

Fehlerbehebung bei POD-Problemen mit 5G SMI CEE

Inhalt

[Einleitung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Hintergrundinformationen](#)

[Was ist SMI?](#)

[Was ist SMI CEE?](#)

[Was sind PODs der CEE?](#)

[Was ist pgpool POD?](#)

[Was ist Postgres POD?](#)

[Problem](#)

[Fehlerbehebung](#)

[Problemumgehung](#)

[CEE herunterfahren](#)

[Löschen von Inhalten aus Ordnern](#)

[Wiederherstellen der CEE](#)

[Nachprüfungen](#)

[Überprüfen Sie, ob Warnmeldungen aus der CEE gelöscht werden.](#)

Einleitung

In diesem Dokument wird beschrieben, wie die Problemumgehung für die POD-Neustartprobleme (Subscriber Microservices Infrastructure (SMI) Common Execution Environment (CEE) Pool of Devices (PGPool) implementiert wird.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

- Cisco SMI CEE (Ultra Cloud Core CEE)
- 5G Cloud Native Deployment Platform (CNDP) oder SMI Bare Metal (BM)-Architektur
- Dockers und Kubertricks

Verwendete Komponenten

Die Informationen in diesem Dokument basierend auf folgenden Software- und Hardware-Versionen:

- SMI 2020.02.2.35
- Kubernetes v1.21.0

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

Hintergrundinformationen

Was ist SMI?

Cisco SMI ist ein mehrschichtiges Stack von Cloud-Technologien und -Standards, die auf Mikroservices basierende Anwendungen der Geschäftsbereiche Cisco Mobility, Cable und Broadband Network Gateway (BNG) ermöglichen - die alle über ähnliche Funktionen zur Teilnehmerverwaltung und ähnliche Datenspeicheranforderungen verfügen.

Die Attribute sind:

- Layer-Cloud-Stack (Technologien und Standards) zur Bereitstellung von Bereitstellungen von oben nach unten sowie zur Einbindung der aktuellen Cloud-Infrastruktur des Kunden
- Die CEE wird von allen Anwendungen für nicht anwendungsbezogene Funktionen (Datenspeicherung, Bereitstellung, Konfiguration, Telemetrie und Alarm) gemeinsam genutzt. Dadurch wird eine konsistente Interaktion und Erfahrung für alle Berührungspunkte und Integrationspunkte der Kunden ermöglicht.
- Anwendungen und die CEE werden in Mikroservice-Containern bereitgestellt und mit einem Intelligent Service Mesh verbunden.
- Bereitstellung, Konfiguration und Management über eine offene API zur Automatisierung

Was ist SMI CEE?

Die CEE ist eine Softwarelösung, die zur Überwachung von Mobil- und Kabelanwendungen entwickelt wurde, die auf dem SMI bereitgestellt werden. Die CEE erfasst Informationen (Schlüsselmetriken) aus den Anwendungen auf zentralisierte Weise für Techniker zum Debuggen und zur Fehlerbehebung.

Die CEE ist der gängige Satz von Tools, die für alle Anwendungen installiert werden. Es verfügt über ein dediziertes Ops Center, das die Benutzeroberfläche (CLI) und APIs zur Verwaltung der Überwachungstools bereitstellt. Für jeden Cluster ist nur ein CEE verfügbar.

Was sind PODs der CEE?

Ein POD ist ein Prozess, der auf Ihrem Kubernetes Cluster ausgeführt wird. Der POD kapselt eine granulare Einheit, die als Container bezeichnet wird. Ein POD enthält einen oder mehrere Container.

Kubernetes stellt einen oder mehrere PODs auf einem einzelnen Knoten bereit, der ein physisches oder virtuelles System sein kann. Jeder POD verfügt über eine eigene Identität mit einer internen IP-Adresse und einem Port-Bereich. Die Container innerhalb eines POD können jedoch die Storage- und Netzwerkressourcen gemeinsam nutzen. CEE verfügt über eine Reihe von PODs

mit einzigartigen Funktionen. Pgpool und Postgres gehören zu mehreren CEE PODs.

Was ist pgpool POD?

Pgpool verwaltet den Postgres-Ressourcenpool für Verbindung, Replikation, Lastenausgleich usw. Pgpool ist eine Middleware, die zwischen PostgreSQL-Servern und einer PostgreSQL-Datenbank arbeitet.

Was ist Postgres POD?

Postgres unterstützt die SQL-Datenbank (Structured Query Language) mit Redundanz zum Speichern von Alarmen und Grafana-Dashboards.

Problem

Die PGPool-PODs werden regelmäßig neu gestartet, während die PODs nach der Konferenz ohne Probleme ausgeführt werden.

Geben Sie den folgenden Befehl ein, um die Warnungen anzuzeigen:

```
show alerts active summary | include "POD_|k8s-pod-"
```

Hier wird eine Beispielwarnung der CEE angezeigt.

```
[pod-name-smf-data/podname] cee# show alerts active summary | include "POD_|k8s-pod-"  
k8s-pod-crashing-loop 1d9d2b113073 critical 12-15T21:47:39 pod-name-smf-data-mas  
Pod cee-podname/grafana-65cbdb9846-krqfj (grafana) is restarting 1.03 times / 5 minutes.  
POD_Restarted 04d42efb81de major 12-15T21:45:44 pgpool-67f48f6565-vjt Container=  
k8s_pgpool_pgpool-67f48f6565-vjttd_cee-podname_a9f68607-eac4-40a9-86ef-db8176e0a22a_1474 of pod=  
pgpool-... POD_Restarted f7657a0505c2 major 12-15T21:45:44 postgres-0 Container=  
k8s_postgres_postgres-0_cee-podname_59e0a768-6870-4550-8db3-32e2ab047ce2_1385 of pod= postgres-0  
in name... POD_Restarted 6e57ae945677 major 12-15T21:45:44 alert-logger-d96644d4 Container=  
k8s_alert-logger_alert-logger-d96644d4-dsc8h_cee-podname_2143c464-068a-418e-b5dd-  
ce1075b9360e_2421 of po... k8s-pod-crashing-loop 5b8e6a207aad critical 12-15T21:45:09 pod-name-  
smf-data-mas Pod  
cee-podname/pgpool-67f48f6565-vjttd (pgpool) is restarting 1.03 times / 5 minutes. POD_Down  
45a6b9bf73dc major 12-15T20:30:44 pgpool-67f48f6565-qbw Pod= pgpool-67f48f6565-qbw52 in  
namespace=  
cee-podname is DOWN for more than 15min POD_Down 4857f398a0ca major 12-15T16:40:44 pgpool-  
67f48f6565-vjt Pod= pgpool-67f48f6565-vjttd in namespace=  
cee-podname is DOWN for more than 15min k8s-pod-not-ready fc65254c2639 critical 12-11T21:07:29  
pgpool-67f48f6565-qbw Pod  
cee-podname/pgpool-67f48f6565-qbw52 has been in a non-ready state for longer than 1 minute. k8s-  
pod-not-ready 008b859e7333 critical 12-11T16:35:49 pgpool-67f48f6565-vjt Pod  
cee-podname/pgpool-67f48f6565-vjttd has been in a non-ready state for longer than 1 minute.
```

Fehlerbehebung

Geben Sie im Kuberettmaster folgenden Befehl ein:

```
kubectl describe pods -n
```

Die Beispielausgabe der POD-Beschreibung ist hier dargestellt. Die Ausgabe wird gekürzt.

Events:

Type Reason Age From Message

```
Normal Scheduled 14m default-scheduler Successfully assigned cee-pod-name-11/postgres-2
to pod-name-master-3
Normal Pulling 14m kubelet Pulling image "docker.10.192.x.x.nip.io/cee-2020.02.2.i38/
smi-libraries/postgresql/2020.02.2/postgres:1.3.0-946d87d"
Normal Pulled 13m kubelet Successfully pulled image "docker.10.192.x.x.nip.io/cee-2020.02.2.i38/
smi-libraries/postgresql/2020.02.2/postgres:1.3.0-946d87d" in 29.048094722s
Warning Unhealthy 12m kubelet Readiness probe failed: [bin][h][ir] >>> [2021-10-11 18:09:48]
pod is not ready
Warning Unhealthy 10m kubelet Readiness probe failed: [bin][h][ir] >>> [2021-10-11 18:11:18]
pod is not ready
Warning Unhealthy 10m kubelet Readiness probe failed: [bin][h][ir] >>> [2021-10-11 18:11:48]
pod is not ready
Warning Unhealthy 9m49s kubelet Readiness probe failed: [bin][h][ir] >>> [2021-10-11 18:12:18]
pod is not ready
Warning Unhealthy 9m19s kubelet Readiness probe failed: [bin][h][ir] >>> [2021-10-11 18:12:48]
pod is not ready
Warning Unhealthy 8m49s kubelet Readiness probe failed: [bin][h][ir] >>> [2021-10-11 18:13:18]
pod is not ready
Warning Unhealthy 8m19s kubelet Readiness probe failed: [bin][h][ir] >>> [2021-10-11 18:13:48]
pod is not ready
Warning Unhealthy 7m49s kubelet Readiness probe failed: [bin][h][ir] >>> [2021-10-11 18:14:18]
pod is not ready
Warning Unhealthy 7m19s kubelet Readiness probe failed: [bin][h][ir] >>> [2021-10-11 18:14:48]
pod is not ready
Warning BackOff 6m44s kubelet Back-off restarting failed container
```

ODER

Events:

Type Reason Age From Message

```
Warning FailedScheduling 13m default-scheduler 0/5 nodes are available: 2 node(s)
didn't match Pod's node affinity/selector, 3 node(s) didn't find available persistent
volumes to bind.
Normal Scheduled 13m default-scheduler Successfully assigned cee-pod-name-11/postgres-0
to pod-name-master-1
Warning FailedScheduling 13m default-scheduler 0/5 nodes are available: 2 node(s)
didn't match Pod's node affinity/selector, 3 node(s) didn't find available
persistent volumes to bind.
Normal Pulling 13m kubelet Pulling image "docker.10.192.x.x.nip.io/cee-2020.02.2.i38/
smi-libraries/postgresql/2020.02.2/postgres:1.3.0-946d87d"
Normal Pulled 12m kubelet Successfully pulled image "docker.10.192.x.x.nip.io/
cee-2020.02.2.i38/smi-libraries/postgresql/2020.02.2/postgres:1.3.0-946d87d"
in 43.011763302s
Warning Unhealthy 7m20s kubelet Liveness probe failed: [bin][h][imm] >>>
[2021-10-11 18:09:16] My name is pg-postgres-0
```

Problemumgehung

Anmerkung: Dieses Verfahren verursacht keine Ausfallzeiten in der Anwendung.

CEE herunterfahren

Geben Sie folgende Befehle aus der CEE ein, um die CEE herunterzufahren:

```
[pod-name-smf-data/podname] cee#  
[pod-name-smf-data/podname] cee# config terminal  
Entering configuration mode terminal  
[pod-name-smf-data/podname] cee(config)# system mode shutdown  
[pod-name-smf-data/podname] cee(config)# commit  
Commit complete.
```

Warten Sie, bis das System auf 100 % wechselt.

Löschen von Inhalten aus Ordnern

Von Master-vip über SSH zu jeder Master-VMs und entfernen Sie den Inhalt dieser Ordner: **/data/cee-podname/data-postgres-[0-2]**.

```
Master 1  
cloud-user@pod-name-smf-data-master-1:~$ sudo rm -rf /data/cee-podname/data-postgres-0  
Master 2  
cloud-user@pod-name-smf-data-master-2:~$ sudo rm -rf /data/cee-podname/data-postgres-1  
Master 3  
cloud-user@pod-name-smf-data-master-3:~$ sudo rm -rf /data/cee-podname/data-postgres-2
```

Wiederherstellen der CEE

Um die CEE wiederherzustellen, geben Sie die folgenden Befehle aus der CEE ein:

```
[pod-name-smf-data/podname] cee#  
[pod-name-smf-data/podname] cee# config terminal  
Entering configuration mode terminal  
[pod-name-smf-data/podname] cee(config)# system mode running  
[pod-name-smf-data/podname] cee(config)# commit  
Commit complete.
```

Warten Sie, bis das System auf 100 % steigt.

Nachprüfungen

Überprüfen Sie Kuberettes vom Master.

```
cloud-user@pod-name-smf-data-master-1:~$ kubectl get pods -A -o wide | egrep 'postgres|pgpool'  
All pods should display up and running without any restarts
```

Überprüfen Sie, ob Warnmeldungen aus der CEE gelöscht werden.

Geben Sie den folgenden Befehl ein, um zu überprüfen, ob Warnungen aus der CEE gelöscht werden:

```
show alerts active summary | include "POD_|k8s-pod-"
```

Sie können diesen Befehl auch eingeben, um sicherzustellen, dass ein Master- und zwei Standby-DBs vorhanden sind:

```

echo "0-----";kubectl
exec -it postgres-0 -n $(kubectl get pods -A | grep postgres | awk '{print $1}' | head -1)
-- /usr/local/bin/cluster/healthcheck/is_major_master.sh;echo "1-----
-----";kubectl exec -it postgres-1 -n $(kubectl get pods -A | grep postgres | awk '{print
$1}'
| head -1) -- /usr/local/bin/cluster/healthcheck/is_major_master.sh;echo "2-----
-----"; kubectl exec -it postgres-2 -n $(kubectl get pods -A | grep postgres |
awk '{print $1}' | head -1) -- /usr/local/bin/cluster/healthcheck/is_major_master.sh;

```

Die erwartete Beispielausgabe lautet:

```

cloud-user@pod-name-smf-data-master-1:~$ echo "0-----";kubectl
exec -it postgres-0 -n $(kubectl get pods -A | grep postgres | awk '{print $1}' | head -1)
-- /usr/local/bin/cluster/healthcheck/is_major_master.sh;echo "1-----
-----";kubectl exec -it postgres-1 -n $(kubectl get pods -A | grep postgres | awk '{print
$1}'
| head -1) -- /usr/local/bin/cluster/healthcheck/is_major_master.sh;echo "2-----
-----"; kubectl exec -it postgres-2 -n $(kubectl get pods -A | grep postgres |
awk '{print $1}' | head -1) -- /usr/local/bin/cluster/healthcheck/is_major_master.sh;
0-----
[bin][h][imm] >>> [2021-12-15 22:05:18] My name is pg-postgres-0
[bin][h][imm] >>> My state is good.
[bin][h][imm] >>> I'm not a master, nothing else to do!
1-----
[bin][h][imm] >>> [2021-12-15 22:05:19] My name is pg-postgres-1
[bin][h][imm] >>> My state is good.
[bin][h][imm] >>> I think I'm master. Will ask my neighbors if they agree.
[bin][h][imm] >>> Will ask nodes from PARTNER_NODES list
[bin][h][imm] >>> Checking node pg-postgres-0
[bin][h][imm] >>>>>>> Count of references to potential master pg-postgres-1 is 1 now
[bin][h][imm] >>> Checking node pg-postgres-1
[bin][h][imm] >>> Checking node pg-postgres-2
[bin][h][imm] >>>>>>> Count of references to potential master pg-postgres-1 is 2 now
[bin][h][imm] >>> Potential masters got references:
[bin][h][imm] >>>>> Node: pg-postgres-1, references: 2
[bin][h][imm] >>> I have 2/2 incoming reference[s]!
[bin][h][imm] >>>> 2 - Does anyone have more?
[bin][h][imm] >>> Yahoo! I'm real master...so I think!
2-----
[bin][h][imm] >>> [2021-12-15 22:05:21] My name is pg-postgres-2
[bin][h][imm] >>> My state is good.
[bin][h][imm] >>> I'm not a master, nothing else to do!

```