

# Procedure om rol en prioriteit van sessiemanagers in een CPS-replicatieset aan te pakken

## Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Achtergrondinformatie](#)

[Probleem](#)

[Procedure om sessionhouder te verplaatsen van Primaire prioriteit en wijzigingssessieprioriteit in een replica-set](#)

[Aanpak 1](#)

[Aanpak 2](#)

## Inleiding

Dit document beschrijft de procedure om sessionhouder van de primaire rol te verplaatsen en de sessieprioriteit te wijzigen in een Cisco Policy Suite (CPS) replica-set.

## Voorwaarden

### Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- Linux
- CPS
- MongoDB

Cisco raadt u aan om voorkeurstoegang tot CPS CLI te hebben.

### Gebruikte componenten

De informatie in dit document is gebaseerd op de volgende software- en hardware-versies:

- CPS 2002
- MongoDB v3.6.17
- UCS-B

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk levend is, zorg er dan voor dat u de

mogelijke impact van om het even welke opdracht begrijpt.

## Achtergrondinformatie

CPS gebruikt MongoDB waar mongoprocessen op Sessionger Virtual Machine (VMs) lopen om er een basisgegevensstructuur van te maken. Er zijn meerdere replica-sets voor verschillende doeleinden, zoals ADMIN, Subscriber Profile Repository (SPR), BALANCE, SESSION, REPORTING en AUDIT.

Een replica-set in MongoDB is een groep mongoprocessen die dezelfde gegevensset onderhouden. Kopieersets bieden redundantie en hoge beschikbaarheid. Met meerdere exemplaren van gegevens op verschillende gegevensservers maakt het laden van opgelezen bewerkingen mogelijk.

Een replica-set bevat verschillende knooppunten met gegevens en optioneel één 'arbiter'-knooppunt. Van de knooppunten die gegevens bevatten, wordt één en slechts één lid geacht het primaire knooppunt, terwijl de andere knooppunten als secundaire knooppunten worden beschouwd (een replica-set kan meerdere seconden hebben). Het primaire knooppunt verwerkt alle schrijfbewerkingen.

De gedetacheerde personen repliceren de Operatie logs van de primaire deelnemer (oplog) en passen de activiteiten toe op hun datasets, zodat de datasets van de gedetacheerden de gegevens van de primaire eenheid weergeven. Als de primaire functie niet beschikbaar is, heeft een daarvoor in aanmerking komend secundair selectie om zichzelf de nieuwe primaire taak te kiezen. Een scheidsrechter neemt deel aan verkiezingen, maar heeft geen gegevens.

Om reproducties status te krijgen, voert u de opdracht **diagnostiek.sh --get\_r** uit ClusterManager of PCclient.

Als hier een monster van een replica set is. **set07**.

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

Gebruik deze stappen om informatie over de configuratie van de Kopie te verkrijgen.

Stap 1. Meld u aan bij het primaire MongoDB-lid van die replica-set. Start deze opdracht van Cluster Manager.

Command template:

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

Sample command:

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

## Stap 2. Start de opdracht om informatie over de configuratie van de replica-instelling te verkrijgen.

```
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>
```

**Opmerking:** Sessionmgr met de hoogste prioriteit in een replica Set functie als primair lid.

## Probleem

Stel dat een sessionhouder de rol van een primair lid in een of meer reproductiesets vervult en in deze gevallen moet u reproductiefunctie overhevelen naar een andere sessie.

1. Telkens wanneer u een activiteit uitvoert waarbij deze sessie wordt beëindigd of VM wordt uitgeschakeld, voor een soepele overgang.

2. Als de gezondheid van de sessionhouder om een of andere reden wordt gedegradeerd, moet de correcte functie van de replica-set met een andere gezonde sessie worden gehandhaafd.

## Procedure om sessionhouder te verplaatsen van Primaire prioriteit en wijzigingssessieprioriteit in een replica-set

### Aanpak 1

Hier veranderde de prioriteit van sessionhouder in een replica-set direct op het niveau van MongoDB. Hier zijn de stappen om sessionmgr02 van een primaire rol in **set07** te verplaatsen.

Optie 1. Verander de prioriteit van sessionhouder02.

Stap 1. Meld u aan bij het primaire MongoDB-lid van die replica-set.

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

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

Stap 2. Start de opdracht om informatie over de configuratie van de replica-instelling te verkrijgen.

```
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>

```

**Opmerking:** Neem nota van de positie van de respectieve sessionhouder in de rs.conf() output.

Stap 3. Start deze opdracht om de terminal naar de configuratie-modus te verplaatsen.

```

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>

```

**Stap 4. Start deze opdracht om de prioriteit van de sessie te wijzigen.**

Command template:

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

sample command:

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

Hier is sessionmgr02 het eerste lid en het standpunt van de partij is 2 en prioriteit is 3.

Om deze sessie af te zetten van de primaire rol, geef het laagste prioriteitsnummer dat groter is dan 0, maar minder dan de prioriteit van een secundair lid dat de hoogste prioriteit heeft, bijv. 1 in deze opdracht.

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

Stap 5. Start deze opdracht om de wijziging aan te geven.

```
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>
```

Stap 6. Start de opdracht opnieuw om de wijzigingen in de sessieprioriteit te controleren.

```
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>

```

**Stap 7. Start de opdracht `diagnostiek.sh —get_r` van ClusterManager of PCclient om wijzigingen in replica Set-status te controleren.**

```

| 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 |
|-----|
|-----|

```

Nu zie je dat sessionhouder02 naar het secundair is verplaatst. Om van sessionhouder02 opnieuw een primair lid te maken, voert u de bovengenoemde stappen 1.0 tot 5.0 uit met deze opdracht in stap 4.

`cfg leden[2].prioriteit = Een aantal groter dan 2 maar minder dan 1001 —> een hogere prioriteit dan die van het huidige primaire lid, dat 2 in de steekproef is.`



```

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>

```

**Start de opdracht om de wijzigingen in sessionprioriteit te controleren.**

```

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" : {

```



```
}
}
}
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>
```

**Start de opdracht om de wijzigingen te bevestigen.**

```
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>

```

Start de opdracht **diagnostics.sh —get\_r** van Cluster Manager of Pict om wijzigingen in replica Set-status te controleren.

```

| 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 |
|-----|

```

Je ziet dat sessionmgr01 primair is geworden, terwijl sessionmgr02 secundair werd.

Om sessionhouder02 opnieuw als primair lid te kunnen benoemen, voert u de bovengenoemde stappen 1. t/m 5. in **optie 1** met deze opdracht in stap 4 uit.

cfg leden[0].prioriteit = Een aantal van minder dan 3 maar groter dan 0 —> Plaats prioriteit lager dan dat van sessionmgr02, dat "3" is in de steekproef.

```

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>

```

Start deze opdracht om de wijzigingen in sessionprioriteit te controleren.

```

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>

```

Start de opdracht **diagnostics.sh —get\_r** van ClusterManager of PCclient om wijzigingen in replica Set-status te controleren.

```

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

```

Nu zie je dat sessionmgr02 primair is geworden, terwijl sessionmgr01 secundair is.

## Aanpak 2

U kunt CPS script **set\_Priority.sh** gebruiken vanuit ClusterManager om de sessionprioriteit in een replica-set te wijzigen. Standaard is de prioriteit van leden in volgorde ingesteld (met hogere prioriteit) zoals gedefinieerd in **/etc/broadhop/mongoConfig.cfg** in ClusterManager.

Neem bijvoorbeeld set07.

```

[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]

```

Om reproductie ingestelde status te krijgen, voer de opdracht **diagnostiek.sh --get\_r** van ClusterManager of PCclient uit.

```

| 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 |
|-----|

```

Wanneer u de bovengenoemde resultaten vergelijkt, kunt u zien dat sessionmgr02 het eerste lid[LID1] van set07 is in **/etc/broadhop/mongoConfig.cfg**, dus sessionmgr02 is standaard het primaire lid in set07.

Hiervoor zijn de CPS High Availability opties beschikbaar die **set\_Priority.sh** script gebruiken om de sessionhouder02 uit de functie van het primaire lid in set07 te verplaatsen.

Stap 1. Stel de prioriteit in oplopende volgorde in.

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 ~]#
```

**Stap 2. Start de opdracht `diagnostiek.sh --get_r` van ClusterManager of PCclient om de wijziging te controleren.**

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

Nu is sessionmgr01 het primaire lid geworden aangezien een prioriteit is ingesteld in de oplopende volgorde zoals gedefinieerd in `/etc/broadhop/mongoConfig.cfg`.

Om van sessionmgr02 opnieuw een primair lid te maken, voert u deze opdracht uit.

```
[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 ~]#

**Opmerking:** Standaard is de prioriteit in aflopende volgorde ingesteld.

Start de opdracht **diagnostics.sh —get\_r** van ClusterManager of PCclient om de wijziging te controleren.

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

Nu zie je dat sessionmgr02 primair is geworden, terwijl sessionmgr01 secundair is.