

CrashLoopBackOff 상태에서 Ops-center Pod 문제 해결

목차

[소개](#)

[약어](#)

[필요한 로그](#)

[문제 해결 순서](#)

[후속 컨피그레이션 복원 문제로 이어질 수 있는 시나리오](#)

[컨피그레이션 사용 불가](#)

[CPU 주기 제약](#)

소개

이 문서에서는 CrashLoopBackOff 상태에서 ops-center Pod를 식별하고 복구하는 방법에 대해 설명합니다.

약어

RCM - Redundancy Configuration Manager

YYYY-MM-DD hh:mm:ss - 년-월-일 시간:분:초

CPU - 중앙 처리 장치

필요한 로그

문제 해결에 필요한 RCM 명령 출력:

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>`

문제 해결 순서

1. 고가용성 쌍에서 다음 명령을 실행하여 영향을 받는 ops-center Pod가 MASTER RCM 또는 BACKUP RCM에 있는지 확인합니다.

<#root>

rcm show-status

Example :

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

2. 영향을 받는 운영 센터 포드의 포드 설명을 수집하고 재시작 카운트 및 컨테이너의 종료 코드가 문제가 있는 상태를 검토합니다. 예를 들어, 컨테이너의 confd 및 confd 알림은 다음과 같이 현재 문제가 있는 상태입니다.

<#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. 종료 코드를 검사하여 초기 컨테이너 재시작의 원인을 파악합니다.

예:

종료 코드 137은 컨테이너/Pod에 메모리가 충분하지 않음을 나타냅니다.

종료 코드 1은 애플리케이션 오류로 인해 컨테이너가 종료되었음을 나타냅니다.

4. journalctl을 검토하여 문제 일정을 확인하고 문제가 관찰된 시점을 이해합니다. 여기에 표시된 대로 컨테이너 컨피그레이트 알림의 재시작을 나타내는 로그를 사용하여 문제 발생 시간의 시작을 식별할 수 있습니다.

<#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. 재시작된 컨테이너의 컨테이너 로그를 검토하고 연속 컨테이너 재시작 루프의 원인을 확인합니다. 이 예에서 컨테이너 로그는 복원 구성을 로드하는 데 실패했음을 나타냅니다.

<#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



경고:

컨테이너 로그가 옵션과 함께 실행되면(이전에 재시작하거나 종료하지 않은 컨테이너에서 실행된 경우) 다음 오류가 반환됩니다.

```
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
```

후속 컨피그레이션 복원 문제로 이어질 수 있는 시나리오

컨피그레이션 사용 불가

- conpd-api-bridge 컨테이너는 conpd에서 컨피그레이션을 읽고 1초마다 백업을 생성하는 기능을 가지고 있다. conpd-api-bridge는 컨피그레이션 맵 ops-center-conpd-<opscenter-name>에 저장합니다.

- confd 컨테이너가 중지된 후 confd-api-bridge에서 컨피그레이션에 대한 응답을 수신하지 못하면 컨피그레이션에 빈 컨피그레이션이 저장됩니다.
- 컨피그레이션 컨테이너가 사용 가능한 백업 컨피그레이션에서 복원하려고 하면 실패하고 CrashLoopBackOff 상태가 됩니다. 이는 confd 컨테이너 로그에서 확인할 수 있습니다.

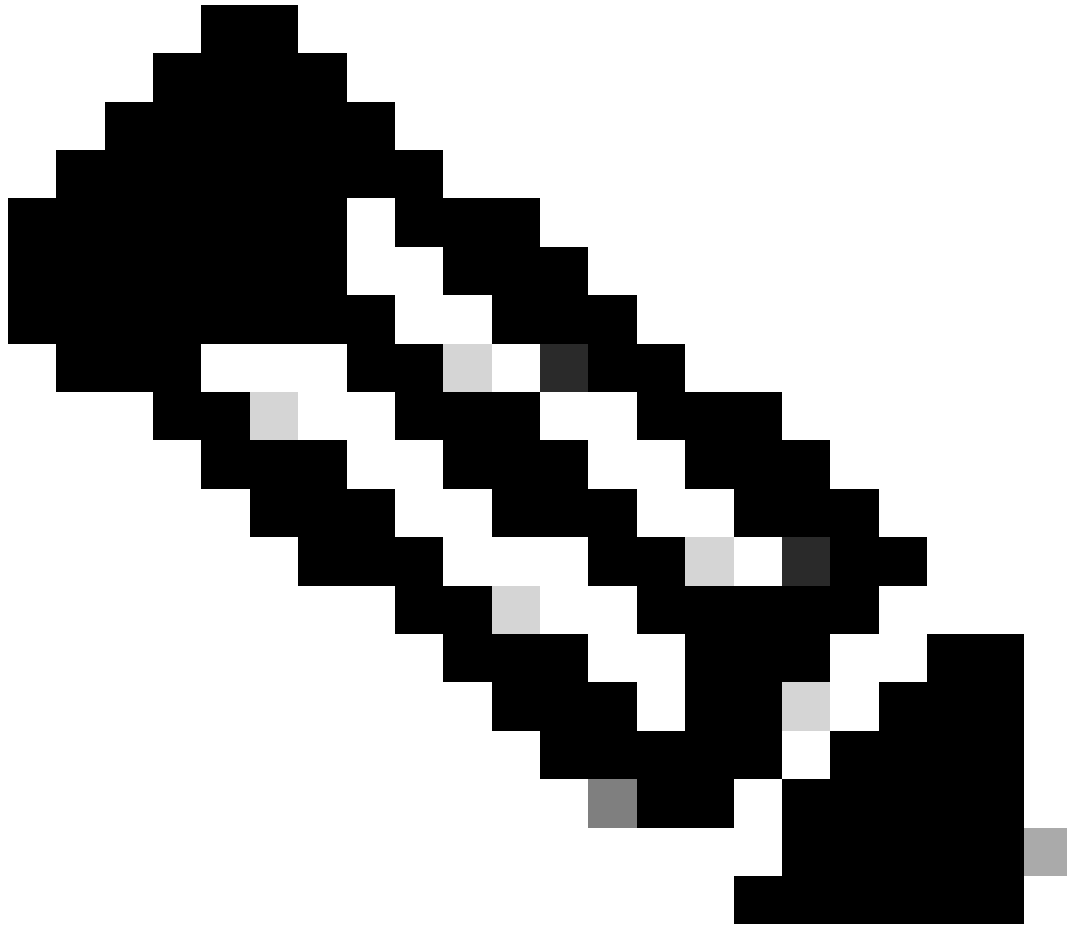
confd_load: 660: maapi_candidate_commit_persistent(sock, NULL) failed: notset (12): /cisco-mobile-produ

이 동작은 Cisco 버그 ID CSCwi15801 [로 해결됩니다](#).

CPU 주기 제약

- 구성된 컨테이너가 복구를 시도할 때 시작이 30초 내에 완료되지 않으면 컨테이너가 재시작됩니다.
- RCM의 높은 CPU 로드로 인해 필요한 CPU 사이클을 수신하지 못하면 시작이 지연됩니다.
- RCM-checkpointmgr과 같은 다른 포드의 부하로 인해 RCM CPU가 점유된 상태로 지속되면, confd 컨테이너가 계속 재시작되어 CrashLoopBackOff 상태가 됩니다.

이 동작은 Cisco 버그 ID CSCwe79529에 의해 [처리됩니다](#).



참고:

- MASTER RCM에 영향을 미치는 경우, BACKUP RCM으로 RCM 전환을 수행한 다음 추가 문제를 해결합니다. BACKUP RCM이 없으면 MASTER RCM의 문제를 계속 해결합니다.
 - CrashLoopBackOff 상태에서 ops-center Pod가 관찰된 경우 해결 방법을 수행하기 전에 Cisco TAC에 문의하는 것이 좋습니다.
-

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.