SUSE Linux Enterprise Server 15

SAN hosts and cloud clients

NetApp
December 18, 2024

目录

SUSE Linux Enterprise Server 15

适用于采用ONTAP的SUSE Linux Enterprise Server 15 SP6的NVMe-oF主机配置

具有非对称命名空间访问(AANA)的SUSE Linux Enterprise Server 15 SP6支持基于网络结构的NVMe (NVMe-oF)、包括基于光纤通道的NVMe (NVMe/FC)和其他传输。在NVMe-oF环境中、ANA相当于iSCSI和FCP环境中的ALUA多路径功能、并可通过内核NVMe多路径实施。

以下支持适用于采用ONTAP的SUSE Linux Enterprise Server 15 SP6的NVMe-oF主机配置：

- 在同一主机上运行NVMe和SCSI流量。例如、您可以为SCSI LUN的SCSI设备配置dm-Multipath `mpath`、并使用NVMe多路径在主机上配置NVMe-oF命名空间设备。
- 支持基于TCP的NVMe (NVMe/TCP)和NVMe/FC。这样、本机软件包中的NetApp插件 `nvme-cli`就能够显示NVMe/FC和NVMe/TCP命名区的ONTAP详细信息。

有关支持的配置的其他详细信息、请参见 "NetApp 互操作性表工具"。

功能

- 支持NVMe安全带内身份验证
- 支持使用唯一发现NQN的永久性发现控制器(PDC)
- 为NVMe/TCP提供TLS 1.3加密支持

已知限制

- 目前不支持使用NVMe-oF协议启动SAN。
- 在SUSE Linux Enterprise Server 15 SP6主机上、NetApp `sanlun`主机实用程序不支持NVMe-oF。而是可以依赖本机软件包中的NetApp插件 `nvme-cli`进行所有NVMe-oF传输。

配置 NVMe/FC

您可以为采用ONTAP配置的SUSE Linux Enterprise Server 15 SP6配置NVMe/FC与Broadcom/Emulex FC或Marvell/Qlogic FC适配器。

Broadcom/Emulex

为Broadcom/Emulex FC适配器配置NVMe/FC。

步骤

1. 确认您使用的是建议的适配器型号：

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

示例输出

```
LPe32002 M2
LPe32002-M2
```

2. 验证适配器型号问题描述：

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

示例输出

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

3. 验证是否正在使用建议的Emulex主机总线适配器(HBA)固件版本：

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

示例输出

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

4. 验证是否正在使用建议的lpfc驱动程序版本：

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

示例输出

```
0:14.4.0.1
```

5. 验证是否可以查看启动程序端口：

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

示例输出

```
0x10000090fae0ec88
0x10000090fae0ec89
```

6. 验证启动程序端口是否联机：

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

示例输出

```
Online
Online
```

7. 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
```

在以下示例中、一个启动程序端口已启用、并与两个目标生命周期关联。

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 x2070d039ea359e4a WWNN x206bd039ea359e4a DID
x0a0a05 TARGET DISCSRVC
ONLINE
NVME Statistics
LS: Xmt 00000003ba Cmpl 00000003ba Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000014e3dfb8 Issue 0000000014e308db OutIO
fffffffffff2923
 abort 00000845 noxri 00000000 nondlp 00000063 qdepth 00000000
wqerr 00000003 err 00000000
FCP CMPL: xb 00000847 Err 00027f33
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 x2071d039ea359e4a WWNN x206bd039ea359e4a DID
x0a0305 TARGET DISCSRVC
ONLINE
NVME Statistics
LS: Xmt 00000003ba Cmpl 00000003ba Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000014e39f78 Issue 0000000014e2b832 OutIO
fffffffffff18ba
 abort 0000082d noxri 00000000 nondlp 00000028 qdepth 00000000
wqerr 00000007 err 00000000
FCP CMPL: xb 0000082d Err 000283bb
```

Marvell/QLogic

SUSE Linux Enterprise Server 15 SP6内核中附带的本机内置qla2xxx驱动程序具有最新的修复程序。这些修复程序对于ONTAP支持至关重要。

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
```

示例输出

```
QLE2742 FW:v9.14.01 DVR: v10.02.09.200-k
QLE2742 FW:v9.14.01 DVR: v10.02.09.200-k
```

2. 验证是否已 `ql2xnvmeenable` 参数设置为1：

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

预期值为1。

启用 1 MB I/O 大小（可选）

ONTAP会在"识别 控制器"数据中报告MDTS (MAX Data传输大小)为8。这意味着最大I/O请求大小最多可以为1 MB。要向Broadcom NVMe/FC主机发出大小为1 MB的I/O请求、应将参数的值 `lpfc_sg_seg_cnt`从默认值64增加 `lpfc`到256。

这些步骤不适用于逻辑NVMe/FC主机。

步骤

1. 将 `lpfc_sg_seg_cnt`参数设置为256：

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

```
options lpfc lpfc_sg_seg_cnt=256
```

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

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

验证NVMe服务

从SUSE Linux Enterprise Server 15 SP6开始、 `nvmefc-boot-connections.service`NVMe/FC包中包含的和 `nvmf-autoconnect.service`启动服务 `nvme-cli`会自动启用、以便在系统启动期间启动。系统启动完成后、您应验证是否已启用启动服务。

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; vendor preset: disabled)
  Active: inactive (dead) since Thu 2024-05-25 14:55:00 IST; 11min
ago
Process: 2108 ExecStartPre=/sbin/modprobe nvme-fabrics (code=exited,
status=0/SUCCESS)
Process: 2114 ExecStart=/usr/sbin/nvme connect-all (code=exited,
status=0/SUCCESS)
Main PID: 2114 (code=exited, status=0/SUCCESS)

systemd[1]: Starting Connect NVMe-oF subsystems automatically during
boot...
nvme[2114]: traddr=nn-0x201700a098fd4ca6:pn-0x201800a098fd4ca6 is
already connected
systemd[1]: nvmf-autoconnect.service: Deactivated successfully.
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; vendor preset: enabled)
   Active: inactive (dead) since Thu 2024-05-25 14:55:00 IST; 11min
ago
 Main PID: 1647 (code=exited, status=0/SUCCESS)

systemd[1]: Starting Auto-connect to subsystems on FC-NVME devices
found during boot...
systemd[1]: nvmefc-boot-connections.service: Succeeded.
systemd[1]: Finished Auto-connect to subsystems on FC-NVME devices
found during boot.
```

配置 NVMe/TCP

NVMe/TCP没有自动连接功能。相反、您可以通过手动执行NVMe/TCP或 `connect-all`操作来发现NVMe/TCP子系统和命名路径 `connect。

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.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.211.67
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.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.111.67
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.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.211.66
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.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.111.66
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.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.211.67
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.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.111.67
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.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
```

```
traddr: 192.168.211.66
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.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.111.66
eflags: none
sectype: none
```

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>
```

示例输出

```
#nvme discover -t tcp -w 192.168.111.79 -a 192.168.111.66
#nvme discover -t tcp -w 192.168.111.79 -a 192.168.111.67
#nvme discover -t tcp -w 192.168.211.79 -a 192.168.211.66
#nvme discover -t tcp -w 192.168.211.79 -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>
```

示例输出

```
# nvme connect-all -t tcp -w 192.168.111.79 -a 192.168.111.66
# nvme connect-all -t tcp -w 192.168.111.79 -a 192.168.111.67
# nvme connect-all -t tcp -w 192.168.211.79 -a 192.168.211.66
# nvme connect-all -t tcp -w 192.168.211.79 -a 192.168.211.67
```

从SUSE Linux Enterprise Server 15 SP6开始、NVMe/TCP超时的默认设置 `ctrl-loss-tmo`已关闭。这意味着重试次数没有限制(无限期重试)，使用或 `nvme connect-all` 命令(选项 `-l`)时无需手动配置特定的`ctrl-loss-tmo`超时持续时间。 `nvme connect`此外、NVMe/TCP控制器在发生路径故障时不会发生超时、并会无限期保持连接。

验证 NVMe-oF

使用以下过程验证采用ONTAP的SUSE Linux Enterprise Server 15 SP6的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
```

预期值为"Y"。

2. 验证主机是否具有适用于ONTAP NVMe命名卷的正确控制器型号：

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

示例输出

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

3. 验证相应ONTAP NVMe I/O控制器的NVMe I/O策略：

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

示例输出

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

4. 验证ONTAP名称卷是否对主机可见：

```
nvme list -v
```

显示示例输出

```
Subsystem        Subsystem-NQN
Controllers
----------------
------------------------------------------------------------------------
----------------- ---------------------
nvme-subsys0     nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hcha p  nvme0, nvme1, nvme2, nvme3

Device   SN                   MN
FR       TxPort Asdress        Subsystem    Namespaces
-------- --------------------
---------------------------------------- --------
-----------------------------------------------
nvme0    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.111.66,trsvcid=4420,host_traddr=192.168.111.79 nvme-
subsys0 nvme0n1
nvme1    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.111.67,trsvcid=4420,host_traddr=192.168.111.79 nvme-
subsys0 nvme0n1
nvme2    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.211.66,trsvcid=4420,host_traddr=192.168.211.79 nvme-
subsys0 nvme0n1
nvme3    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.211.67,trsvcid=4420,host_traddr=192.168.211.79 nvme-
subsys0 nvme0n1
Device        Generic     NSID       Usage                  Format
Controllers
------------ ------------ ----------
--------------------------------------------------------------
/dev/nvme0n1 /dev/ng0n1   0x1     1.07  GB /   1.07  GB    4 KiB +
0 B   nvme0, nvme1, nvme2, nvme3
```

5. 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/<subsystem_name>
```

NVMe/FC

```
nvme list-subsys /dev/nvme2n1
```

显示示例输出

```
nvme-subsys2 - NQN=nqn.1992-
08.com.netapp:sn.06303c519d8411eea468d039ea36a106:subs
ystem.nvme
 hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-
0056-5410-8048-c6c04f425633
 iopolicy=round-robin
\
+- nvme4 fc traddr=nn-0x208fd039ea359e4a:pn-
0x210dd039ea359e4a,host_traddr=nn-0x2000f4c7aa0cd7ab:pn-
0x2100f4c7aa0cd7ab live optimized
+- nvme6 fc traddr=nn-0x208fd039ea359e4a:pn-
0x210ad039ea359e4a,host_traddr=nn-0x2000f4c7aa0cd7aa:pn-
0x2100f4c7aa0cd7aa live optimized
```

NVMe/TCP

```
nvme list-subsys
```

显示示例输出

```
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme
_tcp_1
 hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-
804b-b2c04f444d33
 iopolicy=round-robin
\
+- nvme4 tcp
traddr=192.168.111.66,trsvcid=4420,host_traddr=192.168.111.79,sr
c_addr=192.168.111.79 live
+- nvme3 tcp
traddr=192.168.211.66,trsvcid=4420,host_traddr=192.168.211.79,sr
c_addr=192.168.111.79 live
+- nvme2 tcp
traddr=192.168.111.67,trsvcid=4420,host_traddr=192.168.111.79,sr
c_addr=192.168.111.79 live
+- nvme1 tcp
traddr=192.168.211.67,trsvcid=4420,host_traddr=192.168.211.79,sr
c_addr=192.168.111.79 live
```

6. 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
```

示例输出

```
Device           Vserver    Namespace Path
NSID UUID                                   Size
---------------- ---------- ------------------------------------
---------------------------------------- --------
/dev/nvme0n1     vs_192     /vol/fcnvme_vol_1_1_0/fcnvme_ns      1
c6586535-da8a-40fa-8c20-759ea0d69d33   20GB
```

JSON

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

显示示例输出


```
{
"ONTAPdevices":[
{
"Device":"/dev/nvme0n1",
"Vserver":"vs_192",
"Namespace_Path":"/vol/fcnvme_vol_1_1_0/fcnvme_ns",
"NSID":1,
"UUID":"c6586535-da8a-40fa-8c20-759ea0d69d33",
"Size":"20GB",
"LBA_Data_Size":4096,
"Namespace_Size":262144
}
]
}
```


创建永久性发现控制器

从SUSEL.11.1开始、您可以为ONTAP 9 15 SP6主机创建永久性发现控制器(PDC)。要自动检测NVMe子系统添加或删除操作以及对发现日志页面数据的更改、需要PDC。

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.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.211.67
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.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.111.67
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.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.211.66
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.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.111.66
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.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.211.67
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.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.111.67
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.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
```

```
traddr: 192.168.211.66
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.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.111.66
eflags: none
sectype: none
```

2. 为发现子系统创建PDC：

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

示例输出

```
nvme discover -t tcp -w 192.168.111.79 -a 192.168.111.666 -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_nvme79
Vserver Name: vs_CLIENT116 Controller ID: 00C0h
Discovery Subsystem NQN: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery Logical
Interface UUID: d23cbb0a-c0a6-11ec-9731-d039ea165abc Logical
Interface:
CLIENT116_lif_4a_1
Node: A400-14-124
Host NQN: nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-be09-
74362c0c1afc
Transport Protocol: nvme-tcp
Initiator Transport Address: 192.168.1.16
Host Identifier: 59de25be738348f08a79df4bce9573f3 Admin Queue Depth:
32
Header Digest Enabled: false Data Digest Enabled: false
Vserver UUID: 48391d66-c0a6-11ec-aaa5-d039ea165514
```

设置安全带内身份验证

从Linux.12.1开始、支持在ONTAP 9 15 SP6主机和ONTAP控制器之间通过NVMe/TCP和NVMe/FC进行安全带内身份验证。

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

您可以使用命令行界面或Config 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。

步骤

1. 获取主机NQN：

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

2. 为SUSE Linux Enterprise Server 15 SP6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:d3ca725a- ac8d-4d88-b46a-174ac235139b
DHHC-
1:03:J2UJQfj9f0pLnpF/ASDJRTyILKJRr5CougGpGdQSysPrLu6RW1fGl5VSjbeDF1n
1DEh3nVBe19nQ/LxreSBeH/bx/pU=:
```

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:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjO
   Hg8wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
   DHHC-
   1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjO
   Hg8wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
   DHHC-
   1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjO
   Hg8wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
   DHHC-
   1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjO
   Hg8wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
   ```

   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:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crX
eTUB8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crX
eTUB8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crX
eTUB8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crX
eTUB8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
```

JSON 文件

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`。

要生成JSON文件、可以使用 `-o`选项。有关更多语法选项、请参见NVMe Connect-all手册页。

步骤

1. 配置 JSON 文件：

显示示例输出

```
# cat /etc/nvme/config.json
[
 {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-
4d1b-be09-74362c0c1afc",
    "hostid":"3ae10b42-21af-48ce-a40b-cfb5bad81839",
    "dhchap_key":"DHHC-
1:03:Cu3ZZfIz1WMlqZFnCMqpAgn/T6EVOcIFHez215U+Pow8jTgBF2UbNk3DK4w
fk2EptWpna1rpwG5CndpOgxpRxh9m41w=:"
 },
 {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-
4d1b-be09-74362c0c1afc",
    "subsystems":[
        {
            "nqn":"nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subs
ys_CLIENT116",
            "ports":[
               {
                    "transport":"tcp",
                    "traddr":" 192.168.111.66 ",
                    "host_traddr":" 192.168.111.79",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
               },
               {
                    "transport":"tcp",
                    "traddr":" 192.168.111.66 ",
                    "host_traddr":" 192.168.111.79",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
               },
               {
                    "transport":"tcp",
                   "traddr":" 192.168.111.66 ",
                    "host_traddr":" 192.168.111.79",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
               },
```

```
                {
                      "transport":"tcp",
                      "traddr":" 192.168.111.66 ",
                      "host_traddr":" 192.168.111.79",
                      "trsvcid":"4420",
                      "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
                }
            ]
        }
    ]
 }
]
```

+

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
```

显示示例输出

```
traddr=192.168.111.66 is already connected
traddr=192.168.211.66 is already connected
traddr=192.168.111.66 is already connected
traddr=192.168.211.66 is already connected
traddr=192.168.111.66 is already connected
traddr=192.168.211.66 is already connected
traddr=192.168.111.67 is already connected
traddr=192.168.211.67 is already connected
traddr=192.168.111.67 is already connected
traddr=192.168.211.67 is already connected
traddr=192.168.111.67 is already connected
traddr=192.168.111.67 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
```

示例输出

```
DHHC-1:01:NunEWY7AZlXqxITGheByarwZdQvU4ebZg9HOjIr6nOHEkxJg:
```

b. 验证控制器dhchap密钥：

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

示例输出

```
DHHC-
1:03:2YJinsxa2v3+m8qqCiTnmgBZoH6mIT6G/6f0aGO8viVZB4VLNLH4z8CvK7pV
YxN6S5fOAtaU3DNi12rieRMfdbg3704=:
```

配置传输层安全性

传输层安全(Transport Layer Security、TLS)可为NVMe-oF主机和ONTAP阵列之间的NVMe连接提供安全的端到端加密。从tls.16.1开始、您可以使用ONTAP 9和已配置的预共享密钥(PSK)配置tls.1.3。

关于此任务

您可以在SUSE Linux Enterprise Server 15 SP6主机上执行此过程中的步骤、但指定您在ONTAP控制器上执行某个步骤的情况除外。

步骤

1. 检查主机上是否安装了以下ktls-utils、opensl和libopenssl软件包：

a. `rpm -qa | grep ktls`

示例输出

```
ktls-utils-0.10+12.gc3923f7-150600.1.2.x86_64
```

b. `rpm -qa | grep ssl`

示例输出

```
openssl-3-3.1.4-150600.5.7.1.x86_64
libopenssl1_1-1.1.1w-150600.5.3.1.x86_64
libopenssl3-3.1.4-150600.5.7.1.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. `# cat /etc/nvme/hostnqn`

      示例输出

      ```
      nqn.2014-08.org.nvmexpress:uuid:e58eca24-faff-11ea-8fee-3a68dd3b5c5f
      ```

   b. `# nvme gen-tls-key --hmac=1 --identity=1 --subsysnqn=nqn.1992 -08.com.netapp:sn.1d59a6b2416b11ef9ed5d039ea50acb3:subsystem.sles15`

      示例输出

      ```
      NVMeTLSkey-1:01:dNcby017axByCko8GivzOO9zGlgHDXJCN6KLzvYoA+NpT1uD:
      ```

6. 在ONTAP控制器上、将TLS PSK添加到ONTAP子系统：

   ```
   # nvme subsystem host add -vserver sles15_tls -subsystem sles15 -host
   -nqn nqn.2014-08.org.nvmexpress:uuid:ffa0c815-e28b-4bb1-8d4c-
   7c6d5e610bfc -tls-configured-psk NVMeTLSkey-
   1:01:dNcby017axByCko8GivzOO9zGlgHDXJCN6KLzvYoA+NpT1uD:
   ```

7. 将TLS PSK插入主机内核密钥环：

```
# nvme check-tls-key --identity=1 --subsysnqn=nqn.2014
-08.org.nvmexpress:uuid:ffa0c815-e28b-4bb1-8d4c-7c6d5e610bf
--keydata=NVMeTLSkey
-1:01:dNcby017axByCko8GivzOO9zGlgHDXJCN6KLzvYoA+NpT1uD: --insert
```

示例输出

```
Inserted TLS key 22152a7e
```

PSK显示为"NVMe1R01"、因为它使用TLS握手算法中的"Identity v1"。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
```

示例输出

```
22152a7e I--Q---     1 perm 3b010000     0     0 psk       NVMe1R01
nqn.2014-08.org.nvmexpress:uuid:ffa0c815-e28b-4bb1-8d4c-7c6d5e610bfc
nqn.1992-
08.com.netapp:sn.1d59a6b2416b11ef9ed5d039ea50acb3:subsystem.sles15
UoP9dEfvuCUzzpS0DYxnshKDapZYmvA0/RJJ8JAqmAo=: 32
```

9. 使用插入的TLS PSK连接到ONTAP子系统：

a. `# nvme connect -t tcp -w 20.20.10.80 -a 20.20.10.14 -n nqn.1992-08.com.netapp:sn.1d59a6b2416b11ef9ed5d039ea50acb3:subsystem.sles15 --tls_key=0x22152a7e --tls`

   示例输出

   ```
   connecting to device: nvme0
   ```

b. `# nvme list-subsys`

示例输出

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.1d59a6b2416b11ef9ed5d039ea50acb3:subsystem.sles15
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:ffa0c815-e28b-
4bb1-8d4c-7c6d5e610bfc
               iopolicy=round-robin
\
 +- nvme0 tcp
traddr=20.20.10.14,trsvcid=4420,host_traddr=20.20.10.80,src_addr=20.2
0.10.80 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: sles15_tls
                          Subsystem: sles15
                      Controller ID: 0040h
                  Logical Interface: sles15t_e1a_1
                               Node: A900-17-174
                           Host NQN: nqn.2014-
08.org.nvmexpress:uuid:ffa0c815-e28b-4bb1-8d4c-7c6d5e610bfc
                  Transport Protocol: nvme-tcp
        Initiator Transport Address: 20.20.10.80
                     Host Identifier:
ffa0c815e28b4bb18d4c7c6d5e610bfc
               Number of I/O Queues: 4
                   I/O Queue Depths: 128, 128, 128, 128
                  Admin Queue Depth: 32
              Max I/O Size in Bytes: 1048576
          Keep-Alive Timeout (msec): 5000
                       Vserver UUID: 1d59a6b2-416b-11ef-9ed5-
d039ea50acb3
                    Subsystem UUID: 9b81e3c5-5037-11ef-8a90-
d039ea50ac83
             Logical Interface UUID: 8185dcac-5035-11ef-8abb-
d039ea50acb3
              Header Digest Enabled: false
                Data Digest Enabled: false
       Authentication Hash Function: -
Authentication Diffie-Hellman Group: -
                Authentication Mode: none
       Transport Service Identifier: 4420
                       TLS Key Type: configured
                  TLS PSK Identity: NVMe1R01 nqn.2014-
08.org.nvmexpress:uuid:ffa0c815-e28b-4bb1-8d4c-7c6d5e610bfc
nqn.1992-
08.com.netapp:sn.1d59a6b2416b11ef9ed5d039ea50acb3:subsystem.sles15
UoP9dEfvuCUzzpS0DYxnshKDapZYmvA0/RJJ8JAqmAo=
                         TLS Cipher: TLS-AES-128-GCM-SHA256
```

已知问题

具有ONTAP版本的SUSE Linux Enterprise Server 15 SP6没有已知问题。

适用于采用ONTAP的SUSE Linux Enterprise Server 15 SP5的NVMe-oF主机配置

具有非对称命名空间访问(AANA)的SUSE Linux Enterprise Server 15 SP5支持基于网络结构的NVMe (NVMe-oF)、包括基于光纤通道的NVMe (NVMe/FC)和其他传输。在NVMe-oF环境中、ANA相当于iSCSI和FCP环境中的ALUA多路径功能、并可通过内核NVMe多路径实施。

以下支持适用于采用ONTAP的SUSE Linux Enterprise Server 15 SP5的NVMe-oF主机配置：

- NVMe 和 SCSI 流量均可在同一主机上运行。因此、对于SCSI LUN、您可以为SCSI mpath设备配置dm-path、而可以使用NVMe多路径在主机上配置NVMe-oF命名空间设备。
- 除了NVMe/FC之外、还支持基于TCP的NVMe (NVMe/TCP)。本机中的NetApp插件 `nvme-cli` 软件包可显示NVMe/FC和NVMe/TCP命名库的ONTAP详细信息。

有关支持的配置的其他详细信息、请参见 "NetApp 互操作性表工具"。

功能

- 支持NVMe安全带内身份验证
- 支持使用唯一发现NQN的永久性发现控制器(PDC)

已知限制

- 目前不支持使用NVMe-oF协议启动SAN。
- 不 `sanlun`支持NVMe-oF。因此、在SUSE Linux Enterprise Server 15 SP5主机上、主机实用程序不支持NVMe-oF。您可以使用本机NVMe-CLI软件包中提供的NetApp插件进行所有NVMe-oF传输。

配置 NVMe/FC

您可以为Broadcom/Emulex FC或Marvell/Qlogic FC适配器配置NVMe/FC。

Broadcom/Emulex

步骤

1. 确认您使用的是建议的适配器型号：

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

示例输出：

```
LPe32002 M2
LPe32002-M2
```

2. 验证适配器型号问题描述：

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

示例输出：

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

3. 验证是否正在使用建议的Emulex主机总线适配器(HBA)固件版本：

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

示例输出：

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

4. 验证是否正在使用建议的lpfc驱动程序版本：

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

示例输出：

```
0:14.2.0.13
```

5. 验证是否可以查看启动程序端口：

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

示例输出：

```
0x100000109b579d5e
0x100000109b579d5f
```

6. 验证启动程序端口是否联机：

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

示例输出：

```
Online
Online
```

7. 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
```

示例输出：

在以下示例中、一个启动程序端口已启用、并与两个目标生命周期关联。

```
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b579d5e WWNN x200000109b579d5e DID
x011c00 ONLINE
NVME RPORT WWPN x208400a098dfdd91 WWNN x208100a098dfdd91 DID x011503
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208500a098dfdd91 WWNN x208100a098dfdd91 DID x010003
TARGET DISCSRVC *ONLINE

NVME Statistics
LS: Xmt 0000000e49 Cmpl 0000000e49 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003ceb594f Issue 000000003ce65dbe OutIO
fffffffffffb046f
abort 00000bd2 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 000014f4 Err 00012abd

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b579d5f WWNN x200000109b579d5f DID
x011b00 ONLINE
NVME RPORT WWPN x208300a098dfdd91 WWNN x208100a098dfdd91 DID x010c03
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208200a098dfdd91 WWNN x208100a098dfdd91 DID x012a03
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000e50 Cmpl 0000000e50 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003c9859ca Issue 000000003c93515e OutIO
fffffffffffaf794
abort 00000b73 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000159d Err 000135c3
```

8. 重新启动主机。

Marvell/QLogic

SUSE Linux Enterprise Server 15 SP5内核中附带的本机内置qla2xxx驱动程序具有最新的修复程序。这些修复程序对于ONTAP支持至关重要。

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
```

示例输出：

```
QLE2742 FW:v9.12.01 DVR: v10.02.08.300-k
QLE2742 FW:v9.12.01 DVR: v10.02.08.300-k
```

2. 验证是否已 `ql2xnvmeenable` 参数设置为1：

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

启用 1 MB I/O 大小（可选）

ONTAP会在"识别 控制器"数据中报告MDTS (MAX Data传输大小)为8。这意味着最大I/O请求大小最多可以为1 MB。要向Broadcom NVMe/FC主机发出大小为1 MB的I/O请求、应将参数的值 `lpfc_sg_seg_cnt`从默认值64增加 `lpfc`到256。

这些步骤不适用于逻辑NVMe/FC主机。

步骤

1. 将 `lpfc_sg_seg_cnt`参数设置为256：

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

```
options lpfc lpfc_sg_seg_cnt=256
```

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

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

启用NVMe服务

中包含两个NVMe/FC启动服务 `nvme-cli` 但是、_ONLY _ `nvmefc-boot-connections.service` 已启用、可在系统启动期间启动； `nvmf-autoconnect.service` 未启用。因此、您需要手动启用 `nvmf-autoconnect.service` 在系统引导期间启动。

步骤

1. -enable `nvmf-autoconnect.service`：

```
# systemctl enable nvmf-autoconnect.service
Created symlink /etc/systemd/system/default.target.wants/nvmf-
autoconnect.service → /usr/lib/systemd/system/nvmf-autoconnect.service.
```

2. 重新启动主机。
3. 验证和 `nvmefc-boot-connections.service`是否 `nvmf-autoconnect.service`在系统启动后运行：

   示例输出：

```
# systemctl status nvmf-autoconnect.service
nvmf-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
Loaded: loaded (/usr/lib/systemd/system/nvmf-autoconnect.service;
enabled; vendor preset: disabled)
Active: inactive (dead) since Thu 2023-05-25 14:55:00 IST; 11min
ago
Process: 2108 ExecStartPre=/sbin/modprobe nvme-fabrics (code=exited,
status=0/SUCCESS)
Process: 2114 ExecStart=/usr/sbin/nvme connect-all (code=exited,
status=0/SUCCESS)
Main PID: 2114 (code=exited, status=0/SUCCESS)

systemd[1]: Starting Connect NVMe-oF subsystems automatically during
boot...
nvme[2114]: traddr=nn-0x201700a098fd4ca6:pn-0x201800a098fd4ca6 is
already connected
systemd[1]: nvmf-autoconnect.service: Deactivated successfully.
systemd[1]: Finished Connect NVMe-oF subsystems automatically during
boot.

# systemctl status nvmefc-boot-connections.service
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; vendor preset: enabled)
Active: inactive (dead) since Thu 2023-05-25 14:55:00 IST; 11min ago
Main PID: 1647 (code=exited, status=0/SUCCESS)

systemd[1]: Starting Auto-connect to subsystems on FC-NVME devices found
during boot...
systemd[1]: nvmefc-boot-connections.service: Succeeded.
systemd[1]: Finished Auto-connect to subsystems on FC-NVME devices found
during boot.
```

配置 NVMe/TCP

您可以使用以下操作步骤配置NVMe/TCP。

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>
```

示例输出：

```
# nvme discover -t tcp -w 192.168.1.4 -a 192.168.1.31

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: 0
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.2.117
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.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.1.117
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.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.2.116
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: 3
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.1.116
eflags: explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 4====== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 0
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIEN
```

```
T116
traddr: 192.168.2.117 eflags: not specified 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.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIEN
T116
traddr: 192.168.1.117 eflags: not specified 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.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIEN
T116
traddr: 192.168.2.116 eflags: not specified sectype: none
=====Discovery Log Entry 7====== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 3
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIEN
T116
traddr: 192.168.1.116 eflags: not specified sectype: none
```

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>
```

示例输出：

```
# nvme discover -t tcp -w 192.168.1.4 -a 192.168.1.32
# nvme discover -t tcp -w 192.168.2.5 -a 192.168.2.36
# nvme discover -t tcp -w 192.168.2.5 -a 192.168.2.37
```

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>
```

示例输出：

```
# nvme connect-all -t tcp -w 192.168.1.4 -a 192.168.1.31 -l -1
# nvme connect-all -t tcp -w 192.168.1.4 -a 192.168.1.32 -l -1
# nvme connect-all -t tcp -w 192.168.2.5 -a 192.168.1.36 -l -1
# nvme connect-all -t tcp -w 192.168.2.5 -a 192.168.1.37 -l -1
```

NetApp建议设置 `ctrl-loss-tmo` 选项 `-1` 这样、如果路径丢失、NVMe/TCP启动程序就会无限期地尝试重新连接。

验证 NVMe-oF

您可以使用以下操作步骤验证NVMe-oF。

步骤

1. 验证是否已启用内核 NVMe 多路径：

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

2. 验证主机是否具有适用于ONTAP NVMe命名卷的正确控制器型号：

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

示例输出：

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

3. 验证相应ONTAP NVMe I/O控制器的NVMe I/O策略：

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

示例输出：

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

4. 验证ONTAP名称卷是否对主机可见：

```
nvme list -v
```

示例输出：

```
Subsystem        Subsystem-NQN
Controllers
----------------
------------------------------------------------------------------------
------------ ----------------------
nvme-subsys0     nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_dhcha
p   nvme0, nvme1, nvme2, nvme3


Device   SN                   MN
FR       TxPort Asdress        Subsystem    Namespaces
-------- -------------------- ----------------------------------------
-------- ----------------------------------------------
nvme0    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.2.214,trsvcid=4420,host_traddr=192.168.2.14 nvme-subsys0
nvme0n1
nvme1    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.2.215,trsvcid=4420,host_traddr=192.168.2.14 nvme-subsys0
nvme0n1
nvme2    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.1.214,trsvcid=4420,host_traddr=192.168.1.14 nvme-subsys0
nvme0n1
nvme3    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.1.215,trsvcid=4420,host_traddr=192.168.1.14 nvme-subsys0
nvme0n1


Device       Generic      NSID       Usage                  Format
Controllers
------------ ------------ ----------
----------------------------------------------------------------
/dev/nvme0n1 /dev/ng0n1   0x1     1.07  GB /   1.07  GB    4 KiB +  0 B
nvme0, nvme1, nvme2, nvme3
```

5. 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/<subsystem_name>
```

NVMe/FC

示例输出

```
# nvme list-subsys /dev/nvme1n1
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.04ba0732530911ea8e8300a098dfdd91:subsystem.nvme_145
_1
\
+- nvme2 fc traddr=nn-0x208100a098dfdd91:pn-
0x208200a098dfdd91,host_traddr=nn-0x200000109b579d5f:pn-
0x100000109b579d5f live optimized
+- nvme3 fc traddr=nn-0x208100a098dfdd91:pn-
0x208500a098dfdd91,host_traddr=nn-0x200000109b579d5e:pn-
0x100000109b579d5e live optimized
+- nvme4 fc traddr=nn-0x208100a098dfdd91:pn-
0x208400a098dfdd91,host_traddr=nn-0x200000109b579d5e:pn-
0x100000109b579d5e live non-optimized
+- nvme6 fc traddr=nn-0x208100a098dfdd91:pn-
0x208300a098dfdd91,host_traddr=nn-0x200000109b579d5f:pn-
0x100000109b579d5f live non-optimized
```

NVMe/TCP

示例输出

```
# nvme list-subsys
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
hostnqn=nqn.2014-08.org.nvmexpress:uuid:e58eca24-faff-11ea-8fee-
3a68dd3b5c5f
iopolicy=round-robin

 +- nvme0 tcp
traddr=192.168.2.214,trsvcid=4420,host_traddr=192.168.2.14 live
 +- nvme1 tcp
traddr=192.168.2.215,trsvcid=4420,host_traddr=192.168.2.14 live
 +- nvme2 tcp
traddr=192.168.1.214,trsvcid=4420,host_traddr=192.168.1.14 live
 +- nvme3 tcp
traddr=192.168.1.215,trsvcid=4420,host_traddr=192.168.1.14 live
```

6. 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
```

示例输出：

```
Device            Vserver                   Namespace Path
NSID UUID                                   Size
---------------- ------------------------
------------------------------------------------------------------------
---------------------------
/dev/nvme0n1     vs_CLIENT114
/vol/CLIENT114_vol_0_10/CLIENT114_ns10       1    c6586535-da8a-
40fa-8c20-759ea0d69d33   1.07GB
```

JSON

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

示例输出：

```
{
"ONTAPdevices":[
{
"Device":"/dev/nvme0n1",
"Vserver":"vs_CLIENT114",
"Namespace_Path":"/vol/CLIENT114_vol_0_10/CLIENT114_ns10",
"NSID":1,
"UUID":"c6586535-da8a-40fa-8c20-759ea0d69d33",
"Size":"1.07GB",
"LBA_Data_Size":4096,
"Namespace_Size":262144
}
]
}
```

创建永久性发现控制器

从SuSE ONTAP 9 11.1.开始，您可以为SuSE Linux Enterprise Server 15 SP5主机创建永久性发现控制器(PERIOPPO持 性发现控制器，PDC)。要自动检测NVMe子系统添加或删除方案以及对发现日志页面数据的更改、需要PDC。

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 16, Generation counter 14
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  0
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:discovery
traddr:  192.168.1.214
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:  0
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:discovery
traddr:  192.168.1.215
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:  0
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:discovery
traddr:  192.168.2.215
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:  0
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:discovery
traddr:  192.168.2.214
eflags:  explicit discovery connections, duplicate discovery
information sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr:  192.168.1.214
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr:  192.168.1.215
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr:  192.168.2.215
eflags:  none
sectype: none
```

```
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr:  192.168.2.214
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr:  192.168.1.214
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr:  192.168.1.215
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
```

```
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr:  192.168.2.215
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr:  192.168.2.214
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr:  192.168.1.214
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr:  192.168.1.215
eflags:  none
sectype: none
=====Discovery Log Entry 14======
```

```
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr:  192.168.2.215
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr:  192.168.2.214
eflags:  none
sectype: none
```

2. 为发现子系统创建PDC：

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

示例输出：

```
nvme discover -t tcp -w 192.168.1.16 -a 192.168.1.116 -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
```

示例输出：

```
vserver nvme show-discovery-controller -instance -vserver vs_nvme175
Vserver Name: vs_CLIENT116 Controller ID: 00C0h
Discovery Subsystem NQN: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery Logical
Interface UUID: d23cbb0a-c0a6-11ec-9731-d039ea165abc Logical Interface:
CLIENT116_lif_4a_1
Node: A400-14-124
Host NQN: nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-be09-
74362c0c1afc
Transport Protocol: nvme-tcp
Initiator Transport Address: 192.168.1.16
Host Identifier: 59de25be738348f08a79df4bce9573f3 Admin Queue Depth: 32
Header Digest Enabled: false Data Digest Enabled: false
Vserver UUID: 48391d66-c0a6-11ec-aaa5-d039ea165514
```

设置安全带内身份验证

从Linux.12.1开始、支持在ONTAP 9 15 SP5主机和ONTAP控制器之间通过NVMe/TCP和NVMe/FC进行安全带内身份验证。

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

您可以使用命令行界面或Config 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

步骤

1. 获取主机NQN：

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

2. 为SUSE Linux Enterprise Server 15 SP5主机生成dhchap密钥：

```
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:d3ca725a- ac8d-4d88-b46a-174ac235139b
DHHC-
1:03:J2UJQfj9f0pLnpF/ASDJRTyILKJRr5CougGpGdQSysPrLu6RW1fGl5VSjbeDF1n
1DEh3nVBe19nQ/LxreSBeH/bx/pU=:
```

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
   ```

   单向配置的示例输出：

   ```
   # cat /sys/class/nvme-subsystem/nvme-subsys1/nvme*/dhchap_secret
   DHHC-
   1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
   wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
   DHHC-
   1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
   wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
   DHHC-
   1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
   wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
   DHHC-
   1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
   wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
   ```

   b. 验证控制器dhchap密钥：

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

   双向配置的输出示例：

```
# cat /sys/class/nvme-subsystem/nvme-
subsys6/nvme*/dhchap_ctrl_secret
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
```

JSON 文件

您可以使用 `/etc/nvme/config.json` 文件 `nvme connect-all` 命令ONTAP。

您可以使用生成JSON文件 `-o` 选项有关更多语法选项、请参见NVMe Connect-all手册页。

步骤

1. 配置 JSON 文件：

```
# cat /etc/nvme/config.json
[
 {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-
be09-74362c0c1afc",
    "hostid":"3ae10b42-21af-48ce-a40b-cfb5bad81839",
    "dhchap_key":"DHHC-
1:03:Cu3ZZfIz1WMlqZFnCMqpAgn/T6EVOcIFHez215U+Pow8jTgBF2UbNk3DK4wfk2E
ptWpna1rpwG5CndpOgxpRxh9m41w=:"
 },

 {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-
be09-74362c0c1afc",
    "subsystems":[
        {
            "nqn":"nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_C
LIENT116",
            "ports":[
               {
```

```
                    "transport":"tcp",
                    "traddr":"192.168.1.117",
                    "host_traddr":"192.168.1.16",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
                },
                {
                    "transport":"tcp",
                    "traddr":"192.168.1.116",
                    "host_traddr":"192.168.1.16",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
                },
                {
                    "transport":"tcp",
                    "traddr":"192.168.2.117",
                    "host_traddr":"192.168.2.16",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
                },
                {
                    "transport":"tcp",
                    "traddr":"192.168.2.116",
                    "host_traddr":"192.168.2.16",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
                }
            ]
        }
    ]
 }
]

[NOTE]
In the preceding example, `dhchap_key` corresponds to
`dhchap_secret` and `dhchap_ctrl_key` corresponds to
`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
```

示例输出：

```
traddr=192.168.2.116 is already connected
traddr=192.168.1.116 is already connected
traddr=192.168.2.117 is already connected
traddr=192.168.1.117 is already connected
traddr=192.168.2.117 is already connected
traddr=192.168.1.117 is already connected
traddr=192.168.2.116 is already connected
traddr=192.168.1.116 is already connected
traddr=192.168.2.116 is already connected
traddr=192.168.1.116 is already connected
traddr=192.168.2.117 is already connected
traddr=192.168.1.117 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
   ```

   示例输出：

   ```
   DHHC-1:01:NunEWY7AZlXqxITGheByarwZdQvU4ebZg9HOjIr6nOHEkxJg:
   ```

   b. 验证控制器dhchap密钥：

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

   示例输出：

```
DHHC-
1:03:2YJinsxa2v3+m8qqCiTnmgBZoH6mIT6G/6f0aGO8viVZB4VLNLH4z8CvK7pV
YxN6S5fOAtaU3DNi12rieRMfdbg3704=:
```

已知问题

具有ONTAP版本的SUSE Linux Enterprise Server 15 SP5没有已知问题。

适用于采用ONTAP的SUSE Linux Enterprise Server 15 SP4的NVMe-oF主机配置

具有非对称命名空间访问(AANA)的SUSE Linux Enterprise Server (SLES) 15 SP4支持基于网络结构的NVMe (NVMe-oF)、包括基于光纤通道的NVMe (NVMe/FC)和其他传输。在NVMe-oF环境中、ANA相当于iSCSI和FCP环境中的AUA多路径功能、并通过内核NVMe多路径实施。

以下支持适用于采用ONTAP的SUSE Linux Enterprise Server 15 SP4的NVMe-oF主机配置：

- NVMe 和 SCSI 流量均可在同一主机上运行。因此、对于SCSI LUN、您可以为SCSI mpath设备配置dm-path、而可以使用NVMe多路径在主机上配置NVMe-oF命名空间设备。
- 除了NVMe/FC之外、还支持基于TCP的NVMe (NVMe/TCP)。本机NVMe-CLI软件包中的NetApp插件可显示NVMe/FC和NVMe/TCP命名库的ONTAP详细信息。

有关支持的配置的其他详细信息、请参见 "NetApp 互操作性表工具"。

功能

- 支持NVMe安全带内身份验证
- 支持使用唯一发现NQN的永久性发现控制器(PDC)

已知限制

- 目前不支持使用NVMe-oF协议启动SAN。
- NVMe-oF不支持sanlun。因此、在SUSE Linux Enterprise Server 15 SP5主机上、主机实用程序不支持NVMe-oF。您可以依靠本机NVMe-CLI软件包中提供的NetApp插件来进行所有NVMe-oF传输。

配置 NVMe/FC

您可以为Broadcom/Emulex FC适配器或Marvell/Qlogic FC适配器配置NVMe/FC。

Broadcom/Emulex

步骤

1. 确认您使用的是建议的适配器型号：

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

示例输出：

```
LPe32002 M2
LPe32002-M2
```

2. 验证适配器型号问题描述：

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

示例输出：

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

3. 验证是否正在使用建议的Emulex主机总线适配器(HBA)固件版本：

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

示例输出：

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

4. 验证是否正在使用建议的lpfc驱动程序版本：

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

示例输出：

```
0:14.2.0.6
```

5. 验证是否可以查看启动程序端口：

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

示例输出：

```
0x100000109b579d5e
0x100000109b579d5f
```

6. 验证启动程序端口是否联机：

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

示例输出：

```
Online
Online
```

7. 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
```

示例输出：

在以下示例中、一个启动程序端口已启用、并与两个目标生命周期关联。

```
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b579d5e WWNN x200000109b579d5e DID
x011c00 ONLINE
NVME RPORT WWPN x208400a098dfdd91 WWNN x208100a098dfdd91 DID x011503
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208500a098dfdd91 WWNN x208100a098dfdd91 DID x010003
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000e49 Cmpl 0000000e49 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003ceb594f Issue 000000003ce65dbe OutIO
fffffffffffb046f
abort 00000bd2 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 000014f4 Err 00012abd

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b579d5f WWNN x200000109b579d5f DID
x011b00 ONLINE
NVME RPORT WWPN x208300a098dfdd91 WWNN x208100a098dfdd91 DID x010c03
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208200a098dfdd91 WWNN x208100a098dfdd91 DID x012a03
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000e50 Cmpl 0000000e50 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003c9859ca Issue 000000003c93515e OutIO
fffffffffffaf794
abort 00000b73 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000159d Err 000135c3
```

8. 重新启动主机。

Marvell/QLogic

SUSE Linux Enterprise Server 15 SP4内核中附带的本机内置qla2xxx驱动程序包含最新的修复程序。这些修复程序对于ONTAP支持至关重要。

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
```

示例输出：

```
QLE2742 FW:v9.08.02 DVR:v10.02.07.800-k QLE2742 FW:v9.08.02
DVR:v10.02.07.800-k
```

2. 验证是否已 `ql2xnvmeenable` 参数设置为1：

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

启用 1 MB I/O 大小（可选）

ONTAP会在"识别 控制器"数据中报告MDTS (MAX Data传输大小)为8。这意味着最大I/O请求大小最多可以为1 MB。要向Broadcom NVMe/FC主机发出大小为1 MB的I/O请求、应将参数的值 `lpfc_sg_seg_cnt`从默认值64增加 `lpfc`到256。

这些步骤不适用于逻辑NVMe/FC主机。

步骤

1. 将 `lpfc_sg_seg_cnt`参数设置为256：

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

```
options lpfc lpfc_sg_seg_cnt=256
```

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

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

启用NVMe服务

中包含两个NVMe/FC启动服务 `nvme-cli` 但是、_ONLY _ `nvmefc-boot-connections.service` 已启用、可在系统启动期间启动； `nvmf-autoconnect.service` 未启用。因此、您需要手动启用 `nvmf-autoconnect.service` 在系统引导期间启动。

步骤

1. -enable `nvmf-autoconnect.service`：

```
# systemctl enable nvmf-autoconnect.service
Created symlink /etc/systemd/system/default.target.wants/nvmf-
autoconnect.service → /usr/lib/systemd/system/nvmf-autoconnect.service.
```

2. 重新启动主机。
3. 验证和 `nvmefc-boot-connections.service`是否 `nvmf-autoconnect.service`在系统启动后运行：

   示例输出：

```
# systemctl status nvmf-autoconnect.service
   nvmf-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
     Loaded: loaded (/usr/lib/systemd/system/nvmf-autoconnect.service;
enabled; vendor preset: disabled)
     Active: inactive (dead) since Thu 2023-05-25 14:55:00 IST; 11min
ago
    Process: 2108 ExecStartPre=/sbin/modprobe nvme-fabrics (code=exited,
status=0/SUCCESS)
    Process: 2114 ExecStart=/usr/sbin/nvme connect-all (code=exited,
status=0/SUCCESS)
   Main PID: 2114 (code=exited, status=0/SUCCESS)

   systemd[1]: Starting Connect NVMe-oF subsystems automatically during
boot...
   nvme[2114]: traddr=nn-0x201700a098fd4ca6:pn-0x201800a098fd4ca6 is
already connected
   systemd[1]: nvmf-autoconnect.service: Deactivated successfully.
   systemd[1]: Finished Connect NVMe-oF subsystems automatically during
boot.

# systemctl status nvmefc-boot-connections.service
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; vendor preset: enabled)
   Active: inactive (dead) since Thu 2023-05-25 14:55:00 IST; 11min ago
 Main PID: 1647 (code=exited, status=0/SUCCESS)

systemd[1]: Starting Auto-connect to subsystems on FC-NVME devices found
during boot...
systemd[1]: nvmefc-boot-connections.service: Succeeded.
systemd[1]: Finished Auto-connect to subsystems on FC-NVME devices found
during boot.
```

配置 NVMe/TCP

您可以使用以下操作步骤配置NVMe/TCP。

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>
```

示例输出：

```
# nvme discover -t tcp -w 192.168.1.4 -a 192.168.1.31

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: 0
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.2.117
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.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.1.117
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.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.2.116
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: 3
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.1.116
eflags: explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 4====== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 0
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIEN
```

```
T116
traddr: 192.168.2.117 eflags: not specified 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.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIEN
T116
traddr: 192.168.1.117 eflags: not specified 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.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIEN
T116
traddr: 192.168.2.116 eflags: not specified sectype: none
=====Discovery Log Entry 7====== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 3
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIEN
T116
traddr: 192.168.1.116 eflags: not specified sectype: none
```

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>
```

示例输出：

```
# nvme discover -t tcp -w 192.168.1.4 -a 192.168.1.32
# nvme discover -t tcp -w 192.168.2.5 -a 192.168.2.36
# nvme discover -t tcp -w 192.168.2.5 -a 192.168.2.37
```

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>
```

示例输出：

```
# nvme connect-all -t tcp -w 192.168.1.4 -a 192.168.1.31 -l -1
# nvme connect-all -t tcp -w 192.168.1.4 -a 192.168.1.32 -l -1
# nvme connect-all -t tcp -w 192.168.2.5 -a 192.168.1.36 -l -1
# nvme connect-all -t tcp -w 192.168.2.5 -a 192.168.1.37 -l -1
```

NetApp建议设置 `ctrl-loss-tmo` 选项 `-1` 这样、如果路径丢失、NVMe/TCP启动程序就会无限期地尝试重新连接。

验证 NVMe-oF

您可以使用以下操作步骤验证NVMe-oF。

步骤

1. 验证是否已启用内核 NVMe 多路径：

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

2. 验证主机是否具有适用于ONTAP NVMe命名卷的正确控制器型号：

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

示例输出：

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

3. 验证相应ONTAP NVMe I/O控制器的NVMe I/O策略：

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

示例输出：

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

4. 验证ONTAP名称卷是否对主机可见：

```
nvme list -v
```

示例输出：

```
Subsystem        Subsystem-NQN
Controllers
----------------
------------------------------------------------------------------------
------------ ----------------------
nvme-subsys0     nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_dhcha
p    nvme0, nvme1, nvme2, nvme3


Device   SN                   MN
FR       TxPort Asdress        Subsystem    Namespaces
-------- -------------------- ----------------------------------------
-------- ----------------------------------------------
nvme0    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.2.214,trsvcid=4420,host_traddr=192.168.2.14 nvme-subsys0
nvme0n1
nvme1    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.2.215,trsvcid=4420,host_traddr=192.168.2.14 nvme-subsys0
nvme0n1
nvme2    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.1.214,trsvcid=4420,host_traddr=192.168.1.14 nvme-subsys0
nvme0n1
nvme3    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.1.215,trsvcid=4420,host_traddr=192.168.1.14 nvme-subsys0
nvme0n1


Device       Generic      NSID       Usage                  Format
Controllers
------------ ------------ ----------
----------------------------------------------------------------
/dev/nvme0n1 /dev/ng0n1   0x1     1.07  GB /   1.07  GB    4 KiB +  0 B
nvme0, nvme1, nvme2, nvme3
```

5. 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/<subsystem_name>
```

NVMe/FC

```
# nvme list-subsys /dev/nvme1n1
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.04ba0732530911ea8e8300a098dfdd91:subsystem.nvme_145
_1
\
+- nvme2 fc traddr=nn-0x208100a098dfdd91:pn-
0x208200a098dfdd91,host_traddr=nn-0x200000109b579d5f:pn-
0x100000109b579d5f live optimized
+- nvme3 fc traddr=nn-0x208100a098dfdd91:pn-
0x208500a098dfdd91,host_traddr=nn-0x200000109b579d5e:pn-
0x100000109b579d5e live optimized
+- nvme4 fc traddr=nn-0x208100a098dfdd91:pn-
0x208400a098dfdd91,host_traddr=nn-0x200000109b579d5e:pn-
0x100000109b579d5e live non-optimized
+- nvme6 fc traddr=nn-0x208100a098dfdd91:pn-
0x208300a098dfdd91,host_traddr=nn-0x200000109b579d5f:pn-
0x100000109b579d5f live non-optimized
```

NVMe/TCP

```
# nvme list-subsys
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
hostnqn=nqn.2014-08.org.nvmexpress:uuid:e58eca24-faff-11ea-8fee-
3a68dd3b5c5f
iopolicy=round-robin

 +- nvme0 tcp
traddr=192.168.2.214,trsvcid=4420,host_traddr=192.168.2.14 live
 +- nvme1 tcp
traddr=192.168.2.215,trsvcid=4420,host_traddr=192.168.2.14 live
 +- nvme2 tcp
traddr=192.168.1.214,trsvcid=4420,host_traddr=192.168.1.14 live
 +- nvme3 tcp
traddr=192.168.1.215,trsvcid=4420,host_traddr=192.168.1.14 live
```

6. 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
```

示例输出：

```
Device            Vserver                   Namespace Path
NSID UUID                                   Size
----------------- ------------------------
------------------------------------------------------------------------
---------------------------
/dev/nvme0n1      vs_CLIENT114
/vol/CLIENT114_vol_0_10/CLIENT114_ns10       1    c6586535-da8a-
40fa-8c20-759ea0d69d33   1.07GB
```

JSON

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

示例输出：

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme0n1",
      "Vserver":"vs_CLIENT114",
      "Namespace_Path":"/vol/CLIENT114_vol_0_10/CLIENT114_ns10",
      "NSID":1,
      "UUID":"c6586535-da8a-40fa-8c20-759ea0d69d33",
      "Size":"1.07GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":262144
    }
  ]
}
```

创建永久性发现控制器

从SuSE ONTAP 9 11.1.开始，您可以为SuSE Linux Enterprise Server 15 SP4主机创建永久性发现控制器(PERIOC)(SP4)。要自动检测NVMe子系统添加或删除方案以及对发现日志页面数据的更改、需要PDC。

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 16, Generation counter 14
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  0
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:discovery
traddr:  192.168.1.214
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:  0
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:discovery
traddr:  192.168.1.215
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:  0
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:discovery
traddr:  192.168.2.215
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:  0
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:discovery
traddr:  192.168.2.214
eflags:  explicit discovery connections, duplicate discovery
information sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr:  192.168.1.214
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr:  192.168.1.215
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr:  192.168.2.215
eflags:  none
sectype: none
```

```
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr:  192.168.2.214
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr:  192.168.1.214
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr:  192.168.1.215
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
```

```
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr:  192.168.2.215
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr:  192.168.2.214
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr:  192.168.1.214
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr:  192.168.1.215
eflags:  none
sectype: none
=====Discovery Log Entry 14======
```

```
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr:  192.168.2.215
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr:  192.168.2.214
eflags:  none
sectype: none
```

2. 为发现子系统创建PDC：

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

示例输出：

```
nvme discover -t tcp -w 192.168.1.16 -a 192.168.1.116 -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
```

示例输出：

```
vserver nvme show-discovery-controller -instance -vserver vs_nvme175
Vserver Name: vs_CLIENT116 Controller ID: 00C0h
Discovery Subsystem NQN: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery Logical
Interface UUID: d23cbb0a-c0a6-11ec-9731-d039ea165abc Logical Interface:
CLIENT116_lif_4a_1
Node: A400-14-124
Host NQN: nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-be09-
74362c0c1afc
Transport Protocol: nvme-tcp
Initiator Transport Address: 192.168.1.16
Host Identifier: 59de25be738348f08a79df4bce9573f3 Admin Queue Depth: 32
Header Digest Enabled: false Data Digest Enabled: false
Vserver UUID: 48391d66-c0a6-11ec-aaa5-d039ea165514
```

设置安全带内身份验证

从SuSE ONTAP 9 1.12.1开始、可通过NVMe/TCP和NVMe/FC在SuSE Linux Enterprise Server 15 SP4主机和ONTAP控制器之间进行安全的带内身份验证。

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

您可以使用命令行界面或Config 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

步骤

1. 获取主机NQN：

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

2. 为SUSE Linux Enterprise Server 15 SP4主机生成dhchap密钥：

```
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:d3ca725a-
ac8d-4d88-b46a-174ac235139b
DHHC-
1:03:J2UJQfj9f0pLnpF/ASDJRTyILKJRr5CougGpGdQSysPrLu6RW1fGl5VSjbeDF1n1DE
h3nVBe19nQ/LxreSBeH/bx/pU=:
```

1. 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}
```

2. 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>
```

3. 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
   ```

   单向配置的示例输出：

   ```
   SR650-14-114:~ # cat /sys/class/nvme-subsystem/nvme-
   subsys1/nvme*/dhchap_secret
   DHHC-
   1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
   wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
   DHHC-
   1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
   wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
   DHHC-
   1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
   wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
   DHHC-
   1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
   wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
   ```

   b. 验证控制器dhchap密钥：

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

   双向配置的输出示例：

```
SR650-14-114:~ # cat /sys/class/nvme-subsystem/nvme-
subsys6/nvme*/dhchap_ctrl_secret
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
```

JSON 文件

您可以使用 `/etc/nvme/config.json` 文件 `nvme connect-all` 命令ONTAP。

您可以使用生成JSON文件 `-o` 选项有关更多语法选项、请参见NVMe Connect-all手册页。

步骤

1. 配置 JSON 文件：

```
# cat /etc/nvme/config.json
[
 {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-
be09-74362c0c1afc",
    "hostid":"3ae10b42-21af-48ce-a40b-cfb5bad81839",
    "dhchap_key":"DHHC-
1:03:Cu3ZZfIz1WMlqZFnCMqpAgn/T6EVOcIFHez215U+Pow8jTgBF2UbNk3DK4wfk2E
ptWpna1rpwG5CndpOgxpRxh9m41w=:"
 },

 {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-
be09-74362c0c1afc",
    "subsystems":[
        {
            "nqn":"nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_C
LIENT116",
            "ports":[
               {
```

```
                    "transport":"tcp",
                    "traddr":"192.168.1.117",
                    "host_traddr":"192.168.1.16",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
               },
               {
                    "transport":"tcp",
                    "traddr":"192.168.1.116",
                    "host_traddr":"192.168.1.16",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
               },
               {
                    "transport":"tcp",
                    "traddr":"192.168.2.117",
                    "host_traddr":"192.168.2.16",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
               },
               {
                    "transport":"tcp",
                    "traddr":"192.168.2.116",
                    "host_traddr":"192.168.2.16",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
               }
          ]
       }
   ]
 }
]

[NOTE]
In the preceding example, `dhchap_key` corresponds to
`dhchap_secret` and `dhchap_ctrl_key` corresponds to
`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
```

示例输出：

```
traddr=192.168.2.116 is already connected
traddr=192.168.1.116 is already connected
traddr=192.168.2.117 is already connected
traddr=192.168.1.117 is already connected
traddr=192.168.2.117 is already connected
traddr=192.168.1.117 is already connected
traddr=192.168.2.116 is already connected
traddr=192.168.1.116 is already connected
traddr=192.168.2.116 is already connected
traddr=192.168.1.116 is already connected
traddr=192.168.2.117 is already connected
traddr=192.168.1.117 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
   ```

   示例输出：

   ```
   DHHC-1:01:NunEWY7AZlXqxITGheByarwZdQvU4ebZg9HOjIr6nOHEkxJg:
   ```

   b. 验证控制器dhchap密钥：

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

   示例输出：

```
DHHC-
1:03:2YJinsxa2v3+m8qqCiTnmgBZoH6mIT6G/6f0aGO8viVZB4VLNLH4z8CvK7pVYxN
6S5fOAtaU3DNi12rieRMfdbg3704=:
```

已知问题

具有ONTAP版本的SUSE Linux Enterprise Server 15 SP4没有已知问题。

适用于采用 ONTAP 的 SUSE Linux Enterprise Server 15 SP3 的 NVMe-oF 主机配置

带有ANA (非对称命名空间访问)的SUSE Linux Enterprise Server 15 SP3支持基于网络结构的NVMe或NVMe-oF (包括NVMe/FC和其他传输)。ANA是NVMe-oF环境中的AUA等效产品、目前已通过内核NVMe多路径实施。使用此过程、您可以使用基于SUSE Linux Enterprise Server 15 SP3的ANA和ONTAP作为目标、启用具有内核NVMe多路径的NVMe-oF。

请参见 "NetApp 互操作性表" 有关支持的配置的准确详细信息。

功能

- SUSE Linux Enterprise Server 15 SP3支持NVMe/FC和其他传输。
- NVMe-oF 不支持 sanlun 。因此、在SUSE Linux Enterprise Server 15 SP3上不支持NVMe-oF。您可以依赖适用于NVMe-oF的本机NVMe-CLI软件包中提供的NetApp插件。这应支持所有NVMe-oF传输。
- NVMe 和 SCSI 流量均可在同一主机上运行。实际上，这应该是客户通常部署的主机配置。因此，对于 SCSI ，您可以像往常一样为 SCSI LUN 配置 `dm-multipath` ，从而生成 mpath 设备，而 NVMe 多路径则可用于在主机上配置 NVMe-of 多路径设备。

已知限制

目前不支持使用NVMe-oF协议启动SAN。

启用内核 **NVMe** 多路径

默认情况下、SUSE Linux Enterprise Server主机(例如SUSE Linux Enterprise Server 15 SP3)上已启用内核NVMe多路径。因此，此处不需要其他设置。有关受支持配置的准确详细信息、请参见"NetApp 互操作性表"。

NVMe-oF 启动程序包

请参见 "NetApp 互操作性表" 有关支持的配置的准确详细信息。

1. 验证是否已在SUSE Linux Enterprise Server 15 SP3 MU主机上安装了必需的内核和NVMe-CLI MU软件包。

   示例

```
# uname -r
5.3.18-59.5-default

# rpm -qa|grep nvme-cli
nvme-cli-1.13-3.3.1.x86_64
```

上述 NVMe-CLI MU 软件包现在包括以下内容：

- * NVMe/FC 自动连接脚本 * —在还原命名空间的底层路径以及主机重新启动期间， NVMe/FC 自动（重新）连接所需的脚本：

```
# rpm -ql nvme-cli-1.13-3.3.1.x86_64
/etc/nvme
/etc/nvme/hostid
/etc/nvme/hostnqn
/usr/lib/systemd/system/nvmefc-boot-connections.service
/usr/lib/systemd/system/nvmefc-connect.target
/usr/lib/systemd/system/nvmefc-connect@.service
...
```

- * NVMe udev rule* - 用于确保适用场景多路径循环负载平衡器默认 ONTAP 所有 ONTAP 命名空间的新 udev 规则：

```
# rpm -ql nvme-cli-1.13-3.3.1.x86_64
/etc/nvme
/etc/nvme/hostid
/etc/nvme/hostnqn
/usr/lib/systemd/system/nvmefc-boot-connections.service
/usr/lib/systemd/system/nvmf-autoconnect.service
/usr/lib/systemd/system/nvmf-connect.target
/usr/lib/systemd/system/nvmf-connect@.service
/usr/lib/udev/rules.d/70-nvmf-autoconnect.rules
/usr/lib/udev/rules.d/71-nvmf-iopolicy-netapp.rules
...
# cat /usr/lib/udev/rules.d/71-nvmf-iopolicy-netapp.rules
# Enable round-robin for NetApp ONTAP and NetApp E-Series
ACTION=="add", SUBSYSTEM=="nvme-subsystem", ATTR{model}=="NetApp
ONTAP Controller", ATTR{iopolicy}="round-robin"
ACTION=="add", SUBSYSTEM=="nvme-subsystem", ATTR{model}=="NetApp E-
Series", ATTR{iopolicy}="round-robin"
```

- * 适用于 ONTAP 设备的 NetApp 插件 * —现有的 NetApp 插件现已进行修改，以处理 ONTAP 命名空间。

2. 检查主机上 ` /etc/nve/hostnqn` 处的 hostnqn 字符串，并确保其与 ONTAP 阵列上相应子系统的 hostnqn 字符串正确匹配。例如：

```
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:uuid:3ca559e1-5588-4fc4-b7d6-5ccfb0b9f054
::> vserver nvme subsystem host show -vserver vs_fcnvme_145
Vserver     Subsystem      Host NQN
-------     ---------      ----------------------------------
vs_nvme_145 nvme_145_1 nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-
41a6-a1fd-cf8262c8713f
            nvme_145_2 nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-
41a6-a1fd-cf8262c8713f
            nvme_145_3 nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-
41a6-a1fd-cf8262c8713f
            nvme_145_4 nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-
41a6-a1fd-cf8262c8713f
            nvme_145_5 nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-
41a6-a1fd-cf8262c8713f
5 entries were displayed.
```

根据主机上使用的 FC 适配器，继续执行以下步骤。

配置 NVMe/FC

Broadcom/Emulex

1. 验证您是否具有建议的适配器和固件版本。例如：

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
# cat /sys/class/scsi_host/host*/fwrev
12.8.340.8, sli-4:2:c
12.8.840.8, sli-4:2:c
```

◦ 较新的 lpfc 驱动程序（收件箱和发件箱）已将 lpfc_enable_FC4_type 默认设置为 3 ，因此，您不再需要在 ` /etc/modprobe.d/lpfc.conf` 中明确设置此设置，然后重新创建 `initrd` 。默认情况下，已启用 `lpfc NVMe` 支持：

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

- 现有的原生收件箱 lpfc 驱动程序已经是最新版本，并且与 NVMe/FC 兼容。因此，您无需安装 lpfc OOB 驱动程序。

```
# cat /sys/module/lpfc/version
0:12.8.0.10
```

2. 验证启动程序端口是否已启动且正在运行。

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b579d5e
0x100000109b579d5f
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

3. 确认NVMe/FC启动程序端口已启用、您可以看到目标端口、并且所有端口均已启动且正在运行。+在以下示例中、仅启用了一个启动程序端口并将其与两个目标生命周期关联起来：

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b579d5e WWNN x200000109b579d5e DID
x011c00 ONLINE
NVME RPORT WWPN x208400a098dfdd91 WWNN x208100a098dfdd91 DID x011503
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208500a098dfdd91 WWNN x208100a098dfdd91 DID x010003
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000e49 Cmpl 0000000e49 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003ceb594f Issue 000000003ce65dbe OutIO
fffffffffffb046f
abort 00000bd2 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 000014f4 Err 00012abd
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b579d5f WWNN x200000109b579d5f DID
x011b00 ONLINE
NVME RPORT WWPN x208300a098dfdd91 WWNN x208100a098dfdd91 DID x010c03
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208200a098dfdd91 WWNN x208100a098dfdd91 DID x012a03
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000e50 Cmpl 0000000e50 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003c9859ca Issue 000000003c93515e OutIO
fffffffffffaf794
abort 00000b73 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000159d Err 000135c3
```

4. 重新启动主机。

启用 1 MB I/O 大小（可选）

ONTAP 在 "Identify Controller" （识别控制器）数据中报告 MTS （ MAX Data 传输大小）为 8 ，这意味着最大 I/O 请求大小应最多为 1 MB 。但是，对于 Broadcom NVMe/FC 主机 1 MB 大小的问题描述 I/O 请求， lpfc 参数 `lpfc_sg_seg_cnt` 也应从默认值 64 增加到 256 。请按照以下说明执行此操作：

1. 在相应的 `modprobe lpfc.conf` 文件中附加值 256 ：

```
# cat /etc/modprobe.d/lpfc.conf
options lpfc lpfc_sg_seg_cnt=256
```

2. 运行 `dracut -f`命令并重新启动主机。
3. 重新启动后，通过检查相应的 sysfs 值来验证是否已应用上述设置：

```
# cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
256
```

现在， Broadcom NVMe/FC 主机应该能够在 ONTAP 命名空间设备上最多发送 1 MB 的 I/O 请求。

Marvell/QLogic

更新的SUSE Linux Enterprise Server 15 SP3 MU内核中包含的本机内置qla2xxx驱动程序具有最新的上游修复程序。这些修复程序对于ONTAP支持至关重要。

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，例如：

```
# cat /sys/class/fc_host/host*/symbolic_name
QLE2742 FW:v9.06.02 DVR:v10.02.00.106-k
QLE2742 FW:v9.06.02 DVR:v10.02.00.106-k
```

2. 验证是否已设置 `ql2xnvmeenable` ，以使 Marvell 适配器能够用作 NVMe/FC 启动程序：

   ` # cat /sys/module/qla2xxx/parameters/ql2xnvmeenable 1`

配置 NVMe/TCP

与 NVMe/FC 不同， NVMe/TCP 没有自动连接功能。这表明 Linux NVMe/TCP 主机存在两个主要限制：

- * 恢复路径后不会自动重新连接 * NVMe/TCP 无法自动重新连接到在路径关闭后 10 分钟内恢复的路径，此路径超出了默认值 `Ctrl-los-tm` timer 。
- * 主机启动期间无自动连接 * NVMe/TCP 也无法在主机启动期间自动连接。

您应将故障转移事件的重试期限至少设置为30分钟、以防止超时。您可以通过增加Ctrl_los_TMOs计时器的值来增加重试期限。详细信息如下：

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 192.168.1.8 -a 192.168.1.51
Discovery Log Number of Records 10, Generation counter 119
=====Discovery Log Entry 0======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.2.56
sectype: none
=====Discovery Log Entry 1======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.1.51
sectype: none
=====Discovery Log Entry 2======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_2
traddr: 192.168.2.56
sectype: none
...
```

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

```
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.52
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.56
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.57
```

3. 运行 `nvme connect-all` 命令。确保设置较长的 `ctrl_loss_tmo` 计时器重试期限(例如、30分钟、可设置为到 `-l 1800`)、以便在发生路径丢失时重试较长时间。例如：

```
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.51 -l 1800
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.52 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.56 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.57 -l 1800
```

验证 NVMe-oF

1. 通过检查以下各项验证是否确实已启用内核 NVMe 多路径：

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

2. 验证相应 ONTAP 命名空间的适当 NVMe-oF 设置（例如， `model` set to `NetApp ONTAP Controller` and `load balancing OPolicy` set to `round-robin` ）是否正确反映在主机上：

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller

# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

3. 验证 ONTAP 命名空间是否正确反映在主机上。例如：

```
# nvme list
Node           SN                    Model                     Namespace
------------   --------------------- ----------------------------------
/dev/nvme0n1   81CZ5BQuUNfGAAAAAAAB  NetApp ONTAP Controller   1

Usage                 Format         FW Rev
--------------------  -----------    --------
85.90 GB / 85.90 GB   4 KiB + 0 B    FFFFFFFF
```

另一个示例：

```
# nvme list
Node           SN                    Model                     Namespace
------------   --------------------- --------------------------------
/dev/nvme0n1   81CYrBQuTHQFAAAAAAAC  NetApp ONTAP Controller   1

Usage                 Format         FW Rev
--------------------  -----------    --------
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。例如：

```
# nvme list-subsys /dev/nvme1n1
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.04ba0732530911ea8e8300a098dfdd91:subsystem.nvme_145_1
\
+- nvme2 fc traddr=nn-0x208100a098dfdd91:pn-0x208200a098dfdd91
host_traddr=nn-0x200000109b579d5f:pn-0x100000109b579d5f live non-
optimized
+- nvme3 fc traddr=nn-0x208100a098dfdd91:pn-0x208500a098dfdd91
host_traddr=nn-0x200000109b579d5e:pn-0x100000109b579d5e live non-
optimized
+- nvme4 fc traddr=nn-0x208100a098dfdd91:pn-0x208400a098dfdd91
host_traddr=nn-0x200000109b579d5e:pn-0x100000109b579d5e live optimized
+- nvme6 fc traddr=nn-0x208100a098dfdd91:pn-0x208300a098dfdd91
host_traddr=nn-0x200000109b579d5f:pn-0x100000109b579d5f live optimized
```

另一个示例：

```
#nvme list-subsys /dev/nvme0n1
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.37ba7d9cbfba11eba35dd039ea165514:subsystem.nvme_114_tcp
_1
\
+- nvme0 tcp traddr=192.168.2.36 trsvcid=4420 host_traddr=192.168.1.4
live optimized
+- nvme1 tcp traddr=192.168.1.31 trsvcid=4420 host_traddr=192.168.1.4
live optimized
+- nvme10 tcp traddr=192.168.2.37 trsvcid=4420 host_traddr=192.168.1.4
live non-optimized
+- nvme11 tcp traddr=192.168.1.32 trsvcid=4420 host_traddr=192.168.1.4
live non-optimized
+- nvme20 tcp traddr=192.168.2.36 trsvcid=4420 host_traddr=192.168.2.5
live optimized
+- nvme21 tcp traddr=192.168.1.31 trsvcid=4420 host_traddr=192.168.2.5
live optimized
+- nvme30 tcp traddr=192.168.2.37 trsvcid=4420 host_traddr=192.168.2.5
live non-optimized
+- nvme31 tcp traddr=192.168.1.32 trsvcid=4420 host_traddr=192.168.2.5
live non-optimized
```

5. 验证 NetApp 插件是否为每个 ONTAP 命名空间设备显示了正确的值。例如：

```
# nvme netapp ontapdevices -o column
Device       Vserver          Namespace Path
---------    -------
-------------------------------------------------
/dev/nvme1n1 vserver_fcnvme_145 /vol/fcnvme_145_vol_1_0_0/fcnvme_145_ns

NSID  UUID                                   Size
----  ------------------------------         ------
1      23766b68-e261-444e-b378-2e84dbe0e5e1  85.90GB


# nvme netapp ontapdevices -o json
{
"ONTAPdevices" : [
     {
       "Device" : "/dev/nvme1n1",
       "Vserver" : "vserver_fcnvme_145",
       "Namespace_Path" : "/vol/fcnvme_145_vol_1_0_0/fcnvme_145_ns",
       "NSID" : 1,
       "UUID" : "23766b68-e261-444e-b378-2e84dbe0e5e1",
       "Size" : "85.90GB",
       "LBA_Data_Size" : 4096,
       "Namespace_Size" : 20971520
     }
  ]
}
```

另一个示例：

```
# nvme netapp ontapdevices -o column
Device       Vserver          Namespace Path
---------    -------
-------------------------------------------------
/dev/nvme0n1 vs_tcp_114       /vol/tcpnvme_114_1_0_1/tcpnvme_114_ns

NSID  UUID                                   Size
----  ------------------------------         ------
1      a6aee036-e12f-4b07-8e79-4d38a9165686  85.90GB


# nvme netapp ontapdevices -o json
{
     "ONTAPdevices" : [
     {
          "Device" : "/dev/nvme0n1",
           "Vserver" : "vs_tcp_114",
          "Namespace_Path" : "/vol/tcpnvme_114_1_0_1/tcpnvme_114_ns",
          "NSID" : 1,
          "UUID" : "a6aee036-e12f-4b07-8e79-4d38a9165686",
          "Size" : "85.90GB",
          "LBA_Data_Size" : 4096,
          "Namespace_Size" : 20971520
       }
  ]

}
```

已知问题

没有已知问题。

适用于采用 ONTAP 的 SUSE Linux Enterprise Server 15 SP2 的 NVMe/FC 主机配置

NVMe/FC在SUSE LINUX Enterprise Server 15 SP2的ONTAP 9 6及更高版本上受支持。SUSE Linux Enterprise Server 15 SP2主机可通过相同的光纤通道启动程序适配器端口同时运行NVMe/FC和FCP流量。有关支持的FC适配器和控制器的列表、请参见 "Hardware Universe" 。

有关支持的最新配置和版本列表，请参见 "NetApp 互操作性表"。

> 您可以使用本文档中提供的配置设置来配置连接到的云客户端 "Cloud Volumes ONTAP" 和 "适用于 ONTAP 的 Amazon FSX"。

已知限制

目前不支持使用NVMe-oF协议启动SAN。

在SUSE Linux Enterprise Server 15 SP2上启用NVMe/FC

1. 升级到建议的SUSE Linux Enterprise Server 15 SP2 MU内核版本。
2. 升级原生 NVMe-CLI 软件包。

   此原生 NVMe-CLI 软件包包含 NVMe/FC 自动连接脚本，用于为 NVMe 多路径启用轮循负载平衡的 ONTAP udev 规则以及用于 ONTAP 命名空间的 NetApp 插件。

   ```
   # rpm -qa|grep nvme-cli
   nvme-cli-1.10-2.38.x86_64
   ```

3. 在SUSE Linux Enterprise Server 15 SP2主机上、检查上的主机NQN字符串 `/etc/nvme/hostnqn`、并验证它是否与ONTAP阵列上对应子系统的主机NQN字符串匹配。例如：

   ```
   # cat /etc/nvme/hostnqn
   nqn.2014-08.org.nvmexpress:uuid:3ca559e1-5588-4fc4-b7d6-5ccfb0b9f054
   ```

   ```
   ::> vserver nvme subsystem host show -vserver vs_fcnvme_145
   Vserver Subsystem Host NQN
   ------- ---------
   ------------------------------------------------------------
   vs_fcnvme_145
   nvme_145_1
   nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-41a6-a1fd-cf8262c8713f
   nvme_145_2
   nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-41a6-a1fd-cf8262c8713f
   nvme_145_3
   nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-41a6-a1fd-cf8262c8713f
   nvme_145_4
   nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-41a6-a1fd-cf8262c8713f
   nvme_145_5
   nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-41a6-a1fd-cf8262c8713f
   5 entries were displayed.
   ```

4. 重新启动主机。

为 NVMe/FC 配置 Broadcom FC 适配器

1. 验证您使用的是受支持的适配器。有关支持的适配器的最新列表，请参见 "NetApp 互操作性表"。

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
```

```
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. 验证您使用的是建议的 Broadcom lpfc 固件和原生收件箱驱动程序版本。

```
# cat /sys/class/scsi_host/host*/fwrev
12.6.240.40, sli-4:2:c
12.6.240.40, sli-4:2:c
```

```
# cat /sys/module/lpfc/version
0:12.8.0.2
```

3. 验证 lpfc_enable_FC4_type 是否设置为 3 。

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

4. 验证启动程序端口是否已启动且正在运行。

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b579d5e
0x100000109b579d5f
```

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

5. 验证 NVMe/FC 启动程序端口是否已启用，正在运行且能够查看目标 LIF 。

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b579d5e WWNN x200000109b579d5e DID
x011c00 ONLINE
NVME RPORT WWPN x208400a098dfdd91 WWNN x208100a098dfdd91 DID x011503
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208500a098dfdd91 WWNN x208100a098dfdd91 DID x010003
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000e49 Cmpl 0000000e49 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003ceb594f Issue 000000003ce65dbe OutIO
fffffffffffb046f
abort 00000bd2 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 000014f4 Err 00012abd
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b579d5f WWNN x200000109b579d5f DID
x011b00 ONLINE
NVME RPORT WWPN x208300a098dfdd91 WWNN x208100a098dfdd91 DID x010c03
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208200a098dfdd91 WWNN x208100a098dfdd91 DID x012a03
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000e50 Cmpl 0000000e50 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003c9859ca Issue 000000003c93515e OutIO
fffffffffffaf794
abort 00000b73 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000159d Err 000135c3
```

验证 **NVMe/FC**

1. 验证以下 NVMe/FC 设置。

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

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
```

2. 验证是否已创建命名空间。

```
# nvme list
Node SN Model Namespace Usage Format FW Rev
---------------- --------------------
---------------------------------------- ---------
------------------------- ---------------- --------
/dev/nvme1n1 814vWBNRwfBGAAAAAAAB NetApp ONTAP Controller 1 85.90 GB /
85.90 GB 4 KiB + 0 B FFFFFFFF
```

3. 验证 ANA 路径的状态。

```
# nvme list-subsys /dev/nvme1n1
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.04ba0732530911ea8e8300a098dfdd91:subsystem.nvme_145_1
\
+- nvme2 fc traddr=nn-0x208100a098dfdd91:pn-0x208200a098dfdd91
host_traddr=nn-0x200000109b579d5f:pn-0x100000109b579d5f live
inaccessible
+- nvme3 fc traddr=nn-0x208100a098dfdd91:pn-0x208500a098dfdd91
host_traddr=nn-0x200000109b579d5e:pn-0x100000109b579d5e live
inaccessible
+- nvme4 fc traddr=nn-0x208100a098dfdd91:pn-0x208400a098dfdd91
host_traddr=nn-0x200000109b579d5e:pn-0x100000109b579d5e live optimized
+- nvme6 fc traddr=nn-0x208100a098dfdd91:pn-0x208300a098dfdd91
host_traddr=nn-0x200000109b579d5f:pn-0x100000109b579d5f live optimized
```

4. 验证适用于 ONTAP 设备的 NetApp 插件。

```
# nvme netapp ontapdevices -o column
Device Vserver Namespace Path NSID UUID Size
--------------- -------------------------
------------------------------------------------ ----
------------------------------------ ---------
/dev/nvme1n1 vserver_fcnvme_145 /vol/fcnvme_145_vol_1_0_0/fcnvme_145_ns
1 23766b68-e261-444e-b378-2e84dbe0e5e1 85.90GB

# nvme netapp ontapdevices -o json
{
"ONTAPdevices" : [
     {
       "Device" : "/dev/nvme1n1",
       "Vserver" : "vserver_fcnvme_145",
       "Namespace_Path" : "/vol/fcnvme_145_vol_1_0_0/fcnvme_145_ns",
       "NSID" : 1,
       "UUID" : "23766b68-e261-444e-b378-2e84dbe0e5e1",
       "Size" : "85.90GB",
       "LBA_Data_Size" : 4096,
       "Namespace_Size" : 20971520
     },
  ]
}
```

已知问题

没有已知问题。

为 **Broadcom NVMe/FC** 启用 **1 MB I/O** 大小

ONTAP会在"识别 控制器"数据中报告MDTS (MAX Data传输大小)为8。这意味着最大I/O请求大小最多可以为1 MB。要向Broadcom NVMe/FC主机发出大小为1 MB的I/O请求、应将参数的值 `lpfc_sg_seg_cnt`从默认值64增加 `lpfc`到256。

> 这些步骤不适用于逻辑NVMe/FC主机。

步骤

1. 将 `lpfc_sg_seg_cnt`参数设置为256：

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

```
options lpfc lpfc_sg_seg_cnt=256
```

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

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

lpfc 详细日志记录

设置NVMe/FC的lpfc驱动程序。

步骤

1. 设置 `lpfc_log_verbose` 将驱动程序设置为以下任意值以记录NVMe/FC事件。

```
#define LOG_NVME 0x00100000 /* NVME general events. */
#define LOG_NVME_DISC 0x00200000 /* NVME Discovery/Connect events. */
#define LOG_NVME_ABTS 0x00400000 /* NVME ABTS events. */
#define LOG_NVME_IOERR 0x00800000 /* NVME IO Error events. */
```

2. 设置值后、运行 `dracut-f` 命令并重新启动主机。
3. 验证设置。

```
# cat /etc/modprobe.d/lpfc.conf options lpfc lpfc_log_verbose=0xf00083

# cat /sys/module/lpfc/parameters/lpfc_log_verbose 15728771
```

适用于采用 ONTAP 的 SUSE Linux Enterprise Server 15 SP1 的 NVMe/FC 主机配置

您可以在运行SUSE Linux Enterprise Server 15 SP1和ONTAP的主机上将基于光纤通道的NVMe (NVMe/FC)配置为目标。

以下版本的SUSE Linux Enterprise Server在ONTAP 9 6或更高版本上支持NVMe/FC：

- SUSE Linux Enterprise Server 15 SP1

  SUSE Linux Enterprise Server 15 SP1主机可以通过相同的光纤通道启动程序适配器端口同时运行NVMe/FC和FCP流量。有关支持的FC适配器和控制器的列表、请参见 "Hardware Universe" 。

  有关支持的最新配置和版本列表，请参见 "NetApp 互操作性表"。

- NVMe-CLI 软件包中包含原生 NVMe/FC 自动连接脚本。您可以在SUSE Linux Enterprise Server 15 SP1上使用内置的lpfc驱动程序。

已知限制

目前不支持使用NVMe-oF协议启动SAN。

在SUSE Linux Enterprise Server 15 SP1上启用NVMe/FC

1. 升级到建议的SUSE Linux Enterprise Server 15 SP2 MU内核
2. 升级到建议的 NVMe-CLI MU 版本。

   此NVMe-CLI软件包包含本机NVMe/FC自动连接脚本、因此您无需在SUSE Linux Enterprise Server 15 SP1主机上安装Broadcom提供的外部NVMe/FC自动连接脚本。此软件包还包括用于为 NVMe 多路径启用轮循负载平衡的 ONTAP udev 规则以及用于 ONTAP 设备的 NetApp 插件。

   ```
   # rpm -qa | grep nvme-cli
   nvme-cli-1.8.1-6.9.1.x86_64
   ```

3. 在SUSE Linux Enterprise Server 15 SP1主机上、检查位于的主机NQN字符串 `/etc/nvme/hostnqn`、并验证它是否与ONTAP阵列上对应子系统的主机NQN字符串匹配。例如：

   ```
   # cat /etc/nvme/hostnqn
   nqn.2014-08.org.nvmexpress:uuid:75953f3b-77fe-4e03-bf3c-09d5a156fbcd
   ```

   ```
   *> vserver nvme subsystem host show -vserver vs_nvme_10
   Vserver Subsystem Host NQN
   ------- --------- ------------------------------------ -----------
   sles_117_nvme_ss_10_0
   nqn.2014-08.org.nvmexpress:uuid:75953f3b-77fe-4e03-bf3c-09d5a156fbcd
   ```

4. 重新启动主机。

为 NVMe/FC 配置 Broadcom FC 适配器

1. 验证您使用的是受支持的适配器。有关支持的适配器的最新列表，请参见 "NetApp 互操作性表"。

   ```
   # cat /sys/class/scsi_host/host*/modelname
   LPe32002-M2
   LPe32002-M2
   ```

   ```
   # cat /sys/class/scsi_host/host*/modeldesc
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   ```

2. 验证您使用的是建议的 Broadcom lpfc 固件和原生收件箱驱动程序版本。

```
# cat /sys/class/scsi_host/host*/fwrev
12.4.243.17, sil-4.2.c
12.4.243.17, sil-4.2.c
```

```
# cat /sys/module/lpfc/version
0:12.6.0.0
```

3. 验证 lpfc_enable_FC4_type 是否设置为 3 。

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

4. 验证启动程序端口是否已启动且正在运行。

```
# cat /sys/class/fc_host/host*/port_name
0x10000090fae0ec61
0x10000090fae0ec62
```

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

5. 验证 NVMe/FC 启动程序端口是否已启用，正在运行且能够查看目标 LIF 。

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 NVME 2947 SCSI 2977 ELS 250
NVME LPORT lpfc0 WWPN x10000090fae0ec61 WWNN x20000090fae0ec61 DID
x012000 ONLINE
NVME RPORT WWPN x202d00a098c80f09 WWNN x202c00a098c80f09 DID x010201
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203100a098c80f09 WWNN x202c00a098c80f09 DID x010601
TARGET DISCSRVC ONLINE
NVME Statistics
…
```

验证 NVMe/FC

1. 验证以下 NVMe/FC 设置。

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

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

2. 验证是否已创建命名空间。

```
# nvme list
Node SN Model Namespace Usage Format FW Rev
---------------- -------------------- ----------------------
/dev/nvme0n1 80BADBKnB/JvAAAAAAAC NetApp ONTAP Controller 1 53.69 GB /
53.69 GB 4 KiB + 0 B FFFFFFFF
```

3. 验证 ANA 路径的状态。

```
# nvme list-subsys/dev/nvme0n1
Nvme-subsysf0 – NQN=nqn.1992-
08.com.netapp:sn.341541339b9511e8a9b500a098c80f09:subsystem.sles_117_nvm
e_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
```

4. 验证适用于 ONTAP 设备的 NetApp 插件。

```
# nvme netapp ontapdevices -o column
Device   Vserver  Namespace Path             NSID   UUID   Size
-------  -------- -------------------------  ------ ----- -----
/dev/nvme0n1   vs_nvme_10       /vol/sles_117_vol_10_0/sles_117_ns_10_0
1        55baf453-f629-4a18-9364-b6aee3f50dad   53.69GB

# nvme netapp ontapdevices -o json
{
   "ONTAPdevices" : [
   {
        Device" : "/dev/nvme0n1",
        "Vserver" : "vs_nvme_10",
        "Namespace_Path" : "/vol/sles_117_vol_10_0/sles_117_ns_10_0",
         "NSID" : 1,
         "UUID" : "55baf453-f629-4a18-9364-b6aee3f50dad",
         "Size" : "53.69GB",
         "LBA_Data_Size" : 4096,
         "Namespace_Size" : 13107200
    }
]
```

已知问题

没有已知问题。

为 Broadcom NVMe/FC 启用 1 MB I/O 大小

ONTAP会在"识别 控制器"数据中报告MDTS (MAX Data传输大小)为8。这意味着最大I/O请求大小最多可以为1 MB。要向Broadcom NVMe/FC主机发出大小为1 MB的I/O请求、应将参数的值 `lpfc_sg_seg_cnt`从默认值64增加 `lpfc`到256。

这些步骤不适用于逻辑NVMe/FC主机。

步骤

1. 将 `lpfc_sg_seg_cnt`参数设置为256：

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

```
options lpfc lpfc_sg_seg_cnt=256
```

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

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

lpfc 详细日志记录

设置NVMe/FC的lpfc驱动程序。

步骤

1. 设置 `lpfc_log_verbose` 将驱动程序设置为以下任意值以记录NVMe/FC事件。

```
#define LOG_NVME 0x00100000 /* NVME general events. */
#define LOG_NVME_DISC 0x00200000 /* NVME Discovery/Connect events. */
#define LOG_NVME_ABTS 0x00400000 /* NVME ABTS events. */
#define LOG_NVME_IOERR 0x00800000 /* NVME IO Error events. */
```

2. 设置值后、运行 `dracut-f` 命令并重新启动主机。
3. 验证设置。

```
# cat /etc/modprobe.d/lpfc.conf options lpfc lpfc_log_verbose=0xf00083

# cat /sys/module/lpfc/parameters/lpfc_log_verbose 15728771
```