

Risoluzione dei problemi relativi a Ops-center Pod in stato CrashLoopBackOff

Sommario

[Introduzione](#)

[Acronimi](#)

[Registri necessari](#)

[Sequenza per la risoluzione dei problemi](#)

[Possibili scenari che causano un problema con il successivo ripristino della configurazione](#)

[Non disponibilità della configurazione](#)

[Vincoli del ciclo di CPU](#)

Introduzione

Questo documento descrive come identificare e ripristinare il pod ops-center nello stato CrashLoopBackOff.

Acronimi

RCM - Gestione configurazione ridondanza

AAAA-MM-GG hh:mm:ss - Anno-Mese-Giorno Ora:Minuto:secondo

CPU - Unità di elaborazione centrale

Registri necessari

Output dei comandi di RCM necessari per la risoluzione dei problemi:

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>`

Sequenza per la risoluzione dei problemi

1. Verificare se il pod ops-center interessato si trova in un RCM MASTER o BACKUP eseguendo il comando nella coppia di disponibilità elevata:

<#root>

```
# rcm show-status
```

Example :

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

2. Raccogliere la descrizione del pod del pod centro superiore interessato ed esaminare il conteggio riavvii e quali codici di uscita nei contenitori sono in uno stato problematico. Ad esempio, le notifiche confd e confd dei contenitori sono attualmente in uno stato problematico, come indicato:

<#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. Esaminare il codice di uscita per comprendere la causa del riavvio iniziale del contenitore.

Esempio:

Il codice di uscita 137 indica che i contenitori/pod non dispongono di memoria sufficiente.

Il codice di uscita 1 indica la chiusura del contenitore a causa di un errore dell'applicazione.

4. Esaminare il journalctl per verificare la cronologia del problema e capire da quando il problema è stato rilevato. I log che indicano il riavvio delle notifiche di conferma del contenitore, come illustrato di seguito, possono essere utilizzati per identificare l'inizio del tempo di emissione:

<#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. Esaminare i log dei container riavviati e verificare la causa del loop di riavvio continuo dei container. In questo esempio, i log dei container indicano un errore nel caricamento della configurazione di ripristino:

<#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
```



Avviso:

Se i registri del contenitore vengono eseguiti con l'opzione `—PREVIOUS` su un contenitore che non è stato riavviato o terminato, viene restituito un errore:

```
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
```

Possibili scenari che causano un problema con il successivo ripristino della configurazione

Non disponibilità della configurazione

- Il contenitore `config-api-bridge` è in grado di leggere la configurazione da `config` e creare un

backup ogni secondo. Il bridge config-api lo memorizza nella mappa di configurazione ops-center-config-<opscenter-name>.

- Se il contenitore config viene arrestato e in seguito, il bridge config-api non riceve alcuna risposta per la configurazione, archivia una configurazione vuota nella mappa di configurazione.
- Quando il contenitore di configurazione tenta di eseguire il ripristino dalla configurazione di backup disponibile, l'operazione non riesce e viene generato lo stato CrashLoopBackOff. È possibile verificare questa condizione dai registri del contenitore confd:

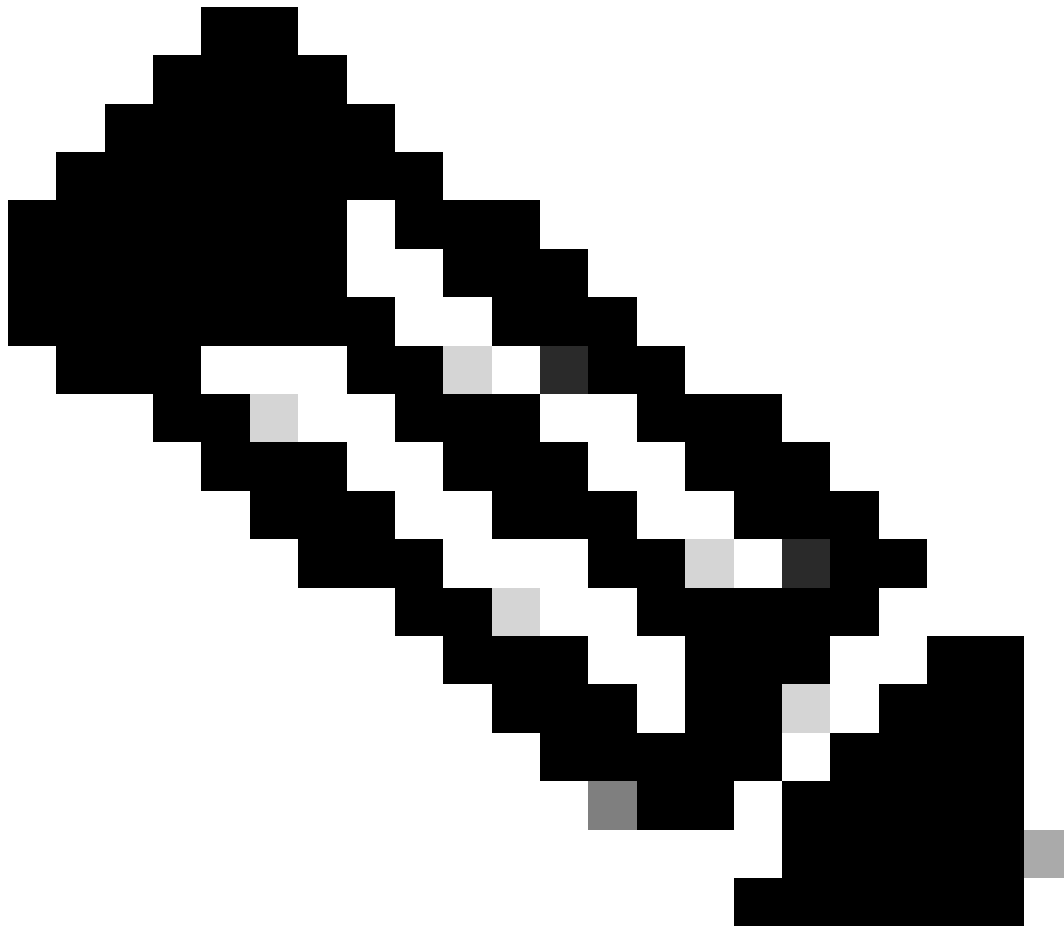
```
confd_load: 660: maapi_candidate_commit_persistent(sock, NULL) failed: notset (12): /cisco-mobile-produ
```

Questo comportamento è risolto da un bug Cisco con ID [CSCwi15801](#).

Vincoli del ciclo di CPU

- Quando il contenitore confd tenta di eseguire il ripristino, se l'avvio non viene completato entro trenta secondi, il contenitore viene riavviato.
- L'avvio viene ritardato se non riceve i cicli della CPU richiesti a causa dell'elevato carico della CPU su RCM.
- Se la CPU di RCM continua ad essere occupata a causa del carico da parte di altri supporti, ad esempio rcm-checkpointmgr, il contenitore confd continua a riavviarsi e a causare lo stato CrashLoopBackOff.

Per risolvere questo problema, consultare l'ID bug Cisco [CSCwe79529](#).



Nota:

- Se il problema riguarda RCM MASTER, eseguire il passaggio di RCM a RCM BACKUP, quindi procedere con la risoluzione del problema. Se invece non è disponibile alcun RCM di BACKUP, continuare con la risoluzione dei problemi di RCM MASTER.
 - Si consiglia di consultare Cisco TAC prima di adottare qualsiasi soluzione alternativa se si rileva un pod ops-center nello stato CrashLoopBackOff.
-

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).