



普罗克斯莫克斯 ONTAP SAN Host Utilities

NetApp
January 29, 2026

目录

普罗克斯莫克斯	1
了解 Proxmox 主机对ONTAP 的支持和功能	1
下一步	1
配置 Proxmox VE 9.x 以支持 NVMe-oF 和ONTAP存储	1
步骤 1: 安装 Proxmox VE 和 NVMe 软件并验证您的配置	2
步骤 2: 配置 NVMe/FC 和 NVMe/TCP	3
步骤 3: 可选, 启用 NVMe/FC 的 1MB I/O。	9
步骤 4: 验证 NVMe 启动服务	10
步骤 5: 验证多路径配置	12
第6步: 查看已知问题	16
配置 Proxmox VE 8.x 以支持 NVMe-oF 和ONTAP存储	17
步骤 1: 安装 Proxmox VE 和 NVMe 软件并验证您的配置	17
步骤 2: 配置 NVMe/FC 和 NVMe/TCP	18
步骤 3: 可选, 启用 NVMe/FC 的 1MB I/O。	27
步骤 4: 验证 NVMe 启动服务	28
步骤 5: 验证多路径配置	29
第6步: 查看已知问题	33

普罗克斯莫克斯

了解 Proxmox 主机对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 Proxmox 的版本而异。

ONTAP功能	Proxmox主机版本	ONTAP 版本
NVMe/TCP 是一项完全受支持的企业级功能	9.0	9.10.1 或更高版本
使用原生功能提供 NVMe/FC 和 NVMe/TCP 命名空间的ONTAP详细信息 `nvme-cli`包裹。	8.0	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。	8.0	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	Proxmox主机版本
这 `nvme-cli` 该软件包包含自动连接脚本，无需第三方脚本。	9.0
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	9.0
默认情况下启用原生 NVMe 多路径。	8.0



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 Proxmox VE 版本是.....	了解.....
9系列	"为 Proxmox VE 9.x 配置 NVMe"
8系列	"为 Proxmox VE 8.x 配置 NVMe"

相关信息

["了解如何管理 NVMe 协议"](#)

配置 Proxmox VE 9.x 以支持 NVMe-oF 和ONTAP存储

Proxmox VE 9.x 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 Proxmox VE 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["ONTAP支持和功能"](#)。

NVMe-oF 与 Proxmox VE 9.x 存在以下已知限制：

- 不支持 NVMe-FC 的 SAN 启动配置。

步骤 1：安装 **Proxmox VE** 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 Proxmox VE 9.x。安装完成后，请确认您运行的是所需的 Proxmox VE 9.x 内核：

```
uname -r
```

Proxmox VE 9.x 内核版本示例：

```
6.14.8-2-pve
```

2. 安装 NVMe-CLI 软件包：

```
apt list|grep nvme-cli
```

下面的例子展示了 `nvme-cli` 软件包版本：

```
nvme-cli/stable,now 2.13-2 amd64
```

3. 安装 libnvme 软件包：

```
apt list|grep libnvme
```

下面的例子展示了 `libnvme` 软件包版本：

```
libnvme-dev/stable 1.13-2 amd64
```

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:39333550-3333-4753-4844-32594d4a524c
```

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_proxmox_FC_NVMeFC
```

显示示例

```
Vserver Subsystem Priority Host NQN
-----
vs_proxmox_FC_NVMeFC
    sub_176
        regular nqn.2014-08.org.nvmexpress:uuid:39333550-3333-4753-4844-32594d4a4834
        regular nqn.2014-08.org.nvmexpress:uuid:39333550-3333-4753-4844-32594d4a524c
2 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 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：

```
SN1700E2P  
SN1700E2P
```

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：

```
HPE SN1700E 64Gb 2p FC HBA  
HPE SN1700E 64Gb 2p FC HBA
```

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：

```
14.4.473.14, sli-4:6:d  
14.4.473.14, 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：

```
0:14.4.0.7
```

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：

```
0x10005ced8c531948  
0x10005ced8c531949
```

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 x10005ced8c531948 WWNN x20005ced8c531948
DID x082400 ONLINE
NVME RPORT          WWPN x200ed039eac79573 WWNN x200dd039eac79573
DID x060902 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x2001d039eac79573 WWNN x2000d039eac79573
DID x060904 TARGET DISCSRVC ONLINE
```

```
NVME Statistics
LS: Xmt 0000000034 Cmpl 0000000034 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000000142cfb Issue 0000000000142cfc OutIO
00000000000000001
          abort 00000005 noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000005 Err 00000005
```

```
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x10005ced8c531949 WWNN x20005ced8c531949
DID x082500 ONLINE
NVME RPORT          WWPN x2010d039eac79573 WWNN x200dd039eac79573
DID x062902 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x2007d039eac79573 WWNN x2000d039eac79573
DID x062904 TARGET DISCSRVC ONLINE
```

```
NVME Statistics
LS: Xmt 0000000034 Cmpl 0000000034 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000000d39f1 Issue 00000000000d39f2 OutIO
00000000000000001
          abort 00000005 noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000005 Err 00000005
```

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：

```
SN1700Q FW:v9.15.05 DVR:v10.02.09.400-k  
SN1700Q FW:v9.15.05 DVR:v10.02.09.400-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 192.168.165.72 -a 192.168.165.51
Discovery Log Number of Records 4, Generation counter 47
====Discovery Log Entry 0=====
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  3
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.c770be5d934811f0b624d039eac809ba:discovery
traddr:  192.168.165.51
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:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.c770be5d934811f0b624d039eac809ba:discovery
traddr:  192.168.166.50
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
====Discovery Log Entry 2=====
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  3
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.c770be5d934811f0b624d039eac809ba:subsystem.sub_
176
traddr:  192.168.165.51
eflags:  none
sectype: none
====Discovery Log Entry 3=====
trtype:  tcp
```

```
adrfam:  ipv4
subtype:  nvme subsystem
treq:     not specified
portid:   1
trsvcid:  4420
subnqn:   nqn.1992-
08.com.netapp:sn.c770be5d934811f0b624d039eac809ba:subsystem.sub_
176
traddr:   192.168.166.50
eflags:   none
sectype:  none
```

2. 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 192.168.166.73 -a 192.168.166.50
nvme connect-all -t tcp -w 192.168.166.73 -a 192.168.166.51
nvme connect-all -t tcp -w 192.168.165.73 -a 192.168.165.50
nvme connect-all -t tcp -w 192.168.165.73 -a 192.168.165.51
```

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 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 `update-initramfs` 命令并重启主机。
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 NVMe 启动服务

这 `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, 软件包会自动启用。

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-fc-autoconnect.service
```

显示示例输出

```
o nvme-autoconnect.service - Connect NVMe-oF subsystems
automatically during boot
   Loaded: loaded (/usr/lib/systemd/system/nvme-
autoconnect.service; enabled; preset: enabled)
   Active: inactive (dead) since Thu 2025-10-16 18:00:39 IST; 6
days ago
  Invocation: e146e0b2c339432aad6e0555a528872c
    Process: 1787 ExecStart=/usr/sbin/nvme connect-all
--context=autoconnect (code=exited, status=0/SUCCESS)
   Main PID: 1787 (code=exited, status=0/SUCCESS)
    Mem peak: 2.4M
      CPU: 12ms
Oct 16 18:00:39 HPE-DL365-14-176 systemd[1]: Starting nvme-
autoconnect.service - Connect NVMe-oF subsystems automatically
during boot...
Oct 16 18:00:39 HPE-DL365-14-176 systemd[1]: nvme-
autoconnect.service: Deactivated successfully.
Oct 16 18:00:39 HPE-DL365-14-176 systemd[1]: Finished nvme-
autoconnect.service - Connect NVMe-oF subsystems automatically
during boot.
```

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

```
systemctl status nvmeboot-connections.service
```

显示示例输出

```
nvme-fc-boot-connections.service - Auto-connect to subsystems on FC-
NVME devices found during boot
   Loaded: loaded (/usr/lib/systemd/system/nvme-fc-boot-
connections.service; enabled; preset: enabled)
   Active: inactive (dead) since Thu 2025-10-16 18:00:35 IST; 6
days ago
  Invocation: acf73ac1ef7a402198d6ecc4d075fab0
    Process: 1173 ExecStart=/bin/sh -c echo add >
/sys/class/fc/fc_udev_device/nvme_discovery (code=exited,
status=0/SUCCESS)
   Main PID: 1173 (code=exited, status=0/SUCCESS)
    Mem peak: 2.1M
       CPU: 11ms

Oct 16 18:00:35 HPE-DL365-14-176 systemd[1]: nvme-fc-boot-
connections.service: Deactivated successfully.
Oct 16 18:00:35 HPE-DL365-14-176 systemd[1]: Finished nvme-fc-boot-
connections.service - Auto-connect to subsystems on FC-NVME devices
found during boot.
```

步骤 5：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. 验证主机上是否正确显示了ONTAP命名空间的相应 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 round-robin）：

- 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	Generic	SN	Model
Namespace	Usage	Format	FW Rev

/dev/nvme2n1	/dev/ng2n1	81PqYFYq2aVAAAAAAAAAB	NetApp ONTAP
Controller	0x1	17.88 GB / 171.80 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/nvme3n9
```

显示示例

```
nvme-subsys3 - NQN=nqn.1992-08.com.netapp:sn.94929fdb84eb11f0b624d039eac809ba:subsystem.sub_176
                hostnqn=nqn.2014-08.org.nvmexpress:uuid:39333550-3333-4753-4844-32594d4a524c
\
+- nvme1 fc traddr=nn-0x200dd039eac79573:pn-0x2010d039eac79573,host_traddr=nn-0x20005ced8c531949:pn-0x10005ced8c531949 live optimized
+- nvme3 fc traddr=nn-0x200dd039eac79573:pn-0x200ed039eac79573,host_traddr=nn-0x20005ced8c531948:pn-0x10005ced8c531948 live optimized
+- nvme5 fc traddr=nn-0x200dd039eac79573:pn-0x200fd039eac79573,host_traddr=nn-0x20005ced8c531949:pn-0x10005ced8c531949 live non-optimized
+- nvme7 fc traddr=nn-0x200dd039eac79573:pn-0x2011d039eac79573,host_traddr=nn-0x20005ced8c531948:pn-0x10005ced8c531948 live non-optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme2n3
```

显示示例

```
nvme-subsys2 - NQN=nqn.1992-  
08.com.netapp:sn.c770be5d934811f0b624d039eac809ba:subsystem.sub_  
176  
                hostnqn=nqn.2014-08.org.nvmexpress:uuid:39333550-  
3333-4753-4844-32594d4a524c  
\  
  +- nvme2 tcp  
traddr=192.168.166.50,trsvcid=4420,host_traddr=192.168.166.73,src_  
_addr=192.168.166.73 live optimized  
  +- nvme4 tcp  
traddr=192.168.165.51,trsvcid=4420,host_traddr=192.168.165.73,src_  
_addr=192.168.165.73 live optimized  
  +- nvme6 tcp  
traddr=192.168.166.51,trsvcid=4420,host_traddr=192.168.166.73,src_  
_addr=192.168.166.73 live non-optimized  
  +- nvme8 tcp  
traddr=192.168.165.50,trsvcid=4420,host_traddr=192.168.165.73,src_  
_addr=192.168.165.73 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/nvme2n9	vs_proxmox_FC_NVMeFC	/vol/vol_180_data_nvme4c4/ns
NSID	UUID	Size
----	-----	-----
1	e3d3d544-de8b-4787-93af-bfec7769e909	32.21GB

JSON

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

显示示例

```
{
  "Device":"/dev/nvme2n9",
  "Vserver":"vs_proxmox_FC_NVMeFC",
  "Subsystem":"sub_176",
  "Namespace_Path":"/vol/vol_180_data_nvme4c4/ns",
  "NSID":9,
  "UUID":"e3d3d544-de8b-4787-93af-bfec7769e909",
  "LBA_Size":4096,
  "Namespace_Size":32212254720,
  "UsedBytes":67899392,
  "Version":"9.17.1"
}
```

第6步：查看已知问题

没有已知问题。

配置 Proxmox VE 8.x 以支持 NVMe-oF 和ONTAP存储

Proxmox VE 8.x 主机支持光纤通道上的 NVMe (NVMe/FC) 和 TCP 上的 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 Proxmox VE 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["ONTAP支持和功能"](#)。

NVMe-oF 与 Proxmox VE 8.x 存在以下已知限制：

- 不支持 NVMe-FC 的 SAN 启动配置。

步骤 1：安装 Proxmox VE 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 Proxmox 8.x。安装完成后，请确认您运行的是指定的 Proxmox 8.x 内核：

```
uname -r
```

以下示例显示了 Proxmox 内核版本：

```
6.8.12-10-pve
```

2. 安装 NVMe-CLI 软件包：

```
apt list|grep nvme-cli
```

下面的例子展示了 `nvme-cli` 软件包版本：

```
nvme-cli/oldstable,now 2.4+really2.3-3 amd64
```

3. 安装 libnvme 软件包：

```
apt list|grep libnvme
```

下面的例子展示了 `libnvme` 软件包版本：

```
libnvme1/oldstable,now 1.3-1+deb12u1 amd64
```

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:1536c9a6-f954-11ea-b24d-0a94efb46eaf
```

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 proxmox_120_122
```

显示示例

```
Vserver Subsystem Priority Host NQN
-----
proxmox_120_122
proxmox_120_122
                                regular nqn.2014-
08.org.nvmexpress:uuid:1536c9a6-f954-11ea-b24d-0a94efb46eaf
                                regular nqn.2014-
08.org.nvmexpress:uuid:991a7476-f9bf-11ea-8b73-0a94efb46c3b
proxmox_120_122_tcp
                                regular nqn.2014-
08.org.nvmexpress:uuid:1536c9a6-f954-11ea-b24d-0a94efb46eaf
                                regular nqn.2014-08.org.nvmexpress:uuid:991a7476-
f9bf-11ea-8b73-0a94efb46c3b
2 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 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：

```
LPe35002-M2  
LPe35002-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 LPe35002-M2 2-Port 32Gb Fibre Channel Adapter  
Emulex LPe35002-M2 2-Port 32Gb 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：

```
14.0.505.12, sli-4:6:d  
14.0.505.12, 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：

```
0:14.2.0.17
```

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：

```
0x100000109b95467e  
0x100000109b95467f
```

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 x10005ced8c531948 WWNN x20005ced8c531948
DID x082400
ONLINE
NVME RPORT WWPN x200ed039eac79573 WWNN x200dd039eac79573 DID
x060902
TARGET DISCSRV ONLINE
NVME RPORT WWPN x2001d039eac79573 WWNN x2000d039eac79573 DID
x060904
TARGET DISCSRV ONLINE
NVME Statistics LS: Xmt 0000000034 Cmpl 0000000034 Abort
00000000 LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000000142cfb Issue 0000000000142cfc OutIO
0000000000000001 abort 00000005 noxri 00000000 nondlp 00000000
qdepth 00000000 wqerr 00000000 err 00000000 FCP CMPL: xb
00000005 Err 00000005 NVME Initiator Enabled XRI Dist lpfc1
Total 6144 IO 5894 ELS 250 NVME LPORT lpfc1 WWPN
x10005ced8c531949 WWNN x20005ced8c531949 DID x082500
ONLINE
NVME RPORT WWPN x2010d039eac79573 WWNN x200dd039eac79573 DID
x062902
TARGET DISCSRV ONLINE
NVME RPORT WWPN x2007d039eac79573 WWNN x2000d039eac79573 DID
x062904
TARGET DISCSRV ONLINE
NVME Statistics LS: Xmt 0000000034 Cmpl 0000000034 Abort
00000000 LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000000d39f1 Issue 00000000000d39f2 OutIO
0000000000000001 abort 00000005 noxri 00000000 nondlp 00000000
qdepth 00000000 wqerr 00000000 err 00000000 FCP CMPL: xb
00000005 Err 00000005
```

NVMe/FC - Marvell/QLogic

为Marvell/QLogic适配器配置NVMe/FC。

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 192.168.2.22 -a 192.168.2.30

Discovery Log Number of Records 12, Generation counter 13
=====Discovery Log Entry 0=====
trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 10
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:discovery
traddr: 192.168.2.30
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: 9
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:discovery
traddr: 192.168.1.30
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: 12
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:discovery
traddr: 192.168.2.25
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: 11
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:discovery
traddr: 192.168.1.25
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 10
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:subsystem.prox
mox_120_122
traddr: 192.168.2.30
eflags: none
sectype: none
=====Discovery Log Entry 5=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 9
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:subsystem.prox
mox_120_122
traddr: 192.168.1.30
eflags: none
sectype: none
=====Discovery Log Entry 6=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 12
trsvcid: 4420
subnqn: nqn.1992-

```

```

08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:subsystem.prox
mox_120_122
traddr: 192.168.2.25
eflags: none
sectype: none
=====Discovery Log Entry 7=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 11
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:subsystem.prox
mox_120_122
traddr: 192.168.1.25
eflags: none
sectype: none
=====Discovery Log Entry 8=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 10
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:subsystem.prox
mox_120_122_tcp
traddr: 192.168.2.30
eflags: none
sectype: none
=====Discovery Log Entry 9=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 9
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:subsystem.prox
mox_120_122_tcp
traddr: 192.168.1.30
eflags: none
sectype: none
=====Discovery Log Entry 10=====
trtype: tcp

```

```

adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  12
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:subsystem.prox
mox_120_122_tcp
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 11=====
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  11
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:subsystem.prox
mox_120_122_tcp
traddr:  192.168.1.25
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 192.168.1.22 -a 192.168.1.30
nvme discover -t tcp -w 192.168.2.22 -a 192.168.2.30
nvme discover -t tcp -w 192.168.1.22 -a 192.168.1.25
nvme discover -t tcp -w 192.168.2.22 -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 192.168.1.22 -a 192.168.1.30
nvme connect-all -t tcp -w 192.168.2.22 -a 192.168.2.30
nvme connect-all -t tcp -w 192.168.1.22 -a 192.168.1.25
nvme connect-all -t tcp -w 192.168.2.22 -a 192.168.2.25
```

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 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 `update-initramfs` 命令并重启主机。
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 NVMe 启动服务

使用 Proxmox 8.x, `nvme-fc-boot-connections.service` 和 `nvmmf-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 `nvmmf-autoconnect.service` 启动服务已启用。

步骤

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

```
systemctl status nvmmf-autoconnect.service
```

显示示例输出

```
○ nvmmf-autoconnect.service - Connect NVMe-oF subsystems
automatically during boot
   Loaded: loaded (/lib/systemd/system/nvmmf-autoconnect.service;
enabled; preset: enabled)
   Active: inactive (dead) since Fri 2025-11-21 19:59:10 IST; 8s
ago
     Process: 256613 ExecStartPre=/sbin/modprobe nvme-fabrics
(code=exited, status=0/SUCCESS)
     Process: 256614 ExecStart=/usr/sbin/nvme connect-all
(code=exited, status=0/SUCCESS)
    Main PID: 256614 (code=exited, status=0/SUCCESS)
      CPU: 18ms
Nov 21 19:59:07 SR665-14-122.lab.eng.btc.netapp.in systemd[1]:
Starting nvmmf-autoconnect.service - Connect NVMe-oF subsystems
automatically during boot...
Nov 21 19:59:10 SR665-14-122.lab.eng.btc.netapp.in nvme[256614]:
Failed to write to /dev/nvme-fabrics: Invalid argument
Nov 21 19:59:10 SR665-14-122.lab.eng.btc.netapp.in nvme[256614]:
Failed to write to /dev/nvme-fabrics: Invalid argument
Nov 21 19:59:10 SR665-14-122.lab.eng.btc.netapp.in systemd[1]: nvmmf-
autoconnect.service: Deactivated successfully.
Nov 21 19:59:10 SR665-14-122.lab.eng.btc.netapp.in systemd[1]:
Finished nvmmf-autoconnect.service - 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

```
o nvme-fc-boot-connections.service - Auto-connect to subsystems on
FC-NVME devices found during boot
   Loaded: loaded (/lib/systemd/system/nvme-fc-boot-
connections.service; enabled; preset: enabled)
   Active: inactive (dead) since Thu 2025-11-20 17:48:29 IST; 1
day 2h ago
   Process: 1381 ExecStart=/bin/sh -c echo add >
/sys/class/fc/fc_udev_device/nvme_discovery (code=exited,
status=0/SUCCESS)
   Main PID: 1381 (code=exited, status=0/SUCCESS)
      CPU: 3ms

Nov 20 17:48:29 SR665-14-122.lab.eng.btc.netapp.in systemd[1]:
Starting nvme-fc-boot-connections.service - Auto-connect to
subsystems on FC-NVME devices found during boot..
Nov 20 17:48:29 SR665-14-122.lab.eng.btc.netapp.in systemd[1]:
nvme-fc-boot-connections.service: Deactivated successfully.
Nov 20 17:48:29 SR665-14-122.lab.eng.btc.netapp.in systemd[1]:
Finished nvme-fc-boot-connections.service - Auto-connect to
subsystems on FC-NVME devices found during boot...
```

步骤 5：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. 验证主机上是否正确显示了ONTAP命名空间的相应 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 round-robin）：

- 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	Generic	SN	Usage
Model		Namespace	
Format	FW Rev		

/dev/nvme2n20	/dev/ng2n20	81K13BUDdygsAAAAAAG	
NetApp ONTAP Controller		10	5.56 GB /
91.27 GB	4 KiB + 0 B	9.18.1	

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/nvme2n20
```

显示示例

```
nvme-subsys2 - NQN= nqn.1992-  
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:subsystem.prox  
mox_120_122_tcp  
\  
+- nvme1 fc traddr=nn-0x200dd039eac79573:pn-  
0x2010d039eac79573,host_traddr=nn-0x20005ced8c531949:pn-  
0x10005ced8c531949 live optimized  
+- nvme3 fc traddr=nn-0x200dd039eac79573:pn-  
0x200ed039eac79573,host_traddr=nn-0x20005ced8c531948:pn-  
0x10005ced8c531948 live optimized  
+- nvme5 fc traddr=nn-0x200dd039eac79573:pn-  
0x200fd039eac79573,host_traddr=nn-0x20005ced8c531949:pn-  
0x10005ced8c531949 live non-optimized  
+- nvme7 fc traddr=nn-0x200dd039eac79573:pn-  
0x2011d039eac79573,host_traddr=nn-0x20005ced8c531948:pn-  
0x10005ced8c531948 live non-optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme2n3
```

显示示例

```
nvme-subsys2 - NQN= qn.1992-  
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:subsystem.prox  
mox_120_122_tcp  
\n  
+- nvme2 tcp  
traddr=192.168.1.30,trsvcid=4420,host_traddr=192.168.1.22,src_ad  
dr=192.168.1.22 live optimized  
+- nvme4 tcp  
traddr=192.168.2.30,trsvcid=4420,host_traddr=192.168.2.22,src_ad  
dr=192.168.2.22 live optimized  
+- nvme6 tcp  
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.22,src_ad  
dr=192.168.1.22 live non-optimized  
+- nvme8 tcp  
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.22,src_ad  
dr=192.168.2.22 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/nvme2n11	proxmox_120_122	/vol/vm120_tcp1/ns
NSID	UUID	Size
1	5aefea74-f0cf-4794-a7e9-e113c4659aca	37.58GB

JSON

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

显示示例

```
{
  "Device": "/dev/nvme2n11",
  "Vserver": "proxmox_120_122",
  "Namespace_Path": "/vol/vm120_tcp1/ns",
  "NSID": 1,
  "UUID": "5aefea74-f0cf-4794-a7e9-e113c4659aca",
  "Size": "37.58GB",
  "LBA_Data_Size": 4096,
  "Namespace_Size": 32212254720
}
```

第6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。