Flume14 自定义拦截器

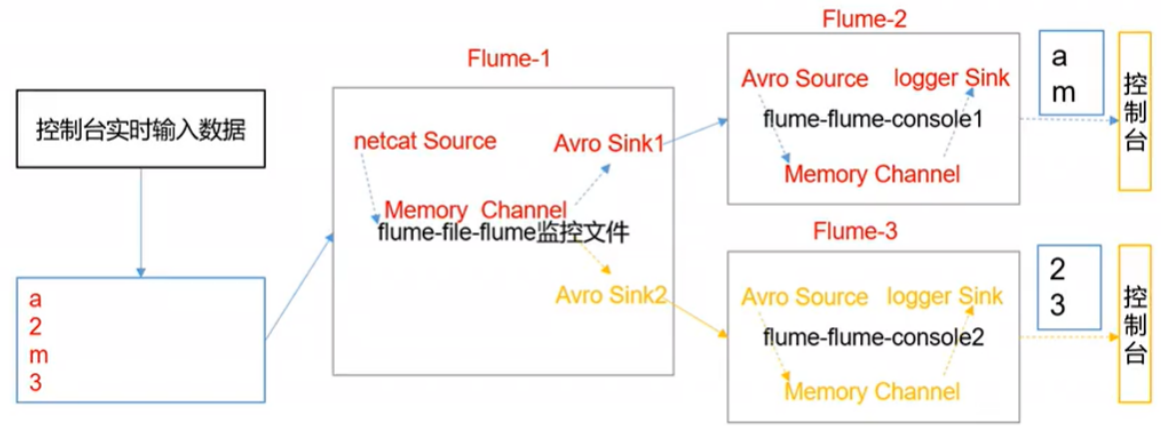
案例需求

使用 Flume 采集服务器本地日志，需要按照日志类型的不同，将不同种类的日志发往不同的分析系统。

需求分析

在实际开发中，一台服务器产生的日志类型可能有很多种，不同类型的日志可能需要发送到不同的分析系统。此时会用到Flume拓扑结构中的Multiplexing结构，Multiplexing的原理是，根据event中Header的某个key值，将不同的event发送到不同的Channel中，所以我们需要自定义一个Interceptor，为不同类型的event的Header中的key赋予不同的值。

在该案例中，我们以端口数据模拟日志，以数字（单个）和字母（单个）模拟不同类型的日志，我们需要自定义interceptor区分数字和字母，将其分别发往不同的分析系统（Channel）。



实验步骤

- 自定义拦截器

  新创建一个 demo， (flume-demo)

  添加依赖

```
<dependencies>
    <dependency>
        <groupId>org.apache.flume</groupId>
        <artifactId>flume-ng-core</artifactId>
        <version>1.7.0</version>
    </dependency>
</dependencies>
```

  打开 src.java 新建一个Class

```
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("type", "atguigu");
        }else{
            headers.put("type", "bigdata");
        }

        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，然后传到 /opt/module/flume/lib 下

- 写配置文件

  写一个interceptor文件

```
mkdir interceptor
cd interceptor
```

  配置 flume2.conf：

```
#Name
a2.sources = r1
a2.sinks = k1 k2
a2.channels = c2 c1

#Source
a2.sources.r1.type = netcat
a2.sources.r1.bind = localhost
a2.sources.r1.port = 44444

#Inteceptor
a2.sources.r1.interceptors = i1
a1.sources.r1.interceptors.i1.type =
com.atguigu.interceptor.TypeInterceptor$Builder

#channel selector
a1.sources.r1.selector.type = multiplexing
a1.sources.r1.selector.header = type
a1.sources.r1.selector.mapping.atguigu = c1
a1.sources.r1.selector.mapping.bigdata = c2

#Channel
a2.channels.c1.type = memory
a2.channels.c1.capacity = 1000
a2.channels.c1.transactionCapacity = 100

a2.channels.c2.type = memory
a2.channels.c2.capacity = 1000
a2.channels.c2.transactionCapacity = 100

#Sink
```

```
a2.sinks.k1.type = avro
a2.sinks.k1.hostname = hadoop103
a2.sinks.k1.port = 4142
a2.sinks.k2.type = avro
a2.sinks.k2.hostname = hadoop104
a2.sinks.k2.port = 4142

#Bind
a2.sources.r1.channels = c1 c2
a2.sinks.k1.channel = c1
a2.sinks.k2.channel = c2
```

配置flume3.conf：

```
#Name the components on this agent
a3.sources = r1
a3.sinks = k1
a3.channels = c2

# Describe/configure the source
a3.sources.r1.type = avro
a3.sources.r1.bind = hadoop103
a3.sources.r1.port = 4142

# Describe the sink
a3.sinks.k1.type = logger

# Describe the channel
a3.channels.c2.type = memory
a3.channels.c2.capacity = 1000
a3.channels.c2.transactionCapacity = 100

# Bind the source and sink to the channel
a3.sources.r1.channels = c2
a3.sinks.k1.channel = c2
```

配置flume4.conf：

```
# Name the components on this agent
a4.sources = r1
a4.sinks = k1
a4.channels = c1

# Describe/configure the source
a4.sources.r1.type = avro
a4.sources.r1.bind = hadoop104
a4.sources.r1.port = 4141

# Describe the sink
a4.sinks.k1.type = logger

# Describe the channel
a4.channels.c1.type = memory
a4.channels.c1.capacity = 1000
a4.channels.c1.transactionCapacity = 100
```

```
# Bind the source and sink to the channel
a4.sources.r1.channels = c1
a4.sinks.k1.channel = c1
```