NVMesh CLI Guide

1.2.1 — Last update: 2018/10/07

Excelero, Inc.

Table of Contents

1. Introduction

The **nvmesh-shell** CLI tool provides a standard command-line user interface to manage NVMesh. This interface can be used to send one-line management commands to NVMesh or write entire OS shell scripts. Additionally, it also offers an interactive shell interface by itself.
nvmesh-shell uses the NVMesh RESTful API and terminal command line tools providing the user a facility to run shell commands as well as an interactive shell for day-to-day management and provisioning activities with homogeneous semantics.

2. Installation

2.1. Supported Environments

Linux and MacOS running Python version 2.

> ! Python minimum requirement is 2.7.5. Python v3 is not supported.

2.2. 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.

2.3. Installation – Non-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 nvmesh-shell from 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 .
   ```
5. Change nvmesh_shell to be executable:
   ```
   chmod a+x nvmesh_shell.py
   ```

2.4. Installation – 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":https://virtualenv.pypa.io/en/stable/.

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 .`
6. Change nvmesh_shell to be executable:
   `chmod a+x nvmesh_shell.py`

3. Using NVMesh CLI

Initially, the **nvmesh-shell** 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 credentials is to launch **nvmesh-shell** and run the `check cluster` command:

```
$ ./nvmesh.py check cluster
```

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

```
nvmesh # define sshuser
Do you require sudo for SSH remote command execution? [Yes|No] :y
Please provide the user name to be used for SSH connectivity: <your username>
Please provide the SSH password:
```

To use root for SSH:

```
nvmesh # define sshuser
Do you require sudo for SSH remote command execution? [Yes|No] :n
Please provide the root level SSH user name: root
Please provide the SSH password:
```

If preshared keys are set up throughout, 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. In addition, the NVMesh management server information is stored in the users home directory.

3.1. nvmesh-shell Files

There are other files stored in the users home directory in addition to the credentials.

~/.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.		~/.n

4. Common Problems

This section describes the most common problems in running **nvmesh-shell**.

4.1. Installation Problems

ImportError: No module named cmd2

Description

nvmesh-shell requires various Python modules. In some cases, one of the modules is not installed correctly. For example:

```
[jack@nvme31 12:55:00 nvmesh-shell]$ ./nvmesh_shell.py
Traceback (most recent call last):
  File "./nvmesh_shell.py", line 24, in <module>
    from cmd2 import Cmd, with_argparser
ImportError: No module named cmd2
```

How to Fix

To diagnose, try to run `pip install .` again. For example:

```
bc.. [jack@nvme31 12:55:00 nvmesh-shell]$ pip install .

...
...
Requirement already satisfied: pycparser in /usr/lib/python2.7/site-packages
(from cffi>=1.1->bcrypt>=3.1.3->paramiko->nvmesh-shell==41) (2.18)
Installing collected packages: pyparsing, subprocess32, Cmd2, pyasn1, bcrypt,
cryptography, pynacl, paramiko, monotonic, humanfriendly, gnureadline,
nvmesh-shell
  Found existing installation: pyparsing 1.5.6
Cannot uninstall 'pyparsing'. It is a distutils installed project and thus we
cannot accurately determine which files belong to it which would lead to only a
partial uninstall.
[eyal@nvme31 12:56:14 nvmesh-shell]$ sudo pip uninstall pyparsing
Cannot uninstall 'pyparsing'. It is a distutils installed project and thus we
cannot accurately determine which files belong to it which would lead to only a
partial uninstall.
```

In the above example, **pyparsing** is installed incorrectly. To fix, it's possible to move it aside, and retry `pip install .`. For example:

```
cd /usr/lib/python2.7/site-package
sudo mv pyparsing* /tmp/
sudo cd ~/nvmesh-shell
sudo pip install .
```

RuntimeError: could not open display

Description

nvmesh-shell requires the GTK Python modules. In some cases, one of the modules is not installed correctly.

5. Command Reference

5.1. add

Name

add – add nvmesh objects to the cluster or nvmesh-shell runtime environment. For example ‘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

```
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]
```

Description

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, 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". This
                        argument is not allowed together with the -m argument
  -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
...]
                        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
```

5.2. alias

Name

alias – Define or display aliases.

Usage

```
alias [name] | [<name> <value>]
    Where:
        name - name of the alias being looked up, added, or replaced
        value - what the alias will be resolved to (if adding or replacing)
                this can contain spaces and does not need to be quoted
```

Description

```
   Without arguments, 'alias' prints a list of all aliases in a reusable form
which
    can be outputted to a startup_script to preserve aliases across sessions.

    With one argument, 'alias' shows the value of the specified alias.
    Example: alias ls  (Prints the value of the alias called 'ls' if it exists)

    With two or more arguments, 'alias' creates or replaces an alias.

    Example: alias ls !ls -lF

    If you want to use redirection or pipes in the alias, then either quote the
tokens with these
    characters or quote the entire alias value.

    Examples:
        alias save_results print_results ">" out.txt
        alias save_results print_results "> out.txt"
        alias save_results "print_results > out.txt"
```

5.3. attach

Name

attach – attach NVMesh volumes to NVMesh clients.

Usage

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

Description

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.
```

5.4. check

Name

check – check and list the status of NVMesh services. It is using SSH connectivity to the NVMesh managers, clients and targets to verify the service status. For example ‘check targets’ will check the NVMesh target services throughout the cluster.

Usage

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

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.
```

5.5. define

Name

define – define or set the shell runtime variables. It can be used to set them temporarily or persistently. Note that it allows to set only a single NVMesh manager. If you try to provide a list it will use the first manager name of that list. For example ‘define apiuser’ will set the NVMesh API user name to be used for all the operations involving the API.

Usage

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

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
```

5.6. delete

Name

delete – delete NVMesh objects from the cluster or nvmesh-shell runtime environment. For example ‘delete hosts’ will delete host entries from the nvmesh-shell environment and ‘delete volume’ will delete NVMesh volumes from the NVMesh cluster.

Usage

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

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 ...]
                        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.
 -f, --force            Use this flag to forcefully delete the volume/s.
 -y, --yes              Automatically answer and skip operational warnings.
```

5.7. detach

Name

detach – detach NVMesh volumes from NVMesh clients.

Usage

```
detach [-h] -c CLIENT [CLIENT ...] -v VOLUME [VOLUME ...] [-y]
```

Description

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.
 -y, --yes              Automatically answer and skip operational warnings.
```

5.8. edit

Name

edit – edit a file in a text editor.

Usage

```
edit [file_path]
    Where:
        * file_path - path to a file to open in editor
```

Description

The editor used is determined by the ``editor`` settable parameter. “set editor (program-name)” to change or set the EDITOR environment variable.

5.9. help

Name

help – List available commands with “help” or detailed help with “help cmd”.

5.10. history

Name

history – view, run, edit, and save previously entered commands.

Usage

```
history [-h] [-r | -e | -s | -o FILE | -t TRANSCRIPT] [arg]
```

Description

Positional Arguments

```
 arg                    empty               all history items
                         a                   one history item by number
                         a..b, a:b, a:, ..b  items by indices (inclusive)
                         [string]            items containing string
```

```
                        /regex/             items matching regular expression

optional arguments:
  -h, --help            show this help message and exit
  -r, --run             run selected history items
  -e, --edit            edit and then run selected history items
  -s, --script          script format; no separation lines
  -o FILE, --output-file FILE
                        output commands to a script file
  -t TRANSCRIPT, --transcript TRANSCRIPT
                        output commands and results to a transcript file
```

5.11. ipy

Name

ipy – enters an interactive IPython shell.

Description

Run python code from external files with ``run filename.py``.
End with ``Ctrl-D`` (Unix) / ``Ctrl-Z`` (Windows), ``quit()``, ‘`exit()``.

5.12. load

Name

load – run commands in a script file that is encoded as either ASCII or UTF-8 text.

Usage

```
load <file_path>

    * file_path - a file path pointing to a script

Script should contain one command per line, just like command would be typed in
console.
```

5.13. py

Name

py – invoke python command, shell, or script.

Usage

```
      py <command>: Executes a Python command.
       py: Enters interactive Python mode.
       End with ``Ctrl-D`` (Unix) / ``Ctrl-Z`` (Windows), ``quit()``,
'`exit()``.
       Non-python commands can be issued with ``cmd("your command")``.
       Run python code from external script files with ``run("script.py")``
```

5.14. pyscript

Name

pyscript – run a python script file inside the console.

Usage

```
pyscript <script_path> [script_arguments]
```

Description

Console commands can be executed inside this script with cmd(“your command”). However, you cannot run nested “py” or “pyscript” commands from within this script. Paths or arguments that contain spaces must be enclosed in quotes.

5.15. quit

Name

exit – exit NVMesh shell.

5.16. restart

Name

restart – 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. For example ‘restart managers’ will restart the NVMesh management service.

Usage

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

Description

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.
 -y, --yes              Automatically answer and skip operational warnings.
```

5.17. runcmd

Name

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

Usage

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

Description

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.
```

5.18. set

Name

set – set a parameter or list the current settings.

Usage

```
set [-h] [-a] [-l] [settable [settable ...]]
```

Description

Accepts abbreviated parameter names so long as there is no ambiguity. Call without arguments for a list of settable parameters with their values.

Positional Arguments

```
 settable    [param_name] [value]
```

Optional Arguments

```
 -h, --help  show this help message and exit
  -a, --all   display read-only settings as well
  -l, --long  describe function of parameter
```

5.19. shell

Name

shell – execute a command as if at the OS prompt.

Usage

```
shell <command> [arguments]
```

5.20. shortcuts

Name

shortcuts – list shortcuts (aliases) available.

5.21. show

Name

show – list and view specific NVMesh objects and its properties. The ‘list sub-command allows output in a table, tabulator separated value or JSON format.
For example ‘list targets’ will list 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 list of servers/targets. For example ‘list targets -s target1 target2’.

Usage

```
show [-h]
{cluster,target,client,volume,drive,manager,sshuser,apiuser,vpg,driveclass,targetclass,hos
[-a] [-C CLASS [CLASS ...]] [-d] [-l] [-j] [-s SERVER [SERVER ...]] [-S] [-t]
[-v VOLUME [VOLUME ...]] [-p VPG [VPG ...]]
```

Description

Positional Arguments

```
{cluster,target,client,volume,drive,manager,sshuser,apiuser,vpg,driveclass,targetclass,hos
                        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.
```

5.22. start

Name

start – 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. For example ‘start cluster’ will start all the NVMesh services throughout the cluster.

Usage

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

Description

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.
```

5.23. stop

Name

stop – 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. For example ‘stop clients’ will stop all the NVMesh clients throughout the cluster.

Usage

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

Description

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.
  -y, --yes             Automatically answer and skip operational warnings.
```

5.24. testssh

Name

testssh – Test the SSH connectivity to all, a list of, or individual servers and hosts. For example: testssh -s servername.

Usage

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

Description

Optional Arguments

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

5.25. unalias

Name

unalias – unset an alias.

Usage

```
unalias [-a] name [name ...]
    Where:
        name - name of the alias being unset

    Options:
        -a     remove all alias definitions
```

5.26. update

Name

update – update and edit an existing NVMesh volume, driveclass or targetclass.

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 ...]]
```

Description

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.
```