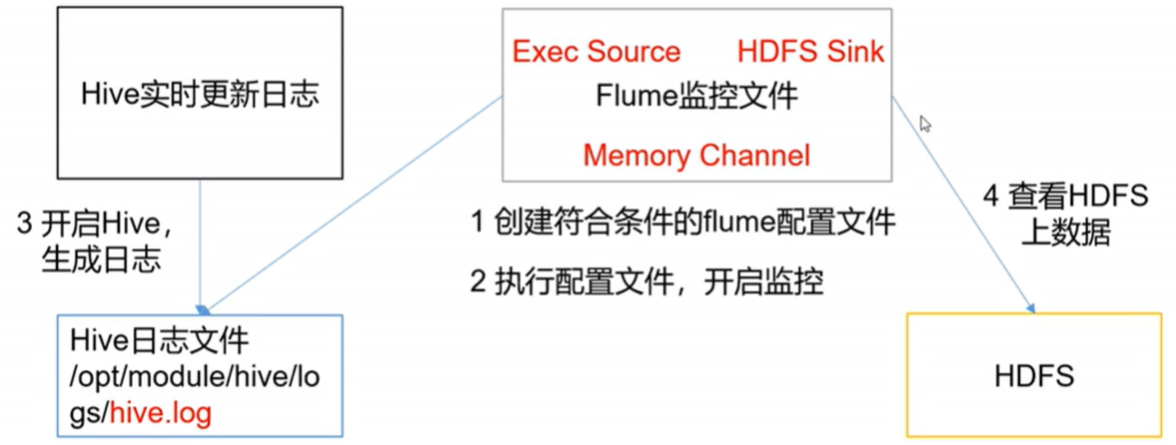
# Flume4 实时监控单个追加文件

## 需求

案例需求： 实时监控Hive日志，并上传到HDFS中

存在单点故障，因为如果agent故障后，可能会丢失数据

## 需求分析



## 实验步骤

### 第一个先提取日志数据输出到控制台

**创建 file-flume-logger.conf 文件**

```
# 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 /opt/module/hive/logs/hive.log

# 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
```

- 执行监控配置

  ```
  bon/flume-ng agent -c conf/ -f job/file-flume-logger.conf -n a1 -
  Dflume.root.logger=INFO,console
  ```

- 开启 Hadoop 和 Hive 并操作 Hive 产生日志

  ```
  [atguigu@hadoop102 hadoop-2.7.2]$ sbin/start-dfs.sh
  [atguigu@hadoop103 hadoop-2.7.2]$ sbin/start-yarn.sh
  [atguigu@hadoop102 hive]$ bin/hive
  hive (default)>
  ```

注意：

```
tail -f
等同于--follow=descriptor，根据文件描述符进行追踪，当文件改名或被删除，追踪停止

等同于--follow=name --retry，根据文件名进行追踪，并保持重试，即该文件被删除或改名后，如果再次创建相同的文件名，会继续追踪
```

## 第二个将数据日志转移到HDFS

- Flume 要想将数据输出到 HDFS，必须持有 Hadoop 相关 jar 包

  ```
  将 commons-configuration-1.6.jar、
  hadoop-auth-2.7.2.jar、
  hadoop-common-2.7.2.jar、
  hadoop-hdfs-2.7.2.jar、
  commons-io-2.4.jar、
  htrace-core-3.1.0-incubating.jar
  拷贝到/opt/module/flume/lib 文件夹下。
  ```

- 创建 flume-file-hdfs.conf 文件

  ```
  [atguigu@hadoop102 job]$ touch flume-file-hdfs.conf
  ```

  注：要想读取 Linux 系统中的文件，就得按照 Linux 命令的规则执行命令。由于 Hive日志在 Linux 系统中所以读取文件的类型选择：exec 即 execute 执行的意思。表示执行 Linux命令来读取文件。

  ```
  [atguigu@hadoop102 job]$ vim flume-file-hdfs.conf
  ```

  添加如下内容：

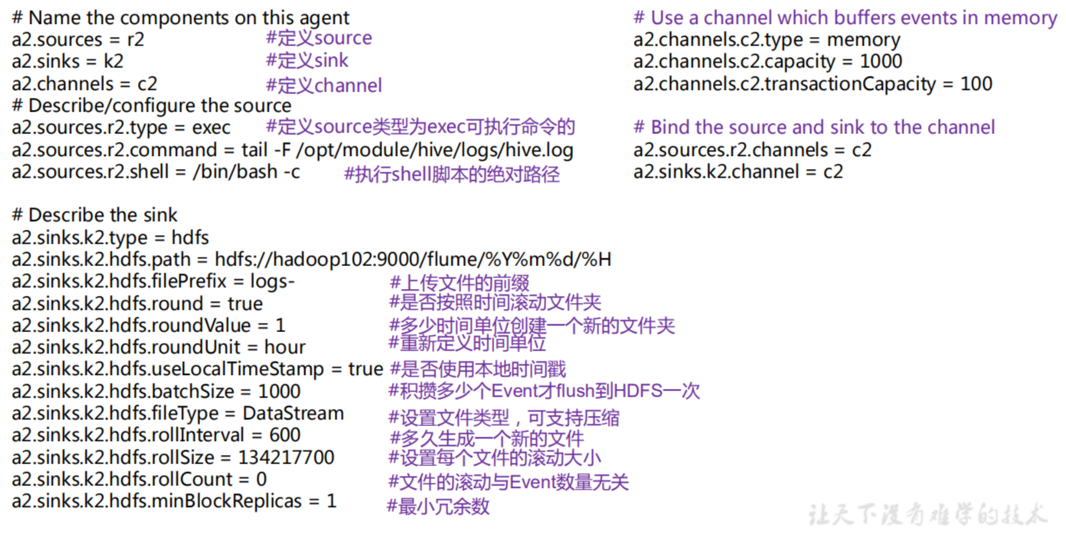
  ```
  # Name the components on this agent
  a2.sources = r2
  a2.sinks = k2
  a2.channels = c2
  ```

```
# Describe/configure the source
a2.sources.r2.type = exec
a2.sources.r2.command = tail -F /opt/module/hive/logs/hive.log
a2.sources.r2.shell = /bin/bash -c

# Describe the sink
a2.sinks.k2.type = hdfs
a2.sinks.k2.hdfs.path = hdfs://hadoop102:9000/flume/%Y%m%d/%H
#上传文件的前缀
a2.sinks.k2.hdfs.filePrefix = logs-
#是否按照时间滚动文件夹
a2.sinks.k2.hdfs.round = true
#多少时间单位创建一个新的文件夹
a2.sinks.k2.hdfs.roundValue = 1
#重新定义时间单位
a2.sinks.k2.hdfs.roundUnit = hour
#是否使用本地时间戳
a2.sinks.k2.hdfs.useLocalTimeStamp = true
#积攒多少个 Event 才 flush 到 HDFS 一次
a2.sinks.k2.hdfs.batchSize = 1000
#设置文件类型，可支持压缩
a2.sinks.k2.hdfs.fileType = DataStream
#多久生成一个新的文件
a2.sinks.k2.hdfs.rollInterval = 60
#设置每个文件的滚动大小
a2.sinks.k2.hdfs.rollSize = 134217700
#文件的滚动与 Event 数量无关
a2.sinks.k2.hdfs.rollCount = 0
#最小冗余数
a2.sinks.k2.hdfs.minBlockReplicas = 1

# Use a channel which buffers events in memory
a2.channels.c2.type = memory
a2.channels.c2.capacity = 1000
a2.channels.c2.transactionCapacity = 100

# Bind the source and sink to the channel
a2.sources.r2.channels = c2
a2.sinks.k2.channel = c2
```



```
# Name the components on this agent
a2.sources = r2                 #定义source
a2.sinks = k2                   #定义sink
a2.channels = c2                #定义channel
# Describe/configure the source
a2.sources.r2.type = exec       #定义source类型为exec可执行命令的
a2.sources.r2.command = tail -F /opt/module/hive/logs/hive.log
a2.sources.r2.shell = /bin/bash -c      #执行shell脚本的绝对路径

# Describe the sink
a2.sinks.k2.type = hdfs
a2.sinks.k2.hdfs.path = hdfs://hadoop102:9000/flume/%Y%m%d/%H
a2.sinks.k2.hdfs.filePrefix = logs-          #上传文件的前缀
a2.sinks.k2.hdfs.round = true                #是否按照时间滚动文件夹
a2.sinks.k2.hdfs.roundValue = 1              #多少时间单位创建一个新的文件夹
a2.sinks.k2.hdfs.roundUnit = hour            #重新定义时间单位
a2.sinks.k2.hdfs.useLocalTimeStamp = true #是否使用本地时间戳
a2.sinks.k2.hdfs.batchSize = 1000            #积攒多少个Event才flush到HDFS一次
a2.sinks.k2.hdfs.fileType = DataStream       #设置文件类型，可支持压缩
a2.sinks.k2.hdfs.rollInterval = 600          #多久生成一个新的文件
a2.sinks.k2.hdfs.rollSize = 134217700        #设置每个文件的滚动大小
a2.sinks.k2.hdfs.rollCount = 0               #文件的滚动与Event数量无关
a2.sinks.k2.hdfs.minBlockReplicas = 1        #最小冗余数

# Use a channel which buffers events in memory
a2.channels.c2.type = memory
a2.channels.c2.capacity = 1000
a2.channels.c2.transactionCapacity = 100

# Bind the source and sink to the channel
a2.sources.r2.channels = c2
a2.sinks.k2.channel = c2
```

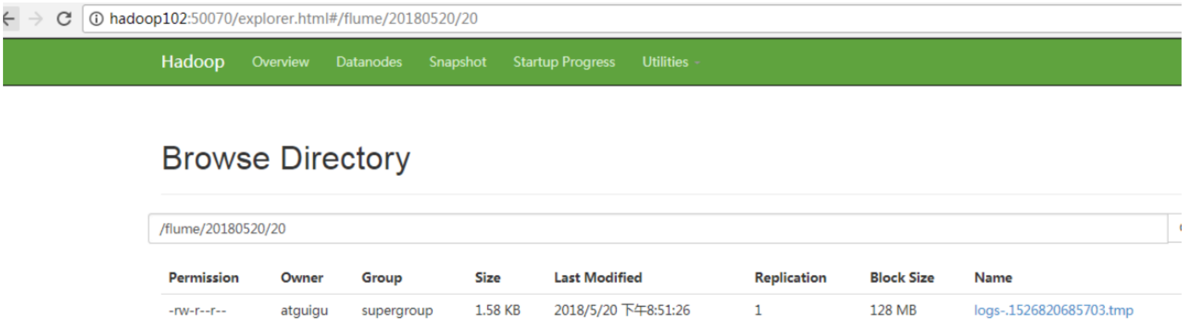
- 执行监控配置

```
[atguigu@hadoop102 flume]$ bin/flume-ng agent --conf conf/ --name a2 --conf-file job/flume-file-hdfs.conf
```

- 开启 Hadoop 和 Hive 并操作 Hive 产生日志

```
[atguigu@hadoop102 hadoop-2.7.2]$ sbin/start-dfs.sh
[atguigu@hadoop103 hadoop-2.7.2]$ sbin/start-yarn.sh
[atguigu@hadoop102 hive]$ bin/hive
hive (default)>
```

- 在 HDFS 上查看文件。



Browse Directory

/flume/20180520/20

| Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name |
|---|---|---|---|---|---|---|---|
| -rw-r--r-- | atguigu | supergroup | 1.58 KB | 2018/5/20 下午8:51:26 | 1 | 128 MB | logs-.1526820685703.tmp |