FLUME 数据采集框架、

# 一、概念

|  |
| --- |
| 1、日志收集框架  主要有三个流程：  1）读数据 ， 会有各种数据源，通过自己实现各种数据源的source类，读取  2）缓存数据， 缓存到内在、本地文件  3）写数据 有可能写入各种地方， 实现sink类，  Agent程序可以在多个机器上运行  无论是缓存数据，还是写数据。都会先将数据封装成event（head{描述信息},body{数据}）对象。 |

# 二、需求：将日志采集到HDFS上

## 1、配置

|  |
| --- |
| #定义三大组件的名称  ag1.sources = source1  ag1.sinks = sink1  ag1.channels = channel1  # 配置source组件  #spooldir ,flume中自带的一种source实现类，读取目录的。行为：设置的目录中，一有新文件增加，就去读取  ag1.sources.source1.type = spooldir  #具体的目录  ag1.sources.source1.spoolDir = /root/log/  #指定后缀名。采取后的文件会给这个文件增加一个后缀名。  ag1.sources.source1.fileSuffix=.FINISHED  # 配置sink组件  #sinks类型，写入hdfs  ag1.sinks.sink1.type = hdfs  #数据存储的目录（hdfs的目录）  ag1.sinks.sink1.hdfs.path =hdfs://hdp-01:9000/access\_log/%y-%m-%d/%H-%M  #写入hdfs中，生成文件的前缀名  ag1.sinks.sink1.hdfs.filePrefix = app\_log  #写入hdfs中，生成文件的后缀  ag1.sinks.sink1.hdfs.fileSuffix = .log  #采集一批后才写入,一批的数量  ag1.sinks.sink1.hdfs.batchSize= 100  #写入数据后的文件类型。 DataStream：写入什么就是什么 。还有序列化，和压缩  ag1.sinks.sink1.hdfs.fileType = DataStream  #  ag1.sinks.sink1.hdfs.writeFormat =Text  ## roll：滚动切换：控制写文件的切换规则  ## 按文件体积（字节）来切  ag1.sinks.sink1.hdfs.rollSize = 512000  ## 按event条数切  ag1.sinks.sink1.hdfs.rollCount = 1000000  ## 按时间间隔切换文件  ag1.sinks.sink1.hdfs.rollInterval = 60  ## 控制生成目录的规则  ag1.sinks.sink1.hdfs.round = true  ag1.sinks.sink1.hdfs.roundValue = 10  ag1.sinks.sink1.hdfs.roundUnit = minute  ag1.sinks.sink1.hdfs.useLocalTimeStamp = true  # channel组件配置  ag1.channels.channel1.type = memory  ## event条数  ag1.channels.channel1.capacity = 500000  ##flume事务控制所需要的缓存容量600条event  ag1.channels.channel1.transactionCapacity = 600  # 绑定source、channel和sink之间的连接  ag1.sources.source1.channels = channel1  ag1.sinks.sink1.channel = channel1 |

## 2、启动

|  |
| --- |
| 1、测试启动，会将操作的日志打印在控制台  bin/flume-ng agent –c ./confi –f dir-hdfs.confg –n ag1  –Dflume.root.logger=DEBUG，console   1. agent 启动一个agent程序 2. -c /confg flume程序的配置文件 3. -f /dir-hdfs.confg flume程序相关组件的配置 4. -n ag1 启用组件配置文件里的某一个组件 5. -Dflume.root.looger=DEBUG,console 启用日志   2、生产中启动  Nohup bin/flume-ng age –c ./conf –f ./dir-hdfs.conf –n –agent1 1>/dev/null 2>&1 & |
|  |

## 3、其它source组件

|  |
| --- |
| 1、exec 读取命令的输出  ag1.sources.source1.type =exec  ag1.sources.source1.command = tail –F /root/log/access.log  -F 一直监视这个文件名的文件  修改二、1里面的source即可，其它不用修改 |

# 三、avro串联

## 1、情景

|  |
| --- |
| 分布式系统中，每个服务器都会产生日志。所以需要在每个服务上运行一个agent，将该服务器上的日志都，发送到一个一个特别的agent上进统一处理。  此时，收集日志的agent，相关于一个socket客户端。统一处理的agen相当于一个socket  的服务端。 |

## 2、客户端配置

|  |
| --- |
| 从tail命令获取数据发送到avro端口  另一个节点可配置一个avro源来中继数据，发送外部存储  ##################  # Name the components on this agent  a1.sources = r1  a1.sinks = k1  a1.channels = c1  # Describe/configure the source  a1.sources.r1.type = exec  a1.sources.r1.command = tail -F /root/log/access.log  # Describe the sink  a1.sinks.k1.type = avro  a1.sinks.k1.hostname = hdp-05  a1.sinks.k1.port = 4141  a1.sinks.k1.batch-size = 2  # Use a channel which buffers events in memory  a1.channels.c1.type = memory  a1.channels.c1.capacity = 1000  a1.channels.c1.transactionCapacity = 100  # Bind the source and sink to the channel  a1.sources.r1.channels = c1  a1.sinks.k1.channel = c1 |

## 3、服务端配置

|  |
| --- |
| #采集配置文件，avro-hdfs.conf  # Name the components on this agent  a1.sources = r1  a1.sinks = k1  a1.channels = c1  # Describe/configure the source  ##source中的avro组件是一个接收者服务  a1.sources.r1.type = avro  a1.sources.r1.bind = hdp-05  a1.sources.r1.port = 4141  # Describe the sink  a1.sinks.k1.type = hdfs  a1.sinks.k1.hdfs.path = /flume/taildata/%y-%m-%d/  a1.sinks.k1.hdfs.filePrefix = tail-  a1.sinks.k1.hdfs.round = true  a1.sinks.k1.hdfs.roundValue = 24  a1.sinks.k1.hdfs.roundUnit = hour  a1.sinks.k1.hdfs.rollInterval = 0  a1.sinks.k1.hdfs.rollSize = 0  a1.sinks.k1.hdfs.rollCount = 50  a1.sinks.k1.hdfs.batchSize = 10  a1.sinks.k1.hdfs.useLocalTimeStamp = true  #生成的文件类型，默认是Sequencefile，可用DataStream，则为普通文本  a1.sinks.k1.hdfs.fileType = DataStream  # Use a channel which buffers events in memory  a1.channels.c1.type = memory  a1.channels.c1.capacity = 1000  a1.channels.c1.transactionCapacity = 100  # Bind the source and sink to the channel  a1.sources.r1.channels = c1  a1.sinks.k1.channel = c1 |

4、启动

|  |
| --- |
| 启动时，先启动相当于socket服务端的agent，再启客户端  bin/flume-ng agent -c conf -f conf/avro-m-log.conf -n a1 -Dflume.root.logger=INFO,console |

## 四、自定义组件

|  |
| --- |
| 以上使用到了三个组件，source，chanel，sink  还有intercpter拦截器。对数据的处理（如加密）可以在这一歩进行。  自定义流程，进行java编程实现功能。打成jar包。运行时指定jar包 |