

Créer un serveur de cache

Par LoiselJP
Le 29/09/2023

1 Sommaire

1	Sommaire.....	2
2	Objectif.....	2
3	Avant-propos.....	2
4	Prérequis.....	3
5	Installation du serveur.....	3
	a - Installer les paquets nécessaires.....	5
	b - Placer une adresse IP fixe.....	5
6	Configurer Squid.....	6
	Démarrer et initialiser Squid.....	9
7	Créer le serveur de cache pour les paquets.....	9
	a - Configurer installer le serveur.....	9
8	Pour information.....	10
	a - Configurer le poste client.....	10
	b - Déplacer les fichiers temporaires et fichiers de log.....	11
	c - Un peu de nettoyage.....	11

2 Objectif

Ce document donne la marche à suivre pour la création d'un serveur de cache.

Ce serveur propose un cache pour les différentes connexions internet.

Il pourrait être utilisé comme proxy, mais l'idée est plus de l'utiliser comme serveur de cache pour les différentes mise à jour. Ce n'est pas un serveur de repos, un serveur de repos propose toutes les mises à jour même si elles ne sont pas utilisées, si vous utilisez différentes distributions vous aurez besoin d'autant plus d'espace, Dans le cas d'une distribution simple, on arrive très vite à des centaines de GO de stockage. Multipliez par autant de distributions, Debian, Ubuntu, Centos, AlmaLinux, Rocky Linux etc. et vous voyez que les repos vont prendre des TO.

Ici c'est bien un serveur qui va mettre en cache les différents fichiers téléchargés, si on l'utilise principalement pour Linux, on pourrait aussi l'utiliser pour Windows.

En plus de l'installer comme proxy, on va y configurer une cache spécifique pour les paquets.

3 Avant-propos

Les choix dans les logiciels utilisés sont nombreux, mais comme il faut s'arrêter quelque part, ils se sont arrêtés sur des systèmes ayant fait leur preuves tant sur le plan des performances que sur le plan de la simplicité de mise en œuvre.

Ce serveur n'a d'autre rôle que d'être un serveur de cache à proprement parler, il sera donc installé sans aucune interface graphique...

Ce document vient en complément de la vidéo disponible : [ICI](#)

Ce document est téléchargeable: [ICI](#)

4 Prérequis

Ici, il est question de créer un serveur particulier, quelques connaissances de base sont nécessaires. Ceci permettra par la suite d'éviter de longues explications et se limiter aux différentes commandes et modifications de fichier de configuration.

Il n'est pas question ici de voir ou revoir toutes les bases de Linux. Pour appliquer cette opération, on suppose alors que sont acquis :

- L'installation de base d'un poste de travail linux,
- L'installation de logiciel:
 - Principe des paquets et leurs dépendances,
 - Les "dépôts" de paquets,
 - Principe d'installation manuelle,
- Les principales commandes du shell,
- Utilisation du programme man et l'aide des commandes (si vous ne savez pas appliquer ce qui est demandé).

Pour le besoin de ce tutoriel, la machine utilisée à un processeur avec 8 cœurs, un disque de 1To, mémoire RAM 16Go et la carte réseau utilisée est une carte 10 GigaBit... *Le système est fonctionnel avec 4Gb Ram et 2 coeurs, mais les performances ne seront pas les mêmes !*

5 Installation du serveur

Chacun choisira la distribution de son choix, dans le cas présent toute l'installation est donnée pour une version à base de Debian. Une versions 12,1 de Debian.

Partitionner les disques

Voici la table des partitions et les points de montage actuellement configurés. Vous pouvez choisir une partition et modifier ses caractéristiques (système de fichiers, point de montage, etc.), un espace libre pour créer une nouvelle partition ou un périphérique pour créer sa table des partitions.

Partitionnement assisté

Configurer le RAID avec gestion logicielle

Configurer le gestionnaire de volumes logiques (LVM)

Configurer les volumes chiffrés

Configurer les volumes iSCSI

▼ /dev/nvme0n1 - 1.1 TB VMware Virtual NVMe Disk

>	n° 1	primaire	10.0 GB	f	swap	swap
>	n° 2	primaire	10.0 GB	f	ext4	/
>	n° 3	primaire	1.1 TB	f	ext4	/datas

Annuler les modifications des partitions

Terminer le partitionnement et appliquer les changements

Une version minimale est installée, sans aucun logiciel particulier.

10 Go sont utilisés pour le système, autant pour le swap, le reste est monté dans le dossier « /datas », mais la configuration du système dépend de votre choix (pour rappel vous devriez savoir installer un serveur au minimum jusqu'ici).

Sélection des logiciels

Actuellement, seul le système de base est installé. Pour adapter l'installation à vos besoins, vous pouvez choisir d'installer un ou plusieurs ensembles prédéfinis de logiciels.

Logiciels à installer :

- environnement de bureau Debian
- ... GNOME
- ... Xfce
- ... bureau GNOME Flashback
- ... KDE Plasma
- ... Cinnamon
- ... MATE
- ... LXDE
- ... LXQt
- serveur web
- serveur SSH
- utilitaires usuels du système

Seul le serveur SSH est ajouté, pour faciliter les connexions à distance. Suivant le type de connexion, par root ou par un compte utilisateur, par clef... C'est de la configuration de base !

a - Installer les paquets nécessaires

Sans aucun paquet installé le serveur démarrer en quelques secondes...

Y ajouter alors les paquets nécessaires au besoin du serveur.

```
apt install squid apt-cacher-ng logrotate
```

b - Placer une adresse IP fixe

Il peut être intéressant de placer une adresse IP fixe à notre serveur (c'est parfois indispensable pour le cas où le serveur peut changer d'adresse – vous pouvez aussi fixer l'adresse sur le serveur dhcp en fonction de l'adresse mac),

```
vi /etc/network/interfaces
```

Placer l'IP fixe dans le fichier de configuration

```
# loopback interface
auto lo
```

```
iface lo inet loopback
address 127.0.0.1
netmask 255.0.0.0

# Static IP for eth card (or auto)
allow-hotplug eth0
iface eth0 inet static

# A adapter suivant le réseau
address 192.168.1.99/24
broadcast 192.168.1.255
gateway 192.168.1.1
dns-nameservers 192.168.1.1
auto eth0
```

Puis redémarrer l'interface réseau :

```
/etc/init.d/networking restart
```

Ou

```
service networking restart
```

Placer le hostname au besoin

```
vi /etc/hostname
```

Au choix donner un nom au serveur :

```
Deb-Cache
Debian-data
Partage
...
```

Pour les noms hosts c'est ici :

```
vi /etc/hosts
```

6 Configurer Squid

Dans ce chapitre ne sera indiqué que les commandes et quelques fichiers de configuration.

Avant toute opération, effectuer une sauvegarde du fichier de configuration :

```
cp /etc/squid/squid.conf /etc/squid/squid.conf.default
```

(ou « mv » si vous remplacez le contenu)

Remplacer son contenu par un modèle de configuration plus « classique » :

```
vi /etc/squid/squid.conf
```

Si vous ne voulez pas prendre le fichier suivant, vous pouvez toujours lire les 9000 lignes du fichier par défaut 😊.

```

9145 # opening restrictions: happy_eyeballs_connect_timeout and
9146 # happy_eyeballs_connect_gap. See the former for related termino
9147 #Default:
9148 # no artificial limit on the number of concurrent spare attempts
9149 █
:set number

```

Placer dans le fichier le contenu ci-dessous

On notera un cache démesuré (100gb) et une limite de 1000Mb pour les fichiers téléchargés, le but de Squid étant ici le cache c'est donc indispensable !

```

##### /etc/squid/squid.conf by LJP-Infos #####
http_port 3128
# env. 10Mb of ram per Go of cache! (This is a cache srv!)
cache_mem 4 GB
acl QUERY urlpath_regex cgi-bin \?
cache deny QUERY
acl apache rep_header Server ^Apache
access_log /var/log/squid/access.log squid
hosts_file /etc/hosts
refresh_pattern ^ftp:      1440 20% 10080
refresh_pattern ^gopher:   1440 0% 1440
refresh_pattern .          0 20% 4320
acl all src all
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl to_localhost dst 127.0.0.0/8
acl SSL_ports port 443      # https
acl SSL_ports port 563      # snews
acl SSL_ports port 873      # rsync
acl Safe_ports port 80      # http
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
acl Safe_ports port 631     # cups
acl Safe_ports port 873     # rsync
acl Safe_ports port 901     # SWAT
acl purge method PURGE
acl CONNECT method CONNECT
acl LAN_SQUID src 192.168.0.0/16
#
# acls for windows update (or all other site for downloads)
# #####
acl bwlimit_domains dstdomain "/etc/squid/bwlimit_domains.acl"
delay_pools 1

```

```
delay_class 1 1
delay_parameters 1 100000/100000
delay_access 1 allow bwlimit_domains
#
http_access allow manager localhost
http_access deny manager
http_access allow purge localhost
http_access deny purge
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost
http_access allow LAN_SQUID
http_access deny all
http_reply_access allow all
icp_access allow all
cache_effective_group proxy
#
# This is a cache srv ( !\ Necessary ram !\ )
coredump_dir /var/spool/squid 10000 16 256
cache_dir ufs /var/spool/squid 1000000 256 256
#
# Allow size objects (cache that you want!)
maximum_object_size 1024 MB
minimum_object_size 0
#
# IF srv is used only to accelerate
redirect_rewrites_host_header off
#
cache_mgr loiseljp@gmail.com
visible_hostname Deb-Cache
##### END /etc/squid/squid.conf #####
```

(Attention aux copier/coller dans un pdf... on copie parfois le pied page et là ça marche moins bien !)

Comme squid.conf contient une acl pour Windows update il faut créer le fichier correspondant aux sites :

```
vi /etc/squid/bwlimit_domains.acl
```

Y ajouter les sites de téléchargement d'update...

```
windowsupdate.microsoft.com
.update.microsoft.com
.download.windowsupdate.com
redir.metaservices.microsoft.com
images.metaservices.microsoft.com
c.microsoft.com
wustat.windows.com
crl.microsoft.com
productactivation.one.microsoft.com
ntservicepack.microsoft.com
sls.microsoft.com
```


Démarrer et initialiser Squid

Avant toute utilisation il sera nécessaire de définir le dossier de cache.

Si vous avez créé une partition dédiée aux datas (comme indiqué au début /datas), on choisit de rediriger le dossier à l'emplacement prévu à cet effet (dossier /datas/cache_squid) :

```
rm -rf /var/spool/squid
mkdir /datas/cache_squid
ln -s /datas/cache_squid /var/spool/squid
```

On pourrait le mettre directement dans le fichier de conf, mais ça sera plus pratique comme ça...

Initialiser squid...

```
systemctl stop squid
chmod 777 -R /var/spool/squid/
/usr/sbin/squid -z
chown -R root:root /var/spool/squid
```

Suivant la config choisie, le nombre de dossiers et sous dossiers définie par les variables `coredump_dir` et `cache_dir` ufs la commande pourra prendre quelques minutes...

Il faudra aussi appuyer sur « Enter » pour en sortir....

On redémarre squid et on vérifie...

```
systemctl start squid
systemctl status squid
```

En cas d'erreur, il arrive parfois (souvent) qu'il soit nécessaire de refaire un :

```
chmod 777 -R /var/spool/squid/
# Puis
systemctl start squid
```

7 Créer le serveur de cache pour les paquets

Bien que Squid fasse très bien son travail de cache pour les paquets deb et rpm il existe un outil encore plus efficace qui gèrera les accès aux dépôts.

Nettement plus intéressant que de créer un serveur de dépôt, « apt-cacher-ng » permet de ne stocker que les paquets nécessaires (ceux qui ont déjà été téléchargé).

« apt-cacher-ng » utilisera le port qui lui est configuré mais par défaut c'est le port 3142 qui sera utilisé.

a - Configurer installer le serveur

Le serveur s'installe avec une simple commande apt-get install (ou juste apt, si ce n'est pas encore fait...):

```
apt install apt-cacher-ng
```

Configurer le serveur suivant les besoins...

```
vi /etc/apt-cacher-ng/acng.conf
```

Les premières versions de apt-cacher-ng ne prenaient en compte que les paquets DEB, aujourd'hui la plupart des distributions sont prises en compte, mais c'est un choix de configuration.

Il est déjà très largement configuré pour le sujet. On va juste modifier la variable CacheDir en :

```
CacheDir: /datas/apt_cache
```

On remarquera qu'il peut lui aussi passer derrière un proxy (un cache derrière un cache...).

Puis on va copier les datas de /var/cache/apt-cacher-ng au bon endroit

```
cp -rp /var/cache/apt-cacher-ng /datas/apt_cache
```

(on préserve /var/cache/apt-cacher-ng pour pouvoir au besoin resetter le cache!)

Redémarrer le service pour prendre en compte la ou les modifications :

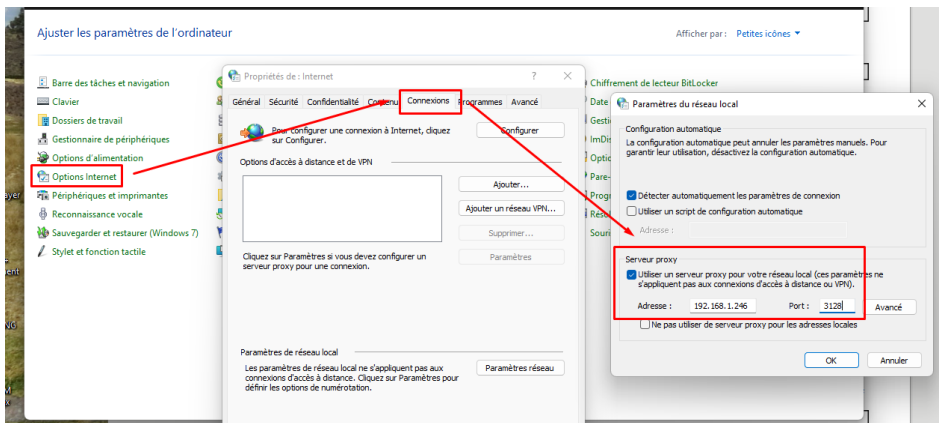
```
systemctl restart apt-cacher-ng
```

8 Pour information

a - Configurer le poste client

Ce n'est pas tout d'avoir un serveur de cache, mais encore faut-il pouvoir configurer le système pour.

Coté Windows c'est assez simple, dans la configuration d'internet, il suffit de lui indiquer d'utiliser le proxy...



Pour Linux, avec interface graphique on va dans le réseau, et on configure le proxy avec l'adresse IP, le port 3128.

Sinon, en ligne de commande, on set la/les variable http_proxy pareil avec l'adresse du serveur et le port :

```
export http_proxy="http://192.168.1.246:3128/"
```

Suivant ce qui aura été configuré et la volonté de cache...

```
export http_proxy="http://192.168.1.246:3128/"
export https_proxy="http://192.168.1.246:3128/"
export ftp_proxy="ftp://192.168.1.246:3128/"
```

En, fait on lui dit de rediriger les appels vers le port du proxy pour les différents type d'appels.

Pour le cache apt, hé bien on ajoute un fichier « apt.conf » dans « /etc/apt » dans lequel on ajoute le proxy avec le port 3128 ou 3142 si vous avez installé apt-cacher-ng :

```
echo 'Acquire::http::proxy "http://192.168.1.246:3142/";' >> /etc/apt/apt.conf
```

Suivant les téléchargement (dépôts des paquets) on pourra avoir différents valeurs :

```
echo 'Acquire::http::proxy "http://192.168.1.246:3142/";' >> /etc/apt/apt.conf
echo 'Acquire::https::proxy "http://192.168.1.246:3142/";' >> /etc/apt/apt.conf
echo 'Acquire::ftp::proxy "ftp://192.168.1.246:3142/";' >> /etc/apt/apt.conf
```

(Attention aussi au coier/coller ' et ' sont parfois différents!)

b - Déplacer les fichiers temporaires et fichiers de log

Ce type de serveur traite souvent des milliers de fichiers. Certains programmes utilisent les dossiers temporaires, ce dossier peut parfois atteindre également des tailles démesurées.

Il sera parfois utile de le déplacer (on pourra alors le déplacer dans la seconde partition ou disque monté dans /datas) :

```
rm -rf /tmp
mkdir /datas/tmp
ln -s /datas/tmp /tmp

rm -rf /var/tmp
mkdir /datas/tmp_var
ln -s /datas/tmp_var /var/tmp
```

c - Un peu de nettoyage

Après quelques temps le cache devrait s'équilibrer aux alentours de 260 à 300 Go bien sur suivant la configuration, il est parfois intéressant de remettre un peu d'ordre avec un redémarrage de squid + nettoyage du dossier de squid:

```
rm -rf /var/cache/apt/archives/*
mkdir /var/cache/apt/archives/partial
rm -rf /tmp/*
rm -rf /var/tmp/*
rm -rf /var/log/*.0
rm -rf /var/log/*.1
rm -rf /var/log/*.2
rm -rf /var/log/*.3
rm -rf /var/log/*.4
```

```
rm -rf /var/log/*.gz
rm -rf /var/log/*/*0
rm -rf /var/log/*/*1
rm -rf /var/log/*/*2
rm -rf /var/log/*/*3
rm -rf /var/log/*/*4
rm -rf /var/log/*/*gz

# Supprime le cache apt
systemctl stop apt-cacher-ng
rm -rf /datas/cache_apt
cp -rp /var/cache/apt-cacher-ng /datas/apt_cache
systemctl start apt-cacher-ng

# Nettoyer le cache Squid
/etc/init.d/squid stop
chmod 777 /var/spool/squid/
rm -rf /var/spool/squid/*
/usr/sbin/squid -z
/etc/init.d/squid restart

# Pour effacer les traces des données
dd if=/dev/zero of=/datas/zeros bs=4096
sync
rm -f /datas/zeros
```