文件收集框架Flume

谭唐华

课程大纲

- Flume的功能
- Flume的部署
- Flume的介绍

Flume 功能

- Flume is a **distributed, reliable, and available** service for efficiently collecting, aggregating, and moving large amounts of log data.

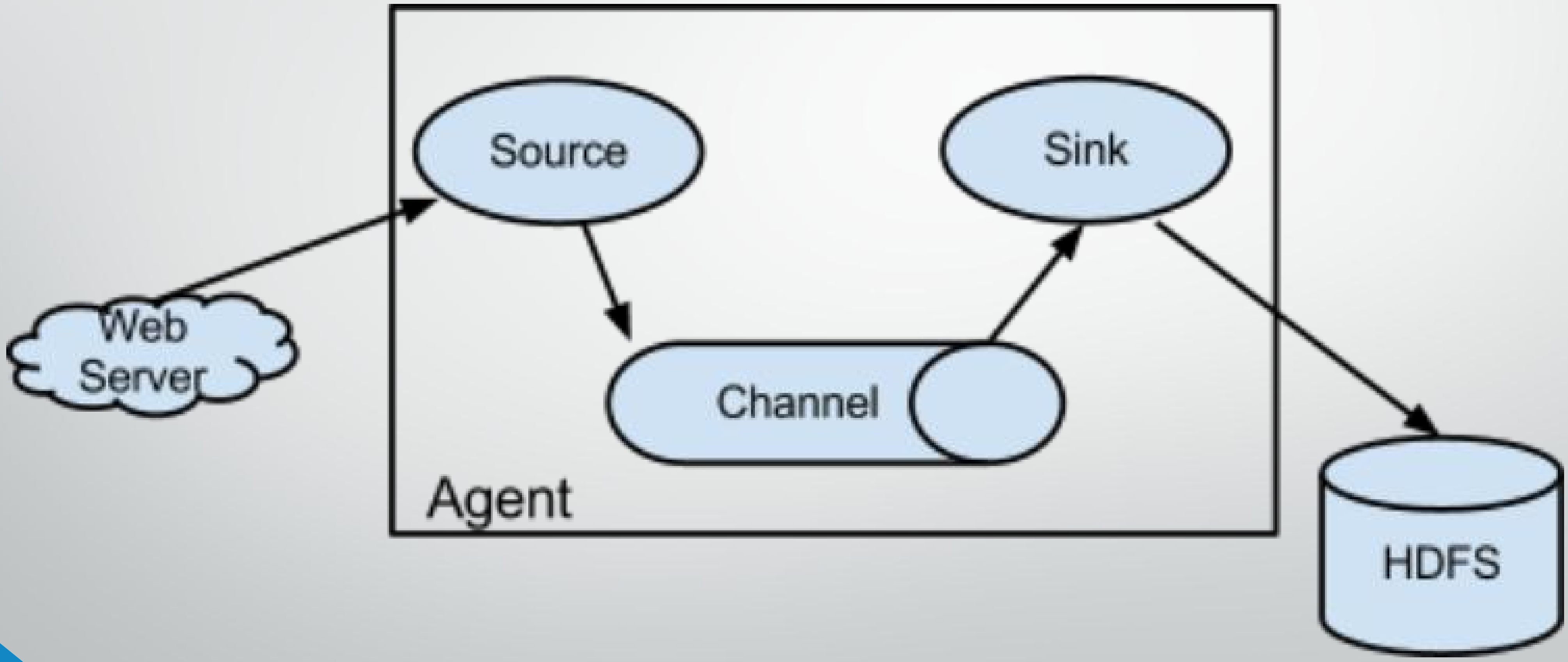
- It has a simple and flexible architecture based on streaming data flows. It is **robust**（健壮） **and fault tolerant** （容错）with tunable reliability mechanisms and many **failover and recovery** mechanisms.

- It uses a simple extensible data model that allows for **online analytic application**.

Flume 功能

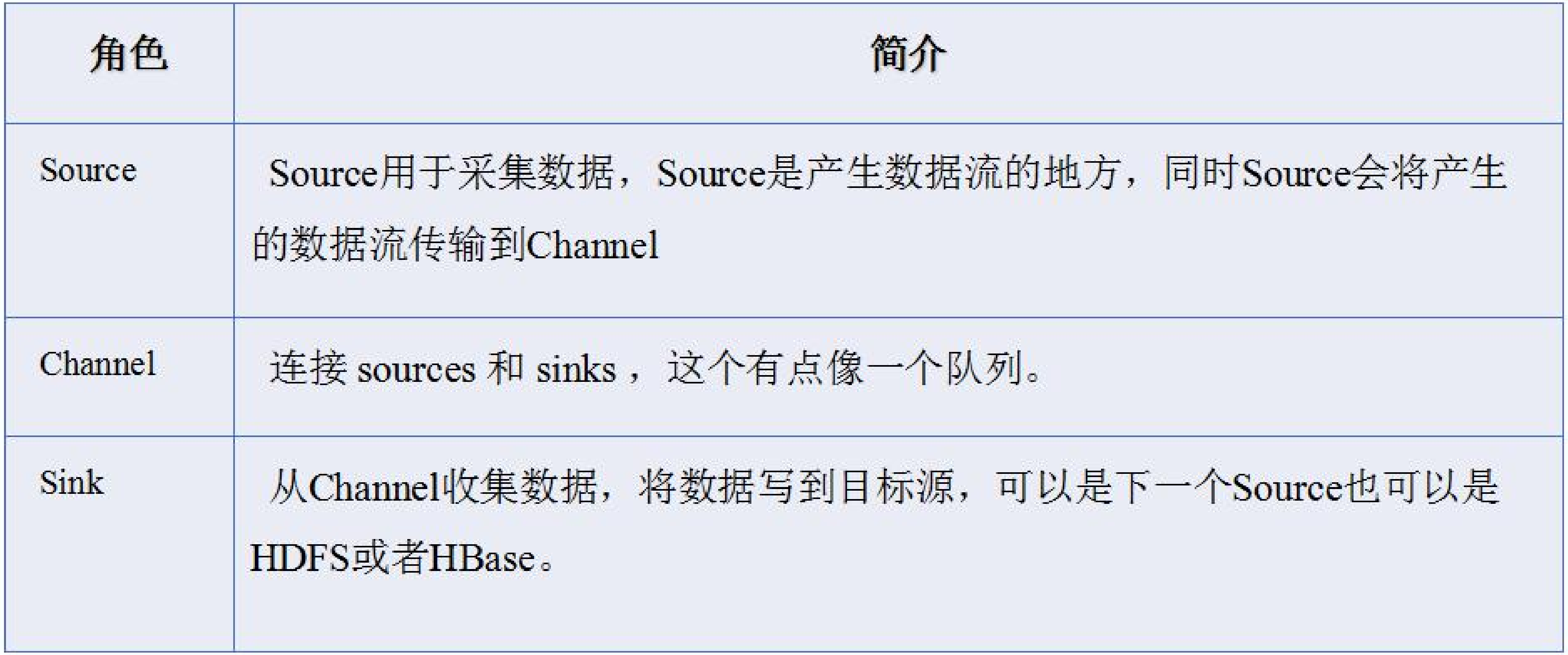
- Flume是一个分布式的，可靠的，可用的，非常有效率的对大数据量的日志数据进行收集、聚集、移动信息的服务。Flume仅仅运行在linux环境下。
- 它是一个基于流式的数据的非常简单的（就写一个配置文件就可以）、灵活的架构，它也是一个健壮的、容错的。它用一个简单的扩展数据模型用于在线实时应用分析。它的简单表现为：写个source、channel、sink，之后一条命令就能操作成功了。
- Flume、kafka实时进行数据收集，spark、storm实时去处理，impala实时去查询。

Apache Flume



Apache Flume

- Flume-ng只有一个角色的节点：agent的角色，agent有source、channel、sink组成。

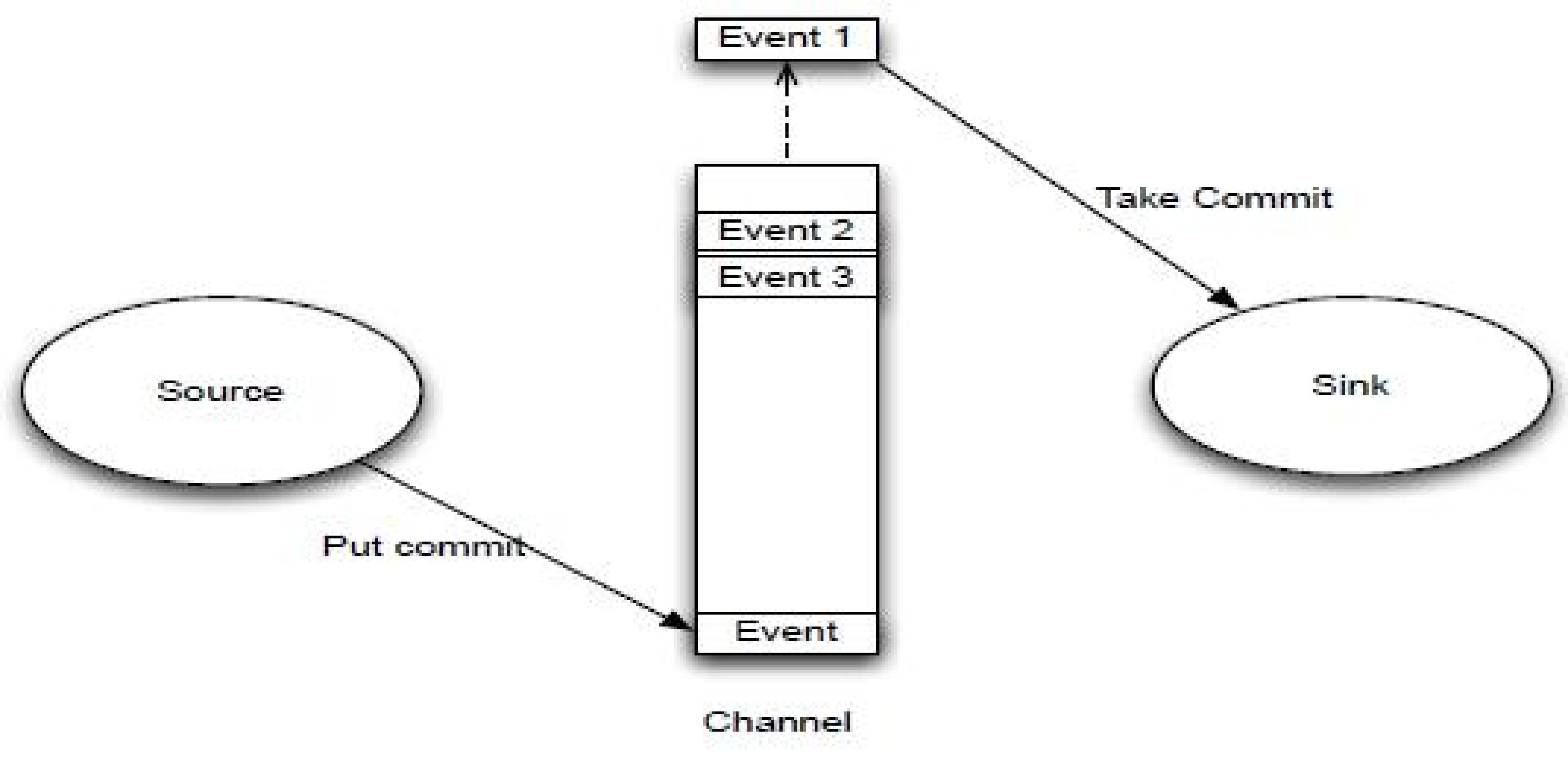


角色	简介
Source	Source用于采集数据，Source是产生数据流的地方，同时Source会将产生的数据流传输到Channel
Channel	连接 sources 和 sinks，这个有点像一个队列。
Sink	从Channel收集数据，将数据写到目标源，可以是下一个Source也可以是HDFS或者HBase。

Events

- Event是Flume数据传输的基本单元
- Flume以事件的形式将数据从源头传送到最终的目的
- Event由可选的header和载有数据的一个byte array构成
  - 载有的数据对flume是不透明的
  - **Header是容纳了key-value字符串对的无序集合，key在集合内是唯一的。**
  - Header可以在上下文路由中使用扩展

Channel/Event/Sink



- source监控某个文件，将数据拿到，封装在一个event当中，并put/commit到chennel当中，chennel是一个队列，队列的优点是先进先出，放好后尾部一个个event出来，sink主动去从chennel当中去拉数据，sink再把数据写到某个地方，比如HDFS上面去。

课程大纲

- Flume的功能
- Flume的部署
- Flume的介绍

System Requirements

- Flume是有Java语言编写的框架，并且仅仅只能运行在Unix系统上
- **Java Runtime Environment**
  - Java 1.6 or later (Java 1.7 Recommended)
- **Memory**
  - Sufficient memory for configurations used by sources, channels or sinks
- **Disk Space**
  - Sufficient disk space for configurations used by channels or sinks
- **Directory Permissions**
  - Read/Write permissions for directories used by agent

Flume 安装部署

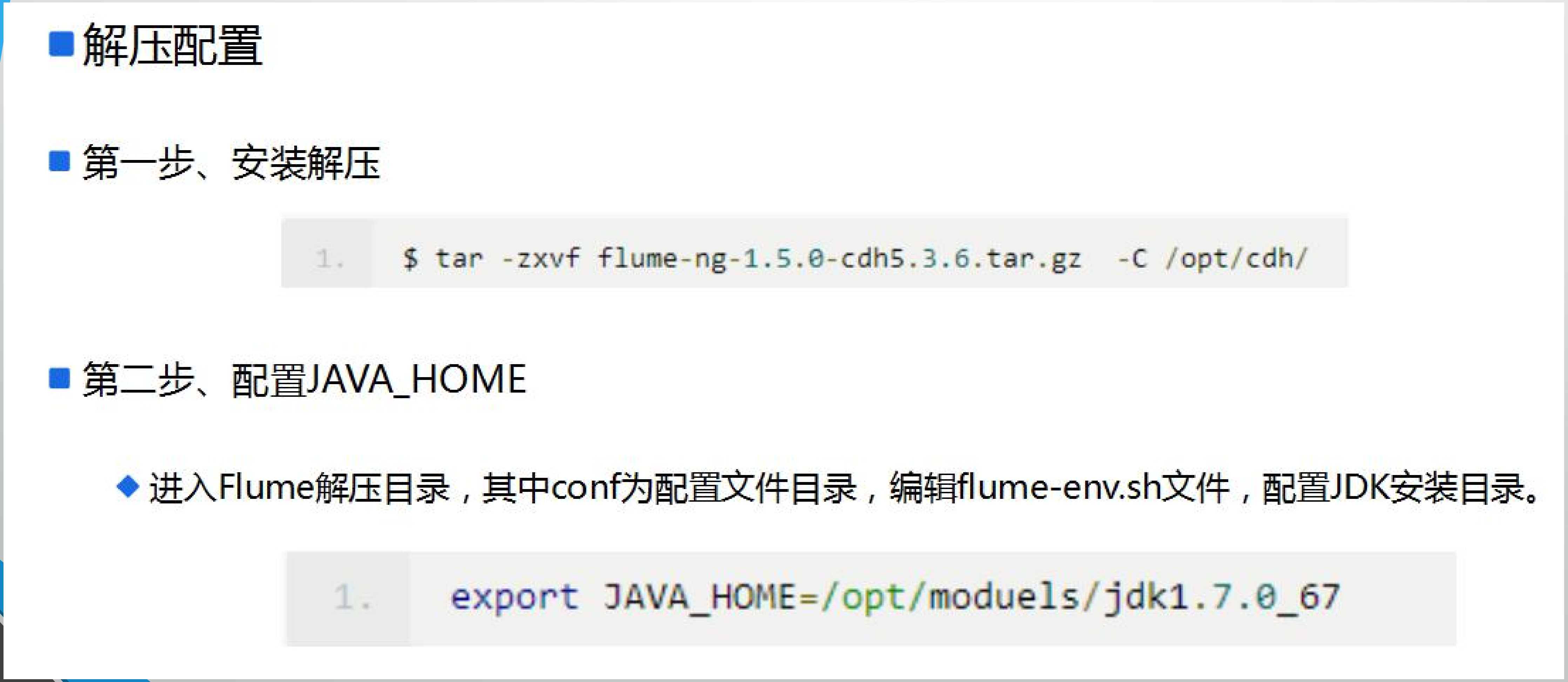
- 下载Flume，此处使用Cloudera发布的CDH5.3.6版本
  - **Apache 版本下载地址：**

    http://archive.apache.org/dist/flume/
  - **CDH 5.3.6下载地址:**

    http://archive.cloudera.com/cdh5/cdh/5/flume-ng-1.5.0-cdh5.3.6.tar.gz
  - **官方文档**

    http://archive.cloudera.com/cdh5/cdh/5/flume-ng-1.5.0-cdh5.3.6/FlumeUserGuide.html

Flume 安装部署



解压配置

- 第一步、安装解压

```
$ tar -zxvf flume-ng-1.5.0-cdh5.3.6.tar.gz  -C /opt/cdh/
```

- 第二步、配置JAVA_HOME

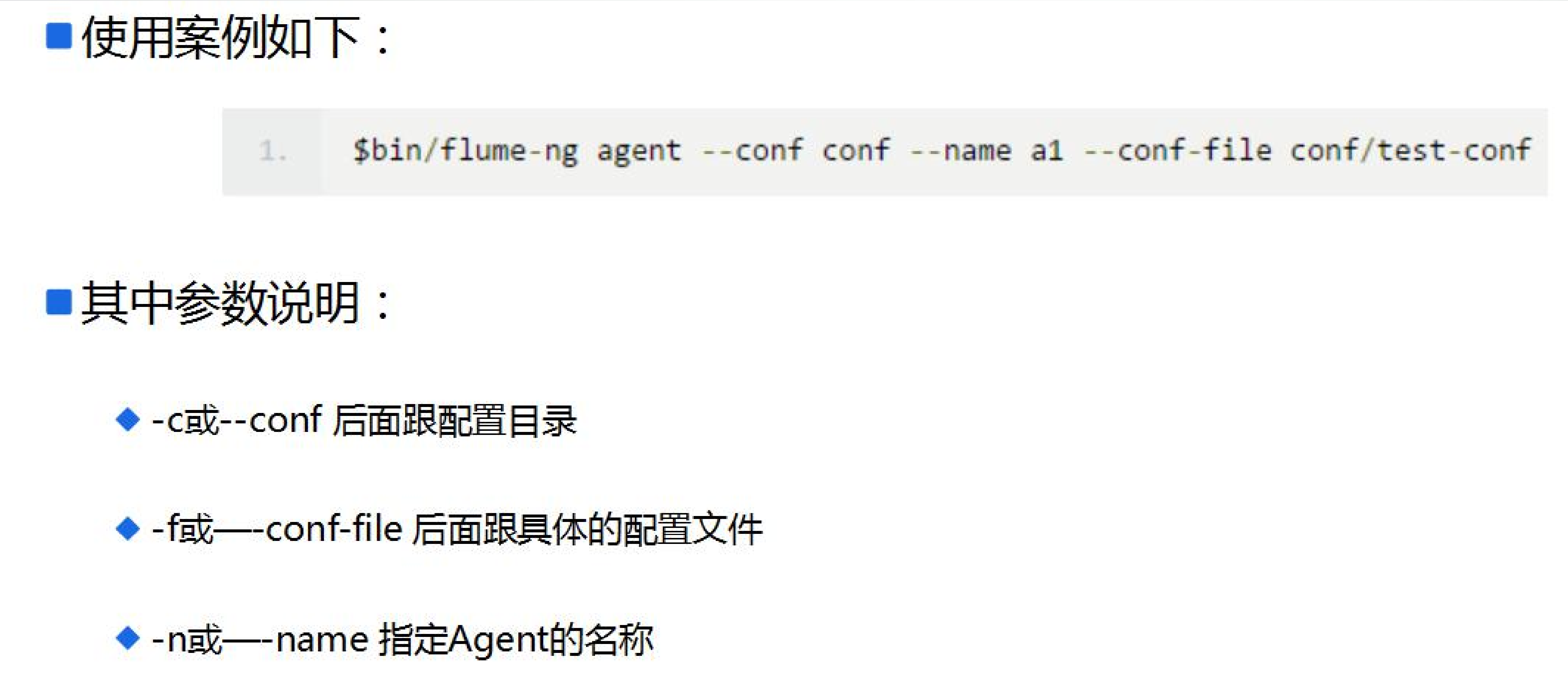
  - 进入Flume解压目录，其中conf为配置文件目录，编辑flume-env.sh文件，配置JDK安装目录。

```
export JAVA_HOME=/opt/moduels/jdk1.7.0_67
```

Flume 安装部署

- 第三步、flume-ng命令使用
  - Flume使用安装目录下bin/flume-ng进行执行程序，实时抽取数据，查看flume-ng使用：
    - bin/flume-ng

Flume 安装部署



- 使用案例如下：

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

- 其中参数说明：
  - -c或--conf 后面跟配置目录
  - -f或—-conf-file 后面跟具体的配置文件
  - -n或—-name 指定Agent的名称

课程大纲

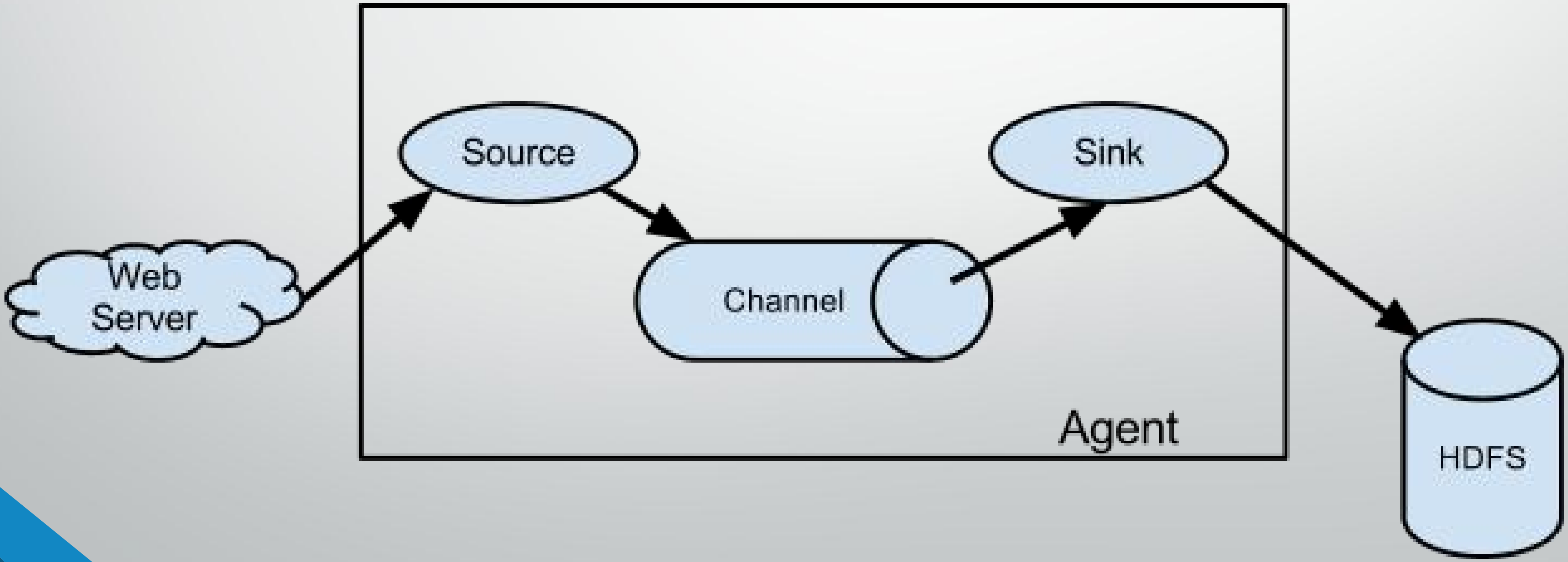
- Flume的功能
- Flume的部署
- Flume的介绍

Starting an agent

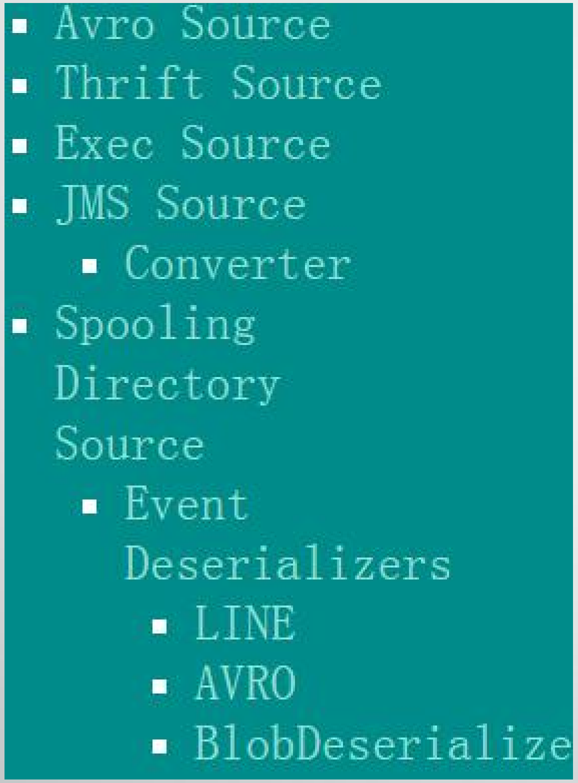
- Flume Agent的配置被存储在一个本地配置文件,这是一个根据**java属性文件**格式的文本文件，在这个配置文件中，包括了对source、sink、channel的属性配置，和其相关联形成数据流的配置。
- 案例功能描述：
  - Flume Agent实时监控端口，收集数据，将其以日志的形式打印在控制台。

Flume Agent

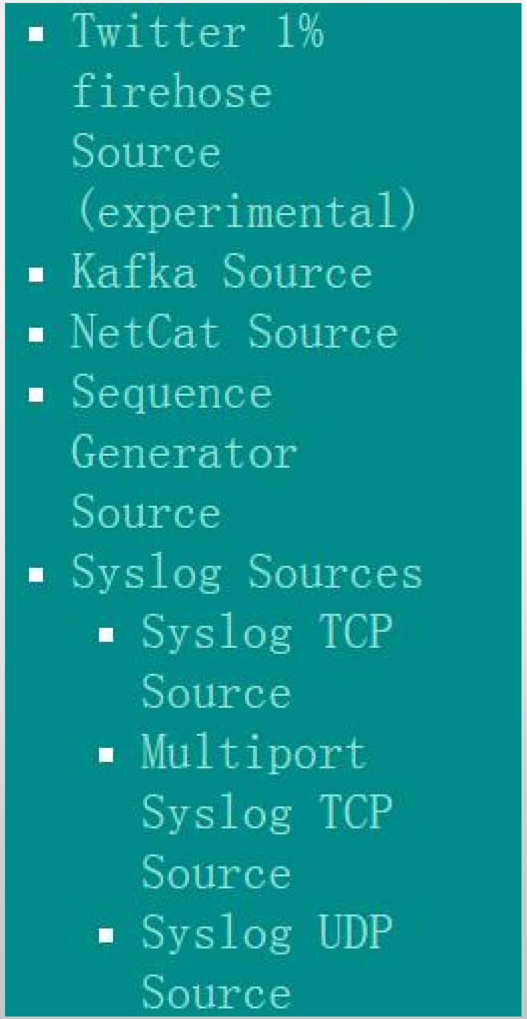
- Flume的开发就是编写配置文件，说白了就说Agent中Source、Channel和Sink的类型及属性。



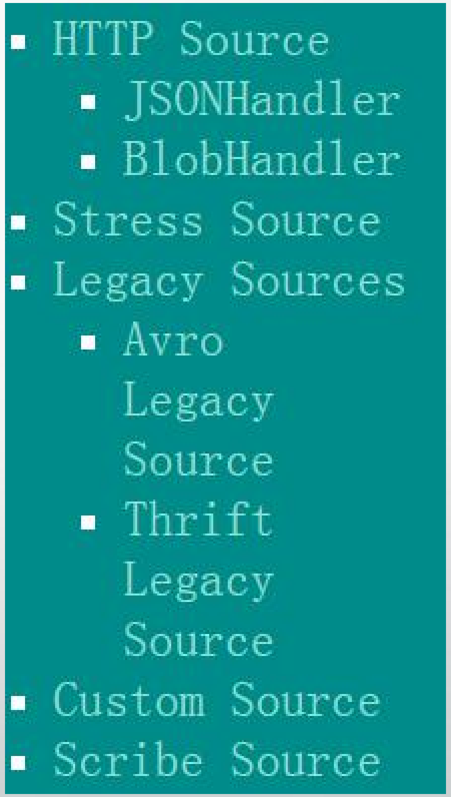
Flume之Sources



- Avro Source
- Thrift Source
- Exec Source
- JMS Source
  - Converter
- Spooling Directory Source
  - Event Deserializers
    - LINE
    - AVRO
    - BlobDeserialize

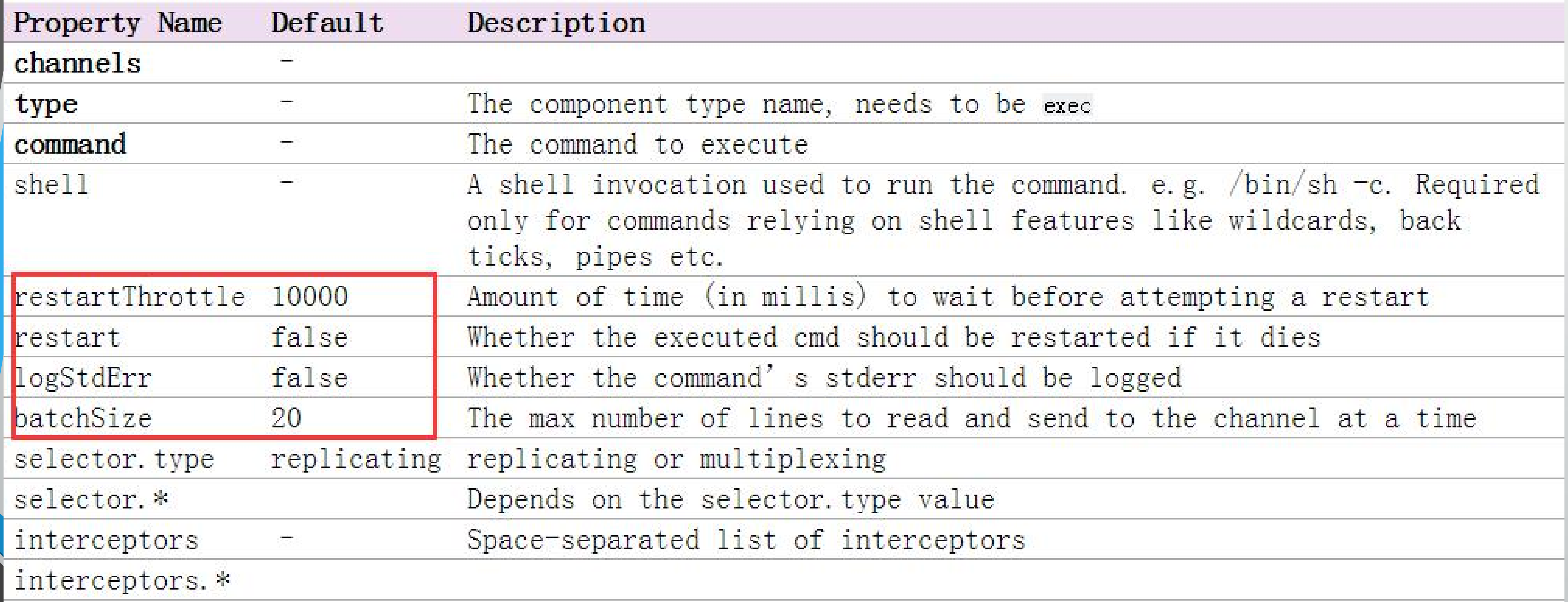


- Twitter 1% firehose Source (experimental)
- Kafka Source
- NetCat Source
- Sequence Generator Source
- Syslog Sources
  - Syslog TCP Source
  - Multiport Syslog TCP Source
  - Syslog UDP Source



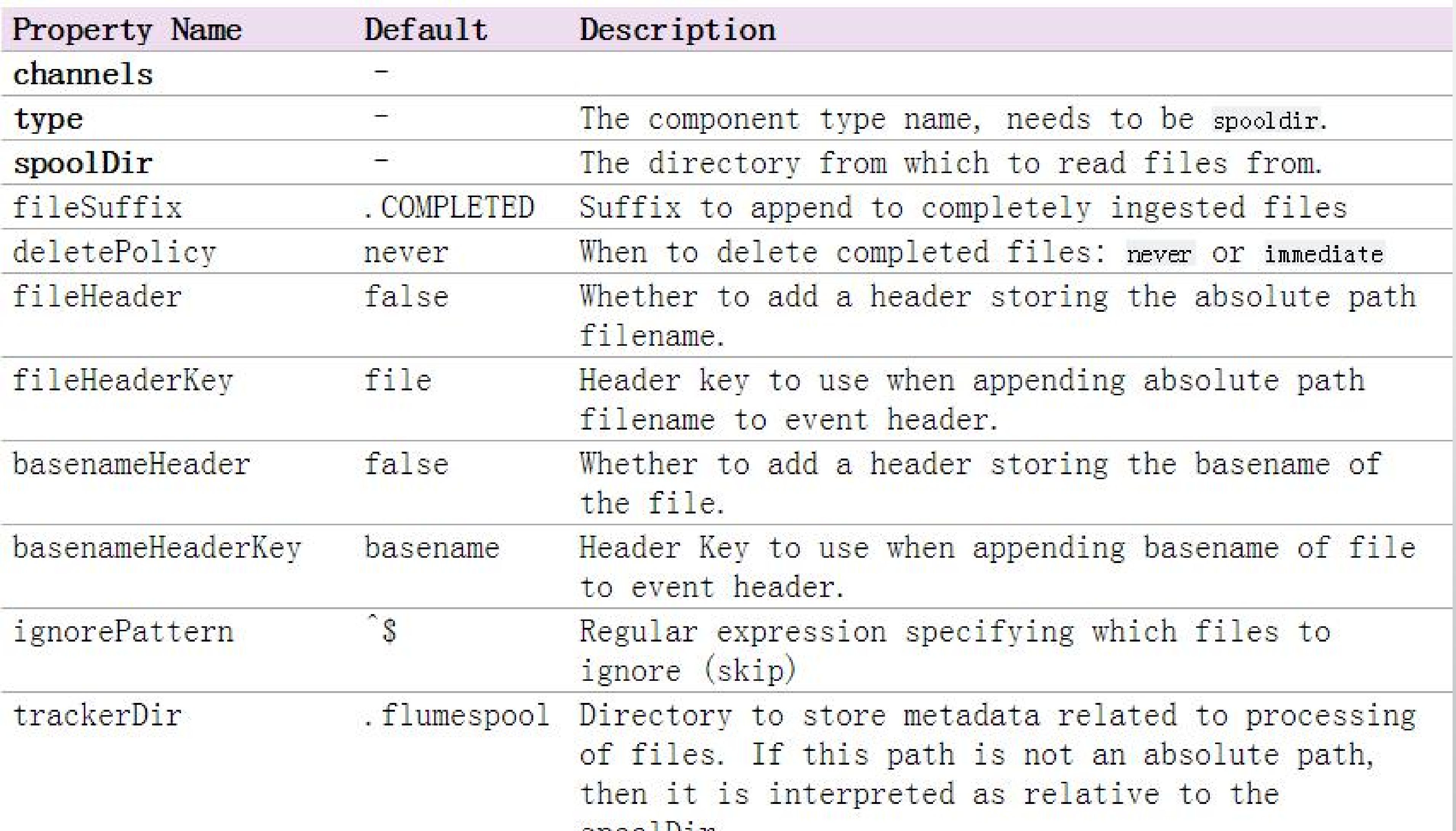
- HTTP Source
  - JSONHandler
  - BlobHandler
- Stress Source
- Legacy Sources
  - Avro Legacy Source
  - Thrift Legacy Source
- Custom Source
- Scribe Source

Exec Source



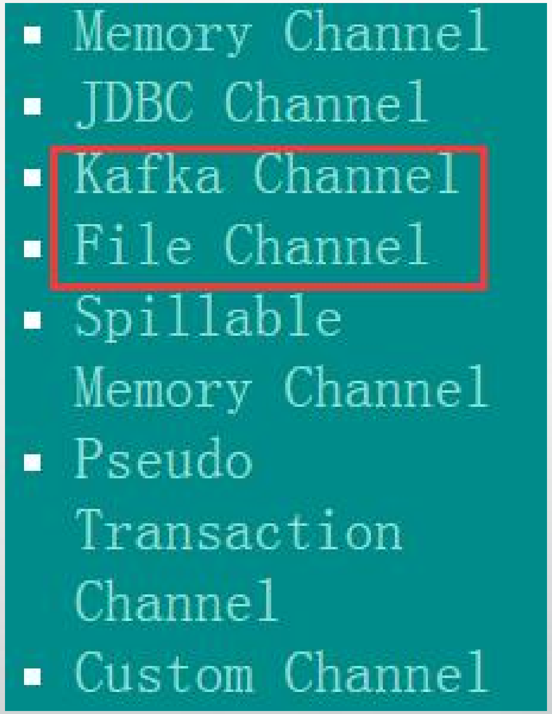
Property Name	Default	Description
channels	–	
type	–	The component type name, needs to be `exec`
command	–	The command to execute
shell	–	A shell invocation used to run the command. e.g. /bin/sh -c. Required only for commands relying on shell features like wildcards, back ticks, pipes etc.
restartThrottle	10000	Amount of time (in millis) to wait before attempting a restart
restart	false	Whether the executed cmd should be restarted if it dies
logStdErr	false	Whether the command’s stderr should be logged
batchSize	20	The max number of lines to read and send to the channel at a time
selector.type	replicating	replicating or multiplexing
selector.*		Depends on the selector.type value
interceptors	–	Space-separated list of interceptors
interceptors.*		

Spooling Directory Source



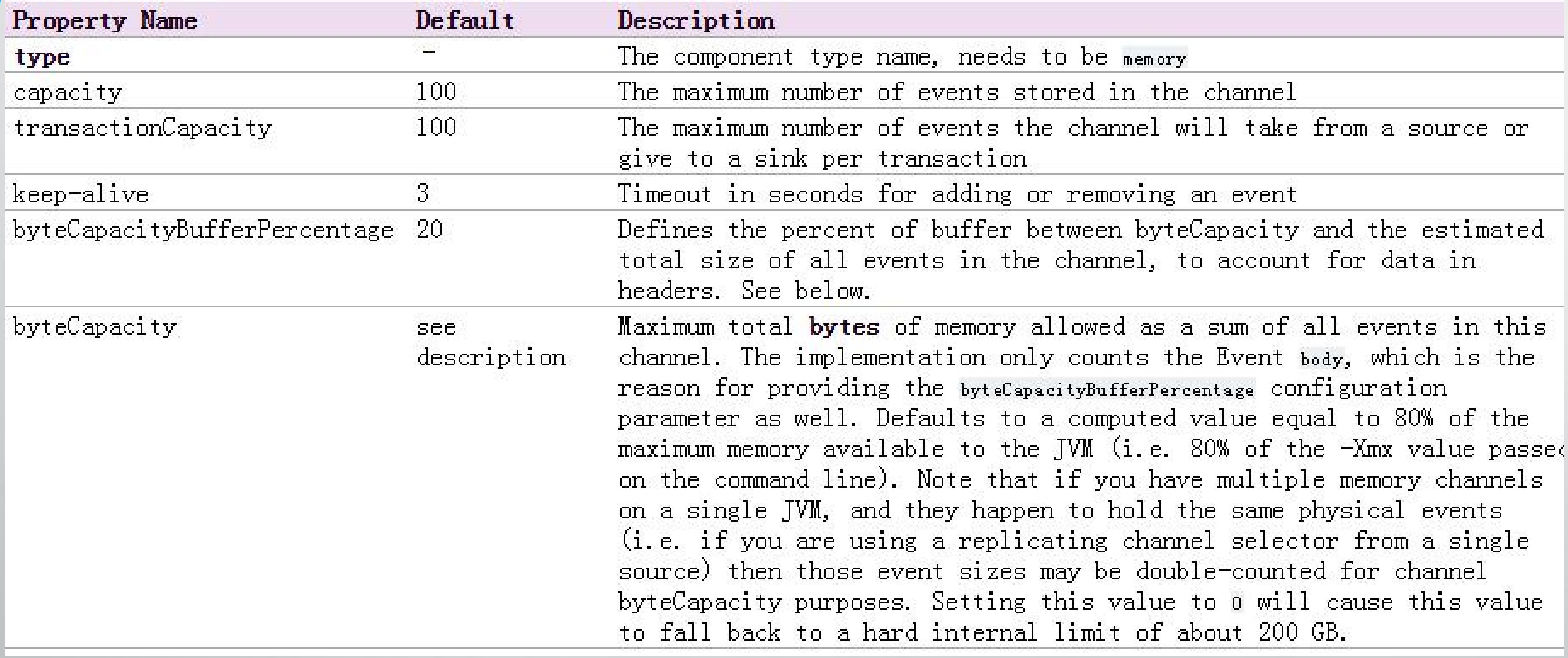
Property Name	Default	Description
channels	–	
type	–	The component type name, needs to be spooldir.
spoolDir	–	The directory from which to read files from.
fileSuffix	.COMPLETED	Suffix to append to completely ingested files
deletePolicy	never	When to delete completed files: never or immediate
fileHeader	false	Whether to add a header storing the absolute path filename.
fileHeaderKey	file	Header key to use when appending absolute path filename to event header.
basenameHeader	false	Whether to add a header storing the basename of the file.
basenameHeaderKey	basename	Header Key to use when appending basename of file to event header.
ignorePattern	^$	Regular expression specifying which files to ignore (skip)
trackerDir	.flumespool	Directory to store metadata related to processing of files. If this path is not an absolute path, then it is interpreted as relative to the spoolDir.

Flume之Channels

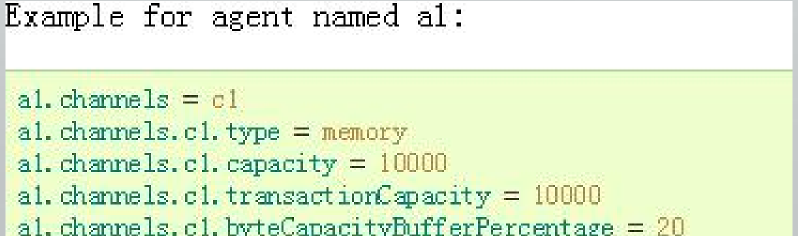


- Memory Channel
- JDBC Channel
- Kafka Channel
- File Channel
- Spillable Memory Channel
- Pseudo Transaction Channel
- Custom Channel

Memory Channel



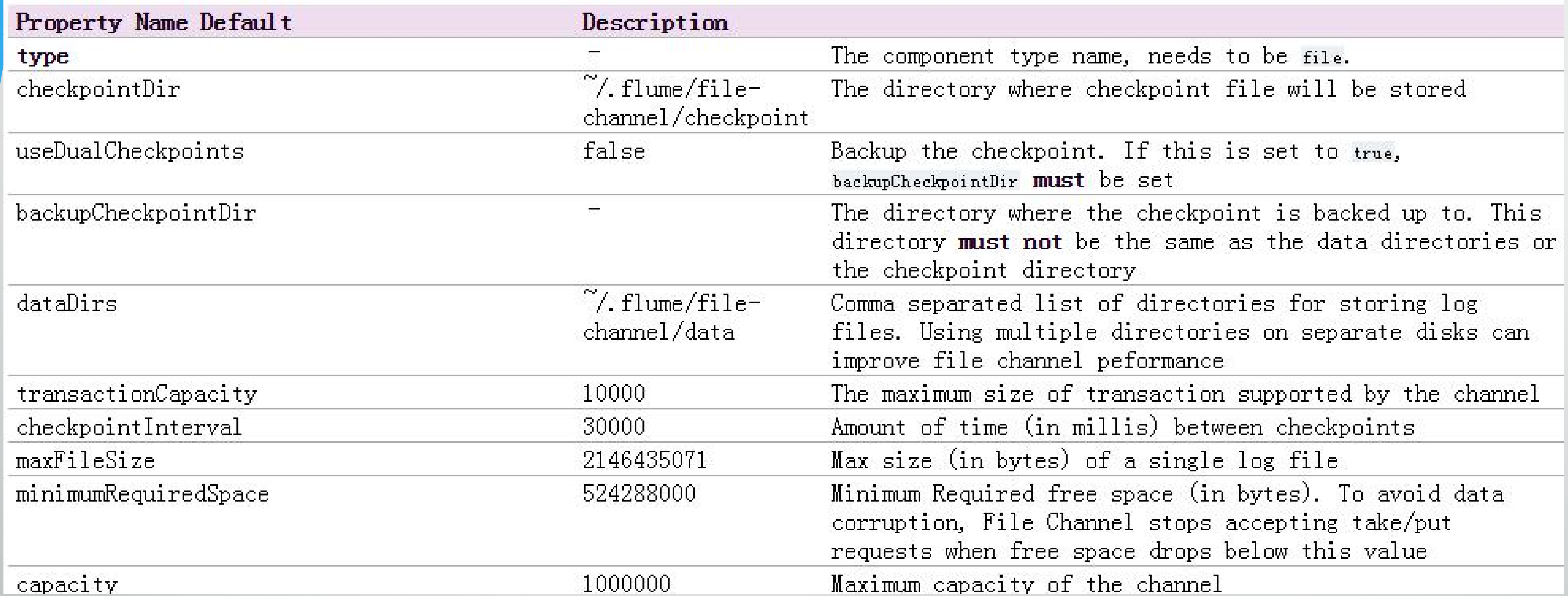
Property Name	Default	Description
type	–	The component type name, needs to be `memory`
capacity	100	The maximum number of events stored in the channel
transactionCapacity	100	The maximum number of events the channel will take from a source or give to a sink per transaction
keep-alive	3	Timeout in seconds for adding or removing an event
byteCapacityBufferPercentage	20	Defines the percent of buffer between byteCapacity and the estimated total size of all events in the channel, to account for data in headers. See below.
byteCapacity	see description	Maximum total **bytes** of memory allowed as a sum of all events in this channel. The implementation only counts the Event `body`, which is the reason for providing the `byteCapacityBufferPercentage` configuration parameter as well. Defaults to a computed value equal to 80% of the maximum memory available to the JVM (i.e. 80% of the -Xmx value passe on the command line). Note that if you have multiple memory channels on a single JVM, and they happen to hold the same physical events (i.e. if you are using a replicating channel selector from a single source) then those event sizes may be double-counted for channel byteCapacity purposes. Setting this value to `0` will cause this value to fall back to a hard internal limit of about 200 GB.



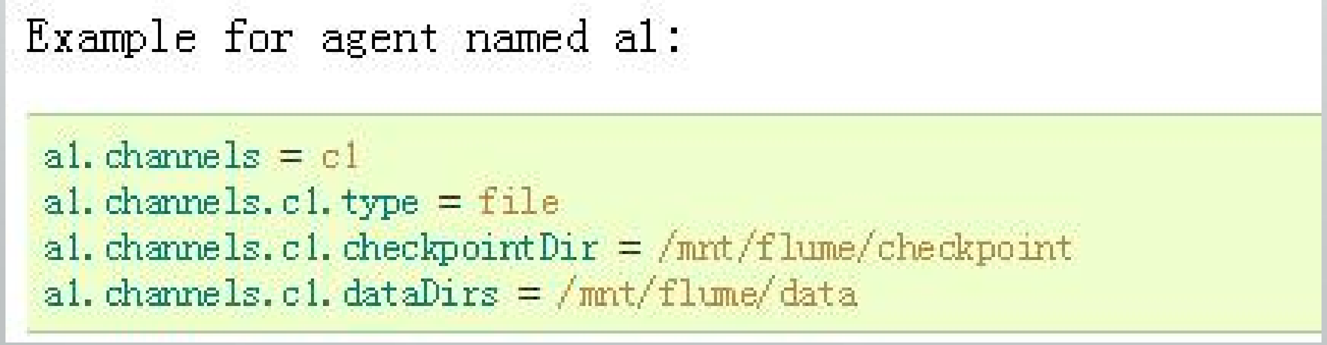
Example for agent named a1:

```
a1.channels = c1
a1.channels.c1.type = memory
a1.channels.c1.capacity = 10000
a1.channels.c1.transactionCapacity = 10000
a1.channels.c1.byteCapacityBufferPercentage = 20
```

File Channel



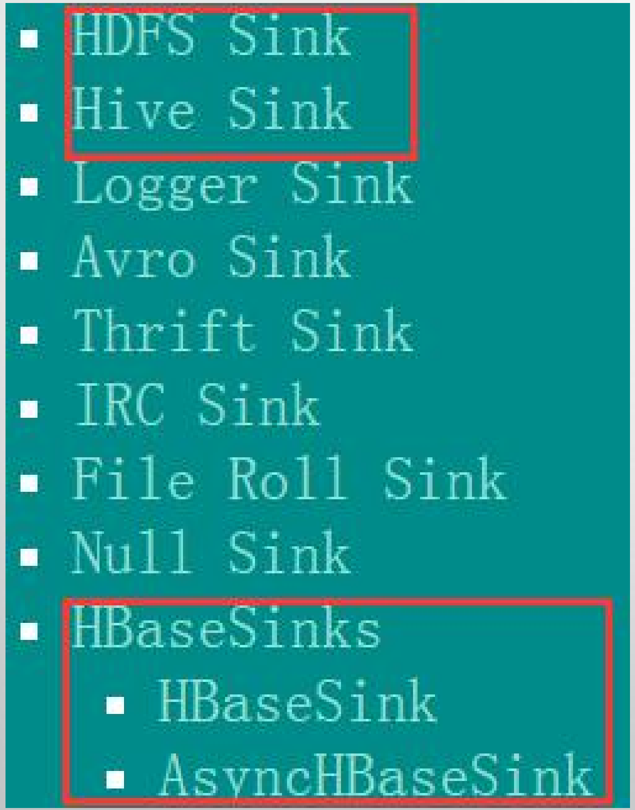
Property Name Default	Description	
type	–	The component type name, needs to be `file`.
checkpointDir	~/.flume/file-channel/checkpoint	The directory where checkpoint file will be stored
useDualCheckpoints	false	Backup the checkpoint. If this is set to `true`, `backupCheckpointDir` **must** be set
backupCheckpointDir	–	The directory where the checkpoint is backed up to. This directory **must not** be the same as the data directories or the checkpoint directory
dataDirs	~/.flume/file-channel/data	Comma separated list of directories for storing log files. Using multiple directories on separate disks can improve file channel peformance
transactionCapacity	10000	The maximum size of transaction supported by the channel
checkpointInterval	30000	Amount of time (in millis) between checkpoints
maxFileSize	2146435071	Max size (in bytes) of a single log file
minimumRequiredSpace	524288000	Minimum Required free space (in bytes). To avoid data corruption, File Channel stops accepting take/put requests when free space drops below this value
capacity	1000000	Maximum capacity of the channel



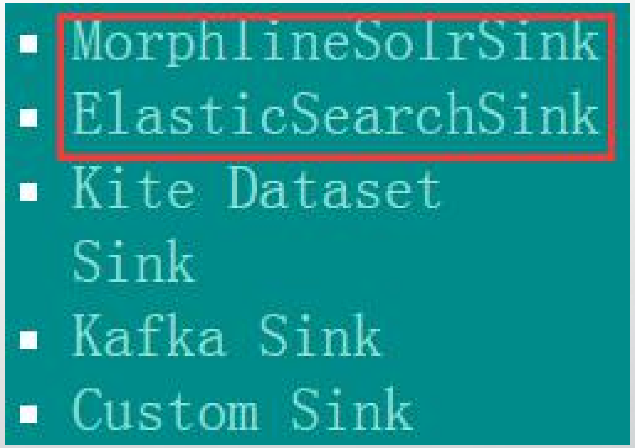
Example for agent named a1:

```
a1.channels = c1
a1.channels.c1.type = file
a1.channels.c1.checkpointDir = /mnt/flume/checkpoint
a1.channels.c1.dataDirs = /mnt/flume/data
```

Flume之Sinks

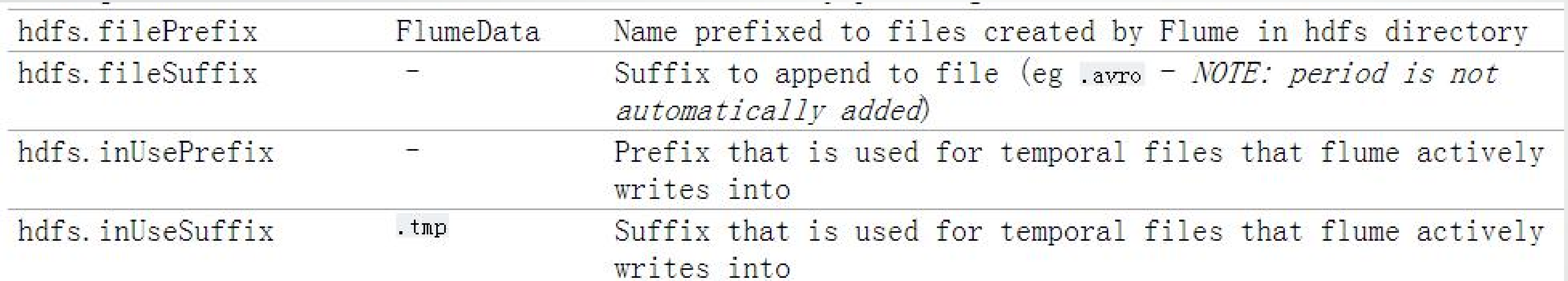


- HDFS Sink
- Hive Sink
- Logger Sink
- Avro Sink
- Thrift Sink
- IRC Sink
- File Roll Sink
- Null Sink
- HBaseSinks
  - HBaseSink
  - AsyncHBaseSink

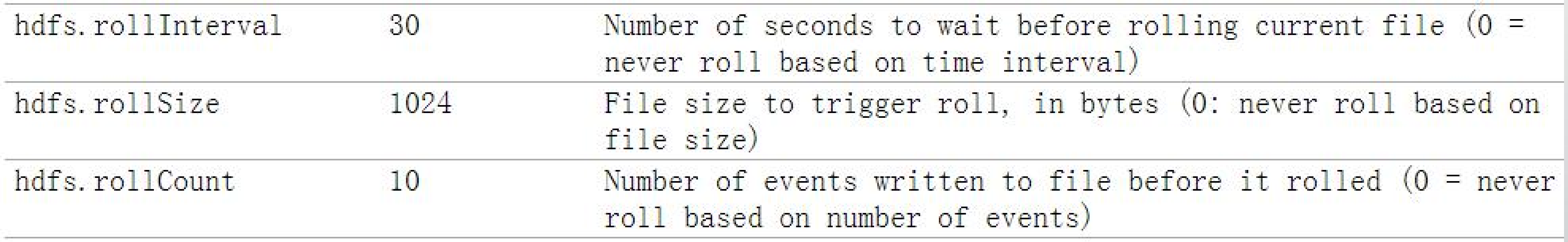


- MorphlineSolrSink
- ElasticSearchSink
- Kite Dataset Sink
- Kafka Sink
- Custom Sink

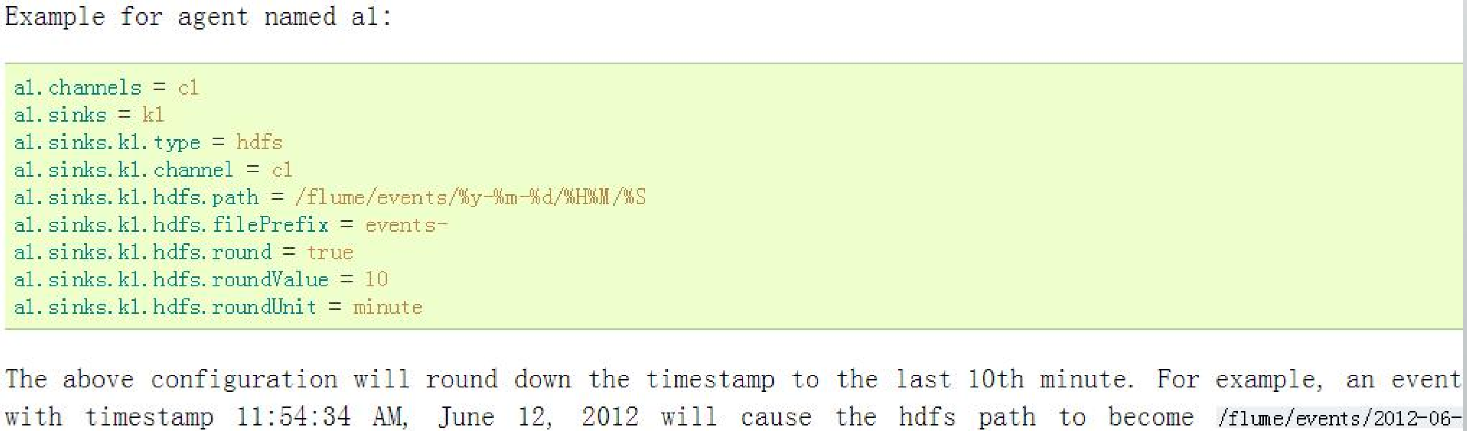
HDFS Sink



hdfs.filePrefix	FlumeData	Name prefixed to files created by Flume in hdfs directory
hdfs.fileSuffix	–	Suffix to append to file (eg `.avro` – *NOTE: period is not automatically added*)
hdfs.inUsePrefix	–	Prefix that is used for temporal files that flume actively writes into
hdfs.inUseSuffix	`.tmp`	Suffix that is used for temporal files that flume actively writes into



hdfs.rollInterval	30	Number of seconds to wait before rolling current file (0 = never roll based on time interval)
hdfs.rollSize	1024	File size to trigger roll, in bytes (0: never roll based on file size)
hdfs.rollCount	10	Number of events written to file before it rolled (0 = never roll based on number of events)

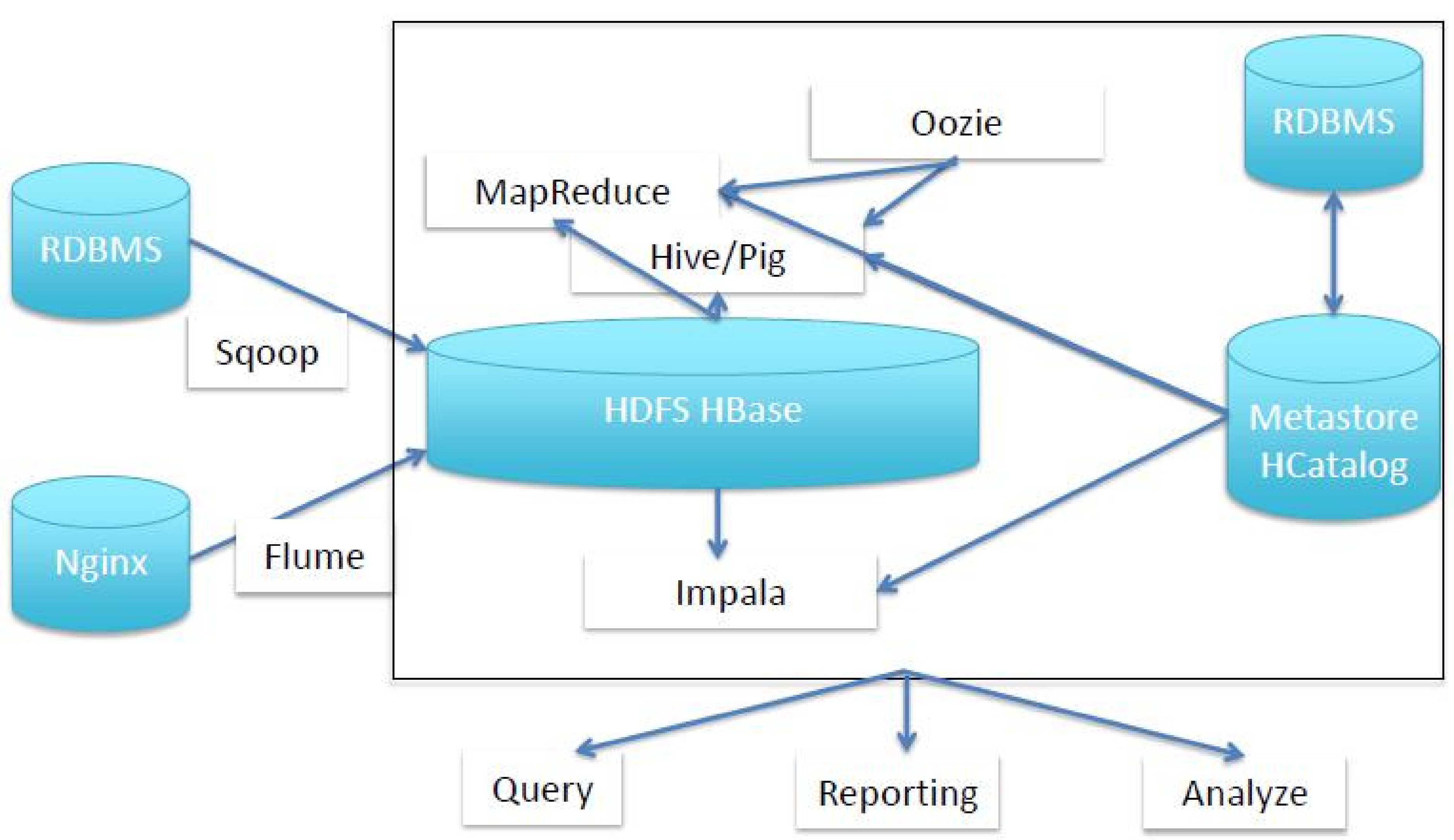


Example for agent named a1:

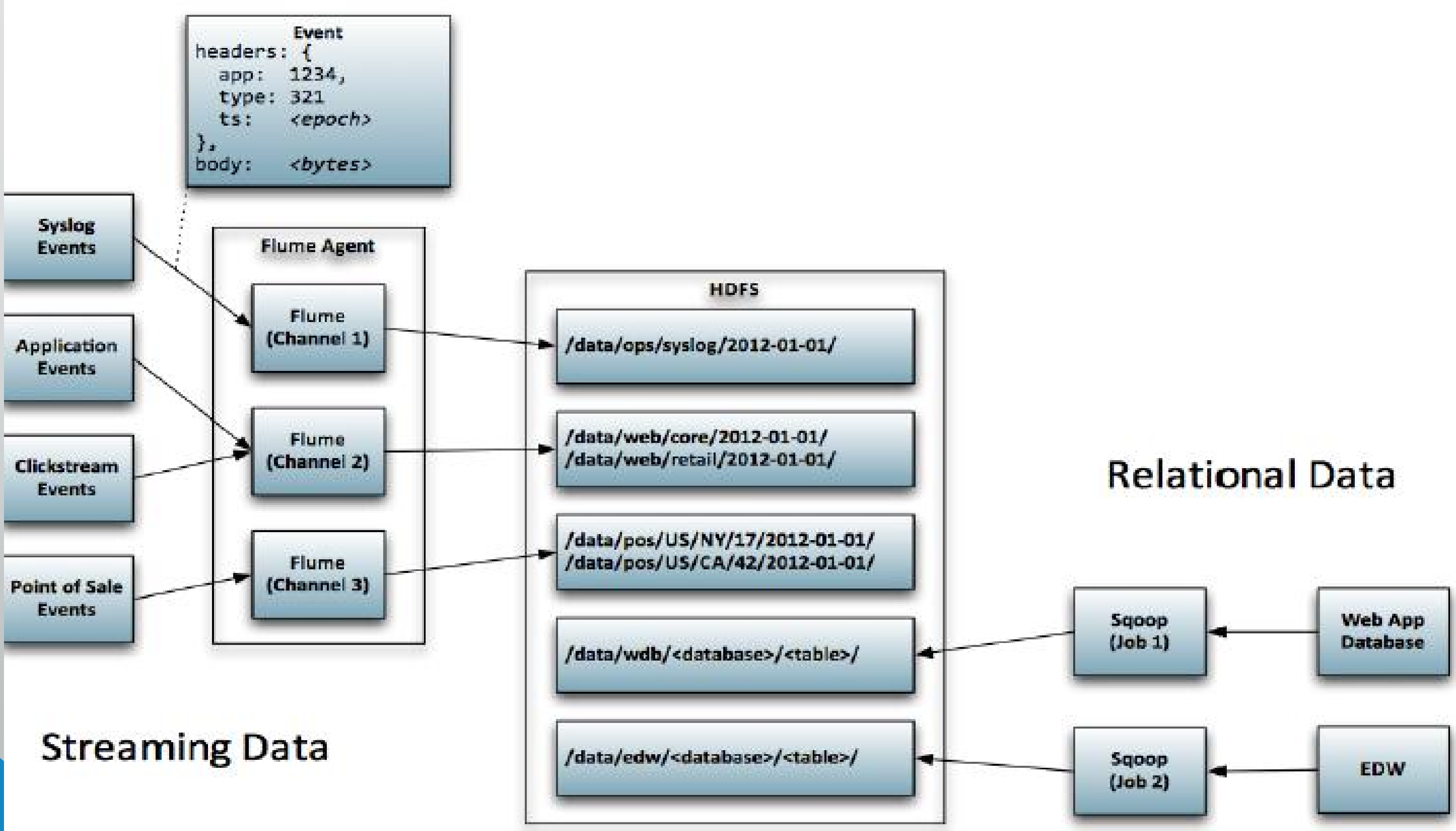
```
a1.channels = c1
a1.sinks = k1
a1.sinks.k1.type = hdfs
a1.sinks.k1.channel = c1
a1.sinks.k1.hdfs.path = /flume/events/%y-%m-%d/%H%M/%S
a1.sinks.k1.hdfs.filePrefix = events-
a1.sinks.k1.hdfs.round = true
a1.sinks.k1.hdfs.roundValue = 10
a1.sinks.k1.hdfs.roundUnit = minute
```

The above configuration will round down the timestamp to the last 10th minute. For example, an event with timestamp 11:54:34 AM, June 12, 2012 will cause the hdfs path to become `/flume/events/2012-06-`

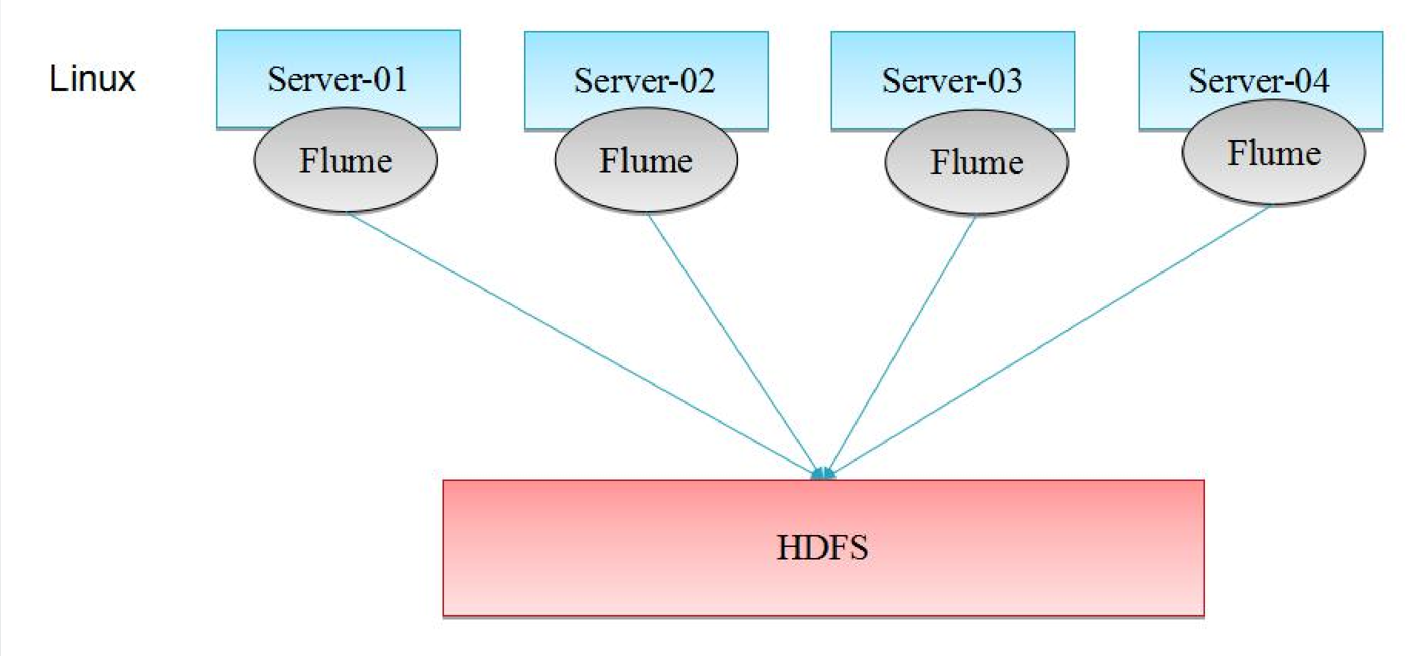
数据仓库架构


Oozie
RDBMS
MapReduce
RDBMS
Hive/Pig
Sqoop
HDFS HBase
Metastore
HCatalog
Nginx
Flume
Impala
Query
Reporting
Analyze

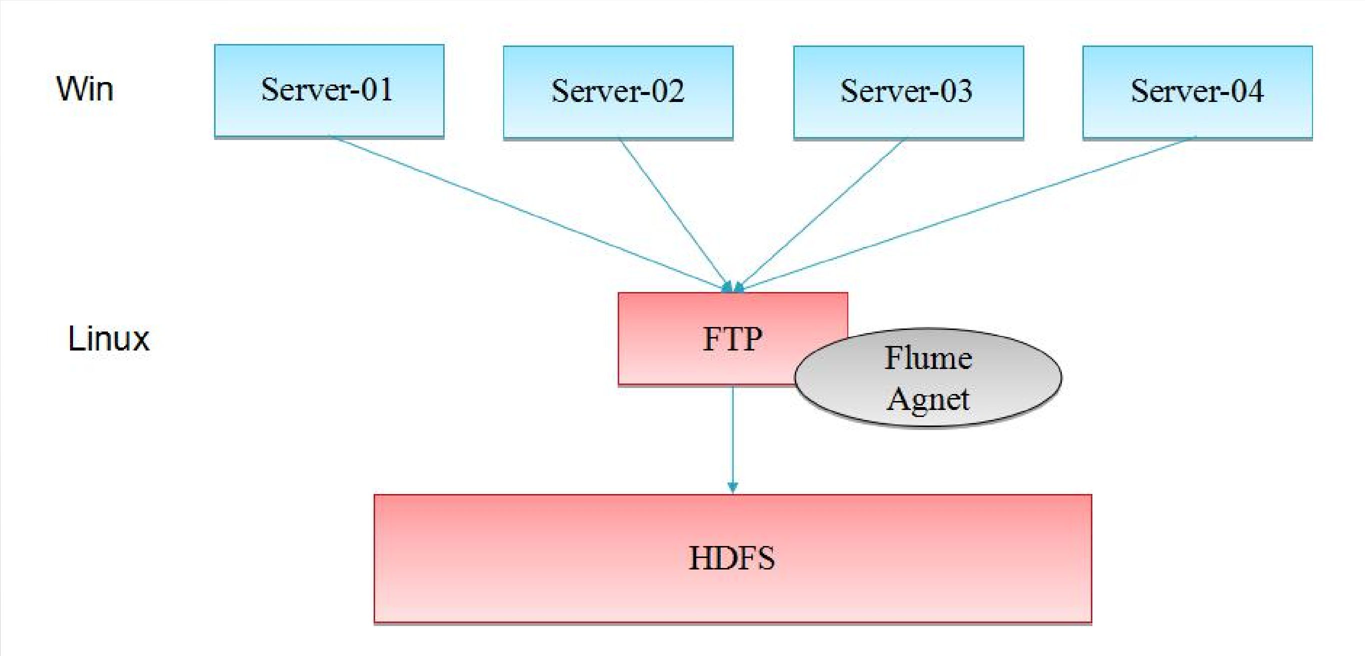
数据仓库架构


Event
headers: {
app: 1234,
type: 321
ts: <epoch>
},
body: <bytes>
Syslog Events
Application Events
Clickstream Events
Point of Sale Events
Flume Agent
Flume (Channel 1)
Flume (Channel 2)
Flume (Channel 3)
HDFS
/data/ops/syslog/2012-01-01/
/data/web/core/2012-01-01/
/data/web/retail/2012-01-01/
/data/pos/US/NY/17/2012-01-01/
/data/pos/US/CA/42/2012-01-01/
/data/wdb/<database>/<table>/
/data/edw/<database>/<table>/
Relational Data
Sqoop (Job 1)
Sqoop (Job 2)
Web App Database
EDW
Streaming Data

企业大数据仓库之数据收集架构



企业大数据仓库之数据收集架构



监控目录实时抽取数据

- **需求说明**
  - 监控某个目录，若目录下产生了符合条件的文件，flume就抽取它到hdfs上，目录下可能有多种文件，比如当文件以log.tmp结尾时表示正在写。对log.tmp文件设置一个size值，一旦到达size，则会变成一个完整文件以.log结尾，则已经是完整文件（往往存在短暂），flume可以抽取其中数据， 以.log.completed结尾则表示flume已经抽取完数据，可以删除掉。
- **业务分析**
  - 从上述需求可知，我们是要监控某个日志目录，所以Flume Agent的Source选择【Sqooling Directory source】，这个source会监控spooling directory下的新文件，并且当新文件出现解析event，上传数据到目标地。当这个文件在channel中被完全读取后，便会被重命名表示完成。
  - 本案例中Flume Agent不再使用前面所说的MemoryChannel，而是使用FileChannel，将Source获取的数据缓存到本地文件系统，要比MemoryChannel更加安全。

总结

- 实时收集文件框架有很多，但是其中Flume使用最广泛，主要由于其架构设计和使用简单清晰明了，又支持Hadoop存储。

- Flume是针对日志文件数据进行实时收集的框架，一个程序其实就是一个Flume Agent，包含三个部分Source、Channel及Sink。

- 在企业中针对实时分析统计要求高，通常使用Flume进行实时收集文件数据，再给Kafka类似的消息队列框架进行可靠性存储，最后又实时分布式框架进行计算；当然也是用Flume将数据收集以后放入HDFS中或检索框架Solr等中。