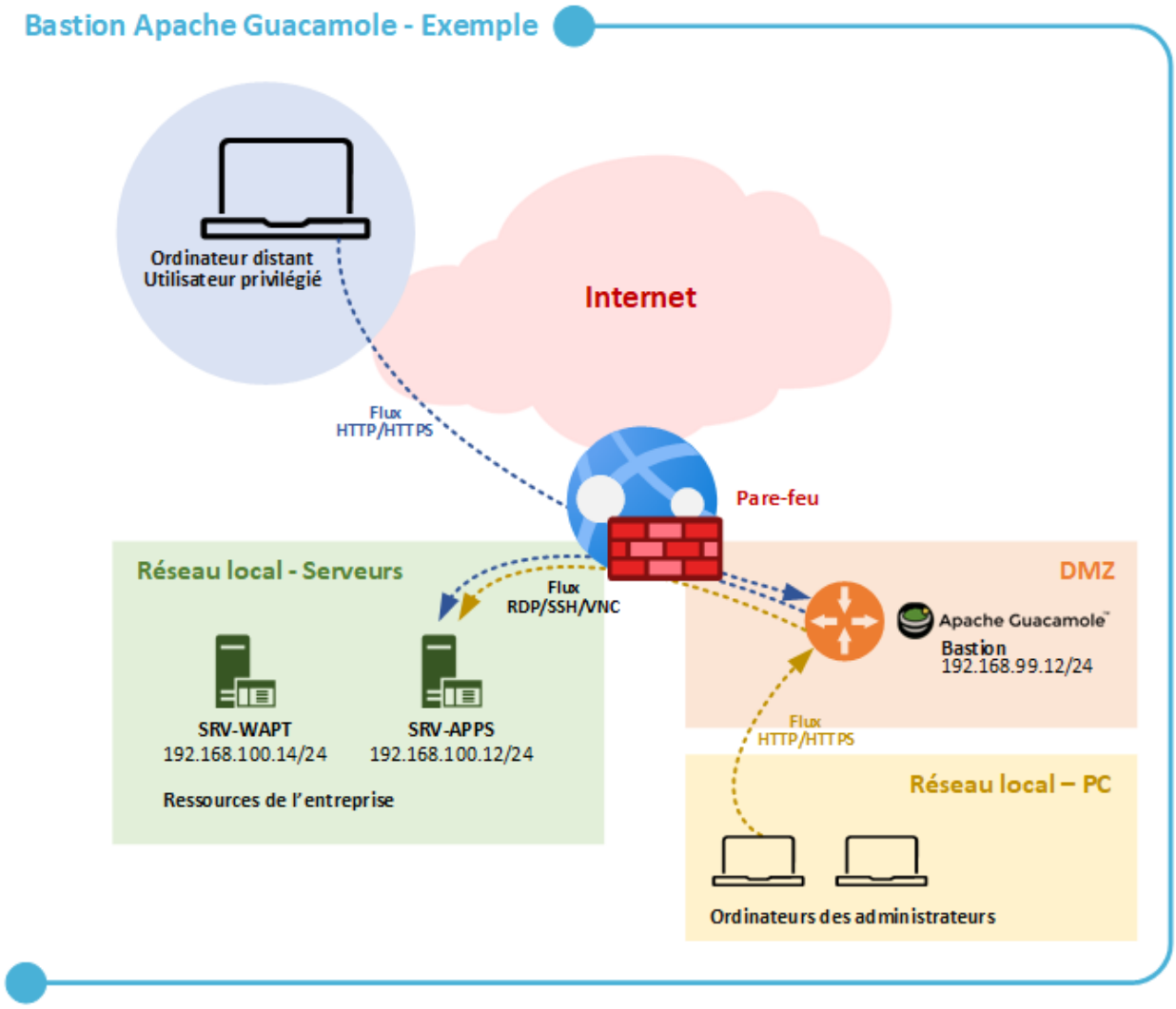
Le serveur Apache Guacamole sera utilisé comme **point d'entrée unique** pour accéder aux serveurs et équipements de l'infrastructure que ce soit via **les protocoles RDP, SSH, VNC et Telnet**, et même Kubernetes. Que l'on soit en externe ou en interne, **les connexions aux serveurs vont obligatoirement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

Dans l'exemple ci-dessous, **l'hôte Apache Guacamole est positionné en DMZ** puisqu'il doit être accessible depuis l'extérieur. L'accès depuis l'extérieur n'est pas obligatoire puisque l'on pourrait imposer une connexion VPN au réseau de l'entreprise avant de permettre la connexion sur l'interface de Guacamole. De la même manière pour publier l'hôte Guacamole sur Internet, il est recommandé de **s'appuyer sur un reverse proxy en frontal** (le pare-feu pourrait très bien assurer cette fonction), ce qui permettra en même temps de passer les flux en HTTPS.



Apache Guacamole devient un élément central de l'infrastructure puisqu'il sert de **passerelle pour administrer les machines**. Rassurez-vous, il est possible d'avoir

plusieurs hôtes Apache Guacamole pour **répartir la charge et assurer la haute disponibilité**.

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.

Version originale de l'article : 21 juin 2023

II. Les fonctions clés d'Apache Guacamole

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

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

Pour les personnes qui utilisent Azure, sachez qu'Apache Guacamole représente une alternative au Bastion Azure. Si ce sujet vous intéresse, lisez cet article :

- Apache Guacamole dans Azure

III. Installer Apache Guacamole sur Debian

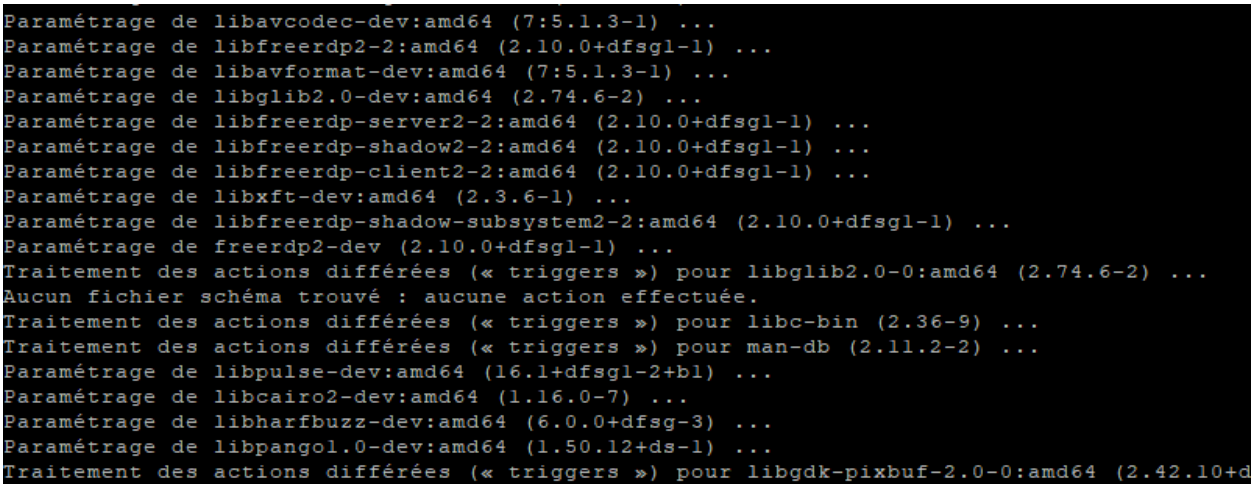
A. Installer les prérequis d'Apache Guacamole

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.

- Documentation - Dépendances Apache Guacamole

Sur la machine Debian, on commence par installer ces fameuses dépendances avec la commande suivante :

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



On attend gentiment la fin de l'installation.

La partie "cliente" d'Apache Guacamole nécessite d'installer un serveur Tomcat, mais nous allons effectuer cette opération plus tard.

Pour effectuer l'installation depuis un compte utilisateur, sans utiliser le compte "**root**" directement, pensez à installer "**sudo**" et à ajouter un utilisateur au groupe correspondant. L'exemple ci-dessous donne les permissions à l'utilisateur "**flo**" :

apt-get install sudo

usermod -aG sudo flo

Ensuite, préfixez par "**sudo**" les commandes qui nécessitent une élévation de privilèges.

sudo apt-get update

B. Compiler et installer Apache Guacamole "Server"

La partie "Server" d'Apache Guacamole doit être téléchargée et compilée en local pour s'installer. La dernière version sera utilisée, à savoir la version 1.5.2. 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 :

```
cd /tmp
wget https://downloads.apache.org/guacamole/1.5.5/source/guacamole-server-1.5.5.tar.gz
```

Par la suite, lorsqu'il y aura une nouvelle version d'Apache Guacamole, il faudra revoir l'URL ci-dessus (et d'autres présentes dans la suite de cet article).

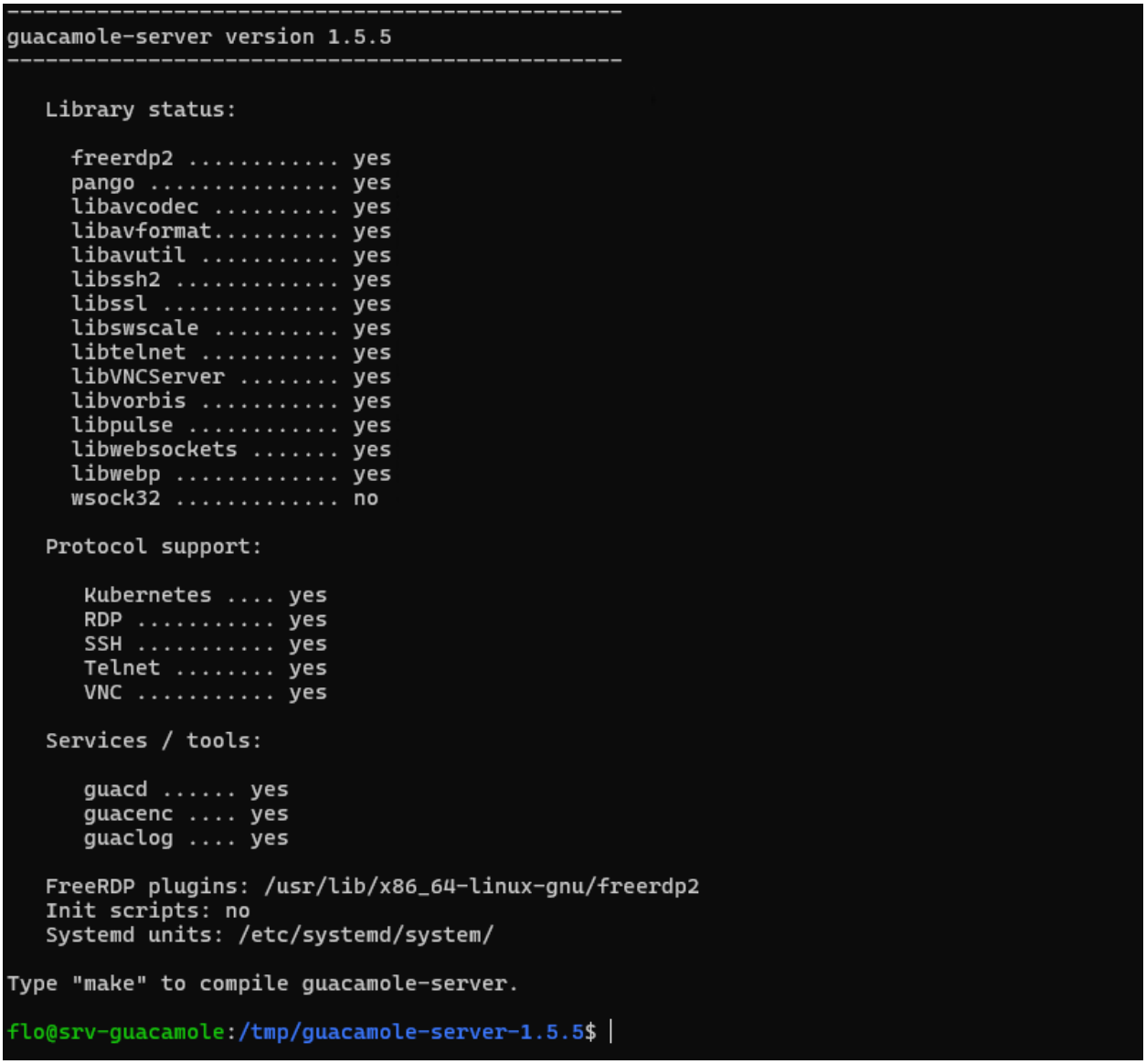
Une fois le téléchargement terminé, on décompresse l'archive tar.gz et on se positionne dans le répertoire obtenu :

```
tar -xzf guacamole-server-1.5.5.tar.gz
cd 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 :

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

Avant de passer à la suite, on vérifie la sortie de cette commande. Normalement, les librairies principales et dont nous avons besoin sont sur le statut "yes". Comme ceci :



Regardez bien la sortie de la commande précédente, afin de vérifier la présence éventuelle d'une erreur. Si vous obtenez une erreur qui spécifie "**guacenc_video_alloc**", c'est lié au composant "**guacenc**" qui est utilisé pour créer les enregistrements au format vidéo (lié à FFmpeg). Dans ce cas, vous pouvez relancer la commande précédente en désactivant ce composant :

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

Ensuite, poursuivez avec la compilation du code source de guacamole-server :

sudo make

Enfin, on termine par installer le composant Guacamole Server :

sudo make install

```
make[2] : on quitte le répertoire « /tmp/guacamole-server-1.5.5/src/guacd »
make[1] : on quitte le répertoire « /tmp/guacamole-server-1.5.5/src/guacd »
Making install in src/guacenc
make[1] : on entre dans le répertoire « /tmp/guacamole-server-1.5.5/src/guacenc »
make[2] : on entre dans le répertoire « /tmp/guacamole-server-1.5.5/src/guacenc »
 /usr/bin/mkdir -p '/usr/local/bin'
  /bin/bash ../../libtool   --mode=install /usr/bin/install -c guacenc '/usr/local/bin'
libtool: install: /usr/bin/install -c .libs/guacenc /usr/local/bin/guacenc
 /usr/bin/mkdir -p '/usr/local/share/man/man1'
 /usr/bin/install -c -m 644 man/guacenc.1 '/usr/local/share/man/man1'
make[2] : on quitte le répertoire « /tmp/guacamole-server-1.5.5/src/guacenc »
make[1] : on quitte le répertoire « /tmp/guacamole-server-1.5.5/src/guacenc »
Making install in src/guaclog
make[1] : on entre dans le répertoire « /tmp/guacamole-server-1.5.5/src/guaclog »
make[2] : on entre dans le répertoire « /tmp/guacamole-server-1.5.5/src/guaclog »
 /usr/bin/mkdir -p '/usr/local/bin'
  /bin/bash ../../libtool   --mode=install /usr/bin/install -c guaclog '/usr/local/bin'
libtool: install: /usr/bin/install -c .libs/guaclog /usr/local/bin/guaclog
 /usr/bin/mkdir -p '/usr/local/share/man/man1'
 /usr/bin/install -c -m 644 man/guaclog.1 '/usr/local/share/man/man1'
make[2] : on quitte le répertoire « /tmp/guacamole-server-1.5.5/src/guaclog »
make[1] : on quitte le répertoire « /tmp/guacamole-server-1.5.5/src/guaclog »
make[1] : on entre dans le répertoire « /tmp/guacamole-server-1.5.5 »
make[2] : on entre dans le répertoire « /tmp/guacamole-server-1.5.5 »
make[2]: rien à faire pour « install-exec-am ».
make[2]: rien à faire pour « install-data-am ».
make[2] : on quitte le répertoire « /tmp/guacamole-server-1.5.5 »
make[1] : on quitte le répertoire « /tmp/guacamole-server-1.5.5 »
flo@srv-guacamole:/tmp/guacamole-server-1.5.5$
```

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

Mais il y a d'autres étapes à réaliser...

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

sudo ldconfig

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.

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

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

sudo systemctl status guacd

C. Créer le répertoire de configuration

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

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

D. Installer Guacamole Client (Web App)

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

Nous allons ajouter un nouveau fichier source pour Apt. Créez le fichier suivant :

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

Ajoutez cette ligne, enregistrez et fermez le fichier.

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

Mettez à jour le cache des paquets :

```
sudo apt-get update
```

Effectuez l'**installation des paquets Tomcat 9 sur Debian 12** avec cette commande :

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

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

```
cd /tmp
wget https://downloads.apache.org/guacamole/1.5.5/binary/guacamole-1.5.5.war
```

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

```
sudo mv guacamole-1.5.5.war /var/lib/tomcat9/webapps/guacamole.war
```

Puis, on relance les services Tomcat9 et Guacamole :

```
sudo systemctl restart tomcat9 guacd
```

Voilà, Apache Guacamole Client est installé !

E. Base de données MariaDB pour l'authentification

Cette dernière étape avant de commencer à utiliser Apache Guacamole consiste à **déployer MariaDB Server (ou MySQL Server, au choix) sur Debian pour qu'Apache Guacamole s'appuie sur une base de données**. Cette base de données sera utilisée pour stocker toutes les informations de l'application.

On commence par installer le paquet MariaDB Server :

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

Puis, on exécute le script ci-dessous pour **sécuriser un minimum notre instance** (changer le mot de passe root, désactiver les accès anonymes, etc...). Si besoin d'aide pour cette partie, je vous encourage à regarder ce tutoriel.

```
sudo mysql_secure_installation
```

Une fois cette étape effectuée, on va se connecter en tant que root à notre instance MariaDB :

mysql -u root -p

Ceci est utile pour **créer une base de données et un utilisateur dédié pour Apache Guacamole**. Les commandes ci-dessous permettent de créer la base de données "**guacadb**", avec l'utilisateur "**guaca_nachos**" associé au mot de passe "**P@ssword!**" (adaptez ces valeurs). Cet utilisateur dispose de quelques droits sur la base de données.

CREATE DATABASE guacadb;

CREATE USER 'guaca_nachos'@'localhost' IDENTIFIED BY 'P@ssword!';

GRANT SELECT,INSERT,UPDATE,DELETE ON guacadb.* TO 'guaca_nachos'@'localhost';

FLUSH PRIVILEGES;

EXIT;

```
Thanks for using MariaDB!
root@srv-guacamole:/tmp# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 38
Server version: 10.5.19-MariaDB-0+deb11u2 Debian 11

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 'guaca_nachos'@'localhost' IDENTIFIED BY '          ';
Query OK, 0 rows affected (0,001 sec)

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

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]>
```

La suite va consister à **ajouter l'extension MySQL à Apache Guacamole** ainsi que le connecteur correspondant. Encore quelques fichiers à télécharger depuis Internet.

Toujours depuis le dépôt officiel, on télécharge cette extension :

cd /tmp
wget https://downloads.apache.org/guacamole/1.5.5/binary/guacamole-auth-jdbc-1.5.5.tar.gz

Puis, on décompresse l'archive tar.gz obtenue :

tar -xzf guacamole-auth-jdbc-1.5.5.tar.gz

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

sudo mv guacamole-auth-jdbc-1.5.5/mysql/guacamole-auth-jdbc-mysql-1.5.5.jar /etc/guacamole/extensions/

Ensuite, le connecteur MySQL doit être téléchargé depuis le site de MySQL (peu importe si vous utilisez MariaDB ou MySQL).

Utilisez le lien ci-dessous pour repérer le lien de la dernière version en choisissant "**Platform Independent**", puis en cliquant sur le bouton "**Download**" permettant d'obtenir la "**Compressed TAR Archive**".

- Télécharger le connecteur MySQL

Une autre page se charge, copiez le lien sous "**No thanks, just start my download.**". Pour la version actuelle, à savoir 8.0.33, le lien est inclus à la commande ci-dessous.

On lance le téléchargement :

cd /tmp
wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-j-9.1.0.tar.gz

Puis, on décompresse l'archive tar.gz :

tar -xzf mysql-connector-j-9.1.0.tar.gz

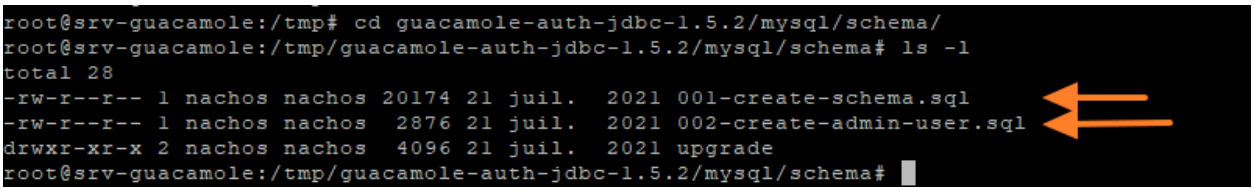
On copie (ou déplace) le fichier .jar du connecteur vers le répertoire "lib" d'Apache Guacamole :

sudo cp mysql-connector-j-9.1.0/mysql-connector-j-9.1.0.jar /etc/guacamole/lib/

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 saisit pour effectuer l'import.

cd guacamole-auth-jdbc-1.5.5/mysql/schema/
cat *.sql | mysql -u root -p guacadb



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.

sudo nano /etc/guacamole/guacamole.properties

Dans ce fichier, insérez les lignes ci-dessous en adaptant les trois dernières lignes avec vos valeurs :

MySQL

mysql-hostname: 127.0.0.1

mysql-port: 3306

mysql-database: guacadb

mysql-username: guaca_nachos

mysql-password: P@ssword!

Enregistrez et fermez le fichier.

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).

sudo nano /etc/guacamole/guacd.conf

Voici le code à intégrer :

[server]

bind_host = 0.0.0.0

bind_port = 4822

On enregistre et on termine par redémarrer les trois services liés à Apache Guacamole :

sudo systemctl restart tomcat9 guacd mariadb

Voilà, l'installation de base est terminée !

IV. Premiers pas avec Apache Guacamole

On va pouvoir se connecter à Apache Guacamole pour effectuer nos premiers pas sur l'interface de la Web App.

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

Une page de connexion va s'afficher :

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

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

Bienvenue sur Apache Guacamole ! Même si pour l'instant, c'est vide... Encore un peu de courage !

A. Créer un nouveau compte admin

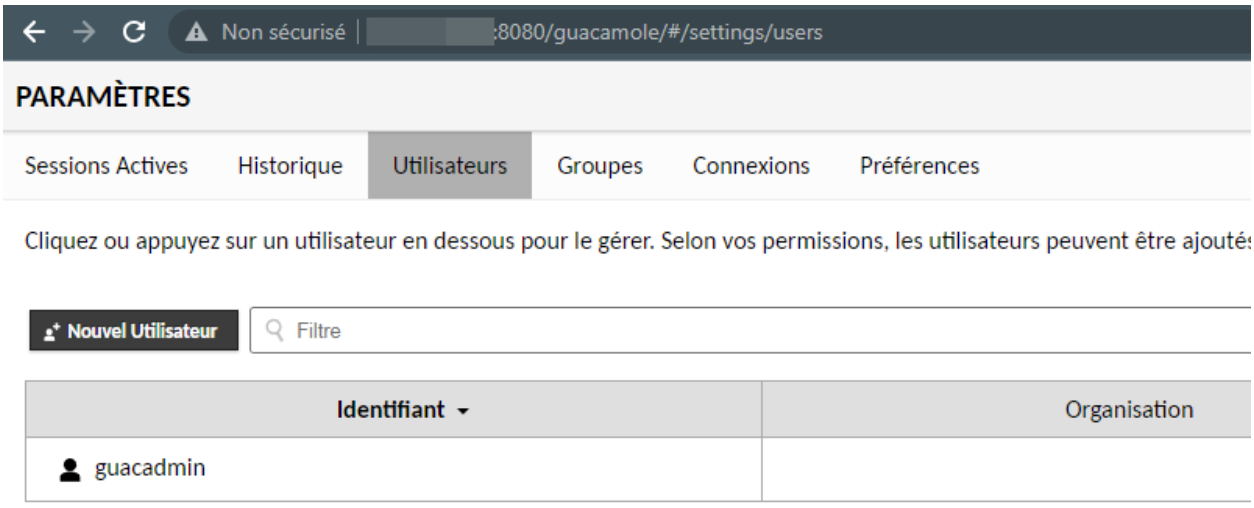
Tout d'abord, nous allons **créer un nouveau compte d'administration** et supprimer le compte par défaut, pour des raisons de sécurité.

Notre objectif est le suivant :

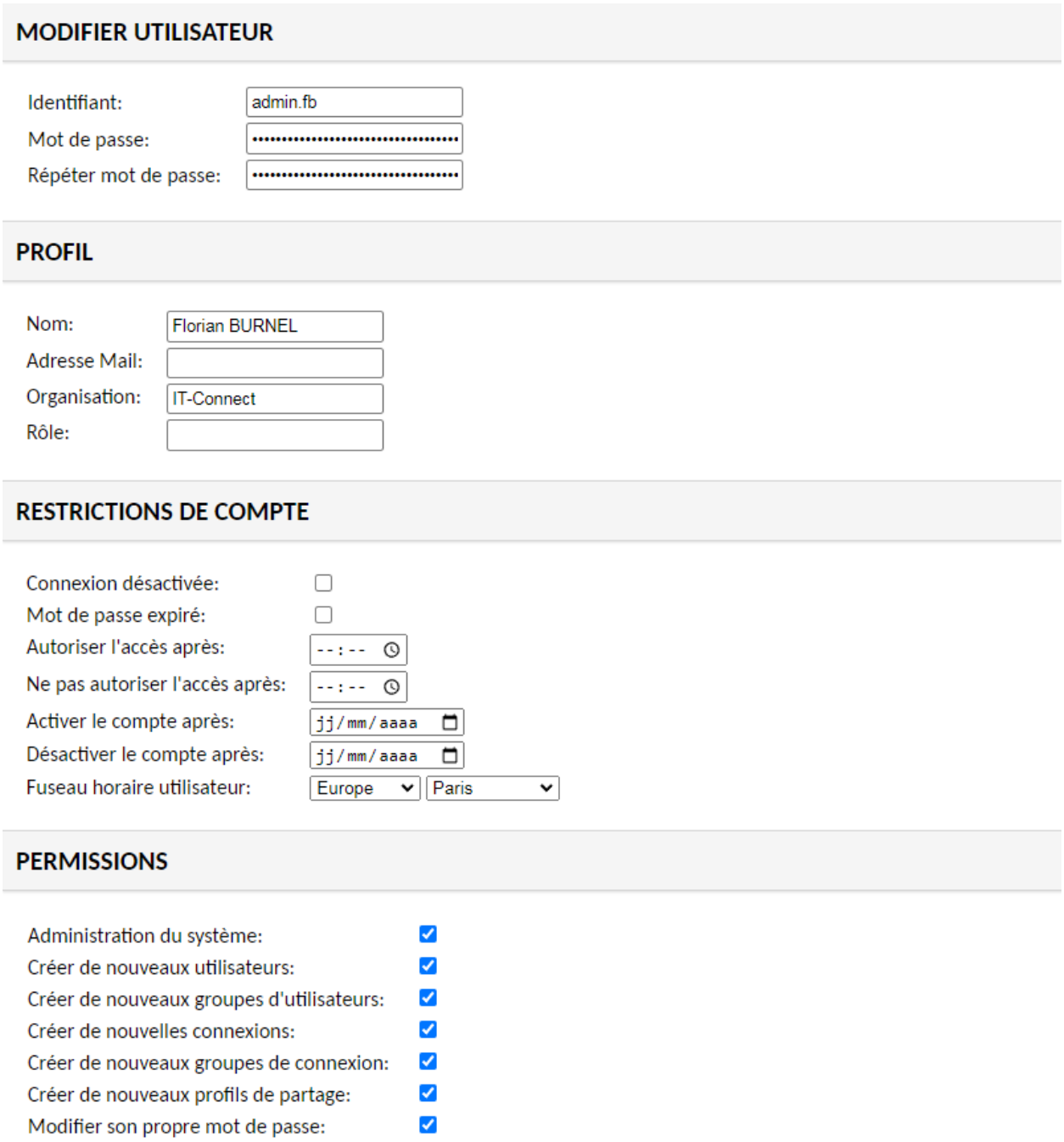
- Créer un nouveau compte administrateur (avec un nom personnalisé)
- Se déconnecter du compte "**guacadmin**"
- Se reconnecter avec le nouveau compte administrateur
- Supprimer le compte "**guacadmin**" par défaut (ou à minima changer son mot de passe et le désactiver)

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

Ensuite, sur l'onglet "**Utilisateurs**" et sur "**Nouvel utilisateur**".



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



Même si cela ne s'applique pas à notre utilisateur actuel, on remarque des options intéressantes dans la section "**Restrictions de compte**". On peut **limiter l'accès aux sessions uniquement sur certaines plages horaires**, mais aussi **activer et désactiver le compte à une date spécifique**. Très intéressant pour donner un accès à un prestataire tout en gardant le contrôle sur les sessions.

Ce nouveau compte est créé, donc suivez les étapes évoquées ci-dessus pour vous débarrasser du compte guacadmin. Tout se passe dans les paramètres puis dans la section "**Utilisateurs**".

Note : Apache Guacamole permet aussi de créer des groupes pour faciliter la gestion des autorisations.

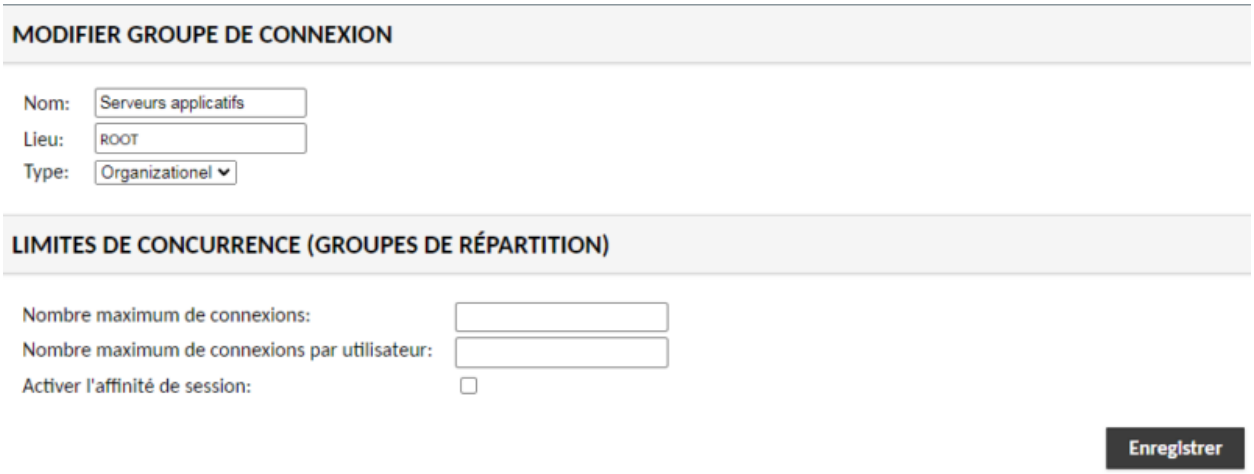
B. Ajouter une connexion RDP

Nous allons créer notre première connexion dans Apache Guacamole, de manière à **se connecter à un serveur en RDP** ! Pour créer une connexion avec un autre protocole tel que SSH, le principe reste le même.

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

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

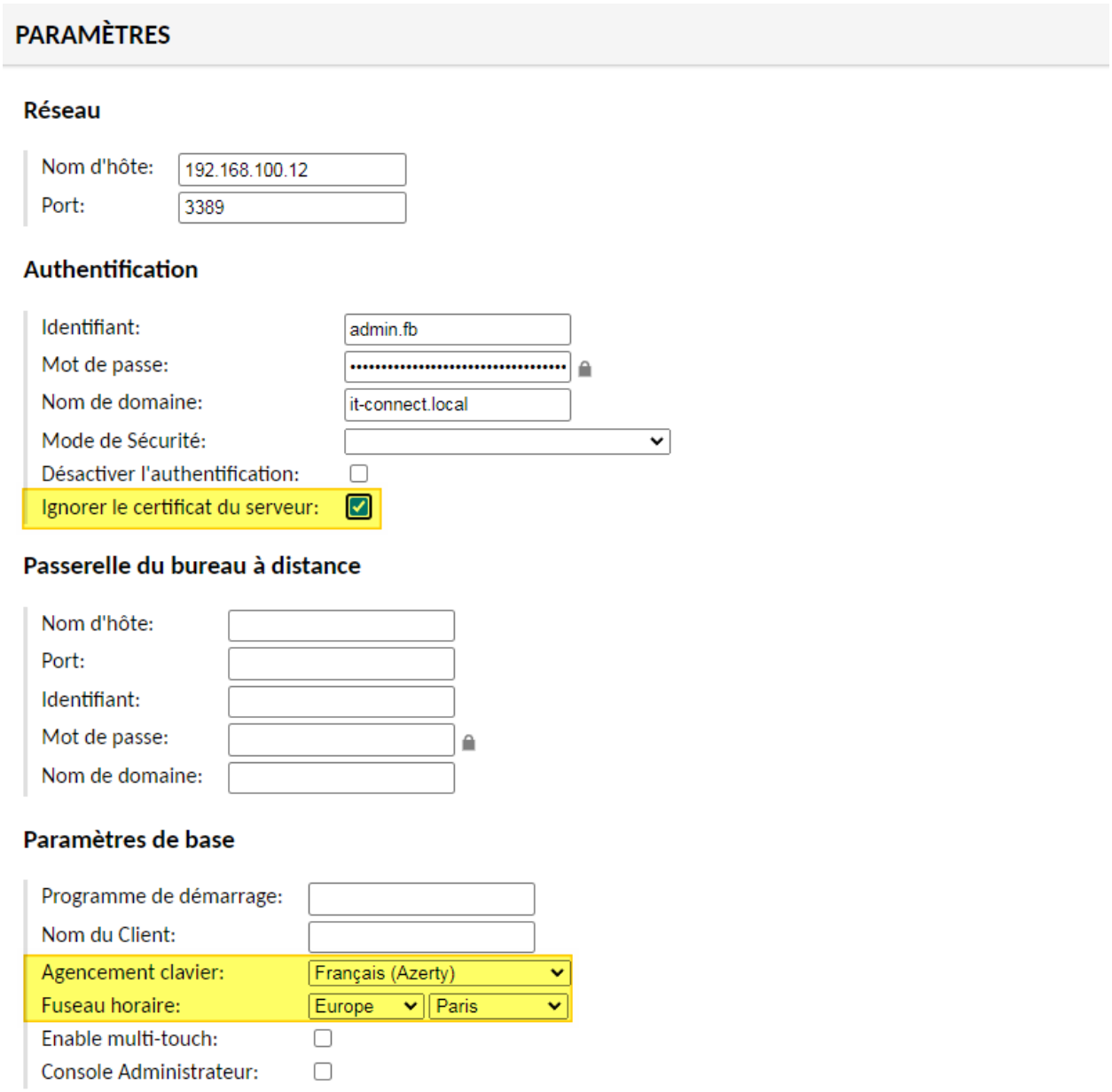
Dans cet exemple, je crée un groupe nommé "**Serveurs applicatifs**". Il sera positionné sous le lieu "**ROOT**" qui est la racine de l'arborescence. Le type de groupe "**Organizationel**" doit être sélectionné pour tous les groupes qui ont pour vocation à organiser les connexions.



On enregistre et on clique sur le bouton "**Nouvelle connexion**". On commence par nommer la connexion, choisir le groupe et le protocole. Ici, c'est sur le serveur "**SRV-APPS**" que je souhaite me connecter, associé à l'adresse IP "**192.168.100.12**".

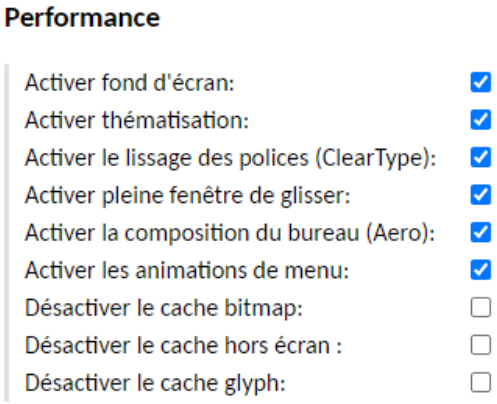
Ensuite, il y a un ensemble de paramètres à renseigner :

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



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

Pour que l'expérience soit un peu plus agréable, on peut cocher les options ci-dessous (mais cela reste facultatif - *testez avec et sans*) :



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

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

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

Voilà, je suis connecté en **Bureau à distance** via le protocole **RDP** à mon serveur en passant par mon serveur Apache Guacamole ! **Si ça ne fonctionne pas, lisez la partie suivante de cet article.** Voici un exemple de connexion RDP via Guacamole, sur une machine sous **Windows Server 2025**.

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

Ce qui permettra aussi d'**ouvrir plusieurs connexions** en même temps et de passer de l'une à l'autre.

C. Apache Guacamole : erreur de connexion en RDP

Que faire si la connexion RDP ne se lance pas ou qu'elle affiche une erreur ?

Retournez sur la ligne de commande de votre serveur et **vérifiez les dernières lignes de logs** qui s'affichent lorsque l'on regarde le statut du service guacd :

sudo systemctl status guacd

Par exemple, on peut trouver ceci :

juin 14 20:15:29 srv-guacamole guacd[31120]: Certificate validation failed

juin 14 20:15:29 srv-guacamole guacd[31120]: RDP server closed/refused connection: SSL/TLS connection failed (untrusted/self-signed certificate?)

Si le certificat RDP ne peut pas être vérifié (auto-signé par exemple) et que l'option "**Ignorer le certificat du serveur**" n'est pas cochée dans les paramètres de la connexion Guacamole, alors cette erreur se produira.

Une autre erreur que vous pourriez rencontrer **si vous avez besoin d'établir des connexions en RDP**, c'est celle-ci :

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

Ce problème est lié au compte utilisateur "**daemon**" utilisé par défaut pour exécuter le service "**guacd**". Vous pouvez le vérifier avec cette commande :

sudo ps aux | grep -v grep| grep guacd

Résultat :

daemon   31513 0.0 0.7 247928 15400 ?   Ss   16:03   0:00 /usr/local/sbin/guacd -f

Nous devons **créer un nouvel utilisateur**, lui associer **les permissions** nécessaires sur les données d'Apache Guacamole, puis **mettre à jour le service** et enfin le **relancer**.

Voici la série de commandes à exécuter, dans l'ordre :

sudo useradd -M -d /var/lib/guacd/ -r -s /sbin/nologin -c "Guacd User" guacd

sudo mkdir /var/lib/guacd

sudo chown -R guacd: /var/lib/guacd

sudo sed -i 's/daemon/guacd/' /etc/systemd/system/guacd.service

sudo systemctl daemon-reload

sudo systemctl restart guacd

Puis, vérifiez l'état du service :

sudo systemctl status guacd

Si c'est bon, vous pouvez tenter une nouvelle connexion RDP.

C. Historique des connexions

Dans les paramètres, la section "**Historique**" permet de visualiser l'ensemble des connexions de tous les utilisateurs Guacamole !

On peut savoir quel utilisateur Guacamole a utilisé quelle session, quand, pendant combien de temps et depuis quelle adresse IP source.

Note : la section "**Sessions Actives**" peut être utilisée comme "kill switch" pour **fermer une ou plusieurs sessions en cours.**

V. Améliorer son installation d'Apache Guacamole

A. 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. Ainsi, lorsqu'un utilisateur va se connecter à Apache Guacamole, il devra configurer ce second facteur d'authentification via une application comme **Microsoft Authenticator, Google Authenticator, FreeOTP, etc...** Attention, ceci s'applique lors de l'authentification à Guacamole, pas lors de l'utilisation d'une connexion (donc à valider une fois).

Puisque Guacamole donne accès aux serveurs de l'entreprise, il me semble **indispensable de mettre en place le MFA**.

Toujours à partir du dépôt officiel d'Apache Guacamole, on va récupérer l'extension "**guacamole-auth-totp**". On télécharge le fichier dans **/tmp** :

```
cd /tmp
wget https://downloads.apache.org/guacamole/1.5.5/binary/guacamole-auth-totp-1.5.5.tar.gz
```

On décompresse l'archive :

```
tar -xzf guacamole-auth-totp-1.5.5.tar.gz
```

Puis, on déplace le fichier "**.jar**" de l'extension dans le répertoire "**extensions**" de Guacamole :

sudo mv guacamole-auth-totp-1.5.5/guacamole-auth-totp-1.5.5.jar /etc/guacamole/extensions/

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

sudo nano /etc/guacamole/guacamole.properties

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

- Apache Guacamole - Paramètres TOTP

Dans le fichier, on va déclarer 4 paramètres :

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

Ce qui donne cette configuration (si l'on passe en SHA256, cela ne fonctionne pas avec Microsoft Authenticator) :

TOTP

totp-issuer: Guacamole IT-Connect

totp-digits: 6

totp-period: 30

totp-mode: sha1

On enregistre, on ferme le fichier et on termine par redémarrer Tomcat pour appliquer les modifications :

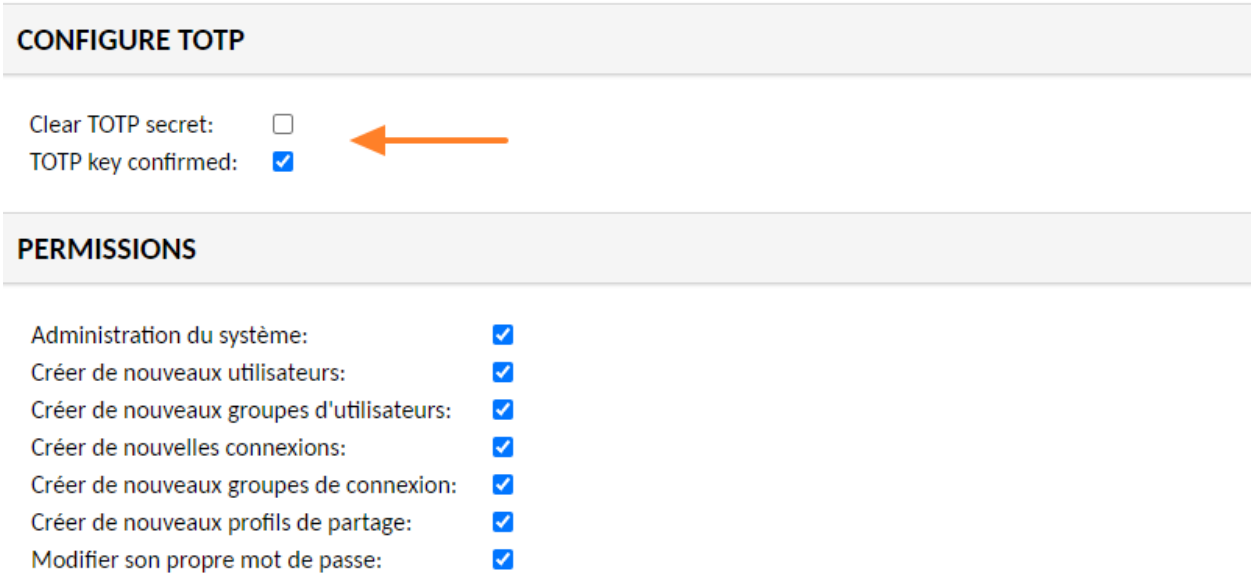
sudo systemctl restart tomcat9

La suite se passe sur l'interface Web.

On se reconnecte avec son compte, et là, on est directement invité à configurer le MFA !

C'est à ce moment-là qu'il faut dégainer **son application sur mobile de manière à scanner le QR Code** puis à finaliser la configuration du MFA sur son compte Guacamole.

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



Voilà l'authentification à deux facteurs est en place sur le serveur Guacamole !

B. Créer un enregistrement vidéo des sessions

Apache Guacamole est capable de **créer un enregistrement vidéo pour chaque session**, avec la possibilité d'activer cette option uniquement sur certaines connexions. Ainsi, cet enregistrement vidéo permet de savoir exactement qui a fait quoi puisque l'on a **le replay de la session**.

Cette fonction s'appuie sur l'extension "**guacamole-history-recording-storage**" qui s'installe sur le même principe que les autres extensions. La procédure qui suit reprend les conseils de la documentation officielle.

- Apache Guacamole - Documentation - Enregistrement des sessions

On commence par télécharger l'archive tar.gz d'Apache Guacamole :

```
cd /tmp
wget https://downloads.apache.org/guacamole/1.5.2/binary/guacamole-history-recording-storage-1.5.5.tar.gz
```

Puis, on décompresse l'archive tar.gz :

```
tar -xzf guacamole-history-recording-storage-1.5.5.tar.gz
```

On déplace le fichier .jar de l'extension vers le répertoire "**extensions**" de Guacamole :

```
sudo mv guacamole-history-recording-storage-1.5.5/guacamole-history-recording-storage-1.5.5.jar /etc/guacamole/extensions/
```

On termine par relancer le service Tomcat9 pour relancer la Web App et charger la nouvelle extension.

sudo systemctl restart tomcat9

L'extension est intégrée à Apache Guacamole.

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

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

Puis, on définit les autorisations sur ce répertoire :

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

sudo chmod 2750 /var/lib/guacamole/recordings

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

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

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

- **Chemin de l'enregistrement** :

${HISTORY_PATH}/${HISTORY_UUID}

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

- **Nom de l'enregistrement** :

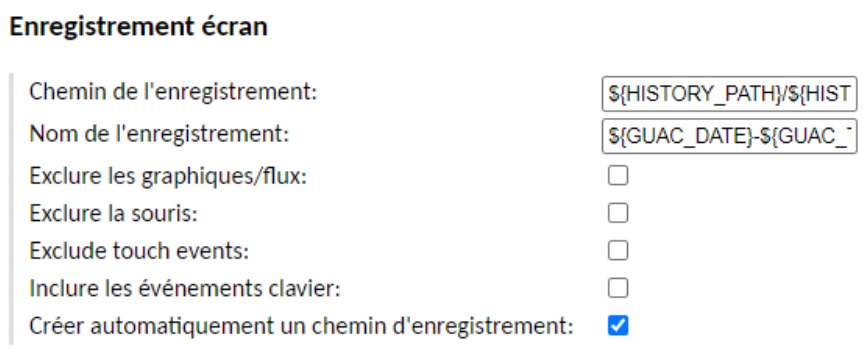
${GUAC_DATE}-${GUAC_TIME} - RDP - ${GUAC_USERNAME}

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

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

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

Ce qui donne :



On enregistre la configuration et **on se connecte au serveur en question pour créer un enregistrement**... On ferme la session... On retourne sur Guacamole.

Au sein des paramètres, on se rend sur "**Historique**". Là, on peut constater que la colonne "**Logs**" intègre le bouton "**View**" lorsqu'un enregistrement est disponible.

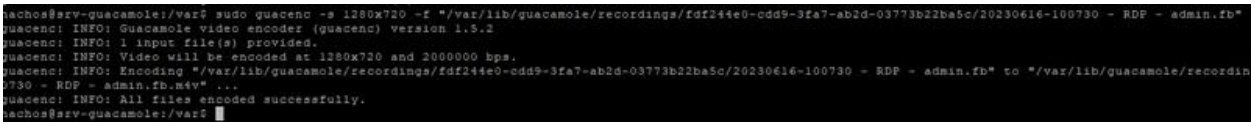
On clique sur "**View**" et là, c'est magique, on peut visualiser le replay de notre session !

Si l'on souhaite **exporter un enregistrement**, il faut le **convertir en ligne de commande** au préalable. En effet, le format de base n'est pas lisible directement.

Apache Guacamole intègre l'outil "**guacenc**" prévu à cet effet pour **créer un fichier vidéo au format M4V**. Pour convertir un enregistrement en fichier de sortie de qualité HD, on utilisera cette commande :

sudo guacenc -s <résolution> -f <fichier à convertir>

sudo guacenc -s 1280x720 -f "/var/lib/guacamole/recordings/fdf244e0-cdd9-3fa7-ab2d-03773b22ba5c/20230616-100730 - RDP - admin.fb"



Ensuite, nous n'avons plus qu'à **transférer le fichier vidéo vers notre PC, via SFTP** par exemple, et à le lire avec un lecteur vidéo. Celui intégré à Windows peut lire le fichier vidéo, sinon on peut aussi utiliser VLC.

À vous les soirées « Guacamole Replay » !

C. Plusieurs utilisateurs vers un même serveur

Comme nous l'avons vu dans cet exemple, lorsque l'on crée une connexion, on spécifie le nom d'utilisateur, son mot de passe et son domaine. Cela signifie que pour une même machine, **on peut avoir plusieurs connexions pour différents utilisateurs.**

Soyons honnêtes, cela peut vite devenir ingérable s'il y a beaucoup de serveurs et plusieurs personnes au service informatique... La bonne nouvelle, c'est que l'on peut faire autrement.

Tout d'abord, on peut utiliser deux variables pour que le nom de l'utilisateur et le mot de passe soient remplacés par ceux du compte Guacamole de l'utilisateur. Pour cela, on utilise ces deux variables :

${GUAC_USERNAME}

${GUAC_PASSWORD}

Concrètement, la partie "**Authentification**" de la connexion sera configurée de cette façon :

Pour que cela fonctionne, **il faut que les identifiants soient identiques entre le compte Guacamole et le compte accepté par l'hôte distant**. Pour que ce soit vraiment efficace, il faudrait mettre en place l'authentification LDAP (Active Directory) pour faire du SSO.

L'autre possibilité est simple : on peut aussi ne rien renseigner pour les champs "**Identifiant**" et "**Mot de passe**", tout en laissant le domaine. Dans ce cas, les informations seront à saisir au moment de la connexion. On pourrait aussi mettre l'identifiant, mais pas le mot de passe... C'est assez flexible, en fait.

VI. Conclusion

Suite à la lecture de ce tutoriel, vous êtes en mesure de **mettre en place Apache Guacamole en tant que bastion pour administrer vos machines Windows, Linux, etc.** ! Avec le MFA et l'enregistrement des sessions en bonus. Reste à bien prendre en main l'outil pour déclarer les utilisateurs, faire des groupes, etc.

Si ce sujet intéresse la communauté, j'essaierai de vous proposer un article avec l'ajout d'un reverse proxy pour publier Apache Guacamole en HTTPS, avec un certificat valide.

En complément de ce tutoriel, voici le lien vers la documentation officielle :

- Apache Guacamole - Documentation

Sur IT-Connect, retrouvez d'autres tutoriels sur Apache Guacamole :

- Configurer l'authentification Active Directory pour Apache Guacamole
- Apache Guacamole avec un reverse proxy basé sur pfSense, HAProxy et Let's Encrypt
- Apache Guacamole avec un reverse proxy basé sur Apache