NVMe protocol

System Manager Classic

NetApp
December 09, 2021

Table of Contents

NVMe protocol

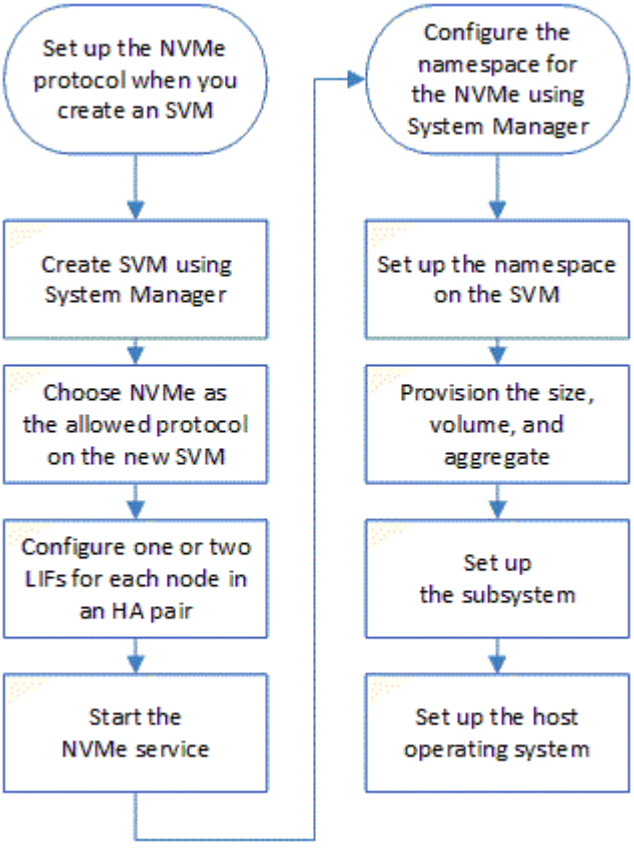
You can use System Manager to configure the NVMe protocol. The NVMe is a transport protocol that provides high speed access to flash-based network storage. Systems that use NVMe protocol have a subsystem consisting of specific NVME controllers, namespaces, nonvolatile storage medium, hosts, ports and interface between the controller and storage medium.

Setting up NVMe

You can set up the NVMe protocol for an SVM using System Manager. When the NVMe protocol is enabled on the SVM, you can then provision a namespace or namespaces and assign them to a host and a subsystem.

Starting with ONTAP 9.5, you must configure at least one NVMe LIF for each node in an HA pair that uses the NVMe protocol. You can also define a maximum of two NVMe LIFs per node. You configure the NVMe LIFs when you create or edit the SVM settings using System Manager.

The following illustration shows the workflow for setting up NVMe:



Create an NVMe namespace

You can use System Manager to create one or more NVMe namespaces and connect each to a host or set of hosts in a storage virtual machine (SVM). The NVMe namespace

is a quantity of memory that can be formatted into logical blocks. Each namespace can be mapped to an NVMe subsystem.

Before you begin

The SVM must already be configured with the NVMe protocol. To map a namespace, at least one LIF with the data protocol NVMe must exist in the node that owns the namespace.

Steps

1. Click **Storage** > **NVMe** > **NVMe namespaces**.
2. Select the SVM that will contain the namespace.
3. Ensure that at least one NVMe LIF is configured for each node of the HA pair. You can create a maximum of two NVMe LIFs per node.
4. Configure the size of the namespace (between 1MB and 16TB).
5. Enter the block size.

   For System Manager 9.5, the block size defaults to 4 KB, and this field is not shown.

   For System Manager 9.6, you can specify a block size of 4 KB or 512 Bytes.
6. Select the existing volume or create a new volume by choosing the aggregate.

   Click on the + symbol to set up additional namespaces (max 250) within the SVM.
7. Select the NVMe subsystem that will be associated with this namespace.

   You can choose from the following options:

   - None: No subsystems are mapped.
   - Use an existing subsystem: The subsystems listed are based on the selected SVM.
   - Create a new subsystem: You can choose to create a new subsystem and map to all the new namespaces.
8. Select the host operating system.
9. Click **Submit**.

Related information

NVMe namespaces window

Editing an NVMe namespace

You can use System Manager to edit the namespace by changing the subsystem that the namespace is mapped to.

About this task

You can only modify the NVMe subsystem settings in this window, you cannot edit the other namespace details.

Steps

1. Click **NVMe** > **NVMe namespaces**.

2. In the **NVMe namespaces window**, select the namespace you want to edit.
3. Select a subsystem option:
    - None: Choosing this option unmaps the existing subsystem mapping for this namespace only. This option is preselected if no subsystem mapping is present for the selected namespace.
    - Use an existing subsystem: This option is preselected if subsystem-to-namespace mapping is present. Choosing a different subsystem maps the new subsystem by unmapping the previously mapped subsystem.

Cloning an NVMe namespace

You can use System Manager to quickly create another namespace of the same configuration by choosing to clone a namespace. You can map the newly cloned namespace to another host NQN.

Before you begin

You must have a FlexClone license to clone a namespace.

About this task

You can clone a namespace with the selected host mapping and associate it with another subsystem.

Steps

1. Click **NVMe** > **NVMe namespaces**.
2. In the **NVMe namespaces window**, select the namespace you want to clone.
3. You can rename the cloned namespace if you need a specific name but it is not required.

   The dialog provides a default name of the namespace to-be-cloned.
4. Modify the subsystem mapping for the cloned namespace.
5. Click **OK**.

   The online, mapped namespace is cloned inside the same SVM with a different name. Host mapping will not be cloned.

Starting and stopping the NVMe service

The NVMe service enables you to manage NVMe adapters for use with namespaces. You can use System Manager to start the NVMe service to bring the adapters online. You can stop the NVMe service to take the NVMe adapters offline and to disable access to the namespaces.

Before you begin

NVMe capable adapters must be present before you start the NVMe service.

Steps

1. Click **Storage** > **SVMs**.
2. Select the SVM, and then click **SVM settings**.

3. In the **Protocols** menu, click **NVMe**.
4. Click **Start** or **Stop** service as required.

What NVMe is

The nonvolatile memory express (NVMe) protocol is a transport protocol used for accessing nonvolatile storage media.

NVMe over Fabrics (NVMeoF) is a specification-defined extension to NVMe that enables NVMe-based communication over connections other than PCIe. This interface allows for external storage enclosures to be connected to a server.

NVMe is designed to provide efficient access to storage devices built with non-volatile memory, from flash technology to higher performing, persistent memory technologies. As such, it does not have the same limitations as storage protocols designed for hard disk drives. Flash and solid state devices (SSDs) are a type of non-volatile memory (NVM). NVM is a type of memory that keeps its content during a power outage. NVMe is a way that you can access that memory.

The benefits of NVMe include increased speeds, productivity, throughput, and capacity for data transfer. Specific characteristics include the following:

- NVMe is designed to have up to 64 thousand queues.

  Each queue in turn can have up to 64 thousand concurrent commands.

- NVMe is supported by multiple hardware and software vendors
- NMVe is more productive with Flash technologies enabling faster response times
- NVMe allows for multiple data requests for each “request” sent to the SSD.

  NVMe takes less time to decode a “request” and does not require thread locking in a multithreaded program.

- NVMe supports functionality that prevents bottlenecking at the CPU level and enables massive scalability as systems expand

What an NVMe subsystem is

An NVMe subsystem includes one or more controllers, one or more namespaces, one or more non-volatile memory (NVM) subsystem ports (FC-NVMe or RDMA transport ports), an NVM storage medium, and an interface between the controllers and the NVM storage medium. For controller mapping and management, an NVM subsystem maps to a vserver in ONTAP.

An NVMe subsystem can be created using System Manager. You can associate the NVMe subsystem with different hosts and namespaces within the vserver. Also, each vserver can support more than one NVMe subsystem. However, you cannot configure a NVMe subsystem to be used on multiple vservers.

An NVMe over Fabric (NVMeoF) subsystem is a separate kernel object that resides in the FreeBSD kernel. The NVMeoF subsystem interfaces with the following components:

- SAN components, such as BCOMKA, FCT, and VDOM
- WAFL
- RAS components, such as CM, ASUP, and EMS

All interfaces with NVMeoF subsystems adhere to the current definitions and patterns found in ONTAP.

Create NVMe subsystems

You can use System Manager to create an NVMe subsystem.

Steps

1. Click **Create** in the **NVMe Subsystems** window.
2. Provide entries in the **NVMe Subsystems: Create** window for the following fields:
   - **SVM**

     From the drop-down menu, select the SVM on which you want to create the subsystem.
   - **Name**

     Enter a name for the subsystem. The subsystem name cannot already exist in the SVM. The name is case-sensitive and is limited to 96 characters. Special characters are allowed.
   - **Host OS**

     From the drop-down menu, select the type of Host OS of the subsystem.
   - **Host NQN**

     Enter the Host NQN attached to the controller. You can enter more than one Host NQN by separating them with commas.
3. Click **Save**.

   The NVMe subsystem is created, and the NVMe Subsystemswindow is displayed.

Related information

NVMe Subsystems window

Editing NVMe subsystems details

You can use System Manager to edit the details of an NVMe subsystem.

Steps

1. Find the NVMe subsystem you want to edit in the **NVMe Subsystem** window.
2. Check the box to the left of the name of the subsystem you want to edit.
3. Click **Edit**.

   The current details of the NVMe subsystem are displayed in the NVMe Subsystems: Editwindow.
4. You can modify only the information in the **Host NQN** field.

- **Host NQN**

  Modify the Host NQN attached to the controller. You can enter more than one Host NQN by separating them with commas.

The **Associated NVMe Namespaces** table displays below the Host NQN field. For each namespace, that table lists the namespace path and namespace ID.

5. Click **Save**.

   The NVMe subsystem details are updated, and the NVMe Subsystems window is displayed.

Related information

NVMe Subsystems window

Deleting an NVMe subsystem

You can use System Manager to delete an NVMe subsystem from a cluster.

About this task

The following actions occur when you delete an NVMe subsystem:

- If the NVMe subsystem has configured hosts, then mapped hosts will be removed.
- If the NVMe subsystem has mapped namespaces, then they will be unmapped.

Steps

1. Find the NVMe subsystem you want to delete on the **NVMe Subsystem** window.
2. Check the box to the left of the name of the subsystem you want to delete.
3. Click **Delete**.

   A Warning message is displayed.

4. Click the **Delete the NVMe Subsystem** check box to confirm the deletion, then click **Yes**.

   The NVMe subsystem is deleted from the cluster, and the NVMe Subsystems window is displayed.

Related information

NVMe Subsystems window

NVMe Subsystems window

The NVMe Subsystems window displays by default an inventory list of NVMe subsystems in a cluster. You can filter the list to display only subsystems that are specific to an SVM. The window also enables you to create, edit, or delete NVMe subsystems. You can access this window by selecting **Storage** > **NVMe** > **Subsystems**.

- NVMe Subsystems table
- Toolbar

NVMe Subsystems table

The NVMe Subsystems table lists the inventory of NVMe subsystems in a cluster. You can refine the list by using the drop-down menu in the **SVM** field to select an SVM to display only the NVMe subsystems associated with that SVM. The **Search** field and **Filtering** drop-down menu enable you to further customize the list.

The NVMe Subsystems table contains the following columns:

- **(check box)**

  Enables you to specify on which subsystems you want to perform actions.

  Click the check box to select the subsystem, then click the action in the toolbar that you want to perform.

- **Name**

  Displays the name of the subsystem.

  You can search for a subsystem by entering its name in the **Search** field.

- **Host OS**

  Displays the name of the host OS associated with the subsystem.

- **Host NQN**

  Displays the NVMe Qualified Name (NQN) attached to the controller. If multiple NQNs are displayed, they are separated by commas.

- **Associated NVMe Namespaces**

  Displays the number of the NVM namespaces associated with the subsystem. You can hover over the number to display the associated namespaces paths. Click on a path to display the Namespace Details window.

Toolbar

The toolbar is located above the column header. You can use the fields and buttons in the toolbar to perform various actions.

- **Search**

  Enables you to search on values that might be found in the **Name** column.

- **Filtering**

  Allows you to select from a drop-down menu that lists various methods of filtering the list.

- **Create**

  Opens the Create NVMe Subsystem dialog box, which enables you to create an NVMe subsystem.

- **Edit**

  Opens the Edit NVMe Subsystem dialog box, which enables you to edit an existing NVMe subsystem.

- **Delete**

  Opens the Delete NVMe Subsystem confirmation dialog box, which enables you to delete an existing NVMe subsystem.

NVMe namespaces

An NVMe namespace is a quantity of non-volatile memory (NVM) that can be formatted into logical blocks. Namespaces are used when a storage virtual machine is configured with the NVMe protocol and are the equivalent of LUNs for FC and iSCSI protocols.

One or more namespaces are provisioned and connected to an NVMe host. Each namespace can support various block sizes.

The NVMe protocol provides access to namespaces through multiple controllers. Using NVMe drivers, which are supported on most operating systems, solid state drive (SSD) namespaces appear as standard-block devices on which file systems and applications can be deployed without any modification.

A namespace ID (NSID) is an identifier used by a controller to provide access to a namespace. When setting the NSID for a host or host group, you also configure the accessibility to a volume by a host. A logical block can only be mapped to a single host group at a time, and a given host group does not have any duplicate NSIDs.

NVMe subsystem provisioning for NVMe namespaces

An NVMe subsystem includes one or more NVMe controllers, namespaces, NVM subsystem ports, an NVM storage medium, and an interface between the controller and the NVM storage medium. When you create an NVMe namespace, you can choose to map an NVMe subsystem to the namespace, as follows:

- **None (default)**

  No NVMe subsystems are mapped to the namespace.

- **Existing subsystem**

  You can select an existing NVMe subsystem to map to the namespace. NVMe subsystems are listed based on the host OS and SVM fields. When you hover the pointer over the NVMe subsystem name, more details are shown about the subsystem.

- **New subsystem**

  You can create a new NVMe subsystem and map it to the namespace. The subsystem is created on the host OS and SVM.

You provision a subsystem by providing the following details:

- **The NVMe subsystem name**

  The NVMe subsystem name is case sensitive. It must contain 1 to 96 characters, and special characters are allowed.

- **Host OS**

The host OS type that the subsystem is being created on.

- **Host NQN**

  The host NVMe qualification name attached to the controller. This column can contain comma-separated values because there can be from one to many hosts attached to a subsystem.

NVMe namespaces window

You can use the NVMe namepaces window to set up and manage your namespaces and associated subsytems for the NVMe protocol. You can search for an existing namespace using the namespace path.

Command Buttons

- **Create**

  Opens the NVMe namespace create dialog box, which allows you to set up a new namespace and map it to an NVMe subsystem.

- **Edit**

  Enables you to edit the namespace mapping.

- **Delete**

  Deletes the selected namespace.

- **More Actions**

  Allows you to create a clone of the selected namespace, which can be associated with an existing subsystem, or you can choose not to map it to a subsystem.

- **Refresh**

  Updates the information in the window.

NVMe List

- **Status**

  Displays if the namespace is online or offline.

- **Namespace Path**

  The path to the new namespace in the `/vol/volume'/file` format. The namespace path is a clickable link. Clicking the link takes you to the namespace detailspage.

- **NVMe Subsystem**

  The name of the subsystem attached to a namespace. If no subsystems are attached, the value of this column is shown as `None`. You can see the list of unmapped namespaces by filtering this column for NVMe subsystem contains `None`.

- **SVMs**

  The SVM name on which the namespace is created. The SVM name is a clickable link. Clicking the link takes you to the existing SVM dashboard page.

  Starting with ONTAP 9.5, at least one NVMe LIF must be configured for each node of a HA pair associated with the SVM. You can create a maximum of two NVMe LIFs for each node in the pair.

- **Namespace ID**

  A unique identifier used by the controller to provide access to a namespace. This is not a user input; it is generated by the system when the new namespace is created.

- **Total Space**

  Displays the total size of the namespace.

- **Used Space**

  Displays the amount of used space in the namespace.

- **%Used**

  Displays the amount of space (in percentage) that is used in the namespace. The value for this field is calculated using total and used space.

Details Area

You can select a namespace to view information about the selected namespace. From this area, you can also edit, delete or clone the namespace.

- **Overview tab**

  Displays general information about the selected namespace, and displays a pictorial representation of the space allocation of the namespace and the performance of the namespace.

  In the Overview tab, the SVM and volume names are clickable links. Clicking the link takes you to the SVM and volume pages, respectively. The number of hosts can be one or more; by default two host names are shown. If more than two host names are shown, you can click a link to access the additional hosts.

  The Overview tab also displays a space chart that shows the total and used space details for the namespace and a performance chart that shows details such as latency, IOPS, and throughput.

- **Status**

  The status of the namespace; the value can be online or offline.

- **Host NQN**

  The host NVMe Qualified Names (NQNs) uniquely describes the host for the purposes of identification and authentication. This field can accept comma separated NVMe qualification name (NQN) values. The host NQN starts with `nqn` and rest of the validation is the same as the initiator qualification name (IQN).

- **Host OS**

The host operating system for the namespace: Hyper-V, Linux, VMware, Windows or Xen.

- **Volume**

  Displays the volume name on which the namespace is hosted.

- **Read-Only**

  Displays whether the namespace is read-only or not.

- **Node**

  The node that owns the namespace.

- **Block Size**

  The size of the storage block.

- **Restore Inaccessible**

  If unmapping a subsystem fails and partial data remains, unmapped namespaces cannot be restored.