Behandlung von Abonnentensitzungen bei Festlegung des Sitzungsreplikats in CPS

Inhalt

[Einleitung](#)

[Voraussetzungen](#)

- [Anforderungen](#)
- [Verwendete Komponenten](#)

[Hintergrundinformationen](#)

[Problem](#)

[Lösung](#)

Einleitung

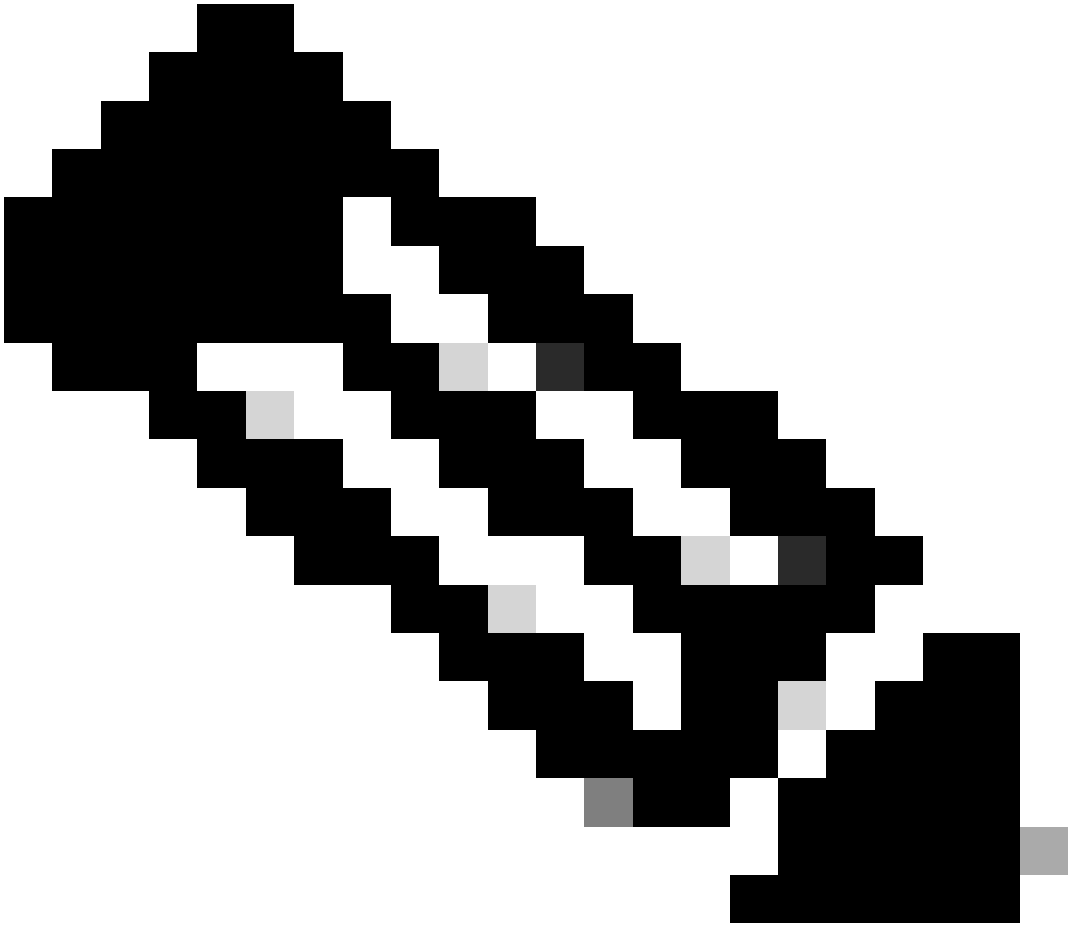
In diesem Dokument wird die Handhabung von Abonnentensitzungen bei Sitzungsreplikaten beschrieben, die in Cisco Policy Suite (CPS) festgelegt wurden.

Voraussetzungen

Anforderungen

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

- Linux
- CPS
- MongoDB



Anmerkung: 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

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 kennen.

Hintergrundinformationen

CPS verwendet MongoDB, wobei mongod-Prozesse auf virtuellen Maschinen (VMs) von sessionmgr ausgeführt werden, um die grundlegende Datenbankstruktur zu bilden.

Die empfohlene Mindestkonfiguration für hohe Verfügbarkeit eines Replikatsatzes besteht aus einem Replikatsatz mit drei Elementen und drei datentragenden Elementen: ein primäres und zwei sekundäre Mitglieder. Unter bestimmten Umständen (z. B. wenn Sie über eine primäre und eine sekundäre Instanz verfügen, Kostenbeschränkungen das Hinzufügen einer weiteren sekundären Instanz jedoch verhindern) können Sie einen Arbiter einbeziehen. Ein Schiedsrichter nimmt an Wahlen teil, verfügt aber nicht über Daten (d. h. er bietet keine Datenredundanz). Im Falle von CPS werden normalerweise Session DB auf diese Weise konfiguriert.

Sie können die Replikatgruppenkonfiguration für Ihr CPS-Setup unter /etc/broadhop/mongoConfig.cfg überprüfen.

```
[SESSION-SET1]
SETNAME=set01a
OPLOG_SIZE=5120
ARBITER1=arbitervip:27717
ARBITER_DATA_PATH=/var/data/sessions.27717
MEMBER1=sessionmgr01:27717
MEMBER2=sessionmgr02:27717
DATA_PATH=/var/data/sessions.1/d
SHARD_COUNT=4
SEEDS=sessionmgr01:sessionmgr02:27717
[SESSION-SET1-END]

[SESSION-SET7]
SETNAME=set01g
OPLOG_SIZE=5120
ARBITER1=arbitervip:37717
ARBITER_DATA_PATH=/var/data/sessions.37717
MEMBER1=sessionmgr02:37717
MEMBER2=sessionmgr01:37717
DATA_PATH=/var/data/sessions.1/g
SHARD_COUNT=2
SEEDS=sessionmgr02:sessionmgr01:37717
[SESSION-SET7-END]
```

MongoDB hat ein anderes Konzept namens Sharding, das Redundanz und Geschwindigkeit für einen Cluster hilft. Durch Shards wird die Datenbank in indizierte Sätze unterteilt, die eine wesentlich höhere Geschwindigkeit für Schreibvorgänge ermöglichen, wodurch die allgemeine Datenbankleistung verbessert wird. Freigegebene Datenbanken werden häufig so eingerichtet, dass jede Freigabe ein Replikatsatz ist.

- Gemeinsame Nutzung der Sitzung: Gemeinsame Sitzungskerne und deren Datenbanken:

```
osgi> listshards
```

```
Shard Id Mongo DB State Backup DB Removed Session Count

1 sessionmgr01:27717/session_cache online false false 109306
2 sessionmgr01:27717/session_cache_2 online false false 109730
3 sessionmgr01:27717/session_cache_3 online false false 109674
4 sessionmgr01:27717/session_cache_4 online false false 108957
```

- Sekundäre Schlüsselfreigabe: Secondary Key teilt Samen und seine Datenbanken:

```
osgi> listskshards

Shard Id Mongo DB State Backup DB Removed Session Count

2 sessionmgr02:37717/sk_cache online false false 150306
3 sessionmgr02:37717/sk_cache_2 online false false 149605
```

Problem

Problem 1: Stetiges Wachstum des Arbeitsspeichers von Datenmitgliedern aufgrund eines Einzelmitgliedsausfalls.

Wenn ein datenführendes Element in 3 Elemente ausfällt (2 datenführende +1 Arbiter), übernimmt das einzige verbleibende datenführende Element die Rolle des primären und Replikatsatzes, funktioniert aber bei hoher Last und Replikatsatz ohne DB-Redundanz. Bei einer PSA-Architektur mit drei Elementen steigt der Cache-Druck, wenn ein datenführender Knoten ausfällt. Dies führt zu einem stetigen Anstieg der Speicherauslastung für den verbleibenden datenführenden Knoten. (Primär), was möglicherweise zum Ausfall des Knotens führt, da der verfügbare Speicher erschöpft wird, wenn dieser unbeaufsichtigt bleibt, und der Replikatsatz letztendlich ausfällt.

```
----------------------------------------------------------------------------------------------
|[SESSION:set01a]|
|[Status via sessionmgr02:27717 ]|
|[Member-1 - 27717 : 192.168.29.100 - ARBITER - arbitervip - ON-LINE
|[Member-2 - 27717 : 192.168.29.35  - UNKNOWN - sessionmgr01 - OFF-LINE 19765 days
|[Member-3 - 27717 : 192.168.29.36  - PRIMARY - sessionmgr02 - ON-LINE
----------------------------------------------------------------------------------------------

----------------------------------------------------------------------------------------------
|[SESSION:set01g]|
|[Status via sessionmgr02:37717 ]|
|[Member-1 - 37717 : 192.168.29.100 - ARBITER - arbitervip - ON-LINE
|[Member-2 - 37717 : 192.168.29.35  - UNKNOWN - sessionmgr01 - OFF-LINE 19765 days
|[Member-3 - 37717 : 192.168.29.36  - PRIMARY - sessionmgr02 - ON-LINE
----------------------------------------------------------------------------------------------
```

Problem 2. Auswirkungen auf die Sitzungsverarbeitung aufgrund eines Doppelmitgliedsfehlers.

Wenn beide Elemente (SessionMgr01 und SessionMgr02) in solchen Replikatgruppen ausfallen (Double-Failure), wird die gesamte Replikatgruppe deaktiviert, und die Basisdatenbankfunktion wird beeinträchtigt.

```
Current setup have problem while connecting to the server on port : 27717

Current setup have problem while connecting to the server on port : 37717
```

Dieser Replikatsatz-Fehler führt im Fall von Sitzungsreplikatsätzen zu einem Anruffehler, da CPS-Prozesse zur Anrufbearbeitung (Quantum Network Suite- (QNS)-Prozesse) nicht auf die Sitzungen zugreifen können, die bereits in diesem fehlerhaften Replikatsatz gespeichert sind.

Lösung

Ansatz 1: Bei Ausfall eines Mitglieds.

Sie müssen in der Lage sein, das fehlerhafte Replikatgruppenmitglied in einer Primary-Secondary-Arbiter (PSA)-Architektur mit kurzer Zeitspanne wiederherzustellen. Falls die Wiederherstellung eines fehlerhaften datentragenden Elements in einer PSA-Architektur Zeit in Anspruch nimmt, müssen Sie das fehlerhafte Element entfernen.

Schritt 1.1: Identifizieren Sie das fehlerhafte datenführende Element in der jeweiligen Replikatgruppe mit der PSA-Architektur mit drei Elementen. Führen Sie diesen Befehl im Cluster-Manager aus.

```
#diagnostics.sh --get_r
```

Schritt 1.2: Entfernen Sie das fehlerhafte Datenelement aus dem Replikatsatz.

```
Syntax:

Usage: build_set.sh <--option1> <--option2> [--setname SETNAME] [--help]

option1: Database name
option2: Build operations (create, add or remove members)

Example:

#build_set.sh --session --remove-failed-members --setname set01a --force
#build_set.sh --session --remove-failed-members --setname set01g --force
```

Schritt 1.3: Überprüfen, ob das fehlerhafte Mitglied aus dem Replikatsatz entfernt wurde

```
#diagnostics.sh --get_r
```

Ansatz 2. Bei Ausfall eines Doppelmitglieds.

Dies ist keine dauerhafte Problemumgehung, wenn beide datentragenden Elemente in einem Replikatsatz mit drei PSA ausfallen. Es handelt sich vielmehr um eine vorübergehende Problemumgehung, um Anruffehler zu vermeiden oder zu reduzieren und eine nahtlose Verarbeitung des Datenverkehrs sicherzustellen, indem fehlgeschlagene Mitglieder aus dem jeweiligen Replikatsatz, der gemeinsamen Nutzung von Sitzungen und der entsprechenden gemeinsamen Nutzung von Aufgaben entfernt werden. Sie müssen an der Wiederherstellung der gescheiterten Mitglieder so schnell wie möglich arbeiten, um weitere unerwünschte Auswirkungen zu vermeiden.

Schritt 2.1. Da die Sitzungsreplikatgruppen deaktiviert sind, während die Sitzungsreplikatgruppen "sessionmgr09" und "sessionmgr10" deaktiviert sind, müssen Sie diese Replikatgruppeneinträge aus der Sitzungsfreigabe und aus der OSGI-Konsole entfernen:

```
#telnet qns0x 9091

osgi> listshards

Shard Id Mongo DB State Backup DB Removed Session Count

1 sessionmgr01:27717/session_cache online false false 109306
2 sessionmgr01:27717/session_cache_2 online false false 109730
3 sessionmgr01:27717/session_cache_3 online false false 109674
4 sessionmgr01:27717/session_cache_4 online false false 108957

osgi> listskshards

Shard Id Mongo DB State Backup DB Removed Session Count
2 sessionmgr02:37717/sk_cache online false false 150306
3 sessionmgr02:37717/sk_cache_2 online false false 149605
```

Schritt 2.2: Entfernen Sie diese Sitzungsfreigaben:

```
osgi> removeshard 1
osgi> removeshard 2
osgi> removeshard 3
osgi> removeshard 4
```

Schritt 2.3: Entfernen Sie diese Scherben:

```
osgi> removeskshard 2
osgi> removeskshard 3
```

Schritt 2.4:Bevor Sie den Rebalancing-Vorgang durchführen, überprüfen Sie die Admin-DB (prüfen Sie, ob die Instanzversion für alle qns VMs übereinstimmt):

<#root>

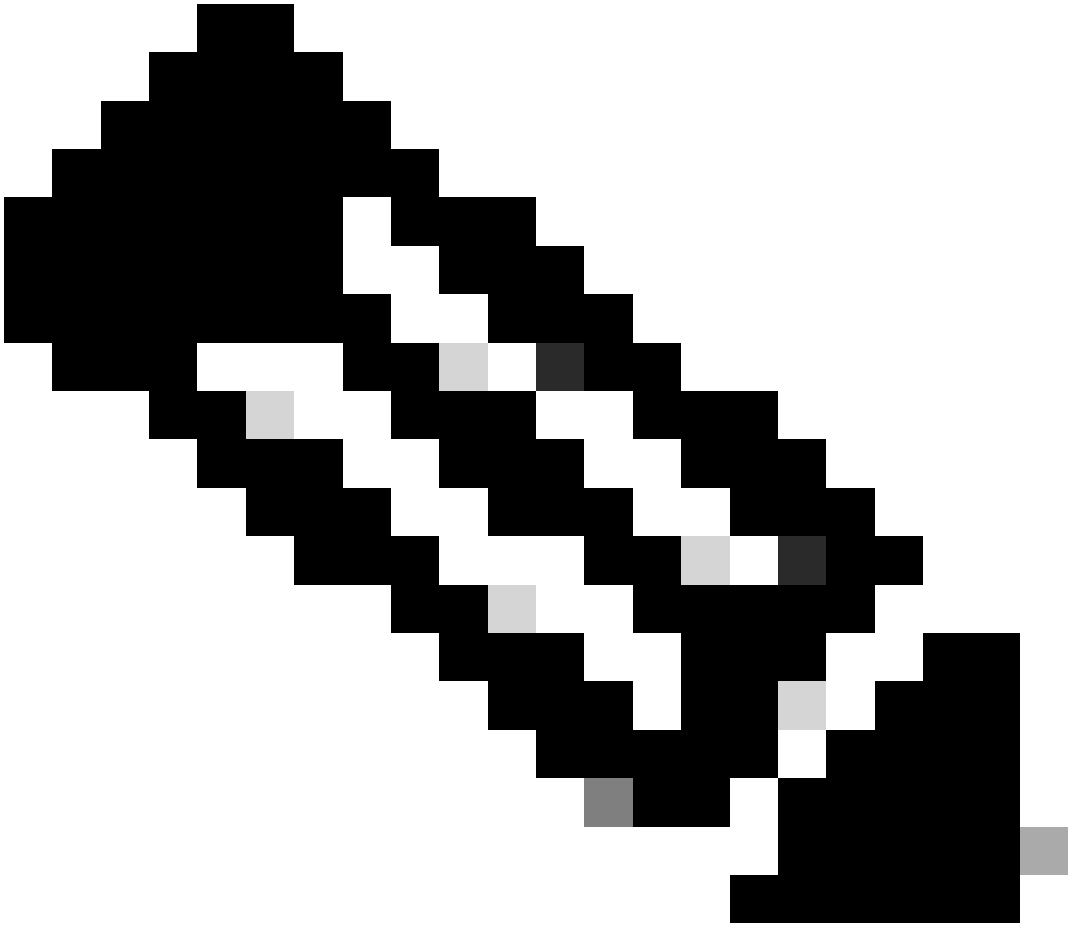
```
 #mongo sessionmgrxx:xxxx/sharding.     [Note: use the primary sessionmgr hostname and respective port

 #set05:PRIMARY>  db.instances.find()
 { "_id" : "qns02-1", "version" : 961 }
 { "_id" : "qns07-1", "version" : 961 }
 { "_id" : "qns08-1", "version" : 961 }
 { "_id" : "qns04-1", "version" : 961 }
 { "_id" : "qns08-1", "version" : 961 }
 { "_id" : "qns05-1", "version" : 961 }
```

Note: if the sharding versions (previous output) are different for some QNS instances. For example, if y

```
{ "_id" : "qns08-1", "version" : 961 }
{ "_id" : "qns04-1", "version" : 962 }
```

Führen Sie diesen Befehl auf der Datenbank für die gemeinsame Nutzung durch den Administrator aus (unter Verwendung des richtigen Hostnamens):



Anmerkung: Wenn Sie ein sekundäres Mitglied sind, verwenden Sie rs.slaveOk(), um Befehle ausführen zu können.

```
[root@casant01-cm csv]#  mongo sessionmgr01:27721/shardin
set05:PRIMARY>
set05:PRIMARY> db.instances.remove({ “_id” : “$QNS_hostname”})


Example:

set05:PRIMARY> db.instances.remove({ “_id” : “qns04-1”})
set05:PRIMARY> exit
```

Schritt 2.5. Führen Sie jetzt eine gemeinsame Sitzungswiederherstellung aus.

```
Login to osgi console.

#telnet qns0x 9091

osgi>listshards

osgi>rebalance

osgi>rebalancestatus

Verify shards:

osgi>listshards
```

Schritt 2.6. Neugewichtung der Freigabe ausführen:

```
Login to osgi console.

#telnet qns0x 9091

osgi>listskshard

osgi>rebalancesk

osgi>rebalanceskstatus

Verify shards:

osgi>listshards
```

Schritt 2.7. Entfernen Sie den Replikatsatz set01a und set01g (auf Cluman ausführen):

```
#build_set.sh --session --remove-replica-set --setname set01a --force
#build_set.sh --session --remove-replica-set --setname set01g --force
```

Schritt 2.8. Starten Sie den qns-Dienst neu (führen Sie ihn auf dem Cluman aus):

```
#restartall.sh
```

Schritt 2.9. Entfernen Sie die Zeilen set01a und set01g aus der Datei mongoConfig.cfg. Führen Sie diesen Befehl auf Cluster Manager aus:

```
#cd /etc/broadhop/
#/bin/cp -p mongoConfig.cfg mongoConfig.cfg_backup_
```

```
#vi mongoConfig.cfg

[SESSION-SET1]
SETNAME=set01a
OPLOG_SIZE=5120
ARBITER1=arbitervip:27717
ARBITER_DATA_PATH=/var/data/sessions.27717
MEMBER1=sessionmgr01:27717
MEMBER2=sessionmgr02:27717
DATA_PATH=/var/data/sessions.1/d
SHARD_COUNT=4
SEEDS=sessionmgr01:sessionmgr02:27717
[SESSION-SET1-END]

[SESSION-SET7]
SETNAME=set01g
OPLOG_SIZE=5120
ARBITER1=arbitervip:37717
ARBITER_DATA_PATH=/var/data/sessions.37717
MEMBER1=sessionmgr02:37717
MEMBER2=sessionmgr01:37717
DATA_PATH=/var/data/sessions.1/g
SHARD_COUNT=2
SEEDS=sessionmgr02:sessionmgr01:37717
[SESSION-SET7-END]
```

Schritt 2.10. Nach dem Entfernen der Zeilen speichern und beenden.

Führen Sie build_etc auf dem Cluster Manager aus.

```
#/var/qps/install/current/scripts/build/build_etc.sh
```

Schritt 2.11: Überprüfen Sie, ob der Replikatsatz "set01d" durch die Diagnose entfernt wurde.

```
#diagnostics.sh --get_r
```

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.