Upgrade support matrix

The following table shows the upgrade path of all supported versions.

Upgrade from version	Supported new version(s)
v1.4.2	v1.5.0
v1.4.1	v1.4.2
v1.4.0	v1.4.1
v1.3.2	v1.4.0
v1.3.1	v1.3.2
v1.2.2/v1.3.0	v1.3.1
v1.2.1	v1.2.2
v1.1.2/v1.1.3/v1.2.0	v1.2.1

Rancher upgrade

If you are using Rancher to manage your Harvester cluster, we recommend upgrading your Rancher server first. For more information, please refer to the Rancher upgrade guide.

For the Harvester & Rancher support matrix, please visit our website here.

:::note

- Upgrading Rancher will not automatically upgrade your Harvester cluster. You still need to upgrade your Harvester cluster after upgrading Rancher.
- Upgrading Rancher will not bring your Harvester cluster down. You can still access your Harvester cluster using its virtual IP.

:::

Before starting an upgrade

Check out the available `upgrade-config` setting to tweak the upgrade strategies and behaviors that best suit your cluster environment.

Start an upgrade

:::caution

- Before you upgrade your Harvester cluster, we highly recommend:
  - Back up your VMs if needed.
- Do not operate the cluster during an upgrade. For example, creating new VMs, uploading new images, etc.
- Make sure your hardware meets the **preferred** hardware requirements. This is due to there will be intermediate resources consumed by an upgrade.
- Make sure each node has at least 30 GiB of free system partition space (`df -h /usr/local/`). If any node in the cluster has less than 30 GiB of free system partition space, the upgrade will be denied. Check free system partition space requirement for more information.
- Run the pre-check script on a Harvester control-plane node. Please pick a script according to your cluster's version: https://github.com/harvester/upgrade-helpers/tree/main/pre-check.
- A number of one-off privileged pods will be created in the `harvester-system` and `cattle-system` namespaces to perform host-level upgrade operations. If pod security admission is enabled, adjust these policies to allow these pods to run.

:::

:::caution

- Make sure all nodes' times are in sync. Using an NTP server to synchronize time is recommended. If an NTP server is not configured during the installation, you can manually add an NTP server **on each node**:

  ```
  $ sudo -i

  # Add time servers
  $ vim /etc/systemd/timesyncd.conf
  [ntp]
  NTP=0.pool.ntp.org

  # Enable and start the systemd-timesyncd
  $ timedatectl set-ntp true

  # Check status
  $ sudo timedatectl status
  ```

:::

:::caution

- NICs that connect to a PCI bridge might be renamed after an upgrade. Please check the knowledge base article for further information.

:::

1. Make sure to read the above `caution`.
2. On the Harvester UI **Dashboard** screen, click **Upgrade**.

   The **Upgrade** button appears whenever a new Harvester version that you can upgrade to becomes available.

   If your environment does not have direct internet access, follow the instructions in Prepare an air-gapped upgrade, which provides an efficient approach to downloading the Harvester ISO.

3. Select a version that you want to upgrade to.

   If you require customizations, see Customize the Version.

4. Click the progress indicator (the **circle** button) to view the status of each related process.

Customize the Version

1. Download the version file (`https://releases.rancher.com/harvester/{version}/version.yaml`).

   Example:

   The Harvester v1.5.0 version file is downloaded as `v1.5.0.yaml`.

   ```
   apiVersion: harvesterhci.io/v1beta1
   kind: Version
   metadata:
     name: v1.5.0-customized # Changed, to avoid duplicated with the official version name
     namespace: harvester-system
   spec:
     isoChecksum:
   'df28e9bf8dc561c5c26dee535046117906581296d633eb2988e4f68390a281b6856a5a0bd2e4b5b988c695a53d0fc86e4e3965f19957682b74317109b1d2fe
   # Don't change
     isoURL: https://releases.rancher.com/harvester/v1.5.0/harvester-v1.5.0-amd64.iso # Official ISO path by default
     releaseDate: '20250425'
   ```

2. Add the necessary annotations.
   - minCertsExpirationInDay Annotation
   - skipGarbageCollectionThresholdCheck Annotation
3. Run `kubectl create -f v1.5.0.yaml` to create the version.

minCertsExpirationInDay Annotation

Harvester checks the validity period of certificates on each node. This check eliminates the possibility of certificates expiring while the upgrade is in progress. If a certificate will expire within 7 days, an error is returned.

Example:

`harvesterhci.io/minCertsExpirationInDay: "14"` - When this annotation is added, Harvester returns an error when it detects a certificate that will expire within 14 days.

For more information, see auto-rotate-rke2-certs.

skipGarbageCollectionThresholdCheck Annotation

Harvester checks the disk space on each node to ensure that the kubelet's image garbage collection threshold is not exceeded when the required images are loaded during upgrades.

Example:

`harvesterhci.io/skipGarbageCollectionThresholdCheck: true` - When this annotation is added, Harvester skips the check.

:::caution

Do not use this annotation in production environments. When the check is skipped, required images might be deleted, causing the upgrade to fail.

:::

For more information, see Free system partition space requirement.

Prepare an air-gapped upgrade

:::caution

Make sure to check Upgrade support matrix section first about upgradable versions.

:::

Prepare the ISO File

1. Download a Harvester ISO file from the Releases page.

2. Save the ISO to a local HTTP server. Assume the file is hosted at `http://10.10.0.1/harvester.iso` .

Prepare the Version

1. Download the version file (`https://releases.rancher.com/harvester/{version}/version.yaml`).

   - Replace `isoURL` value in the `version.yaml` file:

     ```
     apiVersion: harvesterhci.io/v1beta1
     kind: Version
     metadata:
       name: v1.5.0
       namespace: harvester-system
     spec:
       isoChecksum: <SHA-512 checksum of the ISO>
       isoURL: http://10.10.0.1/harvester.iso  # change to local ISO URL
       releaseDate: '20250425'
     ```

   - Assume the file is hosted at `http://10.10.0.1/version.yaml` .

   - If you require customizations, see Customize the Version.

2. Access one of the control plane nodes via SSH and log in using the root account.

3. Create a version object.

   ```
   rancher@node1:~> sudo -i
   rancher@node1:~> kubectl create -f http://10.10.0.1/version.yaml
   ```

Start the Upgrade

The **Upgrade** button appears on the **Dashboard** screen whenever a new Harvester version that you can upgrade to becomes available. Refresh the screen if the button does not appear.

Manually Start an Upgrade before the Harvester Official Upgrade is Available

The **Upgrade** button does not appear on the UI immediately after a new Harvester version is released. If you want to upgrade your cluster before the option becomes available on the UI, follow the steps in Prepare an air-gapped upgrade.

:::tip

In production environments, upgrading clusters via the Harvester UI is recommended.

:::

Free system partition space requirement

Available as of v1.5.0

Harvester loads images on each node during upgrades. When disk usage exceeds the kubelet's garbage collection threshold, the kubelet deletes unused images to free up space. This may cause issues in airgapped environments because the images are not available on the node.

Harvester v1.5.0 includes checks that ensure nodes do not trigger garbage collection after loading new images.

If you want to try upgrading even if the free system partition space is insufficient on some nodes, you can update the `harvesterhci.io/skipGarbageCollectionThresholdCheck: true` annotation of the `Version` object.

```
apiVersion: harvesterhci.io/v1beta1
kind: Version
metadata:
  annotations:
    harvesterhci.io/skipGarbageCollectionThresholdCheck: true
  name: 1.5.0
  namespace: harvester-system
spec:
  isoChecksum: <SHA-512 checksum of the ISO>
  isoURL: http://192.168.0.181:8000/harvester-master-amd64.iso
  minUpgradableVersion: 1.4.1
  releaseDate: "20250630"
```

:::caution

Setting a smaller value than the pre-defined value may cause the upgrade to fail and is not recommended in a production environment.

:::

The following sections describe solutions for issues related to this requirement.

Free System Partition Space Manually

Harvester attempts to remove unnecessary container images after an upgrade is completed. However, this automatic image cleanup may not be performed for various reasons. You can use this script to manually remove images. For more information, see issue #6620.

Set Up a Private Container Registry and Skip Image Preloading

The system partition might still lack free space even after you remove images. To address this, set up a private container registry for both current and new images, and configure the setting `upgrade-config` with following value:

```
{"imagePreloadOption":{"strategy":{"type":"skip"}}, "restoreVM": false}
```

Harvester skips the upgrade image preloading process. When the deployments on the nodes are upgraded, the container runtime loads the images stored in the private container registry.

:::caution

Do not rely on the public container registry. Note any potential internet service interruptions and how close you are to reaching your Docker Hub rate limit. Failure to download any of the required images may cause the upgrade to fail and may leave the cluster in a middle state.

:::

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

Longhorn Manager Crashes Due to Backing Image Eviction

:::caution

When upgrading to Harvester **v1.4.x**, Longhorn Manager may crash if the `EvictionRequested` flag is set to `true` on any node or disk. This issue is caused by a race condition between the deletion of a disk in the backing image spec and the updating of its status.

To prevent the issue from occurring, ensure that the `EvictionRequested` flag is set to `false` before you start the upgrade process.

:::

Re-enable RKE2 ingress-nginx Admission Webhooks (CVE-2025-1974)

If you disabled the RKE2 ingress-nginx admission webhooks to mitigate CVE-2025-1974, you must re-enable the webhook after upgrading to Harvester v1.5.0 or later.

1. Confirm that Harvester is using nginx-ingress v1.12.1 or later.

```
$ kubectl -n kube-system get po -l"app.kubernetes.io/name=rke2-ingress-nginx" -ojsonpath='{.items[].spec.containers[].image}'
rancher/nginx-ingress-controller:v1.12.1-hardened1
```

1. Run `kubectl -n kube-system edit helmchartconfig rke2-ingress-nginx` to **remove** the following configurations from the `HelmChartConfig` resource.
    - `.spec.valuesContent.controller.admissionWebhooks.enabled: false`
    - `.spec.valuesContent.controller.extraArgs.enable-annotation-validation: true`
2. Verify that the new `.spec.ValuesContent` configuration is similar to the following example.

```
apiVersion: helm.cattle.io/v1
kind: HelmChartConfig
metadata:
  name: rke2-ingress-nginx
  namespace: kube-system
spec:
  valuesContent: |-
    controller:
      admissionWebhooks:
        port: 8444
      extraArgs:
        default-ssl-certificate: cattle-system/tls-rancher-internal
      config:
        proxy-body-size: "0"
        proxy-request-buffering: "off"
      publishService:
        pathOverride: kube-system/ingress-expose
```

:::info important If the `HelmChartConfig` resource contains other custom ingress-nginx configuration, you must retain them when editing the resource. :::

1. Exit the `kubectl edit` command execution to save the configuration.

Harvester automatically applies the change once the content is saved.

1. Verify that the `rke2-ingress-nginx-admission` webhook configuration is re-enabled.

```
$ kubectl get validatingwebhookconfiguration rke2-ingress-nginx-admission
NAME                           WEBHOOKS   AGE
rke2-ingress-nginx-admission   1          6s
```

1. Verify that the ingress-nginx pods are restarted successfully.

```
kubectl -n kube-system get po -lapp.kubernetes.io/instance=rke2-ingress-nginx
NAME                                  READY   STATUS    RESTARTS   AGE
rke2-ingress-nginx-controller-l2cxz   1/1     Running   0          94s
```