

Procedure to Handle a Corrupted MongoDB in CPS Mongo Replica Set

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[Problem](#)

[Procedure to Repair the Corrupted DB Member](#)

Introduction

This document describes the procedure to handle a corrupted MongoDB in Cisco Policy Suite (CPS) replica sets.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Linux
- CPS
- MongoDB

Note: Cisco recommends that you must have privilege Root access to CPS CLI.

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

MongoDB is a source-available cross-platform document-oriented database (DB) program.

Classified as a NoSQL DB program. MongoDB is extensively used in CPS to manage its different types of DBs viz SESSION, Subscriber Profile Repository (SPR), Balance, and so on.

Problem

MongoDB gets corrupted when you do an improper db defragmentation while `aido_client` is still active inside the `sessionmgr`.

This leads MongoDB to hold data in memory but not be able to write them locally on the db paths.

This can cause loss of data if the primary member (mongo instance) is restarted on the affected replica set or the `sessionmgr` VM restarts.

In order to understand how a DB member looks to be corrupted, you can log in to one of the problematic members and perform provided checks.

Step1. When you run the command `show dbs`, no output of DB list returned. But when you check the count inside the DB you are aware of, it returns the count.

```
[root@lab-1-pcrfclient01 ~]# mongo --host sessionmgr05:27737
MongoDB shell version v3.6.17
connect to: mongodb://sessionmgr05:27737/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("a8f9b0eb-6e78-4bcd-bd63-60a9a9d813d0") }
MongoDB server version: 3.6.17
Server has startup warnings:
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten]
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten] ** WARNING: Access control is not
enabled for the database.
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten] **           Read and write access to
data and configuration is unrestricted.
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten] **
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten]
2022-03-09T00:53:26.949-0300 I REPL [replexec-0]
2022-03-09T00:53:26.949-0300 I REPL [replexec-0] ** WARNING: This replica set uses arbiters,
but readConcern:majority is enabled
2022-03-09T00:53:26.949-0300 I REPL [replexec-0] **           for this node. This is not a
recommended configuration. Please see
2022-03-09T00:53:26.949-0300 I REPL [replexec-0] **
2022-03-09T00:53:26.949-0300 I REPL [replexec-0] **
set01e:PRIMARY>
set01e:PRIMARY> show dbs ## "no dbs reported"
set01e:PRIMARY> use session_cache ## "Switched to a known DB"
switched to db session_cache
set01e:PRIMARY> db.session.count()
223037 ## "DB has the content inside, hence the total record count is shown"
set01e:PRIMARY> use session_cache_2
switched to db session_cache_2
set01e:PRIMARY> db.session.count()
223643
set01e:PRIMARY> use session_cache_3
switched to db session_cache_3
set01e:PRIMARY> db.session.count()
222939
set01e:PRIMARY> use session_cache_4
switched to db session_cache_4
set01e:PRIMARY> db.session.count()
223692
```

```
set01e:PRIMARY>
set01e:PRIMARY> exit
bye
```

Step 2. When you run **diagnostics.sh --get_shard**, application shard shows the data. This is actually stored in in-memory, not in the DBPATH of the Sessionmgr Virtual Machine (VM).

```
[root@lab-1-pcrfclient01 ~]# diagnostics.sh --get_shard
CPS Diagnostics GR Multi-Node Environment
```

```
-----|
-----|
|                                     SHARD STATUS INFORMATION
|      Date : 2022-03-09 11:00:23      |
|-----|
-----|
```

Shard Id Count	Mongo DB	State	Backup DB	Removed	Session
43	sessionmgr01:27717/session_cache	online	false	false	223873
1	sessionmgr01:27717/session_cache_2	online	false	false	222918
2	sessionmgr01:27717/session_cache_3	online	false	false	223720
3	sessionmgr01:27717/session_cache_4	online	false	false	223393
8	sessionmgr05:27737/session_cache	online	false	false	223188
9	sessionmgr05:27737/session_cache_2	online	false	false	223554
10	sessionmgr05:27737/session_cache_3	online	false	false	222920
11	sessionmgr05:27737/session_cache_4	online	false	false	223562
12	sessionmgr07:27747/session_cache	online	false	false	222663
13	sessionmgr07:27747/session_cache_2	online	false	false	222599
14	sessionmgr07:27747/session_cache_3	online	false	false	222475
15	sessionmgr07:27747/session_cache_4	online	false	false	223446
16	sessionmgr09:27757/session_cache	online	false	false	223246
17	sessionmgr09:27757/session_cache_2	online	false	false	223669
18	sessionmgr09:27757/session_cache_3	online	false	false	223711
19	sessionmgr09:27757/session_cache_4	online	false	false	223311
35	sessionmgr13:27717/session_cache	online	true	false	0
36	sessionmgr13:27717/session_cache_2	online	true	false	0
37	sessionmgr13:27717/session_cache_3	online	true	false	0
38	sessionmgr13:27717/session_cache_4	online	true	false	0

Rebalance Status: Rebalanced

Step 3. This output shows there is no content inside the DB PATH where actual data is supposed to be stored.

```
[SESSION-SET3]
SETNAME=set01e
OPLOG_SIZE=5120
ARBITER=lab-1-arb-sessmgr15:27737
ARBITER_DATA_PATH=/var/data/sessions.1/set01e
PRIMARY-MEMBERS
MEMBER1=lab-1-sessionmgr05:27737
MEMBER2=lab-1-sessionmgr06:27737
SECONDARY-MEMBERS
MEMBER3=lab-2-sessionmgr05:27737
MEMBER4=lab-2-sessionmgr06:27737
DATA_PATH=/var/data/sessions.1/set01e ## "DB DATA Path of set01e replicaset"
[SESSION-SET3-END]
```

Secure Shell (SSH) to the associated sessionmgr and navigate to the DB_PATH mentioned in the mongo configuration. You can see the content inside the DB_PATH is empty.

```
[root@lab-1-sessionmgr05 ~]# cd /var/data/sessions.1/set01e
[root@lab-1-sessionmgr05 ~]# ls -lrt
total 0
[root@lab-1-sessionmgr05 ~]#
```

With these checks, you can come to the conclusion that MongoDB is corrupted.

Procedure to Repair the Corrupted DB Member

Step 1. SSH to the Primary members of the problematic replica set.

Step 2. Stop the `aido_client` (ensure to stop the `aido` client on all the members of the replica set which belongs to `set01e`).

Step 3. Connect to the mongo shell of `set01e` and run these steps.

```
# mongo --port 27737
# show dbs                                # Ensure this returns empty output.
# use admin
# db.repairDatabase()
# use config
# db.repairDatabase()
# exit
```

```
[root@lab-1-sessionmgr05 set01e]# mongo --port 27737
MongoDB shell version v3.6.17
connect to: mongoddb://127.0.0.1:27737/?gssapiServiceName=mongoddb
Implicit session: session { "id" : UUID("ff9df861-0b42-4e8a-99c1-3583670e1926") }
MongoDB server version: 3.6.17
Server has startup warnings:
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten]
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten] ** WARNING: Access control is not
enabled for the database.
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten] **           Read and write access to
data and configuration is unrestricted.
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten] **
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten]
2022-03-09T00:53:26.949-0300 I REPL [replexec-0]
2022-03-09T00:53:26.949-0300 I REPL [replexec-0] ** WARNING: This replica set uses arbiters,
but readConcern:majority is enabled
2022-03-09T00:53:26.949-0300 I REPL [replexec-0] **           for this node. This is not a
recommended configuration. Please see
2022-03-09T00:53:26.949-0300 I REPL [replexec-0] **
2022-03-09T00:53:26.949-0300 I REPL [replexec-0]
set01e:PRIMARY> use admin
switched to db admin
set01e:PRIMARY> db.repairDatabase()
{
  "ok" : 1,
  "operationTime" : Timestamp(1647319246, 352),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1647319246, 352),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}
set01e:PRIMARY>
set01e:PRIMARY> use config
switched to db config
```

```

set01e:PRIMARY> db.repairDatabase()
{
  "ok" : 1,
  "operationTime" : Timestamp(1647319301, 218),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1647319301, 218),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}
set01e:PRIMARY> show dbs
admin    0.031GB
config  0.031GB
set01e:PRIMARY> exit

```

Step 4. Connect back on the same replica instant and perform these commands on all session_cache_dbs. A sample of session_cache DB is briefed here.

```

# mongo --port 27737
# use session_cache
# db.session.count()           # Use this to check that session counts are still intact
# db.stats(1024*1024*1024)    # Use this to verify that the storage size is proper
# db.repairDatabase()
# exit

```

```

[root@lab-1-sessionmgr05 set01e]# mongo --port 27737
MongoDB shell version v3.6.17
connect to: mongod://127.0.0.1:27737/?gssapiServiceName=mongod
Implicit session: session { "id" : UUID("73794d11-0785-4520-ba82-19f0d2bba338") }
MongoDB server version: 3.6.17
Server has startup warnings:
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten]
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten] ** WARNING: Access control is not
enabled for the database.
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten] **           Read and write access to
data and configuration is unrestricted.
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten] **
2022-03-09T00:53:26.910-0300 I CONTROL [initandlisten]
2022-03-09T00:53:26.949-0300 I REPL   [replexec-0]
2022-03-09T00:53:26.949-0300 I REPL   [replexec-0] ** WARNING: This replica set uses arbiters,
but readConcern:majority is enabled
2022-03-09T00:53:26.949-0300 I REPL   [replexec-0] **           for this node. This is not a
recommended configuration. Please see
2022-03-09T00:53:26.949-0300 I REPL   [replexec-0] **
2022-03-09T00:53:26.949-0300 I REPL   [replexec-0]
set01e:PRIMARY>
set01e:PRIMARY>
set01e:PRIMARY>
set01e:PRIMARY> show dbs
admin    0.031GB
config  0.031GB
set01e:PRIMARY> use session_cache
switched to db session_cache
set01e:PRIMARY>
set01e:PRIMARY> db.stats(1024*1024*1024)
{
  "db" : "session_cache",
  "collections" : 3,
  "views" : 0,
  "objects" : 212467,
  "avgObjSize" : 8175.252062673262,

```

```

"dataSize" : 1.6176805645227432,
"storageSize" : 2.471107453107834,
"numExtents" : 22,
"indexes" : 3,
"indexSize" : 0.30870679020881653,
"fileSize" : 0,
"nsSizeMB" : 16,
"extentFreeList" : {
"num" : 0,
"totalSize" : 0
},
"dataFileVersion" : {
"major" : 4,
"minor" : 22
},
"fsUsedSize" : 38.36811065673828,
"fsTotalSize" : 47.044921875,
"ok" : 1,
"operationTime" : Timestamp(1647321405, 102),
"$clusterTime" : {
"clusterTime" : Timestamp(1647321405, 103),
"signature" : {
"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"keyId" : NumberLong(0)
}
}
}
}
set01e:PRIMARY> db.repairDatabase()
{
"ok" : 1,
"operationTime" : Timestamp(1647321444, 84),
"$clusterTime" : {
"clusterTime" : Timestamp(1647321444, 84),
"signature" : {
"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"keyId" : NumberLong(0)
}
}
}
}
set01e:PRIMARY> show dbs
admin          0.031GB
config         0.031GB
session_cache  2.499GB

```

Note: Repeat Step 4. for the rest of the session_cache DBs.

Step 5. Ensure that **show dbs** now lists all the DBs when you connect the same mongo instance back.

```

mongo --port 27737
set01e:PRIMARY> show dbs
admin 0.031GB
config 0.031GB
session_cache 2.499GB
session_cache_2 2.499GB
session_cache_3 2.499GB
session_cache_4 2.499GB

```

Step 6. Ensure that the db path now contains all the data locally inside the sessionmgr. You can

check the respective data path of the replica set. In this case it is **/var/data/sessions.1/set01e**.

```
[root@lab-1-sessionmgr05 set01~]# cd /var/data/sessions.1/set01e
[root@lab-1-sessionmgr05 set01e]# ls
admin session_cache session_cache_2.1 session_cache_2.7 session_cache_3.1 session_cache_3.7
session_cache_4.1 session_cache_4.7 session_cache.8
admin.0 session_cache.0 session_cache_2.2 session_cache_2.8 session_cache_3.2 session_cache_3.8
session_cache_4.2 session_cache_4.8 session_cache.ns
admin.ns session_cache.1 session_cache_2.3 session_cache_2.ns session_cache_3.3
session_cache_3.ns session_cache_4.3 session_cache_4.ns _tmp
config session_cache.2 session_cache_2.4 session_cache.3 session_cache_3.4 session_cache.4
session_cache_4.4 session_cache.5
config.0 session_cache_2 session_cache_2.5 session_cache_3 session_cache_3.5 session_cache_4
session_cache_4.5 session_cache.6
config.ns session_cache_2.0 session_cache_2.6 session_cache_3.0 session_cache_3.6
session_cache_4.0 session_cache_4.6 session_cache.7
```

Step 7. SSH to same site secondary member and perform local sync of data path with the primary member.

```
ssh to lab-1-sessionmgr06 (Secondary member)
Ensure to stop aido_client
```

```
# monit stop aido_client
```

```
Ensure to stop mongo processes
```

```
# /etc/init.d/sessionmgr-27737 stop # Wait for 10 seconds and start the service
back on
```

Ensure that the data path **/var/data/sessions.1/set01e** is empty and if it isn't, then remove with the use of **rm -rf /var/data/sessions.1/set01e/***, then start the mongo process.

```
# /etc/init.d/sessionmgr-27737 start
```

```
[root@lab-1-sessionmgr06 ~]# monit stop aido_client
[root@lab-1-sessionmgr06 ~]# monit status aido_client
Monit 5.26.0 uptime: 52d 20h 59m
```

```
Process 'aido_client'
status Not monitored
monitoring status Not monitored
monitoring mode active
on reboot start
data collected Wed, 23 Mar 2022 08:08:46
```

```
[root@lab-1-sessionmgr06 ~]#
[root@lab-1-sessionmgr06 ~]# /etc/init.d/sessionmgr-27737 stop
stop sessionmgr-27737 (via systemctl): [ OK ]
[root@lab-1-sessionmgr06 ~]# rm -rf /var/data/sessions.1/set01e/*
[root@lab-1-sessionmgr06 ~]# cd /var/data/sessions.1/set01e/
[root@lab-1-sessionmgr06 set01e]# ls
[root@lab-1-sessionmgr06 set01e]#
[root@lab-1-sessionmgr06 set01e]# /etc/init.d/sessionmgr-27737 start
Starting sessionmgr-27737 (via systemctl): [ OK ]
```

Step 8. Verify that the data is now locally copied to /var/data/sessions.1/set01e.

```
[root@lab-1-sessionmgr06 ~]# cd /var/data/sessions.1/set01e/
[root@lab-1-sessionmgr06 set01e]# ls
admin.0 local.1 local.3 local.7 mongod.lock session_cache_2.3 session_cache_2.7
session_cache_3.1 session_cache_3.5 session_cache_3.ns
admin.ns local.10 local.4 local.8 session_cache_2.0 session_cache_2.4 session_cache_2.8
session_cache_3.2 session_cache_3.6 storage.bson
diagnostic.data local.11 local.5 local.9 session_cache_2.1 session_cache_2.5 session_cache_2.ns
session_cache_3.3 session_cache_3.7 _tmp
local.0 local.2 local.6 local.ns session_cache_2.2 session_cache_2.6 session_cache_3.0
session_cache_3.4 session_cache_3.8
[root@lab-1-sessionmgr06 set01e]#
```

Note: Repeat Step 7. and Step 8. for geo site secondary members. Here in lab, the members are lab-2-sessionmgr05 and lab-2-sessionmgr06.

Step 9. Once all the secondary DBs are recovered (local and geo site), then restart the mongo service on the primary member.

```
[root@lab-1-sessionmgr05 ~]# /etc/init.d/sessionmgr-27737 stop
stop sessionmgr-27737 (via systemctl): [ OK ]
```

Wait for 10 seconds and confirm that the primary switch over is successful.

```
[root@lab-1-sessionmgr06 ~]# mongo --port 27737
MongoDB shell version v3.6.17
connect to: mongod://127.0.0.1:27737/?gssapiServiceName=mongod
Implicit session: session { "id" : UUID("ba8e49fa-ad0f-4ac6-8ef8-b4da0a88fe33") }
MongoDB server version: 3.6.17
Server has startup warnings:
2022-03-15T02:54:29.546-0300 I CONTROL [initandlisten]
2022-03-15T02:54:29.546-0300 I CONTROL [initandlisten] ** WARNING: Access control is not enabled
for the database.
2022-03-15T02:54:29.546-0300 I CONTROL [initandlisten] ** Read and write access to data and
configuration is unrestricted.
2022-03-15T02:54:29.546-0300 I CONTROL [initandlisten] **
2022-03-15T02:54:29.546-0300 I CONTROL [initandlisten]
set01e:PRIMARY>
set01e:PRIMARY>
set01e:PRIMARY> show dbs
admin 0.031GB
config 0.031GB
local 5.029GB
session_cache 2.499GB
session_cache_2 2.499GB
session_cache_3 2.499GB
session_cache_4 2.499GB
set01e:PRIMARY> show dbs
admin 0.031GB
config 0.031GB
local 5.029GB
session_cache 2.499GB
session_cache_2 2.499GB
session_cache_3 2.499GB
session_cache_4 2.499GB
set01e:PRIMARY> rs.status()
{
  "set" : "set01e",
  "date" : ISODate("2022-03-15T06:13:19.991Z"),
  "myState" : 1,
```



```
"term" : NumberLong(36),
"syncingTo" : "",
"syncSourceHost" : "",
"syncSourceId" : -1,
"heartbeatIntervalMillis" : NumberLong(2000),
"optimes" : {
  "lastCommittedOpTime" : {
    "ts" : Timestamp(1647324799, 335),
    "t" : NumberLong(36)
  },
  "readConcernMajorityOpTime" : {
    "ts" : Timestamp(1647324799, 335),
    "t" : NumberLong(36)
  },
  "appliedOpTime" : {
    "ts" : Timestamp(1647324799, 338),
    "t" : NumberLong(36)
  },
  "durableOpTime" : {
    "ts" : Timestamp(0, 0),
    "t" : NumberLong(-1)
  }
},
"members" : [
  {
    "_id" : 0,
    "name" : "lab-2-sessionmgr06:27737",
    "health" : 1,
    "state" : 2,
    "stateStr" : "SECONDARY",
    "uptime" : 486,
    "optime" : {
      "ts" : Timestamp(1647324799, 94),
      "t" : NumberLong(36)
    },
    "optimeDurable" : {
      "ts" : Timestamp(0, 0),
      "t" : NumberLong(-1)
    },
    "optimeDate" : ISODate("2022-03-15T06:13:19Z"),
    "optimeDurableDate" : ISODate("1970-01-01T00:00:00Z"),
    "lastHeartbeat" : ISODate("2022-03-15T06:13:19.267Z"),
    "lastHeartbeatRecv" : ISODate("2022-03-15T06:13:18.270Z"),
    "pingMs" : NumberLong(0),
    "lastHeartbeatMessage" : "",
    "syncingTo" : "lab-1-sessionmgr06:27737",
    "syncSourceHost" : "lab-1-sessionmgr06:27737",
    "syncSourceId" : 4,
    "infoMessage" : "",
    "configVersion" : 8
  },
  {
    "_id" : 1,
    "name" : "lab-1-sessionmgr05:27737",
    "health" : 1,
    "state" : 2,
    "stateStr" : "SECONDARY",
    "uptime" : 885,
    "optime" : {
      "ts" : Timestamp(1647324799, 96),
      "t" : NumberLong(36)
    },
    "optimeDurable" : {
      "ts" : Timestamp(0, 0),
```

```
"t" : NumberLong(-1)
},
"optimeDate" : ISODate("2022-03-15T06:13:19Z"),
"optimeDurableDate" : ISODate("1970-01-01T00:00:00Z"),
"lastHeartbeat" : ISODate("2022-03-15T06:13:19.270Z"),
"lastHeartbeatRecv" : ISODate("2022-03-15T06:13:18.270Z"),
"pingMs" : NumberLong(0),
"lastHeartbeatMessage" : "",
"syncingTo" : "lab-1-sessionmgr06:27737",
"syncSourceHost" : "lab-1-sessionmgr06:27737",
"syncSourceId" : 4,
"infoMessage" : "",
"configVersion" : 8
},
{
  "_id" : 2,
"name" : "lab-1-arb-sessmgr15:27737",
"health" : 1,
"state" : 7,
"stateStr" : "ARBITER",
"uptime" : 1130,
"lastHeartbeat" : ISODate("2022-03-15T06:13:19.240Z"),
"lastHeartbeatRecv" : ISODate("2022-03-15T06:13:18.856Z"),
"pingMs" : NumberLong(0),
"lastHeartbeatMessage" : "",
"syncingTo" : "",
"syncSourceHost" : "",
"syncSourceId" : -1,
"infoMessage" : "",
"configVersion" : 8
},
{
  "_id" : 3,
"name" : "lab-1-sessionmgr05:27737",
"health" : 0,
"state" : 8,
"stateStr" : "(not reachable/healthy)",
"uptime" : 0,
"optime" : {
  "ts" : Timestamp(0, 0),
  "t" : NumberLong(-1)
},
"optimeDurable" : {
  "ts" : Timestamp(0, 0),
  "t" : NumberLong(-1)
},
"optimeDate" : ISODate("1970-01-01T00:00:00Z"),
"optimeDurableDate" : ISODate("1970-01-01T00:00:00Z"),
"lastHeartbeat" : ISODate("2022-03-15T06:13:19.299Z"),
"lastHeartbeatRecv" : ISODate("2022-03-15T06:11:58.086Z"),
"pingMs" : NumberLong(0),
"lastHeartbeatMessage" : "Connection refused",
"syncingTo" : "",
"syncSourceHost" : "",
"syncSourceId" : -1,
"infoMessage" : "",
"configVersion" : -1
},
{
  "_id" : 4,
"name" : "lab-1-sessionmgr06:27737",
"health" : 1,
"state" : 1,
"stateStr" : "PRIMARY",
```

```

"uptime" : 1130,
"optime" : {
"ts" : Timestamp(1647324799, 338),
"t" : NumberLong(36)
},
"optimeDate" : ISODate("2022-03-15T06:13:19Z"),
"syncingTo" : "",
"syncSourceHost" : "",
"syncSourceId" : -1,
"infoMessage" : "",
"electionTime" : Timestamp(1647324719, 72),
"electionDate" : ISODate("2022-03-15T06:11:59Z"),
"configVersion" : 8,
"self" : true,
"lastHeartbeatMessage" : ""
}
],
"ok" : 1,
"operationTime" : Timestamp(1647324799, 338),
"$clusterTime" : {
"clusterTime" : Timestamp(1647324799, 338),
"signature" : {
"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"keyId" : NumberLong(0)
}
}
}
}

```

Step 10. Restart the mongo service on lab-1-sessionmgr05, which was the primary member earlier.

```

[root@lab-1-sessionmgr05 ~]# /etc/init.d/sessionmgr-27737 start
Starting sessionmgr-27737 (via systemctl): [ OK ]

```

Step 11. Start the aido_client on all the replica members of the set01e replica set which was stopped in Step 2.