

# 사이트 격리 후 Kubernetes Pod Show in Not Ready State

## 목차

[소개](#)

[문제](#)

[솔루션](#)

[초기 Pod 및 시스템 확인](#)

[영향을 받는 Pod 재시작](#)

[재시작 후 Pod 및 시스템 상태 확인](#)

## 소개

이 문서에서는 Kubernetes 버그 <https://github.com/kubernetes/kubernetes/issues/82346>으로 인해 Cisco SMI(Smart Install) Pod가 준비 안 됨 상태로 전환될 때의 복구 단계를 [설명합니다](#).

## 문제

사이트 격리 후 CEE(Converged Ethernet)는 CEE에서 처리 오류 경보를 보고했습니다. 시스템 준비 상태가 100% 미만입니다.

```
[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
```

## 솔루션

사이트 격리는 버그 <https://github.com/kubernetes/kubernetes/issues/82346>에 대한 트리거입니다. 이러한 포드를 준비 상태로 유지하는 방법은 영향을 받는 포드를 다시 시작하는 것입니다. 이 수정 사항은 향후 CEE 릴리스에 포함되어 있습니다.

# 초기 Pod 및 시스템 확인

CEE CLI에 로그인하고 시스템 상태를 확인합니다.

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

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

## 영향을 받는 Pod 재시작

기본 노드에 로그인하고 기본 노드에서 다음 명령을 실행합니다. 준비 상태의 모든 구성원이 아닌 데몬 세트 및 복제본 세트를 식별합니다.

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

이 명령을 메모장에 복사하여 붙여넣고 모든 cee-xyz를 사이트의 cee 네임스페이스로 바꿉니다.

```
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
```

명령을 실행하고 결과 출력을 수집합니다. 그 결과 출력은 재시작이 필요한 해당 네임스페이스로 포드 이름을 식별합니다.

이러한 명령을 실행할 때 이전에 얻은 목록에서 영향을 받는 모든 Pod를 다시 시작합니다(포드 이름과 네임스페이스를 적절하게 대체).

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

Pod가 아무런 문제 없이 작동 및 실행 중인지 확인합니다.

```
kubect1 get pods -A
```

## 재시작 후 Pod 및 시스템 상태 확인

명령 실행:

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

데몬 세트 및 복제본 세트에 **준비** 상태의 모든 멤버가 표시되는지 확인합니다.

CEE CLI에 로그인하여 **활성 알림 및 시스템 상태가 100%가 아니어야 하는지** 확인합니다.

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

```
show alerts active
```

```
show system status
```