

Fehlerbehebung bei Hardwareproblemen im 5G SMI CNDP-Cluster und Durchführen von Wartungsarbeiten

Inhalt

[Einleitung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Hintergrundinformationen](#)

[Was ist SMI?](#)

[Was ist SMI-BM oder CNDP?](#)

[Was ist SMI Cluster Manager?](#)

[Was ist SMI Cluster Deployer?](#)

[Problem](#)

[Instandhaltungsverfahren](#)

Einleitung

Dieses Dokument beschreibt das Verfahren zur Durchführung von Wartungsarbeiten (Hardware-Ersatz oder -Wartung) wie Firmware-Upgrades usw. im 5-G-Subscriber Microservices Infrastructure (SMI) Cloud Native Deployment Platform (CNDP) Pool of Devices (POD).

Voraussetzungen

Anforderungen

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

- Cisco SMI
- 5G CNDPA- oder SMI-Bare-Metal-Architektur (BM)
- Dockers und kuberenttierte
- Cisco Server der Serie UCS C220

Verwendete Komponenten

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

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

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 BNG ermöglichen - alle mit ähnlichen Teilnehmerverwaltungsfunktionen und ähnlichen Datenspeicheranforderungen.

Attribute:

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

Was ist SMI-BM oder CNDP?

Cisco SMI-Bare Metal oder CNDP ist eine kuratierte Bare-Metal-Plattform, die die Infrastruktur für die Bereitstellung von Virtual Network Functions (VNF) und Cloud-Native Functions (CNFs) bereitstellt und die Geschäftseinheiten Cisco Mobility, Cable und BNG ermöglicht.

Attribute:

- Bare-Metal zur Eliminierung des VIM-bezogenen Overhead
- Verbesserte Leistung Mehr Kerne für Anwendungen Schnellere Anwendungsausführung
- Automatisierter Bereitstellungs-Workflow integriert mit NSO CFP
- Kurated Stack zur Bereitstellung von Cisco 5G NFs
- Vereinfachter Leitfadens für Bestellung und Bereitstellung

Was ist SMI Cluster Manager?

Ein Cluster-Manager ist ein Cluster mit zwei Nodes, der als Ausgangspunkt für die Cluster-Bereitstellung auf Kontrollebene und auf Benutzerebene verwendet wird. Es wird ein einzelnes, kubernetisches Cluster und eine Reihe von PODs ausgeführt, die für die gesamte Cluster-Einrichtung verantwortlich sind. Nur der primäre Cluster-Manager ist aktiv, und der sekundäre übernimmt nur bei einem Ausfall oder bei einem manuellen Ausfall zur Wartung.

Was ist SMI Cluster Deployer?

SMI Deployer ist ein Dienst im Cluster Manager, der VMs erstellen, Host-Betriebssysteme anpassen, Cluster-K8s erstellen, K8s Master starten, Cluster konfigurieren und Apps starten kann usw.

Problem

Hardwarewartung, z. B. Hardwareausfall, Software-/Fw-Upgrade usw., erfordert Ausfallzeiten in den Servern. Welches Verfahren muss eingehalten werden, damit Wartungsarbeiten am POD durchgeführt werden können? So stoppen Sie sanft die Services, um unerwünschte Ausfallzeiten in der Anwendung zu vermeiden.

Instandhaltungsverfahren

Rufen Sie den Cluster-Manager VIP, Kubuntu-Master-VIP (für die entsprechende Anwendung), UCS CIMC IP, UCS CIMC Name und den Server-Hostnamen (OS Hostname) ab, in dem die Wartung durchgeführt werden soll.

Die Anmeldung beim kuberneten Master entspricht dem Dienst und stellen sicher, dass alle PODs in **Running** Zustand sind.

Beispiel für das Ergebnis:

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

2. Melden Sie sich beim Cluster-Manager an, und greifen Sie auf das Ops-Center des SMI-Clusters zu (hier finden Sie das Verfahren zum Suchen der Ops-Center-IP).

```
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)
```

Beispiel für das Ergebnis:

```
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. Melden Sie sich mit diesem Befehl an.

```
ssh -p 2024 admin@10.100.x.x
(2024 is the port used to connect to cluster deployer)
```

4. Überprüfen Sie, ob die Dienste mit dem Befehl **show clusters** der Anwendung entsprechen.

Beispiel für das Ergebnis:

```
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. Entleeren Sie den Knoten, in dem Sie die Wartung mit diesen Befehlen durchführen, und geben Sie **Yes (Ja)** ein (dies leert die PODs sanft aus und startet bei Bedarf in anderen Knoten neu).

Beispiel für das Ergebnis:

```
[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. Verschieben Sie den Knoten mit diesen Befehlen in den Wartungsmodus (dies kann bis zu 30 Minuten dauern).

Beispiel für das Ergebnis:

```
[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. Überprüfen Sie die Protokolle auf den Status.

```
clusters cluster-name nodes worker-11 actions sync logs
```

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

Beispielausgabe (gekürzt):

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

rescued=0 ignored=0

.....

Monday 03 January 2022 06:04:17 +0000 (0:00:01.168) 0:00:10.815 *****

=====

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. Überprüfen Sie den kuberetteten Master-Knoten, und stellen Sie sicher, dass sich der Status des Workerknotens geändert hat.

Beispiel für das Ergebnis:

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. In diesem Schritt sollte der Knoten wartungsbereit sein (alle Anwendungs-PODs müssen entfernt worden sein, mit Ausnahme der PODs, die von Daemonset/Replicaset verwaltet werden usw., die ignoriert werden können).

10. Schließen Sie den Server von der Cisco Integrated Management Console (CIMC) oder einer gleichwertigen Verwaltungskonsole aus, wenn der Server einem anderen Anbieter angehört, und führen Sie die Hardwarewartung durch.

Wenn der Server nach der Wartung wieder online ist und alle Statusprüfungen grün sind, führen Sie dies aus.

11. Legen Sie für den Arbeitsknoten Maintenance = "False" fest, um wieder hinzugefügt und eine Synchronisierung ausgeführt zu werden.

Beispiel für das Ergebnis:

```
[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. Führen Sie die Cluster-Synchronisierung aus, um den Knoten bei der Rotation wiederherzustellen und bereit für den Betrieb zu sein.

Beispielausgabe (abgeschnitten):

```
[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. Überprüfen Sie den Status des Clusters. Die Anzahl der gewünschten PODs muss mit der Anzahl der fertigen PODs übereinstimmen.

```
[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
```