

1) zookeeper安装

kafka对消息保存时根据Topic进行归类，发送消息者成为Producer,消息接受者成为Consumer,此外kafka集群有多个kafka实例组成，每个实例(server)成为broker。无论是kafka集群，还是producer和consumer都依赖于zookeeper来保证系统可用性集群保存一些meta信息。

- 将下载的zookeeper安装包用xftp上传到/data/program目录底下
- 注：软件压缩包统一上传至/data/program

1.解压前先验证包是否完整

```
sha256sum apache-zookeeper-3.6.1-bin.tar.gz
```

预计输出：

```
5066dd085cee2a7435a1bb25677102f0d4ea585f314bd026799f333b0956a06d apache-zookeeper-3.6.1-bin.tar.gz
```

2.进入/data/program目录解压apache-zookeeper-3.6.1-bin.tar.gz到当前目录

```
tar -xzf apache-zookeeper-3.6.1-bin.tar.gz
```

3.在/data/program/目录下创建软链接zookeeper到/data/program/apache-zookeeper-3.6.1-bin

```
ln -s /data/program/apache-zookeeper-3.6.1-bin zookeeper
```

4.将zookeeper加入到环境变量中

```
echo "export ZOOKEEPER_HOME=/data/program/zookeeper" > /etc/profile.d/zookeeper.sh
echo 'PATH=$PATH:$ZOOKEEPER_HOME/bin' >> /etc/profile.d/zookeeper.sh
. /etc/profile.d/zookeeper.sh #让环境变量生效
```

5.进入/data/logs创建zookeeper目录（logs是日志目录）

```
mkdir /data/logs/zookeeper
```

6.进入/data/datalake/创建zkdata目录（datalake是保存数据的目录）

```
mkdir /data/datalake/zkdata
```

7.进入/data/program/zookeeper/conf目录，复制zoo_sample.cfg改名为zoo.cfg

```
cp zoo_sample.cfg zoo.cfg
```

8.进入zoo.cfg修改dataDir=数据目录，并添加日志目录路径

```
vi zoo.cfg
dataDir=/data/datalake/zkdata
# 在末尾添加
dataLogDir=/data/logs/zookeeper/ #日志路径
```

9.进入/data/bin目录 编辑zookeeper启动脚本 (bin是软件快捷启动停止的目录)

```
touch /data/bin/zookeeper_start.sh
chmod +x /data/bin/zookeeper_start.sh
vi /data/bin/zookeeper_start.sh

# 输入如下内容:
$ZOOKEEPER_HOME/bin/zkServer.sh start
```

shell

10.进入/data/bin目录 编写关闭zookeeper脚本

```
touch /data/bin/zookeeper_stop.sh
chmod +x /data/bin/zookeeper_stop.sh
vi /data/bin/zookeeper_stop.sh

# 输入如下内容:
$ZOOKEEPER_HOME/bin/zkServer.sh stop
```

11.启动zookeeper (启动后使用jps -ml查看是否启动成功)

```
/data/bin/zookeeper_start.sh
```

2) kafka的安装与部署(需要配置在master)

- 将下载的kafka安装包用xftp上传到/data/program目录底下

1.解压前先验证包是否完整

```
sha256sum kafka_2.12-2.3.0.tgz
```

预计输出:

```
d86f5121a9f0c44477ae6b6f235daecc3f04ecb7bf98596fd91f402336eee3e7 kafka_2.12-2.3.0.tgz
```

2.进入/data/program目录解压hbase-2.2.4-bin.tar.gz到当前目录

```
tar -xzvf kafka_2.12-2.3.0.tgz
```

3.在/data/program/目录下创建软链接kafka到/data/program/kafka_2.12-2.3.0.tgz

```
ln -s /data/program/kafka_2.12-2.3.0 kafka
```

4.进入/data/logs创建kafka目录 (logs是日志目录)

```
mkdir /data/logs/kafka
```

5.启动kafka

```
# 先进入kafka目录
# 不要关闭当前终端,启动新的终端
bin/kafka-server-start.sh config/server.properties &
```

6.检验kafka的端口是否启动成功

```
netstat -tunlp|egrep "(2181|9092)"
```

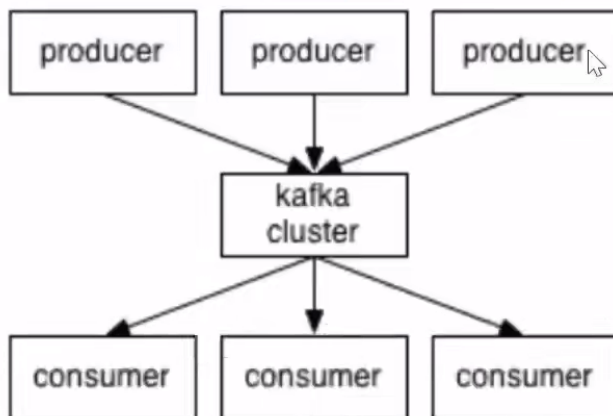
shell

3) kafka 使用案例讲解)

Log Aggregation(对数聚合)

kafka可以批量提交消息/压缩消息等,这对producer端而言,几乎感觉不到性能的开支.此时consumer端可以使hadoop等其他系统化的存储和分析系统.

- 首先启动多个XShell客户端, 一个用于生产者发送消息, 其余用于消费者接受消息
- producer, 指定的Socket(192.168.1.181+9092),说明生产者的消息要发往kafka, 也即是broker
- consumer, 指定的Socket(192.168.1.181+2181),说明消费者的消息来自zookeeper (协调转发) !



(摘自官网)

1.进入kafka目录

```
cd data/program/kafka
```

2.创建一个名为text的topic

```
./bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --  
partitions 1 --topic test
```

3.在一个终端打开producer

```
./bin/kafka-console-producer.sh --broker-list localhost:9092 --topic test #在一个终端打  
开producer
```

4.在另一个终端打开Consumer

```
./bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test --
```

shell