

Troubleshoot SMF PODs that Does not Come Up after the Day 1 Configuration

Contents

[Introduction](#)

[Prerequisite](#)

[Requirements](#)

[Components Used](#)

[Problem](#)

[Abbreviations](#)

[Observations](#)

[Symptoms](#)

[Troubleshoot](#)

[Solution](#)

Introduction

This document describes the issue of SMF NF PODs that does not come up after the Day 1 configuration is loaded in SMF ops-center.

Prerequisite

Requirements

Cisco recommends that you have knowledge of these topics:

- Subscriber Microservices Infrastructure (SMI)
- Docker
- Kubernetes
- 5G

Components Used

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

- SMI
- Ops-Center
- SMF

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.

Problem

In the customer setup, they have two SMF NF that runs with the same version. Both these SMF NFs were upgraded to the latest version last night. Before the upgrade, both the NF were having PODs in running state. The issue is seen only with one SMF i.e. SMF-IMS. The other POD SMF-DATA is upgraded and has all PODs in running state.

- SMF version before upgrade: smf.2020.01.0-12
- SMF version after upgrade: smf.2020.01.0-18

Abbreviations

SMF	Session Management Function
NF	Network Function
CEE	Common Execution Environment
POD	It is the smallest possible unit in the Kubernetes environment i.e. at least one container.
IMS	IP Multimedia Subsystem
SMI	Subscriber Microservices Infrastructure

Observations

- Cluster Sync shows Deployment Successful.
- Kubernetes Master shows the PODS in running state with Day zero configuration.
- When the Day-1 config is loaded, the new PODS does not come up.
- Inside SMF ops-center, you would see the helm charts in the deleted state.
- Change system mode runs to shut down and vice versa did not help.
- Add a new day-1 configuration also did not help.

Symptoms

- SMF-IMS NF shows the PODs with Day-0 configuration.
- Ops-center is allowing us to log in.
- CEE ops-center is up and running.
- SMF-DATA ops-center is up and running with day-1 config - This one is the another NF with working PODs.

```
~ubuntu@crucs501-cnat-cnat-core-master1:~$ kubectl get pods -n smf-ims
```

NAME	READY	STATUS	RESTARTS	AGE
api-smf-ims-ops-center-69f4d8f47b-hsqnx	1/1	Running	0	162m
base-entitlement-smf-998c8b84f-79r8v	1/1	Running	0	162m
documentation-65484db875-n41jq	1/1	Running	0	162m
ops-center-smf-ims-ops-center-6fb57bf79c-9dj29	5/5	Running	2	162m
smart-agent-smf-ims-ops-center-5dd679cf8b-hq4hs	1/1	Running	0	162m
swift-smf-ims-ops-center-745565bbf8-w5d7g	1/1	Running	0	162m

• Helm Chart status

```
crucs501-cnat/ims] smf# show helm
```

CHART RELEASE	INSTANCE	STATUS NAMESPACE	VERSION	REVISION	
infra-charts ims-infra-charts	- smf-ims	DELETED	0.0.2-master-0031-200306111921-107580e	1	smf-
smf-dashboard ims-smf-dashboard	- smf-ims	DELETED	0.0.2-master-0018-200113112417-b028370	1	smf-
smf-configuration ims-smf-configuration	- smf-ims	DELETED	0.0.6-master-1067-200303174113-9ee9665	1	smf-
li-ep ims-li-ep	- smf-ims	DELETED	0.0.1-master-0405-200306144054-3c56b02	1	smf-
smf-nodemgr ims-smf-nodemgr	- smf-ims	DELETED	0.0.2-master-3741-200304171906-5013914	1	smf-
smf-udp-proxy ims-smf-udp-proxy	- smf-ims	DELETED	0.0.2-master-1420-200305182644-ebb4bc9	1	smf-
gtpc-ep ims-gtpc-ep	- smf-ims	DELETED	0.0.3-master-0926-200305203830-3306ff4	1	smf-
smf-protocol ims-smf-protocol	- smf-ims	DELETED	0.0.2-master-4652-200304144735-d1e3798	1	smf-
smf-dns-proxy ims-smf-dns-proxy	- smf-ims	DELETED	0.1.0-master-0541-200304144718-b028370	1	smf-
smf-service ims-smf-service	- smf-ims	DELETED	0.0.5-master-18345-200305110040-5e8938b	1	smf-
smf-rest-ep ims-smf-rest-ep	- smf-ims	DELETED	0.3.3-master-6072-200304171221-7b0ff1a	1	smf-
etcd-cluster ims-etcd-cluster	- smf-ims	DELETED	0.5.2-master-0046-200305044107-60d06f1	1	smf-
ngn-datastore ims-ngn-datastore	- smf-ims	DELETED	1.0.1-master-0619-200305030353-d255520	1	smf-

Troubleshoot

1. Perform the cluster sync multiple times via SMI-Deployer without success
2. Day-1 configuration is verified.
3. Remove the Day-1 config and add back.

4. Delete the ops-center from Kubernetes master.
5. Entire config removal is performed.
6. Delete the Config Maps (CM).
7. Delete the helm charts from the master.
8. Delete the namespace.
9. Remove the supporting files from Deployer.
10. As the same new SMF build works fine on other deployments in the customer environment, it is ruled out that there is any issue with the image.
11. SMF-DATA on the same setup had come up without any issue.

Solution

1. Delete the cluster config of SMF-IMS ops-center from SMI deployer.
2. Sync the cluster.
3. Add back the config.
4. Sync the cluster.

There is one more workaround to solve this problem:

Delete the older version of the SMF package from the directory that SMI Deployer refers to while cluster sync.

Here is the configuration portion that was removed and added back from SMI Deployer ops-center running-config:

```
ops-centers smf ims

repository             https://charts.10.192.1.xxx.nip.io/smf.2020.01.0-18

sync-default-repository true

netconf-ip             10.241.69.xx

netconf-port           2024

ssh-ip                 10.241.69.xx

ssh-port               22

ingress-hostname       10.241.69.xx.nip.io

initial-boot-parameters use-volume-claims true
```

```
initial-boot-parameters first-boot-password <xxxxyyyzzz>
initial-boot-parameters auto-deploy false
initial-boot-parameters single-node false
exit
```

As per the deployments call flow, it is SMI Deployer who takes care of the extraction of the images for PODs from the package that is stored in it.

Normally, the downloaded Software package of SMF are stored local directory, from which SMI deployer extracts and shift them under this directory: **/data/software/packages/**

If the list of packages available under this directory is checked, you can see all the older packages as well available in it along with the new package list.

```
ubuntu@xxxxxx501-cnat-smi-cm-core-cm1:/data/software/packages$ ls -lrt
total 24
drwxrwxr-x 3 root root 4096 Mar 23 13:15 sample
drwxrwxr-x 3 root root 4096 Mar 24 05:48 smf.2020.01.0-12 >>> Older version of SMF
drwxrwxr-x 3 root root 4096 Mar 24 05:48 cee.2020.01.0-1
drwxrwxr-x 3 root root 4096 Apr 13 19:48 smf.2020.01.0-18 >>> Newer version of SMF
drwxr-xr-x 3 root root 4096 May  4 10:10 smf.2020.02.0.i66 >>> Older version os SMF
drwxr-xr-x 3 root root 4096 May  8 12:02 cee.2020.02.0
```

In this output, you can see there are three different SMF packages available. Even though the correct SMF version (i.e. smf.2020.01.0-18) is defined in SMI-Deployer running configuration, still, somehow the SMI-Deployer is not able to get the correct image files for that package.

After the workaround mentioned in the Solution section is performed, the issue was resolved.

Note: Similar issue is observed with CEE PODs as well, for which similar workaround is applied that is mentioned in the Solution section.