

Kafka-学习笔记-常用Shell命令汇总

Topic



创建Topic

```
> bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic test  
  
> bin/kafka-topics.sh --create --zookeeper localhost:2181/dir \  
--replication-factor 1 --partitions 1 --topic test --consumer-property group.id=groupid  
  
> bin/kafka-topics.sh --zookeeper zk_host:port/chroot --create --topic my_topic_name  
--partitions 20 --replication-factor 3 --config x=y
```



查询Topic列表

```
> bin/kafka-topics.sh --list --zookeeper localhost:2181  
如果您正在使用老的高级 consumer 并在ZooKeeper中存储组元数据（即 offsets.storage = zookeeper），则传递  
--zookeeper 而不是bootstrap-server  
  
> bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --list
```



查询Topic描述

```
> bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic my-replicated-topic
```



查询集群描述

```
> bin/kafka-topics.sh --describe --zookeeper localhost:2181
```



增加Topic分区数

```
> bin/kafka-topics.sh --zookeeper localhost:2181 --alter --topic test1 --partitions 10
```



增加一个配置项

```
> bin/kafka-configs.sh --zookeeper zk_host:port/chroot --entity-type topics --entity-name my_topic_name --alter --add-  
config x=y
```



删除一个配置项

```
> bin/kafka-configs.sh --zookeeper zk_host:port/chroot --entity-type topics --entity-name my_topic_name --alter --delete-  
config x
```



删除一个 topic

```
> bin/kafka-topics.sh --zookeeper zk_host:port/chroot --delete --topic my_topic_name
```

Producer



生产信息

```
> bin/kafka-console-producer.sh --broker-list localhost:9092 --topic test
```

Consumer



消费信息

```
> bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test --from-beginning 从头开始  
> bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test 默认从尾部开始  
> bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test --offset earliest --partition 1 指定分区  
> bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test --offset latest --partition 1 --max-  
messages 1 指定个数，默认从尾部开始
```



新消费者

```
> bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test --new-consumer --from-beginning  
--consumer.config config/consumer.properties
```



消费者列表查询

```
> bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --list
```



显示某个消费组的消费进度

```
> bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --describe --group test-consumer-group  
注意：这将仅显示使用Java consumer API（基于非ZooKeeper的 consumer）的 consumer的信息。  
  
> bin/kafka-consumer-groups.sh --zookeeper localhost:2181 --describe --group my-group  
注意：这只会显示关于使用ZooKeeper的 consumer的信息（不是那些使用Java consumer API的消费者）。
```

管理



平衡leader

```
> bin/kafka-preferred-replica-election.sh --zookeeper localhost:2181/chroot
```



集群之间镜像数据

```
> bin/kafka-mirror-maker.sh  
--consumer.config consumer.properties  
--producer.config producer.properties --whitelist my-topic
```



查看流量限制配置

```
> bin/kafka-configs.sh --describe --zookeeper localhost:2181 --entity-type brokers
```



查看节流副本的列表

```
> bin/kafka-configs.sh --describe --zookeeper localhost:2181 --entity-type topics
```



设置配额

```
为 (user = user1, client-id = clientA) 配置自定义配额:  
> bin/kafka-configs.sh --zookeeper localhost:2181 --alter --add-config 'producer_byte_rate=1024,consumer_byte_rate=  
2048,request_percentage=200' --entity-type users --entity-name user1 --entity-type clients --entity-name clientA  
  
为 user = user1配置自定义配额:  
> bin/kafka-configs.sh --zookeeper localhost:2181 --alter --add-config 'producer_byte_rate=1024,consumer_byte_rate=  
2048,request_percentage=200' --entity-type users --entity-name user1  
  
为 client-id=clientA配置自定义配额:  
> bin/kafka-configs.sh --zookeeper localhost:2181 --alter --add-config 'producer_byte_rate=1024,consumer_byte_rate=  
2048,request_percentage=200' --entity-type clients --entity-name clientA  
  
为用户配置默认配额:  
> bin/kafka-configs.sh --zookeeper localhost:2181 --alter --add-config 'producer_byte_rate=1024,consumer_byte_rate=  
2048,request_percentage=200' --entity-type users --entity-default  
  
配置client-id的默认配额:  
> bin/kafka-configs.sh --zookeeper localhost:2181 --alter --add-config 'producer_byte_rate=1024,consumer_byte_rate=  
2048,request_percentage=200' --entity-type clients --entity-default  
  
描述给定用户的配额:  
> bin/kafka-configs.sh --zookeeper localhost:2181 --describe --entity-type users --entity-name user1  
  
描述给定 client-id的配额:  
> bin/kafka-configs.sh --zookeeper localhost:2181 --describe --entity-type clients --entity-name clientA  
  
如果未指定实体名称，则描述指定类型的所有实体:  
> bin/kafka-configs.sh --zookeeper localhost:2181 --describe --entity-type users  
> bin/kafka-configs.sh --zookeeper localhost:2181 --describe --entity-type users --entity-type clients
```

压测

```
> bin/kafka-producer-perf-test.sh  
--topic test  
--num-records 100  
--record-size 1  
--throughput 100  
--producer-props bootstrap.servers=localhost:9092
```

```
> bin/kafka-producer-perf-test.sh \  
--producer.config config/producer-test.properties \  
--topic gnutest \  
--throughput 2000000 \  
--num-records 6000000 \  
--payload-file message.txt
```