Flume 配置案例

## 案例 1：start case （single-node configuration）

**#文件名：**case1\_example.conf

**#配置内容：**

# case1\_example.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 = 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

**#开始命令**

flume-ng agent -c conf -f conf/case1\_example.conf -n a1 -Dflume.root.logger=INFO,console

**#启动参数说明**

-c conf 指定配置目录为conf

-f conf/case1\_example.conf 指定配置文件为conf/case1\_example.conf

-n a1 指定agent名字为a1,需要与case1\_example.conf中的一致

-Dflume.root.logger=INFO,console 指定DEBUF模式在console输出INFO信息

**#在另一个终端进行测试**

telnet 127.0.0.1 44444

Trying 127.0.0.1...

Connected to localhost.localdomain (127.0.0.1).

Escape character is '^]'.

hello world!

OK

**#在启动的终端查看console输出**

2013-05-24 00:00:24,306 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{} body: 68 65 6C 6C 6F 20 77 6F 72 6C 64 21 0D hello world!. }

## 案例2：Test Avro Source

**#文件名：**case2\_avro.conf

**#配置内容：**

# Name the components on this agent

a1.sources = r1

a1.sinks = k1

a1.channels = c1

# Describe/configure the source

a1.sources.r1.type = avro

a1.sources.r1.channels = c1

a1.sources.r1.bind = 192.168.68.129

a1.sources.r1.port = 4141

# Describe the sink

a1.sinks.k1.type = logger

 a1.sinks.k1.channel = c1

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

**#Start flume agent a1**

flume-ng agent -c . -f case2\_avro.conf -n a1 -Dflume.root.logger=INFO,console

**#创建指定文件**

echo "hello world" > /usr/logs/log.10

**#使用avro-client发送文件**

flume-ng avro-client -c . -H 192.168.68.129 -p 4141 -F /usr/logs/log.10

**#在启动的终端查看console输出**

2013-05-27 01:11:45,852 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{} body: 68 65 6C 6C 6F 20 77 6F 72 6C 64 hello world }

## 案例3：Test Exec Source

**#文件名：**case3\_exec.conf

**#配置内容：**

# 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 = cat /flume/logs/log.10

a1.sources.r1.channels = c1

# Describe the sink

a1.sinks.k1.type = logger

 a1.sinks.k1.channel = c1

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

**#启动flume agent a1**

flume-ng agent -c . -f case3\_exec.conf -n a1 -Dflume.root.logger=INFO,console

**#追加内容到文件**

echo "exec test" >> /usr/logs/log.10

**#在启动的终端查看console输出**

2013-05-27 01:50:12,825 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{} body: 68 65 6C 6C 6F 20 77 6F 72 6C 64 hello world }

2013-05-27 01:50:12,826 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{} body: 65 78 65 63 20 74 65 73 74 exec test }

**#如果要使用tail命令，必选使得file足够大才能看到输出内容**

a1.sources.r1.command = tail -F /usr/logs/log.10

**#生成足够多的内容在文件里**

for i in {1..100};do echo "exec test$i" >> /usr/logs/log.10;echo $i;done

**#可以在console看到output**

2013-05-27 19:17:18,157 (lifecycleSupervisor-1-1) [INFO - org.apache.flume.source.ExecSource.start(ExecSource.java:155)] Exec source starting with command:tail -n 5 -F /usr/logs/log.10

2013-05-27 19:19:50,334 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{} body: 65 78 65 63 20 74 65 73 74 37 exec test7 }

## 案例4：Test spool Source

**#文件名：**case4\_spool.conf

**监测配置的目录下新增的文件(实时监控)**

**#配置内容：**

# Name the components on this agent

a1.sources = r1

a1.sinks = k1

a1.channels = c1

# Describe/configure the source

a1.sources.r1.type = spooldir

a1.sources.r1.spoolDir = /flume/tmp/logs

a1.sources.r1.fileHeader = true

a1.sources.r1.channels = c1

# Describe the sink

a1.sinks.k1.type = logger

 a1.sinks.k1.channel = c1

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

**#启动flume agent a1**

flume-ng agent -c . -f case4\_spool.conf -n a1 -Dflume.root.logger=INFO,console

**#追加内容到spool目录**

echo "spool test1" > /usr/logs/flumeSpool/spool1.log

**#在启动的终端查看console输出**

2013-05-27 22:49:06,098 (pool-4-thread-1) [INFO - org.apache.flume.client.avro.SpoolingFileLineReader.retireCurrentFile(SpoolingFileLineReader.java:229)] Preparing to move file /usr/logs/flumeSpool/spool1.log to /usr/logs/flumeSpool/spool1.log.COMPLETED

2013-05-27 22:49:06,101 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{file=/usr/logs/flumeSpool/spool1.log} body: 73 70 6F 6F 6C 20 74 65 73 74 31 spool test1 }

## 案例5：Test Syslog tcp source

**#文件名：**case5\_syslog.conf

**#配置内容：**

# Name the components on this agent

a1.sources = r1

a1.sinks = k1

a1.channels = c1

# Describe/configure the source

a1.sources.r1.type = syslogtcp

a1.sources.r1.port = 5140

a1.sources.r1.host = 192.168.68.129

a1.sources.r1.channels = c1

# Describe the sink

a1.sinks.k1.type = logger

 a1.sinks.k1.channel = c1

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

**#启动flume agent a1**

flume-ng agent -c . -f case5\_syslog.conf -n a1 -Dflume.root.logger=INFO,console

**#测试产生syslog, <37>因为需要wire format数据，否则会报错” Failed to extract syslog wire entry”**

echo "<37>hello via syslog" | nc localhost 5140

**#在启动的终端查看console输出**

2013-05-27 23:39:10,755 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{Severity=5, Facility=4} body: 68 65 6C 6C 6F 20 76 69 61 20 73 79 73 6C 6F 67 hello via syslog }

## 案例6：Test Syslog udp source

**#文件名：**case6\_syslogudp.conf

**#配置内容：**

# Name the components on this agent

a1.sources = r1

a1.sinks = k1

a1.channels = c1

# Describe/configure the source

a1.sources.r1.type = syslogudp

a1.sources.r1.port = 5140

a1.sources.r1.host = localhost

a1.sources.r1.channels = c1

# Describe the sink

a1.sinks.k1.type = logger

 a1.sinks.k1.channel = c1

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

**#启动flume agent a1**

flume-ng agent -c . -f case6\_syslogudp.conf -n a1 -Dflume.root.logger=INFO,console

**#测试产生syslog**

echo "<37>hello via syslogudp" | nc -u localhost 5140

**#在启动的终端查看console输出**

2013-05-27 23:39:10,755 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{Severity=5, Facility=4} body: 68 65 6C 6C 6F 20 76 69 61 20 73 79 73 6C 6F 67 hello via syslogudp }

## 案例7：Test HTTP source JSONHandler

**#文件名：**case7\_httppost.conf

**#配置内容：**

# Name the components on this agent

a1.sources = r1

a1.sinks = k1

a1.channels = c1

# Describe/configure the source

a1.sources.r1.type = org.apache.flume.source.http.HTTPSource

a1.sources.r1.port = 5140

a1.sources.r1.channels = c1

# Describe the sink

a1.sinks.k1.type = logger

 a1.sinks.k1.channel = c1

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

**#启动flume agent a1**

flume-ng agent -c . -f case7\_httppost.conf -n a1 -Dflume.root.logger=INFO,console

**#生成JSON 格式的POST request**

curl -X POST -d '[{ "headers" :{"namenode" : "namenode.example.com","datanode" : "random\_datanode.example.com"},"body" : "really\_random\_body"}]' http://localhost:5140

**#在启动的终端查看console输出**

2013-05-28 01:17:47,186 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{namenode=namenode.example.com, datanode=random\_datanode.example.com} body: 72 65 61 6C 6C 79 5F 72 61 6E 64 6F 6D 5F 62 6F random\_datanode\_body

## 案例8：Test HDFS Sink

**#文件名：**case8\_hdfs.conf

在/usr/local/apache-flume-1.3.1-bin/conf/flume-env.sh加入

export HADOOP\_HOME=/usr/local/hadoop

**#配置内容：**

# Name the components on this agent

a1.sources = r1

a1.sinks = k1

a1.channels = c1

# Describe/configure the source

a1.sources.r1.type = syslogtcp

a1.sources.r1.bind = 192.168.68.129

a1.sources.r1.port = 5140

a1.sources.r1.channels = c1

# Describe the sink

a1.sinks.k1.type = hdfs

a1.sinks.k1.channel = c1

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

a1.sinks.k1.hdfs.filePrefix = Syslog

a1.sinks.k1.hdfs.round = true

a1.sinks.k1.hdfs.roundValue = 10

a1.sinks.k1.hdfs.roundUnit = minute

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

**#启动flume agent a1**

flume-ng agent -c . -f case8\_hdfs.conf -n a1 -Dflume.root.logger=INFO,console

**#测试产生syslog**

echo "<37>hello via syslog to hdfs testing one" | nc 192.168.68.129 5140

**#在启动的终端查看console输出,文件生成成功**

2013-05-29 00:53:58,078 (hdfs-k1-call-runner-0) [INFO - org.apache.flume.sink.hdfs.BucketWriter.doOpen(BucketWriter.java:208)] Creating hdfs://master:9000/user/hadoop/flume/collected//Syslog.1369814037714.tmp

2013-05-29 00:54:28,220 (hdfs-k1-roll-timer-0) [INFO - org.apache.flume.sink.hdfs.BucketWriter.renameBucket(BucketWriter.java:427)] Renaming hdfs://master:9000/user/hadoop/flume/collected/Syslog.1369814037714.tmp to hdfs://master:9000/user/hadoop/flume/collected/Syslog.1369814037714

**#在hadoop上查看文件**

./hadoop dfs -cat hdfs://172.25.4.35:9000/user/hadoop/flume/collected/Syslog.1369814037714

SEQ!org.apache.hadoop.io.LongWritable"org.apache.hadoop.io.BytesWritable^;>Gv$hello via syslog to hdfs testing one

## 案例9：Test HDFS Sink

**#文件名：**case9\_hdfs.conf

在/usr/local/apache-flume-1.3.1-bin/conf/flume-env.sh加入

export HADOOP\_HOME=/usr/local/hadoop

**#配置内容：**

# Name the components on this agent

a1.sources = r1

a1.sinks = k1

a1.channels = c1

# Describe/configure the source

a1.sources.r1.type = org.apache.flume.source.http.HTTPSource

a1.sources.r1.bind = 0.0.0.0

a1.sources.r1.port = 5140

a1.sources.r1.channels = c1

# Describe the sink

a1.sinks.k1.type = hdfs

a1.sinks.k1.channel = c1

a1.sinks.k1.hdfs.path= hdfs://master:9000/user/hadoop/flume/collected/%y-%m-%d/%H%M/%S

a1.sinks.k1.hdfs.filePrefix = Syslog.%{host}

a1.sinks.k1.hdfs.round = true

a1.sinks.k1.hdfs.roundValue = 10

a1.sinks.k1.hdfs.roundUnit = minute

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

**#启动flume agent a1**

flume-ng agent -c . -f case9\_hdfs.conf -n a1 -Dflume.root.logger=INFO,console

**#生成JSON 格式的POST request**, header的timestamp 参数如果格式不对则无法解析

需要生成13为的timestamp才能解析出正确的时间,包含MilliSec

#linux生成当前时间10位Unix timestamp

date +%s

#linux生成当前时间13位Unix timestamp

date +%s%N|awk '{print substr($0,1,13)}'

curl -X POST -d '[{ "headers":{"timestamp":"1369818213654","host":"cc-staging-loginmgr2"},"body": "hello via post"}]' http://localhost:5140

**#在启动的终端查看console输出,文件生成成功**

2013-05-29 02:03:38,646 (hdfs-k1-call-runner-4) [INFO - org.apache.flume.sink.hdfs.BucketWriter.doOpen(BucketWriter.java:208)] Creating hdfs://master:9000/user/hadoop/flume/collected/2013-05-29/0203/cc-staging-loginmgr2..1369818218614.tmp

2013-05-29 02:04:08,714 (hdfs-k1-roll-timer-0) [INFO - org.apache.flume.sink.hdfs.BucketWriter.renameBucket(BucketWriter.java:427)] Renaming hdfs://master:9000/user/hadoop/flume/collected/2013-05-29/0203/cc-staging-loginmgr2..1369818218614.tmp to hdfs://master:9000/user/hadoop/flume/collected/2013-05-29/0203/cc-staging-loginmgr2..1369818218614

**#在hadoop上查看文件**

./hadoop dfs -ls hdfs://172.25.4.35:9000/user/hadoop/flume/collected/2013-05-29/0203

Found 1 items

-rw-r--r-- 3 root supergroup 129 2013-05-29 02:04 /user/hadoop/flume/collected/2013-05-29/0203/cc-staging-loginmgr2..1369818218614

## 案例10：Test Avro Sink

**#文件名：**case10\_avro.conf、case10\_avro\_sink.conf

**#配置内容：**

#case10\_avro.conf

# Name the components on this agent

a1.sources = r1

a1.sinks = k1

a1.channels = c1

# Describe/configure the source

a1.sources.r1.type = avro

a1.sources.r1.channels = c1

a1.sources.r1.bind = 0.0.0.0

a1.sources.r1.port = 4545

# Describe the sink

a1.sinks.k1.type = logger

 a1.sinks.k1.channel = c1

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

#case10\_avro\_sink.conf

# Name the components on this agent

a2.sources = r1

a2.sinks = k1

a2.channels = c1

# Describe/configure the source

a2.sources.r1.type = syslogtcp

a2.sources.r1.port = 5140

a2.sources.r1.host = localhost

a2.sources.r1.channels = c1

# Describe the sink

a2.sinks.k1.type = avro

a2.sinks.k1.channel = c1

a2.sinks.k1.hostname = 172.25.4.23

a2.sinks.k1.port = 4545

# Use a channel which buffers events in memory

a2.channels.c1.type = memory

a2.channels.c1.capacity = 1000

a2.channels.c1.transactionCapacity = 100

**#先启动Avro的Source,监听端口**

flume-ng agent -c . -f case10\_avro.conf -n a1 -Dflume.root.logger=INFO,console

**#再启动Avro的Sink**

flume-ng agent -c . -f case10\_avro\_sink.conf -n a2 -Dflume.root.logger=INFO,console

#可以看到已经建立连接

2013-06-02 19:23:00,237 (pool-5-thread-1) [INFO - org.apache.avro.ipc.NettyServer$NettyServerAvroHandler.handleUpstream(NettyServer.java:171)] [id: 0x7a0e28bf, /172.25.4.32:14894 => /172.25.4.23:4545] CONNECTED: /172.25.4.32:14894

**#在Avro Sink上生成测试log**

echo "<37>hello via avro sink" | nc localhost 5140

**#在Avro Source上可以看到log已经生成**

2013-06-02 19:24:13,740 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{Severity=5, Facility=4} body: 68 65 6C 6C 6F 20 76 69 61 20 61 76 72 6F 20 73 hello via avro sink }

## 案例11：Test File Roll Sink

**#文件名：**case11\_fileroll.conf

**#配置内容：**

# Name the components on this agent

a1.sources = r1

a1.sinks = k1

a1.channels = c1

# Describe/configure the source

a1.sources.r1.type = syslogtcp

a1.sources.r1.port = 5140

a1.sources.r1.host = localhost

a1.sources.r1.channels = c1

# Describe the sink

a1.sinks.k1.type = file\_roll

a1.sinks.k1.channel = c1

a1.sinks.k1.sink.directory = /var/log/flume

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

**#启动file roll 配置文件**

flume-ng agent -c . -f case11\_fileroll.conf -n a1 -Dflume.root.logger=INFO,console

**#生成测试log**

echo "<37>hello via file roll" | nc localhost 5140

echo "<37>hello via file roll 2" | nc localhost 5140

**#查看/var/log/flume下是否生成文件,默认每30秒生成一个新文件**

-rw-r--r-- 1 root root 20 Jun 2 19:44 1370227443397-1

-rw-r--r-- 1 root root 0 Jun 2 19:44 1370227443397-2

-rw-r--r-- 1 root root 22 Jun 2 19:45 1370227443397-3

cat 1370227443397-1 1370227443397-3

hello via file roll

hello via file roll 2

## 案例12：Test Replicating Channel Selector

**#文件名：**case12\_replicate\_sink.conf、case12\_replicate\_s1.conf、case12\_replicate\_s2.conf

**#配置内容：**

#case12\_replicate\_sink.conf

#2个channel和2个sink的配置文件

# Name the components on this agent

a1.sources = r1

a1.sinks = k1 k2

a1.channels = c1 c2

# Describe/configure the source

a1.sources.r1.type = syslogtcp

a1.sources.r1.port = 5140

a1.sources.r1.host = localhost

a1.sources.r1.selector.type = replicating

a1.sources.r1.channels = c1 c2

# Describe the sink

a1.sinks.k1.type = avro

a1.sinks.k1.channel = c1

a1.sinks.k1.hostname = 172.25.4.23

a1.sinks.k1.port = 4545

a1.sinks.k2.type = avro

a1.sinks.k2.channel = c2

a1.sinks.k2.hostname = 172.25.4.33

a1.sinks.k2.port = 4545

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

a1.channels.c2.type = memory

a1.channels.c2.capacity = 1000

a1.channels.c2.transactionCapacity = 100

case12\_replicate\_s1.conf

# Name the components on this agent

a2.sources = r1

a2.sinks = k1

a2.channels = c1

# Describe/configure the source

a2.sources.r1.type = avro

a2.sources.r1.channels = c1

a2.sources.r1.bind = 172.25.4.23

a2.sources.r1.port = 4545

# Describe the sink

a2.sinks.k1.type = logger

 a2.sinks.k1.channel = c1

# Use a channel which buffers events in memory

a2.channels.c1.type = memory

a2.channels.c1.capacity = 1000

a2.channels.c1.transactionCapacity = 100

case12\_replicate\_s2.conf

# Name the components on this agent

a3.sources = r1

a3.sinks = k1

a3.channels = c1

# Describe/configure the source

a3.sources.r1.type = avro

a3.sources.r1.channels = c1

a3.sources.r1.bind = 172.25.4.33

a3.sources.r1.port = 4545

# Describe the sink

a3.sinks.k1.type = logger

 a3.sinks.k1.channel = c1

# Use a channel which buffers events in memory

a3.channels.c1.type = memory

a3.channels.c1.capacity = 1000

a3.channels.c1.transactionCapacity = 100

**#先启动Avro的Source,监听端口**

flume-ng agent -c . -f case12\_replicate\_s1.conf -n a2 -Dflume.root.logger=INFO,console

flume-ng agent -c . -f case12\_replicate\_s2.conf -n a3 -Dflume.root.logger=INFO,console

**#再启动Avro的Sink**

flume-ng agent -c . -f case12\_replicate\_sink.conf -n a1 -Dflume.root.logger=INFO,console

**#查看是否都建立了连接**

2013-06-04 00:01:53,467 (pool-5-thread-1) [INFO - org.apache.avro.ipc.NettyServer$NettyServerAvroHandler.handleUpstream(NettyServer.java:171)] [id: 0x122a0fad, /172.25.4.32:55518 => /172.25.4.23:4545] BOUND: /172.25.4.23:4545

2013-06-04 00:01:53,467 (pool-5-thread-1) [INFO - org.apache.avro.ipc.NettyServer$NettyServerAvroHandler.handleUpstream(NettyServer.java:171)] [id: 0x122a0fad, /172.25.4.32:55518 => /172.25.4.23:4545] CONNECTED: /172.25.4.32:55518

2013-06-04 00:01:53,773 (pool-5-thread-1) [INFO - org.apache.avro.ipc.NettyServer$NettyServerAvroHandler.handleUpstream(NettyServer.java:171)] [id: 0x021881a7, /172.25.4.32:23731 => /172.25.4.33:4545] BOUND: /172.25.4.33:4545

2013-06-04 00:01:53,773 (pool-5-thread-1) [INFO - org.apache.avro.ipc.NettyServer$NettyServerAvroHandler.handleUpstream(NettyServer.java:171)] [id: 0x021881a7, /172.25.4.32:23731 => /172.25.4.33:4545] CONNECTED: /172.25.4.32:23731

**#生成测试log**

echo "<37>hello via channel selector" | nc localhost 5140

#查看2个sink是否得到数据

2013-06-04 00:02:06,479 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{Severity=5, Facility=4} body: 68 65 6C 6C 6F 20 76 69 61 20 63 68 61 6E 6E 65 hello via channe }

2013-06-04 00:02:09,788 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{Severity=5, Facility=4} body: 68 65 6C 6C 6F 20 76 69 61 20 63 68 61 6E 6E 65 hello via channe }

## 案例13：Test Multiplexing Channel Selector

**#文件名：**case13\_multi\_sink.conf、case13\_multi\_s1.conf、case13\_multi\_s2.conf

**#配置内容：**

#2个channel和2个sink的配置文件

a1.sources = r1

a1.sinks = k1 k2

a1.channels = c1 c2

# Describe/configure the source

a1.sources.r1.type = org.apache.flume.source.http.HTTPSource

a1.sources.r1.port = 5140

a1.sources.r1.host = 0.0.0.0

a1.sources.r1.selector.type = multiplexing

a1.sources.r1.channels = c1 c2

a1.sources.r1.selector.header = state

a1.sources.r1.selector.mapping.CZ = c1

a1.sources.r1.selector.mapping.US = c2

a1.sources.r1.selector.default = c1

# Describe the sink

a1.sinks.k1.type = avro

a1.sinks.k1.channel = c1

a1.sinks.k1.hostname = 172.25.4.23

a1.sinks.k1.port = 4545

a1.sinks.k2.type = avro

a1.sinks.k2.channel = c2

a1.sinks.k2.hostname = 172.25.4.33

a1.sinks.k2.port = 4545

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

a1.channels.c2.type = memory

a1.channels.c2.capacity = 1000

a1.channels.c2.transactionCapacity = 100

#case13\_ multi \_s1.conf

# Name the components on this agent

a2.sources = r1

a2.sinks = k1

a2.channels = c1

# Describe/configure the source

a2.sources.r1.type = avro

a2.sources.r1.channels = c1

a2.sources.r1.bind = 172.25.4.23

a2.sources.r1.port = 4545

# Describe the sink

a2.sinks.k1.type = logger

 a2.sinks.k1.channel = c1

# Use a channel which buffers events in memory

a2.channels.c1.type = memory

a2.channels.c1.capacity = 1000

a2.channels.c1.transactionCapacity = 100

#case13\_ multi \_s2.conf

# Name the components on this agent

a3.sources = r1

a3.sinks = k1

a3.channels = c1

# Describe/configure the source

a3.sources.r1.type = avro

a3.sources.r1.channels = c1

a3.sources.r1.bind = 172.25.4.33

a3.sources.r1.port = 4545

# Describe the sink

a3.sinks.k1.type = logger

 a3.sinks.k1.channel = c1

# Use a channel which buffers events in memory

a3.channels.c1.type = memory

a3.channels.c1.capacity = 1000

a3.channels.c1.transactionCapacity = 100

**#先启动Avro的Source,监听端口**

flume-ng agent -c . -f case13\_multi\_s1.conf -n a2 -Dflume.root.logger=INFO,console

flume-ng agent -c . -f case13\_multi\_s2.conf -n a3 -Dflume.root.logger=INFO,console

**#再启动Avro的Sink**

flume-ng agent -c . -f case13\_multi\_sink.conf -n a1 -Dflume.root.logger=INFO,console

**#根据配置文件生成测试的header 为state的POST请求**

curl -X POST -d '[{ "headers" :{"state" : "CZ"},"body" : "TEST1"}]' http://master:5140

curl -X POST -d '[{ "headers" :{"state" : "US"},"body" : "TEST2"}]' http://master:5140

curl -X POST -d '[{ "headers" :{"state" : "SH"},"body" : "TEST3"}]' http://master:5140

**#查看2个sink得到数据是否和配置文件一致**

Sink1:

2013-06-04 23:45:35,296 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{state=CZ} body: 54 45 53 54 31 TEST1 }

2013-06-04 23:45:50,309 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{state=SH} body: 54 45 53 54 33 TEST3 }

Sink2:

2013-06-04 23:45:42,293 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{state=US} body: 54 45 53 54 32 TEST2 }

## 案例14：Test Failover Sink Processor

**#文件名：**case14\_failover\_sink.conf、case14\_failover\_s1.conf、case14\_failover\_s2.conf

**#配置内容：**

case14\_failover\_sink.conf

# Name the components on this agent

a1.sources = r1

a1.sinks = k1 k2

a1.channels = c1 c2

a1.sinkgroups = g1

a1.sinkgroups.g1.sinks = k1 k2

a1.sinkgroups.g1.processor.type = failover

a1.sinkgroups.g1.processor.priority.k1 = 5

a1.sinkgroups.g1.processor.priority.k2 = 10

a1.sinkgroups.g1.processor.maxpenalty = 10000

# Describe/configure the source

a1.sources.r1.type = syslogtcp

a1.sources.r1.port = 5140

a1.sources.r1.host = localhost

a1.sources.r1.selector.type = replicating

a1.sources.r1.channels = c1 c2

# Describe the sink

a1.sinks.k1.type = avro

a1.sinks.k1.channel = c1

a1.sinks.k1.hostname = 172.25.4.23

a1.sinks.k1.port = 4545

a1.sinks.k2.type = avro

a1.sinks.k2.channel = c2

a1.sinks.k2.hostname = 172.25.4.33

a1.sinks.k2.port = 4545

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

a1.channels.c2.type = memory

a1.channels.c2.capacity = 1000

a1.channels.c2.transactionCapacity = 100

case14\_ failover \_s1.conf

# Name the components on this agent

a2.sources = r1

a2.sinks = k1

a2.channels = c1

# Describe/configure the source

a2.sources.r1.type = avro

a2.sources.r1.channels = c1

a2.sources.r1.bind = 172.25.4.23

a2.sources.r1.port = 4545

# Describe the sink

a2.sinks.k1.type = logger

 a2.sinks.k1.channel = c1

# Use a channel which buffers events in memory

a2.channels.c1.type = memory

a2.channels.c1.capacity = 1000

a2.channels.c1.transactionCapacity = 100

case14\_ failover \_s2.conf

# Name the components on this agent

a3.sources = r1

a3.sinks = k1

a3.channels = c1

# Describe/configure the source

a3.sources.r1.type = avro

a3.sources.r1.channels = c1

a3.sources.r1.bind = 172.25.4.33

a3.sources.r1.port = 4545

# Describe the sink

a3.sinks.k1.type = logger

 a3.sinks.k1.channel = c1

# Use a channel which buffers events in memory

a3.channels.c1.type = memory

a3.channels.c1.capacity = 1000

a3.channels.c1.transactionCapacity = 100

**#先启动Avro的Source,监听端口**

flume-ng agent -c . -f case14\_failover\_s1.conf -n a2 -Dflume.root.logger=INFO,console

flume-ng agent -c . -f case14\_failover\_s2.conf -n a3 -Dflume.root.logger=INFO,console

**#再启动Avro的Sink**

flume-ng agent -c . -f case14\_failover\_sink.conf -n a1 -Dflume.root.logger=INFO,console

**#生成测试log**

echo "<37>test1 failover" | nc localhost 5140

**#在sink2上产生log，sink1由于优先级小，没有产生**

2013-06-05 00:10:51,194 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{Severity=5, Facility=4} body: 74 65 73 74 31 20 66 61 69 6C 6F 76 65 72 test1 failover }

**#主动关闭sink2,再次生成测试log**

echo "<37>test2 failover" | nc localhost 5140

**#在sink1上会同时生成test1和test2**

2013-06-05 00:11:14,312 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{Severity=5, Facility=4} body: 74 65 73 74 31 20 66 61 69 6C 6F 76 65 72 test1 failover }

2013-06-05 00:11:14,312 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{Severity=5, Facility=4} body: 74 65 73 74 32 20 66 61 69 6C 6F 76 65 72 test2 failover }

**#再次打开sink2,log会根据优先级再到sink2上**

echo "<37>test4 failover" | nc localhost 5140

echo "<37>test5 failover" | nc localhost 5140

2013-06-05 00:12:33,071 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{Severity=5, Facility=4} body: 74 65 73 74 34 20 66 61 69 6C 6F 76 65 72 test4 failover }

2013-06-05 00:12:55,088 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{Severity=5, Facility=4} body: 74 65 73 74 35 20 66 61 69 6C 6F 76 65 72 test5 failover }

## 案例15：Test Load balancing Sink Processor

**#文件名：**case15\_load\_sink.conf、case15\_load\_s1.conf、case15\_load\_s2.conf

**#配置内容：**

注:load balance type下必须指定同一个channel到不同的sinks,否则不生效

case15\_load\_sink.conf

# Name the components on this agent

a1.sources = r1

a1.sinks = k1 k2

a1.channels = c1

a1.sinkgroups = g1

a1.sinkgroups.g1.sinks = k1 k2

a1.sinkgroups.g1.processor.type = load\_balance

a1.sinkgroups.g1.processor.backoff = true

a1.sinkgroups.g1.processor.selector = round\_robin

# Describe/configure the source

a1.sources.r1.type = syslogtcp

a1.sources.r1.port = 5140

a1.sources.r1.host = localhost

a1.sources.r1.channels = c1

# Describe the sink

a1.sinks.k1.type = avro

a1.sinks.k1.channel = c1

a1.sinks.k1.hostname = 172.25.4.23

a1.sinks.k1.port = 4545

a1.sinks.k2.type = avro

a1.sinks.k2.channel = c1

a1.sinks.k2.hostname = 172.25.4.33

a1.sinks.k2.port = 4545

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

case15\_ load \_s1.conf

# Name the components on this agent

a2.sources = r1

a2.sinks = k1

a2.channels = c1

# Describe/configure the source

a2.sources.r1.type = avro

a2.sources.r1.channels = c1

a2.sources.r1.bind = 172.25.4.23

a2.sources.r1.port = 4545

# Describe the sink

a2.sinks.k1.type = logger

 a2.sinks.k1.channel = c1

# Use a channel which buffers events in memory

a2.channels.c1.type = memory

a2.channels.c1.capacity = 1000

a2.channels.c1.transactionCapacity = 100

case15\_ load \_s2.conf

# Name the components on this agent

a3.sources = r1

a3.sinks = k1

a3.channels = c1

# Describe/configure the source

a3.sources.r1.type = avro

a3.sources.r1.channels = c1

a3.sources.r1.bind = 172.25.4.33

a3.sources.r1.port = 4545

# Describe the sink

a3.sinks.k1.type = logger

 a3.sinks.k1.channel = c1

# Use a channel which buffers events in memory

a3.channels.c1.type = memory

a3.channels.c1.capacity = 1000

a3.channels.c1.transactionCapacity = 100

**#先启动Avro的Source,监听端口**

flume-ng agent -c . -f case15\_load\_s1.conf -n a2 -Dflume.root.logger=INFO,console

flume-ng agent -c . -f case15\_load\_s2.conf -n a3 -Dflume.root.logger=INFO,console

**#再启动Avro的Sink**

flume-ng agent -c . -f case15\_load\_sink.conf -n a1 -Dflume.root.logger=INFO,console

**#生成4个测试log**

echo "<37>test2 loadbalance" | nc 192.168.68.129 5140

echo "<37>test3 loadbalance" | nc 192.168.68.129 5140

echo "<37>test4 loadbalance" | nc 192.168.68.129 5140

echo "<37>test5 loadbalance" | nc 192.168.68.129 5140

**#查看sink输出结果是否为轮询模式**

Sink1:

2013-06-06 01:36:03,516 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{Severity=5, Facility=4} body: 74 65 73 74 32 20 6C 6F 61 64 62 61 6C 61 6E 63 test2 loadbalanc }

2013-06-06 01:36:09,769 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{Severity=5, Facility=4} body: 74 65 73 74 34 20 6C 6F 61 64 62 61 6C 61 6E 63 test4 loadbalanc }

Sink2:

2013-06-06 01:36:05,809 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{Severity=5, Facility=4} body: 74 65 73 74 33 20 6C 6F 61 64 62 61 6C 61 6E 63 test3 loadbalanc }

2013-06-06 01:36:37,057 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{Severity=5, Facility=4} body: 74 65 73 74 35 20 6C 6F 61 64 62 61 6C 61 6E 63 test5 loadbalanc }

## 案例16：Test Body Event Serializers

**#文件名：**case16\_body.conf

**#配置内容：**

# Name the components on this agent

a1.sources = r1

a1.sinks = k1

a1.channels = c1

a1.sources.r1.type = org.apache.flume.source.http.HTTPSource

a1.sources.r1.port = 5140

a1.sources.r1.host = localhost

a1.sources.r1.channels = c1

# Describe the sink

a1.sinks.k1.type = file\_roll

a1.sinks.k1.channel = c1

a1.sinks.k1.sink.directory = /var/log/flume

a1.sinks.k1.sink.serializer = text

a1.sinks.k1.sink.serializer.appendNewline = false

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

**#启动**

flume-ng agent -c . -f case16\_body.conf -n a1 -Dflume.root.logger=INFO,console

**#生成测试log**

curl -X POST -d '[{ "headers" :{"host":"cc-staging-loginmgr2"},"body" : "TEST1 BODY TEXT"}]' http://192.168.68.129:5140

curl -X POST -d '[{ "headers" :{"host":"cc-staging-loginmgr2"},"body" : "TEST2 BODY TEXT"}]' http://192.168.68.129:5140

curl -X POST -d '[{ "headers" :{"host":"cc-staging-loginmgr2"},"body" : "TEST3 BODY TEXT"}]' http://192.168.68.129:5140

**#查看file roll 文件中的文本内容**

cat /var/log/flume/1370675739270-1

TEST1 BODY TEXT

TEST2 BODY TEXT

TEST3 BODY TEXT

#Avro Event Serializer

Alias: avro\_event. This interceptor serializes Flume events into an Avro container file

把flume event变成avro 中包含的文件

## 案例17：Test Timestamp Interceptor

**#文件名：**case17\_timestamp\_hostname.conf

**#配置内容：**

# Name the components on this agent

a1.sources = r1

a1.sinks = k1

a1.channels = c1

# Describe/configure the source

a1.sources.r1.type = syslogtcp

a1.sources.r1.bind = 0.0.0.0

a1.sources.r1.port = 5140

a1.sources.r1.channels = c1

a1.sources.r1.interceptors = i1 i2

a1.sources.r1.interceptors.i1.preserveExisting = false

a1.sources.r1.interceptors.i1.type = timestamp

a1.sources.r1.interceptors.i2.type = host

a1.sources.r1.interceptors.i2.hostHeader = hostname

a1.sources.r1.interceptors.i2.useIP = false

# Describe the sink

a1.sinks.k1.type = hdfs

a1.sinks.k1.channel = c1

a1.sinks.k1.hdfs.path = hdfs://master:9000/user/[Hadoop](http://www.linuxidc.com/topicnews.aspx?tid=13)/flume/collected/%Y-%m-%d/%H%M

a1.sinks.k1.hdfs.filePrefix = %{hostname}.

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

**#启动agent**

flume-ng agent -c . -f case17\_timestamp\_hostname.conf -n a1 -Dflume.root.logger=INFO,console

**#生成测试log**

echo "<37>test dynamic interceptor" | nc localhost 5140

**#查看hdfs生成的文件,可以看到timestamp和hostname都已经生成在header里面,可以根据自定义的格式生成文件夹**

./hadoop dfs -ls hdfs://172.25.4.35:9000/user/hadoop/flume/collected/2013-06-16/2331/

Found 1 items

-rw-r--r-- 3 root supergroup 231 2017-10-03 10:27 /flume/logs/2017-10-03/1027/master..1506997643385

## 案例18：Test static Interceptor

**#文件名：**case18\_static.conf

**#配置内容：**

# Name the components on this agent

a1.sources = r1

a1.sinks = k1

a1.channels = c1

# Describe/configure the source

a1.sources.r1.type = syslogtcp

a1.sources.r1.port = 5140

a1.sources.r1.host = localhost

a1.sources.r1.channels = c1

a1.sources.r1.interceptors = i1

a1.sources.r1.interceptors.i1.type = static

a1.sources.r1.interceptors.i1.key = datacenter

a1.sources.r1.interceptors.i1.value = NEW\_YORK

# 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

**#启动agent**

flume-ng agent -c . -f case18\_static.conf -n a1 -Dflume.root.logger=INFO,console

**#生成测试log**

echo "<37>test1 static interceptor" | nc master 5140

**#查看console输出结果**

2013-06-17 00:15:38,453 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - org.apache.flume.sink.LoggerSink.process(LoggerSink.java:70)] Event: { headers:{Severity=5, Facility=4, datacenter=NEW\_YORK} body: 74 65 73 74 31 20 73 74 61 74 69 63 20 69 6E 74 test1 static int }