# TABLE OF CONTENTS

## Chapter 1. Introduction

1.1. DGX System Health Monitors ................................................................. 1

1.2. Configurable DSHM Features ................................................................. 3

1.2.1. Health Monitor Alerts ................................................................. 3

1.2.2. DSHM Alert List ........................................................................ 3

1.2.3. Health Monitor Policies ................................................................. 6

1.3. Verifying the Installation ................................................................. 6

1.3.1. Verifying DSHM Services ................................................................. 6

1.3.2. Verifying NVSM APIs Services ................................................................. 7

## Chapter 2. Using the NVSM CLI

2.1. Using the NVSM CLI Interactively ................................................................. 8

2.2. Using the NVSM CLI Non-Interactively ................................................................. 9

2.3. Getting Help ........................................................................................ 9

2.3.1. nvsm "man" Page ........................................................................ 10

2.3.2. nvsm --help Flag ........................................................................ 10

2.3.3. Help for NVSM CLI Commands ................................................................. 10

2.4. Examining System Health ...................................................................... 10

2.4.1. List of Basic Commands ................................................................. 10

2.4.2. Show Health .............................................................................. 12

2.4.3. Dump Health .............................................................................. 12

2.4.4. Show Storage ............................................................................. 12

2.4.4.1. Show Storage Alerts ................................................................. 13

2.4.4.2. Show Storage Drives ................................................................. 14

2.4.4.3. Show Storage Volumes ................................................................. 15

2.4.5. Show GPUs .............................................................................. 16

2.4.5.1. Showing Individual GPUs ................................................................. 17

2.4.5.2. Identifying GPU Health Incidents ................................................................. 17

2.4.6. Show Processors ........................................................................ 19

2.4.6.1. Show Processor Alerts ................................................................. 20

2.4.7. Show Memory ........................................................................... 21

2.4.7.1. Show Memory Alerts ................................................................. 22

2.4.8. Show Fans and Temperature ................................................................. 23

2.4.8.1. Show Thermal Alerts ................................................................. 23

2.4.8.2. Show Fans ........................................................................... 24

2.4.8.3. Show Temperatures ................................................................. 25

2.4.9. Show Power Supplies .................................................................... 26

2.4.9.1. Show Power Alerts ................................................................. 27

2.5. System Monitoring Configuration ................................................................. 28

2.5.1. Configuring Email Alerts ................................................................. 28

2.5.2. Understanding System Monitoring Policies ................................................................. 29

---

www.nvidia.com
NVIDIA System Management

DU-09242-001 _v05 | ii
2.5.2.1. Global Monitoring Policy .............................................................................................. 29
2.5.2.2. Memory Monitoring Policy .......................................................................................... 30
2.5.2.3. Processor Monitoring Policy ....................................................................................... 30
2.5.2.4. Storage Monitoring Policy .......................................................................................... 31
2.5.2.5. Thermal Monitoring Policy ......................................................................................... 32
2.5.2.6. Power Monitoring Policy ............................................................................................ 33

2.6. Performing System Management Tasks ........................................................................ 33
  2.6.1. Rebuilding a RAID 1 Array ............................................................................................ 34
  2.6.2. Setting MaxQ/MaxP on DGX-2 Systems ....................................................................... 35

Chapter 3. Configuring NVSM Security .............................................................................. 37
  3.1. Overview of NVSM Security .............................................................................................. 37
  3.2. What You Need to Configure NVSM Security .................................................................. 37
  3.3. How to Configure NVSM Security .................................................................................... 38

Chapter 4. Calling NVSM APIs .......................................................................................... 39
  4.1. Getting Authorization to Call NVSM APIs ..................................................................... 39
  4.2. Calling NVSM APIs Using cURL, Python, or a Browser .................................................... 40

Appendix A. Third-Party Licenses .......................................................................................... 43
Chapter 1.
INTRODUCTION

NVIDIA® System Management (NVSM) is a software framework for monitoring NVIDIA DGX™ nodes in a data center. It includes active health monitoring, system alerts, and log generation. It can be used as a standalone utility from the command line by system administrators.

The following is a high level diagram of the NVSM framework, showing the NVSM API services at the heart of the framework, the DGX System Health Monitors (DSHM) responsible for monitoring the health of key system components, and the NVSM CLI for user control.
1.1. DGX System Health Monitors

The NVSM software incorporates the DGX System Health Monitor (DSHM), which probes critical hardware components in a DGX system and provides notification of fluctuations in system health, faults, and potential failures.

Health monitors are responsible for monitoring the health of critical DGX system components and informing users when an event of significance is detected. Below are the list of health monitors.

- System Health Monitors
  - CPU
  - DIMM
- Storage Health Monitor
- Environment Health Monitors
  - PSU
  - Fan
The following diagram illustrates the individual health monitors within DSHM.

Each health monitor is launched as a systemd service and leverages NVSM APIs to perform health management responsibilities. Periodic polling of critical system events are performed by each monitor and on identifying an event of significance, the monitor raises an alert. The alert is recorded in persistent storage (on the OS drive) and a notification is sent to configured users.

1.2. Configurable DSHM Features

DSHM contains the following features that you can configure using the NVSM CLI:

- Health Monitor Alerts
- Health Monitor Policies

1.2.1. Health Monitor Alerts

Alerts are events of significance that require attention. When a health monitor detects such an event in the subsystem that it monitors, it generates an alert to inform the user. The default behavior is to log the alerts in persistent storage as well as to send an E-mail notification to registered users. Refer to the section Using the NVSM CLI for details about configuring users for receiving alert E-mail notifications.

Each alert has a ‘state’. An active alert can be in a ‘critical’ or ‘warning’ state. Here, ‘critical’ implies an event that needs immediate action, and ‘warning’ implies an event that needs user attention. When the alerting condition is removed, the alert state changes to ‘cleared’. Details of how to view the generated alerts recorded in the database are available in the section Using the NVSM CLI.

1.2.2. DSHM Alert List

The following table describes each DSHM alert ID.

<table>
<thead>
<tr>
<th>Event</th>
<th>Alert ID</th>
<th>Component ID</th>
<th>Message</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drive missing</td>
<td>NV-DRIVE-01</td>
<td>&lt;drive slot&gt;</td>
<td>Drive missing in slot &lt;slot number&gt;</td>
<td>Critical</td>
</tr>
<tr>
<td>Event</td>
<td>Alert ID</td>
<td>Component ID</td>
<td>Message</td>
<td>Severity</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------</td>
<td>--------------</td>
<td>-----------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Media errors in drive</td>
<td>NV-DRIVE-02</td>
<td>&lt;drive slot&gt;</td>
<td>Media errors detected in drive &lt;slot number&gt;</td>
<td>Warning</td>
</tr>
<tr>
<td>IO errors in drive</td>
<td>NV-DRIVE-03</td>
<td>&lt;drive slot&gt;</td>
<td>IO errors detected in drive &lt;slot number&gt;</td>
<td>Warning</td>
</tr>
<tr>
<td>NVMe controller failure in drive</td>
<td>NV-DRIVE-04</td>
<td>&lt;drive slot&gt;</td>
<td>NVMe controller failure detected in drive &lt;slot number&gt;</td>
<td>Critical</td>
</tr>
<tr>
<td>Drive available capacity below 10 percent</td>
<td>NV-DRIVE-05</td>
<td>&lt;drive slot&gt;</td>
<td>Available capacity percentage below critical threshold for drive &lt;slot number&gt;</td>
<td>Critical</td>
</tr>
<tr>
<td>Drive used percentage above 90</td>
<td>NV-DRIVE-06</td>
<td>&lt;drive slot&gt;</td>
<td>Drive used percentage above critical threshold for drive &lt;slot number&gt;</td>
<td>Critical</td>
</tr>
<tr>
<td>Unsupported drive inserted</td>
<td>NV-DRIVE-07</td>
<td>&lt;drive slot&gt;</td>
<td>System has unsupported drive &lt;slot number&gt;</td>
<td>Warning</td>
</tr>
<tr>
<td>RAID-0 corrupted</td>
<td>NV-VOL-01</td>
<td>NA</td>
<td>RAID-0 corrupted</td>
<td>Critical</td>
</tr>
<tr>
<td>RAID-1 corrupted</td>
<td>NV-VOL-02</td>
<td>NA</td>
<td>RAID-1 corrupted</td>
<td>Critical</td>
</tr>
<tr>
<td>ESP-1 corrupted</td>
<td>NV-VOL-03</td>
<td>NA</td>
<td>EFI System Partition 1 is corrupted</td>
<td>Warning</td>
</tr>
<tr>
<td>ESP-2 corrupted</td>
<td>NV-VOL-04</td>
<td>NA</td>
<td>EFI System Partition 2 is corrupted</td>
<td>Warning</td>
</tr>
<tr>
<td>Power supply failure detected</td>
<td>NV-PSU-01</td>
<td>&lt;PSU#&gt; where # is the PSU number.</td>
<td>Power supply module has failed.</td>
<td>Critical</td>
</tr>
<tr>
<td>PSU Predictive failure</td>
<td>NV-PSU-02</td>
<td>&lt;PSU#&gt; where # is the PSU number.</td>
<td>Detected predictive failure of the Power supply module.</td>
<td>Warning</td>
</tr>
<tr>
<td>PSU Input lost (AC/DC)</td>
<td>NV-PSU-03</td>
<td>&lt;PSU#&gt; where # is the PSU number.</td>
<td>Input to the Power supply module is missing</td>
<td>Critical</td>
</tr>
<tr>
<td>PSU input lost or out of range</td>
<td>NV-PSU-04</td>
<td>&lt;PSU#&gt; where # is the PSU number.</td>
<td>Input voltage is out of range for the Power Supply Module.</td>
<td>Critical</td>
</tr>
<tr>
<td>PSU Absent</td>
<td>NV-PSU-05</td>
<td>&lt;PSU#&gt; where # is the PSU number.</td>
<td>PSU is missing.</td>
<td>Warning</td>
</tr>
<tr>
<td>PDB Thermal exceeded</td>
<td>NV-PDB-01</td>
<td>&lt;PDB#&gt; where # is the PDB number</td>
<td>Operating temperature exceeds the thermal specifications of the component.</td>
<td>Critical</td>
</tr>
<tr>
<td>Fan speed exceeded</td>
<td>NV-FAN-01</td>
<td>&lt;FAN#_F&gt; or &lt;FAN#_R&gt;</td>
<td>Fan speed reading has exceeded the expected speed setting</td>
<td>Critical</td>
</tr>
<tr>
<td>Event</td>
<td>Alert ID</td>
<td>Component ID</td>
<td>Message</td>
<td>Severity</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>------------</td>
<td>--------------</td>
<td>-------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Fan speed readings unavailable</td>
<td>NV-FAN-02</td>
<td>&lt;FAN#_F&gt; or &lt;FAN#_R&gt;</td>
<td>Fan readings are inaccessible.</td>
<td>Critical</td>
</tr>
<tr>
<td></td>
<td></td>
<td>where # is the fan module number. F is for front fan. R is for rear fan.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU Internal error</td>
<td>NV-CPU-01</td>
<td>&lt;CPU#&gt;</td>
<td>An unrecoverable CPU Internal error has occurred.</td>
<td>Critical</td>
</tr>
<tr>
<td></td>
<td></td>
<td>where # is the CPU socket number (CPU0 or CPU1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU Thermtrip</td>
<td>NV-CPU-02</td>
<td>&lt;CPU#&gt;</td>
<td>CPU Thermtrip has occurred, processor socket temperature exceeded the thermal specifications of the component.</td>
<td>Critical</td>
</tr>
<tr>
<td></td>
<td></td>
<td>where # is the CPU socket number (CPU0 or CPU1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIMM Uncorrectable ECC</td>
<td>NV-DIMM-01</td>
<td>&lt;CPU#<em>DIMM</em>@$&gt;</td>
<td>Uncorrectable error is reported.</td>
<td>Critical</td>
</tr>
<tr>
<td></td>
<td></td>
<td>where # = (1, 2) @ = (A, B, C, D, E, F) $ = (1, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIMM Correctable ECC</td>
<td>NV-DIMM-02</td>
<td>&lt;CPU#<em>DIMM</em>@$&gt;</td>
<td>Correctable errors reported exceeds the configured threshold.</td>
<td>Warning</td>
</tr>
<tr>
<td></td>
<td></td>
<td>where # = (1, 2) @ = (A, B, C, D, E, F) $ = (1, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIMM Critical</td>
<td>NV-DIMM-03</td>
<td>&lt;CPU#<em>DIMM</em>@$&gt;</td>
<td>Unrecoverable error is observed on the DIMM, specific details of the error are unavailable.</td>
<td>Critical</td>
</tr>
<tr>
<td></td>
<td></td>
<td>where # = (1, 2) @ = (A, B, C, D, E, F) $ = (1, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GPU Critical</td>
<td>NV-GPU-01</td>
<td></td>
<td>System entered degraded mode, GPU is reporting an error</td>
<td>Critical</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 1.2.3. Health Monitor Policies

Users can tune certain aspects of health monitor behavior using health monitor policies. This includes details such as email related configuration for alert notification, selectively disabling devices to be monitored, etc. Details of the supported policies and how to configure them using the CLI are provided in the section Using the NVSM CLI.

### 1.3. Verifying the Installation

Before using NVSM, you can verify the installation to make sure all the services are present.

#### 1.3.1. Verifying DSHM Services

Health monitors are part of the DGX OS image and launched by systemd when DGX boots. You can verify if all the DSHM services are up and running using the systemctl command. Below is an example of verifying whether the environmental DSHM service is functional.

```
$ systemctl status nvsm-env-dshm
```

```
nvsm-env-dshm.service - Environmental DSHM service.
 Loaded: loaded (/user/lib/systemd/system/nvsm-env-dshm.service; enabled; vendor preset; enabled)
 Active: active (running) since Tues 2018-09-11 15:12:06 PDT; 3h 1min ago
 Main PID: 2540 (env_dshm)
 Tasks: 1 (limit 12287)
 CGroup: /system.slice/nvsm-env-dshm.service
    `1_2540 /usr/bin/python /usr/bin/env_dshm
```

Other modules can be verified using similar commands:

To verify the storage module:

```
$ sudo systemctl status nvsm-storage-dshm
```

To verify the system module:

```
$ sudo systemctl status nvsm-sys-dshm
```

To verify the environment module

```
$ sudo systemctl status nvsm-env-dshm
```
1.3.2. Verifying NVSM APIs Services

NVSM-APIS is part of the DGX OS image and is launched by systemd when DGX boots. The following are the services running under NVSM-APIS.

- nvsm-apis-plugin-environment
- nvsm-apis-mqtt
- nvms-apis-plugin-memory
- nvsm-apis-mongodb
- nvsm-apis
- nvsm-apis-selwatcher

You can verify if each NVSM-APIS service is up and running using the ‘systemctl’ command. For example, the following command verifies the memory service.

```bash
$ sudo systemctl status nvsm-apis-plugin-memory
```

You can also view all the NVSM-APIS services and their status with the following command.

```bash
$ sudo systemctl status -all nvsm-apis*
```
Chapter 2.
USING THE NVSM CLI

NVIDIA DGX-2 servers running DGX OS version 4.0.1 or later should come with NVSM pre-installed.

NVSM CLI communicates with the privileged NVSM API server, so NVSM CLI requires superuser privileges to run. All examples given in this guide are prefixed with the "sudo" command.

2.1. Using the NVSM CLI Interactively

Starting an interactive session

The command "sudo nvsm" will start an NVSM CLI interactive session.

```
user@dgx-2:~$ sudo nvsm
[sudo] password for user:
nvsm->
```

Once at the "nvsm->" prompt, the user can enter NVSM CLI commands to view and manage the DGX system.

Example command

One such command is "show fans", which prints the state of all fans known to NVSM.

```
nvsm-> show fans
/chassis/localhost/thermal/fans/FAN10_F
  Properties:
    Status_State = Enabled
    Status_Health = OK
    Name = FAN10_F
    MemberId = 19
    ReadingUnits = RPM
    LowerThresholdNonCritical = 5046.000
    Reading = 9802 RPM
    LowerThresholdCritical = 3596.000
...
/chassis/localhost/thermal/fans/PDB_FAN4
  Properties:
    Status_State = Enabled
    Status_Health = OK
    Name = PDB_FAN4
    MemberId = 23
```
ReadingUnits = RPM
LowerThresholdNonCritical = 11900.000
Reading = 14076 RPM
LowerThresholdCritical = 10744.000

Leaving an interactive session

To leave the NVSM CLI interactive session, use the "exit" command.

```
nvsm-> exit
user@dgx2:$
```

2.2. Using the NVSM CLI Non-Interactively

Any NVSM CLI command can be invoked from the system shell, without starting
an NVSM CLI interactive session. To do this, simply append the desired NVSM CLI
command to the "sudo nvsm" command. The "show fans" command given above can be
invoked directly from the system shell as follows.

```
user@dgx2:$ sudo nvsm show fans
/chassis/localhost/thermal/fans/FAN10_F
Properties:
  Status_State = Enabled
  Status_Health = OK
  Name = FAN10_F
  MemberId = 19
  ReadingUnits = RPM
  LowerThresholdNonCritical = 5046.000
  Reading = 9802 RPM
  LowerThresholdCritical = 3596.000
...
/chassis/localhost/thermal/fans/PDB_FAN4
Properties:
  Status_State = Enabled
  Status_Health = OK
  Name = PDB_FAN4
  MemberId = 23
  ReadingUnits = RPM
  LowerThresholdNonCritical = 11900.000
  Reading = 14076 RPM
  LowerThresholdCritical = 10744.000
user@dgx2:$
```

The output of some NVSM commands can be too large to fit on one screen, it is
sometimes useful to pipe this output to a paging utility such as "less".

```
user@dgx2:$ sudo nvsm show fans | less
```

Throughout this chapter, examples are given for both interactive and non-interactive
NVSM CLI use cases. Note that these interactive and non-interactive examples are
interchangeable.

2.3. Getting Help

Apart from the NVSM CLI User Guide (this document), there are many sources for
finding additional help for NVSM CLI and the related NVSM tools.
2.3.1. nvsm "man" Page

A man page for NVSM CLI is included on DGX systems with NVSM installed. The user can view this man page by invoking the 'man nvsm' command.

```
user@dgx2:~$ man nvsm
```

2.3.2. nvsm --help Flag

By passing the --help flag, the nvsm command itself will print a short description of the command line arguments it recognizes. These arguments affect the behavior of the NVSM CLI interactive session, such as inclusion of color or log messages.

```
user@dgx2:~$ nvsm --help
usage: nvsm [-h] [--color WHEN] [-i] [--] [<command>...]
NVIDIA System Management interface
optional arguments:
  -h, --help            show this help message and exit
  --color WHEN          Control colorization of output. Possible values for WHEN are "always", "never", or "auto". Default value is "auto".
  -i, --interactive     When this option is given, run in interactive mode. The default is automatic.
  --log-level {debug,info,warning,error,critical}
                        Set the output logging level. Default is 'warning'.
```

2.3.3. Help for NVSM CLI Commands

Each NVSM command within the NVSM CLI interactive session, such as show, set, and exit, recognizes a "--help" flag that describes the NVSM command and its arguments.

```
user@dgx2:~$ sudo nvsm
nvsm-> exit --help
usage: exit [-help]
optional arguments:
  -help, -h  show this help message and exit
```

2.4. Examining System Health

The most basic functionality of NVSM CLI is examination of system state. NVSM CLI provides a "show" command for this purpose.

Because NVSM CLI is modeled after the SMASH CLP, the output of the NVSM CLI "show" command should be familiar to users of BMC command line interfaces.

2.4.1. List of Basic Commands

The following table lists the basic commands (primarily "show"). Detailed use of these commands are explained in subsequent sections of the document.
<table>
<thead>
<tr>
<th>Global Commands</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ sudo nvsm show alerts</td>
<td></td>
</tr>
<tr>
<td>$ sudo nvsm show policy</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Health Commands</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ sudo nvsm show health</td>
<td>Displays overall system health</td>
</tr>
<tr>
<td>$ sudo nvsm dump health</td>
<td>Generates a health report file</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Storage Commands</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ sudo nvsm show storage</td>
<td>Displays all storage-related information</td>
</tr>
<tr>
<td>$ sudo nvsm show drives</td>
<td>Displays the storage drives</td>
</tr>
<tr>
<td>$ sudo nvsm show volumes</td>
<td>Displays the storage volumes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GPU Commands</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ sudo nvsm show gpus</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Processor Commands</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ sudo nvsm show processors</td>
<td>Displays information for all CPUs in the system</td>
</tr>
<tr>
<td>$ sudo nvsm show cpus</td>
<td>Alias for &quot;show processors&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Memory Commands</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ sudo nvsm show memory</td>
<td>Displays information for all installed DIMMs</td>
</tr>
<tr>
<td>$ sudo nvsm show dimms</td>
<td>Alias for &quot;show memory&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Thermal Commands</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ sudo nvsm show fans</td>
<td></td>
</tr>
<tr>
<td>$ sudo nvsm show temperatures</td>
<td></td>
</tr>
<tr>
<td>$ sudo nvsm show temps</td>
<td>Alias for &quot;show temperatures&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Power Commands</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ sudo nvsm show power</td>
<td></td>
</tr>
<tr>
<td>$ sudo nvsm show psus</td>
<td>Alias for &quot;show power&quot;</td>
</tr>
</tbody>
</table>
2.4.2. Show Health

The "show health" command can be used to quickly assess overall system health.

```
user@dgx-2:~$ sudo nvsm show health
```

Example output:

```
... Checks
------Verify installed DIMM memory sticks.........................
HealthyNumber of logical CPU cores [96]..........................
HealthyGPU link speed [0000:39:00.0][8GT/s]......................
HealthyGPU link width [0000:39:00.0][x16]........................
Healthy
...
Health Summary
-------------
205 out of 205 checks are Healthy
Overall system status is Healthy
```

If any system health problems are found, this will be reflected in the health summary at the bottom of the "show health" output. Detailed information on health checks performed will appear above.

2.4.3. Dump Health

The "dump health" command produces a health report file suitable for attaching to support tickets.

```
user@dgx-2:~$ sudo nvsm dump health
```

Example output:

```
Writing output to /tmp/nvsm-health-dgx-1-20180907085048.tar.xzDone.
```

The file produced by "dump health" is a familiar compressed tar archive, and its contents can be examined by using the "tar" command as shown in the following example.

```
user@dgx-2:~$ cd /tmp
user@dgx-2:/tmp$ sudo tar xlf nvsm-health-dgx-1-20180907085048.tar.xz
user@dgx-2:/tmp$ sudo ls ./nvsm-health-dgx-1-20180907085048
```

2.4.4. Show Storage

NVSM CLI provides a "show storage" command to view all storage-related information. This command can be invoked from the command line as follows.
Using the NVSM CLI

```bash
user@dgx-2:~$ sudo nvsm show storage
Alternatively, the "show drives" and "show volumes" NVSM commands will show the storage drives or storage volumes respectively.
```

```bash
user@dgx-2:~$ sudo nvsm show drives
...
user@dgx-2:~$ sudo nvsm show volumes
...
```

Within an NVSM CLI interactive session, the CLI targets related to storage are located under the /systems/localhost/storage/1 target.

```bash
user@dgx2:~$ sudo nvsm
nvsm-> cd /systems/localhost/storage/
nvsm(/systems/localhost/storage/)-> show
```

**Example output:**

```
/systems/localhost/storage/
Properties:
    DriveCount = 10
    Volumes = [ md0, md1, nvme0n1p1, nvme1n1p1 ]
Targets:
    alerts
drives
    policy
    volumes
Verbs:
    cd
    show
```

### 2.4.4.1. Show Storage Alerts

Storage alerts are generated when the DSHM monitoring daemon detects a storage-related problem and attempts to alert the user (via email or otherwise). Past storage alerts can be viewed within an NVSM CLI interactive session under the /systems/localhost/storage/1/alerts target.

```bash
user@dgx-2:~$ sudo nvsm
nvsm-> cd /systems/localhost/storage/alerts
nvsm(/systems/localhost/storage/alerts)-> show
```

**Example output:**

```
/systems/localhost/storage/alerts
Targets:
    alert0
    alert1
Verbs:
    cd
    show
```

In this example listing, there appear to be two storage alerts associated with this system. The contents of these alerts can be viewed with the "show" command.

For example:

```bash
nvsm(/systems/localhost/storage/alerts)-> show alert1
```

```
/systems/localhost/storage/alerts/alert1
Properties:
```
The message seen in this alert suggests a possible EFI partition corruption, which is an error condition that might adversely affect this system's ability to boot. Note that the text seen here reflects the exact message that the user would have seen when this alert was generated.

Possible categories for storage alerts are given in the table below.

<table>
<thead>
<tr>
<th>Alert ID</th>
<th>Severity</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>NV-DRIVE-01</td>
<td>Critical</td>
<td>Drive missing</td>
</tr>
<tr>
<td>NV-DRIVE-02</td>
<td>Warning</td>
<td>Media errors detected in drive</td>
</tr>
<tr>
<td>NV-DRIVE-03</td>
<td>Warning</td>
<td>IO errors detected in drive</td>
</tr>
<tr>
<td>NV-DRIVE-04</td>
<td>Critical</td>
<td>NVMe controller failure detected in drive</td>
</tr>
<tr>
<td>NV-DRIVE-05</td>
<td>Warning</td>
<td>Available spare block percentage is below critical threshold of ten percent</td>
</tr>
<tr>
<td>NV-DRIVE-06</td>
<td>Warning</td>
<td>NVM subsystem usage exceeded ninety percent</td>
</tr>
<tr>
<td>NV-DRIVE-07</td>
<td>Warning</td>
<td>System has unsupported drive</td>
</tr>
<tr>
<td>NV-VOL-01</td>
<td>Critical</td>
<td>RAID-0 corruption observed</td>
</tr>
<tr>
<td>NV-VOL-02</td>
<td>Critical</td>
<td>RAID-1 corruption observed</td>
</tr>
<tr>
<td>NV-VOL-03</td>
<td>Warning</td>
<td>EFI System Partition 1 corruption observed</td>
</tr>
<tr>
<td>NV-VOL-04</td>
<td>Warning</td>
<td>EFI System Partition 2 corruption observed</td>
</tr>
</tbody>
</table>

2.4.4.2. Show Storage Drives

Within an NVSM CLI interactive session, each storage drive on the system is represented by a target under the /systems/localhost/storage/drives target. A listing of drives can be obtained as follows.

```
user@dgx-2:~$ sudo nvsm
nvsm-> cd /systems/localhost/storage/drives
nvsm(/systems/localhost/storage/drives)-> show
```
Example output:

```
/systems/localhost/storage/drives
Targets:
  nvme0n1
  nvme1n1
  nvme2n1
  nvme3n1
  nvme4n1
  nvme5n1
  nvme6n1
  nvme7n1
  nvme8n1
  nvme9n1
Verbs:
  cd
  show
```

Details for any particular drive can be viewed with the "show" command.

For example:

```
nvsm(/systems/localhost/storage/drives)-> show nvme2n1
```

```
/systems/localhost/storage/drives/nvme2n1
Properties:
  Capacity = 3840755982336
  BlockSizeBytes = 7501476528
  SerialNumber = 18141C244707
  PartNumber = N/A
  Model = Micron_9200_MTFDHAL3T8TCT
  Revision = 100007C0
  Manufacturer = Micron Technology Inc
  Status_State = Enabled
  Status_Health = OK
  Name = Non-Volatile Memory Express
  MediaType = SSD
  IndicatorLED = N/A
  EncryptionStatus = N/A
  HotSpareType = N/A
  Protocol = NVMe
  NegotiatedSpeedsGbs = 0
  Id = 2
Verbs:
  cd
  show
```

### 2.4.4.3. Show Storage Volumes

Within an NVSM CLI interactive session, each storage volume on the system is represented by a target under the /systems/localhost/storage/volumes target. A listing of volumes can be obtained as follows.

```
user@dgx-2:~$ sudo nvsm
nvsm-> cd /systems/localhost/storage/volumes
nvsm(/systems/localhost/storage/volumes)-> show
```

Example output:

```
/systems/localhost/storage/volumes
Targets:
  md0
  md1
  nvme0n1p1
  nvme1n1p1
```
Details for any particular volume can be viewed with the "show" command.

For example:

```
nvsm(/systems/localhost/storage/volumes)-> show md0
```

```
/systems/localhost/storage/volumes/md0P

Properties:
- Status_State = Enabled
- Status_Health = OK
- Name = md0
- Encrypted = False
- VolumeType = RAID-1
- Drives = [ nvme0n1, nvme1n1 ]
- CapacityBytes = 893.6G
- Id = md0
```

Verbs:
- cd
- show

### 2.4.5. Show GPUs

Information for all GPUs installed on the system can be viewed invoking the "show gpus" command as follows.

```
user@dgx-2:~$ sudo nvsm show gpus
```

Within an NVSM CLI interactive session, the same information can be accessed under the /systems/localhost/gpus CLI target.

```
user@dgx-2:~$ sudo nvsm
nvsm-> cd /systems/localhost/gpus
nvsm(/systems/localhost/gpus)-> show
```

Example output:

```
/systems/localhost/gpus

Targets:
- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
```

Verbs:
- cd
- show

Details for any particular GPU can also be viewed with the "show" command.
For example:

```
nvsm(/systems/localhost/gpus)-> show 6
/systems/localhost/gpus/6
Properties:
    Inventory_ModelName = Tesla V100-SXM3-32GB
    Inventory_UUID = GPU-4c653056-0d6e-df7d-19c0-4663d6745b97
    Inventory_SerialNumber = 0332318502073
    Inventory_PCIeDeviceId = 1DB810DE
    Inventory_PCIeSubSystemId = 12AB10DE
    Inventory_BrandName = Tesla
    Inventory_PartNumber = 699-2G504-0200-000
Verbs:
    cd
    show
```

2.4.5.1. Showing Individual GPUs

Details for any particular GPU can also be viewed with the "show" command.

For example:

```
nvsm(/systems/localhost/gpus)-> show GPU6
/systems/localhost/gpus/GPU6
Properties:
    Inventory_ModelName = Tesla V100-SXM3-32GB
    Inventory_UUID = GPU-4c653056-0d6e-df7d-19c0-4663d6745b97
    Inventory_SerialNumber = 0332318502073
    Inventory_PCIeDeviceId = 1DB810DE
    Inventory_PCIeSubSystemId = 12AB10DE
    Inventory_BrandName = Tesla
    Inventory_PartNumber = 699-2G504-0200-000
    Specifications_MaxPCIeGen = 3
    Specifications_MaxPCIeLinkWidth = 16x
    Specifications_MaxSpeeds_GraphicsClock = 1597 MHz
    Specifications_MaxSpeeds_MemClock = 958 MHz
    Specifications_MaxSpeeds_SMClock = 1597 MHz
    Specifications_MaxSpeeds_VideoClock = 1432 MHz
    Connections_PCIeGen = 3
    Connections_PCIeLinkWidth = 16x
    Connections_PCIeLocation = 00000000:34:00.0
    Power_PowerDraw = 50.95 W
    Stats_ErrorStats_ECCMode = Enabled
    Stats_FrameBufferMemoryUsage_Free = 32510 MiB
    Stats_FrameBufferMemoryUsage_Total = 32510 MiB
    Stats_FrameBufferMemoryUsage_Used = 0 MiB
    Stats_PCIeRxThroughput = 0 KB/s
    Stats_PCIeTxThroughput = 0 KB/s
    Stats_PerformanceState = P0
    Stats_UtilDecoder = 0 %
    Stats_UtilEncoder = 0 %
    Stats_UtilGPU = 0 %
    Stats_UtilMemory = 0 %
    Status_Health = OK
Verbs:
    cd
    show
```

2.4.5.2. Identifying GPU Health Incidents

Explain the benefits of the task, the purpose of the task, who should perform the task, and when to perform the task in 50 words or fewer.
NVSM uses NVIDIA Data Center GPU Manager (DCGM) to continuously monitor GPU health, and reports GPU health issues as "GPU health incidents". Whenever GPU health incidents are present, NVSM indicates this state in the "Status_HealthRollup" property of the /systems/localhost/gpus CLI target.

"Status_HealthRollup" captures the overall health of all GPUs in the system in a single value. Check the "Status_HealthRollup" property before checking other properties when checking for GPU health incidents.

To check for GPU health incidents, do the following,

1. Display the “Properties” section of GPU health

```
~$ sudo nvsm
nvsm-> cd /systems/localhost/gpus
nvsm(/systems/localhost/gpus)-> show -display properties
```

A system with a GPU-related issue might report the following.

```
Properties:
  Status_HealthRollup = Critical
  Status_Health = OK
```

The "Status_Health = OK" property in this example indicates that NVSM did not find any system-level problems, such as missing drivers or incorrect device file permissions.

The "Status_HealthRollup = Critical" property indicates that at least one GPU in this system is exhibiting a "Critical" health incident.

2. To find this GPU, issue the following command to list the health status for each GPU.

```
~$ sudo nvsm
nvsm-> show -display properties=*health /systems/localhost/gpus/*
```

The GPU with the health incidents will be reported as in the following example for GPU14.

```
/systems/localhost/gpus/GPU14
Properties:
  Status_Health = Critical
```

3. Issue the following command to show the detailed health information for a particular GPU (GPU14 in this example).

```
nvsm-> cd /systems/localhost/gpus
nvsm(/systems/localhost/gpus)-> show -level all GPU14/health
```

The output shows all the incidents involving that particular GPU.
Using the NVSM CLI

The output in this example narrows down the scope to a specific incident (or incidents) on a specific GPU. DCGM will monitor for a variety of GPU conditions, so check "Status_HealthRollup" using NVSM CLI to understand each incident.

2.4.6. Show Processors

Information for all CPUs installed on the system can be viewed using the "show processors" command.

```
user@dgx-2$ sudo nvsm show processors
```

From within an NVSM CLI interactive session, the same information is available under the /systems/localhost/processors target.

```
user@dgx-2:$ sudo nvsm
nvsm-> cd /systems/localhost/processors
nvsm(/systems/localhost/processors)-> show
```

Example output:

```
/systems/localhost/processors
Targets:
    CPU0
    CPU1
    alerts
    policy
Verbs:
    cd
    show
```

Details for any particular CPU can be viewed using the "show" command.

For example:

```
nvsm(/systems/localhost/processors)-> show CPU0/systems/localhost/processors/CPU0
Properties:
    Id = CPU0
    InstructionSet = x86-64
    Manufacturer = Intel(R) Corporation
    MaxSpeedMHz = 3600
    Model = Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz
    Name = Central Processor
    ProcessorArchitecture = x86
    ProcessorId_EffectiveFamily = 6
    ProcessorId_EffectiveModel = 85
    ProcessorId_IdentificationRegisters = 0xBFEBFBFF00050654
    ProcessorId_Step = 4
    ProcessorId_VendorId = GenuineIntel
    ProcessorType = CPU
    Socket = CPU 0
    Status_Health = OK
    Status_State = Enabled
    TotalCores = 24
    TotalThreads = 48
```
2.4.6.1. Show Processor Alerts

Processor alerts are generated when the DSHM monitoring daemon detects a CPU Internal Error (IERR) or Thermal Trip and attempts to alert the user (via email or otherwise). Past processor alerts can be viewed within an NVSM CLI interactive session under the /systems/localhost/processors/alerts target.

```
user8@dgx-2:~$ sudo nvsm
nvsm-> cd /systems/localhost/processors/alerts
nvsm(/systems/localhost/processors/alerts)-> show
```

Example output:

```
/systems/localhost/processors/alerts
Targets:
  alert0
  alert1
  alert2
Verbs:
  cd
  show
```

This example listing appears to show three processor alerts associated with this system. The contents of these alerts can be viewed with the "show" command.

For example:

```
nvsm(/systems/localhost/processors/alerts)-> show alert2
```

```
/systems/localhost/processors/alerts/alert2
Properties:
  system_name = xpl-bu-06
  component_id = CPU0
  description = CPU is reporting an error.
  event_time = 2018-07-18T16:42:20.580050
  recommended_action =
    1. Please run nvsysinfo
    2. Please open a case with NVIDIA Enterprise Support at this address https://nvid.nvidia.com/enterpriselogin
    3. Attach this notification and the nvsysinfo log file from /tmp/nvsysinfo-XYZ*
  severity = Critical
  alert_id = NV-CPU-02
  system_serial = To be filled by O.E.M.
  message = System entered degraded mode, CPU0 is reporting an error.
  message_details = CPU Thermtip has occurred, processor socket temperature exceeded the thermal specifications of the component.
Verbs:
  cd
  show
```

Possible categories for processor alerts are given in the table below.

<table>
<thead>
<tr>
<th>Alert ID</th>
<th>Severity</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>NV-CPU-01</td>
<td>Critical</td>
<td>An unrecoverable CPU Internal error has occurred.</td>
</tr>
</tbody>
</table>
Using the NVSM CLI

### Alert ID

<table>
<thead>
<tr>
<th>Alert ID</th>
<th>Severity</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>NV-CPU-02</td>
<td>Critical</td>
<td>CPU Thermtrip has occurred, processor socket temperature exceeded the thermal specifications of the component.</td>
</tr>
</tbody>
</table>

#### 2.4.7. Show Memory

Information for all system memory (i.e. all DIMMs installed near the CPU, not including GPU memory) can be viewed using the "show memory" command.

```
user@dqx-2:~$ sudo nvsm show memory
```

From within an NVSM CLI interactive session, system memory information is accessible under the /systems/localhost/memory target.

```
lab@xpl-dvt-42:~$ sudo nvsm
nvsm-> cd /systems/localhost/memory
nvsm(/systems/localhost/memory)-> show
```

Example output:

```
/systems/localhost/memory
Targets:
  CPU0_DIMM_A1
  CPU0_DIMM_A2
  CPU0_DIMM_B1
  CPU0_DIMM_B2
  CPU0_DIMM_C1
  CPU0_DIMM_C2
  CPU0_DIMM_D1
  CPU0_DIMM_D2
  CPU0_DIMM_E1
  CPU0_DIMM_E2
  CPU0_DIMM_F1
  CPU0_DIMM_F2
  CPU1_DIMM_G1
  CPU1_DIMM_G2
  CPU1_DIMM_H1
  CPU1_DIMM_H2
  CPU1_DIMM_I1
  CPU1_DIMM_I2
  CPU1_DIMM_J1
  CPU1_DIMM_J2
  CPU1_DIMM_K1
  CPU1_DIMM_K2
  CPU1_DIMM_L1
  CPU1_DIMM_L2
alerts    policy
Verbs:
  cd
  show
```

Details for any particular memory DIMM can be viewed using the "show" command.

For example:

```
nvsm(/systems/localhost/memory)-> show CPU2_DIMM_B1
```

```
/systems/localhost/memory/CPU2_DIMM_B1
Properties:
  CapacityMiB = 65536
```
Using the NVSM CLI

DataWidthBits = 64
Description = DIMM DDR4 Synchronous
Id = CPU2_DIMM_B1
Name = Memory Instance
OperatingSpeedMhz = 2666
PartNumber = 72ASS8G72LZ-2G6B2
SerialNumber = 1CD83000
Status.Health = OK
Status.State = Enabled
VendorId = Micron

Verbs:
cd
show

2.4.7.1. Show Memory Alerts

On DGX systems with a Baseboard Management Controller (BMC), the BMC will
monitor DIMMs for correctable and uncorrectable errors. Whenever memory error
counts cross a certain threshold (as determined by SBIOS), a memory alert is generated
by the DSHM daemon in an attempt to notify the user (via email or otherwise).

Past memory alerts are accessible from an NVSM CLI interactive session under the /
systems/localhost/memory/alerts target.

user@dgx-2:$ sudo nvsm
nvsm-> cd /systems/localhost/memory/alerts
nvsm(/systems/localhost/memory/alerts)-> show

Example output:

/systems/localhost/memory/alerts
Targets:
   alert0
Verbs:
cd
show

This example listing appears to show one memory alert associated with this system. The
contents of this alert can be viewed with the "show" command.

For example:

nvsm(/systems/localhost/memory/alerts)-> show alert0

/systems/localhost/memory/alerts/alert0
Properties:
   system.name = xpl-bu-06
   component.id = CPU1_DIMM_A2
   description = DIMM is reporting an error.
   event.time = 2018-07-18T16:48:09.906572
   recommended.action =
      1. Please run nvsysinfo
      2. Please open a case with NVIDIA Enterprise Support at this address
         https://nvid.nvidia.com/enterpriselogin
      3. Attach this notification and the nvsysinfo log file from /tmp/
         nvsysinfo-XYZ*
   severity = Critical
   alert.id = NV-DIMM-01
   system.serial = To be filled by O.E.M.
   message = System entered degraded mode, CPU1_DIMM_A2 is reporting an error.
   message.details = Uncorrectable error is reported.
Verbs:
cd
Possible categories for memory alerts are given in the table below.

<table>
<thead>
<tr>
<th>Alert Type</th>
<th>Severity</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>NV-DIMM-01</td>
<td>Critical</td>
<td>Uncorrectable error is reported.</td>
</tr>
</tbody>
</table>

### 2.4.8. Show Fans and Temperature

NVSM CLI provides a "show fans" command to display information for each fan on the system.

```
~$ sudo nvsm show fans
```

Likewise, NVSM CLI provides a "show temperatures" command to display temperature information for each temperature sensor known to NVSM.

```
~$ sudo nvsm show temperatures
```

Within an NVSM CLI interactive session, targets related to fans and temperature are located under the `/chassis/localhost/thermal` target.

```
~$ sudo nvsm
nvsm-> cd /chassis/localhost/thermal
nvsm(/chassis/localhost/thermal)-> show
```

Example output:

```
/chassis/localhost/thermal
Targets:
   alerts
   fans
   policy
   temperatures
Verbs:
   cd
   show
```

### 2.4.8.1. Show Thermal Alerts

The DSHM daemon monitors fan speed and temperature sensors. When the values of these sensors violate certain threshold criteria, DSHM generates a thermal alert in an attempt to notify the user (via email or otherwise).

Past thermal alerts can be viewed in an NVSM CLI interactive session under the `/chassis/localhost/thermal/alerts` target.

```
user@dgx-2:~$ sudo nvsm
nvsm-> cd /chassis/localhost/thermal/alerts
nvsm(/chassis/localhost/thermal/alerts)-> show
```

Example output:

```
/chassis/localhost/thermal/alerts
Targets:
   alert0
Verbs:
   cd
   show
```
This example listing appears to show one thermal alert associated with this system. The contents of this alert can be viewed with the "show" command.

For example:

```
nvsm(/chassis/localhost/thermal/alerts) -> show alert0
/chassis/localhost/thermal/alerts/alert0
Properties:
    system_name = system-name
    component_id = FAN1_R
    description = Fan Module is reporting an error.
    event_time = 2018-07-12T15:12:22.076814
    recommended_action =
        1. Please run nvsysinfo
        2. Please open a case with NVIDIA Enterprise Support at this address
           https://nvid.nvidia.com/enterpriselogin
        3. Attach this notification and the nvsysinfo log file from /tmp/nvsysinfo-XYZ*
    severity = Critical
    alert_id = NV-FAN-01
    system_serial = To be filled by O.E.M.
    message = System entered degraded mode, FAN1_R is reporting an error.
    message_details = Fan speed reading has fallen below the expected speed setting.
Verbs:    cd    show
```

From the message in this alert, it appears that one of the rear fans is broken in this system. This is the exact message that the user would have received at the time this alert was generated, assuming alert notifications were enabled.

Possible categories for thermal-related (fan and temperature) alerts are given in the table below.

<table>
<thead>
<tr>
<th>Alert ID</th>
<th>Severity</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>NV-FAN-01</td>
<td>Critical</td>
<td>Fan speed reading has fallen below the expected speed setting.</td>
</tr>
<tr>
<td>NV-FAN-02</td>
<td>Critical</td>
<td>Fan readings are inaccessible.</td>
</tr>
<tr>
<td>NV-PDB-01</td>
<td>Critical</td>
<td>Operating temperature exceeds the thermal specifications of the component.</td>
</tr>
</tbody>
</table>

### 2.4.8.2. Show Fans

Within an NVSM CLI interactive session, each fan on the system is represented by a target under the /chassis/localhost/thermal/fans target. The "show" command can be used to obtain a listing of fans on the system.

```
user@dgx-2:~$ sudo nvsm
nvsm-> cd /chassis/localhost/thermal/fans
nvsm(/chassis/localhost/thermal/fans) -> show
Example output:
/chassis/localhost/thermal/fans
Targets:
    FAN10_F
    FAN10_R
    FAN1_F
```
Using the NVSM CLI

Again using the "show" command, the details for any given fan can be obtained as follows.

For example:

```
nvsm(/chassis/localhost/thermal/fans) => show PDB_FAN2
/chassis/localhost/thermal/fans/PDB_FAN2
Properties:
    Status_State = Enabled
    Status_Health = OK
    Name = PDB_FAN2
    MemberId = 21
    ReadingUnits = RPM
    LowerThresholdNonCritical = 11900.000
    Reading = 13804 RPM
    LowerThresholdCritical = 10744.000
Verbs:
    cd
    show
```

2.4.8.3. Show Temperatures

Each temperature sensor known to NVSM is represented as a target under the /chassis/localhost/thermal/temperatures target. A listing of temperature sensors on the system can be obtained using the following commands.

```
nvsm(/chassis/localhost/thermal/temperatures) => show
```

Example output:

```
/chassis/localhost/thermal/temperatures
Targets:
    PDB1
    PDB2
Verbs:
    cd
    show
```
As with fans, the details for any temperature sensor can be viewed with the "show" command.

For example:

```
nvsm(/chassis/localhost/thermal/temperatures)-> show PDB2
/chassis/localhost/thermal/temperatures/PDB2
Properties:
  Status_State = Enabled
  Status_Health = OK
  Name = PDB2
  PhysicalContext = PDB
  MemberId = 1
  ReadingCelsius = 20 degrees C
  UpperThresholdNonCritical = 127.000
  SensorNumber = 66h
  UpperThresholdCritical = 127.000
Verbs:
  cd
  show
```

### 2.4.9. Show Power Supplies

NVSM CLI provides a "show power" command to display information for all power supplies present on the system.

```
user@dgx-2:~ $ sudo nvsm show power
```

From an NVSM CLI interactive session, power supply information can be found under the /chassis/localhost/power target.

```
user@dgx-2:~ $ sudo nvsm
nvsm-> cd /chassis/localhost/power
nvsm(/chassis/localhost/power)-> show
```

Example output:

```
/chassis/localhost/power
Targets:
  PSU1
  PSU2
  PSU3
  PSU4
  PSU5
  PSU6
  alerts    policyVerbs:    cd    show
```

Details for any particular power supply can be viewed using the "show" command as follows.

For example:

```
nvsm(/chassis/localhost/power)-> show PSU4
```

```
/chassis/localhost/power/PSU4
Properties:
  Status_State = Present
  Status_Health = OK
  LastPowerOutputWatts = 442
  Name = PSU4
  SerialNumber = DTHTCD18240
  MemberId = 3
```
2.4.9.1. Show Power Alerts

The DSHM daemon monitors PSU status. When the PSU status is not ok, DSHM generates a power alert in an attempt to notify the user (via email or otherwise).

Prior power alerts can be viewed under the /chassis/localhost/power/alerts target of an NVSM CLI interactive session.

```
user@dgx-2:~$ sudo nvsm
nvsm-> cd /chassis/localhost/power/alerts
nvsm(/chassis/localhost/power/alerts)-> show
```

Example output:

```
/chassis/localhost/power/alerts
Targets:
   alert0
   alert1
   alert2
   alert3
   alert4
Verbs:
   cd
   show
```

This example listing shows a system with five prior power alerts. The details for any one of these alerts can be viewed using the "show" command.

For example:

```
nvsm(/chassis/localhost/power/alerts)-> show alert4
/chassis/localhost/power/alerts/alert4
Properties:
   system_name = system-name
   component_id = PSU4
   description = PSU is reporting an error.
   event_time = 2018-07-18T16:01:27.462005
   recommended_action =
      1. Please run nvsysinfo
      2. Please open a case with NVIDIA Enterprise Support at this address
         https://nvid.nvidia.com/enterprisel ogin
      3. Attach this notification and the nvsysinfo log file from /tmp/
nvsysinfo-XYZ*
   severity = Warning
   alert_id = NV-PSU-05
   system_serial = To be filled by O.E.M.
   message = System entered degraded mode, PSU4 is reporting an error.
   message_details = PSU is missing
Verbs:
   cd
   show
```

Possible categories for power alerts are given in the table below.
## 2.5. System Monitoring Configuration

NVSM provides a DSHM service that monitors the state of the DGX system.

NVSM CLI can be used to interact with the DSHM system monitoring service via the NVSM API server.

### 2.5.1. Configuring Email Alerts

In order to receive the Alerts generated by DSHM through email, configure the Email settings in the global policy using NVSM CLI. User shall receive email whenever a new alert gets generated. The sender address, recipient address(es), SMTP server IP address and SMTP server Port number must be configured according to the SMTP server settings hosted by the user.

Email configuration properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>email_sender</td>
<td>Sender email address&lt;br&gt;Must be a valid email address, otherwise no emails will be sent. [ <a href="mailto:sender@domain.com">sender@domain.com</a> ]</td>
</tr>
<tr>
<td>email_recipients</td>
<td>List of recipients to which the email shall be sent [ <a href="mailto:user1@domain.com">user1@domain.com</a>,<a href="mailto:user2@domain.com">user2@domain.com</a> ]</td>
</tr>
<tr>
<td>email_smtp_server_name</td>
<td>SMTP server name that the user wants to use for relaying email [ smtp.domain.com ]</td>
</tr>
<tr>
<td>email_smtp_server_port</td>
<td>Port Number used by the SMTP server for providing SMTP relay service. Numeric value</td>
</tr>
</tbody>
</table>

The following examples illustrate how to configure email settings in global policy using NVSM CLI.

```
user@dgx-2:~$ sudo nvsm set /policy email_sender=dgx-admin@nvidia.com
```

```
user@dgx-2:~$ sudo nvsm set /policy email_smtp_server_name=smtpserver.nvidia.com
```
Using the NVSM CLI

user@dgx-2:$ sudo nvsm set /policy email_recipients=jdoe@nvidia.com,jdeer@nvidia.com

user@dgx-2:$ sudo nvsm set /policy email_smtp_server_port=465

2.5.2. Understanding System Monitoring Policies

From within an NVSM CLI interactive session, system monitor policy settings are accessible under the following targets.

<table>
<thead>
<tr>
<th>CLI Target</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/policy</td>
<td>Global NVSM monitoring policy, such as email settings for alert notifications.</td>
</tr>
<tr>
<td>/systems/localhost/memory/policy</td>
<td>NVSM policy for monitoring DIMM correctable and uncorrectable errors.</td>
</tr>
<tr>
<td>/systems/localhost/processors/policy</td>
<td>NVSM policy for monitoring CPU machine-check exceptions (MCE)</td>
</tr>
<tr>
<td>/systems/localhost/storage/1/policy</td>
<td>NVSM policy for monitoring storage drives and volumes</td>
</tr>
<tr>
<td>/chassis/localhost/thermal/policy</td>
<td>NVSM policy for monitoring fan speed and temperature as reported by the baseboard management controller (BMC)</td>
</tr>
<tr>
<td>/chassis/localhost/power/policy</td>
<td>NVSM policy for monitoring power supply voltages as reported by the BMC</td>
</tr>
</tbody>
</table>

2.5.2.1. Global Monitoring Policy

Global monitoring policy is represented by the /policy target of NVSM CLI.

user@dgx-2:$ sudo nvsm show /policy

Example output:

/policy
Properties:
  email_sender = NVIDIA DSHM Service
  email_smtp_server_name = smtp.example.com
  email_recipients = jdoe@nvidia.com,jdeer@nvidia.com
  email_smtp_server_port = 465
Verbs:
cd
set
show

The properties for global monitoring policy are described in the table below.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>email_sender</td>
<td>Sender email address</td>
</tr>
<tr>
<td></td>
<td>[ <a href="mailto:sender@domain.com">sender@domain.com</a> ]</td>
</tr>
<tr>
<td>email_recipients</td>
<td>List of recipients to which the email shall be sent</td>
</tr>
<tr>
<td></td>
<td>[ <a href="mailto:user1@domain.com">user1@domain.com</a>,<a href="mailto:user2@domain.com">user2@domain.com</a> ]</td>
</tr>
</tbody>
</table>
Using the NVSM CLI

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>email_smtp_server_name</td>
<td>SMTP server name that the user wants to use for relaying email</td>
</tr>
<tr>
<td></td>
<td>[ smtp.domain.com ]</td>
</tr>
<tr>
<td>email_smtp_server_port</td>
<td>Port Number used by the SMTP server for providing SMTP relay service.</td>
</tr>
<tr>
<td></td>
<td>Numeric value</td>
</tr>
</tbody>
</table>

### 2.5.2.2. Memory Monitoring Policy

Memory monitoring policy is represented by the /systems/localhost/memory/policy target of NVSM CLI.

```
user@dgx-2:~$ sudo nvsm show /systems/localhost/memory/policy
```

Example output:

```
/systems/localhost/memory/policy
Properties:
  mute_notification =
  mute_monitoring =
  poll_interval = 10
Verbs:
  cd
  set
  show
```

The properties for memory monitoring policy are described in the table below.

<table>
<thead>
<tr>
<th>Property</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mute_notification</td>
<td>List of comma separated DIMM IDs</td>
<td>Email alert notification is suppressed for devices in the list.</td>
</tr>
<tr>
<td></td>
<td>Example: CPU1_DIMM_A1,CPU2_DIMM_F2</td>
<td></td>
</tr>
<tr>
<td>mute_monitoring</td>
<td>List of comma separated DIMM IDs</td>
<td>Health monitoring is suppressed for devices in the list.</td>
</tr>
<tr>
<td></td>
<td>Example: CPU1_DIMM_A1,CPU2_DIMM_F2</td>
<td></td>
</tr>
<tr>
<td>poll_interval</td>
<td>Positive integer</td>
<td>DSHM checks the health of the devices periodically. By default, this polling occurs every 10 seconds. The poll interval can be configured through this property.</td>
</tr>
</tbody>
</table>

### 2.5.2.3. Processor Monitoring Policy

Processor monitoring policy is represented by the /systems/localhost/processors/policy target of NVSM CLI.

```
user@dgx-2:~$ sudo nvsm show /systems/localhost/processors/policy
```

Example output:

```
/systems/localhost/processors/policy
Properties:
```
The properties for processor monitoring policy are described in the table below.

<table>
<thead>
<tr>
<th>Property</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mute_notification</td>
<td>List of comma separated CPU IDs. Example: CPU0,CPU1</td>
<td>Email alert notification is suppressed for devices in the list.</td>
</tr>
<tr>
<td>mute_monitoring</td>
<td>List of comma separated CPU IDs Example: CPU0,CPU1</td>
<td>Health monitoring is suppressed for devices in the list.</td>
</tr>
<tr>
<td>poll_interval</td>
<td>Positive integer</td>
<td>DSHM checks the health of the devices periodically. By default, this polling occurs every 10 seconds. The poll interval can be configured through this property.</td>
</tr>
</tbody>
</table>

2.5.2.4. Storage Monitoring Policy

Storage monitoring policy is represented by the /systems/localhost/storage/1/policy target of NVSM CLI.

```
user@dgx-2:~$ sudo nvsm show /systems/localhost/storage/policy
```

Example output:

```
/systems/localhost/storage/policy
Properties:
  volume_mute_monitoring =
  volume_poll_interval = 10
  drive_mute_monitoring =
  drive_mute_notification =
  drive_poll_interval = 10
  volume_mute_notification =
Verbs:
  cd
  set
  show
```

The properties for storage monitoring policy are described in the table below.

<table>
<thead>
<tr>
<th>Property</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>drive_mute_notification</td>
<td>List of comma separated drive slots Example: 0, 1 etc</td>
<td>Email alert notification is suppressed for drives in the list.</td>
</tr>
<tr>
<td>drive_mute_monitoring</td>
<td>List of comma separated drive slots Example: 0, 1 etc</td>
<td>Health monitoring is suppressed for drives in the list.</td>
</tr>
<tr>
<td>Property</td>
<td>Syntax</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>drive_poll_interval</td>
<td>Positive integer</td>
<td>DSHM checks the health of the drives periodically. By default, this polling occurs every 10 seconds. The poll interval can be configured through this property.</td>
</tr>
<tr>
<td>volume_mute_notification</td>
<td>List of comma separated volume identifier</td>
<td>Example: md0, md1 etc</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Email alert notification is suppressed for volumes in the list.</td>
</tr>
<tr>
<td>volume_mute_monitoring</td>
<td>List of comma separated volume identifier</td>
<td>Example: md0, md1 etc</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Health monitoring is suppressed for volumes in the list.</td>
</tr>
<tr>
<td>volume_poll_interval</td>
<td>Positive integer</td>
<td>DSHM checks the health of the volumes periodically. By default, this polling occurs every 10 seconds. The poll interval can be configured through this property.</td>
</tr>
</tbody>
</table>

### 2.5.2.5. Thermal Monitoring Policy

Thermal monitoring policy (for fan speed and temperature) is represented by the /chassis/localhost/thermal/policy target of NVSM CLI.

```bash
user@dgx-2:~$ sudo nvsm show /chassis/localhost/thermal/policy
```

Example output:

```bash
/chassis/localhost/thermal/policy
Properties:
   fan_mute_notification =
   pdb_mute_monitoring =
   fan_mute_monitoring =
   fan_poll_interval = 20
   pdb_poll_interval = 10
   pdb_mute_notification =
Verbs:
   cd
   set
   show
```

The properties for thermal monitoring policy are described in the table below.

<table>
<thead>
<tr>
<th>Property</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fan_mute_notification</td>
<td>List of comma separated FAN IDs.</td>
<td>Email alert notification is suppressed for devices in the list.</td>
</tr>
<tr>
<td></td>
<td>Example: FAN2_R,FAN1_L,PDB_FAN2</td>
<td></td>
</tr>
<tr>
<td>fan_mute_monitoring</td>
<td>List of comma separated FAN IDs</td>
<td>Health monitoring is suppressed for devices in the list.</td>
</tr>
<tr>
<td></td>
<td>Example: FAN6_F,PDB_FAN1</td>
<td></td>
</tr>
<tr>
<td>fan_poll_interval</td>
<td>Positive integer</td>
<td>DSHM checks the health of the devices periodically. By default, this polling occurs every 10 seconds.</td>
</tr>
</tbody>
</table>
2.5.2.6. Power Monitoring Policy

Power monitoring policy is represented by the /chassis/localhost/power/policy target of NVSM CLI.

```bash
user@dgx-2:~$ sudo nvsm show /chassis/localhost/power/policy
```

Example output:

```
/chassis/localhost/power/policy
Properties:
  mute_notification =
  mute_monitoring =
  poll_interval = 10
Verbs:
  cd
  set
  show
```

The properties for power monitoring policy are described in the table below.

<table>
<thead>
<tr>
<th>Property</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mute_notification</td>
<td>List of comma separated PSU IDs.</td>
<td>Email alert notification is suppressed for devices in the list.</td>
</tr>
<tr>
<td></td>
<td>Example: PSU4,PSU2</td>
<td></td>
</tr>
<tr>
<td>mute_monitoring</td>
<td>List of comma separated FAN IDs.</td>
<td>Health monitoring is suppressed for devices in the list.</td>
</tr>
<tr>
<td></td>
<td>Example: PSU1,PSU4</td>
<td></td>
</tr>
<tr>
<td>poll_interval</td>
<td>Positive integer</td>
<td>DSHM checks the health of the devices periodically. By default, this polling occurs every 10 seconds. The poll interval can be configured through this property.</td>
</tr>
</tbody>
</table>

2.6. Performing System Management Tasks

This section describes commands for accomplishing some system management tasks.
2.6.1. Rebuilding a RAID 1 Array

For DGX systems with two NVMe OS drives configure as a RAID 1 array, the operating system is installed on volume md0. You can use NVSM CLI to view the health of the RAID volume and then rebuild the RAID array on two healthy drives.

**Viewing a Healthy RAID Volume**

On a healthy system, this volume appears with two drives and "Status_Health = OK". For example:

```
nvsm-> cd /systems/localhost/storage
nvsm(/systems/localhost/storage)-> show volumes/md0
/systems/localhost/storage/volumes/md0
Properties:  
  Status_State = Enabled  
  Status_Health = OK  
  Name = md0  
  Encrypted = False  
  VolumeType = RAID-1  
  Drives = [ nvme0n1, nvme1n1 ]  
  CapacityBytes = 893.6G  
  Id = md0  
Targets:  
  rebuild  
Verbs:  
  cd  
  show
```

**Viewing a Degraded RAID Volume**

On a system with degraded OS volume, the md0 volume will appear with only one drive, with messages "Status_Health = Warning", and "Status_State = Degraded" reported as follows.

```
nvsm-> cd /systems/localhost/storage
nvsm(/systems/localhost/storage)-> show volumes/md0
/systems/localhost/storage/volumes/md0
Properties:  
  Status_State = Degraded  
  Status_Health = Warning  
  Name = md0  
  Encrypted = False  
  VolumeType = RAID-1  
  Drives = [ nvme1n1 ]  
  CapacityBytes = 893.6G  
  Id = md0
Targets:  
  rebuild  
Verbs:  
  cd  
  show
```

In this situation, the OS volume is missing its parity drive.
Rebuilding the RAID 1 Volume

To rebuild the RAID array, make sure that you have installed a known good NVMe drive for the parity drive.

The RAID rebuilding process should begin automatically upon turning on the system. If it does not start automatically, use NVSM CLI to manually rebuild the array as follows.

1. Start an NVSM CLI interactive session and switch to the storage target.

   ```
   $ sudo nvsm
   nvsm-> cd /systems/localhost/storage
   ```

2. Start the rebuilding process and be ready to enter the device name of the replaced drive.

   ```
   nvsm(/systems/localhost/storage)-> start volumes/md0/rebuild
   PROMPT: In order to rebuild this volume, a spare drive is required. Please specify the spare drive to use to rebuild md0.
   Name of spare drive for md0 rebuild (CTRL-C to cancel): nvmeXn1
   WARNING: Once the volume rebuild process is started, the process cannot be stopped.
   Start RAID-1 rebuild on md0? [y/n] y
   ```

3. After entering `y` at the prompt to start the RAID 1 rebuild, the "Initiating rebuild ..." message appears.

   ```
   /systems/localhost/storage/volumes/md0/rebuild started at 2018-10-12 15:27:26.525187
   Initiating RAID-1 rebuild on volume md0...
   0.0% [ ]
   ```

   After about 30 seconds, the "Rebuilding RAID-1 ..." message should appear.

   ```
   /systems/localhost/storage/volumes/md0/rebuild started at 2018-10-12 15:27:26.525187
   Rebuilding RAID-1 rebuild on volume md0...
   31.0% [ ===========/ ]
   ```

   If this message remains at "Initiating RAID-1 rebuild" for more than 30 seconds, then there is a problem with the rebuild process. In this case, make sure the name of the replacement drive is correct and try again.

   The RAID 1 rebuild process should take about 1 hour to complete.

   For more detailed information on replacing a failed NVMe OS drive, see the NVIDIA DGX-2 Service Manual.

2.6.2. Setting MaxQ/MaxP on DGX-2 Systems

Beginning with DGX OS 4.0.5, you can set two GPU performance modes – MaxQ or MaxP.

Support on DGX-2 systems requires BMC firmware version 1.04.03 or later. MaxQ/MaxP is not supported on DGX-2H systems.
MaxQ

- Maximum efficiency mode
- Allows two DGX-2 systems to be installed in racks that have a power budget of 18 kW.
- Switch to MaxQ mode as follows.

```
$ sudo nvsm set powermode=maxq
```

The settings are preserved across reboots.

MaxP

- Default mode for maximum performance
- GPUs operate unconstrained up to the thermal design power (TDP) level.
  
  In this setting, the maximum DGX-2 power consumption is 10 kW.
- Provides reduced but better performance than MaxQ when only 3 or 4 PSUs are working.
- If you switch to MaxQ mode, you can switch back to MaxP mode as follows:

```
$ sudo nvsm set powermode=maxp
```

The settings are preserved across reboots.
Chapter 3. CONFIGURING NVSM SECURITY

This chapter explains how to secure the NVSM API installation.

3.1. Overview of NVSM Security

NVSM APIs are served using the HTTPS protocol. HTTPS requires the NVSM API server to possess a public-private key pair as well as a certificate that it presents to connecting clients. The certificate also needs to be signed by a certificate authority (CA) using the private key of that CA.

For proper security, this certificate+key should be provided by users. It cannot be provided by NVIDIA because

- The private key should be known only to the user, and should not be known to NVIDIA, and
- NVIDIA is not a Certificate Authority

To allow the NVSM software stack to work right out of the box, the installation process creates some sample key pairs and certificates. These certificates are created with dummy values for country, organization, organization unit, etc. because the installation does not include these details. Also, the generated CA certificate is self signed. These sample certificates must NOT be used in a production environment.

NVSM allows you to provide your own key-pairs and certificates with correct values that are properly signed by a trusted CA. Details of key generation and certificate chains is beyond the scope of this document. However, an example setup is shown below to show how NVSM can be configured with customer provided/generated keys and certificates.

3.2. What You Need to Configure NVSM Security

To configure NVSM security, you need the following, either copied from a CA provider or generated locally and copied to a location on the system.
3.3. How to Configure NVSM Security

1. Edit the NVSM configuration file to use the paths and filenames of your certificate files and key file.

   Edit the `ca_cert`, `https_cert`, and `https_priv_key` configuration parameters to specify the path and filenames that NVSM shall use. The following use the example path and filenames.

   ```
   "ca_cert": "/pki/ca.crt",
   "https_cert": "/pki/node1.crt",
   "https_priv_key": "/pki/node1.key",
   ```

2. Restart the NVSM service.

   `$ sudo systemctl restart nvsm`
Chapter 4.
CALLING NVSM APIS

This section explains how to call NVSM APIs from your application, from the command shell, or from a browser. For a detailed list of NVSM APIs, refer to the NVSM API Reference.

4.1. Getting Authorization to Call NVSM APIs

Since NVSM APIs can be used to get detailed information about the platform as well as to perform potentially disruptive operations on the system (for example, RAID rebuild, firmware update, etc.), they are secured such that only the root user can authorize the use of these APIs.

**JSON Web Token (JWT):** Calling NVSM APIs require JWTs. NVSM provides a command to generate these tokens. The command requires root privileges to run. The resulting token must be passed in the HTTP request header as a bearer authorization token.

**HTTPS:** NVSM APIs are served over HTTPS instead of HTTP. HTTPS requires the API server to present the client with a certificate issued by a Certificate Authority (CA) that the client trusts. If you provisioned NVSM with genuine certificates as detailed in section Configuring NVSM Security, you will need to copy the CA certificate file to the client machine which intends to call NVSM APIs. If you do not take that step, you will get certificate validation errors and will need to ignore them.

The following command can be used to obtain a JWT. This command uses SSH to reach the DGX node (where NVSM is running) and runs the command `nvsm_apis` to generate a token with the specified validity.

**Syntax**

```
ssh -t <user>@<DGX Node IP> "sudo nvsm_apis -gentoken <days of validity>"
```
Example

```bash
apicaller@mylaptop$ ssh -t dgxadmin@dcl_dgx2_node1 "sudo nvsm_apis -gentoken 2"

dgxadmin@dcl_dgx2_node1's password:
[sudo] password for lab:
Token String:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZG1pbiI6dHJ1ZSwiZXhwIjoxNTMzNjk1MTc0fQ.WANIX6_vMdwf1B02oOOG_h4w3M8
Connection to dcl_dgx2_node1 closed.
apicaller@mylaptop$
```

4.2. Calling NVSM APIs Using cURL, Python, or a Browser

Using cURL

Syntax

```bash
```

Example

```bash
$ curl --cacert /pki/dcl_ca.crt -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZG1pbiI6dHJ1ZSwiZXhwIjoxNTMzNjk4NjY0fQ.vskms46o2D3vghygmd6c8VHFHLM7vcCknbbsyC04" https://dcl_dgx2_node1:273/redfish/v1/Systems | jq
```

- Replace `/pki/dcl_ca.crt` with the correct path on the client machine where the CA certificate file is copied.
- If you did not provision NVSM with genuine certificates, replace the option `--cacert <CA certificate file>` with `-k` to make cURL bypass certificate validation.
Using Python

Copy the following code and save as `restclient.py`. `restclient.py` requires a JWT as an argument.

```python
#!/usr/bin/python

# Syntax to run script:
# python ./restclient.py somevalidtoken

''' Sample client program to fetch systems info using NVSM API '''

import requests
import sys

if len(sys.argv) < 2:
    print("\nUsage : restclient.py <ip of node running nvsm-api service> <JWT tokenstring>\")
    exit()

r = requests.get('https://127.0.0.1:273/redfish/v1/Systems', verify="/pki/dc1_ca.crt", headers={'Authorization': 'Bearer ' + sys.argv[1]})
data = r.json()

print("API called : ", 'https://127.0.0.1:273/redfish/v1/Systems')
print("API Return Code : ", r)
print("API Return Payload : ", data)
```

The following is example usage.

```bash
~$ python ./restclient.py
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZG1pbiI6dHJ1ZSwiZXhwIjoxNTMzNjk4NjY0FQ.vskms46o2D3vzGhgyfmd6c8VRfHLM7vcCknbssyC04
#('API called : ', 'https://127.0.0.1:273/redfish/v1/Systems')
#('API Return Code : ', <Response [200]>)
#('API Return Payload : ', [u'@odata.type': u'/'redfish/v1/Systems', u'Name':
  u'Computer System Collection', u'Members': 1, u'@odata.id':
  u'#ComputerSystemCollection.ComputerSystemCollection', u'@odata.context':
  u'/redfish/v1/$metadata#ComputerSystemCollection.ComputerSystemCollection',
  u'Members': [(u'@odata.id': u'/redfish/v1/Systems/1')]])
~$
```

- Replace `/pki/dc1_ca.crt` with the correct path on the client machine where the CA certificate file is copied.
- If you did not provision NVSM with genuine certificates, replace the option `verify="/pki/dc1_ca.crt"` with `verify=false` to bypass certificate validation.

Using a Browser

When using the Chrome browser, you need a browser extension such as ModHeader to make the browser send the JWT in the request headers.

1. Install the extension.
   For example, ModHeader.
2. Add an HTTP request header.
   a) In the ModeHeader extension example, open the extension and then click "+" and select Request Header.
   b) In the Name Field, enter “Authorization” (without the quotes).
   c) In the Value Field, enter “Bearer <JWT string>” (without quotes and replace JWT string with the actual token string).
   Refer to https://mod-header.appspot.com/ for more details if required.

3. Import the CA certificate into the browser.
   a) For Chrome, navigate to Settings->Manage certificates (in the Advanced: Privacy and security section) to open the Certificates dialog.
   b) Click Import, then follow the Certificate Import Wizard to import the CA certificate (for example /pki/dcl_ca.crt) into the browser.

4. In the browser address bar, enter the URI that corresponds to any NVSM API and then press Enter.
   For example, enter https://<dgx-node-ip>:273/redfish/v1/Systems and then press Enter to get information about the DGX system.
Appendix A.
THIRD-PARTY LICENSES

This NVIDIA product contains third party software that is being made available to you under their respective open source software licenses. Some of those licenses also require specific legal information to be included in the product. This section provides such information.

mattn/go-sqlite3

The mattn/go-sqlite3 software (https://github.com/mattn/go-sqlite3) is provided under the following terms:

The MIT License (MIT)

Copyright (c) 2014 Yasuhiro Matsumoto

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Notice

THE INFORMATION IN THIS GUIDE AND ALL OTHER INFORMATION CONTAINED IN NVIDIA DOCUMENTATION REFERENCED IN THIS GUIDE IS PROVIDED “AS IS.” NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE INFORMATION FOR THE PRODUCT, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA’s aggregate and cumulative liability towards customer for the product described in this guide shall be limited in accordance with the NVIDIA terms and conditions of sale for the product.

THE NVIDIA PRODUCT DESCRIBED IN THIS GUIDE IS NOT FAULT TOLERANT AND IS NOT DESIGNED, MANUFACTURED OR INTENDED FOR USE IN CONNECTION WITH THE DESIGN, CONSTRUCTION, MAINTENANCE, AND/OR OPERATION OF ANY SYSTEM WHERE THE USE OR A FAILURE OF SUCH SYSTEM COULD RESULT IN A SITUATION THAT THREATENS THE SAFETY OF HUMAN LIFE OR SEVERE PHYSICAL HARM OR PROPERTY DAMAGE (INCLUDING, FOR EXAMPLE, USE IN CONNECTION WITH ANY NUCLEAR, AVIONICS, LIFE SUPPORT OR OTHER LIFE CRITICAL APPLICATION). NVIDIA EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR SUCH HIGH RISK USES. NVIDIA SHALL NOT BE LIABLE TO CUSTOMER OR ANY THIRD PARTY, IN WHOLE OR IN PART, FOR ANY CLAIMS OR DAMAGES ARISING FROM SUCH HIGH RISK USES.

NVIDIA makes no representation or warranty that the product described in this guide will be suitable for any specified use without further testing or modification. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to ensure the product is suitable and fit for the application planned by customer and to do the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer’s product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this guide. NVIDIA does not accept any liability related to any default, damage, costs or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this guide, or (ii) customer product designs.

Other than the right for customer to use the information in this guide with the product, no other license, either expressed or implied, is hereby granted by NVIDIA under this guide. Reproduction of information in this guide is permissible only if reproduction is approved by NVIDIA in writing, is reproduced without alteration, and is accompanied by all associated conditions, limitations, and notices.

Trademarks

NVIDIA and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation in the Unites States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2019 NVIDIA Corporation. All rights reserved.