

CrashLoopBackOff状態のOps-center Podのトラブルシューティング

内容

[はじめに](#)

[短縮形](#)

[必要なログ](#)

[トラブルシューティングの順序](#)

[その後の設定の復元で問題が発生する可能性のあるシナリオ](#)

[設定の利用不可](#)

[CPUサイクルの制約](#)

はじめに

このドキュメントでは、CrashLoopBackOff状態のops-centerポッドを特定して回復する方法について説明します。

短縮形

RCM：冗長構成マネージャ

YYYY-MM-DD hh:mm:ss – 年 – 月 – 日の時間：分：秒

CPU:Central Processing Unit (中央処理装置)

必要なログ

トラブルシューティングに必要なRCMコマンド出力：

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>`

トラブルシューティングの順序

1. ハイアベイラビリティペアでコマンドを実行して、影響を受けるオペレーションセンターポッドがマスターRCMまたはバックアップRCMにあるかどうかを確認します。

<#root>

rcm show-status

Example :

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

2. 影響を受けるオペレーションセンターのポッドのポッド説明を収集し、リスタート数とコンテナ内のどの終了コードに問題があるかを確認します。たとえば、コンテナconfdおよびconfd通知は現在、次のように問題のある状態になっています。

<#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. 終了コードを調べて、最初のコンテナ再起動の原因を理解します。

例：

終了コード137は、コンテナ/ポッドに十分なメモリがないことを示します。

終了コード1は、アプリケーションエラーによるコンテナのシャットダウンを示します。

4. journalctlを確認して問題のタイムラインを確認し、問題がいつ発生したかを理解します。次に示すように、コンテナのconfd通知の再開を示すログを使用して、問題発生時刻の開始を特定できます。

<#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. 再起動されたコンテナのコンテナログを確認し、コンテナの再起動ループが継続する原因を確認します。この例では、コンテナログは復元設定のロードの失敗を示しています。

<#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



警告：

再起動または終了していないコンテナで—previousオプションを指定してコンテナログを実行すると、次のエラーが返されます。

```
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
```

その後の設定の復元で問題が発生する可能性のあるシナリオ

設定の利用不可

- confd-api-bridgeコンテナには、confdから設定を読み込み、毎秒バックアップを作成する機能があります。confd-api-bridgeは、これをops-center-confd-<opscenter-name>コンフィグマップに保存します。

- confdコンテナが停止した後、confd-api-bridgeが設定に対する応答を受信しない場合、空の設定が設定マップに保存されます。
- confdコンテナが使用可能なバックアップ設定から復元しようとする、失敗し、CrashLoopBackOff状態が発生します。これは、confdコンテナログから確認できます。

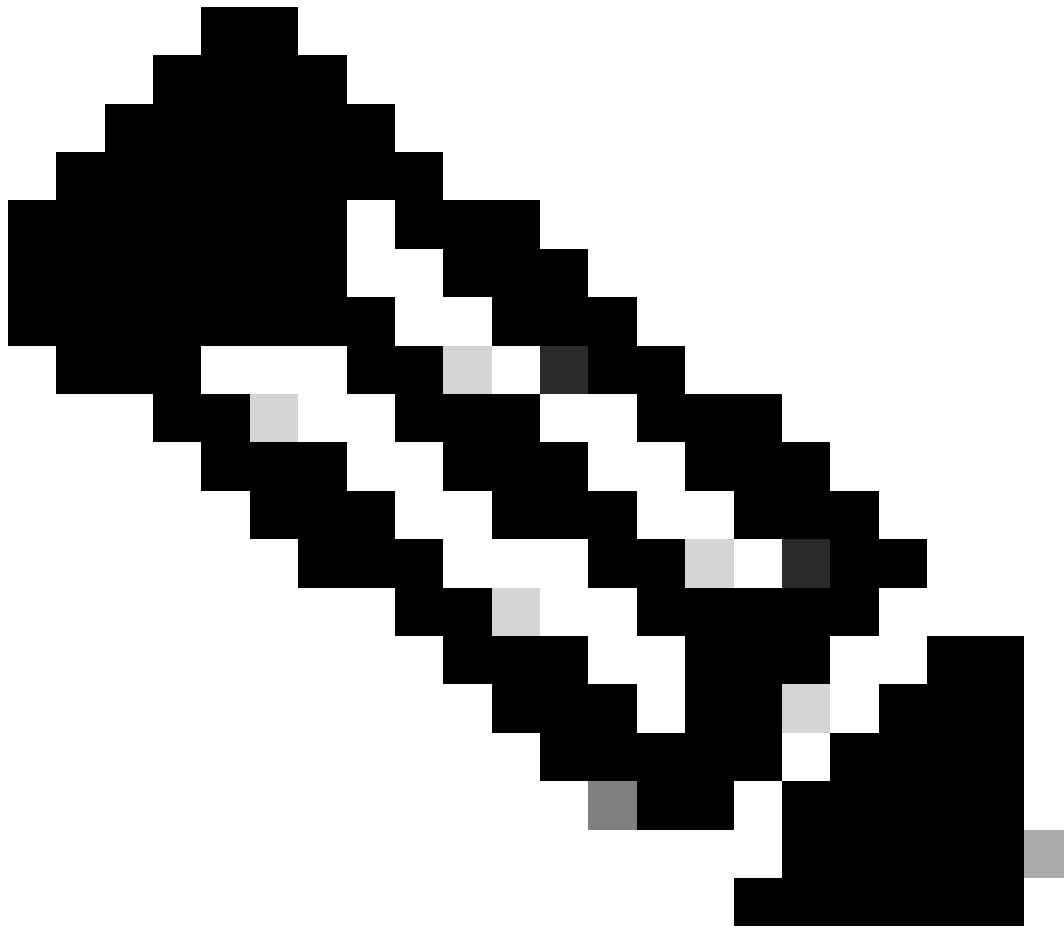
confd_load: 660: maapi_candidate_commit_persistent(sock, NULL) failed: notset (12): /cisco-mobile-produ

この動作は、Cisco Bug ID [CSCwi15801](#)で解決されています。

CPUサイクルの制約

- confdコンテナが復旧を試み、30秒以内に起動が完了しない場合は、コンテナが再起動されます。
- RCMのCPU負荷が高いために必要なCPUサイクルを受信しない場合、起動は遅延します。
- rcm-checkpointmgrなどの他のポッドによる負荷が原因でRCM CPUが引き続き占有された状態になっている場合、confdコンテナは再起動を続け、CrashLoopBackOff状態を引き起こします。

この動作は、Cisco Bug ID [CSCwe79529](#)で解決されています。



注：

- マスターRCMが影響を受ける場合は、バックアップRCMへのRCMスイッチオーバーを実行し、さらにトラブルシューティングを行います。また、使用可能なバックアップRCMがない場合は、マスターRCMのトラブルシューティングを続行します。
 - ops-centerポッドがCrashLoopBackOff状態であることが確認された場合は、回避策を実行する前にCisco TACに問い合わせることをお勧めします。
-

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。