

Installation et configuration d'un proxy squid

Table of Contents

- [Installation et configuration d'un proxy squid](#)
 - [Installation de squid](#)
 - [Construction à partir de la source](#)
 - [Configuration de squid](#)
 - [Exécution et test de squid](#)
 - [Exécution de squid](#)
 - [Test de squid](#)
 - [Utilisation de Squid pour accélérer les téléchargements de packages](#)

Le proxy de mise en cache Squid est un projet open source établi de longue date. Outre les fonctionnalités de proxy et de cache de base, Squid est également idéal pour gérer, filtrer et analyser les accès HTTP et HTTPS. Un exemple de ceci est l'utilisation d'un filtre de contenu pour réécrire ou rediriger les URL, et une application typique pour cela est le blocage des sites de suivi et du contenu répréhensible.

Installation de squid

Cette section présente la construction de Squid à partir de la source (Git) et montre quelques exemples de base sur la façon de l'utiliser.

Construction à partir de la source

Les sources de Squid peuvent être téléchargées sur <http://www.squid-cache.org/>.

```
wget http://www.squid-cache.org/Versions/v4/squid-4.10.tar.xz -O
${LFS}/download/squid-4.10.tar.xz --no-check-certificate
rm -Rf squid-4.10
cd squid-4.10
./bootstrap.sh
automake (1.16.1) : automake
autoconf (2.69) : autoconf
libtool (2.4.6) : libtool
libtool path : /usr/bin
Bootstrapping
parallel-tests: installing 'cfgaux/test-driver'
Fixing configure recursion
Autotool bootstrapping complete.
```

```
mkdir build; cd build
```

Il est maintenant temps de configurer le build (il existe de nombreuses options de configuration disponibles que l'on peut explorer en utilisant `configure --help`).

```
../configure --prefix=/opt/squid --with-default-user=squid --enable-ssl --
disable-inlined \
--disable-optimizations --enable-arp-acl --disable-wccp --disable-wccp2 --
disable-htcp \
--enable-delay-pools --enable-linux-netfilter --disable-translation --
disable-auto-locale \
--with-logdir=/opt/squid/log/squid --with-pidfile=/opt/squid/run/squid.pid

...
configure: creating ./config.status
config.status: creating Makefile
config.status: creating config.h
config.status: executing depfiles commands
config.status: executing libtool commands
```

Construire le code et installer les binaires dans `/opt/squid` comme spécifié ci-dessus lors de la configuration

```
make
make install
```



Le script pour construire squid dans la toolchain LFS est disponible [ici](#)

Configuration de squid

Modifier la configuration par défaut en éditant `etc/squid.conf`. La configuration par défaut de `squid.conf` fournie peut être suffisante pour les tests et l'utilisation de base, mais quelques éléments doivent être modifiés dans `/etc/squid/squid.conf` :

- **Taille maximale de l'objet.** Normalement, c'est 4MB. Cependant, certains des packages de base tels que glibc peuvent dépasser 16 Mo : `maximum_object_size 409600 KB`
- **Politique de remplacement du cache.** Normalement, la politique par défaut `lru` va bien. Mais dans certains cas, pour conserver les objets les plus volumineux plus longtemps et ne télécharger que les objets plus petits on peut utiliser : `cache_replacement_policy heap LFUDA`
- **Taille du cache.** La taille normale est de 100 Mo. Cela doit être d'au moins 1 Go ou plus : `cache_dir ufs /var/spool/squid 1000 16 256`
- **Périphériques/sous-réseau dont l'accès est autorisé:** `acl localnet src 192.168.3.0/24`
`http_access allow localnet`

- **Définir le port d'écoute:** Par défaut, il est défini sur le port 3128. C'est le port que Squid écoutera pour les requêtes. C'est ce port qu'il faut définir dans le navigateur des clients :
`http_port 3128`

Créer maintenant un utilisateur squid et lui donner les autorisations d'écrire dans divers dossiers sous `/opt/squid`:

```
sudo adduser squid

mkdir -p /opt/squid/log
sudo chown -R squid:squid /opt/squid/log
sudo chown -R squid:squid /opt/squid/var
sudo chown -R squid:squid /opt/squid/run
```



Fichiers et dossiers importants:

fichier/dossier	Objet
etc/squid.conf	Fichier de configuration Squid
log/squid/access.log, log/squid/cache.log	Fichiers journaux Squid

Exécution et test de squid

Exécution de squid

Lancer squid avec l'utilisateur squid:

```
$ su squid
$ /opt/squid/sbin/squid
$ ps -e | grep quid
10486 ?          00:00:00 squid
10501 ?          00:00:00 squid

$ more /opt/squid/log/squid/cache.log
2021/06/30 10:42:24 kid1| Set Current Directory to
/opt/squid/var/cache/squid
2021/06/30 10:42:24 kid1| Starting Squid Cache version 5.0.6-VCS for x86_64-
pc-linux-gnu...
2021/06/30 10:42:24 kid1| Service Name: squid
2021/06/30 10:42:24 kid1| Process ID 98677
2021/06/30 10:42:24 kid1| Process Roles: worker
2021/06/30 10:42:24 kid1| With 1024 file descriptors available
2021/06/30 10:42:24 kid1| Initializing IP Cache...
2021/06/30 10:42:24 kid1| DNS Socket created at [::], FD 7
2021/06/30 10:42:24 kid1| DNS Socket created at 0.0.0.0, FD 8
2021/06/30 10:42:24 kid1| Adding nameserver XX.XX.XX.XX from
```

```
/etc/resolv.conf
2021/06/30 10:42:24 kid1| Logfile: opening log
daemon:/opt/squid/log/squid/access.log
2021/06/30 10:42:24 kid1| Logfile Daemon: opening log
/opt/squid/log/squid/access.log
2021/06/30 10:42:24 kid1| WARNING: no_suid: setuid(0): (1) Operation not
permitted
2021/06/30 10:42:24 kid1| Store logging disabled
2021/06/30 10:42:24 kid1| Swap maxSize 0 + 262144 KB, estimated 20164
objects
2021/06/30 10:42:24 kid1| Target number of buckets: 1008
2021/06/30 10:42:24 kid1| Using 8192 Store buckets
2021/06/30 10:42:24 kid1| Max Mem size: 262144 KB
2021/06/30 10:42:24 kid1| Max Swap size: 0 KB
2021/06/30 10:42:24 kid1| Using Least Load store dir selection
2021/06/30 10:42:24 kid1| Set Current Directory to
/opt/squid/var/cache/squid
2021/06/30 10:42:24 kid1| Finished loading MIME types and icons.
2021/06/30 10:42:24 kid1| Squid plugin modules loaded: 0
2021/06/30 10:42:24 kid1| Adaptation support is off.
2021/06/30 10:42:24 kid1| Accepting HTTP Socket connections at conn3
local=[::]:3128 remote=[::] FD 11 flags=9
2021/06/30 10:42:25 kid1| storeLateRelease: released 0 objects
```



Lorsqu'on apporte des modifications à squid.conf, il faut exécuter une reconfiguration :

```
$ /opt/squid/sbin/squid -k reconfigure
```

Test de squid

Se connecter à un hôte Linux, configurer l'utilisation d'un proxy sur la ligne de commande et effectuer un get. (Dans l'exemple ci-dessous, le proxy Squid s'exécute sur 192.168.3.75, et on le teste à partir de 192.168.3.44.)

```
$ export http_proxy=192.168.3.75:3128

$ $ wget http://example.com
--2021-06-30 10:50:17-- http://example.com/
Connecting to 192.168.3.75:3128... connected.
Proxy request sent, awaiting response... 200 OK
Length: 1256 (1.2K) [text/html]
Saving to: 'index.html'
...
```





On peut afficher l'activité de Squid en consultant access.log. Si configuré pour le format natif, chaque ligne sera écrite comme indiqué ci-dessous.

```
1625064617.247 46 192.168.3.20 TCP_MISS/200 1673 GET
http://example.com/ - HIER_DIRECT/93.184.216.34 text/html
```

Utilisation de Squid pour accélérer les téléchargements de packages

Le cache racine contient les packages standard buildroot, mais à mesure qu'il vieillit, de yum update en yum update, plusieurs packages peuvent périmés. De plus, certains paquets ne font pas partie de la buildroot standard, mais sont souvent nécessaires (par exemple, pkgconfig). Ces packages peuvent être servis plus rapidement à partir d'un cache squid que de les télécharger à partir d'Internet pour chaque version de paquet.

Pour que l'utilisateur de la version utilise automatiquement le cache squid, taper:

```
echo export http_proxy=\" http://localhost:3128 \" >>
/home/build/.bash_profile
```

Pour n'utiliser squid pour certaines configurations fictives, on peut, au lieu de définir la variable d'environnement http_proxy , définir la configuration du proxy dans chaque fichier de configuration de yum.conf:

```
config_opts['yum.conf'] = ""
[main]
...
proxy=http://localhost:3128/
...
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=centos:squid>

Last update: **2025/02/19 10:59**

