

# CPSレプリカ・セットのセッション・マネージャの役割と優先度を処理する手順

## 内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[問題](#)

[レプリカ・セットのsessionmgrをプライマリから移動し、sessionmgrの優先度を変更する手順](#)

[アプローチ1](#)

[アプローチ2](#)

## 概要

このドキュメントでは、Cisco Policy Suite(CPS)レプリカセットでsessionmgrをプライマリロールから移動し、sessionmgrの優先度を変更する手順について説明します。

## 前提条件

### 要件

次の項目に関する知識があることが推奨されます。

- Linux
- CPS
- MongoDB

シスコでは、CPS CLIへの特権Rootアクセス権が必要であることを推奨します。

### 使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- CPS 20.2
- MongoDB v3.6.17
- UCS-B

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

# 背景説明

CPSは、Sessionmgr仮想マシン(VM)上でmongodプロセスが実行されるMongoDBを使用して、その基本的なデータベース構造を構成します。ADMIN、SPR(Subscriber Profile Repository)、BALANCE、SESSION、REPORTING、AUDITなど、さまざまな目的に対して複数のレプリカセットが用意されています。

MongoDBのレプリカセットは、同じデータセットを維持するmongodプロセスのグループです。レプリカセットは、冗長性と高可用性を提供します。異なるデータベースサーバ上のデータの複数のコピーを使用すると、ロードシェアリード操作が可能になります。

レプリカセットには、データを保持する複数のノードと、オプションで1つのアービタノードが含まれます。データを保持するノードのうち、1つのメンバのみがプライマリノードと見なされ、他のノードはセカンダリノードと見なされます(レプリカセットには複数のセカンダリを含めることができます)。プライマリノードは、すべての書き込み操作を処理します。

セカンダリはプライマリの運用ログ(oplog)を複製し、セカンダリのデータセットがプライマリのデータセットを反映するように、そのデータセットに運用を適用します。プライマリが使用できない場合、対象セカンダリは自身を新しいプライマリに選出するための選択を保持します。アービタは選挙に参加するが、データを保持しない。

レプリカセットのステータスを取得するには、ClusterManagerまたはpcrfclientから`diagnostics.sh --get_r`コマンドを実行します。

ここでは、レプリカセットの例を示します。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 |
|-----|
|
```

レプリカセットの設定情報を取得するには、次の手順を使用します。

ステップ1: そのレプリカセットのプライマリMongoDBメンバーにログインします。ClusterManagerからこのコマンドを実行します。

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

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

ステップ2: コマンドを実行して、レプリカセット構成情報を取得します。

```
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：レプリカセットの優先順位が最も高いセッションマネージャは、プライマリメンバーとして機能します。**

## 問題

1つ以上のレプリカセットでsessionmgrがプライマリメンバーの役割を実行すると仮定し、その場合はReplica Setプライマリロールを他のsessionmgrに移動する必要があります。

- 1.このsessionmgr VMのシャットダウンに関連するアクティビティを実行するたびに、スムーズな移行が可能になります。
- 2.何らかの理由でsessionmgrの正常性が低下した場合、他の正常なsessionmgrと共にレプリカセットの適切な機能を維持します。

## レプリカ・セットのsessionmgrをプライマリから移動し、sessionmgrの優先度を変更する手順

### アプローチ1

ここでは、レプリカセットのsessionmgrの優先度がMongoDBレベルで直接変更されました。set07のプライマリロールからsessionmg02を移動する手順を次に示します。

オプション1. sessionmgr02の優先順位を変更します。

ステップ1：そのレプリカセットのプライマリMongoDBメンバーにログインします。

Command template:

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

Sample command:

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

ステップ2：コマンドを実行して、レプリカ・セットの構成情報を取得します。

```
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**：rs.conf()出力の各sessionmgrの位置をメモしておきます。

**ステップ3**：このコマンドを実行して、端末をコンフィギュレーションモードに移行します。

```

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>

```

ステップ4：このコマンドを実行して、sessionmgrの優先度を変更します。

Command template:

```
cfg.members[X].priority = X --> put the position here in [].
```

sample command:

```
cfg.members[2].priority = 1
```

ここで、sessionmgr02は現在プライマリメンバで、その位置は2、優先度は3です。

このsessionmgr02をプライマリロールから移動するには、0より大きく、優先度が最も高いセカンダリメンバの優先度より小さい最も低いプライオリティ番号を指定します(例)。1を入力します。

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

ステップ5：このコマンドを実行して、変更を確定します。

```
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>
```

ステップ6：コマンドを再度実行して、sessionmgrの優先度の変更を確認します。

```
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>

```

ステップ7:ClusterManagerまたはpcrfclientからコマンドdiagnostics.sh —get\_rを実行し、レプリカセットステータスの変更を確認します。

```

| 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 |
|-----|

```

これで、sessionmgr02がセカンダリに移動しました。sessionmgr02を再びプライマリメンバーにするには、ステップ4で上記のステップ1. ~ 5.を実行します。

cfg.members[2].priority = 2より大きく1001より小さい任意の数値 ->は、サンプル内の現在のプライマリメンバのプライオリティよりも高いプライオリティを設定します。

```

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,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"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>
```

コマンドを実行して、sessionmgrの優先度の変更を確認します。

```
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>
```

ClusterManagerまたはpcrfclientからdiagnostics.sh —get\_rコマンドを実行し、レプリカセットステータスの変更を確認します。

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

これで、sessionmgr02が再びプライマリになったことがわかります。

オプション2.他のセカンダリセッションmgrの優先順位を変更して、プライマリメンバーにします。はいsessionmgr01です。

sessionmgr01をプライマリメンバーにするには、手順4で次のコマンドを使用して、オプション1.で前述の手順1. ~ 5.を実行します。

cfg.members[0].priority = 3より大きく1001より小さい任意の数 ->サンプルの「3」である現在のプライマリメンバーよりもプライオリティを高くします。

```
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,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"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>
```

コマンドを実行して、変更を確認します。

```
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>
```

Cluster Managerまたはpcrfcleintからdiagnostics.sh —get\_rコマンドを実行し、レプリカ・セットのステータスの変更を確認します。

```
| 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 |
|-----|
```

これで、sessionmgr01がプライマリになり、sessionmgr02がセカンダリになったことがわかります。

sessionmgr02を再びプライマリメンバーにするには、ステップ4で次のコマンドを使用して、オプション1で前述のステップ1から5を実行します。

cfg.members[0].priority = 3未満の任意の数値、0より大きい ->sessionmgr02のプライオリティより低い値を設定します。この値はサンプルの「3」です。

```
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>
```

このコマンドを実行して、sessionmgrの優先度の変更を確認します。

```
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>

```

ClusterManagerまたはpcrfclientからdiagnostics.sh —get\_rコマンドを実行し、レプリカセットステータスの変更を確認します。

```

| 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 |
```

```
|-----|
```

これで、sessionmgr02がプライマリであるのに対し、sessionmgr01はセカンダリであることがわかります。

## アプローチ2

ClusterManagerのCPSスクリプトset\_priority.shを使用して、レプリカセットのsessionmgrの優先度を変更できます。デフォルトでは、ClusterManagerの/etc/broadhop/mongoConfig.cfgで定義されているように、メンバーの優先度が順に設定され(高い優先度)、

たとえば、set07とします。

```
[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]
```

Replica setのステータスを取得するには、ClusterManagerまたはpcrfclientからdiagnostics.sh —get\_rコマンドを実行します。

```
| 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 |
```

```
|-----|
```

上記の結果を比較すると、sessionmgr02が/etc/broadhop/mongoConfig.cfgのset07の1番目のメンバー[MEMBER1]であることがわかります。したがって、sessionmgr02がデフォルトでset07のプライマリメンバーになります。

ここでは、set\_priority.shスクリプトを使用してsessionmgr02をset07のプライマリメンバーのロールから移動するCPSハイアベイラビリティオプションを示します。

ステップ1：優先順位を昇順に設定します。

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 ~]#
```

**ステップ2:ClusterManagerまたはpcrfclientからdiagnostics.sh —get\_rコマンドを実行し、変更を確認します。**

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

これで、sessionmgr01がプライマリメンバーになりました。/etc/broadhop/mongoConfig.cfgで定義されているように、優先順位が昇順に設定されました。

sessionmgr02をプライマリメンバにするには、次のコマンドを実行します。

```
[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：デフォルトでは、優先順位は降順に設定されています。**

ClusterManagerまたはpcrfclientからdiagnostics.sh —get\_rコマンドを実行し、変更を確認します

。

```
| 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 |
|-----|
|-----|
```

これで、sessionmgr02がプライマリであるのに対し、sessionmgr01はセカンダリであることがわかります。