

# 在CPS副本集中管理仲裁器节点的过程

## 目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[问题](#)

[管理副本集中的仲裁器的过程](#)

## 简介

本文档介绍在思科策略套件(CPS)副本集中管理仲裁器节点的过程。

## 先决条件

### 要求

Cisco 建议您了解以下主题：

- Linux
- CPS
- MongoDB

**注意:**Cisco建议您必须具有对CPS CLI的超级用户访问权限。

### 使用的组件

本文档中的信息基于以下软件和硬件版本：

- CPS 20.2
- 统一计算系统(UCS)-B
- MongoDB v3.6.17和v3.4.16

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

## 背景信息

CPS使用MongoDB构建其基本数据库(DB)结构。它拥有多个副本集，用于各种用途：ADMIN、Subscriber Profile Repository(SCR)、BALANCE、SESSION、REPORTING和AUDIT。

MongoDB中的副本集是一组维护相同数据集的mongod进程。副本集提供冗余和高可用性(HA)。通过在不同的数据库服务器上有多个数据副本，它允许负载共享读取操作。

在某些情况下（例如您有一个主实例和一个辅助实例，但成本限制禁止添加另一个辅助实例），您可以选择向副本集添加一个单实例作为仲裁者在选举中进行投票。仲裁员有1张选举票。默认情况下，仲裁员的优先级为0。

仲裁程序是属于副本集但不保存数据的单实例（这意味着它们不提供数据冗余）。不过，他们也可以参加选举。仲裁员参与主节点的选举，但仲裁员没有数据集的副本，因此无法成为主节点。

仲裁程序具有最低的资源要求，不需要专用硬件。您可以在应用服务器或仅监控网络的主机上部署仲裁程序。

仲裁器不存储数据，但在将仲裁器mongod进程添加到副本集之前，仲裁器的作用类似于任何其他单宿主进程，并使用一组数据文件和完整大小的日志启动。

下面是一个示例副本集，即 `set07`。

```
| SET NAME - PORT : IP ADDRESS - REPLICHA STATE - HOST NAME - HEALTH - LAST SYNC -PRIORITY
|-----|
| SESSION:set07 |
| Status via arbitervip:27727 sessionmgr01:27727 sessionmgr02:27727 |
| Member-1 - 27727 : - SECONDARY - sessionmgr01 - ON-LINE - 0 sec - 2 |
| Member-2 - 27727 : 192.168.10.146 - ARBITER - arbitervip - ON-LINE - ----- - 0 |
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - ----- - 3 |
|-----|
```

## 问题

假定仲裁程序有问题，或者需要更改复制副本集中的仲裁程序，则必须删除当前仲裁程序并向复制副本集中添加新仲裁程序。

## 管理副本集中的仲裁器的过程

步骤1:验证CPS和新仲裁程序中的mongo shell版本。从副本集和新仲裁器节点中的主sessionmgr运行此命令。

sessionmgr的输出示例：

```
[root@sessionmgr02 ~]# mongo --version
MongoDB shell version v3.6.17
```

如果主sessionmgr和新仲裁程序中的mongo shell版本相同，或者新仲裁程序mongo shell版本更高，则导航到步骤6。

否则，如果新的仲裁程序mongo shell版本较低，则必须设置 `featureCompatibilityVersion` 作为副本集管理数据库中的较低值，并执行以下步骤。

新仲裁程序的mongo shell版本低于CPS sessionmgr的示例案例：

```
[root@pcrfclient02 ~]# mongo --version
MongoDB shell version v3.4.16
```

第二步：登录到复制副本集的主蒙戈实例。

Command template:

```
#mongo --host <sessionmgrXX> --port <Replica Set port>
```

Sample command:

```
#mongo --host sessionmgr02 --port 27727
```

**第三步：运行此命令可查看当前 featureCompatibilityVersion 在副本集的admin数据库中。**

```
set07:PRIMARY> db.adminCommand( { getParameter: 1, featureCompatibilityVersion: 1 } )
{
  "featureCompatibilityVersion" : {
    "version" : "3.6"
  },
  "ok" : 1,
  "operationTime" : Timestamp(1663914140, 1),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1663914140, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}
```

**第四步：运行此命令以 setfeatureCompatibilityVersion 在副本集的管理数据库中设置为3.4。**

```
set07:PRIMARY> db.adminCommand( { setFeatureCompatibilityVersion: "3.4" } )
{ "ok" : 1 }
```

**第五步：运行此命令以验证 featureCompatibilityVersion 在副本集的管理员数据库中已更改为3.4。**

```
set07:PRIMARY> db.adminCommand( { getParameter: 1, featureCompatibilityVersion: 1 } )
{ "featureCompatibilityVersion" : { "version" : "3.4" }, "ok" : 1 }
```

**第六步：登录到群集管理器并修改 /var/qps/config/deploy/csv/AdditionalHosts.csv 新仲裁员详细信息文件。**

```
#vi /var/qps/config/deploy/csv/AdditionalHosts.csv
```

Provide new arbiter details in this format:

Host Alias IP Address

```
new-arbiter new-arbiter xx.xx.xx.xx
```

**步骤 7.导入CSV配置。**

```
#!/var/qps/install/current/scripts/import/import_deploy.sh
```

**步骤 8验证 /etc/hosts 更新了新仲裁人的信息。**

```
#cat /etc/hosts | grep arbiter
```

**步骤 9运行此命令进行同步 /etc/hosts.**

```
#!/var/qps/bin/update/synchosts.sh
```

```
Syncing to following QNS Servers:
lb01 lb02 sessionmgr01 sessionmgr02 qns01 qns02 pcrfclient01 pcrfclient02
Do you want to Proceed? (y/n):y
lb01
lb02
sessionmgr01
sessionmgr02
qns01
qns02
pcrfclient01
pcrfclient02
```

步骤 10验证pcrfclient虚拟机上的mon\_db脚本已停止。

```
#monsum | grep mon_db_for
如果停止，则输出如下：
```

```
mon_db_for_lb_failover Not monitored Program
mon_db_for_callmodel Not monitored Program
```

如果未停止，则输出如下：

```
mon_db_for_lb_failover OK Program
mon_db_for_callmodel OK Program
```

**注意：**如果mon\_db脚本未停止，请在相应的pcrfclient虚拟机上运行这些命令以手动停止。

```
#monit stop mon_db_for_lb_failover
#monit stop mon_db_for_callmodel
```

步骤 11从pcrfclient01运行此命令可从副本集中删除当前仲裁器（在此步骤中，set07是一个示例）。

```
#build_set.sh --session --remove-members --setname set07
```

```
Please enter the member details which you going to remove from the replica-set
Member:Port -----> arbitervip:27727
arbitervip:27727
Do you really want to remove [yes(y)/no(n)]: y
```

步骤 12从集群管理器运行此命令以验证是否已将仲裁程序从删除 set07，的输出 set07 不能包含当前仲裁者。

```
#diagnostics.sh --get_replica_status
```

Expected output:

```
-----|
|-----|
|-----|
| SESSION:set07 |
| Status via sessionmgr01:27727 sessionmgr02:27727 |
| Member-1 - 27727 : - SECONDARY - sessionmgr01 - ON-LINE - 0 sec -|
| Member-2 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - ----- -|
|-----|
|-----|
```

步骤 13更新 mongoConfig.cfg 文件，以便在修改的副本集中具有正确的仲裁器。使用新的仲裁器

(ARBITER=new-arbiter)替换当前仲裁器(ARBITER=arbiter)。从集群管理器运行此命令。

```
#vi /etc/broadhop/mongoConfig.cfg
```

### 当前配置

```
[SESSION-SET2]
SETNAME=set07
OPLOG_SIZE=5120
ARBITER=arbitervip:27727
ARBITER_DATA_PATH=/var/data/sessions.7
MEMBER1=sessionmgr02:27727
MEMBER2=sessionmgr01:27727
DATA_PATH=/var/data/sessions.1/2
[SESSION-SET2-END]
```

### 所需的配置：

```
[SESSION-SET2]
SETNAME=set07
OPLOG_SIZE=5120
ARBITER=new-arbiter:27727
ARBITER_DATA_PATH=/var/data/sessions.7
MEMBER1=sessionmgr02:27727
MEMBER2=sessionmgr01:27727
DATA_PATH=/var/data/sessions.1/2
[SESSION-SET2-END]
```

步骤 14复制更新的 mongoConfig.cfg 文件到所有VM。从集群管理器运行此命令。

```
#copytoall.sh /etc/broadhop/mongoConfig.cfg /etc/broadhop/mongoConfig.cfg
```

步骤 15将新的仲裁器成员添加到set07。从Cluster Manager，运行 /var/qps/install/current/scripts/build/build\_etc.sh 命令，以生成 /etc/directory。

步骤 16在运行Replica集后，验证是否已将新的仲裁器成员添加到副本集 build\_etc.sh 脚本，现在您必须等待AIDO服务器使用新仲裁程序创建/更新副本集。

```
#diagnostics.sh --get_replica_status
```

Expected Output:

```
| SET NAME - PORT : IP ADDRESS - REPLICHA STATE - HOST NAME - HEALTH - LAST SYNC -PRIORITY
|-----|
|-----|
| SESSION:set07 |
| Status via arbitervip:27727 sessionmgr01:27727 sessionmgr02:27727 |
| Member-1 - 27727 : - SECONDARY - sessionmgr01 - ON-LINE - 0 sec - 2 |
| Member-2 - 27727 : xx.xx.xx.xx - ARBITER - new-arbiter - ON-LINE - ----- - 0 |
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - ----- - 3 |
|-----|
```

**注意：**如果未添加新的仲裁程序成员，请继续执行后续步骤。否则，请导航至步骤18。

步骤 17从群集管理器运行此命令，以便强制添加新的仲裁器成员。

```
#build_set.sh --DB_NAME --add-members --setname Setxxx --force
```

步骤 18.如果仲裁器端口尚未启动，请从新的仲裁器节点运行此命令以启动相同的仲裁器端口。

Command syntax:

```
#/etc/init.d/sessionmgr-XXXXX start
```

Sample command:

```
#/etc/init.d/sessionmgr-27727 start
```

步骤 19.确认新仲裁程序添加成功。

```
#diagnostics.sh --get_replica_status
```

步骤 20.从集群管理器运行此命令可相应地更新数据库优先级。

```
# cd /var/qps/bin/support/mongo/  
# ./set_priority.sh --db session  
# ./set_priority.sh --db spr  
# ./set_priority.sh --db admin  
# ./set_priority.sh --db balance  
# ./set_priority.sh --db audit  
# ./set_priority.sh --db report
```

步骤 21.从群集管理器运行此命令以验证复制副本集中的更改。

```
#diagnostics.sh --get_replica_status
```

Expected Output:

```
| SET NAME - PORT : IP ADDRESS - REPLICA STATE - HOST NAME - HEALTH - LAST SYNC -PRIORITY  
|-----  
|  
| SESSION:set07 |  
| Status via arbitervip:27727 sessionmgr01:27727 sessionmgr02:27727 |  
| Member-1 - 27727 : - SECONDARY - sessionmgr01 - ON-LINE - 0 sec - 2 |  
| Member-2 - 27727 : xx.xx.xx.xx - ARBITER - new-arbiter - ON-LINE - ----- - 0 |  
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - ----- - 3 |  
|-----|
```

步骤 22.验证mon\_db脚本是否已在pcrfclient VM上还原。否则，您必须手动启动它们。

```
#monsum | grep mon_db_for
```

要启用mon\_db脚本，请登录所有pcrfclient VM并运行以下命令：

```
# monit start mon_db_for_lb_failover  
# monit start mon_db_for_callmodel
```

## 关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。