Troubleshoot Hardware Issues in 5G SMI CNDP Cluster and Perform Maintenance

Contents

Introduction

This document describes the procedure to perform maintenance (hardware replacement or maintenance) such as Firmware (FW) upgrade etc, in 5G Subscriber Microservices Infrastructure (SMI) Cloud Native Deployment Platform (CNDP) Pool of Devices (POD).

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco SMI
- 5G CNDPA or SMI-Bare-metal (BM) architecture
- Dockers and kubernetes
- Cisco UCS C220 series servers

Components used

The information in this document is based on these software and hardware versions:

- 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. If your network is live, ensure that you understand the potential impact of any command.

Background Information

What is SMI?

Cisco SMI is a layered stack of cloud technologies and standards that enable microservices-based applications from Cisco Mobility, Cable, and BNG business units – all of which have similar subscriber management functions and similar datastore requirements.

Attributes:

- Layer Cloud Stack (technologies and standards) to provide top to bottom deployments and also accommodate current customer cloud infrastructures.
- Common Execution Environment shared by all applications for non-application functions (data storage, deploy, configure, telemetry, alarm). This provides consistent interaction and experience for all customer touchpoints and integration points.
- Applications and Common Execution Environment are deployed in microservice containers and connected with an Intelligent Service Mesh.
- Exposed API for deployment, configuration, and management, to enable automation.

What is SMI-BM or CNDP?

Cisco SMI–Bare Metal or CNDP is a curated bare-metal platform that provides the infrastructure to deploy Virtual Network Functions (VNF) and Cloud-Native Functions (CNFs), enables Cisco Mobility, Cable, and BNG business units.

Attributes:

- Bare metal that eliminates the VIM related overhead
- Improved performance More Cores for applicationFaster application execution
- Automated deployment workflow; integrated with NSO CFP
- Curated Stack to deploy Cisco 5G NFs
- Simplified order and deployment guide

What is SMI Cluster Manager?

A cluster manager is a 2 node keepalived cluster used as the initial point for both control plane and userplane cluster deployment. It runs a single node kubernetes cluster and a set of PODs which are responsible for entire cluster setup. Only primary cluster manager is active and the secondary takes over only in case of a failure or brought down manually for maintenance.

What is SMI Cluster Deployer?

SMI Deployer is a service in Cluster Manager which can create VMs, customize Host OS, create K8s cluster, launch K8s Master, configure cluster, and launch Apps etc.

Problem

Hardware maintenance such as hardware failure or software/fw upgrade etc need downtime in the

servers. What procedure needs to be followed for maintenance to be performed in the POD. How to gracefully stop the services to avoid unwanted downtime in the application.

Procedure for the Maintenance

Obtain the cluster manager VIP, Kubernetes master VIP (for the respective application), UCS CIMC IP, UCS CIMC Name, and the server hostname (OS hostname) in which maintenance to be performed.

Login to the kubernetes master corresponds to the service and make sure all PODs are in **Running** condition.

Sample output:

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

2. Login to the cluster manager and access tne SMI cluster deployer Ops center (here is the procedure to find the 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)
```

Sample output:

```
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. Login with this command.

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

4. Check the services corresponds to the application with the **show clusters** command.

Sample output:

```
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. Drain the node in which you perform maintenance with these commands and type **Yes** (this will evacuate the PODs gracefully and re-start in other nodes as needed).

Sample output:

```
[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. Move the node to maintenance mode with these commands (this could take up to a maximum of 30 minutes).

Sample output:

```
[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. Check the logs for the status.

```
clusters cluster-name nodes worker-11 actions sync logs
(In this we are dealing with the worker-11 node)
```

Sample output (truncated):

```
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. Check the kubernetes master node and make sure the status of the worker node has changed.

Sample output:

```
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      <none>                  213d   v1.21.0

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. At this step, node should be ready for maintenance (all the application PODs must have been evicted except the pods managed by daemonset/replicaset etc which can be ignored).

10. Shutdown the server from Cisco Integrated Management Console (CIMC) or any equivalent management console if the server belongs to a different vendor, and perform the hardware maintenance.

When the server is back online after the maintenance and when all health check are green, perform this.

11. Set the Worker-Node to Maintenance = “False” to be added back and run a sync.

Sample output:

```
[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. Run the cluster sync to bring back the node on rotation and ready to serve.

Sample output (Truncated):

```
[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. Check the status of the cluster. Pods-desired-count should match 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
```