

Solucionar problemas do Pod do centro de operações no estado CrashLoopBackOff

Contents

[Introdução](#)

[Acrônimos](#)

[Registros necessários](#)

[Sequência para solução de problemas](#)

[Possíveis cenários que levam a um problema com a restauração de configuração subsequente](#)

[Indisponibilidade da configuração](#)

[Restrições de Ciclo da CPU](#)

Introdução

Este documento descreve como identificar e recuperar o pod do centro de operações no estado CrashLoopBackOff.

Acrônimos

RCM - Gerenciador de Configuração de Redundância

AAAA-MM-DD hh:mm:ss - Ano-Mês-Dia Hora:Minuto:segundo

CPU - Unidade central de processamento

Registros necessários

Saídas do comando RCM necessárias para a solução 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>`

Sequência para solução de problemas

1. Verifique se o pod do centro de operações afetado está em um MASTER RCM ou BACKUP RCM executando o comando no par de alta disponibilidade:

<#root>

```
# rcm show-status
```

Example :

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

2. Colete a descrição do pod do centro operacional afetado e analise a contagem de reinicialização e quais códigos de saída nos contêineres estão em um estado problemático. Por exemplo, as notificações de contêineres confd e confd estão atualmente em um estado problemático, como indicado:

<#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 o código de saída para entender a causa da reinicialização inicial do contêiner.

Exemplo:

O código de saída 137 indica que os contêineres/pods não têm memória suficiente.

O código de saída 1 indica um desligamento de contêiner devido a um erro de aplicativo.

4. Revise o diário para verificar o cronograma do problema e entender de quando o problema é observado. Os registros que indicam a reinicialização das notificações confd do contêiner, como mostrado aqui, podem ser usados para identificar o início da hora de emissão:

<#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 os logs do contêiner de contêineres reiniciados e verifique a causa do loop contínuo de reinicialização do contêiner. Neste exemplo, os logs do contêiner indicam uma falha no carregamento da configuração de restauração:

<#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



aviso:

Se os logs de contêiner forem executados com a opção `—previous` em um contêiner que não tenha sido reiniciado ou encerrado, ele retornará um erro:

```
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
```

Possíveis cenários que levam a um problema com a restauração de configuração subsequente

Indisponibilidade da configuração

- O contêiner `confd-api-bridge` tem a função de ler a configuração do `confd` e criar um backup

a cada segundo. A ponte confd-api a armazena no mapa de configuração ops-center-confd-<opscenter-name>.

- Se o contêiner confd for interrompido e subsequentemente, a confd-api-bridge não recebe nenhuma resposta para a configuração, ela armazena uma configuração vazia no mapa de configuração.
- Quando o contêiner confd tenta restaurar a partir da configuração de backup disponível, ele falha e causa o estado CrashLoopBackOff. Isso pode ser verificado a partir dos registros de contêiner confd:

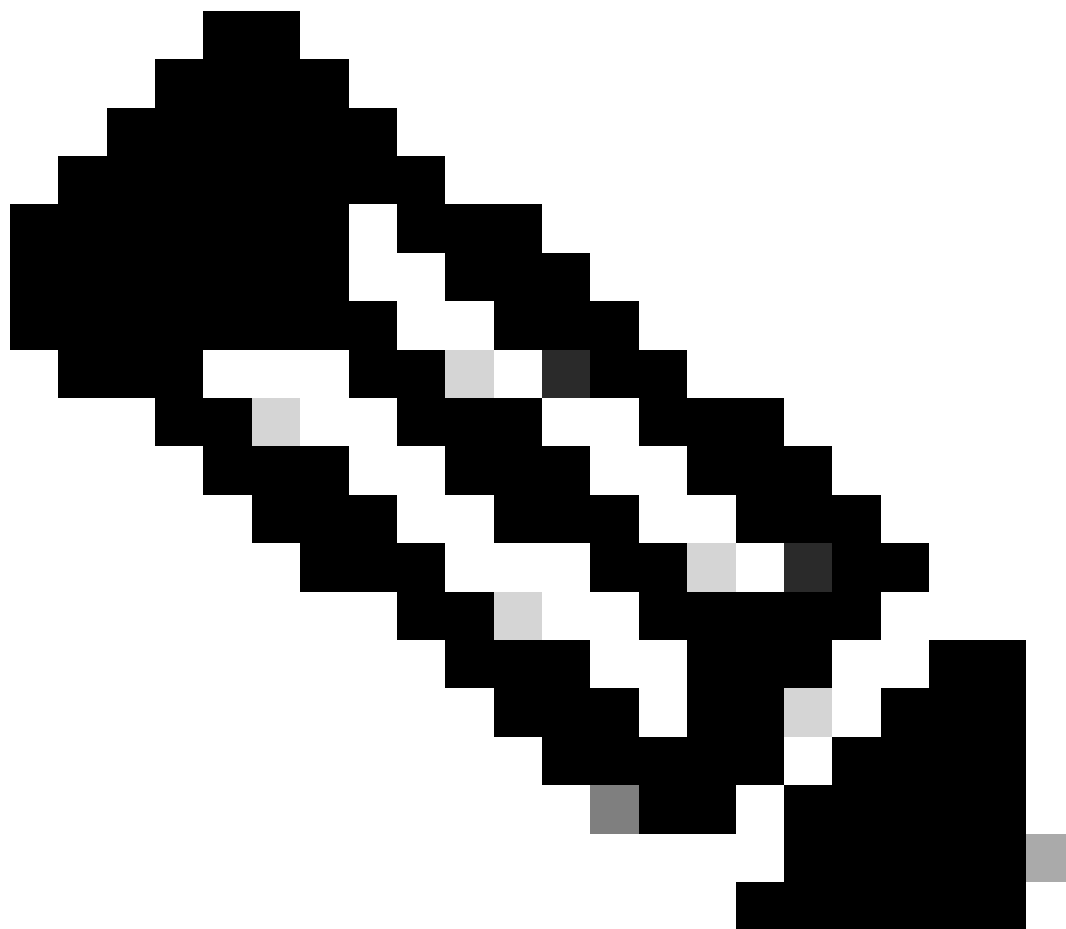
```
confd_load: 660: maapi_candidate_commit_persistent(sock, NULL) failed: notset (12): /cisco-mobile-produ
```

Esse comportamento é solucionado pelo bug da Cisco ID [CSCwi15801](#).

Restrições de Ciclo da CPU

- Quando o contêiner confd tenta se recuperar, se a inicialização não for concluída dentro de trinta segundos, o contêiner é reiniciado.
- A inicialização será atrasada se não receber os ciclos de CPU necessários devido à alta carga de CPU no RCM.
- Se a CPU do RCM continuar em um estado ocupado devido à carga por outros pods, como rcm-checkpointmgr, o contêiner confd continuará a reiniciar e causará o estado CrashLoopBackOff.

Esse comportamento é solucionado pelo bug da Cisco ID [CSCwe79529](#).



Note:

- Se o RCM PRINCIPAL for afetado, execute um switchover de RCM para o RCM DE BACKUP e faça troubleshooting adicional. E se nenhum RCM de BACKUP estiver disponível, continue a identificar e solucionar problemas do RCM MESTRE.
 - É recomendável consultar o Cisco TAC antes de executar qualquer solução alternativa se um pod do centro de operações for observado no estado CrashLoopBackOff.
-

Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.