



## 基于光纤通道的 **NVMe** 设置 E-Series storage systems

NetApp  
January 20, 2026

# 目录

基于光纤通道的 NVMe 设置 .....	1
验证Linux配置支持并查看E系列(基于FC的NVMe)中的限制 .....	1
验证是否支持 Linux 配置 .....	1
查看基于 FC 的 NVMe 的限制 .....	1
在E系列中使用DHCP配置IP地址- Linux (基于FC的NVMe) .....	1
安装适用于SMCli(11.53或更早版本)的SANtricity Storage Manager - Linux (基于FC的NVMe) .....	2
使用SANtricity系统管理器配置存储—Linux (基于FC的NVMe) .....	3
在E系列- Linux (基于FC的NVMe)中配置FC交换机 .....	4
在E系列- Linux中的主机上设置基于FC的NVMe启动程序 .....	5
使用SANtricity系统管理器创建主机—Linux (基于FC的NVMe) .....	7
使用SANtricity系统管理器分配卷—Linux (基于NVMe的FC) .....	8
在E系列- Linux (基于FC的NVMe)中显示主机可见的卷 .....	9
在E系列- Linux (基于FC的NVMe)中的主机上设置故障转移 .....	10
为 SLES 12 启用设备映射器多路径 (DMMP) .....	11
为RHEL 8设置原生 NVMe多路径 .....	11
访问E系列中虚拟设备目标的NVMe卷—Linux (基于FC的NVMe) .....	12
虚拟设备是 I/O 目标 .....	12
示例 .....	12
访问E系列中物理NVMe设备目标的NVMe卷—Linux (基于FC的NVMe) .....	14
物理 NVMe 设备是 I/O 目标 .....	14
在 E 系列中创建文件系统 - SLES 12 (NVMe over FC) .....	16
在 E 系列 Linux RHEL 8、RHEL 9、RHEL 10、SLES 15 和 SLES 16 中创建文件系统 (NVMe over FC) .....	17
在E系列- Linux (基于FC的NVMe)中验证主机上的存储访问 .....	18
在E系列- Linux中记录基于FC的NVMe配置 .....	18
直连拓扑 .....	19
交换机连接拓扑 .....	19
主机标识符 .....	20
目标 NQN .....	20
目标 NQN .....	20
正在映射主机名 .....	21

# 基于光纤通道的 NVMe 设置

## 验证Linux配置支持并查看E系列(基于FC的NVMe)中的限制

首先，您应验证 Linux 配置是否受支持，并查看控制器，主机和恢复限制。

### 验证是否支持 Linux 配置

为了确保操作可靠，您需要创建一个实施计划，然后使用 NetApp 互操作性表工具（IMT）验证是否支持整个配置。

#### 步骤

1. 转至 "[NetApp 互操作性表工具](#)"。
2. 单击 \* 解决方案 Search\* 图块。
3. 在菜单：协议 [SAN 主机] 区域中，单击 \* E 系列 SAN 主机 \* 旁边的 \* 添加 \* 按钮。
4. 单击 \* 查看细化搜索条件 \*。

此时将显示细化搜索条件部分。在本节中，您可以选择适用的协议以及其他配置标准，例如操作系统，NetApp 操作系统和主机多路径驱动程序。

5. 选择您知道配置所需的条件，然后查看适用的兼容配置要素。
6. 根据需要，根据工具中的规定更新操作系统和协议。

单击右页面箭头可在查看支持的配置页面上访问选定配置的详细信息。

### 查看基于 FC 的 NVMe 的限制

在使用基于光纤通道的NVMe之前、请参见 "[NetApp 互操作性表工具](#)" 查看最新的控制器、主机和恢复限制。

#### 存储和灾难恢复限制

- 不支持异步和同步镜像。
- 不支持精简配置（创建精简卷）。

## 在E系列中使用DHCP配置IP地址- Linux (基于FC的NVMe)

要配置管理工作站与存储阵列之间的通信，请使用动态主机配置协议（DHCP）提供 IP 地址。

#### 开始之前

确保您已具备以下条件：

- 与存储管理端口安装和配置在同一子网上的 DHCP 服务器。

#### 关于此任务

每个存储阵列都有一个控制器（单工）或两个控制器（双工），每个控制器有两个存储管理端口。每个管理端口都将分配一个 IP 地址。

以下说明是指具有两个控制器的存储阵列（双工配置）。

#### 步骤

1. 如果尚未连接以太网缆线，请将其连接到管理工作站以及每个控制器（A 和 B）上的管理端口 1。

DHCP 服务器会为每个控制器的端口 1 分配一个 IP 地址。



请勿在任一控制器上使用管理端口 2。端口 2 保留供 NetApp 技术人员使用。



如果断开并重新连接以太网缆线，或者存储阵列已重新启动，则 DHCP 会重新分配 IP 地址。此过程会一直进行，直到配置了静态 IP 地址为止。建议您避免断开电缆连接或重新启动阵列。

如果存储阵列无法在 30 秒内获取 DHCP 分配的 IP 地址，则会设置以下默认 IP 地址：

- 控制器 A，端口 1：169.254.128.101
- 控制器 B，端口 1：169.254.128.102
- 子网掩码：255.255.0.0

2. 找到每个控制器背面的 MAC 地址标签，然后向网络管理员提供每个控制器端口 1 的 MAC 地址。

网络管理员需要使用 MAC 地址来确定每个控制器的 IP 地址。要通过浏览器连接到存储系统，您需要使用这些 IP 地址。

## 安装适用于SMcli(11.53或更早版本)的SANtricity Storage Manager - Linux (基于FC的NVMe)

如果您使用的是 SANtricity 软件 11.53 或更早版本，则可以在管理工作站上安装 SANtricity 存储管理器软件，以帮助管理阵列。

SANtricity 存储管理器包括用于执行其他管理任务的命令行界面（CLI）以及用于通过 I/O 路径将主机配置信息推送到存储阵列控制器的主机上下文代理。



如果您使用的是 SANtricity 软件 11.60 及更高版本，则无需执行以下步骤。SANtricity 安全命令行界面（SMcli）包含在 SANtricity 操作系统中，可通过 SANtricity 系统管理器下载。有关如何通过 SANtricity 系统管理器下载 SMcli 的详细信息，请参阅 ["下载 SANtricity 系统管理器联机帮助下的命令行界面\(CLI\)主题"](#)



从 SANtricity 软件 11.80.1 版开始，不再支持主机上下文代理。

#### 开始之前

确保您已具备以下条件：

- SANtricity 软件 11.53 或更早版本。

- 正确的管理员或超级用户权限。
- SANtricity 存储管理器客户端的系统，最低要求如下：
  - \* RAM \* : 2 GB 用于 Java Runtime Engine
  - \* 磁盘空间 \* : 5 GB
  - \* 操作系统 / 架构 \* : 有关确定支持的操作系统版本和架构的指导, 请访问 ["NetApp 支持"](#)。从 \* 下载 \* 选项卡中, 转到菜单: 下载 [E 系列 SANtricity 存储管理器]。

#### 关于此任务

此任务介绍如何在 Windows 和 Linux 操作系统平台上安装 SANtricity 存储管理器, 因为在数据主机使用 Linux 时, Windows 和 Linux 都是通用的管理工作站平台。

#### 步骤

1. 从下载 SANtricity 软件版本 ["NetApp 支持"](#)。从 \* 下载 \* 选项卡中, 转到菜单: 下载 [E 系列 SANtricity 存储管理器]。
2. 运行 SANtricity 安装程序。

Windows	Linux
双击 SMIA* .exe 安装包以开始安装。	a. 转到 SMIA* .bin 安装包所在的目录。 b. 如果临时挂载点没有执行权限, 请设置 IATEMPDIR 变量。示例: IATEMPDIR=/root ./AMI-LINUX64-11.25.0A00.0002.bin c. 运行 <code>chmod +x sMIA* .bin</code> 命令为文件授予执行权限。 d. 运行 <code>./SMIA* .bin</code> 命令启动安装程序。

3. 使用安装向导在管理工作站上安装软件。

## 使用SANtricity系统管理器配置存储—Linux (基于FC的NVMe)

要配置存储阵列, 您可以使用 SANtricity 系统管理器中的设置向导。

SANtricity System Manager 是一个基于 Web 的界面, 嵌入在每个控制器上。要访问用户界面, 请将浏览器指向控制器的 IP 地址。设置向导可帮助您开始进行系统配置。

#### 开始之前

确保您已具备以下条件:

- 带外管理。
- 用于访问 SANtricity 系统管理器的管理工作站, 其中包括以下浏览器之一:

浏览器	最低版本
Google Chrome	81.

浏览器	最低版本
Microsoft Edge	90
Mozilla Firefox	80
Safari	14.

#### 关于此任务

在打开 System Manager 或刷新浏览器且至少满足以下条件之一时，向导将自动重新启动：

- 未检测到池和卷组。
- 未检测到工作负载。
- 未配置任何通知。

#### 步骤

1. 在浏览器中，输入以下 URL：`https://<DomainNameOrIPAddress>`

`IPAddress` 是一个存储阵列控制器的地址。

首次在尚未配置的阵列上打开 SANtricity 系统管理器时，将显示设置管理员密码提示符。基于角色的访问管理可配置四个本地角色：管理员，支持，安全性和监控。后三个角色具有无法猜测的随机密码。为管理员角色设置密码后，您可以使用管理员凭据更改所有密码。有关四个本地用户角色的详细信息，请参见 SANtricity System Manager 用户界面中提供的联机帮助。

2. 在设置管理员密码和确认密码字段中输入管理员角色的 System Manager 密码，然后单击 \* 设置密码 \*。

如果未配置池，卷组，工作负载或通知，则设置向导将启动。

3. 使用设置向导执行以下任务：

- \* 验证硬件（控制器和驱动器） \* —验证存储阵列中的控制器和驱动器数量。为阵列分配一个名称。
- \* 验证主机和操作系统 \* - 验证存储阵列可以访问的主机和操作系统类型。
- \* 接受池 \* - 接受快速安装方法建议的池配置。池是一组逻辑驱动器。
- \* 配置警报 \* - 允许 System Manager 在存储阵列出现问题时自动接收通知。
- \* 启用 AutoSupport \* - 自动监控存储阵列的运行状况，并将派单发送给技术支持。

4. 如果尚未创建卷，请转到菜单：Storage[ 卷 > 创建 > 卷 ] 创建一个卷。

有关详细信息，请参见 SANtricity System Manager 的联机帮助。

## 在E系列- Linux (基于FC的NVMe)中配置FC交换机

通过配置（分区）光纤通道（FC）交换机，主机可以连接到存储阵列并限制路径数。您可以使用交换机的管理界面对交换机进行分区。

开始之前

确保您已具备以下条件：

- 交换机的管理员凭据。
- 每个主机启动程序端口以及连接到交换机的每个控制器目标端口的 WWPN。（使用 HBA 实用程序进行发现。）

关于此任务

有关对交换机进行分区的详细信息，请参见交换机供应商的文档。

每个启动程序端口都必须位于一个单独的分区中，并具有所有相应的目标端口。

步骤

1. 登录到 FC 交换机管理程序，然后选择分区配置选项。
2. 创建一个新分区，其中包含第一个主机启动程序端口，也包括与启动程序连接到同一 FC 交换机的所有目标端口。
3. 为交换机中的每个 FC 主机启动程序端口创建其他分区。
4. 保存分区，然后激活新的分区配置。

## 在E系列- Linux中的主机上设置基于FC的NVMe启动程序

光纤通道环境中的 NVMe 启动程序配置包括安装和配置 NVMe-CLI 软件包以及在主机上启用 NVMe/FC 启动程序。

关于此任务

以下步骤适用于使用 Broadcom Emulex 或 QLogic NVMe/FC 支持的 FC HBA 的 RHEL 8、RHEL 9、SLES 12 和 SLES 15。有关支持哪些 OS 或 HBA 版本的更多信息，请查阅 ["NetApp 互操作性表工具"](#)。

步骤

1. 安装 NVMe-CLI 软件包：
  - SLES 12 或 SLES 15\*

```
# zypper install nvme-cli
```

### RHEL 8 或 RHEL 9

```
# yum install nvme-cli
```

+

- a. 对于 Qlogic，请在安装 Broadcom NVMe/FC 自动连接脚本后修改 `/lib/systemd/system/nvme-fc-boot-connections.service` 以包含以下内容：

```
[Unit]
Description=Auto-connect to subsystems on FC-NVME devices found
during boot

[Service]
Type=oneshot
ExecStart=/bin/sh -c "echo add >
/sys/class/fc/fc_udev_device/nvme_discovery"

[Install]
WantedBy=default.target
```

2. 启用并启动 `nvme-fc-boot-connections` 服务。

```
systemctl enable nvme-fc-boot-connections.service
```

```
systemctl start nvme-fc-boot-connections.service
```

◦ Emulex HBA 的主机端设置： \*



以下步骤仅适用于 Emulex HBA。

1. 将 `lpfc_enable_FC4_type` 设置为 3 以启用 SLES12 SP4 作为 NVMe/FC 启动程序。

```
# cat /etc/modprobe.d/lpfc.conf
options lpfc lpfc_enable_fc4_type=3
```

2. 重新构建 `initrd` 以获取 Emulex 更改和启动参数更改。

```
# dracut --force
```

3. 重新启动主机以加载对 `lpfc` 驱动程序所做的更改。

```
# reboot
```

主机将重新启动，并在主机上启用 NVMe/FC 启动程序。



完成主机端设置后，将自动通过光纤通道端口连接 NVMe。

# 使用SANtricity系统管理器创建主机—Linux (基于FC的NVMe)

您可以使用 SANtricity 系统管理器定义向存储阵列发送数据的主机。定义主机是让存储阵列知道连接了哪些主机并允许对卷进行 I/O 访问所需的步骤之一。

关于此任务

定义主机时，请牢记以下准则：

- 您必须定义与主机关联的主机标识符端口。
- 请确保提供与主机分配的系统名称相同的名称。
- 如果您选择的名称已在使用中，则此操作不会成功。
- 名称长度不能超过 30 个字符。

步骤

1. 选择菜单： Storage[Hosts] 。
2. 单击菜单： 创建 [ 主机 ] 。

此时将显示创建主机对话框。

3. 根据需要选择主机设置。

正在设置 ...	Description
Name	键入新主机的名称。
主机操作系统类型	从下拉列表中选择以下选项之一： <ul style="list-style-type: none"><li>• 适用于 SANtricity 11.60 及更高版本的 * Linux *</li><li>• 对于 SANtricity 11.60 之前的版本， * Linux DM-MP （内核 3.10 或更高版本） *</li></ul>
主机接口类型	选择要使用的主机接口类型。如果您配置的阵列只有一个可用的主机接口类型，则可能无法选择此设置。

正在设置 ...	Description
主机端口	<p>执行以下操作之一：</p> <ul style="list-style-type: none"><li>• * 选择 I/O 接口 *</li></ul> <p>如果主机端口已登录，则可以从列表中选择主机端口标识符。这是建议的方法。</p> <ul style="list-style-type: none"><li>• * 手动添加 *</li></ul> <p>如果主机端口尚未登录，请查看主机上的 <code>/etc/nve/hostnqn</code> 以查找 <code>hostnqn</code> 标识符并将其与主机定义关联。</p> <p>您可以手动输入主机端口标识符，或者将它们从 <code>/etc/nve/hostnqn</code> 文件（一次一个）复制 / 粘贴到 * 主机端口 * 字段中。</p> <p>您必须一次添加一个主机端口标识符才能将其与主机关联，但您可以继续选择与主机关联的任意数量的标识符。每个标识符都会显示在 * 主机端口 * 字段中。如有必要，您还可以通过选择标识符旁边的 * X * 来删除该标识符。</p>

4. 单击 \* 创建 \*。

结果

成功创建主机后， SANtricity 系统管理器会为主机配置的每个主机端口创建一个默认名称。

默认别名为 ``< 主机名端口号 >``。例如，为 主机 `ipt` 创建的第一个端口的默认别名是 `ipt_1`。

# 使用SANtricity系统管理器分配卷—Linux (基于NVMe的FC)

您必须将卷（命名空间）分配给主机或主机集群，以便用于 I/O 操作。此分配可授予主机或主机集群对存储阵列中一个或多个命名空间的访问权限。

关于此任务

分配卷时，请记住以下准则：

- 一次只能将卷分配给一个主机或主机集群。
- 分配的卷在存储阵列中的控制器之间共享。
- 主机或主机集群不能使用同一命名空间 ID （ NSID ） 两次来访问卷。您必须使用唯一的 NSID 。

在以下情况下，分配卷失败：

- 已分配所有卷。
- 此卷已分配给其他主机或主机集群。

在以下情况下，无法分配卷：

- 不存在有效的主机或主机集群。
- 已定义所有卷分配。

此时将显示所有未分配的卷，但具有或不具有数据保证（Data Assurance，DA）的主机的功能如下所示：

- 对于支持 DA 的主机，您可以选择已启用 DA 或未启用 DA 的卷。
- 对于不支持 DA 的主机，如果选择的卷已启用 DA，则会显示一条警告，指出系统必须先自动关闭卷上的 DA，然后才能将卷分配给主机。

#### 步骤

1. 选择菜单：Storage[Hosts]。
2. 选择要将卷分配到的主机或主机集群，然后单击 \* 分配卷 \*。

此时将显示一个对话框，其中列出了可分配的所有卷。您可以对任意列进行排序或在 \* 筛选器 \* 框中键入内容，以便于查找特定卷。

3. 选中要分配的每个卷旁边的复选框，或者选中表标题中的复选框以选择所有卷。
4. 单击 \* 分配 \* 以完成此操作。

#### 结果

将一个或多个卷成功分配给主机或主机集群后，系统将执行以下操作：

- 分配的卷将接收下一个可用的 NSID。主机使用 NSID 访问卷。
- 用户提供的卷名称将显示在与主机关联的卷列表中。

## 在E系列- Linux (基于FC的NVMe)中显示主机可见的卷

您可以使用 SMdevices 工具查看主机上当前可见的卷。此工具是 NVMe-CLI 软件包的一部分，可用作 `nvme list` 命令的替代方法。

要查看有关 E 系列卷的每个 NVMe 路径的信息，请使用 `nvme netapp smdevices (-o <format>]` 命令。

输出 ``<format>`` 可以是 `normal`（如果不使用 `-o`，则为默认值），`column` 或 `json`。

```
# nvme netapp smdevices
/dev/nvme1n1, Array Name ICTM0706SYS04, Volume Name NVMe2, NSID 1, Volume
ID 000015bd5903df4a00a0980000af4462, Controller A, Access State unknown,
2.15GB
/dev/nvme1n2, Array Name ICTM0706SYS04, Volume Name NVMe3, NSID 2, Volume
ID 000015c05903e24000a0980000af4462, Controller A, Access State unknown,
2.15GB
/dev/nvme1n3, Array Name ICTM0706SYS04, Volume Name NVMe4, NSID 4, Volume
ID 00001bb0593a46f400a0980000af4462, Controller A, Access State unknown,
2.15GB
/dev/nvme1n4, Array Name ICTM0706SYS04, Volume Name NVMe6, NSID 6, Volume
ID 00001696593b424b00a0980000af4112, Controller A, Access State unknown,
2.15GB
/dev/nvme2n1, Array Name ICTM0706SYS04, Volume Name NVMe2, NSID 1, Volume
ID 000015bd5903df4a00a0980000af4462, Controller B, Access State unknown,
2.15GB
/dev/nvme2n2, Array Name ICTM0706SYS04, Volume Name NVMe3, NSID 2, Volume
ID 000015c05903e24000a0980000af4462, Controller B, Access State unknown,
2.15GB
/dev/nvme2n3, Array Name ICTM0706SYS04, Volume Name NVMe4, NSID 4, Volume
ID 00001bb0593a46f400a0980000af4462, Controller B, Access State unknown,
2.15GB
/dev/nvme2n4, Array Name ICTM0706SYS04, Volume Name NVMe6, NSID 6, Volume
ID 00001696593b424b00a0980000af4112, Controller B, Access State unknown,
2.15GB
```

## 在E系列- Linux (基于FC的NVMe)中的主机上设置故障转移

要为存储阵列提供冗余路径，您可以将主机配置为运行故障转移。

开始之前

您必须在系统上安装所需的软件包。

- 对于 Red Hat (RHEL) 主机，运行 `rpm -q device-mapper-multipath` 验证软件包是否已安装
- 对于 SLES 主机，请运行 `rpm -q multipath-tools` 来验证软件包是否已安装

SLES 12 use Device Mapper Multipath (DMMP) for multipathing when using NVMe over Fibre Channel. RHEL 8, RHEL 9, RHEL 10, SLES 15 and SLES 16 use a built-in Native NVMe Failover. Depending on which OS you are running, some additional configuration of multipath is required to get it running properly.

## 为 SLES 12 启用设备映射器多路径 (DMMP)

默认情况下，SLES 中禁用 DM-MP。完成以下步骤以在主机上启用 DM-MP 组件。

### 步骤

1. 将 NVMe E 系列设备条目添加到 `/etc/multipath.conf` 文件的 `devices` 部分，如以下示例所示：

```
devices {
    device {
        vendor "NVME"
        product "NetApp E-Series*"
        path_grouping_policy group_by_prio
        failback immediate
        no_path_retry 30
    }
}
```

2. 将 `multipathd` 配置为在系统启动时启动。

```
# systemctl enable multipathd
```

3. 如果当前未运行，请启动 `multipathd`。

```
# systemctl start multipathd
```

4. 验证 `multipathd` 的状态，以确保其处于活动状态且正在运行：

```
# systemctl status multipathd
```

## 为 RHEL 8 设置原生 NVMe 多路径

### 关于此任务

默认情况下，原生 NVMe 多路径在 RHEL 8 中处于禁用状态，必须使用以下步骤启用。

### 步骤

1. 设置 `modprobe` 规则以启用原生 NVMe 多路径。

```
# echo "options nvme_core multipath=y" >> /etc/modprobe.d/50-nvme_core.conf
```

2. 使用新的 `modprobe` 参数 `remake initramfs`。

```
# dracut -f
```

3. 重新启动服务器以在启用原生 NVMe 多路径的情况下启动它

```
# reboot
```

4. 确认主机重新启动后已启用原生 NVMe 多路径。

```
# cat /sys/module/nvme_core/parameters/multipath
```

- a. 如果命令输出为 `N`，则原生 NVMe 多路径仍处于禁用状态。
- b. 如果命令输出为 `Y`，则原生 NVMe 多路径将处于启用状态，您发现的任何 NVMe 设备将使用该功能。



对于 SLES 15、SLES 16、RHEL 9 和 RHEL 10，默认情况下启用原生 NVMe 多路径，无需额外配置。

## 访问E系列中虚拟设备目标的NVMe卷—Linux (基于FC的NVMe)

您可以根据所使用的操作系统（以及扩展多路径方法）配置定向到设备目标的 I/O。

对于 SLES 12，I/O 由 Linux 主机定向到虚拟设备目标。DM-MP 管理这些虚拟目标背后的物理路径。

### 虚拟设备是 I/O 目标

确保仅对 DM-MP 创建的虚拟设备运行 I/O，而不对物理设备路径运行 I/O。如果对物理路径运行 I/O，DM-MP 将无法管理故障转移事件，并且 I/O 将失败。

您可以通过 `/dev/mapper` 中的 `dm device` 或 `symlink` 访问这些块设备；例如：

```
/dev/dm-1  
/dev/mapper/eui.00001bc7593b7f5f00a0980000af4462
```

### 示例

以下 `nvme list` 命令的输出示例显示了主机节点名称及其与命名空间 ID 的关联。

NODE	SN	MODEL	NAMESPACE
/dev/nvme1n1	021648023072	NetApp E-Series	10
/dev/nvme1n2	021648023072	NetApp E-Series	11
/dev/nvme1n3	021648023072	NetApp E-Series	12
/dev/nvme1n4	021648023072	NetApp E-Series	13
/dev/nvme2n1	021648023151	NetApp E-Series	10
/dev/nvme2n2	021648023151	NetApp E-Series	11
/dev/nvme2n3	021648023151	NetApp E-Series	12
/dev/nvme2n4	021648023151	NetApp E-Series	13

列	Description
节点	<p>节点名称包括两部分：</p> <ul style="list-style-type: none"> <li>• 表示法 nvme1 表示控制器 A， nvme2 表示控制器 B</li> <li>• 表示法 n1， n2 等代表主机角度的命名空间标识符。这些标识符在表中重复出现，一次用于控制器 A，一次用于控制器 B</li> </ul>
命名空间	命名空间列列出了命名空间 ID （ NSID ），从存储阵列角度来看，命名空间 ID 是标识符。

在以下 `multipath -ll` 输出中，优化路径显示为 `prio` 值 50，而非优化路径显示为 `prio` 值 10。

Linux 操作系统会将 I/O 路由到显示为 `status=active` 的路径组，而列为 `status=enabled` 的路径组可用于故障转移。

```
eui.00001bc7593b7f500a0980000af4462 dm-0 NVME,NetApp E-Series
size=15G features='1 queue_if_no_path' hwhandler='0' wp=rw
|+- policy='service-time 0' prio=50 status=active
|  `- #:#:#:# nvme1n1 259:5 active ready running
`+- policy='service-time 0' prio=10 status=enabled
   `- #:#:#:# nvme2n1 259:9 active ready running

eui.00001bc7593b7f5f00a0980000af4462 dm-0 NVME,NetApp E-Series
size=15G features='1 queue_if_no_path' hwhandler='0' wp=rw
|+- policy='service-time 0' prio=0 status=enabled
|  `- #:#:#:# nvme1n1 259:5 failed faulty running
`+- policy='service-time 0' prio=10 status=active
   `- #:#:#:# nvme2n1 259:9 active ready running
```

行项目	Description
policy='service-time 0' prio=50 status=active	此行和以下行显示，nvme1n1 是 NSID 为 10 的命名空间，它在 prio value of 50 和 ststatus value of active 的路径上进行了优化。  此命名空间属于控制器 A
policy='service-time 0' prio=10 status=enabled	此行显示了命名空间 10 的故障转移路径，其中 prio value of 10 and a stStatus value of enabled。此时，I/O 不会定向到此路径上的命名空间。  此命名空间属于控制器 B
policy='service-time 0' prio=0 status=enabled	此示例显示了控制器 A 正在重新启动时不同时间点的 multipath -ll 输出。命名空间 10 的路径显示为 failed faulty running，其中 prio 值为 0，s 状态 值为 enabled。
policy='service-time 0' prio=10 status=active	请注意，active 路径是指 nvme2，因此 I/O 将通过此路径定向到控制器 B

## 访问E系列中物理NVMe设备目标的NVMe卷—Linux (基于FC的NVMe)

您可以根据所使用的操作系统（以及扩展多路径方法）配置定向到设备目标的 I/O。

对于RHEL 8、RHEL 9和SLES 15、Linux主机会将I/O定向到物理NVMe设备目标。原生 NVMe 多路径解决方案用于管理主机显示的单个表面上物理设备的底层物理路径。

### 物理 NVMe 设备是 I/O 目标

最佳做法是对中的链路运行I/O /dev/disk/by-id/nvme-eui.[uuid#] 而不是直接连接到物理NVMe设备路径 /dev/nvme[sys#]n[id#]。可以使用以下命令找到这两个位置之间的链接：

```
# ls /dev/disk/by-id/ -l
lrwxrwxrwx 1 root root 13 Oct 18 15:14 nvme-
eui.0000320f5cad32cf00a0980000af4112 -> ../../nvme0n1
```

I/O运行到 /dev/disk/by-id/nvme-eui.[uuid#] 将直接通过 /dev/nvme[sys#]n[id#] 它使用原生 NVMe多路径解决方案 对其下的所有路径进行虚拟化。

您可以运行以下命令来查看路径：

```
# nvme list-sys
```

示例输出：

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:5700.600a098000a522500000000589aa8a6
\
+- nvme0 rdma traddr=192.4.21.131 trsvcid=4420 live
+- nvme1 rdma traddr=192.4.22.141 trsvcid=4420 live
```

如果在使用 `nvme list-subsys` 命令时指定命名空间设备，则它会提供有关该命名空间路径的追加信息：

```
# nvme list-subsys /dev/nvme0n1
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:5700.600a098000af44620000000058d5dd96
\
+- nvme0 rdma traddr=192.168.130.101 trsvcid=4420 live non-optimized
+- nvme1 rdma traddr=192.168.131.101 trsvcid=4420 live non-optimized
+- nvme2 rdma traddr=192.168.130.102 trsvcid=4420 live optimized
+- nvme3 rdma traddr=192.168.131.102 trsvcid=4420 live optimized
```

此外，多路径命令还会挂接一些，以便您也可以通过这些命令查看原生故障转移的路径信息：

```
#multipath -ll
```



要查看路径信息，必须在 `/etc/multipath.conf` 中设置以下内容：

```
defaults {
    enable_foreign nvme
}
```



这在 RHEL 10 上将不再有效。它适用于 RHEL 9 及更早版本和 SLES 16 及更早版本。

示例输出：

```
eui.0000a0335c05d57a00a0980000a5229d [nvme]:nvme0n9 NVMe,Netapp E-
Series,08520001
size=4194304 features='n/a' hwhandler='ANA' wp=rw
|-+- policy='n/a' prio=50 status=optimized
|  `-- 0:0:1 nvme0c0n1 0:0 n/a optimized    live
`-+- policy='n/a' prio=10 status=non-optimized
   `-- 0:1:1 nvme0c1n1 0:0 n/a non-optimized    live
```

## 在 E 系列中创建文件系统 - SLES 12 (NVMe over FC)

对于 SLES 12, 您可以在所需的 dm 设备上创建文件系统并挂载该文件系统。

### 步骤

1. 运行 `multipath -ll` 命令以获取 `/dev/mapper/dm` 设备的列表。

```
# multipath -ll
```

此命令的结果显示了两个设备, `dm-19` 和 `dm-16` :

```
eui.00001ffe5a94ff8500a0980000af4444 dm-19 NVME,NetApp E-Series
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
|+- policy='service-time 0' prio=50 status=active
|  |- #:#:#:# nvme0n19 259:19  active ready running
|   `-- #:#:#:# nvme1n19 259:115 active ready running
`--+- policy='service-time 0' prio=10 status=enabled
    |- #:#:#:# nvme2n19 259:51  active ready running
    `-- #:#:#:# nvme3n19 259:83  active ready running
eui.00001fd25a94fef000a0980000af4444 dm-16 NVME,NetApp E-Series
size=16G features='1 queue_if_no_path' hwhandler='0' wp=rw
|+- policy='service-time 0' prio=50 status=active
|  |- #:#:#:# nvme0n16 259:16  active ready running
|   `-- #:#:#:# nvme1n16 259:112 active ready running
`--+- policy='service-time 0' prio=10 status=enabled
    |- #:#:#:# nvme2n16 259:48  active ready running
    `-- #:#:#:# nvme3n16 259:80  active ready running
```

2. 在分区上为每个 `/dev/mapper/eui-` 设备创建一个文件系统。

创建文件系统的方法因所选的文件系统而异。此示例显示了如何创建 `ext4` 文件系统。

```
# mkfs.ext4 /dev/mapper/dm-19
mke2fs 1.42.11 (09-Jul-2014)
Creating filesystem with 2620928 4k blocks and 655360 inodes
Filesystem UUID: 97f987e9-47b8-47f7-b434-bf3ebbe826d0
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

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

3. 创建一个文件夹以挂载新设备。

```
# mkdir /mnt/ext4
```

4. 挂载设备。

```
# mount /dev/mapper/eui.00001ffe5a94ff8500a0980000af4444 /mnt/ext4
```

## 在 E 系列 Linux RHEL 8、RHEL 9、RHEL 10、SLES 15 和 SLES 16 中创建文件系统（NVMe over FC）

对于 RHEL 8、RHEL 9、RHEL 10、SLES 15 和 SLES 16，您需要在本地 nvme 设备上创建文件系统并挂载该文件系统。

### 步骤

1. 运行multipath -ll命令以获取NVMe设备列表。

```
# multipath -ll
```

此命令的结果可用于查找关联的设备 /dev/disk/by-id/nvme-eui.[uuid#] 位置。以下示例为 /dev/disc/by-id/nvme-eui.000082dd5c05d39300a0980000a52225。

```
eui.000082dd5c05d39300a0980000a52225 [nvme]:nvme0n6 NVMe,NetApp E-
Series,08520000
size=4194304 features='n/a' hwhandler='ANA' wp=rw
|+- policy='n/a' prio=50 status=optimized
|  `-- 0:0:1 nvme0c0n1 0:0 n/a optimized      live
|+- policy='n/a' prio=50 status=optimized
|  `-- 0:1:1 nvme0c1n1 0:0 n/a optimized      live
|+- policy='n/a' prio=10 status=non-optimized
|  `-- 0:2:1 nvme0c2n1 0:0 n/a non-optimized live
`+- policy='n/a' prio=10 status=non-optimized
   `-- 0:3:1 nvme0c3n1 0:0 n/a non-optimized live
```

2. 使用位置在分区上为所需NVMe设备创建文件系统 /dev/disk/by-id/nvme-eui.[id#]。

创建文件系统的方法因所选的文件系统而异。此示例显示了如何创建 ext4 文件系统。

```
# mkfs.ext4 /dev/disk/by-id/nvme-eui.000082dd5c05d39300a0980000a52225
mke2fs 1.42.11 (22-Oct-2019)
Creating filesystem with 2620928 4k blocks and 655360 inodes
Filesystem UUID: 97f987e9-47b8-47f7-b434-bf3ebbe826d0
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

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

### 3. 创建一个文件夹以挂载新设备。

```
# mkdir /mnt/ext4
```

### 4. 挂载设备。

```
# mount /dev/disk/by-id/nvme-eui.000082dd5c05d39300a0980000a52225
/mnt/ext4
```

## 在E系列- Linux (基于FC的NVMe)中验证主机上的存储访问

在使用命名空间之前，您需要验证主机是否可以将数据写入命名空间并将其读回。

开始之前

确保您已具备以下条件：

- 已初始化的命名空间，使用文件系统进行格式化。

步骤

1. 在主机上，将一个或多个文件复制到磁盘的挂载点。
2. 将文件复制回原始磁盘上的其他文件夹。
3. 运行 diff 命令将复制的文件与原始文件进行比较。

完成后

删除复制的文件和文件夹。

## 在E系列- Linux中记录基于FC的NVMe配置

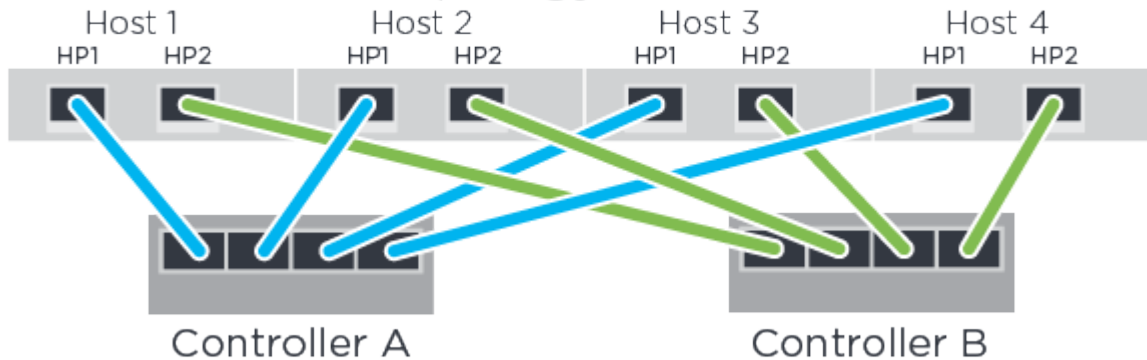
您可以生成并打印此页面的 PDF ，然后使用以下工作表记录基于光纤通道的 NVMe 存储

配置信息。要执行配置任务，您需要此信息。

## 直连拓扑

在直连拓扑中，一个或多个主机直接连接到控制器。

### Direct Connect Topology

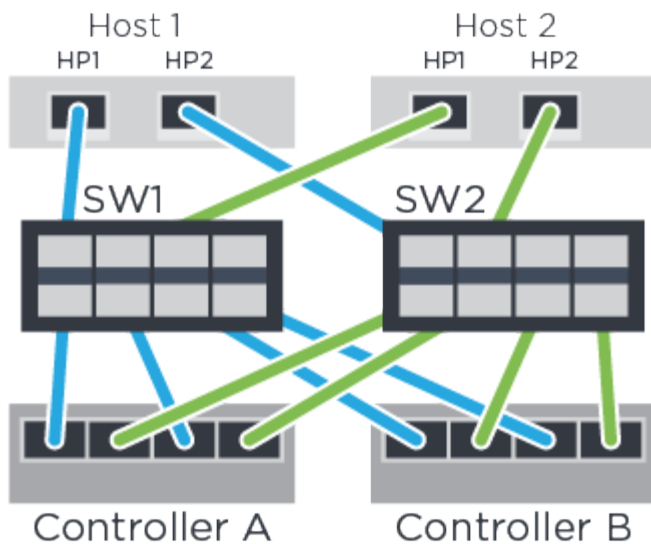


- 主机 1 HBA 端口 1 和控制器 A 主机端口 1
- 主机 1 HBA 端口 2 和控制器 B 主机端口 1
- 主机 2 HBA 端口 1 和控制器 A 主机端口 2
- 主机 2 HBA 端口 2 和控制器 B 主机端口 2
- 主机 3 HBA 端口 1 和控制器 A 主机端口 3
- 主机 3 HBA 端口 2 和控制器 B 主机端口 3
- 主机 4 HBA 端口 1 和控制器 A 主机端口 4
- 主机 4 HBA 端口 2 和控制器 B 主机端口 4

## 交换机连接拓扑

在网络结构拓扑中，使用一个或多个交换机。请参见 ["NetApp 互操作性表工具"](#) 有关支持的交换机的列表。

## Fabric Topology



### 主机标识符

找到并记录每个主机上的启动程序 NQN 。

主机端口连接	主机 NQN
主机（启动程序） 1.	
主机（启动程序） 2.	

### 目标 NQN

记录存储阵列的目标 NQN 。

阵列名称	目标 NQN
阵列控制器（目标）	

### 目标 NQN

记录阵列端口要使用的 NQN 。

阵列控制器（目标）端口连接	NQN
控制器 A ， 端口 1	
控制器 B ， 端口 1	

阵列控制器（目标）端口连接	NQN
控制器 A，端口 2	
控制器 B，端口 2	

正在映射主机名



映射主机名会在工作流期间创建。

正在映射主机名
主机操作系统类型

## 版权信息

版权所有 © 2026 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

## 商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。