**Nginx反向代理：**

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

.. ..

http {

.. ..

upstream webserver {

server 192.168.2.100 weight=1 max\_fails=1 fail\_timeout=30;

server 192.168.2.200 weight=2 max\_fails=2 fail\_timeout=30;

server 192.168.2.101 down;

}

#weight设置服务器权重值，默认值为1

#max\_fails设置最大失败次数

#fail\_timeout设置失败超时时间，单位为秒

#down标记服务器已关机，不参与集群调度

.. ..

server {

listen 80;

server\_name localhost;

location / {

proxy\_pass http://webserver;

}

}

**设置相同客户端访问相同Web服务器：**

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

.. ..

http {

.. ..

upstream webserver {

#通过ip\_hash设置调度规则为：相同客户端访问相同服务器

ip\_hash;

server 192.168.2.100 weight=1 max\_fails=2 fail\_timeout=10;

server 192.168.2.200 weight=2 max\_fails=2 fail\_timeout=10;

}

.. ..

server {

listen 80;

server\_name www.tarena.com;

location / {

proxy\_pass http://webserver;

}

}

**部署支持4层TCP/UDP代理的Nginx服务器：**

[root@proxy nginx-1.12.2]# ./configure \

> --with-http\_ssl\_module //开启SSL加密功能

> --with-stream //开启4层反向代理功能

stream {

upstream backend {

server 192.168.2.100:22; //后端SSH服务器的IP和端口

server 192.168.2.200:22;

}

server {

listen 12345; //Nginx监听的端口

proxy\_connect\_timeout 1s;

proxy\_timeout 3s;

proxy\_pass backend;

}

}

http {

.. ..

}

**Nginx常见问题处理：**

如果客户端访问服务器提示“Too many open files”如何解决

一、修改Nginx配置文件，增加并发量

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

.. ..

worker\_processes 2; //与CPU核心数量一致

events {

worker\_connections 65535; //每个worker最大并发连接数

use epoll;

}

.. ..

[root@proxy ~]# nginx -s reload

二、优化Linux内核参数（最大文件数量）

[root@proxy ~]# ulimit -a //查看所有属性值

[root@proxy ~]# ulimit -Hn 100000 //设置硬限制（临时规则）

[root@proxy ~]# ulimit -Sn 100000 //设置软限制（临时规则）

[root@proxy ~]# vim /etc/security/limits.conf

.. ..

\* soft nofile 100000

\* hard nofile 100000

#该配置文件分4列，分别如下：

#用户或组 硬限制或软限制 需要限制的项目 限制的值

如何解决客户端访问头部信息过长的问题

修改Nginx配置文件，增加数据包头部缓存大小

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

.. ..

http {

client\_header\_buffer\_size 1k; //默认请求包头信息的缓存

large\_client\_header\_buffers 4 4k; //大请求包头部信息的缓存个数与容量

.. ..

}

[root@proxy ~]# nginx -s reload

如何让客户端浏览器缓存数据

修改Nginx配置文件，定义对静态页面的缓存时间

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

server {

...

}

location ~\* \.(jpg|jpeg|gif|png|css|js|ico|xml)$ {

expires 30d; //定义客户端缓存时间为30天

}

如何自定义返回给客户端的404错误页面

修改Nginx配置文件，自定义报错页面

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

.. ..

error\_page 404 /40x.html; //自定义错误页面

.. ..

[root@proxy ~]# vim /usr/local/nginx/html/40x.html //生成错误页面

Oops,No NO no page …

如何查看服务器状态信息

编译安装时使用--with-http\_stub\_status\_module开启状态页面模块

root@proxy nginx-1.12.2]# ./configure \

> --with-http\_ssl\_module //开启SSL加密功能

> --with-stream //开启TCP/UDP代理模块

> --with-http\_stub\_status\_module //开启status状态页面

cat /usr/local/nginx/conf/nginx.conf

location /status {

stub\_status on;

}

curl http://192.168.4.5/status

Active connections：当前活动的连接数量。

Accepts：已经接受客户端的连接总数量。

Handled：已经处理客户端的连接总数量（一般与accepts一致，除非服务器限制了连接数量）。

Requests：客户端发送的请求数量。

Reading：当前服务器正在读取客户端请求头的数量。

Writing：当前服务器正在写响应信息的数量。

Waiting：当前多少客户端在等待服务器的响应。

开启gzip压缩功能，提高数据传输效率

[root@proxy ~]# cat /usr/local/nginx/conf/nginx.conf

http {

.. ..

gzip on; //开启压缩

gzip\_min\_length 1000; //小文件不压缩

gzip\_comp\_level 4; //压缩比率

gzip\_types text/plain text/css application/json application/x-javascript text/xml application/xml application/xml+rss text/javascript;

//对特定文件压缩，类型参考mime.types

.. ..

}

开启文件缓存功能

如果需要处理大量静态文件，可以将文件缓存在内存，下次访问会更快。

http {

open\_file\_cache max=2000 inactive=20s;

open\_file\_cache\_valid 60s;

open\_file\_cache\_min\_uses 5;

open\_file\_cache\_errors off;

//设置服务器最大缓存2000个文件句柄，关闭20秒内无请求的文件句柄

//文件句柄的有效时间是60秒，60秒后过期

//只有访问次数超过5次会被缓存

}