

## 1. 经典web服务器nginx介绍

Nginx是一款自由的、开源的、高性能的HTTP服务器和反向代理服务器；同时也是一个IMAP、POP3、SMTP代理服务器；Nginx可以作为一个HTTP服务器进行网站的发布处理，另外Nginx可以作为**反向代理**进行**负载均衡**的实现。

### (1) 正向代理

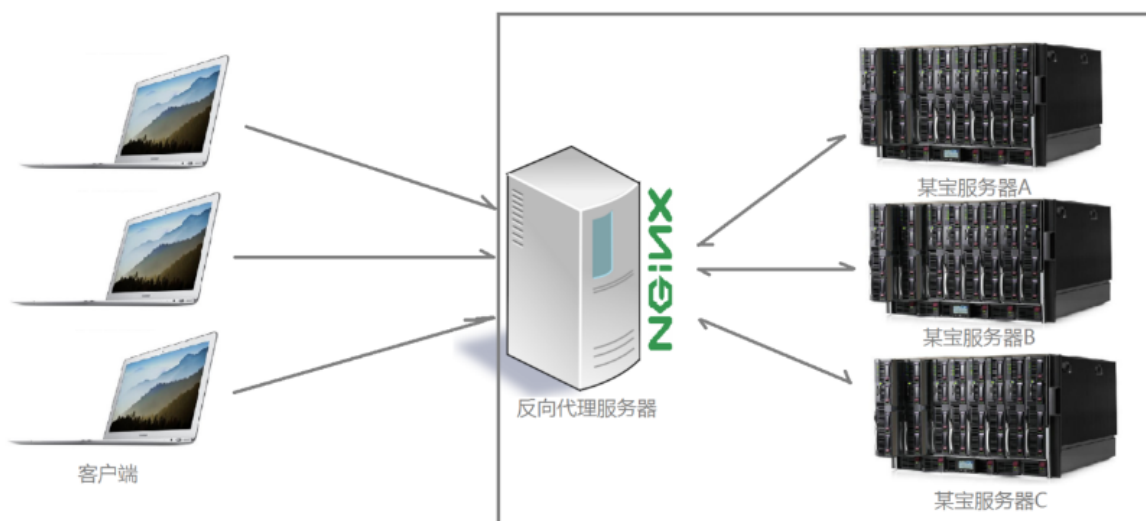
**正向代理最大的特点是客户端非常明确要访问的服务器地址；服务器只清楚请求来自哪个代理服务器，而不清楚来自哪个具体的客户端；正向代理模式屏蔽或者隐藏了真实客户端信息。**



### (2) 反向代理

**客户端是无感知代理的存在的，反向代理对外都是透明的，访问者并不知道自已访问的是一个代理。因为客户端不需要任何配置就可以访问。**

反向代理，"它代理的是服务端"，主要用于服务器集群分布式部署的情况下，**反向代理隐藏了服务器的信息。**



## 2. nginx的基础配置（代理等）

```
1  nginx -c kerwin.conf 加载kerwin.conf 并启动服务器
2  //不能有中文目录
3
4  nginx -s
5  {
6  stop - fast shutdown
7  reload - reloading the configuration file(每次修改完kerwin.conf后 , 都通过
   此方法 重新加载一次)
8
9  }
```

```
1  静态文件serve
2
3  location / {
4  root html;//是当前文件夹下有个html文件夹
5  index index.html index.html
6  }
7
8  location /frontend {
9  root html ;
10 #只要加载localhost/frontend路径 那么就会从 html/frontend/路径提供文件服务
11 }
```

```
1  下面四种情况分别用http://localhost/proxy/test.html 进行访问。
2  // 注意 /proxy/ 后面的/ 需要加上
3
4  (1) location /proxy/ {
5  proxy_pass http://127.0.0.1:8000/;
6  }
7
8  会被代理到http://127.0.0.1:8000/test.html 这个url
9
10 (2)相对于第一种, 最后少一个 /
11 location /proxy/ {
12 proxy_pass http://127.0.0.1:8000;
13 }
14 会被代理到http://127.0.0.1:8000/proxy/test.html 这个url
15
16
17 (3)location /proxy/ {
```

```

18 proxy_pass http://127.0.0.1:8000/xiaoming/;
19 }
20 会被代理到http://127.0.0.1:8000/xiaoming/test.html 这个url。
21
22 (4)相对于第三种，最后少一个 / :
23
24 location /proxy/ {
25 proxy_pass http://127.0.0.1:8000/xiaoming;
26 }
27 会被代理到http://127.0.0.1:8000/xiaomingtest.html 这个url
28
29 // ^~ /proxy 以proxy 开头的

```

### 3. 将nginx部署在线上作为webserver serve 静态资源，并反向代理node服务

## 4. 阿里云

访问规则

入方向

出方向

手动添加

快速添加

全部编辑

| 授权策略 | 优先级 ① | 协议类型    | 端口范围 ①       | 授权对象 ①      |
|------|-------|---------|--------------|-------------|
| 允许   | 1     | 自定义 TCP | 目的:3000/3000 | 源:0.0.0.0/0 |
| 允许   | 1     | 自定义 TCP | 目的:80/80     | 源:0.0.0.0/0 |



华北 2 可用区 H

(公)

(私有)



中

```
1 json-server --watch db.json --host 私有IP地址
```

## 5. 负载均衡

```
upstream kerwin-server {  
    server localhost:3001 weight=1;  
    server localhost:3002 weight=1;  
}  
  
server {  
    listen 80;  
    server_name localhost;  
  
    location / {  
        root dist;  
        index index.html index.htm;  
    }  
  
    location /kerwin/ {  
        proxy_pass http://kerwin-server;  
    }  
}
```