

Concevoir, réaliser et présenter une solution technique

SAE 5.01



Modou DIOP
Valentin ENGRAND
Gavin NSHIMIRIMANA

IUT de Béthune
Pôle de l'université d'Artois
Département R&T



Introduction

Dans le cadre du projet R501 SAE, nous avons été amenés à réaliser un projet intitulé : **Configuration et Déploiement d'une Infrastructure de Salles de TP Virtuelles.**

L'objectif principal de ce projet est de concevoir une infrastructure permettant aux étudiants d'accéder à des machines virtuelles configurées pour des travaux pratiques, et ce, via une interface web accessible à distance. Cette solution devra être sécurisée, fiable et capable de répondre aux besoins académiques de manière optimale.

Pour ce faire, nous avons utilisé des outils comme Proxmox, comme hyperviseur pour sa robustesse et sa compatibilité avec les besoins pédagogiques et les besoins du projet, ainsi que Apache Guacamole pour offrir un accès distant via un navigateur.

Pour atteindre cet objectif, nous mettrons en œuvre les étapes suivantes :

Objectifs:

1. Installer et configurer un hyperviseur sur un serveur dédié.
2. Concevoir des machines virtuelles modèles pour les TP avec les logiciels et environnements nécessaires.
3. Déployer une interface web pour l'accès à distance aux machines virtuelles.
4. Intégrer les aspects de sécurité (authentification, gestion d'accès, etc.).
5. Mettre en place une documentation et des procédures pour la maintenance et la supervision.

Le tout sous 4 grandes phases que nous allons présenter dans ce rapport :

Plan du projet :

Phase 1: Conception et Planification (10%)

Phase 2: Installation et Configuration de l'Hyperviseur (25%)

Phase 3: Création des Machines Virtuelles Modèles (20%)

Phase 4: Interface Web pour l'Accès à Distance (25%)

Phase 5 : Synthèse en anglais: 5 points bonus

Phase 6 :Réflexion sur les aspects de développement durable et de transition énergétique: 5 points bonus

Phase 1 : Conception et Planification (10%)

- **Choix de l'hyperviseur** : Proxmox

C'est quoi Proxmox

Proxmox VE (Virtual Environment) est une plateforme de virtualisation qui repose sur KVM (Kernel-based Virtual Machine) pour la gestion des machines virtuelles et sur LXC (Linux Containers) pour la virtualisation basée sur des conteneurs.

Il est fourni avec une interface web intuitive, accessible depuis n'importe quel navigateur, qui simplifie l'administration des ressources virtuelles.

Pourquoi Proxmox répond à ce projet ?

Création de VM modèles : Proxmox permet de configurer des machines virtuelles avec les logiciels requis (comme Linux, Windows ou outils spécifiques aux TP). Ces VM modèles peuvent ensuite être clonées pour les étudiants.

Gestion centralisée avec une interface web qui offre une supervision facile des ressources comme le CPU, la RAM et le stockage.

Accès distant en intégrant Proxmox avec des outils comme **Apache Guacamole**, il devient simple de permettre un accès sécurisé et ergonomique aux machines virtuelles depuis n'importe où.

Sécurité intégrée avec de l'authentification ou l'intégration LDAP pour gérer les utilisateurs, répondant aux attentes en termes de sécurité.

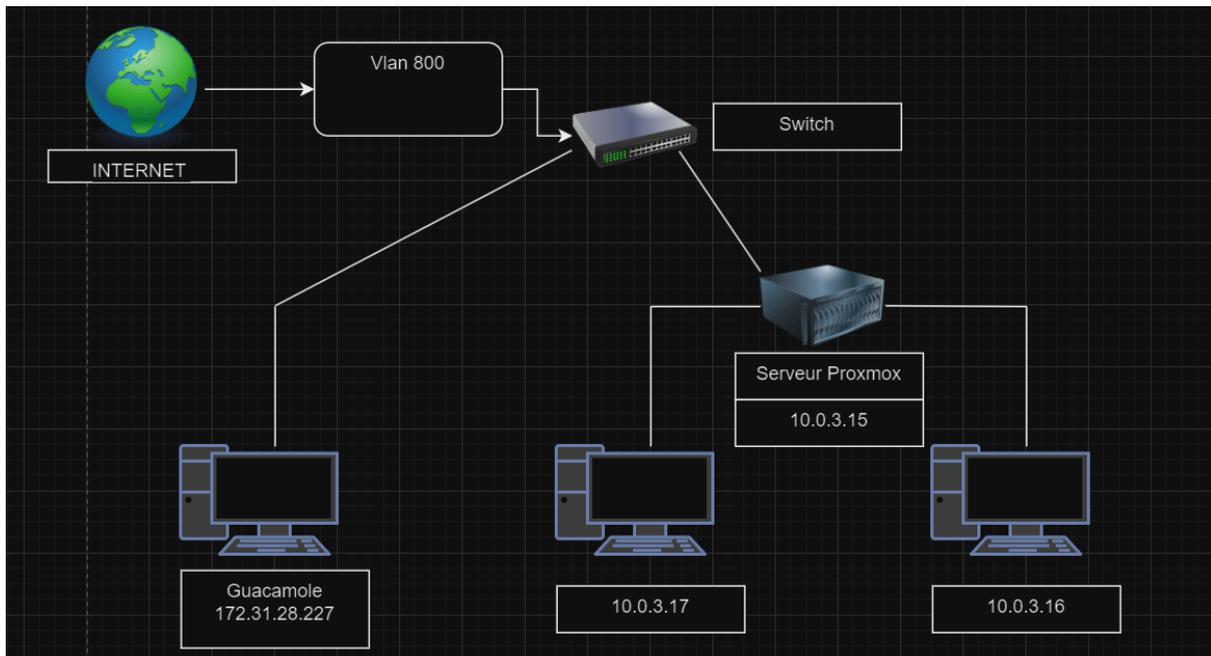
Outils et fonctionnalités utilisés pour le projet ?

Fonctionnalité	Description
Snapshots	Permet de sauvegarder l'état des VMs à différents moments, utile pour restaurer des configurations en cas d'erreurs.
Support des VLAN	Offre la possibilité de segmenter les réseaux selon les besoins pédagogiques.
Clonage de VMs	Facilite le déploiement rapide d'environnements identiques à partir des modèles.
Statistiques en temps réel	Permet de surveiller les performances des VM et d'ajuster les ressources si nécessaire.

- Répartition des tâches :

Diagramme de Gantt

Tâches	Étape 1	Étape 2	Étape 3
Choix de l'hyperviseur & Documentation des besoins Installation et Configuration de l'Hyperviseur	X		
Répartition des tâches & Installations et prise en main de la machine serveur (Proxmox), Création des Machines Virtuelles Modèles	X		
Établissement du planning & Interface Web pour l'Accès à Distance		X	
Interface Web pour l'Accès à Distance & Rédaction des bonus (Anglais & développement durable)			X



Phase 2: Installation et Configuration de l'Hyperviseur (25%)

Cette phase est cruciale pour poser les bases de notre infrastructure. Elle consiste à installer l'hyperviseur Proxmox sur un serveur dédié et à configurer les ressources nécessaires pour la gestion des machines virtuelles.

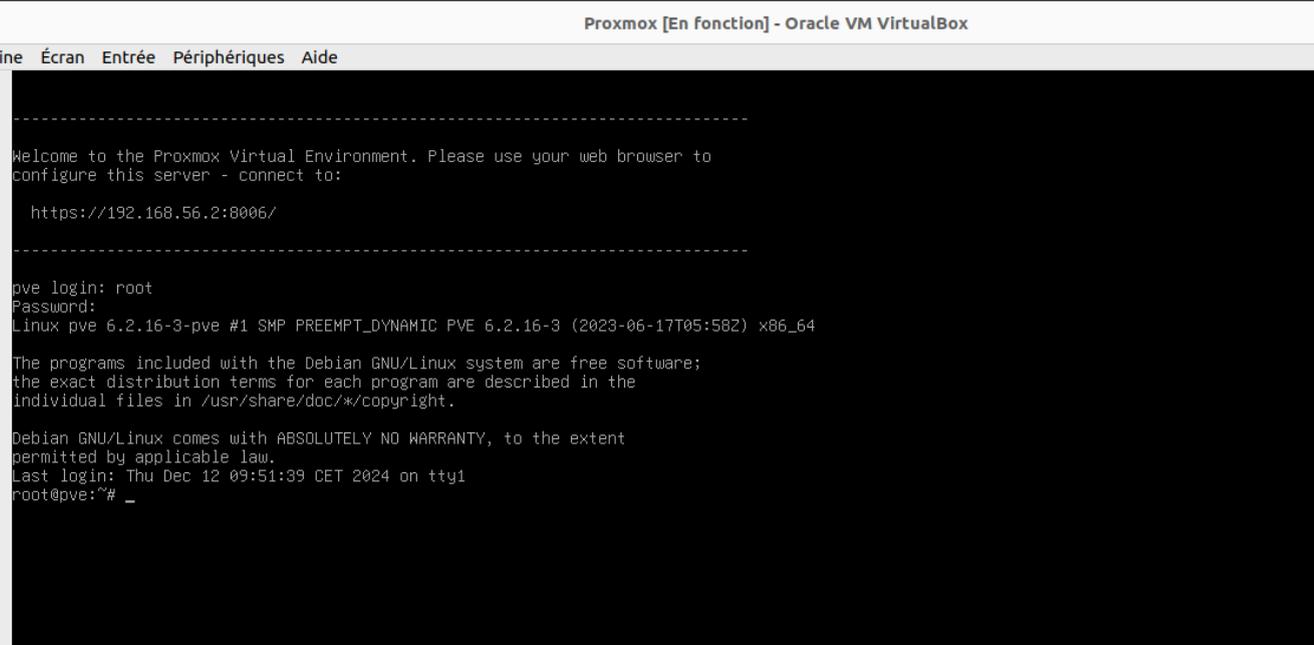
Nous travaillons sur l'environnement de l'IUT, donc nous utilisons du matériel et logiciel : une image ISO de Proxmox.

1. Installation de l'hyperviseur sur un serveur

L'installation débute pour nous au lancement de la machine virtuelle après sa création sur la machine physique de l'IUT.

Lancement de la VM proxmox :

Ici nous pouvons apercevoir que la machine virtuelle Proxmox a été démarré et configurée avec succès.



```
Proxmox [En fonction] - Oracle VM VirtualBox
ine Écran Entrée Périphériques Aide
-----
Welcome to the Proxmox Virtual Environment. Please use your web browser to
configure this server - connect to:

  https://192.168.56.2:8006/
-----
pve login: root
Password:
Linux pve 6.2.16-3-pve #1 SMP PREEMPT_DYNAMIC PVE 6.2.16-3 (2023-06-17T05:58Z) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec 12 09:51:39 CET 2024 on tty1
root@pve:~# _
```

Interface du terminal du serveur de la VM Proxmox

Dans cette phase, il est aussi nécessaire d'ajuster manuellement certains paramètres pour répondre aux besoins spécifiques du système.

Les actions réalisées incluent la configuration d'une adresse IP statique, l'activation des interfaces réseau, et l'application de règles de traduction d'adresse réseau (NAT) à l'aide d'iptables. Ces réglages visent à rediriger le trafic externe vers des services internes tout en maintenant une gestion cohérente des flux réseau.

```

192.168.56.100: any valid prefix is expected rather than 192.168.56.100/24 .
root@pve:~# ip a add 192.168.56.100/24 dev enp0s3
root@pve:~# ip link set enp0s3 up
root@pve:~# iptables -t nat -A PREROUTING -d 192.168.56.100 -p tcp --dport 8006 -j DNAT --to-destination 10.0.3.15:8006
root@pve:~# iptables -t nat -A POSTROUTING -d 10.0.3.15 -p tcp --dport 8006 -j MASQUERADE
root@pve:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:02:8c:17 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.100/24 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe02:8c17/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master vmbri0 state UP group default qlen 1000
    link/ether 08:00:27:cd:74:ee brd ff:ff:ff:ff:ff:ff
4: vmbri0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 08:00:27:cd:74:ee brd ff:ff:ff:ff:ff:ff
    inet 10.0.3.15/24 scope global vmbri0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fedc:74ee/64 scope link
        valid_lft forever preferred_lft forever

```

Configuration de la machine virtuelle Proxmox

Voici les commandes effectuées en clair :

ip a add 192.168.56.100/24 dev enp0s3

Configuration manuelle d'une adresse IP sur l'interface réseau, permettant à la machine de communiquer sur ce sous-réseau.

ip link set enp0s3 up

Cette commande active (met "up") l'interface réseau enp0s3, rendant l'interface prête à émettre et recevoir du trafic.

iptables -t nat -A PREROUTING -d 192.168.56.100 -p tcp --dport 8006 -j DNAT --to-destination 10.0.3.15:8006

Cette règle iptables redirige tout le trafic TCP entrant sur l'adresse 192.168.56.100 et le port 8006 vers l'adresse IP 10.0.3.15 et le port 8006.

iptables -t nat -A POSTROUTING -d 10.0.3.15 -p tcp --dport 8006 -j MASQUERADE

Cette règle modifie les paquets sortants destinés à l'adresse 10.0.3.15:8006, en remplaçant leur adresse source par celle de l'interface réseau de sortie. Cela permet à la réponse des paquets de revenir correctement à la machine d'origine, sans qu'elle ait besoin de connaître directement l'adresse IP 10.0.3.15.

2. Découverte de l'environnement Proxmox via l'interface web

Pour configurer les ressources de notre hyperviseur dans Proxmox, voici les étapes :

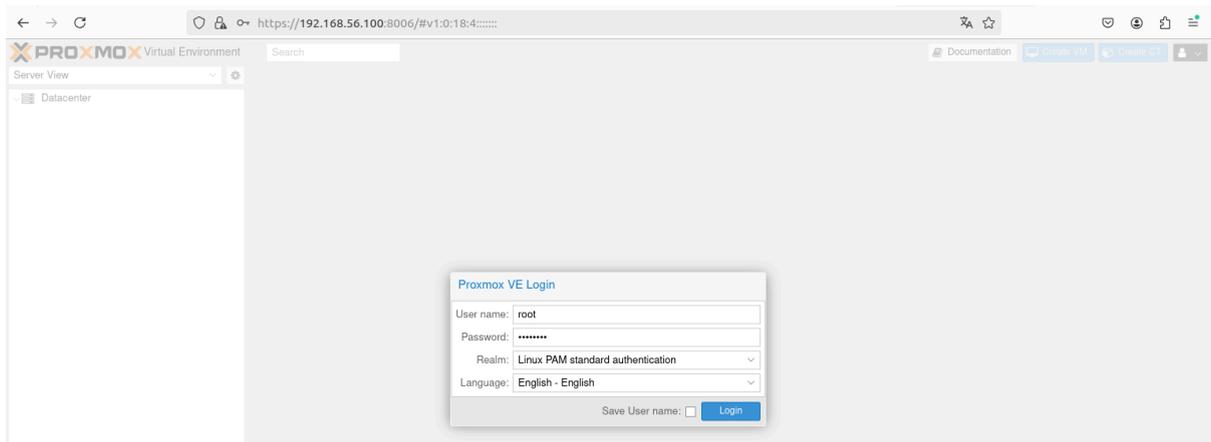
Accès à l'interface web Proxmox :

Nous nous sommes connectés à l'interface via votre navigateur en entrant l'adresse IP : <https://192.168.56.100:8006>.

Utiliser les identifiants fournis pour nous connecter. Par défaut :

- **Nom d'utilisateur** : root
- **Mot de passe** : progr00

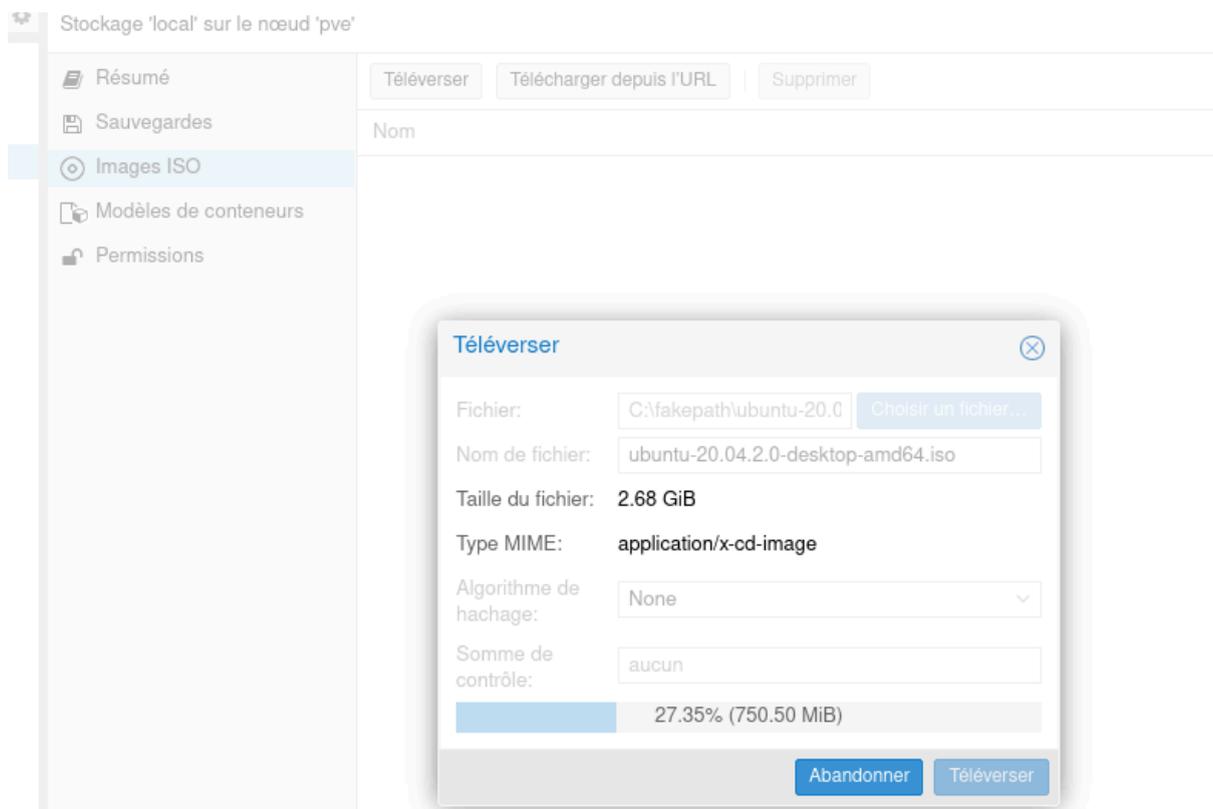
Et puis sélectionner "Linux PAM standard authentication" pour l'authentification.



Ajout d'images ISO pour les systèmes d'exploitation

Accédez à l'onglet **Images ISO** dans la section de stockage local (**local** sur le nœud **pve**).

Cliquez sur **Téléverser** pour importer les fichiers ISO nécessaires, comme ci-dessus nous avons utilisé : **ubuntu-20.04.2.0-desktop-amd64.iso**.



Importation de L'ISO ubuntu 20.04

Une fois l'image ISO ajoutée, elle pourra être utilisée pour créer des machines virtuelles.

Phase 3: Création des Machines Virtuelles Modèles (20%)

1. Définition des besoins en logiciels et environnements pour les TP

Pour répondre aux besoins pédagogiques des étudiants, nous avons identifié **trois types de machines virtuelles (VM)** à mettre en place, chacune dédiée à des usages spécifiques. Voici la description des environnements et logiciels requis.

Type de VM	Système d'exploitation	Logiciels principaux	Utilisation
Linux - Réseau/Programmation	Ubuntu 22.04 LTS	GNS3, Packet Tracer, Python3, Wireshark	TP réseau, programmation, simulation
Windows - Administration	Windows 10 Professionnel	Outils RSAT, gestion utilisateurs, GPO	Administration réseau sous Windows
Asterisk Alpine - VoIP	Alpine Linux	Asterisk, Fanvil, OpenSSH	Déploiement et test de téléphonie VoIP

Problèmes rencontrés :

Les machines virtuelles se sont systématiquement arrêtées de manière inattendue lors de l'utilisation des images ISO Windows et Asterisk, rendant impossible la poursuite du projet avec ces systèmes. De plus, toutes les images Ubuntu ont provoqué des dysfonctionnements similaires, à l'exception de l'image Ubuntu 22.04, que nous avons finalement utilisée pour avancer dans le projet.

Une fois que la machine virtuelle (VM) est configurée avec tous les logiciels nécessaires, on la dupliquera pour créer autant de copies que d'élèves. Cela permet de déployer rapidement les mêmes VMs sans avoir à tout configurer manuellement à chaque fois. La duplication garantit que chaque élève aura une VM identique avec tous les outils déjà installés.

2. Création et configuration des machines virtuelles modèles

The screenshot shows the 'Create: Virtual Machine' dialog box with the 'General' tab selected. The fields are filled with the following values:

- Node: pve
- VM ID: 101
- Name: TEST1
- Resource Pool: (empty)

At the bottom, there is a 'Help' button, an 'Advanced' checkbox (unchecked), and 'Back' and 'Next' buttons.

ISO Ubuntu 20.04 a été utilisé pour éviter les crashes sur les machines virtuelles.

The screenshot shows the 'Create: Virtual Machine' dialog box with the 'OS' tab selected. The 'Use CD/DVD disc image file (iso)' option is selected. The 'Guest OS' section is configured as follows:

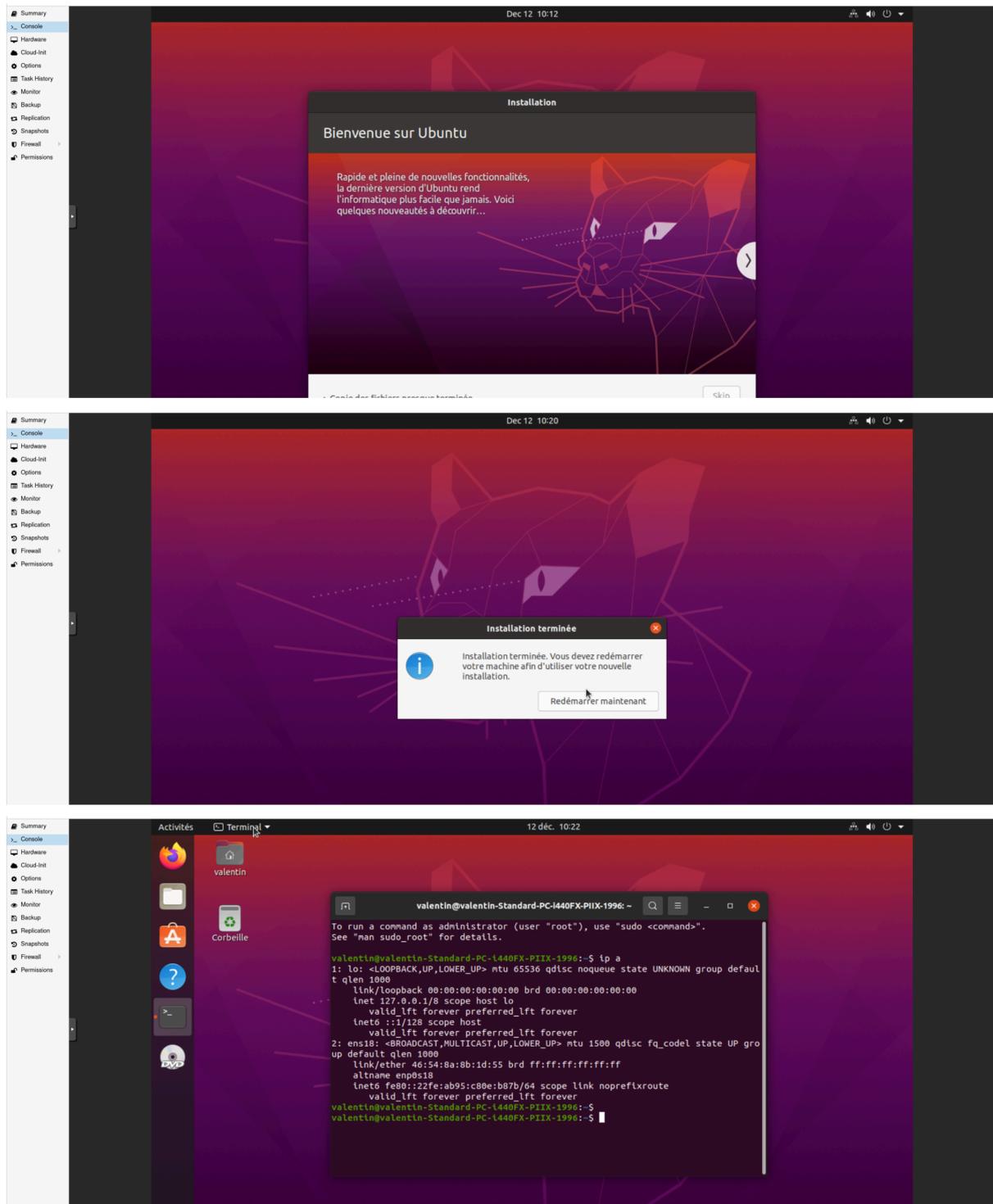
- Storage: local
- Type: Linux
- ISO image: ubuntu-20.04.2.0-desktop-amd64.iso
- Version: 6.x - 2.6 Kernel

Below the 'ISO image' dropdown, a table lists available ISO images:

Name	For...	Size
ubuntu-20.04.2.0-desktop-amd64.iso	iso	2.88 GB
ubuntu-22.04.1-desktop-amd64.iso	iso	3.83 GB

At the bottom, there is an 'Advanced' checkbox (unchecked) and 'Back' and 'Next' buttons.

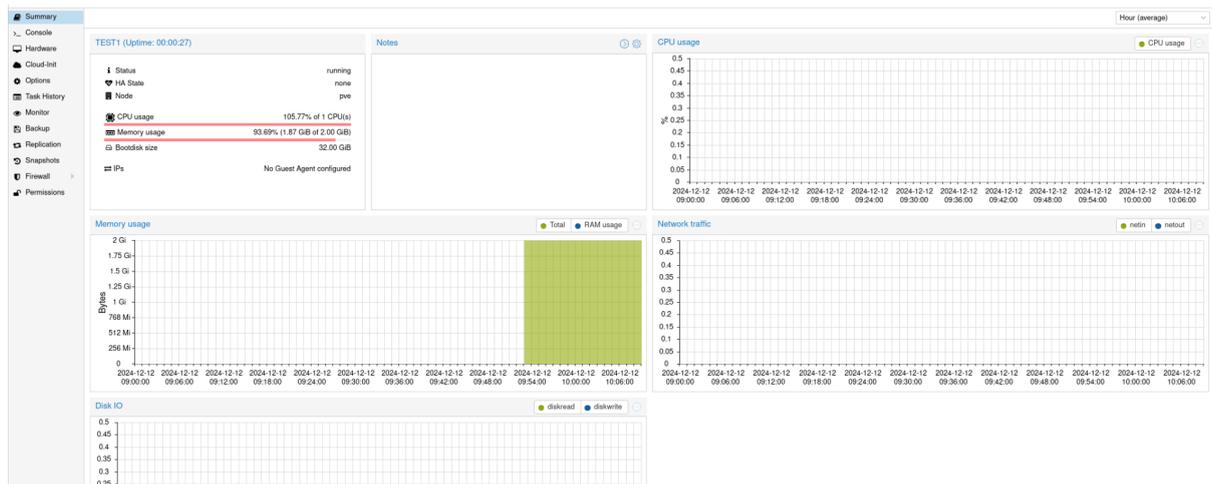
Accès à la machine virtuelle via l'onglet **console** dans Proxmox.



Une fois ces étapes réalisées nous continuons l'installation par défaut.

Vérification des ressources dans l'onglet **Summary**

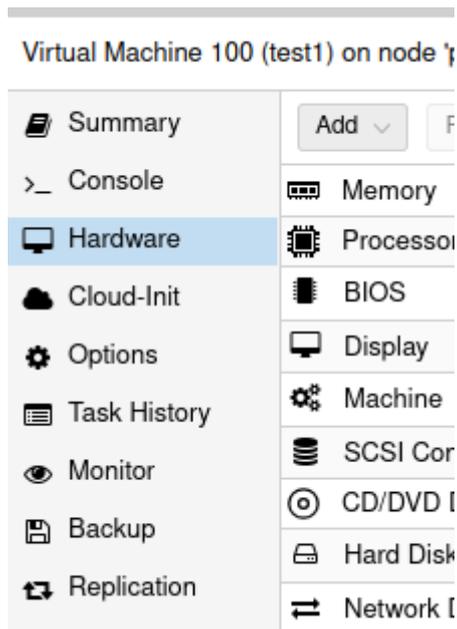
L'onglet **Summary** (Summary) permet de surveiller et vérifier l'état des ressources allouées aux machines virtuelles ou au nœud hôte. Cet outil fournit des informations telles que l'utilisation du CPU, de la mémoire vive, du stockage, ainsi que les statistiques réseau, garantissant ainsi une gestion optimale des ressources et une détection rapide des anomalies.



Une fois la machine virtuelle installée, nous pouvons procéder à l'installation des différents logiciels nécessaires.

```
valentin@valentin-Standard-PC-i440FX-PIIX-1996:~$ sudo apt install python3
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
python3 est déjà la version la plus récente (3.9.2-0ubuntu3)
```

Nous ajoutons ensuite une carte réseau à la machine virtuelle pour lui attribuer une adresse IP.



La vérification de l'adresse IP se fait à l'aide de la commande `ip a`.

```
valentin@valentin-Standard-PC-i440FX-PIIX-1996:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    Terminal 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 4e:23:9b:7c:91:d5 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
3: ens19: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 1a:9f:8a:a8:05:bc brd ff:ff:ff:ff:ff:ff
    altname enp0s19
    inet 10.0.3.16/24 brd 10.0.3.255 scope global dynamic noprefixroute ens19
        valid_lft 84702sec preferred_lft 84702sec
    inet6 fe80::abed:3095:7c11:c2a0/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
valentin@valentin-Standard-PC-i440FX-PIIX-1996:~$
```

Ici nous pouvons observer qu'on obtient bien une adresse IP : 10.0.3.16

Enfin, nous testons la connectivité réseau : la machine virtuelle est capable de **pinguer** le serveur Proxmox ainsi que la machine hébergeant l'application Guacamole, confirmant ainsi une configuration réseau réussie.

```
valentin@valentin-Standard-PC-i440FX-PIIX-1996:~$ ip a
...
    inet6 fe80::abed:3095:7c11:c2a0/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
valentin@valentin-Standard-PC-i440FX-PIIX-1996:~$ ping 10.0.3.15
PING 10.0.3.15 (10.0.3.15) 56(84) bytes of data:
64 octets de 10.0.3.15 : icmp_seq=1 ttl=64 temps=0.147 ms
64 octets de 10.0.3.15 : icmp_seq=2 ttl=64 temps=0.195 ms
64 octets de 10.0.3.15 : icmp_seq=3 ttl=64 temps=0.208 ms
^C
--- statistiques ping 10.0.3.15 ---
3 paquets transmis, 3 reçus, 0 % paquets perdus, temps 2028 ms
rtt min/avg/max/mdev = 0.147/0.183/0.208/0.026 ms
valentin@valentin-Standard-PC-i440FX-PIIX-1996:~$ ping 172.31.28.227
PING 172.31.28.227 (172.31.28.227) 56(84) bytes of data:
64 octets de 172.31.28.227 : icmp_seq=1 ttl=63 temps=0.546 ms
64 octets de 172.31.28.227 : icmp_seq=2 ttl=63 temps=0.518 ms
64 octets de 172.31.28.227 : icmp_seq=3 ttl=63 temps=0.825 ms
^C
--- statistiques ping 172.31.28.227 ---
3 paquets transmis, 3 reçus, 0 % paquets perdus, temps 2044 ms
rtt min/avg/max/mdev = 0.518/0.629/0.825/0.138 ms
valentin@valentin-Standard-PC-i440FX-PIIX-1996:~$
```

Phase 4 : Interface Web pour l'Accès à Distance (25%)

Cette étape vise à permettre aux utilisateurs d'accéder aux machines virtuelles via une interface web, quel que soit leur emplacement.

Pourquoi avons-nous utilisé Apache Guacamole ?

Apache Guacamole permet un accès aux machines virtuelles via un simple navigateur web, sans installer de logiciel supplémentaire.

- Accessibilité : Connexion facile via RDP, SSH ou VNC directement depuis une interface web.
- Sécurité : Accès centralisé et authentifié pour contrôler les droits utilisateurs.
- Pratique : Interface intuitive facilitant l'utilisation pour les étudiants.

Mise en Place d'Apache Guacamole :

- Installation des dépendances, compilation et déploiement de Guacamole Server.
- Configuration de Tomcat9 pour le client Guacamole (Web App).
- Mise en place de l'authentification avec MariaDB.

Contrôle des Accès :

- Création des utilisateurs et gestion des droits d'accès.

1. Déployer une solution utilisateur (guacamole, bureau à distance ...). Les étudiants doivent pouvoir accéder aux VM via l'interface web de n'importe où.

Avant de commencer, configurez le proxy pour accéder aux dépôts nécessaires :

```
export http_proxy="http://cache-etu.univ-artois.fr:3128"  
export https_proxy="http://cache-etu.univ-artois.fr:3128"
```

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

```
apt-get update  
apt-get install build-essential libcairo2-dev libjpeg-turbo8-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
```

Par la suite 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
```

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 :

```
root@rt-mv:/home/administrateur# cd /tmp
root@rt-mv:/tmp# wget https://downloads.apache.org/guacamole/1.5.5/source/guacamole-server-1.5.5.tar.gz
--2024-12-12 10:36:22-- https://downloads.apache.org/guacamole/1.5.5/source/guacamole-server-1.5.5.tar.gz
Résolution de downloads.apache.org (downloads.apache.org)... 135.181.214.104, 88.99.208.237, 2a01:4f9:3a:2c57::2, ...
Connexion à downloads.apache.org (downloads.apache.org)|135.181.214.104|:443... ^
C
root@rt-mv:/tmp# export http_proxy="http://cache-etu.univ-arts.fr:3128"
root@rt-mv:/tmp# export https_proxy="http://cache-etu.univ-arts.fr:3128"
root@rt-mv:/tmp# wget https://downloads.apache.org/guacamole/1.5.5/source/guacamole-server-1.5.5.tar.gz
--2024-12-12 10:40:34-- https://downloads.apache.org/guacamole/1.5.5/source/guacamole-server-1.5.5.tar.gz
Résolution de cache-etu.univ-arts.fr (cache-etu.univ-arts.fr)... 193.49.62.52
Connexion à cache-etu.univ-arts.fr (cache-etu.univ-arts.fr)|193.49.62.52|:3128... connecté.
requête Proxy transmise, en attente de la réponse... 200 OK
Taille : 1136892 (1,1M) [application/x-gzip]
Enregistre : 'guacamole-server-1.5.5.tar.gz'

guacamole-server-1. 100%[=====] 1,08M 2,54MB/s ds 0,4s

2024-12-12 10:40:35 (2,54 MB/s) - 'guacamole-server-1.5.5.tar.gz' enregistré [1136892/1136892]

root@rt-mv:/tmp#
```

```
cd guacamole-server-1.5.5/
```

```
root@rt-mv:/tmp# tar -xzf guacamole-server-1.5.5.tar.gz
cd guacamole-server-1.5.5/
root@rt-mv:/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 :

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

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

```
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.

root@rt-mv:/tmp/guacamole-server-1.5.5#
```

Ensuite on poursuit avec la compilation du code source de guacamole-server :

`sudo make`

```
CC      guacenc-instruction-shade.o
CC      guacenc-instruction-size.o
CC      guacenc-instruction-sync.o
CC      guacenc-instruction-transfer.o
CC      guacenc-jpeg.o
CC      guacenc-layer.o
CC      guacenc-log.o
CC      guacenc-parse.o
CC      guacenc-png.o
CC      guacenc-video.o
CC      guacenc-webp.o
CCLD    guacenc
make[2] : on quitte le répertoire « /tmp/guacamole-server-1.5.5/src/guacenc »
Making all in src/guaclog
make[2] : on entre dans le répertoire « /tmp/guacamole-server-1.5.5/src/guaclog
»
CC      guaclog-guaclog.o
CC      guaclog-instructions.o
CC      guaclog-instruction-key.o
CC      guaclog-interpret.o
CC      guaclog-keydef.o
CC      guaclog-log.o
CC      guaclog-state.o
CCLD    guaclog
make[2] : on quitte le répertoire « /tmp/guacamole-server-1.5.5/src/guaclog »
make[2] : on entre dans le répertoire « /tmp/guacamole-server-1.5.5 »
make[2] : on quitte le répertoire « /tmp/guacamole-server-1.5.5 »
make[1] : on quitte le répertoire « /tmp/guacamole-server-1.5.5 »
root@rt-mv:/tmp/guacamole-server-1.5.5#
```

Enfin, on termine par installer le composant Guacamole Server :

```
»  
/usr/bin/mkdir -p '/usr/local/bin'  
/bin/bash ../../libtool --mode=install /usr/bin/install -c guacenc '/usr/lo  
Fichiers  
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/lo  
cal/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 »  
root@rt-mv:/tmp/guacamole-server-1.5.5# ^[[2~
```

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 bibliothèques (cette commande ne retourne aucun résultat) :

```
root@rt-mv:/tmp/guacamole-server-1.5.5# sudo ldconfig  
root@rt-mv:/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.

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

```
root@rt-mv:/tmp/guacamole-server-1.5.5# sudo systemctl daemon-reload  
sudo systemctl enable --now guacd  
Created symlink /etc/systemd/system/multi-user.target.wants/guacd.service → /etc/systemd/system/guacd.service.  
root@rt-mv:/tmp/guacamole-server-1.5.5#
```

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

```
sudo systemctl status guacd
```

```
e/systemd/system/guacd.service
root@rt-mv:/tmp/guacamole-server-1.5.5# sudo systemctl status guacd
● guacd.service - Guacamole Server
   Loaded: loaded (/etc/systemd/system/guacd.service; enabled; vendor preset:
   Active: active (running) since Thu 2024-12-12 10:52:29 CET; 1min 12s ago
     Docs: man:guacd(8)
    Main PID: 19766 (guacd)
      Tasks: 1 (limit: 2878)
     Memory: 10.2M
        CPU: 6ms
    CGroup: /system.slice/guacd.service
            └─19766 /usr/local/sbin/guacd -f

déc. 12 10:52:29 rt-mv systemd[1]: Started Guacamole Server.
déc. 12 10:52:29 rt-mv guacd[19766]: Guacamole proxy daemon (guacd) version 1.
déc. 12 10:52:29 rt-mv guacd[19766]: guacd[19766]: INFO:          Guacamole prox
déc. 12 10:52:29 rt-mv guacd[19766]: Listening on host 127.0.0.1, port 4822
déc. 12 10:52:29 rt-mv guacd[19766]: guacd[19766]: INFO:          Listening on h
lines 1-16/16 (END)
```

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.

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

Installer Guacamole Client (Web App)

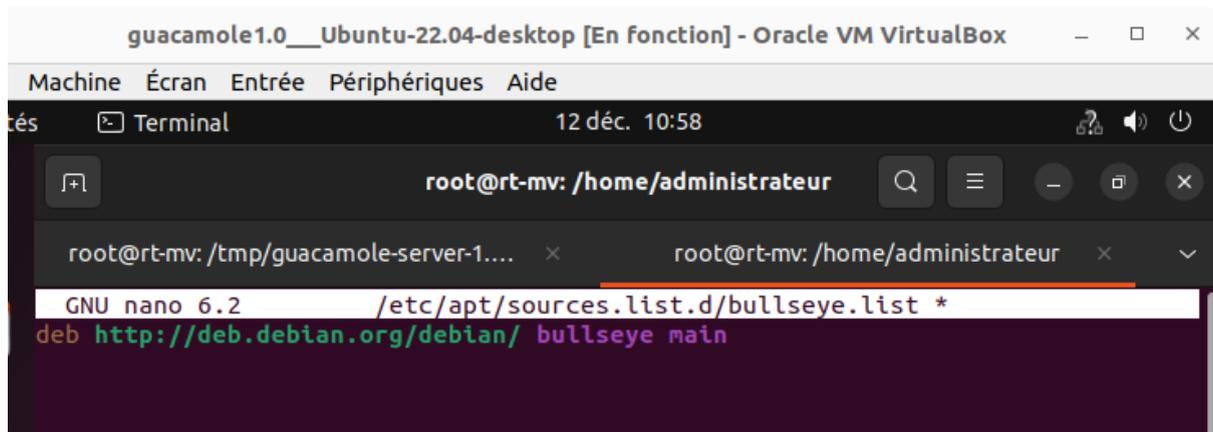
Pour la Web App correspondante à Apache Guacamole, et donc à la partie cliente, nous avons besoin d'un serveur Tomcat 9. Tomcat 10, distribué par défaut via les dépôts de Debian 12, n'est pas pris en charge par Apache Guacamole. Nous devons ajouter le dépôt de Debian 11 sur notre machine Debian 12 afin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

Nous avons ajouté cette ligne, puis enregistrez et fermez le fichier.

→ [deb http://deb.debian.org/debian/bullseye main](http://deb.debian.org/debian/bullseye/main)

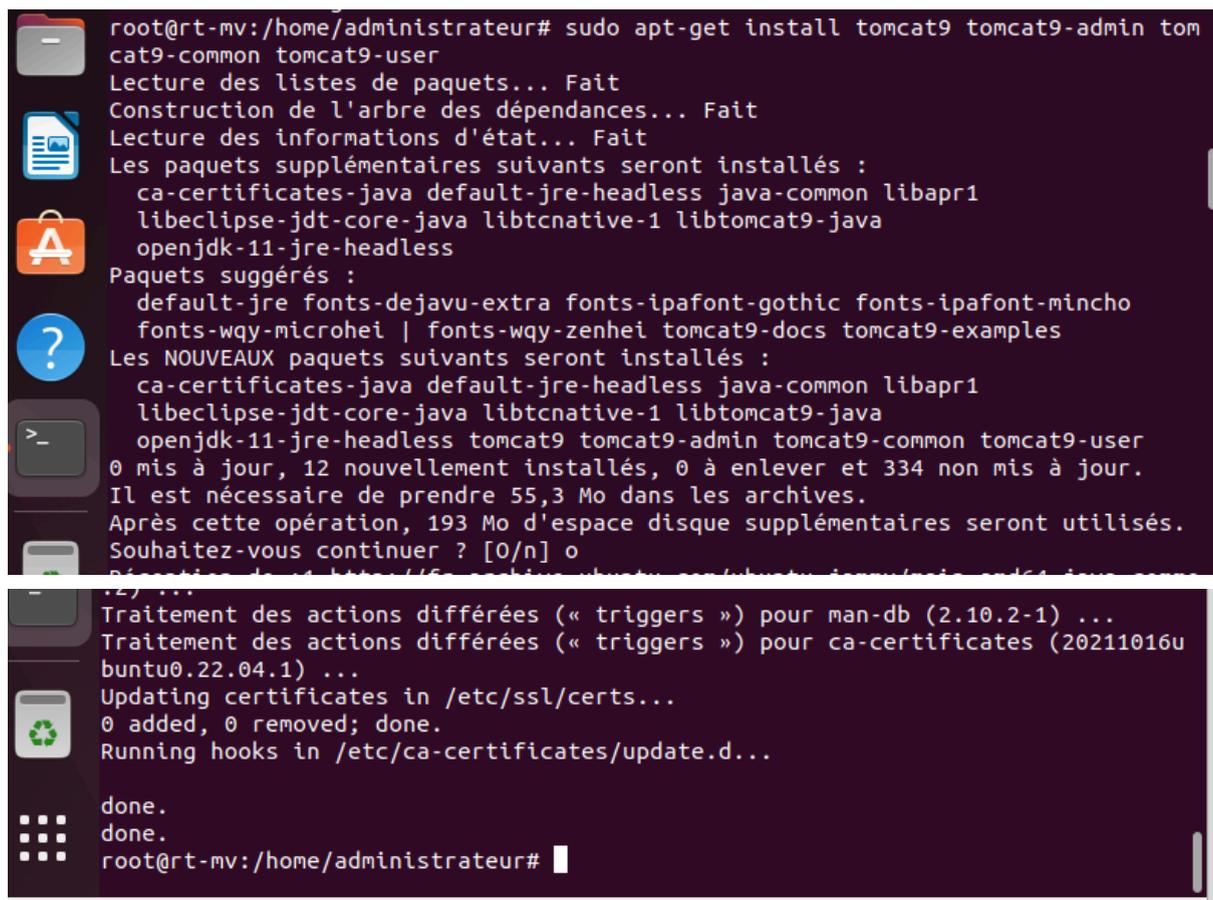


Nous mettons à jour le cache des paquets :

```
sudo apt-get update
```

installation des paquets Tomcat 9 sur Debian 12 avec cette commande :

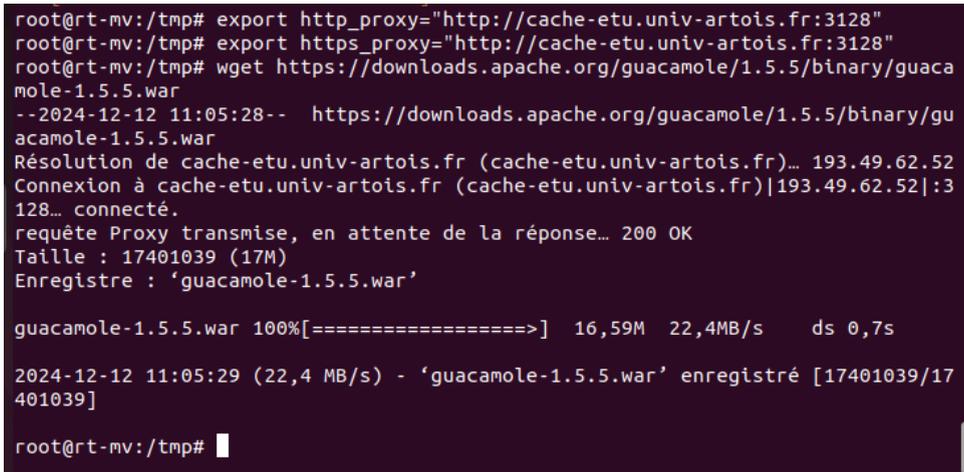
```
sudo apt-get install tomcat9 tomcat9-admin tomcat9-common tomcat9-user
```



Ensuite 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
export http_proxy="http://cache-etu.univ-artois.fr:3128"
export https_proxy="http://cache-etu.univ-artois.fr:3128"
wget https://downloads.apache.org/guacamole/1.5.5/binary/guacamole-1.5.5.war
```



```
root@rt-mv:/tmp# export http_proxy="http://cache-etu.univ-artois.fr:3128"
root@rt-mv:/tmp# export https_proxy="http://cache-etu.univ-artois.fr:3128"
root@rt-mv:/tmp# wget https://downloads.apache.org/guacamole/1.5.5/binary/guacamole-1.5.5.war
--2024-12-12 11:05:28-- https://downloads.apache.org/guacamole/1.5.5/binary/guacamole-1.5.5.war
Résolution de cache-etu.univ-artois.fr (cache-etu.univ-artois.fr)... 193.49.62.52
Connexion à cache-etu.univ-artois.fr (cache-etu.univ-artois.fr)|193.49.62.52|:3128... connecté.
requête Proxy transmise, en attente de la réponse... 200 OK
Taille : 17401039 (17M)
Enregistre : 'guacamole-1.5.5.war'

guacamole-1.5.5.war 100%[=====>] 16,59M 22,4MB/s ds 0,7s

2024-12-12 11:05:29 (22,4 MB/s) - 'guacamole-1.5.5.war' enregistré [17401039|7401039]

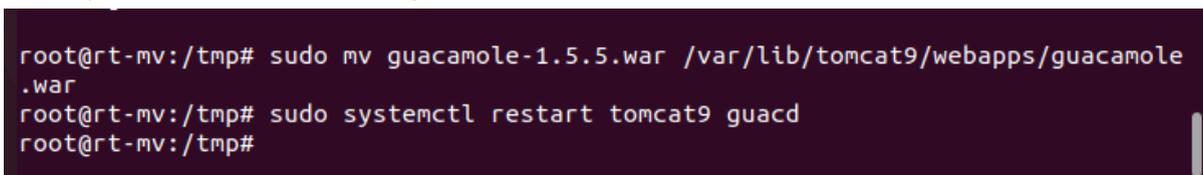
root@rt-mv:/tmp#
```

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
```



```
root@rt-mv:/tmp# sudo mv guacamole-1.5.5.war /var/lib/tomcat9/webapps/guacamole.war
root@rt-mv:/tmp# sudo systemctl restart tomcat9 guacd
root@rt-mv:/tmp#
```

Voilà, Apache Guacamole Client est installé !

2. Mise en œuvre de l'authentification et de la gestion des droits

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...).

```
sudo mysql_secure_installation
```

```
root@rt-mv: /tmp/guacamole-server-1... x root@rt-mv: /tmp x v
Disallow root login remotely? [Y/n] y
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
root@rt-mv:/tmp#
```

```
root@rt-mv:/tmp# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 41
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)]>
```

```
root@rt-mv: /tmp/guacamole-serve... x root@rt-mv: /tmp x root@rt-mv: /tmp/guacamole-auth-j... x
Fichiers up...
All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
root@rt-mv:/tmp# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 56
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 'guaca_nachos'@'localhost' IDENTIFIED BY 'P@ssword!';
Query OK, 0 rows affected (0,001 sec)

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

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

MariaDB [(none)]> EXIT;
Bye
root@rt-mv:/tmp#
```

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
export http_proxy="http://cache-etu.univ-artois.fr:3128"
export https_proxy="http://cache-etu.univ-artois.fr:3128"
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
```

```
root@rt-mv:/tmp# tar -xzf guacamole-auth-jdbc-1.5.5.tar.gz
root@rt-mv:/tmp#
```

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/
```

```
root@rt-mv:/tmp# tar -xzf guacamole-auth-jdbc-1.5.5.tar.gz
root@rt-mv:/tmp# sudo mv guacamole-auth-jdbc-1.5.5/mysql/guacamole-auth-jdbc-my
sql-1.5.5.jar /etc/guacamole/extensions/
root@rt-mv:/tmp#
```

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".

```
root@rt-mv:/tmp# cd guacamole-auth-jdbc-1.5.5/mysql/schema/
root@rt-mv:/tmp/guacamole-auth-jdbc-1.5.5/mysql/schema# cat *.sql | mysql -u root -p guacadb
Enter password:
root@rt-mv:/tmp/guacamole-auth-jdbc-1.5.5/mysql/schema#
```

```
root@rt-mv: /tmp/guacamole-auth-jdbc-1.5.5/mysql/schema
root@rt-mv: /tmp/guacamole-serve... x root@rt-mv: /tmp/guacamole-auth-j... x root@rt-
GNU nano 6.2 /etc/guacamole/guacamole.properties *
# MySQL
mysql-hostname: 127.0.0.1
mysql-port: 3306
mysql-database: guacadb
mysql-username: guaca_nachos
mysql-password: P@ssword!
```

```
root@rt-mv: /tmp/guacamole-auth-jdbc-1.5.5/mysql/schema
root@rt-mv: /tmp/guacamole-serve... x root@rt-mv: /tmp/guacamole-auth-j... x root@rt-
GNU nano 6.2 /etc/guacamole/guacd.conf *
[server]
bind_host = 0.0.0.0
bind_port = 4822
```

```
Query OK, 0 rows affected (0,000 sec)
MariaDB [(none)]> EXIT;
Bye
root@rt-mv:/tmp# cd guacamole-auth-jdbc-1.5.5/mysql/schema/
root@rt-mv:/tmp/guacamole-auth-jdbc-1.5.5/mysql/schema# cat *.sql | mysql -u root -p guacadb
Enter password:
root@rt-mv:/tmp/guacamole-auth-jdbc-1.5.5/mysql/schema# sudo nano /etc/guacamole/guacamole.properties
root@rt-mv:/tmp/guacamole-auth-jdbc-1.5.5/mysql/schema# sudo nano /etc/guacamole/guacd.conf
root@rt-mv:/tmp/guacamole-auth-jdbc-1.5.5/mysql/schema# sudo systemctl restart tomcat9 guacd mariadb
root@rt-mv:/tmp/guacamole-auth-jdbc-1.5.5/mysql/schema#
```

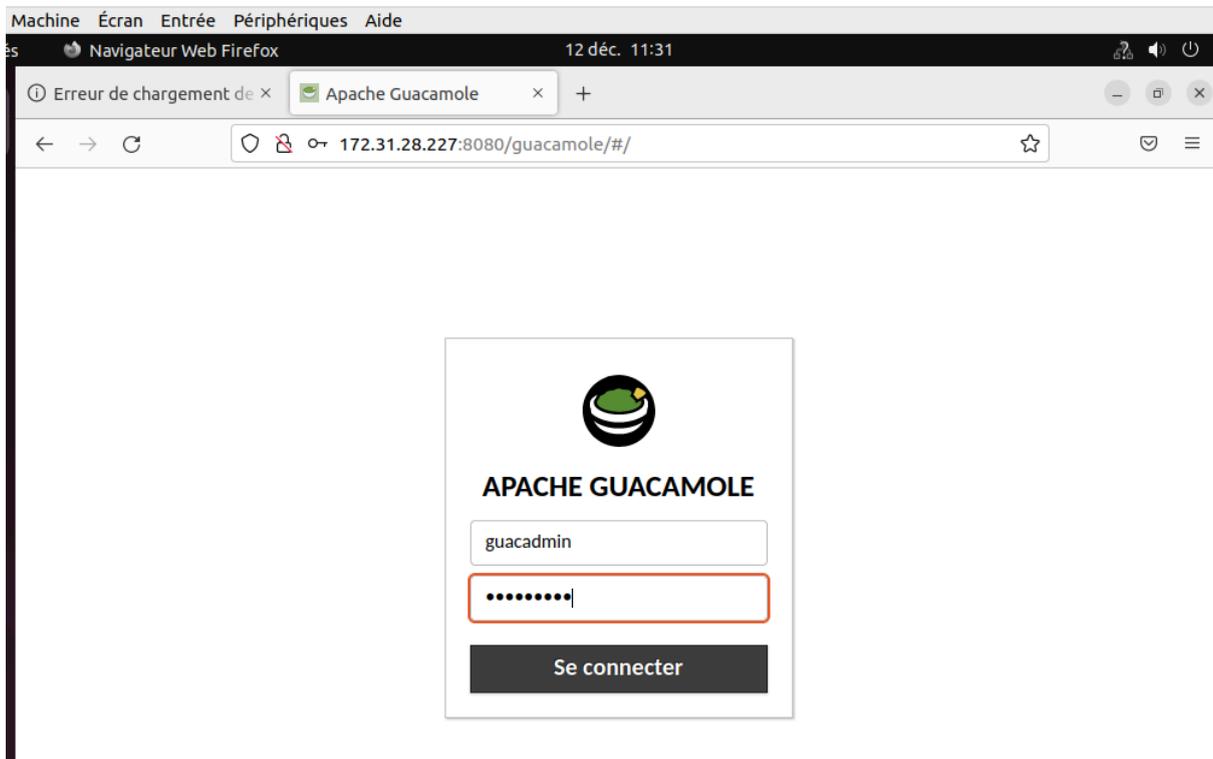
Voilà, l'installation de base est terminée !

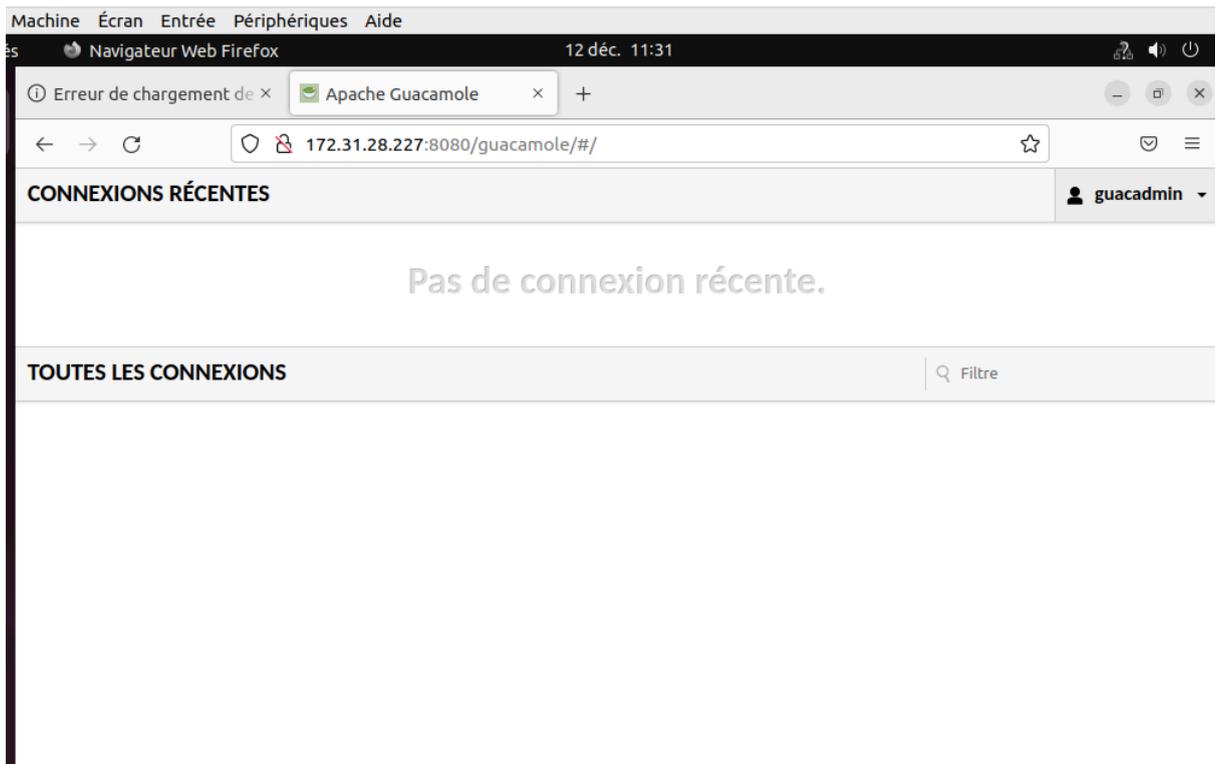


The image shows the Apache Guacamole login interface. At the top center is the Apache Guacamole logo, which consists of a black circle containing a green guacamole bowl with a yellow slice of lime. Below the logo, the text "APACHE GUACAMOLE" is displayed in a bold, black, sans-serif font. Underneath the text are two input fields: the first is labeled "Identifiant" and the second is labeled "Mot de passe". At the bottom of the form is a dark grey button with the white text "Se connecter".

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

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





Nous procédons maintenant à la création d'un utilisateur :

← → ↻ 172.31.28.227:8080/guacamole/#/manage/mysql/users/

MODIFIER UTILISATEUR

Identifiant:

Mot de passe:

Répéter mot de passe:

PROFIL

Nom:

Adresse Mail:

Organisation:

Rôle:

RESTRICTIONS DE COMPTE

Connexion désactivée:

Mot de passe expiré:

Autoriser l'accès après:

Ne pas autoriser l'accès après:

Activer le compte après:

Désactiver le compte après:

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

Connexions en cours **Toutes les Connexions**

☰ test

Il faut spécifier sur quelles machines l'utilisateur, ici appelé *test*, doit pouvoir se connecter via SSH :

MODIFIER CONNEXION

Nom:

Lieu:

Protocole:

Nous allons nous connecter à la machine virtuelle de Gavin.

PARAMÈTRES

Réseau

Nom d'hôte:

Port:

Clé publique de l'hôte (Base64):

Elle à comme adresse IP: 172.31.28.229

Ne pas oublier de mettre le port, ici 22, port par défaut de ssh

SFTP

Activer SFTP:

Cocher la case SFTP.

Elle permet de transférer des fichiers entre chaque machine

Ensuite aller sur la connexion cliente : test2

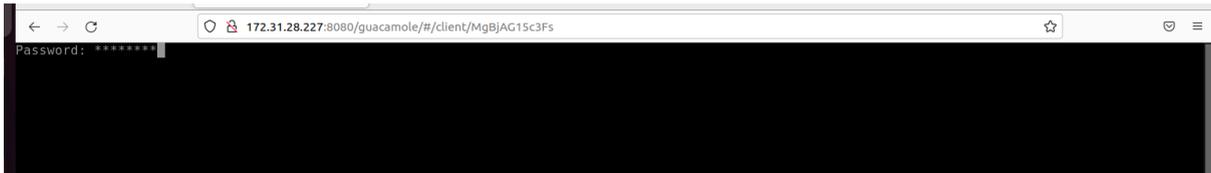
CONNEXIONS RÉCENTES test2



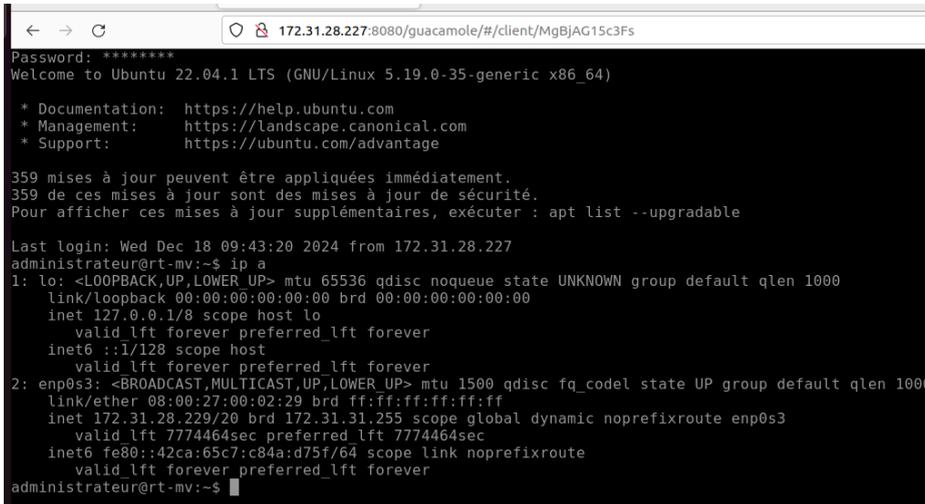
gavin_ssh

Terminal
TOUTES LES CONNEXIONS Filtre

> gavin_ssh
> valentin

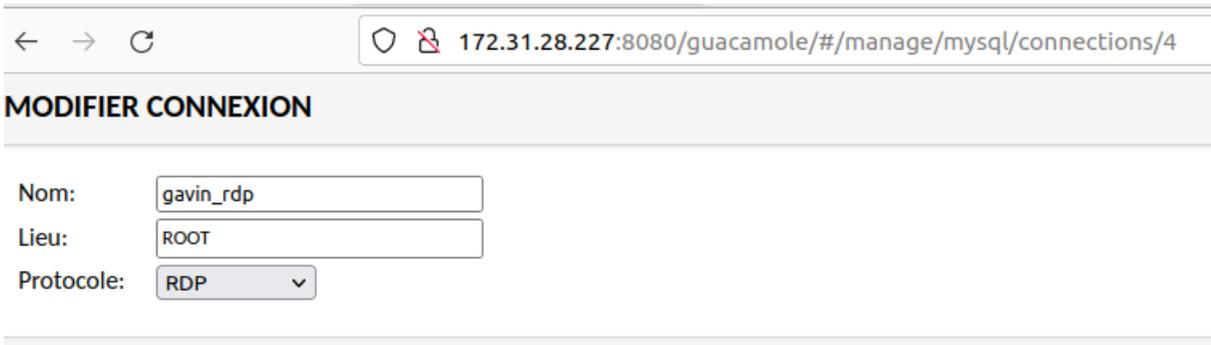


Rentrer les identifiants ici : - administrateur
- progr00



Connexion en RDP:

Nous allons maintenant nous connecter en RDP afin d'avoir la machine virtuelle au format interface Graphique :



Créer une nouvelle connexion et utiliser le protocole RDP
Remettre l'adresse IP et le port par défaut du protocole RDP : 3389

PARAMÈTRES

Réseau

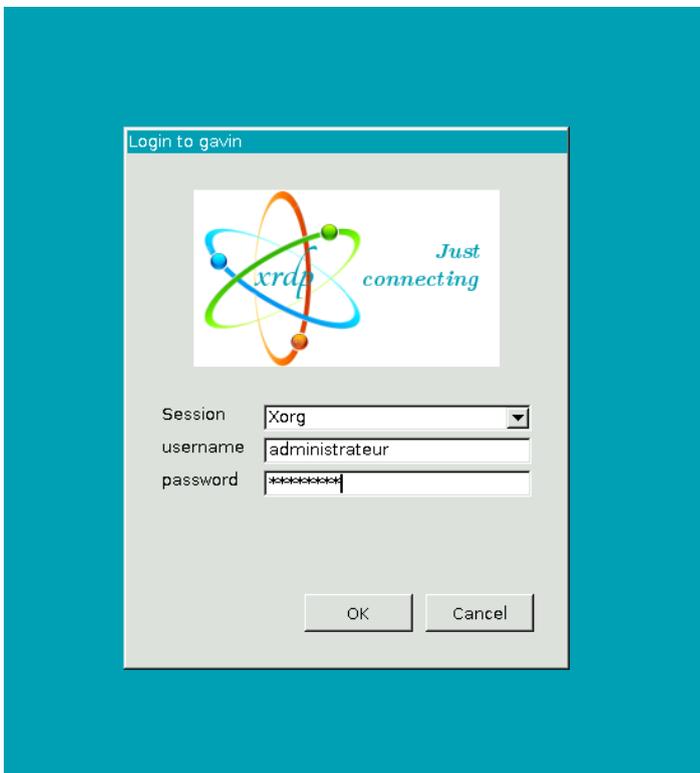
Nom d'hôte:	<input type="text" value="172.31.28.229"/>
Port:	<input type="text" value="3389"/>

Authentification

Sur la machine virtuelle installer le paquet xrdp:

- `sudo apt install xrdp`
- `sudo systemctl start xrdp`
- `sudo systemctl status xrdp`

Maintenant quand nous essayons la connexion il nous demande de nous connecter à la machine:



Sur la machine cliente faire la commande :

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

```
administrateur@rt-mv:~$ sudo tail -f /var/log/xrdp.log
[20241218-09:51:36] [INFO ] xrdp_wm_log_msg: sesman connect ok
[20241218-09:51:36] [INFO ] sesman connect ok
[20241218-09:51:36] [INFO ] sending login info to session manager, please wait..
.
[20241218-09:51:36] [INFO ] xrdp_wm_log_msg: login successful for display 10
[20241218-09:51:36] [INFO ] login successful for display 10
[20241218-09:51:36] [INFO ] loaded module 'libxup.so' ok, interface size 10296,
version 4
[20241218-09:51:36] [INFO ] started connecting
[20241218-09:51:36] [INFO ] lib_mod_connect: connecting via UNIX socket
[20241218-09:51:36] [INFO ] lib_mod_log_peer: xrdp_pid=4300 connected to X11rdp_
pid=4303 X11rdp_uid=1000 X11rdp_gid=1000 client_ip=::ffff:172.31.28.227 client_p
ort=54388
[20241218-09:51:36] [INFO ] connected ok
```

On peut voir que nous arrivons à nous connecter à la machine en RDP

Ressources transversales:

Phase 6 : Réflexion sur les aspects de développement durable et de transition énergétique: 5 points bonus

Synthesis in English

The virtualization project involves setting up a hypervisor on a dedicated server to run virtual machines for educational purposes. The main objective is to provide a remote-access web interface, allowing students to interact with virtual environments from anywhere. Additionally, we focus on integrating security measures such as authentication and access control to ensure secure usage. A thorough documentation and procedure for maintenance will be provided to ensure long-term sustainability and ease of use.

The project is divided into several phases, starting with planning and installation of the hypervisor, followed by the creation of virtual machines tailored to specific software and environments needed for the practical tasks. The remote-access interface will be implemented next, with careful consideration of security and user management. Testing and validation will be done to ensure performance and security, and the final deliverable will be a complete, functional virtualized infrastructure ready for use in educational settings.

In conclusion, virtualization is not only an effective tool for optimizing educational resources but also a step toward more energy-efficient and environmentally friendly computing.