

Kafka 常见问题维护手册 V1.0

Kafka 常见问题维护手册 V1.0

文档版本 01
发布日期 2016-03-31

华为技术有限公司



版权所有 © 华为技术有限公司 2016。 保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址： <http://www.huawei.com>

客户服务邮箱： support@huawei.com

客户服务电话： 4008302118

目 录

KAFKA.....	3
1、基本概念.....	3
【概述】	3
【日志概述】	3
【客户端工具】	5
2、常见问题.....	6
【消息生产消费异常】	6
[KAFKA-10001] Flume 可以正常连接 Kafka，但是发送消息失败。	6
[KAFKA-10002] Consumer 初始化成功，但是无法从 Kafka 中获取指定 Topic 消息。	9
【启动监控异常】	14
[KAFKA-20001]账号锁定导致启动组件失败.....	14
[KAFKA-20002] Kafka 启动后 Broker 处于 Concerning 状态	15
[KAFKA-20003] Kafka Broker 实例处于 Concerning 状态	19
[KAFKA-20004] Kafka 上报服务不可用告警.....	22
【Topic 操作异常】	24
[KAFKA-30001] 执行 Kafka Topic 删除操作，发现无法删除。	24

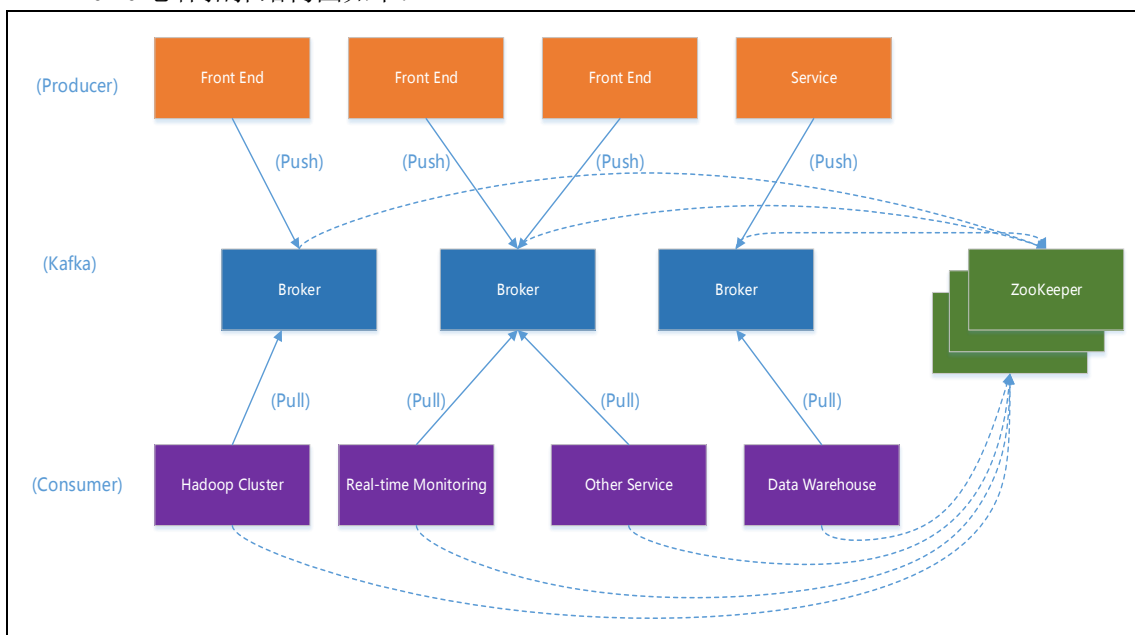
KAFKA

1、基本概念

【概述】

Kafka 是一个高吞吐、分布式、基于发布订阅的消息系统。

Kafka 总体拓扑结构图如下：



一个典型的 Kafka 集群中包含若干 Producer（可以是 web 前端产生的 Page View，或者是服务器日志，系统 CPU、Memory 等），若干 broker（Kafka 支持水平扩展，一般 Broker 数量越多，集群吞吐率越高），若干 Consumer Group，以及一个 Zookeeper 集群。Kafka 通过 Zookeeper 管理集群配置，选举 leader，以及在 Consumer Group 发生变化时进行 rebalance。Producer 使用 push 模式将消息发布到 Broker，Consumer 使用 pull 模式从 Broker 订阅并消费消息。

【日志概述】

日志路径：Kafka 相关日志的默认存储路径为“`/var/log/Bigdata/kafka/broker`”。

日志归档规则：Kafka 的日志启动了自动压缩归档功能，缺省情况下，当日志大小超过 30MB 的时候（此日志文件大小可进行配置，详情请参见“配置日志级别与文件大小”），会自动压缩，压缩后的日志文件名规则为：“<原有日志名>-<yyyy-mm-dd_hh-mm-ss>.[编号].log.zip”。默认最多保留最近的 20 个压缩文件，压缩文件保留个数和压缩文件阈值可以配置。

表 1 Kafka 日志列表		
日志类型	日志文件名	描述
运行日志	server.log	Kafka 服务进程的 server 运行日志。
	controller.log	Kafka 服务进程的 controller 运行日志。
	kafka-request.log	Kafka 服务进程的 request 运行日志。
	log-cleaner.log	Kafka 服务进程的 cleaner 运行日志。
	state-change.log	Kafka 服务进程的 state-change 运行日志。
	kafkaServer-gc.log	Kafka 服务进程的 GC 日志。
	postinstall.log	Kafka 安装后的工作日志。
	prestart.log	Kafka 启动前的工作日志。
	checkService.log	Kafka 服务启动是否成功的检查日志。
	cleanup.log	Kafka 卸载的清理日志。
	start.log	KafkaServer 进程启动日志。
	stop.log	KafkaServer 进程停止日志。
	checkavailable.log	Kafka 服务健康状态检查日志。
	checkInstanceHealth.log	Kafka Broker 健康状态检查日志

日志级别：Kafka 提供了如表 2 所示的日志级别。

运行日志的级别优先级从高到低分别是 OFF、ERROR、WARN、INFO、DEBUG、TRACE，程序会打印高于或等于所设置级别的日志，设置的日志等级越高，打印出来的日志就越少。

表 2 日志级别	
级别	描述
OFF	OFF 表示关闭日志输出。
ERROR	ERROR 表示系统运行的错误信息。
WARN	WARN 表示当前事件处理存在异常信息。
INFO	INFO 表示记录系统及各事件正常运行状态信息。
DEBUG	DEBUG 表示记录系统及系统的调试信息。
TRACE	TRACE 表示记录系统及类调用关系的信息。

如果您需要修改日志级别，请执行如下操作：

1. 登录 FusionInsight Manager。
2. 选择“服务管理”>“Kafka”>“服务配置”。
3. “参数类别”下拉框中选择“全部配置”。
4. 左边菜单栏中选择所需修改的角色所对应的日志菜单。
5. 选择所需修改的日志级别。
6. 单击“保存配置”，然后单击“确定”，保存配置生效。

日志格式：Kafka 的日志格式如下所示

表 3 日志格式		
日志类型	格式	示例
运行日志	<yyyy-MM-dd HH:mm:ss,SSS> <Log Level> <产生该日志的线程名字> <log 中的 message> <日志事件调用类全名>(<日志打印文件>:<行号>)	2015-08-08 11:09:53,483 INFO [main] Loading logs. kafka.log.LogManager (Logging.scala:68)
	<yyyy-MM-dd HH:mm:ss><HostName><组件名><logLevel><Message>	2015-08-08 11:09:51 10-165-0-83 Kafka INFO Running kafka-start.sh.

【客户端工具】

日常定位问题，常常需要使用客户端来连接 Kafka，来进行一些测试验证，从而排除相关可能。

```
total 10124
drwxr-xr-x 5 root root      4096 Feb 29 10:18 HBase
drwxr-xr-x 6 root root      4096 Feb 29 10:18 HDFS
drwxr-xr-x 7 root root      4096 Feb 29 10:19 Hive
drwxr-xr-x 3 root root      4096 Feb 29 10:19 JDK
drwxr-xr-x 4 root root      4096 Feb 29 10:19 Kafka
drwxr-xr-x 4 root root      4096 Feb 29 10:19 KrbClient
drwxr-xr-x 4 root root      4096 Feb 29 10:19 Loader
drwxr-xr-x 4 root root      4096 Feb 29 10:19 SmallFS
drwxr-xr-x 6 root root      4096 Feb 29 10:19 Solr
drwxr-xr-x 5 root root      4096 Feb 29 10:19 Spark
drwxr-xr-x 4 root root      4096 Feb 29 10:20 Yarn
drwxr-xr-x 5 root root      4096 Feb 29 10:20 ZooKeeper
-rwxr-xr-x 1 root root        729 Feb 29 10:20 bigdata_env
-rwxr-xr-x 1 root root      4651 Feb 29 10:20 conf.py
-rwxr-xr-x 1 root root    10281273 Feb 29 10:20 jythonLib.jar
-rwxr-xr-x 1 root root        1799 Feb 29 10:20 switchuser.py
```

注意：下述 IP 地址信息，请根据实际情况进行修改。

List

显示集群中所有的 Topic，Zookeeper 地址是必须的参数

`kafka-topics.sh --list --zookeeper 192.168.234.231:24002/kafka`

Describe

显示 Topic 分区和副本等详细信息，Zookeeper 地址是必须的参数

`kafka-topics.sh --describe --zookeeper 192.168.234.231:24002/kafka`

Producer

通过控制台发送消息到指定 Topic，Zookeeper 地址是必须的参数

```
kafka-console-producer.sh --topic test --broker-list 192.168.234.231:21005
```

Consumer

从 Kafka 指定 Topic 获取信息到控制台，Zookeeper 地址、Topic 名称是必须的参数

```
kafka-console-consumer.sh --topic test --zookeeper 192.168.234.231:24002/kafka --from-beginning
```

Delete

删除集群中指定的 Topic，Zookeeper 地址是必须的参数

```
kafka-topics.sh --delete --topic test --zookeeper 192.168.234.231:24002/kafka
```

2、常见问题

注意：案例中 IP 地址信息，请根据实际情况进行修改。

【消息生产消费异常】

[KAFKA-10001] Flume 可以正常连接 Kafka，但是发送消息失败。

【问题背景与现象】

使用 FusionInsight R2C50 版本安装集群，主要安装 Zookeeper、Flume、Kafka。
在使用 Flume 向 Kafka 发送数据功能时，发现 Flume 发送数据到 Kafka 失败。

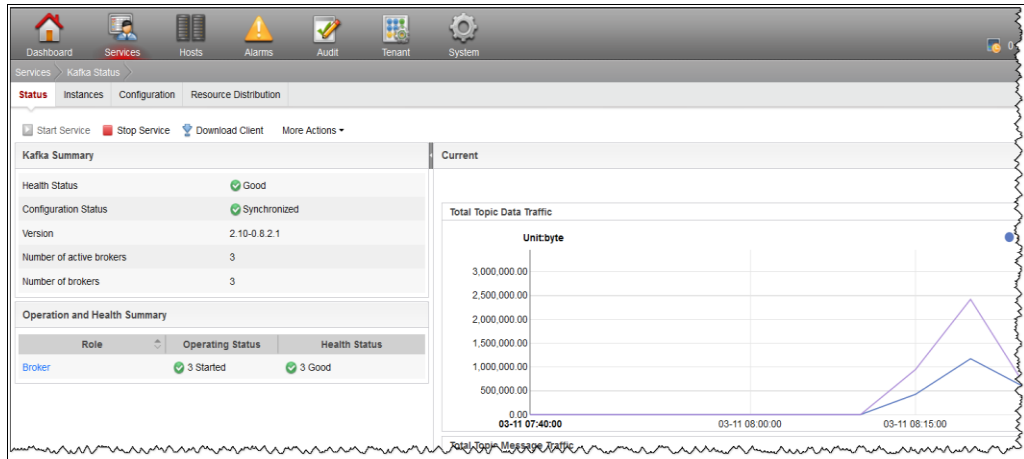
【可能原因】

1. Kafka 服务异常
2. Flume 连接 Kafka 地址错误，导致发送失败
3. Flume 发送超过 Kafka 大小限制的消息，导致发送失败

【原因分析】

Flume 发送数据到 Kafka 失败，可能原因 Flume 侧问题或者 Kafka 侧问题。

1. 通过 FusionInsight Manager 页面，点击 Service->Kafka，查看当前 Kafka 集群当前状态，发现状态为 Good，且监控指标内容显示正确。

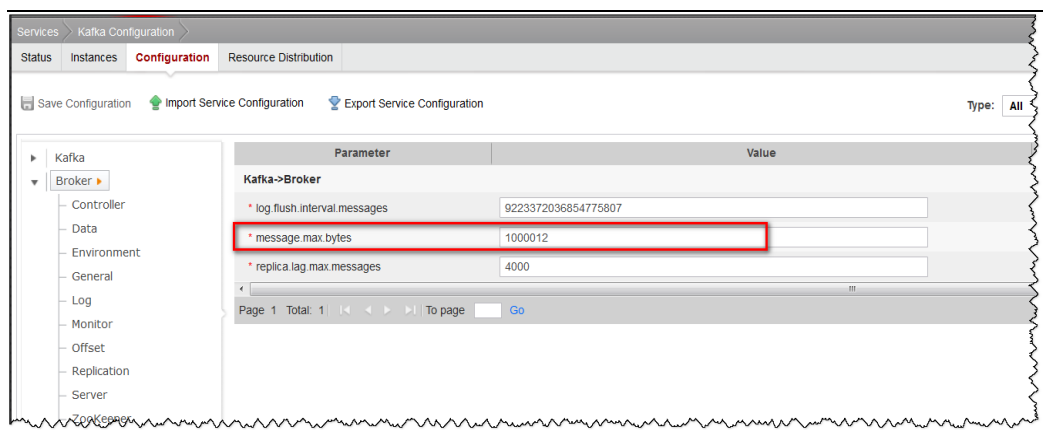


2. 查看 Flume 日志，发现打印 `MessageSizeTooLargeException` 异常信息，如下所示：

```
2016-02-26 14:55:19,126 | WARN | [SinkRunner-PollingRunner-DefaultSinkProcessor] |
Produce request with correlation id 349829 failed due to [LOG,7]:
kafka.common.MessageSizeTooLargeException
kafka.utils.Logging$class.warn(Logging.scala:83)
```

通过异常信息，发现当前 Flume 向 Kafka 写入的数据超过了 Kafka 服务端定义的消息的最大值。

3. 通过 FusionInsight Manager 查看 Kafka 服务端定义的消息的最大值。
选择 `Service->Kafka-Configuration`，进入 Kafka 配置页面，点击右侧 `Type`，选择 `All`，显示所有 Kafka 相关配置，在 `Search` 中输入 `message.max.bytes` 进行检索。

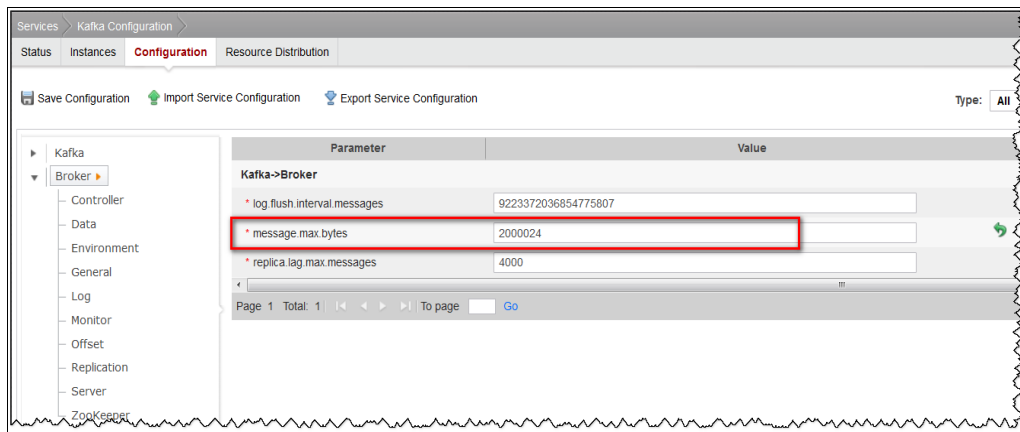


如图所示，FusionInsight 中 Kafka 服务端默认可以接收的消息最大为 `1000012 bytes = 977KB`。

【解决办法】

与客户确认，当前 Flume 发送数据确实存在超过 1M 的消息。因此，为了确保当前这些消息能够写入 Kafka，需要调整 Kafka 服务端相关参数。

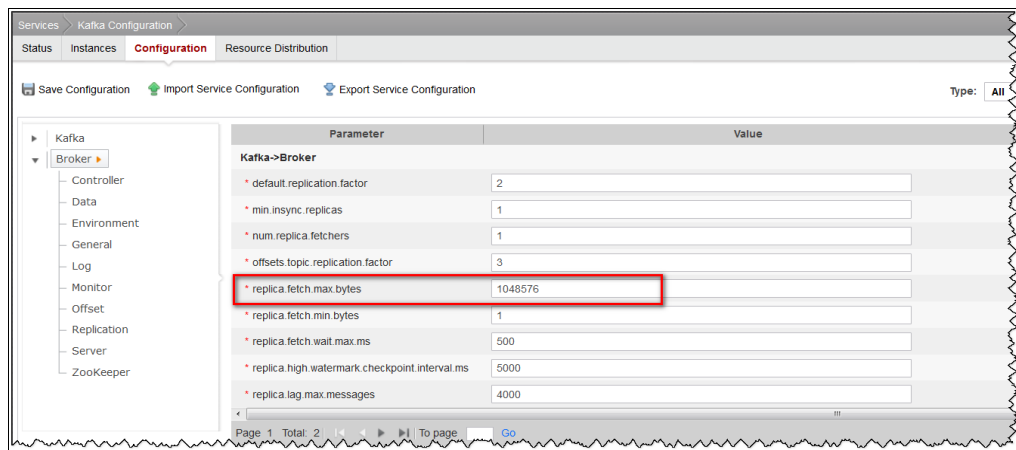
1. 修改 `message.max.bytes`，使得 `message.max.bytes >` 当前业务中消息最大值，使得 Kafka 服务端可以接收全部消息。



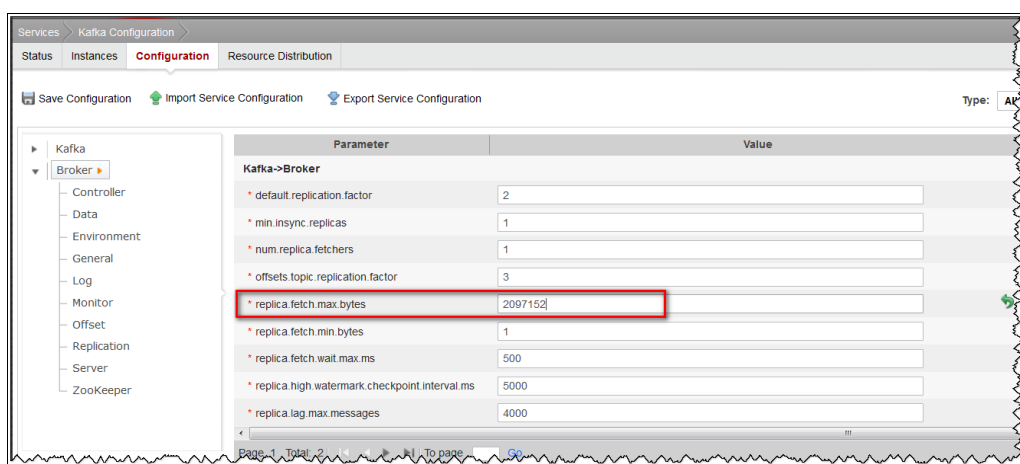
上图为示例，请根据实际情况进行调整。

2. 修改 `replica.fetch.max.bytes`，使得 `replica.fetch.max.bytes >= message.max.bytes`，使得不同 Broker 上的 Partition 的 Replica 可以同步到全部消息。

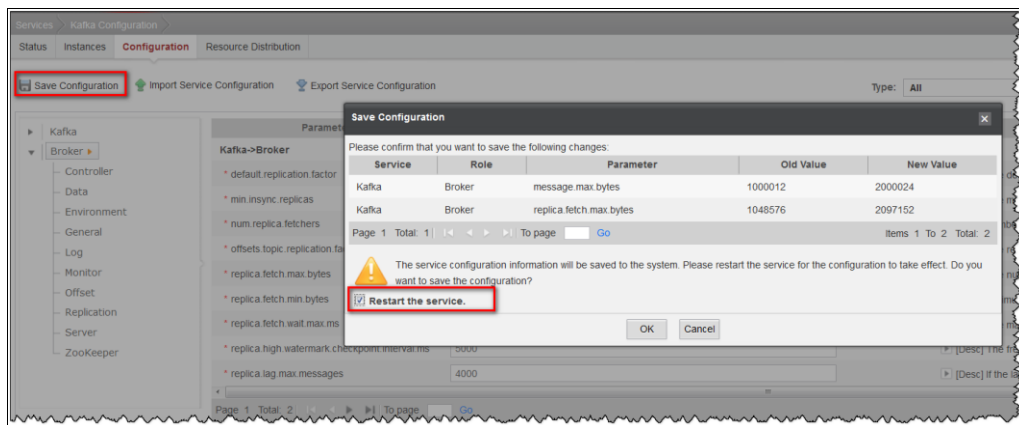
➤ 选择 Service->Kafka-Configuration，进入 Kafka 配置页面，点击右侧 Type，选择 All，显示所有 Kafka 相关配置，在 Search 中输入 `message.max.bytes` 进行检索。



➤ 修改



3. 保存配置，并重启 Kafka 服务，使得 Kafka 相关配置生效。



4. 修改消费者业务应用中 `fetch.message.max.bytes`，使得 `fetch.message.max.bytes >= message.max.bytes`，确保消费者可以消费到全部消息。

[KAFKA-10002] Consumer 初始化成功，但是无法从 Kafka 中获取指定 Topic 消息。

【问题背景与现象】

使用 FusionInsight R2C50 版本安装集群，主要安装 Zookeeper、Flume、Kafka、Storm、Spark。

使用 Storm、Spark、Flume 或者自己编写 consumer 代码来消费 Kafka 中指定 Topic 的消息时，发现消费不到任何数据。

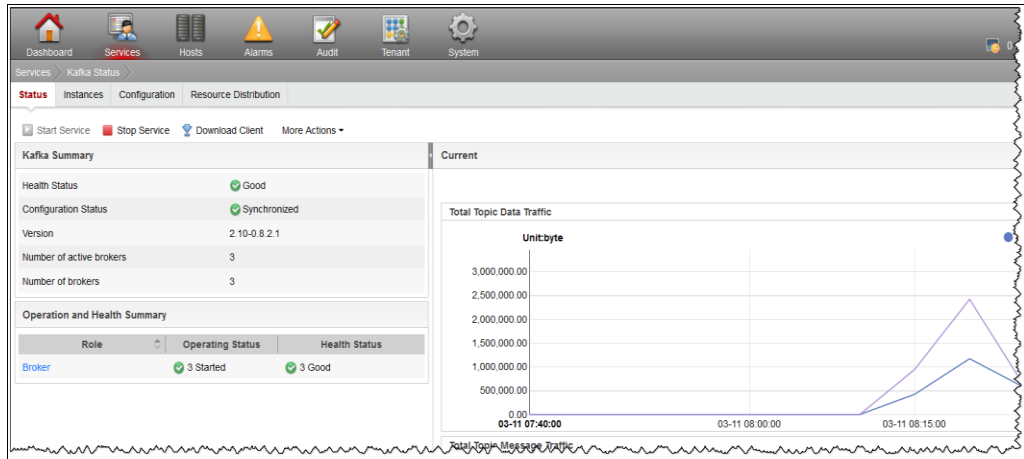
【可能原因】

1. Kafka 服务异常
2. Consumer 中 Zookeeper 相关连接地址配置错误，导致无法消费
3. Consumer 发生 `ConsumerRebalanceFailedException` 异常，导致无法消费
4. Consumer 发生网络导致的 `ClosedChannelException` 异常，导致无法消费

【原因分析】

Storm、Spark、Flume 或者自定义 Consumer 代码可以都称为 Consumer。

1. 通过 FusionInsight Manager 页面，点击 `Service->Kafka`，查看当前 Kafka 集群当前状态，发现状态为 `Good`，且监控指标内容显示正确。



2. 通过 Kafka Client，判断是否可以正常消费数据。

假设客户端已经安装在 /opt/client 目录，test 为需要消费的 Topic 名称，192.168.234.231 为 zookeeper 的 IP 地址

```
cd /opt/client
source bigdata_env
kinit admin

kafka-topics.sh --zookeeper 192.168.234.231:24002/kafka --describe --topic test

kafka-console-consumer.sh --topic test --zookeeper 192.168.234.231:24002/kafka --from-beginning
```

当可以消费到数据时，表示集群服务正常。

3. 查看 Consumer 相关配置，发现 zookeeper 连接地址错误

✧ Flume

```
server.sources.Source02.type=org.apache.flume.source.kafka.KafkaSource
server.sources.Source02.zookeeperConnect=192.168.234.231:24002
server.sources.Source02.topic = test
server.sources.Source02.groupId = test_01
```

✧ Spark

```
val zkQuorum = "192.168.234.231:24002"
```

✧ Storm

```
BrokerHosts brokerHosts = new ZKHosts("192.168.234.231:24002");
```

✧ Consumer API

```
zookeeper.connect="192.168.234.231:24002"
```

FusionInsight Manager 中 Kafka 在 Zookeeper 存储的 ZNode 是以 /kafka 为根路径，有别于开源。Kafka 对应的 Zookeeper 连接配置为 192.168.234.231:24002/kafka Consumer 中配置为 Zookeeper 连接配置为 192.168.234.231:24002，导致无法正确获取 Kafka 中 Topic 相关信息。
解决方法参考 [ZOOKEEPER 连接地址配置错误](#)。

4. 查看 Consumer 相关日志，发现打印 ConsumerRebalanceFailedException 异常

```
2016-02-03 15:55:32,557 | ERROR | [ZkClient-EventThread-75-192.168.234.231:24002/kafka] | Error handling event ZkEvent[New session event sent to kafka.consumer.ZookeeperConsumerConnector$ZKSessionExpireListener@34b41dfe] | org.I0ltec.zkclient.ZkEventThread.run(ZkEventThread.java:77)
kafka.common.ConsumerRebalanceFailedException: pc-zjqbetl86-1454482884879-2ec95ed3 can't rebalance after 4 retries
    at
    kafka.consumer.ZookeeperConsumerConnector$ZKRebalancerListener.syncedRebalance(ZookeeperConsumerConnector.scala:633)
    at
    kafka.consumer.ZookeeperConsumerConnector$ZKSessionExpireListener.handleNewSession(ZookeeperConsumerConnector.scala:487)
    at org.I0ltec.zkclient.ZkClient$4.run(ZkClient.java:472)
    at org.I0ltec.zkclient.ZkEventThread.run(ZkEventThread.java:71)
```

通过异常信息，发现当前 Consumer 没有在指定的重试次数内完成 Rebalance，使得 Consumer 没有被分配 Kafka Topic-Partition，则无法消费消息。
解决方法参考 [REBALANCE 异常](#)

5. 查看 Consumer 相关日志，发现打印 Fetching topic metadata with correlation id 0 for topics [Set(test)] from broker [id:26,host:192-168-234-231,port:21005] failed 错误信息和 ClosedChannelException 异常

```
[2016-03-04 03:33:53,047] INFO Fetching metadata from broker id:26,host: 192-168-234-231,port:21005 with correlation id 0 for 1 topic(s) Set(test) (kafka.client.ClientUtils$)
[2016-03-04 03:33:55,614] INFO Connected to 192-168-234-231:21005 for producing (kafka.producer.SyncProducer)
[2016-03-04 03:33:55,614] INFO Disconnecting from 192-168-234-231:21005 (kafka.producer.SyncProducer)
[2016-03-04 03:33:55,615] WARN Fetching topic metadata with correlation id 0 for topics [Set(test)] from broker [id:26,host: 192-168-234-231,port:21005] failed (kafka.client.ClientUtils$)
java.nio.channels.ClosedChannelException
    at kafka.network.BlockingChannel.send(BlockingChannel.scala:100)
    at kafka.producer.SyncProducer.liftedTree1$1(SyncProducer.scala:73)
    at
    kafka.producer.SyncProducer.kafka$producer$SyncProducer$$doSend(SyncProducer.scala:72)
```

```
at kafka.producer.SyncProducer.send(SyncProducer.scala:113)
at kafka.client.ClientUtils$.fetchTopicMetadata(ClientUtils.scala:58)
at kafka.client.ClientUtils$.fetchTopicMetadata(ClientUtils.scala:93)
at
kafka.consumer.ConsumerFetcherManager$LeaderFinderThread.doWork(ConsumerFetch
erManager.scala:66)
at kafka.utils.ShutdownableThread.run(ShutdownableThread.scala:60)
[2016-03-04 03:33:55,615] INFO Disconnecting from 192-168-234-231:21005
(kafka.producer.SyncProducer)
```

通过异常信息，发现当前 Consumer 无法从 Kafka Broker 192-168-234-231 节点获取元数据，导致无法连接正确的 Broker 获取消息。

6. 可以通过 ping 命令，检查是否可以正常连接 192-168-234-231

```
ping 192-168-234-231
```

Linux 打印如下错误
ping: unknown host 192-168-234-231

Windows 打印如下错误
Ping 请求找不到主机 192-168-234-231。请检查该名称，然后重试。

7. 检查网络是否存在问题，如果网络没有问题，检查是否配置主机和 IP 的对应关系

✧ Linux

```
cat /etc/hosts
```

```
127.0.0.1    localhost
            1
192.168.0.131 192-168-0-131
192.168.0.51  192-168-0-51
192.168.0.122 192-168-0-122
```

✧ Windows

打开 `C:\Windows\System32\drivers\etc\hosts`

```
# For example:
#
#       102.54.94.97    rhino.acme.com          # source server
#       38.25.63.10    x.acme.com             # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1      localhost
#       ::1            localhost
10.82.129.120 rms.huawei.com # modified by IrmTool at 2015-01-18 17:55:13
```

解决方法参考[网络异常](#)

【解决办法】

1. Zookeeper 连接地址配置错误

修改 Consumer 配置中的 Zookeeper 连接地址信息，保证和 FusionInsight 相一致。

✧ Flume

```
server.sources.Source02.type=org.apache.flume.source.kafka.KafkaSource
server.sources.Source02.zookeeperConnect=192.168.234.231:24002/kafka
server.sources.Source02.topic = test
server.sources.Source02.groupId = test_01
```

✧ Spark

```
val zkQuorum = "192.168.234.231:24002/kafka"
```

✧ Storm

```
BrokerHosts brokerHosts = new ZKHosts("192.168.234.231:24002/kafka");
```

✧ Consumer API

```
zookeeper.connect="192.168.234.231:24002/kafka"
```

2. Rebalance 异常

同一个消费者组(consumer group)有多个 consumer 先后启动，就是一个消费者组内有多个 consumer 同时消费多个 partition 数据，consumer 端也会有负载均衡(consumer 个数小于 partitions 数量时)。

consumer 实际上是靠存储在 zk 中的临时节点来表明针对哪个 topic 的那个 partition 拥有读权限的。所在路径为：`/consumers/consumer-group-xxx/owners/topic-xxx/x`。

当触发负载均衡后，原来的 consumer 会重新计算并释放已占用的 partitions，此过程需要一定的处理时间，新来的 consumer 抢占该 partitions 时很有可能会失败。

名称	作用	默认值
rebalance.max.retries	Rebalance 最大重试次数	4
rebalance.backoff.ms	Rebalance 每次重试间隔	2000
zookeeper.session.timeout.ms	Zookeeper 连接会话超时时间	15000

可以适当调大上述三个参数，可以参考如下数值

zookeeper.session.timeout.ms = 45000

rebalance.max.retries = 10

zookeeper.backoff.ms = 5000

参数设置应遵循：

`rebalance.max.retries * rebalance.backoff.ms > zookeeper.session.timeout.ms`

3. 网络异常

在 hosts 文件中没有配置主机名和 IP 的对应关系，导致使用主机名进行访问时，无法获

取信息。

在 `hosts` 文件中添加对应的主机名和 IP 的对应关系。

✧ Linux

```
127.0.0.1      localhost

192.168.0.131  192-168-0-131
192.168.0.51   192-168-0-51
192.168.0.122  192-168-0-122
192.168.234.231 192-168-234-231
```

✧ Windows

```
# For example:
#
#      102.54.94.97      rhino.acme.com      # source server
#      38.25.63.10      x.acme.com        # x client host

# localhost name resolution is handled within DNS itself.
#      127.0.0.1        localhost
#      ::1              localhost
10.82.129.120 rms.huawei.com # modified by IrmTool at 2015-01-18 17:55:13
192.168.234.231 192-168-234-231
```

【启动监控异常】

[KAFKA-20001]账号锁定导致启动组件失败

【问题背景与现象】

新安装集群，启动 Kafka 失败。显示认证失败，导致启动失败。

```
/home/omm/kerberos/bin/kinit -k -t /opt/huawei/Bigdata/etc/2_15_ Broker
/kafka.keytab kafka/hadoop.hadoop.com -c /opt/huawei/Bigdata/etc/2_15_ Broker /11846
failed.
```

```
export key tab file for kafka/hadoop.hadoop.com failed.export and check keytab file
failed, errMsg=]] for Broker #192.168.1.92@192-168-1-92.
```

```
[2015-07-11 02:34:33] RoleInstance started failure for ROLE[name: Broker].
```

```
[2015-07-11 02:34:34] Failed to complete the instances start operation. Current
operation entities: [Broker #192.168.1.92@192-168-1-92], Failure entites : [Broker
```

```
#192.168.1.92@192-168-1-92].Operation Failed.Failed to complete the instances start operation. Current operation entities: [Broker#192.168.1.92@192-168-1-92], Failure entites: [Broker #192.168.1.92@192-168-1-92].
```

【原因分析】

1. 查看 kerberos 日志，/var/log/Bigdata/kerberos/krb5kdc.log，发现有集群外的 IP 使用 kafka 用户连接，导致多次认证失败，导致 hdfs 账号被锁定。

```
Jul 11 02:49:16 192-168-1-91 krb5kdc[1863](info): AS_REQ (2 etypes {18 17})
192.168.1.93: NEEDED_PREAUTH: kafka/hadoop.hadoop.com@HADOOP.COM for
krbtgt/HADOOP.COM@HADOOP.COM, Additional pre-authentication required
Jul 11 02:49:16 192-168-1-91 krb5kdc[1863](info): preauth (encrypted_timestamp) verify
failure: Decrypt integrity check failed
Jul 11 02:49:16 192-168-1-91 krb5kdc[1863](info): AS_REQ (2 etypes {18 17})
192.168.1.93: PREAUTH_FAILED: kafka/hadoop.hadoop.com@HADOOP.COM for
krbtgt/HADOOP.COM@HADOOP.COM, Decrypt integrity check failed
```

【解决办法】

1. 该进入集群外的节点（如 192.168.1.93），断开对 Kafka 的认证。等待 5 分钟，此账号就会被解锁。

[KAFKA-20002] Kafka 启动后 Broker 处于 Concerning 状态

【问题背景与现象】

使用 FusionInsight R2C50 版本安装集群，主要安装 Zookeeper、Kafka。
重新启动 Kafka 后，发现 Broker 实例的状态一直处于 Concerning。

【可能原因】

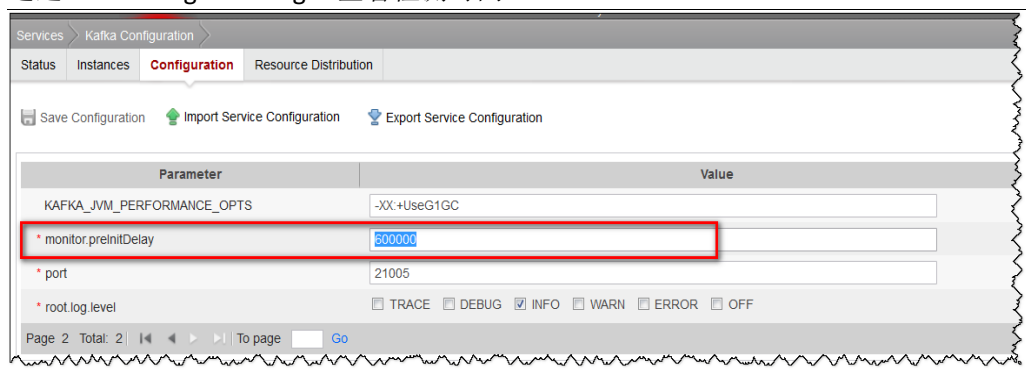
2. Broker 数据目录存储大量数据，导致加载时间过长
3. Broker 数据目录磁盘空间不足，导致加载时异常
4. 其它

【原因分析】

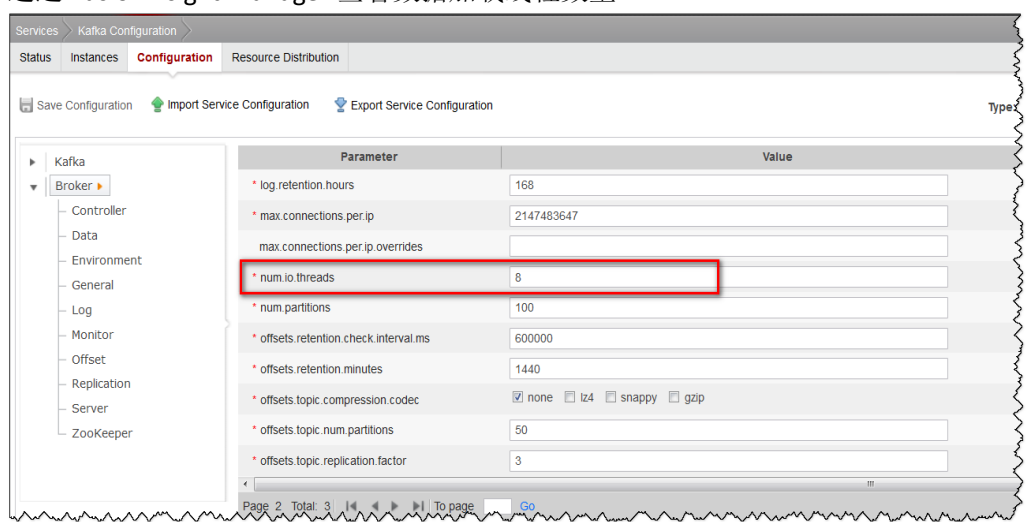
1. Broker 上存放的数据量过大，导致掉 Broker 启动时需要将本地存储的数据进行加载，而加载数据时间过长，Manager 在监控管理该 Broker 时，认为其超过预先设置的合理启动时长，然后又重启了 Broker，导致 Broker 处于 concerning 状态。
2. 通过 `cd /var/log/Bigdata/kafka/broker` 命令进入日志目录查看 server.log，发现如下日志，一直在进行 **load of log**。

```
2016-03-11 11:01:24,543 | INFO | [pool-2-thread-5] | Completed load of log test-46 with
log end offset 0 | kafka.log.Log (Logging.scala:68)
2016-03-11 11:01:24,544 | INFO | [pool-2-thread-4] | Completed load of log test-51 with
log end offset 0 | kafka.log.Log (Logging.scala:68)
```


3. 通过 FusionInsight Manager 查看检测时间



4. 通过 FusionInsight Manager 查看数据加载线程数量



数据加载线程过少，导致加载时间超过检测时间。

解决办法参考 [BROKER 存储大量数据，加载时间较长](#)

5. Broker 启动时需要将本地存储的数据进行加载，在加载时会对文件进行修复，如果此时空间不足会导致修复失败使得 Broker 异常退出，然后又重启了 Broker，导致 Broker 处于 concerning 状态。

6. 通过 `cd /var/log/Bigdata/kafka/broker` 命令进入日志目录

查看 `checkavailable.log`，发现如下日志，打印异常 `No broker exists`。

```
2016-03-23 20:36:54,988 | INFO | [main-EventThread] | zookeeper state changed
(SaslAuthenticated) | org.I0ltec.zkclient.ZkClient (ZkClient.java:449) Check broker count is :
0,[ArrayBuffer()]
2016-03-23 20:36:55,000 | FATAL | [main] | | |
kafka.om.checkBrokerAvailable$ (Logging.scala:113)
kafka.common.KafkaException: No broker exists.
at kafka.om.checkBrokerAvailable$.main(checkBrokerAvailable.scala:57)
at kafka.om.checkBrokerAvailable.main(checkBrokerAvailable.scala)
```

查看 `server.log`，发现如下日志，打印 `during logs loading: java.lang.InternalError: a fault occurred in a recent unsafe memory access operation in compiled Java code`。

```
2016-03-23 20:25:54,563 | ERROR | [main] | There was an error in one of the threads
```

```

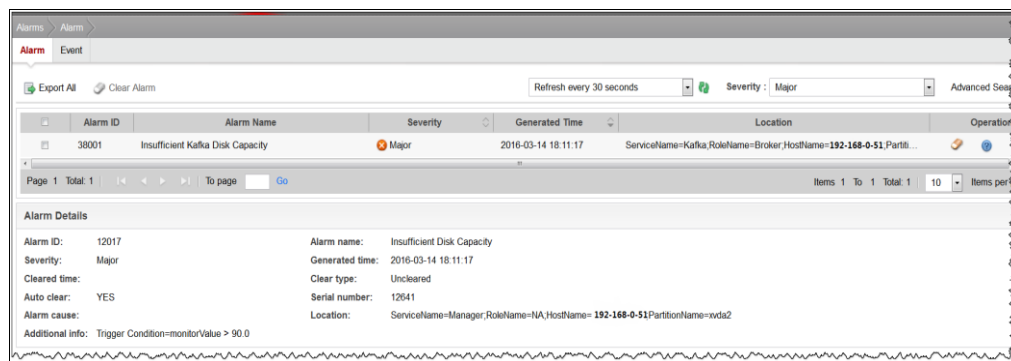
during logs loading: java.lang.InternalError: a fault occurred in a recent unsafe memory
access operation in compiled Java code | kafka.log.LogManager (Logging.scala:97)
2016-03-23 20:25:54,565 | FATAL | [main] | [Kafka Server 3], Fatal error during KafkaServer
startup. Prepare to shutdown | kafka.server.KafkaServer (Logging.scala:116)
java.lang.InternalError: a fault occurred in a recent unsafe memory access operation in
compiled Java code
    at java.nio.HeapByteBuffer.<init>(HeapByteBuffer.java:57)
    at java.nio.ByteBuffer.allocate(ByteBuffer.java:335)
    at kafka.log.FileMessageSet$$anon$1.makeNext(FileMessageSet.scala:188)
    at kafka.log.FileMessageSet$$anon$1.makeNext(FileMessageSet.scala:165)
    at kafka.utils.IteratorTemplate.maybeComputeNext(IteratorTemplate.scala:66)
    at kafka.utils.IteratorTemplate.hasNext(IteratorTemplate.scala:58)
    at kafka.log.LogSegment.recover(LogSegment.scala:175)
    at kafka.log.Log.recoverLog(Log.scala:205)
    at kafka.log.Log.loadSegments(Log.scala:177)
    at kafka.log.Log.<init>(Log.scala:67)
    at
kafka.log.LogManager$$anonfun$loadLogs$2$$anonfun$3$$anonfun$apply$7$$anonfu
n$apply$1.apply$mcV$sp(LogManager.scala:142)
    at kafka.utils.Utills$$anon$1.run(Utills.scala:54)
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
    at java.lang.Thread.run(Thread.java:745)
2016-03-23 20:25:54,567 | INFO | [main] | [Kafka Server 3], shutting down |
kafka.server.KafkaServer (Logging.scala:68)

```

7. 进入数据存储目录。通过 `df-h` 命令查看文件空间，应该保证 Kafka Broker 数据存储目录所在分区磁盘空间小于 80%。

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/xvda1	50G	3.6G	44G	8%	/
devtmpfs	30G	220K	30G	1%	/dev
tmpfs	30G	96K	30G	1%	/dev/shm
/dev/xvda7	79G	40G	36G	54%	/opt
/dev/xvda2	50G	50G	0	100%	/srv/BigData
/dev/xvda3	50G	202M	47G	1%	/tmp
/dev/xvda5	50G	311M	47G	1%	/var
/dev/xvda6	20G	3.0G	16G	16%	/var/log
/dev/xvde1	296G	6.6G	289G	3%	/srv/BigData/dbdata_om
/dev/xvdf1	296G	54G	242G	19%	/srv/BigData/dbdata_service
/dev/xvdg1	296G	271M	295G	1%	/srv/BigData/zookeeper
/dev/xvdh1	296G	193M	296G	1%	/srv/BigData/journalnode
/dev/xvdi1	296G	195M	296G	1%	/srv/BigData/namenode
/dev/xvdj1	296G	87G	210G	30%	/srv/BigData/hadoop/data1
/dev/loop0	3.1G	3.1G	0	100%	/media

查看 FI 告警页面，发现 38001 告警。



发现当前 Kafka Broker 数据存储目录所在分区磁盘空间为 100%，Kafka Broker 无法写入数据。

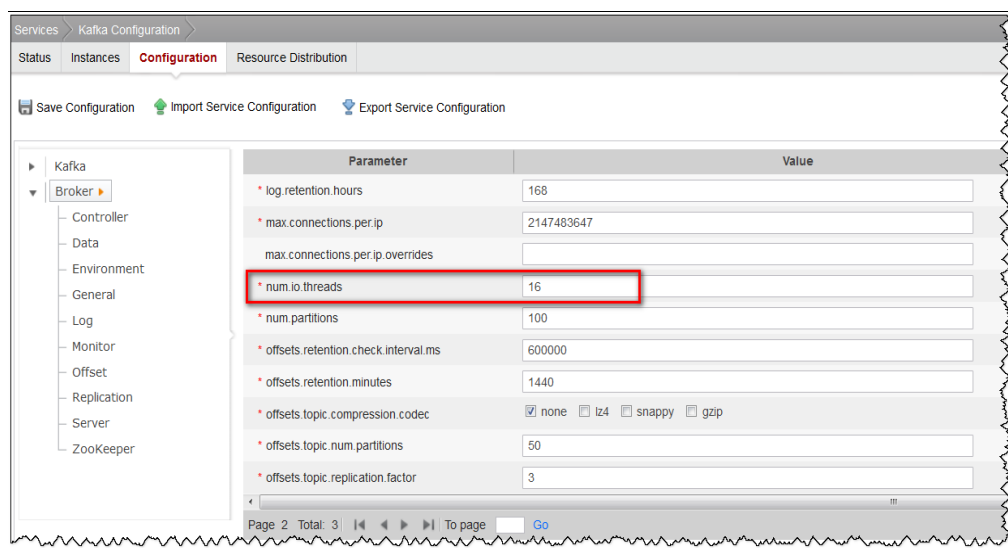
解决办法参考 [BROKER 存储目录磁盘空间不足](#)

8. 如果不是上述原因，则请参考[\[KAFKA-20003\] KAFKA BROKER 实例处于 CONCERNING 状态](#)

【解决办法】

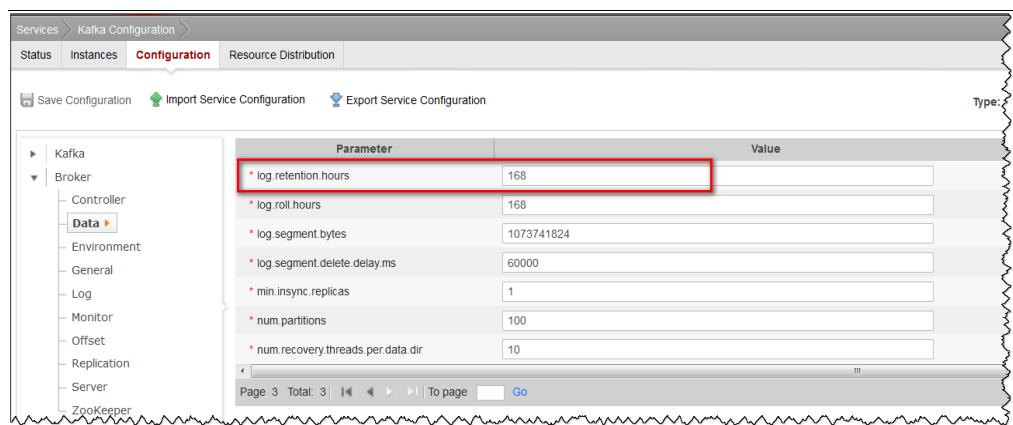
1. Broker 存储大量数据，加载时间较长

通过 FusionInsight Manager 修改数据加载线程数量，该参数表示一个数据目录的加载线程数量。



2. Broker 存储目录磁盘空间不足

先停止该 Broker，对空间不足的数据目录下的数据进行迁移或者删除。
和用户协商，减少数据保存周期和数据保存量。

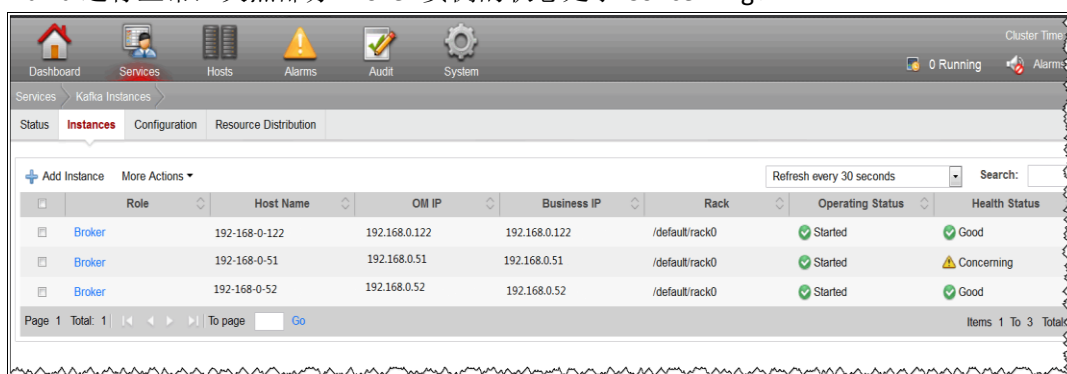


[KAFKA-20003] Kafka Broker 实例处于 Concerning 状态

【问题背景与现象】

使用 FusionInsight R2C50 版本安装集群，主要安装 Zookeeper、Kafka。

Kafka 运行正常，突然部分 Broker 实例的状态处于 Concerning。

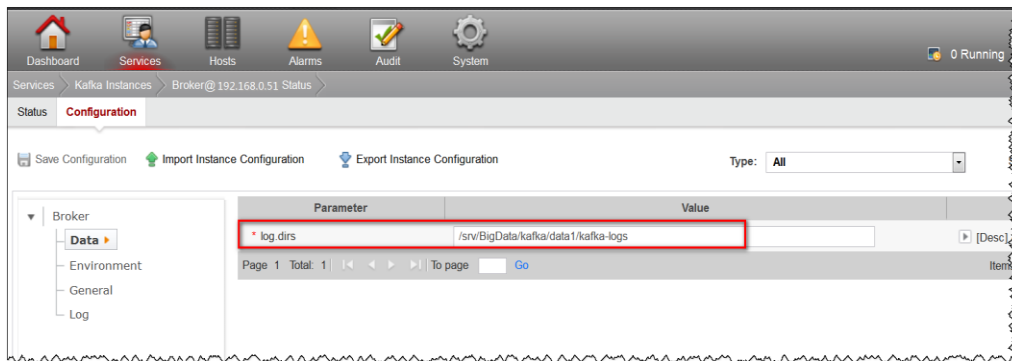


【可能原因】

1. Broker 数据存储空间不足
2. Broker 数据存储目录下文件权限不正确
3. Broker 状态检查发现和 Zookeeper 连接异常

【原因分析】

1. 通过 FusionInsight Manager 页面，点击处于 Concerning 状态的 Broker，进入 Configuration 页面，查看全部配置，获取当前 Kafka Broker 的数据存储目录。



2. 通过 SSH 登录 Kafka Broker，进入数据存储目录。通过 `ll` 命令查看文件权限，数据目录下所有文件目录和文件宿主必须是 `omm:wheel`，且文件目录权限为 `drwx`，文件权限为 `-rw`。

```
total 156
drwx-----. 2 omm wheel 4096 Jan 19 07:39 aaaa-0
drwx-----. 2 omm wheel 4096 Jan 12 11:21 b1-2
drwx-----. 2 omm wheel 4096 Jan 12 11:21 b1-3
drwx-----. 2 omm wheel 4096 Jan 12 11:21 b2-2
drwx-----. 2 omm wheel 4096 Jan 12 11:22 b3-1
drwx-----. 2 omm wheel 4096 Jan 18 20:52 example-metric1-0
drwx-----. 2 omm wheel 4096 Jan 12 11:22 i1-3
drwx-----. 2 omm wheel 4096 Jan 12 11:22 i1-5
drwx-----. 2 omm wheel 4096 Jan 12 11:22 i2-3
drwx-----. 2 omm wheel 4096 Jan 12 11:22 i3-2
drwx-----. 2 omm wheel 4096 Dec 25 10:22 output_timestamp_keyvalue-0
-rw-----. 1 omm wheel 243 Mar 8 08:18 recovery-point-offset-checkpoint
-rw-----. 1 omm wheel 117 Mar 8 08:19 replication-offset-checkpoint
drwx-----. 2 root root 4096 Mar 8 08:18 test
drwx-----. 2 omm wheel 20480 Feb 5 20:57 test-0
drwx-----. 2 omm wheel 4096 Feb 17 08:02 test2-0
```

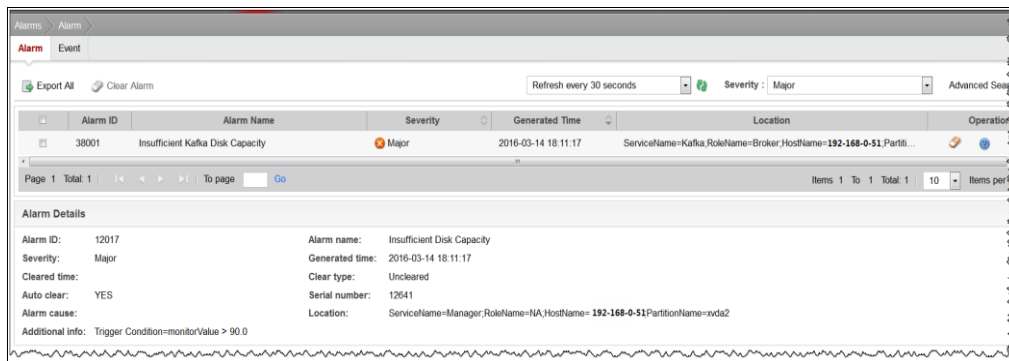
发现 `test` 目录主必须是 `root:root`，且不符合 Kafka 数据目录下文件目录定义规则。应该是人为手工创建。

解决方法参考 [BROKER 数据存储目录下文件权限不正确](#)

3. 通过 SSH 登录 Kafka Broker，进入数据存储目录。通过 `df-h` 命令查看文件空间，应该保证 Kafka Broker 数据存储目录所在分区磁盘空间小于 80%。

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      50G   3.6G   44G    8% /
devtmpfs        30G   220K   30G    1% /dev
tmpfs           30G   96K    30G    1% /dev/shm
/dev/xvda7      79G   40G   36G   54% /opt
/dev/xvda2      50G   50G    0 100% /srv/BigData
/dev/xvda3      50G  202M   47G    1% /tmp
/dev/xvda5      50G  311M   47G    1% /var
/dev/xvda6      20G   3.0G   16G   16% /var/log
/dev/xvde1      296G   6.6G  289G    3% /srv/BigData/dbdata_om
/dev/xvdf1      296G   54G  242G   19% /srv/BigData/dbdata_service
/dev/xvdg1      296G  271M  295G    1% /srv/BigData/zookeeper
/dev/xvdh1      296G  193M  296G    1% /srv/BigData/journalnode
/dev/xvdi1      296G  195M  296G    1% /srv/BigData/namenode
/dev/xvdj1      296G   87G  210G   30% /srv/BigData/hadoop/data1
/dev/loop0      3.1G   3.1G    0 100% /media
```

查看 FI 告警页面，发现 38001 告警。



发现当前 Kafka Broker 数据存储目录所在分区磁盘空间为 100%，Kafka Broker 无法写入数据。

解决办法参考 [BROKER 数据存储空间不足](#)

- 如果文件权限正确，磁盘空间充足，需要检查和 Zookeeper 连接是否异常
通过 `cd /var/log/Bigdata/kafka/broker` 命令进入日志目录
查看 `checkInstanceHealth.log`，发现如下日志，抛出 `ZkTimeoutException` 异常

```
2016-03-08 16:44:13,847 | INFO | [main-EventThread] | EventThread shut down |
org.apache.zookeeper.ClientCnxn (ClientCnxn.java:543)
org.I0ltec.zkclient.exception.ZkTimeoutException: Unable to connect to zookeeper server
within timeout: 15000
2016-03-08 16:44:13 192-168-234-231 Kafka ERROR Execute
kafka.om.checkBrokerInstancHealth failed with errorcode: 1.
2016-03-08 16:44:13 192-168-234-231 Kafka Info The Broker instance health status is bad.
2016-03-08 16:44:13 192-168-234-231 Kafka INFO End kafka-instance-check.sh.
```

Kafka Broker 健康检查脚本连接 Zookeeper 服务超时。

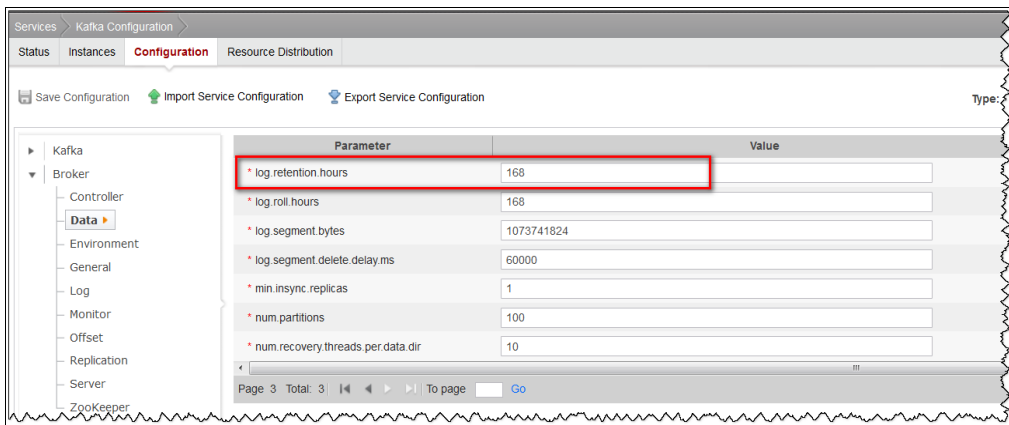
解决办法参考 [BROKER 状态检查发现和 ZOOKEEPER 连接异常](#)

【解决办法】

1. Broker 数据存储空间不足

先停止该 Broker，对空间不足的数据目录下的数据进行迁移或者删除。

和用户协商，减少数据保存周期和数据保存量。



2. Broker 数据存储目录下文件权限不正确

确认文件宿主和权限不正确的文件目录是否有用，或者将其移动到其它目录，或者进行删除。

1) 通过删除文件目录方式来进行处理。**注意：**删除前请确保正确。test 为无效文件

```
rm -rf test
```

2) 通过修改文件目录宿主和权限来进行处理，可能需要使用 root 账号，修改前请确保这个目录属于 Kafka。test-0 为 Kafka 数据

```
chmod omm:wheel -R test-0
```

3. Broker 状态检查发现和 Zookeeper 连接异常

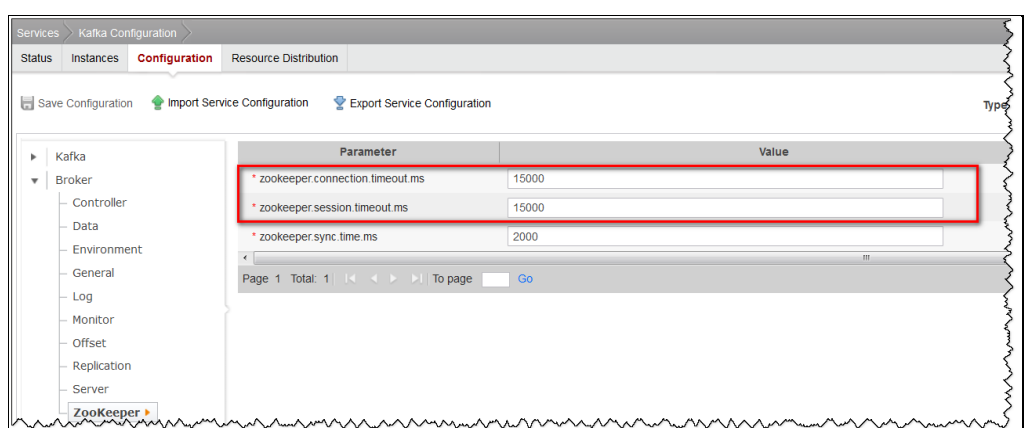
Kafka Broker 健康检查脚本连接 Zookeeper 服务超时。通过 ping 命令，查看是否可以连接 Zookeeper。

```
ping 160.165.234.231
```

如果无法 ping 通，协调网络组进行处理。

```
PING 192.168.234.231 (192.168.234.231) 56(84) bytes of data.  
--- 192.168.234.231 ping statistics ---  
4 packets transmitted, 0 received, 100% packet loss, time 3009ms
```

如果可以 ping 通，且 zookeeper 服务正常，可以适当调整 kafka 连接 zookeeper 的超时时间。



[KAFKA-20004] Kafka 上报服务不可用告警

【问题背景与现象】

使用 FusionInsight R2C50 版本安装集群，主要安装 Zookeeper、Kafka。

Kafka 上报 38000 Kafka Service Unavailable 告警

【可能原因】

1. Kafka 服务异常
2. Kerberos 服务异常
3. Zookeeper 服务异常

【原因分析】

1. 查看 Kafka Broker 实例状态

The screenshot shows the 'Alarm' tab in a management console. It displays a table with columns: Alarm ID, Alarm Name, Severity, Generated Time, Location, and Operation. A single alarm is listed with ID 38000, Name 'Kafka Service Unavailable', and Severity 'Critical'. Below the table, the 'Alarm Details' section provides further information: Alarm ID: 38000, Alarm name: Kafka Service Unavailable, Severity: Critical, Generated time: 2016-03-10 08:04:19, Cleared time: (empty), Clear type: Uncleared, Auto clear: YES, Alarm cause: Some roles are in the Concerning state, Location: ServiceName=Kafka, RoleName=NA, HostName=NA.

Alarm ID	Alarm Name	Severity	Generated Time	Location	Operation
38000	Kafka Service Unavailable	Critical	2016-03-10 08:22:46	ServiceName=Kafka, RoleName=NA, HostName=NA	

Alarm Details

Alarm ID: 38000 Alarm name: Kafka Service Unavailable
Severity: Critical Generated time: 2016-03-10 08:04:19
Cleared time: Clear type: Uncleared
Auto clear: YES Serial number: 10247
Alarm cause: Some roles are in the Concerning state Location: ServiceName=Kafka, RoleName=NA, HostName=NA
Additional info:

The screenshot shows the 'Kafka Instances' tab. It displays a table with columns: Role, Host Name, OM IP, Business IP, Rack, Operating Status, and Health Status. Three Broker instances are listed, all with 'Started' operating status but 'Concerning' health status.

Role	Host Name	OM IP	Business IP	Rack	Operating Status	Health Status
Broker	192-168-0-122	192.168.0.122	192.168.0.122	/default/rack0	Started	Concerning
Broker	192-168-0-51	192.168.0.51	192.168.0.51	/default/rack0	Started	Concerning
Broker	192-168-0-52	192.168.0.52	192.168.0.52	/default/rack0	Started	Concerning

所有的 Broker 实例都处于 Concerning 状态，导致上报 Kafka 服务不可用告警。
解决方法参考 [KAFKA 服务异常](#)

2. 查看 Kafka 依赖的 Kerberos 状态

The screenshot shows a table of various services and their status. Services like Impala, Kafka, KrbServer, and Yarn are marked as 'Bad'.

Service	Status	Health	Details
Impala	Started	Bad	2 StateStore-Catalog 2 HaProxy 3 Impalad
Kafka	Started	Bad	3 Broker
KrbServer	Started	Bad	2 KerberosServer 2 KerberosAdmin
LdapServer	Started	Good	2 SlapdServer (2 Concerning)
Mapreduce	Started	Good	1 JobHistoryServer
Metadata	Started	Good	1 MetadataServer
Solr	Started	Bad	2 SolrServerAdmin (2 Concerning) 1 HBaseIndexer
Spark	Started	Bad	2 JobHistory 2 JDBCServer 3 SparkResource
Yarn	Started	Bad	2 ResourceManager 3 NodeManager
ZooKeeper	Started	Good	3 quorumpeer

The screenshot shows the 'Alarm' tab with a table of alarms. An alarm with ID 32000, Name 'Soir Service Unavailable', and Severity 'Critical' is highlighted. The 'Alarm Details' section below shows: Alarm ID: 32000, Alarm name: Soir Service Unavailable, Severity: Critical, Generated time: 2016-03-11 10:51:14, Cleared time: (empty), Clear type: Uncleared, Auto clear: YES, Alarm cause: KrbServer service unavailable, Location: ServiceName=Soir, RoleName=NA, HostName=NA.

Alarm ID	Alarm Name	Severity	Generated Time	Location
24000	Flume Service Unavailable	Critical	2016-03-11 10:51:29	ServiceName=Flume, RoleName=NA, HostName=NA
38000	Kafka Service Unavailable	Critical	2016-03-11 10:51:27	ServiceName=Kafka, RoleName=NA, HostName=NA
25500	Kerberos Service Unavailable	Critical	2016-03-11 10:51:08	ServiceName=KrbServer, RoleName=NA, HostName=NA
29000	Impala Service Unavailable	Critical	2016-03-11 10:44:58	ServiceName=Impala, RoleName=NA, HostName=NA
16004	Hive Service Unavailable	Critical	2016-03-11 10:44:28	ServiceName=Hive, RoleName=NA, HostName=NA
28001	Spark Service Unavailable	Critical	2016-03-11 10:44:28	ServiceName=Spark, RoleName=NA, HostName=NA

Alarm Details

Alarm ID: 32000 Alarm name: Soir Service Unavailable
Severity: Critical Generated time: 2016-03-11 10:51:14
Cleared time: Clear type: Uncleared
Auto clear: YES Serial number: 11035
Alarm cause: KrbServer service unavailable Location: ServiceName=Soir, RoleName=NA, HostName=NA
Additional info:

解决方法参考 [KRBSERVER 服务异常](#)

3. 查看 Kafka 依赖的 Zookeeper 状态

Hue	✔ Started	✔ Good	✔ Synchronized	2 Hue
Impala	✔ Started	❗ Bad	✔ Synchronized	2 StateStore-Catalog 2 HaProxy 3
Kafka	✔ Started	❗ Bad	✔ Synchronized	3 Broker
KrbServer	✔ Started	✔ Good	✔ Synchronized	2 KerberosServer 2 KerberosAdmin
LdapServer	✔ Started	✔ Good	✔ Synchronized	2 SlapdServer
Mapreduce	✔ Started	✔ Good	✔ Synchronized	1 JobHistoryServer
Metadata	✔ Started	✔ Good	✔ Synchronized	1 MetadataServer
Solr	✔ Started	❗ Bad	✔ Synchronized	1 HBaseIndexer 2 SolrServerAdmin
Spark	✔ Started	❗ Bad	✔ Synchronized	2 JobHistory 2 JDBCServer 3 Spark
Yarn	✔ Started	❗ Bad	✔ Synchronized	2 ResourceManager 3 NodeManager
ZooKeeper	✔ Started	❗ Bad	✔ Synchronized	3 quorumpeer

解决方法参考 [ZOOKEEPER 服务异常](#)

【解决办法】

1. Kafka 服务异常

Broker 实例处于 Concerning 状态，导致上报 kafka 服务不可用，可以参考[\[KAFKA-20003\] KAFKA BROKER 实例处于 CONCERNING 状态](#) 案例进行处理。

2. KrbServer 服务异常

参考 ALM-25500 Kerberos 服务不可用进行处理。

3. Zookeeper 服务异常

参考 ALM-13000 ZooKeeper 服务不可用进行处理。

【Topic 操作异常】

[KAFKA-30001] 执行 Kafka Topic 删除操作，发现无法删除。

【问题背景与现象】

使用 FusionInsight R2C50 版本安装集群，主要安装 Zookeeper、Kafka。

在使用 Kafka 客户端命令删除 Topic 时，发现 Topic 无法被删除。

```
kafka-topics.sh --delete --topic test --zookeeper 192.168.234.231:24002/kafka
```

【可能原因】

1. 客户端命令连接 zookeeper 地址错误
2. Kafka 服务异常 Kafka 部分节点处于停止状态
3. Kafka 服务端配置禁止删除
4. Kafka 配置自动创建，且 Producer 未停止

【原因分析】

1. 客户端命令，打印 **ZkTimeoutException** 异常

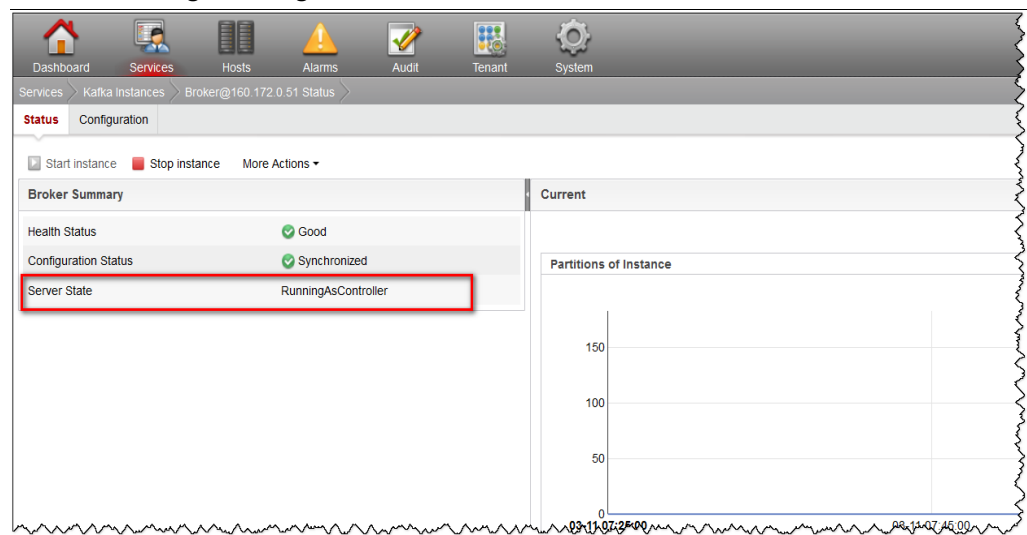
```
[2016-03-09 10:41:45,773] WARN Can not get the principle name from server
192.168.234.231 (org.apache.zookeeper.ClientCnxn)
Exception in thread "main" org.I0Itec.zkclient.exception.ZkTimeoutException: Unable to
connect to zookeeper server within timeout: 30000
    at org.I0Itec.zkclient.ZkClient.connect(ZkClient.java:880)
    at org.I0Itec.zkclient.ZkClient.<init>(ZkClient.java:98)
    at org.I0Itec.zkclient.ZkClient.<init>(ZkClient.java:84)
    at kafka.admin.TopicCommand$.main(TopicCommand.scala:51)
    at kafka.admin.TopicCommand.main(TopicCommand.scala)
```

解决方法参考 [ZOOKEEPER 连接失败导致](#)

2. 客户端查询命令，[kafka-topics.sh --list --zookeeper 192.168.0.122:24002/kafka](#)

test - marked for deletion

通过 FusionInsight Manager 查看 Kafka Broker 实例的服务状态

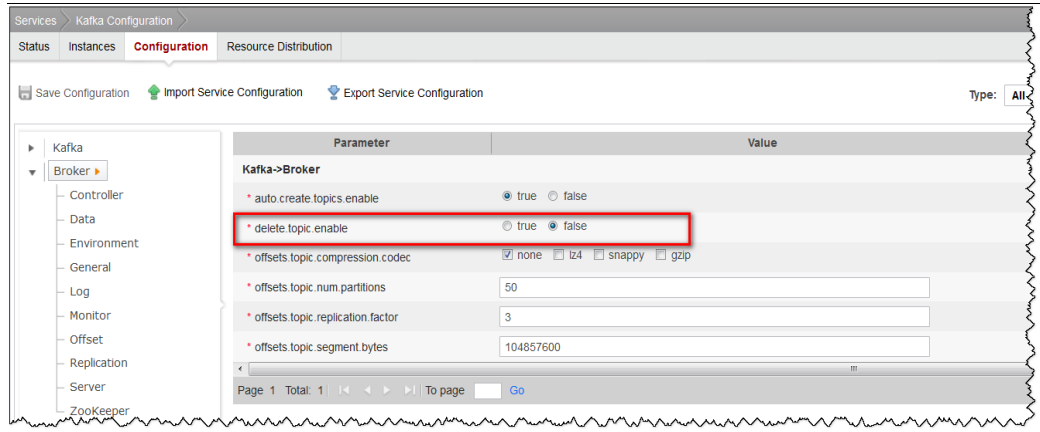


通过 `cd /var/log/Bigdata/kafka/broker` 命令进入 RunningAsController 节点日志目录，在 controller.log 发现 **ineligible for deletion: test**。

2016-03-09 11:11:26,228 | INFO | [main] | [Controller 1]: List of topics to be deleted: |

```
kafka.controller.KafkaController (Logging.scala:68)
2016-03-09 11:11:26,229 | INFO | [main] | [Controller 1]: List of topics ineligible for
deletion: test | kafka.controller.KafkaController (Logging.scala:68)
```

3. 通过 FusionInsight Manager 查询 Broker 删除 Topic 相关配置



解决方法参考 [KAFKA 服务端配置禁止删除](#)

4. 客户端查询命令

```
kafka-topics.sh --describe --topic test --zookeeper 192.168.0.122:24002/kafka
```

```
Topic:test      PartitionCount:10      ReplicationFactor:2      Configs:
Topic: test     Partition: 0           Leader: -1               Replicas: 1,2           Isr:
Topic: test     Partition: 1           Leader: -1               Replicas: 2,3           Isr:
Topic: test     Partition: 2           Leader: -1               Replicas: 3,1           Isr:
Topic: test     Partition: 3           Leader: -1               Replicas: 1,3           Isr:
Topic: test     Partition: 4           Leader: -1               Replicas: 2,1           Isr:
Topic: test     Partition: 5           Leader: -1               Replicas: 3,2           Isr:
Topic: test     Partition: 6           Leader: -1               Replicas: 1,2           Isr:
Topic: test     Partition: 7           Leader: -1               Replicas: 2,3           Isr:
Topic: test     Partition: 8           Leader: -1               Replicas: 3,1           Isr:
Topic: test     Partition: 9           Leader: -1               Replicas: 1,3           Isr:
```

进入 RunningAsController 节点日志目录，在 controller.log 发现 **marked ineligible for deletion**。

```
2016-03-10 11:11:17,989 | INFO | [delete-topics-thread-3] | [delete-topics-thread-3],
Handling deletion for topics test |
kafka.controller.TopicDeletionManager$DeleteTopicsThread (Logging.scala:68)
2016-03-10 11:11:17,990 | INFO | [delete-topics-thread-3] | [delete-topics-thread-3],
Not retrying deletion of topic test at this time since it is marked ineligible for deletion |
kafka.controller.TopicDeletionManager$DeleteTopicsThread (Logging.scala:68)
```

5. 通过 FusionInsight Manager 查询 Broker 状态

Role	Host Name	OM IP	Business IP	Rack	Operating Status	Health Status
Broker	192-168-0-122	192.168.0.122	192.168.0.122	/default/rack0	Stopped	Unknown
Broker	192-168-0-51	192.168.0.51	192.168.0.51	/default/rack0	Started	Good
Broker	192-168-0-52	192.168.0.52	192.168.0.52	/default/rack0	Started	Good

其中一个 Broker 处于停止或者故障状态。Topic 进行删除必须保证该 Topic 的所有 Partition 所在的 Broker 必须处于正常状态。

解决方法参考 [KAFKA 部分节点处于停止或者故障状态](#)

6. 进入 RunningAsController 节点日志目录，在 controller.log 发现 Deletion successfully, 然后又出现 **New topics: [Set(test)]**, 表明被再次创建

```
2016-03-10 11:33:35,208 | INFO | [delete-topics-thread-3] | [delete-topics-thread-3],
Deletion of topic test successfully completed |
kafka.controller.TopicDeletionManager$DeleteTopicsThread (Logging.scala:68)

2016-03-10 11:33:38,501 | INFO | [ZkClient-EventThread-19-
192.168.0.122:24002,160.172.0.52:24002,160.172.0.51:24002/kafka] |
[TopicChangeListener on Controller 3]: New topics: [Set(test)], deleted topics: [Set()], new
partition replica assignment
```

7. 通过 FusionInsight Manager 查询 Broker 创建 Topic 相关配置

Parameter	Value
* auto.create.topics.enable	<input checked="" type="radio"/> true <input type="radio"/> false
* delete.topic.enable	<input type="radio"/> true <input checked="" type="radio"/> false
* offsets.topic.compression.codec	<input checked="" type="checkbox"/> none <input type="checkbox"/> lz4 <input type="checkbox"/> snappy <input type="checkbox"/> gzip
* offsets.topic.num.partitions	50
* offsets.topic.replication.factor	3
* offsets.topic.segment.bytes	104857600

与客户确认，对该 Topic 操作的应用没有停止。

解决方法参考 [KAFKA 配置自动创建，且 PRODUCER 未停止](#)

【解决办法】

1. Zookeeper 连接失败导致

Kafka 客户端连接 Zookeeper 服务超时。通过 ping 命令，查看是否可以连接 Zookeeper。

```
ping -c 4 192.168.234.231
```

```
PING 192.168.234.231 (192.168.234.231) 56(84) bytes of data.  
--- 192.168.234.231 ping statistics ---  
4 packets transmitted, 0 received, 100% packet loss, time 13004ms
```

如果无法 ping 通，通过 FI 界面查看 Zookeeper 服务信息。

Role	Host Name	OM IP	Business IP	Rack	Operating Status	Health Status
quorumpeer	192-168-0-122	192.168.0.122	192.168.0.122	/default/rack0	Started	Good
quorumpeer	192-168-0-51	192.168.0.51	192.168.0.51	/default/rack0	Started	Good
quorumpeer	192-168-0-52	192.168.0.52	192.168.0.52	/default/rack0	Started	Good

配置错误，修改客户端命令中 zookeeper 地址。

2. Kafka 服务端配置禁止删除

修改 delete.topic.enable 为 true。重启服务

Parameter	Value
* auto.create.topics.enable	<input type="radio"/> true <input checked="" type="radio"/> false
* delete.topic.enable	<input checked="" type="radio"/> true <input type="radio"/> false
* offsets.topic.compression.codec	<input checked="" type="checkbox"/> none <input type="checkbox"/> lz4 <input type="checkbox"/> snappy <input type="checkbox"/> gzip
* offsets.topic.num.partitions	50
* offsets.topic.replication.factor	3
* offsets.topic.segment.bytes	104857600

客户端查询命令，无 Topic:test。

```
kafka-topics.sh --list --zookeeper 192.168.0.122:24002/kafka
```

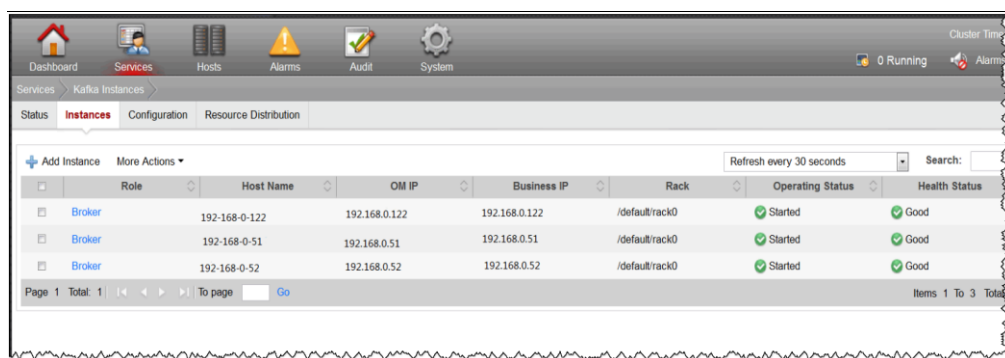
进入 RunningAsController 节点日志目录，在 controller.log 发现 Deletion of topic test successfully

```
2016-03-10 10:39:40,665 | INFO | [delete-topics-thread-3] | [Partition state machine  
on Controller 3]: Invoking state change to OfflinePartition for partitions  
[test,2],[test,15],[test,6],[test,16],[test,12],[test,7],[test,10],[test,13],[test,9],[test,19],[t  
est,3],[test,5],[test,1],[test,0],[test,17],[test,8],[test,4],[test,11],[test,14],[test,18] |  
kafka.controller.PartitionStateMachine (Logging.scala:68)  
2016-03-10 10:39:40,668 | INFO | [delete-topics-thread-3] | [Partition state machine  
on Controller 3]: Invoking state change to NonExistentPartition for partitions
```

```
[test,2],[test,15],[test,6],[test,16],[test,12],[test,7],[test,10],[test,13],[test,9],[test,19],[test,3],[test,5],[test,1],[test,0],[test,17],[test,8],[test,4],[test,11],[test,14],[test,18] |
kafka.controller.PartitionStateMachine (Logging.scala:68)
2016-03-10 10:39:40,977 | INFO | [delete-topics-thread-3] | [delete-topics-thread-3],
Deletion of topic test successfully completed |
kafka.controller.TopicDeletionManager$DeleteTopicsThread (Logging.scala:68)
```

3. Kafka 部分节点处于停止或者故障状态

启动停止的 Broker 实例



客户端查询命令，无 Topic:test。

```
kafka-topics.sh --list --zookeeper 192.168.0.122:24002/kafka
```

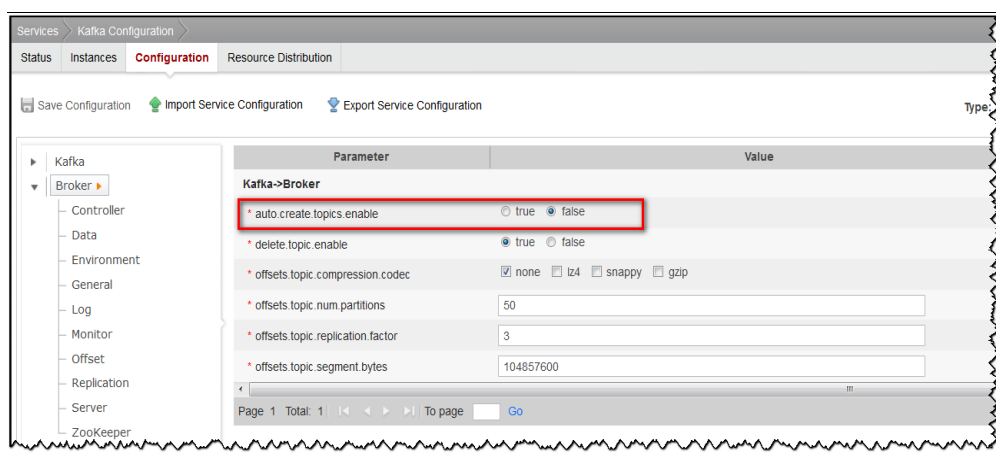
进入 RunningAsController 节点日志目录，在 controller.log 发现 Deletion of topic test successfully

```
2016-03-10 11:17:56,463 | INFO | [delete-topics-thread-3] | [Partition state machine
on Controller 3]: Invoking state change to NonExistentPartition for partitions
[test,4],[test,1],[test,8],[test,2],[test,5],[test,9],[test,7],[test,6],[test,0],[test,3] |
kafka.controller.PartitionStateMachine (Logging.scala:68)
2016-03-10 11:17:56,726 | INFO | [delete-topics-thread-3] | [delete-topics-thread-3],
Deletion of topic test successfully completed |
kafka.controller.TopicDeletionManager$DeleteTopicsThread (Logging.scala:68)
```

4. Kafka 配置自动创建，且 Producer 未停止

停止相关应用。

修改 auto.create.topics.enable 为 false。重启服务。



再次执行 delete 操作