NVM Express (NVMe)* Initiator Wheel

Table of Contents

Note: The NVMe Initiator Wheel script, `nvme-wheel`, was developed and tested on Ubuntu* version 16.04 with Python* version 2.7.14.

Overview

An initiator host polls the Discovery Service to ensure that its connections to remote volumes are up-to-date. And the NVMe Intiator Wheel script, `nvme-wheel`, is a tool that performs these actions.

This ReadMe describes the operation of the NVMe Intiator Wheel, which is installed on an initiator host. The tool can be a cron job script that wraps up the `nvme-cli` utility.

Installation of the NVMe Initiator Wheel

Note: Installation of `nvme-wheel` requires `nvme-cli`. Instructions for how to download and install `nvme-cli` are available at: https://github.com/linux-nvme/nvme-cli. For more information refer to http://nvmexpress.org.

Dependencies

Installation of `nvme-wheel` requires a wheel module linked with Python2*.

Note: Wheel is a packaging format for Python and meant to replace the Egg format. To learn more about Wheel, go to https://pythonwheels.com/.

1. To install Python* and Pip Installs Packages (PIP):

```
[sudo] apt-get install python-pip python-dev build-essential
```

[Alternative] If the previous step does not set a PIP, use an easy install:

```
sudo easy_install pip
```

2. To install the wheel module:

```
pip install wheel
```

3. To install the `nvme-wheel`:

```
[sudo] bash INSTALL.sh
```

or

```
[sudo] -H pip install NVME_Wheel*
```

Note: "Wheel" is a module that allows users to install .whl packages. And `nvme-wheel` is the name of the script that we run in the terminal window.

Running `nvme-wheel`

To show available commands, type:

```
nvme-wheel
```

To run in continuous mode, use the `execute` command:

```
nvme-wheel execute
```

For more information about custom arguments, run:

```
nvme-wheel execute --help
```

Run `nvme-wheel` with Example Parameters

Below is an example of running the `nvme-wheel` command with example parameters:

```
nvme-wheel execute -t rdma -a 10.6.0.101 -s 4420
```

If you use an NVMe Qualified Name (NQN) different than your system Universally Unique Identifier (UUID), type:

```
nvme-wheel execute -t rdma -a 10.6.0.101 -s 4420 -q custom_id_for_nqn
```

Run `nvme-wheel` in the Background

To run in continuous mode and in the background, you can use `nohup`:

```
nohup nvme-wheel execute <<additional_args>> &
```

If `nohup` isn't available on your machine, try:

```
nvme-wheel execute <<additional_args>> &
```

Run `nvme-wheel` in cron table

Run the script automatically after a reboot by adding this line below, into cron table (crontab):

In terminal type `crontab -e` to open a crontab and type:

```
@reboot sudo -H nvme-wheel execute -t rdma -a 10.6.0.101 -s 4420 -q << custom_id_for_nqn >>
```

Get the UUID

To print the UUID, use:

```
nvme-wheel uuid
```

Change the Default Discovery Time

The `nvme-wheel` discovers a new target every five seconds. This default value can be changed by exporting a `NVME_CRON` (for example, change the default discovery time from five to 10 seconds):

```
NVME_CRON=10
```

Check the Status

`nvme-wheel` has the ability to present status information for a specific endpoint: `localhost:8082/health`.

Run the application using the `nvme-wheel execute` command and add an `-e/--endpoint` parameter at the end.

The default option provides health status as a response:

The program will return code `204` if everything works and code `503` if the script stops working (in debug mode program always return `200`).

In `DEBUG` mode you can expand this information with the following:

Item	Description
`status`	Serves an `Ok` or `Error` when the `nvme-wheel` has stopped working.
`last-timestamp`	The timestamp of the last request to the discovery target.
`cron-value`	The time (in seconds) between each discovery iteration.
`number-of-targets`	The number of targets connected to the initiator.
`connected-targets`	The detailed information about connected targets (name, port, address and status).

Change the Default Port

Export:

```
NVME_PORT=port-number
```

Change the Default Mode

Export:

```
NVME_MODE=DEBUG
```

To set the `PRODUCTION` mode, unset `NVME_MODE`.

For more information on the `nvme-cli` utility, refer to section 2.13.3, Provisioning Initiator Hosts, of the *Intel® RSD PSME User Guide v2.3.2* available at intel.com.