Verfahren zum Verwalten von Arbiter-Knoten in CPS-Replikatgruppen

Inhalt

Einleitung

In diesem Dokument wird das Verfahren zum Verwalten des Arbiter-Knotens in der Cisco Policy Suite (CPS)-Replikatgruppe beschrieben.

Voraussetzungen

Anforderungen

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

- Linux
- CPS
- MongoDB

Hinweis: Cisco empfiehlt, dass Sie über Berechtigungen für den 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
- Unified Computing System (UCS)-B
- MongoDB v3.6.17 und v3.4.16

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 möglichen Auswirkungen aller Befehle verstehen.

Hintergrundinformationen

CPS verwendet MongoDB, um seine grundlegende Datenbankstruktur (DB) zu bilden. Es verfügt über mehrere Replikatgruppen für verschiedene Zwecke: ADMIN, Subscriber Profile Repository (SPR), BALANCE, SESSION, REPORTING und AUDIT.

Ein Replikatsatz in MongoDB ist eine Gruppe von mongoden Prozessen, die den gleichen Datensatz verwalten. Replikatgruppen bieten Redundanz und hohe Verfügbarkeit. Bei mehreren Datenkopien auf verschiedenen DB-Servern sind LoadShare-Lesevorgänge möglich.

In einigen Fällen (wie Sie haben eine primäre und eine sekundäre, aber Kostenbeschränkungen verbieten das Hinzufügen einer weiteren sekundären), können Sie wählen, um eine mongode Instanz zu einem Replikat Set als Schiedsrichter bei Wahlen zu wählen. Ein Schiedsrichter hat genau 1 Wahlstimme. Ein Arbiter hat standardmäßig die Priorität 0.

Arbiter sind Mongod-Instanzen, die Teil eines Replikatsatzes sind, aber keine Daten enthalten (d. h., sie bieten keine Datenredundanz). Sie können jedoch an Wahlen teilnehmen. Ein Schiedsrichter nimmt an den Wahlen für den primären teil, aber ein Schiedsrichter hat keine Kopie des Datensatzes und kann nicht zum primären werden.

Schiedsrichter haben minimale Ressourcenanforderungen und benötigen keine dedizierte Hardware. Sie können einen Arbiter auf einem Anwendungsserver oder einem Host bereitstellen, der das Netzwerk lediglich überwacht.

Ein Arbiter speichert keine Daten, aber bis der Arbiter-Mongod-Prozess zum Replikatsatz hinzugefügt wird, verhält sich der Arbiter wie jeder andere Mongod-Prozess und startet mit einem Satz von Datendateien und einem Journal in voller Größe.

Hier ist ein Beispielreplikat, das heißt **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 |
|-------------------------------------------------------------------------------------------------------
---------------------------------------|
```

Problem

Angenommen, es gibt ein Problem mit einem Arbiter oder eine Anforderung, den Arbiter in einem Replikatsatz zu ändern, dann müssen Sie den aktuellen Arbiter entfernen und einen neuen Arbiter zum Replikatsatz hinzufügen.

Verfahren zum Verwalten von Arbitern in einem Replikatsatz

Schritt 1: Überprüfen Sie die Mongo-Shell-Version in CPS und dem neuen Arbiter. Führen Sie diesen Befehl vom primären sessionmgr im Replikatsatz und im neuen Arbiterknoten aus.

Beispielausgabe von sessionmgr:

```
[root@sessionmgr02 ~]# mongo --version
MongoDB shell version v3.6.17
```

Wenn die Mongo-Shell-Version sowohl in Primary sessionmgr als auch in New Arbiter gleich ist oder wenn die neue Arbiter-Mongo-Shell-Version höher ist, navigieren Sie zu Schritt 6.

Andernfalls, wenn die neue Arbiter Mongo Shell-Version niedriger ist, dann müssen Sie setzen **featureCompatibilityVersion** als der niedrigere Wert in der Admin-Datenbank des Replikats mit den nächsten Schritten festgelegt.

Beispiel für den Fall, dass die neue Arbiter-Mongo-Shell-Version niedriger ist als die von CPS sessionmgr:

```
[root@pcrfclient02 ~]# mongo --version
MongoDB shell version v3.4.16
```

Schritt 2: Melden Sie sich bei der primären Mongo-Instanz des Replikatsatzes an.

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

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

Schritt 3: Führen Sie diesen Befehl aus, um die aktuelle **featureCompatibilityVersion** in der Admin-Datenbank der Replikatgruppe.

```
set07:PRIMARY> db.adminCommand( { getParameter: 1, featureCompatibilityVersion: 1 } )
{
"featureCompatibilityVersion" : {
"version" : "3.6"
},
"ok" : 1,
"operationTime" : Timestamp(1663914140, 1),
"$clusterTime" : {
"clusterTime" : Timestamp(1663914140, 1),
"signature" : {
"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"keyId" : NumberLong(0)
}
}
}
set07:PRIMARY>
```

Schritt 4: Führen Sie diesen Befehl aus setfeatureCompatibilityVersion als 3.4 in der Admin-Datenbank des Replikatssatzes.

```
set07:PRIMARY> db.adminCommand( { setFeatureCompatibilityVersion: "3.4" } )
{ "ok" : 1 }
set07:PRIMARY>
```

Schritt 5: Führen Sie diesen Befehl aus, um sicherzustellen, dass **featureCompatibilityVersion** in der Admin-Datenbank des Replikatsatzes auf 3,4 geändert.

```
set07:PRIMARY> db.adminCommand( { getParameter: 1, featureCompatibilityVersion: 1 } )
{ "featureCompatibilityVersion" : { "version" : "3.4" }, "ok" : 1 }
set07:PRIMARY>
```

Schritt 6: Melden Sie sich beim Cluster-Manager an, und ändern Sie die **/var/qps/config/deploy/csv/AdditionalHosts.csv** mit neuen Details zum Schiedsrichter.

```
#vi /var/qps/config/deploy/csv/AdditionalHosts.csv

Provide new arbiter details in this format:

Host Alias IP Address
new-arbiter new-arbiter xx.xx.xx.xx
```

Schritt 7. CSV-Konfiguration importieren

```
#/var/qps/install/current/scripts/import/import_deploy.sh
```

Schritt 8: Überprüfung **/etc/hosts** die mit den Informationen der neuen Schiedsrichter aktualisiert wurden.

```
#cat /etc/hosts | grep arbiter
```

Schritt 9. Führen Sie diesen Befehl zur Synchronisierung aus. **/etc/hosts**.

```
#/var/qps/bin/update/synchosts.sh

Syncing to following QNS Servers:
lb01 lb02 sessionmgr01 sessionmgr02 qns01 qns02 pcrfclient01 pcrfclient02
Do you want to Proceed? (y/n):y
lb01
lb02
sessionmgr01
sessionmgr02
qns01
qns02
pcrfclient01
pcrfclient02
```

Schritt 10. Vergewissern Sie sich, dass mon_db-Skripte auf pcrfclient-VMs angehalten werden.

```
#monsum | grep mon_db_for
```

Wenn sie gestoppt wird, ist dies die Ausgabe:

```
mon_db_for_lb_failover Not monitored Program
mon_db_for_callmodel Not monitored Program
```

Wenn sie nicht angehalten wird, ist dies die Ausgabe:

```
mon_db_for_lb_failover OK Program
mon_db_for_callmodel OK Program
```

> **Hinweis**: Wenn mon_db-Skripte nicht angehalten werden, führen Sie diese Befehle auf den entsprechenden pcffclient-VMs aus, um sie manuell anzuhalten.

```
#monit stop mon_db_for_lb_failover
#monit stop mon_db_for_callmodel
```

Schritt 11. Führen Sie diesen Befehl aus pcrfclient01 aus, um den aktuellen Arbiter aus dem Replikatsatz zu entfernen (set07 ist ein Beispiel in diesem Schritt).

```
#build_set.sh --session --remove-members --setname set07

Please enter the member details which you going to remove from the replica-set
Member:Port --------> arbitervip:27727
arbitervip:27727
Do you really want to remove [yes(y)/no(n)]: y
```

Schritt 12: Führen Sie diesen Befehl im Cluster Manager aus, um zu überprüfen, ob der Arbiter entfernt wurde von **set07**, die Ausgabe von **set07** kann nicht den aktuellen Arbiter darin haben.

```
#diagnostics.sh --get_replica_status

Expected output:
----------|
|-----------------------------------------------------------------------------------------------------
----------------------------------------|
| SESSION:set07 |
| Status via sessionmgr01:27727 sessionmgr02:27727 |
| Member-1 - 27727 : - SECONDARY - sessionmgr01 - ON-LINE - 0 sec -|
| Member-2 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - -------- -|
|-----------------------------------------------------------------------------------------------------
----------------------------------------|
```

Schritt 13: Aktualisieren Sie **mongoConfig.cfg** -Datei, um den richtigen Arbiter im modifizierten Replikatsatz zu haben. Ersetzen Sie den aktuellen Arbiter (ARBITER = Arbiter) durch den neuen Arbiter (ARBITER = new-Arbiter). Führen Sie diesen Befehl im Cluster-Manager aus.

```
#vi /etc/broadhop/mongoConfig.cfg
```

Aktuelle Konfiguration:

```
[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]
```

Erforderliche Konfiguration:

```
[SESSION-SET2]
SETNAME=set07
OPLOG_SIZE=5120
ARBITER=new-arbiter:27727
ARBITER_DATA_PATH=/var/data/sessions.7
MEMBER1=sessionmgr02:27727
MEMBER2=sessionmgr01:27727
DATA_PATH=/var/data/sessions.1/2
[SESSION-SET2-END]
```

Schritt 14: Kopieren Sie die aktualisierte **mongoConfig.cfg** auf alle VMs. Führen Sie diesen Befehl im

Cluster-Manager aus.

```
#copytoall.sh /etc/broadhop/mongoConfig.cfg /etc/broadhop/mongoConfig.cfg
```

Schritt 15: Fügen Sie ein neues Arbitermitglied zu set07 hinzu. Im Cluster Manager ausführen **/var/qps/install/current/scripts/build/build_etc.sh** Befehl, um die **/etc/directory**.

Schritt 16: Überprüfen Sie, ob das neue Arbiterelement dem Replikatsatz hinzugefügt wird, nachdem Sie die **build_etc.sh** Skript, jetzt müssen Sie warten, bis der AIDO-Server die Replikatgruppen mit dem neuen Arbiter erstellt/aktualisiert.

```
#diagnostics.sh --get_replica_status

Expected Output:

| 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 : xx.xx.xx.xx - ARBITER - new-arbiter - ON-LINE - -------- - 0 |
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - -------- - 3 |
|-----------------------------------------|
```

> **Hinweis**: Wenn das neue Arbiter-Mitglied nicht hinzugefügt wird, fahren Sie mit den nächsten Schritten fort. Navigieren Sie zu Schritt 18.

Schritt 17: Führen Sie diesen Befehl im Cluster-Manager aus, um ein neues Arbitermitglied mit Nachdruck hinzuzufügen.

```
#build_set.sh --DB_NAME --add-members --setname Setxxx --force
```

Schritt 18: Wenn der Arbiter-Port noch nicht aktiv ist, führen Sie diesen Befehl vom neuen Arbiter-Knoten aus, um denselben zu starten.

```
Command syntax:
#/etc/init.d/sessionmgr-XXXXX start

Sample command:
#/etc/init.d/sessionmgr-27727 start
```

Schritt 19: Überprüfen Sie, ob der neue Arbiter erfolgreich hinzugefügt wurde.

```
#diagnostics.sh --get_replica_status
```

Schritt 20: Führen Sie diesen Befehl im Cluster Manager aus, um die DB-Priorität entsprechend zu aktualisieren.

```
# cd /var/qps/bin/support/mongo/
# ./set_priority.sh --db session
# ./set_priority.sh --db spr
# ./set_priority.sh --db admin
# ./set_priority.sh --db balance
# ./set_priority.sh --db audit
```

```
# ./set_priority.sh --db report
```

Schritt 21: Führen Sie diesen Befehl im Cluster-Manager aus, um die Änderungen im Replikatsatz zu überprüfen.

```
#diagnostics.sh --get_replica_status

Expected Output:

| 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 : xx.xx.xx.xx - ARBITER - new-arbiter - ON-LINE - -------- - 0 |
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - -------- - 3 |
|----------------------------------------|
```

Schritt 22: Stellen Sie sicher, dass mon_db-Skripts auf pcrfclient-VMs wiederhergestellt werden. Ist dies nicht der Fall, müssen Sie die Programme manuell starten.

```
#monsum | grep mon_db_for
```

Um das mon_db-Skript zu aktivieren, melden Sie sich bei allen pcrfclient-VMs an, und führen Sie die folgenden Befehle aus:

```
# monit start mon_db_for_lb_failover
# monit start mon_db_for_callmodel
```

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.