

在站點隔離後解決Kubernetes Pods顯示為「未就緒」狀態

目錄

[簡介](#)

[問題](#)

[解決方案](#)

[初始Pod和系統驗證](#)

[重新啟動受影響的Pod](#)

[重新啟動後驗證Pod和系統狀態](#)

簡介

本檔案介紹由於Kubernetes錯誤<https://github.com/kubernetes/kubernetes/issues/82346>而使思科智慧安裝(SMI)Pod進入未就緒狀態的復原[步驟](#)。

問題

站點隔離後，融合乙太網(CEE)在CEE中報告了「Processing Error Alarm (處理錯誤警報)」。系統就緒狀態低於100%。

```
[sitelapp/pod1] cee# show alerts active
alerts active k8s-deployment-replica-mismatch f89d8d09389c
state active
severity critical
type "Processing Error Alarm"
startsAt 2021-05-27T08:38:58.703Z
source sitelapp-smi-cluster-policy-oam2
labels [ "component: kube-state-metrics" "deployment: prometheus-scrapeconfigs-synch"
"exported_namespace: cee-pod1" "instance: 192.0.2.37:8080" "job: kubernetes-pods" "namespace:
cee-pod1" "pod: kube-state-metrics-6c476f7494-tqkrc" "pod_template_hash: 6c476f7494" "release:
cee-pod1-cnat-monitoring" ]
annotations [ "summary: Deployment cee-pod1/prometheus-scrapeconfigs-synch has not matched the
expected number of replicas for longer than 2 minutes." ]
```

```
[sitelapp/pod1] cee# show system status
system status deployed true
system status percent-ready 92.68
```

```
ubuntu@sitelapp-smi-cluster-policy-mas01:~$ kubectl get rs -n cee-pod1 | grep scrape
NAME DESIRED CURRENT READY AGE
prometheus-scrapeconfigs-synch-ccd454f76 1 1 0 395d
prometheus-scrapeconfigs-synch-f5544b4f8 0 0 0 408d
```

解決方案

站點隔離是錯誤<https://github.com/kubernetes/kubernetes/issues/82346>的觸發器。使這些Pod處於Ready狀態的解決方法是重新啟動受影響的程式。修復程式包含在即將發佈的CEE版本中。

初始Pod和系統驗證

登入到CEE CLI並檢查系統狀態。

```
ssh -p 2024 admin@`kubect1 get svc -A | grep " ops-center-cee" | awk '{print $4}'`
```

```
show alerts active
```

```
show system status
```

重新啟動受影響的Pod

登入到主節點，在主節點上運行這些命令。並確定並非所有成員都處於「就緒」狀態的守護程式集和復制副本集。

```
kubect1 get daemonsets -A
```

```
kubect1 get rs -A | grep -v '0 0 0'
```

將這些命令複製並貼上到記事本中，並用站點上的cee名稱空間替換所有cee-xyz。

```
kubect1 describe pods core-retriever -n cee-xyz | egrep "^Name:|False" | grep -B1 False
```

```
kubect1 describe pods calico-node -n kube-system | egrep "^Name:|False" | grep -B1 False
```

```
kubect1 describe pods csi-cinder-nodeplugin -n kube-system | egrep "^Name:|False" | grep -B1 False
```

```
kubect1 describe pods maintainer -n kube-system | egrep "^Name:|False" | grep -B1 False
```

```
kubect1 describe pods kube-proxy -n kube-system | egrep "^Name:|False" | grep -B1 False
```

```
kubect1 describe pods path-provisioner -n cee-xyz | egrep "^Name:|False" | grep -B1 False
```

```
kubect1 describe pods logs-retriever -n cee-xyz | egrep "^Name:|False" | grep -B1 False
```

```
kubect1 describe pods node-exporter -n cee-xyz | egrep "^Name:|False" | grep -B1 False
```

```
kubect1 describe pods keepalived -n smi-vips | egrep "^Name:|False" | grep -B1 False
```

```
kubect1 describe pods prometheus-scrapeconfigs-synch -n cee-xyz | egrep "^Name:|False" | grep -B1 False
```

執行命令並收集結果輸出。在結果中，輸出標識了包含需要重新啟動的相應名稱空間的Pod名稱。

在您發出這些命令時，從以前獲取的清單中重新啟動所有受影響的pod（相應地替換pod名稱和名稱空間）。

```
kubect1 delete pods core-retriever-abcde -n cee-xyz
```

```
kubect1 delete pods core-retriever-abcde -n cee-xyz
```

```
...
```

確認電源盒已啟動並運行且沒有任何問題。

```
kubect1 get pods -A
```

重新啟動後驗證Pod和系統狀態

執行命令：

```
kubect1 get daemonsets -A
```

```
kubect1 get rs -A | grep -v '0 0 0'
```

確認守護程式和副本集顯示所有處於**Ready**狀態的成員。

登入到CEE CLI並確認沒有活動警報和系統狀態必須為100%。

```
ssh -p 2024 admin@`kubect1 get svc -A | grep " ops-center-cee" | awk '{print $4}'`
```

```
show alerts active
```

```
show system status
```