General information

An **Upgrade** button appears on the **Dashboard** screen whenever a new Harvester version that you can upgrade to becomes available. For more information, see Start an upgrade.

For air-gapped environments, see Prepare an air-gapped upgrade.

Update Harvester UI Extension on Rancher v2.10.1

To import Harvester v1.4.2 clusters on Rancher v2.10.1, you must use **v1.0.3** of the Rancher UI extension for Harvester.

1. On the Rancher UI, go to **local > Apps > Repositories**.
2. Locate the repository named **harvester**, and then select ⋮ **> Refresh**. This repository has the following properties:
    - URL: **https://github.com/harvester/harvester-ui-extension**
    - Branch: **gh-pages**
3. Go to the **Extensions** screen.
4. Locate the extension named **Harvester**, and then click **Update**.
5. Select version **1.0.3**, and then click **Update**.
6. Allow some time for the extension to be updated, and then refresh the screen.

:::info important The Rancher UI displays an error message after the extension is updated. The error message disappears when you refresh the screen. This issue, which exists in Rancher v2.10.0 and v2.10.1, will be fixed in v2.10.2. :::

Related issues:

- Issue #7234
- Issue #107

VM Backup Compatibility

In Harvester v1.4.2 and later versions, you may encounter certain limitations when creating and restoring backups that involve external storage

Known issues

---

1. Upgrade is stuck in the "Pre-drained" state

The upgrade process may become stuck in the "Pre-drained" state. Kubernetes is supposed to drain the workload on the node, but some factors may cause the process to stall.

A possible cause is processes related to orphan engines of the Longhorn Instance Manager. To determine if this applies to your situation, perform the following steps:

1. Check the name of the `instance-manager` pod on the stuck node.

   Example:

   The stuck node is `harvester-node-1`, and the name of the Instance Manager pod is `instance-manager-d80e13f520e7b952f4b7593fc1883e2a`.

   ```
   $ kubectl get pods -n longhorn-system --field-selector spec.nodeName=harvester-node-1 | grep instance-manager
   instance-manager-d80e13f520e7b952f4b7593fc1883e2a          1/1     Running   0          3d8h
   ```

2. Check the Longhorn Manager logs for informational messages.

   Example:

   ```
   $ kubectl -n longhorn-system logs daemonsets/longhorn-manager
   ...
   time="2025-01-14T00:00:01Z" level=info msg="Node instance-manager-d80e13f520e7b952f4b7593fc1883e2a is marked unschedulable but removing harvester-node-1 PDB is blocked: some volumes are still attached InstanceEngines count 1 pvc-9ae0e9a5-a630-4f0c-98cc-b14893c74f9e-e-0" func="controller.(*InstanceManagerController).syncInstanceManagerPDB" file="instance_manager_controller.go:823" controller=longhorn-instance-manager node=harvester-node-1
   ```

   The `instance-manager` pod cannot be drained because of the engine `pvc-9ae0e9a5-a630-4f0c-98cc-b14893c74f9e-e-0`.

3. Check if the engine is still running on the stuck node.

   Example:

   ```
   $ kubectl -n longhorn-system get engines.longhorn.io pvc-9ae0e9a5-a630-4f0c-98cc-b14893c74f9e-e-0 -o jsonpath='{"Current state: "}{.status.currentState}{"\nNode ID: "}{.spec.nodeID}{"\n"}'
   Current state: stopped
   Node ID:
   ```

   The issue likely exists if the output shows that the engine is not running or even the engine is not found.

4. Check if all volumes are healthy.

   ```
   kubectl get volumes -n longhorn-system -o yaml | yq '.items[] | select(.status.state == "attached")| .status.robustness'
   ```

   All volumes must be marked `healthy`. If this is not the case, please help to report the issue.

5. Remove the `instance-manager` pod's PodDisruptionBudget (PDB).

   Example:

```
kubectl delete pdb instance-manager-d80e13f520e7b952f4b7593fc1883e2a -n
longhorn-system
```

Related issues:

- [BUG] v1.4.0 -> v1.4.1-rc1 upgrade stuck in Pre-drained and the node stay in Cordoned
- [IMPROVEMENT] Cleanup orphaned volume runtime resources if the resources already deleted

2. High CPU Usage

High CPU usage may occur after an upgrade because of the `backup-target` setting's `refreshIntervalInSeconds` field, which was introduced in v1.4.2. If the field is left empty or is set to 0, Harvester constantly refreshes the backup target, resulting in high CPU usage.

To fix the issue, update the value of `refreshIntervalInSeconds` to a larger number (for example, 60) using the command `kubectl edit setting backup-target`. You can also update the value before starting the upgrade to prevent the issue from occurring.

Example:

```
value: '{"type":"nfs","endpoint":"nfs://longhorn-test-nfs-
svc.default:/opt/backupstore", "refreshIntervalInSeconds": 60}'
```

Related issues:

- [BUG] High load and use of cpu time by harvester process

3. Upgrade will start over again unexpectedly after clicking the "Dismiss it" button

When you use Rancher to upgrade Harvester, the Rancher UI displays a dialog with a button labeled "Dismiss it". Clicking this button may result in the following issues:

- The `status` section of the `harvesterhci.io/v1beta1/upgrade` CR is cleared, causing the loss of all important information about the upgrade.
- The upgrade process starts over again unexpectedly.

This issue affects Rancher v2.10.x, which uses v1.0.2, v1.0.3, and v1.0.4 of the Harvester UI Extension. All Harvester UI versions are not affected. The issue will be fixed in Harvester UI Extension v1.0.5 and v1.5.0.

To avoid this issue, perform either of the following actions:

- Use the Harvester UI to upgrade Harvester. Clicking the "Dismiss it" button on the Harvester UI does not result in unexpected behavior.
- Instead of clicking the button on the Rancher UI, run the following command against the cluster:

```
kubectl -n harvester-system label upgrades -l harvesterhci.io/latestUpgrade=true
harvesterhci.io/read-message=true
```

Related issue:

- [BUG] upgrade controller does not handle read-message well due to UI menu `Dismiss it` wipes upgrade CR's status