

5G SMI CNDP 클러스터의 하드웨어 문제 해결 및 유지 관리 수행

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[SMI란 무엇입니까?](#)

[SMI-BM 또는 CNDP란 무엇입니까?](#)

[SMI Cluster Manager란 무엇입니까?](#)

[SMI Cluster Deployer란 무엇입니까?](#)

[문제](#)

[유지 관리 절차](#)

소개

이 문서에서는 5G SMI(Subscriber Microservices Infrastructure) CNDP(Cloud Native Deployment Platform) POD(Pool of Devices)에서 펌웨어(FW) 업그레이드 등의 유지 관리(하드웨어 교체 또는 유지 보수)를 수행하는 절차에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- Cisco SMI
- 5G CNDPA 또는 SMI-Bare-metal(BM) 아키텍처
- 도커 및 쿠버네티스
- Cisco UCS C220 Series 서버

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- SMI 2020.02.2.35
- Kubernetes v1.21.0
- Cisco UCS C220-M5SX-CM

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

잠재적인 영향을 이해해야 합니다.

배경 정보

SMI란 무엇입니까?

Cisco SMI는 Cisco Mobility, Cable 및 BNG 비즈니스 유닛에서 마이크로서비스 기반 애플리케이션을 지원하는 클라우드 기술 및 표준의 계층적 스택으로, 모두 유사한 가입자 관리 기능과 유사한 데이터 저장소 요구 사항이 있습니다.

특성:

- 레이어 클라우드 스택(기술 및 표준)을 통해 위에서 아래로 구축하며 현재 고객 클라우드 인프라를 수용합니다.
- 모든 애플리케이션에서 비애플리케이션 기능(데이터 스토리지, 구축, 구성, 텔레메트리, 경보)을 위해 공유하는 공통 실행 환경 따라서 모든 고객 접점과 통합 지점에 일관된 상호 작용 및 환경이 제공됩니다.
- 애플리케이션 및 공통 실행 환경은 마이크로서비스 컨테이너에 구축되고 Intelligent Service Mesh와 연결됩니다.
- 구축, 구성 및 관리를 위한 노출된 API로 자동화를 지원합니다.

SMI-BM 또는 CNDP란 무엇입니까?

Cisco SMI-Bare Metal 또는 CNDP는 VNF(Virtual Network Functions) 및 CNF(Cloud-Native Functions)를 구축하기 위한 인프라를 제공하고 Cisco Mobility, Cable 및 BNG 비즈니스 유닛을 지원하는 조정된 베어 메탈 플랫폼입니다.

특성:

- VIM 관련 오버헤드를 제거하는 베어 메탈
- 성능 향상 애플리케이션을 위한 더 많은 코어애플리케이션 실행 속도 향상
- 자동화된 배포 워크플로, NSO CFP와 통합
- Cisco 5G NF를 구축하기 위한 커리티드 스택
- 간소화된 주문 및 구축 가이드

SMI Cluster Manager란 무엇입니까?

클러스터 관리자는 제어 평면 및 사용자 평면 클러스터 구축 모두의 초기 지점으로 사용되는 2개의 노드 보존 클러스터입니다. 단일 노드 Kerberos 클러스터 및 전체 클러스터 설정을 담당하는 POD 집합을 실행합니다. 기본 클러스터 관리자만 활성 상태이며, 장애 발생 시에만 보조 서버가 인계되거나 유지 관리를 위해 수동으로 중단된 경우에만 보조 서버가 적용됩니다.

SMI Cluster Deployer란 무엇입니까?

SMI Deployer는 VM을 생성하고, 호스트 OS를 사용자 지정하고, K8s 클러스터를 만들고, K8s Master를 실행하고, 클러스터를 구성하고, Apps를 실행할 수 있는 Cluster Manager의 서비스입니다.

문제

하드웨어 장애 또는 소프트웨어/fw 업그레이드 등의 하드웨어 유지 보수를 위해서는 서버의 다운타임이 필요합니다. POD에서 유지 보수를 수행하려면 어떤 절차를 따라야 합니까? 애플리케이션에서 원치 않는 다운타임을 방지하기 위해 서비스를 원활하게 중지하는 방법

유지 관리 절차

유지 관리를 수행할 클러스터 관리자 VIP, Kubernetes 마스터 VIP(각 애플리케이션용), UCS CIMC IP, UCS CIMC 이름 및 서버 호스트 이름(OS 호스트 이름)을 가져옵니다.

Kubernetes 마스터에 로그인하여 서비스에 해당하고 모든 POD가 **실행** 조건에 있는지 확인합니다.

샘플 출력:

```
cloud-user@pod-name-smf-data-master-1:~$ kubectl get pods -A | grep -v Running
NAMESPACE NAME READY STATUS RESTARTS AGE
```

2. 클러스터 관리자에 로그인하여 SMI 클러스터 구축자 운영 센터에 액세스합니다(운영 센터 IP를 찾는 절차).

```
kubectl get svc -n $(kubectl get ns | grep -i smi-cm | awk '{print $1}') | grep ^ops-center
(Here "smi-cm" is the namespace in which cluster deployer is hosted and the "ops-center" is the starting name of the cluster deployer service name which is "ops-center-smi-cluster-deployer" these names can vary based on the environment setup)
```

샘플 출력:

```
cloud-user@tp-tam-deployer-cm-primary:~$ kubectl get svc -n $(kubectl get ns | grep smi-cm | awk '{print $1}') | grep ^ops-center
ops-center-smi-cluster-deployer ClusterIP 10.100.x.x <none>
8008/TCP,2024/TCP,2022/TCP,7681/TCP,3000/TCP,3001/TCP 154d
```

3. 이 명령으로 로그인합니다.

```
ssh -p 2024 admin@10.100.x.x
```

(2024 is the port used to connect to cluster deployer)

4. show clusters 명령을 사용하여 서비스가 애플리케이션에 해당하는지 확인합니다.

샘플 출력:

```
Welcome to the Cisco SMI Cluster Deployer on tp-tam-deployer-cm-primary
Copyright © 2016-2020, Cisco Systems, Inc.
All rights reserved.
```

```
admin connected from 192.x.x.x using ssh on ops-center-smi-cluster-deployer-5cdc5f94db-bnxqt
[tp-tam-deployer-cm-primary] SMI Cluster Deployer# show clusters
LOCK TO
NAME VERSION
-----
pod-name-smf-data -
pod-name-smf-ims -
```

```
pod1-name-smf-data -
pod1-name-smf-ims -
pod2-name-aio-1 -
pod2-name-aio-2 -
pod2-name-upf-data -
pod2-name-upf-ims -
```

5. 이러한 명령으로 유지 관리를 수행하는 노드를 제거하고 **Yes**를 입력합니다. 그러면 POD가 정상적으로 분리되고 필요에 따라 다른 노드에서 다시 시작됩니다.

샘플 출력:

```
[cluster-name-cm-1] SMI Cluster Deployer# clusters cluster-name nodes worker-11 actions sync
drain remove-node true
```

This will run drain on the node, disrupting pods running on the node. Are you sure? [no,yes] yes
message accepted

6. 이러한 명령을 사용하여 노드를 유지 관리 모드로 이동합니다(최대 30분 정도 걸릴 수 있음).

샘플 출력:

```
[cluster-name-cm-1] SMI Cluster Deployer# config
Entering configuration mode terminal
[cluster-name-cm-1] SMI Cluster Deployer(config)# clusters cluster-name
[cluster-name-cm-1] SMI Cluster Deployer(config-clusters-cluster-name)# nodes worker-11
[cluster-name-cm-1] SMI Cluster Deployer(config-nodes-worker1)# maintenance true
[cluster-name-cm-1] SMI Cluster Deployer(config-nodes-worker1)# commit
Commit complete.
[cluster-name-cm-1] SMI Cluster Deployer(config-nodes-worker1)# end
```

7. 로그에 상태를 확인합니다.

clusters cluster-name nodes worker-11 actions sync logs

(In this we are dealing with the worker-11 node)

샘플 출력(잘림):

```
logs 2022-01-03 06:04:02.755 DEBUG cluster_sync.cluster-name.worker-11: Cluster name: cluster-
name
```

```
2022-01-03 06:04:02.755 DEBUG cluster_sync.cluster-name.worker-11: Node name: worker-11
```

```
2022-01-03 06:04:02.755 DEBUG cluster_sync.cluster-name.worker-11: debug: false
```

```
2022-01-03 06:04:02.755 DEBUG cluster_sync.cluster-name.worker-11: remove_node: false
```

```
PLAY [Check required variables] *****
```

```
TASK [Gathering Facts] *****
```

```
Monday 03 January 2022 06:04:06 +0000 (0:00:00.014) 0:00:00.014 *****
```

ok: [worker-11]

ok: [worker-13]

ok: [worker-11]

ok: [worker-16]
ok: [worker-18]
ok: [worker-17]
ok: [worker-12]
ok: [worker-10]
ok: [worker-19]
ok: [worker-2]
ok: [master-1]
ok: [worker-11]
ok: [worker-15]
ok: [master-3]
ok: [worker-20]
ok: [worker-22]
ok: [worker-21]
....

TASK [Check node_name] *****

Monday 03 January 2022 06:04:13 +0000 (0:00:07.086) 0:00:07.101 *****

skipping: [master-1]
skipping: [master-2]
skipping: [master-3]
skipping: [worker-1]
skipping: [worker-10]
skipping: [worker-11]
skipping: [worker-12]
skipping: [worker-13]
skipping: [worker-11]
skipping: [worker-15]
skipping: [worker-16]
skipping: [worker-17]
skipping: [worker-18]
skipping: [worker-19]
skipping: [worker-2]

skipping: [worker-20]
skipping: [worker-21]
skipping: [worker-22]

.....

PLAY [Wait for ready and ensure uncordoned] *****

TASK [Cordon and drain node] *****

Monday 03 January 2022 06:04:15 +0000 (0:00:01.116) 0:00:08.217 *****

skipping: [master-1]
skipping: [master-2]
skipping: [master-3]
skipping: [worker-11]
skipping: [worker-10]
skipping: [worker-12]
skipping: [worker-13]
skipping: [worker-1]
skipping: [worker-15]
skipping: [worker-16]
skipping: [worker-17]
skipping: [worker-18]
skipping: [worker-19]
skipping: [worker-2]
skipping: [worker-20]
skipping: [worker-21]
skipping: [worker-22]

.....

TASK [upgrade/cordon : Cordon/Drain/Delete node] *****

Monday 03 January 2022 06:04:16 +0000 (0:00:01.430) 0:00:09.647 *****

changed: [worker-11 -> 10.192.x.x]

PLAY RECAP *****

master-1 rescued=0	ignored=0	: ok=1	changed=0	unreachable=0	failed=0	skipped=2
master-2 rescued=0	ignored=0	: ok=1	changed=0	unreachable=0	failed=0	skipped=2
master-3 rescued=0	ignored=0	: ok=1	changed=0	unreachable=0	failed=0	skipped=2
worker-11 rescued=0	ignored=0	: ok=1	changed=0	unreachable=0	failed=0	skipped=2
worker-10 rescued=0	ignored=0	: ok=1	changed=0	unreachable=0	failed=0	skipped=2
worker-11 rescued=0	ignored=0	: ok=2	changed=1	unreachable=0	failed=0	skipped=1
worker-12 rescued=0	ignored=0	: ok=1	changed=0	unreachable=0	failed=0	skipped=2
worker-13 rescued=0	ignored=0	: ok=1	changed=0	unreachable=0	failed=0	skipped=2
worker-1 rescued=0	ignored=0	: ok=1	changed=0	unreachable=0	failed=0	skipped=2
worker-15 rescued=0	ignored=0	: ok=1	changed=0	unreachable=0	failed=0	skipped=2
worker-16 rescued=0	ignored=0	: ok=1	changed=0	unreachable=0	failed=0	skipped=2
worker-17 rescued=0	ignored=0	: ok=1	changed=0	unreachable=0	failed=0	skipped=2
worker-18 rescued=0	ignored=0	: ok=1	changed=0	unreachable=0	failed=0	skipped=2
worker-19 rescued=0	ignored=0	: ok=1	changed=0	unreachable=0	failed=0	skipped=2
worker-2 rescued=0	ignored=0	: ok=1	changed=0	unreachable=0	failed=0	skipped=2
worker-20 rescued=0	ignored=0	: ok=1	changed=0	unreachable=0	failed=0	skipped=2
worker-21 rescued=0	ignored=0	: ok=1	changed=0	unreachable=0	failed=0	skipped=2
worker-22 rescued=0	ignored=0	: ok=1	changed=0	unreachable=0	failed=0	skipped=2

.....

=====

2022-01-03 06:04:17.957 DEBUG cluster_sync.cluster-name.worker-11: **Cluster sync successful**

2022-01-03 06:04:17.958 DEBUG cluster_sync.cluster-name.worker-11: **Ansible sync done**

2022-01-03 06:04:17.961 INFO cluster_sync.cluster-name.worker-11: **_sync finished. Opening lock**

8. kubernetes 마스터 노드를 확인하고 작업자 노드의 상태가 변경되었는지 확인합니다.

샘플 출력:

```
cloud-user@cluster-name-master-1:~$ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
cluster-name-master-1	Ready	control-plane,master	213d	v1.21.0
cluster-name-master-2	Ready	control-plane,master	213d	v1.21.0
cluster-name-master-3	Ready	control-plane,master	213d	v1.21.0
cluster-name-worker-11	Ready	<none>	213d	v1.21.0
cluster-name-worker-10	Ready	<none>	213d	v1.21.0
cluster-name-worker-11	Ready, SchedulingDisabled			
cluster-name-worker-12	Ready	<none>	213d	v1.21.0
cluster-name-worker-13	Ready	<none>	213d	v1.21.0
cluster-name-worker-11	Ready	<none>	213d	v1.21.0

9. 이 단계에서는 노드를 유지 관리할 준비가 되어 있어야 합니다. 단, 대말/복제 세트 등에서 관리할 수 있는 포드를 제외한 모든 애플리케이션 POD가 제거된 상태여야 합니다.

10. 서버가 다른 벤더에 속하는 경우 Cisco CIMC(Integrated Management Console) 또는 그에 상응하는 관리 콘솔에서 서버를 종료하고 하드웨어 유지 관리를 수행합니다.

유지 보수 후 서버가 다시 온라인 상태가 되고 모든 상태 확인이 녹색으로 표시되면 이 작업을 수행합니다.

11. Worker-Node를 Maintenance = "False"로 설정하여 다시 추가하고 동기화를 실행합니다.

샘플 출력:

```
[cluster-name-cm-1] SMI Cluster Deployer# config
Entering configuration mode terminal
[cluster-name-cm-1] SMI Cluster Deployer(config)# clusters cluster-name
[cluster-name-cm-1] SMI Cluster Deployer(config-clusters-cluster-name)# nodes worker-11
[cluster-name-cm-1] SMI Cluster Deployer(config-nodes-worker1)# maintenance false
[cluster-name-cm-1] SMI Cluster Deployer(config-nodes-worker1)# commit
Commit complete.
[cluster-name-cm-1] SMI Cluster Deployer(config-nodes-worker1)# end
```

12. 클러스터 동기화를 실행하여 순환 시 노드를 다시 가져오고 서비스를 제공할 준비가 되었습니다.

샘플 출력(잘림):


```
[cluster-name-cm-1] SMI Cluster Deployer# clusters cluster-name nodes worker-11 actions sync run
debug true
This will run sync. Are you sure? [no,yes] yes
message accepted
```

```
PLAY [Wait for ready and ensure uncordoned] *****
```

```
TASK [Wait for ready and ensure uncordoned] *****
```

```
Monday 03 January 2022 07:12:35 +0000 (0:00:01.151) 0:09:42.974 *****
```

```
skipping: [master-1] => (item=upgrade/wait-for-cluster-ready)
```

```
skipping: [master-1] => (item=upgrade/uncordon)
```

```
skipping: [master-2] => (item=upgrade/wait-for-cluster-ready)
```

```
skipping: [master-2] => (item=upgrade/uncordon)
```

```
skipping: [master-3] => (item=upgrade/wait-for-cluster-ready)
```

```
skipping: [master-3] => (item=upgrade/uncordon)
```

```
skipping: [worker-11] => (item=upgrade/wait-for-cluster-ready)
```

```
skipping: [worker-11] => (item=upgrade/uncordon)
```

```
skipping: [worker-10] => (item=upgrade/wait-for-cluster-ready)
```

```
skipping: [worker-10] => (item=upgrade/uncordon)
```

```
skipping: [worker-12] => (item=upgrade/wait-for-cluster-ready)
```

```
skipping: [worker-12] => (item=upgrade/uncordon)
```

```
skipping: [worker-13] => (item=upgrade/wait-for-cluster-ready)
```

```
skipping: [worker-13] => (item=upgrade/uncordon)
```

```
skipping: [worker-1] => (item=upgrade/wait-for-cluster-ready)
```

```
skipping: [worker-1] => (item=upgrade/uncordon)
```

```
.....
```

```
skipping: [worker-3] => (item=upgrade/wait-for-cluster-ready)
```

```
skipping: [worker-3] => (item=upgrade/uncordon)
```

```
skipping: [worker-4] => (item=upgrade/wait-for-cluster-ready)
```

```
skipping: [worker-4] => (item=upgrade/uncordon)
```

```
skipping: [worker-5] => (item=upgrade/wait-for-cluster-ready)
```

```
skipping: [worker-5] => (item=upgrade/uncordon)
```

```
skipping: [worker-6] => (item=upgrade/wait-for-cluster-ready)
```

```
skipping: [worker-6] => (item=upgrade/uncordon)
```

```
skipping: [worker-7] => (item=upgrade/wait-for-cluster-ready)
```

skipping: [worker-7] => (item=upgrade/uncordon)
skipping: [worker-8] => (item=upgrade/wait-for-cluster-ready)
skipping: [worker-8] => (item=upgrade/uncordon)
skipping: [worker-9] => (item=upgrade/wait-for-cluster-ready)
skipping: [worker-9] => (item=upgrade/uncordon)

TASK [upgrade/uncordon : Restore cordoned node] *****

Monday 03 January 2022 07:12:37 +0000 (0:00:01.539) 0:09:44.513 *****

changed: [worker-11 -> 10.192.x.x]

PLAY RECAP *****

master-1		: ok=38	changed=4	unreachable=0	failed=0	skipped=73
rescued=0	ignored=0					
master-2		: ok=35	changed=3	unreachable=0	failed=0	skipped=73
rescued=0	ignored=0					
master-3		: ok=35	changed=3	unreachable=0	failed=0	skipped=73
rescued=0	ignored=0					
worker-1		: ok=64	changed=3	unreachable=0	failed=0	skipped=83
rescued=0	ignored=0					
worker-10		: ok=63	changed=3	unreachable=0	failed=0	skipped=83
rescued=0	ignored=0					
worker-11		: ok=218	changed=30	unreachable=0	failed=0	skipped=306
rescued=0	ignored=0					
worker-12		: ok=63	changed=3	unreachable=0	failed=0	skipped=83
rescued=0	ignored=0					
worker-13		: ok=63	changed=3	unreachable=0	failed=0	skipped=83
rescued=0	ignored=0					
worker-11		: ok=63	changed=3	unreachable=0	failed=0	skipped=83
rescued=0	ignored=0					
.....						
worker-3		: ok=63	changed=3	unreachable=0	failed=0	skipped=83
rescued=0	ignored=0					
worker-4		: ok=63	changed=3	unreachable=0	failed=0	skipped=83
rescued=0	ignored=0					
worker-5		: ok=63	changed=3	unreachable=0	failed=0	skipped=83
rescued=0	ignored=0					
worker-6		: ok=63	changed=3	unreachable=0	failed=0	skipped=83
rescued=0	ignored=0					
worker-7		: ok=63	changed=3	unreachable=0	failed=0	skipped=83
rescued=0	ignored=0					

worker-8 : ok=63 changed=3 unreachable=0 failed=0 skipped=83
rescued=0 ignored=0

worker-9 : ok=63 changed=3 unreachable=0 failed=0 skipped=83
rescued=0 ignored=0

Monday 03 January 2022 07:12:38 +0000 (0:00:00.967) 0:09:45.481 *****

=====

2022-01-03 07:12:38.854 DEBUG cluster_sync.cluster-name.worker-11: **Cluster sync successful**

2022-01-03 07:12:38.858 DEBUG cluster_sync.cluster-name.worker-11: **Ansible sync done**

2022-01-03 07:12:38.860 INFO cluster_sync.cluster-name.worker-11: **_sync finished. Opening lock**
13. 클러스터의 상태를 확인합니다. Pod-desired-count는 ready-count와 일치해야 합니다.

[cluster-name-cm-1] SMI Cluster Deployer# clusters cluster-name actions k8s cluster-status

Pods-desired-count **678**

Pods-ready-count **678**

Pods-desired-are-ready true

etcd-healthy true

all-ok true