TRANSWARP
星 环 科 技

大数据平台流处理工具：Flume

星环科技
www.transwarp.io

版权声明

议题

- Flume介绍
- Flume的工作模式
- Flume的安装和使用
- Flume使用

Flume简介

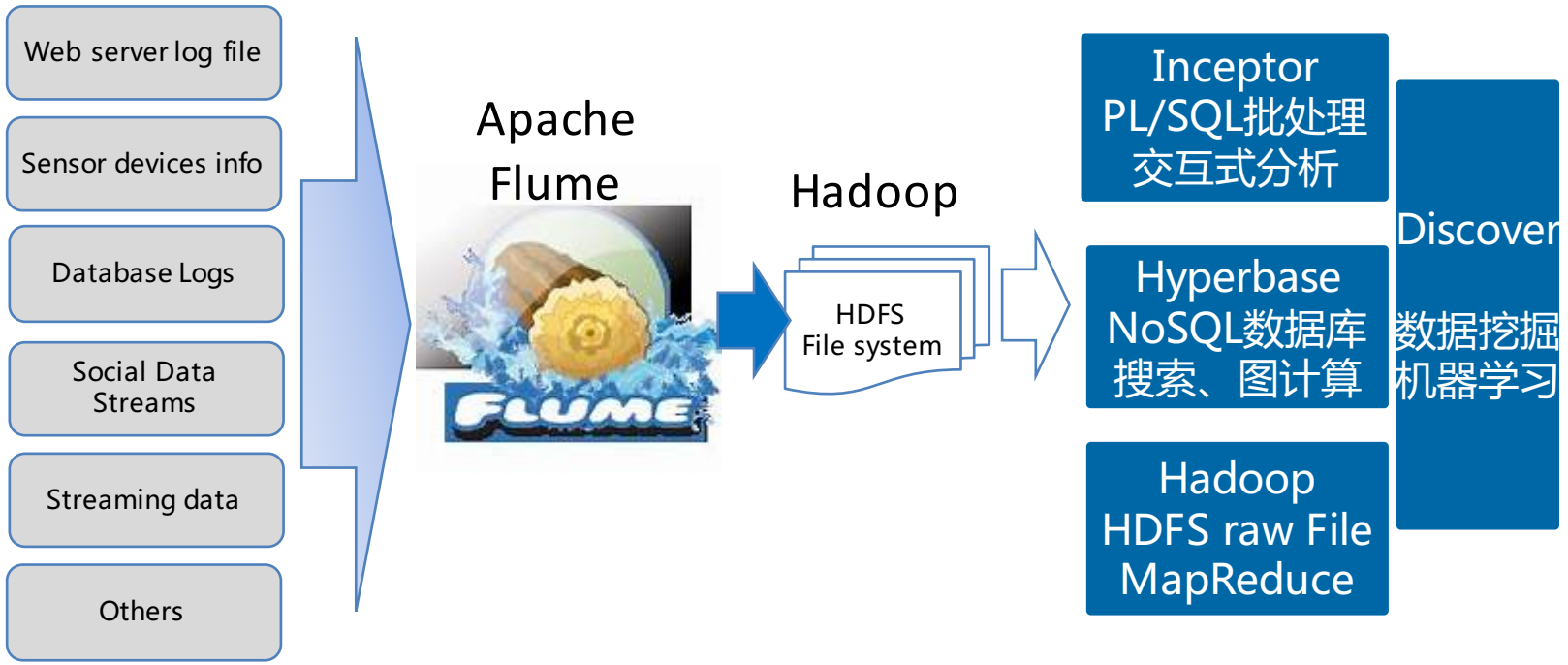
- Apache Flume 是一个分布式，可信任和弹性的系统，用于高效收集、汇聚和移动大规模日志信息从多种不同的数据源到一个集中的数据存储中心。
- flume是一个分布式、可靠、和高可用的海量日志采集、聚合和传输的系统。支持在日志系统中定制各类数据发送方，用于收集数据;同时，Flume提供对数据进行简单处理，并写到各种数据接受方(比如文本、HDFS、Hbase等)的能力 。
- 但是Flume的数据源可以自定义，所以也可以用于其他事件数据的传输，比如网络数据，社交媒体产生的数据，邮件信息等其他数据。
- 它是Apache软件基金的顶级项目。
- 目前有两个可用的分类版本，0.9.x和1.x。

Flume简介

TRANSWARP 星环科技

Apache Flume是一个海量日志采集、传输系统。

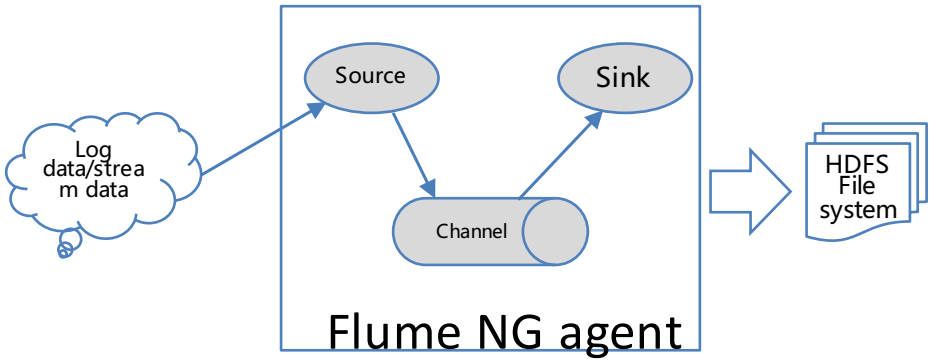
- 开源
- 高可靠
- 易操作
- 可定制



Flume的简单架构--数据流模式

TRANSWARP 星 环 科 技

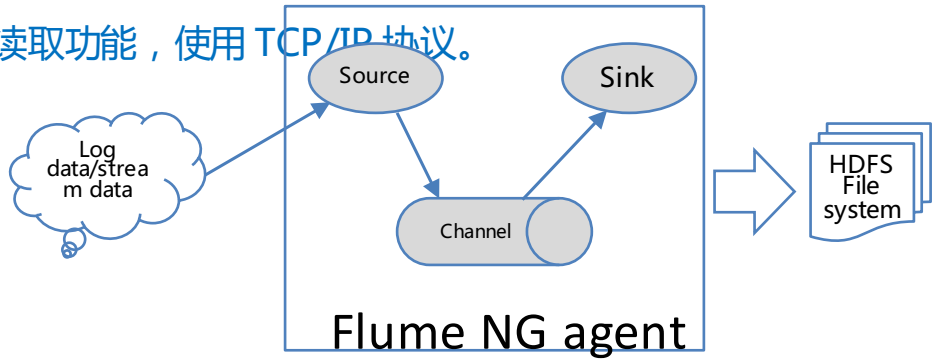
- flume事件
  - 一次flume事件定义为一个单元的数据流，该数据流包含一个字节的负荷以及一组可选的字符串属性。
- flume agent
  - 一个flume Agent是一个JVM进程，用于控制一个组件将外部事件流引导到下一个目的地
- flume源
  - 一个flume源（source）负责一个外部源(如一个web服务器传递给它的事件)。该外部源将它的事件以Flume可以识别的格式发送到flume。
- 通道
  - 当一个flume源接收到一个事件时，其将通过一个或者多个通道存储该事件。通道（channel）采用被动存储的形式，即通道会缓存该事件直到该事件被sink组件处理。
- sink
  - Sink会将事件从通道中移除并将事件放置到外部数据仓库
  - 对于缓存在通道中的事件，source和sink采用异步处理的方式。



Flume能接收的数据源

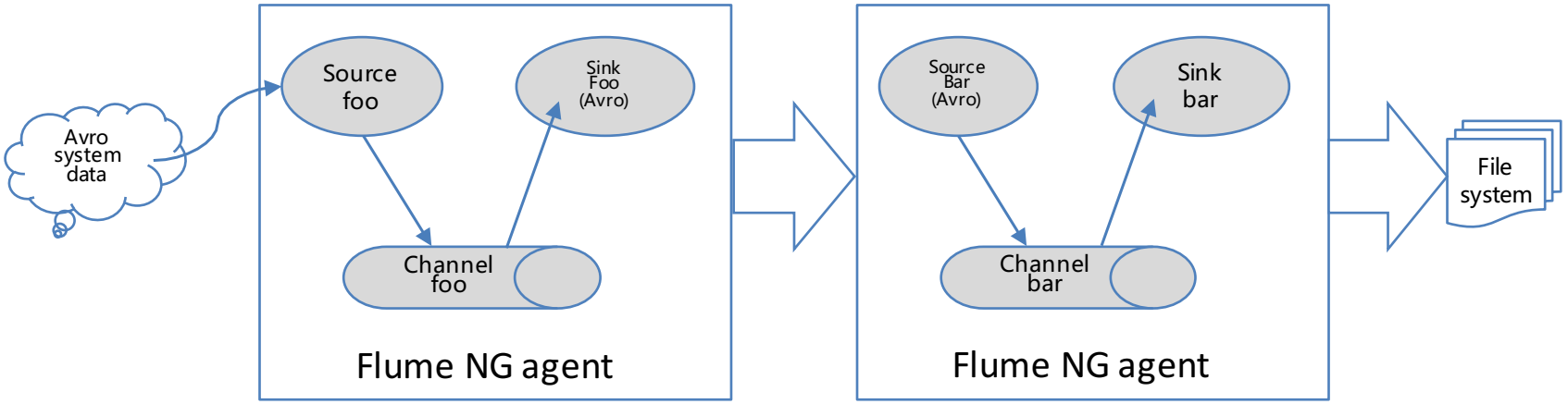
TRANSWARP 星 环 科 技

- Avro
  - Avro是Hadoop中的一个子项目，也是Apache中一个独立的项目，Avro是一个基于二进制数据传输高性能的中间件。在Hadoop的其他项目中例如HBase(Ref)和Hive(Ref)的Client端与服务端的数据传输也采用了这个工具。Avro是一个数据序列化的系统。Avro 可以将数据结构或对象转化成便于存储或传输的格式。Avro设计之初就用来支持数据密集型应用，适合于远程或本地大规模数据的存储和交换 。
- Thrift
  - Thrift是Facebook的一个开源项目，主要是一个跨语言的服务开发框架。它有一个代码生成器来对它所定义的IDL定义文件自动生成服务代码框架。
- Syslog系统日志
  - 服务器或网络路由在运行当中的信息记录，其将通过一个或者多个通道存储该事件。
- Netcat网络监听
  - 可以监听网络端口，对Netcat等网络程序能实现读取功能，使用 TCP/IP 协议。



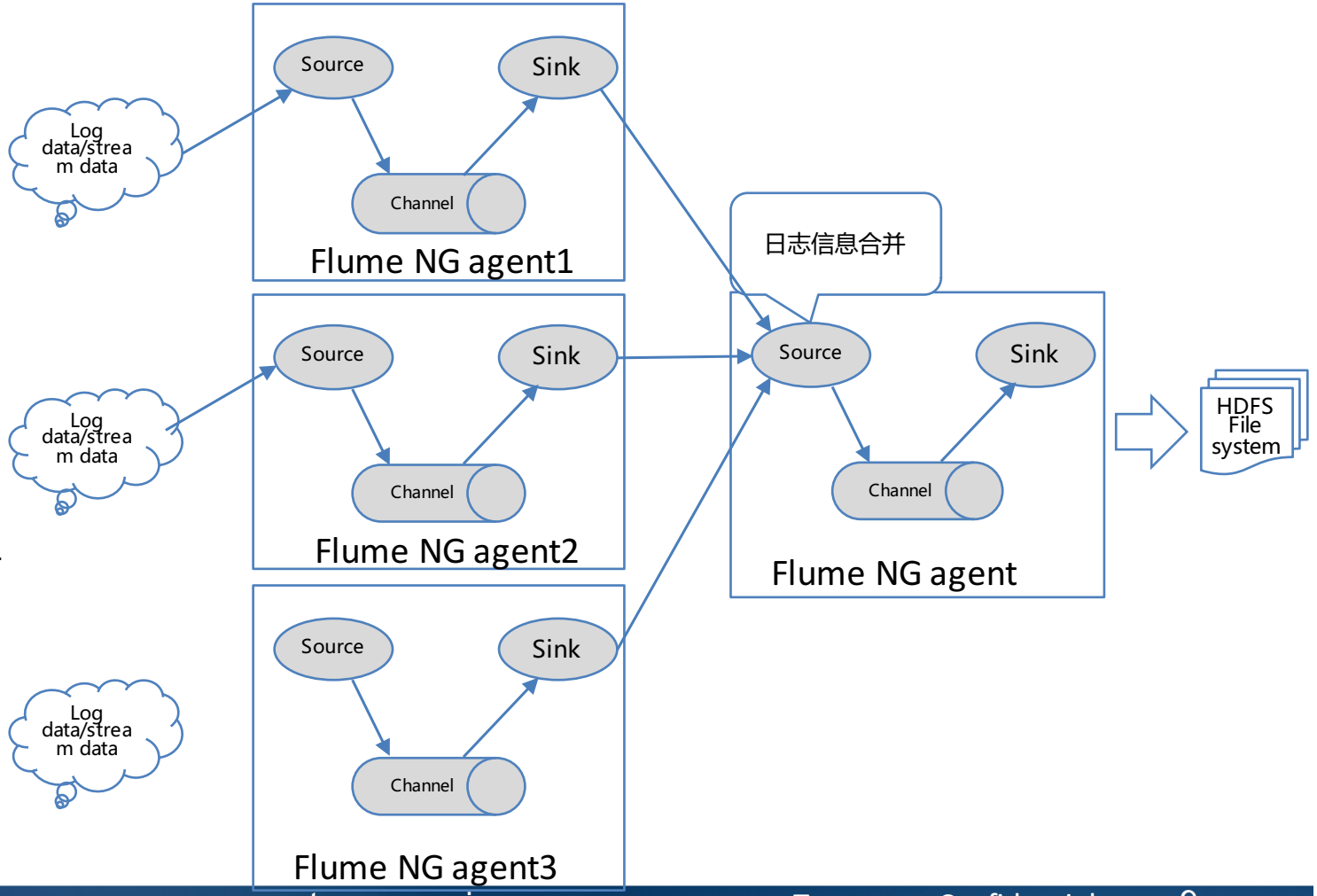
多agent模式

- 为了让流入的数据可以跨越多个代理或者跳转，前一个代理的sink和当前代理的source必须都是avro类型，同时sink需要指向source的主机名和端口号。



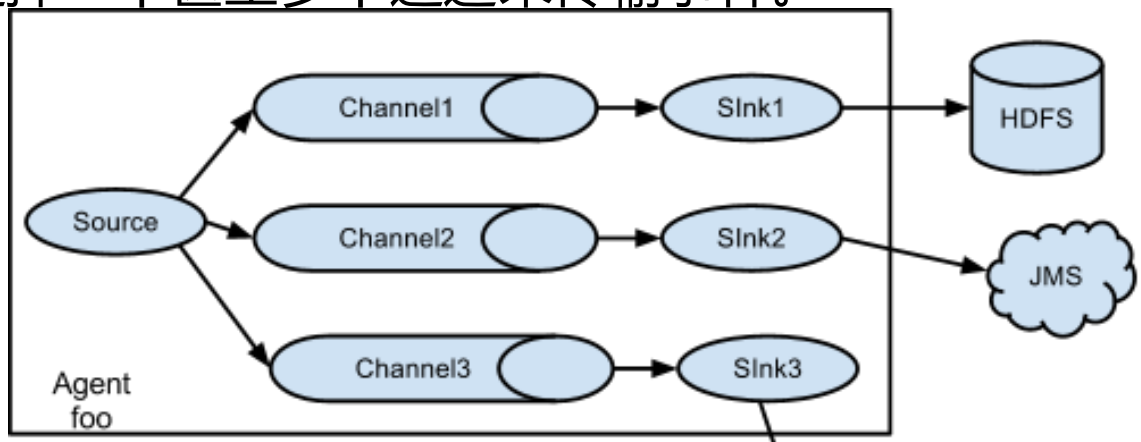
合并模式(Consolidation)

- 一个常见的场景是在日志收集中，大量的日志在客户端产生，然后发送到几个代理并最终存储到数据仓库。
- 如图，大量的web服务器产生的数据被收集并通过多个代理跳转最后存储到HDFS集群中。这种场景可以通过flume实现。通过配置大量的一线代理和Avro sink，然后一线代理的sink都指向一个avro source。该源位于二线的代理中，二线代理合并接收到的事件，通过channel传到sink再到达最终目的地。

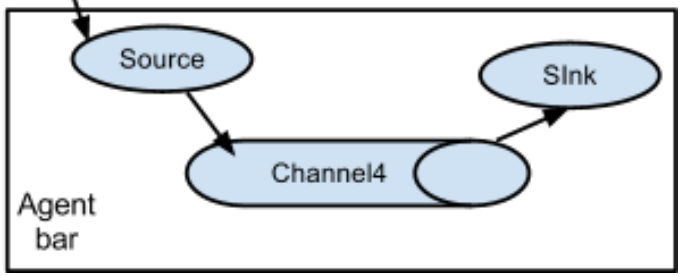


流的复用

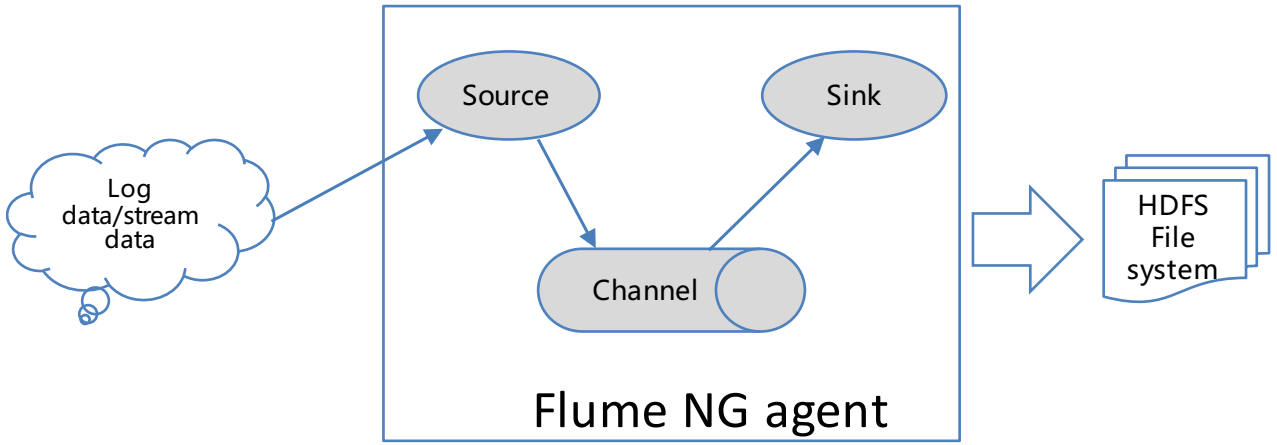
flume支持将事件流复用传输到多个目的地。其实现方式通过定义一个flow multiplexer来复制或选择一个甚至多个通道来传输事件。



如图，一个来带名字为foo的代理的源将流展开成三个不同的通道。该展开可以是复制或者复用。如果是流的复制，则每个事件均被送到三个通道中。如果是流的复用，则一个事件将会使用该事件的数据来匹配一个预先设置的值，然后传输到匹配后的通道集中。例如，一个事件的属性如果命名为“t*n类型”则设置为“customer”，然后该事件会传输到通道1和通道3中。如果是一个“vendor”则传输到通道2，否则传输到通道3.该匹配方法预先设置在代理的配置文件中。



Flume 架构



1个代理(agent)由来源(source)、管道(channel)、接收器(sink) 3大组件构成。

Flume为每个组件提供多种类型，用户可根据需求选择。

Source

- 捕获外部事件
- 接收数据
- 将数据推送至channel

Channel

- 保存事件，直到数据得到处理

Sink

- 处理事件
- 将数据持存储、或产生新的事件推送至下一级source

常用组件类型

TRANSWARP 星 环 科 技

Source

- Spool directory source：监控文件夹，传输所有保存在该目录下的文件。
- Avro source：监听Avro端口，从外部Avro客户端接收事件。
- Exec source：运行可执行命令，传输该命令所返回的数据。

Channel

- Memory channel：事件存储在内存。高性能、高吞吐。
- JDBC channel：事件存储在数据库。支持还原、恢复。

Sink

- HDFS Sink：写入HDFS，支持生成文本文件、数据压缩。
- Avro Sink：将Flume event 转换为Avro event，再发送到指定端口。
- File Roll Sink：将数据存储到本地文件系统。

Flume Channels

TRANSWARP 星 环 科 技

- Memory Channel
- JDBC Channel
- Kafka Channel
- File Channel
- Spillable Memory Channel
- Pseudo Transaction Channel
- Custom Channel

Flume sink

TRANSWARP 星 环 科 技

- HDFS Sink
- Hive Sink
- Logger Sink
- Avro Sink
- Thrift Sink
- IRC Sink
- File Roll Sink
- Null Sink
- HBaseSinks
  - HBaseSink
  - AsyncHBaseSink
- MorphlineSolrSink
- ElasticSearchSink
- Kite Dataset Sink
- Kafka Sink
- Custom Sink

Flume 工作模式

- 对目录进行监控

- 对文件的变化进行监控

- 对端口进行监控


Port端口

Flume，Kafka vs Sqoop

TRANSWARP 星环科技

	Flume	Kafka	Sqoop
发起者	Apple，Cloudera	Linkedin	DELL,Google,Apple, Clodera, Hortonworks等
使用场景	Web日志，离线处理流方式的日志信息	实时处理数据处理和接受	关系型数据库
数据处理方式	流数据(1s)	流数据(100-300ms)	RDD、规范数据
运营方式	分布式，延迟稍高	分布式，延迟低	集中式
数据处理方式	多数据源小数据	大容量的流数据	单数据源大数据

Flume对文件的监控和stream导入HDFS

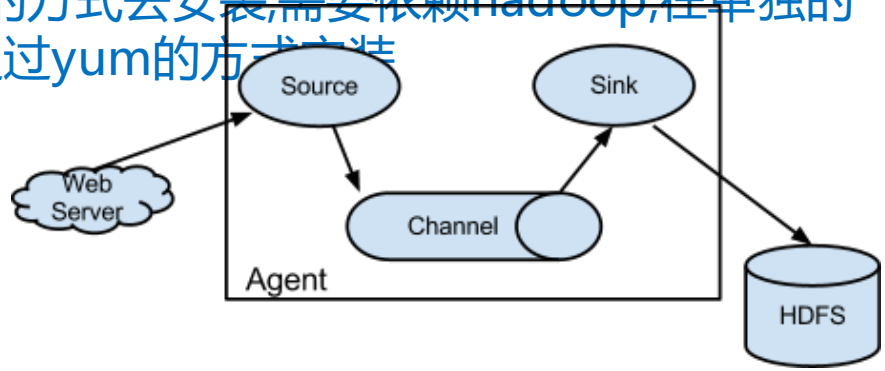
单个代理安装

- 采集节点flume安装:
  - 1. 环境依赖:有JDK环境,最好JDK1.6+

```
[root@host1 ~]# java -version
java version "1.7.0_71"
Java(TM) SE Runtime Environment (build 1.7.0_71-b14)
Java HotSpot(TM) 64-Bit Server VM (build 24.71-b01, mixed mode)
```

  - 2. 开源flume安装,下载http://flume.apache.org/download.html，安装只需要将包解压，并在flume-env.sh中配置JAVA_HOME
  - 3. 目前TDH上是通过yum install flume的方式去安装,需要依赖hadoop,在单独的agent通过yum的方式安装

```
[root@host1 ~]# yum install flume
```



单个代理安装

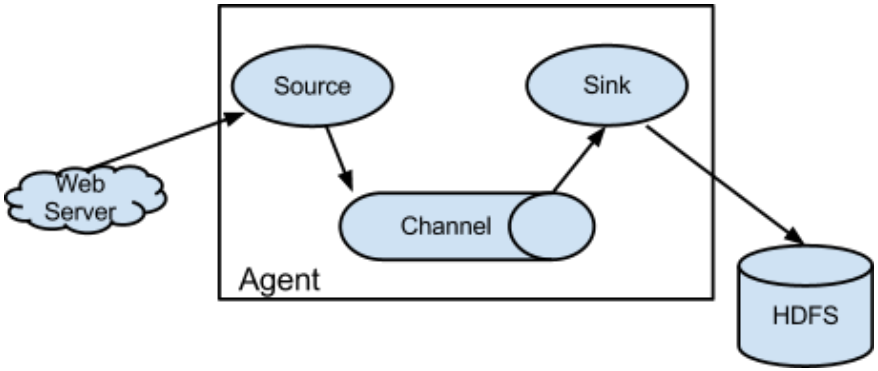
TRANSWARP 星 环 科 技

- 入库节点flume安装
  - 1. 环境依赖:JDK,Hadoop相关lib和配置

```
[root@host2 ~]# java -version
java version "1.7.0_71"
Java(TM) SE Runtime Environment (build 1.7.0_71-b14)
Java HotSpot(TM) 64-Bit Server VM (build 24.71-b01, mixed mode)
```

  - 2. 开源flume安装,配置好hadoop的环境之后,同采集节点的安装,并且在flume-env.sh中配置HADOOP_HOME等环境变量
  -

```
[root@host2 ~]# yum install flume
```



配置示例

TRANSWARP 星环科技

- 配置单个节点的flume的部署。该配置让用户生成的事件并在随后将日志输

```
[root@host1 conf]# pwd
/etc/flume/conf
[root@host1 conf]# cp flume-conf.properties.template flume-conf.properties.test
[root@host1 conf]# cat flume-conf.properties.test
------------------------------------------
# A single-node Flume configuration

# Name the components on this agent
a1.sources = r1
a1.sinks = k1
a1.channels = c1

# Describe/configure the source
a1.sources.r1.type = netcat
a1.sources.r1.bind = localhost
a1.sources.r1.port = 44444
```

```
# Describe the sink
a1.sinks.k1.type = logger

# 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
```

该配置文件定义一个代理命名为a1.a1的源在44444端口监听数据，通道在内存中缓存数据，sink记录事件数据到控制台。配置文件首先命名各个组件，然后描述各个组件的类型和配置参数。一个给定的配置文件可能定义多个代理。当一个指定的flunme进程在运行时，一个标志被传递进来，告知该代理显示哪个名字。

启动代理

- 代理通过flume文件夹中bin目录下的flume-ng的脚本来启动。
  - 在执行脚步命令时，你需要指定代理的名称、配置的目录和配置文件
  - 成功执行命令后，代理会按照给定的配置属性文件来启动运行source和sink。
  - 需要注意的是在一个完整的部署中，我们通常会选择一个以上的选项：--conf=<conf-dir>。其中<conf-dir>的目录需要包括shell脚本flume-env.sh以及一个可选的log4j属性文件。在该例子中，我们使用一个java选项来强制flume将数据输出到控制台，同时我们没有自定义环境脚本。

```
[root@host1  conf]# flume-ng agent --conf conf --conf-file /etc/flume/conf/flume-conf.properties.test --name a1 -Dflume.root.logger=INFO,console
```

测试

TRANSWARP 星环科技

- 通过一个单独的终端去telnet44444端口，然后发送事件到flume。

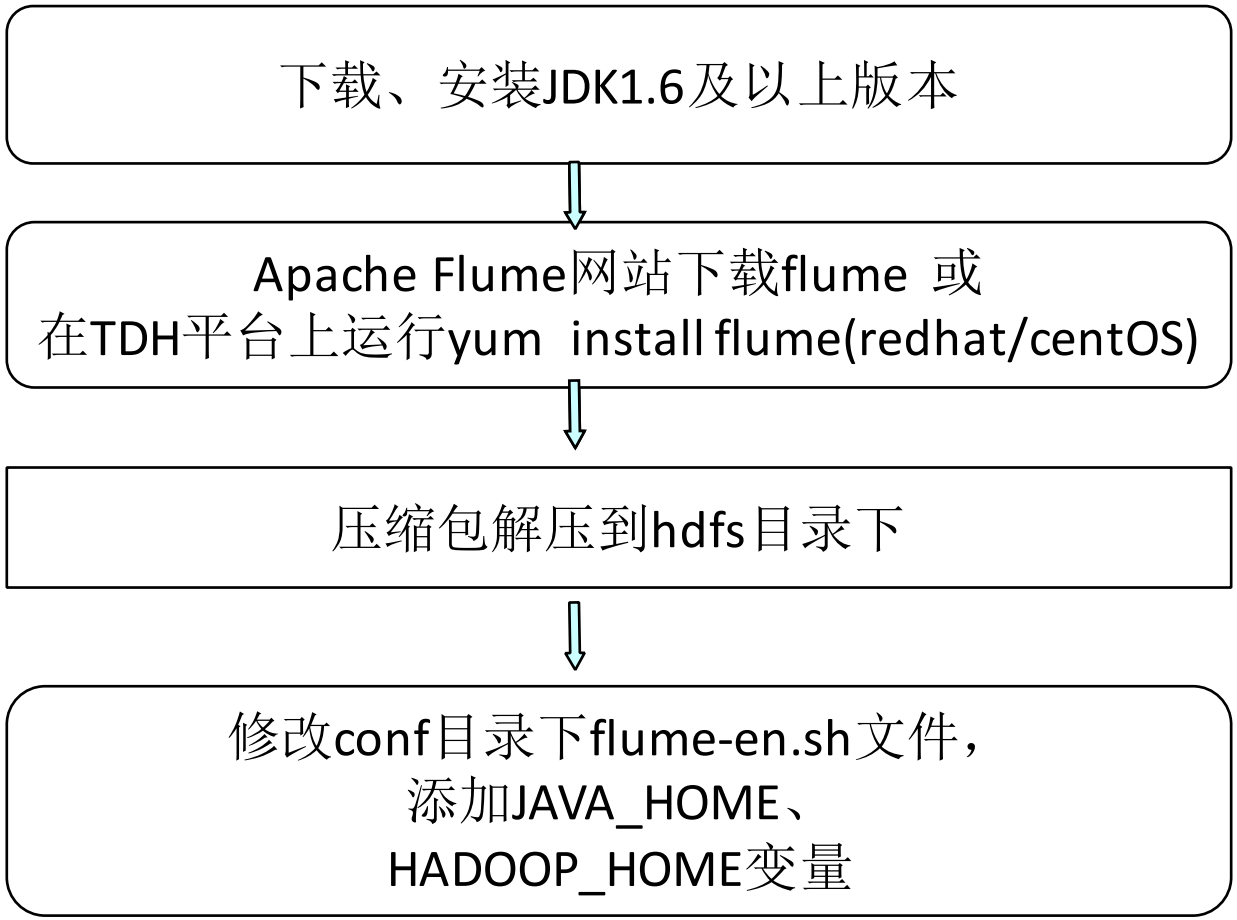
终端1：

```
[root@host1 ~]# telnet localhost 44444
Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Hello Flume!
OK
```

启动agent的终端：

```
2015-12-07 21:54:13,523 INFO node.Application: Starting Sink k1
2015-12-07 21:54:13,524 INFO node.Application: Starting Source r1
2015-12-07 21:54:13,524 INFO source.NetcatSource: Source starting
2015-12-07 21:54:13,545 INFO source.NetcatSource: Created serverSocket:sun.nio.ch.ServerSocketChannelImpl[/127.0.0.1:44444]
2015-12-07 21:59:55,555 INFO sink.LoggerSink: Event: { headers:{} body: 48 65 6C 6C 6F 20 46 6C 75 6D 65 21 0D       Hello Flume! }
```

安装Flume


下载、安装JDK1.6及以上版本
Apache Flume网站下载flume 或
在TDH平台上运行yum install flume(redhat/centOS)
压缩包解压到hdfs目录下
修改conf目录下flume-en.sh文件，
添加JAVA_HOME、
HADOOP_HOME变量

配置Flume 参数

TRANSWARP 星 环 科 技

/etc/profile 文件中添加

```
export JAVA_HOME=/usr/java/latest
export HADOOP_HOME=/usr/lib/hadoop
export HDFS_HOME=/usr/lib/hadoop-hdfs
export HADOOP_CLASSPATH=$HADOOP_HOME:$HADOOP_HOME/lib:$HDFS_HOME/lib
```

/etc/flume/conf/flume-env.sh

```
FLUME_HOME="/usr/lib/flume"
FLUME_CLASSPATH="$FLUME_HOME/lib"
JAVA_HOME="/usr/java/latest"
JAVA_OPTS="-Xms512m  -Xmx1024m  -Dcom.sun.management.jmxremote"
HADOOP_HOME="/usr/lib/hadoop"
HDFS_HOME="/usr/lib/hadoop-hdfs"
HADOOP_CLASSPATH="$HADOOP_HOME:$HADOOP_HOME/lib:$HDFS_HOME/lib"
```

对文件的监控

- 监控某一个文件使用的是exec source模式。
  - Exec source执行一个特定的命令。
  - tail命令可以监控文件尾部增加的数据。
- Flume每秒会扫描文件检查是否有新的更新
- Flume只对stream流方式写入的数据进行更新
- 新的数据自动导入到HDFS文件中并以raw文件方式存储

Flume的使用（对文件的监控）

TRANSWARP 星 环 科 技

运行flume只需两步：
1.创建agent，即编写agent配置文件。
2.命令行运行agent。

编写agent配置文件，其配置文件名为a.conf，如右图，存储在conf目录下。

指明所使用的source、sink、channel的类型，以及相关的参数。

tail命令后跟 -f path 监控文件末尾，有新数据会上传新数据。注意末尾数据应以流方式增加。跟 -F path 也是监控文件末尾，但每次有新增数据，则会将所有数据全部重新上传，效率较低。

Exec source 不保证传输过程的完整性和正确性，传输出错无法自动纠错，只能重新上传。

```
# a.conf: A single-node Flume configuration
# 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/flumelab/stockfile.log
a1.sources.r1.fileHeader  = true
a1.sources.r1.deserializer.outputCharset=UTF-8

# Describe the sink
a1.sinks.k1.type = hdfs
a1.sinks.k1.hdfs.path = hdfs://nameservice1/user/root/file1
a1.sinks.k1.hdfs.fileType = DataStream
a1.sinks.k1.hdfs.writeFormat=Text
a1.sinks.k1.hdfs.maxOpenFiles = 1
a1.sinks.k1.hdfs.rollCount = 0
a1.sinks.k1.hdfs.rollInterval = 0
a1.sinks.k1.hdfs.rollSize = 1000000
a1.sinks.k1.hdfs.batchSize = 10000

# Use a channel which buffers events in memory
a1.channels.c1.type = memory
a1.channels.c1.capacity = 10000
a1.channels.c1.transactionCapacity = 1000000

# Bind the source and sink to the channel
a1.sources.r1.channels = c1
a1.sinks.k1.channel = c1
```

Flume的运行（对文件的监控）

TRANSWARP 星 环 科 技

Agent 运行命令(flume目录下)

#bin/flume-ng agent --conf-file conf/a.conf --name a1 -Dflume.root.logger=INFO,console

例：
1. #bin/flume-ng agent -n a1 -c /etc/flume/conf -f /etc/flume/conf/flume.conf -Dflume.root.logger=INFO,console

2.

*--conf-file后为.conf文件目录
*--name后为agent名称

Flume对目录进行监控并导入到HDFS中

Spooling Directory Source

TRANSWARP 星环科技

- 这个源支持从磁盘中某文件夹获取文件数据。不同于其他异步源，这个源能够避免重启或者发送失败后数据丢失。flume可以监控文件夹，当出现新文件时会读取该文件并获取数据。当一个给定的文件被全部读入到通道中时，该文件会被重命名以标志已经完成。同时，该源需要一个清理进程来定期移除完成的文件。
- 注意：发送失败时候可能会重复发送事件（同其他组件规则一致requierd）。
- 通道可选地将一个完成路径的原始文件插入到每个事件的hearder域中。在读取文件时，source缓存文件数据到内存中。同时，需要确定设置了bufferMaxLineLength选项，以确保该数据远大于输入数据中数据最长的某一行。
- 注意：channel只接收spooling directory中唯一命名的文件。如果文件名重复或文件在读取过程中被修改，则会有读取失败返回异常信息。这种场景下，同名的文件复制到这个目录时建议带唯一标示，比如时间戳。

属性

TRANSWARP 星 环 科 技

Property Name	Default	Description
channels	–	
type	–	The component type name, needs to be `spooldir`
spoolDir	–	The directory where log files will be spooled
fileSuffix	.COMPLETED	Suffix to append to completely ingested files
fileHeader	false	Whether to add a header storing the filename
fileHeaderKey	file	Header key to use when appending filename to header
batchSize	10	Granularity at which to batch transfer to the channel
bufferMaxLines	100	Maximum number of lines the commit buffer can hold
bufferMaxLineLength	5000	Maximum length of a line in the commit buffer
selector.type	replicating	replicating or multiplexing
selector.*		Depends on the selector.type value
interceptors	–	Space separated list of interceptors
interceptors.*		

File Channel

TRANSWARP 星环科技

- 默认情况下，文件通道使用路径来作为校验点和数据的目录（在用户的主目录内）。
- 因此，如果代理中存在多个文件通道，只能有一个通道可用，该通道会锁住其配置的目录，而使得其他通道无法使用该目录而失败。
- 因此，有必要提供明确的路径给所有已配置的通道，最好是在不同的磁盘上。
- 此外，由于文件通道将同步到磁盘在每次提交后，通道与sink、source相连，需要为每组事件提供好的性能当多个硬盘不用于校验点和数据目录时。

属性

TRANSWARP 星 环 科 技

Property Name Default	Description	
type	–	The component type name, needs to be `file`.
checkpointDir	~/.flume/file-channel/checkpoint	The directory where checkpoint file will be stored
dataDirs	~/.flume/file-channel/data	The directory where log files will be stored
transactionCapacity	1000	The maximum size of transaction supported by the channel
checkpointInterval	30000	Amount of time (in millis) between checkpoints
maxFileSize	2146435071	Max size (in bytes) of a single log file
capacity	1000000	Maximum capacity of the channel
keep-alive	3	Amount of time (in sec) to wait for a put operation
write-timeout	3	Amount of time (in sec) to wait for a write operation
checkpoint-timeout	600	Expert: Amount of time (in sec) to wait for a checkpoint
use-log-replay-v1	false	Expert: Use old replay logic
use-fast-replay	false	Expert: Replay without using queue
encryption.activeKey	–	Key name used to encrypt new data
encryption.cipherProvider	–	Cipher provider type, supported types: AESCTRNOPADDING
encryption.keyProvider	–	Key provider type, supported types: JCEKSFILE
encryption.keyProvider.keyStoreFile	–	Path to the keystore file
encrpytion.keyProvider.keyStorePasswordFile	–	Path to the keystore password file
encryption.keyProvider.keys	–	List of all keys (e.g. history of the activeKey setting)
encyption.keyProvider.keys.*.passwordFile	–	Path to the optional key password file

HDFS sink

TRANSWARP 星 环 科 技

- 用于写入HDFS.支持生成文本文件和序列化文件，支持数据压缩。该sink支持文件定期滚动（即关闭当前文件，然后产生新的文件），可以通过时间、文件大小、事件数目等形式来确定是否产生新的文件。同时可以同过时间戳或者时间产生的主机来区分数据。使用该sink需要安装hadoop，flume需要使用hadoop的jar包来与HDFS集群通信。

escape sequences

TRANSWARP 星 环 科 技

Alias	Description
%{host}	header name"host"的替换值，可以任意.
%t	Unix time毫秒级别
%a	locale's short weekday name (Mon, Tue, ...)
%A	locale's full weekday name (Monday, Tuesday, ...)
%b	locale's short month name (Jan, Feb, ...)
%B	locale's long month name (January, February, ...)
%c	locale's date and time (Thu Mar 3 23:05:25 2005)
%d	day of month (01)
%D	date; same as %m/%d/%y
%H	hour (00..23)
%I	hour (01..12)
%j	day of year (001..366)
%k	hour (0..23)
%m	month (01..12)
%M	minute (00..59)
%p	locale's equivalent of am or pm
%s	seconds since 1970-01-01 00:00:00 UTC
%S	second (00..59)
%y	last two digits of year (00..99)
%Y	year (2010)
%z	+hhmm numeric timezone (for example, -0400)

属性

TRANSWARP 星 环 科 技

Name	Default	Description
channel	–	
type	–	The component type name, needs to be `hdfs`
hdfs.path	–	HDFS directory path (eg hdfs://namenode/flume/webdata/)
hdfs.filePrefix	FlumeData	Name prefixed to files created by Flume in hdfs directory
hdfs.fileSuffix	–	Suffix to append to file (eg `.avro` - *NOTE: period is not automatically added*)
hdfs.rollInterval	30	Number of seconds to wait before rolling current file (0 = never roll based on time interval)
hdfs.rollSize	1024	File size to trigger roll, in bytes (0: never roll based on file size)
hdfs.rollCount	10	Number of events written to file before it rolled (0 = never roll based on number of events)
hdfs.idleTimeout	0	Timeout after which inactive files get closed (0 = disable automatic closing of idle files)
hdfs.batchSize	100	number of events written to file before it is flushed to HDFS
hdfs.codeC	–	Compression codec. one of following : gzip, bzip2, lzo, snappy
hdfs.fileType	SequenceFile	File format: currently `SequenceFile`, `DataStream` or `CompressedStream` (1)DataStream will not compress output file and please don't set codeC (2)CompressedStream requires set hdfs.codeC with an available codeC
hdfs.maxOpenFiles	5000	Allow only this number of open files. If this number is exceeded, the oldest file is closed.
hdfs.writeFormat	–	"Text" or "Writable"
hdfs.callTimeout	10000	Number of milliseconds allowed for HDFS operations, such as open, write, flush, close. This number should be increased if many HDFS timeout operations are occurring.
hdfs.threadsPoolSize	10	Number of threads per HDFS sink for HDFS IO ops (open, write, etc.)
hdfs.rollTimerPoolSize	1	Number of threads per HDFS sink for scheduling timed file rolling
hdfs.kerberosPrincipal	–	Kerberos user principal for accessing secure HDFS
hdfs.kerberosKeytab	–	Kerberos keytab for accessing secure HDFS
hdfs.proxyUser		
hdfs.round	false	Should the timestamp be rounded down (if true, affects all time based escape sequences except %t)
hdfs.roundValue	1	Rounded down to the highest multiple of this (in the unit configured using `hdfs.roundUnit`), less than current time.
hdfs.roundUnit	second	The unit of the round down value - `second`, `minute` or `hour`.
hdfs.timeZone	Local Time	Name of the timezone that should be used for resolving the directory path, e.g. America/Los_Angeles.
serializer	`TEXT`	Other possible options include `avro_event` or the fully-qualified class name of an implementation of the `EventSerializer.Builder` interface.
serializer.*		

配置参数(红色为必填项)

TRANSWARP 星 环 科 技

```
agent1.sources = spooldirSource
agent1.channels = fileChannel
agent1.sinks = hdfsSink

# Describe/configure the source
agent1.sources.spooldirSource.type=spooldir agent1.sources.spooldirSource.spoolDir=/data/lwq/new_log  agent1.sources.spooldirSource.channels=fileChannel

# Describe the sink
agent1.sinks.hdfsSink.type=hdfs
agent1.sinks.hdfsSink.hdfs.path=hdfs://dev228:8020/raw/lwq/%y-%m-%d
agent1.sinks.hdfsSink.hdfs.filePrefix=lwq

agent1.sinks.sink1.hdfs.round = true
# Number of seconds to wait before rolling current file (0 = never roll based on time interval)
agent1.sinks.hdfsSink.hdfs.rollInterval = 30
# File size to trigger roll, in bytes (0: never roll based on file size)
agent1.sinks.hdfsSink.hdfs.rollSize = 128000000
agent1.sinks.hdfsSink.hdfs.rollCount = 0
agent1.sinks.hdfsSink.hdfs.batchSize = 1000 #Rounded down to the highest multiple of this (in the unit configured using hdfs.roundUnit), less than current time.
agent1.sinks.hdfsSink.hdfs.roundValue = 1
agent1.sinks.hdfsSink.hdfs.roundUnit = minute
agent1.sinks.hdfsSink.hdfs.useLocalTimeStamp = true
agent1.sinks.hdfsSink.channel=fileChannel
agent1.sinks.hdfsSink.hdfs.fileType = DataStream

# Describe the file channel
agent1.channels.fileChannel.type = file
agent1.channels.fileChannel.checkpointDir=/usr/share/apache-flume-1.5.0-bin/checkpoint  //提前创建
agent1.channels.fileChannel.dataDirs=/usr/share/apache-flume-1.5.0-bin/dataDir//提前创建
```

启动代理

TRANSWARP 星环科技

- 代理通过flume文件夹中bin目录下的flume-ng的脚本来启动。
  - 在执行脚步命令时，你需要指定代理的名称、配置的目录和配置文件
  - 成功执行命令后，代理会按照给定的配置属性文件来启动运行source和sink。
  - 需要注意的是在一个完整的部署中，我们通常会选择一个以上的选项：--conf=<conf-dir>。其中<conf-dir>的目录需要包括shell脚本flume-env.sh以及一个可选的log4j属性文件。在该例子中，我们使用一个java选项来强制flume将数据输出到控制台，同时我们没有自定义环境脚本。

```
[root@host1 conf]# flume-ng agent --conf conf --conf-file /etc/flume/conf/flume-conf.properties.logtohdfs --name agent1
[root@host1 conf]# flume-ng agent --conf conf --conf-file /etc/flume/conf/flume-conf.properties.logtohdfs --name agent1 &
```

运行

flume-ng agent -n a1 -c /etc/flume/conf -f /etc/flume/conf/flume.conf -Dflume.root.logger=INFO,console

-n参数是conf文件中，agent的名字。
-c参数是flume-env.sh和log4j.properties所在目录，log4j.properties不需要修改，本来就在conf目录下。
-f 是conf文件所在目录，要写清楚哪个conf，比如a.conf。
-D flume.root.logge参数就是将运行结果输出到哪里，console就是terminal端。

```
2015-12-17 09:50:19,490 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO -
org.apache.flume.sink.hdfs.BucketWriter.open(BucketWriter.java:261)] Creating hd
fs://nameservice1/tmp/gy/test.1450317019407.tmp
2015-12-17 09:50:50,769 (hdfs-k1-roll-timer-0) [INFO - org.apache.flume.sink.hdf
s.BucketWriter.close(BucketWriter.java:409)] Closing hdfs://nameservice1/tmp/gy/
test.1450317019407.tmp
2015-12-17 09:50:50,770 (hdfs-k1-call-runner-2) [INFO - org.apache.flume.sink.hd
fs.BucketWriter$3.call(BucketWriter.java:339)] Close tries incremented
2015-12-17 09:50:50,851 (hdfs-k1-call-runner-4) [INFO - org.apache.flume.sink.hd
fs.BucketWriter$8.call(BucketWriter.java:669)] Renaming hdfs://nameservice1/tmp/
gy/test.1450317019407.tmp to hdfs://nameservice1/tmp/gy/test.1450317019407
2015-12-17 09:50:50,871 (hdfs-k1-roll-timer-0) [INFO - org.apache.flume.sink.hdf
s.HDFSEventSink$1.run(HDFSEventSink.java:402)] Writer callback called.
2015-12-17 09:50:50,871 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO -
org.apache.flume.sink.hdfs.HDFSEventSink.process(HDFSEventSink.java:428)] Bucket
 was closed while trying to append, reinitializing bucket and writing event.
2015-12-17 09:50:50,871 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO -
org.apache.flume.sink.hdfs.HDFSDataStream.configure(HDFSDataStream.java:58)] Ser
ializer = TEXT, UseRawLocalFileSystem = false
2015-12-17 09:50:50,904 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO -
org.apache.flume.sink.hdfs.BucketWriter.open(BucketWriter.java:261)] Creating hd
fs://nameservice1/tmp/gy/test.1450317050872.tmp
2015-12-17 09:50:51,347 (pool-5-thread-1) [INFO - org.apache.flume.client.avro.R
eliableSpoolingFileEventReader.rollCurrentFile(ReliableSpoolingFileEventReader.j
ava:332)] Preparing to move file /etc/flume/spool/data to /etc/flume/spool/data.
COMPLETED
2015-12-17 09:50:51,348 (pool-5-thread-1) [INFO - org.apache.flume.source.SpoolD
irectorySource$SpoolDirectoryRunnable.run(SpoolDirectorySource.java:254)] Spooli
ng Directory Source runner has shutdown.
2015-12-17 09:50:51,849 (pool-5-thread-1) [INFO - org.apache.flume.source.SpoolD
irectorySource$SpoolDirectoryRunnable.run(SpoolDirectorySource.java:254)] Spooli
ng Directory Source runner has shutdown.
```


TRANSWARP