落基Linux

ONTAP SAN Host Utilities

NetApp
January 30, 2026

目录

落基Linux . . . 1
- 了解 Rocky Linux 对ONTAP 的支持和功能 . . . 1
  - 下一步 . . . 1
- 配置 Rocky Linux 10.x 以支持 NVMe-oF 和ONTAP存储 . . . 2
  - 第1步：(可选)启用SAN启动 . . . 2
  - 步骤 2：安装 Rocky Linux 和 NVMe 软件并验证您的配置 . . . 2
  - 步骤 3：配置 NVMe/FC 和 NVMe/TCP . . . 4
  - 步骤 4：（可选）为 NVMe/FC 启用 1MB I/O . . . 12
  - 步骤 5：验证 NVMe 启动服务 . . . 13
  - 步骤 6：验证多路径配置 . . . 14
  - 步骤 7：设置安全带内身份验证 . . . 18
  - 第8步：查看已知问题 . . . 25
- 配置 Rocky Linux 9.x 以支持 NVMe-oF 和ONTAP存储 . . . 25
  - 第1步：(可选)启用SAN启动 . . . 25
  - 步骤 2：安装 Rocky Linux 和 NVMe 软件并验证您的配置 . . . 25
  - 步骤 3：配置 NVMe/FC 和 NVMe/TCP . . . 27
  - 步骤 4：（可选）为 NVMe/FC 启用 1MB I/O . . . 39
  - 步骤 5：验证 NVMe 启动服务 . . . 40
  - 步骤 6：验证多路径配置 . . . 41
  - 步骤 7：设置安全带内身份验证 . . . 45
  - 第8步：查看已知问题 . . . 53
- 配置 Rocky Linux 8.x 以支持 NVMe-oF 和ONTAP存储 . . . 53
  - 步骤 1：安装 Rocky Linux 和 NVMe 软件并验证您的配置 . . . 53
  - 步骤 2：配置 NVMe/FC 和 NVMe/TCP . . . 55
  - 步骤 3：可选，启用 NVMe/FC 的 1MB I/O。 . . . 66
  - 步骤 4：验证多路径配置 . . . 67
  - 步骤 5：查看已知问题 . . . 71

落基Linux

了解 Rocky Linux 对ONTAP 的支持和功能

使用 NVMe over Fabrics (NVMe-oF) 进行主机配置所支持的功能因ONTAP和 Rocky Linux 的版本而异。

功能	**Rocky Linux** 主机版本	**ONTAP** 版本
RHEL 主机和ONTAP控制器之间通过 NVMe/TCP 支持安全的带内身份验证。	9.3 或更高版本	9.12.1 或更高版本
NVMe/TCP 使用原生命名空间提供命名空间 `nvme-cli` 包裹	8.2 或更高版本	9.10.1 或更高版本
NVMe/TCP 是一项完全受支持的企业级功能	9.0 或更高版本	9.10.1 或更高版本
同一主机上支持 NVMe 和 SCSI 流量，NVMe-oF 命名空间使用 NVMe 多路径，SCSI LUN 使用 dm 多路径。	8.2 或更高版本	9.4 或更高版本

无论系统设置中运行的ONTAP版本如何， ONTAP支持以下 SAN 主机功能。

功能	**Rocky Linux** 主机版本
默认情况下启用原生 NVMe 多路径。	10.0 或更高版本
SAN 启动已通过 NVMe/FC 协议启用	9.4 或更高版本
这 `nvme-cli`该软件包包含自动连接脚本，无需第三方脚本。	8.2 或更高版本
本机 udev 规则在 `nvme-cli` 包中为 NVMe 多路径提供循环负载平衡	8.2 或更高版本

有关受支持配置的详细信息，请参阅"互操作性表工具"。

下一步

如果您的 **Rocky Linux** 版本是……	了解……
10系列	"为 Rocky Linux 10.x 配置 NVMe"
9系列	"为 Rocky Linux 9.x 配置 NVMe"
8系列	"为 Rocky Linux 8.x 配置 NVMe"

相关信息

"了解如何管理 NVMe 协议"

配置 Rocky Linux 10.x 以支持 NVMe-oF 和ONTAP存储

Rocky Linux 主机支持基于光纤通道的 NVMe (NVMe/FC) 和基于 TCP 的 NVMe (NVMe/TCP) 协议，并支持非对称命名空间访问 (ANA)。ANA 提供与 iSCSI 和 FCP 环境中的非对称逻辑单元访问 (ALUA) 等效的多路径功能。

了解如何为 Rocky Linux 10.x 配置 NVMe over Fabrics (NVMe-oF) 主机。如需更多支持和功能信息，请参阅"Rocky Linux ONTAP支持和功能"。

NVMe-oF 与 Rocky Linux 10.x 存在以下已知限制：

- 这 `nvme disconnect-all`该命令会断开根文件系统和数据文件系统，可能会导致系统不稳定。请勿在通过 NVMe-TCP 或 NVMe-FC 命名空间从 SAN 启动的系统上执行此操作。

第1步：(可选)启用SAN启动

您可以配置主机以使用 SAN 启动来简化部署并提高可扩展性。使用"互操作性表工具"验证您的 Linux 操作系统、主机总线适配器 (HBA)、HBA 固件、HBA 启动 BIOS 和ONTAP版本是否支持 SAN 启动。

步骤

1. "创建 NVMe 命名空间并将其映射到主机" 。
2. 在服务器 BIOS 中为 SAN 启动命名空间映射到的端口启用 SAN 启动。

   有关如何启用 HBA BIOS 的信息，请参见供应商专用文档。

3. 重新启动主机并验证操作系统是否已启动并正在运行。

步骤 2：安装 Rocky Linux 和 NVMe 软件并验证您的配置

要为 NVMe-oF 配置主机，您需要安装主机和 NVMe 软件包，启用多路径，并验证主机 NQN 配置。

步骤

1. 在服务器上安装 Rocky Linux 10.x。安装完成后，请确认您运行的是所需的 Rocky Linux 10.x 内核：

   ```
   uname -r
   ```

   Rocky Linux 内核版本示例：

   ```
   6.12.0-55.9.1.el10_0.x86_64
   ```

2. 安装 `NVMe-CLI` 软件包：

   ```
   rpm -qa|grep nvme-cli
   ```

   下面的例子展示了 `nvme-cli`软件包版本：

```
nvme-cli-2.11-5.el10.x86_64
```

3. 安装 `libnvme` 软件包：

```
rpm -qa|grep libnvme
```

下面的例子展示了 `libnvme`软件包版本：

```
libnvme-1.11.1-1.el10.x86_64
```

4. 在主机上，检查 hostnqn 字符串 `/etc/nvme/hostnqn` ：

```
cat /etc/nvme/hostnqn
```

下面的例子展示了 `hostnqn`价值：

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. 在ONTAP系统中，验证以下信息： `hostnqn`字符串匹配 `hostnqn`ONTAP数组中对应子系统的字符串：

```
::> vserver nvme subsystem host show -vserver vs_nvme_194_rockylinux10
```

显示示例

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
-------------------------------------------------
vs_nvme_194_rockylinux10
        nvme4
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-c7c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-c7c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-c7c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-c7c04f425633
4 entries were displayed.
```

如果 `hostnqn`字符串不匹配，请使用 `vserver modify`命令来更新 `hostnqn`相应ONTAP存储系统子系统上的字符串以匹配 `hostnqn`字符串来自 `/etc/nvme/hostnqn`在主机上。

步骤 3：配置 NVMe/FC 和 NVMe/TCP

使用 Broadcom/Emulex 或 Marvell/QLogic 适配器配置 NVMe/FC，或使用手动发现和连接操作配置 NVMe/TCP。

NVMe/FC - 博通/Emulex

为Broadcom/Emulex适配器配置NVMe/FC。

步骤

1. 验证您使用的适配器型号是否受支持：

   a. 显示模型名称：

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   您应看到以下输出：

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   b. 显示模型描述：

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   您应该会看到类似于以下示例的输出：

   ```
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. 确认您使用的是建议的Broadcom `lpfc` 固件和内置驱动程序：

   a. 显示固件版本：

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   该命令返回固件版本：

   ```
   14.0.539.16, sli-4:6:d
   14.0.539.16, sli-4:6:d
   ```

   b. 显示收件箱驱动程序版本：

   ```
   cat /sys/module/lpfc/version
   ```

以下示例显示了驱动程序版本：

```
0:14.4.0.6
```

有关支持的适配器驱动程序和固件版本的最新列表，请参见"互操作性表工具"。

3. 请验证 `lpfc_enable_fc4_type` 设置为 `3`：

```
cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
```

4. 验证是否可以查看启动程序端口：

```
cat /sys/class/fc_host/host*/port_name
```

此时应显示类似于以下内容的输出：

```
0x2100f4c7aa0cd7c2
0x2100f4c7aa0cd7c3
```

5. 验证启动程序端口是否联机：

```
cat /sys/class/fc_host/host*/port_state
```

您应看到以下输出：

```
Online
Online
```

6. 验证NVMe/FC启动程序端口是否已启用且目标端口是否可见：

```
cat /sys/class/scsi_host/host*/nvme_info
```

显示示例

```
NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x202fd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x202dd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000810 Cmpl 0000000810 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007b098f07 Issue 000000007aee27c4 OutIO
fffffffffe498bd
        abort 000013b4 noxri 00000000 nondlp 00000058 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000013b4 Err 00021443

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2033d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2032d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000840 Cmpl 0000000840 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007afd4434 Issue 000000007ae31b83 OutIO
fffffffffe5d74f
        abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

NVMe/FC - Marvell/QLogic

为Marvell/QLogic适配器配置NVMe/FC。

步骤

1. 验证您使用的适配器驱动程序和固件版本是否受支持：

```
cat /sys/class/fc_host/host*/symbolic_name
```

以下示例显示了驱动程序和固件版本：

```
QLE2872 FW:v9.15.00 DVR:v10.02.09.300-k
QLE2872 FW:v9.15.00 DVR:v10.02.09.300-k
```

2. 请验证 `ql2xnvmeenable` 已设置。这样、Marvell适配器便可用作NVMe/FC启动程序：

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

预期输出为1。

NVMe/TCP

NVMe/TCP 协议不支持自动连接操作。相反，您可以通过执行 NVMe/TCP 来发现 NVMe/TCP 子系统和命名空间 `connect`或者 `connect-all`手动操作。

步骤

1. 检查启动器端口是否可以跨支持的 NVMe/TCP LIF 获取发现日志页面数据：

```
nvme discover -t tcp -w host-traddr -a traddr
```

显示示例

```
nvme discover -t tcp -w 192.168.20.1 -a 192.168.20.20

Discovery Log Number of Records 8, Generation counter 18
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:discovery
traddr:  192.168.21.21
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:discovery
traddr:  192.168.20.21
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  3
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:discovery
traddr:  192.168.21.20
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
```

```
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:discovery
traddr:  192.168.20.20
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:subsystem.rock
ylinux10_tcp_subsystem
traddr:  192.168.21.21
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:subsystem.rock
ylinux10_tcp_subsystem
traddr:  192.168.20.21
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  3
trsvcid: 4420
subnqn:  nqn.1992-
```

```
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:subsystem.rock
ylinux10_tcp_subsystem
traddr:  192.168.21.20
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:subsystem.rock
ylinux10_tcp_subsystem
traddr:  192.168.20.20
eflags:  none
sectype: none
```

2. 验证其他 NVMe/TCP 启动器-目标 LIF 组合是否可以成功检索发现日志页面数据：

```
nvme discover -t tcp -w host-traddr -a traddr
```

显示示例

```
nvme discover -t tcp -w 192.168.20.1 -a 192.168.20.20
nvme discover -t tcp -w 192.168.21.1 -a 192.168.21.20
nvme discover -t tcp -w 192.168.20.1 -a 192.168.20.21
nvme discover -t tcp -w 192.168.21.1 -a 192.168.21.21
```

3. 运行 `nvme connect-all` 在节点中所有受支持的NVMe/TCP启动程序-目标SIP上运行命令：

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

显示示例

```
nvme    connect-all -t  tcp -w  192.168.20.1    -a
192.168.20.20
nvme    connect-all -t  tcp -w  192.168.21.1    -a
192.168.21.20
nvme    connect-all -t  tcp -w  192.168.20.1    -a
192.168.20.21
nvme    connect-all -t  tcp -w  192.168.21.1    -a
192.168.21.21
```

从 Rocky Linux 9.4 开始，NVMe/TCP 的设置 `ctrl_loss_tmo timeout`自动设置为“关闭”。因此：

- 重试次数没有限制（无限重试）。
- 您不需要手动配置特定的 `ctrl_loss_tmo timeout`使用时长 `nvme connect`或者 `nvme connect-all`命令（选项 -l ）。
- 如果发生路径故障，NVMe/TCP 控制器不会超时，并且会无限期地保持连接。

步骤 4：（可选）为 NVMe/FC 启用 1MB I/O

ONTAP在识别控制器数据中报告最大数据传输大小 (MDTS) 为 8。这意味着最大 I/O 请求大小可达 1MB。要向 Broadcom NVMe/FC 主机发出 1MB 大小的 I/O 请求，您应该增加 `lpfc`的价值 `lpfc_sg_seg_cnt`参数从默认值 64 更改为 256。

这些步骤不适用于逻辑NVMe/FC主机。

步骤

1. 将 `lpfc_sg_seg_cnt`参数设置为256：

   ```
   cat /etc/modprobe.d/lpfc.conf
   ```

   您应该会看到类似于以下示例的输出：

   ```
   options lpfc lpfc_sg_seg_cnt=256
   ```

2. 运行 `dracut -f`命令并重新启动主机。
3. 验证的值是否 `lpfc_sg_seg_cnt`为256：

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

步骤 5：验证 NVMe 启动服务

这 `nvmefc-boot-connections.service`和 `nvmf-autoconnect.service`NVMe/FC 中包含的启动服务 `nvme-cli`系统启动时，软件包会自动启用。

启动完成后，验证 `nvmefc-boot-connections.service`和 `nvmf-autoconnect.service`启动服务已启用。

步骤

1. 验证是否 `nvmf-autoconnect.service`已启用：

```
systemctl status nvmf-autoconnect.service
```

显示示例输出

```
nvmf-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
     Loaded: loaded (/usr/lib/systemd/system/nvmf-
autoconnect.service; enabled; preset: disabled)
     Active: inactive (dead)

Jun 10 04:06:26 SR630-13-201.lab.eng.btc.netapp.in systemd[1]:
Starting Connect NVMe-oF subsystems automatically during boot...
Jun 10 04:06:26 SR630-13-201.lab.eng.btc.netapp.in systemd[1]: nvmf-
autoconnect.service: Deactivated successfully.
Jun 10 04:06:26 SR630-13-201.lab.eng.btc.netapp.in systemd[1]:
Finished Connect NVMe-oF subsystems automatically during boot.
```

2. 验证是否 `nvmefc-boot-connections.service`已启用：

```
systemctl status nvmefc-boot-connections.service
```

显示示例输出

```
nvmefc-boot-connections.service - Auto-connect to subsystems on FC-
NVME devices found during boot
     Loaded: loaded (/usr/lib/systemd/system/nvmefc-boot-
connections.service; enabled; preset: enabled)
     Active: inactive (dead) since Tue 2025-06-10 01:08:36 EDT; 2h
59min ago
   Main PID: 7090 (code=exited, status=0/SUCCESS)
        CPU: 30ms

Jun 10 01:08:36 localhost systemd[1]: Starting Auto-connect to
subsystems on FC-NVME devices found during boot...
Jun 10 01:08:36 localhost systemd[1]: nvmefc-boot-
connections.service: Deactivated successfully.
Jun 10 01:08:36 localhost systemd[1]: Finished Auto-connect to
subsystems on FC-NVME devices found during boot.
```

步骤 6：验证多路径配置

验证内核NVMe多路径状态、ANA状态和ONTAP命名空间是否适用于NVMe-oF配置。

步骤

1. 验证是否已启用内核NVMe多路径：

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

   您应看到以下输出：

   ```
   Y
   ```

2. 验证相应ONTAP命名库的适当NVMe-oF设置(例如、型号设置为NetApp ONTAP控制器、负载平衡iopolicy设置为循环)是否正确反映在主机上：

   a. 显示子系统：

      ```
      cat /sys/class/nvme-subsystem/nvme-subsys*/model
      ```

      您应看到以下输出：

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

b. 显示策略：

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

您应看到以下输出：

```
round-robin
round-robin
```

3. 验证是否已在主机上创建并正确发现命名空间：

```
nvme list
```

显示示例

```
Node         SN                   Model
------------------------------------------------------------
/dev/nvme4n1 81Ix2BVuekWcAAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
------------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. 验证每个路径的控制器状态是否为活动状态且是否具有正确的ANA状态：

NVMe/FC

```
nvme list-subsys /dev/nvme5n1
```

显示示例

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.f7565b15a66911ef9668d039ea951c46:subsystem.nvme
1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-
0056-5410-8048-c7c04f425633
\
 +- nvme126 fc traddr=nn-0x2036d039ea951c45:pn-
0x2038d039ea951c45,host_traddr=nn-0x2000f4c7aa0cd7c3:pn-
0x2100f4c7aa0cd7c3 live optimized
 +- nvme176 fc traddr=nn-0x2036d039ea951c45:pn-
0x2037d039ea951c45,host_traddr=nn-0x2000f4c7aa0cd7c2:pn-
0x2100f4c7aa0cd7c2 live optimized
 +- nvme5 fc traddr=nn-0x2036d039ea951c45:pn-
0x2039d039ea951c45,host_traddr=nn-0x2000f4c7aa0cd7c2:pn-
0x2100f4c7aa0cd7c2 live non-optimized
 +- nvme71 fc traddr=nn-0x2036d039ea951c45:pn-
0x203ad039ea951c45,host_traddr=nn-0x2000f4c7aa0cd7c3:pn-
0x2100f4c7aa0cd7c3 live non-optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme4n2
```

显示示例

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:subsystem.nvme
4
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-
0035-5910-804b-c2c04f444d33
\
+- nvme102 tcp
traddr=192.168.21.20,trsvcid=4420,host_traddr=192.168.21.1,src_a
ddr=192.168.21.1 live non-optimized
+- nvme151 tcp
traddr=192.168.21.21,trsvcid=4420,host_traddr=192.168.21.1,src_a
ddr=192.168.21.1 live optimized
+- nvme4 tcp
traddr=192.168.20.20,trsvcid=4420,host_traddr=192.168.20.1,src_a
ddr=192.168.20.1 live non-optimized
+- nvme53 tcp
traddr=192.168.20.21,trsvcid=4420,host_traddr=192.168.20.1,src_a
ddr=192.168.20.1 live optimized
```

5. 验证NetApp插件是否为每个ONTAP 命名空间设备显示正确的值：

列

```
nvme netapp ontapdevices -o column
```

显示示例

```
Device        Vserver   Namespace Path
---------------------- ------------------------------------
/dev/nvme10n1     vs_tcp_rockylinux10       /vol/vol10/ns10

NSID       UUID                                   Size
---------------------- ------------------------------------
1    bbf51146-fc64-4197-b8cf-8a24f6f359b3   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

显示示例

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme10n1",
      "Vserver":"vs_tcp_rockylinux10",
      "Namespace_Path":"/vol/vol10/ns10",
      "NSID":1,
      "UUID":"bbf51146-fc64-4197-b8cf-8a24f6f359b3",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
}
]
    }
```

步骤 7：设置安全带内身份验证

Rocky Linux 10.x 主机和ONTAP控制器之间通过 NVMe/TCP 支持安全的带内身份验证。

每个主机或控制器必须与一个 `DH-HMAC-CHAP`密钥来设置安全身份验证。 `DH-HMAC-CHAP`密钥是 NVMe

主机或控制器的 NQN 与管理员配置的身份验证密钥的组合。要对其对等方进行身份验证、NVMe主机或控制器必须识别与对等方关联的密钥。

步骤

使用 CLI 或配置 JSON 文件设置安全带内身份验证。如果需要为不同的子系统指定不同的dhchap密钥、则必须使用config JSON文件。

命令行界面

使用命令行界面设置安全带内身份验证。

1. 获取主机NQN：

```
cat /etc/nvme/hostnqn
```

2. 为 Rocky Linux 10.x 主机生成 dhchap 密钥。

   以下输出描述了 `gen-dhchap-key`命令参数：

```
nvme gen-dhchap-key -s optional_secret -l key_length {32|48|64} -m
HMAC_function {0|1|2|3} -n host_nqn
•   -s secret key in hexadecimal characters to be used to initialize
the host key
•   -l length of the resulting key in bytes
•   -m HMAC function to use for key transformation
0 = none, 1- SHA-256, 2 = SHA-384, 3=SHA-512
•   -n host NQN to use for key transformation
```

   在以下示例中、将生成一个随机dhchap密钥、其中HMAC设置为3 (SHA-512)。

```
nvme gen-dhchap-key -m 3 -n nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-c2c04f444d33
DHHC-
1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia1aKDKJQ2o53pX3wYM9
xdv5DtKNNhJInZ7X8wU2RQpQIngc=:
```

3. 在ONTAP控制器上、添加主机并指定两个dhchap密钥：

```
vserver nvme subsystem host add -vserver <svm_name> -subsystem
<subsystem> -host-nqn <host_nqn> -dhchap-host-secret
<authentication_host_secret> -dhchap-controller-secret
<authentication_controller_secret> -dhchap-hash-function {sha-
256|sha-512} -dhchap-group {none|2048-bit|3072-bit|4096-bit|6144-
bit|8192-bit}
```

4. 主机支持两种类型的身份验证方法：单向和双向。在主机上、连接到ONTAP控制器并根据所选身份验证方法指定dhchap密钥：

```
nvme connect -t tcp -w <host-traddr> -a <tr-addr> -n <host_nqn> -S
<authentication_host_secret> -C <authentication_controller_secret>
```

5. 验证 `nvme connect authentication` 命令、验证主机和控制器dhchap密钥：

   a. 验证主机dhchap密钥：

   ```
   cat /sys/class/nvme-subsystem/<nvme-subsysX>/nvme*/dhchap_secret
   ```

   显示单向配置的示例输出

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys1/nvme*/dhchap_secret
   DHHC-
   1:03:fMCrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jTmzEKUbcWu26I33b93b
   il2WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:
   DHHC-
   1:03:fMCrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jTmzEKUbcWu26I33b93b
   il2WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:
   DHHC-
   1:03:fMCrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jTmzEKUbcWu26I33b93b
   il2WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:
   DHHC-
   1:03:fMCrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jTmzEKUbcWu26I33b93b
   il2WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:
   ```

   b. 验证控制器dhchap密钥：

   ```
   cat /sys/class/nvme-subsystem/<nvme-
   subsysX>/nvme*/dhchap_ctrl_secret
   ```

显示双向配置的示例输出

```
cat /sys/class/nvme-subsystem/nvme-
subsys6/nvme*/dhchap_ctrl_secret
DHHC- 1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia
1aKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:

DHHC- 1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia
1aKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:

DHHC- 1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia
1aKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:

DHHC- 1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia
1aKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:
```

JSON

当ONTAP控制器上有多个 NVMe 子系统可用时，您可以使用 `/etc/nvme/config.json`文件与 `nvme connect-all`命令。

使用 `-o`选项来生成 JSON 文件。有关更多语法选项，请参阅 NVMe connect-all 手册页。

1. 配置 JSON 文件。

> 在以下示例中，`dhchap_key`对应于 `dhchap_secret`和 `dhchap_ctrl_key` 对应于 `dhchap_ctrl_secret`。

显示示例

```
cat /etc/nvme/config.json
[
  {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-
5910-804b-c2c04f444d33",
    "hostid":"4c4c4544-0035-5910-804b-c2c04f444d33",
    "dhchap_key":"DHHC-
1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia1aKDKJQ2o53pX3
wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:",
    "subsystems":[
      {
        "nqn":"nqn.1992-
08.com.netapp:sn.127ade26168811f0a50ed039eab69ad3:subsystem.inba
nd_unidirectional",
        "ports":[
          {
            "transport":"tcp",
            "traddr":"192.168.20.17",
            "host_traddr":"192.168.20.1",
            "trsvcid":"4420"
          },
          {
            "transport":"tcp",
            "traddr":"192.168.20.18",
            "host_traddr":"192.168.20.1",
            "trsvcid":"4420"
          },
          {
            "transport":"tcp",
            "traddr":"192.168.21.18",
            "host_traddr":"192.168.21.1",
            "trsvcid":"4420"
          },
          {
            "transport":"tcp",
            "traddr":"192.168.21.17",
            "host_traddr":"192.168.21.1",
            "trsvcid":"4420"
}
]
```

2. 使用config JSON文件连接到ONTAP控制器：

```
nvme connect-all -J /etc/nvme/config.json
```

显示示例

```
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
```

3. 验证每个子系统的相应控制器是否已启用 dhchap 机密。

   a. 验证主机dhchap密钥：

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys0/nvme0/dhchap_secret
   ```

   以下示例显示了 dhchap 密钥：

   ```
   DHHC-1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia1
   aKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:
   ```

   b. 验证控制器dhchap密钥：

   ```
   cat /sys/class/nvme-subsystem/nvme-
   subsys0/nvme0/dhchap_ctrl_secret
   ```

   您应该会看到类似于以下示例的输出：

```
DHHC-1:03:fMCrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jT
mzEKUbcWu26I33b93bil2WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:
```

第8步：查看已知问题

没有已知问题。

配置 Rocky Linux 9.x 以支持 NVMe-oF 和ONTAP存储

Rocky Linux 主机支持基于光纤通道的 NVMe (NVMe/FC) 和基于 TCP 的 NVMe (NVMe/TCP) 协议，并支持非对称命名空间访问 (ANA)。ANA 提供与 iSCSI 和 FCP 环境中的非对称逻辑单元访问 (ALUA) 等效的多路径功能。

了解如何为 Rocky Linux 9.x 配置 NVMe over Fabrics (NVMe-oF) 主机。如需更多支持和功能信息，请参阅 "Rocky Linux ONTAP支持和功能"。

NVMe-oF 与 Rocky Linux 9.x 存在以下已知限制：

- 这 `nvme disconnect-all`该命令会断开根文件系统和数据文件系统，可能会导致系统不稳定。请勿在通过 NVMe-TCP 或 NVMe-FC 命名空间从 SAN 启动的系统上执行此操作。

第1步：(可选)启用SAN启动

您可以配置主机以使用 SAN 启动来简化部署并提高可扩展性。使用"互操作性表工具"验证您的 Linux 操作系统、主机总线适配器 (HBA)、HBA 固件、HBA 启动 BIOS 和ONTAP版本是否支持 SAN 启动。

步骤

1. "创建 NVMe 命名空间并将其映射到主机" 。
2. 在服务器 BIOS 中为 SAN 启动命名空间映射到的端口启用 SAN 启动。

   有关如何启用 HBA BIOS 的信息，请参见供应商专用文档。

3. 重新启动主机并验证操作系统是否已启动并正在运行。

步骤 2：安装 Rocky Linux 和 NVMe 软件并验证您的配置

要为 NVMe-oF 配置主机，您需要安装主机和 NVMe 软件包，启用多路径，并验证主机 NQN 配置。

步骤

1. 在服务器上安装 Rocky Linux 9.x。安装完成后，请确认您运行的是所需的 Rocky Linux 9.x 内核：

   ```
   uname -r
   ```

   Rocky Linux 内核版本示例：

```
5.14.0-570.12.1.el9_6.x86_64
```

2. 安装 NVMe-CLI 软件包：

```
rpm -qa|grep nvme-cli
```

以下示例显示了 nvme-cli 软件包版本：

```
nvme-cli-2.11-5.el9.x86_64
```

3. 安装 libnvme 软件包：

```
rpm -qa|grep libnvme
```

下面的例子展示了 `libnvme`软件包版本：

```
libnvme-1.11.1-1.el9.x86_64
```

4. 在 Rocky Linux 主机上，检查 hostnqn 字符串 /etc/nvme/hostnqn ：

```
cat /etc/nvme/hostnqn
```

下面的例子展示了 `hostnqn`版本：

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. 在ONTAP系统中，验证以下信息： `hostnqn`字符串匹配 `hostnqn`ONTAP数组中对应子系统的字符串：

```
::> vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

显示示例

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
-------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
4 entries were displayed.
```

如果 `hostnqn` 字符串不匹配、请使用 `vserver modify` 用于更新的命令 `hostnqn` 要匹配的相应ONTAP 阵列子系统上的字符串 `hostnqn` 字符串自 `/etc/nvme/hostnqn` 在主机上。

步骤 3：配置 NVMe/FC 和 NVMe/TCP

使用 Broadcom/Emulex 或 Marvell/QLogic 适配器配置 NVMe/FC，或使用手动发现和连接操作配置 NVMe/TCP。

NVMe/FC - 博通/Emulex

为Broadcom/Emulex适配器配置NVMe/FC。

步骤

1. 验证您使用的适配器型号是否受支持：

   a. 显示模型名称：

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   您应看到以下输出：

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   b. 显示模型描述：

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   您应该会看到类似于以下示例的输出：

   ```
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. 确认您使用的是建议的Broadcom `lpfc` 固件和内置驱动程序：

   a. 显示固件版本：

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   该命令返回固件版本：

   ```
   14.0.539.16, sli-4:6:d
   14.0.539.16, sli-4:6:d
   ```

   b. 显示收件箱驱动程序版本：

   ```
   cat /sys/module/lpfc/version
   ```

以下示例显示了驱动程序版本：

```
0:14.4.0.6
```

有关支持的适配器驱动程序和固件版本的最新列表，请参见"互操作性表工具"。

3. 请验证 `lpfc_enable_fc4_type` 设置为 `3`：

```
cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
```

4. 验证是否可以查看启动程序端口：

```
cat /sys/class/fc_host/host*/port_name
```

以下示例显示端口标识：

```
0x2100f4c7aa0cd7c2
0x2100f4c7aa0cd7c3
```

5. 验证启动程序端口是否联机：

```
cat /sys/class/fc_host/host*/port_state
```

您应看到以下输出：

```
Online
Online
```

6. 验证NVMe/FC启动程序端口是否已启用且目标端口是否可见：

```
cat /sys/class/scsi_host/host*/nvme_info
```

显示示例

```
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b954518 WWNN x200000109b954518
DID x000000 ONLINE

NVME Statistics
LS: Xmt 0000000000 Cmpl 0000000000 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000000000000 Issue 0000000000000000 OutIO
0000000000000000
          abort 00000000 noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000000 Err 00000000

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b954519 WWNN x200000109b954519
DID x020500 ONLINE

NVME Statistics
LS: Xmt 0000000000 Cmpl 0000000000 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000000000000 Issue 0000000000000000 OutIO
0000000000000000
         abort 00000000 noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000000 Err 00000000

NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x200bd039eaa7dfc8 WWNN x2008d039eaa7dfc8
DID x021319 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2155d039eaa7dfc8 WWNN x2154d039eaa7dfc8
DID x02130f TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2001d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x200dd039eaa7dfc8 WWNN x2008d039eaa7dfc8
DID x020b15 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2156d039eaa7dfc8 WWNN x2154d039eaa7dfc8
DID x020b0d TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2003d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE
```

```
NVME Statistics
LS: Xmt 0000003049 Cmpl 0000003049 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000018f9450b Issue 0000000018f5de57 OutIO
fffffffffffc994c
          abort 000036d3 noxri 00000313 nondlp 00000c8d qdepth
00000000 wqerr 00000064 err 00000000
FCP CMPL: xb 000036d1 Err 000fef0f

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2062d039eaa7dfc8 WWNN x2008d039eaa7dfc8
DID x022915 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2157d039eaa7dfc8 WWNN x2154d039eaa7dfc8
DID x02290f TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2002d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2065d039eaa7dfc8 WWNN x2008d039eaa7dfc8
DID x020119 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2158d039eaa7dfc8 WWNN x2154d039eaa7dfc8
DID x02010d TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2004d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000002f2c Cmpl 0000002f2c Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000001aaf3eb5 Issue 000000001aab4373 OutIO
fffffffffffc04be
          abort 000035cc noxri 0000038c nondlp 000009e3 qdepth
00000000 wqerr 00000082 err 00000000
FCP CMPL: xb 000035cc Err 000fcfc0
```

NVMe/FC - Marvell/QLogic

为Marvell/QLogic适配器配置NVMe/FC。

步骤

1. 验证您使用的适配器驱动程序和固件版本是否受支持：

```
cat /sys/class/fc_host/host*/symbolic_name
```

以下示例显示了驱动程序和固件版本：

```
QLE2872 FW:v9.15.00 DVR:v10.02.09.300-k
QLE2872 FW:v9.15.00 DVR:v10.02.09.300-k
```

2. 请验证 `ql2xnvmeenable` 已设置。这样、Marvell适配器便可用作NVMe/FC启动程序：

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

预期输出为1。

NVMe/TCP

NVMe/TCP协议不支持自动连接操作。您必须手动执行 NVMe/TCP 连接或连接所有操作才能发现 NVMe/TCP 子系统和命名空间。

步骤

1. 检查启动器端口是否可以跨支持的 NVMe/TCP LIF 获取发现日志页面数据：

```
nvme discover -t tcp -w host-traddr -a traddr
```

显示示例

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24

Discovery Log Number of Records 20, Generation counter 25
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
```

```
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Unidirectional_DHCP_NONE_1_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Unidirectional_DHCP_NONE_1_4
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
```

```
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Unidirectional_DHCP_NONE_1_5
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Unidirectional_DHCP_2_2
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Unidirectional_DHCP_2_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Unidirectional_DHCP_2_5
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
```

```
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_2_2
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_2_3
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_2_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
```

```
Bidirectional_DHCP_NONE_2_4
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 14======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_NONE_2_5
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_NONE_2_6
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 16======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_NONE_2_7
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 17======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_NONE_2_8
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 18======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_2
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 19======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_NONE_2_9
traddr:  192.168.1.24
eflags:  none
sectype: none
```

2. 验证其他 NVMe/TCP 启动器-目标 LIF 组合是否可以成功获取发现日志页面数据：

```
nvme discover -t tcp -w host-traddr -a traddr
```

显示示例

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.24
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.25
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.25
```

3. 运行 `nvme connect-all` 在节点中所有受支持的NVMe/TCP启动程序-目标SIP上运行命令：

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

显示示例

```
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.24
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.24
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.25
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.25
```

从 Rocky Linux 9.4 开始，NVMe/TCP 的设置 `ctrl_loss_tmo timeout`自动设置为“关闭”。因此：

- 重试次数没有限制（无限重试）。
- 您不需要手动配置特定的 `ctrl_loss_tmo timeout`使用时长 `nvme connect`或者 `nvme connect-all`命令（选项 -l ）。
- 如果发生路径故障，NVMe/TCP 控制器不会超时，并且会无限期地保持连接。

步骤 4：（可选）为 NVMe/FC 启用 1MB I/O

ONTAP在识别控制器数据中报告最大数据传输大小 (MDTS) 为 8。这意味着最大 I/O 请求大小可达 1MB。要向 Broadcom NVMe/FC 主机发出 1MB 大小的 I/O 请求，您应该增加 `lpfc`的价值 `lpfc_sg_seg_cnt`参数从默认值 64 更改为 256。

这些步骤不适用于逻辑NVMe/FC主机。

步骤

1. 将 `lpfc_sg_seg_cnt`参数设置为256：

```
cat /etc/modprobe.d/lpfc.conf
```

您应该会看到类似于以下示例的输出：

```
options lpfc lpfc_sg_seg_cnt=256
```

2. 运行 `dracut -f`命令并重新启动主机。
3. 验证的值是否 `lpfc_sg_seg_cnt`为256：

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

步骤 5：验证 NVMe 启动服务

这 `nvmefc-boot-connections.service`和 `nvmf-autoconnect.service`NVMe/FC 中包含的启动服务 `nvme-cli`系统启动时，软件包会自动启用。

启动完成后，验证 `nvmefc-boot-connections.service`和 `nvmf-autoconnect.service`启动服务已启用。

步骤

1. 验证是否 `nvmf-autoconnect.service`已启用：

```
systemctl status nvmf-autoconnect.service
```

显示示例输出

```
nvmf-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
     Loaded: loaded (/usr/lib/systemd/system/nvmf-
autoconnect.service; enabled; preset: disabled)
     Active: inactive (dead)

Jun 10 04:06:26 SR630-13-201.lab.eng.btc.netapp.in systemd[1]:
Starting Connect NVMe-oF subsystems automatically during boot...
Jun 10 04:06:26 SR630-13-201.lab.eng.btc.netapp.in systemd[1]: nvmf-
autoconnect.service: Deactivated successfully.
Jun 10 04:06:26 SR630-13-201.lab.eng.btc.netapp.in systemd[1]:
Finished Connect NVMe-oF subsystems automatically during boot.
```

2. 验证是否 `nvmefc-boot-connections.service`已启用：

```
systemctl status nvmefc-boot-connections.service
```

显示示例输出

```
nvmefc-boot-connections.service - Auto-connect to subsystems on FC-
NVME devices found during boot
     Loaded: loaded (/usr/lib/systemd/system/nvmefc-boot-
connections.service; enabled; preset: enabled)
     Active: inactive (dead) since Tue 2025-06-10 01:08:36 EDT; 2h
59min ago
   Main PID: 7090 (code=exited, status=0/SUCCESS)
        CPU: 30ms

Jun 10 01:08:36 localhost systemd[1]: Starting Auto-connect to
subsystems on FC-NVME devices found during boot...
Jun 10 01:08:36 localhost systemd[1]: nvmefc-boot-
connections.service: Deactivated successfully.
Jun 10 01:08:36 localhost systemd[1]: Finished Auto-connect to
subsystems on FC-NVME devices found during boot.
```

步骤 6：验证多路径配置

验证内核NVMe多路径状态、ANA状态和ONTAP命名空间是否适用于NVMe-oF配置。

步骤

1. 验证是否已启用内核NVMe多路径：

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

   您应看到以下输出：

   ```
   Y
   ```

2. 验证相应ONTAP命名库的适当NVMe-oF设置(例如、型号设置为NetApp ONTAP控制器、负载平衡iopolicy设置为循环)是否正确反映在主机上：

   a. 显示子系统：

      ```
      cat /sys/class/nvme-subsystem/nvme-subsys*/model
      ```

      您应看到以下输出：

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

b. 显示策略：

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

您应看到以下输出：

```
round-robin
round-robin
```

3. 验证是否已在主机上创建并正确发现命名空间：

```
nvme list
```

显示示例

```
Node         SN                   Model
------------------------------------------------------------
/dev/nvme4n1 81Ix2BVuekWcAAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
------------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. 验证每个路径的控制器状态是否为活动状态且是否具有正确的ANA状态：

NVMe/FC

```
nvme list-subsys /dev/nvme4n5
```

显示示例

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.3a5d31f5502c11ef9f50d039eab6cb6d:subsystem.nvme
_1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:e6dade64-
216d-
11ec-b7bb-7ed30a5482c3
iopolicy=round-robin\
+- nvme1 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2088d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live optimized
+- nvme12 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x208ad039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live non-optimized
+- nvme10 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2087d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live non-optimized
+- nvme3 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2083d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n1
```

显示示例

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
iopolicy=round-robin
\
+- nvme13 tcp
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live optimized
+- nvme14 tcp
traddr=192.168.2.24,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live non-optimized
+- nvme5 tcp
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live optimized
+- nvme6 tcp
traddr=192.168.1.24,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live non-optimized
```

5. 验证NetApp插件是否为每个ONTAP 命名空间设备显示正确的值：

列

```
nvme netapp ontapdevices -o column
```

显示示例

```
Device        Vserver   Namespace Path
---------------------- ------------------------------
/dev/nvme1n1     linux_tcnvme_iscsi
/vol/tcpnvme_1_0_0/tcpnvme_ns

NSID       UUID                                   Size
--------------------------------------------------------------
1    5f7f630d-8ea5-407f-a490-484b95b15dd6   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

显示示例

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme1n1",
      "Vserver":"linux_tcnvme_iscsi",
      "Namespace_Path":"/vol/tcpnvme_1_0_0/tcpnvme_ns",
      "NSID":1,
      "UUID":"5f7f630d-8ea5-407f-a490-484b95b15dd6",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
]
}
```

步骤 7：设置安全带内身份验证

Rocky Linux 9x 主机和ONTAP控制器之间通过 NVMe/TCP 支持安全的带内身份验证。

每个主机或控制器必须与一个 `DH-HMAC-CHAP`密钥来设置安全身份验证。 `DH-HMAC-CHAP`密钥是 NVMe 主机或控制器的 NQN 与管理员配置的身份验证密钥的组合。要对其对等方进行身份验证、NVMe主机或控制器必须识别与对等方关联的密钥。

步骤

使用 CLI 或配置 JSON 文件设置安全带内身份验证。如果需要为不同的子系统指定不同的dhchap密钥、则必须使用config JSON文件。

命令行界面

使用命令行界面设置安全带内身份验证。

1. 获取主机NQN：

```
cat /etc/nvme/hostnqn
```

2. 为 Rocky Linux 9.x 主机生成 dhchap 密钥。

   以下输出描述了 `gen-dhchap-key`命令参数：

```
nvme gen-dhchap-key -s optional_secret -l key_length {32|48|64} -m
HMAC_function {0|1|2|3} -n host_nqn
•   -s secret key in hexadecimal characters to be used to initialize
the host key
•   -l length of the resulting key in bytes
•   -m HMAC function to use for key transformation
0 = none, 1- SHA-256, 2 = SHA-384, 3=SHA-512
•   -n host NQN to use for key transformation
```

   在以下示例中、将生成一个随机dhchap密钥、其中HMAC设置为3 (SHA-512)。

```
nvme gen-dhchap-key -m 3 -n nqn.2014-
08.org.nvmexpress:uuid:e6dade64-216d-11ec-b7bb-7ed30a5482c3
DHHC-
1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjD
WOo0PJJM6yZsTeEpGkDHMHQ255+g=:
```

3. 在ONTAP控制器上、添加主机并指定两个dhchap密钥：

```
vserver nvme subsystem host add -vserver <svm_name> -subsystem
<subsystem> -host-nqn <host_nqn> -dhchap-host-secret
<authentication_host_secret> -dhchap-controller-secret
<authentication_controller_secret> -dhchap-hash-function {sha-
256|sha-512} -dhchap-group {none|2048-bit|3072-bit|4096-bit|6144-
bit|8192-bit}
```

4. 主机支持两种类型的身份验证方法：单向和双向。在主机上、连接到ONTAP控制器并根据所选身份验证方法指定dhchap密钥：

```
nvme connect -t tcp -w <host-traddr> -a <tr-addr> -n <host_nqn> -S
<authentication_host_secret> -C <authentication_controller_secret>
```

5. 验证 `nvme connect authentication` 命令、验证主机和控制器dhchap密钥：

   a. 验证主机dhchap密钥：

   ```
   cat /sys/class/nvme-subsystem/<nvme-subsysX>/nvme*/dhchap_secret
   ```

   显示单向配置的示例输出

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys1/nvme*/dhchap_secret
   DHHC-1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAi:
   DHHC-1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAi:
   DHHC-1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAi:
   DHHC-1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAi:
   ```

   b. 验证控制器dhchap密钥：

   ```
   cat /sys/class/nvme-subsystem/<nvme-
   subsysX>/nvme*/dhchap_ctrl_secret
   ```

   显示双向配置的示例输出

   ```
   cat /sys/class/nvme-subsystem/nvme-
   subsys6/nvme*/dhchap_ctrl_secret
   DHHC-
   1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMp
   zhmyjDWOo0PJJM6yZsTeEpGkDHMHQ255+g=:
   DHHC-
   1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMp
   zhmyjDWOo0PJJM6yZsTeEpGkDHMHQ255+g=:
   DHHC-
   1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMp
   zhmyjDWOo0PJJM6yZsTeEpGkDHMHQ255+g=:
   DHHC-
   1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMp
   zhmyjDWOo0PJJM6yZsTeEpGkDHMHQ255+g=:
   ```

JSON

当ONTAP控制器上有多个 NVMe 子系统可用时，您可以使用 `/etc/nvme/config.json`文件与 `nvme connect-all`命令。

使用 `-o`选项来生成 JSON 文件。有关更多语法选项，请参阅 NVMe connect-all 手册页。

1. 配置 JSON 文件。

> 在以下示例中， `dhchap_key`对应于 `dhchap_secret`和 `dhchap_ctrl_key`对应于 `dhchap_ctrl_secret`。

显示示例

```
cat /etc/nvme/config.json
[
{
  "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:9796c1ec-0d34-11eb-
  b6b2-3a68dd3bab57",
  "hostid":"b033cd4fd6db4724adb48655bfb55448",
  "dhchap_key":" DHHC-
1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAi:"
},
{
  "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-
  804b-b5c04f444d33",
  "subsystems":[
        {
          "nqn":"nqn.1992-

08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.bidi
          r_DHCP",
          "ports":[
              {
                  "transport":"tcp",
                    "traddr":" 192.168.1.24 ",
                  "host_traddr":" 192.168.1.31 ",
                  "trsvcid":"4420",
                  "dhchap_ctrl_key":"DHHC-
                  1:03:
wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjDW
Oo0PJJM6yZsTeEpGkDHMHQ255+g=:"
              },
              {
                  "transport":"tcp",
                  "traddr":" 192.168.1.25 ",
                  "host_traddr":" 192.168.1.31",
                  "trsvcid":"4420",
                  "dhchap_ctrl_key":"DHHC-
                  1:03:
wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjDW
Oo0PJJM6yZsTeEpGkDHMHQ255+g=:"
              },
              {
                  "transport":"tcp",
                 "traddr":" 192.168.2.24 ",
                  "host_traddr":" 192.168.2.31",
```

```
                "trsvcid":"4420",
                "dhchap_ctrl_key":"DHHC-
                1:03:
wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjDW
Oo0PJJM6yZsTeEpGkDHMHQ255+g=:"
            },
            {
                "transport":"tcp",
                "traddr":" 192.168.2.25 ",
                  "host_traddr":" 192.168.2.31",
                "trsvcid":"4420",
                "dhchap_ctrl_key":"DHHC-
                1:03:
wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjDW
Oo0PJJM6yZsTeEpGkDHMHQ255+g=:"
            }
        ]
    }
  ]
}
]
```

2. 使用config JSON文件连接到ONTAP控制器：

```
nvme connect-all -J /etc/nvme/config.json
```

显示示例

```
already connected to hostnqn=nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33,nqn=nqn.1992-
08.com.netapp:sn.8dde3be2cc7c11efb777d039eab6cb6d:subsystem.
bidi
r_DHCP,transport=tcp,traddr=192.168.1.25,trsvcid=4420
already connected to hostnqn=nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33,nqn=nqn.1992-
08.com.netapp:sn.8dde3be2cc7c11efb777d039eab6cb6d:subsystem.
bidi
r_DHCP,transport=tcp,traddr=192.168.2.25,trsvcid=4420
already connected to hostnqn=nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33,nqn=nqn.1992-
08.com.netapp:sn.8dde3be2cc7c11efb777d039eab6cb6d:subsystem.
bidi
r_DHCP,transport=tcp,traddr=192.168.1.24,trsvcid=4420
already connected to hostnqn=nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33,nqn=nqn.1992-
08.com.netapp:sn.8dde3be2cc7c11efb777d039eab6cb6d:subsystem.
bidi
r_DHCP,transport=tcp,traddr=192.168.2.24,trsvcid=4420
```

3. 验证是否已为每个子系统的相应控制器启用dhchap密码：

   a. 验证主机dhchap密钥：

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys0/nvme0/dhchap_secret
   ```

   以下示例显示了 dhchap 密钥：

   ```
   DHHC-1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAi:
   ```

   b. 验证控制器dhchap密钥：

   ```
   cat /sys/class/nvme-subsystem/nvme-
   subsys0/nvme0/dhchap_ctrl_secret
   ```

您应该会看到类似于以下示例的输出：

```
DHHC-
1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjD
WOo0PJJM6yZsTeEpGkDHMHQ255+g=:
```

第8步：查看已知问题

没有已知问题。

配置 Rocky Linux 8.x 以支持 NVMe-oF 和ONTAP存储

Rocky Linux 主机支持基于光纤通道的 NVMe (NVMe/FC) 和基于 TCP 的 NVMe (NVMe/TCP) 协议，并支持非对称命名空间访问 (ANA)。ANA 提供与 iSCSI 和 FCP 环境中的非对称逻辑单元访问 (ALUA) 等效的多路径功能。

了解如何为 Rocky Linux 8.x 配置 NVMe over Fabrics (NVMe-oF) 主机。如需更多支持和功能信息，请参阅 "Rocky Linux ONTAP支持和功能"。

NVMe-oF 与 Rocky Linux 8.x 存在以下已知限制：

- 目前不支持使用 NVMe-oF 协议的 SAN 启动。
- 在 Rocky Linux 8.x 中，NVMe-oF 主机上的内核内 NVMe 多路径默认处于禁用状态；您必须手动启用它。
- 由于已知问题，NVMe/TCP 可作为技术预览使用。

步骤 1：安装 Rocky Linux 和 NVMe 软件并验证您的配置

要为 NVMe-oF 配置主机，您需要安装主机和 NVMe 软件包，启用多路径，并验证主机 NQN 配置。

步骤

1. 在服务器上安装 Rocky Linux 8.x。安装完成后，请确认您运行的是所需的 Rocky Linux 8.x 内核：

   ```
   uname -r
   ```

   Rocky Linux 内核版本示例：

   ```
   5.14.0-570.12.1.el9_6.x86_64
   ```

2. 安装 `NVMe-CLI` 软件包：

   ```
   rpm -qa|grep nvme-cli
   ```

以下示例显示了 nvme-cli 软件包版本：

```
nvme-cli-2.11-5.el9.x86_64
```

3. 安装 `libnvme` 软件包：

```
rpm -qa|grep libnvme
```

下面的例子展示了 `libnvme`软件包版本：

```
libnvme-1.11.1-1.el9.x86_64
```

4. 在 Rocky Linux 主机上，检查 hostnqn 字符串 `/etc/nvme/hostnqn` ：

```
cat /etc/nvme/hostnqn
```

下面的例子展示了 `hostnqn`版本：

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. 在ONTAP系统中，验证以下信息： `hostnqn`字符串匹配 `hostnqn`ONTAP数组中对应子系统的字符串：

```
::> vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

显示示例

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
-------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
4 entries were displayed.
```

如果 `hostnqn` 字符串不匹配、请使用 `vserver modify` 用于更新的命令 `hostnqn` 要匹配的相应ONTAP 阵列子系统上的字符串 `hostnqn` 字符串自 `/etc/nvme/hostnqn` 在主机上。

步骤 2：配置 NVMe/FC 和 NVMe/TCP

使用 Broadcom/Emulex 或 Marvell/QLogic 适配器配置 NVMe/FC，或使用手动发现和连接操作配置 NVMe/TCP。

NVMe/FC - 博通/Emulex

为Broadcom/Emulex适配器配置NVMe/FC。

步骤

1. 验证您使用的适配器型号是否受支持：

   a. 显示模型名称：

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   您应看到以下输出：

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   b. 显示模型描述：

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   您应该会看到类似于以下示例的输出：

   ```
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. 确认您使用的是建议的Broadcom `lpfc` 固件和内置驱动程序：

   a. 显示固件版本：

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   该命令返回固件版本：

   ```
   14.4.317.10, sli-4:6:d
   14.4.317.10, sli-4:6:d
   ```

   b. 显示收件箱驱动程序版本：

   ```
   cat /sys/module/lpfc/version`
   ```

以下示例显示了驱动程序版本：

```
0:14.4.0.2
```

有关支持的适配器驱动程序和固件版本的最新列表，请参见"互操作性表工具"。

3. 验证的预期输出是否 `lpfc_enable_fc4_type`设置为 `3`：

```
cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
```

4. 验证是否可以查看启动程序端口：

```
cat /sys/class/fc_host/host*/port_name
```

以下示例显示端口标识：

```
0x100000109bf044b1
0x100000109bf044b2
```

5. 验证启动程序端口是否联机：

```
cat /sys/class/fc_host/host*/port_state
```

您应看到以下输出：

```
Online
Online
```

6. 验证NVMe/FC启动程序端口是否已启用且目标端口是否可见：

```
cat /sys/class/scsi_host/host*/nvme_info
```

显示示例

```
NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x202fd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x202dd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000810 Cmpl 0000000810 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007b098f07 Issue 000000007aee27c4 OutIO
fffffffffe498bd
        abort 000013b4 noxri 00000000 nondlp 00000058 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000013b4 Err 00021443

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2033d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2032d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000840 Cmpl 0000000840 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007afd4434 Issue 000000007ae31b83 OutIO
fffffffffe5d74f
        abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

NVMe/FC - Marvell/QLogic

为Marvell/QLogic适配器配置NVMe/FC。

步骤

1. 验证您是否正在运行受支持的适配器驱动程序和固件版本：

```
cat /sys/class/fc_host/host*/symbolic_name
```

以下示例显示了驱动程序和固件版本：

```
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
```

2. 请验证 `ql2xnvmeenable` 已设置。这样、Marvell适配器便可用作NVMe/FC启动程序：

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

预期输出为1。

NVMe/TCP

NVMe/TCP 协议不支持自动连接操作。相反，您可以通过执行 NVMe/TCP 来发现 NVMe/TCP 子系统和命名空间 `connect`或者 `connect-all`手动操作。

步骤

1. 检查启动器端口是否可以跨支持的 NVMe/TCP LIF 获取发现日志页面数据：

```
nvme discover -t tcp -w host-traddr -a traddr
```

显示示例

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
Discovery Log Number of Records 20, Generation counter 25
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
```

```
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_1
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_1
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
```

```
_tcp_1
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_1
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_4
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_4
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_4
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_4
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
```

```
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 14======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 16======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_2
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 17======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
```

```
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_2
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 18======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_2
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 19======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_2
traddr:  192.168.1.24
eflags:  none
sectype: none
```

2. 验证其他NVMe/TCP启动程序-目标LIF组合是否能够成功提取发现日志页面数据：

```
nvme discover -t tcp -w host-traddr -a traddr
```

显示示例

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.24
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.25
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.25
```

3. 运行 `nvme connect-all` 在节点中所有受支持的NVMe/TCP启动程序-目标SIP上运行命令：

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

显示示例

```
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.24
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.24
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.25
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.25
```

步骤 3：可选，启用 NVMe/FC 的 1MB I/O。

您可以为配置了 Broadcom 适配器的 NVMe/FC 启用 1MB 大小的 I/O 请求。ONTAP在识别控制器数据中报告的最大数据传输大小 (MDTS) 为 8。这意味着最大I/O请求大小最多可以为1 MB。要发出 1MB 大小的 I/O 请求，您需要增加 `lpfc_sg_seg_cnt`参数从默认值 64 更改为 256。

这些步骤不适用于逻辑NVMe/FC主机。

步骤

1. 将 `lpfc_sg_seg_cnt`参数设置为256：

```
cat /etc/modprobe.d/lpfc.conf
```

```
options lpfc lpfc_sg_seg_cnt=256
```

2. 运行 `dracut -f`命令并重新启动主机。
3. 验证的值是否 `lpfc_sg_seg_cnt`为256：

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

步骤 4：验证多路径配置

验证内核NVMe多路径状态、ANA状态和ONTAP命名空间是否适用于NVMe-oF配置。

步骤

1. 验证是否已启用内核NVMe多路径：

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

您应看到以下输出：

```
Y
```

2. 验证相应ONTAP命名库的适当NVMe-oF设置(例如、型号设置为NetApp ONTAP控制器、负载平衡iopolicy设置为循环)是否正确反映在主机上：

a. 显示子系统：

```
cat /sys/class/nvme-subsystem/nvme-subsys*/model
```

您应看到以下输出：

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

b. 显示策略：

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

您应看到以下输出：

```
round-robin
round-robin
```

3. 验证是否已在主机上创建并正确发现命名空间：

```
nvme list
```

显示示例

```
Node         SN                   Model
------------------------------------------------------------
/dev/nvme4n1 81Ix2BVuekWcAAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format              FW              Rev
------------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. 验证每个路径的控制器状态是否为活动状态且是否具有正确的ANA状态：

NVMe/FC

```
nvme list-subsys /dev/nvme4n5
```

显示示例

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.3a5d31f5502c11ef9f50d039eab6cb6d:subsystem.nvme
_1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:e6dade64-
216d-
11ec-b7bb-7ed30a5482c3
iopolicy=round-robin\
+- nvme1 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2088d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live optimized
+- nvme12 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x208ad039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live non-optimized
+- nvme10 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2087d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live non-optimized
+- nvme3 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2083d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n1
```

显示示例

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
iopolicy=round-robin
\
+- nvme13 tcp
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live optimized
+- nvme14 tcp
traddr=192.168.2.24,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live non-optimized
+- nvme5 tcp
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live optimized
+- nvme6 tcp
traddr=192.168.1.24,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live non-optimized
```

5. 验证NetApp插件是否为每个ONTAP 命名空间设备显示正确的值：

列

```
nvme netapp ontapdevices -o column
```

显示示例

```
Device        Vserver   Namespace Path
--------------------- ------------------------------
/dev/nvme1n1     linux_tcnvme_iscsi
/vol/tcpnvme_1_0_0/tcpnvme_ns

NSID       UUID                                   Size
--------------------------------------------------------------
1    5f7f630d-8ea5-407f-a490-484b95b15dd6   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

显示示例

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme1n1",
      "Vserver":"linux_tcnvme_iscsi",
      "Namespace_Path":"/vol/tcpnvme_1_0_0/tcpnvme_ns",
      "NSID":1,
      "UUID":"5f7f630d-8ea5-407f-a490-484b95b15dd6",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
]
}
```

步骤 5：查看已知问题

这些是已知问题：

NetApp 错误 ID	标题	Description
"1479047"	Rocky Linux 8.x NVMe-oF 主机创建重复的持久发现控制器	在 NVMe-oF 主机上，您可以使用“nvme discover -p”命令创建持久发现控制器 (PDC)。 但是，如果您在 NVMe-oF 主机上运行 Rocky Linux 8.x，则每次执行“nvme discover -p”时都会创建一个重复的 PDC。使用此命令时，每个发起方-目标组合只能创建一个 PDC。但是，如果您在NVMe-oF主机上运行Rocky Linux 8.x ，则每次执行“nvme discover -p”时都会创建一个重复的PDC。这会导致主机和目标设备上资源的不必要消耗。