SUSE Linux Enterprise Server

ONTAP SAN Host Utilities

NetApp
January 30, 2026

Sommario

SUSE Linux Enterprise Server

Scopri di più sul supporto e sulle funzionalità ONTAP per SUSE Linux Enterprise Server

Le funzionalità supportate per la configurazione host con NVMe over Fabrics (NVMe-oF) variano in base alla versione di ONTAP e SUSE Linux Enterprise Server.

Caratteristica	Versione host di SUSE Linux Enterprise Server	Versione ONTAP
NVMe/TCP segnala tutti i percorsi ottimizzati per i sistemi ASA r2	16	9.16.1 o successivo
La crittografia Transport Layer Security (TLS) 1.3 è supportata per NVMe/TCP	15 SP6 o successivo	9.16.1 o successivo
L'autenticazione in-band sicura è supportata tramite NVMe/TCP tra un host RHEL e un controller ONTAP	15 SP4 o successivo	9.12.1 o successivo
I controller di rilevamento persistenti (PDC) sono supportati utilizzando un NQN di rilevamento univoco	15 SP4 o successivo	9.11.1 o successivo
NVMe/TCP fornisce spazi dei nomi utilizzando il nativo `nvme-cli` pacchetto	15 SP4 o successivo	9.10.1 o successivo
Il traffico NVMe e SCSI è supportato sullo stesso host utilizzando NVMe multipath per gli spazi dei nomi NVMe-oF e dm-multipath per le LUN SCSI	15 SP1 o successivo	9.4 o successivo

ONTAP supporta le seguenti funzionalità host SAN indipendentemente dalla versione ONTAP in esecuzione sulla configurazione del sistema.

Caratteristica	Versione host di SUSE Linux Enterprise Server
L'avvio SAN è abilitato tramite il protocollo NVMe/FC	15 SP7 o successivo
Il multipathing NVMe nativo è abilitato per impostazione predefinita	15 SP1 o successivo
IL `nvme-cli` il pacchetto include script di connessione automatica che eliminano la necessità di script di terze parti	15 SP1 o successivo

Per i dettagli sulle configurazioni supportate, vedere"Tool di matrice di interoperabilità" .

Cosa c'è dopo?

Se la versione di SUSE Linux Enterprise Server è …	Scopri …
16	"Configurazione di NVMe per SUSE Linux Enterprise Server 16"

Se la versione di SUSE Linux Enterprise Server è …	Scopri …
Serie 15 SPx	"Configurazione di NVMe per SUSE Linux Enterprise Server 15 SPx"

Informazioni correlate

- "Scopri i sistemi ASA r2"
- "Scopri di più sulla gestione dei protocolli NVMe"

Configurare SUSE Linux Enterprise Server 16 per NVMe-oF con storage ONTAP

L'host SUSE Linux Enterprise Server 16 supporta i protocolli NVMe over Fibre Channel (NVMe/FC) e NVMe over TCP (NVMe/TCP) con Asymmetric Namespace Access (ANA). ANA fornisce funzionalità di multipathing equivalenti ad Asymmetric Logical Unit Access (ALUA) negli ambienti iSCSI e FCP.

Scopri come configurare gli host NVMe over Fabrics (NVMe-oF) per SUSE Linux Enterprise Server 16. Per ulteriori informazioni su supporto e funzionalità, vedi "Supporto e funzionalità ONTAP".

NVMe-oF con SUSE Linux Enterprise Server 16 presenta le seguenti limitazioni note:

- IL `nvme disconnect-all` Il comando disconnette sia il file system root che quello dati e potrebbe causare instabilità del sistema. Non eseguire questa operazione su sistemi che si avviano da SAN tramite namespace NVMe-TCP o NVMe-FC.
- Il supporto dell'utilità host NetApp sanlun non è disponibile per NVMe-oF. In alternativa, puoi affidarti al plug-in NetApp incluso nel nativo `nvme-cli` pacchetto per tutti i trasporti NVMe-oF.

Passaggio 1: Se lo si desidera, attivare l'avvio SAN

È possibile configurare l'host per utilizzare l'avvio SAN per semplificare la distribuzione e migliorare la scalabilità. Utilizzare il"Tool di matrice di interoperabilità" per verificare che il sistema operativo Linux, l'adattatore bus host (HBA), il firmware HBA, il BIOS di avvio HBA e la versione ONTAP supportino l'avvio SAN.

Fasi

1. "Crea uno spazio dei nomi NVMe e mappalo all'host" .
2. Abilitare l'avvio SAN nel BIOS del server per le porte su cui è mappato lo spazio dei nomi di avvio SAN.

   Per informazioni su come attivare il BIOS HBA, consultare la documentazione specifica del vendor.

3. Riavviare l'host e verificare che il sistema operativo sia attivo e funzionante.

Passaggio 2: installare SUSE Linux Enterprise Server e il software NVMe e verificare la configurazione

Per configurare l'host per NVMe-oF è necessario installare i pacchetti software host e NVMe, abilitare il multipathing e verificare la configurazione NQN dell'host.

Fasi

1. Installare SUSE Linux Enterprise Server 16 sul server. Dopo il completamento dell'installazione, verificare di eseguire il kernel SUSE Linux Enterprise Server 16 specificato:

```
uname -r
```

Esempio di versione del kernel di SUSE Linux Enterprise Server:

```
6.12.0-160000.6-default
```

2. Installare `nvme-cli` pacchetto:

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

L'esempio seguente mostra un `nvme-cli` versione del pacchetto:

```
nvme-cli-2.11+29.g35e62868-160000.1.1.x86_64
```

3. Installare `libnvme` pacchetto:

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

L'esempio seguente mostra un `libnvme` versione del pacchetto:

```
libnvme1-1.11+17.g6d55624d-160000.1.1.x86_64
```

4. Sull'host, controlla la stringa hostnqn in `/etc/nvme/hostnqn` :

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

L'esempio seguente mostra un `hostnqn` versione:

```
nqn.2014-08.org.nvmexpress:uuid:d3b581b4-c975-11e6-8425-0894ef31a074
```

5. Sul sistema ONTAP , verificare che `hostnqn` la stringa corrisponde a `hostnqn` stringa per il sottosistema corrispondente nell'array ONTAP :

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

Mostra esempio

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_coexistence_emulex
        nvme1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:d3b581b4-c975-11e6-8425-0894ef31a074
        nvme10
                  regular   nqn.2014-
08.org.nvmexpress:uuid:d3b581b4-c975-11e6-8425-0894ef31a074
        nvme11
                  regular   nqn.2014-
08.org.nvmexpress:uuid:d3b581b4-c975-11e6-8425-0894ef31a074
        nvme12
                  regular   nqn.2014-
08.org.nvmexpress:uuid:d3b581b4-c975-11e6-8425-0894ef31a074
4 entries were displayed.
```

Se il `hostnqn` le stringhe non corrispondono, utilizzare `vserver modify` per aggiornare `hostnqn` Stringa sul sottosistema di array ONTAP corrispondente a `hostnqn` stringa da `/etc/nvme/hostnqn` sull'host.

Passaggio 3: configurare NVMe/FC e NVMe/TCP

Configurare NVMe/FC con adattatori Broadcom/Emulex o Marvell/QLogic oppure configurare NVMe/TCP utilizzando operazioni di rilevamento e connessione manuali.

NVMe/FC - Broadcom/Emulex

Configurare NVMe/FC per un adattatore FC Broadcom/Emulex.

Fasi

1. Verificare di utilizzare il modello di adattatore supportato:

   a. Visualizza i nomi dei modelli:

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

   Viene visualizzato il seguente output:

   ```
   SN37A92079
   SN37A92079
   ```

   b. Visualizza le descrizioni dei modelli:

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

   Viene visualizzato il seguente output:

   ```
   Emulex SN37A92079 32Gb 2-Port Fibre Channel Adapter
   Emulex SN37A92079 32Gb 2-Port Fibre Channel Adapter
   ```

2. Verificare di utilizzare il Broadcom consigliato `lpfc` firmware e driver della posta in arrivo:

   a. Visualizza la versione del firmware:

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

   L'esempio seguente mostra le versioni del firmware:

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

   b. Visualizza la versione del driver in arrivo:

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

   L'esempio seguente mostra la versione del driver:

```
0:14.4.0.11
```

Per l'elenco aggiornato dei driver della scheda di rete supportati e delle versioni del firmware, vedere "Tool di matrice di interoperabilità".

3. Verificare che l'output previsto di `lpfc_enable_fc4_type` sia impostato su `3`:

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

4. Verificare che sia possibile visualizzare le porte dell'iniziatore:

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

Dovresti vedere un output simile a:

```
0x100000109bdacc75
0x100000109bdacc76
```

5. Verificare che le porte dell'iniziatore siano in linea:

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

Viene visualizzato il seguente output:

```
Online
Online
```

6. Verificare che le porte iniziatore NVMe/FC siano abilitate e che le porte di destinazione siano visibili:

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

Mostra output di esempio

```
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109bdacc75 WWNN x200000109bdacc75
DID x060100 ONLINE
NVME RPORT       WWPN x2001d039ea951c45 WWNN x2000d039ea951c45
DID x080801 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2003d039ea951c45 WWNN x2000d039ea951c45
DID x080d01 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2024d039eab31e9c WWNN x2023d039eab31e9c
DID x020a09 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2026d039eab31e9c WWNN x2023d039eab31e9c
DID x020a08 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2003d039ea5cfc90 WWNN x2002d039ea5cfc90
DID x061b01 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2012d039ea5cfc90 WWNN x2011d039ea5cfc90
DID x061b05 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2005d039ea5cfc90 WWNN x2002d039ea5cfc90
DID x061201 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2014d039ea5cfc90 WWNN x2011d039ea5cfc90
DID x061205 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000017242 Cmpl 0000017242 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000000378362 Issue 00000000003783c7 OutIO
0000000000000065
        abort 00000409 noxri 00000000 nondlp 0000003a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000409 Err 0000040a

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109bdacc76 WWNN x200000109bdacc76
DID x062800 ONLINE
NVME RPORT       WWPN x2002d039ea951c45 WWNN x2000d039ea951c45
DID x080701 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2004d039ea951c45 WWNN x2000d039ea951c45
DID x081501 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2025d039eab31e9c WWNN x2023d039eab31e9c
DID x020913 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2027d039eab31e9c WWNN x2023d039eab31e9c
DID x020912 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2006d039ea5cfc90 WWNN x2002d039ea5cfc90
DID x061401 TARGET DISCSRVC ONLINE
```

```
NVME RPORT       WWPN x2015d039ea5cfc90 WWNN x2011d039ea5cfc90
DID x061405 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2004d039ea5cfc90 WWNN x2002d039ea5cfc90
DID x061301 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2013d039ea5cfc90 WWNN x2011d039ea5cfc90
DID x061305 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000017428 Cmpl 0000017428 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000003443be Issue 000000000034442a OutIO
000000000000006c
        abort 00000491 noxri 00000000 nondlp 00000086 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000491 Err 00000494
```

NVMe/FC - Marvell/QLogic

Configurare NVMe/FC per un adattatore Marvell/QLogic.

Fasi

1. Verificare che siano in esecuzione le versioni del firmware e del driver dell'adattatore supportate:

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

   L'esempio seguente mostra le versioni del driver e del firmware:

   ```
   QLE2772 FW:v9.15.06 DVR:v10.02.09.400-k-debug
   QLE2772 FW:v9.15.06 DVR:v10.02.09.400-k-debug
   ```

2. Verificare che `ql2xnvmeenable` è impostato. Ciò consente all'adattatore Marvell di funzionare come iniziatore NVMe/FC:

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

   L'uscita prevista è 1.

NVMe/TCP

Il protocollo NVMe/TCP non supporta l'operazione di connessione automatica. In alternativa, è possibile scoprire i sottosistemi e gli spazi dei nomi NVMe/TCP eseguendo l'NVMe/TCP `connect` O `connect-all` operazioni manualmente.

Fasi

1. Verificare che la porta iniziatore possa recuperare i dati della pagina del registro di rilevamento attraverso le LIF NVMe/TCP supportate:

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

Mostra output di esempio

```
nvme discover -t tcp -w 192.168.38.20 -a 192.168.38.10
Discovery Log Number of Records 8, Generation counter 42
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr:  192.168.211.71
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  3
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr:  192.168.111.71
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr:  192.168.211.70
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr:  192.168.111.70
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.samp
le_tcp_sub
traddr:  192.168.211.71
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  3
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.samp
le_tcp_sub
traddr:  192.168.111.71
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.samp
le_tcp_sub
```

```
traddr:  192.168.211.70
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.samp
le_tcp_sub
traddr:  192.168.111.70
eflags:  none
sectype: none
localhost:~ #
```

2. Verifica che tutte le altre combinazioni di LIF iniziatore NVMe/TCP siano in grado di recuperare con successo i dati della pagina del log di rilevamento:

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

Mostra esempio

```
nvme discover -t tcp -w 192.168.38.20 -a 192.168.38.10
nvme discover -t tcp -w 192.168.38.20 -a 192.168.38.11
nvme discover -t tcp -w 192.168.39.20 -a 192.168.39.10
nvme discover -t tcp -w 192.168.39.20 -a 192.168.39.11
```

3. Eseguire `nvme connect-all` Command tra tutti i LIF target initiator NVMe/TCP supportati nei nodi:

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

Mostra esempio

```
nvme    connect-all -t  tcp -w  192.168.38.20   -a
192.168.38.10
nvme    connect-all -t  tcp -w  192.168.38.20   -a
192.168.38.11
nvme    connect-all -t  tcp -w  192.168.39.20   -a
192.168.39.10
nvme    connect-all -t  tcp -w  192.168.39.20   -a
192.168.39.11
```

L'impostazione per NVMe/TCP `ctrl_loss_tmo timeout` viene automaticamente impostato su "off". Di conseguenza:

- Non ci sono limiti al numero di tentativi (tentativi illimitati).
- Non è necessario configurare manualmente uno specifico `ctrl_loss_tmo timeout` durata quando si utilizza il `nvme connect` O `nvme connect-all` comandi (opzione -l).
- I controller NVMe/TCP non subiscono timeout in caso di errore del percorso e rimangono connessi indefinitamente.

Passaggio 4: Facoltativamente, modificare iopolicy nelle regole udev

A partire da SUSE Linux Enterprise Server 16, la iopolicy predefinita per NVMe-oF è impostata su `queue-depth`. Se si desidera modificare la iopolicy in `round-robin`, modificare il file delle regole udev come segue:

Fasi

1. Aprire il file delle regole udev in un editor di testo con privilegi di root:

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

   Viene visualizzato il seguente output:

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

2. Trova la riga che imposta iopolicy per NetApp ONTAP Controller, come mostrato nella seguente regola di esempio:

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

3. Modifica la regola in modo che `queue-depth` diventi `round-robin`:

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

4. Ricarica le regole udev e applica le modifiche:

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

5. Verificare l'attuale iopolicy per il sottosistema. Sostituisci <sottosistema>, ad esempio, `nvme-subsys0` .

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

Viene visualizzato il seguente output:

```
round-robin
```

> La nuova iopolicy si applica automaticamente ai dispositivi NetApp ONTAP Controller corrispondenti. Non è necessario riavviare.

Passaggio 5: Facoltativamente, abilitare 1 MB di I/O per NVMe/FC

ONTAP segnala una dimensione massima di trasferimento dati (MDTS) pari a 8 nei dati Identify Controller. Ciò significa che la dimensione massima della richiesta di I/O può arrivare fino a 1 MB. Per emettere richieste di I/O di dimensione 1 MB per un host Broadcom NVMe/FC, è necessario aumentare il `lpfc` valore del `lpfc_sg_seg_cnt` parametro a 256 dal valore predefinito di 64.

> Questi passaggi non si applicano agli host Qlogic NVMe/FC.

Fasi

1. Impostare il `lpfc_sg_seg_cnt` parametro su 256:

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

Dovresti vedere un output simile al seguente esempio:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Eseguire il `dracut -f` comando e riavviare l'host.

3. Verificare che il valore per `lpfc_sg_seg_cnt` sia 256:

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

Passaggio 6: verificare i servizi di avvio NVMe

IL `nvmefc-boot-connections.service` E `nvmf-autoconnect.service` servizi di avvio inclusi in NVMe/FC `nvme-cli` i pacchetti vengono abilitati automaticamente all'avvio del sistema.

Dopo aver completato l'avvio, verificare che `nvmefc-boot-connections.service` E `nvmf-autoconnect.service` i servizi di avvio sono abilitati.

Fasi

1. Verificare che `nvmf-autoconnect.service` sia attivato:

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

Mostra output di esempio

```
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. Verificare che `nvmefc-boot-connections.service` sia attivato:

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

Mostra output di esempio

```
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.
```

Passaggio 7: verificare la configurazione del multipathing

Verificare che lo stato multipath NVMe in-kernel, lo stato ANA e i namespace ONTAP siano corretti per la configurazione NVMe-of.

Fasi

1. Verificare che il multipath NVMe nel kernel sia attivato:

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

   Viene visualizzato il seguente output:

   ```
   Y
   ```

2. Verificare che le impostazioni NVMe-oF appropriate (ad esempio, modello impostato su NetApp ONTAP Controller e iopolicy di bilanciamento del carico impostato su queue-depth) per i rispettivi namespace ONTAP si riflettano correttamente sull'host:

   a. Visualizza i sottosistemi:

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

      Viene visualizzato il seguente output:

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

b. Visualizza la politica:

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

Viene visualizzato il seguente output:

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

3. Verificare che gli spazi dei nomi siano stati creati e rilevati correttamente sull'host:

```
nvme list
```

Mostra esempio

```
Node         SN                   Model
------------------------------------------------------------
/dev/nvme7n1 81Ix2BVuekWcAAAAAAAB   NetApp ONTAP Controller

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

4. Verificare che lo stato del controller di ciascun percorso sia attivo e che abbia lo stato ANA corretto:

```
nvme list-subsys /dev/<controller_ID>
```

A partire da ONTAP 9.16.1, NVMe/FC e NVMe/TCP segnalano tutti i percorsi ottimizzati sui sistemi ASA r2.

NVMe/FC

I seguenti output di esempio mostrano uno spazio dei nomi ospitato su un controller ONTAP a due nodi per sistemi AFF, FAS e ASA e su un sistema ASA r2 con NVMe/FC.

Mostra l'output di esempio AFF, FAS e ASA

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

Mostra l'output di esempio di ASA r2

```
nvme-subsys96 - NQN=nqn.1992-
08.om.netapp:sn.b351b2b6777b11f0b3c2d039ea5cfc91:subsystem.nvme2
4
                hostnqn=nqn.2014-
08.org.nvmexpress:uuid:d3b581b4-c975-11e6-8425-0894ef31a074
\
 +- nvme203 fc traddr=nn-0x2011d039ea5cfc90:pn-
0x2015d039ea5cfc90,host_traddr=nn-0x200000109bdacc76:pn-
0x100000109bdacc76 live optimized
 +- nvme25 fc traddr=nn-0x2011d039ea5cfc90:pn-
0x2014d039ea5cfc90,host_traddr=nn-0x200000109bdacc75:pn-
0x100000109bdacc75 live optimized
 +- nvme30 fc traddr=nn-0x2011d039ea5cfc90:pn-
0x2012d039ea5cfc90,host_traddr=nn-0x200000109bdacc75:pn-
0x100000109bdacc75 live optimized
 +- nvme32 fc traddr=nn-0x2011d039ea5cfc90:pn-
0x2013d039ea5cfc90,host_traddr=nn-0x200000109bdacc76:pn-
0x100000109bdacc76 live optimized
```

NVMe/TCP

I seguenti output di esempio mostrano uno spazio dei nomi ospitato su un controller ONTAP a due nodi per sistemi AFF, FAS e ASA e sistemi ASA r2 con NVMe/TCP.

Mostra l'output di esempio AFF, FAS e ASA

```
nvme-subsys9 - NQN=nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme
10
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-
0035-5910-804b-b7c04f444d33
\
 +- nvme105 tcp
traddr=192.168.39.10,trsvcid=4420,host_traddr=192.168.39.20,src_
addr=192.168.39.20 live optimized
 +- nvme153 tcp
traddr=192.168.39.11,trsvcid=4420,host_traddr=192.168.39.20,src_
addr=192.168.39.20 live non-optimized
 +- nvme57 tcp
traddr=192.168.38.11,trsvcid=4420,host_traddr=192.168.38.20,src_
addr=192.168.38.20 live non-optimized
 +- nvme9 tcp
traddr=192.168.38.10,trsvcid=4420,host_traddr=192.168.38.20,src_
addr=192.168.38.20 live optimized
```

Mostra l'output di esempio di ASA r2

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.17e32b6e8c7f11f09545d039eac03c33:subsystem.Bidi
rectional_DHCP_1_0
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-
0054-5110-8039-c3c04f523034
\
+- nvme4 tcp
traddr=192.168.20.28,trsvcid=4420,host_traddr=192.168.20.21,src_
addr=192.168.20.21 live optimized
+- nvme5 tcp
traddr=192.168.20.29,trsvcid=4420,host_traddr=192.168.20.21,src_
addr=192.168.20.21 live optimized
+- nvme6 tcp
traddr=192.168.21.28,trsvcid=4420,host_traddr=192.168.21.21,src_
addr=192.168.21.21 live optimized
+- nvme7 tcp
traddr=192.168.21.29,trsvcid=4420,host_traddr=192.168.21.21,src_
addr=192.168.21.21 live optimized
```

5. Verificare che il plug-in NetApp visualizzi i valori corretti per ciascun dispositivo dello spazio dei nomi ONTAP:

Colonna

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

Mostra esempio

```
Device           Vserver                   Namespace Path
NSID UUID                                   Size
---------------- ------------------------- ------------------
---- ---------------------------------------- ---------
/dev/nvme0n1     vs_coexistence_emulex     ns1                1
79510f05-7784-11f0-b3c2-d039ea5cfc91   21.47GB
```

JSON

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

Mostra esempio

```
{
"ONTAPdevices":[{
      "Device":"/dev/nvme0n1",
      "Vserver":"vs_coexistence_emulex",
      "Namespace_Path":"ns1",
      "NSID":1,
      "UUID":"79510f05-7784-11f0-b3c2-d039ea5cfc91",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    }  ]
}
```

Passaggio 8: creare un controller di scoperta persistente

È possibile creare un controller di individuazione persistente (PDC) per un host SUSE Linux Enterprise Server 16. Un PDC è necessario per rilevare automaticamente un'operazione di aggiunta o rimozione del sottosistema NVMe e le modifiche ai dati della pagina del registro di individuazione.

Fasi

1. Verificare che i dati della pagina del log di rilevamento siano disponibili e possano essere recuperati attraverso la combinazione di porta Initiator e LIF di destinazione:

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

Mostra output di esempio

```
Discovery Log Number of Records 8, Generation counter 10
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  3
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:discovery
traddr:  192.168.39.10
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.9927e165694211f0b4f4d039eab31e9d:discovery
traddr:  192.168.38.10
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:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:discovery
traddr:  192.168.39.11
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:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:discovery
traddr:  192.168.38.11
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  3
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme1
traddr:  192.168.39.10
eflags:  none
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.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme1
traddr:  192.168.38.10
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme1
traddr:  192.168.39.11
eflags:  none
sectype: none
=====Discovery Log Entry 7======
```

```
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme1
traddr:  192.168.38.11
eflags:  none
sectype: none
```

2. Creare un PDC per il sottosistema di rilevamento:

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

Viene visualizzato il seguente output:

```
nvme discover -t tcp -w 192.168.39.20 -a 192.168.39.11 -p
```

3. Dal controller ONTAP, verificare che il PDC sia stato creato:

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

Mostra output di esempio

```
vserver nvme show-discovery-controller -instance -vserver
vs_tcp_sles16
Vserver Name: vs_tcp_sles16
               Controller ID: 0180h
     Discovery Subsystem NQN: nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:discovery
           Logical Interface: lif3
                        Node: A400-12-171
                    Host NQN: nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b7c04f444d33
          Transport Protocol: nvme-tcp
 Initiator Transport Address: 192.168.39.20
Transport Service Identifier: 8009
             Host Identifier: 4c4c454400355910804bb7c04f444d33
            Admin Queue Depth: 32
       Header Digest Enabled: false
         Data Digest Enabled: false
   Keep-Alive Timeout (msec): 30000
```

Passaggio 9: impostare l’autenticazione in-band sicura

L’autenticazione in-band sicura è supportata tramite NVMe/TCP tra un host SUSE Linux Enterprise Server 16 e un controller ONTAP.

Ogni host o controller deve essere associato a un `DH-HMAC-CHAP` chiave per impostare l’autenticazione sicura. Una chiave DH-HMAC-CHAP è una combinazione dell’NQN dell’host o del controller NVMe e di un segreto di autenticazione configurato dall’amministratore. Per autenticare il suo peer, un host o un controller NVMe deve riconoscere la chiave associata al peer.

Fasi

Imposta l’autenticazione in-band sicura tramite la CLI o un file JSON di configurazione. Se è necessario specificare chiavi dhchap diverse per sottosistemi diversi, è necessario utilizzare un file di configurazione JSON.

CLI

Configurare l'autenticazione in banda protetta utilizzando la CLI.

1. Ottenere l'NQN dell'host:

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

2. Genera la chiave dhchap per l'host.

   L'output seguente descrive i `gen-dhchap-key` parametri dei comandi:

```
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
```

   Nell'esempio seguente, viene generata una chiave casuale dhCHAP con HMAC impostato su 3 (SHA-512).

```
nvme gen-dhchap-key -m 3 -n nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b7c04f444d33
DHHC-
1:03:ohdxI1yIS8gBLwIOubcwl57rXcozYuRgBsoWaBvxEvpDlQHn/7dQ4JjFGwmhgwd
JWmVoripbWbMJy5eMAbCahN4hhYU=:
```

3. Sul controller ONTAP, aggiungere l'host e specificare entrambe le chiavi dhchap:

```
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. Un host supporta due tipi di metodi di autenticazione, unidirezionale e bidirezionale. Sull'host, connettersi al controller ONTAP e specificare le chiavi dhchap in base al metodo di autenticazione scelto:

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

5. Convalidare `nvme connect authentication` comando verificando le chiavi dhchap dell'host e del controller:

   a. Verificare le chiavi dhchap dell'host:

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

   Mostra output di esempio per una configurazione unidirezionale

   ```
   # cat /sys/class/nvme-subsystem/nvme-
   subsys1/nvme*/dhchap_secret
   DHHC-1:01:wkwAKk8r9Ip7qECKt7V5aIo/7Y1CH7DWkUfLfMxmseg39DFb:
   DHHC-1:01:wkwAKk8r9Ip7qECKt7V5aIo/7Y1CH7DWkUfLfMxmseg39DFb:
   DHHC-1:01:wkwAKk8r9Ip7qECKt7V5aIo/7Y1CH7DWkUfLfMxmseg39DFb:
   DHHC-1:01:wkwAKk8r9Ip7qECKt7V5aIo/7Y1CH7DWkUfLfMxmseg39DFb:
   ```

   b. Verificare i tasti dhchap del controller:

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

   Mostra output di esempio per una configurazione bidirezionale

   ```
   # cat /sys/class/nvme-subsystem/nvme-
   subsys6/nvme*/dhchap_ctrl_secret
   DHHC-
   1:03:ohdxI1yIS8gBLwIOubcwl57rXcozYuRgBsoWaBvxEvpDlQHn/7dQ4JjFG
   wmhgwdJWmVoripbWbMJy5eMAbCahN4hhYU=:
   DHHC-
   1:03:ohdxI1yIS8gBLwIOubcwl57rXcozYuRgBsoWaBvxEvpDlQHn/7dQ4JjFG
   wmhgwdJWmVoripbWbMJy5eMAbCahN4hhYU=:
   DHHC-
   1:03:ohdxI1yIS8gBLwIOubcwl57rXcozYuRgBsoWaBvxEvpDlQHn/7dQ4JjFG
   wmhgwdJWmVoripbWbMJy5eMAbCahN4hhYU=:
   DHHC-
   1:03:ohdxI1yIS8gBLwIOubcwl57rXcozYuRgBsoWaBvxEvpDlQHn/7dQ4JjFG
   wmhgwdJWmVoripbWbMJy5eMAbCahN4hhYU=:
   ```

JSON

Quando sulla configurazione del controller ONTAP sono disponibili più sottosistemi NVMe, è possibile utilizzare il `/etc/nvme/config.json` file con il `nvme connect-all` comando.

Utilizzare il `-o` opzione per generare il file JSON. Per ulteriori opzioni di sintassi, fare riferimento alle pagine man di NVMe connect-all.

1. Configurare il file JSON:

Mostra output di esempio

```
# cat /etc/nvme/config.json
[
  {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-
5910-804b-b7c04f444d33",
    "hostid":"4c4c4544-0035-5910-804b-b7c04f444d33",
    "dhchap_key":"DHHC-
1:01:wkwAKk8r9Ip7qECKt7V5aIo/7Y1CH7DWkUfLfMxmseg39DFb:",
    "subsystems":[
      {
        "nqn":"nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:subsystem.inba
nd_bidirectional",
        "ports":[
          {
            "transport":"tcp",
            "traddr":"192.168.38.10",
            "host_traddr":"192.168.38.20",
            "trsvcid":"4420",
            "dhchap_ctrl_key":"DHHC-
1:03:ohdxI1yIS8gBLwIOubcwl57rXcozYuRgBsoWaBvxEvpDlQHn/7dQ4JjFGwm
hgwdJWmVoripbWbMJy5eMAbCahN4hhYU=:"
          },
          {
            "transport":"tcp",
            "traddr":"192.168.38.11",
            "host_traddr":"192.168.38.20",
            "trsvcid":"4420",
            "dhchap_ctrl_key":"DHHC-
1:03:ohdxI1yIS8gBLwIOubcwl57rXcozYuRgBsoWaBvxEvpDlQHn/7dQ4JjFGwm
hgwdJWmVoripbWbMJy5eMAbCahN4hhYU=:"
          },
          {
            "transport":"tcp",
            "traddr":"192.168.39.11",
            "host_traddr":"192.168.39.20",
            "trsvcid":"4420",
            "dhchap_ctrl_key":"DHHC-
1:03:ohdxI1yIS8gBLwIOubcwl57rXcozYuRgBsoWaBvxEvpDlQHn/7dQ4JjFGwm
hgwdJWmVoripbWbMJy5eMAbCahN4hhYU=:"
          },
          {
            "transport":"tcp",
            "traddr":"192.168.39.10",
```

```
            "host_traddr":"192.168.39.20",
            "trsvcid":"4420",
            "dhchap_ctrl_key":"DHHC-
1:03:ohdxI1yIS8gBLwIOubcwl57rXcozYuRgBsoWaBvxEvpDlQHn/7dQ4JjFGwm
hgwdJWmVoripbWbMJy5eMAbCahN4hhYU=:"
          }
        ]
      }
    ]
  }
]
```

Nell'esempio seguente, `dhchap_key` corrisponde a `dhchap_secret` E `dhchap_ctrl_key` corrisponde a `dhchap_ctrl_secret` .

2. Connettersi al controller ONTAP utilizzando il file di configurazione JSON:

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

Mostra output di esempio

```
traddr=192.168.38.10is already connected
traddr=192.168.39.10 is already connected
traddr=192.168.38.11 is already connected
traddr=192.168.39.11 is already connected
traddr=192.168.38.10is already connected
traddr=192.168.39.10 is already connected
traddr=192.168.38.11 is already connected
traddr=192.168.39.11 is already connected
traddr=192.168.38.10is already connected
traddr=192.168.39.10 is already connected
traddr=192.168.38.11 is already connected
traddr=192.168.39.11 is already connected
```

3. Verificare che i segreti dhchap siano stati abilitati per i rispettivi controller per ciascun sottosistema:

   a. Verificare le chiavi dhchap dell'host:

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

   L'esempio seguente mostra una chiave dhchap:

```
DHHC-1:01:wkwAKk8r9Ip7qECKt7V5aIo/7Y1CH7DWkUfLfMxmseg39DFb:
```

b. Verificare i tasti dhchap del controller:

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

Dovresti vedere un output simile al seguente esempio:

```
DHHC-
1:03:ohdxI1yIS8gBLwIOubcwl57rXcozYuRgBsoWaBvxEvpDlQHn/7dQ4JjFGwmhgwd
JWmVoripbWbMJy5eMAbCahN4hhYU=:
```

Passaggio 10: configurare la sicurezza del livello di trasporto

Transport Layer Security (TLS) fornisce una crittografia end-to-end sicura per le connessioni NVMe tra host NVMe-oF e un array ONTAP . È possibile configurare TLS 1.3 utilizzando la CLI e una chiave pre-condivisa (PSK) configurata.

Eseguire i seguenti passaggi sull'host SUSE Linux Enterprise Server, tranne nei casi in cui è specificato che si deve eseguire un passaggio sul controller ONTAP .

Fasi

1. Verifica di avere quanto segue `ktls-utils` , `openssl` , E `libopenssl` pacchetti installati sull'host:

a. Verificare il `ktls-utils` :

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

Dovresti vedere visualizzato il seguente output:

```
ktls-utils-0.10+33.g311d943-160000.2.2.x86_64
```

a. Verificare i pacchetti SSL:

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

Mostra output di esempio

```
libopenssl3-3.5.0-160000.3.2.x86_64
openssl-3.5.0-160000.2.2.noarch
openssl-3-3.5.0-160000.3.2.x86_64
libopenssl3-x86-64-v3-3.5.0-160000.3.2.x86_64
```

2. Verificare di disporre della configurazione corretta per `/etc/tlshd.conf`:

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

Mostra output di esempio

```
[debug]
loglevel=0
tls=0
nl=0

[authenticate]
#keyrings= <keyring>;<keyring>;<keyring>

[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. Abilitare `tlshd` per l'avvio all'avvio del sistema:

```
systemctl enable tlshd
```

4. Verificare che il `tlshd` daemon sia in esecuzione:

```
systemctl status tlshd
```

Mostra output di esempio

```
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. Generare TLS PSK utilizzando `nvme gen-tls-key` :

   a. Verifica l'host:

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

   Viene visualizzato il seguente output:

   ```
   nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b7c04f444d33
   ```

   b. Verifica la chiave:

   ```
   nvme gen-tls-key --hmac=1 --identity=1 --subsysnqn= nqn.1992-
   08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme1
   ```

   Viene visualizzato il seguente output:

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

6. Sul controller ONTAP, aggiungere il PSK TLS al sottosistema ONTAP:

Mostra output di esempio

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

7. Inserire TLS PSK nel keyring del kernel host:

```
nvme check-tls-key --identity=1 --subsysnqn=nqn.1992
-08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme1
--keydata=NVMeTLSkey
-1:01:C50EsaGtuOp8n5fGE9EuWjbBCtshmfoHx4XTqTJUmydf0gIj: --insert
```

Dovresti vedere la seguente chiave TLS:

```
Inserted TLS key 069f56bb
```

Il PSK mostra come `NVMe1R01` perché usa `identity v1` dall'algoritmo di handshake TLS. Identity v1 è l'unica versione supportata da ONTAP.

8. Verificare che TLS PSK sia inserito correttamente:

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

Mostra output di esempio

```
069f56bb I-Q-- 5 perm 3b010000 0 0 psk NVMe1R01 nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme1
oYVLelmiOwnvDjXKBmrnIgGVpFIBDJtc4hmQXE/36Sw=: 32
```

9. Connettersi al sottosistema ONTAP utilizzando il PSK TLS inserito:

   a. Verificare il TLS PSK:

```
nvme connect -t tcp -w 192.168.38.20 -a 192.168.38.10  -n nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme1
--tls_key=0x069f56bb –tls
```

Viene visualizzato il seguente output:

```
connecting to device: nvme0
```

a. Verificare list-subsys:

```
nvme list-subsys
```

Mostra output di esempio

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-
0035-5910-804b-b2c04f444d33
**
 +- nvme0 tcp
traddr=192.168.38.10,trsvcid=4420,host_traddr=192.168.38.20,src_a
ddr=192.168.38.20 live
```

10. Aggiungere la destinazione e verificare la connessione TLS al sottosistema ONTAP specificato:

```
nvme subsystem controller show -vserver vs_tcp_sles16 -subsystem nvme1
-instance
```

Mostra output di esempio

```
(vserver nvme subsystem controller show)
                      Vserver Name: vs_tcp_sles16
                         Subsystem: nvme1
                     Controller ID: 0040h
                 Logical Interface: lif1
                              Node: A400-12-171
                          Host NQN: nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
                Transport Protocol: nvme-tcp
       Initiator Transport Address: 192.168.38.20
                    Host Identifier:
4c4c454400355910804bb2c04f444d33
              Number of I/O Queues: 2
                  I/O Queue Depths: 128, 128
                 Admin Queue Depth: 32
             Max I/O Size in Bytes: 1048576
         Keep-Alive Timeout (msec): 5000
                    Subsystem UUID: 62203cfd-826a-11f0-966e-
d039eab31e9d
             Header Digest Enabled: false
               Data Digest Enabled: false
      Authentication Hash Function: sha-256
Authentication Diffie-Hellman Group: 3072-bit
               Authentication Mode: unidirectional
      Transport Service Identifier: 4420
                     TLS Key Type: configured
                TLS PSK Identity: NVMe1R01 nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme1
oYVLelmiOwnvDjXKBmrnIgGVpFIBDJtc4hmQXE/36Sw=
                       TLS Cipher: TLS-AES-128-GCM-SHA256
```

Fase 11: Esaminare i problemi noti

Non ci sono problemi noti.

Configurare SUSE Linux Enterprise Server 15 SPx per NVMe-oF con storage ONTAP

L'host SUSE Linux Enterprise Server 15 SPx supporta i protocolli NVMe su Fibre Channel (NVMe/FC) e NVMe su TCP (NVMe/TCP) con Asymmetric Namespace Access

(ANA). ANA fornisce funzionalità multipathing equivalenti all'accesso asimmetrico alle unità logiche (ALUA) negli ambienti iSCSI e FCP.

Scopri come configurare gli host NVMe over Fabrics (NVMe-oF) per SUSE Linux Enterprise Server 15 SPx. Per ulteriori informazioni sul supporto e sulle funzionalità, vedere "Supporto e funzionalità ONTAP".

NVMe-oF con SUSE Linux Enterprise Server 15 SPx presenta le seguenti limitazioni note:

- IL `nvme disconnect-all` Il comando disconnette sia il file system root che quello dati e potrebbe causare instabilità del sistema. Non eseguire questa operazione su sistemi che si avviano da SAN tramite namespace NVMe-TCP o NVMe-FC.
- Il supporto dell'utilità host NetApp sanlun non è disponibile per NVMe-oF. In alternativa, puoi affidarti al plug-in NetApp incluso nel nativo `nvme-cli` pacchetto per tutti i trasporti NVMe-oF.
- Per SUSE Linux Enterprise Server 15 SP6 e versioni precedenti, l'avvio SAN tramite il protocollo NVMe-oF non è supportato.

Passaggio 1: Se lo si desidera, attivare l'avvio SAN

È possibile configurare l'host per utilizzare l'avvio SAN per semplificare la distribuzione e migliorare la scalabilità. Utilizzare il"Tool di matrice di interoperabilità" per verificare che il sistema operativo Linux, l'adattatore bus host (HBA), il firmware HBA, il BIOS di avvio HBA e la versione ONTAP supportino l'avvio SAN.

Fasi

1. "Crea uno spazio dei nomi NVMe e mappalo all'host" .
2. Abilitare l'avvio SAN nel BIOS del server per le porte su cui è mappato lo spazio dei nomi di avvio SAN.

   Per informazioni su come attivare il BIOS HBA, consultare la documentazione specifica del vendor.

3. Riavviare l'host e verificare che il sistema operativo sia attivo e funzionante.

Passaggio 2: installare SUSE Linux Enterprise Server e il software NVMe e verificare la configurazione

Per configurare l'host per NVMe-oF è necessario installare i pacchetti software host e NVMe, abilitare il multipathing e verificare la configurazione NQN dell'host.

Fasi

1. Installare SUSE Linux Enterprise Server 15 SPx sul server. Una volta completata l'installazione, verificare di eseguire il kernel SUSE Linux Enterprise Server 15 SPx specificato:

   ```
   uname -r
   ```

   Esempio di versione del kernel Rocky Linux:

   ```
   6.4.0-150700.53.3-default
   ```

2. Installare `nvme-cli` pacchetto:

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

L'esempio seguente mostra un `nvme-cli` versione del pacchetto:

```
nvme-cli-2.11+22.gd31b1a01-150700.3.3.2.x86_64
```

3. Installare `libnvme` pacchetto:

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

L'esempio seguente mostra un `libnvme` versione del pacchetto:

```
libnvme1-1.11+4.ge68a91ae-150700.4.3.2.x86_64
```

4. Sull'host, controlla la stringa hostnqn in `/etc/nvme/hostnqn` :

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

L'esempio seguente mostra un `hostnqn` versione:

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

5. Sul sistema ONTAP , verificare che `hostnqn` la stringa corrisponde a `hostnqn` stringa per il sottosistema corrispondente nell'array ONTAP :

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

Mostra esempio

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

Se il `hostnqn` le stringhe non corrispondono, utilizzare `vserver modify` per aggiornare `hostnqn` Stringa sul sottosistema di array ONTAP corrispondente a `hostnqn` stringa da `/etc/nvme/hostnqn` sull'host.

Passaggio 3: configurare NVMe/FC e NVMe/TCP

Configurare NVMe/FC con adattatori Broadcom/Emulex o Marvell/QLogic oppure configurare NVMe/TCP utilizzando operazioni di rilevamento e connessione manuali.

NVMe/FC - Broadcom/Emulex

Configurare NVMe/FC per un adattatore FC Broadcom/Emulex.

Fasi

1. Verificare di utilizzare il modello di adattatore supportato:

   a. Visualizza i nomi dei modelli:

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

   Viene visualizzato il seguente output:

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

   b. Visualizza le descrizioni dei modelli:

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

   Viene visualizzato il seguente output:

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

2. Verificare di utilizzare il Broadcom consigliato `lpfc` firmware e driver della posta in arrivo:

   a. Visualizza la versione del firmware:

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

   L'esempio seguente mostra le versioni del firmware:

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

   b. Visualizza la versione del driver in arrivo:

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

   L'esempio seguente mostra la versione del driver:

```
0:14.4.0.8
```

Per l'elenco aggiornato dei driver della scheda di rete supportati e delle versioni del firmware, vedere "Tool di matrice di interoperabilità".

3. Verificare che l'output previsto di `lpfc_enable_fc4_type` sia impostato su `3`:

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

4. Verificare che sia possibile visualizzare le porte dell'iniziatore:

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

Dovresti vedere un output simile a:

```
0x10000090fae0ec88
0x10000090fae0ec89
```

5. Verificare che le porte dell'iniziatore siano in linea:

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

Viene visualizzato il seguente output:

```
Online
Online
```

6. Verificare che le porte iniziatore NVMe/FC siano abilitate e che le porte di destinazione siano visibili:

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

Mostra output di esempio

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

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

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

NVME Statistics
```

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

NVMe/FC - Marvell/QLogic

Configurare NVMe/FC per un adattatore Marvell/QLogic.

Fasi

1. Verificare che siano in esecuzione le versioni del firmware e del driver dell'adattatore supportate:

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

   L'esempio seguente mostra le versioni del driver e del firmware:

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

2. Verificare che `ql2xnvmeenable` è impostato. Ciò consente all'adattatore Marvell di funzionare come iniziatore NVMe/FC:

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

   L'uscita prevista è 1.

NVMe/TCP

Il protocollo NVMe/TCP non supporta l'operazione di connessione automatica. In alternativa, è possibile scoprire i sottosistemi e gli spazi dei nomi NVMe/TCP eseguendo l'NVMe/TCP `connect` O `connect-all` operazioni manualmente.

Fasi

1. Verificare che la porta iniziatore possa recuperare i dati della pagina del registro di rilevamento attraverso le LIF NVMe/TCP supportate:

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

Mostra output di esempio

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

Discovery Log Number of Records 8, Generation counter 42
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr:  192.168.211.71
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  3
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr:  192.168.111.71
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr:  192.168.211.70
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
```

```
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr:  192.168.111.70
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.samp
le_tcp_sub
traddr:  192.168.211.71
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  3
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.samp
le_tcp_sub
traddr:  192.168.111.71
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.samp
```

```
le_tcp_sub
traddr:  192.168.211.70
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.samp
le_tcp_sub
traddr:  192.168.111.70
eflags:  none
sectype: none
localhost:~ #
```

2. Verifica che tutte le altre combinazioni di LIF iniziatore NVMe/TCP siano in grado di recuperare con successo i dati della pagina del log di rilevamento:

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

Mostra esempio

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

3. Eseguire `nvme connect-all` Command tra tutti i LIF target initiator NVMe/TCP supportati nei nodi:

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

Mostra esempio

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

A partire da SUSE Linux Enterprise Server 15 SP6, l'impostazione per NVMe/TCP `ctrl_loss_tmo timeout` viene automaticamente impostato su "off". Di conseguenza:

- Non ci sono limiti al numero di tentativi (tentativi illimitati).
- Non è necessario configurare manualmente uno specifico `ctrl_loss_tmo timeout` durata quando si utilizza il `nvme connect` O `nvme connect-all` comandi (opzione -l).
- I controller NVMe/TCP non subiscono timeout in caso di errore del percorso e rimangono connessi indefinitamente.

Passaggio 4: Facoltativamente, modificare iopolicy nelle regole udev

A partire da SUSE Linux Enterprise Server 15 SP6, l'iopolicy predefinito per NVMe-oF è impostato su `round-robin`. Se vuoi cambiare iopolicy in `queue-depth`, modificare il file delle regole udev come segue:

Fasi

1. Aprire il file delle regole udev in un editor di testo con privilegi di root:

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

   Viene visualizzato il seguente output:

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

2. Trova la riga che imposta iopolicy per NetApp ONTAP Controller, come mostrato nella seguente regola di esempio:

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

3. Modificare la regola in modo che `round-robin` diventa `queue-depth` :

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

4. Ricarica le regole udev e applica le modifiche:

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

5. Verificare l'attuale iopolicy per il sottosistema. Sostituisci <sottosistema>, ad esempio, `nvme-subsys0` .

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

Viene visualizzato il seguente output:

```
queue-depth.
```

> La nuova iopolicy si applica automaticamente ai dispositivi NetApp ONTAP Controller corrispondenti. Non è necessario riavviare.

Passaggio 5: Facoltativamente, abilitare 1 MB di I/O per NVMe/FC

ONTAP segnala una dimensione massima di trasferimento dati (MDTS) pari a 8 nei dati Identify Controller. Ciò significa che la dimensione massima della richiesta di I/O può arrivare fino a 1 MB. Per emettere richieste di I/O di dimensione 1 MB per un host Broadcom NVMe/FC, è necessario aumentare il `lpfc` valore del `lpfc_sg_seg_cnt` parametro a 256 dal valore predefinito di 64.

> Questi passaggi non si applicano agli host Qlogic NVMe/FC.

Fasi

1. Impostare il `lpfc_sg_seg_cnt` parametro su 256:

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

Dovresti vedere un output simile al seguente esempio:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Eseguire il `dracut -f` comando e riavviare l'host.

3. Verificare che il valore per `lpfc_sg_seg_cnt` sia 256:

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

Passaggio 6: verificare i servizi di avvio NVMe

IL `nvmefc-boot-connections.service` E `nvmf-autoconnect.service` servizi di avvio inclusi in NVMe/FC `nvme-cli` i pacchetti vengono abilitati automaticamente all'avvio del sistema.

Dopo aver completato l'avvio, verificare che `nvmefc-boot-connections.service` E `nvmf-autoconnect.service` i servizi di avvio sono abilitati.

Fasi

1. Verificare che `nvmf-autoconnect.service` sia attivato:

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

Mostra output di esempio

```
nvmf-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
  Loaded: loaded (/usr/lib/systemd/system/nvmf-autoconnect.service;
enabled; preset: enabled)
  Active: inactive (dead) since Fri 2025-07-04 23:56:38 IST; 4 days
ago
  Main PID: 12208 (code=exited, status=0/SUCCESS)
    CPU: 62ms

Jul 04 23:56:26 localhost systemd[1]: Starting Connect NVMe-oF
subsystems automatically during boot...
Jul 04 23:56:38 localhost systemd[1]: nvmf-autoconnect.service:
Deactivated successfully.
Jul 04 23:56:38 localhost systemd[1]: Finished Connect NVMe-oF
subsystems automatically during boot.
```

2. Verificare che `nvmefc-boot-connections.service` sia attivato:

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

Mostra output di esempio

```
nvmefc-boot-connections.service - Auto-connect to subsystems on FC-
NVME devices found during boot
    Loaded: loaded (/usr/lib/systemd/system/nvmefc-boot-
connections.service; enabled; preset: enabled)
    Active: inactive (dead) since Mon 2025-07-07 19:52:30 IST; 1 day
4h ago
  Main PID: 2945 (code=exited, status=0/SUCCESS)
       CPU: 14ms

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

Passaggio 7: verificare la configurazione del multipathing

Verificare che lo stato multipath NVMe in-kernel, lo stato ANA e i namespace ONTAP siano corretti per la configurazione NVMe-of.

Fasi

1. Verificare che il multipath NVMe nel kernel sia attivato:

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

   Viene visualizzato il seguente output:

   ```
   Y
   ```

2. Verificare che le impostazioni NVMe-oF appropriate (ad esempio, modello impostato su NetApp ONTAP Controller e iopolicy di bilanciamento del carico impostato su queue-depth) per i rispettivi namespace ONTAP si riflettano correttamente sull'host:

   a. Visualizza i sottosistemi:

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

      Viene visualizzato il seguente output:

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

b. Visualizza la politica:

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

Viene visualizzato il seguente output:

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

3. Verificare che gli spazi dei nomi siano stati creati e rilevati correttamente sull'host:

```
nvme list
```

Mostra esempio

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


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

4. Verificare che lo stato del controller di ciascun percorso sia attivo e che abbia lo stato ANA corretto:

NVMe/FC

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

Mostra output di esempio

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

NVMe/TCP

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

Mostra output di esempio

```
nvme-subsys9 - NQN=nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.with
_inband_with_json hostnqn=nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
iopolicy=round-robin
\
+- nvme10 tcp
traddr=192.168.111.71,trsvcid=4420,src_addr=192.168.111.80 live
non-optimized
 +- nvme11 tcp
traddr=192.168.211.70,trsvcid=4420,src_addr=192.168.211.80 live
optimized
 +- nvme12 tcp
traddr=192.168.111.70,trsvcid=4420,src_addr=192.168.111.80 live
optimized
 +- nvme9 tcp
traddr=192.168.211.71,trsvcid=4420,src_addr=192.168.211.80 live
non-optimized
```

5. Verificare che il plug-in NetApp visualizzi i valori corretti per ciascun dispositivo dello spazio dei nomi ONTAP:

Colonna

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

Mostra esempio

```
Device           Vserver                   Namespace Path
NSID UUID                                   Size
---------------- ------------------------
------------------------------------------------ ----
---------------------------------- ---------
/dev/nvme0n1     vs_161
/vol/fc_nvme_vol1/fc_nvme_ns1                       1
32fd92c7-0797-428e-a577-fdb3f14d0dc3   5.37GB
```

JSON

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

Mostra esempio

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

Passaggio 8: creare un controller di scoperta persistente

È possibile creare un controller di rilevamento persistente (PDC) per un host SUSE Linux Enterprise Server 15 SPx. È necessario un PDC per rilevare automaticamente un'operazione di aggiunta o rimozione del sottosistema NVMe e le modifiche ai dati della pagina del registro di individuazione.

Fasi

1. Verificare che i dati della pagina del log di rilevamento siano disponibili e possano essere recuperati attraverso la combinazione di porta Initiator e LIF di destinazione:

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

Mostra output di esempio

```
Discovery Log Number of Records 8, Generation counter 18
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:  192.168.111.66
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:  192.168.211.66
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  3
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:  192.168.111.67
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
```

```
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:  192.168.211.67
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:subsystem.pdc
traddr:  192.168.111.66
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:subsystem.pdc
traddr:  192.168.211.66
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  3
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:subsystem.pdc
traddr:  192.168.111.67
eflags:  none
sectype: none
=====Discovery Log Entry 7======
```

```
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:subsystem.pdc
traddr:  192.168.211.67
eflags:  none
sectype: none
```

2. Creare un PDC per il sottosistema di rilevamento:

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

Viene visualizzato il seguente output:

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

3. Dal controller ONTAP, verificare che il PDC sia stato creato:

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

Mostra output di esempio

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

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

Passaggio 9: impostare l’autenticazione in-band sicura

L’autenticazione in-band sicura è supportata tramite NVMe/TCP tra un host SUSE Linux Enterprise Server 15 SPx e un controller ONTAP .

Ogni host o controller deve essere associato a un `DH-HMAC-CHAP` chiave per impostare l’autenticazione sicura. Una chiave DH-HMAC-CHAP è una combinazione dell’NQN dell’host o del controller NVMe e di un segreto di autenticazione configurato dall’amministratore. Per autenticare il suo peer, un host o un controller NVMe deve riconoscere la chiave associata al peer.

Fasi

Imposta l’autenticazione in-band sicura tramite la CLI o un file JSON di configurazione. Se è necessario specificare chiavi dhchap diverse per sottosistemi diversi, è necessario utilizzare un file di configurazione JSON.

CLI

Configurare l'autenticazione in banda protetta utilizzando la CLI.

1. Ottenere l'NQN dell'host:

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

2. Genera la chiave dhchap per l'host.

   L'output seguente descrive i `gen-dhchap-key` parametri dei comandi:

```
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
```

   Nell'esempio seguente, viene generata una chiave casuale dhCHAP con HMAC impostato su 3 (SHA-512).

```
nvme gen-dhchap-key -m 3 -n nqn.2014-
08.org.nvmexpress:uuid:e6dade64-216d-11ec-b7bb-7ed30a5482c3
DHHC-
1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUAg1iGgx
TYqnxukqvYedA55Bw3wtz6sJNpR4=:
```

3. Sul controller ONTAP, aggiungere l'host e specificare entrambe le chiavi dhchap:

```
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. Un host supporta due tipi di metodi di autenticazione, unidirezionale e bidirezionale. Sull'host, connettersi al controller ONTAP e specificare le chiavi dhchap in base al metodo di autenticazione scelto:

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

5. Convalidare `nvme connect authentication` comando verificando le chiavi dhchap dell'host e del controller:
   a. Verificare le chiavi dhchap dell'host:

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

   Mostra output di esempio per una configurazione unidirezionale

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys1/nvme*/dhchap_secret
   DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
   DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
   DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
   DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
   ```

   b. Verificare i tasti dhchap del controller:

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

   Mostra output di esempio per una configurazione bidirezionale

   ```
   cat /sys/class/nvme-subsystem/nvme-
   subsys6/nvme*/dhchap_ctrl_secret
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   ```

JSON

Quando sulla configurazione del controller ONTAP sono disponibili più sottosistemi NVMe, è possibile utilizzare il `/etc/nvme/config.json` file con il `nvme connect-all` comando.

Utilizzare il `-o` opzione per generare il file JSON. Per ulteriori opzioni di sintassi, fare riferimento alle pagine man di NVMe connect-all.

1. Configurare il file JSON:

Mostra output di esempio

```
cat /etc/nvme/config.json
[
 {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-
5910-804b-b2c04f444d33",
    "hostid":"4c4c4544-0035-5910-804b-b2c04f444d33",
    "dhchap_key":"DHHC-
1:01:i4i789R11sMuHLCY27RVI8XloC\/GzjRwyhxip5hmIELsHrBq:",
    "subsystems":[
      {
        "nqn":"nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.samp
le_tcp_sub",
        "ports":[
          {
            "transport":"tcp",
            "traddr":"192.168.111.70",
            "host_traddr":"192.168.111.80",
            "trsvcid":"4420"
            "dhchap_ctrl_key":"DHHC-
1:03:jqgYcJSKp73+XqAf2X6twr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvh
YS0PNK7T+04YD5CRPjh+m3qjJU++yR8s=:"
          },
               {
                    "transport":"tcp",
                    "traddr":"192.168.111.71",
                    "host_traddr":"192.168.111.80",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:03:jqgYcJSKp73+XqAf2X6twr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvh
YS0PNK7T+04YD5CRPjh+m3qjJU++yR8s=:"
               },
               {
                    "transport":"tcp",
                    "traddr":"192.168.211.70",
                    "host_traddr":"192.168.211.80",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:03:jqgYcJSKp73+XqAf2X6twr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvh
YS0PNK7T+04YD5CRPjh+m3qjJU++yR8s=:"
               },
               {
                    "transport":"tcp",
                    "traddr":"192.168.211.71",
```

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

Nell'esempio seguente, `dhchap_key` corrisponde a `dhchap_secret` E `dhchap_ctrl_key` corrisponde a `dhchap_ctrl_secret` .

2. Connettersi al controller ONTAP utilizzando il file di configurazione JSON:

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

Mostra output di esempio

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

3. Verificare che i segreti dhchap siano stati abilitati per i rispettivi controller per ciascun sottosistema:

   a. Verificare le chiavi dhchap dell'host:

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

   L'esempio seguente mostra una chiave dhchap:

```
DHHC-1:01:i4i789R11sMuHLCY27RVI8XloC/GzjRwyhxip5hmIELsHrBq:
```

b. Verificare i tasti dhchap del controller:

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

Dovresti vedere un output simile al seguente esempio:

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

Passaggio 10: configurare la sicurezza del livello di trasporto

Transport Layer Security (TLS) fornisce una crittografia end-to-end sicura per le connessioni NVMe tra host NVMe-oF e un array ONTAP . È possibile configurare TLS 1.3 utilizzando la CLI e una chiave pre-condivisa (PSK) configurata.

Eseguire i seguenti passaggi sull'host SUSE Linux Enterprise Server, tranne nei casi in cui è specificato che si deve eseguire un passaggio sul controller ONTAP .

Fasi

1. Verifica di avere quanto segue `ktls-utils` , `openssl` , E `libopenssl` pacchetti installati sull'host:

a. Verificare il `ktls-utils` :

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

Dovresti vedere visualizzato il seguente output:

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

a. Verificare i pacchetti SSL:

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

Mostra output di esempio

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

2. Verificare di disporre della configurazione corretta per `/etc/tlshd.conf`:

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

Mostra output di esempio

```
[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. Abilitare `tlshd` per l'avvio all'avvio del sistema:

```
systemctl enable tlshd
```

4. Verificare che il `tlshd` daemon sia in esecuzione:

```
systemctl status tlshd
```

Mostra output di esempio

```
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. Generare TLS PSK utilizzando `nvme gen-tls-key` :

   a. Verifica l'host:

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

   Viene visualizzato il seguente output:

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

   b. Verifica la chiave:

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

   Viene visualizzato il seguente output:

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

6. Sul controller ONTAP, aggiungere il PSK TLS al sottosistema ONTAP:

Mostra output di esempio

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

7. Inserire TLS PSK nel keyring del kernel host:

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

Dovresti vedere la seguente chiave TLS:

```
Inserted TLS key 22152a7e
```

> Il PSK mostra come `NVMe1R01` perché usa `identity v1` dall'algoritmo di handshake TLS. Identity v1 è l'unica versione supportata da ONTAP.

8. Verificare che TLS PSK sia inserito correttamente:

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

Mostra output di esempio

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

9. Connettersi al sottosistema ONTAP utilizzando il PSK TLS inserito:

   a. Verificare il TLS PSK:

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

Viene visualizzato il seguente output:

```
connecting to device: nvme0
```

a. Verificare list-subsys:

```
nvme list-subsys
```

Mostra output di esempio

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

10. Aggiungere la destinazione e verificare la connessione TLS al sottosistema ONTAP specificato:

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

Mostra output di esempio

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

Fase 11: Esaminare i problemi noti

Non ci sono problemi noti.

Informazioni sul copyright

Informazioni sul marchio commerciale