

# Procedura per la gestione del ruolo e della priorità dei gestori delle sessioni in un set di repliche CPS

## Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Problema](#)

[Procedura per spostare sessionmgr dalla priorità primaria e modificare la priorità di sessionmgr in un set di repliche](#)

[Approccio 1](#)

[Approccio 2](#)

## Introduzione

In questo documento viene descritta la procedura per spostare sessionmgr dal ruolo primario e modificare la priorità di sessionmgr in un set di repliche di Cisco Policy Suite (CPS).

## Prerequisiti

### Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Linux
- CPS
- MongoDB

Cisco consiglia di disporre del privilegio di accesso root alla CLI di CPS.

### Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

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

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata

ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

## Premesse

CPS utilizza MongoDB dove i processi mongood vengono eseguiti su macchine virtuali (VM) di Sessionmgr per costituire la struttura di base del database. Possiede più set di repliche per vari scopi, ovvero ADMIN, Subscriber Profile Repository (SPR), BALANCE, SESSION, REPORTING e AUDIT.

Un set di repliche in MongoDB è un gruppo di processi mondani che gestiscono lo stesso set di dati. I set di repliche forniscono ridondanza ed elevata disponibilità. Con più copie di dati su server di database diversi, consente operazioni di lettura con condivisione del carico.

Un set di repliche contiene diversi nodi che contengono dati e, facoltativamente, un nodo arbitro. Tra i nodi che contengono dati, un solo membro viene considerato il nodo primario, mentre gli altri nodi vengono considerati nodi secondari (un set di repliche può avere più nodi secondari). Il nodo primario gestisce tutte le operazioni di scrittura.

I database secondari replicano i registri operativi (oplog) del database primario e applicano le operazioni ai relativi set di dati in modo che i set di dati dei database secondari riflettano il set di dati del database primario. Se il database primario non è disponibile, un database secondario idoneo esegue una selezione per selezionare se stesso come nuovo database primario. Un arbitro partecipa alle elezioni ma non possiede dati.

Per ottenere lo stato dei set di replica, eseguire il comando **diagnostics.sh --get\_r** da ClusterManager o pcrfclient.

Viene fornito un esempio di set di repliche. **set07**.

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

Per ottenere informazioni sulla configurazione del set di repliche, eseguire la procedura seguente.

Passaggio 1. Accedere al membro MongoDB primario del set di repliche. Eseguire questo comando da ClusterManager.

Command template:

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

Sample command:

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

Passaggio 2. Eseguire il comando per ottenere le informazioni di configurazione del set di

repliche.

```
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>
```

**Nota:** Sessionmgr con la priorità più alta in un set di repliche funziona come membro primario.

## Problema

Si supponga che sessionmgr svolga il ruolo di membro primario in uno o più set di repliche e in questi casi sia necessario spostare il ruolo primario del set di repliche in un altro sessionmgr.

1. Ogni volta che si esegue un'attività che comporta questa sessione di gestione delle macchine virtuali, per una transizione senza problemi.
2. Se l'integrità di sessionmgr subisce un peggioramento per qualche motivo, per mantenere il corretto funzionamento del set di repliche con altri sessionmgr integri.

## Procedura per spostare sessionmgr dalla priorità primaria e modificare la priorità di sessionmgr in un set di repliche

### Approccio 1

La priorità di sessionmgr in un set di repliche è cambiata direttamente a livello di MongoDB. Di seguito sono riportati i passaggi per spostare sessionmgr02 da un ruolo primario in **set07**.

Opzione 1. Modificare la priorità di sessionmgr02.

Passaggio 1. Accedere al membro MongoDB primario del set di repliche.

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

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

Passaggio 2. Eseguire il comando per ottenere le informazioni di configurazione del set di repliche.

```
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>

```

**Nota:** Prendere nota della posizione di sessionmgr nell'output rs.conf().

Passaggio 3. Eseguire questo comando per spostare il terminale in modalità di configurazione.

```

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>

```

**Passaggio 4.** Eseguire questo comando per modificare la priorità di sessionmgr.

Command template:

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

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

In questo caso, session mgr02 è attualmente il membro principale e la sua posizione è 2 e la priorità è 3.

Per spostare questa sessione mgr02 fuori dal ruolo primario, fornire il numero di priorità più basso maggiore di 0 ma minore della priorità di un membro secondario con la priorità più alta, ad esempio. 1 in questo comando.

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

Passaggio 5. Eseguire questo comando per eseguire il commit della modifica.

```
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>
```

Passaggio 6. Eseguire nuovamente il comando per verificare le modifiche nella priorità sessionmgr.

```
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>

```

**Passaggio 7.** Eseguire il comando **diagnostics.sh —get\_r** da ClusterManager o pcrfclient per verificare le modifiche nello stato del set di repliche.

```

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

```

Si noti che sessionmgr02 è stato spostato in posizione secondaria. Per rendere sessionmgr02 di nuovo membro primario, eseguire i passaggi da 1 a 5 sopra indicati con questo comando al passaggio 4.



cfg.members[2].priority = Qualsiasi numero maggiore di 2 ma minore di 1001 —> aumenta la priorità rispetto a quella del membro primario corrente che è 2 nel campione.

```
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>
```

Eseguire il comando per verificare le modifiche nella priorità sessionmgr.

```
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>

```

Eseguire il comando **diagnostics.sh —get\_r** da ClusterManager o pcrfclient per verificare le modifiche nello stato del set di repliche.

```

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

```

Si noti che sessionmgr02 è diventato nuovamente primario.

Opzione 2. Modificare la priorità di un altro membro secondario di gestione sessione in modo da renderlo un membro primario. Ecco sessionmgr01.

Per rendere sessionmgr01 un membro primario, eseguire i passaggi da 1 a 5 descritti in Opzione 1. con questo comando al Passaggio 4.

cfg.members[0].priority = Qualsiasi numero maggiore di 3 ma minore di 1001 —> Mettere la priorità più alta di quella del membro primario corrente che è "3" nel campione.

```

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>
```

**Eseguire il comando per confermare le modifiche.**

```
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>

```

Eseguire il comando **diagnostics.sh —get\_r** da Gestione cluster o pcrfclient per verificare le modifiche nello stato del set di repliche.

```

| 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 |
|-----|
|

```

Si noti che sessionmgr01 è diventato primario, mentre sessionmgr02 è diventato secondario.

Per rendere sessionmgr02 di nuovo membro primario, eseguire i passaggi da 1 a 5 descritti in precedenza nell'opzione 1 con questo comando nel passaggio 4.

cfg.members[0].priority = Qualsiasi numero minore di 3 ma maggiore di 0 —> Mettere la priorità più bassa di quella di sessionmgr02 che è "3" nel campione.

```

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>

```

Eseguire questo comando per verificare le modifiche nella priorità sessionmgr.

```
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>
```

Eeguire il comando **diagnostics.sh —get\_r** da ClusterManager o pcrfclient per verificare le modifiche nello stato del set di repliche.

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

Si noti che sessionmgr02 è diventato primario, mentre sessionmgr01 è secondario.

## Approccio 2

È possibile utilizzare lo script CPS **set\_priority.sh** da ClusterManager per modificare la priorità sessionmgr in un set di repliche. Per impostazione predefinita, la priorità dei membri è impostata in ordine (con priorità più alta) come definito in **/etc/broadhop/mongoConfig.cfg** in ClusterManager.

Prendiamo ad esempio 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]
```

Per ottenere lo stato del set di repliche, eseguire il comando **diagnostics.sh —get\_r** da ClusterManager o pcrfclient.

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

Se si confrontano i risultati sopra indicati, è possibile notare che sessionmgr02 è il primo membro[MEMBER1] di set07 in **/etc/broadhop/mongoConfig.cfg**, pertanto sessionmgr02 è il membro principale in set07 per impostazione predefinita.

Di seguito vengono fornite le opzioni di alta disponibilità di CPS che utilizzano lo script **set\_priority.sh** per spostare sessionmgr02 dal ruolo di membro primario in set07.

## Passaggio 1. Impostare la priorità in ordine crescente.

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 ~]#
```

## Passaggio 2. Eseguire il comando **diagnostics.sh --get\_r** da ClusterManager o pcrclient per verificare la modifica.

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

sessionmgr01 è diventato il membro primario in quanto è stata impostata una priorità in ordine crescente, come definito in **/etc/broadhop/mongoConfig.cfg**.

Per rendere di nuovo membro primario sessionmgr02, eseguire questo comando.

```
[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 ~]#
```

**Nota:** Per impostazione predefinita, la priorità è stata impostata in ordine decrescente.

Eseguire il comando **diagnostics.sh --get\_r** da ClusterManager o pcrfclient per verificare la modifica.

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

Si noti che sessionmgr02 è diventato primario, mentre sessionmgr01 è secondario.