## 基本概念

反向代理：仅向外部暴露代理服务器接口，不暴露目标服务器接口。

负载均衡：多个客户端请求分发到多台目标服务器上

动静分离：动态资源和静态资源由不同的目标服务器管理

代理服务器

8080:8000

8081:8001

8082:8002

目标服务器A

8000

客户端

目标服务器B

80001

目标服务器C

8002

## 安装

Linux上安装nginx

首先安装Nginx运行所需的依赖pcre、openssl、zlib。下面依次介绍利用yum安装和上传各个依赖包安装的步骤。

### yum安装

一句话搞定，建议使用它

yum -y install make pcre zlib zlib-devel gcc-c++ libtool openssl openssl-devel

### 依赖包安装

#### pcre

1. 到<https://ftp.pcre.org/pub/pcre/>下载pcre.tar.gz包，上传到Linux中
2. 执行命令 tar -xvf pcre-x.x.tar.gz解压包
3. 进入解压后的文件夹，执行命令./configure
4. 执行命令make
5. 执行命令make test检查是否编译成功
6. 执行命令make install安装
7. 执行命令pcre-config –version检查是否安装成功

#### openssl

先执行yum安装语句，这块我要慢慢研究

#### nginx

http://nginx.org/en/download.html下载nginx.tar.gz压缩包。上传到Linux中

1. 执行命令 tar -xvf nginx-x.x.tar.gz解压包
2. 进入解压后的文件夹，执行命令./configure
3. 执行命令make
4. 执行命令make install安装
5. 在usr/local路径下有一个名为nginx的文件证明安装成功

## 配置全局变量

执行命令vim /etc/profile打开配置文件

## 启动nginx

进入usr/local/nginx/sbin目录执行命令./nginx

**注意**：如果成功启动后仍然无法访问，查看防火墙端口开放情况

firewall-cmd --list-all

若不存在nginx配置的端口，请打开防火墙对端口的限制

firewall-cmd --zone=public --add-port=80/tcp –permanent

然后重启防火墙

systemctl restart firewalld.service

nginx常用命令

|  |  |  |
| --- | --- | --- |
| 功能 | 命令 | 备注 |
| 查看版本 | ./nginx -v | 需在nginx/sbin目录下执行 |
| 关闭nginx | ./nginx -s stop | 需在nginx/sbin目录下执行 |
| 启动nginx | ./nginx | 需在nginx/sbin目录下执行 |
| 重载nginx | ./nginx -s reload | 不关停nginx，重新加载配置文件，需在nginx/sbin目录下执行 |

## nginx配置

nginx配置文件是usr/local/nginx/sbin/nginx.config

worker\_processes  1;

events {

    worker\_connections  1024;

}

http {

    include mime.types;

    default\_type application/octet-stream;

    sendfile on;

keepalive\_timeout  65;

    server {

       listen 9001;

       server\_name  10.0.0.200;

       location ~/edu/ {

           proxy\_pass http://127.0.0.1:8081;

       }

       location /vod/ {

           proxy\_pass http://127.0.0.1:8080;

       }

    }

}

### 反向代理配置

   server {

       listen 9001;#监听端口号

       server\_name 10.0.0.200;

       location ~/edu/ {#映射端口号配置

           proxy\_pass <http://127.0.0.1:8081;//>后端地址

       }

       location /vod/ {#映射端口号配置

           proxy\_pass <http://127.0.0.1:8080;//>后端地址

       }

   }

### 负载均衡配置

均衡模式

|  |  |
| --- | --- |
| 策略 | 解释 |
| 轮循 | 轮流分配（默认）  upstream myserver{          server 10.0.0.200:8080;          server 10.0.0.200:8081;      } |
| 权重 | 权重越大的服务器分配的客户端的请求越多  upstream myserver{          server 10.0.0.200:8080 weight=1;          server 10.0.0.200:8081 weight=1;      } |
| ip\_hash | 根据ip的hash结果分配，这样每个访客固定访问一个后端服务器，可以解决session问题  upstream myserver{          ip\_hash;          server 10.0.0.200:8080;          server 10.0.0.200:8081;      } |
| fair | 按照后端服务器的响应时间分配，响应时间越短，优先分配。  upstream myserver{          server 10.0.0.200:8080;          server 10.0.0.200:8081;          fair;      } |

    upstream myserver{

        server 10.0.0.200:8080 weight=1;

        server 10.0.0.200:8081 weight=1;

    }

    server {

        listen       80;

        server\_name  10.0.0.200;

        location / {

            root   html;

            proxy\_pass http://myserver;

            #  http://10.0.0.200:8080;

            index  index.html index.htm;

        }

### 动静分离