

# Troubleshoot Mongod Instance Failure Due to Increased DATA\_PATH Space Utilization

## Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[Problem](#)

[Restore the Mongod Instance in Sessionmgr](#)

## Introduction

This document describes how to troubleshoot the mongod instance failure in Cisco Policy Suite (CPS) sessionmgr due to the increased DATA\_PATH space utilization.

## Prerequisites

### Requirements

Cisco recommends that you have knowledge of these topics:

- Linux
- CPS
- MongoDB

### Components Used

The information in this document is based on these software and hardware versions:

- 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. If your network is live, ensure that you understand the potential impact of any command.

## Background Information

CPS uses MongoDB where mongod processes run on sessionmgr virtual machines (VMs) in order to constitute its basic database structure.

Multiple mongod instances run on a sessionmgr, and each of them has been assigned different port numbers. These mongod instances take part in various Replica Sets.

## Problem

Whenever any particular mongod instance stops due to increased DATA\_PATH space consumption of its associated DATA PATH, you will notice the same in diagnostics for that sessionmgr. Connections to the specific port failed and there is 100% utilization of the /var/data/sessions.X partition. Hence, that mongod instance goes into the OFF-LINE state in the respective Replica Set. Subsequently, its participation status in that Replica Set becomes UNKNOWN.

A sample error in the diagnostics is provided. Enter the `diagnostics.sh` command from ClusterManager or pcrfclient in order to check the current status of mongod and Replica Set.

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

## Restore the Mongod Instance in Sessionmgr

The section details the procedure to restore the mongod instance in sessionmgr if it is down due to increased DATA\_PATH space consumption.

Before you begin this procedure, you must have privilege access to:

- Root access to the CPS CLI
- "qns-svn" user access to the CPS GUIs - Policy builder and CPS Central

Provided here is the procedure for sessionmg02 and port 27718, which is part of set02.

1. Log in to the respective sessionmgr.
2. Enter this command in order to identify the partition where it stored data for that specific set02.

```
[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. Enter this command in order to verify if the `aido_client` process is present or not.

```
[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. If the `aido_client` process is present, enter the `monit stop aido_client` command in order to stop it.
5. Enter this command in order to verify if the respective `mongod` instance process is still active or not.

```
[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. If the `mongod` instance is still active, enter this command to stop it.

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

7. Navigate to the `DATA_PATH` received in step 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. Enter the command `rm -rf *` in order to clear the `DATA_PATH`.

9. Enter this command in order to start the mongod instance. This command takes a couple of minutes to complete.

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

10. If you have stopped the `aido_client` process in step 3, enter the `monit start aido_client` command in order to start it again.

11. Enter the `diagnostics.sh` command from `ClusterManager` or `pcrfclient` in order to confirm the respective mongod instance is restored and became ON-LINE in Replica Set.

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