

Installation d'un serveur Guacamole

- [Guacamole](#)

Guacamole

Installation de Guacamole

Déjà il faut effectuer une mise à jour des paquets :

```
apt update
```

Ensuite il faut installer les paquets nécessaires à l'installation de guacamole :

```
apt install -y build-essential \  
libcairo2-dev \  
libjpeg62-turbo-dev \  
libpng-dev \  
libtool-bin \  
libosspp-uuid-dev \  
libavutil-dev \  
libswscale-dev \  
    freerdp2-dev \  
libpango1.0-dev \  
libssh2-1-dev \  
libvncserver-dev \  
libtelnet-dev \  
libwebsockets-dev \  
libssl-dev \  
libvorbis-dev \  
libwebp-dev \  
libpulse-dev \  
    sudo \  
    vim
```

Pour le bien des commande suivante nous allons définir une variable d'environnement :

```
VER=1.5.3
```

Une fois cela fait, nous pouvons télécharger guacamole-server :

```
wget https://downloads.apache.org/guacamole/$VER/source/guacamole-server-$VER.tar.gz
```

Il faut ensuite extraire l'archive :

```
tar xzf guacamole-server-$VER.tar.gz
```

Il faut ensuite se déplacer dans le dossier :

```
cd guacamole-server-$VER
```

Il faut ensuite configurer guacamole-server :

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

Il faut ensuite compiler guacamole-server :

```
make CFLAGS="-Wno-discarded-qualifiers"  
sudo make install
```

Nous pouvons ensuite créer les différents liens symboliques :

```
ln -s /usr/local/bin/guacd /usr/bin/guacd
```

Il faut ensuite recharger les fichiers de configuration :

```
systemctl daemon-reload
```

Puis il faut activer le service :

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

Nous pouvons ensuite vérifier que le service est bien actif :

```
systemctl status guacd
```

```

root@danette:~# systemctl status guacd
● guacd.service - Guacamole Server
   Loaded: loaded (/etc/systemd/system/guacd.service; enabled; preset: enabled)
   Active: active (running) since Thu 2023-10-19 20:04:47 CEST; 32min ago
     Docs: man:guacd(8)
    Main PID: 626 (guacd)
      Tasks: 1 (limit: 2268)
     Memory: 14.1M
        CPU: 165ms
    CGroup: /system.slice/guacd.service
            └─626 /usr/local/sbin/guacd -f

oct. 19 20:04:47 danette systemd[1]: Started guacd.service - Guacamole Server.
oct. 19 20:04:48 danette guacd[626]: Guacamole proxy daemon (guacd) version 1.5.3 started
oct. 19 20:04:48 danette guacd[626]: guacd[626]: INFO: Guacamole proxy daemon (guacd) version 1.5.3 started
oct. 19 20:04:48 danette guacd[626]: guacd[626]: INFO: Listening on host 0.0.0.0, port 4822
oct. 19 20:04:48 danette guacd[626]: Listening on host 0.0.0.0, port 4822

```

Installation de tomcat

Il faut commencer par ajouter le dépôt de tomcat dans le fichier `/etc/apt/sources.list` :

```
echo "deb http://deb.debian.org/debian/ bullseye main" > /etc/apt/sources.list.d/bullseye.list
```

Il faut ensuite mettre à jour les paquets :

```
apt update
```

Nous pouvons alors installer tomcat :

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

On peut ensuite vérifier que tomcat est bien installé :

```
systemctl status tomcat9.service
```

```

root@danette:~# systemctl status tomcat9.service
● tomcat9.service - Apache Tomcat 9 Web Application Server
   Loaded: loaded (/lib/systemd/system/tomcat9.service; enabled; preset: enabled)
   Active: active (running) since Thu 2023-10-19 20:05:03 CEST; 38min ago
     Docs: https://tomcat.apache.org/tomcat-9.0-doc/index.html
   Process: 632 ExecStartPre=/usr/libexec/tomcat9/tomcat-update-policy.sh (code=exited, status=0/SUCCESS)
    Main PID: 647 (java)
      Tasks: 39 (limit: 2268)
     Memory: 347.6M
        CPU: 42.512s
    CGroup: /system.slice/tomcat9.service
            └─647 /usr/lib/jvm/default-java/bin/java -Djava.util.logging.config.file=/var/lib/tomcat9/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.

oct. 19 20:05:03 danette tomcat9[647]: 20:05:03.777 [main] INFO o.a.g.a.mysql.conf.MySQLEnvironment - Installed JDBC driver for MySQL/MariaDB detected as "MySQL Connector
oct. 19 20:05:05 danette tomcat9[647]: 20:05:05.345 [main] INFO o.a.g.extension.ExtensionModule - Extension "MySQL Authentication" (mysql) loaded.
oct. 19 20:05:05 danette tomcat9[647]: 20:05:05.594 [main] INFO o.a.g.t.w.WebSocketTunnelModule - Loading JSR-356 WebSocket support...
oct. 19 20:05:06 danette tomcat9[647]: 20:05:06.573 [main] WARN o.g.jersey.server.wadl.WadlFeature - JAXBContext implementation could not be found. WADL feature is disabl
oct. 19 20:05:07 danette tomcat9[647]: Le déploiement de l'archive de l'application web [/var/lib/tomcat9/webapps/guacamole.war] s'est terminé en [11 240] ms
oct. 19 20:05:07 danette tomcat9[647]: Déploiement du répertoire d'application web [/var/lib/tomcat9/webapps/ROOT]
oct. 19 20:05:08 danette tomcat9[647]: Au moins un fichier JAR a été analysé pour trouver des TLDs mais il n'en contenait pas, le mode "debug" du journal peut être activé
oct. 19 20:05:08 danette tomcat9[647]: Le déploiement du répertoire [/var/lib/tomcat9/webapps/ROOT] de l'application web s'est terminé en [901] ms
oct. 19 20:05:08 danette tomcat9[647]: Démarrage du gestionnaire de protocole ["http-nio-8080"]
oct. 19 20:05:08 danette tomcat9[647]: Le démarrage du serveur a pris [15870] millisecondes
lines 1-22/22 (END)

```

Il faut également ouvrir le port 8080 :

```
ufw allow 8080/tcp
```

Installation de guacamole-client

Il faut commencer par créer un dossier de configuration :

```
mkdir /etc/guacamole
```

Il faut ensuite télécharger guacamole-client :

```
wget \
https://downloads.apache.org/guacamole/$VER/binary/guacamole-$VER.war \
-O /etc/guacamole/guacamole.war
```

Nous allons créer un lien symbolique de guacamole-client dans le dossier webapps de tomcat :

```
ln -s /etc/guacamole/guacamole.war /var/lib/tomcat9/webapps/
```

Il faut ensuite redémarrer tomcat et guacamole :

```
systemctl restart tomcat9 guacd
```

Configuration de apache guacamole

Il faut commencer par créer un fichier de configuration :

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

Il faut ensuite définir la variable d'environnement du répertoire home de guacamole :

```
echo "GUACAMOLE_HOME=/etc/guacamole" >> /etc/default/tomcat9
```

Nous pouvons alors créer le fichier de configuration `guacamole.properties` dans le dossier `/etc/guacamole` :

```
cat > /etc/guacamole/guacamole.properties << EOL
guacd-hostname: 127.0.0.1
guacd-port: 4822
```

```
user-mapping: /etc/guacamole/user-mapping.xml
auth-provider: net.sourceforge.guacamole.net.basic.BasicFileAuthenticationProvider
EOL
```

Ensuite nous pouvons créer un lien symbolique du fichier de configuration dans le dossier `/usr/share/tomcat9/.guacamole` :

```
ln -s /etc/guacamole /usr/share/tomcat9/.guacamole
```

Nous pouvons ensuite redémarrer tomcat et guacamole :

```
systemctl restart tomcat9 guacd
```

Installation et configuration de mariadb

Nous commençons par installer le paquet MariaDB Server :

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

Puis il nous faut sécuriser l'installation :

```
sudo mysql_secure_installation
```

Une fois ceci fait, vous devrez répondre à une série de questions.

Une fois cette étape effectuée, nous pouvons nous connecter à notre instance MariaDB :

```
mysql -u root -p
```

Nous allons ensuite créer la base de données `guacamole_db`, l'utilisateur `guacamole_user` ainsi que son mot de passe et nous lui donnons les droits sur la base de données `guacamole_db` :

```
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;
```

Les étapes suivantes consistent à ajouter l'extension mysql à apache guacamole ainsi que le connecteur correspondant :

```
cd /tmp
wget https://downloads.apache.org/guacamole/1.5.2/binary/guacamole-auth-jdbc-1.5.2.tar.gz
```

Il faut ensuite extraire l'archive :

```
tar -xzf guacamole-auth-jdbc-1.5.2.tar.gz
```

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

```
sudo mv guacamole-auth-jdbc-1.5.2/mysql/guacamole-auth-jdbc-mysql-1.5.2.jar /etc/guacamole/extensions/
```

Il faut ensuite télécharger le connecteur mysql :

```
cd /tmp
wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-j-8.0.33.tar.gz
```

Il faut ensuite extraire l'archive :

```
tar -xzf mysql-connector-j-8.0.33.tar.gz
```

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

```
sudo cp mysql-connector-j-8.0.33/mysql-connector-j-8.0.33.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-1.5.2/mysql/schema/`. Le mot de passe root de MariaDB doit être saisi pour effectuer l'import.

```
cd guacamole-auth-jdbc-1.5.2/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
```

On ajoute les lignes suivantes :

```
# 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.

Il faut également éditer 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 contenu à ajouter :

```
[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
```

En vous rendant sur l'interface web de Guacamole, vous devriez pouvoir vous connecter avec les identifiants par défaut (***guacadmin/guacadmin***).



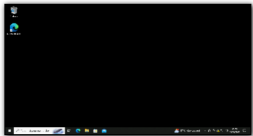
APACHE GUACAMOLE

Se connecter

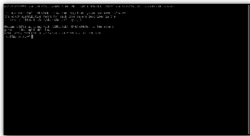
Vous pouvez alors créer un nouvel utilisateur ***nimda/45keuros*** et lui donner les droits d'administration. Ensuite vous pouvez vous connecter avec cet utilisateur et supprimer l'utilisateur guacadmin.

Vous arrivez alors sur la page d'accueil de guacamole :

CONNEXIONS RÉCENTES





Windows 10



Debian 12

TOUTES LES CONNEXIONS

 Debian 12

 Windows 10

🔍

Filtre

Les deux machines seront ajoutées dans la partie configuration.

Configuration administrateur de guacamole

Pour commencer nous allons créer notre nouvel utilisateur, pour cela il nous faut nous rendre dans l'onglet `paramètres` :

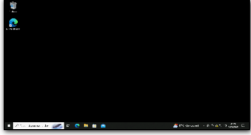
CONNEXIONS RÉCENTES


guacadmin

Accueil

Paramètres

Se déconnecter





Windows 10

Debian 12

TOUTES LES CONNEXIONS

🔍 Filtre

➤ Debian 12

🖥 Windows 10

Nous allons ensuite dans l'onglet `utilisateurs` :

PARAMÈTRES

guacadmin

Sessions Actives

Historique

Utilisateurs

Groupes

Connexions

Préférences

Cette page sera remplie avec des connexions actuellement actives. Les connexions répertoriées et la possibilité de supprimer ces connexions dépendent de votre niveau d'accès. Si vous souhaitez en fermer une ou plusieurs, sélectionner les et cliquer sur "Fermer Sessions".

Fermer Sessions

🔍 Filtre

	Identifiant ▼	Ouvert depuis	Hôte distant	Nom de connexion
--	---------------	---------------	--------------	------------------

Pas de session ouverte

Puis nous cliquons sur `ajouter un utilisateur` et nous saisissons les informations importantes et nous n'oublions pas de cocher toutes les cases afin de donner tous les droits à l'utilisateur :

MODIFIER UTILISATEUR



Identifiant:	<input type="text" value="nimda"/>
Mot de passe:	<input type="password" value="....."/>
Répéter mot de passe:	<input type="password" value="....."/>

PERMISSIONS

Administration du système:	<input checked="" type="checkbox"/>
Créer de nouveaux utilisateurs:	<input checked="" type="checkbox"/>
Créer de nouveaux groupes d'utilisateurs:	<input checked="" type="checkbox"/>
Créer de nouvelles connexions:	<input checked="" type="checkbox"/>
Créer de nouveaux groupes de connexion:	<input checked="" type="checkbox"/>
Créer de nouveaux profils de partage:	<input checked="" type="checkbox"/>
Modifier son propre mot de passe:	<input checked="" type="checkbox"/>

CONNEXIONS

Connexions en cours Toutes les Connexions

- ☐  Debian 12
- ☐  Windows 10

Enregistrer

Annuler

Il nous suffit ensuite de cliquer sur **Enregistrer** et notre utilisateur est créé.

Nous pouvons alors nous déconnecter du compte **guacadmin** et nous connecter avec notre nouvel utilisateur.

Ainsi nous pouvons supprimer l'utilisateur **guacadmin** toujours dans l'onglet **Paramètres -> Utilisateurs**
-> **guacadmin** :

Connexion désactivée: ☐
Mot de passe expiré: ☐
Autoriser l'accès après: -- :-- ⌚
Ne pas autoriser l'accès après: -- :-- ⌚
Activer le compte après: jj/mm/aaaa 📅
Désactiver le compte après: jj/mm/aaaa 📅
Fuseau horaire utilisateur:

PERMISSIONS

Administration du système: ☒
Créer de nouveaux utilisateurs: ☒
Créer de nouveaux groupes d'utilisateurs: ☒
Créer de nouvelles connexions: ☒
Créer de nouveaux groupes de connexion: ☒
Créer de nouveaux profils de partage: ☒
Modifier son propre mot de passe: ☒

CONNEXIONS

🔍 Filtre

Connexions en cours Toutes les Connexions

🗖️ ➡️ ☐ Debian 12
🗖️ 🖥️ ☐ Windows 10

Enregistrer Cloner Annuler Supprimer

Ensuite nous pouvons ajouter nos deux machines dans l'onglet Paramètres -> Connexions -> Nouvelle Connexion :

PARAMÈTRES

👤 nimda ▾

Sessions Actives Historique Utilisateurs Groupes Connexions Préférences

Cliquer ou appuyer sur une connexion en dessous pour la gérer. Selon vos permissions, les connexions peuvent être ajoutées, supprimées, leur propriétés (protocole, nom d'hôte, port, etc) changées.

🗖️ ➡️ Nouvelle Connexion 🗖️ ➡️ Nouveau Groupe

🔍 Filtre

🗖️ ➡️ Debian 12
🗖️ 🖥️ Windows 10

Debian 12

Déjà nous allons commencer par les informations de base :

MODIFIER CONNEXION

Nom:
Lieu:
Protocole:

Nous nous connectons à notre debian via le protocole `SSH`, donc dans la liste déroulante `Protocole` nous sélectionnons `SSH`.

Ensuite il y a les paramètres de connexion :

PARAMÈTRES

Réseau

Nom d'hôte:	<input type="text" value="192.168.220.129"/>
Port:	<input type="text" value="2330"/>
Clé publique de l'hôte (Base64):	<input type="text"/>

Authentification

Identifiant:	<input type="text" value="root"/>
Mot de passe:	<input type="password" value="....."/> 
Clé privée:	<div><div></div></div>
Phrase secrète:	<input type="password"/> 

Le nom d'hôte est l'adresse IP de notre debian, le port est le port SSH par défaut est le port 22 mais nous pouvons le changer si nous le souhaitons avec la commande `sudo nano /etc/ssh/sshd_config` et nous modifions la ligne `#Port 22` en `Port 2330` par exemple.

Ensuite nous avons les paramètres d'authentification. - *Nom d'utilisateur* : `root` - *Mot de passe* : `Votre mot de passe`

Puis tout en bas nous pouvons cliquer sur `Enregistrer` et notre première machine est ajoutée.

Windows 10

Nous allons commencer par les informations de base :

MODIFIER CONNEXION

Nom:

Lieu:

Protocole:

Le protocole utilisé pour se connecter à notre machine windows est le **RDP**, donc dans la liste déroulante **Protocole** nous sélectionnons **RDP**.

Ensuite il y a les paramètres de connexion :

PARAMÈTRES


Réseau

Nom d'hôte:

Port:

Authentification

Identifiant:

Mot de passe: 

Nom de domaine:

Mode de Sécurité:

Désactiver l'authentification: ☐

Ignorer le certificat du serveur: ☒

Nous configurons le clavier et la langue de notre machine windows et nous activons la console d'administration.

Paramètres de base

Programme de démarrage:	<input type="text"/>
Nom du Client:	<input type="text"/>
Agencement clavier:	Français (Azerty) ▾
Fuseau horaire:	Europe ▾ Paris ▾
Enable multi-touch:	<input type="checkbox"/>
Console Administrateur:	<input checked="" type="checkbox"/>

Puis nous pouvons enregistrer notre machine.

Pour que windows RDP fonctionne il faudra faire quelques modifications sur notre debian.

Nous allons créer un compte utilisateur sur notre debian qui sera utilisé pour exécuter le service guacd :

```
useradd -M -d /var/lib/guacd/ -r -s /sbin/nologin -c "Guacd User" guacd
mkdir /var/lib/guacd
chown -R guacd: /var/lib/guacd
```

Ensuite, nous allons mettre à jour l'utilisateur du service guacd :

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

Ensuite nous allons redémarrer le démon et le service guacd :

```
systemctl daemon-reload
systemctl restart guacd
```