

## Volume Migration

This page explains when and how to migrate volumes, including snapshots and clones, from one storage node to another.

There are several cases where a large data center would need to move volumes from one storage node to another, a function known as **Volume Migration**. These include the need to:

- Efficiently manage storage by balancing workloads and capacity, snapshots, and thin provisioned volume expansion
- Simplify maintenance, by evicting data on systems undergoing maintenance tasks such as updating software or replacing drives with expiring warranties.
- Manage Performance, by leveraging SSD performance in multi-tenant environments by migrating volume data as needed.

**Note:** If you want to migrate volumes within the same storage node, use the Evict SSD operation described in [Evict SSD](#).

KumoScale supports volume migration for different volume types including simple volumes, replicable volumes, snapshots, and clones. To determine how to migrate volumes in your deployment, review:

- [Types of Volume Migration](#), explains how volume migration is implemented in KumoScale and differences between host migration and backend migration.
- [Target Per Host](#), explains how KumoScale implemented support for multipathing required for online migrations.
- [Migrating volumes](#) summarizes what KumoScale interfaces are available to perform volume migrations and provides instructions for using [Volume Operations CRD](#) and the [KumoScale Cluster Manager CLI](#).

### Types of Volume Migrations

KumoScale supports both:

- Offline Migration** where a volume is taken out of service during migration, and
- Online Migration** where a volume can continue serving I/O while the migration is in progress. There is no downtime and the process appears transparent to users.

KumoScale supports two ways of migrating volumes:

- Backend Volume Migration** is where a volume is moved from one storage node to another storage node. Backend volume migration supports both offline and online of simple and replicated volumes.
- Host Volume Migration** is where the host agent controls the migration process. A new leg is added to the volume, then the host MD layer uses a sync operation to copy the data to the new leg. Host volume migration can only be done when the volume is online, to allow the host to copy the data.

Differences between the Backend and Host volume migration are summarized in the table below:

Feature	Backend Volume Migration	Host Volume Migration
Online and Offline Volumes Supported?	Online and Offline	Online only
Multipath Required?	Online volumes require Multipath	Not required
Volume Type	Replicable and Simple	Replicable only

#### Impact of Volume Migration on Volume Storage

- Both types of migration add a replica during migration and delete it once the migration is complete.
- Backend migration of snapshot volumes and clones will *flatten* the volume, that is it eliminates the dependency on the source volume. This allows the snapshot or clone to exist independently of the source volume. The total storage consumption is usually higher after this kind of migration.

#### Target Per Host to Support Online Migrations

As noted above, multipathing is required for online volume migrations. MPIO-ANA multipathing requires that:

- all targets connected to the same host have the same NVMe Qualified Name (NQN),
- the source and destination volume have the same Network Service Identifier (NSID).

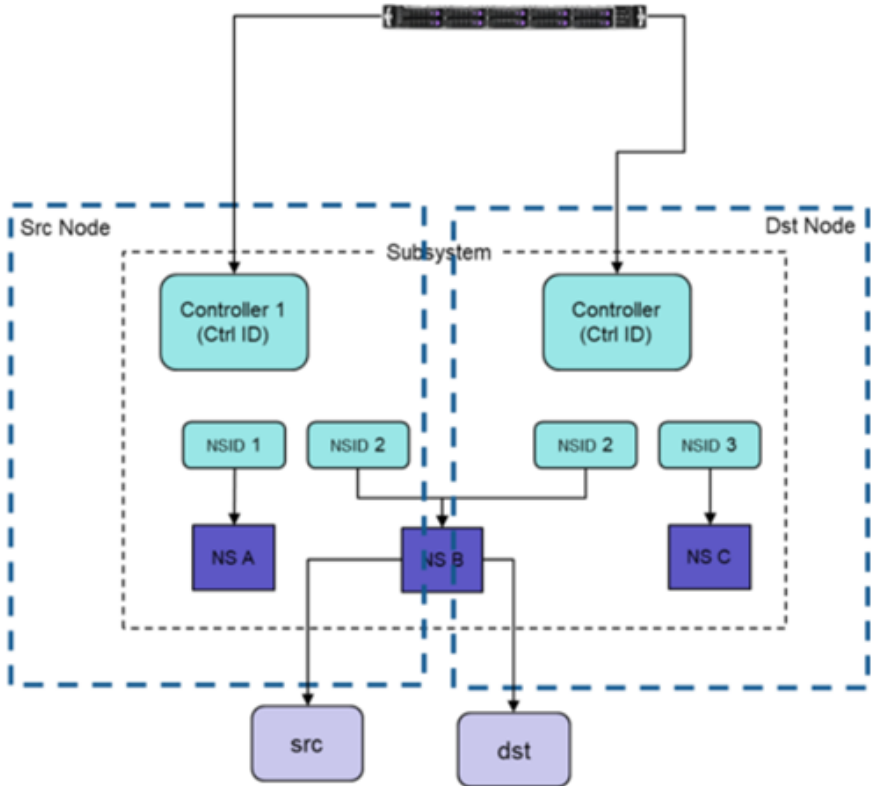
To support this feature KumoScale implemented the concept of **Target per Host** across the cluster to prevent a single point of failure and provide continuous operations.

- Storage nodes receive a unique NodeID which are used to calculate the controller IDs.
- Targets dynamically assign each controller an ID from the entire range of controller IDs.

- Targeting per host is limited to no more than 1000 nodes and a no more than 64 sessions (or controllers per target).

The diagram below illustrates the above

- The host has two controllers, *Controller 1* and *Controller*
- *Controller 1* and *Controller* both include *NSID 2* which are connected to the source volume, *src*, and destination volume, *dst*, through *NS B*.



### Migrating Volumes

The following sections show how to migrate volumes using the Volume Operations CRD or the [KumoScale Cluster Manager CLI](#). You may also perform these activities using other KumoScale interfaces for orchestration: [KumoScale REST API](#), [Ansible](#), [KumoScale Kubernetes CSI Driver](#), or [KumoScale for OpenStack](#).

### Migrating Volumes Using the Volume Operations CRD

The Volume Operations Custom Resource File is designed to support Volume Migration. The file `kumoscale_v1_volumeoperation_cryaml` contains a template for the CRD:

```
apiVersion: kumoscale.kioxia.com/v1
kind: VolumeOperation
metadata:
  name: volumeoperation-sample

spec:
  # Add fields here
  operation: Migrate

  #Set the alias for the volume to migrate
  volume: pvc-#####-####-####-####-#####

  #Set the storage node name that host the replica of the volume you wish to migrate
  storageNode: ks-node3-#####

  #set adminState to ABORT if you want to abort the operation
  #adminState: RUN

  #Set parameters per operation
  #parameters:
  # handler - select from {Backend,Host}
  # handler: Host
  # destinationStorageNode: <storage node name>
```

To migrate data from one device to another:

1. Make a copy of `kioxia.com_v1_volumeoperation_cryaml` for editing, and save to a separate directory (for example, `deploy/crds/myapp_volumeoperation_cryaml`).
2. Update `myapp_volumeoperation_cryaml` file with values for the parameters listed below:

Volume Operations Parameter Name	Description	Optional/Required
volume	The alias of the volume.	Required
storageNode	The name of the storage node that hosts the volume replica.	Required when there is more than one replica
operation	The operation to perform. In this case <b>migrate</b> .	Required
destinationStorageNode	The destination storage node of the data being migrated. The destination storage node must be available.	Optional. If not specified the Provisioner will independently select a suitable storage node

	This parameter applies to Backend Migration only.	according the source volume topology and space requirements.
handler	One of <b>host</b> (default) or backend.	Set <b>handler=backend</b> when operation=migrate
adminState	The state of the operation to perform. Equal to either <b>RUN</b> (default) or ABORT.	Optional

3. Use **kubect**l with the appropriate command to migrate the volume:

```
kubectl apply -f myapp_volumeoperation_cr.yaml
```

The status and result of the migration will be displayed to the screen.

### Migrating Volumes using the KumoScale Cluster Manager CLI

The KumoScale Cluster Manager CLI command [volume-migrate](#) supports migrating a volume to a different storage node.

```
CLI> volume migrate - --name <name> --alias <volume alias> --source-storagenode <storagenode name> --destination-storag
< >
```

where

Parameter	Description	Required?
name	The name of the migration operation	Required
alias	The alias of the volume	Required
source-storagenode	The name of the source storage node	Required only for volume with replicas
destination-storagenode	The name of the destination storage node	Optional
migrate-type	Handler of the migration (aka migration type) Backend or Host. Default value is Hot	Optional Defaults to Host Migration

You can monitor the progress of one or all migrations using the [volume-migrate-show](#) command:

```
CLI> volume-migrate-show --name <name> --detail
```

Parameter	Description	Required?
name	The name of the migration operation	Optional
alias	The alias of the volume	Optional

You can manage the volume migration operation with

- [volume-migrate-update](#), changes the migrated volume (replica) settings. For example, you can change the values of the source storage node, destination storage node, and migration type.
- [volume-migrate-abort](#) to STOP the Host Volume migration process as long as it is in either Pending or Running state
- [volume-migrate-delete](#) to delete the volume migration operation as long as the migration has state of Failed, Aborted, or Done.

**Next:** [Maintaining your KumoScale Deployment](#)

