

ZooKeeper与Kafka环境搭建与集群

ZooKeeper安装

从官网下载ZooKeeper安装包到安装目录 并解压

```
$ tar -zxvf zookeeper-3.4.13.tar.gz
```

进入ZooKeeper的conf目录下, 将zoo_sample.cfg复制一份命名为zoo.cfg

```
$ cd zookeeper-3.4.13/conf  
$ cp zoo_sample.cfg zoo.cfg
```

zoo.cfg配置文件内容

```
# 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 sake.  
dataDir=/tmp/zookeeper/data  
dataLogDir=/tmp/zookeeper/log  
# 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
```

- tickTime: zookeeper服务器之间或客户端与服务器间维持心跳的时间。也就是每隔tickTime时间就会发送一个心跳，单位：毫秒
- dataDir: zookeeper保存数据的目录
- clientPort: 客户端连接zookeeper服务器的端口，默认是2181
- initLimit: zookeeper接收客户端初始化时收到的心跳个数，超过配置数量表明这个客户端连接失败。总的时间长为tickTime * initLimit的值
- syncLimit: Leader和Follower之间发送消息，请求和应答的长度，最大个数为tickTime的值，总的时间长为tickTime * syncLimit

进入ZooKeeper的bin目录下，执行以下命令启动ZooKeeper

```
$ cd zookeeper-3.4.13/bin
$ ./zkServer.sh start    #start:启动服务; stop:关闭服务; status: 查看服务状态
```

安装完成

Zookeeper客户端登录与基本命令

进入ZooKeeper的bin目录下，使用zkCli.sh命令连接zookeeper

```
$ cd zookeeper-3.4.13/bin
$ ./zkCli.sh -server ip:port
```

常用文件操作命令集

ls /	#查看根目录下的文件
create /目录名	#创建一个目录
create /目录名 内容	#创建一个目录并添加
delete /目录名、文件	#删除目录（只能删除空目录或文件）
rmr /目录	#级联删除目录（子文件夹一并删除）
get /目录	#查看目录的信息
get /目录/文件	#查看文件内容

Kafka安装

从官网下载Kafka安装包到安装目录 并解压

```
$ tar -zxvf kafka_2.12-2.1.1.tgz
```

进入Kafka的config目录下，将zoo_sample.cfg复制一份命名为zoo.cfg

进入到kafka的根目录，执行以下语句开启服务

```
$ cd kafka_2.12-2.1.1  
$ bin/kafka-server-start.sh config/server.properties
```

安装完成

Kafka 创建、查看、删除topic

创建topic

```
$ bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --  
partitions 1 --topic test  
  
--replication-factor 2    #复制1份  
--partitions 1            #创建1个分区  
--topic                   #主题为test
```

查看topic

```
$ bin/kafka-topics.sh --list --zookeeper localhost:2181
```

删除topic

```
$ bin/kafka-topics.sh --delete--zookeeper localhost:2181 --topic [主题名]
```

ZooKeeper集群

进入ZooKeeper的conf目录下，在zoo.cfg文件中添加server.myid=ip:port1:port2

```
server.1=ip:2888:3888
server.2=ip:2888:3888
server.3=ip:2888:3888
.....
```

- server.myid=ip:port1:port2, myid是服务器的编号，一个正整数，一般是0、1、2、3等，port1表示的是服务器与集群中的Leader服务器交换信息的端口，一般用2888，Port2表示的是万一集群中的Leader服务器宕机了，需要一个端口来重新进行宣讲，选出一个新的Leader，一般用3888

在zoo.cfg中dataDir配置的文件夹下创建myid文件，内容为服务器的编号，myid是服务器的编号，一个正整数，一般是0、1、2、3等，与zoo.cfg中的配置的server.myid一致

配置完成，启动所有服务，使用客户端连接一台zookeeper服务器，创建test目录，查看其它服务器是否同步。

Kafka集群

进入Kafka的config目录下，在server.properties文件中增加修改以下项

```
broker.id=0
#每台服务器的broker.id唯一

listeners=PLAINTEXT://192.168.9.86:9092
#使用本机的ip地址

zookeeper.connect=192.168.9.86:2181,192.168.9.86:2182,192.168.9.86:2183/kafka
# 设置zookeeper的连接信息，kafka的目录不在zookeeper目录时，只要在最后一个地址上添加目录名
```

所有Kafka服务开启，在其中一台创建3个副本的topic, 创建完，查看其它服务器是否已创建

```
$ bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 3 --
partitions 1 --topic test
```

Kafka集群完成