

DATA_PATH 공간 사용률 증가로 인한 월별 인스턴스 오류 문제 해결

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[문제](#)

[Sessionmgr에서 Monogod 인스턴스 복원](#)

소개

이 문서에서는 DATA_PATH 공간 사용률이 증가하여 Cisco CPS(Policy Suite) sessionmgr에서 monone instance failure를 해결하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- 리눅스
- CPS
- 몽고DB

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

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

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 이해해야 합니다.

배경 정보

CPS는 MongoDB를 사용하여 단일 프로세스가 sessionmgr VM(가상 머신)에서 실행되어 기본 데이터베이스 구조를 구성합니다.

여러 개의 단일 인스턴스가 sessionmgr에서 실행되며 각 인스턴스는 서로 다른 포트 번호가 할당되었습니다. 이러한 월별 인스턴스는 다양한 복제본 세트에 포함됩니다.

문제

연결된 DATA_PATH의 DATA_PATH 공간 소비가 증가하여 특정 단일 인스턴스가 중지될 때마다 해당 세션 관리자에 대한 진단 시 동일한 내용을 확인할 수 있습니다. 특정 포트에 연결하지 못했으며 /var/data/sessions.X 파티션의 사용률이 100%입니다. 따라서 해당 월별 인스턴스는 해당 복제본 세트의 오프라인 상태로 전환됩니다. 그런 다음 해당 복제본 세트에 대한 참가 상태가 UNKNOWN이 됩니다.

진단 유틸리티의 샘플 오류가 제공됩니다. 다음을 입력합니다. diagnostics.sh 명령을 사용하여 monowon 및 복제본 세트의 현재 상태를 확인합니다.

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

Sessionmgr에서 Monogod 인스턴스 복원

이 섹션에서는 DATA_PATH 공간 사용 증가로 인해 중단된 경우 sessionmgr에서 monoth 인스턴스를 복원하는 절차에 대해 자세히 설명합니다.

이 절차를 시작하기 전에 다음에 대한 권한 액세스 권한이 있어야 합니다.

- CPS CLI에 대한 루트 액세스
- CPS GUI에 대한 "qns-svn" 사용자 액세스 - 정책 작성기 및 CPS Central

여기서 set02의 일부인 sessionmg02 및 포트 27718에 대한 절차를 설명합니다.

1. 각 sessionmgr에 로그인합니다.
2. 특정 set02에 대한 데이터를 저장한 파티션을 식별하려면 이 명령을 입력합니다.

```
[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. 이 명령을 입력하여 `aido_client` 프로세스가 존재하거나 없습니다.

```
[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. 이(가) `aido_client` 프로세스가 있으며 `monit stop aido_client` 명령을 사용하여 중지합니다.

5. 해당 월별 인스턴스 프로세스가 계속 활성 상태인지 확인하려면 이 명령을 입력합니다.

```
[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
mmmapv1 --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. 모니터 인스턴스가 여전히 활성 상태이면 이 명령을 입력하여 중지합니다.

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

7. 1단계에서 수신된 `DATA_PATH`로 이동합니다.

```
[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. 명령을 입력합니다. `rm -rf * DATA_PATH`를 지워야 합니다.

9. mongoth 인스턴스를 시작하려면 이 명령을 입력합니다. 이 명령을 완료하는 데 몇 분이 걸립니다.

```

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

```

10. 를 중지한 경우 `aido_client` 3단계에서 프로세스를 입력하고 `monit start adio_client` 명령을 사용하여 다시 시작합니다.

11. 다음을 입력합니다. `diagnostics.sh` 각 monoth 인스턴스가 복구되고 복제본 세트에서 ON-LINE이 되었는지 확인하기 위해 ClusterManager 또는 pcrfclient에서 명령을 실행합니다.

```

|-----|
|----|
| 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 |
|-----|
|----|

```