

# 一、Nginx介绍

---

## 1.1 为什么要使用Nginx

### 1.1 静态资源访问

前端项目在进行服务器部署的时候，如果基于tomcat则不能满足并发的需求，因此我们需要一种能够处理更高并发的适用于静态资源的服务器

### 1.2 代理服务器-负载均衡

## 1.2 Nginx介绍

Nginx (engine x) 是一个高性能的HTTP和反向代理web服务器

- Nginx 是基于HTTP协议的请求/响应（部署web项目） ---- 静态资源服务器
- Nginx可以作为反向代理服务器 ---- 负载均衡服务器(代理服务器)

Nginx特点：

- 稳定性极强，可以实现7\*24不间断运行
- 提供简洁、直观的配置
- 占用内存很小、并发能力很强（5w+）

# 二、Nginx下载、安装及配置

---

## 2.1 windows环境

### 2.1 .1下载

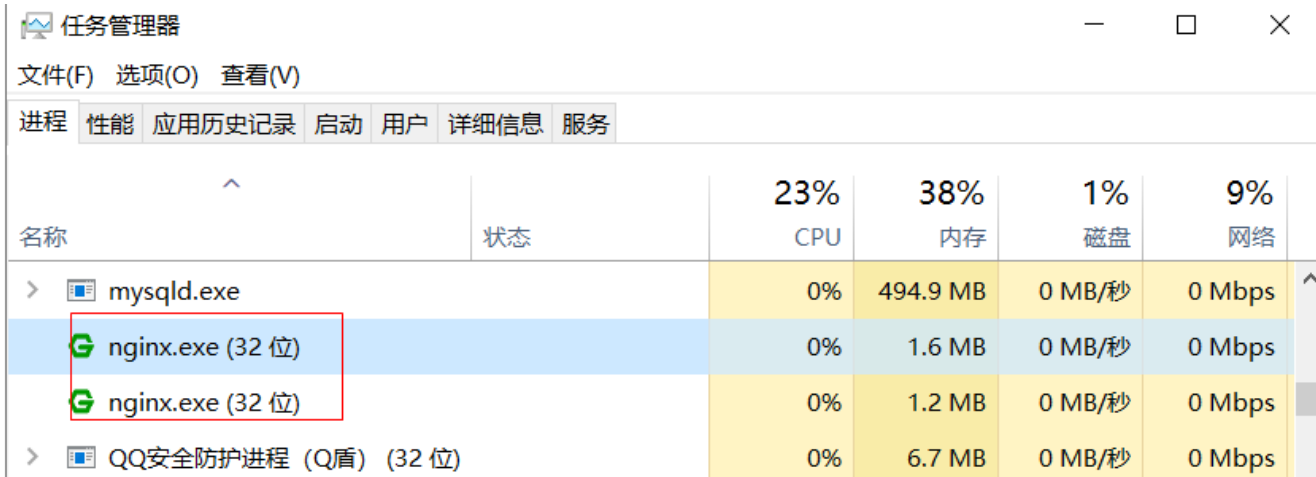
<http://nginx.org/en/download.html>

### 2.1.2 解压到指定目录

解压即可

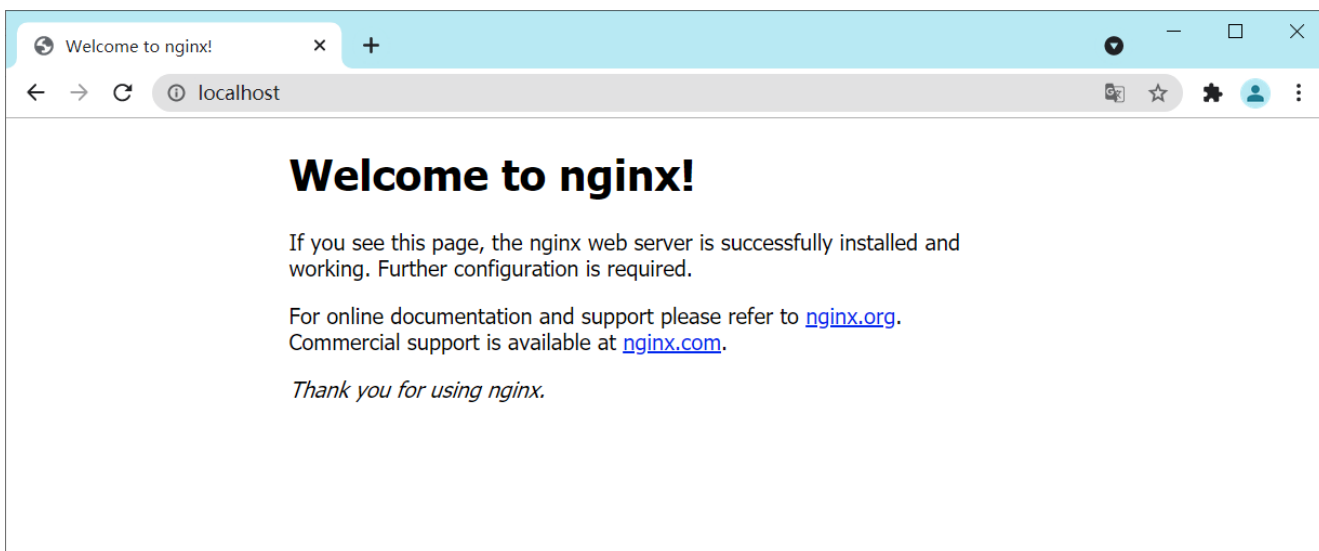
### 2.1.3 运行及访问

- 执行根路径下的nginx.exe



任务管理器						
文件(F) 选项(O) 查看(V)						
进程 性能 应用历史记录 启动 用户 详细信息 服务						
名称	状态	23% CPU	38% 内存	1% 磁盘	9% 网络	
> mysql.exe		0%	494.9 MB	0 MB/秒	0 Mbps	^
nginx.exe (32 位)		0%	1.6 MB	0 MB/秒	0 Mbps	
nginx.exe (32 位)		0%	1.2 MB	0 MB/秒	0 Mbps	
> QQ安全防护进程 (Q盾) (32 位)		0%	6.7 MB	0 MB/秒	0 Mbps	

- nginx 默认端口 80,所以直接访问：<http://localhost>



## 2.2 Linux环境

# 三、Nginx配置

- 配置文件说明

```
1 #user  nobody;
2
3 # 配置处理器个数 (应不大于cpu核心数)
4 worker_processes 1;
5
6 # 配置nginx运行日志文件
7 error_log logs/error.log;
8
9 # 配置Nginx进程的进程号的存储文件
10 # pid logs/nginx.pid;
```

```
11
12
13 # 配置nginx连接数, 数值越大并发能力越强
14 events {
15     worker_connections 2000;
16 }
17
18 http {
19     include mime.types;
20     default_type application/octet-stream;
21
22
23     sendfile on;
24
25     keepalive_timeout 65;
26
27     server {
28         # 配置nginx端口号
29         listen 80;
30         server_name localhost;
31
32         # 地址映射
33         #location / {
34             # root html;
35             # index index.html index.htm;
36         #}
37
38         location / {
39             root fmall-static;
40             index index.html index.htm;
41         }
42
43         # 指定错误页面
44         error_page 500 502 503 504 /50x.html;
45         location = /50x.html {
46             root html;
47         }
48
49     }
50
51 }
52
```

## 四、锋迷商城—前端部署

---

- 安装Nginx
- 将前端项目fmall-static拷贝到nginx的根目录
- 修改nginx.conf:

```
1 location / {  
2     root    fmall-static;  
3     index  index.html index.htm;  
4 }
```

千锋教育Java教研院 关注公众号【Java架构栈】下载所有课程代码课件及工具 让技术回归本该有的纯静！