# mongoDB查询命令

### 库大小

use DBdb.stats();

### 数据大小

db.collection.dataSize()

### 占用磁盘大小

db.collection.storageSize()

### 查看oplog的状态

**sharding架构下，mongos不能查看oplog，但是我们可以到每个mongod实例去看**

包含记录条数和时间范围，可以使用"rs.printReplicationInfo() "命令

shardsvr1:PRIMARY> rs.printReplicationInfo()  
configured oplog size: 2450.97412109375MB // oplog大小是2450MB  
log length start to end: 53841secs (14.96hrs) // 第一条和最后一条日志的时间差是14.96小时  
oplog first event time: Wed Oct 12 2022 20:46:49 GMT+0800 (CST)  
oplog last event time: Thu Oct 13 2022 11:44:10 GMT+0800 (CST)  
now: Thu Oct 13 2022 11:44:20 GMT+0800 (CST)  
shardsvr1:PRIMARY>

*字段说明：*

* configured oplog size： oplog文件大小

* log length start to end: oplog日志的启用时间段

* oplog first event time: 第一个事务日志的产生时间

* oplog last event time: 最后一个事务日志的产生时间

* now: 现在的时间

db.getReplicationInfo() 可以用来查看oplog的状态、大小、存储的时间范围

shardsvr1:PRIMARY> db.getReplicationInfo()   
{  
 "logSizeMB" : 2450.97412109375,  
 "usedMB" : 5.64,  
 "timeDiff" : 61432,  
 "timeDiffHours" : 17.06,  
 "tFirst" : "Wed Oct 12 2022 20:46:49 GMT+0800 (CST)",  
 "tLast" : "Thu Oct 13 2022 13:50:41 GMT+0800 (CST)",  
 "now" : "Thu Oct 13 2022 13:50:43 GMT+0800 (CST)"  
}

### 查看slave状态

通过"db.printSlaveReplicationInfo()"可以查看slave的同步状态

副本节点中执行db.printSlaveReplicationInfo()命令可以查看同步状态信息

* source——从库的IP及端口

* syncedTo——当前的同步情况，延迟了多久等信息

shardsvr1:PRIMARY> db.printSecondaryReplicationInfo()  
source: 192.168.31.52:17002  
 syncedTo: Thu Oct 13 2022 13:57:21 GMT+0800 (CST)  
 0 secs (0 hrs) behind the primary

### 其他命令参考

################################### 注意 #################################################  
# 在分片的主节点也可以同样步骤创建用户并验证，然后可以执行其他命令。  
# 之后，可以在分片副本上用上一步创建的用户并验证，但是show dbs 会出现错误uncaught exception: Error: listDatabases failed:  
## 当前从节点只是一个备份，不是奴隶节点，无法读取数据，写当然更不行。因为默认情况下，从节点是没有读写权限的，可以增加（删除）读的权限，rs.secondaryOk()/rs.secondaryOk(false)。  
# 仲裁节点不会放任何业务数据，可以登陆查看.仲裁节点不能进行数据的查看。  
  
  
#### https://www.cnblogs.com/cy0628/p/15175428.html  
# 查看副本集的配置：返回包含当前副本集配置的文档。rs.conf()/rs.config()是该方法的别名。  
# 查看副本集的状态：返回包含状态信息的文档，次输出结果从副本集的其他成员发送的心跳包中获得的数据反映副本集的当前状态。rs.status()  
# 添加副本从节点：在主节点添加从节点，将其他成员加入到副本集。rs.add(host, arbiterOnly)  
# 查看全局所有账户：db.system.users.find().pretty()  
# 查看当前库下的账户：use test; show users;  
  
  
  
#数据库 命令  
db.serverStatus().connections; //连接数查看  
show collections //查看表信息  
db.test\_shard.find().count() //查看table1数据长度  
db.test\_shard.remove({}) //删除数据表  
db.stats() //查看所有的分片服务器状态  
db.adminCommand( { listShards: 1 } ) //分片列表  
db.test\_shard.find({ age: 36 }).explain() //精确查询  
db.test\_shard.find({ age: { $gte : 36 ,$lt : 73 } }).explain() //范围查询  
  
#分片 操作命令  
sh.enableSharding('testdb') //开启数据库testdb分片  
sh.shardCollection('testdb.users',{uid:1}) //按testdb.users的uid字段分片  
sh.shardCollection("testdb.test\_shard",{"age": 1}) //按ranged分片  
sh.shardCollection("testdb.test\_shard2",{"age": "hashed"}) //按hash分片  
sh.status() //查看分片节点  
sh.addShard() //向集群中添加一个 shard  
sh.getBalancerState() //查看平衡器  
sh.disableBalancing() //禁用平衡器  
sh.enableBalancing() //启用平衡器  
db.runCommand( { removeShard: "mongodb0" } ) //删除分片mongodb0，迁移数据查看命令  
db.runCommand( { movePrimary: "test", to: "mongodb1" }) //将数据库test未分片mongodb0的数据，迁移到mongodb1主分片。  
db.adminCommand("flushRouterConfig") //处理分片后，刷新路由数据。  
  
use config   
db.databases.find() //查看所有数据库使用分片  
db.settings.save({\_id:"chunksize",value:1}) //将 chunk 的大小调整为 1MB  
db.serverStatus().sharding  
  
  
#副本集 操作命令  
rs.status() //查看成员的运行状态等信息  
rs.config() //查看配置信息  
rs.slaveOk() //允许在SECONDARY节点上进行查询操作，默认从节点不具有查询功能  
rs.isMaster() //查询该节点是否是主节点  
rs.add({}) //添加新的节点到该副本集中  
rs.remove() //从副本集中删除节点  
rs.stepDown //降级节点  
db.printSlaveReplicationInfo() //查看同步情况  
rs.addArb("172.20.0.16:27038") //添加仲裁节点  
  
#强制加入仲裁节点：  
config=rs.conf()  
config.members=[config.members[0],config.members[1],{\_id:5,host:"127.0.0.1:27023",priority:5,arbiterOnly:"true"}]  
rs.reconfig(config,{force:true})  
  
#强制主节点：  
cfg = rs.conf()  
cfg.members[0].priority = 0.5  
cfg.members[1].priority = 0.5  
cfg.members[2].priority = 1  
rs.reconfig(cfg)  
  
#备份/恢复  
mongodump -h 127.0.0.1:27017 -d test -o /data/backup/  
mongorestore -h 127.0.0.1:27017 -d test --dir /data/db/test