

# Fehlerbehebung bei Operations Center Pod im CrashLoopBackOff-Zustand

## Inhalt

---

[Einleitung](#)

[Abkürzungen](#)

[Erforderliche Protokolle](#)

[Reihenfolge für die Fehlerbehebung](#)

[Mögliche Szenarien, die zu einem Problem mit der anschließenden Wiederherstellung der Konfiguration führen](#)

[Nicht verfügbare Konfiguration](#)

[Einschränkungen des CPU-Zyklus](#)

---

## Einleitung

In diesem Dokument wird beschrieben, wie der POD im CrashLoopBackOff-Status identifiziert und wiederhergestellt wird.

## Abkürzungen

RCM = Redundancy Configuration Manager

JJJJ-MM-TT hh:mm:ss - Jahr-Monat-Tageszeit:Minute:Sekunde

CPU - Zentraleinheit

## Erforderliche Protokolle

Für die Fehlerbehebung erforderliche RCM-Befehlsausgaben:

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>`

## Reihenfolge für die Fehlerbehebung

1. Überprüfen Sie, ob sich der betroffene Power-Center-POD in einem MASTER RCM oder BACKUP RCM befindet, indem Sie den Befehl im Hochverfügbarkeitspaar ausführen:

<#root>

```
# rcm show-status
```

**Example :**

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

2. Sammeln Sie die POD-Beschreibung des betroffenen Op-Center-PODs, und überprüfen Sie die Neustartzahl sowie die Ausstiegs-codes in den Containern, die sich in einem problematischen Zustand befinden. Beispielsweise befinden sich die Benachrichtigungen für die Container "confd" und "confd" derzeit in einem problematischen Zustand, wie hier dargestellt:

<#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. Untersuchen Sie den Exit-Code, um die Ursache des ersten Containerneustarts zu verstehen.

Beispiel:

Der Exit-Code 137 gibt an, dass die Container/Pod nicht über genügend Speicher verfügen.

Exitcode 1 gibt an, dass der Container aufgrund eines Anwendungsfehlers heruntergefahren wurde.

4. Überprüfen Sie die journalctl, um den Zeitrahmen für das Problem zu überprüfen und zu verstehen, ab wann das Problem festgestellt wurde. Protokolle, die den Neustart der CONFIG-Benachrichtigungen des Containers angeben, können verwendet werden, um den Beginn der Problemzeit zu identifizieren:

```
<#root>
```

```
Nov 29 00:00:01 <nodename> kubelet[30789]: E1129 00:00:01.993620 30789 pod_workers.go:190] "Error syncing pod ops-center-rcm-ops-center (<podUID>): container restarting failed container=confd-notifications pod=ops-center-rcm-ops-center (<podUID>)\\"" pod="rcm/ops-center-rcm-ops-center" podUID=<podUID>
```

5. Überprüfen Sie die Containerprotokolle neu gestarteter Container, und überprüfen Sie die Ursache für die Schleife für den fortlaufenden Containerneustart. In diesem Beispiel weisen die Containerprotokolle auf einen Fehler beim Laden der Wiederherstellungskonfiguration hin:

<#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
```



Warnung:

Wenn Containerprotokolle mit der Option `—` vor einem Container ausgeführt werden, der nicht neu gestartet oder beendet wurde, wird ein Fehler zurückgegeben:

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

---

## Mögliche Szenarien, die zu einem Problem mit der anschließenden Wiederherstellung der Konfiguration führen

### Nicht verfügbare Konfiguration

- Der config-api-bridge-Container hat die Funktion, die Konfiguration aus config auszulesen

und jede Sekunde ein Backup zu erstellen. Die config-api-bridge speichert diese in der configMap ops-center-config-<opscenter-name>.

- Wenn der CONFIG-Container gestoppt wird und anschließend keine Antwort für die Konfiguration an die CONFIG-API-Bridge gesendet wird, wird eine leere Konfiguration in der CONFIGUR-Map gespeichert.
- Wenn der CONFIG-Container versucht, eine Wiederherstellung aus der verfügbaren Sicherungskonfiguration durchzuführen, schlägt er fehl und verursacht den CrashLoopBackOff-Zustand. Dies kann aus den Protokollen des konfd-Containers überprüft werden:

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

Dieses Verhalten wird durch die Cisco Bug-ID [CSCwi15801 behoben](#).

## Einschränkungen des CPU-Zyklus

- Wenn der konfd-Container versucht, sich zu erholen, wird der Container neu gestartet, wenn der Start nicht innerhalb von dreißig Sekunden abgeschlossen ist.
- Der Start wird verzögert, wenn er aufgrund der hohen CPU-Last auf dem RCM nicht die erforderlichen CPU-Zyklen erhält.
- Wenn die RCM-CPU aufgrund der Last anderer PODs wie rcm-checkpointmgr in einem belegten Zustand weiterläuft, wird der CONFIG-Container weiterhin neu gestartet, und es wird der CrashLoopBackOff-Zustand ausgelöst.

Dieses Verhalten wird durch die Cisco Bug-ID [CSCwe79529 behoben](#).



Anmerkung:

- Wenn der MASTER-RCM betroffen ist, führen Sie einen RCM-Switchover zum BACKUP-RCM durch, und beheben Sie die Fehler anschließend. Wenn kein BACKUP-RCM verfügbar ist, fahren Sie mit der Fehlerbehebung beim MASTER-RCM fort.
  - Es wird empfohlen, sich vor der Durchführung von Workarounds mit dem Cisco TAC in Verbindung zu setzen, wenn ein PowerCenter-POD im CrashLoopBackOff-Zustand festgestellt wird.
-

## Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.