

Solução de problemas de falha de instância mondeus devido ao aumento da utilização do espaço DATA_PATH

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Problema](#)

[Restaurar a instância mongodo no Sessionmgr](#)

Introduction

Este documento descreve como solucionar problemas de falha de instância mongood no gerente de sessão do Cisco Policy Suite (CPS) devido ao aumento da utilização do espaço DATA_PATH.

Prerequisites

Requirements

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Linux
- CPS
- MongoDB

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

- 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. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

Informações de Apoio

O CPS usa o MongoDB, onde mongood processes é executado em máquinas virtuais (VMs) do

sessionmgr para constituir sua estrutura de banco de dados básica.

Várias instâncias mongod são executadas em um gerente de sessão, e cada uma delas recebeu números de porta diferentes. Essas instâncias mongod participam de vários Conjuntos de Réplicas.

Problema

Sempre que qualquer instância mongod em particular parar devido ao aumento do consumo de espaço de DATA_PATH de seu caminho de dados associado, você observará o mesmo no diagnóstico desse sessionmgr. As conexões com a porta específica falharam e há 100% de utilização da partição /var/data/sessions.X. Assim, essa instância mongod entra no estado OFF-LINE no respectivo conjunto de réplicas. Subsequentemente, seu status de participação nesse Conjunto de Réplicas se torna DESCONHECIDO.

Um exemplo de erro no diagnóstico é fornecido. Digite o `diagnostics.sh` do ClusterManager ou do `prcflclient` para verificar o status atual do conjunto de réplicas e mongods.

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

Restaurar a instância mongod no Sessionmgr

A seção detalha o procedimento para restaurar a instância mongod no sessionmgr se ela estiver inativa devido ao aumento do consumo de espaço de DATA_PATH.

Antes de iniciar este procedimento, você deve ter acesso de privilégio a:

- Acesso raiz à CLI do CPS
- acesso de usuário "qns-svn" às GUIs do CPS - Policy Builder e CPS Central

Aqui está o procedimento para a sessão mg02 e o porto 27718, que faz parte da set02.

1. Faça login no respectivo gerente de sessão.
2. Insira este comando para identificar a partição onde ela armazenou dados para esse conjunto específico02.

```
[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. Insira este comando para verificar se o comando `aido_client` o processo está presente ou não.

```
[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. Se a `aido_client` o processo está presente, insira o comando `monit stop aido_client` para pará-lo.
5. Insira este comando para verificar se o respectivo processo de instância `mondeus` ainda está ativo ou não.

```
[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. Se a instância `mondeus` ainda estiver ativa, digite este comando para pará-la.

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

7. Navegue até `DATA_PATH` recebido na etapa 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. Insira o comando `rm -rf *` para limpar o DATA_PATH.

9. Insira este comando para iniciar a instância monGod. Esse comando leva alguns minutos para ser concluído.

```

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

```

10. Se você parou o `aido_client` na etapa 3, digite o comando `monit start aido_client` para iniciá-lo novamente.

11. Digite o `diagnostics.sh` do ClusterManager ou do `pcrfclient` para confirmar que a respectiva instância monGod foi restaurada e se tornou ON-LINE no Conjunto 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 |
|-----|
----|

```