高可用 Flum-NG 配置案例 failover

角色分配

Flume 的 Agent 和 Collector 分布如下表所示:

名称	HOST	角色
Agent1	node01	Web Server
Collector1	node02	AgentMstr1
Collector2	node03	AgentMstr2

图中所示，Agent1 数据分别流入到 Collector1 和 Collector2，Flume NG 本身提供了 Failover 机制，可以自动切换和恢复。在上图中，有 3 个产生日志服务器分布在不同的机房，要把所有的日志都收集到一个集群中存储。下 面我们开发配置 Flume NG 集群

node01 安装配置 flume 与拷贝文件脚本

将 node03 机器上面的 flume 安装包拷贝到 node01 机器上面去

node03 机器执行以下命令

```
cd /export/servers
scp -r apache-flume-1.6.0-cdh5.14.0-bin/ node01:$PWD
```

node01 机器配置 agent 的配置文件

cd /export/servers/apache-flume-1.6.0-cdh5.14.0-bin/conf

vim agent.conf

```
#agent1 name
agent1.channels = c1
agent1.sources = r1
agent1.sinks = k1 k2
#
##set gruop
agent1.sinkgroups = g1
#
##set channel
agent1.channels.c1.type = memory
agent1.channels.c1.capacity = 1000
```

```
agent1.channels.c1.transactionCapacity = 100
#
agent1.sources.r1.channels = c1
agent1.sources.r1.type = exec
agent1.sources.r1.command = tail -F
/export/servers/taillogs/access_log
#
agent1.sources.r1.interceptors = i1 i2
agent1.sources.r1.interceptors.i1.type = static
agent1.sources.r1.interceptors.i1.key = Type
agent1.sources.r1.interceptors.i1.value = LOGIN
agent1.sources.r1.interceptors.i2.type = timestamp
#
## set sink1
agent1.sinks.k1.channel = c1
agent1.sinks.k1.type = avro
agent1.sinks.k1.hostname = node02
agent1.sinks.k1.port = 52020
#
## set sink2
agent1.sinks.k2.channel = c1
agent1.sinks.k2.type = avro
agent1.sinks.k2.hostname = node03
agent1.sinks.k2.port = 52020
#
##set sink group
agent1.sinkgroups.g1.sinks = k1 k2
#
##set failover
agent1.sinkgroups.g1.processor.type = failover
agent1.sinkgroups.g1.processor.priority.k1 = 10
agent1.sinkgroups.g1.processor.priority.k2 = 1
agent1.sinkgroups.g1.processor.maxpenalty = 10000
#
```

node02 与 node03 配置 flumecollection

node02 机器修改配置文件

```
cd /export/servers/apache-flume-1.6.0-cdh5.14.0-bin/conf
vim collector.conf
```

```
#set Agent name
a1.sources = r1
a1.channels = c1
a1.sinks = k1
#
##set channel
a1.channels.c1.type = memory
a1.channels.c1.capacity = 1000
a1.channels.c1.transactionCapacity = 100
#
## other node,nna to nns
a1.sources.r1.type = avro
a1.sources.r1.bind = node02
a1.sources.r1.port = 52020
a1.sources.r1.interceptors = i1
a1.sources.r1.interceptors.i1.type = static
a1.sources.r1.interceptors.i1.key = Collector
a1.sources.r1.interceptors.i1.value = node02
a1.sources.r1.channels = c1
#
##set sink to hdfs
a1.sinks.k1.type=hdfs
a1.sinks.k1.hdfs.path= hdfs://node01:8020/flume01/failover/
a1.sinks.k1.hdfs.fileType=DataStream
a1.sinks.k1.hdfs.writeFormat=TEXT
a1.sinks.k1.hdfs.rollInterval=10
a1.sinks.k1.channel=c1
a1.sinks.k1.hdfs.filePrefix=%Y-%m-%d
#
```

node03 机器修改配置文件

```
cd  /export/servers/apache-flume-1.6.0-cdh5.14.0-bin/conf
vim collector.conf
```

```
#set Agent name
a1.sources = r1
a1.channels = c1
```

```
a1.sinks = k1
#
##set channel
a1.channels.c1.type = memory
a1.channels.c1.capacity = 1000
a1.channels.c1.transactionCapacity = 100
#
## other node,nna to nns
a1.sources.r1.type = avro
a1.sources.r1.bind = node03
a1.sources.r1.port = 52020
a1.sources.r1.interceptors = i1
a1.sources.r1.interceptors.i1.type = static
a1.sources.r1.interceptors.i1.key = Collector
a1.sources.r1.interceptors.i1.value = node03
a1.sources.r1.channels = c1
```

```
#
##set sink to hdfs
a1.sinks.k1.type=hdfs
a1.sinks.k1.hdfs.path= hdfs://node01:8020/flume01/failover/
a1.sinks.k1.hdfs.fileType=DataStream
a1.sinks.k1.hdfs.writeFormat=TEXT
a1.sinks.k1.hdfs.rollInterval=10
a1.sinks.k1.channel=c1
a1.sinks.k1.hdfs.filePrefix=%Y-%m-%d
```

顺序启动命令

node03 机器上面启动 flume

```
cd /export/servers/apache-flume-1.6.0-cdh5.14.0-bin
bin/flume-ng agent -n a1 -c conf -f conf/collector.conf -Dflume.root.logger=DEBUG,console
```

node02 机器上面启动 flume

```
cd /export/servers/apache-flume-1.6.0-cdh5.14.0-bin
bin/flume-ng agent -n a1 -c conf -f conf/collector.conf -Dflume.root.logger=DEBUG,console
```

node01 机器上面启动 flume

```
cd /export/servers/apache-flume-1.6.0-cdh5.14.0-bin
bin/flume-ng agent -n agent1 -c conf -f conf/agent.conf -Dflume.root.logger=DEBUG,console
```

node01 机器启动文件产生脚本

```
mkdir -p /export/servers/shells/
mkdir -p /export/servers/taillogs
```

```
cd  /export/servers/shells/
vim tail-file.sh
```

```
#!/bin/bash
while true
do
  date >> /export/servers/taillogs/access_log;
    sleep 0.5;
done
```

启动脚本

```
sh /export/servers/shells/tail-file.sh
```

tail -f access_log