

从所有教程的词条中查询...

Java / 7 Kafka技巧篇

全部开发者教程

3 Kafka使用初体验

4 Kafka核心扩展内容

5 Kafka核心之存储和容错机制

6 Kafka生产者消费者实战

7 Kafka技巧篇

8 Kafka小试牛刀实战篇

9 Kafka核心复盘

第15周-极速上手内存数据库Redis

1 快速了解Redis

2 Redis核心实践

3 Redis封装工具类技巧

4 Redis高级特性



徐老师 • 更新于 2020-09-25

上一节 6 Kafka生产者消费... 8 Kafka小试牛刀... 下一节

Kafka集群参数调优

JVM参数调优

默认启动的Broker进程只会使用1G内存，在实际使用中会导致进程频繁GC，影响Kafka集群的性能和稳定性

通过 `jstat -gcutil <pid> 1000` 查看到kafka进程GC情况

主要看 YGC, YGCT, FGC, FGCT 这几个参数，如果这几个值不是很大，就没什么问题

<> 代码块

```
1 YGC: young gc发生的次数
2 YGCT: young gc消耗的时间
3 FGC: full gc发生的次数
4 FGCT: full gc消耗的时间
```

<> 代码块

```
1 [root@bigdata01 kafka_2.12-2.4.1]# jps
2 13248 Kafka
3 18087 Jps
4 1679 QuorumPeerMain
5 [root@bigdata01 kafka_2.12-2.4.1]# jstat -gcutil 13248 1000
6  S0    S1    E      O      M      CCS    YGC    YGCT    FGC    FGCT    GCT
7   0.00 100.00 78.00 14.50 89.97 92.39   28    0.563    0    0.000    0
8   0.00 100.00 78.00 14.50 89.97 92.39   28    0.563    0    0.000    0
9   0.00 100.00 78.00 14.50 89.97 92.39   28    0.563    0    0.000    0
10  0.00 100.00 78.00 14.50 89.97 92.39   28    0.563    0    0.000    0
11  0.00 100.00 78.00 14.50 89.97 92.39   28    0.563    0    0.000    0
12  0.00 100.00 78.00 14.50 89.97 92.39   28    0.563    0    0.000    0
```

如果你发现 YGC 很频繁，或者 FGC 很频繁，就说明内存分配的少了

此时需要修改 `kafka-server-start.sh` 中的 `KAFKA_HEAP_OPTS`

<> 代码块

```
1 export KAFKA_HEAP_OPTS="-Xmx10g -Xms10g -XX:MetaspaceSize=96m -XX:+UseG1GC ..."
```

这个配置表示给kafka分配了10G内存，我们的Kafka服务器是16G内存

Replication参数调优

`replica.socket.timeout.ms=60000`

这个参数的默认值是30秒，它是控制partiton副本之间socket通信的超时时间，如果设置的太小，有可能会由于网络原因导致造成误判，认为某一个partition副本连不上了。

`replica.lag.time.max.ms=50000`

如果一个副本在指定的时间内没有向leader节点发送任何请求，或者在指定的时间内没有同步完leader中的数据，则leader会将这个节点从Isr列表中移除。

意见反馈

收藏教程

标记书签

如果网络不好，或者kafka压力较大，建议调大该值，否则可能会频繁出现副本丢失，进而导致集群需要频繁复制副本，导致集群压力更大，会陷入一个恶性循环

Log参数调优

这块是针对Kafka中数据文件的删除时机进行设置，不是对kafka本身的日志参数配置

```
log.retention.hours=24
```

这个参数默认值为168，单位是小时，就是7天，默认对数据保存7天，可以在这调整数据保存的时间，我们在实际工作中改为了只保存1天，因为kafka中的数据我们会在hdfs中进行备份，保存一份，所以就没有必要在kafka中保留太长时间了。

在kafka中保留只是为了能够让你在指定的时间内恢复数据，或者重新消费数据，如果没有这种需求，那就没有必要设置太长时间。

注意：

这里分析的 Replication 的参数和 Log 参数都是在 `server.properties` 文件中进行配置

JVM 参数是在 `kafka-server-start.sh` 脚本中配置

Kafka Topic命名小技巧

针对Kafka中Topic命名的小技巧

建议在给topic命名的时候在后面跟上r2p10之类的内容

r2: 表示Partition的副本因子是2

p10: 表示这个Topic的分区数是10

这样的好处是后期我们如果要写消费者消费指定topic的数据，通过topic的名称我们就知道应该设置多少个消费者消费数据效率最高。

因为一个partition同时只能被一个消费者消费，所以效率最高的情况就是消费者的数量和topic的分区数量保持一致。

在这里通过topic的名称就可以直接看到，一目了然。

但是也有一个缺点，就是后期如果我们动态调整了topic的partiton，那么这个topic名称上的partition数量就不准了，针对这个topic，建议大家一开始的时候就提前预估一下，可以多设置一些partition，我们在工作中的时候针对一些数据量比较大的topic一般会设置4050个partition，数据量少的topic一般设置510个partition，这样后期调整topic partiton数量的场景就比较少少了。

Kafka集群监控管理工具

现在我们操作Kafka都是在命令行界面中通过脚本操作的，后面需要传很多参数，用起来还是比较麻烦的，那kafka有没有提供web界面的支持吗？

很遗憾的告诉你，Apache官方并没有提供，不过好消息是有一个由雅虎开源的一个工具，目前用起来还是不错的。

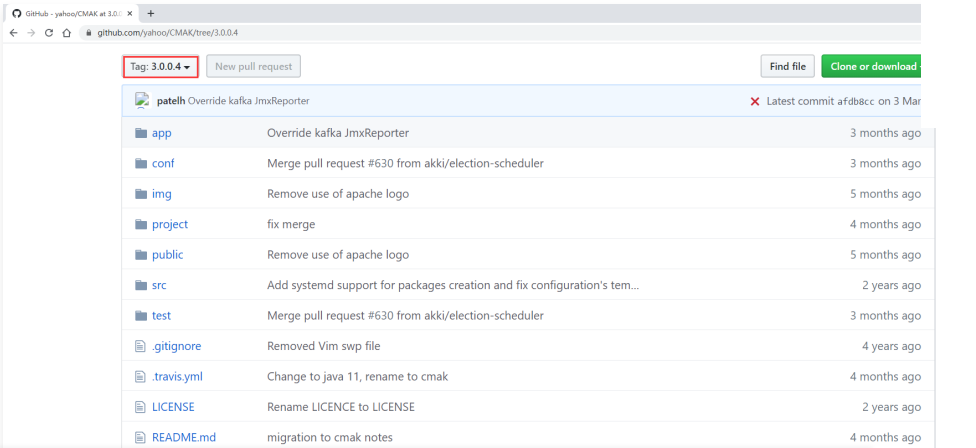
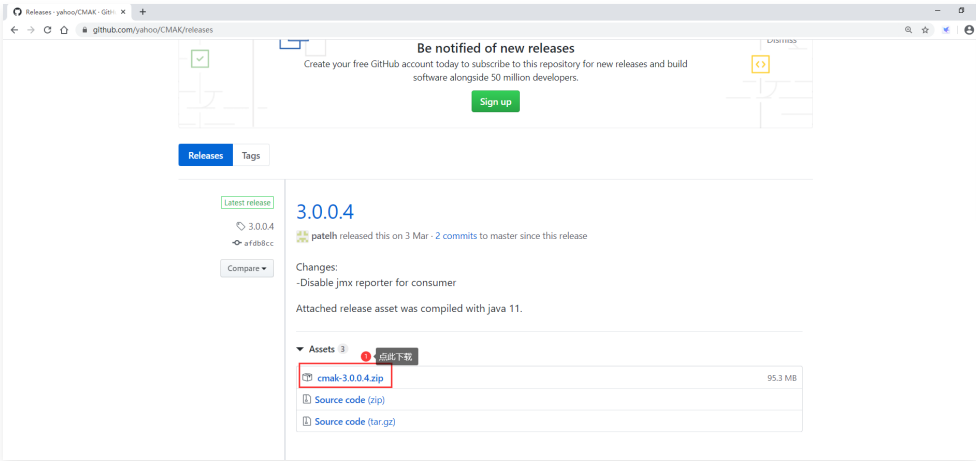
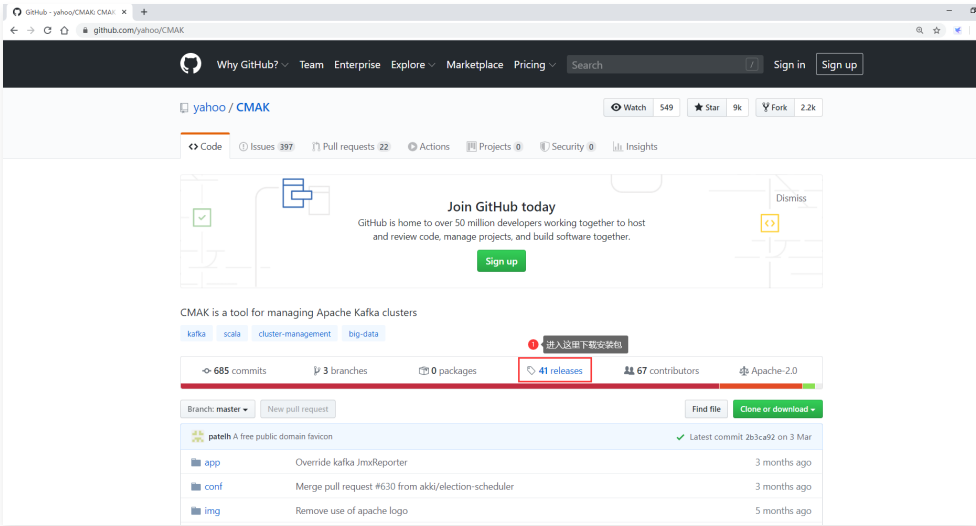
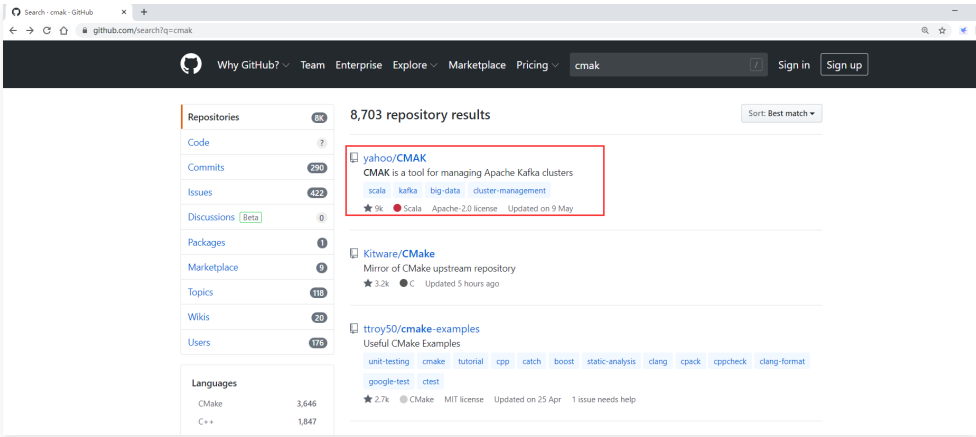
它之前的名字叫 `KafkaManager`，后来改名字了，叫 `CMAK`

CMAK是目前最受欢迎的Kafka集群管理工具，最早由雅虎开源，用户可以在Web界面上操作Kafka：可以轻松检查集群状态(Topic、Consumer、Offset、Brokers、Replica、Partition)

那下面我们先去下载这个CMAK

需要到github上面去下载

在github里面搜索 CMAK即可



意见反馈

收藏教程

标记书签

注意：由于cmak-3.0.0.4.zip版本是在java11这个版本下编译的，所以在运行的时候也需要使用java11这个版本，我们目前服务器上使用的是java8这个版本

我们为什么不使用java11版本呢？因为自2019年1月1日起，java8之后的更新版本在商业用途的时候就需要收费授权了。

在这针对cmak-3.0.0.4这个版本，如果我们想要使用的话有两种解决办法

- 1: 下载cmak的源码，使用jdk8编译
- 2: 额外安装一个jdk11

如果想要编译的话需要安装sbt这个工具对源码进行编译，sbt是Scala的构建工具，类似于Maven。由于我们在这使用不属于商业用途，所以使用jdk11是没有问题的，那就不用重新编译了。

下载jdk11，jdk-11.0.7_linux-x64_bin.tar.gz

将jdk11的安装包上传到bigdata01的/data/soft目录下

只需要解压即可，不需要配置环境变量，因为只有cmak这个工具才需要使用jdk11

<> 代码块

```
1 [root@bigdata01 soft]# tar -zxvf jdk-11.0.7_linux-x64_bin.tar.gz
```

接下来把cmak-3.0.0.4.zip上传到bigdata01的/data/soft目录下

1: 解压

<> 代码块

```
1 [root@bigdata01 soft]# unzip cmak-3.0.0.4.zip
2 -bash: unzip: command not found
```

注意：如果提示-bash: unzip: command not found，则说明目前不支持unzip命令，可以使用yum在线安装

建议先清空一下yum缓存，否则使用yum可能无法安装unzip

<> 代码块

```
1 [root@bigdata01 soft]# yum clean all
2 Loaded plugins: fastestmirror
3 Cleaning repos: base extras updates
4 Cleaning up list of fastest mirrors
```

<> 代码块

```
1 [root@bigdata01 soft]# yum install -y unzip
2 Loaded plugins: fastestmirror
3 .....
4 Running transaction
5   Installing : unzip-6.0-21.el7.x86_64                1/1
6   Verifying   : unzip-6.0-21.el7.x86_64                1/1
7
8 Installed:
9   unzip.x86_64 0:6.0-21.el7
10
11 Complete!
```

再重新解压

<> 代码块

```
1 [root@bigdata01 soft]# unzip cmak-3.0.0.4.zip
```

2: 修改CMAK配置

首先修改bin目录下的 `cmak` 脚本

在里面配置 `JAVA_HOME` 指向 `jdk11` 的安装目录, 否则默认会使用 `jdk8`

<> 代码块

```
1 [root@bigdata01 soft]# cd cmak-3.0.0.4
2 [root@bigdata01 cmak-3.0.0.4]# cd bin/
3 [root@bigdata01 bin]# vi cmak
4 ....
5 JAVA_HOME=/data/soft/jdk-11.0.7
6 .....
```

然后修改 `conf` 目录下的 `application.conf` 文件

只需要在里面增加一行 `cmak.zkhosts` 参数的配置即可, 指定 `zookeeper` 的地址

注意: 在这里指定 `zookeeper` 地址主要是为了让 `CMAK` 在里面保存数据, 这个 `zookeeper` 地址不一定是 `kafka` 集群使用的那个 `zookeeper` 集群, 随便哪个 `zookeeper` 集群都可以。

<> 代码块

```
1 [root@bigdata01 cmak-3.0.0.4]# cd conf/
2 [root@bigdata01 conf]# vi application.conf
3 ....
4 cmak.zkhosts="bigdata01:2181,bigdata02:2181,bigdata03:2181"
5 .....
```

3: 修改kafka启动配置

想要在 `CMAK` 中查看 `kafka` 的一些指标信息, 在启动 `kafka` 的时候需要指定 `JMX_PORT`

停止 `kafka` 集群

<> 代码块

```
1 [root@bigdata01 kafka_2.12-2.4.1]# bin/kafka-server-stop.sh
2 [root@bigdata02 kafka_2.12-2.4.1]# bin/kafka-server-stop.sh
3 [root@bigdata03 kafka_2.12-2.4.1]# bin/kafka-server-stop.sh
```

重新启动 `kafka` 集群, 指定 `JMX_PORT`

<> 代码块

```
1 [root@bigdata01 kafka_2.12-2.4.1]# JMX_PORT=9988 bin/kafka-server-start.sh -d
2 [root@bigdata02 kafka_2.12-2.4.1]# JMX_PORT=9988 bin/kafka-server-start.sh
3 [root@bigdata03 kafka_2.12-2.4.1]# JMX_PORT=9988 bin/kafka-server-start.sh
```

4: 启动cmak

<> 代码块

```
1 [root@bigdata01 cmak-3.0.0.4]# bin/cmak -Dconfig.file=conf/application.conf -
```

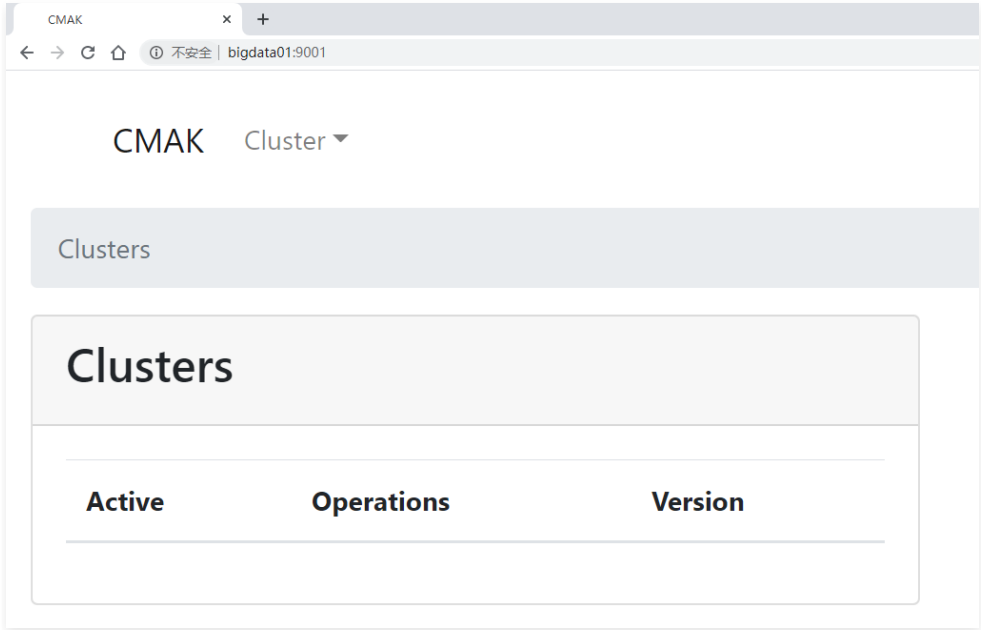
如果要把 `cmak` 放在后台执行的话需要添加上 `nohup` 和 `&`

<> 代码块

lication.

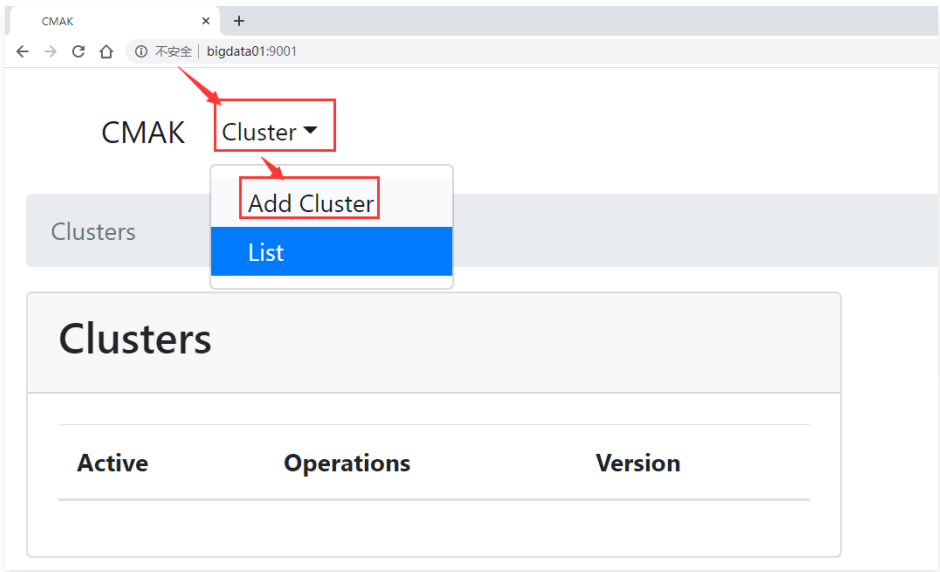
[意见反馈](#)[收藏教程](#)[标记书签](#)

5: 访问cmak
<http://bigdata01:9001/>



6: 操作CMAK

- 添加集群



Add Cluster

← → ↺ ⌂ 不安全 | bigdata01:9001/addCluster

1000

kafkaManagedOffsetMetadataCheckMillis

30000

kafkaManagedOffsetGroupCacheSize

1000000

kafkaManagedOffsetGroupExpireDays

7

Security Protocol

PLAINTEXT

SASL Mechanism (only applies to SASL based security)

DEFAULT

SASL JAAS Config (only applies to SASL based security)

SaveCancel

最后进来是这样的

CMAK

← → ↺ ⌂ 不安全 | bigdata01:9001/clusters/kafka2.4.1

CMAK **kafka2.4.1** Cluster ▾ Brokers ▾ Topic ▾ Preferred Replica Election ▾ Schedule Leader Election ▾ Reassign Partitions ▾ Consumers ▾

Clusters / kafka2.4.1 / Summary

Cluster Information

Zookeepers

bigdata01:2181 bigdata02:2181 bigdata03:2181

Version

2.4.0

Cluster Summary

Topics

3

Brokers

3

- 查看kafak集群的所有broker信息

Broker List

← → ↻ ⌂ ① 不安全 | bigdata01:9001/clusters/kafka2.4.1/brokers

CMAX kafka2.4.1 Cluster **Brokers** Topic ▾ Preferred Replica Election Schedule Leader Election Reassign Partitions Consumers

Clusters / kafka2.4.1 / Brokers

Brokers

Id	Host	Port	JMX Port	Bytes In	Bytes Out
0	bigdata01	PLAINTEXT:9092	9988	0.00	0.00
1	bigdata02	PLAINTEXT:9092	9988	0.03	0.00
2	bigdata03	PLAINTEXT:9092	9988	0.21	0.00

Combined Metrics 聚合指标: 所有Broker节点的指标聚合

Rate

平均值

按分钟统计

	Mean	1 min	5 min	15 min
Messages in /sec	0.01	0.00	0.00	0.00
Bytes in /sec	2.52	0.24	1.11	0.62
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

- 查看kafak集群的所有topic信息

Topic List

← → ↻ ⌂ ① 不安全 | bigdata01:9001/clusters/kafka2.4.1/topics

CMAX kafka2.4.1 Cluster ▾ Brokers Topic ▾ Preferred Replica Election Schedule Leader Election Reassign Partitions Consumers

Clusters / kafka2.4.1 / Topics

Operations

Generate Partition Assignments Run Partition Assignments Add Partitions

Topics

Show 10 entries

Topic	# Partitions	# Brokers	Brokers Spread %	Brokers Skew %	Brokers Leader Skew %	# Replicas	Under Replicated %
88888888	5	3	100	0	0	2	0
88888888-__consumer_offsets	50	3	100	0	0	1	0
hello	5	3	100	0	0	2	0

Showing 1 to 3 of 3 entries

- 查看某一个topic的详细信息

Topic View

← → ↻ ⌂ ① 不安全 | bigdata01:9001/clusters/kafka2.4.1/topics/hello

CMAX kafka2.4.1 Cluster ▾ Brokers Topic ▾ Preferred Replica Election Schedule Leader Election Reassign Partitions Consumers

Clusters / kafka2.4.1 / Topics / hello

hello

Topic Summary

Topic的概述信息

Replication	2
Number of Partitions	5
Sum of partition offsets	0
Total number of Brokers	3
Number of Brokers for Topic	3
Preferred Replicas %	100
Brokers Skewed %	0
Brokers Leader Skewed %	0
Brokers Spread %	100
Under-replicated %	0

Operations

Topic的操作

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?
0	3	2	(0,1,4)	false	false
1	3	1	(1,2,3)	false	false
2	4	2	(0,2,3,4)	false	false

Metrics

topic的聚合指标

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	0.16	0.00	3.14	9.22
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

Partition Information

topic2的分区信息

Partition	Latest Offset	Leader	Replicas	In Sync Replicas	Preferred Leader?	Under Replicated?
0	0	2	(2,0)	(0,2)	true	false
1	0	0	(0,1)	(0,1)	true	false
2	1	1	(1,2)	(2,1)	true	false
3	1	2	(2,1)	(2,1)	true	false
4	3	0	(0,2)	(0,2)	true	false

Consumers consuming from this topic

topic2的消费者信息

注意：启动CMAK之前的消费者信息在这不会显示，可以重新运行一下消费者在这就可以看到

con-2

KF

点击topic的消费者信息是可以进来查看的

Consumed Topic View

bigdata01:9001/clusters/kafka2.4.1/consumers/con-2/topic/hello/type/KF

CMAK kafka2.4.1 Cluster Brokers Topic Preferred Replica Election Schedule Leader Election Reassign Partitions Consumers

Clusters / kafka2.4.1 / Consumers / con-2 / hello

con-2 / hello

消费者的信息

Topic Summary

Total Lag0

% of Partitions assigned to a consumer instance100

hello

Partition	LogSize	Consumer Offset	Lag	Consumer Instance Owner
0	0	0	0	consumer-con-2-1-d45c7d08-381e-4401-a6c5-d15a3230e500/192.168.182.1
1	0	0	0	consumer-con-2-1-d45c7d08-381e-4401-a6c5-d15a3230e500/192.168.182.1
2	1	1	0	consumer-con-2-1-d45c7d08-381e-4401-a6c5-d15a3230e500/192.168.182.1
3	1	1	0	consumer-con-2-1-d45c7d08-381e-4401-a6c5-d15a3230e500/192.168.182.1
4	3	3	0	consumer-con-2-1-d45c7d08-381e-4401-a6c5-d15a3230e500/192.168.182.1

• 创建一个topic

Create Topic

bigdata01:9001/clusters/kafka2.4.1/createTopic

CMAK kafka2.4.1 Cluster Brokers Topic Preferred Replica Election Schedule Leader Election Reassign Partitions Consumers

Clusters / kafka2.4.1 / Topics / Create Topic

Create Topic

Topic

test

Partitions

5

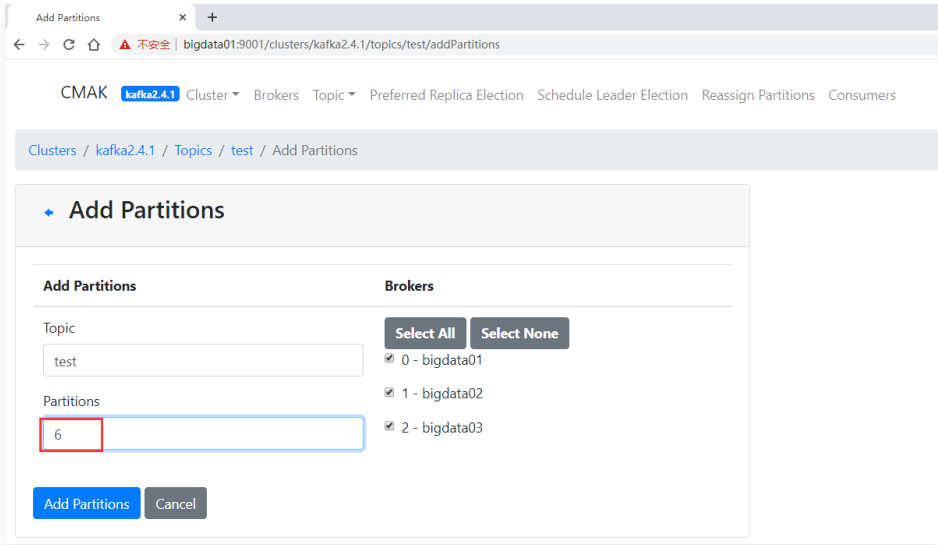
Replication Factor

2

Create

Cancel

• 给topic增加分区



这是CMAK中常见的功能，当然了这里面还要一些我们没有说到的功能就留给大家以后来发掘了。