

Kafka 集群安装步骤

Kafka0.10.1 安装步骤

<http://blog.csdn.net/czmacd/article/details/54410508>

详细 KafkaAPI 请参考官方文档

<http://kafka.apache.org/0101/javadoc/index.html>

1 虚拟机安装

1. 安装 VMware Workstation;
2. 安装三台虚拟机（这里系统使用 CentOS6.5）;
3. 进行网络配置，保证三台虚拟机可以互相通信，能上网更好。

三台虚拟机名称和 IP 分别为：

192.168.85.5 Server1

192.168.85.6 Server2

2 安装 JDK

1. 下载 JDK
这里使用版本为 1.8.0_101
2. 卸载系统原有 JDK;

```
[root@Server1 ~]#rpm -qa | grep java
java-1.7.0-openjdk-1.7.0.45-2.4.3.3.el6.x86_64
tzdata-java-2013g-1.el6.noarch
java-1.6.0-openjdk-1.6.0.0-1.66.1.13.0.el6.x86_64
[root@Server1 ~]#rpm -e --nodeps java-1.7.0-openjdk-1.7.0.45-2.4.3.3.el6.x86_64
[root@Server1 ~]#rpm -e --nodeps tzdata-java-2013g-1.el6.noarch
[root@Server1 ~]#rpm -e --nodeps java-1.6.0-openjdk-1.6.0.0-1.66.1.13.0.el6.x86_64
```

3. 手动安装 JDK

```
[root@Server1 ~]# cd /usr/local
将 jdk-8u101-linux-x64.tar.gz 放入该目录中，解压
[root@Server1 java]# tar -zxvf jdk-8u101-linux-x64.tar.gz
解压完成后，配置环境变量
[root@Server1 java]# vim /etc/profile
```

输入 i, 启用编辑

添加配置项如下：

```
export JAVA_HOME=/usr/local/jdk1.8.0_101
export JRE_HOME=/usr/local/jdk1.8.0_101/jre
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:$JRE_
HOME/lib:$CLASSPATH
export PATH=$JAVA_HOME/bin:$PATH
```

编辑完成后，输入 Esc，退出编辑

输入:wq，保存文件并退出

保存后记得使环境变量配置生效，并验证 JDK 安装是否成功。

```
[root@Server1 java]# source /etc/profile
[root@Server1 java]# java -version
java version "1.8.0_101"
Java(TM) SE Runtime Environment (build 1.8.0_101-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.101-b13, mixed mode)
```

至此，JDK 手动安装成功。

3 安装 Zookeeper 集群

1. 下载 zookeeper 安装包

到 <http://zookeeper.apache.org/releases.html> 下载 zookeeper 的安装包 zookeeper-3.4.9.tar.gz，将文件保存到 Server1 的/usr/local/目录下，目录可以自行设置。

```
[root@Server1 local]# tar -zxvf zookeeper-3.4.9.tar.gz
//该重命名步骤纯属为了以后方便，可以省略
[root@Server1 local]# mv zookeeper-3.4.9 zookeeper
```

2. 解压并重命名

3. 相关参数配置

- a) 进入解压好的 zookeeper 目录中，将 conf/zoo_sample.cfg 拷贝一份命名为 zoo.cfg。

```
[root@Server1 local]# cd zookeeper
[root@Server1 zookeeper]# cd conf
[root@Server1 conf]# cp zoo_sample.cfg zoo.cfg
[root@Server1 conf]# vim zoo.cfg
```

- b) 修改配置项如下：

```
# The number of milliseconds of each tick
tickTime=2000

# The number of ticks that the initial
# synchronization phase can take
initLimit=10

# The number of ticks that can pass between
# sending a request and getting an acknowledgement
syncLimit=5

# the directory where the snapshot is stored.
# do not use /tmp for storage, /tmp here is just
# example sakes.

dataDir=/home/zookeeper/data
dataLogDir=/home/zookeeper/logs

这里默认原为 dataDir=/tmp/zookeeper，这样重启后就会清空
```

```
# the port at which the clients will connect
clientPort=2181

# the maximum number of client connections.
# increase this if you need to handle more clients
#maxClientCnxns=60

#

# Be sure to read the maintenance section of the
# administrator guide before turning on autopurge.
#
#
http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc\_maintenance

#

# The number of snapshots to retain in dataDir
#autopurge.snapRetainCount=3

# Purge task interval in hours
# Set to "0" to disable auto purge feature
#autopurge.purgeInterval=1

server.1=192.168.85.5:2888:3888
server.2=192.168.85.6:2888:3888
```

- c) 在/home/zookeeper 目录下新建 dataDir 和 dataLogDir 文件夹，在 dataDir 和 dataLogDir 目录下新建 myid 文件

```
[root@Server1 conf]# cd /home/zookeeper/data
```

```
[root@Server1 data]# echo 1 >> myid
```

```
[root@Server1 data]# cd /home/zookeeper/logs
```

```
[root@Server1 logs]# echo 1 >> myid
```

注意：这个 id 是 zookeeper 的主机标识，每个主机 id 不同，第一台是 1，第二台是 2，以此类推。

d) 可以考虑将 zookeeper 相关命令配置在环境变量中

e) 重复以上步骤在其它虚拟机上安装 zookeeper

4. 启动 zookeeper

```
[root@Server1 logs]# cd /usr/local/zookeeper/bin
```

```
[root@Server1 bin]# ./zkServer.sh start
```

```
ZooKeeper JMX enabled by default
```

```
Using config: /usr/local/zookeeper/bin/../conf/zoo.cfg
```

```
Starting zookeeper ... STARTED
```

逐个启动每台虚拟机中的 zookeeper。

5. 查看 zookeeper 状态

```
[root@Server1 bin]# ./zkServer.sh status
```

在 zookeeper 集群启动后，查看状态可以看到如下情况：

```
[root@Server2 ~]# zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /usr/local/zookeeper/bin/../conf/zoo.cfg
Mode: leader
-
[root@Server1 bin]# ./zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /usr/local/zookeeper/bin/../conf/zoo.cfg
Mode: follower
- - - - -
```

至此，zookeeper 集群搭建成功。

6 关闭 zookeeper

```
[root@Server1 bin]# ./zkServer.sh stop
```

4 安装 Kafka 集群

1. 下载 Kafka 安装包

到 <http://kafka.apache.org/downloads.html> 下载 kafka 的安装包 kafka_2.11-0.10.1.0.tgz，将文件保存到 server1 的/usr/local/目录下，目录可以自行设置。

2. 解压并重命名

```
[root@Server1 local]# tar -zxvf kafka_2.11-0.10.1.0.tgz
//该重命名步骤纯属为了以后方便，可以省略
[root@Server1 local]# mv kafka_2.11-0.10.1.0 kafka
```

3. 修改服务端配置文件

```
[root@Server1 local]# cd kafka
[root@Server1 kafka]# cd config
[root@Server1 config]# vim server.properties
```

```
broker.id=0（当前 broker 的编号）
listeners=PLAINTEXT://192.168.85.5:9092（当前 broker 的 IP）
zookeeper.connect=192.168.85.5:2181,192.168.85.6:2181,
delete.topic.enable=true
```

4. 修改生产者配置文件

```
[root@Server1 config]# vim producer.properties
```

```
bootstrap.servers=192.168.85.5:9092,192.168.85.6:9092
```

5. 修改消费者配置文件

```
[root@Server1 config]# vim consumer.properties
```

```
zookeeper.connect=192.168.85.5:2181,192.168.85.6:2181
```

6. 然后重复以上步骤，在其它虚拟机上安装，区别在于 `broker.id=0` 要变为 `1,2……`，`listeners` 所配置的 IP 也要进行相应的更改。
7. 可以考虑将 `kafka` 相关命令配置到环境变量中。
8. 进入 `Kafka` 安装目录的 `bin` 目录下，启动 `Kafka`

```
[root@Server1 kafka]#kafka-server-start.sh -daemon config/server.properties
```

由于 `kafka` 的启动依赖于 `zookeeper`，所以启动前要保证 `zookeeper` 启动成功。

9. `Kafka` 集群测试（一下皆为 `bin` 目录下的操作）

```
[root@Server1 ~]# kafka-topics.sh --create --zookeeper localhost:2181
--replication-factor 1 --partitions 3 --topic test-topic
Created topic "test-topic".
```

- a) 创建一个 topic
- b) 查看创建的 topic

```
[root@Server1 ~]# kafka-topics.sh -describe --zookeeper localhost:2181
--topic test-topic
```

- c) `Server1` 创建一个生产者发送消息

```
[root@Server1 ~]#kafka-console-producer.sh --broker-list localhost:9092
--topic test-topic
[2016-10-11 20:54:07,025] aaaaaaaaaaaaaaaaaa
```


d) Server2 创建一个消费者接收消息

```
[root@Server2~]# kafka-console-consumer.sh --bootstrap-server
localhost:9092 --topic test-topic
aaaaaaaaaaaaaaaaaa
```

至此，Kafka 集群搭建成功。

10. 关闭 Kafka

每一台机器上执行如下命令

```
[root@Server2 ~]# kafka-server-stop.sh
```

关闭后使用 `jps` 命令查看有无 Kafka 进程存在

若自带脚本无法关闭 kafka,

修改 `kafka-server-stop.sh` 中的指令如下

```
PIDS=$(jps -l | grep 'kafka\.Kafka' | awk '{print $1}')
if [ -z "$PIDS" ]; then
    echo "No kafka server to stop"
    exit 1
else
    kill -s TERM $PIDS
fi
```