

CPS 복제본 세트에서 세션 관리자의 역할 및 우선 순위를 처리하는 절차

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[문제](#)

[복제 집합에서 Sessionmgr을 주 세션 관리자 및 변경 세션 관리자 우선 순위로 이동하는 절차](#)

[접근 방식 1](#)

[접근 방식 2](#)

소개

이 문서에서는 CPS(Cisco Policy Suite) 복제 세트에서 sessionmgr을 주 역할에서 이동하고 세션 관리자 우선순위를 변경하는 절차에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- 리눅스
- CPS
- 몽고DB

CPS CLI에 대한 권한 루트 액세스 권한이 있어야 합니다.

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 Sessionmgr VM(가상 머신)에서 단일 프로세스가 실행되는 MongoDB를 사용하여 기본 데이터베이스 구조를 구성합니다. ADMIN, SPR(Subscriber Profile Repository), BALANCE, SESSION, REPORTING, AUDIT 등 다양한 용도로 여러 개의 복제본 세트를 보유합니다.

MongoDB의 복제본 세트는 동일한 데이터 집합을 유지하는 단일 프로세스 그룹입니다. 복제 세트는 중복성과 고가용성을 제공합니다. 여러 데이터베이스 서버에 여러 개의 데이터 복제본을 사용하여 로드 공유 읽기 작업을 허용합니다.

복제본 세트에는 데이터가 있는 여러 노드와 선택적으로 한 종재자 노드가 포함됩니다. 데이터가 있는 노드 중 한 개 및 한 개의 멤버만 기본 노드로 간주되고 다른 노드는 보조 노드로 간주됩니다 (복제 세트는 여러 보조를 가질 수 있음). 기본 노드는 모든 쓰기 작업을 처리합니다.

보조는 운영 로그의 작업 로그(oplog)를 복제하고 보조 데이터 세트가 기본 데이터 세트를 반영하도록 해당 데이터 세트에 작업을 적용합니다. 기본 를 사용할 수 없는 경우, 적합한 보조 는 새 주 를 선택할 수 있는 선택을 보유합니다. 종재자는 선거에 참여하지만 데이터를 가지고 있지 않습니다.

복제본 세트 상태를 가져오려면 ClusterManager 또는 pcrfclient에서 **diagnostics.sh --get_r** 명령을 실행합니다.

여기에 예제 복제 세트 가 있습니다. **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 |
|-----|
|
```

복제본 세트 구성 정보를 가져오려면 다음 단계를 수행합니다.

1단계. 해당 복제본 세트의 기본 MongoDB 멤버에 로그인합니다. ClusterManager에서 이 명령을 실행합니다.

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

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

2단계. 복제 세트 구성 정보를 가져오려면 명령을 실행합니다.

```
set07:PRIMARY> rs.conf()
{
  "_id" : "set07",
  "version" : 2,
  "members" : [
    {
      "_id" : 0,
      "host" : "sessionmgr01:27727",
      "arbiterOnly" : false,
      "buildIndexes" : true,
```

```

"hidden" : false,
"priority" : 2,
"tags" : {
},
"slaveDelay" : NumberLong(0),
"votes" : 1
},
{
"_id" : 1,
"host" : "arbitervip:27727",
"arbiterOnly" : true,
"buildIndexes" : true,
"hidden" : false,
"priority" : 0,
"tags" : {
},
"slaveDelay" : NumberLong(0),
"votes" : 1
},
{
"_id" : 2,
"host" : "sessionmgr02:27727",
"arbiterOnly" : false,
"buildIndexes" : true,
"hidden" : false,
"priority" : 3,
"tags" : {
},
"slaveDelay" : NumberLong(0),
"votes" : 1
}
],
"settings" : {
"chainingAllowed" : true,
"heartbeatIntervalMillis" : 2000,
"heartbeatTimeoutSecs" : 1,
"electionTimeoutMillis" : 10000,
"catchUpTimeoutMillis" : -1,
"catchUpTakeoverDelayMillis" : 30000,
"getLastErrorModes" : {
},
"getLastErrorDefaults" : {
"w" : 1,
"wtimeout" : 0
},
"replicaSetId" : ObjectId("61cdb17a80b097a2e7604c97")
}
}
set07:PRIMARY>

```

참고: 복제본 세트에서 우선 순위가 가장 높은 Sessionmgr은 주 멤버로 작동합니다.

문제

sessionmgr이 하나 이상의 복제본 세트에서 주 구성원의 역할을 수행하고 이러한 경우 복제본 세트 주 역할을 다른 세션 관리자로 이동해야 한다고 가정합니다.

1. 원활한 전환을 위해 이 세션 관리자 VM 종료와 관련된 작업을 수행할 때마다

2. 세션 관리자 상태가 어떤 이유로 저하된 경우 다른 정상 세션 관리자(sessionmgr)와 복제 세트(Replica Set)의 적절한 기능을 유지 관리합니다.

복제 집합에서 Sessionmgr을 주 세션 관리자 및 변경 세션 관리자 우선 순위로 이동하는 절차

접근 방식 1

여기서 복제본 세트의 sessionmgr 우선 순위가 MongoDB 수준에서 직접 변경되었습니다. **set07**의 기본 역할에서 sessionmgr02를 이동하는 단계는 다음과 같습니다.

옵션 1. sessionmgr02의 우선순위를 변경합니다.

1단계. 해당 복제본 세트의 기본 MongoDB 멤버에 로그인합니다.

Command template:

```
#mongo --host <sessionmgrXX> --port <Replica Set port>
```

Sample command:

```
#mongo --host sessionmgr02 --port 27727
```

2단계. 명령을 실행하여 복제본 세트 구성 정보를 가져옵니다.

```
set07:PRIMARY> rs.conf()
{
  "_id" : "set07",
  "version" : 2,
  "members" : [
    {
      "_id" : 0, -----> Position 0
      "host" : "sessionmgr01:27727",
      "arbiterOnly" : false,
      "buildIndexes" : true,
      "hidden" : false,
      "priority" : 2,
      "tags" : {

    },
    "slaveDelay" : NumberLong(0),
    "votes" : 1
  },
  {
      "_id" : 1, -----> Position 1
      "host" : "arbitervip:27727",
      "arbiterOnly" : true,
      "buildIndexes" : true,
      "hidden" : false,
      "priority" : 0,
      "tags" : {

    },
    "slaveDelay" : NumberLong(0),
    "votes" : 1
  },
  },
  "slaveDelay" : NumberLong(0),
  "votes" : 1
},
}
```

```

{
  "_id" : 2, -----> Position 2
  "host" : "sessionmgr02:27727",
  "arbiterOnly" : false,
  "buildIndexes" : true,
  "hidden" : false,
  "priority" : 3,
  "tags" : {

},
  "slaveDelay" : NumberLong(0),
  "votes" : 1
}
],
"settings" : {
  "chainingAllowed" : true,
  "heartbeatIntervalMillis" : 2000,
  "heartbeatTimeoutSecs" : 1,
  "electionTimeoutMillis" : 10000,
  "catchUpTimeoutMillis" : -1,
  "catchUpTakeoverDelayMillis" : 30000,
  "getLastErrorModes" : {

},
  "getLastErrorDefaults" : {
    "w" : 1,
    "wtimeout" : 0
  },
  "replicaSetId" : ObjectId("61cdb17a80b097a2e7604c97")
}
}
set07:PRIMARY>

```

참고: rs.conf() 출력에서 각 sessionmgr의 위치를 기록합니다.

3단계. 터미널을 컨피그레이션 모드로 이동하려면 이 명령을 실행합니다.

```

set07:PRIMARY> cfg = rs.conf()
{
  "_id" : "set07",
  "version" : 2,
  "members" : [
    {
      "_id" : 0,
      "host" : "sessionmgr01:27727",
      "arbiterOnly" : false,
      "buildIndexes" : true,
      "hidden" : false,
      "priority" : 2,
      "tags" : {

},
      "slaveDelay" : NumberLong(0),
      "votes" : 1
    },
    {
      "_id" : 1,
      "host" : "arbitervip:27727",
      "arbiterOnly" : true,
      "buildIndexes" : true,
      "hidden" : false,

```

```

"priority" : 0,
"tags" : {
},
"slaveDelay" : NumberLong(0),
"votes" : 1
},
{
"_id" : 2,
"host" : "sessionmgr02:27727",
"arbiterOnly" : false,
"buildIndexes" : true,
"hidden" : false,
"priority" : 3,
"tags" : {
},
"slaveDelay" : NumberLong(0),
"votes" : 1
}
],
"settings" : {
"chainingAllowed" : true,
"heartbeatIntervalMillis" : 2000,
"heartbeatTimeoutSecs" : 1,
"electionTimeoutMillis" : 10000,
"catchUpTimeoutMillis" : -1,
"catchUpTakeoverDelayMillis" : 30000,
"getLastErrorModes" : {
},
"getLastErrorDefaults" : {
"w" : 1,
"wtimeout" : 0
},
"replicaSetId" : ObjectId("61cdb17a80b097a2e7604c97")
}
}
set07:PRIMARY>

```

4단계. 이 명령을 실행하여 sessionmgr의 우선순위를 변경합니다.

Command template:

```
cfg.members[X].priority = X --> put the position here in [].
```

sample command:

```
cfg.members[2].priority = 1
```

여기서 sessionmgr02는 현재 기본 구성원이며 해당 직책은 2이고 우선 순위는 3입니다.

이 sessionmgr02를 기본 역할에서 이동하려면 우선 순위가 가장 높은 보조 구성원의 우선 순위(예: 1).

```
set07:PRIMARY> cfg.members[2].priority = 1
```

```
1
```

```
set07:PRIMARY>
```

5단계. 이 명령을 실행하여 변경 사항을 커밋합니다.

```
set07:PRIMARY> rs.reconfig(cfg)
{
"ok" : 1,
"operationTime" : Timestamp(1641528658, 1),
"$clusterTime" : {
"clusterTime" : Timestamp(1641528658, 1),
"signature" : {
"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"keyId" : NumberLong(0)
}
}
}
2022-01-07T04:10:57.280+0000 I NETWORK [thread1] trying reconnect to sessionmgr02:27727
(192.168.10.140) failed
2022-01-07T04:10:57.281+0000 I NETWORK [thread1] reconnect sessionmgr02:27727 (192.168.10.140)
ok
set07:SECONDARY>
```

6단계. 명령을 다시 실행하여 sessionmgr 우선 순위의 변경 사항을 확인합니다.

```
set07:SECONDARY> rs.conf()
{
"_id" : "set07",
"version" : 3,
"members" : [
{
"_id" : 0,
"host" : "sessionmgr01:27727",
"arbiterOnly" : false,
"buildIndexes" : true,
"hidden" : false,
"priority" : 2,
"tags" : {

},
"slaveDelay" : NumberLong(0),
"votes" : 1
},
{
"_id" : 1,
"host" : "arbitervip:27727",
"arbiterOnly" : true,
"buildIndexes" : true,
"hidden" : false,
"priority" : 0,
"tags" : {

},
"slaveDelay" : NumberLong(0),
"votes" : 1
},
{
"_id" : 2,
"host" : "sessionmgr02:27727",
"arbiterOnly" : false,
"buildIndexes" : true,
"hidden" : false,
"priority" : 1, --> Here priority has been changed from 3 to 1.
"tags" : {

},
"slaveDelay" : NumberLong(0),
```

```

"votes" : 1
}
],
"settings" : {
"chainingAllowed" : true,
"heartbeatIntervalMillis" : 2000,
"heartbeatTimeoutSecs" : 1,
"electionTimeoutMillis" : 10000,
"catchUpTimeoutMillis" : -1,
"catchUpTakeoverDelayMillis" : 30000,
"getLastErrorModes" : {
},
"getLastErrorDefaults" : {
"w" : 1,
"wtimeout" : 0
},
"replicaSetId" : ObjectId("61cdb17a80b097a2e7604c97")
}
}
set07:SECONDARY>

```

7단계. **diagnostics.sh** 명령을 실행하여 ClusterManager 또는 pcrfclient에서 **get_r**을 실행하여 복제본 세트 상태의 변경 사항을 확인합니다.

```

| 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 : - PRIMARY - sessionmgr01 - ON-LINE - ----- - 2 |
| Member-2 - 27727 : 192.168.10.146 - ARBITER - arbitervip - ON-LINE - ----- - 0 |
| Member-3 - 27727 : - SECONDARY - sessionmgr02 - ON-LINE - 0 sec - 1 |
|-----|

```

sessionmgr02가 보조로 이동했습니다. sessionmgr02를 다시 기본 멤버로 만들려면 4단계에서 이 명령을 사용하여 앞서 설명한 단계 1. - 5를 실행하십시오.

cfg.members[2].priority = 2보다 크고 1001보다 작은 숫자 → 는 샘플에 있는 현재 기본 멤버의 우선 순위보다 높은 우선 순위를 지정합니다.

```

set07:PRIMARY> cfg.members[2].priority = 5
5
set07:PRIMARY> rs.reconfig(cfg)
{
"ok" : 1,
"operationTime" : Timestamp(1641531450, 1),
"$clusterTime" : {
"clusterTime" : Timestamp(1641531450, 1),
"signature" : {
"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"keyId" : NumberLong(0)
}
}
}
2022-01-07T04:57:31.247+0000 I NETWORK [thread1] trying reconnect to sessionmgr01:27727 (192.168.10.139) failed
2022-01-07T04:57:31.247+0000 I NETWORK [thread1] reconnect sessionmgr01:27727 (192.168.10.139)
ok
set07:SECONDARY>

```


sessionmgr 우선 순위의 변경 사항을 확인하려면 명령을 실행합니다.

```
set07:SECONDARY> rs.conf()
{
  "_id" : "set07",
  "version" : 4,
  "members" : [
    {
      "_id" : 0,
      "host" : "sessionmgr01:27727",
      "arbiterOnly" : false,
      "buildIndexes" : true,
      "hidden" : false,
      "priority" : 2,
      "tags" : {

    },
    "slaveDelay" : NumberLong(0),
    "votes" : 1
  },
  {
    "_id" : 1,
    "host" : "arbitervip:27727",
    "arbiterOnly" : true,
    "buildIndexes" : true,
    "hidden" : false,
    "priority" : 0,
    "tags" : {

  },
  "slaveDelay" : NumberLong(0),
  "votes" : 1
},
{
  "_id" : 2,
  "host" : "sessionmgr02:27727",
  "arbiterOnly" : false,
  "buildIndexes" : true,
  "hidden" : false,
  "priority" : 5, --> Here priority has been changed from 1 to 5.
  "tags" : {

},
"slaveDelay" : NumberLong(0),
"votes" : 1
}
],
"settings" : {
  "chainingAllowed" : true,
  "heartbeatIntervalMillis" : 2000,
  "heartbeatTimeoutSecs" : 1,
  "electionTimeoutMillis" : 10000,
  "catchUpTimeoutMillis" : -1,
  "catchUpTakeoverDelayMillis" : 30000,
  "getLastErrorModes" : {

},
  "getLastErrorDefaults" : {
    "w" : 1,
    "wtimeout" : 0
  },
  "replicaSetId" : ObjectId("61cdb17a80b097a2e7604c97")
}
```

```
}
}
set07:SECONDARY>
```

ClusterManager 또는 pcrfclient에서 **diagnostics.sh** 명령을 실행하여 복제본 세트 상태의 변경 사항을 확인합니다.

```
| 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 - 14 sec - 2 |
| Member-2 - 27727 : 192.168.10.146 - ARBITER - arbitervip - ON-LINE - ----- - 0 |
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - ----- - 5 |
|-----|
```

이제 sessionmgr02가 다시 기본 유닛이 되었습니다.

옵션 2. 다른 보조 세션 관리자의 우선순위를 변경하여 주 멤버로 만듭니다. sessionmgr01입니다.

sessionmgr01을 기본 멤버로 만들려면 4단계에서 이 명령을 사용하여 옵션 1에서 단계 1.부터 5까지 앞서 설명한 단계를 실행합니다.

cfg.members[0].priority = 3보다 크고 1001보다 작은 숫자 → 샘플의 현재 기본 멤버 중 "3"보다 높은 우선 순위를 지정합니다.

```
set07:PRIMARY> cfg.members[0].priority = 4
4
set07:PRIMARY> rs.reconfig(cfg)
{
  "ok" : 1,
  "operationTime" : Timestamp(1641540587, 1),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1641540587, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}
2022-01-07T07:29:46.141+0000 I NETWORK [thread1] trying reconnect to sessionmgr02:27727 (192.168.10.140) failed
2022-01-07T07:29:46.142+0000 I NETWORK [thread1] reconnect sessionmgr02:27727 (192.168.10.140)
ok
set07:SECONDARY>
```

명령을 실행하여 변경 사항을 확인합니다.

```
set07:SECONDARY> rs.conf()
{
  "_id" : "set07",
  "version" : 4,
  "members" : [
    {
      "_id" : 0,
      "host" : "sessionmgr01:27727",
      "arbiterOnly" : false,
      "buildIndexes" : true,
```

```

"hidden" : false,
"priority" : 4, --> Here priority has been changed from 2 to 4.
"tags" : {

},
"slaveDelay" : NumberLong(0),
"votes" : 1
},
{
"_id" : 1,
"host" : "arbitervip:27727",
"arbiterOnly" : true,
"buildIndexes" : true,
"hidden" : false,
"priority" : 0,
"tags" : {

},
"slaveDelay" : NumberLong(0),
"votes" : 1
},
{
"_id" : 2,
"host" : "sessionmgr02:27727",
"arbiterOnly" : false,
"buildIndexes" : true,
"hidden" : false,
"priority" : 3,
"tags" : {

},
"slaveDelay" : NumberLong(0),
"votes" : 1
}
],
"settings" : {
"chainingAllowed" : true,
"heartbeatIntervalMillis" : 2000,
"heartbeatTimeoutSecs" : 1,
"electionTimeoutMillis" : 10000,
"catchUpTimeoutMillis" : -1,
"catchUpTakeoverDelayMillis" : 30000,
"getLastErrorModes" : {

},
"getLastErrorDefaults" : {
"w" : 1,
"wtimeout" : 0
},
"replicaSetId" : ObjectId("61cdb17a80b097a2e7604c97")
}
}

```

set07:SECONDARY>

명령 diagnostics.sh —클러스터 관리자 또는 pcrfclient에서 **get_r**을 실행하여 복제본 세트 상태의 변경 사항을 확인합니다.

```

| 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 : - PRIMARY - sessionmgr01 - ON-LINE - ----- - 4 |

```

```
| Member-2 - 27727 : 192.168.10.146 - ARBITER - arbitervip - ON-LINE - ----- - 0 |
| Member-3 - 27727 : - SECONDARY - sessionmgr02 - ON-LINE - 0 sec - 3 |
```

이제 sessionmgr01이 기본 유닛이 되었지만 sessionmgr02는 보조 유닛이 되었습니다.

sessionmgr02를 다시 기본 멤버로 만들려면 옵션 1에서 단계 1~5.를 단계 4에서 이 명령으로 실행합니다.

cfg.members[0].priority = 3보다 작거나 0보다 큰 숫자 → 샘플에서 "3"인 sessionmgr02보다 낮은 우선 순위를 지정합니다.

```
set07:PRIMARY> cfg.members[0].priority = 1
1
set07:PRIMARY> rs.reconfig(cfg)
{
"ok" : 1,
"operationTime" : Timestamp(1641531450, 1),
"$clusterTime" : {
"clusterTime" : Timestamp(1641531450, 1),
"signature" : {
"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"keyId" : NumberLong(0)
}
}
}
2022-01-07T08:34:31.165+0000 I NETWORK [thread1] trying reconnect to sessionmgr01:27727 (192.168.10.139) failed
2022-01-07T08:34:31.165+0000 I NETWORK [thread1] reconnect sessionmgr01:27727 (192.168.10.139) ok
set07:SECONDARY>
```

sessionmgr 우선 순위의 변경 사항을 확인하려면 이 명령을 실행합니다.

```
set07:SECONDARY> rs.conf()
{
  "_id" : "set07",
  "version" : 4,
  "members" : [
    {
      "_id" : 0,
      "host" : "sessionmgr01:27727",
      "arbiterOnly" : false,
      "buildIndexes" : true,
      "hidden" : false,
      "priority" : 1, --> Here priority has been changed from 4 to 1.
      "tags" : {

      },
      "slaveDelay" : NumberLong(0),
      "votes" : 1
    },
    {
      "_id" : 1,
      "host" : "arbitervip:27727",
      "arbiterOnly" : true,
      "buildIndexes" : true,
      "hidden" : false,
```

```

"priority" : 0,
"tags" : {
},
"slaveDelay" : NumberLong(0),
"votes" : 1
},
{
"_id" : 2,
"host" : "sessionmgr02:27727",
"arbiterOnly" : false,
"buildIndexes" : true,
"hidden" : false,
"priority" : 3,
"tags" : {
},
"slaveDelay" : NumberLong(0),
"votes" : 1
}
],
"settings" : {
"chainingAllowed" : true,
"heartbeatIntervalMillis" : 2000,
"heartbeatTimeoutSecs" : 1,
"electionTimeoutMillis" : 10000,
"catchUpTimeoutMillis" : -1,
"catchUpTakeoverDelayMillis" : 30000,
"getLastErrorModes" : {
},
"getLastErrorDefaults" : {
"w" : 1,
"wtimeout" : 0
},
"replicaSetId" : ObjectId("61cdb17a80b097a2e7604c97")
}
}
set07:SECONDARY>

```

ClusterManager 또는 pcrfclient에서 **diagnostics.sh** 명령을 실행하여 복제본 세트 상태의 변경 사항을 확인합니다.

```

| 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 - 14 sec - 1 |
| Member-2 - 27727 : 192.168.10.146 - ARBITER - arbitervip - ON-LINE - ----- - 0 |
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - ----- - 3 |
|-----|
|-----|

```

이제 sessionmgr02가 기본 유닛이 되었지만 sessionmgr01은 보조 유닛이 되었습니다.

접근 방식 2

ClusterManager에서 CPS 스크립트 **set_priority.sh**를 사용하여 복제본 세트의 sessionmgr 우선 순위를 변경할 수 있습니다. 기본적으로 ClusterManager의 /etc/broadhop/mongoConfig.cfg에 정의된 대로 순차(우선 순위가 더 높은 멤버)의 우선 순위입니다.

예를 들어 set07을 사용합니다.

```
[root@installer broadhop]# cat mongoConfig.cfg
[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]
```

복제본 세트 상태를 가져오려면 ClusterManager 또는 pcrfclient에서 `diagnostics.sh --get_r` 명령을 실행합니다.

```
| 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 : 192.168.10.146 - ARBITER - arbitervip - ON-LINE - ----- - 0 |
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - ----- - 3 |
|-----|
```

앞서 설명한 결과를 비교할 때 sessionmgr02가 `/etc/broadhop/mongoConfig.cfg`에서 set07의 첫 번째 멤버[MEMBER1]이므로 sessionmgr02는 기본적으로 set07의 기본 멤버입니다.

여기서 `set_priority.sh` 스크립트를 사용하여 set07의 기본 멤버 역할에서 sessionmgr02를 이동하는 CPS 고가용성 옵션이 제공됩니다.

1단계. 우선순위를 오름차순으로 설정합니다.

Command template:

```
sh set_priority.sh --db arg --replSet arg --asc
```

where ,

--db arg --> arg is database name

[all|session|spr|admin|balance|report|portal|audit|bindings|session_configs|bindings_configs|spr_configs]

--replSet arg -->arg is <setname>

Sample command:

```
sh set_priority.sh --db session --replSet set07 --asc
```

```
[root@installer ~]# sh set_priority.sh --db session --replSet set07 --asc
```

Set priorities is in progress. check log /var/log/broadhop/scripts/set_priority.log to know the status

Setting priority for Replica-Set: SESSION-SET2

INFO Parsing Mongo Config file

INFO Priority set operation is completed for SESSION-SET2

INFO Priority set to the Database members is finished

INFO Validating if Priority is set correctly for Replica-Set: SESSION-SET2

WARNING Mongo Server trying to reconnect while getting config. Attempt #1

```
INFO Validated Priority is set correctly for Replica-Set: SESSION-SET2
Primary member sessionmgr01:27727 found for Replica SESSION-SET2
```

Set priorities process successfully completed.

```
[root@installer ~]#
```

2단계. **diagnostics.sh** 명령을 실행하여 ClusterManager 또는 pcrfclient에서 **get_r**을 확인합니다.

```
| 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 : - PRIMARY - sessionmgr01 - ON-LINE - ----- - 3 |
| Member-2 - 27727 : 192.168.10.146 - ARBITER - arbitervip - ON-LINE - ----- - 0 |
| Member-3 - 27727 : - SECONDARY - sessionmgr02 - ON-LINE - 0 sec - 2 |
|-----|
```

이제 sessionmgr01은 /etc/broadhop/mongoConfig.cfg에 정의된 오름차순으로 설정되었으므로 기본 멤버가 되었습니다.

sessionmgr02를 다시 기본 멤버로 만들려면 이 명령을 실행하십시오.

```
[root@installer ~]# sh set_priority.sh --db session --replSet set07
```

Set priorities is in progress. check log /var/log/broadhop/scripts/set_priority.log to know the status

Setting priority for Replica-Set: SESSION-SET2

INFO Parsing Mongo Config file

INFO Priority set operation is completed for SESSION-SET2

INFO Priority set to the Database members is finished

INFO Validating if Priority is set correctly for Replica-Set: SESSION-SET2

WARNING Mongo Server trying to reconnect while getting config. Attempt #1

INFO Validated Priority is set correctly for Replica-Set: SESSION-SET2

Primary member sessionmgr02:27727 found for Replica SESSION-SET2

Set priorities process successfully completed.

```
[root@installer ~]#
```

참고: 기본적으로 우선 순위는 내림차순으로 설정되었습니다.

diagnostics.sh 명령 —ClusterManager 또는 pcrfclient에서 **get_r**을 실행하여 변경 사항을 확인합니다.

```
| 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 : 192.168.10.146 - ARBITER - arbitervip - ON-LINE - ----- - 0 |
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - ----- - 3 |
|-----|
```

이제 sessionmgr02가 기본 유닛이 되었지만 sessionmgr01은 보조 유닛이 되었습니다.