

CPS에서 CRD를 잘못된 상태에서 복원하는 절차

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[문제](#)

[CRD를 BAD 상태에서 복원하는 절차](#)

[접근 1.](#)

[접근 2.](#)

소개

이 문서에서는 CPS(Cisco Policy Suite) CRD(Custom Reference Data) 테이블을 BAD 상태에서 복원하는 절차에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- 리눅스
- CPS
- 몽고DB

Cisco에서는 다음 권한 액세스를 권장합니다.

- CPS CLI에 대한 루트 액세스
- CPS GUI에 대한 "qns-svn" 사용자 액세스(정책 작성기 및 CPS Central)

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.

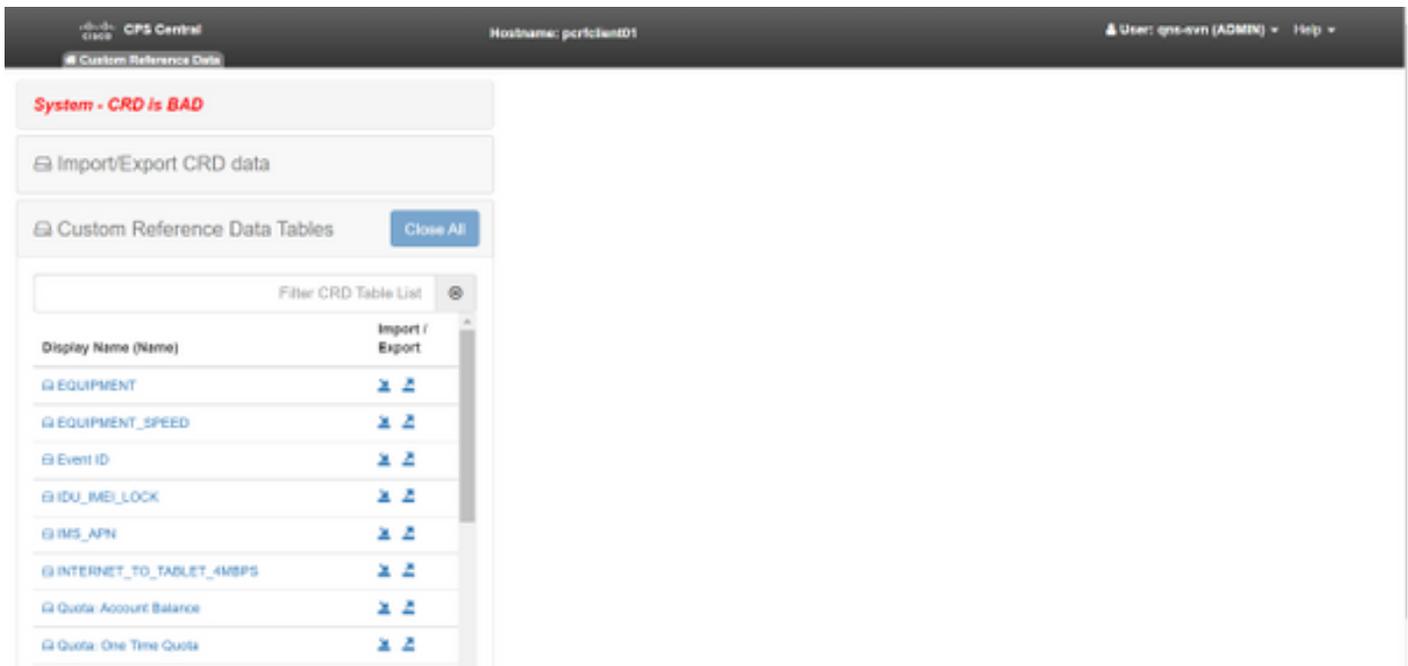
배경 정보

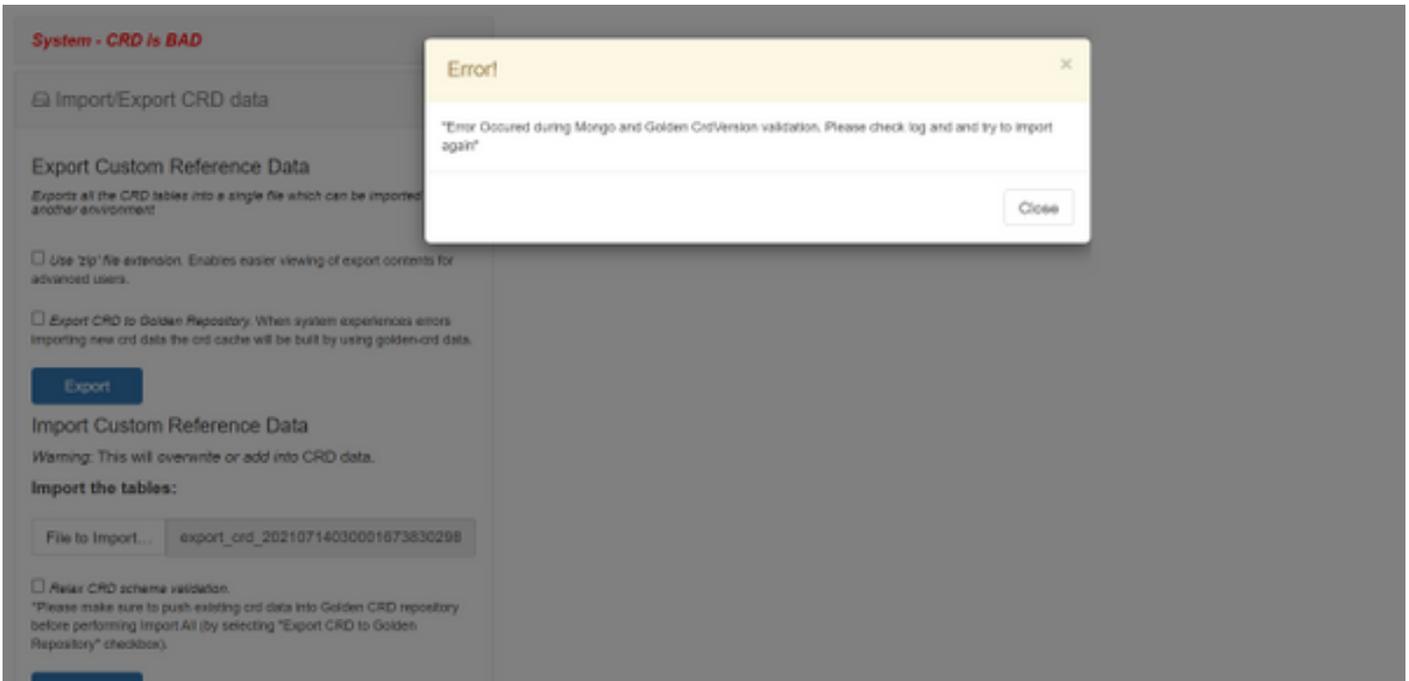
CPS에서 CRD 테이블은 정책 작성기에서 게시되고 sessionmgr에서 호스팅되는 MongoDB 인스턴스에 있는 CRD DB와 연결된 사용자 지정 정책 구성 정보를 저장하는 데 사용됩니다. 내보내기 및 가져오기 작업은 CRD 테이블 데이터를 조작하기 위해 CPS Central GUI를 통해 CRD 테이블에서 수행됩니다.

문제

모든 작업을 가져올 때 어떤 종류의 오류가 발생하면 CPS는 프로세스를 중지하고 시스템을 BAD 상태로 설정하고 CRD API 실행을 차단합니다. CPS는 시스템이 BAD 상태임을 나타내는 오류 응답을 클라이언트에 보냅니다. 시스템이 BAD 상태이고 QNS(Quantum Network Suite)/UDC(User Data Channel) 서버를 다시 시작하면 골든 크리드 데이터를 사용하여 CRD 캐시가 구축됩니다. 시스템 BAD 상태가 FALSE인 경우 CRD 캐시는 MongoDB로 빌드됩니다.

다음은 참조용 CPS Central 오류 이미지입니다.





CRD 시스템이 BAD이면 다음을 수행합니다.

1. CRD 조작이 차단되었습니다. 데이터만 볼 수 있습니다.
2. _import_all, _list, _query를 제외한 CRD API가 차단됩니다.
3. QNS 재시작은 골든 크리드 위치에서 CRD 데이터를 선택합니다.
4. QNS/UDC를 다시 시작하면 시스템 BAD 상태나 통화 드랍을 해결하지 않고 골든 크리드에서 CRD 캐시만 구축됩니다.
5. 골든 크리드 데이터로 구축된 CRD 캐시 시스템 BAD 상태가 FALSE이면 MongoDB를 사용하여 작성된 crd 캐시를 생성합니다.

다음은 CPS qns.log에 연결된 메시지입니다.

```
qns02 qns02 2021-07-29 11:16:50,820 [pool-50847-thread-1]
INFO c.b.c.i.e.ApplicationInterceptor - System -
CRD is in bad state. All CRD APIs (except import all, list and query),
are blocked and user is not allowed to use.
Please verify your crd schema/crd data and try again!
qns02 qns02 2021-07-28 11:33:59,788 [pool-50847-thread-1]
WARN c.b.c.i.e.CustomerReferenceDataManager -
System is in BAD state. Data will be fetched from svn golden-crd repository.
qns01 qns01 2021-07-28 11:55:24,256 [pool-50847-thread-1]
WARN c.b.c.i.e.ApplicationInterceptor - ApplicationInterceptor: Is system bad: true
```

CRD를 BAD 상태에서 복원하는 절차

접근 1.

시스템 상태를 지우려면 CPS Central에서 유효한 CRD 데이터 가져오기와 관련된 Policy Builder에서 유효하고 올바른 CRD 스키마를 가져와야 합니다. 모두 가져오기에 성공하면 시스템 상태가 지워지고 모든 CRD API와 작업이 차단되지 않습니다.

자세한 단계는 다음과 같습니다.

1단계. 이 명령을 실행하여 CRD 데이터베이스를 백업합니다.

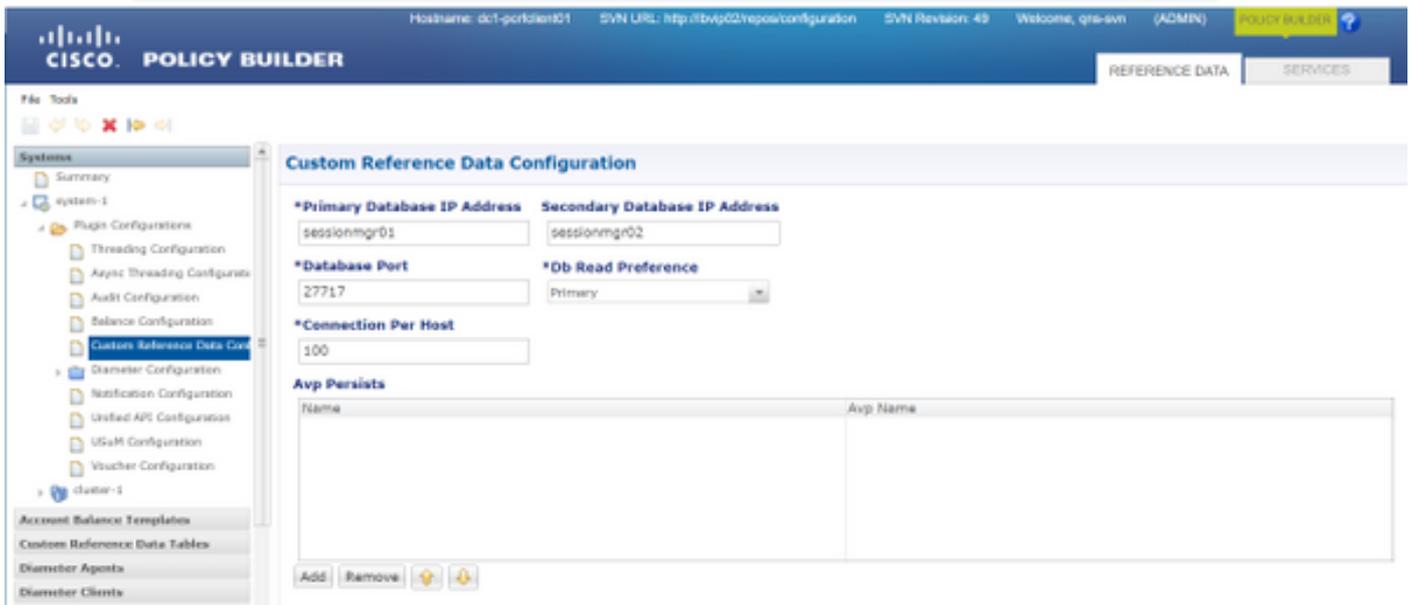
Command template:

```
#mongodump --host <session_manager> --port <cust_ref_data_port>
--db cust_ref_data -o cust_ref_data_backup
```

Sample command:

```
#mongodump --host sessionmgr01 --port 27717 --db cust_ref_data -o cust_ref_data_backup
```

참고: CRD DB 호스트 및 포트의 경우 이 이미지에 표시된 대로 PB의 사용자 지정 참조 데이터 구성을 참조하십시오.



2단계. 이 절차를 사용하여 CRD 테이블(전체 DB)을 삭제합니다.

2.1단계. CRD DB가 있는 mongo 인스턴스에 로그인합니다.

Command template:

```
#mongo --host <sessionmgrXX> --port <cust_ref_data_port>
```

Sample command:

```
#mongo --host sessionmgr01 --port 27717
```

2.2단계. mongo 인스턴스에 있는 모든 DB를 표시하려면 이 명령을 실행합니다.

```
set01:PRIMARY> show dbs
```

```
admin 0.031GB
```

```
config 0.031GB
```

```
cust_ref_data 0.125GB
```

```
local 5.029GB
```

```
session_cache 0.031GB
```

```
sk_cache 0.031GB
```

```
set01:PRIMARY>
```

2.3단계. 이 명령을 실행하여 CRD DB로 전환합니다.

```
set01:PRIMARY> use cust_ref_data
```

```
switched to db cust_ref_data
set01:PRIMARY
```

2.4단계. 이 명령을 실행하여 CRD DB를 삭제합니다.

```
set01:PRIMARY> db.dropDatabase()
{
  "dropped" : "cust_ref_data",
  "ok" : 1,
  "operationTime" : Timestamp(1631074286, 13),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1631074286, 13),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}
```

3단계. show dbs 명령과 함께 이름이 cust_ref_data인 DB가 없는지 확인합니다.

```
set01:PRIMARY> show dbs
admin 0.031GB
config 0.031GB
local 5.029GB
session_cache 0.031GB
sk_cache 0.031GB
set01:PRIMARY>
```

4단계. "qns-svn" 사용자로 정책 작성기에 로그인하고 유효한 CRD 스키마를 게시합니다.

5단계. 클러스터 관리자에서 restart.sh를 사용하여 모든 노드에서 qns 프로세스를 재시작합니다.

6단계. 진단이 정상이고 CRD 테이블에 항목이 없는지 확인합니다. CRD 테이블에는 데이터가 없는 스키마만 있어야 합니다.

7단계. "qns-svn" 사용자로 CPS Central에 로그인하고 유효한 CRD 데이터를 가져옵니다.

8단계. 모든 가져오기를 수행하면 성공 메시지가 반환되고 CPS Central에 표시되지 않는 "system - CRD is BAD" 오류 메시지가 반환되는지 확인합니다.

9단계. 이제 모든 CRD API가 차단되지 않았는지 확인하고 지금 CRD 데이터를 조작할 수 있는지 확인합니다.

첫 번째 접근 방식이 작동하지 않으면 두 번째 접근 방식을 선택합니다.

접근 2.

1단계. ADMIN DB Mongo 인스턴스가 호스팅되는 호스트 및 포트를 diagnostics.sh --get_r 명령으로 식별합니다.

```
[root@installer ~]# diagnostics.sh --get_r
CPS Diagnostics HA Multi-Node Environment
-----
Checking replica sets...
|-----|
|-----|
| Mongo:v3.6.17 MONGODB REPLICAS-SETS STATUS INFORMATION Date : 2021-09-14 02:56:23 |
|-----|
|-----|
| SET NAME - PORT : IP ADDRESS - REPLICAS STATE - HOST NAME - HEALTH - LAST SYNC - PRIORITY |
|-----|
|-----|
| ADMIN:set06 |
| Status via arbitervip:27721 sessionmgr01:27721 sessionmgr02:27721 |
| Member-1 - 27721 : - PRIMARY - sessionmgr01 - ON-LINE - ----- - 3 |
| Member-2 - 27721 : - SECONDARY - sessionmgr02 - ON-LINE - 1 sec - 2 |
| Member-3 - 27721 : 192.168.10.146 - ARBITER - arbitervip - ON-LINE - ----- - 0 |
|-----|
|-----|
```

2단계. ADMIN DB가 있는 mongo 인스턴스에 로그인합니다.

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

```
Sample Command:
#mongo --host sessionmgr01 --port 27721
```

3단계. mongo 인스턴스에 있는 모든 DB를 표시하려면 이 명령을 실행합니다.

```
set06:PRIMARY> show dbs
admin 0.078GB
config 0.078GB
diameter 0.078GB
keystore 0.078GB
local 4.076GB
policy_trace 2.078GB
queueing 0.078GB
scheduler 0.078GB
sharding 0.078GB
set06:PRIMARY>
```

4단계. 이 명령을 실행하여 ADMIN DB로 전환합니다.

```
set06:PRIMARY> use admin
switched to db admin
set06:PRIMARY>
```

5단계. 이 명령을 실행하여 ADMIN DB에 있는 모든 테이블을 표시합니다.

```
set06:PRIMARY> show tables
state
system.indexes
system.keys
system.version
set06:PRIMARY>
```

6단계. 시스템의 현재 상태를 확인하려면 이 명령을 실행합니다.

```
set06:PRIMARY> db.state.find()
```

```
{ "_id" : "state", "isSystemBad" : true, "lastUpdatedDate" : ISODate("2021-08-11T15:01:13.313Z")
}
```

```
set06:PRIMARY>
```

여기서 "isSystemBad"를 볼 수 있습니다. 그렇습니다. 따라서 CRD BAD 상태를 지우려면 다음 단계에서 제공된 명령을 사용하여 이 필드를 "false"로 업데이트해야 합니다.

7단계. **db.state.updateOne({_id:"state"},{\$set:{isSystemBad:false}})** 명령으로 "isSystemBAD" 필드를 업데이트합니다.

```
set06:PRIMARY> db.state.updateOne({_id:"state"},{$set:{isSystemBad:false}})
```

```
{ "acknowledged" : true, "matchedCount" : 0, "modifiedCount" : 0 }
```

```
set06:PRIMARY>
```

8단계. **db.state.find()** 명령을 실행하여 isSystemBad 필드 값이 false로 변경되었는지 확인합니다.

```
set06:PRIMARY> db.state.find()
```

```
{ "_id" : "state", "isSystemBad" : false, "lastUpdatedDate" : ISODate("2021-08-11T15:01:13.313Z") }
```

```
set06:PRIMARY>
```

9단계. 모든 CRD API가 이제 차단되지 않았는지 확인하고 지금 CRD 데이터를 조작할 수 있습니다

.