大数据技术之标题

一、 Flume 简介

1) Flume 提供一个分布式的， 可靠的， 对大数据量的日志进行高效收集、 聚集、 移动的服务，Flume 只能在 Unix 环境下运行。

2) Flume 基于流式架构，容错性强，也很灵活简单。

3) Flume、Kafka 用来实时进行数据收集， Spark、Storm 用来实时处理数据， impala 用来实时查询。

二、 Flume 角色

2.1、 Source

用于采集数据， Source 是产生数据流的地方， 同时 Source 会将产生的数据流传输到 Channel，这个有点类似于 Java IO 部分的 Channel。

2.2、 Channel

用于桥接 Sources 和 Sinks，类似于一个队列。

2.3、 Sink

从 Channel 收集数据，将数据写到目标源 (可以是下一个 Source, 也可以是 HDFS 或者 HBase)。

2.4、 Event

传输单元， Flume 数据传输的基本单元，以事件的形式将数据从源头送至目的地。

三、 Flume 传输过程

source 监控某个文件或数据流，数据源产生新的数据，拿到该数据后，将数据封装在一个 Event 中，并 put 到 channel 后 commit 提交， channel 队列先进先出， sink 去 channel 队列中拉取数据，然后写入到 hdfs 或者 HBase 中。

四、 **Flume** 部署及使用

4.1、文件配置

flume-env.sh 涉及修改项：

```
JAVA_HOME=/home/admin/modules/jdk1.8.0_121
```

4.2、案例

4.2.1、案例一

目标： Flume 监控一端 Console，另一端 Console 发送消息，使被监控端实时显示。

分步实现：

1) 创建 **Flume Agent** 配置文件 **flume-telnet.conf**

```
# 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
```

2) 安装 **telnet** 工具

```
$ sudo rpm -ivh telnet-server-0.17-59.el7.x86_64.rpm

$ sudo rpm -ivh telnet-0.17-59.el7.x86_64.rpm
```

3) 判断 **44444** 端口是否被占用

```
$ netstat -an | grep 44444
```

4) 先开启 **flume** 先听端口

```
$ bin/flume-ng agent --conf conf/ --name a1 --conf-file conf/flume-telnet.conf

-Dflume.root.logger==INFO,console
```

5) 使用 **telnet** 工具向本机的 **44444** 端口发送内容

```
$ telnet localhost 44444
```

4.2.2、案例二

目标：实时监控 hive 日志，并上传到 HDFS 中

分步实现：

1) 拷贝 **Hadoop** 相关 **jar** 到 **Flume** 的 **lib** 目录下（要学会根据自己的目录和版本查找 **jar** 包）

```
$ cp share/hadoop/common/lib/hadoop-auth-2.5.0-cdh5.3.6.jar ./lib/

$ cp share/hadoop/common/lib/commons-configuration-1.6.jar ./lib/

$ cp share/hadoop/mapreduce1/lib/hadoop-hdfs-2.5.0-cdh5.3.6.jar ./lib/

$ cp share/hadoop/common/hadoop-common-2.5.0-cdh5.3.6.jar ./lib/

$ cp ./share/hadoop/hdfs/lib/htrace-core-3.1.0-incubating.jar ./lib/

$ cp ./share/hadoop/hdfs/lib/commons-io-2.4.jar ./lib/
```

尖叫提示： 标红的 jar 为 1.99 版本 flume 必须引用的 jar

2) 创建 **flume-hdfs.conf** 文件

```
# 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

/home/admin/modules/apache-flume-1.7.0-bin/my_custom_logs.txt

a2.sources.r2.shell = /bin/bash -c



# Describe the sink

a2.sinks.k2.type = hdfs

a2.sinks.k2.hdfs.path = hdfs://linux01:8020/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 = 600

#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
```

3) 执行监控配置

```
$ bin/flume-ng agent --conf conf/ --name a2 --conf-file conf/flume-hdfs.conf
```

4.2.3、案例三

目标：使用 flume 监听整个目录的文件

分步实现：

1) 创建配置文件 flume-dir.conf

```
a3.sources = r3

a3.sinks = k3

a3.channels = c3



# Describe/configure the source
```

```
a3.sources.r3.type = spooldir

a3.sources.r3.spoolDir = /home/admin/modules/apache-flume-1.7.0-bin/upload

a3.sources.r3.fileHeader = true

#忽略所有以 .tmp 结尾的文件，不上传

a3.sources.r3.ignorePattern = ([^ ]*\.tmp)



# Describe the sink

a3.sinks.k3.type = hdfs

a3.sinks.k3.hdfs.path = hdfs://linux01:8020/flume/upload/%Y%m%d/%H

#上传文件的前缀

a3.sinks.k3.hdfs.filePrefix = upload-

#是否按照时间滚动文件夹

a3.sinks.k3.hdfs.round = true

#多少时间单位创建一个新的文件夹

a3.sinks.k3.hdfs.roundValue = 1

#重新定义时间单位

a3.sinks.k3.hdfs.roundUnit = hour

#是否使用本地时间戳

a3.sinks.k3.hdfs.useLocalTimeStamp = true

#积攒多少个 Event 才 flush 到 HDFS 一次

a3.sinks.k3.hdfs.batchSize = 1000

#设置文件类型，可支持压缩

a3.sinks.k3.hdfs.fileType = DataStream

#多久生成一个新的文件

a3.sinks.k3.hdfs.rollInterval = 600

#设置每个文件的滚动大小

a3.sinks.k3.hdfs.rollSize = 134217700

#文件的滚动与 Event 数量无关
```

```
a3.sinks.k3.hdfs.rollCount = 0

#最小冗余数

a3.sinks.k3.hdfs.minBlockReplicas = 1



# Use a channel which buffers events in memory

a3.channels.c3.type = memory

a3.channels.c3.capacity = 1000

a3.channels.c3.transactionCapacity = 100



# Bind the source and sink to the channel

a3.sources.r3.channels = c3

a3.sinks.k3.channel = c3
```

2）执行测试

```
$ bin/flume-ng agent --conf conf/ --name a3 --conf-file conf/flume-dir.conf &
```

尖叫提示：

在使用 Spooling Directory Source 时

1）不要在监控目录中创建并持续修改文件

2）上传完成的文件会以 .COMPLETED 结尾

3）被监控文件夹每 600 毫秒扫描一次文件变动

4.2.4、案例四（待完善）

目标：使用 Java API 实现 Flume 开发