

Dépannage d'une défaillance d'instance mongole en raison d'une utilisation accrue de l'espace DATA_PATH

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Informations générales](#)

[Problème](#)

[Restaurer l'instance mongole dans Sessionmgr](#)

Introduction

Ce document décrit comment dépanner l'échec d'instance mongod dans Cisco Policy Suite (CPS) sessionmgr en raison de l'utilisation accrue de l'espace DATA_PATH.

Conditions préalables

Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Linux
- CPS
- MongoDB

Components Used

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

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

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

Informations générales

CPS utilise MongoDB où les processus mongod s'exécutent sur des machines virtuelles sessionmgr afin de constituer sa structure de base de données.

Plusieurs instances mongod s'exécutent sur un sessionmgr, et chacun d'eux a reçu des numéros de port différents. Ces instances mongoles participent à divers jeux de réplicas.

Problème

Chaque fois qu'une instance mongod particulière s'arrête en raison d'une consommation accrue de l'espace DATA_PATH de son chemin d'accès aux DONNÉES associé, vous remarquerez la même chose dans les diagnostics de cette session. Les connexions au port spécifique ont échoué et la partition /var/data/sessions.X est utilisée à 100 %. Par conséquent, cette instance mongod passe à l'état OFF-LINE dans le jeu de réplicas respectif. Par la suite, son état de participation dans ce jeu de réplicas devient INCONNU.

Un exemple d'erreur dans les diagnostics est fourni. Saisissez le `diagnostics.sh` à partir de ClusterManager ou pcrclient afin de vérifier l'état actuel de mongod et de l'ensemble de réplicas.

```
Could not connect to port 27718 on sessionmgr02 (set02)...[FAIL]
```

```
Disk usage on sessionmgr02...[FAIL]
```

```
Disk usage is above critical threshold (97%) on sessionmgr02.
```

```
Results of: ssh root@sessionmgr02 -x 'df -hP -x iso9660'
```

```
-----  
Filesystem Size Used Avail Use% Mounted on  
/dev/sda2 95G 28G 62G 32% /  
tmpfs 48G 0 48G 0% /dev/shm  
tmpfs 57G 0 57G 0% /var/data/sessions.1  
tmpfs 12G 12G 0 100% /var/data/sessions.2  
-----
```

```
|-----|  
| BALANCE:set02 |  
| Status via arbitervip:27718 sessionmgr01:27718 |  
| Member-1 - 27718 : - UNKNOWN - sessionmgr02 - OFF-LINE - 19003 days - 2 |  
| Member-2 - 27718 : - PRIMARY - sessionmgr01 - ON-LINE - ----- - 3 |  
| Member-3 - 27718 : 192.168.10.146 - ARBITER - arbitervip - ON-LINE - ----- - 0 |  
|-----|
```

Restaurer l'instance mongole dans Sessionmgr

La section détaille la procédure de restauration de l'instance mongod dans sessionmgr si elle est désactivée en raison d'une consommation accrue d'espace DATA_PATH.

Avant de commencer cette procédure, vous devez disposer d'un accès privilégié à :

- Accès racine à l'interface de ligne de commande CPS
- Accès utilisateur « qns-svn » aux interfaces utilisateur CPS - Policy Builder et CPS Central

Vous trouverez ici la procédure pour sessionmg02 et le port 27718, qui fait partie de set02.

1. Connectez-vous au gestionnaire de sessions respectif.
2. Entrez cette commande afin d'identifier la partition où elle a stocké les données pour ce set02 spécifique.

```
[root@dc1-sessionmgr02 ~]# cat /etc/broadhop/mongoConfig.cfg | grep -A6 set02 | grep "DATA_PATH"
ARBITER_DATA_PATH=/var/data/sessions.2
DATA_PATH=/var/data/sessions.2
```

3. Entrez cette commande afin de vérifier si `aido_client` est présent ou non.

```
[root@dc1-sessionmgr02 ~]# monsum
Monit 5.26.0 uptime: 11d 2h 9m
```

```
Service Name  Status  Type
dc1-sessionmgr02  OK    System
whisper  OK    Process
snmpd  OK    Process
memcached  OK    Process
collectd  OK    Process
auditrpmsh.sh  OK    Process
aido_client  OK    Process
primary_db_frag  OK    Program
cpu_load_monitor  OK    Program
cpu_load_trap  OK    Program
gen_low_mem_trap  OK    Program
```

4. Si la `aido_client` est présent, saisissez `monit stop aido_client` afin de l'arrêter.

5. Entrez cette commande afin de vérifier si le processus d'instance de mongod respectif est toujours actif ou non.

```
[root@dc1-sessionmgr02 ~]# ps -ef | grep 27718
root 12292 11114 0 02:05 pts/0 00:00:00 grep --color=auto 27718
root 19620 1 0 2021 ? 01:36:51 /usr/bin/mongod --ipv6 --syncdelay 1 --slowms 500 --storageEngine
mmapv1 --bind_ip_all --port 27718 --dbpath=/var/data/sessions.2 --replSet set02 --fork --pidfilepath
/var/run/sessionmgr-27718.pid --oplogSize 5120 --logpath /var/log/mongodb-27718.log --logappend --quiet
[root@dc1-sessionmgr02 ~]#
```

6. Si l'instance mongod est toujours active, entrez cette commande pour l'arrêter.

```
[root@dc1-sessionmgr02 ~]# /etc/init.d/sessionmgr-27718 stop
Stopping sessionmgr-27718 (via systemctl): [ OK ]
[root@dc1-sessionmgr02 ~]#
```

7. Accédez à `DATA_PATH` reçu à l'étape 1.

```
[root@dc1-sessionmgr02 ~]# cd /var/data/sessions.2
[root@dc1-sessionmgr02 sessions.2]# ls -lrt
total 6616100
-rw----- 1 root root 16777216 Jun 22 2018 admin.ns
-rw----- 1 root root 67108864 Jun 22 2018 admin.0
-rw----- 1 root root 69 Nov 10 07:27 storage.bson
-rw----- 1 root root 16777216 Nov 10 07:27 vouchers.ns
-rw----- 1 root root 67108864 Nov 10 07:27 vouchers.0
-rw----- 1 root root 2146435072 Nov 10 07:27 local.2
drwx----- 2 root root 4096 Nov 10 07:27 local
-rw----- 1 root root 67108864 Nov 10 07:27 local.0
-rw----- 1 root root 16777216 Jan 7 14:38 config.ns
-rw----- 1 root root 67108864 Jan 7 14:38 config.0
-rw----- 1 root root 16777216 Jan 11 02:06 local.ns
-rw----- 1 root root 2146435072 Jan 11 02:06 local.1
drwx----- 2 root root 4096 Jan 11 02:06 diagnostic.data
-rw----- 1 root root 2146435072 Jan 11 02:06 local.3
-rw----- 1 root root 0 Jan 11 02:07 mongod.lock
drwx----- 2 root root 4096 Jan 11 02:08 journal
[root@dc1-sessionmgr02 sessions.2]#
```

8. Saisissez la commande `rm -rf *` afin d'effacer `DATA_PATH`.

9. Entrez cette commande afin de démarrer l'instance mongod. Cette commande prend quelques minutes.

```
[root@dc1-sessionmgr02 ~]# /etc/init.d/sessionmgr-27718 start
Starting sessionmgr-27718 (via systemctl): [ OK ]
[root@dc1-sessionmgr02 ~]#
```

10. Si vous avez arrêté `aido_client` à l'étape 3, saisissez `monit start aido_client` afin de le redémarrer.

11. Saisissez le `diagnostics.sh` à partir de ClusterManager ou `pcrfclient` afin de confirmer que l'instance mongod correspondante est restaurée et est devenue ON-LINE dans le jeu de réplicas.

```
|-----|
----|
| BALANCE:set02 |
| Status via arbitervip:27718 sessionmgr01:27718 sessionmgr02:27718 |
| Member-1 - 27718 : - SECONDARY - sessionmgr02 - ON-LINE - 0 sec - 2 |
| Member-2 - 27718 : - PRIMARY - sessionmgr01 - ON-LINE - ----- - 3 |
| Member-3 - 27718 : XX.XX.XX.XX - ARBITER - arbitervip - ON-LINE - ----- - 0 |
|-----|
----|
```