Verfahren zum Behandeln der Rolle und Priorität von Sitzungsmanagern in einem CPS-Replikatsatz

Inhalt

Einleitung

In diesem Dokument wird das Verfahren zum Verschieben von sessionmgr aus der Hauptrolle und zum Ändern der Sessionmgr-Priorität in einem Replikationssatz der Cisco Policy Suite (CPS) beschrieben.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

- Linux
- CPS
- MongoDB

Cisco empfiehlt, dass Sie über einen privilegierten Root-Zugriff auf die CPS-CLI verfügen müssen.

Verwendete Komponenten

Die Informationen in diesem Dokument basierend auf folgenden Software- und Hardware-Versionen:

- CPS 20,2
- MongoDB v3.6.17
- UCS B

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten

Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

Hintergrundinformationen

CPS verwendet MongoDB, wo Mongod-Prozesse auf Sessionmgr Virtual Machine (VMs) ausgeführt werden, um die grundlegende Datenbankstruktur zu bilden. Es verfügt über mehrere Replikationssätze für verschiedene Zwecke, z. B. ADMIN, SPR (Subscriber Profile Repository), BALANCE, SESSION, REPORTING und AUDIT.

Ein Replikat-Satz in MongoDB ist eine Gruppe von mongoten Prozessen, die den gleichen Datensatz verwalten. Replikationssätze bieten Redundanz und hohe Verfügbarkeit. Dank mehrerer Kopien von Daten auf verschiedenen Datenbankservern können Lesevorgänge mit Lastverteilung ausgeführt werden.

Ein Replikationssatz enthält mehrere Knoten, die Daten und optional einen Arbiter-Knoten enthalten. Von den Knoten, die Daten tragen, wird ein und nur ein Member als primärer Knoten angesehen, während die anderen Knoten als sekundäre Knoten gelten (ein Replikationssatz kann mehrere Sekunden haben). Der primäre Knoten verarbeitet alle Schreibvorgänge.

Die Sekunden replizieren die Operationsprotokolle des Primärangehörigen (Oplog) und wenden die Operationen auf ihre Datensätze an, sodass die Datensätze der Sekundäreinheiten den Datensatz des Primär widerspiegeln. Wenn die Primär nicht verfügbar ist, hält ein qualifizierter Sekundär eine Wahl ab, um sich selbst die neue Primär zu wählen. Ein Schiedsrichter nimmt an Wahlen teil, verfügt aber nicht über Daten.

Um den Status der Replica-Datensätze abzurufen, führen Sie den Befehl **diagnostics.sh —get_r** aus ClusterManager oder pcrfclient aus.

Hier ist ein Beispiel Replica Set ie. **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 |
|-------------------------------------------------------------------------------------------
----------------------------------------|
```

Um Konfigurationsinformationen für Replica zu erhalten, gehen Sie wie folgt vor.

Schritt 1: Melden Sie sich beim primären MongoDB-Mitglied dieses Replica Set an. Führen Sie diesen Befehl aus ClusterManager aus.

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

Sample command:
```

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

Schritt 2: Führen Sie den Befehl aus, um die Konfigurationsinformationen für Replica Set abzurufen.

```
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>
```

Anmerkung: Sessionmgr mit der höchsten Priorität in einem Replica Set fungiert als primärer Member.

Problem

Angenommen, ein Sitzungsmanager übernimmt die Rolle eines primären Mitglieds in einem oder mehreren Replikationssätzen. In diesem Fall müssen Sie die primäre Rolle Replica Set auf eine andere Sitzungsgruppe verschieben.

1. Wenn Sie eine Aktivität ausführen, die das Herunterfahren des sitzungsmgr VM beinhaltet, um einen reibungslosen Übergang zu ermöglichen.

2. Wenn sitzungsmgr Gesundheit aus einem bestimmten Grund herabgesetzt wird, um die ordnungsgemäße Funktion von Replica Set mit einigen anderen gesunden Sessionmgr beizubehalten.

Verfahren zum Verschieben von sitzungsmgr aus der primären und der Change-SitzungMgr-Priorität in einem Replikationssatz

Ansatz 1

Hier die Priorität von sessionmgr in einem Replica Set geändert direkt auf MongoDB Ebene. Dies sind die Schritte, um sessionmg02 aus einer Hauptrolle in **set07** zu entfernen.

Option 1: Ändern Sie die Priorität von sessionmgr02.

Schritt 1: Melden Sie sich beim primären MongoDB-Mitglied dieses Replica Set an.

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

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

Schritt 2: Führen Sie den Befehl aus, um die Konfigurationsinformationen für Replica Set abzurufen.

```
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>
```

Anmerkung: Notieren Sie sich die Position der jeweiligen sessionmgr in der Ausgabe rs.conf().

Schritt 3: Führen Sie diesen Befehl aus, um das Terminal in den Konfigurationsmodus zu verschieben.

```
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>
```

Schritt 4: Führen Sie diesen Befehl aus, um die Priorität von sessionmgr zu ändern.

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

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

Hier ist sessionmgr02 derzeit das primäre Mitglied und seine Position ist **2** und die Priorität **3**.

Um diese SitzungMGR02 aus der Hauptrolle zu verschieben, geben Sie die niedrigste Prioritätsnummer größer als 0, aber kleiner als die Priorität eines Sekundärmitglieds mit der höchsten Priorität an, z. B. **1** in diesem Befehl.

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

Schritt 5: Führen Sie diesen Befehl aus, um die Änderung zu bestätigen.

```
set07:PRIMARY> rs.reconfig(cfg)
{
"ok" : 1,
"operationTime" : Timestamp(1641528658, 1),
"$clusterTime" : {
"clusterTime" : Timestamp(1641528658, 1),
"signature" : {
"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"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>
```

Schritt 6: Führen Sie den Befehl erneut aus, um die Änderungen in der Priorität sessionmgr zu überprüfen.

```
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>
```

Schritt 7: Führen Sie den Befehl **diagnostics.sh —get_r** von ClusterManager oder pcrfclient aus, um Änderungen im Status Replica Set zu überprüfen.

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

Nun sehen Sie, dass sessionmgr02 zu sekundär geworden ist. Führen Sie die oben genannten Schritte 1 aus, um sessionmgr02 wieder zum primären Mitglied zu machen. zu 5. mit diesem Befehl in Schritt 4.

cfg.members[2].priority = Jede Zahl größer als 2, aber kleiner als 1001 —> legt die Priorität höher als die des aktuellen primären Members, der im Beispiel **2** ist.

```
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,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"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>
```

Führen Sie den Befehl aus, um die Änderungen in der Priorität sessionmgr zu überprüfen.

```
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>
```

Führen Sie den Befehl **diagnostics.sh —get_r** von ClusterManager oder pcrfclient aus, um Änderungen im Status Replica Set zu überprüfen.

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

Nun können Sie sehen, dass sessionmgr02 wieder primär geworden ist.

Option 2: Ändern Sie die Priorität anderer sekundärer Sitzungen, um sie zu einem primären Mitglied zu machen. Hier ist sessionmgr01.

Führen Sie die oben genannten Schritte 1 aus, um sessionmgr01 zum primären Mitglied zu machen. zu 5. in Option 1. mit diesem Befehl in Schritt 4.

cfg.members[0].priority = Jede Zahl größer als 3, aber kleiner als 1001 —> Legen Sie die Priorität höher als die des aktuellen primären Members fest, der "3" in der Stichprobe ist.

```
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,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"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>
```

Führen Sie den Befehl aus, um die Änderungen zu bestätigen.

```
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>
```

Führen Sie den Befehl **diagnostics.sh —get_r** aus Cluster Manager oder pcrfcleint aus, um Änderungen im Replica Set-Status zu überprüfen.

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

Nun können Sie sehen, dass sessionmgr01 primär geworden ist, während sessionmgr02 sekundär wurde.

Um sessionmgr02 wieder als primäres Mitglied zu machen, führen Sie die oben genannten Schritte 1 aus. zu 5. in **Option 1** mit diesem Befehl in Schritt 4.

cfg.members[0].priority = Jede Zahl kleiner als 3, aber größer als 0 —> Sende-Priorität niedriger als die von sessionmgr02, die "3" im Beispiel ist.

```
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,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"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>
```

Führen Sie diesen Befehl aus, um die Änderungen in der Priorität sessionmgr zu überprüfen.

```
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>
```

Führen Sie den Befehl **diagnostics.sh —get_r** von ClusterManager oder pcrfclient aus, um Änderungen im Status Replica Set zu überprüfen.

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

Nun können Sie sehen, dass sessionmgr02 primär geworden ist, während sessionmgr01 sekundär ist.

Ansatz 2

Sie können das CPS-Skript **set_priority.sh** von ClusterManager verwenden, um die Sessionmgr-Priorität in einem Replica-Satz zu ändern. Standardmäßig wird die Priorität der Member in der Reihenfolge (mit höherer Priorität) festgelegt, wie in **/etc/broadhop/mongoConfig.cfg** im ClusterManager definiert.

Nehmen wir z. B. 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]
```

Um den Replica-Set-Status abzurufen, führen Sie den Befehl **diagnostics.sh —get_r** aus ClusterManager oder pcrfclient aus.

```
| 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 |
```

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

Wenn Sie die zuvor genannten Ergebnisse vergleichen, sehen Sie, dass sessionmgr02 das erste Mitglied [MEMBER1] von set07 in **/etc/broadhop/mongoConfig.cfg** ist, daher ist sessionmgr02 standardmäßig das primäre Mitglied in set07.

Hier sind die CPS-Optionen für hohe Verfügbarkeit aufgeführt, die das Skript **set_priority.sh** verwenden, um die sessionmgr02 aus der primären Mitgliedrolle im set07 zu verschieben.

Schritt 1: Legen Sie die Priorität in aufsteigender Reihenfolge fest.

```
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 ~]#
```

Schritt 2: Führen Sie den Befehl **diagnostics.sh —get_r** von ClusterManager oder pcrfclient aus, um die Änderung zu überprüfen.

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

Jetzt ist sessionmgr01 das primäre Mitglied, da die Priorität in der aufsteigenden Reihenfolge festgelegt wurde, wie in **/etc/broadhop/mongoConfig.cfg** definiert.

Um sessionmgr02 erneut zum primären Member zu machen, führen Sie diesen Befehl aus.

```
[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 ~]#
```

Anmerkung: Die Priorität wurde standardmäßig in absteigender Reihenfolge festgelegt.

Führen Sie den Befehl **diagnostics.sh —get_r** von ClusterManager oder pcrfclient aus, um die Änderung zu überprüfen.

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

Nun können Sie sehen, dass sessionmgr02 primär geworden ist, während sessionmgr01 sekundär ist.