

计算机网络实验报告



实验名称：Web 服务的配置与管理

学院名称：计算机科学与通信工程学院

专业班级：物联网工程2303

学生姓名：邱佳亮

学生学号：3230611072

教师姓名：李峰

报告日期：2024/11/21

# 目录

[目录 1](#_Toc25796)

[1 Nginx 的配置 1](#_Toc3934)

[1.1 实验目的 1](#_Toc30453)

[1.2 实验思路 1](#_Toc30766)

[1.3 实验步骤 1](#_Toc16888)

[1.3.1 安装配置 Nginx 1](#_Toc23532)

[1.3.2 配置负载均衡 3](#_Toc18693)

[1.3.3 配置运行多个网站 4](#_Toc11786)

[1.3.4 配置访问控制 5](#_Toc8828)

[1.4 思考与提高 6](#_Toc8528)

[2 总结和收获 8](#_Toc31814)

## Nginx 的配置

### 实验目的

（1） 了解 Nginx 的工作原理，掌握 Nginx 的安装与配置方法；

（2） 掌握在一台 nginx 服务器上配置运行多个网站的方法；

（3） 了解 Nginx 负载均衡的实现原理，能够利用 Nginx 进行负载均衡；

（4） 了解 Nginx 访问控制的实现原理，能够利用 Nginx 进行访问控制。

### 实验思路

（1）安装配置 Nginx，并验证已经安装成功；

（2）在一台 nginx 服务器上基于不同端口配置运行多个网站；

（3）在 nginx 服务器配置负载均衡，并测试其效果；

（4）在 nginx 服务器配置访问控制，并测试其效果。

### 实验步骤

#### 安装配置 Nginx

访问nginx官网，下载nginx：

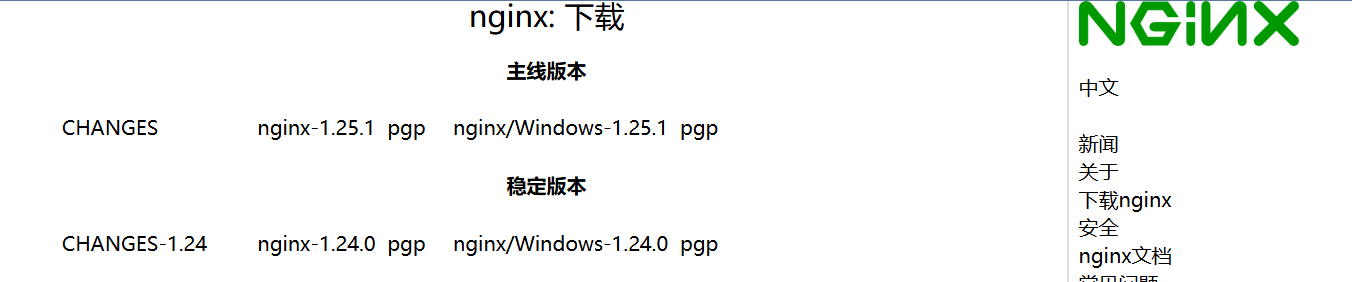


图 1 下载nginx

解压：

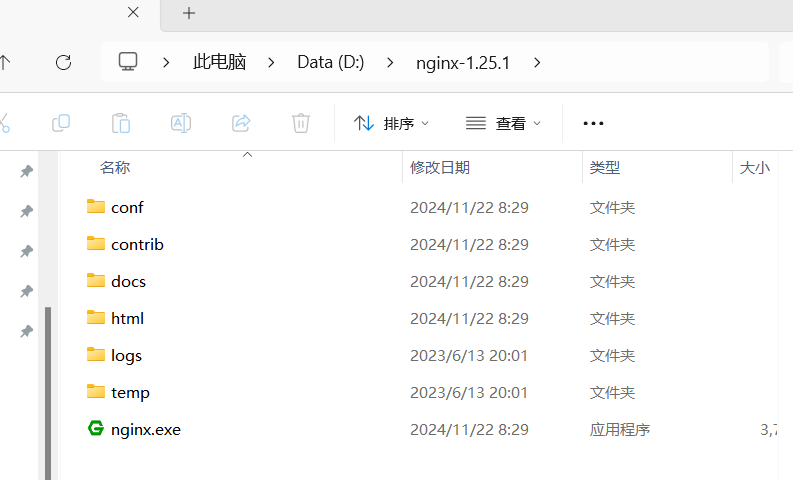


图 2 解压

在/conf下找到配置文件并修改端口为81：



图 3 修改端口

通过命令提示符进入nginx文件夹，启动nginx服务：

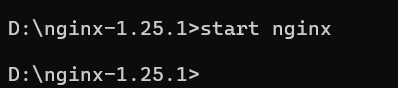


图 4 启动nginx

访问localhost:81，显示访问成功：



图 5 访问成功

#### 配置负载均衡

配置负载均衡，在http块中定义后端服务器，并设置负载均衡规则，设置了两个负载服务器127.0.0.1:8080和127.0.0.1:8081，其root目录分别为backend1和backend2：



图 6 配置负载均衡

通过命令行测试nginx配置文件是否正确，并重新加载nginx：

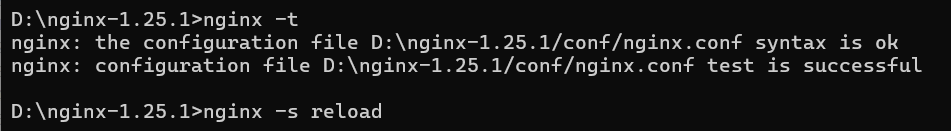


图 7 重新加载nginx

在两个服务器的根目录下创建index.html文件：

屏幕截图 2024-11-22 085737

图 8 index文件

通过命令行启动两个服务：

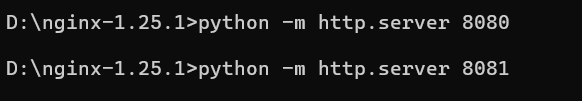


图 9 启动两个服务

此时访问localhost:81，两个后端服务会交替响应：

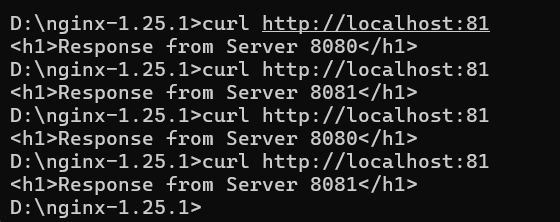


图 10 交替响应

#### 配置运行多个网站

创建两个网站的root目录：

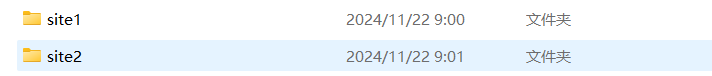


图 11 创建目录

在目录下创建网站的index：

屏幕截图 2024-11-22 090136

图 12 index

修改配置文件，为两个网站添加server配置：

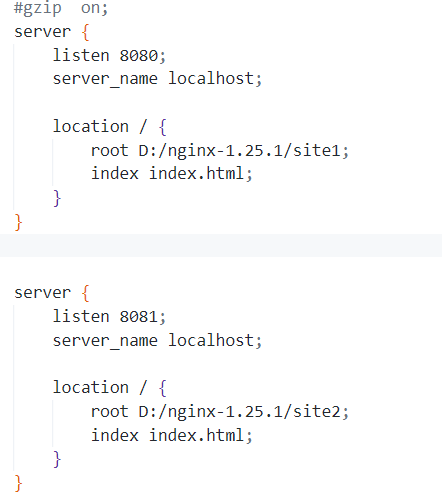


图 13 修改配置

检查配置文件并重启服务：

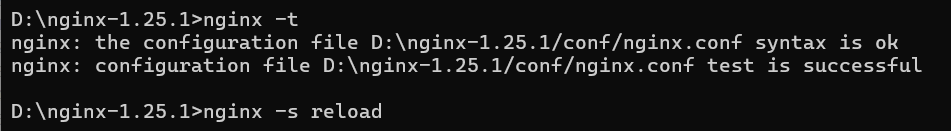


图 14 重启服务

可以访问两个网站的内容：

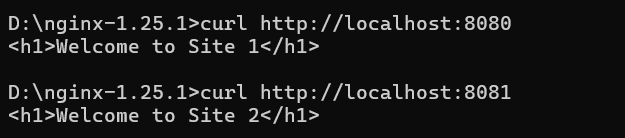


图 15 访问网站

#### 配置访问控制

编辑配置文件，允许特定ip地址访问并拒绝其他ip地址访问：

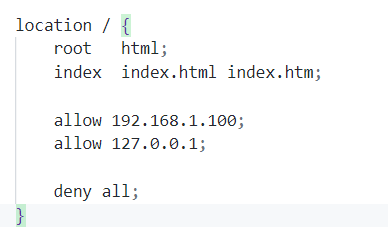


图 16 修改配置文件

检查配置文件并重启服务：

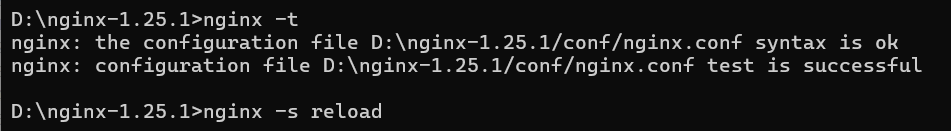


图 17 重启服务

此时非特定ip无法访问nginx服务：

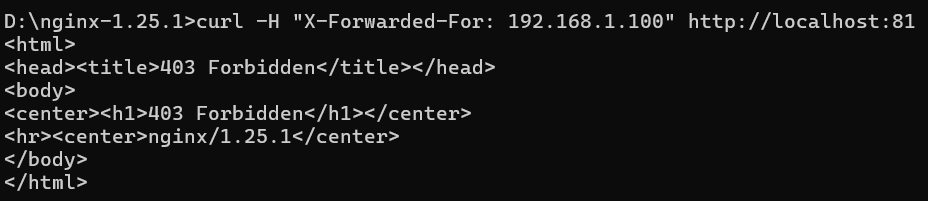


图 18 访问失败

### 思考与提高

跨域问题是指在 B/S架构中，当浏览器发起请求的协议、域名或端口与当前页面的地址不一致时，受到浏览器同源策略的限制而导致请求被阻止的现象。浏览器的同源策略是一种安全机制，旨在防止恶意网站窃取用户数据或发送未经授权的请求。Nginx可以通过配置CORS支持或使用反向代理转发请求解决跨域问题。

在conf文件中添加cors配置：



图 19 添加配置

前端文件如下：



图 20 前端文件

访问页面，点击fetch data按钮进行跨域请求：



图 21 跨域请求

查看浏览器控制台网络部分，发现收到后端返回数据：

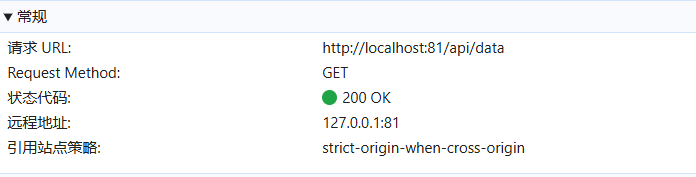


图 22 数据包

查看响应标头，看到cors运行成功：

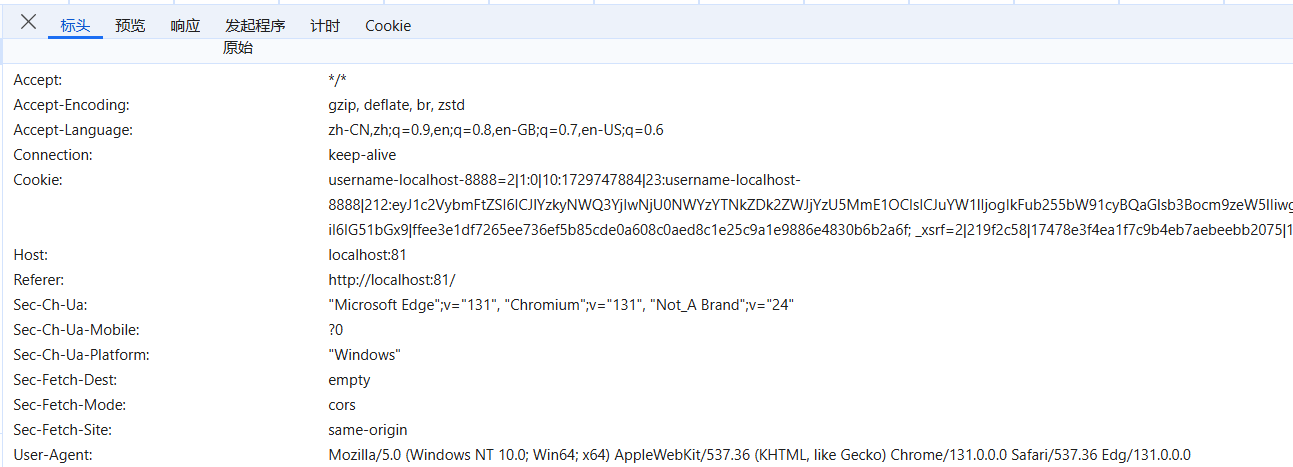


图 23 响应标头

## 总结和收获

通过本次实验，我深入了解了 Nginx 的安装、配置及其多种功能的实现过程。首先，成功完成了 Nginx 的安装与启动，并通过修改配置文件调整服务端口和站点路径，验证了服务的正常运行。接着，在一台 Nginx 服务器上配置了基于不同端口运行多个网站，验证了多站点同时运行的效果。同时，通过设置负载均衡，将用户请求分发至不同的后端服务器，实现了轮询交替响应的功能。此外，还配置了访问控制策略，通过允许特定 IP 地址访问并拒绝其他 IP 地址的方式，验证了访问权限的有效性。通过本次实验，不仅掌握了 Nginx 的基本使用方法，还对其灵活的配置和强大的功能有了更深入的理解，为后续实际应用奠定了良好的基础。