STORAGE DEVELOPER CONFERENCE
SDC 22
Fremont, CA
September 12-15, 2022
BY Developers FOR Developers
A SNIA Event

STorage Appliance Services (nvme-stas)

An automatic NVMe-oF connectivity configurator for Linux Hosts

Presented by Claudio Desanti
Prepared by Martin Bélanger
Dell Technologies CTIO Group

Agenda

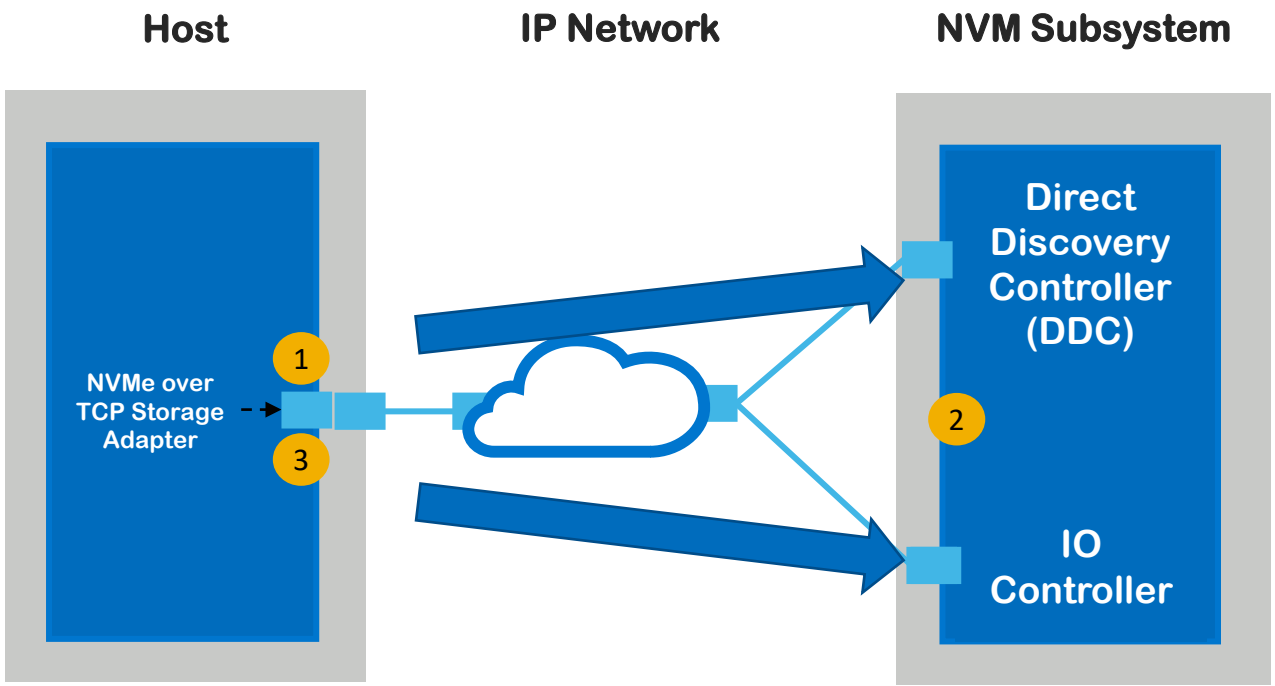
- Motivations for nvme-stas
- How it works
- How to get it
- How to use it
- The future of nvme-stas
- Conclusion

Motivations for nvme-stas

Motivations for nvme-stas

- Automate Host configuration:
  - TP8009 – Automated Discovery of NVMe-oF Discovery Controllers for IP Networks
  - TP8010 – NVMe-oF Centralized Discovery Controller
- Adapt to connectivity configuration changes (e.g., Fabric Zoning)
  - Make sure that Hosts connect/disconnect to/from subsystems on changes of connectivity configuration
  - Provide configurable connection audit modes to remove connections that are not allowed

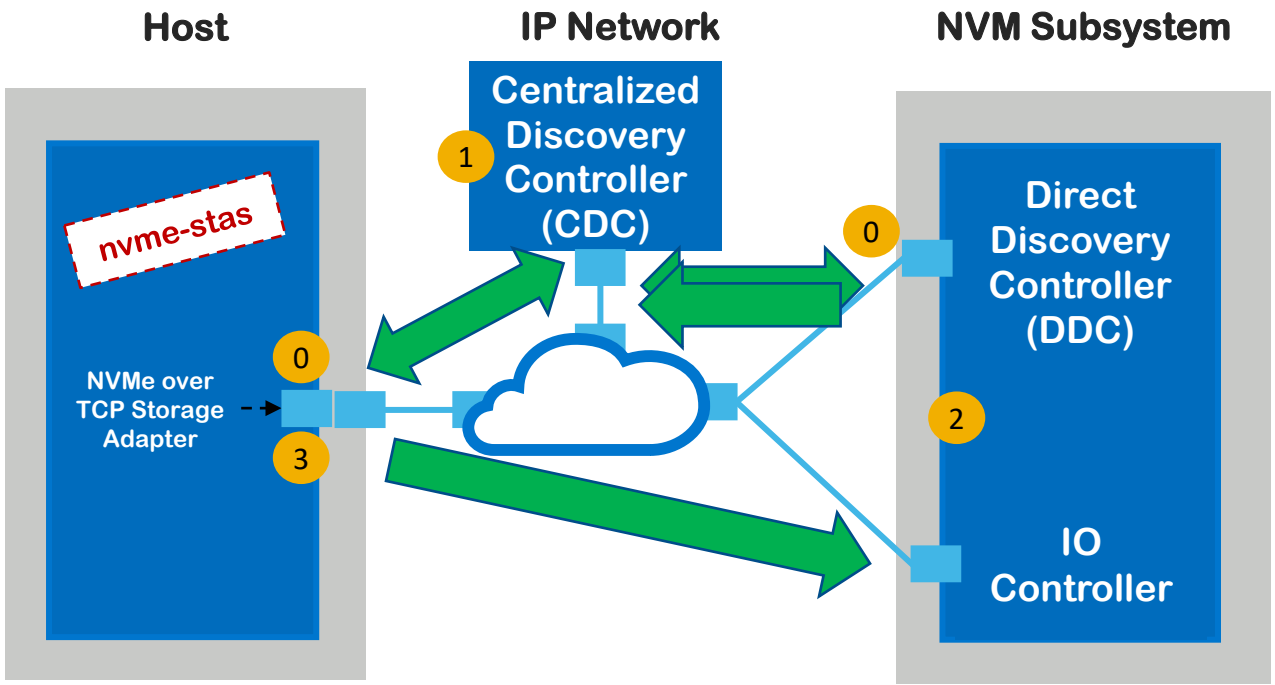
Configuration Steps without TP8009/TP8010



1. Host sends connect to Discovery Controller at IP Address supplied by admin
2. Storage admin provisions Namespaces (Storage) to the Host NQN
3. Host Admin uses nvme connect-all to Discover and connect to IO Controllers on that subsystem.
4. Repeat 1-3 on all Hosts for each subsystem

Labor intensive and error prone

Configuration Steps with TP8009/TP8010



0 Host and subsystems automatically discover the CDC, connect to it and Register Discovery info

1 Optional zoning performed on CDC (central configuration)

2 Storage admin provisions namespaces to the Host NQN. Storage may send zoning info to CDC

3 After zoning, Host receives AEN, uses get log page, and connects to each IO Controller

4 Repeat 1-2 for each Hosts on each subsystem

No manual configuration required at the host

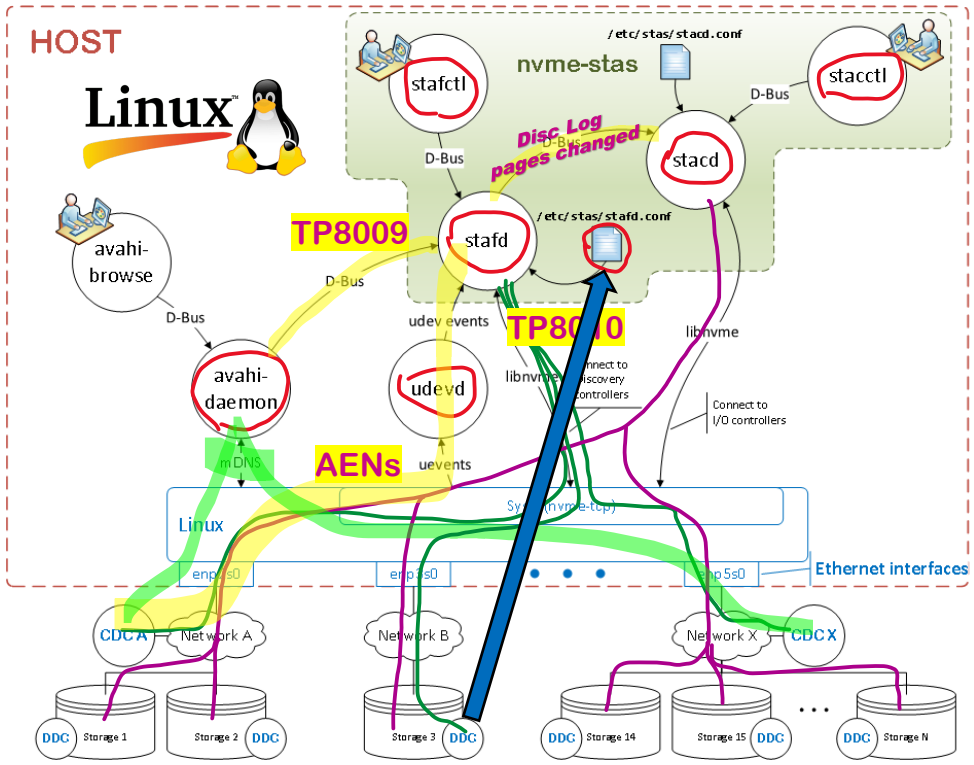
How it works

Looking under the hood

First a few definitions

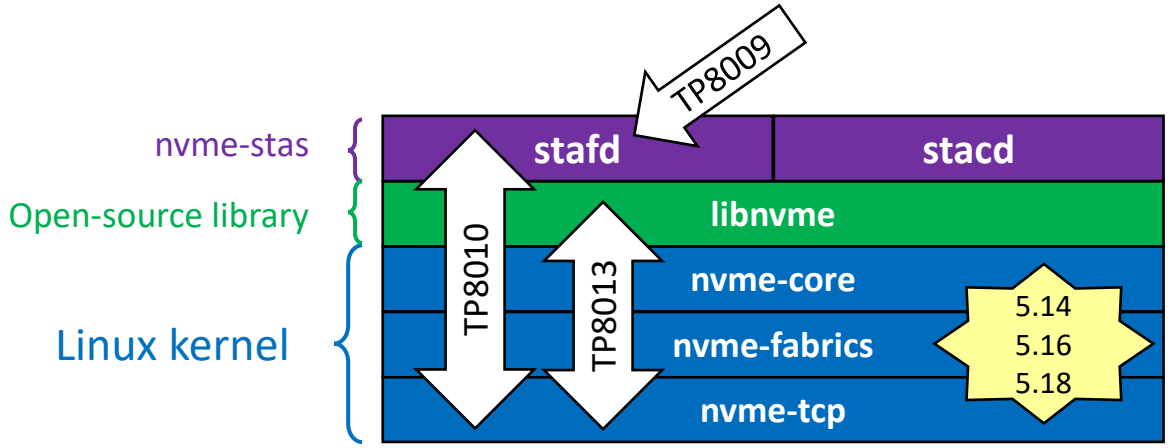
Term/Acronym	Definition
DC • **DDC** • **CDC**	**D**iscovery **C**ontroller • **D**irect **D**iscovery **C**ontroller • **C**entral **D**iscovery **C**ontroller (TP8010)
IOC	**I/O C**ontroller
Fabric Zoning	**Fabric Zoning** defines the connectivity relationships allowed between Host and Subsystem entities that are members of the same fabric. Entities that share a common zone (i.e., are zoned together) are allowed to discover each other and establish connections between them.
DLP / DLPE	A **D**iscovery **L**og **P**age is returned by **DC**s in response to a Get **DLP** command from a Host. A **D**iscovery **L**og **P**age is composed of zero or more **D**iscovery **L**og **P**age **E**ntries. A **DLPE** identifies a controller (**DC** or **IOC**) that the Host can connect to. **DC**s will only return the **DLPE**s for **DC**s and **IOC**s that are zoned together with the Host (**Fabric Zoning**).
AEN / AER	An **A**synchronous **E**vent **N**otification is a completion queue entry for an **A**synchronous **E**vent **R**equest that was previously transmitted by the host to a **DC**. The **AEN** is used to notify Hosts that a change (e.g., a connectivity configuration change) has occurred.

How it works



- **avahi-daemon**: Conventional Linux Service Discovery
  - Zeroconf networking – discover services on local net
- **udevd**: User-space Device daemon
  - Conventional event service for devices (/dev, /sys)
  - Relay kernel events to applications
- **stafd**: **ST**orage **A**pplicance **F**inder **D**aemon
  - **TP8009**: Auto-magically locate Discovery Controllers (DC) with the help of the avahi-daemon (Zeroconf)
  - Config file to manually specify CDCs, DDCs
  - NVMe-oF connectivity:
    - Connect to DCs
    - **TP8010** Perform Explicit Registration (DIM command)
    - Get the *discovery log pages* (DLP) from each DC
    - Maintain an up-to-date cache of the DLP
    - Monitor AENs and Refresh DLP cache
    - Send *Disc. Log pages changed* notifications over D-Bus
- **stacd**: **ST**orage **A**ppliance **C**onnector **D**aemon
  - Retrieve the list of Storage Appliances from stafd's cache
  - Set up I/O connections to Storage Appliances
  - Monitor and react to stafd *Disc. log pages changed* notifications
  - React to DLPE changes (resulting from connectivity configuration changes) and add/remove IOC connections as necessary (configurable)
- **stafctl** / **stacctl**: Companion utility programs used to display info from stafd / stacd

Components / Dependencies

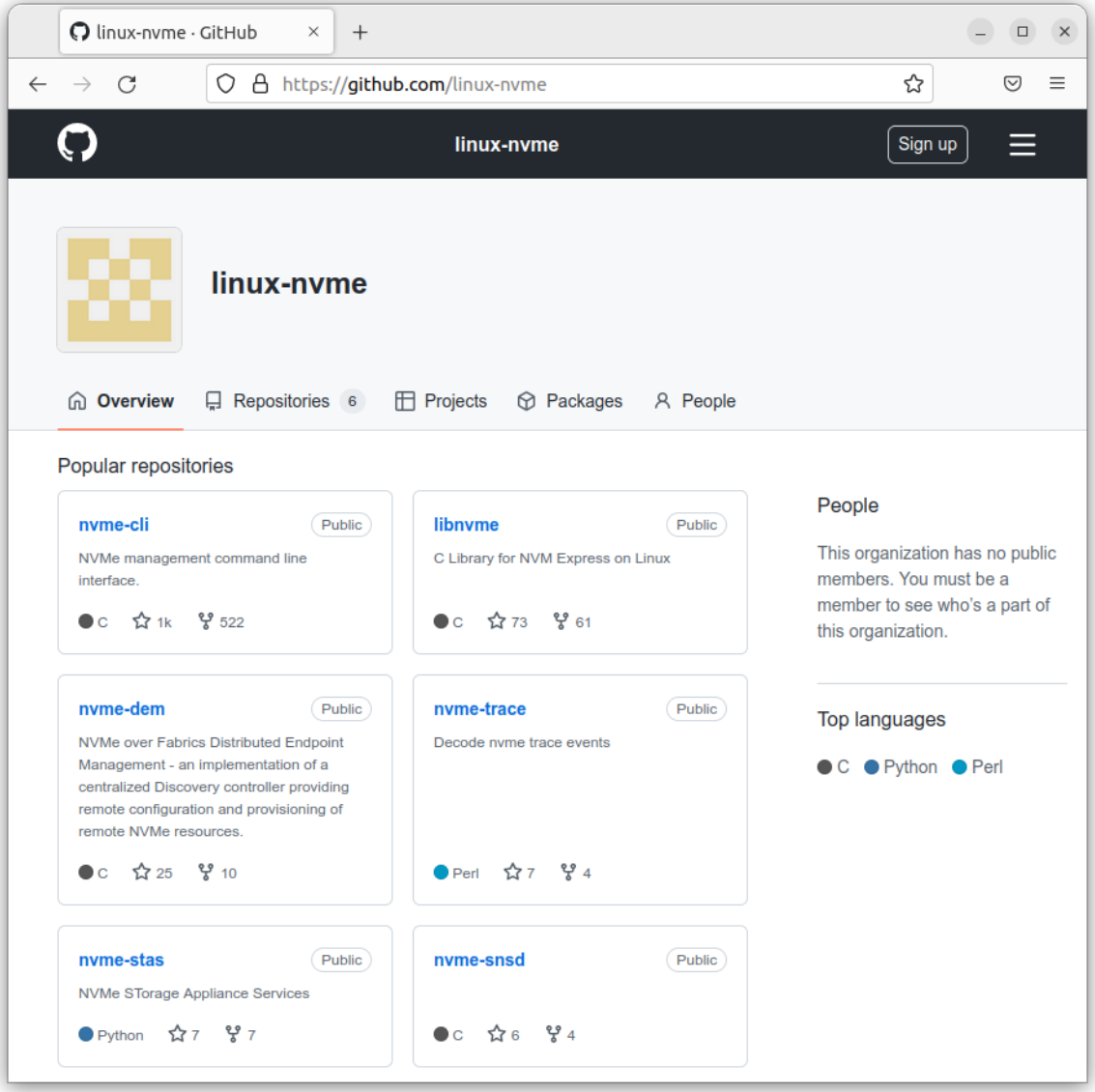


- nvme-stas is entirely written in Python
  - dependencies: dasbus, pyudev, gobject, systemd
- nvme-stas distributed under Apache-2 license
- Processes use D-Bus to talk to each other (conventional Linux IPC)
- nvme-stas D-Bus API:
  - Used by stafctl / stacctl
  - Allow 3rd party applications to interact with nvme-stas
  - Can be used for Automation & Testing

How to get it

It’s open source!

`nvme-stas` is an open-source project



- Hosted by the linux-nvme GitHub project
- Same location as `libnvme` and `nvme-cli`
- Debian, Red Hat, SuSE, etc., already package `nvme-cli` from this location
- Dell is the primary Maintainer for the `nvme-stas` repo
  - It's Open Source
  - Open to contributions
    - Pull request > Review > Merge

How to install and run it?

- Currently available with SLES15 SP4
  - `zypper install nvme-stas`
- To be included in RHEL 9.1 (as tech preview)
  - `yum install nvme-stas`
- For other distros (or older versions)
  - Build/Install from source
    - nvme-stas uses the meson build system (like libnvme)
  - No guarantees it will work.
    - Must meet dependencies minimum version requirements
    - See DISTROS.md for requirement list
- Additionally, it can be run in a container
  - A docker example can be found in the repo

How to use it

Documentation / Configuration

Documentation

GitHub

- README.md
- CONTRIBUTING.md
- DISTROS.md
- TESTING.md
- NEWS.md

man pages

- $ man nvme-stas
- $ man stafd
- $ man stafd.conf
- $ man stafctl
- $ man stacd
- $ man stacd.conf
- $ man stacctl
- etc…

Read the docs

- https://nvme-stas.readthedocs.io

Direct configuration

- **`nvme-stas`** works right out of the box without any manual intervention
  - `zeroconf=enabled # mDNS discovery`
- Configuration files allow tweaking stafd/stacd operation:
  - `/etc/stas/`**`stafd.conf`** and `/etc/stas/`**`stacd.conf`**
  - Config files are self-documented
  - man pages are available:
    - `$ man stafd.conf`
    - `$ man stacd.conf`

```
 GNU nano 4.8                    /etc/stas/stafd.conf                 Modified
# Copyright (c) 2021, Dell Inc. or its subsidiaries.  All rights reserved.
# SPDX-License-Identifier: Apache-2.0
# See the LICENSE file for details.
#
# This file is part of NVMe STorage Appliance Services (nvme-stas).

# ==============================================================================
# STorage Appliance Finder Daemon (stafd) - configuration file
#
# In this file, options that are commented represent the default values used.
# Uncommented options override the default value.


[Global]
# tron:         Trace-ON. Enable additional debug info
#               Type:  boolean
#               Range: [false, true]
#tron=false
tron = true

# hdr-digest:   Protocol Data Unit (PDU) Header Digest. NVMe/TCP facilitates an
#               optional PDU Header digest. Digests are calculated using the
#               CRC32C algorithm.
#               Type:  boolean
#               Range: [false, true]
#hdr-digest=false

# data-digest:  Protocol Data Unit (PDU) Data Digest. NVMe/TCP facilitates an
#               optional PDU Data digest. Digests are calculated using the
#               CRC32C algorithm.
#               Type:  boolean
#               Range: [false, true]
#data-digest=false

# kato:         Keep Alive Timeout (KATO): This field specifies the timeout value
#               for the Keep Alive feature in seconds. The default value for this
#               field is 30 seconds.
#               Type:  Unsigned integer
#               Range: 0..N
#               Unit:  Seconds
#kato=30

# persistent-connections: Whether connections to Discovery Controllers (DC)
#                         are persistent. If stafd is stopped, the connections
#                         will persists. When this is set to false, stafd will
#                         disconnect from all DCs it is connected to.
#                         Type:  boolean
```


Configuration parameters: /etc/stas/**stafd.conf**

- **[Global]**
  - **kato** = **30** # in seconds. 0 to disable.
  - **tron** = [**false** | true] # **TR**ace **ON** – to journal/syslog
  - **hdr-digest** = [**false** | true]
  - **data-digest** = [**false** | true]
  - **ip-family** = [**ipv4+ipv6** | ipv4 | ipv6]
- **[Service Discovery]**
  - **zeroconf** = [**enabled** | disabled]
- **[Controllers]** # Multiple "controller=" or "exclude=" entries are used to define lists of controllers/exclusions
  - **controller** = transport=tcp;traddr=[IP];trsvcid=[8009];nqn=[NQN];host-iface=[I/F]
  - **exclude** = [same as controller= or *a subset*]
    - exclude = traddr=192.168.1.10
      exclude = nqn=nqn.1988-11.com.dell:SFSS:1:20220411152957e8
      exclude = host-iface=enp0s8

Configuration parameters: /etc/stas/**stacd.conf**

- **[Global]**
  - *Same as stafd.conf (except kato defaults to 120 sec instead of 30)*
- **[I/O controller disconnect policy]**
  - **disconnect-scope** = [**only-stas-connections** | all-connections-matching-disconnect-trtypes | no-disconnect]
  - **disconnect-trtypes** = [**tcp** | rdma | fc | tcp+rdma | tcp+fc | tcp+rdma+fc]
- **[Controllers]**
  - *Same as stafd.conf*

The future of nvme-stas

Coming soon…

Features currently on the drawing board

- Security
  - Authentication
  - TLS connections (requires in-kernel TLS)
  - In-kernel support for the Authentication Verification Entity (AVE)
  - AVE Discovery Log page
- Multi driver support
  - Linux kernel (libnvme)
  - SPDK with JSON-RPC
  - Others with gRPC
- Integrate with autofs
- I/O reconnect policies
  - For quick reconnect on reboot
- Use PLEO bit to get only Port Local Entries when retrieving log pages
  - Tell DDC to only send DLPEs that can be reached on the local interface
- Conditional connect based on NCC (No CDC Connectivity) bit
  - When failing to connect to IOC and if the CDC reports that it isn't connected to subsystem (NCC), then stop trying to connect to IOC until CDC clears NCC
- Detect and report conflicts with nvme-cli configuration (`/etc/nvme/discovery.conf`)
- K8s integration
- Ansible playbooks to automate configuration

Conclusion

Conclusion

- When to use `nvme-stas`?
  - You want to automatically discover and connect to discovery controllers (zeroconf)
  - You want to adapt to connectivity configuration changes (e.g., Fabric Zoning)
  - You require more control and monitoring
    - IPv4 and/or IPv6
    - Connection audits/retries
- Nvme-cli interaction
  - `nvme-cli` can work in parallel
  - Be mindful of configuration conflicts between nvme-cli and nvme-stas
    - `/etc/nvme/discover.conf` vs. `/etc/stas/stafd.conf`
    - Working on a conflict detection/resolution tool

Please take a moment to rate this session.

Your feedback is important to us.

Features - comparing nvme-stas with nvme-cli

Feature	nvme-stas	nvme-cli
IP address family filter	**Yes** – /etc/stas/*.conf: **ip-family**=[ipv4, ipv6, ipv4+ipv6]	**No**
Automatic DIM registration with Central Discovery Controller (CDC) per TP8010	**Yes**	**No** – Manual only: nvme dim
Automatic (zeroconf) discovery of Direct/Central Discovery Controller (DDC/CDC)	**Yes** – stas registers with the Avahi daemon to be notified when CDCs or DDCs are detected by mDNS service discovery and automatically connects to them.	**No**
Manual Discovery Controller (DC) config with explicit include/exclude	**Yes** – /etc/stas/stafd.conf: **controller=, exclude=** Exclusion is needed to eliminate unwanted mDNS-discovered DCs.	**Partial** – No way to exclude DCs (*moot point since mDNS is not supported*). Use /etc/nvme/discovery.conf to include controllers.
Manual I/O Controller (IOC) config with explicit include/exclude	**Yes** – /etc/stas/stacd.conf: **controller=, exclude=** Exclusion is needed to eliminate unwanted IOCs from log pages *(although this should really be done by properly defining the zones at the DC)*	**No**
AEN monitoring + Auto Connection/Disconnect for Fabric Zoning support	**Yes** – Adapt to connectivity configuration changes (configurable: /etc/stas/stacd.conf) Connect and Disconnect with retries.	**Partial** – Adapt to connectivity configuration changes Connect-only without retries (one-shot udev rule)
Use PLEO bit to get only Port Local Entries when retrieving log pages	**Planned** – Under design	**No**
Layer 3 connectivity w/o static routes	**Yes** – Automatic/Configurable (/etc/stas/*.conf: ignore-iface=)	**Yes** – Manual (--host-iface)
Explicit exclude of specific interfaces used for discovery	**Yes** – /etc/stas/*.conf: **exclude** = host-iface=<interface>	**No**
AVE client support	**Planned** – Under design	**No**
Support for drivers other than linux-nvme (e.g., SPDK)	**Planned** – Framework in place	**No**
Human-friendly nvme list	**No** – stas only displays data in JSON format (for now). *Not needed since "nvme list -v" does the job so well.*	**Yes** – nvme list -v