

Installation native de Guacamole

- Lien : <https://guacamole.apache.org/doc/gug/guacamole-native.html>
- <https://guacamole.apache.org/releases/1.6.0/>
- <https://www.it-connect.fr/tuto-apache-guacamole-bastion-rdp-ssh-debian/>

Présentation

Guacamole est divisé en deux parties :

- **guacamole-server** :
 - fournit le proxy `guacd` et les bibliothèques associées,
 - doit être compilé à partir du code source.
- **guacamole-client** :
 - fournit le client à servir par votre conteneur de servlets, généralement Apache Tomcat.
 - disponible sous forme binaire.

guacamole-server

- contient tous les composants natifs côté serveur nécessaires par Guacamole pour se connecter aux bureaux à distance.
- fournit une bibliothèque C commune, **libguac**, dont dépendent tous les autres composants natifs, ainsi que des bibliothèques pour chaque protocole supporté,
- **guacd**, le cœur de Guacamole.

`guacd` est le démon proxy qui s'exécute sur le serveur Guacamole :

- accepte les utilisateurs des connexions qui sont tunnelées via l'application web Guacamole,
- puis se connecte aux bureaux à distance en leur nom.

`guacd` est un exécutable appelé `GUACD` qui peut être exécuté manuellement ou automatiquement quand votre ordinateur démarre.

La construction de `guacamole-server` nécessite un compilateur C (comme `gcc`) et des bibliothèques dont dépend `guacamole-server`. Certaines dépendances sont absolument obligatoire, tandis que d'autres sont optionnels. La présence de dépendances optionnelles permet des fonctionnalités supplémentaires.

De nombreuses distributions Linux séparent les paquets de bibliothèques en binaires et des paquets « développement » ; Vous devrez installer les packages de développement. Celles-ci se terminent généralement par un suffixe « -dev » ou « -dev ».

Installation de Guacamole Server

Installer les dépendances indispensables

Pour construire `guacamole-server`, il faut disposer de :

- **Cairo** : utilisé par `libguac` pour le rendu graphique,
- **libjpeg** (ou `libjpeg-turbo`) : utilisé par `libguac` pour fournir la prise en charge JPEG
- **libpng** : utilisé par `libguac` pour écrire des images PNG, le type d'image de base utilisé par le protocole Guacamole
- **libuuid** (ou la bibliothèque OSSP UUID) : utilisé pendant le processus de compilation. `libtool` crée des bibliothèques compilées nécessaire pour le `guacamole`.

```
apt install -y libcairo2-dev libjpeg62-turbo-dev libtool-bin
```

Installer les dépendances optionnelles

Les dépendances optionnelles de Guacamole dictent quelles parties de `guacamole-server` seront construites. Cela inclut la prise en charge de divers protocoles de bureau à distance, ainsi que toutes les fonctionnalités supplémentaires de ces protocoles :

- **FreeRDP** : requis pour le support RDP.
- **Pango** : bibliothèque de mise en page de texte que Guacamole utilise pour rendre le texte protocoles nécessitant un terminal (Kubernetes, SSH et telnet).

- **libssh2** : nécessaire pour le support SSH et SFTP.
- **libVNCServer** : fournit libvncclient, qui est nécessaire pour le support VNC
- **libwebsockets** : nécessaire pour le support de Kubernetes.
- **OpenSSL** : offre la prise en charge de SSL et TLS - deux schémas de chiffrement courants qui constituent la majorité du trafic web chiffré.
- **libwebp** : utilisé par libguac pour écrire des images WebP.
- **PulseAudio** (libpulse-dev) : PulseAudio fournit libpulse, utilisé par le support VNC de Guacamole pour fournir un support audio expérimental.
- **Libvorbis** (libvorbis-dev) : prend en charge Ogg Vorbis - une norme libre et ouverte pour Compression sonore. Si elle est installée, libguac sera construite avec le support d'Ogg Vorbis et les protocoles supportant l'audio utiliseront la compression Ogg Vorbis lorsque Possible. Sinon, le son ne sera encodé que comme WAV (non compressé), et ne sera que disponible si votre navigateur prend également en charge le WAV.
- FFmpeg (libavcodec-dev, libavformat-dev, libavutil-dev, libswscale-dev): Les bibliothèques libavcodec, libavformat, libavutil et libswscale fournies par FFmpeg est utilisé pour encoder des flux vidéo lors de la traduction des enregistrements de sessions de Guacamole
- **libtelnet** (libtelnet-dev) : nécessaire pour le support de telnet.

```
apt install -y freerdp2-dev libpango1.0-dev libssh2-1-dev libvncserver-dev libwebsockets-dev libssl-dev libwebp-dev libpulse-dev libvorbis-dev libavcodec-dev libavformat-dev libavutil-dev libswscale-dev libtelnet-dev
```

Debian 13 ne contient que le dépôt freerdp3 alors que Guacamole nécessite freerdp2. Il est nécessaire de compiler freerdp2 en activant le redimensionnement d'images (-DWITH_SWSCALE=ON).

Installer FreeRDP2 si installation sur Debian 13

- installer le support de Wayland
 - <https://www.malekal.com/wayland-serveur-affichage-linux/>

```
sudo apt install -y libwayland-dev wayland-protocols libxkbcommon-dev
```

- Activer le support ALSA (Advanced Linux Sound Architecture) pour la redirection audio :

```
sudo apt install -y libasound2-dev
```

- Activer le support CUPS (Common Unix Printing System) pour la redirection d'imprimante dans les sessions RDP :

```
sudo apt install -y libcups2-dev
```

- Activer le support PCSC (smart card redirection) pour les sessions RDP :

```
sudo apt install -y libpcsclite-dev
```

- Activer le support XKBFile (extension X11 pour la gestion des fichiers de clavier) :

```
sudo apt install -y libxkbfile-dev
```

- Activer le support Xinerama (extension X11 pour la gestion multi-écrans) :

```
sudo apt install -y libxinerama-dev
```

- Activer le support Xcursor (extension X11 pour la gestion des curseurs) :

```
sudo apt install -y libxcursor-dev
```

- Activer le support Xv (extension X11 pour l'accélération vidéo) :

```
sudo apt install -y libxv-dev
```

- Activer le support Xi (extension X11 pour la gestion avancée des périphériques d'entrée) :

```
sudo apt install -y libxi-dev
```

- Activer le support XRandR (extension X11 pour la gestion dynamique des résolutions d'écran) :

```
sudo apt install -y librandr-dev
```

- Activer le support Xfixes (extension X11 qui ajoute des fonctionnalités utiles comme la gestion avancée des curseurs) :

```
sudo apt install -y libxfixes-dev
```

- Génération des man-pages avec le paquet xsltproc.

```
sudo apt install -y docbook-xsl docbook-xml xsltproc
sudo apt install cmake libsystemd-dev clang-format
sudo apt install -y libusb-1.0-0-dev libudev-dev pkg-config

git clone --branch 2.11.7 https://github.com/FreeRDP/FreeRDP.git
cd FreeRDP
?? git submodule update --init --recursive
mkdir build && cd build

# nettoyer le cache CMake
rm -f CMakeCache.txt
rm -rf CMakeFiles/
rm -rf build

# lancer la configuration
cmake -DCMAKE_INSTALL_PREFIX=/usr/local \
  -DWITH_CAIRO=ON \
  -DWITH_SWSCALE=ON \
  -DWITH_MANPAGES=ON \
  -DWITH_ALSA=ON \
  -DWITH_CUPS=ON \
  -DWITH_PCSC=ON \
  -DWITH_XKBFILE=ON \
  -DWITH_XINERAMA=ON \
  -DWITH_XCURSOR=ON \
  -DWITH_XV=ON \
  -DWITH_XI=ON \
  -DWITH_XRANDR=ON \
  -DWITH_XFIXES=ON \
  -DWITH_WAYLAND=ON \
  -Wno-dev \
  ..

make -j$(nproc)
sudo make install
sudo ldconfig
```

Vérifier :

```
/usr/local/bin/xfreerdp --version
```

Autres dépendances optionnelles

- **libtelnet** (libtelnet-dev) : nécessaire pour le support de telnet.

Obtenir le code source

- créer et utiliser un compte pour guacamole

```
apt install sudo
adduser guacamole
usermod -aG sudo guacamole
```

- avec l'archive

```
wget https://dldcn.apache.org/guacamole/1.6.0/source/guacamole-server-1.6.0.tar.gz
tar -xzf guacamole-server-1.6.0.tar.gz
cd guacamole-server-1.6.0
```

Créer guacamole-server

- installer Autotools et plusieurs bibliothèques :

```
sudo apt install -y build-essential make pkg-config
```

- configurer la compilation avec Debian 13

```
./configure --with-systemd-dir=/etc/systemd/system/  
...  
-----  
guacamole-server version 1.6.0  
-----  
  
Library status:  
  
freerdp ..... yes (3.x)  
pango ..... yes  
libavcodec ..... yes  
libavformat ..... yes  
libavutil ..... yes  
libssh2 ..... yes  
libssl ..... yes  
libswscale ..... yes  
libtelnet ..... yes  
libVNCServer ..... yes  
libvorbis ..... yes  
libpulse ..... yes  
libwebsockets ..... yes  
libwebp ..... yes  
wsock32 ..... no  
  
Protocol support:  
  
Kubernetes .... yes  
RDP ..... yes  
SSH ..... yes  
Telnet ..... yes  
VNC ..... yes  
  
Services / tools:  
  
guacd ..... yes  
guacenc .... yes  
guaclog .... yes  
  
FreeRDP plugins: /usr/lib/x86_64-linux-gnu/pkgconfig/../../../../lib/x86_64-linux-gnu/freerdp3  
Init scripts: no  
Systemd units: /etc/systemd/system/  
  
Type "make" to compile guacamole-server.  
  
configure: WARNING:  
-----  
Support for FreeRDP 3.x is currently experimental. Some features, like  
RemoteApp, have known issues. If not testing, consider using FreeRDP 2.x.  
-----
```

- configurer la compilation avec Debian 12

```
./configure --with-systemd-dir=/etc/systemd/system/  
...  
-----  
guacamole-server version 1.6.0  
-----  
  
Library status:  
  
freerdp ..... yes (2.x)  
pango ..... yes  
libavcodec ..... yes  
libavformat ..... yes  
libavutil ..... yes  
libssh2 ..... yes
```

```

libssl ..... yes
libswscale ..... yes
libtelnet ..... yes
libVNCServer ..... yes
libvorbis ..... yes
libpulse ..... yes
libwebsockets ..... yes
libwebp ..... yes
wsock32 ..... no

```

Protocol support:

```

Kubernetes .... yes
RDP ..... yes
SSH ..... yes
Telnet ..... yes
VNC ..... yes

```

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.

- Compiler **guacamole-server** avec **make** :

```

make
make all-recursive
make[1]: Entering directory '/home/guacamole/guacamole-server-1.6.0'
Making all in src/libguac
...
make[2]: Leaving directory '/home/guacamole/guacamole-server-1.6.0/src/guaclog'
make[2]: Entering directory '/home/guacamole/guacamole-server-1.6.0'
make[2]: Leaving directory '/home/guacamole/guacamole-server-1.6.0'
make[1]: Leaving directory '/home/guacamole/guacamole-server-1.6.0'

```

Installation de guacamole

* commande ****sudo make install**** pour installer les composants qui ont été créés.

```

$ sudo make install
Making install in src/libguac
make[1]: Entering directory '/home/guacamole/guacamole-server-1.6.0/src/libguac'
Making install in .
...
-----
Libraries have been installed in:
  /usr/local/lib

```

If you ever happen to want to link against installed libraries in a given directory, LIBDIR, you must either use libtool, and specify the full pathname of the library, or use the '-LLIBDIR' flag during linking and do at least one of the following:

- add LIBDIR to the 'LD_LIBRARY_PATH' environment variable during execution
- add LIBDIR to the 'LD_RUN_PATH' environment variable during linking
- use the '-Wl,-rpath -Wl,LIBDIR' linker flag
- have your system administrator add LIBDIR to '/etc/ld.so.conf'

See any operating system documentation about shared libraries for more information, such as the ld(1) and ld.so(8) manual pages.

```
...
make[2]: Leaving directory '/home/guacamole/guacamole-server-1.6.0/src/guaclog'
make[1]: Leaving directory '/home/guacamole/guacamole-server-1.6.0/src/guaclog'
make[1]: Entering directory '/home/guacamole/guacamole-server-1.6.0'
make[2]: Entering directory '/home/guacamole/guacamole-server-1.6.0'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/guacamole/guacamole-server-1.6.0'
make[1]: Leaving directory '/home/guacamole/guacamole-server-1.6.0'
```

- Commande **ldconfig** pour mettre à jour les liens entre **guacamole-server** et les bibliothèques (cette commande ne retourne aucun résultat) :

```
sudo ldconfig
```

Démarre le service guacd

Pour 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
```

- Vérifier le statut d'Apache Guacamole Server :

```
sudo systemctl status guacd
```

Créer le répertoire de configuration

Création de l'arborescence pour la configuration d'Apache Guacamole. Cela va donner le répertoire **etc/guacamole** avec les sous-répertoires **extensions** et **lib**, nécessaire pour mettre en place ensuite le stockage des données dans une base de données MariaDB / MySQL.

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

Installation de Guacamole client

La Web App (guacamole client) est la partie cliente qui nécessite un serveur Apache Tomcat 9. Cependant les dépôts de Debian 12 installent Tomcat 10. Il faut installer le dépôt de Debian 11 sur le serveur Debian 12 afin de pouvoir télécharger les paquets correspondants à Tomcat 9.

- Créez un nouveau fichier source bullseye.list

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

- Ajoutez la ligne suivante :

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

- Mettez à jour le cache des paquets :

```
sudo apt update
```

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

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

- Téléchargez la dernière version de la Web App d'Apache Guacamole depuis le dépôt officiel.

```
wget https://dln.apache.org/guacamole/1.6.0/binary/guacamole-1.6.0.war
```

- Déplacez le fichier téléchargé dans la bibliothèque de Web App de Tomcat9 avec cette commande :

```
sudo mv guacamole-1.6.0.war /var/lib/tomcat9/webapps/guacamole.war
```

- Relancez les services Tomcat9 et Guacamole :

```
sudo systemctl restart tomcat9 guacd
```

- pour démarrer automatiquement Guacamole au boot :

```
systemctl enable tomcat9
systemctl enable guacd
```

Base de données MariaDB pour l'authentification

Utilisation d'une base de données pour stocker toutes les informations de l'application.

- Installez le paquet MariaDB Server :

```
sudo apt install -y mariadb-server
```

- Sécurisez la base de données :

```
sudo mysql_secure_installation
```

- création de la base de données **guacamole_db** pour un utilisateur dédié **guacamole_user**'@'**localhost** pour Apache Guacamole:

```
mysql -u root -p
>CREATE DATABASE guacamole_db;
>CREATE USER 'guacamole_user'@'localhost' IDENTIFIED BY 'P@$w0rd2025Secure';
>GRANT SELECT,INSERT,UPDATE,DELETE ON guacamole_db.* TO 'guacamole_user'@'localhost';
>FLUSH PRIVILEGES;
>EXIT;
```

- Ajoutez l'extension MySQL à Apache Guacamole dans le répertoire `"/etc/guacamole/extensions/"` créé précédemment :

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

- Téléchargez le connecteur JDBC MySQL (utilisable avec MariaDB) doit être téléchargé depuis le lien <http://dev.mysql.com/downloads/connector/j/> (en choisissant **Platform Independent**, puis en cliquant sur le bouton **Download** permettant d'obtenir la **Compressed TAR Archive**.) :

```
wget https://cdn.mysql.com//Downloads/Connector-J/mysql-connector-j-9.5.0.tar.gz
```

* Décompression de l'archive Puis, puis déplacement du fichier `.jar` du connecteur vers le répertoire `**lib**` d'Apache Guacamole :

```
tar -xzf mysql-connector-j-9.5.0.tar.gz
sudo cp mysql-connector-j-9.5.0/mysql-connector-j-9.5.0.jar /etc/guacamole/lib/
```

- importation de la structure de la base de données Apache Guacamole dans la base de données **guacamole_db** en utilisant tous les fichiers SQL situés dans le répertoire **guacamole-auth-jdbc-1.6.0/mysql/schema/** :

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

- Création et édition du fichier **guacamole.properties** pour déclarer la connexion à MariaDB. Ce fichier peut être utilisé pour d'autres paramètres, selon les besoins.

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

- Ajoutez dans ce fichier les lignes suivantes en adaptant les trois dernières lignes avec vos valeurs :

```
# MySQL
mysql-hostname: 127.0.0.1
mysql-port: 3306
mysql-database: guacamole_db
mysql-username: guacamole_user
mysql-password: P@$w0rd2025Secure
```

- Déclarez 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
```

- Redémarrez les trois services liés à Apache Guacamole :

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

Compte pour Guacamole

Le compte utilisateur **daemon** utilisé par défaut pour exécuter le service **guacd** n'a pas les permissions suffisantes sur les données d'Apache Guacamole. Cela empêche les connexions.

Vérification :

```
sudo ps aux | grep -v grep | grep guacd
daemon      33197  0.0  0.6 247992 13896 ?        Ss   15:29   0:00 /usr/local/sbin/guacd -f
```

- création d'un nouveau compte system appelé **guacd_user**, sans répertoire personnel dans home, sans shell de connexion et configuration de **guacd** pour remplacer le compte **daemon** par **guacd_user** :

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

- vérifiez l'état du service :

```
sudo systemctl status guacd
```

From:
[/ - Les cours du BTS SIO](#)

Permanent link:
</doku.php/reseau/cloud/accesdistance/guacamolenative?rev=1768152448>

Last update: **2026/01/11 18:27**

