

# Configurar porcentaje de umbral de fragmentación de base de datos personalizado en CPS

## Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Antecedentes](#)

[Configurar](#)

[Configuraciones](#)

[Enfoque para CPS alojado en OpenStack](#)

[Enfoque para CPS alojado en VMWare](#)

[Verificación](#)

[Troubleshoot](#)

## Introducción

Este documento describe cómo configurar porcentajes de umbral de fragmentación de base de datos personalizados en Cisco Policy Suite (CPS).

## Prerequisites

## Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

- Linux
- CPS
- MongoDB

**Nota:** Cisco recomienda que tenga acceso de usuario root con privilegios a CPS CLI.

## Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- CPS 20.2
- Unified Computing System (UCS)-B
- MongoDB v3.6.17

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

## Antecedentes

CPS utiliza MongoDB para constituir su estructura de base de datos básica (DB).

La fragmentación es una característica de MongoDB. Con esta alerta, usted monitorea proactivamente la fragmentación de MongoDB y así evita una posible mayor utilización de recursos (disco y memoria) debido a MongoDB.

CPS genera una alarma de protocolo simple de administración de red (SNMP) cuando el porcentaje de fragmentación de MongoDB supera un valor especificado.

`/etc/collectd.d/dbMonitorList.cfg` El archivo presente en las máquinas virtuales (VM) de sessionmgr contiene la lista de bases de datos y sus respectivos valores porcentuales de umbral de fragmentación.

## Configurar

### Configuraciones

### Enfoque para CPS alojado en OpenStack

Paso 1. Desde la VM del Administrador de clústeres, ejecute este comando para crear una copia de seguridad del archivo de configuración actual.

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

Paso 2. Ejecute este comando desde el Administrador de clústeres para obtener la configuración actual de las VM de sessionmgr (para comparar y validar después del cambio).

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

### Ejemplo de resultado

```
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
```

**Paso 3.** Modifique el umbral predeterminado (40) con el valor recomendado (por ejemplo, 60). Ejecute este comando desde el Administrador de clústeres.

**Nota:** Este comando cambia el umbral para todas las bases de datos. Si el requisito es actualizar el umbral para una base de datos específica, actualice el archivo manualmente.

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

**Paso 4.** Ejecute este comando para comparar los archivos del Administrador de clústeres y validar el cambio.

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

### Ejemplo de resultado

```
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
```

**Paso 5.** Ejecute este comando para generar el cambio en el Administrador de clústeres.

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

**Paso 6.** Ejecute este comando desde el Administrador de clústeres para aplicar el cambio en las VM de 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 ~]#
```

**Paso 7.** Esperar a que el títere se complete. Ejecute este comando desde el Administrador de clústeres para ver el progreso de la configuración de títeres.

```
#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 dc1-sessionmgr02...
```

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

## Enfoque para CPS alojado en VMWare

**Paso 1.** Actualice el `/var/qps/config/deploy/csv/Configuration.csv` en el Administrador de clústeres con el nombre de base de datos necesario y su respectivo valor porcentual de umbral. El formato para proporcionar el valor de porcentaje de umbral personalizado es el siguiente (donde XX es el valor numérico de porcentaje...por ejemplo; 60).

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

**Configuración de ejemplo:**

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

**Paso 2.** Ejecute estos comandos para actualizar el `/etc/collectd.d/dbMonitorList.cfg` para que tenga los nuevos valores de umbral del archivo `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: 'dc1-lb01' (aka 'lb01')
Rebuilding facts for: 'dc1-sessionmgr01' (aka 'sessionmgr01')
Rebuilding facts for: 'dc1-lb02' (aka 'lb02')
Rebuilding facts for: 'dc1-qns01' (aka 'qns01')
Rebuilding facts for: 'dc1-qns02' (aka 'qns02')
Rebuilding facts for: 'dc1-pcrfclient01' (aka 'pcrfclient01')
Rebuilding facts for: 'dc1-sessionmgr02' (aka 'sessionmgr02')
Rebuilding facts for: 'dc1-pcrfclient02' (aka 'pcrfclient02')
No file for VipProxyConfiguration found
Copying /etc/puppet to /var/qps/images/puppet.tar.gz...
Creating MD5 Checksum...
[root@installer ~]#
```

**Paso 3. Ejecute este comando desde el Administrador de clústeres para aplicar el cambio en las VM de 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 ~]#
```

**Paso 4. Esperar a que el títere se complete. Ejecute este comando desde el Administrador de clústeres para ver el progreso de la configuración de títeres.**

```
#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 dc1-sessionmgr02...
[Tue Nov 08 06:48:42 +0000 2022] - NTP sync started, check the logs in vm-init.log
```

## Verificación

Utilize esta sección para confirmar que su configuración funcione correctamente.

Valide la configuración más reciente en las VM de sessionManager y compárela con el resultado del paso 2. Ejecute este comando desde el Administrador de clústeres.

```
[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 ~]#
```

## Troubleshoot

En esta sección se brinda información que puede utilizar para resolver problemas en su configuración.

Esta alerta de fragmentación MongoDB se introdujo en 20.1 y no se midió en versiones anteriores. De forma predeterminada, el valor de umbral de fragmentación es 40%. Este valor de umbral debe modificarse en función del tamaño de implementación, los patrones de tráfico (modelos de llamadas) y otros factores de patrones de tráfico. De lo contrario, CPS emite alertas de no fragmentación de base de datos no deseada o con umbral de fragmentación infringido.

## Acerca de esta traducción

Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).