

kafka 部署和测试

Name : 曲中岭

Email: zlingqu@126.com

Q Q :441869115

一、单机部署

1.1 规划

OS: CentOS 7.5 x64

JDK: 1.8.0_171

Kafka: 2.12-2.1.0

IP: 172.16.6.14

安装目录: /root/kafka_2.12-2.1.0

监听端口: 9092

Zookeeper: localhost:2181

1.2 配置 JDK

下载地址: <http://www.oracle.com/technetwork/cn/java/archive-139210-zhs.html>

找到自己需要的版本, 上传到服务器

```
tar xf jdk-8u171-linux-x64.tar -C /usr/local
mv /usr/local/jdk1.8.0_171 /usr/local/jdk
```

#也可以创建软连接, 看个人习惯

保证/etc/profile 中有以下内容:

```
export JAVA_HOME=/usr/local/jdk
export PATH=$JAVA_HOME/bin:$PATH
```

#\$JAVA_HOME/bin 一定要放到\$PATH 前面, 防止识别到系统环境中其他版本的 jdk

```
source /etc/profile
```

#使配置生效

```
java -version
```

查看是否配置成功

1.3 下载和解压

下载地址: <http://kafka.apache.org/downloads>

选择合适的版本, 比如 2.12, 会跳到

https://www.apache.org/dyn/closer.cgi?path=/kafka/2.1.0/kafka_2.12-2.1.0.tgz

这里可以选择国内的源

HTTP

http://mirror.bit.edu.cn/apache/kafka/2.1.0/kafka_2.12-2.1.0.tgz

http://mirrors.hust.edu.cn/apache/kafka/2.1.0/kafka_2.12-2.1.0.tgz

http://mirrors.shu.edu.cn/apache/kafka/2.1.0/kafka_2.12-2.1.0.tgz

http://mirrors.tuna.tsinghua.edu.cn/apache/kafka/2.1.0/kafka_2.12-2.1.0.tgz

比如我选择清华的地址

```
wget http://mirrors.tuna.tsinghua.edu.cn/apache/kafka/2.1.0/kafka_2.12-2.1.0.tgz
tar xf kafka_2.12-2.1.0.tgz
```

1.4 修改配置

这里全部使用默认配置。具体配置项含义可参考附录章节。

1.5 启动 zk

使用默认配置

```
cd kafka_2.12-2.1.0
./bin/zookeeper-server-start.sh -daemon config/zookeeper.properties
```

启动后 默认监听 2181 端口

停止 zk 服务的命令:

```
bin/zookeeper-server-stop.sh
```

1.6 启动 kafka

使用默认配置

```
cd kafka_2.12-2.1.0
./bin/kafka-server-start.sh -daemon config/server.properties
```

另外停止服务的命令：

```
./bin/kafka-server-stop.sh
```

1.7 观察

使用 jps 和 netstat 查看：

```
[root@tidb5 ~]# jps
9312 kafka
11095 jps
10777 QuorumPeerMain
[root@tidb5 ~]#
[root@tidb5 ~]# netstat -tnlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      947/sshd
tcp6       0      0 :::9092                 :::*                     LISTEN      9312/java
tcp6       0      0 :::2181                 :::*                     LISTEN      10777/java
tcp6       0      0 :::34794                :::*                     LISTEN      9312/java
tcp6       0      0 :::33811                :::*                     LISTEN      10777/java
tcp6       0      0 :::22                   :::*                     LISTEN      947/sshd
[root@tidb5 ~]#
```

进入 zk 查看，会出现如下内容：

```
[root@tidb5 kafka-2.12-2.1.0]#
[root@tidb5 kafka-2.12-2.1.0]# ./bin/zookeeper-shell.sh 127.0.0.1:2181/
Connecting to 127.0.0.1:2181/
Welcome to ZooKeeper!
JLine support is disabled

WATCHER::

watchedEvent state:SyncConnected type:None path:null
ls /
[cluster, controller_epoch, brokers, zookeeper, admin, isr_change_notification, consumers, log_dir_event_notification, latest_producer_id_block, config]
```

二、集群部署

2.1 规划

OS: CentOS 7.5 x64

JDK: 1.8.0_171

Kafka: 2.12-2.1.0

IP: 172.16.6.14-16

安装目录: /root/kafka_2.12-2.1.0

监听端口: 9092

Zookeeper: 10.1.5.203/204/205:2181

2.2 配置 JDK

下载地址: <http://www.oracle.com/technetwork/cn/java/archive-139210-zhs.html>

找到自己需要的版本, 上传到服务器

```
tar xf jdk-8u171-linux-x64.tar -C /usr/local
mv /usr/local/jdk1.8.0_171 /usr/local/jdk
```

#也可以创建软连接, 看个人习惯

保证/etc/profile 中有以下内容:

```
export JAVA_HOME=/usr/local/jdk
export PATH=$JAVA_HOME/bin:$PATH
```

#\$JAVA_HOME/bin 一定要放到\$PATH 前面, 防止识别到系统环境中其他版本的 jdk

```
source /etc/profile
```

#使配置生效

```
java -version
```

查看是否配置成功

2.3 下载和解压

操作主机: 172.16.6.14-16

下载地址: <http://kafka.apache.org/downloads>

选择合适的版本, 比如 2.12, 会跳到

https://www.apache.org/dyn/closer.cgi?path=/kafka/2.1.0/kafka_2.12-2.1.0.tgz

这里可以选择国内的源

HTTP

http://mirror.bit.edu.cn/apache/kafka/2.1.0/kafka_2.12-2.1.0.tgz

http://mirrors.hust.edu.cn/apache/kafka/2.1.0/kafka_2.12-2.1.0.tgz

http://mirrors.shu.edu.cn/apache/kafka/2.1.0/kafka_2.12-2.1.0.tgz

http://mirrors.tuna.tsinghua.edu.cn/apache/kafka/2.1.0/kafka_2.12-2.1.0.tgz

比如我选择清华的地址

```
wget http://mirrors.tuna.tsinghua.edu.cn/apache/kafka/2.1.0/kafka_2.12-2.1.0.tgz
tar xf kafka_2.12-2.1.0.tgz
```

2.4 修改配置文件

操作主机：172.16.6.14-16

kafka 的主配置文件是 /root/kafka_2.12-2.1.0/config/server.properties

主要关注以下 4 个配置项：

默认配置：

```
broker.id=0
#listeners=PLAINTEXT://:9092
log.dirs=/tmp/kafka-logs
zookeeper.connect=localhost:2181
```

只需要修改 broker.id 和 zookeeper.connect 两个配置项即可：

三台服务器可分别修改为如下内容，全局唯一即可。

```
broker.id=0
broker.id=1
broker.id=2
```

zookeeper.connect 都修改为如下：

```
zookeeper.connect=10.1.5.203:2181,10.1.5.204:2181,10.1.5.205:2181
```

注：参考第三章搭建 zk 集群，也可使用 kafka 自带的 zk，启动即可，也可使用单机的 zk，生产环境建议使用 zk 集群

这里只修改这几个配置项，具体配置项含义可参考附录章节。

2.5 启动集群

操作主机：172.16.6.14-16

启动服务前保证 zk 集群可用，参考第三章内容

启动服务：

```
cd kafka_2.12-2.1.0
./bin/kafka-server-start.sh -daemon config/server.properties
```

停止服务的命令：

```
./bin/kafka-server-stop.sh
```

启动后可看到监听默认端口 9092，也可看到相应进程

```
[root@tidb4 ~]#
[root@tidb4 ~]# jps
5604 Jps
4794 Kafka
[root@tidb4 ~]#
[root@tidb4 ~]#
[root@tidb4 ~]# netstat -tnlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              *:*                     LISTEN      946/sshd
tcp6       0      0 :::38435                :::*                     LISTEN      4794/java
tcp6       0      0 :::9092                 :::*                     LISTEN      4794/java
tcp6       0      0 :::22                   :::*                     LISTEN      946/sshd
[root@tidb4 ~]#
```

图中显示还会监听另一个随机端口 38435，该端口用于集群通信。

进入 zk 查看如下：

```
[zk: localhost:2181(CONNECTED) 13]
[zk: localhost:2181(CONNECTED) 13] get /brokers/ids/1
{"listener_security_protocol_map":{"PLAINTEXT":"PLAINTEXT"},"endpoints":["PLAINTEXT://172.16.6.15:9092"],"jmx_port":-1,"host":"172.16.6.15","timestamp":
  "port":9092,"version":4}
  czxid = 0x900000004c
  ctime = wed Jan 30 17:41:38 CST 2019
  mzxid = 0x900000004c
  mtime = wed Jan 30 17:41:38 CST 2019
  pzxid = 0x900000004c
  cversion = 0
  dataversion = 0
  aclVersion = 0
  ephemeralOwner = 0x30111394d0f0002
  datalength = 192
  numChildren = 0
[zk: localhost:2181(CONNECTED) 14] get /brokers/ids/2
{"listener_security_protocol_map":{"PLAINTEXT":"PLAINTEXT"},"endpoints":["PLAINTEXT://172.16.6.16:9092"],"jmx_port":-1,"host":"172.16.6.16","timestamp":
  "port":9092,"version":4}
  czxid = 0x9000000039
  ctime = wed Jan 30 17:41:19 CST 2019
  mzxid = 0x9000000039
  mtime = wed Jan 30 17:41:19 CST 2019
  pzxid = 0x9000000039
  cversion = 0
  dataversion = 0
  aclVersion = 0
  ephemeralOwner = 0x30111394d0f0001
  datalength = 192
  numChildren = 0
[zk: localhost:2181(CONNECTED) 15] ls /brokers/ids
[0, 1, 2]
[zk: localhost:2181(CONNECTED) 16]
[zk: localhost:2181(CONNECTED) 16]
```

2.6 安装 kafka-manage

该项目用于管理 kafka 集群，默认监听 9000 端口，我这里使用 172.16.6.16 部署。

2.6.1 下载和解压

github 地址：<https://github.com/yahoo/kafka-manager>

选择合适版本下载：

```
wget https://github.com/yahoo/kafka-manager/archive/1.3.3.22.tar.gz
tar xf 1.3.3.22.tar.gz
```

2.6.2 编译

编译安装

```
cd kafka-manager-1.3.3.22/  
./sbt clean dist
```

过程如下，比较耗时，耐心等待

```
[root@tidb6 kafka-manager-1.3.3.22]#  
[root@tidb6 kafka-manager-1.3.3.22]# ./sbt clean dist  
Downloading sbt launcher for 0.13.9:  
  From http://repo.typesafe.com/typesafe/ivy-releases/org.scala-sbt/sbt-launch/0.13.9/sbt-launch.jar  
  To /root/.sbt/launchers/0.13.9/sbt-launch.jar  
Getting org.scala-sbt sbt 0.13.9 ...  
  
downloading https://repo.typesafe.com/typesafe/ivy-releases/org.scala-sbt/sbt/0.13.9/jars/sbt.jar ...  
[SUCCESSFUL ] org.scala-sbt#sbt;0.13.9!sbt.jar (4406ms)  
downloading https://center.bintray.com/org/scala-lang/scala-library/2.10.5/scala-library-2.10.5.jar ...  
[SUCCESSFUL ] org.scala-lang#scala-library;2.10.5!scala-library.jar (5503ms)  
downloading https://repo.typesafe.com/typesafe/ivy-releases/org.scala-sbt/main/0.13.9/jars/main.jar ...  
[SUCCESSFUL ] org.scala-sbt#main;0.13.9!main.jar (9216ms)  
downloading https://repo.typesafe.com/typesafe/ivy-releases/org.scala-sbt/compiler-interface/0.13.9/jars/compiler-interface.jar ...  
[SUCCESSFUL ] org.scala-sbt#compiler-interface;0.13.9!compiler-interface.jar (4629ms)  
downloading https://repo.typesafe.com/typesafe/ivy-releases/org.scala-sbt/compiler-interface/0.13.9/jars/compiler-interface-bin.jar ...  
[SUCCESSFUL ] org.scala-sbt#compiler-interface;0.13.9!compiler-interface-bin.jar (5725ms)  
downloading https://repo.typesafe.com/typesafe/ivy-releases/org.scala-sbt/precompiled-2.8.2/0.13.9/jars/compiler-interface-bin.jar ...  
[SUCCESSFUL ] org.scala-sbt#precompiled-2.8.2;0.13.9!compiler-interface-bin.jar (5058ms)  
downloading https://repo.typesafe.com/typesafe/ivy-releases/org.scala-sbt/precompiled-2.9.2/0.13.9/jars/compiler-interface-bin.jar ...  
[SUCCESSFUL ] org.scala-sbt#precompiled-2.9.2;0.13.9!compiler-interface-bin.jar (5898ms)  
downloading https://repo.typesafe.com/typesafe/ivy-releases/org.scala-sbt/precompiled-2.9.3/0.13.9/jars/compiler-interface-bin.jar ...  
[SUCCESSFUL ] org.scala-sbt#precompiled-2.9.3;0.13.9!compiler-interface-bin.jar (5456ms)  
downloading https://repo.typesafe.com/typesafe/ivy-releases/org.scala-sbt/actions/0.13.9/jars/actions.jar ...
```

编译完成之后会生成一个 zip 包，如下图

Your package is ready in /root/kafka-manager-1.3.3.22/target/universal/kafka-manager-1.3.3.22.zip

```
[info] LESS compiling on 1 source(s)  
[info] Packaging /root/kafka-manager-1.3.3.22/target/scala-2.11/kafka-manager_2.11-1.3.3.22-web-assets.jar ...  
[info] Done packaging.  
[info] Packaging /root/kafka-manager-1.3.3.22/target/scala-2.11/kafka-manager_2.11-1.3.3.22.jar ...  
[info] Done packaging.  
[info] Packaging /root/kafka-manager-1.3.3.22/target/scala-2.11/kafka-manager_2.11-1.3.3.22-sans-externalized.jar ...  
[info] Done packaging.  
[info] Your package is ready in /root/kafka-manager-1.3.3.22/target/universal/kafka-manager-1.3.3.22.zip  
[info] [success] Total time: 1031 s, completed 2019-1-31 10:23:50  
[root@tidb6 kafka-manager-1.3.3.22]#  
[root@tidb6 kafka-manager-1.3.3.22]#
```

2.6.3 启动

```
cd target/universal/  
unzip kafka-manager-1.3.3.22.zip  
cd kafka-manager-1.3.3.22
```

修改配置文件：

```
vim conf/application.conf
```

```
kafka-manager.zkhosts="10.1.5.203:2181,10.1.5.204:2181,10.1.5.205:2181"
```

启动：

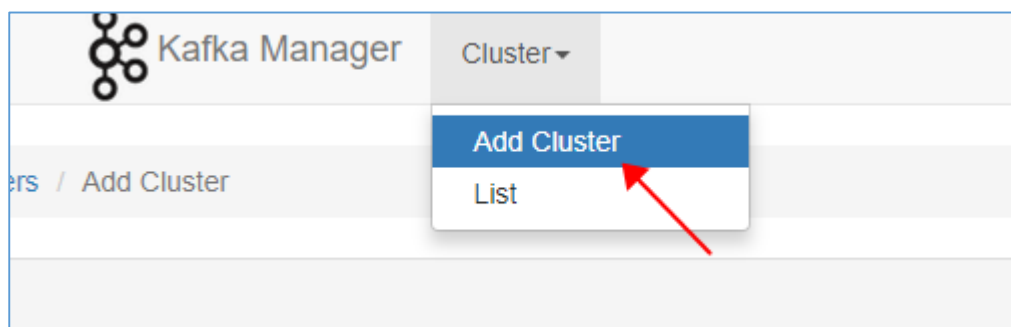
```
nohup ./bin/kafka-manager &
```

启动后监听 9000 端口。

2.6.4 添加集群

使用浏览器访问：<http://172.16.6.16:9000/>

添加集群：



添加信息如下：

The screenshot shows the 'Add Cluster' form. At the top is a back arrow and the title 'Add Cluster'. Below this are three main sections: 'Cluster Name' with a text input containing 'mykafka'; 'Cluster Zookeeper Hosts' with a text input containing '10.1.5.203:2181,10.1.5.204:2181,10.1.5.205:2181'; and 'Kafka Version' with a dropdown menu set to '2.0.0'. At the bottom of this section is a checkbox labeled 'Enable JMX Polling (Set JMX_PORT env variable before starting kafka server)' which is checked.The screenshot shows the configuration options section of the 'Add Cluster' form. It contains a list of checkboxes, all of which are checked: 'JMX with SSL', 'Enable Logkafka', 'Poll consumer information (Not recommended for large # of consumers)', 'Filter out inactive consumers', 'Enable Active OffsetCache (Not recommended for large # of consumers)', and 'Display Broker and Topic Size (only works after applying this patch)'. Below this list is a partially visible label 'brokerViewUpdatePeriodSeconds'.

添加好保存：

The screenshot shows the bottom of the 'Add Cluster' form. It features two buttons: a blue 'Save' button and a white 'Cancel' button with a grey border. A red arrow points from the bottom left towards the 'Save' button.

添加完成之后，可看到如下

Clusters

Clusters

Active	Operations	Version
mykafka	<div>Modify</div> <div>Disable</div>	2.0.0

Clusters / mykafka / Summary

Cluster Information

Zookeepers

10.1.5.203:2181 10.1.5.204:2181 10.1.5.205:2181

Version

2.0.0

Cluster Summary

Topics

1

Brokers

3

Kafka Manager

mykafka

Cluster

Brokers

Topic

Preferred Replica Election

Reassign Partitions

Consumers

Logkafka

Clusters / mykafka / Brokers

← Brokers

Id	Host	Port	JMX Port	Bytes In	Bytes Out	Size
0	172.16.6.14	PLAINTEXT:9092	-1	0.00	0.00	0 B
1	172.16.6.15	PLAINTEXT:9092	-1	0.00	0.00	0 B
2	172.16.6.16	PLAINTEXT:9092	-1	0.00	0.00	0 B

Combined Metrics

Rate	Mean	1 min	5 min	15 min
Messages in /sec	0.00	0.00	0.00	0.00
Bytes in /sec	0.00	0.00	0.00	0.00
Bytes out /sec	0.00	0.00	0.00	0.00
Bytes rejected /sec	0.00	0.00	0.00	0.00
Failed fetch request /sec	0.00	0.00	0.00	0.00
Failed produce request /sec	0.00	0.00	0.00	0.00

三、测试

3.1 添加 topic

```
bin/kafka-topics.sh --create --zookeeper 10.1.5.203:2181 --replication-factor 3 --partitions 1 --topic my-topic
```

```
[root@tidb4 kafka_2.12-2.1.0]# bin/kafka-topics.sh --create --zookeeper 10.1.5.203:2181 --replication-factor 3 --partitions 1 --topic my-topic
Created topic "my-topic".
[root@tidb4 kafka_2.12-2.1.0]#
```

使用 kafka-manage 查看：

Clusters / mykafka / Topics / my-topic

← my-topic

Topic Summary

Replication	3
Number of Partitions	1
Sum of partition offsets	0
Total number of Brokers	3
Number of Brokers for Topic	2
Preferred Replicas %	100
Brokers Skewed %	0
Brokers Leader Skewed %	0
Brokers Spread %	66

Operations

Delete Topic

Reassign Partitions

Generate Partition Assignments

Add Partitions

Update Config

Manual Partition Assignments

Partitions by Broker

Broker	# of Partitions	# as Leader	Partitions	Skewed?	Leader Skewed
1	1	0	(0)	false	false
2	1	1	(0)	false	false

也可使用如下命令查看：

```
bin/kafka-topics.sh --list --zookeeper 10.1.5.203:2181
```

```
[root@tidb4 kafka_2.12-2.1.0]#
[root@tidb4 kafka_2.12-2.1.0]# bin/kafka-topics.sh --list --zookeeper 10.1.5.203:2181
__consumer_offsets
my-topic
[root@tidb4 kafka_2.12-2.1.0]#
```

四、安装 zookeeper 集群

4.1 规划

zookeeper 集群至少需要三个节点，规划如下：

10.1.5.203, clientPort=2181

10.1.5.204, clientPort=2181

10.1.5.205, clientPort=2181

4.2 配置jdk 环境

建议使用 oracle 的 jdk，当然使用系统自带的 openjdk 也是可以的，见 1.4。

4.3 下载 zookeeper

下载地址：<http://www.apache.org/dist/zookeeper/>
`tar xvzf zookeeper-3.4.12.tgz -C /data/`

4.4 修改配置文件

在 10.1.5.203 上：

`cd /data/zookeeper-3.4.12/conf`

`cp zoo_sample.cfg zoo.cfg`

修改配置文件如下：

`cat zoo.cfg`

`tickTime=2000`

`initLimit=10`

`syncLimit=5`

`dataDir=/data/zookeeper-data`

`clientPort=2181`

`server.1=10.1.5.203:2888:3888`

`server.2=10.1.5.204:2888:3888`

`server.3=10.1.5.205:2888:3888`

4.5 配置其他节点

同 6.1.4 完全相同，配置 10.1.5.204 和 10.1.5.205

4.6 新建好相关目录

三台机器上均操作：

```
mkdir -pv /data/zookeeper-data
```

4.7 新建 myid 文件

10.1.5.203 上：

```
echo 1 > /data/zookeeper-data/myid
```

10.1.5.204 上：

```
echo 2 > /data/zookeeper-data/myid
```

10.1.5.205 上：

```
echo 3 > /data/zookeeper-data/myid
```

4.8 启动

```
cd /data/zookeeper-3.4.12/bin
```

```
./zkServer.sh start
```

以此启动 3 个节点。

bin 目录下会生产 zookeeper.out 文件

4.9 查看状态

分布在 3 个节点上执行

```
/data/zookeeper-3.4.12/bin/zkServer.sh status
```

会有以下两种结果，leader 表示主，follower 表示从：

Mode: leader

Mode: follower

3 点的分布输入 jps，会看到类似如下的进程

```
[root@hadoop04 ~]# jps
```

```
11628 QuorumPeerMain
```

```
11950 Jps
```

五、附录-配置文件详解

5.1 server.properties

#broker 的全局唯一编号，集群中不能重复

broker.id=0

#监听的 IP 和端口，producer 或 consumer 将在此端口建立连接

listeners=PLAINTEXT://host_name:9092

#处理网络请求的线程数量，也就是接收消息的线程数。

#接收线程会将接收到的消息放到内存中，然后再从内存中写入磁盘。

num.network.threads=3

#消息从内存中写入磁盘是时使用的线程数量

#用来处理磁盘 IO 的线程数量

num.io.threads=8

#发送套接字的缓冲区大小

socket.send.buffer.bytes=102400

#接受套接字的缓冲区大小

socket.receive.buffer.bytes=102400

#请求套接字的缓冲区大小

socket.request.max.bytes=104857600

#kafka 运行日志存放的路径

log.dirs=/tmp/kafka-logs

#topic 在当前 broker 上的分片个数

num.partitions=2

#我们知道 segment 文件默认会被保留 7 天的时间，超时的话就

#会被清理，那么清理这件事情就需要有一些线程来做。这里就是

#用来设置恢复和清理 data 下数据的线程数量

num.recovery.threads.per.data.dir=1

#上面我们说过接收线程会将接收到的消息放到内存中，然后再从内存

#写到磁盘上，那么什么时候将消息从内存中写入磁盘，就有一个

#时间限制（时间阈值）和一个数量限制（数量阈值），这里设置的是

#数量阈值，下一个参数设置的则是时间阈值。

#partition buffer 中，消息的条数达到阈值，将触发 flush 到磁盘。

`log.flush.interval.messages=10000`

#消息 buffer 的时间，达到阈值，将触发将消息从内存 flush 到磁盘，
#单位是毫秒。

`log.flush.interval.ms=1000`

#segment 文件保留的最长时间，默认保留 7 天（168 小时），
#超时将被删除，也就是说 7 天之前的数据将被清理掉。

`log.retention.hours=168`

#日志文件中每个 segment 的大小，默认为 1G

`log.segment.bytes=1073741824`

#上面的参数设置了每一个 segment 文件的大小是 1G，那么
#就需要有一个东西去定期检查 segment 文件有没有达到 1G，
#多长时间去检查一次，就需要设置一个周期性检查文件大小
#的时间（单位是毫秒）。

`log.retention.check.interval.ms=300000`

#broker 需要使用 zookeeper 保存 meta 数据

`zookeeper.connect=zk01:2181,zk02:2181,zk03:2181`

#zookeeper 连接超时时间

`zookeeper.connection.timeout.ms=6000`