

脚本之家

www.jb51.net

买了!

买一送一千万购物卡

更有1折秒杀, 100%中奖惊喜!

立即购买

12核 / 16G / 2T

百兆独享 · 猛戳我

网页制作

网络编程

脚本专栏

脚本下载

数据库

CMS教程

电子书籍

平面设计

媒体动画

操作系统

网站运营

网络安全

在线手册

星外虚拟主机 华众虚拟主机 Linux win服务器 FTP服务器 DNS服务器 Tomcat nginx zabbix 云和虚拟化 服务器其它

您的位置: 首页 → 网站技巧 → 服务器 → Linux → 正文内容 Flume环境部署和配置详解及案例大全

请输入关键词

搜索

低价出售流量10000IP只需8元

阿里云免费虚拟机+日志分析仅6元

16核16G独服百兆带宽899元/月|创梦网络

美国云20元/香港云29元/韩国云55元

装软件赚钱, 每台电脑秒赚6.3元

《无视CC/DDOS 双线150G单机 899/月》

微子网络 湛江双线高防 抗15G 700元起

中原地区最大IDC数据中心, 首月托管免费

众生网络 基于云计算的互联网基础服务运

服务器租用/托管-域名空间/认准腾佑科技

BGP四线 亿恩1U服务器托管3999元/年

九九数据 — 工信部认可正规资质IDC接入商

枫信科技-江苏双线10M保证-399/元

港湾云主机, 稳定、安全才是硬道理

免备vps20/百独799/双线350/45互联

韩国香港美国站群服务器 巨牛网络

鼎点网络百兆独享服务器仅需999元

[香港双高防]无视CC★DDOS/堪比广东!

畅游网络 百独服务器 包跑满 998元

服务器租用 199元起

8核服务器298元/月 45/月云主机

群英云服务器送10M带宽30G防御,49元起

免费空间

【1000UV流量只要8毛】还有点流量

西部数码

云主机

中国云主机领导品牌

仅需48元

DDoS高防IP 300G无限防

453 Gbps真实防御经验, 支持云外

高防云

158月/月起!

320G 立体防御

免备案云主机

阿里云

30+云计算产品 免费 6 个月

云计算应用更轻松

免费上云

Flume环境部署和配置详解及案例大全

投稿: hebedich 字体: [增加 减小] 类型: 转载 时间: 2014-08-11 我要评论

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。

黑码程序员

手机排行榜

个人医疗保险

打鱼游戏

软件工程师

软件工程师待遇

成人高考报名

拼接屏

手机app开发

ssd固态硬盘

程序员有前途吗

燃气热水器

一、什么是Flume?

flume 作为 cloudera 开发的实时日志收集系统, 受到了业界的认可与广泛应用。Flume 初始的发行版本目前被称为 Flume OG (original generation), 属于 cloudera。但随着 FLume 功能的扩展, Flume OG 代码工程臃肿、核心组件设计不合理、核心配置不标准等缺点暴露出来, 尤其是在 Flume OG 的最后一个发行版本 0.94.0 中, 日志传输不稳定的现象尤为严重, 为了解决这些问题, 2011 年 10 月 22 号, cloudera 完成了 Flume-728, 对 Flume 进行了里程碑式的改动: 重构核心组件、核心配置以及代码架构, 重构后的版本统称为 Flume NG (next generation); 改动的另一原因是将 Flume 纳入 apache 旗下, cloudera Flume 改名为 Apache Flume。

flume的特点:

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。

flume的数据流由事件(Event)贯穿始终。事件是Flume的基本数据单位, 它携带日志数据(字节数组形式)并且携带有头信息, 这些Event由Agent外部的Source生成, 当Source捕获事件后会进行特定的格式化, 然后Source会把事件推入(单个或多个)Channel中。你可以把Channel看作是一个缓冲区, 它将保存事件直到Sink处理完该事件。Sink负责持久化日志或者把事件推向另一个Source。

flume的可靠性

当节点出现故障时, 日志能够被传送到其他节点上而不会丢失。Flume提供了三种级别的可靠性保障, 从强到弱依次分别为: end-to-end (收到数据agent首先将event写到磁盘上, 当数据传送成功后, 再删除;

大家感兴趣的内容

1 apache开启.htaccess及.htaccess

2 Apache Rewrite url重定向功能的

3 CentOS 6.4安装配置LAMP服务器(A

4 CentOS+Nginx+PHP+MySQL详细配置

5 linux下用cron定时执行任务的方法

6 Apache 虚拟目录和默认首页的设置

7 Service Temporarily Unavailabl

8 apache性能测试工具ab使用详解

9 Apache 多端口多站点配置方法

10 apache启动报错: the requested

BRITISH AIRWAYS

英国航空公司

享受知名品牌特价和英航旅客独家折扣

http://www.jb51.net/article/53542.htm

1/15

如果数据发送失败，可以重新发送。），Store on failure（这也是scribe采用的策略，当数据接收方crash时，将数据写到本地，待恢复后，继续发送），Besteffort（数据发送到接收方后，不会进行确认）。

flume的可恢复性：

还是靠Channel。推荐使用FileChannel，事件持久化在本地文件系统里(性能较差)。

flume的一些核心概念：

Agent使用JVM 运行Flume。每台机器运行一个agent，但是可以在一个agent中包含多个sources和sinks。

Client生产数据，运行在一个独立的线程。

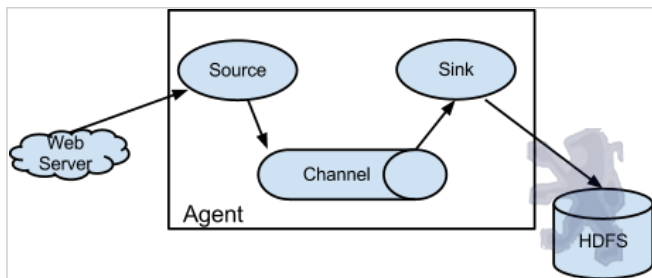
Source从Client收集数据，传递给Channel。

Sink从Channel收集数据，运行在一个独立线程。

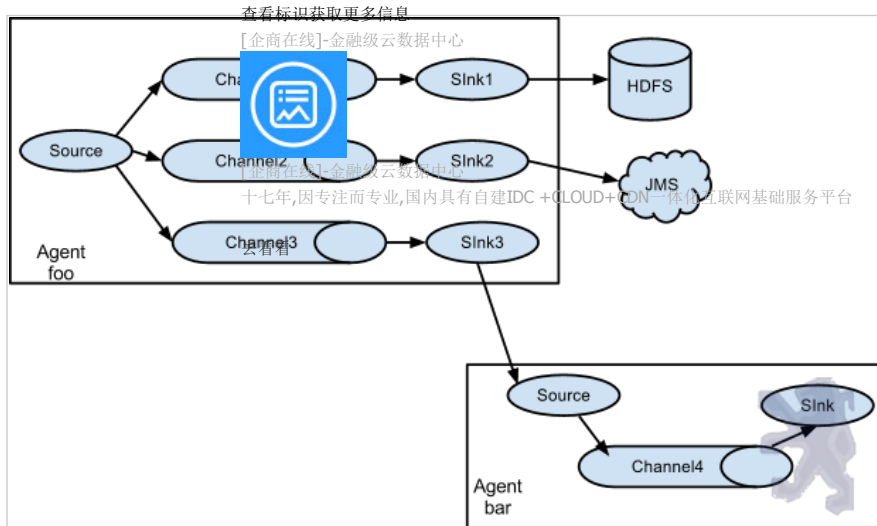
Channel连接 sources 和 sinks，这个有点像一个队列。

Events可以是日志记录、avro 对象等。

Flume以agent为最小的独立运行单位。一个agent就是一个JVM。单agent由Source、Sink和Channel三大组件构成，如下图：



值得注意的是，Flume提供了大量内置的Source、Channel和Sink类型。不同类型的Source,Channel和Sink可以自由组合。组合方式基于用户设置的配置文件，非常灵活。比如：Channel可以把事件暂存在内存里，也可以持久化到本地硬盘上。Sink可以把日志写入HDFS, HBase，甚至是另外一个Source等等。Flume支持用户建立多级流，也就是说，多个agent可以协同工作，并且支持Fan-in、Fan-out、Contextual Routing、Backup Routes，这也正是NB之处。如下图所示：



二、flume的官方网站在哪里？

<http://flume.apache.org/>

三、在哪里下载？

<http://www.apache.org/dyn/closer.cgi/flume/1.5.0/apache-flume-1.5.0-bin.tar.gz>

四、如何安装？

1)将下载的flume包，解压到/home/hadoop目录中，你就已经完成了50%：）简单吧

2)修改 flume-env.sh 配置文件,主要是JAVA\_HOME变量设置

```
1 root@m1:/home/hadoop/flume-1.5.0-bin# cp conf/flume-env.sh.template conf/flume-env.sh
2 root@m1:/home/hadoop/flume-1.5.0-bin# vi conf/flume-env.sh
3 # Licensed to the Apache Software Foundation (ASF) under one
4 # or more contributor license agreements. See the NOTICE file
5 # distributed with this work for additional information
6 # regarding copyright ownership. The ASF licenses this file
7 # to you under the Apache License, Version 2.0 (the
8 # "License"); you may not use this file except in compliance
```

阿里云

高性能云服务器就选阿里云

最低5折抢

## 最近更新的内容

详解Linux多线程使用信号量同步

centos6.5下svn的使用说明

centos6利用yum安装php mysql gd的步骤

CentOS 6.1 环境中部署nginx、php（包括用vnc实现Windows远程连接linux桌面之方法）

Tomcat无法加载css和js等静态资源文件的解决方法

apache中封IP的实现方法

LNAMP架构中后端Apache获取用户真实IP的方法

CentOS 5 上安装yum的方法

linux下的dhcp服务的完全配置(图文详解)

众生网络

品牌服务器租用

集思网络

VPS主机

枫信科技

IDC服务商

## 常用在线小工具

在线JS脚本校验器错误

CSS代码工具

歇后语在线查询

RGB颜色查询对照表 配色工具

在线XML格式化/压缩工具

sql代码在线格式化美化工具

在线XML/JSON互相转换工具

JavaScript压缩/格式化/加密工具

在线生成二维码工具(加强版)

php代码在线格式化美化工具

腾佑科技  
tuidc.com

租用 返3000

丽水双线托管4999元

超低价 敢买就敢送

2核，2G一年只要899元！

```

9 # with the License. You may obtain a copy of the License at
10 #
11 # http://www.apache.org/licenses/LICENSE-2.0
12 #
13 # Unless required by applicable law or agreed to in writing, software
14 # distributed under the License is distributed on an "AS IS" BASIS,
15 # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
16 # See the License for the specific language governing permissions and
17 # limitations under the License.
18
19 # If this file is placed at FLUME_CONF_DIR/flume-env.sh, it will be sourced
20 # during Flume startup.
21
22 # Enviroment variables can be set here.
23
24 JAVA_HOME=/usr/lib/jvm/java-7-oracle
25
26 # Give Flume more memory and pre-allocate, enable remote monitoring via JMX
27 #JAVA_OPTS="-Xms100m -Xmx200m -Dcom.sun.management.jmxremote"
28
29 # Note that the Flume conf directory is always included in the classpath.
30 #FLUME_CLASSPATH=""
31

```

### 3)验证是否安装成功

```

1 root@m1:/home/hadoop# /home/hadoop/flume-1.5.0-bin/bin/flume-ng version
2 Flume 1.5.0
3 Source code repository: https://git-wip-us.apache.org/repos/asf/flume.git
4 Revision: 8633220df808c4cd0c13d1cf0320454a94f1ea97
5 Compiled by hshreedharan on Wed May 7 14:49:18 PDT 2014
6 From source with checksum a01fe726e4380ba0c9f7a7d222db961f
7 root@m1:/home/hadoop#

```

出现上面的信息，表示安装成功了

## 五、flume的案例

### 1)案例1: Avro

Avro可以发送一个给定的文件给Flume。Avro 源使用AVRO RPC机制。

#### a)创建agent配置文件

```

1 root@m1:/home/hadoop#vi /home/hadoop/flume-1.5.0-bin/conf/avro.conf
2
3 a1.sources = r1
4 a1.sinks = k1
5 a1.channels = c1
6
7 # Describe/configure the source
8 a1.sources.r1.type = avro
9 a1.sources.r1.channels = c1
10 a1.sources.r1.bind = 0.0.0.0
11 a1.sources.r1.port = 4141
12
13 # Describe the sink
14 a1.sinks.k1.type = logger
15
16 # Use a channel which buffers events in memory
17 a1.channels.c1.type = memory
18 a1.channels.c1.capacity = 1000
19 a1.channels.c1.transactionCapacity = 100
20
21 # Bind the source and sink to the channel
22 a1.sources.r1.channels = c1
23 a1.sinks.k1.channel = c1

```

#### b)启动flume agent a1

```

1 root@m1:/home/hadoop# /home/hadoop/flume-1.5.0-bin/bin/flume-ng agent -c . -?

```

#### c)创建指定文件

```

1 root@m1:/home/hadoop# echo "hello world" > /home/hadoop/flume-1.5.0-bin/log.

```

#### d)使用avro-client发送文件

```

1 root@m1:/home/hadoop# /home/hadoop/flume-1.5.0-bin/bin/flume-ng avro-client

```

f)在m1的控制台，可以看到以下信息，注意最后一行：

```

1 root@m1:/home/hadoop/flume-1.5.0-bin/conf# /home/hadoop/flume-1.5.0-bin/bin/
2 Info: Sourcing environment configuration script /home/hadoop/flume-1.5.0-bin/

```



口红品牌排行榜



中年离异征婚



大型游戏排行榜

```

3 Info: Including Hadoop libraries found via (/home/hadoop/hadoop-2.2.0/bin/hac
4 Info: Excluding /home/hadoop/hadoop-2.2.0/share/hadoop/common/lib/slf4j-api-1
5 Info: Excluding /home/hadoop/hadoop-2.2.0/share/hadoop/common/lib/slf4j-log4j
6 ...
7 -08-10 10:43:25,112 (New I/O worker #1) [INFO - org.apache.avro.ipc.NettyServ
8 -08-10 10:43:25,112 (New I/O worker #1) [INFO - org.apache.avro.ipc.NettyServ
9 -08-10 10:43:25,112 (New I/O worker #1) [INFO - org.apache.avro.ipc.NettyServ
10 -08-10 10:43:26,718 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - c

```

## 2)案例2: Spool

Spool监测配置的目录下新增的文件,并将文件中的数据读取出来。需要注意两点:

- 1) 拷贝到spool目录下的文件不可以再打开编辑。
- 2) spool目录下不可包含相应的子目录

### a)创建agent配置文件

```

1 root@m1:/home/hadoop# vi /home/hadoop/flume-1.5.0-bin/conf/spool.conf
2 a1.sources = r1
3 a1.sinks = k1
4 a1.channels = c1
5 # Describe/configure the source
6 a1.sources.r1.type = spooldir
7 a1.sources.r1.channels = c1
8 a1.sources.r1.spoolDir = /home/hadoop/flume-1.5.0-bin/logs
9 a1.sources.r1.fileHeader = true
10 # Describe the sink
11 a1.sinks.k1.type = logger
12 # Use a channel which buffers events in memory
13 a1.channels.c1.type = memory
14 a1.channels.c1.capacity = 1000
15 a1.channels.c1.transactionCapacity = 100
16 # Bind the source and sink to the channel
17 a1.sources.r1.channels = c1
18 a1.sinks.k1.channel = c1

```

### b)启动flume agent a1

```

1 root@m1:/home/hadoop# /home/hadoop/flume-1.5.0-bin/bin/flume-ng agent -c . -?

```

### c)追加文件到/home/hadoop/flume-1.5.0-bin/logs目录

```

1 root@m1:/home/hadoop# echo "spool test1" > /home/hadoop/flume-1.5.0-bin/logs

```

### d)在m1的控制台, 可以看到以下相关信息:

```

1 /08/10 11:37:13 INFO source.SpoolDirectorySource: Spooling Directory Source
2 /08/10 11:37:13 INFO source.SpoolDirectorySource: Spooling Directory Source
3 /08/10 11:37:14 INFO avro.ReliableSpoolingFileEventReader: Preparing to move
4 /08/10 11:37:14 INFO source.SpoolDirectorySource: Spooling Directory Source
5 /08/10 11:37:14 INFO source.SpoolDirectorySource: Spooling Directory Source
6 /08/10 11:37:14 INFO sink.LoggerSink: Event: { headers:{file=/home/hadoop/flume-1.5.0-bin/logs/spool
7 /08/10 11:37:15 INFO source.SpoolDirectorySource: Spooling Directory Source
8 /08/10 11:37:15 INFO source.SpoolDirectorySource: Spooling Directory Source
9 /08/10 11:37:16 INFO source.SpoolDirectorySource: Spooling Directory Source
10 /08/10 11:37:16 INFO source.SpoolDirectorySource: Spooling Directory Source
11 /08/10 11:37:17 INFO source.SpoolDirectorySource: Spooling Directory Source

```

## 3)案例3: Exec

EXEC执行一个给定的命令获得输出的源,如果要使用tail命令, 必选使得file足够大才能看到输出内容

### a)创建agent配置文件

```

1 root@m1:/home/hadoop# vi /home/hadoop/flume-1.5.0-bin/conf/exec_tail.conf
2 a1.sources = r1
3 a1.sinks = k1
4 a1.channels = c1
5 # Describe/configure the source
6 a1.sources.r1.type = exec
7 a1.sources.r1.channels = c1
8 a1.sources.r1.command = tail -F /home/hadoop/flume-1.5.0-bin/log_exec_tail
9 # Describe the sink
10 a1.sinks.k1.type = logger
11 # Use a channel which buffers events in memory
12 a1.channels.c1.type = memory
13 a1.channels.c1.capacity = 1000
14 a1.channels.c1.transactionCapacity = 100
15 # Bind the source and sink to the channel
16 a1.sources.r1.channels = c1
17 a1.sinks.k1.channel = c1

```

### b)启动flume agent a1

```

1 root@m1:/home/hadoop# /home/hadoop/flume-1.5.0-bin/bin/flume-ng agent -c . -?

```



口红品牌排行榜



中年离异征婚



大型游戏排行榜



c)生成足够多的内容在文件里

```
1 root@m1:/home/hadoop# for i in {1..100};do echo "exec tail$i" >> /home/hadoop...
```

e)在m1的控制台，可以看到以下信息：

```
1 -08-10 10:59:25,513 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - ?
2 -08-10 10:59:34,535 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - c
3 -08-10 11:01:40,557 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - c
4 -08-10 11:01:41,180 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - c
5 -08-10 11:01:41,180 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - c
6 -08-10 11:01:41,181 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - c
7 -08-10 11:01:41,181 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - c
8 -08-10 11:01:41,181 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - c
9 ....
10 ....
11 ....
12 -08-10 11:01:51,550 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - c
13 -08-10 11:01:51,550 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - c
14 -08-10 11:01:51,551 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - c
15 -08-10 11:01:51,551 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - c
16 -08-10 11:01:51,551 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO - c
```

#### 4)案例4: Syslogtcp

Syslogtcp监听TCP的端口做为数据源

a)创建agent配置文件

```
1 root@m1:/home/hadoop# vi /home/hadoop/flume-1.5.0-bin/conf/syslog_tcp.conf ?
2 a1.sources = r1
3 a1.sinks = k1
4 a1.channels = c1
5 # Describe/configure the source
6 a1.sources.r1.type = syslogtcp
7 a1.sources.r1.port = 5140
8 a1.sources.r1.host = localhost
9 a1.sources.r1.channels = c1
10 # Describe the sink
11 a1.sinks.k1.type = logger
12 # Use a channel which buffers events in memory
13 a1.channels.c1.type = memory
14 a1.channels.c1.capacity = 1000
15 a1.channels.c1.transactionCapacity = 100
16 # Bind the source and sink to the channel
17 a1.sources.r1.channels = c1
18 a1.sinks.k1.channel = c1
```

b)启动flume agent a1

```
1 root@m1:/home/hadoop# /home/hadoop/flume-1.5.0-bin/bin/flume-ng agent -c . -?
```

c)测试产生syslog

```
1 root@m1:/home/hadoop# echo "hello idoall.org syslog" | nc localhost 5140 ?
```

d)在m1的控制台，可以看到以下信息：

```
1 /08/10 11:41:45 INFO node.PollingPropertiesFileConfigurationProvider: Reloa?
2 /08/10 11:41:45 INFO conf.FlumeConfiguration: Added sinks: k1 Agent: a1
3 /08/10 11:41:45 INFO conf.FlumeConfiguration: Processing:k1
4 /08/10 11:41:45 INFO conf.FlumeConfiguration: Processing:k1
5 /08/10 11:41:45 INFO conf.FlumeConfiguration: Post-validation flume configura
6 /08/10 11:41:45 INFO node.AbstractConfigurationProvider: Creating channels
7 /08/10 11:41:45 INFO channel.DefaultChannelFactory: Creating instance of chan
8 /08/10 11:41:45 INFO node.AbstractConfigurationProvider: Created channel c1
9 /08/10 11:41:45 INFO source.DefaultSourceFactory: Creating instance of source
10 /08/10 11:41:45 INFO sink.DefaultSinkFactory: Creating instance of sink: k1,
11 /08/10 11:41:45 INFO node.AbstractConfigurationProvider: Channel c1 connected
12 /08/10 11:41:45 INFO node.Application: Starting new configuration:{ sourceRun
13 /08/10 11:41:45 INFO node.Application: Starting Channel c1
14 /08/10 11:41:45 INFO instrumentation.MonitoredCounterGroup: Monitored counter
15 /08/10 11:41:45 INFO instrumentation.MonitoredCounterGroup: Component type: C
16 /08/10 11:41:45 INFO node.Application: Starting Sink k1
17 /08/10 11:41:45 INFO node.Application: Starting Source r1
18 /08/10 11:41:45 INFO source.SyslogTcpSource: Syslog TCP Source starting...
19 /08/10 11:42:15 WARN source.SyslogUtils: Event created from Invalid Syslog da
20 /08/10 11:42:15 INFO sink.LoggerSink: Event: { headers:{Severity=0, flume.sys
```

#### 5)案例5: JSONHandler

a)创建agent配置文件

```
1 root@m1:/home/hadoop# vi /home/hadoop/flume-1.5.0-bin/conf/post_json.conf ?
2 a1.sources = r1
3 a1.sinks = k1
```



口红品牌排行榜



中年离异征婚



大型游戏排行榜

```

4  a1.channels = c1
5  # Describe/configure the source
6  a1.sources.r1.type = org.apache.flume.source.http.HTTPSource
7  a1.sources.r1.port = 8888
8  a1.sources.r1.channels = c1
9  # Describe the sink
10 a1.sinks.k1.type = logger
11 # Use a channel which buffers events in memory
12 a1.channels.c1.type = memory
13 a1.channels.c1.capacity = 1000
14 a1.channels.c1.transactionCapacity = 100
15 # Bind the source and sink to the channel
16 a1.sources.r1.channels = c1
17 a1.sinks.k1.channel = c1

```

b)启动flume agent a1

```
1 root@m1:/home/hadoop# /home/hadoop/flume-1.5.0-bin/bin/flume-ng agent -c . -?
```

c)生成JSON 格式的POST request

```
1 root@m1:/home/hadoop# curl -X POST -d '{"headers":{"a": "a1","b": "b1"}>
```

d)在m1的控制台，可以看到以下信息：

```

1 08/10 11:49:59 INFO node.Application: Starting Channel c1
2 /08/10 11:49:59 INFO instrumentation.MonitoredCounterGroup: Monitored counter
3 /08/10 11:49:59 INFO instrumentation.MonitoredCounterGroup: Component type: C
4 /08/10 11:49:59 INFO node.Application: Starting Sink k1
5 /08/10 11:49:59 INFO node.Application: Starting Source r1
6 /08/10 11:49:59 INFO mortbay.log: Logging to org.slf4j.impl.Log4jLoggerAdapte
7 /08/10 11:49:59 INFO mortbay.log: jetty-6.1.26
8 /08/10 11:50:00 INFO mortbay.log: Started SelectChannelConnector@0.0.0.0:8888
9 /08/10 11:50:00 INFO instrumentation.MonitoredCounterGroup: Monitored counter
10 /08/10 11:50:00 INFO instrumentation.MonitoredCounterGroup: Component type: S
11 /08/10 12:14:32 INFO sink.LoggerSink: Event: { headers:{b=b1, a=a1} body: 69

```

#### 6)案例6: Hadoop sink

其中关于hadoop2.2.0部分的安装部署，请参考文章《ubuntu12.04+hadoop2.2.0+ookeeper3.4.

5+hbase0.96.2+hive0.13.1分布式环境部署》

a)创建agent配置文件

```

1 root@m1:/home/hadoop# vi /home/hadoop/flume-1.5.0-bin/conf/hdfs_sink.conf
2 a1.sources = r1
3 a1.sinks = k1
4 a1.channels = c1
5 # Describe/configure the source
6 a1.sources.r1.type = syslogtcp
7 a1.sources.r1.port = 5140
8 a1.sources.r1.host = localhost
9 a1.sources.r1.channels = c1
10 # Describe the sink
11 a1.sinks.k1.type = hdfs
12 a1.sinks.k1.channel = c1
13 a1.sinks.k1.hdfs.path = hdfs://m1:9000/user/flume/syslogtcp
14 a1.sinks.k1.hdfs.filePrefix = Syslog
15 a1.sinks.k1.hdfs.round = true
16 a1.sinks.k1.hdfs.roundValue = 10
17 a1.sinks.k1.hdfs.roundUnit = minute
18 # Use a channel which buffers events in memory
19 a1.channels.c1.type = memory
20 a1.channels.c1.capacity = 1000
21 a1.channels.c1.transactionCapacity = 100
22 # Bind the source and sink to the channel
23 a1.sources.r1.channels = c1
24 a1.sinks.k1.channel = c1

```

b)启动flume agent a1

```
1 root@m1:/home/hadoop# /home/hadoop/flume-1.5.0-bin/bin/flume-ng agent -c . -?
```

c)测试产生syslog

```
1 root@m1:/home/hadoop# echo "hello idoall flume -> hadoop testing one" | nc 127.0.0.1 5140
```

d)在m1的控制台，可以看到以下信息：

```

1 /08/10 12:20:39 INFO instrumentation.MonitoredCounterGroup: Monitored count?
2 /08/10 12:20:39 INFO instrumentation.MonitoredCounterGroup: Component type: C
3 /08/10 12:20:39 INFO node.Application: Starting Sink k1

```



口红品牌排行榜



中年离异征婚



大型游戏排行榜

```

4 /08/10 12:20:39 INFO node.Application: Starting Source r1
5 /08/10 12:20:39 INFO instrumentation.MonitoredCounterGroup: Monitored counter
6 /08/10 12:20:39 INFO instrumentation.MonitoredCounterGroup: Component type: S
7 /08/10 12:20:39 INFO source.SyslogTcpSource: Syslog TCP Source starting...
8 /08/10 12:21:46 WARN source.SyslogUtils: Event created from Invalid Syslog da
9 /08/10 12:21:49 INFO hdfs.HDFSSequenceFile: writeFormat = Writable, UseRawLoc
10 /08/10 12:21:49 INFO hdfs.BucketWriter: Creating hdfs://m1:9000/user/flume/sy
11 /08/10 12:22:20 INFO hdfs.BucketWriter: Closing hdfs://m1:9000/user/flume/sy
12 /08/10 12:22:20 INFO hdfs.BucketWriter: Close tries incremented
13 /08/10 12:22:20 INFO hdfs.BucketWriter: Renaming hdfs://m1:9000/user/flume/sy
14 /08/10 12:22:20 INFO hdfs.HDFSEventSink: Writer callback called.

```

e)在m1上再打开一个窗口，去hadoop上检查文件是否生成

```

1 root@m1:/home/hadoop# /home/hadoop/hadoop-2.2.0/bin/hadoop fs -ls /user/flume/
2 Found 1 items
3 -rw-r--r-- 3 root supergroup 155 2014-08-10 12:22 /user/flume/syslogtcp/Sy
4 root@m1:/home/hadoop# /home/hadoop/hadoop-2.2.0/bin/hadoop fs -cat /user/flume
5 SEQ!org.apache.hadoop.io.LongWritable"org.apache.hadoop.io.BytesWritable^;>Gv$

```

## 7)案例7: File Roll Sink

a)创建agent配置文件

```

1 root@m1:/home/hadoop# vi /home/hadoop/flume-1.5.0-bin/conf/file_roll.conf
2 a1.sources = r1
3 a1.sinks = k1
4 a1.channels = c1
5 # Describe/configure the source
6 a1.sources.r1.type = syslogtcp
7 a1.sources.r1.port = 5555
8 a1.sources.r1.host = localhost
9 a1.sources.r1.channels = c1
10 # Describe the sink
11 a1.sinks.k1.type = file_roll
12 a1.sinks.k1.sink.directory = /home/hadoop/flume-1.5.0-bin/logs
13 # Use a channel which buffers events in memory
14 a1.channels.c1.type = memory
15 a1.channels.c1.capacity = 1000
16 a1.channels.c1.transactionCapacity = 100
17 # Bind the source and sink to the channel
18 a1.sources.r1.channels = c1
19 a1.sinks.k1.channel = c1

```

b)启动flume agent a1

```

1 root@m1:/home/hadoop# /home/hadoop/flume-1.5.0-bin/bin/flume-ng agent -c . -?

```

c)测试产生log

```

1 root@m1:/home/hadoop# echo "hello idoall.org syslog" | nc localhost 5555
2 root@m1:/home/hadoop# echo "hello idoall.org syslog 2" | nc localhost 5555

```

d)查看/home/hadoop/flume-1.5.0-bin/logs下是否生成文件,默认每30秒生成一个新文件

```

1 root@m1:/home/hadoop# ll /home/hadoop/flume-1.5.0-bin/logs
2 总用量 272
3 drwxr-xr-x 3 root root 4096 Aug 10 12:50 ./
4 drwxr-xr-x 9 root root 4096 Aug 10 10:59 ../
5 -rw-r--r-- 1 root root 50 Aug 10 12:49 1407646164782-1
6 -rw-r--r-- 1 root root 0 Aug 10 12:49 1407646164782-2
7 -rw-r--r-- 1 root root 0 Aug 10 12:50 1407646164782-3
8 root@m1:/home/hadoop# cat /home/hadoop/flume-1.5.0-bin/logs/1407646164782-1 /
9 hello idoall.org syslog
10 hello idoall.org syslog 2

```

## 8)案例8: Replicating Channel Selector

Flume支持Fan out流从一个源到多个通道。有两种模式的Fan out，分别是复制和复用。在复制的情况下，流的事件被发送到所有的配置通道。在复用的情况下，事件被发送到可用的渠道中的一个子集。Fan out流需要指定源和Fan out通道的规则。

这次我们需要用到m1,m2两台机器

a)在m1创建replicating\_Channel\_Selector配置文件

```

1 root@m1:/home/hadoop# vi /home/hadoop/flume-1.5.0-bin/conf/replicating_Chann
2 a1.sources = r1
3 a1.sinks = k1 k2
4 a1.channels = c1 c2
5 # Describe/configure the source
6 a1.sources.r1.type = syslogtcp
7 a1.sources.r1.port = 5140
8 a1.sources.r1.host = localhost
9 a1.sources.r1.channels = c1 c2
10 a1.sources.r1.selector.type = replicating

```



口红品牌排行榜



中年离异征婚



大型游戏排行榜

```

11 # Describe the sink
12 a1.sinks.k1.type = avro
13 a1.sinks.k1.channel = c1
14 a1.sinks.k1.hostname = m1
15 a1.sinks.k1.port = 5555
16 a1.sinks.k2.type = avro
17 a1.sinks.k2.channel = c2
18 a1.sinks.k2.hostname = m2
19 a1.sinks.k2.port = 5555
20 # Use a channel which buffers events in memory
21 a1.channels.c1.type = memory
22 a1.channels.c1.capacity = 1000
23 a1.channels.c1.transactionCapacity = 100
24 a1.channels.c2.type = memory
25 a1.channels.c2.capacity = 1000
26 a1.channels.c2.transactionCapacity = 100

```

b)在m1创建replicating\_Channel\_Selector\_avro配置文件

```

1 root@m1:/home/hadoop# vi /home/hadoop/flume-1.5.0-bin/conf/replicating_Channel_Selector_avro.conf
2 a1.sources = r1
3 a1.sinks = k1
4 a1.channels = c1
5 # Describe/configure the source
6 a1.sources.r1.type = avro
7 a1.sources.r1.channels = c1
8 a1.sources.r1.bind = 0.0.0.0
9 a1.sources.r1.port = 5555
10 # Describe the sink
11 a1.sinks.k1.type = logger
12 # Use a channel which buffers events in memory
13 a1.channels.c1.type = memory
14 a1.channels.c1.capacity = 1000
15 a1.channels.c1.transactionCapacity = 100
16 # Bind the source and sink to the channel
17 a1.sources.r1.channels = c1
18 a1.sinks.k1.channel = c1

```

c)在m1上将2个配置文件复制到m2上一份

```

1 root@m1:/home/hadoop/flume-1.5.0-bin# scp -r /home/hadoop/flume-1.5.0-bin/conf/replicating_Channel_Selector_avro.conf m2:/home/hadoop/flume-1.5.0-bin/conf
2 root@m1:/home/hadoop/flume-1.5.0-bin# scp -r /home/hadoop/flume-1.5.0-bin/conf/replicating_Channel_Selector_avro.conf m2:/home/hadoop/flume-1.5.0-bin/conf

```

d)打开4个窗口，在m1和m2上同时启动两个flume agent

```

1 root@m1:/home/hadoop# /home/hadoop/flume-1.5.0-bin/bin/flume-ng agent -c . -f replicating_Channel_Selector_avro.conf -n a1 -Dflume.root.logger=INFO,console
2 root@m1:/home/hadoop# /home/hadoop/flume-1.5.0-bin/bin/flume-ng agent -c . -f replicating_Channel_Selector_avro.conf -n a2 -Dflume.root.logger=INFO,console

```

e)然后在m1或m2的任意一台机器上，测试产生syslog

```

1 root@m1:/home/hadoop# echo "hello idoall.org syslog" | nc localhost 5140

```

f)在m1和m2的sink窗口，分别可以看到以下信息,这说明信息得到了同步:

```

1 /08/10 14:08:18 INFO ipc.NettyServer: Connection to /192.168.1.51:46844 disconnected
2 /08/10 14:08:52 INFO ipc.NettyServer: [id: 0x90f8fe1f, /192.168.1.50:35873 => /192.168.1.51:46858]
3 /08/10 14:08:52 INFO ipc.NettyServer: [id: 0x90f8fe1f, /192.168.1.50:35873 => /192.168.1.51:46858]
4 /08/10 14:08:52 INFO ipc.NettyServer: [id: 0x90f8fe1f, /192.168.1.50:35873 => /192.168.1.51:46858]
5 /08/10 14:08:59 INFO ipc.NettyServer: [id: 0xd6318635, /192.168.1.51:46858 => /192.168.1.50:35873]
6 /08/10 14:08:59 INFO ipc.NettyServer: [id: 0xd6318635, /192.168.1.51:46858 => /192.168.1.50:35873]
7 /08/10 14:08:59 INFO ipc.NettyServer: [id: 0xd6318635, /192.168.1.51:46858 => /192.168.1.50:35873]
8 /08/10 14:09:20 INFO sink.LoggerSink: Event: { headers:{Severity=0, flume.syslog=hello idoall.org syslog} }

```

## 9)案例9: Multiplexing Channel Selector

a)在m1创建Multiplexing\_Channel\_Selector配置文件

```

1 root@m1:/home/hadoop# vi /home/hadoop/flume-1.5.0-bin/conf/Multiplexing_Channel_Selector.conf
2 a1.sources = r1
3 a1.sinks = k1 k2
4 a1.channels = c1 c2
5 # Describe/configure the source
6 a1.sources.r1.type = org.apache.flume.source.http.HTTPSource
7 a1.sources.r1.port = 5140
8 a1.sources.r1.channels = c1 c2
9 a1.sources.r1.selector.type = multiplexing
10 a1.sources.r1.selector.header = type
11 #映射允许每个值通道可以重叠。默认值可以包含任意数量的通道。
12 a1.sources.r1.selector.mapping.baidu = c1
13 a1.sources.r1.selector.mapping.ali = c2
14 a1.sources.r1.selector.default = c1
15 # Describe the sink
16 a1.sinks.k1.type = avro
17 a1.sinks.k1.channel = c1

```



口红品牌排行榜



中年离异征婚



大型游戏排行榜



```
18 a1.sinks.k1.hostname = m1
19 a1.sinks.k1.port = 5555
20 a1.sinks.k2.type = avro
21 a1.sinks.k2.channel = c2
22 a1.sinks.k2.hostname = m2
23 a1.sinks.k2.port = 5555
24 # Use a channel which buffers events in memory
25 a1.channels.c1.type = memory
26 a1.channels.c1.capacity = 1000
27 a1.channels.c1.transactionCapacity = 100
28 a1.channels.c2.type = memory
29 a1.channels.c2.capacity = 1000
30 a1.channels.c2.transactionCapacity = 100
```

b)在m1创建Multiplexing\_Channel\_Selector\_avro配置文件

```
1 root@ml1:/home/hadoop# vi /home/hadoop/flume-1.5.0-bin/conf/Multiplexing_Chapter1.conf
2 a1.sources = r1
3 a1.sinks = k1
4 a1.channels = c1
5 # Describe/configure the source
6 a1.sources.r1.type = avro
7 a1.sources.r1.channels = c1
8 a1.sources.r1.bind = 0.0.0.0
9 a1.sources.r1.port = 5555
10 # Describe the sink
11 a1.sinks.k1.type = logger
12 # Use a channel which buffers events in memory
13 a1.channels.c1.type = memory
14 a1.channels.c1.capacity = 1000
15 a1.channels.c1.transactionCapacity = 100
16 # Bind the source and sink to the channel
17 a1.sources.r1.channels = c1
18 a1.sinks.k1.channel = c1
```

c)将2个配置文件复制到m2上一份

```
1 root@m1:/home/hadoop/flume-1.5.0-bin# scp -r /home/hadoop/flume-1.5.0-bin/co ?
2 root@m1:/home/hadoop/flume-1.5.0-bin# scp -r /home/hadoop/flume-1.5.0-bin/conf
```

d)打开4个窗口，在m1和m2上同时启动两个flume agent

```
1 root@m1:/home/hadoop# /home/hadoop/flume-1.5.0-bin/bin/flume-ng agent -c . -?
2 root@m1:/home/hadoop# /home/hadoop/flume-1.5.0-bin/bin/flume-ng agent -c . -f
```

e)然后在m1或m2的任意一台机器上，测试产生syslog

```
root@kali:~# curl -X POST -d '{"headers":{"type": "baidu"}, "body": "idoall_TEST1"}' http://10.10.10.10:8080/
```

f)在m1的sink窗口，可以看到以下信息：

```

1 14/08/10 14:32:21 INFO node.Application: Starting Sink k1
2 14/08/10 14:32:21 INFO node.Application: Starting Source r1
3 14/08/10 14:32:21 INFO source.AvroSource: Starting Avro source r1: { bindAddr
4 14/08/10 14:32:21 INFO instrumentation.MonitoredCounterGroup: Monitored count
5 14/08/10 14:32:21 INFO instrumentation.MonitoredCounterGroup: Component type:
6 14/08/10 14:32:21 INFO source.AvroSource: Avro source r1 started.
7 14/08/10 14:32:36 INFO ipc.NettyServer: [id: 0xc0f00eea6, /192.168.1.50:35916
8 14/08/10 14:32:36 INFO ipc.NettyServer: [id: 0xc0f00eea6, /192.168.1.50:35916
9 14/08/10 14:32:36 INFO ipc.NettyServer: [id: 0xc0f00eea6, /192.168.1.50:35916
10 14/08/10 14:32:44 INFO ipc.NettyServer: [id: 0x432f5468, /192.168.1.51:46945
11 14/08/10 14:32:44 INFO ipc.NettyServer: [id: 0x432f5468, /192.168.1.51:46945
12 14/08/10 14:32:44 INFO ipc.NettyServer: [id: 0x432f5468, /192.168.1.51:46945
13 14/08/10 14:34:11 INFO sink.LoggerSink: Event: { headers:{type=baidu} body: 6
14 14/08/10 14:34:57 INFO sink.LoggerSink: Event: { headers:{type=qq} body: 69

```

g)在m2的sink窗口，可以看到以下信息：

```

1 14/08/10 14:32:27 INFO node.Application: Starting Sink k1
2 14/08/10 14:32:27 INFO node.Application: Starting Source r1
3 14/08/10 14:32:27 INFO source.AvroSource: Starting Avro source r1: { bindAddr
4 14/08/10 14:32:27 INFO instrumentation.MonitoredCounterGroup: Monitored count
5 14/08/10 14:32:27 INFO instrumentation.MonitoredCounterGroup: Component type:
6 14/08/10 14:32:27 INFO source.AvroSource: Avro source r1 started.
7 14/08/10 14:32:36 INFO ipc.NettyServer: [id: 0x7c2f0aec, /192.168.1.50:38104
8 14/08/10 14:32:36 INFO ipc.NettyServer: [id: 0x7c2f0aec, /192.168.1.50:38104
9 14/08/10 14:32:36 INFO ipc.NettyServer: [id: 0x7c2f0aec, /192.168.1.50:38104
10 14/08/10 14:32:44 INFO ipc.NettyServer: [id: 0x3d36f553, /192.168.1.51:48599
11 14/08/10 14:32:44 INFO ipc.NettyServer: [id: 0x3d36f553, /192.168.1.51:48599
12 14/08/10 14:32:44 INFO ipc.NettyServer: [id: 0x3d36f553, /192.168.1.51:48599
13 14/08/10 14:34:33 INFO sink.LoggerSink: Event: { headers:{type=all} body: 69

```



## 口红品牌排行榜



## 中年离异征婚



## 大型游戏排行榜

可以看到，根据header中不同的条件分布到不同的channel上

#### 10)案例10: Flume Sink Processors

failover的机器是一直发送给其中一个sink，当这个sink不可用的时候，自动发送到下一个sink。

a)在m1创建Flume\_Sink\_Processors配置文件

```

1 root@m1:/home/hadoop# vi /home/hadoop/flume-1.5.0-bin/conf/Flume_Sink_Proce?;
2
3 a1.sources = r1
4 a1.sinks = k1 k2
5 a1.channels = c1 c2
6
7 #这个是配置failover的关键，需要有一个sink group
8 a1.sinkgroups = g1
9 a1.sinkgroups.g1.sinks = k1 k2
10 #处理的类型是failover
11 a1.sinkgroups.g1.processor.type = failover
12 #优先级，数字越大优先级越高，每个sink的优先级必须不相同
13 a1.sinkgroups.g1.processor.priority.k1 = 5
14 a1.sinkgroups.g1.processor.priority.k2 = 10
15 #设置为10秒，当然可以根据你的实际状况更改成更快或者很慢
16 a1.sinkgroups.g1.processor.maxpenalty = 10000
17
18 # Describe/configure the source
19 a1.sources.r1.type = syslogtcp
20 a1.sources.r1.port = 5140
21 a1.sources.r1.channels = c1 c2
22 a1.sources.r1.selector.type = replicating
23
24
25 # Describe the sink
26 a1.sinks.k1.type = avro
27 a1.sinks.k1.channel = c1
28 a1.sinks.k1.hostname = m1
29 a1.sinks.k1.port = 5555
30
31 a1.sinks.k2.type = avro
32 a1.sinks.k2.channel = c2
33 a1.sinks.k2.hostname = m2
34 a1.sinks.k2.port = 5555
35
36 # Use a channel which buffers events in memory
37 a1.channels.c1.type = memory
38 a1.channels.c1.capacity = 1000
39 a1.channels.c1.transactionCapacity = 100
40
41 a1.channels.c2.type = memory
42 a1.channels.c2.capacity = 1000
43 a1.channels.c2.transactionCapacity = 100

```

b)在m1创建Flume\_Sink\_Processors\_avro配置文件

```

1 root@m1:/home/hadoop# vi /home/hadoop/flume-1.5.0-bin/conf/Flume_Sink_Proce?;
2
3 a1.sources = r1
4 a1.sinks = k1
5 a1.channels = c1
6
7 # Describe/configure the source
8 a1.sources.r1.type = avro
9 a1.sources.r1.channels = c1
10 a1.sources.r1.bind = 0.0.0.0
11 a1.sources.r1.port = 5555
12
13 # Describe the sink
14 a1.sinks.k1.type = logger
15
16 # Use a channel which buffers events in memory
17 a1.channels.c1.type = memory
18 a1.channels.c1.capacity = 1000
19 a1.channels.c1.transactionCapacity = 100
20
21 # Bind the source and sink to the channel
22 a1.sources.r1.channels = c1
23 a1.sinks.k1.channel = c1

```

c)将2个配置文件复制到m2上一份

```

1 root@m1:/home/hadoop/flume-1.5.0-bin# scp -r /home/hadoop/flume-1.5.0-bin/co?;
2 root@m1:/home/hadoop/flume-1.5.0-bin# scp -r /home/hadoop/flume-1.5.0-bin/conf?;

```

d)打开4个窗口，在m1和m2上同时启动两个flume agent

```

1 root@m1:/home/hadoop# /home/hadoop/flume-1.5.0-bin/bin/flume-ng agent -c . -f ?;
2 root@m1:/home/hadoop# /home/hadoop/flume-1.5.0-bin/bin/flume-ng agent -c . -f ?;

```

HOT



口红品牌排行榜



中年离异征婚



大型游戏排行榜

e)然后在m1或m2的任意一台机器上，测试产生log

```
1 root@m1:/home/hadoop# echo "idoall.org test1 failover" | nc localhost 5140
```

f)因为m2的优先级高，所以在m2的sink窗口，可以看到以下信息，而m1没有：

```
1 14/08/10 15:02:46 INFO ipc.NettyServer: Connection to /192.168.1.51:48692 di?
2 14/08/10 15:03:12 INFO ipc.NettyServer: [id: 0x09a14036, /192.168.1.51:48704 =
3 14/08/10 15:03:12 INFO ipc.NettyServer: [id: 0x09a14036, /192.168.1.51:48704 =
4 14/08/10 15:03:12 INFO ipc.NettyServer: [id: 0x09a14036, /192.168.1.51:48704 =
5 14/08/10 15:03:26 INFO sink.LoggerSink: Event: { headers:{Severity=0, flume.sy
```

g)这时我们停止掉m2机器上的sink(ctrl+c)，再次输出测试数据：

```
1 root@m1:/home/hadoop# echo "idoall.org test2 failover" | nc localhost 5140
```

h)可以在m1的sink窗口，看到读取到了刚才发送的两条测试数据：

```
1 14/08/10 15:02:46 INFO ipc.NettyServer: Connection to /192.168.1.51:47036 di?
2 14/08/10 15:03:12 INFO ipc.NettyServer: [id: 0xbcf79851, /192.168.1.51:47048 =
3 14/08/10 15:03:12 INFO ipc.NettyServer: [id: 0xbcf79851, /192.168.1.51:47048 =
4 14/08/10 15:03:12 INFO ipc.NettyServer: [id: 0xbcf79851, /192.168.1.51:47048 =
5 14/08/10 15:07:56 INFO sink.LoggerSink: Event: { headers:{Severity=0, flume.sy
6 14/08/10 15:07:56 INFO sink.LoggerSink: Event: { headers:{Severity=0, flume.sy
```

i)我们再次在m2的sink窗口中，启动sink：

```
1 root@m1:/home/hadoop# /home/hadoop/flume-1.5.0-bin/bin/flume-ng agent -c . -?
```

j)输入两批测试数据：

```
1 root@m1:/home/hadoop# echo "idoall.org test3 failover" | nc localhost 5140
```

k)在m2的sink窗口，我们可以看到以下信息，因为优先级的关系，log消息会再次落到m2上：

```
1 14/08/10 15:09:47 INFO node.Application: Starting Sink k1
2 14/08/10 15:09:47 INFO node.Application: Starting Source r1
3 14/08/10 15:09:47 INFO source.AvroSource: Starting Avro source r1: { bindAddr
4 14/08/10 15:09:47 INFO instrumentation.MonitoredCounterGroup: Monitored count
5 14/08/10 15:09:47 INFO instrumentation.MonitoredCounterGroup: Component type:
6 14/08/10 15:09:47 INFO source.AvroSource: Avro source r1 started.
7 14/08/10 15:09:54 INFO ipc.NettyServer: [id: 0x96615732, /192.168.1.51:48741
8 14/08/10 15:09:54 INFO ipc.NettyServer: [id: 0x96615732, /192.168.1.51:48741
9 14/08/10 15:09:54 INFO ipc.NettyServer: [id: 0x96615732, /192.168.1.51:48741
10 14/08/10 15:09:57 INFO sink.LoggerSink: Event: { headers:{Severity=0, flume.s
11 14/08/10 15:10:43 INFO ipc.NettyServer: [id: 0x12621f9a, /192.168.1.50:38166
12 14/08/10 15:10:43 INFO ipc.NettyServer: [id: 0x12621f9a, /192.168.1.50:38166
13 14/08/10 15:10:43 INFO ipc.NettyServer: [id: 0x12621f9a, /192.168.1.50:38166
14 14/08/10 15:10:43 INFO sink.LoggerSink: Event: { headers:{Severity=0, flume.s
15 14/08/10 15:10:43 INFO sink.LoggerSink: Event: { headers:{Severity=0, flume.s
```

#### 11)案例11: Load balancing Sink Processor

load balance type和failover不同的地方是，load balance有两个配置，一个是轮询，一个是随机。

两种情况下如果被选择的sink不可用，就会自动尝试发送到下一个可用的sink上面。

a)在m1创建Load\_balancing\_Sink\_Processors配置文件

```
1 root@m1:/home/hadoop# vi /home/hadoop/flume-1.5.0-bin/conf/Load_balancing_S
2
3 a1.sources = r1
4 a1.sinks = k1 k2
5 a1.channels = c1
6
7 #这个是配置Load balancing的关键，需要有一个sink group
8 a1.sinkgroups = g1
9 a1.sinkgroups.g1.sinks = k1 k2
10 a1.sinkgroups.g1.processor.type = load_balance
11 a1.sinkgroups.g1.processor.backoff = true
12 a1.sinkgroups.g1.processor.selector = round_robin
13
14 # Describe/configure the source
15 a1.sources.r1.type = syslogtcp
16 a1.sources.r1.port = 5140
17 a1.sources.r1.channels = c1
18
19
20 # Describe the sink
21 a1.sinks.k1.type = avro
```

HOT



口红品牌排行榜



中年离异征婚



大型游戏排行榜

```

22 a1.sinks.k1.channel = c1
23 a1.sinks.k1.hostname = m1
24 a1.sinks.k1.port = 5555
25
26 a1.sinks.k2.type = avro
27 a1.sinks.k2.channel = c1
28 a1.sinks.k2.hostname = m2
29 a1.sinks.k2.port = 5555
30
31 # Use a channel which buffers events in memory
32 a1.channels.c1.type = memory
33 a1.channels.c1.capacity = 1000
34 a1.channels.c1.transactionCapacity = 100

```

b)在m1创建Load\_balancing\_Sink\_Processors\_avro配置文件

```

1 root@m1:/home/hadoop# vi /home/hadoop/flume-1.5.0-bin/conf/Load_balancing_S?
2
3 a1.sources = r1
4 a1.sinks = k1
5 a1.channels = c1
6
7 # Describe/configure the source
8 a1.sources.r1.type = avro
9 a1.sources.r1.channels = c1
10 a1.sources.r1.bind = 0.0.0.0
11 a1.sources.r1.port = 5555
12
13 # Describe the sink
14 a1.sinks.k1.type = logger
15
16 # Use a channel which buffers events in memory
17 a1.channels.c1.type = memory
18 a1.channels.c1.capacity = 1000
19 a1.channels.c1.transactionCapacity = 100
20
21 # Bind the source and sink to the channel
22 a1.sources.r1.channels = c1
23 a1.sinks.k1.channel = c1

```

c)将2个配置文件复制到m2上一份

```

1 root@m1:/home/hadoop/flume-1.5.0-bin# scp -r /home/hadoop/flume-1.5.0-bin/co?
2 root@m1:/home/hadoop/flume-1.5.0-bin# scp -r /home/hadoop/flume-1.5.0-bin/conf

```

d)打开4个窗口，在m1和m2上同时启动两个flume agent

```

1 root@m1:/home/hadoop# /home/hadoop/flume-1.5.0-bin/bin/flume-ng agent -c . -?
2 root@m1:/home/hadoop# /home/hadoop/flume-1.5.0-bin/bin/flume-ng agent -c . -f

```

e)然后在m1或m2的任意一台机器上，测试产生log，一行一行输入，输入太快，容易落到一台机器上

```

1 root@m1:/home/hadoop# echo "idoall.org test1" | nc localhost 5140
2 root@m1:/home/hadoop# echo "idoall.org test2" | nc localhost 5140
3 root@m1:/home/hadoop# echo "idoall.org test3" | nc localhost 5140
4 root@m1:/home/hadoop# echo "idoall.org test4" | nc localhost 5140

```

f)在m1的sink窗口，可以看到以下信息：

```

1 14/08/10 15:35:29 INFO sink.LoggerSink: Event: { headers:{Severity=0, flume.?}
2 14/08/10 15:35:33 INFO sink.LoggerSink: Event: { headers:{Severity=0, flume.sy

```

g)在m2的sink窗口，可以看到以下信息：

```

1 14/08/10 15:35:27 INFO sink.LoggerSink: Event: { headers:{Severity=0, flume.?}
2 14/08/10 15:35:29 INFO sink.LoggerSink: Event: { headers:{Severity=0, flume.sy

```

说明轮询模式起到了作用。

## 12)案例12: Hbase sink

a)在测试之前，请先参考《ubuntu12.04+hadoop2.2.0+ookeeper3.4.5+hbase0.96.2+hive0.13.1分布式环境部署》将hbase启动

b)然后将以下文件复制到flume中：

```

1 cp /home/hadoop/hbase-0.96.2-hadoop2/lib/protobuf-java-2.5.0.jar /home/hadoo?

```

HOT



口红品牌排行榜



中年离异征婚



大型游戏排行榜



```

2 cp /home/hadoop/hbase-0.96.2-hadoop2/lib/hbase-client-0.96.2-hadoop2.jar /home
3 cp /home/hadoop/hbase-0.96.2-hadoop2/lib/hbase-common-0.96.2-hadoop2.jar /home
4 cp /home/hadoop/hbase-0.96.2-hadoop2/lib/hbase-protocol-0.96.2-hadoop2.jar /home
5 cp /home/hadoop/hbase-0.96.2-hadoop2/lib/hbase-server-0.96.2-hadoop2.jar /home
6 cp /home/hadoop/hbase-0.96.2-hadoop2/lib/hbase-hadoop2-compat-0.96.2-hadoop2.j
7 cp /home/hadoop/hbase-0.96.2-hadoop2/lib/hbase-hadoop-compat-0.96.2-hadoop2.ja
8 cp /home/hadoop/hbase-0.96.2-hadoop2/lib/htrace-core-2.04.jar /home/hadoop/flume

```

HOT

c)确保test\_idoall\_org表在hbase中已经存在, test\_idoall\_org表的格式以及字段请参考《ubuntu12.04+hadoop2.2.0+ookeeper3.4.5+hbase0.96.2+hive0.13.1分布式环境部署》中关于hbase部分的建表代码。

d)在m1创建hbase\_simple配置文件

```

1 root@m1:/home/hadoop# vi /home/hadoop/flume-1.5.0-bin/conf/hbase_simple.conf
2
3 a1.sources = r1
4 a1.sinks = k1
5 a1.channels = c1
6
7 # Describe/configure the source
8 a1.sources.r1.type = syslogtcp
9 a1.sources.r1.port = 5140
10 a1.sources.r1.host = localhost
11 a1.sources.r1.channels = c1
12
13 # Describe the sink
14 a1.sinks.k1.type = logger
15 a1.sinks.k1.type = hbase
16 a1.sinks.k1.table = test_idoall_org
17 a1.sinks.k1.columnFamily = name
18 a1.sinks.k1.column = idoall
19 a1.sinks.k1.serializer = org.apache.flume.sink.hbase.RegexHbaseEventSerializer
20 a1.sinks.k1.channel = memoryChannel
21
22 # Use a channel which buffers events in memory
23 a1.channels.c1.type = memory
24 a1.channels.c1.capacity = 1000
25 a1.channels.c1.transactionCapacity = 100
26
27 # Bind the source and sink to the channel
28 a1.sources.r1.channels = c1
29 a1.sinks.k1.channel = c1

```

e)启动flume agent

```
1 /home/hadoop/flume-1.5.0-bin/bin/flume-ng agent -c . -f /home/hadoop/flume-1.5.0-bin/conf/hbase_simple.conf -Dflume.root.logger=INFO,console
```

f)测试产生syslog

```
1 root@m1:/home/hadoop# echo "hello idoall.org from flume" | nc localhost 5140
```

g)这时登录到hbase中, 可以发现新数据已经插入

```

1 root@m1:/home/hadoop# /home/hadoop/hbase-0.96.2-hadoop2/bin/hbase shell
2 2014-08-10 16:09:48,984 INFO [main] Configuration.deprecation: hadoop.native.
3 HBase Shell; enter 'help<RETURN>' for list of supported commands.
4 Type "exit<RETURN>" to leave the HBase Shell
5 Version 0.96.2-hadoop2, r1581096, Mon Mar 24 16:03:18 PDT 2014
6
7 hbase(main):001:0> list
8 TABLE
9 SLF4J: Class path contains multiple SLF4J bindings.
10 SLF4J: Found binding in [jar:file:/home/hadoop/hbase-0.96.2-hadoop2/lib/slf4j
11 SLF4J: Found binding in [jar:file:/home/hadoop/hadoop-2.2.0/share/hadoop/comm
12 SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanati
13 hbase2hive_idoall
14 hive2hbase_idoall
15 test_idoall_org
16 3 row(s) in 2.6880 seconds
17
18 => ["hbase2hive_idoall", "hive2hbase_idoall", "test_idoall_org"]
19 hbase(main):002:0> scan "test_idoall_org"
20 ROW COLUMN+CELL
21 10086 column=name:idoall, timestamp=1406424831473, v
22 1 row(s) in 0.0550 seconds
23
24 hbase(main):003:0> scan "test_idoall_org"
25 ROW COLUMN+CELL
26 10086 column=name:idoall, timestamp=1406424831473, v
27 1407658495588-XbQCOZrKK8-0 column=name:payload, timestamp=14076
28 2 row(s) in 0.0200 seconds
29
30 hbase(main):004:0> quit

```



口红品牌排行榜



中年离异征婚

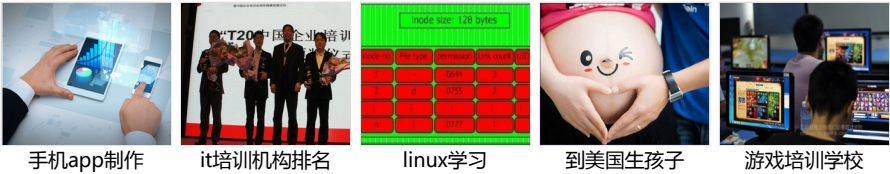


大型游戏排行榜

经过这么多flume的例子测试，如果你全部做完后，会发现flume的功能真的很强大，可以进行各种搭配来完成你想要的工作，俗话说师傅领进门，修行在个人，如何能够结合你的产品业务，将flume更好的应用起来，快去动手实践吧。

HOT

这篇文章做为一个笔记，希望能够对刚入门的同学起到帮助作用。



linux培训   安卓app   软件工程师月薪      代办工作居住证   宝宝奶粉排行榜

小米机械键盘   电脑组装价格   创意设计网站      成人高考   硬盘数据修复   好贷宝

立体车库   机械键盘吧   食堂承包      二手车估价计算器   快餐排名   程序员有前..

透明手机   德云社门票   美国生孩子      家用小型电梯   三星note4   燃气壁挂炉价格

Tags: Flume   环境部署   配置

相关文章

|                                       |            |
|---------------------------------------|------------|
| LNMP下FTP服务器的安装与使用方法(Pureftpd和Proftpd) | 2013-06-06 |
| 用DNSPod和Squid打造自己的CDN (七) 配置Squid     | 2013-04-04 |
| 详解Linux批量更改文件后缀名                      | 2016-11-11 |
| Linux下Squid代理服务器的架设与维护经验分享            | 2011-06-06 |
| CentOS VPS安装Kloxo免费虚拟主机管理系统           | 2011-03-03 |
| Linux下实现定时器Timer的几种方法总结               | 2016-10-10 |
| Apache mod_rewrite实现HTTP和HTTPS重定向跳转   | 2015-06-06 |
| Linux配置远程SSH无密码登录                     | 2016-10-10 |
| 详解linux软连接和硬链接                        | 2016-10-10 |
| Linux系统使用用户密钥ssh主机访问                  | 2016-10-10 |



最新评论

评论(4人参与，0条评论)



来说两句吧...

微博登录

QQ登录

手机登录

还没有评论，快来抢沙发吧！

Powered by 畅言



HOT

|



口红品牌排行榜



中年离异征婚



大型游戏排行榜