SUSE Linux Enterprise Server

ONTAP SAN Host Utilities

NetApp
January 21, 2026

This PDF was generated from https://docs.netapp.com/zh-cn/ontap-sanhost/nvme-sles-supported-features.html on January 21, 2026. Always check docs.netapp.com for the latest.

目录

SUSE Linux Enterprise Server . . . 1
了解 SUSE Linux Enterprise Server 对ONTAP 的支持和功能 . . . 1
下一步 . . . 1
配置 SUSE Linux Enterprise Server 15 SPx 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 SUSE Linux Enterprise Server 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）修改 udev 规则中的 iopolicy . . . 12
步骤 5：可选，启用 NVMe/FC 的 1MB I/O。 . . . 13
步骤 6：验证 NVMe 启动服务 . . . 14
步骤 7：验证多路径配置 . . . 15
步骤 8：创建持久发现控制器 . . . 19
步骤 9：设置安全带内身份验证 . . . 24
步骤 10：配置传输层安全性 . . . 30
第11步：查看已知问题 . . . 35

SUSE Linux Enterprise Server

了解 SUSE Linux Enterprise Server 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 SUSE Linux Enterprise Server 的版本而异。

功能	**SUSE Linux Enterprise Server** 主机版本	**ONTAP** 版本
NVMe/TCP 支持传输层安全协议 (TLS) 1.3 加密。	15 SP6 或更高版本	9.16.1 或更高版本
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。	15 SP4 或更高版本	9.12.1 或更高版本
持久发现控制器 (PDC) 使用唯一的发现 NQN 提供支持	15 SP4 或更高版本	9.11.1 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	15 SP4 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。	15 SP1 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	**SUSE Linux Enterprise Server** 主机版本
SAN 启动已通过 NVMe/FC 协议启用	15 SP7 或更高版本
默认情况下启用原生 NVMe 多路径。	15 SP1 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。	15 SP1 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 **SUSE Linux Enterprise Server** 版本是……	了解……
15 SPx 系列	"为 SUSE Linux Enterprise Server 15 SPx 配置 NVMe"

相关信息

"了解如何管理 NVMe 协议"

配置 SUSE Linux Enterprise Server 15 SPx 以支持 NVMe-oF 和ONTAP存储

SUSE Linux Enterprise Server 15 SPx 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 SUSE Linux Enterprise Server 15 SPx 配置 NVMe over Fabrics (NVMe-oF) 主机。如需更多支持和功能信息，请参阅 "ONTAP支持和功能"。

使用 SUSE Linux Enterprise Server 15 SPx 的 NVMe-oF 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。
- NetApp sanlun 主机实用程序不支持 NVMe-oF。或者，您可以依赖原生产品中包含的NetApp插件。 `nvme-cli` 适用于所有 NVMe-oF 传输的软件包。
- 对于 SUSE Linux Enterprise Server 15 SP6 及更早版本，不支持使用 NVMe-oF 协议进行 SAN 启动。

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 SUSE Linux Enterprise Server 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 SUSE Linux Enterprise Server 15 SPx。安装完成后，请确认您运行的是指定的 SUSE Linux Enterprise Server 15 SPx 内核：

   ```
   uname -r
   ```

   Rocky Linux 内核版本示例：

   ```
   6.4.0-150700.53.3-default
   ```

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+22.gd31b1a01-150700.3.3.2.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：

```
libnvme1-1.11+4.ge68a91ae-150700.4.3.2.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`版本：

```
nqn.2014-08.org.nvmexpress:uuid:f6517cae-3133-11e8-bbff-7ed30aef123f
```

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 FC适配器配置NVMe/FC。

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
   ```

   您应看到以下输出：

```
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
   ```

   以下示例显示固件版本：

   ```
   14.4.393.25, sli-4:2:c
   14.4.393.25, sli-4:2:c
   ```

   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：

```
0:14.4.0.8
```

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
```

您应该看到类似以下内容的输出：

```
0x10000090fae0ec88
0x10000090fae0ec89
```

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
```

显示示例输出

```
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x10000090fae0ec88 WWNN x20000090fae0ec88
DID x0a1300 ONLINE
NVME RPORT       WWPN x23b1d039ea359e4a WWNN x23aed039ea359e4a
DID x0a1c01 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x22bbd039ea359e4a WWNN x22b8d039ea359e4a
DID x0a1c0b TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2362d039ea359e4a WWNN x234ed039ea359e4a
DID x0a1c10 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x23afd039ea359e4a WWNN x23aed039ea359e4a
DID x0a1a02 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x22b9d039ea359e4a WWNN x22b8d039ea359e4a
DID x0a1a0b TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2360d039ea359e4a WWNN x234ed039ea359e4a
DID x0a1a11 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000004ea0 Cmpl 0000004ea0 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000000102c35 Issue 0000000000102c2d OutIO
fffffffffffffff8
        abort 00000175 noxri 00000000 nondlp 0000021d qdepth
00000000 wqerr 00000007 err 00000000
FCP CMPL: xb 00000175 Err 0000058b

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x10000090fae0ec89 WWNN x20000090fae0ec89
DID x0a1200 ONLINE
NVME RPORT       WWPN x23b2d039ea359e4a WWNN x23aed039ea359e4a
DID x0a1d01 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x22bcd039ea359e4a WWNN x22b8d039ea359e4a
DID x0a1d0b TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2363d039ea359e4a WWNN x234ed039ea359e4a
DID x0a1d10 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x23b0d039ea359e4a WWNN x23aed039ea359e4a
DID x0a1b02 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x22bad039ea359e4a WWNN x22b8d039ea359e4a
DID x0a1b0b TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2361d039ea359e4a WWNN x234ed039ea359e4a
DID x0a1b11 TARGET DISCSRVC ONLINE

NVME Statistics
```

```
LS: Xmt 0000004e31 Cmpl 0000004e31 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000001017f2 Issue 00000000001017ef OutIO
ffffffffffffffffd
        abort 0000018a noxri 00000000 nondlp 0000012e qdepth
00000000 wqerr 00000004 err 00000000
FCP CMPL: xb 0000018a Err 000005ca
```

NVMe/FC - Marvell/QLogic

为Marvell/QLogic适配器配置NVMe/FC。

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:v10.02.09.400-k-debug
QLE2742 FW:v9.14.00 DVR:v10.02.09.400-k-debug
```

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。

1. 验证启动程序端口是否可以通过受支持的NVMe/TCP LIF提取发现日志页面数据：

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

显示示例输出

```
nvme discover -t tcp -w 192.168.111.80 -a 192.168.111.70

Discovery Log Number of Records 8, Generation counter 42
=====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.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr:  192.168.211.71
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:  3
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr:  192.168.111.71
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:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr:  192.168.211.70
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.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr:  192.168.111.70
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.f8e2af201b7211f0ac2bd039eab67a95:subsystem.samp
le_tcp_sub
traddr:  192.168.211.71
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  3
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.samp
le_tcp_sub
traddr:  192.168.111.71
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
```

```
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.samp
le_tcp_sub
traddr:  192.168.211.70
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.f8e2af201b7211f0ac2bd039eab67a95:subsystem.samp
le_tcp_sub
traddr:  192.168.111.70
eflags:  none
sectype: none
localhost:~ #
```

2. 验证所有其他NVMe/TCP启动程序-目标LIF组合是否可以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 192.168.111.80 -a 192.168.111.66
nvme discover -t tcp -w 192.168.111.80 -a 192.168.111.67
nvme discover -t tcp -w 192.168.211.80 -a 192.168.211.66
nvme discover -t tcp -w 192.168.211.80 -a 192.168.211.67
```

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  192.168.111.80  -a
192.168.111.66
nvme    connect-all -t  tcp -w  192.168.111.80  -a
192.168.111.67
nvme    connect-all -t  tcp -w  192.168.211.80  -a
192.168.211.66
nvme    connect-all -t  tcp -w  192.168.211.80  -a
192.168.211.67
```

从 SUSE Linux Enterprise Server 15 SP6 开始，NVMe/TCP 的设置发生了变化。 `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）修改 udev 规则中的 iopolicy

从 SUSE Linux Enterprise Server 15 SP6 开始，NVMe-oF 的默认 iopolicy 设置为 `round-robin`。如果您想更改 iopolicy 为 `queue-depth`修改 udev 规则文件如下：

步骤

1. 使用 root 权限在文本编辑器中打开 udev 规则文件：

   ```
   /usr/lib/udev/rules.d/71-nvmf-netapp.rules
   ```

   您应看到以下输出：

   ```
   vi /usr/lib/udev/rules.d/71-nvmf-netapp.rules
   ```

2. 找到为NetApp ONTAP控制器设置 iopolicy 的行，如下例所示：

   ```
   ACTION=="add", SUBSYSTEM=="nvme-subsystem", ATTR{subsystype}=="nvm",
   ATTR{model}=="NetApp ONTAP Controller", ATTR{iopolicy}="round-robin"
   ```

3. 修改规则，以便 `round-robin`变成 `queue-depth`：

```
ACTION=="add", SUBSYSTEM=="nvme-subsystem", ATTR{subsystype}=="nvm",
ATTR{model}=="NetApp ONTAP Controller", ATTR{iopolicy}="queue-depth"
```

4. 重新加载udev规则并应用更改：

```
udevadm control --reload
udevadm trigger --subsystem-match=nvme-subsystem
```

5. 请检查子系统的当前 I/O 策略。例如，替换<子系统> `nvme-subsys0` 。

```
cat /sys/class/nvme-subsystem/<subsystem>/iopolicy
```

您应看到以下输出：

```
queue-depth.
```

> 新的 iopolicy 会自动应用于匹配的NetApp ONTAP控制器设备。您无需重启。

步骤 5：可选，启用 NVMe/FC 的 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 6：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: enabled)
  Active: inactive (dead) since Fri 2025-07-04 23:56:38 IST; 4 days
ago
  Main PID: 12208 (code=exited, status=0/SUCCESS)
    CPU: 62ms

Jul 04 23:56:26 localhost systemd[1]: Starting Connect NVMe-oF
subsystems automatically during boot...
Jul 04 23:56:38 localhost systemd[1]: nvmf-autoconnect.service:
Deactivated successfully.
Jul 04 23:56:38 localhost 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 Mon 2025-07-07 19:52:30 IST; 1 day
4h ago
  Main PID: 2945 (code=exited, status=0/SUCCESS)
      CPU: 14ms

Jul 07 19:52:30 HP-DL360-14-168 systemd[1]: Starting Auto-connect to
subsystems on FC-NVME devices found during boot...
Jul 07 19:52:30 HP-DL360-14-168 systemd[1]: nvmefc-boot-
connections.service: Deactivated successfully.
Jul 07 19:52:30 HP-DL360-14-168 systemd[1]: Finished Auto-connect to
subsystems on FC-NVME devices found during boot.
```

步骤 7：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. 验证相应的ONTAP命名空间的 NVMe-oF 设置（例如，将型号设置为NetApp ONTAP Controller，并将负载均衡 iopolicy 设置为 queue-depth）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：

```
queue-depth
queue-depth
```

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
```

显示示例输出

```
nvme-subsys114 - NQN=nqn.1992-
08.com.netapp:sn.9e30b9760a4911f08c87d039eab67a95:subsystem.sles
_161_27
                 hostnqn=nqn.2014-
08.org.nvmexpress:uuid:f6517cae-3133-11e8-bbff-7ed30aef123f
iopolicy=round-robin\
+- nvme114 fc traddr=nn-0x234ed039ea359e4a:pn-
0x2360d039ea359e4a,host_traddr=nn-0x20000090fae0ec88:pn-
0x10000090fae0ec88 live optimized
+- nvme115 fc traddr=nn-0x234ed039ea359e4a:pn-
0x2362d039ea359e4a,host_traddr=nn-0x20000090fae0ec88:pn-
0x10000090fae0ec88 live non-optimized
+- nvme116 fc traddr=nn-0x234ed039ea359e4a:pn-
0x2361d039ea359e4a,host_traddr=nn-0x20000090fae0ec89:pn-
0x10000090fae0ec89 live optimized
+- nvme117 fc traddr=nn-0x234ed039ea359e4a:pn-
0x2363d039ea359e4a,host_traddr=nn-0x20000090fae0ec89:pn-
0x10000090fae0ec89 live non-optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme9n1
```

显示示例输出

```
nvme-subsys9 - NQN=nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.with
_inband_with_json hostnqn=nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
iopolicy=round-robin
\
+- nvme10 tcp
traddr=192.168.111.71,trsvcid=4420,src_addr=192.168.111.80 live
non-optimized
 +- nvme11 tcp
traddr=192.168.211.70,trsvcid=4420,src_addr=192.168.211.80 live
optimized
 +- nvme12 tcp
traddr=192.168.111.70,trsvcid=4420,src_addr=192.168.111.80 live
optimized
 +- nvme9 tcp
traddr=192.168.211.71,trsvcid=4420,src_addr=192.168.211.80 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
NSID UUID                                     Size
---------------- -------------------------
------------------------------------------------------ ----
---------------------------------------- ---------
/dev/nvme0n1     vs_161
/vol/fc_nvme_vol1/fc_nvme_ns1                        1    32fd92c7-
0797-428e-a577-fdb3f14d0dc3   5.37GB
```

JSON

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

显示示例

```
{
      "Device":"/dev/nvme98n2",
      "Vserver":"vs_161",
      "Namespace_Path":"/vol/fc_nvme_vol71/fc_nvme_ns71",
      "NSID":2,
      "UUID":"39d634c4-a75e-4fbd-ab00-3f9355a26e43",
      "LBA_Size":4096,
      "Namespace_Size":5368709120,
      "UsedBytes":430649344,
    }
  ]
}
```

步骤 8：创建持久发现控制器

您可以为 SUSE Linux Enterprise Server 15 SPx 主机创建持久发现控制器 (PDC)。需要 PDC 来自动检测 NVMe 子系统的添加或删除操作以及发现日志页面数据的更改。

步骤

1. 验证发现日志页面数据是否可用、并且可以通过启动程序端口和目标LIF组合进行检索：

```
nvme discover -t <trtype> -w <host-traddr> -a <traddr>
```

显示示例输出

```
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.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:  192.168.111.66
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.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:  192.168.211.66
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.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:  192.168.111.67
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.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:  192.168.211.67
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.4f7af2bd221811f0afadd039eab0dadd:subsystem.pdc
traddr:  192.168.111.66
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.4f7af2bd221811f0afadd039eab0dadd:subsystem.pdc
traddr:  192.168.211.66
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.4f7af2bd221811f0afadd039eab0dadd:subsystem.pdc
traddr:  192.168.111.67
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.4f7af2bd221811f0afadd039eab0dadd:subsystem.pdc
traddr:  192.168.211.67
eflags:  none
sectype: none
```

2. 为发现子系统创建PDC：

```
nvme discover -t <trtype> -w <host-traddr> -a <traddr> -p
```

您应看到以下输出：

```
nvme discover -t tcp -w 192.168.111.80 -a 192.168.111.66 -p
```

3. 从ONTAP控制器中、验证是否已创建PDC：

```
vserver nvme show-discovery-controller -instance -vserver <vserver_name>
```

显示示例输出

```
vserver nvme show-discovery-controller -instance -vserver vs_pdc

           Vserver Name: vs_pdc
               Controller ID: 0101h
     Discovery Subsystem NQN: nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:discovery
           Logical Interface: lif2
                        Node: A400-12-181
                    Host NQN: nqn.2014-
08.org.nvmexpress:uuid:9796c1ec-0d34-11eb-b6b2-3a68dd3bab57
          Transport Protocol: nvme-tcp
 Initiator Transport Address: 192.168.111.80
Transport Service Identifier: 8009
             Host Identifier: 9796c1ec0d3411ebb6b23a68dd3bab57
           Admin Queue Depth: 32
       Header Digest Enabled: false
         Data Digest Enabled: false
   Keep-Alive Timeout (msec): 30000
```

步骤 9：设置安全带内身份验证

支持通过 NVMe/TCP 在 SUSE Linux Enterprise Server 15 SPx 主机和ONTAP控制器之间进行安全的带内身份验证。

每个主机或控制器都必须与一个 `DH-HMAC-CHAP` 设置安全认证的关键。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。为了验证对等方的身份，NVMe 主机或控制器必须识别与对等方关联的密钥。

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. 为主机生成 dhchap 密钥。

   以下输出说明了 `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-
1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUAg1iGgx
TYqnxukqvYedA55Bw3wtz6sJNpR4=:
```

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:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
   DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
   DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
   DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
   ```

   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-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   ```

JSON

如果ONTAP控制器配置中有多个NVMe子系统、则可以将文件与命令结合 `nvme connect-all`使用 `/etc/nvme/config.json`。

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：

显示示例输出

```
cat /etc/nvme/config.json
[
 {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-
5910-804b-b2c04f444d33",
    "hostid":"4c4c4544-0035-5910-804b-b2c04f444d33",
    "dhchap_key":"DHHC-
1:01:i4i789R11sMuHLCY27RVI8XloC\/GzjRwyhxip5hmIELsHrBq:",
    "subsystems":[
      {
        "nqn":"nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.samp
le_tcp_sub",
        "ports":[
          {
            "transport":"tcp",
            "traddr":"192.168.111.70",
            "host_traddr":"192.168.111.80",
            "trsvcid":"4420"
            "dhchap_ctrl_key":"DHHC-
1:03:jqgYcJSKp73+XqAf2X6twr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvh
YS0PNK7T+04YD5CRPjh+m3qjJU++yR8s=:"
          },
               {
                    "transport":"tcp",
                    "traddr":"192.168.111.71",
                    "host_traddr":"192.168.111.80",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:03:jqgYcJSKp73+XqAf2X6twr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvh
YS0PNK7T+04YD5CRPjh+m3qjJU++yR8s=:"
               },
               {
                    "transport":"tcp",
                    "traddr":"192.168.211.70",
                    "host_traddr":"192.168.211.80",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:03:jqgYcJSKp73+XqAf2X6twr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvh
YS0PNK7T+04YD5CRPjh+m3qjJU++yR8s=:"
               },
               {
                    "transport":"tcp",
```

```
                    "traddr":"192.168.211.71",
                    "host_traddr":"192.168.211.80",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:03:jqgYcJSKp73+XqAf2X6twr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvh
YS0PNK7T+04YD5CRPjh+m3qjJU++yR8s=:"
                }
            ]
        }
    ]
  }
]
```

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

2. 使用config JSON文件连接到ONTAP控制器：

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

显示示例输出

```
traddr=192.168.211.70 is already connected
traddr=192.168.111.71 is already connected
traddr=192.168.211.71 is already connected
traddr=192.168.111.70 is already connected
traddr=192.168.211.70 is already connected
traddr=192.168.111.70 is already connected
traddr=192.168.211.71 is already connected
traddr=192.168.111.71 is already connected
traddr=192.168.211.70 is already connected
traddr=192.168.111.71 is already connected
traddr=192.168.211.71 is already connected
traddr=192.168.111.70 is already connected
```

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:i4i789R11sMuHLCY27RVI8XloC/GzjRwyhxip5hmIELsHrBq:
```

b. 验证控制器dhchap密钥：

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

您应该看到类似于以下示例的输出：

```
DHHC-
1:03:jqgYcJSKp73+XqAf2X6twr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvhYS0P
NK7T+04YD5CRPjh+m3qjJU++yR8s=:
```

步骤 10：配置传输层安全性

传输层安全协议 (TLS) 为 NVMe-oF 主机和ONTAP阵列之间的 NVMe 连接提供安全的端到端加密。您可以使用 CLI 和已配置的预共享密钥 (PSK) 配置 TLS 1.3。

> 除特别说明需要在ONTAP控制器上执行的步骤外，请在 SUSE Linux Enterprise Server 主机上执行以下步骤。

步骤

1. 检查您是否具有以下内容 `ktls-utils` ， `openssl` ， 和 `libopenssl`主机上安装的软件包：

   a. 验证 `ktls-utils`：

   ```
   rpm -qa | grep ktls
   ```

   您应该看到显示以下输出：

   ```
   ktls-utils-0.10+33.g311d943-150700.1.5.x86_64
   ```

   a. 验证 SSL 包：

   ```
   rpm -qa | grep ssl
   ```

显示示例输出

```
libopenssl3-3.2.3-150700.3.20.x86_64
openssl-3-3.2.3-150700.3.20.x86_64
libopenssl1_1-1.1.1w-150700.9.37.x86_64
```

2. 验证是否已正确设置 `/etc/tlshd.conf`：

```
cat /etc/tlshd.conf
```

显示示例输出

```
[debug]
loglevel=0
tls=0
nl=0
[authenticate]
keyrings=.nvme
[authenticate.client]
#x509.truststore= <pathname>
#x509.certificate= <pathname>
#x509.private_key= <pathname>
[authenticate.server]
#x509.truststore= <pathname>
#x509.certificate= <pathname>
#x509.private_key= <pathname>
```

3. 启用 `tlshd`以在系统启动时启动：

```
systemctl enable tlshd
```

4. 验证守护进程是否 `tlshd`正在运行：

```
systemctl status tlshd
```

显示示例输出

```
tlshd.service - Handshake service for kernel TLS consumers
   Loaded: loaded (/usr/lib/systemd/system/tlshd.service; enabled;
preset: disabled)
   Active: active (running) since Wed 2024-08-21 15:46:53 IST; 4h
57min ago
     Docs: man:tlshd(8)
Main PID: 961 (tlshd)
   Tasks: 1
     CPU: 46ms
   CGroup: /system.slice/tlshd.service
       └─961 /usr/sbin/tlshd
Aug 21 15:46:54 RX2530-M4-17-153 tlshd[961]: Built from ktls-utils
0.11-dev on Mar 21 2024 12:00:00
```

5. 使用生成TLS PSK `nvme gen-tls-key`：

   a. 验证主机：

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

   您应看到以下输出：

   ```
   nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
   ```

   b. 验证密钥：

   ```
   nvme gen-tls-key --hmac=1 --identity=1 --subsysnqn= nqn.1992-
   08.com.netapp:sn.a2d41235b78211efb57dd039eab67a95:subsystem.nvme1
   ```

   您应看到以下输出：

```
NVMeTLSkey-1:01:C50EsaGtuOp8n5fGE9EuWjbBCtshmfoHx4XTqTJUmydf0gIj:
```

6. 在ONTAP控制器上、将TLS PSK添加到ONTAP子系统：

显示示例输出

```
nvme subsystem host add -vserver vs_iscsi_tcp -subsystem nvme1 -host
-nqn nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b2c04f444d33 -tls-configured-psk NVMeTLSkey-
1:01:C50EsaGtuOp8n5fGE9EuWjbBCtshmfoHx4XTqTJUmydf0gIj:
```

7. 将TLS PSK插入主机内核密钥环：

```
nvme check-tls-key --identity=1 --subsysnqn=nqn.1992
-08.com.netapp:sn.a2d41235b78211efb57dd039eab67a95:subsystem.nvme1
--keydata=NVMeTLSkey
-1:01:C50EsaGtuOp8n5fGE9EuWjbBCtshmfoHx4XTqTJUmydf0gIj: --insert
```

您应该看到以下 TLS 密钥：

```
Inserted TLS key 22152a7e
```

> PSK 显示为 `NVMe1R01`因为它使用 `identity v1`来自 TLS 握手算法。Identity v1是ONTAP唯一支持的版本。

8. 验证是否已正确插入TLS PSK：

```
cat /proc/keys | grep NVMe
```

显示示例输出

```
069f56bb I--Q---     5 perm 3b010000     0     0 psk       NVMe1R01
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
nqn.1992-
08.com.netapp:sn.a2d41235b78211efb57dd039eab67a95:subsystem.nvme1
oYVLelmiOwnvDjXKBmrnIgGVpFIBDJtc4hmQXE/36Sw=: 32
```

9. 使用插入的TLS PSK连接到ONTAP子系统：

   a. 验证 TLS PSK：

```
nvme connect -t tcp -w 192.168.111.80 -a 192.168.111.66  -n nqn.1992-
08.com.netapp:sn.a2d41235b78211efb57dd039eab67a95:subsystem.nvme1
--tls_key=0x069f56bb -tls
```

您应看到以下输出：

```
connecting to device: nvme0
```

a. 验证列表子系统：

```
nvme list-subsys
```

显示示例输出

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.a2d41235b78211efb57dd039eab67a95:subsystem.nvme1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-
0035-5910-804b-b2c04f444d33
\
 +- nvme0 tcp
traddr=192.168.111.66,trsvcid=4420,host_traddr=192.168.111.80,src
_addr=192.168.111.80 live
```

10. 添加目标、并验证与指定ONTAP子系统的TLS连接：

```
nvme subsystem controller show -vserver sles15_tls -subsystem sles15
-instance
```

显示示例输出

```
(vserver nvme subsystem controller show)
                          Vserver Name: vs_iscsi_tcp
                          Subsystem: nvme1
                      Controller ID: 0040h
                  Logical Interface: tcpnvme_lif1_1
                               Node: A400-12-181
                           Host NQN: nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
                  Transport Protocol: nvme-tcp
        Initiator Transport Address: 192.168.111.80
                     Host Identifier:
4c4c454400355910804bb2c04f444d33
               Number of I/O Queues: 2
                   I/O Queue Depths: 128, 128
                  Admin Queue Depth: 32
              Max I/O Size in Bytes: 1048576
          Keep-Alive Timeout (msec): 5000
                      Subsystem UUID: 8bbfb403-1602-11f0-ac2b-
d039eab67a95
              Header Digest Enabled: false
                Data Digest Enabled: false
       Authentication Hash Function: sha-256
Authentication Diffie-Hellman Group: 3072-bit
                Authentication Mode: unidirectional
       Transport Service Identifier: 4420
                        TLS Key Type: configured
                   TLS PSK Identity: NVMe1R01 nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
nqn.1992-
08.com.netapp:sn.a2d41235b78211efb57dd039eab67a95:subsystem.nvme1
oYVLelmiOwnvDjXKBmrnIgGVpFIBDJtc4hmQXE/36Sw=
                          TLS Cipher: TLS-AES-128-GCM-SHA256
```

第11步：查看已知问题

没有已知问题。

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。