:::caution

A known issue in v0.1.20 of the Harvester CSI driver causes volumes to get stuck when the host cluster is running a Harvester version that was released before v1.4.0.

This issue was fixed in v0.1.21. If your system is affected, you can follow the suggested workaround.

Harvester CSI Driver Version	Harvester Version	Affected
v0.1.21 and later	All versions	No
v0.1.20	v1.4.0 and later	No
v0.1.20	v1.3.2 and earlier	Yes
v0.1.18 and earlier	All versions	No

:::

The Harvester Container Storage Interface (CSI) Driver provides a standard CSI interface used by guest Kubernetes clusters in Harvester. It connects to the host cluster and hot-plugs host volumes to the virtual machines (VMs) to provide native storage performance.

Deploying

Prerequisites

- The Kubernetes cluster is built on top of Harvester virtual machines.
- The Harvester virtual machines that run as guest Kubernetes nodes are in the same namespace.

:::note

Currently, the Harvester CSI driver only supports single-node read-write(RWO) volumes. Please follow the issue #1992 for future multi-node `read-only` (ROX) and `read-write` (RWX) support.

:::

Deploying with Harvester RKE1 node driver

:::caution

Rancher Kubernetes Engine (RKE) will reach the end of its life on **July 31, 2025**. Harvester **v1.6.0** and later versions will not support RKE. Switching to RKE2, which provides a more secure and efficient environment, is recommended.

In-place upgrades are not an option, so you must create new RKE2 clusters and migrate the workloads from your existing RKE clusters (known as replatforming). For more information, see RKE End of Life.

:::

- Select the `Harvester(Out-of-tree)` option.

- Install `Harvester CSI Driver` from the Rancher marketplace.

Deploying with Harvester RKE2 node driver

When spinning up a Kubernetes cluster using Rancher RKE2 node driver, the Harvester CSI driver will be deployed automatically when Harvester cloud provider is selected.

select-harvester-cloud-provider

Install CSI driver manually in the RKE2 cluster

If you prefer to install the Harvester CSI driver without enabling the Harvester cloud provider, you can refer to the following steps:

Prerequisites of manual install

Ensure that you have the following prerequisites in place:

- You have `kubectl` and `jq` installed on your system.
- You have the `kubeconfig` file for your bare-metal Harvester cluster. You can find the `kubeconfig` file from one of the Harvester management nodes in the `/etc/rancher/rke2/rke2.yaml` path.

  ```
  export KUBECONFIG=/path/to/your/harvester-kubeconfig
  ```

Perform the following steps to deploy the Harvester CSI driver manually:

Deploy Harvester CSI driver

1. Generate the `cloud-config`. You can generate the `cloud-config` file using the [generate_addon_csi.sh]() script. It is available on the [harvester/harvester-csi-driver]() repo.

   `<serviceaccount name>` usually corresponds to your guest cluster name, and `<namespace>` should match the machine pool's namespace.

   ```
   ./generate_addon_csi.sh <serviceaccount name> <namespace> RKE2
   ```

   The generated output will be similar to the following one:

   ```
   ########## cloud-config ############
   apiVersion: v1
   clusters:
   - cluster: <token>
       server: https://<YOUR HOST HARVESTER VIP>:6443
     name: default
   contexts:
   - context:
       cluster: default
       namespace: default
   ```

```
    user: rke2-guest-01-default-default
  name: rke2-guest-01-default-default
current-context: rke2-guest-01-default-default
kind: Config
preferences: {}
users:
- name: rke2-guest-01-default-default
  user:
    token: <token>

########## cloud-init user data ############
write_files:
  - encoding: b64
    content:
YXBpVmVyc2lvbjogdjEKY2x1c3RlcnM6Ci0gY2x1c3RlcjoKICAgIGNlcnRpZmljYXRlLWF1dGhvcml0
    owner: root:root
    path: /var/lib/rancher/rke2/etc/config-files/cloud-provider-config
    permissions: '0644'
```

2. Copy and paste the `cloud-init user data` content to **Machine Pools** > **Show Advanced** > **User Data**.

   The `cloud-provider-config` file will be created after you apply the cloud-init user data above. You can find it on the guest Kubernetes nodes at the path `/var/lib/rancher/rke2/etc/config-files/cloud-provider-config`.

3. Configure the **Cloud Provider** either to **Default - RKE2 Embedded** or **External**.

4. Select **Create** to create your RKE2 cluster.

5. Once the RKE2 cluster is ready, install the **Harvester CSI Driver** chart from the Rancher marketplace. You do not need to change the **cloud-config** path by default.

:::note

If you prefer not to install the Harvester CSI driver using Rancher (**Apps** > **Charts**), you can use Helm instead. The Harvester CSI driver is packaged as a Helm chart. For more information, see https://charts.harvesterhci.io.

:::

By following the above steps, you should be able to see those CSI driver pods are up and running on the `kube-system` namespace, and you can verify it by provisioning a new PVC using the default StorageClass `harvester` on your RKE2 cluster.

Deploying with Harvester K3s node driver

You can follow the [Deploy Harvester CSI Driver](#) steps described in the RKE2 section.

The only difference is in generating the `cloud-init` config where you need to specify the provider type as `k3s`:

```
./generate_addon_csi.sh <serviceaccount name> <namespace> k3s
```

Customize the Default StorageClass

The Harvester CSI driver provides the interface for defining the default StorageClass. If the default StorageClass in unspecified, the Harvester CSI driver uses the default StorageClass of the host Harvester cluster.

You can use the parameter `host-storage-class` to customize the default StorageClass.

1. Create a StorageClass for the host Harvester cluster.

   Example:

2. Deploy the CSI driver with the parameter `host-storage-class`.

   Example:

3. Verify that the Harvester CSI driver is ready.

   1. On the **PersistentVolumeClaims** screen, create a PVC. Select **Use a Storage Class to provision a new Persistent Volume** and specify the StorageClass you created.

      Example:

   2. Once the PVC is created, note the name of the provisioned volume and verify that the status is **Bound**.

      Example:

   3. On the **Volumes** screen, verify that the volume was provisioned using the StorageClass that you created.

      Example:

Passthrough Custom StorageClass

Beginning with Harvester CSI driver v0.1.15, it's possible to create a PersistentVolumeClaim (PVC) using a different Harvester StorageClass on the guest Kubernetes Cluster.

:::note

Harvester CSI driver v0.1.15 is supported out of the box starting with the following RKE2 versions. For RKE1, manual installation of the CSI driver chart is required:

- v1.23.16+rke2r1 and later
- v1.24.10+rke2r1 and later

- v1.25.6+rke2r1 and later
- v1.26.1+rke2r1 and later
- v1.27.1+rke2r1 and later

:::

Prerequisites

Add the following prerequisites to your Harvester cluster to ensure the Harvester CSI driver displays error messages correctly. Proper RBAC settings are essential for error message visibility, especially when creating a PVC with a non-existent StorageClass, as shown in the image below:

Follow these steps to set up **RBAC** for error message visibility:

1. Create a new `clusterrole` named `harvesterhci.io:csi-driver` using the following manifest.

   ```
   apiVersion: rbac.authorization.k8s.io/v1
   kind: ClusterRole
   metadata:
     labels:
       app.kubernetes.io/component: apiserver
       app.kubernetes.io/name: harvester
       app.kubernetes.io/part-of: harvester
     name: harvesterhci.io:csi-driver
   rules:
   - apiGroups:
     - storage.k8s.io
     resources:
     - storageclasses
     verbs:
     - get
     - list
     - watch
   ```

2. Create a new `clusterrolebinding` associated with the `clusterrole` above with the relevant `serviceaccount` using the following manifest.

   ```
   apiVersion: rbac.authorization.k8s.io/v1
   kind: ClusterRoleBinding
   metadata:
     name: <namespace>-<serviceaccount name>
   roleRef:
     apiGroup: rbac.authorization.k8s.io
     kind: ClusterRole
     name: harvesterhci.io:csi-driver
   subjects:
   - kind: ServiceAccount
     name: <serviceaccount name>
     namespace: <namespace>
   ```

Make sure the `serviceaccount name` and `namespace` match your cloud provider settings. Perform the following steps to retrieve these details.

1. Find the `rolebinding` associated with your cloud provider:

   ```
   $ kubectl get rolebinding -A |grep harvesterhci.io:cloudprovider
   default                                  default-rke2-guest-01
   ClusterRole/harvesterhci.io:cloudprovider             7d1h
   ```

2. Extract the `subjects` information from this `rolebinding` :

   ```
   $ kubectl get rolebinding default-rke2-guest-01 -n default -o yaml |yq -
   e '.subjects'
   ```

3. Identify the `ServiceAccount` information:

   ```
   - kind: ServiceAccount
     name: rke2-guest-01
     namespace: default
   ```

Deploying

Now you can create a new StorageClass that you intend to use in your guest Kubernetes cluster.

1. For administrators, you can create a desired StorageClass (e.g., named **replica-2**) in your bare-metal Harvester cluster.

2. Then, on the guest Kubernetes cluster, create a new StorageClass associated with the StorageClass named **replica-2** from the Harvester Cluster:

   :::note

   - When choosing a **Provisioner**, select **Harvester (CSI)**. The **Host StorageClass** parameter should match the StorageClass name created on the Harvester Cluster.
   - For guest Kubernetes owners, you may request that the Harvester cluster administrator create a new StorageClass.
   - If you leave the `Host StorageClass` field empty, the default StorageClass of the Harvester cluster will be used.

   :::

3. You can now create a PVC based on this new **StorageClass**, which utilizes the **Host StorageClass** to provision volumes on the bare-metal Harvester cluster.

RWX Volumes Support

:::caution

RWX volumes currently only work with a dedicated storage network. GitHub issue #7218 tracks the enhancement that will allow RWX volumes to use various VLANs on guest clusters.

:::

Prerequisites

- Harvester v1.4 or later is installed on the host cluster.
- A storage network is configured on the Harvester cluster.

  Use **exclude** to reserve a range of IP addresses for the guest cluster virtual machines.

- The **Storage Network for RWX Volume** setting on the embedded Longhorn UI is enabled.

  Go to **General**, and then select **Storage Network for RWX Volume Enabled**.

- You have created an RWX StorageClass on the host Harvester cluster.

  On the **Storage Class: Create** screen, click **Edit as YAML** and specify the following:

  ```
  kind: StorageClass
  apiVersion: storage.k8s.io/v1
  metadata:
    name: longhorn-rwx
  provisioner: driver.longhorn.io
  allowVolumeExpansion: true
  reclaimPolicy: Delete
  volumeBindingMode: Immediate
  parameters:
    numberOfReplicas: "3"
    staleReplicaTimeout: "2880"
    fromBackup: ""
    fsType: "ext4"
    nfsOptions: "vers=4.2,noresvport,softerr,timeo=600,retrans=5"
  ```

- The role-based access control (RBAC) settings are up-to-date.

  RBAC authorization uses a specific Kubernetes API group to drive authorization decisions regarding access to computer or network resources.

  The Harvester CSI driver requires the new RBAC settings to support RWX volumes. To check the RBAC settings, run the command `kubectl get clusterrole harvesterhci.io:csi-driver -o yaml`.

```
# kubectl get clusterrole harvesterhci.io:csi-driver -o yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
...
  name: harvesterhci.io:csi-driver
...
rules:
- apiGroups:
  - storage.k8s.io
  resources:
  - storageclasses
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - harvesterhci.io
  resources:
  - networkfilesystems
  - networkfilesystems/status
  verbs:
  - '*'
- apiGroups:
  - longhorn.io
  resources:
  - volumes
  - volumes/status
  verbs:
  - get
  - list
```

- The networkfs-manager pods are running.

  To check the status of the networkfs-manager pods, run the command `kubectl get pods -n harvester-system | grep networkfs-manager`.

  Example:

  ```
  # kubectl get pods -n harvester-system | grep networkfs-manager
  harvester-networkfs-manager-2pxhm                       1/1     Running     4
  (34m ago)    3h41m
  harvester-networkfs-manager-8tst2                       1/1     Running     4
  (37m ago)    3h41m
  harvester-networkfs-manager-xvkgp                       1/1     Running     4
  (37m ago)    3h41m
  ```

- The VM must have two interfaces. The first one is the default network interface for cluster/external networking. The second one must be in a network which can connect to the storage network.

  The NAD **default/vlan101** is used for the storage network.

- The Harvester CSI driver version is v0.1.20 or later.

- The NFS client is installed on each node in the guest cluster.

  Run any of the following commands to install the NFS client.

  - Debian and Ubuntu: `apt-get install -y nfs-common`
  - CentOS and RHEL: `yum install -y nfs-utils`
  - SUSE and OpenSUSE: `zypper install -y nfs-client`

- An IP is manually assigned to the storage network interface.

  You can assign any of the reserved IPs using the following commands:

  ```
  $ ip link set <storage network nic> up
  $ ip a add <reserved IP> dev <storage network nic>
  ```

  :::info important

  An IP that is assigned using the given commands does not persist after a reboot. To make the IP persistent, you must add it to the network configuration file of your guest operating system.

  :::

Usage

1. Create a new StorageClass on the guest cluster.

On the **StorageClass: Create** screen, add a **Host Storage Class** parameter and specify the RWX StorageClass that you created on the host Harvester cluster.

1. Create an RWX PersistentVolumeClaim (PVC).

On the **PersistentVolumeClaim: Create** screen, configure the following settings:

- **Volume Claim** tab: Specify the new StorageClass.
- **Customize** tab: Select **Many Nodes Read-Write**.

1. Verify that the RWX PVC was created successfully.

1. Create two pods.

On the **Pod: Create** screen, specify the RWX PVC.

:::note

You can follow the same steps to create an RWX PVC on the guest cluster and then use it on pods that require RWX volumes.

:::

Upgrade the CSI Driver

Upgrade RKE2

To upgrade the CSI driver, use the Rancher UI to upgrade RKE2. Ensure the new RKE2 version supports/bundled with the updated CSI driver version.

1. Go to ≡ > **Cluster Management**.

2. Find the guest cluster that you want to upgrade and select ⋮ > **Edit Config**.

3. Select **Kubernetes Version**.

4. Click **Save**.

Upgrade RKE and K3s

You can upgrade RKE and K3s using the Rancher UI.

1. Go to ≡ > **RKE/K3s Cluster** > **Apps** > **Installed Apps**.

2. Find the CSI driver chart and select ⋮ > **Edit/Upgrade**.

3. Select **Version**.

4. Select **Next** > **Update**.