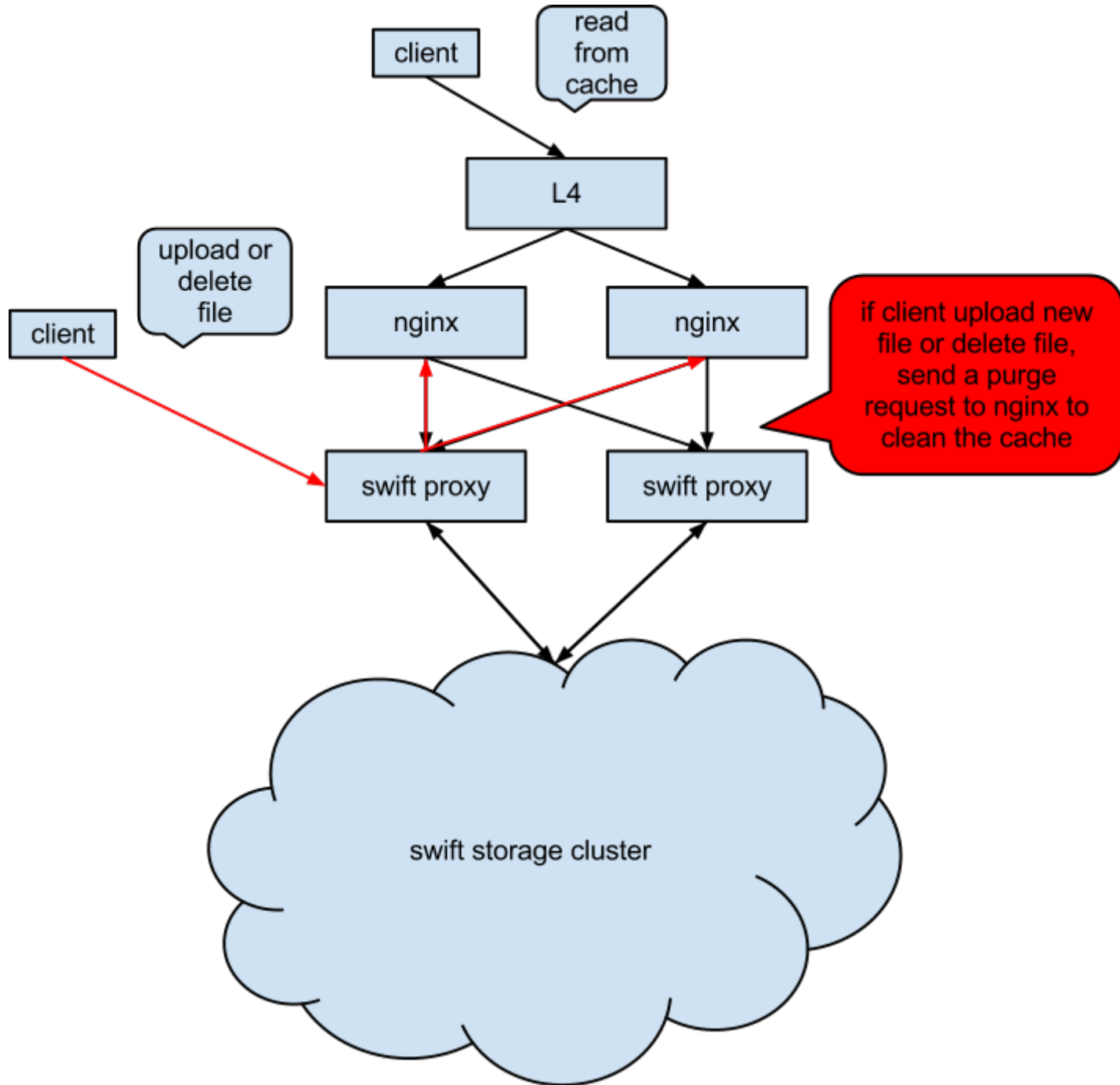


how to use nginx to build a swift cache proxy server and static web server cluster

1. architecture (you must already have a swift filter module first and make sure the container is public):



2. nginx config:

```

upstream backend {
    #this is one of the swift proxy server
    server 10.38.10.127:8080;
}

proxy_temp_path /usr/local/nginx/proxy_temp_dir;
proxy_cache_path /usr/local/nginx/proxy_cache_dir/cache levels=1:2
keys_zone=staticcache:180m inactive=1d max_size=500m;

server {

    #skip some normal config like listen, log
    .....

    #set proxy cache config, you can find all these command in nginx wiki page
    proxy_set_header Host $host;
    proxy_set_header Accept "*/*";
    proxy_cache_valid 200 1d;
    proxy_read_timeout 120;
    proxy_send_timeout 120;
    proxy_cache_key "$host$uri";
    proxy_cache_methods GET;

    #set a return http header of cache status, so that you can see the status of file cache
    add_header X-Cache $upstream_cache_status;

    location / {
        root html;
        index index.html index.htm;

        if ($host ~* (.*)\.(.*)\.storage\.brihost\.com) {
            set $account $1;
            set $container $2;
            set $index_uri $request_uri;
        }

        #if request like: account.container.storage.brihost.com it will get home page
        #this feature is for static web service with swift
        if ($request_uri = '/')
        {
            set $index_uri '/index.html';
        }

        if ($request_uri = "")
        {
            set $index_uri '/index.html';
        }

        proxy_pass http://backend/v1/AUTH_$account/$container$index_uri;
    }
}

```

```
proxy_cache staticcache;

}
```

3. static service access:

<http://testa27.mycontainer2.storage.brighthost.com>

you will see the index.html file display if you already upload it to swift.

4. nginx purge cache (make sure you already build nginx with ngx\_purge\_cache module):

```
location ~ /purge/(.*) {
    allow 127.0.0.1;
    deny all;
    proxy_cache_purge staticcache $1;
}
```

curl "http://127.0.0.1:8088/purge/testa27.mycontainer2.storage.brighthost.com/index.html"

5. logs:

1) create a cache container in swift log account.

2) split log file hourly and use slogging uploader to upload log file to swift log account cache container.

3) write a swift slogging plugin to analysis the log file.

6. risk:

1) can not delete cache file when delete a container or account, just wait cache expired.

2) if you want to delete cache file when file update, you must also send a purge request to cache server , that maybe a heavy workload when you have to much upload file request.

3) if you have a lot cache server, you must use a async queue to send purge request.

7. ref:

<http://wiki.openstack.org/development/swift/slogging>

<http://wiki.openstack.org/development/swift/filter>

