Procedimiento para administrar un nodo de árbitro en un conjunto de réplicas de CPS

Contenido

Introducción

Este documento describe el procedimiento para administrar el nodo Árbitro en el conjunto de réplicas de Cisco Policy Suite (CPS).

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

- Linux
- CPS
- MongoDB

Nota: Cisco recomienda que tenga acceso de raíz 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 y v3.4.16

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). Posee varios conjuntos de réplicas para diversos fines: ADMIN, Subscriber Profile Repository (SPR), BALANCE, SESSION, REPORTING y AUDIT.

Un conjunto de réplicas en MongoDB es un grupo de procesos monopolares que mantienen el mismo conjunto de datos. Los conjuntos de réplicas proporcionan redundancia y alta disponibilidad (HA). Con varias copias de datos en diferentes servidores de base de datos, permite operaciones de lectura de carga compartida.

En algunas circunstancias (por ejemplo, si tiene una instancia primaria y una secundaria pero las restricciones de costo prohíben la adición de otra secundaria), puede seleccionar agregar una instancia monbuena a un conjunto de réplicas como árbitro para votar en las elecciones. Un árbitro tiene exactamente 1 voto electoral. De forma predeterminada, un árbitro tiene prioridad 0.

Los árbitros son instancias únicas que forman parte de un conjunto de réplicas pero no contienen datos (lo que significa que no proporcionan redundancia de datos). Sin embargo, pueden participar en las elecciones. Un árbitro participa en las elecciones para la primaria, pero un árbitro no tiene una copia del conjunto de datos y no puede convertirse en una primaria.

Los árbitros tienen unos requisitos de recursos mínimos y no necesitan hardware dedicado. Puede implementar un árbitro en un servidor de aplicaciones o un host que simplemente supervise la red.

Un árbitro no almacena datos, pero hasta que se agrega el proceso mongood del árbitro al conjunto de réplicas, el árbitro actúa como cualquier otro proceso mongood y comienza con un conjunto de archivos de datos y un diario de tamaño completo.

Aquí hay un conjunto de réplicas de muestra, es decir **set07**.

```
| SET NAME - PORT : IP ADDRESS - REPLICA STATE - HOST NAME - HEALTH - LAST SYNC -PRIORITY
|-------------------------------------------------------------------------------------------------
---------------------------------------|
| SESSION:set07 |
| Status via arbitervip:27727 sessionmgr01:27727 sessionmgr02:27727 |
| Member-1 - 27727 : - SECONDARY - sessionmgr01 - ON-LINE - 0 sec - 2 |
| Member-2 - 27727 : 192.168.10.146 - ARBITER - arbitervip - ON-LINE - -------- - 0 |
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - -------- - 3 |
|-------------------------------------------------------------------------------------------------
---------------------------------------|
```

Problema

Suponga que hay un problema con un árbitro o un requisito para cambiar el árbitro en un conjunto de réplicas, debe quitar el árbitro actual y agregar un nuevo árbitro al conjunto de réplicas.

Procedimiento para administrar un árbitro en un conjunto de réplicas

Paso 1. Verifique la versión del shell mongo en CPS y el nuevo árbitro. Ejecute este comando desde el session mgr principal en el conjunto de réplicas y el nuevo nodo de árbitro.

Ejemplo de salida de sessionmgr:

```
[root@sessionmgr02 ~]# mongo --version
MongoDB shell version v3.6.17
```

Si la versión del shell de mongo es la misma en session mgr principal y en el nuevo árbitro o si la nueva versión del shell de mongo de árbitro es superior, navegue hasta el paso 6.

De lo contrario, si la nueva versión del shell mongo del árbitro es inferior, debe establecer **featureCompatibilityVersion** como valor inferior en la base de datos admin del conjunto de réplicas con los pasos siguientes.

Ejemplo de caso en el que la nueva versión del shell mongo del árbitro es inferior a la de session mgr de CPS:

```
[root@pcrfclient02 ~]# mongo --version
MongoDB shell version v3.4.16
```

Paso 2. Inicie sesión en la instancia mongo principal del conjunto de réplicas.

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

Sample command:
#mongo --host sessionmgr02 --port 27727
```

Paso 3. Ejecute este comando para ver el **featureCompatibilityVersion** en la base de datos de administración del conjunto de réplicas.

```
set07:PRIMARY> db.adminCommand( { getParameter: 1, featureCompatibilityVersion: 1 } )
{
"featureCompatibilityVersion" : {
"version" : "3.6"
},
"ok" : 1,
"operationTime" : Timestamp(1663914140, 1),
"$clusterTime" : {
"clusterTime" : Timestamp(1663914140, 1),
"signature" : {
"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
"keyId" : NumberLong(0)
}
}
}
set07:PRIMARY>
```

Paso 4. Ejecute este comando para setfeatureCompatibilityVersion como 3.4 en la base de datos admin del conjunto de réplicas.

```
set07:PRIMARY> db.adminCommand( { setFeatureCompatibilityVersion: "3.4" } )
{ "ok" : 1 }
set07:PRIMARY>
```

Paso 5. Ejecute este comando para verificar que **featureCompatibilityVersion** ha cambiado a 3.4 en la base de datos de administración del conjunto de réplicas.

```
set07:PRIMARY> db.adminCommand( { getParameter: 1, featureCompatibilityVersion: 1 } )
{ "featureCompatibilityVersion" : { "version" : "3.4" }, "ok" : 1 }
set07:PRIMARY>
```

Paso 6. Inicie sesión en el Administrador de clústeres y modifique el **/var/qps/config/deploy/csv/AdditionalHosts.csv** archivo con los nuevos detalles del árbitro.

```
#vi /var/qps/config/deploy/csv/AdditionalHosts.csv

Provide new arbiter details in this format:

Host Alias IP Address
new-arbiter new-arbiter xx.xx.xx.xx
```

Paso 7. Importe la configuración CSV.

```
#/var/qps/install/current/scripts/import/import_deploy.sh
```

Paso 8. Verificación **/etc/hosts** que se actualizaron con la información de los nuevos árbitros.

```
#cat /etc/hosts | grep arbiter
```

Paso 9. Ejecute este comando para sincronizar **/etc/hosts**.

```
#/var/qps/bin/update/synchosts.sh

Syncing to following QNS Servers:
lb01 lb02 sessionmgr01 sessionmgr02 qns01 qns02 pcrfclient01 pcrfclient02
Do you want to Proceed? (y/n):y
lb01
lb02
sessionmgr01
sessionmgr02
qns01
qns02
pcrfclient01
pcrfclient02
```

Paso 10. Verifique que las secuencias de comandos mon_db se detengan en las VM pcrfclient.

```
#monsum | grep mon_db_for
```

Si se detiene, éste es el resultado:

```
mon_db_for_lb_failover Not monitored Program
mon_db_for_callmodel Not monitored Program
```

Si no se detiene, éste es el resultado:

```
mon_db_for_lb_failover OK Program
mon_db_for_callmodel OK Program
```

> Nota: Si los scripts mon_db no se detienen, ejecute estos comandos en las respectivas VM de pcffclient para detenerlos manualmente.

```
#monit stop mon_db_for_lb_failover
#monit stop mon_db_for_callmodel
```

Paso 11. Ejecute este comando desde pcrfclient01 para quitar el árbitro actual del conjunto de réplicas (set07 es un ejemplo en este paso).

```
#build_set.sh --session --remove-members --setname set07

Please enter the member details which you going to remove from the replica-set
Member:Port --------> arbitervip:27727
arbitervip:27727
Do you really want to remove [yes(y)/no(n)]: y
```

Paso 12. Ejecute este comando desde el Administrador de clústeres para comprobar si se quitó el árbitro de **set07**, el resultado de **set07** no puede contener el árbitro actual.

```
#diagnostics.sh --get_replica_status

Expected output:
----------|
|-------------------------------------------------------------------------------------------------
----------------------------------------|
| SESSION:set07 |
| Status via sessionmgr01:27727 sessionmgr02:27727 |
| Member-1 - 27727 : - SECONDARY - sessionmgr01 - ON-LINE - 0 sec -|
| Member-2 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - -------- -|
|-------------------------------------------------------------------------------------------------
----------------------------------------|
```

Paso 13. Actualice el **mongoConfig.cfg** para tener el árbitro adecuado en el conjunto de réplicas modificado. Sustituya el árbitro actual (ÁRBITRO=árbitro) por el nuevo (ÁRBITRO=nuevo-árbitro). Ejecute este comando desde el Administrador de clústeres.

```
#vi /etc/broadhop/mongoConfig.cfg
```

Configuración actual

```
[SESSION-SET2]
SETNAME=set07
OPLOG_SIZE=5120
ARBITER=arbitervip:27727
ARBITER_DATA_PATH=/var/data/sessions.7
MEMBER1=sessionmgr02:27727
MEMBER2=sessionmgr01:27727
DATA_PATH=/var/data/sessions.1/2
[SESSION-SET2-END]
```

Configuración necesaria:

```
[SESSION-SET2]
SETNAME=set07
OPLOG_SIZE=5120
ARBITER=new-arbiter:27727
ARBITER_DATA_PATH=/var/data/sessions.7
MEMBER1=sessionmgr02:27727
MEMBER2=sessionmgr01:27727
DATA_PATH=/var/data/sessions.1/2
[SESSION-SET2-END]
```

Paso 14. Copie el archivo **mongoConfig.cfg** a todas las VM. Ejecute este comando desde el Administrador de clústeres.

```
#copytoall.sh /etc/broadhop/mongoConfig.cfg /etc/broadhop/mongoConfig.cfg
```

Paso 15. Agregue un nuevo miembro de árbitro a set07. En el Administrador de clústeres, ejecute

/var/qps/install/current/scripts/build/build_etc.sh para generar el comando **/etc/directory**.

Paso 16. Compruebe que el nuevo miembro del árbitro se agrega al conjunto de réplicas después de ejecutar el **build_etc.sh** , ahora debe esperar a que el servidor AIDO cree/actualice los conjuntos de réplicas con el nuevo árbitro.

```
#diagnostics.sh --get_replica_status

Expected Output:

| SET NAME - PORT : IP ADDRESS - REPLICA STATE - HOST NAME - HEALTH - LAST SYNC -PRIORITY
|-----------------------------------------------------------------------------------------------------
----------------------------------------|
| SESSION:set07 |
| Status via arbitervip:27727 sessionmgr01:27727 sessionmgr02:27727 |
| Member-1 - 27727 : - SECONDARY - sessionmgr01 - ON-LINE - 0 sec - 2 |
| Member-2 - 27727 : xx.xx.xx.xx - ARBITER - new-arbiter - ON-LINE - -------- - 0 |
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - -------- - 3 |
|----------------------------------------|
```

> Nota: Si no se agrega el nuevo miembro del árbitro, continúe con los siguientes pasos. Si no, vaya al paso 18.

Paso 17. Ejecute este comando desde el Administrador de clústeres para agregar un nuevo miembro de árbitro de forma forzada.

```
#build_set.sh --DB_NAME --add-members --setname Setxxx --force
```

Paso 18. Si el puerto del árbitro aún no está activo, ejecute este comando desde el nuevo nodo del árbitro para iniciar el mismo.

```
Command syntax:
#/etc/init.d/sessionmgr-XXXXX start

Sample command:
#/etc/init.d/sessionmgr-27727 start
```

Paso 19. Verifique que el nuevo Árbitro se haya agregado correctamente.

```
#diagnostics.sh --get_replica_status
```

Paso 20. Ejecute este comando desde el Administrador de clústeres para actualizar la prioridad de la base de datos en consecuencia.

```
# cd /var/qps/bin/support/mongo/
# ./set_priority.sh --db session
# ./set_priority.sh --db spr
# ./set_priority.sh --db admin
# ./set_priority.sh --db balance
# ./set_priority.sh --db audit
# ./set_priority.sh --db report
```

Paso 21. Ejecute este comando desde el Administrador de clústeres para comprobar los cambios en el conjunto de réplicas.

```
#diagnostics.sh --get_replica_status

Expected Output:

| SET NAME - PORT : IP ADDRESS - REPLICA STATE - HOST NAME - HEALTH - LAST SYNC -PRIORITY
|-------------------------------------------------------------------------------------------------------------
----------------------------------------|
| SESSION:set07 |
| Status via arbitervip:27727 sessionmgr01:27727 sessionmgr02:27727 |
| Member-1 - 27727 : - SECONDARY - sessionmgr01 - ON-LINE - 0 sec - 2 |
| Member-2 - 27727 : xx.xx.xx.xx - ARBITER - new-arbiter - ON-LINE - -------- - 0 |
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - -------- - 3 |
|----------------------------------------|
```

Paso 22. Verifique que los scripts mon_db se restauren en las VM pcrfclient. Si no es así, debe iniciarlos manualmente.

```
#monsum | grep mon_db_for
```

Para habilitar el script mon_db, inicie sesión en todas las máquinas virtuales pcrfclient y ejecute estos comandos:

```
# monit start mon_db_for_lb_failover
# monit start mon_db_for_callmodel
```

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).