BROADCOM®

Release Notes

Emulex® NVMe over Fibre Channel for SLES 12 SP3 and SLES 15

Release	SLES 12 SP3	11.4.329.43MU
	SLES 15	11.4.410.0

Purpose and Contact Information

These release notes describe the interoperability matrices, known issues, and technical tips associated with this release of the Emulex® Nonvolatile Memory Express (NVMe) over FC drivers and scripts for SLES 12 SP3 and SLES 15 on LPe31000-series and LPe32000-series adapters in initiator and target systems.

For the latest product documentation, go to www.broadcom.com. If you have questions or require additional information, contact an authorized Broadcom® Technical Support representative at ecd-tech.support@broadcom.com.

NVMe over FC Interoperability Matrices

You can build NVMe over FC end-to-end solutions with initiators from SUSE servers, with a Linux target “just a bunch of flash” (JBOF) based on SUSE. This capability enables you to build low-cost, high-performance NVMe over FC storage systems. The following table provides the required initiator and target pairs across the supported out-of-box Emulex NVMe over FC drivers.

If You Are Using This Initiator Driver	Use This Target Driver
`elx-lpfc-dd-sles12spMU-11.4.329.43-ds-1.tar.gz` for SLES 12 SP3 maintenance kernel 718 (4.4.126-94.22.1) from SUSE, dated April 23, 2018, or later	`elx-lpfc-dd-sles12spMU-11.4.329.43-ds-1.tar.gz` or later for SLES 12 SP3
`elx-lpfc-dd-sles15sp-11.4.410.0-ds-1.tar.gz` for SLES 15 GA release or later	`elx-lpfc-dd-sles12spMU-11.4.329.43-ds-1.tar.gz` or later for SLES 12 SP3

NOTE: Install firmware version 11.4.329.13 or later to ensure interoperability and to include the latest standards and performance updates.

Known Issues

1. NVMe multipath is enabled by default in SLES 15. Emulex did not test the NVMe multipath feature.

   Workaround

   You can disable NVMe multipath by adding `nvme-core.multipath=N` as a boot parameter in the Grand Unified Bootloader (`grub`) package.

2. In SLES 12 SP3, the `nvme list` command might not display all namespaces.

   Workaround

   Use the `lsblk` command to display all namespaces.

3. In a point-to-point connection, port speed changes might generate the following error:
   `Thread:16 System call error:5 - Input/output error (error in write completion)`
4. Running the `nvmetcli clear` command on an NVMe target when the NVMe initiator port is down might cause the target system to stop responding. Make sure the NVMe initiator port is up when running the `nvmetcli clear` command.
5. Warning messages appear when a new namespace is created on a target.
6. The deliberate faulting of NVME discovery commands (also called *jamming*) is not supported. If a `discovery` command is faulted during initial linkup or during the LIP linkup recovery time, NVME discovery fails on the affected controller.

   Workaround

   Perform a LIP on the initiator link, and remove the condition that is faulting discovery.
7. Only a maximum of 256 subsystems are discovered.
8. If you are adding a subsystem dynamically, you must perform a manual scan. See Technical Tips for more information.
9. Creation of N_Port ID Virtualization (NPIV) connections on initiator ports that are configured for NVMe over FC is not supported. However, initiator ports can connect to NPIV Fibre Channel Protocol (FCP) and NVMe targets simultaneously.
10. Fabric-Assigned Port World Wide Name (FA-PWWN) is not supported on initiator ports configured for NVMe over FC.
11. This release supports FC-NVMe specification version 1.18.
12. Target port reset is not supported in this release.
13. Dynamic NVMe target management, including reconfiguration, and dynamic initiator rescan are not supported. Any reconfiguration requires a target server reboot.

    After the target server reboots and the configuration is activated, connected initiators might require a link bounce to rescan the target.

Technical Tips

To manually scan for targets or dynamically added subsystems, type the following command (all on one line):

`nvme connect-all --transport=fc --host-traddr=nn-<`*`initiator_WWNN`*`>:pn-<`*`initiator_WWPN`*`> --traddr=nn-<`*`target_WWNN`*`>:pn-<`*`target_WWPN`*`>`

where:

- *<initiator_WWNN>* is the WWNN of the initiator, in hexadecimal.
- *<initiator_WWPN>* is the WWPN of the initiator, in hexadecimal.
- *<target_WWNN>* is the WWNN of the target, in hexadecimal.
- *<target_WWPN>* is the WWPN of the target, in hexadecimal.

For example:

```
# nvme connect-all --transport=fc --host-traddr=nn-0x20000090fa942779:pn-0x10000090fa942779 --traddr=nn-0x20000090fae39706:pn-0x10000090fae39706
```