

## Volume Management

This section describes some of the advanced storage functionality provided by KumoScale software, such as volume expansion, and snapshots.

KumoScale volume management features are available using KumoScale interfaces for Kubernetes orchestration: [KumoScale REST API](#), [Cluster Manager CLI](#), [Ansible](#), [KumoScale Kubernetes CSI Driver](#), or [KumoScale for OpenStack](#). This section notes relevant KumoScale Cluster Manager CLI or REST API commands. To use these features with another orchestration environment, you will need to reference the relevant [KumoScale interface guide](#).

Topics included on this page:

- [Adding a Volume to a storage cluster](#)
- [Target management: Initiator to Volume Mapping](#)
  - [Publishing a volume to initiators](#)
  - [Unpublishing volumes](#)
  - [Viewing target information](#)
  - [Advanced target management](#)
- [Deleting volumes](#)
- [Expanding volumes](#)
- [Replicable volumes](#)
- [Cloning volumes](#)
- [Thin-provisioned volumes](#)
- [Snapshot management](#)

### Adding a Volume to a Storage Cluster

Volumes are added to storage clusters using the storagenode CRD as described in [Creating and Managing Storage Nodes in KumoScale](#). They may also be added through one of the KumoScale interfaces. For example, the KumoScale Cluster Manager CLI includes the [volume-add](#) command for adding a volume to a cluster and the [volume-replica-add](#) command for adding a replica to a cluster.

In the example below,

1. Create a replicable volume called **myvolume** with **20GiB[1]** capacity and 1 replica (the default value).

```
CLI> volume-add --alias myvolume --capacity 20 -- replicable Yes
```

The volume myvolume is created with UUID myvol-xxxxxxxxxx.

2. Later we decide to add a replica:

```
CLI> volume-replica-add myvol-xxxxxxxxxx
```

Once you have added the volume to the cluster, you will need to identify the application initiators (hosts) which will be used to manage the volume as explained in the next section, [Target Management: the Initiator to Volume Mapping](#).

### Target Management: the Initiator to Volume Mapping

Once a volume is available to a storage cluster, you must define how volumes will connect to application initiators (hosts). Application initiators (hosts) and compute nodes establish an NVMe-oF protocol connection to the volumes using a logical entity called a **target**. The process of defining how volumes are mapped to initiators is known as **Target Management**.

Connectivity between application initiators (hosts) and targets is controlled in the *Access Control List (ACL)*. When a volume is created, it is added to a target with the corresponding ACL as a **namespace**, thus enabling the initiator to connect to it. The ACL is set for each initiator and is associated with the initiator/target pair. The permitted connection types are Read Only and Read/Write access. This was discussed earlier in the *Targets and Access Control List (ACL)* section of the [KumoScale Software Overview](#).

Target Management allows to you customize which application initiators are available to a volume with the [publishing](#) process described below.

#### Publishing a Volume to One or More Initiators

Volumes and volume replicas may be added to clusters using any of the KumoScale interfaces as explained above To make them available to your application you must associate them with one or more application initiators (hosts). This is referred to as *publishing*. The KumoScale Cluster Manager CLI command [volume-host-connect](#) creates the mapping for a single initiator and supports multiple tenants. You can also publish a volume to more than one initiator. Associating volumes to multiple initiators enables faster recovery in situations such as failover or support of the OpenStack live migration feature.

**CAUTION:** When attaching a volume to multiple initiators, it is expected that the underlying applications maintain data integrity.

When a volume is attached to multiple initiators, it must be removed from all initiators with the *Unpublish* command before it can be deleted. See [Unpublishing a Volume](#) for details.

### Unpublishing a Volume

You can remove the association between a volume and an application initiator (host), a process known as *unpublishing*. The KumoScale Cluster Manager CLI command [volume-host-disconnect](#) supports this feature. When a volume is unpublished, the namespace associated with the target is also removed from the initiator. At this point, the associated volume may be removed from the cluster and deleted as explained in [Deleting a Volume](#).

To save resources, the KumoScale Provisioner service deletes a target if it has no namespaces.

#### CAUTIONS:

- (1) Once you remove a namespace from a target, all initiators (hosts) connected to the target will lose access to their data on that namespace.
- (2) Removing a namespace from a target does not delete the data from the volume. To delete the data, delete the volume after removing it from the target.

### Viewing Target Information

KumoScale supports a number of commands for getting information on the target mapping. For example the KumoScale Cluster Manager CLI supports:

- [hosts-show](#), lists all hosts in a cluster with their targets and namespaces
- [host-targets-show](#), lists all hosts with their targets.
- [volume-targets-show](#), lists all targets associated with a volume.

### Advanced Target Management

KumoScale administrators may want more direct control over how targets are managed. The KumoScale REST API supports a number of commands for modifying how targets and namespaces are defined:

- Add Namespaces*- adds namespaces to an existing target.
- Get Namespaces*- shows details for all KumoScale configured namespaces.
- Remove Namespaces*- removes Namespaces from an existing target.
- Create Target*- configures a Target entity in order to allow an NVMe / RDMA storage connection
- Get Target*- shows details for all KumoScale configured targets with or without Namespaces.
- Remove Target*- removes a configured target.

**CAUTION:** Removing a target does not delete its volumes but will delete the target with its namespaces. To ensure data is deleted, administrators must delete the volumes after deleting the target.

### Deleting a Volume

KumoScale allows you to remove a volume from a cluster and to delete it completely. For example, the KumoScale Cluster Manager CLI command [volume-remove](#) removes a volume from the cluster. Once a volume is removed from the cluster, you can delete the volume and all its data with the KumoScale REST API command *Delete Volume*. This command is irreversible.

**Note:** The REST API command *Delete Volume* will work only on volumes that are not in use. That is, only volumes not added as namespaces to a target can be deleted.

**CAUTION:** Data lost through deleting a volume cannot be recovered.

### Expanding a Volume

You can expand an existing volume as long as there is free capacity on its underlying SSD. The KumoScale Cluster Manager CLI command [volume-expand](#) will expand a volume. If the volume is allowed to span more than one SSD, it can be expanded to the maximum available capacity of its underlying group.

The expansion may be performed online non-disruptively, even while initiators are connected and using it.

### Replicable Volumes: Adding and Removing a Replica

KumoScale software allows adding a replica to a resilient volume using the KumoScale Cluster Manager CLI command [volume-replica-add](#). You can also remove a replica from a volume using [volume-replica-remove](#). Removing a replica is useful in the event a replica is permanently down, or when preparing for a planned maintenance operation, where a rack will be shut down or disconnected.

You can add a replica to a simple volume after it's initially created. For example, you may create a simple volume with **replicas**=1 for proof of concept and later decide you want to add replicas for testing. To do this, use the [volume-replica-add](#) command and change the values of the fields **replicable** (from false to true) and **replicas** (from 1 to a number greater than 1).

A resilient volume may temporarily contain up to four (4) replicas.

At least one replica should be available when adding a new replica. The KumoScale Provisioner service will use the capacity and Quality of Service (QoS) requirements from the existing replica(s), and whether spanning is allowed (allowSpan field). Other parameters, such as topology, will be read from command parameters (or ignored, if not specified).

**Note:** When calling the KumoScale Provisioner service directly, the administrator must add or remove the new replica to/from the initiator’s agent. This caveat is not applicable when working with the CSI driver. The Ansible playbooks contain an example (self-healing) of how to implement something similar.

### Cloning Volumes

Cloning a volume is equivalent to creating a volume with the starting image of an existing volume. The total number of clones and snapshots created from the same source volume must be no more than eight (8). The KumoScale Cluster Manager CLI command [volume-clone](#) may be used to clone an existing volume.

### Thin-Provisioned Volumes

Thin-provisioned volumes are created by defining the provisioningType as *thin* in the storage class CRD. The default provisioningType value is *thick*. Note the following constraints for thin-provisioned volumes:

- The minimum size for a thin-provisioned volume is 40GB.
- The maximum size for a thin-provisioned volume is 512 terabytes<sup>[1]</sup> (TB).
- A minimal assured space may be specified using the reservedSpacePercentage field (the default is 10%) in the storage class CRD.
- The block size must be set to 4KB, in order to reduce metadata.
- KumoScale software requires a small amount of capacity for metadata management (<100MB) when creating a thin-provisioned volume. This is subtracted from the group’s overall capacity.

When a thin-provisioned volume reaches its reserved space minus a 10GB margin, the actual allocated space is expanded to the maximum between 20GB and 2% of the volume’s capacity. If required, the input/output (I/O) will be throttled during the expansion to ensure no write errors are received.

### Snapshots

**Note:** To maintain data coherency, the application layer should pause any volume activity when creating the snapshot.

A snapshot is a point in time representation of the volume, but it cannot be exposed or used. To use a snapshot, administrators must create a snapshot volume over the snapshot. This is useful for backup and restore, or for testing and development purposes where you need use real data without harming the production environment.

### Creating a Snapshot and Snapshot Volumes

Administrators can take up to eight snapshots over a volume using the KumoScale Cluster Manager CLI command [snapshot-add](#).

Once a snapshot is created, KumoScale administrators can use [snapshot-volume-add](#) to create up to 512 snapshot volumes over the snapshot. Each snapshot volume may be either *Read/Write* or *Read Only*.

Notes:

(1) There must be at least 1GB free capacity on an SSD hosting the source volume to take a snapshot over it.

(2) The snapshot and writable snapshot volume require a small amount of space for metadata handling. This capacity shall be subtracted from the SSDs total available space.

(3) Administrators can set the value of reserved space when creating the snapshot or the writeable snapshot volume, as a percentage of the original volume capacity. The default value is 10%.

(4) If the reserved space is completely utilized, KumoScale software will attempt to auto-expand space. If this is not possible, the snapshot or the snapshot volume may become invalid.

(5) A snapshot volume does not inherit the QoS properties of its source volume. Administrators can specify different QoS requirements when creating a new snapshot volume.

### Viewing a Snapshot and its Snapshot Volumes

Administrators can view the list of snapshots in a cluster using the KumoScale Cluster Manager CLI command [snapshot-show](#). Snapshot volumes appear along with non-snapshot volumes in the reply to [volume-show](#). The snapshot volumes may be distinguished from non-snapshot volumes by the value of *snapshot UUID* – only snapshot volumes will return this field.

The result for each of these commands includes the snapshot data state. The various states, when they appear in volume, snapshots or snapshot volumes, include the following:

Data State	Volume	Snapshot	Snapshot Volume
Invalid	N/A	There was not enough space, the snapshot is invalid (failed to save a change).	Occurs when the origin snapshot is invalid.
Valid	The data is valid		
Corrupted	KumoScale software detected a corruption in the underlying volume's data.		

### Deleting a Snapshot and its Snapshot Volumes

To delete a snapshot volume, follow the instructions in [Deleting a Volume](#).

You may delete a snapshot only after all snapshot volumes using that snapshot are first deleted. Once that is true, you can use the KumoScale Cluster Manager CLI command [snapshot-remove](#) to remove the snapshot.

[1] Definition of capacity - KIOXIA Corporation defines a megabyte (MB) as 1,000,000 bytes, a gigabyte (GB) as 1,000,000,000 bytes and a terabyte (TB) as 1,000,000,000,000 bytes. A computer operating system, however, reports storage capacity using powers of 2 for the definition of 1Gbit = 230 bits = 1,073,741,824 bits, 1GB = 230 bytes = 1,073,741,824 bytes and 1TB = 240 bytes = 1,099,511,627,776 bytes and therefore shows less storage capacity. Available storage capacity (including examples of various media files) will vary based on file size, formatting, settings, software and operating system, and/or pre-installed software applications, or media content. Actual formatted capacity may vary.

Next: [Volume Migration](#)