PSMM

Installation du paquet nginx sur la machine serveur:

Nginx sudo apt install nginx

Installation des paquets sur la machine client:

MySQL/MariaDB (client uniquement) :

sudo apt update
sudo apt install mariadb-client

Client FTP : Il existe plusieurs clients FTP disponibles. Un choix populaire est LFTP :

sudo apt install ftp

Outils pour envoyer des mails en Python

Python a une bibliothèque intégrée pour envoyer des e-mails, mais vous aurez besoin de quelques dépendances supplémentaires pour une utilisation plus avancée. Voici comment les installer

sudo apt install python3-pip
pip3 install smtplib
pip3 install email

Pour exécuter toutes ces installations en une seule commande, vous pouvez utiliser

sudo apt update && sudo apt install -y mariadb-client lftp python3-pip && pip3 install smtplib email

Après avoir exécuté ces commandes, vous aurez :

- Le client MySQL/MariaDB
- LFTP comme client FTP
- Les bibliothèques Python nécessaires pour envoyer des e-mails

Si la machine ne reçoit pas d'ip vérifier le fichier `/etc/network/interfaces` :

auto ens33
iface ens33 inet dhcp

Création d'une clé SSH

Pour créer une clé SSH, suivez ces étapes sur votre machine cliente

ssh-keygen -t rsa -b 4096 -C "votreemail@exemple.com"

Appuyez sur Entrée pour accepter l'emplacement par défaut du fichier. Entrez un mot de passe pour la clé si vous le souhaitez (recommandé pour plus de sécurité).

Vous pouvez créer une nouvelle paire de clés sans phrase de passe, appuyez simplement sur Enter lorsqu'on vous demande la phrase de passe (deux fois) pour créer une clé sans phrase de passe.

ssh-copy-id -i ~/.ssh/id_rsa.pub utilisateur@adresse_ip_du_serveur

Remplacez "utilisateur" et "adresse_ip_du_serveur" par les informations appropriées.

```
dome@Plateflop-Client:~/.ssh$ sudo ssh-copy-id -i ~/.ssh/webid_rsa.pub dome@192.168.11.129
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/dome/.ssh/webid_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
dome@192.168.11.129's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'dome@192.168.11.129'"
and check to make sure that only the key(s) you wanted were added.

dome@Plateflop-Client:~/.ssh$
```

Configurer SSH côté serveur

Éditez le fichier de configuration SSH :

sudo nano /etc/ssh/sshd_config

Assurez-vous que les lignes suivantes sont présentes et non commentées :
PermitRootLogin no
PubkeyAuthentication yes

AuthorizedKeysFile .ssh/authorized_keys

désactiver l'authentification par mot de passe (recommandé une fois que l'authentification par clé fonctionne)

PasswordAuthentication no

Configurer SSH côté client

Modifiez votre fichier ~/.ssh/config pour utiliser cette nouvelle clé:

```
Host 192.168.10.113
        User monitor
        IdentityFile ~/.ssh/domid_rsa_nopw
        IdentitiesOnly yes
```

Changeons le propriétaire et permissions du fichier config
sudo chown domenico:domenico ~/.ssh/config
chmod 600 ~/.ssh/config

La configuration dans le fichier `~/.ssh/config` sert à plusieurs choses importantes.
Simplification des commandes SSH : Au lieu de taper `ssh monitor@192.168.10.113`, vous pouvez simplement taper `ssh 192.168.10.113`. SSH utilisera automatiquement le nom d'utilisateur spécifié dans le fichier config.

1. Spécification de la clé à utiliser : La ligne `IdentityFile ~/.ssh/domid_rsa` indique à SSH quelle clé privée utiliser pour cette connexion spécifique. Cela est utile si vous avez plusieurs clés pour différents serveurs.
2. Limitation des tentatives de clés : `IdentitiesOnly yes` force SSH à n'utiliser que la clé spécifiée, ce qui peut accélérer la connexion et éviter des tentatives inutiles avec d'autres clés.
3. Possibilité d'ajouter d'autres options : Vous pouvez ajouter d'autres options spécifiques à cette connexion, comme des configurations de port, de compression, etc.
4. Organisation et gestion : Cela vous permet de gérer facilement les configurations pour différents serveurs dans un seul fichier.
5. Méthode à utiliser si plusieurs clé sont enregistré sur la machine client

Utiliser ssh-agent.

Pour éviter d'avoir à entrer la phrase de passe à chaque connexion, nous pouvons utiliser ssh-agent.

Démarrez ssh-agent (s'il n'est pas déjà en cours d'exécution)

```
eval $(ssh-agent -s)
```

Ajoutez votre clé à ssh-agent

```
ssh-add ~/.ssh/domid_rsa
```

Vous devrez entrer la phrase de passe une fois.

Maintenant, essayez de vous connecter

```
ssh monitor@192.168.10.113
```

Vous ne devriez plus avoir à entrer la phrase de passe

Pour rendre cela permanent, vous pouvez ajouter les lignes suivantes à votre fichier ~/.bashrc ou ~/.profile

```
if  [ -z "$SSH_AUTH_SOCK" ] ; then
        eval $(ssh-agent -s)
        ssh-add ~/.ssh/domid_rsa
fi
```

Cela démarrera automatiquement ssh-agent et ajoutera votre clé à chaque fois que vous ouvrirez un nouveau terminal.

Alternativement à ces deux dernières méthodes , si vous préférez ne pas utiliser ssh-agent, vous pouvez créer une nouvelle paire de clés sans phrase de passe.

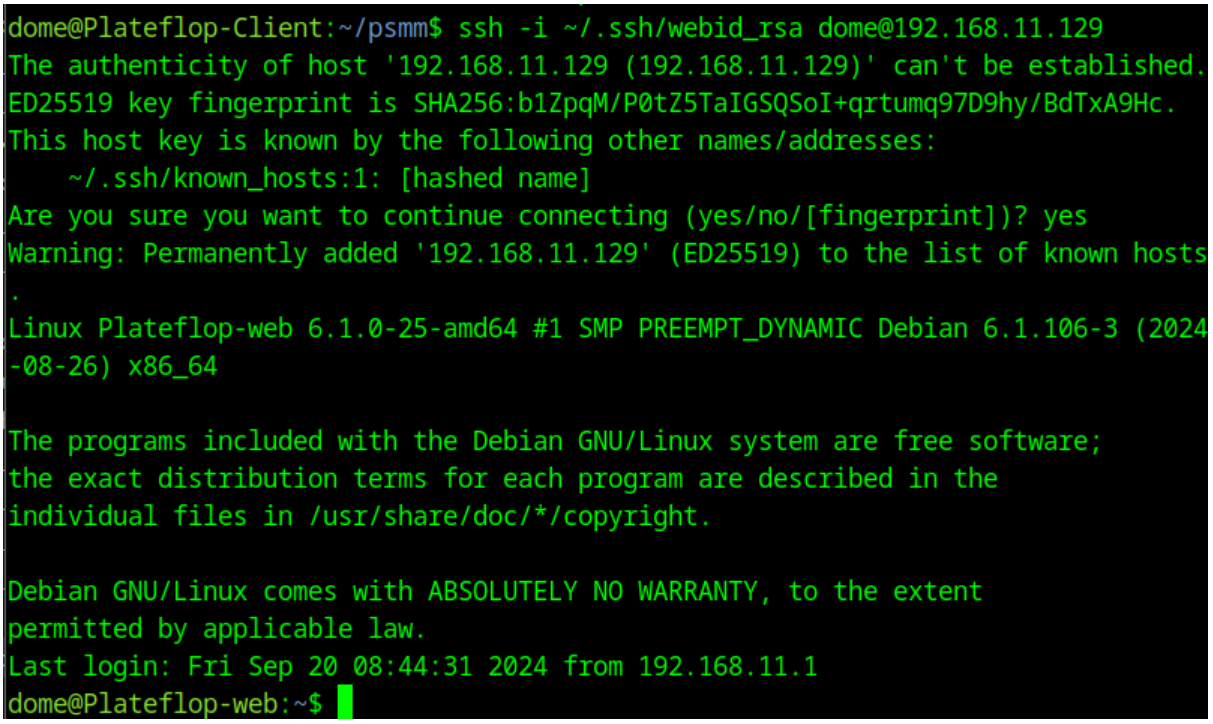
Appuyez simplement sur Enter lorsqu'on vous demande la phrase de passe (deux fois) pour créer une clé sans phrase de passe.

Redémarrer le service SSH :

sudo systemctl restart sshd

Tester la connexion:

ssh -i ~/.ssh/webid_rsa dome@192.168.11.129



Vérifier le pare-feu : Assurez-vous que le port SSH (généralement 22) est ouvert dans le pare-feu du serveur.

Après avoir suivi ces étapes, votre serveur devrait être configuré pour accepter votre clé SSH. Si vous rencontrez encore des problèmes, les messages d'erreur dans les logs du serveur (/var/log/auth.log) nous fourniront plus d'informations pour résoudre le problème.

Installation environnement virtuel Python pour exécuter les script

sudo apt install -y python3-venv
python3 -m venv ~/myenv
source ~/myenv/bin/activate

```
domenico@cli-domenico:~$ sudo apt install -y python3-venv
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Le paquet suivant a été installé automatiquement et n'est plus nécessaire :
  linux-image-6.1.0-22-amd64
Veuillez utiliser « sudo apt autoremove » pour le supprimer.
Les NOUVEAUX paquets suivants seront installés :
  python3-venv
0 mis à jour, 1 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 1 200 o dans les archives.
Après cette opération, 6 144 o d'espace disque supplémentaires seront utilisés.
Réception de :1 http://deb.debian.org/debian bookworm/main amd64 python3-venv amd64 3.11.2-1+b1 [1 200 B]
1 200 o réceptionnés en 0s (40,4 ko/s)
Sélection du paquet python3-venv précédemment désélectionné.
(Lecture de la base de données... 55731 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../python3-venv_3.11.2-1+b1_amd64.deb ...
Dépaquetage de python3-venv (3.11.2-1+b1) ...
Paramétrage de python3-venv (3.11.2-1+b1) ...
domenico@cli-domenico:~$ python3 -m venv ~/myenv
domenico@cli-domenico:~$ source ~/myenv/bin/activate
(myenv) domenico@cli-domenico:~$
```

Installation des paquets Pyton nécessaires

pip install paramiko
pip install paramiko mysql-connector-python

Activer l'environnement virtuel

chaque fois qu'on utilise un script

source ~/myenv/bin/activate
python votre_script.py

Pour demmarrer le script “ssh_login.py”

python3 ssh_login.py 192.168.10.113 monitor ~/.ssh/domid_rsa "df"

```
(myenv) domenico@cli-domenico:~/psmm$ python3 ssh_login.py 192.168.10.113 monitor ~/.ssh/domid_rsa "df"
Entrez la phrase de passe pour la clé (laissez vide si pas de phrase de passe) :
Sortie de la commande:
Sys. de fichiers blocs de 1K Utilisé Disponible Uti% Monté sur
udev                  983472       0     983472   0% /dev
tmpfs                 201444     844     200600   1% /run
/dev/sda1           15421320 2167724   12448428  15% /
tmpfs                1007204       0    1007204   0% /dev/shm
tmpfs                   5120       0       5120   0% /run/lock
tmpfs                 201440       0     201440   0% /run/user/0
tmpfs                 201440       0     201440   0% /run/user/1000
```

Pour demmarrer le script “ssh_login_sudo.py”

python3 ssh_login_sudo.py 192.168.10.113 monitor ~/.ssh/domid_rsa "apt update"

1. Configuration du serveur Nginx

1.1. Installation de Nginx

Si Nginx n'est pas déjà installé sur votre serveur:

```
sudo apt update
sudo apt install nginx
```

Vérifiez que Nginx est en cours d'exécution:

```
sudo systemctl status nginx
```

1.2. Configuration de SSH pour l'authentification par clé

Sur votre machine cliente, générez une paire de clés SSH si ce n'est pas déjà fait:

```
ssh-keygen -t rsa -b 4096 -C "votre_email@example.com"
```

Copiez la clé publique sur le serveur:

```
ssh-copy-id utilisateur@adresse_ip_serveur
```

Sur le serveur, éditez le fichier de configuration SSH:
```
sudo nano /etc/ssh/sshd_config
```

Assurez-vous que les lignes suivantes sont présentes et non commentées:

```
PubkeyAuthentication yes
AuthorizedKeysFile          .ssh/authorized_keys .ssh/authorized_keys2
PasswordAuthentication no
```

Redémarrez le service SSH:

```
sudo systemctl restart sshd
```

Configurer un site Nginx avec authentification

Éditez le fichier de configuration Nginx:

```
sudo nano /etc/nginx/nginx.conf
```
```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
error_log /var/log/nginx/error.log;
include /etc/nginx/modules-enabled/*.conf;

events {
        worker_connections 768;
        # multi_accept on;
}

http {

        ##
        # Basic Settings
```

```
        ##

        sendfile on;
        tcp_nopush on;
        types_hash_max_size 2048;
        # server_tokens off;

        # server_names_hash_bucket_size 64;
        # server_name_in_redirect off;

        include /etc/nginx/mime.types;
        default_type application/octet-stream;


        ##
        # SSL Settings
        ##

        ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref:
POODLE
        ssl_prefer_server_ciphers on;

        ##
        # Logging Settings
        ##

        access_log /var/log/nginx/access.log;

        ##
        # Gzip Settings
        ##

        gzip on;

        # gzip_vary on;
        # gzip_proxied any;
        # gzip_comp_level 6;
        # gzip_buffers 16 8k;
        # gzip_http_version 1.1;
        # gzip_types text/plain text/css application/json application/javascript text/xml
application/xml application/xml+rss text/javascript;

        ##
        # Virtual Host Configs
```

```
        ##

        include /etc/nginx/conf.d/*.conf;
        include /etc/nginx/sites-enabled/*;
}


#mail {
#       # See sample authentication script at:
#       # http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript
#
#       # auth_http localhost/auth.php;
#       # pop3_capabilities "TOP" "USER";
#       # imap_capabilities "IMAP4rev1" "UIDPLUS";
#
#       server {
#               listen   localhost:110;
#               protocol   pop3;
#               proxy           on;
#       }
#
#
#       server {
#               listen   localhost:143;
#               protocol   imap;
#               proxy           on;
#       }
#}
```

Éditez le fichier de configuration du stie:

```
sudo nano /etc/nginx/sites-available/mon-site.conf
```

```
server {
        listen 80;
        server_name srv-web.homelab.lan;
        return 301 https://$server_name$request_uri;
}

server {
        listen 443 ssl;
        server_name srv-web.homelab.lan;

        ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt;
```

```
        ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key;

        error_log  /var/log/nginx/monsite.error.log;
        access_log /var/log/nginx/monsite.acess.log;
        root /var/www/mon-site;
        index index.html;

        default_type text/html;

        location / {
                try_files $uri $uri/ =404;
                auth_basic "Zone restreinte";
                auth_basic_user_file /etc/nginx/.htpasswd;
        }
}
```

Redémarrez Nginx:

```
sudo systemctl restart nginx
```

Configurer l'authentification basique

Installez apache2-utils :

```
sudo apt install apache2-utils
```

Créez un fichier de mots de passe

```
sudo htpasswd -c /etc/nginx/.htpasswd utilisateur
```

Remplacez "utilisateur" par le nom d'utilisateur souhaité et entrez un mot de passe quand on vous le demande.

Modifiez la configuration Nginx

```
sudo nano /etc/nginx/sites-available/mon-site
```

Ajoutez les lignes suivantes dans le bloc `server`

```
auth_basic "Zone restreinte";
auth_basic_user_file /etc/nginx/.htpasswd;
```

Vérifiez et redémarrez Nginx :

```
sudo nginx -t
sudo systemctl restart nginx
```

Ajouter l'utilisateur au groupe approprié :

Généralement, les fichiers de log Nginx appartiennent au groupe "adm" ou "nginx". Ajoutez l'utilisateur "monitor" à ce groupe :

sudo usermod -a -G adm monitor
ou
sudo usermod -a -G nginx monitor

Modifier les permissions du fichier :

Si l'ajout au groupe ne suffit pas, vous pouvez modifier directement les permissions du fichier :

sudo chmod 644 /var/log/nginx/monsite.error.log
sudo chown root:adm /var/log/nginx/monsite.error.log

Démarrer une session depuis Vm Client (sans interface)

curl -k -u user:user https://monsite.lan

```
<!DOCTYPE html>

<html lang="fr">
<head>
        <meta charset="UTF-8">
        <title>Mon Site Web</title>
</head>

<body>
<h1>Bienvenue sur mon site web sécurisé!</h1>
<p>Ceci est un exemple de contenu pour mon site web.</p>
</body>
</html>
domenico@cli-domenico:~$
```

Vérifier les log

Pour voir les dernières lignes du log d'erreur :
sudo tail -f /var/log/nginx/monsite.error.log
Pour voir les dernières lignes du log d'accès :
sudo tail -f /var/log/nginx/monsite.acess.log

Demmarrer le script “ssh_web_error.py”

python3 ssh_web_errors.py

Se connecter à MySQL

Depuis une VM client nous allons nous connecter à Base de Données pour vérifier les log par ligne de commande.

1. Ouvrez un terminal
2. Connectez-vous à MySQL avec la commande :

   mysql -u root -p

Remplacez "root" par votre nom d'utilisateur MySQL si nécessaire. Entrez votre mot de passe quand il vous est demandé.

Afficher les bases de données

Une fois connecté à MySQL, vous pouvez :

1. Lister toutes les bases de données :

   SHOW DATABASES;

Sélectionner la base de données que vous voulez examiner :

USE nom_de_votre_base;

Afficher le contenu de la base de données

Pour voir le contenu de votre base de données, vous pouvez :

1. Lister toutes les tables :

   ```
   SHOW TABLES;
   ```

2. Afficher la structure d'une table spécifique :

   ```
   DESCRIBE nom_de_la_table;
   ```

3. Afficher le contenu d'une table :

   ```
   SELECT * FROM nom_de_la_table;
   ```

Si vous voulez limiter le nombre de lignes affichées, ajoutez LIMIT :

```
SELECT * FROM nom_de_la_table LIMIT 10;
```

Quitter MySQL

Pour quitter l'interface MySQL, tapez :

```
EXIT;
```

Server_Name choses à retenir

N'oubliez pas que ces commandes sont sensibles à la casse. Utilisez le point-virgule à la fin de chaque commande MySQL

Quelques points importants à noter pour utiliser Gmail :

1. Vous devrez peut-être activer "l'accès des applications moins sécurisées" dans les paramètres de votre compte Gmail. Cependant, cette option n'est pas recommandée pour la sécurité à long terme.
2. Une meilleure approche serait d'utiliser un "mot de passe d'application" spécifique pour ce script. Voici comment le configurer :
   - Allez dans les paramètres de votre compte Google
   - Activez l'authentification à deux facteurs si ce n'est pas déjà fait
   - Dans la section "Sécurité", cherchez "Mots de passe des applications"
   - Générez un nouveau mot de passe d'application pour ce script
3. Utilisez ce mot de passe d'application généré comme `smtp_password` dans la configuration.

Cronetab choses à retenir

Pour planifier l'exécution toutes les 3 heures, vous pouvez utiliser cron. Voici comment configurer cron :

1. Ouvrez votre crontab en exécutant :

```
crontab -e
```

Choisissez l'option 1 pour nano, qui est généralement plus facile à utiliser.
Une fois dans l'éditeur, vous devez ajouter une nouvelle ligne pour votre tâche cron. La ligne correcte devrait ressembler à ceci :

```
0 */3 * * * /usr/bin/python3 /home/domenico/psmm/ssh_cron_backup.py
```

Cette ligne signifie : exécuter la commande toutes les 3 heures, à la minute 0.

Après avoir ajouté cette ligne, sauvegardez et quittez l'éditeur :

- Pour nano : appuyez sur Ctrl+X, puis Y, puis Enter.

Vous devriez voir un message confirmant que le crontab a été installé.

“crontab: installing new crontab”

Après ces étapes, votre tâche cron devrait être correctement configurée. Pour vérifier, vous pouvez exécuter `crontab -l` pour voir le contenu de votre crontab.

Assurez-vous également que votre script `ssh_cron_backup.py` est bien situé dans le répertoire `/home/domenico/psmm/` et qu'il a les permissions d'exécution :

```
chmod +x /home/domenico/psmm/ssh_cron_backup.py
```

cpu_command = "top -bn2 -d1 | grep 'Cpu(s)' | tail -n1"
"grep 'cpu ' /proc/stat | awk '{usage=($2+$4)*100/($2+$4+$5)} END {print usage}'")