

Procédure de restauration de CRD à partir d'un état incorrect dans CPS

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Informations générales](#)

[Problème](#)

[Procédure de restauration de CRD à partir de l'état BAD](#)

[Approche 1.](#)

[Approche 2.](#)

Introduction

Ce document décrit la procédure de restauration de la table CRD (Custom Reference Data) de Cisco Policy Suite (CPS) à partir de l'état BAD.

Conditions préalables

Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Linux
- CPS
- MongoDB

Cisco vous recommande de disposer d'un accès privilégié :

- Accès racine à l'interface CLI CPS
- Accès utilisateur « qns-svn » aux interfaces utilisateur CPS (Policy Builder et CPS Central)

Composants utilisés

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.

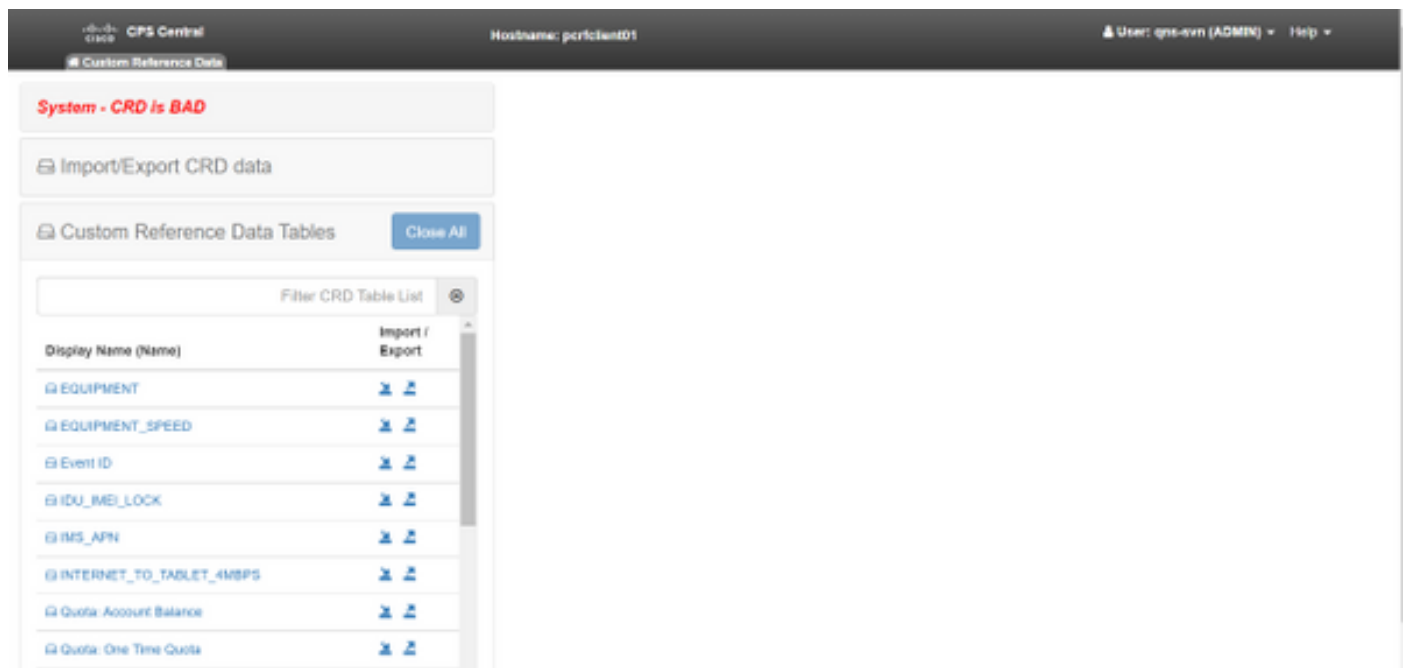
Informations générales

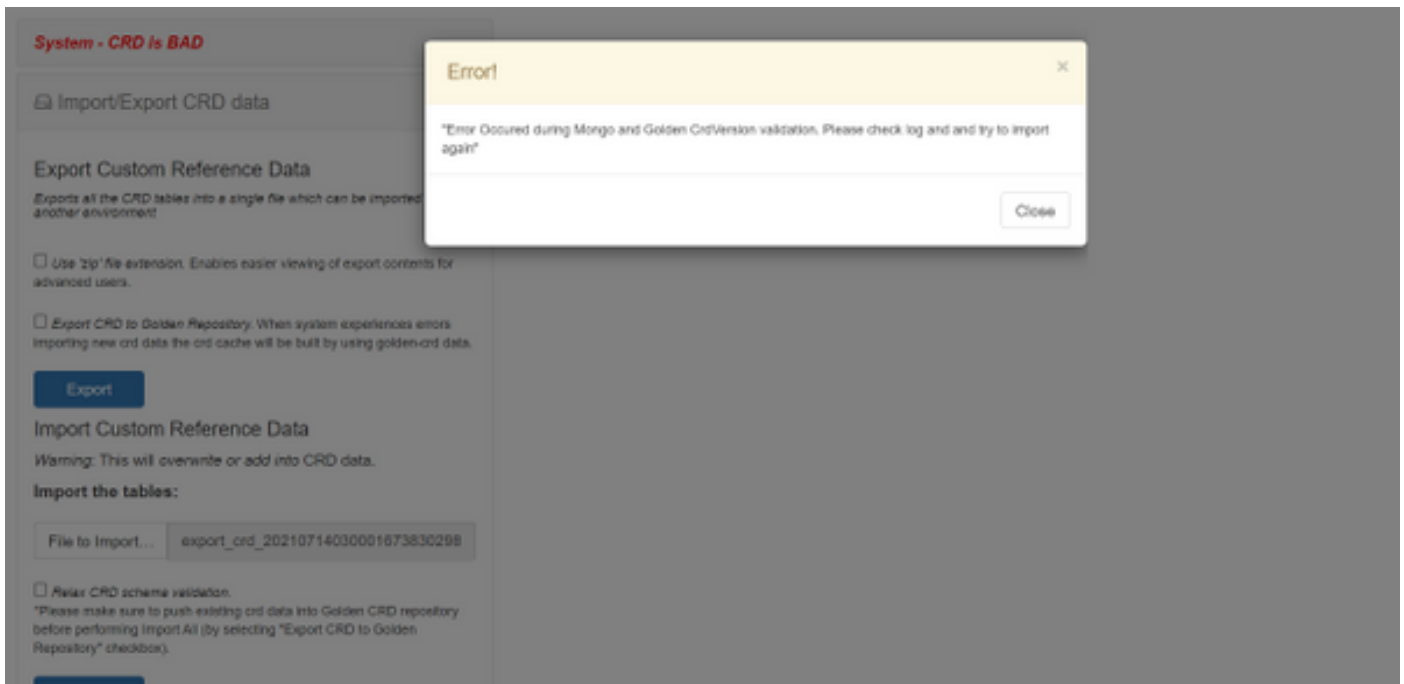
Dans CPS, la table CRD est utilisée pour stocker des informations de configuration de stratégie personnalisées qui sont publiées à partir de Policy Builder et associées à CRD DB qui est présent dans l'instance MongoDB hébergée sur sessionmgr. Les opérations d'exportation et d'importation sont effectuées dans la table CRD via l'interface utilisateur graphique CPS Central afin de manipuler les données de la table CRD.

Problème

En cas d'erreur quelconque lors de l'importation de toutes les opérations, CPS arrête le processus, définit le système dans l'état BAD et bloque l'exécution des API CRD. CPS envoie une réponse d'erreur au client qui indique que le système est dans l'état BAD. Si le système est dans l'état BAD et que vous redémarrez le serveur QNS (Quantum Network Suite)/UDC (User Data Channel), le cache CRD est généré par l'utilisation de données golden-crd. Si l'état BAD du système est FALSE, le cache CRD est construit avec MongoDB.

Voici des images d'erreur CPS Central à titre de référence.





Si le système CRD est MAUVAIS, alors :

1. La manipulation CRD est bloquée. Vous ne pouvez afficher que les données.
2. Les API CRD, à l'exception de `_import_all`, `_list`, `_query`, sont bloquées.
3. QNS restart récupère les données CRD à partir de l'emplacement `golden-crd`.
4. Un redémarrage de QNS/UDC ne corrige pas l'état BAD du système ni les abandons d'appel, il construit uniquement le cache CRD à partir de `golden-crd`.
5. Cache CRD construit avec des données de carte d'or. Si l'état BAD du système est FALSE, alors le cache de carte est construit avec MongoDB.

Voici les messages associés dans CPS qns.log :

```
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
```

Procédure de restauration de CRD à partir de l'état BAD

Approche 1.

Afin d'effacer l'état du système, vous devez importer un schéma CRD valide et correct à partir de Policy Builder qui implique l'importation de données CRD valides à partir de CPS Central. Si l'importation réussit, elle efface l'état du système et toutes les API et opérations CRD sont débloquées.

Voici les étapes détaillées :

Étape 1. Exécutez cette commande pour sauvegarder la base de données CRD.

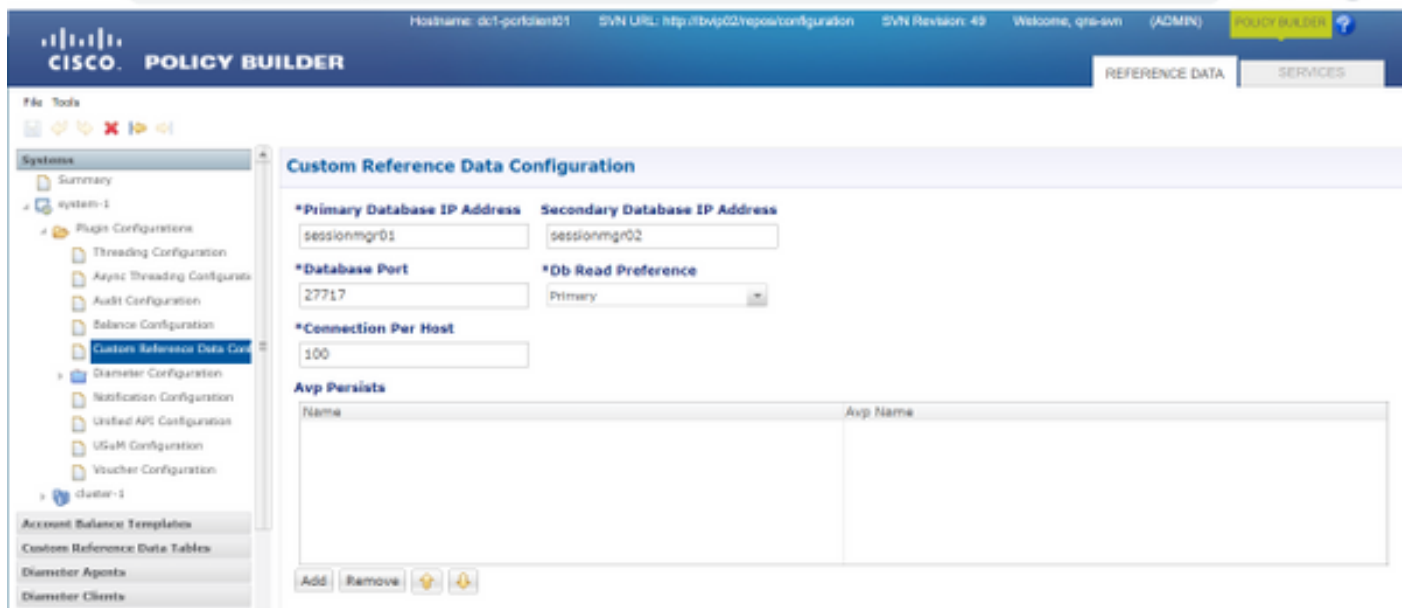
Command template:

```
#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
```

Note: Pour l'hôte et le port de la base de données CRD, référez-vous à Configuration des données de référence personnalisée en PB, comme illustré dans cette image.



Étape 2. Déposez la table CRD (DB entier) à l'aide de cette procédure.

Étape 2.1. Connectez-vous à l'instance mongo où CRD DB est présent.

Command template:

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

Sample command:

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

Étape 2.2. Exécutez cette commande afin d'afficher toutes les DB présentes dans l'instance de mongo.

```
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>
```

Étape 2.3. Exécutez cette commande pour passer à CRD DB.

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

Étape 2.4. Exécutez cette commande pour supprimer la base de données CRD.

```
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)
    }
  }
}
```

Étape 3. Vérifiez qu'il n'y a pas de base de données avec le nom `cust_ref_data` qui existe avec la commande **show dbs**.

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

Étape 4. Connectez-vous à Policy Builder avec l'utilisateur « `qns-svn` » et publiez un schéma CRD valide.

Étape 5. Redémarrez le processus `qns` sur tous les noeuds avec **reartall.sh** à partir du Gestionnaire de cluster.

Étape 6. Vérifiez que les diagnostics sont corrects et qu'il n'y a aucune entrée dans la table CRD. Il ne doit y avoir que le schéma présent dans les tables CRD, c'est-à-dire.. sans données.

Étape 7. Connectez-vous à CPS Central avec l'utilisateur « `qns-svn` » et importez des données CRD valides.

Étape 8. Vérifiez que, l'importation de tous les retours de message réussi et du message d'erreur « `system - CRD is BAD` » non affiché dans CPS Central.

Étape 9. Vérifiez que toutes les API CRD sont maintenant débloquées, vous pouvez manipuler les données CRD maintenant.

Si la première approche n'a pas fonctionné, alors optez pour la deuxième approche.

Approche 2.

Étape 1. Identifiez l'hôte et le port dans lesquels l'instance de la base de données ADMIN est hébergée à l'aide de la commande **diagnostics.sh --get_r**.

```
[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 - REPLICA 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 |
|-----|
|-----|
```

Étape 2. Connectez-vous à l'instance mongo où la base de données ADMIN est présente.

Command template:

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

Sample Command:

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

Étape 3. Exécutez cette commande pour afficher toutes les bases de données présentes dans l'instance de mongo.

```
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>
```

Étape 4. Exécutez cette commande pour passer à ADMIN DB.

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

Étape 5. Exécutez cette commande pour afficher toutes les tables présentes dans la base de données ADMIN.

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

```
system.version
set06:PRIMARY>
```

Étape 6. Exécutez cette commande pour vérifier l'état actuel du système.

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

Ici, vous pouvez voir que **"isSystemBad" : vrai**. Par conséquent, vous devez mettre à jour ce champ sur **"false"** afin d'effacer l'état CRD BAD, avec la commande fournie à l'étape suivante.

Étape 7. Mettez à jour le champ « isSystemBAD » avec la commande **db.state.updateOne({_id: »state »},{ \$set:{isSystemBad:false}})**.

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

Étape 8. Exécutez la commande **db.state.find()** afin de vérifier si la valeur du champ **isSystemBad** a été remplacée par **false**.

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

Étape 9. Vérifiez que toutes les API CRD sont maintenant débloquées, vous pouvez manipuler les données CRD maintenant.