



# Nginx 服务器

🏷 标签      空

+ 新增属性

官网地址: <https://nginx.org/>



Nginx 是一款轻量级的 Web 服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器。

## Windows 启动 nginx

纯文本

```
1 1. 下载 : http://nginx.org/
2 2. 修改 conf/nginx.conf
3 2. 启动 nginx.exe
```

## 目录介绍

名称	修改日期	类型	大小
conf	2020/9/29 星期...	文件夹	
contrib	2020/9/29 星期...	文件夹	
docs	2020/9/29 星期...	文件夹	
html	2020/9/29 星期...	文件夹	
logs	2021/2/4 星期四 ...	文件夹	
temp	2020/10/5 星期...	文件夹	
nginx	2020/9/29 星期...	应用程序	3,653 KB

配置文件目录

可执行程序，双击运行

conf: 内部为各种配置文件; html: 以 nginx 作为静态服务器时, 默认的静态文件存放位置。

## Nginx 配置

打开 nginx.conf 文件并删掉注释部分后, 配置内容如下:

```

1
2 worker_processes 1;
3
4 events {
5     # 连接数上限
6     worker_connections 1024;
7 }
8
9 # 设定http服务器，可以进行反向代理、负载均衡
10 http {
11     include mime.types;
12     default_type application/octet-stream;
13
14     sendfile on;
15     keepalive_timeout 65; # 连接超时时间
16
17     server {
18         listen 80; # 监听的端口
19         server_name myserver; # 服务名称
20
21         #-----默认代理页面: url【nginx所在主机IP:监听的端口/】访问的文件
22         location / {
23             root html; # 指定根目录
24             index index.html index.htm; # 指定启动文件index.html
25         }
26
27         error_page 500 502 503 504 /50x.html;
28         location = /50x.html {
29             root html;
30         }
31     }
32 }
33
34
35

```

[https://blog.csdn.net/qq\\_52681418](https://blog.csdn.net/qq_52681418)

在这里插入图片描述

配置文件中网络主要分3块：

### 1.http协议块

```

1 # 协议级别
2 http {
3     # 服务级别
4     server {
5         # 请求级别
6         location / {
7             ...
8         }
9     }
10 }

```

nginx

### 2.server服务块

```

1 server {
2     listen 8080; # 监听端口，也是外部进行请求的端口。
3     server_name myServer; # 服务名称。
4     location /user { # 代理的请求
5         # http://localhost:8080/user -> http://www.baidu.com/
6         proxy_pass http://www.baidu.com/;
7     }
8 }
9 }

```

nginx

### 3.location请求块

```

1 server {
2     listen 8080;
3     server_name myServer;
4
5     # 代理本地静态资源
6     location /index {
7         root html; # 文件目录，html目录下
8         index index.html index.htm; # 入口文件
9     }
10
11     # 代理远程地址
12     location /user {
13         proxy_pass http://www.baidu.com/;
14     }
15 }
16 }

```

nginx

location匹配规则：

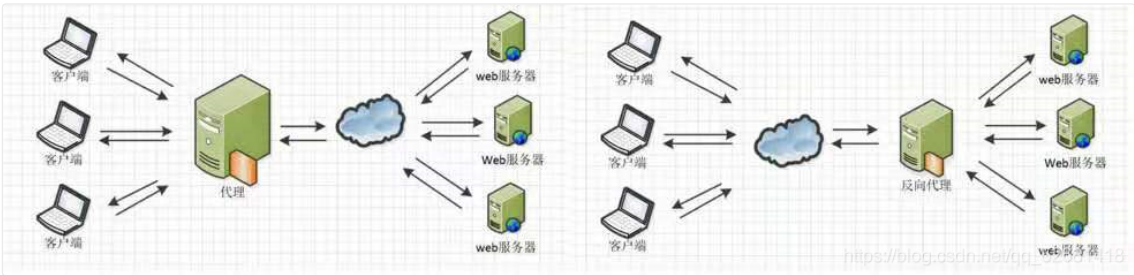
nginx

```
1 location [起始符+] 正则表达式{
2     .....
3 }
```

起始符	作用
无	请求时，路径开头部分和location一致即可匹配
=	请求时，路径必须和=号后面的内容完全一致才能匹配 (不包括请求参数)
~	后面的正则表达式区分大小写，满足正则表达式即可匹配
~*	后面的正则表达式不区分大小写，满足正则表达式即可匹配

反向代理

- 正向代理：将不同的请求地址，代理到同一个服务地址。
- 反向代理：将同一个请求地址，代理到不同的服务地址。



在这里插入图片描述

正向代理举例

```
server {
    listen      8081;
    server_name myserver;

    location /user {
        proxy_pass http://192.168.25.200:8080/;
    }
}

server {
    listen      8082;
    server_name myserver;

    location /user {
        proxy_pass http://192.168.25.200:8080/;
    }
}

server {
    listen      8083;
    server_name myserver;

    location /user {
        proxy_pass http://192.168.25.200:8080/;
    }
}
```

正向代理，将8081、8082、8083代理到：http://192.168.25.200:8080/

在这里插入图片描述

反向代理举例

```
server {
    listen      8080;
    server_name myserver;

    location /user1 {
        proxy_pass http://192.168.25.200:8081/;
    }
    location /user2 {
        proxy_pass http://192.168.25.200:8082/;
    }
    location /user3 {
        proxy_pass http://192.168.25.200:8083/;
    }
}
```

反向代理：将3个不同地址都代理到本机8080端口，分别是/user1、/user2、/user3

[https://blog.csdn.net/qq\\_52681418](https://blog.csdn.net/qq_52681418)

在这里插入图片描述

## 负载均衡

nginx 负载均衡是通过反向代理实现的，对多个不同主机上的相同服务进行反向代理，并实现平衡调用。在 http 块中添加如下配置：

```
# 配置代理服务器的地址，即Nginx安装的服务器地址、监听端口、默认地址
# weigh参数表示权值，权值越大，使用的几率越大
# down表示不参与负载
# backup，备用，其它机器忙时才用它

upstream nacos{
    server 127.0.0.1:9848 weight=2;
    server 127.0.0.1:9858 weight=1;
    server 127.0.0.1:9868 weight=1;
}
```

R

在 server 块中设置代理：

```
# 配置nginx服务
server {
    listen      9099;          # 1.监听端口，即访问的端口
    server_name localhost;    # 服务名：对于server_name,如果要将多个域名的请求进行反向代理，可以配置多个server_name来满足要

    #对上面三台nacos进行代理： 问本机9099时会随机访问上面3个地址
    location /nacos/ {
        proxy_pass http://nacos;
    }
    ...
}
```

R

## 静态服务器

静态文件即js、html、图片等文件，通常情况下这些文件放在服务器上，外部无法直接访问，因此放在 tomcat 下可以实现访问。有时候我们不想用 tomcat，那么通过使用 nginx 也可以部署静态资源。

在服务器上部署静态资源很简单，下面就以前端 vue 项目举例：

1. 下载nginx
2. 添加一个location配置，根目录指向项目的根目录（vue打包后为list目录），起始页填项目的首页文件名。
3. 启动nginx，即可访问。

纯文本

在步骤 2 中，可以不修改配置，因为默认的起始页为 html/index.html 文件，因此将 list 内的文件复制到 nginx 下的 html 目录即可。

linux 上常用 nginx 命令

PowerShell