1. *简介：rocketmq是一款分布式、队列模型的消息中间件，具有以下特点：能够保证严格的消息顺序、提供丰富的消息拉取模式、高效的订阅者水平扩展能力、实时的消息订阅机制、亿级消息堆积能力，rocketmq不遵循任何规范，设计初衷是为电商设计的*
2. *常用名词：producer() consumer() puch consumer() pull consumer() producer group()*

*Consumer group() broker(消息中转角色，负责存储消息和发送消息) 广播消费；rq中只有一种模式，就是发布--订阅机制，不存在点对点消费；集群消费 顺序消息（保证严格的顺序消息），持久化方式：基于文件的持久化，可以持久化到mysql等*

1. *rocketmq的环境搭建：如果不是并发量不是太高的话，不要使用多master多slave模式(分为异步复制和同步双写，一般用异步复制，效率好，同步双写在服务器宕机的情况下不会有消息丢失的情况，如果关于钱的话，建议使用同步双写)，使用多master就可以了。*
2. *双master环境搭建：一、修改hosts文件，能ping的通就可以,一般nameserver和master放在一个同一个服务器上就可以了，保持心跳的一致性。Rocketmq的堆内存最小为1个g*
3. *消息消费是非顺序的，也可以设置为顺序消费*
4. *Consumer端的消息重试机制：timeout，如果消息没有推送到消费者端，则尝试无限次发送，当消息重复消费次数达到设定值时，进行业务处理，记录日志或者是持久化到数据库中*
5. *Producer端的消息重试机制：超时未发送，则尝试重新发送*
6. *生产环境中，一般都是先启动消费者端，即先进行订阅，然后再启动生产者端*
7. *Rocketmq天然支持生成者、消费者负载均衡*
8. *如果先开启生产者后启动消费者，那么有可能产生消息重复消费的现象。因为消息处理之后会给mq返回信息，mq判断消费者是否已经消费了此消息。*
9. *Rocketmq是没有办法避免重复消费的，所以必须在处理业务的时候进行去重。或者是记录日志或者是主键设置数据库消息插不进去。*
10. *消费端默认为集群消费，支持负载均衡，还可以设置为广播机制，即每一个客户端都要进行处理消息。*
11. *Rocketmq的九个模块：rocketmq-filter（需要手动的开启这个服务）、rocketmq-common、rocketmq-client、 rocketmq-store、 rocketmq-srvutil、rocketmq-namesrv（需要手动开启的服务）、rocketmq-remoting、rocketmq-tools、rocketmq-broker（需要手动开启的服务）*
12. *Rocketmq API：重要概念：groupName的定义*
13. *Rocketmq支持顺序消费，每个topic底下默认总共有四个队列，可以根据这些队列实现顺序消费。*
14. *Rocketmq实现分布式消息事务，如：不同的银行的转账，保证数据的一致性，典型的分布式事务问题。分布式，即分布式系统之间接口调用的成功和失败的一致性。如果都在一个系统中，则不会有这样的问题。*
15. *Rocketmq处理分布式事务的方式：首先发送一个预处理消息，然后提交本地事务，然后发送确认消息，确认消息发送成功后，消息对于消费者端可见，这样就可以保证系统事务的一致性。*