

课程介绍

- MongoDB集群之复制集
- MongoDB集群之分片集群
- 日志规范
- 异常规范
- 其它规范

1、MongoDB集群之复制集

1.1、简介

一组MongoDB复制集，就是一组mongod进程，这些进程维护同一个数据集合。复制集提供了数据冗余和高等级的可靠性，这是生产部署的基础。

目的

- 保证数据在生产部署时的冗余和可靠性，通过在不同的机器上保存副本来保证数据的不会因为单点损坏而丢失。能够随时应对数据丢失、机器损坏带来的风险。
- 还能提高读取能力，用户的读取服务器和写入服务器在不同的地方，而且，由不同的服务器为不同的用户提供服务，提高整个系统的负载。

机制

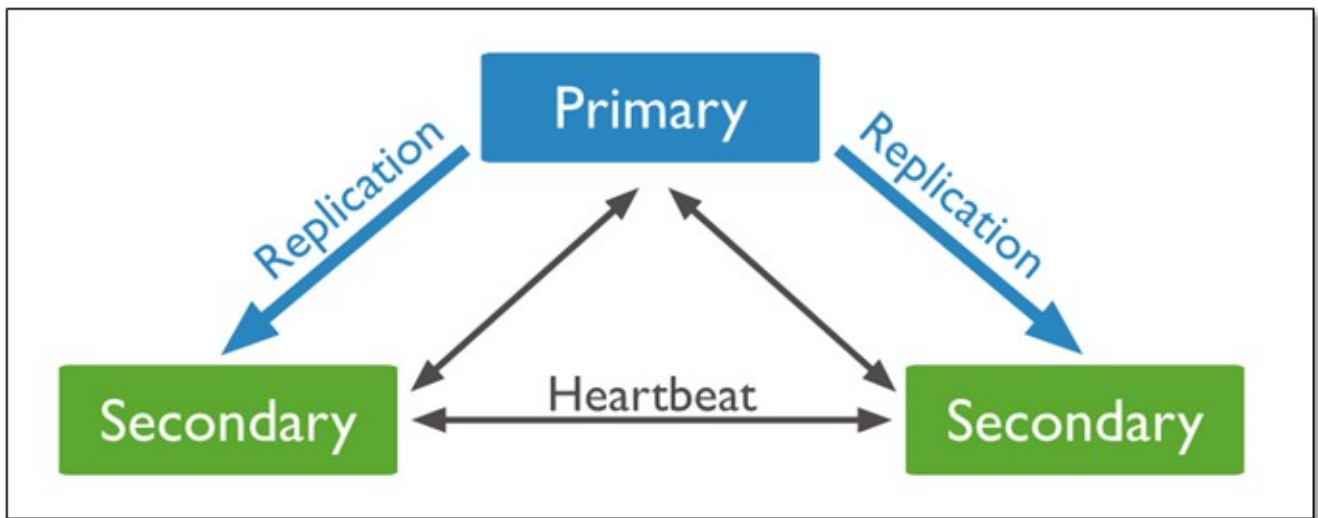
- 一组复制集就是一组mongod实例掌管同一个数据集合，实例可以在不同的机器上面。实例中包含一个主导（Primary），接受客户端所有的写入操作，其他都是副本实例（Secondary），从主服务器上获得数据并保持同步。
主服务器很重要，包含了所有的改变操作（写）的日志。但是副本服务器集群包含有所有的主服务器数据，因此当主服务器挂掉了，就会在副本服务器上重新选取一个成为主服务器。
- 每个复制集还有一个仲裁者（Arbiter），仲裁者不存储数据，只是负责通过心跳包来确认集群中集合的数量，并在主服务器选举的时候作为仲裁决定结果。

1.2、架构

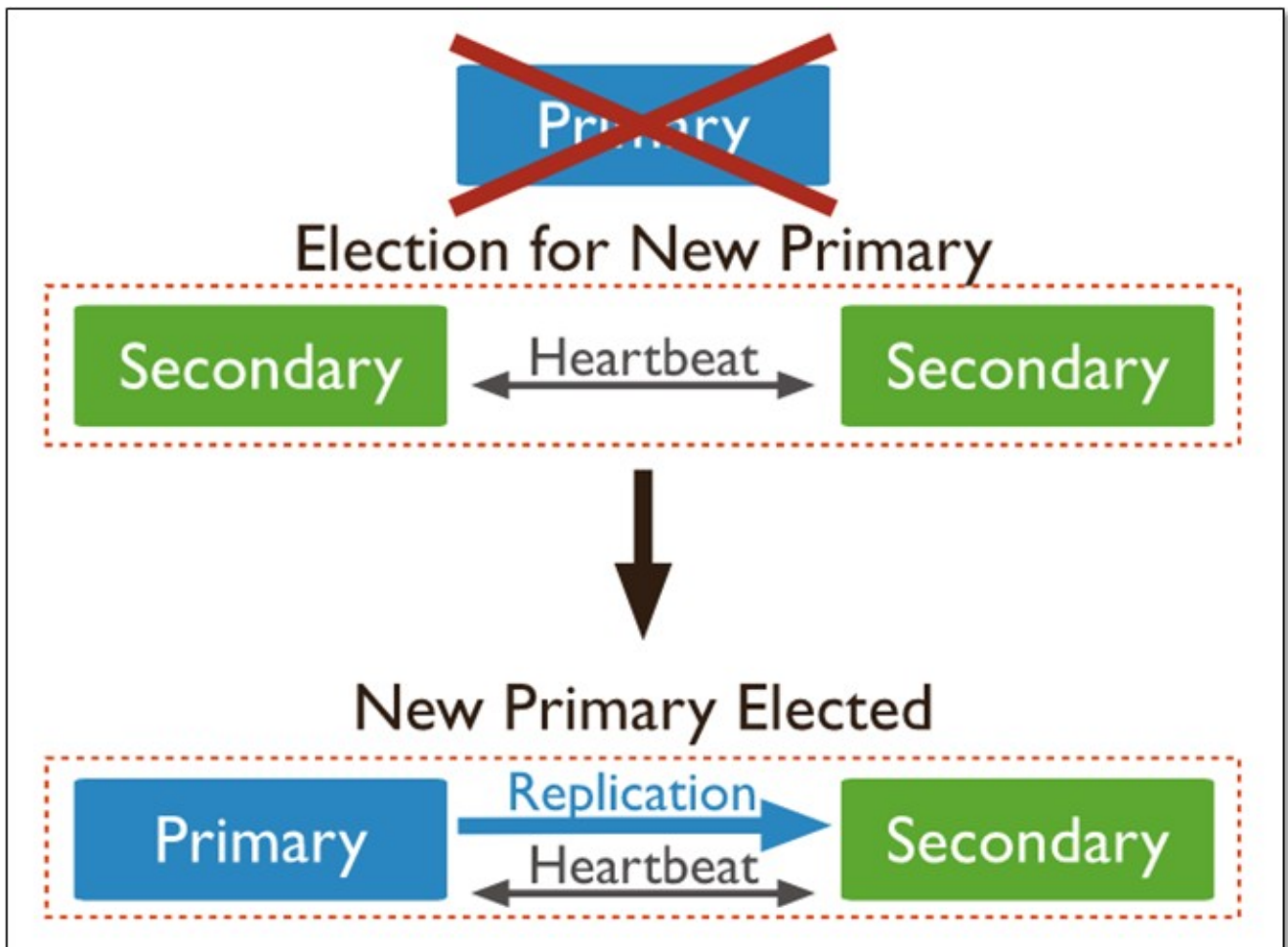
基本的架构由3台服务器组成，一个三成员的复制集，由三个有数据，或者两个有数据，一个作为仲裁者。

1.2.1、三个存储数据的复制集

一个主，两个从库组成，主库宕机时，这两个从库都可以被选为主库。

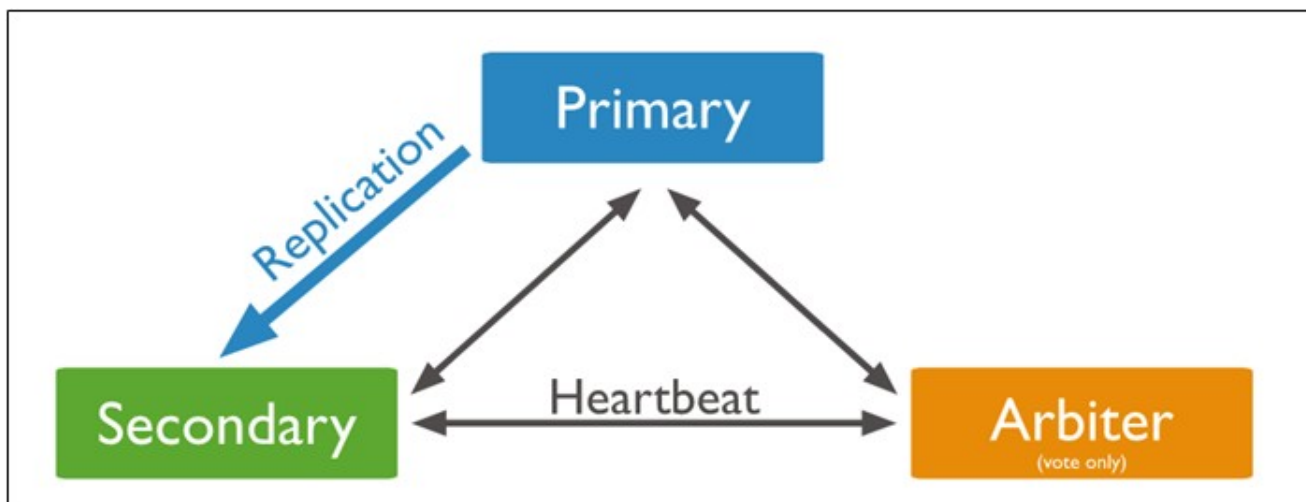


当主库宕机后,两个从库都会进行竞选, 其中一个变为主库, 当原主库恢复后, 作为从库加入当前的复制集群即可。



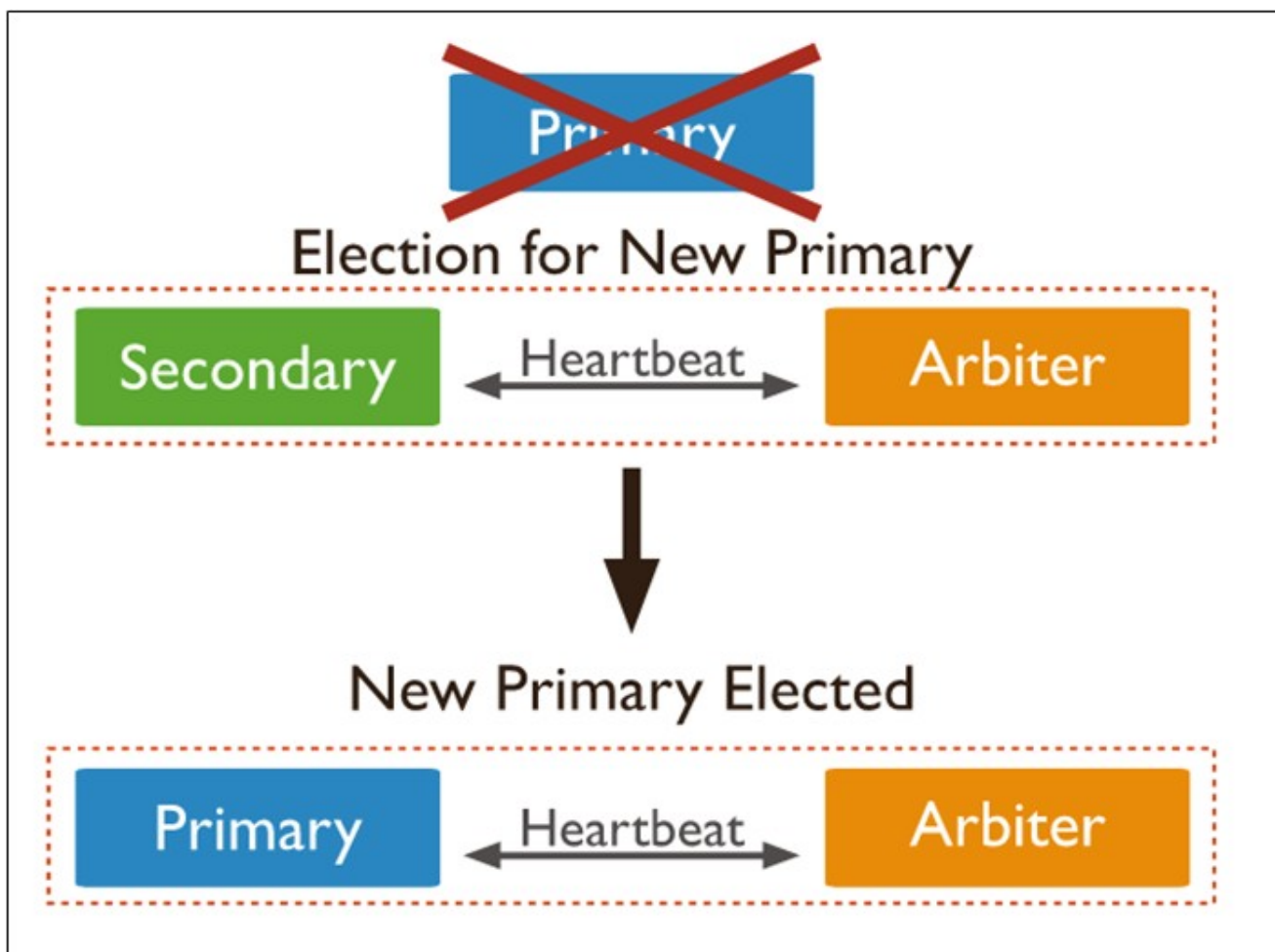
1.2.2、存在arbiter节点的复制集

一个主库, 一个从库, 可以在选举中成为主库, 一个arbiter节点, 在选举中, 只进行投票, 不能成为主库。



说明：由于arbiter节点没有复制数据，因此这个架构中仅提供一个完整的数据副本。arbiter节点只需要更少的资源，代价是更有限的冗余和容错。

当主库宕机时，将会选择从库成为主，主库修复后，将其加入到现有的复制集群中即可。



1.3、Primary选举

复制集通过replSetInitiate命令（或mongo shell的rs.initiate()）进行初始化，初始化后各个成员间开始发送心跳消息，并发起Primary选举操作，获得『大多数』成员投票支持的节点，会成为Primary，其余节点成为Secondary。

『大多数』的定义

假设复制集内投票成员数量为N，则大多数为 $N/2 + 1$ ，当复制集内存活成员数量不足大多数时，整个复制集将无法选举出Primary，复制集将无法提供写服务，处于只读状态。

1.4、成员说明

成员	说明
Primary	Primary的作用是接收用户的写入操作，将自己的数据同步给其他的Secondary。
Secondary	正常情况下，复制集的Secondary会参与Primary选举（自身也可能被选为Primary），并从Primary同步最新写入的数据，以保证与Primary存储相同的数据。Secondary可以提供读服务，增加Secondary节点可以提供复制集的读服务能力，同时提升复制集的可用性。另外，Mongodb支持对复制集的Secondary节点进行灵活的配置，以适应多种场景的需求。
Arbiter	Arbiter节点只参与投票，不能被选为Primary，并且不从Primary同步数据。比如你部署了一个2个节点的复制集，1个Primary，1个Secondary，任一节点宕机，复制集将不能提供服务了（无法选出Primary），这时可以给复制集添加一个Arbiter节点，即使有节点宕机，仍能选出Primary。Arbiter本身不存储数据，是非常轻量级的服务，当复制集成员为偶数时，最好加入一个Arbiter节点，以提升复制集可用性。
Priority0	Priority0节点的选举优先级为0，不会被选举为Primary。比如你跨机房A、B部署了一个复制集，并且想指定Primary必须在A机房，这时可以将B机房的复制集成员Priority设置为0，这样Primary就一定会是A机房的成员。（注意：如果这样部署，最好将『大多数』节点部署在A机房，否则网络分区时可能无法选出Primary）
Vote0	Mongodb 3.0里，复制集成员最多50个，参与Primary选举投票的成员最多7个，其他成员（Vote0）的vote属性必须设置为0，即不参与投票。
Hidden	Hidden节点不能被选为主（Priority为0），并且对Driver不可见。因Hidden节点不会接受Driver的请求，可使用Hidden节点做一些数据备份、离线计算的任务，不会影响复制集的服务。
Delayed	Delayed节点必须是Hidden节点，并且其数据落后与Primary一段时间（可配置，比如1个小时）。因Delayed节点的数据比Primary落后一段时间，当错误或者无效的数据写入Primary时，可通过Delayed节点的数据来恢复到之前的时间点。

1.5、搭建复制集

```
1 #创建3个mongo容器
2 docker create --name mongo01 -p 27017:27017 -v mongo-data-01:/data/db mongo:4.0.3 --
  replset "rs0" --bind_ip_all
3 docker create --name mongo02 -p 27018:27017 -v mongo-data-02:/data/db mongo:4.0.3 --
  replset "rs0" --bind_ip_all
4 docker create --name mongo03 -p 27019:27017 -v mongo-data-03:/data/db mongo:4.0.3 --
  replset "rs0" --bind_ip_all
5
6 #启动容器
7 docker start mongo01 mongo02 mongo03
8
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/775313220131011232>