



SLES

SAN hosts and cloud clients

NetApp
March 29, 2024

Table of Contents

SLES	1
SLES 15	1

SLES

SLES 15

NVMe-oF host configuration for SUSE Linux Enterprise Server 15 SP5 with ONTAP

NVMe over Fabrics (NVMe-oF), including NVMe over Fibre Channel (NVMe/FC) and other transports, is supported with SUSE Linux Enterprise Server (SLES) 15 SP5 with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is the equivalent of ALUA multipathing in iSCSI and FCP environments and is implemented with in-kernel NVMe multipath.

The following support is available for the NVMe-oF host configuration for SLES 15 SP5 with ONTAP:

- Both NVMe and SCSI traffic can be run on the same co-existent host. Therefore, for SCSI LUNs, you can configure dm-multipath for SCSI mpath devices, whereas you might use NVMe multipath to configure NVMe-oF namespace devices on the host.
- Support for NVMe over TCP (NVMe/TCP) in addition to NVMe/FC. The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.

For additional details on supported configurations, see the [NetApp Interoperability Matrix Tool](#).

Features

- Support for NVMe secure, in-band authentication
- Support for persistent discovery controllers (PDCs) using a unique discovery NQN

Known limitations

- SAN booting using the NVMe-oF protocol is currently not supported.
- There's no `sanlun` support for NVMe-oF. Therefore, the host utility support isn't available for NVMe-oF on an SLES 15 SP5 host. You can rely on the NetApp plug-in included in the native `nvme-cli` package for all NVMe-oF transports.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex FC or Marvell/Qlogic FC adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the recommended adapter model:

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

Example output:

```
LPe32002 M2  
LPe32002-M2
```

2. Verify the adapter model description:

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

Example output:

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

3. Verify that you are using the recommended Emulex host bus adapter (HBA) firmware versions:

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

Example output:

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

4. Verify that you are using the recommended LPFC driver version:

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

Example output:

```
0:14.2.0.13
```

5. Verify that you can view your initiator ports:

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

Example output:

```
0x100000109b579d5e  
0x100000109b579d5f
```

6. Verify that your initiator ports are online:

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

Example output:

```
Online  
Online
```

7. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

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

Example output:

In this example, one initiator port is enabled and connected with two target LIFs.

```

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 DISCSRV ONLINE
NVME RPORT WWPN x208500a098dfdd91 WWNN x208100a098dfdd91 DID x010003
TARGET DISCSRV *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
ffffffffffffb046f
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 DISCSRV ONLINE
NVME RPORT WWPN x208200a098dfdd91 WWNN x208100a098dfdd91 DID x012a03
TARGET DISCSRV 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
fffffffffffffaf794
abort 00000b73 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000159d Err 000135c3

```

8. Reboot the host.

Marvell/QLogic

Steps

1. The native inbox qla2xxx driver included in the SLES 15 SP5 kernel has the latest fixes essential for ONTAP support. Verify that you are running the supported adapter driver and firmware versions:

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

Example output:

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

2. Verify that the `ql2xnvmeenable` parameter is set to 1:

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

Enable 1MB I/O size (Optional)

ONTAP reports an MDTs (Max Data Transfer Size) of 8 in the Identify Controller data which means the maximum I/O request size can be up to 1MB. However, to issue I/O requests of size 1 MB for a Broadcom NVMe/FC host, you must increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256.

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

2. Run a `dracut -f` command, and reboot the host.
3. Verify that `lpfc_sg_seg_cnt` is 256.

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



This is not applicable to Qlogic NVMe/FC hosts.

Enable NVMe services

There are two NVMe/FC boot services included in the `nvme-cli` package, however, *only* `nvme-fc-boot-connections.service` is enabled to start during system boot; `nvme-fc-autoconnect.service` is not enabled. Therefore, you need to manually enable `nvme-fc-autoconnect.service` to start during system boot.

Steps

1. Enable `nvme-fc-autoconnect.service`:

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

2. Reboot the host.
3. Verify that `nvme-autoconnect.service` and `nvme-fc-boot-connections.service` are running after the system boots up:

Example output:

```
# systemctl status nvme-autoconnect.service
nvme-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
Loaded: loaded (/usr/lib/systemd/system/nvme-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]: nvme-autoconnect.service: Deactivated successfully.
systemd[1]: Finished Connect NVMe-oF subsystems automatically during
boot.

# systemctl status nvme-fc-boot-connections.service
nvme-fc-boot-connections.service - Auto-connect to subsystems on FC-NVME
devices found during boot
Loaded: loaded (/usr/lib/systemd/system/nvme-fc-boot-
connections.service; enabled; 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]: nvme-fc-boot-connections.service: Succeeded.
systemd[1]: Finished Auto-connect to subsystems on FC-NVME devices found
during boot.
```


Configure NVMe/TCP

You can use the following procedure to configure NVMe/TCP.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

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

Example output:

```
# 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. Verify that all other NVMe/TCP initiator-target LIF combinations can successfully fetch discovery log page data:

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

Example output:

```

# 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. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

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

Example output:

```
# 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 recommends setting the `ctrl-loss-tmo` option to `-1` so that the NVMe/TCP initiator attempts to reconnect indefinitely in the event of a path loss.

Validate NVMe-oF

You can use the following procedure to validate NVMe-oF.

Steps

1. Verify that in-kernel NVMe multipath is enabled:

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

2. Verify that the host has the correct controller model for the ONTAP NVMe namespaces:

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

Example output:

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

3. Verify the NVMe I/O policy for the respective ONTAP NVMe I/O controller:

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

Example output:

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

4. Verify that the ONTAP namespaces are visible to the host:

```
nvme list -v
```

Example output:

```
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 Address          Subsystem    Namespaces
-----
-----
nvme0      81LGgBUqsI3EAAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.2.214,trsvcid=4420,host_traddr=192.168.2.14 nvme-subsys0
nvme0n1
nvme1      81LGgBUqsI3EAAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.2.215,trsvcid=4420,host_traddr=192.168.2.14 nvme-subsys0
nvme0n1
nvme2      81LGgBUqsI3EAAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.1.214,trsvcid=4420,host_traddr=192.168.1.14 nvme-subsys0
nvme0n1
nvme3      81LGgBUqsI3EAAAAAAAAE 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. Verify that the controller state of each path is live and has the correct ANA status:

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

NVMe/FC

Example output

```
# 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

Example output

```
# 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. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

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

Example output:

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
```

Example output:

```
{
  "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
    }
  ]
}
```

Create a persistent discovery controller

Beginning with ONTAP 9.11.1, you can create a persistent discovery controller (PDC) for your SLES 15 SP5 host by using the following procedure. A PDC is required to automatically detect NVMe subsystem add or remove scenarios and changes to the discovery log page data.

Steps

1. Verify that the discovery log page data is available and can be retrieved through the initiator port and target LIF combination:

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


Example output:

```
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.0501daf15dda11eeab68d039eea7a232: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.0501daf15dda11eeab68d039eea7a232:subsystem.unidir_d
hchap
traddr:  192.168.2.214
eflags:  none
sectype: none

```

2. Create a PDC for the discovery subsystem:

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

Example output:

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

3. From the ONTAP controller, verify that the PDC has been created:

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

Example output:

```
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-
74362c0cla1fc
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
```

Set up secure in-band authentication

Beginning with ONTAP 9.12.1, secure in-band authentication is supported over NVMe/TCP and NVMe/FC between your SLES 15 SP5 host and your ONTAP controller.

To set up secure authentication, each host or controller must be associated with a `DH-HMAC-CHAP` key, which is a combination of the NQN of the NVMe host or controller and an authentication secret configured by the administrator. To authenticate its peer, an NVMe host or controller must recognize the key associated with the peer.

You can set up secure in-band authentication using the CLI or a config JSON file. If you need to specify different dhchap keys for different subsystems, you must use a config JSON file.

CLI

Steps

1. Obtain the host NQN:

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

2. Generate the dhchap key for the SLES15 SP5 host:

```
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

In the following example, a random dhchap key with HMAC set to 3 (SHA-512) is generated.

```
# 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. On the ONTAP controller, add the host and specify both dhchap keys:

```
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. A host supports two types of authentication methods, unidirectional and bidirectional. On the host, connect to the ONTAP controller and specify dhchap keys based on the chosen authentication method:

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

5. Validate the `nvme connect authentication` command by verifying the host and controller `dhchap` keys:

a. Verify the host `dhchap` keys:

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

Example output for unidirectional configuration:

```
# cat /sys/class/nvme-subsystem/nvme-subsys1/nvme*/dhchap_secret
DHC-
1:03:je1nQCmjJLUKD62mpYbz1puw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
DHC-
1:03:je1nQCmjJLUKD62mpYbz1puw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
DHC-
1:03:je1nQCmjJLUKD62mpYbz1puw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
DHC-
1:03:je1nQCmjJLUKD62mpYbz1puw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
```

b. Verify the controller `dhchap` keys:

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

Example output for bidirectional configuration:


```
# cat /sys/class/nvme-subsystem/nvme-
subsys6/nvme*/dhchap_ctrl_secret
DHHC-
1:03:WorVEV83eY053kV4Iel5OpphbX5LAph03F8fgH3913tlrkSGDBJTt3crXeTU
B8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eY053kV4Iel5OpphbX5LAph03F8fgH3913tlrkSGDBJTt3crXeTU
B8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eY053kV4Iel5OpphbX5LAph03F8fgH3913tlrkSGDBJTt3crXeTU
B8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eY053kV4Iel5OpphbX5LAph03F8fgH3913tlrkSGDBJTt3crXeTU
B8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
```

JSON file

You can use the `/etc/nvme/config.json` file with the `nvme connect-all` command when multiple NVMe subsystems are available on the ONTAP controller configuration.

You can generate the JSON file using `-o` option. Refer to the NVMe connect-all man pages for more syntax options.

Steps

1. Configure the JSON file:

```
# 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
ptWpnalrpwG5CndpOgxprXh9m4lw=: "
  },
  {
    "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. Connect to the ONTAP controller using the config JSON file:

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

Example output:

```
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. Verify that the dhchap secrets have been enabled for the respective controllers for each subsystem:

a. Verify the host dhchap keys:

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

Example output:

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

b. Verify the controller dhchap keys:

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

Example output:

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

Known issues

There are no known issues for the SLES 15 SP5 with ONTAP release.

NVMe-oF host configuration for SUSE Linux Enterprise Server 15 SP4 with ONTAP

NVMe over Fabrics (NVMe-oF), including NVMe over Fibre Channel (NVMe/FC) and other transports, is supported with SUSE Linux Enterprise Server (SLES) 15 SP4 with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is the equivalent of ALUA multipathing in iSCSI and FCP environments and is implemented with in-kernel NVMe multipath.

The following support is available for the NVMe-oF host configuration for SLES 15 SP4 with ONTAP:

- Both NVMe and SCSI traffic can be run on the same co-existent host. Therefore, for SCSI LUNs, you can configure dm-multipath for SCSI mpath devices, whereas you might use NVMe multipath to configure NVMe-oF namespace devices on the host.
- Support for NVMe over TCP (NVMe/TCP) in addition to NVMe/FC. The NetApp plug-in in the native nvme-cli package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.

For additional details on supported configurations, see the [NetApp Interoperability Matrix Tool](#).

Features

- Support for NVMe secure, in-band authentication
- Support for persistent discovery controllers (PDCs) using a unique discovery NQN

Known limitations

- SAN booting using the NVMe-oF protocol is currently not supported.
- There's no sanlun support for NVMe-oF. Therefore, the host utility support isn't available for NVMe-oF on an SLES15 SP5 host. You can rely on the NetApp plug-in included in the native nvme-cli package for all NVMe-oF transports.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex FC adapters or Marvell/Qlogic FC adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the recommended adapter model:

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

Example output:

```
LPe32002 M2  
LPe32002-M2
```

2. Verify the adapter model description:

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

Example output:

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

3. Verify that you are using the recommended Emulex host bus adapter (HBA) firmware versions:

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

Example output:

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

4. Verify that you are using the recommended LPFC driver version:

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

Example output:

```
0:14.2.0.6
```

5. Verify that you can view your initiator ports:

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

Example output:

```
0x100000109b579d5e  
0x100000109b579d5f
```

6. Verify that your initiator ports are online:

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

Example output:

```
Online  
Online
```

7. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

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

Example output:

In this example, one initiator port is enabled and connected with two target LIFs.

```

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 DISCSRV ONLINE
NVME RPORT WWPN x208500a098dfdd91 WWNN x208100a098dfdd91 DID x010003
TARGET DISCSRV 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
ffffffffffffb046f
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 DISCSRV ONLINE
NVME RPORT WWPN x208200a098dfdd91 WWNN x208100a098dfdd91 DID x012a03
TARGET DISCSRV 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
fffffffffffffaf794
abort 00000b73 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000159d Err 000135c3

```

8. Reboot the host.

Marvell/QLogic

Steps

1. The native inbox qla2xxx driver included in the SLES 15 SP4 kernel has the latest fixes essential for ONTAP support. Verify that you are running the supported adapter driver and firmware versions:

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

Example output:

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

2. Verify that the `ql2xnvmeenable` parameter is set to 1:

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

Enable 1MB I/O size (Optional)

ONTAP reports an MDTs (Max Data Transfer Size) of 8 in the Identify Controller data which means the maximum I/O request size can be up to 1MB. However, to issue I/O requests of size 1 MB for a Broadcom NVMe/FC host, you must increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256.

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

2. Run a `dracut -f` command, and reboot the host.
3. Verify that `lpfc_sg_seg_cnt` is 256.

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



This is not applicable to Qlogic NVMe/FC hosts.

Enable NVMe services

There are two NVMe/FC boot services included in the `nvme-cli` package, however, *only* `nvme-fc-boot-connections.service` is enabled to start during system boot; `nvme-fc-autoconnect.service` is not enabled. Therefore, you need to manually enable `nvme-fc-autoconnect.service` to start during system boot.

Steps

1. Enable `nvme-fc-autoconnect.service`:


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

2. Reboot the host.
3. Verify that `nvme-autoconnect.service` and `nvme-fc-boot-connections.service` are running after the system boots up:

Example output:

```
# systemctl status nvme-autoconnect.service
nvme-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
   Loaded: loaded (/usr/lib/systemd/system/nvme-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]: nvme-autoconnect.service: Deactivated successfully.
systemd[1]: Finished Connect NVMe-oF subsystems automatically during
boot.

# systemctl status nvme-fc-boot-connections.service
nvme-fc-boot-connections.service - Auto-connect to subsystems on FC-NVME
devices found during boot
   Loaded: loaded (/usr/lib/systemd/system/nvme-fc-boot-
   connections.service; enabled; 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]: nvme-fc-boot-connections.service: Succeeded.
systemd[1]: Finished Auto-connect to subsystems on FC-NVME devices found
during boot.
```

Configure NVMe/TCP

You can use the following procedure to configure NVMe/TCP.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

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

Example output:

```
# 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. Verify that all other NVMe/TCP initiator-target LIF combinations can successfully fetch discovery log page data:

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

Example output:

```

# 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. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

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

Example output:

```
# 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 recommends setting the `ctrl-loss-tmo` option to `-1` so that the NVMe/TCP initiator attempts to reconnect indefinitely in the event of a path loss.

Validate NVMe-oF

You can use the following procedure to validate NVMe-oF.

Steps

1. Verify that in-kernel NVMe multipath is enabled:

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

2. Verify that the host has the correct controller model for the ONTAP NVMe namespaces:

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

Example output:

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

3. Verify the NVMe I/O policy for the respective ONTAP NVMe I/O controller:

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

Example output:

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

4. Verify that the ONTAP namespaces are visible to the host:

```
nvme list -v
```

Example output:

```
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 Address          Subsystem    Namespaces
-----
-----
nvme0      81LGgBUqsI3EAAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.2.214,trsvcid=4420,host_traddr=192.168.2.14 nvme-subsys0
nvme0n1
nvme1      81LGgBUqsI3EAAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.2.215,trsvcid=4420,host_traddr=192.168.2.14 nvme-subsys0
nvme0n1
nvme2      81LGgBUqsI3EAAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.1.214,trsvcid=4420,host_traddr=192.168.1.14 nvme-subsys0
nvme0n1
nvme3      81LGgBUqsI3EAAAAAAAAE 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. Verify that the controller state of each path is live and has the correct ANA status:

```
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. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

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

Example output:

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
```

Example output:

```
{
  "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
    }
  ]
}
```

Create a persistent discovery controller

Beginning with ONTAP 9.11.1, you can create a persistent discovery controller (PDC) for your SLES 15 SP4 host by using the following procedure. A PDC is required to automatically detect NVMe subsystem add or remove scenarios and changes to the discovery log page data.

Steps

1. Verify that the discovery log page data is available and can be retrieved through the initiator port and target

LIF combination:

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


Example output:

```
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.0501daf15dda11eeab68d039eea7a232: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.0501daf15dda11eeab68d039eea7a232:subsystem.unidir_d
hchap
traddr:  192.168.2.214
eflags:  none
sectype: none

```

2. Create a PDC for the discovery subsystem:

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

Example output:

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

3. From the ONTAP controller, verify that the PDC has been created:

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

Example output:

```
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-
74362c0cla1fc
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
```

Set up secure in-band authentication

Beginning with ONTAP 9.12.1, secure, in-band authentication is supported over NVMe/TCP and NVMe/FC between your SLES 15 SP4 host and your ONTAP controller.

To set up secure authentication, each host or controller must be associated with a `DH-HMAC-CHAP` key, which is a combination of the NQN of the NVMe host or controller and an authentication secret configured by the administrator. To authenticate its peer, an NVMe host or controller must recognize the key associated with the peer.

You can set up secure in-band authentication using the CLI or a config JSON file. If you need to specify different `dhchap` keys for different subsystems, you must use a config JSON file.

CLI

Steps

1. Obtain the host NQN:

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

2. Generate the dhchap key for the SLES15 SP4 host:

```
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

+

In the following example, a random dhchap key with HMAC set to 3 (SHA-512) is generated.

```
# nvme gen-dhchap-key -m 3 -n nqn.2014-08.org.nvmexpress:uuid:d3ca725a-  
ac8d-4d88-b46a-174ac235139b  
DHHC-  
1:03:J2UJQfj9f0pLnPF/ASDJRTyILKJRR5CougGpGdQSysPrLu6RW1fG15VSjbeDF1n1DE  
h3nVBe19nQ/LxreSBeH/bx/pU=:
```

1. On the ONTAP controller, add the host and specify both dhchap keys:

```
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. A host supports two types of authentication methods, unidirectional and bidirectional. On the host, connect to the ONTAP controller and specify dhchap keys based on the chosen authentication method:

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

3. Validate the `nvme connect` authentication command by verifying the host and controller dhchap keys:

a. Verify the host dhchap keys:

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

Example output for unidirectional configuration:

```
SR650-14-114:~ # cat /sys/class/nvme-subsystem/nvme-  
subsys1/nvme*/dhchap_secret  
DHC-  
1:03:je1nQCmjJLUKD62mpYbz1puw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8  
wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:  
DHC-  
1:03:je1nQCmjJLUKD62mpYbz1puw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8  
wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:  
DHC-  
1:03:je1nQCmjJLUKD62mpYbz1puw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8  
wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:  
DHC-  
1:03:je1nQCmjJLUKD62mpYbz1puw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8  
wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
```

b. Verify the controller dhchap keys:

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

Example output for bidirectional configuration:


```
SR650-14-114:~ # cat /sys/class/nvme-subsystem/nvme-
subsys6/nvme*/dhchap_ctrl_secret
DHHC-
1:03:WorVEV83eY053kV4Iel5OpphbX5LAph03F8fgH3913tlrkSGDBJTt3crXeTU
B8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eY053kV4Iel5OpphbX5LAph03F8fgH3913tlrkSGDBJTt3crXeTU
B8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eY053kV4Iel5OpphbX5LAph03F8fgH3913tlrkSGDBJTt3crXeTU
B8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eY053kV4Iel5OpphbX5LAph03F8fgH3913tlrkSGDBJTt3crXeTU
B8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
```

JSON file

You can use the `/etc/nvme/config.json` file with the `nvme connect-all` command when multiple NVMe subsystems are available on the ONTAP controller configuration.

You can generate the JSON file using `-o` option. Refer to the NVMe connect-all man pages for more syntax options.

Steps

1. Configure the JSON file:

```
# 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
ptWpnalrpwG5CndpOgxprXh9m4lw=: "
  },
  {
    "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. Connect to the ONTAP controller using the config JSON file:

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

Example output:

```
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. Verify that the dhchap secrets have been enabled for the respective controllers for each subsystem:

a. Verify the host dhchap keys:

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

Example output:

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

b. Verify the controller dhchap keys:

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

Example output:

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

Known issues

There are no known issues for the SLES 15 SP4 with ONTAP release.

NVMe-oF Host Configuration for SUSE Linux Enterprise Server 15 SP3 with ONTAP

NVMe over Fabrics or NVMe-oF (including NVMe/FC and other transports) is supported with SUSE Linux Enterprise Server 15 SP3 (SLES15 SP3) with ANA (Asymmetric Namespace Access). ANA is the ALUA equivalent in NVMe-oF environment, and is currently implemented with in-kernel NVMe Multipath. The details for enabling NVMe-oF with in-kernel NVMe Multipath using ANA on SLES15 SP3 and ONTAP as the target has been documented [here](#).

Refer to the [NetApp Interoperability Matrix](#) for accurate details regarding supported configurations.

Features

- SLES15 SP3 supports NVMe/FC and other transports.
- There is no sanlun support for NVMe-oF. Therefore, there is no LUHU support for NVMe-oF on SLES15 SP3. You can rely on the NetApp plug-in included in the native nvme-cli for the same instead. This should work for all NVMe-oF transports.
- Both NVMe and SCSI traffic can be run on the same co-existent host. In fact, that is expected to be the commonly deployed host config for customers. Therefore, for SCSI, you may configure `dm-multipath` as usual for SCSI LUNs resulting in `mpath` devices, whereas NVMe multipath might be used to configure NVMe-oF multipath devices on the host.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Enable in-kernel NVMe Multipath

In-kernel NVMe multipath is already enabled by default on SLES hosts such as SLES15 SP3. Therefore, no additional setting is required here. Refer to the [NetApp Interoperability Matrix](#) for accurate details regarding supported configurations.

NVMe-oF initiator packages

Refer to the [NetApp Interoperability Matrix](#) for accurate details regarding supported configurations.

1. Verify that you have the requisite kernel & nvme-cli MU packages installed on the SLES15 SP3 MU host.

Example:

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

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

The above nvme-cli MU package now includes the following:

- **NVMe/FC auto-connect scripts** - Required for NVMe/FC auto-(re)connect when underlying paths to the namespaces are restored as well as during the host reboot:

```
# 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
...
```

- **ONTAP udev rule** - New udev rule to ensure NVMe multipath round-robin loadbalancer default applies to all ONTAP namespaces:

```
# 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"
```

- **NetApp plug-in for ONTAP devices** - The existing NetApp plug-in has now been modified to handle ONTAP namespaces as well.
2. Check the hostnqn string at /etc/nvme/hostnqn on the host and ensure that it properly matches with the hostnqn string for the corresponding subsystem on the ONTAP array. For example,

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

Proceed with the below steps depending on the FC adapter being used on the host.

Configure NVMe/FC

Broadcom/Emulex

1. Verify that you have the recommended adapter and firmware versions. For example,

```
# 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
```

- The newer lpfc drivers (both inbox and outbox) already have `lpfc_enable_fc4_type` default set to 3, therefore, you no longer need to set this explicitly in the `/etc/modprobe.d/lpfc.conf`, and recreate the `initrd`. The `lpfc nvme` support is already enabled by default:

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

- The existing native inbox lpfc driver is already the latest and compatible with NVMe/FC. Therefore, you do not need to install the lpfc oob driver.

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

2. Verify that the initiator ports are up and running:

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

3. Verify that the NVMe/FC initiator ports are enabled and you are able to see the target ports, and all are up and running. In this example, only 1 initiator port is enabled and connected with two target LIFs as seen in the output:

```
# 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 DISCSRV ONLINE
NVME RPORT WWPN x208500a098dfdd91 WWNN x208100a098dfdd91 DID x010003
TARGET DISCSRV 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
ffffffffffffb046f
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 DISCSRV ONLINE
NVME RPORT WWPN x208200a098dfdd91 WWNN x208100a098dfdd91 DID x012a03
TARGET DISCSRV 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
fffffffffffffaf794
abort 00000b73 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000159d Err 000135c3
```

4. Reboot the host.

Enable 1MB I/O Size (Optional)

ONTAP reports an MDTs (Max Data Transfer Size) of 8 in the Identify Controller data which means the maximum I/O request size should be up to 1 MB. However, to issue I/O requests of size 1 MB for the Broadcom NVMe/FC host, the lpfc parameter `lpfc_sg_seg_cnt` should also be bumped up to 256 from the default value of 64. Use the following instructions to do so:

1. Append the value 256 in the respective modprobe `lpfc.conf` file:


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

2. Run a `dracut -f` command, and reboot the host.
3. After reboot, verify that the above setting has been applied by checking the corresponding sysfs value:

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

Now the Broadcom NVMe/FC host should be able to send up 1MB I/O requests on the ONTAP namespace devices.

Marvell/QLogic

The native inbox `qla2xxx` driver included in the newer SLES15 SP3 MU kernel has the latest upstream fixes, essential for ONTAP support.

1. Verify that you are running the supported adapter driver and firmware versions, for example:

```
# 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. Verify `ql2xnvmeenable` is set which enables the Marvell adapter to function as a NVMe/FC initiator:

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

Configure NVMe/TCP

Unlike NVMe/FC, NVMe/TCP has no auto-connect functionality. This manifests two major limitations on the Linux NVMe/TCP host:

- **No auto-reconnect after paths get reinstated** NVMe/TCP cannot automatically reconnect to a path that is reinstated beyond the default `ctrl_loss_tmo` timer of 10 minutes following a path down.
- **No auto-connect during host bootup** NVMe/TCP cannot automatically connect during host bootup as well.

You should set the retry period for failover events to at least 30 minutes to prevent timeouts. You can increase the retry period by increasing the value of the `ctrl_loss_tmo` timer. Following are the details:

Steps

1. Verify whether the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```

# 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.56e362e9bb4f11ebbade039ea165abc: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.56e362e9bb4f11ebbade039ea165abc: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.56e362e9bb4f11ebbade039ea165abc:subsystem.nvme_118_tcp
_2
traddr: 192.168.2.56
sectype: none
...

```

2. Verify that other NVMe/TCP initiator-target LIF combos are able to successfully fetch discovery log page data. For example,

```
# 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. Run `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes. Ensure you set a longer `ctrl_loss_tmo` timer retry period (for example, 30 minutes, which can be set through `-l 1800`) during the connect-all so that it would retry for a longer period of time in the event of a path loss. For example,

```
# 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
```

Validate NVMe-oF

1. Verify that in-kernel NVMe multipath is indeed enabled by checking:

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

2. Verify that the appropriate NVMe-oF settings (such as, `model` set to NetApp ONTAP Controller and load balancing `iopolicy` set to round-robin) for the respective ONTAP namespaces properly reflect on the host:

```
# 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. Verify that the ONTAP namespaces properly reflect on the host. For example,

```
# nvme list
```

Node	SN	Model	Namespace
/dev/nvme0n1	81CZ5BQuUNfGAAAAAAB	NetApp ONTAP Controller	1

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

Another example:

```
# 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. Verify that the controller state of each path is live and has proper ANA status. For example,

```
# 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
```

Another example:

```
#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. Verify that the NetApp plug-in displays proper values for each ONTAP namespace device. For example,

```
# 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
    }
  ]
}
```

Another example:

```
# nvme netapp ontapdevices -o column
Device          Vserver          Namespace Path
-----
-----
/dev/nvme0n1 vs_tcp_114          /vol/tcpcnvme_114_1_0_1/tcpcnvme_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/tcpcnvme_114_1_0_1/tcpcnvme_114_ns",
      "NSID" : 1,
      "UUID" : "a6aee036-e12f-4b07-8e79-4d38a9165686",
      "Size" : "85.90GB",
      "LBA_Data_Size" : 4096,
      "Namespace_Size" : 20971520
    }
  ]
}
```

== Known issues

There are no known issues.

NVMe/FC Host Configuration for SUSE Linux Enterprise Server 15 SP2 with ONTAP

NVMe/FC is supported on ONTAP 9.6 and above with SLES15 SP2. SLES15 SP2 host can run both NVMe/FC, & FCP traffic through the same fibre channel initiator adapter ports. See the [Hardware Universe](#) for a list of supported FC adapters and controllers.

For the most current list of supported configurations & versions, see the [NetApp Interoperability Matrix](#).



You can use the configuration settings provided in this document to configure cloud clients connected to [Cloud Volumes ONTAP](#) and [Amazon FSx for ONTAP](#).

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Enable NVMe/FC on SLES15 SP2

1. Upgrade to the recommended SLES15 SP2 MU kernel version.
2. Upgrade the native nvme-cli package.

This native nvme-cli package contains the NVMe/FC auto-connect scripts, ONTAP udev rule which enables round-robin load balancing for NVMe Multipath as well as the NetApp plug-in for ONTAP namespaces.

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

3. On the SLES15 SP2 host, check the host NQN string at /etc/nvme/hostnqn and verify that it matches the host NQN string for the corresponding subsystem on the ONTAP array. For example:

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

```
::> vserver nvme subsystem host show -vserver vs_fc_nvme_145
Vserver Subsystem Host NQN
-----
-----
vs_fc_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.
```

4. Reboot the host.

Configure the Broadcom FC Adapter for NVMe/FC

1. Verify that you are using the supported adapter. For the most current list of supported adapters see the [NetApp Interoperability Matrix](#).

```
# 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. Verify that you are using the recommended Broadcom lpfc firmware and native inbox driver versions.

```
# 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. Verify that lpfc_enable_fc4_type is set to 3.

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

4. Verify that the initiator ports are up and running.

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

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

5. Verify that the NVMe/FC initiator ports are enabled, running and able to see the target LIFs.

```
# 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 DISCSRV ONLINE
NVME RPORT WWPN x208500a098dfdd91 WWNN x208100a098dfdd91 DID x010003
TARGET DISCSRV 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
ffffffffffffb046f
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 DISCSRV ONLINE
NVME RPORT WWPN x208200a098dfdd91 WWNN x208100a098dfdd91 DID x012a03
TARGET DISCSRV 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
fffffffffffffaf794
abort 00000b73 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000159d Err 000135c3
```

Validate NVMe/FC

1. Verify the following NVMe/FC settings.

```
# 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. Verify that the namespaces are created.

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

3. Verify the status of the ANA paths.

```
# 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. Verify the NetApp plug-in for ONTAP devices.

```
# 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
    },
  ]
}
```

== Known issues

There are no known issues.

Enable 1MB I/O Size for Broadcom NVMe/FC

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data which means the maximum I/O request size can be up to 1MB. However, to issue I/O requests of size 1 MB for a Broadcom NVMe/FC host, you must increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256.

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

2. Run a `dracut -f` command, and reboot the host.
3. Verify that `lpfc_sg_seg_cnt` is 256.

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



This is not applicable to Qlogic NVMe/FC hosts.

LPFC Verbose Logging

Set the lpfc driver for NVMe/FC.

Steps

1. Set the `lpfc_log_verbose` driver setting to any of the following values to log NVMe/FC events.

```
#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. After setting the values, run the `dracut-f` command and reboot the host.
3. Verify the settings.

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

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

NVMe/FC Host Configuration for SUSE Linux Enterprise Server 15 SP1 with ONTAP

You can configure NVMe over Fibre Channel (NVMe/FC) on hosts running SUSE Linux Enterprise Server 15 SP1 and ONTAP as the target.

NVMe/FC is supported on ONTAP 9.6 or later for the following versions of SLES:

- SLES15 SP1

SLES15 SP1 host can run both NVMe/FC, & FCP traffic through the same fibre channel initiator adapter ports. See the [Hardware Universe](#) for a list of supported FC adapters and controllers.

For the most current list of supported configurations & versions, see the [NetApp Interoperability Matrix](#).

- Native NVMe/FC auto-connect scripts are included in the `nvme-cli` package. You can use the native inbox `lpfc` driver on SLES15 SP1.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Enable NVMe/FC on SLES15 SP1

1. Upgrade to the recommended SLES15 SP2 MU kernel
2. Upgrade to the recommended nvme-cli MU version.

This nvme-cli package contains the native NVMe/FC auto-connect scripts, so you do not need to install the external NVMe/FC auto-connect scripts provided by Broadcom on the SLES15 SP1 host. This package also includes the ONTAP udev rule which enables round-robin load balancing for NVMe multipath, and the NetApp plug-in for ONTAP devices.

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

3. On the SLES15 SP1 host, check the host NQN string at `/etc/nvme/hostnqn` and verify that it matches the host NQN string for the corresponding subsystem on the ONTAP array. For example:

```
# 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. Reboot the host.

Configure the Broadcom FC Adapter for NVMe/FC

1. Verify that you are using the supported adapter. For the most current list of supported adapters see the [NetApp Interoperability Matrix](#).

```
# 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. Verify that you are using the recommended Broadcom lpfc firmware and native inbox driver versions.

```
# 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. Verify that `lpfc_enable_fc4_type` is set to 3.

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

4. Verify that the initiator ports are up and running.

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

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

5. Verify that the NVMe/FC initiator ports are enabled, running and able to see the target LIFs.

```
# 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 DISCSRV ONLINE
NVME RPORT WWPN x203100a098c80f09 WWNN x202c00a098c80f09 DID x010601
TARGET DISCSRV ONLINE
NVME Statistics
...
```

Validate NVMe/FC

1. Verify the following NVMe/FC settings.

```
# 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. Verify that the namespaces are created.

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

3. Verify the status of the ANA paths.

```
# 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. Verify the NetApp plug-in for ONTAP devices.


```
# 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
    }
  ]
}
```

== Known issues

There are no known issues.

Enable 1MB I/O Size for Broadcom NVMe/FC

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data which means the maximum I/O request size can be up to 1MB. However, to issue I/O requests of size 1 MB for a Broadcom NVMe/FC host, you must increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256.

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

2. Run a `dracut -f` command, and reboot the host.
3. Verify that `lpfc_sg_seg_cnt` is 256.

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



This is not applicable to Qlogic NVMe/FC hosts.

LPFC Verbose Logging

Set the lpfc driver for NVMe/FC.

Steps

1. Set the `lpfc_log_verbose` driver setting to any of the following values to log NVMe/FC events.

```
#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. After setting the values, run the `dracut-f` command and reboot the host.
3. Verify the settings.

```
# cat /etc/modprobe.d/lpfc.conf options lpfc lpfc_log_verbose=0xf00083  
  
# cat /sys/module/lpfc/parameters/lpfc_log_verbose 15728771
```

Copyright information

Copyright © 2024 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.