

# INSTALLATION DE SQUID SOUS RASPBERRY PI

Raspberry Pi – Debian 12  
**Configuration de base**

Tutoriel **SQUID** - RASPBERRY PI  
30 décembre 2024

David GOÏTRÉ

## Table des matières

Introduction .....	1
1. Pré requis .....	1
2. Paramétrage de connexion au serveur .....	1
3. Paramétrage du serveur .....	2
4. Installation de Squid.....	3
5a. Les contrôles d'accès.....	3
5b. Configuration l'authentification basée sur IP .....	4
5c. Fichier de configuration de base .....	5
6. Configuration de Squid pour anonymiser le trafic.....	6
7. Intégrité de la configuration de squid.....	6
8. Mise à jour de Squid.....	7
9. Désinstallation de Squid.....	7
10. Outils de Squid .....	7
11. Paramétrage du proxy dans le navigateur .....	8
12. Liens annexes .....	8
13. Commandes RaspberryPi .....	8
14. Conclusion .....	9

## Introduction

Un serveur **Squid** est un serveur mandataire (proxy) et un **mandataire inverse** conçu pour relayer les protocoles FTP, HTTP, Gopher, et HTTPS.

Contrairement aux serveurs proxy classiques, un serveur Squid gère toutes les requêtes en un seul processus d'entrée/sortie asynchrone. C'est un logiciel libre distribué sous licence GNU GPL.

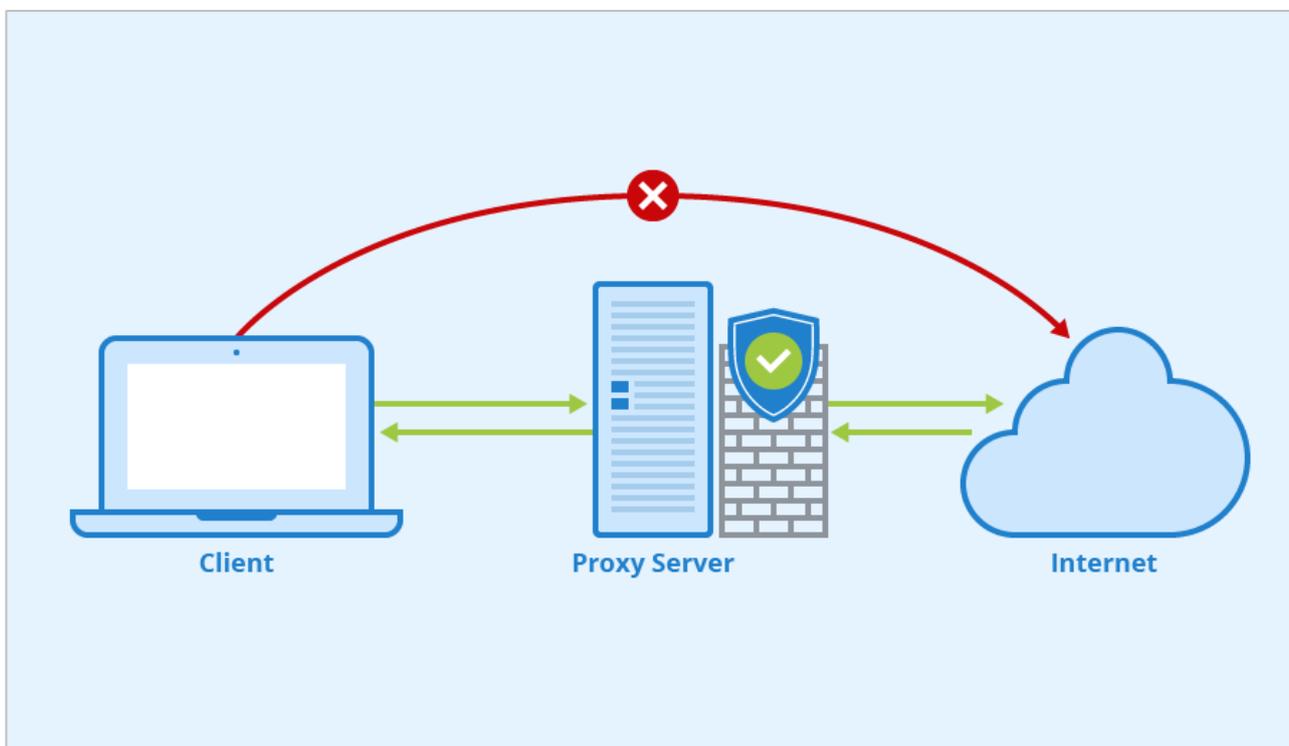
Squid garde les métadonnées et plus particulièrement les données les plus fréquemment utilisées en mémoire. Il conserve aussi en mémoire les requêtes DNS, ainsi que les requêtes ayant échoué. Les requêtes DNS sont non bloquantes.

## 1. Pré requis

On a besoin des différents matériels et logiciels pour la création d'un Serveur Squid avec un RaspberryPi.

- Un ou des PC client sous Windows
- Une Box (Free, Orange, Sfr...)
- Un Raspberry 3B+ avec [l'OS RaspbianOS](#) installé avec [Etcher](#)
- Le logiciel [Putty](#) pour se connecter en SSH au serveur
- Connaître l'interface réseau (eth0, br0, ens3...) via la commande : `ip a`  
Pour notre test c'est **l'interface eth0** qui sera utilisée

Voici un exemple de ce que l'on doit obtenir une fois le serveur **Squid** mise en place



## 2. Paramétrage de connexion au serveur

a) Le **SSH** est activé par défaut sur le serveur

b) Ouvrir **Putty** et se connecter au serveur avec les identifiants (par défaut **pi/raspberry**)

c) Exécuter la commande suivante pour mettre à jour et mettre à niveau les packages du système

```
# apt update && apt upgrade
```

### 3. Paramétrage du serveur

Avant d'aller plus loin, il nous faut connaître l'interface réseau de notre serveur **RaspberryPI** et lui attribuer une adresse IP fixe.

a) Lister les interfaces

```
# ip a # liste les interfaces  
# grep -i ethernet /var/run/dmesg.boot # liste les propriétés de l'interface
```

b) Définir une adresse IP fixe

```
# apt install nano # installe le logiciel nano  
# nano /etc/network/interfaces # ouvre le fichier des interfaces
```

c) Copier le texte ci-dessous dans le fichier **/etc/network/interface**

```
# Interface reseau de bouclage  
auto lo  
iface lo inet loopback  
# Interface reseau principale  
allow-hotplug eth0  
iface eth0 inet static  
address 192.xxx.xxx.xxx  
netmask 255.255.255.0  
gateway 192.xxx.xxx.xxx
```

d) Vérifier les DNS

```
# cat /etc/resolv.conf # affiche le contenu du fichier
```

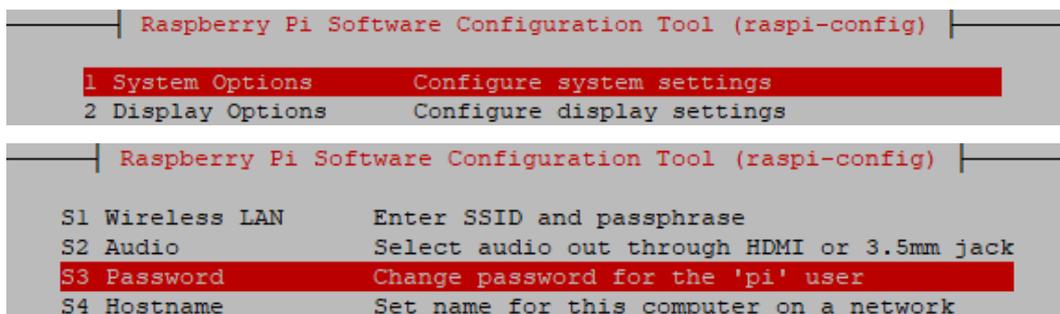
e) Redémarrer le serveur

```
# reboot
```

f) Modifier le mot de passe

```
$ raspi-config # ouvre l'utilitaire
```

Sélectionner le menu **System Options**, puis cliquer sur le menu **S3 Password** pour modifier le mot de passe et **S4 Hostname** pour modifier le nom du serveur.



```
Raspberry Pi Software Configuration Tool (raspi-config)  
1 System Options      Configure system settings  
2 Display Options     Configure display settings  
  
Raspberry Pi Software Configuration Tool (raspi-config)  
S1 Wireless LAN      Enter SSID and passphrase  
S2 Audio              Select audio out through HDMI or 3.5mm jack  
S3 Password           Change password for the 'pi' user  
S4 Hostname           Set name for this computer on a network
```

Sélectionner le menu **Localisation Options**. Dans la **liste déroulante** sélectionner **fr\_FR UTF-8** pour ajouter les paramètres linguistiques français au système.

## 4. Installation de Squid

Normalement, Squid fait déjà partie des dépôts de Debian, la simple commande **apt install** suffit.

a) Installation de squid

```
# apt update
# apt install squid -y
```

b) Vérification l'état du service Squid

```
# systemctl status squid
```

Résultat de la sortie

```
kungen@raspberrypi:~ $ systemctl status squid
● squid.service - Squid Web Proxy Server
   Loaded: loaded (/lib/systemd/system/squid.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-12-28 12:26:07 GMT; 17min ago
     Docs: man:squid(8)
  Process: 7266 ExecStartPre=/usr/sbin/squid --foreground -z (code=exited, status=0/SUCCESS)
 Main PID: 7270 (squid)
    Tasks: 5 (limit: 763)
       CPU: 3.810s
   CGroup: /system.slice/squid.service
           └─7270 /usr/sbin/squid --foreground -sYC
             └─7272 "(squid-1)" --kid squid-1 --foreground -sYC
               └─7273 "(logfile-daemon)" /var/log/squid/access.log
                 └─7274 "(unlinkd)"
                   └─7275 "(pinger)"

Dec 28 12:26:07 raspberrypi squid[7272]:      0 Objects expired.
Dec 28 12:26:07 raspberrypi squid[7272]:      0 Objects cancelled.
Dec 28 12:26:07 raspberrypi squid[7272]:      0 Duplicate URLs purged.
```

c) Vérification du port d'écoute 3128 de Squid

```
# netstat -plnt | grep 3128
```

Résultat de la sortie

```
tcp60      0      :::3128      :::*      LISTEN      50017/(squid-1)
```

## 5a. Les contrôles d'accès

Les possibilités de contrôler l'accès à Internet sont nombreuses, comme limiter cet accès à des plages horaires particulières, fournir des informations depuis le cache, autoriser seulement certains sites ou groupes de sites, etc... Squid dispose pour ces contrôles de deux types de composants : les éléments ACL (**Access Control List**) et la liste d'accès. Une liste d'accès, autorise ou refuse l'accès au service.

**Ci-dessous quelques-uns des plus importants éléments ACL**

- src : Source c-à-d. l'adresse IP du client
- dst : Destination c-à-d. l'adresse IP du serveur
- srcdomain : Source c-à-d. le nom de domaine du client
- dstdomain : Destination c-à-d. le nom de domaine du serveur
- time : Heure du jour et jour de la semaine
- url\_regex : Expression régulière décrivant une catégorie d'URL
- urlpath\_regex: Expression régulière décrivant un ensemble d'URL sans protocole ni nom d'hôte
- proxy\_auth : Procédé externe d'authentification d'un utilisateur
- maxconn : Nombre maximum de connexions pour une adresse IP cliente

Pour activer le contrôle, il faut d'abord définir un ensemble d'ACL et ensuite y appliquer des règles. Le format d'une ACL suit la syntaxe : `acl acl_element_name type_of_acl_element values_to_acl`

- **acl\_element\_name** peut être n'importe quel nom attribué par l'utilisateur à un élément ACL
- Deux éléments distincts ne peuvent avoir le même nom
- Chaque ACL est une liste de valeurs. Pendant la vérification, les valeurs multiples utilisent un OU logique. Autrement dit un élément ACL correspond si l'une des valeurs est reconnue
- Tous les éléments ACL ne sont pas utilisables avec tous les types de listes d'accès
- Différents éléments ACL occupent plusieurs lignes et Squid les amalgame en une seule liste

Différentes listes d'accès sont disponibles. Celles que nous utiliserons sont décrites ci-dessous :

- **http\_access** : Autorise les clients HTTP à accéder au port HTTP. C'est l'ACL primaire.
- **no\_cache** : Définit le cache pour les réponses aux requêtes

Une règle de liste d'accès comporte les mots **allow** ou **deny**, ce qui autorise ou refuse un service pour un élément ACL particulier ou pour un groupe d'éléments.

- Les règles sont vérifiées dans l'ordre où elles ont été écrites et se terminent dès qu'une correspondance a été établie
- Une ACL peut comporter plusieurs règles
- Si aucune correspondance n'est trouvée, l'action par défaut est l'inverse de la dernière règle
- Liste. Il est donc préférable d'être explicite sur l'action par défaut
- Tous les éléments d'une même entrée d'accès sont associés par un ET
- Les règles sont lues de haut en bas

## 5b. Configuration l'authentification basée sur IP

Il existe plusieurs façons de restreindre l'accès du client à Internet. Dans cette section, on va configurer Squid pour s'authentifier en fonction des clients. **Attention les règles ACL spécifiques doivent être placées avant les règles générales.**

a) Modifier le fichier de configuration `/etc/squid/squid.conf`

```
# nano /etc/squid/squid.conf
```

b) Ajout les lignes ci-dessous dans la section ACL pour autoriser tous les clients du réseau local

```
acl localnet src 192.168.1.0/24
http_access allow localnet
```

c) Ajout les lignes ci-dessous dans la section ACL pour autoriser les clients inscrits

```
acl client1 src 192.168.1.10
http_access allow 192.168.1.10
```

d) Redémarrer ensuite le service Squid pour appliquer les modifications

```
# systemctl restart squid
```

e) Démarrage automatique de Squid

```
# sudo systemctl enable squid
```

## 5c. Fichier de configuration de base

Exemple d'un fichier de configuration

```
#####Mémoire
cache_mem 512 MB
#####Anonymiser le trafic (chapitre 6)
request_header_access Allow allow all
...
#####Autorisation
acl localnet src 192.168.0.0/24
acl SSL_ports port 443          # https
acl SSL_ports port 8096        # jellyfin
acl Safe_ports port 80         # http
acl Safe_ports port 8096       # jellyfin
acl Safe_ports port 21         # ftp
acl Safe_ports port 443        # https
acl Safe_ports port 70         # gopher
acl Safe_ports port 210        # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280        # http-mgmt
acl Safe_ports port 488        # gss-http
acl Safe_ports port 591        # filemaker
acl Safe_ports port 777        # multiling http
#####Methode de connexion
acl CONNECT method CONNECT
#####Autoriser l'accès à cachemgr qu'à partir de localhost
http_access allow localhost manager
http_access allow localnet
http_access allow localhost
#####Interdire la connexion à des ports SSL non sécurisés et autres
http_access deny CONNECT !SSL_ports
http_access deny !Safe_ports
http_access deny manager
http_access deny all
#####Laisser les coredumps dans le premier cache dir
coredump_dir /var/spool/squid
#####Déclaration du dossier de cache utilisé par Squid et de la taille utile (ici 80%)
cache_effective_user proxy
cache_effective_group proxy
cache_dir ufs /CacheSquid 358 16 256
#####Désactiver le traçage de l'adresse IP source
via off
forwarded_for off
follow_x_forwarded_for deny all
request_header_access X-Forwarded-For deny all
header_access X_Forwarded_For deny all
http_access deny
```

## 6. Configuration de Squid pour anonymiser le trafic

On doit ajouter des règles pour masquer les adresses IP des clients des serveurs qui reçoivent le trafic de notre proxy HTTP Squid.

a) Toujours en éditant le fichier `/etc/squid/squid.conf`

```
# nano /etc/squid/squid.conf
```

b) Ajout des lignes ci-dessous à la fin du fichier

```
request_header_access Allow allow all
request_header_access Authorization allow all
request_header_access WWW-Authenticate allow all
request_header_access Proxy-Authorization allow all
request_header_access Proxy-Authenticate allow all
request_header_access Cache-Control allow all
request_header_access Content-Encoding allow all
request_header_access Content-Length allow all
request_header_access Content-Type allow all
request_header_access Date allow all
request_header_access Expires allow all
request_header_access Host allow all
request_header_access If-Modified-Since allow all
request_header_access Last-Modified allow all
request_header_access Location allow all
request_header_access Pragma allow all
request_header_access Accept allow all
request_header_access Accept-Charset allow all
request_header_access Accept-Encoding allow all
request_header_access Accept-Language allow all
request_header_access Content-Language allow all
request_header_access Mime-Version allow all
request_header_access Retry-After allow all
request_header_access Title allow all
request_header_access Connection allow all
request_header_access Proxy-Connection allow all
request_header_access User-Agent allow all
request_header_access Cookie allow all
request_header_access All deny all
```

c) Redémarrer ensuite le service Squid pour appliquer les modifications

```
# systemctl restart squid
```

## 7. Intégrité de la configuration de squid

Vérifier l'intégrité du fichier de configuration

```
# systemctl status squid.service
```

## 8. Mise à jour de Squid

Effectuer la mise à jour manuellement

```
# apt update
# apt upgrade
```

## 9. Désinstallation de Squid

Désinstaller Squid peut-être parfois nécessaire pour le réinstaller

```
# systemctl stop squid
# apt remove squid
```

## 10. Outils de Squid

Il est important de connaître la version de squid à des fins de bonnes configurations

a) Vérifier la version de Squid

```
# squid -v
```

Résultat de la sortie

```
Squid Cache: Version 5.7
Service Name: squid
Debian Linux
configure options: '--build=aarch64-linux-gnu' '--prefix=/usr' '--includedir=${prefix}/include' '--mandir=${prefix}/share/man' '--infodir=${prefix}/share/info' '--sysconfdir=/etc' '--localstatedir=/var' '--disable-option-checking' '--disable-silent-rules' '--libdir=${prefix}/lib/aarch64-linux-gnu' '--runstatedir=/run' '--disable-maintainer-mode' '--disable-dependency-tracking' 'BUILD_CXXFLAGS=-g -O2 -ffile-prefix-map=/build/reproducible-path/squid-5.7= -fstack-protector-strong -Wformat -Werror=format-security -Wno-error=deprecated-declarations -Wdate-time -D_FORTIFY_SOURCE=2 -Wl,-z,relro -Wl,-z,now' 'BUILD_CXX=g++' '--with-build-environment=default' '--enable-build-info=Debian Linux' '--datadir=/usr/share/squid' '--sysconfdir=/etc/squid' '--libexecdir=/usr/lib/squid' '--mandir=/usr/share/man' '--enable-inline' '--disable-arch-native' '--enable-async-io=9' '--enable-store-to=ufs,aufs,diskd,rock' '--enable-removal-policies=lru,heap' '--enable-delay-pools' '--enable-cache-digests' '--enable-icap-client' '--enable-follow-x-forwarded-for' '--enable-auth-basic=LDAP,NCSA,PLAIN,POP3,RADIUS,SASL,SMB' '--enable-auth-digest=file,LDAP' '--enable-auth-negotiate=kerberos,wrapper' '--enable-auth-ntlm=fake,SMB,LM' '--enable-external-acl-helpers=file,userip,kerberos,ldap_group,LDAP_group,session,SQL_session,time_quota,unix_group,wbinfo_group' '--enable-security-cert=validators=fake' '--enable-storeid-rewrite-helpers=file' '--enable-url-rewrite-helpers=fake' '--enable-aui' '--enable-esi' '--enable-icmp' '--enable-zph-qos' '--enable-ecap' '--enable-translation' '--with-swappdir=/var/spool/squid' '--with-logdir=/var/log/squid' '--with-pidfile=/run/squid.pid' '--with-filedescriptors=65536' '--with-large-files' '--with-default-user=proxy' '--enable-linux-netfilter' '--with-systemd' '--with-gnutls' 'build_alias=aarch64-linux-gnu' 'CFLAGS=-g -O2 -ffile-prefix-map=/build/reproducible-path/squid-5.7= -fstack-protector-strong -Wformat -Werror=format-security -Wno-error=deprecated-declarations' 'LDFLAGS=-Wl,-z,relro -Wl,-z,now' 'CXXFLAGS=-Wdate-time -D_FORTIFY_SOURCE=2' 'CXXFLAGS=-g -O2 -ffile-prefix-map=/build/reproducible-path/squid-5.7= -fstack-protector-strong -Wformat -Werror=format-security -Wno-error=deprecated-declarations'
```

b) Vérifier les journaux de squid pour toute erreur sur les connexions

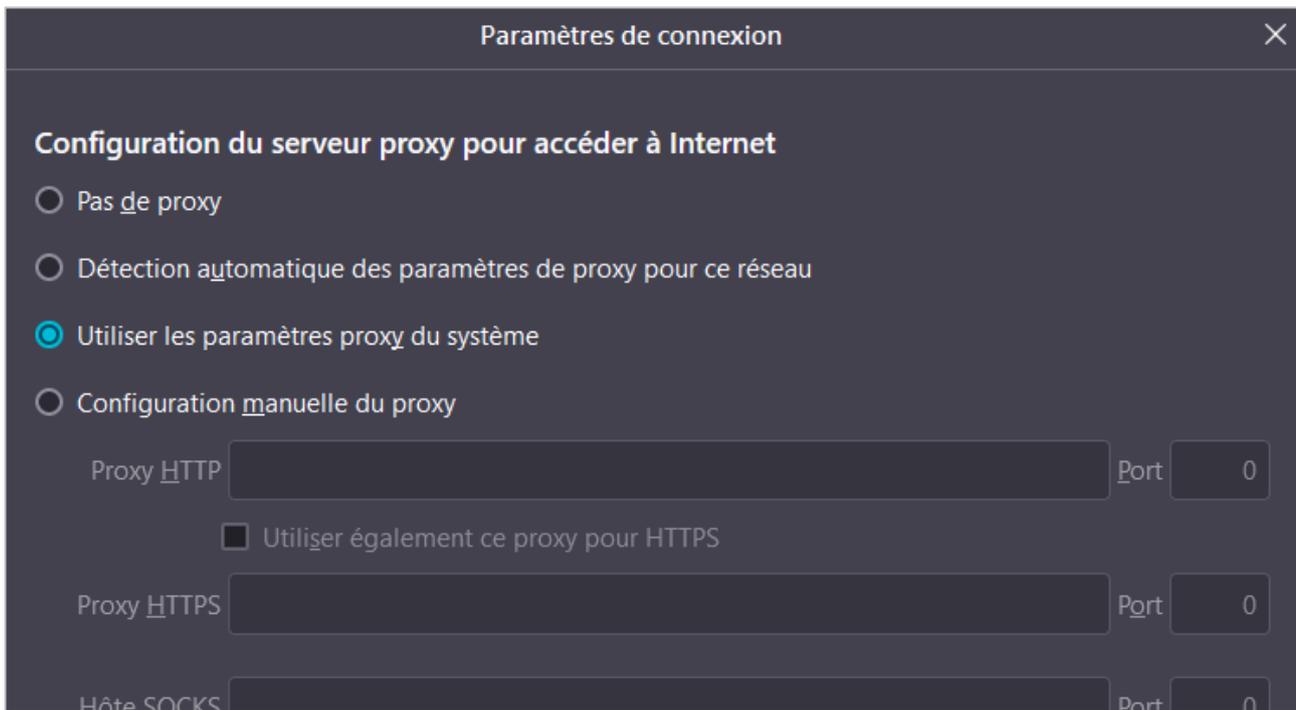
```
# tail -f /var/log/squid/access.log
```

c) Vérifier le fichier squid.conf

```
# squid -k parse
# squid -k reconfigure (recharge la conf sans redémarrer le serveur)
```

## 11. Paramétrage du proxy dans le navigateur

- Ouvrir le navigateur **Mozilla Firefox**
- Sélectionner le menu **Modifier/Préférences**
- En bas de page de la section **Paramètres réseaux**, cliquer sur le bouton **Paramètres...**



- Si le proxy est **configuré dans Windows**, cocher la case **Utiliser les paramètres proxy du système**
- Si le proxy n'est pas configuré dans Windows, cocher la case **Configuration manuelle du proxy** et saisir **l'@IP du serveur Squid** et le **port 3128** dans les 3 champs.

## 12. Liens annexes

Liste de contenu à consulter ou à télécharger pour **Squid**

- Squid Ubuntu : <https://fr.linux-terminal.com>
- Squid Docker : <https://fariszr.com/squid-proxy>
- Documentation : <https://openclassrooms.com/fr/squid>

## 13. Commandes RaspberryPi

- Liste des commandes basique à la gestion du serveur RaspberryPi

```
# shutdown -h now # éteint le serveur en toute sécurité
# shutdown -r now # redémarre le serveur en toute sécurité
# apt install xrdp # installe le bureau à distance RDP
# nano /etc/rc.conf -> service_enable=YES # active le service au démarrage
##### Désactive la mise en veille #####
# systemctl mask sleep.target suspend.target hibernate.target hybrid-sleep.target
```

## 14. Conclusion

**Squid** est installé et configuré avec succès sur le serveur **RaspberryPi Debian 12**. On peut désormais naviguer rapidement sur Internet.

Destiné au RaspberryPi (Raspbian), **Squid** fonctionne aussi parfaitement sur une distribution Ubuntu, CentOS, Windows, Docker...

Pour Linux CentOS : <https://www.linuxtricks.fr/wiki/centos>

Pour Linux RedHat : <https://docs.redhat.com/fr/documentation>