

Procédure de traitement d'une MongoDB endommagée dans le jeu de réplicas Mongo CPS

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Informations générales](#)

[Problème](#)

[Procédure de réparation du membre de base de données endommagé](#)

Introduction

Ce document décrit la procédure à suivre pour gérer une base de données MongoDB (DB) endommagée dans les jeux de réplicas de Cisco Policy Suite (CPS).

Conditions préalables

Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Linux
- CPS
- MongoDB

Note: Cisco vous recommande de disposer d'un privilège d'accès racine à l'interface de ligne de commande CPS.

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

MongoDB est un programme de base de données (DB) multi-plate-forme disponible à la source. Classé comme programme de base de données NoSQL. La base de données MongoDB est largement utilisée dans CPS pour gérer ses différents types de base de données (SESSION, Subscriber Profile Repository (SPR), Balance, etc.).

Problème

MongoDB est corrompu lorsque vous faites une défragmentation de base de données incorrecte alors qu'aido_client est toujours actif dans sessionmgr.

Cela conduit MongoDB à stocker des données en mémoire mais ne peut pas les écrire localement sur les chemins de la base de données.

Cela peut entraîner la perte de données si le membre principal (instance mongo) est redémarré sur le jeu de réplicas affecté ou si la machine virtuelle sessionmgr redémarre.

Afin de comprendre comment un membre DB semble être corrompu, vous pouvez vous connecter à l'un des membres problématiques et effectuer les vérifications fournies.

Étape 1. Lorsque vous exécutez la commande **show dbs**, aucune sortie de la liste DB n'a retourné. Mais lorsque vous vérifiez le nombre dans la base de données que vous connaissez, il retourne le nombre.

```
[root@lab-1-pcrfclient01 ~]# mongo --host sessionmgr05:27737
MongoDB shell version v3.6.17
connect to: mongod://sessionmgr05:27737/?gssapiServiceName=mongod
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

```

Étape 2. Lorsque vous exécutez **diagnostics.sh --get_shard**, le partage d'applications affiche les données. Ceci est en fait stocké en mémoire, pas dans le DBPATH de la machine virtuelle Sessionmgr (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	Mongo DB	State	Backup DB	Removed	Session Count
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

Étape 3. Cette sortie montre qu'il n'y a aucun contenu dans le chemin d'accès de la base de données où les données réelles sont supposées être stockées.

```

[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) vers le sessionmgr associé et accédez au DB_PATH mentionné dans la configuration du mongo. Vous pouvez voir que le contenu dans DB_PATH est vide.

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

Avec ces contrôles, vous pouvez en arriver à la conclusion que MongoDB est corrompu.

Procédure de réparation du membre de base de données endommagé

Étape 1. SSH aux membres principaux du jeu de réplicas problématique.

Étape 2. Arrêtez le client aido (assurez-vous d'arrêter le client aido sur tous les membres du jeu de réplicas qui appartient à set01e).

Étape 3. Connectez-vous au shell mongo de set01e et exécutez ces étapes.

```
# 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: mongod://127.0.0.1:27737/?gssapiServiceName=mongod
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

```

Étape 4. Reconnectez-vous sur le même réplica instantané et exécutez ces commandes sur tous les `db_session_cache`. Un exemple de la base de données `session_cache` est présenté ici.

```

# 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: Répétez l'étape 4. pour le reste des bases de données session_cache.

Étape 5. Assurez-vous que **show dbs** répertorie maintenant toutes les DB lorsque vous reconnectez la même instance de mongo.

```
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
```

Étape 6. Assurez-vous que le chemin d'accès de la base de données contient désormais toutes les données localement à l'intérieur de sessionmgr. Vous pouvez vérifier le chemin d'accès des données respectives du jeu de réplicas. Dans ce cas, il est `/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
```

Étape 7. SSH au même membre secondaire du site et synchroniser localement le chemin de données avec le membre principal.

```
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
```

Assurez-vous que le chemin de données `/var/data/sessions.1/set01e` est vide et, si ce n'est pas le cas, supprimez-le à l'aide de `rm -rf /var/data/sessions.1/set01e/*`, puis lancez le processus mongo.

```
# /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 ]
```

Étape 8. Vérifiez que les données sont maintenant copiées localement dans `/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: Répétez l'étape 7. et Étape 8. pour les membres secondaires du site géo. Dans les travaux pratiques, les membres sont lab-2-session mgr05 et lab-2-session mgr06.

Étape 9. Une fois que toutes les bases de données secondaires sont récupérées (site local et site géo), redémarrez le service mongo sur le membre principal.

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

Patientez 10 secondes et vérifiez que le commutateur principal a réussi.

```
[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,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
    "keyId" : NumberLong(0)
  }
}
}
}

```

Étape 10. Redémarrez le service mongo sur lab-1-session mgr05, qui était auparavant le membre principal.

```

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

```

Étape 11. Démarrez aido_client sur tous les membres de réplica du jeu de réplicas set01e qui a été arrêté à l'étape 2.