



Oracle Linux

ONTAP SAN Host Utilities

NetApp
January 26, 2026

This PDF was generated from <https://docs.netapp.com/zh-cn/ontap-sanhost/nvme-ol-supported-features.html> on January 26, 2026. Always check docs.netapp.com for the latest.

目录

Oracle Linux	1
了解ONTAP对 Oracle Linux 主机的支持和功能。	1
下一步	1
配置 Oracle Linux 9.x 和 NVMe-oF 以用于ONTAP存储	2
第1步：(可选)启用SAN启动	2
步骤 2：安装 Oracle Linux 和 NVMe 软件并验证配置	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	13
步骤 7：验证多路径配置	15
步骤 8：设置安全带内身份验证	20
第9步：查看已知问题	25
配置 Oracle Linux 8.x 和 NVMe-oF 以用于ONTAP存储	25
步骤 1：安装 Oracle Linux 和 NVMe 软件并验证您的配置	25
步骤 2：配置 NVMe/FC 和 NVMe/TCP	27
步骤 3：可选，启用 NVMe/FC 的 1MB I/O。	34
步骤 4：验证多路径配置	35
步骤 5：可选，启用 1MB I/O 大小	39
第6步：查看已知问题	40
配置 Oracle Linux 7.x 和 NVMe-oF 以用于ONTAP存储	40
步骤 1：安装 Oracle Linux 和 NVMe 软件并验证您的配置	41
步骤 2：配置 NVMe/FC	42
步骤 3：可选，启用 NVMe/FC 的 1MB I/O。	44
步骤 4：验证多路径配置	45
步骤 5：查看已知问题	47

Oracle Linux

了解ONTAP对 Oracle Linux 主机的支持和功能。

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

功能	Oracle Linux 主机版本	ONTAP 版本
Oracle Linux 主机和ONTAP控制器之间通过 NVMe/TCP 支持安全的带内身份验证。	9.4 或更高版本	9.12.1 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
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
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。	7.7 或更高版本	9.4 或更高版本



NetApp sanlun NVMe-oF 不支持主机实用程序。您可以使用本机 `nvme-cli` 适用于所有 NVMe-oF 传输。

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	Oracle Linux 主机版本
本机 udev 规则在 `nvme-cli` 包中为 NVMe 多路径提供队列深度负载平衡	9.6 或更高版本
SAN 启动已通过 NVMe/FC 协议启用	9.5 或更高版本
内核中 NVMe 多路径功能默认对 NVMe 命名空间启用。	8.3 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。	8.3 或更高版本
本机 udev 规则在 <code>nvme-cli</code> 包中为 NVMe 多路径提供循环负载平衡	8.3 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 Oracle Linux 版本是.....	了解.....
9系列	"为 Oracle Linux 9.x 配置 NVMe"
8系列	"为 Oracle Linux 8.x 配置 NVMe"

如果您的 Oracle Linux 版本是.....	了解.....
7系列	"为 Oracle Linux 7.x 配置 NVMe"

相关信息

["了解如何管理 NVMe 协议"](#)

配置 Oracle Linux 9.x 和 NVMe-oF 以用于ONTAP存储

Oracle 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 Oracle 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["Oracle Linux ONTAP支持和功能"](#)。

NVMe-oF 与 Oracle 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 Oracle Linux 和 NVMe 软件并验证配置

使用以下步骤验证支持的最低 Oracle Linux 9.x 软件版本。

步骤

1. 在服务器上安装Oracle Linux 9.x。安装完成后，请确认您运行的是指定的 Oracle Linux 9.x 内核。

```
uname -r
```

Oracle Linux 内核版本示例：

```
6.12.0-1.23.3.2.el9uek.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.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 Oracle Linux 9.x 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:b1d95cd0-1f7c-11ec-b8d1-3a68dd61a1cb
```

5. 在 ONTAP 系统中，验证以下信息：`hostnqn` 字符串匹配 `hostnqn` ONTAP 存储系统中相应子系统的字符串：

```
vserver nvme subsystem host show -vserver vs_203
```

```

Vserver Subsystem Priority Host NQN
-----
vs_203 Nvme1 regular nqn.2014-
08.org.nvmexpress:uuid:b1d95cd0-1f7c-11ec-b8d1-3a68dd61a1cb
Nvme10 regular nqn.2014-
08.org.nvmexpress:uuid:b1d95cd0-1f7c-11ec-b8d1-3a68dd61a1cb
Nvme11 regular nqn.2014-
08.org.nvmexpress:uuid:b1d95cd0-1f7c-11ec-b8d1-3a68dd61a1cb
Nvme12 regular nqn.2014-
08.org.nvmexpress:uuid:b1d95cd0-1f7c-11ec-b8d1-3a68dd61a1cb
Nvme13 regular nqn.2014-
08.org.nvmexpress:uuid:b1d95cd0-1f7c-11ec-b8d1-3a68dd61a1cb
Nvme14 regular nqn.2014-
08.org.nvmexpress:uuid:b1d95cd0-1f7c-11ec-b8d1-3a68dd61a1cb

```



如果 `hostnqn` 字符串不匹配、您可以使用 `\vserver modify` 命令更新 `\hostnqn` 相应ONTAP阵列子系统上的字符串、使其与主机上的字符串 `\etc/nvme/hostnqn` 匹配 `\hostnqn`。

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. 确认您使用的是受支持的适配器型号：

a. 显示模型名称：

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

您应看到以下输出：

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

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 LPe36002-M64-D 2-Port 64Gb Fibre Channel Adapter  
Emulex LPe36002-M64-D 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：

```
14.4.576.17, sli-4:6:d  
14.4.576.17, 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：

```
0:14.4.0.8
```

+

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. 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：

```
0x2100f4c7aa9d7c5c  
0x2100f4c7aa9d7c5d
```

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 x100000620b3c0869 WWNN x200000620b3c0869
DID x080e00 ONLINE
NVME RPORT          WWPN x2001d039eabac36f WWNN x2000d039eabac36f
DID x021401 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x20e2d039eabac36f WWNN x20e1d039eabac36f
DID x02141f TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x2011d039eabac36f WWNN x2010d039eabac36f
DID x021429 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x2002d039eabac36f WWNN x2000d039eabac36f
DID x021003 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x20e4d039eabac36f WWNN x20e1d039eabac36f
DID x02100f TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x2012d039eabac36f WWNN x2010d039eabac36f
DID x021015 TARGET DISCSRVC ONLINE
```

```
NVME Statistics
LS: Xmt 0000027ccf Cmpl 0000027cca Abort 00000014
LS XMIT: Err 00000005 CMPL: xb 00000014 Err 00000014
Total FCP Cmpl 000000000000613ff Issue 000000000000613fc OutIO
fffffffffffffffffd
          abort 00000007 noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 0000000a Err 0000000d
```

```
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000620b3c086a WWNN x200000620b3c086a
DID x080000 ONLINE
NVME RPORT          WWPN x2004d039eabac36f WWNN x2000d039eabac36f
DID x021501 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x20e3d039eabac36f WWNN x20e1d039eabac36f
DID x02150f TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x2014d039eabac36f WWNN x2010d039eabac36f
DID x021515 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x2003d039eabac36f WWNN x2000d039eabac36f
DID x02110b TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x20e5d039eabac36f WWNN x20e1d039eabac36f
DID x02111f TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x2013d039eabac36f WWNN x2010d039eabac36f
DID x021129 TARGET DISCSRVC ONLINE
```

```
NVME Statistics
```

```
LS: Xmt 0000027ca3 Cmpl 0000027ca2 Abort 00000017
LS XMIT: Err 00000001 Cmpl: xb 00000017 Err 00000017
Total FCP Cmpl 000000000006369d Issue 000000000006369a OutIO
fffffffffffffffffd
        abort 00000007 noxri 00000000 nondlp 00000011 qdepth
00000000 wqerr 00000000 err 00000000
FCP Cmpl: xb 00000008 Err 0000000c
```

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：

```
QLE2872 FW:v9.15.03 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. 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

```
nvme discover -t tcp -w 192.168.30.10 -a 192.168.30.58
=====Discovery Log Entry 0=====
trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 8
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:discovery
traddr: 192.168.31.99
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: 6
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:discovery
traddr: 192.168.30.99
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: 7
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:discovery
traddr: 192.168.31.98
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:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:discovery
traddr:  192.168.30.98
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4=====
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  8
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:subsystem.subs
ys_kvm
traddr:  192.168.31.99
eflags:  none
sectype: none
=====Discovery Log Entry 5=====
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  6
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:subsystem.subs
ys_kvm
traddr:  192.168.30.99
eflags:  none
sectype: none
=====Discovery Log Entry 6=====
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  7
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:subsystem.subs
ys_kvm

```

```
traddr: 192.168.31.98
eflags: none
sectype: none
=====Discovery Log Entry 7=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 5
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.064a9b19b3ee11f09dcad039eabac370:subsystem.subs
ys_kvm
traddr: 192.168.30.98
eflags: none
sectype: none
```

2. 验证其他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 192.168.30.10 -a 192.168.30.58
nvme discover -t tcp -w 192.168.30.10 -a 192.168.30.59
nvme discover -t tcp -w 192.168.31.10 -a 192.168.31.58
nvme discover -t tcp -w 192.168.31.10 -a 192.168.31.59
```

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 192.168.30.10 -a 192.168.30.58
nvme connect-all -t tcp -w 192.168.30.10 -a 192.168.30.59
nvme connect-all -t tcp -w 192.168.31.10 -a 192.168.31.58
nvme connect-all -t tcp -w 192.168.31.10 -a 192.168.31.59
```

从 Oracle 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）修改 udev 规则中的 iopolicy

主机上的 Oracle Linux 9.x 将 NVMe-oF 的默认 iopolicy 设置为 `round-robin`。从 Oracle Linux 9.6 开始，您可以将 iopolicy 更改为 `queue-depth` 通过修改 udev 规则文件。

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. 找到设置 NetApp ONTAP 控制器 iopolicy 的那行代码。

以下示例展示了一条示例规则：

```
ACTION=="add", SUBSYSTEM=="nvme-subsystem", ATTR{subsysname}=="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{subsysname}=="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。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

从 Oracle Linux 9.5 开始，`nvme-fc-boot-connections.service` 和 `nvme-fc-autoconnect.service` NVMe/FC 中包含的启动服务 `nvme-cli` 系统启动时会自动启用软件包。

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

步骤

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

```
systemctl status nvme-autoconnect.service
```

显示示例输出

```
nvme-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
   Loaded: loaded (/usr/lib/systemd/system/nvme-
autoconnect.service; enabled; preset: disabled)
   Active: inactive (dead) since Tue 2025-10-07 09:48:11 EDT; 1
week 0 days ago
     Main PID: 2620 (code=exited, status=0/SUCCESS)
        CPU: 19ms

Oct 07 09:48:11 R650xs-13-211 systemd[1]: Starting Connect NVMe-oF
subsystems automatically during boot...
Oct 07 09:48:11 R650xs-13-211 systemd[1]: nvme-autoconnect.service:
Deactivated successfully.
Oct 07 09:48:11 R650xs-13-211 systemd[1]: Finished Connect NVMe-oF
subsystems automatically during boot.
```

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

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


显示示例输出

```
nvmeofc-boot-connections.service - Auto-connect to subsystems on FC-
NVME devices found during boot
   Loaded: loaded (/usr/lib/systemd/system/nvmeofc-boot-
connections.service; enabled; preset: enabled)
   Active: inactive (dead) since Tue 2025-10-07 09:47:07 EDT; 1
week 0 days ago
     Main PID: 1651 (code=exited, status=0/SUCCESS)
        CPU: 14ms

Oct 07 09:47:07 R650xs-13-211 systemd[1]: Starting Auto-connect to
subsystems on FC-NVME devices found during boot...
Oct 07 09:47:07 R650xs-13-211 systemd[1]: nvmeofc-boot-
connections.service: Deactivated successfully.
Oct 07 09:47:07 R650xs-13-211 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
```

您应该能看到 iopolicy 的值设置, 例如:

```
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	Generic	SN	Model
Namespace	Usage	Format	FW Rev

/dev/nvme102n1	/dev/ng102n1	81LLqNYTindCAAAAAAAk	NetApp ONTAP
Controller	0x1	2.25 GB / 5.37 GB	4 KiB + 0 B
9.17.1			
/dev/nvme102n2	/dev/ng102n2	81LLqNYTindCAAAAAAAk	NetApp ONTAP
Controller	0x2	2.25 GB / 5.37 GB	4 KiB + 0 B
9.17.1			
/dev/nvme106n1	/dev/ng106n1	81LLqNYTindCAAAAAAAs	NetApp ONTAP
Controller	0x1	2.25 GB / 5.37 GB	4 KiB + 0 B
9.17.1			
/dev/nvme106n2	/dev/ng106n2	81LLqNYTindCAAAAAAAs	NetApp ONTAP
Controller	0x2	2.25 GB / 5.37 GB	4 KiB + 0 B
9.17.1			

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.f9c6d0cb4fef11f08579d039eaa8138c:discovery
hostnqn=nqn.2014-08.org.nvmexpress:uuid:b1d95cd0-1f7c-11ec-b8d1-3a68dd61a1cb \
+- nvme2 fc traddr=nn-0x201ad039eabac36f:pn-0x201bd039eabac36f,host_traddr=nn-0x2000f4c7aa9d7c5c:pn-0x2100f4c7aa9d7c5c live optimized
+- nvme8 fc traddr=nn-0x201ad039eabac36f:pn-0x201dd039eabac36f,host_traddr=nn-0x2000f4c7aa9d7c5d:pn-0x2100f4c7aa9d7c5d live non-optimized
+- nvme2 fc traddr=nn-0x201ad039eabac36f:pn-0x201bd039eabac36f,host_traddr=nn-0x2000f4c7aa9d7c5c:pn-0x2100f4c7aa9d7c5c live non-optimized
+- nvme8 fc traddr=nn-0x201ad039eabac36f:pn-0x201dd039eabac36f,host_traddr=nn-0x2000f4c7aa9d7c5d:pn-0x2100f4c7aa9d7c5d live optimized
```

NVMe/TCP

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

显示示例

```
nvme-subsys98 - NQN=nqn.1992-  
08.com.netapp:sn.f9c6d0cb4fef11f08579d039eaa8138c:subsystem.Nvme  
9  
                hostnqn=nqn.2014-  
08.org.nvmexpress:uuid:b1d95cd0-1f7c-11ec-b8d1-3a68dd61a1cb  
\  
  +- nvme100 fc traddr=nn-0x201ad039eabac36f:pn-  
0x201dd039eabac36f,host_traddr=nn-0x2000f4c7aa9d7c5d:pn-  
0x2100f4c7aa9d7c5d live non-optimized  
  +- nvme101 fc traddr=nn-0x201ad039eabac36f:pn-  
0x201cd039eabac36f,host_traddr=nn-0x2000f4c7aa9d7c5c:pn-  
0x2100f4c7aa9d7c5c live non-optimized  
  +- nvme98  fc traddr=nn-0x201ad039eabac36f:pn-  
0x201bd039eabac36f,host_traddr=nn-0x2000f4c7aa9d7c5c:pn-  
0x2100f4c7aa9d7c5c live optimized  
  +- nvme99  fc traddr=nn-0x201ad039eabac36f:pn-  
0x201ed039eabac36f,host_traddr=nn-0x2000f4c7aa9d7c5d:pn-  
0x2100f4c7aa9d7c5d live optimized  
[root@SR630-13-203 ~]#
```

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 Size	Vserver	Namespace Path	NSID	UUID

/dev/nvme102n1 00e760c9-e4ca-4d9f-b1d4-e9a930bf53c0 5.37GB	vs_203	/vol/Nvmevol135/ns35	1	
/dev/nvme102n2 1fa97524-7dc2-4dbc-b4cf-5dda9e7095c0 5.37GB	vs_203	/vol/Nvmevol183/ns83	2	

JSON

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

显示示例

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme11n1",
      "Vserver":"vs_203",
      "Namespace_Path":"/vol/Nvmevol116/ns16",
      "NSID":1,
      "UUID":"18a88771-8b5b-4eb7-bff0-2ae261f488e4",
      "LBA_Size":4096,
      "Namespace_Size":5368709120,
      "UsedBytes":2262282240,
      "Version":"9.17.1"
    }
  ]
}
```

步骤 8：设置安全带内身份验证

Oracle Linux 9.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。

每个主机或控制器都必须与一个 DH-HMAC-CHAP 密钥关联，才能建立安全身份验证。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。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。

步骤

使用 CLI 或配置 JSON 文件设置安全的带内身份验证。如果需要为不同的子系统指定不同的 dhchap 密钥，请使用 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 Linux 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:4c4c4544-0056-5410-8048-c4c04f425633
DHHC-
1:03:xhAfbAD5IVLZDxiVbmFEOA5JZ3F/ERqTXhHzZQJKgkYkTbPI9dhRyVtr4dBD+SG
iAJ03by4FbnVtov1Lmk+86+nNc6k=:
```

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:Y5VkkESgmtTGNdX842qemNpFK6BXYVwwnqErgt3IQKP5Fbjje\JSBOjG  
5Ea3NBLRfuiAuUSDUto6eY\GwKoRp6AwGkw=:  
DHHC-  
1:03:Y5VkkESgmtTGNdX842qemNpFK6BXYVwwnqErgt3IQKP5Fbjje\JSBOjG  
5Ea3NBLRfuiAuUSDUto6eY\GwKoRp6AwGkw=:  
DHHC-  
1:03:Y5VkkESgmtTGNdX842qemNpFK6BXYVwwnqErgt3IQKP5Fbjje\JSBOjG  
5Ea3NBLRfuiAuUSDUto6eY\GwKoRp6AwGkw=:  
DHHC-  
1:03:Y5VkkESgmtTGNdX842qemNpFK6BXYVwwnqErgt3IQKP5Fbjje\JSBOjG  
5Ea3NBLRfuiAuUSDUto6eY\GwKoRp6AwGkw=:
```

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:frpLlTrnOYtcWDxPzq4ccxU1UrH2FjV7hYw5s2XEDB+lo+TjMsOwHR\N  
FtM0nBBidx+gdoyUcC5s6h00tTLDGcz0Kbs=:  
DHHC-  
1:03:frpLlTrnOYtcWDxPzq4ccxU1UrH2FjV7hYw5s2XEDB+lo+TjMsOwHR\N  
FtM0nBBidx+gdoyUcC5s6h00tTLDGcz0Kbs=:  
DHHC-  
1:03:frpLlTrnOYtcWDxPzq4ccxU1UrH2FjV7hYw5s2XEDB+lo+TjMsOwHR\N  
FtM0nBBidx+gdoyUcC5s6h00tTLDGcz0Kbs=:  
DHHC-  
1:03:frpLlTrnOYtcWDxPzq4ccxU1UrH2FjV7hYw5s2XEDB+lo+TjMsOwHR\N  
FtM0nBBidx+gdoyUcC5s6h00tTLDGcz0Kbs=:
```

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、请参见NVMe Connect-all手册页。

1. 配置 JSON 文件：

显示示例

```
[
  {
    "hostnqn": "nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-c4c04f425633",
    "hostid": "4c4c4544-0056-5410-8048-c4c04f425633",
    "dhchap_key": "DHHC-1:01:nFg06gV0FNpXqoiLOF0L+swULQpZU/PjU9v/McDeJHjTZFlF:",
    "subsystems": [
      {
        "nqn": "nqn.1992-08.com.netapp:sn.09035a8d8c8011f0ac0fd039eabac370:subsystem.subsys",
        "ports": [
          {
            "transport": "tcp",
            "traddr": "192.168.30.69",
            "host_traddr": "192.168.30.10",
            "trsvcid": "4420",
            "dhchap_ctrl_key": "DHHC-1:03:n3F8d+bvxKW/s+lEhqXaOohI2sxrQ9iLutzduuFq49JgdjjaFtTpDSO9kQl/bvZj+Bo3rdHh3xPXeP6a4xyhcRyqdds="
          }
        ]
      }
    ]
  }
]
```



在上述示例中，`dhchap\_key`对应于，`dhchap\_ctrl\_key`对应`dhchap\_ctrl\_secret`于`dhchap\_secret`。

2. 使用config JSON文件连接到ONTAP控制器：

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

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:nFg06gV0FNpXqoiLOF0L+swULQpZU/PjU9v/McDeJHjTZFlF:
```

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:n3F8d+bvxKW/s+lEhqXaOohI2sxrQ9iLutzduuFq49JgdjjaFtTpDSO9kQ1/bvZ  
j+Bo3rdHh3xPXeP6a4xyhcRyqdds=:
```

第9步：查看已知问题

没有已知问题。

配置 Oracle Linux 8.x 和 NVMe-oF 以用于ONTAP存储

Oracle 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 Oracle 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["Oracle Linux ONTAP支持和功能"](#)。

NVMe-oF 与 Oracle Linux 8.x 存在以下已知限制：

- 不支持使用 NVMe-oF 协议进行 SAN 启动。
- NetApp sanlun 主机实用程序不支持 Oracle Linux 8.x 主机上的 NVMe-oF。相反，您可以依赖本机中包含的NetApp插件 `nvme-cli` 适用于所有 NVMe-oF 传输的包。
- 对于 Oracle Linux 8.2 及更早版本，nvme-cli 软件包中不提供原生 NVMe/FC 自动连接脚本。使用 HBA 供应商提供的外部自动连接脚本。
- 对于 Oracle Linux 8.2 及更早版本，默认情况下不会为 NVMe 多路径启用轮询负载均衡。要启用此功能，请转到以下步骤。[编写 udev 规则](#)。

步骤 1：安装 Oracle Linux 和 NVMe 软件并验证您的配置

使用以下步骤验证支持的最低 Oracle Linux 8.x 软件版本。

步骤

1. 在服务器上安装Oracle Linux 8.x。安装完成后，请确认您运行的是指定的 Oracle Linux 8.x 内核。

```
uname -r
```

Oracle Linux 内核版本示例：

```
5.15.0-206.153.7.1.el8uek.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-1.16-9.el8.x86_64
```

3. 对于 Oracle Linux 8.2 及更早版本，请添加以下字符串作为单独的 udev 规则。
/lib/udev/rules.d/71-nvme-iopolicy-netapp-ONTAP.rules。这样就实现了 NVMe 多路径的轮询负载均衡。

```
cat /lib/udev/rules.d/71-nvme-iopolicy-netapp-ONTAP.rules
Enable round-robin for NetApp ONTAP
ACTION=="add", SUBSYSTEMS=="nvme-subsystem", ATTRS{model}=="NetApp ONTAP
Controller", ATTR{iopolicy}="round-robin"
```

4. 在 Oracle Linux 8.x 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:edd38060-00f7-47aa-a9dc-4d8ae0cd969a
```

5. 在ONTAP系统中，验证以下信息：`hostnqn`字符串匹配`hostnqn`ONTAP存储系统中相应子系统的字符串：

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

```

Vserver Subsystem Priority Host NQN
-----
vs_coexistence_LPE36002
    nvme
        regular nqn.2014-
08.org.nvmexpress:uuid:edd38060-00f7-47aa-a9dc-4d8ae0cd969a
    nvme1
        regular nqn.2014-
08.org.nvmexpress:uuid:edd38060-00f7-47aa-a9dc-4d8ae0cd969a
    nvme2
        regular nqn.2014-
08.org.nvmexpress:uuid:edd38060-00f7-47aa-a9dc-4d8ae0cd969a
    nvme3
        regular nqn.2014-
08.org.nvmexpress:uuid:edd38060-00f7-47aa-a9dc-4d8ae0cd969a
4 entries were displayed.

```



如果 `hostnqn` 字符串不匹配、请使用 ``vserver modify`` 命令更新 ``hostnqn`` 相应 ONTAP 阵列子系统上的字符串、使其与主机上的字符串 ``/etc/nvme/hostnqn`` 匹配 ``hostnqn``。

6. 此外，为了在同一主机上同时运行 NVMe 和 SCSI 流量，NetApp 建议对 ONTAP 命名空间使用内核 NVMe 多路径。``dm-multipath`` 分别对应 ONTAP LUN。这样应该可以将 ONTAP 命名空间排除在外。``dm-multipath`` 并阻止 ``dm-multipath`` 拒绝声明 ONTAP 命名空间设备。

- a. 添加 ``enable_foreign`` 设置 ``/etc/multipath.conf`` 文件。

```

cat /etc/multipath.conf
defaults {
    enable_foreign      NONE
}

```

- b. 重新启动 ``multipathd`` 守护进程应用新设置。

```
systemctl restart multipathd
```

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。

FC——Broadcom/Emulex

为Broadcom/Emulex适配器配置NVMe/FC。

步骤

1. 确认您使用的是受支持的适配器型号：

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 LPe36002-M64 2-Port 64Gb Fibre Channel Adapter  
Emulex 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：

```
14.4.317.10, 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：

```
0:14.2.0.13
```

+

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：

```
0x100000109bf0449c  
0x100000109bf0449d
```

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 x100000109bf0449c WWNN x200000109bf0449c
DID x061500 ONLINE
NVME RPORT          WWPN x200bd039eab31e9c WWNN x2005d039eab31e9c
DID x020e06 TARGET DISCSRV ONLINE
NVME RPORT          WWPN x2006d039eab31e9c WWNN x2005d039eab31e9c
DID x020a0a TARGET DISCSRV ONLINE
NVME Statistics
LS: Xmt 000000002c Cmpl 000000002c Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000000008ffe8 Issue 000000000008ffb9 OutIO
ffffffffffffffffd1
          abort 0000000c noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 0000000c Err 0000000c
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109bf0449d WWNN x200000109bf0449d
DID x062d00 ONLINE
NVME RPORT          WWPN x201fd039eab31e9c WWNN x2005d039eab31e9c
DID x02090a TARGET DISCSRV ONLINE
NVME RPORT          WWPN x200cd039eab31e9c WWNN x2005d039eab31e9c
DID x020d06 TARGET DISCSRV ONLINE
NVME Statistics
LS: Xmt 0000000041 Cmpl 0000000041 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000000936bf Issue 000000000009369a OutIO
fffffffffffffffffdb
          abort 00000016 noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000016 Err 00000016
```

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：

```
QLE2772 FW:v9.15.00 DVR:v10.02.09.100-k  
QLE2772 FW:v9.15.00 DVR:v10.02.09.100-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。

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

```
nvme discover -t tcp -w 192.168.6.1 -a 192.168.6.24 Discovery
Log Number of Records 20, Generation counter 45
====Discovery Log Entry 0=====
trtype:  tcp
adrfam:  ipv4
subtype: unrecognized
treq:    not specified
portid:  6
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.e6c438e66ac211ef9ab8d039eab31e9d:discovery
traddr:  192.168.6.25
sectype: none
====Discovery Log Entry 1=====
trtype:  tcp
adrfam:  ipv4
subtype: unrecognized
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.e6c438e66ac211ef9ab8d039eab31e9d:discovery
traddr:  192.168.5.24
sectype: none
====Discovery Log Entry 2=====
trtype:  tcp
adrfam:  ipv4
subtype: unrecognized
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.e6c438e66ac211ef9ab8d039eab31e9d:discovery
traddr:  192.168.6.24
sectype: none
====Discovery Log Entry 3=====
trtype:  tcp
adrfam:  ipv4
subtype: unrecognized
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
```

```

08.com.netapp:sn.e6c438e66ac211ef9ab8d039eab31e9d:discovery
traddr: 192.168.5.25
sectype: none
=====Discovery Log Entry 4=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 6
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.e6c438e66ac211ef9ab8d039eab31e9d:subsystem.nvme
_tcp_4
traddr: 192.168.6.25
sectype: none
=====Discovery Log Entry 5=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.e6c438e66ac211ef9ab8d039eab31e9d:subsystem.nvme
_tcp_4
.....

```

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 192.168.6.1 -a 192.168.6.24
nvme discover -t tcp -w 192.168.6.1 -a 192.168.6.25
nvme discover -t tcp -w 192.168.5.1 -a 192.168.5.24
nvme discover -t tcp -w 192.168.5.1 -a 192.168.5.25

```

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 -l  
<ctrl_loss_timeout_in_seconds>
```

显示示例

```
nvme connect-all -t tcp -w 192.168.5.1 -a 192.168.5.24  
-l -l  
nvme connect-all -t tcp -w 192.168.5.1 -a 192.168.5.25  
-l -l  
nvme connect-all -t tcp -w 192.168.6.1 -a 192.168.6.24  
-l -l  
nvme connect-all -t tcp -w 192.168.6.1 -a 192.168.6.25  
-l -l
```

NetApp建议设置 `ctrl-loss-tmo option` 到 `-1` 这样，当路径丢失时，NVMe/TCP 发起程序会无限期地尝试重新连接。

步骤 3：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 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/nvme0n1	814vWBNRwf9HAAAAAAB	NetApp ONTAP Controller
/dev/nvme0n2	814vWBNRwf9HAAAAAAB	NetApp ONTAP Controller
/dev/nvme0n3	814vWBNRwf9HAAAAAAB	NetApp ONTAP Controller

Namespace	Usage	Format	FW	Rev

1		85.90 GB / 85.90 GB	4 KiB + 0 B	FFFFFFFF
2		85.90 GB / 85.90 GB	24 KiB + 0 B	FFFFFFFF
3		85.90 GB / 85.90 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 list-subsys /dev/nvme0n1
```

展示 NVMe/FC 示例

```
nvme-subsys0 - NQN=nqn.1992-08.com.netapp:
4b4d82566aab11ef9ab8d039eab31e9d:subsystem.nvme\
+- nvme1 fc traddr=nn-0x2038d039eab31e9c:pn-0x203ad039eab31e9c
host_traddr=nn-0x200034800d756a89:pn-0x210034800d756a89 live
optimized
+- nvme2 fc traddr=nn-0x2038d039eab31e9c:pn-0x203cd039eab31e9c
host_traddr=nn-0x200034800d756a88:pn-0x210034800d756a88 live
optimized
+- nvme3 fc traddr=nn-0x2038d039eab31e9c:pn-0x203ed039eab31e9c
host_traddr=nn-0x200034800d756a89:pn-0x210034800d756a89 live non-
optimized
+- nvme7 fc traddr=nn-0x2038d039eab31e9c:pn-0x2039d039eab31e9c
host_traddr=nn-0x200034800d756a88:pn-0x210034800d756a88 live non-
optimized
```

展示 NVMe/TCP 示例

```
nvme-subsys0 - NQN=nqn.1992- 08.com.netapp:
sn.e6c438e66ac211ef9ab8d039eab31e9d:subsystem.nvme_tcp_4
\
+- nvme1 tcp traddr=192.168.5.25 trsvcid=4420
host_traddr=192.168.5.1 src_addr=192.168.5.1 live optimized
+- nvme10 tcp traddr=192.168.6.24 trsvcid=4420
host_traddr=192.168.6.1 src_addr=192.168.6.1 live optimized
+- nvme2 tcp traddr=192.168.5.24 trsvcid=4420
host_traddr=192.168.5.1 src_addr=192.168.5.1 live non-optimized
+- nvme9 tcp traddr=192.168.6.25 trsvcid=4420
host_traddr=192.168.6.1 src_addr=192.168.6.1 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	Size
NSID	UUID			

/dev/nvme0n1	vs_coexistence_QLE2772			
/vol/fcnvme_1_1_0/fcnvme_ns	1	159f9f88-be00-4828-aef6-197d289d4bd9		10.74GB
/dev/nvme0n2	vs_coexistence_QLE2772			
/vol/fcnvme_1_1_1/fcnvme_ns	2	2c1ef769-10c0-497d-86d7-e84811ed2df6		10.74GB
/dev/nvme0n3	vs_coexistence_QLE2772			
/vol/fcnvme_1_1_2/fcnvme_ns	3	9b49bf1a-8a08-4fa8-baf0-6ec6332ad5a4		10.74GB

JSON

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


显示示例

```
{
  "ONTAPdevices" : [
    {
      "Device" : "/dev/nvme0n1",
      "Vserver" : "vs_coexistence_QLE2772",
      "Namespace_Path" : "/vol/fcnvme_1_1_0/fcnvme_ns",
      "NSID" : 1,
      "UUID" : "159f9f88-be00-4828-aef6-197d289d4bd9",
      "Size" : "10.74GB",
      "LBA_Data_Size" : 4096,
      "Namespace_Size" : 2621440
    },
    {
      "Device" : "/dev/nvme0n2",
      "Vserver" : "vs_coexistence_QLE2772",
      "Namespace_Path" : "/vol/fcnvme_1_1_1/fcnvme_ns",
      "NSID" : 2,
      "UUID" : "2c1ef769-10c0-497d-86d7-e84811ed2df6",
      "Size" : "10.74GB",
      "LBA_Data_Size" : 4096,
      "Namespace_Size" : 2621440
    },
    {
      "Device" : "/dev/nvme0n4",
      "Vserver" : "vs_coexistence_QLE2772",
      "Namespace_Path" : "/vol/fcnvme_1_1_3/fcnvme_ns",
      "NSID" : 4,
      "UUID" : "f3572189-2968-41bc-972a-9ee442dfaed7",
      "Size" : "10.74GB",
      "LBA_Data_Size" : 4096,
      "Namespace_Size" : 2621440
    }
  ],
}
```

步骤 5: 可选, 启用 1MB I/O 大小

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
```

第6步：查看已知问题

这些是已知问题:

NetApp 错误 ID	标题	Description
"1479047"	Oracle Linux 8.x NVMe-oF 主机创建重复的持久发现控制器 (PDC)	在 NVMe-oF 主机上, 您可以使用 <code>nvme discover -p</code> 命令创建 PDC。使用此命令时, 每个发起方-目标组合只能创建一个 PDC。However, if you are running Oracle Linux 8.x with an NVMe-oF host, a duplicate PDC is created each time <code>nvme discover -p</code> is executed.这会导致主机和目标设备上资源的不必要消耗。

配置 Oracle Linux 7.x 和 NVMe-oF 以用于ONTAP存储

Oracle 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 Oracle Linux 7.x 配置 NVMe over Fabrics (NVMe-oF) 主机。如需更多支持和功能信息, 请参阅["Oracle Linux ONTAP支持和功能"](#)。

NVMe-oF 与 Oracle Linux 7.x 存在以下已知限制:

- 不支持使用 NVMe-oF 协议进行 SAN 启动。
- NetApp sanlun 主机实用程序不支持 Oracle Linux 7.x 主机上的 NVMe-oF。相反, 您可以依赖本机中包含的 NetApp 插件 `nvme-cli` 适用于所有 NVMe-oF 传输的包。
- NVMe-CLI 软件包中不提供原生 NVMe/FC 自动连接脚本。使用 HBA 供应商提供的外部自动连接脚本。

- NVMe 多路径默认情况下未启用轮询负载均衡。要启用此功能，请编写 udev 规则。

步骤 1：安装 Oracle Linux 和 NVMe 软件并验证您的配置

使用以下步骤验证支持的最低 Oracle Linux 7.x 软件版本。

步骤

1. 在服务器上安装 Oracle Linux 7.x。安装完成后，请确认您运行的是指定的 Oracle Linux 7.x 内核。

```
uname -r
```

Oracle Linux 内核版本示例：

```
5.4.17-2011.6.2.el7uek.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-1.8.1-3.el7.x86_64
```

3. 将以下字符串添加为单独的 udev 规则 `/lib/udev/rules.d/71-nvme-iopolicy-netapp-ONTAP.rules`。这样就实现了 NVMe 多路径的轮询负载均衡。

```
cat /lib/udev/rules.d/71-nvme-iopolicy-netapp-ONTAP.rules
Enable round-robin for NetApp ONTAP
ACTION=="add", SUBSYSTEMS=="nvme-subsystem", ATTRS{model}=="NetApp ONTAP
Controller", ATTR{iopolicy}="round-robin"
```

4. 在 Oracle Linux 7.x 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:497ad959-e6d0-4987-8dc2-a89267400874
```

5. 在ONTAP系统中，验证以下信息：`hostnqn`字符串匹配`hostnqn`ONTAP存储系统中相应子系统的字符串：

```
*> vserver nvme subsystem host show -vserver vs_nvme_10
```

显示示例

```
Vserver Subsystem Host NQN
-----
ol_157_nvme_ss_10_0
nqn.2014-08.org.nvmexpress:uuid:497ad959-e6d0-4987-8dc2-a89267400874
```



如果`hostnqn`字符串不匹配、请使用`vserver modify`命令更新`hostnqn`相应ONTAP阵列子系统上的字符串、使其与主机上的字符串`/etc/nvme/hostnqn`匹配`hostnqn`。

6. 重新启动主机。

步骤 2：配置 NVMe/FC

为Broadcom/Emulex适配器配置NVMe/FC。

1. 确认您使用的是受支持的适配器型号：

- a. 显示模型名称：

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

您应看到以下输出：

```
LPe32002-M2
LPe32002-M2
```

- 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 LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. 验证是否 `lpfc\_enable\_fc4\_type` 设置为"3":

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

3. 安装推荐的 lpfc 自动连接脚本:

```
rpm -ivh nvmeofc-connect-12.8.264.0-1.noarch.rpm
```

4. 确认自动连接脚本已安装:

```
rpm -qa | grep nvmeofc
```

您应看到以下输出:

```
nvmeofc-connect-12.8.264.0-1.noarch
```

5. 验证启动程序端口是否联机:

a. 显示端口名称:

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

您应看到以下输出:

```
0x10000090fae0ec61
0x10000090fae0ec62
```

b. 显示端口名称:

```
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 NVME 2947 SCSI 2947 ELS 250
NVME LPORT lpfc0 WWPN x10000090fae0ec61 WWNN x20000090fae0ec61 DID
x012000 ONLINE
NVME RPORT WWPN x202d00a098c80f09 WWNN x202c00a098c80f09 DID x010201
TARGET DISCSRV ONLINE
NVME RPORT WWPN x203100a098c80f09 WWNN x202c00a098c80f09 DID x010601
TARGET DISCSRV ONLINE
```

步骤 3：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 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 Namespace Usage Format FW Rev
-----
/dev/nvme0n1 80BADBKnB/JvAAAAAAC NetApp ONTAP Controller 1 53.69 GB
/ 53.69 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 list-subsys /dev/nvme0n1
```

显示示例

```
Nvme-subsysf0 - NQN=nqn.1992-
08.com.netapp:sn.341541339b9511e8a9b500a098c80f09:subsystem.ol_157_n
vme_ss_10_0
\
+- nvme0 fc traddr=nn-0x202c00a098c80f09:pn-0x202d00a098c80f09
host_traddr=nn-0x20000090fae0ec61:pn-0x10000090fae0ec61 live
optimized
+- nvme1 fc traddr=nn-0x207300a098dfdd91:pn-0x207600a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme2 fc traddr=nn-0x207300a098dfdd91:pn-0x207500a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live
optimized
+- nvme3 fc traddr=nn-0x207300a098dfdd91:pn-0x207700a098dfdd91 host
traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live inaccessible
```

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_nvme_10				
/vol/rhel_141_vol_10_0/ol_157_ns_10_0			1	55baf453-f629-4a18-9364-b6aee3f50dad	53.69GB

JSON

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

显示示例

```
{
  "ONTAPdevices" : [
    {
      "Device" : "/dev/nvme0n1",
      "Vserver" : "vs_nvme_10",
      "Namespace_Path" :
"/vol/rhel_141_vol_10_0/ol_157_ns_10_0",
      "NSID" : 1,
      "UUID" : "55baf453-f629-4a18-9364-b6aee3f50dad",
      "Size" : "53.69GB",
      "LBA_Data_Size" : 4096,
      "Namespace_Size" : 13107200
    }
  ]
}
```

步骤 5: 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。