Risoluzione dei problemi relativi al riavvio continuo di Kube-Apiserver Pod

Sommario

Introduzione

Questo documento descrive una soluzione per ripristinare il riavvio continuo di kube-apiserver pod.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Docker e Kubernetes
- Cisco Subscriber Microservices Infrastructure (SMI) Ultra Cloud Core Common Execution Environment (CEE)

Componenti usati

Il riferimento delle informazioni contenute in questo documento è la versione 1.21.0 di Kubernetes.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Che cos'è kube-apiserver?

- Il server API Kubernetes Application Programming Interface (API) convalida e configura i dati per gli oggetti API che includono pod, servizi, controller di replica e altri. I servizi del server API eseguono operazioni di trasferimento dello stato di presentazione (REST) e forniscono il

front-end allo stato condiviso del cluster tramite il quale interagiscono tutti gli altri componenti.
- Il server API Kubernetes è responsabile dell'autenticazione e della convalida delle richieste, nonché del recupero e dell'aggiornamento dei dati nell'archivio dati etcd. Infatti, il server kube-API è l'unico componente che interagisce direttamente con l'archivio dati etcd.
- Di seguito sono riportati i passi che il server kube-API esegue quando viene creato un pod nel cluster:

r. Autentica utente

b. Convalida richiesta

c. Recupera dati

d. Aggiorna ETCD

e. Scheduler

f. Kubelet

- Gli altri componenti, quali l'utilità di pianificazione, kube-controller-manager e kubelet, utilizzano il server API per eseguire aggiornamenti nel cluster nelle rispettive aree.

Problema

Il riavvio kube-apiserver-smf-data-master-3 viene osservato continuamente. In questo caso, eseguire kubectl CLI **kubectl get pods -A -o wide | grep apiserver** per individuare il problema:

```
cloud-user@smf-data-master-1:~$ kubectl get pods -A -o wide | grep apiserver

kube-system      kube-apiserver-smf-data-master-1                  1/1     Running   4
68d    10.192.1.22       smf-data-master-1   <none>           <none>

kube-system      kube-apiserver-smf-data-master-2                  1/1     Running   4
68d    10.192.1.23       smf-data-master-2   <none>           <none>

kube-system      kube-apiserver-smf-data-master-3                  0/1     Running   2
68d    10.192.1.24       smf-data-master-3   <none>           <none>

cloud-user@smf-data-master-1:~$
```

Questi errori sono stati osservati nei **registri kubectl <nome_kube-apiserver_pod> -n kube-system:**

```
cloud-user@smf-data-master-1:~$ kubectl logs  kube-apiserver-smf-data-master-3 -n kube-system
E1116 20:09:52.635602       1 cacher.go:419] cacher (*core.Secret): unexpected ListAndWatch
error: failed to list *core.Secret: unable to transform key "/registry/secrets/cee-
dnceed21/alert-logger-sa-token-dzhkb": invalid padding on input; reinitializing...
E1116 20:09:53.691253       1 cacher.go:419] cacher (*core.Secret): unexpected ListAndWatch
error: failed to list *core.Secret: unable to transform key "/registry/secrets/cee-
dnceed21/alert-logger-sa-token-dzhkb": invalid padding on input; reinitializing...
E1116 20:09:54.751145       1 cacher.go:419] cacher (*core.Secret): unexpected ListAndWatch
error: failed to list *core.Secret: unable to transform key "/registry/secrets/cee-
dnceed21/alert-logger-sa-token-dzhkb": invalid padding on input; reinitializing...
E1116 20:09:55.808782       1 cacher.go:419] cacher (*core.Secret): unexpected ListAndWatch
error: failed to list *core.Secret: unable to transform key "/registry/secrets/cee-
dnceed21/alert-logger-sa-token-dzhkb": invalid padding on input; reinitializing...
```

```
E1116 20:09:56.865492       1 cacher.go:419] cacher (*core.Secret): unexpected ListAndWatch
error: failed to list *core.Secret: unable to transform key "/registry/secrets/cee-
dnceed21/alert-logger-sa-token-dzhkb": invalid padding on input; reinitializing...
E1116 20:09:57.906426       1 cacher.go:419] cacher (*core.Secret): unexpected ListAndWatch
error: failed to list *core.Secret: unable to transform key "/registry/secrets/cee-
dnceed21/alert-logger-sa-token-dzhkb": invalid padding on input; reinitializing...
E1116 20:09:58.963801       1 cacher.go:419] cacher (*core.Secret): unexpected ListAndWatch
error: failed to list *core.Secret: unable to transform key "/registry/secrets/cee-
dnceed21/alert-logger-sa-token-dzhkb": invalid padding on input; reinitializing...
E1116 20:10:00.027583       1 cacher.go:419] cacher (*core.Secret): unexpected ListAndWatch
error: failed to list *core.Secret: unable to transform key "/registry/secrets/cee-
dnceed21/alert-logger-sa-token-dzhkb": invalid padding on input; reinitializing...
E1116 20:10:01.084615       1 cacher.go:419] cacher (*core.Secret): unexpected ListAndWatch
error: failed to list *core.Secret: unable to transform key "/registry/secrets/cee-
dnceed21/alert-logger-sa-token-dzhkb": invalid padding on input; reinitializing...
E1116 20:10:02.206947       1 cacher.go:419] cacher (*core.Secret): unexpected ListAndWatch
error: failed to list *core.Secret: unable to transform key "/registry/secrets/cee-
dnceed21/alert-logger-sa-token-dzhkb": invalid padding on input; reinitializing...
E1116 20:10:03.256261       1 cacher.go:419] cacher (*core.Secret): unexpected ListAndWatch
error: failed to list *core.Secret: unable to transform key "/registry/secrets/cee-
dnceed21/alert-logger-sa-token-dzhkb": invalid padding on input; reinitializing...
E1116 20:10:04.313860       1 cacher.go:419] cacher (*core.Secret): unexpected ListAndWatch
error: failed to list *core.Secret: unable to transform key "/registry/secrets/cee-
dnceed21/alert-logger-sa-token-dzhkb": invalid padding on input; reinitializing...
E1116 20:10:05.363353       1 cacher.go:419] cacher (*core.Secret): unexpected ListAndWatch
error: failed to list *core.Secret: unable to transform key "/registry/secrets/cee-
dnceed21/alert-logger-sa-token-dzhkb": invalid padding on input; reinitializing...
```

Per eseguire il ripristino, è necessario provare a riavviare il kube-apiserver pod con CLI **kubectl delete pod <kube-apiserver_pod_name> -n kube-system,** ma questo non aiuta.

Analisi della causa principale

In un'ulteriore analisi, è stato rilevato che la differenza nel valore **segreto** tra master-3, dove il kube-apiserver è stato riavviato continuamente, e altri nodi master, ha causato questo problema.

```
From Master-1:

cloud-user@smf-data-master-1:~$ cat /data/kubernetes/secrets.conf

apiVersion: apiserver.config.k8s.io/v1

kind: EncryptionConfiguration

resources:

 - resources:

   - secrets

   providers:

   - aescbc:

       keys:

       - name: key1

         secret: BG5hleucjlD5ZDkFYUxoGLHHhBA/AeoNruHM0i70/ZI=   <<<<<<<<<<

   - identity: {}
```

```
cloud-user@smf-data-master-1:~$
```

From Master-3:

```
cloud-user@smf-data-master-3:~$ cat /data/kubernetes/secrets.conf
apiVersion: apiserver.config.k8s.io/v1
kind: EncryptionConfiguration
resources:
 - resources:
   - secrets
   providers:
   - aescbc:
       keys:
       - name: key1
         secret: XK+7mbh3YEnMdqswtySQ1d6QRehg+K6/J1d2e3EnMvI=   <<<<<<<<
   - identity: {}
cloud-user@smf-data-master-3:~$
```

Procedura di ripristino

1. Come parte del ripristino, copiare il segreto corrente di master-3 in un file di backup:

```
cloud-user@smf-data-master-3:~$ sudo cp /data/kubernetes/secrets.conf
/data/kubernetes/secrets.conf-bkp
```

2. Modificare i segreti, configurarli in Master-3 e modificare il valore di **secret** allo stesso valore visualizzato in altri nodi principali.

```
cloud-user@smf-data-master-3:~$ sudo vim /data/kubernetes/secrets.conf
apiVersion: apiserver.config.k8s.io/v1
kind: EncryptionConfiguration
resources:
 - resources:
   - secrets
```

```
   providers:

   - aescbc:

       keys:

       - name: key1

         secret: XK+7mbh3YEnMdqswtySQ1d6QRehg+K6/J1d2e3EnMvI=     <----  Change this value to
“BG5hleucjlD5ZDkFYUxoGLHHhBA/AeoNruHM0i70/ZI=“ as in other Master nodes

   - identity: {}
```

3. Riavviare il contenitore kube-apiserver su Master-3:

```
cloud-user@smf-data-master-3:~$ sudo docker ps -f "name=k8s_kube-apiserver" -q | xargs sudo
docker restart
```

Registra assegni

Verifica Kubernetes dal master:

```
cloud-user@pod-name-smf-master-1:~$ kubectl get pods -A -o wide | grep kube-apiserver
```

Ora, tutti i baccelli devono essere alzati e funzionare senza riavvii.