

Currently, there are three ways that are supported to create an image: uploading images via URL, uploading images via local files, and creating images via volumes.

Upload Images via URL

To import virtual machine images in the **Images** page, enter a URL that can be accessed from the cluster. Description and labels are optional.

:::note

- The image name will be auto-filled using the URL address's filename. You can customize the image name at any time.
- Avoid using a daily build URL (for example, the [Ubuntu Jammy daily build](#)). When all replicas of a Longhorn backing image are lost, Longhorn attempts to download the file again for self-healing purposes. Using a daily build URL is problematic because the URL itself changes, causing a checksum mismatch and a conflict that results in lost replicas.

:::

To import a virtual machine image from a repository using the API, create a `VirtualMachineImage` object. You must specify a URL that can be accessed from the cluster.

Example:

```
apiVersion: harvesterhci.io/v1beta1
kind: VirtualMachineImage
metadata:
  name: opensuse-leap
  namespace: default
spec:
  description: A human-readable description for the VM image
  displayName: openSUSE-Leap
  sourceType: download
  url:
    "https://download.opensuse.org/repositories/Cloud:/Images:/Leap_15.5/images/openSUSE-Leap-15.5.x86_64-NoCloud.qcow2"
  checksum: 80c27afb7cd791ac86ee1b0b0c572a242f6142579db5beac841e71151d370cd6
```

For more information, see the [API reference](#).

```
resource "harvester_image" "opensuse154" {
  name      = "opensuse154"
  namespace = "harvester-public"

  display_name = "openSUSE-Leap-15.4.x86_64-NoCloud.qcow2"
  source_type  = "download"
  url          = "https://downloadcontent-us1.opensuse.org/repositories/Cloud:/Images:/Leap_15.4/images/openSUSE-Leap-15.4.x86_64-NoCloud.qcow2"
}
```

Upload Images via Local File

Currently, qcow2, raw, and ISO images are supported.

:::note

- Please do not refresh the page until the file upload is finished.

:::

HTTP 413 Error in Rancher Multi-Cluster Management

You can upload images from the [Multi-Cluster Management](#) screen on the **Rancher UI**. When the status of an image is *Uploading* but the progress indicator displays 0% for an extended period, check the HTTP response status code. *413* indicates that the size of the request body exceeds the limit.

The maximum request body size should be specific to the cluster that is hosting Rancher (for example, RKE2 clusters have a default limit of 1 MB but no such limit exists in K3s clusters).

The current workaround is to upload images from the **Harvester UI**. If you choose to upload images from the Rancher UI, you may need to configure related settings on the ingress server (for example, [proxy-body-size](#) in NGINX).

If Rancher is deployed on an RKE2 cluster, perform the following steps:

1. Edit the Rancher ingress.

```
$ kubectl -n cattle-system edit ingress rancher
```

2. Specify a value for `nginx.ingress.kubernetes.io/proxy-body-size` .

Example:

3. Delete the stuck image, and then restart the upload process.

Prolonged Uploading of Large Images in Rancher Multi-Cluster Management

If you upload a very large image (over 10 GB) from the **Multi-Cluster Management** screen on the Rancher UI, the operation may take longer than usual and the image status (Uploading) may not change.

This behavior is related to *proxy-request-buffering* in the ingress configuration, which is also specific to the cluster that is hosting Rancher.

The current workaround is to upload images from the **Harvester UI**. If you choose to upload images from the Rancher UI, you may need to configure related settings on the ingress server (for example, [proxy-request-buffering](#) in NGINX).

If Rancher is deployed on an RKE2 cluster, perform the following steps:

1. Edit the Rancher ingress.

```
$ kubectl -n cattle-system edit ingress rancher
```

2. Turn off `nginx.ingress.kubernetes.io/proxy-request-buffering` .

Example:

3. Delete the stuck image, and then restart the upload process.

Uploading Images Previously Downloaded from Harvester

Starting with **v1.5.5**, Longhorn [compresses backing images for downloading](#). If you attempt to upload a compressed backing image, Harvester rejects the attempt and displays the message **Upload failed: the uploaded file size xxxx should be a multiple of 512 bytes since Longhorn uses directIO by default** because the compressed data violates Longhorn's data alignment.

Before uploading, decompress backing images using the command `$ gzip -d <file name>` .

Create Images via Volumes

On the **Volumes** page, click **Export Image**. Enter the image name and select a StorageClass to create an image.

Image StorageClass

When creating an image, you can select a [StorageClass](#) and use its pre-defined parameters like replicas, node selectors and disk selectors .

:::note

The image will not use the `StorageClass` selected here directly. It's just a `StorageClass` template.

Instead, it will create a special StorageClass under the hood with a prefix name of `longhorn-` . This is automatically done by the Harvester backend, but it will inherit the parameters from the StorageClass you have selected.

:::

Image Labels

You can add labels to the image, which will help identify the OS type more accurately. Also, you can add any custom labels for filtering if needed.

If your image name or URL contains any valid information, the UI will automatically recognize the OS type and image category for you. If not, you can also manually specify those corresponding labels on the UI.