

# Installing **squid** as a sibling to an already existing Parent **squid**.

Squid is a caching and forwarding web proxy. It has a wide variety of uses, from speeding up a web server by caching repeated requests; to caching web, DNS and other computer network lookups for a group of people sharing network resources; to aiding security by filtering traffic.

1. We are trying to create a sibling **squid** which connects to a Parent **squid**.
2. Parent Squid only allows with **username/password**.

Assumption : We already have a **parent squid** which uses authentication.

The reason why we do this is to make a single squid authenticate on behalf of all the nodes in the network.

In the current scenario, all the node which are in 172.22.x.x series which need to access the internet proxy, would have to add **username/password** to their proxy setting, which is not secure and very difficult to maintain.

So what we do is to create a sibling **squid** which does the authentication for us.

```
[172.22.x.x] | |
+--> [sibling squid] <=====(authenticate)=====> [parent squid] | +-----> (( internet ))
```

Command below install **squid**.

```
[root@server-cloudera-manager yum.repos.d]# yum install squid
Loaded plugins: product-id, rhnplugin, security, subscription-manager
There was an error communicating with RHN.
RHN Satellite or RHN Classic support will be disabled.
Error communicating with server. The message was:
Temporary failure in name resolution
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package squid.x86_64 7:3.1.23-9.el6 will be installed
--> Processing Dependency: libltdl.so.7()(64bit) for package: 7:squid-3.1.2
--> Running transaction check
---> Package libtool-ltdl.x86_64 0:2.2.6-15.5.el6 will be installed
--> Finished Dependency Resolution
```

Dependencies Resolved

```
=====
Package                Arch             Version          Repository
=====
Installing:
 squid                 x86_64           7:3.1.23-9.el6  rhel-6-serv
Installing for dependencies:
 libtool-ltdl          x86_64           2.2.6-15.5.el6  rhel-6-serv
=====
```

## Transaction Summary

=====  
Install 2 Package(s)

Total download size: 1.9 M

Installed size: 6.4 M

Is this ok [y/N]: y

Downloading Packages:

(1/2): libtool-ltdl-2.2.6-15.5.el6.x86\_64.rpm

(2/2): squid-3.1.23-9.el6.x86\_64.rpm

-----  
Total

233 kB/s

Running rpm\_check\_debug

Running Transaction Test

Transaction Test Succeeded

Running Transaction

Installing : libtool-ltdl-2.2.6-15.5.el6.x86\_64

Installing : 7:squid-3.1.23-9.el6.x86\_64

rhel-6-server-rpms/productid

rhel-server-dts-6-rpms/productid

Verifying : 7:squid-3.1.23-9.el6.x86\_64

Verifying : libtool-ltdl-2.2.6-15.5.el6.x86\_64

Installed:

squid.x86\_64 7:3.1.23-9.el6

Dependency Installed:

libtool-ltdl.x86\_64 0:2.2.6-15.5.el6

Complete!

Installation in complete. Now setting chkconfig for squid.

```
[root@server-cloudera-manager yum.repos.d]# chkconfig squid on
```

## Configuration of Squid.

We are configuring UK squid as a sibling to the parent squid running in Atlanta. This is how its going to work.

- Step 1. Configure squid with cache\_peer on cloudera-manager.
- Step 2. Configure all the datanodes to connect to our proxy.

Working. And more on this here : <http://wiki.squid-cache.org/Features/CacheHierarchy>

More on cache\_peer : [http://www.squid-cache.org/Doc/config/cache\\_peer/](http://www.squid-cache.org/Doc/config/cache_peer/)

1. 172.22.x.x will connect to sibling squid on port 18716
2. Sibling squid will by-pass the requests to parent squid on port 18717 using login authentication.

## squid Configuration.

Configuration currently on squid. Assuming we are allowing 172.22.0.0/16 network.

Important lines to look for.

Allowing network.

```
# Allowing `172.22.0.0/16` network.
acl localnet src 172.22.0.0/16 # RFC1918 possible internal network
```

Setting port - sibling squid will be listening on this port.

```
# Squid normally listens to port 3128
http_port 18716
```

Important cache\_peer to connect to parent. Here parent\_ip\_addr will be the parent IP.

```
# setting `cache_peer`
cache_peer parent_ip_addr parent 18717 0 proxy-only login=username:password default
```

Making sure all the request go through the parent

```
# we are making sure that all the traffic goes through the parent.
never_direct allow all
```

**Here is the complete configuration.**

```
#
# Recommended minimum configuration:
#
acl manager proto cache_object
acl localhost src 127.0.0.1/32 ::1
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32 ::1

# Example rule allowing access from your local networks.
# Adapt to list your (internal) IP networks from where browsing
# should be allowed

# Allowing `172.22.0.0/16` network.
acl localnet src 172.22.0.0/16 # RFC1918 possible internal network

acl localnet src fc00::/7 # RFC 4193 local private network range
acl localnet src fe80::/10 # RFC 4291 link-local (directly plugged) machines

acl SSL_ports port 443
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 CONNECT method CONNECT

#
# Recommended minimum Access Permission configuration:
#
```

```

# Only allow cachemgr access from localhost
http_access allow manager localhost
http_access deny manager

# Deny requests to certain unsafe ports
http_access deny !Safe_ports

# Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports

# We strongly recommend the following be uncommented to protect innocent
# web applications running on the proxy server who think the only
# one who can access services on "localhost" is a local user
#http_access deny to_localhost

#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#

# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
http_access allow localnet
http_access allow localhost

# And finally deny all other access to this proxy
http_access deny all

# Squid normally listens to port 3128
http_port 18716

# Uncomment and adjust the following to add a disk cache directory.
#cache_dir ufs /var/spool/squid 100 16 256

# Leave coredumps in the first cache dir
coredump_dir /var/spool/squid

# setting `cache_peer`
cache_peer parent_ip_addr parent 18717 0 proxy-only login=username:password default

# we are making sure that all the traffic goes through the parent.
never_direct allow all

# Add any of your own refresh_pattern entries above these.
refresh_pattern ^ftp:          1440    20%    10080
refresh_pattern ^gopher:      1440    0%     1440
refresh_pattern -i (/cgi-bin/|\?) 0     0%     0
refresh_pattern .              0      20%    4320

```

## Restart squid

```
[root@server ~]# service squid restart
```

### **Configuration of clients to connect.**

Add the below lines to the `env` of the node which needs internet connection.

```
export http_proxy=http://sibling_squid_ip_addr:18716
export https_proxy=$http_proxy
export ftp_proxy=$http_proxy
```

### **Configuration on `yum.conf` on all HOSTS.**

Add the below line in the end of `yum.conf` file.

```
proxy=http://sibling_squid_ip_addr:18716
```