

CPSでCRDを不良状態から復元する手順

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[問題](#)

[CRDをBAD状態から復元する手順](#)

[アプローチ1](#)

[アプローチ2](#)

概要

このドキュメントでは、BAD状態からCisco Policy Suite(CPS)カスタムリファレンスデータ(CRD)テーブルを復元する手順について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Linux
- CPS
- MongoDB

次の特権アクセス権が必要です。

- CPS CLIへのルートアクセス
- CPS GUIへの「qns-svn」ユーザアクセス (Policy BuilderおよびCPS Central)

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- CPS 20.2
- MongoDB v3.6.17
- UCS-B

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期 (デフォルト) 設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

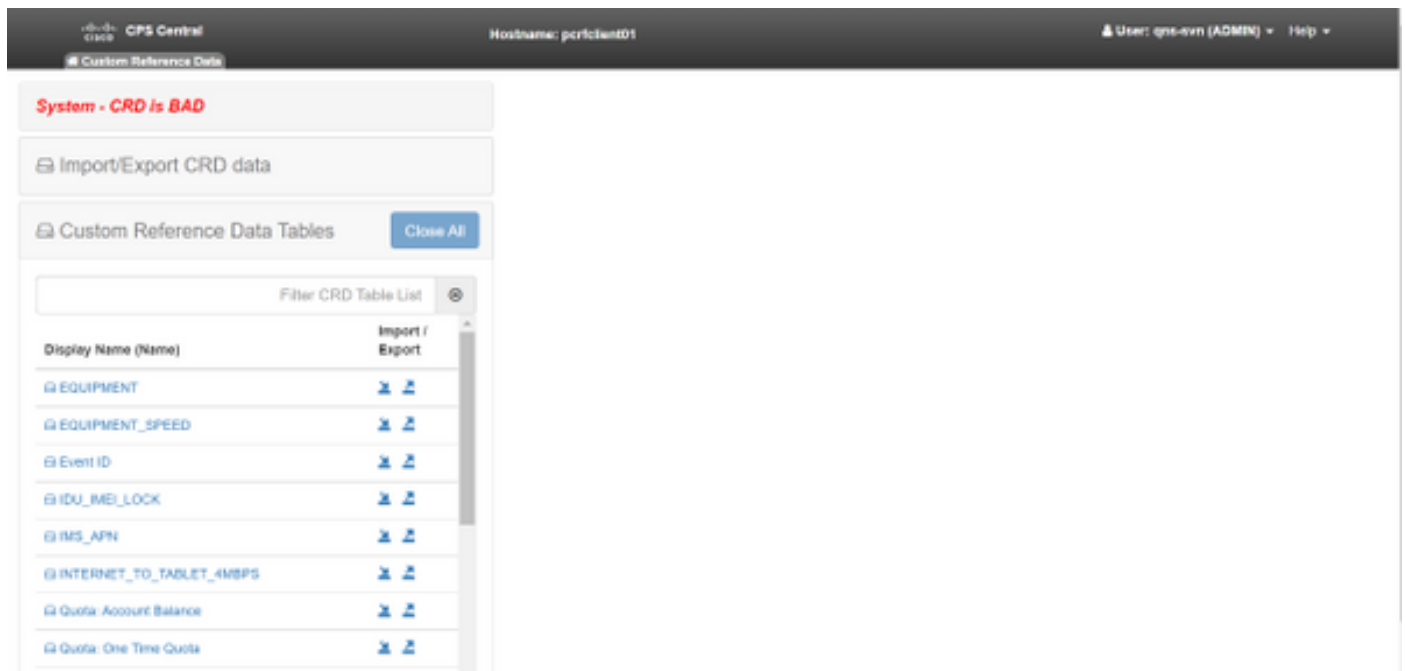
背景説明

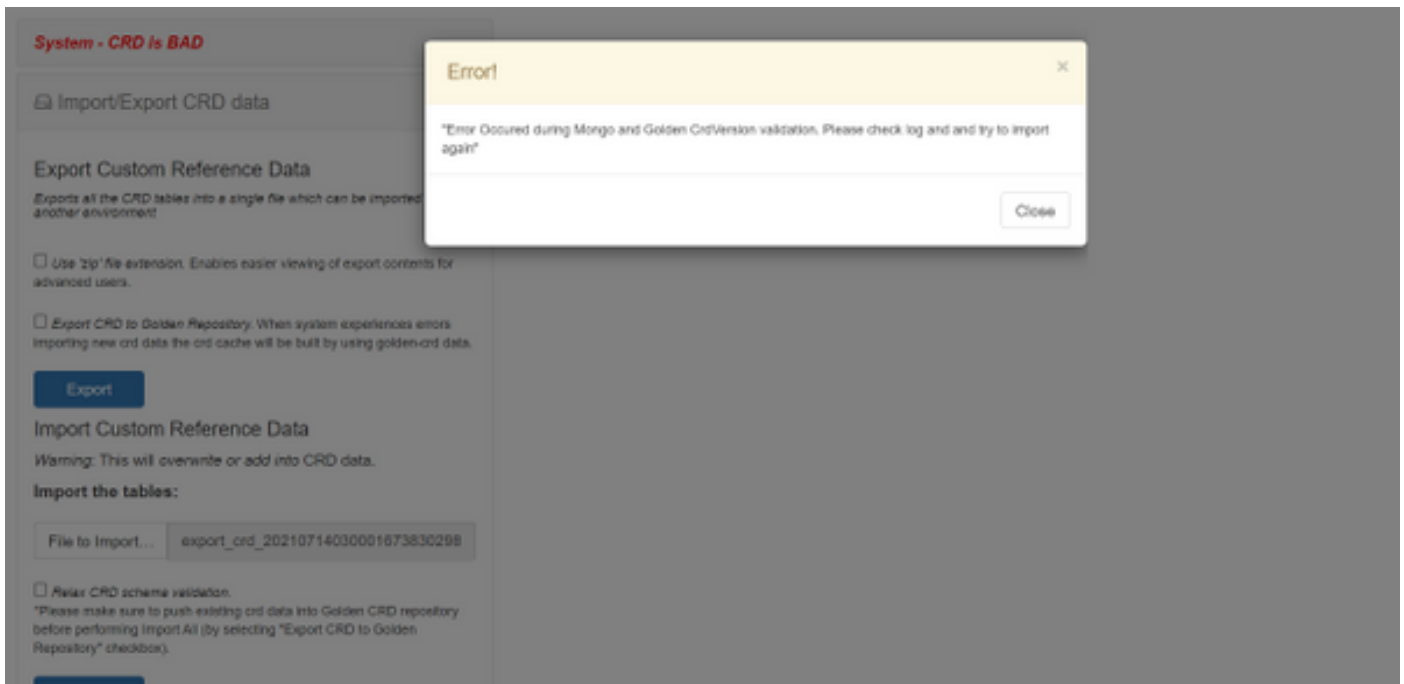
CPSでは、CRDテーブルを使用して、Policy Builderから発行され、sessionmgrでホストされるMongoDBインスタンスに存在するCRD DBに関連付けられたカスタムポリシー設定情報を保存します。エクスポートおよびインポート操作は、CRDテーブルデータを操作するために、CPS Central GUIを使用してCRDテーブルで実行されます。

問題

すべてのインポート操作を実行するときに何らかのエラーが発生した場合、CPSはプロセスを停止し、システムをBAD状態に設定してCRD APIの実行をブロックします。CPSは、システムがBAD状態であることを示すエラー応答をクライアントに送信します。システムがBAD状態で、Quantum Network Suite(QNS)/User Data Channel(UDC)サーバを再起動すると、CRDキャッシュはgolden-crdデータを使用して構築されます。system BAD状態がFALSEの場合、CRDキャッシュはMongoDBで構築されます。

参照用のCPS Central Errorイメージを次に示します。





CRDシステムがBADの場合、次のようになります。

1. CRD操作はブロックされます。表示できるのはデータのみです。
2. `_import_all`、`_list`、`_query`以外のCRD APIはブロックされます。
3. QNS再起動は、golden-crdロケーションからCRDデータをピックアップします。
4. QNS/UDCを再起動しても、システムのBAD状態もコールドドロップも修正されず、golden-crdからCRDキャッシュが構築されるだけです。
5. golden-crdデータで構築されたCRDキャッシュ。システムのBAD状態がFALSEの場合、MongoDBで構築されたcrdキャッシュ。

CPS qns.logの関連メッセージを次に示します。

```
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
```

CRDをBAD状態から復元する手順

アプローチ1

システム状態をクリアするには、CPS Centralからの有効なCRDデータのインポートを含むPolicy Builderから有効で正しいCRDスキーマをインポートする必要があります。インポートが成功すると、システム状態がクリアされ、すべてのCRD APIと操作がブロック解除されます。

詳細な手順は次のとおりです。

ステップ1：このコマンドを実行してCRDデータベースをバックアップします。

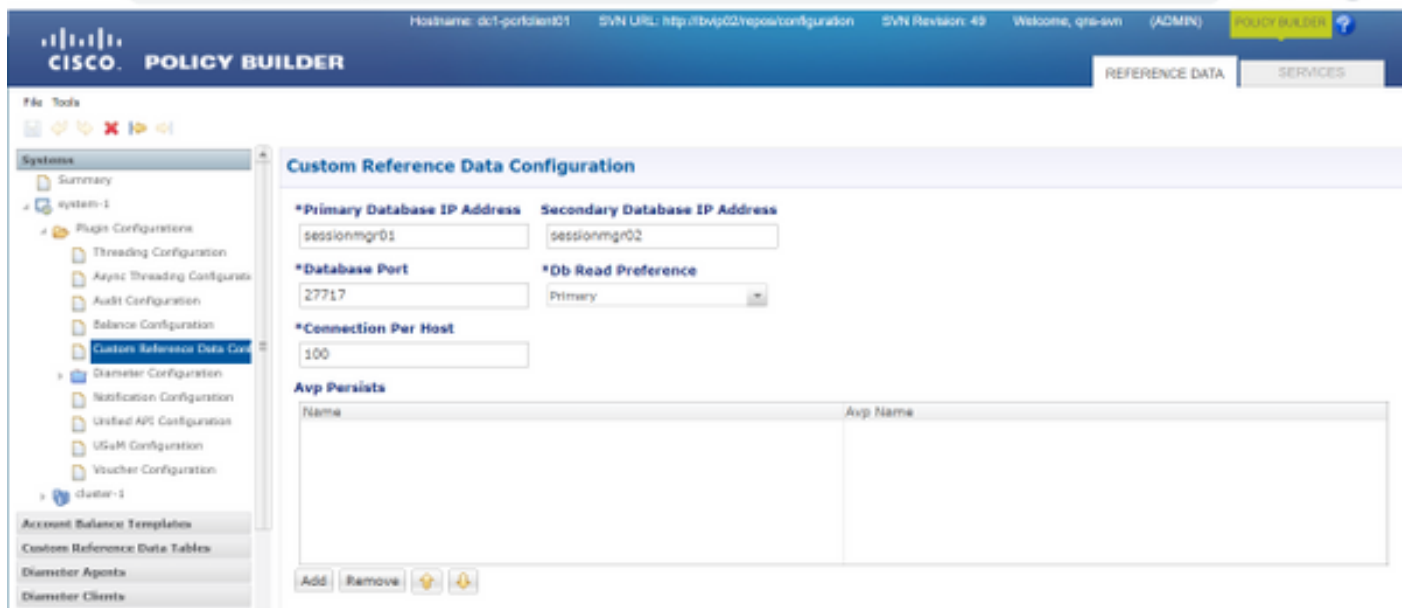
Command template:

```
#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
```

注：CRD DBホストおよびポートについては、次の図に示すように、「PBでのカスタム参照データ設定」を参照してください。



ステップ2：この手順を使用して、CRDテーブル (DB全体) をドロップします。

ステップ2.1:CRD DBがあるmongoインスタンスにログインします。

Command template:

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

Sample command:

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

ステップ2.2：このコマンドを実行して、mongoインスタンスに存在するすべてのDBを表示します。

```
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>
```

ステップ2.3：このコマンドを実行して、CRD DBに切り替えます。

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

ステップ2.4 : このコマンドを実行してCRD DBをドロップします。

```
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)
    }
  }
}
```

ステップ3 : コマンドshow dbsを使用して、cust_ref_dataという名前のdbが存在しないことを確認します。

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

ステップ4: 「qns-svn」ユーザでPolicy Builderにログインし、有効なCRDスキーマを公開します。

ステップ5 : クラスタマネージャからrestartall.shを使用してすべてのノードでqnsプロセスを再起動します。

ステップ6 : 診断が正常で、CRDテーブルにエントリがないことを確認します。CRDテーブルには、データなしでスキーマのみが存在する必要があります。

ステップ7: 「qns-svn」ユーザを使用してCPS Centralにログインし、有効なCRDデータをインポートします。

ステップ8:import all returns successfulメッセージと「system - CRD is BAD」エラーメッセージがCPS Centralに表示されていないことを確認します。

ステップ9 : すべてのCRD APIがブロック解除され、CRDデータを操作できることを確認します。

最初のアプローチが機能しない場合は、2番目のアプローチに進みます。

アプローチ2

ステップ1: コマンド**diagnostics.sh --get_r**を使用して、ADMIN DB Mongoインスタンスがホストされているホストとポートを特定します。

```
[root@installer ~]# diagnostics.sh --get_r
CPS Diagnostics HA Multi-Node Environment
```

```
-----
Checking replica sets...
```

```
|-----|
|-----|
| Mongo:v3.6.17 MONGODB REPLICA-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 |
|-----|
|-----|
```

ステップ2:ADMIN DBがあるmongoインスタンスにログインします。

Command template:

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

Sample Command:

```
#mongo --host sessionmgr01 --port 27721
```

ステップ3: このコマンドを実行して、mongoインスタンスに存在するすべてのDBを表示します

。

```
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>
```

ステップ4: このコマンドを実行して、ADMIN DBに切り替えます。

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

ステップ5: このコマンドを実行して、ADMIN DBにあるすべてのテーブルを表示します。

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

```
set06:PRIMARY>
```

ステップ6：このコマンドを実行して、システムの現在の状態を確認します。

```
set06:PRIMARY> db.state.find()
{ "_id" : "state", "isSystemBad" : true, "lastUpdatedDate" : ISODate("2021-08-11T15:01:13.313Z")
}
```

```
set06:PRIMARY>
```

ここでは、「isSystemBad」が表示されています。true に設定します。したがって、次の手順で提供されるコマンドを使用してCRD BAD状態をクリアするには、このフィールドを「false」に更新する必要があります。

ステップ7：コマンドdb.state.updateOne({_id:"state"},{\$set:{isSystemBad:false}})を使用してフィールド"isSystemBAD"を更新します。

```
set06:PRIMARY> db.state.updateOne({_id:"state"},{$set:{isSystemBad:false}})
{ "acknowledged" : true, "matchedCount" : 0, "modifiedCount" : 0 }
```

```
set06:PRIMARY>
```

ステップ8：コマンドdb.state.find()を実行して、isSystemBadフィールドの値がfalseに変更されたかどうかを確認します。

```
set06:PRIMARY> db.state.find()
{ "_id" : "state", "isSystemBad" : false, "lastUpdatedDate" : ISODate("2021-08-11T15:01:13.313Z") }
```

```
set06:PRIMARY>
```

ステップ9：すべてのCRD APIがブロック解除されたことを確認します。CRDデータを今すぐ操作できます。