

Solucionar problemas de hardware no cluster CNDP SMI 5G e executar manutenção

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[O que é SMI?](#)

[O que é SMI-BM ou CNDP?](#)

[O que é o SMI Cluster Manager?](#)

[O que é o instalador de cluster SMI?](#)

[Problema](#)

[Procedimento de manutenção](#)

Introduction

Este documento descreve o procedimento para executar manutenção (substituição ou manutenção de hardware) como atualização de firmware (FW) etc., no Pool de Dispositivos (POD) de CNDP (Cloud Native Deployment Platform) (Cloud Native Deployment Platform) (SMI) de Infraestrutura de Microserviços de Assinante de 5 G.

Prerequisites

Requirements

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Cisco SMI
- Arquitetura 5G CNDPA ou SMI-Bare-metal (BM)
- Dockers e kubernetes
- Servidores Cisco UCS série C220

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

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

The information in this document was created from the devices in a specific lab environment. All of

the devices used in this document started with a cleared (default) configuration. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

Informações de Apoio

O que é SMI?

O Cisco SMI é uma pilha em camadas de tecnologias e padrões de nuvem que permitem aplicativos baseados em microsserviços das unidades de negócios Cisco Mobility, Cable e BNG - todas com funções semelhantes de gerenciamento de assinantes e requisitos semelhantes de armazenamento de dados.

Atributos:

- Pilha de nuvem de camada (tecnologias e padrões) para fornecer implantações de ponta a ponta e também acomodar a infraestrutura de nuvem atual do cliente.
- Common Execution Environment compartilhado por todos os aplicativos para funções que não são de aplicação (armazenamento de dados, implantação, configuração, telemetria, alarme). Isso proporciona interação e experiência consistentes para todos os pontos de contato e pontos de integração do cliente.
- Os aplicativos e o Common Execution Environment são implantados em contêineres de microsserviço e conectados a uma malha de serviço inteligente.
- API exposta para implantação, configuração e gerenciamento, para permitir a automação.

O que é SMI-BM ou CNDP?

O Cisco SMI-Bare Metal ou CNDP é uma plataforma bare-metal curada que fornece a infraestrutura para implantar VNF (Virtual Network Functions, funções de rede virtual) e CNFs (Cloud-Native Functions, funções nativas de nuvem), habilita unidades de negócios de mobilidade, cabo e BNG da Cisco.

Atributos:

- Bare metal que elimina a sobrecarga relacionada ao VIM
- Melhor desempenho Mais núcleos para aplicativos Execução mais rápida de aplicativos
- Fluxo de trabalho de implantação automatizado; integrado à PCP NSO
- Pilha programada para implantar NFs Cisco 5G
- Guia simplificado de pedidos e implantação

O que é o SMI Cluster Manager?

Um gerenciador de cluster é um cluster mantido com 2 nós usado como o ponto inicial para o plano de controle e a implantação do cluster do plano de usuário. Ele executa um cluster kubernetes de nó único e um conjunto de PODs responsáveis por toda a configuração do cluster. Somente o gerenciador de cluster principal está ativo e o secundário assume somente em caso de falha ou interrompido manualmente para manutenção.

O que é o instalador de cluster SMI?

O SMI Deployer é um serviço no Cluster Manager que pode criar VMs, personalizar o SO do host, criar cluster K8s, iniciar K8s Master, configurar cluster e iniciar aplicativos etc.

Problema

A manutenção de hardware, como falha de hardware ou atualização de software/fw, etc., precisa de tempo de inatividade nos servidores. Qual procedimento precisa ser seguido para que a manutenção seja executada no POD. Como interromper graciosamente os serviços para evitar tempo de inatividade indesejado no aplicativo.

Procedimento de manutenção

Obtenha o VIP do gerenciador de cluster, o Kubernetes master VIP (para o respectivo aplicativo), o UCS CIMC IP, o UCS CIMC Name e o nome do host do servidor (OS hostname) no qual a manutenção deve ser executada.

O login no mestre kubernetes corresponde ao serviço e certifique-se de que todos os PODs estejam em condição **de execução**.

Saída de exemplo:

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

2. Faça login no cluster manager e acesse o centro de operações do cluster SMI (eis o procedimento para localizar o IP do centro de operações).

```
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)
```

Saída de exemplo:

```
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. Faça login com este comando.

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

4. Verifique se os serviços correspondem ao aplicativo com o comando **show clusters**.

Saída de exemplo:

```
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. Dissipe o nó em que você executa a manutenção com esses comandos e digite **Sim** (isso evacuará os PODs com cuidado e reiniciará em outros nós, conforme necessário).

Saída de exemplo:

```
[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. Mova o nó para o modo de manutenção com esses comandos (isso pode levar até um máximo de 30 minutos).

Saída de exemplo:

```
[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. Verifique o status nos registros.

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

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

Exemplo de saída (truncada):

```
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. Verifique o nó mestre do kubernetes e certifique-se de que o status do nó de trabalhador foi alterado.

Saída de exemplo:

```
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. Nesta etapa, o nó deve estar pronto para manutenção (todos os PODs do aplicativo devem ter sido removidos, exceto os pods gerenciados por daemonset/replicaset etc. que podem ser ignorados).

10. Desligue o servidor do Cisco Integrated Management Console (CIMC) ou de qualquer console de gerenciamento equivalente se o servidor pertencer a um fornecedor diferente e faça a manutenção de hardware.

Quando o servidor estiver novamente on-line após a manutenção e quando toda a verificação de integridade estiver verde, faça isso.

11. Defina o nó do trabalhador como Manutenção = "Falso" para ser adicionado de volta e executar uma sincronização.

Saída de exemplo:


```
[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. Execute a sincronização do cluster para retornar o nó na rotação e pronto para servir.

Exemplo de saída (truncada):

```
[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. Verifique o status do cluster. Pods-desejado-count deve corresponder 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
```