**实验报告（第 8 次）**

实验名称 Nginx负载均衡集群

实验时间 2025.06.03

同组同学 无 小组分工 暂无

**一、实验目的**

1.理解负载均衡的概念：掌握负载均衡的基本原理，了解如何通过Nginx实现请求的分发，提升系统的可扩展性、可用性和性能。

2.掌握Nginx配置：学习如何配置Nginx进行反向代理和负载均衡，掌握Nginx负载均衡算法的使用与配置。

3.分布式部署的实现：通过分布式部署多个应用实例，并配置Nginx进行负载均衡，确保多个服务器之间的流量均衡分配，提高系统的容错性和高可用性。

4.提高系统可靠性：通过Nginx进行健康检查和故障转移，保证系统在某一节点宕机时，流量能够平滑切换到其他节点。

5.优化系统性能：通过负载均衡将请求均匀分配到多台服务器，提高系统的吞吐量和响应速度，减轻单台服务器的压力。

**二、实验仪器设备或材料**

1.计算机：运行Windows、Linux或macOS操作系统的计算机。

2.Nginx环境：已安装并配置好Nginx，作为反向代理服务器进行负载均衡。

3.文本编辑器或IDE：GoLand（推荐）或其他，用于编写Go代码。

4.Go语言环境：已安装并配置好Go语言开发环境。

5.后端服务：至少两台或更多后端服务器，用于实现分布式部署，可以使用本地多个端口模拟多个服务实例。

**三、实验原理**

1.负载均衡原理：负载均衡是指将客户端的请求按一定策略（如轮询、最少连接数、IP哈希等）分发到多个后端服务器，从而实现流量的均衡分配。通过负载均衡，可以提高系统的吞吐量、容错性，并实现高可用的分布式架构。

2. Nginx反向代理：Nginx作为反向代理服务器，客户端的请求首先到达Nginx，由Nginx根据配置将请求转发到后端的多个服务器。Nginx负载均衡通过代理请求的分发来实现系统的分布式部署。

3. Nginx负载均衡算法：

轮询（Round Robin）：请求均匀地分配到所有服务器。

加权轮询（Weighted Round Robin）：根据每台服务器的权重分配请求，权重高的服务器将接收到更多的请求。

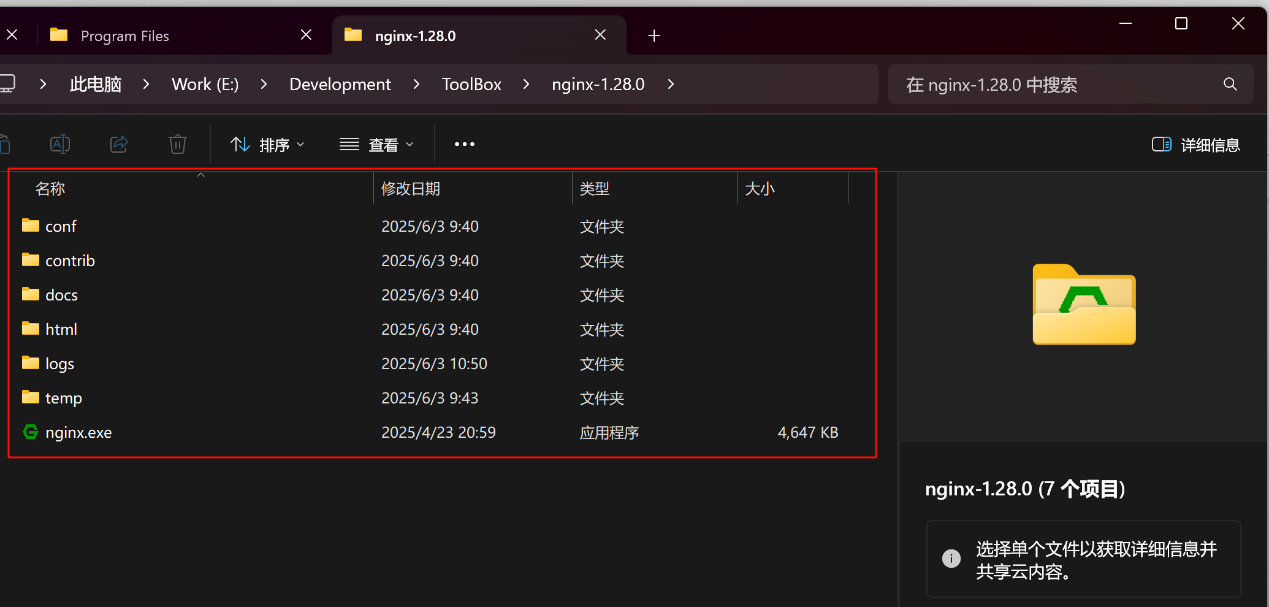
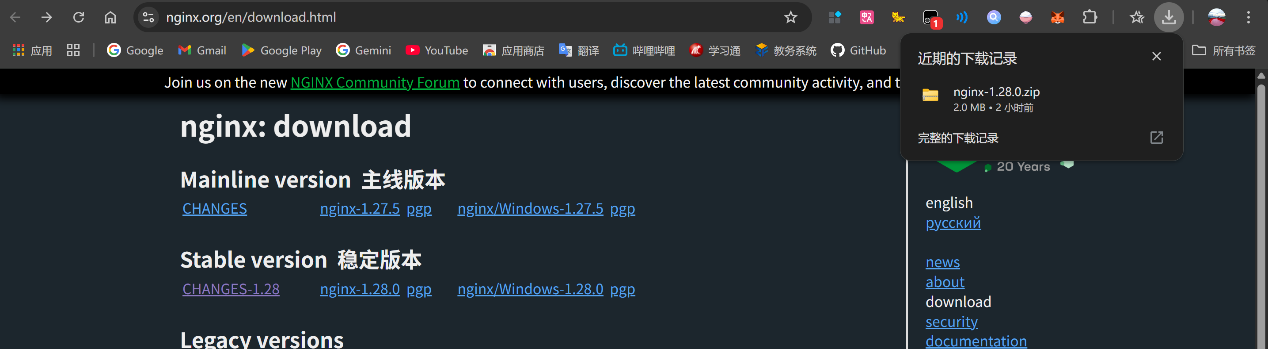
最少连接（Least Connections）：将请求转发给当前连接数最少的服务器。

IP哈希（IP Hash）：根据客户端的IP地址进行哈希计算，将请求转发到特定的服务器。

4.健康检查：Nginx支持定期检查后端服务器的健康状态，如果某台服务器不可用，Nginx会自动将请求转发到健康的服务器，确保系统的高可用性。

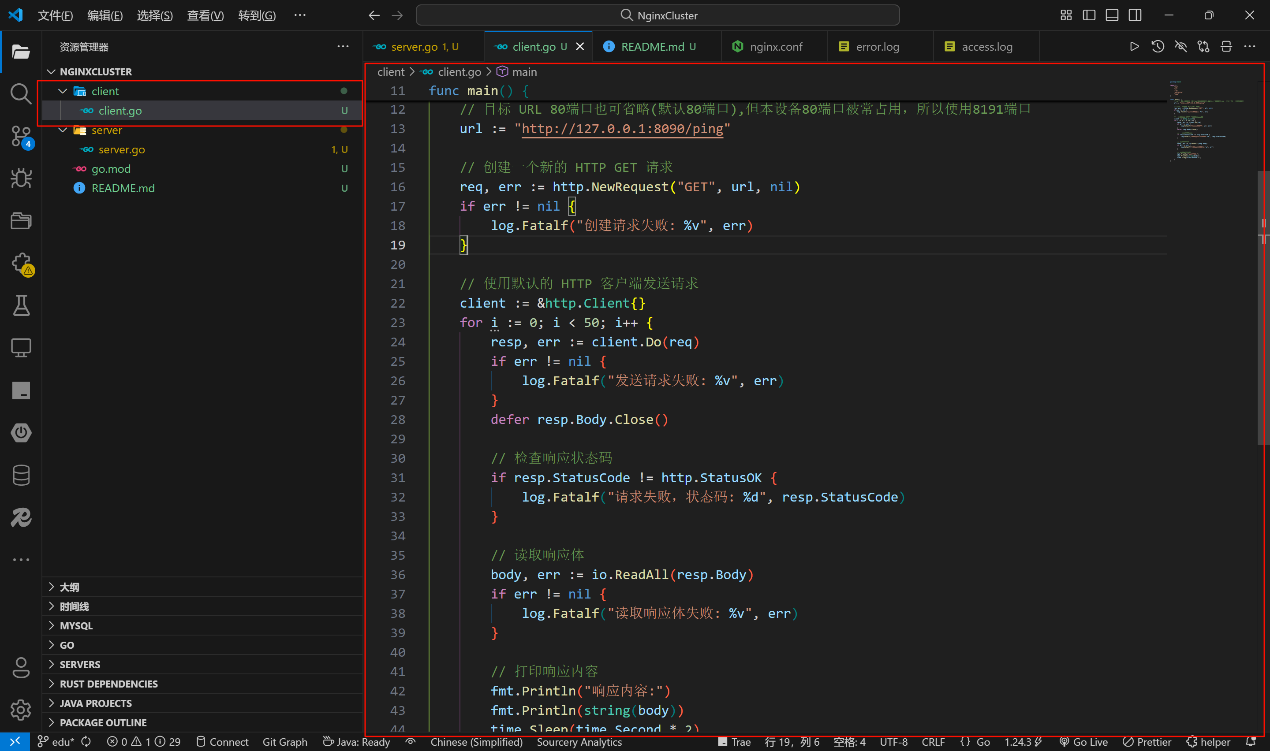
**四、实验内容与步骤**

1. 前往官网下载Nginx最新Windows压缩包

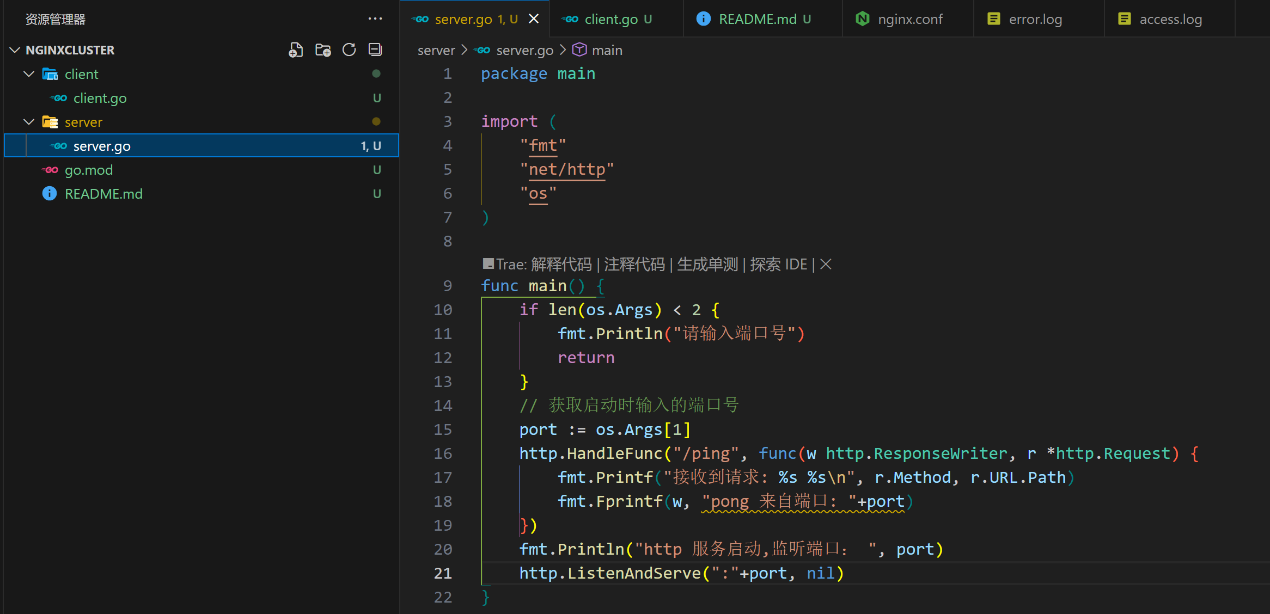


2. 创建NginxCluster项目，并新建相关文件和自定义第三方包

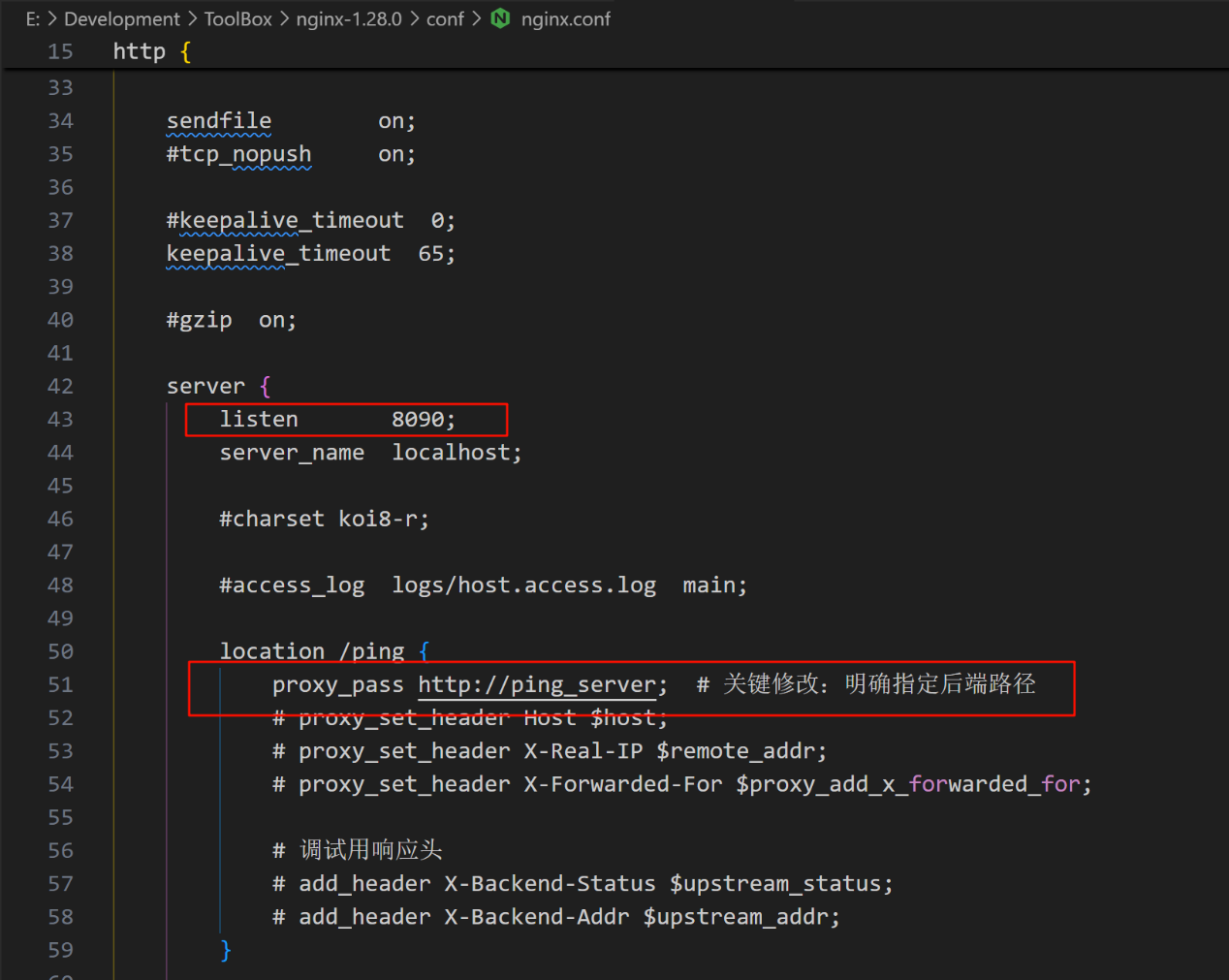
client\client.go

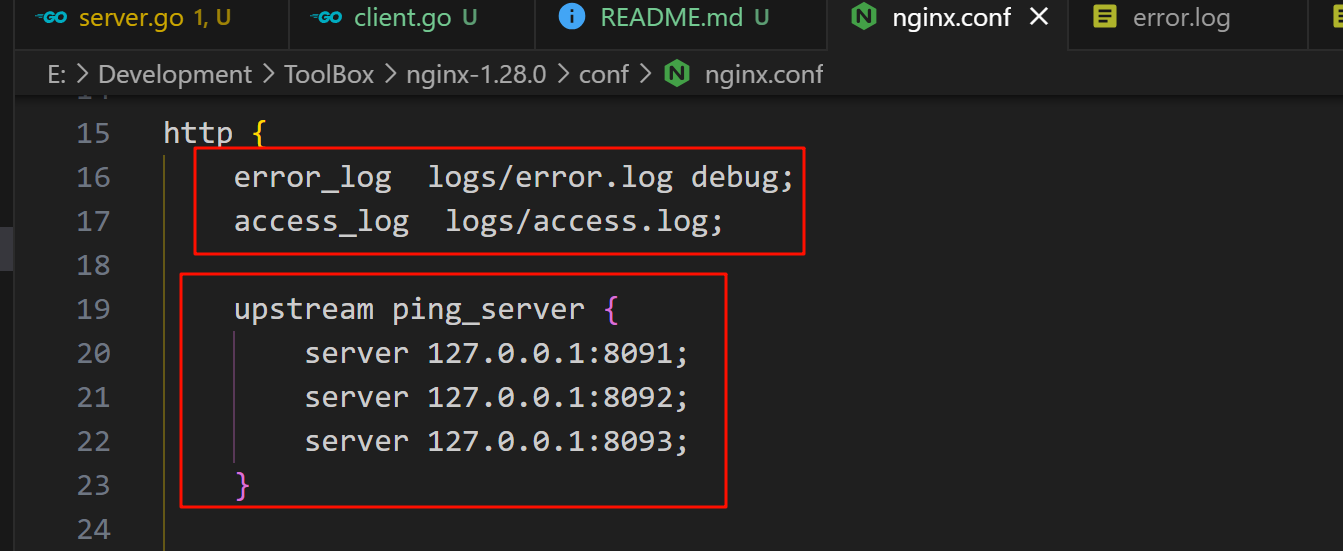


server\server.go

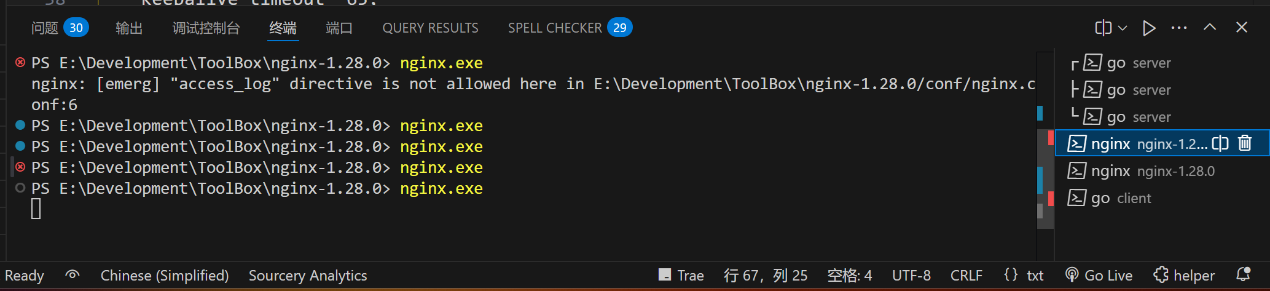


3．配置nginx.conf文件

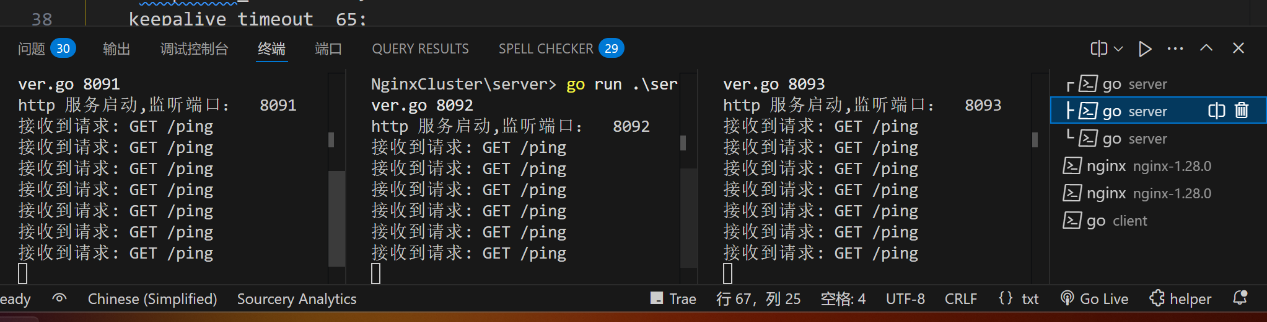




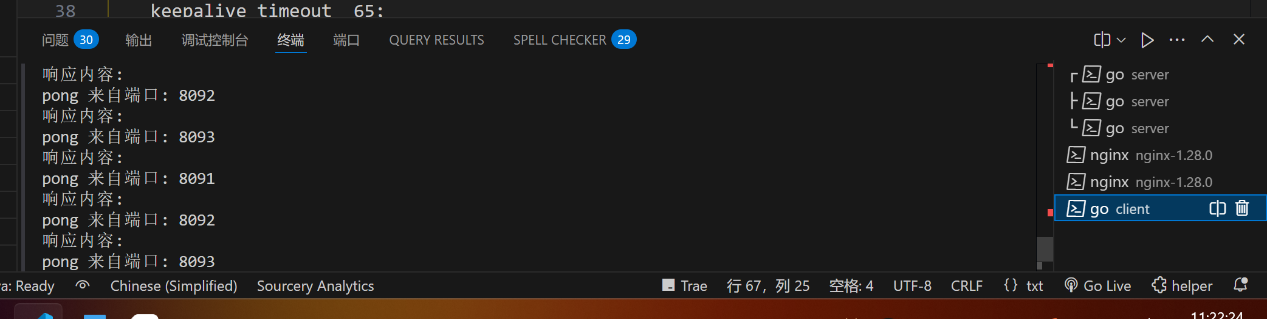
4.检查是否有何缺漏，go mod tidy后确保万无一失，进入nginx目录启动nginx.exe



5.启动三个服务端，分别代表91、92、93端口

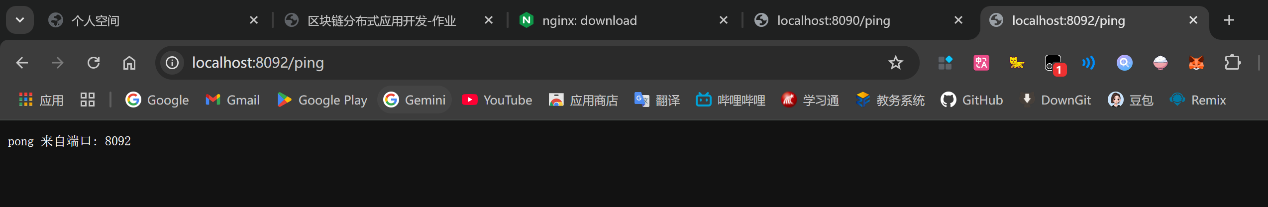


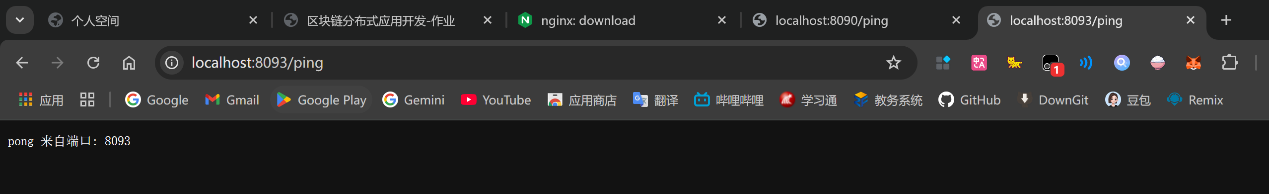
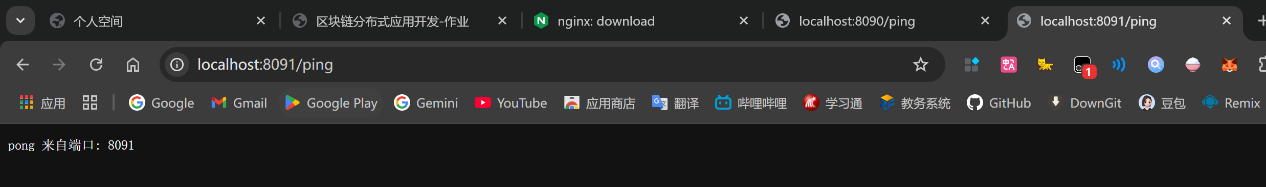
5.启动客户端，用以接收请求



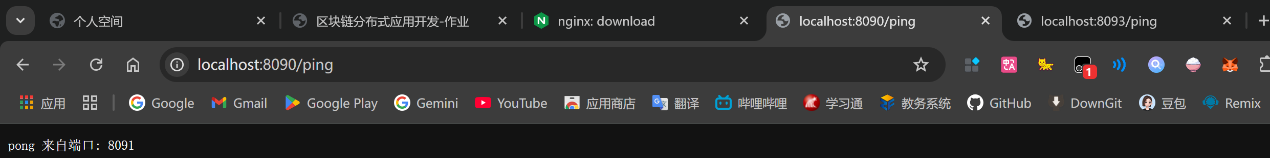
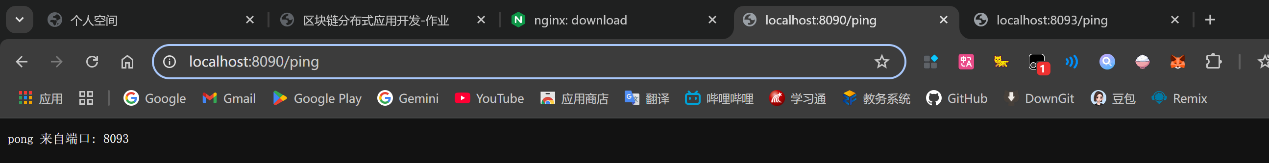
**五、实验结果与分析**

1. 前往浏览器访问各个服务端对应地址（http://localhost:8092/ping）

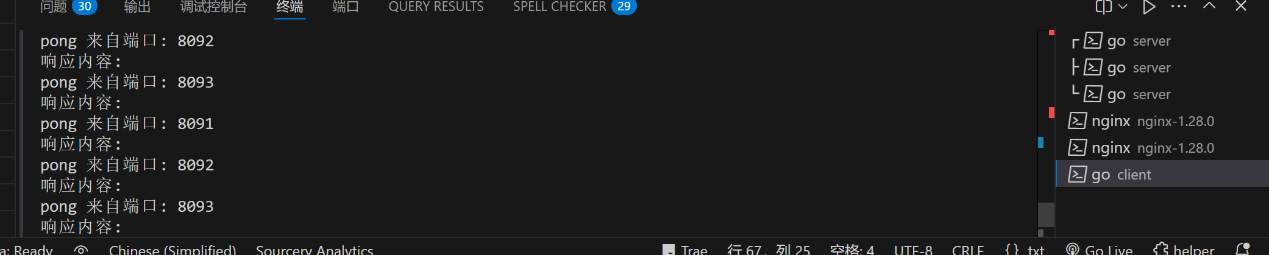




2. 跳转至总集群地址来刷新查看接收的请求（http://localhost:8090/ping）



3. 对应的终端显示



**六、结论与体会**

通过本次 Nginx 负载均衡集群实验，我深入理解了负载均衡在分布式系统中的核心作用。实验中通过配置 Nginx 的upstream模块，成功将请求按轮询策略分发到 8091、8092、8093 三个后端服务端口，验证了流量均衡分配的效果。当手动停止某一端口服务时，Nginx 自动将请求转发至健康节点，体现了其故障转移机制的可靠性。

过程中遇到 Nginx 配置语法错误（如access\_log指令位置不当），通过查阅官方文档和日志定位解决，强化了对 Nginx 配置文件结构的理解。此外，使用 Go 编写多端口模拟服务，结合 Nginx 的健康检查机制，直观感受到负载均衡对系统吞吐量和容错性的提升。此次实验为构建高可用分布式系统奠定了实践基础，未来可进一步探索加权轮询、IP 哈希等算法，以及结合 Docker 容器实现动态负载均衡集群。

**七、教师评语**