CNDP에서 Kubernetes 포드 충돌 트러블슈팅

목차

[소개](#)
[사전 요구 사항](#)
[요구 사항](#)
[사용되는 구성 요소](#)
[배경 정보](#)
[문제](#)
[분석](#)
[실행 계획](#)

소개

이 문서에서는 CNDP(Cloud Native Deployment Platform)에서 포드 충돌 문제를 해결하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

이 문서에 대한 특정 요건이 없습니다.

사용되는 구성 요소

이 문서는 특정 소프트웨어 및 하드웨어 버전으로 한정되지 않습니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경 정보

이 설정에서는 CNDP(Cloud Native Deployment Platform)가 SMF(Session Management Function)를 호스팅합니다.

문제

CEE(Common Execution Environment)에서 Pod Crash에 대한 경고를 볼 수 있습니다.

Command:

```
cee# show alerts active summary summary
```

Example:

```
[smf-rcdn/cee-rcdn] cee# show alerts active summary summary
NAME UID SUMMARY
-------------------------------------------------------------------------------------------
k8s-pod-crashing-loop bd4394046466 Pod smf-rcdn/smf-service-n0-6 (smf-service) is...
k8s-pod-crashing-loop 0ac1019911e3 Pod smf-rcdn/smf-service-n0-14 (smf-service) i...
k8s-pod-crashing-loop eeff8fa16660 Pod smf-rcdn/smf-service-n0-9 (smf-service) is...
k8s-pod-crashing-loop 470ff66822dc Pod smf-rcdn/smf-service-n0-5 (smf-service) is...
k8s-pod-crashing-loop cc8950f07ace Pod smf-rcdn/smf-service-n0-15 (smf-service) i...
k8s-pod-crashing-loop 05a7d1e291a6 Pod smf-rcdn/smf-service-n0-3 (smf-service) is...
```

분석

마스터 노드에 연결하고 crash한 모든 kubernetes 포드를 표시합니다. Grep에 대한 CrashLoopBackOff. 동일한 출력에서 이 Pod가 다시 시작된 횟수를 볼 수 있습니다.

Command:

```
master$ kubectl get pods -n
```

```
cloud-user@smf-rcdn-master-1:~$ kubectl get pods -n smf-rcdn |grep -v Running
NAME                                                  READY   STATUS             RESTARTS
AGE
smf-service-n0-10                                     1/2     CrashLoopBackOff   1224
6d7h
smf-service-n0-11                                     1/2     CrashLoopBackOff   1242
6d7h
smf-service-n0-15                                     1/2     CrashLoopBackOff   1244
6d7h
smf-service-n0-2                                      1/2     CrashLoopBackOff   1241
6d7h
smf-service-n0-3                                      1/2     CrashLoopBackOff   1251
6d7h
smf-service-n0-5                                      1/2     CrashLoopBackOff   1231
6d7h
smf-service-n0-7                                      1/2     CrashLoopBackOff   1249
6d7h
```

추락한 포드에 대해 설명해주세요. 이렇게 하면 Pod가 충돌한 이유에 대한 자세한 정보를 얻을 수 있습니다. Events(이벤트) 아래의 로그를 **확인합니다.**

Command:

```
master$ kubectl describe pod -n
```

```
:
cloud-user@smf-rcdn-master-1:~$ kubectl describe pod -n smf-rcdn smf-service-n0-11 |grep -i start
Start Time: Tue, 09 Aug 2022 03:13:54 +0000 Started: Tue, 09 Aug 2022 03:13:56 +0000 Restart
Count: 0 Started: Mon, 15 Aug 2022 11:33:10 +0000 Started: Mon, 15 Aug 2022 11:26:55 +0000
Restart Count: 1263 Started: Tue, 09 Aug 2022 03:13:58 +0000 Restart Count: 0  Events:  Type
Reason Age From Message ---- ------ ---- ---- ------- Warning BackOff 65s (x15210 over 3d6h)
kubelet Back-off restarting failed container
```

예를 들어, Pod를 smf-service-n1-0 NGFW에 연결해야 합니다 smf-rcdn-service-ims2 를 클릭하면 코어 파일을 수집할 수 있습니다.

```
ubuntu@smf-rcdn-master1:~$ kubectl get pods -n smf-ims -o wide | grep smf-service-n1-0
NAME                 READY   STATUS    RESTARTS   AGE   IP           NODE
NOMINDATEDN NODE   READINESS GATES
smf-service-n1-0     2/2     Running   10         9h    10.20.9.142 smf-rcdn-service-ims2
```

Connect to the Node(노드에 연결)는 충돌이 발생하여 이진 파일을 수집하는 호스트 Pod입니다. 이 파일은 Cisco의 분석에 필요합니다.

Command:

```
master1:~$ kubectl cp
```

Example:

```
ubuntu@smf-rcdn-master1:~$ kubectl cp smf-ims/smf-service-n1-0:/opt/workspace/smf-service
/tmp/smf-service
```

Connect to the Node(노드에 연결)는 충돌이 발생한 호스트 Pod이며 folder/var/lib/systemd**/coredump/로** 이동하여 내용을 표시합니다. 생성된 경우 이 폴더에서 볼 수 있습니다.

Example:

```
ubuntu@smf-rcdn-master1:~$ ssh smf-rcdn-service-ims2
ubuntu@smf-rcdn-service-ims2:~$ cd /var/lib/systemd/coredump/
ubuntu@smf-rcdn-service-ims2:/var/lib/systemd/coredump$ ls -ltr
total 982340
-rw-r----- 1 root root 52968460 Sep 21 16:40 core.smf-
service.0.a829fbabe2e649a7ab02150838fe47ae.1232.1599842408000000.lz4
-rw-r----- 1 root root 61609776 Sep 21 16:41 core.smf-
service.0.a829fbabe2e649a7ab02150838fe47ae.3468.1599842463000000.lz4
-rw-r----- 1 root root 74233259 Sep 21 16:46 core.smf-
service.0.a829fbabe2e649a7ab02150838fe47ae.28259.1599842775000000.lz4
-rw-r----- 1 root root 58241763 Sep 21 16:52 core.smf-
service.0.a829fbabe2e649a7ab02150838fe47ae.17155.1599843174000000.lz4
-rw-r----- 1 root root 43732684 Sep 21 16:56 core.smf-
service.0.a829fbabe2e649a7ab02150838fe47ae.3076.1599843385000000.lz4
-rw-r----- 1 root root 52377930 Sep 21 17:06 core.smf-
service.0.a829fbabe2e649a7ab02150838fe47ae.8024.1599844002000000.lz4
-rw-r----- 1 root root 63990106 Sep 21 17:07 core.smf-
service.0.a829fbabe2e649a7ab02150838fe47ae.26962.1599844074000000.lz4
-rw-r----- 1 root root 98058261 Sep 21 17:15 core.smf-
service.0.a829fbabe2e649a7ab02150838fe47ae.13026.1599844546000000.lz4
-rw-r----- 1 root root 59586871 Sep 21 17:24 core.smf-
service.0.a829fbabe2e649a7ab02150838fe47ae.21720.1599845052000000.lz4
-rw-r----- 1 root root 71187759 Sep 21 17:50 core.smf-
service.0.a829fbabe2e649a7ab02150838fe47ae.19705.1599846648000000.lz4
-rw-r----- 1 root root 96949278 Sep 21 17:57 core.smf-
service.0.a829fbabe2e649a7ab02150838fe47ae.11744.1599847049000000.lz4
-rw-r----- 1 root root 6052439 Sep 21 17:57 core.smf-
service.0.a829fbabe2e649a7ab02150838fe47ae.23846.1599847052000000.lz4
-rw-r----- 1 root root 70642243 Sep 21 17:58 core.smf-
service.0.a829fbabe2e649a7ab02150838fe47ae.18327.1599847110000000.lz4
-rw-r----- 1 root root 66052273 Sep 21 18:10 core.smf-
```

```
service.0.a829fbabe2e649a7ab02150838fe47ae.1504.1599847843000000.lz4
-rw-r----- 1 root root 65132876 Sep 21 18:10 core.smf-
service.0.a829fbabe2e649a7ab02150838fe47ae.12528.1599847855000000.lz4
-rw-r----- 1 root root 65000665 Sep 21 18:32 core.smf-
service.0.a829fbabe2e649a7ab02150838fe47ae.9462.1599849167000000.lz4
ubuntu@smf-rcdn-master1:~$:/var/lib/systemd/coredump$
```

폴더 내의 모든 파일을 tar합니다.

ubuntu@smf-rcdn-service-ims2:~$ **sudo tar czvfsmf-rcdn-service-ims2.tar.gz *.lz4**

마스터 SFTP에서 코어가 있는 노드로 이동한 다음 이를 Master /tmp 폴더에 다운로드하고 PC로 가져옵니다.

ubuntu@smf-rcdn-master1:~$: **sftp smf-rcdn-service-ims2**

명령은 마지막 Pod 재시작 전에 로그를 인쇄하고 충돌 서명을 캡처합니다.

Command:

```
master:~$ kubectl logs -n
```

Example:

```
ubuntu@smf-rcdn-master1:~$ kubectl logs -n smf-ims -p smf-service-n1-0 -c smf-service
/usr/local/go/src/runtime/asm_amd64.s:1357 (0x462d01) panic: runtime error: invalid memory
address or nil pointer dereference [signal SIGSEGV: segmentation violation code=0x1 addr=0x50
pc=0x13d92f6] goroutine 839296 [running]: panic(0x196c320, 0x3441300)
/usr/local/go/src/runtime/panic.go:722 +0x2c2 fp=0xc000a9d050 sp=0xc000a9cfc0 pc=0x432d82
runtime.panicmem(...) /usr/local/go/src/runtime/panic.go:199 runtime.sigpanic()
/usr/local/go/src/runtime/signal_unix.go:394 +0x3ec fp=0xc000a9d080 sp=0xc000a9d050 pc=0x4487cc
smf-service/userplane.(*UpfServData).

     ProcessSessionModificationResponse(0xc0059fe660, 0xc005b98f00, 0xc00aa6e3c0,
0x2001181ae72b892, 0xc00ea43570, 0x3, 0x4,



0xc005cd0820, 0xc005b11410, 0xc005b10d20, ...) /opt/workspace/smf-service/src/smf-
service/userplane/upfSessionModification.go:743 +0x526 fp=0xc000a9d408 sp=0xc000a9d080
pc=0x13d92f6 smf-
service/procedures/4g/pdn5g4gHo.(*Pdn5g4gHoProcedure).awtUpfModifyProcN4ModifyResp(0xc005a17440,
0xc0099e36c0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0) /opt/workspace/smf-service/src/smf-
service/procedures/4g/pdn5g4gHo/mbrUtils.go:485 +0x24d fp=0xc000a9d630 sp=0xc000a9d408
pc=0x1562d0d smf-
service/procedures/4g/pdn5g4gHo.(*Pdn5g4gHoProcedure).handleUpfModifyEvents(0xc005a17440,
0xc0099e36c0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0) /opt/workspace/smf-service/src/smf-
service/procedures/4g/pdn5g4gHo/stateHandler.go:196 +0x4a1 fp=0xc000a9d768 sp=0xc000a9d630
pc=0x1570d31 smf-service/procedures/4g/pdn5g4gHo.(*Pdn5g4gHoProcedure).HandleEvent(0xc005a17440,
0xc0099e36c0, 0x6, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, ...) /opt/workspace/smf-service/src/smf-
service/procedures/4g/pdn5g4gHo/procedure.go:364 +0x707 fp=0xc000a9d8d0 sp=0xc000a9d768
pc=0x1567887 smf-service/vendor/wwwin-github.cisco.com/mobile-cnat-smf/smf-common.git/src/smf-
common/callflow.(*BaseProcedure).Handle(0xc00568b4a0, 0xc0099e36c0, 0x0,
0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0) /opt/workspace/smf-service/src/smf-service/vendor/wwwin-
github.cisco.com/mobile-cnat-smf/smf-common.git/src/smf-common/callflow/BaseProcedure.go:54
+0xdb
```

```
fp=0xc000a9d978 sp=0xc000a9d8d0 pc=0xf5996b smf-service/vendor/wwwin-github.cisco.com/mobile-
cnat-smf/smf-common.git/src/smf-common/callflow.(*SessionState).ProcessContinue(0xc00b79b6d0,
0xc0099e36c0,
0xc00568b4a0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, ...) /opt/workspace/smf-service/src/smf-
service/vendor/wwwin-github.cisco.com/mobile-cnat-smf/smf-common.git/src/smf-
common/callflow/SessionState.go:169 +0x1f2
fp=0xc000a9da20 sp=0xc000a9d978 pc=0xf5d552 smf-
service/processor.(*SmfAppMessageProcessor).ProcessContinue(0x3a31da0, 0xc005b98f00, 0x1d34988,
0x35, 0x9, 0x1d34988, 0x35) /opt/workspace/smf-service/src/smf-
service/processor/grpc_message_processor.go:430 +0x4ab fp=0xc000a9dc20 sp=0xc000a9da20
pc=0x174fc0b smf-service/vendor/wwwin-github.cisco.com/mobile-cnat-golang-lib/app-
infra.git/src/app-infra/infra.(*masterBlueprint).processTransaction
(0xc0003141e0, 0xc005b98f00, 0xc000a9dd98) /opt/workspace/smf-service/src/smf-
service/vendor/wwwin-github.cisco.com/mobile-cnat-golang-lib/app-infra.git/src/app-
infra/infra/MasterBlueprint.go:301
+0x1a7 fp=0xc000a9dce8 sp=0xc000a9dc20 pc=0xd39ca7 smf-service/vendor/wwwin-
github.cisco.com/mobile-cnat-golang-lib/app-infra.git/src/app-infra/infra.(*masterBlueprint).
processTransactionWithCR(0xc0003141e0, 0xc005b98f00, 0x1cfeb00) /opt/workspace/smf-
service/src/smf-service/vendor/wwwin-github.cisco.com/mobile-cnat-golang-lib/app-
infra.git/src/app-infra/infra/MasterBlueprint.go:234
+0x394 fp=0xc000a9de78 sp=0xc000a9dce8 pc=0xd396e4 smf-service/vendor/wwwin-
github.cisco.com/mobile-cnat-golang-lib/app-infra.git/src/app-infra/infra.(*masterBlueprint).
processSessionTransaction(0xc0003141e0, 0xc005b98f00, 0x1, 0x0) /opt/workspace/smf-
service/src/smf-service/vendor/wwwin-github.cisco.com/mobile-cnat-golang-lib/app-
infra.git/src/app-infra/infra/MasterBlueprint.go:177
+0x124 fp=0xc000a9ded0 sp=0xc000a9de78 pc=0xd39104 smf-service/vendor/wwwin-
github.cisco.com/mobile-cnat-golang-lib/app-infra.git/src/app-infra/infra.(*masterBlueprint).
processEvent(0xc0003141e0, 0xc005b98f00, 0x1d02487) /opt/workspace/smf-service/src/smf-
service/vendor/wwwin-github.cisco.com/mobile-cnat-golang-lib/app-infra.git/src/app-
infra/infra/MasterBlueprint.go:138 +0x5fc
fp=0xc000a9df88 sp=0xc000a9ded0 pc=0xd3869c smf-service/vendor/wwwin-github.cisco.com/mobile-
cnat-golang-lib/app-infra.git/src/app-infra/infra.(*ApplicationContext).NewTransaction.func2
(0xc0006af400, 0xc005b98f00) /opt/workspace/smf-service/src/smf-service/vendor/wwwin-
github.cisco.com/mobile-cnat-golang-lib/app-infra.git/src/app-
infra/infra/ApplicationContext.go:1268
+0x7c fp=0xc000a9dfd0 sp=0xc000a9df88 pc=0xd9b69c runtime.goexit()
/usr/local/go/src/runtime/asm_amd64.s:1357 +0x1 fp=0xc000a9dfd8 sp=0xc000a9dfd0 pc=0x462d01
created by smf-service/vendor/wwwin-github.cisco.com/mobile-cnat-golang-lib/app-
infra.git/src/app-infra/infra.(*ApplicationContext).NewTransaction /opt/workspace/smf-
service/src/smf-service/vendor/wwwin-github.cisco.com/mobile-cnat-golang-lib/app-
infra.git/src/app-infra/infra/ApplicationContext.go:1266 +0x62c goroutine 1 [sleep]:
runtime.gopark(0x1dbaa10, 0x34ef580, 0xc001f01313, 0x2) /usr/local/go/src/runtime/proc.go:304
+0xe0 fp=0xc000a3bca8 sp=0xc000a3bc88 pc=0x434ea0 runtime.goparkunlock(...)
```

CEE에 연결하고 Pod 충돌이 발생하기 전후에 tac-debug를 수집합니다.

`tac-debug-pkg create from yyyy-mm-dd_hh:mm:ss to yyyy-mm-dd_hh:mm:ss` tac-debug-pkg create from yyyy-mm-dd_hh:mm:ss to yyyy-mm-dd_hh:mm:ss

실행 계획

이 충돌의 근본 원인을 찾기 위해 Cisco TAC에 대한 서비스 요청을 엽니다.

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.