



Full credit is given to the above companies including the OS that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'nvme-discover.1'

\$ man nvme-discover.1

NVME-DISCOVER(1) NVMe Manual NVME-DISCOVER(1)

NAME

nvme-discover - Send Get Log Page request to Discovery Controller.

SYNOPSIS

nvme discover

```
 [--transport=<trtype> | -t <trtype>]
 [--nqn=<subnqn> | -n <subnqn>]
 [--traddr=<traddr> | -a <traddr>]
 [--trsvcid=<trsvcid> | -s <trsvcid>]
 [--host-traddr=<traddr> | -w <traddr>]
 [--host-iface=<iface> | -f <iface>]
 [--hostnqn=<hostnqn> | -q <hostnqn>]
 [--hostid=<hostid> | -l <hostid>]
 [--raw=<filename> | -r <filename>]
 [--device=<device> | -d <device>]
 [--cfg-file=<cfg> | -C <cfg> ]
 [--keep-alive-tmo=<sec> | -k <sec>]
 [--reconnect-delay=<#> | -c <#>]
```

```
[--ctrl-loss-tmo=<#> | -l <#>]
[--hdr_digest | -g]
[--data_digest | -G]
[--nr-io-queues=<#> | -i <#>]
[--nr-write-queues=<#> | -W <#>]
[--nr-poll-queues=<#> | -P <#>]
[--queue-size=<#> | -Q <#>]
[--persistent | -p]
[--quiet | -S]
[--dump-config | -O]
[--output-format=<fmt> | -o <fmt>]
[--force]
```

DESCRIPTION

Send one or more Get Log Page requests to a NVMe-over-Fabrics Discovery Controller.

If no parameters are given, then `nvme discover` will attempt to find a `/etc/nvme/discovery.conf` file to use to supply a list of Discovery commands to run. If no `/etc/nvme/discovery.conf` file exists, the command will quit with an error.

Otherwise, a specific Discovery Controller should be specified using the `--transport`, `--traddr`, and if necessary the `--trsvcid` flags. A Discovery request will then be sent to the specified Discovery Controller.

BACKGROUND

The NVMe-over-Fabrics specification defines the concept of a Discovery Controller that an NVMe Host can query on a fabric network to discover NVMe subsystems contained in NVMe Targets which it can connect to on the network. The Discovery Controller will return Discovery Log Pages that provide the NVMe Host with specific information (such as network address and unique subsystem NQN) the NVMe Host can use to issue an NVMe connect command to connect itself to a storage resource contained in that NVMe subsystem on the NVMe Target.

Note that the base NVMe specification defines the NQN (NVMe Qualified

Name) format which an NVMe endpoint (device, subsystem, etc) must follow to guarantee a unique name under the NVMe standard. In particular, the Host NQN uniquely identifies the NVMe Host, and may be used by the the Discovery Controller to control what NVMe Target resources are allocated to the NVMe Host for a connection.

A Discovery Controller has its own NQN defined in the NVMe-over-Fabrics specification, nqn.2014-08.org.nvmexpress.discovery. All Discovery Controllers must use this NQN name. This NQN is used by default by nvme-cli for the discover command.

OPTIONS

`-t <trtype>, --transport=<trtype>`

This field specifies the network fabric being used for a NVMe-over-Fabrics network. Current string values include:

????????????????????????????????????????????

?Value ? Definition ?

????????????????????????????????????????????

?rdma ? The network fabric is an ?

? ? rdma network (RoCE, iWARP, ?

? ? Infiniband, basic rdma, ?

? ? etc) ?

????????????????????????????????????????????

?fc ? WIP The network fabric is ?

? ? a Fibre Channel network. ?

????????????????????????????????????????????

?tcp ? The network fabric is a ?

? ? TCP/IP network. ?

????????????????????????????????????????????

?loop ? Connect to a NVMe over ?

? ? Fabrics target on the ?

? ? local host ?

????????????????????????????????????????????

`-n <subnqn>, --nqn <subnqn>`

This field specifies the name for the NVMe subsystem to connect to.

-a <traddr>, --traddr=<traddr>

This field specifies the network address of the Discovery Controller. For transports using IP addressing (e.g. rdma) this should be an IP-based (ex. IPv4) address.

-s <trsvcid>, --trsvcid=<trsvcid>

This field specifies the transport service id. For transports using IP addressing (e.g. rdma) this field is the port number. By default, the IP port number for the RDMA transport is 4420.

-w <traddr>, --host-traddr=<traddr>

This field specifies the network address used on the host to connect to the Controller. For TCP, this sets the source address on the socket.

-f <iface>, --host-iface=<iface>

This field specifies the network interface used on the host to connect to the Controller (e.g. IP eth1, enp2s0, enx78e7d1ea46da). This forces the connection to be made on a specific interface instead of letting the system decide.

-q <hostnqn>, --hostnqn=<hostnqn>

Overrides the default host NQN that identifies the NVMe Host. If this option is not specified, the default is read from /etc/nvme/hostnqn first. If that does not exist, the autogenerated NQN value from the NVMe Host kernel module is used next.

-l <hostid>, --hostid=<hostid>

UUID(Universally Unique Identifier) to be discovered which should be formatted.

-r <filename>, --raw=<filename>

This field will take the output of the nvme discover command and dump it to a raw binary file. By default nvme discover will dump the output to stdout.

-d <device>, --device=<device>

This field takes a device as input. Device is in the format of nvme*, eg. nvme0, nvme1

-C <cfg>, --config-file=<cfg>

Use the specified JSON configuration file instead of the default /etc/nvme/config.json file or none to not read in an existing configuration file. The JSON configuration file format is documented in

<https://github.com/linux-nvme/libnvme/doc/config-schema.json>

-k <#>, --keep-alive-tmo=<#>

Overrides the default timeout (in seconds) for keep alive. This option will be ignored for the discovery, and it is only implemented for completeness.

-c <#>, --reconnect-delay=<#>

Overrides the default delay (in seconds) before reconnect is attempted after a connect loss.

-l <#>, --ctrl-loss-tmo=<#>

Overrides the default controller loss timeout period (in seconds).

-g, --hdr_digest

Generates/verifies header digest (TCP).

-G, --data_digest

Generates/verifies data digest (TCP).

-i <#>, --nr-io-queues=<#>

Overrides the default number of I/O queues create by the driver. This option will be ignored for the discovery, and it is only implemented for completeness.

-W <#>, --nr-write-queues=<#>

Adds additional queues that will be used for write I/O.

-P <#>, --nr-poll-queues=<#>

Adds additional queues that will be used for polling latency sensitive I/O.

-Q <#>, --queue-size=<#>

Overrides the default number of elements in the I/O queues created by the driver which can be found at drivers/nvme/host/fabrics.h. This option will be ignored for the discovery, and it is only implemented for completeness.

-p, --persistent

Persistent discovery connection.

-S, --quiet

Suppress already connected errors.

-O, --dump-config

Print out resulting JSON configuration file to stdout.

-o <format>, --output-format=<format>

Set the reporting format to normal, json, or binary. Only one output format can be used at a time.

--force

Disable the built-in persistent discover connection rules. Combined with --persistent flag, always create new persistent discovery connection.

EXAMPLES

? Query the Discover Controller with IP4 address 192.168.1.3 for all resources allocated for NVMe Host name host1-rogue-nqn on the RDMA network. Port 4420 is used by default:

```
# nvme discover --transport=rdma --traddr=192.168.1.3 \  
--hostnqn=host1-rogue-nqn
```

? Issue a nvme discover command using a /etc/nvme/discovery.conf

file:

```
# Machine default 'nvme discover' commands. Query the  
# Discovery Controller's two ports (some resources may only  
# be accessible on a single port). Note an official  
# nqn (Host) name defined in the NVMe specification is being used  
# in this example.  
  
-t rdma -a 192.168.69.33 -s 4420 -q nqn.2014-08.com.example:nvme:nvm-subsystem-sn-d78432  
-t rdma -a 192.168.1.4 -s 4420 -q nqn.2014-08.com.example:nvme:nvm-subsystem-sn-d78432
```

At the prompt type "nvme discover".

SEE ALSO

nvme-connect(1) nvme-connect-all(1)

AUTHORS

This was written by Jay Freyensee[1]

Part of the nvme-user suite

NOTES

1. Jay Freyensee

<mailto:james.p.freyensee@intel.com>

NVMe

06/23/2023

NVME-DISCOVER(1)