A StorageClass allows administrators to describe the **classes** of storage they offer. Different Longhorn StorageClasses might map to replica policies, or to node schedule policies, or disk schedule policies determined by the cluster administrators. This concept is sometimes called **profiles** in other storage systems.

:::note

For support with other storage, please refer to Third-Party Storage Support

:::

Creating a StorageClass

You can create one or more StorageClasses from the **Advanced > StorageClasses** page.

:::note

After a StorageClass is created, nothing can be changed except `Description`.

:::

Header Section

1. **Name**: name of the StorageClass
2. **Description** (optional): description of the StorageClass

Parameters Tab

Number of Replicas

The number of replicas created for each volume in Longhorn. Defaults to `3`.

Stale Replica Timeout

Determines when Longhorn would clean up an error replica after the replica's status is ERROR. The unit is minute. Defaults to `30` minutes in Harvester.

Node Selector (Optional)

Select the node tags to be matched in the volume scheduling stage. You can add node tags by going to **Host > Edit Config**.

Disk Selector (Optional)

Select the disk tags to be matched in the volume scheduling stage. You can add disk tags by going to **Host > Edit Config**.

Migratable

Whether Live Migration is supported. Defaults to `Yes`.

Customize Tab

Reclaim Policy

Volumes dynamically created by a StorageClass will have the reclaim policy specified in the `reclaimPolicy` field of the class. The `Delete` mode is used by default.

1. `Delete`: Deletes volumes and the underlying devices when the volume claim is deleted.
2. `Retain`: Retains the volume for manual cleanup.

Allow Volume Expansion

Volumes can be configured to be expandable. This feature is `Enabled` by default, which allows users to resize the volume by editing the corresponding PVC object.

:::note

You can only use the volume expansion feature to grow a Volume, not to shrink it.

:::

Volume Binding Mode

The `volumeBindingMode` field controls when volume binding and dynamic provisioning should occur. The `Immediate` mode is used by default.

1. `Immediate`: Binds and provisions a persistent volume once the PersistentVolumeClaim is created.
2. `WaitForFirstConsumer`: Binds and provisions a persistent volume once a VM using the PersistentVolumeClaim is created.

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  annotations:
    storageclass.beta.kubernetes.io/is-default-class: 'true'
    storageclass.kubernetes.io/is-default-class: 'true'
  name: single-replica
parameters:
  migratable: 'false'
  numberOfReplicas: '1'
  staleReplicaTimeout: '30'
```

```
provisioner: driver.longhorn.io
reclaimPolicy: Delete
volumeBindingMode: Immediate
allowVolumeExpansion: true
```

```
resource "harvester_storageclass" "single-replica" {
  name = "single-replica"

  is_default = "true"
  allow_volume_expansion = "true"
  volume_binding_mode = "immediate"
  reclaim_policy = "delete"

  parameters = {
    "migratable"          = "false"
    "numberOfReplicas"    = "1"
    "staleReplicaTimeout" = "30"
  }
}
```

Data Locality Settings

You can use the `dataLocality` parameter when at least one replica of a Longhorn volume must be scheduled on the same node as the pod that uses the volume (whenever possible).

Harvester officially supports data locality as of **v1.3.0**. This applies even to volumes created from images. To configure data locality, create a new StorageClass on the Harvester UI (**Storage Classess** > **Create** > **Parameters**) and then add the following parameter:

- **Key**: `dataLocality`
- **Value**: `disabled` or `best-effort`

Data Locality Options

Harvester currently supports the following options:

- `disabled` : When applied, Longhorn may or may not schedule a replica on the same node as the pod that uses the volume. This is the default option.
- `best-effort` : When applied, Longhorn always attempts to schedule a replica on the same node as the pod that uses the volume. Longhorn does not stop the volume even when a local replica is unavailable because of an environmental limitation (for example, insufficient disk space or incompatible disk tags).

:::note Longhorn provides a third option called `strict-local` , which forces Longhorn to keep only one replica on the same node as the pod that uses the volume. Harvester does not support this option because it can affect certain operations such as VM Live Migration :::

For more information, see Data Locality in the Longhorn documentation.

Appendix - Use Case

HDD Scenario

With the introduction of *StorageClass*, users can now use **HDDs** for tiered or archived cold storage.

:::caution

HDD is not recommended for guest RKE2 clusters or VMs with good performance disk requirements.

:::

Recommended Practice

First, add your HDD on the `Host` page and specify the disk tags as needed, such as `HDD` or `ColdStorage`. For more information on how to add extra disks and disk tags, see Multi-disk Management for details.

Then, create a new `StorageClass` for the HDD (use the above disk tags). For hard drives with large capacity but slow performance, the number of replicas can be reduced to improve performance.

You can now create a volume using the above `StorageClass` with HDDs mostly for cold storage or archiving purpose.