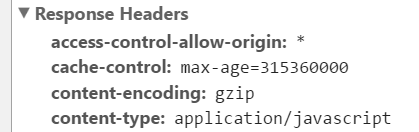
# 1.压缩优化



我能接受的压缩算法 gzip



节省带宽和响应速度

文档:http://nginx.org/en/docs/http/ngx\_http\_gzip\_module.html

# 2.

# gzip on;

#gzip\_buffers 32 4k;//压缩块

#gzip\_com\_level 6;//压缩等级，越高越小号cpu

#gzip\_min\_length 200;//多少长度压缩

#gzip\_types text/css text/javascript

##demo

gzip on;

gzip\_buffers 32 4k;

gzip\_comp\_level 6;

gzip\_min\_length 4000;#4k

gzip\_types text/css text/xml application/x-javascript;

太小就不用压缩

# nginx缓存配置优化，提高网站性能

用到nginx 的expires的设置

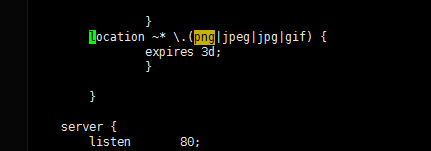
在location或者if段来写，

格式 expires 30s; 30秒过期

expires 30m; 30分钟过期

expires 30h;

expires 30d;



location ~\* \.(png|jpeg|jpg|gif|bmp) {

expires 3d;

}

}

Nginx 反向代理服务器+负载均衡

支持的用法 1个proxy 1个upstream分别用来做反向代理和负载均衡（动静分离）

反向代理

Location ~ \.php$ {

Proxy\_pass http://127.0.0.1:8080

}

负载均衡：

把多态服务器使用up\_stream 绑定在一起起一个组名儿

http://nginx.org/en/docs/stream/ngx\_stream\_upstream\_module.html