IBM Storage Ceph 7

NVMe-oF Gateway Support Guide
2023-11-27

IBM

Important

DRAFT BETA 1 PROVIDED WITH NO SUPPORT: Do not upgrade a production installation to a beta release. Upgrade support is not provided between beta releases. No automated upgrade path is provided from a beta release to the final GA release. The cluster will need to be rebuilt for production use.

Last updated: 2023-11-27

Contents

Important.. i

Figures...v

Tables.. vii

NVMe over Fabrics...9

Ceph NVMe-oF gateway (Technology Preview)...11

Installing the NVMe-oF gateway...13

Installing the Ceph NVMe-oF gateway using the command-line interface.....................13

Configuring the NVMe-oF gateway target...14

Defining an NVMe-oF subsystem...14

Defining block devices to use NVMe/TCP...16

Configuring the NVMe-oF gateway initiator..18

Configuring the NVMe-oF initiator for Red Hat Enterprise Linux...................................18

Configuring the NVMe-oF initiator for VMware ESXi..20

NVMe-oF Gateway performance best practices...22

Removing NVMe-oF service..23

Figures

1. Ceph NVMe-oF Gateway.. 12

Tables

1. Rule of thumb sizing guide....................22

NVMe over Fabrics

Ceph Block Device now offers NVMe-oF gateway support as a Technology Preview. Non-Volatile Memory express (NVMe) and NVMe over Fabrics (NVMe-oF) protocols are designed specifically to enable the full capabilities of parallelism and performance with all-flash storage systems.

Important: Technology Preview features are not supported with IBM production service level agreements (SLAs), might not be functionally complete, and IBM does not recommend using them for production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

NVMe-oF enables the performance of direct attached all-flash storage along with the flexibility and total cost of ownership (TCO) savings of shared storage. Implementing NVMe-oF support in enterprise storage arrays allows the combination of the NVMe protocol performance with the rich feature set of modern storage arrays. Using NVMe protocol enables modern storage arrays to meet growing customer demands.

NVMe-oF uses the TCP protocol as a ubiquitous transport that does not require special network configuration, by using existing Ethernet connectivity. Ethernet speeds currently support up to 400 Gib per second and the use of Ethernet technology is significant in data centers.

For more information about Ceph Block Device NVMe-oF gateway support, see Ceph NVMe-oF gateway.

Ceph NVMe-oF gateway (Technology Preview)

Storage administrators can install and configure an NVMe over Fabrics (NVMe-oF) gateway for an IBM Storage Ceph cluster. With the Ceph NVMe-oF gateway, you can effectively run a fully integrated block storage infrastructure with all features and benefits of a conventional Storage Area Network (SAN).

Important: Technology Preview features are not supported with IBM production service level agreements (SLAs), might not be functionally complete, and IBM does not recommend using them for production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

Traditionally, block-level access to a Ceph storage cluster has been limited to QEMU and `librbd`, which is a key enabler for adoption within OpenStack environments. Starting from IBM Storage Ceph 7, block-level access to the Ceph storage cluster can take advantage of the NVMe-oF standard to provide data storage.

The NVMe-oF gateway integrates IBM Storage Ceph with the NVMe over TCP (NVMe/TCP) protocol to provide an NVMe/TCP target that exports RADOS Block Device (RBD) images. The NVMe/TCP protocol allows clients, which are known as initiators, to send NVMe-oF commands to storage devices, which are known as targets, over an Internet Protocol network.

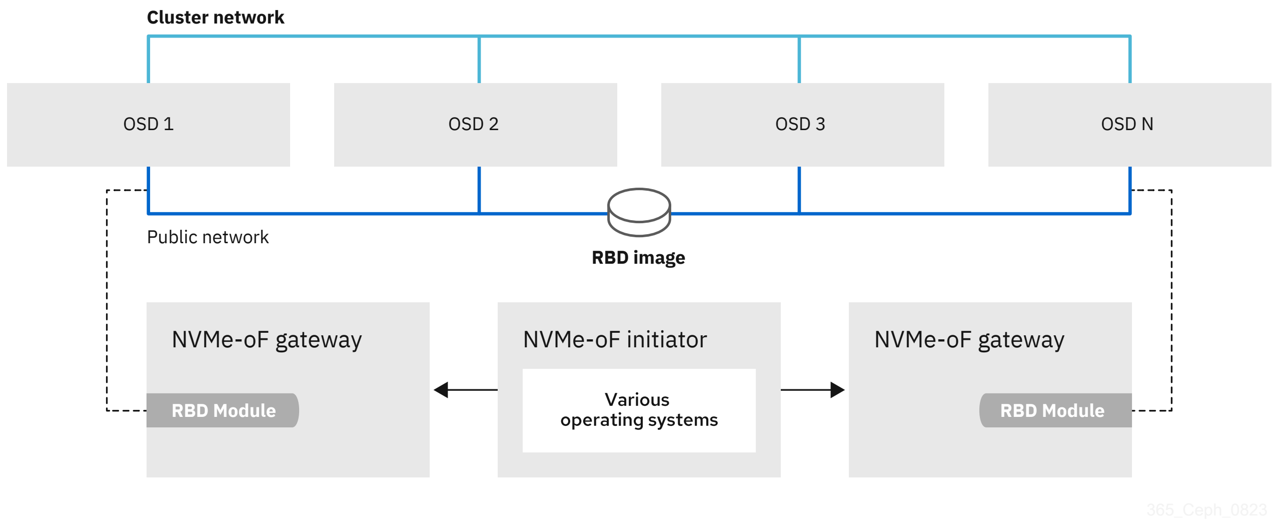

Figure 1. Ceph NVMe-oF Gateway

For more information about the NVMe over Fabrics protocol, see NVMe over Fabrics.

Installing the NVMe-oF gateway

Before you can utilize the benefits of the Ceph NVMe-oF gateway, you must install the required software packages. You can install the Ceph NVMe-oF gateway by using the command-line interface.

Each NVMe-oF gateway runs the Storage Performance Development Kit (SPDK) to provide NVMe-oF protocol support. SPDK utilizes a user-space implementation of the NVMe-oF protocol to interact with the Ceph `librbd` library to expose RBD images to NVMe-oF clients. With the Ceph NVMe-oF gateway you can effectively run a fully integrated block storage infrastructure with all features and benefits of a conventional Storage Area Network (SAN).

Installing the Ceph NVMe-oF gateway using the command-line interface

The Ceph NVMe-oF gateway is the NVMe-oF target node and also a Ceph client node. The Ceph NVMe-oF gateway can be a stand-alone node or be collocated on a Ceph Object Storage Daemon (OSD) node.

Before you begin

Installing the NVMe-oF gateway requires a storage system that is running IBM Storage Ceph 7 or later cluster.

Procedure

Complete the following steps to install the Ceph NVMe-oF gateway.

1. Create a pool in which the gateway group configuration is to be managed.

   ```
   ceph osd pool create NVME-OF_POOL_NAME
   ```

   For example,

   ```
   [root@host01 ~]# ceph osd pool create nvmeof_pool01
   ```

2. Enable the RADOS Block Device (RBD) application on the NVMe-oF pool.

   ```
   rbd pool init NVME-OF_POOL_NAME
   ```

   For example,

   ```
   [root@host01 ~]# rbd pool init nvmeof_pool01
   ```

3. Set NVMe-oF container image before service deployment.

   ```
   [ceph: root@host01 /]# ceph config set mgr mgr/cephadm/container_image_nvmeof icr.io/ibm-
   ceph-beta/nvmeof-rhel9:latest
   ```

4. Deploy the `nvmeof` manager daemons that use placement specification on a specific set of nodes.

   ```
   ceph orch apply nvmeof NVME-OF_POOL_NAME --placement="NODE1, NODE2,..."
   ```

   For example,

   ```
   [ceph: root@host01 /]# ceph orch apply nvmeof nvmeof_pool01 --placement="host01, host02"
   ```

   Note: To remove a node from a daemon, redeploy the `nvmeof` manager daemon on the wanted nodes.

5. You can install NVMe manager daemons using the service specification file with the `cpumask` option. By using this option, you can increase the number of cores, thereby improving the NVMe performance.

   a. Create a specification file with the following details.

```
service_type: nvmeof
service_id: NVME-OF_POOL_NAME
placement:
  hosts:
    - HOST_NAME
spec:
  pool: NVME-OF_POOL_NAME
  tgt_cmd_extra_args: --cpumask=0xFF
```

Where FF is a hexadecimal bit mask based on the number of cores that are used in the cluster.

For example,

```
cat file.yaml
service_type: nvmeof
service_id: nvmeof_pool01
placement:
  hosts:
    - host01
    - host02
spec:
  pool: nvmeof_pool01
  tgt_cmd_extra_args: --cpumask=0xFF
```

In this example, since 8 cores are used, the value of `--cpumask` is `0xFF`. For 6 cores, the value is `0x3F`.

b. Deploy the NVMe daemons on the admin node.

```
ceph orch apply -i NVME_FILE
```

Configuring the NVMe-oF gateway target

Installing the Ceph NVMe-oF gateway using the command-line interfaceConfigure targets, LUNs, and clients, using the Ceph gateway **`nvmeof-cli`** command-line utility.

Configuring the NVMe-oF gateway target requires a storage system running IBM Storage Ceph 7 or later cluster and a running Ceph NVMe-oF gateway.

Defining an NVMe-oF subsystem

Define an NVMe-oF subsystem. Defining the NVMe subsystem includes creating an NVMe subsystem, configuring the IP port for communication, and enabling hosts to use the subsystem.

About this task

Configure the Ceph NVMe-oF gateway by using the gateway `nvmeof-cli` utility.

Before you begin

The gateway `nvmeof-cli` container is automatically downloaded during the subsystem creation. Alternatively, the `nvmeof-cli` container can be downloaded before first use.

To download the gateway `nvmeof-cli` container, use the following command:

```
podman pull icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest
```

Procedure

These commands can be run on either podman or `docker`.

1. Create an NVMe subsystem.

Important: Run this command only *once*. Only one NVMe subsystem is supported.

```
podman run -it <icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest> --server-address GATEWAY_IP --
server-port 5500 create_subsystem --subnqn SUBSYSTEM_NQN --max-namespaces 256
```

The *SUBSYSTEM_NQN* is a user defined string. In this example it is defined as `nqn.2016-06.io.spdk:cnode1`.

For example,

```
[root@host01 ~]# podman run -it <icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest> --server-
address 10.172.19.21 --server-port 5500 create_subsystem --subnqn nqn.2016-06.io.spdk:cnode1
--max-namespaces 256
```

Note: For additional input, `--serial` *SERIAL_NUMBER* can be added to the command.

2. Define the IP port on the gateway that is to process NVMe/TCP commands and I/O operations.
   a) From the installer node, get the NVMe-oF gateway name.

   ```
   ceph orch ps | grep nvme
   ```

   For example,

   ```
   [ceph: root@host01 /]# ceph orch ps | grep nvme

   nvmeof.rbd.host010.nl host010*:5500,4420,8009
   running (13d) - 13d 1970M - 470dd4ee78f0 e313e8b13a31
   ```

   In this example, the gateway name is `client.nvmeof.rbd.host010.nl`.

   b) Define the IP port on the gateway.

   ```
   podman run -it icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest --server-address GATEWAY_IP
   --server-port 5500 create_listener -n SUBSYSTEM_NQN -g GATEWAY_NAME -a GATEWAY_NODE_IP -s
   4420
   ```

   For example,

   ```
   [root@host01 ~]# podman run -it icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest --server-
   address 10.172.19.01 --server-port 5500 create_listener -n nqn.2016-06.io.spdk:cnode1 -g
   client.nvmeof.test-nvmeof-pool.rhel9-ceph-node1.ogqdhx -a 10.172.19.01 -s 4420
   ```

3. Get the host NVMe Qualified Name (NQN) for each host.

   For Red Hat Enterprise Linux initiators

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

   For example,

   ```
   # cat /etc/nvme/hostnqn
   nqn.2014-08.org.nvmexpress:uuid:950ddadf-f995-47b7-9416-b9bb233f66e3
   ```

   In this example `950ddadf-f995-47b7-9416-b9bb233f66e3` is the UUID.

   For VMware ESXi initiators

   ```
   esxcli nvme info get
   ```

   For example,

   ```
   esxcli nvme info get
   Host NQN: nqn.2014-08.com.ibm.ceph:nvme:host01
   ```

4. Allow the NVMe initiator host to run NVMe/TCP commands to the newly created NVMe subsystem.

Note: Specific hosts can be selected, as in the following example. To specify all hosts, use `--host "*"`.

```
podman run -it icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest --server-address NODE_IP --
server-port 5500 add_host --subnqn SUBSYSTEM_NQN --host "HOST01_NQN,
HOST02_NQN"
```

For example,

```
[root@host01 ~]# podman run -it icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest
--server-address 10.172.19.01 --server-port 5500 add_host --subnqn
nqn.2016-06.io.spdk:cnode1 --host "nqn.2014-08.org.nvmexpress:uuid:950ddadf-f995-47b7-9416-
b9bb233f66e3,nqn.2014-010.org.nvmexpress:uuid:960ddadf-f995-47b7-9416-b9bb233f66e3"
```

5. List all NVMe-OF Gateway target entities configured on Gateway.

```
podman run -it <icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest> --server-address GATEWAY_IP --
server-port 5500 get_subsystems
```

For example,

```
[root@host01 ~]# podman run -it <icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest> --server-
address 10.172.19.01 --server-port 5500 get_subsystems
INFO:__main__:Get subsystems:
[
    {
        "nqn": "nqn.2014-08.org.nvmexpress.discovery",
        "subtype": "Discovery",
        "listen_addresses": [],
        "allow_any_host": true,
        "hosts": []
    },
    {
        "nqn": "nqn.2016-06.io.spdk:cnode1",
        "subtype": "NVMe",
        "listen_addresses": [
            {
                "transport": "TCP",
                "trtype": "TCP",
                "adrfam": "IPv4",
                "traddr": "10.172.19.01",
                "trsvcid": "4420"
            }
        ],
        "allow_any_host": true,
        "hosts": [],
        "serial_number": "1",
        "model_number": "SPDK bdev Controller",
        "max_namespaces": 256,
        "min_cntlid": 1,
        "max_cntlid": 65519,
        "namespaces": []
    }
]
```

Defining block devices to use NVMe/TCP

Define storage block devices to be used with NVMe/TCP.

About this task

Note: This procedure only needs to be run once per block device, even if there are multiple gateways.

Procedure

1. Create an image within any RBD application-enabled pool.

```
rbd create IMAGE_NAME --size MEGABYTES --pool POOL_NAME
```

For example,

```
[root@host01 ~]# rbd create image-1 --size 1024 --pool pool01
```

For more information about creating an image, see Creating images.

2. Using the gateway `nvmeof-cli` utility, create a block device.

   Run this command for every RBD image that needs to be exposed through NVMe/TCP.

   ```
   podman run -it <icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest> --server-address GATEWAY_IP --
   server-port 5500 create_bdev --pool POOL_NAME --image IMAGE_NAME --bdev BLOCK_DEVICE_NAME
   ```

   For example,

   ```
   [root@host01 ~]# podman run -it <icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest> --server-
   address 10.172.19.21 --server-port 5500 create_bdev --pool pool01 --image image-01 --bdev ceph-
   block01
   ```

3. Add a namespace to each block device that was created in “Defining an NVMe-oF subsystem” on page 14.

   ```
   podman run -it <icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest> --server-address GATEWAY_IP --
   server-port 5500 add_namespace --subnqn SUBSYSTEM_NQN --bdev BLOCK_DEVICE_NAME
   ```

   For example,

   ```
   [root@host01 ~]# podman run -it <icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest> --server-
   address 10.172.19.21 --server-port 5500 add_namespace --subnqn nqn.2016-06.io.spdk:cnode1 --
   bdev ceph-block01
   ```

4. List all NVMe-OF Gateway target entities configured on Gateway.

   ```
   podman run -it <icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest> --server-address GATEWAY_IP --
   server-port 5500 get_subsystems
   ```

   For example,

```
[root@host01 ~]# podman run -it <icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest> --server-
address 10.172.19.01 --server-port 5500 get_subsystems
INFO:__main__:Get subsystems:
[
    {
        "nqn": "nqn.2014-08.org.nvmexpress.discovery",
        "subtype": "Discovery",
        "listen_addresses": [],
        "allow_any_host": true,
        "hosts": []
    },
    {
        "nqn": "nqn.2016-06.io.spdk:cnode1",
        "subtype": "NVMe",
        "listen_addresses": [
            {
                "transport": "TCP",
                "trtype": "TCP",
                "adrfam": "IPv4",
                "traddr": "10.172.19.01",
                "trsvcid": "4420"
            }
        ],
        "allow_any_host": true,
        "hosts": [],
        "serial_number": "1",
        "model_number": "SPDK bdev Controller",
        "max_namespaces": 256,
        "min_cntlid": 1,
        "max_cntlid": 65519,
        "namespaces": [
            {
                "nsid": 1,
                "bdev_name": "ceph-block01",
                "name": "ceph-block01",
                "nguid": "637DBF16AE6C47CF850A81F317E3BFEF",
                "uuid": "637dbf16-ae6c-47cf-850a-81f317e3bfef"
            },

        ]
    }
]
```

Configuring the NVMe-oF gateway initiator

Configure the initiator to allow the NVMe/TCP protocol to send NVMe-oF commands to targets over an Internet Protocol network.

The NVMe-oF gateway initiator can be configured on either of the following platform version:

- Red Hat Enterprise Linux 9.2 or later
- VMware vSphere Hypervisor (ESXi) 7.0U3 or later

Configuring the NVMe-oF initiator for Red Hat Enterprise Linux

Configure the NVMe-oF initiator for Red Hat Enterprise Linux.

Before you begin

- Red Hat Enterprise Linux 9.2 or later

Procedure

1. Install the `nvme-cli`.

   ```
   yum install nvme-cli
   ```

2. Install the necessary NVMe packages.

   ```
   modprobe nvme-fabrics
   ```

3. Verify that the target is reachable from the initiator.

   ```
   nvme discover -t tcp -a GATEWAY_IP -s 4420
   ```

   For example,

   ```
   [root@host01 ~]# nvme discover -t tcp -a 10.172.19.01 -s 4420
   ```

4. Connect to the NVMe-oF target.

   Use the *SUBSYSTEM_NQN* that was defined in "Defining an NVMe-oF subsystem" on page 14.

   ```
   nvme connect -t tcp -a GATEWAY_IP -n SUBSYSTEM_NQN
   ```

   For example,

   ```
   [root@host01 ~]# nvme connect -t tcp -a 10.172.19.01 -n nqn.2016-06.io.spdk:cnode1
   ```

What to do next

Verify that the initiator is set up correctly.

1. List the NVMe-oF block devices.

   ```
   nvme list
   ```

   For example,

   ```
   [root@host01 ~]# nvme list
   Node                   Generic            SN                   Model
   Namespace Usage                      Format           FW Rev
   ---------------------   ----------------  -------------------  -----------------------
   --------- -------------------------- ---------------- --------
   /home/nvme01_node01     /home/ng1n1       SPDK00000000000001   SPDK bdev Controller
   1          10,49  MB /  10,49  MB      4 KiB +  0 B   23.01
   ...
   ```

2. Create a filesystem on the desired target, found in step "1" on page 19.

   ```
   mkfs NVME_NODE_PATH
   ```

   For example,

   ```
   [root@host01 ~]# mkfs /home/nvme01_node01
   mke2fs 1.46.5 (20-Dec-2023)
   Discarding device blocks: done
   Creating filesystem with 2560 4k blocks and 2560 inodes

   Allocating group tables: done
   Writing inode tables: done
   Writing superblocks and filesystem accounting information: done
   ```

3. Mount the NVMe node on the NVMe-oF directory.

   a. Mount NVMe-oF.

      ```
      mkdir /mnt/nvmeof
      ```

      For example,

      ```
      [root@host01 ~]# mkdir /mnt/nvmeof
      ```

   b. Mount the node on within the NVMe-oF directory.

      ```
      mount NVME_NODE_PATH /mnt/nvmeof
      ```

      For example,

```
[root@host01 ~]# mount /home/nvme01_node01 /mnt/nvmeof
```

4. Using sudo commands, list mounted NVMe-oF files.

   ```
   ls /mnt/nvmeof
   ```

   For example,

   ```
   $ ls /mnt/nvmeof
   lost+found
   ```

5. Create a text file within the `mnt/nvmeof` directory.

   For example,

   ```
   $ sudo bash -c "echo Hello NVMe-oF > /mnt/nvmeof/hello.txt"
   ```

6. Verify that the text file can now be reached.

   For example,

   ```
   $ cat /mnt/nvmeof/hello.txt
   Hello NVMe-oF
   ```

Configuring the NVMe-oF initiator for VMware ESXi

Configure the NVMe-oF initiator for VMware vSphere Hypervisor (ESXi). You can set up a VMware ESXi host as a NVMe/TCP initiator.

About this task

NVMe/TCP is supported by VMware vSphere Hypervisor (ESXi) 7.0U3 or later.

Before you begin

- A VMware ESXi host running VMware vSphere Hypervisor (ESXi) 7.0U3 version or later.
- Ceph NVMe-oF gateway deployed.
- IBM Storage Ceph cluster and `ceph-nvmeof` configuration is ready and healthy.
- A subsystem defined within the gateway. For more information, see "Defining an NVMe-oF subsystem" on page 14.
- NVMe/TCP adapter is configured.
  - Enabled NVMe/TCP on a physical network interface controller (NIC).

    ```
    esxcli nvme fabrics enable --protocol TCP --device vmnicN
    ```

    Replace *N* with the number of NIC.
  - Tag a VMkernel NIC to permit NVMe/TCP traffic.

    ```
    esxcli network ip interface tag add --interface-name vmkN --tagname NVMeTCP
    ```

    Replace *N* with the ID of the VMkernel.

Procedure

Configuring the VMware ESXi host for NVMe/TCP transport includes discovering the NVMe/TCP targets and connecting to them.

1. List the NVMe-oF adapter.

   ```
   esxcli nvme adapter list
   ```

For example,

```
[root@host01:~] esxcli nvme adapter list
Adapter  Adapter Qualified Name           Transport Type  Driver     Associated Devices
-------  -------------------------------- --------------  ---------  ------------------
vmhba64  aqn:nvmetcp:ac-1f-6b-0a-18-74-T  TCP             nvmetcp    vmnic0
```

2. Discover any NVMe-oF-gateway subsystems.

```
esxcli nvme fabrics discover -a NVME_TCP_ADAPTER -i GATEWAY_IP -p 4420
```

For example,

```
[root@host01:~] esxcli nvme fabrics discover -a vmhba64 -i 10.0.211.196 -p 4420

Transport Type Address Family Subsystem Type Controller ID Admin Queue Max Size Transport
Address Transport Service ID Subsystem NQN             Connected
-------------- -------------- -------------- ------------- --------------------
----------------- -------------------- ------------------------- ---------
TCP            IPv4           NVM            65535         128
10.0.211.196     4420                nqn.2016-06.io.spdk:cnode1  false
```

3. Connect to NVMe-oF gateway subsystem.

```
esxcli nvme fabrics connect -a NVME_TCP_ADAPTER -i GATEWAY_IP -p 4420 -s SUBSYSTEM_NQN
```

For example,

```
[root@host01:~] esxcli nvme fabrics connect -a vmhba64 -i 10.0.211.196 -p 4420 -s
nqn.2016-06.io.spdk:cnode1
```

4. List NVMe/TCP controller list.

```
esxcli nvme controller list
```

```
[root@host01:~] esxcli nvme controller list
Name
Controller Number  Adapter  Transport Type  Is Online
------------------------------------------------------------------------------------------
-----------------  -------  --------------  ---------
nqn.2016-06.io.spdk:cnode1#vmhba64#10.0.211.196:4420
           301  vmhba64  TCP                  true
```

5. List NVMe-oF namespaces in the subsystem.

```
esxcli nvme namespace list
```

For example,

```
[root@host01:~] esxcli nvme namespace list
Name                                Controller Number  Namespace ID  Block Size  Capacity in MB
------------------                  -----------------  ------------  ----------  --------------
eui.0100000001000000e4d25c00001ae214               256            1         512
953869
eui.01abc123def456g7e4d25c00001ae214               301            1         512
500
eui.02abc123def456g7e4d25c00001ae215               301            2         512
500
eui.03abc123def456g7e4d25c00001ae216               301            3         512
500
```

What to do next

Verify that the initiator is set up correctly.

1. From the vSphere Client, go to the ESXi host.
2. On the **Storage** page, go to the **Devices** tab.
3. Verify that the NVMe/TCP disks are listed in the table.

NVMe-oF Gateway performance best practices

For high IOPS requirements, use a dedicated host for the NVMe-oF Gateway.

1. Colocation with OSDs is possible, but the high CPU demand from the gateway might introduce CPU contention especially for write client workload and when the node has multiples OSDs.
2. For dedicated gateways, the gateway needs a minimum of 16 cores. For higher performance requirements, IBM recommends 30 or more cores.

Rule of thumb (RoT) sizing guide

The following RoT sizing is based on the mixed workload scenario at latencies <= 10ms, with an environment based on 100Gb networking, with 72 namespaces.

Table 1. Rule of thumb sizing guide

IOPS Goal	Number of reactor cores	Total CPU demand (cores)	`tgt_cmd_extra_rags` specification parameter
Up to 100K	1	16	`--cpumask=0x1`
Up to 200K	2	18	`--cpumask=0x3`
Up to 250K	4	22	`--cpumask=0xF` (default)
Up to 300K	8	30	`--cpumask=0xFF`

Modify the cores

To modify the cores that are used by Storage Performance Development Kit (SPDK) to handle the I/O, follow this procedure:

1. Fetch the specification file of the cluster.

```
ceph orch ls --export > FILE.yaml
```

2. Modify the file to include or modify the **`tgt_cmd_extra_args`** parameter. For example, change from the default **`tgt_cmd_extra_args: --cpumask=0xF`** to **`tgt_cmd_extra_args: --cpumask=0xFF`**.

```
service_type: nvmeof
service_id: NVME-OF_POOL_NAME
placement:
  hosts:
    - HOST_NAME
spec:
  pool: NVME-OF_POOL_NAME
  tgt_cmd_extra_args: --cpumask=0xFF
```

For example,

```
[ceph: root@host01/ ]# cat file.yaml
service_type: nvmeof
service_id: nvmeof_pool01
placement:
  hosts:
    - host01
    - host02
spec:
  pool: nvmeof_pool01
  tgt_cmd_extra_args: --cpumask=0xFF
```

3. Redeploy the service.

```
ceph orch apply -i FILE.yaml
```

4. Reconfigure the NVMe-oF gateway.

   ```
   ceph orch reconfig NVME-OF_SERVICE_ID
   ```

   For example,

   ```
   [ceph: root@host01/ ]# ceph orch reconfig nvmeof_pool01
   ```

VMware ESX recommendation

1. Allocate more smaller data stores instead of a few large data stores. This approach does not carry capacity cost as Ceph has thin provisions by default, and ensures that me Ceph I/O contexts are available for the gateway.
2. Consider isolating applications with high I/O demands to use a dedicated datastore to ensure adequate host adapter queue depth.
3. Use jumbo frames when possible for ESXi to Gateway network links. This is beneficial for workloads that use larger I/O block sizes.

Removing NVMe-oF service

Remove NVMe-oF service.

Before you begin

A running IBM Storage Ceph cluster with NVMe-oF gateway configured on the system.

Note: Ensure that the volumes are stopped before the service is removed.

About this task

The following are the three main steps to remove the NVMe-oF service:

1. Disconnect the NVMe-oF Gateway initiator.
2. Remove the NVMe-oF Gateway target.
3. Remove the NVMe-oF service.

Procedure

1. Disconnect the NVMe-oF initiator from NVMe-oF target.

   Red Hat Enterprise Linux

   ```
   nvme disconnect-all
   ```

   VMware esx initiator

   ```
   esxcli nvme disconnect -n CONTROLLER_NUMBER -a NVME_TCP_ADAPTER -s SUBSYSTEM_NQN
   ```

   where *CONTROLLER_NUMBER* is the number that is fetched from **`esxcli nvme namespace list`** command, *NVME_TCP_ADAPTER* and *SUBSYSTEM_NQN* are values from the **`esxcli nvme fabrics connect`** command.

2. Remove NVMe-oF Gateway target.

   a) Remove existing namespace from subsystem.

   ```
   podman run -it <icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest> --server-address GATEWAY_IP
   --server-port 5500 remove_namespace -n SUBSYSTEM_NQN -i NSID
   ```

For example,

```
[root@host01 ~]# podman run -it <icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest> --server-
address 10.172.19.01 --server-port 5500 remove_namespace -n nqn.2016-06.io.spdk:cnode1 -i 1
```

b) Delete `bdevs` created on target.

```
podman run -it <icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest> --server-address GATEWAY_IP
--server-port 5500 delete_bdev -b BLOCK_DEVICE_NAME
```

For example,

```
[root@host01 ~]# podman run -it <icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest> --server-
address 10.172.19.01 --server-port 5500 delete_bdev -b ceph-block01
```

c) Delete subsystem created on target.

```
podman run -it <icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest> --server-address GATEWAY_IP
--server-port 5500 delete_subsystem -n SUBSYSTEM_NQN
```

For example,

```
[root@host01 ~]# podman run -it <icr.io/ibm-ceph-beta/nvmeof-cli-rhel9:latest> --server-
address 10.172.19.01 --server-port 5500 delete_subsystem -n nqn.2016-06.io.spdk:cnode1
```

3. Remove NVMe-oF service.

a) On the Ceph admin node, remove the NVMe-oF service.

```
ceph orch rm nvmeof.NVME-OF_POOL_NAME
```

For example,

```
[ceph: root@host01 /]# ceph orch rm nvmeof.nvmeof_pool01
```

b) Ensure that the **`mon_allow_pool_delete`** parameter is set to `true`

```
ceph config set mon mon_allow_pool_delete true
```

c) On the Ceph admin node, remove the *NVME-OF_POOL* created to host NVMe-oF service.

```
ceph osd pool rm NVME-OF_POOL_NAME NVME-OF_POOL_NAME --yes-i-really-really-mean-it
```

Note: You need to use the pool name *NVME-OF_POOL_NAME* twice in the command to remove the pool.

For example,

```
[ceph: root@host01 /]# ceph osd pool rm nvmeof_pool01 nvmeof_pool01 --yes-i-really-really-
mean-it
```