

排除5G SMI CNDP群集中的硬體問題並執行維護

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[背景資訊](#)

[什麼是SMI?](#)

[什麼是SMI-BM或CNDP?](#)

[什麼是SMI Cluster Manager?](#)

[什麼是SMI Cluster Deployer?](#)

[問題](#)

[維護程式](#)

簡介

本檔案介紹在5G使用者微服務基礎架構(SMI)Cloud Native Deployment Platform(CNDP)裝置池(POD)中執行維護 (硬體更換或維護) (例如韌體(FW)升級等)的程式。

必要條件

需求

思科建議您瞭解以下主題：

- Cisco SMI
- 5G CNDPA或SMI裸機(BM)架構
- 多克和庫伯內特
- Cisco UCS C220系列伺服器

採用元件

本文中的資訊係根據以下軟體和硬體版本：

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

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除 (預設) 的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

背景資訊

什麼是SMI?

Cisco SMI是雲技術和標準的分層堆疊，支援來自思科移動性、電纜和BNG業務部門的基於微服務的應用 — 所有這些業務部門都具有相似的使用者管理功能和相似的資料儲存要求。

屬性：

- 層雲堆疊（技術和標準），提供自上而下的部署，並適應當前的客戶雲基礎設施。
- 所有應用程式共用通用執行環境，用於非應用程式功能（資料儲存、部署、配置、遙測、警報）。這為所有客戶接觸點和整合點提供一致的互動和體驗。
- 應用和通用執行環境部署在微服務容器中，並與智慧服務網狀網連線。
- 公開的API用於部署、配置和管理，以實現自動化。

什麼是SMI-BM或CNDP?

Cisco SMI裸機或CNDP是一種受管理的裸機平台，它提供部署虛擬網路功能(VNF)和雲原生功能(CNF)的基礎設施，支援思科移動性、電纜和BNG業務部門。

屬性：

- 消除了與VIM相關的開銷的裸機
- 改進的效能 更多應用核心更快的應用程式執行
- 自動化部署工作流程；與NSO CFP整合
- 用於部署Cisco 5G NF的託管堆疊
- 簡化訂單和部署指南

什麼是SMI Cluster Manager?

集群管理器是一個雙節點保持可用的集群，用作控制平面和使用平面集群部署的初始點。它運行單個節點kubernetes群集和一組POD，它們負責整個群集設定。只有主群集管理器處於活動狀態，輔助群集管理器僅在出現故障或因維護而手動關閉時進行接管。

什麼是SMI Cluster Deployer?

SMI Deployer是群集管理器中的一項服務，可以建立VM、自定義主機OS、建立K8群集、啟動K8的主機、配置群集和啟動應用等。

問題

硬體維護（如硬體故障或軟體/韌體升級等）需要伺服器停機。在POD中執行維護需要遵循什麼程式。如何優雅地停止服務以避免應用程式出現意外停機。

維護程式

獲取要在其中執行維護的群集管理器VIP、Kubernetes主VIP（用於各自的應用程式）、UCS CIMC IP、UCS CIMC名稱和伺服器主機名（OS主機名）。

登入到kubernetes主目錄與服務對應，並確保所有POD都處於**運行**狀態。

輸出示例：

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

2.登入到群集管理器並訪問SMI群集部署器操作中心（下面是查詢操作中心IP的過程）。

```
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)
```

輸出示例：

```
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.使用此命令登入。

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

4.使用show clusters命令檢查與應用程式對應的服務。

輸出示例：

```
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.使用這些命令排空您執行維護的節點，然後鍵入Yes（這樣可以正常清空POD，並根據需要在其他節點中重新啟動）。

輸出示例：

```
[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.使用這些命令將節點移至維護模式 (最多可能需要30分鐘)。

輸出示例：

```
[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.檢查日誌的狀態。

clusters cluster-name nodes worker-11 actions sync logs

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

輸出示例 (截斷)：

```
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      : 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.檢查kubernetes主節點，並確保工作節點的狀態已更改。

輸出示例：

```
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.在此步驟中，節點應做好維護準備（所有應用程式POD必須已逐出，但守護程式/複製集等管理的POD除外，這些程式可以忽略）。

10.從Cisco Integrated Management Console(CIMC)或任何等效管理控制檯（如果伺服器屬於其他供應商）關閉伺服器，並執行硬體維護。

當伺服器在維護後重新聯機並且所有運行狀況檢查為綠色時，請執行此步驟。

11.將Worker-Node設定為Maintenance = "False"以新增回並運行同步。

輸出示例：

```

[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.運行群集同步以恢復節點輪換並準備提供服務。

輸出示例（截斷）：

```

[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.檢查集群的狀態。Pods-desired-count應與ready-count匹配。

```

```
[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
```