

Risoluzione dei problemi hardware nel cluster CNDP 5G SMI ed esecuzione della manutenzione

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Cos'è SMI?](#)

[Che cos'è SMI-BM o CNDP?](#)

[Cos'è SMI Cluster Manager?](#)

[Che cos'è SMI Cluster Deployer?](#)

[Problema](#)

[Procedura di manutenzione](#)

Introduzione

In questo documento viene descritta la procedura per eseguire la manutenzione (sostituzione o manutenzione dell'hardware), ad esempio l'aggiornamento del firmware (FW) e così via, in POD (Cloud Native Deployment Platform) 5G Subscriber Microservices Infrastructure (SMI).

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Cisco SMI
- Architettura 5G CNDPA o SMI-Bare-Metal (BM)
- Docker e kubernetes
- Cisco UCS serie C220 server

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

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

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

Cos'è SMI?

Cisco SMI è uno stack a più livelli di tecnologie e standard cloud che consentono applicazioni basate su microservizi dalle unità aziendali Cisco Mobility, Cable e BNG - tutte con funzioni di gestione degli abbonati e requisiti di datastore simili.

Attributi:

- Stack di cloud di livello (tecnologie e standard) per fornire installazioni top-to-bottom e supportare anche l'attuale infrastruttura cloud del cliente.
- Ambiente di esecuzione comune condiviso da tutte le applicazioni per funzioni non applicative (storage dei dati, installazione, configurazione, telemetria, allarme). Ciò consente un'interazione e un'esperienza coerenti per tutti i punti di contatto e i punti di integrazione dei clienti.
- Le applicazioni e Common Execution Environment vengono distribuite in contenitori di microservizi e collegate a una rete Mesh di servizio intelligente.
- API esposta per la distribuzione, la configurazione e la gestione, per consentire l'automazione.

Che cos'è SMI-BM o CNDP?

Cisco SMI-Bare Metal o CNDP è una piattaforma bare-metal gestita che fornisce l'infrastruttura per implementare le funzionalità di rete virtuale (VNF, Virtual Network Functions) e le funzionalità native del cloud (CNF, Cloud-Native Functions), consentendo l'uso delle unità aziendali Cisco Mobility, Cable e BNG.

Attributi:

- Bare metal che elimina il sovraccarico relativo a VIM
- Prestazioni migliorate Altri core per l'applicazione Esecuzione più rapida delle applicazioni
- Flusso di lavoro di installazione automatizzato; integrato con NSO CFP
- Stack curato per l'installazione di NF Cisco 5G
- Guida semplificata all'ordine e all'installazione

Cos'è SMI Cluster Manager?

Un cluster manager è un cluster con 2 nodi mantenuto come punto iniziale per la distribuzione di cluster control plane e userplane. Esegue un cluster kubernetes a nodo singolo e un set di POD che sono responsabili della configurazione dell'intero cluster. Solo il cluster manager primario è attivo e il cluster secondario subentra solo in caso di errore o viene disattivato manualmente per motivi di manutenzione.

Che cos'è SMI Cluster Deployer?

SMI Deployer è un servizio in Cluster Manager che può creare VM, personalizzare il sistema operativo host, creare il cluster K8s, avviare il master K8s, configurare il cluster, avviare le applicazioni e così via.

Problema

La manutenzione dell'hardware, ad esempio guasti hardware o aggiornamenti software e così via, richiede tempi di inattività dei server. Quale procedura è necessario seguire per eseguire la manutenzione nel POD? Come arrestare i servizi in modo corretto per evitare tempi di inattività indesiderati nell'applicazione.

Procedura di manutenzione

Ottenere l'indirizzo VIP del gestore cluster, l'indirizzo VIP del master Kubernetes (per la rispettiva applicazione), l'indirizzo IP CIMC UCS, il nome CIMC UCS e il nome host del server (nome host del sistema operativo) in cui eseguire la manutenzione.

Il login al master Kubernetes corrisponde al servizio e assicurarsi che tutti i POD siano in **esecuzione** condizione.

Output di esempio:

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

2. Accedere al cluster manager e al centro operativo del cluster SMI (procedura per trovare l'indirizzo IP del centro operativo).

```
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)
```

Output di esempio:

```
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. Accedere con questo comando.

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

4. Verificare che i servizi corrispondano all'applicazione con il comando **show clusters**.

Output di esempio:

```
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. Svotare il nodo in cui si esegue la manutenzione con questi comandi e digitare **Yes** (in questo modo i POD verranno svuotati e riavviati in altri nodi secondo necessità).

Output di esempio:

```
[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. Spostare il nodo in modalità manutenzione con questi comandi (l'operazione potrebbe richiedere fino a un massimo di 30 minuti).

Output di esempio:

```
[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. Controllare lo stato nei registri.

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

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

Output di esempio (troncato):

```
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. Controllare il nodo principale kubernetes e assicurarsi che lo stato del nodo di lavoro sia cambiato.

Output di esempio:

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 questa fase, il nodo deve essere pronto per la manutenzione (tutti i POD dell'applicazione devono essere stati eliminati ad eccezione dei pod gestiti da daemonset/replicaset, che possono essere ignorati).

10. Arrestare il server da Cisco Integrated Management Console (CIMC) o da qualsiasi console di gestione equivalente se il server appartiene a un altro fornitore ed eseguire la manutenzione dell'hardware.

Quando il server torna in linea dopo la manutenzione e quando tutti i controlli di integrità sono verdi, eseguire questa operazione.

11. Impostare Worker-Node su Maintenance = "False" per aggiungerlo nuovamente ed eseguire una sincronizzazione.

Output di esempio:


```
[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. Eseguire la sincronizzazione del cluster per riportare il nodo in rotazione e pronto per essere utilizzato.

Output di esempio (troncato):

```
[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. Controllare lo stato del cluster. Pods-desired-count deve corrispondere a ready-count.**

```
[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
```