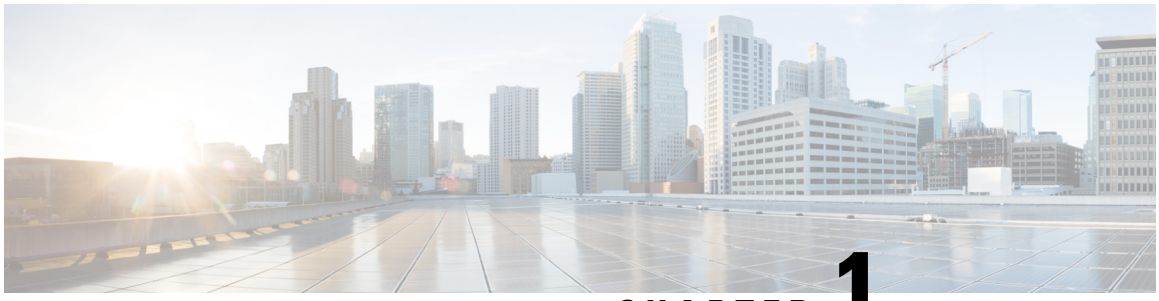
Cisco Virtualized Infrastructure Manager Administrator Guide, Release 2.0

First Published: 2017-05-23

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Managing Cisco NFVI

The following topics provide general management procedures that you can perform if your implementation is Cisco VIM by itself or is Cisco VIM and Cisco VIM Insight.

Managing Cisco NFVI Pods

You can perform OpenStack management operations on Cisco NFVI pods including addition and removal of Cisco NFVI compute and Ceph nodes, and replacement of controller nodes. Each action is mutually exclusive. Only one pod management action can be performed at any time. Before you perform a pod action, verify that the following requirements are met:

- The node is part of an existing pod.
- The node information exists in the setup_data.yaml file, if the pod management task is removal or replacement of a node.
- The node information does not exist in the setup_data.yaml file, if the pod management task is to add a node.

  To perform pod actions, see the Managing Hosts in Cisco VIM or NFVI Pods , on page 3section.

General Guidelines for Pod Management

The setup_data.yaml file is the only user-generated configuration file that is used to install and manage the cloud. While many instances of pod management dictates that the setup_data.yaml file is modified, the administrator does not update the system generated setup_data.yaml file directly.

To update the setup_data.yaml file, do the following:

1. Copy the setup data into a local directory

```
[root@mgmt1 ~]# cd /root/
[root@mgmt1 ~]# mkdir MyDir
[root@mgmt1 ~]# cd MyDir
[root@mgmt1 ~]# cp /root/openstack-configs/setup_data.yaml <my_setup_data.yaml>
```

2. Update the setup data manually.

```
[root@mgmt1 ~]# vi my_setup_data.yaml (update the targeted fields for the setup_data)
```

3. Run the reconfiguration command:

```
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim --setupfile ~/MyDir/<my_setup_data.yaml>
<pod_management_action>
```

In Cisco VIM 2.0, you can edit and enable a selected set of options in the setup_data.yaml file using the reconfigure option. After installation, you can change the values of the feature parameters. But, Cisco VIM 2.0 does not support deconfiguring of the feature .

The following table summarizes the list of features that can be reconfigured after the installation of the pod.

Features that can be enabled post pod deployment	Comments
Optional Services	• Heat: OpenStack Orchestration Program. • Keystone v3: Pod running Keystone v2 can be migrated to Keystone v3. • LDAP: Works only with Keystone v3. Full or partial reconfiguration can be done. All attributes except domain is reconfigurable.
Pod Monitoring	• Collectd: Monitors Systems at host level. • NFVIMON: Third party Monitoring from host to service level; Does need involvement and planning with Cisco Advance Services. Also, supported with only Keystone v2 in VIM 2.0.
Export of ELK logs to External Syslog Server	Reduces single point of failure on management node and provides data aggregation.
Admin Source Networks	White list filter for accessing management node admin service.
NFVBench	Tool to help measure cloud performance. Management node needs a 10G Intel NIC (4x10G 710, or 2x10G 520 Intel NIC).

Features that can be enabled post pod deployment	Comments
ELK settings	ELK rotation frequency and size.
OpenStack service password	Implemented for security reasons, so that OpenStack passwords can be reset on-demand.
SwiftStack Post Install	Integration with third party Object-Store. SwiftStack install is done independent of VIM and ahead of time; only works with Keystone v2.
TENANT_VLAN_RANGES and PROVIDER_VLAN_RANGES	Ability to increase the tenant and provider vlan ranges on a pod that is up and running. It gives the customer the flexibility in network planning.

Identifying the Install Directory

If the administrator is using CLI to manage the pod, the administrator must know the directory where the pod is installed from(refer to installer directory). To identify the installer directory of a pod, execute the following commands:

```
[root@mgmt1 ~]# cd /root/
[root@mgmt1 ~]# ls -lrt | grep openstack-configs
lrwxrwxrwx.  1 root root      38 Mar 12 21:33 openstack-configs ->
/root/installer-<tagid>/openstack-configs
```

From the output, you can understand that the OpenStack-configs is a symbolic link to the installer directory.

Verify that the REST API server is running from the same installer directory location, by executing the following command:

```
# cd installer-<tagid>/tools
#./restapi.py -a status
Status of the REST API Server:   active (running) since Thu 2016-08-18 09:15:39 UTC; 9h ago
REST API launch directory: /root/installer-<tagid>/
```

Managing Hosts in Cisco VIM or NFVI Pods

To perform actions on the pod, run the commands specified in the following table. If you log in as root, manually change the directory to /root/installer-xxx to get to the correct working directory for these Cisco NFVI pod commands.

Table 1: Cisco NFVI Pod Management

Action	Steps	Restrictions

Remove block_storage or compute node	1. Remove the node information from the ROLES and SERVERS section of the setup_data.yaml file for the specific node.
	2. Enter one of the following commands.
	For compute nodes:
	`./ciscovimclient/ciscovim remove-computes --setupfile ~/MyDir/my_setup_data.yaml <"compute-1,compute-2">`
	For storage nodes:
	`./ciscovimclient/ciscovim remove-storage --setupfile ~/MyDir/my_setup_data.yaml <"storage-1">`
		You can remove multiple compute nodes and only one storage at a time;
		The pod must have a minimum of one compute and two storage nodes after the removal action.
		In Cisco VIM the number of ceph OSD nodes can vary from 3 to 20. You can remove one OSD node at a time as part of the pod management.

Add block_storage or compute node	1. Add the node information from the ROLES and SERVERS section of the setup_data.yaml file for the specific node.
	2. Enter one of the following commands.
	For compute nodes:
	`./ciscovimclient/ciscovim add-computes --setupfile ~/MyDir/my_setup_data.yaml <"compute-1,compute-2">`
	For storage nodes:
	`./ciscovimclient/ciscovim add-storage --setupfile ~/MyDir/my_setup_data.yaml <"storage-1">`
		You can add multiple compute nodes and only one storage node at a time.
		The pod must have a minimum of one compute, and two storage nodes before the addition action.
		In Cisco VIM the number of ceph OSD nodes can vary from 3 to 20. You can add one OSD node at a time as part of the pod management.

Action	Steps	Restrictions
Replace controller node	1. If the controller node is in a UCS C-Series pod, update the CIMC info node in the SERVERS section of the setup_data.yaml file for the specific node 2. For B-series only update the blade and chassis info 3. Enter the following command: **`./ciscovimclient/ciscovim replace-controller --setupfile ~/MyDir/my_setup_data.yaml <"control-1">`**	You can replace only one controller node at a time. The pod can have a maximum of three controller nodes. In Cisco VIM the replace controller node operation is supported in micro-pod. **Note** While replacing the controller node, the IP address and hostname are reused. So, do not update any other controller information other than CIMC access for C-series, and blade and chassis information for B-series.

When you add a compute or storage node to a UCS C-Series pod, you can increase the management/provision address pool. Similarly, for a UCS B-Series pod, you can increase the Cisco IMC pool to provide routing space flexibility for pod networking. Along with server information, these are the only items you can change in the setup_data.yaml file after the pod is deployed. To make changes to the management or provisioning sections and/or CIMC (for UCS B-Series pods) network section, you must not change the existing address block as defined on day 0. You can add only to the existing information by adding new address pool block(s) of address pool as shown in the following example:

```
NETWORKING:
  :
  :

  networks:
  -
    vlan_id: 99
    subnet: 172.31.231.0/25
    gateway: 172.31.231.1
    ## 'pool' can be defined with single ip or a range of ip
    pool:
      - 172.31.231.2, 172.31.231.5 -→ IP address pool on Day-0
      - 172.31.231.7 to 172.31.231.12 -→ IP address pool ext. on Day-n
      - 172.31.231.20
    segments:
    ## CIMC IP allocation. Needs to be an external routable network
      - cimc
  -
    vlan_id: 2001
    subnet: 192.168.11.0/25
    gateway: 192.168.11.1
    ## 'pool' can be defined with single ip or a range of ip
    pool:
      - 192.168.11.2 to 192.168.11.5   -→ IP address pool on Day-0
      - 192.168.11.7 to 192.168.11.12  → IP address pool on day-n
      - 192.168.11.20 → IP address pool on day-n
    segments:
    ## management and provision goes together
      - management
   - provision
```

```
:
:
```

The IP address pool is the only change allowed in the networking space of the specified networks management/provision and/or CIMC (for B-series). The overall network must have enough address space to accommodate for future enhancement on day-0. After making the changes to servers, roles, and the corresponding address pool, you can execute the add compute/storage CLI shown above to add new nodes to the pod.

Recovering Cisco NFVI Pods

This section describes the recovery processes for Cisco NFVI control node and the pod that is installed through Cisco VIM. For recovery to succeed, a full Cisco VIM installation must have occurred in the past, and recovery is caused by failure of one or more of the controller services for example, Rabbit MQ, MariaDB, and other services. The management node must be up and running and all the nodes must be accessible through SSH without passwords from the management node. You can also use this procedure to recover from a planned shutdown or accidental power outage.

Cisco VIM supports the following control node recovery command:

```
# ./ciscovimclient/ciscovim partition-recovery
```

The control node will recover after the network partition is resolved.

To make sure Nova services are good across compute nodes, execute the following command:

```
# source /root/openstack-configs/openrc
# nova service-list
```

To check for the overall cloud status, execute the following:

```
#  cd installer-<tagid>/tools
# ./cloud_sanity.py -c all
```

In case of a complete pod outage, you must follow a sequence of steps to bring the pod back. The first step is to bring up the management node, and check that the management node containers are up (and not in exited state) using **docker ps –a** command. After you bring up the management node, bring up all the other pod nodes. Make sure every node is reachable through password-less SSH from the management node. Verify that no network IP changes have occurred. You can get the node SSH IP access information from /root/openstack-config/mercury_servers_info.

Execute the following command sequence:

- Check the setup_data.yaml file and runtime consistency on the management node:

  ```
  # cd /root/installer-<tagid>/tools
  # ./ciscovimclient/ciscovim run --perform 1,3 -y
  ```

- Execute the cloud sanity command:

  ```
  # cd/root/installer-<tagid>/tools
  # ./cloud_sanity.py -c all
  ```

- Check the status of the REST API server and the corresponding directory where it is running:

  ```
  # cd/root/installer-<tagid>/tools
  #./restapi.py -a status
  Status of the REST API Server:  active (running) since Thu 2016-08-18 09:15:39 UTC; 9h
   ago
  ```

```
REST API launch directory: /root/installer-<tagid>/
```

- If the REST API server is not running from the right installer directory, execute the following to get it running from the correct directory:

```
# cd/root/installer-<tagid>/tools
#./restapi.py -a setup

Check if the REST API server is running from the correct target directory
#./restapi.py -a status
Status of the REST API Server:  active (running) since Thu 2016-08-18 09:15:39 UTC; 9h
 ago
REST API launch directory: /root/new-installer-<tagid>/
```

- Verify Nova services are good across the compute nodes, execute the following command:

```
# source /root/openstack-configs/openrc
# nova service-list
```

If cloud-sanity fails, execute cluster-recovery (./ciscovimclient/ciscovim cluster-recovery), then re-execute the cloud-sanity and nova service-list steps as listed above.

Recovery of compute and OSD nodes requires network connectivity and reboot so that they can be accessed using SSH without password from the management node. To shutdown, bring the pod down in the following sequence:

1. Shut down all VMs, then all the compute nodes.
2. Shut down all storage nodes serially.
3. Shut down all controllers one at a time.
4. Shut down the management node.
5. Shut down the networking gears.

   Bring the nodes up in reverse order, that is, start with networking gears, then the management node, storage nodes, control nodes, and compute nodes. Make sure each node type is completely booted up before you move on to the next node type. Then validate the Cisco API server.

   ./ciscovimclient/ciscovim run --perform 1,3 -y

After the pod is up, run the partition-recovery playbook (./ciscovimclient/ciscovim partition-recovery) on the management node, which will determine the correct order to bring up the Galera cluster by commit ID and start the remaining downed services.

Validate if all the VMs are up (not in shutdown state). If any of the VMs are in down state, bring them UP using the horizon dashboard.

Managing Nova Compute Scheduler Filters and User Data

OpenStack Nova is an OpenStack component that provides on-demand access to compute resources by provisioning large networks of virtual machines (VMs). In addition to the standard Nova filters, Cisco VIM supports the following additional scheduler filters:

- ServerGroupAffinityFilter—Ensures that an instance is scheduled onto a host from a set of group hosts. To use this filter, you must create a server group with an affinity policy and pass a scheduler hint using

group as the key and the server group UUID as the value. Use the **nova** command-line tool and the **--hint** flag. For example:

```
$ nova server-group-create --policy affinity group-1
$ nova boot --image IMAGE_ID --flavor 1 --hint group=SERVER_GROUP_UUID server-1
```

- ServerGroupAntiAffinityFilter—Ensures that each group instance is on a different host. To use this filter, you must create a server group with an anti-affinity policy and pass a scheduler hint, using group as the key and the server group UUID as the value. Use the **nova** command-line tool and the **--hint** flag. For example:

```
$ nova server-group-create --policy anti-affinity group-1
$ nova boot --image IMAGE_ID --flavor 1 --hint group=SERVER_GROUP_UUID server-1
```

- SameHostFilter—Within an instance set, schedules one instance on the same host as another instance. To use this filter, pass a scheduler hint using **same_host** as the key and a list of instance UUIDs as the value. Use the **nova** command-line tool and the **--hint** flag. For example:

```
$ nova boot --image IMAGE_ID --flavor 1 --hint same_host=INSTANCE_ID server-1
```

- DifferentHostFilter—Within an instance set, schedules one instance on a different host than another instance. To use this filter, pass a scheduler hint using **different_host** as the key and a list of instance UUIDs as the value. The filter is the opposite of SameHostFilter. Use the **nova**command-line tool and the **--hint** flag. For example:

```
$ nova boot --image IMAGE_ID --flavor 1 --hint different_host=INSTANCE_ID server-1
```

In addition to scheduler filters, you can set up user data files for cloud application initializations. A user data file is a special key in the metadata service that holds a file that cloud-aware applications in the guest instance can access. For example, one application that uses user data is the cloud-init system, an open-source package that is available on various Linux distributions. The cloud-init system handles early cloud instance initializations. The typical use case is to pass a shell script or a configuration file as user data during the Nova boot, for example:

```
$ nova boot --image IMAGE_ID --flavor 1 --hint user-data FILE_LOC server-1
```

Utilizing NUMA in Cisco NFV Infrastructure

Non-uniform memory access (NUMA) is a method of configuring microprocessor clusters in a multiprocessing system so that they can share memory locally, improving performance, and future system expansion. NUMA is used in symmetric multiprocessing (SMP) systems.

Cisco VIM supports the nova scheduler ability to choose the hypervisor to spawn VMs based on NUMA topologies. If the instances do not have NUMA topologies specified, the filter places no constraints on the compute hosts selection. NUMA allows applications, such as NFV applications, to benefit from CPU pinning and huge pages supported for better performance.

To utilize NUMA, you must define and maintain a list of compute hosts to serve as the NFV hosts. You can configure this list in the setup_data.yaml NFV_HOSTS field. After Cisco VIM is deployed, you must create the host aggregation and NFV flavors that contain the compute hosts where the NFV instances will be scheduled.

By default, CPU has two cores to run host-level general purpose tasks; no additional cores are isolated for host level tasks. The huge page size of Cisco VIM is two MB.

1. While most of the NUMA configurations are enabled on day-0, you can roll-in the disable hyperthreading feature on a newly added compute node. To disable the hyper-threading on a new compute node, update the setup_data.yaml file with the following option and then initiate the addition of the compute nodes:

```
DISABLE_HYPERTHREADING: True or False; this is optional and
default value is false.
```

Note You can not revert the disable hyperthreading feature as the disable-hyperthreading is a non-reversible option in the setup_data.yaml file.

2. To disable hyper-threading, on existing computes on day-n, the administrator should remove the existing compute nodes and add new computes.

Note We recommend you to not run a pod in an hybrid-mode; that is, a sub-set of compute nodes have hyper-threading disabled in them. The administrator has to remove the existing compute nodes and add them as new compute nodes, so that the hyper-threading feature is disabled in all the compute nodes in a pod.

Monitoring Cisco NFVI Health with CloudPulse

You can query the state of various Cisco NFVI OpenStack end points using CloudPulse, an OpenStack health-checking tool. By default, the tool automatically polls OpenStack Cinder, Glance, Nova, Neutron, Keystone, Rabbit, Mariadb, and Ceph every four minutes. However, you can use a CLI REST API call from the management node to get the status of these services in real time. Also you can integrate the CloudPulse API into your applications and get the health of the OpenStack services on demand. You can find additional information about using CloudPulse in the following OpenStack sites:

- https://wiki.openstack.org/wiki/Cloudpulse
- https://wiki.openstack.org/wiki/Cloudpulseclient
- https://wiki.openstack.org/wiki/Cloudpulse/DeveloperNotes
- https://wiki.openstack.org/wiki/Cloudpulse/OperatorTests
- https://wiki.openstack.org/wiki/Cloudpulse/APIDocs

CloudPulse has two set of tests: endpoint_scenario (runs as a cron or manually) and operator test (run manually). The supported Cloudpulse tests groups include:

- nova_endpoint
- neutron_endpoint
- keystone_endpoint
- glance_endpoint
- cinder_endpoint

Operator tests include:

- ceph_check—Executes the command, "ceph -f json status" on the Ceph-mon nodes and parses the output. If the result of the output is not “HEALTH_OK” ceph_check will report an error.

- docker_check—Finds out if all the Docker containers are in the running state in all the nodes. It will report an error if any containers are in the Exited state. It runs the command “docker ps -aq --filter 'status=exited'”.
- galera_check—Executes the command, "mysql 'SHOW STATUS;” on the controller nodes and displays the status.
- node_check—Checks if all the nodes in the system are up and online. It also compares the result of “nova hypervisor list” and finds out if all the computes are available.
- rabbitmq_check—Runs the command, “rabbitmqctl cluster_status” on the controller nodes and finds out if the rabbitmq cluster is in quorum. If nodes are offline in the cluster rabbitmq_check will report as failed.

CloudPulse servers are installed in containers on all control nodes. The CloudPulse client is installed on the management node by the Cisco VIM installer. To execute CloudPulse, source the openrc file in the openstack-configs directory and execute the following:

```
[root@MercRegTB1 openstack-configs]# cloudpulse --help
usage: cloudpulse [--version] [--debug] [--os-cache]
                  [--os-region-name <region-name>]
                  [--os-tenant-id <auth-tenant-id>]
                  [--service-type <service-type>]
                  [--endpoint-type <endpoint-type>]
                  [--cloudpulse-api-version <cloudpulse-api-ver>]
                  [--os-cacert <ca-certificate>] [--insecure]
                  [--bypass-url <bypass-url>] [--os-auth-system <auth-system>]
                  [--os-username <username>] [--os-password <password>]
                  [--os-tenant-name <tenant-name>] [--os-token <token>]
                  [--os-auth-url <auth-url>]
                  <subcommand> ...
```

To check results of periodic CloudPulse runs:

```
# cd /root/openstack-configs
# source openrc
# cloudpulse result
+--------------------------------------+------+-------------------+----------+---------+
| uuid                                 | id   | name              | testtype | state   |
+--------------------------------------+------+-------------------+----------+---------+
| bf7fac70-7e46-4577-b339-b1535b6237e8 | 3788 | glance_endpoint   | periodic | success |
| 1f575ad6-0679-4e5d-bc15-952bade09f19 | 3791 | nova_endpoint     | periodic | success |
| 765083d0-e000-4146-8235-ca106fa89864 | 3794 | neutron_endpoint  | periodic | success |
| c1c8e3ea-29bf-4fa8-91dd-c13a31042114 | 3797 | cinder_endpoint   | periodic | success |
| 04b0cb48-16a3-40d3-aa18-582b8d25e105 | 3800 | keystone_endpoint | periodic | success |
| db42185f-12d9-47ff-b2f9-4337744bf7e5 | 3803 | glance_endpoint   | periodic | success |
| 90aa9e7c-99ea-4410-8516-1c08beb4144e | 3806 | nova_endpoint     | periodic | success |
| d393a959-c727-4b5e-9893-e229efb88893 | 3809 | neutron_endpoint  | periodic | success |
| 50c31b57-d4e6-4cf1-a461-8228fa7a9be1 | 3812 | cinder_endpoint   | periodic | success |
| d1245146-2683-40da-b0e6-dbf56e5f4379 | 3815 | keystone_endpoint | periodic | success |
| ce8b9165-5f26-4610-963c-3ff12062a10a | 3818 | glance_endpoint   | periodic | success |
| 6a727168-8d47-4a1d-8aa0-65b942898214 | 3821 | nova_endpoint     | periodic | success |
| 6fbf48ad-d97f-4a41-be39-e04668a328fd | 3824 | neutron_endpoint  | periodic | success |
+--------------------------------------+------+-------------------+----------+---------+
```

To view all CloudPulse tests:

```
# cd /root/openstack-configs
# source openrc
# cloudpulse test-list
```

To run a CloudPulse test on demand:

```
# cd /root/openstack-configs
# source openrc
# cloudpulse run --name <test_name>
# cloudpulse run  --all-tests
# cloudpulse run --all-endpoint-tests
# cloudpulse run --all-operator-tests
```

To run a specific CloudPulse test on demand:

```
# cloudpulse run --name neutron_endpoint
+------------+--------------------------------------+
| Property   | Value                                |
+------------+--------------------------------------+
| name       | neutron_endpoint                     |
| created_at | 2016-03-29T02:20:16.840581+00:00     |
| updated_at | None                                 |
| state      | scheduled                            |
| result     | NotYetRun                            |
| testtype   | manual                               |
| id         | 3827                                 |
| uuid       | 5cc39fa8-826c-4a91-9514-6c6de050e503 |
+------------+--------------------------------------+
```

To show detailed results of a specific CloudPulse run:

```
#cloudpulse show 5cc39fa8-826c-4a91-9514-6c6de050e503
+------------+--------------------------------------+
| Property   | Value                                |
+------------+--------------------------------------+
| name       | neutron_endpoint                     |
| created_at | 2016-03-29T02:20:16+00:00            |
| updated_at | 2016-03-29T02:20:41+00:00            |
| state      | success                              |
| result     | success                              |
| testtype   | manual                               |
| id         | 3827                                 |
| uuid       | 5cc39fa8-826c-4a91-9514-6c6de050e503 |
+------------+--------------------------------------+
```

To see the CloudPulse options, source the openrc file in openstack-configs dir and execute:

```
#cloudpulse --help
```

The cloudpulse project has a RESTful Http service called the Openstack Health API. Through this API cloudpulse allows the user to list the cloudpulse tests, create new cloudpulse tests and see the results of the cloudpulse results.

All API calls described in this documentation require keystone authentication. We can use the keystone v2 or v3 version for the authentication. The corresponding configuration should be configured properly in the cloudpulse config in order that the cloudpulse can reach the v2 or the v3 keystone API

The Identity service generates authentication tokens that permit access to the Cloudpulse REST APIs. Clients obtain this token and the URL endpoints for other service APIs by supplying their valid credentials to the authentication service. Each time you make a REST API request to Cloudpulse, you supply your authentication token in the X-Auth-Token request header.

Service Catalog URL

The OpenStack Keystone service catalog allows API clients to dynamically discover and navigate to cloud services. Cloudpulse has its own service URL which is added to the keystone service catalog. Send a token request to Keystone to find the service URL of cloudpulse . The token request will list all the catalog of services available.

Get Token from Keystone

To get the token from keystone we have to use the following request:

Resource URI

Verb	URI
POST	http://<controller_lb_ip>:5000/v2.0/tokens

Example

```
JSON Request
POST / v2.0/tokens
Accept: application/json
{
    "auth": {
        "passwordCredentials":{
                  "username": "admin",
                  "password": "iVP1YciVKoMGId1O"
        }
    }
}

JSON Response
200 OK
Content-Type: application/json
{
  "access": {
    "token": {
      "issued_at": "2017-03-29T09:54:01.000000Z",
      "expires": "2017-03-29T10:54:01Z",
      "id":
"gAAAAABY24Q5TDIqizuGmhOXakV2rIzSvSPQpMAmC7SA2UzUXZQXSH-ME98d3Fp4Fsj16G561a420B4BK0fylcykL22EcO9",
……
……..
}
```

Get Service Catalog URL for Cloudpulse

Resource URI

Verb	URI
GET	http://<controller_ip>:35357/v2.0/endpoints

Example

```
JSON Request
GET /v2.0/endpoints
Accept: application/json

JSON Response
200 OK
Content-Type: application/json
{"endpoints": [
     {"internalurl": "http://<controller>:9999",
      "adminurl": "http://<controller>:9999",
      "publicurl":"http://<controller>:9999"
}]}
}
```

Cloudpulse API's

The following are a list of API's and the corresponding functions that the API performs. The cloudpulse API's should always be accessed with the X-Auth-Token which contains the token which is received from the keystone token generation API mentioned i the preceding section.

List of Cloudpulse Tests

To get the list of cloudpulse tests:

Resource URI

Verb	URI
GET	http://<controller_ip>:9999/cpulse

Example

```
JSON Request
GET /cpulse
Accept: application/json

JSON Response
200 OK
Content-Type: application/json
{
  "cpulses": [
    {
      "name": "galera_check",
      "state": "success",
      "result":"ActiveNodes:16.0.0.37,16.0.0.17,16.0.0.27",
      "testtype": "periodic",
      "id": 4122,
      "uuid": "a1b52d0a-ca72-448a-8cc0-5bf210438d89"
    }]
}
```

Get detailed result of 1 test

To get detailed result of the test.

Resource URI

Verb	URI
GET	http://<controller_ip>:9999/cpulse/<uuid>

Uuid : uuid of the test

Example

```
JSON Request
GET /cpulse/e6d4de91-8311-4343-973b-c507d8806e94
Accept: application/json

JSON Response
200 OK
Content-Type: application/json
{
      "name": "galera_check",
      "state": "success",
      "result":"ActiveNodes:16.0.0.37,16.0.0.17,16.0.0.27",
      "testtype": "periodic",
      "id": 4122,
      "uuid": " e6d4de91-8311-4343-973b-c507d8806e94"
}
```

Get List of Tests Available

To get a list of available cloudpulse tests:

Resource URI

Verb	URI
GET	http://<controller_ip>:9999/cpulse/list_tests

Example

```
JSON Request
GET /cpulse/list_tests
Accept: application/json

JSON Response
200 OK
Content-Type: application/json
{
  "endpoint_scenario":
"all_endpoint_tests\ncinder_endpoint\nglance_endpoint\nkeystone_endpoint\nneutron_endpoint\nnova_endpoint",

  "operator_scenario":
"all_operator_tests\nceph_check\ndocker_check\ngalera_check\nnode_check\nrabbitmq_check"
}
```

Schedule a manual cloudpulse test:

To schedule a manual test of cloudpulse

Resource URI

Verb	URI
POST	http://<controller_ip>:9999/cpulse

Example

```
JSON Request
POST /cpulse
Accept: application/json
{
"name": "galera_check"
}

JSON Response
200 OK
Content-Type: application/json
{
      "name": "galera_check",
      "state": "scheduled",
      "result":"NotYetRun",
      "testtype": "manual",
      "id": 4122,
      "uuid": " e6d4de91-8311-4343-973b-c507d8806e94"
}
```

-

Remove the results of a test

To remove the results of a test.

Resource URI

Verb	URI
DELETE	http://<controller_ip>:9999/cpulse/<uuid>

Uuid : uuid of the test

Example

```
JSON Request
DELETE /cpulse/68ffaae3-9274-46fd-b52f-ba2d039c8654
Accept: application/json

JSON Response
204 No Content
```

Checking Network Connections

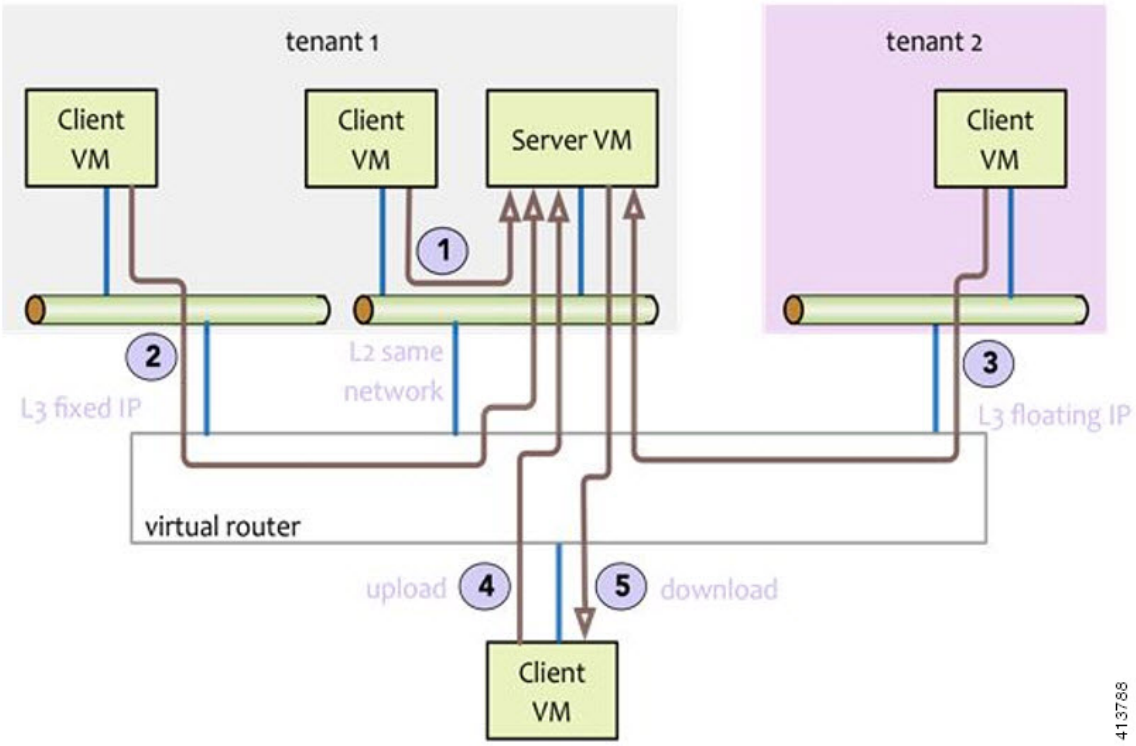
You can use Virtual Machine ThroughPut (VMTP) to check Layer 2 and Layer 3 data plane traffic between Cisco NFVI compute nodes. VMTP performs ping connectivity, round trip time measurement (latency), and TCP/UDP throughput measurement for the following Cisco NFVI east to west VM-to-VM flows:

- Same network (private fixed IP, flow number 1).
- Different network using fixed IP (same as intra-tenant L3 fixed IP, flow number 2).
- Different network using floating IP and NAT (same as floating IP inter-tenant L3, flow number 3.)
- When an external Linux host is available for testing north to south flows, external host to VM download and upload throughput and latency (L3/floating IP, flow numbers 4 and 5).

The following figure shows the traffic flows VMTP measures. Cloud traffic flows are checked during Cisco VIM installation and can be checked at any later time by entering the following command:

```
$ ./ciscovimclient/ciscovim run --perform 8 -y
```

Figure 1: VMTP Cloud Traffic Monitoring



Enabling NFVBench Post Deployment

NFVBench is a data plane performance benchmark tool for NFVI that can be optionally installed after the pod deployment.

NFVBench is used to:

- Verify that the data plane is working properly and efficiently when using well defined packet paths that are typical of NFV service chains.
- Measure the actual performance of your data plane so that you can estimate what VNFs can expect from the infrastructure when it comes to receiving and sending packets.

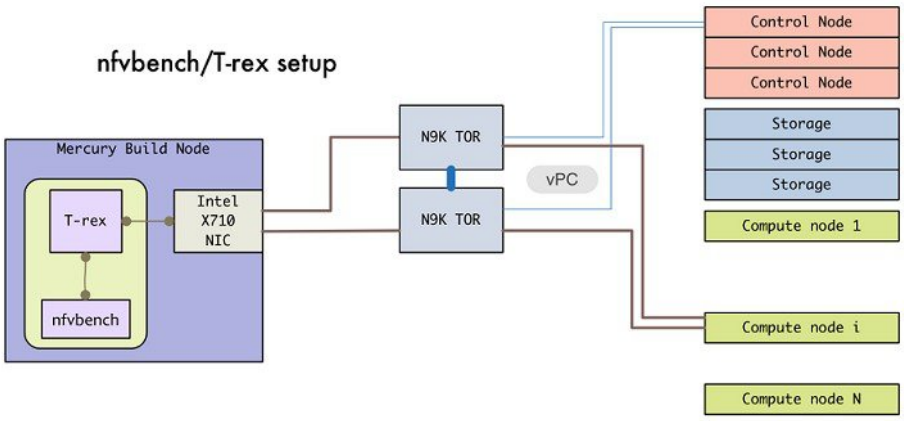
While VMTP only measures VM to VM traffic, NFVBench measures traffic flowing from an integrated software traffic generator (TRex) running on the management node to the ToR switches to test VMs running in compute nodes.

In Cisco VIM, the NFVBench (performance benchmark) is an optional tool. You can deploy NFVBench after the installation of the pod.

Before you begin

- A 10GE Intel NIC (Intel X710 NIC (4 x 10G)) must be installed on a management node.
- A TRex traffic generator which uses DPDK interface to interact with Intel NIC and makes use of hardware, instead of software to generate packets. This approach is more scalable and enables NFVBench to perform tests without software limitations.
- Wire two physical interfaces of the Intel NIC to the TOR switches (as shown in the following figure).

Figure 2: NFVBench topology setup



Step 1 Enable the NFVBench configuration in the setup_data.yaml file.

Sample configuration files for OVS/VLAN or VPP mechanism driver:

```
NFVBENCH:
   enabled: True    # True or False
   tor_info: {TORa: eth1/42, TORb: eth1/42} # mandatory
#  tor_info: {TOR: 'eth1/42,eth1/43'} # use if there is only one TOR switch
#  nic_ports: 3,4    # Optional input, indicates which 2 of the 4 available ports
                     # of 10G Intel NIC on the management node is NFVbench tool using
                     # to send and receive traffic.
                     # Defaults to the first 2 ports of NIC (ports 1 and 2) if not specified.
                     # Port number must be between 1 and 4, one port cannot be used twice.
                     # Example:
                     # nic_ports: 1,4    # the first and the last port of Intel NIC are used
```

Sample configuration for VTS mechanism driver:

```
NFVBENCH:
   enabled: True    # True or False
   tor_info: {TORa: eth1/42, TORb: eth1/42} # mandatory
   vtep_vlans: 1500,1501    # Mandatory and needed only for VTS/VXLAN.
                            # Specify any pair of unused VLAN ids to be used
```

```
                              # for VLAN to VxLAN encapsulation in TOR switch.
#  tor_info: {TOR: 'eth1/42,eth1/43'} # Use if there is only one TOR switch.
#  nic_ports: 3,4    # Optional input, indicates which 2 of the 4 available ports
                     # of 10G Intel NIC on the management node is NFVbench tool using
                     # to send and receive traffic.
                     # Defaults to the first 2 ports of NIC (ports 1 and 2) if not specified.
                     # Port number must be between 1 and 4, one port cannot be used twice.
                     # Example:
                     # nic_ports: 1,4    # the first and the last port of Intel NIC are used

VTS_PARAMETERS:
   …
   VTS_NCS_IP: '11.11.11.111'   # '<vts_ncs_ip>', mandatory when VTS enabled
   VTC_SSH_USERNAME: 'admin'   # '<vtc_ssh_username>', mandatory for NFVbench
   VTC_SSH_PASSWORD: 'my_password'   # '<vtc_ssh_password>', mandatory for NFVbench
```

Step 2 Configuring minimal settings of NFVBench:

```
# Minimal settings required for NFVbench
TORSWITCHINFO:
   CONFIGURE_TORS: <True or False> # True if switches should be configured to support NFVbench
   …
   SWITCHDETAILS:
   - hostname: 'TORa'   # Hostname matching 'tor_info' switch name.
     username: 'admin'   # Login username for switch user.
     password: 'my_password'  # Login password for switch user.
     ssh_ip: '172.31.230.123' # SSH IP for switch.
   - hostname: 'TORb'
     username: 'admin'
     password: 'my_password'
     ssh_ip: '172.31.230.124'
```

TOR switches will be configured based on information provided in tor_info. Two ports specified by interfaces are configured in trunk mode. In order to access them and retrieve TX/RX counters you need the Login details for TOR switches. It is not required to set 'CONFIGURE_TORS' to 'True', but then manual configuration is necessary.

With VTS as mechanism driver additional settings are needed. NFVBench needs access to VTS NCS to perform cleanup after it detaches the traffic generator port from VTS. Also a pair of VTEP VLANs is required for VLAN to VxLAN mapping. Value can be any pair of unused VLAN ID.

Step 3 Reconfigure Cisco VIM to create a NFVBench container. To reconfigure add necessary configuration to the setup_data.yaml file, run the reconfigure command as follows.

```
[root@mgmt1 ~]# cd /root/
[root@mgmt1 ~]# mkdir MyDir
[root@mgmt1 ~]# cp /root/openstack-configs/setup_data.yaml /root/MyDir/
[root@mgmt1 ~]# cd /root/
[root@mgmt1 ~]# # update the setup_data to include NFVBENCH section
[root@mgmt1 ~]# cd /root/MyDir/
[root@mgmt1 ~]# vi setup_data.yaml
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim --setupfile /root/MyDir/setup_data.yaml reconfigure
```

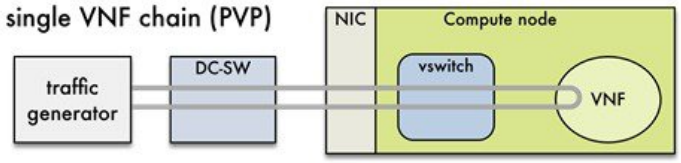
After reconfiguration is done, you can see NFVBench container up and is ready to use.

NFVBench Usage

Built-in packet paths

NFVBench can setup and stage three different packet paths.

The default packet path is called **PVP** (Physical - VM - Physical) and represents a typical service chain made of 1 VNF/VM:



The traffic generator runs inside the NFVBench container on the management node. DC-SW represents the top of rack switch(es). The VNF is a test VM that contains a fast L3 router based on FD.io VPP. This VNF image can also be configured to run an L2 forwarder based on DPDK testpmd (both options generally yield roughly similar throughput results).

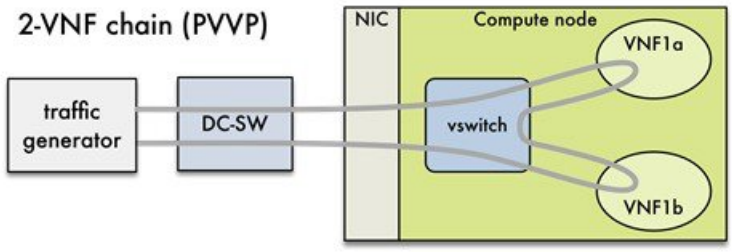
Traffic is made of UDP packets generated on the 2 physical interfaces (making it a bi-directional traffic). Packets are forwarded by the switch to the appropriate compute node before arriving to the virtual switch, then to the VNF before looping back to the traffic generator on the other interface. Proper stitching of the traffic on the switch is performed by NFVbench by using the appropriate mechanism (VLAN tagging for VLAN based deployments, VxLAN VTEP in the case of VTS deployments).

The performance of the PVP packet path provides a very good indication of the capabilities and efficiency of the NFVi data plane in the case of a single service chain made of 1 VNF/VM.

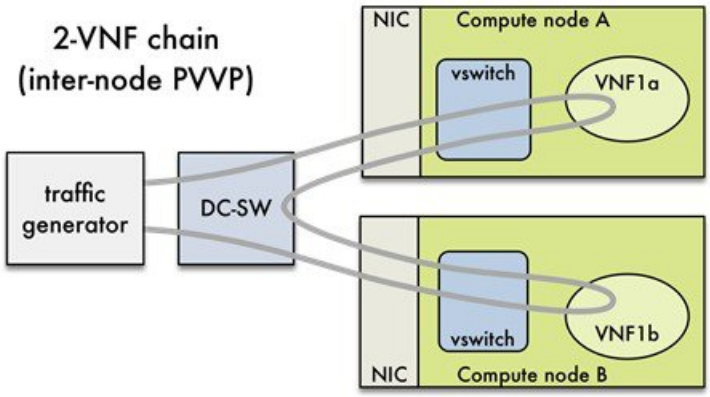
NFVBench also supports more complex service chains made of 2 VM in sequence and called PVVP (Physical-VM-VM-Physical).

In a PVVP packet path, the 2 VMs can reside on the same compute node (PVVP intra-node) or on different compute nodes (PVVP inter-node).

PVVP intra-node is more efficient when a virtual switch is used as packets do not have to go through the switch between the 2 VMs:



PVVP inter-node requires packets to go through the switch and back between the 2 VMs.



NFVBench Command Line Options

The common NFVBench command line options can be displayed using the --help option:

```
[root@mgmt1 ~]# nfvbench --help
```

Control Plane Verification

If you are trying NFVBench for the first time, verify that the tool can stage the default packet path properly without sending any traffic.

The --no-traffic option will exercise the control plane by creating a single test service chain with one VM but does not send any traffic.

The following command will only stage the default PVP packet path (but will not generate any traffic):

```
[root@mgmt1 ~]# nfvbench --no-traffic
```

Fixed Rate Run Test

The data plane traffic test is to generate traffic at a fixed rate for a fixed duration. For example, to generate a total of 10,000 packets per second (which is 5,000 packets per second per direction) for the default duration (60 seconds) and using the default frame size of 64 bytes::

```
[root@mgmt1 ~]# nfvbench --help
```

Packet Sizes

You can specify any list of frame sizes using the -frame-size option (pass as many as desired), including IMIX.

Following is an example, to run a fixed rate with IMIX and 1518 byte frames:

```
[root@mgmt1 ~]# nfvbench --rate 10kpps -frame-size IMIX -frame-size 1518
```

NDR and PDR Test

NDR and PDR test is used to determine the performance of the data plane in terms of throughput at given drop rate.

- No Drop Rate(NDR)-- It is the highest throughput achieved while allowing zero packet drop. (allows a very low drop rate usually lesser than 0.001%).
- Partial Drop Rate (PDR)--It is the highest throughput achieved while allowing most at a given drop rate (typically less than 0.1%).

NDR is always less or equal to PDR.

To calculate the NDR and PDR for your pod run the following command:

```
[root@mgmt1 ~]# nfvbench --rate ndr_pdr
```

Multi-chain Test

In multi-chain test, each chain represents an independent packet path symbolizing real VNF chain. You can run multiple concurrent chains and better simulate network conditions in real production environment. Results with single chain versus with multiple chains usually vary because of services competing for resources (RAM, CPU, and network).

To stage and measure multiple service chains at the same time, use *--service-chain-count* flag or shorter *-scc* version.

The following example shows how to run the fixed rate run test with ten PVP chains:

```
[root@mgmt1 ~]# nfvbench -scc 10 --rate 100kpps
```

The following example shows how to run the NDR/PDR test with ten PVP chains:

```
[root@mgmt1 ~]# nfvbench -scc 10 --rate ndr_pdr
```

Multi-flow Test

One flow is defined by a source and destination MAC/IP/port tuple in the generated packets. It is possible to have many flows per chain. The maximum number of flows supported is in the order of 1 million flows per direction.

The following command will run three chains with a total of 100K flows per direction (for all chains):

```
[root@mgmt1 ~]# nfvbench -scc 3 -fc 100k
```

External Chain Test

NFVBench measures the performance of chains that are pre-staged (using any means external to NFVBench). Such chains can be real VNFs with L3 routing capabilities or L2 forwarding chains.

This test is used when user wants to use NFVBench for only traffic generation. In this case, NFVBench sends traffic from traffic generator and reports results without performing any configuration.

You must do necessary configurations such as creating networks and VMs with configuration that allows generated traffic to pass. NFVBench needs to know the 2 edge networks to which the traffic generator will be attached.

If the external chains only support L2 forwarding, the NFVBench configuration must:

- Enable VLAN tagging and define the VLAN IDs to use - if applicable (or disable vlan tagging if it is not needed).
- The destination MAC to use in each direction (this will depend on the L2 forwarding mode in place in the service chain).

If the external chains support IPv4 routing, the NFVBench configuration must:

- Define the public IP addresses of the service chain end points (gateway IP) that will be used to discover destination MAC using ARP.
- Set the vlan tagging appropriately.

To measure performance for external chains, use the *--service-chain EXT* (or *-sc EXT*) option:

```
[root@mgmt1 ~]# nfvbench -sc EXT
```

Note NFVBench will not access ToR switches or v-switch in compute node.

NFVBench Result Generation and Storage

NFVBench detailed results can be stored in JSON format if passed the --json option with a destination file name or the --std-json option with a destination folder pathname (if you want to use a standard file name generated by NFVBench). It is also possible to use both methods to generate the output in to two different files at the same time:

```
[root@mgmt1 ~]# nfvbench -scc 3 -fc 10 -fs 64 --json /tmp/nfvbench/my.json --std-json
/tmp/nfvbench
```

The above command will create two JSON files in /tmp/nfvbench container directory, which is mapped to the host directory. The first file will be named my.json.

With the--std-jsonoption, the standard NFVBench filename format follows this pattern:

<service-chain-type>-<service-chain-count>-<flow-count>-<frame-sizes>.json

Default chain is PVP and flag *-fs* was used to override traffic profile in configuration file. With 3 chains and 10 flows specified, file *PVP-3-10-64.json* was created

Interpretation of Results

NFVBench prints data to command line in a table form. The data includes the current configuration, test devices details, and results computed based on traffic statistics.

Fixed Rate

Run the following command on NFVBench to view the traffic generated at fixed rate at different components of the packet path:

```
[root@mgmt1 ~]# nfvbench --rate 5kpps -fs IMIX
```

NFVBench summary consists of multiple blocks. In some cases, NFVBench displays lesser data. For example, in the output, the EXT chain does not access some path components (like switch). Therefore, the summary does not display

```
========== NFVBench Summary ==========
Date: 2017-03-28 19:59:53
NFVBench version 0.3.5
Openstack Neutron:
  vSwitch: VTS
  Encapsulation: VxLAN
Benchmarks:
> Networks:
  > Components:
```

```
    > TOR:
        Type: N9K
        Version:
          10.28.108.249:
            BIOS: 07.34
            NXOS: 7.0(3)I2(2b)
          10.28.108.248:
            BIOS: 07.34
            NXOS: 7.0(3)I2(2b)
    > VTC:
        Version:
          build_date: 2017-03-03-05-41
          build_number: 14
          git_revision: 0983910
          vts_version: 2.3.0.40
          git_branch: vts231newton
          job_name: vts231newton_gerrit_nightly
    > Traffic Generator:
        Profile: trex-local
        Tool: TRex≠
        Version:
          build_date: Feb 16 2017
          version: v2.18
          built_by: hhaim
          build_time: 18:59:02
  > Service chain:
    > PVP:
      > Traffic:
          VPP version:
            sjc04-pod3-compute-4: v17.04-rc0~98-g8bf68e8
          Profile: custom_traffic_profile
          Bidirectional: True
          Flow count: 1
          Service chains count: 1
          Compute nodes: [u'nova:sjc04-pod3-compute-4']

            Run Summary:

+----------------+------------+---------------------+----------------------+----------------------+
              |  L2 Frame Size  |  Drop Rate  |   Avg Latency (usec) |   Min Latency (usec)
 |   Max Latency (usec) |
+================+============+=====================+======================+======================+
              |      IMIX       |   0.0000%   |                16.50 |                10.00
 |              241.00 |
+----------------+------------+---------------------+----------------------+----------------------+


            L2 frame size: IMIX
            Chain analysis duration: 73 seconds

            Run Config:

              +-------------+------------------+--------------+-----------+
              |  Direction  |   Duration (sec) |     Rate     |   Rate    |
              +=============+==================+==============+===========+
              |   Forward   |               60 | 7.6367 Mbps  | 2,500 pps |
              +-------------+------------------+--------------+-----------+
              |   Reverse   |               60 | 7.6367 Mbps  | 2,500 pps |
              +-------------+------------------+--------------+-----------+
```

```
                |    Total    |               60 | 15.2733 Mbps | 5,000 pps |
                +-------------+------------------+--------------+-----------+

             Chain Analysis:

+-----------------+--------+-------------+-----------+-------------+-------------+-----------+-----------+
|      Interface       | Device | Packets (fwd) | Drops (fwd) | Drop% (fwd) | Packets (rev) | Drops (rev) | Drop% (rev) |
+=================+========+=============+===========+=============+=============+===========+===========+
|  traffic-generator   |  trex  |     150,042     |             |     150,042     |       0       |  0.0000%    |
+-----------------+--------+-------------+-----------+-------------+-------------+-----------+-----------+
|       vni-5098       |  n9k   |     150,042     |       0       |  0.0000%    |     150,042     |       0       |  0.0000%    |
+-----------------+--------+-------------+-----------+-------------+-------------+-----------+-----------+
|    vxlan_tunnel0     |  vpp   |     150,042     |       0       |  0.0000%    |     150,042     |       0       |  0.0000%    |
+-----------------+--------+-------------+-----------+-------------+-------------+-----------+-----------+
| VirtualEthernet0/0/1 |  vpp   |     150,042     |       0       |  0.0000%    |     150,042     |       0       |  0.0000%    |
+-----------------+--------+-------------+-----------+-------------+-------------+-----------+-----------+
| VirtualEthernet0/0/0 |  vpp   |     150,042     |       0       |  0.0000%    |     150,042     |       0       |  0.0000%    |
+-----------------+--------+-------------+-----------+-------------+-------------+-----------+-----------+
|    vxlan_tunnel1     |  vpp   |     150,042     |       0       |  0.0000%    |     150,042     |       0       |  0.0000%    |
+-----------------+--------+-------------+-----------+-------------+-------------+-----------+-----------+
|       vni-5099       |  n9k   |     150,042     |       0       |  0.0000%    |     150,042     |       0       |  0.0000%    |
+-----------------+--------+-------------+-----------+-------------+-------------+-----------+-----------+
|  traffic-generator   |  trex  |     150,042     |               |             |     150,042     |       0       |  0.0000%    |
+-----------------+--------+-------------+-----------+-------------+-------------+-----------+-----------+

Run as:
nfvbench -c /tmp/nfvbench/nfvbench.cfg --rate 5kpps -fs IMIX
```

Summary Interpretation:

Lines 1-34: General information about host system and used components.

Lines 35-45: Test-specific information about service chain, traffic profile, and compute nodes.

Lines 46-53: Summary of traffic profile run with results. A new row is added in the table for every packet size in test. The output displays the run summary for the IMIX packet size, but lines for 64B and 1518B can also be present as well. Table contains following columns:

The run summary table includes the following columns:

- Drop Rate: The percentage of total drop rate for all chains and flows from the total traffic sent.
- Avg Latency: Average latency of average chain latencies in microseconds.
- Min Latency: Minimum latency of all chain latencies in microseconds.
- Max Latency: Maximum latency of all chain latencies in microseconds.

Lines 54-68: Length of test in seconds for given packet size and table displaying rate and duration which was used for running traffic in the test.

Lines 69-89: Detailed analysis of test. Each row represents one interface on packet path in order they are visited. Left side of the table is for forward direction (from traffic generator port 0 to port 1), right side is for reverse direction (from traffic generator port 1 to port 0).

The chain analysis table has following columns:

- Interface: Interface name on devices in packet path.
- Device: Device name on which the interface is displayed in the first column is available.
- Packets (fwd): RX counter on given interface, only the first row is TX counter (it is the beginning of packet path).
- Drops (fwd): Amount of packets being dropped between current and previous interface.
- Drop% (fwd): Percentage of dropped packets on this interface to the total packet drops.
- Packets (rev): Similar as Packets (fwd) but for reverse direction.
- Drops (rev): Similar as Drops (fwd) but for reverse direction.
- Drop% (rev): Similar as Drop% (fwd) but for reverse direction.

This type of summary is very useful for finding bottlenecks or to verify if the system can handle certain fixed rate of traffic.

NDR/PDR

The test result shows throughput values in different units for both NDR and PDR with latency statistics (minimum, maximum, average) for each test.

```
[root@mgmt1 ~]# nfvbench --rate ndr_pdr -fs IMIX

========== NFVBench Summary ==========
Date: 2017-03-28 20:20:46
NFVBench version 0.3.5
Openstack Neutron:
  vSwitch: VTS
  Encapsulation: VxLAN
Benchmarks:
> Networks:
  > Components:
    > TOR:
        Type: N9K
        Version:
          10.28.108.249:
            BIOS: 07.34
            NXOS: 7.0(3)I2(2b)
          10.28.108.248:
            BIOS: 07.34
            NXOS: 7.0(3)I2(2b)
    > VTC:
        Version:
```

```
          build_date: 2017-03-03-05-41
          build_number: 14
          git_revision: 0983910
          vts_version: 2.3.0.40
          git_branch: vts231newton
          job_name: vts231newton_gerrit_nightly
    > Traffic Generator:
        Profile: trex-local
        Tool: TRex
        Version:
          build_date: Feb 16 2017
          version: v2.18
          built_by: hhaim
          build_time: 18:59:02
  > Measurement Parameters:
      NDR: 0.001
      PDR: 0.1
  > Service chain:
    > PVP:
      > Traffic:
          VPP version:
            sjc04-pod3-compute-4: v17.04-rc0~98-g8bf68e8
          Profile: custom_traffic_profile
          Bidirectional: True
          Flow count: 1
          Service chains count: 1
          Compute nodes: [u'nova:sjc04-pod3-compute-4']
Run Summary:

+---+-----------+-------------+-------------+-------------+---------------+---------------+---------------+

              |  -  |  L2 Frame Size  |  Rate (fwd+rev)  |  Rate (fwd+rev)  |  Avg Drop
Rate  |   Avg Latency (usec) |   Min Latency (usec) |   Max Latency (usec) |

+===+===========+=============+=============+=============+===============+===============+===============+

              | NDR |      IMIX       |   4.5703 Gbps    |  1,496,173 pps   |     0.0006%
    |               131.33 |                10.00 |               404.00 |

+---+-----------+-------------+-------------+-------------+---------------+---------------+---------------+

              | PDR |      IMIX       |   4.7168 Gbps    |  1,544,128 pps   |     0.0553%
    |               205.72 |                20.00 |               733.00 |

+---+-----------+-------------+-------------+-------------+---------------+---------------+---------------+

                        L2 frame size: IMIX
            Chain analysis duration: 961 seconds
            NDR search duration: 661 seconds
            PDR search duration: 300 seconds
                        Run Config:
                          +-------------+------------------+-------------+---------------+
                  |  Direction  |   Duration (sec) |    Rate     |     Rate      |
                  +=============+==================+=============+===============+
                  |   Forward   |               60 | 2.3584 Gbps |  772,064 pps  |
                  +-------------+------------------+-------------+---------------+
                  |   Reverse   |               60 | 2.3584 Gbps |  772,064 pps  |
                  +-------------+------------------+-------------+---------------+
                  |    Total    |               60 | 4.7168 Gbps | 1,544,128 pps |
                  +-------------+------------------+-------------+---------------+
```

Lines 1-48: Similar to the fixed rate run output explained above.

Lines 49-58: Summary of test run with benchmark data. For each packet size, there is a row with NDR/PDR or both values depending on chosen rate. The output displays the run summary for the IMIX packet size.

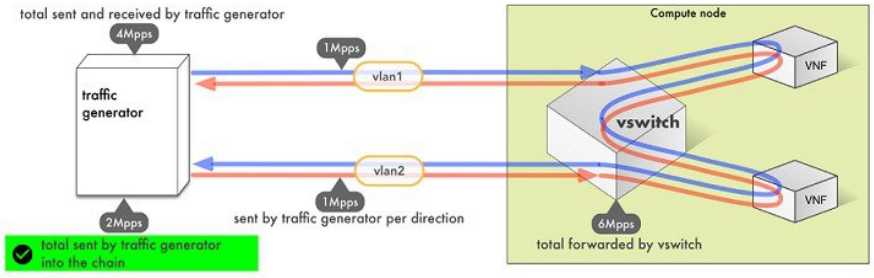
The table includes the following columns:

- L2 Frame Size: Packet size used in test, can be one of 64B, IMIX, 1518B
- Rate (fwd+rev): Total rate satisfying NDR/PDR condition in a unit bps/pps
- Avg Drop Rate: Average drop rate of test iteration which satisfied NDR/PDR condition
- Avg Latency: Average packet latency of test iteration which satisfied NDR/PDR condition in microseconds
- Min Latency: Minimum latency of test iteration which satisfied NDR/PDR condition in microseconds
- Max Latency: Maximum latency of test iteration which satisfied NDR/PDR condition in microseconds

The NDR and PDR values along with the latency information are good indicators of NFVI solution.

The following figure explains different approaches to interpret the same results.

Figure 3:



NFVBench always reports measured rate from the traffic generator perspective as total amount sent into the chain.

Advanced Configuration

More advanced use-cases require customization of the NFVbench configuration file. Steps below are optional.

The default NFVbench configuration file can be obtained by using the -show-default-config option.

For example, go to the host directory mapped to container (/root/nfvbench) and copy default NFVbench configuration file as follows:

```
[root@mgmt1 ~]# cd /root/nfvbench
[root@mgmt1 ~]# nfvbench --show-default-config > nfvbench.cfg
```

You can then edit the nfvbench.cfg using any Linux text editor (read and follow nested comments to do custom configuration) and pass the new configuration file to NFVBench using the -c option.

Cisco VIM CLI

An alternative way to NFVBench CLI is to use ciscovimclient. Ciscovimclient is meant to provide an interface that is more consistent with the CiscoVIM CLI and can run remotely while the NFVBench CLI is executed on the management node.

Pass JSON configuration matching structure of the NFVBench config file to start a test:

```
[root@mgmt1 ~]# ciscovim nfvbench --config '{"rate": "10kpps"}
+------------------+--------------------------------------+
| Name             | Value                                |
+------------------+--------------------------------------+
```

```
| status           | not_run                              |
| nfvbench_request | {"rate": "5kpps"}                    |
| uuid             | 0f131259-d20f-420f-840d-363bdcc26eb9 |
| created_at       | 2017-06-26T18:15:24.228637           |
+------------------+--------------------------------------+
```

Run the following command with the returned UUID to poll status:

```
[root@mgmt1 ~]# ciscovim nfvbench --stat  0f131259-d20f-420f-840d-363bdcc26eb9
+------------------+--------------------------------------+
| Name             | Value                                |
+------------------+--------------------------------------+
| status           | nfvbench_running                     |
| nfvbench_request | {"rate": "5kpps"}                    |
| uuid             | 0f131259-d20f-420f-840d-363bdcc26eb9 |
| created_at       | 2017-06-26T18:15:24.228637           |
| updated_at       | 2017-06-26T18:15:32.385080           |
+------------------+--------------------------------------+

+------------------+--------------------------------------+
| Name             | Value                                |
+------------------+--------------------------------------+
| status           | nfvbench_completed                   |
| nfvbench_request | {"rate": "5kpps"}                    |
| uuid             | 0f131259-d20f-420f-840d-363bdcc26eb9 |
| created_at       | 2017-06-26T18:15:24.228637           |
| updated_at       | 2017-06-26T18:18:32.045616           |
+------------------+--------------------------------------+
```

Once the test is done, retrieve results in a JSON format:

```
[root@mgmt1 ~]# ciscovim nfvbench --json 0f131259-d20f-420f-840d-363bdcc26eb9
{"status": "PROCESSED", "message": {"date": "2017-06-26 11:15:37", …}}
```

NFVBench REST Interface

When enabled, the NFVBench container can also take benchmark request from a local REST interface. Access is only local to the management node in the current Cisco VIM version (that is the REST client must run on the management node).

Details on the REST interface calls can be found in Chapter 2, Cisco VIM REST API Resources.

Enabling or Disabling Autobackup of Management Node

Cisco VIM 2.0 introduces the backup and recovery of the management node. By default, the feature is enabled. Auto-snapshots of the management node happens during pod management operation. You can disable the autobackup of the management node.

To enable or disable the management node, update the setup_data.yaml file as follows:

```
# AutoBackup Configuration
# Default is True
#autobackup: <True or False>
```

Take a backup of **setupdata** file and update it manually with the configuration details by running the following command:

```
[root@mgmt1 ~]# cd /root/
[root@mgmt1 ~]# mkdir MyDir
[root@mgmt1 ~]# cp /root/openstack-configs/setup_data.yaml /root/MyDir/
[root@mgmt1 ~]# # update the setup_data to change autobackup
[root@mgmt1 ~]# cd /root/MyDir/
```

```
[root@mgmt1 ~]# vi setup_data.yaml
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim --setupfile /root/MyDir/setup_data.yaml reconfigure
```

Forwarding ELK logs to External Syslog Server

Cisco VIM, supports backup and recovery of the management node. To keep the process predictable and to avoid loss of logs, the software supports the capability of forwarding the ELK logs to an external syslog server. Capability has been introduced to enable this feature after the pod is up and running, with Cisco VIM, through the reconfigure option.

The Syslog Export reconfigure option supports the following options:

- Enable forwarding of ELK logs to External Syslog Server on a pod that is already up and running.
- Reconfigure existing External Syslog Setting to point to a different syslog cluster.

The following section needs to be configured in the setup_data.yaml file.

```
#####################################
## SYSLOG EXPORT SETTINGS
#####################################
SYSLOG_EXPORT_SETTINGS:
  remote_host: <syslog_ip_addr>   # required, IP address of the remote syslog server
  protocol: udp                   # optional; tcp/udp, defaults to udp
  facility: local5
  severity: debug                 # <string; suggested value: debug>
  port: 514                       # optional; destination port number, defaults to 514
  clients: 'ELK'                  # optional; defaults and restricted to ELK in 2.0

# Please note other than the remote IP address host, most of the other info is not needed;
 Also the client list in 2.0 is restricted to ELK only.
```

Take a backup of the setupdata file and update the file manually with the configs listed in the preceding section, then run the reconfigure command as follows:

```
[root@mgmt1 ~]# cd /root/
[root@mgmt1 ~]# mkdir MyDir
[root@mgmt1 ~]# cp /root/openstack-configs/setup_data.yaml /root/MyDir/
[root@mgmt1 ~]# # update the setup_data to include Syslog Export info
[root@mgmt1 ~]# cd /root/MyDir/
[root@mgmt1 ~]# vi setup_data.yaml
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim --setupfile /root/MyDir/setup_data.yaml reconfigure
```

With this configuration, you should now be able to use export ELK logs to an external syslog server. On the remote host, verify if the logs are forwarded from the management node.

Updating Containers in a Running Cisco VIM Cloud

Cisco VIM allows you to update all OpenStack and infrastructure services such as RabbitMQ, MariaDB, HAProxy, and management node containers such as Cobbler, ELK, VMTP and repo containers with almost no impact to the Cisco NFVI implementation. Updates allows you to integrate Cisco VIM patch releases without redeploying the Cisco NFVI stack from the beginning. Updates have minimal service impact because they run serially component by component one node at a time. If an error occurs during an update, auto-rollback

is triggered to return the cloud to its pre-update state. After an update you can check for any functional impacts on the cloud. If everything is fine you can commit the update, which deletes the old containers and old images from the nodes. Should you see any functional cloud impact you can perform a manual rollback to start the old containers again.

Before you begin a container update, keep the following in mind:

- Updates are not supported for registry-related containers and authorized_keys.
- You cannot roll back the repo containers on the management node to an older version after they are updated because rollbacks will delete node packages and might cause the cloud to destabilize.
- To prevent double-faults, a cloud sanity check is performed before the update is started. A cloud sanity check is performed as the last step of the update.

The following table provides an overview to the methods to start the OpenStack update using Cisco VIM. The Internet options refer to management node connectivity to the Internet. If your management server lacks Internet access, you must have a staging server with Internet access to download the Cisco VIM installation artifacts to a USB stick. Cisco recommends selecting one method and staying with it for the full pod lifecycle.

Table 2: OpenStack Update Options

	Without Cisco VIM Insight	With Cisco VIM Insight
Without Internet	• Prepare the USB on a staging server • Plug the USB into the management node. • Follow the update steps in the update without Internet procedure.	• Prepare the USB on a staging server • Plug the USB into the management node. • Follow the update steps in the update without Internet procedure.
With Internet	• Download the .tgz file from the registry. • Follow the update steps in the update with Internet procedure.	• Download the .tgz file from the registry. • Follow the update steps in the update with Internet procedure.

Updating Cisco VIM Software Using a USB

The following procedure tells you how to load the Cisco VIM installation files onto a Cisco NFVI management node that does not have Internet access. Installation files include: buildnode-K9.iso, mercury-installer.tar.gz, nova-libvirt.tar, registry-2.3.1.tar.gz, and respective checksums..

Before you begin

This procedure requires a CentOS 7 staging server (VM, laptop, or UCS server) with a 64 GB USB 2.0 stick. The staging server must have Internet access (wired access is recommended) to download the Cisco VIM installation files, which you will load onto the USB stick. You then use the USB stick to load the installation files onto the management node. The installation files are around 24 GB in size, downloading them to the

USB stick might take several hours, depending on the speed of your Internet connection, so plan accordingly. Before you begin, disable the CentOS sleep mode.

Step 1 On the staging server, use yum to install the following packages:

- PyYAML (yum install PyYAML)
- python-requests (yum install python-requests)

Step 2 Connect to the Cisco VIM software download site using a web browser and login credentials provided by your account representative and download the **getartifacts.py** script from external registry.

```
# download the new getartifacts.py file (see example below)
curl -o getartifacts.py
https://<username>:<password>@cvim-registry.com/mercury-releases/mercury-rhel7-osp8/releases/<1.0.1>/getartifacts.py

curl -o getartifacts.py-checksum.txt
https://<username>:<password>@cvim-registry.com/mercury-releases/mercury-rhel7-osp8/releases/1.0.1/getartifacts.py-checksum.txt

# calculate the checksum and verify that with one in getartifacts.py-checksum.txt
sha512sum getartifacts.py

# Change the permission of getartificats.py
chmod +x getartifacts.py
```

Step 3 Run the **getartifacts.py** script. The script formats the USB 2.0 stick and downloads the installation artifacts. You will need to provide the registry username and password, the tag ID, and the USB partition on the staging server. For example:

To identify the USB drive, execute the **lsblk** command before and after inserting the USB stick. (The command displays a list of available block devices.) The output delta will help find the USB drive location. Provide the entire drive path in the –d option, instead of any partition.

sudo ./ getartifacts.py -t <tag_id> -u <username> -p <password> -d </dev/sdc>

Note Do not remove the USB stick while the synchronization is under way.

Step 4 Verify the integrity of the downloaded artifacts and the container images:

```
# create a directory
sudo mkdir -p /mnt/Cisco

# /dev/sdc is the USB drive, same as supplied in get artifacts.py python script
sudo mount /dev/sdc1 /mnt/Cisco
cd /mnt/Cisco

# execute the verification script
./test-usb

# failures will be explicitly displayed on screen, sample success output below
# sample output of ./test-usb execution with 2.0.1 release
[root@mgmtnode Cisco]# ./test-usb
INFO: Checking the integrity of this USB stick
INFO: Checking artifact buildnode-K9.iso
INFO: Checking artifact mercury-version.txt
INFO: Checking artifact registry-2.3.1.tar.gz
INFO: Checking artifact nova-libvirt-K9.tar.gz
INFO: Checking required layers:
INFO: 395 layer files passed checksum.
[root@mgmtnode Cisco]#
```

Step 5 To resolve download artifact failures, unmount the USB and run the getartifacts command again with the --retry option:

```
sudo ./getartifacts.py -t <tag_id> -u <username> -p <password> -d </dev/sdc> --retry
```

Step 6 Mount the USB and then run the test-usb command to validate all the files are downloaded:

```
# /dev/sdc is the USB drive, same as supplied in get artifacts.py python script
sudo mount /dev/sda1 /mnt/Cisco
cd /mnt/Cisco

# execute the verification script
./test-usb

# In case of failures the out of the above command will explicitly display the same on the screen
```

Step 7 After the synchronization finishes, unmount the USB stick:

```
sudo umount /mnt/Cisco
```

Step 8 After the synchronization finishes, remove the USB stick from the staging server then insert it into the management node.

Step 9 Complete the following steps to import the Cisco NFVI installation artifacts onto the management node:

a) Identify the USB on the management node:

```
blkid -L Cisco-VIM
```

b) Mount the USB device on the management node:

```
 mount < /dev/sdc > /mnt/
  cd /tmp/
```

c) Extract the import_artifacts.py script:

```
tar --no-same-owner -xvzf /mnt/mercury-installer.tar.gz installer-< xxxx >/
tools/import_artifacts.sh
```

d) Copy the import_artifacts.py script:

```
cp installer-< xxxx >/tools/import_artifacts.sh /root/
rm -fr /tmp/installer-< xxxx >
```

e) Unmount the USB device:

```
umount /mnt/cd /root/
```

f) Import the artifacts:

```
./import_artifacts.sh
```

Step 10 Verify the image version and change ID for the software update.

```
cat /var/cisco/artifacts/mercury-version.txt
```

Step 11 Execute the update from the old working directory:

```
a.cd $old_workspace/installer;
b./ciscovimclient/ciscovim update --file /var/cisco/artifacts/mercury-installer.tar.gz
```

After the update is complete, use the newly created directory from here onwards (unless a rollback is planned).

Step 12 Commit the update by running the following command:

```
./ciscovimclient/ciscovim commit # from the new workspace
```

Step 13 To revert the update changes before entering the commit command, enter:

```
./ciscovimclient/ciscovim rollback # and then use older workspace
```

Note Do not run any other Cisco VIM actions while the update is underway.

Updating Cisco VIM Software Using Network Installation

Step 1 From the download site provided by your Cisco account representative, download the mercury-installer.gz

```
curl -o mercury-installer.tar.gz
https://{username}:{password}@mercury-registry.cisco.com/
mercury-releases/mercury-rhel7-osp8/releases/{release number}/
mercury-installer.tar.gz
```

The link to the tar ball above is an example.

Step 2 Execute the update from the old working directory:

Note Do not run any other Cisco VIM actions while the update is underway.

```
cd $old_workspace/installer;
/ciscovimclient/ciscovim update –file /root/mercury-installer.tar.gz
```

After the update is complete, use the newly created directory from here onwards (unless a rollback is planned).

Step 3 Commit the update by running the following command:

```
./ciscovimclient/ciscovim commit
```

Step 4 To revert the update changes before entering the commit command, enter:

```
./ciscovimclient/ciscovim rollback # and then use older workspace
```

VM Resizing

VM resize is the process of changing the flavor of an existing VM. Thus, using VM resize you can upscale a VM according to your needs. The size of a VM is indicated by the flavor based on which the VM is launched.

Resizing an instance means using a different flavor for the instance.

By default, the resizing process creates the newly sized instance on a new node, if more than one compute node exists and the resources are available. By default, the software, allows you to change the RAM size, VDISK size, or VCPU count of an OpenStack instance using **nova resize**. Simultaneous or individual adjustment of properties for the target VM is allowed. If there is no suitable flavor for the new properties of the VM, you can create a new one.

```
nova resize [--poll] <server> <flavor>
```

The resize process takes some time as the VM boots up with the new specifications. For example, the Deploying a Cisco CSR (size in MB) would take approximately 60mins. After the resize process, execute `nova resize-confirm <server>` to overwrite the old VM image with the new one. If you face any issue, you revert to the old VM using the `nova-resize-revert <server>` command. to drop the new VM image and use the old one. At this point, you can access the VM through SSH and check if the right image is configured.

Note OpenStack will **shutdown** the VM before the resize, so you have to plan for a **downtime.**

Note We recommend you not to resize a vdisk to a smaller value, as there is the risk of losing data.

Nova Migrate

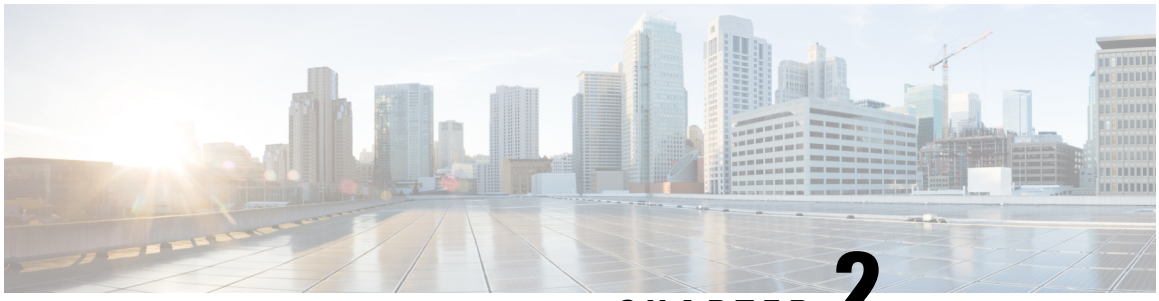
The nova migrate command is used to move an instance from one compute host to another compute host. The scheduler chooses the destination compute host based on the availability of the zone settings. This process does not assume that the instance has shared storage available on the target host.

To initiate the cold migration of the VM, you can execute the following command:

```
nova migrate [--poll] <server>
```

The VM migration can take a while, as the VM boots up with the new specifications. After the VM migration process, you can execute `nova resize-confirm <server>` --to overwrite the old VM image with the new one.. If you face any issue, use the `nova-resize-revert <server>` command to revert to the old VM image. At this point, access the VM through SSH and check the right image is configured.

Note OpenStack will **shutdown** the VM before the migrate, so plan for a **downtime**.



CHAPTER 2

Cisco VIM REST API

The following topics explain how to use the Cisco VIM REST API to manage Cisco NFVI.

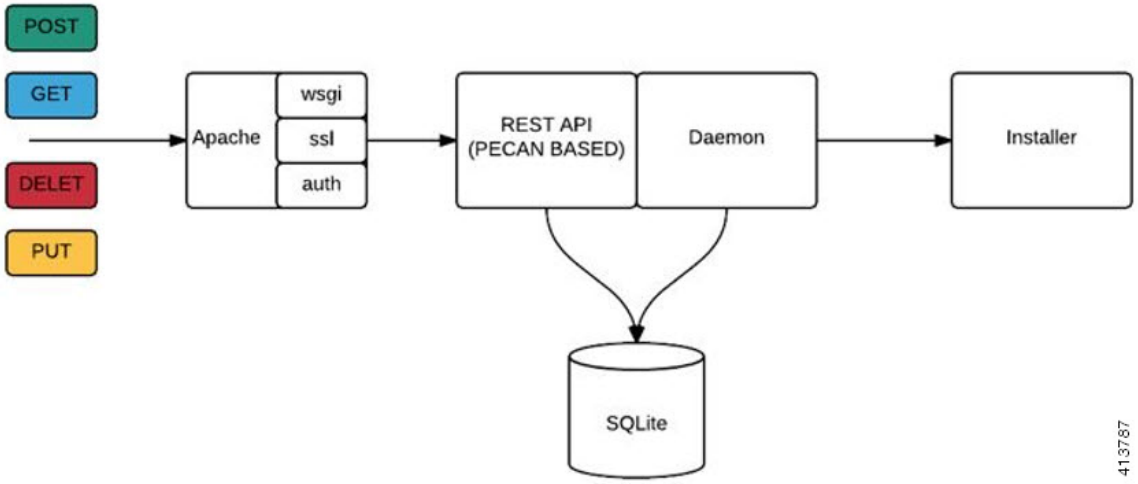
Overview to Cisco VIM REST API

Cisco VIM provides a Representational State Transfer (REST) API that you can use to install, expand, and update Cisco VIM. Actions you can perform using the RESTful API include:

- Install Cisco VIM on Cisco NFVI pods.
- Add and delete pods to and from Cisco NFVI installations.
- Update Cisco VIM software.
- Replace controller nodes.
- Perform cloud maintenance operations.
- Run cloud validations using Virtual Machine ThroughPut (VMTP), a data path performance measurement tool for OpenStack clouds.

The figure below shows the Cisco VIM REST API flow.

Figure 4: Cisco VIM REST API Flow



The Cisco VIM REST API security is provided by the Secure Sockets Layer (SSL) included on the Apache webserver. The Pecan-based web application is called by mod_wsgi, which runs the Rest API server. The Pecan REST API server requires a username and password to authorize REST API server requests. Apache handles the authorization process, which authorizes the request to access the Pecan web application. You can use the Cisco VIM API to upload a new setup_data.yaml file, and start, stop, and query the state of the installation. You can also use it to manage the cloud, add and remove compute and Ceph nodes, and replace the controller nodes. A REST API to launch VMTP (L2/L3 data plane testing) and CloudPulse is also provided.

The Cisco VIM REST API is enabled by default in the management node, if you are using the supplied Cisco VIM buildnode.iso. You can access API server on the br_api interface on port 8445. Authentication is enabled by default in the web service.

The API end points can be reached with the following URL format:

https://<Management_node_api_ip>:8445

The API endpoint expects a basic authentication which is enabled by default in the management node. The authentication credentials can be found in /opt/cisco/ui_config.json in the management node. Sample ui_config.json contents are as shown below:

```
{
  "Kibana-Url": "http://10.10.10.10:5601",
  "RestAPI-Url": "https:// 10.10.10.10:8445",
  "RestAPI-Username": "admin",
  "RestAPI-Password": "a96e86ccb28d92ceb1df",
  "BuildNodeIP": "10.10.10.10"
}
```

Cisco VIM REST API Resources

Setupdata

REST wrapper for setupdata. Provides methods for listing, creating, modifying and deleting setupdata.

Retrieving the setupdata

Resource URI

Verb	URI
GET	/v1/setupdata

Example

JSON Request

```
GET /v1/setupdata
Accept: application/json
```

JSON Response

```
200 OK
Content-Type: application/json
{"setupdatas": [{
     "status": "Active",
     "name":"GG34",
     "uuid": "123"
     "meta":{
           "user":"root"
     },
     "jsondata":{
       .......
     }
 }]}
```

Creating the setupdata

Resource URI

Verb	URI
POST	/v1/setupdata

Example

JSON Request

```
POST /v1/setupdata
Accept: application/json

{
    "name":"GG34",
    "uuid": "123"
    "meta":{
         "user":"root"
  },
  "jsondata":{
    .......
  }
}
```

JSON Response

```
201 OK
Content-Type: application/json
{
     "status": "Active",
     "name":"GG34",
     "uuid": "123"
```

```
     "meta":{
         "user":"root"
  },
  "jsondata":{
    .......
  }
}

400 Bad Request
Content-Type: application/json
{
    "debuginfo": null
    "faultcode": "Client"
    "faultstring": "Error"
}

409 CONFLICT
Content-Type: application/json
{
     "debuginfo": null
     "faultcode": "Client"
     "faultstring": "Error"
}
```

Retrieving a single setupdata

Resource URI

Verb	URI
GET	/v1/setupdata/(id)

Property:

id - the id of the setupdata to be queried.

Example

JSON Request

```
GET /v1/setupdata/123
Accept: application/json
```

JSON Response

```
200 OK
Content-Type: application/json
{
     "status": "Active",
     "name":"GG34",
     "uuid": "123"
     "meta":{
         "user":"root"
  },
  "jsondata":{
    .......
  }
}

404 NOT FOUND
Content-Type: application/json
{
     "debuginfo": null
```

```
     "faultcode": "Client"
     "faultstring": "Setupdata could not be found."
}
```

Updating a setupdata

Resource URI

Verb	URI
PUT	/v1/setupdata/(id)

Property:

id - the id of the setupdata to be updated.

Example

JSON Request

```
PUT /v1/setupdata/123
Accept: application/json
```

JSON Response

```
200 OK
Content-Type: application/json
{
     "status": "Active",
     "name":"GG34",
     "uuid": "123"
     "meta":{
         "user":"root"
  },
  "jsondata":{
    .......
  }
}

404 NOT FOUND
Content-Type: application/json
{
     "debuginfo": null
     "faultcode": "Client"
     "faultstring": "Setupdata could not be found."
}
```

Deleting a setupdata

Resource URI

Verb	URI
DELETE	/v1/setupdata/(id)

Property:

id - the id of the setupdata to be deleted.

Example

JSON Request

```
DELETE /v1/setupdata/123
Accept: application/json
```

JSON Response

```
204 NO CONTENT
Returned on success

404 NOT FOUND
Content-Type: application/json
{
    "debuginfo": null
    "faultcode": "Client"
    "faultstring": "Setupdata could not be found."
}
400 BAD REQUEST
Content-Type: application/json

{
     "debuginfo": null
     "faultcode": "Client"
     "faultstring": "Setupdata cannot be deleted when it is being used by an installation"
}
```

Install resource

REST wrapper for install. Provides methods for starting, stopping, and viewing the status of the installation process.

Return a list of installation

Resource URI

Verb	URI
GET	/v1/install

Example

JSON Request

```
GET /v1/install
Accept: application/json
```

JSON Response

```
200 OK
Content-Type: application/json
{"installs": [{
      "ceph": "Skipped",
      "uuid": "123",
      "setupdata": "345",
      "vmtpresult": "{
         "status": "PASS",
         "EXT_NET": []
      }",
      "baremetal": "Success",
      "orchestration": "Success",
      "validationstatus": "{
```

```
         "status": "PASS",
         "Software_Validation": [],
         "Hardware_Validation": []
      }",
      "currentstatus": "Completed",
      "validation": "Success",
      "hostsetup": "Success",
      "vmtp": "Skipped"
    }]
}
```

Create an installation

Resource URI

Verb	URI
POST	/v1/install

Example

JSON Request

```
GET /v1/install
Accept: application/js
{
     "setupdata": "123",
     "stages": [
       "validation",
       "bootstrap",
       "runtimevalidation",
       "baremetal",
       "orchestration",
       "hostsetup",
       "ceph",
       "vmtp"
     ]
}
```

JSON Response

```
201  CREATED
Content-Type: application/json
{
    "ceph": "Skipped",
    "uuid": "123",
    "setupdata": "345",
    "vmtpresult": "{
       "status": "PASS",
       "EXT_NET": []
    }",
    "baremetal": "Success",
    "orchestration": "Success",
    "validationstatus": "{
       "status": "PASS",
       "Software_Validation": [],
       "Hardware_Validation": []
    }",
     "currentstatus": "Completed",
     "validation": "Success",
     "hostsetup": "Success",
     "vmtp": "Skipped"
```

```
 }


409 CONFLICT
Content-Type: application/json
{
     "debuginfo": null
     "faultcode": "Client"
     "faultstring": "Install already exists"
}
```

Retrieve the installation

Resource URI

Verb	URI
GET	/v1/install/{id}

Property:

id - the id of the install to be queried.

Example

JSON Request

```
GET /v1/install/345
Accept: application/js
```

JSON Response

```
200  OK
Content-Type: application/json
{
    "ceph": "Skipped",
    "uuid": "123",
    "setupdata": "345",
    "vmtpresult": "{
       "status": "PASS",
       "EXT_NET": []
    }",
    "baremetal": "Success",
    "orchestration": "Success",
    "validationstatus": "{
      "status": "PASS",
      "Software_Validation": [],
      "Hardware_Validation": []
    }",
    "currentstatus": "Completed",
    "validation": "Success",
    "hostsetup": "Success",
    "vmtp": "Skipped"
}


404 NOT FOUND
Content-Type: application/json
{
     "debuginfo": null
     "faultcode": "Client"
     "faultstring": "Install doesn't exists"
```

```
}
```

Stop the installation

Resource URI

Verb	URI
DELETE	/v1/install/{id}

Property:

id - the id of the install to be stopped.

Example

JSON Request

```
DELETE /v1/install/345
Accept: application/js
```

JSON Response

```
204 NO CONTENT
Content-Type: application/json

404 NOT FOUND
Content-Type: application/json
{
     "debuginfo": null
     "faultcode": "Client"
     "faultstring": "Install doesn’t exists"
}
```

Nodes

Getting a list of nodes

Resource URI

Verb	URI
GET	/v1/nodes

Example

JSON Request

```
Get /v1/nodes
Accept: application/js
```

JSON Response

```
200 OK
Content-Type: application/json
{
    "nodes": [
        [
           "status": "Active",
           "uuid": "456",
```

```
            "setupdata": "123",
            "node_data": "{
              "rack_info": {
                 "rack_id": "RackA"
              },
              "cimc_info": {
                "cimc_ip": "10.10.10.10"
              },
              "management_ip": "7.7.7.10"
              }",
              "updated_at": null,
              "mtype": "compute",
              "install": "345",
              "install_logs": "logurl",
              "created_at":"2016-0710T06:17:03.761152",
              "name": " compute-1"
              }
          ]
}
```

Add new nodes

The nodes are in compute or block_storage type. Before adding the nodes to the system, the name of the nodes and other necessary information like cimc_ip and rackid must be updated in the setupdata object. If the setupdata object is not updated, the post call will not allow you to add the node.

Resource URI

Verb	URI
POST	/v1/nodes

Example

JSON Request

```
POST /v1/nodes
Accept: application/js
{
    "name" : "compute-5"
}
```

JSON Response

```
201 CREATED
Content-Type: application/json
{
     "status": "ToAdd",
     "uuid": "456",
     "setupdata": "123",
     "node_data": "{
        "rack_info": {
        "rack_id": "RackA"
        },
        "cimc_info": {
         "cimc_ip": "10.10.10.10"
        },
        "management_ip": "7.7.7.10"
        }",
        "updated_at": null,
        "mtype": "compute",
```

```
        "install": "345",
        "install_logs": "logurl",
        "created_at":"2016-0710T06:17:03.761152",
        "name": " compute-1"
}
```

Retrieve information about a particular node

Resource URI

Verb	URI
GET	/v1/nodes{id}

Property:

id - the id of the node to be queried.

Example

JSON Request

```
POST /v1/nodes
Accept: application/js
```

JSON Response

```
200 OK
Content-Type: application/json
{
     "status": "Active",
     "uuid": "456",
     "setupdata": "123",
     "node_data": "{
       "rack_info": {
        "rack_id": "RackA"
       },
       "cimc_info": {
         "cimc_ip": "10.10.10.10"
       },
       "management_ip": "7.7.7.10"
       }",
       "updated_at": null,
       "mtype": "compute",
       "install": "345",
       "install_logs": "logurl",
       "created_at":"2016-0710T06:17:03.761152",
       "name": " compute-1"
}

404 NOT FOUND
Content-Type: application/json
{
    "debuginfo": null
    "faultcode": "Client"
    "faultstring": "Node doesn't exists"
}
```

Remove a node

The node that must be deleted must be removed from the setupdata object. Once the setupdata object is updated, you can safely delete of the node. The node object will not be deleted until it calls the remove node backend and succeeds.

Resource URI

Verb	URI
DELETE	/v1/nodes{id}

Property:

id - the id of the node to be removed.

Example

JSON Request

```
DELETE /v1/nodes/456
Accept: application/js
```

JSON Response

```
204 ACCEPTED
Content-Type: application/json

404 NOT FOUND
Content-Type: application/json
{
     "debuginfo": null
     "faultcode": "Client"
     "faultstring": "Node doesn’t exists"
}
```

For clearing the database and deleting the entries in the nodes, the delete api is called with special parameters that are passed along with the delete request. The JSON parameters are in the following format.

JSON Request

```
DELETE /v1/nodes/456
Accept: application/js
{
    "clear_db_entry":"True"\
}
```

JSON Response

```
204 ACCEPTED
Content-Type: application/json

404 NOT FOUND
Content-Type: application/json
{
    "debuginfo": null
    "faultcode": "Client"
    "faultstring": "Node doesn’t exists"
}
```

Note This is done only if the node is deleted from the REST API database. The failure reason of the node must be rectified manually apart from the API. True is a string and not a boolean in the above line.

Replace a controller

Resource URI

Verb	URI
PUT	/v1/nodes{id}

Property:

id - the id of the controller to be replaced.

Example

JSON Request

```
PUT /v1/nodes/456
Accept: application/js
```

JSON Response

```
200 OK
Content-Type: application/json

404 NOT FOUND
Content-Type: application/json
{
    "debuginfo": null
    "faultcode": "Client"
    "faultstring": "Node doesn't exists"
}
```

Offline validation

REST wrapper does the offline validation of setupdata. This will only do S/W Validation of the input setupdata.

Create an offline validation operation

Resource URI

Verb	URI
POST	/v1/offlinevalidation

Example

JSON Request

```
POST /v1/offlinevalidation
Accept: application/json
{
      "jsondata": ".. .. .."
}
```

JSON Response

```
201 CREATED
Content-Type: application/json
{
     "status": "NotValidated",
     "uuid": "bb42e4ba-c8b7-4a5c-98b3-1f384aae2b69",
     "created_at": "2016-02-03T02:05:28.384274",
     "updated_at": "2016-02-03T02:05:51.880785",
     "jsondata": "{}",
     "validationstatus": {
        "status": "PASS",
        "Software_Validation": [],
        "Hardware_Validation": []
      }
}
```

Retrieve the results of offline validation

Resource URI

Verb	URI
GET	/v1/offlinevalidation

Property:

id - the id of the node to be queried.

Example

JSON Request

```
GET /v1/offlinevalidation/789
Accept: application/json
```

JSON Response

```
200 OK
Content-Type: application/json
{
    "status": " ValidationSuccess",
    "uuid": "bb42e4ba-c8b7-4a5c-98b3-1f384aae2b69",
    "created_at": "2016-02-03T02:05:28.384274",
    "updated_at": "2016-02-03T02:05:51.880785",
    "jsondata": "{}",
    "validationstatus": {
       "status": "PASS",
       "Software_Validation": [],
       "Hardware_Validation": []
     }
}
```

Update

Start an update process

Resource URI

Verb	URI

POST	/v1/update

Parameters:

- fileupload - "tar file to upload"
- filename - "Filename being uploaded"

Example

JSON Request

```
curl -sS -X POST --form
"fileupload=@Test/installer.good.tgz" --form
"filename=installer.good.tgz"
https://10.10.10.8445/v1/update
```

Note This curl request is done as a form request.

JSON Response

```
200 OK
Content-Type: application/json
{
    "update_logs": "logurl",
    "update_status": "UpdateSuccess",
    "update_filename": "installer-4579.tgz",
    "created_at": "2016-07-10T18:33:52.698656",
    "updated_at": "2016-07-10T18:54:56.885083"
}

409 CONFLICT
Content-Type: application/json
{
    "debuginfo": null
    "faultcode": "Client"
    "faultstring": "Uploaded file is not in tar format"
 }
```

Rollback an update

Resource URI

Verb	URI
PUT	/v1/update

Example

JSON Request

```
PUT /v1/update
Accept: application/json
{
     "action":"rollback"
}
```

JSON Response

```
200 OK
Content-Type: application/json
{
    "update_logs": "logurl",
    "update_status": "ToRollback",
    "update_filename": "installer-4579.tgz",
    "created_at": "2016-07-10T18:33:52.698656",
    "updated_at": "2016-07-10T18:54:56.885083"
}
```

Commit an update

Resource URI

Verb	URI
PUT	/v1/update

Example

JSON Request

```
PUT /v1/update
Accept: application/json
{
"action":"commit"
}
```

JSON Response

```
200 OK
Content-Type: application/json
{
    "update_logs": "logurl",
    "update_status": "ToCommit",
    "update_filename": "installer-4579.tgz",
    "created_at": "2016-07-10T18:33:52.698656",
    "updated_at": "2016-07-10T18:54:56.885083"
}
```

Retrieve the details of an update

Resource URI

Verb	URI
GET	/v1/update

Example

JSON Request

```
GET /v1/update
Accept: application/json
```

JSON Response

```
200 OK
Content-Type: application/json
{
    "update_logs": "logurl",
    "update_status": "UpdateSuccess",
    "update_filename": "installer-4579.tgz",
    "created_at": "2016-07-10T18:33:52.698656",
    "updated_at": "2016-07-10T18:54:56.885083"
}
```

Secrets

Retrieve the list of secrets associated with the OpenStack Setup

You can retrieve the set of secret password associated with the OpenStack setup using the above api. This gives the list of secrets for each service in OpenStack.

Resource URI

Verb	URI
GET	/v1/secrets

Example

JSON Request

```
GET /v1/secrets
Accept: application/json
```

JSON Response

```
200 OK
Content-Type: application/json
{
"HEAT_KEYSTONE_PASSWORD": "xxxx",
"CINDER_KEYSTONE_PASSWORD": "xxxxx",
….
….
"RABBITMQ_PASSWORD": "xxxxx"
}
```

OpenStack Configs

Retrieve the list of OpenStack configs associated with the OpenStack Setup

You can retrieve the set of OpenStack configs associated with the OpenStack setup using the above api. This gives the current settings of different configs like verbose logging, debug logging for different OpenStack services.

Resource URI

Verb	URI
GET	/v1/openstack_config

Example

JSON Request

```
GET /v1/openstack_config
Accept: application/json
```

JSON Response

```
200 OK
Content-Type: application/json
{
"CINDER_DEBUG_LOGGING": false,
"KEYSTONE_DEBUG_LOGGING": false,
….
….
"NOVA_VERBOSE_LOGGING": true
}
```

Version

Retrieve the version of the Cisco Virtualized Infrastructure Manager.

Resource URI

Verb	URI
GET	/v1/version

Example

JSON Request

```
GET /v1/version
Accept: application/json
```

JSON Response

```
200 OK
Content-Type: application/json
{"version": "1.9.1"}
```

Health of the Management Node

Retrieve the health of the Management node

This api can be used to retrieve the health of the management node. It checks various parameters like partitions, space and so on.

Resource URI

Verb	URI
GET	/v1/health

Example

JSON Request

```
GET /v1/health
Accept: application/json
```

JSON Response

```
200 OK
Content-Type: application/json
{
    "status": "PASS",
    "BuildNode Validation": {
        "Check Docker Pool Settings": {"status": "Pass", "reason": "None"}
        ….
        ….
```

```
    }
}
```

Hardware Information

REST wrapper to do hardware information of setupdata. This will return the hardware information of all hardware available in the setupdata.

Create a HWinfo operation

Resource URI

Verb	URI
GET	/v1/hwinfo

Example

JSON Request

```
POST /v1/hwinfo
Accept: application/json
{
       "setupdata":"c94d7973-2fcc-4cd1-832d-453d66e6b3bf"
}
```

JSON Response

```
201 CREATED
Content-Type: application/json
{
  "status": "hwinfoscheduled",
  "uuid": "928216dd-9828-407b-9739-8a7162bd0676",
  "setupdata": "c94d7973-2fcc-4cd1-832d-453d66e6b3bf",
  "created_at": "2017-03-19T13:41:25.488524",
  "updated_at": null,
  "hwinforesult": ""
}
```

Retrieve the results of Hwinfo Operation

Resource URI

Verb	URI
GET	/v1/hwinfo/{id}

Property:

id - the id of the node to be queried.

Example

JSON Request

```
GET /v1/hwinfo/789
Accept: application/json
```

JSON Response

```
200 OK
Content-Type: application/json
{
  "status": "hwinfosuccess",
  "uuid": "928216dd-9828-407b-9739-8a7162bd0676",
```

```
  "setupdata": "c94d7973-2fcc-4cd1-832d-453d66e6b3bf",
  "created_at": "2017-03-19T13:41:25.488524",
  "updated_at": "2017-03-19T13:42:05.087491",
  "hwinforesult": "{\"172.29.172.73\": {\"firmware\": ……..
  ……
  ……..
}
```

Release mapping Information

This api is used to see the list of Features included and list of options which can be reconfigured in the Openstack Setup.

Retrieve the release mapping information

Resource URI

Verb	URI
GET	/v1/releasemapping

JSON Request

```
GET /v1/releasemapping
Accept: application/json
```

JSON Response

```
200 OK
Content-Type: application/json
[
  {
    "SWIFTSTACK": {
      "feature_status": true,
      ],
      "desc": "swift stack feature"
    }
  },……..
  ……..
}
```

POST Install operations

The following are the post install operations that can be carried on once the OpenStack installation is carried out successfully. It uses a common api. So only one operation is given as an example below:

1. reconfigure,
2. reconfigure -regenerate passwords
3. reconfigure -setpasswords,setopenstack_configs,
4. check-fernet-keys
5. period-rotate-fernet-keys
6. resync-fernet-keys
7. rotate-fernet-keys

Create a post install operation

Resource URI

Verb	URI
POST	/v1/misc

Example

JSON Request

```
POST /v1/misc
Accept: application/json
{"action": {"reconfigure": true}}
```

JSON Response

```
201 CREATED
Content-Type: application/json
{
  "uuid": "7e30a671-bacf-4e3b-9a8f-5a1fd8a46733",
  "created_at": "2017-03-19T14:03:39.723914",
  "updated_at": null,
  "operation_status": "OperationScheduled",
  "operation_logs": "",
  "operation_name": "{"reconfigure": true}"
}
```

Retrieve a status of the post install operation

Resource URI

Verb	URI
GET	/v1/misc

Example

JSON Request

```
GET /v1/misc
Accept: application/json
```

JSON Response

```
201 CREATED
Content-Type: application/json
{
  "uuid": "7e30a671-bacf-4e3b-9a8f-5a1fd8a46733",
  "created_at": "2017-03-19T14:03:39.723914",
  "updated_at": "2017-03-19T14:03:42.181180",
  "operation_status": "OperationRunning",
  "operation_logs": "xxxxxxxxxxxxxxxx",
  "operation_name": "{\"reconfigure\": true}"
}
```

NFVBench Network Performance Testing

Create NFVBench Run

Starts network performance test with provided configuration.

Resource URI

Verb	URI
POST	/v1/nfvbench

Example

JSON Request

```
NDR/PDR Run
POST /v1/nfvbench
Accept: application/json
{“nfvbench_request”:
{
    "duration_sec": 20,
    "traffic_profile":  [
         {
            "name": "custom",
            "l2frame_size": [
               "64",
               "IMIX",
               "1518"
            ]
         }
   ],
   "traffic": {
      "bidirectional": true,
      "profile": "custom"
   },
   "flow_count": 1000
}
}

Fixed Rate Run
POST /v1/nfvbench
Accept: application/json
{“nfvbench_request”:
{
    "rate": "1000000pps",
    "duration_sec": 60,
    "traffic_profile":  [
         {
            "name": "custom",
            "l2frame_size": [
               "1518"
            ]
         }
   ],
   "traffic": {
      "bidirectional": true,
      "profile": "custom"
   },
   "flow_count": 1000
}
}
```

JSON Response

```
JSON Response
201 CREATED
Content-Type: application/json
{
    "created_at": "2017-05-30T21:40:40.394274",
    "nfvbench_request": "{\"duration_sec\": 20, \"traffic_profile\": …}",
    "nfvbench_result": "",
    "status": "not_run",
    "updated_at": null,
    "uuid": "e08d5642-3361-41e0-b91a-13036705011a"
}
```

Status Polling

Polling of NFVbench run status which is one of "nfvbench_running", "nfvbench_failed", "nfvbench_completed".

Resource URI

Verb	URI
GET	/v1/nfvbench/<uuid>

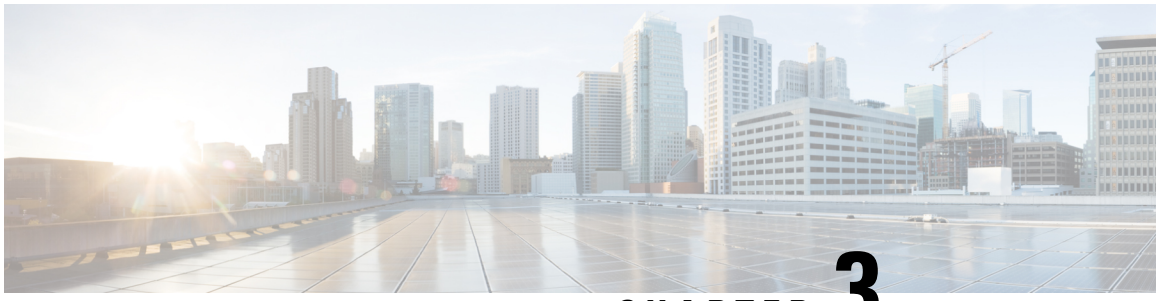
Example

JSON Request

```
UUID is returned by POST call.
GET /v1/nfvbench/e08d5642-3361-41e0-b91a-13036705011a
Accept: application/json
```

JSON Response

```
Run in Progress
200 OK
Content-Type: application/json
{
    "created_at": "2017-05-30T21:40:40.394274",
    "nfvbench_request": "{\"duration_sec\": 20, \"traffic_profile\": …}",
    "nfvbench_result": "",
    "status": "nfvbench_running",
    "updated_at": "2017-05-30T21:40:41.367279",
    "uuid": "e08d5642-3361-41e0-b91a-13036705011a"
}
Run Completed
200 OK
Content-Type: application/json
{
    "created_at": "2017-05-30T21:40:40.394274",
    "nfvbench_request": "{\"duration_sec\": 20, \"traffic_profile\": …}",
    "nfvbench_result": "{\"status\": \"PROCESSED\", \"message\": {\"date\": \"2017-05-30
14:40:46\", ...}}"
    "status": "nfvbench_completed",
    "updated_at": "2017-05-30T22:29:56.970779",
    "uuid": "e08d5642-3361-41e0-b91a-13036705011a"
}
```



CHAPTER 3

Monitoring Cisco NFVI Performance

The following topics tell you how to display logs to monitor Cisco VIM performance.

Logging and Monitoring in Cisco NFVI

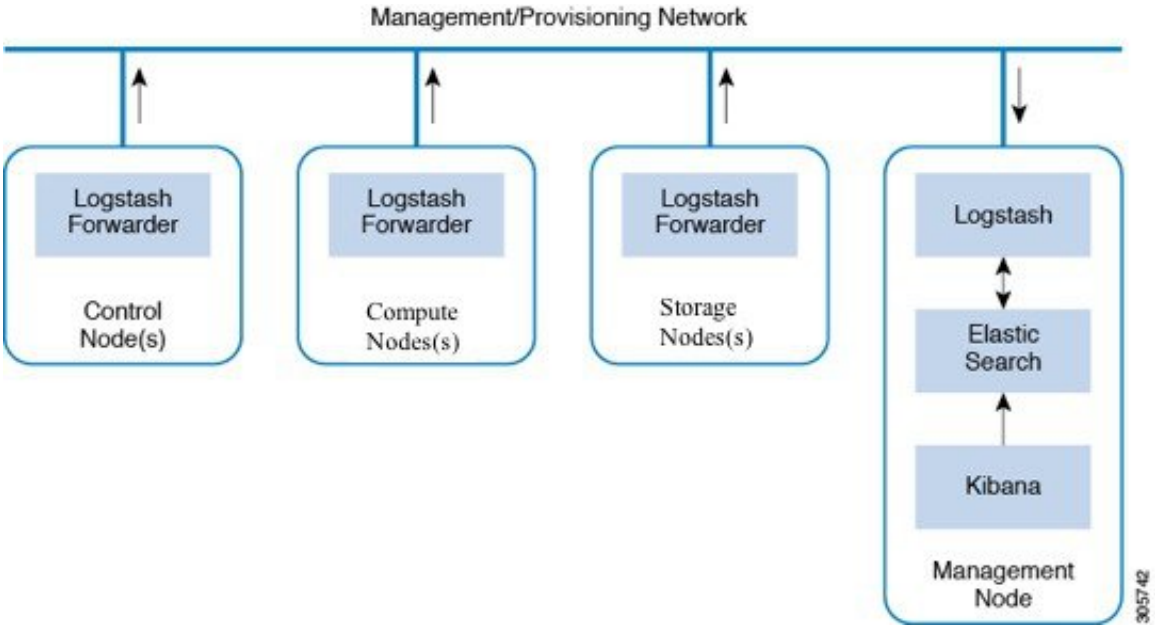
Cisco VIM uses a combination of open source tools to collect and monitor the Cisco OpenStack services including Elasticsearch, Logstash, and the Kibana (ELK) dashboard. OpenStack services that are tracked by ELK include:

- MariaDB—A relational database management system based on MySQL. All the OpenStack components store their data in MariaDB.
- HAProxy—HAProxy is a free open source software that provides a high-availability load balancer, and proxy server for TCP and HTTP-based applications that spreads requests across multiple servers.
- Keystone—Keystone is an OpenStack project that provides identity, token, catalog, and policy services for use specifically by projects in the OpenStack.
- Glance—An OpenStack project that allows you to upload and discover data assets meant for use with other services.
- Neutron—An OpenStack project that provides network connectivity between interface devices, such as vNICs, managed by other OpenStack services, such as Nova.
- Nova—An OpenStack project designed to provide massively scalable, on demand, self-service access to compute resources.
- HTTPD—The Apache HTTP Server Project, an effort to develop and maintain an open-source HTTP server.
- Cinder—An OpenStack block storage service designed to present storage resources to end users that are consumed by the OpenStack compute project (Nova).
- Memcached—A general purpose distributed memory caching system.

- CloudPulse—Is an OpenStack tool that checks the health of the cloud. CloudPulse includes operator and endpoint tests.
- Heat—The main OpenStack Orchestration program. Heat implements an orchestration engine to launch multiple composite cloud applications based on text file templates.
- Other OpenStack services—RabbitMQ, Ceph, Open vSwitch, Linux bridge, Neutron VTS (optional), and others.
- VMTP—Integrated control and data plane log for testing the cloud.
- NFVbench—Network performance benchmarking tool.

A Logstash forwarder container resides on each control, compute, and storage nodes. They forward logs to the Logstash server residing on the management node. The following figure shows a high-level schematic of the Logstash service assurance architecture.

Figure 5: Logstash Service Assurance Architecture



The Logstash flow includes:

- The Logstash forwarder verifies the identity of the Logstash server running on the management node and sends the logs from all the compute, controller, and storage nodes to the Logstash server.
- Logstash extracts the relevant data from the forwarded logs using the custom grok filters and makes it available to Elasticsearch.
- Elasticsearch stores the data, indexes it, and supports extremely fast queries against a large amount of log data.
- Kibana visualizes the data in Elasticsearch using a custom dashboard with REST API calls to Elasticsearch.

Displaying Cisco VIM Log Files Using the CLI

Cisco VIM log file location depends on the node and log type. Installer logs can be found on the management node under /var/log/mercury/<install_uuid>/ directory. The last twenty log directories are tarred and kept in this directory. These files contain logs related to bootstrap, buildorchestration, baremetal, common setup, and OpenStack orchestration. If the installer fails, look at the last tar.gz file for logs, for example:

```
[root@mgmtnode mercury]# ls -lrt
total 20
drwxr-xr-x. 2 root root   80 Jul 19 23:42 573f2b7f-4463-4bfa-b57f-98a4a769aced
drwxr-xr-x. 2 root root 4096 Jul 20 03:29 installer
drwxr-xr-x. 2 root root   79 Jul 20 03:29 e9117bc5-544c-4bda-98d5-65bffa56a18f
drwxr-xr-x. 2 root root   79 Jul 20 04:54 36cdf8b5-7a35-4e7e-bb79-0cfb1987f550
drwxr-xr-x. 2 root root   79 Jul 20 04:55 bd739014-fdf1-494e-adc0-98b1fba510bc
drwxr-xr-x. 2 root root   79 Jul 20 04:55 e91c4a6c-ae92-4fef-8f7c-cafa9f5dc1a3
drwxr-xr-x. 2 root root   79 Jul 20 04:58 1962b2ba-ff15-47a6-b292-25b7fb84cd28
drwxr-xr-x. 2 root root   79 Jul 20 04:59 d881d453-f6a0-448e-8873-a7c51d8cc442
drwxr-xr-x. 2 root root   78 Jul 20 05:04 187a15b6-d425-46a8-a4a2-e78b65e008b6
drwxr-xr-x. 2 root root 4096 Jul 20 06:47 d0346cdd-5af6-4058-be86-1330f7ae09d1
drwxr-xr-x. 2 root root   79 Jul 20 17:09 f85c8c6c-32c9-44a8-b649-b63fdb11a79a
drwxr-xr-x. 2 root root   67 Jul 20 18:09 179ed182-17e4-4f1f-a44d-a3b6c16cf323
drwxr-xr-x. 2 root root   68 Jul 20 18:13 426cb05f-b1ee-43ce-862d-5bb4049cc957
drwxr-xr-x. 2 root root   68 Jul 20 18:13 1d2eec9d-f4d8-4325-9eb1-7d96d23e30fc
drwxr-xr-x. 2 root root   68 Jul 20 18:13 02f62a2f-3f59-46a7-9f5f-1656b8721512
drwxr-xr-x. 2 root root   68 Jul 20 18:14 c7417be9-473e-49da-b6d0-d1ab8fb4b1fc
drwxr-xr-x. 2 root root   68 Jul 20 18:17 b4d2077b-c7a9-46e7-9d39-d1281fba9baf
drwxr-xr-x. 2 root root   68 Jul 20 18:35 21972890-3d45-4642-b41d-c5fadfeba21a
drwxr-xr-x. 2 root root   80 Jul 20 19:17 d8b1b54c-7fc1-4ea6-83a5-0e56ff3b67a8
drwxr-xr-x. 2 root root   80 Jul 20 19:17 23a3cc35-4392-40bf-91e6-65c62d973753
drwxr-xr-x. 2 root root   80 Jul 20 19:17 7e831ef9-c932-4b89-8c81-33a45ad82b89
drwxr-xr-x. 2 root root   80 Jul 20 19:18 49ea0917-f9f4-4f5d-82d9-b86570a02dad
drwxr-xr-x. 2 root root   80 Jul 20 19:18 21589a61-5893-4e30-a70e-55ad0dc2e93f
drwxr-xr-x. 2 root root   80 Jul 20 19:22 6ae6d136-7f87-4fc8-92b8-64cd542495bf
drwxr-xr-x. 2 root root 4096 Jul 20 19:46 1c6f4547-c57d-4dcc-a405-ec509306ee25
drwxr-xr-x. 2 root root   68 Jul 20 21:20 c6dcc98d-b45b-4904-a217-d25001275c85
drwxr-xr-x. 2 root root   68 Jul 20 21:40 ee58d5d6-8b61-4431-9f7f-8cab2c331637
drwxr-xr-x. 2 root root 4096 Jul 20 22:06 243cb0f8-5169-430d-a5d8-48008a00d5c7
drwxr-xr-x. 2 root root 4096 Jul 20 22:16 188d53da-f129-46d9-87b7-c876b1aea70c
```

On controller and compute nodes, all services are run within their respective Docker™ containers. To list the Docker containers in the node, execute the following:

```
[root@control-server-2 ~]# docker ps -a
CONTAINER ID        IMAGE                                                                  COMMAND
                  CREATED           STATUS          PORTS      NAMES
258b2ca1d46a        172.31.228.164:5000/mercury-rhel7-osp8/nova-scheduler:4780
"/usr/bin/my_init /no"   25 minutes ago   Up 25 minutes              novascheduler_4780
ffe70809bbe0        172.31.228.164:5000/mercury-rhel7-osp8/nova-novncproxy:4780
"/usr/bin/my_init /st"   25 minutes ago   Up 25 minutes              novanovncproxy_4780
12b92bcb9dc0        172.31.228.164:5000/mercury-rhel7-osp8/nova-consoleauth:4780
"/usr/bin/my_init /st"   26 minutes ago   Up 26 minutes


……
novaconsoleauth_4780
7295596f5167        172.31.228.164:5000/mercury-rhel7-osp8/nova-api:4780
"/usr/bin/my_init /no"   27 minutes ago   Up 27 minutes              novaapi_4780
```

To view the Docker logs of any container, execute the following on the corresponding host:

```
ls –l /var/log/<service_name>/<log_filename>
e.g. ls -l /var/log/keystone/keystone.log
```

To get into a specific container, execute the following:

```
[root@control-server-2 ~]# alias | grep container
     root@control-server-2 ~]# source /root/.bashrc
#execute the alias:
  [root@control-server-2 ~]# novaapi
novaapi_4761 [nova@control-server-2 /]$
 novaapi_4761 [nova@control-server-2 /]$ exit
exit
```

If the Docker status indicates a container is down (based on output of "docker ps –a"), collect the Docker service logs as well:

```
cd /etc/systemd/system/multi-user.target.wants/
ls docker* # get the corresponding service name from the output
systemctl status <service_name> -n 1000 > /root/filename # redirects the output to the file
```

For storage nodes running Ceph, execute the following to check the cluster status:

```
ceph –v # on monitor nodes (controller), show's ceph version

ceph –s # on monitor nodes (controller), show cluster status

ceph osd lspools #on monitor nodes (controller),list pools

ceph mon stat # summarize monitor status

ceph-disk list # on OSD / storage nodes; List disks, partitions, and Ceph OSDs

rbd list images # on monitor nodes (controller); dump list of image snapshots

rbd list volumes # on monitor nodes (controller); dump list of volumes
```

Logging Into the Kibana Dashboard

Kibana is an open source data visualization platform that you can use to explore Cisco VIM logs. To log into the Kibana dashboard:

Step 1 Using a terminal client, use SSH to log into your management node and enter the password to login. In the example below, the management node has an IP address of 17.0.0.2

```
# ssh root@17.0.0.2
root@17.0.0.2's password
```

Step 2 In the SSH terminal session, locate the line containing ELK_PASSWORD in /root/installer-{tag id}/openstack-configs/secrets.yaml. Note the value of the ELK_PASSWORD. It will be used in Step 4.

```
cat /root/installer-{tag-id}/openstack-configs/secrets.yaml
...
ELK_PASSWORD: <note this value>
...
```

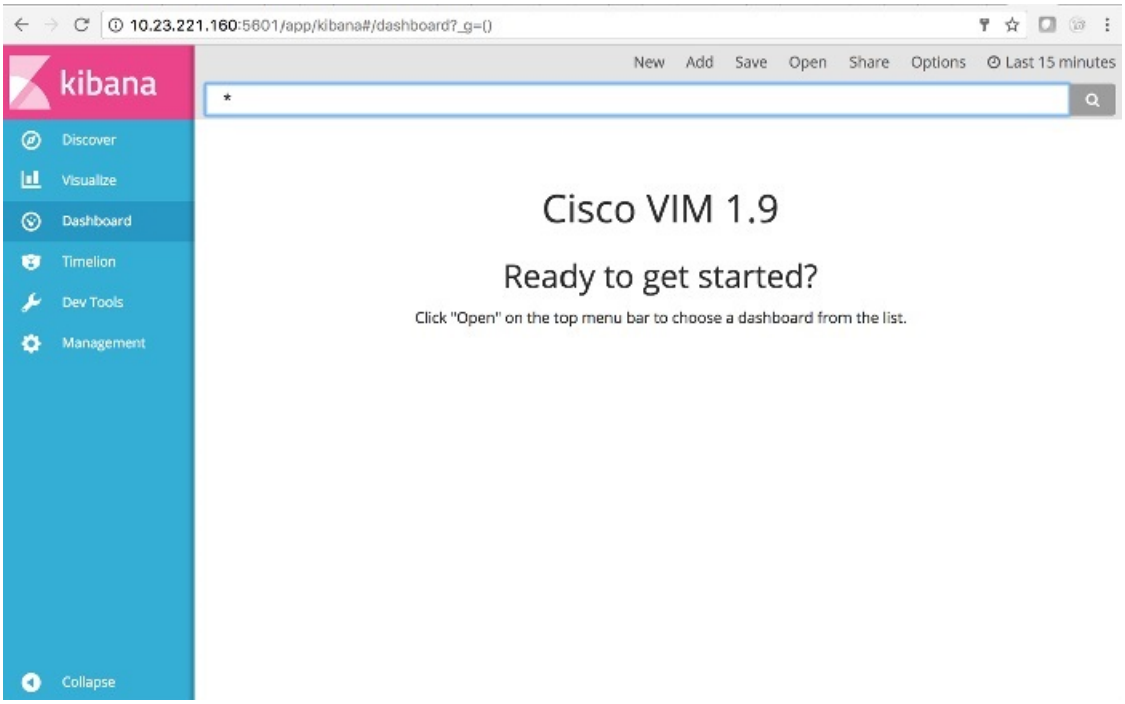
Step 3 Using your web browser, go to http://<management_node_ip_address>:5601.

Step 4 When prompted, log in with the following credentials:

User Name: admin

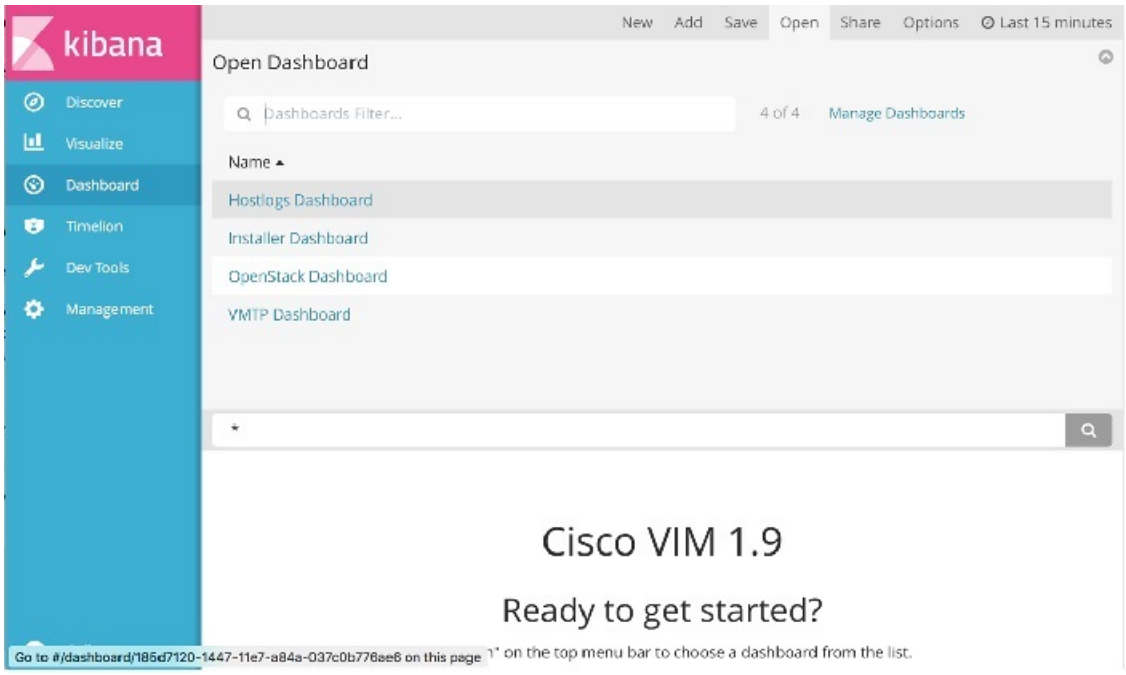
Password: <value of ELK_PASSWORD from Step 2>

The Kibana dashboard appears allowing you to display the Cisco VIM service and installer logs.

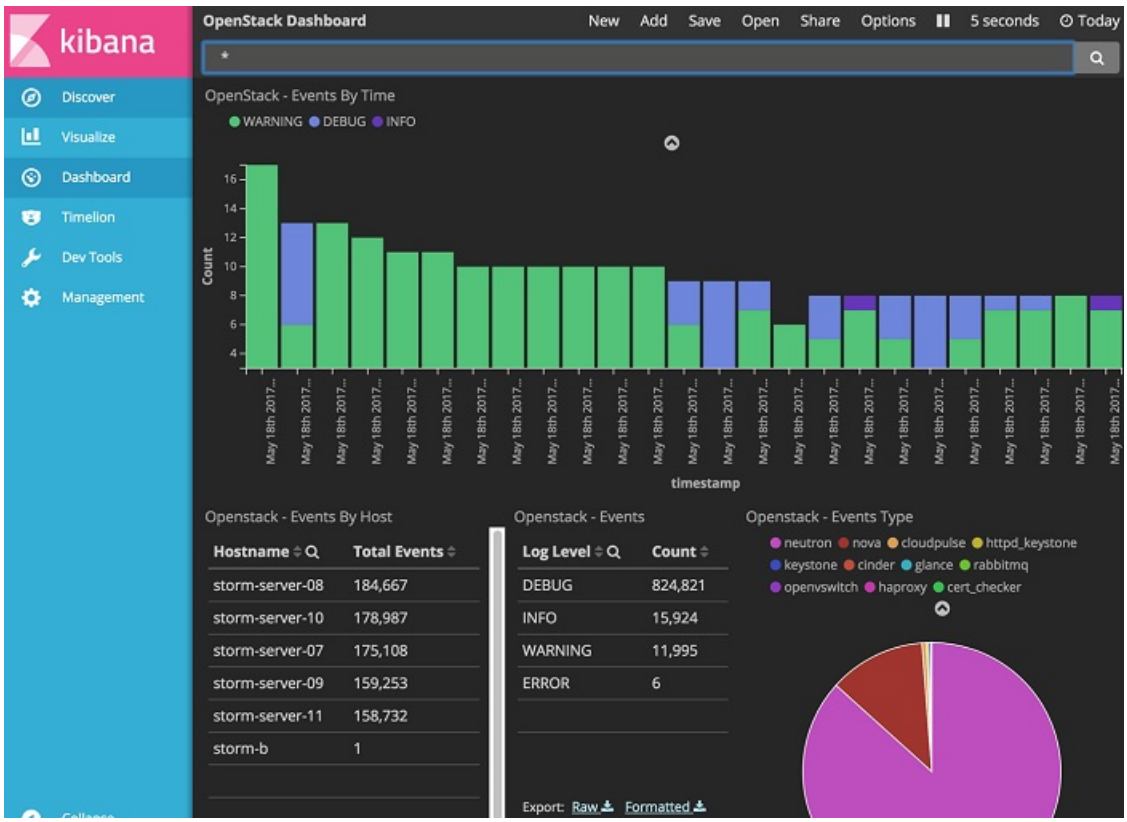


Note In the figure, the Cisco VIM version is displayed as 1.9. But, in your environment, the Cisco VIM version will appear as 2.0 in the Kibana Dashboard.

Step 5 Navigate to the dashboards by clicking **Open** menu bar and choose the dashboard. It is not recommended to use visualize/Timelion/DevTools or Management options on the left side.

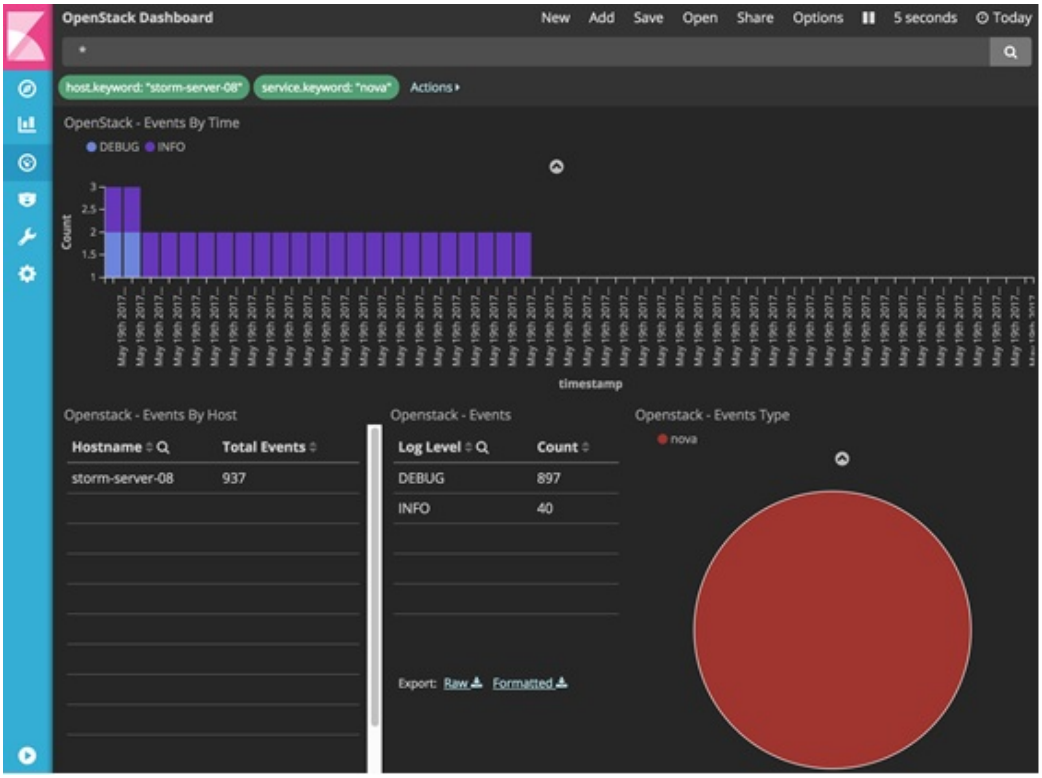


For example, if you click on OpenStack dashboard, the following screen appears.



Note You can navigate from dashboard to the other by selecting the appropriate dashboard from the right top bar menu.

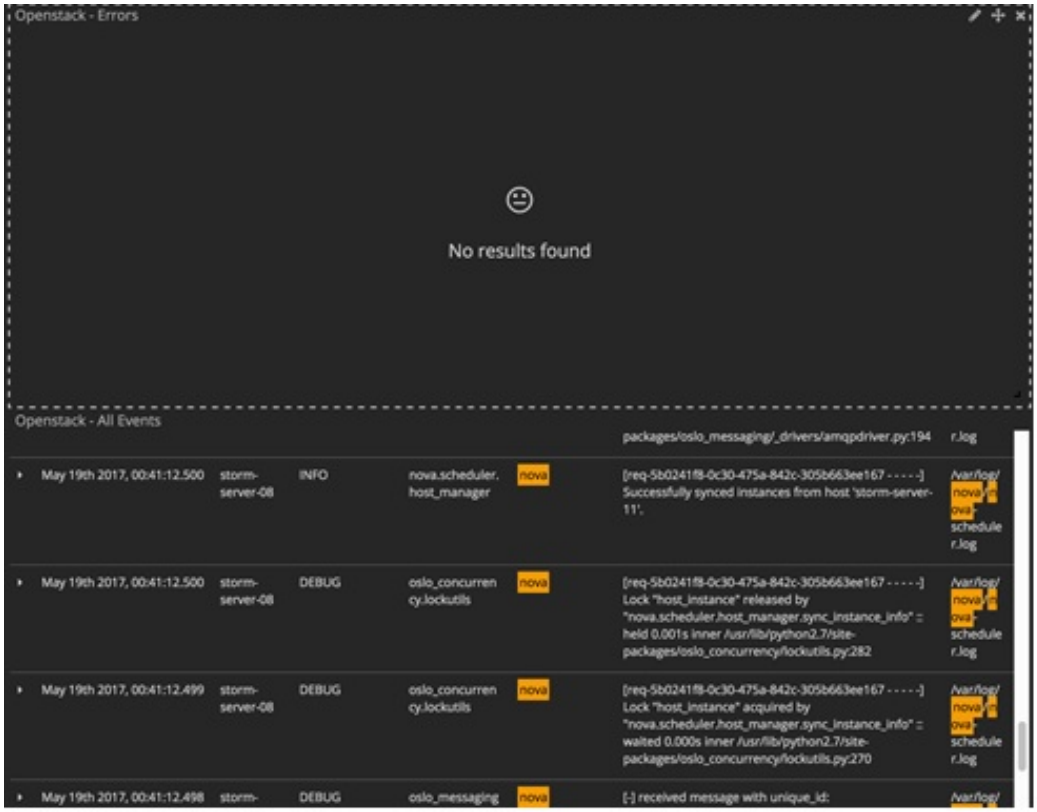
For more information on using Kibana, see the *OpenStack Kibana documentation*.



Step 6 Scroll down on the page to see the Errors and All Events sections.

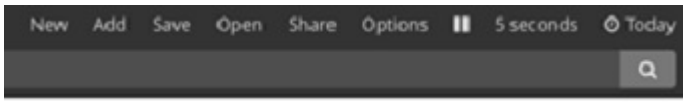
If there are no errors, the No Results Found message will be displayed.

Figure 6: OpenStack Errors



Note Cisco VIM stores the OpenStack logs in Elasticsearch and its snapshot indicates the location of the data stored. This data is rotated on a period basis and you may not see older data in Kibana, if it has been already rotated out and/or deleted.

Note Logs are updated in kibana as they get updated in Elasticsearch simultaneously. You can temporarily stop the visualization to debug any operation on Kibana by clicking the pause button on the top navigation bar and restore it later by clicking the same button.



Rotation of the Cisco VIM Logs

Cisco VIM stores all logs in Elasticsearch. Elasticsearch indices are rotated on a periodic basis to prevent disk space overflow by creating snapshots. Snapshots are defined in openstack_config.yaml as listed below:

```
# vi ~/openstack-configs/openstack_config.yaml
…
# Elk rotation parameters
elk_rotation_frequency: "monthly"  # Available: "daily", "weekly", "fortnightly", "monthly"
```

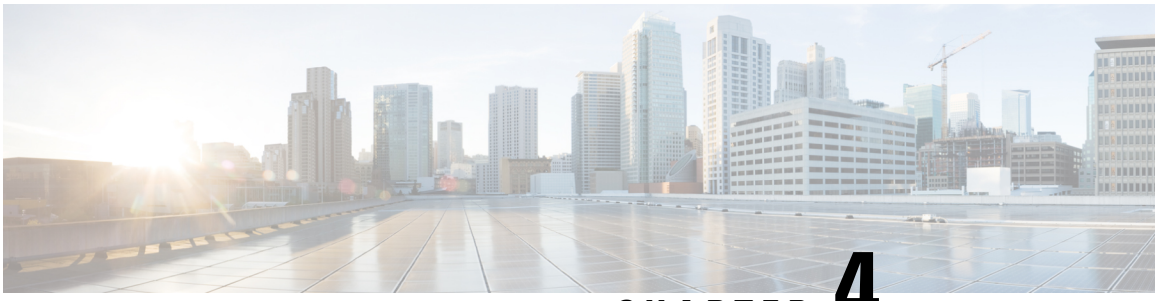
```
elk_rotation_size: 2                # Unit is in Gigabytes (float is allowed)
elk_rotation_del_older: 10         # Delete older than 10 units (where units depends on the
 value set on elk_rotation_frequency)
…
```

You can change the frequency of the rotation by changing the values. For more information on how to set the Elasticsearch parameters via VIM API/CLI, please refer to the section *Reconfiguring Passwords and OpenStack Configurations*.

Cisco VIM uses the open sourceElasticsearch Curator tool to manage the Elasticsearch indices and snapshots. For more information about Elasticsearch handles snapshots, please look at the official information on Elastic.co (version 5.2).

Network Performance Test with NFVBench

NFVBench is a network performance benchmarking tool integrated with Cisco VIM. For more details, refer to NFVBench section of Chapter 1 in the admin guide for details.



CHAPTER 4

Managing Cisco NFVI Security

The following topics describe Cisco NFVI network and application security and best practices.

Verifying Management Node Network Permissions

The Cisco NFVI management node stores sensitive information related to Cisco NFVI operations. Access to the management node can be restricted to requests coming from IP addresses known to be used by administrators. The administrator source networks is configured in the setup file, under **[NETWORKING]** using the **admin_source_networks** parameter. To verify this host based firewall setting, log into the management node as an admin user and list the rules currently enforces by iptables. Verify that the source networks match the values configured. If no source networks have been configured, then all source traffic is allowed. However, note that only traffic destined to ports with known admin services is allowed to pass. The **admin_source_networks** value can be set at install time or changed through a reconfigure.

```
[root@j11-control-server-1 ~]# iptables -list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     icmp --  anywhere             anywhere
ACCEPT     tcp  --  10.0.0.0/8           anywhere            tcp dpt:ssh
ACCEPT     tcp  --  172.16.0.0/12        anywhere            tcp dpt:ssh
ACCEPT     tcp  --  10.0.0.0/8           anywhere            tcp dpt:https
ACCEPT     tcp  --  172.16.0.0/12        anywhere            tcp dpt:https
ACCEPT     tcp  --  10.0.0.0/8           anywhere            tcp dpt:4979
ACCEPT     tcp  --  172.16.0.0/12        anywhere            tcp dpt:4979
ACCEPT     tcp  --  10.0.0.0/8           anywhere            tcp dpt:esmagent
```

```
ACCEPT     tcp  --  172.16.0.0/12        anywhere             tcp dpt:esmagent
ACCEPT     tcp  --  10.0.0.0/8           anywhere             tcp dpt:8008
ACCEPT     tcp  --  172.16.0.0/12        anywhere             tcp dpt:8008
ACCEPT     tcp  --  10.0.0.0/8           anywhere             tcp dpt:copy
ACCEPT     tcp  --  172.16.0.0/12        anywhere             tcp dpt:copy
ACCEPT     tcp  --  10.0.0.0/8           anywhere             tcp dpt:22250
ACCEPT     tcp  --  172.16.0.0/12        anywhere             tcp dpt:22250
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED
DROP       all  --  anywhere             anywhere
```

Verifying Management Node File Permissions

The Cisco NFVI management node stores sensitive information related to Cisco NFVI operations. These files are secured by strict file permissions. Sensitive files include secrets.yaml, openrc, *.key, and *.pem. To verify the file permissions, log into the management node as an admin user and list all of the files in the ***~/openstack-configs/*** directory. Verify that only the owner has read and write access to these files. For example:

```
[root@j11-control-server-1 ~]# ls -l ~/openstack-configs
total 172
-rw-------. 1 root root  3272 Jun 21 17:57 haproxy.key
-rw-------. 1 root root  5167 Jun 21 17:57 haproxy.pem
-rw-------. 1 root root   223 Aug  8 18:09 openrc
-rw-------. 1 root root   942 Jul  6 19:44 secrets.yaml

[…]
```

Viewing Administrator Access Attempts

Because the UCS servers are part of the critical Cisco NFVI infrastructure, Cisco recommends monitoring administrator login access periodically. To view the access attempts, use the **journalctl** command to view the log created by sshd. For example:

```
[root@control-server-1 ~]# journalctl –u sshd
-- Logs begin at Tue 2016-06-21 17:39:35 UTC, end at Mon 2016-08-08 17:25:06 UTC. --
Jun 21 17:40:03 hh23-12 systemd[1]: Started OpenSSH server daemon.
Jun 21 17:40:03 hh23-12 systemd[1]: Starting OpenSSH server daemon...
Jun 21 17:40:03 hh23-12 sshd[2393]: Server listening on 0.0.0.0 port 22.
Jun 21 17:40:03 hh23-12 sshd[2393]: Server listening on :: port 22.
Jun 21 17:40:43 hh23-12 sshd[12657]: Connection closed by 171.70.163.201 [preauth]
Jun 21 17:41:13 hh23-12 sshd[12659]: Accepted password for root from 171.70.163.201 port
40499
Jun 21 17:46:41 hh23-12 systemd[1]: Stopping OpenSSH server daemon...
Jun 21 17:46:41 hh23-12 sshd[2393]: Received signal 15; terminating.
Jun 21 17:46:41 hh23-12 systemd[1]: Started OpenSSH server daemon.
Jun 21 17:46:41 hh23-12 systemd[1]: Starting OpenSSH server daemon...
Jun 21 17:46:41 hh23-12 sshd[13930]: Server listening on 0.0.0.0 port 22.
Jun 21 17:46:41 hh23-12 sshd[13930]: Server listening on :: port 22.
Jun 21 17:50:45 hh23-12 sshd[33964]: Accepted password for root from 171.70.163.201 port
40545
Jun 21 17:56:36 hh23-12 sshd[34028]: Connection closed by 192.168.212.20 [preauth]
Jun 21 17:57:08 hh23-12 sshd[34030]: Accepted publickey for root from 10.117.212.20 port
62819
Jun 22 16:42:40 hh23-12 sshd[8485]: Invalid user user1 from 10.117.212.20
Jun 22 16:42:40 hh23-12 sshd[8485]: input_userauth_request: invalid user user1 [preauth]
s
```

Verifying SELinux

To minimize the impact of a security breach on a Cisco NFVI server, the Cisco VM enables SELinux (Security Enhanced Linux) to protect the server resources. To validate that SELinux is configured and running in enforcing mode, use the **sestatus** command to view the status of SELinux and verify that its status is enabled and in enforcing mode. For example:

```
[root@mgmt1 ~]# /usr/sbin/sestatus -v
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          permissive
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Max kernel policy version:      28
```

Validating Port Listening Services

To prevent access by unauthorized users and processes, Cisco NFVI has no extra services listening on network ports. To verify this, use the netstat -plnt command to get a list of all services listening on the node and verify that no unauthorized services are listening. For example:

```
[root@j11-control-server-1 ~]# netstat -plnt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address     State       PID/Program
name
tcp        0      0 23.23.4.101:8776        0.0.0.0:*           LISTEN      24468/python2
tcp        0      0 23.23.4.101:5000        0.0.0.0:*           LISTEN      19874/httpd
tcp        0      0 23.23.4.101:5672        0.0.0.0:*           LISTEN      18878/beam.smp

tcp        0      0 23.23.4.101:3306        0.0.0.0:*           LISTEN      18337/mysqld
tcp        0      0 127.0.0.1:11211         0.0.0.0:*           LISTEN      16563/memcached
tcp        0      0 23.23.4.101:11211       0.0.0.0:*           LISTEN      16563/memcached
tcp        0      0 23.23.4.101:9292        0.0.0.0:*           LISTEN      21175/python2
tcp        0      0 23.23.4.101:9999        0.0.0.0:*           LISTEN      28555/python
tcp        0      0 23.23.4.101:80          0.0.0.0:*           LISTEN      28943/httpd
tcp        0      0 0.0.0.0:4369            0.0.0.0:*           LISTEN      18897/epmd

tcp        0      0 127.0.0.1:4243          0.0.0.0:*           LISTEN      14673/docker

tcp        0      0 0.0.0.0:22              0.0.0.0:*           LISTEN      2909/sshd

tcp        0      0 23.23.4.101:4567        0.0.0.0:*           LISTEN      18337/mysqld

tcp        0      0 23.23.4.101:15672       0.0.0.0:*           LISTEN      18878/beam.smp

tcp        0      0 0.0.0.0:35672           0.0.0.0:*           LISTEN      18878/beam.smp

tcp        0      0 127.0.0.1:25            0.0.0.0:*           LISTEN      4531/master

tcp        0      0 23.23.4.101:35357       0.0.0.0:*           LISTEN      19874/httpd

tcp        0      0 23.23.4.101:8000        0.0.0.0:*           LISTEN      30505/python

tcp        0      0 23.23.4.101:6080        0.0.0.0:*           LISTEN      27996/python2
```

```
tcp        0      0 23.23.4.101:9696        0.0.0.0:*               LISTEN      22396/python2
tcp        0      0 23.23.4.101:8004        0.0.0.0:*               LISTEN      30134/python
tcp        0      0 23.23.4.101:8773        0.0.0.0:*               LISTEN      27194/python2
tcp        0      0 23.23.4.101:8774        0.0.0.0:*               LISTEN      27194/python2
tcp        0      0 23.23.4.101:8775        0.0.0.0:*               LISTEN      27194/python2
tcp        0      0 23.23.4.101:9191        0.0.0.0:*               LISTEN      20752/python2
tcp6       0      0 :::9200                 :::*                    LISTEN      18439/xinetd
tcp6       0      0 :::4369                 :::*                    LISTEN      18897/epmd
tcp6       0      0 :::22                   :::*                    LISTEN      2909/sshd
tcp6       0      0 ::1:25                  :::*                    LISTEN      4531/master
```

Validating Non-Root Users for OpenStack Services

To prevent unauthorized access, Cisco NFVI runs OpenStack processes as a non-root user. To verify OpenStack processes are not running as root, use the **ps** command to get a list of all node processes. In the following example the user is 162:

```
[root@j11-control-server-1 ~]# ps -aux | grep nova-api
162      27194  0.6  0.0 360924 132996 ?       S    Aug08  76:58 /usr/bin/python2
/usr/bin/nova-api
162      27231  0.0  0.0 332192 98988 ?        S    Aug08   0:01 /usr/bin/python2
/usr/bin/nova-api
162      27232  0.0  0.0 332192 98988 ?        S    Aug08   0:01 /usr/bin/python2
/usr/bin/nova-api
162      27233  0.0  0.0 332192 98988 ?        S    Aug08   0:01 /usr/bin/python2
/usr/bin/nova-api
```

Verifying Password Strength

Password strength is critical to Cisco NFVI security. Cisco NFVI passwords can be generated in one of two ways during installation:

- The Cisco NFVI installer generates unique passwords automatically for each protected service.
- You can provide an input file containing the passwords you prefer.

Cisco-generated passwords will be unique, long, and contain a mixture of uppercase, lowercase, and numbers. If you provide the passwords, password strength will be your responsibility. You can view the passwords by displaying the secrets.yaml file. For example:

```
[root@mgmt1 ~]# cat ~/openstack-configs/secrets.yaml
ADMIN_USER_PASSWORD: QaZ12n13wvvNY7AH
CINDER_DB_PASSWORD: buJL8pAfytoJ0Icm
CINDER_KEYSTONE_PASSWORD: AYbcB8mx6a5Ot549
CLOUDPULSE_KEYSTONE_PASSWORD: HAT6vbl7Z56yZLtN
COBBLER_PASSWORD: bax8leYFyyDon0ps
CPULSE_DB_PASSWORD: aYGSzURpGChztbMv
DB_ROOT_PASSWORD: bjb3Uvwus6cvaNe5
```

```
ELK_PASSWORD: c50e57Dbm7LF0dRV
[…]
```

Reconfiguring Passwords and OpenStack Configurations

Note This topic does not apply if you installed the optional Cisco Virtual Topology System. For information about use of passwords when VTS is installed, see the *Installing Cisco VTS* section in the *Cisco NFV Infrastructure 2.0 Installation Guide*.

You can reset some configurations after installation including the OpenStack service password and debugs, TLS certificates, ELK configurations, and collectd intervals. Two files, secrets.yaml and openstack_config.yaml, located in : /root/installer-{tag id}/openstack-configs/, contain the passwords, debugs, TLS file location, ELK and collectd configurations. Also, Elasticsearch uses disk space for the data that is sent to it. These files can grow in size, and Cisco VIM has configuration variables that establishes the frequency and file size under which they will be rotated.

The Cisco VIM installer dynamically generates the OpenStack service and database passwords with 16 alphanumeric characters and stores those in /root/openstack-configs/secrets.yaml. You can change the OpenStack service and database passwords using the password reconfigure command on the deployed cloud. The command identifies the containers affected by the password change and restarts them so the new password can take effect. Always schedule password reconfigurations in a maintenance window because container restarts might disrupt the control plane. You can list the password and configuration that can be changed using following:

```
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 installer-xxxx]# ciscovimclient/ciscovim help reconfigure
usage: ciscovim reconfigure [--regenerate_secrets] [--setpassword <secretkey>]
                            [--setopenstackconfig <option>]

Reconfigure the openstack cloud
Optional arguments:
  --regenerate_secrets           Regenerate All Secrets
  --setpassword <secretkey>      Set of secret keys to be changed.
  --setopenstackconfig <option>  Set of Openstack config to be changed.
[root@mgmt1 ~]#ciscovimclient/ciscovim list-openstack-configs
+-------------------------------+----------------------------------------+
|              Name             |                 Option                 |
+-------------------------------+----------------------------------------+
|      CINDER_DEBUG_LOGGING     |                 False                  |
|     KEYSTONE_DEBUG_LOGGING    |                 False                  |
|   CLOUDPULSE_VERBOSE_LOGGING  |                  True                  |
|     MAGNUM_VERBOSE_LOGGING    |                  True                  |
|       NOVA_DEBUG_LOGGING      |                  True                  |
|    NEUTRON_VERBOSE_LOGGING    |                  True                  |
|      external_lb_vip_cert     |  /root/openstack-configs/haproxy.pem   |
|     GLANCE_VERBOSE_LOGGING    |                  True                  |
| COLLECTD_RECONFIGURE_interval |                   30                   |
|     elk_rotation_frequency    |                monthly                 |
|   CEILOMETER_VERBOSE_LOGGING  |                  True                  |
|     elk_rotation_del_older    |                   10                   |
|       HEAT_DEBUG_LOGGING      |                 False                  |
|    KEYSTONE_VERBOSE_LOGGING   |                  True                  |
|     external_lb_vip_cacert    | /root/openstack-configs/haproxy-ca.crt |
|      MAGNUM_DEBUG_LOGGING     |                  True                  |
|     CINDER_VERBOSE_LOGGING    |                  True                  |
|       elk_rotation_size       |                   2                    |
```

```
|    CLOUDPULSE_DEBUG_LOGGING   |                 False                 |
|     NEUTRON_DEBUG_LOGGING     |                  True                 |
|      HEAT_VERBOSE_LOGGING     |                  True                 |
|    CEILOMETER_DEBUG_LOGGING   |                 False                 |
|      GLANCE_DEBUG_LOGGING     |                 False                 |
|      NOVA_VERBOSE_LOGGING     |                  True                 |
+-------------------------------+---------------------------------------+
[root@mgmt1 installer-xxxx]#
[root@mgmt1 installer-xxxx]# ciscovimclient/ciscovim list-password-keys
+----------------------------------+
| Password Keys                    |
+----------------------------------+
| COBBLER_PASSWORD                 |
| CPULSE_DB_PASSWORD               |
| DB_ROOT_PASSWORD                 |
| ELK_PASSWORD                     |
| GLANCE_DB_PASSWORD               |
| GLANCE_KEYSTONE_PASSWORD         |
| HAPROXY_PASSWORD                 |
| HEAT_DB_PASSWORD                 |
| HEAT_KEYSTONE_PASSWORD           |
| HEAT_STACK_DOMAIN_ADMIN_PASSWORD |
| HORIZON_SECRET_KEY               |
| KEYSTONE_ADMIN_TOKEN             |
| KEYSTONE_DB_PASSWORD             |
| METADATA_PROXY_SHARED_SECRET     |
| NEUTRON_DB_PASSWORD              |
| NEUTRON_KEYSTONE_PASSWORD        |
| NOVA_DB_PASSWORD                 |
| NOVA_KEYSTONE_PASSWORD           |
| RABBITMQ_ERLANG_COOKIE           |
| RABBITMQ_PASSWORD                |
| WSREP_PASSWORD                   |
+----------------------------------+
[root@mgmt1 installer-xxxx]#
```

You can change specific password and configuration identified from the available list. The password and configuration values can be supplied on the command line as follows:

```
[root@mgmt1 ~]# ciscovimclient/ciscovim help reconfigure
usage: ciscovim reconfigure [--regenerate_secrets] [--setpassword <secretkey>]
                            [--setopenstackconfig <option>]

Reconfigure the Openstack cloud

Optional arguments:
  --regenerate_secrets           Regenerate All Secrets
  --setpassword <secretkey>      Set of secret keys to be changed.
  --setopenstackconfig <option>  Set of Openstack config to be changed.
[root@mgmt1 ~]# ciscovimclient/ciscovim reconfigure --setpassword
ADMIN_USER_PASSWORD,NOVA_DB_PASSWORD --setopenstackconfig
HEAT_DEBUG_LOGGING,HEAT_VERBOSE_LOGGING
Password for ADMIN_USER_PASSWORD:
Password for NOVA_DB_PASSWORD:
Enter T/F for option HEAT_DEBUG_LOGGING:T
Enter T/F for option HEAT_VERBOSE_LOGGING:T
```

The supplied password must be alphanumeric chars and can be maximum of 32 characters in length. Below are the available configuration parameters for OpenStack:

Configuration Parameter	Allowed Values
CEILOMETER_DEBUG_LOGGING	T/F (True or False)

CEILOMETER_VERBOSE_LOGGING	T/F (True or False)
CINDER_DEBUG_LOGGING	T/F (True or False)
CINDER_VERBOSE_LOGGING	T/F (True or False)
CLOUDPULSE_DEBUG_LOGGING	T/F (True or False)
CLOUDPULSE_VERBOSE_LOGGING	T/F (True or False)
GLANCE_DEBUG_LOGGING	T/F (True or False)
GLANCE_VERBOSE_LOGGING	T/F (True or False)
HEAT_DEBUG_LOGGING	T/F (True or False)
HEAT_VERBOSE_LOGGING	T/F (True or False)
KEYSTONE_DEBUG_LOGGING	T/F (True or False)
KEYSTONE_VERBOSE_LOGGING	T/F (True or False)
MAGNUM_DEBUG_LOGGING	T/F (True or False)
MAGNUM_VERBOSE_LOGGING	T/F (True or False)
NEUTRON_DEBUG_LOGGING	T/F (True or False)
NEUTRON_VERBOSE_LOGGING	T/F (True or False)
NOVA_DEBUG_LOGGING	T/F (True or False)
NOVA_VERBOSE_LOGGING	T/F (True or False)
COLLECTD_RECONFIGURE_interval	Collectd metric gathering interval (seconds)
elk_rotation_del_older	Days after which older logs will be purged
elk_rotation_frequency	Available options: "daily", "weekly", "fortnightly", "monthly"
elk_rotation_size	Gigabytes (entry of type float/int is allowed)
external_lb_vip_cacert	Location of HAProxy CA certificate
external_lb_vip_cert	Location of HAProxy certificate

Alternatively, you can dynamically regenerate all passwords using regenerate_secrets command option as follows:

```
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim reconfigure --regenerate_secrets
```

In addition to the services passwords, you can change the debug and verbose options for Heat, Glance, Cinder, Nova, Neutron, Keystone and Cloudpulse in /root/openstack-configs/openstack_config.yaml. Other configurations you can modify include ELK configuration parameters, collectd intervals, API and Horizon TLS certificates, and RootCA. , and admin source networks. When reconfiguring these options (for example

TLS), always remember that some control plane downtime will occur, so plan the changes during maintenance windows. The command to reconfigure these elements is:

```
./ciscovimclient/ciscovim reconfigure
```

The command includes a built-in validation to ensure you do not enter typos in the secrets.yaml or openstack_config.yaml files.

When reconfiguration of password or enabling of openstack-services fails, all subsequent pod management operations will be blocked. In this case, it is recommended to contact Cisco TAC to resolve the situation.

Enabling NFVIMON Post Pod Install

Cisco VIM 2.0 can be optionally installed with a 3rd party software (aka NFVIMON), that can monitor the health and performance of the NFV infrastructure. The NFVIMON feature enables extensive monitoring and performance data for various components of the cloud infrastructure including Cisco UCS blade and rack servers, service profiles, Nexus top of rack switches, fabric connections and also the OpenStack instances. The monitoring system is designed such that it can monitor single or multiple Pods from a single management system. NFVIMON can be enabled by extending the setup_data.yaml with relevant information on an existing pod, via the reconfigure option. NFVIMON consists of 4 components: dispatcher, collector, Resource Manager (RM) and control-center with Cisco Zenpacks (CC). Since NVIFMON is a 3rd party software, care has been taken to make sure its integration into VIM is loosely coupled and the VIM automation only deals with installing the minimal software piece (dispatcher) needed to monitor the pod. The installing of the other NFVIMON components (collector, Resource Manager (RM) and control-center with Cisco Zenpacks (CC)), are Cisco Advance Services led activity and those steps are outside the scope of the current install guide.

Before you Begin

Please ensure that you have engaged with Cisco Advance Services on the planning and installation of the NFVIMON accessories along with its network requirements. Also, the image information of collector, Resource Manager (RM) and control-center with Cisco Zenpacks (CC)) is available only through Cisco Advance Services. At a high level, please have a node designated to host a pair of collector VM for each pod, and a common node to host CC and RM VMs, which can aggregate and display monitoring information from multiple pods. In terms of networking, the collectors VMs need to have 2 interfaces: an interface in br_mgmt of the VIM, and another interface that is routable, which can reach the VIM Installer REST API and the RM VMs. Since the collector VM is sitting in an independent node, 4 IPs from the management network of the pod should be pre-planned and reserved. Install steps of the collector, Resource Manager (RM) and control-center with Cisco Zenpacks (CC)) are Cisco Advance Services led activity.

Installation of NFVIMON Dispatcher

The dispatcher is the only component in NFVIMON offering that is managed by VIM orchestrator. While the dispatcher acts as a conduit to pass openstack information of the pod to the collectors, it is the Cisco NFVI Zenpack sitting in the CC/RM node, that gathers the node level information. To enable dispatcher as part of the VIM Install, update the setup_data with the following information:

```
#Define the PODNAME
PODNAME: <PODNAME with no space>; ensure that this is unique across all the pods
NFVIMON:
  MASTER:                  # Master Section
    admin_ip: <IP address of Control Centre VM>
  COLLECTOR:               # Collector Section
```

```
management_vip: <VIP for ceilometer/dispatcher to use> #Should be unique across the VIM
Pod; Should be part of br_mgmt network
    Collector_VM_Info:
      -
        hostname: <hostname of Collector VM 1>
        password: <password_for_collector_vm1>  # max length of 32
       ccuser_password: <password from master for 'ccuser' (to be used for self monitoring)>
  # max length of 32
        admin_ip: <ssh_ip_collector_vm1> # Should be part of br_api network
        management_ip: <mgmt_ip_collector_vm1> # Should be part of br_mgmt network
      -
        hostname: <hostname of Collector VM 2>
        password: <password_for_collector_vm2>  # max length of 32
       ccuser_password: <password from master for 'ccuser' (to be used for self monitoring)>
  # max length of 32
        admin_ip: <ssh_ip_collector_vm2> # Should be part of br_api network
        management_ip: <mgmt_ip_collector_vm2> # Should be part of br_mgmt network
  DISPATCHER:
    rabbitmq_username: admin  # Pod specific user for dispatcher module in
ceilometer-collector
```

To monitor TOR, ensure that the following TORSWITCHINFO sections are defined in the setup_data.yaml.

```
TORSWITHCINFO:
  SWITCHDETAILS:
 -
      hostname: <switch_a_hostname>:     # Mandatory for NFVIMON if switch monitoring is
needed
      username: <TOR switch username>    # Mandatory for NFVIMON if switch monitoring is
needed
      password: <TOR switch password>    # Mandatory for NFVBENCH; Mandatory for NFVIMON
if switch monitoring is needed
      ssh_ip: <TOR switch ssh ip>        # Mandatory for NFVIMON if switch monitoring is
needed
      ....
  -
      hostname: <switch_b_hostname>:     # Mandatory for NFVIMON if switch monitoring is
needed
      username: <TOR switch username>    # Mandatory for NFVIMON if switch monitoring is
needed
      password: <TOR switch password>    # Mandatory for NFVIMON if switch monitoring is
needed
      ssh_ip: <TOR switch ssh ip>        # Mandatory for NFVIMON if switch monitoring is
needed
      ....
```

To initiate the integration of NFVIMON on an existing pod, copy the setupdata into a local dir and update it manually with information listed above, then run reconfiguration command as follows:

```
[root@mgmt1 ~]# cd /root/
[root@mgmt1 ~]# mkdir MyDir
[root@mgmt1 ~]# cd MyDir
[root@mgmt1 ~]# cp /root/openstack-configs/setup_data.yaml <my_setup_data.yaml>
[root@mgmt1 ~]# vi my_setup_data.yaml (update the setup_data to include NFVIMON related
info)
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim --setupfile ~/MyDir/<my_setup_data.yaml> reconfigure
```

It should be noted that un-configuration of this feature is not supported today. Additionally, NFVIMON is only supported on a pod running keystone v2.

Fernet Key Operations

Keystone fernet token format is based on the cryptographic authentication method - Fernet. Fernet is an implementation of Symmetric Key Encryption. Symmetric key encryption is a cryptographic mechanism that uses the same cryptographic key to encrypt plaintext and the same cryptographic key to decrypt ciphertext. Fernet authentication method also supports multiple keys where it takes a list of symmetric keys, performs all encryption using the first key in a list and attempts to decrypt using all the keys from that list.

The Cisco NFVI pods uses Fernet keys by default. The following operations can be carried out in Cisco NFVI pods.

To check if the fernet keys are successfully synchronized across the keystone nodes.

```
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim help check-fernet-keys
usage: ciscovim check-fernet-keys

Check whether the fernet keys are successfully synchronized across keystone nodes.
```

To set the fernet key frequency:

```
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim help period-rotate-fernet-keys
usage: ciscovim period-rotate-fernet-keys <SET_PERIOD_ROTATION_FERNET_KEYS>
Set the frequency of fernet keys rotation on keystone
Positional arguments:
  <SET_PERIOD_ROTATION_FERNET_KEYS>
Frequency to set for period rotation
```

To forcefully rotate the fernet keys:

```
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim help rotate-fernet-keys
usage: ciscovim rotate-fernet-keys
Trigger rotation of the fernet keys on keystone
```

To resync the fernet keys across the keystone nodes:

```
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim help resync-fernet-keys
usage: ciscovim resync-fernet-keys
Resynchronize the fernet keys across all the keystone nodes
```

Managing Certificates

When TLS protection is configured for the OpenStack APIs, the two certificate files, haproxy.pem and haproxy-ca.crt, are stored in the /root/openstack-configs/ directory. Clients running on servers outside of the deployed cloud to verify cloud authenticity need a copy of the root certificate (haproxy-ca.crt). If a well-known certificate authority has signed the installed certificate, no additional configuration is needed on client servers. However, if a self-signed or local CA is used, copy haproxy-ca.crt to each client. Following instructions specific to the client operating system or browser to install the certificate as a trusted certificate.

Alternatively, you can explicitly reference the certificate when using the OpenStack CLI by using the environment variable OS_CACERT or command line parameter –cacert.

While Cisco NFVI is operational, a daily check is made to monitor the expiration dates of the installed certificates. If certificates are not nearing expiration, an informational message is logged. As the certificate approaches expiration, an appropriate warning or critical message is logged.

```
2017-04-24T13:56:01 INFO Certificate for OpenStack Endpoints at 192.168.0.2:5000 expires
in 500 days
```

It is important to replace the certificates before they expire. After Cisco NFVI is installed, you can update the certificates by replacing the haproxy.pem and haproxy-ca.crt files and running the reconfigure command:

```
cd ~/installer-xxxx; ./ciscovimclient/ciscovim reconfigure
```

Reconfiguring TLS Certificates

Cisco VIM provides a way to configure TLS certificates on-demand for any reason. For Example: certificate expiration policies governing certificate management.

Reconfiguration of certificates in general is supported in the following components:

- Cisco VIM Rest API endpoints:
  - Steps to be performed to reconfigure certificate and key file are as follows:
  - Operator will copy the new key, CA root and certificate files into the ~/openstack-configs folder under the following filenames

    ```
    cp <new-ca-root-cert> ~/openstack-configs/mercury-ca.crt
    cp <new-key-file> ~/openstack-configs/mercury.key
    cp <new-cert-file> ~/openstack-configs/mercury.crt
    ```

  - Once copied run the reconfigure steps as under:

    ```
    cd ~/installer-xxxx/tools
    ./restapi.py -a reconfigure-tls
    ```

- SwiftStack Service through Horizon and CinderBackup Service.
  - Reconfiguring TLS certificates for SwiftStack mainly involves client side certificate updates. The CA root certificate in both these cases is updated for components within OpenStack that are clients of the SwiftStack service in general.
  - Copy the new CA root certificate to the ~/openstack-configs folder and run reconfigure.

    ```
    cp <new-ca-root-cert> ~/openstack-configs/haproxy-ca.crt
     cd ~/installer-xxxx; ./ciscovimclient/ciscovim reconfigure
    ```

- Logstash service and Logstash forwarder (client-side certificates).
  - For the Logstash service on the management node, both the key and certificate file will be reconfigured as part of the reconfigure operation.
  - For the Logstash forwarder service on the controllers, compute and storage nodes, the certificate file will be reconfigured as part of the reconfigure operation.
  - Copy of the key and certificate files to the ~/openstack-configs folder on the management node and run reconfigure operation.

    ```
    cp <new-key-file> ~/openstack-configs/logstash-forwarder.key
     cp <new-cert-file> ~/openstack-configs/logstash-forwarder.crt
    cd ~/installer-xxxx; ./ciscovimclient/ciscovim reconfigure
    ```

Enabling Keystone v3 on an Existing Install

To continue enhancing our security portfolio, and multi-tenancy with the use of domains, Keystone v3 support has been added in Cisco VIM 2.0 from an authentication end-point. It should be noted that Keystone v2 and v3 are mutually exclusive; i.e. the administrator has to decide during install time the authentication end-point version to go with. By default, VIM orchestrator picks keystone v2 as the authentication end-point. So one can enable Keystonev3 as an install option on day-0 (see 2.0 CiscoVIM install guide), or enable it as a reconfigure option after the pod is installed. To enable Keystone v3 after the pod is installed, one needs to define the following under the optional service section in the setup_data.yam filel.

```
# Optional Services:
OPTIONAL_SERVICE_LIST:
- keystonev3
```

To initiate the integration of Keystone v3 on an existing pod, copy the setupdata into a local dir and update it manually, then run reconfiguration command as follows:

```
[root@mgmt1 ~]# cd /root/
[root@mgmt1 ~]# mkdir MyDir
[root@mgmt1 ~]# cd MyDir
[root@mgmt1 ~]# cp /root/openstack-configs/setup_data.yaml <my_setup_data.yaml>
[root@mgmt1 ~]# vi my_setup_data.yaml (update the setup_data to include keystone v3 info)
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim --setupfile ~/MyDir/<my_setup_data.yaml> reconfigure
```

It should be noted that un-configuration of this feature is not supported today. Additionally, re-versioning Keystone API from v3 to v2 is also not supported.

LDAP support with Keystone v3

With the introduction of KeystoneV3, the openstack service authentication can now we delegated to an external LDAP server. In Cisco VIM 2.0, this feature has been introduced optionally if the authorization is done by Keystone v3. Just like Keystonev3, this feature can be enabled on an existing pod running Cisco VIM 2.0. To avail of this feature post pod deployment, the setup_data needs to be augmented with the following information during the installation of the pod.

An important pre-requisite for enabling LDAP integration is that the LDAP endpoint MUST be reachable from all the Controller nodes that run OpenStack Keystone Identity Service.

```
LDAP:
  domain: <Domain specific name>
  user_objectclass: <objectClass for Users> # e.g organizationalPerson
  group_objectclass: <objectClass for Groups> # e.g. groupOfNames
  user_tree_dn: '<DN tree for Users>' # e.g. 'ou=Users,dc=cisco,dc=com'
  group_tree_dn: '<DN tree for Groups>' # e.g. 'ou=Groups,dc=cisco,dc=com'
  suffix: '<suffix for DN>' # e.g. 'dc=cisco,dc=com'
  url: '<ldap:// host:port>' # e.g. 'ldap://172.26.233.104:389'
  user: '<DN of bind user>' # e.g. 'dc=admin,dc=cisco,dc=com'
  password: <password> # e.g. password of bind user
```

To initiate the integration of LDAP with Keystone v3 on an existing pod, copy the setupdata into a local dir and update it manually with the relevant LDAP and Keystone v3 (if absent from before) configuration, then run reconfiguration command as follows:

```
[root@mgmt1 ~]# cd /root/
[root@mgmt1 ~]# mkdir MyDir
[root@mgmt1 ~]# cd MyDir
[root@mgmt1 ~]# cp /root/openstack-configs/setup_data.yaml <my_setup_data.yaml>
```

```
[root@mgmt1 ~]# vi my_setup_data.yaml (update the setup_data to include LDAP info)
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim --setupfile ~/MyDir/<my_setup_data.yaml> reconfigure
```

The reconfigure feature supports a full or partial reconfiguration of the LDAP integration service.

Note All the parameters within the LDAP stanza are configurable with the exception of the domain parameter.

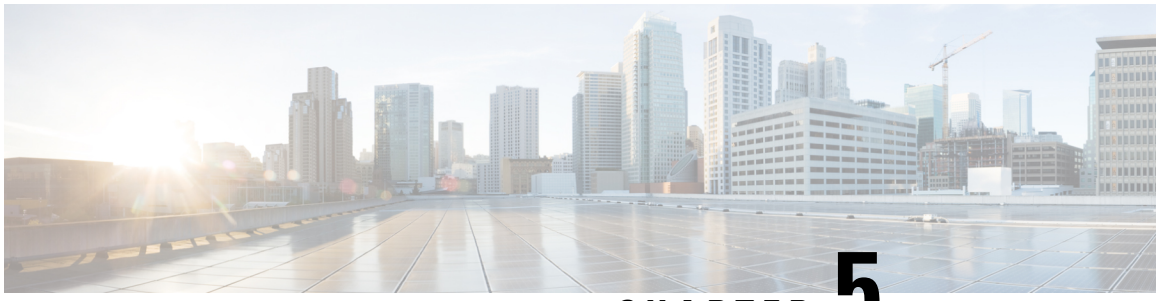
Integrating identity with LDAP over TLS: The automation supports keystone integration with LDAP over TLS. In order to enable TLS, the CA root certificate must be presented as part of the /root/openstack-configs/haproxy-ca.crt file. The url parameter within the LDAP stanza must be set to ldaps.

Additionally, the url parameter supports the following formats: *<ldaps | ldap>://<FQDN | IP-Address>:[port]*.

The protocol can be ldap for non-ssl OR ldaps if tls is to be enabled.

The ldap host can be a fully-qualified domain name or an ip address depending on how the SSL certificates were generated.

The port number is optional and if not provided assumes that the ldap services are running on the default ports. For Example:. 389 for non-ssl and 636 for ssl. However, if these are not the defaults, then the non-standard port numbers must be provided.



CHAPTER 5

Managing Cisco NFVI Storage

This chapter describes basic architectural concepts that will help you understand the Cisco NFVI data storage architecture and data flow. It also provides techniques you can use to monitor the storage cluster health and the health of all systems that depend on it

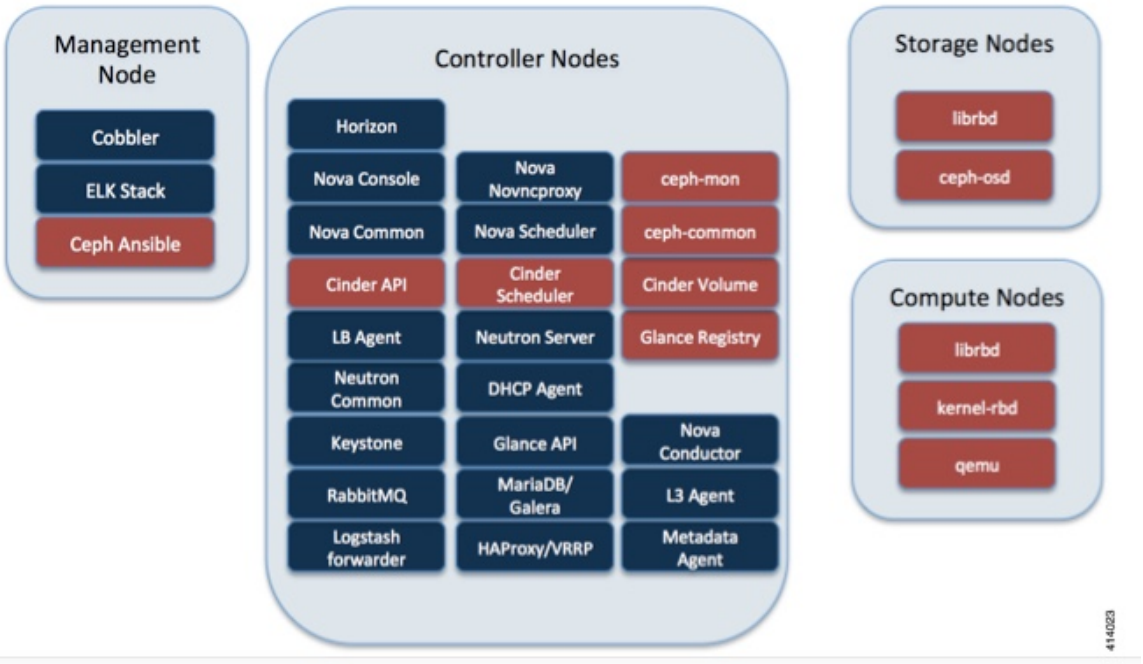
Cisco NFVI Storage Architecture

OpenStack has multiple storage back ends. Cisco NFVI uses the Ceph back end. Ceph supports both block and object storage and is therefore used to store VM images and volumes that can be attached to VMs. Multiple OpenStack services that depend on the storage backend include:

- Glance (OpenStack image service)—Uses Ceph to store images.
- Cinder (OpenStack storage service)—Uses Ceph to create volumes that can be attached to VMs.
- Nova (OpenStack compute service)—Uses Ceph to connect to the volumes created by Cinder.

The following figure shows the Cisco NFVI storage architecture component model.

Figure 7: Cisco NFVI Storage Architecture



Verifying and Displaying Ceph Storage Pools

Ceph is configured with four independent pools: images, volumes, vms, and backups. (A default rbd pool is used internally.) Each Ceph pool is mapped to an OpenStack service. The Glance service stores data in the images pool, and the Cinder service stores data in the volumes pool. The Nova service can use the vms pool to boot ephemeral disks directly from the Ceph cluster depending on how the NOVA_BOOT_FROM option in the ~/openstack-configs/setup_data.yaml was configured prior to Cisco NFVI installation. If NOVA_BOOT_FROM is set to ceph before you run the Cisco NFVI installation, the Nova service will boot from the Ceph vms pool. By default, NOVA_BOOT_FROM is set to local, which means that all VM ephemeral disks are stored as files in the compute nodes. Changing this option after installation does not affect the use of the vms pool for ephemeral disks.

The Glance, Cinder, and Nova OpenStack services depend on the Ceph cluster for backend storage. Therefore, they need IP connectivity to the controller nodes. The default port used to connect Glance, Cinder, and Nova to the Ceph cluster is 6789. Authentication through cephx is required, which means authentication tokens, called keyrings, must be deployed to the OpenStack components for authentication.

To verify and display the Cisco NFVI Ceph storage pools:

Step 1 Launch a SSH session to a controller node, for example:

```
[root@management-server-cisco ~]# ssh root@controller_server-1
```

Step 2 Navigate to the Ceph Monitor container:

```
[root@controller_server-1 ~]# cephmon
```

Step 3 List the Ceph pools:

```
cephmon_4612 [root@controller_server-1 ~]# ceph osd lspools
0 rbd,1 images,2 volumes,3 vms,4 backups,
```

Step 4 List the images pool content:

```
cephmon_4612 [ceph@controller_server-1 /]$ rbd list images
a4963d51-d3b7-4b17-bf1e-2ebac07e1593
```

Checking the Storage Cluster Health

Cisco recommends that you perform a few verifications to determine whether the Ceph cluster is healthy and is connected to the Glance, Cinder, and Nova OpenStack services, which have Ceph cluster dependencies. The first task to check the health of the cluster itself by completing the following steps:

Step 1 From the Cisco NFVI management node, launch a SSH session to a controller node, for example:

```
[root@management-server-cisco ~]# ssh root@controller_server-1
```

Step 2 Navigate to the Ceph Monitor container:

```
[root@controller_server-1 ~]# cephmon
```

Step 3 Check the Ceph cluster status:

```
cephmon_4612 [ceph@controller_server-1 ceph]$ ceph status
```

Sample response:

```
cluster dbc29438-d3e0-4e0c-852b-170aaf4bd935
     health HEALTH_OK
     monmap e1: 3 mons at {ceph-controller_server-1=20.0.0.7:6789/0,
    ceph-controller_server-2=20.0.0.6:6789/0,ceph-controller_server-3=20.0.0.5:6789/0}
            election epoch 8, quorum 0,1,2 ceph-controller_server-3,
    ceph-controller_server-2,ceph-controller_server-1
     osdmap e252: 25 osds: 25 up, 25 in
      pgmap v593: 1024 pgs, 5 pools, 406 MB data, 57 objects
            2341 MB used, 61525 GB / 61527 GB avail
                1024 active+clean
```

This example displays three monitors, all in good health, and 25 object storage devices (OSDs). All OSDs show as up and in the cluster.

Step 4 To see a full listing of all OSDs sorted by storage node, enter:

```
cephmon_4612 [ceph@controller_server-1 ceph]$ ceph osd tree
```

Sample response:

```
ID WEIGHT   TYPE NAME               UP/DOWN REWEIGHT PRIMARY-AFFINITY
-1 60.18979 root default
-2 18.96994     host controller_server-2
 1  2.70999         osd.1                up  1.00000          1.00000
 5  2.70999         osd.5                up  1.00000          1.00000
 6  2.70999         osd.6                up  1.00000          1.00000
```

```
11  2.70999         osd.11              up  1.00000          1.00000
12  2.70999         osd.12              up  1.00000          1.00000
17  2.70999         osd.17              up  1.00000          1.00000
20  2.70999         osd.20              up  1.00000          1.00000
-3 18.96994     host controller_server-1
 0  2.70999         osd.0               up  1.00000          1.00000
 4  2.70999         osd.4               up  1.00000          1.00000
 8  2.70999         osd.8               up  1.00000          1.00000
10  2.70999         osd.10              up  1.00000          1.00000
13  2.70999         osd.13              up  1.00000          1.00000
16  2.70999         osd.16              up  1.00000          1.00000
18  2.70999         osd.18              up  1.00000          1.00000
-4 18.96994     host controller_server-3
 2  2.70999         osd.2               up  1.00000          1.00000
 3  2.70999         osd.3               up  1.00000          1.00000
 7  2.70999         osd.7               up  1.00000          1.00000
 9  2.70999         osd.9               up  1.00000          1.00000
14  2.70999         osd.14              up  1.00000          1.00000
15  2.70999         osd.15              up  1.00000          1.00000
19  2.70999         osd.19              up  1.00000          1.00000
-5  3.27997     host controller_server-4
21  0.81999         osd.21              up  1.00000          1.00000
22  0.81999         osd.22              up  1.00000          1.00000
23  0.81999         osd.23              up  1.00000          1.00000
24  0.81999         osd.24              up  1.00000          1.00000
```

What to do next

After you verify the Ceph cluster is in good health, check that the individual OpenStack components have connectivity and their authentication tokens—keyrings—match the Ceph Monitor keyrings. The following procedures show how to check the connectivity and authentication between Ceph and Glance, Ceph and Cinder, and Ceph and Nova.

Checking Glance Connectivity

The Glance API container must be connected to the Cisco NFVI controller nodes. Complete the following steps to verify the Glance to controller node connectivity:

Step 1 From the management node, launch a SSH session to a controller node, for example:

```
[root@management-server-cisco ~]# ssh root@controller_server-1
```

Step 2 Navigate to the Glance API container:

```
[root@controller_server-1 ~]# glanceapi
```

Step 3 Check the Glance API container connectivity to a controller node different from the one entered in Step 1, in this case, controller_server 2:

```
glanceapi_4612 [glance@controller_server-1 /]$ curl controller_server-2:6789
```

If the connection is successful, you will see a message like the following:

```
glanceapi_4612 [glance@controller_server-1 /]$ curl controller_server-2:6789
ceph v027?
```

If the connection is not successful, you will see a message like the following:

```
glanceapi_4612 [glance@controller_server-1 /]$ curl controller_server-2:6789
curl: (7) Failed connect to controller_server-2:6789; Connection refused
```

A message like the one above means the Ceph monitor running on the target controller node controller_server-2 is not listening on the specified port or there is no route to it from the Glance API container.

Checking one controller node should be enough to ensure one connection path available for the Glance API. However, because Cisco NFVI controller nodes run as part of an HA cluster, you should run Step 3 above targeting all the controller nodes in the Cisco NFVI pod.

What to do next

After you verify the Glance API connectivity to all Cisco NFVI controller nodes, check the Glance keyring to ensure it matches the Ceph monitor keyring.

Verifying Glance and Ceph Monitor Keyrings

Complete the following steps to verify the Glance API keyring matches the Ceph Monitor keyring.

Step 1 Launch a SSH session to a controller node, for example:

```
[root@management-server-cisco ~]# ssh root@controller_server-1
```

Step 2 Navigate to the Glance API container:

```
[root@controller_server-1 ~]# glanceapi
```

Step 3 Check the Glance keyring content, for example:

```
glanceapi_4612 [glance@controller_server-1 /]$ cat /etc/ceph/client.glance.keyring
[client.glance]
key = AQA/pY1XBAnHMBAAeS+0Wmh9PLZe1XqkIW/p0A==
```

Step 4 Navigate to the Ceph Monitor container:

```
[root@controller_server-1 ~]# cephmon
```

Step 5 Display the Ceph Monitor keyring content:

```
cephmon_4612 [ceph@controller_server-1 ceph]$ cat /etc/ceph/ceph.client.glance.keyring
[client.glance]

key = AQA/pY1XBAnHMBAAeS+0Wmh9PLZe1XqkIW/p0A==
```

Verify the keyring matches the Glance API keyring displayed in Step 3.

What to do next

A final check to ensure that Ceph and Glance are connected is to actually import a Glance image using Horizon or the Glance CLI. After you import an image, compare the IDs seen by Glance and by Ceph. They should match, indicating Ceph is handling the backend for Glance.

Verifying Glance Image ID on Ceph

The following steps verify Ceph is properly handling new Glance images by checking that the image ID for a new Glance image is the same as the image ID displayed in Ceph.

Step 1 From the management node, load the OpenStack authentication variables:

```
[root@management-server-cisco ~]# source ~/openstack-configs/openrc
```

Step 2 Import any Glance image. In the example below, a RHEL 7.1 qcow2 image is used.

```
[root@management-server-cisco images]# glance image-create
--name "rhel" --disk-format qcow2  --container-format bare --file
rhel-guest-image-7.1-20150224.0.x86_64.qcow2
```

Step 3 List the Glance images:

```
[root@management-server-cisco images]# glance image-list | grep rhel
| a4963d51-d3b7-4b17-bf1e-2ebac07e1593 | rhel
```

Step 4 Navigate to the Ceph Monitor container:

```
[root@controller_server-1 ~]# cephmon
```

Step 5 Display the contents of the Ceph images pool:

```
cephmon_4612 [ceph@controller_server-1 ceph]$ rbd list images | grep
a4963d51-d3b7-4b17-bf1e-2ebac07e1593
a4963d51-d3b7-4b17-bf1e-2ebac07e1593
```

Step 6 Verify that the Glance image ID displayed in Step 3 matches the image ID displayed by Ceph.

Checking Cinder Connectivity

The Cinder volume container must have connectivity to the Cisco NFVI controller nodes. Complete the following steps to verify Cinder volume has connectivity to the controller nodes:

Step 1 From the management node, launch a SSH session to a controller node, for example:

```
[root@management-server-cisco ~]# ssh root@controller_server-1
```

Step 2 Navigate to the Cinder volume container:

```
[root@controller_server-1 ~]# cindervolume
```

Step 3 Check the Cinder volume container connectivity to a controller node different from the one entered in Step 1, in this case, controller_server-2:

```
cindervolume_4612 [cinder@controller_server-1 /]$ curl controller_server-2:6789
```

If the connection is successful, you will see a message like the following:

```
cindervolume_4612 [cinder@controller_server-1 /]$ curl controller_server-2:6789
ceph v027?
```

If the connection is not successful, you will see a message like the following:

```
cindervolume_4612 [cinder@controller_server-1 /]$ curl controller_server-2:6789
curl: (7) Failed connect to controller_server-2:6789; Connection refused
```

A message like the one above means the Ceph monitor running on the target controller node controller_server-2 is not listening on the specified port or there is no route to it from the Cinder volume container.

Checking one controller node should be enough to ensure one connection path is available for the Cinder volume. However, because Cisco NFVI controller nodes run as part of an HA cluster, repeat Step 3 targeting all the controller nodes in the Cisco NFVI pod.

What to do next

After you verify the Cinder volume connectivity to all Cisco NFVI controller nodes, check the Cinder keyring to ensure it matches the Ceph monitor keyring.

Verifying the Cinder and Ceph Monitor Keyrings

Complete the following steps to verify the Cinder volume keyring matches the Ceph Monitor keyring.

Step 1 From the management node, launch a SSH session to a controller node, for example:

```
[root@management-server-cisco ~]# ssh root@controller_server-1
```

Step 2 Navigate to the Cinder volume container:

```
[root@controller_server-1 ~]# cindervolume
```

Step 3 Check the Cinder keyring content, for example:

```
cindervolume_4612 [cinder@controller_server-1 /]$ cat /etc/ceph/client.cinder.keyring
[client.cinder]
key = AQA/pY1XBAnHMBAAeS+0Wmh9PLZe1XqkIW/p0A==
```

Step 4 Navigate to the Ceph Monitor container:

```
[root@controller_server-1 ~]# cephmon
```

Step 5 Display the Ceph Monitor keyring content:

```
cephmon_4612 [ceph@controller_server-1 ceph]$ cat /etc/ceph/ceph.client.cinder.keyring
[client.cinder]

key = AQA/pY1XBAnHMBAAeS+0Wmh9PLZe1XqkIW/p0A==
```

Verify the keyring matches the Cinder volume keyring displayed in Step 3.

What to do next

As a final Ceph and Cinder connectivity verification, import a Cinder image using Horizon or the Cinder CLI. After you import the image, compare the IDs seen by Cinder and by Ceph. They should match, indicating Ceph is handling the backend for Cinder.

Verifying the Cinder Volume ID on Ceph

The following steps verify Ceph is properly handling new Cinder volumes by checking that the volume ID for a new Cinder volume is the same as the volume ID displayed in Ceph.

Step 1 From the management node, load the OpenStack authentication variables:

```
[root@management-server-cisco ~]# source ~/openstack-configs/openrc
```

Step 2 Create an empty volume:

```
[root@management-server-cisco ~]# cinder create --name ciscovol1 5
```

The above command will create a new 5 GB Cinder volume named ciscovol1.

Step 3 List the Cinder volumes:

```
[[root@management-server-cisco ~]# cinder list
+--------------------------------------+--------+------------------+--
|                  ID                  | Status | Migration Status |…

| dd188a5d-f822-4769-8a57-c16694841a23 | in-use |        -         |…
+--------------------------------------+------+
```

Step 4 Navigate to the Ceph Monitor container:

```
[root@controller_server-1 ~]# cephmon
```

Step 5 Display the contents of the Ceph volumes pool:

```
cephmon_4612 [ceph@controller_server-1 ceph]$ rbd list volumes
volume-dd188a5d-f822-4769-8a57-c16694841a23
```

Step 6 Verify that the Cinder volume ID displayed in Step 3 matches the volume ID displayed by Ceph, excluding the "volume-" prefix.

Checking Nova Connectivity

The Nova libvirt container must have connectivity to the Cisco NFVI controller nodes. Complete the following steps to verify Nova has connectivity to the controller nodes:

Step 1 From the management node, launch a SSH session to a controller node, for example:

```
[root@management-server-cisco ~]# ssh root@Computenode_server-1
```

Step 2 Navigate to the Nova libvirt container:

```
[root@compute_server-1 ~]# libvirt
```

Step 3 Check the Nova libvirt container connectivity to a controller node, in this case, controller_server 1:

```
novalibvirt_4612 [root@compute_server-1 /]$ curl controller_server-2:6789
```

If the connection is successful, you will see a message like the following:

```
novalibvirt_4612 [root@compute_server-1 /]$ curl controller_server-1:6789
ceph v027?
```

If the connection is not successful, you will see a message like the following:

```
novalibvirt_4612 [root@compute_server-1 /]$ curl controller_server-1:6789
curl: (7) Failed connect to controller_server-1:6789; Connection refused
```

A message like the one above means the Ceph monitor running on the target controller node controller_server-1 is not listening on the specified port or there is no route to it from the Nova libvirt container.

Checking one controller node should be enough to ensure one connection path available for the Nova libvirt. However, because Cisco NFVI controller nodes run as part of an HA cluster, you should run Step 3 above targeting all the controller nodes in the Cisco NFVI pod.

What to do next

After you verify the Nova libvirt connectivity to all Cisco NFVI controller nodes, check the Nova keyring to ensure it matches the Ceph monitor keyring.

Verifying the Nova and Ceph Monitor Keyrings

Complete the following steps to verify the Nova libvert keyring matches the Ceph Monitor keyring.

Step 1 From the management node, launch a SSH session to a controller node, for example:

```
[root@management-server-cisco ~]# ssh root@controller_server-1
```

Step 2 Navigate to the Nova libvert container:

```
[root@compute_server-1 ~]# libvirt
```

Step 3 Extract the libvirt secret that contains the Nova libvirt keyring:

```
novalibvirt_4612 [root@compute_server-1 /]# virsh secret-list
UUID                                   Usage …
--------------------------------------------------------------
b5769938-e09f-47cb-bdb6-25b15b557e84  ceph client.cinder …
```

Step 4 Get the keyring from the libvert secret:

```
novalibvirt_4612 [root@controller_server-1 /]# virsh secret-get-value
b5769938-e09f-47cb-bdb6-25b15b557e84
AQBApY1XQCBBEBAAroXvmiwmlSMEyEoXKl/sQA==
```

Step 5 Navigate to the Ceph Monitor container:

```
[root@controller_server-1 ~]# cephmon
```

Step 6 Display the Ceph Monitor keyring content:

```
cephmon_4612 [ceph@controller_server-1 ceph]$ cat /etc/ceph/ceph.client.cinder.keyring
[client.cinder]

key = AQBApY1XQCBBEBAAroXvmiwmlSMEyEoXKl/sQA==
```

Verify the keyring matches the Nova libvert keyring displayed in Step 3. Notice that in the above example the Cinder keyring is checked even though this procedure is for the Nova libvert keyring. This occurs because the Nova services need access to the Cinder volumes and so authentication to Ceph uses the Cinder keyring.

What to do next

Complete a final check to ensure that Ceph and Nova are connected by attaching a Nova volume using Horizon or the Nova CLI. After you attach the Nova volume, check the libvert domain.

Verifying Nova Instance ID

From the management node, complete the following steps to verify the Nova instance ID:

Step 1 Load the OpenStack authentication variables:

```
[root@management-server-cisco installer]# source ~/openstack-configs/openrc
```

Step 2 List the Nova instances:

```
[root@management-server-cisco images]# nova list
+--------------------------------------+------------+--------+--------
| ID                                   | Name       | Status | Task
+--------------------------------------+------------+--------+--------
| 77ea3918-793b-4fa7-9961-10fbdc15c6e5 | cisco-vm   | ACTIVE | -
+--------------------------------------+-----------+-------+-
```

Step 3 Show the Nova instance ID for one of the instances:

```
[root@management-server-cisco images]# nova show
77ea3918-793b-4fa7-9961-10fbdc15c6e5 | grep instance_name
| OS-EXT-SRV-ATTR:instance_name        | instance-00000003
```

The Nova instance ID in this example is instance-00000003. This ID will be used later with the virsh command. Nova instance IDs are actually the libvirt IDs of the libvirt domain associated with the Nova instance.

Step 4 Identify the compute node where the VM was deployed:

```
[root@management-server-cisco images]# nova show 77ea3918-793b-4fa7-9961-10fbdc15c6e5 | grep
hypervisor
| OS-EXT-SRV-ATTR:hypervisor_hostname  | compute_server-1
```

The compute node in this case is compute_server-1. You will connect to this compute node to call the virsh commands. Next, you get the volume ID from the libvirt domain in the Nova libvirt container.

Step 5 Launch a SSH session to the identified compute node, compute_server-1:

```
[root@management-server-cisco ~]# ssh root@compute_server-1
```

Step 6 Navigate to the Nova libvirt container:

```
[root@compute_server-1 ~]# libvirt
```

Step 7 Get the instance libvirt domain volume ID:

```
novalibvirt_4612 [root@compute_server-1 /]# virsh dumpxml instance-00000003 | grep rbd
 <source protocol='rbd' name='volumes/volume-dd188a5d-f822-4769-8a57-c16694841a23'>
```

Step 8 Launch a SSH session to a controller node:

```
[root@management-server-cisco ~]# ssh root@controller_server-1
```

Step 9 Navigate to the Ceph Monitor container:

```
[root@compute_server-1 ~]# cephmon
```

Step 10 Verify volume ID matches the ID in Step 7:

```
cephmon_4612 [ceph@controller_server-1 ceph]
$ rbd list volumes | grep volume-dd188a5d-f822-4769-8a57-c16694841a23
volume-dd188a5d-f822-4769-8a57-c16694841a23
```

Displaying Docker Disk Space Usage

Docker supports multiple storage back ends such as Device Mapper, thin pool, overlay, and AUFS. Cisco VIM uses the devicemapper storage driver because it provides strong performance and thin provisioning. Device Mapper is a kernel-based framework that supports advanced volume management capability. Complete the following steps to display the disk space used by Docker containers.

Step 1 Launch a SSH session to a controller or compute node, for example:

```
[root@management-server-cisco ~]# ssh root@controller_server-1
```

Step 2 Enter the docker info command to display the disk space used by Docker containers:

```
[root@controller_server_1 ~]# docker info
Containers: 24
Images: 186
Storage Driver: devicemapper
 Pool Name: vg_var-docker--pool
 Pool Blocksize: 524.3 kB
 Backing Filesystem: xfs
 Data file:
 Metadata file:
 Data Space Used: 17.51 GB
 Data Space Total: 274.9 GB
 Data Space Available: 257.4 GB…
```

Reconfiguring SwiftStack Integration

Cisco VIM 2.0 provides integration with SwiftStack, an object storage solution. The key aspect of the SwiftStack integration is to add a SwfitStack endpoint to an existing pod running on Cisco VIM 2.0 through the reconfigure option. In this case the SwiftStack is installed and managed outside the Cisco VIM ahead of time, and the

VIM orchestrator adds the relevant Keystone configuration details to access the SwiftStack endpoint (see the Cisco VIM 2.0 install guide for more details of SwiftStack).

The following options support the SwiftStack reconfiguration:

- Enable SwiftStack integration if it is not present.
- Reconfigure the existing SwiftStack PAC endpoint to point to a different cluster (cluster_api_endpoint).
- Reconfigure the Reseller_prefix of the existing SwiftStack installation.
- Reconfigure the admin password (admin_password) of an existing SwiftStack Install.

Integrating SwiftStack over TLS

The automation supports SwiftStack integration over TLS. To enable TLS, the CA root certificate must be presented as part of the `/root/openstack-configs/haproxy-ca.crt` file. The protocol parameter within the SWIFTSTACK stanza must be set to https. As a pre-requisite, the SwiftStack cluster needs to be configured to enable HTTPS connections for the SwiftStack APIs with termination at the proxy servers.

The following section needs to be configured in the Setup_data.yaml file.

```
########################################
# Optional Swift configuration section
########################################
# SWIFTSTACK: # Identifies the objectstore provider by name
#   cluster_api_endpoint: <IP address of PAC (proxy-account-container) endpoint>
#   reseller_prefix: <Reseller_prefix as configured for Keysone Auth,AuthToken support in
Swiftstack E.g KEY_>
#   admin_user: <admin user for swift to authenticate in keystone>
#   admin_password: <swiftstack_admin_password>
#   admin_tenant: <The service tenant corresponding to the Account-Container used by
Swiftstack
    protocol: <http or https> # protocol that swiftstack is running on top
```

Note The operator should pay attention while updating the settings to ensure that SwiftStack over TLS are appropriately pre-configured in the customer-managed SwiftStack controller as specified in the Install guide.

To initiate the integration, copy the `setupdata` into a local directory by running the following command:

```
[root@mgmt1 ~]# cd /root/
[root@mgmt1 ~]# mkdir MyDir
[root@mgmt1 ~]# cd MyDir
[root@mgmt1 ~]# cp /root/openstack-configs/setup_data.yaml <my_setup_data.yaml>
```

Update the `setupdata` by running the following command:

```
[root@mgmt1 ~]# vi my_setup_data.yaml (update the setup_data to include SwiftStack info)
```

Run the reconfiguration command as follows:

```
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim --setupfile ~/MyDir/<my_setup_data.yaml> reconfigure
```

Cinder Volume Backup on SwiftStack

Cisco VIM 2.0 enables cinder service to be configured to backup its block storage volumes to the SwiftStack object store. This feature is automatically configured if the SWIFTSTACK stanza is present in the setup_data.yaml file. The mechanism is to authenticate against SwiftStack during volume backups leverages. The same keystone SwiftStack endpoint is configured to manage objects. The default SwiftStack container that manages cinder volumes within the account (Keystone Tenant as specified by admin_tenant) is currently defaulted to volumebackups.

Reconfiguring Administrator Source Networks

To access the administrator services, Cisco VIM 2.0 provides source IP based filtering of network requests on the management node. These services include SSH and Kibana dashboard access. When the services are configured all admin network requests made to the management node are dropped, except those from white listed addresses in the configuration.

Reconfiguring administrator source network supports the following options:

- Set administrator source network list: Network addresses can be added or deleted from the configuration; the list is replaced in whole during a reconfigure operation.
- Remove administrator source network list: If the **admin_source_networks** option is removed, then the source address will not filter the incoming admin service requests.

The following section needs to be configured in the Setup_data.yaml file:

```
admin_source_networks: # optional, host based firewall to white list admin's source IP
  - 10.0.0.0/8
  - 172.16.0.0/12
```

Note The operator should to be careful while updating the source networks. If the list is mis-configured, operators may lock themselves out of access to the management node through SSH. If this happens, an operator must log into the management node through the console port to repair the configuration.

To initiate the integration, copy the `setupdata` into a local directory by running the following command:

```
[root@mgmt1 ~]# cd /root/
[root@mgmt1 ~]# mkdir MyDir
[root@mgmt1 ~]# cd MyDir
[root@mgmt1 ~]# cp /root/openstack-configs/setup_data.yaml <my_setup_data.yaml>
```

Update the `setupdata` by running the following command:

```
[root@mgmt1 ~]# vi my_setup_data.yaml (update the setup_data to include SwiftStack info)
```

Run the reconfiguration command as follows:

```
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim --setupfile ~/MyDir/<my_setup_data.yaml> reconfigure
```

Password Reset for Cisco VIM Management Node

Run the following command to reset the Root Password of Cisco VIM management node **RHEL-7 / systemd** .

1. Boot your system and wait until the **GRUB2** menu appears.
2. In the **boot loader** menu, highlight any entry and press **e**.
3. Find the line beginning with linux. At the end of this line, append the following:

   ```
   init=/bin/sh
   ```

   Or if you face any alarm, instead of **ro** change **rw** to **sysroot** as shown in the following example:

   ```
   rw init=/sysroot/bin/sh
   ```

4. Press **Ctrl+X** to boot the system using the options you just edited.

   Once the system boots, you will be presented with a shell prompt without having to enter any user name or password:

   ```
   sh-4.2#
   ```

5. Load the installed SELinux policy by running the following command:

   ```
   sh-4.2# /usr/sbin/load_policy -i
   ```

6. Execute the following command to remount your root partition:

   ```
   sh4.2#
       mount -o remount,rw /
   ```

7. Reset the root password by running the following command:

   ```
   sh4.2# passwd root
   ```

   When prompted, enter your new root password and confirm by pressing the **Enter** key. Enter the password for the second time to make sure you typed it correctly and confirm with **Enter** again. If both the passwords match, a message informing you of a successful root password change will appear.

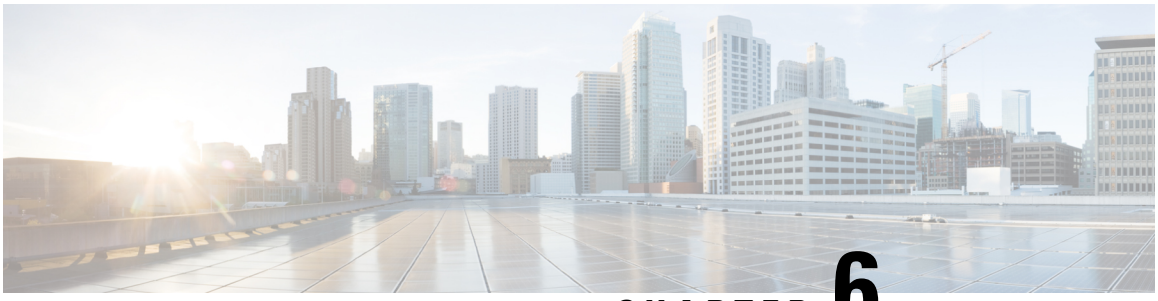
8. Execute the following command to remount the root partition again, this time as read-only:

   ```
   sh4.2#
       mount -o remount,ro /
   ```

9. Reboot the system. Now you will be able to log in as the root user using the new password set up during this procedure.

   To reboot the system, enter **exit** and **exit** again to leave the environment and reboot the system.

References: https://access.redhat.com/solutions/918283.



CHAPTER 6

Overview to Cisco VIM Insight

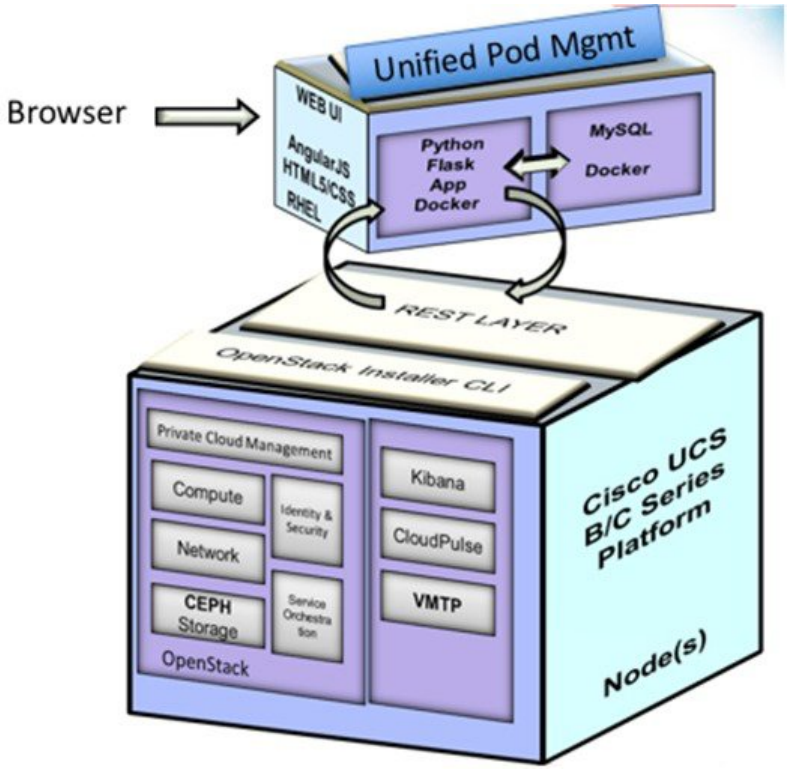
Cisco VIM Insight is an optional application, which acts as a single point of management for the Cisco VIM. If inclusive of your Cisco NFVI package, you can use Cisco VIM Insight to manage Cisco NFVI for day-0 and day-n and for multi-site and multi-pod management features.

Cisco VIM Insight Overview (Tech Preview)

Cisco VIM 2.0 provides an Intuitive and easy way to deploy and manage the NFVI platform, reducing user-error and providing visualization deployment which is a mechanism to manage multiple Cisco VIM Pods from a single portal. In CiscoVIM 2.0, a light-weight UI which is a dockerized application, that supports multi-tenancy with local RBAC support and CiscoVIM Rest layer are integrated. The container based UI platform manages multiple CiscoVIM pods right from day-0, or above in the lifecycle of the cloud.

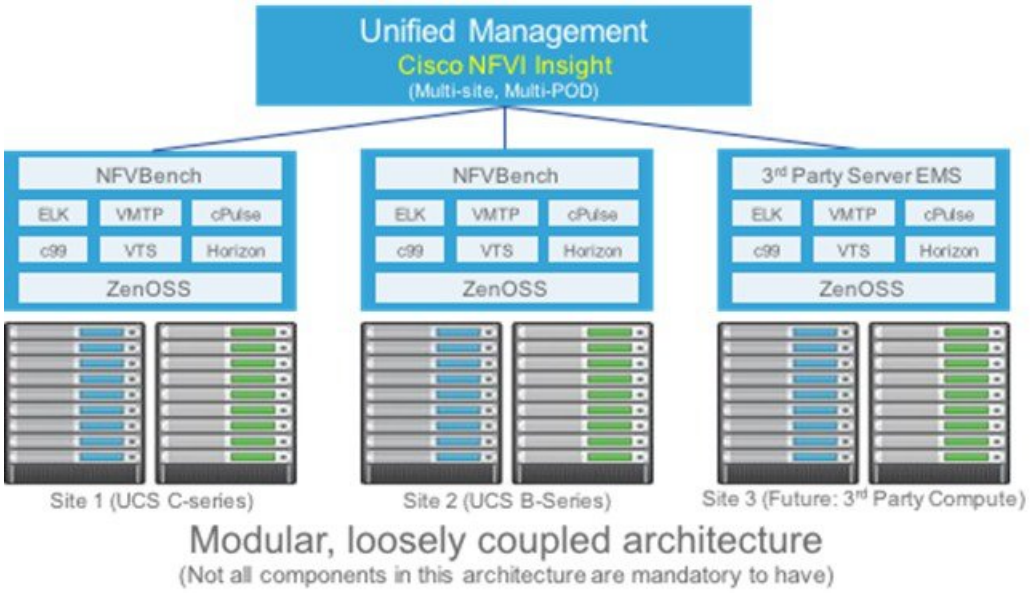
The architecture of the CiscoVIM Insight's interaction with a pod is depicted below:

Figure 8: Cisco VIM Insight's Interaction with a Pod



The architecture of the CiscoVIM Insight is light-weight, hierarchical, and scalable. Each local site is autonomous with localized toolsets. Global Unified Management UI, provides ease of management with multi-site multi-pod capability for distributed NFV deployment at scale. This facility can be used through browsers such as IE, Firefox, Safari, and Chrome. CiscoVIM Insight by itself, is designed to operate in HA as an option. The platform is a modular, loosely coupled architecture, that provides the capability to manage multiple pods, with RBAC support as depicted in the following figure:

Figure 9: Cisco VIM Insight Architecture



The Cisco VIM insight can be installed in:

- Standalone/non-HA mode: One can start off by installing in a Standalone/non-HA mode initially (on the management node of the pod) or a standalone (BOM same as the management node) server and eventually move to a three-node system (BOM is same as the management node) that will provide an HA of the UI system (in future release).
- HA mode (in future release)

As the UI interacts with each pod through REST API and very little RBAC information of the admin and user is kept in the DB, rendering the migrating from one install mode to another can be done effectively.

The UI has two types of views: As part of bootstrap UI admin info is needed, who can add more users as UI and/or Pod Admin.

- UI Admin: UI Admin can add users as UI and/or Pod Admin.
- Pod admin: Pod Admin only have privilege at Pod level, unless he is also an UI admin.

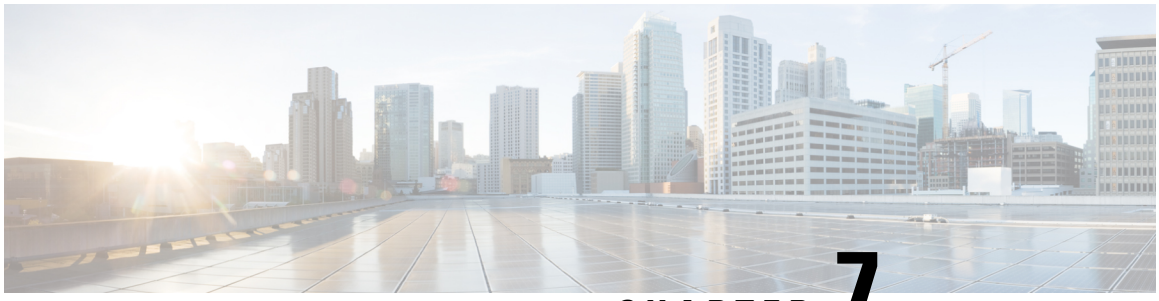
Cisco VIM Insight Admin UI Overview

To ensure the right level of security and delegation, the Admin UI is responsible for management of UI and/or Pod admin, which includes adding and revoking user privileges. Also, the UI admin can delete an existing pod from the overall management pane.

Cisco VIM Insight Pod UI Overview

The pod UI, is responsible for managing each pod. VIM Insight gives easy access to switch between multiple pods, that is being managed by a given Pod Admin. Through the pod UI, a Pod Admin can manage users and

their respective roles and responsibilities. Also, the Pod UI provides the user to execute day-0 (install) and day-n (pod management, software update, and so on.) activities seamlessly. ELK, Horizon Web UI, and so on are also cross-launched and visible for each pod through the Pod UI.



CHAPTER 7

Managing Cisco VIM through Insight (Tech Preview)

Cisco VIM Insight consists of two types of views:

- UI Admin
- Pod Admin

This design brings in clear separation of roles. It does not store any pod related details obtained directly through Rest API from the pods locally, except for RBAC information.

UI Administrators Privileges and Responsibilities

The Insight UI Admin has the following privileges and responsibilities:

1. Insight UI Admin(s) can only add Pod Admin.
2. Insight UI Admin can manage all the users in Insight from **Manage Pod Users**.
   - UI Admin can revoke permission of Users: If UI Admin wants to revoke a user from a Pod, click **Revoke Permission** icon under Action column.
   - UI Admin can also Delete a User: If UI Admin wants to delete a user from the Insight, Click **Delete** icon under Action column. If there is only one user associated with a pod then UI Admin needs to delete the pod and then delete or revoke the user permission.
3. Insight UI Admin can manage Pod Admin(s) from **Manage Pod Admin**.
   - UI Admin can add a new Pod Admin in Insight.
   - UI Admin can revoke permission of a user from being a Pod Admin.
4. Insight UI Admin can manage Pods from Manage Pods.

- UI Admin can delete a Pod from Insight.
- UI Admin can also update password for the REST incase there was a system update on the pod and REST password was changed in that process.

5. Insight UI Admin can manage other UI Admin(s) from Manage UI Admin Users page.
   - Insight UI Admin can add another UI admin.
   - Insight UI Admin can revoke permission of the user from being an UI Admin.

Note If there is only one UI Admin for Insight then revoke permission icon will be disabled for the user.

Pod UI Privileges and Responsibilities

As Cisco VIM is Rest API based, one can manage a pod through CLI, Rest API or UI. Users can always bring in a partial or fully functional pod and register with VIM Insight. Insight will query the pod status through Rest API and reflect the same. Despite the flexibility in design, it is recommended that the admin chooses to manage the pod one way, to avoid confusion. For steps to bootstrap VIM Insight, users are requested to refer to the VIM 2.0 Install guide. Post bootstrap of VIM Insight, listed below are the steps one can take to add Cisco VIM Pod.

Adding Cisco VIM Pod

Before you begin

Following assumption should be completed to add a Cisco VIM Pod:

- Bootstrap of VIM Insight is complete and successful as per the install guide.
- At minimum, a UI and Pod Admin exists as per the install guide.

Step 1 Navigate to https://br_api:9000

Step 2 Click the **Register Management Node** link.

- Enter the Endpoint IP which is the **br_api** of your pod.

  Note Run time validation to check if the Endpoint IP is already registered to Insight.

- Give a name or tag for the pod you are registering.
- Enter the REST API password for the pod.
  - You can locate the REST API password on the pod you are registering.
  - The path to locate REST API password is : /opt/cisco/ui_config.json.

- A brief description about Management Node. Description field is optional and can be left blank.
- Enter the email-id of the Pod Admin.
  - Run time validation to check if the email id is pod admin or not.
  - If false, the Insight will throw an error "User is not registered as Pod Admin".
  - If True, the User Name section will be filled automatically and the **Register** button will be enabled.

Step 3 Click **Register** button to redirect the user to the landing or login page. Notification mail would be received by the Pod admin.

Deleting Pod from Cisco VIM insight

When you delete a pod from Cisco VIM Insight, you are ensuring that Cisco VIM Insight cannot manage that Cisco VIM pod. You are not deleting the pod from your OpenStack deployment.

Before you begin

Following assumption should be covered to delete a Cisco VIM Pod:

- Bootstrap of VIM Insight is complete and successful as per the install guide.
- At least one UI and Pod Admin exists as per the install guide.
- The targeted pod is being managed by Insight.

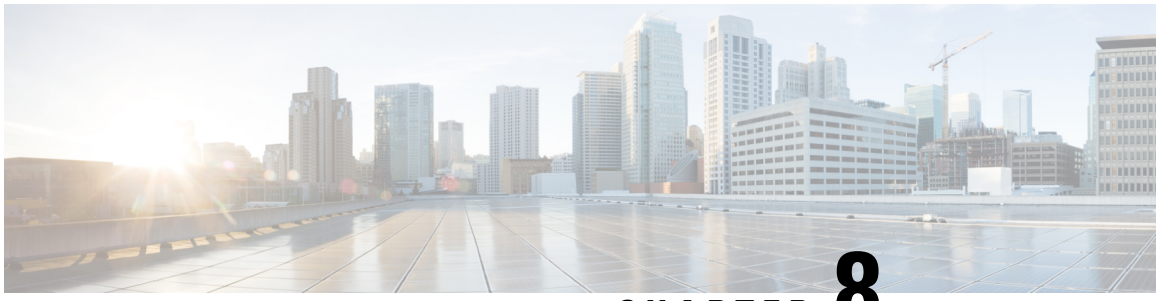
Step 1 Login as the Insight UI Admin.
Step 2 In the Navigation pane, click **Manage Pods**.
Step 3 Choose the pod that you want to delete and click **Delete**.
Step 4 Click **Proceed**, to confirm the deletion.

Context Switching within Insight

There are two ways a user can switch to another pod.

- **Context Switching Icon**: Context Switching Icon is situated on the top right corner and is the third icon from the right tool tip of the UI. Click **Context Switching** Icon to view all the pods that you can access. Pod with a red dot indicates that the REST Password entered during registration of Management node does not match with the current REST Password for that of particular node. In such a situation the Pod admin or User should reach out to UI admin to update the password for that Node. UI admin updates the password from Manage Pods in Insight UI admin Portal.
- **Switch Between Management Nodes**: Switch Between Management Nodes is available in the Dashboard. The user can see all the pods in the table and can navigate to any Pod using a single click. If mouse

pointer changes from hand or cursor to a red dot sign it indicates that the REST Password entered during registration of Management node does not match with the current REST Password for that particular node.



CHAPTER 8

Managing Blueprints

The following topics tell you how to manage Cisco NFVI Blueprints.

Blueprints

Blueprints contain the configuration metadata required to deploy an OpenStack system through a Cisco VIM pod in Cisco VIM Insight. You can create a blueprint in Cisco Insight or you can upload a yaml file that contains the metadata for a blueprint. You can also create a blueprint from an existing OpenStack system that you are configuring as a Cisco VIM pod.

The configuration in the blueprint is specific to the type of Cisco UCS server that is in the OpenStack system. A blueprint for a C-Series server-based OpenStack system cannot be used to configure a B-Series server-based OpenStack system. Cisco Insight will display an error if the blueprint does not match the configuration of the OpenStack system.

The blueprint enables you to quickly change the configuration of an OpenStack system. While only one blueprint can be active, you can create or upload multiple blueprints for a Cisco VIM pod. If you change the active blueprint for a pod, you update the configuration of the OpenStack system to match the new blueprint.

You can modify and validate an existing blueprint, or delete a blueprint. However, you cannot modify any of the configuration metadata in the active blueprint for a Cisco VIM pod.

Blueprint Activation

A blueprint becomes active when you use it in a successful installation for a Cisco VIM pod. Any other blueprints that you created or uploaded to that pod are in non-active state.

Uploading or creating a blueprint does not activate that blueprint for the pod. You need to install a blueprint through the **Cisco VIM Suite** wizard. If the installation is successful, the selected blueprint becomes active.

Note If you want to activate a new blueprint in an existing pod, you need to delete certain accounts and the credential policies for that pod before you activate the blueprint. See Activating a Blueprint in an Existing Pod with OpenStack Installed, on page 126.

Viewing Blueprint Details

You can view the details of an OpenStack installation blueprint. To view blueprint details:

Step 1 Log in to Cisco VIM Insight as pod User.

Step 2 In the Dashboard's Switch between Management Nodes, select the Cisco VIM pod with the blueprint that you want to view.

Step 3 Click **Menu** button at the top left corner to expand the navigation pane.

Step 4 Choose **Pre-Install** > **Blueprint Management.**

Step 5 Choose a blueprint from the list.

Step 6 Click **Preview & Download YAML.**

Creating a Blueprint for B-Series Server Platform

Typically, you create the blueprint when you create the Cisco VIM pod. Follow the instructions below to create an additional blueprint for a pod that uses B-Series servers.

Before you begin

Create a Cisco VIM Insight User Account and Register the respective Pod.

Step 1 Log-in to Cisco VIM Insight.

Step 2 In the **Navigation** pane, expand the **Pre-Install Section**.

Step 3 Click **Blueprint Setup**.

Step 4 On the **Blueprint Initial Setup** page of the Cisco VIM Insight, complete the following fields:

Name	Description
Blueprint Name field.	Enter the name for the blueprint configuration.
Platform Type drop-down list.	• B-Series (By Default) • C-Series
Tenant Network drop-down. list.	Choose one of the following tenant network types: • Linux Bridge/VXLAN • OVS/VLAN

Name	Description
Ceph Mode drop-down list.	Choose one of the following Ceph types: • Dedicated (By Default) • Central
Optional Features and Services checkbox.	Swiftstack, LDAP, Syslog Export Settings, COLLECTD, Install Mode, TorSwitch Information, TLS, Nfvmon, Pod Name, VMTP, Nfvbench, Auto Backup, Heat, Keystone v3. If any one is selected, the corresponding section is visible in various Blueprint sections. By default all options are disabled.
Import Existing YAML field.	If you have an existing B Series YAML file you can use this feature to upload the file. Insight will automatically fill in the fields and if any mandatory fields are missed then the respective section will be highlighted.

Step 5 Click **Physical Setup** to advance to the **Registry Setup** configuration page. Fill in the following details for Registry Setup:

Name	Description
Registry User Name text field.	User-Name for Registry **(Mandatory)**.
Registry Password text field.	Password for Registry **(Mandatory)**.
Registry Email text field.	Email ID for Registry **(Mandatory)**.

Once all Mandatory fields are filled the **Validation Check Registry** page will be changed to a Green Tick.

Step 6 Click **UCSM Common** tab and fill the following fields:

Name	Description
User name disabled field.	By default value is admin.
Password text field.	Enter Password for UCSM Common **(Mandatory)**.
UCSM IP text field.	Enter IP Address for UCSM Common **(Mandatory)**.
Resource Prefix text field	Enter the resource prefix (Mandatory)
QOS Policy Type drop-down.	Choose one of the following types: • NFVI (Default) • Media

Name	Description
Enable Prov FI PIN optional checkbox.	Default is false.
MRAID-CARD optional checkbox.	Enables JBOD mode to be set on disks. Applicable only if you have RAID controller configured on Storage C240 Rack servers.
Enable UCSM Plugin optional checkbox.	Visible when Tenant Network type is OVS/VLA.
Enable QoS Policy optional Checkbox.	Visible only when UCSM Plugin is enabled. If UCSM Plugin is disabled then this option will be set to False.
SRIOV Multi VLAN Trunk optional grid.	Visible when UCSM Plugin is enabled. Enter the values for network and vlans ranges. Grid can handle all CRUD operations like Add, Delete, Edit and Multiple Delete.

Step 7 Click **Networking** to advance to the networking section of the Blueprint.

Name	Description
Domain Name field.	Enter the domain name **(Mandatory)**.
NTP Servers field.	Enter a maximum of four and minimum of one IPv4 addresses in the table.
Domain Name Servers field.	Enter a maximum of three and minimum of one IPv4 addresses.
HTTP Proxy Server field.	If your configuration uses an HTTP proxy server, enter the IP address of the server.
HTTPS Proxy Server field.	If your configuration uses an HTTPS proxy server, enter the IP address of the server.

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Network table.</td><td>Network table is pre-populated with Segments. To add Networks you can either clear all the table using Delete all or click Edit icon for each segment and fill in the details.

You can add, edit, or delete network information in the table.

• Click Edit to enter new entries (networks) to the table.
• Specify the following fields in the Edit Entry to Networks dialog:

<table>
<tr><td>Segment drop-down list.</td><td>By Default already Selected.</td></tr>
<tr><td>Management Node IP field.</td><td>Enter the IP address of the build node.

This field is only available for the Mgmt/Provision segment and is only Mandatory if Zenoss is selected to be a part of Blueprint.</td></tr>
<tr><td>VALN field.</td><td>Enter the VLAN ID. For Segment - Provider, the VLAN ID value is always none.</td></tr>
<tr><td>Subnet ID field.</td><td>Enter the IPv4 address for the subnet.</td></tr>
<tr><td>Gateway field.</td><td>Enter the IPv4 address for the Gateway.</td></tr>
<tr><td>If Rack is chosen, the Rack Unit ID field is displayed</td><td>Enter a Rack Unit ID.</td></tr>
<tr><td>Pool field.</td><td>Enter the pool information in the required format, for example: 10.1.1.5-10.1.1.10,102.1.5-102.1.10

This field is only available for the Mgmt/Provision, Storage, and Tenant segments.</td></tr>
</table>
</td></tr>
<tr><td colspan="2">Click Save.</td></tr>
</table>

Step 8 On the **Servers and Roles** page of the **Cisco VIM Suite** wizard, click **Add (+)** to add a new entry in the table, and complete the following fields:

Name	Description
Add Entry to Servers and Roles.	Click **Edit** or + to add a new server and role to the table.
	Server Name — Enter a server name.
	Server Type drop-down list. — Choose Blade or Rack from the drop-down list.
	Rack ID field. — The Rack ID for the server.
	Chassis ID field — Enter a Chassis ID.
	If Rack is chosen, the **Rack Unit ID** field is displayed. — Enter a Rack Unit ID.
	If Blade is chosen, the **Blade ID** field is displayed. — Enter a Blade ID.
	Select the **Role** from the drop down list. — If Server type is Blade then Control and Compute. If Rack is selected then Block Storage.
	Management IP. — It is an optional field but if provided for one server then it is mandatory to provide it for other Servers as well.
Click **Save or Add** button.	Clicking **Save or Add** button, adds all information for Servers and Roles. Fill in all mandatory fields.

Step 9 Click **ToR Switch** checkbox in Blueprint Initial Setup to enable the **TOR SWITCH** configuration page. It is an **Optional** section in Blueprint Setup, but once all the fields are filled in then it will become a part of the Blueprint.

Name	Description
Configure ToR optional checkbox .	If you enable this checkbox, the Configure ToR section will change from false to true.

<table>
<tr><th>Name</th><th colspan="2">Description</th></tr>
<tr><td>ToR Switch Information mandatory table if you want to enter Tor information.</td><td colspan="2">Click + to add information for Tor Switch.
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Name</td><td>ToR switch name.</td></tr>
<tr><td>Username</td><td>ToR switch username.</td></tr>
<tr><td>Password</td><td>ToR switch Password.</td></tr>
<tr><td>SSH IP</td><td>ToR switch SSH IP Address.</td></tr>
<tr><td>SSN Num</td><td>ToR switch ssn num.</td></tr>
<tr><td>VPC Peer Keepalive</td><td>Peer Management IP. You need not define if there is no peer.</td></tr>
<tr><td>VPC Domain</td><td>Need not define if there is no peer.</td></tr>
<tr><td>VPC Peer port</td><td>Interface for vpc peer ports.</td></tr>
<tr><td>VPC Peer VLAN Info</td><td>vlan ids for vpc peer ports (optional).</td></tr>
<tr><td>BR Management Port Info</td><td>Management interface of build node.</td></tr>
<tr><td>BR Management PO Info</td><td>Port channel number for management interface of build node.</td></tr>
<tr><td>BR Management VLAN info</td><td>vlan id for management interface of build node (access).</td></tr>
</table></td></tr>
<tr><td rowspan="2">On clicking Save button, Add Tor Info Connected to Fabric field will be visible.</td><td>Port Channel field.</td><td>Enter the Port Channel input.</td></tr>
<tr><td>Switch Name field.</td><td>Enter the friendly name.</td></tr>
</table>

Step 10 Click **OpenStack Setup** tab to advance to the **OpenStack Setup** page.

Step 11 On the **OpenStack Setup** page of the Cisco VIM Insight wizard, complete the following fields:

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>HA Proxy</td><td>Fill in the mandatory fields:<table>
<tr><td>External VIP Address</td><td>Enter IP Address of External VIP.</td></tr>
<tr><td>Virtual Router ID</td><td>Enter the Router ID for HA.</td></tr>
<tr><td>Internal VIP Address</td><td>Enter IP Address of Internal VIP.</td></tr>
</table></td></tr>
<tr><td>Keystone</td><td>Mandatory field and pre-populated. This option is always true.<table>
<tr><td>Admin Username</td><td>admin.</td></tr>
<tr><td>Admin Tenant Name</td><td>admin.</td></tr>
</table></td></tr>
<tr><td>Ldap on keystone.</td><td>Ldap enable checkbox by default is false.<table>
<tr><td>Domain Namefield.</td><td>Enter name for Domain name.</td></tr>
<tr><td>Object class for Userfield.</td><td>Enter a string as input.</td></tr>
<tr><td>Object class for Group</td><td>Enter a string.</td></tr>
<tr><td>Domain Name tree for Groups</td><td>Enter a string.</td></tr>
<tr><td>Domain Name tree for User field.</td><td>Enter a string.</td></tr>
<tr><td>Suffix for domain name field.</td><td>Enter a string.</td></tr>
<tr><td>URL field.</td><td>Enter a URL with ending port number.</td></tr>
<tr><td>Domain Name for Bind User field.</td><td>Enter a string.</td></tr>
<tr><td>Password field.</td><td>Enter Password as string format.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Neutron</td><td>Neutron fields change on the basis of Tenant Network Type Selection from Blueprint Initial Setup page.

Following are the options available for Neutron for OVS/VLAN:

<table>
<tr><td>Tenant Network Type</td><td>Auto Filled based on the Tenant Network Type selected in the Blueprint Initial Setup page.</td></tr>
<tr><td>Mechanism Drivers</td><td>Auto Filled based on the Tenant Network Type selected in Blueprint Initial Setup page.</td></tr>
<tr><td>NFV Hosts</td><td>Auto filled with the Compute you added in Server and Roles.

If you select All in this section NFV_HOSTS: ALL will be added to the Blueprint or you can select one particular compute. For Eg:

NFV_HOSTS: compute-server-1, compute-server-2.</td></tr>
<tr><td>Tenant VLAN Ranges</td><td>List of ranges separated by comma of formstart:end.</td></tr>
<tr><td>Enable Jumbo Frames</td><td>Check Box</td></tr>
</table>
For Tenant Network Type Linux Bridge, everything will remain the same except Tenant VLAN Ranges which will be removed.</td></tr>
<tr><td>CEPH</td><td>Ceph has two pre-populated fields

• CEPH Mode: By default Dedicated.

• NOVA Boot from: From the drop-down, choose Ceph or local.</td></tr>
<tr><td>GLANCE</td><td>By default Populated for CEPH Dedicated with Store Backend value as CEPH.</td></tr>
<tr><td>CINDER</td><td>By default Populated for CEPH Dedicated with Volume Driver value as CEPH.</td></tr>
</table>

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>VMTP optional section will only be visible once VMTP is selected from Blueprint Initial Setup.</td><td>Check one of the check boxes to specify a VMTP network:
• Provider Network
• External Network

For the Provider Network complete the following:
<table><tr><td>Network Name field.</td><td>Enter the name for the external network.</td></tr><tr><td>IP Start field.</td><td>Enter the starting floating IPv4 address.</td></tr><tr><td>IP End field.</td><td>Enter the ending floating IPv4 address.</td></tr><tr><td>Gateway field</td><td>Enter the IPv4 address for the Gateway.</td></tr><tr><td>DNS Server field.</td><td>Enter the DNS server IPv4 address.</td></tr><tr><td>Segmentation ID field.</td><td>Enter the segmentation ID.</td></tr><tr><td>Subnet</td><td>Enter the Subnet for Provider Network.</td></tr></table>For External Network fill in the following details:
<table><tr><td>Network Name field.</td><td>Enter the name for the external network.</td></tr><tr><td>Network IP Start field.</td><td>Enter the starting floating IPv4 address.</td></tr><tr><td>Network IP End field.</td><td>Enter the ending floating IPv4 address.</td></tr><tr><td>Network Gateway field</td><td>Enter the IPv4 address for the Gateway.</td></tr><tr><td>DNS Server field.</td><td>Enter the DNS server IPv4 address.</td></tr><tr><td>Subnet</td><td>Enter the Subnet for External Network.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>TLS section will be visible if TLS is selected from Blueprint Initial Setup Page.</td><td>TLS has two options:
External LB VIP FQDN - Text Field.
External LB VIP TLS - True/False. By default this option is false.</td></tr>
<tr><td>SwiftStack optional section will be visible if SwiftStack is selected from Blueprint Initial Setup Page. SwiftStack is only supported with KeyStonev2 . If you select Keystonev3, swiftstack cannot be configured.</td><td>Following are the options that needs to be filled for SwiftStack:<table>
<tr><td>Cluster End Point</td><td>IP address of PAC (proxy-account-container) endpoint.</td></tr>
<tr><td>Admin User</td><td>Admin user for swift to authenticate in keystone.</td></tr>
<tr><td>Admin Tenant</td><td>The service tenant corresponding to the Account-Container used by Swiftstack.</td></tr>
<tr><td>Reseller Prefix</td><td>Reseller_prefix as configured for Keysone Auth,AuthToken support in Swiftstack E.g KEY_</td></tr>
<tr><td>Admin Password</td><td>swiftstack_admin_password</td></tr>
<tr><td>Protocol</td><td>http or https</td></tr>
</table></td></tr>
</table>

Step 12 If **Syslog Export** or **NFVBENCH** is selected in **Blueprint Initial Setup** Page, then **Services Setup** page would be **enabled** for user to view. Following are the options under **Services Setup Tab**:

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Syslog Export.</td><td>Following are the options for Syslog Settings:<table>
<tr><td>Remote Host</td><td>Enter Syslog IP Address.</td></tr>
<tr><td>Facility</td><td>Defaults to local5</td></tr>
<tr><td>Severity</td><td>Defaults to debug</td></tr>
<tr><td>Clients</td><td>Defaults to ELK</td></tr>
<tr><td>Port</td><td>Defaults to 514 but can be modified by the User.</td></tr>
</table></td></tr>
</table>

Name	Description
NFVBENCH	Enable checkbox which by default is **false**. Add Tor information connected to switch: • Select a TOR Switch and Enter the Switch name. • Enter the port number. For example: eth1/5 • NIC Ports: INT1 and INT2 optional input, enter the 2 port numbers of the 4-port 10G Intel NIC at the management node used for NFVBench.

Step 13 Click **Offlinevalidation** button to initiate an offline Blueprint validation.

Step 14 Once the **Offlinevalidation** is successful, **Save** option will be enabled which will redirect you to the **Blueprint Management** page.

Creating a Blueprint for C-Series Server Platform

Typically, you create the blueprint when you create the Cisco VIM pod. You can use this procedure to create an additional blueprint for a pod that uses C-Series servers.

Before you begin

Create a Cisco VIM Insight User Account and register the respective Pod.

Step 1 Log-in to **CISCO VIM Insight**.

Step 2 In the **Navigation** pane, expand the **Pre-Install Section**.

Step 3 Click **Blueprint Setup**.

Step 4 On the **Blueprint Initial Setup** page of the Cisco VIM Insight , complete the following fields:

Name	Description
Blueprint Name field	Enter the name for the blueprint configuration.
Platform Type drop-down list	• B-Series (By Default) • C-Series (Select C Series)
Tenant Network drop-down list	Choose one of the following tenant network types: • Linux Bridge/VXLAN • OVS/VLAN • VTS/VLAN • ML2VPP/VLAN

Name	Description
Ceph Mode drop-down list	Choose one of the following Ceph types: • Dedicated (By Default) • Central
Optional Features and Services checkbox.	Swiftstack, LDAP, Syslog Export Settings, COLLECTD, Install Mode, TorSwitch Information, TLS, Nfvmon, Pod Name, VMTP, Nfvbench, Auto Backup, Heat, Keystone v3 If any one is selected, the corresponding section is visible in various Blueprint sections. By default all options are disabled.
Import Existing YAML file	If you have an existing C Series YAML file you can use this feature to upload the file. Insight will automatically fill in the fields and if any mandatory field is missed then would highlight it in the respective section.

Step 5 Click **Physical Setup** to advance to the **Registry Setup** configuration page. Fill in the following details for Registry Setup.

Name	Description
Registry User Name text field.	User-Name for Registry **(Mandatory)**.
Registry Password text field.	Password for Registry **(Mandatory)**.
Registry Email text field.	Email ID for Registry **(Mandatory)**.

Once all Mandatory fields are filled, the **Validation Check Registry** page will indicate a green tick.

Step 6 Click **CIMC Common** tab and complete the following fields:

Name	Description
User Name disabled field.	By default value is Admin.
Password text field.	Enter Password for UCSM Common **(Mandatory)**.

Step 7 Click **Networking** to advance to the networking section of the Blueprint.

Name	Description
Domain Name field.	Enter the domain name **(Mandatory).**
NTP Servers field.	Enter a maximum of four and minimum of one IPv4 addresses in the table.

Name	Description
Domain Name Servers field	Enter a maximum of three and minimum of one IPv4 addresses
HTTP Proxy Server field	If your configuration uses an HTTP proxy server, enter the IP address of the server.
HTTPS Proxy Server field.	If your configuration uses an HTTPS proxy server, enter the IP address of the server.
Networks table	Network table is pre-populated with segments. To add Networks you can either clear all the table using **Delete all** or click **Edit** icon for each segment and fill in the details. You can add, edit, or delete network information in the table.

Step 8 Click **Edit** to enter new entries (networks) to the table. Specify the following fields in the **Edit Entry** to Networks dialog:

Name	Description
Segment drop-down list	By default Selected.
Management Node IP field.	Enter the IP address of the build node. This field is only available for the Mgmt/Provision segment and is only Mandatory if Zenoss is selected as part of Blueprint.
VLAN field	Enter the VLAN ID. For Segment - Provider , the VLAN ID value is always **none**.
Subnet field	Enter the IPv4 address for the subnet.
Gateway field	Enter the IPv4 address for the Gateway.
Pool field	Enter the pool information in the required format, for example: 10.1.1.5-10.1.1.10,10.2.1.5-10.2.1.10 This field is only available for the Mgmt/Provision, Storage, and Tenant segments.
Click **Save**.	

Step 9 On the **Servers and Roles** page of the **Cisco VIM Suite** wizard, click **Add (+)** to add a new entry in the table, and complete the following fields:

You can edit or delete existing entries in the **Server and Roles** table.

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Add Entry to Servers and Roles .</td><td>Click Edit or + to add a new server and role to the table.
<table>
<tr><td>Server Name</td><td>Entry a friendly name .</td></tr>
<tr><td>Boot Drive drop-down list.</td><td>Choose LOCALHDD or SDCARD from the drop-down list.</td></tr>
<tr><td>Rack ID field.</td><td>The rack ID for the server.</td></tr>
<tr><td>VIC Slot field.</td><td>Enter a VIC Slot.</td></tr>
<tr><td>CIMC IP field.</td><td>Enter a IP address.</td></tr>
<tr><td>CIMC Username field.</td><td>Enter a Username.</td></tr>
<tr><td>CIMC Password field.</td><td>Enter a Password for CIMC</td></tr>
<tr><td>Select the Role from the drop down list.</td><td>Choose Control or Compute or BlockStorage from the drop-down list.</td></tr>
<tr><td>Management IP</td><td>It is an optional field but if provided for one server then it is mandatory to provide it for other servers.</td></tr>
</table></td></tr>
<tr><td>If ToR checkbox is selected with an entry, this field will be displayed.</td><td>• Port Channel field.
• Switch Name field.</td><td>• Enter the Port Channel input
• Enter the friendly name</td></tr>
<tr><td>If Intel NIC support is checked in server and roles with ToR being selected.</td><td>Add SRIOV ToR info connected to switch.</td><td>Enter the switch-name.</td></tr>
<tr><td>If Intel NIC is checked with an entry of integer value, then Add DP ToR info is connected to switch filed.</td><td>• Port Channel field.
• Switch-Name field.</td><td>• Enter the Port channel.
• Enter the string.</td></tr>
<tr><td>Click Save or Add button.</td><td colspan="2">If all mandatory fields are filled click Save or Add button information for Servers and Roles</td></tr>
</table>

Step 10 Click **Tor Switch** Checkbox in **Blueprint Initial Setup** to enable the **TOR SWITCH** configuration page. It is an **Optional** section in Blueprint Setup but once all the fields are filled, it becomes a part of the Blueprint.

Name	Description
Configure TOR optional checkbox.	If you enable this checkbox configure tor section would be changed from false to true.

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>TOR Switch Information mandatory table if you want to enter ToR information.</td><td>Click + to add information for ToR Switch.
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Name</td><td>ToR Switch Name.</td></tr>
<tr><td>Username</td><td>TOR switch username</td></tr>
<tr><td>Password</td><td>ToR switch Password</td></tr>
<tr><td>SSH IP</td><td>TOR switch ssh ip</td></tr>
<tr><td>SSN Num</td><td>TOR switch ssn num</td></tr>
<tr><td>VPC Peer Keepalive</td><td>Peer Management IP. Do not define if there is no peer</td></tr>
<tr><td>VPC Domain</td><td>Do not define if there is no peer</td></tr>
<tr><td>VPC Peer Port Info</td><td>Interface for vpc peer ports</td></tr>
<tr><td>VPC Peer VLAN Info</td><td>vlan ids for vpc peer ports (optional)</td></tr>
<tr><td>BR Management Port Info</td><td>Management interface of build node</td></tr>
<tr><td>BR Management PO Info</td><td>Port channel number for management interface of build node</td></tr>
<tr><td>BR Management VLAN info</td><td>vlan id for management interface of build node (access)</td></tr>
</table></td></tr>
<tr><td colspan="2">Click Save.</td></tr>
</table>

Step 11 Click **OpenStack Setup** Tab to advance to the **OpenStack Setup** page.

Step 12 In the **OpenStack Setup** page of the Cisco VIM Insight wizard, complete the following fields:

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>HA Proxy</td><td>Mandatory Field. Fill in the following details:
<table>
<tr><td>External VIP Address</td><td>Enter IP Address of External VIP</td></tr>
<tr><td>Virtual Router ID</td><td>Enter the Router ID for HA</td></tr>
<tr><td>Internal VIP Address</td><td>Enter IP Address of Internal VIP</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Keystone</td><td>Mandatory field and prepopulated. This option would always be true.<table><tr><td>Admin Username</td><td>admin</td></tr><tr><td>Admin Tenant Name</td><td>admin</td></tr></table></td></tr>
<tr><td>Ldap on keystone</td><td>Ldap enable checkbox by default is false.<table><tr><td>Domain Namefield.</td><td>Enter name for Domain name.</td></tr><tr><td>Object class for Userfield.</td><td>Enter a string as input.</td></tr><tr><td>Object class for Group</td><td>Enter a string.</td></tr><tr><td>Domain Name tree for Groups</td><td>Enter a string.</td></tr><tr><td>Domain Name tree for User field.</td><td>Enter a string.</td></tr><tr><td>Suffix for domain name field.</td><td>Enter a string.</td></tr><tr><td>URL field.</td><td>Enter a URL with ending port number.</td></tr><tr><td>Domain Name for Bind User field.</td><td>Enter a string.</td></tr><tr><td>Password field.</td><td>Enter Password as string format.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Neutron</td><td>Neutron fields would change on the basis of Tenant Network Type Selection from Blueprint Initial Setup. Following are the options available for Neutron for OVS/VLAN:
<table>
<tr><td>Tenant Network Type</td><td>Auto Filled based on the Tenant Network Type selection in Blueprint Initial Setup page.</td></tr>
<tr><td>Mechanism Drivers</td><td>Auto Filled based on the Tenant Network Type selection in Blueprint Initial Setup page.</td></tr>
<tr><td>NFV Hosts</td><td>Auto filled with the Compute you added in Server and Roles. If you select All in this section NFV_HOSTS: "ALL" will be added to the Blueprint or else you can select particlula computes as well for eg: NFV_HOSTS: "compute-server-1, compute-server-2"</td></tr>
<tr><td>Tenant VLAN Ranges</td><td>Only with VTS/VLAN and VPP/VLAN</td></tr>
<tr><td>Enable Jumbo Frames</td><td>Check Box default is false</td></tr>
</table>
For Tenant Network Type Linux Bridge everything will remain the same but Tenant VLAN Ranges will be removed.</td></tr>
<tr><td>CEPH</td><td>Ceph has two pre-populated fields
CEPH Mode : By default Dedicated.
NOVA Boot from: Drop Down selection. You can choose Ceph or local.</td></tr>
<tr><td>GLANCE</td><td>By default populated for CEPH Dedicated with Store Backend value as CEPH.</td></tr>
<tr><td>CINDER</td><td>By default Populated for CEPH Dedicated with Volume Driver value as CEPH.</td></tr>
</table>

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>VMTP optional section will only be visible once VMTP is selected from Blueprint Initial Setup.</td><td>Check one of the check boxes to specify a VMTP network:
• Provider Network
• External Network

For the Provider Network complete the following:
<table><tr><td>Network Name field</td><td>Enter the name for the external network.</td></tr><tr><td>IP Start field</td><td>Enter the starting floating IPv4 address.</td></tr><tr><td>IP End field</td><td>Enter the ending floating IPv4 address.</td></tr><tr><td>Gateway field</td><td>Enter the IPv4 address for the Gateway.</td></tr><tr><td>DNS Server field</td><td>Enter the DNS server IPv4 address.</td></tr><tr><td>Segmentation ID field</td><td>Enter the segmentation ID.</td></tr><tr><td>Subnet</td><td>Enter the Subnet for Provider Network.</td></tr></table>For External Network fill in the following details:
<table><tr><td>Network Name field</td><td>Enter the name for the external network.</td></tr><tr><td>Network IP Start field</td><td>Enter the starting floating IPv4 address.</td></tr><tr><td>Network IP End field</td><td>Enter the ending floating IPv4 address.</td></tr><tr><td>Network Gateway field</td><td>Enter the IPv4 address for the Gateway.</td></tr><tr><td>DNS Server field</td><td>Enter the DNS server IPv4 address.</td></tr><tr><td>Subnet</td><td>Enter the Subnet for External Network.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>TLS This optional section will only be visible once TLS is selected from Blueprint Initial Setup Page.</td><td>TLS has two options:
External LB VIP FQDN - Text Field.
External LB VIP TLS - True/False. By default this option is false.</td></tr>
<tr><td>SwiftStack optional section will be visible once SwiftStack is selected from Blueprint Initial Setup Page. SwiftStack is only supported with KeyStonev2 . If you select Keystonev3, swiftstack will not be available for configuration.</td><td>Following are the options that needs to be filled for SwiftStack:<table>
<tr><td>Cluster End Point</td><td>IP address of PAC (proxy-account-container) endpoint.</td></tr>
<tr><td>Admin User</td><td>Admin user for swift to authenticate in keystone.</td></tr>
<tr><td>Admin Tenant</td><td>The service tenant corresponding to the Account-Container used by Swiftstack.</td></tr>
<tr><td>Reseller Prefix</td><td>Reseller_prefix as configured for Keysone Auth,AuthToken support in Swiftstack E.g KEY_</td></tr>
<tr><td>Admin Password</td><td>swiftstack_admin_password</td></tr>
<tr><td>Protocol</td><td>http or https ?</td></tr>
</table></td></tr>
</table>

Step 13 If **Syslog Export** or **NFVBENCH** is selected in **Blueprint Initial Setup** Page then, **Services Setup** page will be enabled for User to view. Following are the options under Services Setup Tab:

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Syslog Export</td><td>Following are the options for Syslog Settings:<table>
<tr><td>Remote Host</td><td>Enter Syslog IP Address</td></tr>
<tr><td>Protocol</td><td>Drop-down selection for UDP and TCP. By default its UDP</td></tr>
<tr><td>Facility</td><td>Defaults to local5</td></tr>
<tr><td>Severity</td><td>Defaults to debug</td></tr>
<tr><td>Clients</td><td>Defaults to ELK</td></tr>
<tr><td>Port</td><td>Defaults to 514 but can be modified by the User.</td></tr>
</table></td></tr>
</table>

NFVBENCH	enable checkbox which by default is **false**. Add Tor info connected to switch: • Select a TOR Switch Switch- (switch name) • Enter the port number, e.g. eth1/5 • NIC Ports: INT1 & INT2 Optional input, enter the 2 port numbers of the 4-port 10G Intel NIC at the management node used for NFVBench

Step 14 Click **Offlinevalidation** button to initiate an offline validation of the Blueprint.

Step 15 Once the **Offlinevalidation** is successful, **Save** option will be enabled for you which when clicked would redirect you to the **Blueprint Management** Page.

Creating a Blueprint using Upload Functionality

Before you begin

- You should have a YAML file (B series or C Series) on your system.
- Only one blueprint can be uploaded at a time. To create a blueprint off-line, please refer to the setup_data.yaml.B_Series_EXAMPLE or setup_data.yaml.C_Series_EXAMPLE.
- The respective keys in the sample YALM should match or the corresponding section will not be populated during upload.

Step 1 Log-in to **CISCO VIM Insight**.

Step 2 In the **Navigation** pane, expand the **Pre-Install** Section.

Step 3 Click **Blueprint** Setup.

Step 4 Click the **Browse** button in the **Blueprint Initial Setup** page.

Step 5 Click **Select**.

Step 6 Click on **Load** button in the **Insight UI Application**.
All the fields present in the YAML file will be uploaded to the respective fields in the UI.

Step 7 Provide a **Name for the Blueprint**.

Make sure the blueprint name is unique while saving it.

Step 8 Click **Offline Validation**.

- If all the mandatory fields in the UI are populated, then Offline Validation of the Blueprint will commence, or else a pop up message indicating the section of Blueprint Creation that has missing information error shows up.

Step 9 On Offline Blueprint Validation being successful , **Save Blueprint** and **Cancel** button will be enabled.

Note If the Blueprint Validation Fails, only the **Cancel** button will be enabled.

Activating a Blueprint in an Existing Pod with OpenStack Installed

Before you begin

You must have a POD which has an active Installation of OpenStack. If the OpenStack installation is in Failed State, then Insight UI will not be able to fetch the Blueprint.

Step 1 Go to the **landing page** of the Insight Login.

Step 2 Click **Register Management Node**.

Step 3 Enter the following details:

- Management Node IP Address.
- Management Node Name (Any friendly Name).
- REST API Password (/opt/cisco/ui_config.json).
- Description about the Management Node.
- POD Admin's Email ID.

A notification email will be sent to the email id entered during registration.

Step 4 Login using the same email id and password.

Step 5 In the Navigation pane, click **Pre-Install** > **Blueprint Management**.

In the **Blueprint Management** Page you will see **NEWSETUPDATA**.

This is the same setup data which was used by ciscovimclient to run the installation on the Management Node.

Downloading Blueprint

Before you begin

You must have atleast one blueprint (In any state Active/In-Active or In-progress), in the **Blueprint Management Page**.

Step 1 Log-in to **CISCO VIM Insight**.

Step 2 In the **Navigation** pane, expand the **Pre-Install Section**.

Step 3 Click **Blueprint Management**.

Step 4 Go-to **Download** button for any Blueprint under Action title. (**Download Button** > **Downward Arrow** (with tooltip Preview & Download YAML).

Step 5 Click the **Download** icon.

A pop to view the Blueprint in the YAML format will be displayed.

Step 6 Click the **Download** button at the bottom left of the pop-up window.
YAML will be saved locally with the same name of the Blueprint.

Validating Blueprint

Step 1 Log-in to **CISCO VIM Insight**.

Step 2 In the **Navigation** pane, expand the **Pre-Install Section**.

Step 3 Click **Blueprint Creation**.

Step 4 Upload an existing YAML, or create a **New Blueprint**.

Fill all the mandatory fields so that all Red Cross changes to **Green Tick**.

Step 5 Enter the name of the Blueprint.

Step 6 Click the **Offline Validation** button.
Only if the Validation is successful, the Insight will allow the user to save the blueprint.

What to do next

If you see any errors, then hyperlink will be created for those errors. Click on the link to be navigated to the page where error has been encountered.

Managing Post Install Features

Cisco VIM provides an orchestration that helps in lifecycle management of a cloud. VIM is responsible for pod management activities which includes fixing both hardware and software issues with one-touch automation. VIM Insight provides the visualization of the stated goal. As a result, it integrates with POST install features that Cisco VIM offers through its Rest API. These features are enabled only if there is an active Blueprint deployment on the pod.

Monitoring the Pod

In VIM 2.0, we use ELK (elasticsearch, logstash and Kibana) to monitor the openstack services, by cross-launching the Kibana dashboard.

To cross launch Kibana, complete the following instructions:

Step 1 In the **Navigation** pane, click **Post-Install** > **Monitoring**.
The **Authentication Required** browser pop up is displayed.

Step 2 Enter the **username** as Admin.

Step 3 Enter the ELK_PASSWORD password obtained from /root/installer-<tagid>/openstack-configs/secrets.yaml in the management node.
Kibana is launched in an I-Frame.

Note You can also view Kibana Logs in a new tab by clicking the **Click here to view Kibana logs in new tab** link.

Cross Launching Horizon

Horizon is the canonical implementation of OpenStack's Dashboard, which provides a web based user interface to OpenStack services including Nova, Swift and, Keystone.

Step 1 In the **Navigation** pane, click **Post-Install** > **Horizon**.

Step 2 Click the link **Click here to view Horizon logs in new tab**.
You will be redirected to Horizon landing page in a new tab.

Run VMTP

Run VMTP is divided in two sections:

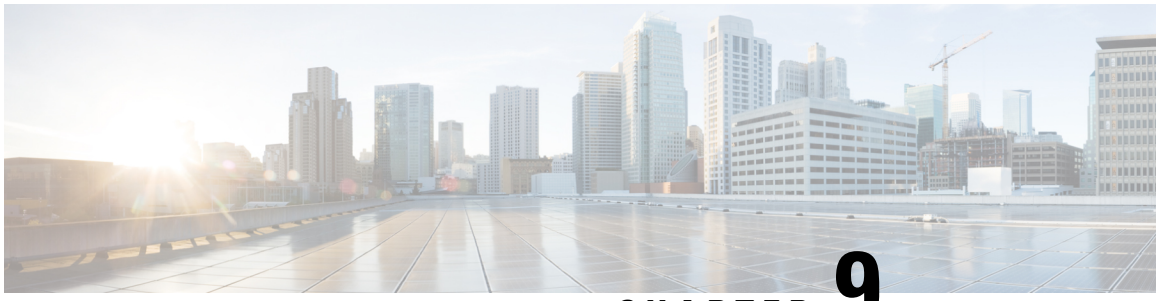
- **Results for Auto Run**: This will show the results of VMTP which was run during cloud deployment (Blueprint Installation).
- **Results for Manual Run**: Here you have an option to run the VMTP on demand. To run VMTP on demand just click **Run VMTP** button.

Note If VMTP stage was skipped/not-run during Blueprint Installation, this section of POST Install would be disabled for the user.

Run CloudPulse

Following are the tests supported in CloudPulse:

1. cinder_endpoint
2. glace_endpoint
3. keystone_endpoint
4. nova_endpoint
5. neutron_endpoint
6. rabbitmq_check
7. galera_check
8. ceph_check



CHAPTER 9

Managing Pod Through Cisco VIM Insight

The following topics tell you how to install and replace Cisco Virtual Infrastructure Manager (VIM) nodes using Cisco VIM Insight.

Managing Hardware

Management of your Cisco VIM pods includes adding, removing, or replacing the nodes.

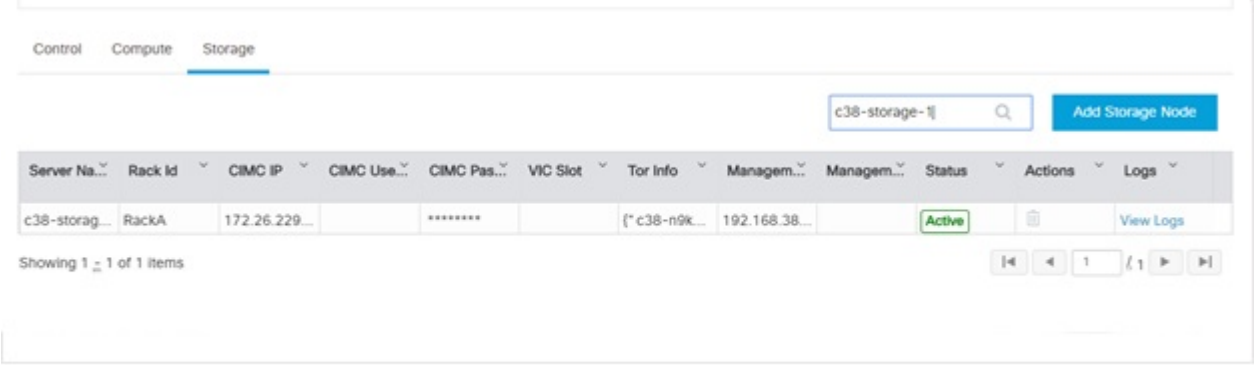
In a pod, multiple nodes cannot be changed at the same time. For example, if you want to replace two control nodes, you must successfully complete the replacement of the first node before you start to replace the second node. Same restriction applies for addition and removal of storage nodes. Only, in case of Compute Nodes you can add/remove multiple nodes together; however, there must always be one active compute node in the pod at any given point. VNF manager stays active and monitors the compute nodes thereby moving the VNFs accordingly as compute node management happens.

Note When you change a control, storage, or compute node in a Cisco VIM pod using Insight, it will automatically update the server and role in the active blueprint, as a result, your OpenStack deployment will change. When a node is removed from Cisco VIM, sensitive data may remain on the drives of the server. Administrator is advices to use Linux tools to wipe the storage server before using the same server for another purpose. The drives used by other application server should also be wiped out before being added to Cisco VIM.

Searching Compute and Storage Nodes

This functionality allows you to search the Compute and Storage nodes by server names only. The search result is generated or shows an empty grid if there are no results.

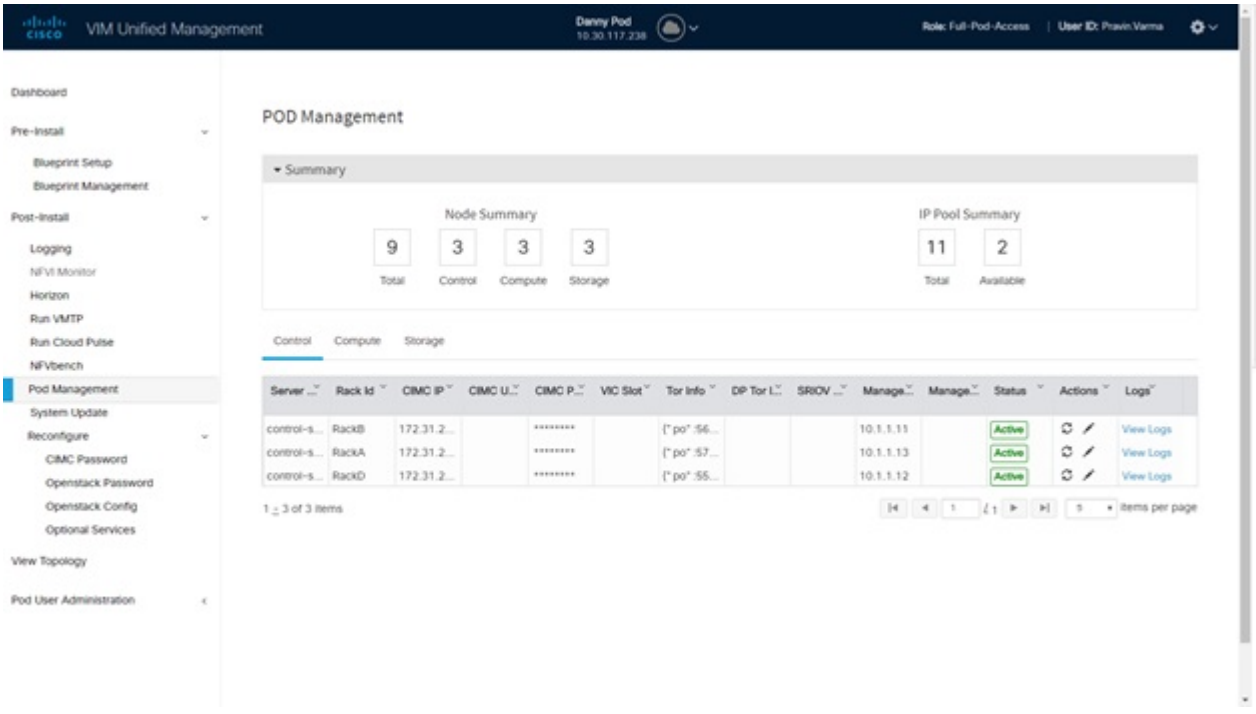
Figure 10: Search Storage Nodes



POD Management

Cisco VIM allows the admin to perform pod life-cycle management from a hardware and software perspective. VIM provides the ability to power on/off compute node, add, remove or replace nodes based on the respective roles when the nodes of a given pod corrupts at times.

Figure 11: POD Management



Pod Management page has two sections–

1. **Node Summary:** Node Summary section shows how many nodes are available and the detailed count of Control, Compute and Storage type.

2. **IP Pool Summary:** IP Pool Summary section shows the Total Pool Summary and the current available pool count.

The opeartions performed on the running pod are:

Replace Control Nodes: We do not support double fault scenarios, replacement of one controller at a time is supported.

Add Computes/Storage Nodes: N-computes nodes can be replaced simultaneously; however at any given point, at least one compute node ahs to be active.

Power On/ Off compute Nodes: You can Power On or Power Off compute node. At least one compute node must be powered on.

Remove Compute/Storage Nodes: You can add one node at a time, given that we run Ceph as a distributed storage offering.

Add Pool: You can increase pool size at any time.

Managing Storage Nodes

Before you add or remove a storage node, review the following guidelines for Managing Storage Nodes.

- **Required Number of Storage Nodes**: A Cisco VIM pod must have a minimum of three and a maximum of 20 storage nodes. If your pod has only two storage nodes, you cannot delete another storage node until you add another storage node. If you have fewer than three storage nodes, you can add one node at a time until you get to 20 storage nodes.
- **Validation of Nodes**: When you add a storage node to a pod, Cisco VIM Insight validates that all the nodes in the pod meet the minimum requirements and are in active state. If you have a control or compute node in a faulty state, you must either correct, delete or replace that node before you can add a storage node.
- **Update Blueprint**: When you add or delete a storage node, Insight will update the blueprint for the Cisco VIM pod.
- **Storage Node Logs**: You can access the logs for each storage node from the link in the Log column on the **Storage Nodes** tab.

Failed Addition or Deletion of Storage Node: If your attempt to add or delete a storage node fails, complete the following instructions:

Step 1 Choose the storage node in the list and click **Delete**.

Step 2 If the delete fails, click **Clear DB Entry** to remove the failed node from the database.

If you do not remove the node from the database, you cannot add another node to bring the number of nodes to three.

Adding Storage Node

You cannot add more than one storage node at a time. Complete the following instructions to add a storage node:

Before you begin

- Remove the non-functional storage node from the pod. You can have maximum 20 storage nodes in a Cisco VIM pod.
- Ensure that the server for the new storage node is in powered state in OpenStack for C Series.

Step 1 In the **Navigation pane**, choose **Post-Install** > **Pod Management**.

Step 2 Click the + icon next to **Sufficient Pool Present** next to **Add/Remove Storage Node**.

- Sufficient POOL check is automatically filled by Insight.
- It checks the number of IP present in the POOL for Management or Provision.
- If number of nodes is equal to pool size then the option would be disabled and you need to increase the pool size by clicking + icon.
- Validation check to see if the IP is valid and belongs to the same subnet of the existing IP Pool are applied by insight before you add IPs to Management/Provision network.

Step 3 For C Series, add the following details:

- **Server Name**: Name for the Storage Server to be added.
- **Rack ID**: Enter the Rack ID. (Accepts String format).
- **CIMC IP**: Enter the CIMC IP.
- **CIMC User Name**: User name for the CIMC.
- **CIMC Password**: Enter the password for the CIMC
- **VIC Slot**: Enter the VIC Slot (Optional).

Step 4 For B Series, add the following details:

- **Server Name**: Name for the Storage Server to be added.
- **Rack ID**: Enter the Rack ID. (Accepts String format).
- **Rack Unit ID**: Enter the Rack Unit ID.

If all mandatory fields are filled in correctly then **Add Storage Button** will be enabled.

Step 5 Click **Add Storage** button.
Add node initialized message will be displayed.

Step 6 To view logs, click **Click here to see the logs** under Logs column.
The status of the POD will change to Active.

Step 7 Two kinds of failure may occur:

- **Add Node Pre-Failed**: When addition of node failed before the bare-metal stage (step 4) the Active Blueprint will be modified but the Node is not yet added in the Cloud. If you press **X** Icon, then Insight will delete the node information form the Blueprint and the state would be restored.

- **Add Node Post-Failed**: When addition of node failed after the bare-metal stage (step 4) the Active Blueprint will be modified and the node is registered in the cloud. If you press **X** Icon, then Insight will first delete the node from the Blueprint and then node removal from cloud would be initiated.

You can view the logs for this operation under **Logs** column.

Deleting Storage Node

You cannot delete more than one storage node at a time.

Step 1 In the Navigation pane, choose **Post-Install** > **POD Management**.

Step 2 Click **X** adjacent to the storage node you want to delete.

Step 3 **Node Removal Initiated successfully** message will be displayed.

To view logs, click **Click here to see the logs** under logs column.

- If deletion is successful, the storage node will be removed from the list under **Add/Remove storage Node**.
- In deletion failed, a new button **Clear Failed Nodes** will be displayed. Click the button to remove the node form cloud and Blueprint.

Managing Compute Nodes

Before you add or remove a compute node, review the following guidelines:

- **Required Number of Compute Nodes**: A Cisco VIM pod must have a minimum of one compute node and a maximum of 96 compute nodes. If your pod has only one compute node, you cannot delete that node until you add another compute node.
- **Update Blueprint**: When you add or remove a compute node, Insight will update the blueprint for the Cisco VIM pod.
- **Compute Node Logs**: You can access the logs for each compute node from the link in the Log column on the Compute Nodes table.

Adding Compute Node

Complete the instructions, to add a compute node:

Before you begin

Ensure that the server for the new compute node is in powered state in OpenStack. You can add more than one compute node at a time.

Step 1 In the Navigation pane, click **Post-Install** > **Pod Management**.

Step 2 Click + icon next to **Sufficient Pool Present** adjacent to **Add/Remove Storage Node**.

- Sufficient POOL check is automatically filled by Insight.
- It checks the number of IP present in the POOL for Management or Provision.
- If number of nodes is equal to pool size then the option would be disabled and you need to increase the pool size by clicking + icon.
- Validation check to see if the IP is valid and belongs to the same subnet of the existing IP Pool are applied by insight before you add IPs to Management/Provision network.

Step 3 For C Series, add the following details:

- **Server Name**: Name for the Storage Server to be added.
- **Rack ID**: Enter the Rack ID. (Accepts String format).
- **CIMC IP**: Enter the CIMC IP.
- **CIMC User Name**: User name for the CIMC.
- **CIMC Password**: Enter the password for the CIMC
- **VIC Slot**: Enter the VIC Slot (Optional).

Step 4 For B Series, add the following details:

- **Server Name**: Name for the Storage Server to be added.
- **Rack ID**: Enter the Rack ID. (Accepts String format).
- **Rack Unit ID**: Enter the Rack Unit ID.
- **Chassis ID**: Enter the Chassis ID. Range for Chassis ID is 1-24.
- **Blade ID**: Enter the Blade ID. Range for Blade ID is 1-8.
- **CIMC Password**: Enter the CIMC Password.

  If all mandatory fields are filled in correctly then click **Save**

Note: Add Compute process can initiate multiple add of compute nodes. Fill in the mandatory fields to save new compute node or press cancel to exit message will be displayed.

Step 5 You may perform one among these steps mentioned below:

- Clicking **Cancel** will display the compute node information listed in the table and **Add Compute Node** button is enabled.
- If you feel you have filled in a wrong entry for the compute node information, then click **Delete**. This will delete the entry from the table as this information is not added in the Blueprint.
- Clicking **Add Compute** button, displays **Add node initialized** message.

Step 6 To view logs, click **Click here to see the logs** under Logs column.
The status of the POD will change to Active.

Step 7 Two kinds of failure may occur:

- **Add Node Pre-Failed**: When addition of node failed before the bare-metal stage (step 4) the Active Blueprint will be modified but the Node is not yet added in the Cloud. If you press **X** Icon, then Insight will delete the node information form the Blueprint and the state would be restored.
- **Add Node Post-Failed**: When addition of node failed after the bare-metal stage (step 4) the Active Blueprint will be modified and the node is registered in the cloud. If you press **X** Icon, then Insight will first delete the node from the Blueprint and then node removal from cloud would be initiated.

You can view the logs for this operation under **Logs** column.

Deleting Compute Node

A compute node is deleted due to a hardware failure. You can delete one compute node at a time. If your pod has only one compute node, you cannot delete that node until you add another compute node.

Step 1 In the **Navigation** pane, choose **Post-Install** > **POD Management**.

Step 2 Click **X** for the compute node to be deleted.
Node Removal Initiated successfully message will be displayed.

Step 3 To view the logs, click **Click here to see the logs** under Logs column.

- If deletion is successful, you will not be able to view that compute node in the list under **Add/Remove Compute Node**.
- If deletion has failed, a new button **Clear Failed Nodes** will be visible.

Step 4 Click **Clear Failed Nodes** to remove the node form cloud and Blueprint.

Managing Control Nodes

Before you replace a control node, review the following guidelines:

- **Required Number of Control Nodes**: A Cisco VIM pod must have three control nodes and you can only replace one node at a time.
- **Validation of Nodes**: When you replace a control node, Cisco VIM Insight validates if all the other nodes in the pod meet the minimum requirements and are in active state. If you have a storage or compute node in a faulty state, you must correct the faulty state or delete or replace that node before you can replace the control node.
- **Update Blueprint**: When you replace a control node, Insight will update the Active blueprint for the Cisco VIM pod.
- **Control Node Logs**: You can access the logs for each control node from the link in the **Logs** column of Control Nodes table.

Replacing Control Node

You can replace only one control node at a time.

Step 1 In the Navigation pane, click **Post-Install** > **Pod Management**.

Step 2 Click **Replace** under the Action column of the **Replace Control** table.

Step 3 On success, **Replace Node Initiated Successfully** message will be displayed.

Step 4 You can view the logs in the **Logs** column on the Control Nodes table.

What to do next

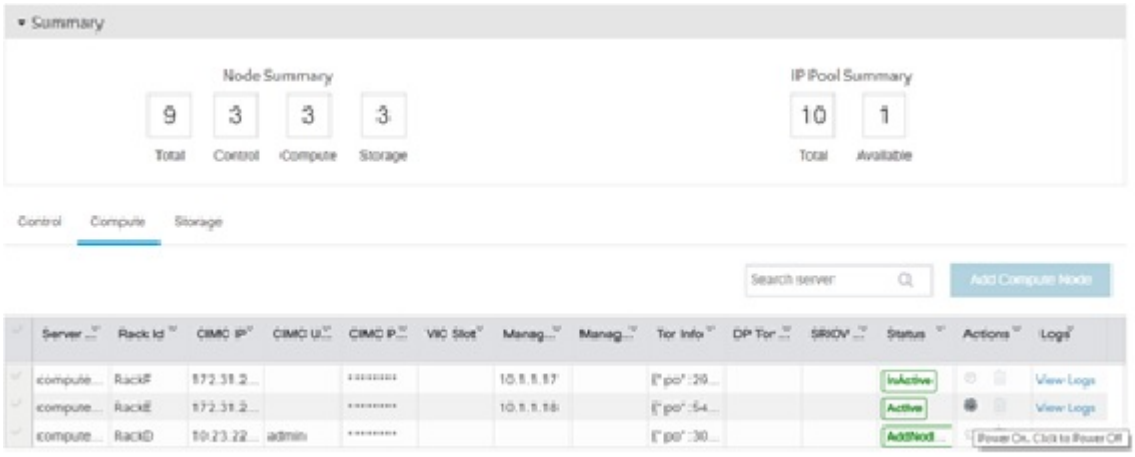
If the replacement of the control node fails, do the following:

- Click on the link in the Logs column.
- Check the logs to determine the cause of the failure.
- Correct the issue and attempt to replace the control node again.

Power Management

Compute node can be powered on or powered off from the Compute Tab in Pod Management section. There is a power button associated with each compute with information provided as tooltip when you hover on that icon. To power on/off multiple compute node, user can click on the power button located to the left of delete button. This power button is disabled by default. When user selects the compute nodes by clicking on checkboxes in the first column, the corresponding power button will be enabled.

Power ON a Compute Node



Click the **Power** button of a compute node which is currently powered off (grey icon). Message showing Power on is displayed. The *power* button starts blinking with the tooltip message as 'Powering on'. During, this time **Add Compute** button is disabled.

Figure 12:

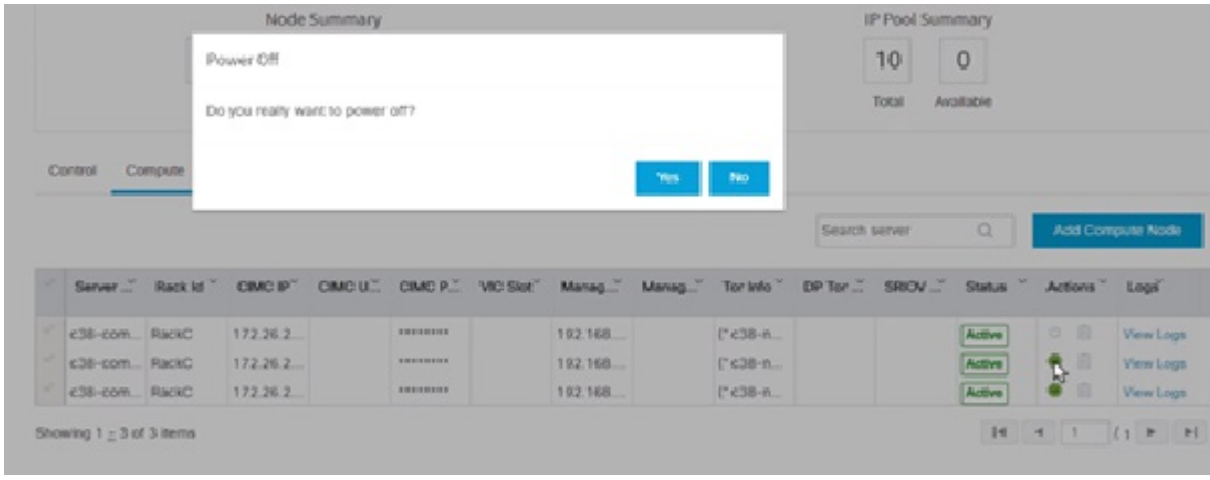
Once, the compute node is *Powered on,* the icon stops blinking and the color of power icon changes to green. The **Add Compute Node** button is enabled immediately.

Powering Off Compute Node

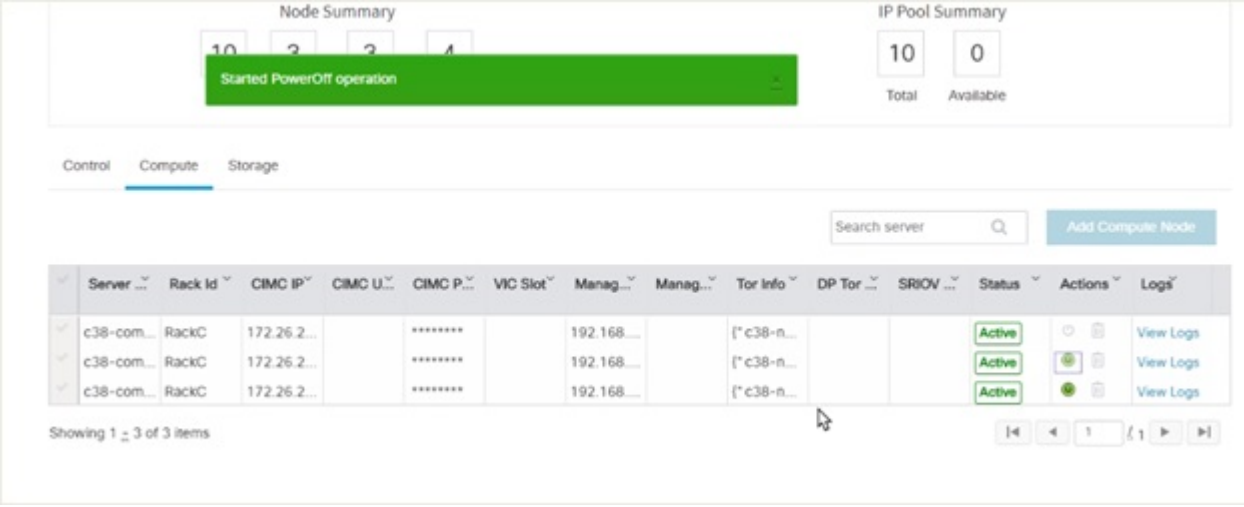
Note You cannot power off all the Compute nodes. There must be at least one Compute node that is in the On state.

Follow these steps to power off a Compute node:

1. Click the **Compute** tab.
2. In the Pod Management area, under the Actions column, click the **Power** button of the Compute node that you want to power off.



3. Click **Yes** in the confirmation dialog box.



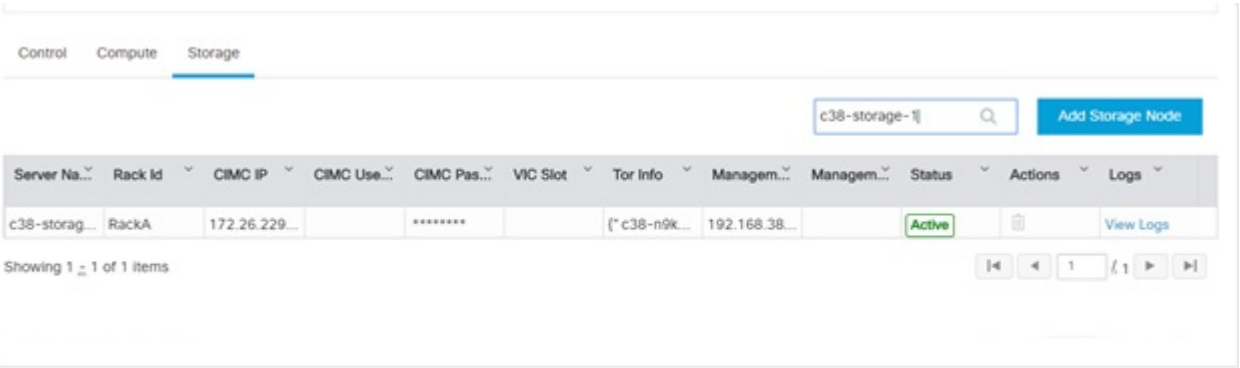
It may take a few minutes for the Compute node to power off. The tooltip of the power button displays the status of the Compute node. Once the compute node is powered off, the Power button stops blinking and its color changes to grey.

Note If there is only one compute node in the grid, and you try to power off it, a message *Last compute node can't be powered off* is displayed. Also, when you power off the last available compute node in the list of nodes, then the message *At least one compute node should be powered on* is displayed.

Searching Compute and Storage Nodes

This functionality allows you to search the Compute and Storage nodes by server names only. The search result is generated or shows an empty grid if there are no results.

Figure 13: Search Storage Nodes



Managing Software

Software management of your Cisco VIM pods includes software update, reconfigure of openstack services and password, etc.

VIM Software Update

As part of the lifecycle management of the cloud, VIM has the ability to bring in patches (bug fixes related to code, security, etc.), thereby providing cloud management facility from software point of view. Software update of the cloud is achieved by uploading a valid tar file, following initiation of a System Update form the Insight as follows:

Step 1 In the Navigation pane, click **Post-Install** > **System Update**.

Step 2 Click **Browse** and select the valid tar file.

Step 3 Click **Open**.

Step 4 Click **Upload and Update**.
Update started Successfully message will be displayed.

Step 5 Update status will be shown as **ToUpdate**.

Click the hyperlink to view the reconfigure logs for install logs.

Reconfigure status will be available on the page or the dashboard under **POD Operation** details.

What to do next

System Update has been initiated message will be displayed. Logs front-ended by hyperlink will be in the section below in-front of **Update Logs** which shows the progress of the update. During the software update, all other pod management activities will be disabled. Post-update, normal cloud management will commence. Once update has completed you will see the status of update in the box below.

If log update fails, **Auto-RollBack** will be initiated automatically.

If log update is Successful, you will have two options to be performed:

1. **Commit**—To proceed with the update.
2. **RollBack**—To cancel the update.

If Auto-rollback fails during software update fails through Insight UI, it is advised that the administrator contact Cisco TAC for help. Do not re-try the update or delete the new or the old installer workspace.

Reconfigure Password

- **Regenerate all passwords**: Click **Regenerate all passwords** checkbox and click **Set Password**. This will automatically regenerate all passwords in alphanumeric format.
- **Regenerate single or more password**: This will set a specific password by doing an inline edit for any service like Horizon's ADMIN_USER_PASSWORD. Double click on the filed under Password and enter the password to enable **Set Password** button.

During the reconfiguration of password, all other pod management activities will be disabled. Post-update, normal cloud management will commence. If the reconfigure of the password fails, all subsequent pod management operations will be blocked. It is advised to contact Cisco TAC to resolve the situation through CLI.

Reconfigure OpenStack Services, TLS Certificates and ELK Configurations

Cisco VIM supports the reconfiguration of OpenStack log level services, TLS certificates, and ELK configuration. Listed below are the steps to reconfigure the OpenStack and other services:

Step 1 In the Navigation pane, click **Post-Install** > **Reconfigure Openstack Config**.

Step 2 Click on the specific item that you want to change and update. For example: to update TLS certificate click the path to certificate location.

Step 3 Enter **Set Config** to commence the process.

What to do next

During the reconfiguration process, all other pod management activities will be disabled. Post-update, normal cloud management will commence. If reconfigure of OpenStack Services fail, all subsequent pod management operations will be blocked. It is advised to contact Cisco TAC to resolve the situation through CLI.

Reconfiguring CIMC Password through Unified Management

Cisco VIM allows you to Update the cimc_password in the CIMC-COMMON section, and/or the individual cimc_password for each server and then run the update password option.

You need to match the following Password rule to update the Password:

- Must contain at least one lower case letter.
- Must contain at least one upper case letter.
- Must contain at least one digit between 0 to 9.
- One of these special characters !$#@%^-_+=*&
- Your password has to be 8 to 14 characters long.

Before you begin

You must have a C-series pod up and running with Cisco VIM to reconfigure CIMC password.

Note Reconfigure CIMC password section will be disabled if the pod is in failed state as indicated by ciscovim install-status.

Step 1 Log-in to **CISCO VIM Insight**.

Step 2 In the navigation pane, select **Post-Install**.

Step 3 Click **Reconfigure CIMC Password**.

Step 4 You can reconfigure the CIMC Password at global level by adding new CIMC_COMMON Password or to reconfigure CIMC Password for individual servers double click the server password you want to edit.

Step 5 Click **Reconfigure**to initiate reconfigure process.

Reconfigure Optional Services

Cisco VIM offers optional services such as heat, migration to Keystone v3, NFVBench, NFVIMON, etc, that can be enabled post-pod deployment. These services can be enabled in one-shot or selectively.

Listed below are the steps to enable optional services:

Step 1 In the Navigation pane, click **Post-Install** > **Reconfigure Optional Services**.

Step 2 Choose the right services and update the fields with the right values.

Step 3 Click **Offline validation**. Once offline validation is successful.

Step 4 Click **Reconfigure** to commence the process.

During the reconfiguration process, all other pod management activities will be disabled. Post-update, normal cloud management will commence.

If reconfigured OpenStack Services fail, all subsequent pod management operations are blocked. Contact Cisco TAC to resolve the situation through CLI.

Note All reconfigure operation feature contains repeated deployment true or false.

- Repeated re-deployment true - Feature can be re-deployed again.
- Repeated re-deployment false- Deployment of feature allowed only once.

Deployment Status :

Optional Features	Repeated re-deployment Option
APICINFO	True
EXTERNAL_LB_VIP_FQDN	False
EXTERNAL_LB_VIP_TLS	False
INSTALL_MODE	True
HTTP_PROXY & HTTPS_PROXY	True
LDAP	True
NETWORKING	True
NFVBENCH	False
NFVIMON	False

Optional Features	Repeated re-deployment Option
PODNAME	False
PROVIDER_VLAN_RANGES	True
SWIFTSTACK	True
SYSLOG_EXPORT_SETTINGS	False
TENANT_VLAN_RANGES	True
TORSWITCHINFO	False
VIM _ ADMINS	True
VMTP	False
VTS_PARAMETERS	False
AUTOBACKUP	` True
Heat	False
Keystone v3	False

Reconfiguring Optional Features Through Unified Management

Step 1 Log-in to Cisco VIM UM.

Step 2 In the **Navigation** pane, expand the **Post-Install Section**.

Step 3 Click **Reconfiguring Optional Feature through UM**.

Step 4 On the **Reconfiguring Optional Feature through UM** page of the Cisco VIM UM, complete the following fields:

Name	Description
Heat check box	• Enable **Heat**. • Click **Offline Validation** . • When Offline Validation is successful, click **Reconfigure** to commence the process..
Keystone v3 check box	• Enable **Keystone v3**. • Click **Offline Validation** . • When Offline Validation is successful, click **Reconfigure** to commence the process.

Name	Description
ENABLE_ESC_PRIV	• Enable **ENABLE_ESC_PRIV** . • Click **Offline Validation** . • When Offline Validation is successful, click **Reconfigure** to commence the process.
Autobackup check box	• Enable/Disable **Autobackup**. • Click **Offline Validation** . • When Offline Validation is successful, click **Reconfigure** to commence the process.
External LB VIP TLS check box	• Enable **External LB VIP TLS**. • Click **Offline Validation** . • When Offline Validation is successful, click **Reconfigure** to commence the process.
External LB VIP FQDN check box	• Enter Input as a string. • Click **Offline Validation** . • When Offline Validation is successful, click **Reconfigure** to commence the process.
Pod Name	• Enter Input as a string. • Click **Offline Validation** . • When Offline Validation is successful, click **Reconfigure** to commence the process.
Tenant Vlan Ranges	• Augment tenant vlan ranges input. For Example: 3310:3315. • Click **Offline Validation** . • When Offline Validation is successful, click **Reconfigure** to commence the process.
Provider VLAN Ranges	• Enter input to tenant vlan ranges. For Example: 3310:3315. • Click **Offline Validation** . • When Offline Validation is successful, click **Reconfigure** to commence the process.

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Install Mode</td><td>• Select Connected or Disconnected, any one form the drop-down list.
• Click Offline Validation .
• When Offline Validation is successful, click Reconfigure to commence the process.</td></tr>
<tr><td>Syslog Export Settings</td><td>Following are the options for Skylog Settings:<table><tr><td>Remote Host</td><td>Enter Syslog IP Address.</td></tr><tr><td>Facility</td><td>Defaults to local5</td></tr><tr><td>Severity</td><td>Defaults to debug</td></tr><tr><td>Clients</td><td>Defaults to ELK</td></tr><tr><td>Port</td><td>Defaults to 514 but is modified by the User.</td></tr><tr><td>Protocol</td><td>Supports only UDP</td></tr></table>• Click Offline Validation .
• When Offline Validation is successful, click Reconfigure to commence the process.</td></tr>
<tr><td>Configure ToR checkbox</td><td>True or False. Default is false.</td></tr>
</table>

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>ToR Switch Information</td><td>Click + to add information for ToR Switch.
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Name</td><td>ToR switch name.</td></tr>
<tr><td>Username</td><td>ToR switch username.</td></tr>
<tr><td>Password</td><td>ToR switch Password.</td></tr>
<tr><td>SSH IP</td><td>ToR switch SSH IP Address.</td></tr>
<tr><td>SSN Num</td><td>ToR switch ssn num. output of show license host-id.</td></tr>
<tr><td>VPC Peer Keepalive</td><td>Peer Management IP. You need not define if there is no peer.</td></tr>
<tr><td>VPC Domain</td><td>Need not define if there is no peer.</td></tr>
<tr><td>VPC Peer port</td><td>Interface for vpc peer ports.</td></tr>
<tr><td>VPC Peer VLAN Info</td><td>vlan ids for vpc peer ports (optional).</td></tr>
<tr><td>BR Management Port Info</td><td>Management interface of the build node.</td></tr>
<tr><td>BR Management PO Info</td><td>Port channel number for the management interface of the build node.</td></tr>
</table>
Click Save
• Click Offline Validation .
• When Offline Validation is successful, click Reconfigure to commence the process.</td></tr>
</table>

Note When setup data is ACI VLAN with TOR then reconfigure options are:

<table>
<tr>
<td>TORSwitch Information mandatory table if you want to enter ToR information</td>
<td>Click + to add information for ToR Switch.
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Host Name</td><td>ToR switch name.</td></tr>
<tr><td>VPC Peer Keepalive</td><td>Peer Management IP.</td></tr>
<tr><td>VPC Domain</td><td>Do not define if there is no</td></tr>
<tr><td>Node ID</td><td>Integer, unique across all switches</td></tr>
</table>
Click Save
• Click Offline Validation .
• When Offline Validation is successful, click Reconfigure to commence the process.</td>
</tr>
<tr>
<td>NFV Bench</td>
<td>Enable check box which by default is false.
Add Tor info connected to switch:
• Select a TOR Switch and Enter the Switch name.
• Enter the port number. For example: eth1/5
• NIC Ports: INT1 and INT2 optional input, enter the 2 port numbers of the 4-port 10G Intel NIC at the management node used for NFVBench.
• Click Offline Validation .
• When Offline Validation is successful, click Reconfigure to commence the process.
Note If ToR is already present in Setup-data or already deployed. Then no need add Tor info, by default ToR info switchname is mapped in NFV bench.</td>
</tr>
</table>

Swiftstack

SwiftStack is only supported with Keystone v2. If you select Keystone v3, swiftstack will not be available for configuration.

Cluster End Point	IP address of PAC (proxy-account-container) endpoint.
Admin User	Admin user for swift to authenticate in keystone.
Admin Tenant	The service tenant corresponding to the Account-Container used by Swiftstack.
Reseller Prefix	Reseller_prefix as configured for Keystone Auth,AuthToken support in Swiftstack E.g KEY_
Admin Password	swiftstack_admin_password
Protocol drop-down list	http or https

- Click **Offline Validation** .
- When Offline Validation is successful, click **Reconfigure** to commence the process.

<table>
<tr>
<td>LDAP with Keystone v3</td>
<td>
<table>
<tr><td>Domain Name field</td><td>Enter the Domain name.</td></tr>
<tr><td>Object Class for Users field</td><td>Enter a string as input.</td></tr>
<tr><td>Object Class for Groups</td><td>Enter a string.</td></tr>
<tr><td>Domain Name Tree for Users</td><td>Enter a string.</td></tr>
<tr><td>Domain Name Tree for Groups field</td><td>Enter a string.</td></tr>
<tr><td>Suffix for Domain Name field</td><td>Enter a string.</td></tr>
<tr><td>URL field</td><td>Enter a URL with port number.</td></tr>
<tr><td>Domain Name for Bind User field</td><td>Enter a string.</td></tr>
<tr><td>Password field</td><td>Enter Password as string format.</td></tr>
<tr><td>User Filter</td><td>Enter filter name as string.</td></tr>
<tr><td>User ID Attribute</td><td>Enter a string.</td></tr>
<tr><td>User Name Attribute</td><td>Enter a string.</td></tr>
<tr><td>User Mail Attribute</td><td>Enter a string.</td></tr>
<tr><td>Group Name Attribute</td><td>Enter a string.</td></tr>
</table>

Click Offline Validation .
When Offline Validation is successful, click Reconfigure to commence the process.

</td>
</tr>
</table>

<table>
<tr><td>NFV Monitoring</td><td>Followings are the field values for NFV Monitoring:<table>
<tr><td>Master Admin IP field.</td><td>Enter Input as IP format.</td></tr>
<tr><td>Collector Management IP field</td><td>Enter Input as IP format.</td></tr>
<tr><td>Collector VM1 info</td><td></td></tr>
<tr><td>Host Name field</td><td>Enter Host Name as a string.</td></tr>
<tr><td>CCUSER password field</td><td>Enter Password.</td></tr>
<tr><td>Password field</td><td>Enter password.</td></tr>
<tr><td>Admin IP field</td><td>Enter Input as IP format.</td></tr>
<tr><td>Management IP field</td><td>Enter Input as IP format.</td></tr>
<tr><td>Collector VM2 info</td><td></td></tr>
<tr><td>Host Namefield</td><td>Enter a string.</td></tr>
<tr><td>CCUSER field</td><td>Enter Password.</td></tr>
<tr><td>Management IP field</td><td>Enter Input as IP format.</td></tr>
<tr><td>Dispatcher</td><td></td></tr>
<tr><td>Rabbit MQ Username Field</td><td>Enter a string.</td></tr>
</table>
• Click Offline Validation .
• When Offline Validation is successful, click Reconfigure to commence the process.</td></tr>
<tr><td>VTS Parameter</td><td>Following are the fields to reconfigure for VTS parameters<table>
<tr><td>VTC SSH Username field.</td><td>Enter the string.</td></tr>
<tr><td>VTC SSH Username field.</td><td>Enter the password.</td></tr>
</table>
• Click Offline Validation .
• When Offline Validation is successful, click Reconfigure to commence the process.</td></tr>
</table>

<table>
<tr>
<td>VMTP</td>
<td>Check one of the check boxes to specify a VMTP network:
• Provider Network
• External Network

For the Provider Network complete the following:
<table><tr><td>Network Name field.</td><td>Enter the name for the external network.</td></tr><tr><td>IP Start field.</td><td>Enter the starting floating IPv4 address.</td></tr><tr><td>IP End field.</td><td>Enter the ending floating IPv4 address.</td></tr><tr><td>Gateway field</td><td>Enter the IPv4 address for the Gateway.</td></tr><tr><td>DNS Server field.</td><td>Enter the DNS server IPv4 address.</td></tr><tr><td>Segmentation ID field.</td><td>Enter the segmentation ID.</td></tr><tr><td>Subnet</td><td>Enter the Subnet for Provider Network.</td></tr></table>
For External Network fill in the following details:
<table><tr><td>Network Name field.</td><td>Enter the name for the external network.</td></tr><tr><td>Network IP Start field.</td><td>Enter the starting floating IPv4 address.</td></tr><tr><td>Network IP End field.</td><td>Enter the ending floating IPv4 address.</td></tr><tr><td>Network Gateway field</td><td>Enter the IPv4 address for the Gateway.</td></tr><tr><td>DNS Server field.</td><td>Enter the DNS server IPv4 address.</td></tr><tr><td>Subnet</td><td>Enter the Subnet for External Network.</td></tr></table>
• Click Offline Validation .
• When Offline Validation is successful, click Reconfigure to commence the process.</td>
</tr>
</table>

<table>
<tr>
<td>Networking

In Reconfigure optional services networking, you can reconfigure IP tables, or add http_proxy/https_proxy.</td>
<td>To reconfigure networking, update the relevant information:

<table>
<tr><td>IP Tables</td><td>Click Add(+) to add a table. Enter input as subnet format.

E.g. 12.1.0.1/2</td></tr>
<tr><td>http_proxy_server</td><td>Enter HTTP_PROXY_SERVER

E.g. <a.b.c.d:port></td></tr>
<tr><td>https_proxy_server</td><td>Enter HTTP_PROXY_SERVER

E.g. <a.b.c.d:port></td></tr>
</table>

• Click Save.
• Click Offline Validation.
• When Offline Validation is successful, click Reconfigure to commence the process.</td>
</tr>
<tr>
<td>APICINFO

Note Reconfigure optional services only APIC hosts can be reconfigure.</td>
<td>To reconfigure APICINFO, follow the process:

• Enter input for APIC hosts format. <ip1|host1>:[port] or eg.12.1.0.12
• Click Save.
• Click Offline Validation.
• When Offline Validation is successful, click Reconfigure to commence the process.

Note APIC hosts can be reconfigure minimum 1 host and max 3 but not 2 hosts.</td>
</tr>
<tr>
<td>Vim_admins</td>
<td>To reconfigure vim_admins, follow the process:

• To add a new root user, Click + and add the Username and admin hash password (Starting with $6).
• To remove the existing user, Click -.
• When Offline Validation is successful, click Reconfigure to commence the process.</td>
</tr>
</table>

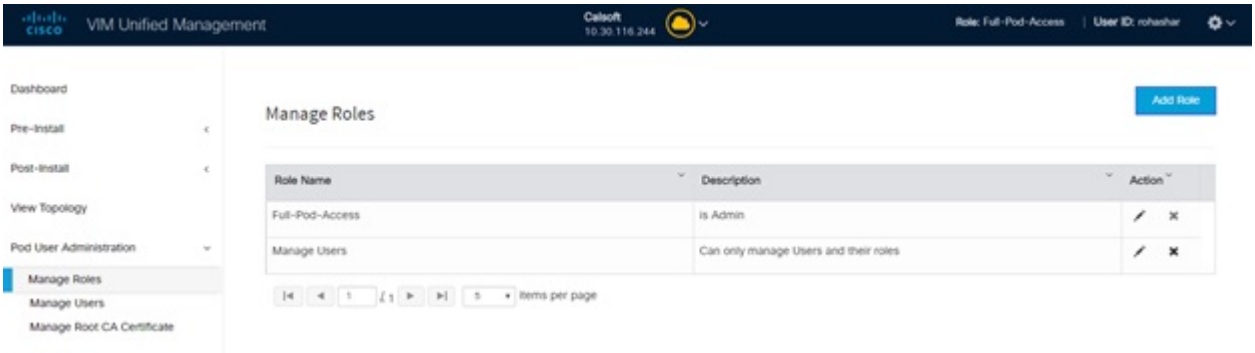
Pod User Administration

Cisco VIM UM offers Users (Pod Admins or Pod Users) to manage Users and roles that are associated with them.

Managing Roles

User can create multiple Roles and assign them to other pod users. System has a default role that is named as Full-Pod-Access which is assigned to the person who registers the Pod.

Manage Roles



Step 1 Click **Login as POD User.**

Step 2 Navigate to **Pod User Administration** and click **Manage Roles**. By default you see full-pod-access role in the table.

Step 3 Click **Add New Role** to create a new role.

Step 4 Complete the following fields in the **Add Roles** page in Cisco VIM UM:

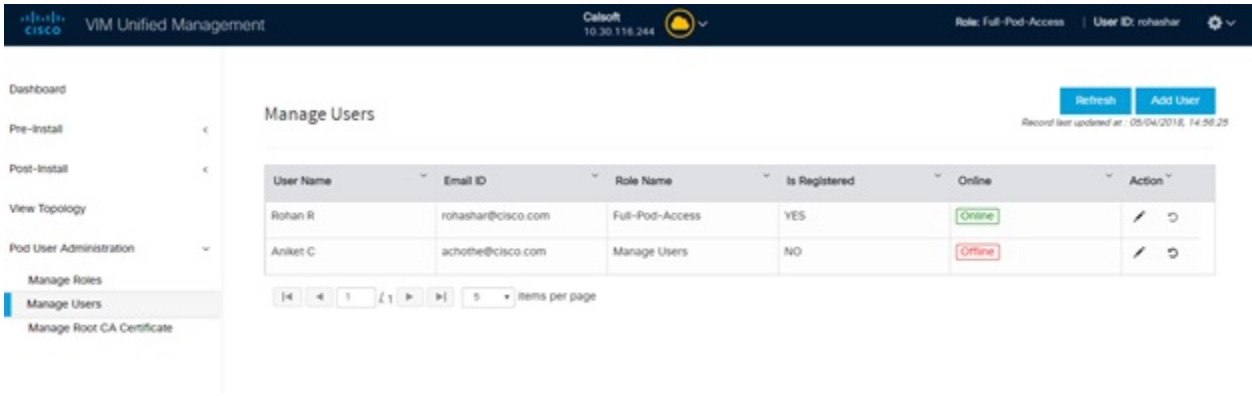
Field Name	Field Description
Role	Enter the name of the role.
Description	Enter the description of the role.
Permission	Check the **Permission** check box to select the permission.
Click **Save**.	Once the Blueprint is in Active state all the permissions are same for C-series and B-series Pods other than Reconfigure CIMC Password which is missing for B-series Pod.

Note Permissions are divided in the granular level where viewing Dashboard is the default role that is implicitly added while creating a role.

Note Permissions are divided in the granular level where viewing **Dashboard** is the default role that is implicitly added while creating a role.

Managing Users

This section allows you to add the users. It shows all the users associated with the Pod. You can check the online status of all the user. Click**Refresh** on upper right corner to check the status.



To add a new user:

Step 1 Click **Login as POD User**.

Step 2 Navigate to **POD User Administration** and click **Manage Users** .

Step 3 Click **Add Users** to add a new user.

Step 4 Complete the following fields in the **Add Users** pane of the Cisco VIM Insight:

Field Name	Field Description
Email ID	Enter the Email ID of the User.
User Name	Enter the User Name if the User is new. If the User is already registered to the Insight the User-Name gets auto-populated.
Role	Select the Role from the drop-down list.

Step 5 Click **Save** Once the Blueprint is in Active state all the permissions are same for C-series and B-series Pods other than Reconfigure CIMC Password which is missing for B-series Pod.

Revoke Users

User with Full-Pod-Access or Manage Users permission can revoke other users from the specific Pod.

To revoke users:

Step 1 Click **Undo** icon. A confirmation pop up will appear.

Step 2 Click **Proceed** to continue.

Note Self revoke is not permitted. After revoking the another user, if the user is not associated with any other pod then the revoked user will be auto deleted from the system.

Edit Users

User with Full-Pod-Access or Manage Users permission can edit other user's permission for that specific Pod.

To edit user's permission

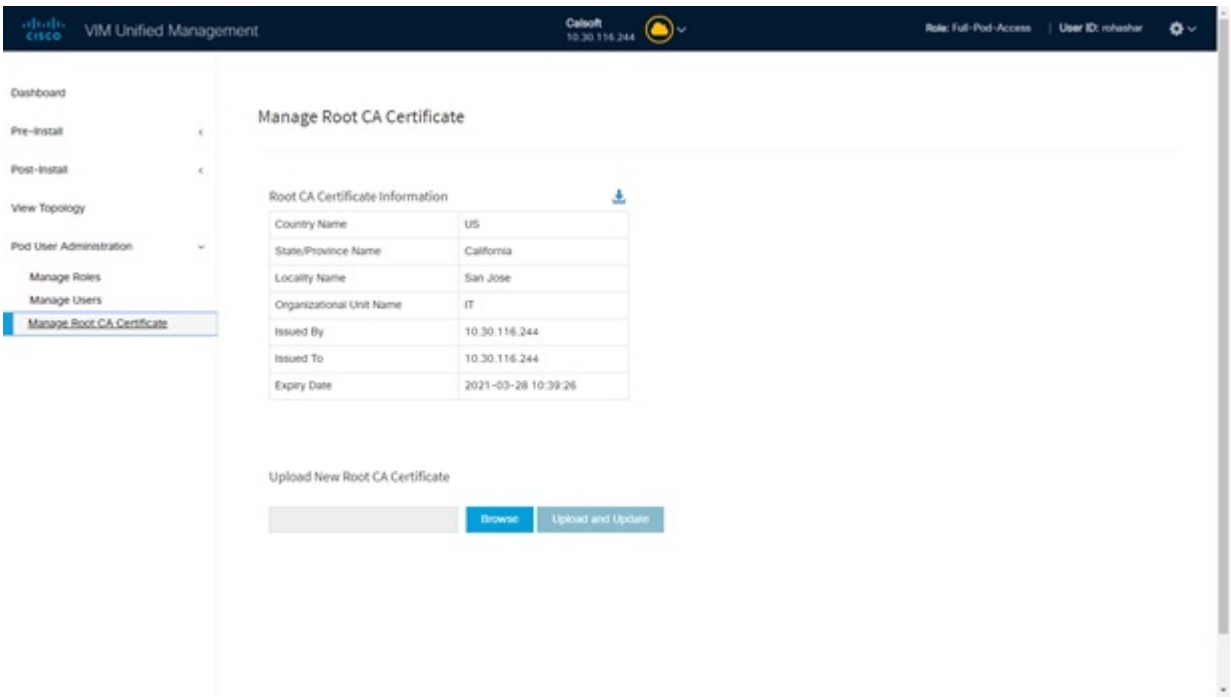
Step 1 Click **Edit** icon.

Step 2 Update the permission.

Step 3 Click **Save**. The Grid will get refreshed automatically.

Managing Root CA Certificate

You can update the CA Certificate during the registration of the POD. Once, logged in as POD User and if you have the permission to update the certificate you can view under POD User Administration>> Manage Root CA Certificate.



To update the Certificate:

Step 1 Click **Login as POD User**

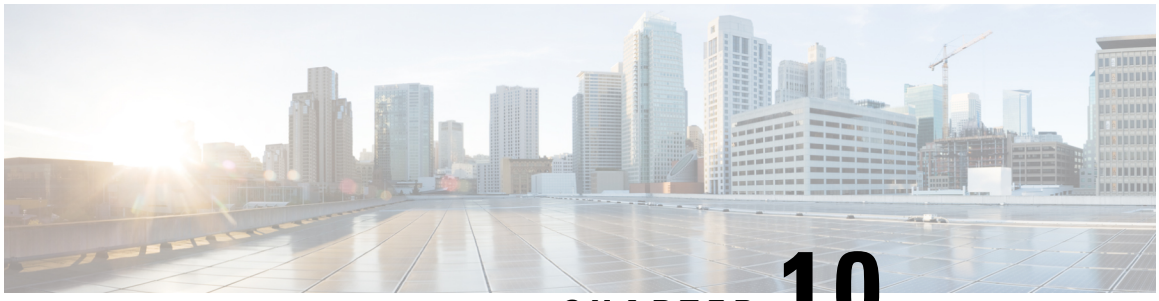
Step 2 Navigate to **POD User Administration>>Manage Root CA certificate**.

Step 3 Click **Browse** and select the certificate that you want to upload.

Step 4 Click **Upload.**

- If the certificate is Invalid, and does not matches with the certificate on the management node located at (var/www/mercury/mercury-ca.crt) then Insight reverts the certificate which was working previously.
- If the Certificate is valid, Insight runs a management node health check and then update the certificate with the latest one.

Note The CA Certificate which is uploaded should be same as the one which is in the management node.



CHAPTER 10

Shutting Down and Restarting Cisco VIM Insight

The following topic guides you how to shutdown and restart the Cisco VIM Insight.

Shutting Down Cisco VIM Insight

To stop the Cisco VIM Insight Container services, shut down Cisco UCS VIM Insight by running the **systemctl stop `service`** command.

Step 1 Log in to a server in which the Insight container is running.

Step 2 Stop the Insight service by running the following command from the Shell window:

```
systemctl stop docker-insight
```

a) Check the status of Insight Container by running the following command: **docker ps -a | grep** insight.

```
STATUS
Up 6 seconds
```

b) Check the status of the service by running the following command:

```
systemctl staus docker-insight
```

The following information is displayed

```
Docker-insight.service - Insight Docker Service
Loaded: loaded (/usr/lib/systemd/system/docker-insight.service; enabled; vendor preset: disabled)
Active: inactive (dead) since <Date and Time since it was last active>
```

Restarting Cisco VIM Insight

Step 1 Log in to the server in which the Insight container was stopped.

Step 2 Restart the Insight service by running the following command from the shell window:

```
systemctl restart docker-insight
```

a) Check the status of Insight container by running the following command: **docker ps -a | grep** insight.

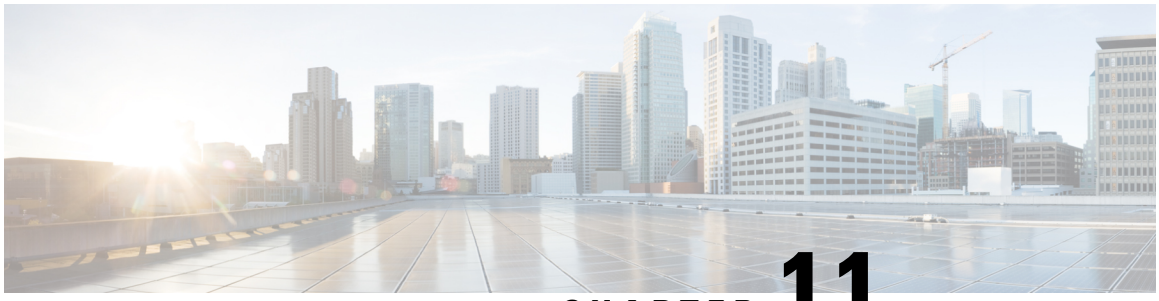
```
STATUS
Up 6 seconds
```

b) Check the status of the service by running the following command:

```
systemctl staus docker-insight
```

The following output is displayed:

```
Docker-insight.service - Insight Docker Service
Loaded: loaded (/usr/lib/systemd/system/docker-insight.service; enabled; vendor preset: disabled)
Active: active (running) since <Date and Time when it got active.>
```



CHAPTER 11

Overview to the Cisco Virtual Topology System

The Cisco Virtual Topology System (VTS) is an optional Cisco NFVI application that uses the Neutron driver and supports Cisco Vector Packet Processing. The following topics provide an overview to VTS architecture and features. When using VTS with Cisco NFVI, keep the following OpenStack tenant restrictions in mind:

Restriction	Description
Nova flavors: VM RAM > 512MB and equal to a multiple of 512MB	This limitation is due to NUMA and hugepages details.
Nova Flavors: nova flavor-key m1.medium set hw:numa_nodes=1 nova flavor-key m1.medium set hw:mem_page_size=large	VHOST mode is the only mode supported by the VTS installation at this time. To support VHOST connections nova needs the following configurations on each flavor that will be used.
OpenStack doesn't provide VM tenant IP allocation	VTS installed with Cisco NFVI does not provide support for OpenStack DHCP servers.

Understanding Cisco VTS

The Cisco Virtual Topology System (VTS) is a standards-based, open, overlay management and provisioning system for data center networks. It automates DC overlay fabric provisioning for both physical and virtual workloads.

Cisco VTS provides a network virtualization architecture and software-defined networking (SDN) framework that meets the requirements of multitenant data centers for cloud services. It enables a policy-based approach for overlay provisioning.

Cisco VTS automates complex network overlay provisioning and management tasks through integration with cloud orchestration systems such as OpenStack and VMware vCenter and abstracts out the complexity involved in managing heterogeneous network environments. The solution can be managed from the embedded Cisco VTS GUI or entirely by a set of northbound Representational State Transfer (REST) APIs that can be consumed by orchestration and cloud management systems.

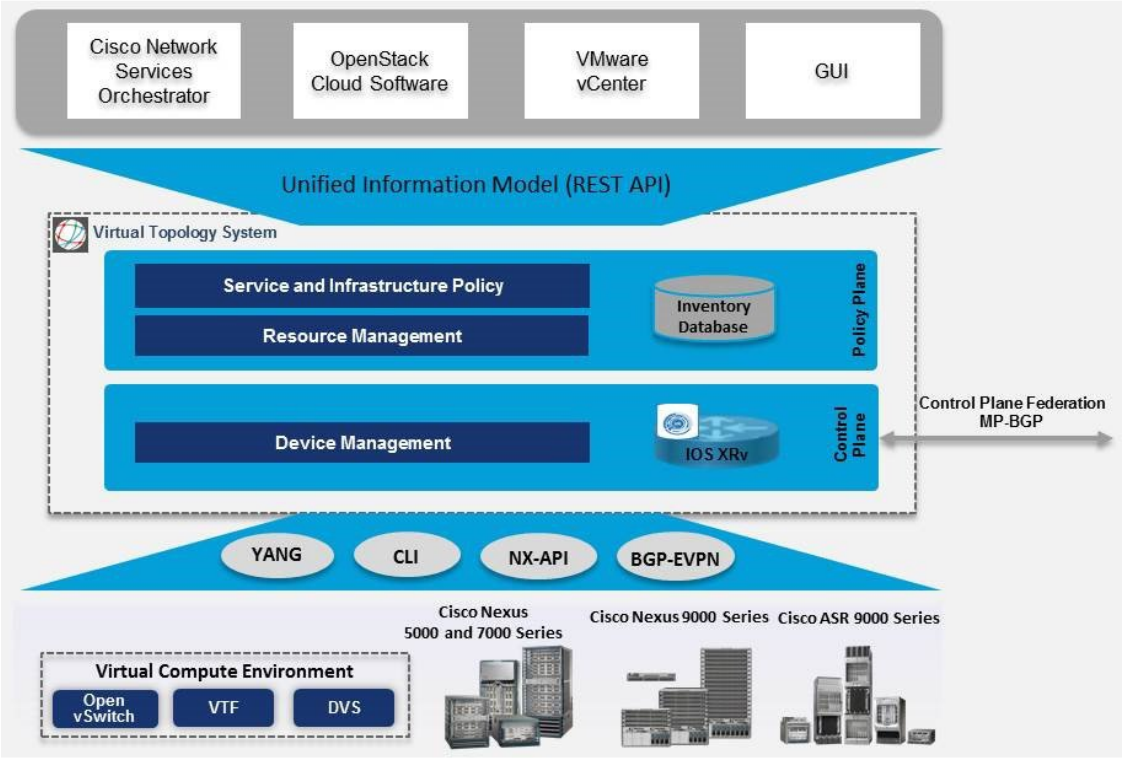
Cisco VTS provides:

- Fabric automation
- Programmability
- Open, scalable, standards based solution
- Cisco Nexus 2000, 3000, 5000, 7000, and 9000 Series Switches. For more information, see Supported Platforms in *Cisco VTS 2.5 Installation Guide*.
- Software forwarder (Virtual Topology Forwarder [VTF])

Cisco VTS Architecture Overview

Cisco VTS architecture has two main components: the Policy Plane and the Control Plane. These perform core functions such as SDN control, resource allocation, and core management function.

Figure 14: Cisco VTS Architecture



- **Policy Plane**: The policy plane enables Cisco VTS to implement a declarative policy model designed to capture user intent and render it into specific device-level constructs. The solution exposes a set of modular policy constructs that can be flexibly organized into user-defined services for use cases across service provider and cloud environments. These policy constructs are exposed through a set of REST APIs that can be consumed by orchestrators and applications to express user intent, or instantiated through the Cisco VTS GUI. Policy models are exposed as system policies or service policies.

System policies allow administrators to logically group devices into pods within or across data centers to define Admin Domains with common system parameters (for example, BGP-EVPN control plane with distributed Layer 2 and 3 gateways).

The inventory module maintains a database of the available physical entities (for example, data center interconnect [DCI] routers and top-of-rack leaf, spine, and border-leaf switches) and virtual entities (for example, VTFs) in the Virtual Topology System domain. The database also includes interconnections between these entities and details about all services instantiated within a Virtual Topology System domain.

The resource management module manages all available resource pools in the Virtual Topology System domain, including VLANs, VXLAN Network Identifiers (VNIs), IP addresses, and multicast groups.

- **Control Plane**: The control plane module serves as the SDN control subsystem that programs the various data planes including the VTFs residing on the x86 servers, hardware leafs, DCI gateways. The Control plane hosts Service Routing (SR) module, which provides routing services to Cisco VTS. The Service Routing (SR) module is responsible for calculating L2 and L3tables and routes to provide connectivity between the different VMs for a given tenant and service chaining. The main components of this module are the VTSR and VTF. VTSR is the controller and Virtual topology forwarder (VTF) runs on each compute server hosting the tenant VMs.

Virtual Topology Forwarder

Virtual Topology Forwarder (VTF) runs on each compute server in the DC and provides connectivity to all tenant VMs hosted on the compute server. VTF supports both intra and inter DC/WAN connectivity. VTF allows Cisco VTS to terminate VXLAN tunnels on host servers by using the VTF as a Software VXLAN Tunnel Endpoint (VTEP). Cisco VTS also supports hybrid overlays by stitching together physical and virtual endpoints into a single VXLAN segment.

VTF has 2 major components—Cisco's VPP (Vector Packet Processing)and VPFa. VPFA is a Cisco agent running on each VMM compute resource. VPFA is FIB agent which receives L2/L3 table forwarding information from VTSR needed to provide the connectivity to local tenant VMs hosted on its compute, and programs them in the VPP.

VTF is deployed as a virtual machine or in vhost mode, to deliver a high-performance software data plane on a host server.

Overview to Cisco VTF and VPP

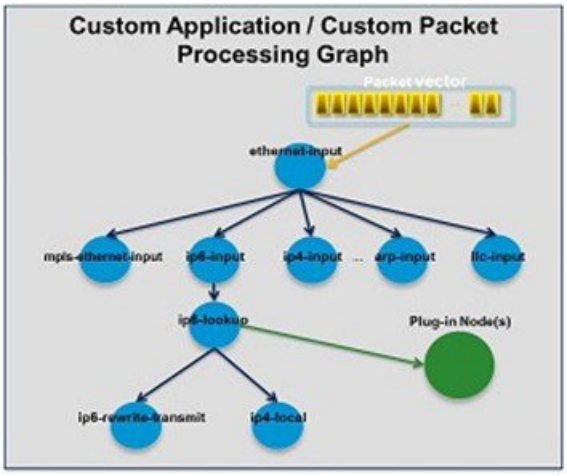
Cisco VTF is a Cisco Soft switch built on the Cisco Vector Packet Processing (VPP) technology.

The VPP platform is an extensible framework that provides extremely productive and quality switch/router functionality. It is the open source version of the Cisco VPP technology, which is a high performance, packet-processing stack that can run on commodity CPUs.

The benefits of VPP are its high performance, proven technology, modularity, flexibility, and rich feature set.

The VPP platform is built on a packet-processing graph. This modular approach allows anyone to plugin new graph nodes. This makes extensibility rather simple, and the plugins can be customized for specific purposes.

Figure 15: VPP Platform



The VPP platform grabs all available packets from RX rings to form a vector of packets. A packet-processing graph is applied, node by node (including plugins) to the entire packet vector. Graph nodes are small and modular, and loosely coupled which makes it easy to include new graph nodes and rewire existing graph nodes.

A plugin can introduce new graph nodes or rearrange the packet-processing graph. You can also build a plugin independent from the VPP source and consider it as an independent component. A plugin can be installed by adding it to a plugin directory.

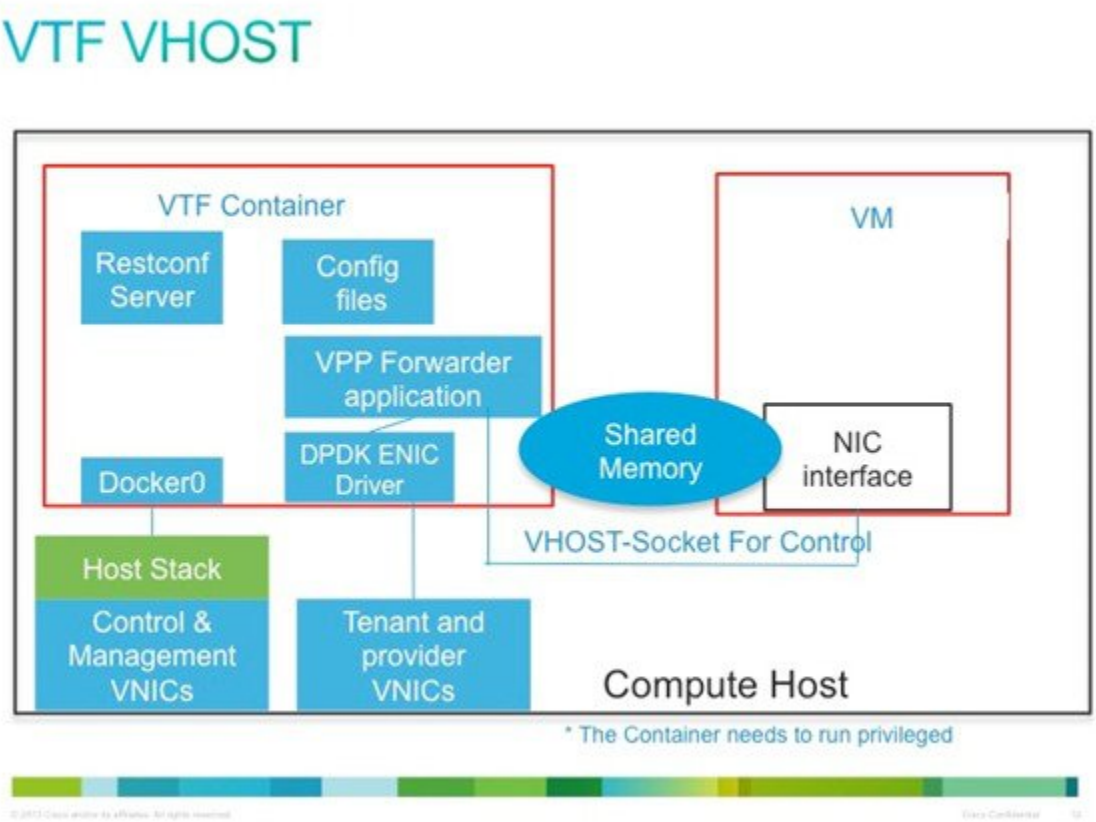
VTF uses remote plugin that binds into VPP using VPFA (VPF agent). The VPFA interacts with VPP application using low level API. The VPFA exposes netconf or yang based API for remote devices to program the VTF through the VPFA.

VPP + VHOSTUSER

vhost is a solution that allows the user space process to share a number of virtqueues directly with a Kernel driver. The transport mechanism in this case is the ability of the kernel side to access the user space application memory, and a number of ioeventfds and irqfds to serve as the kick mechanism. A QEMU guest uses an emulated PCI device, as the control plane is still handled by QEMU. However once a virtqueue has been set up, the QEMU guest will use the vhost API to pass direct control of a virtqueue to a Kernel driver.

In this model, a vhost_net driver directly passes the guest network traffic to a TUN device directly from the Kernel side, improving performance significantly.

Figure 16: VTF vhost



In the above implementation, the guest NFV application directly writes packets into the TX rings, which is shared through a common vhost socket as the RX ring on the VPP . The VPP grabs these packets from the RX ring buffer and forwards the packets using the vector graphs it maintains.

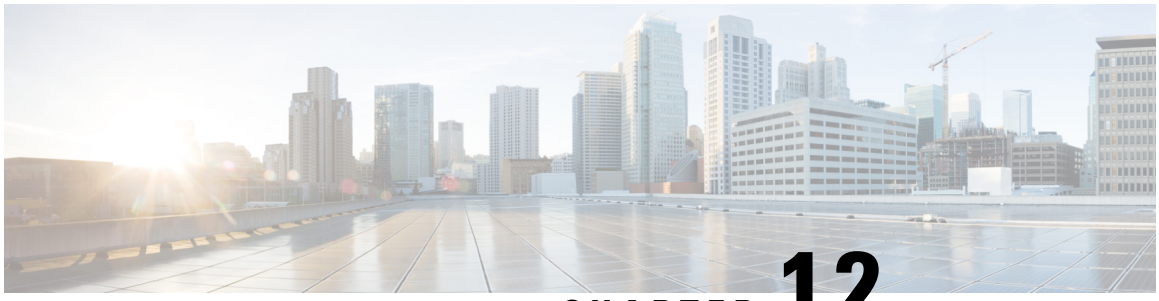
Virtual Topology System High Availability

The Virtual Topology System solution is designed to support redundancy, with two solution instances running on separate hosts in an active-standby configuration.

During initial setup, each instance is configured with both an underlay IP address and a virtual IP address. Virtual Router Redundancy Protocol (VRRP) is used between the instances to determine which instance is active.

The active-instance data is synchronized with the standby instance after each transaction to help ensure consistency of the control-plane information to accelerate failover after a failure. BGP peering is established from both Virtual Topology System instances for the distribution of tenant-specific routes. During the switchover, nonstop forwarding (NSF) and graceful restart help ensure that services are not disrupted.

See the *Installing VTS in High Availability Mode* section of the *Cisco VTS 2.4.2 Installation Guide* for the detailed procedure about setting up high availability.



CHAPTER 12

Managing Backup and Restore Operations

The following topics describe Cisco NFVI management node backup and restore operations.

Managing Backup and Restore Operations

The management node hosts critical services such as Cisco VIM REST API, Cobbler for PXE, ELK for Logging/Kibana dashboard, and VMTP for cloud validation in Cisco VIM.

The management node is not redundant during the initial Cisco VIM offering, hence it is recommended to take backup of the management node. Using the saved management node information, you can restore the management node if you are facing any issues with the platform.

Backing up the Management Node

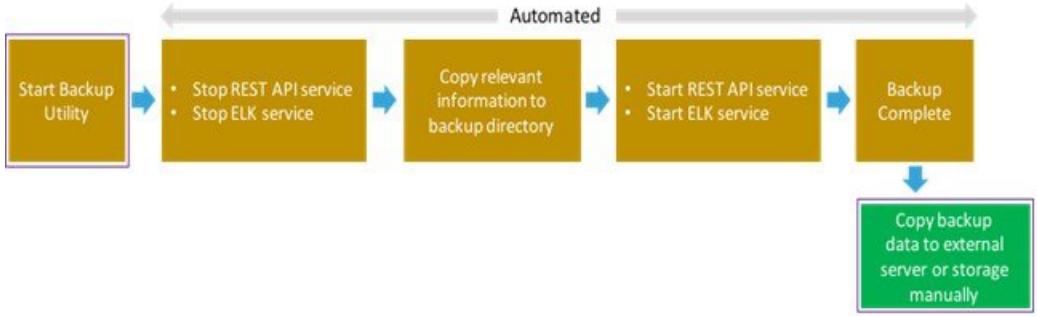
An administrator must maintain the number of back up snapshots on the management node. The backup of the management node is possible only after complete deployment of at least one Cisco VIM. Two copies of backup directories are maintained at the management node itself and the older copy will be overwritten when a next backup is performed.

During the backup operation, activities such as pod management, software update or upgrade, and addition or deletion or replacement of nodes cannot be performed.

The REST API and ELK services are stopped during the backup operation, the OpenStack logs are cached on the control, compute, and storage nodes till the restoration of the management node is completed.

As part of the backup operation, two files are created: .backup_files and .backup_hash. .backup_files is a list of files that are backed up, while the second one is the hash. These two files are placed under the backup directory /var/cisco/backup_<tag>_<date-time> at the management node and also at the /var/cisco/ directory of all three controllers. These two files are used during the restore validation. When user attempt to restore from a particular backup, these two files within this backup are compared to those at the controllers. If there is any discrepancy, the restore validation will fail and user will be prompted to either terminate the restore operation or continue despite the validation failure. Only one copy of the .backup_files and .backup_hash are kept at the controllers, that is every time a new backup is created, these two files are overwritten with the most recent ones. Hence the restore validation will only pass when the latest backup is used for restore.

Figure 17: Cisco NFVI Management Node Backup Operation



Before you begin

- Save the management node information (for example, IP address of the management node) for use during the restore operation.
- Ensure that you have the br_mgmt and br_api IP addresses and respective network information.

Step 1 Launch a SSH session to the Cisco NFVI management node.

Step 2 Navigate to the <installer-ws>/tools/mgmt/ directory.

Step 3 Execute **mgmt_node_backup.py**.

What to do next

The backup operation takes approximately 30 minutes and creates the backup_<tag>_<date-time> directory in the /var/cisco/ path.

Copy the directory to a remote server to recover the management node using rsync.

For example, to copy the backup directory to the remote server 20.0.0.5 /var/cisco/directory , execute the following command sequence:

```
rsync -e ssh -rtvpX --numeric-ids /var/cisco/backup_2017-01-09_14-04-38
root@20.0.0.5:/var/cisco/
```

Note On the remote server, protect the backup directory for any unauthorized access as the backup files may contain sensitive information

At the remote server, change directory to where the backup directory is copied to; in this example /var/cisco/backup_<tag>_<date-time>/.

To verify if the backup is not corrupted or modified, execute **./check_integrity**.

Check_integrity depends on the following packages, they should be installed on the server where check_integrity is executed.

```
python-prettytable
python-jinja2
```

```
python-babel
python-markupsafe
python-setuptools
pytz
```

Backup with Forwarding ELK logs to External Syslog Server

When the feature Forwarding ELK logs to External Syslog Server is enabled, during the backup process, in both the auto-backup and manual backup, the ELK logs are not collected. For manual backups, user can override by appending the -a or --add-elk option to the backup command. The -s or --skip-elk option is to skip the ELK logs collection regardless of the forwarding feature is enabled or not.

```
# cd installer/tools/mgmt
# ./mgmt_node_backup.py --help
Usage:  ./mgmt_node_backup.py [options]
Options:
    -h, --help      show this help message and exit
    -s, --skip-elk  do not collect ELK logs during backup
    -a, --add-elk   force to also collect ELK logs on backup
```

Restoring the Management Node

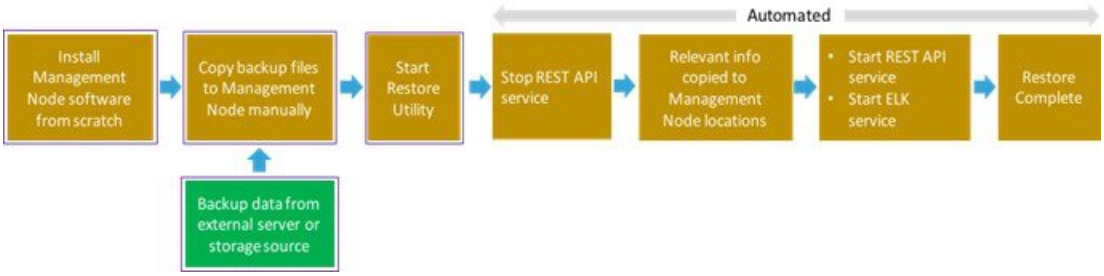
As an administrator, you have to re-image the management node with the same ISO version when the backup is performed, before initiating the restore operation. Restore will fail when there is a version mismatch.

Note Version checking is available only for offline installation.

As part of the restore operation, system checks for management node's IP address information to match the prior configuration. Logs are cached on the control, compute and storage nodes from the moment of the management node fails until its restoration.

If you are using Cisco VIM Insight (in Tech Preview), in the same management node, you will have to re-bootstrap it for installation. During installation, RBAC and Pod registration information will be lost, hence it is advised to make a note of the RBAC and Pod information.

Figure 18: Cisco NFVI Management Node Restore Operation



Before you begin

Ensure that you have the br_mgmt and br_api IP addresses of the failed management node.

Step 1 Re-image the management node with the ISO version with which you want to restore the node, and with the same IP address that is used before the failure of the node.

Step 2 Navigate to /var/cisco/directory at the remote server where the backup directory is copied during the backup operation. Execute **./check_integrity** to verify if the backup is not corrupted or modified.

Step 3 Copy the backup file to the /var/cisco/directory of the re-imaged management node.

For example, to copy the backup directory from the remote host 20.0.0.5 to the management node /var/cisco/directory, execute the following command sequence:

```
rsync -e ssh -rtvpX --numeric-ids root@20.0.0.5:/var/cisco/backup_2017-01-09_14-04-38 /var/cisco/
```

Step 4 Navigate to the backup directory and execute the following command to verify if the backup is not corrupted or modified.

```
# cd /var/cisco/backup_<date-time>
# ./check-integrity
```

Step 5 In /var/cisco/backup_<date-time> directory, execute the following command:

```
/var/cisco/backup_<date-time> # ./restore
```

The restore operation takes around 45 minutes.

Step 6 Before restoration, the restore script performs validation of the backup directory. If validation fails, restore operation will be halted and an error message will be displayed. The script will also verify the last performed backup directory in the Management Node, and if any defects are detected, the user needs to confirm to proceed with restore operation.

```
 …
2017-02-02 21:25:23 INFO Starting Cisco VIM restore...
 2017-02-02 21:25:23 INFO Cisco VIM restore: estimated run time is approx. 45 mins...
 2017-02-02 21:25:23 INFO Please see progress log for restore at
/var/log/mercury/installer/restore_2017-02-02_21:25:23.log
 2017-02-02 21:25:27 ERROR Error: Backup id is not the one expected
 Error: Found hashID file only in controller(s): j10-controller-2, j10-controller-3
 Management backup files are ok (as per j10-controller-2)
 Management backup files are ok (as per j10-controller-3)
 The management node changed after the last backup was stored. Do you still want to proceed restoring
 this management node? [Y/n] y
 2017-02-02 22:17:55 INFO Workspace restored to /root/installer-6518
 2017-02-02 22:17:55 INFO Cisco VIM restore: Executing restore playbook ...
 2017-02-02 22:18:47 INFO Cisco VIM restore: Executing bootstrap playbook ...
```

Note The default behavior is to continue by keying **Return** or **Y**. Keying **N** will abort the restore operation.

```
 …
2017-02-02 21:25:23 INFO Starting Cisco VIM restore...
2017-02-02 21:25:23 INFO Cisco VIM restore: estimated run time is approx. 45 mins...
2017-02-02 21:25:23 INFO Please see progress log for restore at
/var/log/mercury/installer/restore_2017-02-02_21:25:23.log
2017-02-02 21:25:27 ERROR Error: Backup id is not the one expected
Error: Found hashID file only in controller(s): j10-controller-2, j10-controller-3
Management backup files are ok (as per j10-controller-2)
Management backup files are ok (as per j10-controller-3)
The management node changed after the last backup was stored. Do you still want to proceed restoring
 this management node? [Y/n] n
 Aborting the restore operation as per user request
```

Once restore operation ends, several health check points will be automatically executed and the summary of results for that particular cloud reachability will be display.

Step 7 User can run the following checks manually to verify the status of the restore:

- Check the status of the REST API server:

```
# cd installer-<tagid>/tools
#./restapi.py -a status
Status of the REST API Server: active (running) since Thu 2016-08-18 09:15:39 UTC; 9h ago
REST API launch directory: /root/installer-<tagid>/
```

- Check the setup_data and runtime consistency of the management node:

```
# cd installer-<tagid>/; ./ciscovimclient/ciscovim run --perform 1,3 -y
```

- Execute the cloud sanity command:

```
#   cd installer-<tagid>/tools
# ./cloud_sanity.py -c all
```

Management Node Auto-backup

After the successful completion of certain Pod management operations, a backup of the management node is performed automatically. Only one copy of the auto-backup directory is kept at /var/cisco/ at any given time. The directory format is autobackup_<tag>_<timestamp>

Below is a list of operations:

- Fresh install of Cisco VIM
- Commit an update
- Replace controller
- Add or Remove compute nodes
- Add or Remove storage node
- Reconfigure
- NFVIMON

Enabling or disabling the variable auto-backup, is defined in the setup_data.yaml file. It is enabled by default.

Add the following setup-data.yaml file snippet:

```
######################################################
# AutoBackup configuration
######################################################
#Default is True
#autobackup: True or False
```

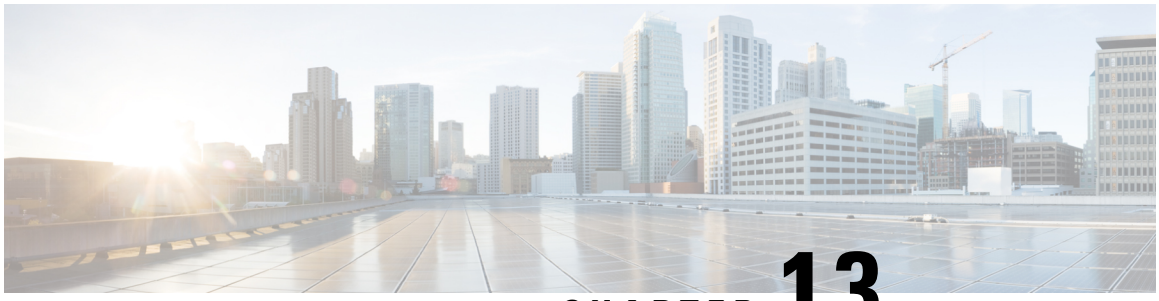
The following tables shows when an auto-backup is performed during update or rollback or commit.

POD operation	Auto-backup performed
Update	No
Rollback	No
Commit	Yes

Update fail with auto rollback	No

After successful auto-backup directory creation, user can copy it to an external server for later restoration as mentioned in Restoring the Management Node.

During the auto backup, if **Forwarding ELK logs to External Syslog server** option is enabled, the ElasticSearch database will not be maintained and the ELK logs will not be recovered after restoring the management node.



CHAPTER 13

Troubleshooting

Displaying Cisco NFVI Node Names and IP Addresses

Complete the following steps to display the Cisco NFVI node names and IP addresses.

Step 1 Log into the Cisco NFVI build node.

Step 2 The openstack-configs/mercury_servers_info file displays the node name and the address as follows.

```
# more openstack-configs/mercury_servers_info Total nodes: 5
Controller nodes: 3
+------------------+--------------+-------------+-------------+-----------------+---------+
| Server | CIMC | Management | Provision | Tenant | Storage |
+------------------+--------------+-------------+-------------+-----------------+---------+
| test-c-control-1 | 10.10.223.13 | 10.11.223.22 | 10.11.223.22 | 169.254.133.102 | None |
| | | | | | |
| test-c-control-3 | 10.10.223.9  | 10.11.223.23 | 10.11.223.23 | 169.254.133.103 | None |
```

```
| | | | | | |
| test-c-control-2 | 10.10.223.10 | 10.11.223.24 | 10.11.223.24 | 169.254.133.104 | None |
| | | | | | |
+------------------+--------------+--------------+--------------+-----------------+---------+
Compute nodes: 2
+------------------+--------------+--------------+--------------+-----------------+---------+
| Server | CIMC |  Management | Provision | Tenant | Storage |
+------------------+--------------+--------------+--------------+-----------------+---------+
| test-c-compute-1 | 10.10.223.11 | 10.11.223.25 | 10.11.223.25 | 169.254.133.105 | None |
| | | | | | |
| test-c-compute-2 | 10.10.223.12 | 10.11.223.26 | 10.11.223.26 | 169.254.133.106 | None |
| | | | | | |
+
```

Note During the Cisco NFVI deployment, SSH public keys for each node are added to .../.ssh/authorized_keys, so you should be able to log in from the build node into each of the Cisco NFVI nodes without passwords. If, for some reason you do need account information, see the openstack-configs/secrets.yaml file on the build node.

Verifying Cisco NFVI Node Interface Configurations

Complete the following steps to verify the interface configurations of Cisco NFVI nodes:

Step 1 SSH into the target node, for example, one of the Cisco VIM controllers:

```
[root@ j11-build-1~]# ssh root@j11-control-server-1
[root@j11-control-server-1 ~]#
```

Step 2 Enter the **ip a** command to get a list of all interfaces on the node:

```
[root@j11-control-server-1 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: enp8s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP qlen 1000
    link/ether 54:a2:74:7d:42:1d brd ff:ff:ff:ff:ff:ff
3: enp9s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP qlen 1000
    link/ether 54:a2:74:7d:42:1e brd ff:ff:ff:ff:ff:ff
4: mx0: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc mq master mx state UP qlen 1000
    link/ether 54:a2:74:7d:42:21 brd ff:ff:ff:ff:ff:ff
5: mx1: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc mq master mx state UP qlen 1000
    link/ether 54:a2:74:7d:42:21 brd ff:ff:ff:ff:ff:ff
6: t0: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc mq master t state UP qlen 1000
    link/ether 54:a2:74:7d:42:23 brd ff:ff:ff:ff:ff:ff
7: t1: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc mq master t state UP qlen 1000
    link/ether 54:a2:74:7d:42:23 brd ff:ff:ff:ff:ff:ff
8: e0: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc mq master e state UP qlen 1000
    link/ether 54:a2:74:7d:42:25 brd ff:ff:ff:ff:ff:ff
9: e1: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc mq master e state UP qlen 1000
    link/ether 54:a2:74:7d:42:25 brd ff:ff:ff:ff:ff:ff
10: p0: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc mq master p state UP qlen 1000
    link/ether 54:a2:74:7d:42:27 brd ff:ff:ff:ff:ff:ff
11: p1: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc mq master p state UP qlen 1000
    link/ether 54:a2:74:7d:42:27 brd ff:ff:ff:ff:ff:ff
12: a0: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc mq master a state UP qlen 1000
    link/ether 54:a2:74:7d:42:29 brd ff:ff:ff:ff:ff:ff
```

```
13: a1: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc mq master a state UP qlen 1000
    link/ether 54:a2:74:7d:42:29 brd ff:ff:ff:ff:ff:ff
14: bond0: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN
    link/ether 4a:2e:2a:9e:01:d1 brd ff:ff:ff:ff:ff:ff
15: a: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue master br_api state UP
    link/ether 54:a2:74:7d:42:29 brd ff:ff:ff:ff:ff:ff
16: br_api: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    link/ether 54:a2:74:7d:42:29 brd ff:ff:ff:ff:ff:ff
17: e: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    link/ether 54:a2:74:7d:42:25 brd ff:ff:ff:ff:ff:ff
18: mx: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue master br_mgmt state UP
    link/ether 54:a2:74:7d:42:21 brd ff:ff:ff:ff:ff:ff
19: br_mgmt: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    link/ether 54:a2:74:7d:42:21 brd ff:ff:ff:ff:ff:ff
    inet 10.23.221.41/28 brd 10.23.221.47 scope global br_mgmt
       valid_lft forever preferred_lft forever
20: p: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    link/ether 54:a2:74:7d:42:27 brd ff:ff:ff:ff:ff:ff
21: t: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    link/ether 54:a2:74:7d:42:23 brd ff:ff:ff:ff:ff:ff
    inet 17.16.3.8/24 brd 17.16.3.255 scope global t
       valid_lft forever preferred_lft forever
22: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
    link/ether 02:42:70:f6:8b:da brd ff:ff:ff:ff:ff:ff
    inet 172.17.42.1/16 scope global docker0
       valid_lft forever preferred_lft forever
24: mgmt-out@if23: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master br_mgmt state
 UP qlen 1000
    link/ether 5a:73:51:af:e5:e7 brd ff:ff:ff:ff:ff:ff link-netnsid 0
26: api-out@if25: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master br_api state UP
 qlen 1000
    link/ether 6a:a6:fd:70:01:f9 brd ff:ff:ff:ff:ff:ff link-netnsid 0
```

Displaying Cisco NFVI Node Network Configuration Files

Complete the following steps to view a Cisco NFVI node network configuration files:

Step 1 SSH into the target node, for example, one of the Cisco VIM controllers:

```
[root@ j11-build-1~]# ssh root@j11-control-server-1
[root@j11-control-server-1 ~]#
```

Step 2 List all of the network configuration files in the /etc/sysconfig/network-scripts directory, for example:

```
[root@j11-control-server-1 ~]# ls /etc/sysconfig/network-scripts/
ifcfg-a         ifcfg-enp15s0  ifcfg-mx0    ifdown-ib        ifup           ifup-ppp
ifcfg-a0        ifcfg-enp16s0  ifcfg-mx1    ifdown-ippp      ifup-aliases   ifup-routes
ifcfg-a1        ifcfg-enp17s0  ifcfg-p      ifdown-ipv6      ifup-bnep      ifup-sit
ifcfg-br_api    ifcfg-enp18s0  ifcfg-p0     ifdown-isdn      ifup-eth       ifup-Team
ifcfg-br_mgmt   ifcfg-enp19s0  ifcfg-p1     ifdown-post      ifup-ib        ifup-TeamPort
ifcfg-e         ifcfg-enp20s0  ifcfg-t      ifdown-ppp       ifup-ippp      ifup-tunnel
ifcfg-e0        ifcfg-enp21s0  ifcfg-t0     ifdown-routes    ifup-ipv6      ifup-wireless
ifcfg-e1        ifcfg-enp8s0   ifcfg-t1     ifdown-sit       ifup-isdn      init.ipv6-global
ifcfg-enp12s0   ifcfg-enp9s0   ifdown       ifdown-Team      ifup-plip      network-functions
```

```
ifcfg-enp13s0  ifcfg-lo       ifdown-bnep  ifdown-TeamPort  ifup-plusb    network-functions-ipv6
ifcfg-enp14s0  ifcfg-mx       ifdown-eth   ifdown-tunnel    ifup-post
```

Viewing Cisco NFVI Node Interface Bond Configuration Files

Complete the following steps to view the Cisco NFVI node interface bond configuration files:

Step 1 SSH into the target node, for example, one of the Cisco VIM controllers:

```
[root@ j11-build-1~]# ssh root@j11-control-server-1
[root@j11-control-server-1 ~]#
```

Step 2 List all of the network bond configuration files in the /proc/net/bonding/ directory:

```
[root@j11-control-server-1 ~]# ls /proc/net/bonding/
a  bond0  e  mx  p  t
```

Step 3 To view more information about a particular bond configuration, enter:

```
[root@j11-control-server-1 ~]# more /proc/net/bonding/a
Ethernet Channel Bonding Driver: v3.7.1 (April 27, 2011)

Bonding Mode: load balancing (xor)
Transmit Hash Policy: layer3+4 (1)
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 0
Down Delay (ms): 0

Slave Interface: a0
MII Status: up
Speed: 10000 Mbps
Duplex: full
Link Failure Count: 1
Permanent HW addr: 54:a2:74:7d:42:29
Slave queue ID: 0

Slave Interface: a1
MII Status: up
Speed: 10000 Mbps
Duplex: full
Link Failure Count: 2
Permanent HW addr: 54:a2:74:7d:42:2a
Slave queue ID: 0
```

Viewing Cisco NFVI Node Route Information

Complete the following steps to view Cisco NFVI node route information. Note that this is not the HAProxy container running on the controller. The default gateway should point to the gateway on the management network using the br_mgmt bridge.

Step 1 SSH into the target node, for example, one of the Cisco VIM controllers:

```
[root@ j11-build-1~]# ssh root@j11-control-server-1
[root@j11-control-server-1 ~]#
```

Step 2 View the routing table (verify the default gateway) of the Cisco NFVI node:

```
[root@j11-control-server-1 ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.23.221.33    0.0.0.0         UG    0      0        0 br_mgmt
10.23.221.32    0.0.0.0         255.255.255.240 U     0      0        0 br_mgmt
17.16.3.0       0.0.0.0         255.255.255.0   U     0      0        0 t
169.254.0.0     0.0.0.0         255.255.0.0     U     1016   0        0 br_api
169.254.0.0     0.0.0.0         255.255.0.0     U     1017   0        0 e
169.254.0.0     0.0.0.0         255.255.0.0     U     1019   0        0 br_mgmt
169.254.0.0     0.0.0.0         255.255.0.0     U     1020   0        0 p
169.254.0.0     0.0.0.0         255.255.0.0     U     1021   0        0 t
172.17.0.0      0.0.0.0         255.255.0.0     U     0      0        0 docker0
```

Viewing Linux Network Namespace Route Information

Complete the following steps to view the route information of the Linux network namespace that the HAProxy container uses on a Cisco NFVI controller node. The default gateway should point to the gateway on the API network using the API interface in the Linux network namespace.

Step 1 SSH into the target node, for example, one of the Cisco VIM controllers:

```
[root@ j11-build-1~]# ssh root@j11-control-server-1
[root@j11-control-server-1 ~]#
```

Step 2 Enter the **ip netns** command to find the name of the network namespace:

```
[root@j11-control-server-2 ~]# ip netns 17550 (id: 0)
```

Step 3 Enter the **ip netns exec** command to view the routing table (verify the default gateway) of the Linux network namespace:

```
[root@j11-control-server-2 ~]# ip netns exec 17550 route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.29.86.1     0.0.0.0         UG    0      0        0 api
10.23.221.32    0.0.0.0         255.255.255.240 U     0      0        0 mgmt
172.29.86.0     0.0.0.0         255.255.255.0   U     0      0        0 api
```

Prior to Remove Storage Operation

Upon completion of the pod management operations such as add-storage, the operator needs to ensure that any subsequent operation such as remove-storage on the same storage node is done after accounting for all

of the devices and their corresponding OSDs have been marked in the persistent crush map as shown in the output of the ceph osd crush tree.

Execute the following command on the storage node where a remove-storage pod operation is performed, to get a list of all the devices configured for ceph osds:

```
[root@storage-3 ~]$ df | grep -oh ceph-[0-9]*
[root@storage-3 ~]$ df | grep -oh ceph-[0-9]*
ceph-1
ceph-5
ceph-7
ceph-10
```

Login to any of the controller nodes and run the following commands within the ceph mon container:

```
$ cephmon
$ ceph osd crush tree
```

From the json output, locate the storage node to be removed and ensure all of the devices listed for ceph osds have corresponding osd entries for them by running the following commands:

```
{
                "id": -3,
                "name": "storage-3",
                "type": "host",
                "type_id": 1,
                "items": [
                    {
                        "id": 1,
                        "name": "osd.1",
                        "type": "osd",
                        "type_id": 0,
                        "crush_weight": 1.091095,
                        "depth": 2
                    },
                    {
                        "id": 5,
                        "name": "osd.5",
                        "type": "osd",
                        "type_id": 0,
                        "crush_weight": 1.091095,
                        "depth": 2
                    },
                    {
                        "id": 7,
                        "name": "osd.7",
                        "type": "osd",
                        "type_id": 0,
                        "crush_weight": 1.091095,
                        "depth": 2
                    },
                    {
                        "id": 10,
                        "name": "osd.10",
                        "type": "osd",
                        "type_id": 0,
                        "crush_weight": 1.091095,
                        "depth": 2
                    }
                ]
            },
```

Troubleshooting Cisco NFVI

The following topics provide Cisco NFVI general troubleshooting procedures.

Container Download Problems

1. Check installer logs log file /var/log/mercury/mercury_buildorchestration.log for any build node orchestration failures including stuck "registry-Populate local registry". In some cases, the Docker container download from your management node might be slow.
2. Check the network connectivity between the management node and the remote registry in defaults.yaml on the management node (grep "^registry:" openstack-configs/defaults.yaml).
3. Verify valid remote registry credentials are defined in setup_data.yaml file.
4. A proxy server might be needed to pull the container images from remote registry. If a proxy is required, exclude all IP addresses for your setup including management node.

PXE Boot Problems

1. Check log file /var/log/mercury/mercury_baremetal_install.log and connect to failing node CIMC KVM console to find out more on PXE boot failure reason.
2. Ensure all validations (step 1) and hardware validations (step 3) pass.
3. Check the kickstart file used in setup_data.yaml for controller, compute, and storage nodes and that matches with the hardware of corresponding nodes.
4. Check the Cobbler web interface to see all the configured systems got populated:

   ```
   https://<management_node_ip>/cobbler_web/system/list username:
   [cobbler password]: grep COBBLER ~/openstack-configs/secrets.yaml | awk -F":" '{print
    $2}'
   ```

5. Check that the gateway of management/provision network is not the same as that of management interface IP address of the management node.
6. Check L2/L3 network connectivity between the failing node and the management node. Also, check the VPC configuration and port-channel status.
7. Check that the actual PXE boot order is not different from the configured boot-order.
8. Check that PXE (DHCP/TFTP) packets arrive at the management node by performing tcpdump on management interface and looking for UDP 67 or UDP 69 port.
9. Check that HTTP request reaches management node by performing tcpdump on management node management interface on TCP 80 and TCP 443 port. Also check the Docker logs for HTTP request from the failing node: docker exec -it repo_mirror tail -f /var/log/httpd/access_log.
10. Verify all nodes are running on supported Cisco NFVI CIMC firmware or above.
11. Verify supported VIC cards are installed (UCS C-series 1225/1227; UCS B-series: 1240/1280/1340/1380).
12. There are times (especially on redeployment) when the overcloud nodes (controller or compute) for some reason do not get their boot order set correctly even though CIMC says the order is correct. In such instances step 4 of runner fails as PXE boot never completes in time. The workaround is to manually enter the boot menu from the KVM console of the affected nodes and make sure the correct order (NIC interface) is chosen for the PXE boot. Note: the solution is likely to clear the BIOS CMOS before PXE

booting. Will confirm if this solves the problem to avoid waiting on the KVM console at the time of PXE booting

Cisco IMC Connection Problems during Bare Metal Installation

The likely cause is Cisco IMC has too many connections, so the installer cannot connect to it. Clear the connections by logging into your Cisco IMC, going into the Admin->Sessions tab and clearing the connections.

API VIP Connection Problems

Verify the active HAProxy container is running in one of the controller nodes. On that controller within the HAProxy container namespace verify the IP address is assigned to the API interface. Also, verify that your ToR and the network infrastructure connecting your ToR is provisioned with API network segment VLAN.

HAProxy Services Downtime after Initial Installation or HA Failover

The HAProxy web interface can be accessed on TCP port 1936

```
http://<external_lb_vip_address>:1936/
Username: haproxy
Password: <HAPROXY_PASSWORD> from secrets.yaml file
```

After initial installation, the HAProxy web interface might report several OpenStack services with downtime depending upon when that OpenStack service was installed after HAProxy install. The counters are not synchronized between HAProxy active and standby. After HA proxy failover, the downtime timers might change based on the uptime of new active HAproxy container.

Management Node Problems

See the Management Node Recovery Scenarios, on page 183.

Service Commands

To identify all the services that are running, enter:

```
$ systemctl -a | grep docker | grep service
 On controller ignore status of:
docker-neutronlb
On compute ignore status of:
docker-neutronlb, docker-keystone
```

To start a service on a host, enter:

```
$ systemctl start <service_name>
```

To stop a service on a host, enter:

```
$ systemctl stop <service_name>
```

To restart a service on a host, enter:

```
$ systemctl restart <service_name>
```

To check service status on a host, enter:

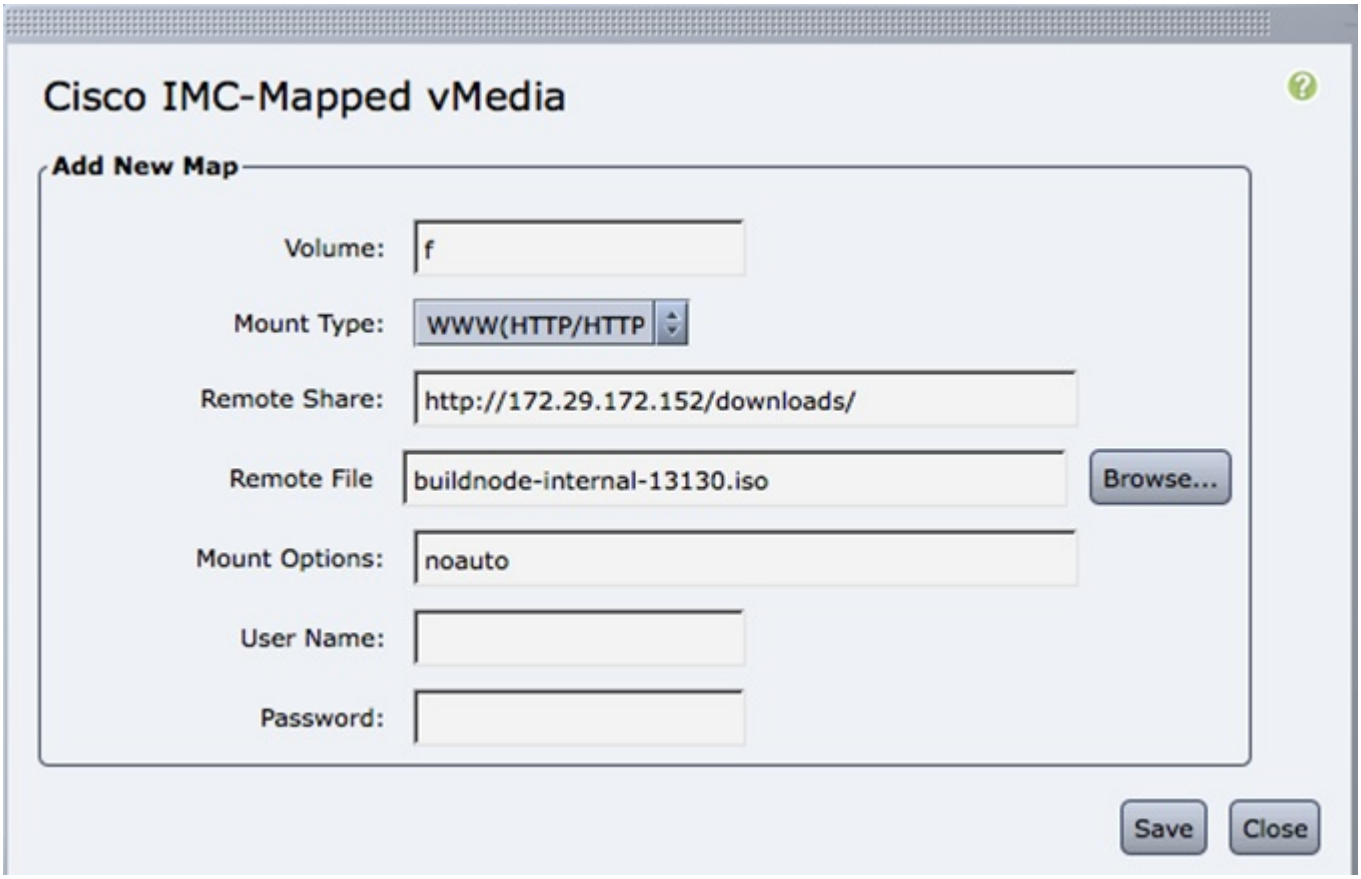
```
$ systemctl status <service_name>
```

Managing CIMC and ISO Installation

When you are remote it is good to map the ISO through the CIMC Mapped vMedia.

To add new mapping:

Step 1 Click **Server > Remote Presence > Virtual Media > Add New Mapping.**



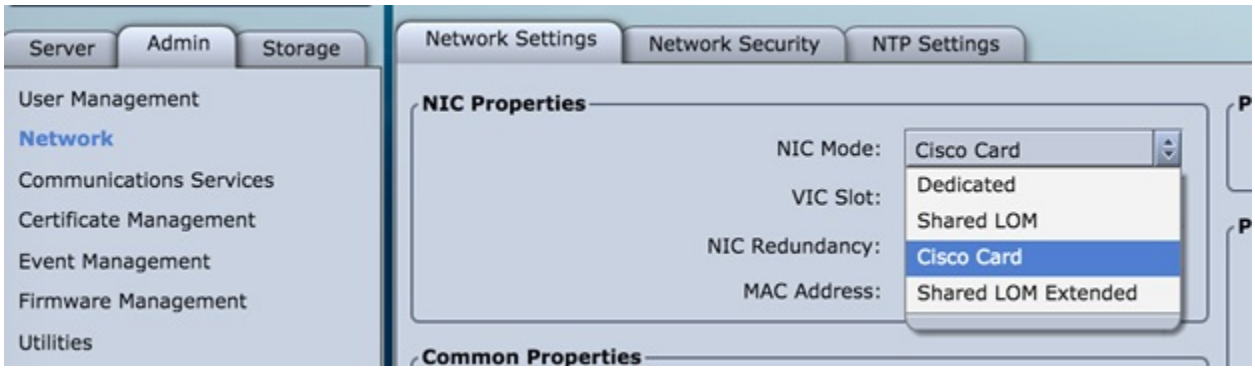
Step 2 Enter the field values such as the Volume, Mount Type, Remote Share, Remote File, User name, and Password.

Step 3 Click **Save.** .The CIMC pulls the ISO directly from the HTTP server.

Management Node Installation Fails

Management node installation fails if the CIMC is configured for cisco card mode.

Choose the dedicated mode in the following screen:

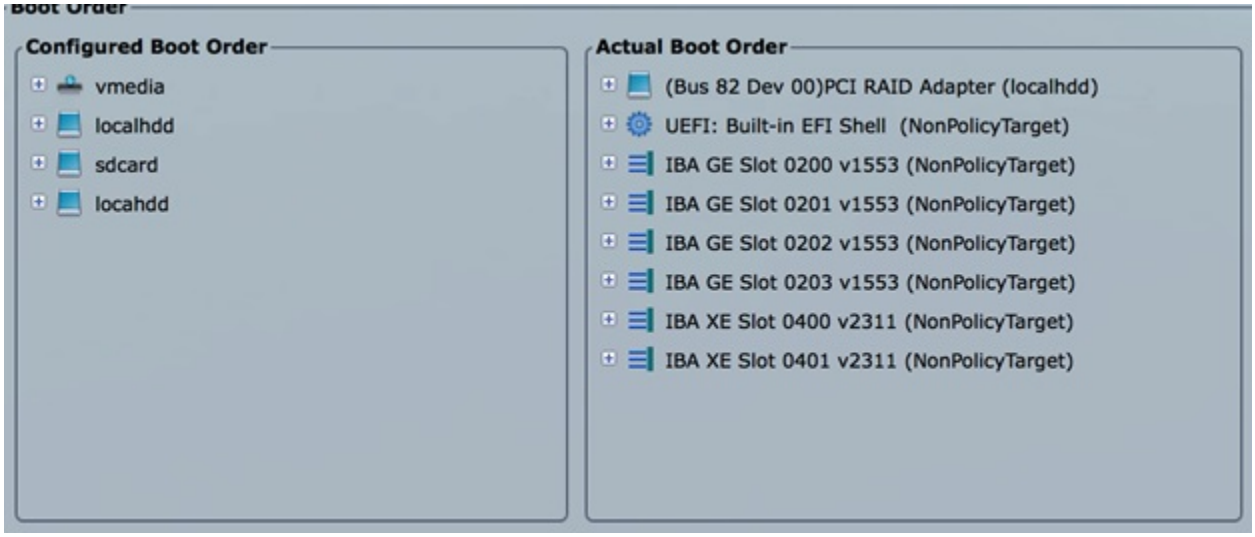


The selected method that is shown in the preceding screen is the incorrect mode.

Configuring Boot Order

Management node does not come up post reboot. It must boot from hard drive to check for the actual boot order.

Choose **Server > BIOS > Configure Boot Order > Boot Order.**



PXE Failure Issue During Baremetal Step

Perform the following steps in case of PXE boot failure:

Step 1 Check log file /var/log/mercury/mercury_baremetal_install.log and connect to failing node CIMC KVM console to find out more on PXE boot failure reason.

Step 2 Ensure all validations (step 1) and hardware validations (step 3) pass.

Step 3 Check log file /var/log/mercury/<UUID>/mercury_baremetal_install.log.

Step 4 Connect to KVM console of failing node(s) to find out more on PXE boot failure.

Step 5 Check L2/L3 network connectivity between failing node(s) and management node.

Step 6 Check for VPC configuration and port-channel status of failing node(s) and ensure *no lacp suspend-individual* is configured on the port-channel.

Step 7 Check the actual PXE boot order must not differ from the boot-order configured.

Step 8 Perform tcpdump on the management node interface br_mgmt to watch for UDP port 67 (dhcp) or UDP pot 69 (tftp) tcpdump -I br_mgmt port 67 or port 69 # on the management node.

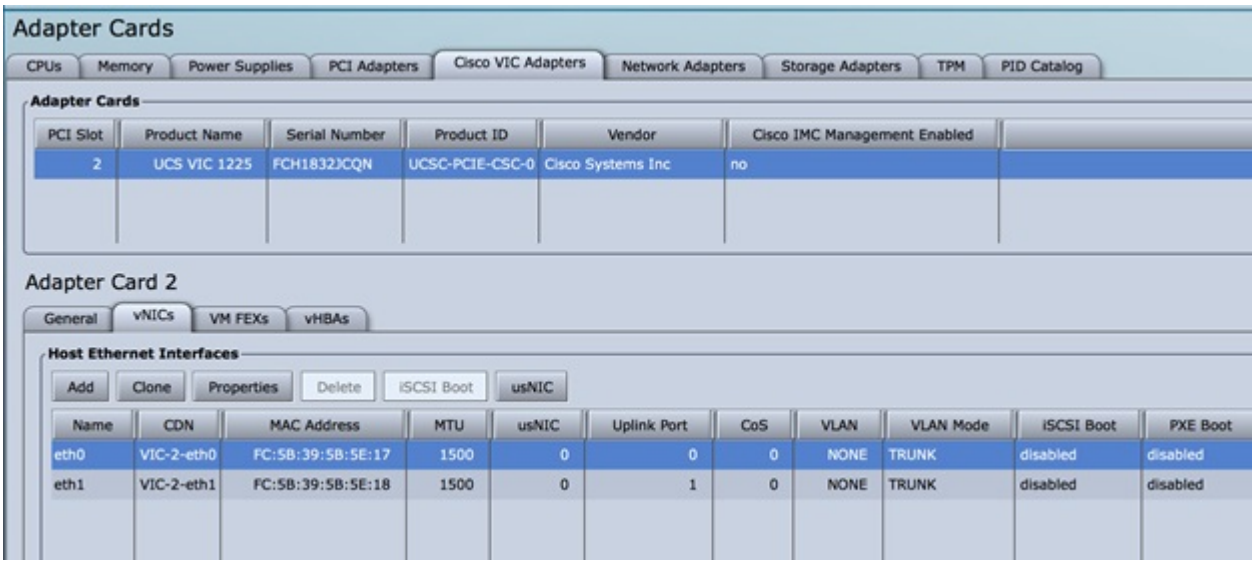
Step 9 Perform tcpdump on the management node management interfacebr_mgmt on TCP 80 tcpdump -I br_mgmt port 80 # on the management node.

Step 10 Check the apache log to watch the management IP address of failing node (if static allocated) tail -f /var/log/cobblerhttpd/access_log # on the management node.

Step 11 For Authorization Required error messages during bare metal (Step 4) with CIMC operations such as hardware validations or cleaning up vNIC, check whether the maximum allowed simultaneous connection (4) are in use. All four connections are run when the 3rd party application monitoring CIMC does not properly close CIMC. This makes CiscoVIM installer not to log in using xmlapi with valid username and password. Check Cisco IMC logs on CIMC (Server > Faults and Logs > Cisco IMC Logs) for the reason why user was denied the access (maximum session, incorrect credentials.). The workaround is to disable 3rd party monitoring, wait at least 10 minutes and then perform CiscoVIM operations.

Step 12 In case none of the nodes are getting DHCP address; DHCP requests arrive at the management node but no response goes out, then check CIMC VIC adapter settings. Server > Inventory > Cisco VIC Adapters > vNICs | VLAN & VLAN Mode. Ensure the VLAN (both id and mode) configured does not matche with that of N9K switch

Option	Description
CIMC	Trunk:None
Switch	Access:vlan_mgmt



The following topics provide Cisco NFVI general troubleshooting procedures.

Container Download Problems

1. Check installer logs log file /var/log/mercury/mercury_buildorchestration.log for any build node orchestration failures including stuck "registry-Populate local registry". Downloaing the Docker container from your management node can be slow.
2. Check the network connectivity between the management node and the remote registry in defaults.yaml on the management node (grep "^registry:" openstack-configs/defaults.yaml).
3. Verify valid remote registry credentials are defined in setup_data.yaml file.
4. A proxy server is required to pull the container images from remote registry. If a proxy is required, exclude all IP addresses for your setup including management node.

Cisco IMC Connection Problems during Bare Metal Installation

The cause may be Cisco IMC has too many connections, so the installer cannot connect to it. Clear the connections by logging into your Cisco IMC, going into the Admin->Sessions tab and clearing the connections.

API VIP Connection Problems

Verify the active HAProxy container is running in one of the controller nodes. On that controller within the HAProxy container namespace verify the IP address is assigned to the API interface. Also, verify that your ToR and the network infrastructure connecting your ToR is provisioned with API network segment VLAN.

HAProxy Services Downtime after Initial Installation or HA Failover

The HAProxy web interface can be accessed on TCP port 1936

```
http://<external_lb_vip_address>:1936/
Username: haproxy
Password: <HAPROXY_PASSWORD> from secrets.yaml file
```

After initial installation, the HAProxy web interface can report to several OpenStack services with downtime depending upon when that OpenStack service was installed after HAProxy install. The counters are not synchronized between HAProxy active and standby. After HA proxy failover, the downtime timers can change based on the uptime of new active HAproxy container.

Management Node Problems

Service Commands

To identify all the services that are running, enter:

```
$ systemctl -a | grep docker | grep service
 On controller ignore status of:
docker-neutronlb
On compute ignore status of:
docker-neutronlb, docker-keystone
```

To start a service on a host, enter:

```
$ systemctl start <service_name>
```

To stop a service on a host, enter:

```
$ systemctl stop <service_name>
```

To restart a service on a host, enter:

```
$ systemctl restart <service_name>
```

To check service status on a host, enter:

```
$ systemctl status <service_name>
```

Connecting to Docker Container

To connect to the docket container do the following:

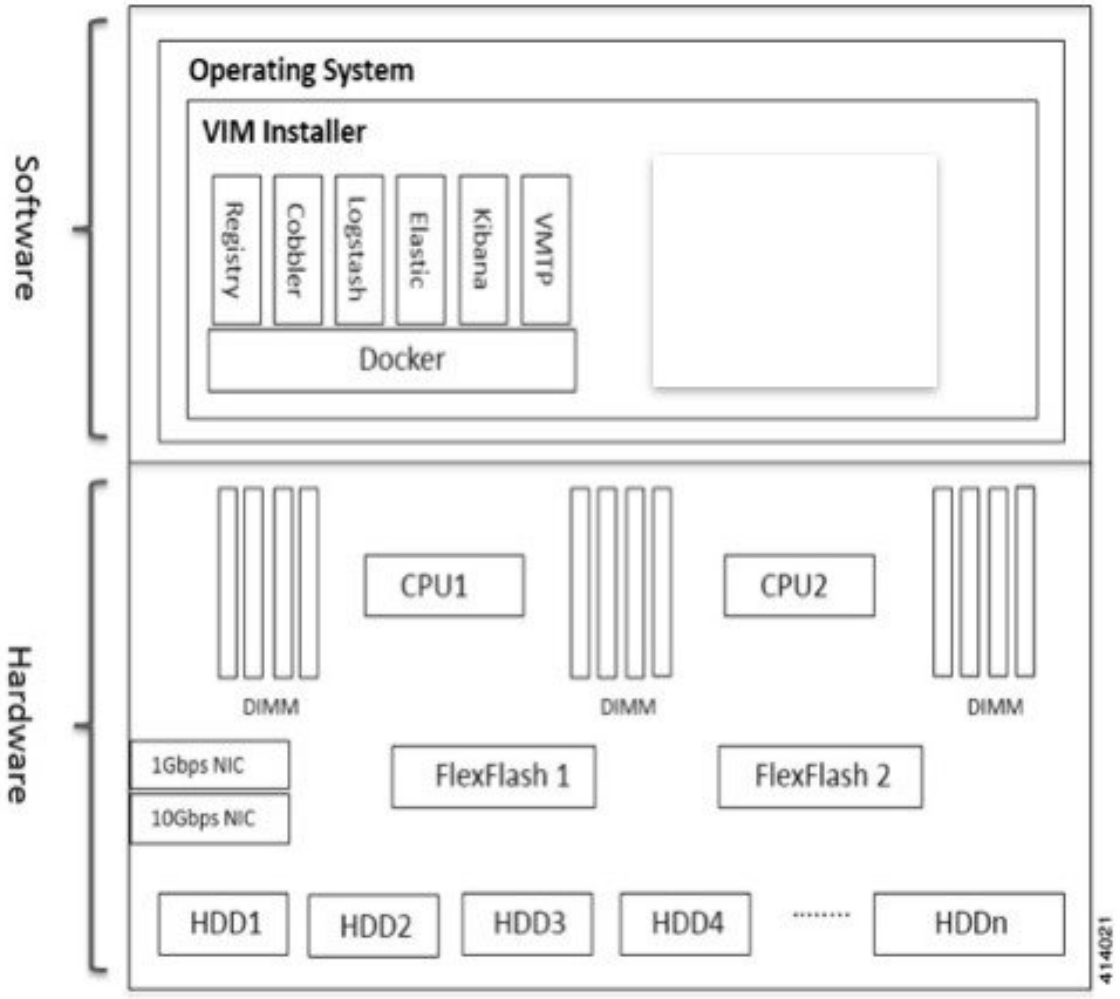
```
# generally, aliases are created for all containers
# use alias to identify those
alias | grep in_container
# checking specific alias by name
alias cobbler

# check docker containers
# alias created by CVIM
dp
# list docker containers
docker ps -a
# list docker images
docker images

# connecting to container
docker exec -it my_cobbler_<tag_id> /bin/bash

# connecting to docker container as privileged user
docker exec -it -u root my_cobbler_<tag_id> /bin/bash

# systemctl files
systemctl -a | egrep "docker-.*.service"

# check specific service
systemctl status mercury-restapi -l
systemctl status docker-vmtp

# restart specific service
systemctl restart docker-vmtp
```

Management Node Recovery Scenarios

The Cisco NFVI management node hosts the Cisco VIM Rest API service, Cobbler for PXE services, ELK for Logging/Kibana dashboard services and VMTP for cloud validation. Because the maintenance node currently does not have redundancy, understanding its points of failure and recovery scenarios is important. These are described in this topic.

The management node architecture includes a Cisco UCS C240 M4 server with dual CPU socket. It has a 1 Gbps on-board (LOM) NIC and a 10 Gbps Cisco VIC mLOM. HDDs are used in 8,16, or 24 disk configurations. The following figure shows a high level maintenance node hardware and software architecture.

Figure 19: Cisco NFVI Management Node Architecture



Different management node hardware or software failures can cause Cisco NFVI service disruptions and outages. Some failed services can be recovered through manual intervention. In cases where the system is operational during a failure, double faults might not be recoverable. The following table lists different management node failure scenarios and their recovery options.

Table 3: Management Node Failure Scenarios

Scenario #	Failure/Trigger	Recoverable?	Operational Impact
1	Failure of 1 or 2 active HDD	Yes	No
2	Simultaneous failure of more than 2 active HDD	No	Yes
3	Spare HDD failure: 4 spare for 24 HDD; or 2 spare for 8 HDD	Yes	No

Scenario #	Failure/Trigger	Recoverable?	Operational Impact
4	Power outage/hard reboot	Yes	Yes
5	Graceful reboot	Yes	Yes
6	Docker daemon start failure	Yes	Yes
7	Service container (Cobbler, ELK) start failure	Yes	Yes
8	One link failure on bond interface	Yes	No
9	Two link failures on bond interface	Yes	Yes
10	REST API service failure	Yes	No
11	Graceful reboot with Cisco VIM Insight	Yes	Yes; CLI alternatives exist during reboot.
12	Power outage/hard reboot with Cisco VIM Insight	Yes	Yes
13	VIM Insight Container reinstallation	Yes	Yes; CLI alternatives exist during re-insight.
14	Cisco VIM Insight Container reboot	Yes	Yes; CLI alternatives exist during reboot.
15	Intel 1350 1Gbps LOM failure	Yes	Yes
16	Cisco VIC 1227 10 Gbps mLOM failure	Yes	Yes
17	DIMM memory failure	Yes	No
18	One CPU failure	Yes	No

Scenario 1: Failure of one or two active HDDs

The management node has either 8,16, or 24 HDDs. The HDDs are configured with RAID 6, which helps enable data redundancy and storage performance and overcomes any unforeseen HDD failures.

- When 8 HDDs are installed, 7 are active disks and one is spare disk.
- When 16 HDDs are installed, 14 are active disks and two are spare disks.
- When 24 HDDs are installed, 20 are active disks and four are spare disks.

With RAID 6 up to two simultaneous active HDD failures can occur. When an HDD fails, the system starts automatic recovery by moving the spare disk to active state and starts recovering and rebuilding the new active HDD. It takes approximately four hours to rebuild the new disk and move to synchronized state. During this operation, the system is completely functional and no impacts are seen. However, you must monitor the system to ensure that additional failures do not occur to enter into a double fault situation.

You can use the **storcli** commands to check the disk and RAID state as shown below:

Note Make sure the node is running with hardware RAID by checking the storcli output and comparing to the one preceding. If hardware RAID is not found, refer to Cisco NFVI Admin Guide 1.0 for HDDs replacement or contact TAC.

```
[root@mgmt-node ~]# /opt/MegaRAID/storcli/storcli64 /c0 show

<…snip…>

TOPOLOGY:
========
-----------------------------------------------------------------------------
DG Arr Row EID:Slot DID Type  State BT       Size PDC  PI SED DS3  FSpace TR
-----------------------------------------------------------------------------
 0 -   -   -        -   RAID6 Optl  N    4.087 TB dflt N  N   dflt N      N
 0 0   -   -        -   RAID6 Optl  N    4.087 TB dflt N  N   dflt N      N       <== RAID
 6 in optimal state
 0 0   0   252:1    1   DRIVE Onln  N  837.258 GB dflt N  N   dflt -      N
 0 0   1   252:2    2   DRIVE Onln  N  837.258 GB dflt N  N   dflt -      N
 0 0   2   252:3    3   DRIVE Onln  N  930.390 GB dflt N  N   dflt -      N
 0 0   3   252:4    4   DRIVE Onln  N  930.390 GB dflt N  N   dflt -      N
 0 0   4   252:5    5   DRIVE Onln  N  930.390 GB dflt N  N   dflt -      N
 0 0   5   252:6    6   DRIVE Onln  N  930.390 GB dflt N  N   dflt -      N
 0 0   6   252:7    7   DRIVE Onln  N  930.390 GB dflt N  N   dflt -      N
 0 -   -   252:8    8   DRIVE DHS   -  930.390 GB -    -  -   -    -      N

-----------------------------------------------------------------------------

<…snip…>

PD LIST:
=======
-------------------------------------------------------------------------
EID:Slt DID State DG       Size Intf Med SED PI SeSz Model            Sp
-------------------------------------------------------------------------
252:1     1 Onln   0 837.258 GB SAS  HDD N   N  512B ST900MM0006      U   <== all disks
functioning
252:2     2 Onln   0 837.258 GB SAS  HDD N   N  512B ST900MM0006      U
252:3     3 Onln   0 930.390 GB SAS  HDD N   N  512B ST91000640SS     U
252:4     4 Onln   0 930.390 GB SAS  HDD N   N  512B ST91000640SS     U
252:5     5 Onln   0 930.390 GB SAS  HDD N   N  512B ST91000640SS     U
252:6     6 Onln   0 930.390 GB SAS  HDD N   N  512B ST91000640SS     U
252:7     7 Onln   0 930.390 GB SAS  HDD N   N  512B ST91000640SS     U
252:8     8 DHS    0 930.390 GB SAS  HDD N   N  512B ST91000640SS     D
-------------------------------------------------------------------------


[root@mgmt-node ~]# /opt/MegaRAID/storcli/storcli64 /c0 show

<…snip…>

TOPOLOGY :
========
-----------------------------------------------------------------------------
DG Arr Row EID:Slot DID Type  State BT       Size PDC  PI SED DS3  FSpace TR
-----------------------------------------------------------------------------
 0 -   -   -        -   RAID6 Pdgd  N    4.087 TB dflt N  N   dflt N      N    <== RAID 6
in degraded state
 0 0   -   -        -   RAID6 Dgrd  N    4.087 TB dflt N  N   dflt N      N
 0 0   0   252:8    8   DRIVE Rbld  Y  930.390 GB dflt N  N   dflt -      N
 0 0   1   252:2    2   DRIVE Onln  N  837.258 GB dflt N  N   dflt -      N
```

```
 0 0   2   252:3    3   DRIVE Onln  N  930.390 GB dflt N  N   dflt -      N
 0 0   3   252:4    4   DRIVE Onln  N  930.390 GB dflt N  N   dflt -      N
 0 0   4   252:5    5   DRIVE Onln  N  930.390 GB dflt N  N   dflt -      N
 0 0   5   252:6    6   DRIVE Onln  N  930.390 GB dflt N  N   dflt -      N
 0 0   6   252:7    7   DRIVE Onln  N  930.390 GB dflt N  N   dflt -      N
-------------------------------------------------------------------------------

<…snip…>

PD LIST :
========
-------------------------------------------------------------------------
EID:Slt DID State DG       Size Intf Med SED PI SeSz Model            Sp
-------------------------------------------------------------------------
252:1     1 UGood -  837.258 GB SAS  HDD N   N  512B ST900MM0006      U    <== active disk
 in slot 1 disconnected from drive group 0
252:2     2 Onln  0  837.258 GB SAS  HDD N   N  512B ST900MM0006      U
252:3     3 Onln  0  930.390 GB SAS  HDD N   N  512B ST91000640SS     U
252:4     4 Onln  0  930.390 GB SAS  HDD N   N  512B ST91000640SS     U
252:5     5 Onln  0  930.390 GB SAS  HDD N   N  512B ST91000640SS     U
252:6     6 Onln  0  930.390 GB SAS  HDD N   N  512B ST91000640SS     U
252:7     7 Onln  0  930.390 GB SAS  HDD N   N  512B ST91000640SS     U
252:8     8 Rbld  0  930.390 GB SAS  HDD N   N  512B ST91000640SS     U    <== spare disk
in slot 8 joined drive group 0 and in rebuilding state
-------------------------------------------------------------------------



[root@mgmt-node ~]# /opt/MegaRAID/storcli/storcli64 /c0/e252/s8 show rebuild
Controller = 0
Status = Success
Description = Show Drive Rebuild Status Succeeded.

-------------------------------------------------------
Drive-ID    Progress% Status      Estimated Time Left
-------------------------------------------------------
/c0/e252/s8        20 In progress 2 Hours 28 Minutes     <== spare disk in slot 8 rebuild
status
-------------------------------------------------------
```

To replace the failed disk and add it back as a spare:

```
[root@mgmt-node ~]# /opt/MegaRAID/storcli/storcli64 /c0/e252/s1 add hotsparedrive dg=0
Controller = 0
Status = Success
Description = Add Hot Spare Succeeded.


[root@mgmt-node ~]# /opt/MegaRAID/storcli/storcli64 /c0 show

<…snip…>

TOPOLOGY :
========
-----------------------------------------------------------------------------
DG Arr Row EID:Slot DID Type  State BT       Size PDC  PI SED DS3  FSpace TR
-----------------------------------------------------------------------------
 0 -   -   -        -   RAID6 Pdgd  N    4.087 TB dflt N  N   dflt N      N
 0 0   -   -        -   RAID6 Dgrd  N    4.087 TB dflt N  N   dflt N      N
 0 0   0   252:8    8   DRIVE Rbld  Y  930.390 GB dflt N  N   dflt -      N
 0 0   1   252:2    2   DRIVE Onln  N  837.258 GB dflt N  N   dflt -      N
 0 0   2   252:3    3   DRIVE Onln  N  930.390 GB dflt N  N   dflt -      N
 0 0   3   252:4    4   DRIVE Onln  N  930.390 GB dflt N  N   dflt -      N
```

```
 0 0   4   252:5    5   DRIVE Onln  N  930.390 GB dflt N  N   dflt -      N
 0 0   5   252:6    6   DRIVE Onln  N  930.390 GB dflt N  N   dflt -      N
 0 0   6   252:7    7   DRIVE Onln  N  930.390 GB dflt N  N   dflt -      N
 0 -   -   252:1    1   DRIVE DHS   -  837.258 GB -    -  -   -    -      N
------------------------------------------------------------------------------

<…snip…>

PD LIST :
=======
-------------------------------------------------------------------------
EID:Slt DID State DG       Size Intf Med SED PI SeSz Model            Sp
-------------------------------------------------------------------------
252:1     1 DHS   0  837.258 GB SAS  HDD N   N  512B ST900MM0006      U    <== replacement
 disk added back as spare
252:2     2 Onln  0  837.258 GB SAS  HDD N   N  512B ST900MM0006      U
252:3     3 Onln  0  930.390 GB SAS  HDD N   N  512B ST91000640SS     U
252:4     4 Onln  0  930.390 GB SAS  HDD N   N  512B ST91000640SS     U
252:5     5 Onln  0  930.390 GB SAS  HDD N   N  512B ST91000640SS     U
252:6     6 Onln  0  930.390 GB SAS  HDD N   N  512B ST91000640SS     U
252:7     7 Onln  0  930.390 GB SAS  HDD N   N  512B ST91000640SS     U
252:8     8 Rbld  0  930.390 GB SAS  HDD N   N  512B ST91000640SS     U
-------------------------------------------------------------------------
```

Scenario 2: Simultaneous failure of more than two active HDDs

If more than two HDD failures occur at the same time, the management node goes into an unrecoverable failure state because RAID 6 allows for recovery of up to two simultaneous HDD failures. To recover the management node, reinstall the operating system.

Scenario 3: Spare HDD failure

When the management node has 24 HDDs, four are designated as spares. Failure of any of the disks does not impact the RAID or system functionality. Cisco recommends replacing these disks when they fail (see the steps in Scenario 1) to serve as standby disks and so when an active disk fails, an auto-rebuild is triggered.

Scenario 4: Power outage/hard reboot

If a power outage or hard system reboot occurs, the system will boot up and come back to operational state. Services running on management node during down time will be disrupted. See the steps in Scenario 9 for the list of commands to check the services status after recovery.

Scenario 5: System reboot

If a graceful system reboot occurs, the system will boot up and come back to operational state. Services running on management node during down time will be disrupted. See the steps in Scenario 9 for the list of commands to check the services status after recovery.

Scenario 6: Docker daemon start failure

The management node runs the services using Docker containers. If the Docker daemon fails to come up, it causes services such as ELK, Cobbler and VMTP to go into down state. You can use the **systemctl** command to check the status of the Docker daemon, for example:

```
# systemctl status docker
docker.service - Docker Application Container Engine
Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
Active: active (running) since Mon 2016-08-22 00:33:43 CEST; 21h ago
Docs: http://docs.docker.com
Main PID: 16728 (docker)
```

If the Docker daemon is in down state, use the **systemctl restart docker** command to restart the Docker service. Run the commands listed in Scenario 9 to verify that all the Docker services are active.

Scenario 7: Service container (Cobbler, ELK) start failure

As described in Scenario 8, all the services run as Docker containers on the management node. To find all services running as containers, use the **docker ps –a** command. If any services are in Exit state, use the **systemctl** command and grep for Docker to find the exact service name, for example:

```
# systemctl | grep docker- | awk '{print $1}'
  docker-cobbler-tftp.service
  docker-cobbler-web.service
  docker-cobbler.service
  docker-container-registry.service
  docker-elasticsearch.service
  docker-kibana.service
  docker-logstash.service
  docker-vmtp.service
```

If any services need restarting, use the **systemctl** command. For example, to restart a Kibana service:

```
# systemctl restart docker-kibana.service
```

Scenario 8: One link failure on the bond Interface

The management node is set up with two different networks: br_api and br_mgmt. The br_api interface is the external. It is used for accessing outside services such as the Cisco VIM REST API, Kibana and Cobbler. The br_mgmt interface is internal. It is used for provisioning and to provide management connectivity to all OpenStack nodes (control, compute and storage). Each network has two ports that are bonded to provide redundancy. If one port fails, the system will remain completely functional through the other port. If a port fails, check for physical network connectivity and/or remote switch configuration to debug the underlying cause of the link failure.

Scenario 9: Two link failures on the bond Interface

As described in Scenario 10, each network is configured with two ports. If both ports are down, the system is not reachable and management node services could be disrupted. After the ports are back up, the system is fully operational. Check the physical network connectivity and/or the remote switch configuration to debug the underlying link failure cause.

Scenario 10: REST API service failure

The management node runs the REST API service for Cisco VIM clients to reach the server. If the REST service is down, Cisco VIM clients cannot reach the server to trigger any server operations. However, with the exception of the REST service, other management node services remain operational.

To verify the management node REST services are fully operational, use the following command to check that the httpd and mercury-restapi services are in active and running state:

```
# systemctl status httpd
httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)

   Active: active (running) since Mon 2016-08-22 00:22:10 CEST; 22h ago

# systemctl status mercury-restapi.service
mercury-restapi.service - Mercury Restapi
   Loaded: loaded (/usr/lib/systemd/system/mercury-restapi.service; enabled; vendor preset:
 disabled)
   Active: active (running) since Mon 2016-08-22 00:20:18 CEST; 22h ago
```

A tool is also provided so that you can check the REST API server status and the location of the directory it is running from. To execute run the following command:

```
# cd installer-<tagid>/tools
#./restapi.py -a status
```

```
   Status of the REST API Server:  active (running) since Thu 2016-08-18 09:15:39 UTC; 9h
ago
   REST API launch directory: /root/installer-<tagid>/
```

Confirm the server status is active and check that the restapi launch directory matches the directory where the installation was launched. The restapi tool also provides the options to launch, tear down, and reset password for the restapi server as shown below:

```
# ./restapi.py -h

usage: restapi.py [-h] --action ACTION [--yes] [--verbose]

REST API setup helper

optional arguments:
  -h, --help            show this help message and exit
  --action ACTION, -a ACTION
                        setup - Install and Start the REST API server.
                        teardown - Stop and Uninstall the REST API
       server.
                        restart - Restart the REST API server.
                        regenerate-password - Regenerate the password for
       REST API server.
                        reset-password - Reset the REST API password with
      user given password.
                        status - Check the status of the REST API server
  --yes, -y             Skip the dialog. Yes to the action.
  --verbose, -v         Perform the action in verbose mode.
```

If the REST API server is not running, execute **./ciscovimclient/ciscovim** to show the following error message:

```
# cd installer-<tagid>/
#./ciscovimclient/ciscovim -setupfile ~/Save/<setup_data.yaml> run
  ERROR: Error communicating with https://<api_ip:8445> [Errno 111] Connection refused
```

If the installer directory or the REST API state is not correct or points to an incorrect REST API launch directory, go to the installer-<tagid>/tools directory and execute:

```
# ./restapi.py -action setup
```

To confirm that the REST API server state and launch directory is correct run the following command:

```
# ./restapi.py -action status
```

Scenario 11: Graceful reboot with Cisco VIM Insight

Cisco VIM Insight runs as a container on the management node. After a graceful reboot of the management node, the VIM Insight and its associated database containers will come up. So there is no impact on recovery.

Scenario 12: Power outage or hard reboot with VIM Insight

The Cisco VIM Insight container will come up automatically following a power outage or hard reset of the management node.

Scenario 13: Cisco VIM Insight reinstallation

If the management node which is running the Cisco VIM Insight fails and cannot come up, you must uninstall and reinstall the Cisco VIM Insight. After the VM Insight container comes up, add the relevant bootstrap steps as listed in the install guide to register the pod. VIM Insight then automatically detects the installer status and reflects the current status appropriately.

To clean up and reinstall Cisco VIM Insight run the following command:

```
# cd /root/installer-<tagid>/insight/
# ./bootstrap_insight.py –a uninstall –o standalone –f </root/insight_setup_data.yaml>
```

Scenario 14: VIM Insight Container reboot

On Reboot of the VIM Insight container, services will continue to work as it is.

Scenario 15: Intel (I350) 1Gbps LOM failure

The management node is set up with an Intel (I350) 1 Gbps LOM for API connectivity. Two 1 Gbps ports are bonded to provide connectivity redundancy. No operational impact occurs if one of these ports goes down. However, if both ports fail, or the LOM network adapter fails, the system cannot be reached through the API IP address. If this occurs you must replace the server because the LOM is connected to the system motherboard. To recover the management node with a new server, complete the following steps. Make sure the new management node hardware profile matches the existing server and the Cisco IMC IP address is assigned.

1. Shut down the existing management node.
2. Unplug the power from the existing and new management nodes.
3. Remove all HDDs from existing management node and install them in the same slots of the new management node.
4. Plug in the power to the new management node, but do not boot the node.
5. Verify the configured boot order is set to boot from local HDD.
6. Verify the Cisco NFVI management VLAN is configured on the Cisco VIC interfaces.
7. Boot the management node for the operating system to start.

   After the management node is up, the management node bond interface will be down due to the incorrect MAC address on the ifcfg files. It will point to old node network card MAC address.
8. Update the MAC address on the ifcfg files under /etc/sysconfig/network-scripts.
9. Reboot the management node.

   It will come up and be fully operational. All interfaces should be in an up state and be reachable.
10. Verify that Kibana and Cobbler dashboards are accessible.
11. Verify the Rest API services are up. See Scenario 15 for any recovery steps.

Scenario 16: Cisco VIC 1227 10Gbps mLOM failure

The management node is configured with a Cisco VIC 1227 dual port 10 Gbps mLOM adapter for connectivity to the other Cisco NFVI nodes. Two 10 Gbps ports are bonded to provide connectivity redundancy. If one of the 10 Gbps ports goes down, no operational impact occurs. However, if both Cisco VIC 10 Gbps ports fail, the system goes into an unreachable state on the management network. If this occurs, you must replace the VIC network adapters. Otherwise pod management and the Logstash forwarding service will be disrupted. If you replace a Cisco VIC, update the management and provisioning VLAN for the VIC interfaces using Cisco IMC and update the MAC address in the interfaces under /etc/sysconfig/network-scripts interface configuration file.

Scenario 17: DIMM memory failure

The management node is set up with multiple DIMM memory across different slots. Failure of one or memory modules could cause the system to go into unstable state, depending on how many DIMM memory failures occur. DIMM memory failures are standard system failures like any other Linux system server. If a DIMM memory fails, replace the memory module(s) as soon as possible to keep the system in stable state.

Scenario 18: One CPU failure

Cisco NFVI management nodes have dual core Intel CPUs (CPU1 and CPU2). If one CPU fails, the system remains operational. However, always replace failed CPU modules immediately. CPU failures are standard system failures like any other Linux system server. If a CPU fails, replace it immediately to keep the system in stable state.

Recovering Compute Node Scenario

The Cisco NFVI Compute node hosts the OpenStack services to provide processing, network, and storage resources to run instances. The node architecture includes a Cisco UCS C220 M4 server with dual CPU socket, 10 Gbps Cisco VIC mLOM, and two HDDs in RAID 1 configuration.

Failure of one active HDD

With RAID 1, data are mirrored and can allow up to one active HDD failure. When a HDD fails, the node is still functional with no impacts. However, the data are no longer mirrored and losing another HDD will result in unrecoverable and operational downtime. The failed disk should be replaced soon as it takes approximately two hours to rebuild the new disk and move to synchronized state.

To check the disk and RAID state run the storcli commands as follows:

Note Make sure that the node is running with hardware RAID by checking the storcli output and comparing to the one below. If hardware RAID is not found, refer to Cisco NFVI Admin Guide 1.0 for HDDs replacement or contact TAC.

```
[root@compute-node ~]# /opt/MegaRAID/storcli/storcli64 /c0 show

<…snip…>

TOPOLOGY :
========
-----------------------------------------------------------------------------
DG Arr Row EID:Slot DID Type  State BT       Size PDC  PI SED DS3  FSpace TR
-----------------------------------------------------------------------------
 0 -   -   -        -   RAID1 Optl  N  837.258 GB dflt N  N   dflt N      N  <== RAID 1 in
 optimal state
 0 0   -   -        -   RAID1 Optl  N  837.258 GB dflt N  N   dflt N      N
 0 0   0   252:2    9   DRIVE Onln  N  837.258 GB dflt N  N   dflt -      N
 0 0   1   252:3    11  DRIVE Onln  N  837.258 GB dflt N  N   dflt -      N
-----------------------------------------------------------------------------

<…snip…>

Physical Drives = 2

PD LIST :
=======
-------------------------------------------------------------------------
EID:Slt DID State DG       Size Intf Med SED PI SeSz Model            Sp
-------------------------------------------------------------------------
252:2     9 Onln   0 837.258 GB SAS  HDD N   N  512B ST900MM0006      U   <== all disks
functioning
252:3    11 Onln   0 837.258 GB SAS  HDD N   N  512B ST900MM0006      U
-------------------------------------------------------------------------
```

```
[root@compute-node ~]# /opt/MegaRAID/storcli/storcli64 /c0 show

<…snip…>

TOPOLOGY :
========
-----------------------------------------------------------------------------
DG Arr Row EID:Slot DID Type  State BT       Size PDC  PI SED DS3  FSpace TR
-----------------------------------------------------------------------------
 0 -   -   -        -   RAID1 Dgrd  N  837.258 GB dflt N  N   dflt N      N  <== RAID 1 in
 degraded state.
 0 0   -   -        -   RAID1 Dgrd  N  837.258 GB dflt N  N   dflt N      N
 0 0   0   -        -   DRIVE Msng  -  837.258 GB -    -  -   -    -      N
 0 0   1   252:3    11  DRIVE Onln  N  837.258 GB dflt N  N   dflt -      N
-----------------------------------------------------------------------------

<…snip…>

PD LIST :
=======
-------------------------------------------------------------------------
EID:Slt DID State DG       Size Intf Med SED PI SeSz Model            Sp
-------------------------------------------------------------------------
252:2     9 UGood -  837.258 GB SAS  HDD N   N  512B ST900MM0006      U   <== active disk
in slot 2 disconnected from drive group 0
252:3    11 Onln  0  837.258 GB SAS  HDD N   N  512B ST900MM0006      U
-------------------------------------------------------------------------
```

To replace the failed disk and add it back as a spare run the following command:

```
[root@compute-node ~]# /opt/MegaRAID/storcli/storcli64 /c0/e252/s2 add hotsparedrive dg=0
Controller = 0
Status = Success
Description = Add Hot Spare Succeeded.


[root@compute-node ~]# /opt/MegaRAID/storcli/storcli64 /c0 show

<…snip…>

TOPOLOGY :
========
-----------------------------------------------------------------------------
DG Arr Row EID:Slot DID Type  State BT       Size PDC  PI SED DS3  FSpace TR
-----------------------------------------------------------------------------
 0 -   -   -        -   RAID1 Dgrd  N  837.258 GB dflt N  N   dflt N      N
 0 0   -   -        -   RAID1 Dgrd  N  837.258 GB dflt N  N   dflt N      N
 0 0   0   252:2    9   DRIVE Rbld  Y  837.258 GB dflt N  N   dflt -      N
 0 0   1   252:3    11  DRIVE Onln  N  837.258 GB dflt N  N   dflt -      N
-----------------------------------------------------------------------------

<…snip…>

PD LIST :
=======
-------------------------------------------------------------------------
EID:Slt DID State DG       Size Intf Med SED PI SeSz Model            Sp
-------------------------------------------------------------------------
252:2     9 Rbld   0 837.258 GB SAS  HDD N   N  512B ST900MM0006      U   <== replacement
disk in slot 2 joined device group 0 and in rebuilding state
252:3    11 Onln   0 837.258 GB SAS  HDD N   N  512B ST900MM0006      U
-------------------------------------------------------------------------
```

```
[root@compute-node ~]# /opt/MegaRAID/storcli/storcli64 /c0/e252/s2 show rebuild
Controller = 0
Status = Success
Description = Show Drive Rebuild Status Succeeded.


------------------------------------------------------
Drive-ID    Progress% Status      Estimated Time Left
------------------------------------------------------
/c0/e252/s2       10 In progress 1 Hours 9 Minutes    <== replacement disk in slot 2 rebuild
 status
------------------------------------------------------
```

Running the Cisco VIM Technical Support Tool

Cisco VIM includes a tech-support tool that you can use to gather Cisco VIM information to help solve issues working with Cisco Technical Support. The tech-support tool can be extended to execute custom scripts. It can be called after runner is executed at least once. The tech-support tool uses a configuration file that specifies what information to collect. The configuration file is located in the following location: /root/openstack-configs/tech-support/tech_support_cfg.yaml.

The tech-support tool keeps track of the point where the Cisco VIM installer has executed and collects the output of files or commands indicated by the configuration file. For example, if the installer fails at Step 3 (VALIDATION), the tech-support will provide information listed in the configuration file up to Step 3 (included). You can override this default behavior by adding the --stage option to the command.

The tech-support script is located at the management node /root/installer-{tag-id}/tech-support directory. To run it after the runner execution, enter the following command:

```
./tech-support/tech_support.py
```

The command creates a compressed tar file containing all the information it gathered. The file location is displayed in the console at the end of the execution. You do not need to execute the command with any options. However, if you want to override any default behavior, you can use the following options:

```
/tech_support.py --help
Usage: tech_support.py [options]

tech_support.py collects information about your cloud
Options:
  -h, --help            show this help message and exit
  --stage=STAGE         specify the stage where installer left off
  --config-file=CFG_FILE specify alternate configuration file
  --tmp-dir=TMP_DIR     specify alternate temporary directory
  --file-size=TAIL_SIZE specify max size (in KB) of each file collected
```

Where:

- stage—tells the tech-support at which state the installer left off. The possible values are: INPUT_VALIDATION, BUILDNODE_ORCHESTRATION, VALIDATION, BAREMETAL_INSTALL, COMMON_SETUP, CEPH, ORCHESTRATION or VMTP
- config-file—Provides the path for a specific configuration file. Make sure that your syntax is correct. Look at the default /root/tech-support/openstack-configs/tech_support_cfg.yaml file as an example on how to create a new config-file or modify the default file.

- tmp-dir—Provides the path to a temp directory tech-support can use to create the compressed tar file. The tech-support tool provides the infrastructure to execute standard Linux commands from packages included in the Cisco VIM installation. This infrastructure is extensible and you can add commands, files, or custom bash/Python scripts into the appropriate configuration file sections for the tool to collect the output of those commands/scripts. (See the README section at the beginning of the file for more details on how to do this.)
- file-size—Is an integer that specifies (in KB) the maximum file size that tech-support will capture and tail the file if needed. By default, this value is set to 10 MB. For example, if no file-size option is provided and the tech-support needs to collect /var/log/mercury/data.log and the data.log is more than 10 MB, tech-support will get the last 10 MB from /var/log/mercury/data.log.

Tech-support configuration file

Cisco VIM tech-support is a utility tool designed to collect the VIM pod logs which help users to debug the issues offline. The administrator uses the tech-support configuration files to provide the list of commands or configuration files. The tech support tool of the Cisco VIM gathers list of commands or configuration files for the offline diagnostic or debugging purposes.

By default the tech-support configuration file is located at the /root/openstack-configs/tech-support/tech_support_cfg.yaml file. Alternatively, you can use a different one by specifying the -config-file option. The syntax of this configuration file must be as follows:

The tech-support configuration file section is divided into eight sections which corresponds to each of the installer stages:

- INPUT_VALIDATION
- BUILDNODE_ORCHESTRATION
- VALIDATION
- BAREMETAL_INSTALL
- COMMON_SETUP
- CEPH
- ORCHESTRATION
- VMTP

Inside each of these eight sections, there are tags divided on hierarchical levels. At the first level, the tag indicates the host(s) or path on which the command(s) run and from where the file(s) can be collected. The possible tags are as follows:

- - HOSTS_MANAGEMENT: Run in Management node only
- - HOSTS_CONTROL: Run in all the Control nodes
- - HOSTS_COMPUTE: Run in all the Compute nodes
- - HOSTS_STORAGE: Run in all the Storage nodes
- - HOSTS_COMMON: Run in all the Compute and Control nodes

- - HOSTS_ALL: Run in all the Compute, Control and Storage nodes

Note In any of these eight sections, if HOSTS tag is not specified then no information is collected for that stage.

For each of the hosts mentioned above there will be a second level tag which specifies where to run the command. The possible values of those tags are as follows:

- - SERVER_FILES: Path(s) to the file(s) that tech-support needs to collect.
- - SERVER_COMMANDS: Command(s) or script name(s) which need to be executed directly on the desired server. The command(s) need to be already included in the $PATH. For the scripts, please refer to the Custom Scripts paragraph below.
- - CONTAINERS: Indicates the tech-support tool that the command(s) need to be executed and the files to be gathered from inside a container. See the following steps for more specific information of what can be added in this section.

In the CONTAINERS section, indicate the path in which container the commands should be executed or gathered from. This is done with a <container_name> tag (Shown below are example of where to get the string for the <container_name> tag):

- all_containers: Execute inside all containers (regardless of the state).
- <container_name>: This must be the name of a container and it indicates in which container to run the command or gather the information. This will run commands inside the container only if the mentioned container is up (as we cannot run commands on dead containers). Example of how to get the container name:
  - Execute **docker ps** and get the name (without any numbers) of the last column of output **docker ps -a**.

    For example:

```
CONTAINER ID  IMAGE                 COMMAND   <snip>  NAMES
81bc4e54cbfb  <registry>/vmtp:4263  /bin/bash"        vmtp_4263
```

The tech-support runs the linux commands on the server (from packages included in RHEL7.3). Add the name of the commands under the SERVER_COMMANDS section of the configuration file to run the commands.

However, if the administrator wants to add a custom bash or python script to be executed in some set of servers in the cloud. In this case the user just needs to add the script into the custom-scripts directory on the current directory path (/root/openstack-configs/tech-support/) and add the script name into the corresponding SERVER_COMMANDS section.

The tech-support tool will scp the script(s) included in the custom-scripts directory into the appropriate cloud nodes where it will be executed (as# indicated in this config file) and capture the output (stdout and stderr) and add it to the collection of files collected by the tech-support tool. It is assumed that the scripts are self-standing and independent and needs no external input.

Following is an example of a custom tech-support configuration file. This is just an example of what information the tech-support tool will gather if given the following configuration file:

```
COMMON_SETUP:
  HOSTS_ALL:   # All compute, control and storage hosts
    SERVER_FILES:
```

```
  - /usr/lib/docker-storage-setup
SERVER_COMMANDS:
  - docker info
  - my_script.sh
CONTAINERS:
  all_containers:  #execute in all containers (even if they are in down state)
    CONTAINER_COMMANDS:
      - docker inspect
      - docker logs
  logstash:
    CONTAINER_FILES:
      - /var/log/
    CONTAINER_COMMANDS:
      - ls -l
```

Given this example of configuration, and assuming that the installer ended in at least the COMMON_SETUP state, the tech-support tool will run under all OpenStack nodes (Compute, Control and Storage) and it will:

- Gather (if exists) the contents of /usr/lib/docker-storage-setup file.
- Run **docker info** command and collect the output.
- Run **my_script.sh** and collect the output. The **my_script.sh i**s an example of a bash script which the user previously added to the /root/openstack-configs/tech-support/custom-scripts directory.
- Collect the output of docker inspect and docker logs for all containers.
- Collect the files in /var/log inside the logstash container (if there is container with that name). This is equivalent to running the following command (where /tmp indicates a temporary location where the tech-support tool gathers all the information): **docker cp logstash_{tag}:/var/log/ /tmp.**
- Collect the output of the command **docker exec logstash_{{tag}}: ls -l.**

.

Tech-Support When Servers Are Offline

It is difficult to collect the information from the servers if one or more cloud nodes are not reachable. In this case, you can connect through the KVM console into those servers and run the local tech-support tool.

Step 1 To run the local tech-support tool run the following command:

```
/root/tech_support_offline
```

Step 2 Cisco VIM tech_support _offline collects the Logs and other troubleshooting output from the server and place it in the location of the other server:

```
/root/tech_support
```

Note After the server is reachable, you can use the Cisco VIM tech-support tool which collects all the files under the /root/tech-support/ directory which can be used to debug any issue which are offline.

Disk-Maintenance Tool to Manage Physical Drives

In VIM you can use the disk-maintenance tool to check the status of all physical drives that are present in running and operational nodes in the following roles -

- Management
- Control (all or specific nodes)
- Compute (all or specific nodes) (Expect for third party)

This provides the information about the present status of the physical drives - if they are in Online, Offline, Rebuilding, Unconfigured Good or JBOD states if all disks are ok. If not, the disks that have gone bad are displayed with the slot number and server information, that has to be replaced. When multiple disks have to be replaced, we recommend you to execute remove or add of the node.

- Physically remove and insert a new disk before attempting to replace.
- For smooth operation, wipe out disk before attempting replace operations.
- Call Cisco TAC if you face any issue. Do not reattempt.

Note Make sure that each node is running with hardware RAID , the steps for which can be found in the section titled Recovering Compute Node Scenario. Refer to step 15 of the section "Upgrading Cisco VIM Software Using a USB" on how to move the pod from hardware RAID to software RAID.

To check the status of the Diskmgmt log in to the management node and run the ciscovim command with the diskmgmt option. The design of the diskmgmt user interface follows a test job create, list, show, and delete workflow.

Diskmgmt user workflow:

A database of disk operation results is maintained so that you can keep the results of multiple disk check or replace and view them at any time.

Step 1 Run the Help command to see all available command line options:

```
# ciscovim help diskmgmt
usage: ciscovim diskmgmt [--server <node1,node2,...>] [--id <id>]
                         [--locator {on,off}] [--json-display] [-y]
                         create|delete|list|show check-disks|replace-disks
                         all|management|control|compute

HDD maintenance helper

Positional arguments:
  create|delete|list|show         The control command to perform
  check-disks|replace-disks       The identity of the task/action
  all|management|control|compute  The role of the target host(s)

Optional arguments:
  --server <node1,node2,...>      List of specific control/compute host names
                                  within the target role.
```

```
  --id <id>                       ID used to identify specific item to
                                  show/delete.
  --locator {on,off}              Turn on|off locator LED for server with bad
                                  disks and for the physical drives.
  --json-display                  Shows output in JSON format.
  -y, --yes                       Yes option to perform the action
```

Step 2 Check Disk operation creates check-disks operation for all control nodes in the POD. The system responds with a message indicating the Time, ID and when it was Created. Run the following check-disk operation command:

```
# ciscovim diskmgmt create check-disks control
+------------+--------------------------------------+
| Field      | Value                                |
+------------+--------------------------------------+
| action     | check-disks                          |
| command    | create                               |
| created_at | 2018-03-07T21:12:20.684648+00:00     |
| id         | 0c6d27c8-bdac-493b-817e-1ea8640dae57 |
| locator    | False                                |
| result     |                                      |
| role       | control                              |
| servers    | None                                 |
| status     | not_run                              |
| updated_at | None                                 |
+------------+--------------------------------------+
```

Step 3 The cisco vim diskmgmt list command is used to monitor a currently running task, and the completed tasks. The list command can filter based on the role. Using 'all' command lists all tests that are in the database.

```
# ciscovim diskmgmt list check-disks control
+--------------------------------------+-------------+---------+----------+----------------------------+
| ID                                   | Action      | Role    | Status   | Created
   |
+--------------------------------------+-------------+---------+----------+----------------------------+
| 861d4d73-ffee-40bf-9348-13afc697ee3d | check-disks | control | Complete | 2018-03-05 14:44:47+00:00
 |
| 0c6d27c8-bdac-493b-817e-1ea8640dae57 | check-disks | control | Running  | 2018-03-07 21:12:20+00:00
 |
+--------------------------------------+-------------+---------+----------+----------------------------+
[root@F24-Michigan ~]# ciscovim diskmgmt list check-disks compute
+--------------------------------------+-------------+---------+----------+----------------------------+
| ID                                   | Action      | Role    | Status   | Created
   |
+--------------------------------------+-------------+---------+----------+----------------------------+
| 0be7a55a-37fe-43a1-a975-cbf93ac78893 | check-disks | compute | Complete | 2018-03-05 14:45:45+00:00
 |
+--------------------------------------+-------------+---------+----------+----------------------------+
[root@F24-Michigan ~]# ciscovim diskmgmt list check-disks all
+--------------------------------------+-------------+---------+----------+----------------------------+
| ID                                   | Action      | Role    | Status   | Created
   |
+--------------------------------------+-------------+---------+----------+----------------------------+
| cdfd18c1-6346-47a2-b0f5-661305b5d160 | check-disks | all     | Complete | 2018-03-05 14:43:50+00:00
 |
| 861d4d73-ffee-40bf-9348-13afc697ee3d | check-disks | control | Complete | 2018-03-05 14:44:47+00:00
 |
| 0be7a55a-37fe-43a1-a975-cbf93ac78893 | check-disks | compute | Complete | 2018-03-05 14:45:45+00:00
 |
| 0c6d27c8-bdac-493b-817e-1ea8640dae57 | check-disks | control | Complete | 2018-03-07 21:12:20+00:00
 |
+--------------------------------------+-------------+---------+----------+----------------------------+
```

Step 4 Run the following command to show the detailed results of a diskmgmt check-disks operation:

```
# ciscovim diskmgmt show check-disks control --id 0c6d27c8-bdac-493b-817e-1ea8640dae57
+--------------------------+--------------------+----------------------------+---------+-------------------------+
| Message                  | Host               | Role                       | Server  | State
                  |
+--------------------------+--------------------+----------------------------+---------+-------------------------+
| Raid Health Status       | f24-michigan-micro-1 | block_storage control compute | 7.7.7.7 |
Optimal                  |
|                          | f24-michigan-micro-2 | block_storage control compute | 7.7.7.6 |
Optimal                  |
|                          | f24-michigan-micro-3 | block_storage control compute | 7.7.7.5 |
Optimal                  |
|                          |                      |                               |         |
                    |
| VD Health Status         | f24-michigan-micro-1 | block_storage control compute | 7.7.7.7 |
Optimal                  |
|                          | f24-michigan-micro-2 | block_storage control compute | 7.7.7.6 |
Optimal                  |
|                          | f24-michigan-micro-3 | block_storage control compute | 7.7.7.5 |
Optimal                  |
|                          |                      |                               |         |
                    |
| RAID Level and Type      | f24-michigan-micro-1 | block_storage control compute | 7.7.7.7 | Type
 - HW; Level - RAID1 |
|                          | f24-michigan-micro-2 | block_storage control compute | 7.7.7.6 | Type
 - HW; Level - RAID1 |
|                          | f24-michigan-micro-3 | block_storage control compute | 7.7.7.5 | Type
 - HW; Level - RAID1 |
|                          |                      |                               |         |
                    |
| Number of Physical Disks | f24-michigan-micro-1 | block_storage control compute | 7.7.7.7 | 8
                    |
|                          | f24-michigan-micro-2 | block_storage control compute | 7.7.7.6 | 8
                    |
|                          | f24-michigan-micro-3 | block_storage control compute | 7.7.7.5 | 8
                    |
|                          |                      |                               |         |
                    |
| Number of Virtual Disks  | f24-michigan-micro-1 | block_storage control compute | 7.7.7.7 | 1
                    |
|                          | f24-michigan-micro-2 | block_storage control compute | 7.7.7.6 | 1
                    |
|                          | f24-michigan-micro-3 | block_storage control compute | 7.7.7.5 | 1
                    |
|                          |                      |                               |         |
                    |
| Boot Drive Disk Media-Type | f24-michigan-micro-1 | block_storage control compute | 7.7.7.7 | HDD
                    |
|                          | f24-michigan-micro-2 | block_storage control compute | 7.7.7.6 | HDD
                    |
|                          | f24-michigan-micro-3 | block_storage control compute | 7.7.7.5 | SSD
                    |
+--------------------------+--------------------+----------------------------+---------+-------------------------+
State Keys:
DHS-Dedicated Hot Spare|UGood-Unconfigured Good|GHS-Global Hotspare
UBad-Unconfigured Bad|Onln-Online|Offln-Offline
Rbld-Rebuilding|JBOD-Just a Bunch Of Disks
```

Step 5 Run the following command to delete the diskmgmt check-disks:

```
Delete a diskmgmt check-disks result:
```

Note We recommand you to delete the tests which are not in use.

OSD-Maintenance Tool

In VIM you can use the osd-maintenance tool to check the status of all OSDs that are present in running and operational block storage nodes. OSD maintenance tool gives you the detailed information about the status of the OSDs - if they are Up or Down, in addition to what HDD corresponds to which OSD, including the slot number and server hostname.

- If it is down OSD is discovered after check_osds is performed, run the cluster recovery and recheck.
- If still down, wait 30 minutes before attempting replace - time for ceph-mon to sync.
- Physically remove and insert a new disk before attempting replace.
- For smooth operation, wipe out disk before attempting replace operations.
- Need a dedicated journal SSD for each storage server where osdmgmt is attempted.
- Only allowed to replace one OSD at a time. Space out each replace OSD by 30 minutes - time for ceph-mon to sync.
- Call TAC if any issue is hit. Do not reattempt.

To check the status of the osdmgmt tool log in to the management node and run the ciscovim command with the osdmgmt option. The osdmgmt user interface allows you to create, list, show, and delete workflow.

- Use ‘ciscovim osdmgmt create …’ command to initiate a check and replace OSD operation
- Use ‘ciscovim osdmgmt list …’ command to view summary and status of current OSD operations
- Use ‘ciscovim osdmgmt show … --id <ID>’ command to view detail OSD operation results
- Use ‘ciscovim osdmgmt delete … --id <ID>’ command to delete the results.

Examples of usage of this tool:

Step 1 Run the Help command to see all the option:

```
# ciscovim help osdmgmt
usage: ciscovim osdmgmt [--server <node1,node2,...>] [--detail] [--id <id>]
                        [--osd <osd_name>] [--locator {on,off}]
                        [--json-display] [-y]
                        create|delete|list|show check-osds|replace-osd

OSD maintenance helper

Positional arguments:
  create|delete|list|show     The control command to perform
  check-osds|replace-osd      The identity of the task/action

Optional arguments:
  --server <node1,node2,...>  List of specific block_storage hostnames
  --detail                    Display full OSD details
```

```
  --id <id>                   ID used to identify specific item to
                              show/delete.
  --osd <osd_name>            Name of down OSD to replace. Eg. 'osd.xx'
  --locator {on,off}          Turn on|off locator LED for server with bad OSDs
                              and for the physical drives.
  --json-display              Show output will be in JSON format.
  -y, --yes                   Yes option to perform the action

--+------------+----------+-----------+--------------------------+-----------+
```

Step 2 To check the osds run the following command:

```
# ciscovim osdmgmt create check-osds
+------------+--------------------------------------+
| Field      | Value                                |
+------------+--------------------------------------+
| action     | check-osds                           |
| command    | create                               |
| created_at | 2018-03-08T21:11:13.611786+00:00     |
| id         | 5fd4f9b5-786a-4a21-a70f-bffac70a3f3f |
| locator    | False                                |
| osd        | None                                 |
| result     |                                      |
| servers    | None                                 |
| status     | not_run                              |
| updated_at | None                                 |
+------------+--------------------------------------+
```

Step 3 Monitor the osdmgmt check operations using te list command. Cisco Vim Osd mgmt list commands are used to monitor the currently running test. It also helps you to view the tests that are run/ completed.

```
# ciscovim osdmgmt list check-osds
+--------------------------------------+------------+----------+---------------------------+
| ID                                   | Action     | Status   | Created                   |
+--------------------------------------+------------+----------+---------------------------+
| 5fd4f9b5-786a-4a21-a70f-bffac70a3f3f | check-osds | Complete | 2018-03-08 21:11:13+00:00 |
| 4efd0be8-a76c-4bc3-89ce-142de458d844 | check-osds | Complete | 2018-03-08 21:31:01+00:00 |
+--------------------------------------+------------+----------+---------------------------+
```

Step 4 To show the detailed results of a osdmgmt check-osds operation run the following command:

```
# ciscovim osdmgmt show check-osds  --id 5fd4f9b5-786a-4a21-a70f-bffac70a3f3f
+--------------------+----------------------+--------------------------------+---------+----------+
| Message            | Host                 | Role                           | Server  | State    |
+--------------------+----------------------+--------------------------------+---------+----------+
| Overall OSD Status | f24-michigan-micro-3 | block_storage control compute | 7.7.7.5 | Optimal |
|                    | f24-michigan-micro-1 | block_storage control compute | 7.7.7.7 | Optimal |
|                    | f24-michigan-micro-2 | block_storage control compute | 7.7.7.6 | Optimal |
|                    |                      |                                |         |          |
| Number of OSDs     | f24-michigan-micro-3 | block_storage control compute | 7.7.7.5 | 5        |
|                    | f24-michigan-micro-1 | block_storage control compute | 7.7.7.7 | 5        |
|                    | f24-michigan-micro-2 | block_storage control compute | 7.7.7.6 | 5        |
+--------------------+----------------------+--------------------------------+---------+----------+
+----------------------+--------+--------+----+-----------+-----------+--------------------------+-----------+
| Host                 | OSDs   | Status | ID | HDD Slot  | Path      | Mount                    |
 Journal   |
+----------------------+--------+--------+----+-----------+-----------+--------------------------+-----------+
| f24-michigan-micro-3 | osd.0  | up     | 0  |  4 (JBOD) | /dev/sda1 | /var/lib/ceph/osd/ceph-0  |
 /dev/sdf1 |
|                      | osd.1  | up     | 1  |  5 (JBOD) | /dev/sdb1 | /var/lib/ceph/osd/ceph-1  |
 /dev/sdf2 |
|                      | osd.3  | up     | 3  |  7 (JBOD) | /dev/sdc1 | /var/lib/ceph/osd/ceph-3  |
 /dev/sdf3 |
```

```
|                       | osd.5  | up     | 5  |  8 (JBOD) | /dev/sdd1 | /var/lib/ceph/osd/ceph-5  |
 /dev/sdf4 |
|                       | osd.6  | up     | 6  |  6 (JBOD) | /dev/sde1 | /var/lib/ceph/osd/ceph-6  |
 /dev/sdf5 |
|                       |        |        |    |           |           |                           |
           |
| f24-michigan-micro-1 | osd.2  | up     | 2  |  5 (JBOD) | /dev/sda1 | /var/lib/ceph/osd/ceph-2  |
 /dev/sdf1 |
|                       | osd.7  | up     | 7  |  7 (JBOD) | /dev/sdb1 | /var/lib/ceph/osd/ceph-7  |
 /dev/sdf2 |
|                       | osd.9  | up     | 9  |  8 (JBOD) | /dev/sdc1 | /var/lib/ceph/osd/ceph-9  |
 /dev/sdf3 |
|                       | osd.11 | up     | 11 |  6 (JBOD) | /dev/sdd1 | /var/lib/ceph/osd/ceph-11 |
 /dev/sdf4 |
|                       | osd.13 | up     | 13 |  4 (JBOD) | /dev/sde1 | /var/lib/ceph/osd/ceph-13 |
 /dev/sdf5 |
|                       |        |        |    |           |           |                           |
           |
| f24-michigan-micro-2 | osd.4  | up     | 4  |  8 (JBOD) | /dev/sda1 | /var/lib/ceph/osd/ceph-4  |
 /dev/sdf1 |
|                       | osd.8  | up     | 8  |  5 (JBOD) | /dev/sdb1 | /var/lib/ceph/osd/ceph-8  |
 /dev/sdf2 |
|                       | osd.10 | up     | 10 |  4 (JBOD) | /dev/sdc1 | /var/lib/ceph/osd/ceph-10 |
 /dev/sdf3 |
|                       | osd.12 | up     | 12 |  6 (JBOD) | /dev/sdd1 | /var/lib/ceph/osd/ceph-12 |
 /dev/sdf4 |
|                       | osd.14 | up     | 14 |  7 (JBOD) | /dev/sde1 | /var/lib/ceph/osd/ceph-14 |
 /dev/sdf5 |
+----------------------+--------+--------+----+-----------+-----------+---------------------------+-----------+
```

Step 5 To delete the check-disk osds run the following command:

```
# ciscovim osdmgmt delete check-osds --id 5fd4f9b5-786a-4a21-a70f-bffac70a3f3f

Perform the action. Continue (Y/N)Y
Delete of UUID 5fd4f9b5-786a-4a21-a70f-bffac70a3f3f Successful

[root@F24-Michigan ~]# ciscovim osdmgmt list check-osds
+--------------------------------------+------------+----------+----------------------------+
| ID                                   | Action     | Status   | Created                    |
+--------------------------------------+------------+----------+----------------------------+
| 4efd0be8-a76c-4bc3-89ce-142de458d844 | check-osds | Complete | 2018-03-08 21:31:01+00:00 |
+--------------------------------------+------------+----------+----------------------------+
```

Utility to Resolve Cisco VIM Hardware Validation Failures

The Cisco VIM Hardware Validation utility tool is used to perform hardware validation during the installation of UCS C-series servers. It captures the user and environmental hardware validation errors that occur during the installation process. The tool enables you to fix these errors that are based on the inputs you provide at the Command Line Interface (CLI). It validates the updated configurations to verify if the changes are applied properly. After the error is resolved, you can resume the installation from the point of failure.

The ciscovim hardware-mgmt user interface allows you to test the job validate orresolve-failures(create), list, show, and delete workflow

Hardware-mgmt user workflow:

1. Use “ciscovim hardware-mgmt validate …” command to initiate a validation.
2. Use “ciscovim hardware-mgmt list …” command to view summary/status of current test jobs.
3. Use “ciscovim hardware-mgmt show … --id <ID>” command to view detail test results
4. Use “ciscovim hardware-mgmt delete … --id <ID>” to delete test results.

A database of results is maintained so that the user can keep the results of multiple hardware-mgmt operations and view them at any time.

Note You cannot use the utility for the following tasks:

- Configuring BIOS settings for the B-series pods.
- Upgrading or changing the firmware version.
- Resolving hardware failures other than lom, hba, flexflash, pcie_slot, power, and vnic_pxe_boot.

Command Usage

To capture the list of failures that can be resolved by using the utility, go to the install directory and execute the help command:

 # cd <installer-id>/clouddeploy

 # python hw_validations.py -help .

The following shows the output of the help command.

```
usage: hw_validations.py [-h] [--resolve-failures RESOLVE_FAILURES]
[--validate VALIDATE_OF] [-y] [--host HOSTS]
[--file SETUP_FILE_LOCATION]
UCS Hardware Validations
optional arguments:
-h, --help show this help message and exit
--resolve-failures RESOLVE_FAILURES, -rf RESOLVE_FAILURES
                      all - Fix all the failures.
                      lom - Fix LOM port(s) status failures.
                      hba - Fix HBA port status failures.
                      flexflash - Fix Flexflash failures.
                      pcie_slot - Fix PCIe slot status failures.
                      power - Fix Power failures.
                      vnic_pxe_boot - Fix Vnic PXE_Boot statusfailures
-y, -yes
--host HOSTS Comma separated list of hostnames
--file SETUP_FILE_LOCATION, -f SETUP_FILE_LOCATION
            Provide a valid 'setup_data.yaml' file
```

Command Syntax

hw_validations.py [-h] [--resolve-failures RESOLVE_FAILURES] [--validate VALIDATE_OF] [-y] [--host HOSTS] [--file SETUP_FILE_LOCATION]

The following table provides the description of the parameters of the command.

Optional	Description
[-h], --help	Provides detailed information about the command.
[--resolve-failures RESOLVE_FAILURES], -rf RESOLVE_FAILURES	Enables you to specify the failure that you want to resolve. The optional arguments are as follows:
[-y]	Yes
[--host HOSTS]	Enables you to specify the hostname of the server for which you want to resolve failures. You cannot specify the IP address or CIMC IP address of servers as arguments. You can specify a list of hostnames as comma-separated arguments. If the -host option is not specified, the failures of all the servers that are specified in the setup_data.yaml file are resolved.
[--file SETUP_FILE_LOCATION] [-f SETUP_FILE_LOCATION]	Enables you to specify the name of a setup_data.yaml file.

Examples of Command Usage

The following table provides the commonly used commands along with their examples.

Purpose	Syntax	Example
To resolve all failures.	python hw_validations.py --resolve-failures all -y	python hw_validations.py --resolve-failures all -y
To simultaneously resolve one or more failures.	python hw_validations.py --resolve-failures <failure-1>,<failure-2> -y	To resolve the lom and hba status failures: python hw_validations.py --resolve-failures lom,hba -y
To resolve the errors by using the setup_data.yaml file.	python hw_validations.py --resolve-failures <failure-1>,<failure-2> -y --file <location-of-yaml file>	To resolve the LOM status failures by using ~/save/setup_data.yaml file: python hw_validations.py --resolve-failures lom,hba -y --file ~/save/setup_data.yaml
To resolve failures on a particular server as specified in the setup_data.yaml file by using the *--host* option.	*python hw_validations.py --resolve-failures <failure-1> -y --host <name-of-host-server-1>,<name-of-host-server-2>*	To resolve the PCIe slot failures on hiccup-controller-1 server as specified in the setup_data.yaml: *python hw_validations.py --resolve-failures pcie_slot -y --host hiccup-controller-1*

Cisco VIM Client Debug Option

The --debug option enables you to get verbose logging on the ciscovim client console. You can use verbose logging to troubleshoot issues with the ciscovim client.

The debug option has the following parts:

- Curl Command: Curl command can be used for debugging. It can be executed standalone. Curl Command also displays the REST API Endpoint and the Request Payload.
- Response of REST API

Examples of Using debug Option to get list of passwords

```
# ciscovim --debug list-password-keys
2018-05-28 22:13:21,945 DEBUG [ciscovimclient.common.httpclient][MainThread] curl -i -X GET
 -H 'Content-Type: application/json' -H 'Authorization: ****' -H 'Accept: application/json'
 -H 'User-Agent: python-ciscovimclient' --cacert /var/www/mercury/mercury-ca.crt
https://172.31.231.17:8445/secrets
2018-05-28 22:13:21,972 DEBUG [ciscovimclient.common.httpclient][MainThread]
HTTP/1.1 200 OK
content-length: 1284
x-xss-protection: 1
x-content-type-options: nosniff
strict-transport-security: max-age=31536000
server: WSGIServer/0.1 Python/2.7.5
cache-control: no-cache, no-store, must-revalidate, max-age=0
date: Tue, 29 May 2018 05:13:21 GMT
x-frame-options: SAMEORIGIN
content-type: application/json; charset=UTF-8

{u'HEAT_KEYSTONE_PASSWORD': '****', u'CINDER_KEYSTONE_PASSWORD': '****',
u'METADATA_PROXY_SHARED_SECRET': '****', u'WSREP_PASSWORD': '****', u'ETCD_ROOT_PASSWORD':
 '****', u'HEAT_DB_PASSWORD': '****', u'CINDER_DB_PASSWORD': '****', u'KEYSTONE_DB_PASSWORD':
 '****', u'NOVA_DB_PASSWORD': '****', u'GLANCE_KEYSTONE_PASSWORD': '****',
u'CLOUDPULSE_KEYSTONE_PASSWORD': '****', u'VPP_ETCD_PASSWORD': '****', u'COBBLER_PASSWORD':
 '****', u'DB_ROOT_PASSWORD': '****', u'NEUTRON_KEYSTONE_PASSWORD': '****',
u'HEAT_STACK_DOMAIN_ADMIN_PASSWORD': '****', u'KIBANA_PASSWORD': '****',
u'IRONIC_KEYSTONE_PASSWORD': '****', u'ADMIN_USER_PASSWORD': '****', u'HAPROXY_PASSWORD':
'****', u'NEUTRON_DB_PASSWORD': '****', u'IRONIC_DB_PASSWORD': '****', u'GLANCE_DB_PASSWORD':
 '****', u'RABBITMQ_ERLANG_COOKIE': '****', u'NOVA_KEYSTONE_PASSWORD': '****',
u'CPULSE_DB_PASSWORD': '****', u'HORIZON_SECRET_KEY': '****', u'RABBITMQ_PASSWORD': '****'}

+----------------------------------+
| Password Keys                    |
+----------------------------------+
| ADMIN_USER_PASSWORD              |
| CINDER_DB_PASSWORD               |
| CINDER_KEYSTONE_PASSWORD         |
| CLOUDPULSE_KEYSTONE_PASSWORD     |
| COBBLER_PASSWORD                 |
| CPULSE_DB_PASSWORD               |
| DB_ROOT_PASSWORD                 |
| ETCD_ROOT_PASSWORD               |
| GLANCE_DB_PASSWORD               |
| GLANCE_KEYSTONE_PASSWORD         |
| HAPROXY_PASSWORD                 |
| HEAT_DB_PASSWORD                 |
| HEAT_KEYSTONE_PASSWORD           |
| HEAT_STACK_DOMAIN_ADMIN_PASSWORD |
| HORIZON_SECRET_KEY               |
```

```
| IRONIC_DB_PASSWORD                |
| IRONIC_KEYSTONE_PASSWORD          |
| KEYSTONE_DB_PASSWORD              |
| KIBANA_PASSWORD                   |
| METADATA_PROXY_SHARED_SECRET      |
| NEUTRON_DB_PASSWORD               |
| NEUTRON_KEYSTONE_PASSWORD         |
| NOVA_DB_PASSWORD                  |
| NOVA_KEYSTONE_PASSWORD            |
| RABBITMQ_ERLANG_COOKIE            |
| RABBITMQ_PASSWORD                 |
| VPP_ETCD_PASSWORD                 |
| WSREP_PASSWORD                    |
+-----------------------------------+
```

Examples of Using debug option to get list of nodes

```
# ciscovim --debug list-nodes
2018-05-28 22:13:31,572 DEBUG [ciscovimclient.common.httpclient][MainThread] curl -i -X GET
 -H 'Content-Type: application/json' -H 'Authorization: ****' -H 'Accept: application/json'
 -H 'User-Agent: python-ciscovimclient' --cacert /var/www/mercury/mercury-ca.crt
https://172.31.231.17:8445/nodes
2018-05-28 22:13:31,599 DEBUG [ciscovimclient.common.httpclient][MainThread]
HTTP/1.1 200 OK
content-length: 2339
x-xss-protection: 1
x-content-type-options: nosniff
strict-transport-security: max-age=31536000
server: WSGIServer/0.1 Python/2.7.5
cache-control: no-cache, no-store, must-revalidate, max-age=0
date: Tue, 29 May 2018 05:13:31 GMT
x-frame-options: SAMEORIGIN
content-type: application/json; charset=UTF-8

{u'nodes': {u'status': u'Active', u'uuid': u'6b1ea6ee-b15b-41ca-9d79-3bb9ec0002bc',
u'setupdata': u'fe78b5f9-5a46-447c-9317-2bf7362c1e81', u'node_data': {u'rack_info':
{u'rack_id': u'RackD'}, u'cimc_info': {u'cimc_ip': u'172.29.172.81'}, u'management_ip':
u'21.0.0.10'}, u'updated_at': u'2018-05-25T11:14:46+00:00', u'reboot_required': u'No',
u'mtype': u'control', u'install': u'372aa3c1-1ab0-4dd0-a8a8-1853a085133c', u'power_status':
 u'PowerOnSuccess', u'install_logs':
u'https://172.31.231.17:8008//edd3975c-8b7c-4d3c-93de-a033ae10a6b6', u'created_at':
u'2018-05-21T13:25:50+00:00', u'name': u'gg34-2'}}

+-----------+--------+---------+---------------+
| Node Name | Status |   Type  | Management IP |
+-----------+--------+---------+---------------+
|   gg34-1  | Active | control |   21.0.0.12  |
|   gg34-2  | Active | control |   21.0.0.10  |
|   gg34-3  | Active | control |   21.0.0.11  |
|   gg34-4  | Active | compute |   21.0.0.13  |
+-----------+--------+---------+---------------+
```

Example of Getting Response from REST API using Curl Commands

```
Get the REST API Password.
# cat /opt/cisco/ui_config.json
{
"Kibana-Url": "http://172.31.231.17:5601",
"RestAPI-Url": "https://172.31.231.17:8445",
"RestAPI-Username": "admin",
"RestAPI-Password": "********************",
"RestDB-Password": "********************",
"BuildNodeIP": "172.31.231.17"
}
```

```
Form the Curl Command.
curl -k -u <RestAPI-Username>:<RestAPI-Password>  <RestAPI-Url>/<Endpoint>
E.g. To get Nodes Info of Cloud
curl -k -u admin:**** http://172.31.231.17:5601/v1/nodes
```

Examples of Response of REST APIs

```
API "/"

# curl -k -u admin:**** https://172.31.231.17:8445/

{"default_version": {"id": "v1", "links": [{"href": "http://127.0.0.1:8083/v1/", "rel":
"self"}]}, "versions": [{"id": "v1", "links": [{"href": "http://127.0.0.1:8083/v1/", "rel":
 "self"}]}], "name": "Virtualized Infrastructure Manager Rest API", "description":
"Virtualized Infrastructure Manager Rest API is used to invoke installer from API."}


API "/v1/setupdata/"

# curl -k -u admin:**** https://172.31.231.17:8445/v1/setupdata/

{"setupdatas": [. . .]}


API "/v1/nodes"

# curl -k -u admin:**** https://172.31.231.17:8445/v1/nodes

{"nodes": [{"status": "Active", "uuid": "0adabc97-f284-425b-ac63-2d336819fbaf", "setupdata":
 "fe78b5f9-5a46-447c-9317-2bf7362c1e81", "node_data": "{\"rack_info\": {\"rack_id\":
\"RackC\"}, \"cimc_info\": {\"cimc_ip\": \"172.29.172.75\"}, \"management_ip\":
\"21.0.0.13\"}", "updated_at": "2018-05-21T15:11:05+00:00", "reboot_required": "No", "mtype":
 "compute", "install": "372aa3c1-1ab0-4dd0-a8a8-1853a085133c", "power_status":
"PowerOnSuccess", "install_logs":
"https://172.31.231.17:8008//edd3975c-8b7c-4d3c-93de-a033ae10a6b6", "created_at":
"2018-05-21T13:25:50+00:00", "name": "gg34-4"}, . . . . ]}


API "/v1/secrets"

# curl -k -u admin:**** https://172.31.231.17:8445/v1/secrets

{"HEAT_KEYSTONE_PASSWORD": "5oNff4jWsvAwnWk1", "CINDER_KEYSTONE_PASSWORD": "Hq4i6S5CnfQe7Z2W",
 "RABBITMQ_ERLANG_COOKIE": "XRMHBQHTLVJSVWDFKJUX", "METADATA_PROXY_SHARED_SECRET":
"XNzrhosqW4rwiz7c", "WSREP_PASSWORD": "z1oQqhKd1fXDxJTV", "ETCD_ROOT_PASSWORD":
"LMLC8gvi1IA3KiIc", "HEAT_DB_PASSWORD": "J8zt8ldMvdtJxAtG", "CINDER_DB_PASSWORD":
"BVX3y2280DSx2JkY", "KEYSTONE_DB_PASSWORD": "55fVNzxR1VxCNOdh", "NOVA_DB_PASSWORD":
"Rk1MK1OIJgsjGZal", "IRONIC_KEYSTONE_PASSWORD": "9tYZgIw6SZERZ1dZ", "ADMIN_USER_PASSWORD":
 "DjDQrk4QT7pgHy94", "GLANCE_KEYSTONE_PASSWORD": "w4REb8uhrHquCfRm", "HAPROXY_PASSWORD":
"oB0v7VJoo2IfB8OW", "CLOUDPULSE_KEYSTONE_PASSWORD": "q6QVvxBQhrqv6Zhx", "NEUTRON_DB_PASSWORD":
 "FZVMWgApcZR4us5q", "IRONIC_DB_PASSWORD": "dq3Udmu95DWyX1jy", "GLANCE_DB_PASSWORD":
"O7vQ2emuPDrrvD4x", "KIBANA_PASSWORD": "azHHhP4ewxpZVwcg", "VPP_ETCD_PASSWORD":
"NLyIAvECMW2qI7Bp", "NOVA_KEYSTONE_PASSWORD": "JUfMNGz0BZG7JwXV", "NEUTRON_KEYSTONE_PASSWORD":
 "QQ0lo8Q87BjFoAYQ", "CPULSE_DB_PASSWORD": "DaFthNtpX2RvwTSs", "COBBLER_PASSWORD":
"XoIJ9mbWcmVyzvvN", "HORIZON_SECRET_KEY":
"NHkA0qwHIWUSwhPZowJ8Ge3RyRd6oM8XjOT8PHnZdckxgm3kbb1MSltsw0TAQJnx", "DB_ROOT_PASSWORD":
"seqh5DRIKP6ZsKJ8", "HEAT_STACK_DOMAIN_ADMIN_PASSWORD": "Vu6LexEadAxscsvY",
"RABBITMQ_PASSWORD": "LBoYoxuvGsMsl1TX"}

API "/v1/nodes/mgmt._node"

# curl -k -u admin:**** https://172.31.231.17:8445/v1/nodes/mgmt_node
```

```
{"api_ip": "172.31.231.17", "mgmt_ip": "21.0.0.2"}
```