Flume采集处理日志文件

1. Flume简介

Flume是Cloudera提供的一个高可用的，高可靠的，分布式的海量日志采集、聚合和传输的系统，Flume支持在日志系统中定制各类数据发送方，用于收集数据；同时，Flume提供对数据进行简单处理，并写到各种数据接受方（可定制）的能力。

1. 系统功能

1. 日志收集

Flume最早是Cloudera提供的日志收集系统，目前是Apache下的一个孵化项目，Flume支持在日志系统中定制各类数据发送方，用于收集数据。

1. 数据处理

Flume提供对数据进行简单处理，并写到各种数据接受方（可定制）的能力 Flume提供了从console（控制台）、RPC（Thrift-RPC）、text（文件）、tail（UNIX tail）、syslog（syslog日志系统，支持TCP和UDP等2种模式），exec（命令执行）等数据源上收集数据的能力。

1. 工作方式

Flume采用了多Master的方式。为了保证配置数据的一致性，Flume[1] 引入了ZooKeeper，用于保存配置数据，ZooKeeper本身可保证配置数据的一致性和高可用，另外，在配置数据发生变化时，ZooKeeper可以通知Flume Master节点。Flume Master间使用gossip协议同步数据。

1. 流程结构

Flume的结构主要分为三部分：source、channel以及sink.其中source为源头，负责采集日志；channel为通道，负责传输和暂时储存；sink为目的地，将采集到的日志保存起来。在真正日志采集的过程中，根据待采集日志的类型以及存储需求，选择相应的类型的source、channel和sink进行配置，从而将日志采集并且保存起来。

1. Flume采集日志方案

1. 需求分析

1. 日志分类

操作系统：linux

日志更新类型：产生新日志，原日志结尾处追加

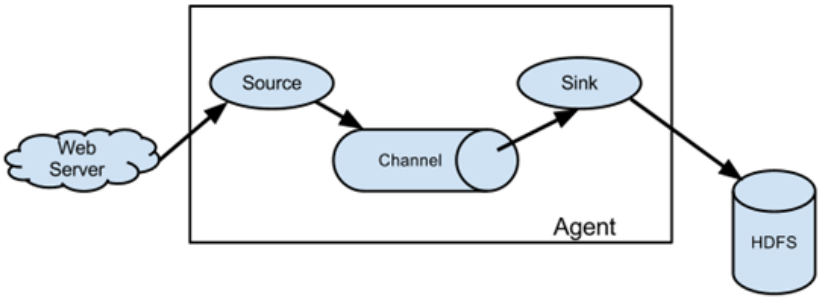
1. 采集时间需求

采集周期：短周期（一天之内）

1. 采集方案

1. 采集构架

使用flume采集日志文件的过程较简洁，只需选择恰当的source、channel和sink并且配置起来即可，若有特殊需求也可自己进行二次开发实现个人需求。



具体过程为：按照需求配置一个agent，选取适当的source和sink，然后启动该agent，开始采集日志。

1. **source**

flume提供多种source供用户进行选择，尽可能多的满足大部分日志采集的需求，常用的source的类型包括avro、exec、netcat、spooling-directory和syslog等。具体的使用范围和配置方法详见 source .

1. **channel**

flume中的channel不如source和sink那么重要，但却是不可忽视的组成部分。常用的channel为memory-channel，同时也有其他类型的channel，如JDBC、file-channel、custom-channel等，详情见 channel .

1. **sink**

flume的sink也有很多种，常用的包括avro、logger、HDFS、hbase以及file-roll等，除此之外还有其他类型的sink，如thrift、IRC、custom等。具体的使用范围和使用方法详见 sink .

1. Flume处理日志

Flume不止可以采集日志，还可以对日志进行简单的处理，在source处可以通过interceptor对日志正文处的重要内容进行过滤提取，在channel处可以通过header进行分类，将不同类型的日志投入不同的通道中，在sink处可以通过正则序列化来将正文内容进行进一步的过滤和分类。

1. Flume Source Interceptors

Flume可以通过interceptor将重要信息提取出来并且加入到header中，常用的interceptor有时间戳、主机名和UUID等，用户也可以根据个人需求编写正则过滤器，将某些特定格式的日志内容过滤出来，以满足特殊需求。

1. Flume Channel Selectors

Flume可以根据需求将不同的日志传输进不同的channel，具体方式有两种：复制和多路传输。复制就是不对日志进行分组，而是将所有日志都传输到每个通道中，对所有通道不做区别对待；多路传输就是根据指定的header将日志进行分类，根据分类规则将不同的日志投入到不同的channel中，从而将日志进行人为的初步分类。

1. Flume Sink Processors

Flume在sink处也可以对日志进行处理，常见的sink处理器包括custom、failover、load balancing和default等，和interceptor一样，用户也可以根据特殊需求使用正则过滤处理器，将日志内容过滤出来，但和interceptor不同的是在sink处使用正则序列化过滤出的内容不会加入到header中，从而不会使日志的header显得过于臃肿。

1. 附录

1. 常见的source

1. avro source

avro可以监听和收集指定端口的日志，使用avro的source需要说明被监听的主机ip和端口号，下面给出一个具体的例子：

```
a1.sources = r1
```

5. 【PWN】pwnable.kr ech...

推荐排行榜

1. 【Security】常用的渗透测...

```
a1.channels = c1

a1.sources.r1.type = avro

a1.sources.r1.channels = c1

a1.sources.r1.bind = 0.0.0.0

a1.sources.r1.port = 4141
```

1. **exec source**

exec可以通过指定的操作对日志进行读取，使用exec时需要指定shell命令，对日志进行读取，下面给出一个具体的例子：

```
a1.sources = r1

a1.channels = c1

a1.sources.r1.type = exec

a1.sources.r1.command = tail -F /var/log/secure

a1.sources.r1.channels = c1
```

1. **spooling-directory source**

spo_dir可以读取文件夹里的日志，使用时指定一个文件夹，可以读取该文件夹中的所有文件，需要注意的是该文件夹中的文件在读取过程中不能修改，同时文件名也不能修改。下面给出一个具体的例子：

```
agent-1.channels = ch-1

agent-1.sources = src-1


agent-1.sources.src-1.type = spooldir

agent-1.sources.src-1.channels = ch-1

agent-1.sources.src-1.spoolDir = /var/log/apache/flumeSpool

agent-1.sources.src-1.fileHeader = true
```

1. **syslog source**

syslog可以通过syslog协议读取系统日志，分为tcp和udp两种，使用时需指定ip和端口，下面给出一个udp的例子：

```
a1.sources = r1

a1.channels = c1

a1.sources.r1.type = syslogudp

a1.sources.r1.port = 5140

a1.sources.r1.host = localhost

a1.sources.r1.channels = c1
```

1. **常见的channel**

Flume的channel种类并不多，最常用的是memory channel，下面给出例子：

```
a1.channels = c1

a1.channels.c1.type = memory

a1.channels.c1.capacity = 10000

a1.channels.c1.transactionCapacity = 10000

a1.channels.c1.byteCapacityBufferPercentage = 20

a1.channels.c1.byteCapacity = 800000
```

1. # 常见的sink

    1. ## logger sink

logger顾名思义，就是将收集到的日志写到flume的log中，是个十分简单但非常实用的sink

1. ## avro sink

avro可以将接受到的日志发送到指定端口，供级联agent的下一跳收集和接受日志，使用时需要指定目的ip和端口：例子如下：

```
a1.channels = c1

a1.sinks = k1

a1.sinks.k1.type = avro

a1.sinks.k1.channel = c1

a1.sinks.k1.hostname = 10.10.10.10

a1.sinks.k1.port = 4545
```

1. ## file roll sink

file_roll可以将一定时间内收集到的日志写到一个指定的文件中，具体过程为用户指定一个文件夹和一个周期，然后启动agent，这时该文件夹会产生一个文件将该周期内收集到的日志全部写进该文件内，直到下一个周期再次产生一个新文件继续写入，以此类推，周而复始。下面给出一个具体的例子：

```
a1.channels = c1

a1.sinks = k1

a1.sinks.k1.type = file_roll

a1.sinks.k1.channel = c1

a1.sinks.k1.sink.directory = /var/log/flume
```

1. ## hdfs sink

hdfs与file roll有些类似，都是将收集到的日志写入到新创建的文件中保存起来，但区别是file roll的文件存储路径为系统的本地路径，而hdfs的存储路径为分布式的文件系统hdfs的路径，同时hdfs创建新文件的周期可以是时间，也可以是文件的大小，还可以是采集日志的条数。具体实例如下：

```
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
```

1. ## hbase sink

hbase是一种数据库，可以储存日志，使用时需要指定存储日志的表名和列族名，然后agent就可以将收集到的日志逐条插入到数据库中。例子如下：

```
a1.channels = c1

a1.sinks = k1

a1.sinks.k1.type = hbase

a1.sinks.k1.table = foo_table

a1.sinks.k1.columnFamily = bar_cf
```

```
a1.sinks.k1.serializer =
org.apache.flume.sink.hbase.RegexHbaseEventSerializer

a1.sinks.k1.channel = c1
```


deadbeef
关注 - 1
粉丝 - 3
+加关注

» 下一篇： 某互联网创业公司人才需求书

posted @ 2014-07-28 10:33 deadbeef 阅读(1130) 评论(0) 编辑 收藏