

<http://www.aboutyun.com/?61> [收藏] [复制] [分享] [RSS]

- | | | | | | | | | | |
|------|----|----|----|----|----|----|----|-----|------|
| 空间首页 | 动态 | 记录 | 日志 | 相册 | 广播 | 主题 | 分享 | 留言板 | 个人资料 |
|------|----|----|----|----|----|----|----|-----|------|

日志

热度 2 已有 537 次阅读2014-6-2 12:55

问题导读:

- 1.什么是flume?
- 2.如何安装flume?
- 3.flume的配置文件与其它软件有什么不同?



-  查看广播
  收听TA
-  加为好友
  给我留言
-  打个招呼
  发送消息

1.flume是什么?

这里简单介绍一下，它是Cloudera的一个产品

- ## 2.flume是干什么的?

收集日志的

- ### 3.flume如何搜集日志?

我们把flume比作情报人员

- (1) 搜集信息
- (2) 获取记忆信息
- (3) 传递报告间谍信息

flume是怎么完成上面三件事情的，三个组件：

source: 搜集信息

channel: 传递信息

sink: 存储信息

上面有点简练，详细可以参考Flume内置channel,source,sink三组件介绍

上面我们认识了，flume。

下面我们来安装flume1.5

1. 下载安装包

□ □ □ □ □ □ □ □ □ □ □ □ □ □

[illegible]

□ □ □ □

(1) 官网下载

apache-flume-1.5.0-bin.tar.gz

apache-flume-1.5.0-src.tar.gz

(2) 百度网盘下载

链接: <http://pan.baidu.com/s/1dDip8RZ> 密码: 268r

我们走到这一步，我们会想到一个问题，我的电脑是32位的，不知道能否安装？如果我的电脑是64位的，能否安装。之前我们装的hadoop就分为32位和64位，想到这个问题是正常的，但是这里不用担心，因为我们下载的是二进制包，也就是说你32位和64位都可以安装。

2.分别解压:

下载之后，我们看到下面两个包：

(1) 上传Linux

```
aboutyun@master:~/flume$  
apache-flume-1.5.0-bin.tar.gz  
apache-flume-1.5.0-src.tar.gz  
aboutyun@master:~/flume$
```

上面两个包，可以下载window，然后通过WinSCP,如果不会 新手指导：使用 WinSCP（下载） 上文件到 Linux图文教程

(2) 解压包

解压apache-flume-1.5.0-bin.tar.gz，解压到usr文件夹下面

```
]]]] sudo tar zxvf apache-flume-1.5.0-bin.tar.gz
```

```
aboutyun@master:/usr$ sudo tar zxvf apache-flume-1.5.0-bin.tar.gz
```

解压apache-flume-1.5.0-src.tar.gz，解压到usr文件夹下面

```
]]]] sudo tar zxvf apache-flume-1.5.0-src.tar.gz
```

```
aboutyun@master:/usr$ sudo tar zxvf apache-flume-1.5.0-src.tar.gz
```

(3) src里面文件内容，覆盖解压后bin文件里面的内容

```
❏❏❏ sudo cp -ri apache-flume-1.5.0-src/* apache-flume-1.5.0-bin
```

100

```
aboutyun@master:/usr$ sudo cp -ri apache-flume-1.5.0-src/* apache-flume-1.5.0-bin
```

(4)重命名

```
]]]] mv apache-flume-1.5.0-bin/ flume
```

```
aboutyun@master:/usr$ mv apache-flume-1.5.0-bin/ flume
```

3.配置环境变量：

```
PATH="/usr/flume/bin:/usr/hive/bin:/usr/hbase/bin:/usr/hadoop/bin:/usr/
```

配置环境变量生效

```
1□□ source /etc/environment
```

3.建立配置文件

这里的配置文件还是比较特别的，不同于以往我们安装的软件，我们这里可以自己建立配置文件。

首先我们建立一个 example 文件

```
1□□ vi example
```

，然后把下面内容，粘帖到里面就可以了，注意不要有乱码，有乱码的话，可以直接创建一个文件，然后上传。方法也有很多，能解决就好。

对于下面红字部分，记得创建文件夹，并且注意他们的权限一致，这个比较简单的，就不在书写了。对于下面的配置项，可以参考flume参考文档，这里面的参数很详细。

```
agent1表示代理名称
agent1.sources=source1
agent1.sinks=sink1
agent1.channels=channel1

#配置source1
agent1.sources.source1.type=spooldir
agent1.sources.source1.spoolDir=/usr/aboutyunlog
agent1.sources.source1.channels=channel1
agent1.sources.source1.fileHeader = false

#配置sink1
agent1.sinks.sink1.type=hdfs
agent1.sinks.sink1.hdfs.path=hdfs://master:8020/aboutyunlog
agent1.sinks.sink1.hdfs.fileType=DataStream
agent1.sinks.sink1.hdfs.writeFormat=TEXT
agent1.sinks.sink1.hdfs.rollInterval=4
agent1.sinks.sink1.channel=channel1

#配置channel1
agent1.channels.channel1.type=file
agent1.channels.channel1.checkpointDir=/usr/aboutyun_tmp123
```

```
agent1.channels.channel1.dataDirs=/usr/aboutyun_tmp
```

```
agent1表示代理名称
agent1.sources=source1
agent1.sinks=sink1
agent1.channels=channel1

#配置source1
agent1.sources.source1.type=spooldir
agent1.sources.source1.spoolDir=/usr/aboutyunlog
agent1.sources.source1.channels=channel1
agent1.sources.source1.fileHeader = false

#配置sink1
agent1.sinks.sink1.type=hdfs
agent1.sinks.sink1.hdfs.path=hdfs://master:8020/aboutyunlog
agent1.sinks.sink1.hdfs.fileType=DataStream
agent1.sinks.sink1.hdfs.writeFormat=TEXT
agent1.sinks.sink1.hdfs.rollInterval=4
agent1.sinks.sink1.channel=channel1
```

注意：执行命令是
./flume-ng agent -n agent1 -c ~/soft/apache-flume-1.5.0.1-bin/conf/ -f ~/soft/apache-flume-1.5.0.1-bin/conf/flume-conf.properties -Dflume.root.logger=ALL,console
执行时会报错，一些类找不到，需要执行下面的命令，拷贝hadoop一些jar过去

```
cp ~/soft/hadoop-2.2.0/share/hadoop/common/hadoop-common-2.2.0.jar ~/soft/apache-flume-1.5.0.1-bin/lib/
cp ~/soft/hadoop-2.2.0/share/hadoop/common/lib/commons-configuration-1.6.jar ~/soft/apache-flume-1.5.0.1-bin/lib/
cp ~/soft/hadoop-2.2.0/share/hadoop/common/lib/hadoop-auth-2.2.0.jar ~/soft/apache-flume-1.5.0.1-bin/lib/
cp ~/soft/hadoop-2.2.0/share/hadoop/hdfs/hadoop-hdfs-2.2.0.jar ~/soft/apache-flume-1.5.0.1-bin/lib/
```

```
flume-ng agent -n agent1 -c conf -f /usr/flume/conf/example -Dflume.root.logger=DEBUG,console
```

上面注意红字部分，是我们自己建立的文件，而对于绿色部分，则是输出调试信息，也可以在配置文件中配置。

5.我们启动flume之后

会看到下面信息，并且信息不停的重复。这个其实是在空文件的时候，监控的信息输出。

```
2014-06-02 12:23:42,488 (conf-file-poller-0) [DEBUG - org.apache.flume.node.PollingPropertiesFileConfigurationProvider$FileWatcherRunnable.run(PollingPropertiesFileConfigurationProvider.java:126)] Checking file:conf/example1 for changes
2014-06-02 12:24:11,017 (Log-BackgroundWorker-channel1) [DEBUG - org.apache.flume.channel.file.FlumeEventQueue.checkpoint(FlumeEventQueue.java:137)] Checkpoint not required
2014-06-02 12:24:12,489 (conf-file-poller-0) [DEBUG - org.apache.flume.node.PollingPropertiesFileConfigurationProvider$FileWatcherRunnable.run(PollingPropertiesFileConfigurationProvider.java:126)] Checking file:conf/example1 for changes
2014-06-02 12:24:41,018 (Log-BackgroundWorker-channel1) [DEBUG - org.apache.flume.channel.file.FlumeEventQueue.checkpoint(FlumeEventQueue.java:137)] Checkpoint not required
2014-06-02 12:24:42,490 (conf-file-poller-0) [DEBUG - org.apache.flume.node.PollingPropertiesFileConfigurationProvider$FileWatcherRunnable.run(PollingPropertiesFileConfigurationProvider.java:126)] Checking file:conf/example1 for changes
```

一旦有文件输入，我们会看到下面信息。

注意：这个不要关闭，我们另外开启一个shell，在监控文件夹中放入要上传的文件

比如我们在监控文件夹下，创建一个test1文件，内容如下

```
hello 1
hello 2
hello aboutyun
~
~
~
~
~
```

这时候flume监控shell，会有相应的如下下面变化

```
2014-06-02 12:01:04,066 (pool-6-thread-1) [INFO -
org.apache.flume.client.avro.ReliableSpoolingFileEventReader.rollCurrentFile(ReliableSpoolingFileEventRea
der.java:332)] Preparing to move file /usr/aboutyunlog/test1 to /usr/aboutyunlog/test1.COMPLETED
2014-06-02 12:01:04,070 (pool-6-thread-1) [ERROR -
org.apache.flume.source.SpoolDirectorySource$SpoolDirectoryRunnable.run(SpoolDirectorySource.java:25
6)] FATAL: Spool Directory source source1: { spoolDir: /usr/aboutyunlog }: Uncaught exception in
SpoolDirectorySource thread. Restart or reconfigure Flume to continue processing.
java.lang.IllegalStateException: File name has been re-used with different files. Spooling assumptions
violated for /usr/aboutyunlog/test1.COMPLETED
at
org.apache.flume.client.avro.ReliableSpoolingFileEventReader.rollCurrentFile(ReliableSpoolingFileEventRea
der.java:362)
at
org.apache.flume.client.avro.ReliableSpoolingFileEventReader.retireCurrentFile(ReliableSpoolingFileEventR
eader.java:314)
at
org.apache.flume.client.avro.ReliableSpoolingFileEventReader.readEvents(ReliableSpoolingFileEventReader
.java:243)
at
org.apache.flume.source.SpoolDirectorySource$SpoolDirectoryRunnable.run(SpoolDirectorySource.java:22
7)
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:471)
at java.util.concurrent.FutureTask.runAndReset(FutureTask.java:304)
at
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$301(ScheduledThreadPoolExe
cutor.java:178)
at
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExecut
or.java:293)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)
at java.lang.Thread.run(Thread.java:744)
2014-06-02 12:01:07,749 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO -
org.apache.flume.sink.hdfs.HDFSDataStream.configure(HDFSDataStream.java:58)] Serializer = TEXT,
UseRawLocalFileSystem = false
2014-06-02 12:01:07,803 (SinkRunner-PollingRunner-DefaultSinkProcessor) [INFO -
org.apache.flume.sink.hdfs.BucketWriter.open(BucketWriter.java:261)] Creating
hdfs://master:8020/aboutyunlog/FlumeData.1401681667750.tmp
```

```
2014-06-02 12:01:07,871 (hdfs-sink1-call-runner-2) [DEBUG -
org.apache.flume.sink.hdfs.AbstractHDFSWriter.reflectGetNumCurrentReplicas(AbstractHDFSWriter.java:1
95)] Using getNumCurrentReplicas--HDFS-826
2014-06-02 12:01:07,871 (hdfs-sink1-call-runner-2) [DEBUG -
org.apache.flume.sink.hdfs.AbstractHDFSWriter.reflectGetDefaultReplication(AbstractHDFSWriter.java:223
)] Using FileSystem.getDefaultReplication(Path) from HADOOP-8014
2014-06-02 12:01:10,945 (Log-BackgroundWorker-channel1) [INFO -
org.apache.flume.channel.file.EventQueueBackingStoreFile.beginCheckpoint(EventQueueBackingStoreFile.j
ava:214)] Start checkpoint for /usr/aboutyun_tmp123/checkpoint, elements to sync = 3
2014-06-02 12:01:10,949 (Log-BackgroundWorker-channel1) [INFO -
org.apache.flume.channel.file.EventQueueBackingStoreFile.checkpoint(EventQueueBackingStoreFile.java:2
39)] Updating checkpoint metadata: logWriteOrderID: 1401681430998, queueSize: 0, queueHead: 11
2014-06-02 12:01:10,952 (Log-BackgroundWorker-channel1) [INFO -
org.apache.flume.channel.file.Log.writeCheckpoint(Log.java:1005)] Updated checkpoint for file:
/usr/aboutyun_tmp/log-8 position: 2482 logWriteOrderID: 1401681430998
2014-06-02 12:01:10,953 (Log-BackgroundWorker-channel1) [DEBUG -
org.apache.flume.channel.file.Log.removeOldLogs(Log.java:1067)] Files currently in use: [8]
2014-06-02 12:01:11,872 (hdfs-sink1-roll-timer-0) [DEBUG -
org.apache.flume.sink.hdfs.BucketWriter$2.call(BucketWriter.java:303)] Rolling file
(hdfs://master:8020/aboutyunlog/FlumeData.1401681667750.tmp): Roll scheduled after 4 sec elapsed.
2014-06-02 12:01:11,873 (hdfs-sink1-roll-timer-0) [INFO -
org.apache.flume.sink.hdfs.BucketWriter.close(BucketWriter.java:409)] Closing
hdfs://master:8020/aboutyunlog/FlumeData.1401681667750.tmp
2014-06-02 12:01:11,873 (hdfs-sink1-call-runner-7) [INFO -
org.apache.flume.sink.hdfs.BucketWriter$3.call(BucketWriter.java:339)] Close tries incremented
2014-06-02 12:01:11,895 (hdfs-sink1-call-runner-8) [INFO -
org.apache.flume.sink.hdfs.BucketWriter$8.call(BucketWriter.java:669)] Renaming
hdfs://master:8020/aboutyunlog/FlumeData.1401681667750.tmp to
hdfs://master:8020/aboutyunlog/FlumeData.1401681667750
2014-06-02 12:01:11,897 (hdfs-sink1-roll-timer-0) [INFO -
org.apache.flume.sink.hdfs.HDFSEventSink$1.run(HDFSEventSink.java:402)] Writer callback called.
2014-06-02 12:01:12,423 (conf-file-poller-0) [DEBUG -
org.apache.flume.node.PollingPropertiesFileConfigurationProvider$FileWatcherRunnable.run(PollingPropert
iesFileConfigurationProvider.java:126)] Checking file:conf/example for changes
2014-06-02 12:01:40,953 (Log-BackgroundWorker-channel1) [DEBUG -
org.apache.flume.channel.file.FlumeEventQueue.checkpoint(FlumeEventQueue.java:137)] Checkpoint not
required
```

上传成功之后，我们去hdfs上，查看上传文件：

```
aboutyun@master:/usr/aboutyunlog$ hadoop fs -text /aboutyunlog/FlumeData.1401681
667750
hello 1
hello 2
hello aboutyun
```

这样我们做到了flume上传到hadoop2.2。

完毕



举报

邀请

分享


收藏

评论 (2 个评论)

发表评论

作者的其他最新日志


全部



hadoop_lwq 2014-6-17 14:54


回复


哥们，这程序都报错了，你再copy一个文件放在flume的监控目录下，flume还能传输吗？你是怎么解决的？



pig2 2014-6-17 19:40

回复

 **hadoop_lwq**: 哥们，这程序都报错了，你再copy一个文件放在flume的监控目录下，flume还能传输吗？你是怎么解决的？



可以正常传输，上传几个都没有问题。检查下自己的配置。

需要保证flume一直运行。

第二次重启的时候，需要清空上传目录。

 涂鸦板

您需要登录后才可以评论 [登录](#) | [立即注册](#)

评论