

处理CPS副本集中会话管理器角色和优先级的过程

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[问题](#)

[从主会话管理器和在副本集中更改会话管理器优先级的过程](#)

[方法1](#)

[方法2](#)

简介

本文档介绍在思科策略套件(CPS)副本集中从主角色移动会话管理器和更改会话管理器优先级的过程。

先决条件

要求

Cisco 建议您了解以下主题：

- Linux
- CPS
- MongoDB

思科建议您必须拥有对CPS CLI的根访问权限。

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
- MongoDB v3.6.17
- UCS-B

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，其中mongod进程在Sessionmgr虚拟机(VM)上运行，以构成其基本数据库结构。它具有多个用于各种目的的副本集，这些副本集是ADMIN、用户配置文件存储库(SPR)、BALANCE、SESSION、REPORTING和AUDIT。

MongoDB中的复制副本集是维护相同数据集的一组单一进程。副本集提供冗余和高可用性。通过在不同数据库服务器上提供多个数据副本，它允许负载共享读取操作。

副本集包含多个承载数据的节点和可选的一个仲裁节点。在承载数据的节点中，一个成员和一个成员被视为主节点，而其他节点被视为辅助节点（副本集可以有多个辅助节点）。主节点处理所有写操作。

从站点复制主站点的操作日志(oplog)并将操作应用到其数据集，以便从站点的数据集反映主站点的数据集。如果主要不可用，合格的辅助将进行选举以选择自己为新的主要。仲裁器参与选举，但不保存数据。

要获取副本集状态，请从ClusterManager或pcrfclient中运行命令`diagnostics.sh --get_r`。

此处提供了一个副本集示例。**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 |
|-----|
```

要获取副本集配置信息，请执行以下步骤。

步骤1.登录到该复制副本集的主MongoDB成员。从ClusterManager运行此命令。

```
Command template:
#mongo --host <sessionmgrXX> --port <Replica Set port>
```

```
Sample command:
#mongo --host sessionmgr02 --port 27727
```

步骤2.运行命令以获取副本集配置信息。

```
set07:PRIMARY> rs.conf()
{
  "_id" : "set07",
  "version" : 2,
  "members" : [
    {
      "_id" : 0,
      "host" : "sessionmgr01:27727",
      "arbiterOnly" : false,
      "buildIndexes" : true,
      "hidden" : false,
      "priority" : 2,
      "tags" : {
```

```

},
"slaveDelay" : NumberLong(0),
"votes" : 1
},
{
  "_id" : 1,
  "host" : "arbitervip:27727",
  "arbiterOnly" : true,
  "buildIndexes" : true,
  "hidden" : false,
  "priority" : 0,
  "tags" : {
  },
  "slaveDelay" : NumberLong(0),
  "votes" : 1
},
{
  "_id" : 2,
  "host" : "sessionmgr02:27727",
  "arbiterOnly" : false,
  "buildIndexes" : true,
  "hidden" : false,
  "priority" : 3,
  "tags" : {
  },
  "slaveDelay" : NumberLong(0),
  "votes" : 1
}
],
"settings" : {
  "chainingAllowed" : true,
  "heartbeatIntervalMillis" : 2000,
  "heartbeatTimeoutSecs" : 1,
  "electionTimeoutMillis" : 10000,
  "catchUpTimeoutMillis" : -1,
  "catchUpTakeoverDelayMillis" : 30000,
  "getLastErrorModes" : {
  },
  "getLastErrorDefaults" : {
    "w" : 1,
    "wtimeout" : 0
  },
  "replicaSetId" : ObjectId("61cdb17a80b097a2e7604c97")
}
}
set07:PRIMARY>

```

注意：副本集中优先级最高的会话管理器用作主成员。

问题

假设会话管理器在一个或多个副本集中执行主成员的角色，在这些情况下，必须将副本集主角色移到其他一些会话管理器，

1. 无论何时执行任何涉及此会话管理器VM关闭的活动，都可实现平稳过渡。

2.如果会话管理器运行状况由于某种原因而降低，则与其他一些正常会话管理器一起维护副本集的适当功能。

从主会话管理器和在副本集中更改会话管理器优先级的过程

方法1

此处，副本集中会话管理器的优先级在MongoDB级别直接更改。以下是将sessionmg02从set07的主角色中移出的步骤。

选项1.更改sessionmgr02的优先级。

步骤1.登录到该复制副本集的主MongoDB成员。

Command template:

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

Sample command:

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

步骤2.运行命令以获取副本集配置信息。

```
set07:PRIMARY> rs.conf()
{
  "_id" : "set07",
  "version" : 2,
  "members" : [
    {
      "_id" : 0, -----> Position 0
      "host" : "sessionmgr01:27727",
      "arbiterOnly" : false,
      "buildIndexes" : true,
      "hidden" : false,
      "priority" : 2,
      "tags" : {

    },
    "slaveDelay" : NumberLong(0),
    "votes" : 1
  },
  {
    "_id" : 1, -----> Position 1
    "host" : "arbiterVIP:27727",
    "arbiterOnly" : true,
    "buildIndexes" : true,
    "hidden" : false,
    "priority" : 0,
    "tags" : {

  },
  "slaveDelay" : NumberLong(0),
  "votes" : 1
},
{
  "_id" : 2, -----> Position 2
  "host" : "sessionmgr02:27727",
  "arbiterOnly" : false,
```

```

"buildIndexes" : true,
"hidden" : false,
"priority" : 3,
"tags" : {
},
"slaveDelay" : NumberLong(0),
"votes" : 1
}
],
"settings" : {
"chainingAllowed" : true,
"heartbeatIntervalMillis" : 2000,
"heartbeatTimeoutSecs" : 1,
"electionTimeoutMillis" : 10000,
"catchUpTimeoutMillis" : -1,
"catchUpTakeoverDelayMillis" : 30000,
"getLastErrorModes" : {
},
"getLastErrorDefaults" : {
"w" : 1,
"wtimeout" : 0
},
"replicaSetId" : ObjectId("61cdb17a80b097a2e7604c97")
}
}
set07:PRIMARY>

```

注意：记录rs.conf()输出中各会话管理器的位置。

步骤3.运行此命令将终端移至配置模式。

```

set07:PRIMARY> cfg = rs.conf()
{
  "_id" : "set07",
  "version" : 2,
  "members" : [
    {
      "_id" : 0,
      "host" : "sessionmgr01:27727",
      "arbiterOnly" : false,
      "buildIndexes" : true,
      "hidden" : false,
      "priority" : 2,
      "tags" : {
    },
    "slaveDelay" : NumberLong(0),
    "votes" : 1
  },
  {
    "_id" : 1,
    "host" : "arbitervip:27727",
    "arbiterOnly" : true,
    "buildIndexes" : true,
    "hidden" : false,
    "priority" : 0,
    "tags" : {
  },
}

```

```

"slaveDelay" : NumberLong(0),
"votes" : 1
},
{
  "_id" : 2,
  "host" : "sessionmgr02:27727",
  "arbiterOnly" : false,
  "buildIndexes" : true,
  "hidden" : false,
  "priority" : 3,
  "tags" : {

  },
  "slaveDelay" : NumberLong(0),
  "votes" : 1
}
],
"settings" : {
  "chainingAllowed" : true,
  "heartbeatIntervalMillis" : 2000,
  "heartbeatTimeoutSecs" : 1,
  "electionTimeoutMillis" : 10000,
  "catchUpTimeoutMillis" : -1,
  "catchUpTakeoverDelayMillis" : 30000,
  "getLastErrorModes" : {

  },
  "getLastErrorDefaults" : {
    "w" : 1,
    "wtimeout" : 0
  },
  "replicaSetId" : ObjectId("61cdb17a80b097a2e7604c97")
}
}
set07:PRIMARY>

```

步骤4.运行此命令以更改sessionmgr的优先级。

Command template:
 cfg.members[X].priority = X --> put the position here in [].

sample command:
 cfg.members[2].priority = 1

此处，sessionmgr02当前是主成员，其位置为2，优先级为3。

要将此sessionmgr02移出主角色，请提供大于0但小于具有最高优先级的辅助成员的优先级的最低优先级编号，例如。1。

```

set07:PRIMARY> cfg.members[2].priority = 1
1
set07:PRIMARY>

```

步骤5.运行此命令以提交更改。

```

set07:PRIMARY> rs.reconfig(cfg)
{
  "ok" : 1,
  "operationTime" : Timestamp(1641528658, 1),

```

```

"$clusterTime" : {
"clusterTime" : Timestamp(1641528658, 1),
"signature" : {
"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"keyId" : NumberLong(0)
}
}
}
2022-01-07T04:10:57.280+0000 I NETWORK [thread1] trying reconnect to sessionmgr02:27727
(192.168.10.140) failed
2022-01-07T04:10:57.281+0000 I NETWORK [thread1] reconnect sessionmgr02:27727 (192.168.10.140)
ok
set07:SECONDARY>

```

步骤6.再次运行命令以验证sessionmgr优先级中的更改。

```

set07:SECONDARY> rs.conf()
{
  "_id" : "set07",
  "version" : 3,
  "members" : [
    {
      "_id" : 0,
      "host" : "sessionmgr01:27727",
      "arbiterOnly" : false,
      "buildIndexes" : true,
      "hidden" : false,
      "priority" : 2,
      "tags" : {

    },
    "slaveDelay" : NumberLong(0),
    "votes" : 1
  },
  {
    "_id" : 1,
    "host" : "arbitervip:27727",
    "arbiterOnly" : true,
    "buildIndexes" : true,
    "hidden" : false,
    "priority" : 0,
    "tags" : {

  },
  "slaveDelay" : NumberLong(0),
  "votes" : 1
},
{
  "_id" : 2,
  "host" : "sessionmgr02:27727",
  "arbiterOnly" : false,
  "buildIndexes" : true,
  "hidden" : false,
  "priority" : 1, --> Here priority has been changed from 3 to 1.
  "tags" : {

},
"slaveDelay" : NumberLong(0),
"votes" : 1
}
],
"settings" : {

```

```

"chainingAllowed" : true,
"heartbeatIntervalMillis" : 2000,
"heartbeatTimeoutSecs" : 1,
"electionTimeoutMillis" : 10000,
"catchUpTimeoutMillis" : -1,
"catchUpTakeoverDelayMillis" : 30000,
"getLastErrorModes" : {
},
"getLastErrorDefaults" : {
"w" : 1,
"wtimeout" : 0
},
"replicaSetId" : ObjectId("61cdb17a80b097a2e7604c97")
}
}
set07:SECONDARY>

```

步骤7.从ClusterManager或pcrfclient运行diagnostics.sh —get_r命令，以验证副本集状态中的更改。

```

| 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 : - PRIMARY - sessionmgr01 - ON-LINE - ----- - 2 |
| Member-2 - 27727 : 192.168.10.146 - ARBITER - arbitervip - ON-LINE - ----- - 0 |
| Member-3 - 27727 : - SECONDARY - sessionmgr02 - ON-LINE - 0 sec - 1 |
|-----|

```

现在，您看到sessionmgr02已移至辅助。要使sessionmgr02再次成为主要成员，请在步骤4中使用此命令运行上述步骤1.到5。

cfg.members[2].priority =大于2但小于1001的任意数字 —>优先级比示例中为2的当前主成员的优先级高。

```

set07:PRIMARY> cfg.members[2].priority = 5
5
set07:PRIMARY> rs.reconfig(cfg)
{
"ok" : 1,
"operationTime" : Timestamp(1641531450, 1),
"$clusterTime" : {
"clusterTime" : Timestamp(1641531450, 1),
"signature" : {
"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"keyId" : NumberLong(0)
}
}
}
2022-01-07T04:57:31.247+0000 I NETWORK [thread1] trying reconnect to sessionmgr01:27727
(192.168.10.139) failed
2022-01-07T04:57:31.247+0000 I NETWORK [thread1] reconnect sessionmgr01:27727 (192.168.10.139)
ok
set07:SECONDARY>

```

运行命令以验证sessionmgr优先级中的更改。


```

set07:SECONDARY> rs.conf()
{
  "_id" : "set07",
  "version" : 4,
  "members" : [
    {
      "_id" : 0,
      "host" : "sessionmgr01:27727",
      "arbiterOnly" : false,
      "buildIndexes" : true,
      "hidden" : false,
      "priority" : 2,
      "tags" : {

    },
    "slaveDelay" : NumberLong(0),
    "votes" : 1
  },
  {
    "_id" : 1,
    "host" : "arbitervip:27727",
    "arbiterOnly" : true,
    "buildIndexes" : true,
    "hidden" : false,
    "priority" : 0,
    "tags" : {

  },
  "slaveDelay" : NumberLong(0),
  "votes" : 1
},
{
  "_id" : 2,
  "host" : "sessionmgr02:27727",
  "arbiterOnly" : false,
  "buildIndexes" : true,
  "hidden" : false,
  "priority" : 5, --> Here priority has been changed from 1 to 5.
  "tags" : {

},
"slaveDelay" : NumberLong(0),
"votes" : 1
}
],
"settings" : {
  "chainingAllowed" : true,
  "heartbeatIntervalMillis" : 2000,
  "heartbeatTimeoutSecs" : 1,
  "electionTimeoutMillis" : 10000,
  "catchUpTimeoutMillis" : -1,
  "catchUpTakeoverDelayMillis" : 30000,
  "getLastErrorModes" : {

},
  "getLastErrorDefaults" : {
    "w" : 1,
    "wtimeout" : 0
  },
  "replicaSetId" : ObjectId("61cdb17a80b097a2e7604c97")
}
}
set07:SECONDARY>

```

从ClusterManager或pcrfclient运行diagnostics.sh —get_r命令，以验证副本集状态中的更改。

```
| 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 - 14 sec - 2 |  
| Member-2 - 27727 : 192.168.10.146 - ARBITER - arbitervip - ON-LINE - ----- - 0 |  
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - ----- - 5 |  
|-----|  
|
```

现在，您可以看到sessionmgr02已再次成为主。

选项2.更改其他辅助会话管理器的优先级，使其成为主成员。这是sessionmgr01。

要使sessionmgr01成为主成员，请在步骤4中使用此命令运行选项1.中的上述步骤1.到5.。

cfg.members[0].priority =大于3但小于1001的任意数字 —>优先级应高于示例中“3”的当前主成员的优先级。

```
set07:PRIMARY> cfg.members[0].priority = 4  
4  
set07:PRIMARY> rs.reconfig(cfg)  
{  
"ok" : 1,  
"operationTime" : Timestamp(1641540587, 1),  
"$clusterTime" : {  
"clusterTime" : Timestamp(1641540587, 1),  
"signature" : {  
"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),  
"keyId" : NumberLong(0)  
}  
}  
}  
2022-01-07T07:29:46.141+0000 I NETWORK [thread1] trying reconnect to sessionmgr02:27727  
(192.168.10.140) failed  
2022-01-07T07:29:46.142+0000 I NETWORK [thread1] reconnect sessionmgr02:27727 (192.168.10.140)  
ok  
set07:SECONDARY>
```

运行命令以确认更改。

```
set07:SECONDARY> rs.conf()  
{  
"_id" : "set07",  
"version" : 4,  
"members" : [  
{  
"_id" : 0,  
"host" : "sessionmgr01:27727",  
"arbiterOnly" : false,  
"buildIndexes" : true,  
"hidden" : false,  
"priority" : 4, --> Here priority has been changed from 2 to 4.  
"tags" : {  
  
},  
}
```

```

"slaveDelay" : NumberLong(0),
"votes" : 1
},
{
  "_id" : 1,
  "host" : "arbitervip:27727",
  "arbiterOnly" : true,
  "buildIndexes" : true,
  "hidden" : false,
  "priority" : 0,
  "tags" : {

  },
  "slaveDelay" : NumberLong(0),
  "votes" : 1
},
{
  "_id" : 2,
  "host" : "sessionmgr02:27727",
  "arbiterOnly" : false,
  "buildIndexes" : true,
  "hidden" : false,
  "priority" : 3,
  "tags" : {

  },
  "slaveDelay" : NumberLong(0),
  "votes" : 1
}
],
"settings" : {
  "chainingAllowed" : true,
  "heartbeatIntervalMillis" : 2000,
  "heartbeatTimeoutSecs" : 1,
  "electionTimeoutMillis" : 10000,
  "catchUpTimeoutMillis" : -1,
  "catchUpTakeoverDelayMillis" : 30000,
  "getLastErrorModes" : {

  },
  "getLastErrorDefaults" : {
    "w" : 1,
    "wtimeout" : 0
  },
  "replicaSetId" : ObjectId("61cdb17a80b097a2e7604c97")
}
}
set07:SECONDARY>

```

从Cluster Manager或pcrfcleint运行diagnostics.sh —get_r命令，以验证副本集状态中的更改。

```

| 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 : - PRIMARY - sessionmgr01 - ON-LINE - ----- - 4 |
| Member-2 - 27727 : 192.168.10.146 - ARBITER - arbitervip - ON-LINE - ----- - 0 |
| Member-3 - 27727 : - SECONDARY - sessionmgr02 - ON-LINE - 0 sec - 3 |
|-----|
|

```

现在，您可以看到sessionmgr01已成为主，而sessionmgr02成为辅助。

要使sessionmgr02再次成为主成员，请在步骤4中使用此命令运行选项1中的上述步骤1.到5。

cfg.members[0].priority = 小于3但大于0的任意数字 —>将优先级设置为低于示例中为"3"的sessionmgr02的优先级。

```
set07:PRIMARY> cfg.members[0].priority = 1
1
set07:PRIMARY> rs.reconfig(cfg)
{
  "ok" : 1,
  "operationTime" : Timestamp(1641531450, 1),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1641531450, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}
2022-01-07T08:34:31.165+0000 I NETWORK [thread1] trying reconnect to sessionmgr01:27727
(192.168.10.139) failed
2022-01-07T08:34:31.165+0000 I NETWORK [thread1] reconnect sessionmgr01:27727 (192.168.10.139)
ok
set07:SECONDARY>
```

运行此命令以验证sessionmgr优先级中的更改。

```
set07:SECONDARY> rs.conf()
{
  "_id" : "set07",
  "version" : 4,
  "members" : [
    {
      "_id" : 0,
      "host" : "sessionmgr01:27727",
      "arbiterOnly" : false,
      "buildIndexes" : true,
      "hidden" : false,
      "priority" : 1, --> Here priority has been changed from 4 to 1.
      "tags" : {
      },
      "slaveDelay" : NumberLong(0),
      "votes" : 1
    },
    {
      "_id" : 1,
      "host" : "arbiterVIP:27727",
      "arbiterOnly" : true,
      "buildIndexes" : true,
      "hidden" : false,
      "priority" : 0,
      "tags" : {
      },
      "slaveDelay" : NumberLong(0),
      "votes" : 1
    }
  ]
}
```

```

"_id" : 2,
"host" : "sessionmgr02:27727",
"arbiterOnly" : false,
"buildIndexes" : true,
"hidden" : false,
"priority" : 3,
"tags" : {
},
"slaveDelay" : NumberLong(0),
"votes" : 1
},
"settings" : {
"chainingAllowed" : true,
"heartbeatIntervalMillis" : 2000,
"heartbeatTimeoutSecs" : 1,
"electionTimeoutMillis" : 10000,
"catchUpTimeoutMillis" : -1,
"catchUpTakeoverDelayMillis" : 30000,
"getLastErrorModes" : {
},
"getLastErrorDefaults" : {
"w" : 1,
"wtimeout" : 0
},
"replicaSetId" : ObjectId("61cdb17a80b097a2e7604c97")
}
set07:SECONDARY>

```

从ClusterManager或pcrfclient运行diagnostics.sh —get_r命令，以验证副本集状态中的更改。

```

| 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 - 14 sec - 1 |
| Member-2 - 27727 : 192.168.10.146 - ARBITER - arbitervip - ON-LINE - ----- - 0 |
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - ----- - 3 |
|-----|

```

现在，您可以看到sessionmgr02已成为主，而sessionmgr01是辅助。

方法2

可以使用ClusterManager中的CPS脚本set_priority.sh更改副本集中的sessionmgr优先级。默认情况下，按顺序（具有更高优先级）设置成员的优先级，如ClusterManager中/etc/broadhop/mongoConfig.cfg中所定义。

以set07为例。

```

[root@installer broadhop]# cat mongoConfig.cfg
[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]
```

要获取副本集状态，请从ClusterManager或pcrfclient运行命令diagnostics.sh —get_r。

```
| 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 : 192.168.10.146 - ARBITER - arbitervip - ON-LINE - ----- - 0 |
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - ----- - 3 |
|-----|
```

在比较上述结果时，您可以看到sessionmgr02是/etc/broadhop/mongoConfig.cfg中set07的第1个成员[MEMBER1]，因此，sessionmgr02默认情况下是set07中的主成员。

此处提供了CPS高可用性选项，这些选项使用set_priority.sh脚本将sessionmgr02从set07中的主成员角色中移出。

步骤1.按升序设置优先级。

Command template:

```
sh set_priority.sh --db arg --replSet arg --asc
```

where ,

--db arg --> arg is database name

[all|session|spr|admin|balance|report|portal|audit|bindings|session_configs|bindings_configs|spr_configs]

--replSet arg -->arg is <setname>

Sample command:

```
sh set_priority.sh --db session --replSet set07 --asc
```

```
[root@installer ~]# sh set_priority.sh --db session --replSet set07 --asc
```

Set priorities is in progress. check log /var/log/broadhop/scripts/set_priority.log to know the status

```
Setting priority for Replica-Set: SESSION-SET2
```

```
INFO Parsing Mongo Config file
```

```
INFO Priority set operation is completed for SESSION-SET2
```

```
INFO Priority set to the Database members is finished
```

```
INFO Validating if Priority is set correctly for Replica-Set: SESSION-SET2
```

```
WARNING Mongo Server trying to reconnect while getting config. Attempt #1
```

```
INFO Validated Priority is set correctly for Replica-Set: SESSION-SET2
```

```
Primary member sessionmgr01:27727 found for Replica SESSION-SET2
```

Set priorities process successfully completed.

```
[root@installer ~]#
```

步骤2.从ClusterManager或pcrfclient运行命令diagnostics.sh —get_r以验证更改。

```

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

```

现在，sessionmgr01已成为主要成员，因为优先级已按照/etc/broadhop/mongoConfig.cfg中定义的升序设置。

要使sessionmgr02再次成为主成员，请运行此命令。

```
[root@installer ~]# sh set_priority.sh --db session --replSet set07
```

```
Set priorities is in progress. check log /var/log/broadhop/scripts/set_priority.log to know the status
```

```
Setting priority for Replica-Set: SESSION-SET2
INFO Parsing Mongo Config file
INFO Priority set operation is completed for SESSION-SET2
INFO Priority set to the Database members is finished
INFO Validating if Priority is set correctly for Replica-Set: SESSION-SET2
WARNING Mongo Server trying to reconnect while getting config. Attempt #1
INFO Validated Priority is set correctly for Replica-Set: SESSION-SET2
Primary member sessionmgr02:27727 found for Replica SESSION-SET2
```

```
Set priorities process successfully completed.
```

```
[root@installer ~]#
```

注意：默认情况下，优先级按降序设置。

从ClusterManager或pcrfclient运行命令diagnostics.sh —get_r以验证更改。

```

| SET NAME - PORT : IP ADDRESS - REPLICAS 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 |
|-----|

```

现在，您可以看到sessionmgr02已成为主，而sessionmgr01是辅助。