Please visit website: http://cxyroad.com

```

再次执行生产唯一id，看到生成了唯一id

```
[root@xg-200 kafka_2.13-3.1.0]# bin/kafka-storage.sh random-uuid
sQNj3XoUQv6BHx_KCpnx4w
```

使用此id格式化存储目录(每个节点都要执行)

```
bin/kafka-storage.sh format -t sQNj3XoUQv6BHx_KCpnx4w -c
/home/xiangguo/kafka_2.13-3.1.0/config/kraft/server.properties

# 看到格式化完成
Formatting /home/xiangguo/kafka_2.13-3.1.0/datas
```

同理，在另外两个节点执行格式化。

3. 配置kafka环境变量

```
vi /etc/profile
# KAFKA环境变量
export KAFKA_HOME=/home/xiangguo/kafka_2.13-3.1.0
export PATH=$PATH:$KAFKA_HOME/bin
```

使环境变量生效
```

```
source /etc/profile
```

另外两个节点，同理设置环境变量。

4. 创建启动/停止脚本

```
vi /usr/bin/kafka
```

输入内容如下

```
#! /bin/bash

if [ $# -lt 1 ]; then
echo "No Args Input..."
exit
fi

case $1 in
"start") {
for i in 192.168.56.200 192.168.56.202 192.168.56.203; do
echo " --------启动 $i Kafka-------"
ssh $i "source /etc/profile;kafka-server-start.sh -daemon
/home/xiangguo/kafka_2.13-3.1.0/config/kraft/server.properties"
done
} ;;
"stop") {
for i in 192.168.56.200 192.168.56.202 192.168.56.203; do
echo " --------停止 $i Kafka-------"
ssh $i "source /etc/profile;kafka-server-stop.sh"
done
} ;;
*)
echo "Input Args Error..."
;;
esac
```

```

然后添加执行权限

```
chmod +x /usr/bin/kafka
```

启动/停止集群

```
kafka start/stop
```

启动kafka节点之后，过段时间服务会挂掉，如下9092端口都没有了。

```
[root@xg-200 ~]# netstat -anptu | grep 9092
tcp6       0      0 192.168.56.200:9092     :::*                    LISTEN      8211/java
[root@xg-200 ~]# netstat -anptu | grep 9092
```

查看下kafka日志，排查下问题。我们不以demo方式启动以此来查看日志。

```
kafka-server-start.sh /home/xiangguo/kafka_2.13-3.1.0/config/kraft/server.properties
```

果然，我们发现了关键的报错信息。意思是我们的集群id不一致。
```

```
Unexpected error INCONSISTENT_CLUSTER_ID in VOTE response:
InboundResponse

```

我们把各个节点的datas目录下面的meta.properties文件中的cluster.id都统一设置为节点1中的cluster.id。

```
# 节点3的配置
#Wed May 01 14:25:08 CST 2024
cluster.id=sQNj3XoUQv6BHx_KCpnx4w
version=1
node.id=3

```

然后再次启动3个节点，此时我们发现3个节点启动运行正常，也没有中途挂掉的问题。

```
[2024-05-02 09:35:40,712] INFO [SocketServer
listenerType=CONTROLLER, nodeId=1] Stopping socket server request
processors (kafka.network.SocketServer)
[2024-05-02 09:35:40,728] INFO [SocketServer
listenerType=CONTROLLER, nodeId=1] Stopped socket server request
processors (kafka.network.SocketServer)
[2024-05-02 09:35:40,728] INFO [Controller 1]
QuorumController#beginShutdown: shutting down event queue.
(org.apache.kafka.queue.KafkaEventQueue)
[2024-05-02 09:35:40,729] INFO [SocketServer
listenerType=CONTROLLER, nodeId=1] Shutting down socket server
(kafka.network.SocketServer)
[2024-05-02 09:35:40,766] INFO [SocketServer
listenerType=CONTROLLER, nodeId=1] Shutdown completed
(kafka.network.SocketServer)
[2024-05-02 09:35:40,767] INFO [data-plane Kafka Request Handler on
Broker 1], shutting down (kafka.server.KafkaRequestHandlerPool)
[2024-05-02 09:35:40,770] INFO [data-plane Kafka Request Handler on
Broker 1], shut down completely (kafka.server.KafkaRequestHandlerPool)
[2024-05-02 09:35:40,770] INFO [ExpirationReaper-1-AlterAcls]:
Shutting down (kafka.server.DelayedOperationPu     // 发送失败的处理
                log.error("生产者发送消息失败，topic：{}, throwable: {}",
```

```
TOPIC_TEST, throwable.getMessage());
            }

            @Override
            public void onSuccess(SendResult<String, Object>
stringObjectSendResult) {
                // 成功的处理
                log.info("生产者发送消息成功, topic:{},
SendResult:{}",TOPIC_TEST, stringObjectSendResult.toString());
            }
        });
    }
}

```

4. 消息消费

从topic订阅接收、处理消息。

```
@Component
@Slf4j
public class KafkaConsumer {

    /**
     * 从topic订阅接收、处理数据
     * @param record
     * @param ack
     * @param topic
     */
    @KafkaListener(topics = TOPIC_TEST, groupId =
CONSUMER_GROUP)
    public void handle(ConsumerRecord<?, ?> record, Acknowledgment
ack, @Header(KafkaHeaders.RECEIVED_TOPIC) String topic) {
        Optional message = Optional.ofNullable(record.value());
        if (message.isPresent()) {
            Object msg = message.get();
            log.info("消费者组：{} 从topic:{} 订阅接收、处理消息:{}",
CONSUMER_GROUP, topic, msg);
            ack.acknowledge();
        }
    }

}
```

```

5. 发布订阅验证

生产者发送测试数据

```
@SpringBootTest
@RunWith(SpringRunner.class)
@Slf4j
public class AppTest {

    @Resource
    private KafkaProducer kafkaProducer;

    /**
     * kafka生产者发送消息
     */
    @Test
    public void sendMsg() throws InterruptedException {
        for(int i = 0; i<10; i++) {
            kafkaProducer.send("测试数据: " + i);
            Thread.sleep(1000);
        }

    }
}

```

消费到数据。

```
2024-05-03 10:34:03.034 INFO 35416 --- [ntainer#0-0-C-1] o.e.r.d.c.service.KafkaConsumer   : 消费者组：consumer.group 从 topic:topic.test 订阅接收、处理消息:测试数据: 0
2024-05-03 10:34:04.024 INFO 35416 --- [ntainer#0-0-C-1] o.e.r.d.c.service.KafkaConsumer   : 消费者组：consumer.group 从 topic:topic.test 订阅接收、处理消息:测试数据: 1
2024-05-03 10:34:05.030 INFO 35416 --- [ntainer#0-0-C-1] o.e.r.d.c.service.KafkaConsumer   : 消费者组：consumer.group 从
```
```

```
topic:topic.test 订阅接收、处理消息:测试数据: 2
2024-05-03 10:34:06.024  INFO 35416 --- [ntainer#0-0-C-1]
o.e.r.d.c.service.KafkaConsumer          : 消费者组：consumer.group 从
topic:topic.test 订阅接收、处理消息:测试数据: 3
2024-05-03 10:34:07.029  INFO 35416 --- [ntainer#0-0-C-1]
o.e.r.d.c.service.KafkaConsumer          : 消费者组：consumer.group 从
topic:topic.test 订阅接收、处理消息:测试数据: 4
2024-05-03 10:34:08.025  INFO 35416 --- [ntainer#0-0-C-1]
o.e.r.d.c.service.KafkaConsumer          : 消费者组：consumer.group 从
topic:topic.test 订阅接收、处理消息:测试数据: 5
2024-05-03 10:34:09.029  INFO 35416 --- [ntainer#0-0-C-1]
o.e.r.d.c.service.KafkaConsumer          : 消费者组：consumer.group 从
topic:topic.test 订阅接收、处理消息:测试数据: 6
2024-05-03 10:34:10.033  INFO 35416 --- [ntainer#0-0-C-1]
o.e.r.d.c.service.KafkaConsumer          : 消费者组：consumer.group 从
topic:topic.test 订阅接收、处理消息:测试数据: 7
2024-05-03 10:34:11.030  INFO 35416 --- [ntainer#0-0-C-1]
o.e.r.d.c.service.KafkaConsumer          : 消费者组：consumer.group 从
topic:topic.test 订阅接收、处理消息:测试数据: 8
2024-05-03 10:34:12.033  INFO 35416 --- [ntainer#0-0-C-1]
o.e.r.d.c.service.KafkaConsumer          : 消费者组：consumer.group 从
topic:topic.test 订阅接收、处理消息:测试数据: 9

```
原文链接: https://juejin.cn/post/7363828811363221555