

CPS 복제본 집합의 중재자 노드 관리 절차

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[문제](#)

[복제 데이터베이스 집합의 중재자 관리 절차](#)

소개

이 문서에서는 CPS(Cisco Policy Suite) 복제본 세트의 중재자 노드를 관리하는 절차에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- Linux
- CPS
- 몽고DB

참고: CPS CLI에 대한 루트 액세스 권한이 있어야 합니다.

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- CPS 20.2
- UCS(Unified Computing System)-B
- MongoDB v3.6.17 및 v3.4.16

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

배경 정보

CPS는 MongoDB를 사용하여 기본 데이터베이스(DB) 구조를 구성합니다. ADMIN, SPR(Subscriber Profile Repository), BALANCE, SESSION, REPORTING, AUDIT 등 다양한 용도로 여러 복제 세트를 보유합니다.

MongoDB의 복제 데이터베이스 집합은 동일한 데이터 집합을 유지 관리하는 복제 프로세스 그룹입니다. 복제 세트는 이중화 및 고가용성(HA)을 제공합니다. 서로 다른 DB 서버에 여러 개의 데이터 복사본을 사용하여 로드 공유 읽기 작업을 수행할 수 있습니다.

일부 경우(예: 기본 및 보조 인스턴스가 있지만 비용 제약으로 인해 다른 보조 인스턴스를 추가할 수 없는 경우), 선택 시 투표할 중재자로 복제 세트에 단일 인스턴스를 추가할 수 있습니다. 중재인은 정확히 한 번의 선거 투표를 갖는다. 기본적으로 중재자는 우선 순위가 0입니다.

중재자는 복제 세트의 일부이지만 데이터를 보유하지 않는(데이터 이중화를 제공하지 않는다는 의미) 유일한 인스턴스입니다. 그러나 그들은 선거에 참여할 수 있다. 중재자는 기본(primary)에 대한 선거에 참여하지만, 중재자는 데이터 세트의 복사본을 가지고 있지 않으며 기본(primary)이 될 수 없습니다.

중재자는 최소한의 리소스 요구 사항을 가지며 전용 하드웨어가 필요하지 않습니다. 애플리케이션 서버 또는 네트워크를 모니터링하는 호스트에 중재자를 구축할 수 있습니다.

중재자는 데이터를 저장하지 않지만, 중재자 화폐 프로세스가 복제 세트에 추가될 때까지 중재자는 다른 화폐 프로세스와 마찬가지로 데이터 파일 세트 및 전체 크기의 저널로 시작합니다.

다음은 샘플 복제본 세트입니다. **set07**.

```
| SET NAME - PORT : IP ADDRESS - REPLICHA STATE - HOST NAME - HEALTH - LAST SYNC -PRIORITY
|-----|
| SESSION:set07 |
| Status via arbitervip:27727 sessionmgr01:27727 sessionmgr02:27727 |
| Member-1 - 27727 : - SECONDARY - sessionmgr01 - ON-LINE - 0 sec - 2 |
| Member-2 - 27727 : 192.168.10.146 - ARBITER - arbitervip - ON-LINE - ----- - 0 |
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - ----- - 3 |
|-----|
```

문제

복제본 세트의 중재자 또는 중재자를 변경하기 위한 요건에 문제가 있다고 가정하고, 현재 중재자를 제거하고 복제본 세트에 새 중재자를 추가해야 합니다.

복제 데이터베이스 집합의 중재자 관리 절차

1단계. CPS 및 새 조정기에서 mongo 셸 버전을 확인합니다. 복제 데이터베이스 집합 및 새 중재자 노드의 주 sessionmgr에서 이 명령을 실행합니다.

sessionmgr의 샘플 출력:

```
[root@sessionmgr02 ~]# mongo --version
MongoDB shell version v3.6.17
```

mongo 셸 버전이 기본 sessionmgr 및 새 중재자 모두에서 동일하거나 새 중재자 mongo 셸 버전이 더 높은 경우 6단계로 이동합니다.

그렇지 않으면 새 아비터 몽고 셸 버전이 더 낮으면 다음을 설정해야 합니다
featureCompatibilityVersion 복제 데이터베이스 집합의 관리 데이터베이스에서 더 낮은 값으로 다음 단계를 수행합니다.

새 증재자 mongo 셸 버전이 CPS sessionmgr 버전보다 낮은 예제 사례:

```
[root@pcrfclient02 ~]# mongo --version
MongoDB shell version v3.4.16
```

2단계. 복제본 세트의 기본 mongo 인스턴스에 로그인합니다.

```
Command template:
#mongo --host <sessionmgrXX> --port <Replica Set port>
```

```
Sample command:
#mongo --host sessionmgr02 --port 27727
```

3단계. 이 명령을 실행하여 현재 **featureCompatibilityVersion** 복제본 집합의 관리 데이터베이스에 있습니다.

```
set07:PRIMARY> db.adminCommand( { getParameter: 1, featureCompatibilityVersion: 1 } )
{
  "featureCompatibilityVersion" : {
    "version" : "3.6"
  },
  "ok" : 1,
  "operationTime" : Timestamp(1663914140, 1),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1663914140, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}
```

4단계. 이 명령을 실행하여 **setfeatureCompatibilityVersion** 복제본 집합의 관리 데이터베이스에 있는 3.4입니다.

```
set07:PRIMARY> db.adminCommand( { setFeatureCompatibilityVersion: "3.4" } )
{ "ok" : 1 }
```

5단계. 이 명령을 실행하여 **featureCompatibilityVersion** 복제본 세트의 관리 데이터베이스에서 3.4로 변경되었습니다.

```
set07:PRIMARY> db.adminCommand( { getParameter: 1, featureCompatibilityVersion: 1 } )
{ "featureCompatibilityVersion" : { "version" : "3.4" }, "ok" : 1 }
```

6단계. Cluster Manager에 로그인하고 **/var/qps/config/deploy/csv/AdditionalHosts.csv** 새 증재자 세부 정보로 파일을 만듭니다.

```
#vi /var/qps/config/deploy/csv/AdditionalHosts.csv
```

Provide new arbiter details in this format:

```
Host Alias IP Address
new-arbiter new-arbiter xx.xx.xx.xx
```

7단계. CSV 컨피그레이션을 가져옵니다.

```
#!/var/qps/install/current/scripts/import/import_deploy.sh
```

8단계. 다음을 확인합니다. `/etc/hosts` 새 중재자 정보로 업데이트했습니다.

```
#cat /etc/hosts | grep arbiter
```

9단계. 이 명령을 실행하여 동기화 `/etc/hosts`.

```
#!/var/qps/bin/update/synchosts.sh
```

```
Syncing to following QNS Servers:
```

```
lb01 lb02 sessionmgr01 sessionmgr02 qns01 qns02 pcrfclient01 pcrfclient02
```

```
Do you want to Proceed? (y/n):y
```

```
lb01
```

```
lb02
```

```
sessionmgr01
```

```
sessionmgr02
```

```
qns01
```

```
qns02
```

```
pcrfclient01
```

```
pcrfclient02
```

10단계. `mon_db` 스크립트가 `pcrfclient` VM에서 중지되었는지 확인합니다.

```
#monsum | grep mon_db_for
```

중지된 경우 다음과 같은 출력이 표시됩니다.

```
mon_db_for_lb_failover Not monitored Program
```

```
mon_db_for_callmodel Not monitored Program
```

중지되지 않은 경우 다음 출력이 표시됩니다.

```
mon_db_for_lb_failover OK Program
```

```
mon_db_for_callmodel OK Program
```

참고: `mon_db` 스크립트가 중지되지 않은 경우 각 `pcrfclient` VM에서 이러한 명령을 실행하여 수동으로 중지합니다.

```
#monit stop mon_db_for_lb_failover
```

```
#monit stop mon_db_for_callmodel
```

11단계. `pcrfclient01`에서 이 명령을 실행하여 복제본 세트에서 현재 중재자를 제거합니다(이 단계의 예는 `set07`).

```
#build_set.sh --session --remove-members --setname set07
```

```
Please enter the member details which you going to remove from the replica-set
```

```
Member:Port -----> arbitervip:27727
```

```
arbitervip:27727
```

```
Do you really want to remove [yes(y)/no(n)]: y
```

12단계. 클러스터 관리자에서 이 명령을 실행하여 중재자가 제거되었는지 확인합니다. `set07`, 출력 `set07` 현재 중재자를 포함할 수 없습니다.

```
#diagnostics.sh --get_replica_status
```

Expected output:

```

-----|
|-----|
|-----|
| SESSION:set07 |
| Status via sessionmgr01:27727 sessionmgr02:27727 |
| Member-1 - 27727 : - SECONDARY - sessionmgr01 - ON-LINE - 0 sec -|
| Member-2 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - ----- -|
|-----|
|-----|

```

13단계. 업데이트 **mongoConfig.cfg** 파일을 수정하여 수정된 복제 세트에 알맞은 중재자를 지정합니다. 현재 중재자(ARBITER=arbiter)를 새 중재자(ARBITER=new-arbiter)로 바꿉니다. 클러스터 관리자에서 이 명령을 실행합니다.

```
#vi /etc/broadhop/mongoConfig.cfg
```

현재 구성:

```

[SESSION-SET2]
SETNAME=set07
OPLOG_SIZE=5120
ARBITER=arbitervip:27727
ARBITER_DATA_PATH=/var/data/sessions.7
MEMBER1=sessionmgr02:27727
MEMBER2=sessionmgr01:27727
DATA_PATH=/var/data/sessions.1/2
[SESSION-SET2-END]

```

필수 구성:

```

[SESSION-SET2]
SETNAME=set07
OPLOG_SIZE=5120
ARBITER=new-arbiter:27727
ARBITER_DATA_PATH=/var/data/sessions.7
MEMBER1=sessionmgr02:27727
MEMBER2=sessionmgr01:27727
DATA_PATH=/var/data/sessions.1/2
[SESSION-SET2-END]

```

14단계. 업데이트된 파일 복사 **mongoConfig.cfg** 파일을 모든 VM에 저장합니다. 클러스터 관리자에서 이 명령을 실행합니다.

```
#copytoall.sh /etc/broadhop/mongoConfig.cfg /etc/broadhop/mongoConfig.cfg
```

15단계. set07에 새 중재자 멤버를 추가합니다. 클러스터 관리자에서 다음을 실행합니다.

/var/qps/install/current/scripts/build/build_etc.sh 명령을 사용하여 **/etc/directory**.

16단계. 다음을 실행한 후 새 중재자 멤버가 복제본 세트에 추가되었는지 확인합니다. **build_etc.sh** 스크립트입니다. 이제 AIDO 서버가 새 중재자로 복제본 세트를 생성/업데이트할 때까지 기다려야 합니다.

```
#diagnostics.sh --get_replica_status
```

Expected Output:

```
| SET NAME - PORT : IP ADDRESS - REPLICA STATE - HOST NAME - HEALTH - LAST SYNC -PRIORITY
|-----|
| SESSION:set07 |
| Status via arbitervip:27727 sessionmgr01:27727 sessionmgr02:27727 |
| Member-1 - 27727 : - SECONDARY - sessionmgr01 - ON-LINE - 0 sec - 2 |
| Member-2 - 27727 : xx.xx.xx.xx - ARBITER - new-arbiter - ON-LINE - ----- - 0 |
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - ----- - 3 |
|-----|
```

주: 새 증재자 멤버가 추가되지 않은 경우 다음 단계를 진행합니다. 그렇지 않으면 18단계로 이동합니다.

17단계. 새 증재자 멤버를 강제로 추가하려면 클러스터 관리자에서 이 명령을 실행합니다.

```
#build_set.sh --DB_NAME --add-members --setname Setxxx --force
```

18단계. 증재자 포트가 아직 작동되지 않은 경우 새 증재자 노드에서 이 명령을 실행하여 동일한 포트를 시작합니다.

Command syntax:

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

Sample command:

```
#/etc/init.d/sessionmgr-27727 start
```

19단계. 새 증재자가 성공적으로 추가되었는지 확인합니다.

```
#diagnostics.sh --get_replica_status
```

20단계. 클러스터 관리자에서 이 명령을 실행하여 그에 따라 DB 우선순위를 업데이트합니다.

```
# cd /var/qps/bin/support/mongo/
# ./set_priority.sh --db session
# ./set_priority.sh --db spr
# ./set_priority.sh --db admin
# ./set_priority.sh --db balance
# ./set_priority.sh --db audit
# ./set_priority.sh --db report
```

21단계. 클러스터 관리자에서 이 명령을 실행하여 복제본 세트의 변경 사항을 확인합니다.

```
#diagnostics.sh --get_replica_status
```

Expected Output:

```
| SET NAME - PORT : IP ADDRESS - REPLICA STATE - HOST NAME - HEALTH - LAST SYNC -PRIORITY
|-----|
| SESSION:set07 |
| Status via arbitervip:27727 sessionmgr01:27727 sessionmgr02:27727 |
| Member-1 - 27727 : - SECONDARY - sessionmgr01 - ON-LINE - 0 sec - 2 |
| Member-2 - 27727 : xx.xx.xx.xx - ARBITER - new-arbiter - ON-LINE - ----- - 0 |
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - ----- - 3 |
|-----|
```

22단계. mon_db 스크립트가 pcrfclient VM에서 복원되는지 확인합니다. 그렇지 않은 경우 수동으로

시작해야 합니다.

```
#monsum | grep mon_db_for
```

mon_db 스크립트를 활성화하려면 모든 pcrfclient VM에 로그인하고 다음 명령을 실행합니다.

```
# monit start mon_db_for_lb_failover
```

```
# monit start mon_db_for_callmodel
```

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.