Troubleshoot Path-Provisioner Memory Alert in PCF

Contents

Introduction

This document describes the procedure to troubleshoot the Path-Provisioner Memory Alert seen in the Policy Control Function (PCF).

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- PCF
- 5G Cloud Native Deployment Platform (CNDP)
- Dockers and Kubernetes

Components Used

The information in this document is based on these software and hardware versions:

- PCF REL_2023.01.2
- Kubernetes v1.24.6

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. If your network is live, ensure that you understand the potential impact of any command.

Background Information

In this setup, the CNDP hosts PCF.

A path provisioner, in the context of computer systems and infrastructure, typically refers to a component or tool that manages and provisions storage paths or volumes for applications or services.

A path provisioner is often associated with dynamic storage allocation and management in cloud environments or containerized setups. It allows applications or containers to request storage volumes or paths on-demand, without manual intervention or pre-allocation.

A path provisioner can handle tasks such as creating or mounting storage volumes, managing access permissions, and mapping them to specific application instances. It abstracts the underlying storage infrastructure, providing a simplified interface for applications to interact with storage resources.

Problem

Log in to the Common Execution Environment (CEE) Ops-center and verify that the on-path-provisioner pods report the Out of Memory (OOM) Alarms.

```
Command:
cee# show alerts active summary summary
Example:
[pcf01/pcfapp] cee# show alerts active summary
NAME UID SEVERITY STARTS AT DURATION SOURCE SUMMARY
-------------------------------------------------------------------------------------------------------
container-memory-usag 10659b0bcae0 critical 01-22T22:59:46 path-provisioner-pxps Pod cee-pcf/path-provi
container-memory-usag b2f10b3725e7 critical 01-22T15:51:36 path-provisioner-pxps Pod cee-pcf/path-provi
```

Analysis

Whenever you receive alarms for high memory usage on pods or containers of path-provisioner. Kubernetes(K8s) restart the pod when it reaches the max memory limit.
Alternatively, the pod can be manually restarted when it goes beyond the 80% threshold in order to avoid the high memory alerts.

Step 1. Check and verify the pod name reported in the active summary and output of this command.

```
Command:
cloud-user@pcf01-master-1$ kubectl get pods --all-namespaces | grep "path-provisioner"
Example:
cloud-user@pcf01-master-1:~$ kubectl get pods --all-namespaces | grep "path-provisioner"
NAMESPACE NAME READY STATUS RESTARTS AGE
cee-pcf path-provisioner-27bjx 1/1 Running 0 110d
cee-pcf path-provisioner-4mlq8 1/1 Running 0 110d
cee-pcf path-provisioner-4zvjd 1/1 Running 0 110d
cee-pcf path-provisioner-566pn 1/1 Running 0 110d
cee-pcf path-provisioner-6d2dr 1/1 Running 0 110d
cee-pcf path-provisioner-7g6l4 1/1 Running 0 110d
cee-pcf path-provisioner-8psnx 1/1 Running 0 110d
cee-pcf path-provisioner-94p9f 1/1 Running 0 110d
cee-pcf path-provisioner-bfr5w 1/1 Running 0 110d
cee-pcf path-provisioner-clpq6 1/1 Running 0 110d
cee-pcf path-provisioner-dbjft 1/1 Running 0 110d
cee-mpcf path-provisioner-dx9ts 1/1 Running 0 110d
cee-pcf path-provisioner-fx72h 1/1 Running 0 110d
cee-pcf path-provisioner-hbxgd 1/1 Running 0 110d
cee-pcf path-provisioner-k6fzc 1/1 Running 0 110d
```

```
cee-pcf path-provisioner-l4mzz 1/1 Running 0 110d
cee-pcf path-provisioner-ldxbb 1/1 Running 0 110d
cee-pcf path-provisioner-lf2xx 1/1 Running 0 110d
cee-pcf path-provisioner-lxrjx 1/1 Running 0 110d
cee-pcf path-provisioner-mjhlw 1/1 Running 0 110d
cee-pcf path-provisioner-pq65p 1/1 Running 0 110d
cee-pcf path-provisioner-pxpss 1/1 Running 0 110d
cee-pcf path-provisioner-q4b7m 1/1 Running 0 110d
cee-pcf path-provisioner-qlkjb 1/1 Running 0 110d
cee-pcf path-provisioner-s2jth 1/1 Running 0 110d
cee-pcf path-provisioner-vhzhg 1/1 Running 0 110d
cee-pcf path-provisioner-wqpmr 1/1 Running 0 110d
cee-pcf path-provisioner-xj5k4 1/1 Running 0 110d
cee-pcf path-provisioner-z4h98 1/1 Running 0 110d
cloud-user@pcf01-master-1:~$
```

Step 2. Verify the total count for the Active Path-Provisioner Pods.

<#root>

```
cloud-user@pcf01-master-1:~$

kubectl get pods --all-namespaces | grep "path-provisioner" | wc -l

29
cloud-user@pcf01-master-1:~$
```

Solution

Step 1. Execute the restart of the path-provisioner pods under the CEE names space login to the master node.

```
cloud-user@pcf01-master-1:~$ kubectl delete pod -n cee-pcf path-provisioner-pxpss
pod "path-provisioner-pxpss" deleted
```

Step 2. Verify the pods from Kubernetes are back online.

```
cloud-user@pcf01-master-1:~$ kubectl get pods --all-namespaces | grep "path-provisioner"
cee-pcf path-provisioner-27bjx 1/1 Running 0 110d
cee-pcf path-provisioner-4mlq8 1/1 Running 0 110d
cee-pcf path-provisioner-4zvjd 1/1 Running 0 110d
cee-pcf path-provisioner-566pn 1/1 Running 0 110d
cee-pcf path-provisioner-6d2dr 1/1 Running 0 110d
cee-pcf path-provisioner-7g6l4 1/1 Running 0 110d
cee-pcf path-provisioner-8psnx 1/1 Running 0 110d
cee-pcf path-provisioner-94p9f 1/1 Running 0 110d
cee-pcf path-provisioner-bfr5w 1/1 Running 0 110d
cee-pcf path-provisioner-clpq6 1/1 Running 0 110d
```

```
cee-pcf path-provisioner-dbjft 1/1 Running 0 110d
cee-pcf path-provisioner-dx9ts 1/1 Running 0 110d
cee-pcf path-provisioner-fx72h 1/1 Running 0 110d
cee-pcf path-provisioner-hbxgd 1/1 Running 0 110d
cee-pcf path-provisioner-k6fzc 1/1 Running 0 110d
cee-pcf path-provisioner-l4mzz 1/1 Running 0 110d
cee-pcf path-provisioner-ldxbb 1/1 Running 0 110d
cee-pcf path-provisioner-lf2xx 1/1 Running 0 110d
cee-pcf path-provisioner-lxrjx 1/1 Running 0 110d
cee-pcf path-provisioner-mjhlw 1/1 Running 0 110d
cee-pcf path-provisioner-pq65p 1/1 Running 0 110d
cee-pcf path-provisioner-pxpss 1/1 Running 0 7s
cee-pcf path-provisioner-q4b7m 1/1 Running 0 110d
cee-pcf path-provisioner-qlkjb 1/1 Running 0 110d
cee-pcf path-provisioner-s2jth 1/1 Running 0 110d
cee-pcf path-provisioner-vhzhg 1/1 Running 0 110d
cee-pcf path-provisioner-wqpmr 1/1 Running 0 110d
cee-pcf path-provisioner-xj5k4 1/1 Running 0 110d
cee-pcf path-provisioner-z4h98 1/1 Running 0 110d
cloud-user@pcf01-master-1:~$
```

Step 3. Verify that the total count for the Active Path-Provisioner Pods is the same as before the restart.

<#root>

```
cloud-user@pcf01-master-1:~$
```

```
kubectl get pods --all-namespaces | grep "path-provisioner" | wc -l
```

```
29
```

```
cloud-user@pcf01-master-1:~$
```

Step 4. Verify the active alerts and ensure that alerts related to the path-provisioner are cleared.

```
[pcf01/pcfapp] cee# show alerts active summary
NAME UID SEVERITY STARTS AT SOURCE SUMMARY
-------------------------------------------------------------------------------------------------------
watchdog 02d125c1ba48 minor 03-29T10:48:08 System This is an alert meant to ensure that the entire a...
```