Nginx Config Cheat Sheet

by hulj (hulj) via cheatography.com/74222/cs/18778/

CLI

nginx -c	有需要时指定nginx配置文件位置
nginx -s quit	**优雅**关闭nginx进程
nginx -s stop	立即结束nginx进程
nginx -s reload	**优雅**重启
nginx -s reopen	向新文件写入日志
nginx -t	检查配置文件是否有语法错误
nginx -T	检查是否有语法错误并输出配置信息
nginx -V	输出版本信息和已加载的module

Serve Static Content

```
server {
    listen 80;
    ser ver _name www.ex amp le.com;
    loc ation / {
       # 文件根目录
       root /usr/s har e/n gin x/html;
        index index.html index.htm;
   }
}
```

nginx进程需要有目标文件的访问权限

Http Load Balance

```
upstream backend {
    server 10.10.1 23.45 weight=1;
    server exampl e.com weight=2;
}
server {
    loc ation / {
        pro xy_pass http:/ /ba ckend;
   }
}
```

TCP Load Balance

```
stream {
    ups tream mysql_read {
        server r1.com :3306 weight=5;
        server r2.com :3306;
        server r3.com :3306 backup;
   }
    server {
        listen 3306;
        pro xy_pass mysql_ read;
   }
}
```

Limit Bandwidth

```
location /download/ {
    lim it_ rat e_after 10m;
    lim it_rate 1m;
}
```

下载大小超过10m后, 下载速率限制为1m/s

Access Based on IP Address

```
location /admin/ {
   deny 10.0.0.1;
    allow 10.0.0.0/20;
   deny all;
   ...
}
```

allow 和 deny必须同时指定

HTTPS Redirects

```
server {
    listen 80 defaul t_s erver;
    ser ver _name a.com;
    return 301 https: //$ hos t$r equ est _uri;
}
```

重定向a.com的所有http请求到https



Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

HTTP Basic Authentication

```
# 生成密码
openssl passwd ${password}
# 创建一个存放用户名密码的文件
name1:password1:comment1
name2:password2:comment2
# nginx配置
location / {
auth_basic "Private site";
auth_basic_user_file conf.d/passwd;
}
```

- 使用openssl passwd 生成密码，并保存到指定文件
- 文件内容如：name:${加密后的密码}:注释内容
- auth_basic_user_file ${file_path}

Health Check

```
upstream backend {
    server a.com:1234 max_fa ils=3 fail_t ime -
out=3s;
    server b.com:1234 max_fa ils=3 fail_t ime -
out=3s;
}
```

被动式Health Check, 响应fails with an error达到3次时, 上游server将-会被标记为不可用

TCP Health Check

```
stream {
    server {
        listen 3306;
        pro xy_pass read_b ackend;
        hea lth _check interv al=10 passes=2
fails=3;
   }
}
```

主动式Health Check, 每10秒发送一次check, 成功2次才被视为Health

Use Includes for Clean Configs

```
http {
    include config.d/ com pre ssi on.c onf;
    include sites- ena ble d/*.conf
}
```

配置文件更为清晰，易于维护

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com