

## 目录

1. 实验一：SQOOP 数据导入导出 .....	4
1.1. 实验目的 .....	4
1.2. 实验要求 .....	4
1.3. 实验环境 .....	4
1.4. 实验视图 .....	4
1.5. 实验过程 .....	5
1.5.1. 实验任务一：mysql 表中的数据导入到 hdfs.....	5
1.5.1.1. 步骤一：进入 mysql shell.....	5
1.5.1.2. 步骤二：创建数据库 mysql_hdfs .....	5
1.5.1.3. 步骤三：在 mysql_hdfs 上创建表 users .....	5
1.5.1.4. 步骤四：在 users 表中插入数据 .....	5
1.5.1.5. 步骤五：查看表中数据是否存在.....	5
1.5.1.6. 步骤六：退出 mysql.....	6
1.5.1.7. 步骤七：查看 Sqoop 版本 .....	6
1.5.1.8. 步骤八：启动 hadoop，查看进程 .....	6
1.5.1.9. 步骤九：将 mysql 中的数据导入到 hdfs .....	6
1.5.1.10. 步骤十：查看 sqoop 数据是否导入成功.....	6
1.5.2. 实验任务二：hdfs 上的数据导出到 mysql 表中 .....	6
1.5.2.1. 步骤一：进入 mysql shell.....	6
1.5.2.2. 步骤二：在 mysql_hdfs 上创建表：users2.....	7
1.5.2.3. 步骤三：查看 hdfs 中数据.....	7
1.5.2.4. 步骤四：导出数据.....	7
1.5.2.5. 步骤五：查看数据库表中数据.....	7
2. 实验二：Flume 组件常用代理配置 .....	8
2.1. 实验目的 .....	8
2.2. 实验要求 .....	8
2.3. 实验环境 .....	8
2.4. 实验视图 .....	8
2.5. 实验过程 .....	9
2.5.1. 实验任务一：Flume 组件配置 Hadoop sink.....	9
2.5.1.1. 步骤一：启动 hadoop 集群 .....	9
2.5.1.2. 步骤二：创建 agent 代理文件.....	9
2.5.1.3. 步骤三：启动 flume 进程 .....	10
2.5.1.4. 步骤四：向监听端口发送信息.....	10
2.5.1.5. 步骤五：在 hdfs 中查看获取数据信息.....	11

3. 实验三: Kafka 组件部署 .....	12
3.1. 实验目的 .....	12
3.2. 实验要求 .....	12
3.3. 实验环境 .....	12
3.4. 实验视图 .....	12
3.5. 实验过程 .....	12
3.5.1. 实验任务一: Kafka 组件部署 .....	12
3.5.1.1. 步骤一: 将 kafka 的 tar.gz 安装包解压.....	12
3.5.1.2. 步骤二: 将解压好的 kafka 文件重命名 .....	13
3.5.1.3. 步骤三: 修改 server.properties .....	13
3.5.1.4. 步骤四: 使用 scp 命令把 kafka 发送到各个节点 .....	13
3.5.1.5. 步骤五: 修改各个节点中的 server.properties .....	13
3.5.1.6. 步骤六: 启动 ZOOKEEPER 集群 .....	13
3.5.1.7. 步骤七: 在各个节点启动 kafka 服务 .....	14
3.5.2. 实验任务二: Kafka 组件验证部署 .....	14
3.5.2.1. 步骤一: 创建一个名为 hello 的 topic.....	14
3.5.2.2. 步骤二: 查看 topic 是否创建成功.....	14
3.5.2.3. 步骤三: 在 master 节点中创建一个生产者 .....	14
3.5.2.4. 步骤四: 在 slave1 节点中创建一个消费者 .....	14
3.5.2.5. 步骤五: 在创建的生产者中输入信息 .....	15
3.5.2.6. 步骤六: 在创建的消费者中查看信息 .....	15
4. 实验四 Kafka 与 Flume 组合完成数据收集 .....	16
4.1. 实验目的 .....	16
4.2. 实验要求 .....	16
4.3. 实验环境 .....	16
4.4. 实验视图 .....	16
4.5. 实验过程 .....	16
4.5.1. 实验任务一: Flume 和 Kafka 组件整合配置 .....	16
4.5.1.1. 步骤一: 新建一个 syslog_mem_kafka.conf 文件 .....	16
4.5.1.2. 步骤二: 配置 syslog_mem_kafka.conf 文件 .....	16
4.5.1.3. 步骤三: 创建名为 flumekafka 的 topic.....	17
4.5.1.4. 步骤四: 启动 flume 进程 .....	17
4.5.1.5. 步骤五: 在 slave1 中创建消费者 .....	18
4.5.1.6. 步骤六: 使用 nc 命令 向 master: 6868 发送信息.....	18
4.5.1.7. 步骤七: 查看 slave1 中的消费者 .....	18



# 1. 实验一：SQOOP 数据导入导出

## 1.1. 实验目的

完成本实验，您应该能够：

- 掌握 Sqoop 导入数据的方法
- 掌握 Sqoop 导出数据的方法

## 1.2. 实验要求

- 熟悉常用 Linux 操作系统命令
- 熟悉数据库的基本命令
- 熟悉 hdfs 的基本命令

## 1.3. 实验环境

本实验所需之主要资源环境如表 1-1 所示。

表 1-1 资源环境

服务器集群	3 个以上节点, 节点间网络互通, 各节点最低配置: 双核 CPU、8GB 内存、100G 硬盘
运行环境	CentOS 7.4
用户名/密码	root/password hadoop/password
服务和组件	HDFS、Sqoop 1.4.7、mysql5.7.18

## 1.4. 实验视图

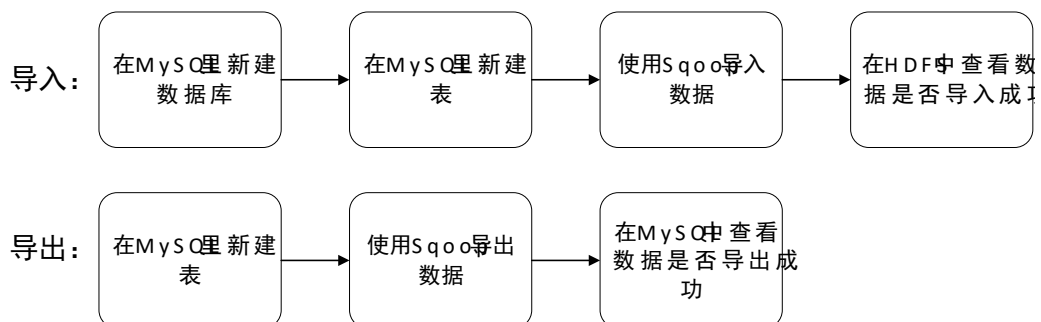


图 1-1 实验流程

## 1.5. 实验过程

### 1.5.1. 实验任务一：mysql 表中的数据导入到 hdfs

#### 1.5.1.1. 步骤一：进入 mysql shell

本实验需要进入 mysql，mysql 的密码是:Password123\$

```
[hadoop@master ~]$ mysql -u root -p
```

输入密码:Password123\$，进入 mysql。

#### 1.5.1.2. 步骤二：创建数据库 mysql\_hdfs

使用 create database 命令创建数据库。

```
mysql> create database mysql_hdfs;
```

使用 show databases 查看数据库是否存在。

```
mysql> show databases;
```

```
+-----+
| Database          |
+-----+
| information_schema |
| hive_db           |
| mysql             |
| mysql_hdfs        |
| performance_schema |
| sys               |
+-----+
6 rows in set (0.15 sec)
```

#### 1.5.1.3. 步骤三：在 mysql\_hdfs 上创建表 users

将默认的数据库指定为 mysql\_hdfs

```
mysql> use mysql_hdfs;
```

创建 users 表指定 id,name,age,sex 字段的类型

```
mysql> create table users(id varchar(11), name varchar(11), age int, sex varchar(11) );
```

#### 1.5.1.4. 步骤四：在 users 表中插入数据

使用 insert into 命令插入数据，每次插入一条数据。

```
mysql> insert into users value('0001', 'jbw', 23, 'man');
```

```
mysql> insert into users value('0010', 'jbw4', 23, 'man');
```

#### 1.5.1.5. 步骤五：查看表中数据是否存在

```
mysql> select * from users;
```

```
+-----+-----+-----+-----+
```

id	name	age	sex
0001	jbw	23	man
0010	jbw4	23	man

#### 1.5.1.6. 步骤六: 退出 mysql

使用 exit 命令退出 mysql。

```
mysql> exit;
```

#### 1.5.1.7. 步骤七: 查看 Sqoop 版本

```
[hadoop@master ~]$ sqoop version
```

```
Sqoop 1.4.7
```

```
Compiled by maugli on Thu Dec 21 15:59:58 STD 2017
```

#### 1.5.1.8. 步骤八: 启动 hadoop, 查看进程

使用 start-all.sh 启动 hadoop 进程。

```
[hadoop@master ~]$ start-all.sh
```

#### 1.5.1.9. 步骤九: 将 mysql 中的数据导入到 hdfs

```
[hadoop@master ~]$ sqoop import --connect jdbc:mysql://master:3306/mysql_hdfs -username root -password Password123$ --table users --columns id,name,age,sex -m 1 --target-dir '/sqoop/users'
```

Import 参数指定 sqoop 为导入数据, connect 参数为数据库连接的地址, username 为数据库的用户名, password 为数据库的密码, table 指定数据库表名称, columns 为表中的字段, m 为启动 MapReduce 的数量, target-dir 为数据导入到 hdfs 的路径

#### 1.5.1.10. 步骤十: 查看 sqoop 数据是否导入成功

```
[hadoop@master ~]$ hdfs dfs -cat /sqoop/users/*
```

```
0010, jbw4, 23, man
```

```
0001, jbw, 23, man
```

### 1.5.2. 实验任务二: hdfs 上的数据导出到 mysql 表中

#### 1.5.2.1. 步骤一: 进入 mysql shell

本实验需要进入 mysql, mysql 的密码是:Password123\$

```
[hadoop@master ~]$ mysql -u root -p
```

输入密码:Password123\$, 进入 mysql。

**1.5.2.2. 步骤二：在 mysql\_hdfs 上创建表：users2**

将默认的数据库指定为 mysql\_hdfs

```
mysql> use mysql_hdfs;
```

创建 users2 表指定 id,name,age,sex 字段的类型

```
mysql> create table users2(id varchar(11), name varchar(11), age int, sex varchar(11) );
```

**1.5.2.3. 步骤三：查看 hdfs 中数据**

```
[hadoop@master ~]$ hdfs dfs -cat /sqoop/users/*
```

```
0010, jbw4, 23, man
```

```
0001, jbw, 23, man
```

**1.5.2.4. 步骤四：导出数据**

```
[hadoop@master ~]$ sqoop export --connect jdbc:mysql://master:3306/mysql_hdfs --username root --password Password123$ --table users2 --export-dir /sqoop/users/part-m-00000 --input-fields-terminated-by ','
```

export 参数指定 sqoop 为导出数据，connect 参数为数据库连接的地址，username 为数据库的用户名，password 为数据库的密码，table 指定数据库表名称，export-dir 为指定要导出数据的路径，input-fields-terminated-by 为指定源数据的分隔符。

**1.5.2.5. 步骤五：查看数据库表中数据**

```
mysql> use mysql_hdfs;
```

```
mysql> select * from users2;
```

```
+-----+-----+-----+-----+
```

```
| id   | name | age | sex |
```

```
+-----+-----+-----+-----+
```

```
| 0001 | jbw  |  23 | man |
```

```
| 0010 | jbw4 |  23 | man |
```

```
+-----+-----+-----+-----+
```

## 2. 实验二：Flume 组件常用代理配置

### 2.1. 实验目的

完成本实验，您应该能够：

- 掌握 Flume 组件代理常用配置的方法
- 掌握 Flume 组件数据获取的方法
- 掌握 Flume 组件管理的方法

### 2.2. 实验要求

- 熟悉常用 Linux 操作系统命令
- 熟悉 Flume 安装部署
- 了解 Flume 的运行机制

### 2.3. 实验环境

本实验所需之主要资源环境如表 1-1 所示。

表 1-1 资源环境

服务器集群	3 个以上节点，节点间网络互通，各节点最低配置：双核 CPU、8GB 内存、100G 硬盘
运行环境	CentOS 7.4
用户名/密码	root/password hadoop/password
服务和组件	Flume1.6.0、Telnet 等，其他服务根据实验需求安装

### 2.4. 实验视图

Flume 实验流程如图 1-1

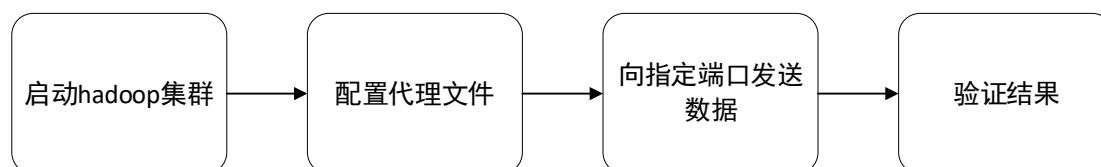


图 1-1 实验流程



## 2.5. 实验过程

### 2.5.1. 实验任务一：Flume 组件配置 Hadoop sink

#### 2.5.1.1. 步骤一：启动 hadoop 集群

```
[hadoop@master ~]$ start-all.sh
```

#### 2.5.1.2. 步骤二：创建 agent 代理文件

在/usr/local/src/flume/conf 目录下创建 hdfs\_sink.conf 文件

```
[hadoop@master ~]$ vim /usr/local/src/flume/conf/hdfs_sink.conf
```

添加如下内容

#指定 sources 的别名

a1.sources = r1

#指定 sinks 的别名

a1.sinks = k1

#指定 channels 的别名

a1.channels = c1

# Describe/configure the source

#指定 sources 的类型

a1.sources.r1.type = syslogtcp

#指定 sources 的端口

a1.sources.r1.port = 5140

#指定 sources 的主机名

a1.sources.r1.host = localhost

# Describe the sink

#指定 sinks 的类型

a1.sinks.k1.type = hdfs

#指定 sinks 的 hdfs 的路径

a1.sinks.k1.hdfs.path = hdfs://master:8020/user/flume/syslogtcp

#指定 sinks 的 hdfs 的文件名前缀

a1.sinks.k1.hdfs.filePrefix = Syslog

#指定时间戳需要四舍五入

a1.sinks.k1.hdfs.round = true

#四舍五入到小于当前时间的最高倍数

a1.sinks.k1.hdfs.roundValue = 10

# 使用本地时间戳

a1.sinks.k1.hdfs.useLocalTimeStamp=true

#四舍五入值的单位，秒、分钟或小时

a1.sinks.k1.hdfs.roundUnit = minute

```
# Use a channel which buffers events in memory
```

```
#指定 channels 的类型
```

```
a1.channels.c1.type = memory
```

```
# Bind the source and sink to the channel
```

```
#将 sources 和 channels 连接
```

```
a1.sources.r1.channels = c1
```

```
#将 sinks 和 channels 连接
```

```
a1.sinks.k1.channel = c1
```

### 2.5.1.3. 步骤三：启动 flume 进程

```
[hadoop@master ~]$ /usr/local/src/flume/bin/flume-ng agent -c /usr/local/src/flume/conf/ -f /usr/local/src/flume/conf/hdfs_sink.conf -n a1 -Dflume.root.logger=DEBUG,console
```

执行后不要关闭终端

-c 的意思是在 conf 目录使用配置文件。指定配置文件放在上面目录

-f 指定一个配置文件

-n agent 的名称（必填）

-D 表示 flume 运行时动态修改 flume.root.logger 参数属性值，并将控制台日志打印级别设置为 DEBUG 级别。日志级别包括:log、info、warn、error。

### 2.5.1.4. 步骤四：向监听端口发送信息

（1）打开一个新的 terminal 终端 ,输入如下命令

```
[hadoop@master ~]$ telnet localhost 5140
```

（2）编辑任意测试内容回车发送，agent 终端的日志显示成功获取数据并创建 hdfs 文件记录。

```
[hadoop@master ~]$ telnet localhost 5140
```

```
Trying ::1...
```

```
telnet: connect to address ::1: Connection refused
```

```
Trying 127.0.0.1...
```

```
Connected to localhost.
```

```
Escape character is '^['.
```

```
hello flume
```

在启动 flume 进程的终端中可以看到以下信息

```
20/06/21 23:46:35 WARN source.SyslogUtils: Event created from Invalid Syslog data.
```

```
20/06/21 23:46:36 INFO hdfs.HDFSSequenceFile: writeFormat = Writable, UseRawLocalFileSystem = false
```

```
20/06/21 23:46:37 INFO hdfs.BucketWriter: Creating
```

```
hdfs://master:9000/user/flume/syslogtcp/Syslog.1592808396936.tmp
```

```
20/06/21 23:47:07 INFO hdfs.BucketWriter: Closing
```

```
hdfs://master:9000/user/flume/syslogtcp/Syslog.1592808396936.tmp
```

```
20/06/21 23:47:07 INFO hdfs.BucketWriter: Renaming
```

```
hdfs://master:9000/user/flume/syslogtcp/Syslog.1592808396936.tmp to
```

```
hdfs://master:9000/user/flume/syslogtcp/Syslog.1592808396936
```

**2.5.1.5. 步骤五: 在 hdfs 中查看获取数据信息**

```
[hadoop@master ~]$ hdfs dfs -ls /user/flume/syslogtcp
```

```
Found 1 items
```

```
-rw-r--r--      1  hadoop  supergroup                127  2020-06-21  23:47  
/user/flume/syslogtcp/Syslog.1592808396936
```

## 3. 实验三：Kafka 组件部署

### 3.1. 实验目的

完成本实验，您应该能够：

- 掌握 Kafka 组件部署的方法
- 掌握 Kafka 组件部署的验证方法

### 3.2. 实验要求

- 熟悉常用 Linux 操作系统命令

### 3.3. 实验环境

本实验所需之主要资源环境如表 1-1 所示。

表 1-1 资源环境

服务器集群	3 个以上节点，节点间网络互通，各节点最低配置：双核 CPU、8GB 内存、100G 硬盘
运行环境	CentOS 7.4
用户名/密码	root/password hadoop/password
服务和组件	Kafka 1.0.0、Zookeeper 3.4.8

### 3.4. 实验视图

Kafka 部署流程：

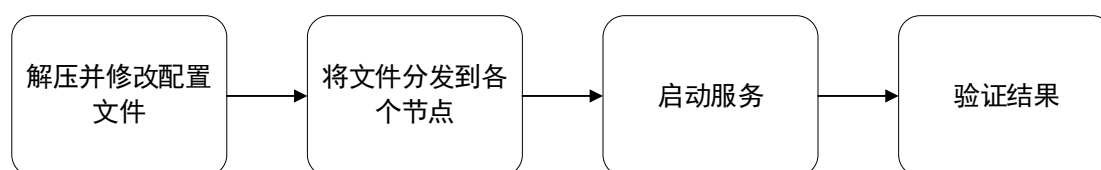


图 1-1 部署流程

### 3.5. 实验过程

#### 3.5.1. 实验任务一：Kafka 组件部署

##### 3.5.1.1. 步骤一：将 kafka 的 tar.gz 安装包解压

将/opt/software/目录下 kafka1.0.0.tgz 的安装包解压到/usr/local/src/

```
[root@master ~]# cd /usr/local/src
```

```
[root@master src]# tar -zxvf /opt/software/kafka1.0.0.tgz -C /usr/local/src/
```

### 3.5.1.2. 步骤二：将解压好的 kafka 文件重命名

```
[root@master src]# mv kafka_2.11-1.0.0/ kafka
```

### 3.5.1.3. 步骤三：修改 server.properties

打开文件/usr/local/src/kafka/config/server.properties

```
[root@master src]# vi /usr/local/src/kafka/config/server.properties
```

在 server.properties 文件下找到下列配置项，并修改为：

broker.id=0

zookeeper.connect=master,slave1,slave2

### 3.5.1.4. 步骤四：使用 scp 命令把 kafka 发送到各个节点

```
[root@master src]# scp -r /usr/local/src/kafka/ root@slave1:/usr/local/src/kafka/
```

```
[root@master src]# scp -r /usr/local/src/kafka/ root@slave2:/usr/local/src/kafka/
```

```
[root@master src]# chown -R hadoop:hadoop /usr/local/src/kafka
```

```
[root@slave1 src]# chown -R hadoop:hadoop /usr/local/src/kafka
```

```
[root@slave2 src]# chown -R hadoop:hadoop /usr/local/src/kafka
```

### 3.5.1.5. 步骤五：修改各个节点中的 server.properties

进入 hadoop 用户,打开文件/usr/local/src/kafka/config/server.properties

```
[root@slave1 src]# su hadoop
```

```
[root@slave2 src]# su hadoop
```

```
[hadoop@slave1 src]$ vi /usr/local/src/kafka/config/server.properties
```

```
[hadoop@slave2 src]$ vi /usr/local/src/kafka/config/server.properties
```

在 server.properties 文件下找到 broker.id 分别修改为

Slave1:

broker.id=1

Slave2:

broker.id=2

### 3.5.1.6. 步骤六：启动 ZOOKEEPER 集群

在 master 节点上启动 zookeeper

```
[hadoop@master src]$ cd
```

```
[hadoop@master ~]$ zkServer.sh start
```

在 slave1 节点上启动 zookeeper

```
[hadoop@slave1 src]$ cd
```

```
[hadoop@slave1 ~]$ zkServer.sh start
```

在 slave2r 节点上启动 zookeeper

```
[hadoop@slave2 src]$ cd
```

```
[hadoop@slave2 ~]$ zkServer.sh start
```

### 3.5.1.7. 步骤七：在各个节点启动 kafka 服务

每个节点服务启动后终端不要关闭

在 master 节点上启动 kafka 服务

```
[hadoop@master ~]$ /usr/local/src/kafka/bin/kafka-server-start.sh  
/usr/local/src/kafka/config/server.properties
```

在 slave1 节点上启动 kafka 服务

```
[hadoop@slave1 ~]$ /usr/local/src/kafka/bin/kafka-server-start.sh  
/usr/local/src/kafka/config/server.properties
```

在 slave2 节点上启动 kafka 服务

```
[hadoop@ slave2 ~]$ /usr/local/src/kafka/bin/kafka-server-start.sh  
/usr/local/src/kafka/config/server.properties
```

## 3.5.2. 实验任务二：Kafka 组件验证部署

### 3.5.2.1. 步骤一：创建一个名为 hello 的 topic

在 master 打开一个新终端

```
[hadoop@master ~]$ /usr/local/src/kafka/bin/kafka-topics.sh --create --zookeeper  
master:2181,slave1:2181,slave2:2181 --replication-factor 2 --topic hello --partitions 1
```

Create 参数代表创建, zookeeper 参数为 zookeeper 集群的主机名, replication-factor 代表生成多少个副本文件, topic 为 topic 的名称, partitions 指定多少个分区

### 3.5.2.2. 步骤二：查看 topic 是否创建成功

```
[hadoop@master ~]$ /usr/local/src/kafka/bin/kafka-topics.sh --list --zookeeper  
master:2181,slave1:2181,slave2:2181
```

### 3.5.2.3. 步骤三：在 master 节点中创建一个生产者

使用 kafka-console-producer.sh 脚本来创建生产者

```
[hadoop@master ~]$ /usr/local/src/kafka/bin/kafka-console-producer.sh --broker-list  
master:9092,slave1:9092,slave2:9092 --topic hello
```

执行后终端不要关闭

broker-list 指定服务器, 在 Kafka 集群包含一个或多个服务器, 这种服务器被称为 broker。

Topic 指定在 hello 上创建生产者。

### 3.5.2.4. 步骤四：在 slave1 节点中创建一个消费者

在 slave1 中打开一个新终端, 使用 kafka-console-consumer.sh 脚本来创建消费者

```
[hadoop@slave1 ~]$ /usr/local/src/kafka/bin/kafka-console-consumer.sh --zookeeper
master:2181,slave1:2181,slave2:2181 --topic hello --from-beginning
```

执行后终端不要关闭

zookeeper 参数为 zookeeper 集群的主机名

Topic 指定在 hello 上创建消费者

from-beginning 读取历史未消费的数据

#### 3.5.2.5. 步骤五：在创建的生产者中输入信息

在创建的生产者的终端中输出信息

```
>hello kafka
```

#### 3.5.2.6. 步骤六：在创建的消费者中查看信息

接受成功，则 kafka 组件验证成功

在创建消费者的终端中可以看到以下输出信息

```
Using the ConsoleConsumer with old consumer is deprecated and will be removed in a future
major release. Consider using the new consumer by passing [bootstrap-server] instead of
[zookeeper].
```

```
hello kafka
```

## 4. 实验四 Kafka 与 Flume 组合完成数据收集

### 4.1. 实验目的

完成本实验，您应该能够：

- 掌握 Kafka 组件与 flume 组件连用的方法

### 4.2. 实验要求

- 熟悉常用 Linux 操作系统命令
- 熟悉 flume 基本操作
- 熟悉 kafka 的基本操作

### 4.3. 实验环境

服务器集群	3 个以上节点，节点间网络互通，各节点最低配置：双核 CPU、8GB 内存、100G 硬盘
运行环境	CentOS 7.4
用户名/密码	root/password hadoop/password
服务和组件	HDFS , Flume 1.6.0, Kafka 1.0.0

### 4.4. 实验视图

Kafka 与 Flume 组合数据收集流程：

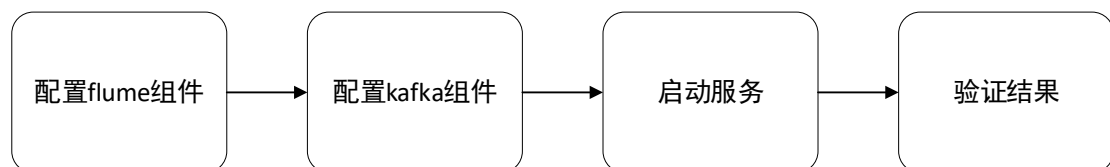


图 1-1 实验流程

### 4.5. 实验过程

#### 4.5.1. 实验任务一：Flume 和 Kafka 组件整合配置

##### 4.5.1.1. 步骤一：新建一个 syslog\_mem\_kafka.conf 文件

在/usr/local/src/flume/conf /目录下创建 syslog\_mem\_kafka.conf 文件

```
[hadoop@master ~]$ vim /usr/local/src/flume/conf/syslog_mem_kafka.conf
```

##### 4.5.1.2. 步骤二：配置 syslog\_mem\_kafka.conf 文件

添加如下内容：



```

#指定 source 的别名为 src
agent1.sources = src
#指定 channels 的别名为 ch1
agent1.channels = ch1
#指定 sinks 的别名为 des1
agent1.sinks = des1

# Describe/configure the source
#指定 sources 的类型
agent1.sources.src.type = syslogtcp
#指定 sources 的端口
agent1.sources.src.port = 6868
#指定 sources 的主机名
agent1.sources.src.host = master

# Use a channel which buffers events in memory
#指定 channels 的类型
agent1.channels.ch1.type = memory

# Describe the sink
#指定 sinks 的类型
agent1.sinks.des1.type = org.apache.flume.sink.kafka.KafkaSink
#指定 kafka 中多个服务器名称
agent1.sinks.des1.brokerList = master:9092,slave1:9092,slave2:9092
#指定 kafka 中 topic 名称
agent1.sinks.des1.topic = flumekafka
#将 sources 和 channels 连接
agent1.sources.src.channels = ch1
#将 sinks 和 channels 连接
agent1.sinks.des1.channel = ch1

```

#### 4.5.1.3. 步骤三：创建名为 flumekafka 的 topic

```
[hadoop@master ~]$ /usr/local/src/kafka/bin/kafka-topics.sh --create --zookeeper
master:2181,slave1:2181,slave2:2181 --replication-factor 2 --topic flumekafka --partitions 1
```

Create 参数代表创建, zookeeper 参数为 zookeeper 集群的主机名， replication-factor 代表生成多少个副本文件， topic 为 topic 的名称， partitions 指定多少个分区

#### 4.5.1.4. 步骤四：启动 flume 进程

```
[hadoop@master ~]$ /usr/local/src/flume/bin/flume-ng agent -c /usr/local/src/flume/conf/ -f
/usr/local/src/flume/conf/syslog_mem_kafka.conf -n agent1 -
Dflume.root.logger=DEBUG,console
```

运行后不要关闭终端

-c 的意思是在 conf 目录使用配置文件。指定配置文件放在上面目录

-f 指定一个配置文件  
-n agent 的名称（必填）  
-D 表示 flume 运行时动态修改 flume.root.logger 参数属性值，并将控制台日志打印级别设置为 INFO 级别。日志级别包括:log、info、warn、error。

#### 4.5.1.5. 步骤五：在 slave1 中创建消费者

```
[hadoop@slave1 ~]$ /usr/local/src/kafka/bin/kafka-console-consumer.sh --zookeeper master:2181,slave1:2181,slave2:2181 --topic flumekafka --from-beginning
```

运行后不要关闭终端

zookeeper 参数为 zookeeper 集群的主机名

Topic 指定在 hello 上创建消费者

from-beginning 读取历史未消费的数据

#### 4.5.1.6. 步骤六：使用 nc 命令 向 master: 6868 发送信息

如果连接成功，这时候客户端输入文本信息回车就可以发送到服务端，一旦有人连接，第二个会话就连接不上。

打开新的 master 终端

```
[hadoop@master ~]$ nc master 6868
```

Hello flumekafka

#### 4.5.1.7. 步骤七：查看 slave1 中的消费者

在前面 slave1 创建消费者的终端中可以看到输出信息

```
Using the ConsoleConsumer with old consumer is deprecated and will be removed in a future major release. Consider using the new consumer by passing [bootstrap-server] instead of [zookeeper].
```

hello flumekafka