NVMesh Shell/CLI Tool

Version

This documentation is for version 0.3.9

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.

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 preshared SSH key set up with all the involved hosts, servers and clients, the root SSH credential is required as well.
The easiest and quickest way to configure the required credential is to launch the nvmesh-shell tool and run the check cluster command. Please see as below.

```
# nvmesh-shell check cluster
```

The tool will ask will ask for the SSH credentials where you provide the root level username, i.e. root in most cases. 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 for the root level user was set up.
Then it will ask for the NVMesh API user credentials and the management server to be used.

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
~/.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

```
List available commands with "help" or detailed help with "help <command>"
```

define | nvmesh define

```
Usage: define [-h] {manager,sshuser,sshpassword,apiuser,apipassword} [-t] [-p
PASSWORD] [-u USER] [-s SERVER [SERVER ...]]

The 'define' sub-command defines/sets the shell runtime variables. It can be
```

```
used to set them temporarily or persistently. Please note that in its current
version allows to set only one NVMesh manager. If you try to provide a list
it
will use the first manager name of that list. E.g. 'define apiuser' will set
the NVMesh API user name to be used for all the operations involving the API

Usage examples to set or update:
       - the NVMesh management server:               define manager
       - the NVMesh management API user name:define -t apiuser -u <your
username>
       - the NVmesh management API password: define -t api password

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

optional arguments:
  -h, --help            show this help message and exit
  -t, --persistent      Define/Set the NVMesh runtime variable persistently.
  -p PASSWORD, --password PASSWORD
                        The password for the user to be used.
  -u USER, --user USER  The username name for the user to be used.
  -s SERVER [SERVER ...], --server SERVER [SERVER ...]
                        The NVMesh management server name
```

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 manager
show cluster
runcmd cluster -Pp -c "cat /var/log/NVMesh/toma_leader_name"
show drive -d
show volume -d -l
check manager -P
check target -P -d
check client -P -d
```

quick and basic nvmesh cli health check example output

```
nvmesh # load nvmesh_shell_sample_script
[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      |
------------------------------------------------------------------------------------------------------------------------------


[toma leader]
uslab-21 uslab-21.uslab.excelero.com
uslab-22 uslab-23.uslab.excelero.com
uslab-23 uslab-23.uslab.excelero.com
uslab-24 uslab-23.uslab.excelero.com
------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------
| Vendor | Model                                          | Drive ID       | Size       |
Status | BS        | Wear | Target                          | Numa | PCI root | SubQ |
------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------
| None   | SDLC2LLR-038T-3BA2_____________________ | A0342961.1     | 3.49 TiB   | Ok
| 512 bytes | 0 %  | uslab-21.uslab.excelero.com |    1 |        1 |   64 |
| Micron | MTFDHAL2T4MCF-1AN1ZABYY                | P61015031037.1 | 2.18 TiB   | Ok
| 512 bytes | 0 %  | uslab-23.uslab.excelero.com |    1 |        1 |  128 |
| Micron | MTFDHAL800MCE-1AN1ZABYY                | P60913037990.1 | 745.21 GiB | Ok
| 512 bytes | 0 %  | uslab-22.uslab.excelero.com |    1 |        1 |  128 |
| Micron | MTFDHAL800MCE-1AN1ZABYY                | P60913038002.1 | 745.21 GiB | Ok
| 512 bytes | 0 %  | uslab-22.uslab.excelero.com |    1 |        1 |  128 |
| Micron | MTFDHAL800MCE-1AN1ZABYY                | P60913038027.1 | 745.21 GiB | Ok
| 4 KiB     | 0 %  | uslab-24.uslab.excelero.com |    1 |        1 |  128 |
| Micron | MTFDHAL800MCE-1AN1ZABYY                | P60913038105.1 | 745.21 GiB | Ok
| 512 bytes | 0 %  | uslab-24.uslab.excelero.com |    1 |        1 |  128 |
| Micron | MTFDHAL800MCE-1AN1ZABYY                | P60913038242.1 | 745.21 GiB | Ok
| 512 bytes | 0 %  | uslab-23.uslab.excelero.com |    1 |        1 |  128 |
| Micron | Micron_9200_MTFDHAL3T2TCU______________ | 174019659D6C.1 | 2.91 TiB   | Ok
| 512 bytes | 0 %  | uslab-21.uslab.excelero.com |    1 |        1 |  128 |
| Micron | Micron_9200_MTFDHAL3T2TCU______________ | 174019659D6F.1 | 2.91 TiB   | Ok
| 512 bytes | 0 %  | uslab-24.uslab.excelero.com |    1 |        1 |  128 |
| Micron | Micron_9200_MTFDHAL3T2TCU______________ | 174019659D7E.1 | 2.91 TiB   | Ok
| 512 bytes | 0 %  | uslab-22.uslab.excelero.com |    1 |        1 |  128 |
| Micron | Micron_9200_MTFDHAL3T2TCU______________ | 174019659D88.1 | 2.91 TiB   | Ok
| 512 bytes | 0 %  | uslab-22.uslab.excelero.com |    1 |        1 |  128 |
| Micron | Micron_9200_MTFDHAL3T2TCU______________ | 174019659DA4.1 | 2.91 TiB   | Ok
| 512 bytes | 0 %  | uslab-21.uslab.excelero.com |    1 |        1 |  128 |
| Micron | Micron_9200_MTFDHAL3T2TCU______________ | 174019659DC7.1 | 2.91 TiB   | Ok
| 512 bytes | 0 %  | uslab-23.uslab.excelero.com |    1 |        1 |  128 |
------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------


[volume info]
Volume Name: testvol001
Volume Health: critical
Volume Status: offline
Volume Type: Striped RAID-0
Volume Size: 274 GiB
Stripe Width: 2
```

```
Dirty Bits: 0 bytes
Target Names: uslab-22.uslab.excelero.com uslab-21.uslab.excelero.com uslab-
24.uslab.excelero.com uslab-23.uslab.excelero.com
Target Disks: P61015031037.1 P60913037990.1 174019659D6C.1 174019659DC7.1
P60913038002.1 174019659D7E.1 A0342961.1 174019659D88.1 174019659D6F.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 |   5999391 | normal | A0342961.1     |
uslab-21.uslab.excelero.com |
|     0 |      1 |       0 | data |   3907264 |   5217983 | normal | 174019659DC7.1 |
uslab-23.uslab.excelero.com |
|     1 |      0 |       0 | data | 161975072 | 167217951 | normal | A0342961.1     |
uslab-21.uslab.excelero.com |
|     1 |      1 |       0 | data |  26189504 |  31432383 | normal | 174019659D7E.1 |
uslab-22.uslab.excelero.com |
|     2 |      0 |       0 | data | 167747264 | 174300863 | normal | 174019659DC7.1 |
uslab-23.uslab.excelero.com |
|     2 |      1 |       0 | data | 167747264 | 174300863 | normal | 174019659D6F.1 |
uslab-24.uslab.excelero.com |
|     3 |      0 |       0 | data | 488344352 | 504072991 | normal | A0342961.1     |
uslab-21.uslab.excelero.com |
|     3 |      1 |       0 | data | 335519424 | 351248063 | normal | 174019659D88.1 |
uslab-22.uslab.excelero.com |
|     4 |      0 |       0 | data | 665820864 | 672374463 | normal | 174019659D6C.1 |
uslab-21.uslab.excelero.com |
|     4 |      1 |       0 | data | 470857568 | 477411167 | normal | P61015031037.1 |
uslab-23.uslab.excelero.com |
|     5 |      0 |       0 | data | 187099264 | 187623551 | normal | P60913037990.1 |
uslab-22.uslab.excelero.com |
|     5 |      1 |       0 | data | 187099264 | 187623551 | normal | P60913038002.1 |
uslab-22.uslab.excelero.com |
-------------------------------------------------------------------------------------
-----------------------------
-------------------------------------------------------------------------------------
-------------------------------------------------------------------------


[manager status]
uslab-22 Check OK
The command 'netstat' wasn't found, can't check if listening on ports
Management is up!

uslab-23 Check OK
The command 'netstat' wasn't found, can't check if listening on ports
Management is up!

uslab-24 Check OK
The command 'netstat' wasn't found, can't check if listening on ports
Management is up!


[target status]
uslab-21 Check Failed
service nvmeshtarget status Detected Kernel: 3.10.0-862.9.1.el7.x86_64
Installed Ofed: MLNX_OFED_LINUX-4.3-3.0.2.1
```

```
NVMesh-target Version: 1.2.1-320

 All modules up

Managed NVMe Drives by Serial Number:
174019659D6C, 174019659DA4, A0342961
ManagementCM process running
Toma process not running
nvmeshtarget status [FAILED]

uslab-22 Check OK
Detected Kernel: 3.10.0-862.9.1.el7.x86_64
Installed Ofed: MLNX_OFED_LINUX-4.3-3.0.2.1
NVMesh-target Version: 1.2.1-320

 All modules up

Managed NVMe Drives by Serial Number:
174019659D88, 174019659D7E, P60913038002, P60913037990
ManagementCM process running
Toma process running

uslab-23 Check OK
Detected Kernel: 3.10.0-862.9.1.el7.x86_64
Installed Ofed: MLNX_OFED_LINUX-4.3-3.0.2.1
NVMesh-target Version: 1.2.1-320

 All modules up

Managed NVMe Drives by Serial Number:
P60913038242, P61015031037
ManagementCM process running
Toma process running

uslab-24 Check OK
Detected Kernel: 3.10.0-862.9.1.el7.x86_64
Installed Ofed: MLNX_OFED_LINUX-4.3-3.0.2.1
NVMesh-target Version: 1.2.1-320

 All modules up

Managed NVMe Drives by Serial Number:
P60913038105, P60913038027
ManagementCM process running
Toma process running


[client status]
uslab-21 Check OK
Detected Kernel: 3.10.0-862.9.1.el7.x86_64
Installed Ofed: MLNX_OFED_LINUX-4.3-3.0.2.1
NVMesh-client Version: 1.2.1-320

 All modules up

Attached Volumes:

Management Agent process running
ManagementCM process running

uslab-22 Check OK
Detected Kernel: 3.10.0-862.9.1.el7.x86_64
Installed Ofed: MLNX_OFED_LINUX-4.3-3.0.2.1
```

```
NVMesh-client Version: 1.2.1-320

 All modules up

Attached Volumes:

Management Agent process running
ManagementCM process running

uslab-23 Check OK
Detected Kernel: 3.10.0-862.9.1.el7.x86_64
Installed Ofed: MLNX_OFED_LINUX-4.3-3.0.2.1
NVMesh-client Version: 1.2.1-320

 All modules up

Attached Volumes:

Management Agent process running
ManagementCM process running

uslab-24 Check OK
Detected Kernel: 3.10.0-862.9.1.el7.x86_64
Installed Ofed: MLNX_OFED_LINUX-4.3-3.0.2.1
NVMesh-client Version: 1.2.1-320

 All modules up

Attached Volumes:
testvol001
Management Agent process running
ManagementCM process running
```

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,driveclass,ta
rgetclass,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] [-N NUMBER_OF_MIRRORS]
[-O CLASSDOMAIN [CLASSDOMAIN ...]] [-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 or
nvmesh-shell runtime environment. E.g. 'add hosts' will add host entries to
your nvmesh-shell environment while 'add volume' will create and add a new
volume to the NVMesh cluster.

Usage example: add volume -n vol -c 10 -S 1t -r 10 -w 2
This will create ten RAID10 volumes with a size of 1 TiB, stripe width of 2
and the names of vol[001:010]
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 drive classes automatically grouped by the
                        available drive models or target classes for each
target.
  -r RAID_LEVEL, --raid_level RAID_LEVEL
                        The RAID level of the volume. Options: lvm, 0, 1, 10
  -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. Needs toExample: -f /path/to/file"
  -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.
  -N NUMBER_OF_MIRRORS, --number-of-mirrors NUMBER_OF_MIRRORS
                        Number of mirrors to use.
  -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
  -c COUNT, --count COUNT
                        Number of volumes to create and add. 100 Max.
  -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.
  -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

```
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

```
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

```
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          Adds the host name at the beginning of each line.
This
                        helps to identify the content when piping into a grep
                        or similar tasks.
  -P, --parallel        Runs the remote command on the remote hosts in
                        parallel.
  -s SERVER [SERVER ...], --server SERVER [SERVER ...]
                        Specify list of servers and or hosts.
```

delete | nvmesh delete

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

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

Usage example: delete volume -v vol001 vol002 -f
This will forefully delete the two volumes vol001 and vol002

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

```
optional arguments:
  -h, --help            show this help message and exit
  -s SERVER [SERVER ...], --server SERVER [SERVER ...]
  -f, --force           Use this flag to forcefully delete the volume/s.
                        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.
  -v VOLUME [VOLUME ...], --volume VOLUME [VOLUME ...]
                        Specify a single volume or a space separated list of
                        volumes.
```

check | nvmesh check

```
Usage: check [-h] {client,target,manager,cluster} [-d] [-p] [-P] [-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.
  -p, --prefix          Adds the host name at the beginning of each line.
This
                        helps to identify the content when piping into a grep
                        or similar
  -P, --parallel        Check the hosts/servers in parallel.
  -s SERVER [SERVER ...], --server SERVER [SERVER ...]
                        Specify a single or a space separated list of
                        managers, targets or clients.
```

stop | nvmesh stop

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

```
[-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'
  -p, --prefix          Adds the host name at the beginning of each line.
This
                        helps to identify the content when piping into a grep
                        or similar
  -P, --parallel        Stop the NVMesh services in parallel.
  -s SERVER [SERVER ...], --server SERVER [SERVER ...]
                        Specify a single or a space separated list of
                        managers, targets or clients.
```

start | nvmesh start

```
Usage: start [-h] {client,target,manager,cluster,mcm} [-d] [-p] [-P] [-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.
  -p, --prefix          Adds the host name at the beginning of each line.
This
                        helps to identify the content when piping into a grep
```

```
                        or similar
  -P, --parallel        Start the NVMesh services on the hosts/servers in
                        parallel.
  -s SERVER [SERVER ...], --server SERVER [SERVER ...]
                        Specify a single or a space separated list of
servers.
```

restart | nvmesh restart

```
Usage: restart [-h] {client,target,manager,cluster,mcm} [-d] [-g
{True,False}] [-p] [-P] [-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
  -p, --prefix          Adds the host name at the beginning of each line.
This
                        helps to identify the content when piping into a grep
                        or similar
  -P, --parallel        Restart the NVMesh services on the hosts/servers in
                        parallel.
  -s SERVER [SERVER ...], --server SERVER [SERVER ...]
                        Specify a single or a space separated list of
servers.
```

testssh | nvmesh testssh

```
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 -P -p -c "sed  -i
's,^baseurl=https://<username>:<password>@repo.excelero.com/repos/NVMesh/redh
at/7.4,baseurl=https://<username>:<password>@repo.excelero.com/repos/NVMesh/r
edhat/7.5,g' /etc/yum.repos.d/nvmesh.repo"
```

Delete all the volumes in an NVMesh cluster

```
# for volume in $(nvmesh show volumes -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-shell show volumes -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 -P -p -c ip link | grep mtu | awk '{if ($0 ~/(lo:)/)
{next;};print $1 " " $3 "\t" $5 " " $6}'
```