

# 什么是Nginx?

Nginx("engine x")是一款是由俄罗斯的程序设计师Igor Sysoev所开发的高性能的Web和反向代理服务器，也是一个IMAP/POP3/SMTP代理服务器。

由于它的内存占用少，启动极快，高并发能力强，在互联网项目中广泛应用。



## Nginx安装

Ubuntu20.04为例:

```
1 | sudo apt update
2 | sudo apt install nginx
```

安装完成，检查版本号，如果版本号正确显示，则安装成功:

```
1 | nginx -v
```

```
cat@blackcat:~$ nginx -v
nginx version: nginx/1.14.0 (Ubuntu)
```

## Nginx基础命令和配置

基础命令:

```
1 | sudo service nginx status
2 | sudo service nginx stop
3 | sudo service nginx start
4 | sudo service nginx restart
5 | sudo nginx -t          检查默认配置文件
6 | sudo nginx -s reload   重新载入配置文件
```

Nginx日志目录: /var/log/nginx/

Nginx工作目录: /etc/nginx

```
cat@blackcat:~$ cd /etc/nginx
cat@blackcat:/etc/nginx$ ls
conf.d          koi-win          nginx.conf       sites-enabled
fastcgi.conf    mime.types       proxy_params     snippets
fastcgi_params  modules-available scgi_params      uwsgi_params
koi-utf         modules-enabled  sites-available  win-utf
```

Nginx全局配置文件: nginx.conf

```
40
41     access_log /var/log/nginx/access.log;
42     error_log /var/log/nginx/error.log;
43
44     ##
45     # Gzip Settings
46     ##
47
48     gzip on;
49
50     # gzip_vary on;
51     # gzip_proxied any;
52     # gzip_comp_level 6;
53     # gzip_buffers 16 8k;
54     # gzip_http_version 1.1;
55     # gzip_types text/plain text/css application/json application/javascript
56     t text/xml application/xml application/xml+rss text/javascript;
57
58     ##
59     # Virtual Host Configs
60     ##
61     include /etc/nginx/conf.d/*.conf;
62     include /etc/nginx/sites-enabled/*;
63 }
64
```

nginx.conf [R0] 49,0-1 65%

我们在conf.d目录下, 新建.conf文件, 写入自己的nginx配置。

查看进程号:

```
hioier@pc:/run$ cat /run/nginx.pid
2940
```

一个进程允许处理的最大连接数: worker\_connections 768;

## 返回静态页面

先在全局配置文件http段中加入server段:

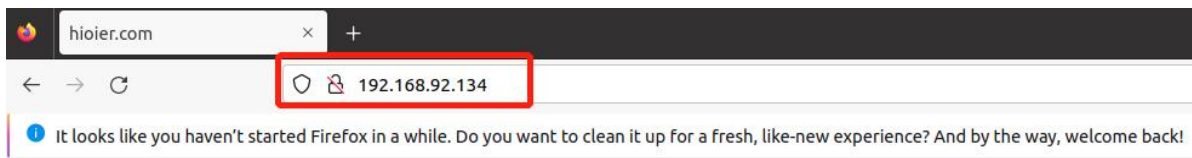
```

60
61     include /etc/nginx/conf.d/*.conf;
62     include /etc/nginx/sites-enabled/*;
63
64     server {
65         listen 192.168.92.134:80;
66         location / {
67             root /etc/nginx/html;
68             index index.html;
69         }
70     }
71 }
72

```

然后执行：sudo nginx -t 测试是否成功。如果成功，执行sudo nginx -s reload重新载入配置文件。

默认80端口，访问成功：



**hioier.com**

**注意：非root用户修改全局配置文件先要执行chmod增添写入权限。**

然而，一般我们不会直接修改默认配置文件，在conf.d目录下新建配置文件：

```

1  server {
2      listen 192.168.92.134:8001;
3      location / {
4          root /etc/nginx/conf.d/html;
5          index index.html;
6      }
7  }

```

```

server {
    listen 192.168.92.134:8001;
    location / {
        root /etc/nginx/conf.d/html;
        index index.html;
    }
}

```

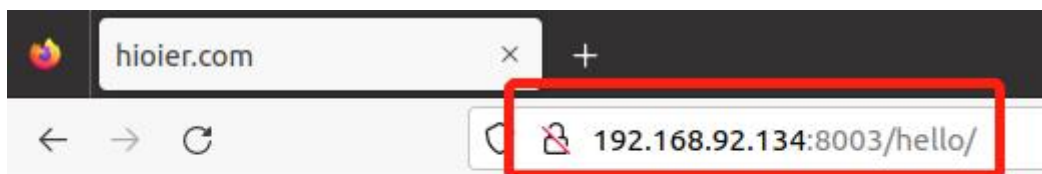
return 302 重定向到其他网站：

```
server {  
    listen 192.168.92.134:8002;  
    location / {  
        return 302 http://47.101.159.27:9001;  
    }  
}
```

使用302重定向，访问本地网址，跳转后，地址栏显示的还是跳转后的地址。比如，我更换了网站域名，但是希望老用户可以继续找得到新网站，所以做个跳转功能。

root 表示 location 匹配内容的相对路径，alias 表示 一个绝对路径,而且必须以"/"结尾：

```
server {  
    listen 192.168.92.134:8003;  
    location /hello {  
        alias /etc/nginx/conf.d/html/;  
        index index2.html;  
    }  
}
```

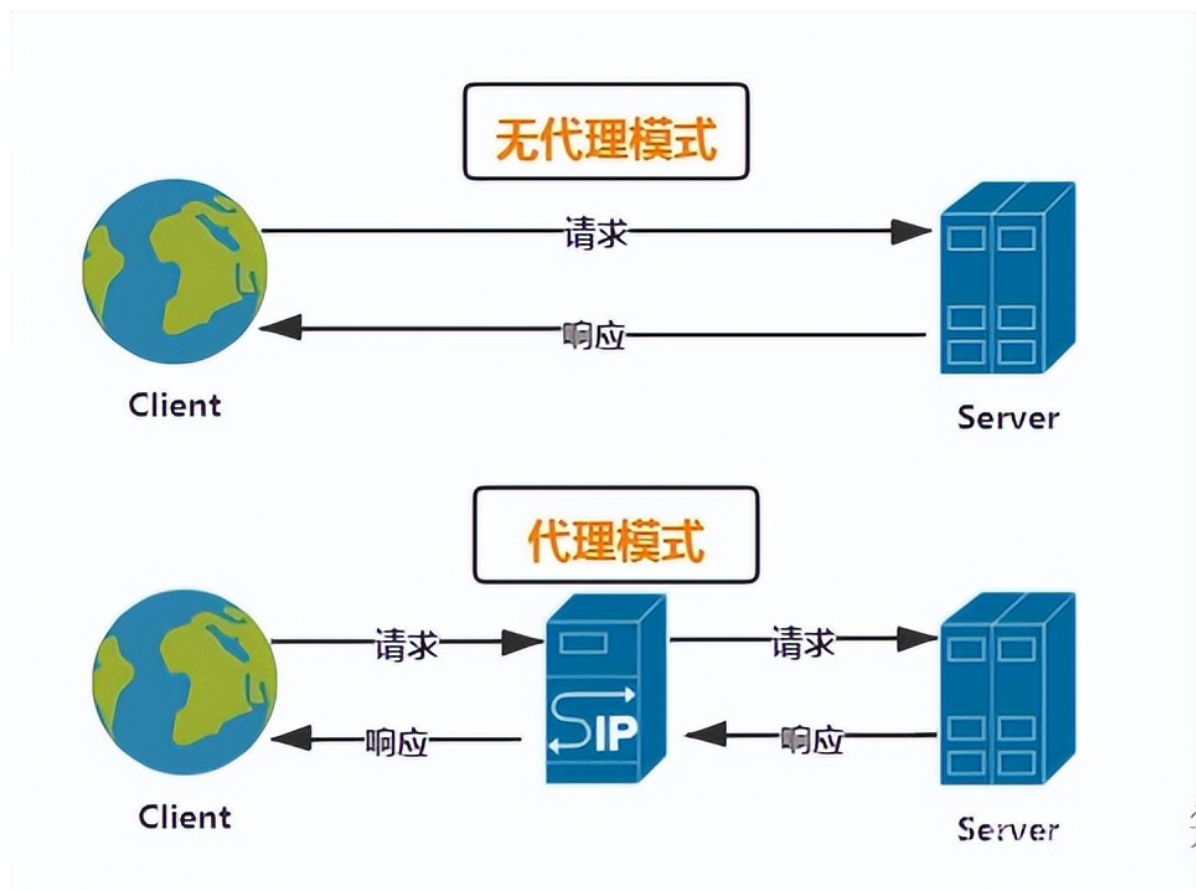


**hioier.com 8003 hello**

## 反向代理

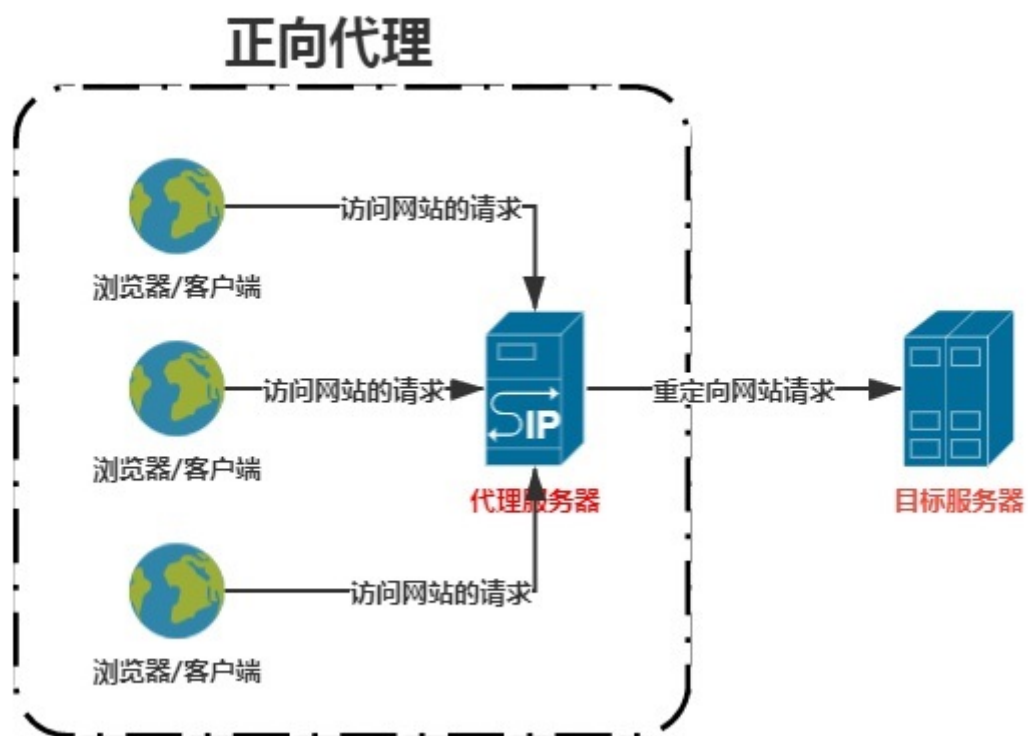
什么是代理服务器？举个例子，找工作直接和老板谈薪资待遇，这就是没有代理，但是大公司老板不会直接面试招聘，就通过HR进行代理，然后老板只负责盖章敲定。





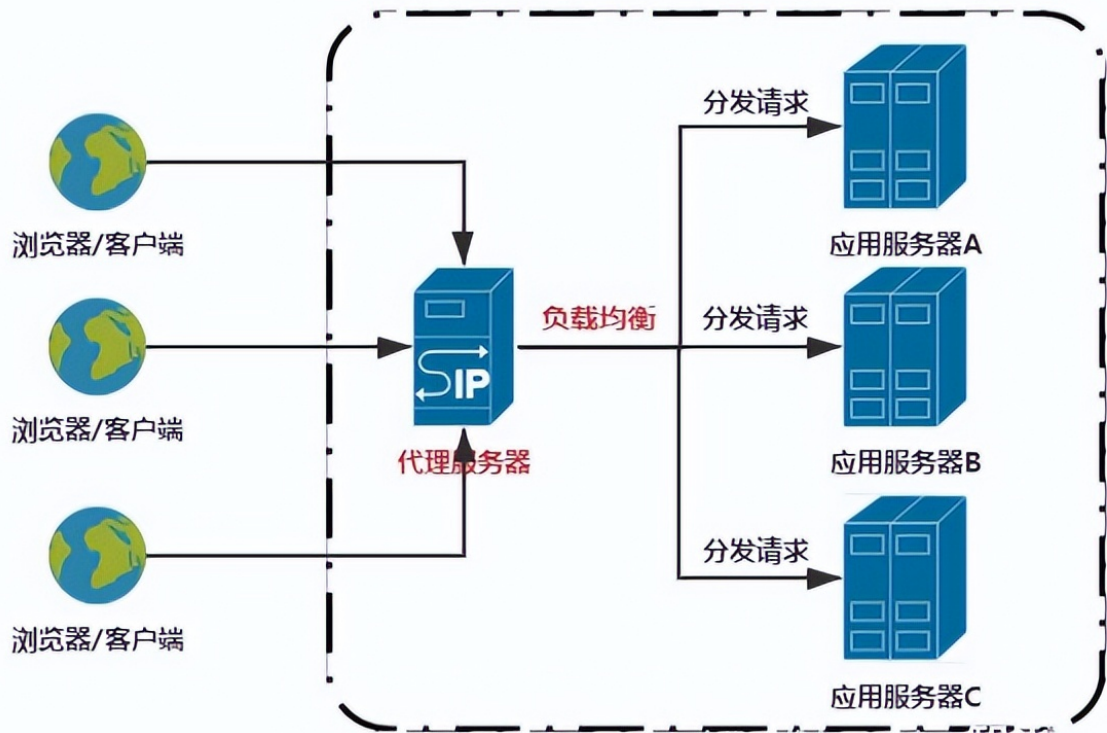
真实客户端直接访问真实服务器就是没有代理。网络中，代理又分为正向代理和反向代理。

**正向代理：**比如我在大陆服务器要访问Google，但是无法直接访问，然而我可以访问香港服务器，香港服务器可以访问Google，那我就用香港服务器做代理，实现在大陆直接访问外网。



**正向代理**是隐藏客户端，**反向代理**是隐藏服务器。比如，张艺谋的《影》，讲的是关于替身的故事，替身即使被杀死，也不会影响真身。所以，**真实服务器作为大Boss，让小弟服务器做代理，黑客只能攻击小弟代理服务器，不会影响真实服务器。**

# 反向代理



如上图，大家只可以访问代理服务器，代理服务器再向真实服务器请求数据，同时真实服务器可以是多个，这就叫做负载均衡。

```
1 server {
2     listen 80;
3     server_name blog.hioier.com;
4     location / {
5         proxy_pass http://localhost:8080;
6     }
7 }
8
9 server {
10    listen 80;
11    server_name blackcat1995.com www.blackcat1995.com;
12    location / {
13        proxy_pass http://localhost:9001;
14    }
15 }
16
```

除了隐藏真实服务器，还有一个现实的需求，那就是一个服务器要开启多个服务，但是，一个服务器只有一个80端口，如何让多个服务共用80端口？

可以先给不同服务分配不同端口，比如8001,8002，但是都监听80端口。服务器名字用不同的域名区分，访问不同的域名，默认都是80端口，然而实际上访问的是真正的端口。

## 负载均衡

当访问量急剧增大，只有一台服务器难以应答，可以考虑使用多个服务器分散压力。比如，一个请求到达，按照服务器1,2,3的顺序依次应答，这样就可以减轻单个服务器的压力。

```
upstream fuzai {  
    server 47.101.159.27:9001;  
    server 192.168.92.134:8001;  
    server 192.168.92.134:8003;  
}  
  
server {  
    listen 192.168.92.134:80;  
    location / {  
        proxy_pass http://fuzai;  
    }  
}
```

正常每个服务器的内容都是一样的，但是为了测试方便，使用不同的网页应答，这种默认的负载均衡方式就是面对请求，依次应答的，这种形式也叫做轮询算法。

然而，不同服务器可能性能不同，一般主服务器性能最佳，访问时应答权重更高一些，能力越大责任越大，这就可以使用加权轮询算法，给不同服务器分配应答权重。

```
upstream fuzai {  
    server 47.101.159.27:9001 weight=1;  
    server 192.168.92.134:8001 weight=3;  
}
```

## ssl证书

SSL证书是一种数字证书，主要是给予网站HTTPS安全协议加密传输与信任的功能。ssl证书分为免费版和付费版，一般网站长期使用还是付费证书比较稳定，到期续费，平时使用可以搜索一下免费证书以供练习。

数字证书管理服务

SSL证书

概览

SSL证书3

私有证书

证书应用仓库NEW

域名监控

网站代理HTTPS

托管服务

订单管理

信息管理

证书工具

专家一对一服务

数字证书管理服务 / SSL证书

SSL证书

证书管理免费证书上传证书CSR管理订单管理

### 免费证书年度清零说明

尊敬的用户，根据阿里云免费证书年度使用规则，2023年1月1日0点清除历史免费证书。已签发证书可以继续使用无任何影响。如需继续使用，点击立即购买重新领取。

[查看免费证书使用规则](#)

立即购买创建证书 18 / 20全部状态请输入域名

证书	品牌/算法	状态	绑定域名	有效期限	已部署	操作
cert-8762769 DigiCert 免费版 SSL 资源ID: cas-ivauto-bd8yu1 <a href="#">标签未设置标签</a>	RSA	已签发	cat.hioier.com	1年 2024-01-24	--	<a href="#">部署</a> <a href="#">下载</a> <a href="#">部署服务</a>
cert-8762234 DigiCert 免费版 SSL 资源ID: cas-ivauto-hqhgux <a href="#">标签未设置标签</a>	RSA	已签发	blog.hioier.com	1年 2024-01-24	--	<a href="#">部署</a> <a href="#">下载</a> <a href="#">部署服务</a>

每页显示 | 10 | 20 | 50 共 2 条数据 < 上一页 1 下一页

然后访问https就可以啦：

# hioier.com



```
8      ssl_ciphers ECDHE-RSA-AES128-GCM-  
SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4;  
9      ssl_protocols TLSv1.1 TLSv1.2 TLSv1.3;  
10     ssl_prefer_server_ciphers on;  
11  
12     location / {  
13         root /etc/nginx/conf.d/html;  
14         index index.html;  
15     }  
16 }
```