NVMesh Shell/CLI Tool

- Version
- Introduction
- Feature/functionality map-out and feature/functionality availability snapshot
- Supported Environments
- Installation requirements
- Installation Steps - no virtual environment
- Installation requirements - virtual environment
- Installation Steps - virtual environment
- Using it the first time
- Available commands, features, and its usage
  - help | nvmesh help
  - define | nvmesh define
  - load | nvmesh load
    - quick and basic nvmesh cli health check example output
  - show | nvmesh show
  - add | nvmesh add
  - update | nvmesh update
  - attach | nvmesh attach
  - detach | nvmesh detach
  - runcmd | nvmesh runcmd
  - evict | nvmesh evict
  - check | nvmesh check
  - stop | nvmesh stop
  - start | nvmesh start
  - restart | nvmesh restart
  - testssh | nvmesh testssh
- Advanced Usage Examples
  - Change the NVMesh yum repository link throughout the whole NVMesh cluster
  - Delete all the volumes in an NVMesh cluster
  - Delete all the volumes in an NVMesh cluster where the volume name contains "test"
  - Test the SSH connectivity to only the NVMesh target servers throughout the NVMesh cluster
  - Check the MTU settings for all the network interfaces throughout your NVMesh cluster

Version

This documentation is for NVMesh CLI tool version v1.3-49

Introduction

The nvmesh-shell cli tool is developed and maintained by the solutions engineering/field engineering team providing a regular cli tool/program which can be used to write OS shell scripts, e.g. bash shell scripts, or one-liners. Additionally, it also offers an interactive shell interface by itself. All subcommands are available any way you decide to use it.
The tool is a shim layer on top of the product provided RESTful API and terminal command line tools plus providing a facility to run shell commands providing the user with an interface for day-to-day management and provisioning activities with homogeneous semantics.

> **Please Note:**
> As it is a shim/orchestration layer and tool unifying various tools and API provided by the product and the operating system, it will not add any new product features nor bridge other gaps. If you are missing product features and functionality in the actual product, please reach out to the product team and discuss any product requirements you may have with them.

Feature/functionality map-out and feature/functionality availability snapshot

<br/>

Supported Environments

Linux and MacOS running Python version 2.

> Python minimum requirement is 2.7.5. and v3 is not supported currently.

Installation requirements

You need a working pip environment before attempting to install the tool.
More information and how to install pip can be found here: Installing pip

Installation Steps - no virtual environment

1. `mkdir` a new directory or change into the directory where you want to save the nvmesh-shell source code for the installation
2. Download here: nvmesh-shell source and copy the source into the directory to be used for the install or run: `git clone https://github.com/excelero/nvmesh-shell` from within that directory
3. Change into the 'nvmesh-shell/' source directory
4. Run `pip install .`

Installation requirements - virtual environment

In addition to pip, you also need the python virtualenv package installed and working properly.
More details on how to install and use python virtualenv can be found here: Python Virtualenv

Installation Steps - virtual environment

1. Create a new virtual environment
2. Change into the new virtual environment and execute `source bin/activate` to activate this environment
3. Run: `git clone https://github.com/excelero/nvmesh-shell` from within that directory
4. Change into the 'nvmesh-shell/' source directory
5. Run `pip install .`

Using it the first time

Initially, the tool doesn't know anything about the NVMesh environment and no credentials are set. The tool requires the NVMesh management / API login information (administrative account) and if there is no pre-shared SSH key set up with all the involved hosts, servers and clients, the SSH credentials is required for certain functionality.
You need to run the following commands to setup your environment:

```
# nvmesh-shell define sshuser
```

```
# nvmesh-shell define apiuser
```

```
# nvmesh-shell define manager
```

The tool will ask will ask for the SSH credentials where you can choose between sudo and root.

If preshared keys are set up throughout, please leave the password prompt empty and just hit enter. There is no need to provide a password if preshared keys was set up.

Use sudo for SSH

```
nvmesh # define sshuser
Please provide the user name to be used for SSH connectivity: <user name
to be used>
Please provide the SSH password: <the user's password if no pre-shared
key authentication>
Do you require sudo for SSH remote command execution? [Yes|No] :
```

The NVMesh management api user setup:

```
nvmesh # define apiuser
Please provide a administrative API user name: <the api user name to be
used>
Please provide the API password:<the api users password to be used>
```

The NVMesh management server setup. The CLI supports up to three NVMesh manager servers if configured in HA mode.

```
nvmesh # define manager
Provide a space separated list, min. one, of the NVMesh manager server
name/s:<server1 server2 server3>
```

The API user and password, and the SSH user and password are stored under the users home directory.
Passwords are stored encoded and obfuscated as additional protection.
Also, the NVMesh management server information is stored in the users home directory.

There are other files stored in the users home directory in addition to the credentials, please see the details as below:

~/.nvmesh_api_secrets - stores the API username and password
~/.nvmesh_manager - stores the NVMesh management server name
~/.nvmesh_shell_history - stores the NVMesh shell cli tool command history
~/.nvmesh_shell_secrets - stores the SSH user information
~/.nvmesh_shell_sudo - stores the SSH sudo requirements [True | False]
~/.nvmeshcli.log - traces and keeps the API activity and payload send to and received back from the NVMesh management API and also logs the SSH activities

Available commands, features, and its usage

help | nvmesh help

```
nvmesh # help

Documented commands (type help <topic>):

NVMesh Resource Management
==========================
add     check   delete  evict   restart  show   stop     update
attach  define  detach  format  runcmd   start  testssh

Other
=====
alias  edit  help  history  load  quit  set  shell  shortcuts  unalias
```

define | nvmesh define

```
Usage: define [-h] {manager,sshuser,apiuser}

The 'define' sub-command defines/sets the shell runtime variables as
NVMesh
management servers and user credentials to be used. E.g. 'define
apiuser' will
set the NVMesh API user name to be used for all the operations involving
the
API

positional arguments:
  {manager,sshuser,apiuser}
                        Specify the NVMesh shell runtime variable you
want to
                        define.

optional arguments:
  -h, --help            show this help message and exit
```

load | nvmesh load

```
Runs commands in nvmesh shell/cli script file that is encoded as either
ASCII or UTF-8 text.

    Usage:  load <file_path>

    * file_path - a file path pointing to a nvmesh shell/script

Script should contain one command per line, just like command would be
typed in console.
Nvmesh Shell/CLI example script to perform a basic clsuter health check:
show cluster
show manager
show drivemodel
show drive -d
runcmd cluster -c "cat /var/log/NVMesh/toma_leader_name"
show target -d
show client
show volume -d -l
```

quick and basic nvmesh cli health check example output

```
nvmesh # load nvmesh_shell_sample_script
[Cluster high level info]
```

```
--------------------------------------------------------------------------
-----------------------------------------------------------------
| Total Servers | Offline Servers | Total Clients | Offline Clients |
Volumes                      | Total Capacity | Available Space |
--------------------------------------------------------------------------
-----------------------------------------------------------------
|             4 |               0 |             4 |               0 | 1
Striped & Mirrored RAID-10 | 20.96 TiB      | 18.96 TiB       |
--------------------------------------------------------------------------
-----------------------------------------------------------------

[Manager info]
--------------------------------------------------------------------------
---------------------------------------------
| Manager                     | IP        | Current Connection | Use SSL
| Port | Outbound Socket | Inbound Socket |
--------------------------------------------------------------------------
---------------------------------------------
| uslab-22.uslab.excelero.com | 10.0.1.22 |                    |         |
4001 | n/a             | n/a            |
| uslab-23.uslab.excelero.com | 10.0.1.23 |                     |
| 4001 | connected       | connected      |
| uslab-24.uslab.excelero.com | 10.0.1.24 |                     |
| 4001 | connected       | connected      |
--------------------------------------------------------------------------
---------------------------------------------

[Drive model and count per model]
---------------------------------------
| Drive Model               | Drives |
---------------------------------------
| MTFDHAL800MCE-1AN1ZABYY   |      5 |
| MTFDHAL2T4MCF-1AN1ZABYY   |      1 |
| SDLC2LLR-038T-3BA2        |      1 |
| Micron_9200_MTFDHAL3T2TCU |      4 |
---------------------------------------

[Drive details]
--------------------------------------------------------------------------
--------------------------------------------------------------------------
---------------------
| Vendor | Model                                   | Drive ID       |
Size       | Status | BS        | Wear | Target                      |
Numa | PCI root | SubQ |
--------------------------------------------------------------------------
--------------------------------------------------------------------------
---------------------
| None   | SDLC2LLR-038T-3BA2_____________________ | A0342961.1     |
3.49 TiB   |        | 512 bytes | 0 %  | uslab-21.uslab.excelero.com |
1 |        1 |   64 |
| Micron | MTFDHAL2T4MCF-1AN1ZABYY                 | P61015031037.1 |
2.18 TiB   |       | 512 bytes | 0 %  | uslab-23.uslab.excelero.com |
1 |        1 |  128 |
| Micron | MTFDHAL800MCE-1AN1ZABYY                 | P60913037990.1 |
745.21 GiB |       | 512 bytes | 0 %  | uslab-22.uslab.excelero.com |
1 |        1 |  128 |
| Micron | MTFDHAL800MCE-1AN1ZABYY                 | P60913038002.1 |
745.21 GiB |       | 512 bytes | 0 %  | uslab-22.uslab.excelero.com |
1 |        1 |  128 |
| Micron | MTFDHAL800MCE-1AN1ZABYY                 | P60913038027.1 |
745.21 GiB |       | 4 KiB     | 0 %  | uslab-24.uslab.excelero.com |
1 |        1 |  128 |
| Micron | MTFDHAL800MCE-1AN1ZABYY                 | P60913038105.1 |
745.21 GiB |       | 512 bytes | 0 %  | uslab-24.uslab.excelero.com |
1 |        1 |  128 |
| Micron | MTFDHAL800MCE-1AN1ZABYY                 | P60913038242.1 |
745.21 GiB |       | 512 bytes | 0 %  | uslab-23.uslab.excelero.com |
1 |        1 |  128 |
| Micron | Micron_9200_MTFDHAL3T2TCU______________ | 174019659D6C.1 |
2.91 TiB   |       | 512 bytes | 0 %  | uslab-21.uslab.excelero.com |
1 |        1 |  128 |
| Micron | Micron_9200_MTFDHAL3T2TCU______________ | 174019659D7E.1 |
2.91 TiB   |       | 512 bytes | 0 %  | uslab-22.uslab.excelero.com |
1 |        1 |  128 |
| Micron | Micron_9200_MTFDHAL3T2TCU______________ | 174019659D88.1 |
2.91 TiB   |       | 512 bytes | 0 %  | uslab-22.uslab.excelero.com |
1 |        1 |  128 |
| Micron | Micron_9200_MTFDHAL3T2TCU______________ | 174019659DA4.1 |
2.91 TiB   |       | 512 bytes | 0 %  | uslab-21.uslab.excelero.com |
1 |        1 |  128 |
------------------------------------------------------------------------
------------------------------------------------------------------------
---------------------

[TOMA leader information]
uslab-21 uslab-21.uslab.excelero.com
uslab-22 uslab-21.uslab.excelero.com
uslab-23 uslab-21.uslab.excelero.com
uslab-24 uslab-21.uslab.excelero.com

[Target information]
------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------------------------------------
| Target Name                  | Target Health | NVMesh Version | Target
Disks                                                    | Target NICs
|
------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------------------------------------
```

```
| uslab-21.uslab.excelero.com | Healthy      | 1.2.1-320      |
174019659D6C.1 174019659DA4.1 A0342961.1                     |
0xfe80000000000000248a0703008a83ae 0xfe80000000000000248a0703008a83af |
| uslab-22.uslab.excelero.com | Healthy      | 1.2.1-320      |
174019659D88.1 174019659D7E.1 P60913038002.1 P60913037990.1 |
0xfe80000000000000248a0703008a8606 0xfe80000000000000248a0703008a8607 |
| uslab-23.uslab.excelero.com | Healthy      | 1.2.1-320      |
P60913038242.1 P61015031037.1                                |
0xfe80000000000000248a0703008a8392 0xfe80000000000000248a0703008a8393 |
| uslab-24.uslab.excelero.com | Healthy      | 1.2.1-320      |
P60913038105.1 P60913038027.1                                |
0xfe80000000000000248a0703008a838e 0xfe80000000000000248a0703008a838f |
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------

[Client information]
------------------------------------------------------------------------
---------
| Client Name                 | Client Health | Client Version | Client
Volumes |
------------------------------------------------------------------------
---------
| uslab-21.uslab.excelero.com | Healthy      | 1.2.1-320      |
|
| uslab-22.uslab.excelero.com | Healthy      | 1.2.1-320      | test
|
| uslab-23.uslab.excelero.com | Healthy      | 1.2.1-320      |
|
| uslab-24.uslab.excelero.com | Healthy      | 1.2.1-320      |
|
------------------------------------------------------------------------
---------

[Volume information]
------------------------------------------------------------------------
-----------------------------------------------
Volume Name: test
Volume Health: Healthy
Volume Status: Online
Volume Type: Striped & Mirrored RAID-10
Volume Size: 1 TiB
Stripe Width: 2
Dirty Bits: 0 bytes
Target Names: uslab-22.uslab.excelero.com uslab-21.uslab.excelero.com
Target Disks: A0342961.1 174019659DA4.1 174019659D88.1 174019659D7E.1
Target Classes: n/a
Drive Classes: n/a
Awareness/Domain: n/a
Volume Layout:
```

```
------------------------------------------------------------------------
-----------------------------------------------
| Chunk | Stripe | Segment | Type     | LBA Start | LBA End   | Status |
Disk ID        | Last Known Target           |
------------------------------------------------------------------------
-----------------------------------------------
|     0 |      0 |       0 | data     | 4688672   | 138906399 |       |
A0342961.1     | uslab-21.uslab.excelero.com |
|     0 |      0 |       1 | data     | 3907264   | 138124991 |       |
174019659D88.1 | uslab-22.uslab.excelero.com |
|     0 |      0 |       2 | raftonly | n/a       | n/a       |       |
P60913038027.1 | uslab-24.uslab.excelero.com |
|     0 |      1 |       0 | data     | 3907264   | 138124991 |       |
174019659D7E.1 | uslab-22.uslab.excelero.com |
|     0 |      1 |       1 | data     | 3907264   | 138124991 |       |
174019659DA4.1 | uslab-21.uslab.excelero.com |
|     0 |      1 |       2 | raftonly | n/a       | n/a       |       |
P60913038027.1 | uslab-24.uslab.excelero.com |
------------------------------------------------------------------------
```

```
-----------------------------------------------
------------------------------------------------------------------------
-----------------------------------------------
```

show | nvmesh show

```
Show and view specific NVMesh objects and its properties. The 'show sub-
command allows output in a table, tabulator separated value or JSON
format.
E.g 'show targets' will show all targets. In case you want to see the
properties of only one or just a few you need to use the '-s' or
'--server'
option to specify single or a space separated list of servers/targets.
E.g. 'show targets -s
target1 target2'

Usage example: show volume -d -l
This will show/list all the volumes, the details and volume layout

positional arguments:

{cluster,target,client,volume,drive,manager,sshuser,apiuser,vpg,drivecla
ss,targetclass,host,log,drivemodel}
                        Define/specify the scope or the NVMesh object
you want
                        to list or view.

optional arguments:
  -h, --help            show this help message and exit
  -a, --all             Show all logs. Per default only alerts are
shown.
  -C CLASS [CLASS ...], --Class CLASS [CLASS ...]
                        A single or a space separated list of NVMesh
drives or
                        target classes.
  -d, --detail          Show more details.
  -l, --layout          Show the volume layout details. To be used
together
                        with the "-d" switch.
  -j, --json            Format output as JSON.
  -s SERVER [SERVER ...], --server SERVER [SERVER ...]
                        Space separated list or single server.
  -S, --short-name      Show short hostnames.
  -t, --tsv             Format output as tabulator separated values.
  -v VOLUME [VOLUME ...], --volume VOLUME [VOLUME ...]
                        View a single NVMesh volume or a list of
volumes.
  -p VPG [VPG ...], --vpg VPG [VPG ...]
                        View a single or a list of NVMesh volume
provisioning
                        groups.
```

add | nvmesh add

```
Usage: add [-h] {host,volume,driveclass,targetclass} [-a] [-r
RAID_LEVEL] [-v VPG] [-o DOMAIN] [-D DESCRIPTION] [-l LIMIT_BY_DISK
[LIMIT_BY_DISK ...]] [-L LIMIT_BY_TARGET [LIMIT_BY_TARGET ...]] [-m
DRIVE [DRIVE ...] | -f FILE] [-M MODEL] [-n NAME] [-O CLASSDOMAIN
[CLASSDOMAIN ...]] [-P PARITY] [-R NODE_REDUNDANCY] [-c COUNT] [-t
TARGET_CLASS [TARGET_CLASS ...]] [-d DRIVE_CLASS [DRIVE_CLASS ...]] [-w
STRIPE_WIDTH] [-s SERVER [SERVER ...]] [-S SIZE]

The 'add' sub-command will let you add nvmesh objects to your cluster.
E.g.
'add host' will add host entries to your nvmeshcli environment while
'add
volume' will create and add a new volume to the NVMesh cluster.

positional arguments:
  {host,volume,driveclass,targetclass}
                        Add hosts to this shell environment or
add/create new
                        NVMesh volumes or drive classes.

optional arguments:
  -h, --help            show this help message and exit
  -a, --autocreate      Create the drive classes automatically grouped
by the
                        available drive models.
  -r RAID_LEVEL, --raid_level RAID_LEVEL
                        The RAID level of the volume. Options: lvm, con,
0, 1, 10 and ec
  -v VPG, --vpg VPG     Optional - The volume provisioning group to use.
  -o DOMAIN, --domain DOMAIN
                        Awareness domain information to use for new
volume/s
                        or a VPG.
  -D DESCRIPTION, --description DESCRIPTION
                        Optional - Volume description
  -l LIMIT_BY_DISK [LIMIT_BY_DISK ...], --limit-by-disk LIMIT_BY_DISK
[LIMIT_BY_DISK ...]
                        Optional - Limit volume allocation to specific
drives.
  -L LIMIT_BY_TARGET [LIMIT_BY_TARGET ...], --limit-by-target
LIMIT_BY_TARGET [LIMIT_BY_TARGET ...]
                        Optional - Limit volume allocation to specific
target
                        nodes.
  -m DRIVE [DRIVE ...], --drive DRIVE [DRIVE ...]
                        Drive/media information. Needs to include the
drive
```

```
                        ID/serial and the targetnode/server name in the
format
                        driveId:targetNameExample: -m "Example:
                        174019659DA4.1:test.lab"
  -f FILE, --file FILE  Path to the file containing the
driveId:targetName
                        information. Example: -f "/path/to/file". This
                        argument is not allowed together with the -m
argument. Needs to be in the driveId:targetName format.
  -M MODEL, --model MODEL
                        Drive model information for the new drive class.
Note:
                        Must be the exactly the same model designator as
when
                        running the"show drivemodel -d" or "show drive
-d"
                        command!
  -n NAME, --name NAME  Name of the volume, must be unique, will be the
ID of
                        the volume.
  -O CLASSDOMAIN [CLASSDOMAIN ...], --classdomain CLASSDOMAIN
[CLASSDOMAIN ...]
                        Awareness domain/s information of the target or
drive
                        class. A domain has a scope and identifier
component.
                        You must provide both components for each domain
to be
                        used/created.-O scope:Rack&identifier:A or in
case you
                        want to use more than one domain descriptor:-O
                        scope:Rack&identifier:A
                        scope:Datacenter&identifier:DRsite
  -P PARITY, --parity PARITY
                        Parity configuration. Required for Erasure
Coding
                        NVMesh volumes. Example: "8+2" which equals to 8
data
                        blocks + 2 parity blocks
  -R NODE_REDUNDANCY, --node-redundancy NODE_REDUNDANCY
                        NVMesh Target node redundancy configuration.
Required
                        for Erasure Coding NVMesh volumes. NVMesh
supports
                        three target node redundancy levels, aka.
protection
                        levels.0 = no separation or redundancy on the
node
                        level; 1 = N+1 node redundancy or minimal
separation;
```

```
                        2 = N+2 redundancy or maximal separation. Chose
                        between 0, 1, or 2.
  -c COUNT, --count COUNT
                        Number of volumes to create and add. 100 Max.
  -t TARGET_CLASS [TARGET_CLASS ...], --target-class TARGET_CLASS
[TARGET_CLASS ...]
                        Limit volume allocation to specific target
classes.
  -d DRIVE_CLASS [DRIVE_CLASS ...], --drive-class DRIVE_CLASS
[DRIVE_CLASS ...]
                        Limit volume allocation to specific drive
classes.
  -w STRIPE_WIDTH, --stripe-width STRIPE_WIDTH
                        Number of disks to use. Required for R0 and R10.
  -s SERVER [SERVER ...], --server SERVER [SERVER ...]
                        Specify a single server or a space separated
list of
                        servers.
  -S SIZE, --size SIZE  Specify the size of the new volume. The volumes
size
                        value is base*2/binary. Example: -S 12GB or
12GiB will
                        create a volume with a size of 12884901888
bytes.Some
```

```
                        valid input formats samples: xGB, x GB, x
gigabyte, x
                        GiB or xG
```

update | nvmesh update

```
Usage: update [-h] {volume,driveclass,targetclass} -n NAME [-S SIZE
[SIZE ...]] [-D DESCRIPTION [DESCRIPTION ...]] [-s SERVER [SERVER ...]]
[-m DRIVE [DRIVE ...] | -f FILE] [-l LIMIT_BY_DISK [LIMIT_BY_DISK ...]]
[-L LIMIT_BY_TARGET [LIMIT_BY_TARGET ...]] [-t TARGET_CLASS
[TARGET_CLASS ...]] [-d DRIVE_CLASS [DRIVE_CLASS ...]]

Update and edit an existing NVMesh volume, driveclass or targetclass.

Usage Example: update volume -n vol006 -S 8t
This will update/change the size of volume vol006 to 8TiB.

positional arguments:
  {volume,driveclass,targetclass}
                        Specify the NVMesh object to be updated.

optional arguments:
  -h, --help            show this help message and exit
  -n NAME, --name NAME  The name of the object to be updated.
  -S SIZE [SIZE ...], --size SIZE [SIZE ...]
                        The new/updated size/capacity of the volume. The
                        volumes size value is base*2/binary. Example: -s
12GB
                        or 12GiB will size the volume with a size of
                        12884901888 bytes. Some valid input formats
samples:
                        xGB, x GB, x gigabyte, x GiB or xG
  -D DESCRIPTION [DESCRIPTION ...], --description DESCRIPTION
[DESCRIPTION ...]
                        The new/updated name of the NVMesh object.
  -s SERVER [SERVER ...], --server SERVER [SERVER ...]
                        Specify a single server or a space separated
list of
                        servers.
  -m DRIVE [DRIVE ...], --drive DRIVE [DRIVE ...]
                        Drive/media information. Needs to include the
drive
                        ID/serial and the targetnode/server name in the
format
                        driveId:targetNameExample: -m "Example:
                        174019659DA4.1:test.lab"
  -f FILE, --file FILE  Path to the file containing the
driveId:targetName
```

```
                        information. Needs toExample: -f
"/path/to/file". This
                        argument is not allowed together with the -m
argument
  -l LIMIT_BY_DISK [LIMIT_BY_DISK ...], --limit-by-disk LIMIT_BY_DISK
[LIMIT_BY_DISK ...]
                        Optional - Limit volume allocation to specific
drives.
  -L LIMIT_BY_TARGET [LIMIT_BY_TARGET ...], --limit-by-target
LIMIT_BY_TARGET [LIMIT_BY_TARGET ...]
                        Optional - Limit volume allocation to specific
target
                        nodes.
  -t TARGET_CLASS [TARGET_CLASS ...], --target-class TARGET_CLASS
[TARGET_CLASS ...]
                        Optional - Limit volume allocation to specific
target
                        classes.
  -d DRIVE_CLASS [DRIVE_CLASS ...], --drive-class DRIVE_CLASS
[DRIVE_CLASS ...]
```

```
                        Optional - Limit volume allocation to specific
drive
                        classes.
```

attach | nvmesh attach

SSH credentials required

```
Usage: attach [-h] -c CLIENT [CLIENT ...] -v VOLUME [VOLUME ...]

The 'attach' sub-command will let you attach NVMesh volumes to the
clients in
your NVMesh cluster.

Usage example: add -volume vol001 vol002 -c all
This will attach the two volumes vol001 and vol002 to all the clients in
the cluster.

optional arguments:
  -h, --help            show this help message and exit
  -c CLIENT [CLIENT ...], --client CLIENT [CLIENT ...]
                        Specify a single server or a space separated
list of
                        servers.
  -v VOLUME [VOLUME ...], --volume VOLUME [VOLUME ...]
                        Specify a single volume or a space separated
list of
                        volumes.
```

detach | nvmesh detach

SSH credentials required

```
Usage: detach [-h] -c CLIENT [CLIENT ...] -v VOLUME [VOLUME ...]

The 'detach' sub-command will let you detach NVMesh volumes in your
NVMesh
cluster.

Usage example: detach -v all -c all
This will detach all all the NVMesh volumes from all the NVMesh client

optional arguments:
  -h, --help            show this help message and exit
  -c CLIENT [CLIENT ...], --client CLIENT [CLIENT ...]
                        Specify a single server or a space separated
list of
                        servers.
  -v VOLUME [VOLUME ...], --volume VOLUME [VOLUME ...]
                        Specify a single volume or a space separated
list of
                        volumes.
```

runcmd | nvmesh runcmd

SSH credentials required

```
Usage: runcmd [-h] {client,target,manager,cluster,host} -c COMMAND
[COMMAND ...] [-p] [-P] [-s SERVER [SERVER ...]]

Run a remote shell command across the whole NVMesh cluster, or just the
targets, clients, managers or a list of selected servers and hosts.
Excample:
runcmd manager -c systemctl status mongod

Usage example: runcmd cluster -P -p -c date
This this will run the command 'date' in parallel on all managers,
tragets, and clients throughout the cluster.

positional arguments:
  {client,target,manager,cluster,host}
                        Specify the scope where you want to run the
command.

optional arguments:
  -h, --help            show this help message and exit
  -c COMMAND [COMMAND ...], --command COMMAND [COMMAND ...]
                        The command you want to run on the servers. Use
quotes
                        if the command needs to run with flags by
itself,
                        like: runcmd cluster -c "uname -a"
  -p, --prefix          Enabled per default. Adds the host name at the
beginning of each line. This
                        helps to identify the content when piping into a
grep
                        or similar tasks. To disable use '-p/--prefix
False'
  -P, --parallel        Enabled per default. Runs the remote command on
the remote hosts in
                        parallel. To disable use '-P/--parallel False'
  -s SERVER [SERVER ...], --server SERVER [SERVER ...]
                        Specify list of servers and or hosts.
```

delete | nvmesh delete

```
Usage: delete [-h] {host,volume,drive,driveclass,targetclass,vpg} [-s
SERVER [SERVER ...]] [-t TARGET_CLASS [TARGET_CLASS ...]] [-d
DRIVE_CLASS [DRIVE_CLASS ...]] [-D DRIVE [DRIVE ...]] [-v VOLUME [VOLUME
...]] [-f] [-y]

The 'delete' sub-command will let you delete nvmesh objects in your
cluster or
nvmesh-shell runtime environment. E.g. 'delete host' will delete host
entries
in your nvmesh-shell environment and 'delete volume' will delete NVMesh
volumes in your NVMesh cluster.

positional arguments:
  {host,volume,drive,driveclass,targetclass,vpg}
                        Delete hosts, servers, drives, drive classes,
and
                        target classes.

optional arguments:
  -h, --help            show this help message and exit
  -s SERVER [SERVER ...], --server SERVER [SERVER ...]
                        Specify a single server or a list of servers.
  -t TARGET_CLASS [TARGET_CLASS ...], --target-class TARGET_CLASS
[TARGET_CLASS ...]
                        Specify a single target class or a space
separated
                        list of target classes.
  -d DRIVE_CLASS [DRIVE_CLASS ...], --drive-class DRIVE_CLASS
[DRIVE_CLASS ...]
                        Specify a single drive class or a space
separated list
                        of drive classes.
  -D DRIVE [DRIVE ...], --drive DRIVE [DRIVE ...]
                        The drive ID of the drive to be deleted in the
NVMesh
                        cluster. Provide a space separated list for
multiple drives to be evicted.
  -v VOLUME [VOLUME ...], --volume VOLUME [VOLUME ...]
                        Specify a single volume or a space separated
list of
                        volumes.
  -f, --force           Use this flag to forcefully delete the volume/s.
  -y, --yes             Automatically answer 'yes' and skip operational
                        warnings.
```

evict | nvmesh evict

```
Usage: evict [-h] -d DRIVE [DRIVE ...] [-y]

Evict a drive in the NVMesh cluster.

optional arguments:
  -h, --help            show this help message and exit
  -d DRIVE [DRIVE ...], --drive DRIVE [DRIVE ...]
                        The drive ID of the drive to evict. Provide a
space separated list for multiple drives to be evicted.
  -y, --yes             Automatically answer 'yes' and skip operational
                        warnings.
```

check | nvmesh check

SSH credentials required

```
Usage: check [-h] {client,target,manager,cluster} [-d] [-s SERVER
[SERVER ...]]

The 'check' sub-command checks and let you list the status of the actual
NVMesh services running in your cluster. It is using SSH connectivity to
the
NVMesh managers, clients and targets to verify the service status. E.g.
'check
targets' will check the NVMesh target services throughout the cluster.

Usage example: check cluster -P
This will check all the NVMesh services throughout the NVMesh cluster.
The -P flag
has the tool executing and connecting via SSH to the servers in
parallel.

positional arguments:
  {client,target,manager,cluster}
                        Specify where you want to check the NVMesh
services
                        status.

optional arguments:
  -h, --help            show this help message and exit
  -d, --detail          Show detailed service information.
  -s SERVER [SERVER ...], --server SERVER [SERVER ...]
                        Specify a single or a space separated list of
                        managers, targets or clients.
```

stop | nvmesh stop

SSH credentials required

```
Usage: stop [-h] {client,target,manager,cluster,mcm} [-d] [-g
{True,False}]
[-s SERVER [SERVER ...]]

The 'stop' sub-command will stop the selected NVMesh services on all
managers,
targets and clients. Or it will stop the entire NVMesh cluster. It uses
SSH
connectivity to manage the NVMesh services. E.g. 'stop clients' will
stop all
the NVMesh clients throughout the cluster.

Usage example: stop client -P
This will stop all the NVMesh client services throughout the NVMesh
cluster in
parallel.

positional arguments:
  {client,target,manager,cluster,mcm}
                        Specify the NVMesh service type you want to top.

optional arguments:
  -h, --help            show this help message and exit
  -d, --detail          List and view the service details.
  -g {True,False}, --graceful {True,False}
                        Graceful stop of all NVMesh targets in the
cluster.
                        The default is set to 'True'
  -s SERVER [SERVER ...], --server SERVER [SERVER ...]
                        Specify a single or a space separated list of
                        managers, targets or clients.
```

start | nvmesh start

SSH credentials required

```
Usage: start [-h] {client,target,manager,cluster,mcm} [-d][-s SERVER
[SERVER ...]]

The 'start' sub-command will start the selected NVMesh services on all
managers, targets and clients. Or it will start the entire NVMesh
cluster. It
uses SSH connectivity to manage the NVMesh services. E.g. 'start
cluster' will
start all the NVMesh services throughout the cluster.

Usage example: start target -P
This will start the NVMesh target services throughout the NVMesh
cluster.

positional arguments:
  {client,target,manager,cluster,mcm}
                        Specify the NVMesh service type you want to
start.

optional arguments:
  -h, --help            show this help message and exit
  -d, --detail          List and view the service details.
  -s SERVER [SERVER ...], --server SERVER [SERVER ...]
                        Specify a single or a space separated list of
servers.
```

restart | nvmesh restart

SSH credentials required

```
Usage: restart [-h] {client,target,manager,cluster,mcm} [-d] [-g
{True,False}] [-s SERVER [SERVER ...]]

The 'restart' sub-command will restart the selected NVMesh services on
all
managers, targets and clients. Or it will restart the entire NVMesh
cluster.
It uses SSH connectivity to manage the NVMesh services. E.g. 'restart
managers' will restart the NVMesh management service.

Usage example: restart target
This will restart the target services throughout the NVMesh cluster.
It will use the more graceful API enpoint to stop all the target
services,
while the -g False option flag would use SSH connectivity and stop the
services
in parallel at once.

positional arguments:
  {client,target,manager,cluster,mcm}
                        Specify the NVMesh service which you want to
restart.

optional arguments:
  -h, --help            show this help message and exit
  -d, --detail          List and view the service details.
  -g {True,False}, --graceful {True,False}
                        Restart with a graceful stop of the targets in
the
                        cluster.The default is set to True
  -s SERVER [SERVER ...], --server SERVER [SERVER ...]
                        Specify a single or a space separated list of
servers.
```

testssh | nvmesh testssh

SSH credentials required

```
usage: testssh [-h] [-s SERVER [SERVER ...]]

Test the SSH connectivity to all, a list of, or individual servers and
hosts.

Usage excample: testssh -s servername
Or: testssh
The latter will test the ssh connectivity to all mamangers clients and
targets in your NVMesh cluster.

optional arguments:
  -h, --help            show this help message and exit
  -s SERVER [SERVER ...], --server SERVER [SERVER ...]
                        Specify a server or a space separated list of
servers and/or hosts.
```

Advanced Usage Examples

Change the NVMesh yum repository link throughout the whole NVMesh cluster

```
nvmesh # runcmd cluster -c “sed  -i
's,^baseurl=https://<username>:<password>@repo.excelero.com/repos/NVMesh/redhat/7.4,baseurl=https://<userna
me>:<password>@repo.excelero.com/repos/NVMesh/redhat/7.5,g' /etc/yum.repos.d/nvmesh.repo”
```

Delete all the volumes in an NVMesh cluster

```
# for volume in $(nvmesh show volume -t | awk '{ print $1}'); do nvmesh delete volume -v $volume; done
```

Delete all the volumes in an NVMesh cluster where the volume name contains "test"

```
# for volume in $(nvmesh show volume -t | grep test | awk '{ print $1}'); do nvmesh delete volume -v
$volume; done
```

Test the SSH connectivity to only the NVMesh target servers throughout the NVMesh cluster

```
# for target in $(nvmesh show target -t | awk '{ print $1}'); do nvmesh testssh '-s' $target; done
```

Check the MTU settings for all the network interfaces throughout your NVMesh cluster

```
# nvmesh runcmd cluster -c ip link | grep mtu | awk '{if ($0 ~/(lo:)/) {next;};print $1 " " $3 "\t" $5 " "
$6}'
```