Flume知识点整理

Sources

netcat Source

decr:端口source

```
a1.sources = r1
a1.sources.r1.type = netcat
a1.sources.r1.bind = 主机ip
a1.sources.r1.port = 端口号
```

avro Source

decr:RPCsource

```
a1.sources = r1
a1.sources.r1.type = avro
a1.sources.r1.bind = 主机ip
a1.sources.r1.port = 端口号
```

thrift Source

decr:监听外部thrift流事件

```
a1.sources = r1
a1.sources.r1.type = thrift
a1.sources.r1.bind = 主机ip
a1.sources.r1.port = 端口号
```

exec Source

decr:将终端可执行命令的输出结果作为source

```
a1.sources = r1
a1.sources.r1.type = exec
a1.sources.r1.command = 终端可执行命令
```

spooling directory Source

decr:监听一个文件夹，若文件夹下有新文件产生则读入，其不能察觉文件内部的变化，容易丢失数据

```
a1.sources = src-1
a1.sources.src-1.type = spooldir
a1.sources.src-1.spoolDir = 目标目录
# 是否在最后的文件名中添加文件绝对路径
a1.sources.src-1.fileHeader = true
```

taildir Source

decr:监听一个文件夹，不仅可以监听是否有新文件产生，还可以监听单个文件的内部变化

```
a1.sources = r1
a1.sources.r1.type = TAILDIR
a1.sources.r1.positionFile = /var/log/flume/taildir_position.json
a1.sources.r1.filegroups = f1 f2
a1.sources.r1.filegroups.f1 = /var/log/test1/example.log
a1.sources.r1.headers.f1.headerKey1 = value1
a1.sources.r1.filegroups.f2 = /var/log/test2/.*log.*
a1.sources.r1.headers.f2.headerKey1 = value2
a1.sources.r1.headers.f2.headerKey2 = value2-2
a1.sources.r1.fileHeader = true
a1.sources.ri.maxBatchCount = 1000
```

kafka Source

decr:从kafka topic中读取数据作为数据源

```
a1.sources.r1.type = org.apache.flume.source.kafka.KafkaSource
a1.sources.r1.kafka.bootstrap.servers = kafka节点1:9092
# 指定从哪个主题中获取数据
a1.sources.r1.kafka.topics = topic名称
或者
a1.sources.r1.kafka.topics.regex = 正则表达式匹配主题名称
```

netcat udp Source

decr:通过udp协议获取数据， udp面向无连接，有损失但速度快

```
a1.sources = r1
a1.sources.r1.type = netcatudp
a1.sources.r1.bind = 主机ip
a1.sources.r1.port = 端口号
```

http Source

decr:通过http协议和指定端口获取指定接口数据

```
a1.sources = r1
a1.sources.r1.type = http
a1.sources.r1.port = 5140
a1.sources.r1.handler = org.example.rest.RestHandler
a1.sources.r1.handler.nickname = random props
a1.sources.r1.HttpConfiguration.sendServerVersion = false
a1.sources.r1.ServerConnector.idleTimeout = 300
```

stress Source

decr:通过短时间大量的event测试系统性能，一般用于压力测试

```
a1.sources = r1
a1.sources.r1.type = org.apache.flume.source.StressSource
a1.sources.r1.size = 10240
a1.sources.r1.maxTotalEvents = 1000000
```

legacy Source

decr:允许不同版本flume之间互相传输agent

avro legacy source

```
a1.sources = r1
a1.sources.r1.type = org.apache.flume.source.avroLegacy.AvroLegacySource
a1.sources.r1.host = 机器ip
a1.sources.r1.bind = 端口号
```

thrift legacy source

```
a1.sources = r1
a1.sources.r1.type = org.apache.flume.source.thriftLegacy.ThriftLegacySource
a1.sources.r1.host = 机器ip
a1.sources.r1.bind = 端口号
```

custom Source

decr:用自己编写的java类运行结果作为数据源

```
a1.sources = r1
a1.sources.r1.type = org.example.MySource
```

Channels

memory Channel

decr:内存channel

```
a1.channels = c1
a1.channels.c1.type = memory
# channel中能存储的最大event数量
a1.channels.c1.capacity = 1000
# 一次事务中写入或读取的最大event数量
a1.channels.c1.transactionCapacity = 100
```

jdbc Channel

decr:以数据库作为缓冲管道，目前只支持derby

```
a1.channels = c1
a1.channels.c1.type = jdbc
a1.channels.c1.driver.class = 驱动类名称
a1.channels.c1.driver.url = 连接数据库url
a1.channels.c1.db.username = 数据库用户名
a1.channels.c1.db.password = 数据库用户对应的密码
```

kafka Channel

decr:以kafka作为event缓冲管道

```
a1.channels.channel1.type = org.apache.flume.channel.kafka.KafkaChannel
a1.channels.channel1.kafka.bootstrap.servers = kafka-1:9092,kafka-2:9092,kafka-3:9092
a1.channels.channel1.kafka.topic = channel1
```

file Channel

decr:以本地文件系统作为events缓冲管道

```
a1.channels = c1
a1.channels.c1.type = file
a1.channels.c1.checkpointDir = 检查点存放目录
a1.channels.c1.dataDirs = 缓冲文件目录
```

custom Channel

decr:以自己编写的java类作为events缓冲管道

```
a1.channels = c1
a1.channels.c1.type = 自己编写的类路径
```

Sinks

hdfs Sink

decr:将event存入hdfs中

```
a1.sinks = k1
a1.sinks.k1.type = hdfs
a1.sinks.k1.hdfs.path = /flume/events/%Y-%m-%d/%H%M/%S
a1.sinks.k1.hdfs.filePrefix = events-
a1.sinks.k1.hdfs.round = true
a1.sinks.k1.hdfs.roundValue = 10
a1.sinks.k1.hdfs.roundUnit = minute
```

hive Sink

decr:将event存入hive中

```
a1.sinks = k1
a1.sinks.k1.type = hive
a1.sinks.k1.hive.metastore = thrift://127.0.0.1:9083
a1.sinks.k1.hive.database = hive数据库名称
a1.sinks.k1.hive.table = hive数据表名称
```

logger Sink

decr:控制台打印sink，一般用于测试

```
a1.sinks = k1
# logger类型的channel 可以将数据打印在控制台
a1.sinks.k1.type = logger
```

avro Sink

decr:将数据传入本地端口

```
a1.sinks = k1
a1.sinks.k1.type = avro
a1.sinks.k1.hostname = 主机ip
a1.sinks.k1.port = 端口号
```

thrift Sink

```
a1.sinks = k1
a1.sinks.k1.type = thrift
a1.sinks.k1.hostname = 主机ip
a1.sinks.k1.port = 端口号
```

file Sink

decr:将event存入本地文件系统中

```
a1.sinks = k1
a1.sinks.k1.type = file_roll
a1.sinks.k1.sink.directory = 目标路径
```

kafka Sink

decr:将event存入kafka topic中

```
a1.sinks = k1
a1.sinks.k1.type = org.apache.flume.sink.kafka.KafkaSink
a1.sinks.k1.kafka.topic = topic名称
a1.sinks.k1.kafka.bootstrap.servers = kafka节点:9092
a1.sinks.k1.kafka.flumeBatchSize = 20
a1.sinks.k1.kafka.producer.acks = 1
a1.sinks.k1.kafka.producer.linger.ms = 1
a1.sinks.k1.kafka.producer.compression.type = snappy
```

http Sink

decr:将event以post方式提交给指定接口

```
a1.sinks = k1
a1.sinks.k1.type = http
a1.sinks.k1.endpoint = 接口url
```

custom Sink

decr：将event传入自己编写的数据处理类

```
a1.sinks = k1
a1.sinks.k1.type = 自己编写的类路径
```