Ultra Cloud Core Subscriber Microservices Infrastructure - Troubleshooting Guide

First Published: 2020-11-12

Last Modified: 2023-07-21

CONTENTS

Contents

About this Guide

Note

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. While any existing biased terms are being substituted, exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

This preface describes the Cisco *Subscriber Microservices Infrastructure (SMI)*, how it is organized and its document conventions.

This guide describes common issues associated with the SMI components and its resolution.

Conventions Used

The following tables describe the conventions used throughout this documentation.

Notice Type	Description
Information Note	Provides information about important features or instructions.
Caution	Alerts you of potential damage to a program, device, or system.
Warning	Alerts you of potential personal injury or fatality. May also alert you of potential electrical hazards.

Typeface Conventions	Description
Text represented as a `screen display`	This typeface represents displays that appear on your terminal screen, for example: `Login:`

Typeface Conventions	Description
Text represented as **commands**	This typeface represents commands that you enter, for example: **show ip access-list** This document always gives the full form of a command in lowercase letters. Commands are not case sensitive.
Text represented as a **command** *variable*	This typeface represents a variable that is part of a command, for example: **show card** *slot_number* *slot_number* is a variable representing the desired chassis slot number.
Text represented as menu or sub-menu names	This typeface represents menus and sub-menus that you access within a software application, for example: Click the **File** menu, then click **New**

CHAPTER 1

SMI Cluster Manager - Validator

This chapter describes the procedures required to perform validations on the cluster configurations and clusters using the SMI Cluster Manager Validator (validation tool).

- SMI Cluster Manager - Validator, on page 1

SMI Cluster Manager - Validator

The SMI Cluster Manager comes equipped with a tool called Validator, which enables validating cluster configurations prior to deployment and validating clusters after deployment. With Validator, you can detect possible errors in cluster configuration before and after deployment. Loaded with a Synchronizer, the Validator performs validation on the cluster configurations from a SMI Cluster Manager perspective.

> ✎
>
> **Note** The Validator works only for VMware based installations.

Prerequisites

The following are the prerequisites for performing validations using SMI Cluster Manager Validator:

1. SMI Cluster Manager.

Validating Cluster Configurations using Validator

To validate the cluster configurations:

1. Login to the SMI Cluster Manager CLI.

2. Run the following command to validate the cluster configuration using the Validator.

   clusters *cluster_name* **actions validate-config run log-level DEBUG**

   Example:

   The following example initiates validation of cluster configuration using the Validator:

   ```
   SMI Cluster Manager# clusters cluster1 actions validate-config run log-level DEBUG
   This will run validation. Are you sure? [no,yes] yes
   ```

Note The Validator performs validation of cluster configuration in the following order:

    a. Validates netplan configuration.

    b. Validates SSH configuration.

    c. Validates NTP servers and associated configurations.

    d. Validates proxy configuration.

    e. Validates VMware configuration. For instance, it validates configuration details of vCenter, hosts, datacenter, datastores and so on.

    f. Validates node configuration. For instance, it validates the size requirements, number of control plane, ETCD and worker nodes, OAM node labeling requirements, and so on.

Validating the Clusters after Deployment using the Validator

Using the Validator, you can validate the status of different components within the cluster after deployment. To validate the clusters after deployment:

1. Login to the SMI Cluster Manager CLI.

2. Run the following command to validate the clusters using the Validator.

clusters *cluster_name* **actions validate-cluster run log-level DEBUG**

Example:

The following example initiates cluster validation using the Validator:

```
SMI Cluster Manager# clusters cluster1 actions validate-cluster run log-level DEBUG
This will run validation. Are you sure? [no,yes] yes
```

Note The Validator performs validation in the following order:

    a. Verifies the readiness of core-dns.

    b. Verifies the readiness of Worker nodes.

    c. Verifies whether the pods are up and running.

    d. Verifies Chrony status and clock skew.

    e. Verifies the connectivity to all ingresses.

Examples of CLI and REST API Mapping

The SMI Cluster Manager supports RESTCONF APIs. You can call these APIs through REST calls. At present, the SMI Cluster Manager APIs supports VM provisioning through VMware EsXi environment alone.

The following are examples illustrate the SMI Cluster Manager CLI and REST API mapping.

1. In the following example, the status of cluster synchronization is verified after deploying it through SMI Cluster Manager.

```
CLI: [installer-controlplane] SMI Cluster Manager# clusters <cluster-name> actions sync
 run
RESTAPI:
curl -k -X POST -u <username>:<password>
https://restconf.smi-cluster-manager.<ip>.nip.io/api/running/clusters/<cluster-name>/actions/sync/_operations/status
 -H "Content-Type: application/vnd.yang.operation+json"
{
  "tailf-smi-cloud:output": {
    "nodes": [
      {
        "node": "controlplane1",
        "state": "JOINED",
        "node": "controlplane2",
        "state": "JOINED",
        "node": "controlplane3",
        "state": "JOINED",
        "node": "oam1",
        "state": "JOINED",
        "node": "oam2",
        "state": "JOINED",
        "node": "oam3",
        "state": "JOINED",
        "node": "protocol1",
        "state": "JOINED",
        "node": "protocol2",
        "state": "JOINED",
        "node": "protocol3",
        "state": "JOINED",
        "node": "session1",
        "state": "JOINED",
        "node": "session2",
        "state": "JOINED",
        "node": "session3",
        "state": "JOINED"
      }
    ],
    "cluster": {
      "state": "DEPLOYED",
      "sync-status": "DONE"
    }
  }
}
```

2. In the following example, the status of a specific node is verified.

```
CLI: [installer-controlplane] SMI Cluster Manager# clusters <cluster-name> nodes
<node-name> actions k8s pod-status show-pod-details
Value for 'show-pod-details' [false,true]: true
RESTAPI:
curl -k -X POST -u <username>:<password>
https://restconf.smi-cluster-manager.<ip>.nip.io/api/running/clusters/<cluster-name>/nodes/<node-name>/actions/_operations/k8s/pod-status
 -H "Content-Type: application/vnd.yang.operation+json" -d '{ "show-pod-details":"true"}'
{
  "tailf-smi-cloud:output": {
    "pods": [
      {
        "name": "alertmanager-0",
        "namespace": "cee-data",
        "owner-kind": "StatefulSet",
        "owner-name": "alertmanager",
```

```
                            "ready": "true"
                        },
                        {
                          "name": "core-retriever-mp6gk",
                          "namespace": "cee-data",
                          "owner-kind": "DaemonSet",
                          "owner-name": "core-retriever",
                          "ready": "true"
                        },
                        {
                          "name": "logs-retriever-98qmj",
                          "namespace": "cee-data",
                          "owner-kind": "DaemonSet",
                          "owner-name": "logs-retriever",
                          "ready": "true"
                        },
                        {
                          "name": "node-exporter-knqdh",
                          "namespace": "cee-data",
                          "owner-kind": "DaemonSet",
                          "owner-name": "node-exporter",
                          "ready": "true"
                        },
                        {
                          "name": "path-provisioner-lbsvw",
                          "namespace": "cee-data",
                          "owner-kind": "DaemonSet",
                          "owner-name": "path-provisioner",
                          "ready": "true"
                        },
                        {
                          "name": "postgres-0",
                          "namespace": "cee-data",
                          "owner-kind": "StatefulSet",
                          "owner-name": "postgres",
                          "ready": "true"
                        },
                        {
                          "name": "postgres-1",
                          "namespace": "cee-data",
                          "owner-kind": "StatefulSet",
                          "owner-name": "postgres",
                          "ready": "true"
                        },
                        {
                          "name": "postgres-2",
                          "namespace": "cee-data",
                          "owner-kind": "StatefulSet",
                          "owner-name": "postgres",
                          "ready": "true"
                        },
                        {
                          "name": "bulk-stats-0",
                          "namespace": "cee-voice",
                          "owner-kind": "StatefulSet",
                          "owner-name": "bulk-stats",
                          "ready": "true"
                        },
                        {
                          "name": "core-retriever-wrpxn",
                          "namespace": "cee-voice",
                          "owner-kind": "DaemonSet",
                          "owner-name": "core-retriever",
                          "ready": "true"
```

```
            },
            {
              "name": "logs-retriever-725cr",
              "namespace": "cee-voice",
              "owner-kind": "DaemonSet",
              "owner-name": "logs-retriever",
              "ready": "true"
            },
            {
              "name": "path-provisioner-8tnzp",
              "namespace": "cee-voice",
              "owner-kind": "DaemonSet",
              "owner-name": "path-provisioner",
              "ready": "true"
            },
            {
              "name": "postgres-0",
              "namespace": "cee-voice",
              "owner-kind": "StatefulSet",
              "owner-name": "postgres",
              "ready": "true"
            },
            {
              "name": "postgres-1",
              "namespace": "cee-voice",
              "owner-kind": "StatefulSet",
              "owner-name": "postgres",
              "ready": "true"
            },
            {
              "name": "postgres-2",
              "namespace": "cee-voice",
              "owner-kind": "StatefulSet",
              "owner-name": "postgres",
              "ready": "true"
            },
            {
              "name": "prometheus-hi-res-0",
              "namespace": "cee-voice",
              "owner-kind": "StatefulSet",
              "owner-name": "prometheus-hi-res",
              "ready": "true"
            },
            {
              "name": "prometheus-rules-598b765cc6-7sb22",
              "namespace": "cee-voice",
              "owner-kind": "ReplicaSet",
              "owner-name": "prometheus-rules-598b765cc6",
              "ready": "true"
            },
            {
              "name": "prometheus-scrapeconfigs-synch-76fc9c848c-8vghv",
              "namespace": "cee-voice",
              "owner-kind": "ReplicaSet",
              "owner-name": "prometheus-scrapeconfigs-synch-76fc9c848c",
              "ready": "true"
            },
            {
              "name": "pv-manager-5cbbb67d5d-6kgd2",
              "namespace": "cee-voice",
              "owner-kind": "ReplicaSet",
              "owner-name": "pv-manager-5cbbb67d5d",
              "ready": "true"
            },
```

```
{
  "name": "istio-citadel-59bbc75849-kl9qp",
  "namespace": "istio-system",
  "owner-kind": "ReplicaSet",
  "owner-name": "istio-citadel-59bbc75849",
  "ready": "true"
},
{
  "name": "istio-pilot-567f7cf7b4-d4b87",
  "namespace": "istio-system",
  "owner-kind": "ReplicaSet",
  "owner-name": "istio-pilot-567f7cf7b4",
  "ready": "true"
},
{
  "name": "istio-sidecar-injector-566954b97d-xh6gl",
  "namespace": "istio-system",
  "owner-kind": "ReplicaSet",
  "owner-name": "istio-sidecar-injector-566954b97d",
  "ready": "true"
},
{
  "name": "calico-node-dkgx5",
  "namespace": "kube-system",
  "owner-kind": "DaemonSet",
  "owner-name": "calico-node",
  "ready": "true"
},
{
  "name": "coredns-64dfd65858-98trp",
  "namespace": "kube-system",
  "owner-kind": "ReplicaSet",
  "owner-name": "coredns-64dfd65858",
  "ready": "true"
},
{
  "name": "kube-proxy-bqd5r",
  "namespace": "kube-system",
  "owner-kind": "DaemonSet",
  "owner-name": "kube-proxy",
  "ready": "true"
},
{
  "name": "maintainer-cnfhg",
  "namespace": "kube-system",
  "owner-kind": "DaemonSet",
  "owner-name": "maintainer",
  "ready": "true"
},
{
  "name": "nginx-ingress-controller-8674778f6f-rdmnh",
  "namespace": "nginx-ingress",
  "owner-kind": "ReplicaSet",
  "owner-name": "nginx-ingress-controller-8674778f6f",
  "ready": "true"
},
{
  "name": "keepalived-zblx4",
  "namespace": "smi-vips",
  "owner-kind": "DaemonSet",
  "owner-name": "keepalived",
  "ready": "true"
}
],
```

```
    "pods-count": 28,
    "pods-available-to-drain-count": 17
  }
}
```

3. In the following example, a specified cluster configuration is validated.

```
CLI: [installer-controlplane] SMI Cluster Manager# clusters <cluster-name> actions
validate-config run log-level DEBUG
This will run validation.  Are you sure? [no,yes] yes
RESTAPI:
curl -k -X POST -u <username>:<password>
https://restconf.smi-cluster-manager.<ip>.nip.io/api/running/clusters/<cluster-name>/actions/validate-config/_operations/run
 -H "Content-Type: application/vnd.yang.operation+json" -d '{ "log-level":"DEBUG"}'
{
  "tailf-smi-cloud:output": {
    "message": "2019-12-15 21:11:52.801 INFO __main__: Verifying ntp config
......\n\n2019-12-15 21:11:52.801 DEBUG __main__: Collecting ntp servers at node-defaults
 level ......\n\n2019-12-15 21:11:52.801 DEBUG __main__: Collecting ntp servers at node
 level .....\n\n2019-12-15 21:11:52.802 DEBUG __main__: Running cmd: ntpdate -q
clock.cisco.com\n\n2019-12-15 21:11:52.811 DEBUG root: Running command: ntpdate -q
clock.cisco.com\n\n2019-12-15 21:11:58.964 INFO __main__: NTP server clock.cisco.com is
valid\n\n2019-12-15 21:11:58.964 INFO __main__: Verifying ssh keys config
......\n\n2019-12-15 21:11:58.964 DEBUG __main__: Running cmd: ssh-keygen -y -f
test-keys/private-key.pem\n\n2019-12-15 21:11:58.967 DEBUG root: Running command:
ssh-keygen -y -f test-keys/private-key.pem\n\n2019-12-15 21:11:58.968 INFO __main__:
Assinged Keys are in sync and correct\n\n2019-12-15 21:11:58.969 INFO __main__: Verifying
 netplan config ......\n\n2019-12-15 21:11:58.969 DEBUG __main__: Running cmd : netplan
 generate --debug --root-dir test-dir\n\n2019-12-15 21:11:58.970 DEBUG root: Running
command: netplan generate --debug --root-dir test-dir\n\n2019-12-15 21:11:59.043 INFO
__main__: Netplan config is correct\n\n2019-12-15 21:11:59.043 INFO __main__: Verifying
 http, https and no proxy configuration ......\n\n2019-12-15 21:11:59.043 DEBUG __main__:
 Collecting proxy servers at node-defaults level ......\n\n2019-12-15 21:11:59.044 DEBUG
 __main__: Collecting proxy servers at node level ......\n\n2019-12-15 21:11:59.044 DEBUG
 __main__: Running cmd: curl -s -o /dev/null http://proxy-wsa.esl.cisco.com:80
--connect-timeout 2 --max-time 2\n\n2019-12-15 21:11:59.046 DEBUG root: Running command:
 curl -s -o /dev/null http://proxy-wsa.esl.cisco.com:80 --connect-timeout 2 --max-time
2\n2019-12-15 21:11:59.068 INFO __main__: http-proxy and https-proxy in proxy config is
 valid\n\n2019-12-15 21:11:59.068 INFO __main__: no-proxy in proxy config is
valid\n\n2019-12-15 21:11:59.068 INFO __main__: Performing pre sync Vmware checks
......\n\n2019-12-15 21:11:59.148 DEBUG lib.vmware: Connecting to server:
dvtest-ccmts-vcs.cisco.com user: auto-ccmts@vsphere.local port: 443\n2019-12-15
21:11:59.218 INFO __main__: vcenter environment variables and credentials are
valid!\n\n2019-12-15 21:11:59.218 INFO __main__: Collecting the nodes of the cluster
....\n\n2019-12-15 21:11:59.218 INFO __main__: Control Plane nodes: 3, Etcd nodes: 0,
Worker nodes: 9, OAM nodes: 3\n\n2019-12-15 21:11:59.218 DEBUG __main__: Collecting the
 vsphere volume provider configuration if it exists ..... \n\n2019-12-15 21:11:59.218
DEBUG __main__: Assessing VMware vsphere node configuration ....\n\n2019-12-15 21:12:00.147
 INFO __main__: vmware node configuration is valid !\n\n2019-12-15 21:12:00.147 INFO
__main__: Performing kubernetes node configuration checks ......\n\n2019-12-15
21:12:00.147 DEBUG __main__: Checking kubernetes cluster node configuration, HA
configuration ..........\n\n2019-12-15 21:12:00.147 ERROR __main__: Expect at least 3
etcd nodes in a fucntional-test-ha or production deployment! Config is not valid
...\n\n2019-12-15 21:12:00.147 ERROR __main__: Checks failed in the cluster kali
are:\n\n2019-12-15 21:12:00.147 ERROR __main__: Check: k8s-node-checks failed.\n\n",
    "valid": "FALSE"
  }
}
```

4. In the following example, a deployed cluster is validated.

```
curl -k -X POST -u <username>:<password>
https://restconf.smi-cluster-manager.<ip>.nip.io/api/running/clusters/<cluster-name>/actions/validate-cluster/_operations/run
 -H "Content-Type: application/vnd.yang.operation+json"
{
```

```
"tailf-smi-cloud:output": {
  "message": "\nPLAY [Check Cluster Health]
*****************************************************\n\nTASK [Gathering Facts]
****************************************************\n\nSunday 15 December 2019
21:45:06 +0000 (0:00:00.060)        0:00:00.060 ******* \n\u001B[0;32mok:
[controlplane3]\u001B[0m\n\u001B[0;32mok: [controlplane1]\u001B[0m\n\u001B[0;32mok:
[controlplane2]\u001B[0m\n\nTASK [cluster-health : set common variables]
***********************************\nSunday 15 December 2019  21:45:07 +0000 (0:00:01.235)
      0:00:01.296 ******* \n\u001B[0;32mok: [controlplane1]\u001B[0m\n\u001B[0;32mok:
[controlplane2]\u001B[0m\n\u001B[0;32mok: [controlplane3]\u001B[0m\n\nTASK [cluster-health
 : Check core-dns ready replicas] **************************\nSunday 15 December 2019
21:45:08 +0000 (0:00:00.126)        0:00:01.422 ******* \n\u001B[0;32mok: [controlplane1
 -> 172.22.18.107]\u001B[0m\n\u001B[0;32mok: [controlplane2 ->
172.22.18.107]\u001B[0m\n\u001B[0;32mok: [controlplane3 -> 172.22.18.107]\u001B[0m\n\nTASK
 [cluster-health : Wait for Coredns to be Running] *************************\nSunday 15
 December 2019  21:45:08 +0000 (0:00:00.437)        0:00:01.860 ******* \n\u001B[0;32mok:
 [controlplane1 -> 172.22.18.107]\u001B[0m\n\u001B[0;32mok: [controlplane2 ->
172.22.18.107]\u001B[0m\n\u001B[0;32mok: [controlplane3 -> 172.22.18.107]\u001B[0m\n\nTASK
 [cluster-health : Wait for worker nodes are Ready - from cluster perspective] ***\nSunday
 15 December 2019  21:45:08 +0000 (0:00:00.411)        0:00:02.271 *******
\n\u001B[0;32mok: [controlplane1 -> 172.22.18.107]\u001B[0m\n\nTASK [cluster-health :
Wait for Pods to be Running] ****************************\nSunday 15 December 2019
21:45:09 +0000 (0:00:00.318)        0:00:02.589 ******* \n\u001B[0;32mok: [controlplane1
 -> 172.22.18.107]\u001B[0m\n\nTASK [cluster-health : verify_chrony_status]
***********************************\nSunday 15 December 2019  21:45:09 +0000 (0:00:00.300)
        0:00:02.889 ******* \n\u001B[0;33mchanged:
[controlplane1]\u001B[0m\n\u001B[0;33mchanged:
[controlplane2]\u001B[0m\n\u001B[0;33mchanged: [controlplane3]\u001B[0m\n\nTASK
[cluster-health : check_system_time] ***************************************\nSunday 15
December 2019  21:45:09 +0000 (0:00:00.261)        0:00:03.151 *******
\n\u001B[0;33mchanged: [controlplane1]\u001B[0m\n\u001B[0;33mchanged:
[controlplane3]\u001B[0m\n\u001B[0;33mchanged: [controlplane2]\u001B[0m\n\nTASK
[cluster-health : Fetch all ingresses] ************************************\nSunday 15
December 2019  21:45:10 +0000 (0:00:00.297)        0:00:03.448 ******* \n\u001B[0;32mok:
 [controlplane3]\u001B[0m\n\u001B[0;32mok: [controlplane1]\u001B[0m\n\u001B[0;32mok:
[controlplane2]\u001B[0m\n\nTASK [cluster-health : Verify curl all ingresses]
*****************************\nSunday 15 December 2019  21:45:10 +0000 (0:00:00.412)
      0:00:03.861 ******* \n\u001B[0;33mchanged: [controlplane1] =>
(item=docs.cee-data-product-documentation.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane2] =>
(item=docs.cee-data-product-documentation.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane3] =>
(item=docs.cee-data-product-documentation.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane1] =>
(item=cli.cee-data-ops-center.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
[controlplane3] =>
(item=cli.cee-data-ops-center.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
[controlplane1] =>
(item=documentation.cee-data-ops-center.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane3] =>
(item=documentation.cee-data-ops-center.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane2] =>
(item=cli.cee-data-ops-center.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
[controlplane3] => (item=grafana.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
[controlplane1] => (item=grafana.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
[controlplane2] =>
(item=documentation.cee-data-ops-center.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane3] =>
(item=restconf.cee-data-ops-center.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane2] => (item=grafana.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
[controlplane3] =>
(item=show-tac-manager.cee-data-smi-show-tac.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane2] =>
(item=restconf.cee-data-ops-center.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
```

```
                     [controlplane3] =>
(item=docs.cee-voice-product-documentation.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane2] =>
(item=show-tac-manager.cee-data-smi-show-tac.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane1] =>
(item=restconf.cee-data-ops-center.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane3] =>
(item=cli.cee-voice-ops-center.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
[controlplane2] =>
(item=docs.cee-voice-product-documentation.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane1] =>
(item=show-tac-manager.cee-data-smi-show-tac.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane1] =>
(item=docs.cee-voice-product-documentation.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane3] =>
(item=documentation.cee-voice-ops-center.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane2] =>
(item=cli.cee-voice-ops-center.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
[controlplane1] =>
(item=cli.cee-voice-ops-center.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
[controlplane3] => (item=grafana.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
[controlplane2] =>
(item=documentation.cee-voice-ops-center.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane1] =>
(item=documentation.cee-voice-ops-center.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane2] => (item=grafana.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
[controlplane2] =>
(item=restconf.cee-voice-ops-center.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane3] =>
(item=restconf.cee-voice-ops-center.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane2] =>
(item=show-tac-manager.cee-voice-smi-show-tac.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane3] =>
(item=show-tac-manager.cee-voice-smi-show-tac.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane1] => (item=grafana.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
[controlplane1] =>
(item=restconf.cee-voice-ops-center.172.22.18.113.nip.io)\u001B[0m\n\u001B[0;33mchanged:
 [controlplane1] =>
(item=show-tac-manager.cee-voice-smi-show-tac.172.22.18.113.nip.io)\u001B[0m\n\nPLAY
RECAP
**********************************************************************\n\u001B[0;33mcontrolplane1\u001B[0m
                     : \u001B[0;32mok=10   \u001B[0m \u001B[0;33mchanged=3    \u001B[0m
unreachable=0    failed=0    \n\u001B[0;33mcontrolplane2\u001B[0m                         :
\u001B[0;32mok=8    \u001B[0m \u001B[0;33mchanged=3    \u001B[0m unreachable=0    failed=0
    \n\u001B[0;33mcontrolplane3\u001B[0m                         : \u001B[0;32mok=8    \u001B[0m
 \u001B[0;33mchanged=3    \u001B[0m unreachable=0    failed=0    \n\nSunday 15 December
2019  21:45:36 +0000 (0:00:25.577)         0:00:29.438 *******
\n==============================================================================  \n",
     "valid": "TRUE"
  }
}
```

Troubleshooting Information

This chapter describes the common issues associated with SMI components and its resolution.

Debugging OpenStack CSI Plugin for Kubernetes

This section describes how to debug the OpenStack CSI plugin for Kubernetes.

To debug the OpenStack CSI plugin for Kubernetes, validate the following:

- Connection to the OpenStack metadata service.

- OpenStack credentials.

- Cinder CSI label on the nodes.

Validating the Connection to the OpenStack Metadata Service

1. Validate the connection to the OpenStack metadata service using the following curl command.

```
ubuntu@cn2smi-controlplane1:~$ curl http://169.254.169.254/openstack
2012-08-10
2013-04-04
2013-10-17
2015-10-15
2016-06-30
2016-10-06
2017-02-22
latest
```

2. If the curl command fails to display any output, verify whether a proper route is there for 169.254.169.254.

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.16.181.1    0.0.0.0         UG    100    0        0 ens4
```

```
10.192.1.0       0.0.0.0         255.255.255.0   U    0      0          0 bond0.2400
169.254.169.254 172.16.180.1    255.255.255.255 UGH  100    0          0 ens3
169.254.169.254 172.16.181.1    255.255.255.255 UGH  100    0          0 ens4
```

Validating OpenStack Credentials

1. From the control plane node, debug the loaded OpenStack credentials using the following command.

   ```
   kubectl get secrets -n kube-system cloud-config -o json  | grep '"cloud.conf"'  | cut
   -f 4 -d \" | base64 --decode
   ```

   A sample output is shown below:

   ```
   [Global]
   username=core
   password="Csco@123"
   auth-url=http://10.81.68.210:5000//v3
   tenant-id=eb3c1fda05ab48faad1f94e015b84f63
   domain-id=default

   [BlockStorage]
   bs-version=auto
   ```

   Verify if the values specified in the following fields are accurate:

   - Username

   - Password

   - Auth URL

   - Tenant ID

   - Domain ID

   The OpenStack CSI plugin will fail to connect, if the specified values are inaccurate.

Validating the Cinder CSI Label on the Nodes

The following label must be configured on the nodes with openstack CSI controller plugin and node-plugin pods.

```
"topology.cinder.csi.openstack.org/zone":"nova"
```

For verifying the labels on the primary control plane node, use the following command:

```
kubectl get nodes --show-labels
```

ErrImagePull for System Kubernetes Images

This section describes how to resolve the ErrImagePull status in system Kuberenetes images.

Problem

The Kubernetes kubelet cmdlet attempts to purge unused disk images and pods from the system, when the image file system (or file system) is under pressure (disk space reaches lesser than 15%). This results in the system Kubernetes images being purged from the system and the pods end up in an "ErrImagePull" status.

A sample "ErrImagePull" status appearing for the kube-system pods are shown below:

```
kube-system     calico-node-lnz6g                                          0/1
Init:ErrImagePull   0           41m     40.40.40.154    cn-svi-tb10-service2
kube-system     kube-proxy-zkk46                                           0/1
ErrImagePull        0           51m     40.40.40.154    cn-svi-tb10-service2
smi-vips        keepalived-km4wx                                          0/2
ErrImagePull        0           40m     40.40.40.154    cn-svi-tb10-service2
```

Resolution

You can restore the local images to resolve the "ErrImagePull" status in system Kubernetes images.

To restore the local images, execute the following command.

```
sudo docker load < /var/tmp/k8s-offline.tgz
```

Kubernetes Volumes Freezes on Node Shutdown

In the event of an unclean node shutdown, the detach and reattach logic of persistent volumes in Kubernetes are prone for issues. For more information on the issues, see https://github.com/kubernetes/kubernetes/issues/65392. This section describes the root cause of these issues and how to resolve these issues in the production system.

Problem

The following are some of the scenarios where the Kubernetes volumes freezes after an unclean node shutdown:

- When you shutdown a node that has a non-local persistent volumes (Cinder or vSphere) attached to it without using the cordon or drain approach recommended by Kubernetes.

```
ubuntu@cn2smi-controlplane1:~$ kubectl get nodes
NAME                    STATUS      ROLES           AGE    VERSION
cn2smi-controlplane1    Ready       control-plane   21h    v1.15.3
cn2smi-controlplane2    Ready       control-plane   21h    v1.15.3
cn2smi-controlplane3    Ready       control-plane   21h    v1.15.3
cn2smi-oam1             NotReady    <none>          14h    v1.15.3
```

- When the pods are in "creation" or "initialization" state waiting for the volumes to attach. The pods freeze approximately after thirty seconds.

```
ubuntu@cn2smi-controlplane1:~$ kubectl get pods -w -A -o wide| grep oam | grep bulk
cee-global     bulk-stats-68dc684d57-hqtdj                               3/3
    Terminating       0         4m38s   192.200.7.68    cn2smi-oam1
<none>          <none>
cee-global     bulk-stats-68dc684d57-sdjgl                               0/3
    ContainerCreating   0         15s     <none>          cn2smi-oam2
<none>          <none>
```

- The pods in the "ContainerCreating" state displays a "Multi-Attach error" event.

```
Events:
  Type     Reason           Age    From                    Message
  ----     ------           ----   ----                    -------
  Normal   Scheduled        76s    default-scheduler       Successfully assigned
cee-global/bulk-stats-68dc684d57-sdjgl to cn2smi-oam2
  Warning  FailedAttachVolume 76s   attachdetach-controller  Multi-Attach error for
volume "pvc-b01a4434-190f-4eec-b289-41ae990b0025" Volume is already used by pod(s)
bulk-stats-68dc684d57-hqtdj
  ubuntu
```

Resolution

You can delete the "Terminating" node to resolve this issue. To delete the "Terminating" node, run the following command:

```
kubectl delete pod -n <namespace> <pod-name> --force --grace-period=0
```

> **Note** You must use the `grace-period=0` and `force` options for deleting the node.

The pod reschedules approximately after seven minutes .

Log Forwarding - Missing or No Log

This section describes how to resolve the "missing logs" issue when Log Forwarding is enabled.

Problem

Log forwarding is enabled, but logs are missing or not seen on the external collector.

Resolution

1. Ping the Kubernetes node from the collection host to verify the network connectivity between each Kubernetes nodes and the collection server (where Fluentd or Fluentbit endpoint is hosted).

   ```
   ping [k8s-node-ip]
   ```

2. Verify whether the client machine's system clock is in synchronization with the Kubernetes node's clock. Any mismatch in the time will result in incorrect query in the front-end visualization tool (Kibana or Grafana), which uses the host clock as query parameter.

3. When Logs Forwarding is enabled, the Logs Forwarder dumps the entire JournalD entries - from the beginning - to the external collector. Also, the number of nodes on the deployment increases the amount of data available for processing. It may take a while for pushing out and processing the log entries before the frontend tools visualize it.

Show TAC Missing or No Information

This section describes how to resolve the failure of logs or Cores with the Show TAC feature.

Problem

When the Show TAC feature is executed, the logs or cores fail to show up in the Apache server.

Resolution

1. Verify the status of the Show TAC process.

   ```
   tac-debug-pkg status
   ```

2. Verify the parameters and syntax.

> • The TAC package creation time must be in *yyyy-mm-dd_hh:mm:ss* format.

> • The parameters specified for filtering must match the microservices labeling.

3. In the target node, run the following command to verify if the targeted JournalD entries or Core files are within the specified time range.

```
journalctl
```

✎

Note The first line of the output specifies the starting time of the current logs repository. Also, JournalID starts aging out old entries (log rotation) when the log size reaches the specified limit.

Similarly, for the Core files, run the following command to display the available Core files and its corresponding time stamp.

```
ls -al /var/lib/systemd/coredump
```

4. Verify if the JournalID is rate-limiting the entries in the target node. Filter for statements like the one shown in the following example:

```
journalctl | grep Suppressed
Suppressed <number> messages from /system.slice/...
```

✎

Note JournalD is configured to allow 10000 messages per 10 seconds period.

Recovering Postgres Authentication

This section describes how to resolve recovering Postgres authentication issue.

Problem

The Postgres database is password protected. The passwords are stored as Kubernetes secrets. Also, the nodes in the Kubernetes cluster store the persisted data in the following location:

```
/data/<cee-namespace>/data-postgres-x.
```

You may encounter an authentication error while accessing Postgres database. A sample authentication error message is shown below:

```
2019-11-12 21:32:18.856 UTC [239] FATAL:  password authentication failed for user
"replica_user"
2019-11-12 21:32:18.856 UTC [239] DETAIL:  Password does not match for user "replica_user".
```

The following are the probable causes for the authentication error:

> • The Kubernetes secrets are deleted.

> • The `/data` folder is corrupted.

Resolution

When the Kubernetes secrets are deleted

When you remove the CEE Ops Center deployment from the SMI Cluster Manager, the CEE namespace is also deleted. This action in turn deletes the secrets from the Kubernetes, which restricts access to the old database. Even when you have the CEE Ops Center deployed again, you will only have a new secret created and cannot restore the deleted secret.

To resolve this issue:

- Delete the persisted database manually on the OAM nodes located here:

  `/data/<cee-namespace>/data-postgres-x`). or

- Upgrade the deployed software of the network function. To upgrade, perform the following:

  1. Edit the default helm repository the software uses through NETCONF/RESTCONF or CLI.

     helm repository base-repos url *<url from the SMI deployer>*

     Example:

     ```
     helm repository base-repos url https://charts.10.192.1.101.nip.io/cee-2019-09-30
     ```

> **Note** The **show system status** command indicates the percentage of software upgrade process completed for the network function.

When the /data is corrupted

When the `/data` is corrupted, you must remove the `/data` to resolve the authentication error.

Network Connection Reset

This section describes how to resolve the network connection reset issue in Geographical Redundancy (GR) deployment.

Problem

During GR deployments, the SMI Cluster Manager is deployed on one or more remote sites. When the SMI Cluster Manager is deployed in at least two remote sites and if those deployed SMI Cluster Managers use a common *Vrrp-router ID* over a common VLAN/network with different IPs for running the traffic on SBI interface, the network connection resets with the following error:

Network error connection resets by peer

Resolution

The network connection resets because the Virtual IP (VIP) router ID is not unique across the two SMI Cluster Managers. You must ensure that the VIP router ID is unique while configuring the K8s clusters. For more details on configuring the cluster, see *Clusters Configuration* section in *UCC SMI Operations Guide* Guide.

Vulnerability Scan Reports a False Error Message During the Directory Listing

This section describes how to handle the false error that gets displayed during the vulnerability scan.

Problem

The Show TAC debug package interface requires the Director Listing feature configured. For debugging issues, the operator collects the TAC logs from the directory listing. However, the vulnerability scan incorrectly reports a security error during the listing.

Resolution

The error is invalid since the show-tac-url is secured through password protection, indicating no vulnerability threat associated with this operation.