



REPOBLIKAN'I MADAGASIKARA
Fitiavana - Tanindrazana – Fandrosoana
MINISTÈRE DE L'ENSEIGNEMENT
SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE



UNIVERSITÉ D'ANTSIRANANA
ÉCOLE SUPÉRIEURE POLYTECHNIQUE D'ANTSIRANANA

Mention : STIC
Groupe 3

Rapport de mini projet

Installation et configuration d'un Serveur Proxy Squid sur Debian avec VirtualBox et Docker

Réalisé par :

NOMENJARA Judalien
RAKOTONANDRASANA Danio
RATODISO Claudien Fridelin Bebe

Responsable :

RANDRIAMASINORO Njakarison Menja

Année universitaire : 2024 – 2025

Table des matières

1	Création de l'environnement de travail	1
1.1	Installation de VirtualBox et création de la machine virtuelle	1
1.2	Installation de Debian	1
2	Installation et configuration de Squid	1
2.1	Installation de Squid	1
2.2	Configuration de base	2
3	Automatisation avec un script Bash	2
3.1	Script : <code>install_squid_uto.sh</code>	2
3.2	Exécution	4
4	Tests et validation	5
4.1	Test via navigateur ou curl	5
4.2	Analyse des logs	5
5	Conteneurisation avec Docker	5
5.1	Conteneurisation de l'installation de Squid sous Docker en créant un Dockerfile.	5
5.1.1	Création du Dockerfile	6
5.1.2	Construction et exécution	6
5.2	Test dans navigateur ou via curl	7
5.3	Documentation sur les installations et l'utilisation de l'image Docker créé .	7

Introduction

Dans un contexte numérique où la maîtrise des flux Internet et la sécurisation des accès réseau sont devenues fondamentaux, les serveurs proxy occupent une place stratégique dans les infrastructures informatiques. Ce mini-projet a pour objectif la mise en place d'un serveur proxy basé sur Squid, un outil libre et puissant, permettant de filtrer, surveiller et optimiser les connexions Internet.

Le projet s'inscrit dans une démarche pédagogique visant à développer les compétences techniques en administration système Linux (Debian), en script Bash pour l'automatisation, et en virtualisation ou conteneurisation avec VirtualBox et Docker. Il couvre l'installation manuelle de Squid, sa configuration personnalisée, l'automatisation via un script shell, des tests de validation, et enfin la conteneurisation complète du service.

1 Création de l'environnement de travail

1.1 Installation de VirtualBox et création de la machine virtuelle

VirtualBox est un logiciel de virtualisation permettant de créer et gérer des machines virtuelles. On installe VirtualBox depuis <https://www.virtualbox.org>.

On crée ensuite une nouvelle machine virtuelle avec :

- Type : Linux
- Version : Debian (64-bit)
- RAM : 2 Go minimum
- Disque dur : 10 Go minimum (VDI, dynamique alloué)

1.2 Installation de Debian

Debian est une distribution GNU/Linux stable, libre et open-source, très utilisée comme base pour les serveurs.

Voici quelques étapes pour installer le Debian :

- On télécharge l'image ISO officielle depuis <https://www.debian.org>.
- On monte l'ISO dans VirtualBox.
- On suit l'installation standard (locale, mot de passe root, etc.).
- Et on choisit le mode réseau **NAT** ou **Bridge** pour permettre l'accès à Internet.

2 Installation et configuration de Squid

2.1 Installation de Squid

Squid est un serveur proxy HTTP/HTTPS permettant de filtrer les connexions, contrôler l'accès à Internet et effectuer de la mise en cache pour accélérer les réponses.

```
sudo apt update
sudo apt install squid -y
sudo cp /etc/squid/squid.conf /etc/squid/squid.conf.backup
```

2.2 Configuration de base

L'éditeur de texte nano permet d'éditer le fichier :

```
sudo nano /etc/squid/squid.conf
```

Voici le contenu du fichier squid.conf avec les autorisations et les configurations suivantes :

```
http_port 3128                #Port d'écoute du proxy
acl localnet src 10.0.2.0/24  #ACL (Access Control List) pour le
    réseau local
http_access allow localnet    # Autorisation de l'accès pour
    localnet
http_access deny all          # Refus de tous les autres
visible_hostname proxy-server # Nom visible du proxy
```

Après, on redémarrer le service :

```
sudo systemctl restart squid
```

Et on verifie si Squid écoute bien sur le port défini :

```
ss -tuln | grep 3128
```

3 Automatisation avec un script Bash

3.1 Script : `install_squid_automato.sh`

```
\#!/bin/bash

echo "-----"

echo "----_Installation_&_Configuration_de_Squid_----"

echo "-----"

\# 1. Mise à jour du système

echo "Mise_à_jour_du_système..."

apt update -y \&& apt upgrade -y

\# 2. Installation de Squid et apache2-utils

echo "Installation_de_Squid_et_apache2-utils..."

apt install -y squid apache2-utils

\# 3. Sauvegarde de l'ancien fichier de configuration

echo "Sauvegarde_de_la_configuration_originale..."

cp /etc/squid/squid.conf /etc/squid/squid.conf.backup
```

```

\# 4. Nouvelle configuration de Squid (sans authentification)

echo "Configuration de Squid..."

cat > /etc/squid/squid.conf << EOF

\# -----

\# Configuration Squid (sans authentification)

\# -----

http\_port 3128

acl localnet src 10.0.2.0/24

http\_access allow localnet

http\_access deny all

visible\_hostname proxy-server

EOF

\# 5. Redémarrage de Squid

echo "Redémarrage de Squid..."

systemctl restart squid

\# 6. Vérification de l'installation

echo "-----"

echo "Vérification de l'installation..."

echo "-----"

\# Vérifier si Squid est installé

if command -v squid > /dev/null; then

echo "Squid est installé."

else

echo "Squid n'est pas installé."

fi

```

```

\# Vérifier si le service Squid tourne

if systemctl is-active --quiet squid; then

echo "Le service Squid est actif."

else

echo "Le service Squid ne fonctionne pas."

fi

\# Vérifier si le port 3128 est ouvert

if ss -ltnp | grep -q ':3128'; then

echo "Le port 3128 est bien ouvert."

else

echo "Le port 3128 n'est pas ouvert."

fi

\# Vérifier la configuration

if grep -q "http\_port\_3128" /etc/squid/squid.conf \&\& grep -q "
http\_access\_allow" /etc/squid/squid.conf; then

echo "Le fichier squid.conf est bien configuré."

else

echo "Problème dans squid.conf."

fi

echo "-----"

echo "Vérification terminée."

echo "Configurez votre navigateur avec le proxy : IP\_MACHINE:3128
"

echo "-----"

```

3.2 Exécution

```

chmod +x install_squid.sh
sudo ./install_squid.sh

```

Ce script garantit une configuration rapide et cohérente sur plusieurs machines.

4 Tests et validation

4.1 Test via navigateur ou curl

On fait un Test sur le proxy via curl ou navigateur

```
curl -x http://localhost:3128 http://example.com
```

En faisant test sur le proxy, on tape ce commande et on observe un code HTML et CSS.

Mais en faisant un test dans un navigateur, on configure un proxy HTTP manuellement comme suit :

- **Hôte** : localhost ou l'adresse IP locale (pour notre cas, IP locale est 10.0.2.15)
- **Port** : 3128

Et on observe une interface web comme suit,



4.2 Analyse des logs

Squid garde une trace des requêtes dans ses journaux :

```
cat /var/log/squid/access.log
```

Les logs permettent de contrôler l'activité du proxy, analyser les requêtes filtrées ou autorisées.

5 Conteneurisation avec Docker

5.1 Conteneurisation de l'installation de Squid sous Docker en créant un Dockerfile.

Voici les étapes pour installer Docker sur Debian

```
sudo apt update
sudo apt install -y docker.io
sudo systemctl enable docker
sudo systemctl start docker
```

Et on verifie si le Docker est bien installé et fonctionne normal avec cette commande :

```
docker --version
```

Puis on crée un répertoire de travail squid-docker

```
mkdir /etc/squid-docker
cd ~/squid-docker
```

Et puis on crée un fichier nommé squid.conf (pour Docker) qui contient la configuration de squid, dans ce même repertoire :

```
nano squid.conf
```

Et on configure ce fichier par ce script (pour Docker)

```
http_port 3128
acl localnet src 0.0.0.0/0
http_access allow localnet
http_access deny all
visible_hostname squid-docker
```

5.1.1 Création du Dockerfile

On crée un fichier Dockerfile dans le même répertoire squid-docker.

Un Dockerfile contient les instructions suivantes pour construire l'image d'un conteneur.

```
nano Dockerfile
```

Voici l'étape de création de ce fichier Dockerfile

```
FROM debian:12
LABEL maintainer="TonNom <ton.email@example.com>"
LABEL description="Image Docker de Squid Proxy simple"
\# Installer Squid
RUN apt update \&\& apt install -y squid \&\& apt clean
\# Copier la configuration personnalisée
COPY squid.conf /etc/squid/squid.conf
\# Ouvrir le port du proxy
EXPOSE 3128
\# Lancer squid au démarrage du conteneur
CMD ["squid", "-N", "-d", "1"]
```

5.1.2 Construction et exécution


```
cd /etc/squid-docker

docker build -t squid-proxy .
docker run -d --name squid-container -p 3128:3128 squid-proxy
```

5.2 Test dans navigateur ou via curl

Avant tout, il faut bien vérifier si le conteneur tourne ou non :

```
docker ps
```

Et puis on fait un test dans le navigateur comme on a fait pendant le test du serveur proxy mais cette fois avec la conteneurisation :

- On va dans les paramètres réseau → proxy manuel
 - Adresse : localhost
 - Port : 3128
 - Teste `http://example.com`
- Pour autre vérification

```
curl -x http://localhost:3128 http://example.com
```

Et après la page HTML s'affiche, ça veut dire que le proxy fonctionne bien

5.3 Documentation sur les installations et l'utilisation de l'image Docker créé

Pour permettre un déploiement rapide et portable via Docker, il nous faut une conteneurisation du Squid Proxy. Pour ce faire, on a besoin deux fichiers qui sont Dockerfile et squid.conf. Le fichier Docker contient : l'image de base qui est Debian 12, l'installation de Squid, copie d'une configuration minimale, exposition du port 3128.

Voici les étapes et les commandes utilisés pour le lancer, en résumé :

```
\# Construire l image
docker build -t squid-proxy .
\# Lancer le conteneur
docker run -d --name squid-container -p 3128:3128 squid-proxy
\# Vérifier si ça fonctionne
curl -x http://localhost:3128 http://example.com
```

Et il est bien à noter que notre conteur fonctionne sans authentification et peut être adapté pour l'intégrer à un réseau interne.

Conclusion

Ce mini-projet a permis d'explorer et de maîtriser plusieurs aspects fondamentaux de l'administration réseau et système, en s'appuyant sur un outil professionnel et libre qui est Squid. À travers les différentes étapes, l'installation sur une machine Debian, jusqu'à la création d'une image Docker fonctionnelle, nous avons mis en œuvre une solution efficace, portable et automatisée de proxy HTTP.

Les tests réalisés ont confirmé le bon fonctionnement du serveur, avec des règles d'accès configurées et des connexions filtrées selon des politiques précises. L'automatisation par script Bash facilite la reproductibilité de l'installation, et la conteneurisation via Docker assure une grande souplesse de déploiement, quelles que soient les plateformes.

Ce projet constitue une base solide pour des extensions futures telles que : l'intégration d'une authentification utilisateur, la mise en place d'un cache avancé, le filtrage par domaine, ou encore l'administration à distance via une interface Web.