Implementing NVMe Drives on Lenovo Servers

Last Update: June 2018

Introduces the use of Non-Volatile Memory Express (NVMe) drives

Explains how to use NVMe drives with Microsoft Windows, Linux and VMware ESXi

Describes how to create RAID volumes using operating system tools

Describes how to recover a RAID array when an NVMe drive has failed

Ilya Solovyev

David Watts

LENOVO PRESS

Abstract

This paper describes the use of Non-Volatile Memory Express (NVMe) drives in Lenovo® ThinkSystem™, System x®, ThinkServer® and Flex System™ servers. We introduce the components and explain the key characteristics of the drives. We also explain how to use the tools supplied by key supported operating systems (Windows, Linux and VMware) to form RAID volumes using the NVMe drives. Finally, we describe how to properly recovery a RAID array in the event of a drive failure.

This paper is for IT specialists wanting to learn how to properly install, use and manage NVMe drives in supported Lenovo servers.

At Lenovo Press, we bring together experts to produce technical publications around topics of importance to you, providing information and best practices for using Lenovo products and solutions to solve IT challenges.

See a list of our most recent publications at the Lenovo Press web site:

http://lenovopress.com

Do you have the latest version? We update our papers from time to time, so check whether you have the latest version of this document by clicking the **Check for Updates** button on the front page of the PDF. Pressing this button will take you to a web page that will tell you if you are reading the latest version of the document and give you a link to the latest if needed. While you're there, you can also sign up to get notified via email whenever we make an update.

Contents

Introduction to NVMe

Non-Volatile Memory Express (NVMe) is new PCIe 3.0 high performance solid-state drive (SSD) technology that provides high I/O throughput and low latency. NVMe interfaces remove SAS/SATA bottlenecks and enable all of the capabilities of contemporary NAND flash memory.

Figure 1 shows the PCIe NVMe SSDs of three different vendors: Toshiba, Intel and Samsung.

Figure 1 NVMe PCIe SSDs: (l-r): Toshiba, Intel and Samsung

Each NVMe SSD has direct PCIe 3.0 x4 connection, which provides at least 2x more bandwidth and 2x lower latency than SATA/SAS-based SSD solutions. NVMe drives are also optimized for heavy multi-threaded workloads by using internal parallelism and many other improvements, such as enlarged I/O queues.

NVMe technology has the following key characteristics:

- PCIe 3.0 connection. There is a PCIe 3.0 x4 connection for each NVMe drive with up to 4 GBps overall throughput.
- Low I/O latency. For example, the average read/write latency for the Intel P4800X Optane drives is 10 µs.
- High sequential throughput. For example, Toshiba drives offer up to 3100 MBps sequential read speed with 128 KB blocks, and up to 2350 MBps sequential write speed with 128 KB blocks per drive.
- High I/O operations per second. For example the Toshiba drives support up to 666,000 IOPS of random read with 4 KB blocks, and up to 105,000 IOPS of random writes with 4 KB blocks.
- A total of 65,536 I/O queues supported and 65,536 commands per queue supported, which provides great performance on heavily multithreaded workloads with combined sequential and random access.
- High endurance: The Intel P4800X Optane drives, for example, include features which combine NAND silicon enhancements and SSD NAND management techniques to extend SSD write endurance up to 30 drive writes per day (DWPD) for 5 years.
- Support for software RAID under operating system management.
- Hot add and hot remove features are available on specific servers with supported operating systems.

Hot-swap support: Not all servers that support NVMe drives support the hot-swap capability of those drives. See Table 1.

- Most operating systems have native support of NVMe drives or provide support through software drivers, such as
  - RHEL 6.5 and later
  - SLES 11 SP3 and later
  - Windows Server 2008 R2 and later
  - VMware ESXi 5.5 and later
- NVMe drives can be used as boot drives.
- NVMe drives are supported in a variety of Lenovo servers, as listed in Table 1. The table also lists whether the servers support hot-add or hot-replace of NVMe drives.

Table 1 NVMe support

Lenovo server	NVMe support	Hot-add/replace support[a]
Rack servers		
ThinkSystem SR950 Server	Yes	Yes
ThinkSystem SR860 Server	Yes	Yes
ThinkSystem SR850 Server	Yes	Yes
ThinkSystem SR650 Server	Yes	Yes
ThinkSystem SR630 Server	Yes	Yes
ThinkSystem SR590 Server	Yes	Yes
ThinkSystem SR570 Server	Yes	Yes
System x3850 X6	Yes	Yes
System x3950 X6	Yes	Yes
System x3650 M5	Yes	Yes
System x3550 M5	Yes	No
ThinkServer RD650	Yes	No
ThinkServer RD550	Yes	No
Tower servers		
ThinkSystem ST550 Server	Yes	Yes
Density-optimized servers		
ThinkSystem SD530 Server	Yes	Yes
ThinkSystem SD650 Server	Yes	No
Blade servers		
ThinkSystem SN550 Server	Yes	Yes
ThinkSystem SN850 Server	Yes	Yes
Flex System x240 M5	Yes	Yes

a. Informed hot removal and hot insertion. Surprise removal not supported.

NVMe drives attach to a drive backplane, similar to SAS or SATA drives, however, unlike SAS or SATA drives, the NVMe backplane connects directly to the PCIe bus rather than through a RAID controller or SAS HBA. Depending on the server, the PCIe connection is either a port on the system board or a PCIe extender adapter which is installed in a PCIe slot. The lack of a protocol conversion from PCIe to SAS/SATA is why NVMe SSD drives have better performance than SAS or SATA SSDs.

For example, in the x3850 X6, an extender adapter is used to connect the backplanes to the PCIe bus. The extender adapter is shown in Figure 2. Each NVMe PCIe extender supports one or two NVMe drives. You can install up to two NVMe PCI extenders in each Storage Book of the x3850 X6, which means you have up to four NVMe PCIe drives in one Storage Book.

Figure 2 NVMe PCIe SSD Extender Adapter

The extender adapter is a PCIe 3.0 x8 device which is why the adapter only supports two NVMe drives (each of which is a PCIe 3.0 x4 device).

Setting up NVMe drives in the operating system

In this section, we describe the planning and use of these drives. This section includes the following topics:

PCIe slot numbering

NVMe drives and NVMe extender adapters are seen by the operating system PCIe devices. As a result, it is important to know the slot numbering and drive bay numbers so that you can determine exactly which drive is which when working with NVMe drives in the OS.

For example, NVMe drives installed in an x3850 X6 or x3850 X6 are as follows:

- The Extender Adapters are installed in the following slots of the Storage Books:
  - For the x3850 X6, the slots are PCIe slots 11 and 12.
  - For the x3950 X6, the slots are PCIe slots 11, 12, 43, and 44.
- Table 2 shows the connectivity and slot installation ordering of the PCIe Extender Adapter, backplane, and NVMe drives.

Table 2 x3850 X6 NVMe slot and PCIe installation ordering

NVMe PCIe Extender Adapter	Extender Adapter location	PCIe signal cable connections	NVMe SSD drives population order, location in the Storage book, PCIe slot used by drive
NVMe PCIe extender adapter 1	I/O book slot 11	Adapter port 0 to backplane port 0	Drive 1, bay 7, PCIe slot 19
		Adapter port 1 to backplane port 1	Drive 2, bay 6, PCIe slot 18
NVMe PCIe extender adapter 2	I/O book slot 12	Adapter port 0 to backplane port 2	Drive 3, bay 5, PCIe slot 17
		Adapter port 1 to backplane port 3	Drive 4, bay 4, PCIe slot 16

Figure 3 shows 2.5-inch NVMe SSD drives location in the Storage book, bays and PCIe slots used by them:

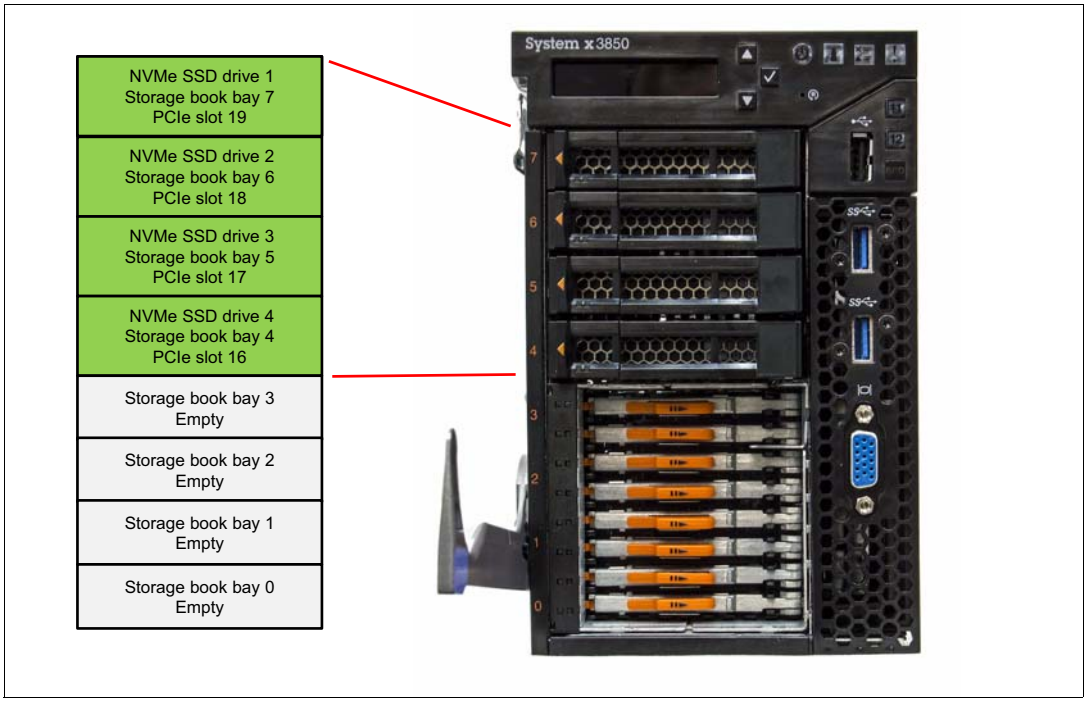


Figure 3 2.5" NVMe SSD drives location in the Storage Book

Figure 4 shows the ports of the Extender Adapter.

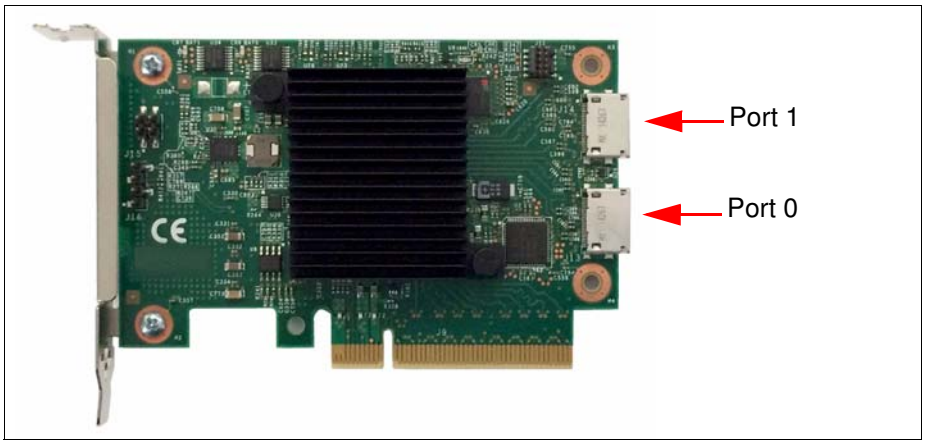


Figure 4 NVMe Extender Adapter port numbering

Figure 5 shows the ports of the NVMe SSD backplane.

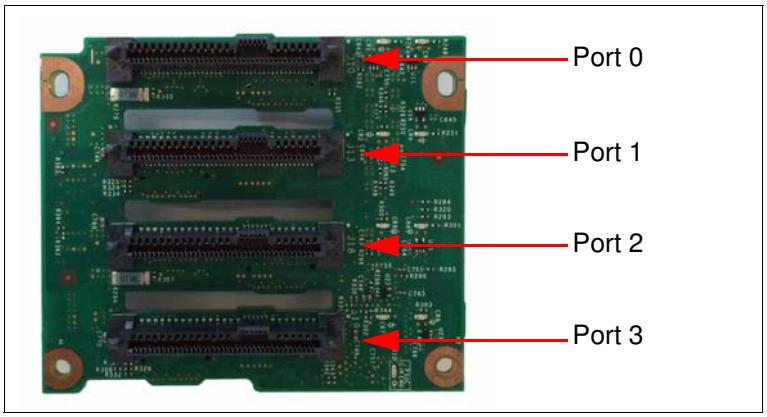
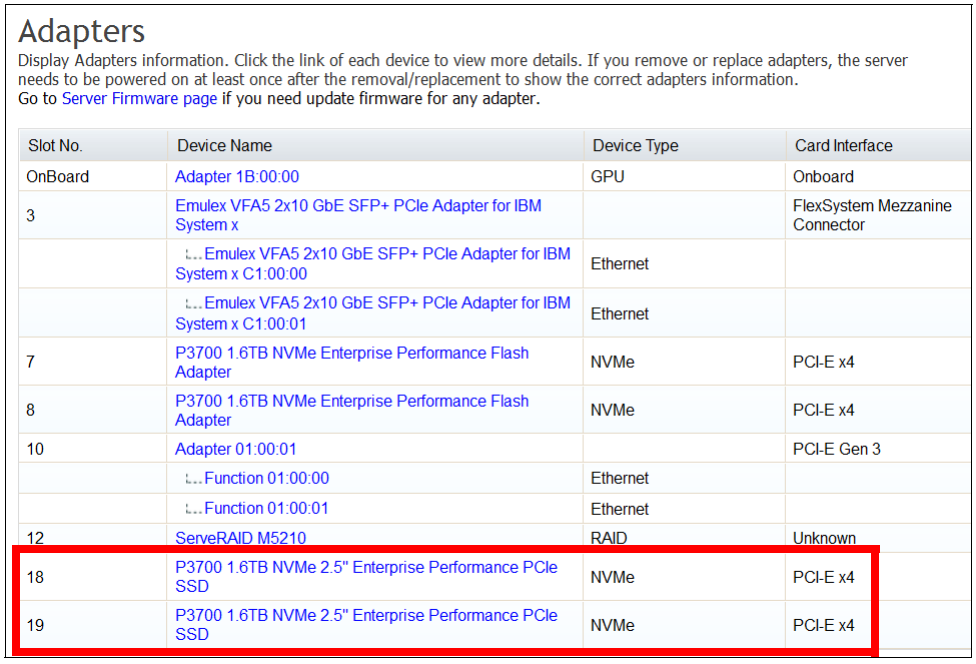


Figure 5 NVMe backplane port numbering

The operating system and UEFI report the NVMe drives attached to the 4x2.5-inch NVMe PCIe backplane as PCI devices, connected to PCIe slots 16-19. You can check connected NVMe SSD drives from IMM web-interface at **Server Management** → **Adapters** page, as shown in Figure 6:



Adapters

Display Adapters information. Click the link of each device to view more details. If you remove or replace adapters, the server needs to be powered on at least once after the removal/replacement to show the correct adapters information.
Go to Server Firmware page if you need update firmware for any adapter.

Slot No.	Device Name	Device Type	Card Interface
OnBoard	Adapter 1B:00:00	GPU	Onboard
3	Emulex VFA5 2x10 GbE SFP+ PCIe Adapter for IBM System x		FlexSystem Mezzanine Connector
	Emulex VFA5 2x10 GbE SFP+ PCIe Adapter for IBM System x C1:00:00	Ethernet	
	Emulex VFA5 2x10 GbE SFP+ PCIe Adapter for IBM System x C1:00:01	Ethernet	
7	P3700 1.6TB NVMe Enterprise Performance Flash Adapter	NVMe	PCI-E x4
8	P3700 1.6TB NVMe Enterprise Performance Flash Adapter	NVMe	PCI-E x4
10	Adapter 01:00:01		PCI-E Gen 3
	Function 01:00:00	Ethernet	
	Function 01:00:01	Ethernet	
12	ServeRAID M5210	RAID	Unknown
18	P3700 1.6TB NVMe 2.5" Enterprise Performance PCIe SSD	NVMe	PCI-E x4
19	P3700 1.6TB NVMe 2.5" Enterprise Performance PCIe SSD	NVMe	PCI-E x4

Figure 6 PCIe slots used by NVMe SSD drives

Know your PCIe slot numbers: It's important to know PCIe slots numbers used by NVMe drives: during the software RAID maintenance and NVMe SSD drives replacement these PCIe slot numbers allows you to distinguish the appropriate drive in the set of similar NVMe drives.

Using NVMe drives with Linux

NVMe drives are supported on the following Linux distributions:

- Red Hat Enterprise Linux 6.5 and later
- Red Hat Enterprise Linux 7.0 and later
- SUSE Linux Enterprise Server 11 SP3 and later
- SUSE Linux Enterprise Server 12 and later

Other Linux distributions might have NVMe support, depending on the kernel version.

The RHEL and SLES distributions have NVMe kernel modules; therefore, no other drivers are required to use NVMe drives. NVMe drives are represented in the OS as block devices with device names, such as `/dev/nvmeXn1`, where `X` is a number that is associated with each NVMe drive that is installed in the server. For example, for four NVMe drives and one NVMe adapter installed, the device names are `nvme0n1, nvme1n1, ..., nvme4n1,` which could be located in `/dev` directory, as shown in Figure 7 on page 9.

Note: In this document, we used the x3850 X6 server as our test system.

```
[root@localhost ~]# ls -1 /dev/nvme*
/dev/nvme0
/dev/nvme0n1
/dev/nvme0n1p1
/dev/nvme1
/dev/nvme1n1
/dev/nvme1n1p1
/dev/nvme2
/dev/nvme2n1
/dev/nvme2n1p1
/dev/nvme3
/dev/nvme3n1
/dev/nvme4
/dev/nvme4n1
/dev/nvme4n1p1
/dev/nvme4n1p2
/dev/nvme4n1p3
/dev/nvme4n1p4
/dev/nvme4n1p5
/dev/nvme4n1p6
/dev/nvme4n1p7
/dev/nvme4n1p8
[root@localhost ~]#
```

Figure 7 NVMe drives device names in /dev directory

Devices `/dev/nvme1 ... /dev/nvme4` represent associated controllers.

Each drive may have several partitions. For every partition associated block device is created in `/dev` folder. In previous example the drive `/dev/nvme4n1` has eight partitions and the following devices are created accordingly: `/dev/nvme4n1p1 ... /dev/nvme4n1p8`.

Figure 8 shows other Linux commands that can show the NVMe drives.

```
[root@localhost ~]# lspci | grep -i non-vol
09:00.0 Non-Volatile memory controller: Intel Corporation PCIe Data Center SSD (rev 01)
0e:00.0 Non-Volatile memory controller: Samsung Electronics Co Ltd Device a804
49:00.0 Non-Volatile memory controller: Samsung Electronics Co Ltd Device a804
4e:00.0 Non-Volatile memory controller: Toshiba America Info Systems Device 010e (rev
01)
95:00.0 Non-Volatile memory controller: Intel Corporation PCIe Data Center SSD (rev 01)
[root@localhost ~]# lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
nvme2n1     259:0    0   1.8T  0 disk
??nvme2n1p1 259:1    0   128M  0 part
nvme3n1     259:2    0 894.3G  0 disk
nvme4n1     259:3    0   1.5T  0 disk
??nvme4n1p1 259:4    0   200M  0 part
??nvme4n1p2 259:5    0   128M  0 part
??nvme4n1p3 259:6    0  97.7G  0 part
??nvme4n1p4 259:7    0    50G  0 part
??nvme4n1p5 259:8    0   502M  0 part
??nvme4n1p6 259:9    0   300M  0 part /boot/efi
??nvme4n1p7 259:10   0  19.5G  0 part /
??nvme4n1p8 259:11   0 476.9G  0 part
nvme0n1     259:12   0   1.8T  0 disk
??nvme0n1p1 259:13   0 476.9G  0 part
nvme1n1     259:14   0   1.8T  0 disk
??nvme1n1p1 259:15   0   128M  0 part
sda           8:0    1  14.5G  0 disk
??sda1        8:1    1     4M  0 part
??sda5        8:5    1   250M  0 part
??sda6        8:6    1   250M  0 part
??sda7        8:7    1   110M  0 part
??sda8        8:8    1   286M  0 part
??sda9        8:9    1   2.5G  0 part
sr0          11:0    1  1024M  0 rom
```

Figure 8 lspci and lsblk output

As shown in Figure 8, `lspci` and `lsblk` commands show that five NVMe controllers are connected to PCIe bus and five block devices `nvme0n1 ... nvme4n1` are available in the operating system. You can also run simple performance tests by using the `hdparm` utility, as shown in Figure 9.

```
[root@localhost ~]# hdparm -tT --direct /dev/nvme0n1

/dev/nvme0n1:
 Timing O_DIRECT cached reads:   4594 MB in  2.00 seconds = 2298.38 MB/sec
 Timing O_DIRECT disk reads: 8314 MB in  3.00 seconds = 2770.65 MB/sec
```

Figure 9 Performance test by using hdparm utility

As shown in Figure 9, direct read speed from one NVMe drive is 2.7 GBps and cached read speed is almost 2.3 GBps.

You can work with NVMe drives as with other block devices, such as SATA or SAS drives. You can use `fdisk` or `parted` utilities to manage disk partitions, create any supported file systems by using standard Linux commands, and mount these file systems.

For example, you can create partition with `parted` utility, as shown in Figure 10.

```
[root@localhost ~]# parted /dev/nvme3n1
GNU Parted 2.1
Using /dev/nvme3n1
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: Unknown (unknown)
Disk /dev/nvme3n1: 960GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start  End  Size  File system  Name  Flags

(parted) mkpart primary ext4 1M 960GB
(parted) print
Model: Unknown (unknown)
Disk /dev/nvme3n1: 960GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End    Size   File system  Name     Flags
 1      1049kB  960GB  960GB               primary

(parted) quit
```

Figure 10 Partition creation with parted utility

When you create partition on an NVMe drive, a new block device appears in the `/dev/` directory. For example, for `/dev/nvme3n1` drive, `/dev/nvme3n1p1` is created.

After that you can create the ext4 file system on that partition, as shown in Figure 11.

```
[root@localhost ~]# mkfs.ext4 /dev/nvme3n1p1
mke2fs 1.41.12 (17-May-2010)
Discarding device blocks: done
warning: 512 blocks unused.

Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
58720032 inodes, 234422272 blocks
11721139 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
7154 block groups
32768 blocks per group, 32768 fragments per group
8208 inodes per group
Superblock backups stored on blocks:
   32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
   4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
   102400000, 214990848

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 36 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

Figure 11 ext4 file system creation

Then, you can mount the new ext4 file system to the file system tree, as shown in Figure 12.

```
[root@localhost ~]# mkdir /media/nvme3
[root@localhost ~]# mount /dev/nvme3n1p1 /media/nvme3/
[root@localhost ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme4n1p7   20G  5.7G   13G  32% /
tmpfs           253G     0  253G   0% /dev/shm
/dev/nvme4n1p6  300M   27M  274M   9% /boot/efi
/dev/nvme3n1p1  881G   72M  836G   1% /media/nvme3
```

Figure 12 File system mounting

You can manage other NVMe drives in the same way by using other Linux features, such as Logical Volume Manager (LVM) and software RAID, if needed.

In previous example you also can see, that another NVMe device is used as bootable disk for OS: the disk partition `/dev/nvme4n1p7` contains root file system, `/dev/nvme4n1p6` is used for `/boot/efi`.

Using NVMe drives with Microsoft Windows Server

NVMe drives are supported on following operation systems:

- Windows Server 2008 R2
- Microsoft Windows Server 2012
- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2016

Microsoft Windows Server 2012 R2 and Windows Server 2016 have native NVMe driver support and no other drivers are required to start use NVMe drivers. Other Windows version might require drivers.

Note: In this document, we used the x3850 X6 server as our test system.

Complete the following steps to check that NVMe drives are recognized by Windows:

1. Open Device Manager and the expand **Disk drives** section. All installed NVMe drives should present. As examples, two different hardware configurations are shown in Figure 13.
   - The configuration on the left contains two Intel P3700 drives
   - The configuration on the right contains an Intel P3700 drive and P3700 adapter, as well as two Samsung NVMe drives and one Toshiba NVMe drive.

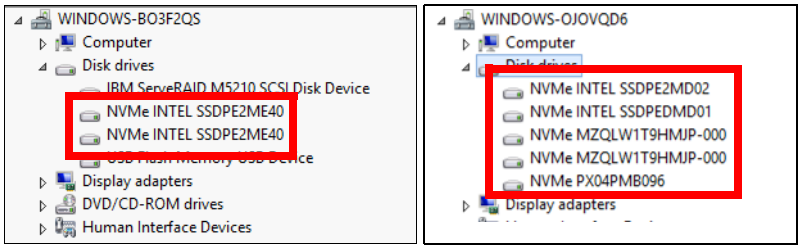


Figure 13 NVMe devices as shown in Device Manager

2. Open the Disk Management tool, you should see all installed NVMe drives. For our example, both installed NVMe drives are presented as Disk 1 and Disk 2, as shown in Figure 14 on page 14.

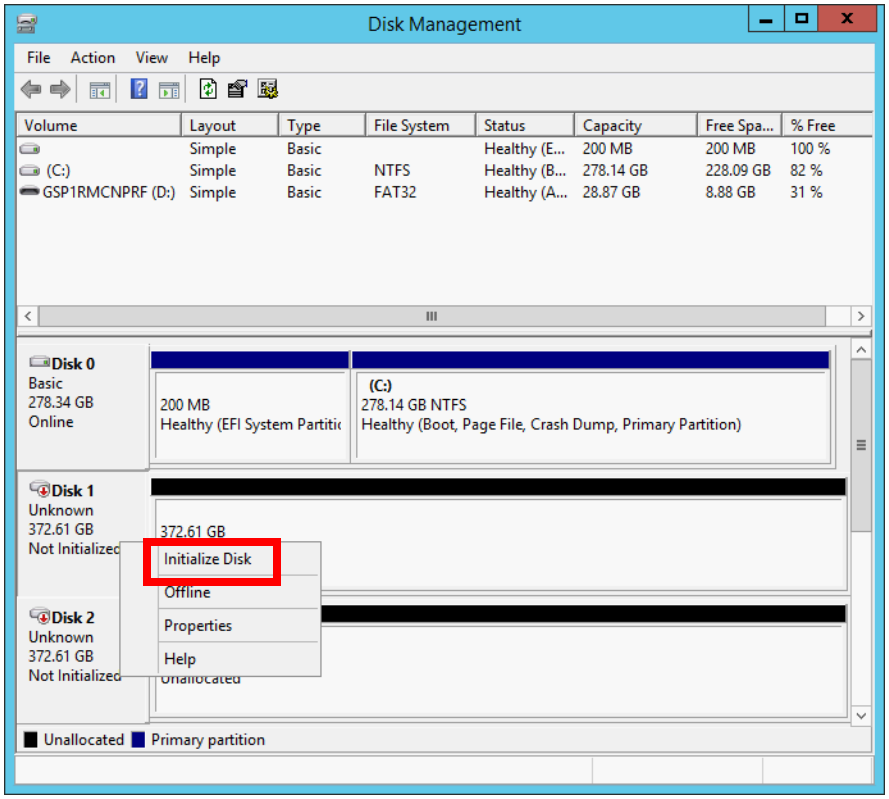


Figure 14 Disk Management tool and NVMe drives

3. Both NVMe drives must be online and initialized. To initialize the drives, right-click the appropriate disk (Disk 1 or Disk 2 as shown in Figure 14) and select **Initialize Disk**. The Initialize Disk window opens, as shown in Figure 15.

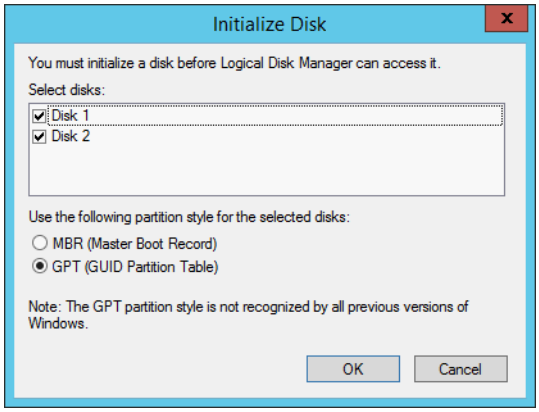


Figure 15 Disk initialization

4. After the disks are initialized, you can create volumes. Right-click the NVMe drive and select the required volume to create, as shown in Figure 16 on page 15.

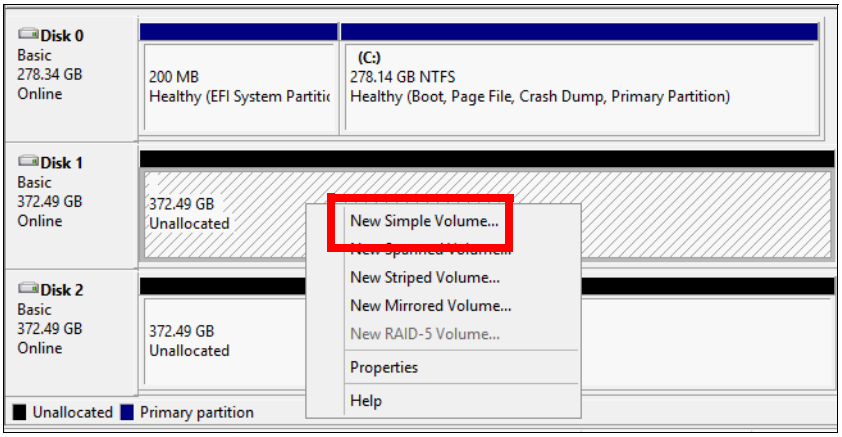


Figure 16 Creating a volume

5. For example, choose **New Simple Volume**. The volume creation wizard opens. Click **Next** and specify a volume size, as shown in Figure 17.

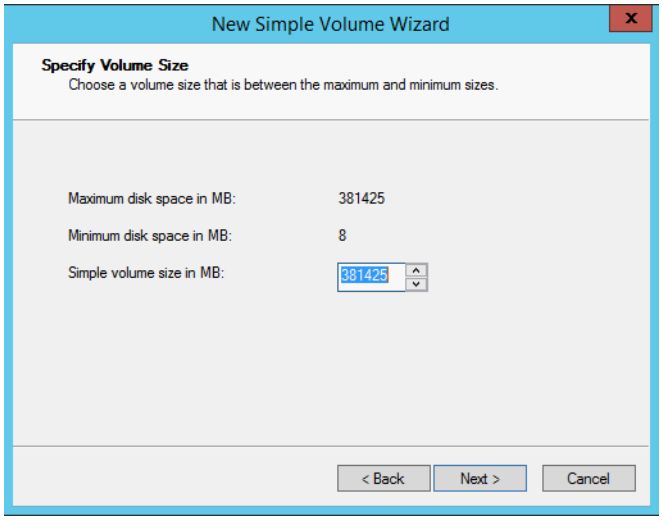


Figure 17 Specify a volume size

6. You also must assign a drive letter or path for a new volume, as shown in Figure 18 on page 16.

Figure 18 Assign drive letter or path

7. You must format the new volume and specify the file system parameters, such as block size and volume label, as shown in Figure 19.

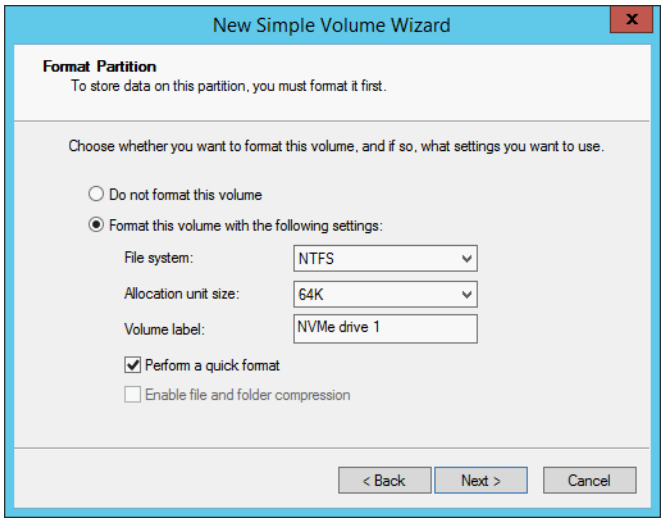


Figure 19 Format partition

8. Review all parameters and click **Finish**, as shown in Figure 20.

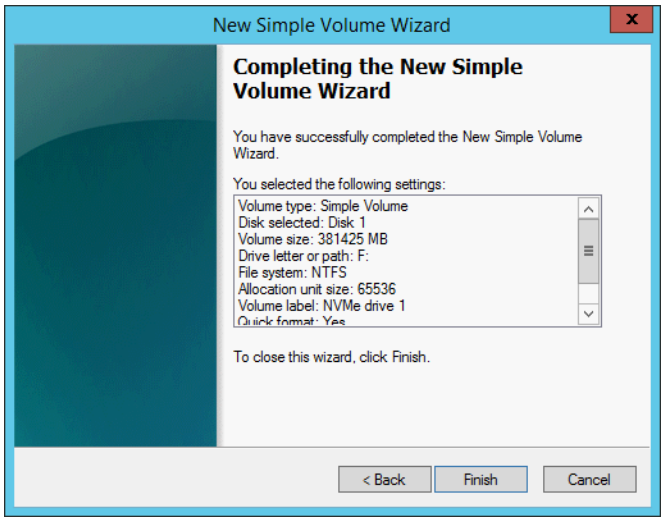


Figure 20 Completing the wizard

9. After the New Simple Volume wizard completes, you can see the new volume NVMe drive 1 (F:) by using the Disk Management tool, as shown in Figure 21.

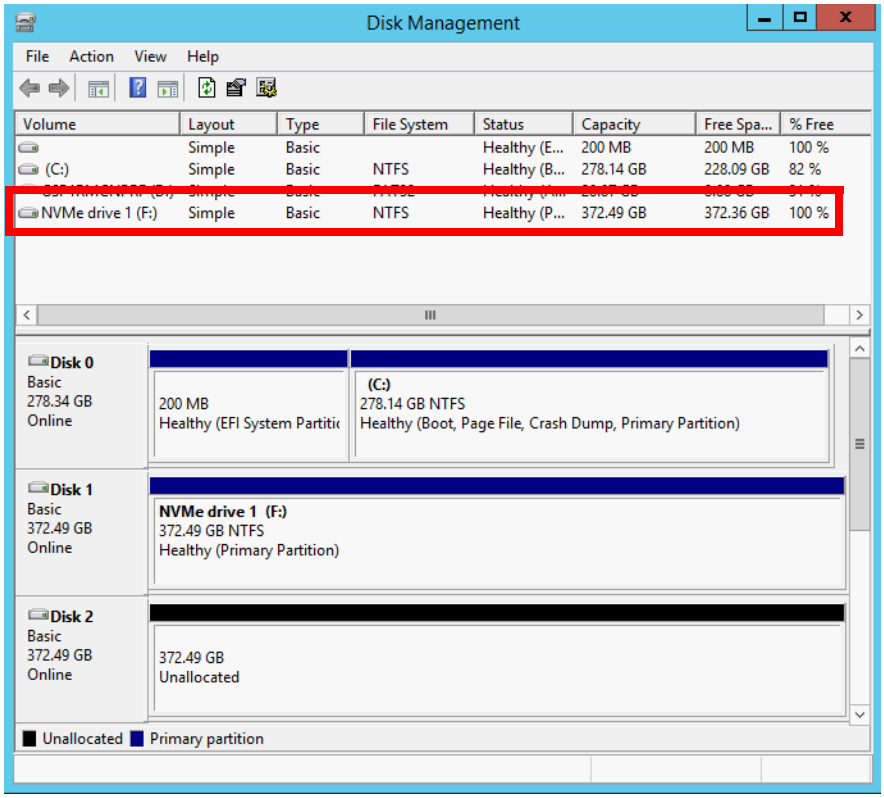


Figure 21 New NVMe volume

You now have NVMe drives that are available for storage. You can create software RAID arrays of different types by using two or more drives.

Using NVMe drives with VMware ESXi server

NVMe drives are supported by ESXi in the following configurations:

- VMware ESXi 6.0 and 6.5: Native support of NVMe drivers
- VMware ESXi 5.5: Requires additional driver to install

Note: In this document, we used the x3850 X6 server as our test system.

The ESXi 5.5 driver for NVMe drives drivers can be downloaded from the following VMware web page:

https://my.vmware.com/web/vmware/info/slug/datacenter_cloud_infrastructure/vmware_vsphere/5_5#drivers_tools

Complete the following steps to install the NVMe driver on ESXi 5.5:

1. Download VMware ESXi 5.5 NVMe driver from the above web page.
2. Enable SSH on the ESXi server.
3. Copy the VMware ESXi 5.5 NVMe driver to the ESXi server by using any SSH client, such as lSCP or WinSCP by using a command that is similar to the command that is shown in Figure 22.

```
scp VMW-ESX-5.5.0-nvme-1.2.0.27-3205218.zip root@172.16.32.222:/tmp/
```

Figure 22 SCP usage for driver copying

4. Log in to the ESXi server by using SSH client and extract the ZIP file, as shown in Figure 23.

```
~ # cd /tmp
/tmp # unzip VMW-ESX-5.5.0-nvme-1.2.0.27-3205218.zip
Archive:  VMW-ESX-5.5.0-nvme-1.2.0.27-3205218.zip
  inflating: VMW-ESX-5.5.0-nvme-1.2.0.27-offline_bundle-3205218.zip
  inflating: nvme-1.2.0.27-4vmw.550.0.0.1331820.x86_64.vib
  inflating: doc/README.txt
  inflating: source/driver_source_nvme_1.2.0.27-4vmw.550.0.0.1331820.tgz
  inflating: doc/open_source_licenses_nvme_1.2.0.27-4vmw.550.0.0.1331820.txt
  inflating: doc/release_note_nvme_1.2.0.27-4vmw.550.0.0.1331820.txt
/tmp #
```

Figure 23 Extracting drivers from the archive

5. Install the extracted NVMe driver on the ESXi server, as shown in Figure 24.

```
/tmp # esxcli software vib install -d
/tmp/VMW-ESX-5.5.0-nvme-1.2.0.27-offline_bundle-3205218.zip
Installation Result
   Message: The update completed successfully, but the system needs to be rebooted
for the changes to be effective.
   Reboot Required: true
   VIBs Installed: VMware_bootbank_nvme_1.2.0.27-4vmw.550.0.0.1331820
   VIBs Removed:
   VIBs Skipped:
/tmp #
```

Figure 24 NVMe driver installation process

6. For Intel NVMe SSDs, an additional driver is recommended. Download the latest version of “NVMe driver for Intel” from the following page:

   https://my.vmware.com/web/vmware/info/slug/datacenter_cloud_infrastructure/vmware_vsphere/5_5#drivers_tools

   For Toshiba and Samsung NVMe drives no additional drivers are required.

7. Upload, extract and install additional driver, as shown in Figure 25.

```
/tmp # unzip  VMW-ESX-5.5.0-intel-nvme-1.0e.2.0-3132116.zip
Archive:  VMW-ESX-5.5.0-intel-nvme-1.0e.2.0-3132116.zip
  inflating: VMW-ESX-5.5.0-intel-nvme-1.0e.2.0-offline_bundle-3132116.zip
  inflating: intel-nvme-1.0e.2.0-1OEM.550.0.0.1391871.x86_64.vib
  inflating: doc/README.txt
  inflating: doc/release_note_intel-nvme_1.0e.2.0-1OEM.550.0.0.1391871.pdf
/tmp # esxcli software vib install -d
/tmp/VMW-ESX-5.5.0-intel-nvme-1.0e.2.0-offline_bundle-3132116.zip
Installation Result
   Message: The update completed successfully, but the system needs to be rebooted
for the changes to be effective.
   Reboot Required: true
   VIBs Installed: Intel_bootbank_intel-nvme_1.0e.2.0-1OEM.550.0.0.1391871
   VIBs Removed:
   VIBs Skipped:
/tmp #
```

Figure 25 NVMe driver for Intel installation

8. Reboot the ESXi server.

9. To confirm that the installed NVMe SSDs are recognized by ESXi server after restart, open an SSH connection to the ESXi server and run the following command, as shown in Figure 26.

```
~ # esxcli storage core device list | grep -i nvm
t10.NVMe____MZQLW1T9HMJP2D000V3_____________________S3JZNX0HB00306_____00000001
   Display Name: Local NVMe Disk (t10.NVMe____MZQLW1T9HMJP2D000V3_____________________S3JZNX0HB00306_____00000001)
   Devfs Path: /vmfs/devices/disks/t10.NVMe____MZQLW1T9HMJP2D000V3_____________________S3JZNX0HB00306_____00000001
   Vendor: NVMe
t10.NVMe____PX04PMB096_____________________________0034010510390000
   Display Name: Local NVMe Disk (t10.NVMe____PX04PMB096_____________________________0034010510390000)
   Devfs Path: /vmfs/devices/disks/t10.NVMe____PX04PMB096_____________________________0034010510390000
   Vendor: NVMe
t10.NVMe____INTEL_SSDPE2MD020T4L___________________CVFT601200182P0KGN__00000001
   Display Name: Local NVMe Disk (t10.NVMe____INTEL_SSDPE2MD020T4L___________________CVFT601200182P0KGN__00000001)
   Devfs Path: /vmfs/devices/disks/t10.NVMe____INTEL_SSDPE2MD020T4L___________________CVFT601200182P0KGN__00000001
   Vendor: NVMe
t10.NVMe____INTEL_SSDPEDMD016T4L___________________CVFT5170000D1P6DGN__00000001
   Display Name: Local NVMe Disk (t10.NVMe____INTEL_SSDPEDMD016T4L___________________CVFT5170000D1P6DGN__00000001)
   Devfs Path: /vmfs/devices/disks/t10.NVMe____INTEL_SSDPEDMD016T4L___________________CVFT5170000D1P6DGN__00000001
   Vendor: NVMe
t10.NVMe____MZQLW1T9HMJP2D000V3_____________________S3JZNX0HB00301_____00000001
   Display Name: Local NVMe Disk (t10.NVMe____MZQLW1T9HMJP2D000V3_____________________S3JZNX0HB00301_____00000001)
   Devfs Path: /vmfs/devices/disks/t10.NVMe____MZQLW1T9HMJP2D000V3_____________________S3JZNX0HB00301_____00000001
   Vendor: NVMe
~ #
```

Figure 26 List of discovered NVMe devices

The NVMe drives are now available for use. You can use NVMe drives as VMFS datastores or as Virtual Flash to improve I/O performance for all virtual machines or pass-through NVMe drives to the dedicated virtual machines.

Ongoing NVMe drive management

We discuss the ongoing management of NVMe drives, including how to correctly work with failed drives in a RAID array, in , “Managing NVMe drives and software RAID recovery” on page 20.

Managing NVMe drives and software RAID recovery

In this section we describe the NVMe drive replacement procedure and software RAID recovery for Linux and Windows operating systems. We show you how to locate a failed NVMe drive, how to gracefully hot-remove it from the server while the system is running (where supported) and how to recover the software RAID after drive replacement.

Hot-swap support: Not all servers that support NVMe drives support the hot-swap capability of those drives. See Table 1 on page 4.

Note: In this document, we used the x3850 X6 server as our test system.

Software RAID initialization in Linux

Linux natively supports software RAID technology and the `mdadm` utility is a standard RAID management tool available in most Linux distributions. mdadm supports the most common RAID levels like RAID-0, RAID-1, RAID-5, RAID-10. In this section we show how to initialize software RAID5 consisting of four NVMe drives on RHEL 7.2.

To create a new software RAID array, follow these steps:

1. Check that all installed NVMe drives are recognized by OS

   Use `lspci` command to get the list of recognized NVMe drives, as shown in Figure 27:

```
[root@rhel7-n6hlne8 ~]# lspci | grep -i "non-vol"
41:00.0 Non-Volatile memory controller: Intel Corporation PCIe Data Center SSD (rev 01)
49:00.0 Non-Volatile memory controller: Intel Corporation PCIe Data Center SSD (rev 01)
4e:00.0 Non-Volatile memory controller: Intel Corporation PCIe Data Center SSD (rev 01)
51:00.0 Non-Volatile memory controller: Intel Corporation PCIe Data Center SSD (rev 01)
```

Figure 27 lspci command output

   As you can see in previous Figure 27, Linux has found four NVMe drives installed in the server. At the beginning of each line you will see the unique PCIe address of each NVMe drive. You can get more information about any drive using its PCIe address, as shown in Figure 28:

```
[root@rhel7-n6hlne8 ~]# lspci -s 49:00.0 -v
49:00.0 Non-Volatile memory controller: Intel Corporation PCIe Data Center SSD (rev
01) (prog-if 02 [NVM Express])
        Subsystem: Intel Corporation DC P3700 SSD [2.5" SFF]
        Physical Slot: 19
        Flags: bus master, fast devsel, latency 0, IRQ 38
        Memory at e7cfc000 (64-bit, non-prefetchable) [size=16K]
        Expansion ROM at e7c00000 [disabled] [size=64K]
        Capabilities: [40] Power Management version 3
        Capabilities: [50] MSI-X: Enable+ Count=32 Masked-
        Capabilities: [60] Express Endpoint, MSI 00
        Capabilities: [100] Advanced Error Reporting
        Capabilities: [150] Virtual Channel
        Capabilities: [180] Power Budgeting <?>
        Capabilities: [190] Alternative Routing-ID Interpretation (ARI)
        Capabilities: [270] Device Serial Number 55-cd-2e-41-4c-9d-08-a1
        Capabilities: [2a0] #19
        Kernel driver in use: nvme
```

Figure 28 Detailed lspci output for specific NVMe drive

   As shown in previous Figure 28, using verbose (**-v**) mode of the `lspci` command you can get PCIe slot number and serial number of the drive, which we will use later during the drive replacement procedure.

2. Check that every NVMe drive has associated block device. Use `ls` command to locate NVMe block devices in `/dev` directory, as shown in Figure 29:

```
[root@rhel7-n6hlne8 ~]# ls /dev/nvme*
/dev/nvme0  /dev/nvme0n1  /dev/nvme1  /dev/nvme1n1  /dev/nvme2  /dev/nvme2n1
/dev/nvme3  /dev/nvme3n1
```

Figure 29 List of NVMe block devices in /dev directory

Every NVMe drive is represented in the OS as `/dev/nvmeXn1` device. As shown in previous Figure 29 on page 21, four block devices were created.

You can also check the drive capacity using `parted` utility, as shown in Figure 30:

```
[root@rhel7-n6hlne8 ~]# parted /dev/nvme0n1 print
Error: /dev/nvme0n1: unrecognised disk label
Model: Unknown (unknown)
Disk /dev/nvme0n1: 1600GB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:
[root@rhel7-n6hlne8 ~]#
```

Figure 30 parted utility output

3. Create a new software RAID using NVMe drives

   Using mdadm utility you can initialize a new array `/dev/md0`. In this example we create a RAID-5 array consisting of four drives `/dev/nvme0n1 /dev/nvme1n1 /dev/nvme2n1 /dev/nvme3n1`, as shown in Figure 31:

```
[root@rhel7-n6hlne8 ~]# mdadm -C /dev/md0 --force --level=raid5 --bitmap=internal
--raid-devices=4 --assume-clean /dev/nvme0n1 /dev/nvme1n1 /dev/nvme2n1 /dev/nvme3n1
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

Figure 31 Software RAID-5 initialization

To check the status of the array run the following commands, as shown in Figure 32 and Figure 33 on page 23:

```
[root@rhel7-n6hlne8 ~]# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
md0 : active raid5 nvme3n1[3] nvme2n1[2] nvme1n1[1] nvme0n1[0]
      4688047104 blocks super 1.2 level 5, 512k chunk, algorithm 2 [4/4] [UUUU]
      bitmap: 0/12 pages [0KB], 65536KB chunk

unused devices: <none>
```

Figure 32 Array status

As you can see in previous Figure 32 on page 22, the array is in active state and all four drives are available.

```
[root@rhel7-n6hlne8 ~]# mdadm --detail /dev/md0
/dev/md0:
        Version : 1.2
  Creation Time : Fri May 20 19:21:42 2016
     Raid Level : raid5
     Array Size : 4688047104 (4470.87 GiB 4800.56 GB)
  Used Dev Size : 1562682368 (1490.29 GiB 1600.19 GB)
   Raid Devices : 4
  Total Devices : 4
    Persistence : Superblock is persistent

  Intent Bitmap : Internal

    Update Time : Fri May 20 19:21:42 2016
          State : clean
 Active Devices : 4
Working Devices : 4
 Failed Devices : 0
  Spare Devices : 0

         Layout : left-symmetric
     Chunk Size : 512K

           Name : rhel7-n6hlne8.poc.bts.lab:0  (local to host
rhel7-n6hlne8.poc.bts.lab)
           UUID : 6acdc9c0:a56492f6:d13cfd69:fdf81253
         Events : 0

    Number   Major   Minor   RaidDevice State
       0     259        2        0      active sync   /dev/nvme0n1
       1     259        1        1      active sync   /dev/nvme1n1
       2     259        3        2      active sync   /dev/nvme2n1
       3     259        0        3      active sync   /dev/nvme3n1
```

Figure 33 Detailed information about the array

As you can see in previous example (Figure 33), the total array size is 4480.56 GB, all drives are active and in sync state, the array has no failed drives.

4. You can also use `mdadm` command to generate config file, as shown in Figure 34:

```
[root@rhel7-n6hlne8 ~]# mdadm -E -s
ARRAY /dev/md/0  metadata=1.2 UUID=6acdc9c0:a56492f6:d13cfd69:fdf81253
name=rhel7-n6hlne8.poc.bts.lab:0
[root@rhel7-n6hlne8 ~]# mdadm -E -s >> /etc/mdadm.conf
```

Figure 34 mdadm configuration file creation

When you complete this procedure, you will have a working software RAID. You can use it as a regular block device: you can create partitions and file systems, mount it to the file system tree.

NVMe drive hot-replacement in Linux

In this section we cover the hot-replacement procedure of the failed NVMe drive in RHEL7.2. Hot-replacement means that we perform graceful hot-remove and hot-plug procedure on the running system without any interruption in service or downtime.

Note 1: Not every Linux distribution supports NVMe drive hot-replacement; it depends on Linux kernel version. Here is the list of distributions and Linux kernels that were validated at the time of writing:

- RHEL 7.0 and higher, kernel 3.10.0-123.el7.x86_64 and higher
- RHEL 6.6 and higher, kernel 2.6.32-500.el6.x86_64 and higher
- SLES 12, kernel 3.12.28-2 rc 3 and higher

Note 2: Not all Lenovo servers that support NVMe drives also support hot-add and hot-replace functions with those NVMe drives. See Table 1 on page 4.

To enable hot-replacement feature in Linux you need to set the following kernel parameter: `pci=pcie_bus_perf`. To do that you need to add that line as the kernel boot argument to the bootloader configuration file (grub.cfg or elilo.conf).

In this section we simulate the outage of one of the NVMe drive just to demonstrate the hot-replacement concept. We use hardware and RAID configuration described in the previous section “Software RAID initialization in Linux” on page 21.

Follow the described procedure below, to perform a graceful NVMe drive hot-replacement operation:

1. Make sure that required Linux kernel is running and `pci` kernel parameter has required value.
2. Run the following command to check the running kernel version and its boot parameters, as shown in Figure 35:

```
[root@rhel7-n6hlne8 ~]# cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-3.10.0-327.el7.x86_64 root=/dev/mapper/rhel-root ro
rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet pci=pcie_bus_perf
```

Figure 35 kernel boot parameters

3. Mark the failed NVMe drive as a “failed” drive in mdadm configuration

   Let’s assume that one of the installed NVMe drive is failed, `/dev/nvme1n1` for example. First of all, you need to mark this drive as a “failed” drive in mdadm configuration, as shown in Figure 36:

```
[root@rhel7-n6hlne8 ~]# mdadm --manage /dev/md0 --fail /dev/nvme1n1
mdadm: set /dev/nvme1n1 faulty in /dev/md0
```

Figure 36 Failed drive designation

To make sure, that array status has changed, run the following command, as shown in Figure 37:

```
[root@rhel7-n6hlne8 ~]#  cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
md0 : active raid5 nvme1n1[1](F) nvme2n1[2] nvme3n1[3] nvme0n1[0]
      4688047104 blocks super 1.2 level 5, 512k chunk, algorithm 2 [4/3] [U_UU]
      bitmap: 0/12 pages [0KB], 65536KB chunk

unused devices: <none>
```

Figure 37 Array status

As you can see in Figure 37, `nvme1n1` drive is in the failed state and the array now has only 3 active drives.

4. Determine the PCIe address and PCIe slot number used by the failed drive

   You need to run a couple of commands to locate the failed NVMe drive in the server. First of all, you need to find out the PCIe address of the `nvme1n1` drive. To do that, run the following command, as shown in Figure 38:

```
[root@rhel7-n6hlne8 ~]# find /sys/devices | egrep 'nvme1[0-9]?$'
/sys/devices/pci0000:40/0000:40:02.2/0000:47:00.0/0000:48:02.0/0000:49:00.0/nvme/nvme1
```

Figure 38 PCIe address location of the failed drive

As you can see, the failed `nvme1n1` drive has PCIe address `0000:49:00.0`. To determine the PCIe slot number of the failed drive, you can use `lspci` command, as show in Figure 28 on page 21, or you can run the following command, as shown in Figure 39:

```
[root@rhel7-n6hlne8 ~]# grep '49:00' /sys/bus/pci/slots/*/address
/sys/bus/pci/slots/19/address:0000:49:00
```

Figure 39 PCIe slot number determination

As you can see, both mentioned commands show the same result – the `nvme1n1` drive is located in PCIe slot 19, the upper drive bay in the Storage book (check Figure 3 on page 6).

5. Power off the failed NVMe drive

   Now you need to gracefully power off the failed NVMe drive located in PCIe slot 19. To perform that you need to run the following command, as shown in Figure 40:

```
[root@rhel7-n6hlne8 ~]# echo 0 > /sys/bus/pci/slots/19/power
```

Figure 40 Power off the failed NVMe drive

You can check that the drive is shut down and is not represented in OS any more using `lspci` and `lsblk` command, as shown in following Figure 41:

```
[root@rhel7-n6hlne8 ~]# lspci | grep -i "non-vol"
41:00.0 Non-Volatile memory controller: Intel Corporation PCIe Data Center SSD (rev 01)
4e:00.0 Non-Volatile memory controller: Intel Corporation PCIe Data Center SSD (rev 01)
51:00.0 Non-Volatile memory controller: Intel Corporation PCIe Data Center SSD (rev 01)
[root@rhel7-n6hlne8 ~]# lsblk
NAME          MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
sda             8:0    0 744.1G  0 disk
??sda1          8:1    0   200M  0 part  /boot/efi
??sda2          8:2    0   128M  0 part
??sda3          8:3    0 353.2G  0 part
??sda4          8:4    0   500M  0 part  /boot
??sda5          8:5    0   104G  0 part
? ??rhel-root 253:0    0   100G  0 lvm   /
? ??rhel-swap 253:1    0     4G  0 lvm   [SWAP]
??sda6          8:6    0     2G  0 part
??sda7          8:7    0    80G  0 part
nvme0n1       259:2    0   1.5T  0 disk
??md0           9:0    0   4.4T  0 raid5
nvme2n1       259:3    0   1.5T  0 disk
??md0           9:0    0   4.4T  0 raid5
nvme3n1       259:0    0   1.5T  0 disk
??md0           9:0    0   4.4T  0 raid5
```

Figure 41 lspci and lsblk output

As you can see, `lspci` shows only that now only three NVMe drives are available, `lsblk` also shows three drives: `nvme0n1`, `nvme2n1 and nvme3n1`, which are combined in RAID-5.

6. Replace the failed NVMe drive

   As we found out previously, the failed `nvme1n1` drive is located in PCIe slot 19, in the Storage book bay 7. Now it's safe to remove the NVMe drive from the Storage book, you can replace the failed drive on the new one (same model and capacity).

The described procedure is applicable to NVMe drives of any vendor. For example, for two 900 GB Toshiba NVMe drives you may have, the hot-replacement procedure is as follows:

1. Determine that all Toshiba NVMe drives installed in the system are recognized by OS

   Run the lspci and lsblk commands, as shown in Figure 42:

```
linux-8aq1:~ # lspci | grep -i Non-Volatile
09:00.0 Non-Volatile memory controller: Toshiba America Info Systems Device 010e (rev 01)
0e:00.0 Non-Volatile memory controller: Toshiba America Info Systems Device 010e (rev 01)
49:00.0 Non-Volatile memory controller: Intel Corporation PCIe Data Center SSD (rev 01)
4e:00.0 Non-Volatile memory controller: Intel Corporation PCIe Data Center SSD (rev 01)
95:00.0 Non-Volatile memory controller: Intel Corporation PCIe Data Center SSD (rev 01)
linux-8aq1:~ #

linux-8aq1:~ # lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
nvme0n1     259:2    0 894.3G  0 disk
nvme1n1     259:3    0 894.3G  0 disk
nvme2n1     259:1    0   1.8T  0 disk
nvme3n1     259:0    0   1.8T  0 disk
nvme4n1     259:4    0   1.5T  0 disk
??nvme4n1p1 259:5    0   200M  0 part
??nvme4n1p2 259:6    0   128M  0 part
??nvme4n1p3 259:7    0  97.7G  0 part
??nvme4n1p4 259:8    0    50G  0 part /
??nvme4n1p5 259:9    0   502M  0 part /boot/efi
linux-8aq1:~ #
```

Figure 42 lspci and lsblk output

   As you can see two 900 GB Toshiba NVMe drives are recognized by OS, device names `nvme0n1` and `nvme1n1` have been assigned to them.

2. Determine PCIe slot numbers used by the drives

   Run the following commands to determine PCIe slot numbers, as shown in Figure 43:

```
linux-8aq1:~ # find /sys/devices | egrep 'nvme0[0-9]?$'
/sys/devices/pci0000:00/0000:00:02.2/0000:07:00.0/0000:08:02.0/0000:09:00.0/nvme/nvme0
linux-8aq1:~ # grep '09:00' /sys/bus/pci/slots/*/address
/sys/bus/pci/slots/17/address:0000:09:00
linux-8aq1:~ # find /sys/devices | egrep 'nvme1[0-9]?$'
/sys/devices/pci0000:00/0000:00:02.3/0000:0c:00.0/0000:0d:03.0/0000:0e:00.0/nvme/nvme1
linux-8aq1:~ # grep '0e:00' /sys/bus/pci/slots/*/address
/sys/bus/pci/slots/16/address:0000:0e:00
```

Figure 43 PCIe slot number determination

   As you can see in the text in red, `nvme0n1` and `nvme1n1` drives are located in PCIe slots 17 and 16.

3. Power off the required NVMe drives

   You can gracefully power off one or both NVMe drives located in PCIe slots 16 and 17. In order to do that, you need to run the following commands, as shown in Figure 44:

```
linux-8aq1:~ # echo 0 > /sys/bus/pci/slots/16/power
linux-8aq1:~ # echo 0 > /sys/bus/pci/slots/17/power
```

Figure 44 Power off the NVMe drives

   To check that both drives are offline and don't exists in OS any more, run `lsblk` command, as shown in following Figure 45:

```
linux-8aq1:~ # lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme2n1     259:1    0  1.8T  0 disk
nvme3n1     259:0    0  1.8T  0 disk
nvme4n1     259:4    0  1.5T  0 disk
??nvme4n1p1 259:5    0  200M  0 part
??nvme4n1p2 259:6    0  128M  0 part
??nvme4n1p3 259:7    0 97.7G  0 part
??nvme4n1p4 259:8    0   50G  0 part /
??nvme4n1p5 259:9    0  502M  0 part /boot/efi
linux-8aq1:~ #
```

Figure 45 lsblk output after NVMe drives shutdown

   As you can see both Toshiba NVMe drives are not in the list of available block devices any more.

4. Replace required Toshiba NVMe drives

NVMe drive hot-plug and software RAID recovery in Linux

As described in the previous section , "NVMe drive hot-replacement in Linux" on page 24, you can perform hot-removal operation for the failed NVMe drive, by following the described procedure. When the drive is replaced, you need to power it on and then recover related software RAID. In order to do that, follow the next procedure:

Note: In this section we describe hot-plug procedure using Intel NVMe drives. For Samsung and Toshiba NVMe drives the procedure is the same.

1. Power on the new drive

   Similar to the power-off procedure, as shown in Figure 40 on page 25, run the following command:

```
[root@rhel7-n6hlne8 ~]# echo 1 > /sys/bus/pci/slots/19/power
```

Figure 46 Power on the new NVMe drive

   If you put the new NVMe drive in the same Storage book bay, where the failed drive was located, PCIe slot number remains the same – in this example that is PCIe slot 19.

2. Ensure that the new drive has been successfully started and recognized by the OS by using the `lspci` and `lsblk` commands, as shown in Figure 47:

```
[root@rhel7-n6hlne8 ~]# lspci | grep -i "non-vol"
41:00.0 Non-Volatile memory controller: Intel Corporation PCIe Data Center SSD (rev 01)
49:00.0 Non-Volatile memory controller: Intel Corporation PCIe Data Center SSD (rev 01)
4e:00.0 Non-Volatile memory controller: Intel Corporation PCIe Data Center SSD (rev 01)
51:00.0 Non-Volatile memory controller: Intel Corporation PCIe Data Center SSD (rev 01)
[root@rhel7-n6hlne8 ~]# lsblk
NAME          MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
sda             8:0    0 744.1G  0 disk
??sda1          8:1    0   200M  0 part  /boot/efi
??sda2          8:2    0   128M  0 part
??sda3          8:3    0 353.2G  0 part
??sda4          8:4    0   500M  0 part  /boot
??sda5          8:5    0   104G  0 part
? ??rhel-root 253:0    0   100G  0 lvm   /
? ??rhel-swap 253:1    0     4G  0 lvm   [SWAP]
??sda6          8:6    0     2G  0 part
??sda7          8:7    0    80G  0 part
nvme0n1       259:2    0   1.5T  0 disk
??md0           9:0    0   4.4T  0 raid5
nvme1n1       259:1    0   1.5T  0 disk
nvme2n1       259:3    0   1.5T  0 disk
??md0           9:0    0   4.4T  0 raid5
nvme3n1       259:0    0   1.5T  0 disk
??md0           9:0    0   4.4T  0 raid5
```

Figure 47 lspci and lsblk output

As you can see, `lspci` has shown that the new NVMe drive in PCIe slot 19 has been recognized by Linux kernel. `lsblk` command also has shown that the appropriate block device – `nvme1n1` has been created. However, notice that `nvme1n1` is not associated with any array yet.

3. Recover the software RAID array

The existing RAID-5 now is in degraded state – only 3 of 4 drive are available and active. You you need to add a new drive to the array to recover it. In order to do that, you need to run the following command:

```
[root@rhel7-n6hlne8 ~]# mdadm --manage /dev/md0 --add /dev/nvme1n1
mdadm: added /dev/nvme1n1
```

Figure 48 Addining a new NVMe drive to the existing array

You can check RAID status using the comands in Figure 49.

```
[root@rhel7-n6hlne8 ~]# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
md0 : active raid5 nvme1n1[4] nvme2n1[2] nvme3n1[3] nvme0n1[0]
      4688047104 blocks super 1.2 level 5, 512k chunk, algorithm 2 [4/3] [U_UU]
      [======>.............]  recovery = 31.8% (497008068/1562682368) finish=92.0min
speed=192951K/sec
      bitmap: 0/12 pages [0KB], 65536KB chunk

unused devices: <none>
```

Figure 49 Recovery process

As you can see in previous Figure 49, the `nvme1n1` drive has been added successfully and the array has started a recovery process. That may spend some time and affect on the array performance. When it's done, you will have redundant and fully operating RAID.

Software RAID initialization in Windows

In this section we cover software RAID initialization in Windows using Windows Server 2012 R2 as an example. We assume that NVMe drives are installed correctly in the server, recognized by OS and initialized. For more information about NVMe drive initialization in Windows refer to "Using NVMe drives with Microsoft Windows Server" on page 13.

In this example we have two 1.6 TB NVMe drives and we can configure a mirrored volume using both NVMe drives. Follow the next procedure to initialize software RAID1:

1. Open the Disk Management tool, you should see two installed NVMe drives. For our example, both installed NVMe drives are presented as Disk 0 and Disk 1. Right-click on Disk 0 and select **New Mirrored Volume**, as shown in Figure 50:

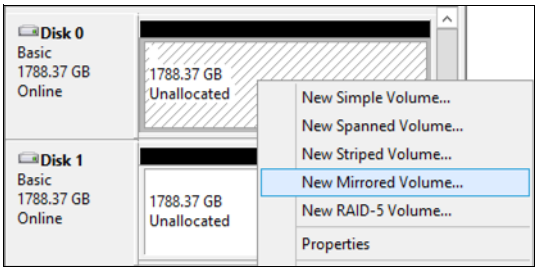


Figure 50 Disk Management tool and new mirrored volume

2. New Mirrored Volume Wizard will be started. Click **Next** and select the second NVMe drive (Disk 1) to add to RAID1, as shown in Figure 51 on page 31:

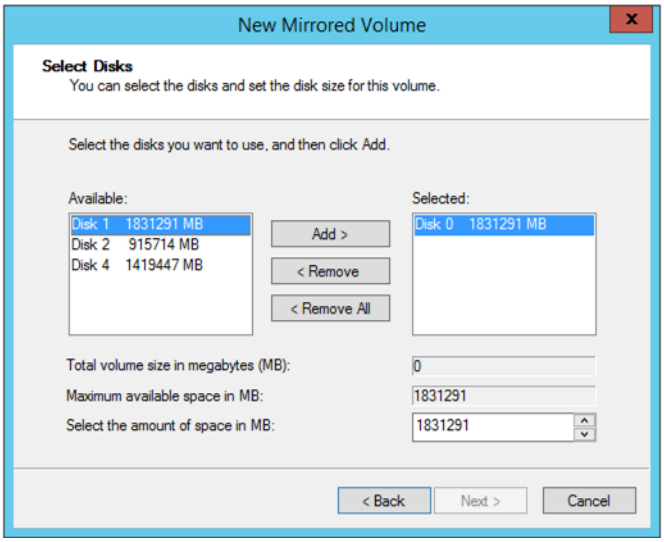


Figure 51 Second NVMe drive selection

3. In Selected column both NVMe drives (Disk 0 and Disk 1) should be selected, as shown in Figure 52:

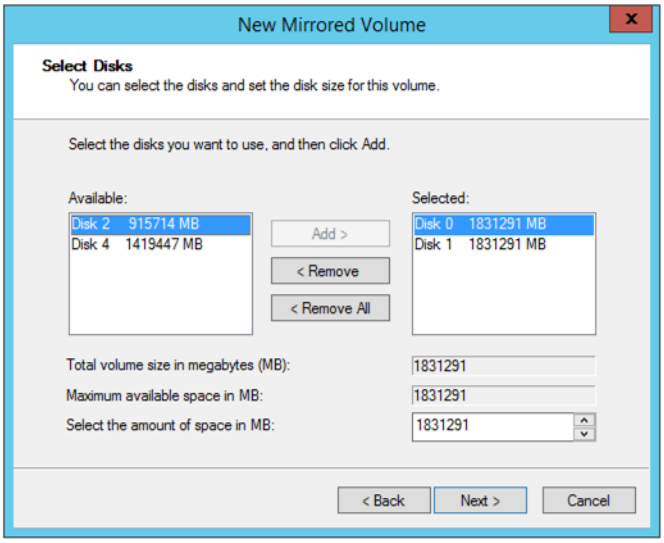


Figure 52 Selected drives for the new mirrored volume

4. Assign a drive letter for the new mirrored volume, as shown in Figure 53:

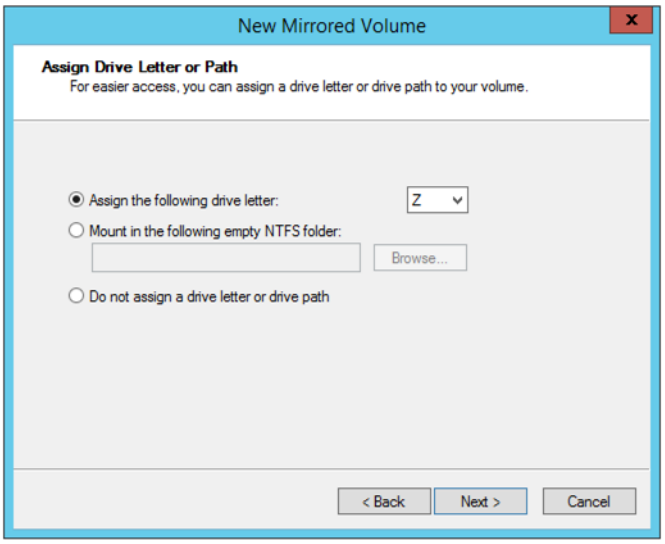


Figure 53 A drive letter assignment

5. Create a file system on the new mirrored volume, as shown in the following Figure 54:

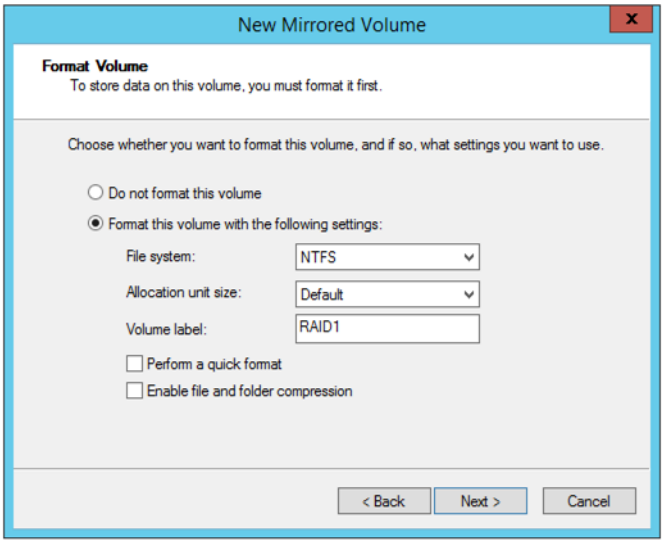


Figure 54 Format volume

6. Confirm settings and finish the new RAID1 setup, as shown in Figure 55:

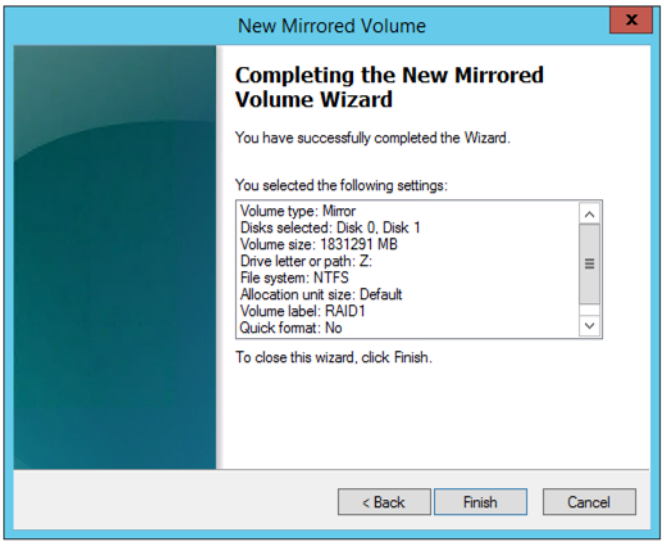


Figure 55 Wizard completion

7. Confirm that you agree to convert selected NVMe drives to dynamic disks, as shown in Figure 56:

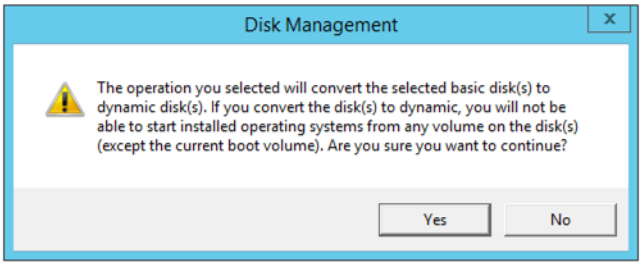


Figure 56 Convertion to dynamic disks

8. As soon as you confirm the last action the formatting procedure will start. It takes time, but you can monitor the progress, as shown in Figure 57:

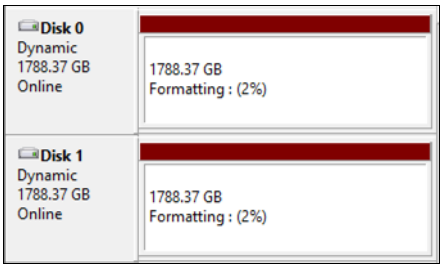


Figure 57 New mirrored volume formatting

Now you have a new software RAID1 consisting of two NVMe drives.

NVMe drive hot-replacement in Windows

In this section we describe how to perform hot-replacement procedure of failed NVMe drive in Windows Server using Windows 2012 R2 as an example. By saying hot-replacement, we mean a replacement procedure of the failed drive on the running system without any interruption in service or downtime.

The NVMe drive and software RAID initialization in Windows 2012 R2 is very simple and straightforward, we don't cover this procedure in this section. For more information about NVMe drive initialization in Windows, refer to , "Using NVMe drives with Microsoft Windows Server" on page 13 and "Software RAID initialization in Windows" on page 30.

Let's assume we have two different hardware configurations with NVMe drives of all three vendors in the system, as shown in the following Figure 58:

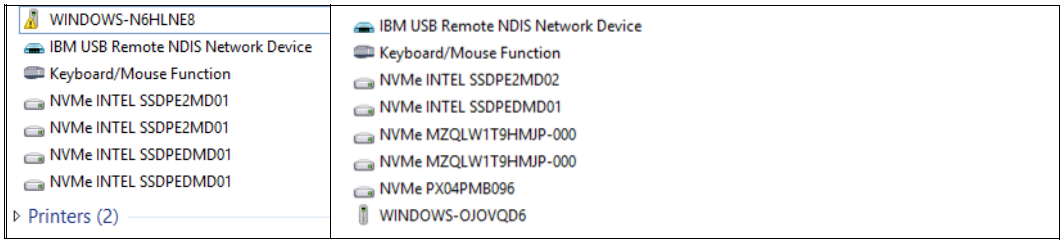


Figure 58 Intel, Toshiba and Samsung NVMe drives in one system

Let's also assume that one of these drives fails and we need to replace it. Follow the next procedure described below, in order to perform the hot-replacement procedure:

1. Put the failed drive to Offline mode

   To be on the safe side, it's recommended to put the failed drive to offline mode. To do that, open **Windows Disk Management** tool, right-click on the failed drive **(Disk 1** or any other failed disk**)** and choose **Offline** from the pop-up menu, as shown in Figure 59.

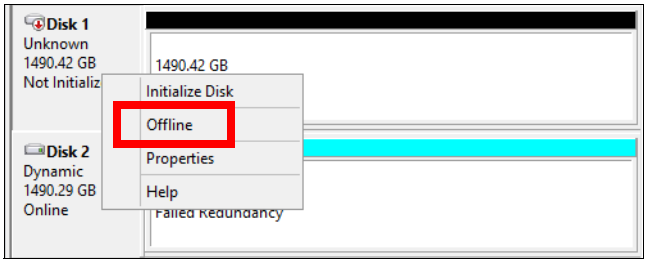


Figure 59 Drive offline mode

2. Locate the failed NVMe drive

   You need to do several steps to find out a physical location of the failed NVMe drive in the Storage book.

   a. Select the failed disk in **Disk Manager** and open **Properties** window, as shown in Figure 60 on page 35:

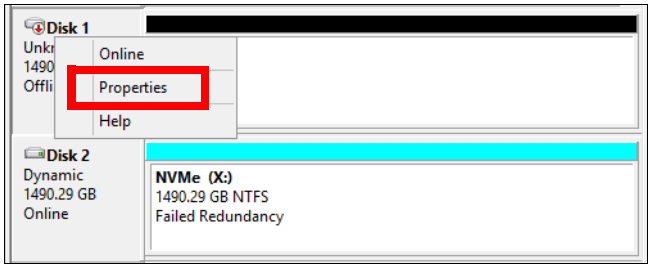


Figure 60 Disk Properties

b. In the **Properties** window, on the **General** tab you can see the PCIe slot number of the associated NVMe drive, as shown in Figure 61:

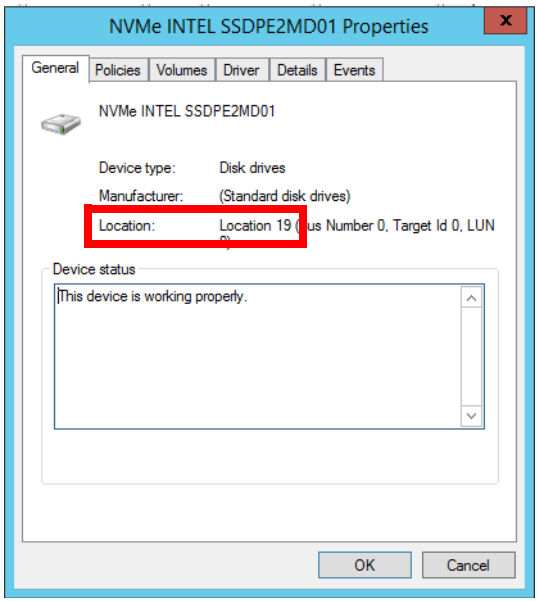


Figure 61 Properties window, General tab, Intel drive location

As you can see on the **General** tab, the location of this particular Intel drive is 19—that means the NVMe drive is located in PCIe slot 19 (bay 7 is the Storage book). For more information about NVMe drives location in the Storage book, refer to , “PCIe slot numbering” on page 6.

Check **Properties** of other NVMe drives if needed. For example, locate the failed Toshiba or Samsung NVMe drive, as shown in Figure 62 on page 36:

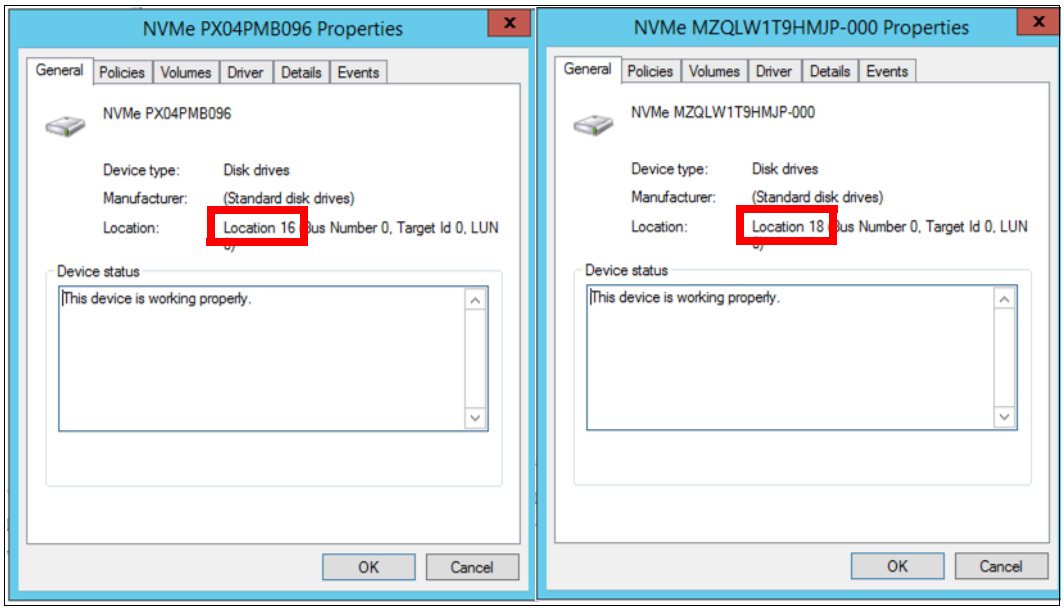


Figure 62 Toshiba and Samsung NVMe drives properties

As you can see, selected Toshiba NVMe drive (presented to OS as PX04PMB096) is located in PCIe slot 16, Samsung NVMe drive (presented to OS as MZQLW1T9HMJP-000) is located in PCIe slot 18.

3. Power off the failed NVMe drive

Now you need to shut down the device from OS. Open the **Devices and Printers** window, there you should see all NVMe drives installed in the server, as shown in Figure 63:

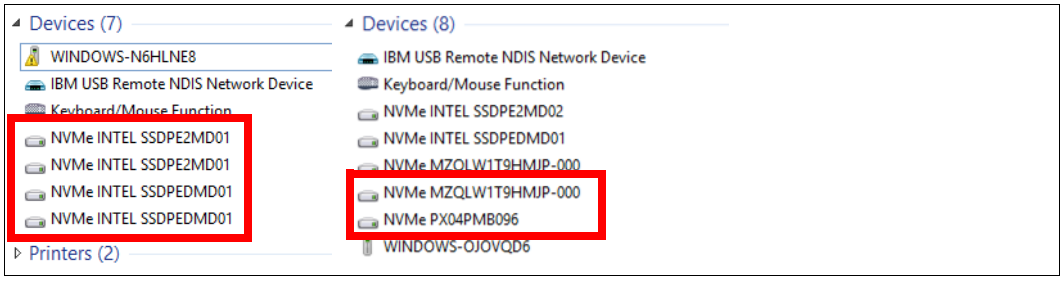


Figure 63 Devices and Printers

If you have several drives of the same type, you may need to double-check their PCIe slot numbers. To do that, right-click on one of the NVMe drives and select **Properties** from the pop-up menu, as shown in Figure 64:

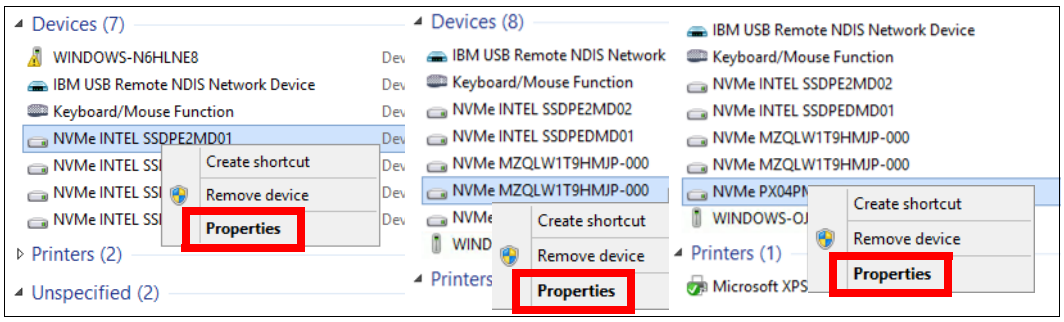


Figure 64 Device properties

Check the **Hardware** tab of the **Properties** window and locate a NVMe drive, which has the Location number 19 for Intel drive, Location number 18 for Samsung drive or Location number 16 for Toshiba drive, as shown in Figure 65:

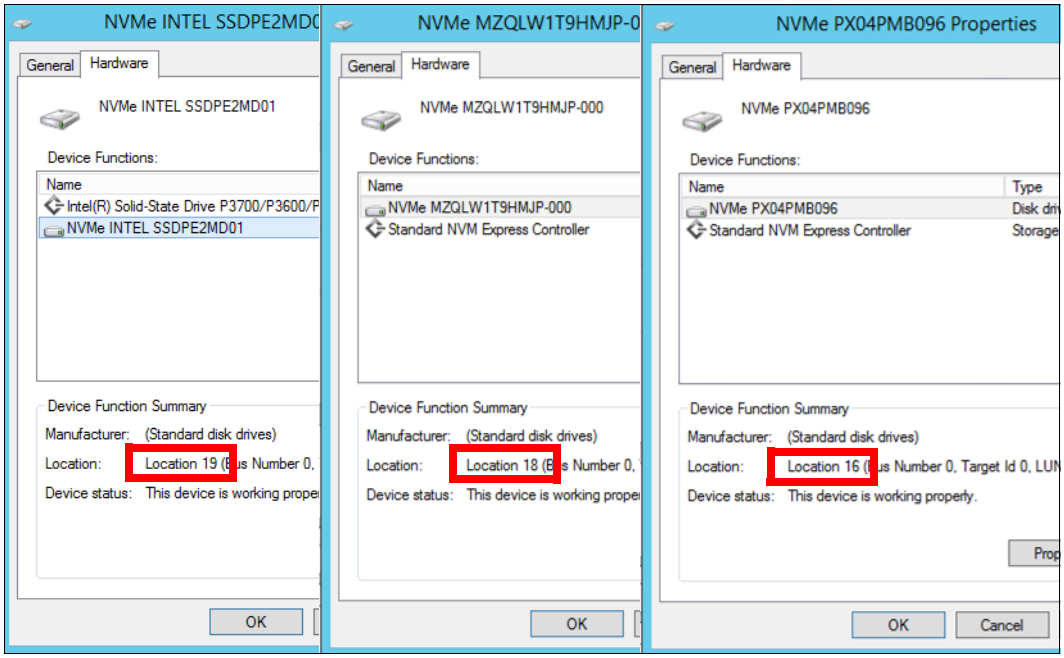


Figure 65 Drive location

Remove the failed drive from OS. Right-click on the appropriate drive in the **Devices and Printers** window and select **Remove device** action, as shown in Figure 66:

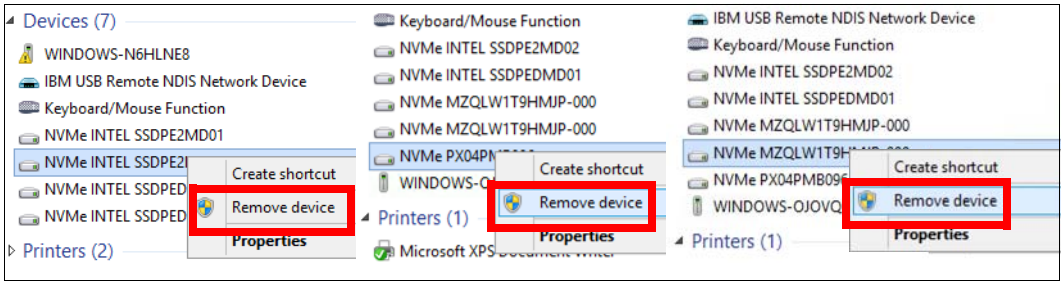


Figure 66 Remove device

4. Replace the failed NVMe drive

   Now you can replace the failed NVMe drive located in appropriate PCIe slot using the same drive model with equal drive capacity.

5. Rescan available devices

   When NVMe drive replacement is performed, you need to rescan available devices from the OS. To do that, open **Device Manager**, choose **Action** from menu and select **Scan for hardware changes**, as shown in Figure 67:

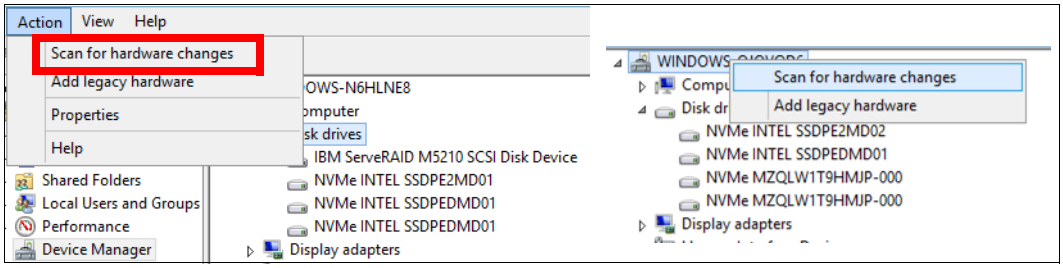


Figure 67 Scan for new devices

When scanning process is finished, you will see a new NVMe drive in list of available devices:

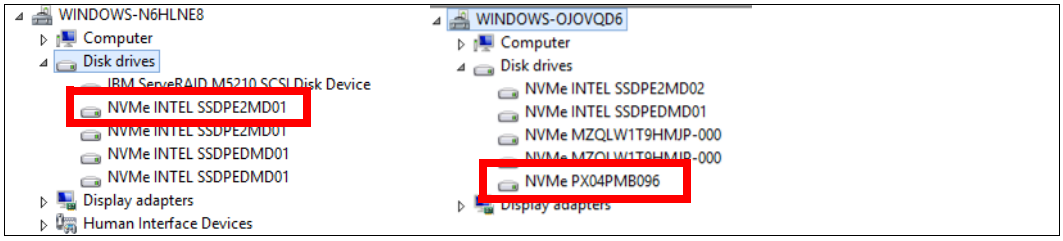


Figure 68 New NVMe drive is discovered

Now you finished NVMe drive hot-replacement procedure.

Software RAID recovery in Windows

In the previous section “NVMe drive hot-replacement in Windows” on page 34 we described how to perform hot-replacement procedure in Windows. In this section we explain how to recover software RAID after drive replacement.

To demonstrate software RAID recovery procedure, let's assume we have four NVMe drives installed in the server and combined in one array – software RAID5. Windows Disk Management tool shows that we have disk **X:** (the volume label is "NVMe"), dynamic RAID5 volume with capacity of 4470.87 GB. Status of the volume is Healthy, all related NVMe drives are online, as shown in Figure 69:

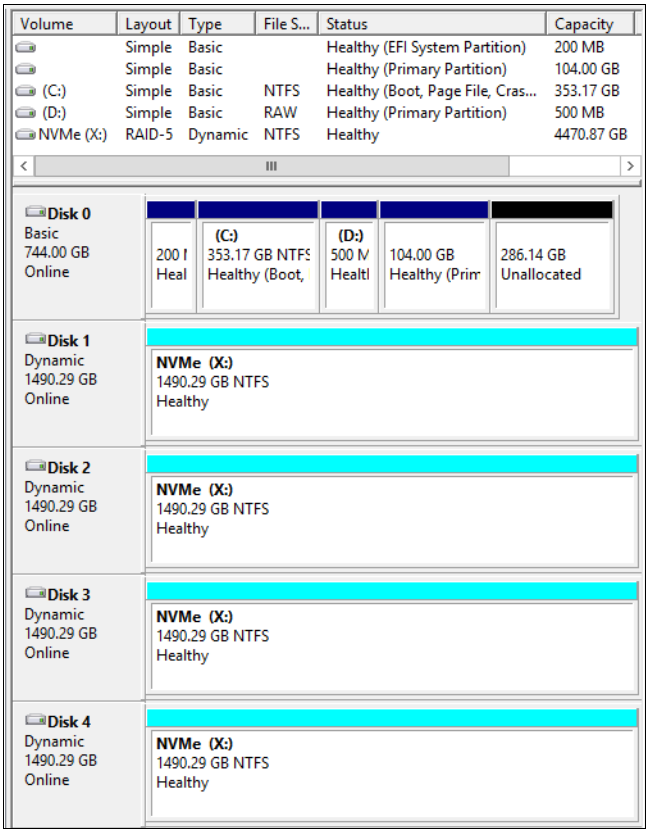


Figure 69 Initial state of the array

Let's also assume one of the NVMe drives is failed – **Disk 1**, for example. The array in this case is still operating, but it has failed redundancy. In **Disk Manager** you may see the following picture, as shown in Figure 70:

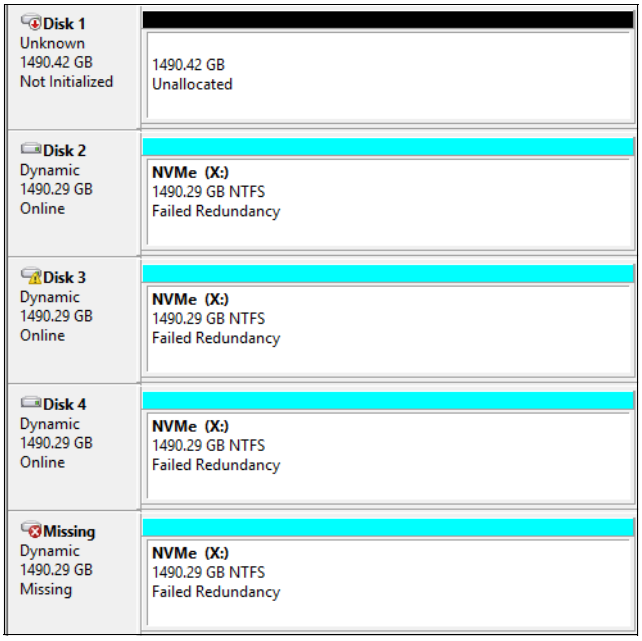


Figure 70 Disk 1 is failed

To perform RAID recovery operation, follow the procedure described below:

1. Perform failed NVMe drive hot-replacement operation

   Follow the procedure described in "NVMe drive hot-replacement in Windows" on page 34 in order to perform drive hot-replacement operation.

2. Initialize a new NVMe drive

   When failed drive is replaced, you need to initialize a new NVMe drive installed in the server. Open **Disk Manager**, where you should see a new not initialized drive (**Disk 1**), as shown in Figure 71:

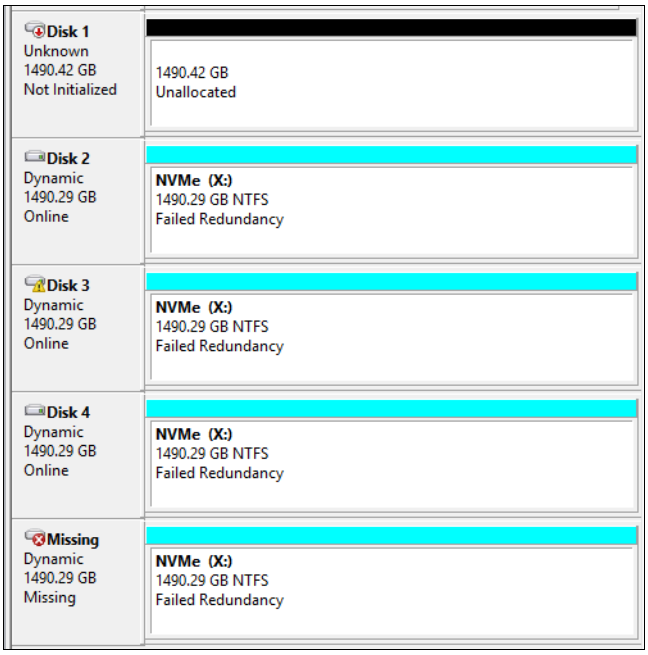


Figure 71 New not initialized drive

Right-click on the new drive (Disk 1) and select **Initialize disk**, as shown in Figure 72:

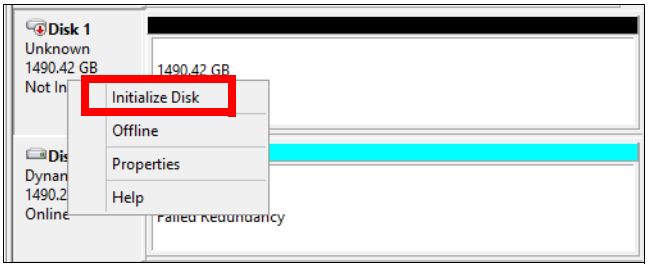


Figure 72 Initialize a new drive

3. Repair volume

When disk initialization is complete you can start an array recovery procedure. To do that, right-click on the **NVMe** volume (disk **X:**) and select **Repair Volume** option from the pop-up menu:

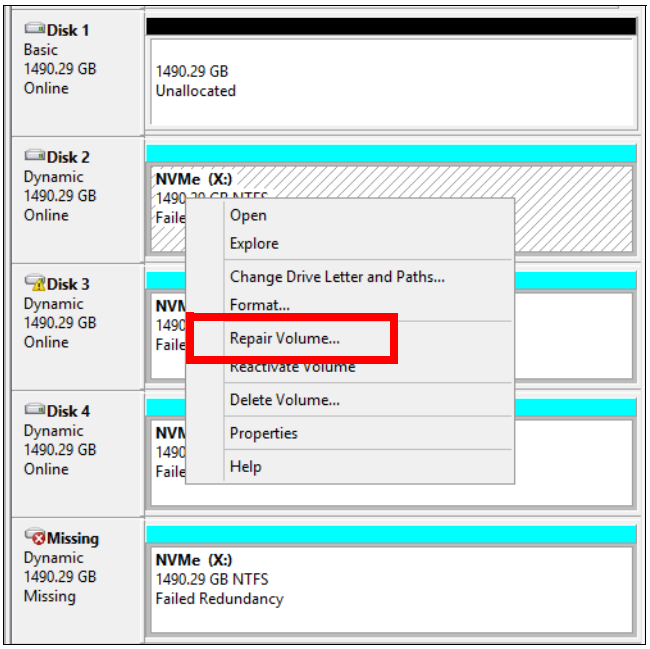


Figure 73 Repair Volume

Choose the new drive (Disk 1) to replace the failed drive in the array, as shown in Figure 74:

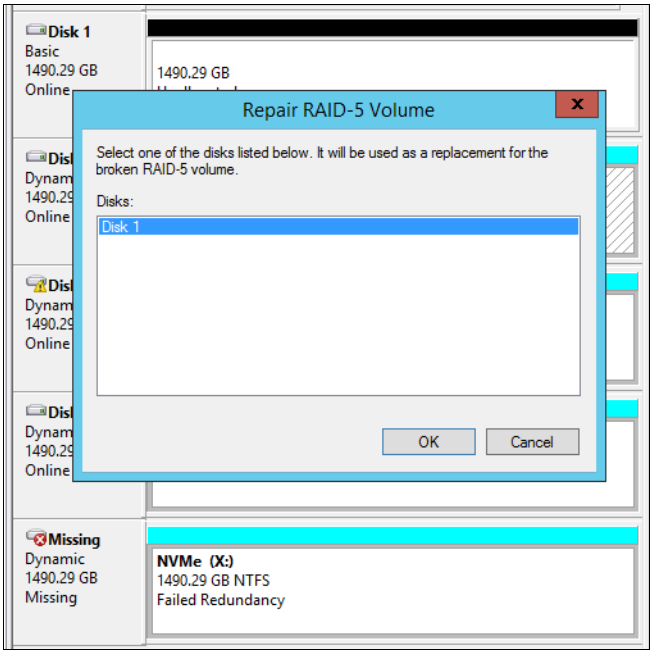


Figure 74 Select a new drive for the array

Confirm the action:

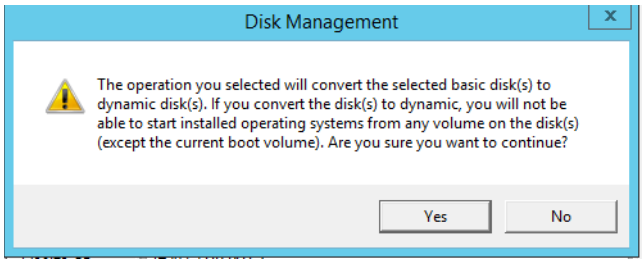


Figure 75 Confirm the action

4. Check the synchronization status

   When new drive is added to the array, the synchronization process will run automatically. You can check synchronization status of the array in **Disk Manager**, as shown in Figure 76:

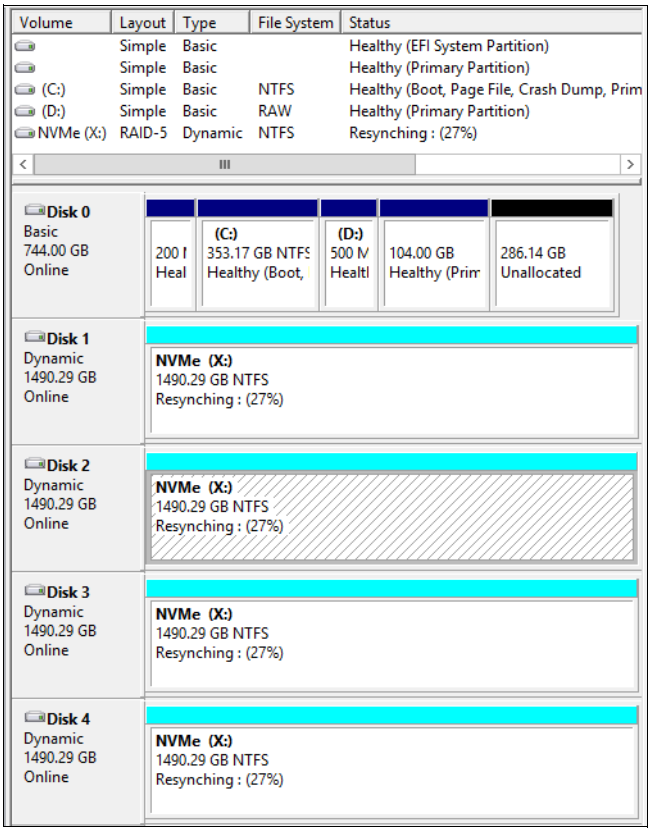


Figure 76 Resyncing process

When resyncing process is completed, the array will become redundant and fully operational again.

NVMe drives endurance analyzing

In this section we talk about SSD endurance. By saying “SSD endurance” we mean how much data you can write to SSD keeping drive healthy and reliable. There are several factors which affect SSD endurance:

- NAND techology used in the solid state drive
- Spare area capacity
- Workload

Workload is probably the most important factor. Workload patterns like random big block writes, random small writes or sequential writes affect on SSD endurance in different ways.

In this section, we cover several tools which allow you to estimate or calculate the endurance of your NVMe drives.

Lenovo SSD Wear Gauge CLI Utility

Lenovo SSD Wear Gauge CLI Utility is a simple tool which allows you to get status of supported SSDs including NVMe drives of all three vendors, Intel, Toshiba and Samsung.

You can download this tool from Lenovo support site:

https://datacentersupport.lenovo.com

Lenovo SSD Wear Gauge CLI Utility is available for the following operating systems:

- Windows 2008 (64-bit)
- Windows 2012
- Windows 2012 R2
- Windows 2016
- Red Hat Enterprise Linux 6
- Red Hat Enterprise Linux 7
- SUSE Enterprise Linux 11
- SUSE Enterprise Linux 12
- VMware ESXi 5.x
- VMware ESXi 6.0
- VMware ESXi 6.5

To demonstate how to use Lenovo SSD Wear Gauge CLI Utility we downloaded the version for Linux.

When Lenovo SSD Wear Gauge CLI Utility is downloaded, put it to some directory (`/tmp` for example), make it executable and run it, as shown in the following Figure 77:

```
[root@localhost ~]# /tmp/lnvgy_utl_ssd_7.16-17.05.09p_linux_x86-64.bin -s
Running under OS in online mode.
Running in 64 mode.

(c)Copyright Lenovo 2016.
Portions (c)Copyright IBM Corporation.

SSDCLI -- Display SMART Info                              v:7.0.4[Wed Apr 26 10:06:33
2017]
------------------------------------------------------------------------------
1   PN:SSS7A06660-01GR657    SN:A6X52099 FW:CG35
    Number bytes written to SSD: 77.6GB
    Number bytes supported by warranty: 2733000GB
    Life Remaining Gauge: 100%
    SSD temperature:33(c)      Spec Max: 70(c)
    PFA trip: No
    Warranty Exceed:No

2   PN:SSD0L20444-00LF425 SN:SW8AL7YS FW:3008T11H
    Number bytes written to SSD: 77.5GB
    Number bytes supported by warranty: 17520000GB
    Life Remaining Gauge: 100%
    SSD temperature:32(c)      Spec Max: 70(c)
    PFA trip: No
    Warranty Exceed:No

3   PN:SSS7A06660-01GR657    SN:A6X52094 FW:CG35
    Number bytes written to SSD: 77.5GB
    Number bytes supported by warranty: 2733000GB
    Life Remaining Gauge: 100%
    SSD temperature:32(c)      Spec Max: 70(c)
    PFA trip: No
    Warranty Exceed:No

4   PN:00D8457-           SN:60120018 FW:8DV1LP11
    Number bytes written to SSD: 77.5GB
    Number bytes supported by warranty: 36500000GB
    Life Remaining Gauge: 100%
    SSD temperature:32(c)      Spec Max: 70(c)
    PFA trip: No
    Warranty Exceed:No

5   PN:00D8452-           SN:5170000D FW:8DV1LP11
    Number bytes written to SSD: 5878.8GB
    Number bytes supported by warranty: 29200000GB
    Life Remaining Gauge: 100%
    SSD temperature:29(c)      Spec Max: 70(c)
    PFA trip: No
    Warranty Exceed:No


    5 Device(s) Found(SATA:0  SAS:0  NVME:5)
[root@localhost ~]#
```

Figure 77 Lenovo SSD Wear Gauge CLI Utility output

As you can see from the output, you can get the following parameters for every single NVMe drive:

- Number of bytes supported by warranty: How much data can be written to the drive based on the stated endurance of the drive.
- Number of bytes written to SSD: How much data has already been written to the drive.
- Life Remaining Gauge: NVMe drive health represented in percent. Note that the “life” gauge is determined using several parameters, not just using the number of bytes written and supported by warranty.

Intel SSD Data Center Tool

Intel also provides a specific software tool for its SSDs including modern NVMe drives. This tool has many different features and options, but in this section we cover only one of them, SSD Endurance Analyzer feature.

Briefly speaking, SSD Endurance Analyzer feature allows you to calculate drive life expectancy depending on your current workload. All you need to do is just to enable Endurance Analyzer feature on your NVMe drive under workload, wait some time (one hour is minimum, at least several hours are recommended) and then read Endurance Analyzer value calculated depending on your workload. This value represents drive’s life expectancy of your particular Intel NVMe drive in years.

You can download this tool from Intel web-site following the next link:

https://downloadcenter.intel.com/download/26749/Intel-SSD-Data-Center-Tool?v=t

The following operating systems are supported:

- Red Hat Enterprise Linux 6.5 and higher
- Red Hat Enterprise Linux 7.0 and higher
- SUSE Linux Enterprise Server 11 SP3 and higher
- SUSE Linux Enterprise Server 12 SP1 and higher
- Windows Server 2016
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2
- VMware ESXi 5.0 and higher

Note: The Intel provided driver must be installed in order to use Data Center tool for Windows. The tool will not work with the in-box Windows NVMe driver. You can download the required driver from the Intel web site:

https://downloadcenter.intel.com/download/26833/Intel-SSD-Data-Center-Family-for-NVMe-Drivers?v=t

We use Linux version of Data Center tool to demonstate Endurance Analyzer usage. Use the following procedure to get Intel NVMe drive's life expectancy:

1. Install Intel Data Center software

   Install the downloaded RPM-package, as shown in the following Figure 78:

```
[root@localhost ~]# rpm -Uhv /tmp/isdct-3.0.4.400-17.x86_64.rpm
warning: /tmp/isdct-3.0.4.400-17.x86_64.rpm: Header V4 RSA/SHA256 Signature, key ID
5d1bc6fe: NOKEY
Preparing...                          ################################# [100%]
Updating / installing...
   1:isdct-3.0.4.400-17               ################################# [100%]
[root@localhost ~]#
```

Figure 78 Intel Data Center tool installation process

2. Reset all timed workload SMART attributes

   You should flush all required SMART counters before you run your workload. To do that, run the following command, as shown in Figure 79:

```
[root@localhost ~]# /usr/bin/isdct -intelssd 0 EnduranceAnalyzer=reset
```

Figure 79 SMART attributes reset

   The addtitional parameter `-intelssd` is required to define the index number of your Intel NVMe drive. The index number can be 0, 1, 2 etc.

3. Apply your workload at least for 60 minutes
4. Check for the values of EnduranceAnalyzer parameter

   Run the following command to get your Intel NVMe drive's life expectancy, as shown in Figure 80:

```
[root@localhost ~]# isdct show -d EnduranceAnalyzer -intelssd 0

- Intel SSD DC P3700 Series CVFT5170000D1P6DGN -

EnduranceAnalyzer : 4943.71 years

[root@localhost ~]#
```

Figure 80 EnduranceAnalyzer value

In the previous example the IO workload on the Intel P3700 NVMe drive was not very heavy. So, the forecast of expected lifetime is almost 5000 years!

Related publications and links

Consult these documents for more information:

- Lenovo Press product guides on Lenovo NVMe SSDs

  https://lenovopress.com/servers/options/drives#term=nvme&rt=product-guide

- Comparing the Effect of PCIe Host Connections on NVMe Drive Performance

  https://lenovopress.com/lp0865
- VMware ESXi 5.5 NVMe driver

  https://my.vmware.com/web/vmware/details?productId=353&downloadGroup=DT-ESXI55-VMWARE-NVME-10E030-1VMW
- Intel paper *Hot-Plug Capability of NVMe SSDs in Server Platforms*:

  http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/333596-hot-plug-capability-nvme-ssds-paper.pdf

Change history

- 11 June 2018:
  - Added servers to Table 1 on page 4
- 7 July 2017:
  - Added content on the new Samsung and Toshiba NVMe drives
  - Added a section on Software RAID initialization for Windows
  - Added a section on analyzing NVMe drives endurance
- 23 September 2016:
  - System x3650 M5 now supports informed hot-insertion and hot-removal, Table 1 on page 4.

Authors

This paper was produced by the following team of specialists:

Ilya Solovyev is a Senior Technical Consultant in Lenovo Professional Services team, based in Moscow. He currently provides technical consulting services for Lenovo hardware and solutions. His areas of expertise include Linux systems, virtualization, cloud, SAP HANA and HPC solutions. Ilya is a certified Red Hat Linux and SUSE Linux Administrator, he has a Bachelor degree in Mathematics.

David Watts is a Senior IT Consultant and the program lead for Lenovo Press. He manages residencies and produces pre-sale and post-sale technical publications for hardware and software topics that are related to System x, ThinkServer, Flex System, and BladeCenter® servers. He has authored over 300 books and papers. David has worked in the IT industry, both in the U.S. and Australia, since 1989, and is currently based in Morrisville, North Carolina. David holds a Bachelor of Engineering degree from the University of Queensland (Australia).

Thanks to the following people for their contributions to this project:

- Ron Birtles
- John Encizo
- Mike French
- Shane Carroll
- Steven Livaccari

Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area. Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service.

Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Lenovo (United States), Inc.
1009 Think Place - Building One
Morrisville, NC 27560
U.S.A.
Attention: Lenovo Director of Licensing

LENOVO PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary.

Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk.

Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This document was created or updated on June 11, 2018.

Send us your comments via the **Rate & Provide Feedback** form found at
http://lenovopress.com/lp0508

Trademarks

Lenovo, the Lenovo logo, and For Those Who Do are trademarks or registered trademarks of Lenovo in the United States, other countries, or both. These and other Lenovo trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by Lenovo at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of Lenovo trademarks is available on the Web at http://www.lenovo.com/legal/copytrade.html.

The following terms are trademarks of Lenovo in the United States, other countries, or both:

BladeCenter®
Flex System™
Lenovo®
Lenovo(logo)®
System x®
ThinkServer®
ThinkSystem™

The following terms are trademarks of other companies:

Intel, and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows Server, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.