SupremeRAID™ User Guide for Linux

July 2024

Graid Technology Inc.

Publication: July 5, 2024

CHANGE HISTORY

Revision No.	Date	Prepared by /Modified by	Description
1.0.0	July 5, 2024	Mia Lin	Initial Version

TABLE OF CONTENTS

INTRODUCTION

SupremeRAID™ is the most powerful, high-speed data protection solution specially designed for NVMe SSDs. SupremeRAID™ installs a virtual NVMe controller onto the operating system and integrates a high-performance, GPU-base PCIe RAID card into the system to manage the RAID operations of the virtual NVMe controller.

This document explains how to install the SupremeRAID™ software package for Linux and how to manage the RAID components using the command-line interface.

Software Module Overview

The SupremeRAID™ Software module has the following major components:

- graidctl — command-line management tool
- graid_server — management daemon that handles requests from graidctl to control the driver
- graid.ko — driver kernel module
- graid_core — GPU instance

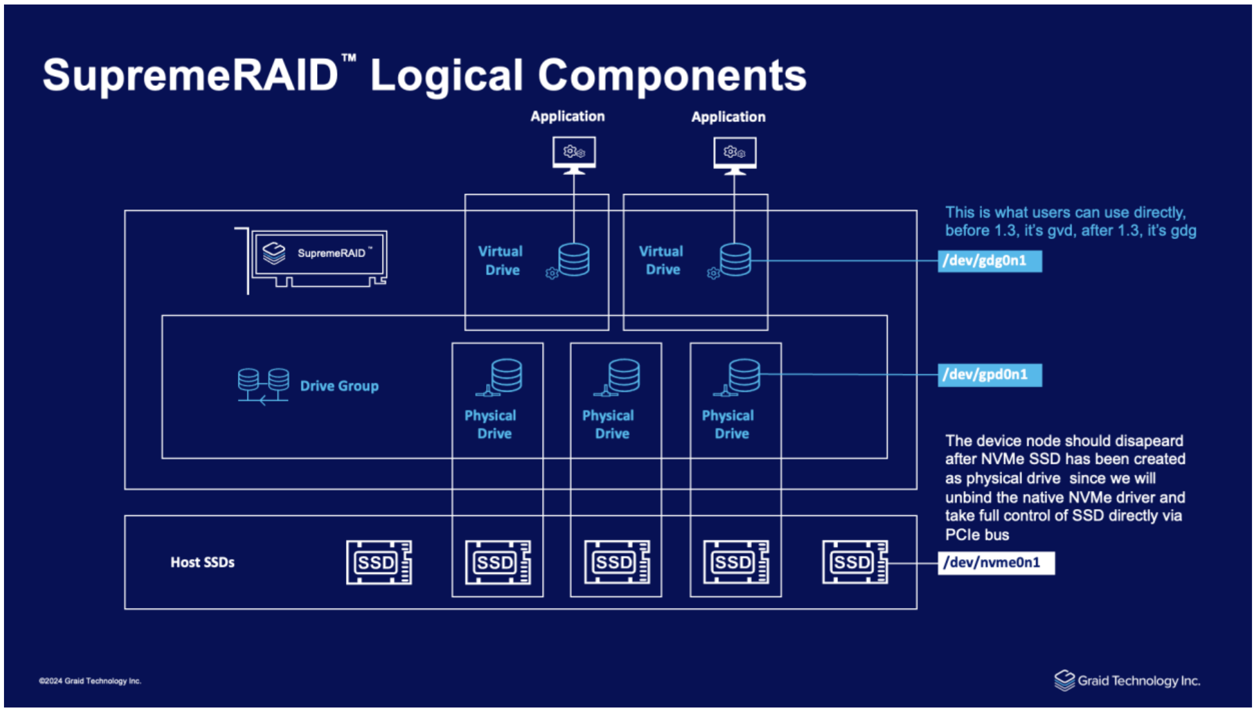
SupremeRAID™ Specification

SupremeRAID™ Driver Specifications	
Supported models:	SR-1000, SR-1010, SR-1001
Supported RAID levels:	RAID 0, 1, 5, 6, 10
Recommended minimum drive number for each RAID level:	RAID 0 : at least one drives RAID 1 : at least two drives RAID 5 : at least three drives RAID 6 : at least four drives RAID 10 : at least two drives
Maximum number of physical drives:	32
Maximum number of drive groups:	8
Maximum number of virtual drives per drive group:	1,023
Maximum size of the drive group:	Defined by the physical drive sizes
Configurable strip size (RAID0, RAID10)	4k, 8k, 16k, 32k, 64k,128k

RAID Components

SupremeRAID™ has three major RAID logical components:

- Physical Drive (PD)
- Drive Group (DG)
- Virtual Drive (VD)
- Controller (CX)


SupremeRAID™ Logical Components
Application
Application
SupremeRAID™
Virtual Drive
Virtual Drive
This is what users can use directly, before 1.3, it's gvd, after 1.3, it's gdg
/dev/gdg0n1
Drive Group
Physical Drive
Physical Drive
Physical Drive
/dev/gpd0n1
The device node should disapeard after NVMe SSD has been created as physical drive since we will unbind the native NVMe driver and take full control of SSD directly via PCIe bus
Host SSDs
SSD
/dev/nvme0n1
©2024 Graid Technology Inc.
Graid Technology Inc.

Physical Drive (PD)

Since NVMe drives are not directly attached to the SupremeRAID™ controller, you must tell the controller which SSDs can be managed. After an SSD is created as a physical drive, the SupremeRAID™ driver unbinds the SSD from the operating system, meaning the device node (/dev/nvmeX) disappears and is no longer accessible. At the same time, the SupremeRAID™ driver creates a corresponding device node (/dev/gpdX). You can check the SSD information, such as SSD model or SMART logs, using this device node. To control and access the SSD using /dev/nvmeXn1, you must first delete the corresponding physical drive.

SupremeRAID™ supports 32 physical drives, regardless of whether the physical drives are created from a native NVMe SSD, a drive connected through NVMe-oF, or a SAS/SATA disk.

Drive Group (DG)

The main component of RAID logic is a RAID group. When the drive group is created, the SupremeRAID™ driver initializes the physical drives with the corresponding RAID mode to ensure that the data and parity are synchronized.

There are two types of initialization processes.

- Fast Initialization: When all of the physical drives in the drive group (DG) support the de-allocate dataset management command, the SupremeRAID™ driver performs fast initialization by default, which optimizes the drive group state immediately.
- Background Initialization: Performance is slightly affected by the initialization traffic, but you can still create the virtual drive and access the virtual drive during a background initialization.

SupremeRAID™ supports eight drive groups, with a maximum of 32 physical drives in one drive group.

Virtual Drive (VD)

The virtual drive is equivalent to the RAID volume. You can create multiple virtual drives in the same drive group for multiple applications. The corresponding device node (/dev/gdgXnY) appears on the operating system when you create a virtual drive, and you can make the file system or running application directly on this device node. Currently, the SupremeRAID™ driver supports a maximum of 1023 virtual drives in each drive group.

Controller (CX)

The controller is the core component of the RAID system. It provides detailed hardware information such as GPU serial number, temperature, and fan speed. RAID management relies on the controller, so the controller's state directly affects the underlying drive group operations.

In the Linux driver, users can have dual controllers in the system and manage them separately. By enabling the high-availability function in a drive group, the backup controller will take over drive group management if the primary controller fails or goes missing. Additionally, you can set up drive groups on a specified controller or within the same NUMA node as the controller to minimize negative influences.

Note: If you upgrade from version 1.2.x to version 1.6.x of the graid driver, the device path changes from /dev/gvdXn1 to /dev/gdgXnY.

Features Overview

The SupremeRAID™ presents a range of features that facilitate convenient data storage methods and incorporate diverse protection mechanisms to ensure data integrity. The following will outline key features that contribute to achieving our objectives and fostering a foundational understanding of our services.

Ensuring Data Integrity with Consistency Checks

The SupremeRAID™ is designed to provide high reliability and data integrity levels. A key feature that enables this is the consistency check function.

The consistency check function allows administrators to ensure that the data stored on the SupremeRAID™ system is intact and uncorrupted. These checks can be performed on a regular schedule or manually initiated as needed. While running the consistency check, the system compares the data on each disk to identify any discrepancies or errors.

Depending on the settings chosen by the administrator, the consistency check function can either automatically fix any errors that are found or stop the check and alert the administrator to any detected errors. This feature provides administrators with flexibility and control over how the system responds to errors.

For detailed information about graid commands for the consistency check, see Using Consistency Checks to Ensure Data Integrity.

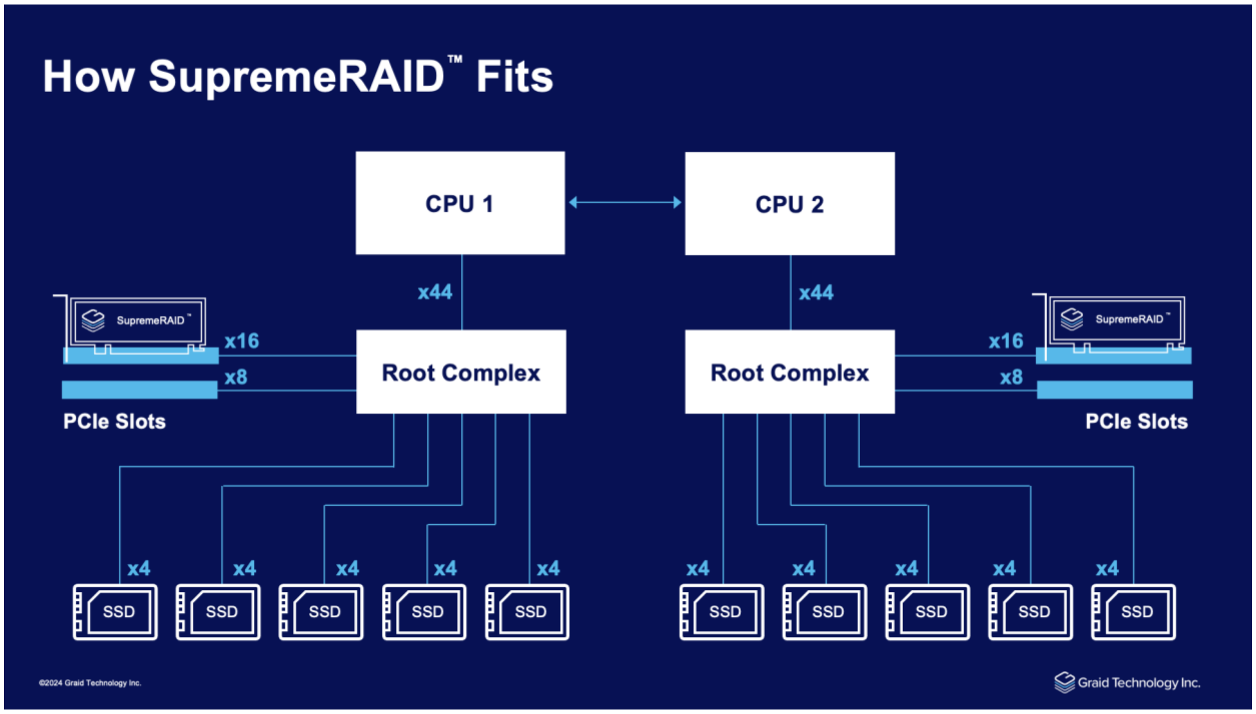
Note: The consistency check function is not supported on SupremeRAID™ systems configured in RAID0 mode because RAID0 does not provide data redundancy and does not require data consistency checks.

SupremeRAID™'s Dual-Controller Architecture for Auto-Failover and High- Availability

This feature enables the SupremeRAID™ system to automatically fail over to another SupremeRAID™ card when one SupremeRAID™ card experiences an issue without any interruption in service. This increased reliability and availability ensures that the system remains operational even in the event of a single card failure.

SupremeRAID™ supports dual-controller configurations in two modes: dual-active and active-passive. This enhances our RAID solution with comprehensive protection and security. Additionally, the high availability (HA) functionality remains unaffected by the root complex. Whether within the same root complex or across different root complexes, we have implemented failover mechanisms to ensure high availability.

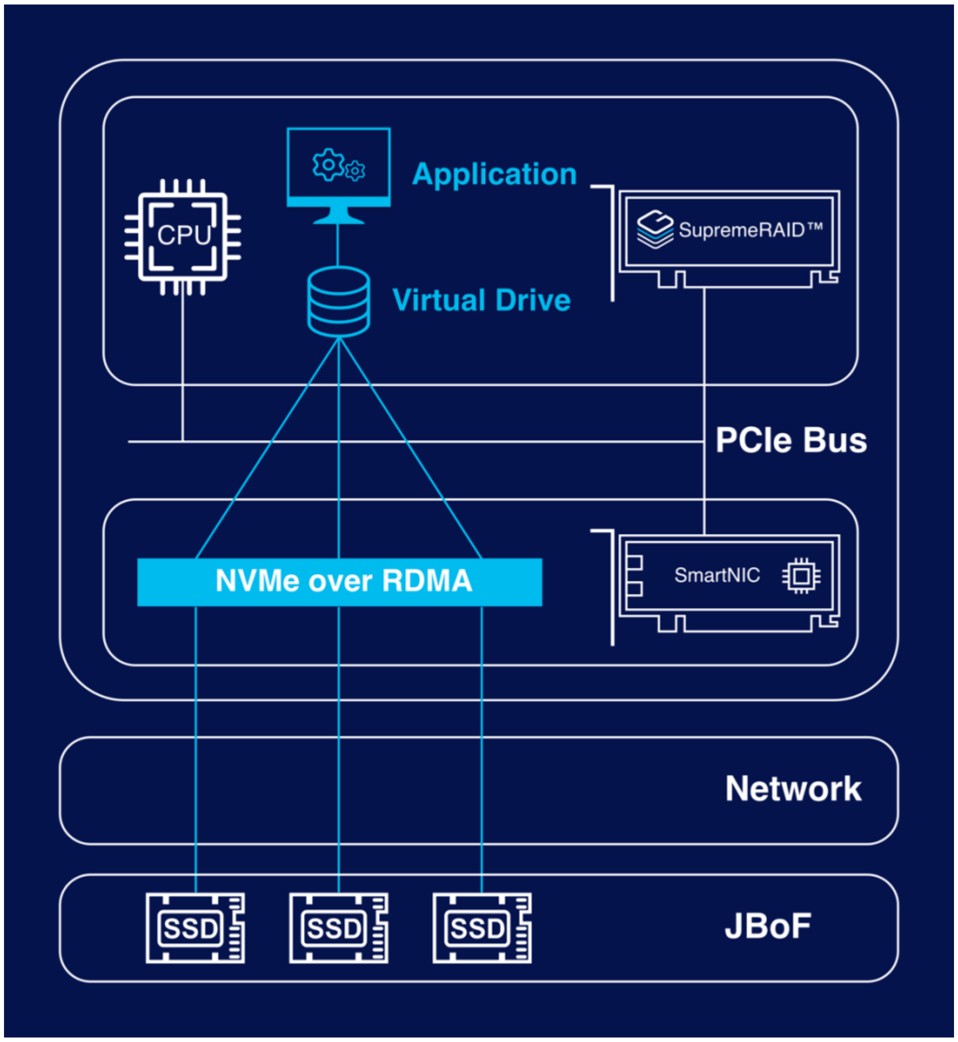
For detailed information about graid commands for setup dual-controller, please see Setting Up the Dual-Controller to Enable HA and Auto-Failover.



Setting Up the NVMe-oF Initiator Server and Managing Your RAID Components

The SupremeRAID™ allows you to easily manage a remote target server or storage pool that uses NVMe-over-Fabrics (NVMe- oF) technology. Both TCP and RDMA connections are supported, providing flexibility and compatibility with a wide range of systems. With the SupremeRAID™, you can create a virtual volume with RAID capabilities without the need for reconfiguration or re-cabling on the host server. This allows you to take advantage of the benefits of NVMe-oF, including increased capacity and improved data protection.

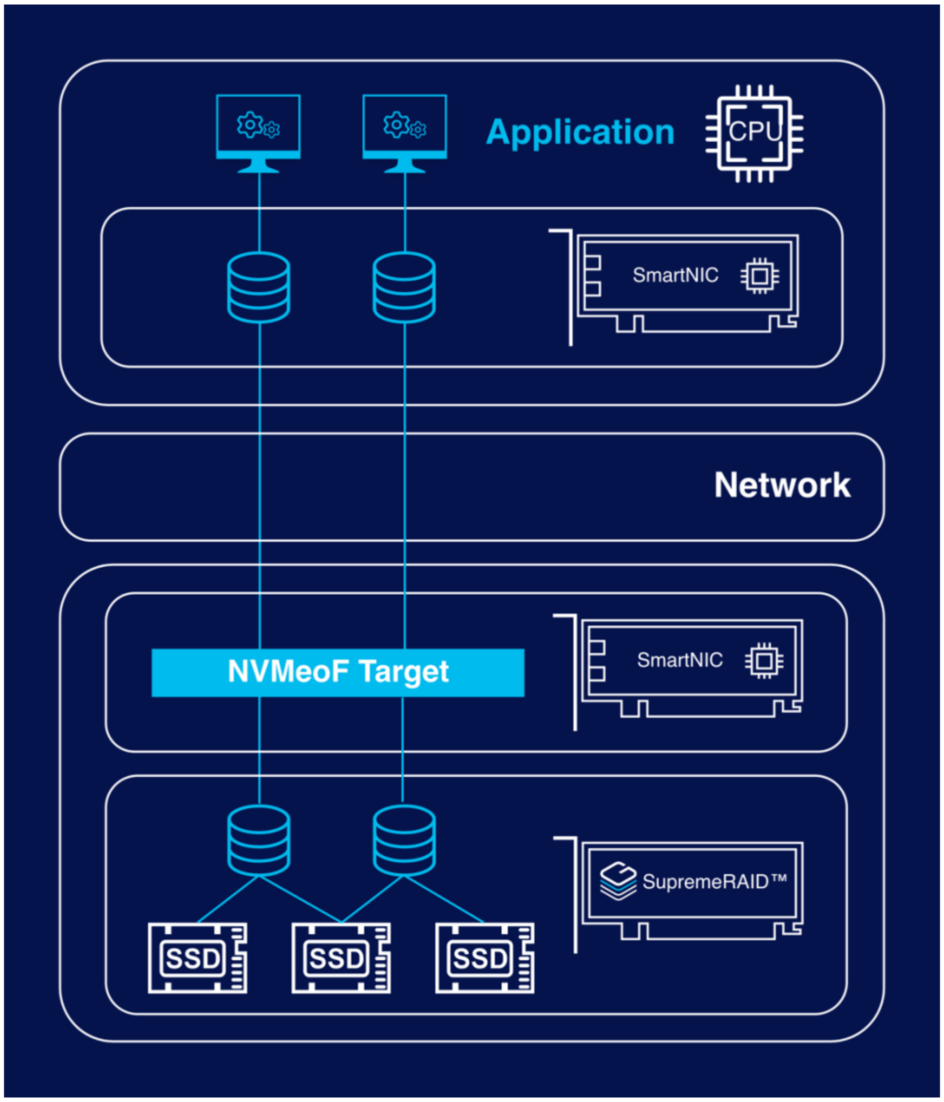
For detailed information about graid commands for the NVMe-oF initiator, see Managing Remote NVMe-oF Targets.



Sharing the SupremeRAID™ Volume as a NVMe-oF Target Server

The SupremeRAID™ allows you to easily compose local NVMe devices into a RAID array and share that array as an NVMe- over-Fabrics (NVMe-oF) target server. By using a SmartNIC to accelerate data transfer, you can achieve low latencies and high performance for your remote NVMe-oF clients.

For detailed information about graid commands for the NVMe-oF target, see Exporting NVMe-oF Target Management.

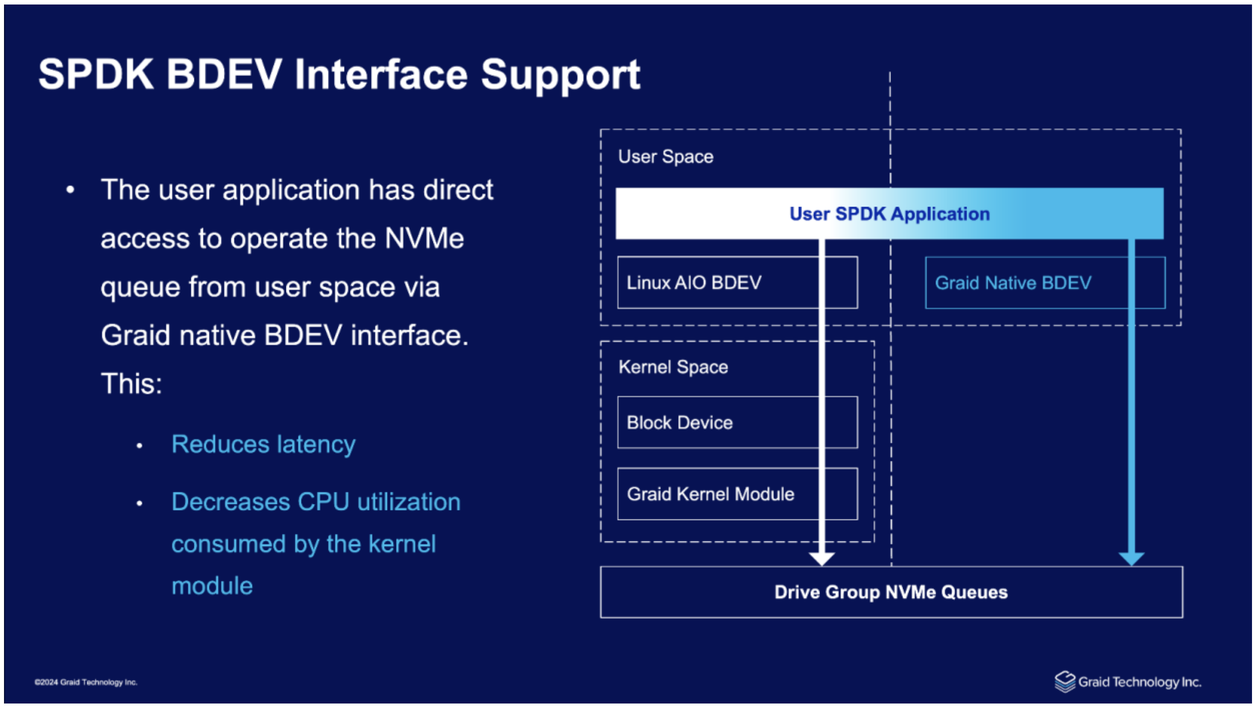


SPDK BDEV Feature of SupremeRAID™

The SupremeRAID™ software incorporates SPDK (Storage Performance Development Kit) feature, enabling direct access to operate the NVMe queue from user space through the SupremeRAID™ native BDEV (Block Device) interface. This integration offers significant benefits that enhance the overall performance and efficiency of the system.

The SPDK feature facilitates direct user application access to NVMe queues from user space. This minimizes data access and processing latency, resulting in enhanced system responsiveness through reduced overhead and fewer context switches. Moreover, this direct access eliminates the necessity for data transfers between user space and kernel space, thereby decreasing CPU utilization caused by kernel module activity. This optimization enables the CPU to prioritize crucial tasks, leading to improved overall system performance.

The SPDK feature in SupremeRAID™ contributes to an optimized storage solution, particularly in high-performance scenarios, where latency reduction and improved CPU utilization are crucial factors. By harnessing the power of SPDK, Graid ensures that users can maximize the potential of their NVMe devices while experiencing enhanced data processing capabilities with minimal overhead.



SPDK BDEV Interface Support

- The user application has direct access to operate the NVMe queue from user space via Graid native BDEV interface. This:
  - Reduces latency
  - Decreases CPU utilization consumed by the kernel module

Double Failure Protection with Distributed Journaling

SupremeRAID™ incorporates a distributed journaling mechanism specifically designed to safeguard data during abnormal shutdowns in double-failure scenarios. This system ensures data integrity by logging data in a dedicated journaling space before writing it to the storage area, any incomplete I/O operations are replayed upon service restart to maintain data consistency.

This journaling feature is automatically enabled in degraded mode to uphold data integrity. Additionally, users still have the flexibility to bypass journaling space reservation when creating a drive group.

For detailed information about graid commands for the modifying journal mode for RAID5 and RAID6 drive group, please see Modifying Journal Mode on a RAID-5 Drive Group.

SupremeRAID™ Graphical Management Console

To enhance the SupremeRAID™ management tool, we offer an intuitive graphical console. Users can effortlessly navigate through the console using the navigation bar, which includes sections for Dashboard, Hosts, RAID Management, Events, and Statistics to display system workloads. Additionally, administrators have access to Licenses, User Management, and Email Notification sections.

The system offers a comprehensive suite of features designed to enhance user experience and system management. The Dashboard and Statistics page provides an overview of system efficiency and health status, allowing users to monitor RAID utility performance and resource utilization. For hands-on management, the Host and RAID Management interface facilitates the conversion of storage devices into RAID resources.

Advanced features cater to administrator needs: the License Management function tracks SupremeRAID™ license status, while User Management allows for the creation and modification of user accounts with varying permission levels. To ensure timely alerts, administrators can configure SMTP settings in the Email Notification page and enable mail functions for specific users, thereby maintaining a robust notification system for critical events.

For detailed information about graid commands for the enabling UI Management console, please see Setup Graphical Management Console.

INSTALLATION

This section describes how to install the SupremeRAID™ hardware and software package for Linux operating systems.

Prerequisites

Before proceeding with the installation, make sure the system meets the following requirements:

- Minimum system requirements
  - CPU: 2 GHz or faster with at least 8 cores
  - RAM: 16 GB
  - Supported operating system: see Drivers & Documentation section on our website.
  - An available PCIe Gen3 or Gen4 x16 slot
- The SupremeRAID™ card must be installed into a PCIe x16 slot.
- The SupremeRAID™ software package, which includes the Pre-Installer and Installer, can be downloaded directly from the Graid Technology website. The Pre-Installer configures all necessary dependencies and environment settings automatically prior to installing the graid driver. The Installer contains the graid driver package and will automatically detect your Linux distributions and install the appropriate files.
- Make sure a SupremeRAID™-compatible SSD drive is being used. For a list of compatible drives, see the Drivers & Documentation section on our website.
- [OPTIONAL] The IOMMU function (AMD) or VT-d function (Intel) is recommend disabled in the system BIOS, typically found on the BIOS Advanced page.
- [OPTIONAL] It is highly recommended to disable the UEFI Secure Boot function on the BIOS security page, If UEFI Secure Boot is not applicable in your system, you will need to sign the NVIDIA Kernel Module. For further information and troubleshooting, please refer to the Nvidia website.

Note: To use virtualization services such as ESXi, you must enable the IOMMU (AMD) or VT-d (Intel) function. For more information, see ESXi Virtual Machine Support Using GPU Passthrough.

Installing the Hardware

ESD Warning

Electronic components and circuits are sensitive to ElectroStatic Discharge (ESD). When handling any circuit board assemblies including Connect Tech carrier assemblies, it is recommended that ESD safety precautions be observed. ESD safe best practices include, but are not limited to:

- Leaving circuit boards in their antistatic packaging until they are ready to be installed.
- Using a grounded wrist strap when handling circuit boards, at a minimum you should touch a grounded metal object to dissipate any static charge that may be present on you.
- Only handling circuit boards in ESD safe areas, which may include ESD floor and table mats, wrist strap stations and ESD safe lab coats.
- Avoiding handling circuit boards in carpeted areas.
- Try to handle the board by the edges, avoiding contact with components.

Installation Procedure

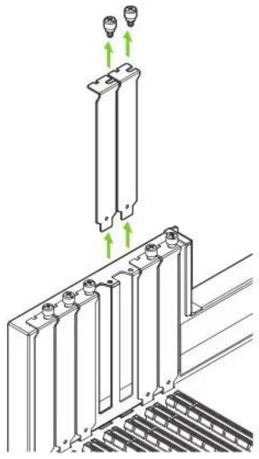
Perform the following procedure to install SupremeRAID™ into your system.

Step 1 Power down your system.

Step 2 Unplug the power cord from the AC power source.

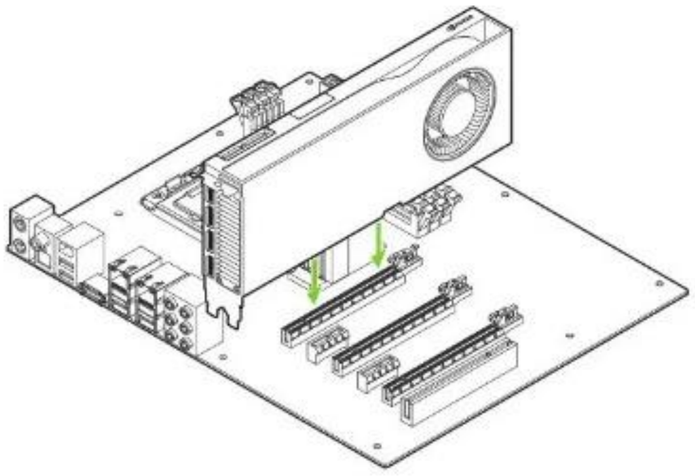
Step 3 Remove the side panel from your system to gain access to the motherboard.

Step 4 If your system has a PCIe card, remove it. If a retention bar is holding the card in place, remove the screw securing the card. If there is no existing PCIe card, remove the access covers from the primary x16 PCI express slot.



Note: The SupremeRAID™ SR-1010 is dual-slot card and requires you to remove two adjacent slot covers. The SupremeRAID™ SR-1000 and SR-1001 are single slot cards and require only a single- slot.

Step 5 Install the card into the primary x16 PCI Express slot. Press gently on the card until it is seated securely in the slot and reattach the SupremeRAID™ card bracket retention mechanism.



Note: Install the SupremeRAID™ card into the primary x16 PCI Express slot. The SupremeRAID™ SR-1010 is dual-slot card and covers the adjacent slot. The SupremeRAID™ SR-1000 and SupremeRAID™ SR-1001 are single-slot cards. For more information, see NVIDIA RTX Ampere Architecture-Based Graphics Card User Guide.

Step 6 Secure the card to the system frame using the screw(s) you removed in step 4.

Step 7 Install the side panel you removed in step 3.

Installing the Software Driver

The recommended and quickest way to install the graid software is by using the pre-installer scripts and installer (described below).

However, if you prefer to install the software manually or your environment lacks Internet access, follow the manual installation procedure to configure the environment settings and install the Graid driver manually. If you have already installed the software and only wish to upgrade it, please refer to the instructions for the upgrade configuration.

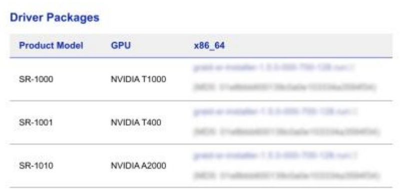
Using the Pre-installer and Installer

The graid pre-installer is an executable file that contains the required dependencies and a setup script that installs the NVIDIA driver. The script makes it easy to prepare the environment and install the SupremeRAID™ driver in every supported Linux distribution. Use the following steps to prepare the environment and install the SupremeRAID™ driver using the pre-installer in supported Linux distributions.

Note: To run the pre-installer, the system must have internet access to download the required dependencies from the official mirror.

Step 1 Go to the Graid Technology website to download the latest version of the pre-installer and make it executable, please download the package in Drivers & Documentation.

```
$ sudo chmod +x [Filename]
```

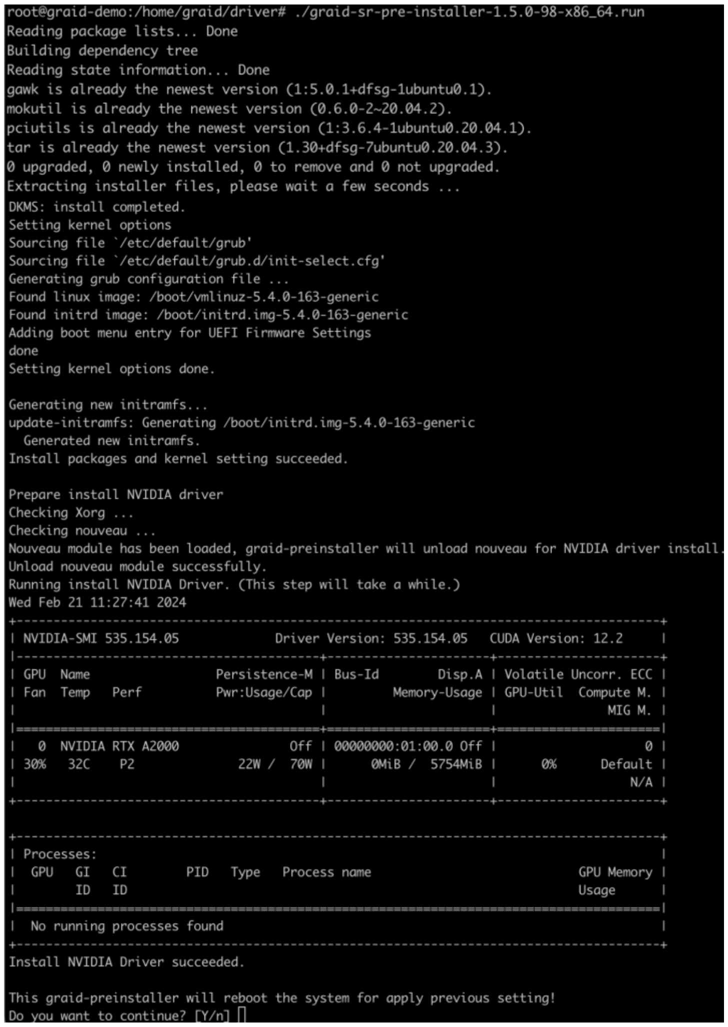


Driver Packages

Product Model	GPU	x86_64
SR-1000	NVIDIA T1000	[illegible]
SR-1001	NVIDIA T400	[illegible]
SR-1010	NVIDIA A2000	[illegible]

Step 2 Execute the pre-installer and follow the instructions to complete the pre-installation process, as shown in the following figure.

```
$ sudo ./[filename]
```

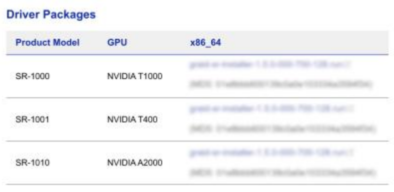


```
root@graid-demo:/home/graid/driver# ./graid-sr-pre-installer-1.5.0-98-x86_64.run
Reading package lists... Done
Building dependency tree
Reading state information... Done
gawk is already the newest version (1:5.0.1+dfsg-1ubuntu0.1).
mokutil is already the newest version (0.6.0-2~20.04.2).
pciutils is already the newest version (1:3.6.4-1ubuntu0.20.04.1).
tar is already the newest version (1.30+dfsg-7ubuntu0.20.04.3).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Extracting installer files, please wait a few seconds ...
DKMS: install completed.
Setting kernel options
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.4.0-163-generic
Found initrd image: /boot/initrd.img-5.4.0-163-generic
Adding boot menu entry for UEFI Firmware Settings
done
Setting kernel options done.

Generating new initramfs...
update-initramfs: Generating /boot/initrd.img-5.4.0-163-generic
  Generated new initramfs.
Install packages and kernel setting succeeded.

Prepare install NVIDIA driver
Checking Xorg ...
Checking nouveau ...
Nouveau module has been loaded, graid-preinstaller will unload nouveau for NVIDIA driver install.
Unload nouveau module successfully.
Running install NVIDIA Driver. (This step will take a while.)
Wed Feb 21 11:27:41 2024
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.154.05             Driver Version: 535.154.05   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  NVIDIA RTX A2000               Off | 00000000:01:00.0 Off |                    0 |
| 30%   32C    P2              22W /  70W |      0MiB /  5754MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|  No running processes found                                                           |
+---------------------------------------------------------------------------------------+
Install NVIDIA Driver succeeded.

This graid-preinstaller will reboot the system for apply previous setting!
Do you want to continue? [Y/n] 
```

Step 3 After running the pre-installation script, type Y when prompted to reboot the system.

Step 4 Go to the Graid Technology website, download the latest version of the installer in Drivers & Documentation and make it executable.

```
$ sudo chmod +x [filename]
```



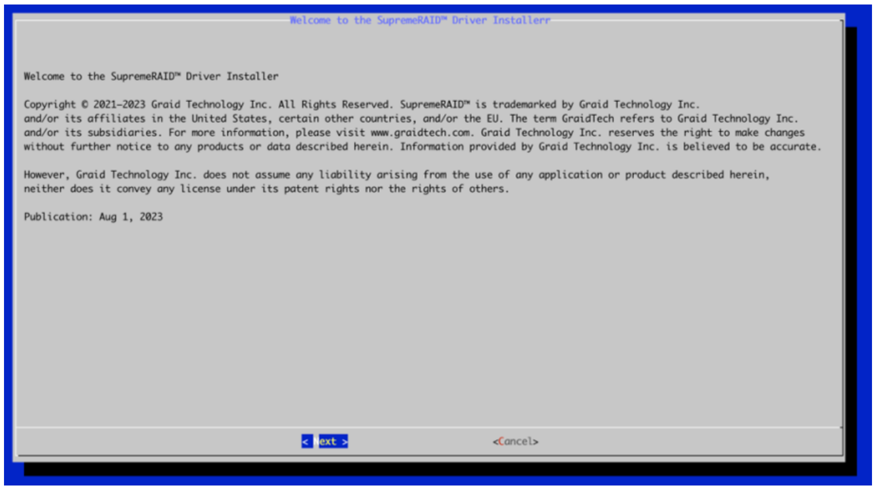
Driver Packages

Product Model	GPU	x86_64
SR-1000	NVIDIA T1000	[illegible]
SR-1001	NVIDIA T400	[illegible]
SR-1010	NVIDIA A2000	[illegible]

Step 5 Execute the installer and follow the provided steps to complete the installation.

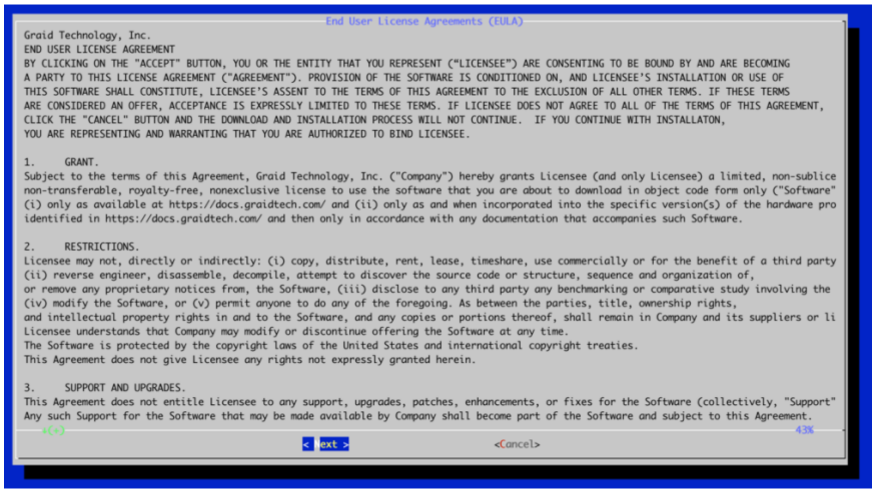
```
$ sudo ./[filename]
```

A At the Welcome page, select Next and click Enter to view the end-user license agreement.



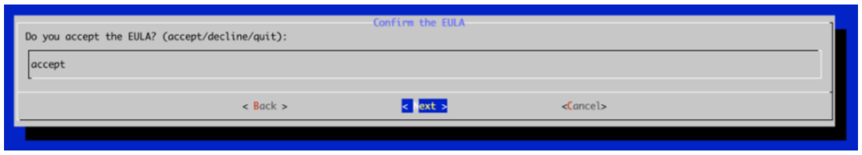
```
Welcome to the SupremeRAID™ Driver Installerr

Welcome to the SupremeRAID™ Driver Installer

Copyright © 2021-2023 Graid Technology Inc. All Rights Reserved. SupremeRAID™ is trademarked by Graid Technology Inc.
and/or its affiliates in the United States, certain other countries, and/or the EU. The term GraidTech refers to Graid Technology Inc.
and/or its subsidiaries. For more information, please visit www.graidtech.com. Graid Technology Inc. reserves the right to make changes
without further notice to any products or data described herein. Information provided by Graid Technology Inc. is believed to be accurate.

However, Graid Technology Inc. does not assume any liability arising from the use of any application or product described herein,
neither does it convey any license under its patent rights nor the rights of others.

Publication: Aug 1, 2023

                              < Next >                  <Cancel>
```

B In the end-user license agreement page, use the spacebar to scroll down the content. After you review the license, select Next and click Enter.



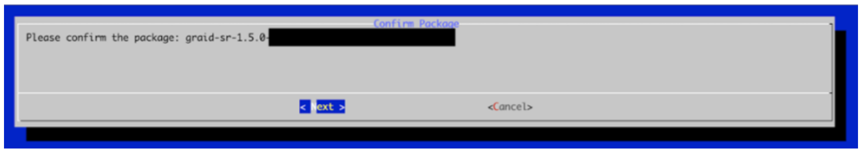
```
End User License Agreements (EULA)
Graid Technology, Inc.
END USER LICENSE AGREEMENT
BY CLICKING ON THE "ACCEPT" BUTTON, YOU OR THE ENTITY THAT YOU REPRESENT ("LICENSEE") ARE CONSENTING TO BE BOUND BY AND ARE BECOMING
A PARTY TO THIS LICENSE AGREEMENT ("AGREEMENT"). PROVISION OF THE SOFTWARE IS CONDITIONED ON, AND LICENSEE'S INSTALLATION OR USE OF
THIS SOFTWARE SHALL CONSTITUTE, LICENSEE'S ASSENT TO THE TERMS OF THIS AGREEMENT TO THE EXCLUSION OF ALL OTHER TERMS. IF THESE TERMS
ARE CONSIDERED AN OFFER, ACCEPTANCE IS EXPRESSLY LIMITED TO THESE TERMS. IF LICENSEE DOES NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT,
CLICK THE "CANCEL" BUTTON AND THE DOWNLOAD AND INSTALLATION PROCESS WILL NOT CONTINUE.  IF YOU CONTINUE WITH INSTALLATON,
YOU ARE REPRESENTING AND WARRANTING THAT YOU ARE AUTHORIZED TO BIND LICENSEE.

1.      GRANT.
Subject to the terms of this Agreement, Graid Technology, Inc. ("Company") hereby grants Licensee (and only Licensee) a limited, non-sublice
non-transferable, royalty-free, nonexclusive license to use the software that you are about to download in object code form only ("Software"
(i) only as available at https://docs.graidtech.com/ and (ii) only as and when incorporated into the specific version(s) of the hardware pro
identified in https://docs.graidtech.com/ and then only in accordance with any documentation that accompanies such Software.

2.      RESTRICTIONS.
Licensee may not, directly or indirectly: (i) copy, distribute, rent, lease, timeshare, use commercially or for the benefit of a third party
(ii) reverse engineer, disassemble, decompile, attempt to discover the source code or structure, sequence and organization of,
or remove any proprietary notices from, the Software, (iii) disclose to any third party any benchmarking or comparative study involving the
(iv) modify the Software, or (v) permit anyone to do any of the foregoing. As between the parties, title, ownership rights,
and intellectual property rights in and to the Software, and any copies or portions thereof, shall remain in Company and its suppliers or li
Licensee understands that Company may modify or discontinue offering the Software at any time.
The Software is protected by the copyright laws of the United States and international copyright treaties.
This Agreement does not give Licensee any rights not expressly granted herein.

3.      SUPPORT AND UPGRADES.
This Agreement does not entitle Licensee to any support, upgrades, patches, enhancements, or fixes for the Software (collectively, "Support"
Any such Support for the Software that may be made available by Company shall become part of the Software and subject to this Agreement.
  ↓(+)                                                                                                                            43%
                              < Next >                  <Cancel>
```

C Type accept, click tab, select next, and click enter to accept the license agreement.

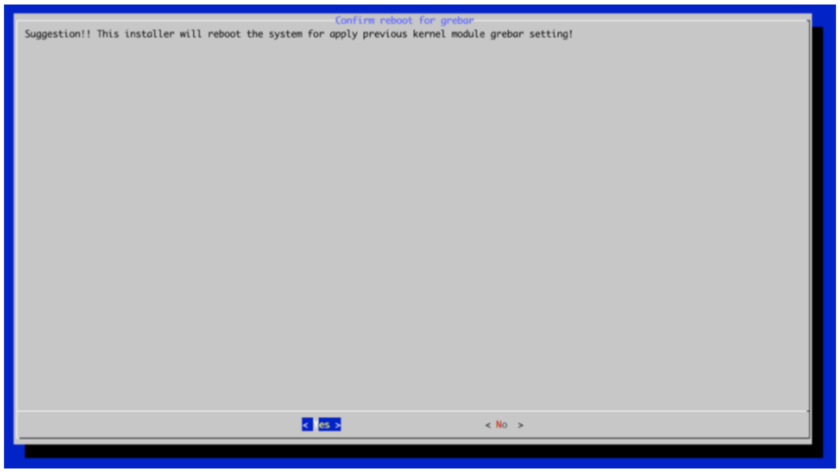


```
Confirm the EULA
Do you accept the EULA? (accept/decline/quit):
accept

                  < Back >          < Next >          <Cancel>
```

D Confirm the installation package, and then Click Next to continue with the installation.


Confirm Package
Please confirm the package: graid-sr-1.5.0-[redacted]

E Complete the installation, and the installer will reboot system.


Confirm reboot for grebar
Suggestion!! This installer will reboot the system for apply previous kernel module grebar setting!

Step 6 To activate the software, apply the SupremeRAID™ license key.

```
$ sudo graidctl apply license [LICENSE_KEY]
```

Using Installer for Silent Installation

This section is designed for users who require mass deployment and may be designing scripts for installation. However, we strongly recommend using the GUI installation process for the best user experience and comprehensive configuration options.

Step 1 Please follow the steps from the previous section to download the pre-installer and installer, make them executable, and use the pre-installer to install the dependencies required by the SupremeRAID™ service.

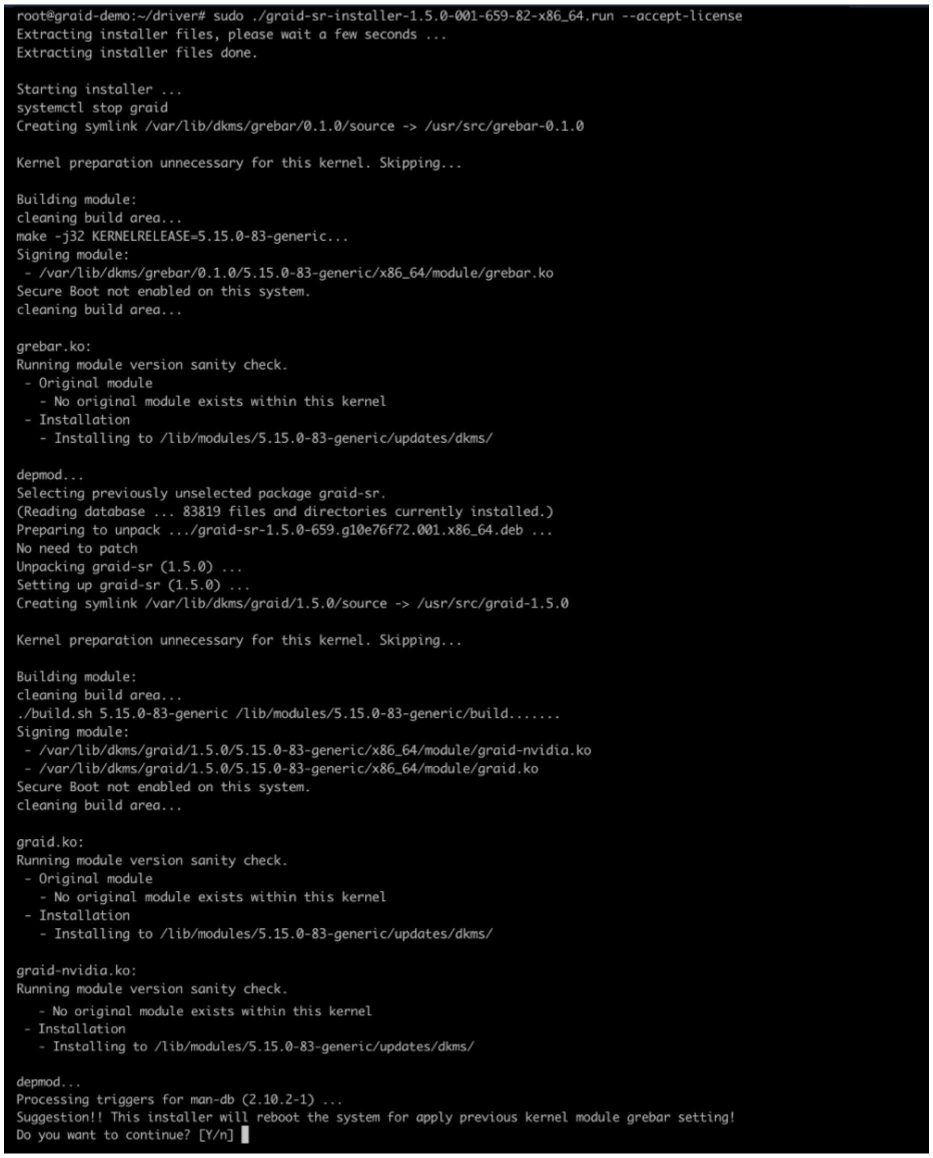
```
$ sudo chmod +x [filename]
```

Step 2 To install pre-installer without interactive mode, add “--yes” while executing the pre-installer.

```
$ sudo ./[filename] --yes
```

Step 3 To install the driver directly with the ULAs license acceptance, simply add the command '--accept-license' in the end when executing the installer.

```
$ sudo ./[filename] --accept-license
```



```
root@graid-demo:~/driver# sudo ./graid-sr-installer-1.5.0-001-659-82-x86_64.run --accept-license
Extracting installer files, please wait a few seconds ...
Extracting installer files done.

Starting installer ...
systemctl stop graid
Creating symlink /var/lib/dkms/grebar/0.1.0/source -> /usr/src/grebar-0.1.0

Kernel preparation unnecessary for this kernel. Skipping...

Building module:
cleaning build area...
make -j32 KERNELRELEASE=5.15.0-83-generic...
Signing module:
 - /var/lib/dkms/grebar/0.1.0/5.15.0-83-generic/x86_64/module/grebar.ko
Secure Boot not enabled on this system.
cleaning build area...

grebar.ko:
Running module version sanity check.
 - Original module
   - No original module exists within this kernel
 - Installation
   - Installing to /lib/modules/5.15.0-83-generic/updates/dkms/

depmod...
Selecting previously unselected package graid-sr.
(Reading database ... 83819 files and directories currently installed.)
Preparing to unpack .../graid-sr-1.5.0-659.g10e76f72.001.x86_64.deb ...
No need to patch
Unpacking graid-sr (1.5.0) ...
Setting up graid-sr (1.5.0) ...
Creating symlink /var/lib/dkms/graid/1.5.0/source -> /usr/src/graid-1.5.0

Kernel preparation unnecessary for this kernel. Skipping...

Building module:
cleaning build area...
./build.sh 5.15.0-83-generic /lib/modules/5.15.0-83-generic/build.......
Signing module:
 - /var/lib/dkms/graid/1.5.0/5.15.0-83-generic/x86_64/module/graid-nvidia.ko
 - /var/lib/dkms/graid/1.5.0/5.15.0-83-generic/x86_64/module/graid.ko
Secure Boot not enabled on this system.
cleaning build area...

graid.ko:
Running module version sanity check.
 - Original module
   - No original module exists within this kernel
 - Installation
   - Installing to /lib/modules/5.15.0-83-generic/updates/dkms/

graid-nvidia.ko:
Running module version sanity check.
   - No original module exists within this kernel
 - Installation
   - Installing to /lib/modules/5.15.0-83-generic/updates/dkms/

depmod...
Processing triggers for man-db (2.10.2-1) ...
Suggestion!! This installer will reboot the system for apply previous kernel module grebar setting!
Do you want to continue? [Y/n]
```

Step 4 To activate the software, apply the SupremeRAID™ license key.

```
$ sudo graidctl apply license [LICENSE_KEY]
```

Manual Installation

The following procedure describes how to manually install the Graid software on various operating systems. The reference for packages and dependencies for each operating system is provided below.

- For CentOS, Rocky Linux, AlmaLinux, and RHEL operating systems.
- For Ubuntu operating systems.
- For openSUSE operating systems.
- For SLES operating systems,.

Note: For systems without internet access, download required dependencies from official repositories. See the distribution section below for details. Only perform manual installation if necessary or if the pre-installer fails. For most cases, check Supported Operating Systems on our website and use the automated pre-installer script to install the graid software.

Dependency Table for Manual Installation

Here is the dependency tree for manual installation and the comparison table for each operating system.

RHEL	CentOS/Rocky/ Almalinux/Oracle	SLES	Debian/Ubuntu
automake	automake	automake	automake
dialog	dialog	dialog	dialog
dkms	dkms	dkms	dkms
gcc	gcc	gcc	gcc
ipmitool	ipmitool	ipmitool	ipmitool
make	make	make	make
mdadm	mdadm	mdadm	mdadm
mokutil	mokutil	mokutil	mokutil
pciutils	pciutils	pciutils	pciutils
tar	tar	tar	tar
vim	vim	vim	vim

wget	wget	wget	wget
sg3_utils	sg3_utils	libsgutils-devel	libsgutils2-2
--	--	libpci3	libpci3
--	--	libpci3	libpci3
sqlite-libs	sqlite-libs	sqlite3	sqlite3
--	--	libudev-devel	--
--	--	--	initramfs-tools
--	--	--	gawk
gcc-c++-$(VERSION_ID)	gcc-c++	g++	g++
gcc-$(VERSION_ID)	--	--	--
kernel-devel-$(kernel_version)	kernel-devel-$(kernel_version)	--	--
kernel-headers-$(kernel_version)	kernel-headers-$(kernel_version)	-C kernel-default-devel=$(kernel_version_suse)	linux-headers-$(kernel_version)

Note: To determine the kernel version for RHEL, you can use the command **uname -r**. For SUSE, extract the kernel version using **uname -r | awk -F"-default" '{print $1}'**. Additionally, please using **awk -F'=' '/VERSION_ID/{ gsub(/"/,""); print $2}' /etc/os-release** to retrieve the version ID.

Manual Installation on a CentOS, Rocky Linux, AlmaLinux, and RHEL Operating Systems

Graid Technology, Inc. recommends referring to Supported Operating Systems on our website and using the pre-installer to configure the environmental settings.

Step 1 Install the package dependencies and build for Dynamic Kernel Module Support (DKMS) based on your operating system.

- For CentOS, Rocky Linux, and AlmaLinux: issue the following commands.

```
$ sudo yum install --enablerepo=extras epel-release

$ sudo yum install vim wget make automake gcc gcc-c++ kernel-devel
kernel-headers kernel dkms ipmitool tar mdadm sg3_utils sqlite-libs
automake dialog
```

- For RHEL8, issue the following commands:

```
$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-
latest-8.noarch.rpm

$ sudo yum install vim wget make automake kernel-devel-$(uname -r)
kernel-headers-$(uname -r) dkms gcc gcc-c++ ipmitool tar mdadm
sg3_utils sqlite-libs automake dialog
```

- For RHEL7.9: issue the following commands.

```
$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-
latest-7.noarch.rpm

$ sudo yum install gcc-$(awk -F'=' '/VERSION_ID/{ gsub(/"/,""); print
$2}' /etc/os-release) gcc-c++-$(awk -F'=' '/VERSION_ID/{ gsub(/"/,"");
print $2}' /etc/os-release)

$ sudo yum install vim wget make automake kernel-devel-$(uname -r)
kernel-headers-$(uname -r) dkms ipmitool tar mdadm sg3_utils sqlite-
libs automake dialog
```

Step 2 Add the kernel option. This step prevents the Nouveau driver from loading during installation and disables IOMMU in the system BIOS.

```
$ sudo vim /etc/default/grub
```

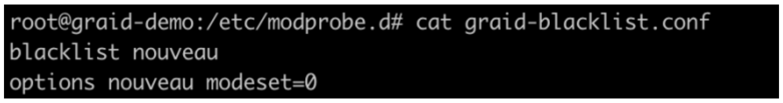
Step 3 Append the command line parameters and then update the grub configuration based on your operating system.

- For RHEL8, append iommu=pt and 'nvme_core.multipath=Y' to GRUB_CMDLINE_LINUX_DEFAULT.
- For RHEL7.9, append iommu=pt to 'GRUB_CMDLINE_LINUX_DEFAULT'.

```
$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

Step 4 Append **blacklist nouveau** and **options nouveau modeset=0** to the end of the /etc/modprobe.d/graid- blacklist.conf file to disable the Nouveau driver and update initramfs.

```
$ sudo update-initramfs -u
```



```
root@graid-demo:/etc/modprobe.d# cat graid-blacklist.conf
blacklist nouveau
options nouveau modeset=0
```

- For CentOS, Rocky Linux, and AlmaLinux: Find the latest version of the kernel and assign it to –kver.

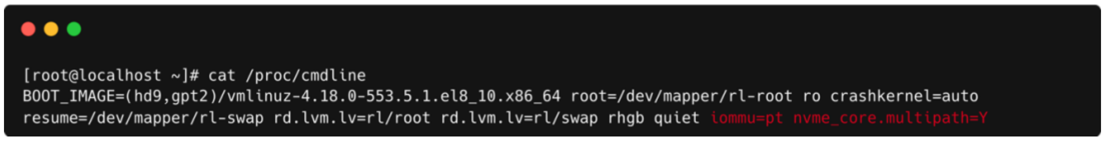
```
$ sudo dracut -f --kver `rpm -qa | grep kernel-headers | awk -
F'kernel-headers-' {'print $2'}`
```

- For RHEL: issue the following command.

```
$ sudo dracut -f
```

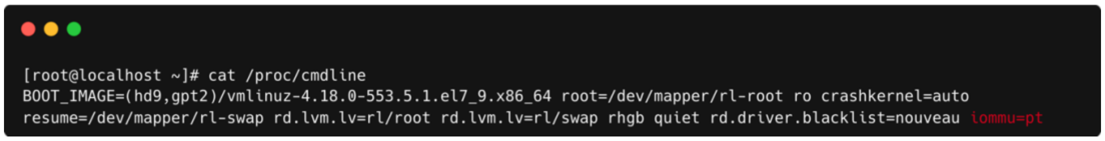
Step 5 Reboot the system and make sure the grub configuration was applied. You can check /proc/cmdline for the grub configuration in use. For example:

- For RHEL8:



```
[root@localhost ~]# cat /proc/cmdline
BOOT_IMAGE=(hd9,gpt2)/vmlinuz-4.18.0-553.5.1.el8_10.x86_64 root=/dev/mapper/rl-root ro crashkernel=auto
resume=/dev/mapper/rl-swap rd.lvm.lv=rl/root rd.lvm.lv=rl/swap rhgb quiet iommu=pt nvme_core.multipath=Y
```

- For RHEL7:



```
[root@localhost ~]# cat /proc/cmdline
BOOT_IMAGE=(hd9,gpt2)/vmlinuz-4.18.0-553.5.1.el7_9.x86_64 root=/dev/mapper/rl-root ro crashkernel=auto
resume=/dev/mapper/rl-swap rd.lvm.lv=rl/root rd.lvm.lv=rl/swap rhgb quiet rd.driver.blacklist=nouveau iommu=pt
```

Step 6 Install the NVIDIA driver.

```
$ wget https://us.download.nvidia.com/XFree86/Linux-
x86_64/550.67/NVIDIA-Linux-x86_64-550.67.run

$ chmod +x ./NVIDIA-Linux-x86_64-550.67.run
```

- For CentOS: Use the latest version of kernel-headers to install the NVIDIA driver.

```
$ sudo ./NVIDIA-Linux-x86_64-550.67.run -s --no-systemd --no-opengl-
files --no-nvidia-modprobe --dkms -k `rpm -qa | grep kernel-headers |
awk -F'kernel-headers-' {'print $2'}`
```

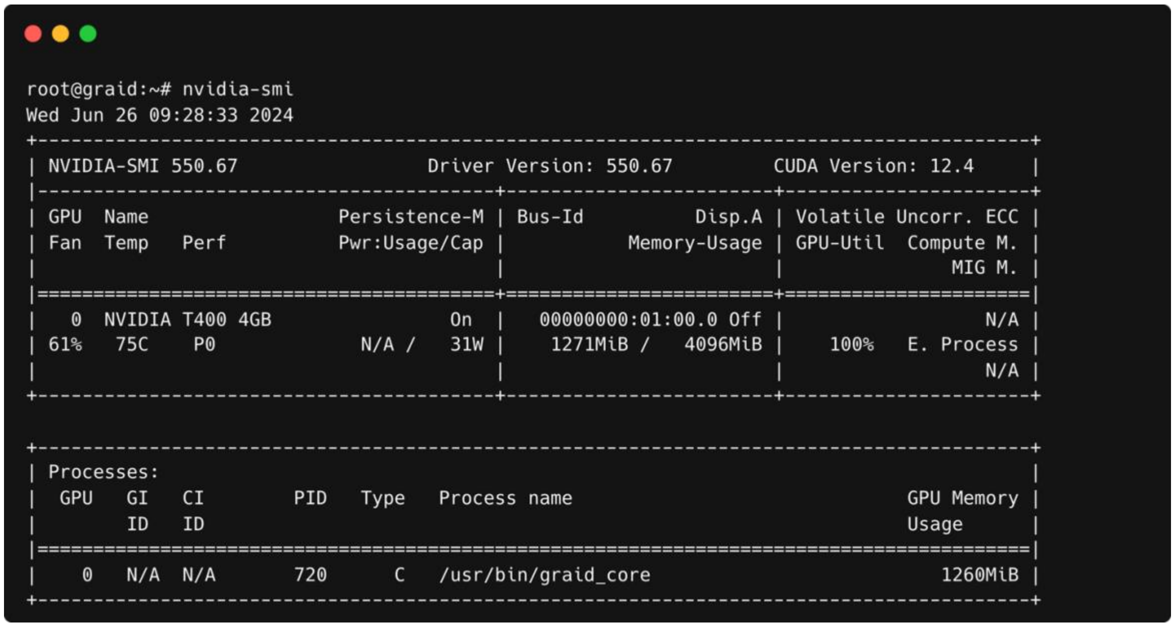
- For RHEL:

```
$ sudo ./NVIDIA-Linux-x86_64-550.67.run -s --no-systemd --no-opengl-
files --no-nvidia-modprobe --dkms -k `rpm -qa | grep kernel-headers |
awk -F'kernel-headers-' {'print $2'}`
```

Step 7 The Nouveau driver is now disabled. Reboot and install the NVIDIA driver before proceeding with the installation.

```
$ sudo reboot
```

Step 8 Use the **nvidia-smi** command to confirm that the NVIDIA GPU is working. The following figure shows an output example of a successful installation.



```
root@graid:~# nvidia-smi
Wed Jun 26 09:28:33 2024
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.67                 Driver Version: 550.67         CUDA Version: 12.4     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA T400 4GB                On  |   00000000:01:00.0 Off |                  N/A |
| 61%   75C    P0             N/A /   31W |    1271MiB /   4096MiB |    100%   E. Process |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI        PID   Type   Process name                              GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|    0   N/A  N/A       720      C   /usr/bin/graid_core                          1260MiB |
+-----------------------------------------------------------------------------------------+
```

Step 9 From the Graid Technology website, download the latest version of the installer and make it executable.

```
$ sudo chmod +x [filename]
```

Step 10 Proceed to Executing the Installer and Completing the Installation to execute the installer and to complete the installation.

Manual Installation on an Ubuntu Operating System

Step 1 Graid Technology, Inc. recommends referring to Supported Operating Systems on our website and using the pre-installer to configure the environmental settings.

Step 2 Install the package dependencies and build for DKMS.

```
$ sudo apt-get update

$ sudo apt-get install make automake gcc g++ linux-headers-$(uname -r)
dkms ipmitool initramfs-tools tar mdadm libsgutils2-2 libudev-dev
libpci3 sqlite automake dialog
```

Step 3 Disable Ubuntu daily upgrade.

```
$ sed -i '/Unattended-Upgrade "1"/ s/"1"/"0"/'
/etc/apt/apt.conf.d/20auto-upgrades

$ sed -i '/Update-Package-Lists "1"/ s/"1"/"0"/'
/etc/apt/apt.conf.d/20auto-upgrades
```

Step 4 Add the kernel option. This step prevents the Nouveau driver from loading during installation and disables IOMMU in the system BIOS.

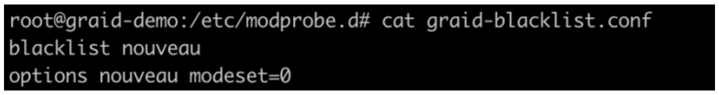
```
$ sudo vim /etc/default/grub
```

Step 5 Append **iommu=pt** and **nvme_core.multipath=Y** to GRUB_CMDLINE_LINUX_DEFAULT, and then update the grub configuration.

```
$ sudo update-grub
```

Step 6 Append **blacklist nouveau** and **options nouveau modeset=0** to the end of the /etc/modprobe.d/graid-blacklist.conf file to disable the Nouveau driver and update initramfs.

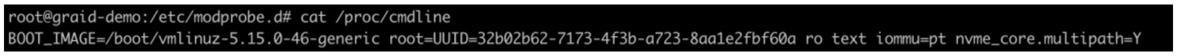
```
$ sudo update-initramfs -u
```



```
root@graid-demo:/etc/modprobe.d# cat graid-blacklist.conf
blacklist nouveau
options nouveau modeset=0
```

Note: You might need to manually create the /etc/modprobe.d/graid-blacklist.conf file and append **blacklist nouveau** and **options nouveau modeset=0**.

Step 7 Reboot the system and make sure the grub configuration was applied. You can check **/proc/cmdline** for the grub configuration in use. For example:



```
root@graid-demo:/etc/modprobe.d# cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-5.15.0-46-generic root=UUID=32b02b62-7173-4f3b-a723-8aa1e2fbf60a ro text iommu=pt nvme_core.multipath=Y
```

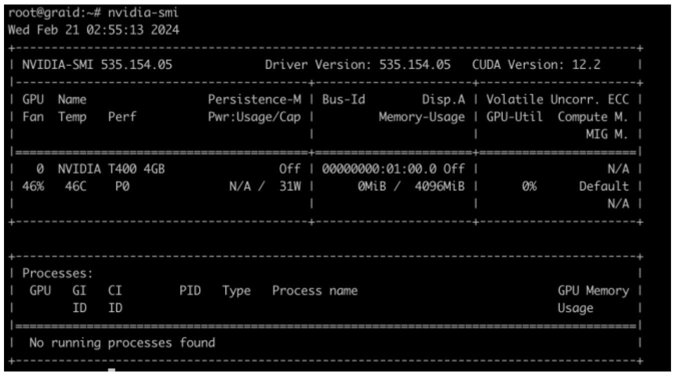
Step 8 Install the NVIDIA driver.

```
$ wget https://us.download.nvidia.com/XFree86/Linux-
x86_64/550.67/NVIDIA-Linux-x86_64-550.67.run

$ sudo chmod +x ./NVIDIA-Linux-x86_64-550.67.run

$ sudo ./NVIDIA-Linux-x86_64-550.67.run -s --no-systemd --no-opengl-
files --no-nvidia-modprobe --dkms --disable-nouveau
```

Step 9 The Nouveau driver is now disabled. Reboot and install the NVIDIA driver before proceeding with the installation.

```
$ sudo reboot
```

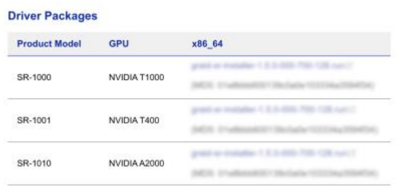
Step 10 Use the **nvidia-smi** command to confirm that the NVIDIA GPU is working. The following figure shows an output example of a successful installation.



```
root@graid:~# nvidia-smi
Wed Feb 21 02:55:13 2024
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.154.05             Driver Version: 535.154.05   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  NVIDIA T400 4GB                Off | 00000000:01:00.0 Off |                  N/A |
| 46%   46C    P0              N/A /  31W |      0MiB /  4096MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|  No running processes found                                                           |
+---------------------------------------------------------------------------------------+
```

Step 11 Go to the Graid Technology website to download the latest version of the pre-installer and make it executable, please download the package in Drivers & Documentation.

```
$ sudo chmod +x [Filename]
```



Driver Packages

Product Model	GPU	x86_64
SR-1000	NVIDIA T1000	[illegible]
SR-1001	NVIDIA T400	[illegible]
SR-1010	NVIDIA A2000	[illegible]

Step 12 Proceed to Executing the Installer and Completing the Installation to execute the installer and to complete the installation.

Manual Installation on an openSUSE Operating System

Graid Technology, Inc. recommends referring to Supported Operating Systems on our website and using the pre-installer to configure the environmental settings.

Step 1 Install openSUSE and select all online repositories.

Step 2 Install the package dependencies and build for DKMS.

```
$ sudo zypper addrepo -f
https://download.opensuse.org/distribution/leap/15.3/repo/oss/ leap-
15.3

$ sudo zypper --gpg-auto-import-keys refresh

$ sudo zypper install sudo vim wget libpci3 dkms ipmitool tar mdadm
libsgutils-devel libudev-devel sqlite3 automake dialog

$ sudo zypper install -C kernel-default-devel=$(uname -r | awk -F"-
default" '{print $1}')
```

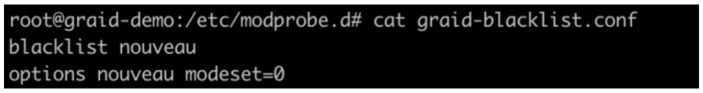
Step 3 Add the kernel option. This step prevents the Nouveau driver from loading during installation and disables IOMMU in the system BIOS.

```
$ sudo vim /etc/default/grub
```

Step 4 Append iommu=pt and 'nvme_core.multipath=Y' to GRUB_CMDLINE_LINUX_DEFAULT, and then update the grub configuration.

```
$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

Step 5 Append 'blacklist nouveau' to the end of the /etc/modprobe.d/graid-blacklist.conf file to disable the Nouveau driver. You might need to manually create the /etc/modprobe.d/graid-blacklist.conf file and append **blacklist nouveau** and **options nouveau modeset=0**.

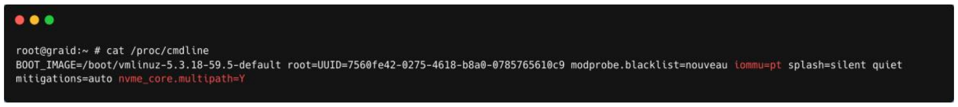


```
root@graid-demo:/etc/modprobe.d# cat graid-blacklist.conf
blacklist nouveau
options nouveau modeset=0
```

Step 6 Set the **allow_unsupported_modules** option to **1** in the /etc/modprobe.d/10-unsupported-modules.conf file and update initrd.

```
$ sudo mkinitrd
```

Step 7 Reboot the system and make sure the grub configuration was applied. You can check **/proc/cmdline** for the grub configuration in use. For example:



```
root@graid:~ # cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-5.3.18-59.5-default root=UUID=7560fe42-0275-4618-b8a0-0785765610c9 modprobe.blacklist=nouveau iommu=pt splash=silent quiet
mitigations=auto nvme_core.multipath=Y
```

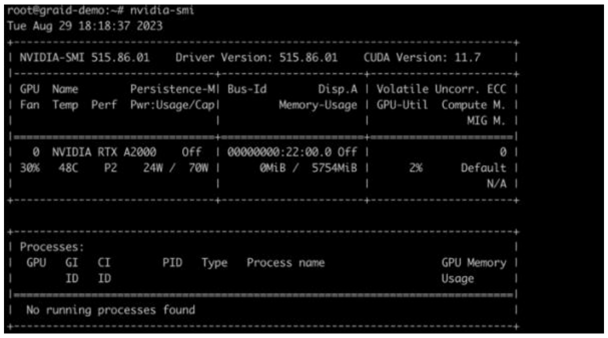
Step 8 Install the NVIDIA driver.

```
$ wget https://us.download.nvidia.com/XFree86/Linux-
x86_64/550.67/NVIDIA-Linux-x86_64-550.67.run

$ sudo chmod +x ./NVIDIA-Linux-x86_64-550.67.run

$ sudo ./NVIDIA-Linux-x86_64-550.67.run -s --no-systemd --no-opengl-
files --no-nvidia-modprobe --dkms --disable-nouveau
```

Step 9 The Nouveau driver is now disabled. Reboot and install the NVIDIA driver before proceeding with the installation.

```
$ sudo reboot
```

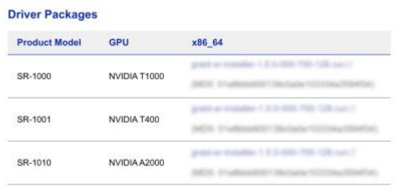
Step 10 Use the **nvidia-smi** command to confirm that the NVIDIA GPU is working. The following figure shows an output example of a successful installation.



```
root@graid-demo:~# nvidia-smi
Tue Aug 29 18:18:37 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 515.86.01    Driver Version: 515.86.01    CUDA Version: 11.7     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA RTX A2000    Off  | 00000000:22:00.0 Off |                    0 |
| 30%   48C    P2    24W /  70W |      0MiB /  5754MiB |      2%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Step 11 Go to the Graid Technology website to download the latest version of the pre-installer and make it executable, please download the package in Drivers & Documentation.

```
$ sudo chmod +x [Filename]
```



Driver Packages

Product Model	GPU	x86_64
SR-1000	NVIDIA T1000	[illegible]
SR-1001	NVIDIA T400	[illegible]
SR-1010	NVIDIA A2000	[illegible]

Step 12 Proceed to Executing the Installer and Completing the Installation to execute the installer and to complete the installation.

Manual Installation on a SLES Operating System

Graid Technology, Inc. recommends referring to Supported Operating Systems on our website and using the pre-installer to configure the environmental settings.

Step 1 Install SLES with the following extensions and modules:

- SUSE Package Hub 15 SP3 x86_64
- Desktop Applications Module 15 SP3 x86_64
- Development Tools Module 15 SP3 x86_64

Step 2 Install the package dependencies and build for DKMS.

```
$ sudo zypper addrepo -f
https://download.opensuse.org/distribution/leap/15.3/repo/oss/ leap-
15.3

$ sudo zypper --gpg-auto-import-keys refresh

$ sudo zypper install sudo vim wget libpci3 dkms ipmitool tar mdadm
libsgutils-devel libudev-devel sqlite3 automake dialog

$ sudo zypper install -C kernel-default-devel=$(uname -r | awk -F”-
default” ‘{print $1}’)
```

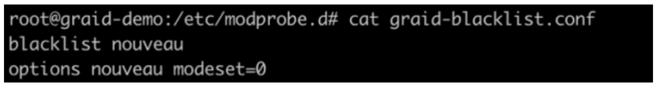
Step 3 Add the kernel option. This step prevents the Nouveau driver from loading during installation and disables IOMMU in the system BIOS.

```
$ sudo vim /etc/default/grub
```

Step 4 Append **iommu=pt** and **nvme_core.multipath=Y** to GRUB_CMDLINE_LINUX_DEFAULT, and then update the grub configuration:

```
$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

Step 5 Append **blacklist nouveau** to the end of the /etc/modprobe.d/graid-blacklist.conf file to disable the Nouveau driver. You might need to manually modify the configuration file.

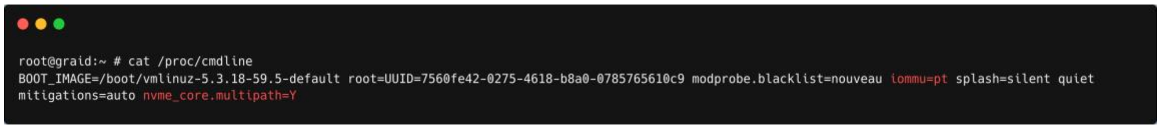


```
root@graid-demo:/etc/modprobe.d# cat graid-blacklist.conf
blacklist nouveau
options nouveau modeset=0
```

Step 6 Set the **allow_unsupported_modules** option to 1 in the /etc/modprobe.d/10-unsupported-modules.conf file and update initrd.

```
$ sudo mkinitrd
```

Step 7 Reboot the system and make sure the grub configuration was applied. You can check **/proc/cmdline** for the grub configuration in use. For example:



```
root@graid:~ # cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-5.3.18-59.5-default root=UUID=7560fe42-0275-4618-b8a0-0785765610c9 modprobe.blacklist=nouveau iommu=pt splash=silent quiet
mitigations=auto nvme_core.multipath=Y
```

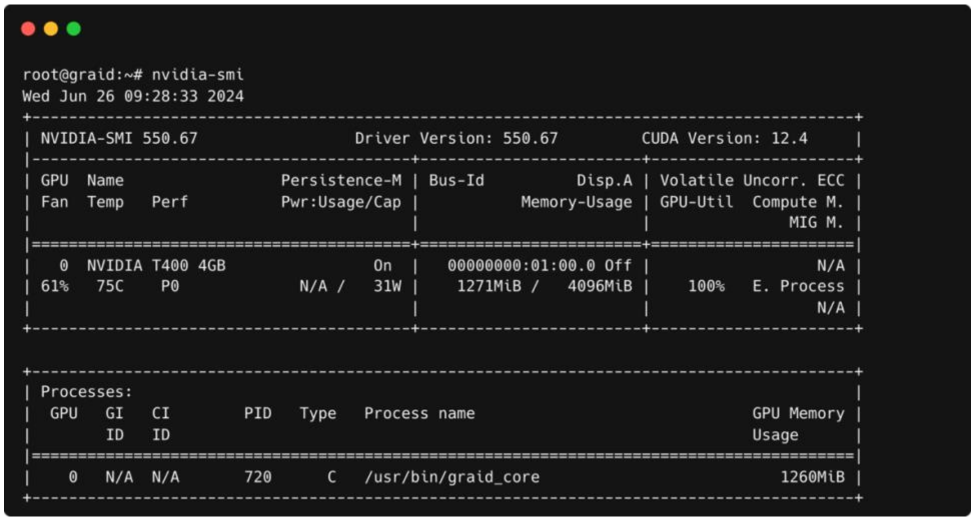
Step 8 Install the NVIDIA driver.

```
$ wget https://us.download.nvidia.com/XFree86/Linux-
x86_64/550.67/NVIDIA-Linux-x86_64-550.67.run

$ sudo chmod +x ./NVIDIA-Linux-x86_64-550.67.run

$ sudo ./NVIDIA-Linux-x86_64-550.67.run -s --no-systemd --no-opengl-
files --no-nvidia-modprobe --dkms --disable-nouveau

$ sudo reboot
```

Step 9 The Nouveau driver is now disabled. Reboot and install the NVIDIA driver before proceeding with the installation.

```
$ sudo reboot
```

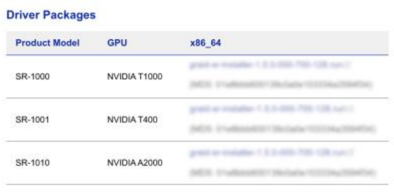
Step 10 Use the **nvidia-smi** command to confirm that the NVIDIA GPU is working. The following figure shows an output example of a successful installation.



```
root@graid:~# nvidia-smi
Wed Jun 26 09:28:33 2024
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.67                 Driver Version: 550.67         CUDA Version: 12.4     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA T400 4GB                On  |   00000000:01:00.0 Off |                  N/A |
| 61%   75C    P0             N/A /   31W |    1271MiB /   4096MiB |    100%   E. Process |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI        PID   Type   Process name                              GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|    0   N/A  N/A       720      C   /usr/bin/graid_core                          1260MiB |
+-----------------------------------------------------------------------------------------+
```

Step 11 Go to the Graid Technology website to download the latest version of the pre-installer and make it executable, please download the package in [Drivers & Documentation].

```
$ sudo chmod +x [Filename]
```



Driver Packages

Product Model	GPU	x86_64
SR-1000	NVIDIA T1000	[illegible]
SR-1001	NVIDIA T400	[illegible]
SR-1010	NVIDIA A2000	[illegible]

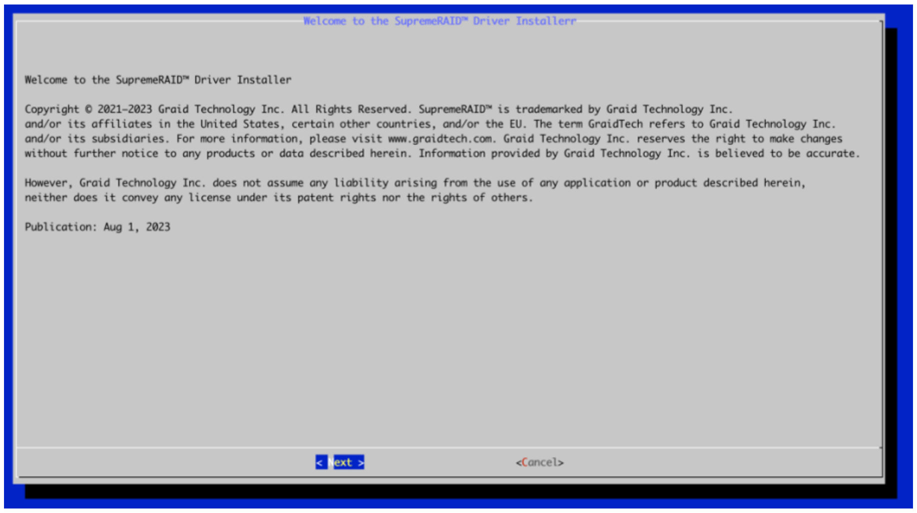
Step 12 Proceed to next part of [Executing the Installer and Completing the Installation] to execute the installer and to complete the installation.

Executing the Installer and Completing the Installation

Step 1 Execute the installer and follow the provided steps to complete the installation.

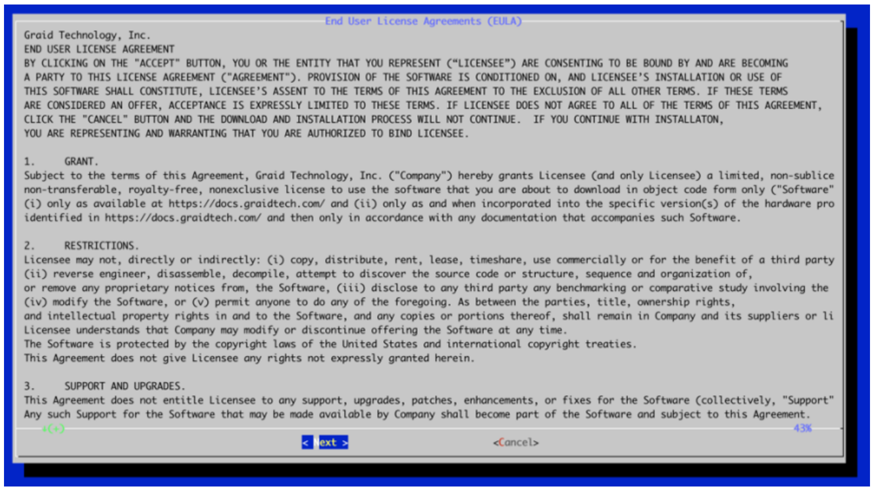
```
$ sudo ./[filename]
```

Step 2 At the Welcome page select **Next** and click **Enter** to view the end-user license agreement.



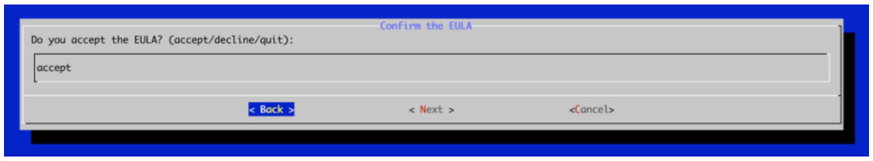
```
Welcome to the SupremeRAID™ Driver Installerr

Welcome to the SupremeRAID™ Driver Installer

Copyright © 2021-2023 Graid Technology Inc. All Rights Reserved. SupremeRAID™ is trademarked by Graid Technology Inc.
and/or its affiliates in the United States, certain other countries, and/or the EU. The term GraidTech refers to Graid Technology Inc.
and/or its subsidiaries. For more information, please visit www.graidtech.com. Graid Technology Inc. reserves the right to make changes
without further notice to any products or data described herein. Information provided by Graid Technology Inc. is believed to be accurate.

However, Graid Technology Inc. does not assume any liability arising from the use of any application or product described herein,
neither does it convey any license under its patent rights nor the rights of others.

Publication: Aug 1, 2023
```

Step 3 In the end-user license agreement, use the spacebar to scroll through the content. When you complete your review, select **Next** and click **Enter** to proceed.



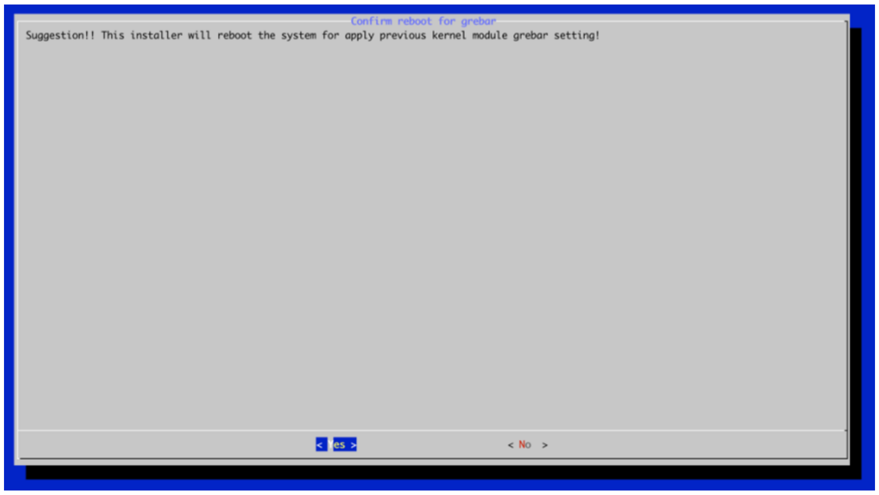
```
End User License Agreements (EULA)
Graid Technology, Inc.
END USER LICENSE AGREEMENT
BY CLICKING ON THE "ACCEPT" BUTTON, YOU OR THE ENTITY THAT YOU REPRESENT (“LICENSEE”) ARE CONSENTING TO BE BOUND BY AND ARE BECOMING
A PARTY TO THIS LICENSE AGREEMENT ("AGREEMENT"). PROVISION OF THE SOFTWARE IS CONDITIONED ON, AND LICENSEE’S INSTALLATION OR USE OF
THIS SOFTWARE SHALL CONSTITUTE, LICENSEE’S ASSENT TO THE TERMS OF THIS AGREEMENT TO THE EXCLUSION OF ALL OTHER TERMS. IF THESE TERMS
ARE CONSIDERED AN OFFER, ACCEPTANCE IS EXPRESSLY LIMITED TO THESE TERMS. IF LICENSEE DOES NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT,
CLICK THE "CANCEL" BUTTON AND THE DOWNLOAD AND INSTALLATION PROCESS WILL NOT CONTINUE.  IF YOU CONTINUE WITH INSTALLATON,
YOU ARE REPRESENTING AND WARRANTING THAT YOU ARE AUTHORIZED TO BIND LICENSEE.

1.     GRANT.
Subject to the terms of this Agreement, Graid Technology, Inc. ("Company") hereby grants Licensee (and only Licensee) a limited, non-sublice
non-transferable, royalty-free, nonexclusive license to use the software that you are about to download in object code form only ("Software"
(i) only as available at https://docs.graidtech.com/ and (ii) only as and when incorporated into the specific version(s) of the hardware pro
identified in https://docs.graidtech.com/ and then only in accordance with any documentation that accompanies such Software.

2.     RESTRICTIONS.
Licensee may not, directly or indirectly: (i) copy, distribute, rent, lease, timeshare, use commercially or for the benefit of a third party
(ii) reverse engineer, disassemble, decompile, attempt to discover the source code or structure, sequence and organization of,
or remove any proprietary notices from, the Software, (iii) disclose to any third party any benchmarking or comparative study involving the
(iv) modify the Software, or (v) permit anyone to do any of the foregoing. As between the parties, title, ownership rights,
and intellectual property rights in and to the Software, and any copies or portions thereof, shall remain in Company and its suppliers or li
Licensee understands that Company may modify or discontinue offering the Software at any time.
The Software is protected by the copyright laws of the United States and international copyright treaties.
This Agreement does not give Licensee any rights not expressly granted herein.

3.     SUPPORT AND UPGRADES.
This Agreement does not entitle Licensee to any support, upgrades, patches, enhancements, or fixes for the Software (collectively, "Support"
Any such Support for the Software that may be made available by Company shall become part of the Software and subject to this Agreement.
```

Step 4 Type **accept**, click tab, select **Next**, and click **Enter** to accept the license agreement.



```
Confirm the EULA
Do you accept the EULA? (accept/decline/quit):
accept
< Back >          < Next >          <Cancel>
```

Step 5 Complete the installation, and the installer will reboot the system.



```
Confirm reboot for grebar
Suggestion!! This installer will reboot the system for apply previous kernel module grebar setting!
< es >          < No  >
```

Step 6 To activate the software, apply the SupremeRAID™ license key.

```
$ sudo graidctl apply license [LICENSE_KEY]
```

USING THE SUPREMERAID™ DRIVER

This section describes how to use the basic functions of SupremeRAID™. It consists of step- by-step examples and command instructions that guide you to accessing all SupremeRAID™ features.

- To activate the SupremeRAID™ service, see Activating the SupremeRAID™ Driver and Managing the License(s).
- To set up a local volume (Virtual Drive), see Creating a RAID-5 Virtual Drive with Five NVMe SSDs.
- To create drive group without journaling space, see Creating a RAID-6 Drive Group without Journaling Space.
- To edit journal mode of a drive group, see Modifying Journal Mode on a RAID-5 Drive Group.
- To set up an Initiator server, see Creating a Physical Drive from the Remote NVMe-oF Targets.
- To replace the physical drive, see Replace the Nearly Worn-out or Broken SSD.
- To set up a Target server, see Exporting the Virtual Drive as an NVMe-oF Target Drive Using RDMA to the Initiator.
- To set up the high availability (HA) feature in one server, see Setting Up the Dual-Controller to Enable HA and Auto-Failover.

Activating the SupremeRAID™ Driver and Managing the License(s)

When you install the SupremeRAID™ driver, you must activate the SupremeRAID™ service by applying a specific license key prior to use the SupremeRAID™ service, and the license key you could get from your vendor. Once this is done, you can perform activities such as creating drive groups and virtual drives to use the SupremeRAID™.

- To check the SupremeRAID™ driver version, issue:

```
$ sudo graidctl version
```

- To activate the SupremeRAID™ software, issue:

```
$ sudo graidctl apply license [LICENSE_KEY]
```

- To check the license information, issue:

```
$ sudo graidctl describe license
```

- To check the controller status, issue:

```
$ sudo graidctl list controller
```

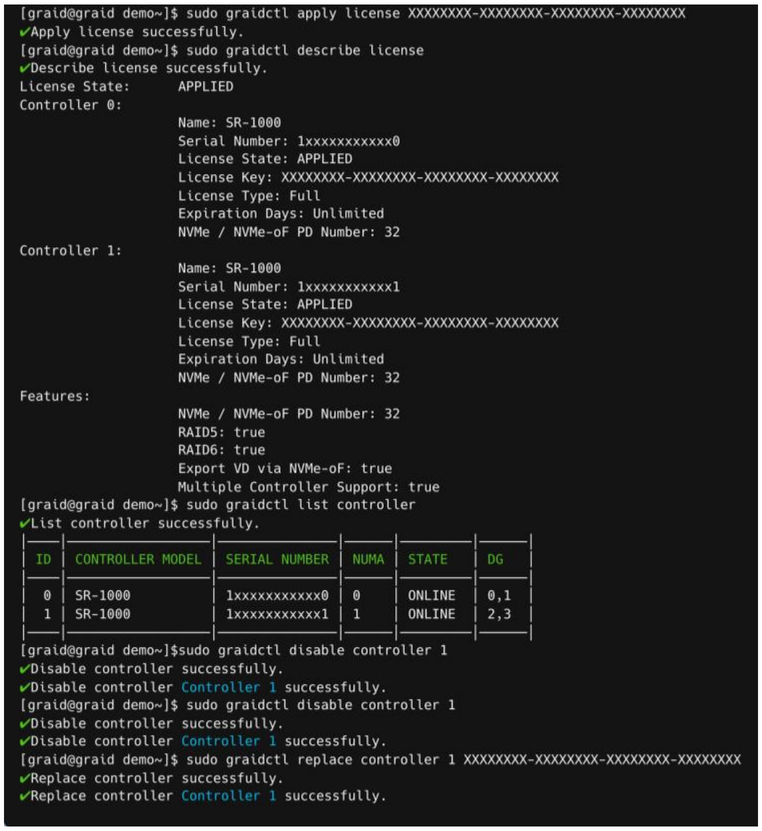
- To replace a new controller with the same model of the controller when the old controller is failure or missing, issue:

```
$ sudo graidctl disable controller [Controller_ID]
$ sudo graidctl replace controller [Controller_ID] [LICENSE_KEY]
```

- To delete the old controller that failed, missing, or disabled, issue:

```
$ sudo graidctl delete controller [Controller_ID]
```

Output example:



```
[graid@graid demo~]$ sudo graidctl apply license XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX
✔Apply license successfully.
[graid@graid demo~]$ sudo graidctl describe license
✔Describe license successfully.
License State:      APPLIED
Controller 0:
                    Name: SR-1000
                    Serial Number: 1xxxxxxxxxxx0
                    License State: APPLIED
                    License Key: XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX
                    License Type: Full
                    Expiration Days: Unlimited
                    NVMe / NVMe-oF PD Number: 32
Controller 1:
                    Name: SR-1000
                    Serial Number: 1xxxxxxxxxxx1
                    License State: APPLIED
                    License Key: XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX
                    License Type: Full
                    Expiration Days: Unlimited
                    NVMe / NVMe-oF PD Number: 32
Features:
                    NVMe / NVMe-oF PD Number: 32
                    RAID5: true
                    RAID6: true
                    Export VD via NVMe-oF: true
                    Multiple Controller Support: true
[graid@graid demo~]$ sudo graidctl list controller
✔List controller successfully.
|────|──────────────────|───────────────|──────|─────────|───────|
| ID | CONTROLLER MODEL | SERIAL NUMBER | NUMA | STATE   | DG    |
|────|──────────────────|───────────────|──────|─────────|───────|
|  0 | SR-1000          | 1xxxxxxxxxxx0 | 0    | ONLINE  | 0,1   |
|  1 | SR-1000          | 1xxxxxxxxxxx1 | 1    | ONLINE  | 2,3   |
|────|──────────────────|───────────────|──────|─────────|───────|
[graid@graid demo~]$sudo graidctl disable controller 1
✔Disable controller successfully.
✔Disable controller Controller 1 successfully.
[graid@graid demo~]$ sudo graidctl disable controller 1
✔Disable controller successfully.
✔Disable controller Controller 1 successfully.
[graid@graid demo~]$ sudo graidctl replace controller 1 XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX
✔Replace controller successfully.
✔Replace controller Controller 1 successfully.
```

Note: To apply the license, you might need to provide the NVIDIA GPU serial number to Graid Technology Technical Support. Use either of the following commands to obtain the serial number for all NVIDIA cards in your environment:

```
$ sudo nvidia-smi --query-gpu=name,index,serial --format=csv
```

OR

```
$ sudo nvidia-smi -q | grep -i serial
```

Note: If two controllers are activated in the graid.conf system configuration file, the SupremeRAID™ service prevents you from activating any additional controllers until one of the existing controllers is removed. This safeguard prevents conflicts and ensures proper system operation. Exercise caution and consult the software documentation or seek professional assistance if needed.

Creating a RAID-5 Virtual Drive with Five NVMe SSDs

To create a RAID-5 virtual drive with 5 NVMe SSDs:

Step 1 Create a physical drive.

```
$ sudo graidctl create physical_drive /dev/nvme0-4
```

Step 2 Create a drive group.

```
$ sudo graidctl create drive_group raid5 0-4
```

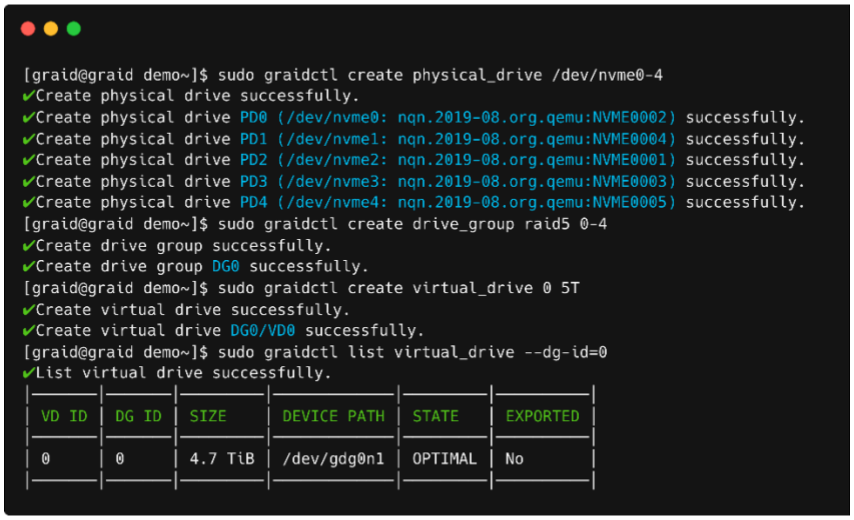
Step 3 Create a virtual drive with a 5TB volume size.

```
$ sudo graidctl create virtual_drive 0 5T
```

Step 4 Check the device path of the new virtual drive.

```
$ sudo graidctl list virtual_drive --dg-id=0
```

Output example:



```
[graid@graid demo~]$ sudo graidctl create physical_drive /dev/nvme0-4
✔Create physical drive successfully.
✔Create physical drive PD0 (/dev/nvme0: nqn.2019-08.org.qemu:NVME0002) successfully.
✔Create physical drive PD1 (/dev/nvme1: nqn.2019-08.org.qemu:NVME0004) successfully.
✔Create physical drive PD2 (/dev/nvme2: nqn.2019-08.org.qemu:NVME0001) successfully.
✔Create physical drive PD3 (/dev/nvme3: nqn.2019-08.org.qemu:NVME0003) successfully.
✔Create physical drive PD4 (/dev/nvme4: nqn.2019-08.org.qemu:NVME0005) successfully.
[graid@graid demo~]$ sudo graidctl create drive_group raid5 0-4
✔Create drive group successfully.
✔Create drive group DG0 successfully.
[graid@graid demo~]$ sudo graidctl create virtual_drive 0 5T
✔Create virtual drive successfully.
✔Create virtual drive DG0/VD0 successfully.
[graid@graid demo~]$ sudo graidctl list virtual_drive --dg-id=0
✔List virtual drive successfully.
│───────│───────│─────────│─────────────│─────────│──────────│
│ VD ID │ DG ID │ SIZE    │ DEVICE PATH │ STATE   │ EXPORTED │
│───────│───────│─────────│─────────────│─────────│──────────│
│ 0     │ 0     │ 4.7 TiB │ /dev/gdg0n1 │ OPTIMAL │ No       │
│───────│───────│─────────│─────────────│─────────│──────────│
```

Creating a RAID-6 Drive Group without Journal Space

To create a RAID-6 drive group without journal space.

Step 1 Create physical drives.

```
$ sudo graidctl create physical_drive /dev/nvme0-4
```

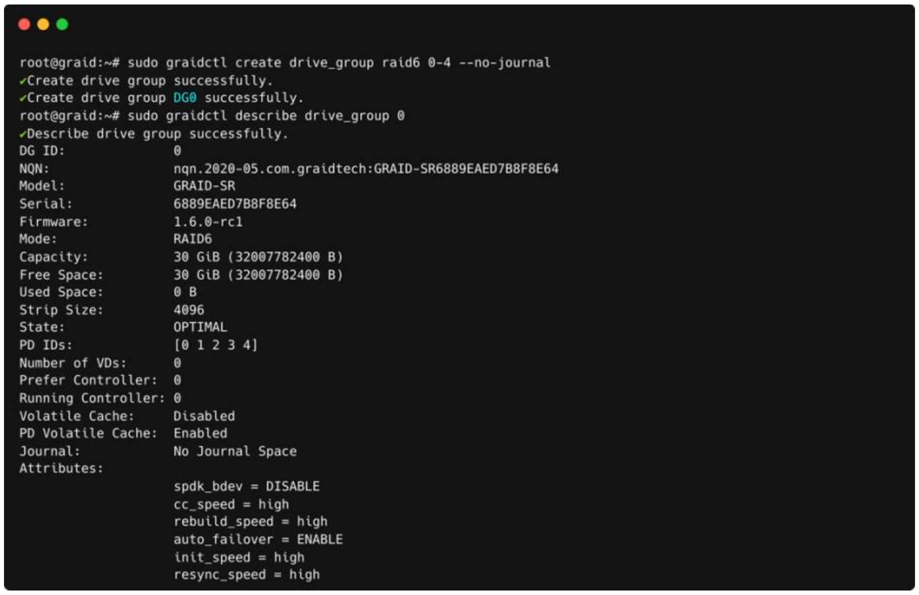
Step 2 Create a RAID-6 drive group without journal space.

```
$ sudo graidctl create drive_group raid6 0-4 --no-journal
```

Step 3 List the drive group configuration, the journal section should show 'No Journal Space'.

```
$ sudo graidctl describe drive_group [DG_ID]
```

Output example:



```
root@graid:~# sudo graidctl create drive_group raid6 0-4 --no-journal
✓Create drive group successfully.
✓Create drive group DG0 successfully.
root@graid:~# sudo graidctl describe drive_group 0
✓Describe drive group successfully.
DG ID:              0
NQN:                nqn.2020-05.com.graidtech:GRAID-SR6889EAED7B8F8E64
Model:              GRAID-SR
Serial:             6889EAED7B8F8E64
Firmware:           1.6.0-rc1
Mode:               RAID6
Capacity:           30 GiB (32007782400 B)
Free Space:         30 GiB (32007782400 B)
Used Space:         0 B
Strip Size:         4096
State:              OPTIMAL
PD IDs:             [0 1 2 3 4]
Number of VDs:      0
Prefer Controller:  0
Running Controller: 0
Volatile Cache:     Disabled
PD Volatile Cache:  Enabled
Journal:            No Journal Space
Attributes:
                    spdk_bdev = DISABLE
                    cc_speed = high
                    rebuild_speed = high
                    auto_failover = ENABLE
                    init_speed = high
                    resync_speed = high
```

Note: Once the drive group is set up, the journal space cannot be recreated. Without journal space, you cannot edit journal mode.

Modifying Journal Mode on a RAID-5 Drive Group

To edit the journal mode of a RAID-5 drive group.

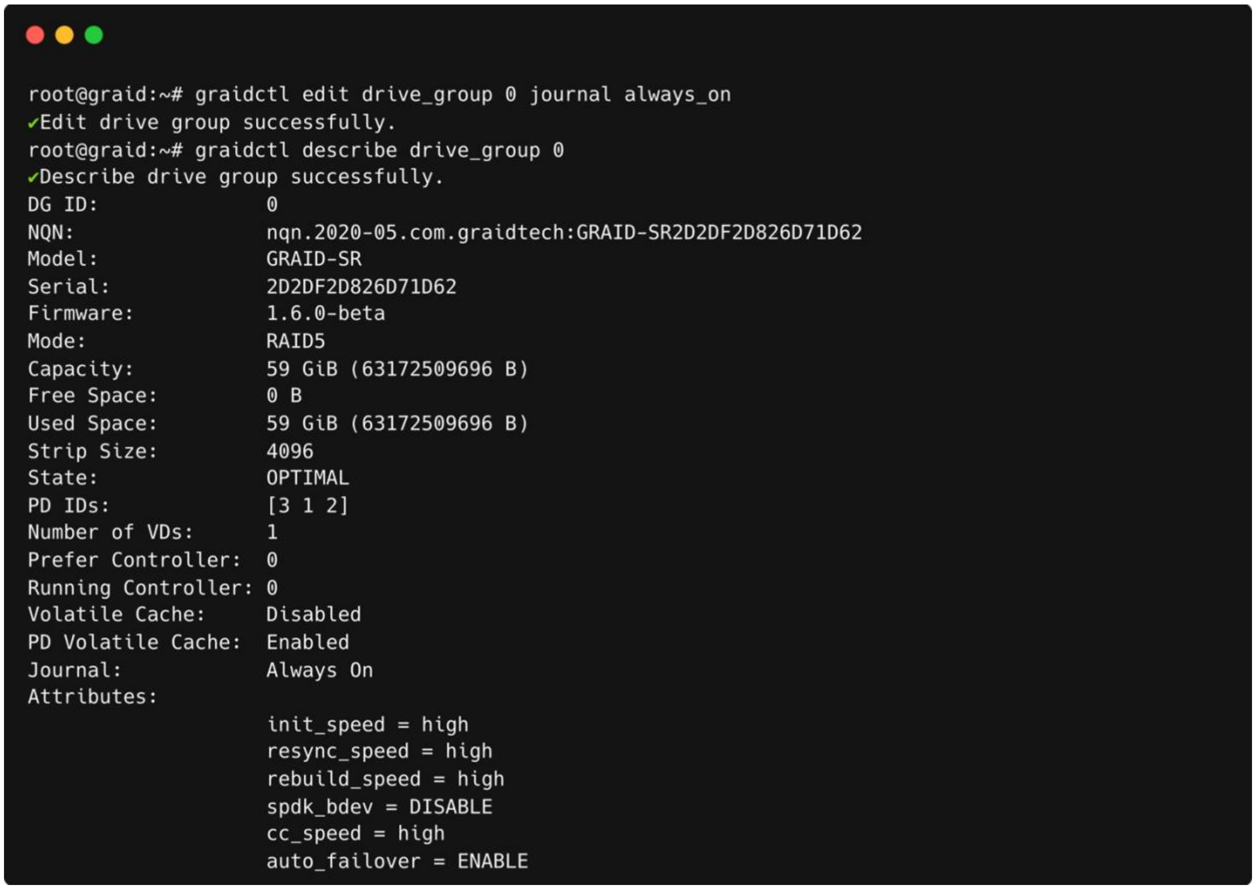
Step 1 List current drive group configuration.

```
$ sudo graidctl describe drive_group
```

Step 2 Modify the journal mode.

```
$ sudo graidctl edit drive_group [DG_ID] journal [JOURNAL_MODE]
```

Output example:



```
root@graid:~# graidctl edit drive_group 0 journal always_on
✓Edit drive group successfully.
root@graid:~# graidctl describe drive_group 0
✓Describe drive group successfully.
DG ID:                0
NQN:                  nqn.2020-05.com.graidtech:GRAID-SR2D2DF2D826D71D62
Model:                GRAID-SR
Serial:               2D2DF2D826D71D62
Firmware:             1.6.0-beta
Mode:                 RAID5
Capacity:             59 GiB (63172509696 B)
Free Space:           0 B
Used Space:           59 GiB (63172509696 B)
Strip Size:           4096
State:                OPTIMAL
PD IDs:               [3 1 2]
Number of VDs:        1
Prefer Controller:    0
Running Controller:   0
Volatile Cache:       Disabled
PD Volatile Cache:    Enabled
Journal:              Always On
Attributes:
                      init_speed = high
                      resync_speed = high
                      rebuild_speed = high
                      spdk_bdev = DISABLE
                      cc_speed = high
                      auto_failover = ENABLE
```

Note: Only RAID5/6 can enable the journal function. If the user bypasses the creation of the journal space, it cannot be recreated.

Creating a Physical Drive from the Remote NVMe-oF Targets

To create a physical drive from the Remote NVMe-oF targets:

Step 1 Connect to the remote NVMe-oF target.

```
$ sudo graidctl connect remote_target [tcp|rdma|fc] [addr] [address
family] [service id]
```

Step 2 Check the NVMe drives from the remote NVMe-oF target.

```
$ sudo graidctl list nvme_drive
```

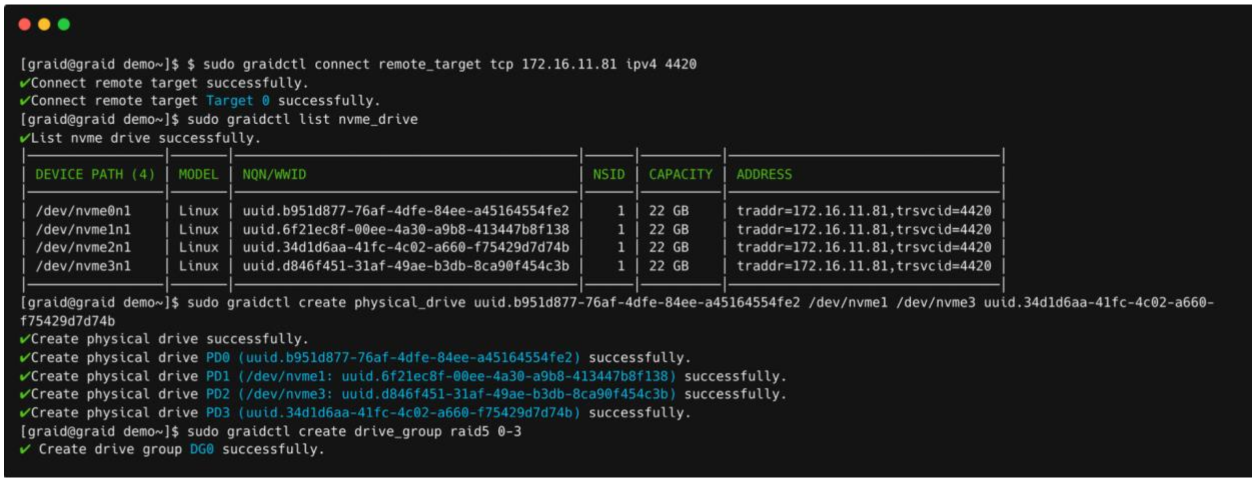
Step 3 Create the physical drives.

```
$ sudo graidctl create physical_drive [nqn or devpath]...
```

Step 4 Create a RAID5 drive group with four physical drives.

```
$ sudo graidctl create drive_group [Mode] [PD_ID]... [flags]
```

Output example:



```
[graid@graid demo~]$ $ sudo graidctl connect remote_target tcp 172.16.11.81 ipv4 4420
✔Connect remote target successfully.
✔Connect remote target Target 0 successfully.
[graid@graid demo~]$ sudo graidctl list nvme_drive
✔List nvme drive successfully.
|──────────────────|───────|─────────────────────────────────────────────|──────|──────────|────────────────────────────────|
| DEVICE PATH (4)  | MODEL | NQN/WWID                                    | NSID | CAPACITY | ADDRESS                        |
|──────────────────|───────|─────────────────────────────────────────────|──────|──────────|────────────────────────────────|
| /dev/nvme0n1     | Linux | uuid.b951d877-76af-4dfe-84ee-a45164554fe2   |    1 | 22 GB    | traddr=172.16.11.81,trsvcid=4420 |
| /dev/nvme1n1     | Linux | uuid.6f21ec8f-00ee-4a30-a9b8-413447b8f138   |    1 | 22 GB    | traddr=172.16.11.81,trsvcid=4420 |
| /dev/nvme2n1     | Linux | uuid.34d1d6aa-41fc-4c02-a660-f75429d7d74b   |    1 | 22 GB    | traddr=172.16.11.81,trsvcid=4420 |
| /dev/nvme3n1     | Linux | uuid.d846f451-31af-49ae-b3db-8ca90f454c3b   |    1 | 22 GB    | traddr=172.16.11.81,trsvcid=4420 |
|──────────────────|───────|─────────────────────────────────────────────|──────|──────────|────────────────────────────────|
[graid@graid demo~]$ sudo graidctl create physical_drive uuid.b951d877-76af-4dfe-84ee-a45164554fe2 /dev/nvme1 /dev/nvme3 uuid.34d1d6aa-41fc-4c02-a660-
f75429d7d74b
✔Create physical drive successfully.
✔Create physical drive PD0 (uuid.b951d877-76af-4dfe-84ee-a45164554fe2) successfully.
✔Create physical drive PD1 (/dev/nvme1: uuid.6f21ec8f-00ee-4a30-a9b8-413447b8f138) successfully.
✔Create physical drive PD2 (/dev/nvme3: uuid.d846f451-31af-49ae-b3db-8ca90f454c3b) successfully.
✔Create physical drive PD3 (uuid.34d1d6aa-41fc-4c02-a660-f75429d7d74b) successfully.
[graid@graid demo~]$ sudo graidctl create drive_group raid5 0-3
✔ Create drive group DG0 successfully.
```

Replace the Nearly Worn-out or Broken SSD.

To replace the SSD that is nearly worn-out or broken:

Step 1 Check the status of the physical drive. If the drive is already displaying as MISSING or another abnormal status, you can skip step 2 and go directly to step 3.

```
$ sudo graidctl list pd
```

Step 2 If the physical drive status is "online", mark the physical drive as BAD.

```
$ sudo graidctl edit pd [OLD_PD_ID] marker bad
```

Step 3 Replace the NVMe SSD. The state of the previous physical drive will indicate FAILED.

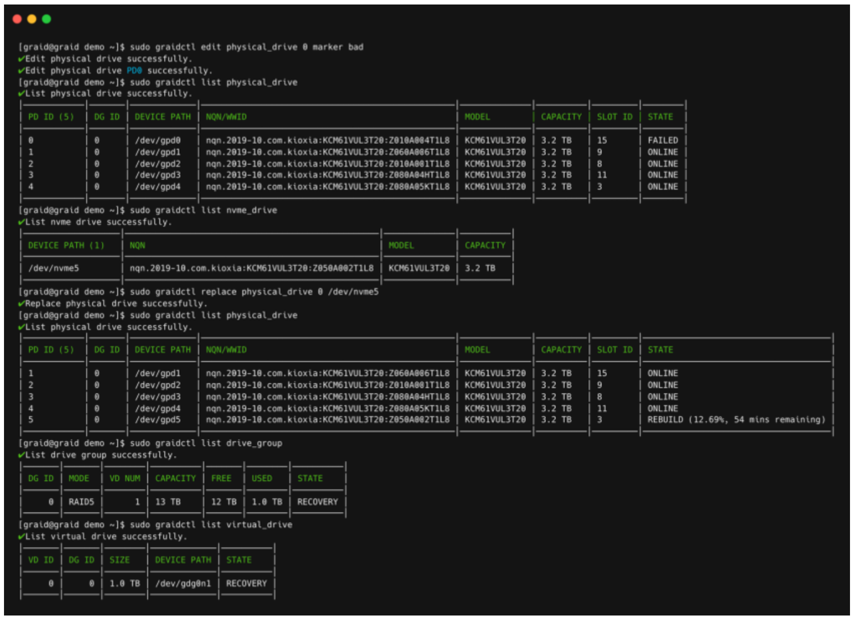
Step 4 Check the NQN of the new SSD.

```
$ sudo graidctl list nvme_drive
```

Step 5 Replace the physical drive.

```
$ sudo graidctl replace physical_drive [OLD_PD_ID]
[DEVICE_PATH|NQN|WWID]
```

Output example:



```
[graid@graid demo ~]$ sudo graidctl edit physical_drive 0 marker bad
✔Edit physical drive successfully.
✔Edit physical drive PD0 successfully.
[graid@graid demo ~]$ sudo graidctl list physical_drive
✔List physical drive successfully.
|-----------|-------|-------------|---------------------------------------------------|-------------|----------|---------|--------|
| PD ID (5) | DG ID | DEVICE PATH | NQN/WWID                                          | MODEL       | CAPACITY | SLOT ID | STATE  |
|-----------|-------|-------------|---------------------------------------------------|-------------|----------|---------|--------|
| 0         | 0     | /dev/gpd0   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z010A004T1L8 | KCM61VUL3T20 | 3.2 TB   | 15      | FAILED |
| 1         | 0     | /dev/gpd1   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z060A006T1L8 | KCM61VUL3T20 | 3.2 TB   | 9       | ONLINE |
| 2         | 0     | /dev/gpd2   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z010A001T1L8 | KCM61VUL3T20 | 3.2 TB   | 8       | ONLINE |
| 3         | 0     | /dev/gpd3   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A04HT1L8 | KCM61VUL3T20 | 3.2 TB   | 11      | ONLINE |
| 4         | 0     | /dev/gpd4   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A05KT1L8 | KCM61VUL3T20 | 3.2 TB   | 3       | ONLINE |
|-----------|-------|-------------|---------------------------------------------------|-------------|----------|---------|--------|
[graid@graid demo ~]$ sudo graidctl list nvme_drive
✔List nvme drive successfully.
|-----------------|---------------------------------------------------|--------------|----------|
| DEVICE PATH (1) | NQN                                               | MODEL        | CAPACITY |
|-----------------|---------------------------------------------------|--------------|----------|
| /dev/nvme5      | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z050A002T1L8 | KCM61VUL3T20 | 3.2 TB   |
|-----------------|---------------------------------------------------|--------------|----------|
[graid@graid demo ~]$ sudo graidctl replace physical_drive 0 /dev/nvme5
✔Replace physical drive successfully.
[graid@graid demo ~]$ sudo graidctl list physical_drive
✔List physical drive successfully.
|-----------|-------|-------------|---------------------------------------------------|-------------|----------|---------|--------------------------------------|
| PD ID (5) | DG ID | DEVICE PATH | NQN/WWID                                          | MODEL       | CAPACITY | SLOT ID | STATE                                |
|-----------|-------|-------------|---------------------------------------------------|-------------|----------|---------|--------------------------------------|
| 1         | 0     | /dev/gpd1   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z060A006T1L8 | KCM61VUL3T20 | 3.2 TB   | 15      | ONLINE                               |
| 2         | 0     | /dev/gpd2   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z010A001T1L8 | KCM61VUL3T20 | 3.2 TB   | 9       | ONLINE                               |
| 3         | 0     | /dev/gpd3   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A04HT1L8 | KCM61VUL3T20 | 3.2 TB   | 8       | ONLINE                               |
| 4         | 0     | /dev/gpd4   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A05KT1L8 | KCM61VUL3T20 | 3.2 TB   | 11      | ONLINE                               |
| 5         | 0     | /dev/gpd5   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z050A002T1L8 | KCM61VUL3T20 | 3.2 TB   | 3       | REBUILD (12.69%, 54 mins remaining) |
|-----------|-------|-------------|---------------------------------------------------|-------------|----------|---------|--------------------------------------|
[graid@graid demo ~]$ sudo graidctl list drive_group
✔List drive group successfully.
|-------|-------|--------|----------|-------|--------|----------|
| DG ID | MODE  | VD NUM | CAPACITY | FREE  | USED   | STATE    |
|-------|-------|--------|----------|-------|--------|----------|
|     0 | RAID5 |      1 | 13 TB    | 12 TB | 1.0 TB | RECOVERY |
|-------|-------|--------|----------|-------|--------|----------|
[graid@graid demo ~]$ sudo graidctl list virtual_drive
✔List virtual drive successfully.
|-------|-------|--------|-------------|----------|
| VD ID | DG ID | SIZE   | DEVICE PATH | STATE    |
|-------|-------|--------|-------------|----------|
|     0 |     0 | 1.0 TB | /dev/gdg0n1 | RECOVERY |
|-------|-------|--------|-------------|----------|
```

Note: Make sure that the system or other applications are not utilizing the physical drive before initiating the creation or replacement process.

Exporting the Virtual Drive as an NVMe-oF Target Drive Using RDMA to the Initiator

To export the virtual drive as an NVMe-oF target drive using RDMA to the initiator:

Step 1 Create the RDMA/TCP NVMe-of target port services.

```
$ sudo graidctl create nvmeof_target [tcp|rdma] [interface] [address
family] [srvcid] [flags]
```

Step 2 Export a virtual drive as an NVMe-of target.

```
$ sudo graidctl export virtual_drive [DG_ID] [VD_ID]... [flags]
```

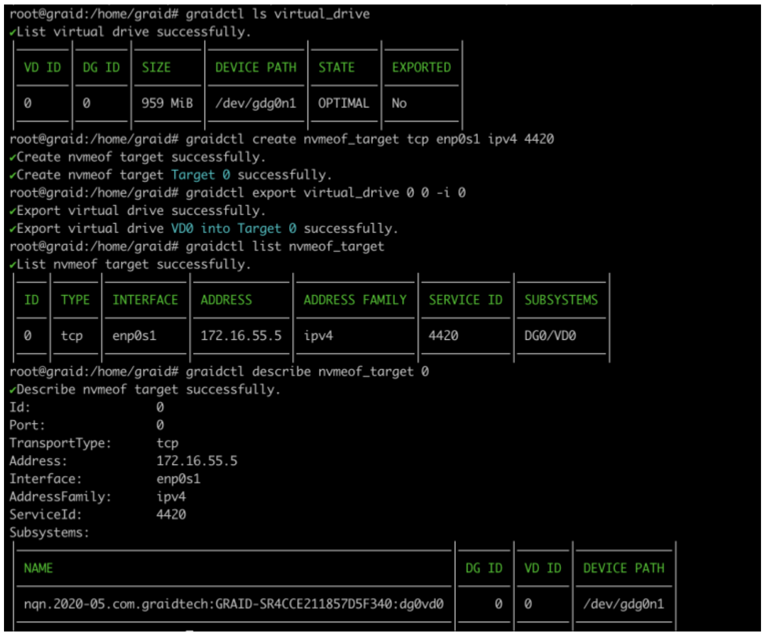
Step 3 List all NVMe-oF targets.

```
$ sudo graidctl list nvmeof_target [flags]
```

Step 4 Describe the detailed information for an NVMe-of target.

```
$ sudo graidctl describe nvmeof_target [PORT_ID] [flags]
```

Output example:



```
root@graid:/home/graid# graidctl ls virtual_drive
✓List virtual drive successfully.
│ VD ID │ DG ID │ SIZE    │ DEVICE PATH │ STATE   │ EXPORTED │
│ 0     │ 0     │ 959 MiB │ /dev/gdg0n1 │ OPTIMAL │ No       │
root@graid:/home/graid# graidctl create nvmeof_target tcp enp0s1 ipv4 4420
✓Create nvmeof target successfully.
✓Create nvmeof target Target 0 successfully.
root@graid:/home/graid# graidctl export virtual_drive 0 0 -i 0
✓Export virtual drive successfully.
✓Export virtual drive VD0 into Target 0 successfully.
root@graid:/home/graid# graidctl list nvmeof_target
✓List nvmeof target successfully.
│ ID │ TYPE │ INTERFACE │ ADDRESS     │ ADDRESS FAMILY │ SERVICE ID │ SUBSYSTEMS │
│ 0  │ tcp  │ enp0s1    │ 172.16.55.5 │ ipv4           │ 4420       │ DG0/VD0    │
root@graid:/home/graid# graidctl describe nvmeof_target 0
✓Describe nvmeof target successfully.
Id:                0
Port:              0
TransportType:     tcp
Address:           172.16.55.5
Interface:         enp0s1
AddressFamily:     ipv4
ServiceId:         4420
Subsystems:
│ NAME                                              │ DG ID │ VD ID │ DEVICE PATH │
│ nqn.2020-05.com.graidtech:GRAID-SR4CCE211857D5F340:dg0vd0 │     0 │ 0     │ /dev/gdg0n1 │
```

Setting Up the Dual-Controller to Enable HA and Auto-Failover

To activate the HA feature, you need two SupremeRAID™ cards installed in your server model and have the service activated. The total drive group count is four, with at least one drive group allocated to each controller. However, the number of drive groups assigned to each controller does not need to be equal.

If one controller fails and the auto-failover function is turned on (it is enabled by default), the drive group under the failed controller fails over immediately to the functioning controller. To ensure data integrity, the drive group statuses that failover switch to Resync mode.

Step 1 Activate two cards to enable the HA feature.

```
$ sudo graidctl apply license [LICENSE_KEY]
```

Step 2 Check the controller status.

```
$ sudo graidctl list controller
```

Step 3 Check the NVMe devices' NUMA location.

```
$ sudo graidctl list nvme_drive -n [NUMA_ID]
```

Step 4 Create physical drives.

```
$ sudo graidctl create physical_drive [DEVICE_PATH|NQN|WWID]
```

Step 5 Create two drive groups with specific controllers.

```
$ sudo graidctl create drive_group [RAID_MODE] [PD_IDs] -c
[Controller_ID]
```

Step 6 Create a specific virtual drive with a different drive group.

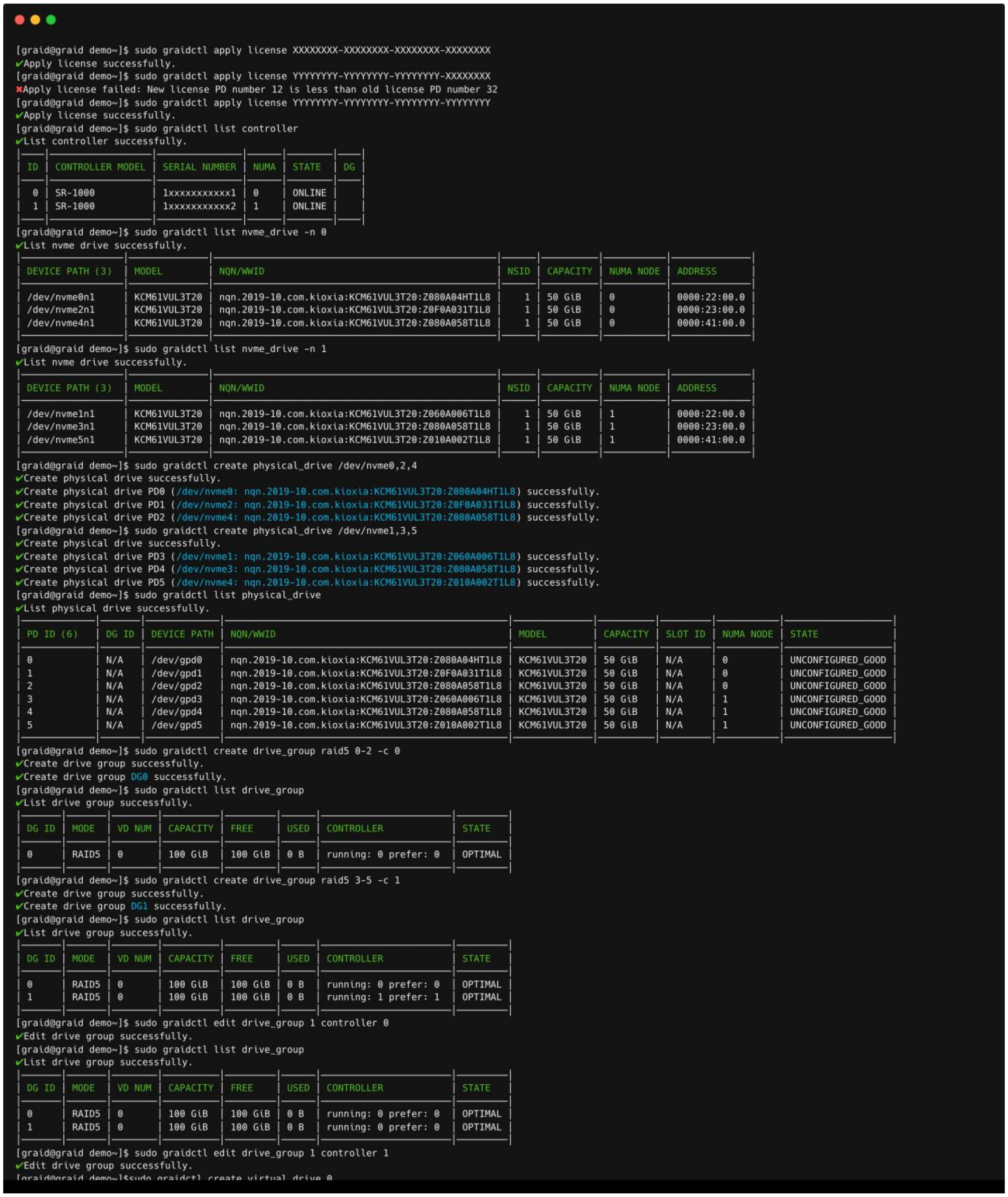
```
$ sudo graidctl create virtual_drive [DG_ID] [VD_SIZE]
```

Step 7 The drive group can optionally be assigned to a specific controller by editing it.

```
$ sudo graidctl edit [DG_ID] controller [Controller_ID]
```

Note: Typically, there is no need to set the controller manually while creating a drive group because SupremeRAID™ selects the optimal controller automatically based on the chosen physical drive. However, it is possible to adjust the controller manually for the drive group by making edits to it.

Output example:



```
[graid@graid demo~]$ sudo graidctl apply license XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX
✔Apply license successfully.
[graid@graid demo~]$ sudo graidctl apply license YYYYYYYY-YYYYYYYY-YYYYYYYY-XXXXXXXX
✖Apply license failed: New license PD number 12 is less than old license PD number 32
[graid@graid demo~]$ sudo graidctl apply license YYYYYYYY-YYYYYYYY-YYYYYYYY-YYYYYYYY
✔Apply license successfully.
[graid@graid demo~]$ sudo graidctl list controller
✔List controller successfully.
│────│──────────────────│──────────────│──────│─────────│────│
│ ID │ CONTROLLER MODEL │ SERIAL NUMBER│ NUMA │ STATE   │ DG │
│────│──────────────────│──────────────│──────│─────────│────│
│  0 │ SR-1000          │ 1xxxxxxxxxx1 │ 0    │ ONLINE  │    │
│  1 │ SR-1000          │ 1xxxxxxxxxx2 │ 1    │ ONLINE  │    │
│────│──────────────────│──────────────│──────│─────────│────│
[graid@graid demo~]$ sudo graidctl list nvme_drive -n 0
✔List nvme drive successfully.
│──────────────────│──────────────│──────────────────────────────────────────────────│──────│──────────│───────────│──────────────│
│ DEVICE PATH (3)  │ MODEL        │ NQN/WWID                                         │ NSID │ CAPACITY │ NUMA NODE │ ADDRESS      │
│──────────────────│──────────────│──────────────────────────────────────────────────│──────│──────────│───────────│──────────────│
│ /dev/nvme0n1     │ KCM61VUL3T20 │ nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A04HT1L8 │    1 │ 50 GiB   │ 0         │ 0000:22:00.0 │
│ /dev/nvme2n1     │ KCM61VUL3T20 │ nqn.2019-10.com.kioxia:KCM61VUL3T20:Z0F0A031T1L8 │    1 │ 50 GiB   │ 0         │ 0000:23:00.0 │
│ /dev/nvme4n1     │ KCM61VUL3T20 │ nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A058T1L8 │    1 │ 50 GiB   │ 0         │ 0000:41:00.0 │
│──────────────────│──────────────│──────────────────────────────────────────────────│──────│──────────│───────────│──────────────│
[graid@graid demo~]$ sudo graidctl list nvme_drive -n 1
✔List nvme drive successfully.
│──────────────────│──────────────│──────────────────────────────────────────────────│──────│──────────│───────────│──────────────│
│ DEVICE PATH (3)  │ MODEL        │ NQN/WWID                                         │ NSID │ CAPACITY │ NUMA NODE │ ADDRESS      │
│──────────────────│──────────────│──────────────────────────────────────────────────│──────│──────────│───────────│──────────────│
│ /dev/nvme1n1     │ KCM61VUL3T20 │ nqn.2019-10.com.kioxia:KCM61VUL3T20:Z060A006T1L8 │    1 │ 50 GiB   │ 1         │ 0000:22:00.0 │
│ /dev/nvme3n1     │ KCM61VUL3T20 │ nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A058T1L8 │    1 │ 50 GiB   │ 1         │ 0000:23:00.0 │
│ /dev/nvme5n1     │ KCM61VUL3T20 │ nqn.2019-10.com.kioxia:KCM61VUL3T20:Z010A002T1L8 │    1 │ 50 GiB   │ 1         │ 0000:41:00.0 │
│──────────────────│──────────────│──────────────────────────────────────────────────│──────│──────────│───────────│──────────────│
[graid@graid demo~]$ sudo graidctl create physical_drive /dev/nvme0,2,4
✔Create physical drive successfully.
✔Create physical drive PD0 (/dev/nvme0: nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A04HT1L8) successfully.
✔Create physical drive PD1 (/dev/nvme2: nqn.2019-10.com.kioxia:KCM61VUL3T20:Z0F0A031T1L8) successfully.
✔Create physical drive PD2 (/dev/nvme4: nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A058T1L8) successfully.
[graid@graid demo~]$ sudo graidctl create physical_drive /dev/nvme1,3,5
✔Create physical drive successfully.
✔Create physical drive PD3 (/dev/nvme1: nqn.2019-10.com.kioxia:KCM61VUL3T20:Z060A006T1L8) successfully.
✔Create physical drive PD4 (/dev/nvme3: nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A058T1L8) successfully.
✔Create physical drive PD5 (/dev/nvme4: nqn.2019-10.com.kioxia:KCM61VUL3T20:Z010A002T1L8) successfully.
[graid@graid demo~]$ sudo graidctl list physical_drive
✔List physical drive successfully.
│────────────│────────│─────────────│──────────────────────────────────────────────────│──────────────│──────────│─────────│───────────│───────────────────│
│ PD ID (6)  │ DG ID  │ DEVICE PATH │ NQN/WWID                                         │ MODEL        │ CAPACITY │ SLOT ID │ NUMA NODE │ STATE             │
│────────────│────────│─────────────│──────────────────────────────────────────────────│──────────────│──────────│─────────│───────────│───────────────────│
│ 0          │ N/A    │ /dev/gpd0   │ nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A04HT1L8 │ KCM61VUL3T20 │ 50 GiB   │ N/A     │ 0         │ UNCONFIGURED_GOOD │
│ 1          │ N/A    │ /dev/gpd1   │ nqn.2019-10.com.kioxia:KCM61VUL3T20:Z0F0A031T1L8 │ KCM61VUL3T20 │ 50 GiB   │ N/A     │ 0         │ UNCONFIGURED_GOOD │
│ 2          │ N/A    │ /dev/gpd2   │ nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A058T1L8 │ KCM61VUL3T20 │ 50 GiB   │ N/A     │ 0         │ UNCONFIGURED_GOOD │
│ 3          │ N/A    │ /dev/gpd3   │ nqn.2019-10.com.kioxia:KCM61VUL3T20:Z060A006T1L8 │ KCM61VUL3T20 │ 50 GiB   │ N/A     │ 1         │ UNCONFIGURED_GOOD │
│ 4          │ N/A    │ /dev/gpd4   │ nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A058T1L8 │ KCM61VUL3T20 │ 50 GiB   │ N/A     │ 1         │ UNCONFIGURED_GOOD │
│ 5          │ N/A    │ /dev/gpd5   │ nqn.2019-10.com.kioxia:KCM61VUL3T20:Z010A002T1L8 │ KCM61VUL3T20 │ 50 GiB   │ N/A     │ 1         │ UNCONFIGURED_GOOD │
│────────────│────────│─────────────│──────────────────────────────────────────────────│──────────────│──────────│─────────│───────────│───────────────────│
[graid@graid demo~]$ sudo graidctl create drive_group raid5 0-2 -c 0
✔Create drive group successfully.
✔Create drive group DG0 successfully.
[graid@graid demo~]$ sudo graidctl list drive_group
✔List drive group successfully.
│───────│───────│────────│──────────│─────────│──────│───────────────────────│─────────│
│ DG ID │ MODE  │ VD NUM │ CAPACITY │ FREE    │ USED │ CONTROLLER            │ STATE   │
│───────│───────│────────│──────────│─────────│──────│───────────────────────│─────────│
│ 0     │ RAID5 │ 0      │ 100 GiB  │ 100 GiB │ 0 B  │ running: 0 prefer: 0  │ OPTIMAL │
│───────│───────│────────│──────────│─────────│──────│───────────────────────│─────────│
[graid@graid demo~]$ sudo graidctl create drive_group raid5 3-5 -c 1
✔Create drive group successfully.
✔Create drive group DG1 successfully.
[graid@graid demo~]$ sudo graidctl list drive_group
✔List drive group successfully.
│───────│───────│────────│──────────│─────────│──────│───────────────────────│─────────│
│ DG ID │ MODE  │ VD NUM │ CAPACITY │ FREE    │ USED │ CONTROLLER            │ STATE   │
│───────│───────│────────│──────────│─────────│──────│───────────────────────│─────────│
│ 0     │ RAID5 │ 0      │ 100 GiB  │ 100 GiB │ 0 B  │ running: 0 prefer: 0  │ OPTIMAL │
│ 1     │ RAID5 │ 0      │ 100 GiB  │ 100 GiB │ 0 B  │ running: 1 prefer: 1  │ OPTIMAL │
│───────│───────│────────│──────────│─────────│──────│───────────────────────│─────────│
[graid@graid demo~]$ sudo graidctl edit drive_group 1 controller 0
✔Edit drive group successfully.
[graid@graid demo~]$ sudo graidctl list drive_group
✔List drive group successfully.
│───────│───────│────────│──────────│─────────│──────│───────────────────────│─────────│
│ DG ID │ MODE  │ VD NUM │ CAPACITY │ FREE    │ USED │ CONTROLLER            │ STATE   │
│───────│───────│────────│──────────│─────────│──────│───────────────────────│─────────│
│ 0     │ RAID5 │ 0      │ 100 GiB  │ 100 GiB │ 0 B  │ running: 0 prefer: 0  │ OPTIMAL │
│ 1     │ RAID5 │ 0      │ 100 GiB  │ 100 GiB │ 0 B  │ running: 0 prefer: 0  │ OPTIMAL │
│───────│───────│────────│──────────│─────────│──────│───────────────────────│─────────│
[graid@graid demo~]$ sudo graidctl edit drive_group 1 controller 1
✔Edit drive group successfully.
[graid@graid demo~]$sudo graidctl create virtual_drive 0
```

Upgrading the Software

To upgrade the Linux Driver, we offer two methods: silent upgrade and manual setup. Please follow the steps below for your preferred method. Perform the following procedure exactly as described. If you encounter any abnormal failure messages during the driver upgrade, please collect the logs and contact Graid Technical Support team.

Silent Upgrade

In the SupremeRAID™ Linux Driver, if you have already installed the SupremeRAID™ driver, there's no need to uninstall it. Simply run the pre-installer and installer then include '--accept-license' in the upgrade command to automatically apply the license key to the new software.

Step 1 Stop all applications running on the virtual drive.

Step 2 Stop the management service. If you have already enabled the graphical management console, please ensure to disable it as well.

```
$ sudo systemctl stop graid
$ sudo systemctl stop graid-mgr.service
```

Step 3 Download the upgrade driver package and make it executable.

Step 4 Run the pre-installer directly, it will automatically check the required dependencies.

```
$ sudo ./[filename] --yes
```

Step 5 Run the installer and add 'accept-license' to automatically apply the license key.

```
$ sudo ./[filename] --accept-license
```

Step 6 Check the driver version to ensure the upgrade is successful.

```
$ sudo graidctl version
```

Step 7 Use nvidia-smi to check the serial number of the SupremeRAID™ Card.

```
$ nvidia-smi -q
```

Step 8 Find the matching license key for the serial number, and then apply the license.

```
$ sudo graidctl apply license [LICENSE_KEY]
```

Manual Upgrade

If you need to perform a manual upgrade, please follow the steps below to upgrade the software.

Step 1 Stop all applications running on the virtual drive.

Step 2 Stop the management service. If you have already enabled the graphical management console, please ensure to disable it as well.

```
$ sudo systemctl stop graid
$ sudo systemctl stop graid-mgr.service
```

Step 3 Make sure the SupremeRAID™ kernel module is unloaded.

```
$ sudo rmmod graid_nvidia graid
```

Step 4 Check the NVIDIA driver DKMS status.

```
$ sudo dkms status nvidia
```

Step 5 The version of the NVIDIA driver installed in the kernel must match the SupremeRAID™ driver version. If they do not match, perform the following steps to uninstall the NVIDIA driver.

A Dracut the initramfs (Centos, Rocky Linux, AlmaLinux, and RHEL).

```
$ sudo dracut --omit-drivers "nvidia graid" -f
```

B Uninstall the NVIDIA driver.

```
$ sudo ./usr/bin/nvidia-uninstall
```

C Install the new NVIDIA driver.

```
$ sudo ./NVIDIA-Linux-x86_64-550.67.run -s --no-systemd --no-opengl-
files --no-nvidia-modprobe --dkms --disable-nouveau
```

D Reboot the server.

Step 6 Uninstall the package using the command appropriate for your operating system.

- For Centos, Rocky Linux, AlmaLinux, RHEL, openSUSE, and SLES:

```
$ sudo rpm -e graid-sr
```

- For Ubuntu:

```
$ sudo dpkg -r graid-sr
```

Step 7 Confirm that the SupremeRAID™ module is unloaded. There should not be any output.

```
$ sudo lsmod | grep graid
```

Step 8 Confirm that the SupremeRAID™ package is uninstalled using the command appropriate for your operating system, the output should be empty.

- For Centos, Rocky Linux, AlmaLinux, RHEL, openSUSE, and SLES:

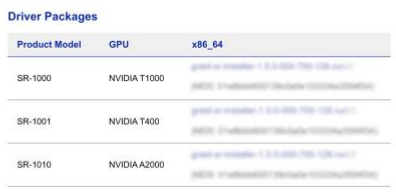
```
$ sudo rpm -qa | grep graid
```

- For Ubuntu:

```
$ sudo dpkg -l | grep graid
```

Step 9 Go to the Graid Technology website to download the latest version of the pre-installer and make it executable, please download the package in [Drivers & Documentation](#).

```
$ sudo chmod +x [Filename]
```



Driver Packages

Product Model	GPU	x86_64
SR-1000	NVIDIA T1000	[illegible]
SR-1001	NVIDIA T400	[illegible]
SR-1010	NVIDIA A2000	[illegible]

Step 10 Proceed to [Executing the Installer and Completing the Installation](#) to execute the installer and to complete the installation.

Step 11 Start the SupremeRAID™ service.

```
$ sudo systemctl enable graid
$ sudo systemctl start graid
```

OR

```
$ sudo systemctl --now enable graid
```

Note: If you upgrade from version 1.2.x to version 1.6.x of the graid driver, the device path changes from /dev/gvdXn1 to /dev/gdgXnY.

Replacing a SupremeRAID™ CardStop all applications running on the virtual drive.

Step 1 Stop the management service. If you have already enabled the graphical management console, please ensure to disable it as well.

```
$ sudo systemctl stop graid
$ sudo systemctl stop graid-mgr.service
```

Step 2 Back up the configuration file.

```
$ sudo cp /etc/graid.conf graid.conf.bak
```

Step 3 Make sure the SupremeRAID™ kernel module is unloaded.

```
$ sudo rmmod graid_nvidia graid
```

Step 4 Check the NVIDIA driver DKMS status.

```
$ sudo dkms status nvidia
```

Note: The NVIDIA driver version installed in the kernel must match the graid driver version. Perform step 5 if the versions do not match.

Step 5 Uninstall the package using the command appropriate for your operating system:

- For Centos, Rocky Linux, AlmaLinux, RHEL, openSUSE, and SLES:

```
$ sudo rpm -e graid-sr
```

- For Ubuntu:

```
$ sudo dpkg -r graid-sr
```

Step 6 Confirm that the SupremeRAID™ module is unloaded, the output should be empty.

```
$ sudo lsmod | grep graid
```

Step 7 Confirm that the SupremeRAID™ package is uninstalled using the command appropriate for your operating system, the output should be empty.

- Centos, Rocky Linux, AlmaLinux, RHEL, openSUSE, and SLES:

```
$ sudo rpm -qa | grep graid
```

- Ubuntu:

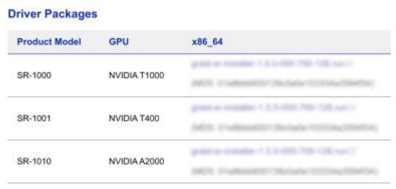
```
$ sudo dpkg -l | grep graid
```

Step 8 Power-off the server, and then install the new card into the server.

Step 9 Power-on the server.

Step 10 Go to the Graid Technology website to download the latest version of the pre-installer and make it executable, please download the package in Drivers & Documentation.

```
$ sudo chmod +x [Filename]
```



Driver Packages

Product Model	GPU	x86_64
SR-1000	NVIDIA T1000	[illegible]
SR-1001	NVIDIA T400	[illegible]
SR-1010	NVIDIA A2000	[illegible]

Step 11 Proceed to Executing the Installer and Completing the Installation to execute the installer and to complete the installation.

Step 12 When the installer finishes, restart the graidservice.

```
$ sudo systemctl restart graid
```

If the settings do not return properly after restarting graidservice, see Manually Migrating the RAID Configuration Between Hosts.

Note: If you are replacing a card in the system, deleting any inactive or invalid licenses associated with the old card is essential. Failing to do so may prevent other cards from becoming active, which is key for multi-controller systems.

COMMANDS AND SHORTCUTS

Syntax

Use the following syntax to run graidctl commands from the terminal window:

```
$ sudo graidctl [command] [OBJECT_TYPE] [OBJECT_ID] [flags]
```

where command, OBJECT_TYPE, OBJECT_ID, and flags are:

- **command**: Specifies the operation to perform on one or more resources (for example create, list, describe, and delete.
- OBJECT_TYPE: Specifies the object type. Object types are case-sensitive (for example license, physical_drive, and drive_group).
- OBJECT_ID: Specifies the object ID. Some commands support simultaneous operations on multiple objects. You can specify the OBJECT_ID individually or use a dash to describe an OBJECT_ID range. For example, to delete physical drives 1, 3, 4, and 5 simultaneously, issue the command:

  ```
  $ sudo graidctl delete physical_drive 1 3-5
  ```
- flags: Specifies optional flags. For example:
  - -force forces the deletion of a physical drive.

    ```
    $ sudo graidctl delete physical_drive 0 –force
    ```
  - -json prints output in JSON format. This flag can also assist with API implementation.

    ```
    $ sudo graidctl list virtual_drive --format json
    ```

For help, run graidctl help from the terminal window.

Command and Subcommand Quick Reference

General

Category	Commands	Alias	Sub-Commands	alias
Common	version			
License	apply		license	lic
	describe	desc	license	lic

Resources

Category	Commands	Alias	Sub-Commands	alias
NVMe Drive	list	l, ls	nvme_drive	nd
SCSi Drive	list	l, ls	scsi_drive	sd
Physical Drive	create	c, cre, crt	physical_drive	pd
	icreate	ic, icre, icrt	physical_drive	pd
	delete	d, del	physical_drive	pd
	describe	desc	physical_drive	pd
	edit	e	physical_drive	pd
	list	l, ls	physical_drive	pd
	replace	en	physical_drive	pd
Drive Group	create	c, cre, crt	drive_group	dg
	icreate	ic, icre, icrt	drive_group	dg
	delete	d, del	drive_group	dg
	describe	desc	drive_group	dg

Category	Commands	Alias	Sub-Commands	alias
	edit	e	drive_group	dg
	list	l, ls	drive_group	dg
Virtual Drive	create	c, cre, crt	virtual_drive	vd
	icreate	ic, icre, icrt	virtual_drive	vd
	delete	d, del	virtual_drive	vd
	describe	desc	virtual_drive	vd
	edit	e	virtual_drive	vd
	list	l, ls	virtual_drive	vd
Controller	enable		controller	cx
	disable		controller	cx
	delete	d, del	controller	cx
	list	l, ls	controller	cx
	replace	en	controller	cx
MD Boot Drive	import	im, imp	md_drive	md
	replace	en	md_drive	md
Config	describe	desc	config	conf
	edit	e	config	conf
	delete	d, del	config	conf
	restore	Re	Config	conf
Event	delete	d, del	event	ev
	list	l, ls	event	ev

Features

Category	Commands	Alias	Sub-Commands	alias
Consistency Check	describe	desc	consistency_check	cc
	set		consistency_check	cc
	start		consistency_check	cc
	stop		consistency_check	cc
Export NVMe-oF	create	c, cre, crt	nvmeof_target	nt
	describe	desc	nvmeof_target	nt
	delete	d, del	nvmeof_target	nt
	list	l, ls	nvmeof_target	nt
	export	ex, exp	virtual_drive	vd
	unexport	unex, unexp	virtual_drive	vd
Import NVMe-oF	connect	conn	remote_target	rt
	disconnect	dis, disconn	remote_target	rt
	list	l, ls	remote_target	rt

Managing Licenses

You can apply the license and check license information.

Applying the License

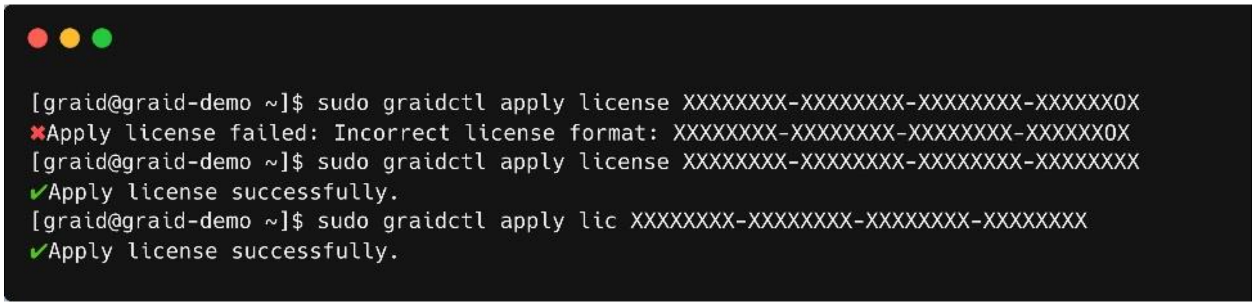
To apply the license and complete the installation, issue the following command:

```
$ sudo graidctl apply license [LICENSE_KEY] [flags]
```

OR

```
$ sudo graidctl apply lic [LICENSE_KEY] [flags]
```

Output example: for invalid and valid licenses is shown below:



```
[graid@graid-demo ~]$ sudo graidctl apply license XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXX0X
✖Apply license failed: Incorrect license format: XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXX0X
[graid@graid-demo ~]$ sudo graidctl apply license XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX
✔Apply license successfully.
[graid@graid-demo ~]$ sudo graidctl apply lic XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX
✔Apply license successfully.
```

Note: When applying the license, you must provide the serial number of the NVIDIA GPU to Graid Technology Technical Support.

To obtain NVIDIA GPU serial number, issue the following command:

```
$ sudo nvidia-smi --query-gpu=name,index,serial --format=csv
```

OR

```
$ sudo nvidia-smi -q | grep -i serial
```

This command lists all NVIDIA cards in your environment and their serial number.

Checking License Information

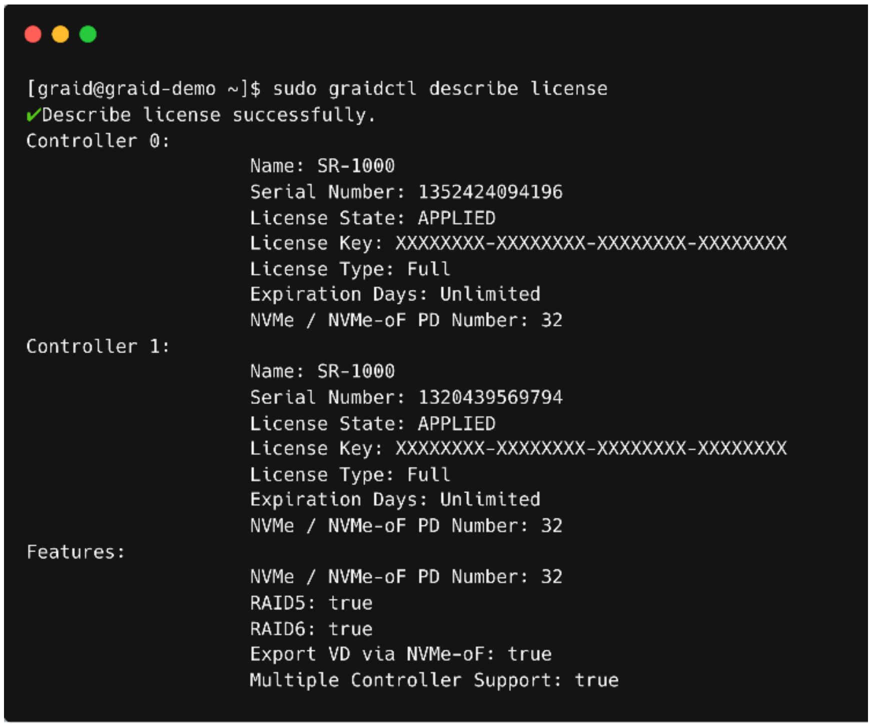
To obtain the license information, issue the following command:

```
$ sudo graidctl describe license [flags]
```

OR

```
$ sudo graidctl desc lic [flags]
```

Output example:



```
[graid@graid-demo ~]$ sudo graidctl describe license
✔Describe license successfully.
Controller 0:
                    Name: SR-1000
                    Serial Number: 1352424094196
                    License State: APPLIED
                    License Key: XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX
                    License Type: Full
                    Expiration Days: Unlimited
                    NVMe / NVMe-oF PD Number: 32
Controller 1:
                    Name: SR-1000
                    Serial Number: 1320439569794
                    License State: APPLIED
                    License Key: XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX
                    License Type: Full
                    Expiration Days: Unlimited
                    NVMe / NVMe-oF PD Number: 32
Features:
                    NVMe / NVMe-oF PD Number: 32
                    RAID5: true
                    RAID6: true
                    Export VD via NVMe-oF: true
                    Multiple Controller Support: true
```

Output content:

Field	Description
Name	Product SKU
Serial Number	Applied controller's serial number
License State	License state (see the following table)
License Key	Applied license key
License Type	License type (Full or Essential)
Expiration Days	Expiration date of the license key
NVMe / NVMe-oF PD Number	This license allows for a maximum number of PDs for NVMe/NVMe-oF.

License state:

State	Description
UNAPPLIED	License was not applied.
APPLIED	A valid license was applied.
INVALID	A valid license was applied, but a valid RAID card cannot be detected.

Feature support:

Features	Description	Value
NVMe / NVMe-oF PD Number	Accept total create maximum amount of the PD	Integer
RAID5	Support RAID5 function	Boolean
RAID6	Support RAID6 function	Boolean
Export VD via NVMe-oF	Support Export NVMe-of function	Boolean
Multiple Controller Support	Support Multiple Controller function	Boolean

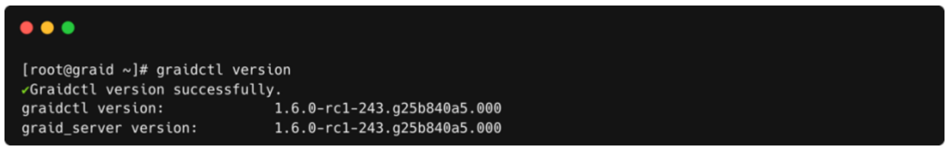
Checking the SupremeRAID™ Driver Version

You can prompt the version command to check graidservice information.

To obtain the graidservice version information, issue the following command:

```
$ sudo graidctl version [flags]
```

Output example:



```
[root@graid ~]# graidctl version
✓Graidctl version successfully.
graidctl version:            1.6.0-rc1-243.g25b840a5.000
graid_server version:        1.6.0-rc1-243.g25b840a5.000
```

Viewing Host Drive Information

Listing NVMe Drives

To list all the directly attached NVMe drives or NVMe-oF target drives that can be used to create physical drives, issue the following command:

```
$ sudo graidctl list nvme_drive [flags]
```

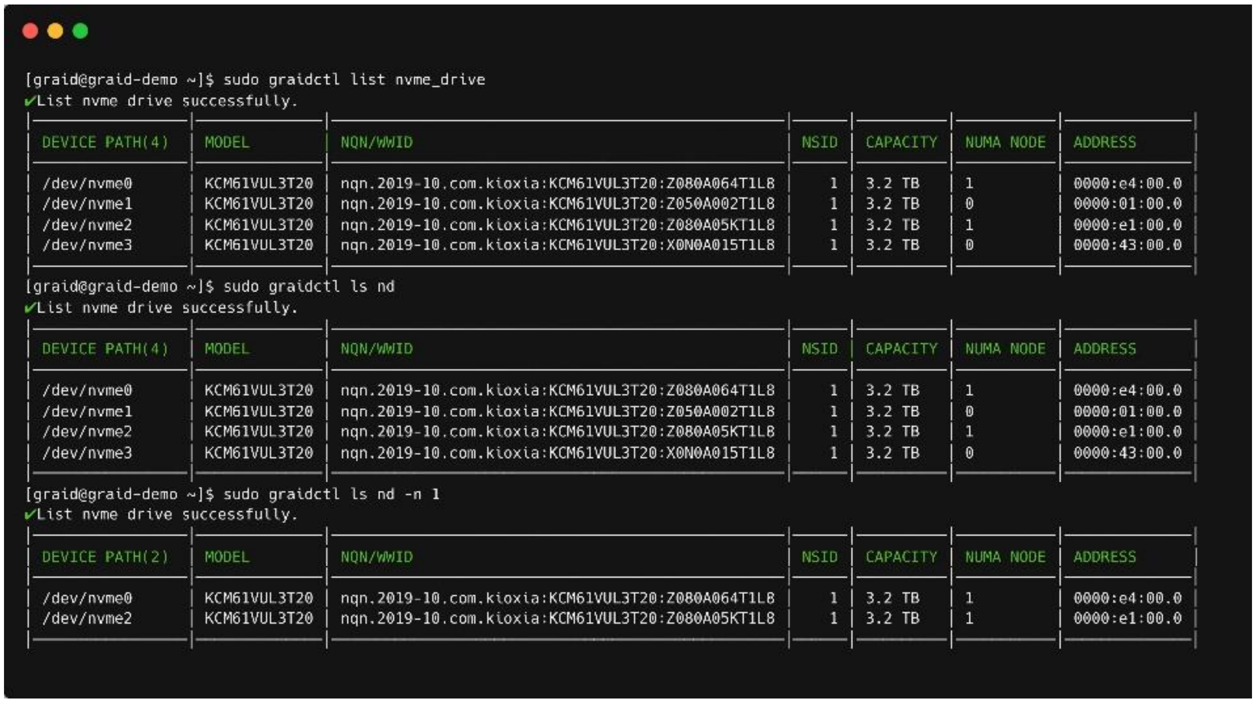
OR

```
$ sudo graidctl ls nd [flags]
```

Related command flags:

Flag	Description
-h, --help	Help for the list nvme_drive command
-n, --numa-node	[int32] Filter by numa node Default: -1

Output example:



```
[graid@graid-demo ~]$ sudo graidctl list nvme_drive
✔List nvme drive successfully.
|──────────────────|──────────────|──────────────────────────────────────────────────|──────|───────────|───────────|──────────────|
| DEVICE PATH(4)   | MODEL        | NQN/WWID                                         | NSID | CAPACITY  | NUMA NODE | ADDRESS      |
|──────────────────|──────────────|──────────────────────────────────────────────────|──────|───────────|───────────|──────────────|
| /dev/nvme0       | KCM61VUL3T20 | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A064T1L8 |    1 | 3.2 TB    | 1         | 0000:e4:00.0 |
| /dev/nvme1       | KCM61VUL3T20 | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z050A002T1L8 |    1 | 3.2 TB    | 0         | 0000:01:00.0 |
| /dev/nvme2       | KCM61VUL3T20 | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A05KT1L8 |    1 | 3.2 TB    | 1         | 0000:e1:00.0 |
| /dev/nvme3       | KCM61VUL3T20 | nqn.2019-10.com.kioxia:KCM61VUL3T20:X0N0A015T1L8 |    1 | 3.2 TB    | 0         | 0000:43:00.0 |
|──────────────────|──────────────|──────────────────────────────────────────────────|──────|───────────|───────────|──────────────|
[graid@graid-demo ~]$ sudo graidctl ls nd
✔List nvme drive successfully.
|──────────────────|──────────────|──────────────────────────────────────────────────|──────|───────────|───────────|──────────────|
| DEVICE PATH(4)   | MODEL        | NQN/WWID                                         | NSID | CAPACITY  | NUMA NODE | ADDRESS      |
|──────────────────|──────────────|──────────────────────────────────────────────────|──────|───────────|───────────|──────────────|
| /dev/nvme0       | KCM61VUL3T20 | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A064T1L8 |    1 | 3.2 TB    | 1         | 0000:e4:00.0 |
| /dev/nvme1       | KCM61VUL3T20 | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z050A002T1L8 |    1 | 3.2 TB    | 0         | 0000:01:00.0 |
| /dev/nvme2       | KCM61VUL3T20 | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A05KT1L8 |    1 | 3.2 TB    | 1         | 0000:e1:00.0 |
| /dev/nvme3       | KCM61VUL3T20 | nqn.2019-10.com.kioxia:KCM61VUL3T20:X0N0A015T1L8 |    1 | 3.2 TB    | 0         | 0000:43:00.0 |
|──────────────────|──────────────|──────────────────────────────────────────────────|──────|───────────|───────────|──────────────|
[graid@graid-demo ~]$ sudo graidctl ls nd -n 1
✔List nvme drive successfully.
|──────────────────|──────────────|──────────────────────────────────────────────────|──────|───────────|───────────|──────────────|
| DEVICE PATH(2)   | MODEL        | NQN/WWID                                         | NSID | CAPACITY  | NUMA NODE | ADDRESS      |
|──────────────────|──────────────|──────────────────────────────────────────────────|──────|───────────|───────────|──────────────|
| /dev/nvme0       | KCM61VUL3T20 | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A064T1L8 |    1 | 3.2 TB    | 1         | 0000:e4:00.0 |
| /dev/nvme2       | KCM61VUL3T20 | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A05KT1L8 |    1 | 3.2 TB    | 1         | 0000:e1:00.0 |
|──────────────────|──────────────|──────────────────────────────────────────────────|──────|───────────|───────────|──────────────|
```

Output content:

Field	Description
DEVICE PATH	Block device path of the drive
NQN	NVMe Qualified Name of the drive
MODEL	Model number of the drive
CAPACITY	Capacity of the drive
NUMA NODE	NUMA NODE of the drive

Listing SAS/SATA Drives

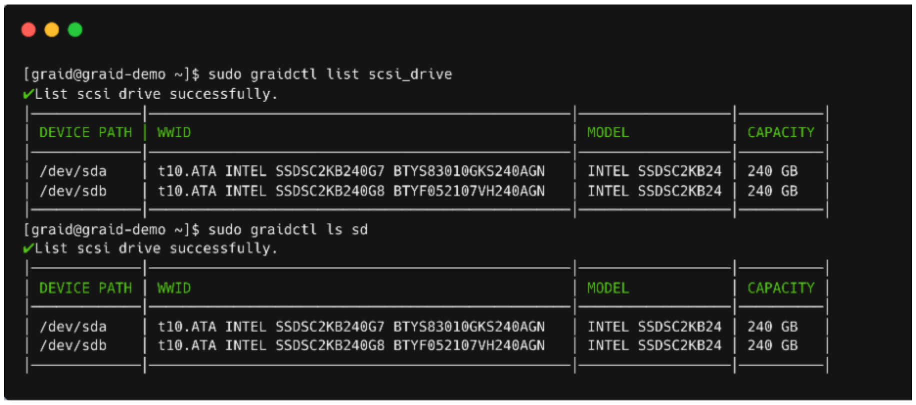
To list all SAS/SATA drives that can be used as physical drives, issue the following command:

```
$ sudo graidctl list scsi_drive
```

OR

```
$ sudo  ls sd
```

Output example:



```
[graid@graid-demo ~]$ sudo graidctl list scsi_drive
✔List scsi drive successfully.
|─────────────|──────────────────────────────────────────────────|──────────────────|──────────|
| DEVICE PATH | WWID                                             | MODEL            | CAPACITY |
|─────────────|──────────────────────────────────────────────────|──────────────────|──────────|
| /dev/sda    | t10.ATA INTEL SSDSC2KB240G7 BTYS83010GKS240AGN   | INTEL SSDSC2KB24 | 240 GB   |
| /dev/sdb    | t10.ATA INTEL SSDSC2KB240G8 BTYF052107VH240AGN   | INTEL SSDSC2KB24 | 240 GB   |
|─────────────|──────────────────────────────────────────────────|──────────────────|──────────|
[graid@graid-demo ~]$ sudo graidctl ls sd
✔List scsi drive successfully.
|─────────────|──────────────────────────────────────────────────|──────────────────|──────────|
| DEVICE PATH | WWID                                             | MODEL            | CAPACITY |
|─────────────|──────────────────────────────────────────────────|──────────────────|──────────|
| /dev/sda    | t10.ATA INTEL SSDSC2KB240G7 BTYS83010GKS240AGN   | INTEL SSDSC2KB24 | 240 GB   |
| /dev/sdb    | t10.ATA INTEL SSDSC2KB240G8 BTYF052107VH240AGN   | INTEL SSDSC2KB24 | 240 GB   |
|─────────────|──────────────────────────────────────────────────|──────────────────|──────────|
```

Output content:

Field	Description
DEVICE PATH	Block device path of the drive
WWID	Worldwide Identification of the drive
MODEL	Model number of the drive
CAPACITY	Capacity of the drive

Managing Physical Drives

Creating a Physical Drive

To create a physical drive, issue the following command:

```
$ sudo  create physical_drive [DEVICE_PATH|NQN|WWID] [flag]
```

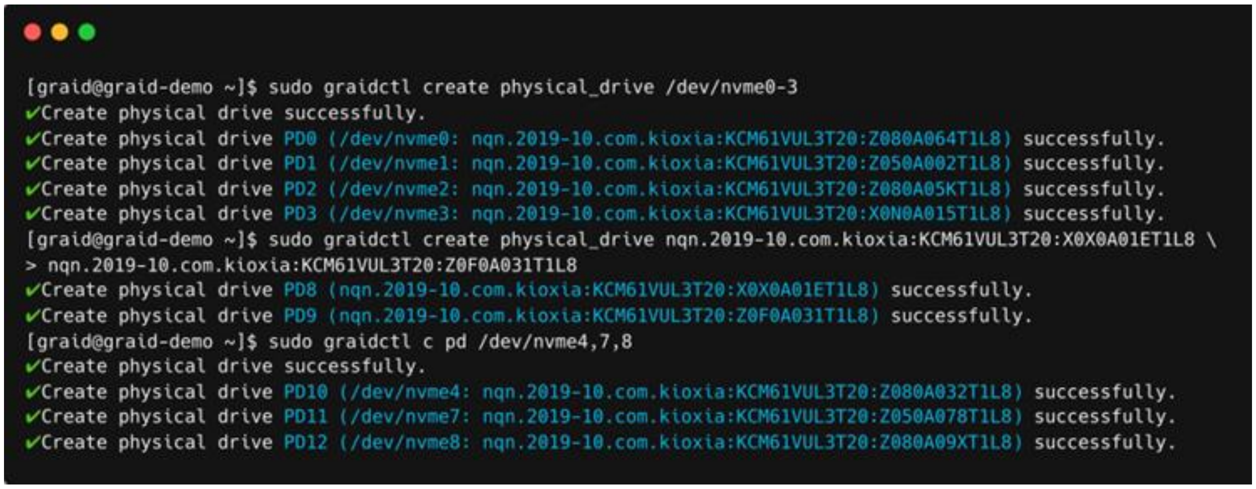
OR

```
$ sudo graidctl c pd [DEVICE_PATH|NQN|WWID] [flag]
```

Related command flags:

Flag	Description
-h, --help	Help for the list physical_drive command
-f, --dblfwd	Door Bell Forwarding

The following figure shows an output example when creating multiple physical drives simultaneously with the device path and NQN.



```
[graid@graid-demo ~]$ sudo graidctl create physical_drive /dev/nvme0-3
✔Create physical drive successfully.
✔Create physical drive PD0 (/dev/nvme0: nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A064T1L8) successfully.
✔Create physical drive PD1 (/dev/nvme1: nqn.2019-10.com.kioxia:KCM61VUL3T20:Z050A002T1L8) successfully.
✔Create physical drive PD2 (/dev/nvme2: nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A05KT1L8) successfully.
✔Create physical drive PD3 (/dev/nvme3: nqn.2019-10.com.kioxia:KCM61VUL3T20:X0N0A015T1L8) successfully.
[graid@graid-demo ~]$ sudo graidctl create physical_drive nqn.2019-10.com.kioxia:KCM61VUL3T20:X0X0A01ET1L8 \
> nqn.2019-10.com.kioxia:KCM61VUL3T20:Z0F0A031T1L8
✔Create physical drive PD8 (nqn.2019-10.com.kioxia:KCM61VUL3T20:X0X0A01ET1L8) successfully.
✔Create physical drive PD9 (nqn.2019-10.com.kioxia:KCM61VUL3T20:Z0F0A031T1L8) successfully.
[graid@graid-demo ~]$ sudo graidctl c pd /dev/nvme4,7,8
✔Create physical drive successfully.
✔Create physical drive PD10 (/dev/nvme4: nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A032T1L8) successfully.
✔Create physical drive PD11 (/dev/nvme7: nqn.2019-10.com.kioxia:KCM61VUL3T20:Z050A078T1L8) successfully.
✔Create physical drive PD12 (/dev/nvme8: nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A09XT1L8) successfully.
```

Note: Be sure the system or other applications are not on the physical drive before creating or replacing the drive.

Listing the Physical Drives

To list all of the physical drives, issue the following command:

```
$ sudo graidctl list physical_drive [flag]
```

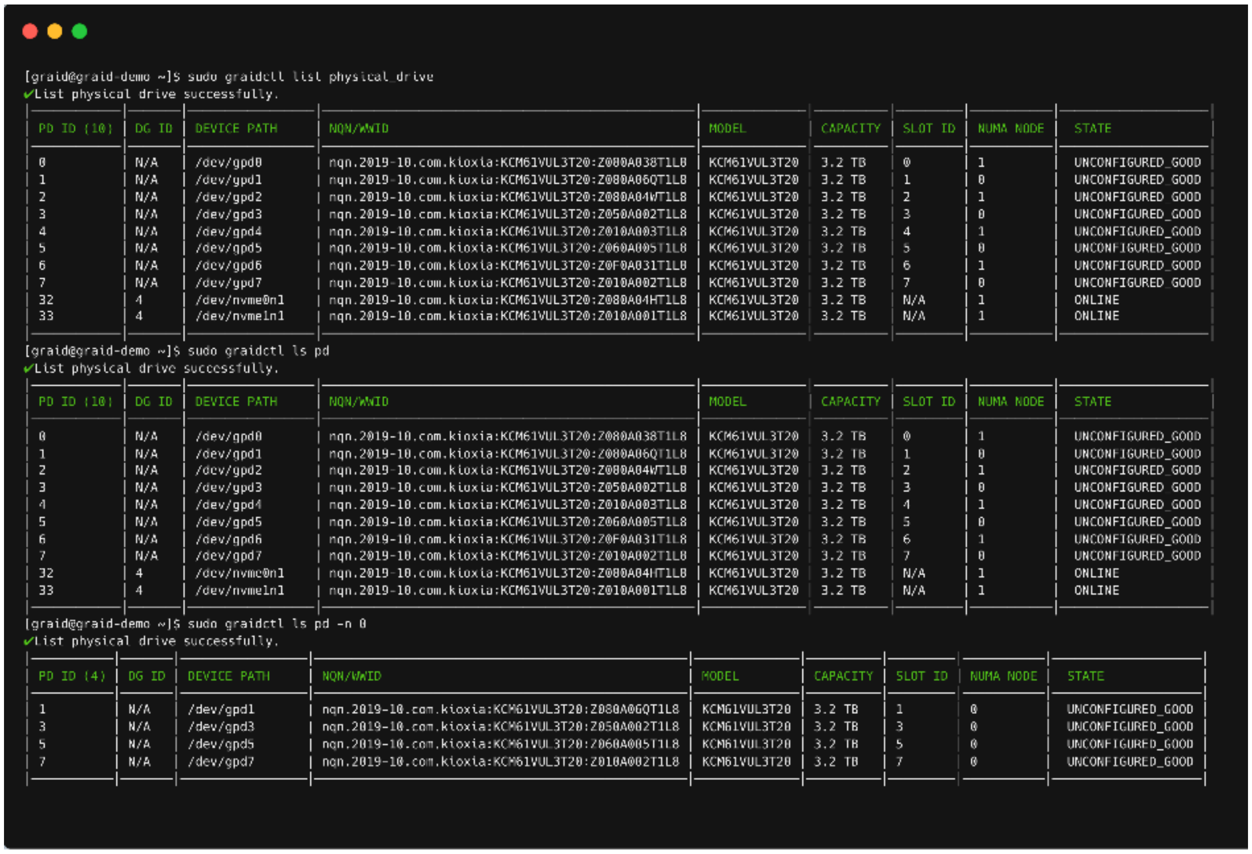
OR

```
$ sudo graidctl ls pd [flag]
```

Related command flags:

Flag	Description
-h, --help	Help for the list physical_drive command
-d, --dg-id	[int32] Filter result by drive group ID Default: -1
-f, --free	List unused PDs
-l, --locating	List locating PDs
-n, --numa-node	[int32] Filter by numa node Default: -1

Output example:



```
[graid@graid-demo ~]$ sudo graidctl list physical_drive
✔List physical drive successfully.
|-------------|--------|---------------|-------------------------------------------------|-------------|----------|---------|-----------|------------------|
| PD ID (10)  | DG ID  | DEVICE PATH   | NQN/WWID                                        | MODEL       | CAPACITY | SLOT ID | NUMA NODE | STATE            |
|-------------|--------|---------------|-------------------------------------------------|-------------|----------|---------|-----------|------------------|
| 0           | N/A    | /dev/gpd0     | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A038T1L8 | KCM61VUL3T20 | 3.2 TB   | 0       | 1         | UNCONFIGURED_GOOD |
| 1           | N/A    | /dev/gpd1     | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A06QT1L8 | KCM61VUL3T20 | 3.2 TB   | 1       | 0         | UNCONFIGURED_GOOD |
| 2           | N/A    | /dev/gpd2     | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A04WT1L8 | KCM61VUL3T20 | 3.2 TB   | 2       | 1         | UNCONFIGURED_GOOD |
| 3           | N/A    | /dev/gpd3     | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z050A002T1L8 | KCM61VUL3T20 | 3.2 TB   | 3       | 0         | UNCONFIGURED_GOOD |
| 4           | N/A    | /dev/gpd4     | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z010A003T1L8 | KCM61VUL3T20 | 3.2 TB   | 4       | 1         | UNCONFIGURED_GOOD |
| 5           | N/A    | /dev/gpd5     | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z060A005T1L8 | KCM61VUL3T20 | 3.2 TB   | 5       | 0         | UNCONFIGURED_GOOD |
| 6           | N/A    | /dev/gpd6     | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z0F0A031T1L8 | KCM61VUL3T20 | 3.2 TB   | 6       | 1         | UNCONFIGURED_GOOD |
| 7           | N/A    | /dev/gpd7     | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z010A002T1L8 | KCM61VUL3T20 | 3.2 TB   | 7       | 0         | UNCONFIGURED_GOOD |
| 32          | 4      | /dev/nvme0n1  | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A04HT1L8 | KCM61VUL3T20 | 3.2 TB   | N/A     | 1         | ONLINE           |
| 33          | 4      | /dev/nvme1n1  | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z010A001T1L8 | KCM61VUL3T20 | 3.2 TB   | N/A     | 1         | ONLINE           |
|-------------|--------|---------------|-------------------------------------------------|-------------|----------|---------|-----------|------------------|
[graid@graid-demo ~]$ sudo graidctl ls pd
✔List physical drive successfully.
|-------------|--------|---------------|-------------------------------------------------|-------------|----------|---------|-----------|------------------|
| PD ID (10)  | DG ID  | DEVICE PATH   | NQN/WWID                                        | MODEL       | CAPACITY | SLOT ID | NUMA NODE | STATE            |
|-------------|--------|---------------|-------------------------------------------------|-------------|----------|---------|-----------|------------------|
| 0           | N/A    | /dev/gpd0     | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A038T1L8 | KCM61VUL3T20 | 3.2 TB   | 0       | 1         | UNCONFIGURED_GOOD |
| 1           | N/A    | /dev/gpd1     | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A06QT1L8 | KCM61VUL3T20 | 3.2 TB   | 1       | 0         | UNCONFIGURED_GOOD |
| 2           | N/A    | /dev/gpd2     | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A04WT1L8 | KCM61VUL3T20 | 3.2 TB   | 2       | 1         | UNCONFIGURED_GOOD |
| 3           | N/A    | /dev/gpd3     | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z050A002T1L8 | KCM61VUL3T20 | 3.2 TB   | 3       | 0         | UNCONFIGURED_GOOD |
| 4           | N/A    | /dev/gpd4     | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z010A003T1L8 | KCM61VUL3T20 | 3.2 TB   | 4       | 1         | UNCONFIGURED_GOOD |
| 5           | N/A    | /dev/gpd5     | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z060A005T1L8 | KCM61VUL3T20 | 3.2 TB   | 5       | 0         | UNCONFIGURED_GOOD |
| 6           | N/A    | /dev/gpd6     | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z0F0A031T1L8 | KCM61VUL3T20 | 3.2 TB   | 6       | 1         | UNCONFIGURED_GOOD |
| 7           | N/A    | /dev/gpd7     | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z010A002T1L8 | KCM61VUL3T20 | 3.2 TB   | 7       | 0         | UNCONFIGURED_GOOD |
| 32          | 4      | /dev/nvme0n1  | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A04HT1L8 | KCM61VUL3T20 | 3.2 TB   | N/A     | 1         | ONLINE           |
| 33          | 4      | /dev/nvme1n1  | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z010A001T1L8 | KCM61VUL3T20 | 3.2 TB   | N/A     | 1         | ONLINE           |
|-------------|--------|---------------|-------------------------------------------------|-------------|----------|---------|-----------|------------------|
[graid@graid-demo ~]$ sudo graidctl ls pd -n 0
✔List physical drive successfully.
|-----------|--------|--------------|-------------------------------------------------|-------------|----------|---------|-----------|------------------|
| PD ID (4) | DG ID  | DEVICE PATH  | NQN/WWID                                        | MODEL       | CAPACITY | SLOT ID | NUMA NODE | STATE            |
|-----------|--------|--------------|-------------------------------------------------|-------------|----------|---------|-----------|------------------|
| 1         | N/A    | /dev/gpd1    | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A06QT1L8 | KCM61VUL3T20 | 3.2 TB   | 1       | 0         | UNCONFIGURED_GOOD |
| 3         | N/A    | /dev/gpd3    | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z050A002T1L8 | KCM61VUL3T20 | 3.2 TB   | 3       | 0         | UNCONFIGURED_GOOD |
| 5         | N/A    | /dev/gpd5    | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z060A005T1L8 | KCM61VUL3T20 | 3.2 TB   | 5       | 0         | UNCONFIGURED_GOOD |
| 7         | N/A    | /dev/gpd7    | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z010A002T1L8 | KCM61VUL3T20 | 3.2 TB   | 7       | 0         | UNCONFIGURED_GOOD |
|-----------|--------|--------------|-------------------------------------------------|-------------|----------|---------|-----------|------------------|
```

Output content:

Field	Description
SLOT ID	Slot ID of the corresponding NVMe/SAS/SATA drive. The PD ID is not related to the SLOT ID. To set the physical drives, use the PD ID.
DG ID	Drive group ID of the physical drive
PD ID	PD ID. The PD ID is a unique ID provided by the SupremeRAID™ driver when the physical drive is created. It is not related to any SSD information such as slot ID or NQN. The PD ID is used for all further operations.
NQN/WWID	NQN or WWID of corresponding NVMe/SAS/SATA drive
MODEL	Model number of the corresponding NVMe/SAS/SATA drive
CAPACITY	Capacity of corresponding NVMe/SAS/SATA drive
NODE	NUMA NODE of the corresponding NVMe/SAS/SATA drive

Field	Description
STATE	State of the physical drive (see the following table).

Physical drive state:

State	Description
ONLINE	Physical drive was added to a drive group and is ready to work.
HOTSPARE	Physical drive is configured as a hot spare drive.
FAILED	Physical drive is detected, but it is not operating normally.
OFFLINE	Physical drive is marked as offline.
REBUILD	Physical drive is being rebuilt.
MISSING	Physical drive cannot be detected.
UNCONFIGURED_GOOD	Physical drive did not join a drive group.
UNCONFIGURED_BAD	Physical drive did not join a drive group and is not operating normally.

Deleting a Physical Drive

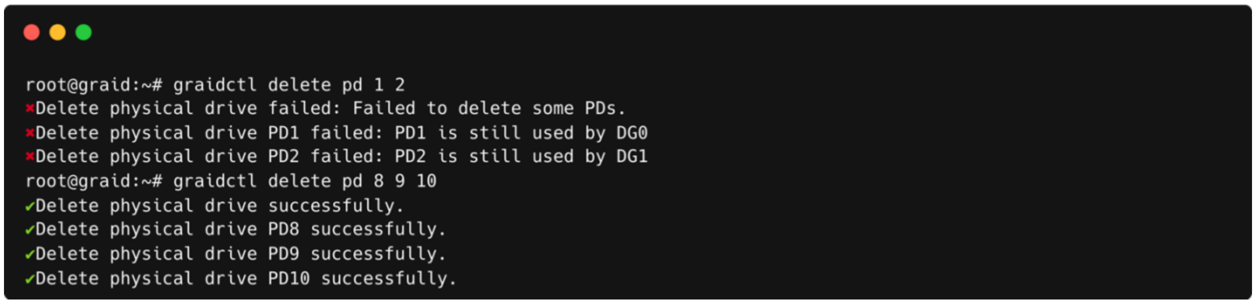
To delete a physical drive, issue the following command:

```
$ sudo graidctl delete physical_drive [PD_ID]
```

OR

```
$ sudo graidctl del pd [PD_ID]
```

The following figure shows an output example for deleting multiple physical drives simultaneously.



```
root@graid:~# graidctl delete pd 1 2
×Delete physical drive failed: Failed to delete some PDs.
×Delete physical drive PD1 failed: PD1 is still used by DG0
×Delete physical drive PD2 failed: PD2 is still used by DG1
root@graid:~# graidctl delete pd 8 9 10
✓Delete physical drive successfully.
✓Delete physical drive PD8 successfully.
✓Delete physical drive PD9 successfully.
✓Delete physical drive PD10 successfully.
```

The output shows that a physical drive cannot be deleted when it is part of a drive group.

Describing a Physical Drive

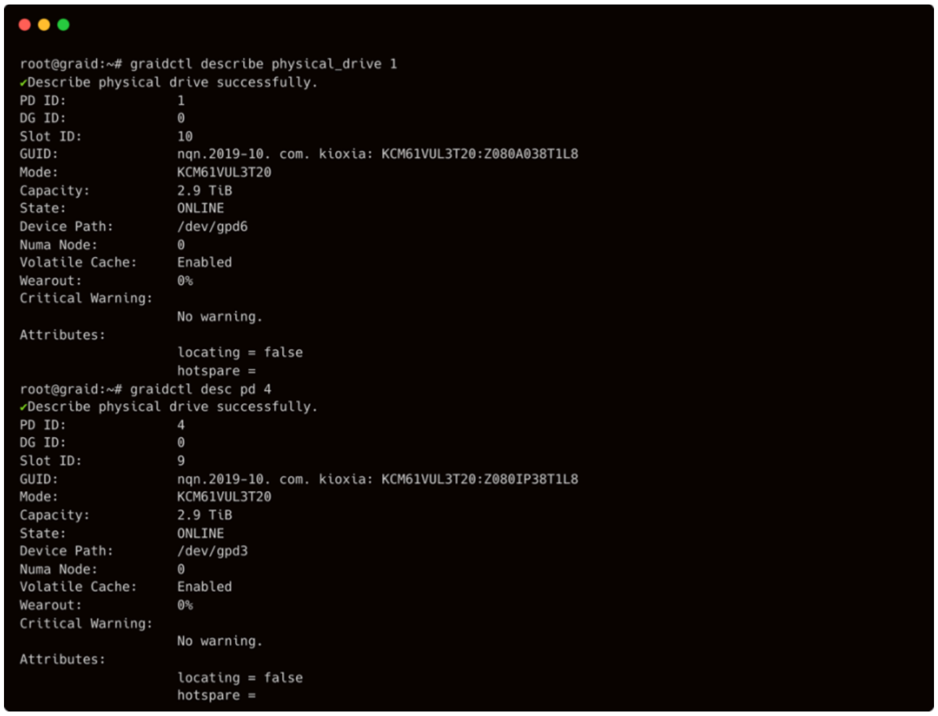
To view detailed information for a physical drive, issue the following command:

```
$ sudo graidctl describe physical_drive [PD_ID]
```

OR

```
$ sudo graidctl desc pd [PD_ID]
```

Output example:



```
root@graid:~# graidctl describe physical_drive 1
✓Describe physical drive successfully.
PD ID:              1
DG ID:              0
Slot ID:            10
GUID:               nqn.2019-10. com. kioxia: KCM61VUL3T20:Z080A038T1L8
Mode:               KCM61VUL3T20
Capacity:           2.9 TiB
State:              ONLINE
Device Path:        /dev/gpd6
Numa Node:          0
Volatile Cache:     Enabled
Wearout:            0%
Critical Warning:
                    No warning.
Attributes:
                    locating = false
                    hotspare =
root@graid:~# graidctl desc pd 4
✓Describe physical drive successfully.
PD ID:              4
DG ID:              0
Slot ID:            9
GUID:               nqn.2019-10. com. kioxia: KCM61VUL3T20:Z080IP38T1L8
Mode:               KCM61VUL3T20
Capacity:           2.9 TiB
State:              ONLINE
Device Path:        /dev/gpd3
Numa Node:          0
Volatile Cache:     Enabled
Wearout:            0%
Critical Warning:
                    No warning.
Attributes:
                    locating = false
                    hotspare =
```

Locating a Physical Drive

To locate a physical drive, issue the following command:

```
$ sudo graidctl edit physical_drive [PD_ID] locating start
```

To stop locating a physical drive, issue the following command:

```
$ sudo graidctl edit physical_drive [PD_ID] locating stop
```

Marking a Physical Drive Online or Offline

To mark a physical drive as online or offline, issue the following command:

```
$ sudo graidctl edit physical_drive [PD_ID] marker [offline|online]
```

Note: Marking a physical drive as offline, even briefly, puts the physical drive in the **REBUILD** state.

Assigning a Hot Spare Drive

To assign a physical drive as global hot spare, issue the following command:

```
$ sudo graidctl edit physical_drive [PD_ID] hotspare global
```

To assign a physical drive as the hot spare for a specific drive group, issue the following command:

```
$ sudo graidctl edit physical_drive [PD_ID] hotspare [DG_ID]
```

To assign a physical drive as a hot spare for multiple drive groups, use a comma (,) to separate the drive group IDs.

Replacing a Nearly Worn-Out or Broken SSD

Note: Make sure the system or other applications are not on the physical drive before creating or replacing the drive.

To replace a nearly worn-out or broken SSD:

Step 1 If the physical drive is in the MISSING or other abnormal state, skip this step. Otherwise, issue the following command to mark the physical drive as bad:

```
$ sudo graidctl edit pd [OLD_PD_ID] marker bad
```

Step 2 Replace the NVMe SSD. The state of the prior physical drive indicates **FAILED**.

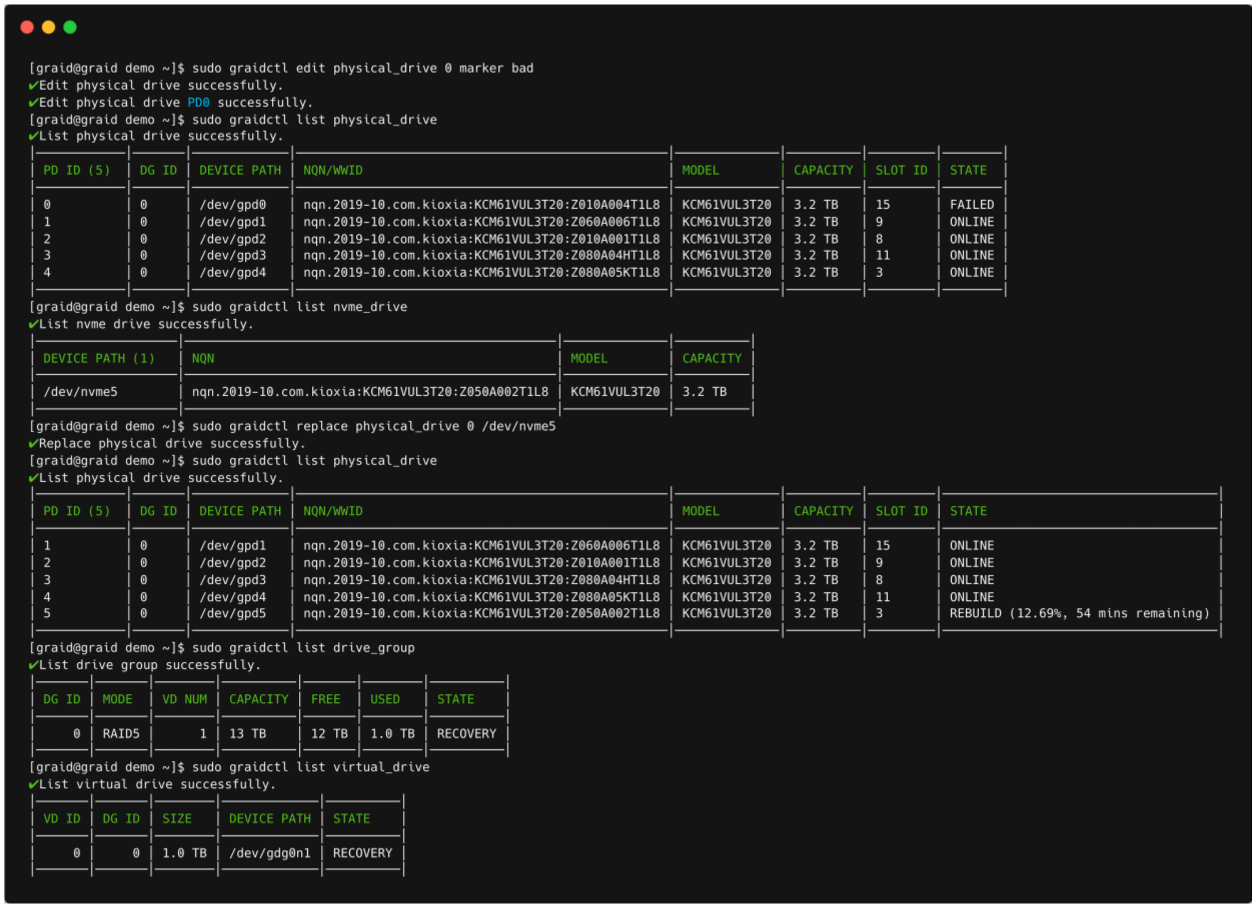
Step 3 Check the NQN of the new SSD.

```
$ sudo graidctl list nvme_drive
```

Step 4 Replace the physical drive.

```
$ sudo graidctl replace physical_drive [OLD_PD_ID]
[DEVICE_PATH|NQN|WWID]
```

Output example:



```
[graid@graid demo ~]$ sudo graidctl edit physical_drive 0 marker bad
✔Edit physical drive successfully.
✔Edit physical drive PD0 successfully.
[graid@graid demo ~]$ sudo graidctl list physical_drive
✔List physical drive successfully.
|───────────|───────|─────────────|──────────────────────────────────────────────|─────────────|──────────|─────────|────────|
| PD ID (5) | DG ID | DEVICE PATH | NQN/WWID                                     | MODEL       | CAPACITY | SLOT ID | STATE  |
|───────────|───────|─────────────|──────────────────────────────────────────────|─────────────|──────────|─────────|────────|
| 0         | 0     | /dev/gpd0   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z010A004T1L8 | KCM61VUL3T20 | 3.2 TB | 15      | FAILED |
| 1         | 0     | /dev/gpd1   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z060A006T1L8 | KCM61VUL3T20 | 3.2 TB | 9       | ONLINE |
| 2         | 0     | /dev/gpd2   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z010A001T1L8 | KCM61VUL3T20 | 3.2 TB | 8       | ONLINE |
| 3         | 0     | /dev/gpd3   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A04HT1L8 | KCM61VUL3T20 | 3.2 TB | 11      | ONLINE |
| 4         | 0     | /dev/gpd4   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A05KT1L8 | KCM61VUL3T20 | 3.2 TB | 3       | ONLINE |
|───────────|───────|─────────────|──────────────────────────────────────────────|─────────────|──────────|─────────|────────|
[graid@graid demo ~]$ sudo graidctl list nvme_drive
✔List nvme drive successfully.
|─────────────────|──────────────────────────────────────────────────|──────────────|──────────|
| DEVICE PATH (1) | NQN                                              | MODEL        | CAPACITY |
|─────────────────|──────────────────────────────────────────────────|──────────────|──────────|
| /dev/nvme5      | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z050A002T1L8 | KCM61VUL3T20 | 3.2 TB   |
|─────────────────|──────────────────────────────────────────────────|──────────────|──────────|
[graid@graid demo ~]$ sudo graidctl replace physical_drive 0 /dev/nvme5
✔Replace physical drive successfully.
[graid@graid demo ~]$ sudo graidctl list physical_drive
✔List physical drive successfully.
|───────────|───────|─────────────|──────────────────────────────────────────────────|──────────────|──────────|─────────|───────────────────────────────────────|
| PD ID (5) | DG ID | DEVICE PATH | NQN/WWID                                         | MODEL        | CAPACITY | SLOT ID | STATE                                 |
|───────────|───────|─────────────|──────────────────────────────────────────────────|──────────────|──────────|─────────|───────────────────────────────────────|
| 1         | 0     | /dev/gpd1   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z060A006T1L8 | KCM61VUL3T20 | 3.2 TB   | 15      | ONLINE                                |
| 2         | 0     | /dev/gpd2   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z010A001T1L8 | KCM61VUL3T20 | 3.2 TB   | 9       | ONLINE                                |
| 3         | 0     | /dev/gpd3   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A04HT1L8 | KCM61VUL3T20 | 3.2 TB   | 8       | ONLINE                                |
| 4         | 0     | /dev/gpd4   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A05KT1L8 | KCM61VUL3T20 | 3.2 TB   | 11      | ONLINE                                |
| 5         | 0     | /dev/gpd5   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z050A002T1L8 | KCM61VUL3T20 | 3.2 TB   | 3       | REBUILD (12.69%, 54 mins remaining)   |
|───────────|───────|─────────────|──────────────────────────────────────────────────|──────────────|──────────|─────────|───────────────────────────────────────|
[graid@graid demo ~]$ sudo graidctl list drive_group
✔List drive group successfully.
|───────|───────|────────|──────────|───────|────────|──────────|
| DG ID | MODE  | VD NUM | CAPACITY | FREE  | USED   | STATE    |
|───────|───────|────────|──────────|───────|────────|──────────|
|     0 | RAID5 |      1 | 13 TB    | 12 TB | 1.0 TB | RECOVERY |
|───────|───────|────────|──────────|───────|────────|──────────|
[graid@graid demo ~]$ sudo graidctl list virtual_drive
✔List virtual drive successfully.
|───────|───────|────────|─────────────|──────────|
| VD ID | DG ID | SIZE   | DEVICE PATH | STATE    |
|───────|───────|────────|─────────────|──────────|
|     0 |     0 | 1.0 TB | /dev/gdg0n1 | RECOVERY |
|───────|───────|────────|─────────────|──────────|
```

Managing Drive Groups

Creating Drive Groups

To create a drive group or groups, issue the following command:

```
$ sudo graidctl create drive_group [RAID_MODE] [PD_IDs] [flag]
```

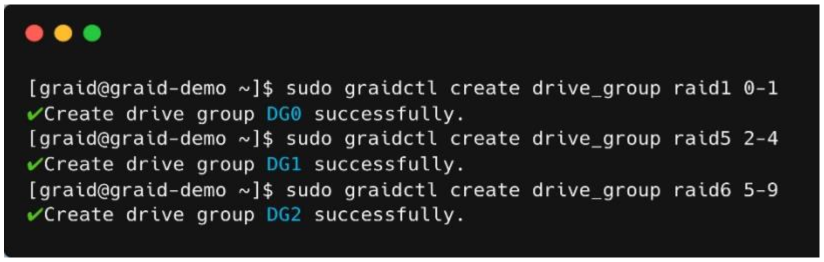
OR

```
$ sudo graidctl c dg [RAID_MODE] [PD_IDs] [flag]
```

Related command flags:

Flag	Description
-h, --help	Help for the create drive_group command
-b, --background-init	Background initialization
-c, --controller	[int32] Specific controller id Default: -1
-f, --force-clean	Ignore initialization (Danger)
-z, --foreground-init	Foreground initialization (Write Zeros)
-s, --strip-size	[uint32] Strip Size (KiB) Values: 4, 8, 16, 32, 64, 128 Default: 4

Output example:



```
[graid@graid-demo ~]$ sudo graidctl create drive_group raid1 0-1
✔Create drive group DG0 successfully.
[graid@graid-demo ~]$ sudo graidctl create drive_group raid5 2-4
✔Create drive group DG1 successfully.
[graid@graid-demo ~]$ sudo graidctl create drive_group raid6 5-9
✔Create drive group DG2 successfully.
```

Required parameters:

Option	Description
RAID_MODE	RAID mode of the drive group. Entries must be all uppercase or all lowercase. For example, RAID6 or raid6 are both correct.
PD_IDs	ID of the physical drive joining the drive group.

Optional parameters:

Option	Description	Behavior
--background - init, -b	Default option. Use standard methods to initialize the drive group. When all the physical drives in the drive group support the de-allocate dataset management command, it is used to synchronize the data, or parity, between the physical drives during the creation of the drive group.	An I/O-capable device path similar to /dev/gdg0n1 is created.
--foreground - init, -z	Initializing foreground. This method writes zeros to whole drives	The virtual drive appears in the system after initialization is complete. Use the following command to check the initialization progress: `$ sudo graidctl list drive_group`
--force - clean, -f	Force bypass initialize. Assumes that the drives are all clean.	The drive group STATE immediately becomes OPTIMAL, indicating that the drive group is available for use.
--controller, -c	Specific controller to control this drive_group. Default: -1, [Int32]	The drive group control by specific controller.
--no-journal	Bypass creating journal space in the drive group.	The drive group would not create journal space.
--strip-size, -s	Strip size of the drive_group. [RAID0,RAID10] Values: 4, 8, 16, 32, 64, 128 Default: 4, [Int32]	Adjust RAID0/RAID10 strip size to a specific size: (4k, 8k, 16k, 32k, 64k, or 128k)

Wait for the drive group initialization to complete. DO NOT power-off or reboot the system when the drive_group state is INIT, RESYNC, or RECOVERY. To check the drive_group state, issue the following command:

```
$ sudo graidctl list drive_group
```

OR

```
$ sudo graidctl ls dg
```

Output content:

Flag	Description
DG ID	Drive group ID
MODE	Drive group RAID mode
VD NUM	Number of virtual drives in the drive group
CAPACITY	Total usable capacity of the drive group
FREE	Unused space of the drive group
USED	Used space of the drive group
CONTROLLER	Drive group controlled by the specific controller
STATE	Drive group state (see the following table)

Drive group state:

State	Description
OFFLINE	Drive group is not working properly. This condition usually occurs when the number of damaged physical drives exceeds the limit.
OPTIMAL	Drive group is in optimal state.
OPTIMAL (!)	Drive group is in optimal state, but found inconsistency data.
OPTIMAL (cc)	Drive group is in optimal state and the consistency check task is ongoing.
OPTIMAL (cc!)	Drive group is in optimal state and the consistency check task is ongoing, but found inconsistent data.
DEGRADED	Drive group is available and ready, but the number of missing or failed physical drives has reached the limit.
PARTIALLY_DEGRADED	Drive group is available and ready for use, but some physical drives are missing or failed.
RECOVERY	Drive group is recovering
FAILED	Drive group is not working normally.
INIT	Drive group is initializing.

State	Description
RESYNC	Drive group is resynchronizing. This condition usually occurs when the system encounters an abnormal crash. Do not replace the physical drive in this state until the resynchronization process completes.
RESCUE	Drive group is in rescue mode.

Deleting Drive Groups

To delete a drive group, issue the following command:

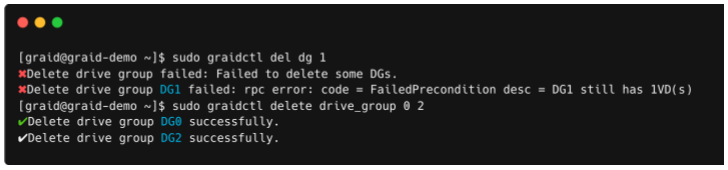
Note: You cannot delete a drive group that contains a virtual drive.

```
$ sudo graidctl delete drive_group [DG_ID] [flag]
```

OR

```
$ sudo graidctl del dg [DG_ID] [flag]
```

In this example, drive group 1 was not deleted because it contains a virtual drive. Drive groups 0 and 2 were deleted successfully.



```
[graid@graid-demo ~]$ sudo graidctl del dg 1
✖Delete drive group failed: Failed to delete some DGs.
✖Delete drive group DG1 failed: rpc error: code = FailedPrecondition desc = DG1 still has 1VD(s)
[graid@graid-demo ~]$ sudo graidctl delete drive_group 0 2
✔Delete drive group DG0 successfully.
✔Delete drive group DG2 successfully.
```

Displaying Drive Group Information

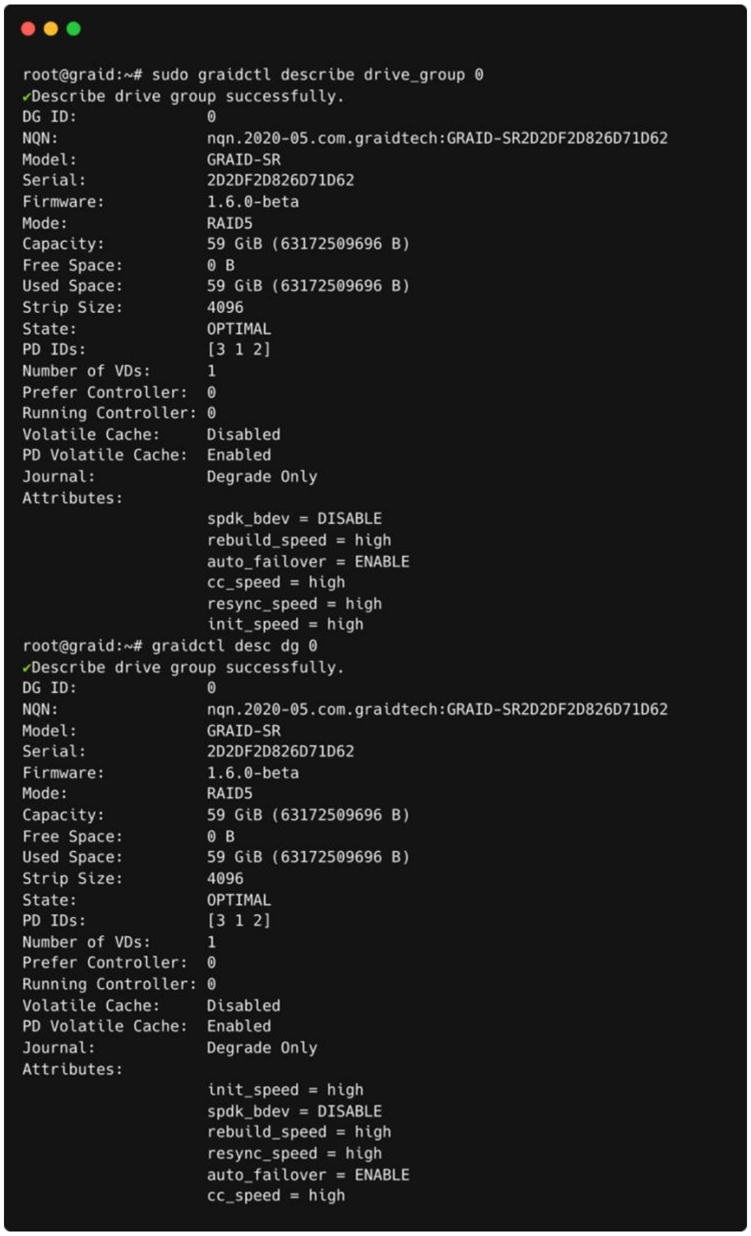
To display detailed information about a drive group, issue the following command:

```
$ sudo graidctl describe drive_group [DG_ID] [flag]
```

OR

```
$ sudo graidctl desc dg [DG_ID] [flag]
```

Output example:



```
root@graid:~# sudo graidctl describe drive_group 0
✓Describe drive group successfully.
DG ID:                0
NQN:                  nqn.2020-05.com.graidtech:GRAID-SR2D2DF2D826D71D62
Model:                GRAID-SR
Serial:               2D2DF2D826D71D62
Firmware:             1.6.0-beta
Mode:                 RAID5
Capacity:             59 GiB (63172509696 B)
Free Space:           0 B
Used Space:           59 GiB (63172509696 B)
Strip Size:           4096
State:                OPTIMAL
PD IDs:               [3 1 2]
Number of VDs:        1
Prefer Controller:    0
Running Controller:   0
Volatile Cache:       Disabled
PD Volatile Cache:    Enabled
Journal:              Degrade Only
Attributes:
                      spdk_bdev = DISABLE
                      rebuild_speed = high
                      auto_failover = ENABLE
                      cc_speed = high
                      resync_speed = high
                      init_speed = high
root@graid:~# graidctl desc dg 0
✓Describe drive group successfully.
DG ID:                0
NQN:                  nqn.2020-05.com.graidtech:GRAID-SR2D2DF2D826D71D62
Model:                GRAID-SR
Serial:               2D2DF2D826D71D62
Firmware:             1.6.0-beta
Mode:                 RAID5
Capacity:             59 GiB (63172509696 B)
Free Space:           0 B
Used Space:           59 GiB (63172509696 B)
Strip Size:           4096
State:                OPTIMAL
PD IDs:               [3 1 2]
Number of VDs:        1
Prefer Controller:    0
Running Controller:   0
Volatile Cache:       Disabled
PD Volatile Cache:    Enabled
Journal:              Degrade Only
Attributes:
                      init_speed = high
                      spdk_bdev = DISABLE
                      rebuild_speed = high
                      resync_speed = high
                      auto_failover = ENABLE
                      cc_speed = high
```

Output content:

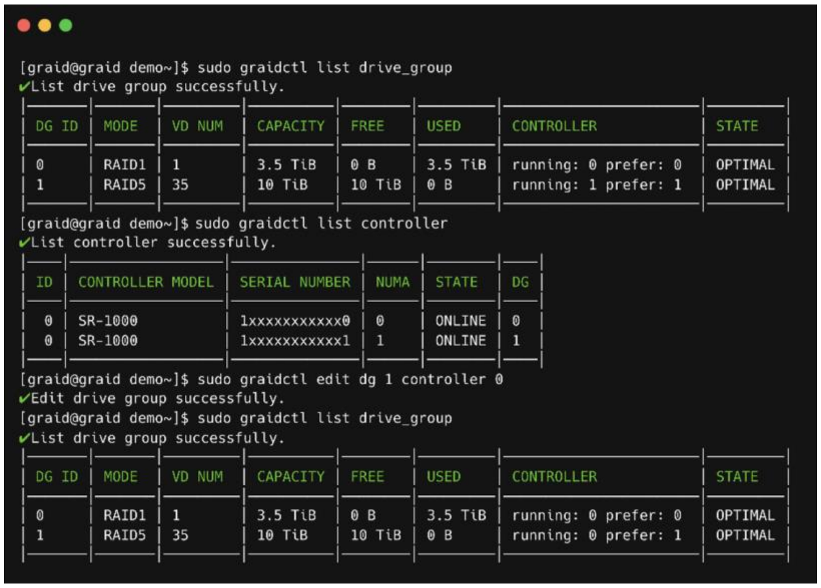
Flag	Description
DG ID	Drive group ID
NQN	Drive group NQN
Model	Model number of the drive group
Serial	Serial number of the drive group
Firmware	Firmware version of the drive group
Mode	RAID mode of the drive group
Capacity	Capacity of the drive
Free Space	Remaining space on the drive
Used Space	Used space of the drive
Strip Size	Strip size (B) of the drive
PD IDs	All PDs of the drive
Number of VDs	Number of VDs of the drive Maximum: 1023
Prefer Controller	Preferred controller of the drive
Running Controller	Running controller number of the drive
Volatile Cache	VMC status for drive group
PD Volatile Cache	VMC status for physical drive
Journal	Journal mode of the drive group
Attributes	Status of all attributes of the drive

Selecting the Controller for a Drive Group

To set the controller to control a drive group, issue the following command:

```
$ sudo graidctl edit drive_group [DG_ID] controller [CX_ID]
```

Output example:



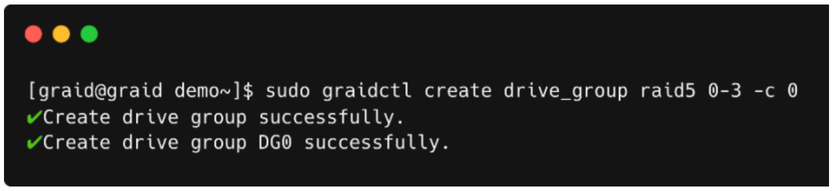
```
[graid@graid demo~]$ sudo graidctl list drive_group
✔List drive group successfully.
|───────|───────|─────────|──────────|────────|─────────|────────────────────────|─────────|
| DG ID | MODE  | VD NUM  | CAPACITY | FREE   | USED    | CONTROLLER             | STATE   |
|───────|───────|─────────|──────────|────────|─────────|────────────────────────|─────────|
| 0     | RAID1 | 1       | 3.5 TiB  | 0 B    | 3.5 TiB | running: 0 prefer: 0   | OPTIMAL |
| 1     | RAID5 | 35      | 10 TiB   | 10 TiB | 0 B     | running: 1 prefer: 1   | OPTIMAL |
|───────|───────|─────────|──────────|────────|─────────|────────────────────────|─────────|
[graid@graid demo~]$ sudo graidctl list controller
✔List controller successfully.
|────|──────────────────|───────────────|──────|────────|────|
| ID | CONTROLLER MODEL | SERIAL NUMBER | NUMA | STATE  | DG |
|────|──────────────────|───────────────|──────|────────|────|
|  0 | SR-1000          | 1xxxxxxxxxxx0 | 0    | ONLINE | 0  |
|  0 | SR-1000          | 1xxxxxxxxxxx1 | 1    | ONLINE | 1  |
|────|──────────────────|───────────────|──────|────────|────|
[graid@graid demo~]$ sudo graidctl edit dg 1 controller 0
✔Edit drive group successfully.
[graid@graid demo~]$ sudo graidctl list drive_group
✔List drive group successfully.
|───────|───────|─────────|──────────|────────|─────────|────────────────────────|─────────|
| DG ID | MODE  | VD NUM  | CAPACITY | FREE   | USED    | CONTROLLER             | STATE   |
|───────|───────|─────────|──────────|────────|─────────|────────────────────────|─────────|
| 0     | RAID1 | 1       | 3.5 TiB  | 0 B    | 3.5 TiB | running: 0 prefer: 0   | OPTIMAL |
| 1     | RAID5 | 35      | 10 TiB   | 10 TiB | 0 B     | running: 0 prefer: 1   | OPTIMAL |
|───────|───────|─────────|──────────|────────|─────────|────────────────────────|─────────|
```

Assigning a Controller to a Drive Group

To assign a controller to control a drive group, issue the following command:

```
$ sudo graidctl create drive_group [RAID_Type] [PD_IDs] -c [CX_ID]
```

Output example:



```
[graid@graid demo~]$ sudo graidctl create drive_group raid5 0-3 -c 0
✔Create drive group successfully.
✔Create drive group DG0 successfully.
```

Managing Background Task Speed

To set the background task speed for a drive group, issue the following command:

```
$ sudo graidctl edit drive_group [DG_ID] rebuild_speed {low|normal|high}
```

Locating the Physical Drives in the Drive Group

To locate all the physical drives in a drive group, issue the following command:

```
$ sudo graidctl edit drive_group [DG_ID] locating start
```

To stop locating all the physical drives in a drive group, issue the following command:

```
$ sudo graidctl edit drive_group [DG_ID] locating stop
```

Degradation and Recovery

If multiple drive groups require simultaneous recovery, the drive groups recover individually.

If multiple physical drives in the same drive group require rebuilding, the physical drives are rebuilt simultaneously.

Rescue Mode

If a damaged drive group is initialized or a recovering drive group encounters an abnormal system crash, the data integrity of the drive group is affected. In this event, the drive group is forced offline to prevent data from being written to the drive group. To read the data for the drive group, force the drive group to go online using Rescue mode.

Note: A drive group in Rescue mode is read-only. Rescue mode cannot be disabled.

To enter rescue mode, issue the following command:

```
$ sudo graidctl edit drive_group [DG_ID] rescue_mode on
```

Managing Virtual Drives

Creating a Virtual Drive

To create a virtual drive, issue the following command:

```
$ sudo graidctl create virtual_drive [DG_ID] [VD_SIZE] [flags]
```

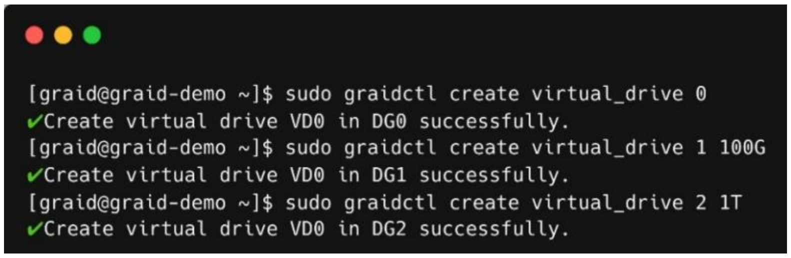
OR

```
$ sudo graidctl c vd [DG_ID] [VD_SIZE] [flags]
```

Related command flags:

Flag	Description
-h, --help	Help for the create virtual_drive command
-s, --serial	[string] Use user-specified serial ID

Output example:



```
[graid@graid-demo ~]$ sudo graidctl create virtual_drive 0
✔Create virtual drive VD0 in DG0 successfully.
[graid@graid-demo ~]$ sudo graidctl create virtual_drive 1 100G
✔Create virtual drive VD0 in DG1 successfully.
[graid@graid-demo ~]$ sudo graidctl create virtual_drive 2 1T
✔Create virtual drive VD0 in DG2 successfully.
```

Note: See Setting Up the Auto-mount File Systems on Linux Using the SupremeRAID™ Driver. It is critically important to follow these instructions to guarantee that the RAID group mounts automatically during system boot and to avoid any improper or unclear shutdown processes that could cause the RAID group to enter resync mode.

Listing Virtual Drives

To list virtual drives, issue the following command:

```
$ sudo graidctl list virtual_drive [flag]
```

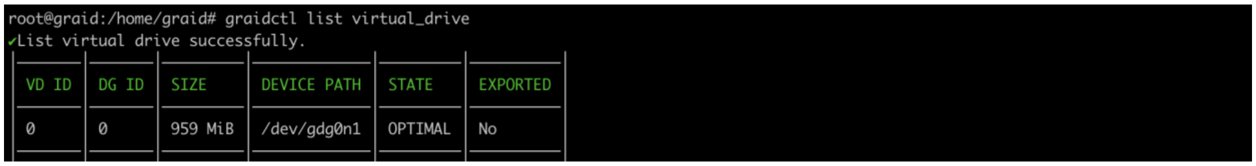
OR

```
$ sudo graidctl ls vd [flag]
```

Related command flags:

Flag	Description
-h, --help	Help for the list virtual_drive command
-d, --dg-id	[string] List VDs of a certain DG ID
-v, --vd-id	[string] List certain VD IDs

Output example:



```
root@graid:/home/graid# graidctl list virtual_drive
✓List virtual drive successfully.
│ VD ID │ DG ID │ SIZE    │ DEVICE PATH │ STATE   │ EXPORTED │
│ 0     │ 0     │ 959 MiB │ /dev/gdg0n1 │ OPTIMAL │ No       │
```

Output content:

Flag	Description
DG ID	Drive group ID
VD ID	Virtual drive ID
SIZE	Usable size of the virtual drive
DEVICE PATH	Device path of the virtual drive
NQN	NQN of the virtual drive
STATE	Virtual drive state - identical to the drive group state (see the following table)
EXPORTED	Shows whether the virtual drive was exported using NVMe-oF or iSCSI

Note: Do not perform I/O before the virtual drive is initialized and the device path (for example, /dev/gdgXnY) is created.

Virtual drive state:

State	Description
OFFLINE	Drive group is not working normally. This condition is usually caused when the number of damaged physical drives exceeds the limit.
OPTIMAL	Drive group is in the optimal state.
PARTIALLY_DEGRADED	Drive group is available and ready for use, but some physical drives are missing or failed.
RECOVERY	Drive group is recovering.
FAILED	Drive group is not working normally.
INIT	Drive group is initializing.
RESYNC	Drive group is resynchronizing. This condition usually occurs when the system encounters an abnormal crash. Do not replace the physical drive in this state until the resynchronization process completes.
RESCUE	Drive group is in rescue mode.

Stripe-cache state:

State	Description
OFFLINE	Stripe cache drive group is OFFLINE.
CLEAN	Stripe cache write-back has finished.
PURGE	Stripe cache is writing data into the virtual drive.
ACTIVE	Stripe cache is in optimal state.

Deleting Virtual Drives

To delete virtual drives, issue the following command:

```
$ sudo graidctl delete virtual_drive [DG_ID] [VD_ID] [flags]
```

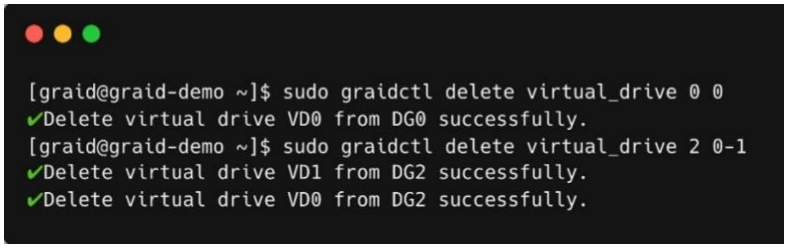
OR

```
$ sudo graidctl del vd [DG_ID] [VD_ID] [flags]
```

Related command flags:

Flag	Description
-h, --help	Help for the delete virtual_drive command
-f, --force	Delete VD forcibly

The following example shows that a virtual drive being used by the application cannot be deleted without adding the force flag.



```
[graid@graid-demo ~]$ sudo graidctl delete virtual_drive 0 0
✔Delete virtual drive VD0 from DG0 successfully.
[graid@graid-demo ~]$ sudo graidctl delete virtual_drive 2 0-1
✔Delete virtual drive VD1 from DG2 successfully.
✔Delete virtual drive VD0 from DG2 successfully.
```

Displaying Virtual Drive Information

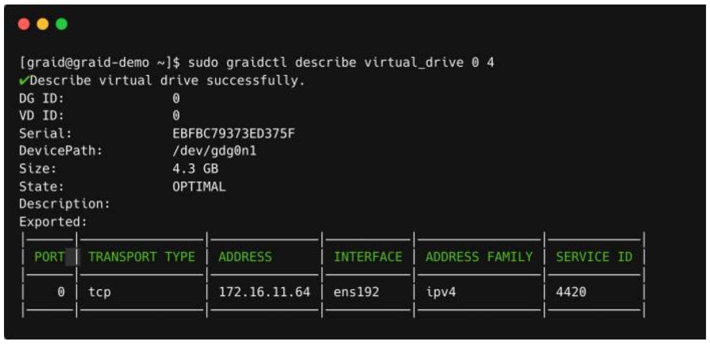
To display detailed information about a virtual drive, issue the following command:

```
$ sudo graidctl describe virtual_drive [DG_ID] [VD_ID] [flags]
```

OR

```
$ sudo graidctl desc vd [DG_ID] [VD_ID] [flags]
```

Output example:



```
[graid@graid-demo ~]$ sudo graidctl describe virtual_drive 0 4
✔Describe virtual drive successfully.
DG ID:              0
VD ID:              0
Serial:             EBFBC79373ED375F
DevicePath:         /dev/gdg0n1
Size:               4.3 GB
State:              OPTIMAL
Description:
Exported:
|-------|----------------|---------------|-----------|----------------|------------|
| PORT  | TRANSPORT TYPE | ADDRESS       | INTERFACE | ADDRESS FAMILY | SERVICE ID |
|-------|----------------|---------------|-----------|----------------|------------|
|     0 | tcp            | 172.16.11.64  | ens192    | ipv4           | 4420       |
|-------|----------------|---------------|-----------|----------------|------------|
```

Setting Up a Stripe Cache

Setting up a stripe cache improves HDD RAID 5 and RAID 6 sequential write performance. To set up a stripe cache:

Step 1 Create a stripe cache with a 4GB virtual drive.

```
$ sudo graidctl create virtual_drive 0 4GB
```

Note: For best practices, use a 4GB stripe whenever possible.

Step 2 Assign a 4GB virtual disk as the stripe cache.

```
$ sudo graidctl edit virtual_drive 0 0 stripecache 1 0
```

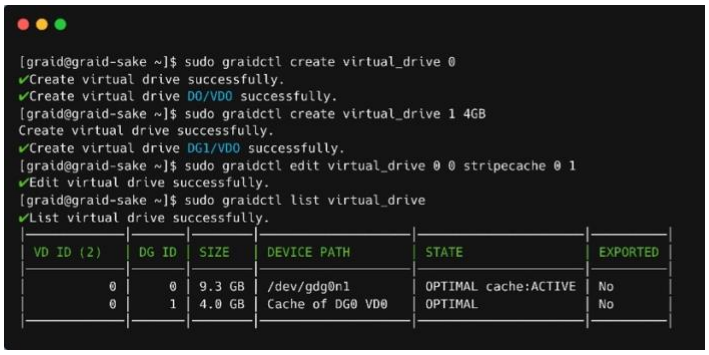
Step 3 Check the stripe cache.

```
$ sudo graidctl list virtual_drive
```

Step 4 To flush the stripe cache, issue the following command.

```
$ sudo graidctl edit vd 0 0 stripecache none
```

The following output is assigned virtual drive is listed as = Stripe Cache = in the DEVICE PATH column.



```
[graid@graid-sake ~]$ sudo graidctl create virtual_drive 0
✔Create virtual drive successfully.
✔Create virtual drive D0/VD0 successfully.
[graid@graid-sake ~]$ sudo graidctl create virtual_drive 1 4GB
Create virtual drive successfully.
✔Create virtual drive DG1/VD0 successfully.
[graid@graid-sake ~]$ sudo graidctl edit virtual_drive 0 0 stripecache 0 1
✔Edit virtual drive successfully.
[graid@graid-sake ~]$ sudo graidctl list virtual_drive
✔List virtual drive successfully.
|─────────────|───────|─────────|──────────────────|─────────────────────|──────────|
| VD ID (2)   | DG ID | SIZE    | DEVICE PATH      | STATE               | EXPORTED |
|─────────────|───────|─────────|──────────────────|─────────────────────|──────────|
|           0 |     0 | 9.3 GB  | /dev/gdg0n1      | OPTIMAL cache:ACTIVE | No      |
|           0 |     1 | 4.0 GB  | Cache of DG0 VD0 | OPTIMAL             | No       |
|─────────────|───────|─────────|──────────────────|─────────────────────|──────────|
```

Managing Controllers

Activating a Controller

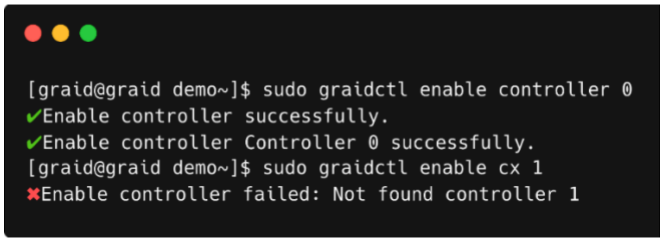
To enable a controller, issue the following command:

```
$ sudo graidctl enable controller [Controller_ID] [flags]
```

OR

```
$ sudo graidctl enable cx [Controller_ID] [flags]
```

Output example:



```
[graid@graid demo~]$ sudo graidctl enable controller 0
✔Enable controller successfully.
✔Enable controller Controller 0 successfully.
[graid@graid demo~]$ sudo graidctl enable cx 1
✖Enable controller failed: Not found controller 1
```

Deactivating a Controller

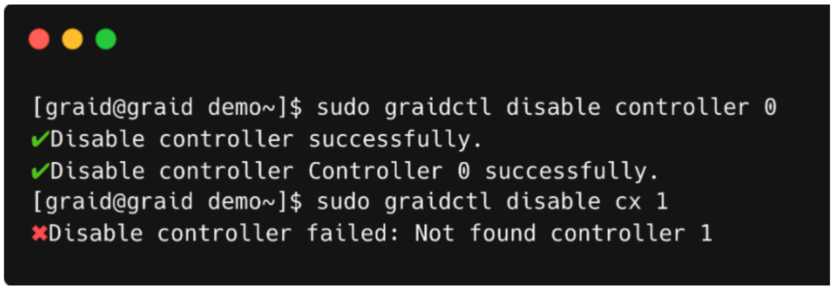
To disable a controller, issue the following command:

```
$ sudo graidctl disable controller [Controller_ID] [flags]
```

OR

```
$ sudo graidctl disable cx [Controller_ID] [flags]
```

Output example:



```
[graid@graid demo~]$ sudo graidctl disable controller 0
✔Disable controller successfully.
✔Disable controller Controller 0 successfully.
[graid@graid demo~]$ sudo graidctl disable cx 1
✖Disable controller failed: Not found controller 1
```

Listing Controllers

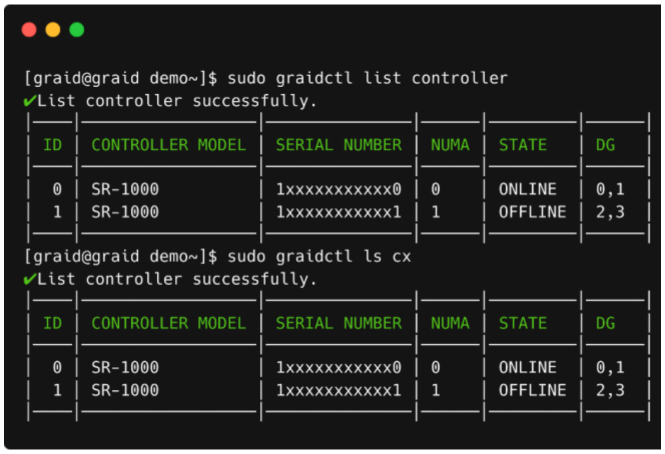
To list controllers, issue the following command:

```
$ sudo graidctl list controller [flag]
```

OR

```
$ sudo graidctl ls cx [flag]
```

Output example:



```
[graid@graid demo~]$ sudo graidctl list controller
✔List controller successfully.
|────|──────────────────|──────────────|──────|─────────|─────|
| ID | CONTROLLER MODEL | SERIAL NUMBER | NUMA | STATE   | DG  |
|────|──────────────────|──────────────|──────|─────────|─────|
|  0 | SR-1000          | 1xxxxxxxxxxx0 | 0    | ONLINE  | 0,1 |
|  1 | SR-1000          | 1xxxxxxxxxxx1 | 1    | OFFLINE | 2,3 |
|────|──────────────────|──────────────|──────|─────────|─────|
[graid@graid demo~]$ sudo graidctl ls cx
✔List controller successfully.
|────|──────────────────|──────────────|──────|─────────|─────|
| ID | CONTROLLER MODEL | SERIAL NUMBER | NUMA | STATE   | DG  |
|────|──────────────────|──────────────|──────|─────────|─────|
|  0 | SR-1000          | 1xxxxxxxxxxx0 | 0    | ONLINE  | 0,1 |
|  1 | SR-1000          | 1xxxxxxxxxxx1 | 1    | OFFLINE | 2,3 |
|────|──────────────────|──────────────|──────|─────────|─────|
```

Display Controller Information

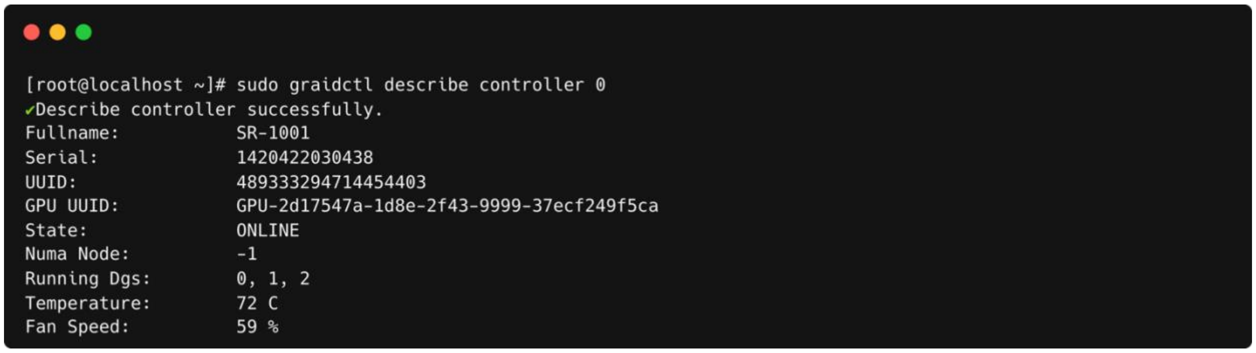
To display the controller information, issue the following command:

```
$ sudo graidctl describe controller [Controller_ID] [flag]
```

OR

```
$ sudo graidctl desc cx [Controller_ID] [flag]
```

Output example:



```
[root@localhost ~]# sudo graidctl describe controller 0
✔Describe controller successfully.
Fullname:           SR-1001
Serial:             1420422030438
UUID:               489333294714454403
GPU UUID:           GPU-2d17547a-1d8e-2f43-9999-37ecf249f5ca
State:              ONLINE
Numa Node:          -1
Running Dgs:        0, 1, 2
Temperature:        72 C
Fan Speed:          59 %
```

Deleting a Controller

To delete a controller, issue the following command:

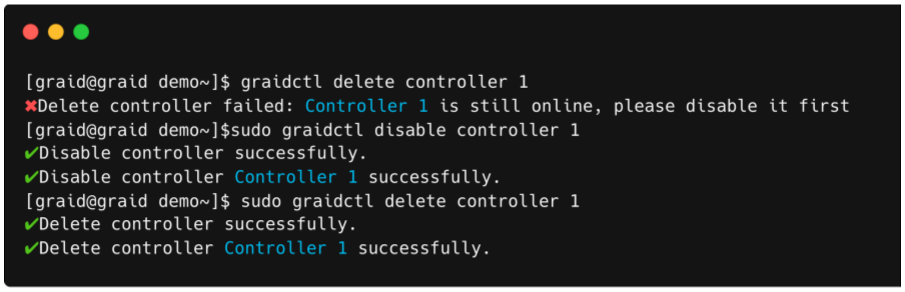
```
$ sudo graidctl delete controller [Controller_ID] [flag]
```

OR

```
$ sudo graidctl del cx [Controller_ID] [flag]
```

Note: You must disable the SupremeRAID™ controller before you can delete it. Disabling the controller prevents further access to it and its associated drives, allowing you to delete the controller safely without affecting the system's operation.

Output example:



```
[graid@graid demo~]$ graidctl delete controller 1
✖Delete controller failed: Controller 1 is still online, please disable it first
[graid@graid demo~]$sudo graidctl disable controller 1
✔Disable controller successfully.
✔Disable controller Controller 1 successfully.
[graid@graid demo~]$ sudo graidctl delete controller 1
✔Delete controller successfully.
✔Delete controller Controller 1 successfully.
```

Replacing a Controller License Key

To replace a controller’s license key, issue the following command:

```
$ sudo graidctl replace controller [Controller_ID] [License_Key] [flags]
```

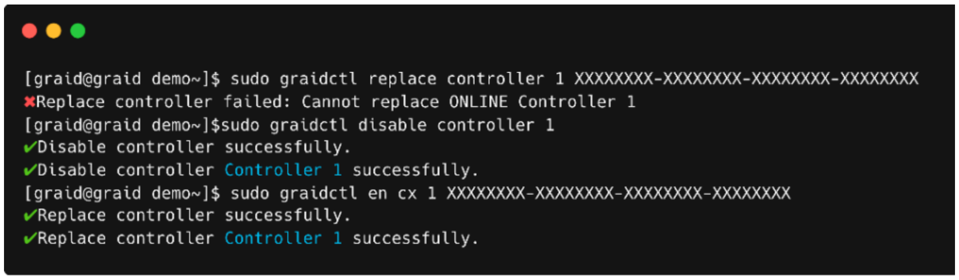
OR

```
$ sudo graidctl en cx [Controller_ID] [License_Key] [flags]
```

Observe the following guidelines when replacing a controller license key:

- To replace the license key for a controller in SupremeRAID™, disable the controller first to ensure that the controller is not in use and can be updated safely. Disabling the controller prevents further access to it or its associated drives, allowing you to safely replace the license key without affecting the operation of the system.
- You cannot replace a license key with one that has a different architecture or supported features. Use the same license key or a compatible replacement to avoid replacement issues.
- If you are replacing a card in the system, deleting any inactive or invalid licenses associated with the old card is essential. Failing to do so may prevent other cards from becoming active, which is crucial for multi-controller systems.

Output example:



```
[graid@graid demo~]$ sudo graidctl replace controller 1 XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX
✖Replace controller failed: Cannot replace ONLINE Controller 1
[graid@graid demo~]$sudo graidctl disable controller 1
✔Disable controller successfully.
✔Disable controller Controller 1 successfully.
[graid@graid demo~]$ sudo graidctl en cx 1 XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX
✔Replace controller successfully.
✔Replace controller Controller 1 successfully.
```

Importing and Controlling MD Bootable NVMe RAIDs

After installing the SupremeRAID™ driver and the graidctl utility, SupremeRAID™ can import and control an MD bootable NVMe RAID. This feature makes it easy to swap drives if a bootable drive malfunctions.

Note: You must disable the SupremeRAID™ controller before you can delete it. Disabling the controller prevents further access to it and its associated drives, allowing you to delete the controller safely without affecting the system's operation. For instructions on setting up the MD bootable NVMe RAID, see Configuring Boot-Drive Devices.

Importing an MD Bootable NVMe RAID

Note: You can import only MD bootable NVMe RAID1.

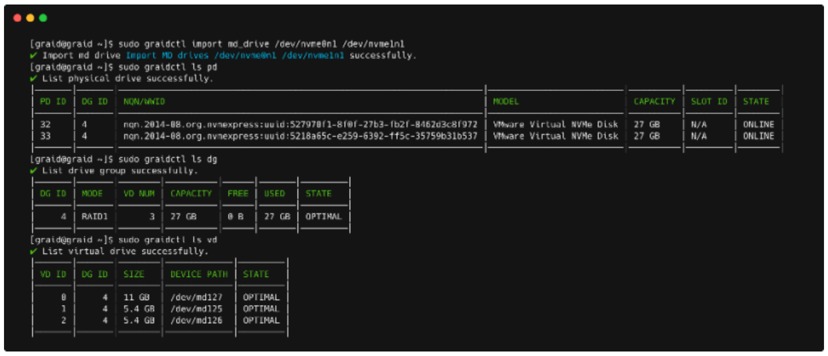
To import an MD bootable NVMe RAID, issue the following command:

```
$ sudo graidctl import md_drive [DEVICE_PATH_0] [DEVICE_PATH_1] [flags]
```

OR

```
$ sudo graidctl imp md [DEVICE_PATH_0] [DEVICE_PATH_1] [flags]
```

Output example:



```
[graid@graid ~]$ sudo graidctl import md_drive /dev/nvme0n1 /dev/nvme1n1
✔ Import md drive Import MD drives /dev/nvme0n1 /dev/nvme1n1 successfully.
[graid@graid ~]$ sudo graidctl ls pd
✔ List physical drive successfully.
│───────│───────│─────────────────────────────────────────────────────────────────│──────────────────────│──────────│─────────│────────│
│ PD ID │ DG ID │ NQN/WWID                                                        │ MODEL                │ CAPACITY │ SLOT ID │ STATE  │
│───────│───────│─────────────────────────────────────────────────────────────────│──────────────────────│──────────│─────────│────────│
│ 32    │ 4     │ nqn.2014-08.org.nvmexpress:uuid:527978f1-8f0f-27b3-fb2f-8462d3c8f972 │ VMware Virtual NVMe Disk │ 27 GB    │ N/A     │ ONLINE │
│ 33    │ 4     │ nqn.2014-08.org.nvmexpress:uuid:5218a65c-e259-6392-ff5c-35759b31b537 │ VMware Virtual NVMe Disk │ 27 GB    │ N/A     │ ONLINE │
│───────│───────│─────────────────────────────────────────────────────────────────│──────────────────────│──────────│─────────│────────│
[graid@graid ~]$ sudo graidctl ls dg
✔ List drive group successfully.
│───────│───────│────────│──────────│──────│───────│─────────│
│ DG ID │ MODE  │ VD NUM │ CAPACITY │ FREE │ USED  │ STATE   │
│───────│───────│────────│──────────│──────│───────│─────────│
│     4 │ RAID1 │      3 │ 27 GB    │ 0 B  │ 27 GB │ OPTIMAL │
│───────│───────│────────│──────────│──────│───────│─────────│
[graid@graid ~]$ sudo graidctl ls vd
✔ List virtual drive successfully.
│───────│───────│────────│─────────────│─────────│
│ VD ID │ DG ID │ SIZE   │ DEVICE PATH │ STATE   │
│───────│───────│────────│─────────────│─────────│
│     0 │     4 │ 11 GB  │ /dev/md127  │ OPTIMAL │
│     1 │     4 │ 5.4 GB │ /dev/md125  │ OPTIMAL │
│     2 │     4 │ 5.4 GB │ /dev/md126  │ OPTIMAL │
│───────│───────│────────│─────────────│─────────│
```

Replacing an MD Bootable NVMe RAID1

Note: You can replace only MD bootable NVMe RAID1.

To replace an MD bootable NVMe RAID 1, replace the old NVMe SSD with the new one. The old physical drive state should indicate MISSING.

```
$ sudo graidctl replace md_drive [OLD_MD)PD_ID] [NEW_DEVICE_PATH] [flags]
```

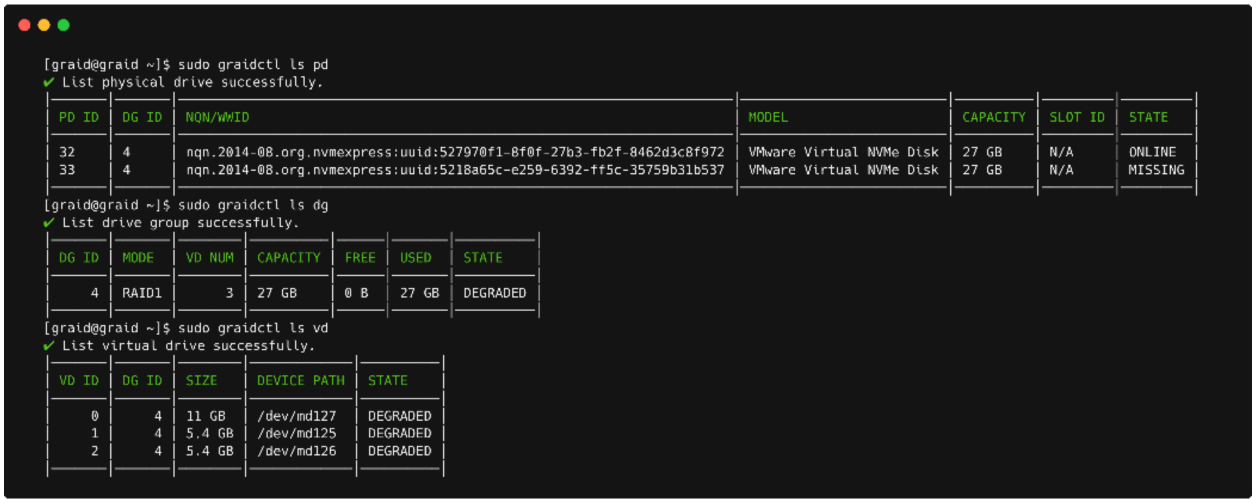
OR

```
$ sudo graidctl en md [OLD_MD)PD_ID] [NEW_DEVICE_PATH] [flags]
```

Related command flags:

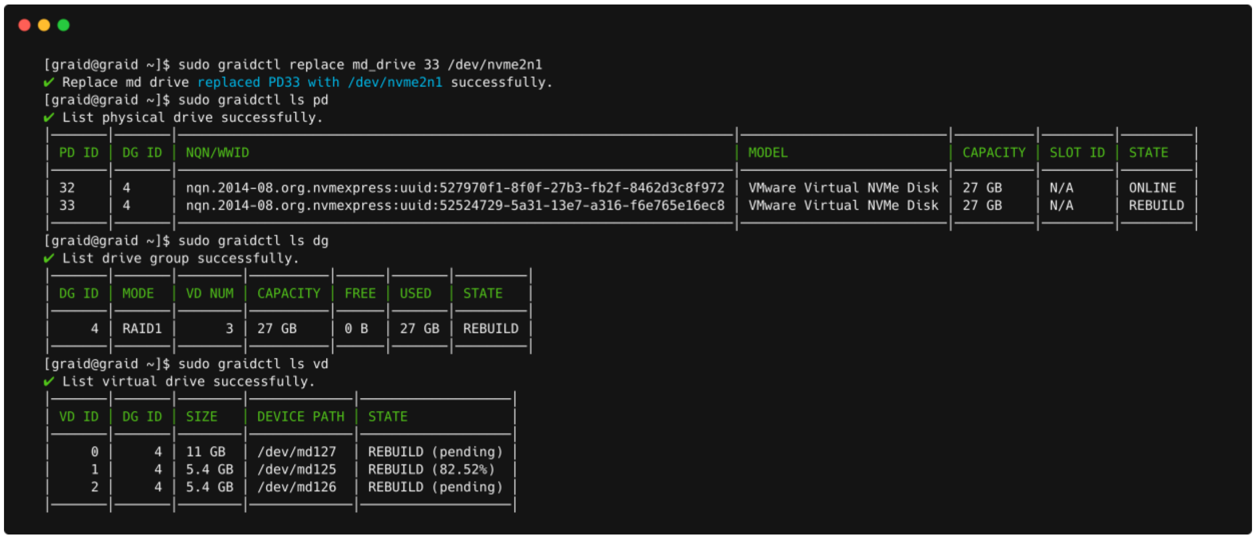
Flag	Description
-h, --help	Help for the replace md_drive command
-f, --force	Replace ONLINE MD forcibly

The following example shows an MD missing.



```
[graid@graid ~]$ sudo graidctl ls pd
✔ List physical drive successfully.
│───────│───────│──────────────────────────────────────────────────────────────────│─────────────────────────│─────────│────────│─────────│
│ PD ID │ DG ID │ NQN/WWID                                                         │ MODEL                   │ CAPACITY│ SLOT ID │ STATE   │
│───────│───────│──────────────────────────────────────────────────────────────────│─────────────────────────│─────────│────────│─────────│
│ 32    │ 4     │ nqn.2014-08.org.nvmexpress:uuid:527970f1-8f0f-27b3-fb2f-8462d3c8f972 │ VMware Virtual NVMe Disk │ 27 GB   │ N/A     │ ONLINE  │
│ 33    │ 4     │ nqn.2014-08.org.nvmexpress:uuid:5218a65c-e259-6392-ff5c-35759b31b537 │ VMware Virtual NVMe Disk │ 27 GB   │ N/A     │ MISSING │
│───────│───────│──────────────────────────────────────────────────────────────────│─────────────────────────│─────────│────────│─────────│
[graid@graid ~]$ sudo graidctl ls dg
✔ List drive group successfully.
│───────│───────│────────│──────────│──────│───────│──────────│
│ DG ID │ MODE  │ VD NUM │ CAPACITY │ FREE │ USED  │ STATE    │
│───────│───────│────────│──────────│──────│───────│──────────│
│     4 │ RAID1 │      3 │ 27 GB    │ 0 B  │ 27 GB │ DEGRADED │
│───────│───────│────────│──────────│──────│───────│──────────│
[graid@graid ~]$ sudo graidctl ls vd
✔ List virtual drive successfully.
│───────│───────│────────│─────────────│──────────│
│ VD ID │ DG ID │ SIZE   │ DEVICE PATH │ STATE    │
│───────│───────│────────│─────────────│──────────│
│     0 │     4 │ 11 GB  │ /dev/md127  │ DEGRADED │
│     1 │     4 │ 5.4 GB │ /dev/md125  │ DEGRADED │
│     2 │     4 │ 5.4 GB │ /dev/md126  │ DEGRADED │
│───────│───────│────────│─────────────│──────────│
```

The following example shows a replaced drive. The bootable RAID group rebuilds immediately after replacing the drive.



```
[graid@graid ~]$ sudo graidctl replace md_drive 33 /dev/nvme2n1
✔ Replace md drive replaced PD33 with /dev/nvme2n1 successfully.
[graid@graid ~]$ sudo graidctl ls pd
✔ List physical drive successfully.
|───────|───────|─────────────────────────────────────────────────────────────────────|──────────────────────────|──────────|─────────|─────────|
| PD ID | DG ID | NQN/WWID                                                            | MODEL                    | CAPACITY | SLOT ID | STATE   |
|───────|───────|─────────────────────────────────────────────────────────────────────|──────────────────────────|──────────|─────────|─────────|
| 32    | 4     | nqn.2014-08.org.nvmexpress:uuid:527970f1-8f0f-27b3-fb2f-8462d3c8f972 | VMware Virtual NVMe Disk | 27 GB    | N/A     | ONLINE  |
| 33    | 4     | nqn.2014-08.org.nvmexpress:uuid:52524729-5a31-13e7-a316-f6e765e16ec8 | VMware Virtual NVMe Disk | 27 GB    | N/A     | REBUILD |
|───────|───────|─────────────────────────────────────────────────────────────────────|──────────────────────────|──────────|─────────|─────────|
[graid@graid ~]$ sudo graidctl ls dg
✔ List drive group successfully.
|───────|───────|────────|──────────|──────|───────|─────────|
| DG ID | MODE  | VD NUM | CAPACITY | FREE | USED  | STATE   |
|───────|───────|────────|──────────|──────|───────|─────────|
|     4 | RAID1 |      3 | 27 GB    | 0 B  | 27 GB | REBUILD |
|───────|───────|────────|──────────|──────|───────|─────────|
[graid@graid ~]$ sudo graidctl ls vd
✔ List virtual drive successfully.
|───────|───────|────────|─────────────|───────────────────|
| VD ID | DG ID | SIZE   | DEVICE PATH | STATE             |
|───────|───────|────────|─────────────|───────────────────|
|     0 |     4 | 11 GB  | /dev/md127  | REBUILD (pending) |
|     1 |     4 | 5.4 GB | /dev/md125  | REBUILD (82.52%)  |
|     2 |     4 | 5.4 GB | /dev/md126  | REBUILD (pending) |
|───────|───────|────────|─────────────|───────────────────|
```

Dismissing an Imported MD Bootable NVMe RAID1

Note: You can dismiss only MD bootable NVMe RAID1.

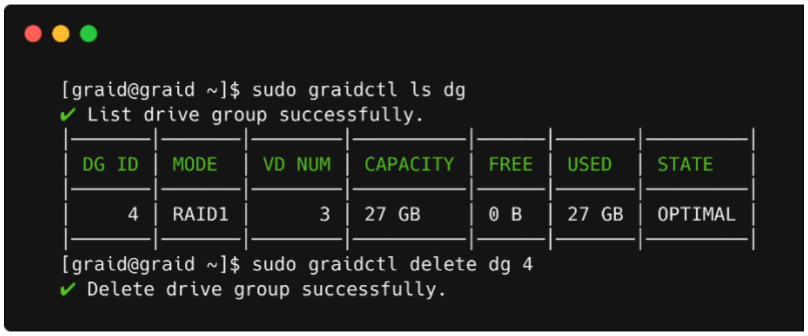
To dismiss an imported MD bootable NVMe RAID 1, issue the following command:

```
$ sudo graidctl delete drive_group [DG_ID] [flags]
```

OR

```
$ sudo graidctl del dg [DG_ID] [flags]
```

Output example:



```
[graid@graid ~]$ sudo graidctl ls dg
✔ List drive group successfully.
|───────|───────|────────|──────────|──────|───────|─────────|
| DG ID | MODE  | VD NUM | CAPACITY | FREE | USED  | STATE   |
|───────|───────|────────|──────────|──────|───────|─────────|
|     4 | RAID1 |      3 | 27 GB    | 0 B  | 27 GB | OPTIMAL |
|───────|───────|────────|──────────|──────|───────|─────────|
[graid@graid ~]$ sudo graidctl delete dg 4
✔ Delete drive group successfully.
```

Adjusting or Updating Configuration Settings for the SupremeRAID™ Add-on

The add-on for SupremeRAID™ provides enhanced configuration options and allows you to fine-tune system settings to meet your specific needs. Follow these steps to ensure that the add-on is configured optimally for maximum system performance.

Editing Configuration Settings

To edit the configuration, issue the following command:

```
$ sudo graidctl edit config [config_name] [value] [flags]
```

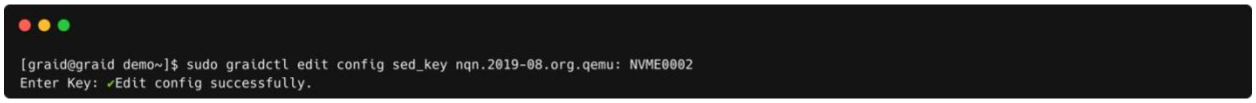
OR

```
$ sudo graidctl e conf [config_name] [value] [flags]
```

Configuration options:

Field	Description
SED_KEY	Add single SED key for specific device

Output example:



```
[graid@graid demo~]$ sudo graidctl edit config sed_key nqn.2019-08.org.qemu: NVME0002
Enter Key: ✓Edit config successfully.
```

Describing Configuration Settings

To describe the configuration, issue the following command:

```
$ sudo graidctl describe config [config_name] [flags]
```

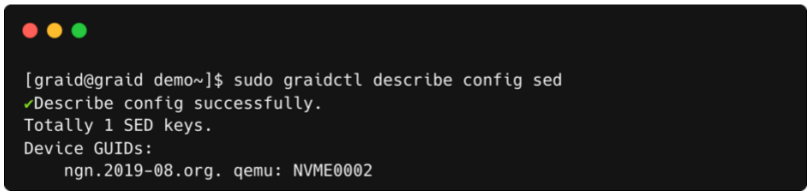
OR

```
$ sudo graidctl desc conf [config_name] [flags]
```

Configuration options:

Field	Description
LED	Obtain the imported LED configuration files
SED	Obtain the SED key information

Output example:



```
[graid@graid demo~]$ sudo graidctl describe config sed
✓Describe config successfully.
Totally 1 SED keys.
Device GUIDs:
    ngn.2019-08.org. qemu: NVME0002
```

Deleting Configuration Settings

To delete the configuration, issue the following command:

```
$ sudo graidctl delete config [config_name] [flags]
```

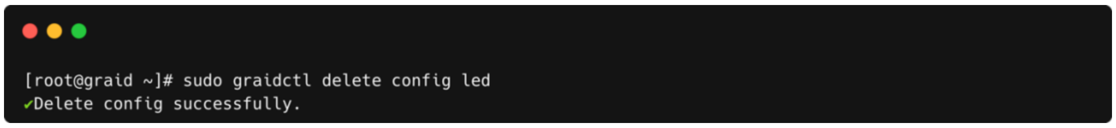
OR

```
$ sudo graidctl del conf [config_name] [flags]
```

Configuration options:

Field	Description
LED	Obtain the imported LED configuration files
SED	Obtain the SED key information

Output example:



```
[root@graid ~]# sudo graidctl delete config led
✓Delete config successfully.
```

Restoring SupremeRAID™ Configuration Settings

To scan all NVMe and SCSI drives and restore the latest SupremeRAID™ configuration, issue the following command:

```
$ sudo graidctl restore config [flags]
```

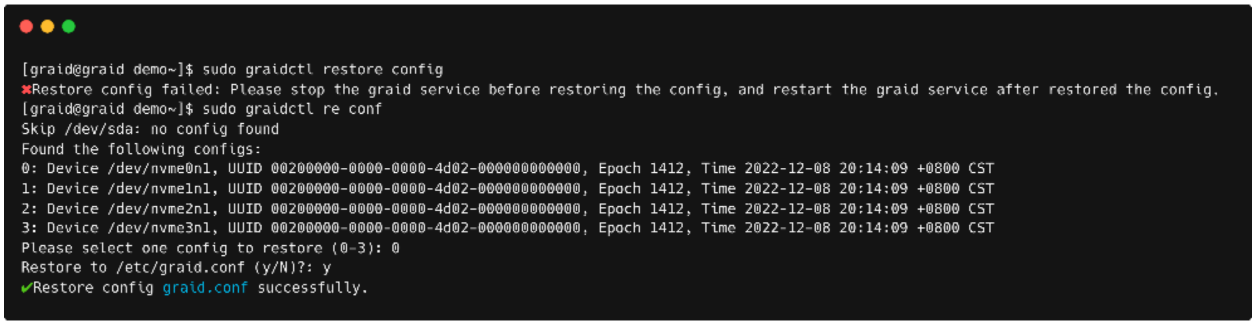
OR

```
$ sudo graidctl re conf [flags]
```

Related command flags:

Flag	Description
-h, --help	Help for the restore config command
-a, --auto	Selects the last configuration automatically

Output example:



```
[graid@graid demo~]$ sudo graidctl restore config
✖Restore config failed: Please stop the graid service before restoring the config, and restart the graid service after restored the config.
[graid@graid demo~]$ sudo graidctl re conf
Skip /dev/sda: no config found
Found the following configs:
0: Device /dev/nvme0n1, UUID 00200000-0000-0000-4d02-000000000000, Epoch 1412, Time 2022-12-08 20:14:09 +0800 CST
1: Device /dev/nvme1n1, UUID 00200000-0000-0000-4d02-000000000000, Epoch 1412, Time 2022-12-08 20:14:09 +0800 CST
2: Device /dev/nvme2n1, UUID 00200000-0000-0000-4d02-000000000000, Epoch 1412, Time 2022-12-08 20:14:09 +0800 CST
3: Device /dev/nvme3n1, UUID 00200000-0000-0000-4d02-000000000000, Epoch 1412, Time 2022-12-08 20:14:09 +0800 CST
Please select one config to restore (0-3): 0
Restore to /etc/graid.conf (y/N)?: y
✔Restore config graid.conf successfully.
```

Managing Events

Listing Events

To check detailed information from record, issue the following command:

```
$ sudo graidctl list event [flags]
```

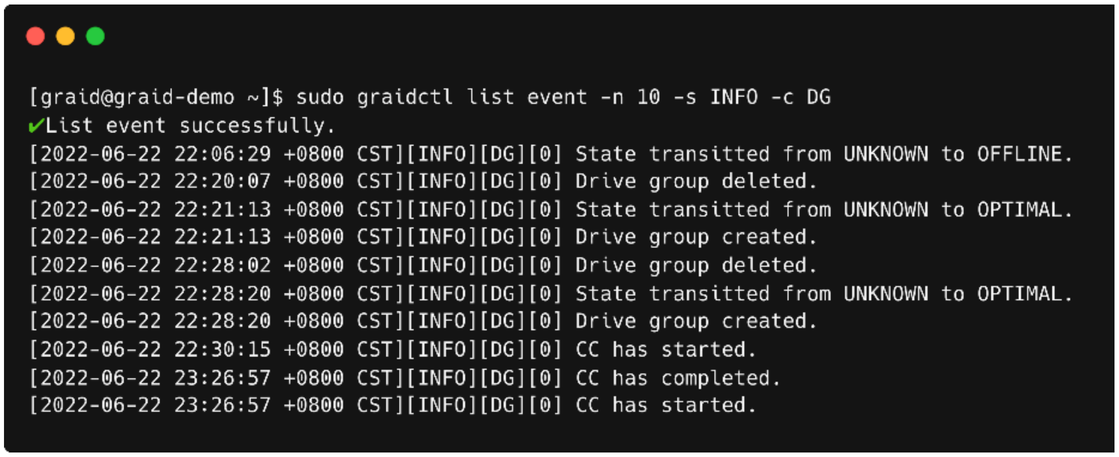
OR

```
$ sudo graidctl ls event [flags]
```

Related command flags:

Flag	Description
-h, --help	Help for the list event command
-c, --component	[string] Filter events by component
-n, --max_entries	[int32] Limit the number of events returned
-o, --output	[string] Output to a file
-s, --severity	[string] Filter events by severity

Output example:



```
[graid@graid-demo ~]$ sudo graidctl list event -n 10 -s INFO -c DG
✔List event successfully.
[2022-06-22 22:06:29 +0800 CST][INFO][DG][0] State transitted from UNKNOWN to OFFLINE.
[2022-06-22 22:20:07 +0800 CST][INFO][DG][0] Drive group deleted.
[2022-06-22 22:21:13 +0800 CST][INFO][DG][0] State transitted from UNKNOWN to OPTIMAL.
[2022-06-22 22:21:13 +0800 CST][INFO][DG][0] Drive group created.
[2022-06-22 22:28:02 +0800 CST][INFO][DG][0] Drive group deleted.
[2022-06-22 22:28:20 +0800 CST][INFO][DG][0] State transitted from UNKNOWN to OPTIMAL.
[2022-06-22 22:28:20 +0800 CST][INFO][DG][0] Drive group created.
[2022-06-22 22:30:15 +0800 CST][INFO][DG][0] CC has started.
[2022-06-22 23:26:57 +0800 CST][INFO][DG][0] CC has completed.
[2022-06-22 23:26:57 +0800 CST][INFO][DG][0] CC has started.
```

Deleting Events

To delete events, issue the following command:

```
$ sudo graidctl delete event [flags]
```

OR

```
$ sudo graidctl del event [flags]
```

Related command flags:

Flag	Description
-h, --help	Help for the delete event command
-d, --date	[string] Delete event entries before the date
-e, --entries	int32] Keep the latest number of entries Default: -1

Managing Remote NVMe-oF Targets

Before you can create physical drives from NVMe-oF devices, you must connect to the NVMe-oF target.

Connecting to a Remote NVMe-oF Target

To connect to a remote NVMe-oF target, issue the following command:

```
$ sudo graidctl connect remote_target [transport type] [addr] [address
family] [port service id]
```

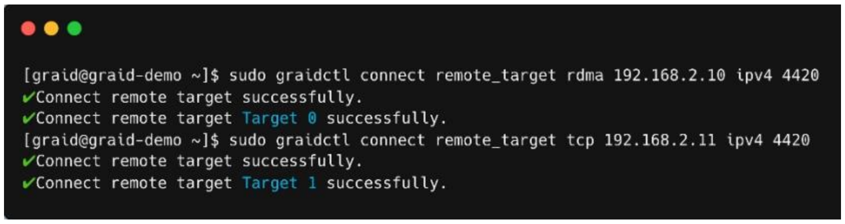
OR

```
$ sudo graidctl con rt [transport type] [addr] [address family] [port service
id]
```

Required parameters:

Option	Description
transport type	Network fabric used for a NVMe-over-Fabrics network. Current string values include: • RDMA = network fabric is an RDMA network (RoCE, iWARP, InfiniBand, basic RDMA, etc.) • TCP = network fabric is a TCP/IP network.
ip address	Network address of the controller
address family	Network address protocol. Current string values include ipv4/ipv6.
port service	Transport service ID

Output example:



```
[graid@graid-demo ~]$ sudo graidctl connect remote_target rdma 192.168.2.10 ipv4 4420
✔Connect remote target successfully.
✔Connect remote target Target 0 successfully.
[graid@graid-demo ~]$ sudo graidctl connect remote_target tcp 192.168.2.11 ipv4 4420
✔Connect remote target successfully.
✔Connect remote target Target 1 successfully.
```

Listing Connected Remote NVMe-oF Targets

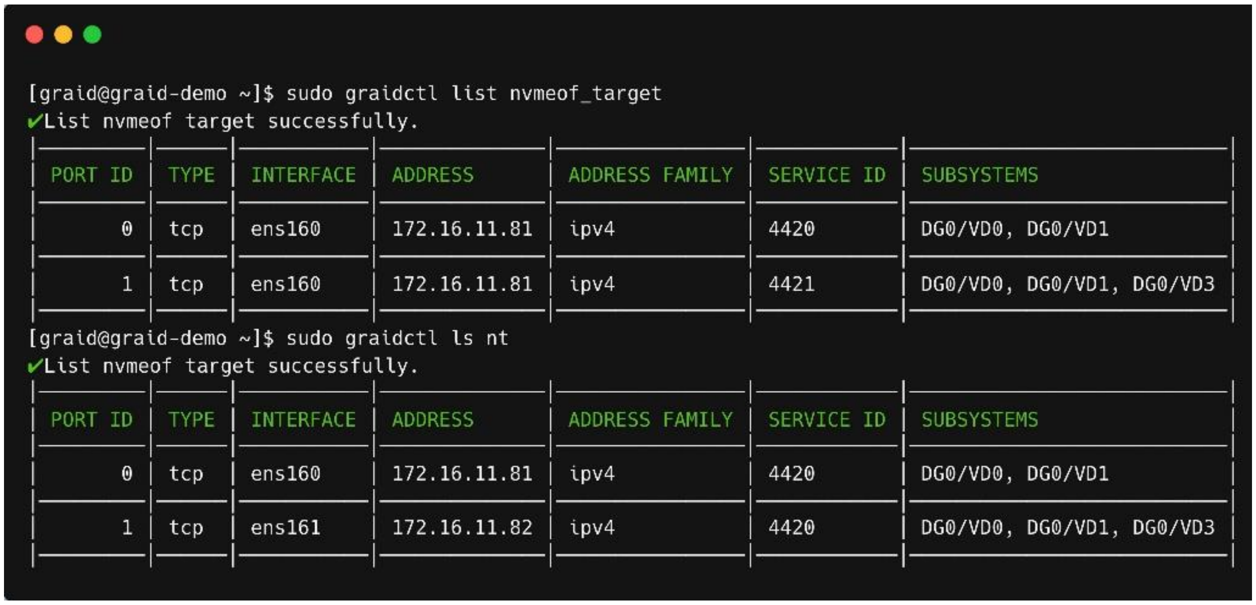
To list all of the connected NVMe-oF targets, issue the following command:

```
$ sudo graidctl list remote_target
```

OR

```
$ sudo graidctl ls rt
```

Output example:



```
[graid@graid-demo ~]$ sudo graidctl list nvmeof_target
✔List nvmeof target successfully.
|---------|------|-----------|--------------|----------------|------------|------------------------------|
| PORT ID | TYPE | INTERFACE | ADDRESS      | ADDRESS FAMILY | SERVICE ID | SUBSYSTEMS                   |
|---------|------|-----------|--------------|----------------|------------|------------------------------|
|       0 | tcp  | ens160    | 172.16.11.81 | ipv4           | 4420       | DG0/VD0, DG0/VD1             |
|---------|------|-----------|--------------|----------------|------------|------------------------------|
|       1 | tcp  | ens160    | 172.16.11.81 | ipv4           | 4421       | DG0/VD0, DG0/VD1, DG0/VD3    |
|---------|------|-----------|--------------|----------------|------------|------------------------------|
[graid@graid-demo ~]$ sudo graidctl ls nt
✔List nvmeof target successfully.
|---------|------|-----------|--------------|----------------|------------|------------------------------|
| PORT ID | TYPE | INTERFACE | ADDRESS      | ADDRESS FAMILY | SERVICE ID | SUBSYSTEMS                   |
|---------|------|-----------|--------------|----------------|------------|------------------------------|
|       0 | tcp  | ens160    | 172.16.11.81 | ipv4           | 4420       | DG0/VD0, DG0/VD1             |
|---------|------|-----------|--------------|----------------|------------|------------------------------|
|       1 | tcp  | ens161    | 172.16.11.82 | ipv4           | 4420       | DG0/VD0, DG0/VD1, DG0/VD3    |
|---------|------|-----------|--------------|----------------|------------|------------------------------|
```

Disconnecting from Remote NVMe-oF Targets

Note: You cannot delete the target when there are physical drives created from the target.

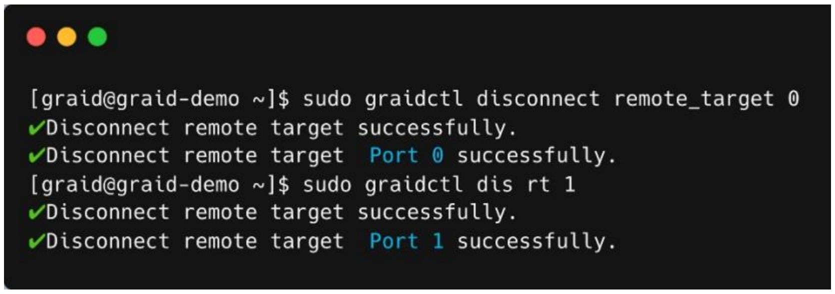
To disconnect from an NVMe-oF target, issue the following command:

```
$ sudo graidctl disconnect remote_target [target id]
```

OR

```
$ sudo graidctl dis rt [target id]
```

Output example:



```
[graid@graid-demo ~]$ sudo graidctl disconnect remote_target 0
✔Disconnect remote target successfully.
✔Disconnect remote target  Port 0 successfully.
[graid@graid-demo ~]$ sudo graidctl dis rt 1
✔Disconnect remote target successfully.
✔Disconnect remote target  Port 1 successfully.
```

Exporting NVMe-oF Target Management

You can export the virtual drive to other initiators.

Creating the NVMe-oF Target Port Service

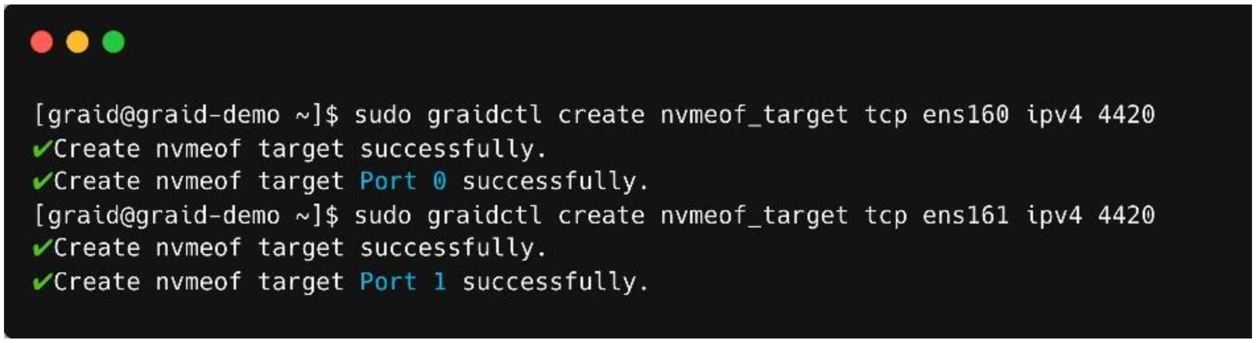
To create the NVMe-oF target port service, issue the following command:

```
$ sudo graidctl create nvmeof_target [tcp|rdma] [interface] [address family]
[srvcid] [flags]
```

OR

```
$ sudo graidctl c nt [tcp|rdma] [interface] [address family] [srvcid] [flags]
```

Output example:



```
[graid@graid-demo ~]$ sudo graidctl create nvmeof_target tcp ens160 ipv4 4420
✔Create nvmeof target successfully.
✔Create nvmeof target Port 0 successfully.
[graid@graid-demo ~]$ sudo graidctl create nvmeof_target tcp ens161 ipv4 4420
✔Create nvmeof target successfully.
✔Create nvmeof target Port 1 successfully.
```

Exporting NVMe-oF Targets

To export NVMe-oF targets using the service port you created, issue the following command:

```
$ sudo graidctl export virtual_drive [DG_ID] [VD_ID] [flags]
```

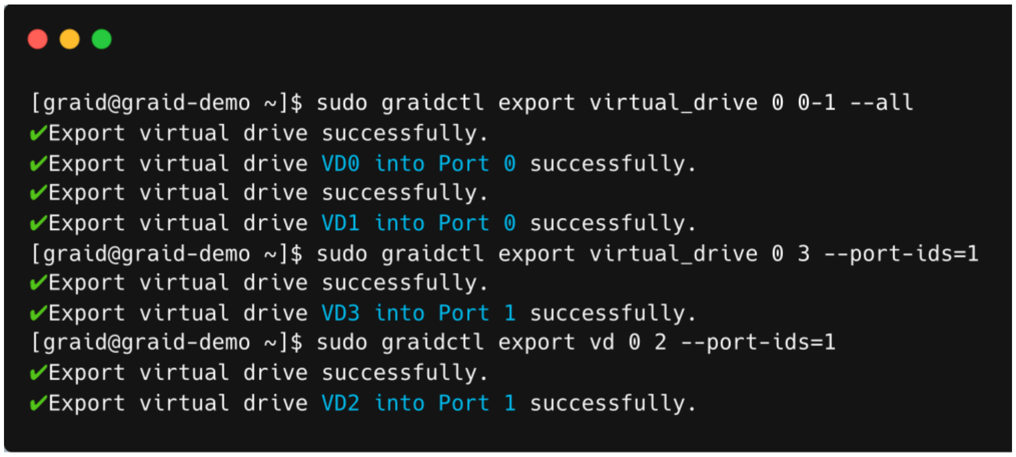
OR

```
$ sudo graidctl exp vd [DG_ID] [VD_ID] [flags]
```

Related command flags:

Flag	Description
-h, --help	Help for the export NVMe-oF targets command
-a, --all	Export all NVMe-oF target into all ports
-p, --port-ids	Port IDs [Int32]

Output example:



```
[graid@graid-demo ~]$ sudo graidctl export virtual_drive 0 0-1 --all
✔Export virtual drive successfully.
✔Export virtual drive VD0 into Port 0 successfully.
✔Export virtual drive successfully.
✔Export virtual drive VD1 into Port 0 successfully.
[graid@graid-demo ~]$ sudo graidctl export virtual_drive 0 3 --port-ids=1
✔Export virtual drive successfully.
✔Export virtual drive VD3 into Port 1 successfully.
[graid@graid-demo ~]$ sudo graidctl export vd 0 2 --port-ids=1
✔Export virtual drive successfully.
✔Export virtual drive VD2 into Port 1 successfully.
```

Listing Created NVMe-oF Targets

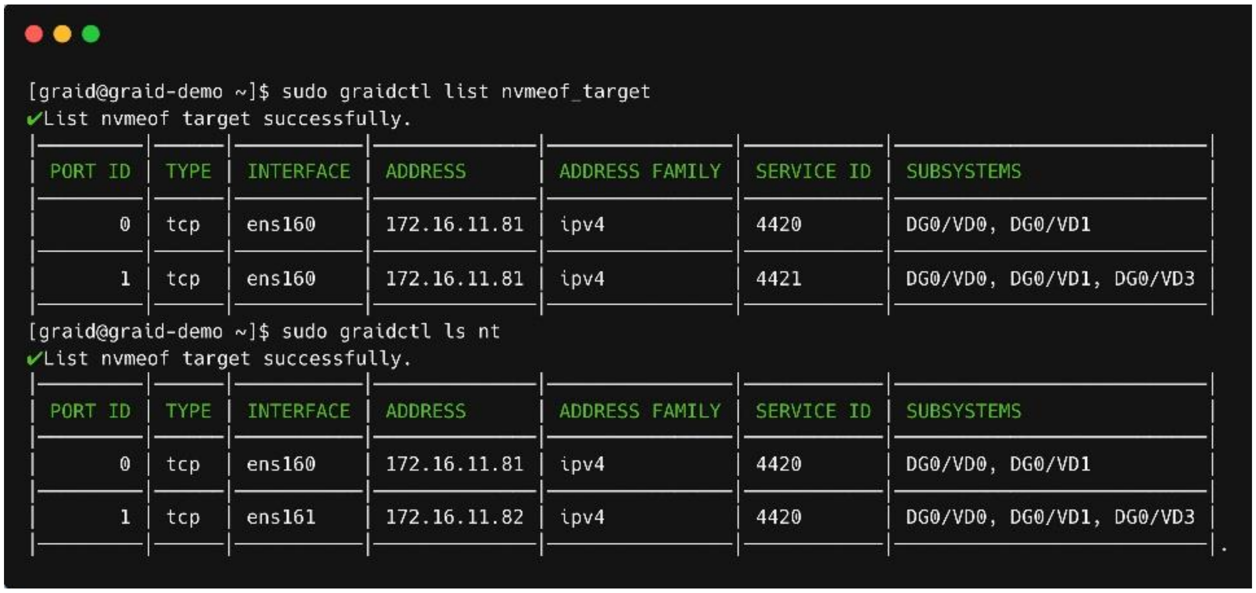
To list all created NVMe-oF target devices, issue the following command:

```
$ sudo graidctl list nvmeof_target
```

OR

```
$ sudo graidctl ls nt
```

Output example:



```
[graid@graid-demo ~]$ sudo graidctl list nvmeof_target
✔List nvmeof target successfully.
| PORT ID | TYPE | INTERFACE | ADDRESS     | ADDRESS FAMILY | SERVICE ID | SUBSYSTEMS                 |
|       0 | tcp  | ens160    | 172.16.11.81 | ipv4          | 4420       | DG0/VD0, DG0/VD1           |
|       1 | tcp  | ens160    | 172.16.11.81 | ipv4          | 4421       | DG0/VD0, DG0/VD1, DG0/VD3  |
[graid@graid-demo ~]$ sudo graidctl ls nt
✔List nvmeof target successfully.
| PORT ID | TYPE | INTERFACE | ADDRESS     | ADDRESS FAMILY | SERVICE ID | SUBSYSTEMS                 |
|       0 | tcp  | ens160    | 172.16.11.81 | ipv4          | 4420       | DG0/VD0, DG0/VD1           |
|       1 | tcp  | ens161    | 172.16.11.82 | ipv4          | 4420       | DG0/VD0, DG0/VD1, DG0/VD3  |
```

Deleting the NVMe-oF Target Port Service Unexporting NVMe-oF Targets

To delete the NVMe-oF target port service, issue the following command:

```
$ sudo graidctl delete nvmeof_target [PORT_ID] [flags]
```

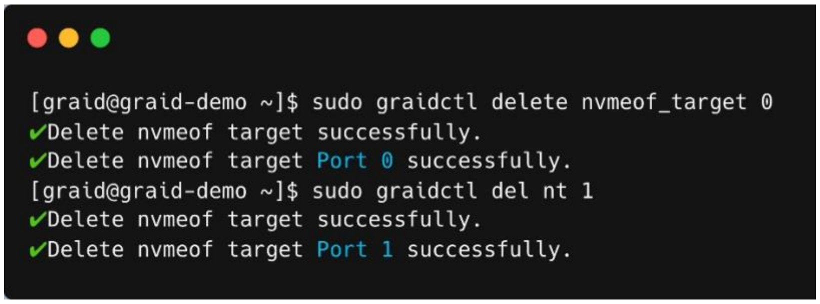
OR

```
$ sudo graidctl del nt [PORT_ID] [flags]
```

Related command flags:

Flag	Description
-h, --help	Help for the delete nvmeof_target command
-f, --force	Force delete ports

Output example:



```
[graid@graid-demo ~]$ sudo graidctl delete nvmeof_target 0
✔Delete nvmeof target successfully.
✔Delete nvmeof target Port 0 successfully.
[graid@graid-demo ~]$ sudo graidctl del nt 1
✔Delete nvmeof target successfully.
✔Delete nvmeof target Port 1 successfully.
```

Unexporting NVMe-oF Targets

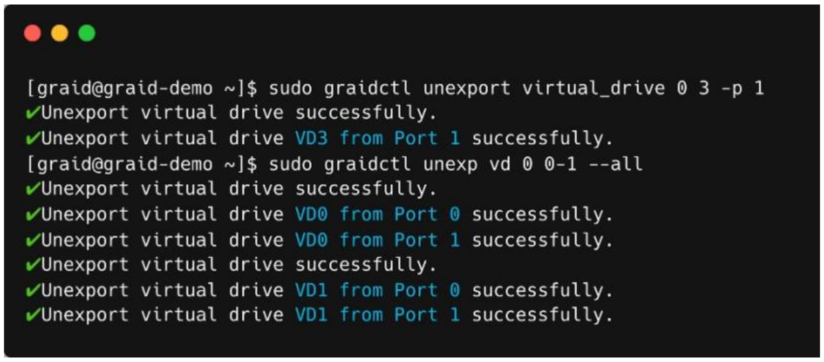
To unexport an NVMe-oF target, issue the following command:

```
$ sudo graidctl unexport virtual_drive [DG_ID] [VD_ID] [flags]
```

OR

```
$ sudo graidctl unexp vd [DG_ID] [VD_ID] [flags]
```

Output example:



```
[graid@graid-demo ~]$ sudo graidctl unexport virtual_drive 0 3 -p 1
✔Unexport virtual drive successfully.
✔Unexport virtual drive VD3 from Port 1 successfully.
[graid@graid-demo ~]$ sudo graidctl unexp vd 0 0-1 --all
✔Unexport virtual drive successfully.
✔Unexport virtual drive VD0 from Port 0 successfully.
✔Unexport virtual drive VD0 from Port 1 successfully.
✔Unexport virtual drive successfully.
✔Unexport virtual drive VD1 from Port 0 successfully.
✔Unexport virtual drive VD1 from Port 1 successfully.
```

Using Consistency Checks to Ensure Data Integrity

The consistency check operation verifies that the data is correct in DGs that use RAID levels 1, 5, 6, and 10. In a system with parity, for example, checking consistency calculates the data on one drive and compares the results to the contents of the parity drive.

Note: You cannot perform a consistency check on RAID 0 because it does not provide data redundancy. Additionally, a consistency check can only run when the DG is in OPTIMAL or PARTIALLY_DEGRADED state.

The consistency check function records all events to the event database, and graidctl provides commands to retrieve the events. The maximum number of event entries is 1,000. The system deletes event entries periodically. You can also delete entries manually.

Starting Consistency Checks Manually

To start a consistency check manually, issue the following command:

```
$ sudo graidctl start consistency_check manual_task [flags]
```

OR

```
$ sudo graidctl start cc [flags]
```

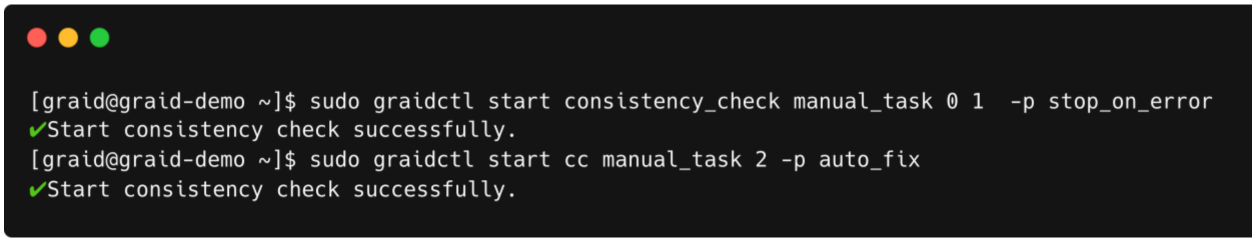
Related command flags:

Flag	Description
-h, --help	Help for the start consistency_check manual command
-p, --policy	[string] Specify CC policy [stop_on_error/auto_fix]

DG state for consistency check: Enabling a consistency check task will add the following annotations beside the output string of the DG state.

DG State	Description
OPTIMAL	Normal state without enabling consistency check
OPTIMAL (!)	Inconsistency found
OPTIMAL (cc)	Consistency check ongoing
OPTIMAL (cc!)	Consistency check ongoing and inconsistency found

Output example:



```
[graid@graid-demo ~]$ sudo graidctl start consistency_check manual_task 0 1  -p stop_on_error
✔Start consistency check successfully.
[graid@graid-demo ~]$ sudo graidctl start cc manual_task 2 -p auto_fix
✔Start consistency check successfully.
```

Stopping Consistency Check

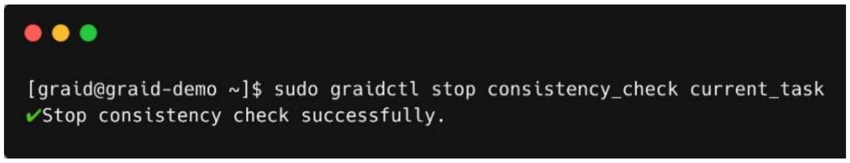
To stop a consistency check task, issue the following command:

```
$ sudo graidctl stop consistency_check current_task [flags]
```

OR

```
$ sudo graidctl stop cc current_task [flags]
```

Output example:



```
[graid@graid-demo ~]$ sudo graidctl stop consistency_check current_task
✔Stop consistency check successfully.
```

Scheduling Consistency Checks

To schedule a consistency check task, issue the following command:

```
$ sudo graidctl set consistency_check schedule_mode
[off|continuously|hourly|daily|weekly|monthly][yyyy/mm/dd] [hh] [flags]
```

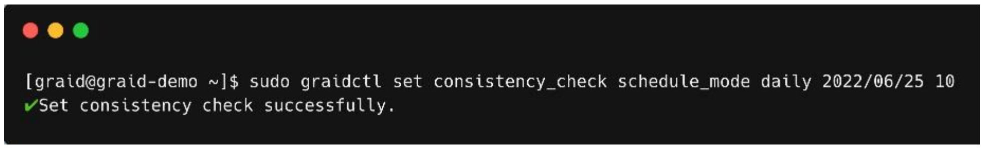
OR

```
$ sudo graidctl set cc schedule_mode
[off|continuously|hourly|daily|weekly|monthly] [yyyy/mm/dd] [hh] [flags]
```

DG State: Enabling a consistency check task adds the following annotations beside the output string of the DG state.

DG State	Description
OPTIMAL	Normal state without enabling consistency check
OPTIMAL (!)	Inconsistency found
OPTIMAL (cc)	Consistency check ongoing
OPTIMAL (cc!)	Consistency check ongoing and inconsistency found

Output example:



```
[graid@graid-demo ~]$ sudo graidctl set consistency_check schedule_mode daily 2022/06/25 10
✔Set consistency check successfully.
```

Viewing Consistency Check Information

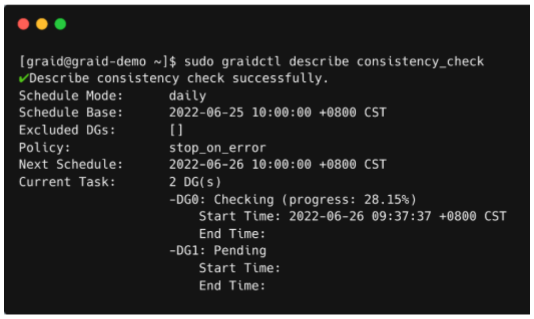
To view detailed consistency check information, issue the following command:

```
$ sudo graidctl describe consistency_check [flags]
```

OR

```
$ sudo graidctl desc consistency_check [flags]
```

Output example:



```
[graid@graid-demo ~]$ sudo graidctl describe consistency_check
✔Describe consistency check successfully.
Schedule Mode:      daily
Schedule Base:      2022-06-25 10:00:00 +0800 CST
Excluded DGs:       []
Policy:             stop_on_error
Next Schedule:      2022-06-26 10:00:00 +0800 CST
Current Task:       2 DG(s)
                    -DG0: Checking (progress: 28.15%)
                        Start Time: 2022-06-26 09:37:37 +0800 CST
                        End Time:
                    -DG1: Pending
                        Start Time:
                        End Time:
```

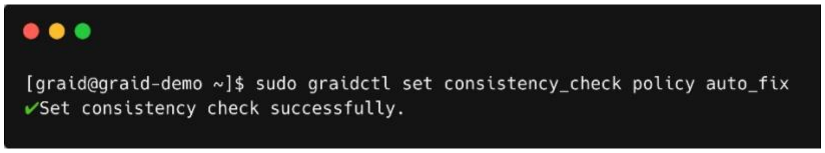
Setting the Consistency Check Policy

To set a consistency check policy, issue the following command.

Note: By default, the consistency check runs on all drive_groups. To exclude drive groups, run the xcluded_dgs command.

```
$ sudo graidctl set consistency_check policy [auto_fix|stop_on_error] [flags]
```

Output example:



```
[graid@graid-demo ~]$ sudo graidctl set consistency_check policy auto_fix
✔Set consistency check successfully.
```

Excluding Drive Groups from the Consistency Check Policy

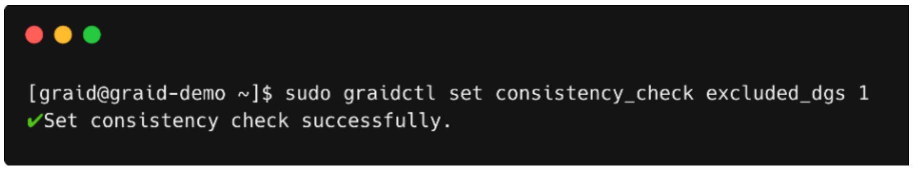
To exclude some drive groups from a consistency check policy, issue the following command:

```
$ sudo graidctl set consistency_check excluded_dgs [DG_IDs]
```

OR

```
$ sudo graidctl set cc excluded_dgs [DG_IDs]
```

Output example:



```
[graid@graid-demo ~]$ sudo graidctl set consistency_check excluded_dgs 1
✔Set consistency check successfully.
```

ADDITIONAL FUNCTIONS

This chapter describes the following additional tasks you can perform with SupremeRAID™.

- Configuring Boot-Drive Devices
- Manually Migrating the RAID Configuration Between Hosts
- Restarting the SupremeRAID™ Service After Upgrading the System Kernel
- Obtaining SMART Information from Devices
- Monitoring System Input/Output Statistics for Devices Using iostat
- Setting Up the Auto-mount File Systems on Linux Using the SupremeRAID™ Driver
- ESXi Virtual Machine Support Using GPU Passthrough
- Using Self-Encrypting Drives

Configuring Boot-Drive Devices

You can configure two NVMe SSDs as RAID1 boot devices and control them using SupremeRAID™. The procedure you use depends on the operating system.

- For CentOS, see Procedure for CentOS.
- For Ubuntu, see Procedure for Ubuntu.
- For SLES 15 SP2 and SP3, see Procedure for SLES 15 SP2, and SP3.

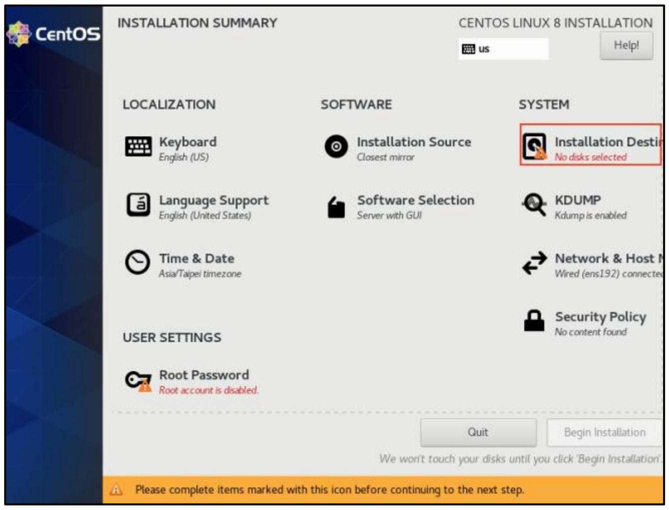
Note: Please note, these procedures are provided for reference only. Your actual steps may vary depending on your Linux distribution and version. For complete and up-to-date information, please refer to your Linux distro's documentation or contact the distro's support team for further information. You cannot configure boot-drive devices across multiple operating systems.

Procedure for CentOS

Assigning RAID1 Boot Devices Manually

You assign RAID1 boot devices when you install CentOS. If the CentOS GUI does not prompt you to assign the boot devices, you can assign them manually.

Step 1 From the INSTALLATION SUMMARY page, select **SYSTEM > Installation Destination**.



INSTALLATION SUMMARY

CENTOS LINUX 8 INSTALLATION

us

Help!

LOCALIZATION

Keyboard
English (US)

Language Support
English (United States)

Time & Date
Asia/Taipei timezone

SOFTWARE

Installation Source
Closest mirror

Software Selection
Server with GUI

SYSTEM

Installation Desti
No disks selected

KDUMP
Kdump is enabled

Network & Host N
Wired (ens192) connecte

Security Policy
No content found

USER SETTINGS

Root Password
Root account is disabled.

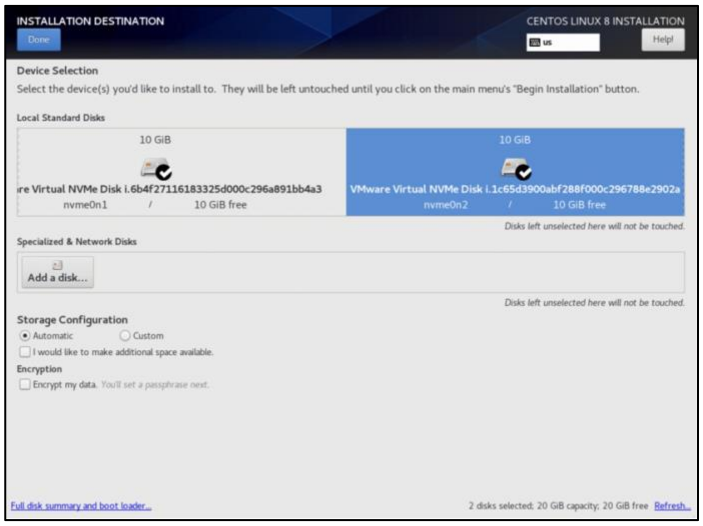
Quit

Begin Installation

We won't touch your disks until you click 'Begin Installation'.

Please complete items marked with this icon before continuing to the next step.

Step 2 From the INSTALLATION DESTINATION page, select the two NVMe SSDs that you want to set as RAID1 boot devices.



INSTALLATION DESTINATION

Done

CENTOS LINUX 8 INSTALLATION

us

Help!

Device Selection

Select the device(s) you'd like to install to. They will be left untouched until you click on the main menu's "Begin Installation" button.

Local Standard Disks

10 GiB
re Virtual NVMe Disk i.6b4f27116183325d000c296a891bb4a3
nvme0n1 / 10 GiB free

10 GiB
VMware Virtual NVMe Disk i.1c65d3900abf288f000c296788e2902a
nvme0n2 / 10 GiB free

Disks left unselected here will not be touched.

Specialized & Network Disks

Add a disk...

Disks left unselected here will not be touched.

Storage Configuration

[✓ Automatic] Custom

☐ I would like to make additional space available.

Encryption

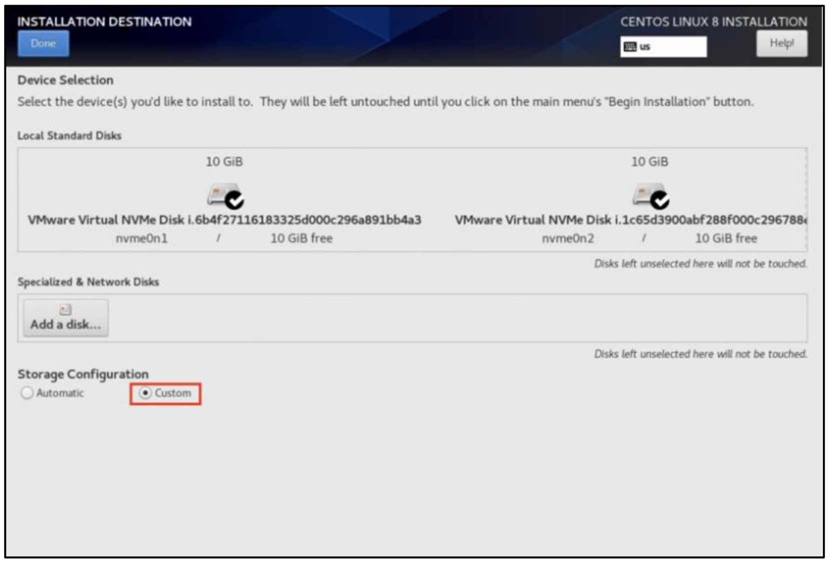
☐ Encrypt my data. You'll set a passphrase next.

Full disk summary and boot loader...

2 disks selected; 20 GiB capacity; 20 GiB free Refresh...

Note: To select multiple devices, use the Ctrl key.

Step 3 For Storage Configuration, select Custom.



Device Selection

Select the device(s) you'd like to install to. They will be left untouched until you click on the main menu's "Begin Installation" button.

Local Standard Disks

10 GiB — VMware Virtual NVMe Disk i.6b4f27116183325d000c296a891bb4a3 — nvme0n1 / 10 GiB free

10 GiB — VMware Virtual NVMe Disk i.1c65d3900abf288f000c296788 — nvme0n2 / 10 GiB free

Disks left unselected here will not be touched.

Specialized & Network Disks

Disks left unselected here will not be touched.

Storage Configuration

Automatic [✓ Custom]

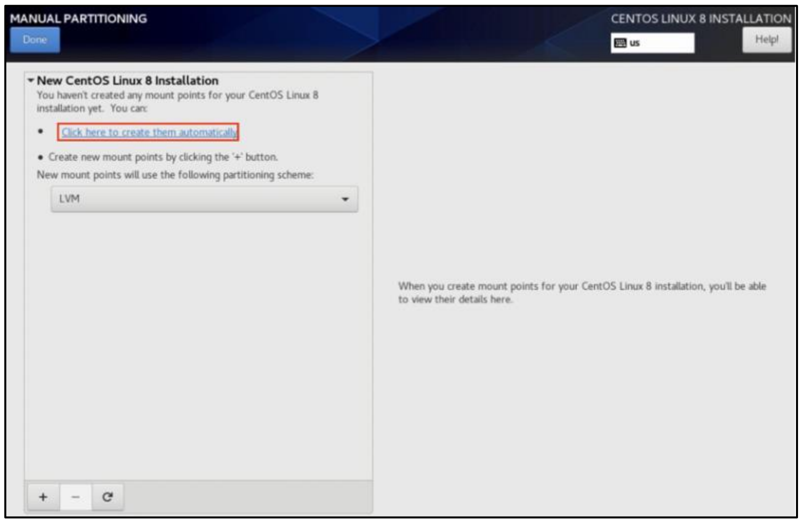
Step 4 Click **Done**.

Creating Storage Partitions Manually

You manually create the storage partitions on CentOS systems. Each partition function as a software RAID.

Step 1 From the MANUAL PARTITIONING page, select New CentOS Linux 8 Installation.

Step 2 Click here to create them automatically to create the mount points.



New CentOS Linux 8 Installation

You haven't created any mount points for your CentOS Linux 8 installation yet. You can:

- Click here to create them automatically
- Create new mount points by clicking the '+' button.

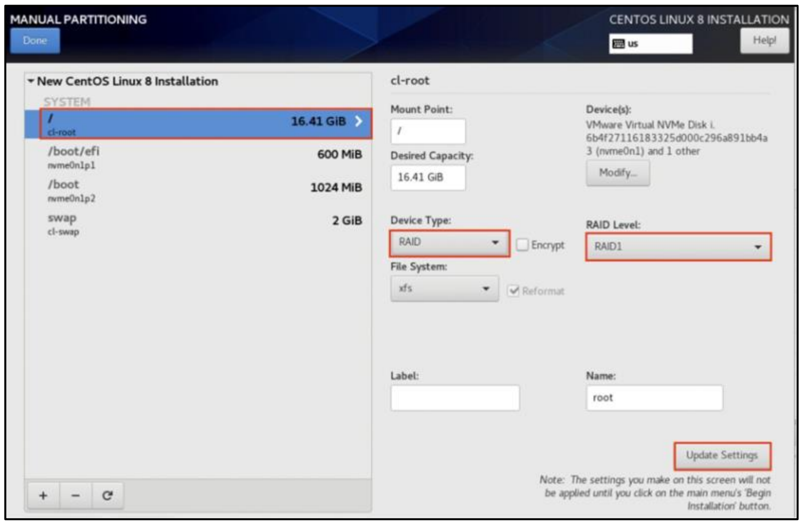
New mount points will use the following partitioning scheme:

LVM

When you create mount points for your CentOS Linux 8 installation, you'll be able to view their details here.

Step 3 Set Device Type to RAID and set RAID LEVEL to RAID 1.

Step 4 Click Update Settings. Each partition function as a software RAID.



cl-root

Mount Point: /

Desired Capacity: 16.41 GiB

Device(s): VMware Virtual NVMe Disk i. 6b4f27116183325d000c296a891bb4a3 (nvme0n1) and 1 other

Device Type: RAID

File System: xfs

RAID Level: RAID1

Label:

Name: root

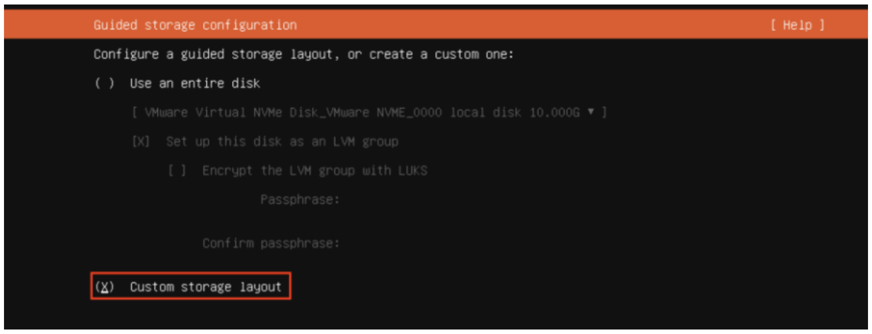
Note: The settings you make on this screen will not be applied until you click on the main menu's 'Begin Installation' button.

Procedure for Ubuntu

Creating and Configuring Storage Partitions

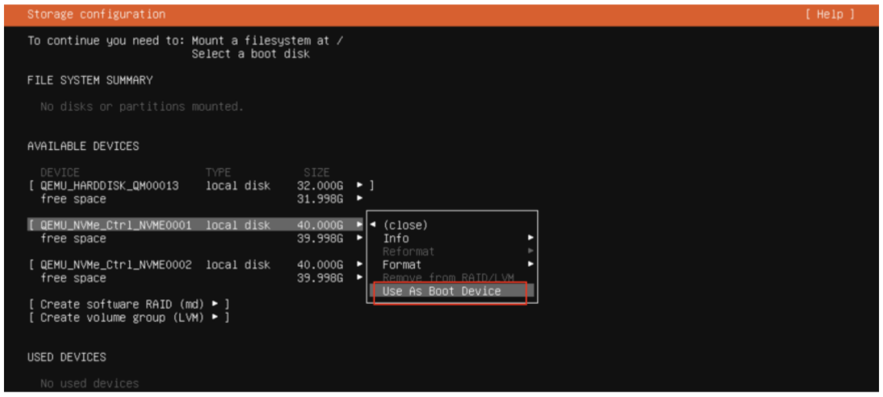
Storage partitions must be created and configured during the Ubuntu Server 20.04 installation. The partitions are required for mounting /boot, swap, and root/. Each partition functions as a software RAID.

Step 1 From the Guided storage configuration page, select **Custom storage layout**.



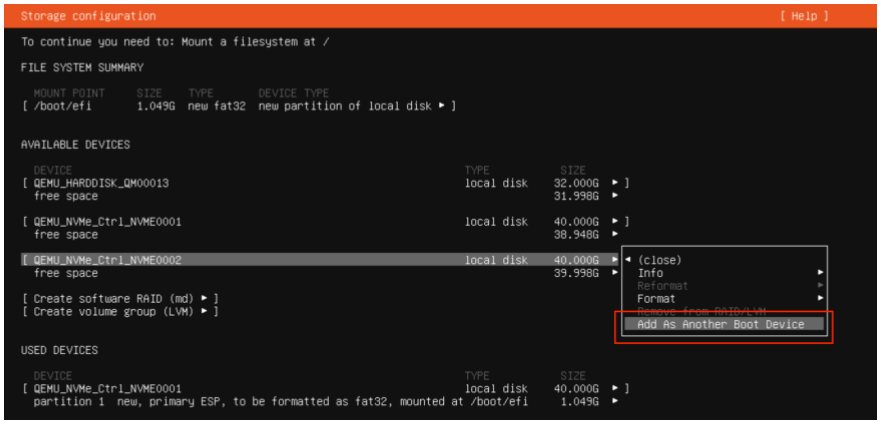
```
Guided storage configuration                                          [ Help ]
Configure a guided storage layout, or create a custom one:
( )  Use an entire disk
     [ VMware Virtual NVMe Disk_VMware NVME_0000 local disk 10.000G ▾ ]
     [X]  Set up this disk as an LVM group
          [ ]  Encrypt the LVM group with LUKS
                    Passphrase:

               Confirm passphrase:

(X)  Custom storage layout
```

Step 2 From the Storage configuration page, select the first disk and choose **Use As Boot Device**.



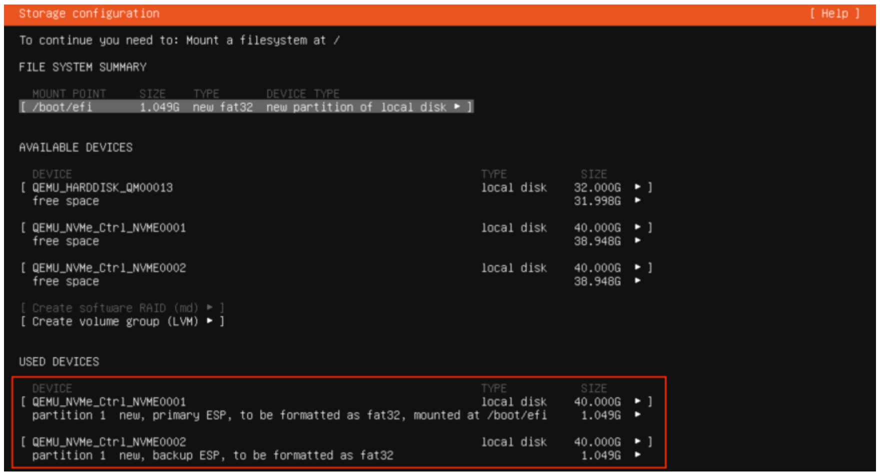
```
Storage configuration                                                  [ Help ]
To continue you need to: Mount a filesystem at /
                         Select a boot disk

FILE SYSTEM SUMMARY

  No disks or partitions mounted.

AVAILABLE DEVICES

  DEVICE                  TYPE         SIZE
[ QEMU_HARDDISK_QM00013   local disk   32.000G  ► ]
  free space                           31.998G  ►

[ QEMU_NVMe_Ctrl_NVME0001  local disk   40.000G  ► ◂ (close)
  free space                           39.998G  ►    Info               ►
                                                     Reformat           ►
[ QEMU_NVMe_Ctrl_NVME0002  local disk   40.000G  ►   Format             ►
  free space                           39.998G  ►    Remove from RAID/LVM
                                                     Use As Boot Device
[ Create software RAID (md) ► ]
[ Create volume group (LVM) ► ]

USED DEVICES

  No used devices
```

Step 3 From the Storage Configuration page, select the second disk and **Use As Another Device**.



```
Storage configuration                                                                     [ Help ]
To continue you need to: Mount a filesystem at /

FILE SYSTEM SUMMARY

  MOUNT POINT     SIZE    TYPE      DEVICE TYPE
[ /boot/efi       1.049G  new fat32  new partition of local disk ▸ ]

AVAILABLE DEVICES

  DEVICE                                                        TYPE         SIZE
[ QEMU_HARDDISK_QM00013                                         local disk   32.000G  ▸ ]
  free space                                                                 31.998G  ▸

[ QEMU_NVMe_Ctrl_NVME0001                                       local disk   40.000G  ▸ ]
  free space                                                                 38.948G  ▸

[ QEMU_NVMe_Ctrl_NVME0002                                       local disk   40.000G  ▸ ◂ (close)
  free space                                                                 39.998G  ▸   Info                   ▸
                                                                                          Reformat               ▸
[ Create software RAID (md) ▸ ]                                                           Format                 ▸
[ Create volume group (LVM) ▸ ]                                                           Remove from RAID/LVM
                                                                                          Add As Another Boot Device

USED DEVICES

  DEVICE                                                        TYPE         SIZE
[ QEMU_NVMe_Ctrl_NVME0001                                       local disk   40.000G  ▸ ]
  partition 1  new, primary ESP, to be formatted as fat32, mounted at /boot/efi  1.049G  ▸
```

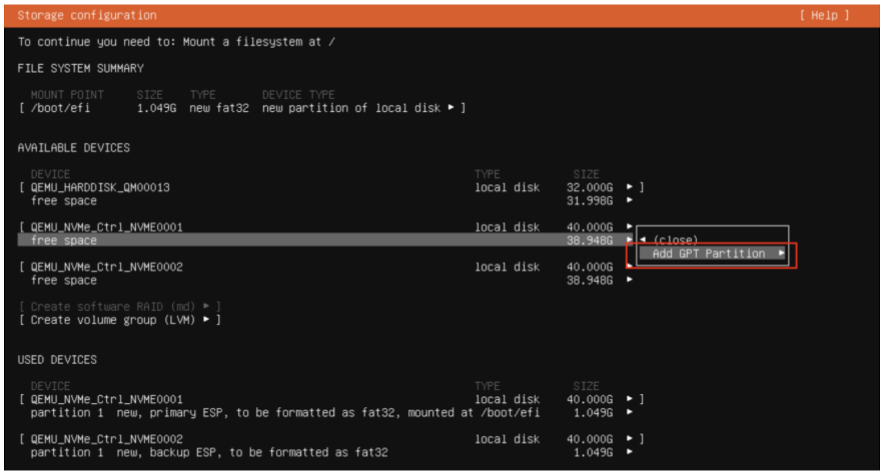
Step 4 Devices used for the MD bootable RAID will be listed as **USED DEVICES** in the interface.



```
Storage configuration                                                                     [ Help ]
To continue you need to: Mount a filesystem at /

FILE SYSTEM SUMMARY

  MOUNT POINT     SIZE    TYPE      DEVICE TYPE
[ /boot/efi       1.049G  new fat32  new partition of local disk ▸ ]

AVAILABLE DEVICES

  DEVICE                                                        TYPE         SIZE
[ QEMU_HARDDISK_QM00013                                         local disk   32.000G  ▸ ]
  free space                                                                 31.998G  ▸

[ QEMU_NVMe_Ctrl_NVME0001                                       local disk   40.000G  ▸ ]
  free space                                                                 38.948G  ▸

[ QEMU_NVMe_Ctrl_NVME0002                                       local disk   40.000G  ▸ ]
  free space                                                                 38.948G  ▸

[ Create software RAID (md) ▸ ]
[ Create volume group (LVM) ▸ ]

USED DEVICES

  DEVICE                                                        TYPE         SIZE
[ QEMU_NVMe_Ctrl_NVME0001                                       local disk   40.000G  ▸ ]
  partition 1  new, primary ESP, to be formatted as fat32, mounted at /boot/efi  1.049G  ▸

[ QEMU_NVMe_Ctrl_NVME0002                                       local disk   40.000G  ▸ ]
  partition 1  new, backup ESP, to be formatted as fat32                      1.049G  ▸
```

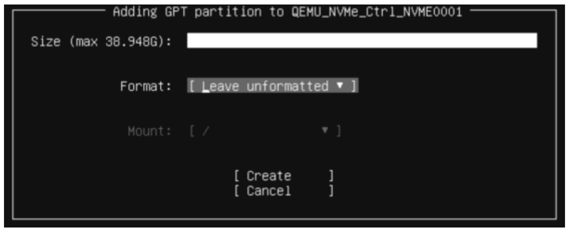
Step 5 From the Disk menu, select **free space** and choose **Add GPT Partition**. Leave both disks unformatted.

A Select first drive and select **Add GPT Partition**.



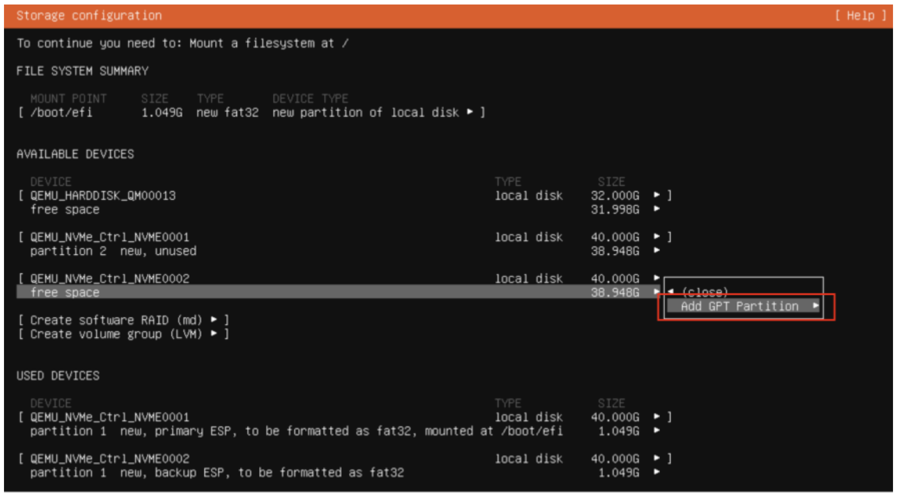
```
Storage configuration                                                          [ Help ]
To continue you need to: Mount a filesystem at /

FILE SYSTEM SUMMARY

  MOUNT POINT    SIZE    TYPE       DEVICE TYPE
[ /boot/efi      1.049G  new fat32  new partition of local disk ▸ ]

AVAILABLE DEVICES

  DEVICE                                              TYPE        SIZE
[ QEMU_HARDDISK_QM00013                               local disk  32.000G  ▸ ]
  free space                                                      31.998G  ▸

[ QEMU_NVMe_Ctrl_NVME0001                             local disk  40.000G  ▸
  free space                                                      38.948G  ▸ ◂ (close)
                                                                             Add GPT Partition ▸
[ QEMU_NVMe_Ctrl_NVME0002                             local disk  40.000G  ▸
  free space                                                      38.948G  ▸

[ Create software RAID (md) ▸ ]
[ Create volume group (LVM) ▸ ]

USED DEVICES

  DEVICE                                              TYPE        SIZE
[ QEMU_NVMe_Ctrl_NVME0001                             local disk  40.000G  ▸ ]
  partition 1  new, primary ESP, to be formatted as fat32, mounted at /boot/efi  1.049G  ▸

[ QEMU_NVMe_Ctrl_NVME0002                             local disk  40.000G  ▸ ]
  partition 1  new, backup ESP, to be formatted as fat32           1.049G  ▸
```

B Leave the drive unformatted.



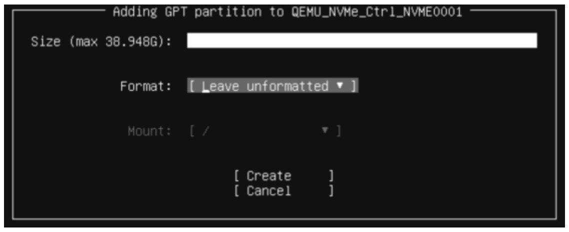
```
 Adding GPT partition to QEMU_NVMe_Ctrl_NVME0001

  Size (max 38.948G):

            Format:  [ Leave unformatted ▾ ]

             Mount:  [ /              ▾ ]

                          [ Create     ]
                          [ Cancel     ]
```

C Select another drive for OS bootable RAID.



```
Storage configuration                                                          [ Help ]
To continue you need to: Mount a filesystem at /

FILE SYSTEM SUMMARY

  MOUNT POINT    SIZE    TYPE       DEVICE TYPE
[ /boot/efi      1.049G  new fat32  new partition of local disk ▸ ]

AVAILABLE DEVICES

  DEVICE                                              TYPE        SIZE
[ QEMU_HARDDISK_QM00013                               local disk  32.000G  ▸ ]
  free space                                                      31.998G  ▸

[ QEMU_NVMe_Ctrl_NVME0001                             local disk  40.000G  ▸ ]
  partition 2  new, unused                                        38.948G  ▸

[ QEMU_NVMe_Ctrl_NVME0002                             local disk  40.000G  ▸
  free space                                                      38.948G  ▸ ◂ (close)
                                                                             Add GPT Partition ▸
[ Create software RAID (md) ▸ ]
[ Create volume group (LVM) ▸ ]

USED DEVICES

  DEVICE                                              TYPE        SIZE
[ QEMU_NVMe_Ctrl_NVME0001                             local disk  40.000G  ▸ ]
  partition 1  new, primary ESP, to be formatted as fat32, mounted at /boot/efi  1.049G  ▸

[ QEMU_NVMe_Ctrl_NVME0002                             local disk  40.000G  ▸ ]
  partition 1  new, backup ESP, to be formatted as fat32           1.049G  ▸
```

D Leave the drive also unformatted.



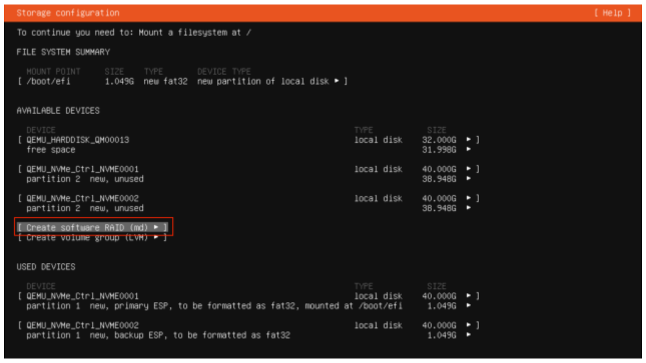
```
Adding GPT partition to QEMU_NVMe_Ctrl_NVME0001

Size (max 38.948G):

         Format:  [ Leave unformatted ▼ ]

          Mount:  [ /                ▼ ]

                     [ Create      ]
                     [ Cancel      ]
```

Note: You must use **[Leave unformatted]**. DO NOT mount the partition. Setting RAID1 and mounting partitions on multiple drives (MD) occurs later in this procedure.

Creating a Software RAID for Multiple Devices (MD)

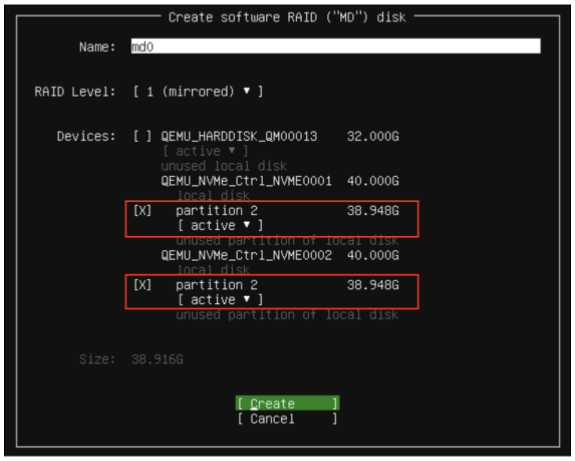
To create the software RAID on multiple devices, from the Storage configuration page, select **Create software RAID (md)**.

Step 1 Select **Create Software RAID (md)** for the previously configured disks.

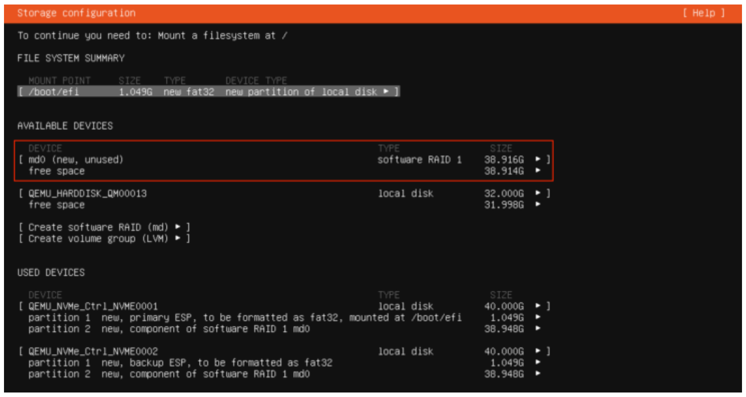


```
Storage configuration                                                    [ Help ]
To continue you need to: Mount a filesystem at /

FILE SYSTEM SUMMARY

  MOUNT POINT     SIZE    TYPE       DEVICE TYPE
[ /boot/efi       1.049G  new fat32  new partition of local disk ► ]


AVAILABLE DEVICES

  DEVICE                                                TYPE        SIZE
[ QEMU_HARDDISK_QM00013                                 local disk   32.000G  ► ]
  free space                                                         31.998G  ►

[ QEMU_NVMe_Ctrl_NVME0001                               local disk   40.000G  ► ]
  partition 2  new, unused                                           38.948G  ►

[ QEMU_NVMe_Ctrl_NVME0002                               local disk   40.000G  ► ]
  partition 2  new, unused                                           38.948G  ►

[ Create software RAID (md) ► ]
[ Create volume group (LVM) ► ]


USED DEVICES

  DEVICE                                                TYPE        SIZE
[ QEMU_NVMe_Ctrl_NVME0001                               local disk   40.000G  ► ]
  partition 1  new, primary ESP, to be formatted as fat32, mounted at /boot/efi   1.049G  ►

[ QEMU_NVMe_Ctrl_NVME0002                               local disk   40.000G  ► ]
  partition 1  new, backup ESP, to be formatted as fat32              1.049G  ►
```

Step 2 Select the configured partitions on both disks, then create the Software RAID (md).



```
Create software RAID ("MD") disk

       Name:  md0

 RAID Level:  [ 1 (mirrored) ▼ ]

    Devices:  [ ] QEMU_HARDDISK_QM00013    32.000G
                  [ active ▼ ]
                  unused local disk
                  QEMU_NVMe_Ctrl_NVME0001  40.000G
                    local disk
              [X]   partition 2            38.948G
                    [ active ▼ ]
                    unused partition of local disk
                  QEMU_NVMe_Ctrl_NVME0002  40.000G
                    local disk
              [X]   partition 2            38.948G
                    [ active ▼ ]
                    unused partition of local disk


       Size:  38.916G


                  [ Create      ]
                  [ Cancel      ]
```



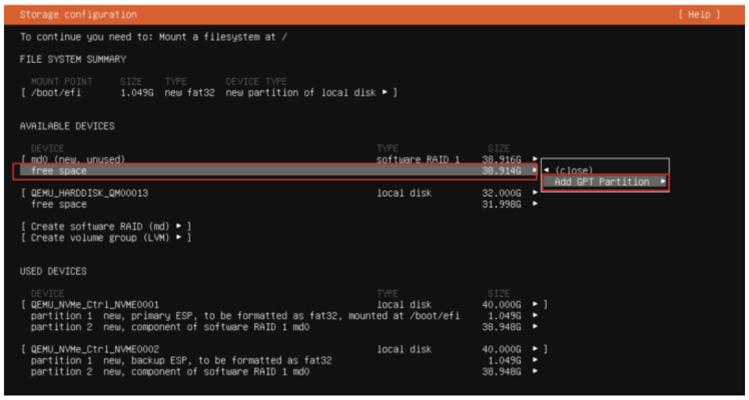
```
Storage configuration                                                            [ Help ]
To continue you need to: Mount a filesystem at /

FILE SYSTEM SUMMARY

  MOUNT POINT     SIZE    TYPE       DEVICE TYPE
[ /boot/efi       1.049G  new fat32  new partition of local disk ▸ ]

AVAILABLE DEVICES

  DEVICE                                                TYPE               SIZE
[ md0 (new, unused)                                     software RAID 1    38.916G  ▸ ]
  free space                                                               38.914G  ▸

[ QEMU_HARDDISK_QM00013                                 local disk         32.000G  ▸ ]
  free space                                                               31.998G  ▸

[ Create software RAID (md) ▸ ]
[ Create volume group (LVM) ▸ ]

USED DEVICES

  DEVICE                                                TYPE               SIZE
[ QEMU_NVMe_Ctrl_NVME0001                               local disk         40.000G  ▸ ]
  partition 1  new, primary ESP, to be formatted as fat32, mounted at /boot/efi   1.049G  ▸
  partition 2  new, component of software RAID 1 md0                       38.948G  ▸

[ QEMU_NVMe_Ctrl_NVME0002                               local disk         40.000G  ▸ ]
  partition 1  new, backup ESP, to be formatted as fat32                    1.049G  ▸
  partition 2  new, component of software RAID 1 md0                       38.948G  ▸
```

Configuring the Boot Partition for MD

The following procedure describes how to configure the /boot, swap, and root/ partitions on both disks

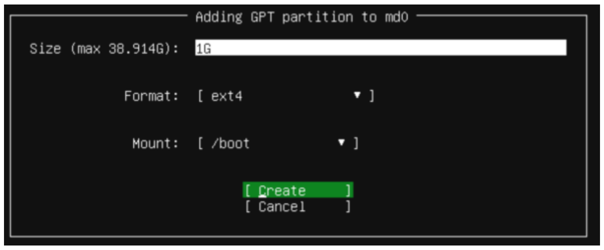
To set MD as the mounting point:

Step 3 Select the **free space** option in the md, then Choose **Add GPT Partition**.

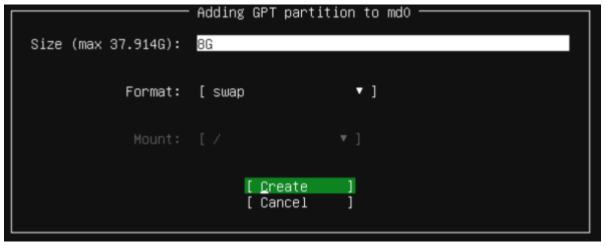


```
Storage configuration                                                            [ Help ]
To continue you need to: Mount a filesystem at /

FILE SYSTEM SUMMARY

  MOUNT POINT     SIZE    TYPE       DEVICE TYPE
[ /boot/efi       1.049G  new fat32  new partition of local disk ▸ ]

AVAILABLE DEVICES

  DEVICE                                                TYPE               SIZE
[ md0 (new, unused)                                     software RAID 1    38.916G  ▸
  free space                                                               38.914G  ▸  ◂ (close)
                                                                                        Add GPT Partition ▸
[ QEMU_HARDDISK_QM00013                                 local disk         32.000G  ▸
  free space                                                               31.998G  ▸

[ Create software RAID (md) ▸ ]
[ Create volume group (LVM) ▸ ]

USED DEVICES

  DEVICE                                                TYPE               SIZE
[ QEMU_NVMe_Ctrl_NVME0001                               local disk         40.000G  ▸ ]
  partition 1  new, primary ESP, to be formatted as fat32, mounted at /boot/efi   1.049G  ▸
  partition 2  new, component of software RAID 1 md0                       38.948G  ▸

[ QEMU_NVMe_Ctrl_NVME0002                               local disk         40.000G  ▸ ]
  partition 1  new, backup ESP, to be formatted as fat32                    1.049G  ▸
  partition 2  new, component of software RAID 1 md0                       38.948G  ▸
```

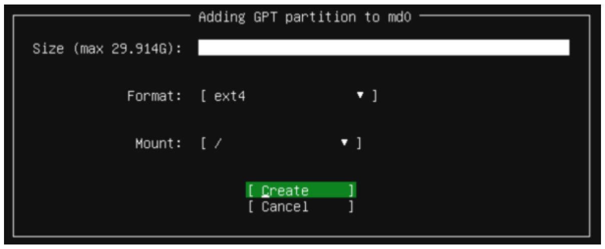
Step 4 Set the size of the EFI System Partition (ESP). Allocate sufficient capacity for each partition based on anticipated usage.



```
──────────────── Adding GPT partition to md0 ────────────────

  Size (max 38.914G): 1G

             Format:  [ ext4             ▾ ]

              Mount:  [ /boot           ▾ ]

                          [ Create      ]
                          [ Cancel      ]
```

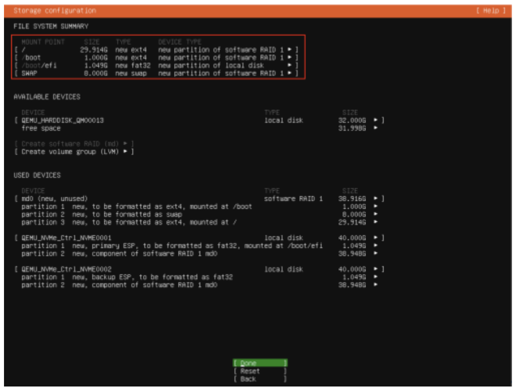


```
Adding GPT partition to md0

Size (max 37.914G): 8G

Format:  [ swap           ▼ ]

Mount:   [ /              ▼ ]

         [ Create    ]
         [ Cancel    ]
```



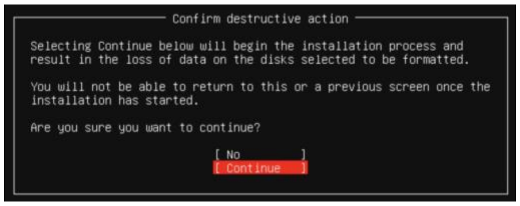
```
Adding GPT partition to md0

Size (max 29.914G):

Format:  [ ext4           ▼ ]

Mount:   [ /              ▼ ]

         [ Create    ]
         [ Cancel    ]
```

Step 5 After creating the partitions, the md configuration should display the following information.



```
Storage configuration                                                              [ Help ]
FILE SYSTEM SUMMARY

  MOUNT POINT     SIZE    TYPE      DEVICE TYPE
[ /             29.914G  new ext4   new partition of software RAID 1 ▸ ]
[ /boot          1.000G  new ext4   new partition of software RAID 1 ▸ ]
[ /boot/efi      1.049G  new fat32  new partition of local disk      ▸ ]
[ SWAP           8.000G  new swap   new partition of software RAID 1 ▸ ]

AVAILABLE DEVICES

  DEVICE                                                   TYPE            SIZE
[ QEMU_HARDDISK_QM00013                                    local disk      32.000G  ▸ ]
  free space                                                               31.998G  ▸

[ Create software RAID (md) ▸ ]
[ Create volume group (LVM) ▸ ]

USED DEVICES

  DEVICE                                                   TYPE            SIZE
[ md0 (new, unused)                                        software RAID 1  38.916G  ▸ ]
  partition 1  new, to be formatted as ext4, mounted at /boot               1.000G  ▸
  partition 2  new, to be formatted as swap                                 8.000G  ▸
  partition 3  new, to be formatted as ext4, mounted at /                  29.914G  ▸

[ QEMU_NVMe_Ctrl_NVME0001                                  local disk      40.000G  ▸ ]
  partition 1  new, primary ESP, to be formatted as fat32, mounted at /boot/efi  1.049G  ▸
  partition 2  new, component of software RAID 1 md0                       38.948G  ▸

[ QEMU_NVMe_Ctrl_NVME0002                                  local disk      40.000G  ▸ ]
  partition 1  new, backup ESP, to be formatted as fat32                    1.049G  ▸
  partition 2  new, component of software RAID 1 md0                       38.948G  ▸

                              [ Done      ]
                              [ Reset     ]
                              [ Back      ]
```

Step 6 From the Confirm destructive action popup, select **Continue**. The partition settings are now in effect.

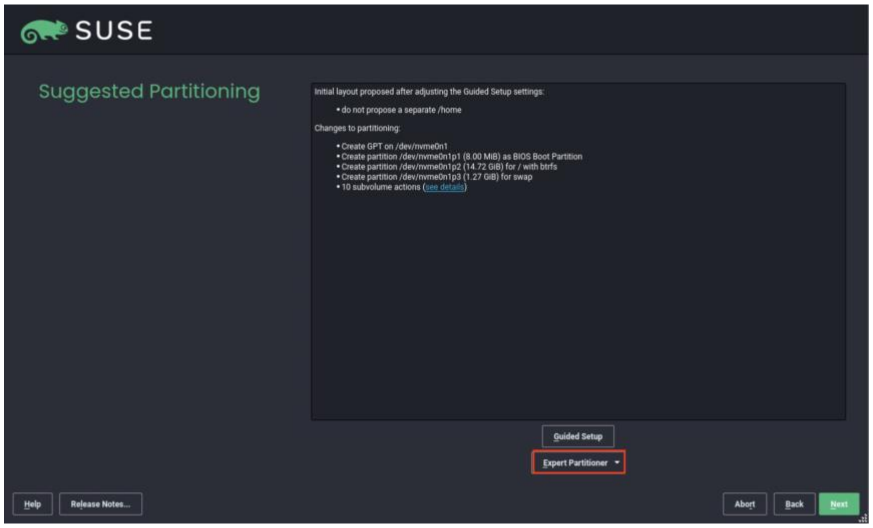


```
Confirm destructive action

Selecting Continue below will begin the installation process and
result in the loss of data on the disks selected to be formatted.

You will not be able to return to this or a previous screen once the
installation has started.

Are you sure you want to continue?

                  [ No           ]
                  [ Continue     ]
```

Procedure for SLES 15 SP2, and SP3

When installing SLES 15 SP2 or SP3, you must manually create RAID1 and configure the partitions. To manually create RAID1 and configure the partitions:

Step 1 From the SUSE Suggested Partitioning page, select **Expert Partitioner** > **Next**.



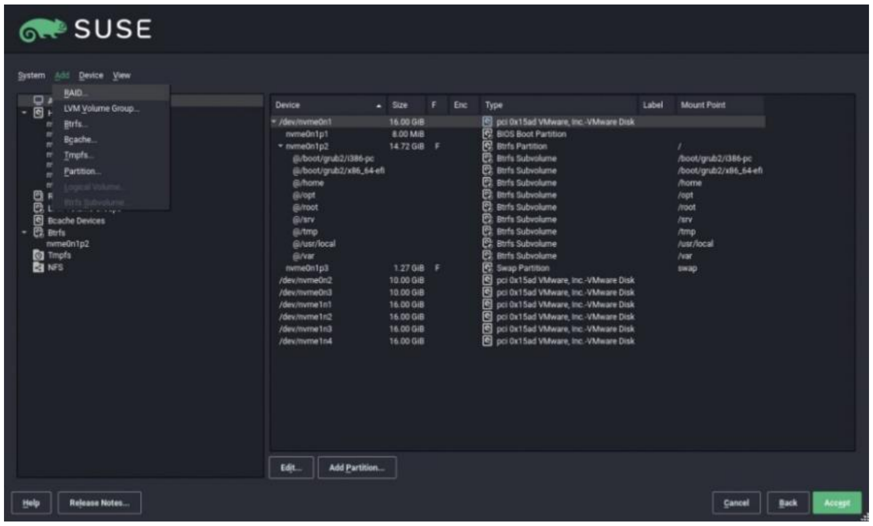
Initial layout proposed after adjusting the Guided Setup settings:

- do not propose a separate /home

Changes to partitioning:

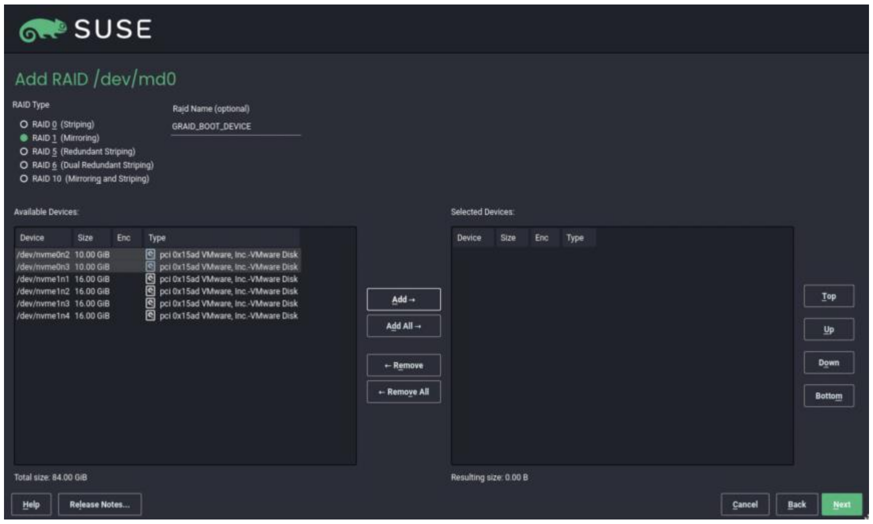
- Create GPT on /dev/nvme0n1
- Create partition /dev/nvme0n1p1 (8.00 MiB) as BIOS Boot Partition
- Create partition /dev/nvme0n1p2 (14.72 GiB) for / with btrfs
- Create partition /dev/nvme0n1p3 (1.27 GiB) for swap
- 10 subvolume actions (see details)

Step 2 From the SUSE Add menu, select **Add** > **RAID**.



Device	Size	F	Enc	Type	Label	Mount Point
/dev/nvme0n1	16.00 GiB			pci 0x15ad VMware, Inc.-VMware Disk		
nvme0n1p1	8.00 MiB			BIOS Boot Partition		
nvme0n1p2	14.72 GiB	F		Btrfs Partition		/
@/boot/grub2/i386-pc				Btrfs Subvolume		/boot/grub2/i386-pc
@/boot/grub2/x86_64-efi				Btrfs Subvolume		/boot/grub2/x86_64-efi
@/home				Btrfs Subvolume		/home
@/opt				Btrfs Subvolume		/opt
@/root				Btrfs Subvolume		/root
@/srv				Btrfs Subvolume		/srv
@/tmp				Btrfs Subvolume		/tmp
@/usr/local				Btrfs Subvolume		/usr/local
@/var				Btrfs Subvolume		/var
nvme0n1p3	1.27 GiB	F		Swap Partition		swap
/dev/nvme0n2	10.00 GiB			pci 0x15ad VMware, Inc.-VMware Disk		
/dev/nvme0n3	10.00 GiB			pci 0x15ad VMware, Inc.-VMware Disk		
/dev/nvme1n1	16.00 GiB			pci 0x15ad VMware, Inc.-VMware Disk		
/dev/nvme1n2	16.00 GiB			pci 0x15ad VMware, Inc.-VMware Disk		
/dev/nvme1n3	16.00 GiB			pci 0x15ad VMware, Inc.-VMware Disk		
/dev/nvme1n4	16.00 GiB			pci 0x15ad VMware, Inc.-VMware Disk		

Step 3 From the SUSE Add RAID page, select **RAID 1 (Mirroring)** for the **RAID Type**.



Add RAID /dev/md0

RAID Type

- ○ RAID 0 (Striping)
- ● RAID 1 (Mirroring)
- ○ RAID 5 (Redundant Striping)
- ○ RAID 6 (Dual Redundant Striping)
- ○ RAID 10 (Mirroring and Striping)

Raid Name (optional): GRAID_BOOT_DEVICE

Available Devices:

Device	Size	Enc	Type
/dev/nvme0n2	10.00 GiB		pci 0x15ad VMware, Inc.-VMware Disk
/dev/nvme0n3	10.00 GiB		pci 0x15ad VMware, Inc.-VMware Disk
/dev/nvme1n1	16.00 GiB		pci 0x15ad VMware, Inc.-VMware Disk
/dev/nvme1n2	16.00 GiB		pci 0x15ad VMware, Inc.-VMware Disk
/dev/nvme1n3	16.00 GiB		pci 0x15ad VMware, Inc.-VMware Disk
/dev/nvme1n4	16.00 GiB		pci 0x15ad VMware, Inc.-VMware Disk

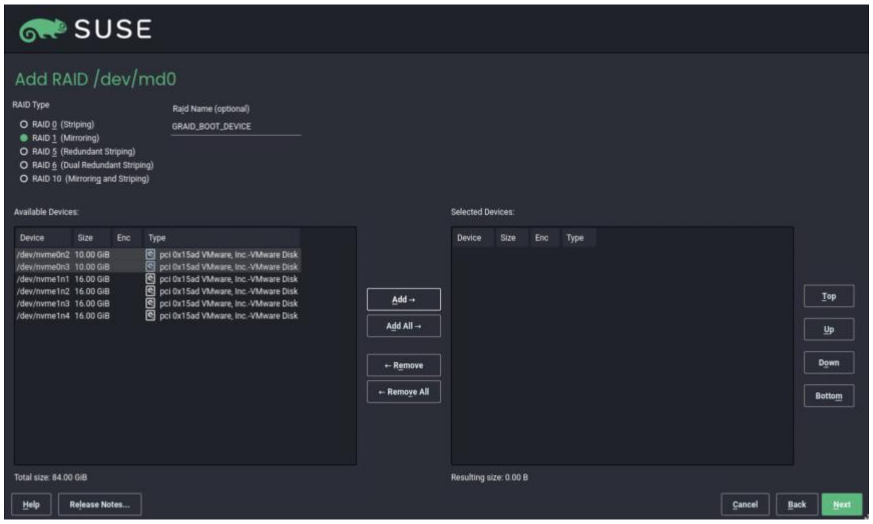
Total size: 84.00 GiB

Selected Devices:

Device	Size	Enc	Type

Resulting size: 0.00 B

Step 4 From the **Selected Devices** list, select two NVMe disks and click **Add**.



Add RAID /dev/md0

RAID Type

- ○ RAID 0 (Striping)
- ● RAID 1 (Mirroring)
- ○ RAID 5 (Redundant Striping)
- ○ RAID 6 (Dual Redundant Striping)
- ○ RAID 10 (Mirroring and Striping)

Raid Name (optional): GRAID_BOOT_DEVICE

Available Devices:

Device	Size	Enc	Type
/dev/nvme0n2	10.00 GiB		pci 0x15ad VMware, Inc.-VMware Disk
/dev/nvme0n3	10.00 GiB		pci 0x15ad VMware, Inc.-VMware Disk
/dev/nvme1n1	16.00 GiB		pci 0x15ad VMware, Inc.-VMware Disk
/dev/nvme1n2	16.00 GiB		pci 0x15ad VMware, Inc.-VMware Disk
/dev/nvme1n3	16.00 GiB		pci 0x15ad VMware, Inc.-VMware Disk
/dev/nvme1n4	16.00 GiB		pci 0x15ad VMware, Inc.-VMware Disk

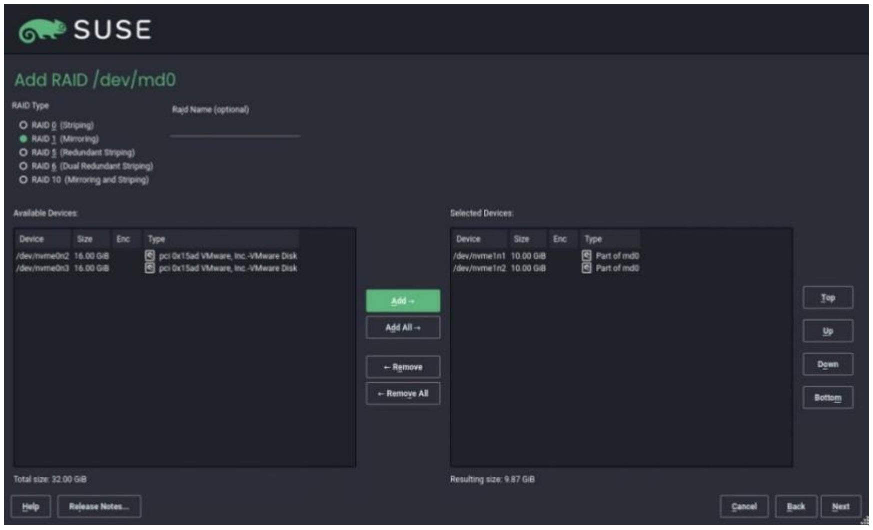
Total size: 84.00 GiB

Selected Devices:

Device	Size	Enc	Type

Resulting size: 0.00 B

Step 5 Click **Next** to continue with the installation.



Add RAID /dev/md0

RAID Type

- RAID 0 (Striping)
- [✓ RAID 1 (Mirroring)]
- RAID 5 (Redundant Striping)
- RAID 6 (Dual Redundant Striping)
- RAID 10 (Mirroring and Striping)

Raid Name (optional)

Available Devices:

Device	Size	Enc	Type
/dev/nvme0n2	16.00 GiB		pci 0x15ad VMware, Inc.-VMware Disk
/dev/nvme0n3	16.00 GiB		pci 0x15ad VMware, Inc.-VMware Disk

Total size: 32.00 GiB

Selected Devices:

Device	Size	Enc	Type
/dev/nvme1n1	10.00 GiB		Part of md0
/dev/nvme1n2	10.00 GiB		Part of md0

Resulting size: 9.87 GiB

Manually Migrating the RAID Configuration Between Hosts

The following procedure describes how to migrate the RAID configuration manually between hosts.

Restoring a RAID Configuration from a Backup Configuration File

To restore a RAID configuration from a backup configuration file:

Step 1 Periodically back up the configuration file /etc/graid.conf from the original host. Use cp or scp to move the configuration file to another system.

Step 2 Set up the target host and ensure that the SupremeRAID™ service is stopped.

Note: If the target host already contains an installed and running SupremeRAID™ card, stop the service and copy the graid.conf file from the original system. On the original system, stop any running applications or unmount the mountpoint before starting the SupremeRAID™ service.

Step 3 Move all the SSDs from the original host to the new host.

Step 4 Install the SupremeRAID™ driver on the new server. Stop the SupremeRAID™ service before copying the configuration backup file to the new host using the same path (/etc/graid.conf). If you have already enabled the graphical management console, please ensure to disable it as well.

```
$ sudo systemctl stop graid
$ sudo systemctl stop graid-mgr.service
```

Step 5 Copy the configuration file.

```
$ sudo cp graid.conf /etc/graid.conf
```

Step 6 If the original card also moved to the new host, start the SupremeRAID™ service directly.

```
$ sudo systemctl start graid
```

Step 7 (Optional) If the card changed, you must apply the new license.

```
$ sudo graidctl apply license [LICENSE_KEY]
```

Restoring a RAID Configuration from SSD Metadata

The SupremeRAID™ system provides robust support for restoring RAID configurations from SSD metadata. This feature allows you to recover a RAID configuration quickly and easily in case of a failure or other issues. Perform the following procedure to restore the RAID configuration and get the SupremeRAID™ system back online.

To restore a RAID configuration from an SSD's metadata:

Step 1 Set up the target host and make sure that the SupremeRAID™ service is stopped.

Note: If the target host already contains an installed and running SupremeRAID™ card, stop the service the SupremeRAID™ service before restoring the configuration. On the original system, stop any running applications or unmount the mountpoint before starting the SupremeRAID™ service.

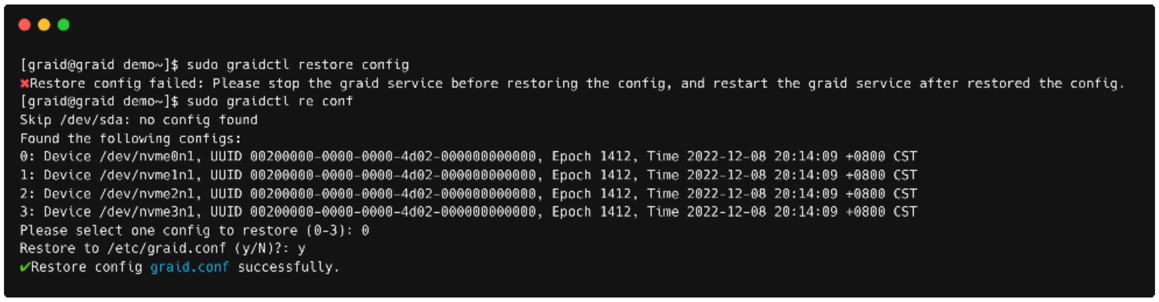
Step 2 Move all the SSDs from the original host to the new host.

Step 3 Install the SupremeRAID™ driver on the new server and stop the SupremeRAID™ service before restoring the configuration file. If you have already enabled the graphical management console, please ensure to disable it as well.

```
$ sudo systemctl stop graid
$ sudo systemctl stop graid-mgr.service
```

Step 4 Run the restore command and restore the configuration file from SSD's metadata.

```
$ sudo graidctl restore config
```



```
[graid@graid demo~]$ sudo graidctl restore config
✘Restore config failed: Please stop the graid service before restoring the config, and restart the graid service after restored the config.
[graid@graid demo~]$ sudo graidctl re conf
Skip /dev/sda: no config found
Found the following configs:
0: Device /dev/nvme0n1, UUID 00200000-0000-0000-4d02-000000000000, Epoch 1412, Time 2022-12-08 20:14:09 +0800 CST
1: Device /dev/nvme1n1, UUID 00200000-0000-0000-4d02-000000000000, Epoch 1412, Time 2022-12-08 20:14:09 +0800 CST
2: Device /dev/nvme2n1, UUID 00200000-0000-0000-4d02-000000000000, Epoch 1412, Time 2022-12-08 20:14:09 +0800 CST
3: Device /dev/nvme3n1, UUID 00200000-0000-0000-4d02-000000000000, Epoch 1412, Time 2022-12-08 20:14:09 +0800 CST
Please select one config to restore (0-3): 0
Restore to /etc/graid.conf (y/N)?: y
✔Restore config graid.conf successfully.
```

Step 5 If the original card also moved to the new host, start the SupremeRAID™ service directly.

```
$ sudo systemctl start graid
```

Step 6 (Optional) If the card changed, you must apply the new license.

```
$ sudo graidctl apply license [LICENSE_KEY]
```

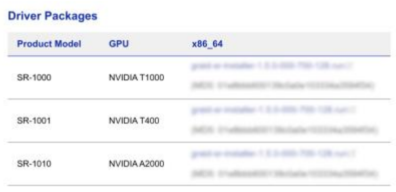
Restarting the SupremeRAID™ Service After Upgrading the System Kernel

If the SupremeRAID™ service does not start properly after upgrading the kernel, reinstall the SupremeRAID™ pre-installer and the installer to ensure that they are configured properly for the new kernel environment.

To reinstall the SupremeRAID™ pre-installer and installer on new kernel, follow these steps:

Step 1 Go to the Graid Technology website to download the latest version of the pre-installer and make it executable, please download the package in Drivers & Documentation.

```
$ sudo chmod +x [Filename]
```



Driver Packages

Product Model	GPU	x86_64
SR-1000	NVIDIA T1000	[illegible]
SR-1001	NVIDIA T400	[illegible]
SR-1010	NVIDIA A2000	[illegible]

Step 2 Open a terminal window and log in to the system as a user with root privileges.

Step 3 Use the cd command to navigate to the directory where the downloaded installer files are located.

Step 4 Run the graid-sr-pre-installer and follow the on-screen instructions to complete the pre-installation process.

Step 5 Run the graid-sr-installer and follow the on-screen instructions to complete the installation process.

Step 6 After installing the SupremeRAID™ pre-installer and installer, restart the SupremeRAID™ service and verify it is running correctly in the new kernel environment.

```
sudo systemctl restart graid
```

Obtaining SMART Information from Devices

Self-Monitoring, Analysis and Reporting Technology (SMART) data is a set of metrics and parameters that SSDs collect and monitor to assess their health and performance. Although the specific information included in the SMART data varies by manufacturer and drive model, it typically reports on the temperature, available spare capacity, power-on hours, error rates, and other details that are used to monitor the health of the SSD and predict its future performance.

By monitoring the SMART data for an SSD, you can identify a potential issue or degradation of the drive before it becomes a serious problem.

To check the SMART information for the gpd device using the NVMe smart-log or smartctl command, follow these steps:

Step 1 Open a terminal window and log in to the system with administrative privileges.

Step 2 Use the list physical drives command to identify the device name for the gpd device, such as /dev/gpdx.

```
$ sudo graidctl list physical_drive
```

Step 3 Use the **nvme** command to display the SMART data for the gpd device:

```
$ sudo nvme smart-log /dev/gpd[#]
```

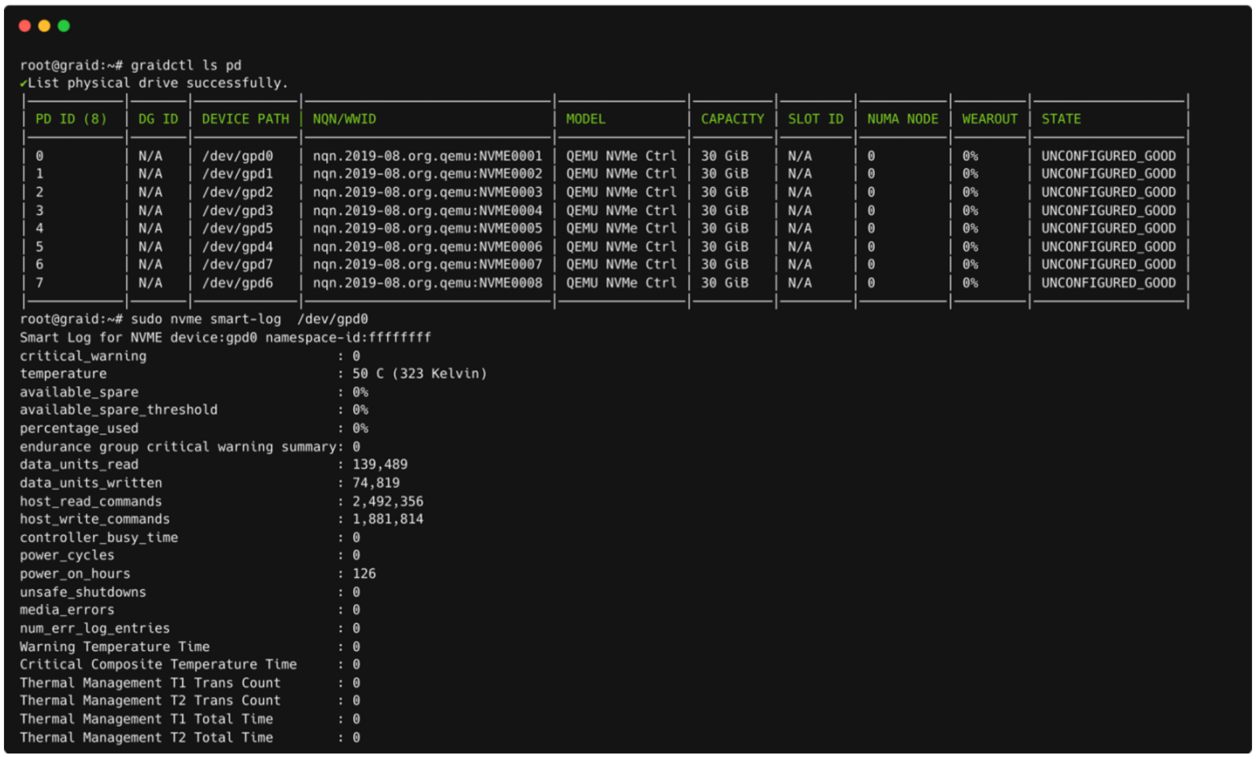
- Alternatively, you can use the smartctl command to display the SMART data for the gpd device:

```
$ sudo smartctl -d nvme -a /dev/gpd[#]
```

A detailed report of the SMART data for the gpd device, including the temperature, available spare capacity, and other details, appears. Use this information to monitor the health and performance of the device and to diagnose any potential issues.

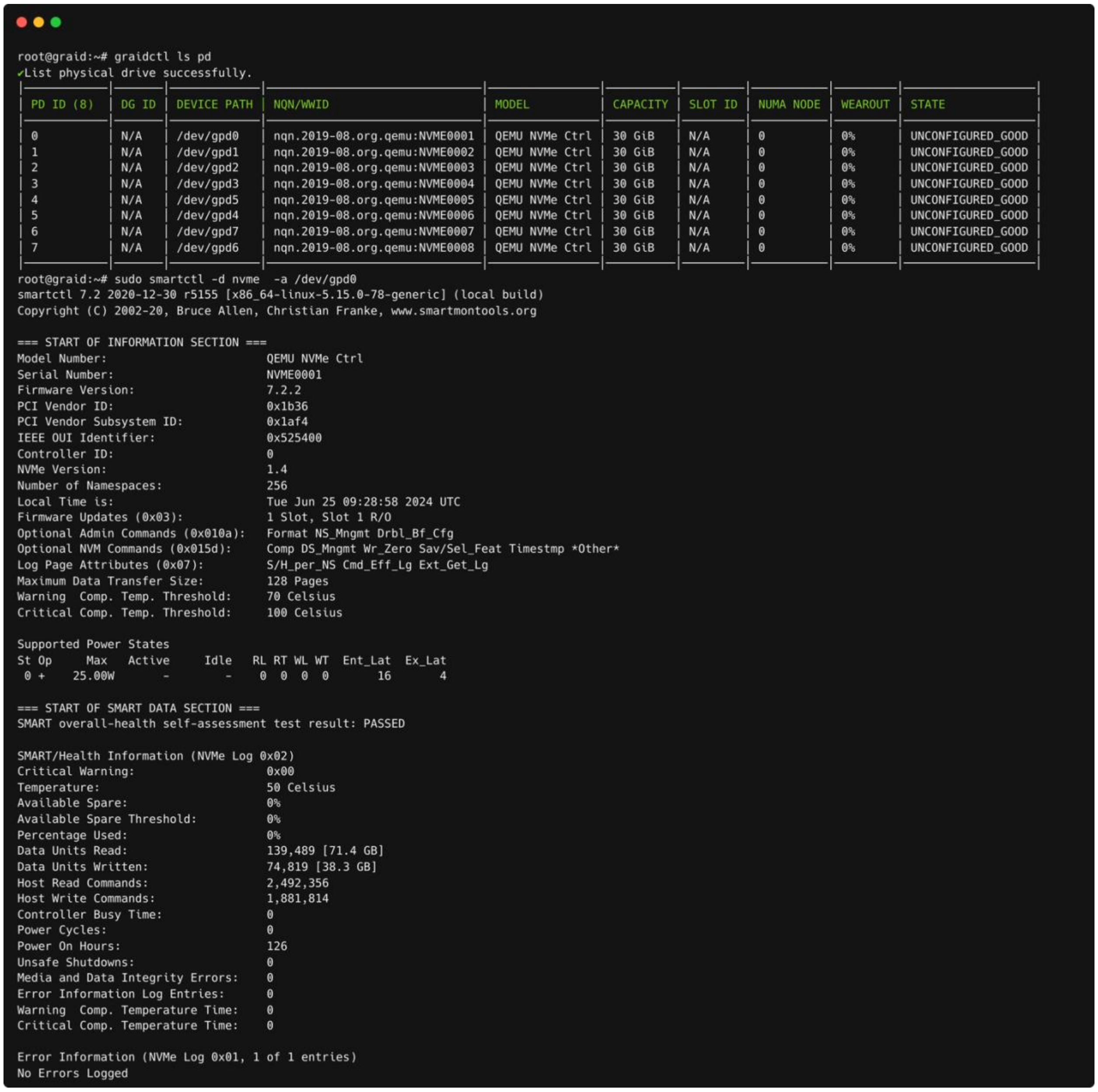
Note: The specific steps and commands used to display SMART data may vary, depending on your system and the version of the nvme or smartctl command in use. Be sure to use the correct device name for the gpd device in the command.

The following figure shows an output example using nvme smart-log.



```
root@graid:~# graidctl ls pd
✓List physical drive successfully.
|------------|--------|-------------|------------------------------|-------------|----------|---------|-----------|---------|------------------|
| PD ID (8)  | DG ID  | DEVICE PATH | NQN/WWID                     | MODEL       | CAPACITY | SLOT ID | NUMA NODE | WEAROUT | STATE            |
|------------|--------|-------------|------------------------------|-------------|----------|---------|-----------|---------|------------------|
| 0          | N/A    | /dev/gpd0   | nqn.2019-08.org.qemu:NVME0001 | QEMU NVMe Ctrl | 30 GiB | N/A    | 0         | 0%      | UNCONFIGURED_GOOD |
| 1          | N/A    | /dev/gpd1   | nqn.2019-08.org.qemu:NVME0002 | QEMU NVMe Ctrl | 30 GiB | N/A    | 0         | 0%      | UNCONFIGURED_GOOD |
| 2          | N/A    | /dev/gpd2   | nqn.2019-08.org.qemu:NVME0003 | QEMU NVMe Ctrl | 30 GiB | N/A    | 0         | 0%      | UNCONFIGURED_GOOD |
| 3          | N/A    | /dev/gpd3   | nqn.2019-08.org.qemu:NVME0004 | QEMU NVMe Ctrl | 30 GiB | N/A    | 0         | 0%      | UNCONFIGURED_GOOD |
| 4          | N/A    | /dev/gpd5   | nqn.2019-08.org.qemu:NVME0005 | QEMU NVMe Ctrl | 30 GiB | N/A    | 0         | 0%      | UNCONFIGURED_GOOD |
| 5          | N/A    | /dev/gpd4   | nqn.2019-08.org.qemu:NVME0006 | QEMU NVMe Ctrl | 30 GiB | N/A    | 0         | 0%      | UNCONFIGURED_GOOD |
| 6          | N/A    | /dev/gpd7   | nqn.2019-08.org.qemu:NVME0007 | QEMU NVMe Ctrl | 30 GiB | N/A    | 0         | 0%      | UNCONFIGURED_GOOD |
| 7          | N/A    | /dev/gpd6   | nqn.2019-08.org.qemu:NVME0008 | QEMU NVMe Ctrl | 30 GiB | N/A    | 0         | 0%      | UNCONFIGURED_GOOD |
|------------|--------|-------------|------------------------------|-------------|----------|---------|-----------|---------|------------------|
root@graid:~# sudo nvme smart-log  /dev/gpd0
Smart Log for NVME device:gpd0 namespace-id:ffffffff
critical_warning                        : 0
temperature                             : 50 C (323 Kelvin)
available_spare                         : 0%
available_spare_threshold               : 0%
percentage_used                         : 0%
endurance group critical warning summary: 0
data_units_read                         : 139,489
data_units_written                      : 74,819
host_read_commands                      : 2,492,356
host_write_commands                     : 1,881,814
controller_busy_time                    : 0
power_cycles                            : 0
power_on_hours                          : 126
unsafe_shutdowns                        : 0
media_errors                            : 0
num_err_log_entries                     : 0
Warning Temperature Time                : 0
Critical Composite Temperature Time     : 0
Thermal Management T1 Trans Count       : 0
Thermal Management T2 Trans Count       : 0
Thermal Management T1 Total Time        : 0
Thermal Management T2 Total Time        : 0
```

The following figure shows an output example using smartctl.



```
root@graid:~# graidctl ls pd
✓List physical drive successfully.
|──────────|────────|─────────────|────────────────────────────────|────────────────|──────────|─────────|────────────|─────────|──────────────────|
| PD ID (8) | DG ID | DEVICE PATH | NQN/WWID                       | MODEL          | CAPACITY | SLOT ID | NUMA NODE  | WEAROUT | STATE            |
|──────────|────────|─────────────|────────────────────────────────|────────────────|──────────|─────────|────────────|─────────|──────────────────|
| 0        | N/A    | /dev/gpd0   | nqn.2019-08.org.qemu:NVME0001 | QEMU NVMe Ctrl | 30 GiB   | N/A     | 0          | 0%      | UNCONFIGURED_GOOD |
| 1        | N/A    | /dev/gpd1   | nqn.2019-08.org.qemu:NVME0002 | QEMU NVMe Ctrl | 30 GiB   | N/A     | 0          | 0%      | UNCONFIGURED_GOOD |
| 2        | N/A    | /dev/gpd2   | nqn.2019-08.org.qemu:NVME0003 | QEMU NVMe Ctrl | 30 GiB   | N/A     | 0          | 0%      | UNCONFIGURED_GOOD |
| 3        | N/A    | /dev/gpd3   | nqn.2019-08.org.qemu:NVME0004 | QEMU NVMe Ctrl | 30 GiB   | N/A     | 0          | 0%      | UNCONFIGURED_GOOD |
| 4        | N/A    | /dev/gpd5   | nqn.2019-08.org.qemu:NVME0005 | QEMU NVMe Ctrl | 30 GiB   | N/A     | 0          | 0%      | UNCONFIGURED_GOOD |
| 5        | N/A    | /dev/gpd4   | nqn.2019-08.org.qemu:NVME0006 | QEMU NVMe Ctrl | 30 GiB   | N/A     | 0          | 0%      | UNCONFIGURED_GOOD |
| 6        | N/A    | /dev/gpd7   | nqn.2019-08.org.qemu:NVME0007 | QEMU NVMe Ctrl | 30 GiB   | N/A     | 0          | 0%      | UNCONFIGURED_GOOD |
| 7        | N/A    | /dev/gpd6   | nqn.2019-08.org.qemu:NVME0008 | QEMU NVMe Ctrl | 30 GiB   | N/A     | 0          | 0%      | UNCONFIGURED_GOOD |
|──────────|────────|─────────────|────────────────────────────────|────────────────|──────────|─────────|────────────|─────────|──────────────────|
root@graid:~# sudo smartctl -d nvme  -a /dev/gpd0
smartctl 7.2 2020-12-30 r5155 [x86_64-linux-5.15.0-78-generic] (local build)
Copyright (C) 2002-20, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF INFORMATION SECTION ===
Model Number:                       QEMU NVMe Ctrl
Serial Number:                      NVME0001
Firmware Version:                   7.2.2
PCI Vendor ID:                      0x1b36
PCI Vendor Subsystem ID:            0x1af4
IEEE OUI Identifier:                0x525400
Controller ID:                      0
NVMe Version:                       1.4
Number of Namespaces:               256
Local Time is:                      Tue Jun 25 09:28:58 2024 UTC
Firmware Updates (0x03):            1 Slot, Slot 1 R/O
Optional Admin Commands (0x010a):   Format NS_Mngmt Drbl_Bf_Cfg
Optional NVM Commands (0x015d):     Comp DS_Mngmt Wr_Zero Sav/Sel_Feat Timestmp *Other*
Log Page Attributes (0x07):         S/H_per_NS Cmd_Eff_Lg Ext_Get_Lg
Maximum Data Transfer Size:         128 Pages
Warning  Comp. Temp. Threshold:     70 Celsius
Critical Comp. Temp. Threshold:     100 Celsius

Supported Power States
St Op     Max   Active     Idle   RL RT WL WT  Ent_Lat  Ex_Lat
 0 +    25.00W       -        -    0  0  0  0       16       4

=== START OF SMART DATA SECTION ===
SMART overall-health self-assessment test result: PASSED

SMART/Health Information (NVMe Log 0x02)
Critical Warning:                   0x00
Temperature:                        50 Celsius
Available Spare:                    0%
Available Spare Threshold:          0%
Percentage Used:                    0%
Data Units Read:                    139,489 [71.4 GB]
Data Units Written:                 74,819 [38.3 GB]
Host Read Commands:                 2,492,356
Host Write Commands:                1,881,814
Controller Busy Time:               0
Power Cycles:                       0
Power On Hours:                     126
Unsafe Shutdowns:                   0
Media and Data Integrity Errors:    0
Error Information Log Entries:      0
Warning  Comp. Temperature Time:    0
Critical Comp. Temperature Time:    0

Error Information (NVMe Log 0x01, 1 of 1 entries)
No Errors Logged
```

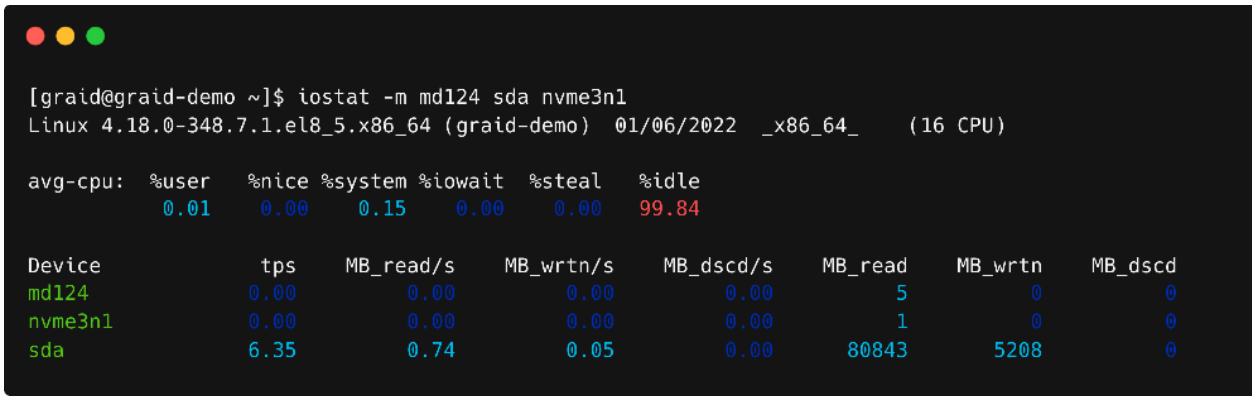
Monitoring System Input/Output Statistics for Devices Using iostat

The sysstat package contains the tools most commonly used to monitor I/O statistics in Linux systems. The sysstat package includes the iostat tool, which monitors system I/O device loading by observing the time the devices are active relative to their average transfer rates. The **iostat** command generates reports that allow you to fine-tune the system configuration to better balance the I/O load between physical disks.

For example, to monitor specific devices and display statistics in megabytes per second (Mbps), issue the following command:

```
$ iostat -m md124 sda nvme0n1
```

The following figure shows an output example.



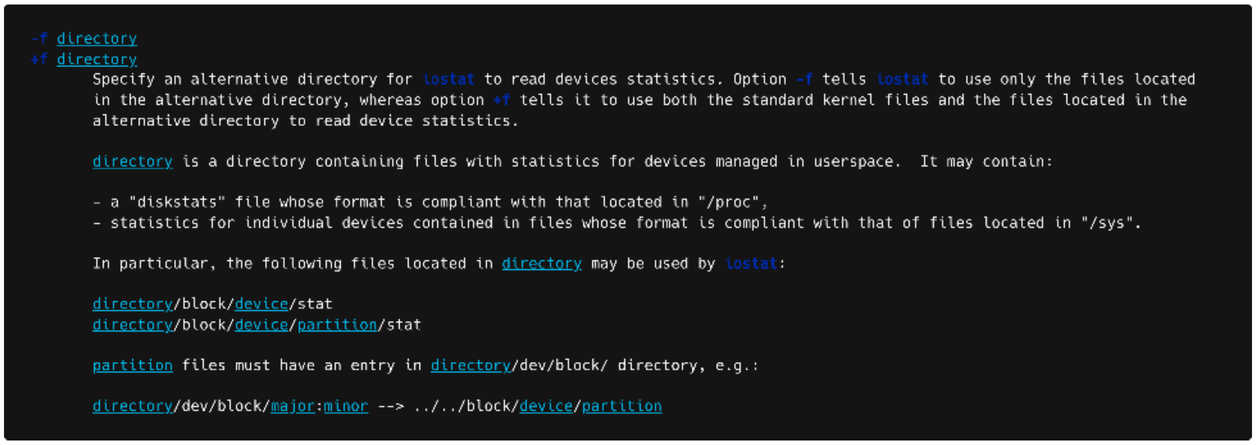
```
[graid@graid-demo ~]$ iostat -m md124 sda nvme3n1
Linux 4.18.0-348.7.1.el8_5.x86_64 (graid-demo)  01/06/2022  _x86_64_    (16 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.01    0.00    0.15    0.00    0.00   99.84

Device             tps    MB_read/s    MB_wrtn/s    MB_dscd/s    MB_read    MB_wrtn    MB_dscd
md124             0.00         0.00         0.00         0.00          5          0          0
nvme3n1           0.00         0.00         0.00         0.00          1          0          0
sda               6.35         0.74         0.05         0.00      80843       5208          0
```

sysstat Versions v12.3.3 and Later

For sysstat versions v12.3.3 and later, the iostat tool includes an alternative directory feature that allows you to specify the directory from which to read device statistics.

- Add a **+f** parameter to the tool and use the /sys/devices/virtual/graid/graid sysfs device path to read device statistics from both the standard kernel files and the files in the alternative directory.
- Add a **-f** parameter to the tool and use the /sys/devices/virtual/graid/graid sysfs device path to read device statistics from the files in the alternative directory.

The following figure shows an alternative directory description from the iostat manual page.

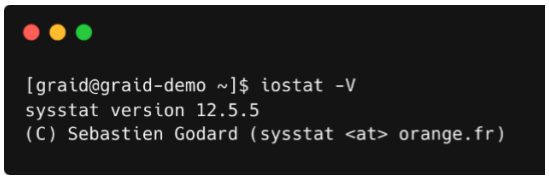


```
-f directory
+f directory
       Specify an alternative directory for iostat to read devices statistics. Option -f tells iostat to use only the files located
       in the alternative directory, whereas option +f tells it to use both the standard kernel files and the files located in the
       alternative directory to read device statistics.

       directory is a directory containing files with statistics for devices managed in userspace.  It may contain:

       - a "diskstats" file whose format is compliant with that located in "/proc",
       - statistics for individual devices contained in files whose format is compliant with that of files located in "/sys".

       In particular, the following files located in directory may be used by iostat:

       directory/block/device/stat
       directory/block/device/partition/stat

       partition files must have an entry in directory/dev/block/ directory, e.g.:

       directory/dev/block/major:minor --> ../../block/device/partition
```

To check the iostat version, issue the following command:

```
$ iostat -V
```

The following figure shows an output example.

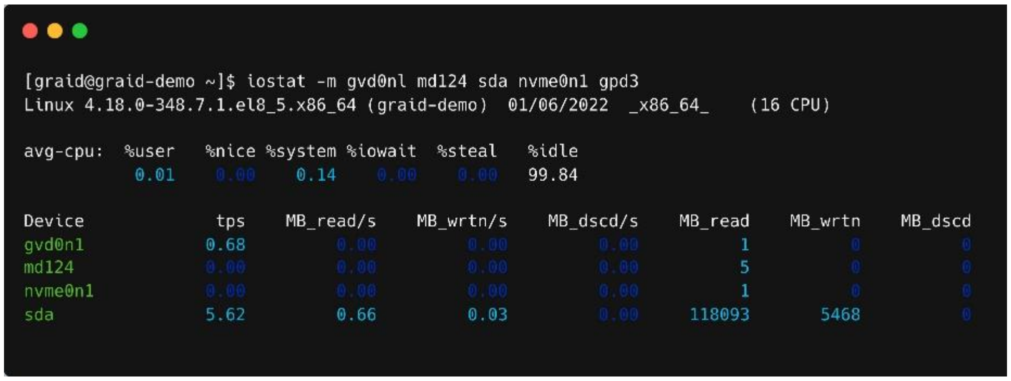


```
[graid@graid-demo ~]$ iostat -V
sysstat version 12.5.5
(C) Sebastien Godard (sysstat <at> orange.fr)
```

The gpd# statistics are not displayed in the iostat report without appending the +f parameter and defining the sysfs path.

```
$ iostat -m +f /sys/devices/virtual/graid/graid gdg0nl md124 sda nvme0n1 gpd3
```

The following figure shows an output example.

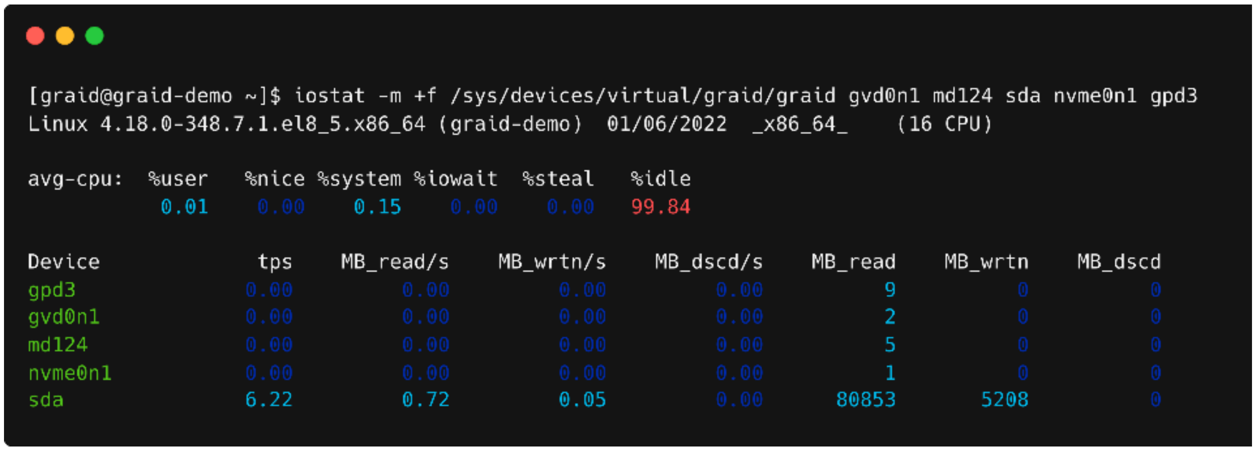


```
[graid@graid-demo ~]$ iostat -m gvd0n1 md124 sda nvme0n1 gpd3
Linux 4.18.0-348.7.1.el8_5.x86_64 (graid-demo)  01/06/2022  _x86_64_    (16 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.01    0.00    0.14    0.00    0.00   99.84

Device             tps    MB_read/s    MB_wrtn/s    MB_dscd/s    MB_read    MB_wrtn    MB_dscd
gvd0n1            0.68         0.00         0.00         0.00          1          0          0
md124             0.00         0.00         0.00         0.00          5          0          0
nvme0n1           0.00         0.00         0.00         0.00          1          0          0
sda               5.62         0.66         0.03         0.00     118093       5468          0
```

The gpd# statistics are displayed when the +f parameter is appended and the sysfs path is defined.

```
$ iostat -m +f /sys/devices/virtual/graid/graid gdg0nl md124 sda nvme0n1 gpd3
```

The following figure shows an output example.



```
[graid@graid-demo ~]$ iostat -m +f /sys/devices/virtual/graid/graid gvd0n1 md124 sda nvme0n1 gpd3
Linux 4.18.0-348.7.1.el8_5.x86_64 (graid-demo)  01/06/2022  _x86_64_    (16 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.01    0.00    0.15    0.00    0.00   99.84

Device             tps    MB_read/s    MB_wrtn/s    MB_dscd/s    MB_read    MB_wrtn    MB_dscd
gpd3              0.00         0.00         0.00         0.00          9          0          0
gvd0n1            0.00         0.00         0.00         0.00          2          0          0
md124             0.00         0.00         0.00         0.00          5          0          0
nvme0n1           0.00         0.00         0.00         0.00          1          0          0
sda               6.22         0.72         0.05         0.00      80853       5208          0
```

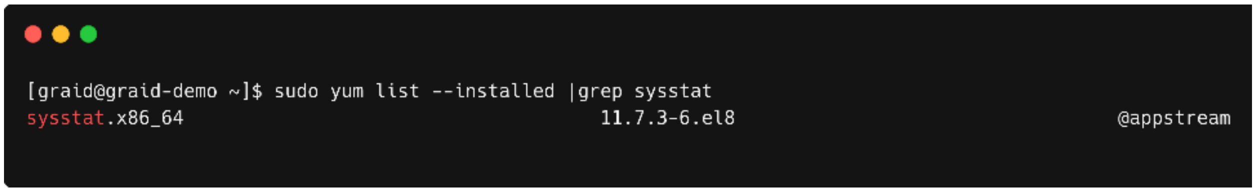
sysstat Versions Prior to v12.3.3

For operating systems with sysstat versions prior to v12.3.3 (for example, CentOS), Graid Technology provides an alternate tool called giostat to display device statistics.

In the following example, the operating system version of iostat is prior to v12.3.3.

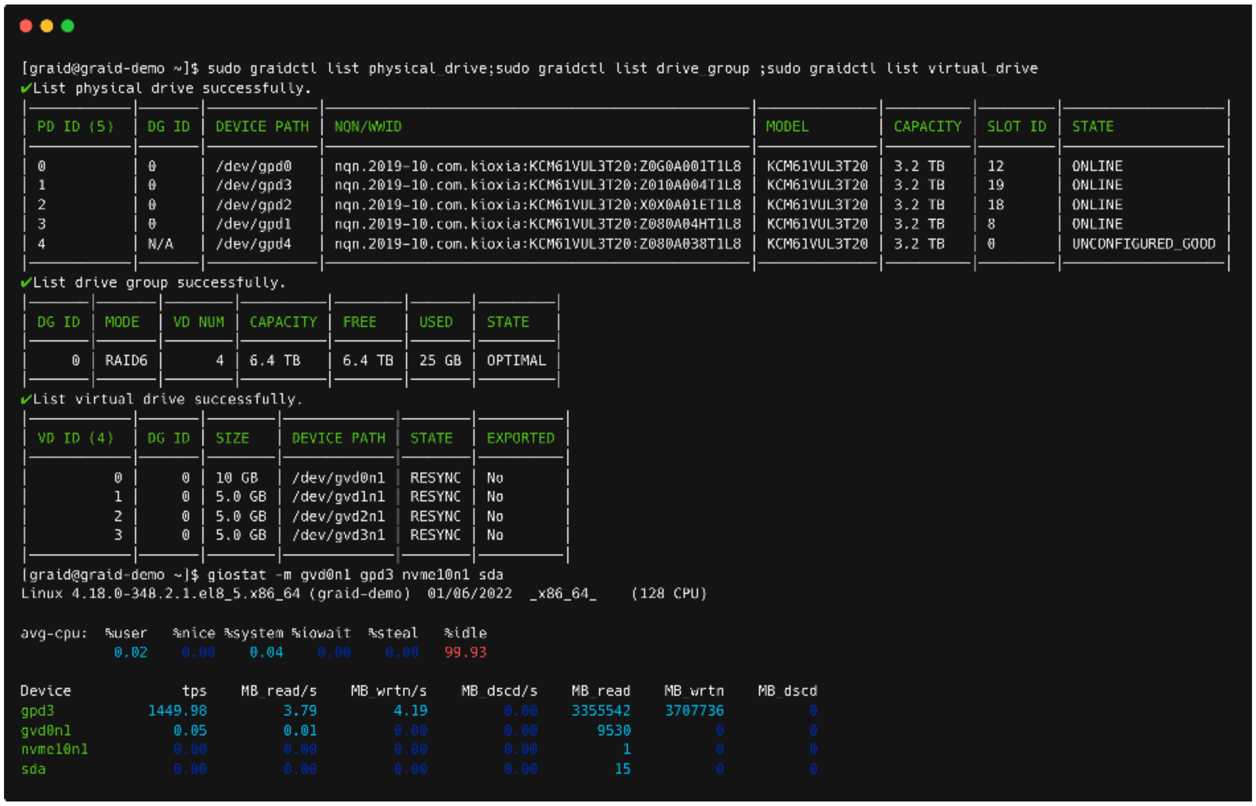
```
$ sudo yum list --installed |grep sysstat
```

The following figure shows an output example.



```
[graid@graid-demo ~]$ sudo yum list --installed |grep sysstat
sysstat.x86_64                                    11.7.3-6.el8                                  @appstream
```

The giostat and iostat tools are very similar and their usage is the same. Set the parameter preferences using giostat. The following figure shows an output example.



```
[graid@graid-demo ~]$ sudo graidctl list physical_drive;sudo graidctl list drive_group ;sudo graidctl list virtual_drive
✔List physical drive successfully.
|-----------|--------|-------------|---------------------------------------------------|--------------|----------|---------|----------------|
| PD ID (5) | DG ID  | DEVICE PATH | NQN/WWID                                          | MODEL        | CAPACITY | SLOT ID | STATE          |
|-----------|--------|-------------|---------------------------------------------------|--------------|----------|---------|----------------|
| 0         | 0      | /dev/gpd0   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z0G0A001T1L8  | KCM61VUL3T20 | 3.2 TB   | 12      | ONLINE         |
| 1         | 0      | /dev/gpd3   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z010A004T1L8  | KCM61VUL3T20 | 3.2 TB   | 19      | ONLINE         |
| 2         | 0      | /dev/gpd2   | nqn.2019-10.com.kioxia:KCM61VUL3T20:X0X0A01ET1L8  | KCM61VUL3T20 | 3.2 TB   | 18      | ONLINE         |
| 3         | 0      | /dev/gpd1   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A04HT1L8  | KCM61VUL3T20 | 3.2 TB   | 8       | ONLINE         |
| 4         | N/A    | /dev/gpd4   | nqn.2019-10.com.kioxia:KCM61VUL3T20:Z080A038T1L8  | KCM61VUL3T20 | 3.2 TB   | 0       | UNCONFIGURED_GOOD |
|-----------|--------|-------------|---------------------------------------------------|--------------|----------|---------|----------------|
✔List drive group successfully.
|-------|-------|--------|----------|--------|-------|---------|
| DG ID | MODE  | VD NUM | CAPACITY | FREE   | USED  | STATE   |
|-------|-------|--------|----------|--------|-------|---------|
|     0 | RAID6 |      4 | 6.4 TB   | 6.4 TB | 25 GB | OPTIMAL |
|-------|-------|--------|----------|--------|-------|---------|
✔List virtual drive successfully.
|-----------|--------|--------|--------------|--------|----------|
| VD ID (4) | DG ID  | SIZE   | DEVICE PATH  | STATE  | EXPORTED |
|-----------|--------|--------|--------------|--------|----------|
|         0 |      0 | 10 GB  | /dev/gvd0n1  | RESYNC | No       |
|         1 |      0 | 5.0 GB | /dev/gvd1n1  | RESYNC | No       |
|         2 |      0 | 5.0 GB | /dev/gvd2n1  | RESYNC | No       |
|         3 |      0 | 5.0 GB | /dev/gvd3n1  | RESYNC | No       |
|-----------|--------|--------|--------------|--------|----------|
[graid@graid-demo ~]$ giostat -m gvd0n1 gpd3 nvme10n1 sda
Linux 4.18.0-348.2.1.el8_5.x86_64 (graid-demo)  01/06/2022  _x86_64_    (128 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.02    0.00    0.04    0.00    0.00   99.93

Device             tps    MB_read/s    MB_wrtn/s    MB_dscd/s    MB_read    MB_wrtn    MB_dscd
gpd3           1449.98         3.79         4.19         0.00    3355542    3707736          0
gvd0n1            0.05         0.01         0.00         0.00       9530          0          0
nvme10n1          0.00         0.00         0.00         0.00          1          0          0
sda               0.00         0.00         0.00         0.00         15          0          0
```

Setting Up the Auto-mount File Systems on Linux Using the SupremeRAID™ Driver

To set up the auto-mount file systems on Linux using the SupremeRAID™ driver:

Step 1 Create a virtual drive.

```
$ sudo graidctl create virtual_drive [DG_ID] [size] [flags]
```

Step 2 Format the virtual drive and create a mount point for it.

```
$ sudo mkdir /mnt/[name-of-the-drive]
$ sudo mkfs.[file-system-type] /dev/gdgXnY
$ sudo mount /dev/gdgXnY /mnt/[name-of-the-drive]/
```

Step 3 Obtain the name, and file system type.

```
$ ls -l  /dev/[disk]/[by-id]/
```

Step 4 Edit the /etc/fstab file:

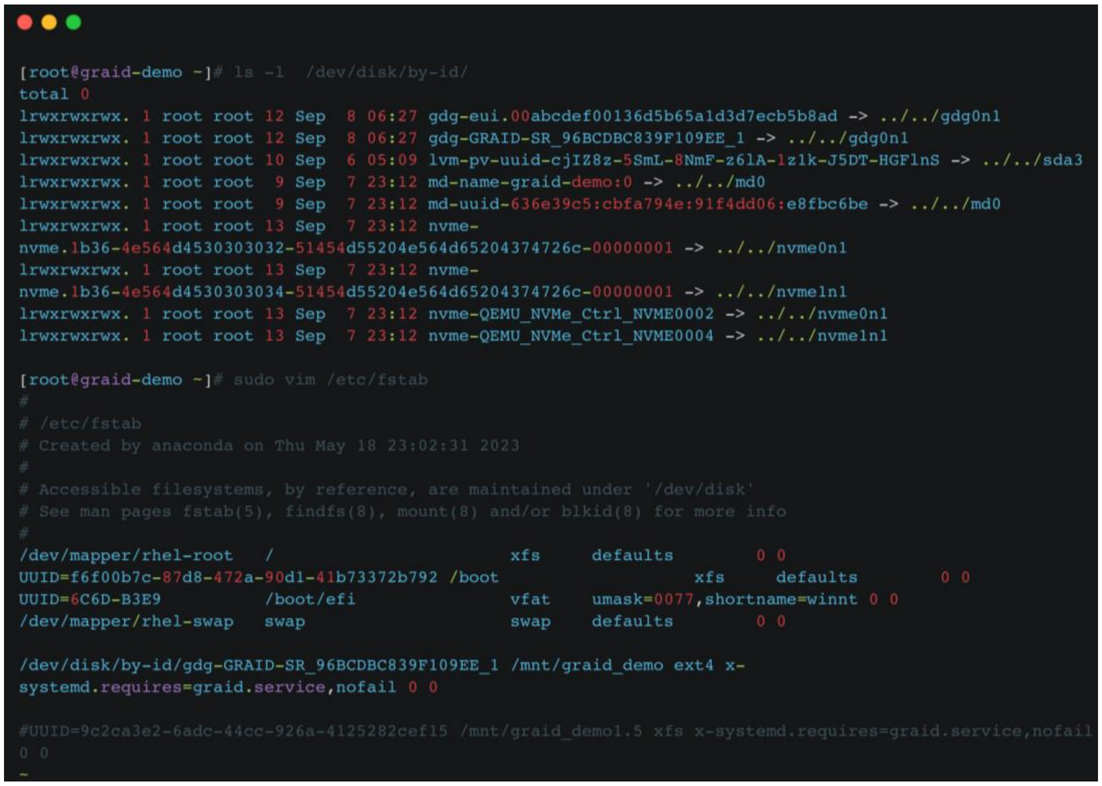
A Edit the /etc/fstab file.

```
$ sudo vim /etc/fstab
```

B Append one line of code to the end of the file using the following format.

```
$ /dev/[disk]/[by-id] [mount-point] [file-system-format]
x-systemd.requires=graid.service,nofail [dump] [pass]
```

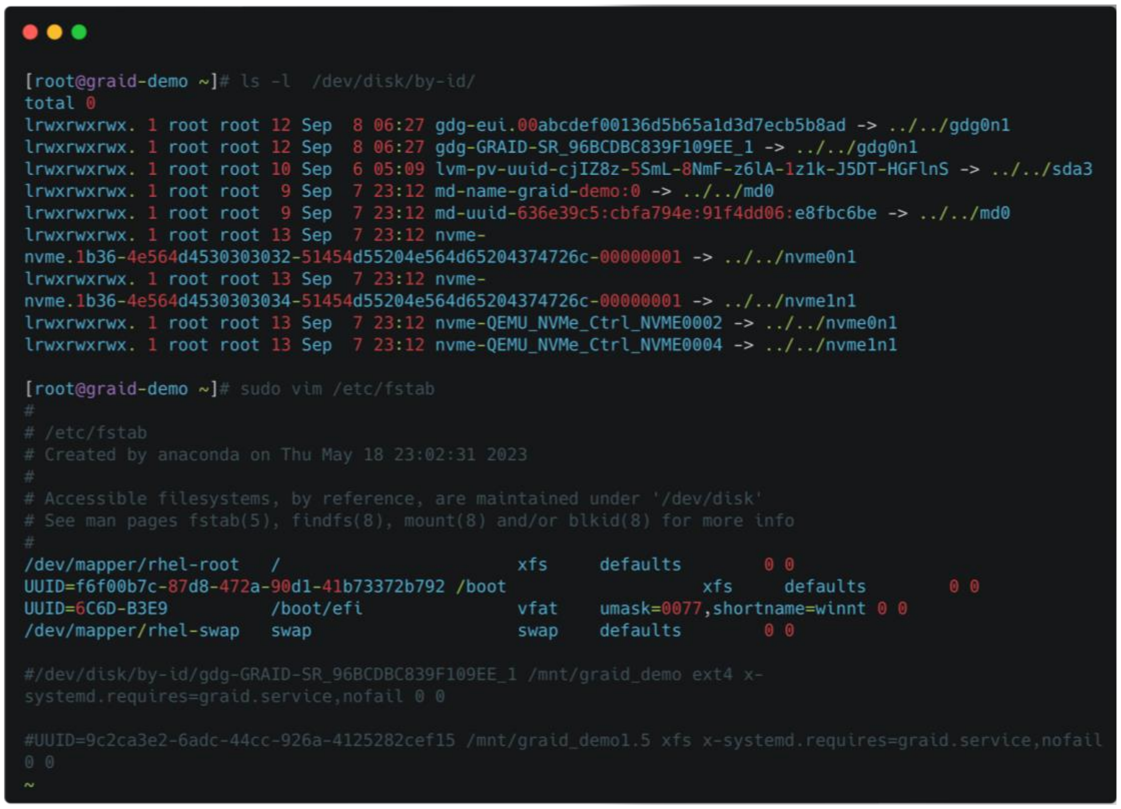
C Show the output example.



```
[root@graid-demo ~]# ls -l  /dev/disk/by-id/
total 0
lrwxrwxrwx. 1 root root 12 Sep  8 06:27 gdg-eui.00abcdef00136d5b65a1d3d7ecb5b8ad -> ../../gdg0n1
lrwxrwxrwx. 1 root root 12 Sep  8 06:27 gdg-GRAID-SR_96BCDBC839F109EE_1 -> ../../gdg0n1
lrwxrwxrwx. 1 root root 10 Sep  6 05:09 lvm-pv-uuid-cjIZ8z-5SmL-8NmF-z61A-1z1k-J5DT-HGFlnS -> ../../sda3
lrwxrwxrwx. 1 root root  9 Sep  7 23:12 md-name-graid-demo:0 -> ../../md0
lrwxrwxrwx. 1 root root  9 Sep  7 23:12 md-uuid-636e39c5:cbfa794e:91f4dd06:e8fbc6be -> ../../md0
lrwxrwxrwx. 1 root root 13 Sep  7 23:12 nvme-
nvme.1b36-4e564d4530303032-51454d55204e564d65204374726c-00000001 -> ../../nvme0n1
lrwxrwxrwx. 1 root root 13 Sep  7 23:12 nvme-
nvme.1b36-4e564d4530303034-51454d55204e564d65204374726c-00000001 -> ../../nvme1n1
lrwxrwxrwx. 1 root root 13 Sep  7 23:12 nvme-QEMU_NVMe_Ctrl_NVME0002 -> ../../nvme0n1
lrwxrwxrwx. 1 root root 13 Sep  7 23:12 nvme-QEMU_NVMe_Ctrl_NVME0004 -> ../../nvme1n1

[root@graid-demo ~]# sudo vim /etc/fstab
#
# /etc/fstab
# Created by anaconda on Thu May 18 23:02:31 2023
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/rhel-root   /                       xfs     defaults        0 0
UUID=f6f00b7c-87d8-472a-90d1-41b73372b792 /boot                   xfs     defaults        0 0
UUID=6C6D-B3E9          /boot/efi               vfat    umask=0077,shortname=winnt 0 0
/dev/mapper/rhel-swap   swap                    swap    defaults        0 0

/dev/disk/by-id/gdg-GRAID-SR_96BCDBC839F109EE_1 /mnt/graid_demo ext4 x-
systemd.requires=graid.service,nofail 0 0

#UUID=9c2ca3e2-6adc-44cc-926a-4125282cef15 /mnt/graid_demo1.5 xfs x-systemd.requires=graid.service,nofail
0 0
~
```

Step 5 Remove the device line and reboot the system.

```
$ sudo vim /etc/fstab
```



```
[root@graid-demo ~]# ls -l  /dev/disk/by-id/
total 0
lrwxrwxrwx. 1 root root 12 Sep  8 06:27 gdg-eui.00abcdef00136d5b65a1d3d7ecb5b8ad -> ../../gdg0n1
lrwxrwxrwx. 1 root root 12 Sep  8 06:27 gdg-GRAID-SR_96BCDBC839F109EE_1 -> ../../gdg0n1
lrwxrwxrwx. 1 root root 10 Sep  6 05:09 lvm-pv-uuid-cjIZ8z-5SmL-8NmF-z6lA-1z1k-J5DT-HGFlnS -> ../../sda3
lrwxrwxrwx. 1 root root  9 Sep  7 23:12 md-name-graid-demo:0 -> ../../md0
lrwxrwxrwx. 1 root root  9 Sep  7 23:12 md-uuid-636e39c5:cbfa794e:91f4dd06:e8fbc6be -> ../../md0
lrwxrwxrwx. 1 root root 13 Sep  7 23:12 nvme-
nvme.1b36-4e564d4530303032-51454d55204e564d65204374726c-00000001 -> ../../nvme0n1
lrwxrwxrwx. 1 root root 13 Sep  7 23:12 nvme-
nvme.1b36-4e564d4530303034-51454d55204e564d65204374726c-00000001 -> ../../nvme1n1
lrwxrwxrwx. 1 root root 13 Sep  7 23:12 nvme-QEMU_NVMe_Ctrl_NVME0002 -> ../../nvme0n1
lrwxrwxrwx. 1 root root 13 Sep  7 23:12 nvme-QEMU_NVMe_Ctrl_NVME0004 -> ../../nvme1n1

[root@graid-demo ~]# sudo vim /etc/fstab
#
# /etc/fstab
# Created by anaconda on Thu May 18 23:02:31 2023
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/rhel-root   /                       xfs     defaults        0 0
UUID=f6f00b7c-87d8-472a-90d1-41b73372b792 /boot                   xfs     defaults        0 0
UUID=6C6D-B3E9          /boot/efi               vfat    umask=0077,shortname=winnt 0 0
/dev/mapper/rhel-swap   swap                    swap    defaults        0 0

#/dev/disk/by-id/gdg-GRAID-SR_96BCDBC839F109EE_1 /mnt/graid_demo ext4 x-
systemd.requires=graid.service,nofail 0 0

#UUID=9c2ca3e2-6adc-44cc-926a-4125282cef15 /mnt/graid_demo1.5 xfs x-systemd.requires=graid.service,nofail
0 0
~
```

Note: To disable the automount point or delete the virtual drive, edit the /etc/fstab file to delete/comment that entry, and then reboot the system.

ESXi Virtual Machine Support Using GPU Passthrough

You can create virtual machines with SupremeRAID™ support to maximize performance.

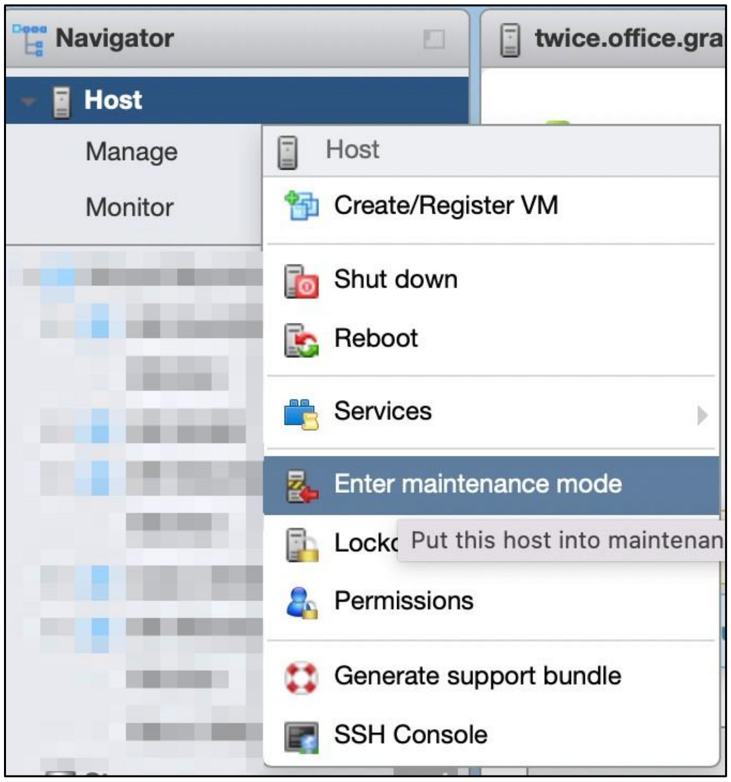
The following procedure describes how to set a single VM with SupremeRAID™. This setup is for use only within a single virtual machine and cannot be shared from the volume back to ESXi to a datastore for other virtual machines.

Hypervisor VMware support is ESXi 7.0U3.

Configuring Hosts for NVIDIA GPU Device Passthrough

Setting the ESXi Host in Maintenance Mode

From the Navigator menu, select Host > Enter maintenance mode.



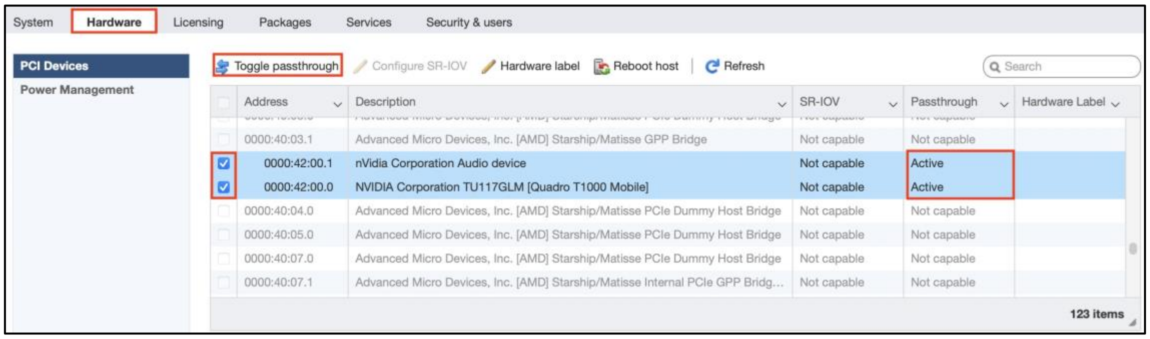
Managing PCI Device Passthrough

Step 1 From the Navigator menu, select Manage > Hardware > PCI Devices. The Passthrough Configuration page appears, listing all available passthrough devices.

Step 2 Select the NVIDIA T1000 (Quadro T1000 Mobile) and its Audio device.

Step 3 Click Toggle passthrough.

Step 4 Confirm that the Passthrough status is Active.



	Address	Description	SR-IOV	Passthrough	Hardware Label
☐	0000:40:03.1	Advanced Micro Devices, Inc. [AMD] Starship/Matisse GPP Bridge	Not capable	Not capable	
✓	0000:42:00.1	nVidia Corporation Audio device	Not capable	Active	
✓	0000:42:00.0	NVIDIA Corporation TU117GLM [Quadro T1000 Mobile]	Not capable	Active	
☐	0000:40:04.0	Advanced Micro Devices, Inc. [AMD] Starship/Matisse PCIe Dummy Host Bridge	Not capable	Not capable	
☐	0000:40:05.0	Advanced Micro Devices, Inc. [AMD] Starship/Matisse PCIe Dummy Host Bridge	Not capable	Not capable	
☐	0000:40:07.0	Advanced Micro Devices, Inc. [AMD] Starship/Matisse PCIe Dummy Host Bridge	Not capable	Not capable	
☐	0000:40:07.1	Advanced Micro Devices, Inc. [AMD] Starship/Matisse Internal PCIe GPP Bridg...	Not capable	Not capable	

123 items

Note: If you move the SupremeRAID™ card to a different hardware slot or plan to do so, you MUST cancel its passthrough before shutting down the ESXi server. After the hardware change, you MUST set up the passthrough again; otherwise, the virtual machine will not recognize the PCIe device properly.

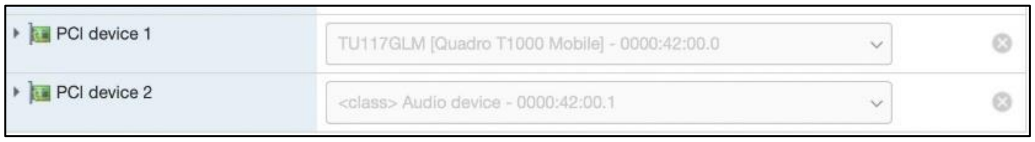
Configuring Virtual Machines

Attaching PCI Devices to the Virtual Machine

To attach PCI devices to the virtual machine:

Step 1 From the Edit VM setting page, select Virtual Hardware > Add other device > PCI device.

Step 2 Select Quadro T1000 and its Audio device as the two PCI devices.

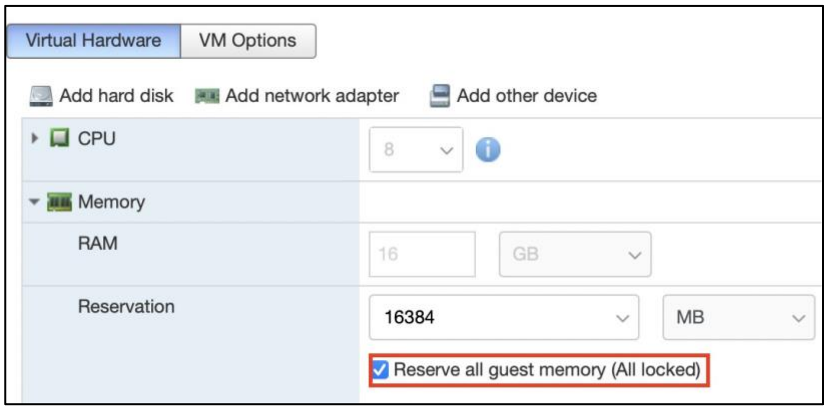


PCI device 1	TU117GLM [Quadro T1000 Mobile] - 0000:42:00.0
PCI device 2	<class> Audio device - 0000:42:00.1

Note: When the T1000 PCI device is assigned to the virtual machine, you must set the memory reservation to accommodate the fully configured memory size.

Step 3 Select Virtual Hardware > Memory.

Step 4 Check Reserve all guest memory (All locked).

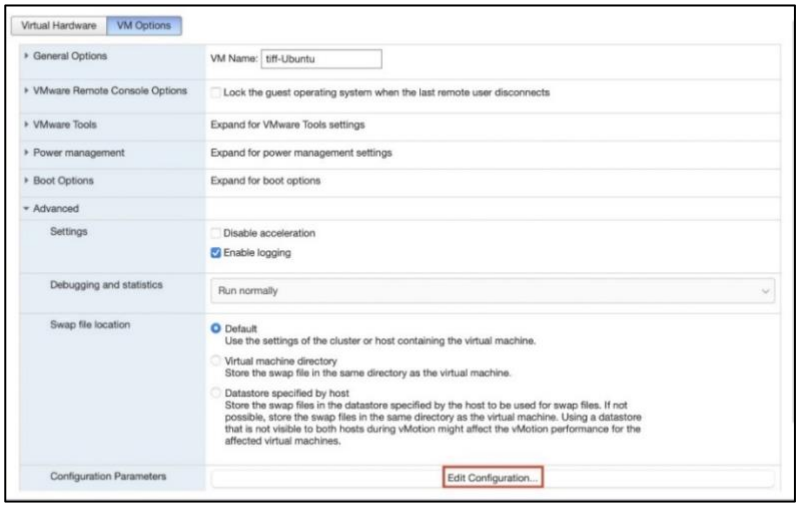


CPU	8
RAM	16 GB
Reservation	16384 MB
	☑ Reserve all guest memory (All locked)

Enabling Point-to-Point (P2P) on the Virtual Machine

Enabling P2P on the virtual machine optimizes performance. To enable P2P on the virtual machine:

Step 1 From the Edit VM setting page, select VM Options > Advanced > Configuration Parameters > Edit Configuration....



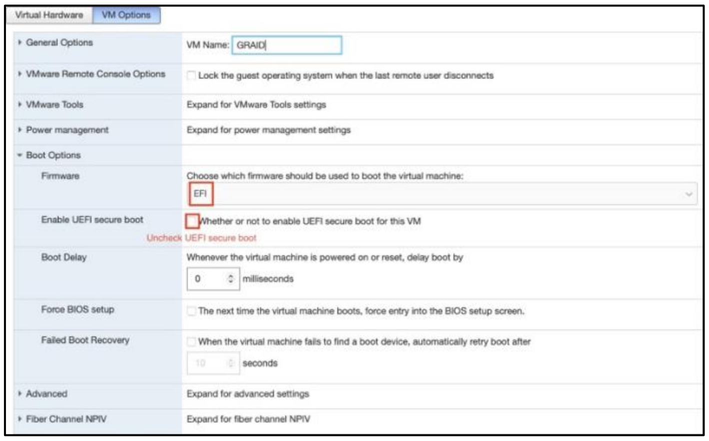
General Options	VM Name: tiff-Ubuntu
VMware Remote Console Options	☐ Lock the guest operating system when the last remote user disconnects
VMware Tools	Expand for VMware Tools settings
Power management	Expand for power management settings
Boot Options	Expand for boot options
Advanced	
Settings	☐ Disable acceleration ☑ Enable logging
Debugging and statistics	Run normally
Swap file location	● Default Use the settings of the cluster or host containing the virtual machine. ○ Virtual machine directory Store the swap file in the same directory as the virtual machine. ○ Datastore specified by host Store the swap files in the datastore specified by the host to be used for swap files. If not possible, store the swap files in the same directory as the virtual machine. Using a datastore that is not visible to both hosts during vMotion might affect the vMotion performance for the affected virtual machines.
Configuration Parameters	Edit Configuration...

Step 2 Add the following two parameters:

```
hypervisor.cpuid.v0 = "FALSE"
pciPassthru.allowP2P = "TRUE" pciPassthru.use64bitMMIO= "TRUE"
```

Step 3 From the Edit VM setting page, select VM Options > Boot Options > Firmware > EFI.

Step 4 Uncheck Whether or not to enable UEFI secure boot for this VM.



General Options	VM Name: GRAID
VMware Remote Console Options	☐ Lock the guest operating system when the last remote user disconnects
VMware Tools	Expand for VMware Tools settings
Power management	Expand for power management settings
Boot Options	
Firmware	Choose which firmware should be used to boot the virtual machine: EFI
Enable UEFI secure boot	☐ Whether or not to enable UEFI secure boot for this VM
Boot Delay	Whenever the virtual machine is powered on or reset, delay boot by 0 milliseconds
Force BIOS setup	☐ The next time the virtual machine boots, force entry into the BIOS setup screen.
Failed Boot Recovery	☐ When the virtual machine fails to find a boot device, automatically retry boot after 10 seconds
Advanced	Expand for advanced settings
Fiber Channel NPIV	Expand for fiber channel NPIV

Using Self-Encrypting Drives

A self-encrypting drive (SED) uses native full-disk encryption. SupremeRAID™ supports SEDs and SED key management. When the SED key is configured, SupremeRAID™ uses the imported key to unlock the SED.

Before configuring a SED, observe the following guidelines:

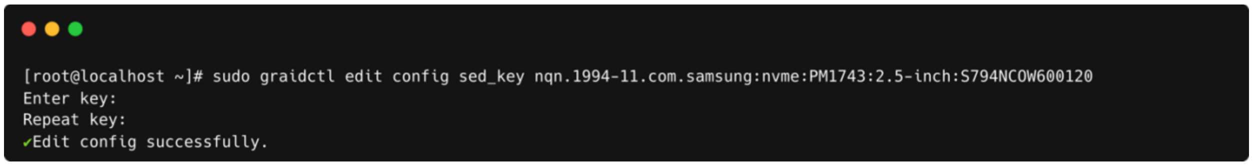
- Configure the SED key using the graidctl tool before creating the physical drives.
- Only NVMe devices are supported.
- Only the global range parameter is supported.

Importing a Single SED Key Using NQN/WWID

To import a single SED key using NQN/WWID, issue the following command:

```
$ sudo graidctl edit config sed_key [NQN/WWID]
```

The following figure shows an example.



```
[root@localhost ~]# sudo graidctl edit config sed_key nqn.1994-11.com.samsung:nvme:PM1743:2.5-inch:S794NCOW600120
Enter key:
Repeat key:
✓Edit config successfully.
```

Importing a Batched SED Key Using NQN/WWID

To import a batched SED key using NQN/WWID, issue the following command:

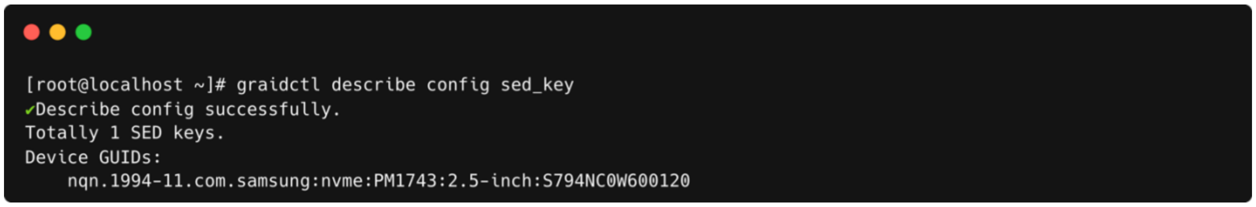
```
$ sudo graidctl edit config sed_key --input-file [filename]
file content format:
[NQN1/WWID1], [KEY1]
[NQN1/WWID1], [KEY2]
...
[NQNn/WWIDn], [KEYn]
```

Displaying SED Key Information

To display SED key information, issue the following command:

```
$ sudo graidctl describe config sed
```

The following figure shows an example.



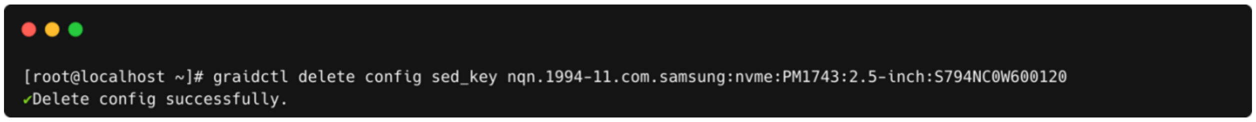
```
[root@localhost ~]# graidctl describe config sed_key
✓Describe config successfully.
Totally 1 SED keys.
Device GUIDs:
    nqn.1994-11.com.samsung:nvme:PM1743:2.5-inch:S794NC0W600120
```

Deleting SED Keys

To delete a SED key, issue the following command:

```
$ sudo graidctl delete config sed_key [GUID]
```

The following figure shows an example.

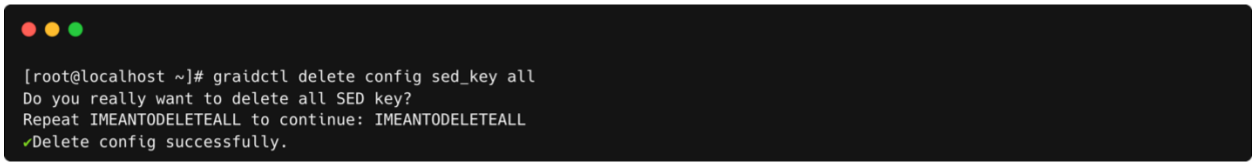


```
[root@localhost ~]# graidctl delete config sed_key nqn.1994-11.com.samsung:nvme:PM1743:2.5-inch:S794NC0W600120
✓Delete config successfully.
```

To delete all SED keys, issue the following command:

```
$ sudo graidctl delete config sed_key all
```

The following figure shows an example.



```
[root@localhost ~]# graidctl delete config sed_key all
Do you really want to delete all SED key?
Repeat IMEANTODELETEALL to continue: IMEANTODELETEALL
✓Delete config successfully.
```

Rotating SED Key Information

To rotate the SED key, issue the following command:

```
$ sudo graidctl edit pd 0 sed_key [ORIGINAL_KEY] [NEW_KEY]
```

To rotate multiple SED keys, issue the following command:

```
$ sudo graidctl edit pd 0-22 sed_key [ORIGINAL_KEY] [NEW_KEY]
```

Setup Mail Notification Service

SupremeRAID™ offers a daemon service in Linux that enables users to receive email notifications for monitoring service status. This includes actions like creating or deleting physical drives (PD), drive groups (DG), or virtual drives (VD) and so on.

Install the Mail Notification Service

Step 1 Download the Mail Notification installation package.

Step 2 Install the Mail Notification package.

- For CentOS, Rocky Linux, and Alma Linux.

```
$ sudo rpm -ivh [filename]
```

- For Ubuntu

```
$ sudo dpkg -i [filename]
```

Step 3 Edit the configuration file, save the changes, and then exit.

```
$ sudo vim /etc/graid/sendmail.toml
```

Step 4 Start the Mail Notification service.

```
$ sudo systemctl start graid-mail-notification
```

Step 5 Send a testing mail to specified email address.

```
$ sudo graid_mail_notification -t
```

Step 6 Check if the mail is received the correctly.

Note: The mail daemon service will monitor for changes every 30 seconds and send an email to the user's specified address whenever it detects events that alter the Graid management daemon status.

Remove the Mail Notification Service

For CentOS, RHEL Rocky Linux and Alma Linux

```
$ sudo rpm -e graid-mail-notification
```

For Ubuntu

```
$ sudo dpkg -r graid-mail-notification
```

Setup Graphical Management Console

SupremeRAID™ offers a graphical management console for user to control the RAID resource via web portal. This intuitive interface streamlines the process, enhancing user experience and operational fluency.

Install the Graphical Management Console Service

Step 1 Download the installer and finish SupremeRAID™ installation process.

Step 2 Apply license before enable the service, the license state should be 'APPLIED'.

```
$ sudo graidctl apply license <LICENSE_KEY>
$ sudo graidctl describe license
```

Step 3 Add port of management console into firewall service, then reload it.

```
$ sudo firewall-cmd --zone=public --add-port=50060/tcp
$ sudo firewall-cmd --zone=public --permanent --add-port=50060/tcp
$ sudo firewall-cmd --reload
```

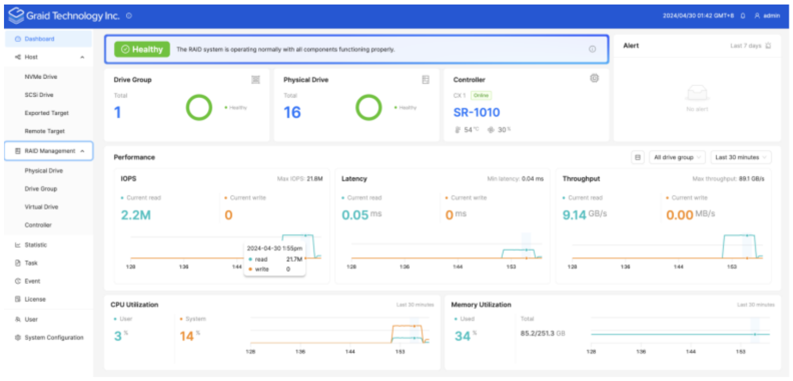
Step 4 Enable the graphical management console service and start it.

```
$ sudo systemctl enable graid-mgr.service
$ sudo systemctl start graid-mgr.service
```

OR

```
$ sudo systemctl --now enable graid-mgr.service
```

Step 5 Open web browser, key-in 'https://<SYSTEM-IP>:50060' and login with default account.



The RAID system is operating normally with all components functioning properly.

Note: The default account and password for graphical management console is 'admin' and 'admin'.

TROUBLESHOOTING

Sequential Read Performance is Not as Expected on a New Drive Group

Unlike SAS/SATA hard drives, many NVMe SSDs support the de-allocate dataset management command. Using this command, you can reset all data in the NVMe SSD immediately, eliminating the need to synchronize data between physical drives when creating a drive group.

For other SSDs, however, the performance is not as expected when reading unwritten sectors after issuing the de-allocate dataset management command. While this behavior also impacts the performance of the new drive group, it does not affect the applications because they do not read sectors that do not contain data.

To test SupremeRAID™ performance, write the entire virtual drive sequentially using a large block size.

Kernel Log Message "failed to set APST feature (-19)" Appears When Creating Physical Drives

Some NVMe SSD models might display a "failed to set APST feature (-19)" message in the kernel log when creating the physical drive.

When SupremeRAID™ creates the physical drive, the SSD is unbound from the operating system so the SupremeRAID™ can control the SSD. When the APST feature is enabled during the unbinding process, the NVMe driver tries and fails to set the APST state to SSD and the error message is issued. This message is expected and can be ignored. SupremeRAID™ is working normally.

Decoding LED Patterns on the Backplane

You might notice that the HDD/SSD activity indicator blink pattern is different on SupremeRAID™ than on traditional RAID cards.

SupremeRAID™ does not require a buffering or caching mechanism to improve read/write performance as do traditional RAID cards. This feature causes SupremeRAID™ indicators to blink differently than traditional RAID cards.

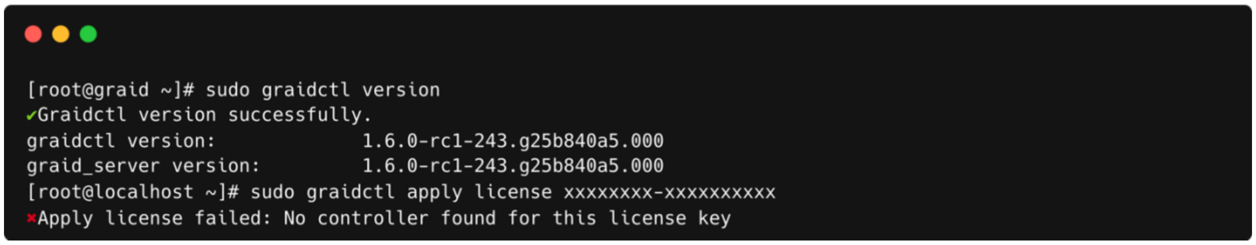
Received "The arch of the controller and graid software mismatched" Message When Applying License

To activate the SupremeRAID™ server with your license key, it's essential to install the correct driver version that matches your specific SupremeRAID™ model. If the incorrect version is installed, the following error message appears when you try to activate the SupremeRAID™ server with a license key: Apply license failed: The arch of the controller and graid software mismatched.

To ascertain which model you installed, use the command graidctl version. Issuing this command displays the model information at the end of the string.

001 -> SupremeRAID™ SR-1001
000 -> SupremeRAID™ SR-1000
010 -> SupremeRAID™ SR-1010

The following figure shows an example of the message, if you receive the error message, uninstall the incorrect driver, and then install the correct one.



```
[root@graid ~]# sudo graidctl version
✓Graidctl version successfully.
graidctl version:            1.6.0-rc1-243.g25b840a5.000
graid_server version:        1.6.0-rc1-243.g25b840a5.000
[root@localhost ~]# sudo graidctl apply license xxxxxxxx-xxxxxxxxxx
×Apply license failed: No controller found for this license key
```

Step 1 Stop SupremeRAID™ service. If you have already enabled the graphical management console, please ensure to disable it as well.

```
$ sudo systemctl stop graid
$ sudo systemctl stop graid-mgr.service
```

Step 2 Unload the kernel model of graid.

```
$ sudo rmmod graid_nvidia graid
```

Step 3 Uninstall the package using the command appropriate for your operating system:

- For Centos, Rocky Linux, AlmaLinux, RHEL, openSUSE, and SLES:

```
$ sudo rpm -e graid-sr
```

- For Ubuntu:

```
$ sudo dpkg -r graid-sr
```

Step 4 Confirm that the SupremeRAID™ module is unloaded. The output should be empty.

```
$ sudo lsmod | grep graid
```

Step 5 Confirm that the SupremeRAID™ package is uninstalled using the command appropriate for your operating system, the output should be empty.

- For Centos, Rocky Linux, AlmaLinux, RHEL, openSUSE, and SLES:

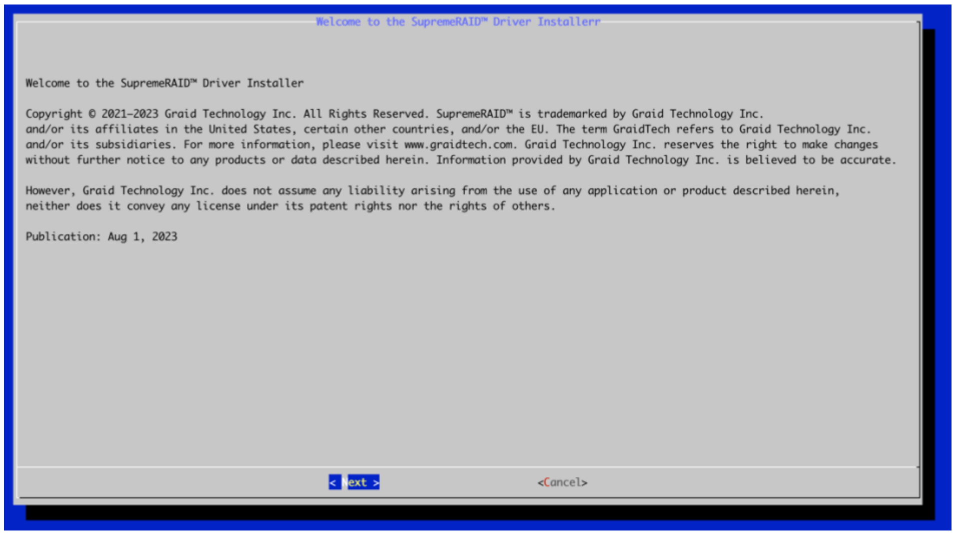
```
$ sudo rpm -qa | grep graid
```

- For Ubuntu:

```
$ sudo dpkg -l | grep graid
```

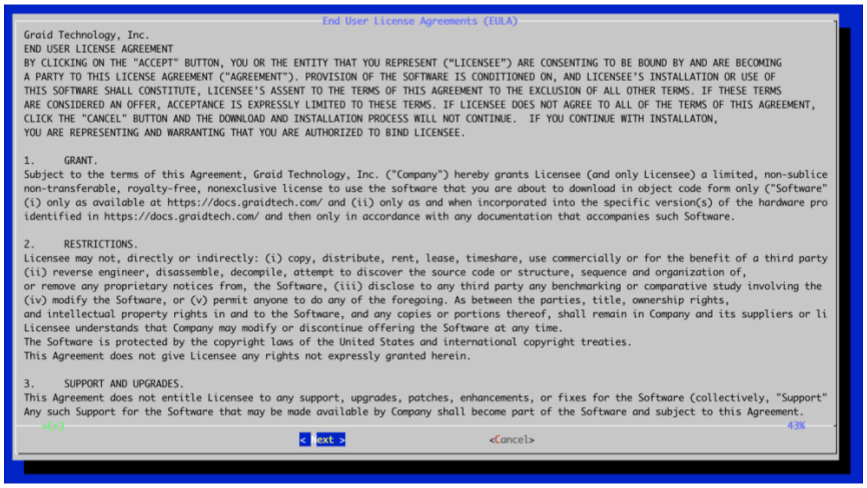
Step 6 Install the correct graid driver:

A At the Welcome page, select Next and click Enter to view the end-user license agreement.



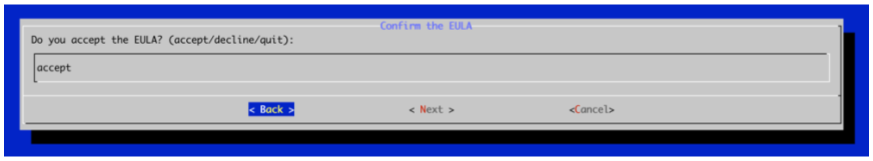
```
Welcome to the SupremeRAID™ Driver Installer

Copyright © 2021-2023 Graid Technology Inc. All Rights Reserved. SupremeRAID™ is trademarked by Graid Technology Inc.
and/or its affiliates in the United States, certain other countries, and/or the EU. The term GraidTech refers to Graid Technology Inc.
and/or its subsidiaries. For more information, please visit www.graidtech.com. Graid Technology Inc. reserves the right to make changes
without further notice to any products or data described herein. Information provided by Graid Technology Inc. is believed to be accurate.

However, Graid Technology Inc. does not assume any liability arising from the use of any application or product described herein,
neither does it convey any license under its patent rights nor the rights of others.

Publication: Aug 1, 2023
```

B In the end-user license agreement, use the spacebar to scroll through the content. When you complete your review, select Next and click Enter to proceed.



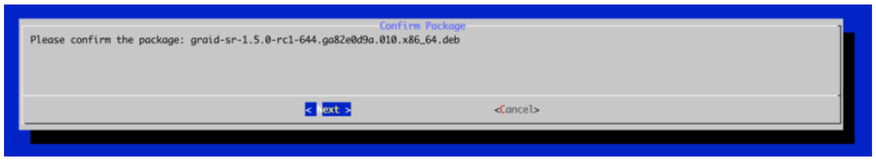
```
Graid Technology, Inc.
END USER LICENSE AGREEMENT
BY CLICKING ON THE "ACCEPT" BUTTON, YOU OR THE ENTITY THAT YOU REPRESENT ("LICENSEE") ARE CONSENTING TO BE BOUND BY AND ARE BECOMING
A PARTY TO THIS LICENSE AGREEMENT ("AGREEMENT"). PROVISION OF THE SOFTWARE IS CONDITIONED ON, AND LICENSEE'S INSTALLATION OR USE OF
THIS SOFTWARE SHALL CONSTITUTE, LICENSEE'S ASSENT TO THE TERMS OF THIS AGREEMENT TO THE EXCLUSION OF ALL OTHER TERMS. IF THESE TERMS
ARE CONSIDERED AN OFFER, ACCEPTANCE IS EXPRESSLY LIMITED TO THESE TERMS. IF LICENSEE DOES NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT,
CLICK THE "CANCEL" BUTTON AND THE DOWNLOAD AND INSTALLATION PROCESS WILL NOT CONTINUE.  IF YOU CONTINUE WITH INSTALLATON,
YOU ARE REPRESENTING AND WARRANTING THAT YOU ARE AUTHORIZED TO BIND LICENSEE.

1.     GRANT.
Subject to the terms of this Agreement, Graid Technology, Inc. ("Company") hereby grants Licensee (and only Licensee) a limited, non-sublice
non-transferable, royalty-free, nonexclusive license to use the software that you are about to download in object code form only ("Software"
(i) only as available at https://docs.graidtech.com/ and (ii) only as and when incorporated into the specific version(s) of the hardware pro
identified in https://docs.graidtech.com/ and then only in accordance with any documentation that accompanies such Software.

2.     RESTRICTIONS.
Licensee may not, directly or indirectly: (i) copy, distribute, rent, lease, timeshare, use commercially or for the benefit of a third party
(ii) reverse engineer, disassemble, decompile, attempt to discover the source code or structure, sequence and organization of,
or remove any proprietary notices from, the Software, (iii) disclose to any third party any benchmarking or comparative study involving the
(iv) modify the Software, or (v) permit anyone to do any of the foregoing. As between the parties, title, ownership rights,
and intellectual property rights in and to the Software, and any copies or portions thereof, shall remain in Company and its suppliers or li
Licensee understands that Company may modify or discontinue offering the Software at any time.
The Software is protected by the copyright laws of the United States and international copyright treaties.
This Agreement does not give Licensee any rights not expressly granted herein.

3.     SUPPORT AND UPGRADES.
This Agreement does not entitle Licensee to any support, upgrades, patches, enhancements, or fixes for the Software (collectively, "Support"
Any such Support for the Software that may be made available by Company shall become part of the Software and subject to this Agreement.
```

C Type accept, click tab, select Next, and click Enter to accept the license agreement.



```
Do you accept the EULA? (accept/decline/quit):
accept
```

D Check the package version and click NEXT.



```
Please confirm the package: graid-sr-1.5.0-rc1-644.ga82e0d9a.010.x86_64.deb
```

E To activate the software, apply the SupremeRAID™ license key.

```
$ sudo graidctl apply license [LICENSE_KEY]
```

SupremeRAID™ Service Fail to Start

The SupremeRAID™ service may fail to run if there is insufficient root disk space available. Ensure that you have adequate free space in the root partition for the graid service to operate correctly. Lack of sufficient disk space can cause the graid_service to fail during the enabling process.

SAFETY INFORMATION

English Version

CE Directives Declaration: NVIDIA Corporation hereby declares that this device complies with all material requirements and other relevant provisions of the 2014/30/EU and 2011/65/EU. A copy of the Declaration of Conformity may be obtained directly from NVIDIA GmbH(Bavaria Towers - Blue Tower, Einsteinstrasse 172, D-81677 Munich, Germany)

NVIDIA products are designed to operate safely when installed and used according to the product instructions and general safety practices. The guidelines included in this document explain the potential risks associated with equipment operation and provide important safety practices designed to minimize these risks. By carefully following the information contained in this document, you can protect yourself from hazards and create a safer environment.

This product is designed and tested to meet IEC 60950-1 and IEC 62368-1 Safety Standards for Information Technology Equipment. This also covers the national implementations of IEC 70950-1/62368-1 based safety standards around the world e.q. UL 62368-1. These standards reduce the risk of injury from the following hazards:

- Electric shock: Hazardous voltage levels contained in parts of the product
- Fire: Overload, temperature, material flammability
- Energy: Circuits with high energy levels (240-volt amperes) or potential as burn hazards.
- Heat: Accessible parts of the product at high temperatures.
- Chemical: Chemical fumes and vapors
- Radiation: Noise, ionizing, laser, ultrasonic waves

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

This product, as well as its related consumables and spares, complies with the reduction in hazardous substances provisions of the "India E-waste (Management and Handling) Rule 2016". It does not contain lead, mercury, hexavalent chromium, polybrominated biphenyls or polybrominated diphenyl ethers in concentrations exceeding 0.1 weight % and 0.01 weight % for cadmium, except for where allowed pursuant to the exemptions set in Schedule 2 of the Rule.

Retain and follow all product safety and operating instructions.

Always refer to the documentation supplied with your equipment. Observe all warnings on the product and in the operating instructions found on the product's User Guide.

This is a recycling symbol indicating that the product/battery cannot be disposed of in the trash and must be recycled according to the regulations and/or ordinances of the local community.

Hot surface warning. Contact may cause burns. Allow to cool before servicing.

Chinese Version

NVIDIA 产品在设计时充分考虑到操作安全性，可根据产品说明和常规安全做法进行安全安装和使用。本文档中包含的准则解释了设备操作所涉及的风险，并提供了最大限度降低这些风险的重要安全做法。请详细阅读本文档中的信息并按要求操作，这样可保护您免遭受为显并创建一个更加安全的环境。

本产品按照信息技术设备安全标准 IEC 60950-1 **和** IEC 62368-1 进行设计，并且经测试表明符合这些设备。此处所述标准也包括全球各国/**地区**实施的基于 IEC 60950-1/62368-1 **的安全**标准，例如 UL 62328-1。这些标准**降低了因以下危**险而受伤的风险：

- 电击：部分产品中包含的危险电压水平起火：超载、高温、可燃性材料
- **机械**：锋利的边缘、活动部件、不稳定性
- 电源：高电压电路（240 **伏安）或潜在的**烧伤风险
- 高温：产品的可触及部分存在高温化学：化学烟雾和蒸气
- 辐射：噪音、电离、激光、超声波

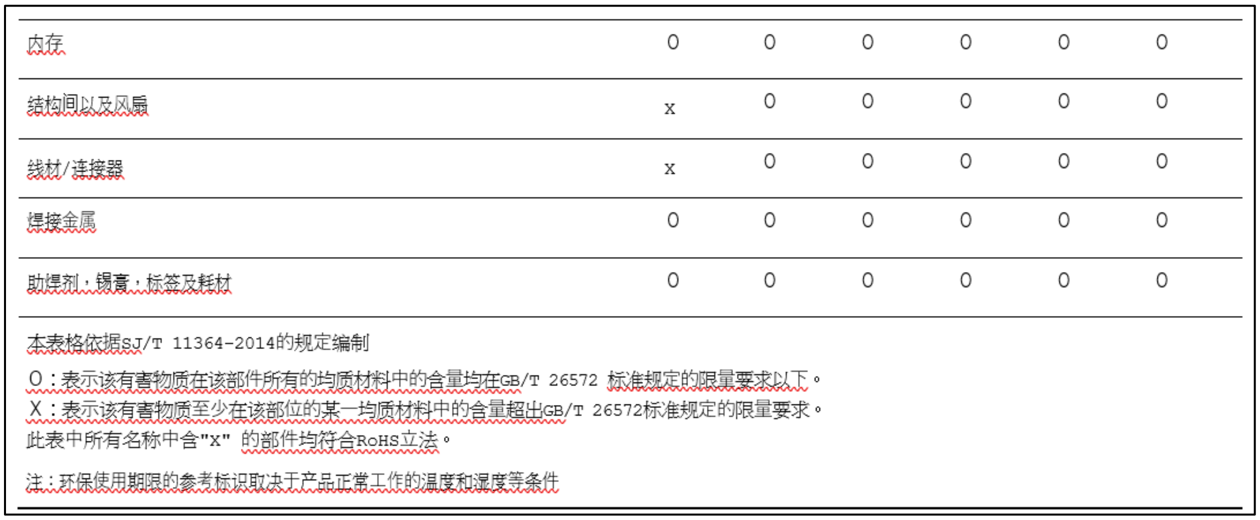
请牢记并遵守所有产品安全和操作说明。请务必参考您的设备随附的说明文档。请注意产品上以及产品用户指南的操作说明中列

示的所有警告。

这是一个通用的回收标志，表示产品/电池不能以丢弃的方式处置，必须按造本**地社区的法**规和/**或条例回收。**

警告！表面高溫。接觸可能導致灼傷。請再冷卻後再使用。

产品中有害物质的名称及含量根据中国 电器电子产品有害物质限制使用管理办 法）



内存	0	0	0	0	0	0
结构间以及风扇	x	0	0	0	0	0
线材/连接器	x	0	0	0	0	0
焊接金属	0	0	0	0	0	0
助焊剂，锡膏，标签及耗材	0	0	0	0	0	0

本表格依据SJ/T 11364-2014的规定编制

O：表示该有害物质在该部件所有的均质材料中的含量均在GB/T 26572 标准规定的限量要求以下。
X：表示该有害物质至少在该部位的某一均质材料中的含量超出GB/T 26572标准规定的限量要求。
此表中所有名称中含"X" 的部件均符合RoHS立法。

注：环保使用期限的参考标识取决于产品正常工作的温度和湿度等条件

Chinese Version (TC)

在遵照產品說明與一般安全做法進行安裝與使用產品的情況下，NVIDIA 產品可安全地操作。本文件所列的準則說明與設備操作相關的潛在風險，同時也提供將這些風險降到最低的重要安全做法。謹慎遵守本文件中的資訊，您就可以避免危險並創造更安全的環境。

此產品係根據 Safety Standards for Information Technology Equipment(**資訊技術設備安全標準**) IEC 60950-1 **和** IEC 62368-1 **進**

行設計與測試。同時也涵蓋全世界國家以 IEC 60950-1/62368-1 **為根據的安全標準， 例如** UL 62368-1。**這些標準可降低下列危險造成的傷害的風險**：

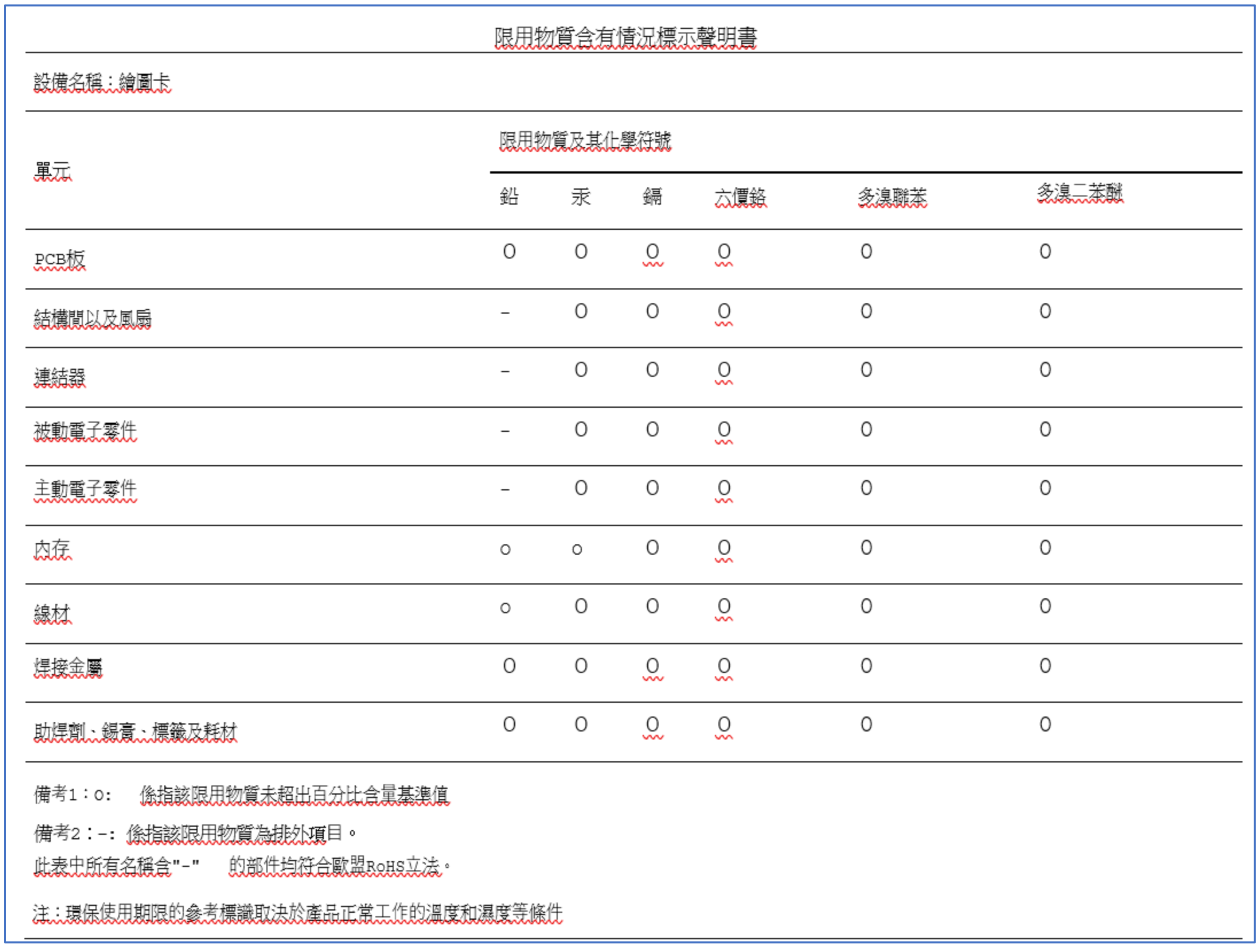
- **觸電危險**：**本**產品部分零件的電壓等級具危險性
- **火災危險**：**超載、溫度、材料可燃性**
- **機械危險**：**尖**銳邊緣、移動零件、不穩定性

- **電燒**力危險：電路電壓高（240 **電壓）或具有潛在起火燃燒熱能危險**：產品表面可能達到高溫，注意燙傷危機
- **化學危險**：**化學異味氣體與蒸氣**
- **輻射危險**：**噪**音、游離輻射、雷射、超音波

請保留並遵守所有產品安全與操作說明的相關規定。請務必參閱設備隨附的文件。請遵守產品上，和產品使用者只能中操作說明裡的警告規定。

此國際回收標誌表示此產品/**電池不能棄置於垃圾桶中,**
必須根據當地社區的規範和/或法令回收。

表面高溫警告。接觸時可能燙傷。使用前請先降溫。



限用物質含有情況標示聲明書

設備名稱：繪圖卡

單元	限用物質及其化學符號					
	鉛	汞	鎘	六價鉻	多溴聯苯	多溴二苯醚
PCB板	O	O	O	O	O	O
結構間以及風扇	–	O	O	O	O	O
連結器	–	O	O	O	O	O
被動電子零件	–	O	O	O	O	O
主動電子零件	–	O	O	O	O	O
內存	o	o	O	O	O	O
線材	o	O	O	O	O	O
焊接金屬	O	O	O	O	O	O
助焊劑、錫膏、標籤及耗材	O	O	O	O	O	O

備考1：o:　係指該限用物質未超出百分比含量基準值

備考2：-:　係指該限用物質為排外項目。

此表中所有名稱含"-"　的部件均符合歐盟RoHS立法。

注：環保使用期限的參考標識取決於產品正常工作的溫度和濕度等條件

ATTACHMENTS

Events for SupremeRAID™

Category	Severity	Description
Physical Drive	Warning	Physical Drive <PD_ID> state has transitioned from <STATE_OLD> to unconfigured bad.
	Critical	Physical Drive <PD_ID> state has transitioned from <OLD_STATE> to failed.
	Warning	Physical Drive <PD_ID> state has transitioned from <OLD_STATE> to offline.
	Critical	Physical Drive <PD_ID> state has transitioned from <OLD_STATE> to missing.
	Info	Physical Drive <PD_ID> state has transitioned from <OLD_STATE> to online.
	Info	Physical Drive <PD_ID> state has transitioned from <OLD_STATE> to rebuild.
	Info	Physical Drive <PD_ID> state has transitioned from <OLD_STATE> to unconfigured good.
	Info	Physical Drive <PD_ID> has been successfully created.
	Info	Physical Drive <PD_ID> has been deleted.
	Info	Physical Drive <PD_ID> has been hot-plugged.
	Warning	Physical Drive <PD_ID> has been hot-removed.
	Warning	The temperature of Physical Drive <PD_ID> is currently <CURRENT_TEMP> degrees, which exceeds the Warning threshold of <THRESHOLD_TEMP> degrees. Critical Warning error code: ERROR_CODE.
	Critical	The temperature of Physical Drive <PD_ID> is currently <CURRENT_TEMP> degrees, which exceeds the Critical threshold of <THRESHOLD_TEMP> degrees. Critical Warning error code: ERROR_CODE.
	Critical	The available spare capacity <AVAIL_SPARE> of Physical Drive <PD_ID> has fallen below the threshold <SPARE_THRESHOLD>. Critical Warning error code: <ERROR_CODE>.
	Critical	The NVM subsystem reliability of Physical Drive <PD_ID> has been degraded due to significant media related errors or any internal error that degrades NVM subsystem reliability. Critical Warning error code: <ERROR_CODE>.
	Critical	All of the media of Physical Drive <PD_ID> has been placed in read only mode. Critical Warning error code: <ERROR_CODE>.

	Critical	The volatile memory backup device of Physical Drive <PD_ID> has failed. Critical Warning error code: <ERROR_CODE>.
	Critical	The Persistent Memory Region of Physical Drive <PD_ID> has become read-only or unreliable. Critical Warning error code: <ERROR_CODE>.
	Warning	Physical Drive <PD_ID> is currently experiencing a wearout level of WEAROUT, surpassing the Warning threshold of <THRESHOLD_WEAROUT>.
	Critical	Physical Drive <PD_ID> is currently experiencing a wearout level of WEAROUT, surpassing the Critical threshold of <THRESHOLD_WEAROUT>.
Drive Group	Fatal	Drive Group <DG_ID> state has transitioned from <OLD_STATE> to failed.
	Critical	Drive Group <DG_ID> state has transitioned from <OLD_STATE> to offline.
	Critical	Drive Group <DG_ID> state has transitioned from <OLD_STATE> to degraded.
	Warning	Drive Group <DG_ID> state has transitioned from <OLD_STATE> to rescue.
	Warning	Drive Group <DG_ID> state has transitioned from <OLD_STATE> to partially degraded.
	Info	Drive Group <DG_ID> state has transitioned from <OLD_STATE> to optimal.
	Info	Drive Group <DG_ID> state has transitioned from <OLD_STATE> to recovery.
	Info	Drive Group <DG_ID> state has transitioned from <OLD_STATE> to init.
	Info	Drive Group <DG_ID> state has transitioned from <OLD_STATE> to resync.
	Info	Drive Group <DG_ID> has been successfully created.
	Info	Drive Group <DG_ID> has been deleted.
	Info	Consistency Check for Drive Group <DG_ID> has been manually aborted.
	Info	Consistency Check for Drive Group <DG_ID> has been aborted due to the deletion of the Drive Group.
	Info	Consistency Check for Drive Group <DG_ID> was aborted due to the Drive Group migrating from Controller <CX_OLD> to <CX_NEW>.
	Info	Consistency Check for Drive Group <DG_ID> has been aborted due to the Drive Group's state transitioning to <DG_STATE>.
	Info	Manual Consistency Check for Drive Group <DG_ID> has been completed.
	Info	Scheduled Consistency Check for Drive Group <DG_ID> has completed.
	Info	Manual Consistency Check for Drive Group <DG_ID> has started.

	Info	Scheduled Consistency Check for Drive Group <DG_ID> has started.
	Info	Inconsistency in Drive Group <DG_ID> has been fixed at: Drive Group block range: <DG_INTERS>.
	Critical	Inconsistency detected in Drive Group <DG_ID> at: Drive Group block range: <DG_INTERS>.
	Critical	Consistency Check for Drive Group <DG_ID> has been aborted due to the 'stop_on_error' policy.
	Critical	Consistency Check for Drive Group <DG_ID> has been aborted due to numerous inconsistencies found and fixed.
	Info	Journal Replay for Drive Group <DG_ID> has started.
	Info	Journal Replay for Drive Group <DG_ID> has been completed. Entry replayed <REPLAYNR>.
	Critical	Journal Replay for Drive Group <DG_ID> has been waiting Physical Drive <PD_ID> to be active.
	Critical	Journal Replay for Drive Group <DG_ID> has been aborted due to inconsistency detected on journal.
Virtual Drive	Info	Inconsistency for Virtual Drive <VD_ID> within Drive Group <DG_ID> has been fixed at: Virtual Drive block range: <VD_OFFSETS>.
	Critical	Inconsistency found in Virtual Drive VD_ID of Drive Group <DG_ID> at: Virtual Drive block range: <VD_OFFSETS>.
	Info	Virtual Drive VD_ID for Drive Group <DG_ID> has been created successfully.
	Info	Virtual Drive VD_ID for Drive Group <DG_ID> has been deleted.
	Info	Stripe cache for Virtual Drive <VD_ID> on Drive Group <DG_ID> has been deleted.
	Info	Stripe cache for Virtual Drive <VD_ID> on Drive Group <DG_ID> has been created successfully.
Controller	Warning	The temperature of Controller <CX_ID> is currently <CURRENT_TEMP> degrees, which exceeds the GPU threshold of <THRESHOLD_TEMP> degrees.
	Warning	The temperature of Controller <CX_ID> is currently <CURRENT_TEMP> degrees, which exceeds the GPU memory threshold of <THRESHOLD_TEMP> degrees.
	Warning	The temperature of Controller <CX_ID> is currently <CURRENT_TEMP> degrees, it will cause controller slowdown.
	Critical	The temperature of Controller <CX_ID> is currently <CURRENT_TEMP> degrees, it will cause controller shutdown.