

# Configurer le pourcentage de seuil de fragmentation de base de données personnalisé dans CPS

## Table des matières

[Introduction](#)

[Conditions préalables](#)

[Exigences](#)

[Composants utilisés](#)

[Informations générales](#)

[Configurer](#)

[Configurations](#)

[Approche pour CPS hébergé dans OpenStack](#)

[Approche pour CPS hébergé dans VMWare](#)

[Vérifier](#)

[Dépannage](#)

## Introduction

Ce document décrit comment configurer des pourcentages de seuil de fragmentation de base de données personnalisés dans Cisco Policy Suite (CPS).

## Conditions préalables

### Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Linux
- CPS
- base de données mongole

**Remarque** : Cisco recommande que vous disposiez d'un accès utilisateur racine privilégié à l'interface de ligne de commande CPS.

### Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- CPS 20.2
- Système d'informatique unifiée (UCS)-B
- MongoDB v3.6.17

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

CPS utilise MongoDB pour constituer sa structure de base de données (DB).

La fragmentation est une caractéristique de la MongoDB. Par cette alerte, vous surveillez de manière proactive la fragmentation de MongoDB et évitez ainsi une utilisation possible des ressources plus élevées (disque et mémoire) en raison de MongoDB.

CPS génère une alarme SNMP (Simple Network Management Protocol) lorsque le pourcentage de fragmentation MongoDB dépasse une valeur spécifiée.

LES `/etc/collectd.d/dbMonitorList.cfg` Le fichier présent sur les machines virtuelles Sessionmgr contient la liste des bases de données et leurs valeurs de pourcentage de seuil de fragmentation respectives.

## Configurer

### Configurations

### Approche pour CPS hébergé dans OpenStack

Étape 1. À partir de la machine virtuelle Cluster Manager, exécutez cette commande pour créer une sauvegarde du fichier de configuration actuel.

```
#cp /etc/puppet/modules/qps/templates/collectd_worker/collectd.d/dbMonitorList.cfg  
/etc/puppet/modules/qps/templates/collectd_worker/collectd.d/dbMonitorList.cfg.bkp
```

Étape 2. Exécutez cette commande à partir de Cluster Manager pour obtenir la configuration actuelle des machines virtuelles Sessionmgr (pour comparer et valider les modifications ultérieures).

```
#for host in $(hosts-all.sh | grep 'sessionmgr'); do echo checking in $host; ssh $host "cat  
/etc/collectd.d/dbMonitorList.cfg"; done
```

Exemple de sortie :

```
checking in sessionmgr01  
session_cache|session|40  
sk_cache|secondary_key|40  
diameter|endpoints|40  
spr|subscriber|40  
balance_mgmt|account|40  
checking in sessionmgr02  
session_cache|session|40  
sk_cache|secondary_key|40  
diameter|endpoints|40  
spr|subscriber|40  
balance_mgmt|account|40
```

Étape 3. Modifiez le seuil par défaut (40) à la valeur recommandée (par exemple, 60). Exécutez cette commande à partir du Gestionnaire de cluster.

**Remarque** : cette commande modifie le seuil pour toutes les bases de données. Si vous devez mettre à jour le seuil d'une base de données spécifique, mettez à jour le fichier manuellement.

```
#sed -i 's/40/60/g'  
/etc/puppet/modules/qps/templates/collectd_worker/collectd.d/dbMonitorList.cfg
```

Étape 4. Exécutez cette commande pour comparer les fichiers dans le Gestionnaire de cluster afin de valider la modification.

```
#diff /etc/puppet/modules/qps/templates/collectd_worker/collectd.d/dbMonitorList.cfg  
/etc/puppet/modules/qps/templates/collectd_worker/collectd.d/dbMonitorList.cfg.bkp
```

Exemple de sortie :

```
4c4  
<session_cache|session|60  
---  
>session_cache|session|40  
9c9  
<sk_cache|secondary_key|60  
---  
>sk_cache|secondary_key|40  
14c14  
<diameter|endpoints|60  
---  
>diameter|endpoints|40  
19c19  
<spr|subscriber|60  
---  
>spr|subscriber|40  
24c24  
<balance_mgmt|account|60  
---  
>balance_mgmt|account|40
```

Étape 5. Exécutez cette commande pour générer la modification dans le Gestionnaire de cluster.

```
[root@installer ~]# /var/qps/bin/build/build_puppet.sh  
Copying /etc/puppet to /var/qps/images/puppet.tar.gz...  
Creating MD5 Checksum...  
[root@installer ~]#
```

Étape 6. Exécutez cette commande à partir de Cluster Manager pour appliquer la modification dans les machines virtuelles Sessionmgr.

```
[root@installer ~]# for host in $(hosts-all.sh | grep 'sessionmgr'); do echo starting vm-init in  
$host; ssh $host "/etc/init.d/vm-init > /dev/null 2>&1 &"; done  
starting vm-init in sessionmgr01  
starting vm-init in sessionmgr02  
[root@installer ~]#
```

Étape 7. Attendez que la marionnette soit terminée. Exécutez cette commande à partir du Gestionnaire de cluster pour afficher la progression de la configuration Puppet.

```
#for host in $(hosts-all.sh | grep 'sessionmgr' | tail -1); do echo checking in $host; ssh $host  
"tail -f /var/log/puppet.log"; done
```

```
2022-11-08 06:32:23 +0000 Service[whisper](provider=cps) (info): whisper will be managed using  
monit.
```

```
2022-11-08 06:32:23 +0000 Service[whisper](provider=cps) (info): whisper will be managed using  
monit.
```

```
2022-11-08 06:32:23 +0000 /Stage[main]/Whisper/Service[whisper] (notice): Triggered 'refresh'  
from 1 event
```

```
2022-11-08 06:32:27 +0000 Stage[main] (info): Unsheduling all events on Stage[main]
```

```
2022-11-08 06:32:28 +0000 Puppet (notice): Applied catalog in 83.52 seconds
```

```
[Tue Nov 08 06:32:30 +0000 2022] * Completed puppet configuration for dcl-sessionmgr02...
```

```
[Tue Nov 08 06:32:30 +0000 2022] - NTP sync started, check the logs in vm-init.log
```

## Approche pour CPS hébergé dans VMWare

Étape 1. Mettre à jour `/var/qps/config/deploy/csv/Configuration.csv` dans le Gestionnaire de clusters avec le nom de base de données requis et leur valeur de seuil respective en pourcentage. Le format pour fournir la valeur de pourcentage de seuil personnalisée est le suivant (où XX est la valeur numérique du pourcentage...par exemple ; 60).

```
session_cache,XX,  
sk_cache,XX,  
diameter,XX,  
spr,XX,  
balance_mgmt,XX,
```

Exemple de configuration :

```
session_cache,60,  
sk_cache,60,  
diameter,60,  
spr,60,  
balance_mgmt,60,
```

Étape 2. Exécutez ces commandes pour mettre à jour le `/etc/collectd.d/dbMonitorList.cfg` afin qu'il ait les nouvelles valeurs de seuil du fichier Configuration.csv :

```
[root@installer ~]# /var/qps/install/current/scripts/import/import_deploy.sh  
Filenames that will be processed
```

```
AdditionalHosts.csv Configuration.csv DBConfigServer.csv Definitions.csv Hosts.csv  
ReplicationSets.csv SessionCache.csv VLANs.csv VMSpecification.csv SecureConfig.csv  
VipProxyConfiguration.csv DSCPConfig.csv CriticalFiles.csv
```

```
The CSV files in /var/qps/config/deploy/csv are converted to json files in  
/var/qps/config/deploy/json..
```

```
build the hosts file to /var/www/html/hosts...
```

```
build the /etc/hosts file from the json configuration... /etc/hosts is backed to /etc/hosts.back
```

```
Skipping backup of '/etc/hosts' -- no changes detected.
```

```
Redis by default disabled -DenableQueueSystem=false in /etc/broadhop/qns.conf
```

```
Removing feature configs moved to core
```

```
Removing ws feature from pb and pcrf feature file
```

```
Building /etc/broadhop...
```

```
Copying to /var/qps/images/etc.tar.gz...
```

```
Creating MD5 Checksum...
```

```
Generating /etc/broadhop/servers.all
```

```
Rebuilding facts for: 'installer' (aka 'installer')
```

```
Creating md5sum for hosts file to validate later
Rebuilding facts for: 'dcl-lb01' (aka 'lb01')
Rebuilding facts for: 'dcl-sessionmgr01' (aka 'sessionmgr01')
Rebuilding facts for: 'dcl-lb02' (aka 'lb02')
Rebuilding facts for: 'dcl-qns01' (aka 'qns01')
Rebuilding facts for: 'dcl-qns02' (aka 'qns02')
Rebuilding facts for: 'dcl-pcrfclient01' (aka 'pcrfclient01')
Rebuilding facts for: 'dcl-sessionmgr02' (aka 'sessionmgr02')
Rebuilding facts for: 'dcl-pcrfclient02' (aka 'pcrfclient02')
No file for VipProxyConfiguration found
Copying /etc/puppet to /var/qps/images/puppet.tar.gz...
Creating MD5 Checksum...
[root@installer ~]#
```

**Étape 3. Exécutez cette commande à partir de Cluster Manager pour appliquer la modification dans les machines virtuelles Sessionmgr.**

```
[root@installer ~]# for host in $(hosts-all.sh | grep 'sessionmgr'); do echo starting vm-init in
$host; ssh $host "/etc/init.d/vm-init > /dev/null 2>&1 &"; done
starting vm-init in sessionmgr01
starting vm-init in sessionmgr02
[root@installer ~]#
```

**Étape 4. Attendez que la marionnette soit terminée. Exécutez cette commande à partir du Gestionnaire de cluster pour afficher la progression de la configuration Puppet.**

```
#for host in $(hosts-all.sh | grep 'sessionmgr' | tail -1); do echo checking in $host; ssh $host
"tail -f /var/log/puppet.log"; done
```

```
2022-11-08 06:48:34 +0000 Service[whisper](provider=cps) (info): whisper will be managed using
monit.
2022-11-08 06:48:34 +0000 Service[whisper](provider=cps) (info): whisper will be managed using
monit.
2022-11-08 06:48:34 +0000 /Stage[main]/Whisper/Service[whisper] (notice): Triggered 'refresh'
from 1 event
2022-11-08 06:48:39 +0000 Stage[main] (info): Unsheduling all events on Stage[main]
2022-11-08 06:48:40 +0000 Puppet (notice): Applied catalog in 93.27 seconds
[Tue Nov 08 06:48:42 +0000 2022] * Completed puppet configuration for dcl-sessionmgr02...
[Tue Nov 08 06:48:42 +0000 2022] - NTP sync started, check the logs in vm-init.log
```

## Vérifier

Utilisez cette section pour confirmer que votre configuration fonctionne correctement.

Validez la dernière configuration dans les machines virtuelles Sessionmgr et comparez-la avec le résultat de l'étape 2. Exécutez cette commande à partir du Gestionnaire de cluster.

```
[root@installer ~]# for host in $(hosts-all.sh | grep 'sessionmgr'); do echo checking in $host;
ssh $host "cat /etc/collectd.d/dbMonitorList.cfg"; done
checking in sessionmgr01
session_cache|session|60
sk_cache|secondary_key|60
diameter|endpoints|60
spr|subscriber|60
balance_mgmt|account|60
checking in sessionmgr02
session_cache|session|60
sk_cache|secondary_key|60
diameter|endpoints|60
```

```
spr|subscriber|60  
balance_mgmt|account|60  
[root@installer ~]#
```

## Dépannage

Cette section fournit des informations que vous pouvez utiliser pour dépanner votre configuration.

Cette alerte de fragmentation MongoDB a été introduite dans la version 20.1 et n'a pas été mesurée dans les versions précédentes. Par défaut, le seuil de fragmentation est de 40 %. Cette valeur de seuil doit être modifiée en fonction de la taille de votre déploiement, des modèles de trafic (modèles d'appel) et d'autres facteurs de modèles de trafic. Sinon, CPS émet des alertes de dépassement de seuil de fragmentation de base de données non autorisé/non souhaité.

À propos de cette traduction

Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.