**广 州 商 学 院**

**实验报告（第 6 次）**

实验名称 Golang使用Kafka消息队列

实验时间 2025.05.06

同组同学 唐璇 小组分工 暂无

**一、实验目的**

1.理解消息队列的概念：掌握Kafka消息队列的基本原理，理解它在分布式系统中的作用，特别是在解耦、异步处理和高吞吐量数据传输中的应用。

2.掌握Kafka的基本操作：能够使用Go语言与Kafka进行交互，包括生产消息、消费消息、处理消息队列的基本操作。

3.学习如何使用`sarama`库：掌握使用`github.com/IBM/sarama`库与Kafka进行连接、消息发送和接收，处理Kafka的消息传递机制。

4.理解Kafka的高吞吐量和扩展性：通过实验，理解如何利用Kafka的分布式特性实现高吞吐量的消息处理，及其如何支持大规模的消息处理场景。

5.应用Kafka实现系统解耦：通过消息队列实现微服务之间的解耦，提高系统的灵活性、可扩展性和容错性。

**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.Go语言环境：已安装并配置好Go语言环境，包括GOPATH、GOROOT等环境变量的设置。

3.文本编辑器或IDE：如VS Code、GoLand等，用于编写Go代码。

4.Kafka环境：已安装并配置好Kafka集群，可以本地运行Kafka或者使用云服务提供的Kafka实例。

5.Kafka客户端库：使用`github.com/IBM/sarama`库进行Kafka消息队列的操作。

**三、实验原理**

1.Kafka消息队列原理：Kafka是一个分布式的消息队列系统，它使用分区和副本机制来保证消息的高吞吐量、容错性和扩展性。生产者将消息发送到Kafka的主题（Topic），消费者从主题中读取消息。Kafka保证消息的顺序性和可靠性。

2. Kafka分区与副本：Kafka将消息主题分成多个分区，每个分区可以由多个消费者并发处理。每个分区有一个或多个副本，副本之间同步，保证数据的高可用性。

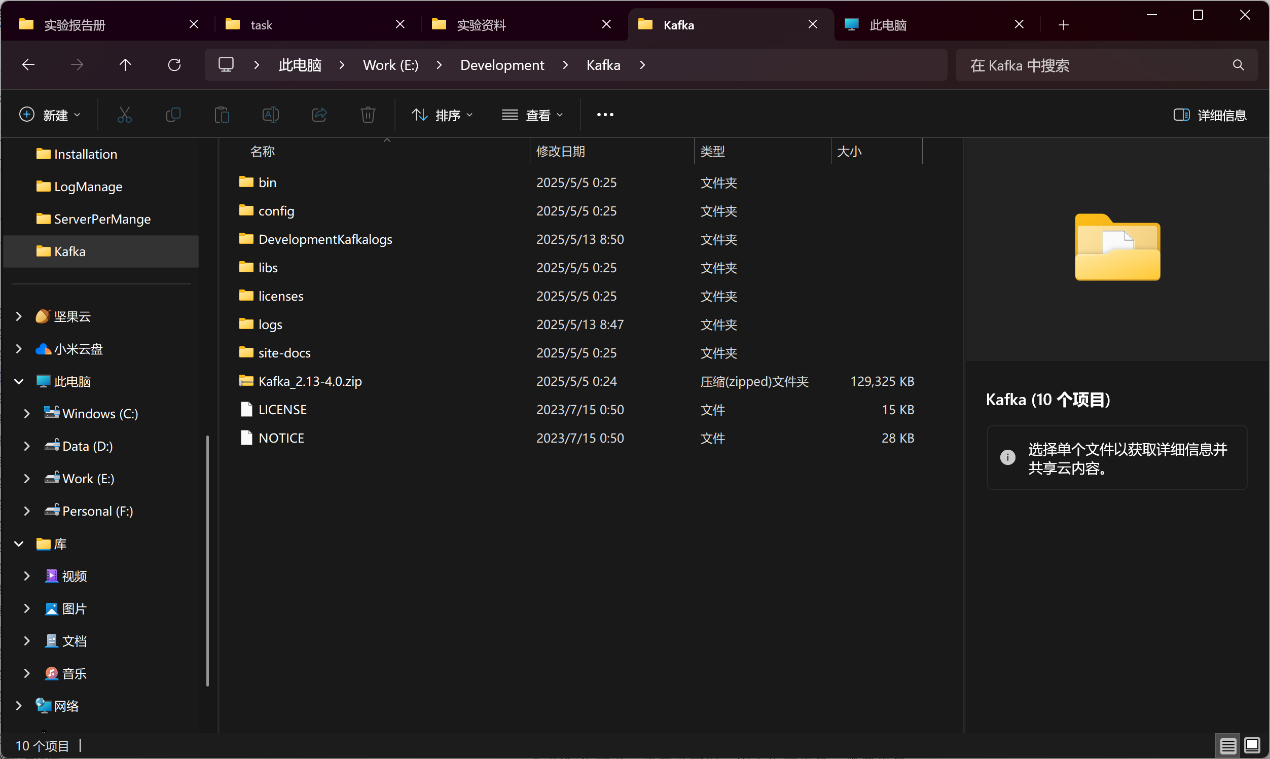
3. sarama库的使用：`sarama`是Go语言中操作Kafka的客户端库，支持Kafka的生产者和消费者功能。它提供了高效的接口来发送、接收消息，并处理Kafka的连接、消息序列化等。

4. 生产者与消费者：生产者将消息发送到Kafka的指定主题，消费者从主题中消费消息。`sarama`提供了同步和异步的生产者接口，消费者支持单个消费者和消费者组。

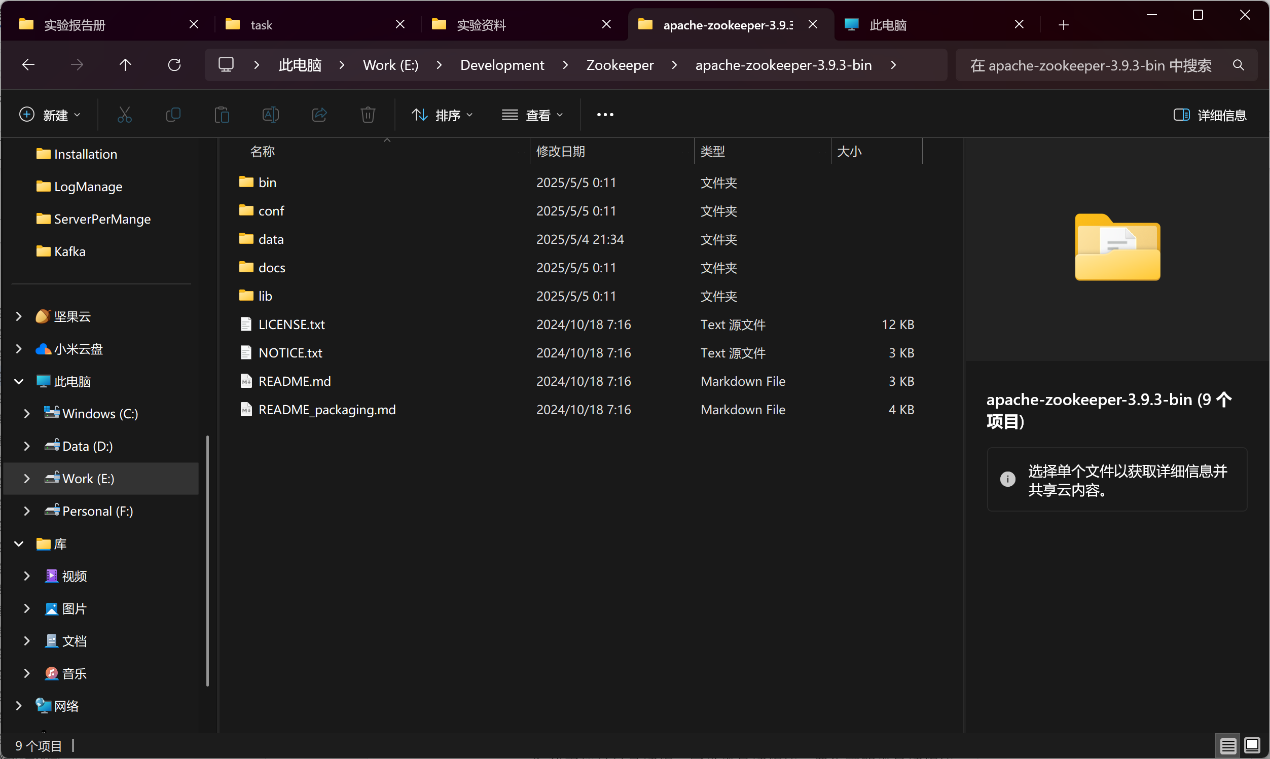
**四、实验内容与步骤**

1.安装并配置Kafka：安装并启动Kafka集群，确保Kafka和Zookeeper正常运行。

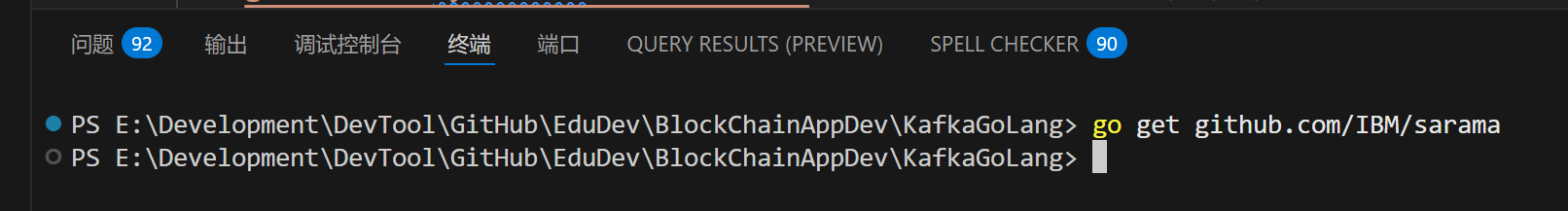
Kafka目录



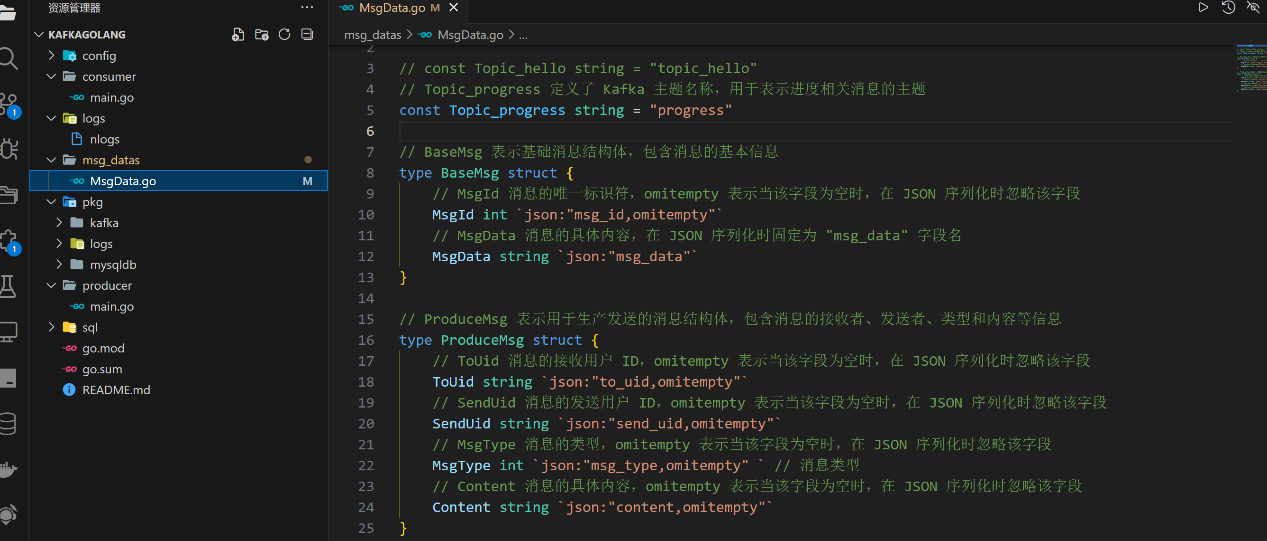
Zookeeper目录



2.安装`sarama`库：使用Go模块或`go get`命令安装`github.com/IBM/sarama`库。

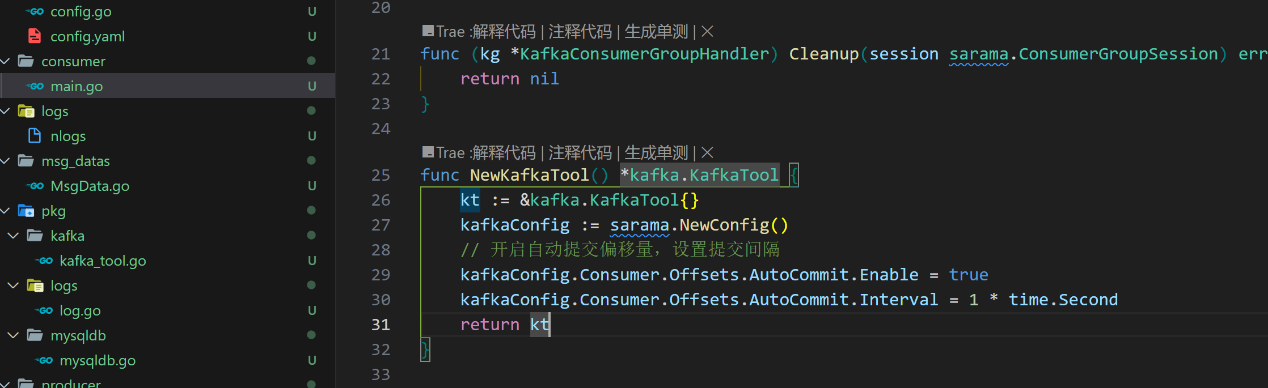


3.建立项目目录结构，定义消息结构体：要生产的消息结构体。



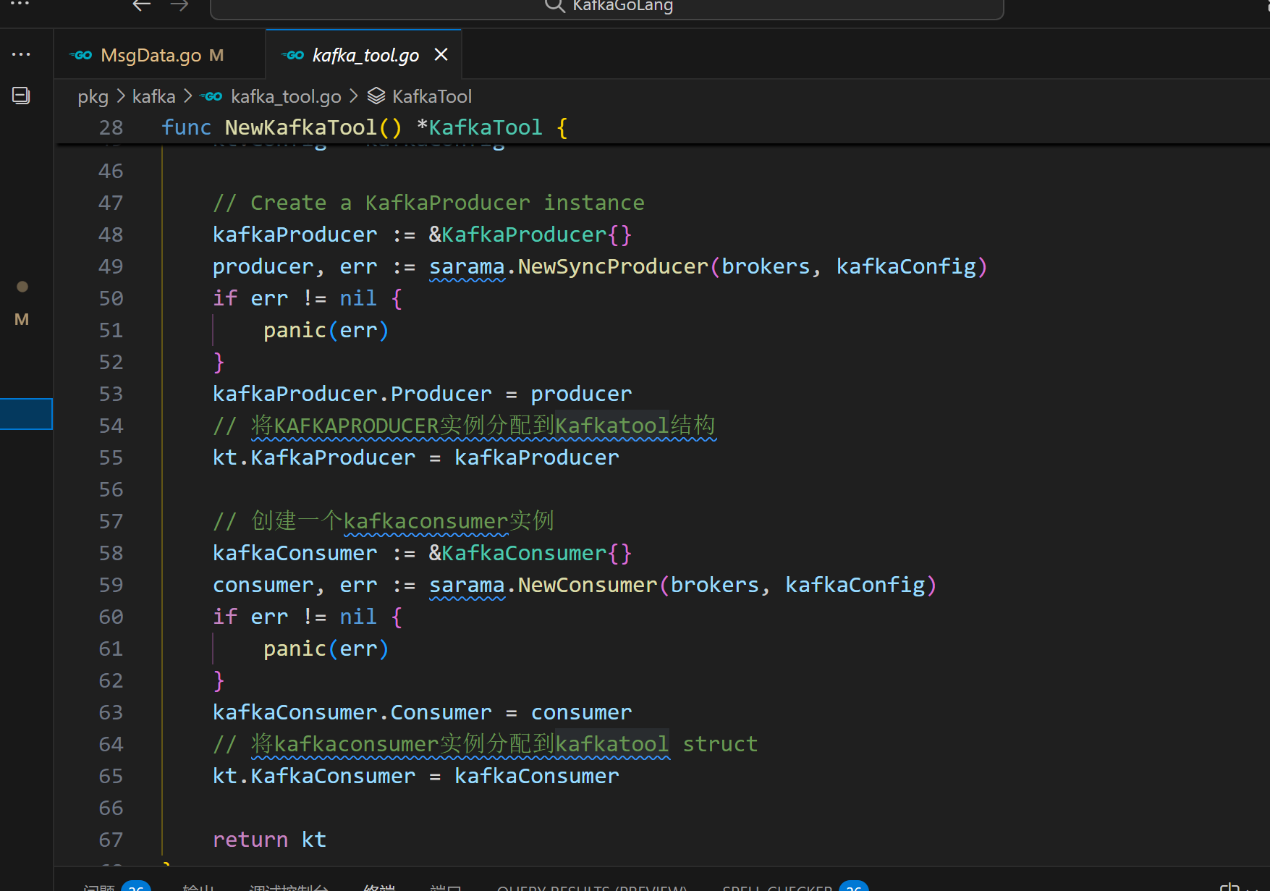
4.封装核心类KafkaTool：KafkaProducer 封装了 sarama.SyncProducer，用于发送消息到 Kafka，KafkaConsumer 封装了 sarama.Consumer 和 sarama.ConsumerGroup，用于从 Kafka 消费消息。

consumer\main.go



定义结构 KafkaTool，包含了 Kafka 集群的 broker 地址、配置以及生产者和消费者实例。并实现相应的功能。

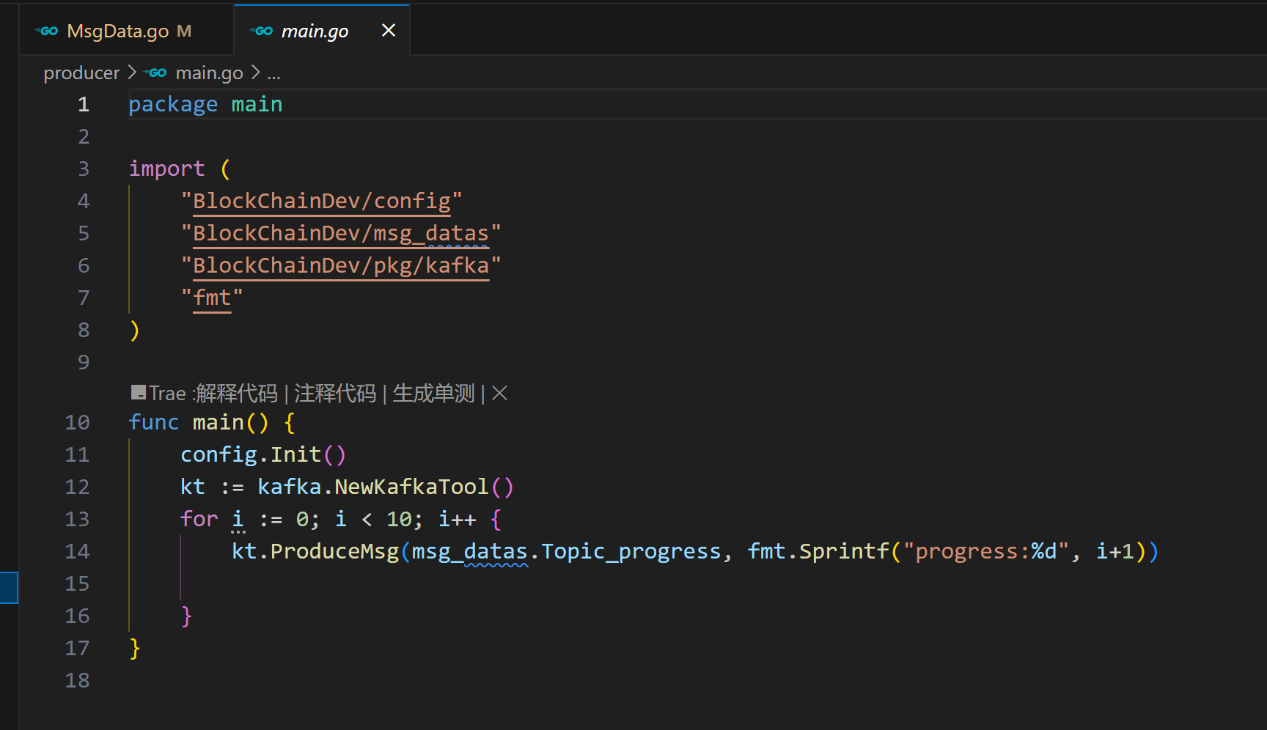
pkg\kafka\kafka\_tool.go



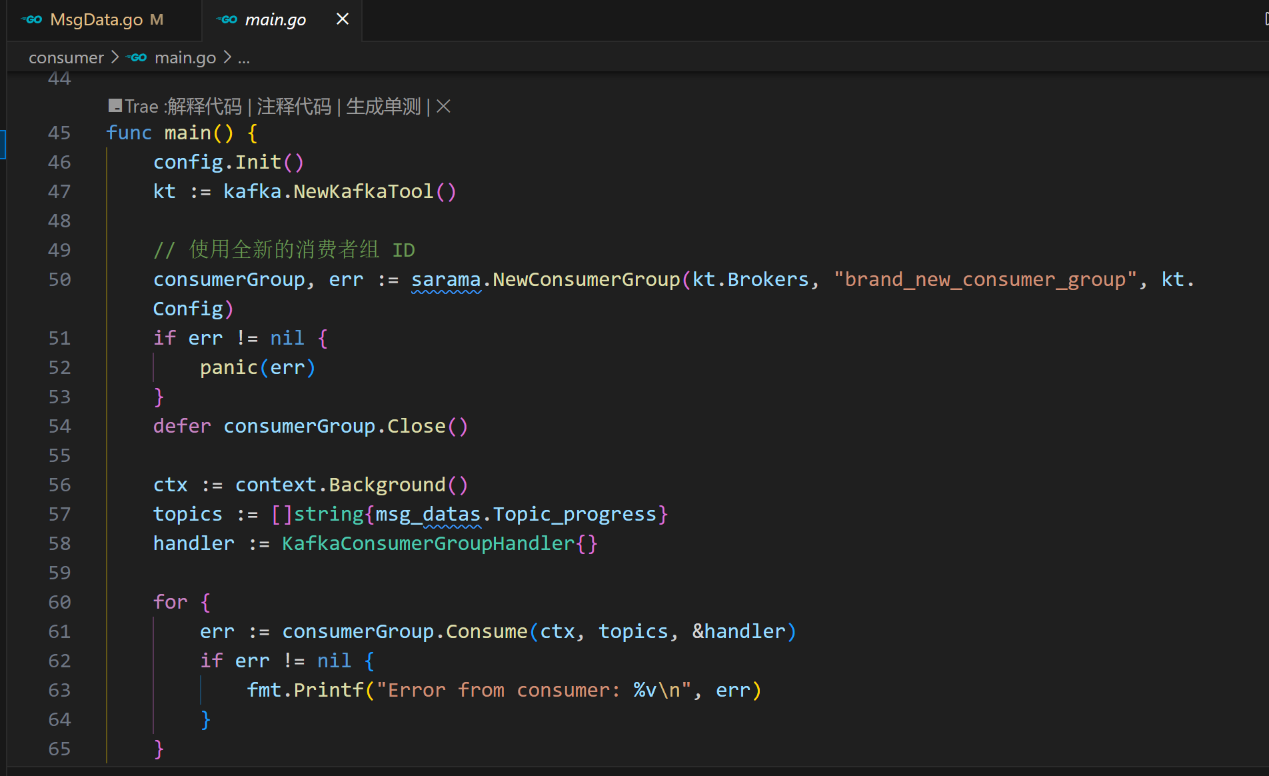
5.编写生产者、消费者实例，用于生产消息、消费消息。

msg\_datas\MsgData.go

生产者



消费者

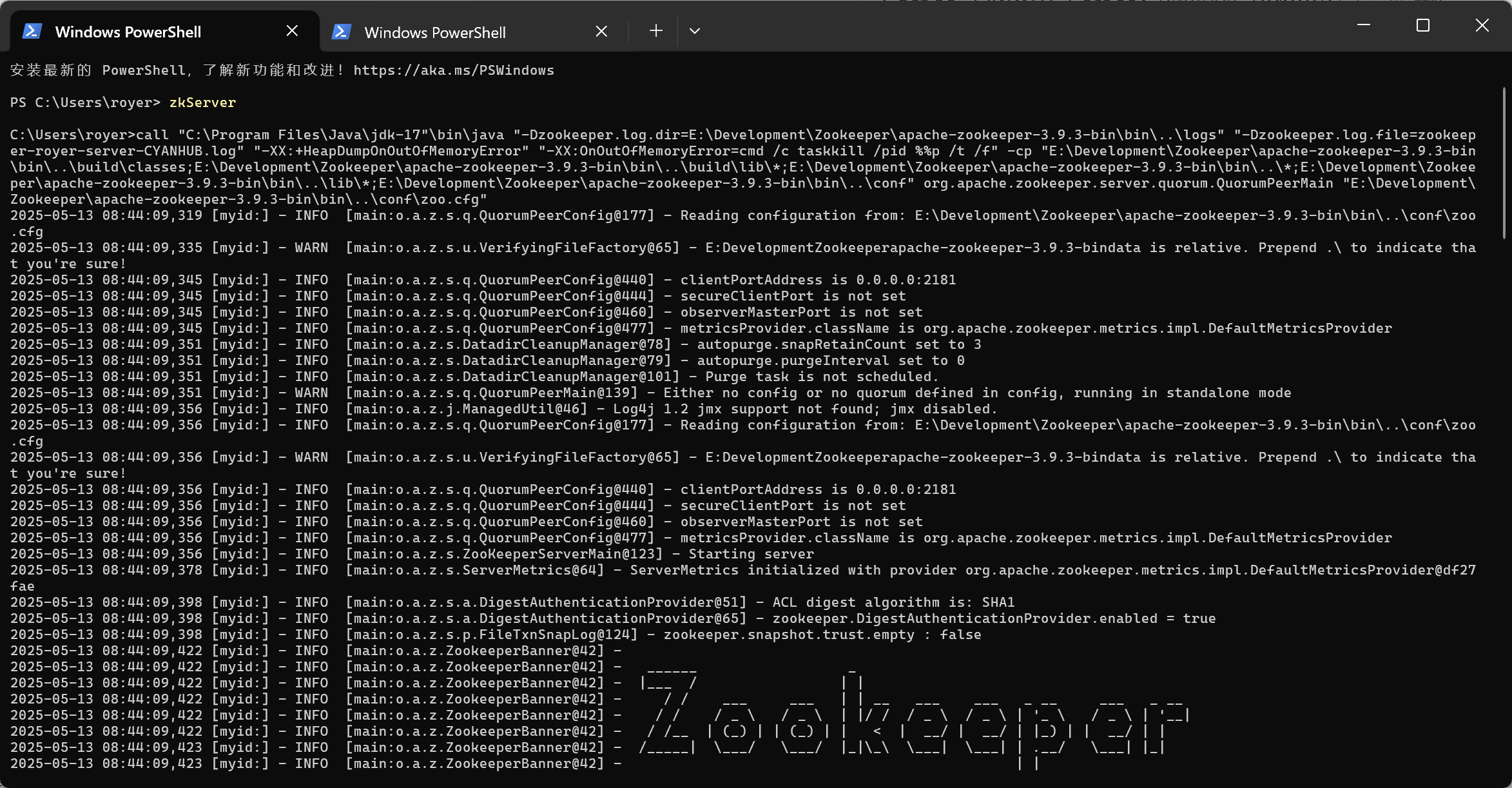


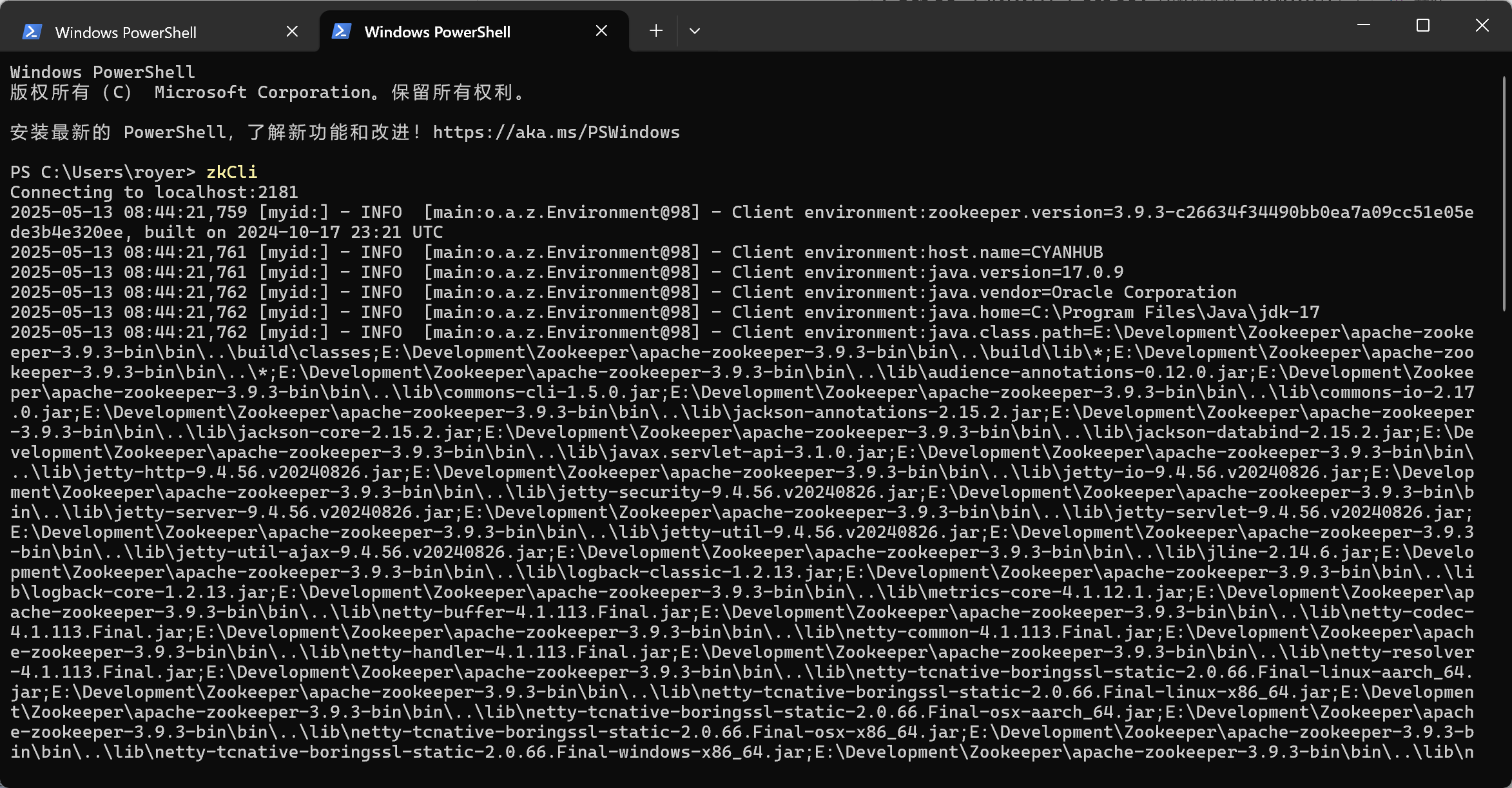
6.最后运行测试。

**五、实验结果与分析**

1.启动Zookeeper和kafka

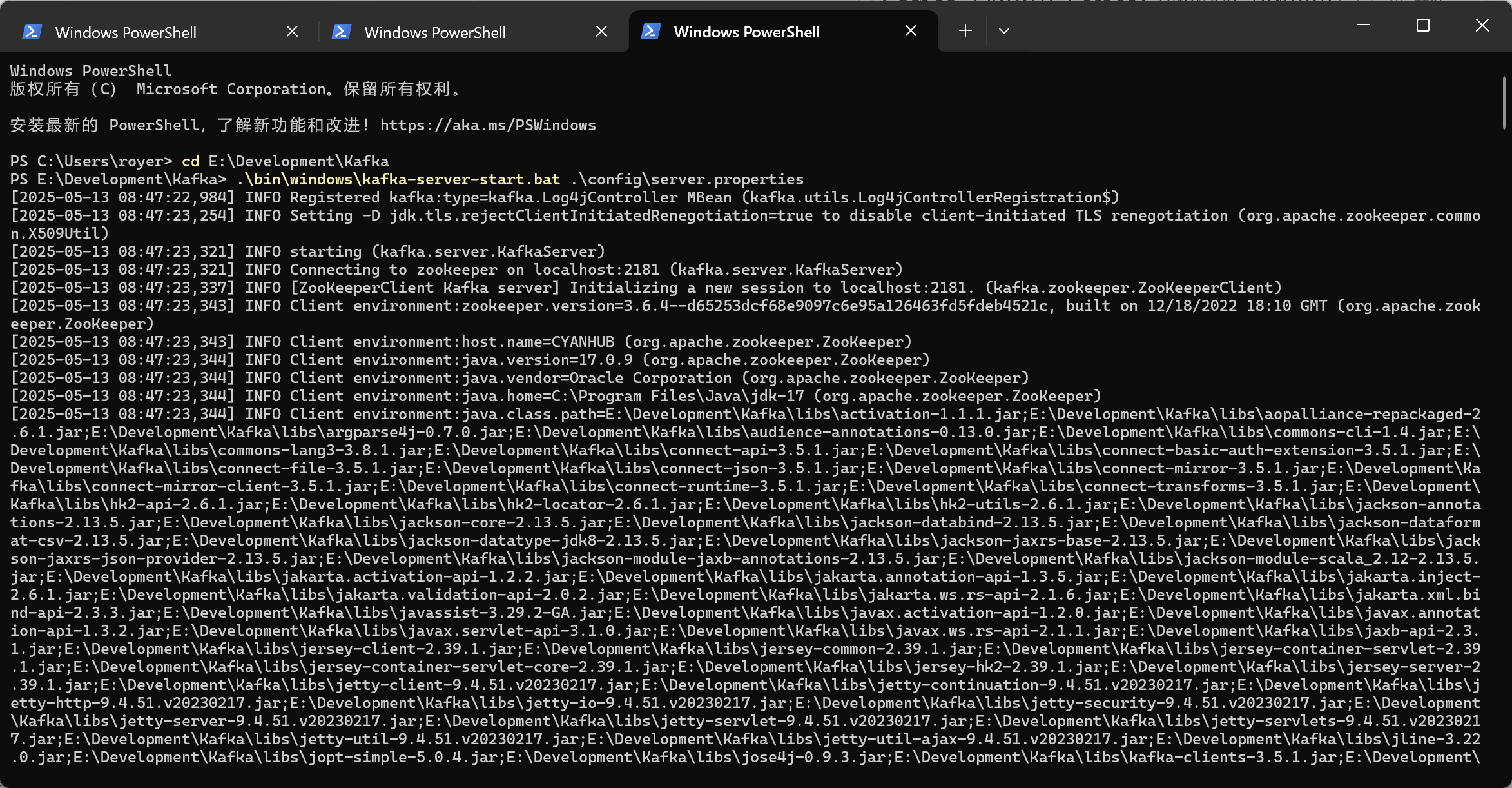
Zookeeper启动（zkServer&zkCli）



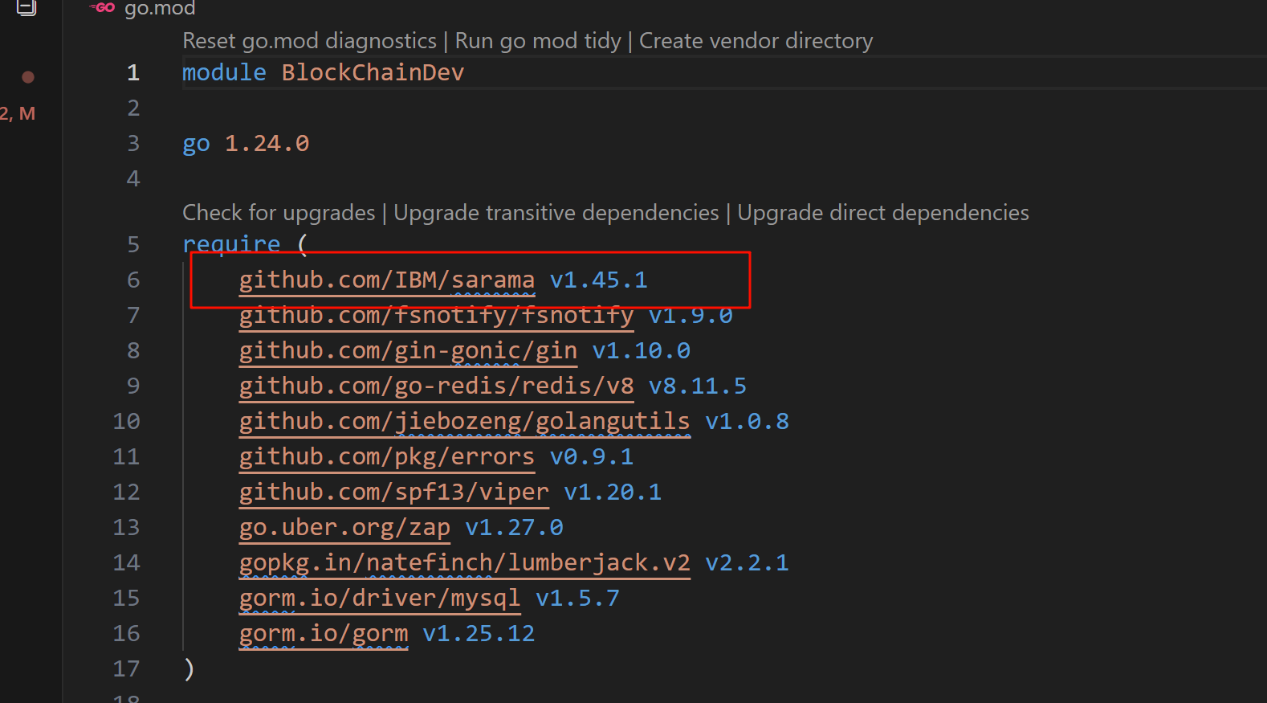


Kafka启动（cd E:\Development\Kafka）

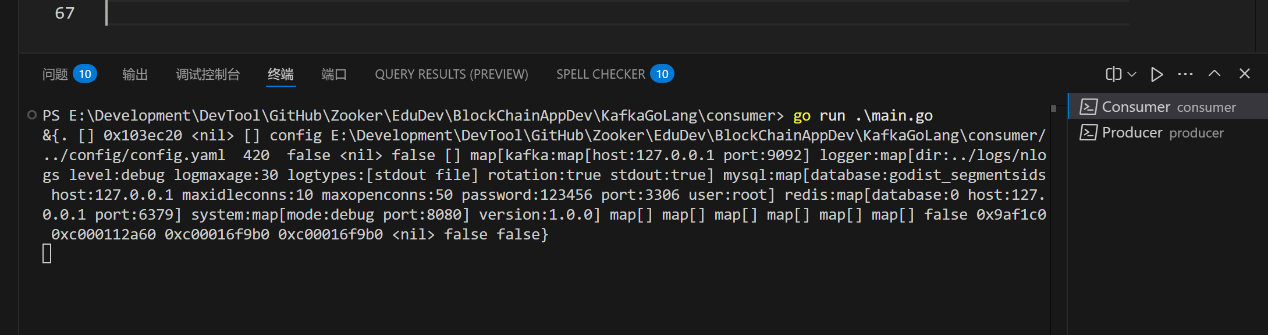
（.\bin\windows\kafka-server-start.bat .\config\server.properties）



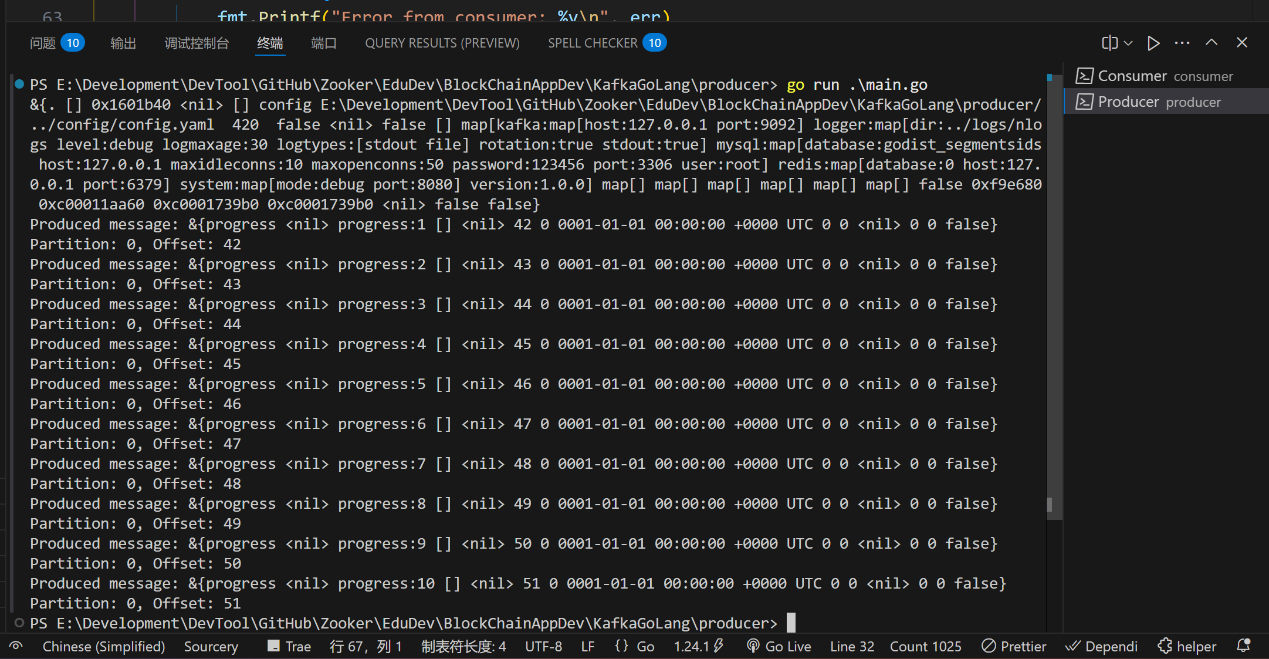
2.获取相关第三方包（go get github.com/IBM/sarama）



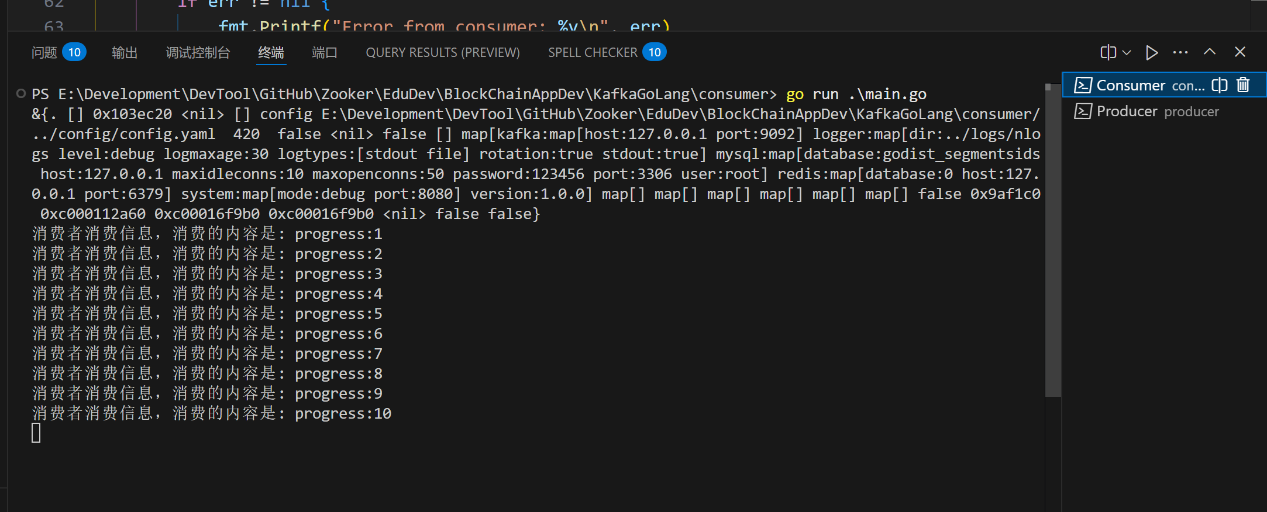
3. 启动Consumer实例进行监听（在**consumer**目录下终端执行**go run .\main.go**）



4. 启动Producer实例进行模拟消费（在**producer**目录下终端执行**go run .\main.go**）



5.返回Consumer终端查看同步情况，确认无误，实验结束



**六、结论与体会**

通过本次实验，我深入理解了Kafka消息队列在分布式系统中的重要性。在实践过程中，我掌握了Kafka与Zookeeper的安装配置，学会运用`sarama`库实现消息的生产与消费。这不仅提升了我的编程技能，还让我体会到消息队列在解耦系统、异步处理方面的强大能力。

同时，在处理实验问题时，我积累了排查和解决问题的经验。此次实验为我后续开发分布式应用奠定了坚实基础，让我更有信心应对复杂系统开发中的技术挑战，我也期待在未来项目中进一步探索Kafka的更多应用场景。

**七、教师评语**