日志数据收集系统 Flume

一、实训环境

已经完成配置的 hadoop 完全分布式环境

二、实训内容

1. 解压 flume 安装包

```
[hadoop@master ~]$ sudo tar -zxvf /home/hadoop/apache-flume-1.9.0-bin.tar.gz -C /usr
```

2. 重命名安装路径

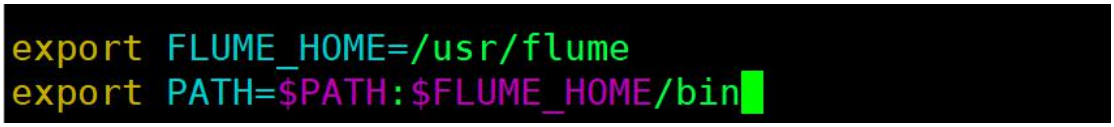
```
[hadoop@master ~]$ sudo mv /usr/apache-flume-1.9.0-bin/ /usr/flume
```

3. 配置 flume 环境变量

```
[hadoop@master ~]$ sudo vim /etc/profile
```

加入以下环境变量

```
export FLUME_HOME=/usr/flume
export PATH=$PATH:$FLUME_HOME/bin
```



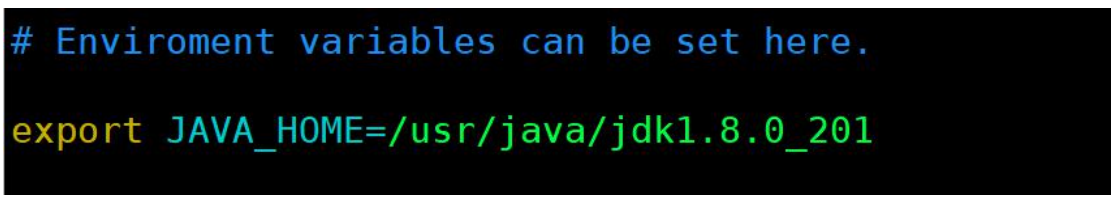
4. 使环境变量生效

```
[hadoop@master ~]$ source /etc/profile
```

5. 配置 flume-env.sh 文件

```
[hadoop@master ~]$ cp $FLUME_HOME/conf/flume-env.sh.template $FLUME_HOME/conf/flume-env.sh
[hadoop@master ~]$ sudo vim $FLUME_HOME/conf/flume-env.sh
```

更改配置文件里面的 JAVA_HOME



6. 更改文件属主权限

```
[hadoop@master ~]$ sudo chown -R hadoop:hadoop /usr/flume/
```

三、验证测试

```
[hadoop@master ~]$ flume-ng version
```

四、应用案例

1. 创建一个实例

1）创建一个实例配置文件

```
[hadoop@master ~]$ touch flumec.conf
```

2）编辑实例文件

```
[hadoop@master ~]$ vim flumec.conf
```

实例文件内容：

```
#Agent 名称，source、channel、sink 的名称

f1.sources=r1

f1.channels=c1

f1.sinks=k1


# 具体定义 source
```

```
f1.sources.r1.type=netcat
f1.sources.r1.bind=192.168.3.101 #收集日志数据主机的 IP
f1.sources.r1.port=55555
f1.sources.r1.max-line-length=1000000
f1.sources.r1.channels=c1

# channel 具体配置
f1.channels.c1.type=memory
f1.channels.c1.capacity=1000
f1.channels.c1.transactionCapacity=1000
f1.channels.c1.keep-alive=30

#具体定义 sink
f1.sinks.k1.type = hdfs
f1.sinks.k1.channel=c1
f1.sinks.k1.hdfs.path =hdfs:/
#前缀
f1.sinks.k1.hdfs.filePrefix=%Y%m%d-
#类型
f1.sinks.k1.hdfs.fileType=DataStream
f1.sinks.k1.hdfs.useLocalTimeStamp = true
#不按照条数生成文件
f1.sinks.k1.hdfs.rollCount=0
#HDFS 上的文件达到 128M 时生成一个文件
f1.sinks.k1.hdfs.rollSize=1048576
#HDFS 上的文件达到 60 分钟生成一个文件
f1.sinks.k1.hdfs.rollInterval=3600
```

2. 启动实例：

```
[hadoop@master ~]$ flume-ng agent -n f1 -c conf -p 55555 -f flumec.conf
-Dflume.hadoop.logger=INFO,console
```

```
[hadoop@master ~]$ flume-ng agent -n f1 -c conf -p 55555 -f flumec.conf -Df
lume.hadoop.logger=INFO,console
Info: Including Hadoop libraries found via (/usr/hadoop/bin/hadoop) for HDF
S access
Info: Including Hive libraries found via () for Hive access
+ exec /usr/java/jdk1.8.0_201/bin/java -Xmx20m -Dflume.hadoop.logger=INFO,c
onsole -cp 'conf:/usr/flume/lib/*:/usr/hadoop/etc/hadoop:.:/usr/hadoop/shar
e/hadoop/common/lib/*:/usr/hadoop/share/hadoop/common/*:/usr/hadoop/share/h
adoop/hdfs:/usr/hadoop/share/hadoop/hdfs/lib/*:/usr/hadoop/share/hadoop/hdf
s/*:/usr/hadoop/share/hadoop/mapreduce/lib/*:/usr/hadoop/share/hadoop/mapre
duce/*:/usr/hadoop/share/hadoop/yarn:/usr/hadoop/share/hadoop/yarn/lib/*:/u
sr/hadoop/share/hadoop/yarn/*:/usr/java/jdk1.8.0_201/lib:/usr/java/jdk1.8.0
_201/jre/lib:/lib/*' -Djava.library.path=:/usr/hadoop/lib/native org.apache
.flume.node.Application -n f1 -p 55555 -f flumec.conf
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/flume/lib/slf4j-log4j12-1.7.25.jar!/
org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hadoop/share/hadoop/common/lib/slf4j
-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explana
```

3. 测试访问实例

1）新建测试实例文件

```
[hadoop@master ~]$ touch main.py
```

2）编辑实例测试文件

```
[hadoop@master ~]$ vim main.py
```

实例测试文件内容：

```
#!/bin/python


import os

import time

import socket

import datetime


class Flume_test(object):

    def  __init__(self):

        self.flume_host ='192.168.224.134'

        self.flume_port = 55555


    def gen_conn(self):

        tcp_cli = socket.socket(socket.AF_INET,socket.SOCK_STREAM)

        return tcp_cli
```

```
    def gen_data(self):
        return 'Flume test ,datetime:[%s]\n'%datetime.datetime.now()


    def main(self):
        cli = self.gen_conn()
        cli.connect((self.flume_host,self.flume_port))
        while 1:
               data = self.gen_data()
               print(data)
               cli.sendall(bytes(data))
               recv = cli.recv(1024)
               print(recv)
               time.sleep(1)
        s.close()


if __name__ == '__main__':
    ft = Flume_test()
    ft.main()
```

3）修改实例测试文件权限

```
[hadoop@master ~]$ chmod +x main.py
```

4）启动实例测试文件

```
[hadoop@master ~]$ ./main.py
```

```
[hadoop@master ~]$ ./main.py
Flume test ,datetime:[2019-08-02 02:58:46.481642]

OK

Flume test ,datetime:[2019-08-02 02:58:47.497789]

OK

Flume test ,datetime:[2019-08-02 02:58:48.500419]
```