

Solucionar problemas de reinicialização contínua de pods do Kube-Apiserver

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[O que é o kube-apiserver?](#)

[Problema](#)

[Análise da causa raiz](#)

[Etapas para a recuperação](#)

[Pós-cheques](#)

Introduction

Este documento descreve uma solução para recuperar o reinício contínuo do pod do kube-apiserver.

Prerequisites

Requirements

A Cisco recomenda que você conheça estes tópicos:

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

Componentes Utilizados

As informações neste documento são baseadas na versão Kubernetes v1.21.0.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

O que é o kube-apiserver?

- O servidor da API (Application Programming Interface) do Kubernetes valida e configura dados para os objetos API, que incluem pods, serviços, controladores de replicação e outros. O Servidor API atende às operações REpresentational State Transfer (REST) e fornece o front-end para o estado compartilhado do cluster através do qual todos os outros

componentes interagem.

- O servidor API Kubernetes é responsável por autenticar e validar solicitações, recuperar e atualizar dados no armazenamento de dados etcd. Na verdade, o servidor kube-API é o único componente que interage diretamente com o armazenamento de dados etcd.
- Estas são as etapas que o servidor kube-API executa quando um pod é criado no cluster:

a. Autenticar usuário

b. Validar solicitação

c. Recuperar dados

d. Atualizar ETCD

e. Agendador

f. Kubelet

- Os outros componentes, como o agendador, kube-controller-manager e kubelet, usam o servidor de API para executar atualizações no cluster em suas respectivas áreas.

Problema

O reinício kube-apiserver-smf-data-master-3 é observado continuamente. Neste caso, execute kubectl CLI `kubectl get pods -A -o wide | grep apiserver` para identificar o 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:~$
```

Esses erros foram observados nos logs `kubectl <kube-apiserver_pod_name> -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/ce-
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/ce-
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/ce-
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/ce-
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/ce-
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/ce-
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/ce-
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/ce-
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/ce-
dnceed21/alert-logger-sa-token-dzhkb": invalid padding on input; reinitializing...
```

Para se recuperar, você deve tentar reiniciar o pod kube-apiserver com CLI `kubectl delete pod <kube-apiserver_pod_name> -n kube-system`, mas isso não ajuda.

Análise da causa raiz

Após análise mais detalhada, descobriu-se que a diferença no valor **secreto** entre o master-3, onde o kube-apiserver reiniciou continuamente, e outros nós mestres, causou essa questão.

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: BG5hleucj1D5ZDkFYUxoGLHHhBA/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:~$
```

Etapas para a recuperação

1. Como parte da recuperação, copie o segredo atual do master-3 em um arquivo de backup:

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

2. Edite os segredos, configure-os em Mestre-3 e altere o valor de **segredo** para o mesmo valor visto em outros nós mestres.

```
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
"BG5hleucj1D5ZDkFYUxoGLHHhBA/AeoNruHM0i70/ZI=" as in other Master nodes

- identity: {}

3. Reinicie o contêiner kube-apiserver no Master-3:

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

Pós-cheques

Verifique os Kubernetes do mestre:

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

Agora, todos os pods devem estar ativos e devem funcionar sem qualquer reinicialização.