Kafka15 flume连接Kafka

> flume是日志采集，为什么对接kafka呢?
>
> 可以动态添加。
>
> 可以做到采集日志文件给多个人用

- 配置 flume(flume-kafka.conf)

```
# define
a1.sources = r1
a1.sinks = k1
a1.channels = c1

# source
a1.sources.r1.type = exec
a1.sources.r1.command = tail -F -c +0 /opt/module/datas/flume.log
a1.sources.r1.shell = /bin/bash -c

# sink
a1.sinks.k1.type = org.apache.flume.sink.kafka.KafkaSink
a1.sinks.k1.kafka.bootstrap.servers =
hadoop102:9092,hadoop103:9092,hadoop104:9092
a1.sinks.k1.kafka.topic = first
a1.sinks.k1.kafka.flumeBatchSize = 20
a1.sinks.k1.kafka.producer.acks = 1
a1.sinks.k1.kafka.producer.linger.ms = 1

# channel
a1.channels.c1.type = memory
a1.channels.c1.capacity = 1000
a1.channels.c1.transactionCapacity = 100

# bind
a1.sources.r1.channels = c1
a1.sinks.k1.channel = c1
```

2） 启动 kafkaIDEA 消费者

3） 进入 flume 根目录下，启动 flume

```
$ bin/flume-ng agent -c conf/ -n a1 -f jobs/flume-kafka.conf
```

4） 向 /opt/module/datas/flume.log 里追加数据，查看 kafka 消费者消费情况

```
$ echo hello > /opt/module/datas/flume.log
```

分类

代码

```
package com.atguigu.interceptor;

import org.apache.flume.interceptor.Interceptor;

public class TypeInterceptor implements Interceptor{

    //声明一个存放事件得集合
    private List<Event> addHeaderEvents;

    @Override
    public void initialize(){

        //初始化
        addHeaderEvents = new ArrayList<>();

    }

    //单个事件拦截
    @Override
    public Event intercept(Even event){

        //1. 获取header
        Map<String, String> headers = event.getHeaders();

        //2. 获取事件中得body信息
        String body = new String(event.getBody());

        //3. 根据body中是否有“hello”来决定添加怎样得头信息
        if(body.contains("hello")){
            //4. 添加头信息
            headers.put("topic", "first");
        }else{
            headers.put("topic", "second");
        }

        return event;
    }

    //批量事件拦截
    @Override
    public List<Event> intercept(List<Event> events){

        //1. 清空集合
        addHeaderEvents.clear();

        //2. 遍历events
        for(Event event : events){
            //3.给每个事件添加头信息
             addHeaderEvents.add(intercept(event));
        }

        return addHeaderEvents;
    }

    @Override
    public void close(){
```

```
    }

    public static class Builder implements Interceptor.Builder{

        @Override
        public Inteceptor build(){

            return new TypeInterceptor();
        }

        @Override
        public void configure(Context context){

        }

    }

}
```

- 打包上传 flume/lib 下
- 配置flume

```
# define
a1.sources = r1
a1.sinks = k1
a1.channels = c1

# source
a1.sources.r1.type = exec
a1.sources.r1.command = tail -F -c +0 /opt/module/datas/flume.log
a1.sources.r1.shell = /bin/bash -c

#Inteceptor
a1.sources.r1.interceptors = i1
a1.sources.r1.interceptors.i1.type =
com.atguigu.interceptor.TypeInterceptor$Builder

# sink
a1.sinks.k1.type = org.apache.flume.sink.kafka.KafkaSink
a1.sinks.k1.kafka.bootstrap.servers =
hadoop102:9092,hadoop103:9092,hadoop104:9092
a1.sinks.k1.kafka.topic = first
a1.sinks.k1.kafka.flumeBatchSize = 20
a1.sinks.k1.kafka.producer.acks = 1
a1.sinks.k1.kafka.producer.linger.ms = 1

# channel
a1.channels.c1.type = memory
a1.channels.c1.capacity = 1000
a1.channels.c1.transactionCapacity = 100

# bind
a1.sources.r1.channels = c1
a1.sinks.k1.channel = c1
```

- 启动两消费者

```
bin/kafka-console-consumer.sh --zookeeper hadoop102:2181 --topic first

bin/kafka-console-consumer.sh --zookeeper hadoop102:2181 --topic second
```

- 启动flume

```
bin/flume-ng agent -c conf/ -f job/type_kafka.conf -n a1
```

- 发送

```
nc localhost 44444
hello
lwlw
```