

Resolución de problemas del grupo de dispositivos del centro de operaciones en estado CrashLoopBackOff

Contenido

[Introducción](#)

[Acrónimos](#)

[Registros necesarios](#)

[Secuencia para solucionar problemas](#)

[Posibles situaciones que provocan un problema con la restauración de la configuración posterior](#)

[No disponibilidad de configuración](#)

[Restricciones del ciclo de CPU](#)

Introducción

Este documento describe cómo identificar y recuperar el grupo de dispositivos del centro de operaciones en el estado CrashLoopBackOff.

Acrónimos

RCM - Administrador de configuración de redundancia

AAAA-MM-DD hh:mm:ss - Año-Mes-Día Hora:Minuto:segundo

CPU - Unidad central de procesamiento

Registros necesarios

Resultados del comando RCM necesarios para la resolución de problemas:

1. `kubectl get pods --namespace <namespace>`
2. `kubectl describe pods <podname> --namespace <namespace>`
3. `journalctl --since "YYYY-MM-DD hh:mm:ss" --until "YYYY-MM-DD hh:mm:ss" > /tmp/<filename>`
4. `kubectl --namespace rcm logs --previous <pod name> --container <container name> > /tmp/<filename>`

Secuencia para solucionar problemas

1. Verifique si la vaina del centro de operaciones afectada está en un RCM MAESTRO o RCM DE RESPALDO ejecutando el comando en el par de alta disponibilidad:

<#root>

```
# rcm show-status
```

Example :

```
[unknown] rcm# rcm show-status
message :
{"status": "MASTER"}
```

2. Recopile la descripción de la vaina del centro de operaciones afectada y revise el conteo de reinicio y qué códigos de salida en los contenedores están en un estado problemático. Por ejemplo, los contenedores confd y confd notificaciones están actualmente en un estado problemático, como se indica:

<#root>

Example:

```
rcm # kubectl describe pods ops-center-rcm-ops-center --namespace rcm
Name:          ops-center-rcm-ops-center
Namespace:     rcm
...
Containers:
  confd:
    ...
```

Last State: Terminated

Reason: Error

Exit Code: 137

Started: Fri, 01 Dec 2023 12:44:13 +0530
Finished: Fri, 01 Dec 2023 12:46:09 +0530
Ready: False

Restart Count: 8097

```
...
confd-api-bridge:
...
```

State: Running
Started: Tue, 09 May 2023 02:36:37 +0530
Ready: True

```

Restart Count: 0
...
product-confd-callback:
...
  State:      Running
  Started:    Tue, 09 May 2023 02:36:38 +0530
  Ready:      True
  Restart Count: 0
...
confd-notifications:
...
  State:      Running
  Started:    Fri, 01 Dec 2023 12:46:14 +0530

Last State:      Terminated

Reason:      Error

Exit Code:      1

  Started:    Fri, 01 Dec 2023 12:40:50 +0530
  Finished:   Fri, 01 Dec 2023 12:46:00 +0530
  Ready:      True

Restart Count: 5278

```

...

3. Examine el código de salida para comprender la causa del reinicio inicial del contenedor.

Ejemplo:

El código de salida 137 indica que los contenedores/grupo de dispositivos no tienen memoria suficiente.

El código de salida 1 indica un cierre del contenedor debido a un error de la aplicación.

4. Revise el diario para verificar el cronograma del problema y comprenda desde cuándo se observa el problema. Los registros que indican el reinicio de las notificaciones confd del contenedor, como se muestra aquí, se pueden utilizar para identificar el inicio del tiempo de emisión:

<#root>

```

Nov 29 00:00:01 <nodename> kubelet[30789]: E1129 00:00:01.993620    30789 pod_workers.go:190] "Error syn
restarting failed container=confd-notifications

pod=ops-center-rcm-ops-center (<podUID>)\\"" pod="rcm/ops-center-rcm-ops-center" podUID=<podUID>

```

5. Revise los registros de contenedores de los contenedores reiniciados y verifique la causa del bucle de reinicio continuo del contenedor. En este ejemplo, los registros de contenedores indican un error al cargar la configuración de restauración:

<#root>

Example:

```
rcm # kubectl --namespace rcm logs --previous ops-center-rcm-ops-center --container confd
ConfD started
Failed to connect to server
All callpoints are registered - exiting
ConfD restore
Failure loading the restore configuration
ConfD load nodes config
DEBUG Failed to connect to ConfD: Connection refused
confd_load: 290: maapi_connect(sock, addr, addrlen) failed: system call failed (24): Failed to connect
...
Failure loading the nodes config
ConfD load day-N config
Failure loading the day-N config
...
Failure in starting confd - see previous errors - killing 1

rcm # kubectl --namespace rcm logs --previous ops-center-rcm-ops-center --container confd-notifications
...
Checking that ConfD is running.
Checking that ConfD is running.
ConfD is up and running
Failed to load schemas from confd
```



Advertencia:

Si los registros del contenedor se ejecutan con la opción `—previous` en un contenedor que no se ha reiniciado o terminado, devuelve un error:

```
rcm:~# kubectl --namespace rcm logs --previous ops-center-rcm-ops-center --container confd-api-br
Error from server (BadRequest): previous terminated container "confd-api-bridge" in pod "ops-cent
```

Posibles situaciones que provocan un problema con la restauración de la configuración posterior

No disponibilidad de configuración

- El contenedor `confd-api-bridge` tiene la función de leer la configuración de `confd` y crear una

copia de seguridad cada segundo. El comando confd-api-bridge lo almacena en el comando configmap ops-center-confd-<opscenter-name>.

- Si el contenedor confd se detiene y posteriormente, confd-api-bridge no recibe respuesta para la configuración, almacena una configuración vacía en el mapa de configuración.
- Cuando el contenedor confd intenta restaurar a partir de la configuración de copia de seguridad disponible, se produce un error y se produce el estado CrashLoopBackOff. Esto se puede verificar desde los registros del contenedor confd:

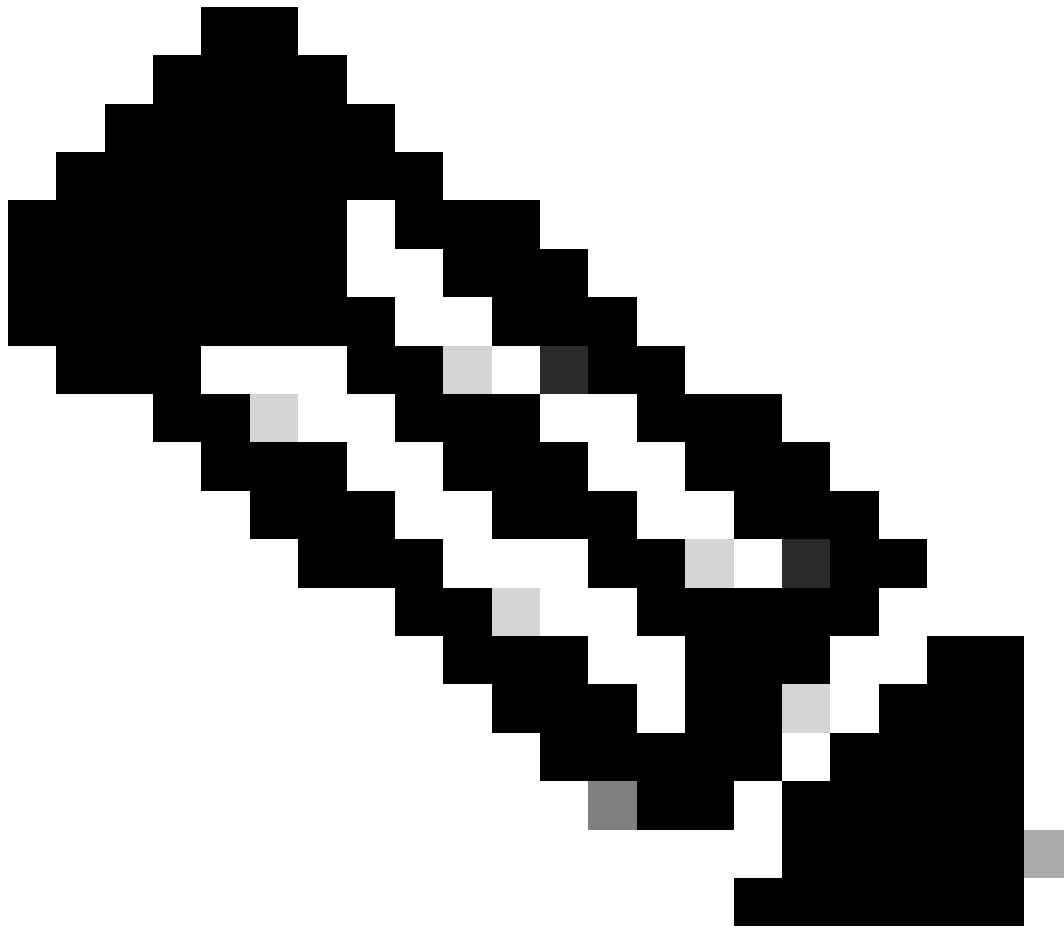
```
confd_load: 660: maapi_candidate_commit_persistent(sock, NULL) failed: notset (12): /cisco-mobile-produ
```

Este comportamiento se resuelve mediante el ID de bug de Cisco [CSCwi15801](#).

Restricciones del ciclo de CPU

- Cuando el contenedor confd intenta recuperarse, si el inicio no se completa en treinta segundos, el contenedor se reinicia.
- El inicio se retrasa si no recibe los ciclos de CPU requeridos debido a la alta carga de CPU en el RCM.
- Si la CPU del RCM continúa en un estado ocupado debido a la carga por otros grupos de dispositivos como rcm-checkpointmgr, el contenedor confd continúa reiniciándose y provoca el estado CrashLoopBackOff.

Este comportamiento se resuelve mediante el ID de bug de Cisco [CSCwe79529](#).



Nota:

- Si el RCM MAESTRO se ve afectado, efectúe un cambio del RCM al RCM DE APOYO y, a continuación, efectúe la localización de averías. Si no hay ningún RCM de APOYO disponible, continúe con la localización de averías del RCM MAESTRO.
 - Se recomienda consultar con Cisco TAC antes de realizar cualquier solución alternativa si se observa un grupo de dispositivos del centro de operaciones en el estado CrashLoopBackOff.
-

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).