

A [Harvester cluster](#) with three or more nodes is required to fully realize multi-node features such as high availability. Certain versions of Harvester allow you to create clusters with two management nodes and one [witness node](#) (and optionally, one or more worker nodes). You can also create [single-node clusters](#) that support most Harvester features (excluding high availability, multi-replica support, and live migration).

This guide walks you through the steps required to deploy a **single-node cluster** and virtual machines (VMs) that can host [guest clusters](#) and run custom workloads.

## 1. Verify that the minimum hardware and network requirements are met.

Harvester is built for bare metal servers using enterprise-grade open-source software components. The installer automatically checks the hardware and displays warning messages if the minimum [requirements](#) are not met.

## 2. Prepare the installation files based on the installation method that you want to use.

You can download the installation files from the [Harvester Releases](#) page. The **Downloads** section of the release notes contains links to the ISO files and related artifacts. The following types of ISO files are available:

- **Full ISO:** Contains the core operating system components and all required container images, which are preloaded during installation. You must use a full ISO when installing Harvester behind a firewall or proxy, and in environments without internet connectivity.
- **Net install ISO:** Contains only the core operating system components. After installation is completed, the operating system pulls all required container images from the internet (mostly from Docker Hub).

Method	Required Installation Files	Other Requirements
<a href="#">ISO</a>	ISO	N/A
<a href="#">USB</a>	ISO	USB flash drive; utility such as <a href="#">balenaEtcher</a> or the Linux <a href="#">dd command</a>
<a href="#">PXE</a>	ISO, Linux kernel image (vmlinuz), initrd, SquashFS image	Directory on the HTTP server (for serving boot files); iPXE boot scripts (for automatic installation); DHCP server configuration

## 3. Prepare the cluster configuration requirements.

- Cluster token: ASCII string that nodes use when joining the cluster
- Fixed IP address for each node: May be assigned statically or using DHCP (host reservation)
- Fixed virtual IP address (VIP) to be used as the cluster management address: VIP that you connect to when performing administration tasks after the cluster is deployed
- Addresses of DNS servers, NTP servers, and the proxy server (if necessary)

## 4. Deploy the cluster node.

Deployment involves installing the operating system and other components on the host, and then rebooting once installation is completed. Deploying the node creates the cluster, and the node is assigned the

management role by default.

During installation, you must configure node settings, define the cluster management address (VIP) and the cluster token, and specify other information. If necessary, you can configure more settings using a [Harvester configuration](#) file.

Once installation is completed, the node restarts and then the Harvester console appears. The console displays information about the cluster (management URL and status) and the node (hostname, IP address, and status). After the cluster is initialized and all services start running, the cluster status changes to **Ready**.

## 5. Configure a strong password for the default `admin` user on the Harvester UI.

Once the cluster status changes to **Ready**, you can access the [Harvester UI](#) using the management URL displayed on the console.

## 6. Configure the default StorageClass.

Harvester uses StorageClasses to describe how Longhorn must provision volumes. Each StorageClass has a parameter that defines the number of replicas to be created for each volume.

The default StorageClass `harvester-longhorn` has a replica count value of **3** for high availability. If you use `harvester-longhorn` in your single-node cluster, Longhorn is unable to create the default number of replicas, and volumes are marked as *Degraded* on the Harvester UI.

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

- Change the [replica count](#) of `harvester-longhorn` to **1** using a [Harvester configuration](#) file.
- [Create a new StorageClass](#) with the **Number of Replicas** parameter set to **1**. Once created, locate the new StorageClass in the list and then select : > **Set as Default**.

## 7. Create a custom cluster network and a VM network. (Optional)

Networking in Harvester involves three major concepts:

- **Cluster network**: Traffic-isolated forwarding path for transmission of network traffic in the Harvester cluster.

During deployment, a cluster network called `mgmt` is created for intra-cluster communications. `mgmt` allows VMs to be accessed from the infrastructure network (external to the Harvester cluster) to which each Harvester node attaches with management NICs for cluster management purposes. Harvester also allows you to create [custom cluster networks](#) that can be dedicated to VM traffic.

- **Network configuration**: Definition of how cluster nodes connect to a specific cluster network.

Each network configuration corresponds to a set of nodes with uniform network specifications. Only nodes that are covered by the network configuration can access the associated cluster network. This arrangement offers you flexibility when configuring a heterogeneous cluster, particularly when the network interface names are different for each node.

- **VM network**: Virtual network that VMs use to communicate with other VMs and external networks.

Each VM network is linked to a specific cluster network, which is used for transmission of VM traffic. You can create either a [VLAN network](#) or an [untagged network](#) based on your requirements, such as traffic isolation, network segmentation, ease of management, or alignment with the external network environment.

You can create a VM network that uses `mgmt` when testing Harvester with a single-node cluster.

## 8. Import VM images.

On the Harvester UI, you can import ISO, qcow2, and raw [images](#) by uploading an image from the local file system, or by specifying the URL of an image that can be accessed from the cluster.

## 9. Import SSH keys. (Recommended)

You can store SSH public keys in Harvester. When a VM is launched, a stored key can be [injected](#) into the VM to allow secure access via SSH. Validated keys are displayed on the **SSH Keys** screen on the Harvester UI.

## 10. Create VMs.

You can create [Linux VMs](#) using one of the following methods:

- Harvester UI: On the **Virtual Machines** screen, click **Create** and configure the settings on each tab.
- Kubernetes API: Create a `VirtualMachine` object.
- [Harvester Terraform Provider](#): Define a `harvester_virtualmachine` resource block.

Creating [Windows VMs](#) on the Harvester UI involves slightly different steps. Harvester provides a VM template named `windows-iso-image-base-template` that adds a volume with the Virtio drivers for Windows, which streamlines the VM configuration process. If you require Virtio devices but choose to not use the template, you must add your own Virtio drivers for Windows to enable correct hardware detection.

## What's Next

The following sections provide guides that walk you through how to back up and restore VMs, manage hosts, and use Rancher with Harvester.

- [VM Backup, Snapshot & Restore](#)
- [Host Management](#)
- [Rancher Integration](#)