

MongoDB课程

(讲师：程道)

第八部分 MongoDB监控和数据备份与恢复

8.1 MongoDB监控

8.1.1 MongoDB Ops Manager 简介

MongoDB Ops Manager(MMS) 是用于监控和备份MongoDB的基础设施服务。MongoDB OPS Manager是一个Web应用程序，它需要1个mongodb数据库，这个数据库是用来支持本身的MongoDB OPS Manager来运行的。因此，如果我们想要MongoDB OPS Manager运行起来，最少也需要安装一个MongoDB数据库。

8.1.2 Ops Manager 作用

- 简易的自动化数据库部署、扩展、升级和任务管理；
- 通过 OPS 平台提供的超过 100 项仪表、图表，可以对 mongodb 进行多种监控；
- 支持单节点、分片集群的备份和恢复；

8.1.3 安装Ops Manager

1.下载对应版本的 Ops Manager 安装包

<https://docs.opsmanager.mongodb.com/current/installation/>

2.上传到服务器并解压

```
tar -xvf mongodb-mms-4.1.3.53428.20190304T2149Z-1.x86_64.tar.gz
```

3.编辑配置文件

```
vi conf-mms.properties
```

根据自己的mongodb 进行配置 比如我这里改成了

```
mongodb://192.168.211.136:27777/?maxPoolSize=150
```

你需要先启动一个27777的mongodb实例

```
mongo.mongouri=mongodb://127.0.0.1:27777/?maxPoolSize=150
mongo.ssl=false
```

```
vi mms.conf
```

```
BASE_PORT=8080
BASE_SSL_PORT=8443

# JVM configurations
JAVA_MMS_UI_OPTS="$${JAVA_MMS_UI_OPTS} -Xss328k -Xmx4352m -Xms4352m -
XX:NewSize=600m -Xmn1500m -XX:ReservedCodeCacheSize=128m -XX:-
OmitStackTraceInFastThrow"
```

一般修改端口 和 内存 如果虚拟机内存不太够 可以适当减少内存配置 比如-Xmx4352m -Xms4352m
改成 -Xmx2352m -Xms2352m

4.配置完成后，启动Ops Manager 启动时间会比较长

启动之前确保 Ops Manager对应的 MongoDB 数据库已经启动

./bin/mongodb-mms start

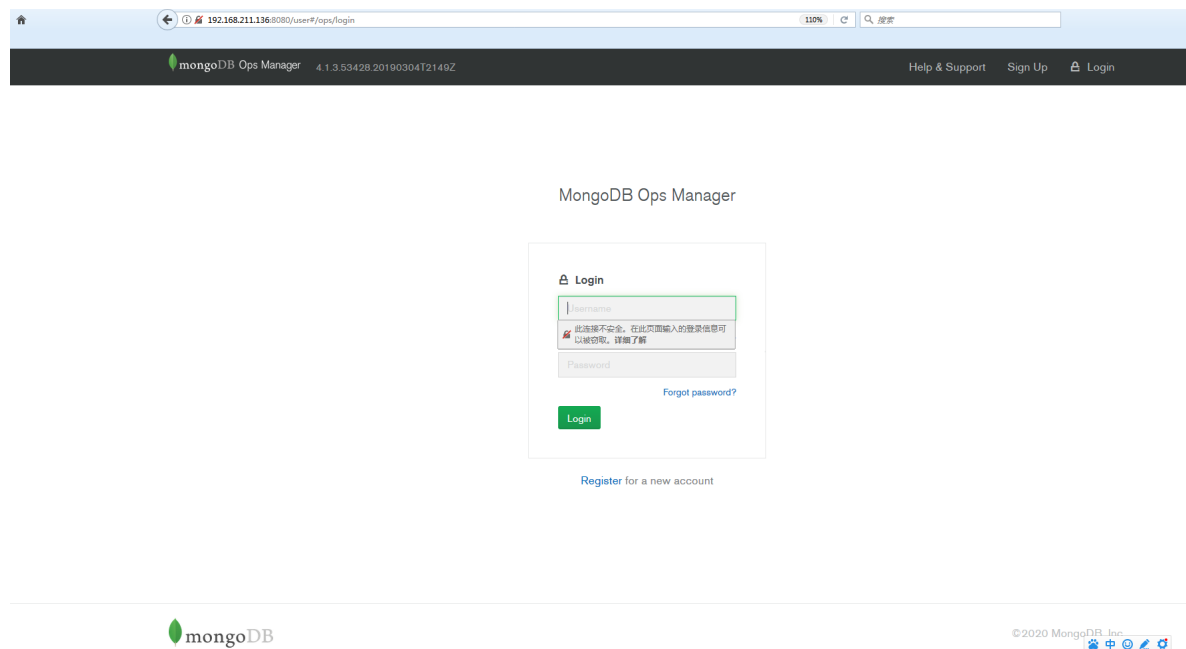
5.访问OPS Manager的首页，来进行应用配置

这里的端口是在 mms.conf 中配置的

http://主机:端口

8.1.4 配置 Ops Manager

1.登录界面



2.注册账号

Register for MongoDB Ops Manager

Email Address
chengdao

Password

✓ 8-character-minimum
✓ One-letter
✓ One-number
✓ One-special-character

First Name
cheng

Last Name
daq

☐ I agree to the [Evaluation Agreement](#)

Create Account

3.使用创建的用户登录。来到OPS Manager的配置页面。

Configure Ops Manager

Before you can start using Ops Manager, you'll need to tell us about your configuration. This setup wizard will guide you through entering all of the required and optional configuration. The configuration can be updated later by any Ops Manager user with the Global Owner permission by visiting the "Admin" link in the top right corner.

MongoDB Connection String: `mongodb://127.0.0.1:27017/?maxPoolSize=150`

Web Server

* required ** restart required

URL To Access Ops Manager *

Fully qualified URL, including the port number, of the Ops Manager Application.

Example: `http://opsmanager.example.com:9080`

HTTPS PEM Key File **

Absolute path to the PEM file that contains a valid certificate and private key for the Web Server. The PEM file is required for the web server to run with HTTPS. The default port for HTTPS is 8443. The Ops Manager URL may need to be updated.

Example: `/etc/security/opsmanager.pem`

上面的配置很多 大家只填写要求的字段即可 continue 很多 下面截图了几个关键图

192.168.211.136:8080/v2/admin#setup/webServerAndEmail

110%

搜索

mongoDB Ops Manager

Need Help?

Configure Ops Manager

Before you can start using Ops Manager, you'll need to tell us about your configuration. This setup wizard will guide you through entering all of the required and optional configuration. The configuration can be updated later by any Ops Manager user with the Global Owner permission by visting the "Admin" link in the top right corner.

MongoDB Connection String: `mongodb://192.168.211.136:27777/?maxPoolSize=150`

Web Server

* required ** restart required

URL To Access Ops Manager *

Fully qualified URL, including the port number, of the Ops Manager Application.
Example: `http://opsmanager.example.com:8080`

`http://192.168.211.136:8080`

HTTPS PEM Key File **

Absolute path to the PEM file that contains a valid certificate and private key for the Web Server. The PEM file is required for the web server to run with HTTPS. The default port for HTTPS is 8443. The Ops Manager URL may need to be updated.
Example: `/etc/security/opsmanager.pem`

8.1.5 配置MongoDB Ops Manager Agent

1.点击 下面界面中的 Manage your existing deployment

mongoDB Ops Manager 4.1.3 OK All Clusters

Please set your time zone Admin cheng

CONTEXT
Project 0

ORGANIZATION 0 > PROJECT 0

Deployment

Add New

PROJECT ACTIVE

Deployment

Alerts

Backup

Access

Settings

Docs

Support

Processes Servers Agents Security More

You don't have any deployments yet
Let's get started!

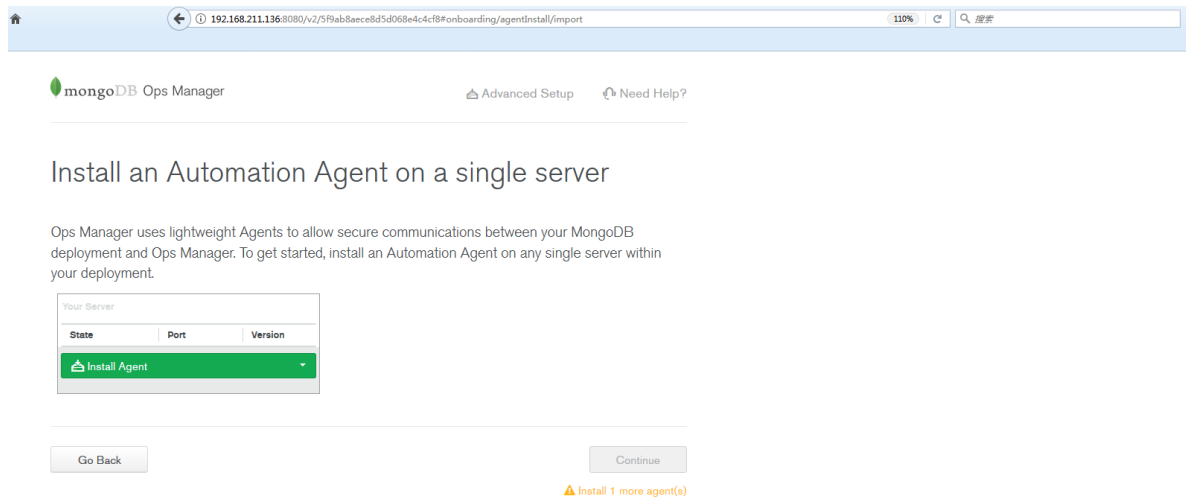
Build New Deployment

Already have a MongoDB deployment? [Manage your existing deployment.](#)

Will someone else be setting up your MongoDB deployment? [Add them to your project.](#)

Last Login: 192.168.211.1 Version: 4.1.3.53428.20190304T2149Z
Since: 2020-11-19T04:26:32Z Central URL: http://192.168.211.136:8080 Organization Name: Organization 0
©2020 MongoDB, Inc.

2.单击绿色install Agent



根据我们目标数据库的操作系统，我们可以选择相应的Agent来安装。（PS：实际的操作系统版本可能更多，大致上一致即可，只要能装上）

3.根据弹出的安装步骤 安装agent

To save time, you can repeat each step of these instructions in parallel across servers with the same OS

1. Download the agent

```
curl -OL http://192.168.211.136:8080/download/agent/automation/mongodb-mms-automation-agent-m
```

and install the package.

```
sudo rpm -U mongodb-mms-automation-agent-manager-6.3.1.5645-1.x86_64.rhel7.rpm
```

2. Create a new Agent API Key. After being generated, keys will only be shown once.
Treat this API Key like a password.

[+ Generate Key](#)

3. Next, open the config file

```
sudo vi /etc/mongodb-mms/automation-agent.config
```

and enter your API key, Project ID, and Ops Manager Base URL as shown below.

```
mmsGroupId=5fb5f3f8ce8d5d25dd148bf3
```

```
mmsApiKey=<Insert Agent API Key Here>
```

```
mmsBaseUrl=http://192.168.211.136:8080
```

To manage your API keys, visit the [Agent API Keys](#) tab.

4. Start the agent.

```
sudo systemctl start mongodb-mms-automation-agent.service
```

On SUSE, it may be necessary to run:

```
sudo /sbin/service mongodb-mms-automation-agent start
```

4.按照上面的提示的步骤一步步操作即可 下面是主要的截图

Install an Automation Agent on a single server

Ops Manager uses lightweight Agents to allow secure communications between your MongoDB deployment and Ops Manager. To get started, install an Automation Agent on any single server within your deployment.

localhost		
State	Port	Version
Automation Agent		6.3.1.5645
Automation Agent Successfully Verified		

Go Back

Continue

✓ Agents Verified

Set up Ops Manager Monitoring

The Automation Agent will now install a Monitoring Agent on your server. The Monitoring Agent will be used to collect health and performance metrics from your MongoDB deployment. A Backup Agent will also be installed, but will lie dormant until you choose to enable our Backup feature.

localhost		
State	Port	Version
Automation Agent		6.3.1.5645
Monitoring Agent		70.0.481
Backup Agent		75.0.1051
Monitoring Agent Successfully Verified		

Go Back

Continue

8.1.6 监控现有的Sharding Cluster服务

1.确保你的shard集群已经启动

确保路由 配置集群 分片集群都启动 （注意dbpath 和 logpath使用绝对路径 否则监控有警告）

2.在安装好的agent界面点击继续

Set up Ops Manager Monitoring

The Automation Agent will now install a Monitoring Agent on your server. The Monitoring Agent will be used to collect health and performance metrics from your MongoDB deployment. A Backup Agent will also be installed, but will lie dormant until you choose to enable our Backup feature.

localhost		
State	Port	Version
Automation Agent		6.3.1.5645
Monitoring Agent		70.0.481
Backup Agent		75.0.1051
Monitoring Agent Successfully Verified		

Go Back

Continue

3.配置监控服务 单机实例或者集群实例都可以

Import your deployment for monitoring

To get started, we'll need some basic information about one item in your MongoDB deployment before we can start monitoring it. From this single seed item, Ops Manager will automatically discover all connected processes. If you have a sharded cluster, seed Ops Manager with a single mongos. If you have a replica set, seed Ops Manager with the primary.

Hostname

This is the hostname of the seed MongoDB process as seen from the Ops Manager agent. This hostname must be unique and resolvable from any server within your deployment. Do not use 'localhost'.

192.168.211.136

Port

This is the port that the seed MongoDB process is running on. To test that you have chosen the right hostname and port, login to the server on which you installed the Ops Manager Agent and test connecting via the MongoDB shell:

```
mongo hostname:port
```

27017

Enable Authentication

Does your MongoDB deployment require authentication?

☐ NO

4.配置完成点击continue 会进行集群发现

the primary.

Hostname

192.168.211.136

This is the hostname of the seed MongoDB process as seen from the Ops Manager agent. This hostname must be unique and resolvable from any server within your deployment. Do not use 'localhost'.

Port

27017

This is the port that the seed MongoDB process is running on. To test that you have chosen the right hostname and port, login to the server on which you installed the Ops Manager Agent and test connecting via the MongoDB shell:

```
mongo hostname:port
```

Enable Authentication

NO

Does your MongoDB deployment require authentication?

Use TLS/SSL

NO

Do you want to use TLS/SSL for MongoDB connections?

Deployment found

继续continue

Ops Manager is discovering all MongoDB processes in your deployment. This may take up to 10 minutes, please be patient. Once you see all of the processes in your deployment here, you can hit continue. If not all processes are discovered, make sure that each process is able to accept connections from the server on which you installed the Ops Manager Agent.

Make sure you see all of the processes in your deployment before continuing

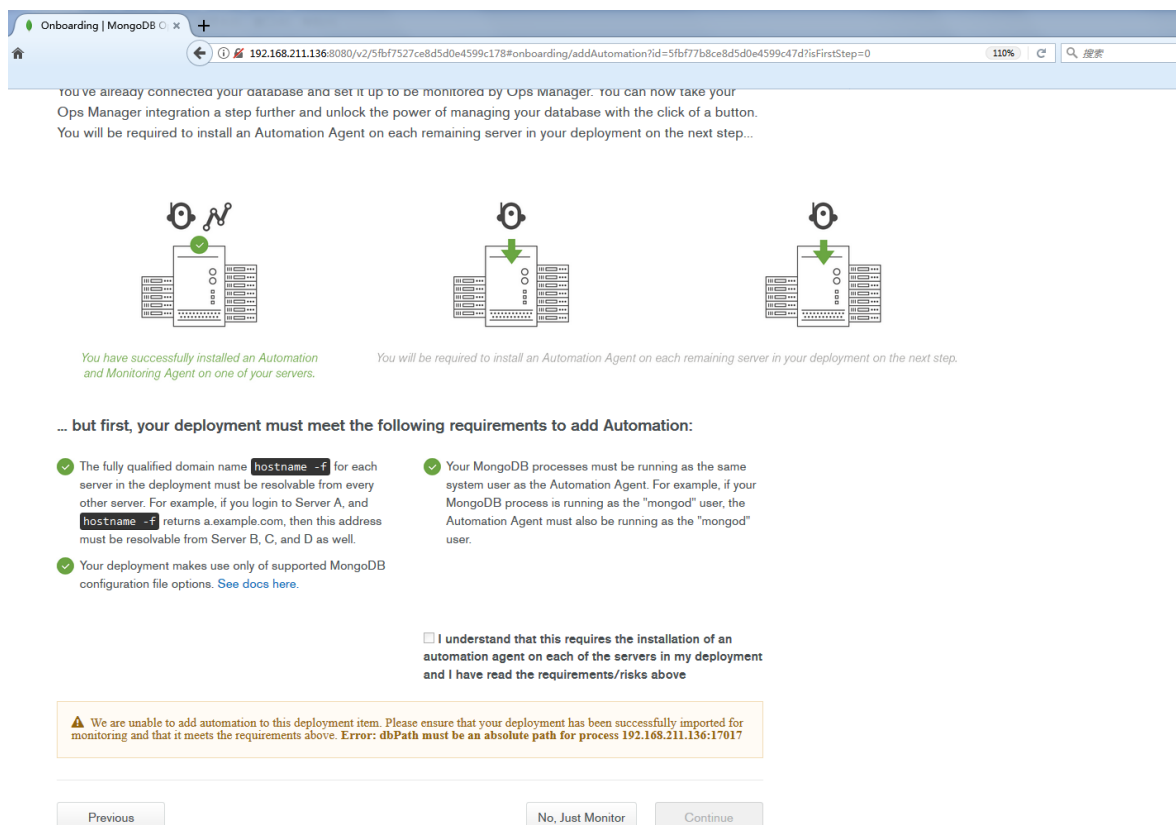
192.168.211.136		
OS Name: CentOS Linux release 7.4.1708 (Core)		
RAM: 1823 MB		
State	Port	Version
is shard1	37017	4.1.3
is shard1	37018	4.1.3
is shard1	37019	4.1.3
is shard2	47017	4.1.3
is shard2	47018	4.1.3
is shard2	47019	4.1.3
is config	17017	4.1.3
is config	17018	4.1.3
is config	17019	4.1.3
is mongos	27017	4.1.3

Go Back

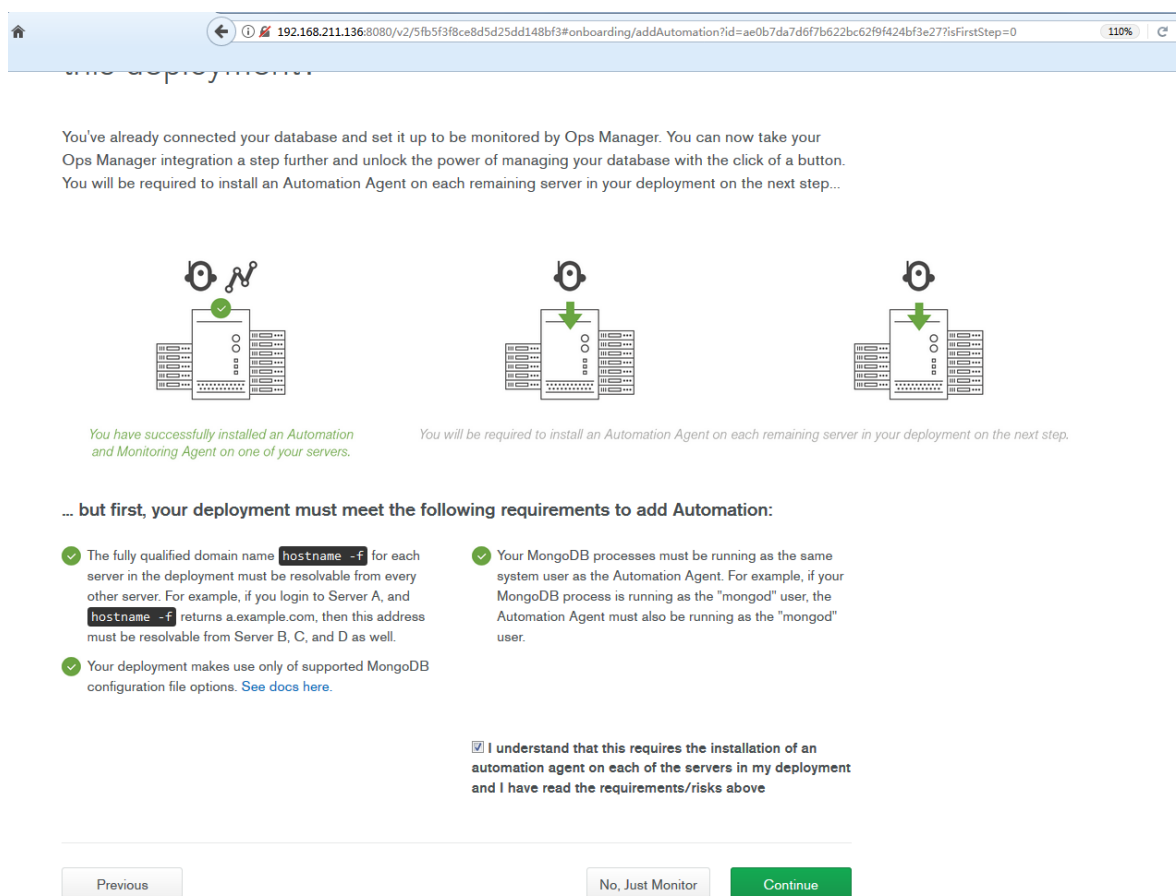
Continue

10 processes and 1 server discovered.

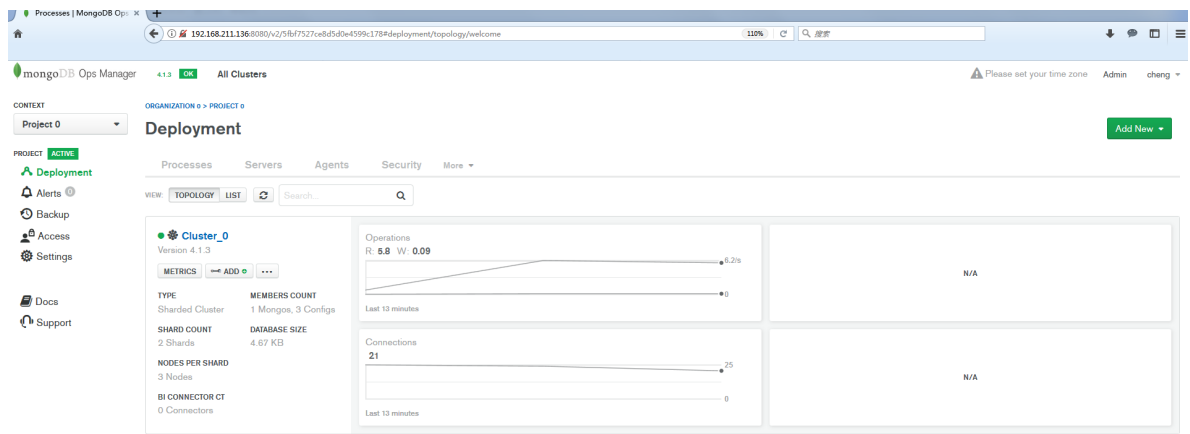
5.集群发现要求配置节点和分片节点必须是绝对路径 注意一下即可



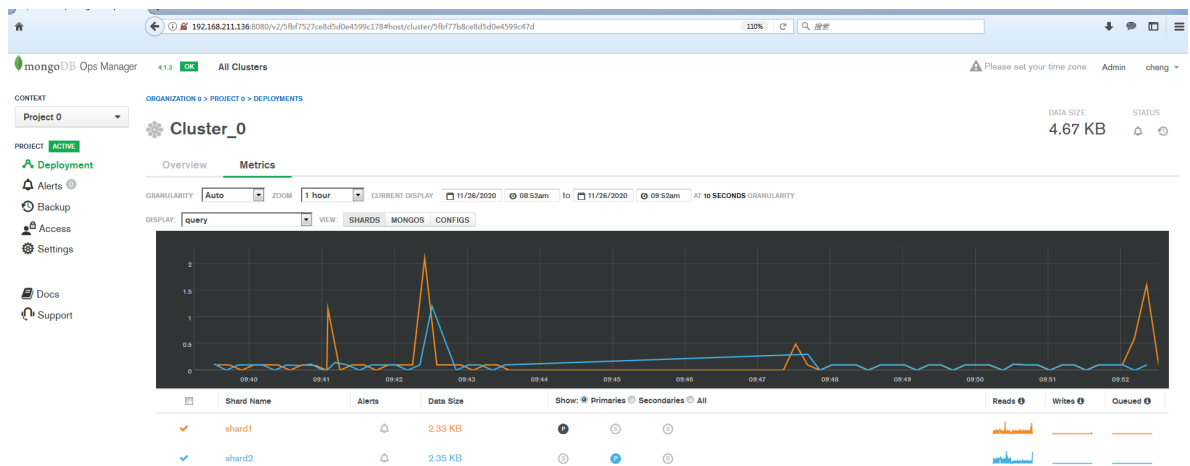
6.根据提示做修改 直到出现下面的界面



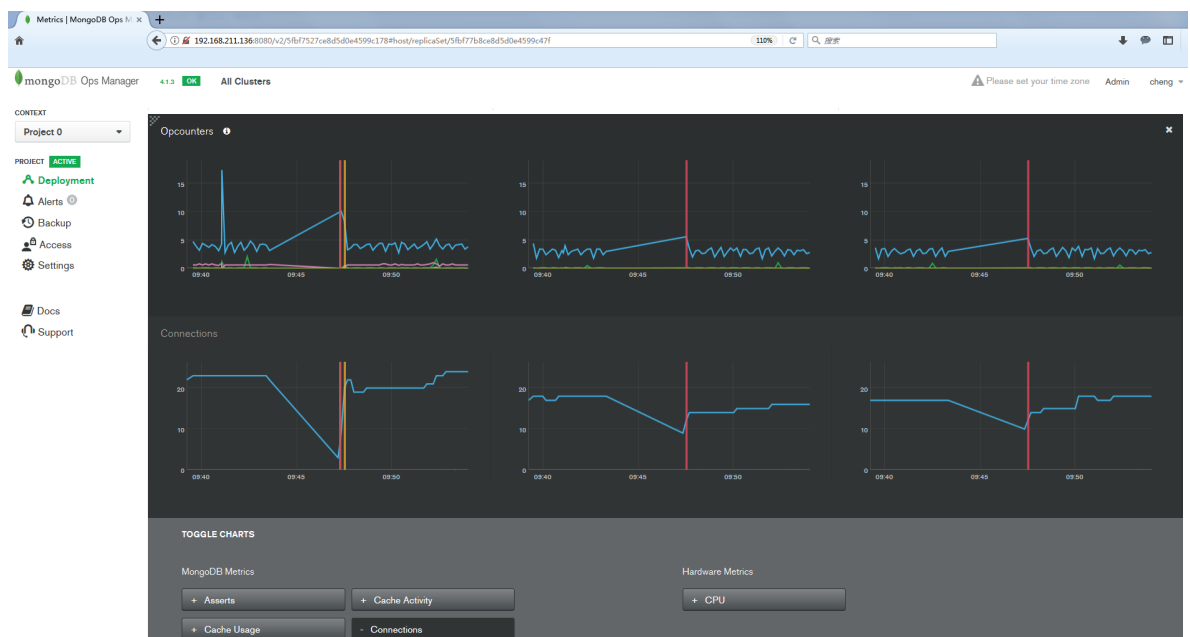
7.点击JustMonitor 即可看到集群信息



点击METRICS



8.查看具体的分片 和 监控指标



8.2 MongoDB 数据备份与恢复

8.2.1 备份的目的

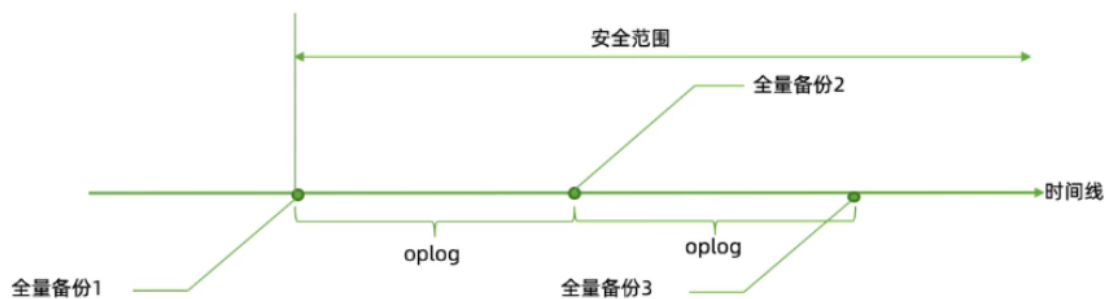
- * 防止硬件故障引起的数据丢失
- * 防止人为错误误删数据
- * 时间回溯
- * 监管要求

8.2.2 备份机制和实现方式

全量备份实现方式

- * 文件系统快照
- * 复制数据文件
- * mongodump

解决方案



- 最近的oplog已经在oplog.rs集合中，因此可以在定期从集合中导出便得到了oplog；
- 如果主节点上的oplog.rs集合足够大，全量备份足够密集，自然也可以不用备份oplog；
- 只要有覆盖整个时间段的oplog，就可以结合全量备份得到任意时间点的备份。



8.2.3 mongodump

在MongoDB 中我们使用mongodump命令来备份MongoDB数据。该命令可以导出所有数据到指定目录中。 mongodump命令可以通过参数指定导出的数据库或者集合。

mongodump命令脚本语法如下：

```
>mongodump -h dbhost -d dbname -o dbdirectory
```

- -h：

MongoDB所在服务器地址，例如：127.0.0.1，当然也可以指定端口号：127.0.0.1:37017

- --db 或者 -d :
需要备份的数据库实例，例如：lg
- -o :
备份的数据存放位置，例如：/root/bdatas 在备份完成后，系统自动在root目录下建立一个 bdatas目录，这个目录里面存放该数据库实例的备份数据。

mongodump 命令可选参数列表如下所示：

语法	描述	实例
mongodump --host HOST_NAME - -port PORT_NUMBER	该命令将备份所有 MongoDB数据	mongodump --host 192.168.211.136 --port 37017
mongodump --db DB_NAME --out BACKUP_DIRECTORY	备份指定的数据库	mongodump --port 37017 --d dbname --out /data/backup/
mongodump --collection COLLECTION --db DB_NAME	该命令将备份指定数 据库的集合。	mongodump --collection mycol -d lg

举例:

```
./bin/mongodump -h 127.0.0.1:37017 -d lg -o /root/bdatas
```

```
./bin/mongodump --host=192.168.211.136 --port=37017 -d local -c oplog.rs -o=/root/oplog_bak
```

8.2.4 mongorestore

mongodb使用 mongorestore 命令来恢复备份的数据。

mongorestore命令脚本语法如下：

```
>mongorestore -h <hostname><:port> -d dbname <path>
```

- --host <:port>, -h <:port> :
MongoDB所在服务器地址，默认为：localhost:37017
- --db 或者 -d :
需要恢复的数据库实例，例如：test，当然这个名称也可以和备份时候的不一样，比如test2
- --drop :
恢复的时候，先删除当前数据，然后恢复备份的数据。就是说，恢复后，备份后添加修改的数据都会被删除，慎用哦！
- :
mongorestore 最后的一个参数，设置备份数据所在位置，例如：/root/bdatas/lg
你不能同时指定 和 --dir 选项，--dir也可以设置备份目录。
注意: 恢复指定的数据库 需要在恢复的路径中出现数据库的名字
- --dir :
指定备份的目录
你不能同时指定 和 --dir 选项。

举例:

```
./bin/mongorestore -h 127.0.0.1:37017 -d lg /root/bdatas/lg
```

8.2.5 备份和恢复的重要选项 --oplog --oplogReplay --oplogLimit

mongodump有一个值得一提的选项是--oplog，注意这是replica set或者master/slave模式专用，standalone模式运行mongodb并不推荐。

来看看mongodump的选项：

--oplog选项只对全库导出有效，所以不能指定-d选项

--oplog的作用：oplog的幂等性：已存在的数据，重做oplog不会重复；不存在的数据重做oplog就可以进入数据库。

举例:

```
./bin/mongodump -h 127.0.0.1:37017 --oplog -o /root/bdatas
```

来看看mongorestore的选项。

--oplogReplay：可以重放oplog.bson中的操作内容

--oplogLimit：回放的时间节点，即此时间之前的数据恢复，假设你后面有误操作，误操作的不恢复

```
mongorestore -h localhost:37017 --oplogReplay /root/dump
```

通过 oplog 查询误操作的最后时间

```
/root/mongodb/bin/bsondump oplog.rs.bson | grep '"op":"d"' | head
```

或者 使用

```
db.oplog.rs.find({"op" : "d"}).sort({"ts":-1})
```

举例:

```
mongorestore -h localhost:37017 --oplogReplay --oplogLimit "1443024507:1" /root/dump/local
```

8.2.6 全量加增量备份和恢复案例

注意的问题:

删除复制集中原来的数据文件目录 重新建立数据目录

重新启动复制集中的实例 进行复制集的配置

```
var cfg = {"_id": "lagouCluster",
           "protocolVersion" : 1,
           "members": [
             {"_id": 1, "host": "192.168.211.136:37017", "priority": 10},
             {"_id": 2, "host": "192.168.211.136:37018"},
             {"_id": 3, "host": "192.168.211.136:37019"}
           ]
}

rs.initiate(cfg)
```

1.进入mongodb 插入两条数据

```
use lg
db.lg_resume.insert({name:"test1",salary:18000.5})
db.lg_resume.insert({name:"test2",salary:15000.5})
```

2.进行全量备份

```
./bin/mongodump --host=192.168.211.136 --port=37017 --out=/root/fullbackup
```

3.继续插入数据 并更新

```
db.lg_resume.insert({name:"test3",salary:28000.5})
db.lg_resume.insert({name:"test4",salary:35000.5})
db.lg_resume.insert({name:"test5",salary:45000.5})
db.lg_resume.update({name:"test3"},{$set:{salary:38000.5}})
db.lg_resume.update({name:"test4"},{$set:{salary:45000.5}})
```

4.做增量备份

```
./bin/mongodump --host=192.168.211.136 --port=37017 -d local -c oplog.rs -
o=/root/oplog_bak
```

5.删除所有的数据

```
db.lg_resume.remove({})
db.lg_resume.find()
```

6.先恢复全量数据

```
./bin/mongorestore --host=192.168.211.136 --port=37017 --dir=/root/fullbackup
查看数据恢复
db.lg_resume.find()
```

7.恢复数据到指定的时间点

改变oplog.rs.bson 为 oplog.bson 删除oplog.rs.metadata.bson

```
mv /root/oplog_bak/local/oplog.rs.bson /root/oplog_bak/local/oplog.bson
rm /root/oplog_bak/local/oplog.rs.metadata.json -rf
```

找出第一次更新的时间

```
use local
db.oplog.rs.find({"op" : "u"}).sort({"ts":1})
```

恢复到指定的时间点的数据

```
./bin/mongorestore --host=192.168.211.136 --port=37017 --oplogReplay --
oplogLimit "实际查询出来的时间" /root/oplog_bak/local
1606651336
```

```
./bin/mongorestore --host=192.168.211.136 --port=37017 --oplogReplay --
oplogLimit "1606651336:4" /root/oplog_bak/local
```

查看数据恢复

```
use lg
db.lg_resume.find()
```

8.恢复所有的增量数据

```
./bin/mongorestore --host=192.168.211.136 --port=37017 --oplogReplay  
/root/oplog_bak/local  
查看数据恢复  
db.lg_resume.find()
```

8.2.7 定时备份

1.准备备份目录

```
mkdir -p /root/backup/mongod_bak/mongod_bak_now  
/root/backup/mongod_bak/mongod_bak_list
```

2.编写备份脚本

```
vi /root/backup/mongobk.sh
```

```
#!/bin/sh  
# dump 命令执行路径，根据mongodb安装路径而定  
DUMP=/root/mongodb/bin/mongodump  
# 临时备份路径  
OUT_DIR=/root/backup/mongod_bak/mongod_bak_now  
# 压缩后的备份存放路径  
TAR_DIR=/root/backup/mongod_bak/mongod_bak_list  
# 当前系统时间  
DATE=`date +%Y_%m_%d%H%M%S`  
# 数据库账号  
#DB_USER=user  
# 数据库密码  
#DB_PASS=password  
# 代表删除7天前的备份，即只保留近 7 天的备份  
DAYS=7  
# 最终保存的数据库备份文件  
TAR_BAK="mongod_bak_${DATE}.tar.gz"  
cd $OUT_DIR  
rm -rf $OUT_DIR/*  
mkdir -p $OUT_DIR/$DATE  
$DUMP -h 127.0.0.1 --port 37017 -o $OUT_DIR/$DATE  
# 压缩格式为 .tar.gz 格式  
tar -zPcvf $TAR_DIR/$TAR_BAK $OUT_DIR/$DATE  
# 删除 7 天前的备份文件  
find $TAR_DIR/ -mtime +$DAYS -delete  
  
exit
```

3.修改脚本权限

```
chmod +x /root/backup/mongobk.sh
```

4.编辑crontab

crontab -e

```
#表示每天凌晨2点30执行备份  
30 2 * * * /root/backup/mongobk.sh
```

测试的时候 可以改成 一分钟备份一次

```
* * * * * /root/backup/mongobk.sh
```

5.查看crontab 的状态

service crond status

6.如果没有启动 可以使用下面的命令 启动定时服务 和加入开机自启动

```
# 启动定时任务  
service crond start  
# 加入开机自动启动  
chkconfig --level 35 crond on
```

7.查看定时任务和删除定时任务

```
crontab -l  
crontab -r  
crontab -e
```