASS=admin
```

On lance e conteneur

```
zafar@auth-srv:~/nagios$ docker-compose up -d
Creating network "nagios_default" with the default driver
Pulling nagios (jasonrivers/nagios:)...
latest: Pulling from jasonrivers/nagios
ff65ddf9395b: Pulling fs layer
785b9873bdf4: Pulling fs layer
785b9873bdf4: Extracting [================================>                  ]  181.6MB/275.3MB
53aff88babc4: Download complete
d72f92e29533: Download complete
706ed7d4ce0a: Download complete
```

```
9a90645e352c: Pull complete
8e911c59da28: Pull complete
c219d58cc3f9: Pull complete
b0e280e9aa8c: Pull complete
8c389e58e867: Pull complete
Digest: sha256:2a7c2b20d118baf92b47b69a3901e68dd7664617801b94e560bc4d6564d6ae54
Status: Downloaded newer image for jasonrivers/nagios:latest
Creating nagios ... done
zafar@auth-srv:~/nagios$
```

Nagios fonctionne

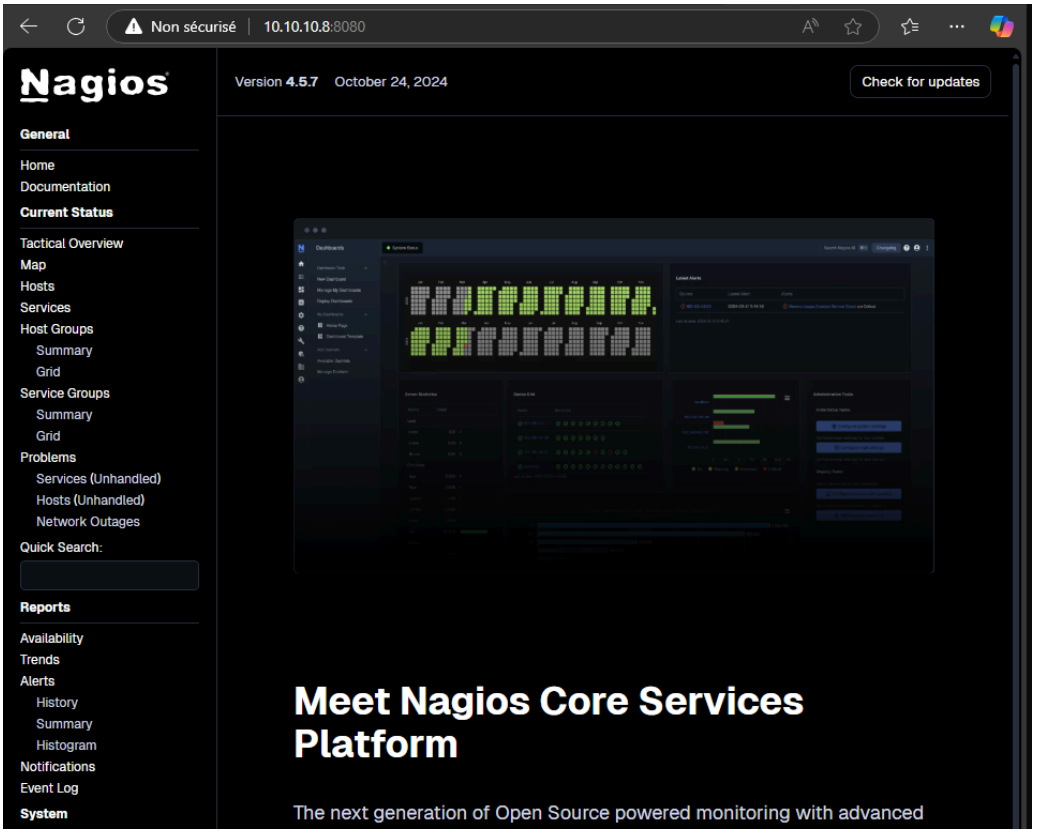
```
zafar@auth-srv:~/nagios$ docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED              STATUS              PORTS                                                NAMES
b7dd8225b84e   jasonrivers/nagios   "/usr/local/bin/star…"   About a minute ago   Up About a minute   5667/tcp, 0.0.0.0:8080->80/tcp, :::8080->80/tcp   nagios
zafar@auth-srv:~/nagios$
```

Accéder à l'interface Web de Nagios

👉 **http://=10.10.10.8:8080**

👤 **Utilisateur :** `admin`

🔑 **Mot de passe :** `admin`



Configuration de Nagios pour Superviser Guacamole

Nous allons maintenant configurer **Nagios** pour superviser **Guacamole**. On va utiliser le protocole **HTTP** pour vérifier si l'interface Web de Guacamole est accessible.

On accède au conteneur Nagios

```
zafar@auth-srv:~/nagios$ docker exec -it nagios /bin/bash
root@b7dd8225b84e:/#
```

On édite le fichier de configuration des hôtes

```
zafar@auth-srv:~/nagios$ docker exec -it nagios /bin/bash
root@b7dd8225b84e:/# nano /opt/nagios/etc/objects/guacamole.cfg
bash: nano: command not found
root@b7dd8225b84e:/# apt update && apt install nano -y
Get:1 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
```

```
  GNU nano 7.2                    /home/zafar/nagios/nagios/etc/objects/guacamole.cfg *
define host {
    use linux-server
    host_name guacamole-server
    alias Guacamole Server
    address 10.10.10.4  ; Mets l'IP correcte de ton serveur Guacamole
    max_check_attempts 5
    check_period 24x7
    notification_interval 30
    notification_period 24x7
}

define service {
    use generic-service
    host_name guacamole-server
    service_description HTTP Web Interface
    check_command check_http
}
```

On Inclure ce fichier dans la config Nagios

```
root@b7dd8225b84e:/# echo 'cfg_file=/opt/nagios/etc/objects/guacamole.cfg' >> /opt/nagios/etc/nagios.cfg
root@b7dd8225b84e:/#
```

Puis on redémarre le service

```
zafar@auth-srv:~/nagios$ docker restart nagios
nagios
zafar@auth-srv:~/nagios$ docker ps
CONTAINER ID   IMAGE               COMMAND                  CREATED          STATUS         PORTS
                NAMES
b7dd8225b84e   jasonrivers/nagios   "/usr/local/bin/star…"   10 minutes ago   Up 5 seconds   5667/tcp, 0.0.0.0:8080->80/tcp,
:::8080->80/tcp   nagios
```

Fichier des commande

```
root@b7dd8225b84e:/# nano /opt/nagios/etc/objects/commands.cfg
root@b7dd8225b84e:/#
```

```
zafar@auth-srv:~/nagios$ docker exec -it nagios /bin/bash
root@b7dd8225b84e:/# apt update && apt install -y nagios-plugins
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
```

```
root@b7dd8225b84e:/# ls -l /usr/lib/nagios/plugins/check_http
-rwxr-xr-x 1 root root 81608 Apr  1  2024 /usr/lib/nagios/plugins/check_http
root@b7dd8225b84e:/# /usr/lib/nagios/plugins/check_http -H 10.10.10.4
HTTP OK: HTTP/1.1 200 - 3127 bytes in 0.002 second response time |time=0.002388s;;;0.000000;10.000000 size=3127B;;;0;
```

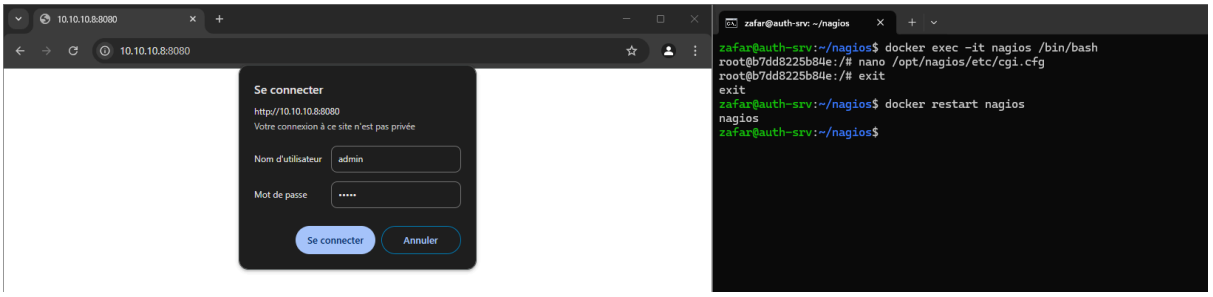
Droit :

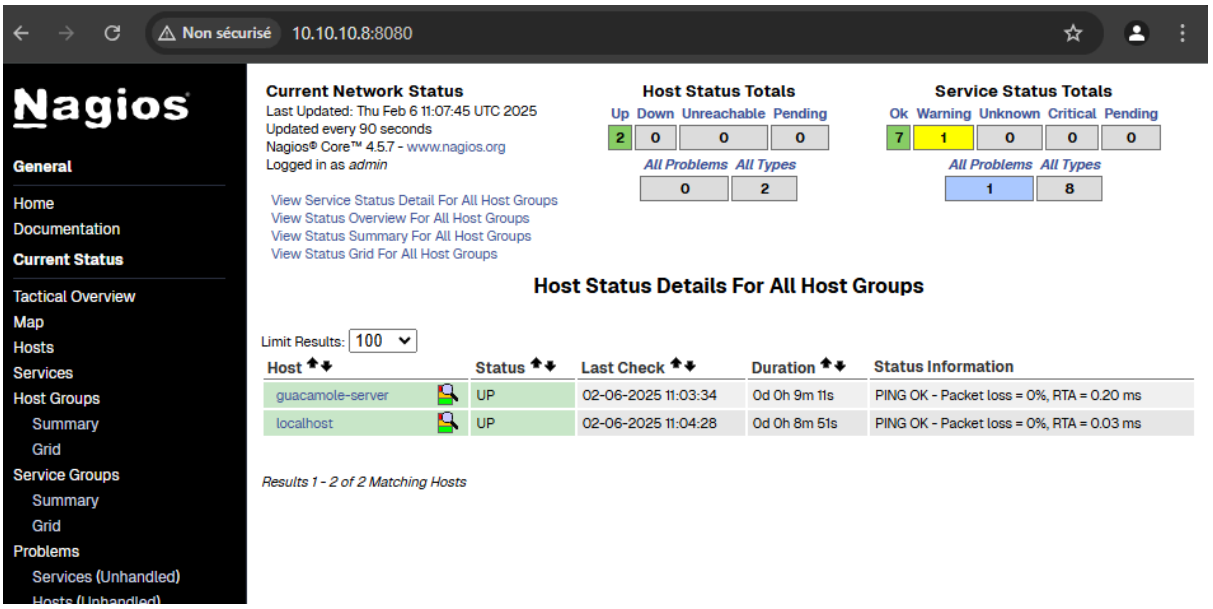
```
zafar@auth-srv:~/nagios$ docker exec -it nagios /bin/bash
root@b7dd8225b84e:/# chown -R nagios:nagios /opt/nagios/var
root@b7dd8225b84e:/# chmod -R 775 /opt/nagios/var
root@b7dd8225b84e:/#
```

```
root@b7dd8225b84e:/# exit
exit
zafar@auth-srv:~/nagios$ docker restart nagios
nagios
zafar@auth-srv:~/nagios$
```

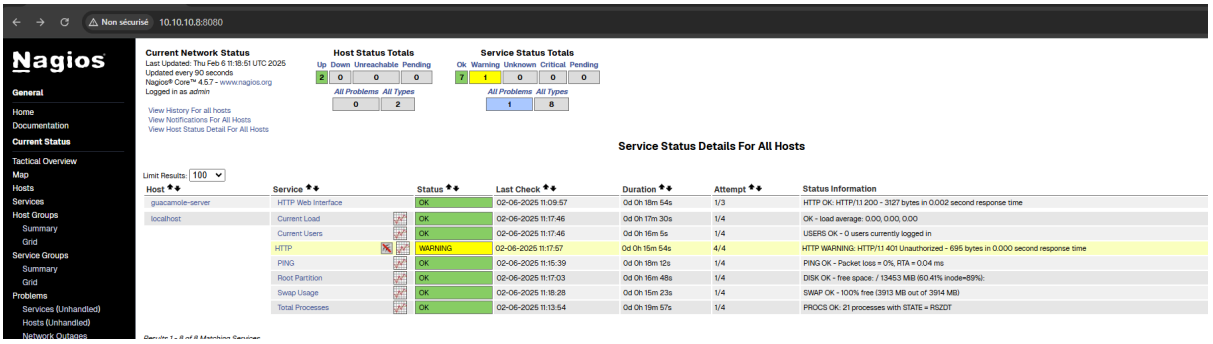
Authorisation de tous les service et configuration pour user admin

```
 GNU nano 7.2                                  /opt/nagios/etc/cgi.cfg


authorized_for_all_services=admin
authorized_for_all_hosts=admin



# GLOBAL HOST/SERVICE COMMAND ACCESS
# These two options are comma-delimited lists of all usernames that
# can issue host or service related commands via the command
# CGI (cmd.cgi) for all hosts and services that are being monitored.
# By default, users can only issue commands for hosts or services
# that they are contacts for (unless you you choose to not use
# authorization).  You may use an asterisk (*) to authorize any
# user who has authenticated to the web server.

authorized_for_all_service_commands=admin
authorized_for_all_host_commands=admin
```





Super ! 🎉 mon serveur Guacamole est bien surveillé avec Nagios. Maintenant, on peut aller plus loin avec des configurations avancées.



Super ! 🎉 mon serveur Guacamole est bien surveillé avec Nagios. Maintenant, on peut aller plus loin avec des configurations avancées.

2️⃣ Vérifier si le processus Guacamole tourne

On cree notre commande

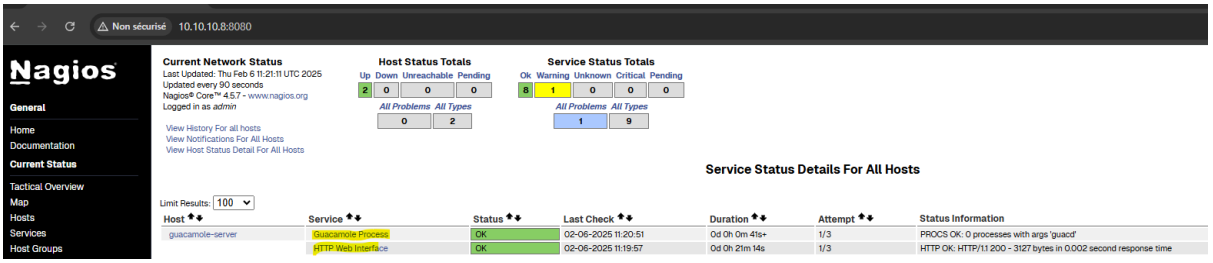
```
/opt/nagios/etc/objects/commands.cfg *
```

```
define command {
    command_name    check_procs
    command_line    $USER1$/check_procs -c $ARG1$ -a $ARG2$
}
```

```
/opt/nagios/etc/objects/guacamole.cfg *
```

```
define service {
    use                 generic-service
    host_name           guacamole-server
    service_description Guacamole Process
    check_command       check_procs!1!guacd
}
```

```
root@b7dd8225b84e:/# ls /usr/lib/nagios/plugins/check_procs
/usr/lib/nagios/plugins/check_procs
root@b7dd8225b84e:/# nano /opt/nagios/etc/objects/commands.cfg
root@b7dd8225b84e:/# nano /opt/nagios/etc/objects/guacamole.cfg
root@b7dd8225b84e:/# /opt/nagios/bin/nagios -v /opt/nagios/etc/nagios.cfg
```

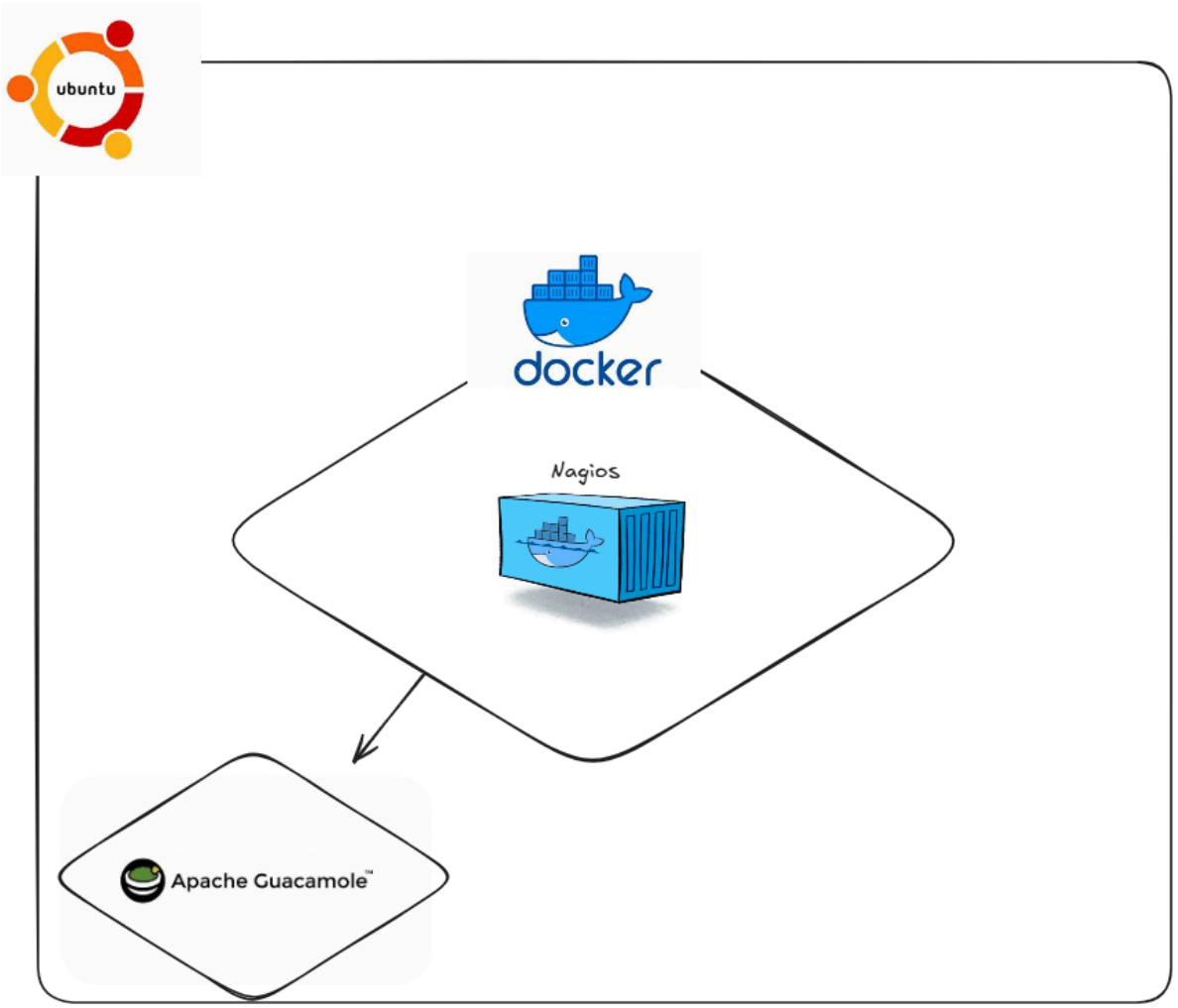


Super ! 🎉

```
define contact {

    contact_name            nagiosadmin             ; Short name of user
    use                     generic-contact         ; Inherit default values from generic-contact te
    alias                   Nagios Admin            ; Full name of user
    email                   suprot@daudruy.fr ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS ******
}
```

Tous les service de supervision :

```
root@b7dd8225b84e:/# ls /opt/nagios/libexec/
check-mqtt.py   check_ftp          check_mailq              check_oracle       check_ssmtp
check_apt       check_game         check_mem.pl             check_overcr       check_swap
check_breeze    check_hpjd         check_mrtg               check_pgsql        check_tcp
check_by_ssh    check_http         check_mrtgtraf           check_ping         check_time
check_clamd     check_icmp         check_mssql_database.py  check_pop          check_udp
check_cluster   check_ide_smart    check_mssql_server.py    check_procs        check_ups
check_dbi       check_ifoperstatus check_nagios             check_real         check_uptime
check_dhcp      check_ifstatus     check_ncpa.py            check_rpc          check_users
check_dig       check_imap         check_nntp               check_sensors      check_vpn
check_disk      check_ircd         check_nntps              check_simap        check_wave
check_disk_smb  check_jabber       check_nrpe               check_smtp         mibs
check_dns       check_jenkins      check_nt                 check_snmp         negate
check_dummy     check_ldap         check_ntp                check_spop         remove_perfdata
check_file_age  check_ldaps        check_ntp_peer           check_sql          urlize
check_flexlm    check_load         check_ntp_time           check_ssh          utils.pm
check_fping     check_log          check_nwstat             check_ssl_validity utils.sh
root@b7dd8225b84e:/#
```


ubuntu
docker
Nagios
Apache Guacamole™

Phase de mise en production guacamole

Changer le mote de passe root

```
root@apache-guaca:~# sudo passwd root
New password:
Retype new password:
passwd: password updated successfully
root@apache-guaca:~# _
```

Changer le mote de passe admin

D'ailluer Nous avons tenté de changer le nom de l'utilisateur configuré pour Guacamole, mais cette modification a entraîné des erreurs liées aux permissions, à la base de données et aux services système (Tomcat, MariaDB). Après plusieurs essais, la configuration ne fonctionnait plus correctement.

Étant en dernière semaine de stage et avec l'accord de mon responsable, j'ai décidé de conserver l'ancien nom d'utilisateur afin d'assurer la stabilité du service et d'éviter toute interruption.

```
root@apache-guaca:~# passwd zafar
New password:
Retype new password:
passwd: password updated successfully
root@apache-guaca:~#
```

Changement les information de la base de données

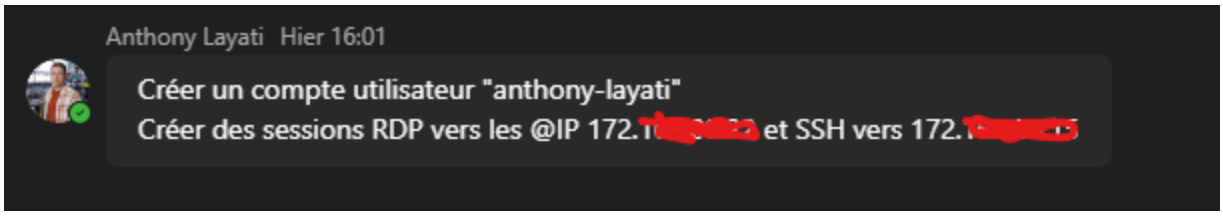
```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SELECT user, host FROM mysql.user;
+-------------+-----------+
| User        | Host      |
+-------------+-----------+
| mariadb.sys | localhost |
| mysql       | localhost |
| root        | localhost |
| userdb      | localhost |
+-------------+-----------+
4 rows in set (0,001 sec)

MariaDB [(none)]> RENAME USER 'userdb'@'localhost' TO 'admin'@'localhost';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> ALTER USER 'admin'@'localhost' IDENTIFIED BY 'Daudruy@2025';
Query OK, 0 rows affected (0,000 sec)

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

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

```
zafar@apache-guaca:~# mysql -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 96
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON guacamole_db.* TO 'admin'@'localhost' IDENTIFIED BY '[redacted]';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
```

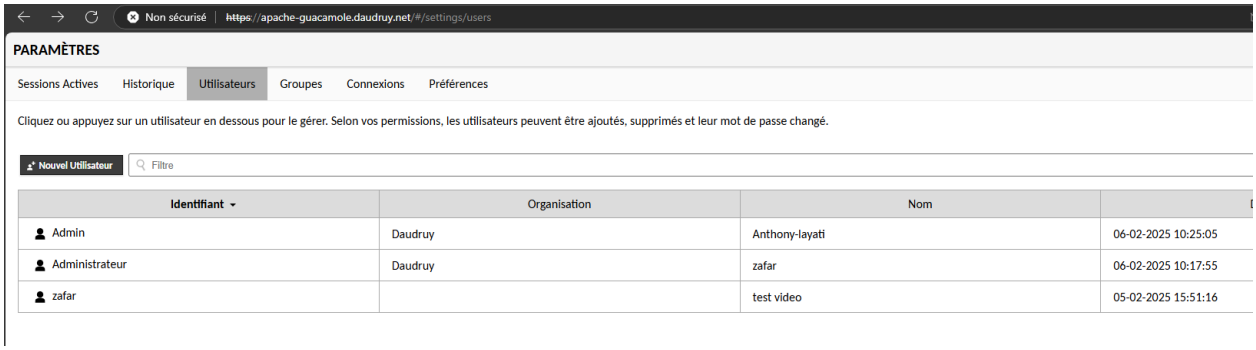
```
MariaDB [(none)]> SELECT user, host FROM mysql
+-------------+-----------+
| User        | Host      |
+-------------+-----------+
| admin       | localhost |
| mariadb.sys | localhost |
| mysql       | localhost |
| root        | localhost |
+-------------+-----------+
```

Premier connexion situation reel

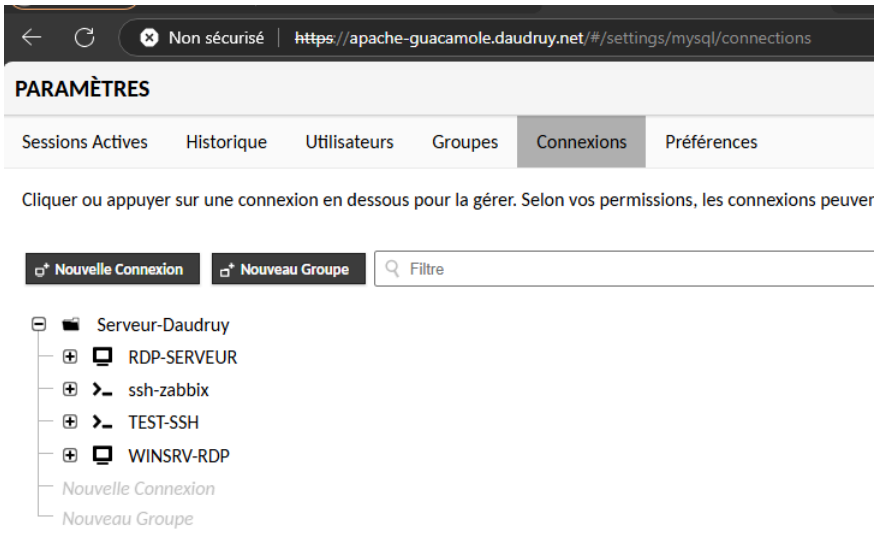



```
  GNU nano 6.2                                /etc/guacamole/guacamole.properties
#declaration de de la connexion a Mariadb
#ce fichier est utile aussi pour d'autre parametres

# MySQL ------------------------
mysql-hostname: 127.0.0.1
mysql-port: 3[redacted]
mysql-database: guacadb
mysql-username: admin
mysql-password: Daud[redacted]
#-------------------------------
```

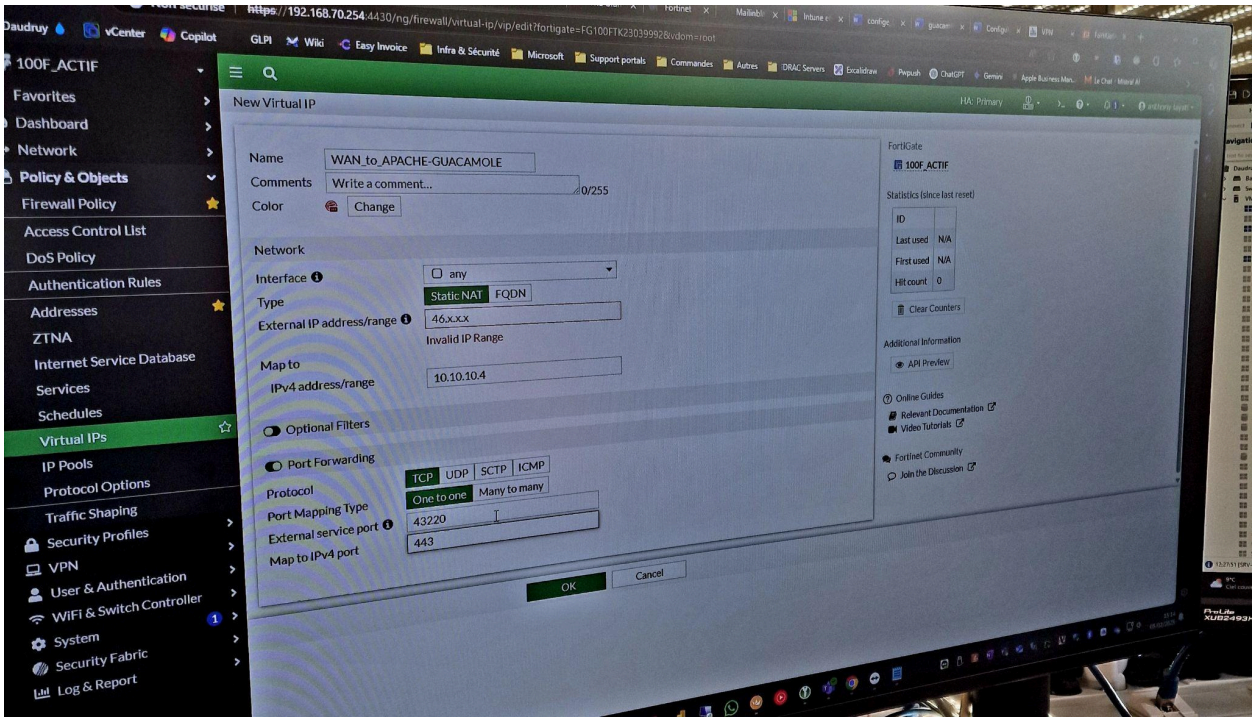
Chagemengem admin guacamole

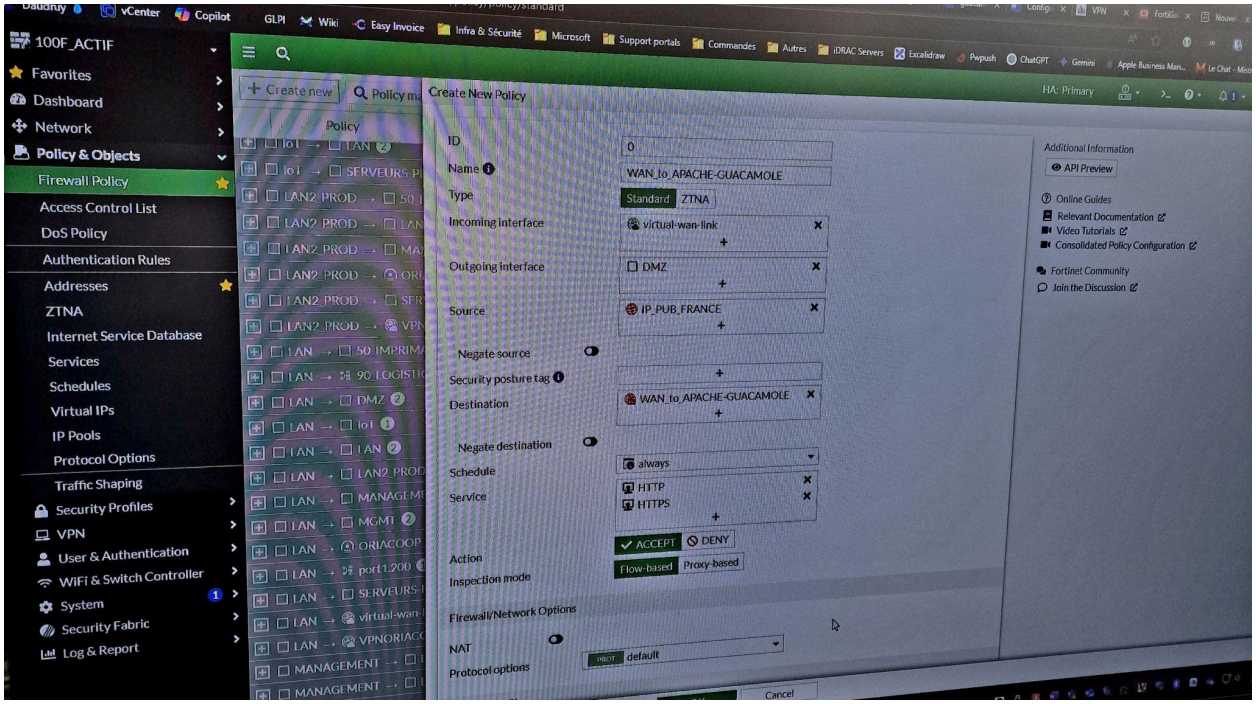


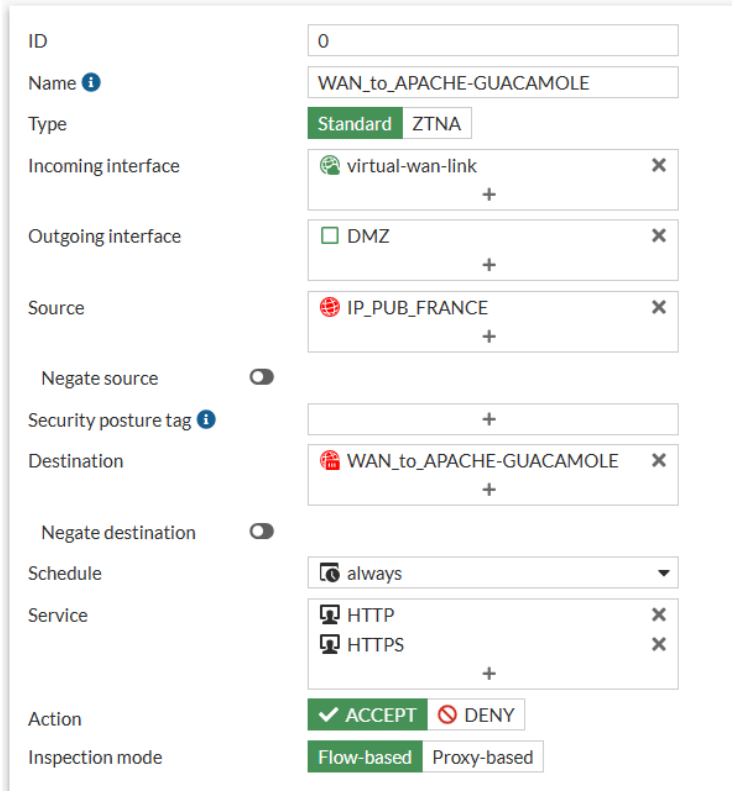
Création des nouveau connexion



Configuration NAT




100F_ACTIF
Favorites
Dashboard
Network
Policy & Objects
Firewall Policy
Access Control List
DoS Policy
Authentication Rules
Addresses
ZTNA
Internet Service Database
Services
Schedules
Virtual IPs
IP Pools
Protocol Options
Traffic Shaping
Security Profiles
VPN
User & Authentication
WiFi & Switch Controller
System
Security Fabric
Log & Report
Create new
Policy
Create New Policy
ID
0
Name
WAN_to_APACHE-GUACAMOLE
Type
Standard
ZTNA
Incoming Interface
virtual-wan-link
Outgoing Interface
DMZ
Source
IP_PUB_FRANCE
Negate source
Security posture tag
Destination
WAN_to_APACHE-GUACAMOLE
Negate destination
Schedule
always
Service
HTTP
HTTPS
Action
ACCEPT
DENY
Inspection mode
Flow-based
Proxy-based
Firewall/Network Options
NAT
Protocol options
default
Additional Information
API Preview
Online Guides
Relevant Documentation
Video Tutorials
Consolidated Policy Configuration
Fortinet Community
Join the Discussion
HA: Primary
Cancel



Config ip et port pour avoir accès depuis internet

N configure le ip et le port de guacamole pour répondre sur internet car sur wan pour le moment il y a pas de DNS et Certificate

```
 GNU nano 6.2                                guacamole.conf *
<VirtualHost *:443>
    ServerName 10.10.10.4  # depuis interent il reponde avec cette ip apres on pourra la chagenr avec un DNS
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/ton-certificat.crt
    SSLCertificateKeyFile /etc/ssl/private/ton-certificat.key

    ProxyPreserveHost On
    ProxyPass / http://127.0.0.1:8080/  # Guacamole tourne sur Tomcat sur le port 8080
    ProxyPassReverse / http://127.0.0.1:8080/

    ErrorLog ${APACHE_LOG_DIR}/guacamole_error.log
    CustomLog ${APACHE_LOG_DIR}/guacamole_access.log combined
</VirtualHost>
```

On as testé ce jour là de se connecter depuis internet en tapantt 10.10.10.4:443 et ca marche de coup il reste pour le equip de chète le certificat et un nome de domaine

En local

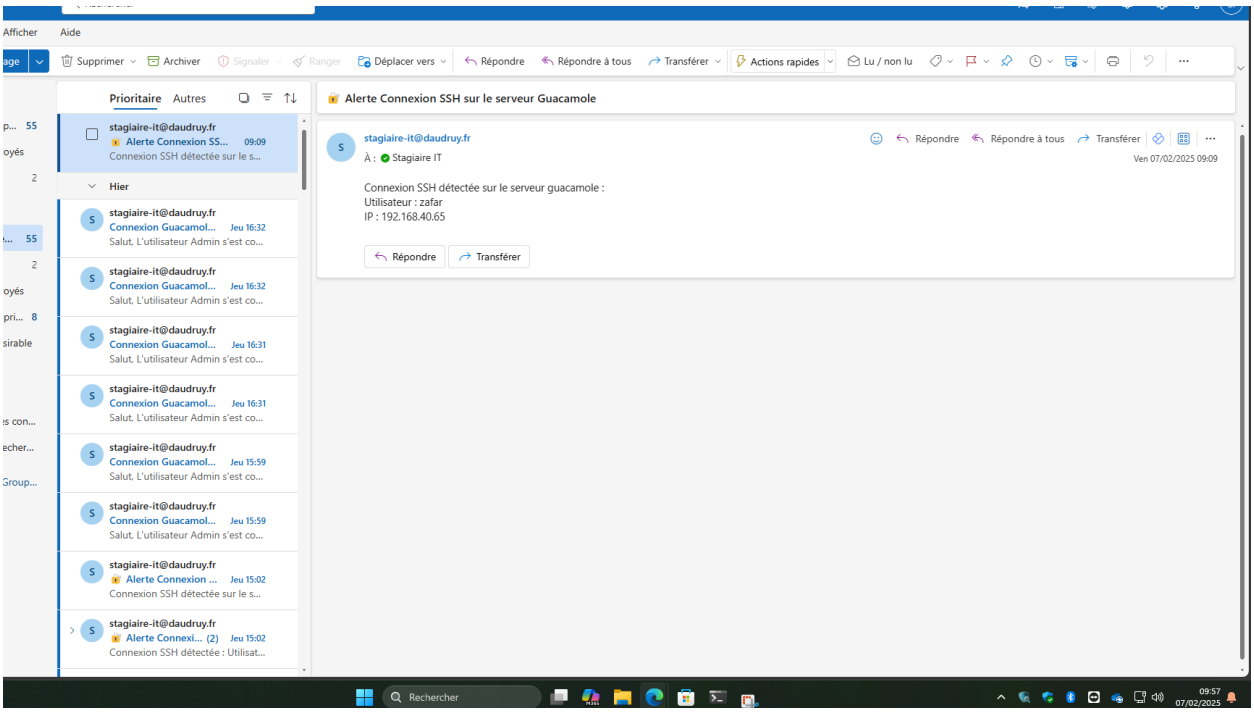
En local il ya bien un dsn configurer et un certificate autosigné que j'ai déposé sur pc après pour le dépôt de certi on peut aussi utiliser gpo

```
zafar@apache-guaca:/etc/apache2/sites-available# ls
000-default.conf  apache-guacamole.conf  default-ssl.conf  guacamole.conf
zafar@apache-guaca:/etc/apache2/sites-available#
```

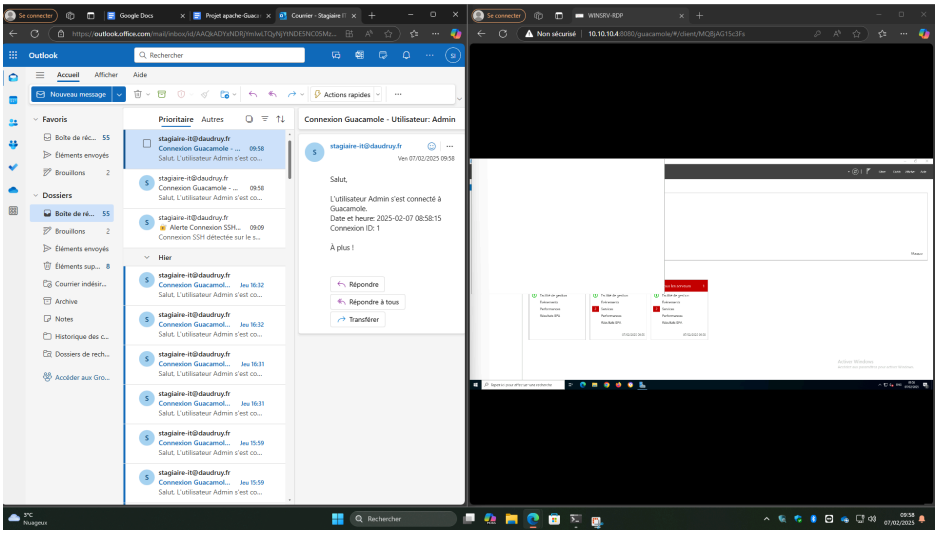
```
 GNU nano 6.2                                apache-guacamole.conf
<VirtualHost *:80>
    ServerName apache-guacamole.daudruy.net

    ProxyPreserveHost On
    ProxyPass / http://127.0.0.1:8080/guacamole/
    ProxyPassReverse / http://127.0.0.1:8080/guacamole/

    ErrorLog ${APACHE_LOG_DIR}/guacamole_error.log
    CustomLog ${APACHE_LOG_DIR}/guacamole_access.log combined
</VirtualHost>

<VirtualHost *:443>
    ServerName apache-guacamole.daudruy.net

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/apacheguac.crt
    SSLCertificateKeyFile /etc/ssl/private/apacheguac.key

    ProxyPreserveHost On
    ProxyPass / http://127.0.0.1:8080/guacamole/
    ProxyPassReverse / http://127.0.0.1:8080/guacamole/

    ErrorLog ${APACHE_LOG_DIR}/guacamole_ssl_error.log
    CustomLog ${APACHE_LOG_DIR}/guacamole_ssl_access.log combined
</VirtualHost>
```

Test des script

Script envoi notification connexion ssh sur serveur guacamole



Script envie notification connexion rdp :



Automatiser le script de conversion vidéo et envoi vers NAS

```
  GNU nano 6.2                                  /tmp/crontab.2WItPE/crontab
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command



# script s'exécute tous les 10 jours à 13h (1 PM) pour supprimer les videos de NAS
0 13 */10 * * /opt/scripts/nas_supprime.sh



#Automatiser l'exécution du script pour envoie et conversion des video
0 */2 * * * /bin/bash /opt/scripts/envoie_et_nettoie.sh >> /var/log/envoie_et_nettoie.log 2>&1
```

```
Étape 1.1 : Conversion du fichier 20250128-125117 - RDP - Administrateur en .m4v
guacenc: INFO: Guacamole video encoder (guacenc) version 1.5.5
guacenc: INFO: 1 input file(s) provided.
guacenc: INFO: Video will be encoded at 1280x720 and 2000000 bps.
guacenc: INFO: Encoding "/var/lib/guacamole/recordings/d9666d3f-5d6a-335d-bf49-e052d6130967/20250128-125117 - RDP -
nistrateur" to "/var/lib/guacamole/recordings/d9666d3f-5d6a-335d-bf49-e052d6130967/20250128-125117 - RDP - Administr
.m4v"
```

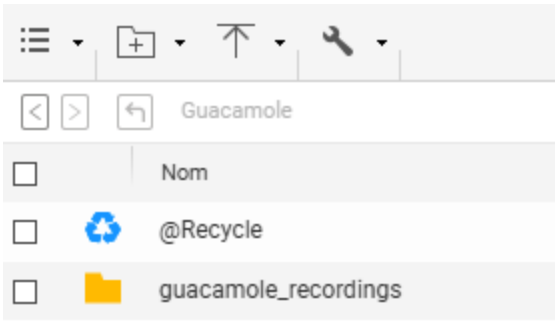
```
sent 8.355.152 bytes  received 35 bytes  16.710.374,00 bytes/sec
total size is 8.352.991  speedup is 1,00
Fichier 20250205-145117 - RDP - zafar transféré avec succès vers le NAS.==========================
Étape 1.3 : Suppression du fichier local /var/lib/guacamole/recordings/ff8a25da-2118-3dd8-a03f-fe1af05a6896/20250205-145
117 - RDP - zafar et fichier converti
Fichier local 20250205-145117 - RDP - zafar supprimé avec succès.
Étape 1.4 : Vérification et suppression du répertoire /var/lib/guacamole/recordings/ff8a25da-2118-3dd8-a03f-fe1af05a6896
```

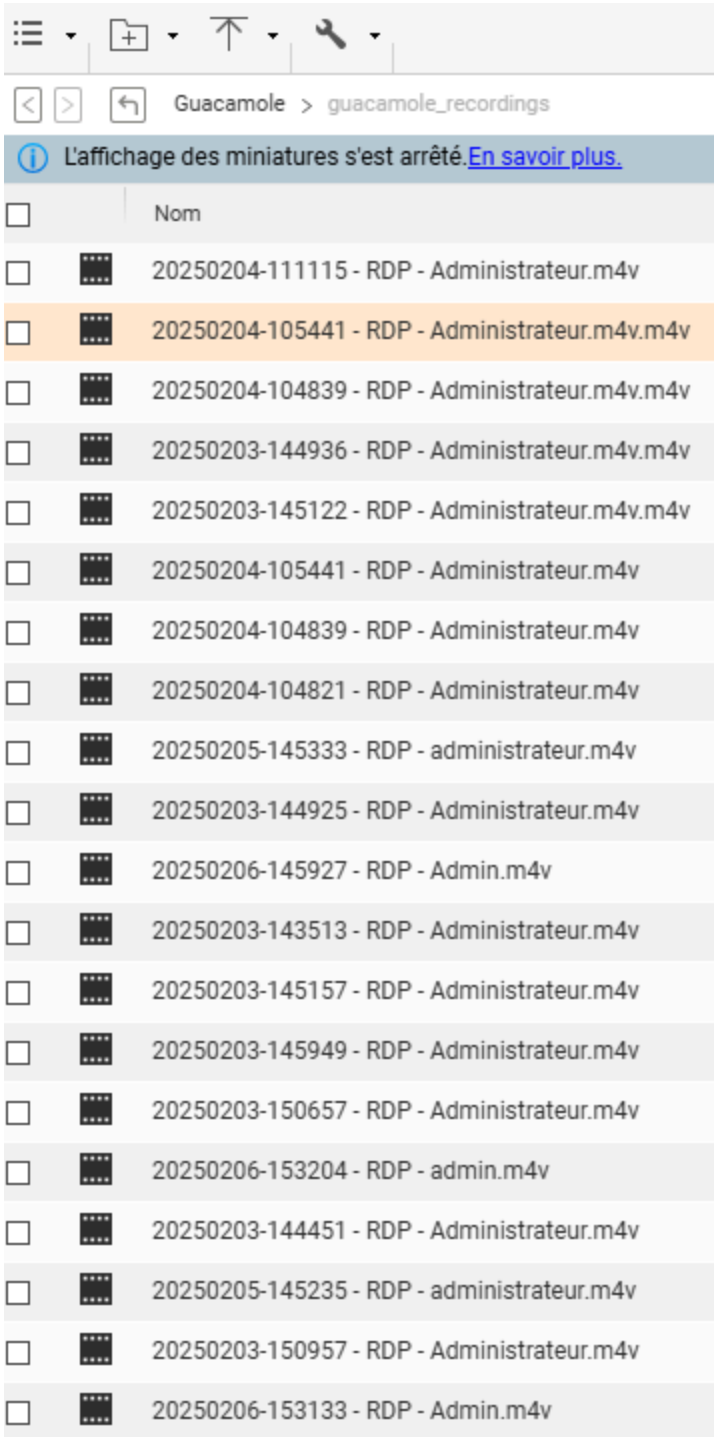
Le vedio sont transmis sur nas

```
zafar@apache-guaca:/opt/scripts$ sudo ls -l /mnt/nas/guacamole_recordings/
total 525652
-rwxr-xr-x 1 root root   5684234 févr.  7 09:06 '20250128-125117 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root   1666610 févr.  7 09:05 '20250203-143513 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root     10761 févr.  7 09:06 '20250203-144441 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root   1195583 févr.  7 09:05 '20250203-144451 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root     10761 févr.  7 09:02 '20250203-144925 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root   1329092 févr.  7 09:05 '20250203-144936 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root     10761 févr.  4 11:12 '20250203-144936 - RDP - Administrateur.m4v.m4v'
-rwxr-xr-x 1 root root   1742395 févr.  7 09:05 '20250203-145122 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root     10761 févr.  4 11:12 '20250203-145122 - RDP - Administrateur.m4v.m4v'
-rwxr-xr-x 1 root root   5517383 févr.  7 09:05 '20250203-145157 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root     10761 févr.  7 09:05 '20250203-145232 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root     10761 févr.  7 09:05 '20250203-145949 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root   2930123 févr.  7 09:05 '20250203-150657 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root     10761 févr.  7 09:05 '20250203-150742 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root     10761 févr.  7 09:06 '20250203-150806 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root     10761 févr.  7 09:05 '20250203-150957 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root   1844228 févr.  4 11:15 '20250204-104821 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root   2261745 févr.  4 11:15 '20250204-104839 - RDP - Administrateur.m4v'
-rwxr-xr-x 1 root root     10761 févr.  4 11:12 '20250204-104839 - RDP - Administrateur.m4v.m4v'
-rwxr-xr-x 1 root root  21576876 févr.  4 11:15 '20250204-105441 - RDP - Administrateur.m4v'
```

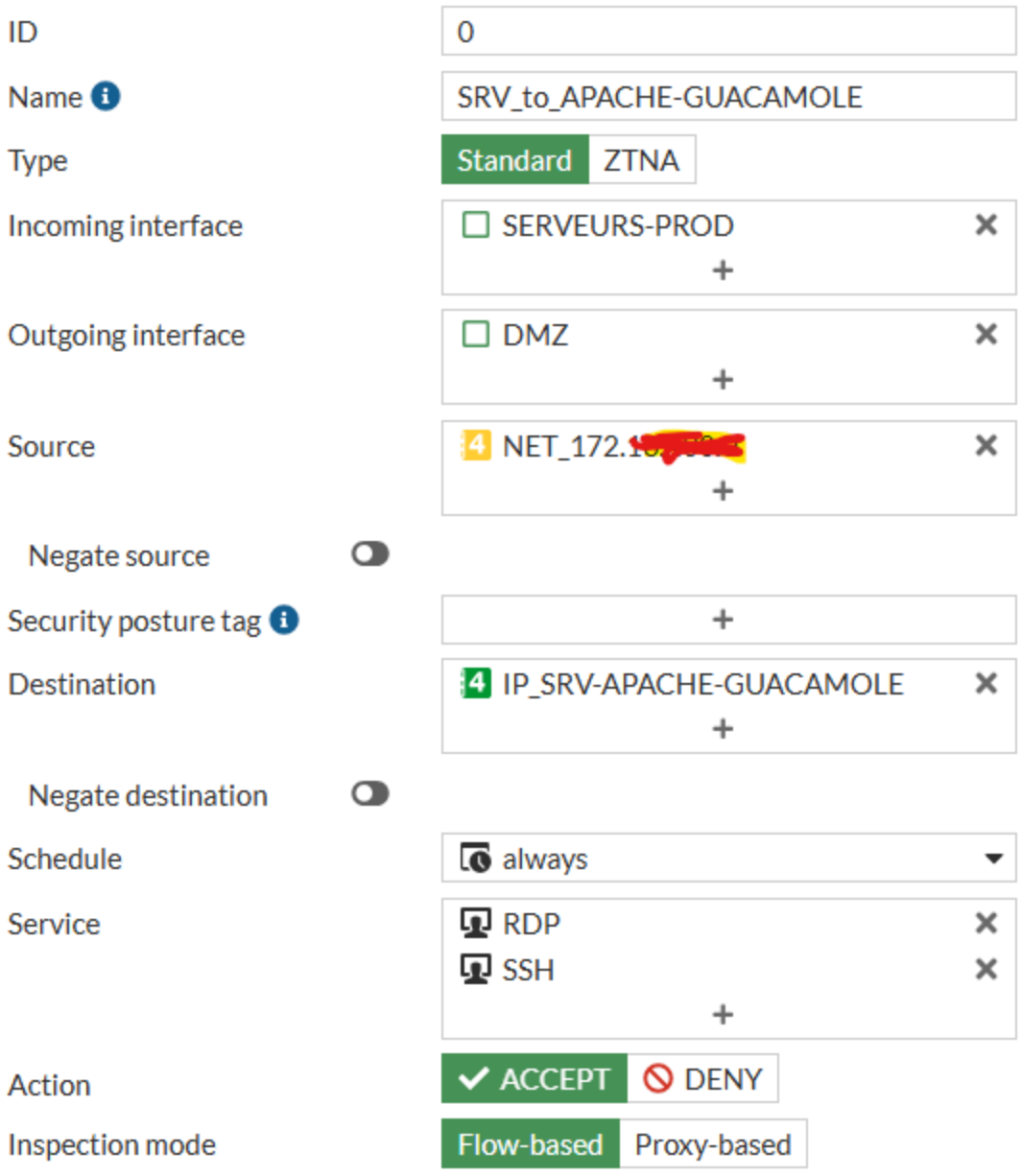
Dossier local ets vide apres execution de script

```
zafar@apache-guaca:/var/lib/guacamole/recordings$ ls
zafar@apache-guaca:/var/lib/guacamole/recordings$
```




Guacamole > guacamole_recordings
L'affichage des miniatures s'est arrêté. En savoir plus.
Nom
20250204-111115 - RDP - Administrateur.m4v
20250204-105441 - RDP - Administrateur.m4v.m4v
20250204-104839 - RDP - Administrateur.m4v.m4v
20250203-144936 - RDP - Administrateur.m4v.m4v
20250203-145122 - RDP - Administrateur.m4v.m4v
20250204-105441 - RDP - Administrateur.m4v
20250204-104839 - RDP - Administrateur.m4v
20250204-104821 - RDP - Administrateur.m4v
20250205-145333 - RDP - administrateur.m4v
20250203-144925 - RDP - Administrateur.m4v
20250206-145927 - RDP - Admin.m4v
20250203-143513 - RDP - Administrateur.m4v
20250203-145157 - RDP - Administrateur.m4v
20250203-145949 - RDP - Administrateur.m4v
20250203-150657 - RDP - Administrateur.m4v
20250206-153204 - RDP - admin.m4v
20250203-144451 - RDP - Administrateur.m4v
20250205-145235 - RDP - administrateur.m4v
20250203-150957 - RDP - Administrateur.m4v
20250206-153133 - RDP - Admin.m4v

Création de policy sur pare-feu pour laisser paser les connexion entre guacamole et les service de production



Nettoyage de disque local et nas

```
zafar@apache-guaca:/var/lib/guacamole/recordings$ sudo /opt/scripts/
envoie_et_nettoie.sh      monitor_guacamole_ssh.py  watch_guac_log.sh
guac_notify.sh            nas_supprime.sh
zafar@apache-guaca:/var/lib/guacamole/recordings$ sudo /opt/scripts/
envoie_et_nettoie.sh      monitor_guacamole_ssh.py  watch_guac_log.sh
guac_notify.sh            nas_supprime.sh
zafar@apache-guaca:/var/lib/guacamole/recordings$ sudo /opt/scripts/nas_supprime.sh
Suppression de tous les fichiers dans le répertoire /mnt/nas/guacamole_recordings/
Tous les fichiers ont été supprimés avec succès.
zafar@apache-guaca:/var/lib/guacamole/recordings$ ls -l /mnt/nas/guacamole_recordings/
total 0
zafar@apache-guaca:/var/lib/guacamole/recordings$
```

Création un super utilisateur

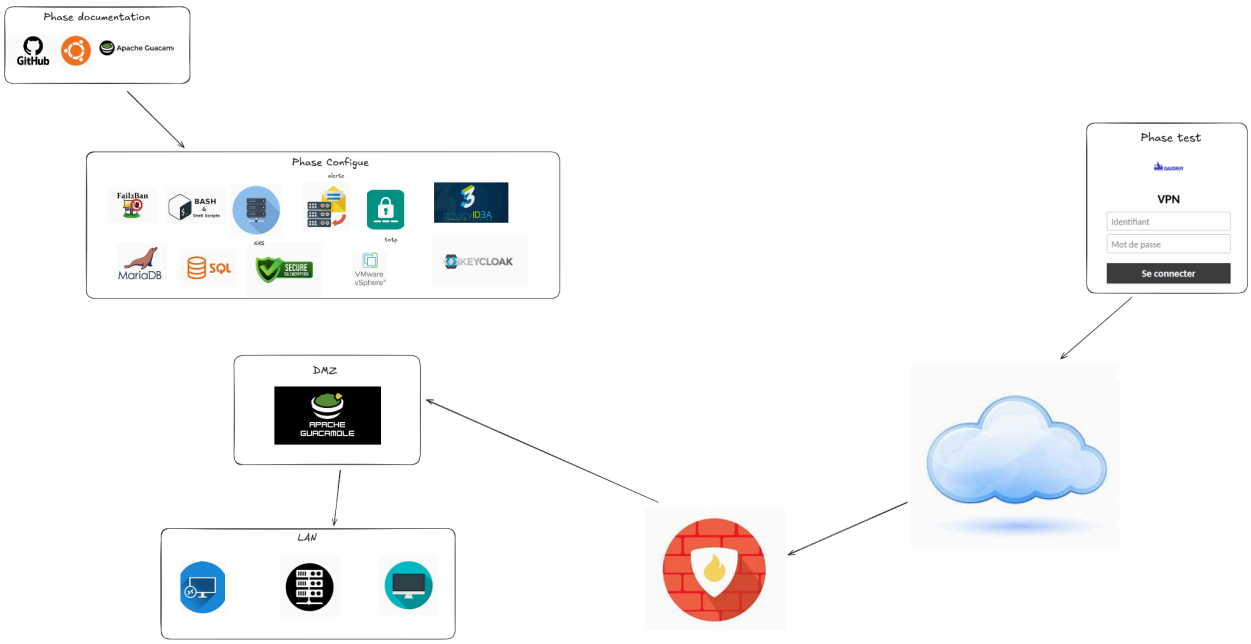

```
C:\Users\stagiaire-it>ssh admin@10.10.10.4
################################################################################
#
#    AVERTISSEMENT DE SÉCURITÉ - ENTREPRISE DAUDRUY
#
# Vous accédez à un système sécurisé de l'entreprise Daudruy. Toute
# connexion est enregistrée, y compris votre adresse IP, votre heure de
# connexion et votre nom d'utilisateur. Ces informations peuvent être
# utilisées à des fins de sécurité et de conformité avec la législation
```

```
Last login: Fri Feb  7 13:41:55 2025 from 192.168.40.65
 apache-guaca  admin  ~
```

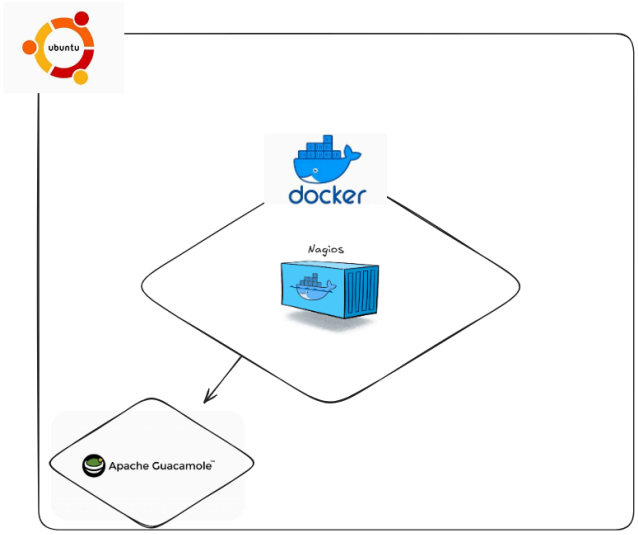
Changer le banner

```
C:\Users\stagiaire-it>ssh admin@10.10.10.4
##############################################################################
#      🚨 AVERTISSEMENT DE SÉCURITÉ – ENTREPRISE DAUDRUY 🚨
#
#      ⚠️ Accès strictement réservé aux utilisateurs autorisés.
#      🕵️ Toute connexion est enregistrée et notifiée à l'équipe IT.
#      🔴 Une tentative d'accès non autorisée peut entraîner des poursuites.
#
#      📩 Support IT : support@daudruy.fr
#
#      ──────────────────────────────────────────────────────────
#      🔒 SECURITY WARNING – DAUDRUY COMPANY 🔒
#
#      ⚠️ Authorized users only. Unauthorized access is prohibited.
#      🕵️ All logins are logged and notified to the IT team.
#      🔴 Violations may result in legal action.
#
#      📩 IT Support : support@daudruy.fr
##############################################################################
```

Test depuis wan




Nagios
KEYCLOAK
privacyIDEA
Fail2Ban
Apache Guacamole
utile


ubuntu
docker
Nagios
Apache Guacamole