Flume整合Kafka

实验一：Flume整合Kafka（SpoolDir Source 采集文件）

一、实验目的

通过本实验，学生将能够：

1. 理解 **Flume + Kafka** 的整体数据流转过程。
2. 掌握 **SpoolDir Source** 的作用和使用方法。
3. 学会编写 Flume 配置文件，实现：
    - 监听指定目录中的新文件
    - 自动读取文件内容并发送到 Kafka 指定主题（Topic）。
4. 学会使用 Kafka 的 **Topic 创建、消费者验证** 等基本操作。

二、实验背景与原理

1. 实验场景

企业中，常有大量**离线生成的日志文件 / 数据文件**需要导入到大数据平台中做分析。
本实验模拟的场景是：

> 将某目录中新生成的招聘信息文件，自动采集并发送到 Kafka，供下游程序消费分析。

2. 数据流方案：Spool Source + Kafka Sink

整体数据流如下：

> **文件 → Flume（SpoolDir Source）→ Channel → Kafka Sink → Kafka Topic → 消费者**

- **SpoolDir Source**
    - 持续监控一个目录（例如：`/opt/data/flume/logs`）
    - 一旦发现有**新文件放入**目录，就会自动读取文件内容，封装为一条条 Event。
- **Channel**（这里使用 Memory Channel）
    - 类似"临时缓冲区"，负责**暂存** Event。
    - 上游：Source 把数据写入 Channel
    - 下游：Sink 从 Channel 读出数据
- **Kafka Sink**
    - 从 Channel 中取出 Event
    - 把 Event 的 **body 内容作为消息** 写入 Kafka 指定 Topic（如：`mylog`）

Flume + Kafka 实验一

三、实验环境与前置条件

1. 已搭建好的 **三节点集群**（master、slave1、slave2）
2. 已安装并配置好：
    - Zookeeper 集群
    - Kafka 集群
    - Hadoop 集群（本实验用不到 HDFS 写入，但假设环境已在）
    - Flume（安装在 master 节点）
3. 各节点之间可以通过主机名互通（如 `master、slave1、slave2`）。

四、实验步骤

> **建议顺序：先启动底层服务 → 再创建 Topic 和消费者 → 最后启动 Flume 和复制文件。**

步骤 1：启动 Zookeeper 集群

在 **master / slave1 / slave2** 三台机器分别执行：

```
zkServer.sh start
```

> 说明：Zookeeper 为 Kafka 提供协调与管理服务，必须先启动。

步骤 2：启动 Kafka 集群

在 **master / slave1 / slave2** 三台机器分别执行：

```
kafka-server-start.sh -daemon /opt/apps/kafka/config/server.properties
```

> 说明：这里使用 `-daemon` 方式后台启动 Kafka Broker。

步骤 3：启动 Hadoop 集群（如已启动可略过）

在 **master** 启动 HDFS：

```
start-dfs.sh
```

在 **slave1** 启动 YARN：

```
start-yarn.sh
```

> 虽然本实验没有直接往 HDFS 写数据，但完整的大数据实验环境通常需要 Hadoop 正常运行。

Flume + Kafka 实验一

步骤 4：编写 Flume 配置文件（file-to-kafka.conf）

在 **master 节点**上创建 Flume 配置文件：

```
cd /opt/apps/flume/conf

# 创建配置文件
touch file-to-kafka.conf
```

编写以下内容：

```
# 定义 Flume Agent 的组件
# a1 是该 Agent 的名称，用于后续配置中指代此 Agent
a1.sources = r1
# 定义 Sink 名称为 k1
a1.sinks = k1
# 定义 Channel 名称为 c1
a1.channels = c1

# 配置 Source 组件 r1
# Source 是数据的输入端，负责从指定目录中监视和采集文件内容
a1.sources.r1.type = spooldir
# 被监视的目录路径。Flume 会监视该目录，当检测到新文件时，将自动读取文件内容并进行处理。
# 需要确保此目录存在，并且初始为空。
a1.sources.r1.spoolDir = /opt/data/flume/logs

# 配置 Channel 组件 c1
# Channel 是 Flume 中用于连接 Source 和 Sink 的通道，存储传输过程中的数据
# 配置 Channel 通道类型为 memory，用于在内存中缓冲数据
a1.channels.c1.type = memory
# Channel 的容量最多存储 1000 条事件（event）
# 当事件数达到 1000 时，Channel 将暂时停止接收新事件，直到有事件被消费（发送到 Sink），腾出空间后才会继续
接收新事件
a1.channels.c1.capacity = 1000
# Source 和 Sink 从 memory Channel 每次事务传输的最大事件数量为 100
a1.channels.c1.transactionCapacity = 100

# 配置 Sink 组件 k1（输出到 Kafka）
# Sink 是数据的输出端，用于将数据写入 Kafka
a1.sinks.k1.type = org.apache.flume.sink.kafka.KafkaSink
# 设置 Kafka 主题名称，将日志数据写入 Kafka 的 mylog 主题
a1.sinks.k1.topic = mylog
# 设置 Kafka 的 broker 列表，格式为 "主机名:端口号"
a1.sinks.k1.brokerList = master:9092,slave1:9092,slave2:9092
# 每次批量传输 20 条事件到 Kafka，优化性能
a1.sinks.k1.batchSize = 20

# 绑定 Source、Sink 和 Channel 之间的关系
# 将 Source r1 绑定到 Channel c1，数据从 Source 传递到 Channel
a1.sources.r1.channels = c1
# 将 Sink k1 绑定到 Channel c1，数据从 Channel 传递到 Sink
a1.sinks.k1.channel = c1
```

Flume + Kafka 实验一

步骤 5：在 Kafka 中创建 Topic 并查看

在 **master 节点**上创建 `mylog` 主题：

```
# 在master中创建topic【注意不能存在有相同名称的主题】
kafka-topics.sh --create \
--bootstrap-server master:9092,slave1:9092,slave2:9092 \
--topic mylog \
--partitions 1 --replication-factor 3

# 查看topic，确保能看到 mylog
kafka-topics.sh --list \
--bootstrap-server master:9092,slave1:9092,slave2:9092
```

步骤 6：启动 Kafka 消费者（用于验证）

在 **slave1 节点**上开启消费者，实时监听 `mylog` 主题：

```
# 在slave1运行Kafka自带的消费者脚本，监听mylog主题
kafka-console-consumer.sh \
--bootstrap-server master:9092,slave1:9092,slave2:9092 \
--topic mylog
```

⚠️ **不要关闭这个终端窗口**，后面验证数据是否成功发送全靠它来显示。

步骤 7：创建 Flume 监听目录

在 **master 节点**上创建 Flume 监控目录：

```
mkdir -p /opt/data/flume/logs
```

说明：这个目录路径要与配置文件中的 `a1.sources.r1.spoolDir` 一致。

步骤 8：启动 Flume Agent

仍在 **master 节点**上执行：

```
# 在master中启动flume
flume-ng agent \
-n a1 \
-c /opt/apps/flume/conf \
-f /opt/apps/flume/conf/file-to-kafka.conf \
-Dflume.root.logger=INFO,console
```

Flume + Kafka 实验一

注：保留该终端窗口，观察 Flume 日志输出情况，便于出错时排查问题。

步骤 9：准备待传输的日志文件

9.1 在 slave2 上准备源文件

在 **slave2 节点**上操作：

```
mkdir -p /opt/data/flume/logs

# 在 /opt/data/flume/logs 目录中创建 messages 文件
touch /opt/data/flume/logs/messages
```

示例内容（招聘信息，每行一条）：

```
1,玛纳斯分公司（白酒销售）销售员,新疆五谷传奇酒业销售有限公司石...,玛纳斯,3800-10000元/月
2,仪表工,新疆雅澳科技有限责任公司,玛纳斯,5000-7000元/月
3,加液工,石河子市物源盛通商贸有限公司,北泉镇,3500-4500元/月
4,大锅饭师傅,石河子市天石肉类加工有限公司,石河子,3000-3300元/月
5,维修工,石河子市天石肉类加工有限公司,北泉镇,5000-6000元/月
6,车间统计员,石河子市天石肉类加工有限公司,北泉镇,4500-5000元/月
7,保洁,石河子开发区大河宴鱼馆,不限,3200-3500元/月
8,电工,石河子市天石肉类加工有限公司,北泉镇,4500-6000元/月
.....................
.....................
```

9.2 将文件复制到 Flume 监听目录（master）

> 👉 **将日志文件复制到 Flume 的监听目录，触发 SpoolDir Source 自动检测并读取该文件，把内容发送到 Kafka。**

```
# 在slave2上执行：拷贝文件到master节点Flume监听的目录，模拟日志变化
scp /opt/data/flume/logs/messages  master:/opt/data/flume/logs
```

> 这一刻起，Flume 的 `SpoolDir Source` 会检测到 `master:/opt/data/flume/logs` 目录中新出现了一个 **完整文件**，并开始读取其中内容发送到 Kafka。

步骤 10：验证实验结果

1. 回到 **slave1** 上的 Kafka 消费者窗口（刚才一直在监听 `mylog`）。
2. 如果配置正确，你将看到终端中依次打印出 `messages` 文件中的每一行内容，例如：

Flume + Kafka 实验一

```
1,玛纳斯分公司（白酒销售）销售员,新疆五谷传奇酒业销售有限公司石...,玛纳斯,3800-10000元/月
2,仪表工,新疆雅澳科技有限责任公司,玛纳斯,5000-7000元/月
3,加液工,石河子市物源盛通商贸有限公司,北泉镇,3500-4500元/月
4,大锅饭师傅,石河子市天石肉类加工有限公司,石河子,3000-3300元/月
5,维修工,石河子市天石肉类加工有限公司,北泉镇,5000-6000元/月
6,车间统计员,石河子市天石肉类加工有限公司,北泉镇,4500-5000元/月
7,保洁,石河子开发区大河宴鱼馆,不限,3200-3500元/月
8,电工,石河子市天石肉类加工有限公司,北泉镇,4500-6000元/月
.....................
.....................
```

> ✅ 说明：Flume 已成功采集文件，并通过 KafkaSink 写入 `mylog` 主题，消费者顺利接收到了消息。
>
> ⚠️ 提示：Flume 采集完文件后，会把文件重命名为 *.COMPLETED，表示采集完成。

五、重要注意事项（新手必看）

注意点	说明
❗ 文件名不可重复	SpoolDir 默认只处理一次某个文件名，如果同名文件再次放入，将被直接跳过
❗ 不要修改已被监控目录中**正在采集 / 已采集**的文件	可能导致重复发送或采集失败
❗ `spoolDir` 目录下**不能有子目录**	否则 Flume 会报错，无法正常启动
✅ SpoolDir 只适用于“**完整文件**”	不支持持续追加的文件，适合一次性写好的离线文件导入

六、实验小结

Flume 中 **SpoolDir Source** 的配置与使用方法；

如何通过 Flume 将文件内容发送到 Kafka 主题；

Kafka 中 Topic 的创建、查看以及使用 `kafka-console-consumer.sh` 验证数据；

理解了“`文件 → Flume → Kafka → 消费者`”的完整数据链路。

Flume + Kafka 实验一