



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-lbsw",
    "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\n\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\n\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 fuctional-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]
*****\nSunday 15 December 2019
21:45:06 +0000 (0:00:00.060)    0:00:00.060 ***** \n\u001B[0;32mok:
[controlplane3]\u001B[0m\u001B[0;32mok: [controlplane1]\u001B[0m\u001B[0;32mok:
[controlplane2]\u001B[0m\u001B[0;32mok: [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\u001B[0;32mok:
[controlplane2]\u001B[0m\u001B[0;32mok: [controlplane3]\u001B[0m\u001B[0;32mok:
: 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\u001B[0;32mok: [controlplane2 ->
172.22.18.107]\u001B[0m\u001B[0;32mok: [controlplane3 -> 172.22.18.107]\u001B[0m\u001B[0;32mok:
[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\u001B[0;32mok: [controlplane2 ->
172.22.18.107]\u001B[0m\u001B[0;32mok: [controlplane3 -> 172.22.18.107]\u001B[0m\u001B[0;32mok:
[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\u001B[0;32mok: [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\u001B[0;32mok: [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\u001B[0;33mchanged:
[controlplane2]\u001B[0m\u001B[0;33mchanged: [controlplane3]\u001B[0m\u001B[0;33mchanged:
[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\u001B[0;33mchanged:
[controlplane3]\u001B[0m\u001B[0;33mchanged: [controlplane2]\u001B[0m\u001B[0;33mchanged:
[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\u001B[0;32mok: [controlplane1]\u001B[0m\u001B[0;32mok:
[controlplane2]\u001B[0m\u001B[0;32mok: [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\u001B[0;33mchanged:
[controlplane2] =>
(item=docs.cee-data-product-documentation.172.22.18.113.nip.io)\u001B[0m\u001B[0;33mchanged:
[controlplane3] =>
(item=docs.cee-data-product-documentation.172.22.18.113.nip.io)\u001B[0m\u001B[0;33mchanged:
[controlplane1] =>
(item=cli.cee-data-ops-center.172.22.18.113.nip.io)\u001B[0m\u001B[0;33mchanged:
[controlplane3] =>
(item=cli.cee-data-ops-center.172.22.18.113.nip.io)\u001B[0m\u001B[0;33mchanged:
[controlplane1] =>
(item=documentation.cee-data-ops-center.172.22.18.113.nip.io)\u001B[0m\u001B[0;33mchanged:
[controlplane3] =>
(item=documentation.cee-data-ops-center.172.22.18.113.nip.io)\u001B[0m\u001B[0;33mchanged:
[controlplane2] =>
(item=cli.cee-data-ops-center.172.22.18.113.nip.io)\u001B[0m\u001B[0;33mchanged:
[controlplane3] => (item=grafana.172.22.18.113.nip.io)\u001B[0m\u001B[0;33mchanged:
[controlplane1] => (item=grafana.172.22.18.113.nip.io)\u001B[0m\u001B[0;33mchanged:
[controlplane2] =>
(item=documentation.cee-data-ops-center.172.22.18.113.nip.io)\u001B[0m\u001B[0;33mchanged:
[controlplane3] =>
(item=restconf.cee-data-ops-center.172.22.18.113.nip.io)\u001B[0m\u001B[0;33mchanged:
[controlplane2] => (item=grafana.172.22.18.113.nip.io)\u001B[0m\u001B[0;33mchanged:
[controlplane3] =>
(item=show-tac-manager.cee-data-smi-show-tac.172.22.18.113.nip.io)\u001B[0m\u001B[0;33mchanged:
[controlplane2] =>
(item=restconf.cee-data-ops-center.172.22.18.113.nip.io)\u001B[0m\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\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"
  }
}

```

