

サイト分離後にKubernetesポッドがNot Ready状態と表示される問題の解決

内容

[概要](#)

[問題](#)

[解決方法](#)

[初期ポッドとシステムの検証](#)

[該当するポッドの再起動](#)

[再起動後のポッドとシステムステータスの確認](#)

概要

このドキュメントでは、Cisco Smart Install(SMI)ポッドがKubernetesのバグ <https://github.com/kubernetes/kubernetes/issues/82346>が原因でNot Ready状態になる場合の回復手順について説明します。

問題

サイト分離後、コンバージドイーサネット(CEE)はCEEで処理エラーアラームを報告しました。システムの準備完了ステータスが100 %未満です。

```
[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
```

解決方法

サイトの分離は、バグ <https://github.com/kubernetes/kubernetes/issues/82346>のトリガーです。これらのポッドをReady状態にする回避策は、影響を受けるポッドを再起動することです。この修

正は、今後のCEEリリースに含まれています。

初期ポッドとシステムの検証

CEE CLIにログインし、システムステータスを確認します。

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

該当するポッドの再起動

プライマリノードにログインし、プライマリで次のコマンドを実行します。また、すべてのメンバーがReady状態でないデーモン・セットとレプリカ・セットを特定できます。

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

これらのコマンドをコピーしてメモ帳に貼り付け、すべてのcee-xyzを、サイト上のcee名前空間に置き換えます。

```
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
```

コマンドを実行し、結果の出力を収集します。その結果、出力は、再起動が必要な対応する名前空間を持つポッド名を特定します。

上記のコマンドを発行したときに取得したリストから、影響を受けるすべてのポッドを再起動します (ポッド名と名前空間を適宜置き換えます)。

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

ポッドが正常に動作していることを確認します。

```
kubect1 get pods -A
```

再起動後のポッドとシステムステータスの確認

実行コマンド：

```
kubectl get daemonsets -A
kubectl get rs -A | grep -v '0 0 0'
```

デーモン・セットとレプリカ・セットで、すべてのメンバーが受信可状態であることを確認します。

CEE CLIにログインし、アクティブなアラートとシステムステータスが100%になっていないことを確認します。

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

```
show alerts active
show system status
```