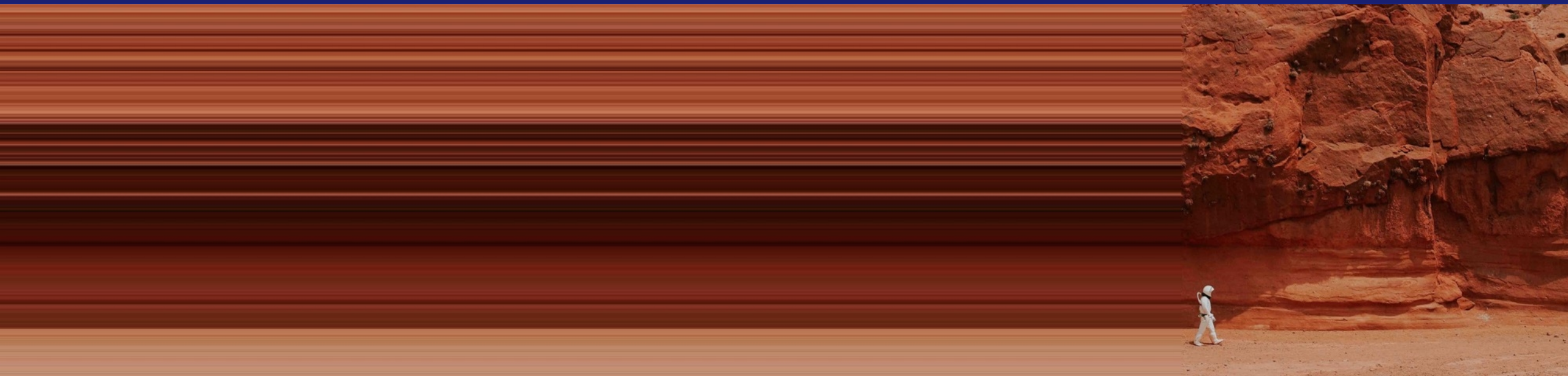
SUSE

eHRV101:

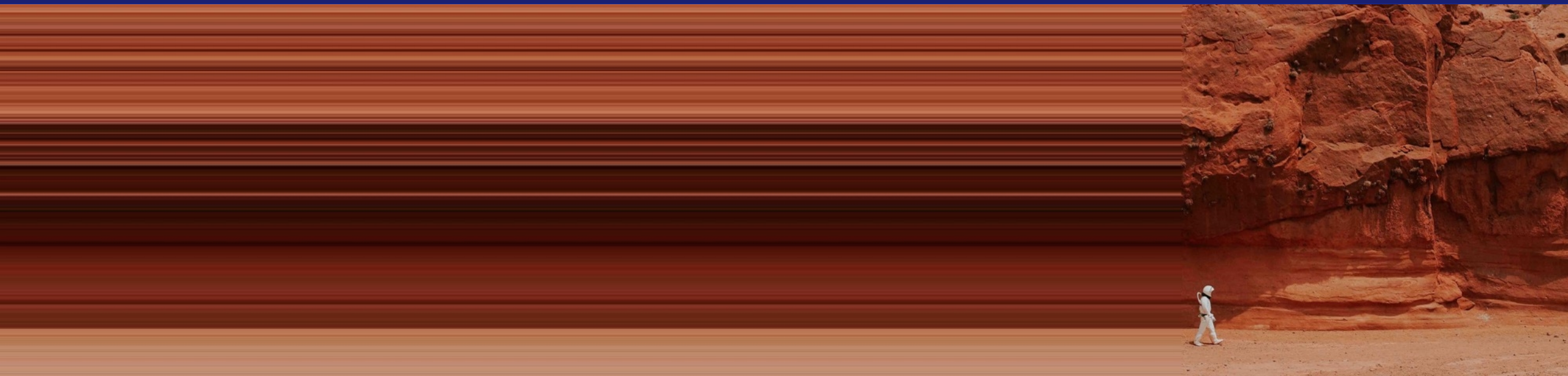
Introduction to Harvester



SUSE

Section 1:

Course Introduction



Topics

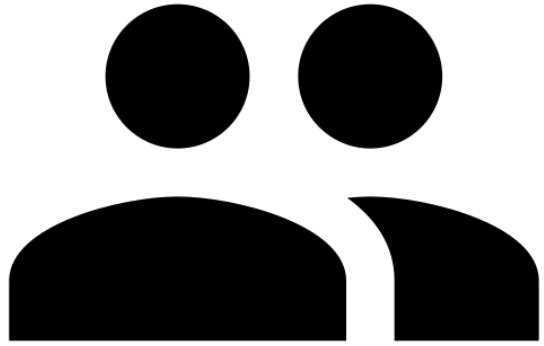
Course Objectives and Audience

Course Overview

- Provide an introduction and overview of the Harvester hyperconverged infrastructure (HCI) platform.
- Discuss and demonstrate the deployment of Harvester.
- Use Harvester to perform basic virtualization management tasks.
- Integrate Harvester into Rancher and deploy Kubernetes clusters onto Harvester using Rancher.

Audience

- Anyone interested in learning about the Harvester HCI platform.
- System administrators interested in deploying an HCI platform for Kubernetes cluster deployment.
- System administrators interested in deploying an HCI platform for general purpose virtualization.



Course Prerequisites

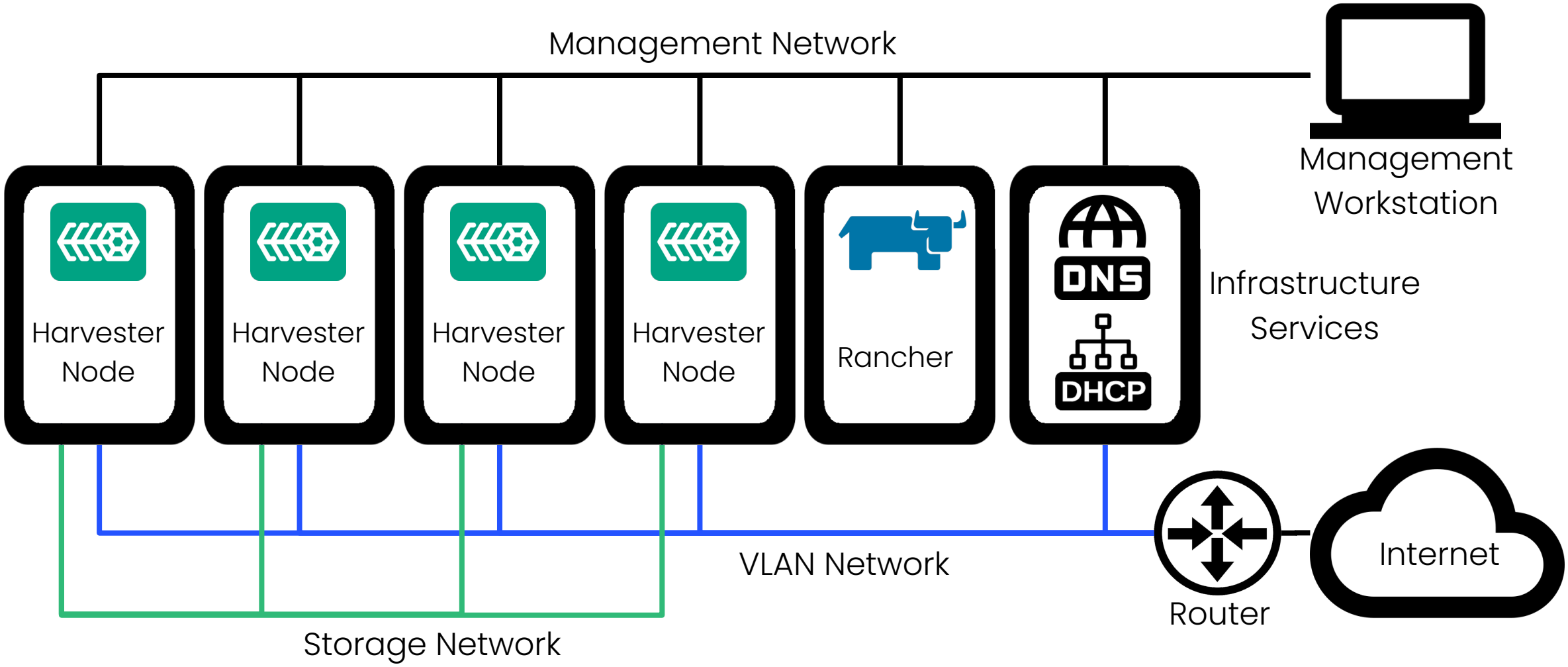
- Experience with Linux administration is preferred.
- Experience with virtualization administration, preferably Libvirt+KVM, is beneficial.
- Experience with Kubernetes administration is not required but beneficial.

Course Agenda

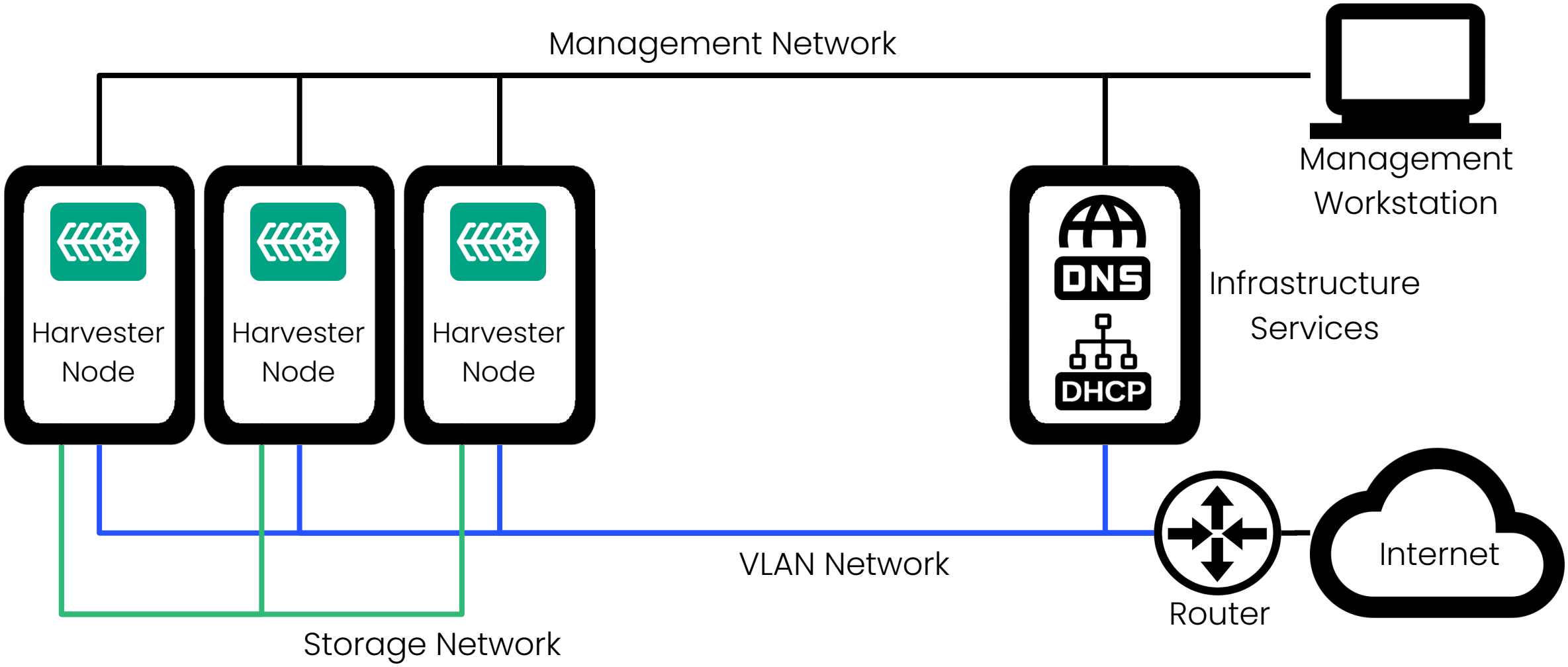
- Section 1: Course Introduction
- Section 2: Introduction to Harvester
- Section 3: Harvester Cluster Deployment
- Section 4: General Harvester Configuration
- Section 5: Networking in Harvester
- Section 6: Storage in Harvester
- Section 7: Virtualization Management with Harvester
- Section 8: Harvester Maintenance, Troubleshooting Tools and Cluster Upgrade
- Section 9: Harvester and Rancher Integration

Course Lab Environment Overview

Lab Environment Diagram - Main



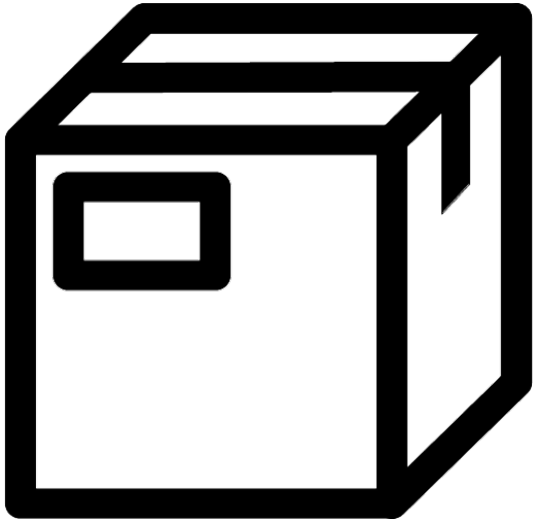
Lab Environment Diagram - Deployment/Upgrade



Required Minimum Product Version

This course is based on the following product version

Product: Harvester

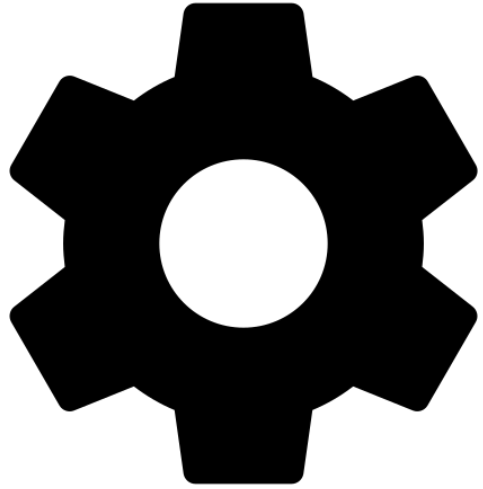


Version: 1.2.0

This is the minimum version required to run the course.
The material in the course may apply to subsequent versions as well.

Lab Environment Hardware / Software Requirements

The lab environment was developed and tested on the following hardware/OS specification:



Physical Cluster Nodes

CPU: 8 Core Intel or AMD x86_64

RAM: 32GB

Disk: 1TB NVMe, 500GB SSD

Host OS: Harvester (installed on bare metal)

VM Host (Running Management and Rancher VMs)

CPU: 8 Core Intel or AMD x86_64

RAM: 32GB

Disk: 1TB NVMe

Host OS: openSUSE Leap 15.5

Additional SUSE Training

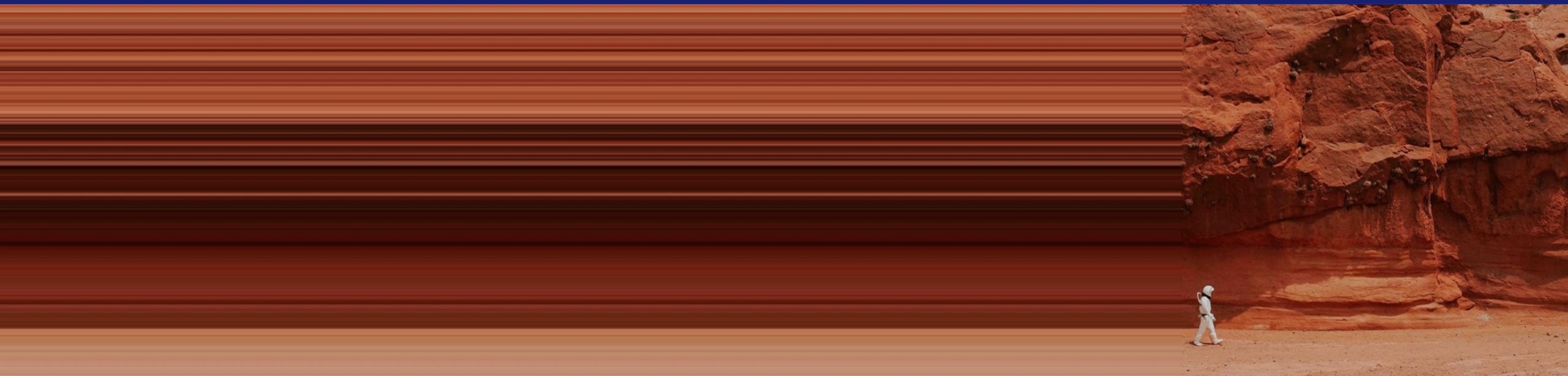
Additional SUSE Training

- SUSE Training is available across a wide range of products across our Business Critical Linux, Enterprise Container Management and Edge solutions.
- For more information:
  - https://www.suse.com/training

SUSE

Section: 2

Introduction to Harvester



Learning Objectives

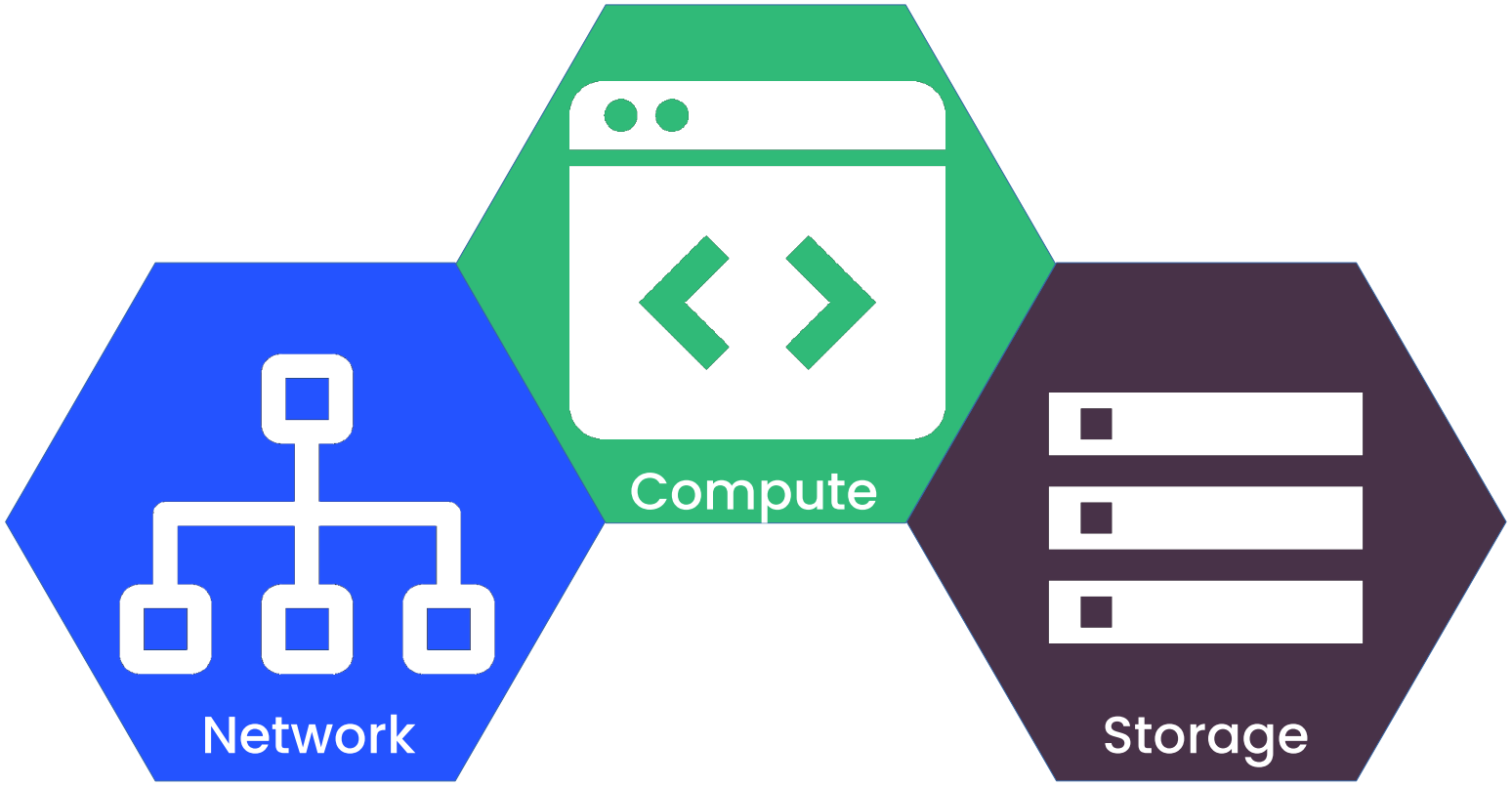
At the end of this section you will be able to:

- Describe the basic concepts of Hyperconverged Infrastructure.
- Describe the basic features and functionality of Harvester.

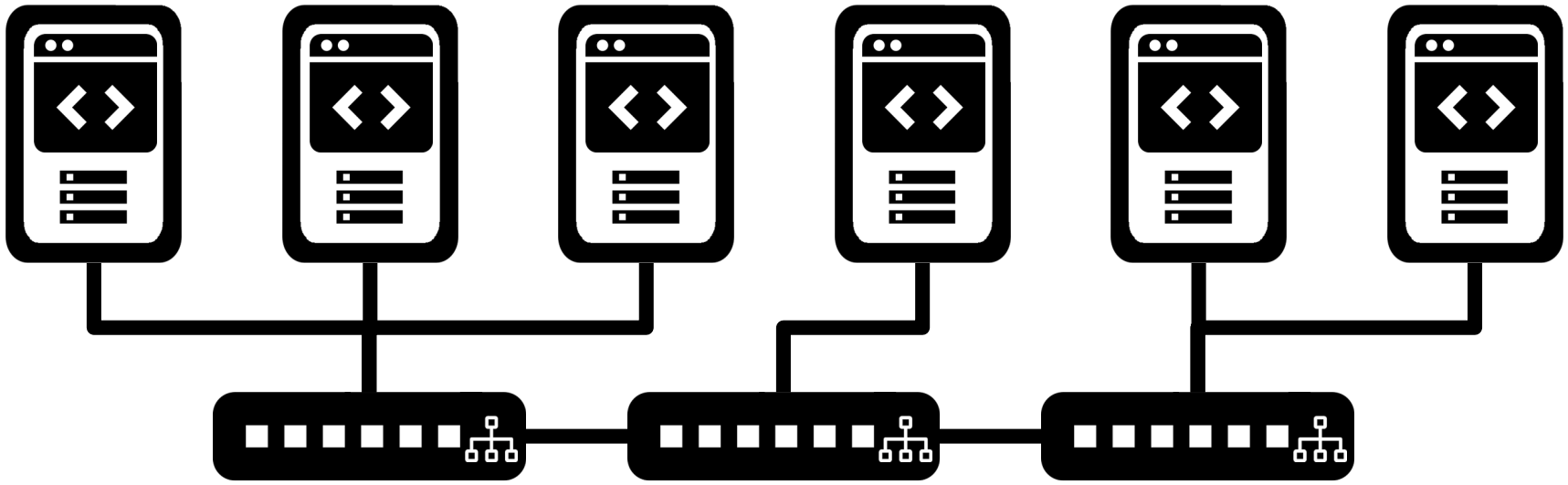
Topics

Hyperconverged Infrastructure Concepts

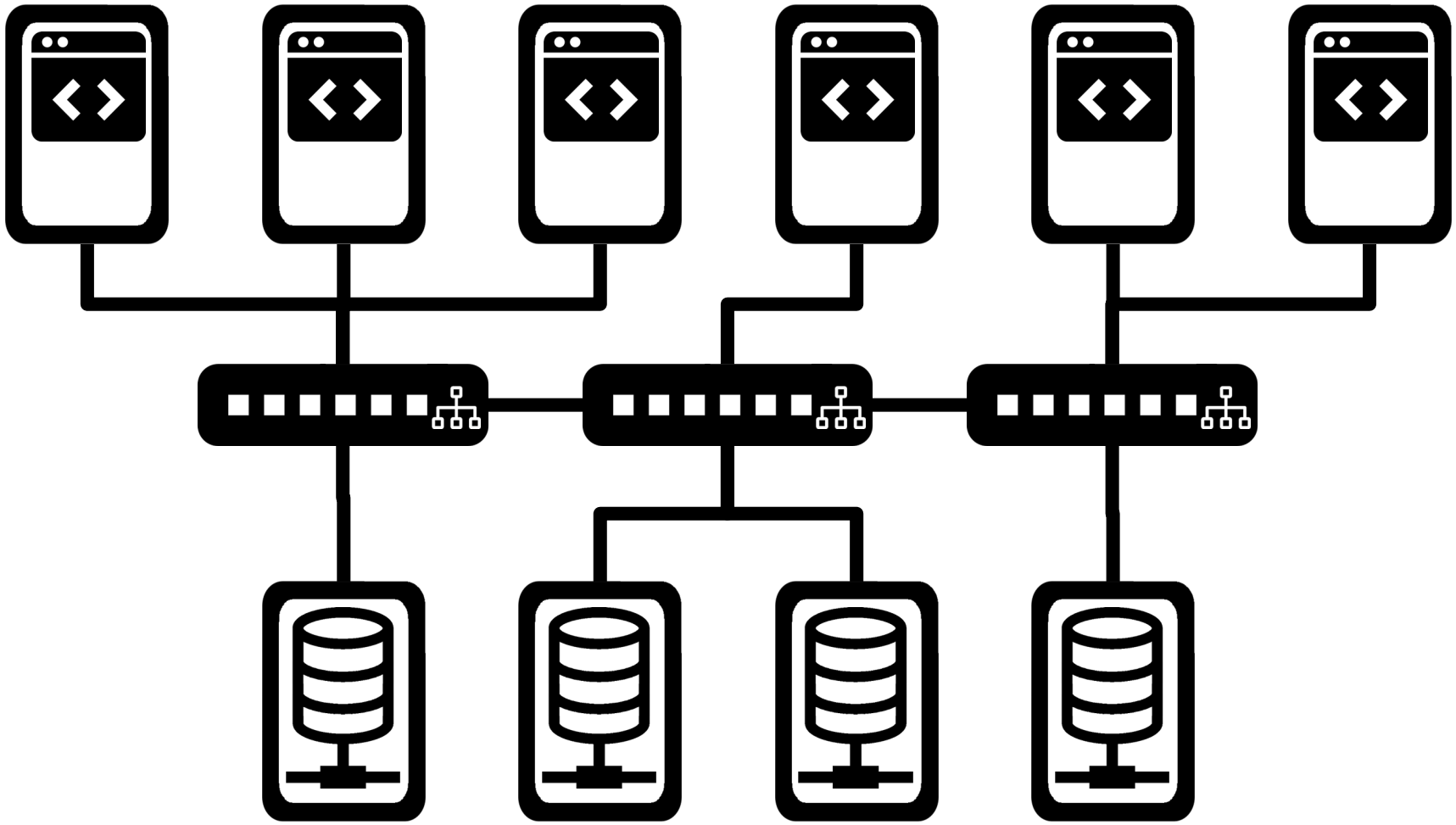
Fundamental Areas of Computing



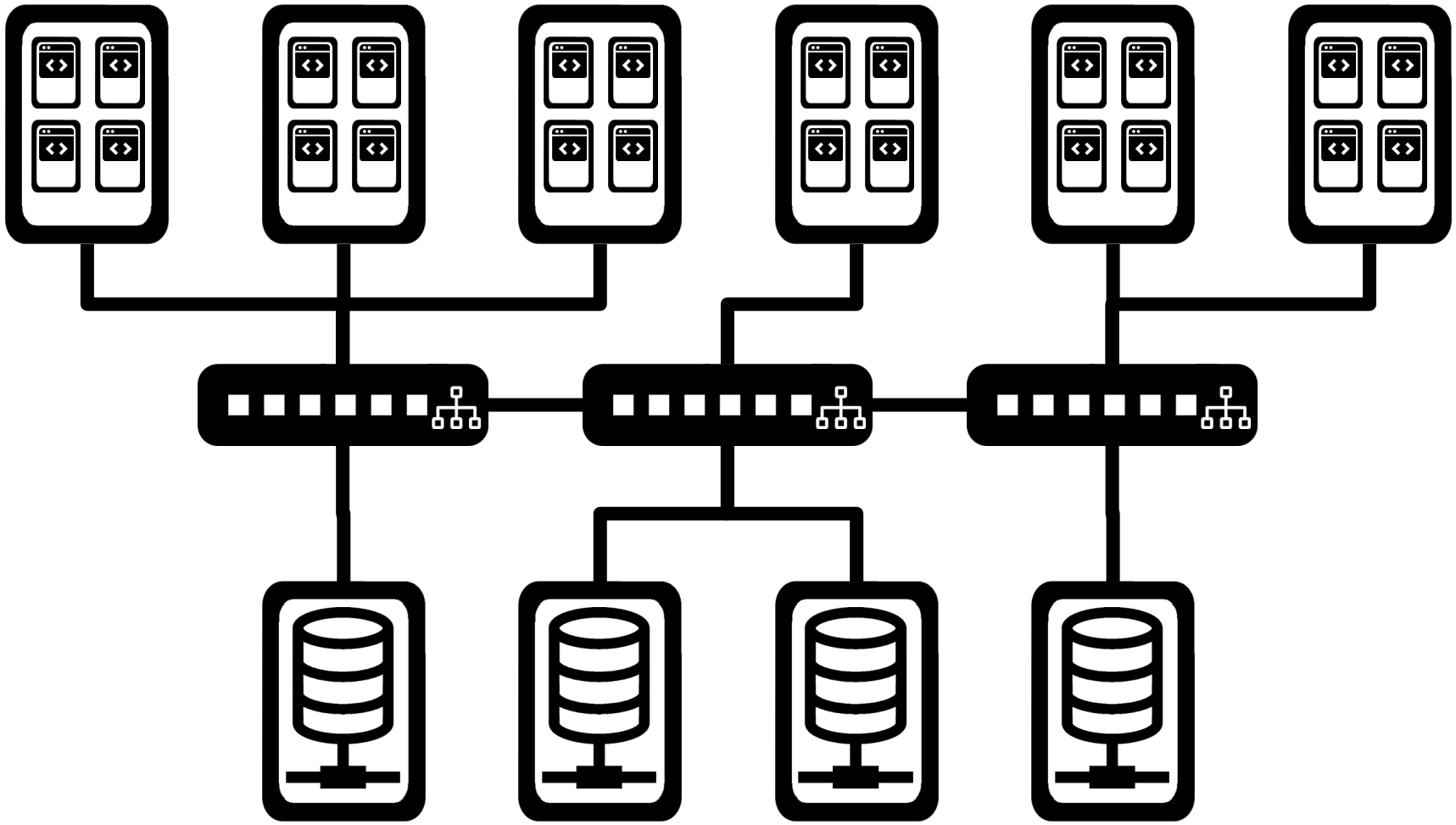
Traditional Architecture



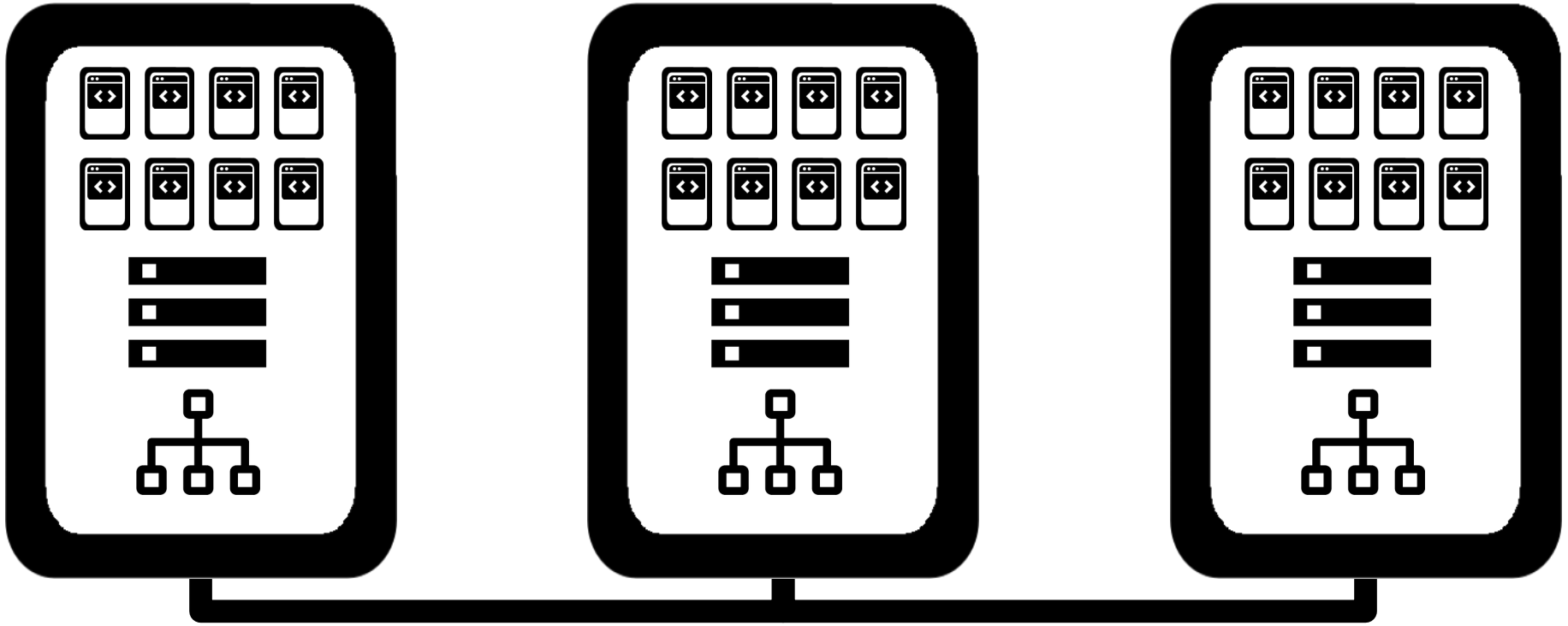
Traditional Architecture with Network Storage



Virtualized Architecture



Hyperconverged Architecture



Why Use a Hyperconverged Infrastructure?

- Simplified Design
- Scalability
- Data Protection

Why Use a Hyperconverged Infrastructure?

Simplified Design

- Single vendor nature minimizes integration and compatibility headaches.
- Small number of standardized building blocks provide simplicity.
- Reduced need for specialty skills such as storage and networking.

Why Use a Hyperconverged Infrastructure?

Scalability

- Predefined building blocks make it easy to add capacity as needed.
- Start small and scale up as needed.
- Software defined nature allows for agility and adaptability.
- Easier upgrades due to fewer parts and vendors.

Why Use a Hyperconverged Infrastructure?

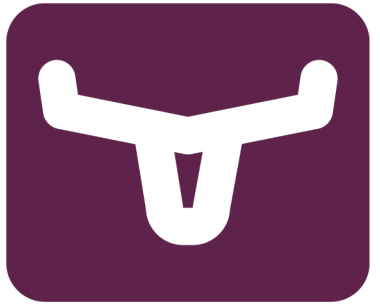
Data Protection

- Built-in storage redundancy.
- Multi-node architecture is more fault tolerant.
- Workloads decoupled from underlying hardware allow easy migration.
- Snapshot, clones and backups allow for easy recovery.

Harvester Features and Functionality

What is Harvester?

Harvester is a modern hyperconverged infrastructure (HCI) solution built for bare metal servers using enterprise-grade open source technologies including Kubernetes, KubeVirt and Longhorn.



What is Harvester?

- Harvester implements HCI on bare metal servers.
- Harvester is designed to use local, direct attached storage instead of complex external Storage Area Networks (SANs).
- Harvester is designed to use standard VLAN and untagged networking infrastructure.
- Harvester uses proven open source virtualization technologies.
- Harvester uses Kubernetes to provide virtualization management and a standard management API.

Components of Harvester

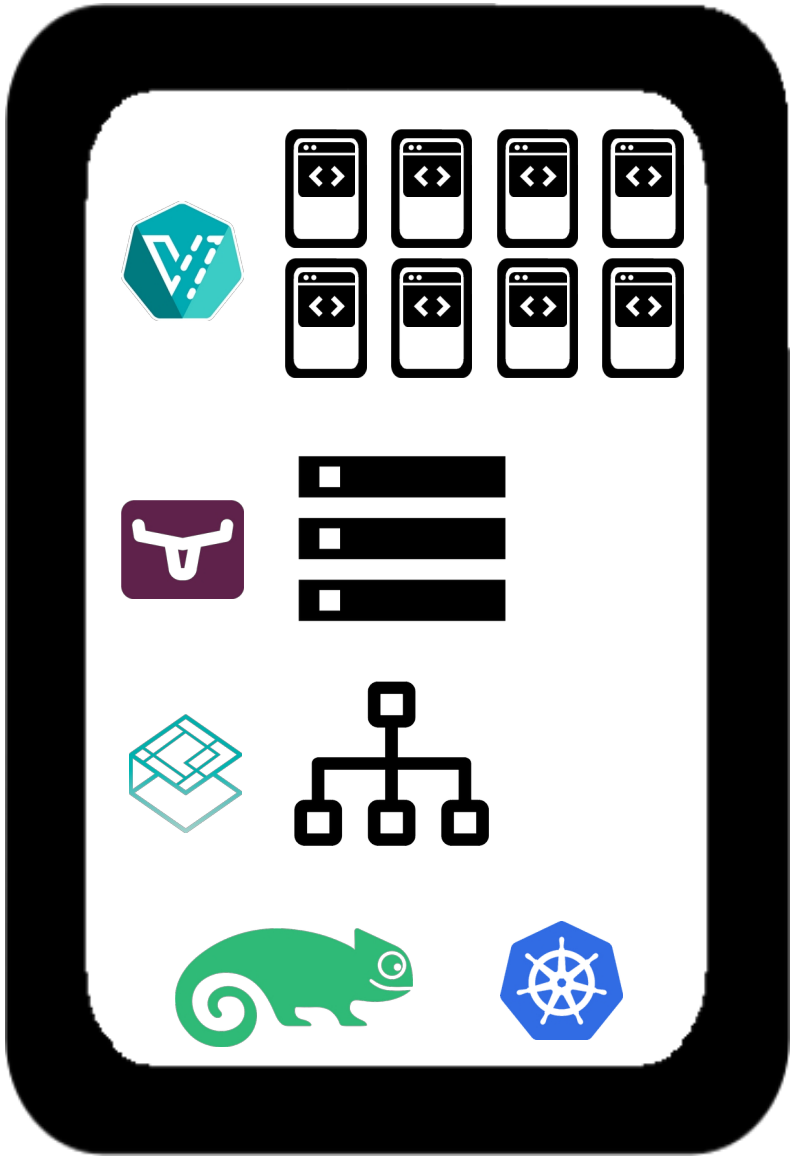
Virtualization Layer: KubeVirt

Storage Layer: Longhorn

Network Layer: Kubernetes CNIs
(Container Network Interfaces)

Orchestration Layer: Kubernetes

Operating System: SLE-Micro

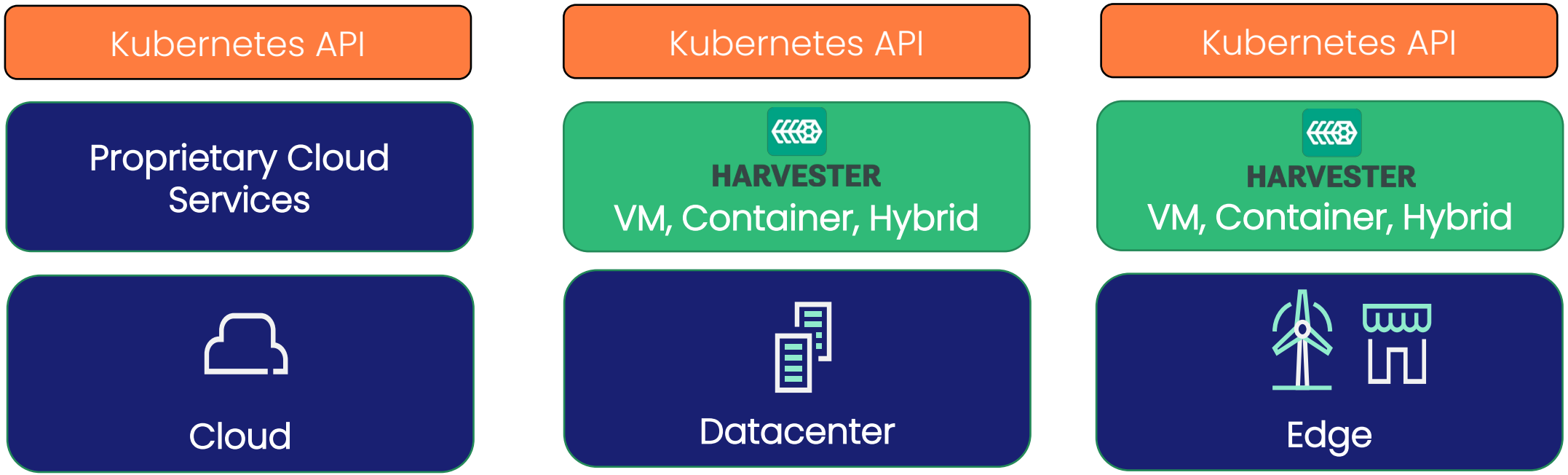


Hardware and Infrastructure Requirements

CPU:	x86_64, 8 Core minimum, 16 core or more recommended
Memory:	32 GB minimum, 64 GB or more recommended
Disk Capacity:	250 GB minimum, 500 GB or more recommended
Disk Performance:	5000+ random IOPS (SSD/NVMe)
Network Speed:	1 Gbps minimum, 10 Gbps recommended
Network Interfaces:	2 interfaces recommended (one for Management one for VLANS) Additional recommended for use with network bonding and dedicated storage network
Network Switch:	Trunking of ports required for VLANs

Why Harvester?

- Kubernetes becomes the standard API.
- GitOps and cloud native tooling can manage the entirety of your infrastructure.
- Both VMs and containers can be managed from the same place.



Target Use Cases for Harvester

Primary Use Case:

- Underlying virtualization infrastructure for deploying Kubernetes clusters

Secondary Use Cases:

- General Purpose HCI platform
- Kubernetes platform for running both VMs and containers

Harvester: The Open, Interoperable HCI Solution

Unify your existing virtualized infrastructure and accelerate the adoption of containers from core to edge.

Next-gen, turnkey hyperconverged solution without ‘the vTax’

Bridging the gap between legacy and cloud-native IT with Rancher integration

Reduces reliance on costly, boutique SAN or HCI hardware

100% open-source HCI solution with zero platform dependency

Section Summary

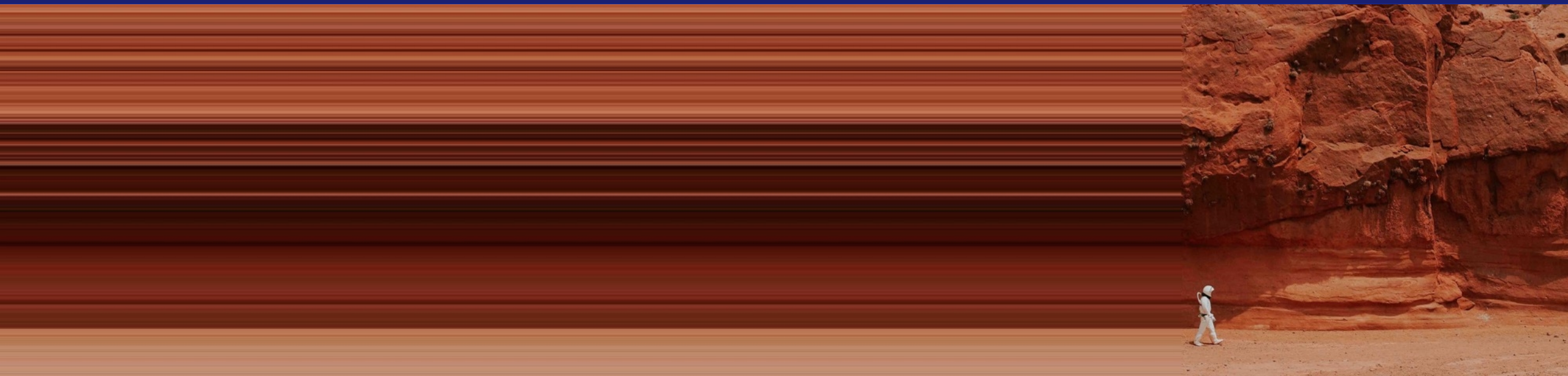
Section Summary

- Describe Hyperconverged Infrastructure Concepts
  - Hyperconverged Infrastructure (HCI) is a platform that combines compute/virtualization, network and storage components onto a single infrastructure node.
  - The capacity of HCI platforms is extended by adding additional nodes.
  - HCI uses software defined storage/network/etc. to improve agility, flexibility and ease of configuration.
- Describe Harvester Features and Functionality
  - Harvester uses Kubernetes for the underlying orchestration layer which in turn provides a standard management API.
  - Harvester uses Longhorn for the storage layer, KubeVirt for the compute layer and Kubernetes CNIs for the networking layer.

SUSE

Section: 3

Harvester Cluster Deployment



Learning Objectives

At the end of this section you will be able to:

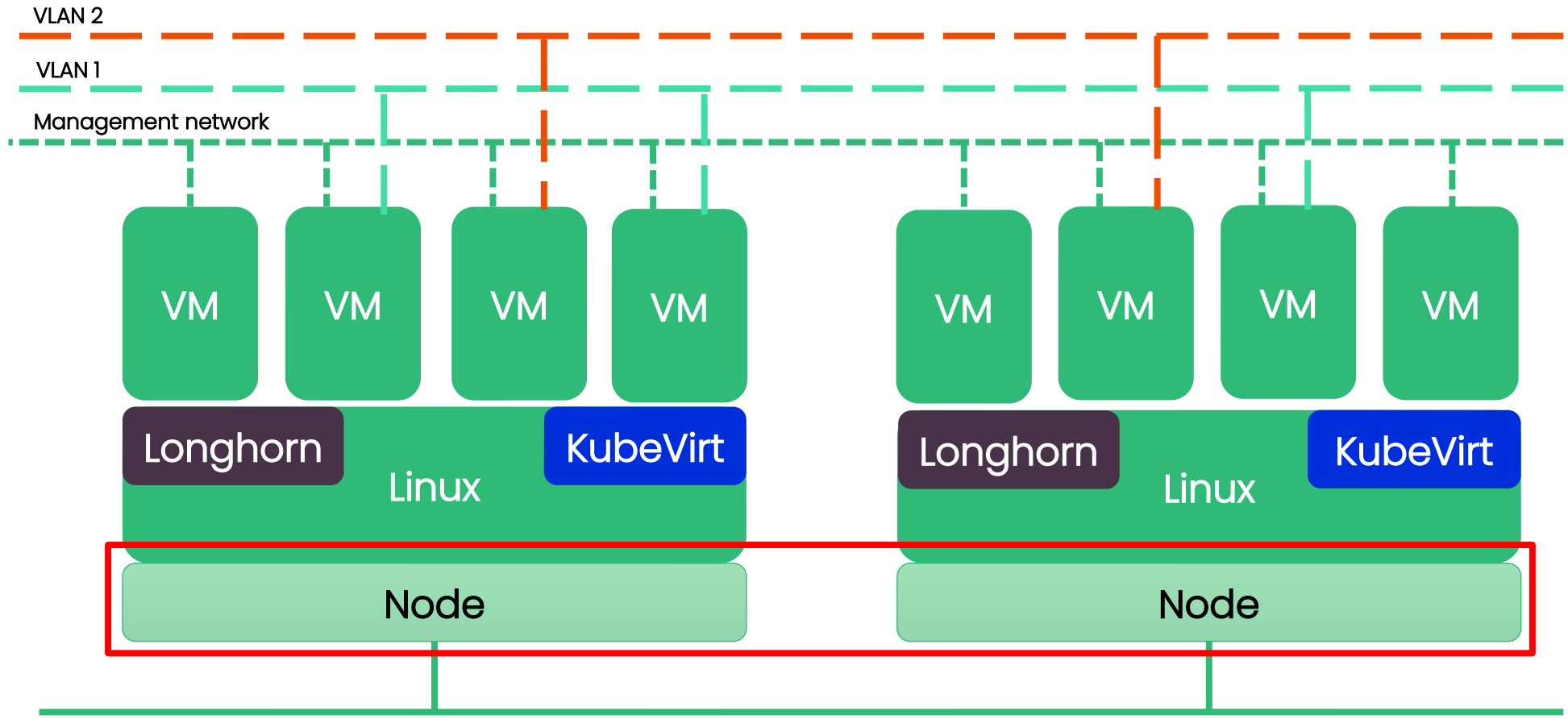
- Manually deploy a Harvester cluster.
- Automate the deployment of a Harvester cluster.

Topics

The Harvester Installation Process

Harvester Is a Virtualization Platform

- Harvester is designed to be installed on bare metal.
- Harvester provides a platform to run virtual machines.



Harvester Installation Methods

- Manual installation from ISO
- Manual installation from USB
- Manual installation from the network

Harvester Deployment Process

- Create the cluster by deploying the initial cluster node.
- Additional cluster nodes can then be deployed and joined to the cluster.

Note: The first 3 cluster nodes will act as the controller nodes for the cluster and run processes such as etcd and the Kubernetes API. Any additional nodes will just be workers.

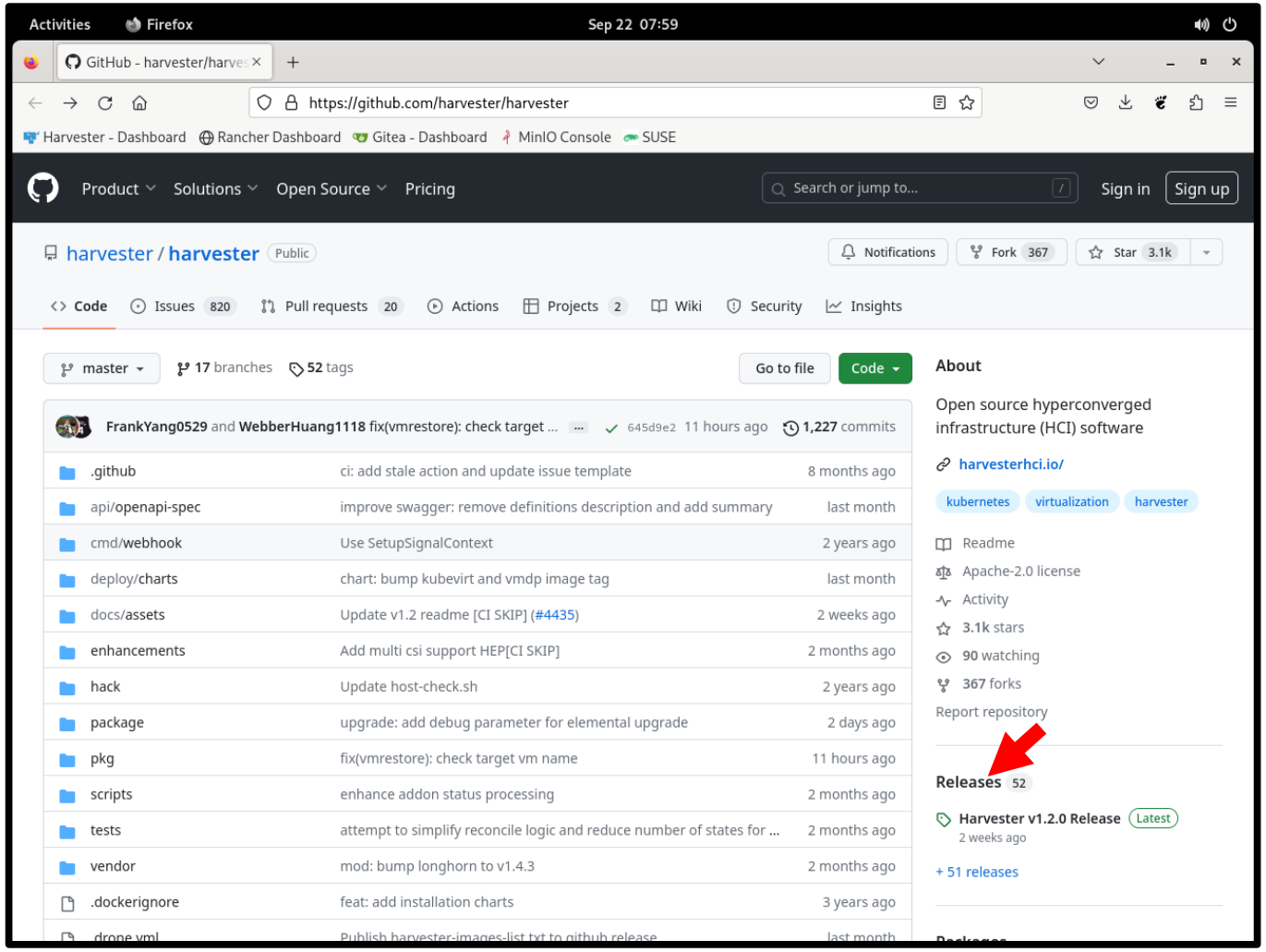
Harvester Cluster Deployment Preparation

Download Installation Media

- The installation files are available on the Harvester GitHub page:

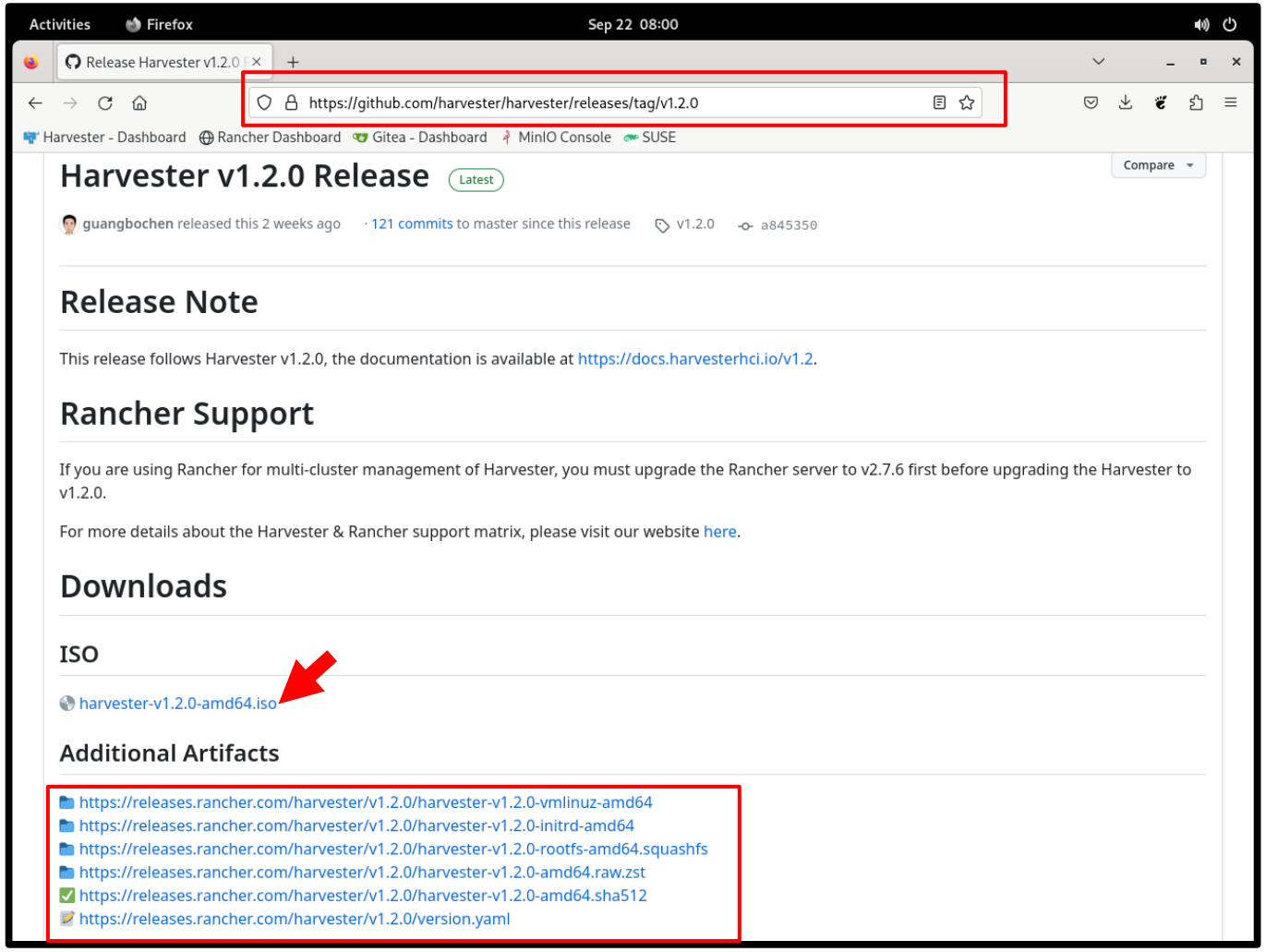
  http://github.com/harvester/harvester

- Click on **Releases** to go to where the files are posted.



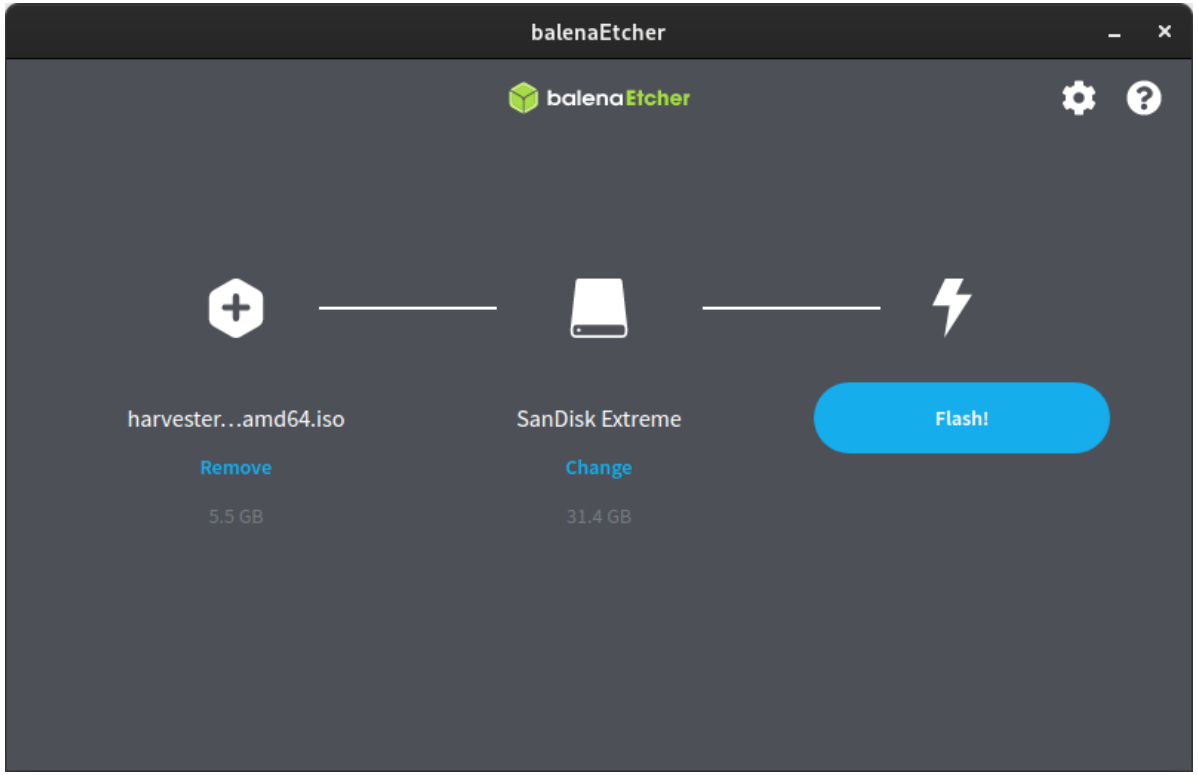
Download Installation Media

- On the **Releases** page download the Harvester installation **ISO**.
- Download the **Additional Artifacts** if you will be using the network boot method of installation.



Create Installation USB

- Use a tool such as balenaEtcher to write the ISO image to a USB disk.
- The USB disk can then be used to boot the machine being installed in the same manner as an installation ISO.



Network Booting - iPXE

- The recommended method for network booting a Harvester installation is iPXE (https://ipxe.org).

- iPXE provides some benefits over traditional PXE:
  - Allows for retrieval of files via HTTP/HTTPS.
  - Easier configuration.
  - More feature rich configuration options.

Network Booting - Installation Directory Structure

Harvester:

- Create a directory on the web server to share the Harvester installation files.

iPXE:

- Create a directory on the web server to share the iPXE files.

```
/var/lib/rmt/public/repo/Install/Harvester/
|          |-harvester-v1.2.0-amd64.iso
|          |-harvester-v1.2.0-initrd-amd64
|          |-harvester-v1.2.0-rootfs-amd64.squashfs
|          |-harvester-v1.2.0-vmlinuz-amd64
|
|ipxe/
     |-ipxe-boot
     |-ipse-boot-efi
     |-ipxe-x86_64.efi
```

iPXE Config

- Edit the iPXE config file to:
  - Create a menu item for the manual install.
  - Create a config block with the info required for network booting.
- The following parameters are essential in the config block:
  - `kernel`
  - `root`
  - `harvester.install.iso_url`
  - `initrd`

```
#!ipxe
set install_server 172.31.9.2
set install_base http://${install_server}/repo/Install

set hrv_ver v1.2.0
set hrv_arch amd64
set hrv_base ${install_base}/Harvester


menu iPXE Install Server - Choose an option
item localboot                              Boot from local disk
item --gap --                               --- Harvester Installation Options ---
item harvester_install_manual               Harvester ${hrv_ver} Installation (manual)
choose --default localboot --timeout 5000 target && goto ${target}

:localboot
sanboot --no-describe --drive 0x80

:harvester_install_manual
kernel ${hrv_base}/harvester-${hrv_ver}-vmlinuz-${hrv_arch} ip=dhcp net.ifnames=1
rd.cos.disable rd.noverifyssl console=tty1 root=live:${hrv_base}/harvester-${hrv_ver}-
rootfs-${hrv_arch}.squashfs harvester.install.iso_url=${hrv_base}/harvester-${hrv_ver}-$
{hrv_arch}.iso
initrd ${hrv_base}/harvester-${hrv_ver}-initrd-${hrv_arch}
boot
```

DHCP Config

- Edit the **dhcpd.conf** file to add a group for the iPXE booing.
- The **PXE Boot** section specifies the PXE binaries to use chainload iPXE.
- The **HTTP Boot** section specifies the PXE binary to use to chainload iPXE if the system is using direct HTTP booting via UEFI.
- The **iPXE Boot** section references the config files to use depending on if the system boots via UEFI or not.

```
group {
  # id="ipxe"
  if exists user-class and option user-class = "iPXE" {
    # iPXE boot
    if option architecture-type = 00:07 {
      filename "http://172.31.9.1/repo/Install/ipxe/ipxe-boot-efi";
    } else {
      filename "http://172.31.9.1/repo/Install/ipxe/ipxe-boot";
    }
  } elsif substring (option vendor-class-identifier, 0, 10) = "HTTPClient" {
    # UEFI HTTP Boot
    option vendor-class-identifier "HTTPClient";
    filename "http://172.31.9.1/repo/Install/ipxe/ipxe-x86_64.efi";
  } else {
    # PXE boot
    if option architecture-type = 00:07 {
      # UEFI
      filename "ipxe-x86_64.efi";
    } else {
      # Non-UEFI
      filename "undionly.kpxe";
    }
  }
```

DHCP Config

- **Host reservations** can be added to the **iPXE group** in the dhcpd.conf file to assign persistent IP addresses to the cluster nodes and to enable them to PXE boot using the iPXE configuration.

```
group {
  # id="ipxe"
  if exists user-class and option user-class = "iPXE" {
    # iPXE boot
    if option architecture-type = 00:07 {
      filename "http://172.31.9.1/repo/Install/ipxe/ipxe-boot-efi";
    } else {
      filename "http://172.31.9.1/repo/Install/ipxe/ipxe-boot";
    }
  } elsif substring (option vendor-class-identifier, 0, 10) = "HTTPClient" {
    # UEFI HTTP Boot
    option vendor-class-identifier "HTTPClient";
    filename "http://172.31.9.1/repo/Install/ipxe/ipxe-x86_64.efi";
  } else {
    # PXE boot
    if option architecture-type = 00:07 {
      # UEFI
      filename "ipxe-x86_64.efi";
    } else {
      # Non-UEFI
      filename "undionly.kpxe";
    }
  }

  host node01 {
    option host-name "node01";
    fixed-address 172.31.9.31;
    hardware ethernet 52:54:00:0c:31:01;
  }
  host node02 {
    option host-name "node02";
    fixed-address 172.31.9.32;
    hardware ethernet 52:54:00:0c:32:01;
  }
  host node03 {
    option host-name "node03";
    fixed-address 172.31.9.33;
    hardware ethernet 52:54:00:0c:33:01;
  }
```

Cluster Configuration Requirements

- A cluster token that will be used by nodes to join the cluster.
- IP addresses for each node that will not change.
- An IP address to use as the cluster management address.
- Addresses of DNS servers, NTP servers and, if required, the Proxy server.

Harvester Cluster Manual Deployment

Step 1 - Install First Cluster Node

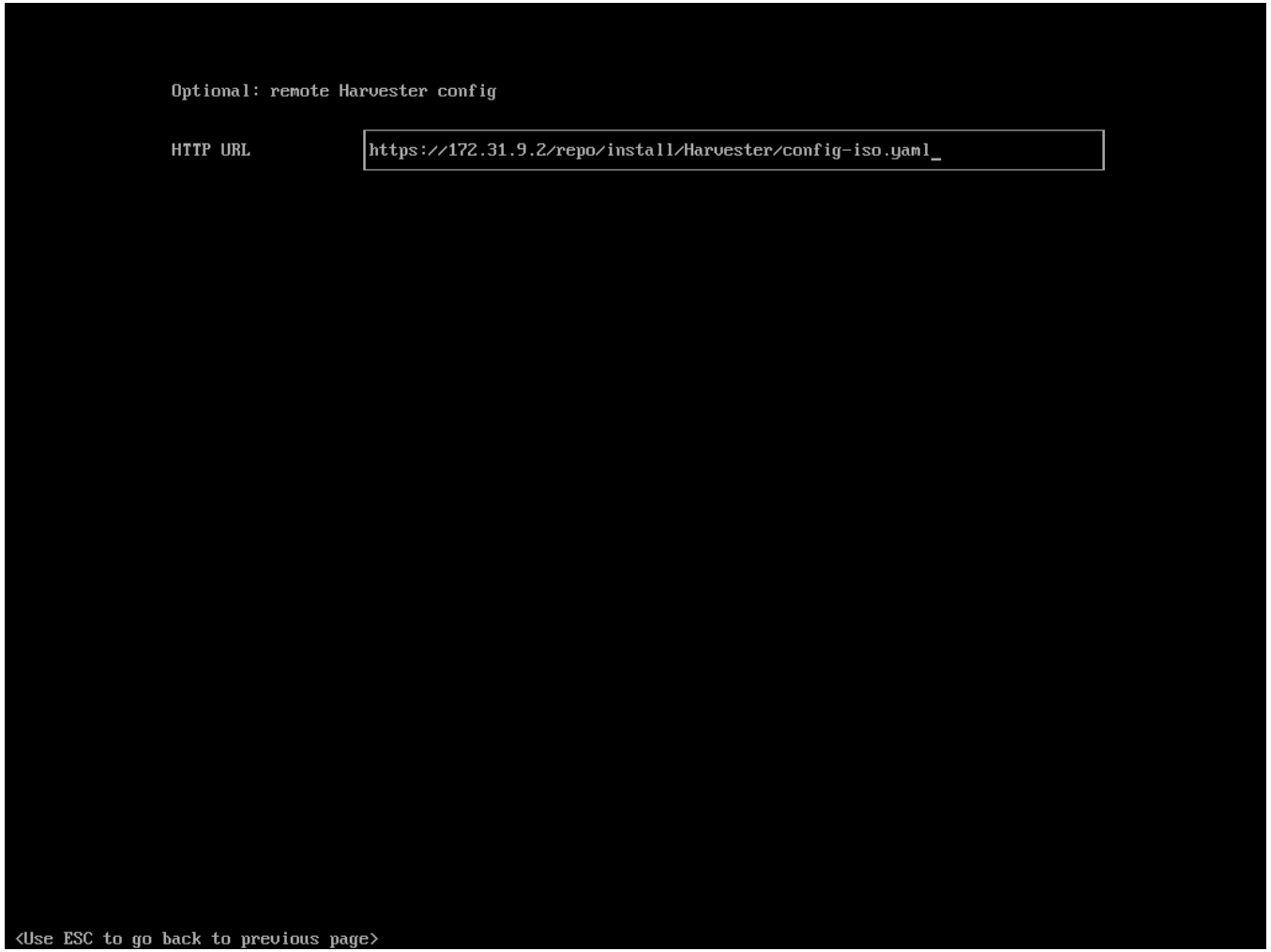
- Installation of the first cluster nodes creates the cluster.
- Installation is started by booting from the installation media in one of the supported methods.
- Installation of the first node entails:
  - Specifying node specific configuration (OS/data disk, Networking, local password, etc.)
  - Defining the Cluster Token
  - Defining the cluster management/virtual IP (VIP) address

Step 2 - Install Additional Cluster Nodes

- All additional nodes join the cluster created on the first node.
- Installation is started by booting from the installation media in one of the supported methods.
- Installation of additional nodes entails:
  - Specifying node specific configuration (OS/data disk, Networking, local password, etc.).
  - Providing the Cluster Token and cluster management/VIP address to join the cluster.

Additional Configuration During Installation

- Any additional custom installation configuration can be supplied via **remote Harvester config**.
- The remote in **remote Harvester config** refers to the configuration information residing remotely and has to be retrieved before it can be processed and integrated with the manually provided configuration.
- The configuration must be in YAML format and follows the same spec used when performing an automated installation.



Remote Config Example: Installation ISO

When installing manually from the network, the location of the Harvester installation ISO must be provided as remote config.

```
install:
  iso_url: http://172.31.9.2/repo/Install/Harvester/harvester-v1.2.0-amd64.iso
```

Installation Complete - Cluster Ready

- When a node installation is complete the node will reboot.
- Once the cluster services have all started and are running, the cluster status is: Ready
  - Note: The first cluster node will take a bit longer to reach this state than the additional cluster nodes as it has to initialize the cluster.

Harvester Cluster Automated Deployment

Automate Harvester Deployment

- The installation of the Harvester cluster nodes and the creation of a new Harvester cluster can be automated when using the network/PXE installation method.
- Config files used to automate the installation are formatted using standard YAML.
- The YAML config files can be used to fully automate the deployment or to provide additional configuration parameters during a manual install.

Installation Directory Structure - Automation

- Create the YAML config files for the creation and joining in the same directory as the other Harvester installation files.
- Edit the ipxe-boot files to add the sections referencing the YAML files.

```
/var/lib/rmt/public/repo/Install/Harvester/
                                          |          |-config-create-harvester01.yaml
                                          |          |-config-join-harvester01.yaml
                                          |          |-harvester-v1.2.0-amd64.iso
                                          |          |-harvester-v1.2.0-initrd-amd64
                                          |          |-harvester-v1.2.0-rootfs-amd64.squashfs
                                          |          |-harvester-v1.2.0-vmlinuz-amd64
                                          |
                                          |ipxe/
                                               |-ipxe-boot
                                               |-ipse-boot-efi
                                                |-ipxe-x86_64.efi
```

Cluster Create YAML File

- The config file for the initial cluster node must contain all required options provided during a manual install plus any additional options supplied via remote config.
- The **`token`** specified here is used by the additional nodes to join the cluster.
- Parameters specific to this file are the **`vip`** related parameters.

```
scheme_version: 1
token: harvester01
os:
  ssh_authorized_keys:
  - <ssh public key goes here>
  dns_nameservers:
  - 172.31.9.2
  - 172.31.9.1
  - 8.8.8.8
  ntp_servers:
  - 0.us.pool.ntp.org
  - 1.us.pool.ntp.org
  password: linux
install:
  mode: create
  management_interface:
    interfaces:
    - name: enp12s0
    default_route: true
    method: dhcp
    bond_options:
      mode: balance-tlb
      miimon: 100
  device: /dev/nvme0n1
  data_disk: /dev/sda
  iso_url: http://172.31.9.2/repo/Install/Harvester/harvester-v1.2.0-amd64.iso
  vip: 172.31.9.30
  vip_mode: static
```

Cluster Join YAML File

- The config file for the additional cluster nodes must contain all required options provided during a manual install plus any additional options supplied via remote config.
- A parameter specific to this file is the **`server_url`**.
- The **`token`** here must match the one specified in the initial node's co

```
scheme_version: 1
server_url: https://172.31.9.30:443
token: harvester01
os:
  ssh_authorized_keys:
  - <ssh public key goes here>
  dns_nameservers:
  - 172.31.9.2
  - 172.31.9.1
  - 8.8.8.8
  ntp_servers:
  - 0.us.pool.ntp.org
  - 1.us.pool.ntp.org
  password: linux
install:
  mode: join
  management_interface:
    interfaces:
    - name: enp62s0u1
    default_route: true
    method: dhcp
    bond_options:
      mode: balance-tlb
      miimon: 100
  device: /dev/nvme0n1
  data_disk: /dev/sda
  iso_url: http://172.31.9.2/repo/Install/Harvester/harvester-v1.2.0-amd64.iso
```

iPXE Config

- Separate menu items and config blocks are specified for the initial and additional cluster node boot entries.
- Each entry specifies the corresponding YAML config file.
- The parameter:

  `harvester.install.automatic=true`

  is used to put the installer into automatic mode.

```
#!ipxe
set install_server 172.31.9.2
set install_base http://${install_server}/repo/Install

set hrv_ver v1.2.0
set hrv_arch amd64
set hrv_base ${install_base}/Harvester


menu iPXE Install Server - Choose an option
item localboot                              Boot from local disk
item --gap --                               --- Harvester Installation Options ---
item harvester_install_auto_create          Harvester Auto Installation (create)
item harvester_install_auto_join            Harvester Auto Installation (join)
choose --default localboot --timeout 5000 target && goto ${target}

:localboot
sanboot --no-describe --drive 0x80


:harvester_install_auto_create
kernel ${hrv_base}/harvester-${hrv_ver}-vmlinuz-${hrv_arch} ip=dhcp net.ifnames=1
rd.cos.disable rd.noverifyssl console=tty1 root=live:${hrv_base}/harvester-${hrv_ver}-
rootfs-${hrv_arch}.squashfs harvester.install.automatic=true harvester.install.config_url=$
{hrv_base}/config-create-harvester01.yaml
initrd ${hrv_base}/harvester-${hrv_ver}-initrd-${hrv_arch}
boot

:harvester_install_auto_join
kernel ${hrv_base}/harvester-${hrv_ver}-vmlinuz-${hrv_arch} ip=dhcp net.ifnames=1
rd.cos.disable rd.noverifyssl console=tty1 root=live:${hrv_base}/harvester-${hrv_ver}-
rootfs-${hrv_arch}.squashfs harvester.install.automatic=true harvester.install.config_url=$
{hrv_base}/config-join-harvester01.yaml
initrd ${hrv_base}/harvester-${hrv_ver}-initrd-${hrv_arch}
boot
```

Section Summary

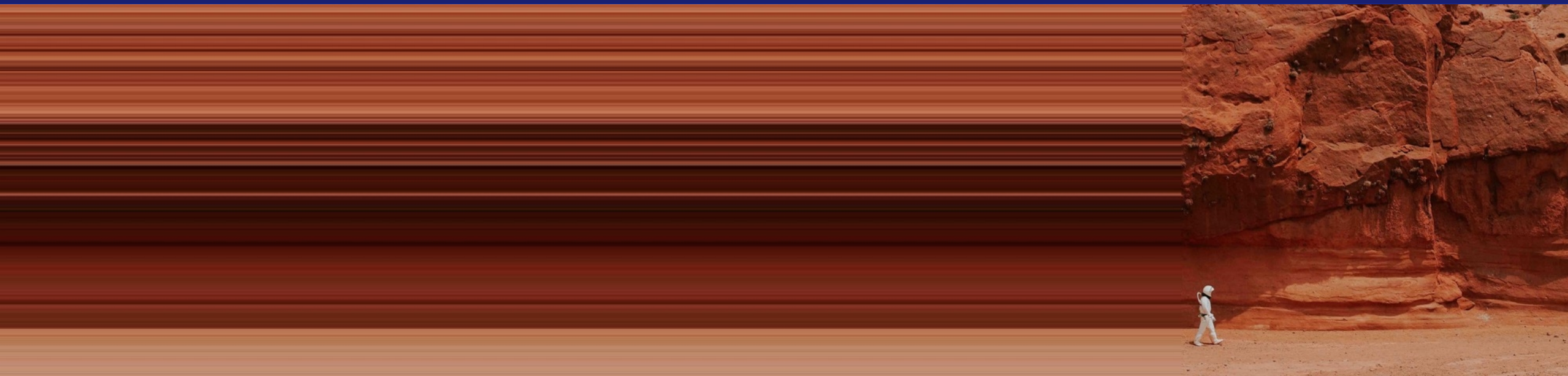
Section Summary

- Manually deploy a Harvester cluster
  - Harvester can be deployed from ISO, USB or via the Network.
  - Installing the first node of a cluster creates the cluster.
  - Additional nodes must join the cluster created on the first node when they are installed.
- Automate the deployment of a Harvester cluster
  - YAML files can be used at installation time to fully automate the installation and cluster joining process.
  - These YAML files can also be used during manual installs to supply additional installation configuration.

SUSE

Section: 4

General Harvester Configuration



Learning Objectives

At the end of this section you will be able to:

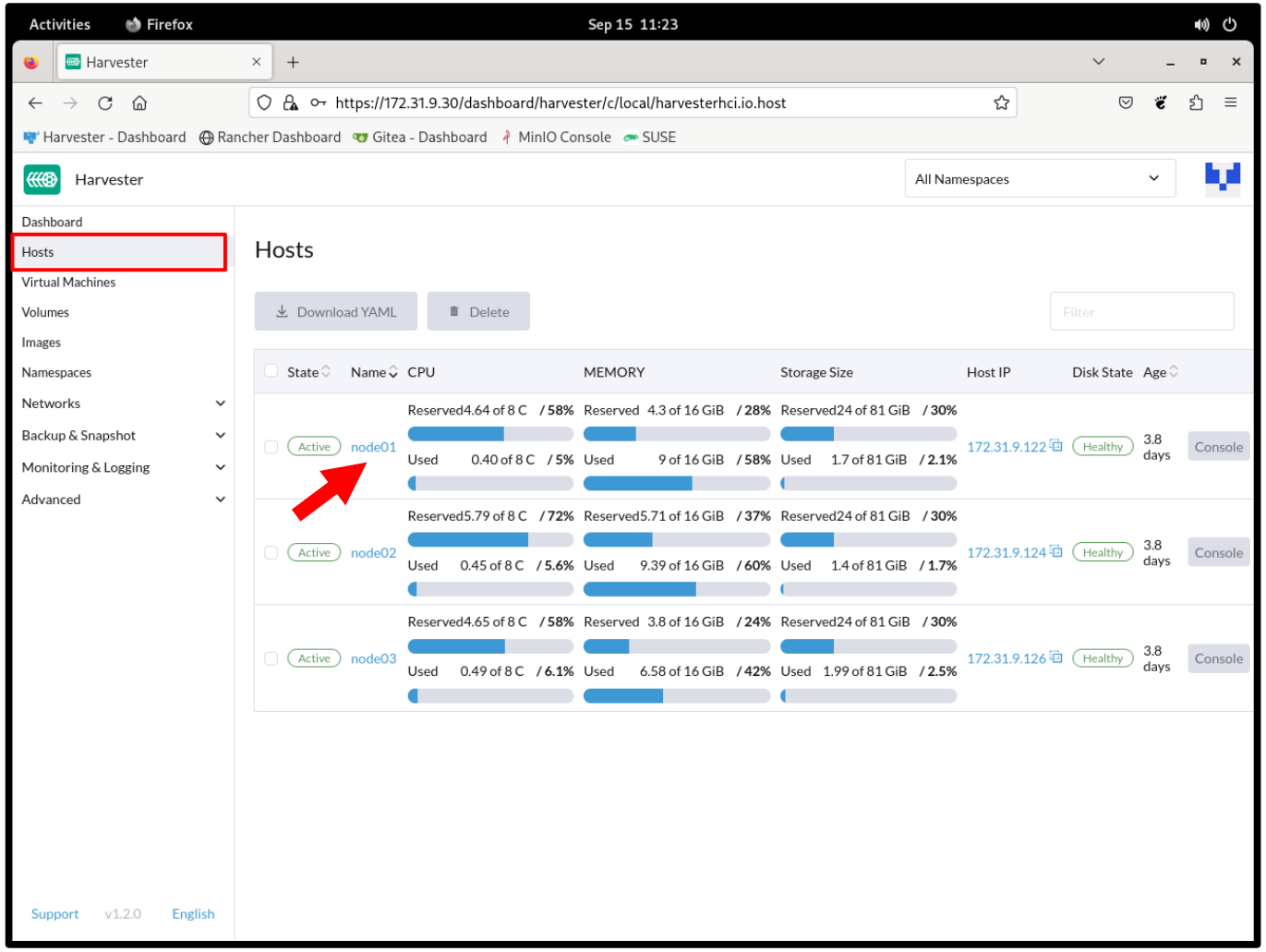
- Perform basic configuration of a Harvester cluster.

Topics

Harvester Cluster Node Configuration

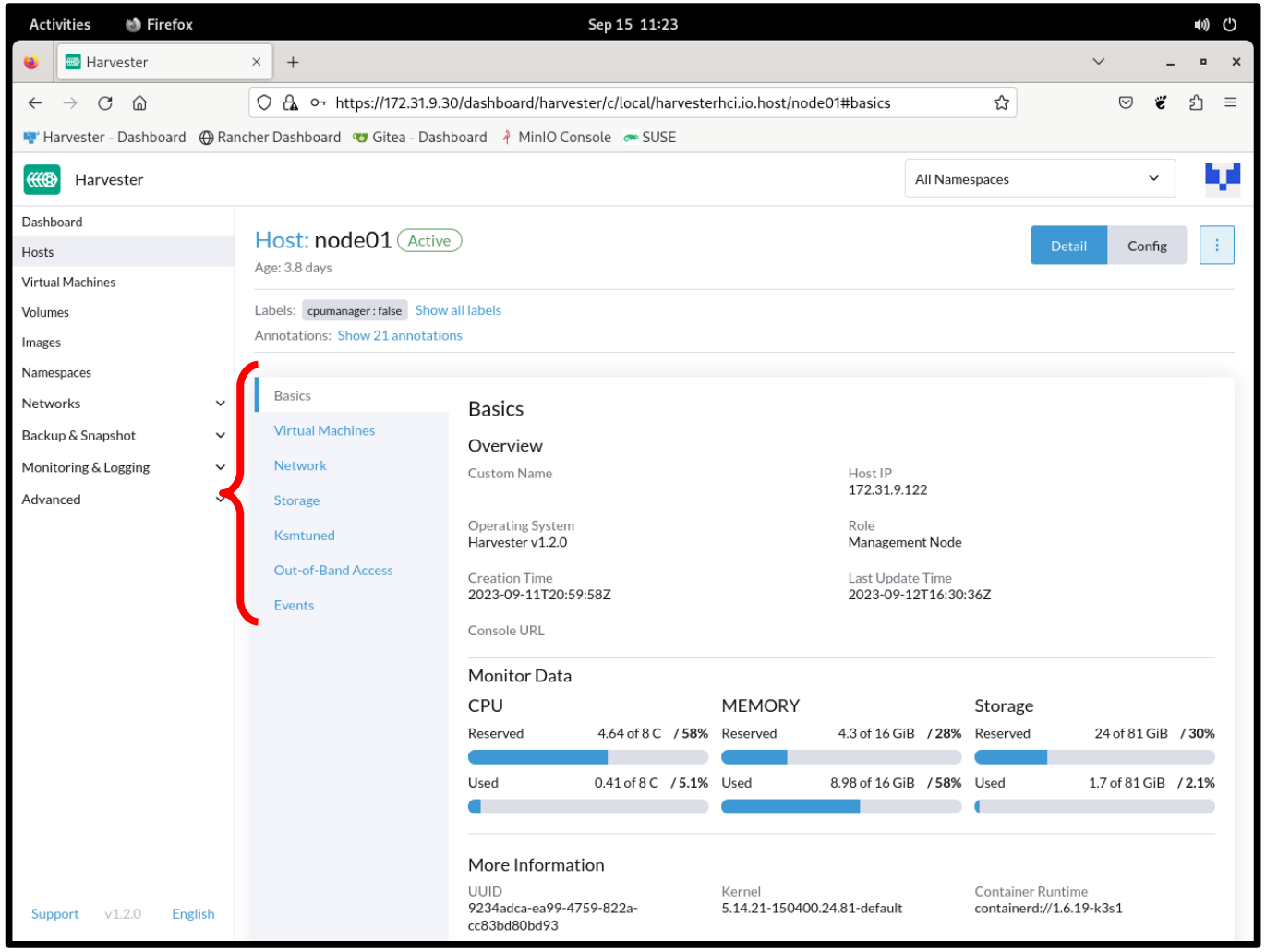
Harvester Hosts (Cluster Nodes)

- The **Hosts** screen displays information about the individual cluster nodes.
- Information such as the State of the hosts and how much CPU/Memory/Storage the hosts are consuming is displayed.
- You can display host specific information by clicking on the host's name.



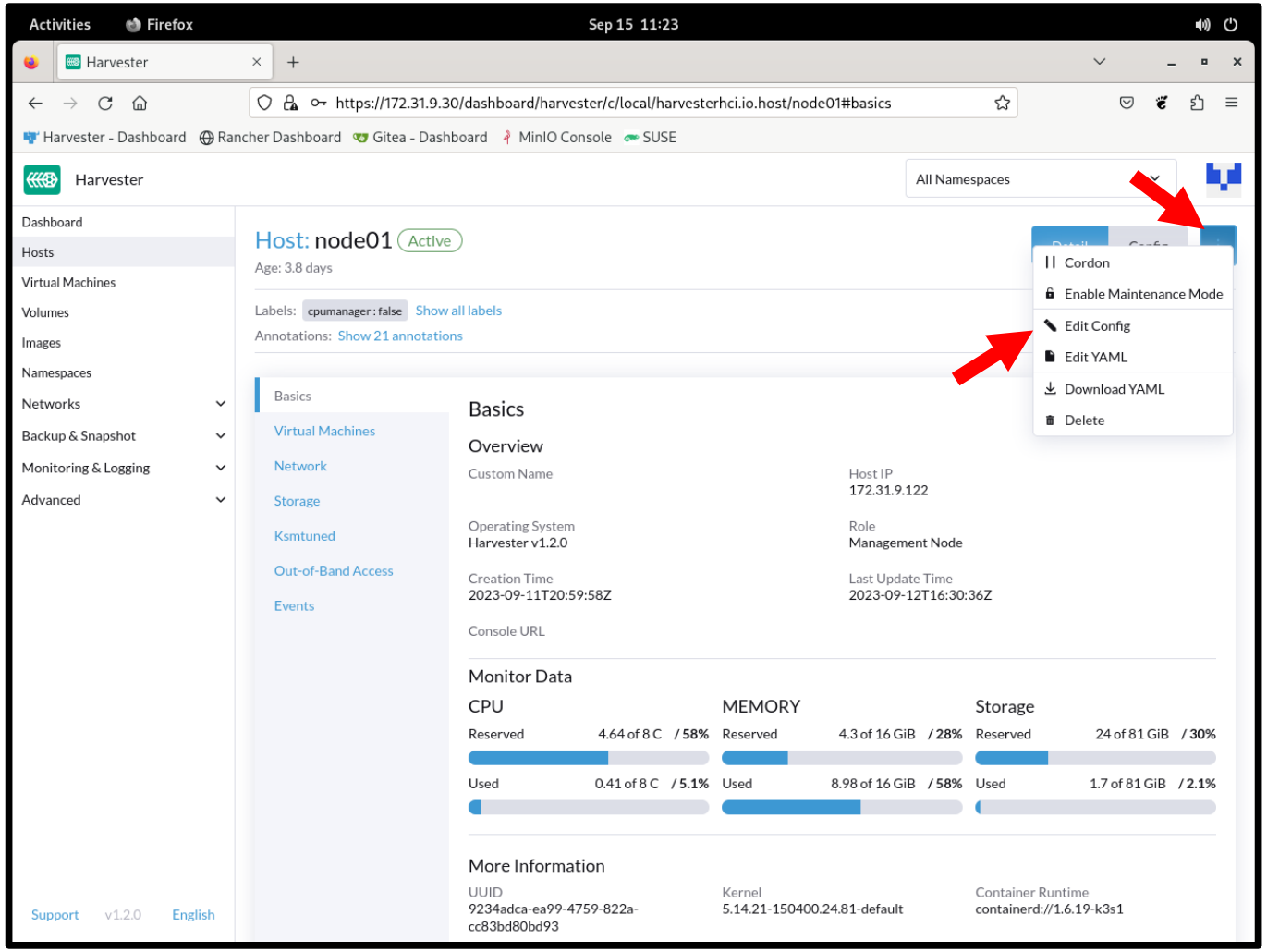
Cluster Host Info

- Host specific information is displayed on the individual host's **Host** screen.
- Tabs for basic info, info on the VMs running on the host and the host's disk and network are available



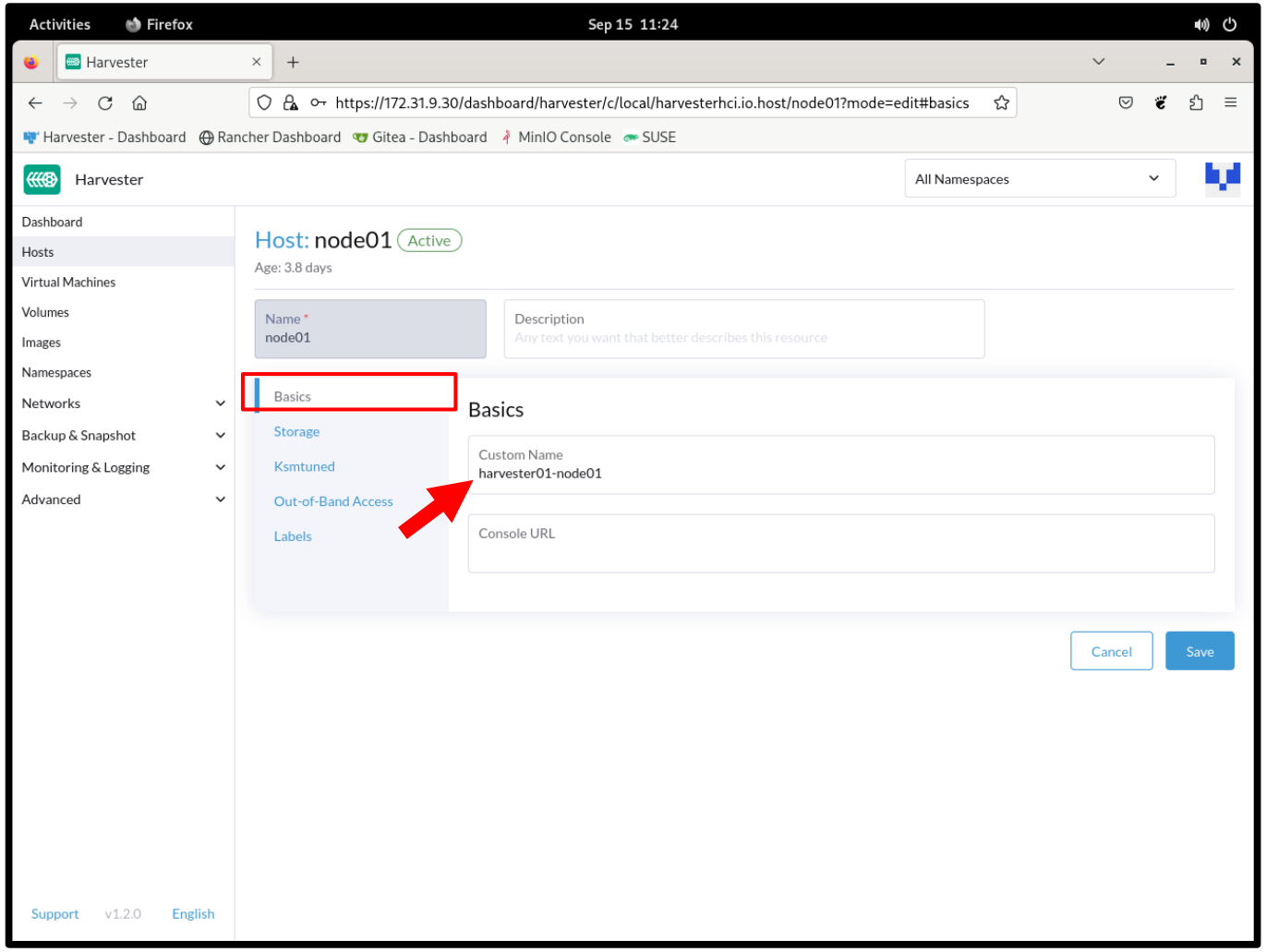
Cluster Host - Edit Config

- Additional operations that can be performed on the host are available from the **3 dot** menu.
- To edit a host's configuration select: **Edit Config**



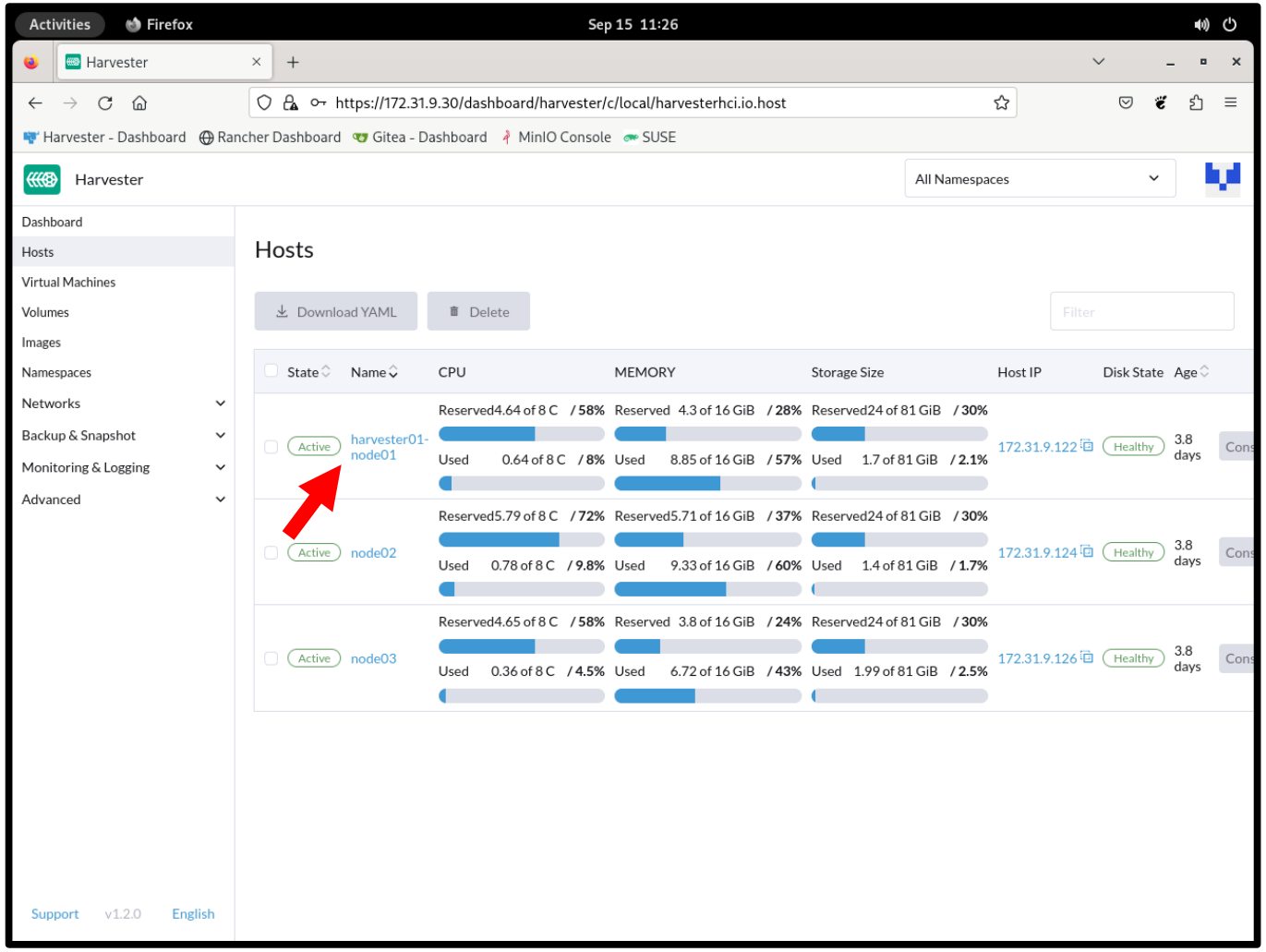
Cluster Host - Edit Config - Example: Basic

- On the **Basic** tab a **Custom Name** and or URL for the host can be modified.



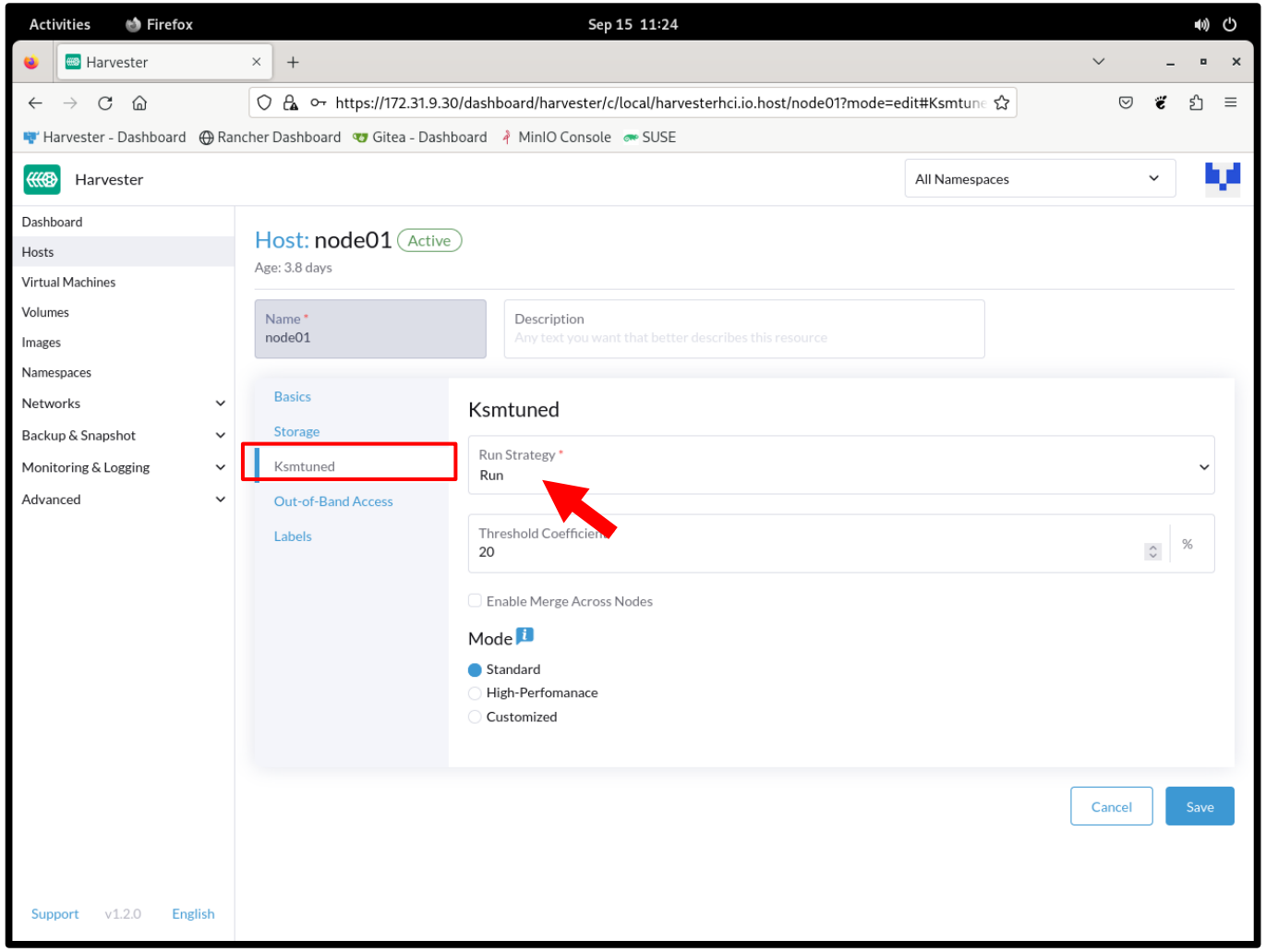
Host Name

- If a **Custom Name** is assigned to the host then that name is displayed on the main Hosts screen instead of the hostname.



Cluster Host - Edit Config - Example: Ksmtuned

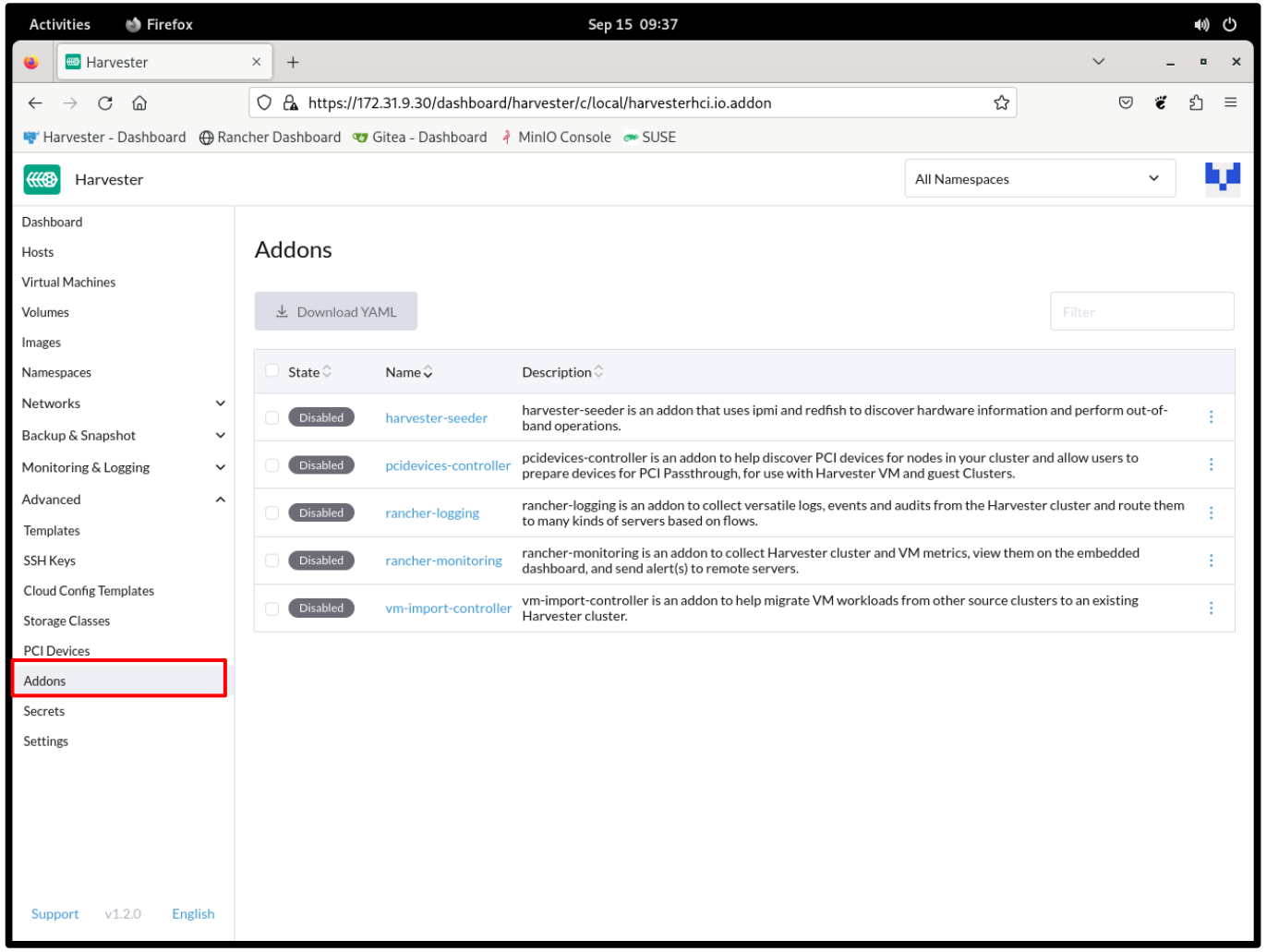
- **KSM** (Kernal Samepage Merging) can reduce memory use of VMs by remapping pointers to memory in VMs that are the same a to common shared memory pages.
- The **Ksmtuned** controller can be enabled and the **Run Strategy** can be set to:
  - **Run** - Start Ksmtuned
  - **Stop** - Stop Ksmtuned (VMs can still use shared memory pages)
  - **Prune** - Stop Ksmtuned and prune memory pages



Harvester Addons

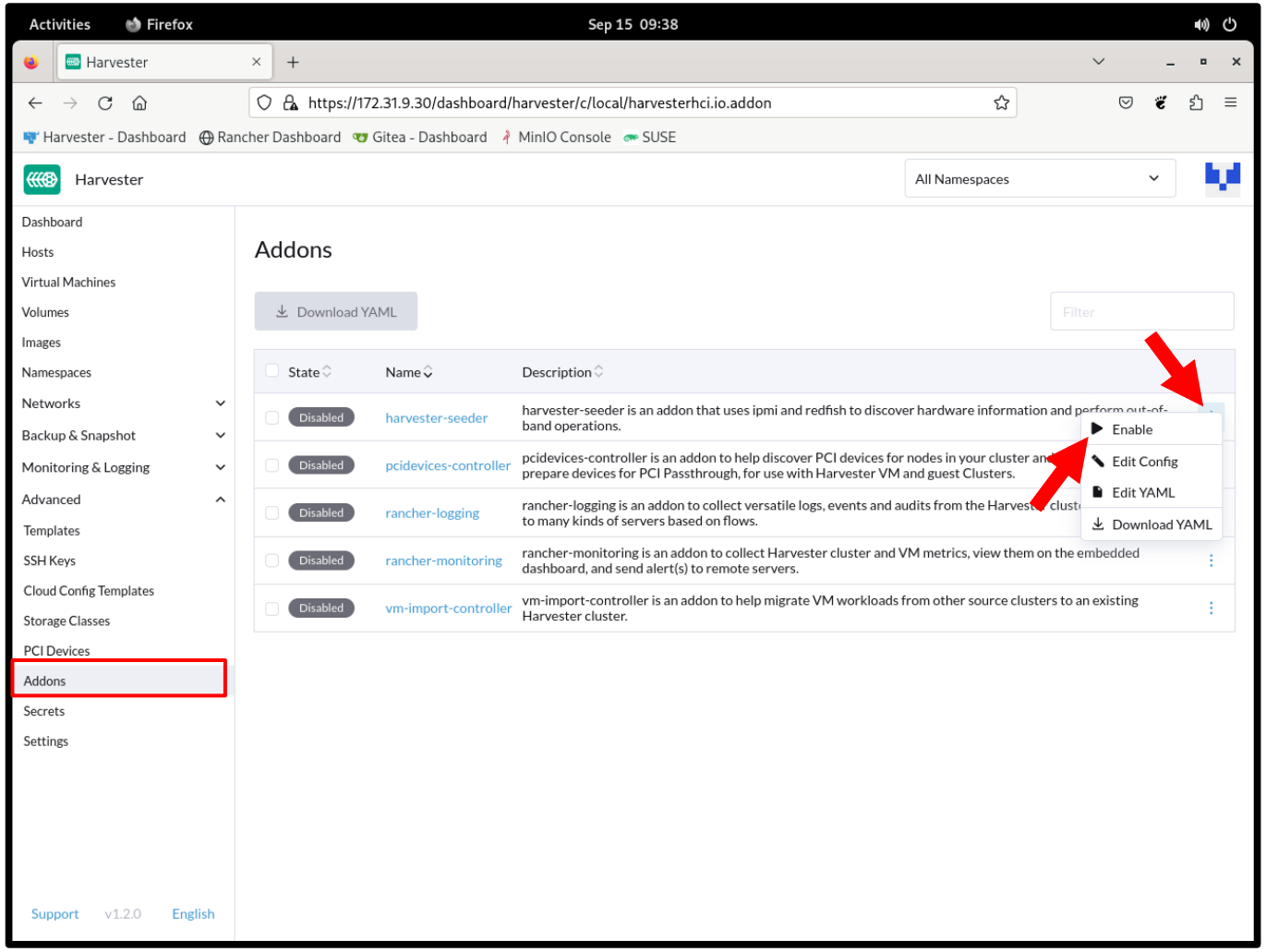
Harvester Addons

- The Harvester UI has been made more modular by creating the concepts of **Addons**.
- Addons and their state can be seen on the **Advanced → Addons** screen.
- Addons, when enabled/disabled, will install/remove additional functionality and display additional sections in the Harvester UI.



Enable/Disable Addons

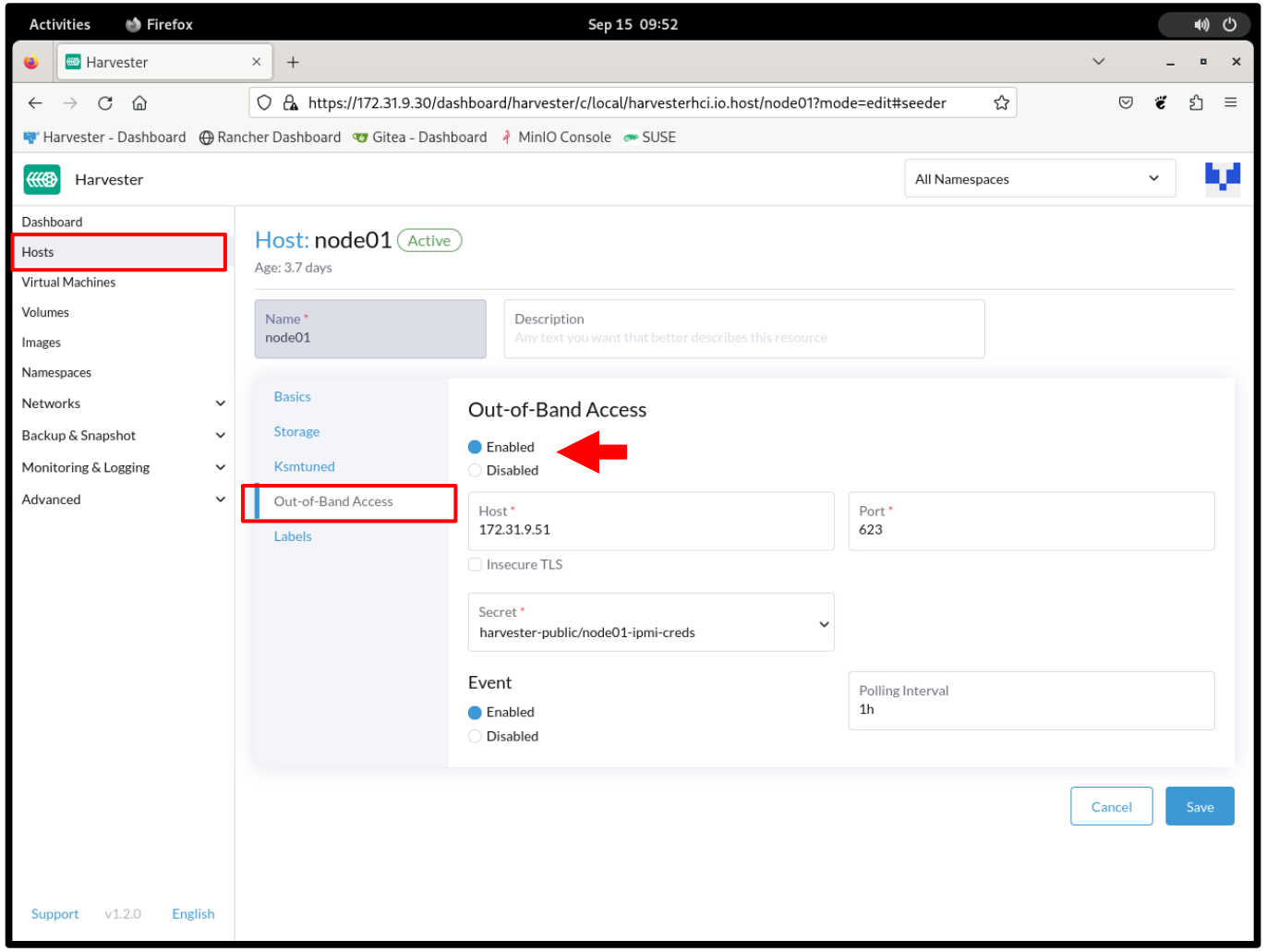
- To enable an addon, go to the **Advanced → Addons** screen, locate the **addon** you wish to enable, click on the **3 dot** menu next to it and select **Enable**.



Addon - Example: harvester-seeder

- For example, when the **harvester-seeder** addon is enabled the ability to configure **Out-of-Band Access** can enabled in a host's configuration where otherwise it couldn't be.

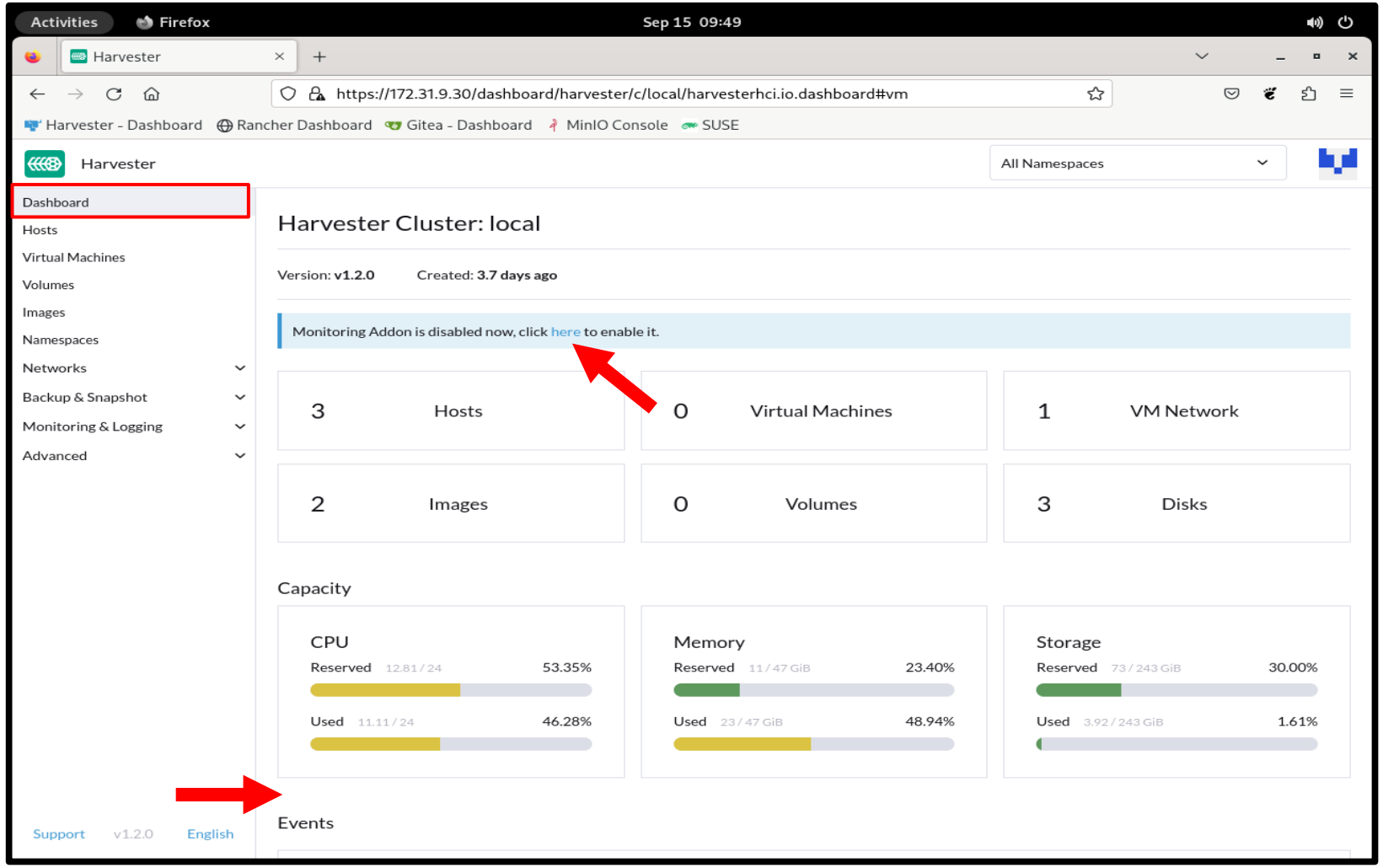
  (Other addons will be covered in subsequent sections of the course)



Cluster Monitoring and Metrics

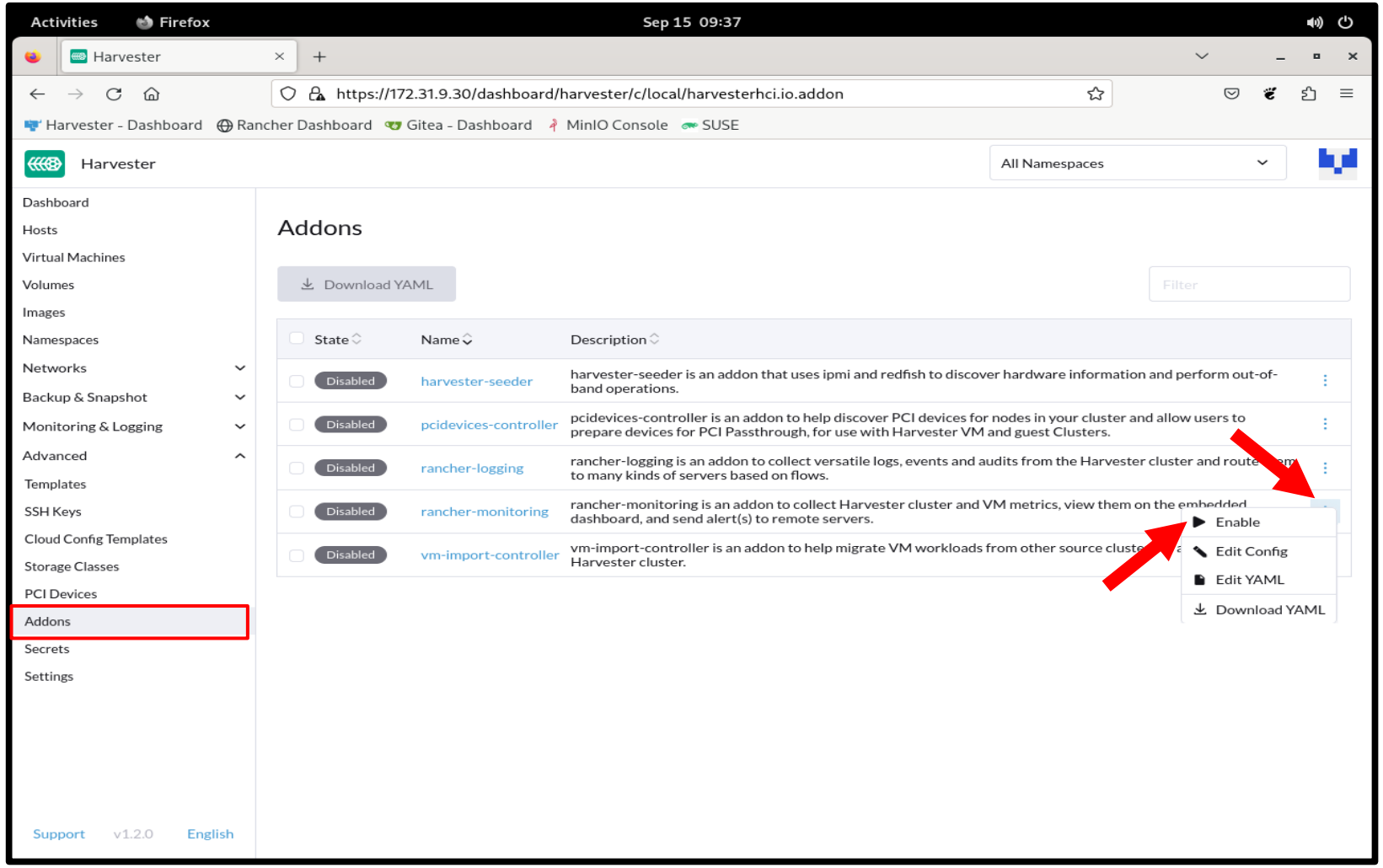
Monitoring Addon

- Monitoring is no longer enabled by default starting with Harvester v1.2.0.
- An addon must be enabled to install and display the metrics on the Dashboard.



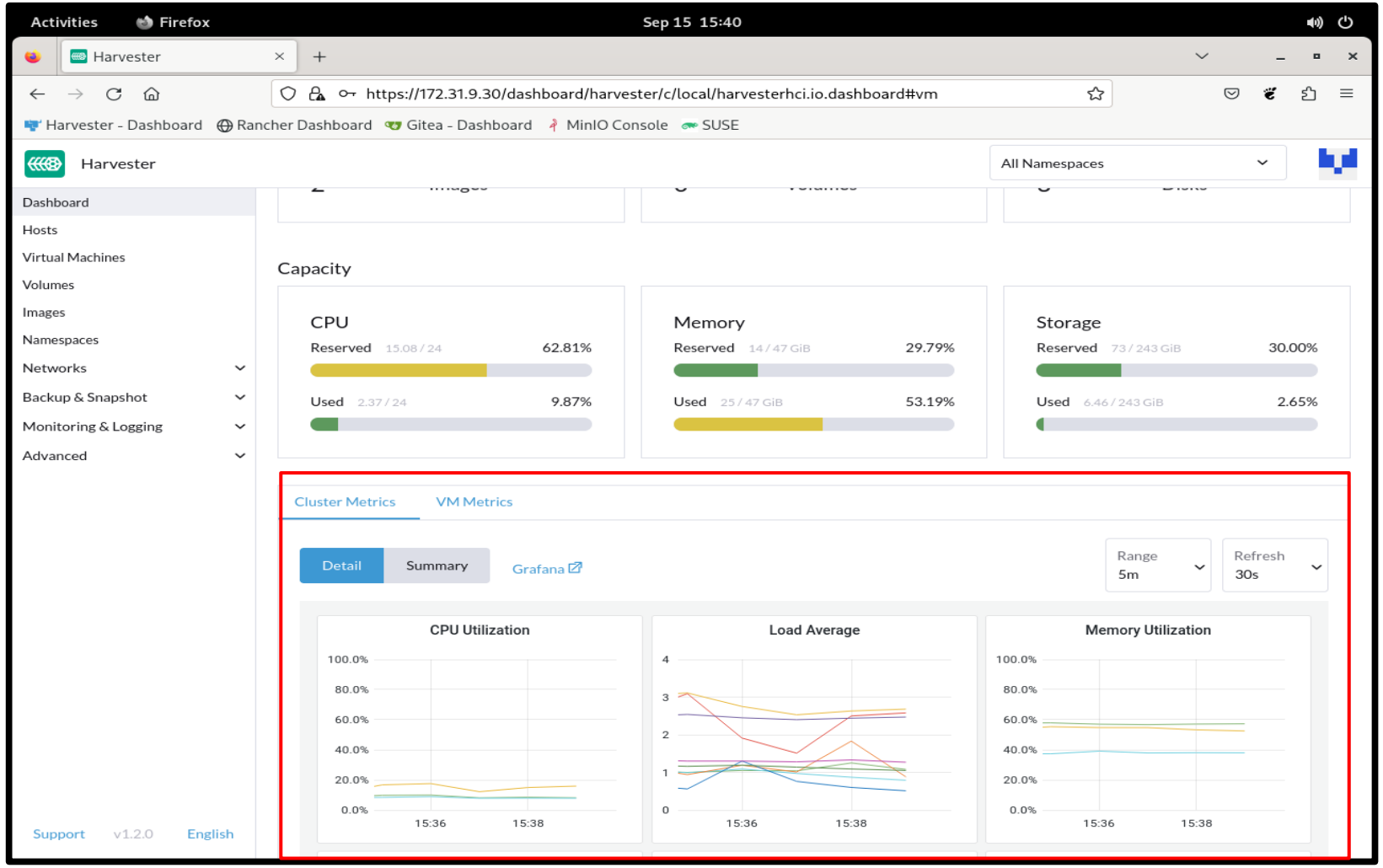
Enable the rancher-monitoring Addon

- To enable monitoring/metrics go to the **Advanced → Addons** screen.
- Click on the **3 dot** menu next to the **rancher-monitoring** addon and select: **Enable**



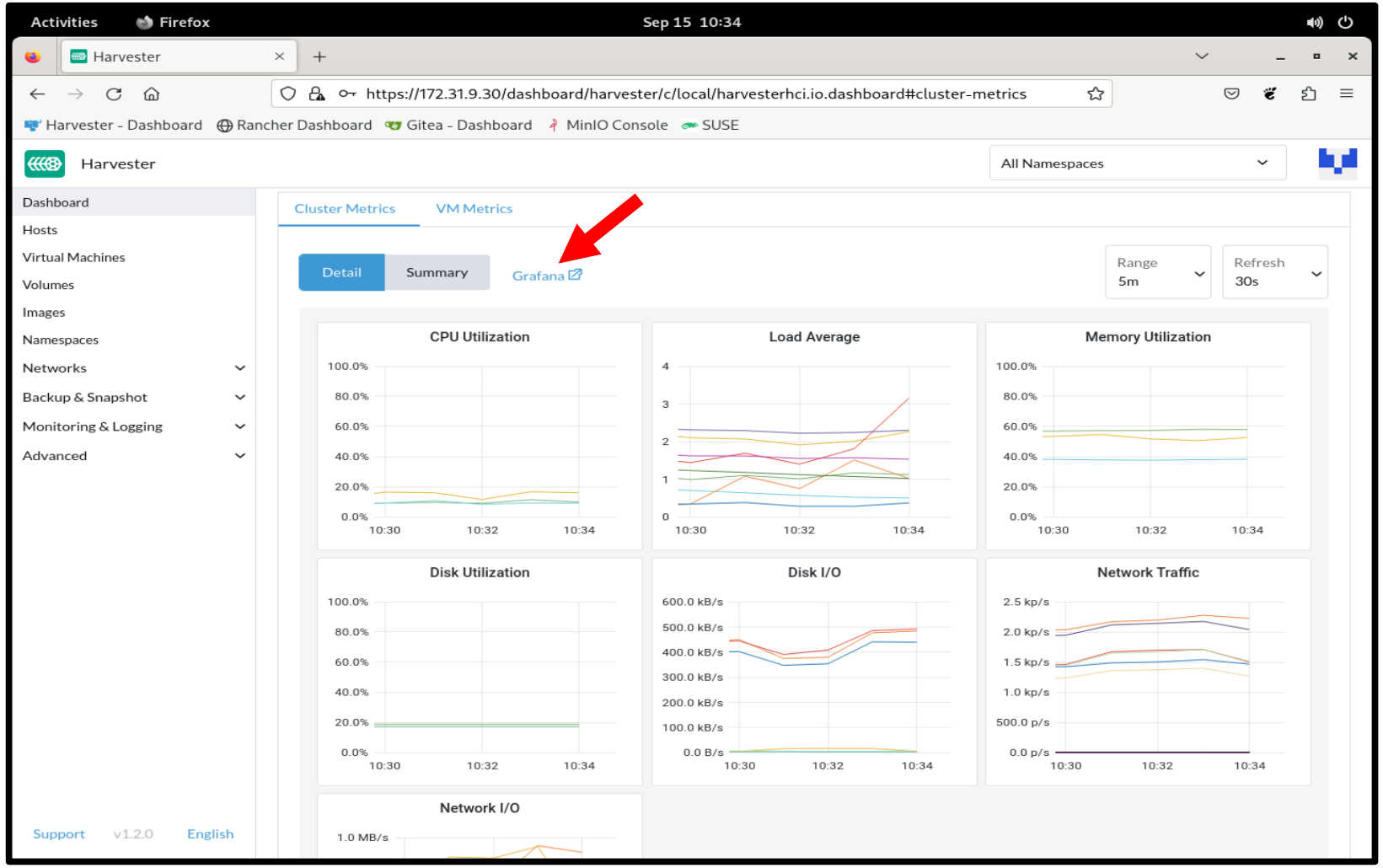
Cluster Metrics Displayed

- Once the **rancher-monitoring** addon is enabled, installing the required components in the process, the cluster and VM metrics are displayed on the Dashbo



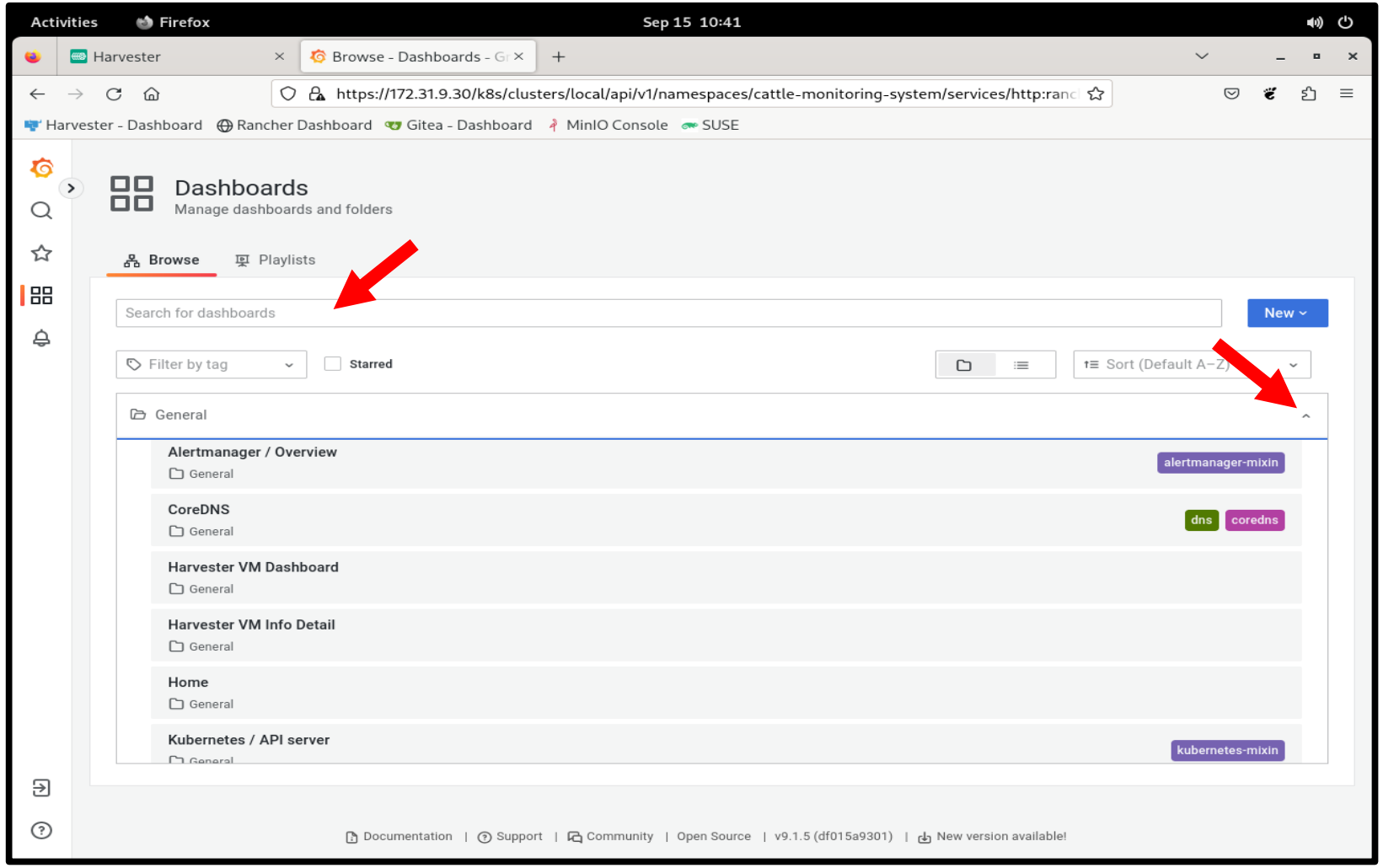
Cluster Metrics

- Metrics can be used to get an overall view of the health and performance of the cluster and its hosted VMs
- The metrics displayed on the **Dashboard** screen are provided by **Grafana** which in turn gets its data from the metrics data collected by Prometheus.
- The **Grafana** service can be accessed directly if desired.



Grafana - Browse Dashboards

- Additional dashboards can be selected or searched for on the Grafana **Dashboards** screen.
- The additional dashboards provide more detailed metrics for all aspect of the Harvester cluster that aren't exposed on the main Harvester Dashboard screen



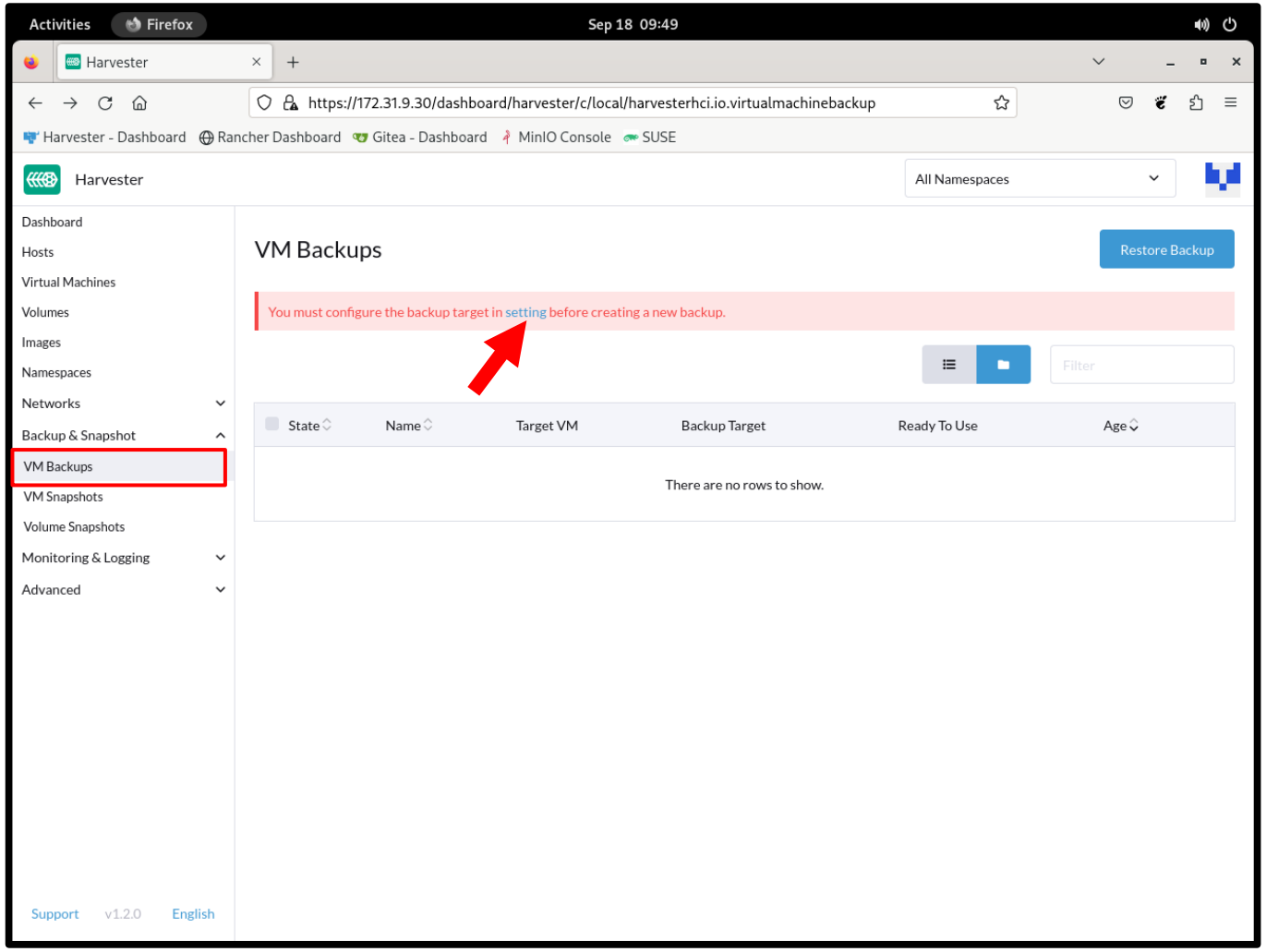
VM Backup Target Configuration

Backup Target

- Harvester provides VM backup and restore functionality.
- A backup target must be configured to store backups of the VMs.
- The backup target functionality is inherited from the backup feature of the embedded Longhorn cluster.

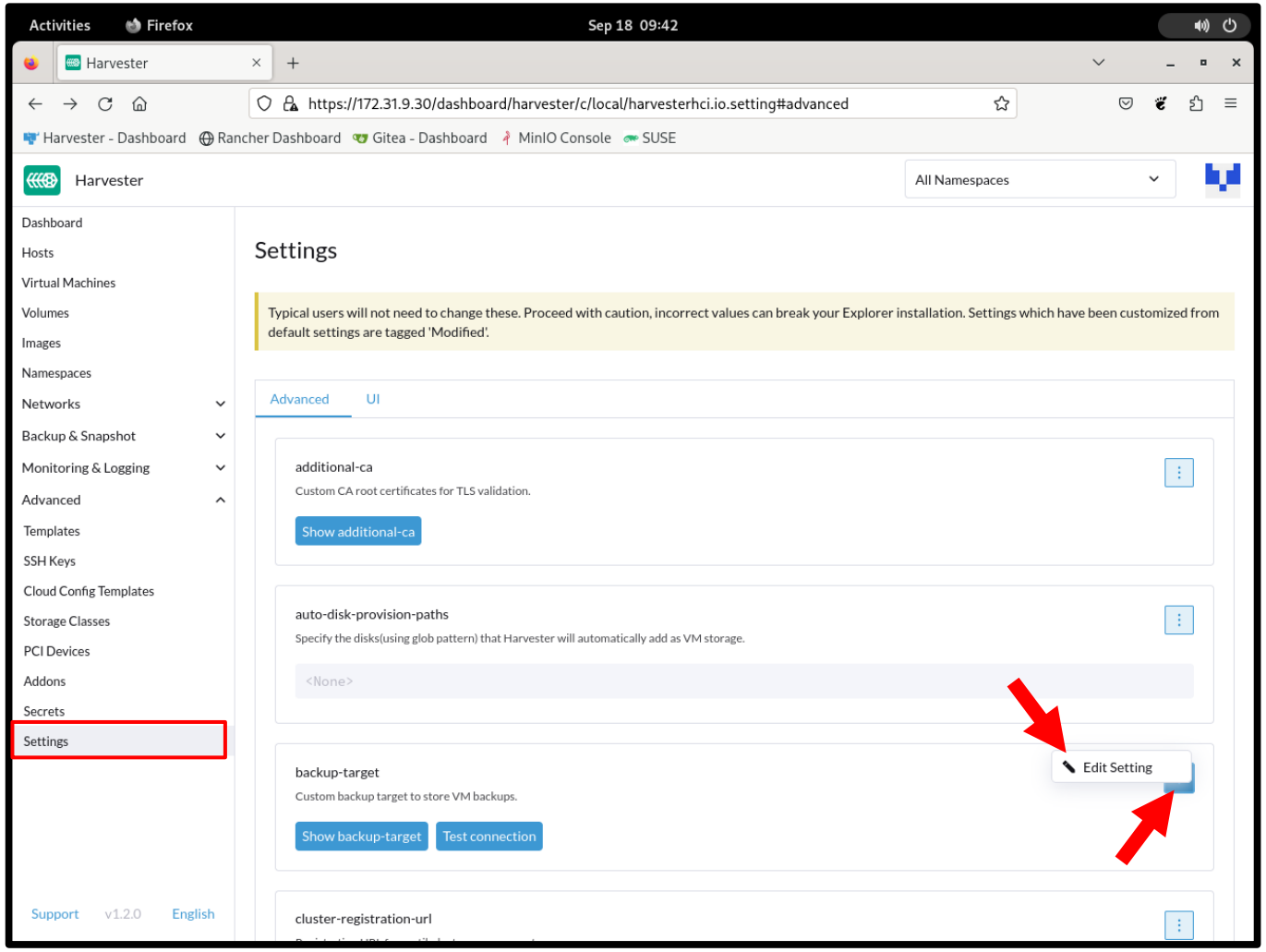
Backups

- VM backups are displayed on the **Backups & Snapshots → VM Backups** screen.
- If a backup target has not been configured, a warning message will be displayed.
- Clicking the **setting** link takes you directly to where the backup target is configured (**Advanced→Settings→backup-target**)



Backup Target Configuration

- The backup target is configured on the **Advanced → Settings** screen.
- Click the **3 dot** menu next to **backup-target**
- Select: **Edit Setting**



Restore to New Cluster

- An existing backup target that contains VM backups can be connected to a new cluster and the VMs can be restored.

- This functionality can be used:
  - As a disaster recovery method for the VMs.
  - To move VMs to another Harvester cluster.

Section Summary

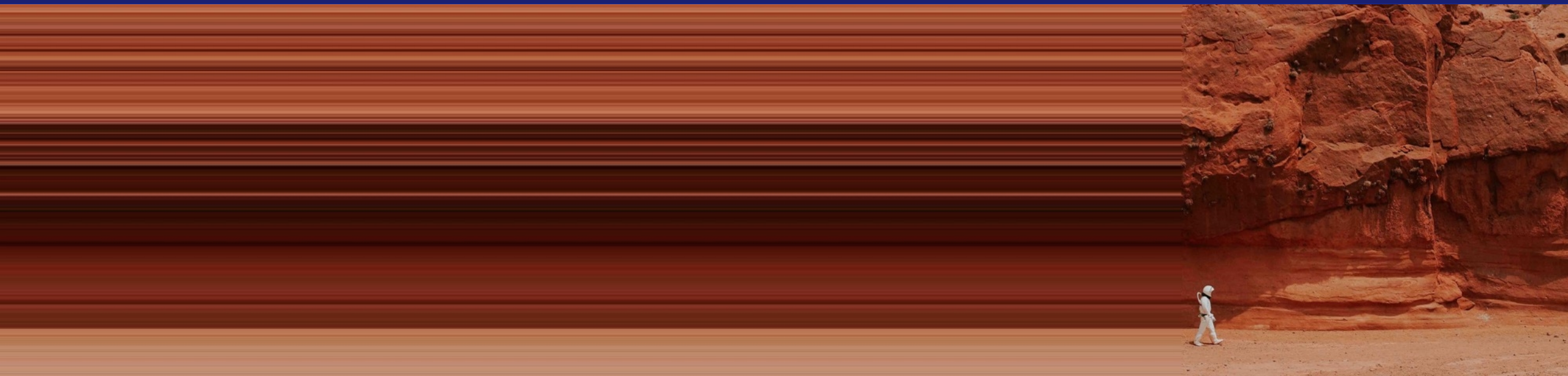
Section Summary

- Perform basic configuration of a Harvester cluster
  - Cluster nodes can be individually configured from the Hosts screen for things such as their custom name, Ksmtuned and other host specific configuration.
  - Harvester addons can be enabled to extend the functionality of the cluster.
  - Cluster metrics can be viewed on the main Dashboard screen once the addon has been enabled and more detailed metrics can be viewed by accessing the embedded Grafana service directly.
  - An NFS server or an S3 bucket can be configured to store VM backups.

SUSE

Section: 5

Networking in Harvester



Learning Objectives

At the end of this section you will be able to:

- Describe Harvester networking concepts.
- Create/modify/delete cluster networks and network configs.
- Create/modify/delete VM networks.

Topics

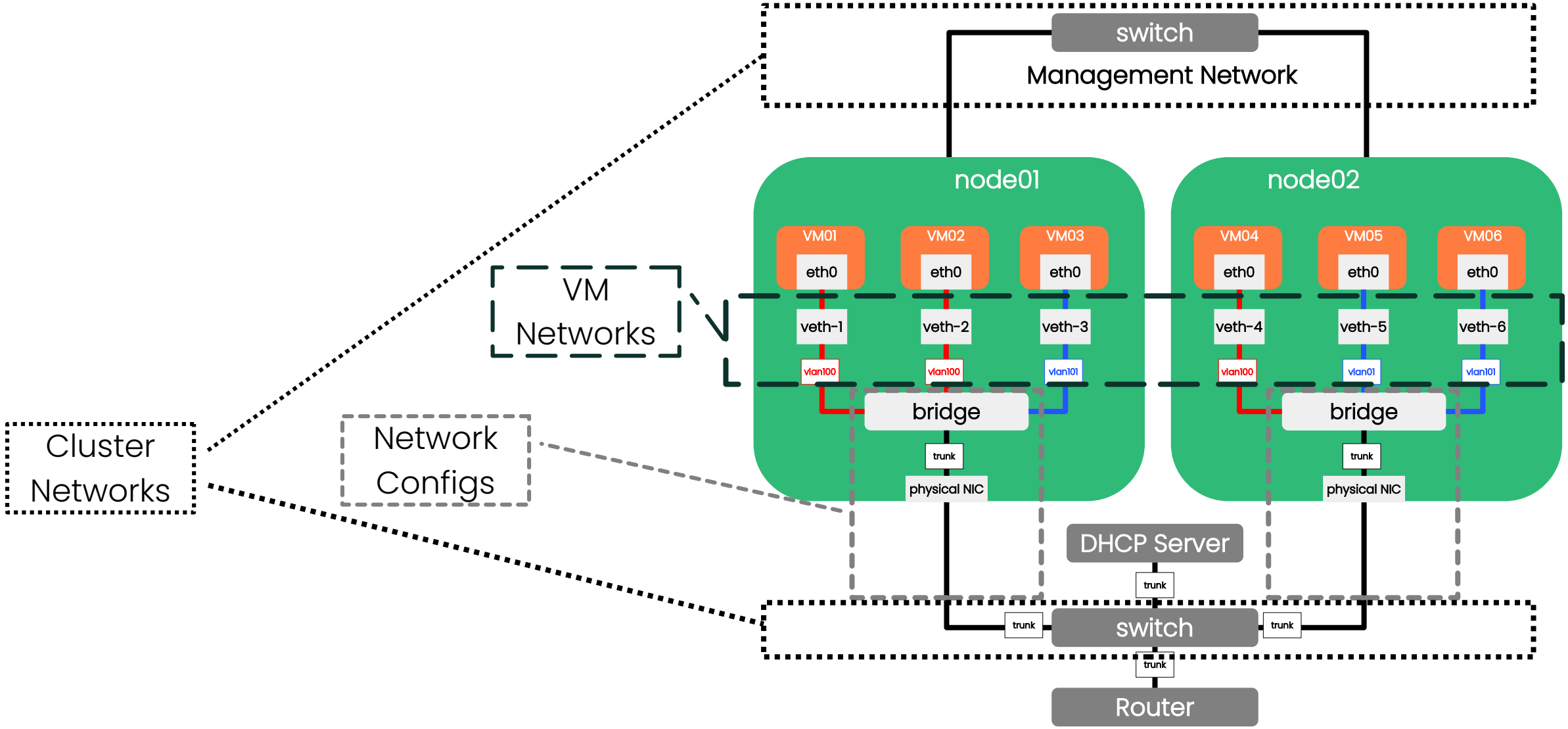
Overview of Networking in a Harvester Cluster

Harvester Networking Concepts

Harvester has three main networking concepts that must be understood to configure Harvester networking.

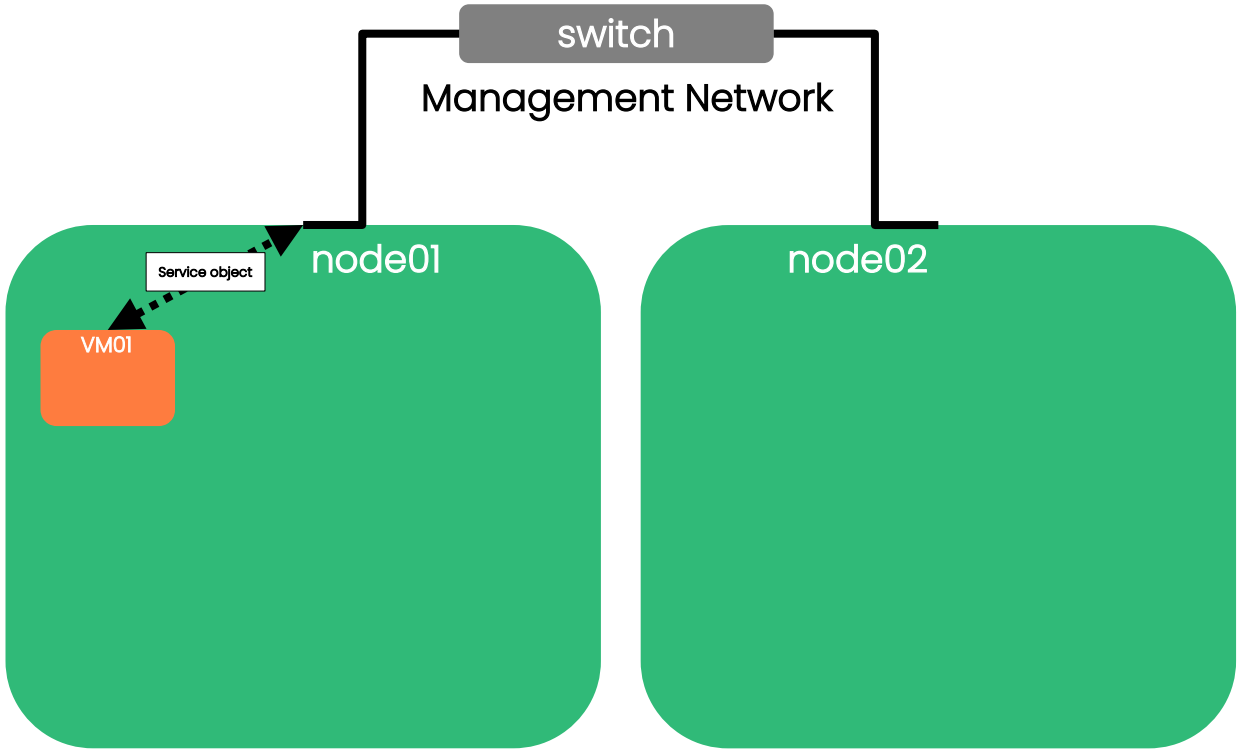
Concept	Description
Cluster Network	• Traffic-isolated forwarding path (i.e. network) used in the harvester cluster and data center to carry network traffic.
Network Config	• Defines how cluster nodes connect to cluster networks (network interface, bond type, etc).
VM Network	• Network virtual machines are connected to and use for communication. • Associated with a Cluster Network.

Harvester Network Examples Diagram



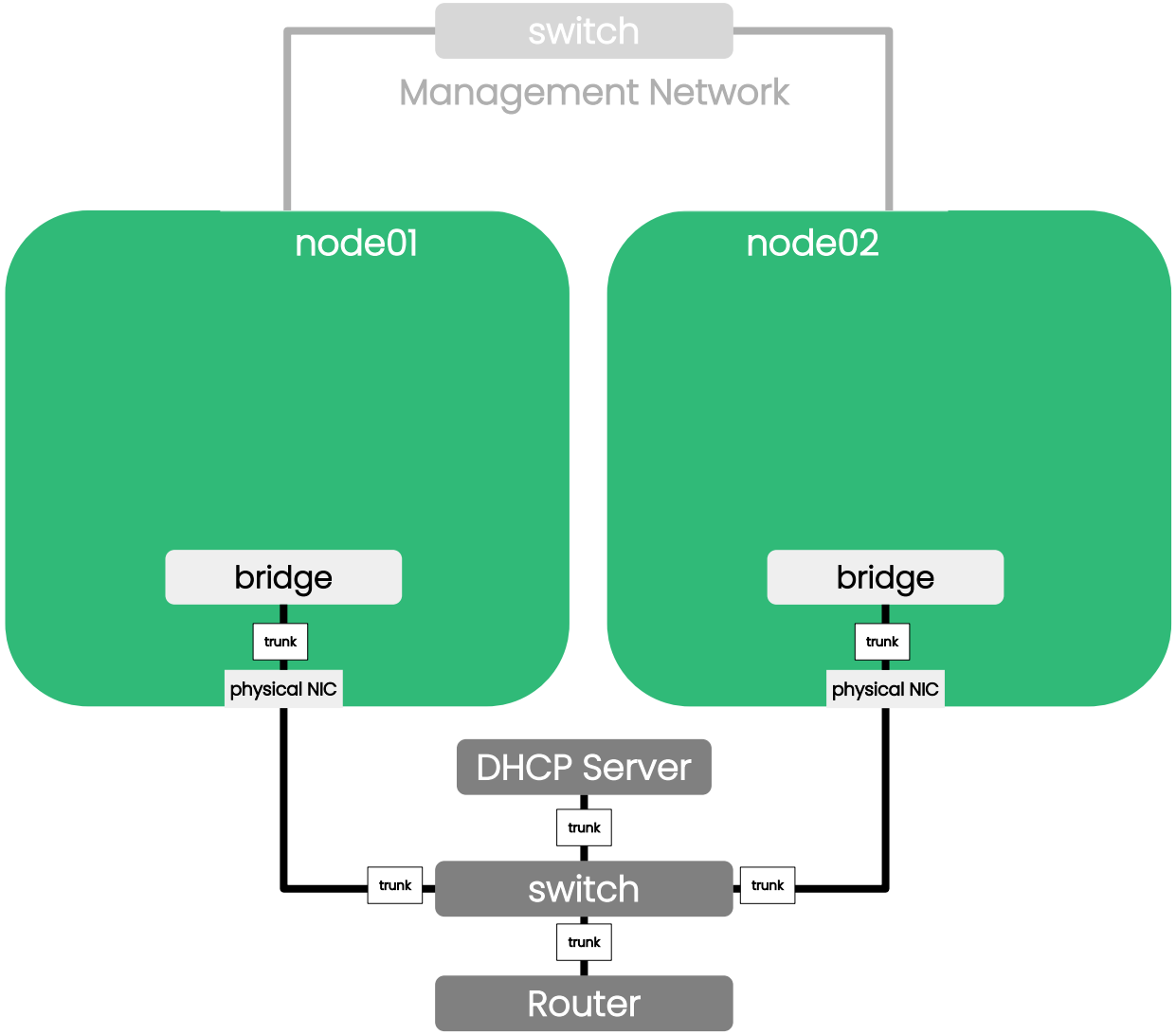
Cluster Network - Management Network

- The Management Network is a Cluster Network created by the Canal CNI in the underlying Kubernetes cluster.
- Cluster nodes are connected to the management network and typically only use the management network for the purpose of cluster communication.
- A Kubernetes service object can be created that enables stable IPs for VMs to communicate on the management network.
  - However, VM Networks are the recommended way to provide network connectivity to the VMs.



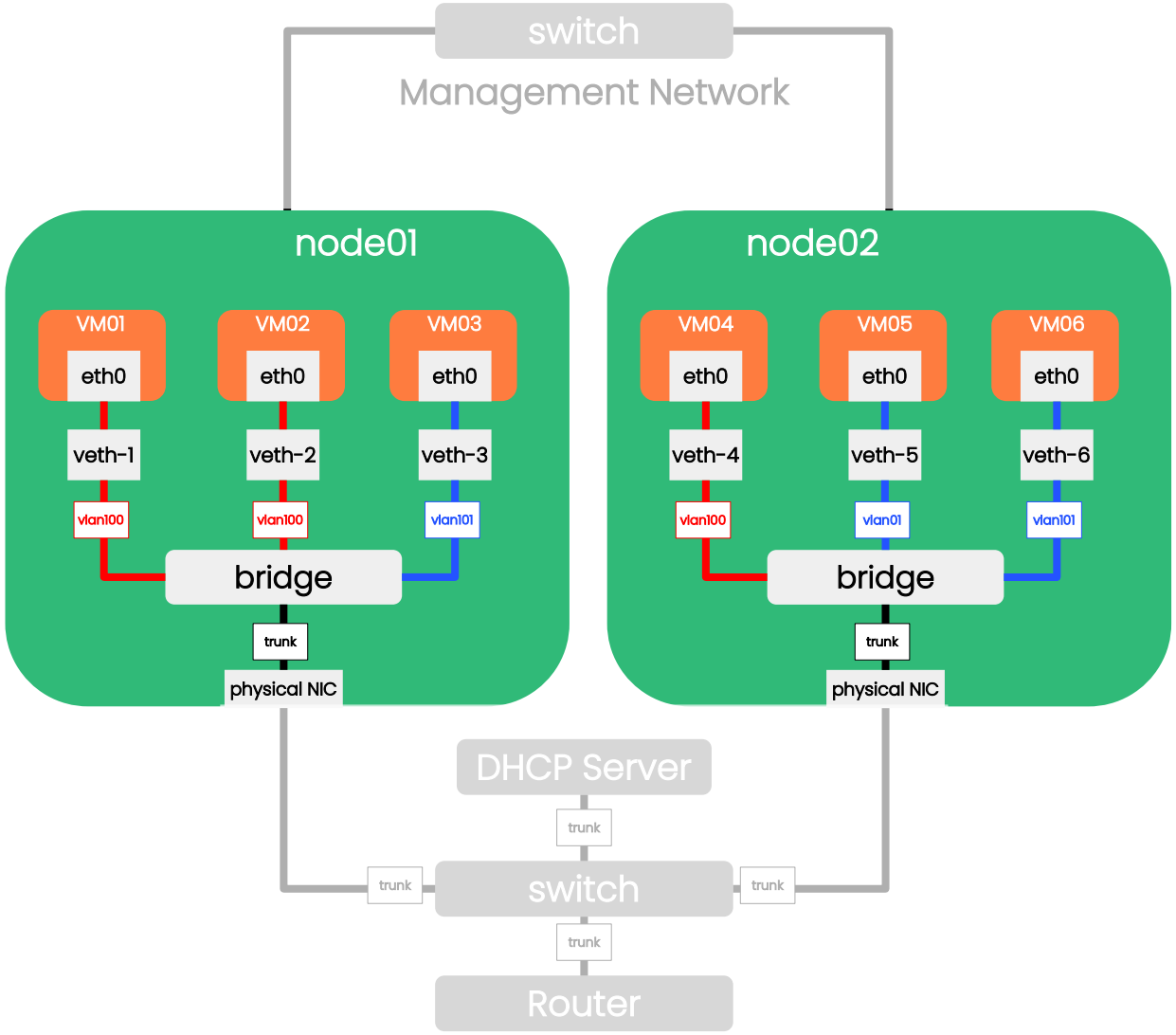
Cluster Network – VLAN Trunk Network

- Commonly a VLAN trunk network is configured and connected to additional network interfaces in the cluster nodes.
- The VLAN trunk Cluster Network will be used to host the VLAN networks and isolate that traffic away from cluster communication.



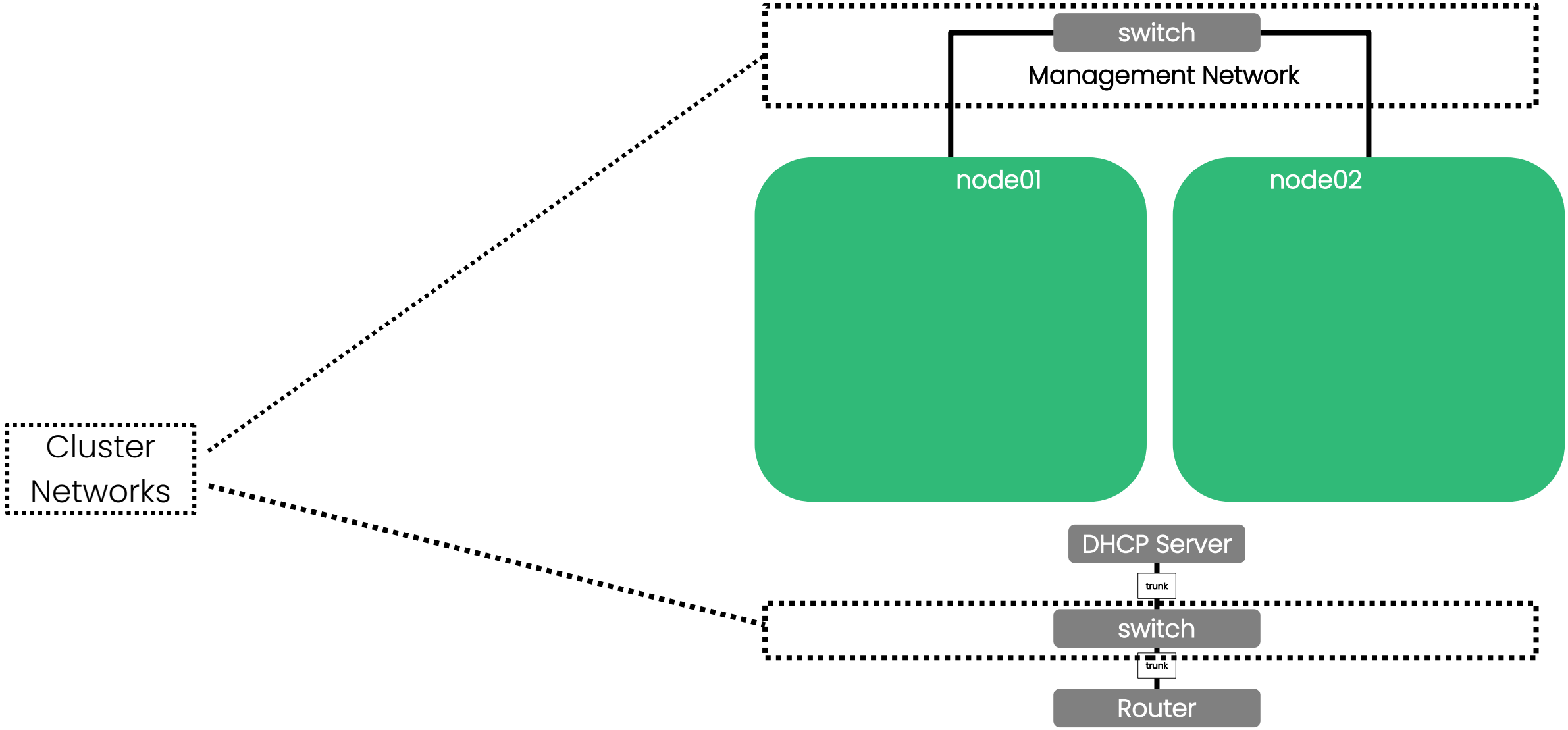
VM Networks

- VM Networks are implemented as VLANs or as untagged networks.
- VM Networks can be added to any Cluster Network.



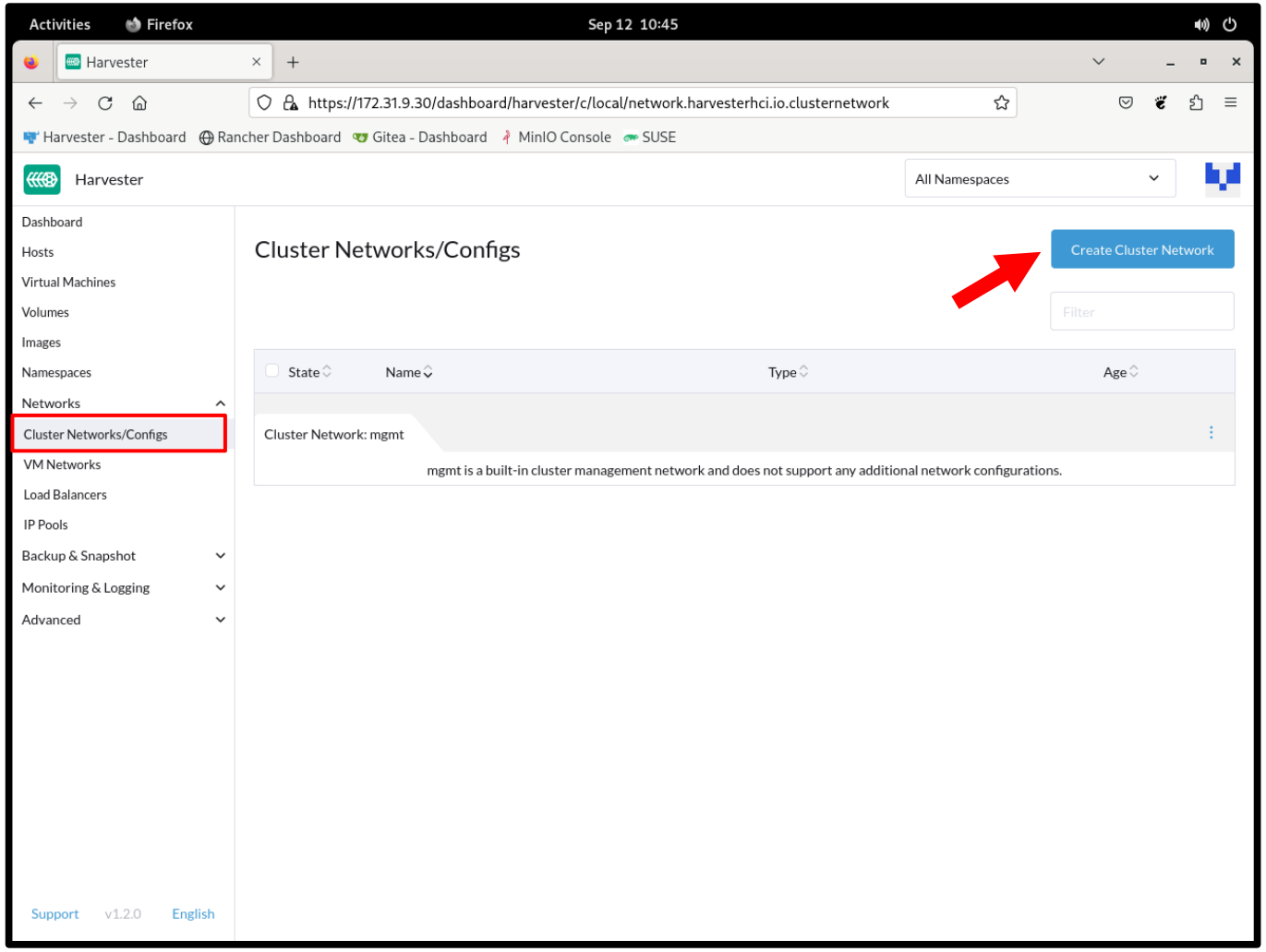
Cluster Network Definitions

Cluster Networks



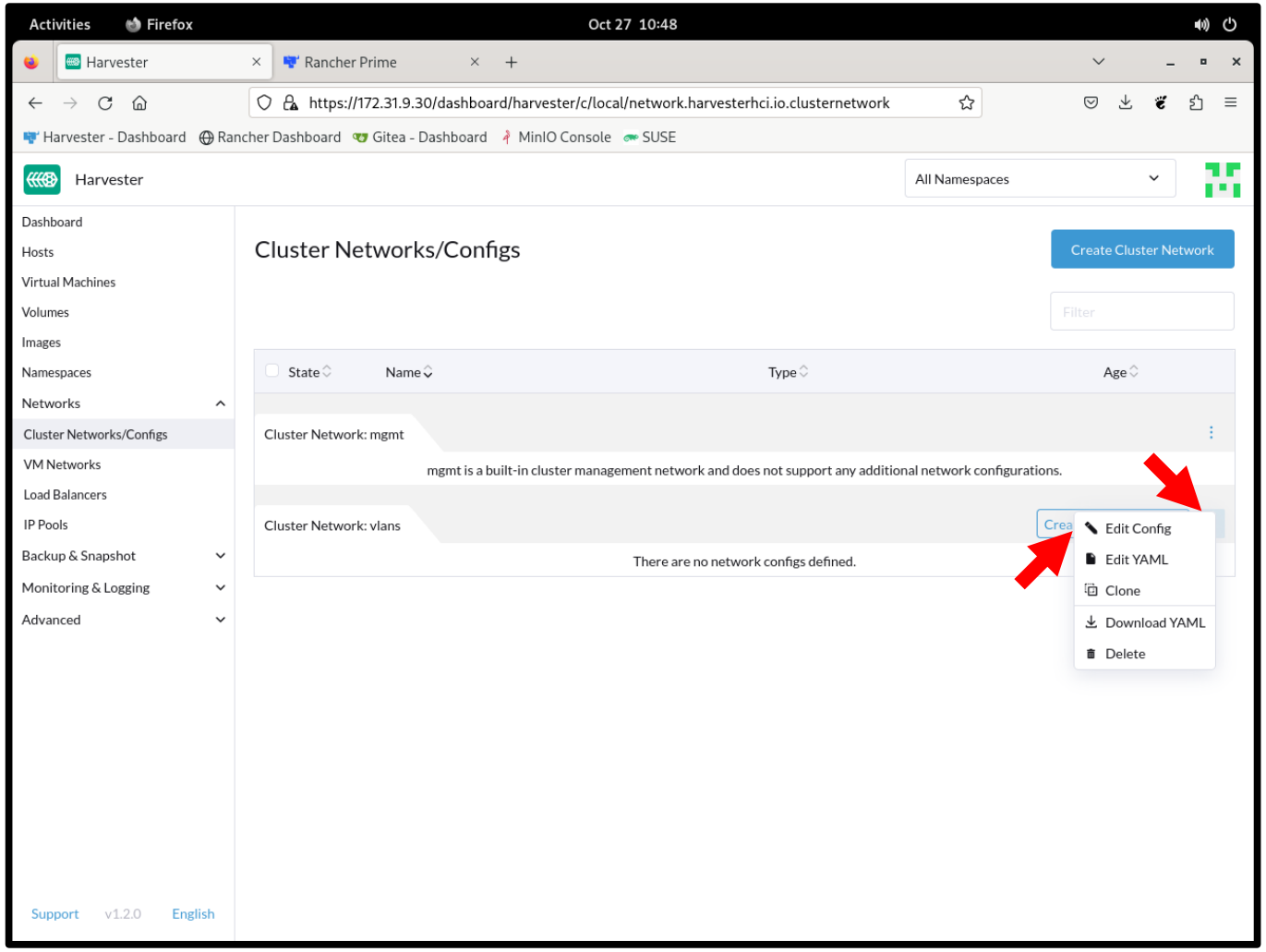
Create a Cluster Network

- Cluster Networks are displayed and managed on the **Networks → Cluster Networks/Co**



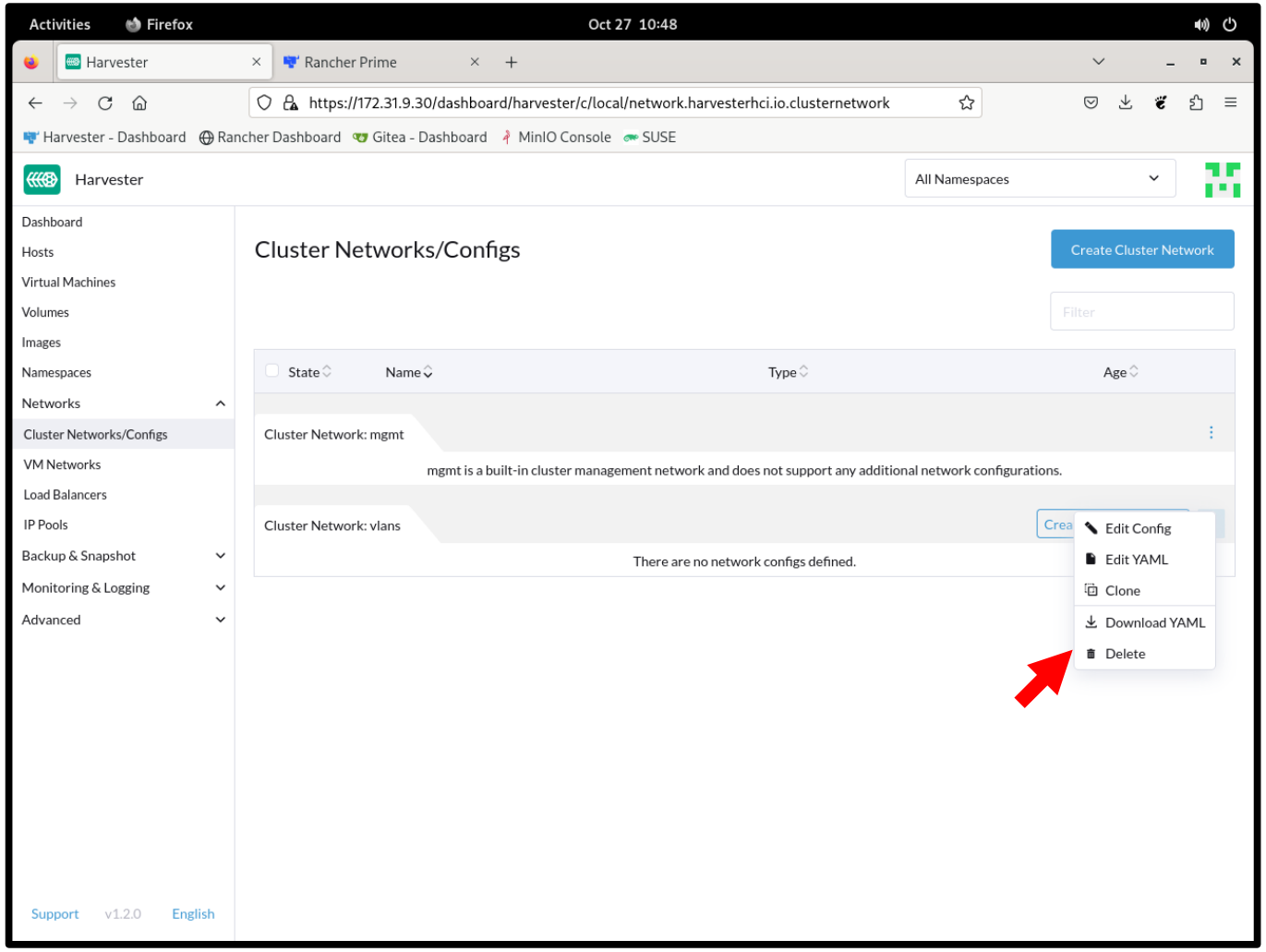
Modify Cluster Networks

- Cluster Networks, other than the management (mgmt) Cluster Network, can be modified after creation.
- Click on the **3 dot** menu next to the Cluster Network and select: **Edit Config**



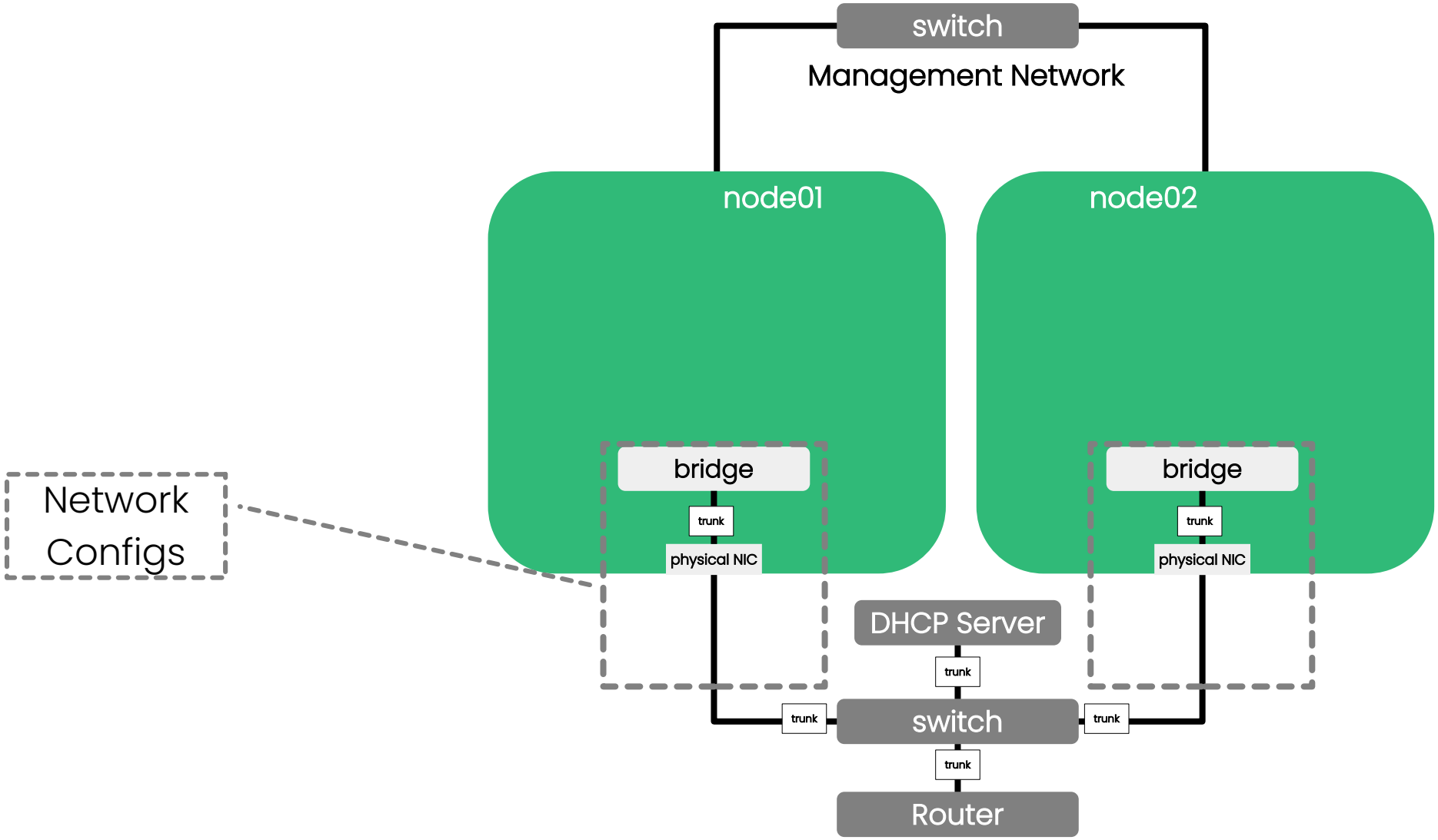
Delete Cluster Networks

- Cluster Networks, other than the management (mgmt) Cluster Network, can also be deleted.
- Cluster Networks cannot be deleted if they have Network Configs created on them, the Network Configs must be deleted first.



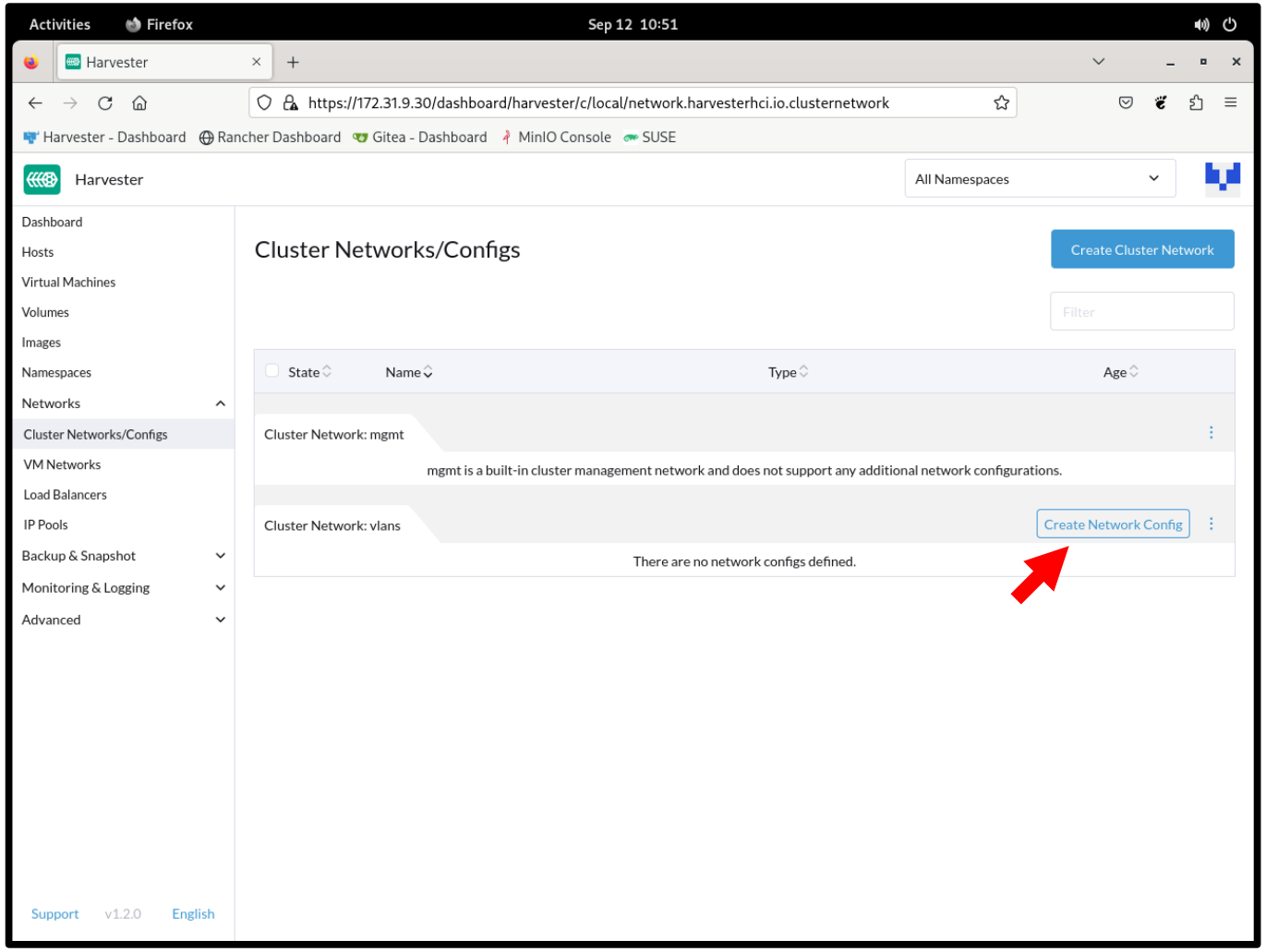
Network Configs

Network Configs



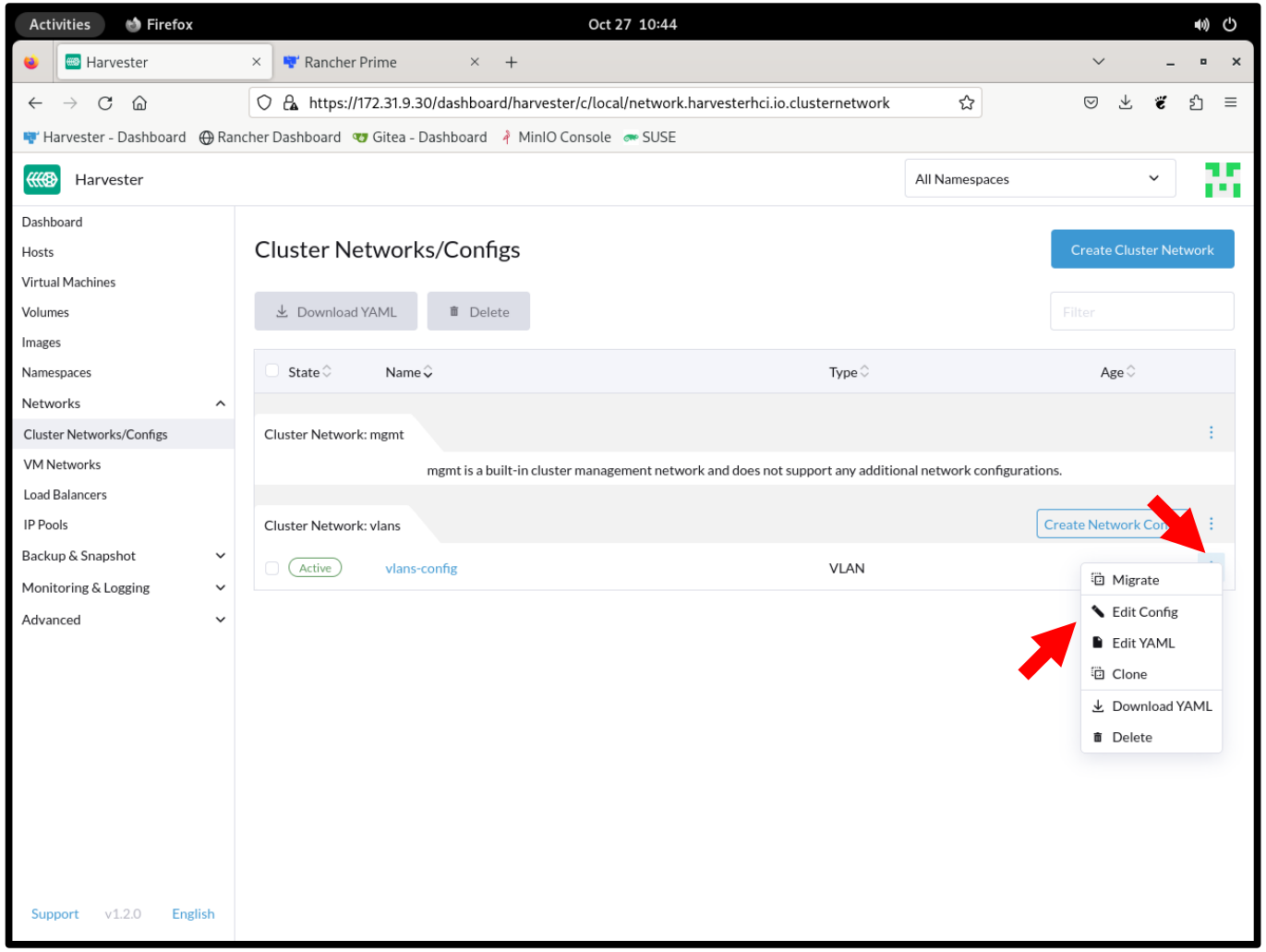
Create a Network Config

- On the **Cluster Networks/Configs** screen locate the Cluster Network on which you want to create the Network Config.
- Next to that Cluster Network click: **Create Network Config**



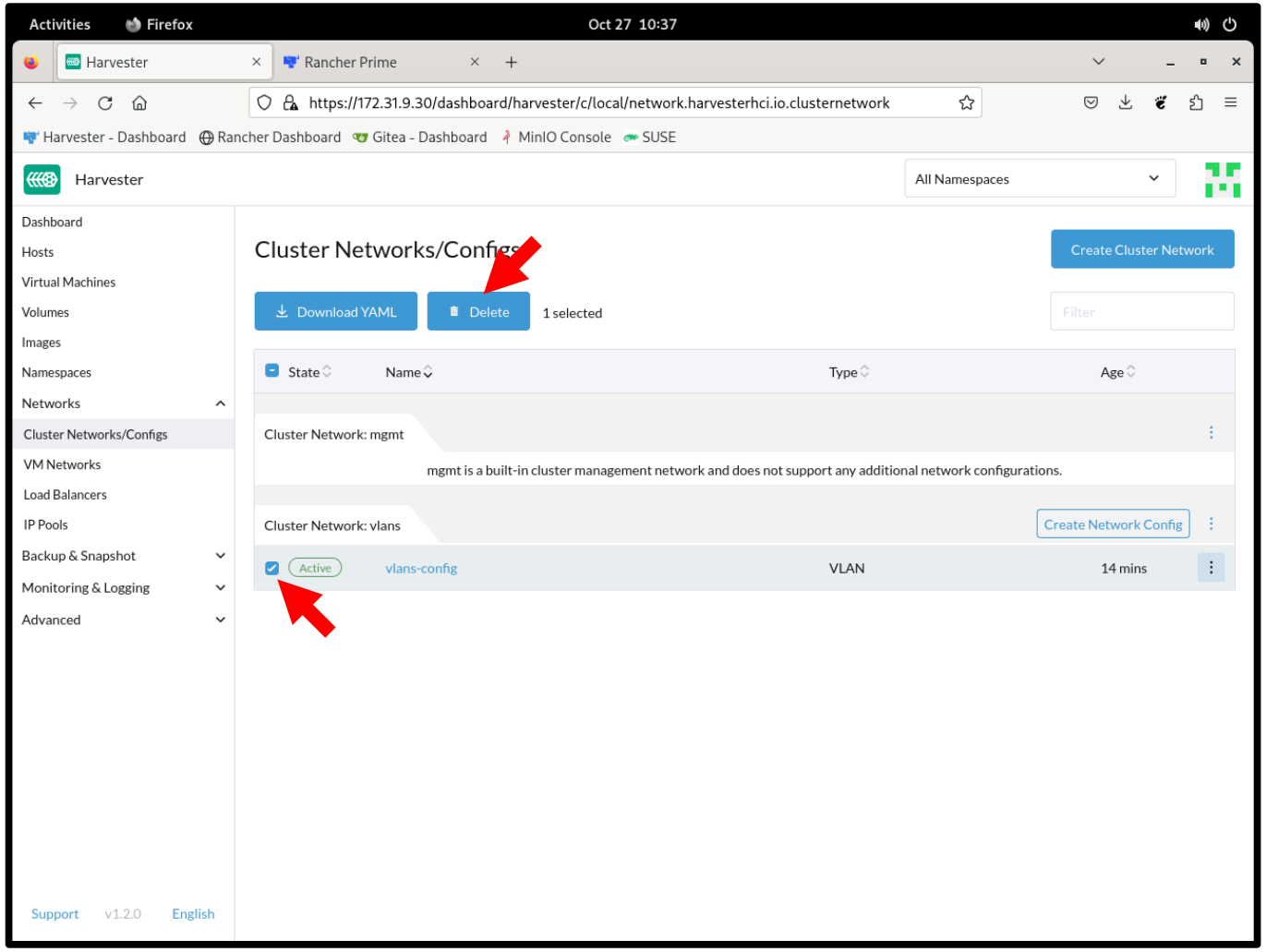
Modify Network Config

- Existing Network Configs can be modified after creation.
- Click on the **3 dot** menu and select: **Edit Config**



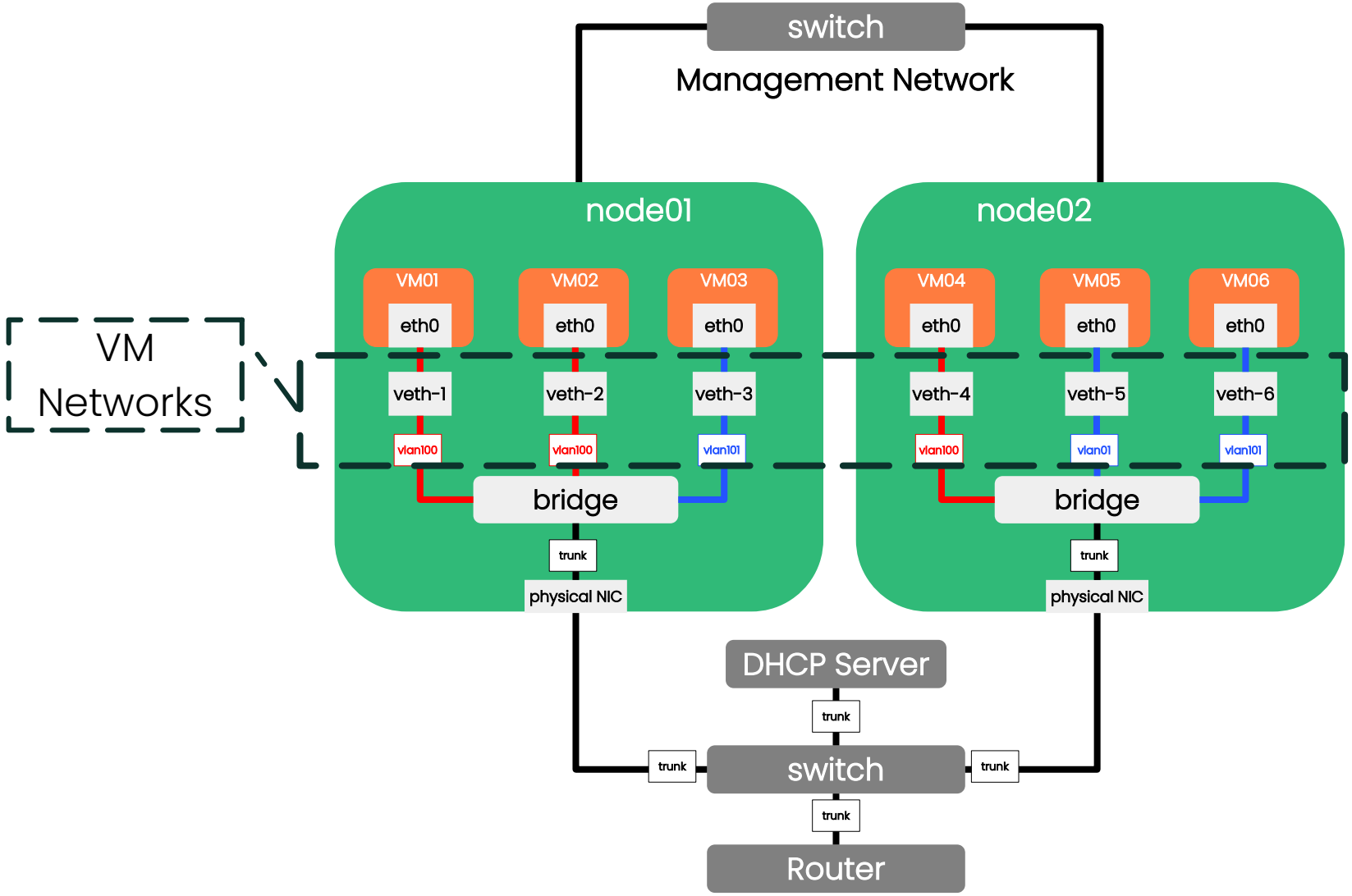
Delete Network Config

- Network Configs can also be deleted.
- Select the **check box** next to the Network Config and click: **Delete**



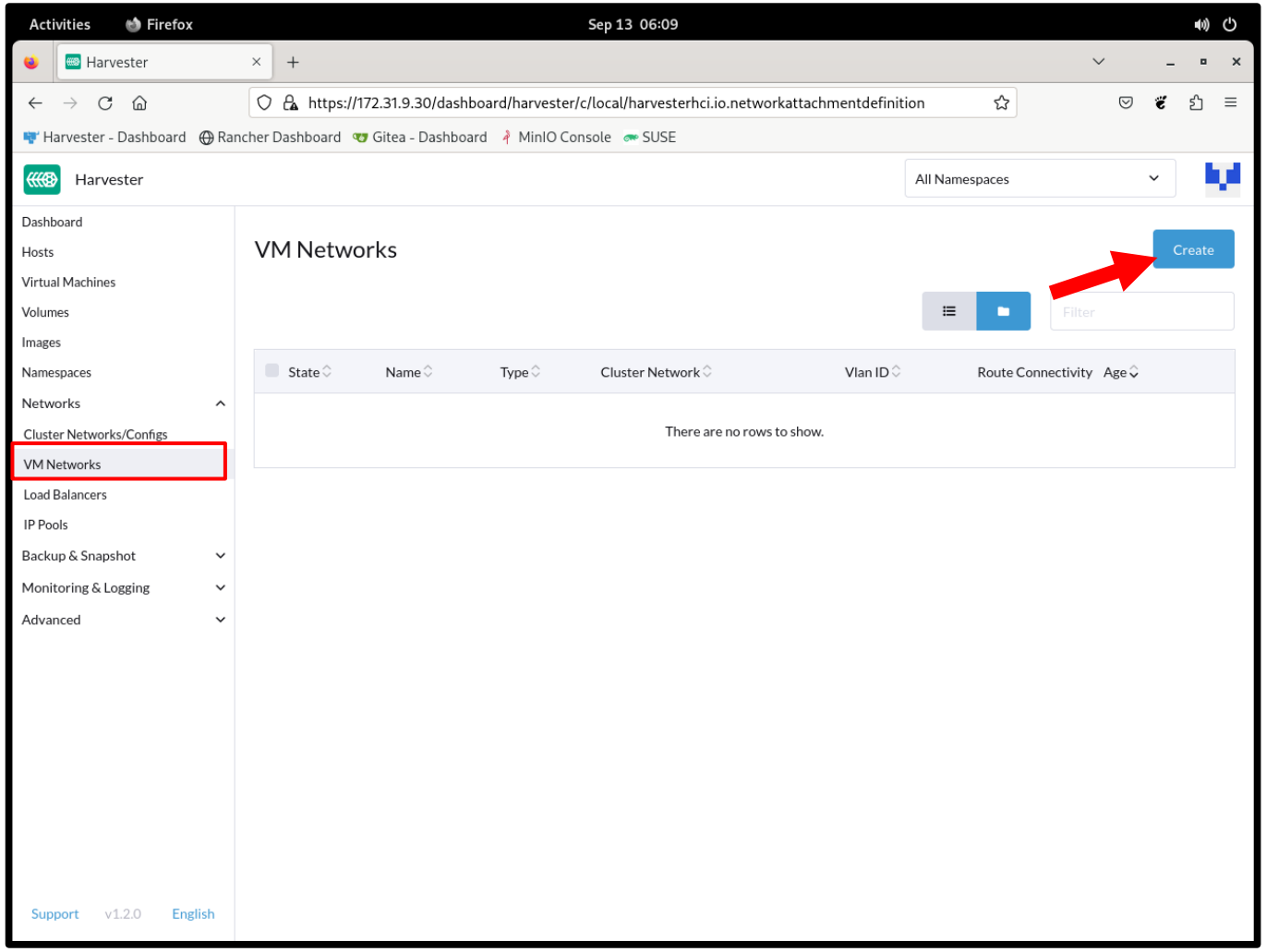
VM Networks

VM Networks



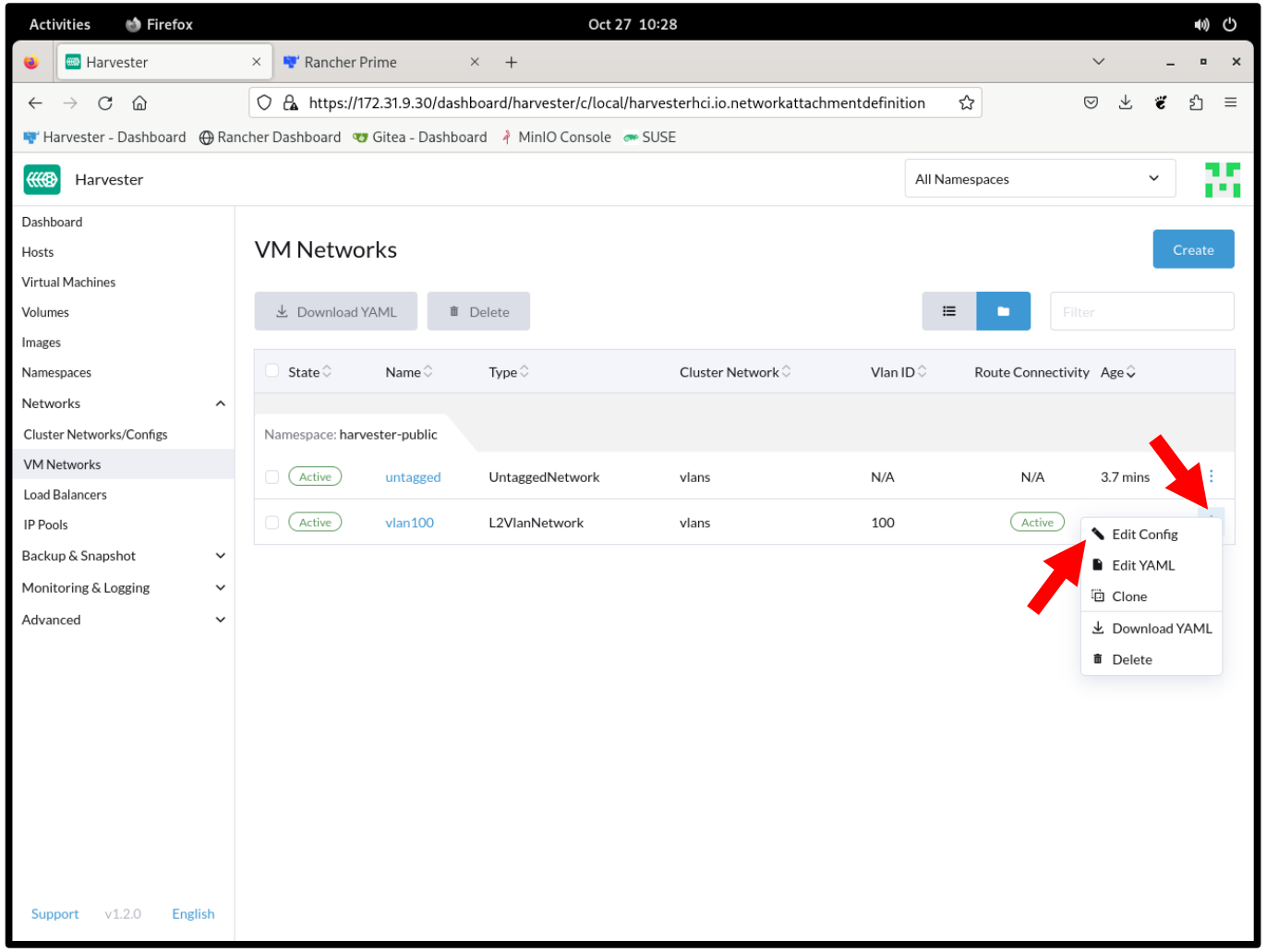
View/Create Networks

- Virtual networks are displayed and managed on the **Networks → VM Networks** screen.
- To create a new VM network click: **Create**



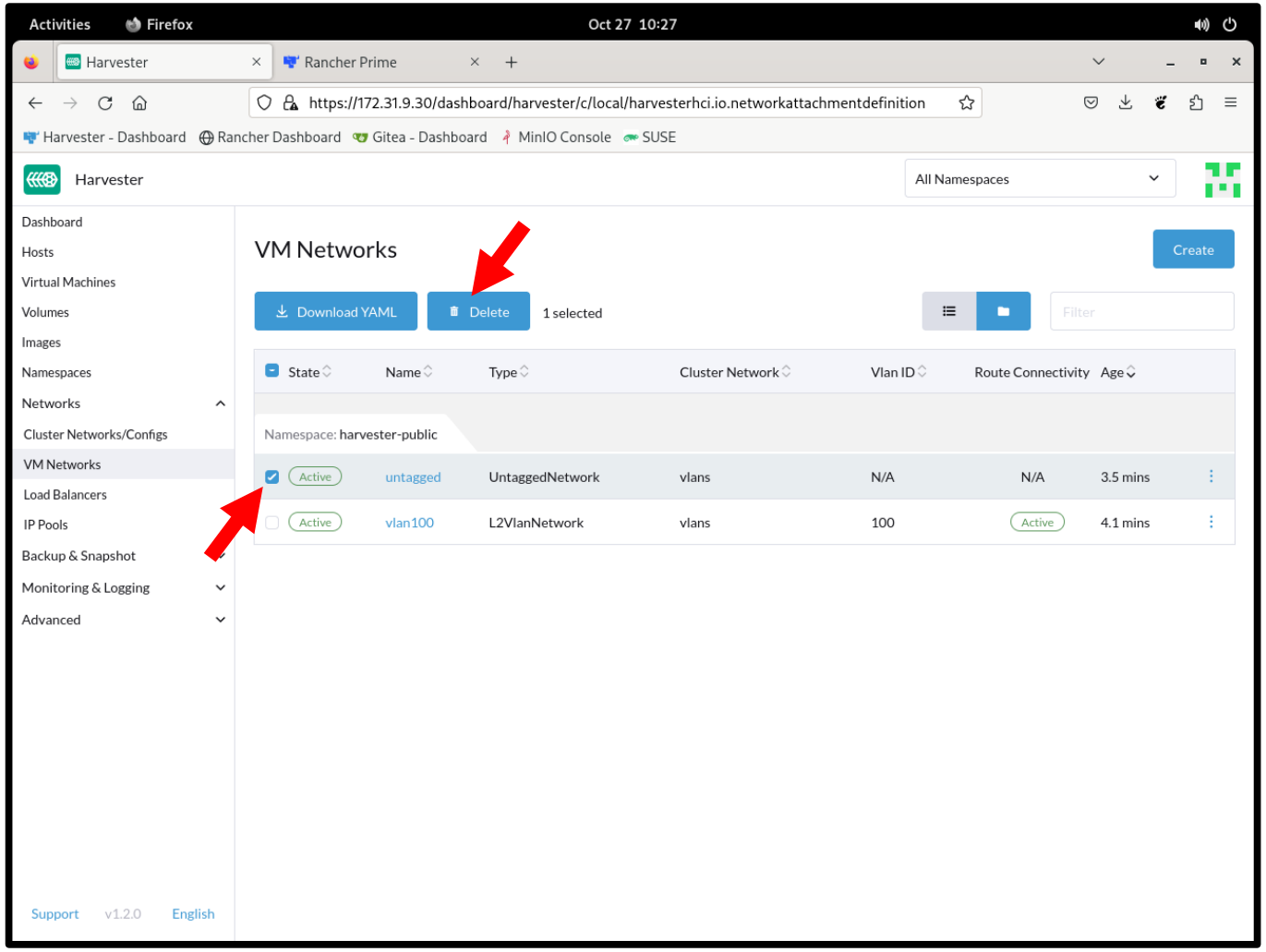
Modify Configured Networks

- Existing VM networks can be modified after creation.
- Click on the **3 dot** menu and select: **Edit Config**
- Before modifying VM networks, ensure that VMs using those network are shut down before making the modifications.



Delete Configured Networks

- VM Networks can be deleted.
- Select the **check box** next to the network and click: **Delete**
- Before deleting VM networks, ensure that VMs using those network are shut down first.



Section Summary

Section Summary

- Describe Harvester networking concepts
  - Harvester uses the concepts of Cluster Networks, Network Configs and VM Networks to provide networking in the cluster.
  - Cluster Networks are traffic-isolated forwarding paths.
  - Network Configs define how cluster nodes attach to Cluster Networks.
  - VM Networks are how VMs connect and communicate over the network.
  - Harvester uses a built-in cluster network named management for all intracluster communication.

Section Summary

- Create/modify/delete Cluster Networks and Network Configs
  - Cluster Networks must first be defined before Network Configs can be defined.
  - Network Configs are associated with Cluster Networks and a Cluster Network can have multiple Network Configs associated with it.
  - Network Configs define which NICs in cluster nodes to connect to a Cluster Network, the bonding type to use with those NICs and attributes of the connection such as MTU.
  - Cluster Networks, other than the management network, can be modified or deleted after creation however they cannot be deleted while they have Network Configs associated with them.

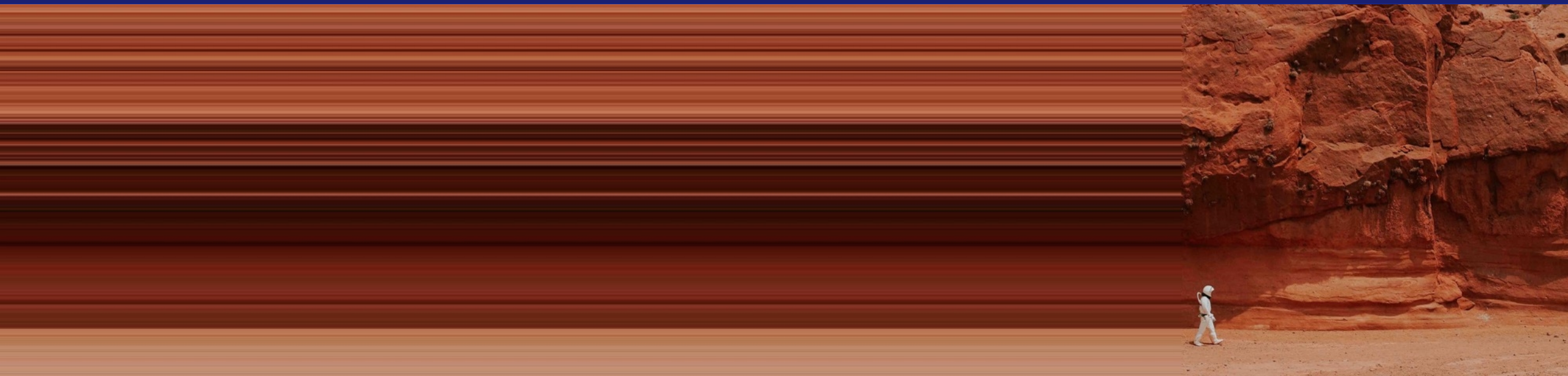
Section Summary

- Create/modify/delete VM networks
  - VM Networks can either be VLANs or untagged networks.
  - VM Networks are associated with Cluster Networks.
  - VM Networks define the type of network (VLAN or untagged) along with the IP address, gateway and, in the case of VLANs, the VLAN ID used by the network.
  - VM networks can be modified or deleted after creation but all VMs using the network must be powered off before they are modified or deleted.

SUSE

Section: 6

Storage in Harvester



Learning Objectives

At the end of this section you will be able to:

- Describe how storage is implemented and used in Harvester.
- Create an use storage volumes.
- Manage the amount of storage available in a Harvester cluster.

Topics

1 Longhorn Basics

2 Dedicated Storage Network

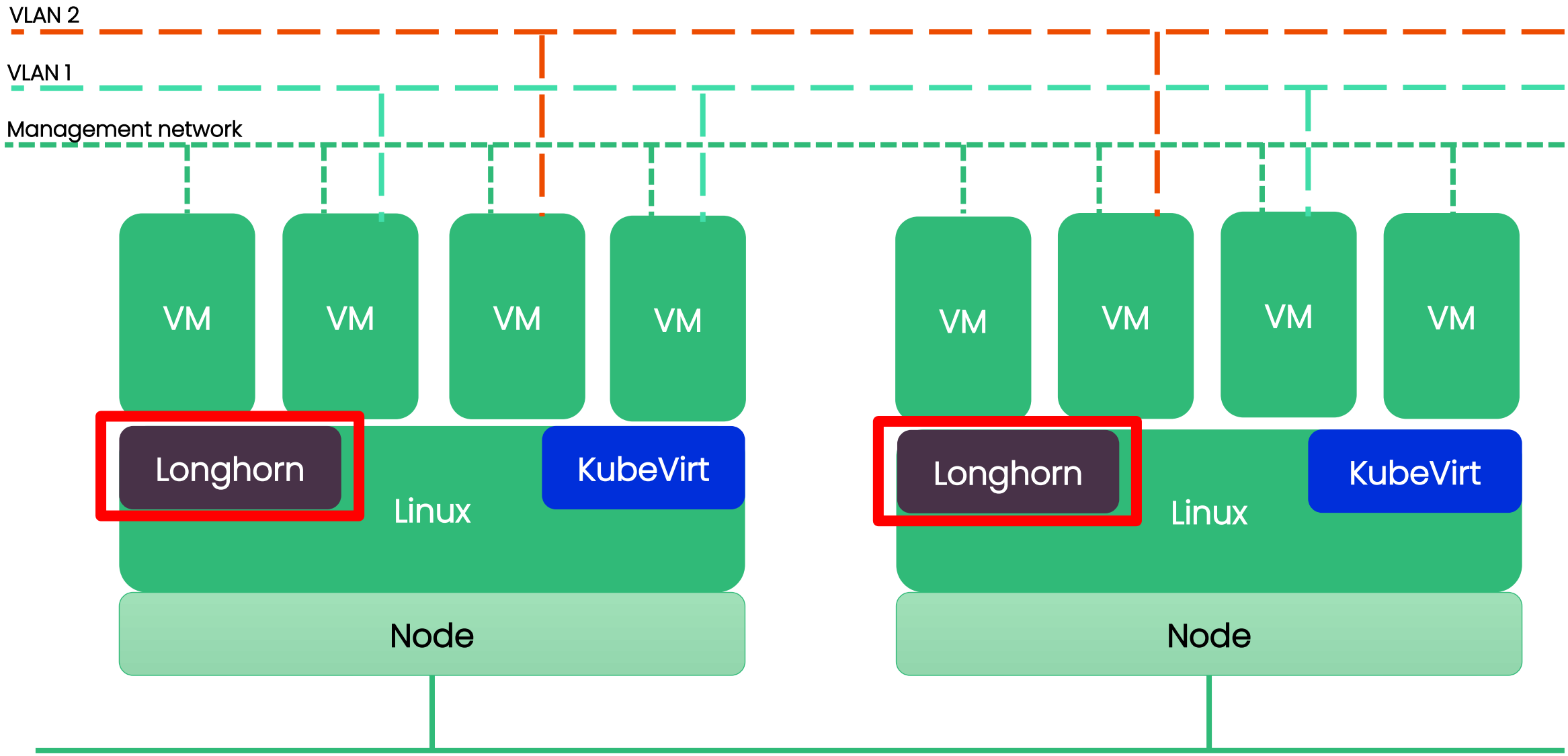
3 Storage Volumes

4 Storage Expansion

5 Storage Classes

Longhorn Basics

Harvester Components: Longhorn

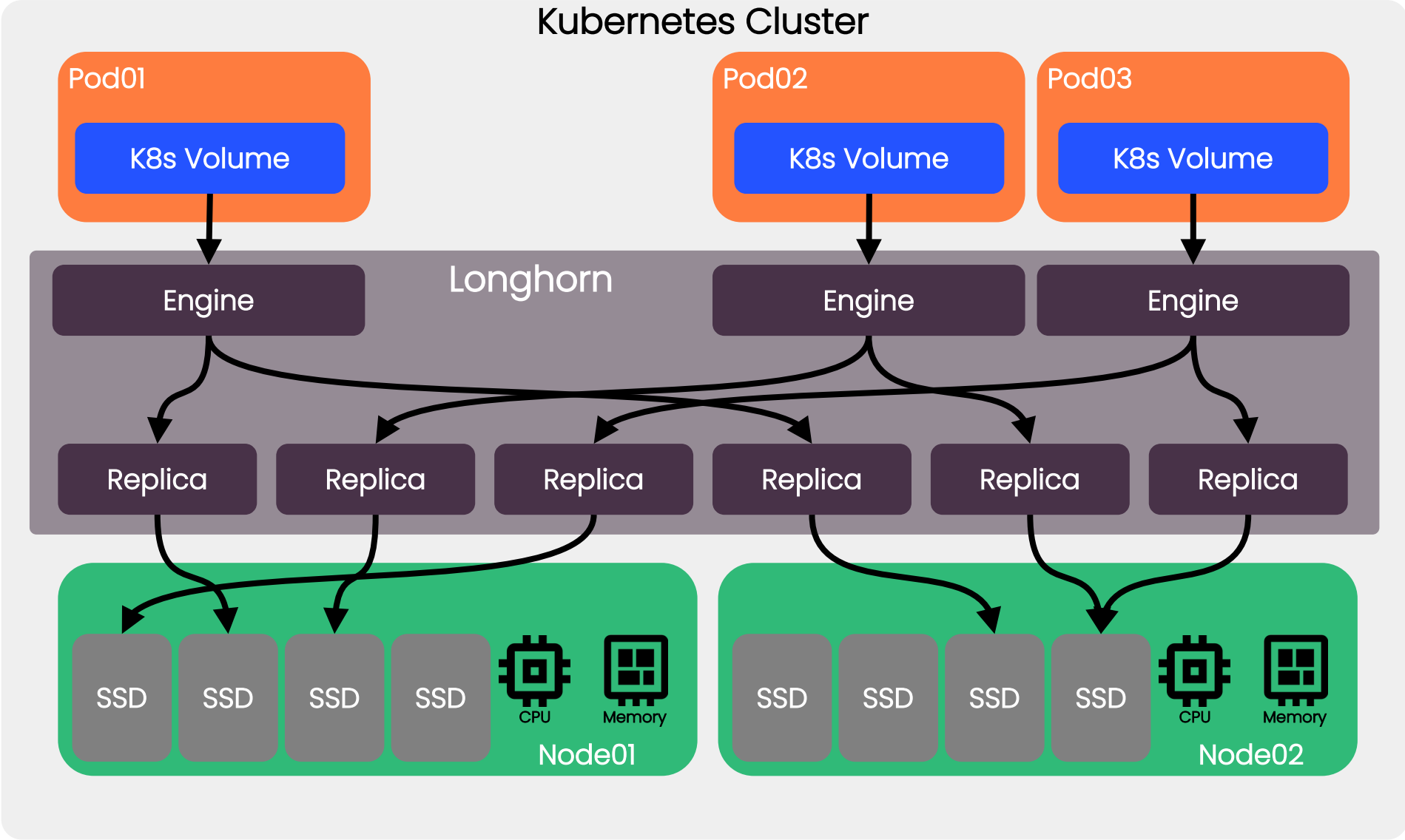


What is Longhorn?

- It is a lightweight, reliable, powerful distributed block storage for Kubernetes.
- It implements distributed block storage using containers and microservices.
- It provides:
  - Distributed block storage with no single point of failure using thin provisioned volume replicas.
  - Snapshot and Backup support.
  - Native Kubernetes Persistent Volume (PV) and Persistent Volume Claim (PVC) support via a Kubernetes Container Storage Interface (CSI) driver.

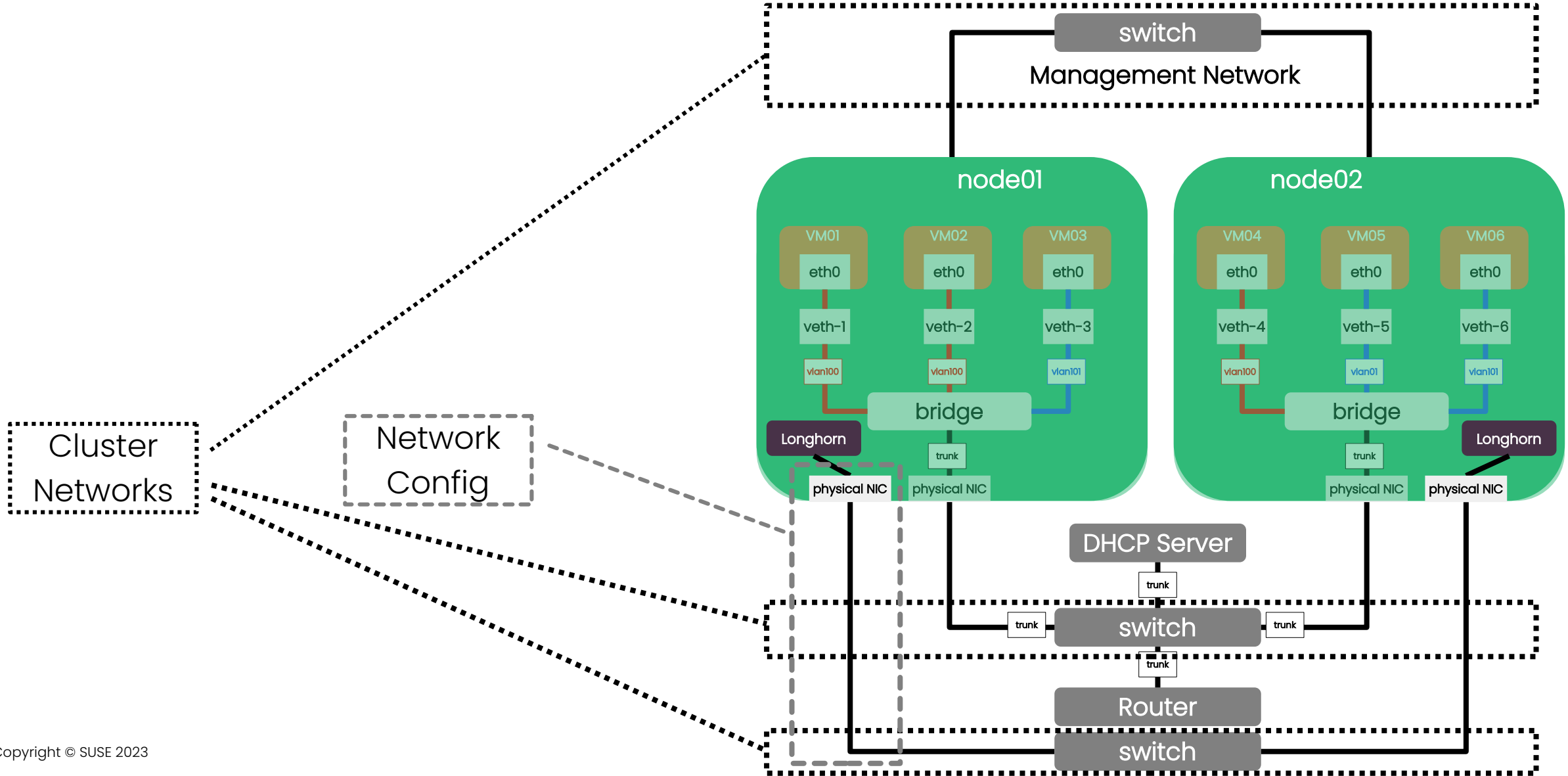
Longhorn Architecture

- Persistent **volumes** created by Longhorn are attached to **pods**.
- The Longhorn engine creates **replicas** of the volumes.
- The replicas are spread across the **nodes** and **disks**.



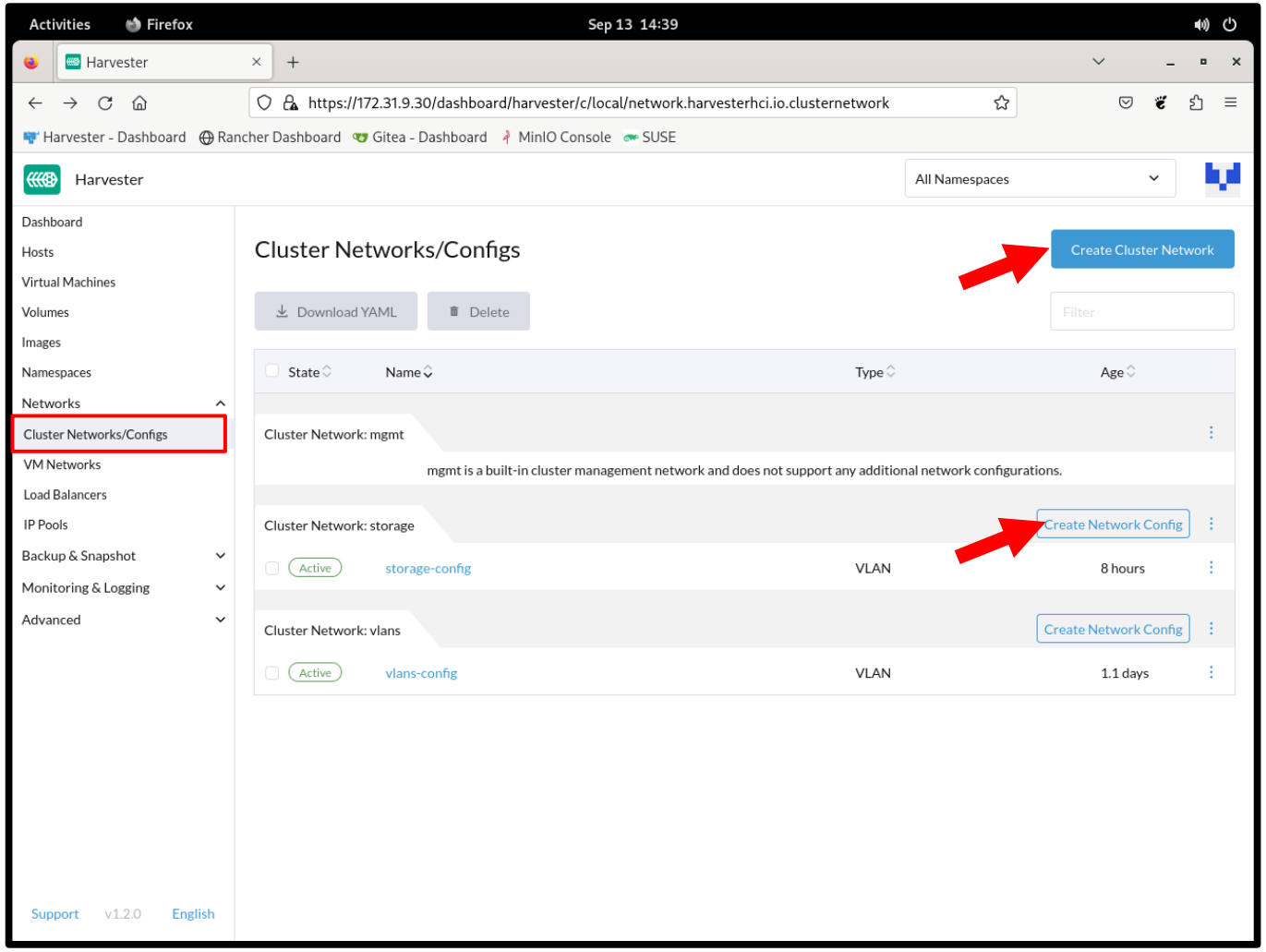
Dedicated Storage Network

Dedicated Storage Network



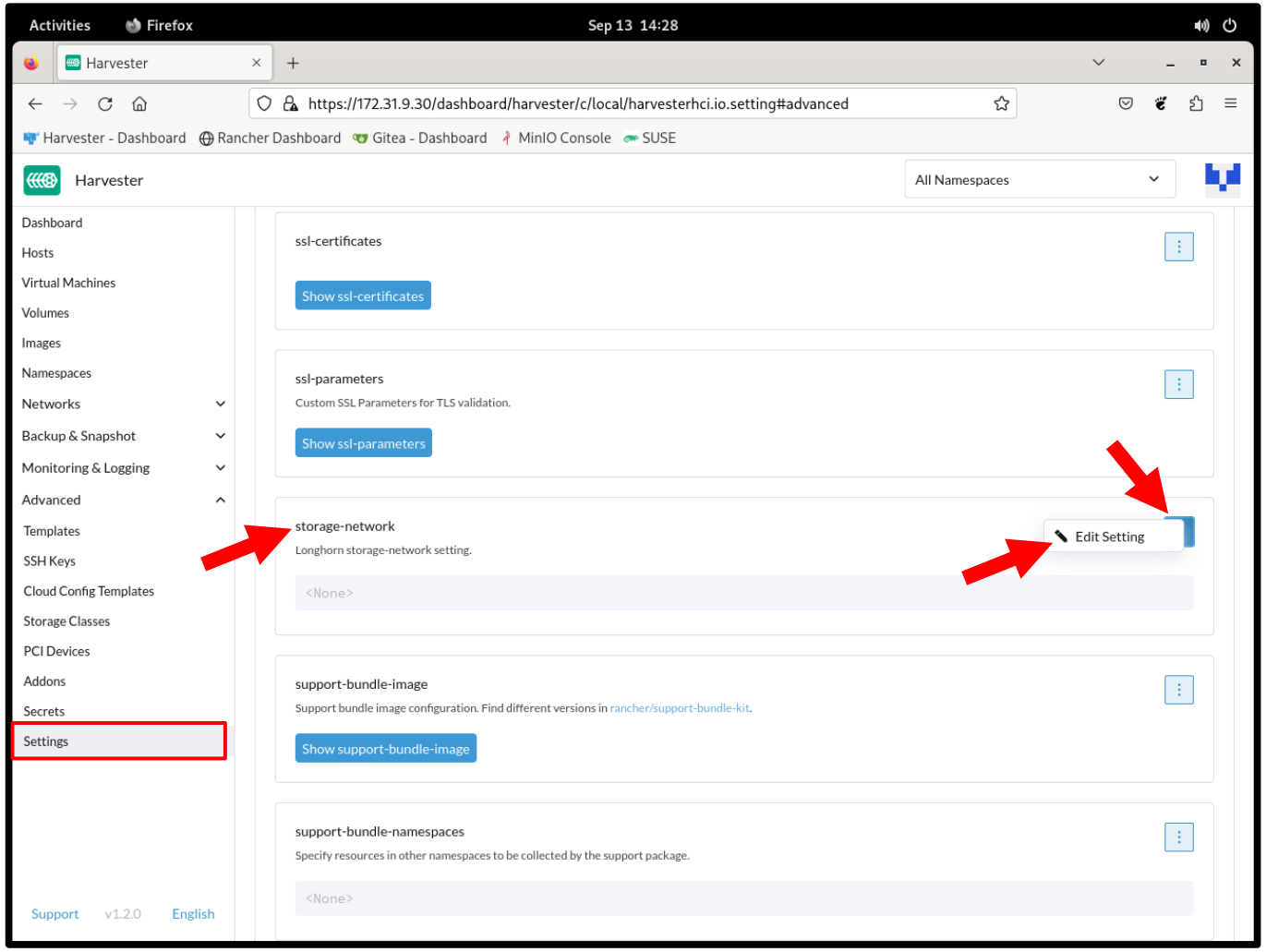
Create Cluster Network for Storage

- The **Cluster Network** and **Network Config** are created in the same manner as other cluster networks by going to the **Networks → Cluster Networks/Configs** screen.
- First a **Cluster Network** must be created by clicking: **Create Cluster Network**
- Then a **Network Config** must be created for that **Cluster Network** by clicking: **Create Network Config**



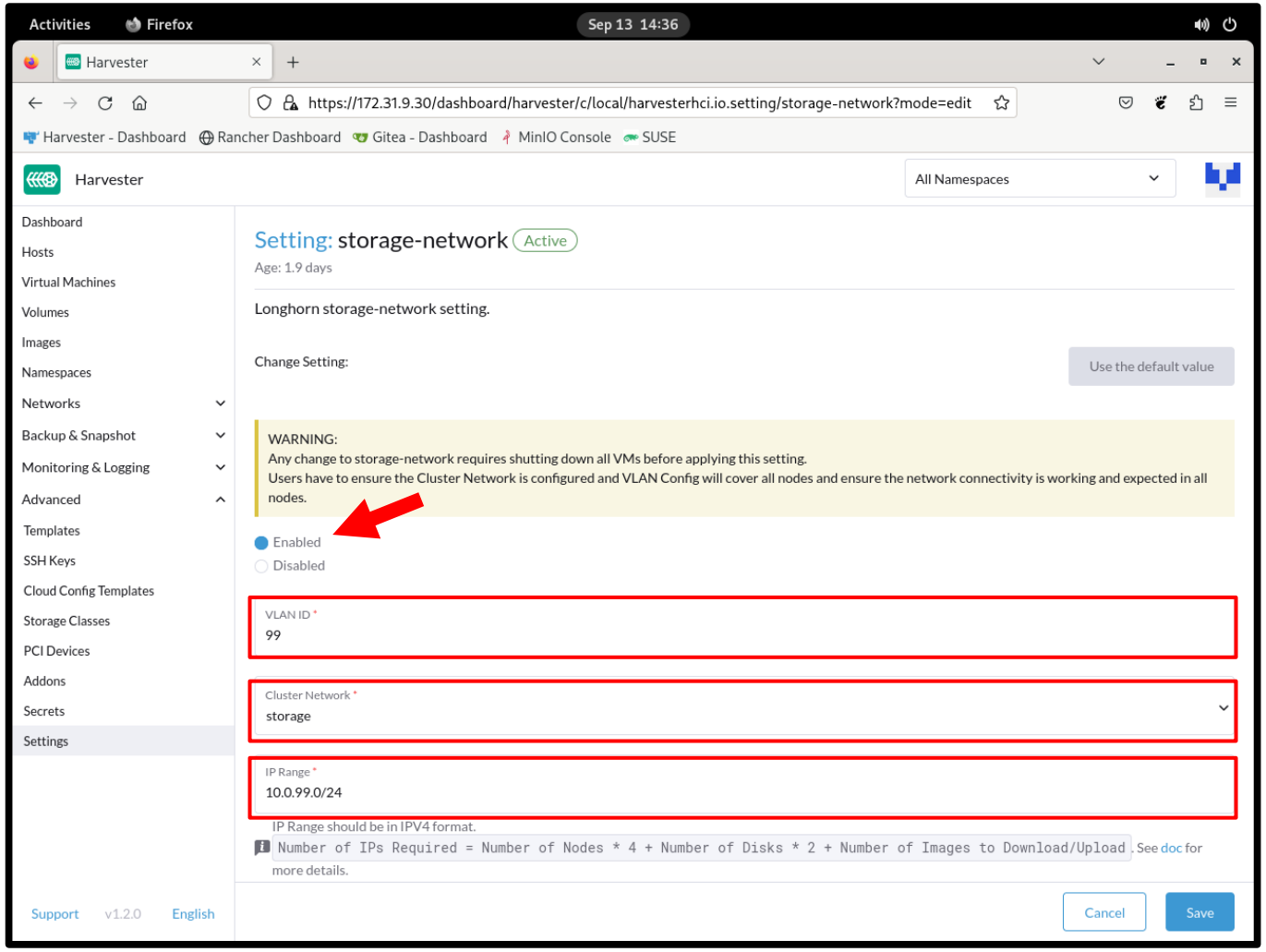
Configure Harvester to Use the Storage Network

- To configure Longhorn to use the Storage network, go to the **Settings** screen.
- Scroll down to the **storage-network** section.
- Click the **3-dot** menu and select: **Edit Setting**



Configure Harvester to Use the Storage Network

- On the **Setting: storage-network** screen select **Enabled**.
- Enter a **VLAN ID** for the Storage network.
- Select the **Cluster Network** to use.
- Enter an **IP Range** for the storage resources.
  - Note the formula listed below the IP Range field to use for th



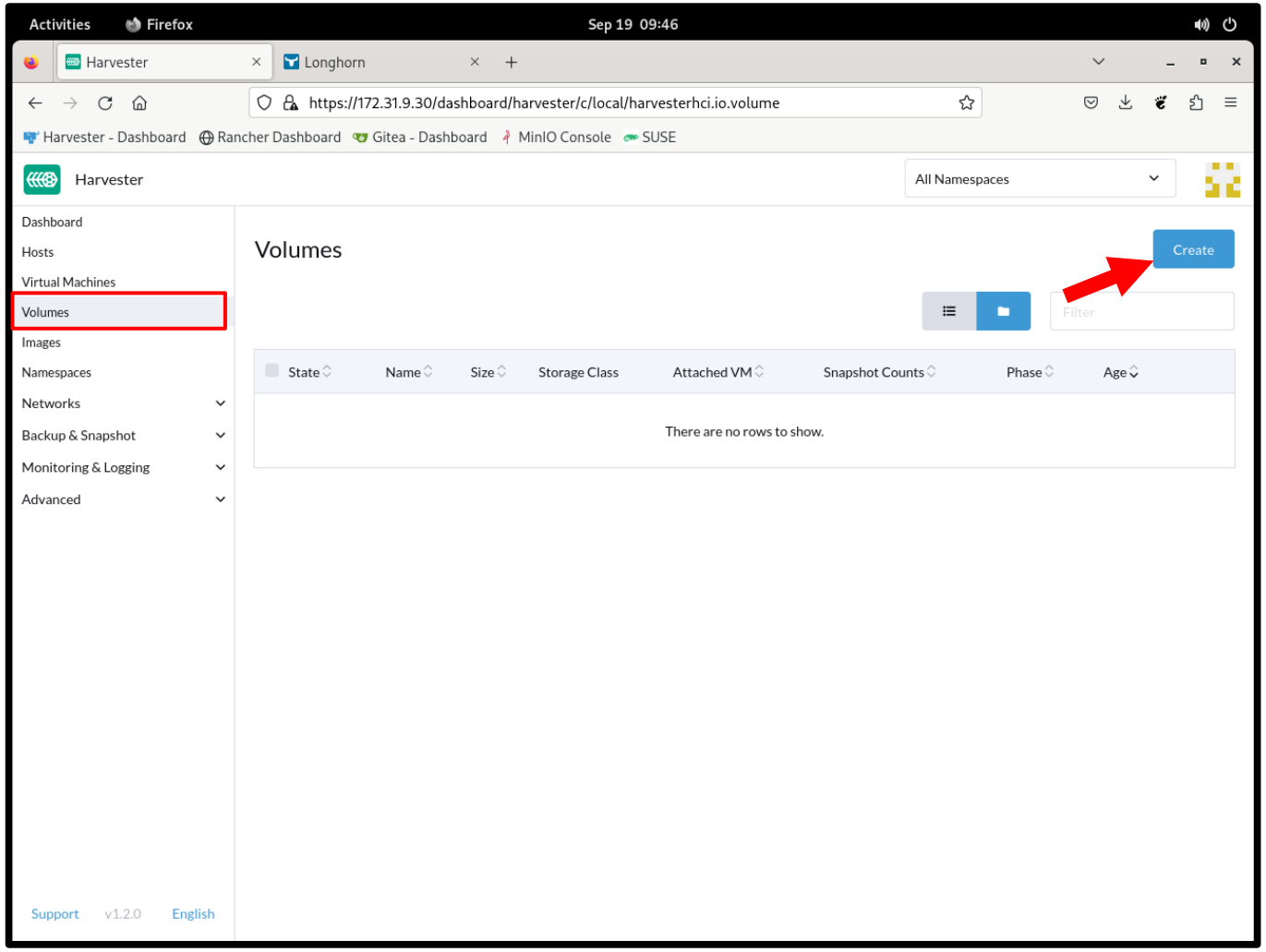
Storage Volumes

Storage Volumes in Harvester

- Storage volumes can be created and used as follows:
  - Volumes are created automatically when VM's are created and used as the VMs' rootdisks.
  - Volumes can be manually created as empty volumes that can then be later attached to VMs.
  - Volumes can be created from images that can then be later attached to VMs.

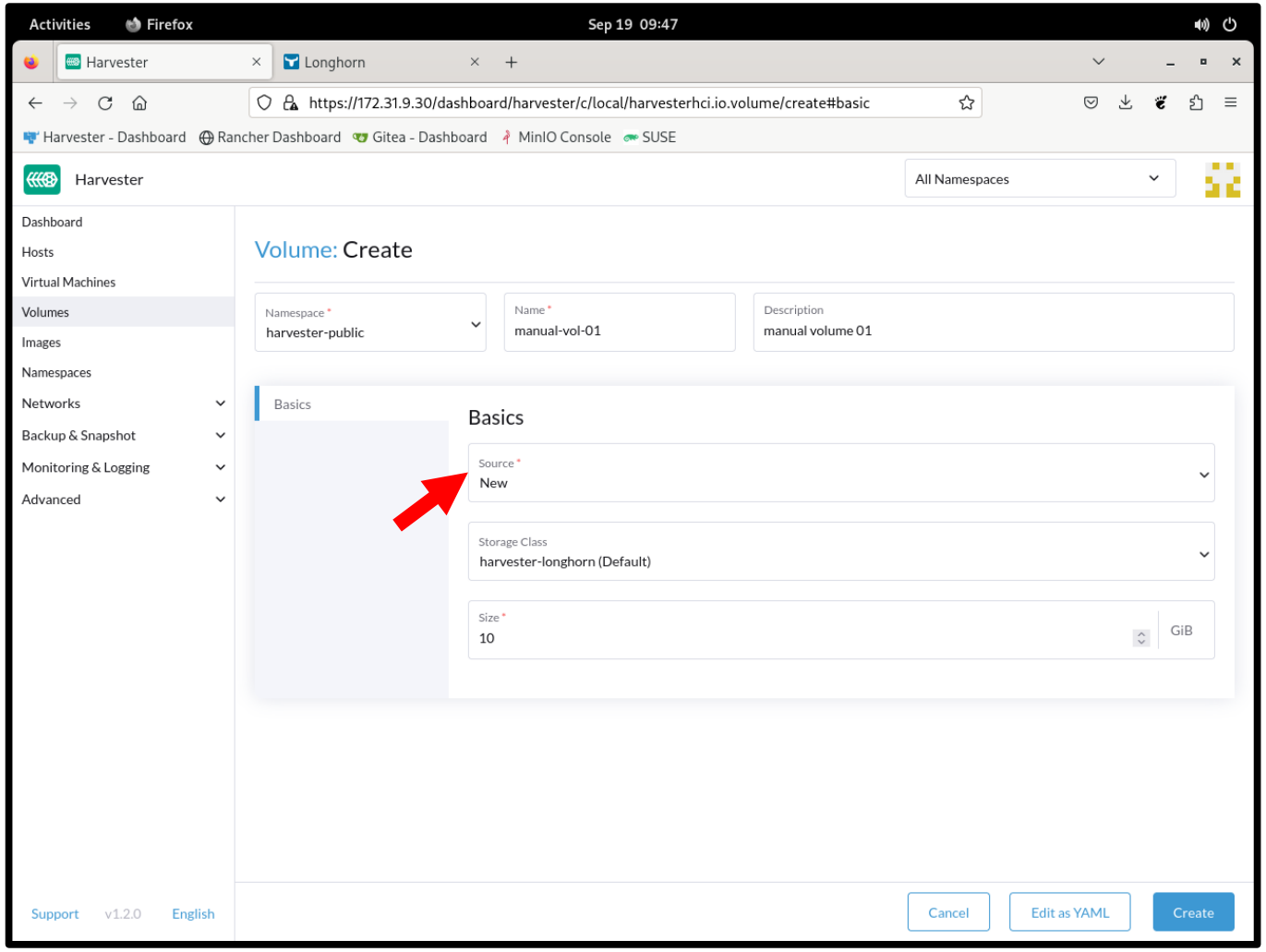
View and Manually Create Volumes

- All existing volumes are listed on the **Volumes** screen.
- To manually create a volume click: **Create**



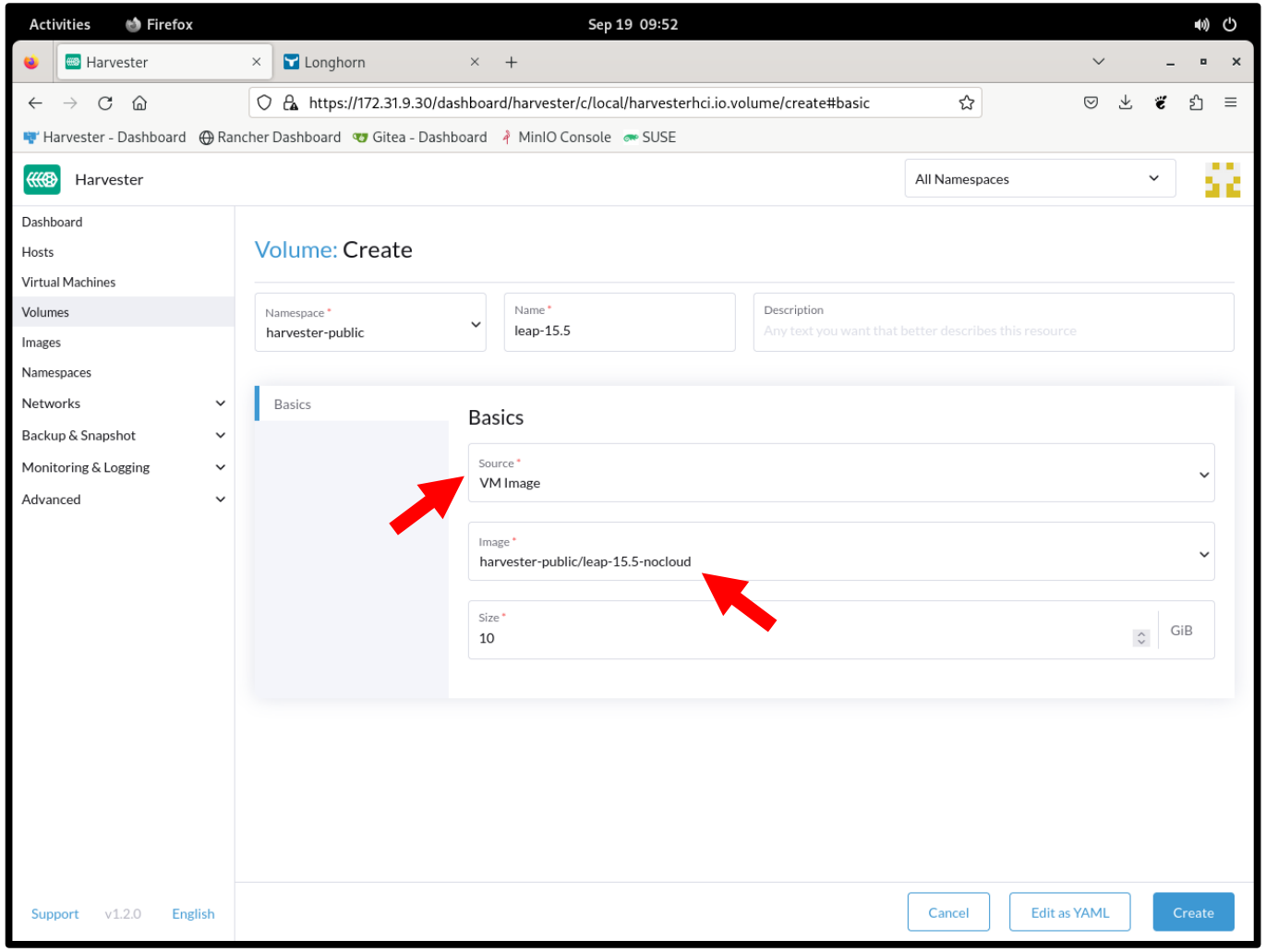
Manually Create an Empty Volume

- New empty volumes can be created by selecting **New** for the **Source**.



Manually Create a Volume from an Image

- New volumes that are already populated with data can be created by selecting **VM Image** for the **Source** and then selecting the existing **Image**.



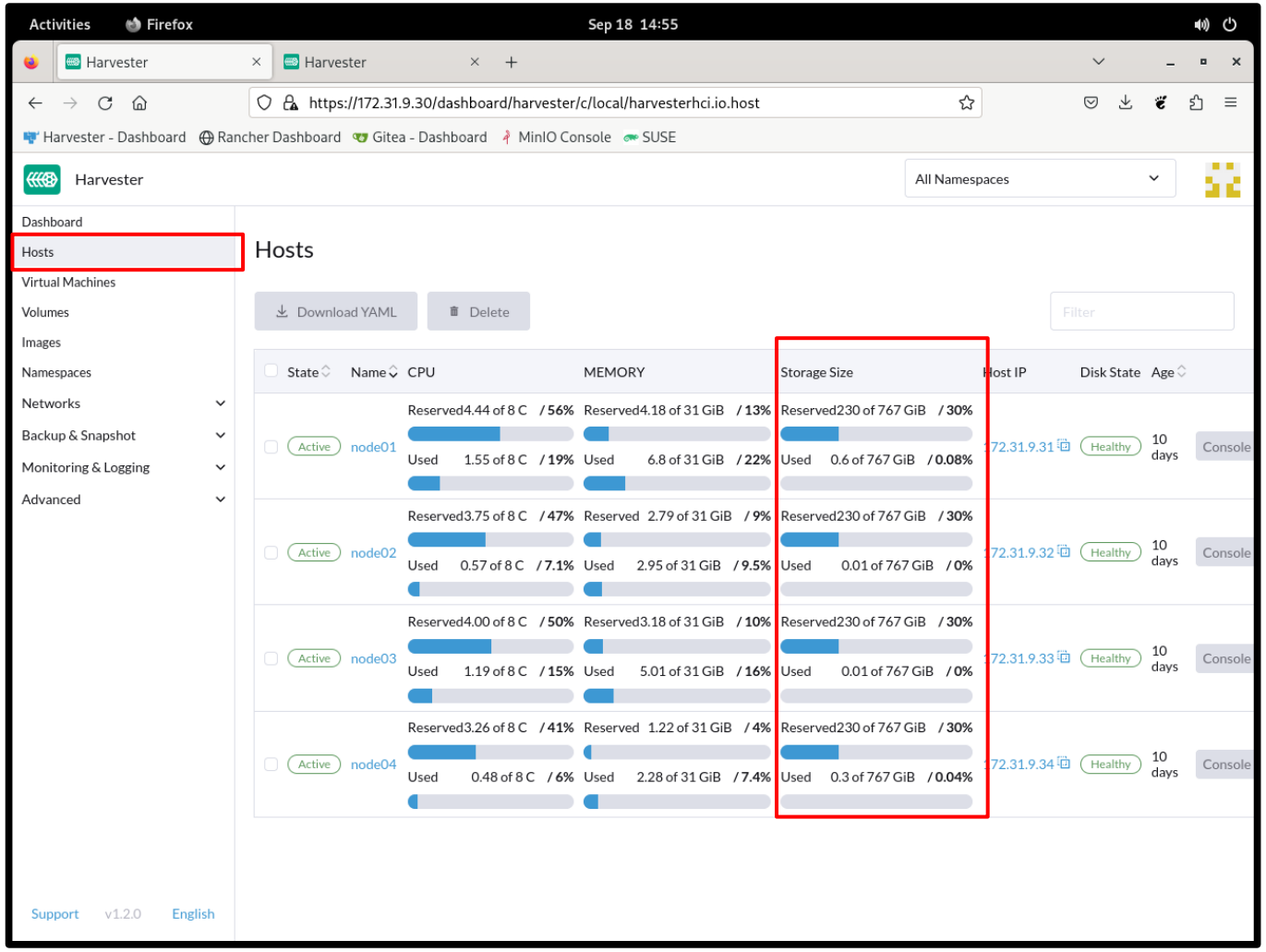
Storage Expansion

Add Additional Storage

- Additional storage can be added to a Harvester cluster post deployment.
- Additional disks in the cluster nodes can be added to extend the amount of storage available to the embedded Longhorn cluster.
- Any additional disks added post install can also be removed from the Harvester cluster as long as there is sufficient space in the cluster for the used data to be relocated.

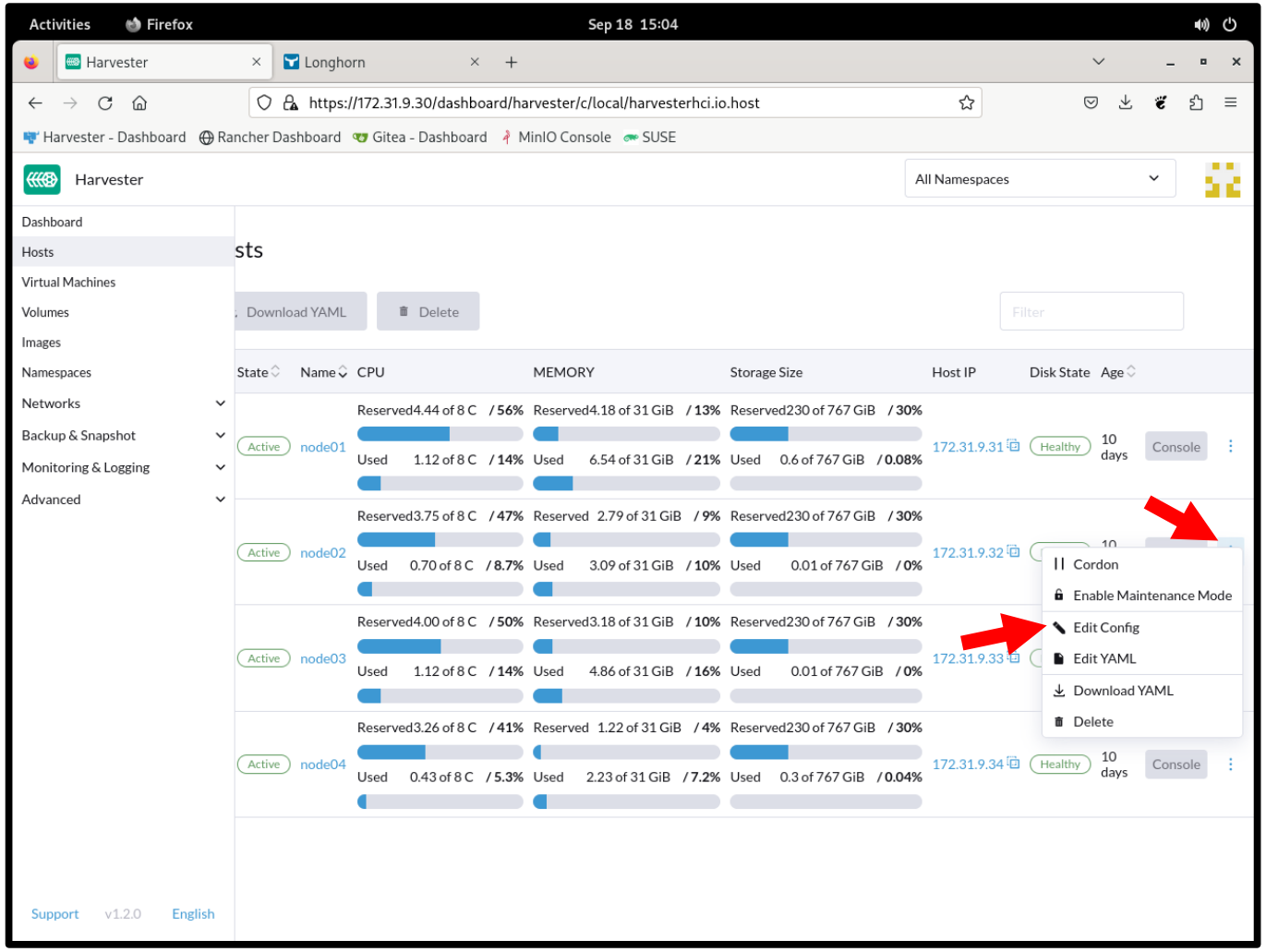
Hosts: Storage Size

- The **Hosts** screen displays the storage size and utilization for all cluster nodes.



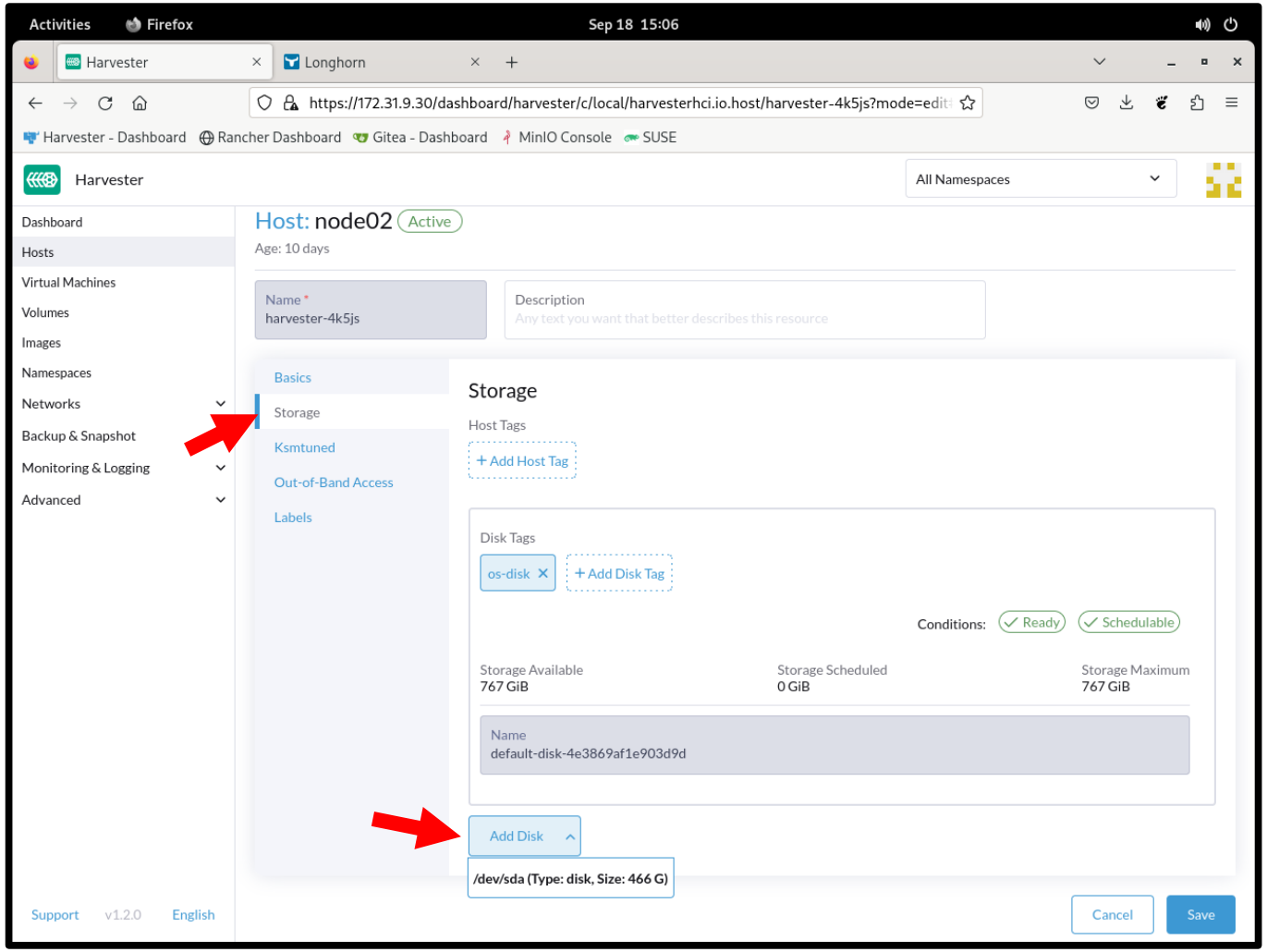
Hosts: Edit Host Config

- To add additional storage to a node click the **3 dot** menu next to the node.
- Select: **Edit Config**



Add Additional Disk

- On the node's edit config page select the **Storage** tab.
- From the **Add Disk** drop-down list select the disk to add.



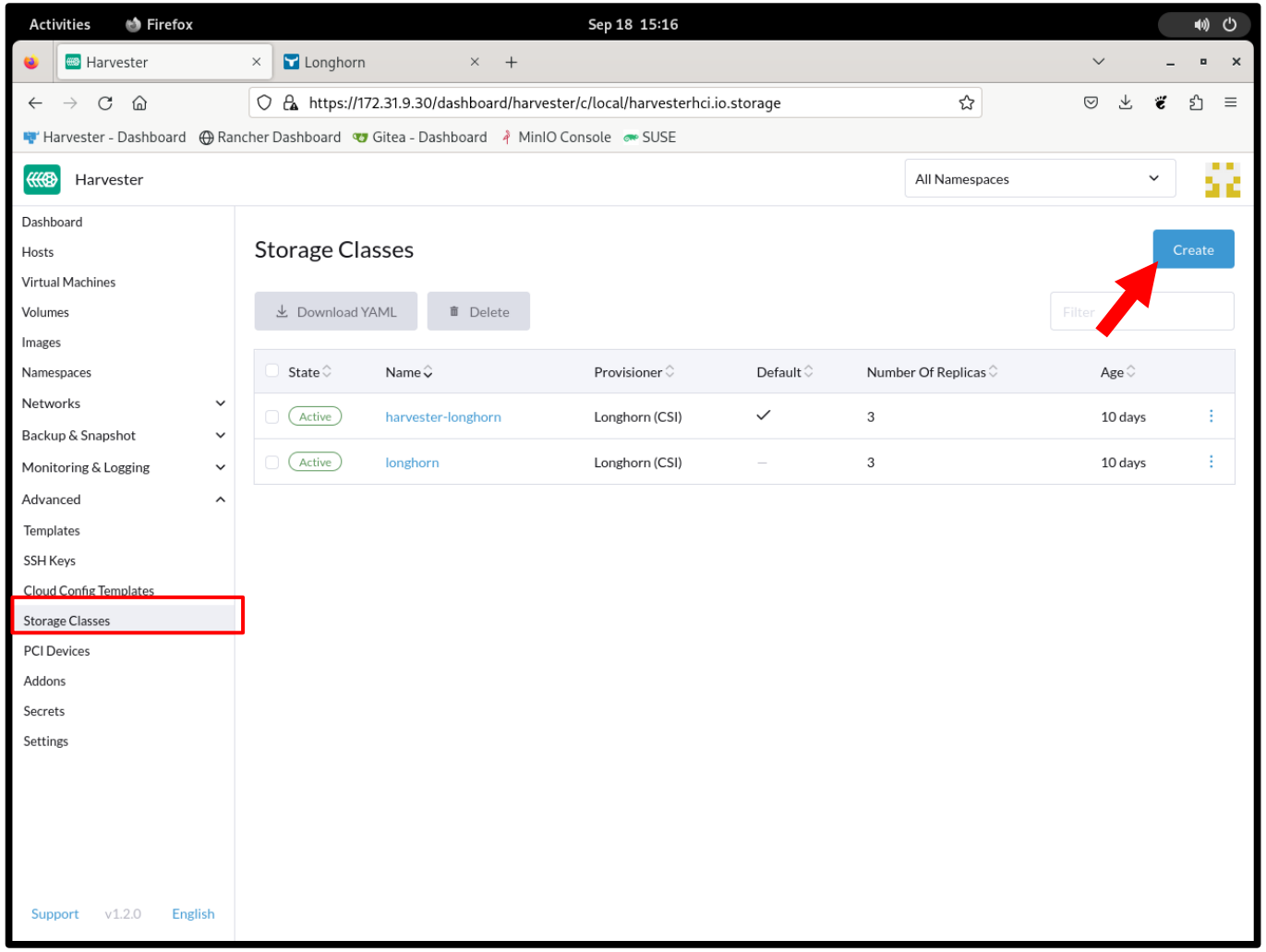
Storage Classes

Storage Classes in Harvester

- Storage Classes can be used to provide different storage features or levels of performance.
- Default storage classes named **harvester-longhorn** and **longhorn** are created during the install.
- Other storage classes can be configured post install.

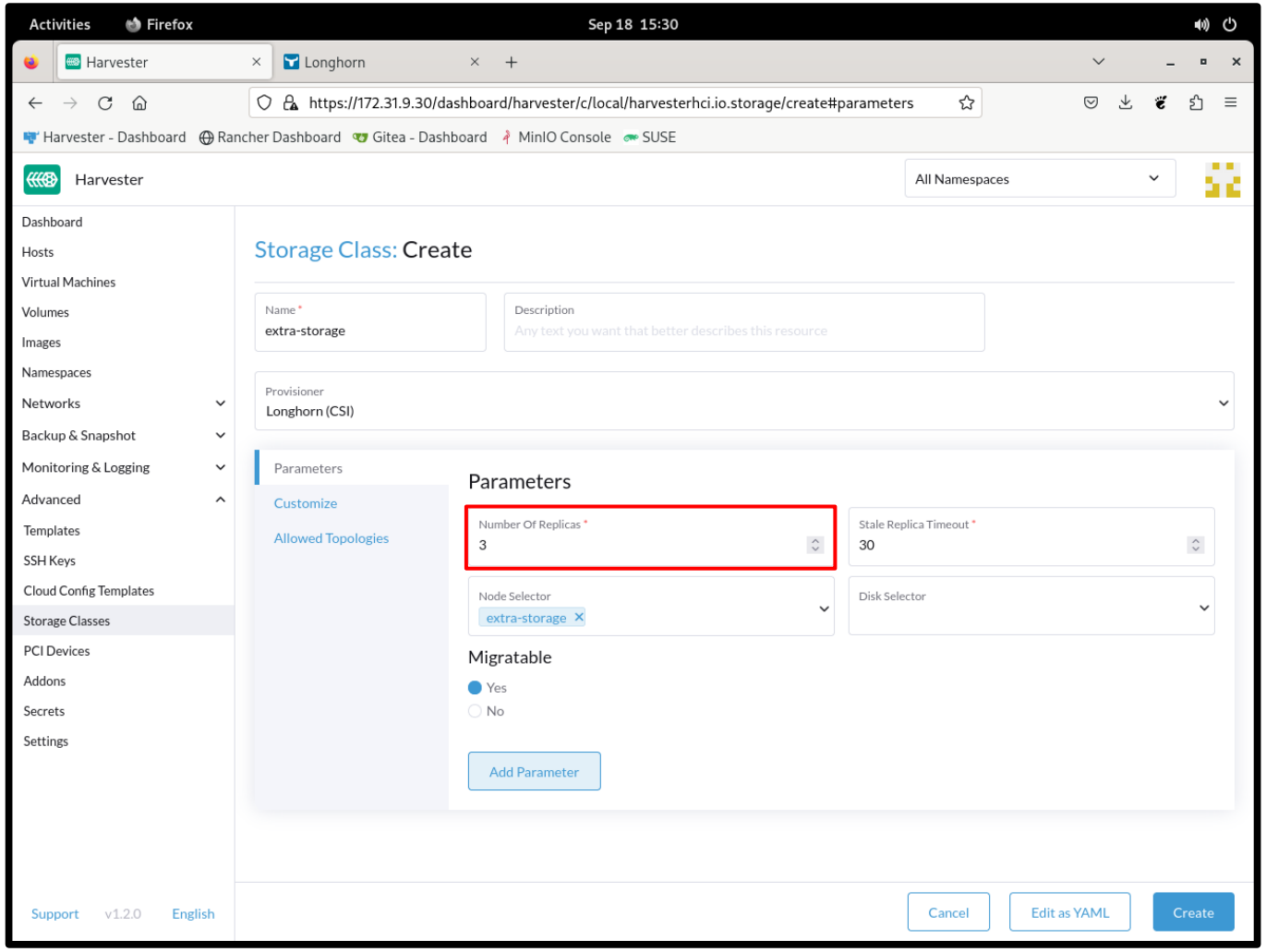
Create a Storage Class

- Storage Classes are managed on the **Advanced → Storage Classes** screen.
- To create a new Storage Class click: **Create**



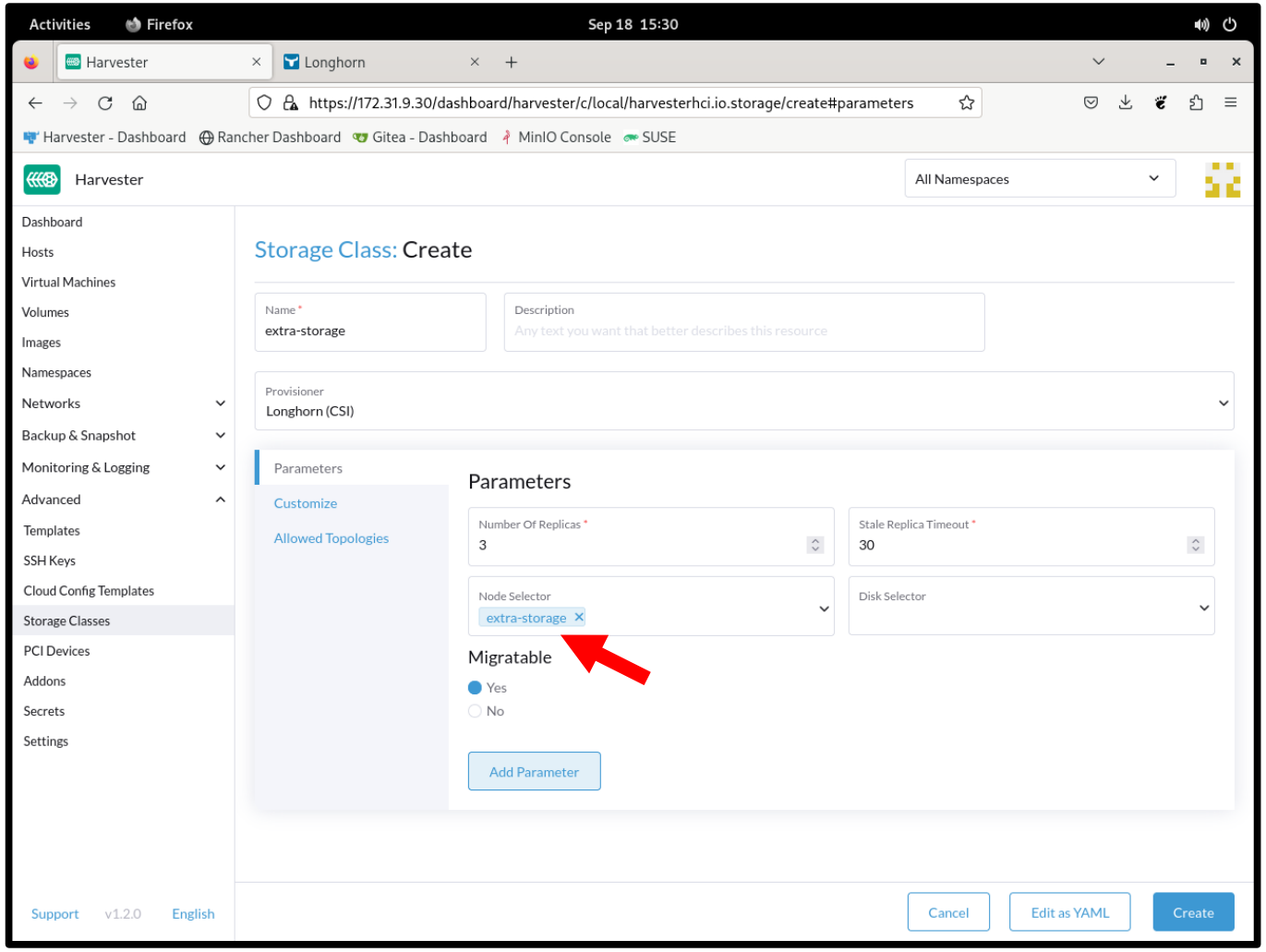
Create a Storage Class - Replicas

- The number of **Number of Replicas** of a volume that will be created can be specified.
- The more replicas created for a volume the more redundancy and therefore higher fault tolerance it has.
- The minimum recommended number of replicas is **3**.



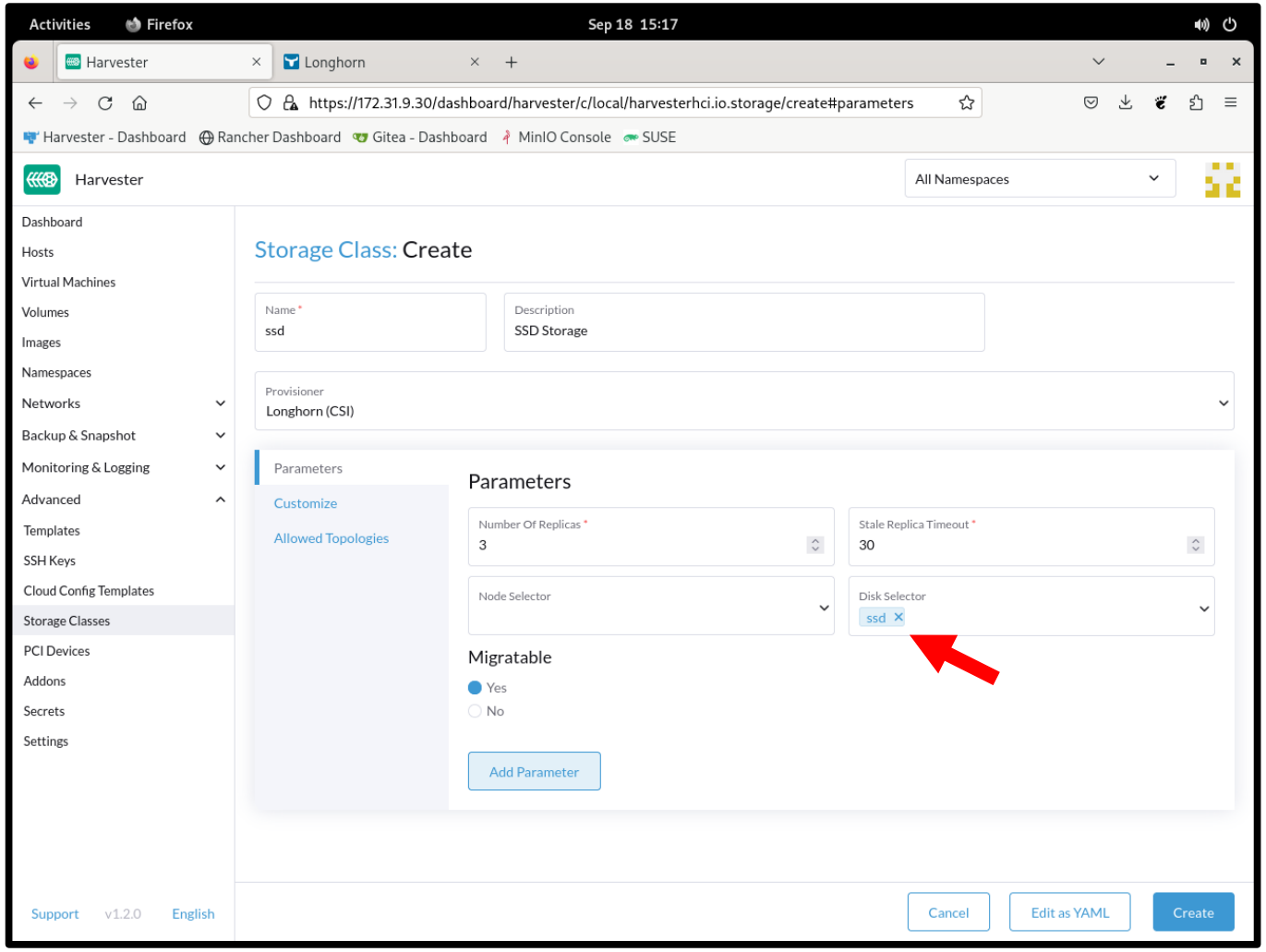
Create a Storage Class - Node Selector

- A **Storage Class** can be created to use **Node Selectors**.
- **Node Selectors** are used by Longhorn to know on which nodes to create volumes.



Create a Storage Class - Disk Selector

- A **Storage Class** can also be created to use **Disk Selectors**.
- **Disk Selectors** are used by Longhorn to know on which disks in which nodes to create volumes.



Section Summary

Section Summary

- Describe how storage is implemented and used in a Harvester Cluster
  - Harvester has an embedded instance of Longhorn to provide storage volumes for all cluster needs.
  - Volumes are used as both the root disks of VMs as well as data disks.
  - A dedicated Cluster Network can be created for use by Longhorn to isolate storage traffic from other cluster traffic and VM traffic.
- Create and use storage volumes
  - Storage volumes are created automatically when VMs are created.
  - Storage volumes can be manually created and attached to VMs.

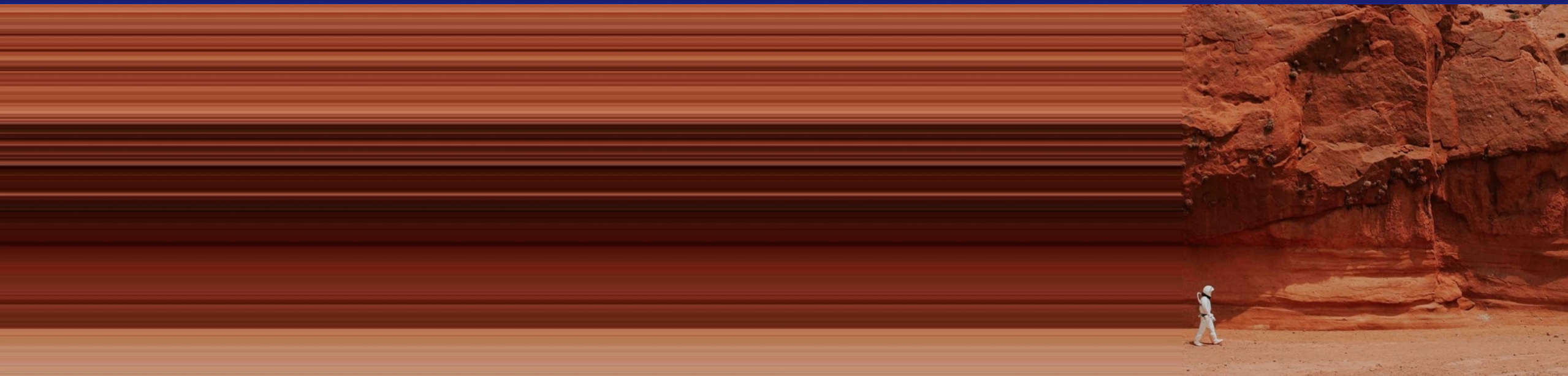
Section Summary

- Manage the amount of storage available in a Harvester cluster
  - Additional storage can be added to the embedded Longhorn cluster by adding disks/partitions/logical volumes attached to the cluster nodes.

SUSE

Section: 7

Virtualization Management in Harvester



Learning Objectives

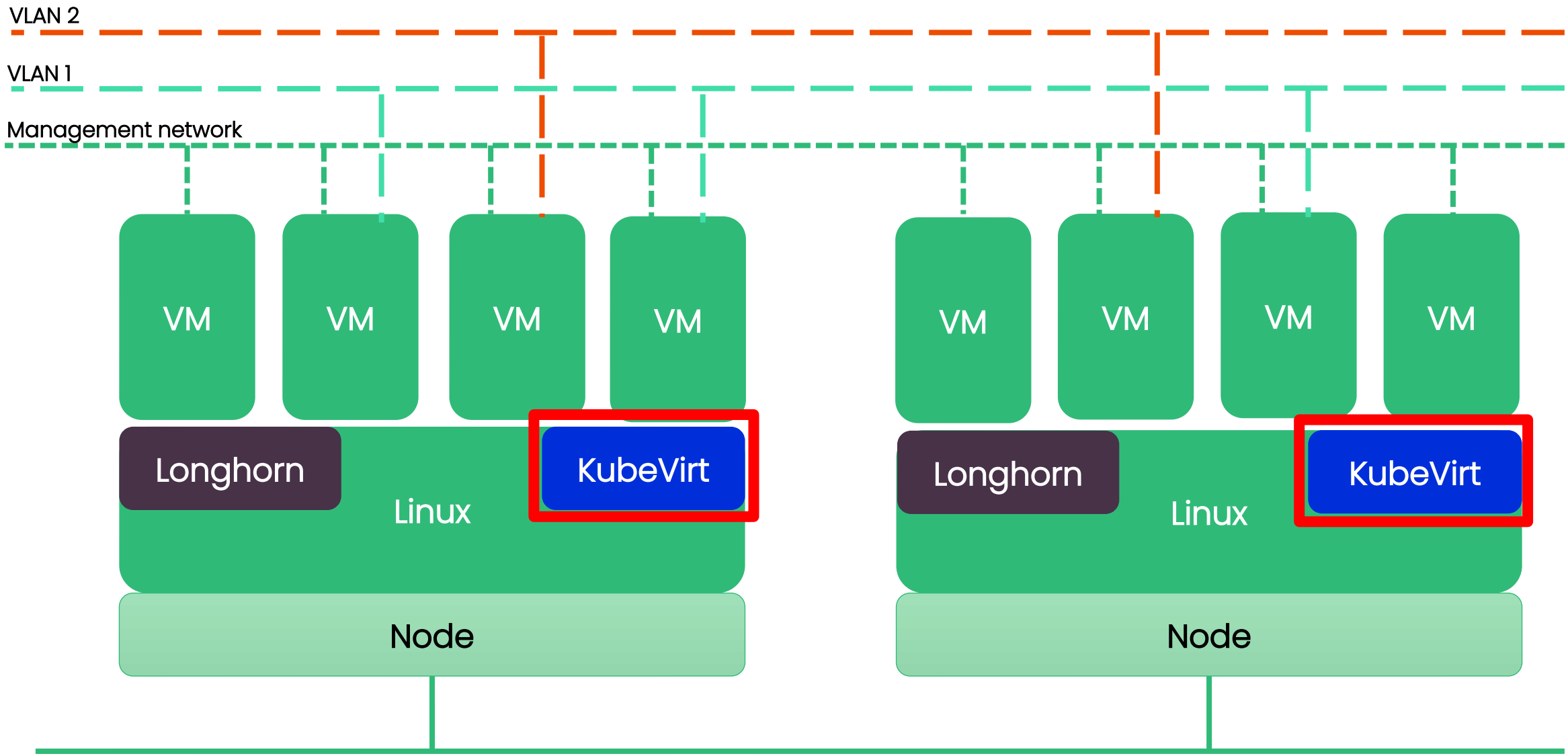
At the end of this section you will be able to:

- Describe virtualization concepts in Harvester.
- Upload and use virtual machine images.
- Create and manage the lifecycle of virtual machines on Harvester.
- Backup and restore virtual machines.

Topics

Virtual Machine Management Concepts in Harvester

Virtualization in Kubernetes - KubeVirt

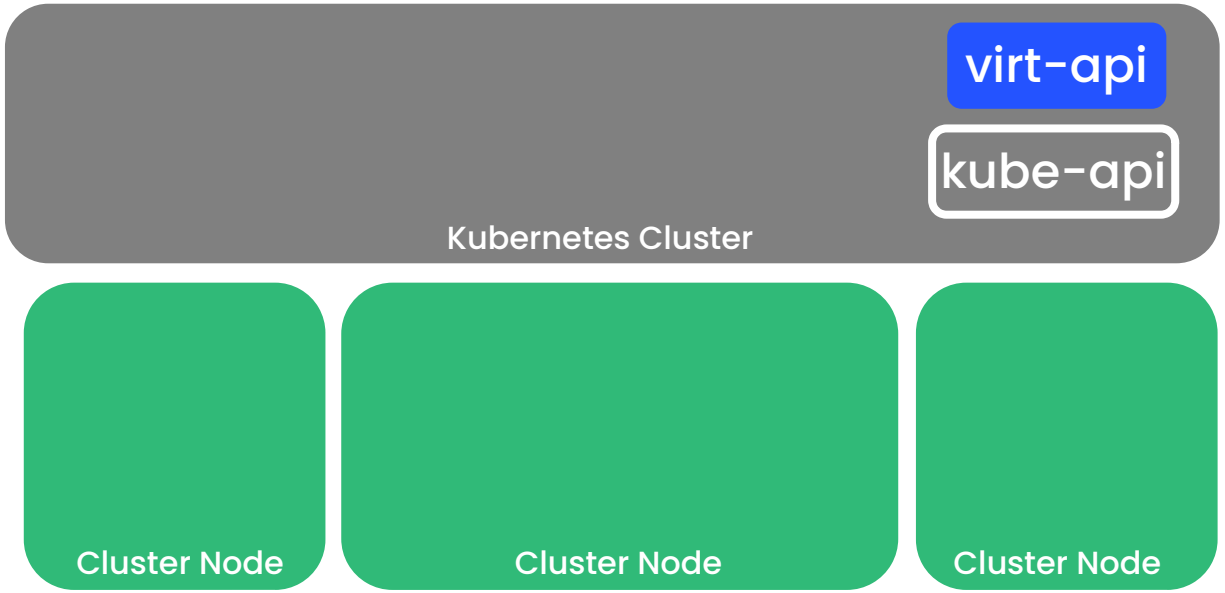


KubeVirt in Harvester

- KubeVirt extends Kubernetes by adding Custom Resource Definitions (CRDs) that allow Kubernetes to understand additional resource types.
- Harvester uses the following new virtualization related resource types:
  - VirtualMachine (VM)
  - VirtualMachineInstance (VMI)

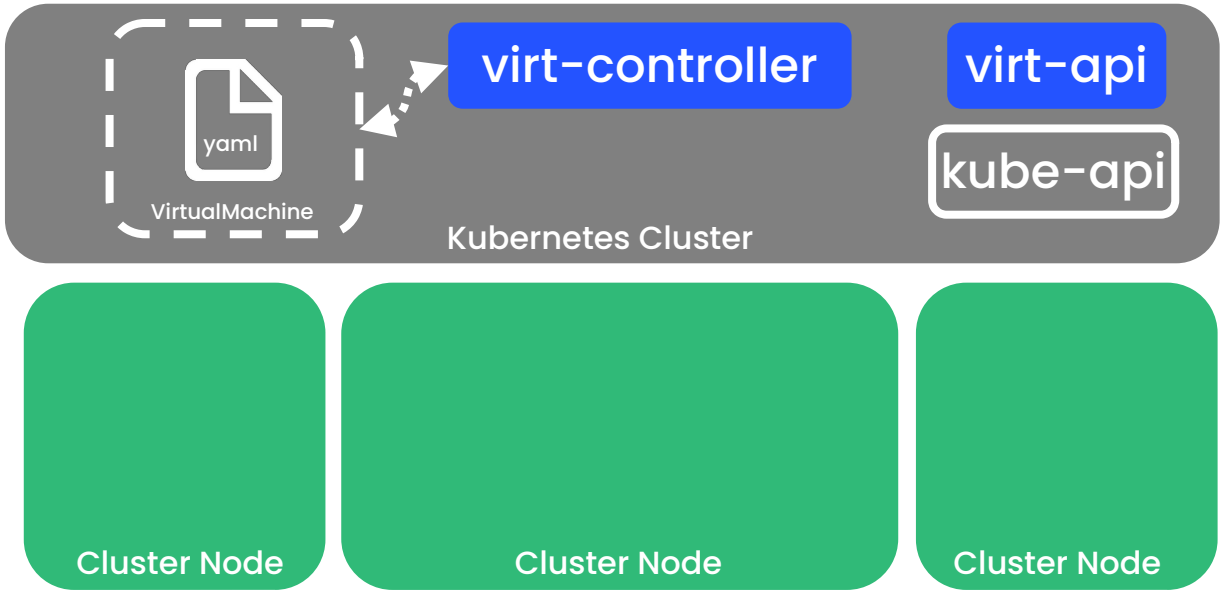
KubeVirt: virt-api

- A new API (**virt-api**) is defined that interfaces with the Kubernetes API server (**kube-api**), allowing management of these new virtualization resources types



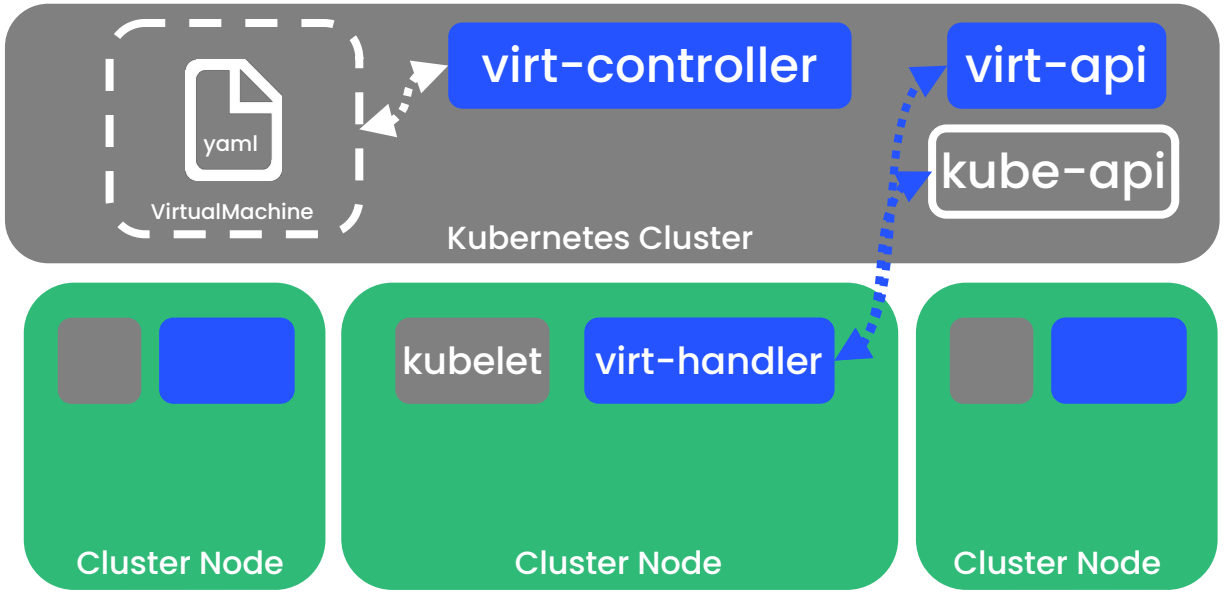
KubeVirt: virt-controller

- A controller (**virt-controller**) is deployed on the Kubernetes cluster.
- The **virt-controller** is responsible for taking a provided **VirtualMachine** spec and turning it into a Kubernetes object.



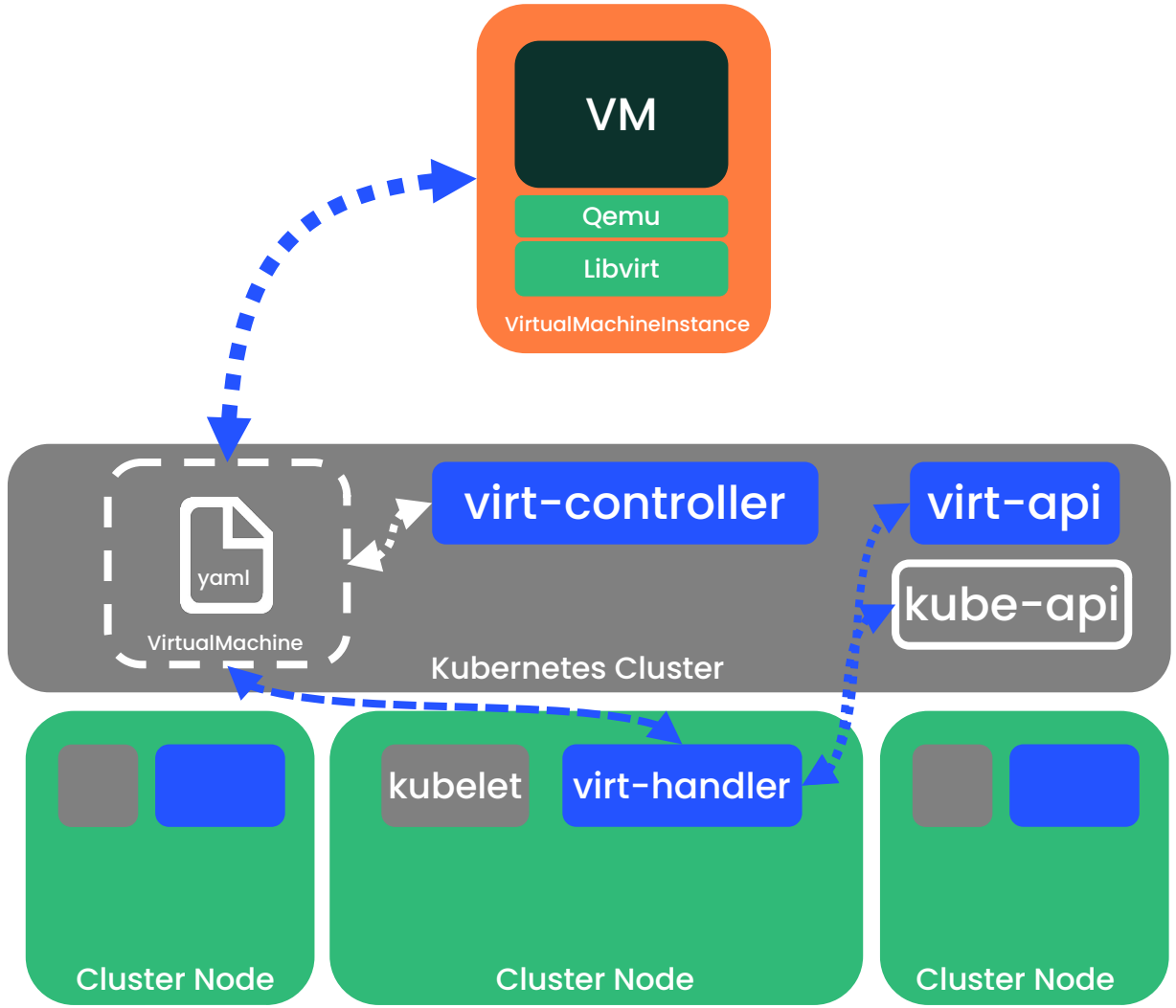
KubeVirt: virt-handler

- A daemon (**virt-handler**) runs on a node alongside the **kubelet**.
- The **virt-handler** communicates with the **virt-api** and **kube-api** services and is responsible for instantiating a **VirtualMachine** as a **VirtualMachineInstance** (i.e. create a pod that contains the unique pieces of the VM: VM config, disk images, etc.).



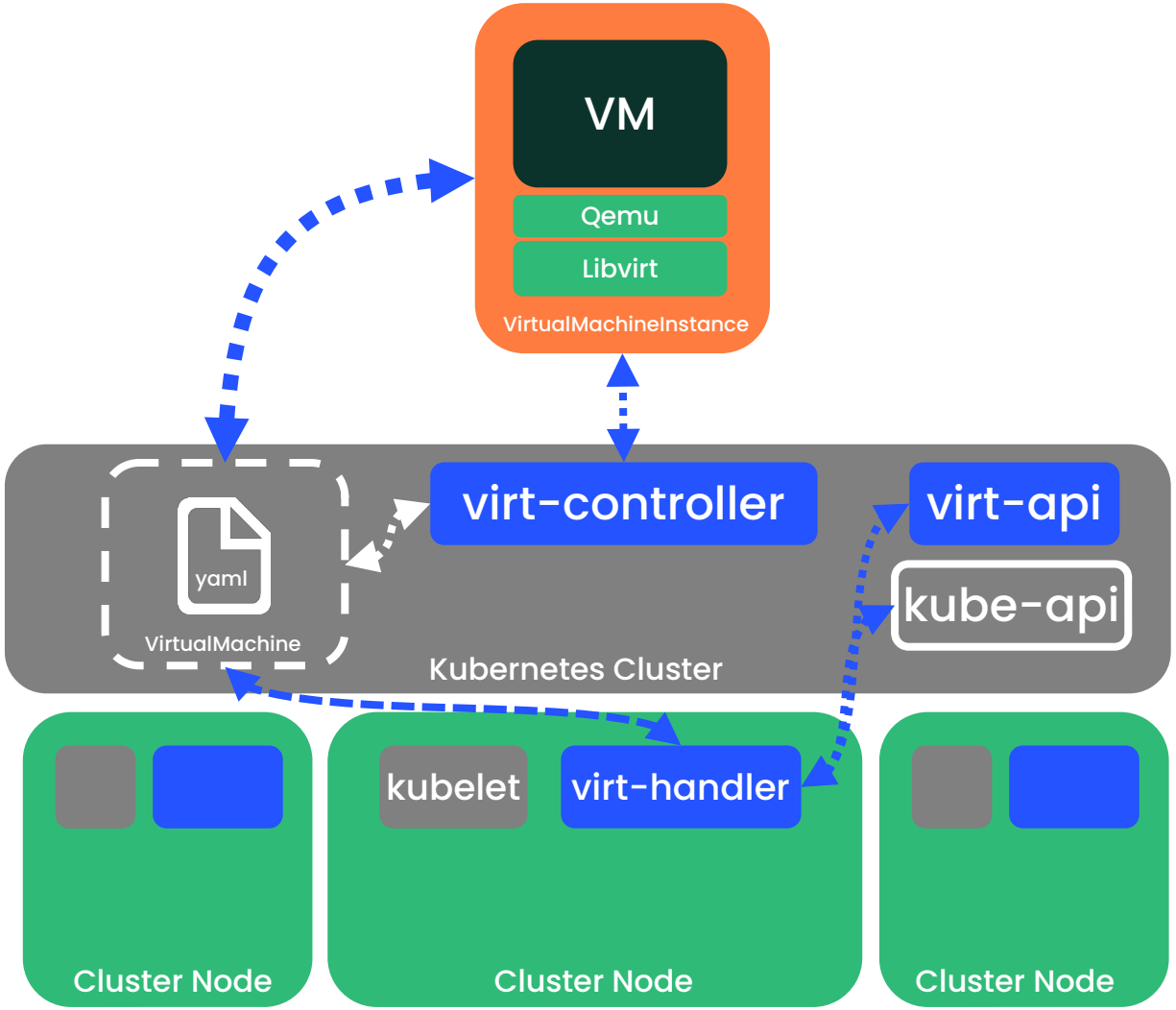
KubeVirt: VM Instantiation

- When a VirtualMachine is powered on, the **virt-handler** takes the VM's config (**VirtualMachine**) and hands it off to an instance of Libvirt running in a pod to define a Libvirt VM (**VirtualMachineInstance**).
- Libvirt then launches an instance of Qemu in the pod to create the VM based on the provided configuration.



VirtualMachine vs VirtualMachineInstance

- A **VirtualMachine** is a stateful controller (i.e. StatefulSet) as it can be stopped and started while retaining the VM data and state.
- A **VirtualMachineInstance** is an instantiation of a VM with a 1:1 relationship between the VM (controller) and the instance (VMI).



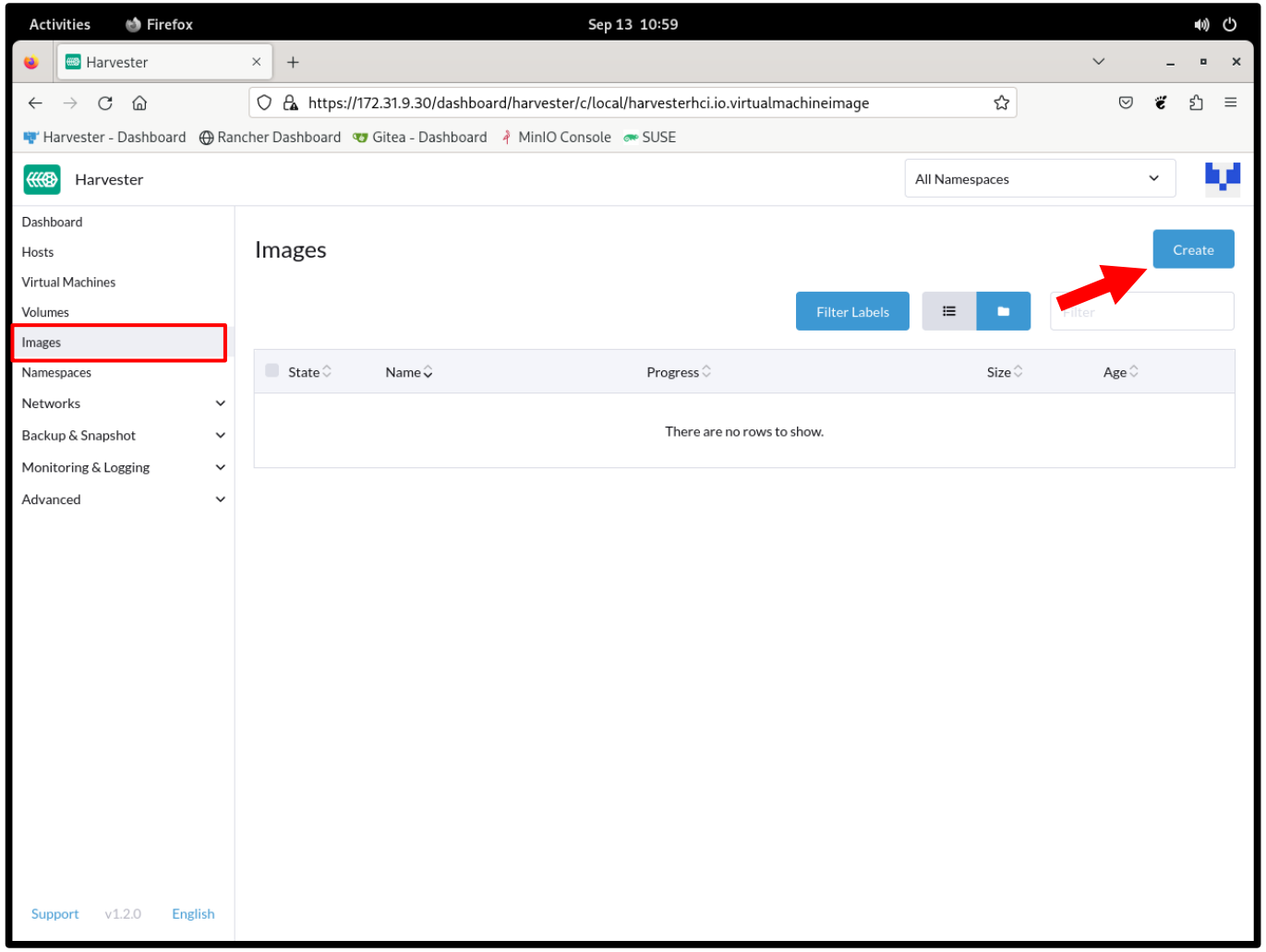
Virtual Machine Images

Virtual Machine Image Requirements

- Image files must be in one of the following formats:
  - qcow2, raw, ISO
- Images can also be created from volumes.
- The OS in the image must be configured to use cloud-init for initial runtime configuration.
- Images are uploaded and stored in the cluster (in Longhorn) for use.

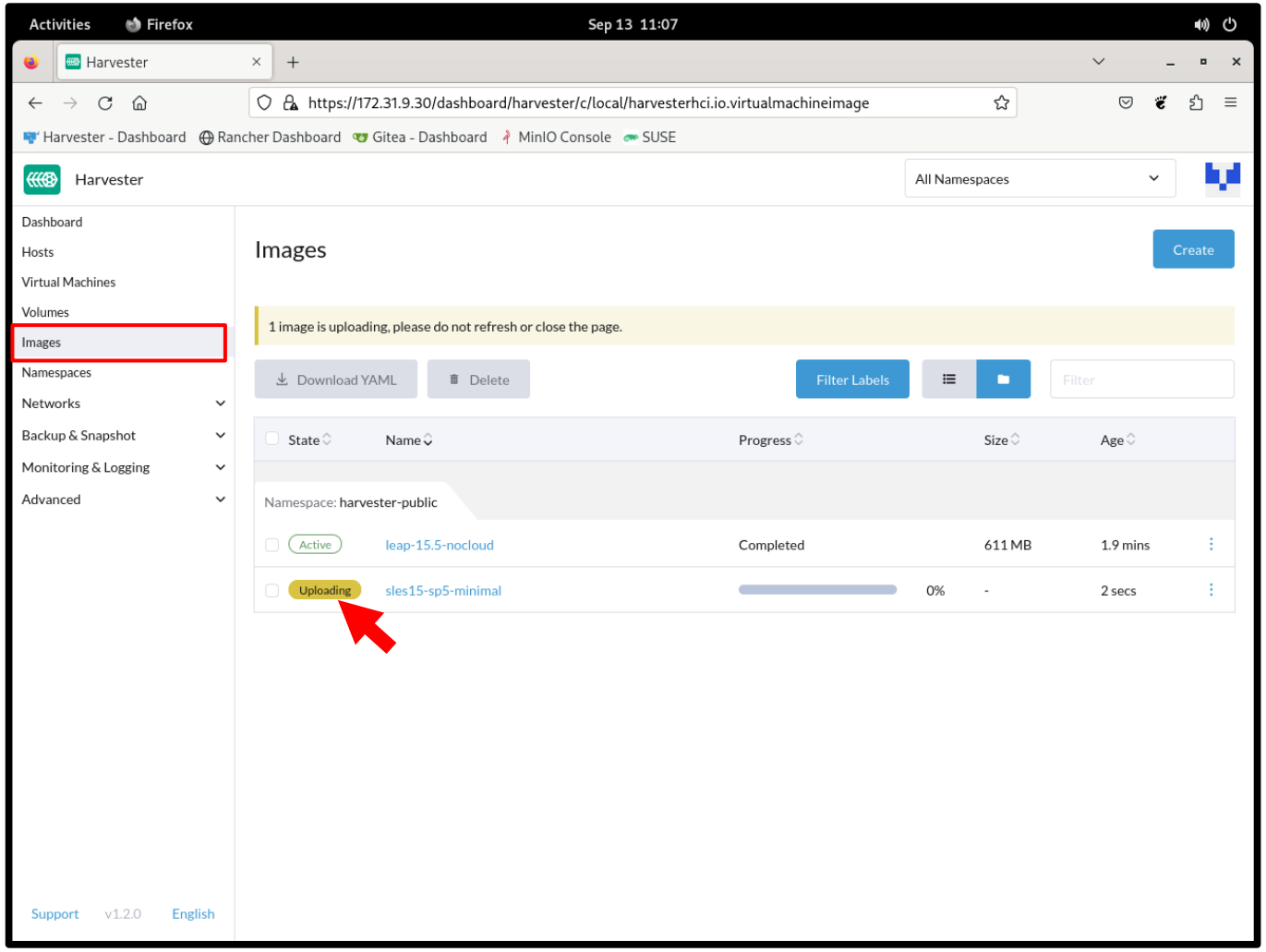
View/Create Images

- All available images are displayed on the **Images** screen.
- To add a new image to Harvester click: **Create**



Available Images and Status

- Images are displayed on the **Images** screen.
- The status will show as **Uploading** (when created from file) or **Downloading** (when created from a URL) while an image is being imported.



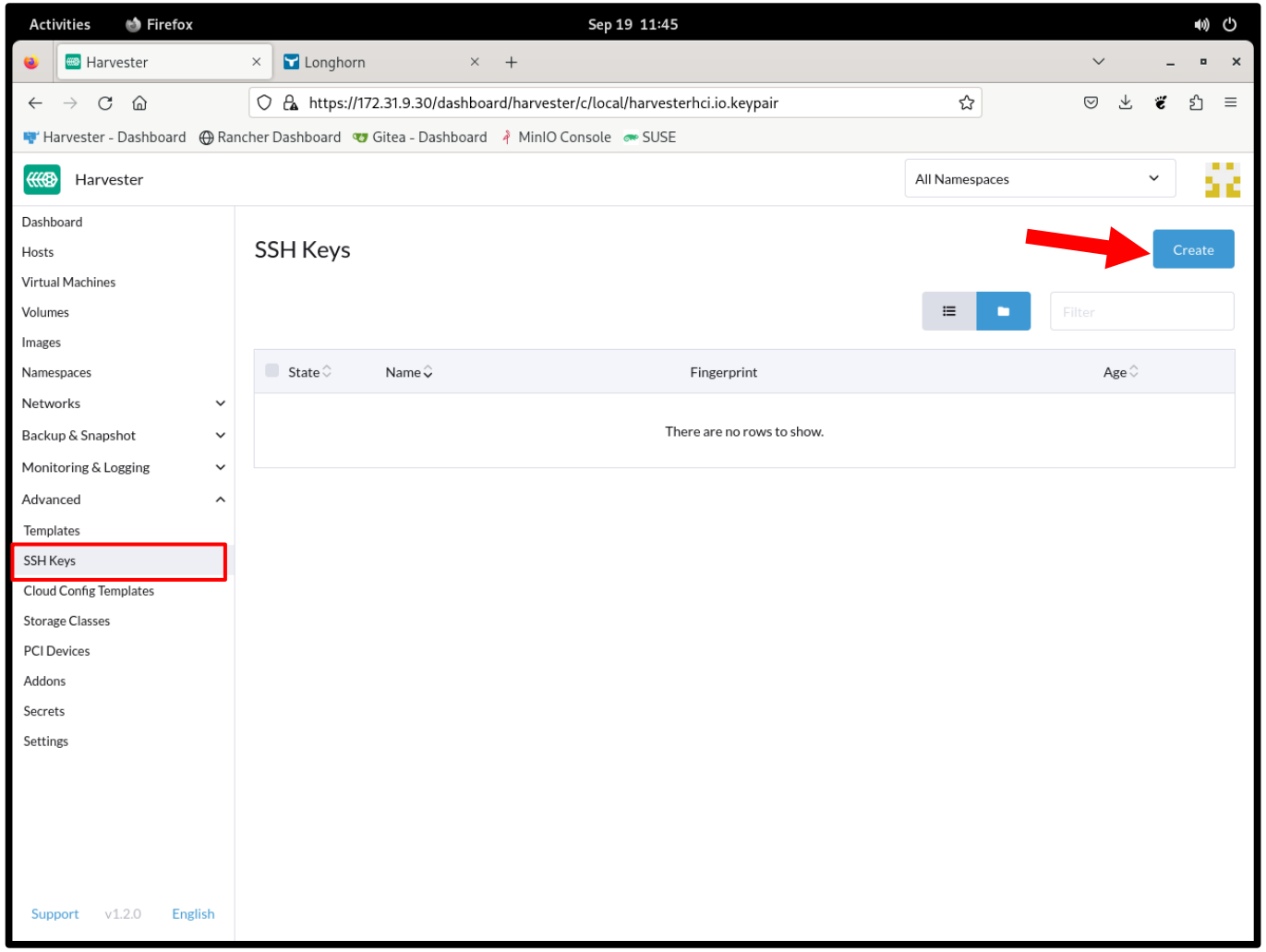
SSH Keys

SSH Keys

- SSH public keys can be stored in Harvester for use when launching VMs.
- When a VM is launched, a stored SSH key can be injected into the VM to allow secure access.

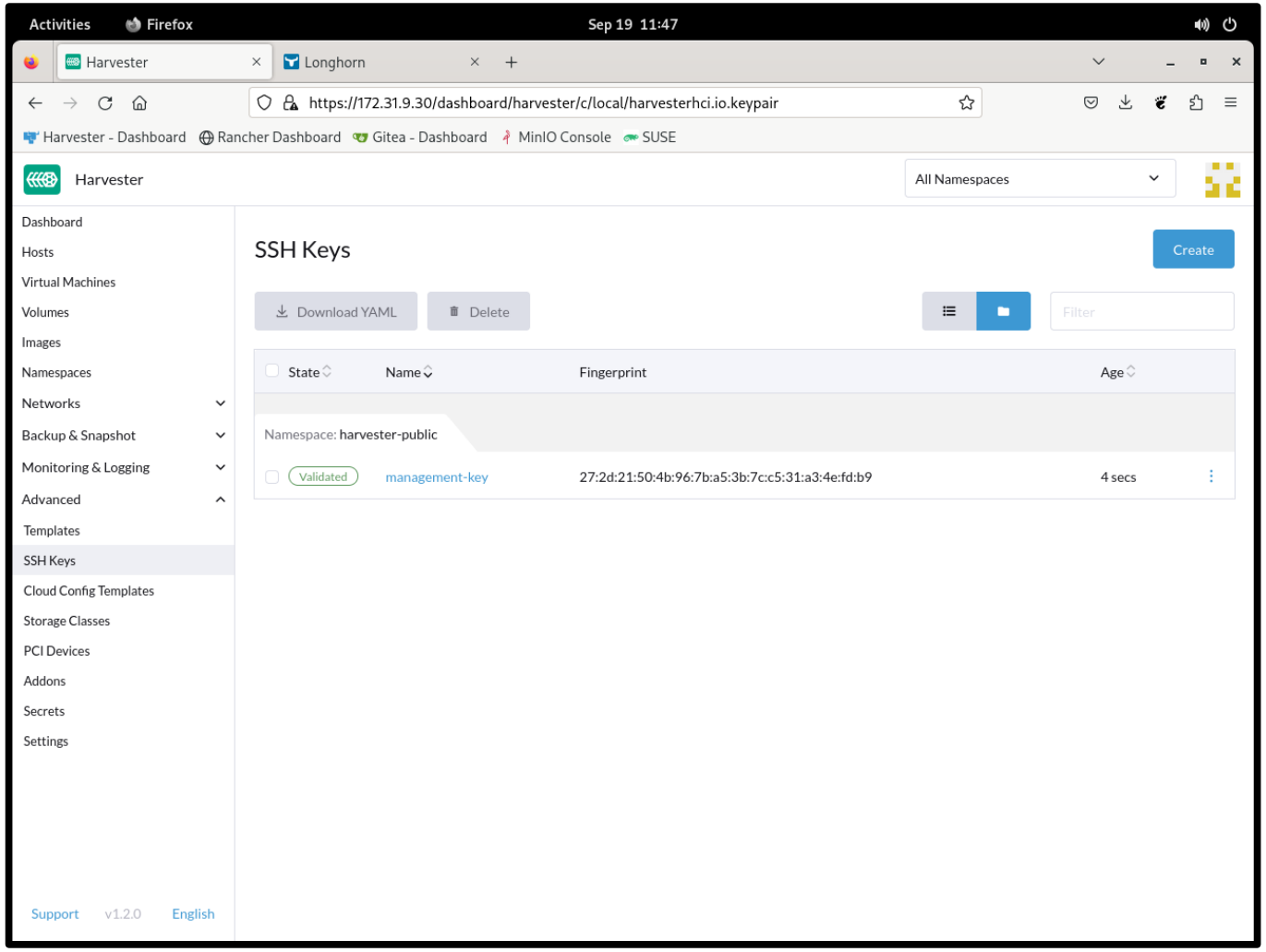
View and Create SSH Keys

- SSH keys are displayed and managed on the **Advanced → SSH Keys** screen.
- To add a new SSH key to Harvester click: **Create**



View Available Keys

- Keys that have been imported are displayed on the **SSH Keys** screen after they have been validated.



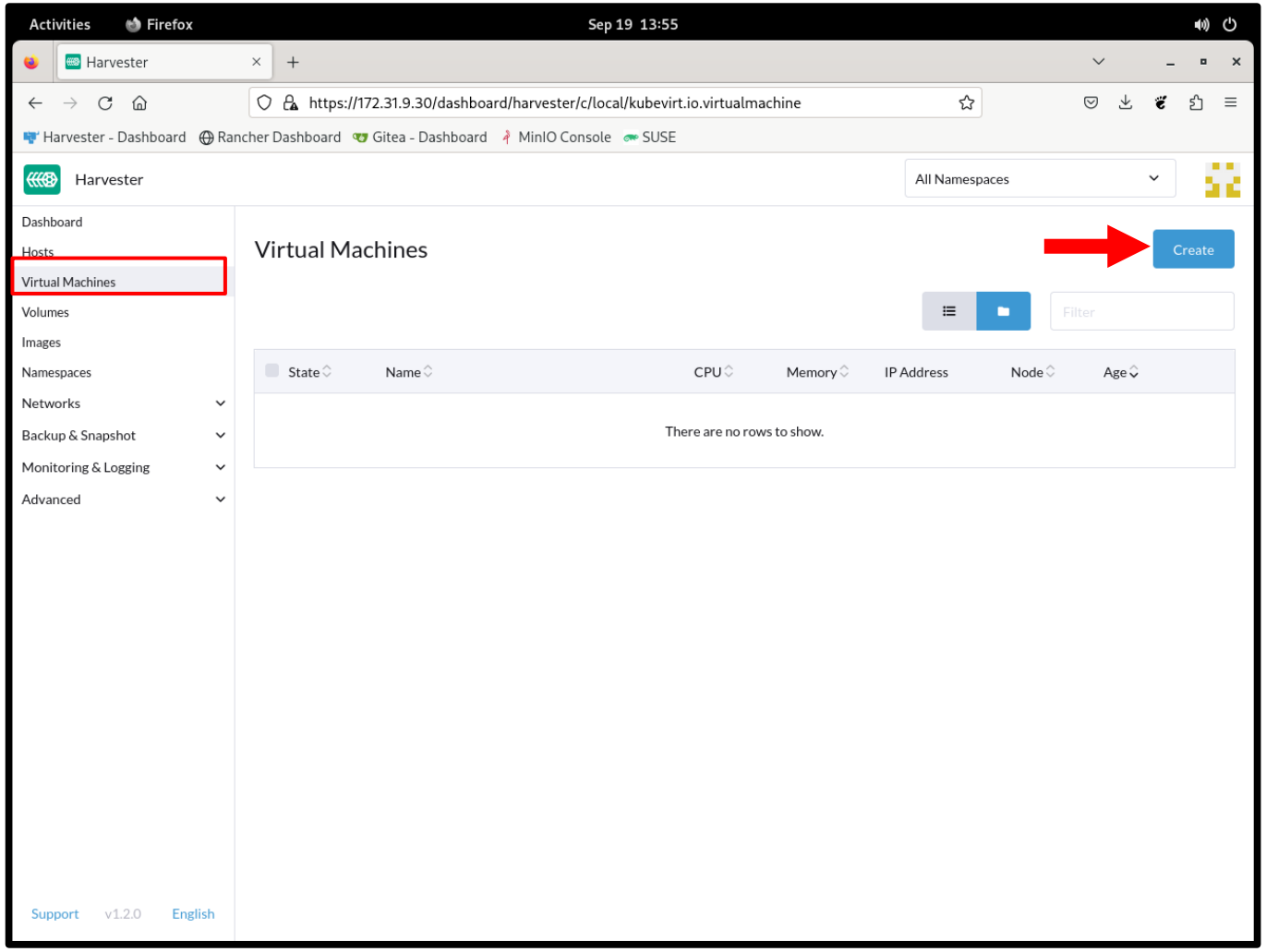
Virtual Machine Creation

Virtual Machine Creation Workflow Options

- VMs can be created using the Harvester Web UI.
- Configuration options are supplied to define what the VM should look like including CPUs, RAM, disks, etc.
- A single VM can be created from a set of configuration options.
- Multiple unique VMs can also be create simultaneously from a single set of configuration options.

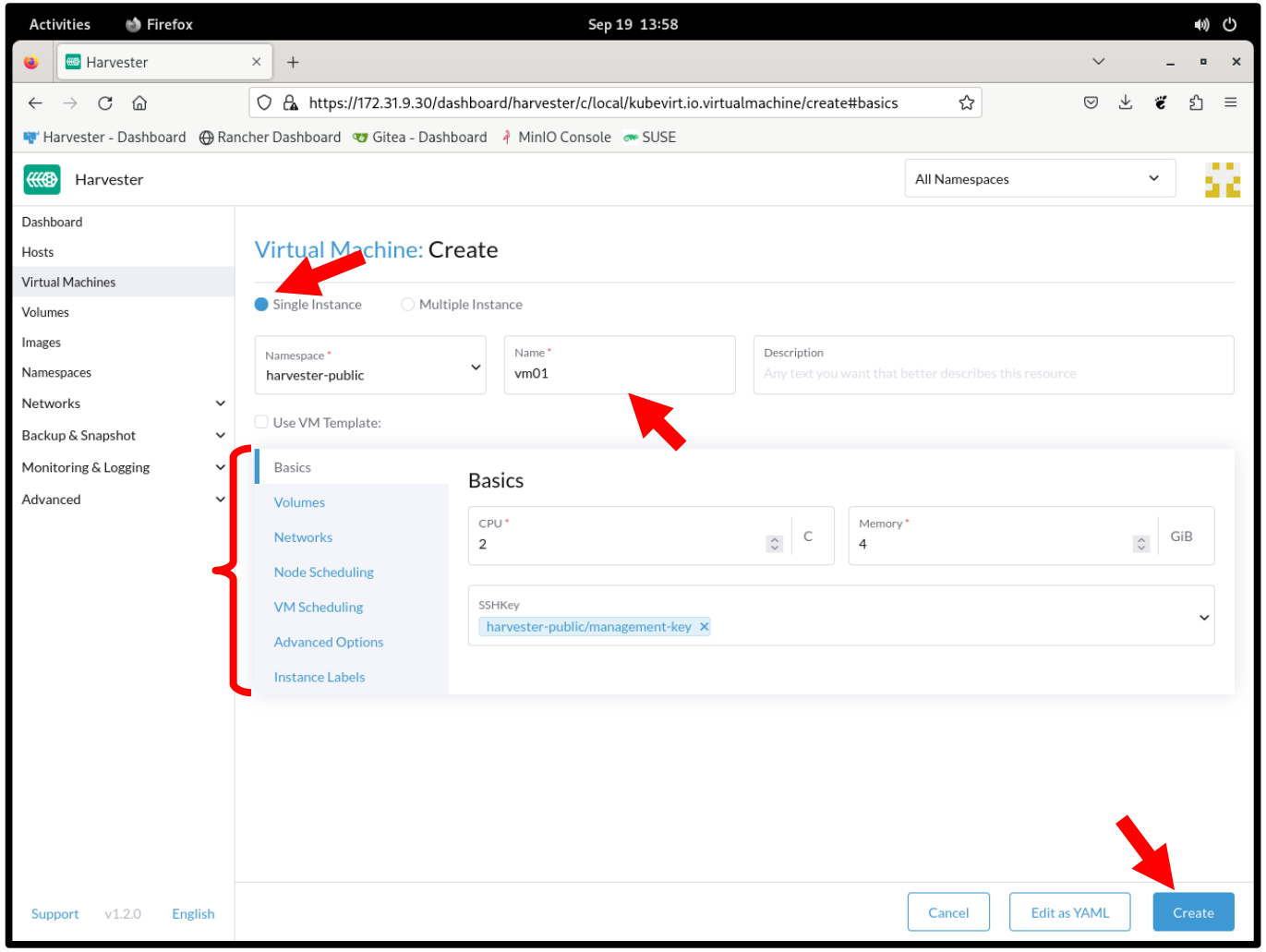
Create a VM

- VMs are displayed on the **Virtual Machines** screen.
- The current state of the VMs is displayed along with their resource utilization and the nodes on which they are running.
- The individual VMs can be managed from this screen as well.
- New VMs are created by clicking: **Create**



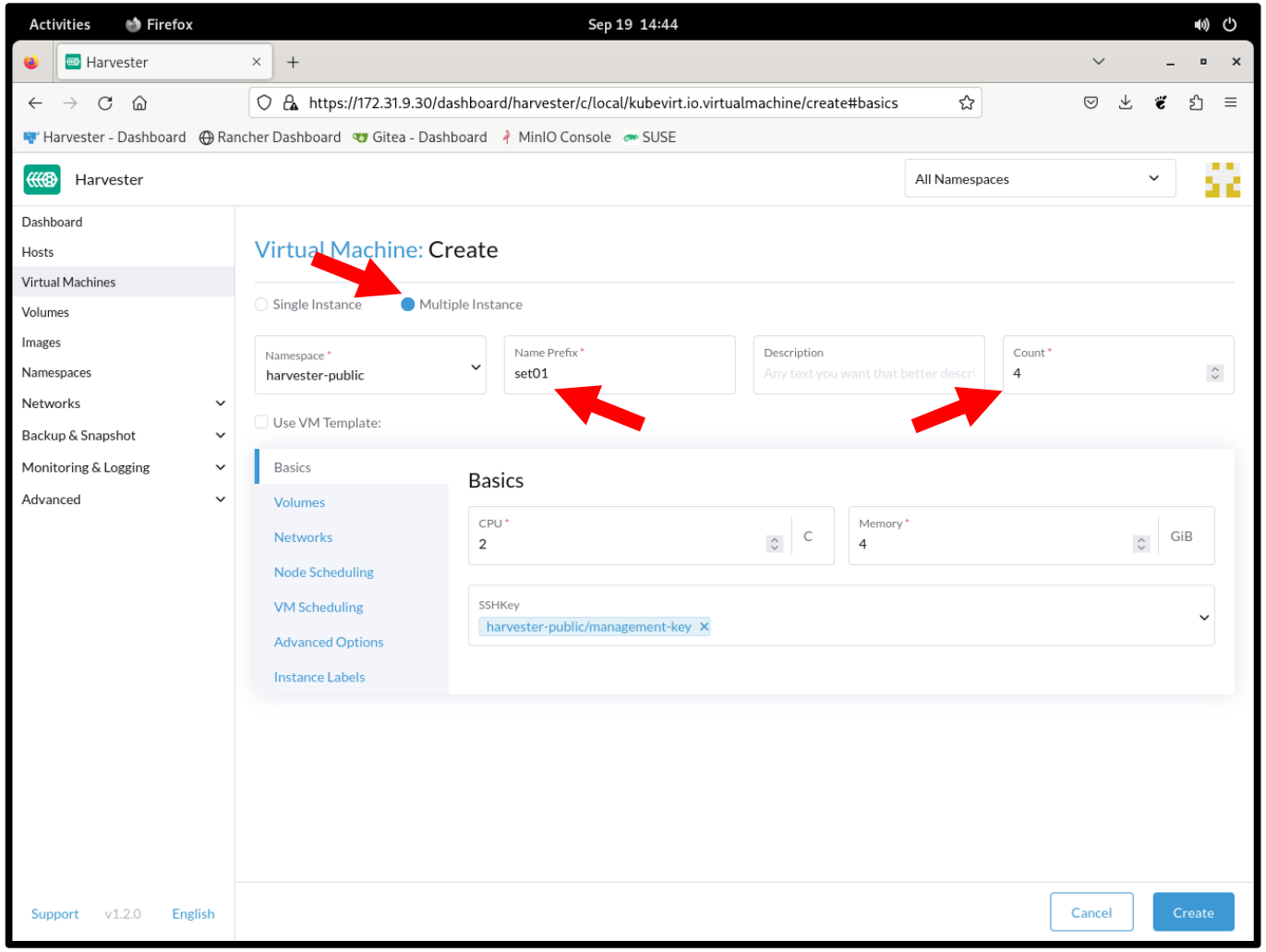
Virtual Machine: Create - Basics

- To create a single VM from the configuration options you provide, select: **Single Instance**
- Give the VM a **Name** .
- Continue through the tabs to enter/select all other desired configuration options.
- To finish creating the VM using the specified configuration options click: **Create**



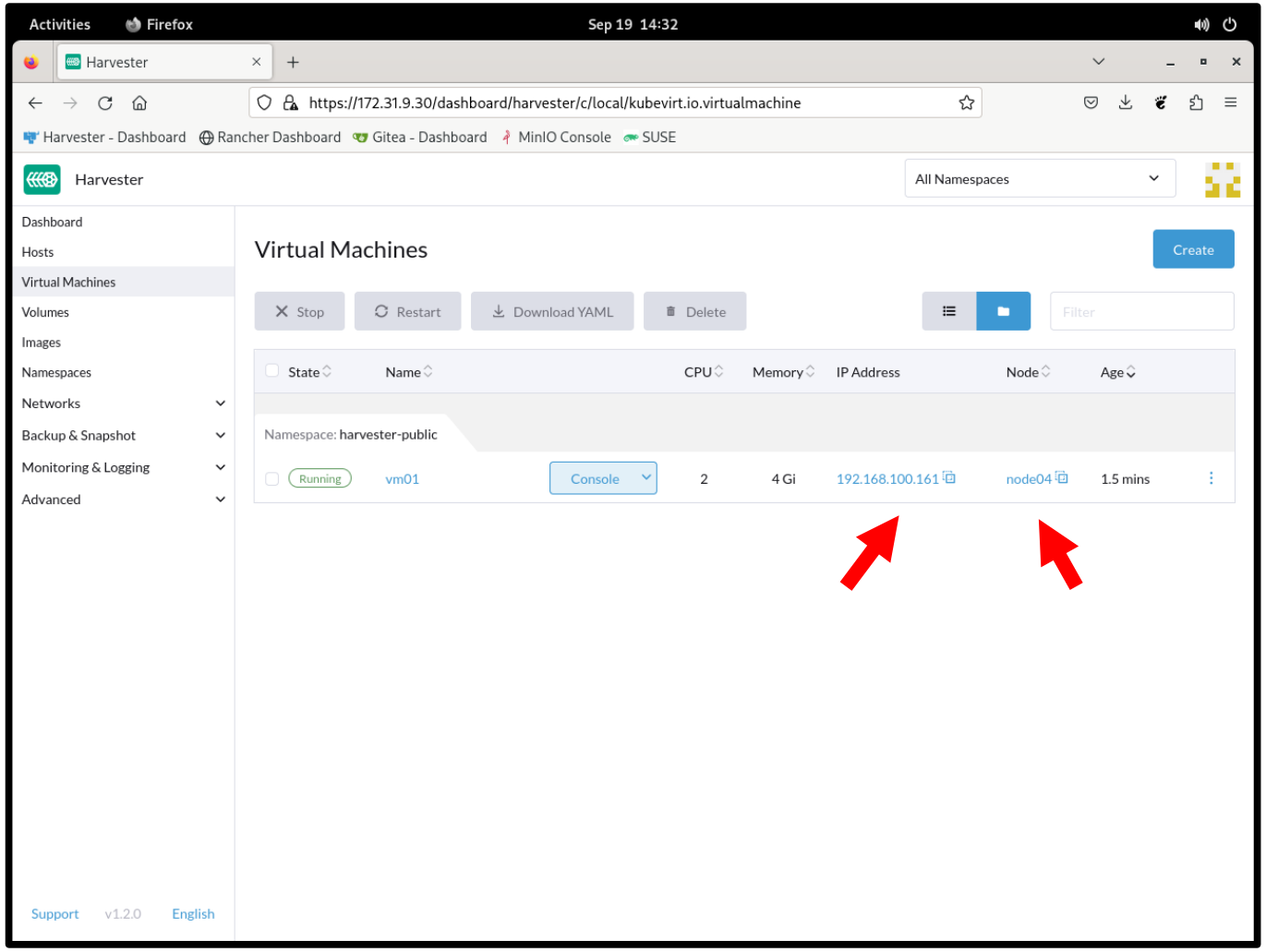
Virtual Machine: Create Multiple Instances

- To create multiple unique instances of a VM from a single set of configuration options select: **Multiple Instances**
- Enter a **Name Prefix**.
  - This is a text string to be prepended to the VM names when they are created.
- Enter the **Count**
  - To specify the number of instances to create.
- All other VM config is the same as when Single Instance is selected.
- Each VM will be individual with their own disk and copy of the config.



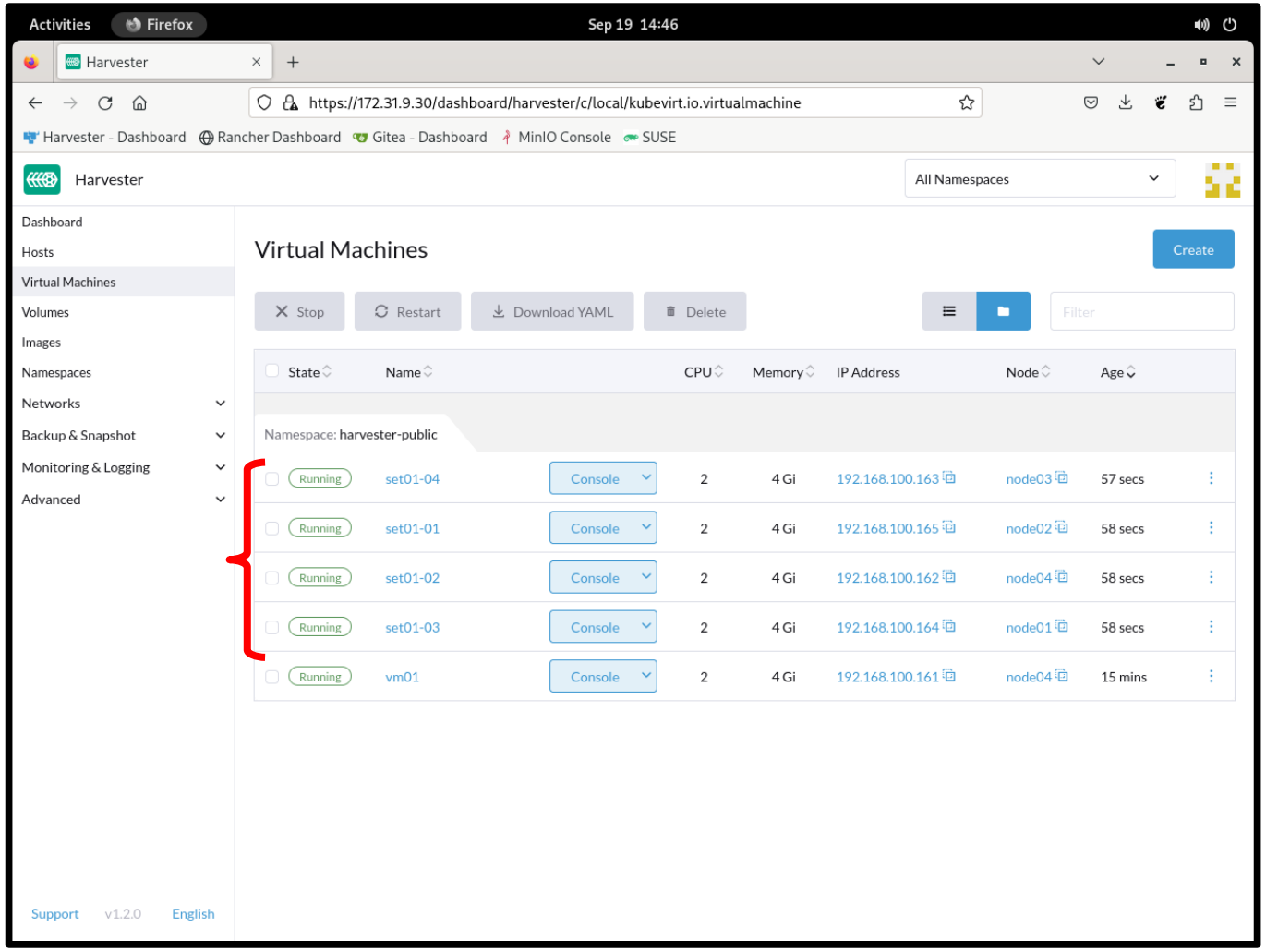
Virtual Machines – New VM Started

- When the VM is created it will be displayed on the **Virtual Machines** screen with its state, resource utilization and the host on which it is running.
- The IP address of the VM is probed and will also be displayed.



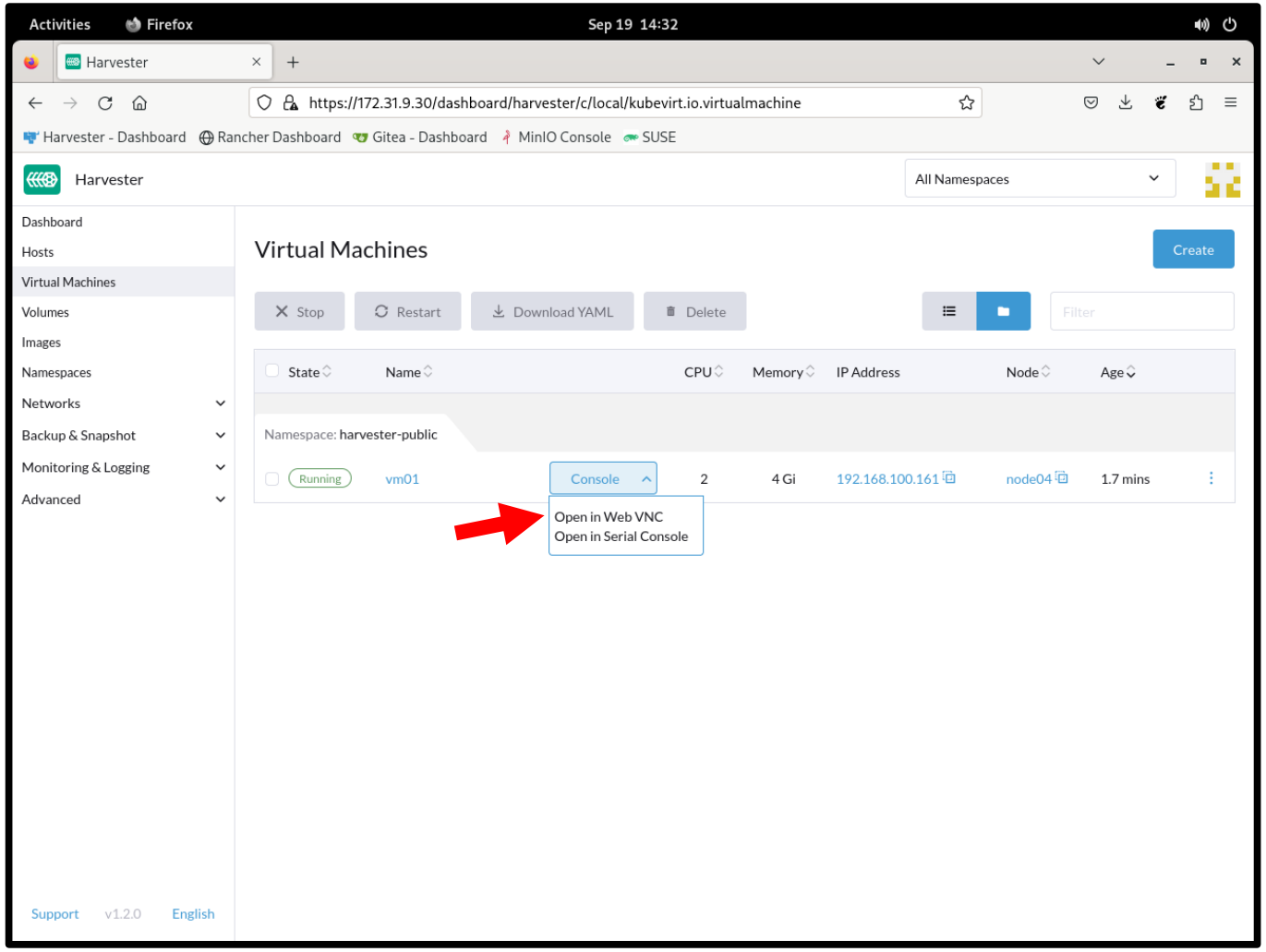
Virtual Machine: Create Multiple Instances

- When Multiple Instance VMs are created they will all be displayed on the Virtual Machine's screen.
- Their names will contain the **Name Prefix** specified during creation along with the instance number.
- Each VM can be managed independently.



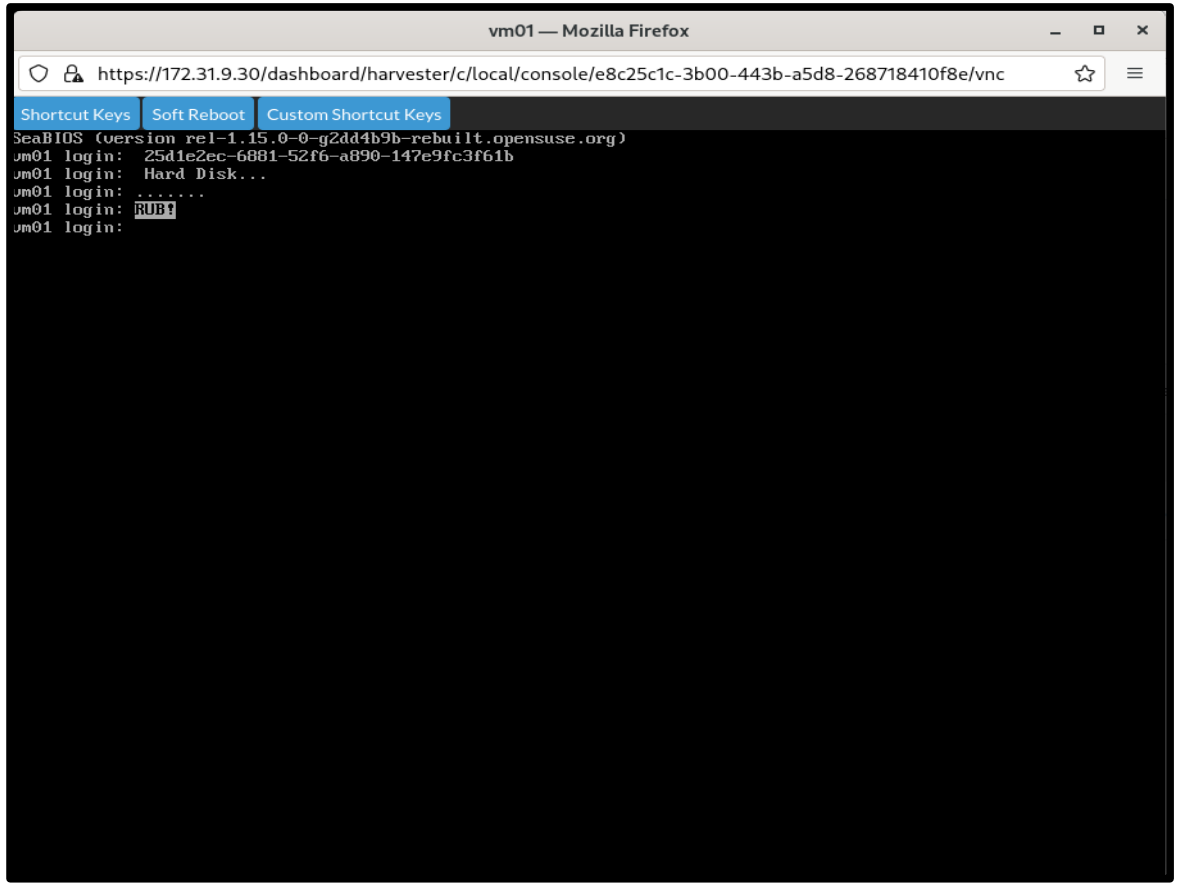
Virtual Machine – Access VM Console

- It is possible to connect to the console of running VMs.
- The options for console access are: **Web VNC** or **Serial Console**



Virtual Machine: VM Console Types

- A **Web VNC** console gives access to the virtual framebuffer of the VM, meaning you can interact with both virtual terminals and a GUI if it is available.
- A **Serial Console** gives access to a terminal only.



VM Cloning

Creating New VMs from Existing VMs

- If you need to create a new VM that is based on an existing VM you have a few options:
  - VM Cloning
  - Create an Image from a VM's disk
  - Restore a snapshot or backup as a new VM (covered in a later section)
- The method you choose depends on whether you want the new VM to start with existing VM configuration or just a disk image.

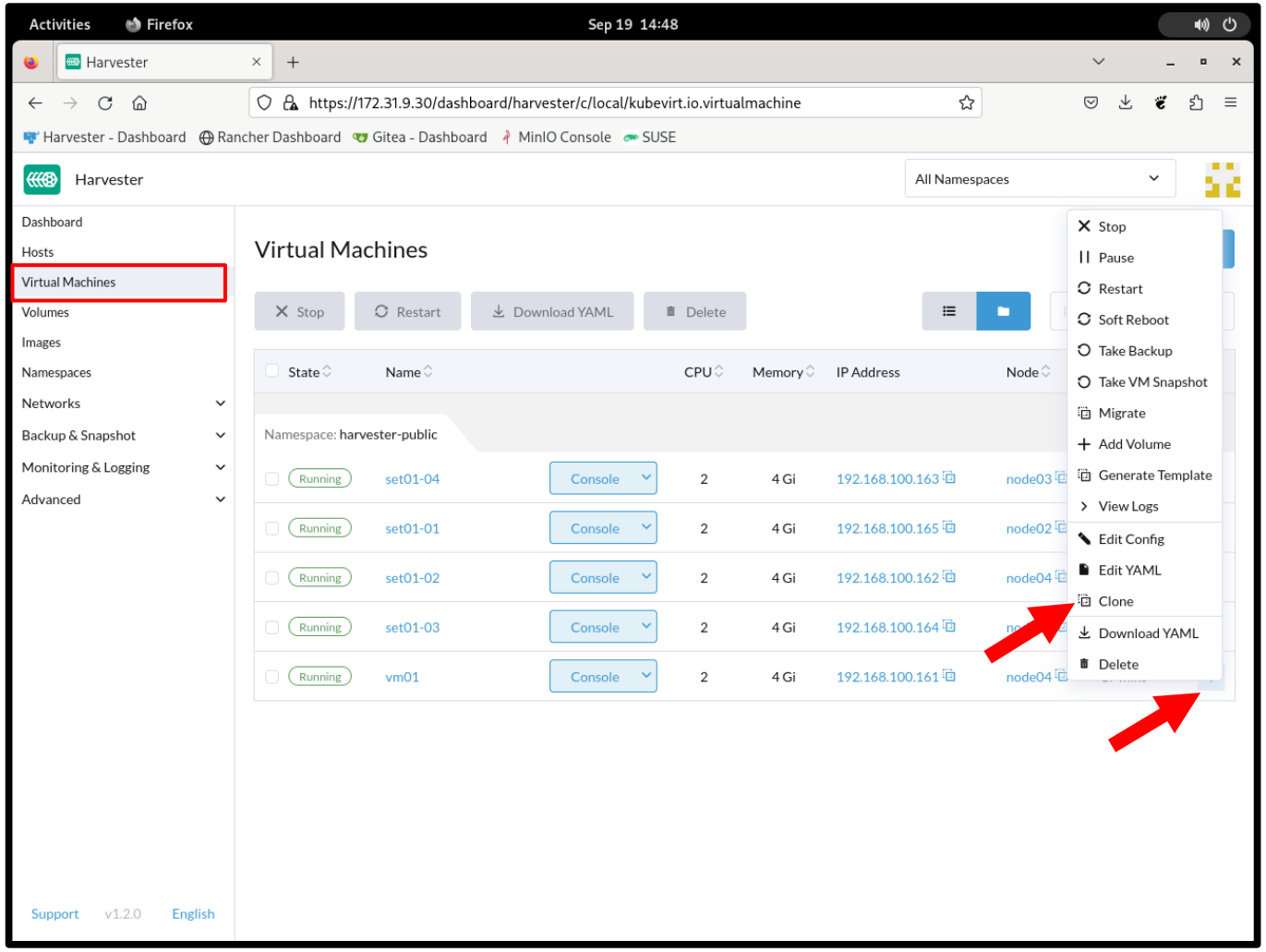
Existing Configuration as a Starting Point

- **Problem:**
  - You want a new VM that is like an existing VM and you want to start with existing configuration.
- **Solution:**
  - Cloning a VM is the Harvester solution.
- **Other Details:**
  - A cloned VM is a new VirtualMachine object that is based on an existing VirtualMachine object.
  - A cloned VM will get an exact copy of the source VM's disk.
  - You will initially be presented with a copy of the source VM's configuration which you can modify if desired before the clone is created.

Virtual Machine: Clone a VM

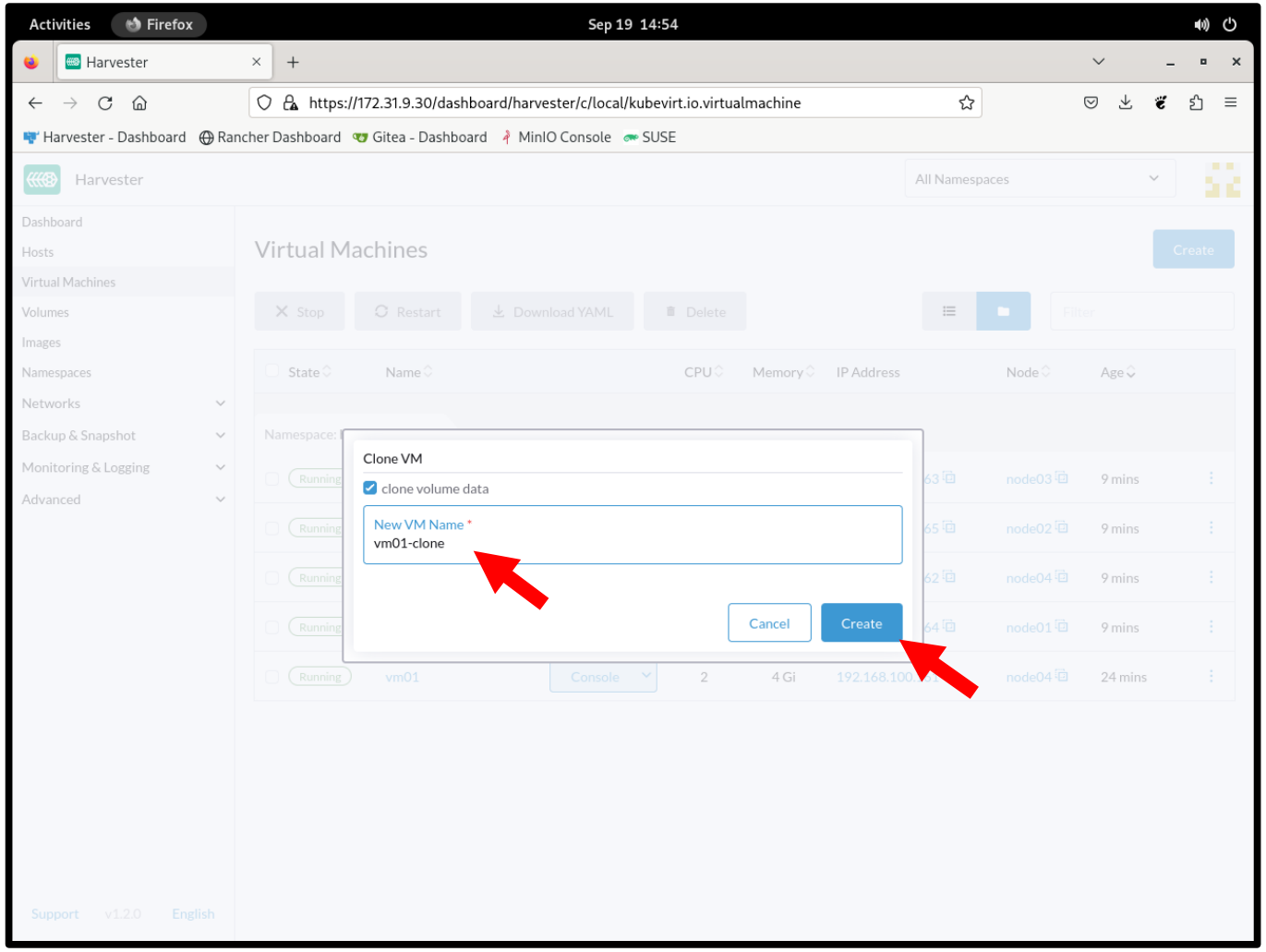
- On the **Virtual Machines** screen, locate the VM you wish to clone.

  (Though not required, powering it off first can give you a disk with a more stable state)
- From the **3 dot** menu on the right of the VM select: **Clone**



Virtual Machine: Clone a VM

- Enter the **Name** for the VM.
- Click: **Create**
- You will then be presented with the **Virtual Machine: Create** dialog that will be pre-populated with the source VM's configuration options.

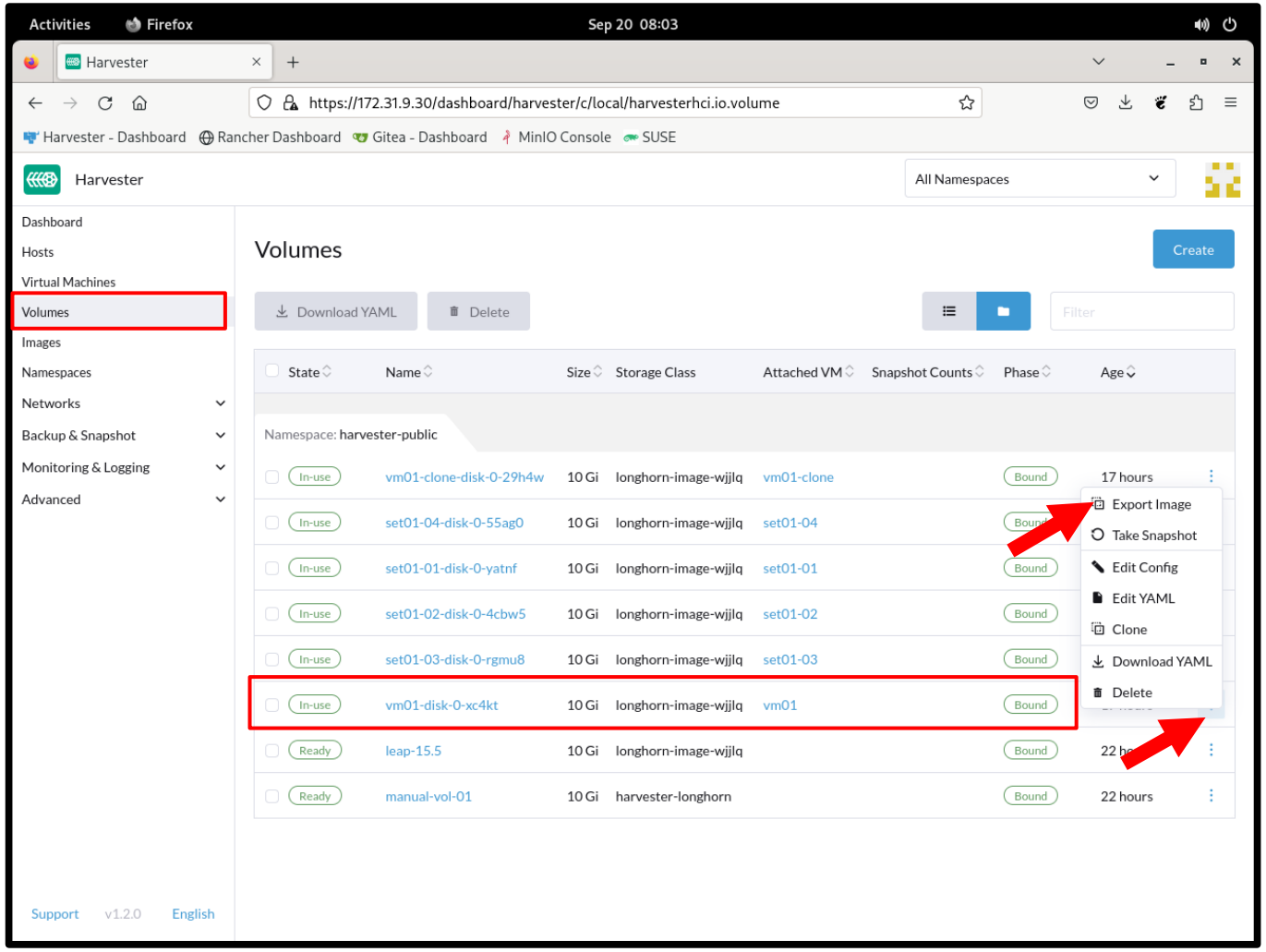


Only Disk Image as a Starting Point

- **Problem:**
  - You want the ability to create VMs in the future based on an existing VM but not start from existing configuration.
- **Solution:**
  - Creating an Image from an existing VM's rootdisk is Harvesters solution.
- **Other Details:**
  - Images created from an existing VM's rootdisk will be identical to the source VM.
  - Images allow you to create a new "clone" of a VM after the source VM has been deleted.
  - Images allow you to preserve customization made to a live VM's disk for use in the future.
  - Creating a new VM from an Image created from another VM requires specifying new VM configuration from scratch.

Create an Image from a Volume

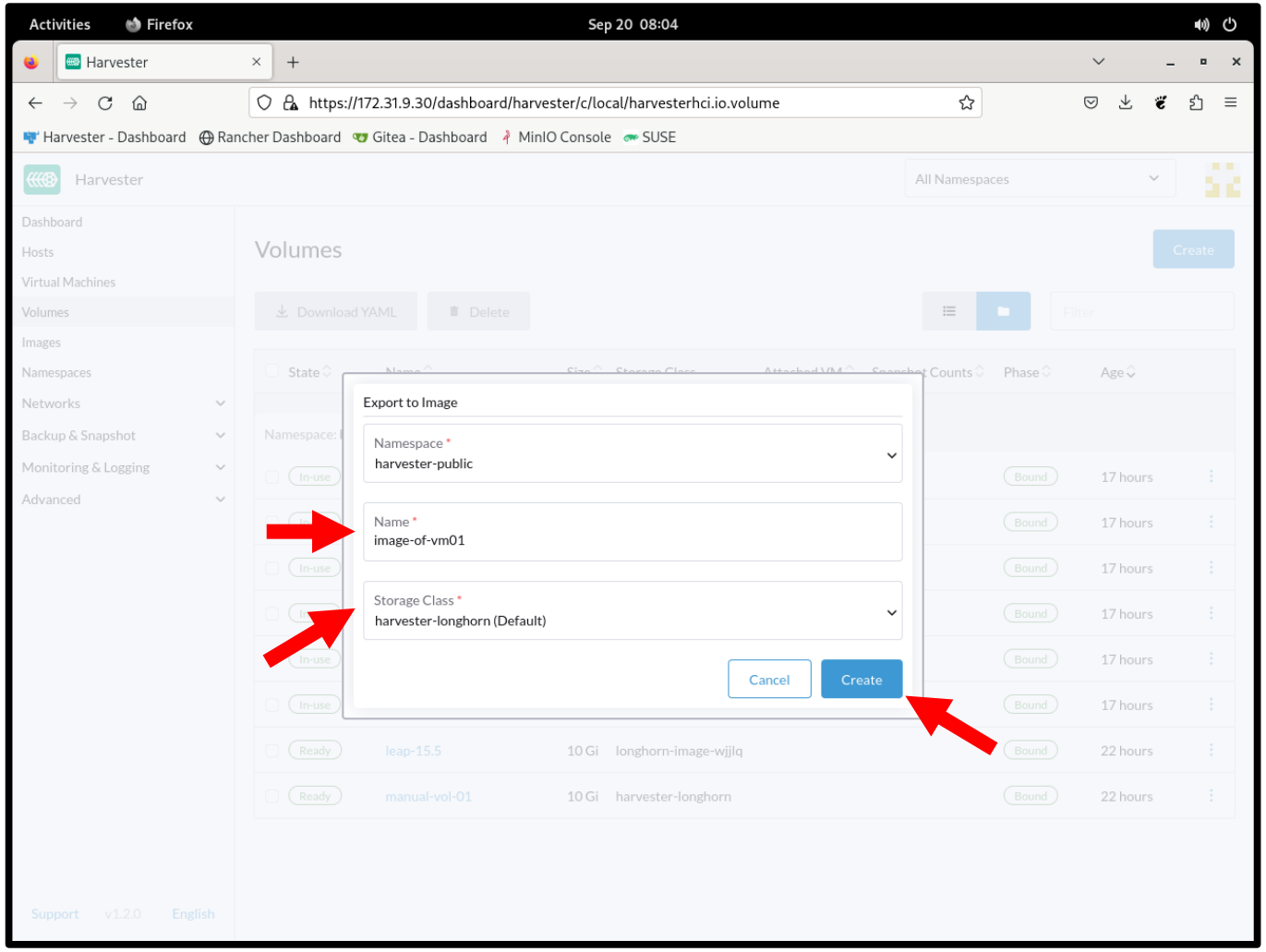
- On the **Volumes** screen, locate the volume use as the root disk of the VM you want to use as an image.
- From the **3 dot** menu on the right of the volume select **Export Image**.



Create an Image from a Volume

- Give the image a **Name**.
- Select the **Storage Class** in which to save the image.
- Click: **Create**

Creating a VM from this image will be the same as manually creating a new VM from scratch where you specify all VM configuration options and select this image as its root disk.



Cloud Config Templates

Cloud-init Primer

- Cloud-init is a service that runs early in the Linux boot process.
- Cloud-init uses modules to extend the functions it is able to perform.
- Cloud-init operates in 5 different stages of the boot process:
  - **Generator**: Determines if cloud-init.target should be included in the boot goals.
  - **Local**: Locate local data sources and apply network config.
  - **Network**: Apply user-data located locally and from network sources (disk prep, user creation, etc.).
  - **Config**: Additional config that doesn't need early execution is run here.
  - **Final**: Run config as late as possible (i.e. anything that would run after logging in).

Cloud Config

- Main config file: `/etc/cloud/cloud.cfg`
  - Can contain specific directives as well as define which modules to run and when to run them.
  - Is modular in nature with a default include directory: `/etc/cloud/cloud.cfg.d/`
- Additional cloud-init config can be retrieved from a number of different sources, both local and remote.
- All cloud config that is retrieved is aggregated and processed.

Harvester Cloud Config Templates

Two types of templates can be defined in Harvester:

- Network Data
  - Contains network specific configuration
- User Data
  - Contains all other configuration

Cloud Config: Network Data – 1st NIC with DHCP

- Configure the physical network interface named eth0 to use DHCP.

```
version: 1
config:
  - type: physical
    name: eth0
    subnets:
      - type: dhcp
```

Cloud Config: Network Data – 1st & 2nd NIC with DHCP

- Configure the physical network interface named eth0 to use DHCP.
- Configure the physical network interface named eth1 to use DHCP.

```
version: 1
config:
  - type: physical
    name: eth0
    subnets:
      - type: dhcp
  - type: physical
    name: eth1
    subnets:
      - type: dhcp
```

Cloud Config: User Data – Default User Password

- Set a password for the default user.
- Set the password to not expire.
- Enable SSH password authentication.

```
password: password
chpasswd:
  expire: false
ssh_pwauth: true
```

Cloud Config: User Data – Create User with Password

- Create a user named **tux** with a primary group of **users**.
- Set a password for the **tux** user.
- Enable **tux** to use sudo without requiring a password.

```
users:
 - name: tux
   shell: /bin/bash
   groups: users
   lock_passwd: false
   passwd:
$6$SalTsaLt$47wJgOF2AZFKYiSxDKM8beRyIM5JYg28lg
Eiry25V5FbC4nDuk9OHpxC0XtsJkY1OYZNYHWMnTLhXsgP
wL5SS.
   sudo: ALL=(ALL) NOPASSWD:ALL
```

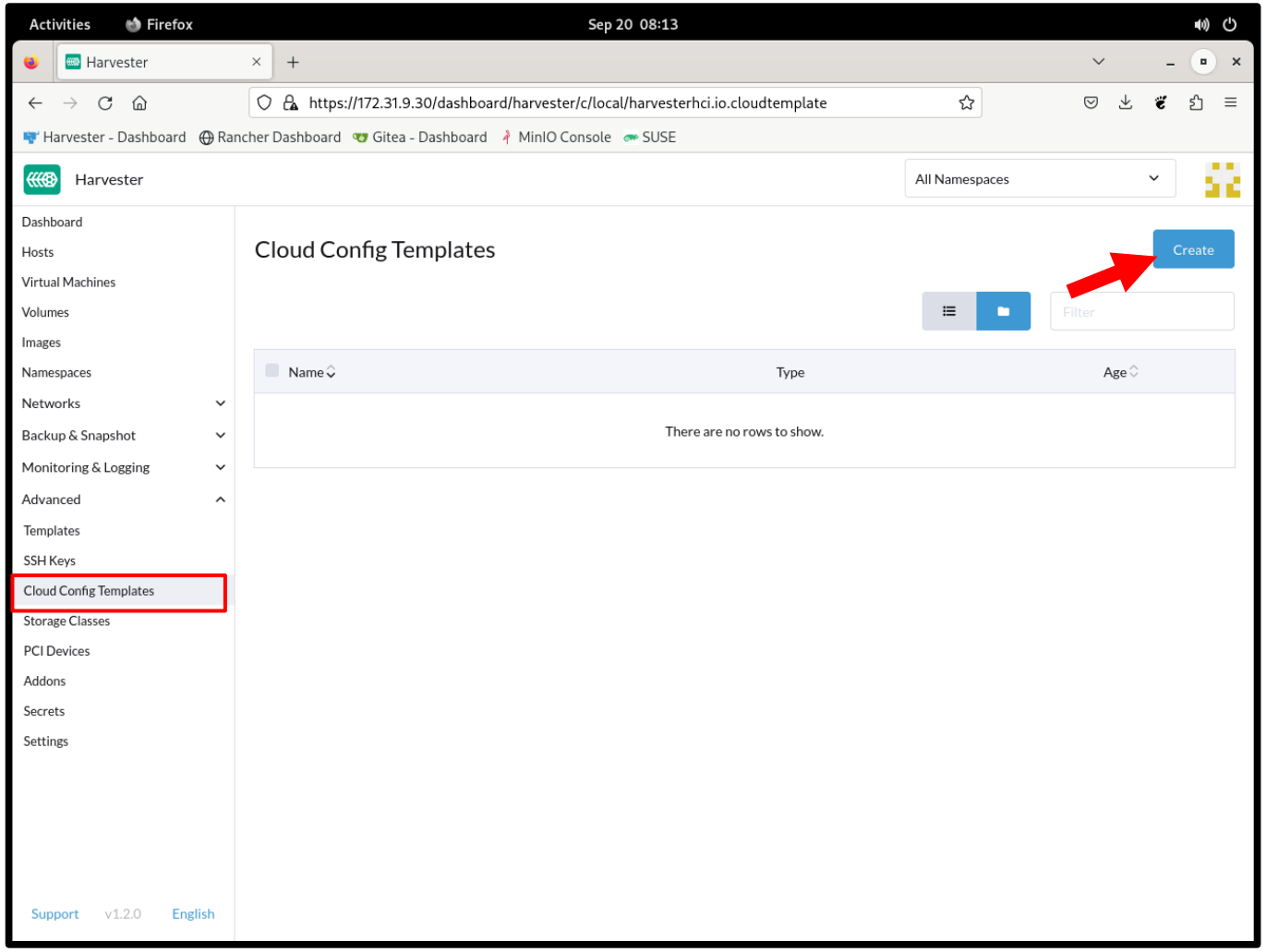
Cloud Config: User Data – Create User with SSH Key

- Create a user named **tux** with a primary group of **users**.
- Disable the password for the **tux** user.
- Enable **tux** to use sudo without requiring a password.
- Add two SSH keys to the **tux** users authorized_keys file.

```
users:
 - name: tux
   shell: /bin/bash
   groups: users
   ssh_import_id: None
   lock_passwd: true
   sudo: ALL=(ALL) NOPASSWD:ALL
   ssh_authorized_keys:
     - <SSH_PUBLIC_KEY_1>
     - <SSH_PUBLIC_KEY_2>
```

Cloud Config Templates

- The **Advanced → Cloud Config Templates** screen displays the current list of cloud config templates.
- To create a new template click **Create**.



VM Templates

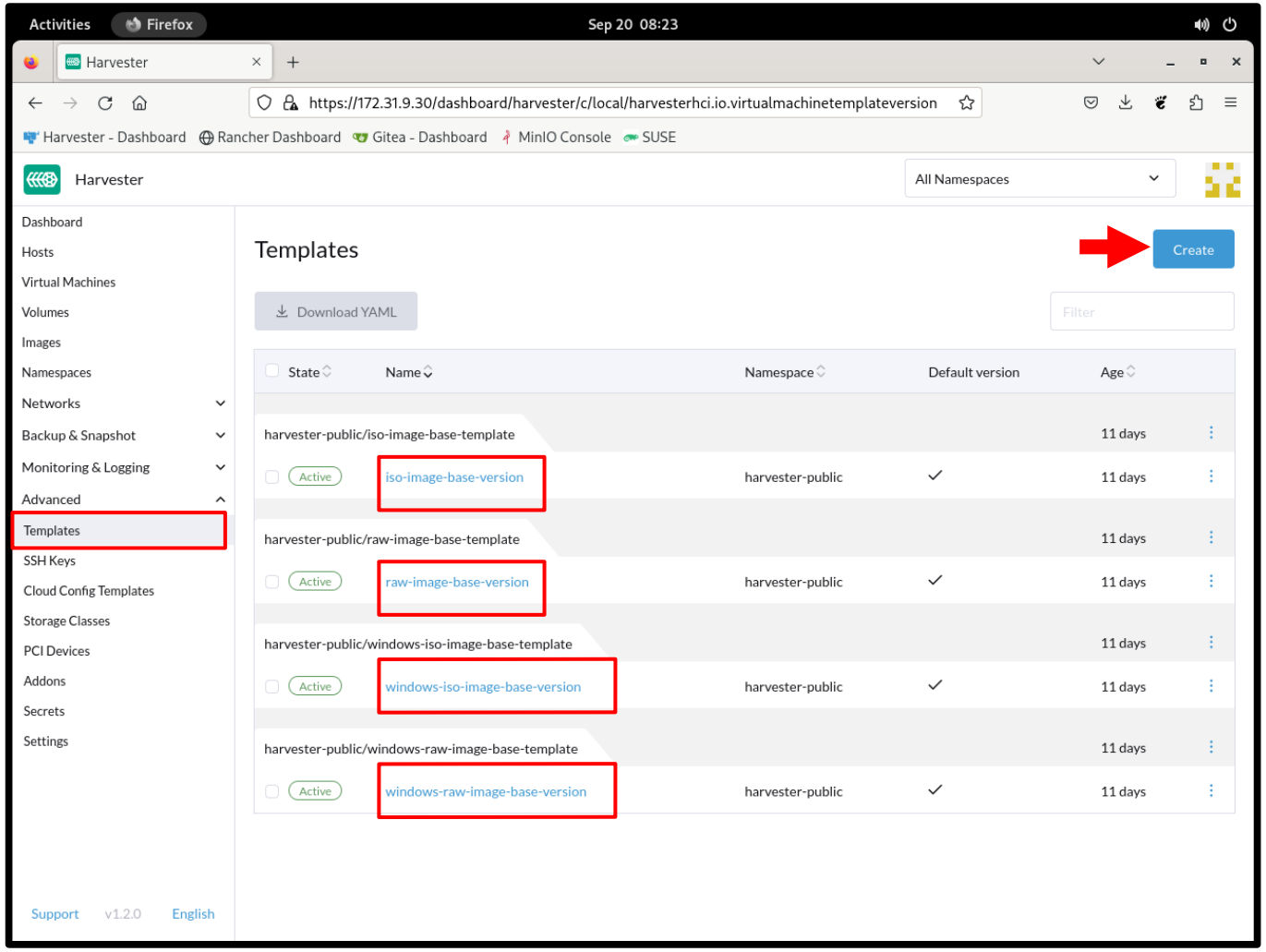
VM Templates

- Templates allow for the creation of predefined VM configuration that can be used to create new VMs.

- Templates allow for specifying configuration for all aspects of the VM's definition the same as when you create a new VM.

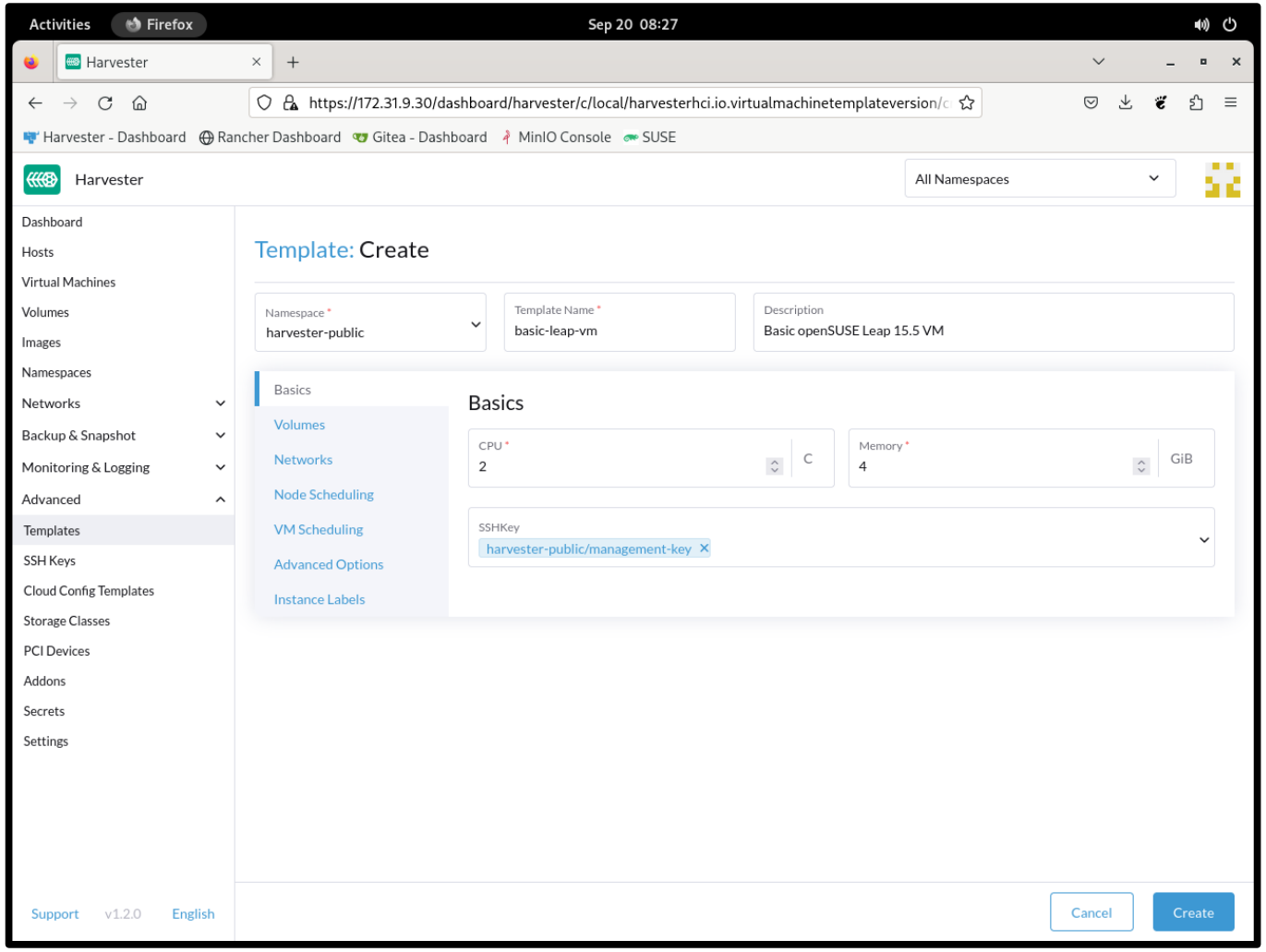
VM Templates

- The **Advanced → Templates** screen displays the current list of templates.
- Harvester ships with preconfigured templates to install VMs from an ISO, a simple raw image based VM and Windows VMs.
- To create a new template click: **Create**



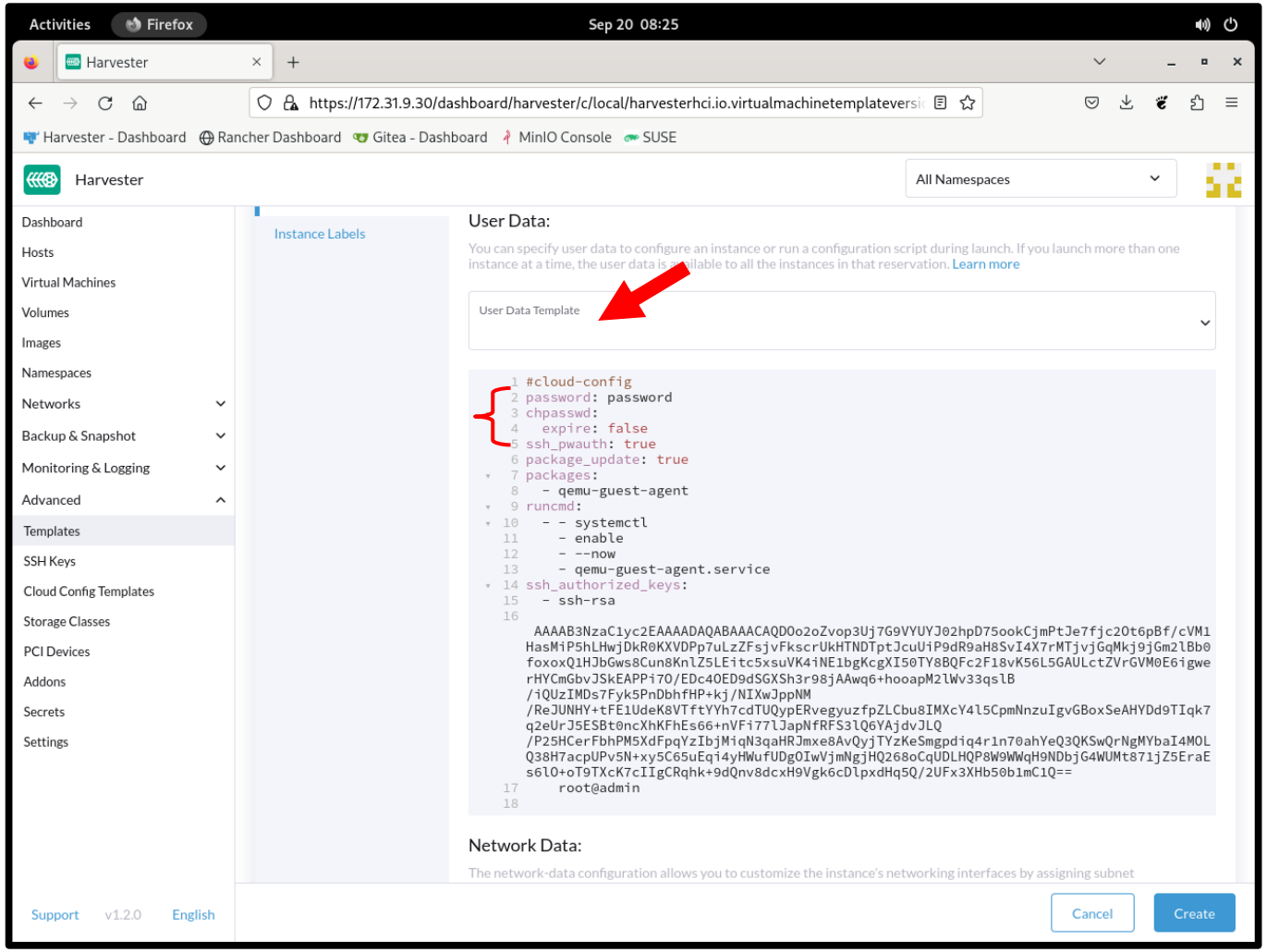
VM Templates: Create Template

- The same parameters can be set in a template as are available when creating a VM, hence the template creation screens look very similar to the VM creation screens.



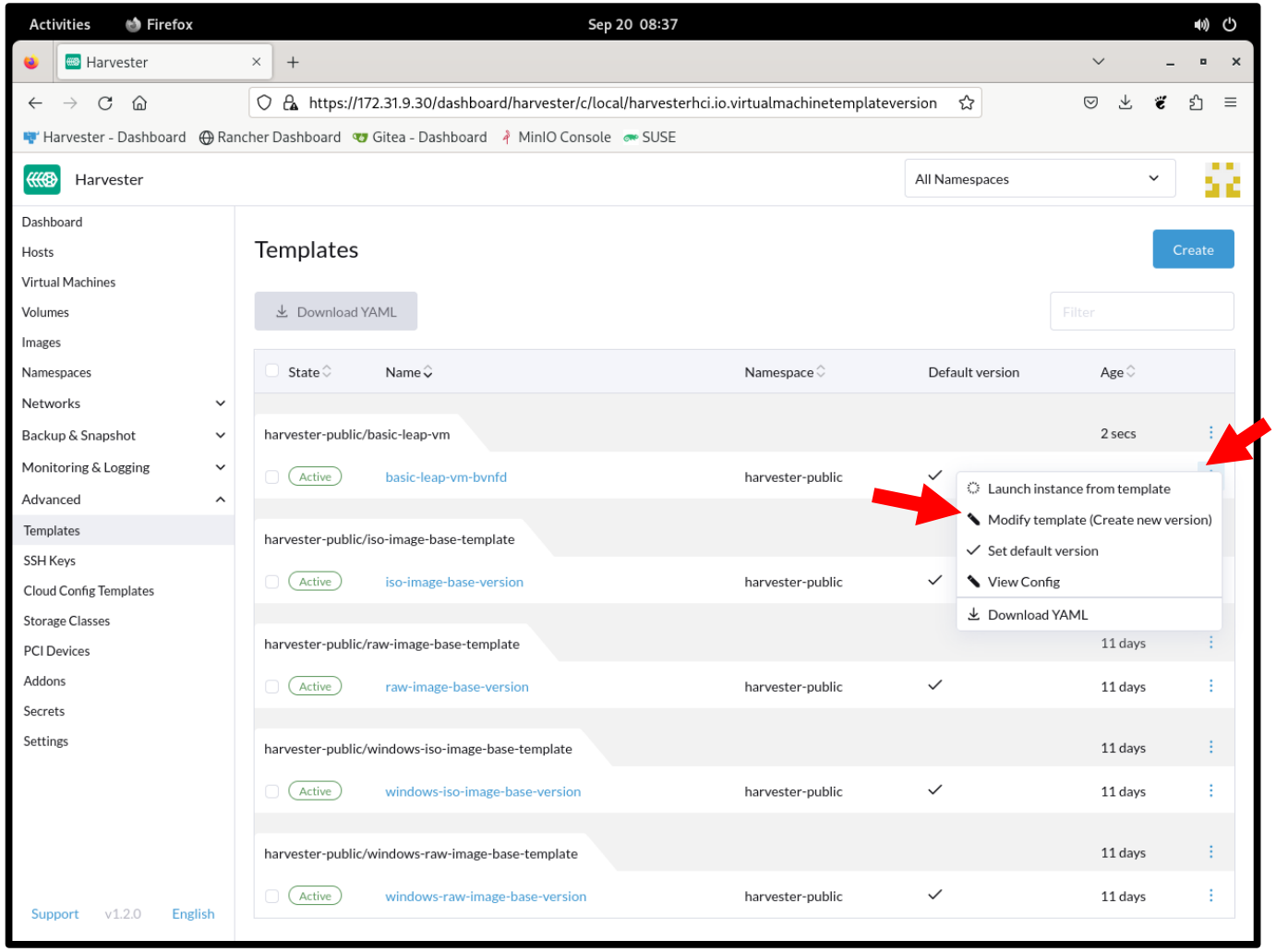
VM Templates: Create

- **Cloud Config Templates** can be referenced in a VM template.



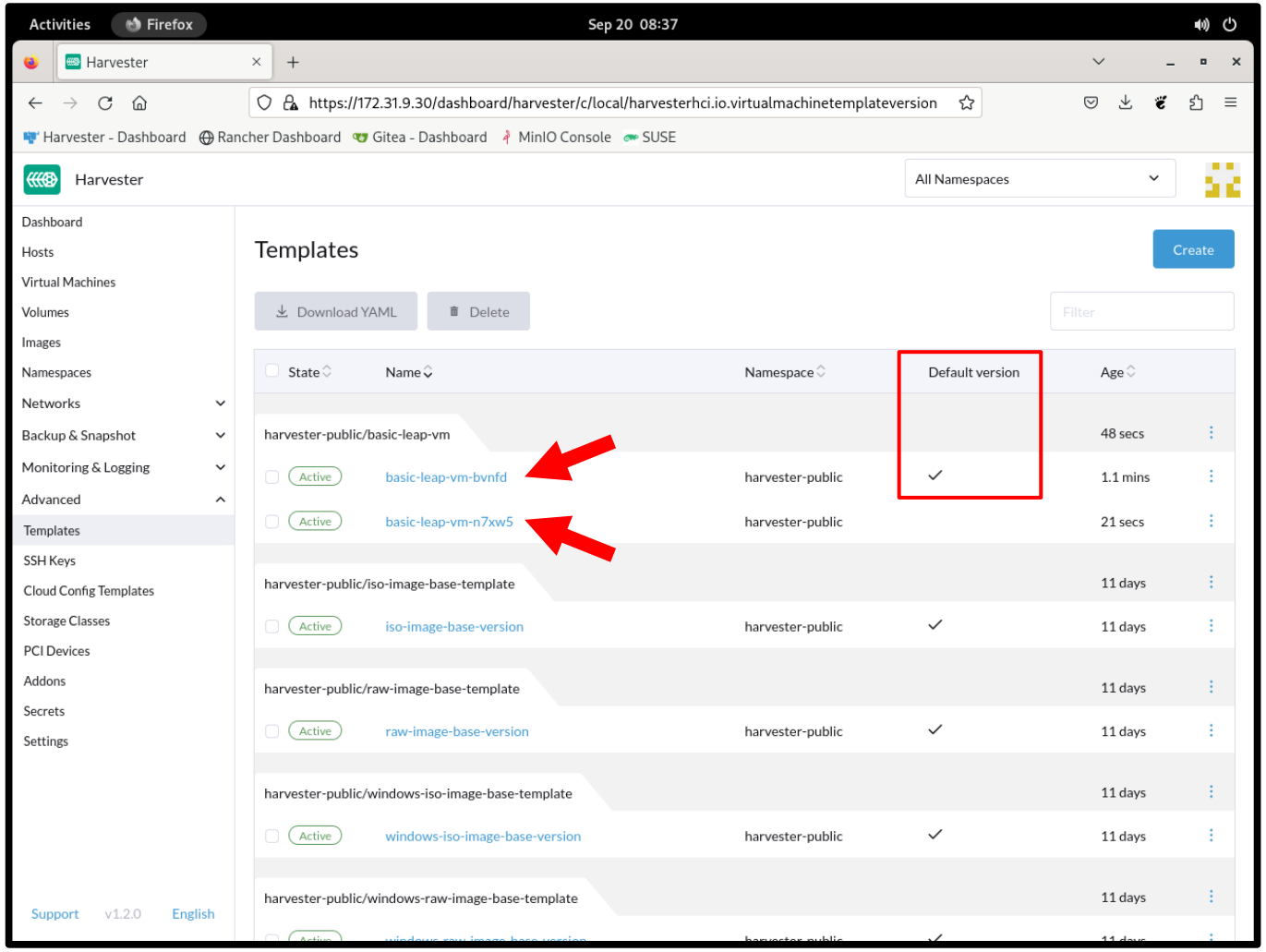
VM Templates: Modify Templates

- Existing templates can be modified creating a new version of the template by clicking the **3 dot** menu next to the template and selecting **Modify template (create new version)**.



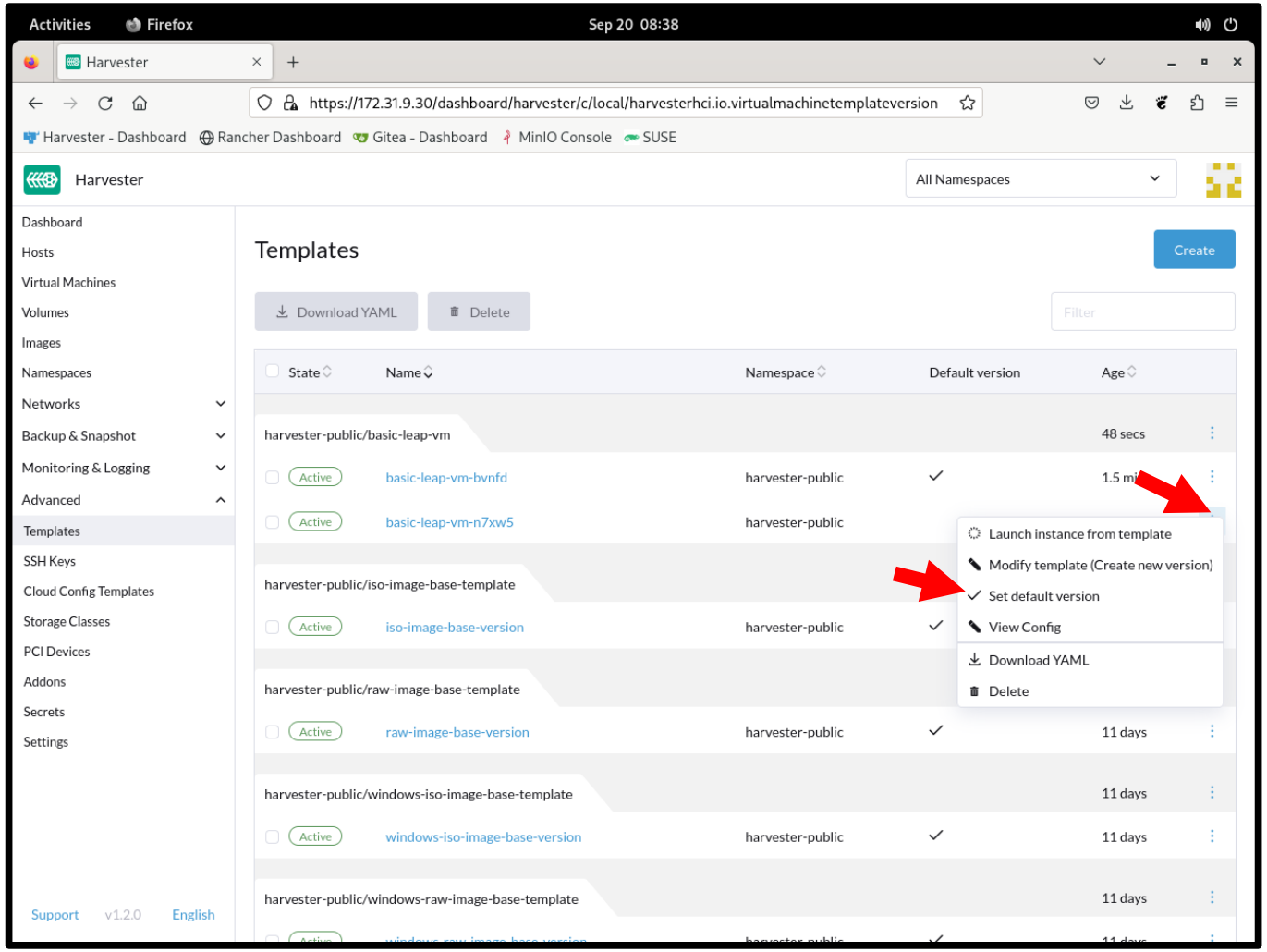
VM Templates: Template Version

- All versions of a template are displayed and available for use.



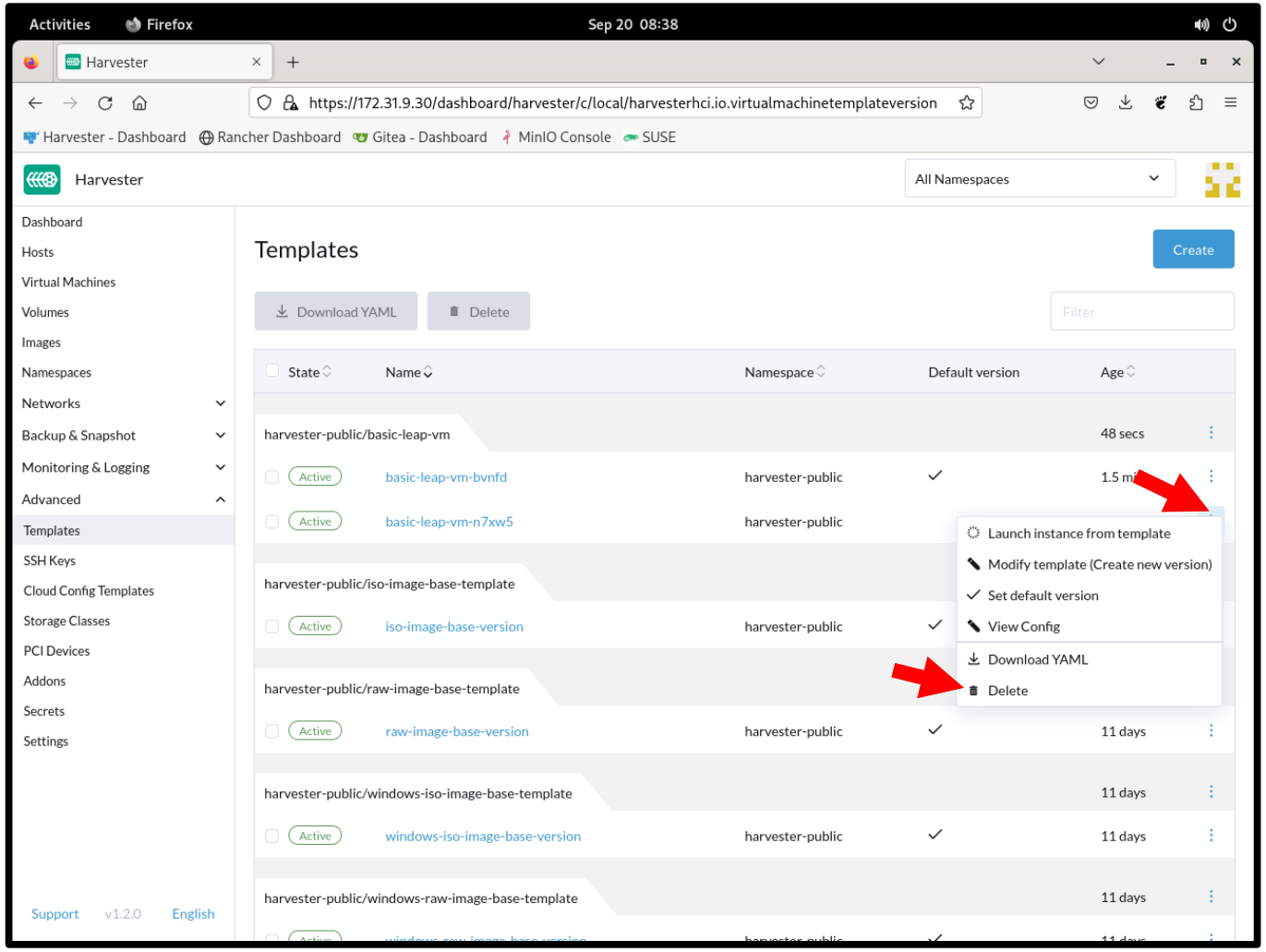
VM Templates: Default Version of a Template

- To set the default version of a template, click the **3 dot** menu next to the version you wish to set and select: **Set Default Version**



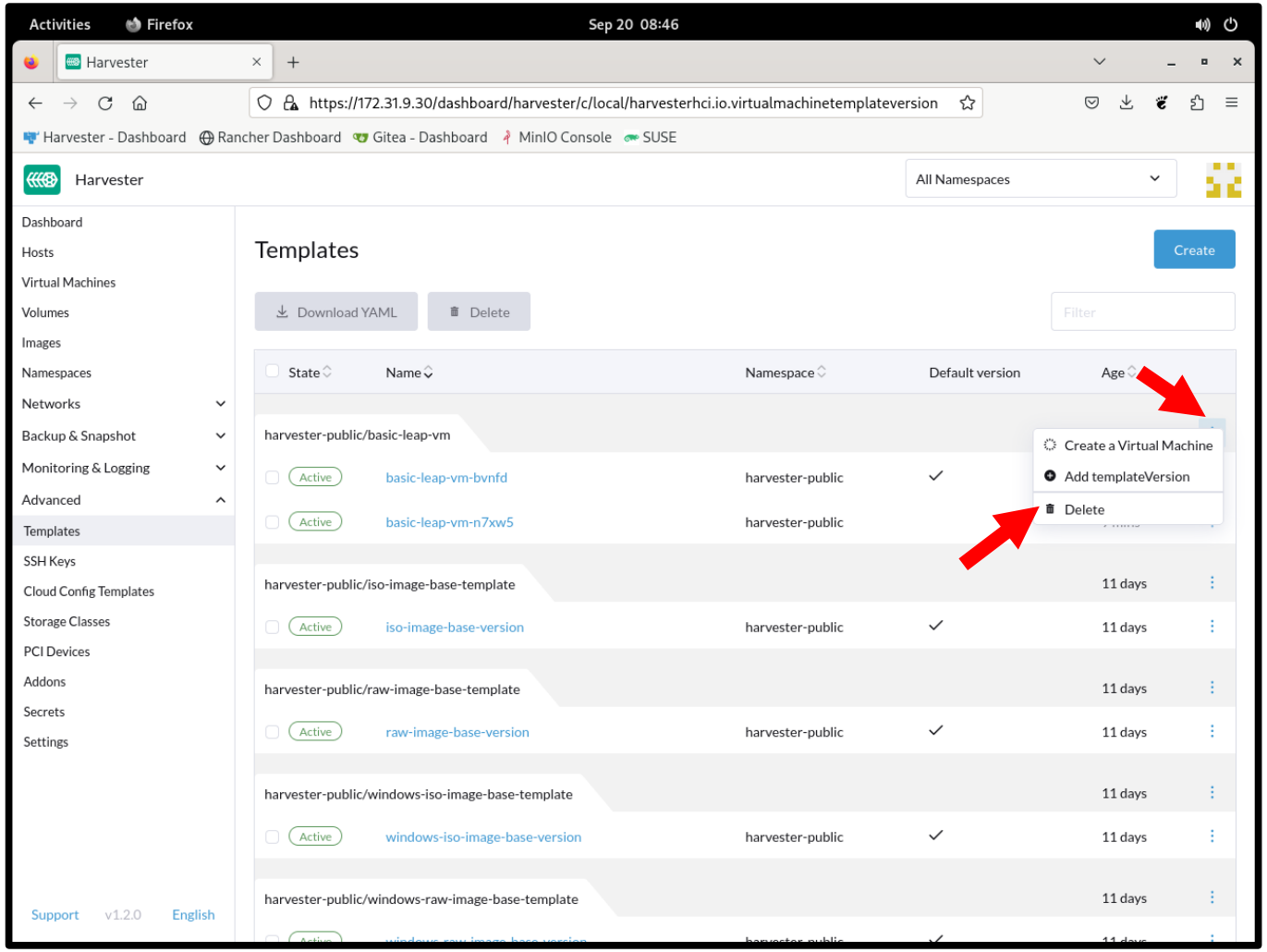
VM Templates: Delete a Version of a Template

- To delete a version of a template, select the **3 dot** menu on the right of the template version you wish to delete (in the white area next to the name of the template in blue text).
- Click: **Delete**
- Any version of a template may be deleted except the default version.



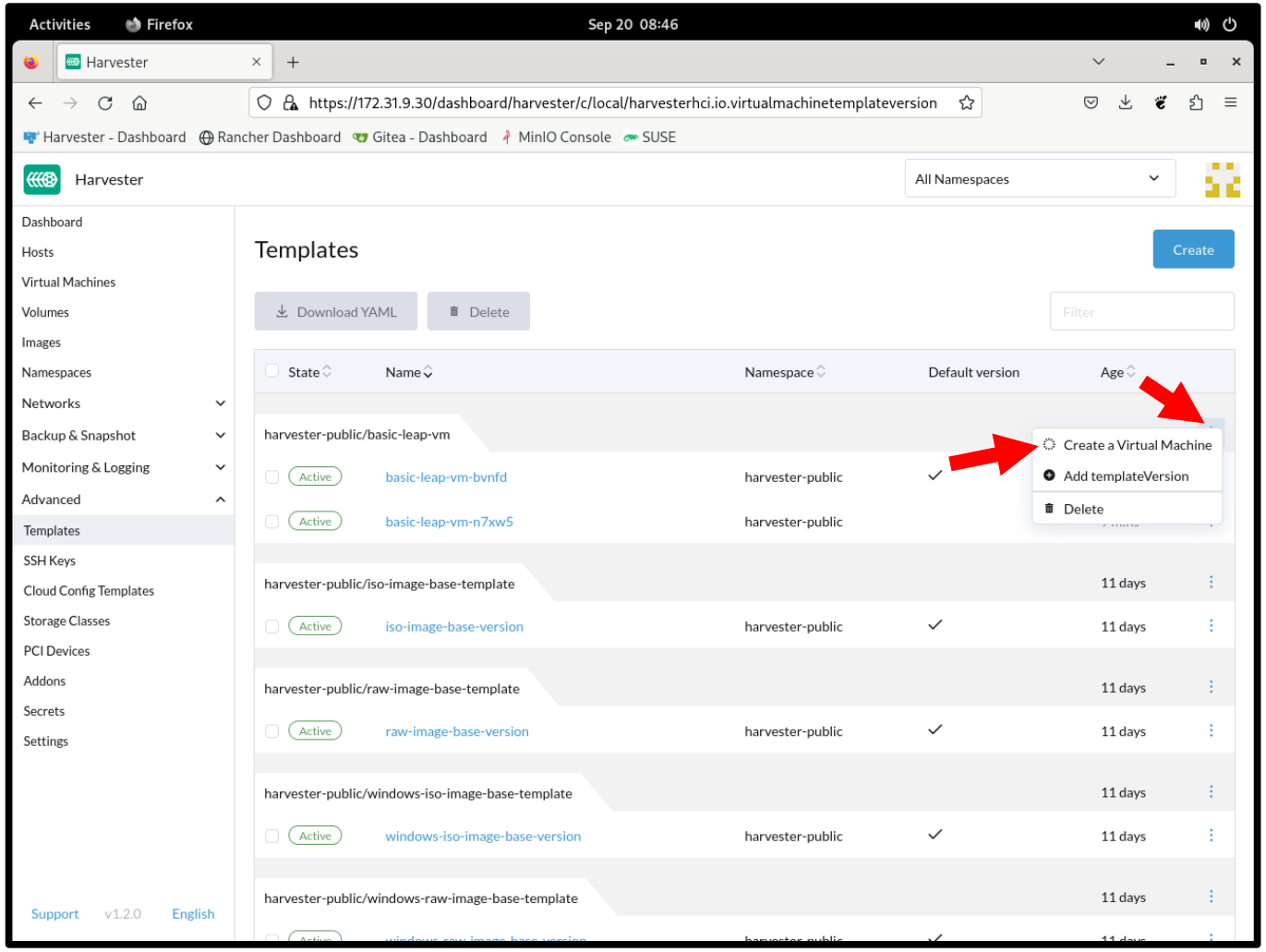
VM Templates: Delete a Template

- To delete a template, select the **3 dot** menu next to the template name (in the gray area next to the name of the template in black text).
- Click: **Delete**



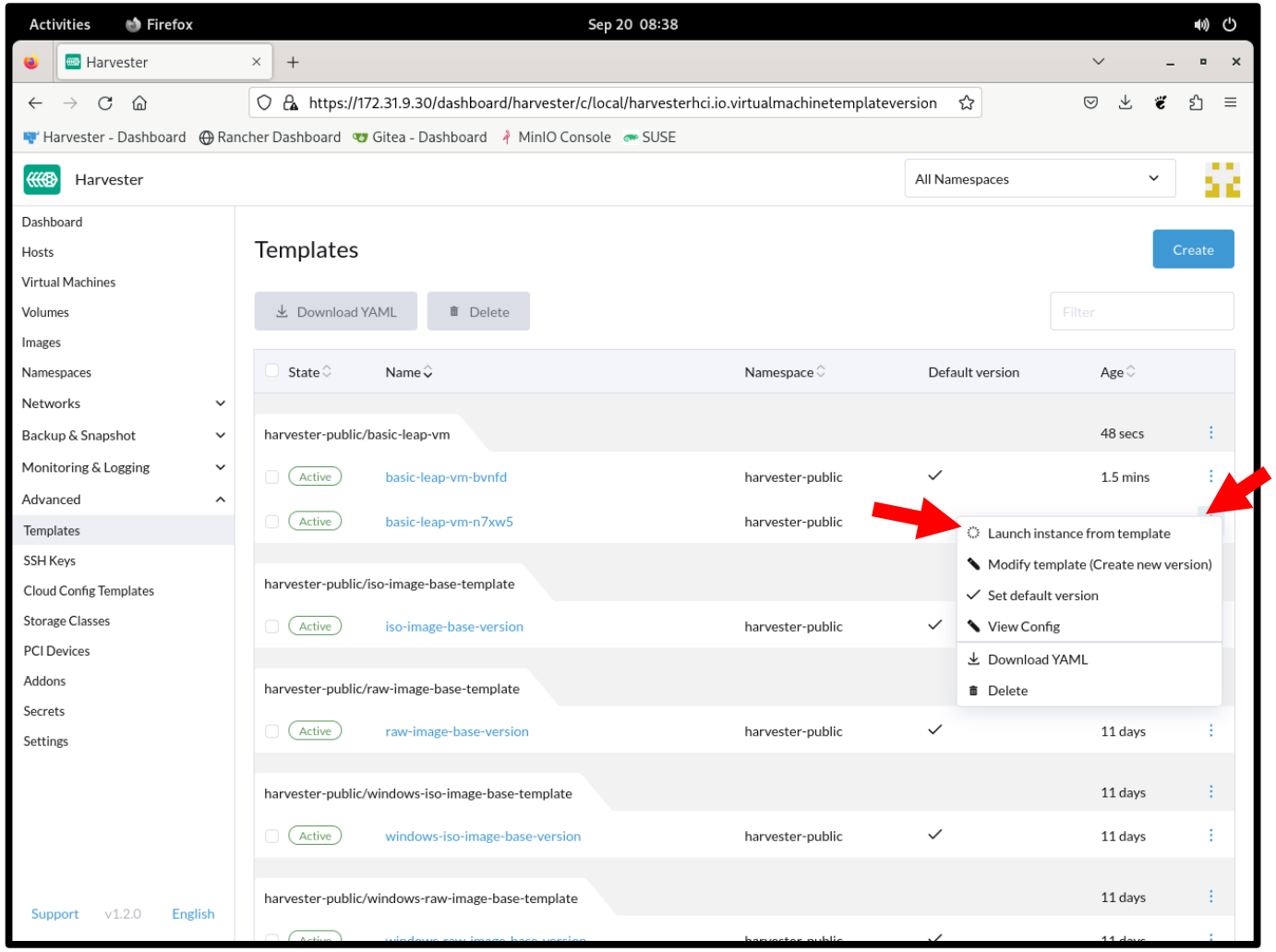
VM Templates: Launch Instance of a Template

- A new VM created from the default version of a template can be launched directly from the **Templates** screen.
- Click the **3 dot** menu next to the template you wish to use and select: **Create a Virtual Machine**



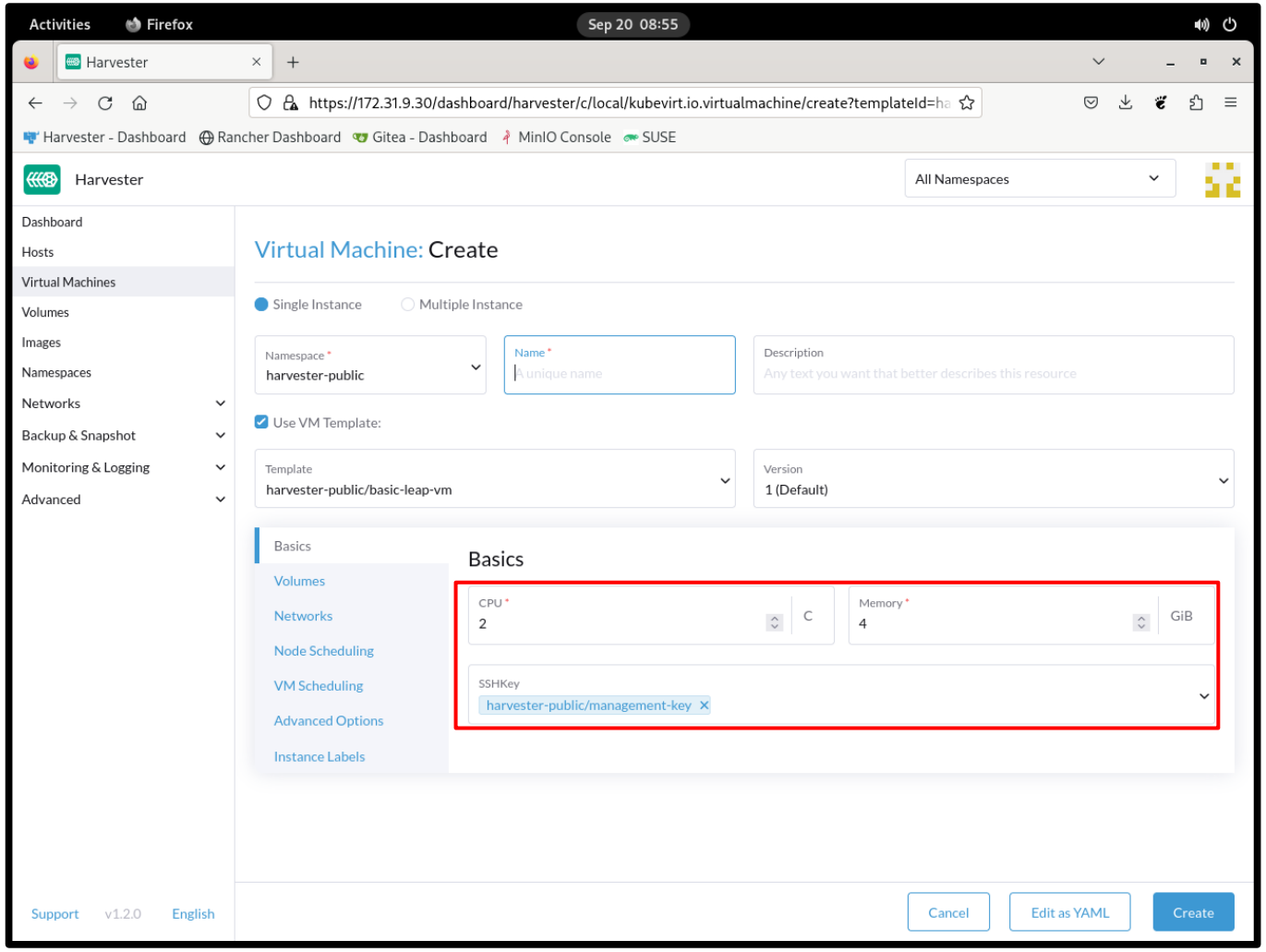
VM Templates: Launch Instance of a Template Version

- A new VM created from a specific version of a template can be launched directly from the **Templates** screen.
- Click the **3 dot** menu next to the template version you wish to use and select: **Launch instance from template**



VM Templates: VM Config from a Template

- The **Virtual Machine: Create** screen will be opened with all of the configuration preset from the default version of the selected template.
- The VM config can be edited further or the VM can simply be given a name and then created.



Live Migration

VM Live Migration

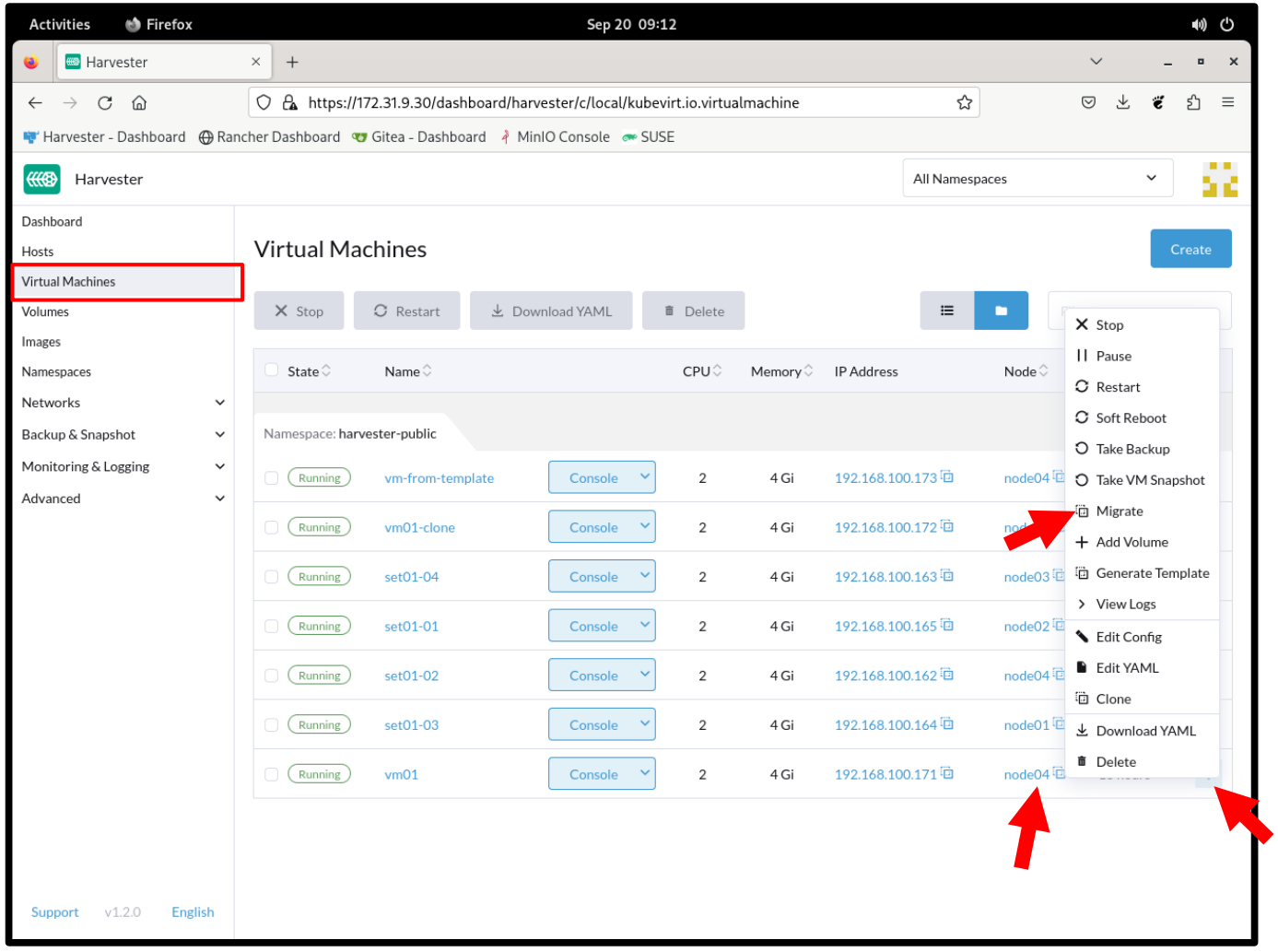
- If you need to manually change which cluster nodes VMs are running on, running VMs can be live migrated between cluster nodes with minimal or no downtime.

- VMs can still be accessed during the migration process.
  - There is a fraction of a second when they are not accessible at the end of the migration process while the VM is paused and the current working set of RAM is moved.
  - This is generally imperceptible depending on the VM's workload.

- Any cluster node that is not cordoned or in maintenance mode can have VMs migrated to it.

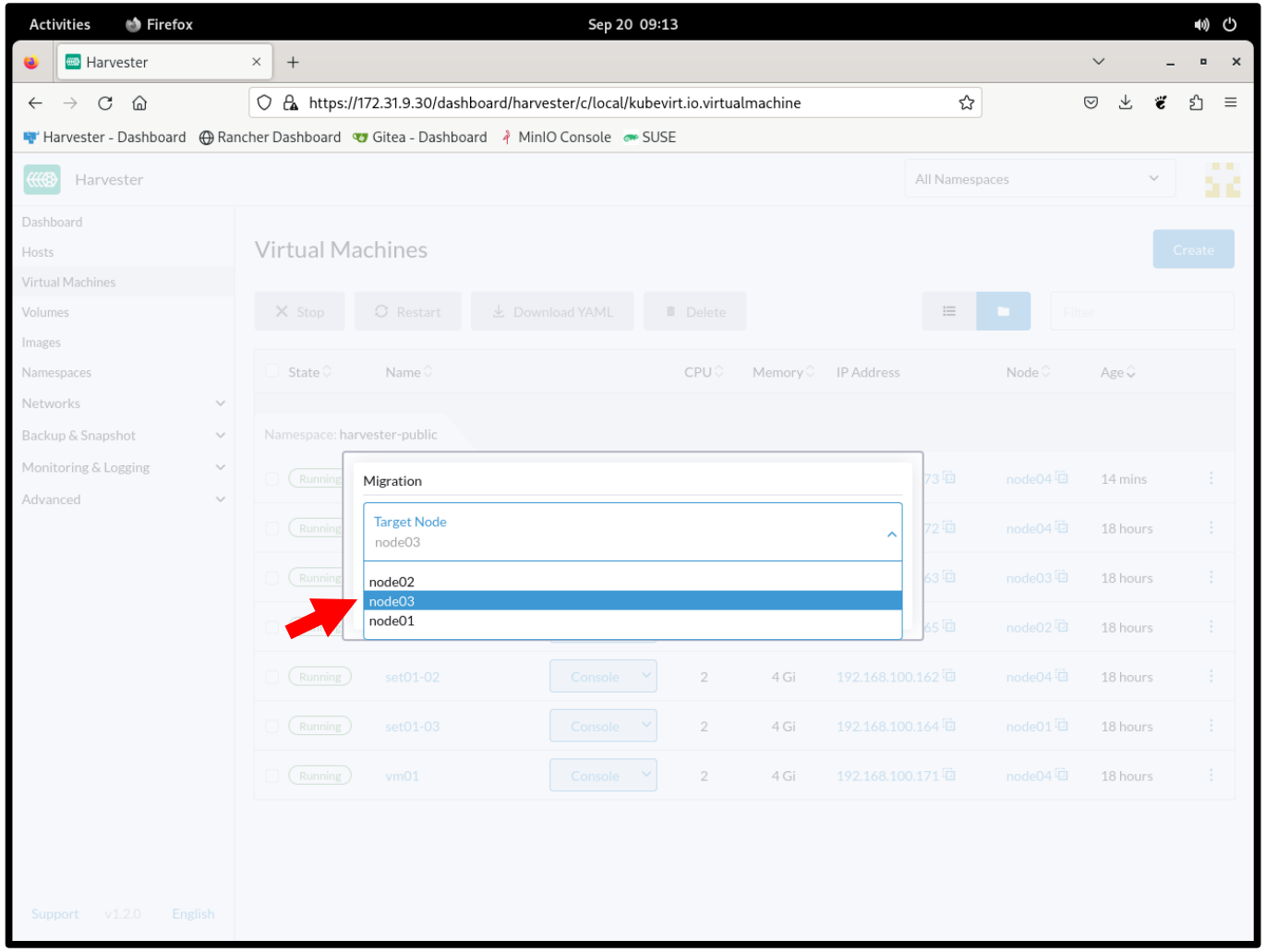
VM Live Migration

- VM live migration is initiated on the **Virtual Machines** screen.
- Note the cluster node on which the VM is currently running.
- Click the **3-dot** menu next to the VM you want to migrate and select: **Migrate**



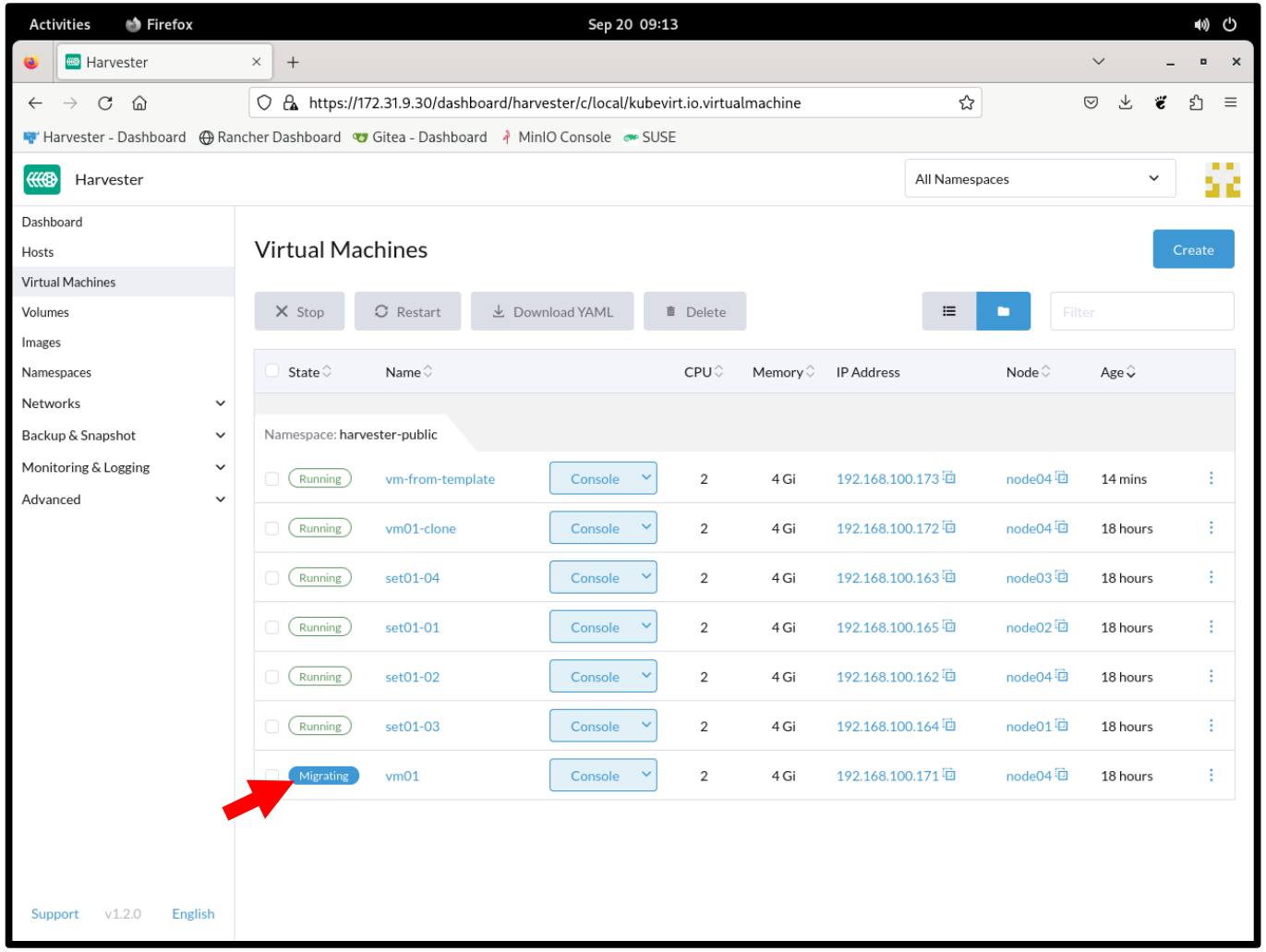
VM Live Migration

- On the **Migration** pop-up window select the target node to migrate the VM to.



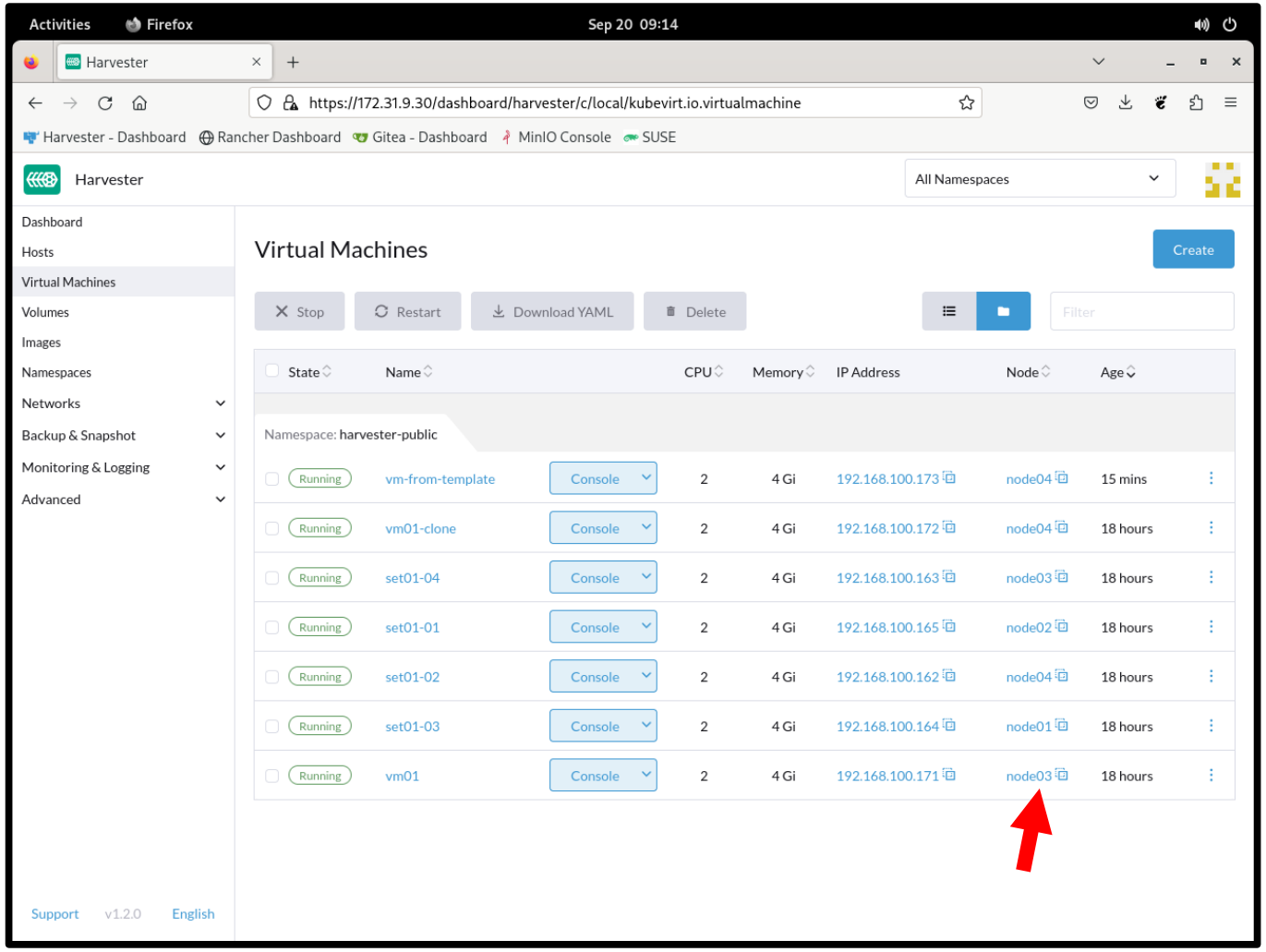
VM Live Migration

- As the VM is being live migrated the VM's status will show as : **Migrating**



VM Live Migration

- When the live migration is complete the VM will show as running on the new target cluster node.



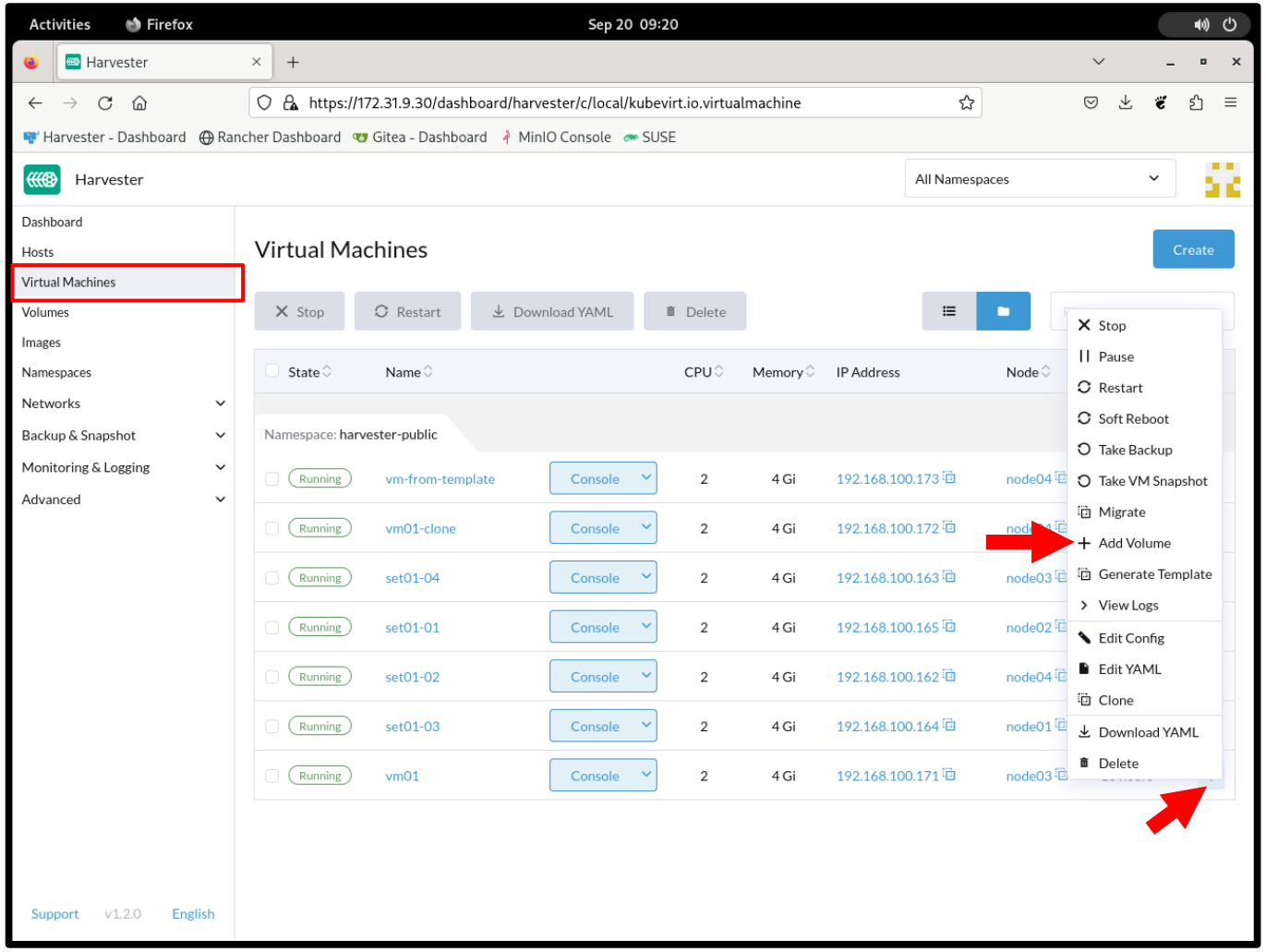
Hot-plug Volumes

Hot-plug Volumes

- Storage volumes are hot-pluggable for VMs running on Harvester.
- Volumes can be added to a running VM and will be accessible without having to reboot the VM.
- Volumes can be removed from a running VM without having to shut the VM down.
  - **Important**: Ensure that the volume is not being used inside of the VM before removing it.

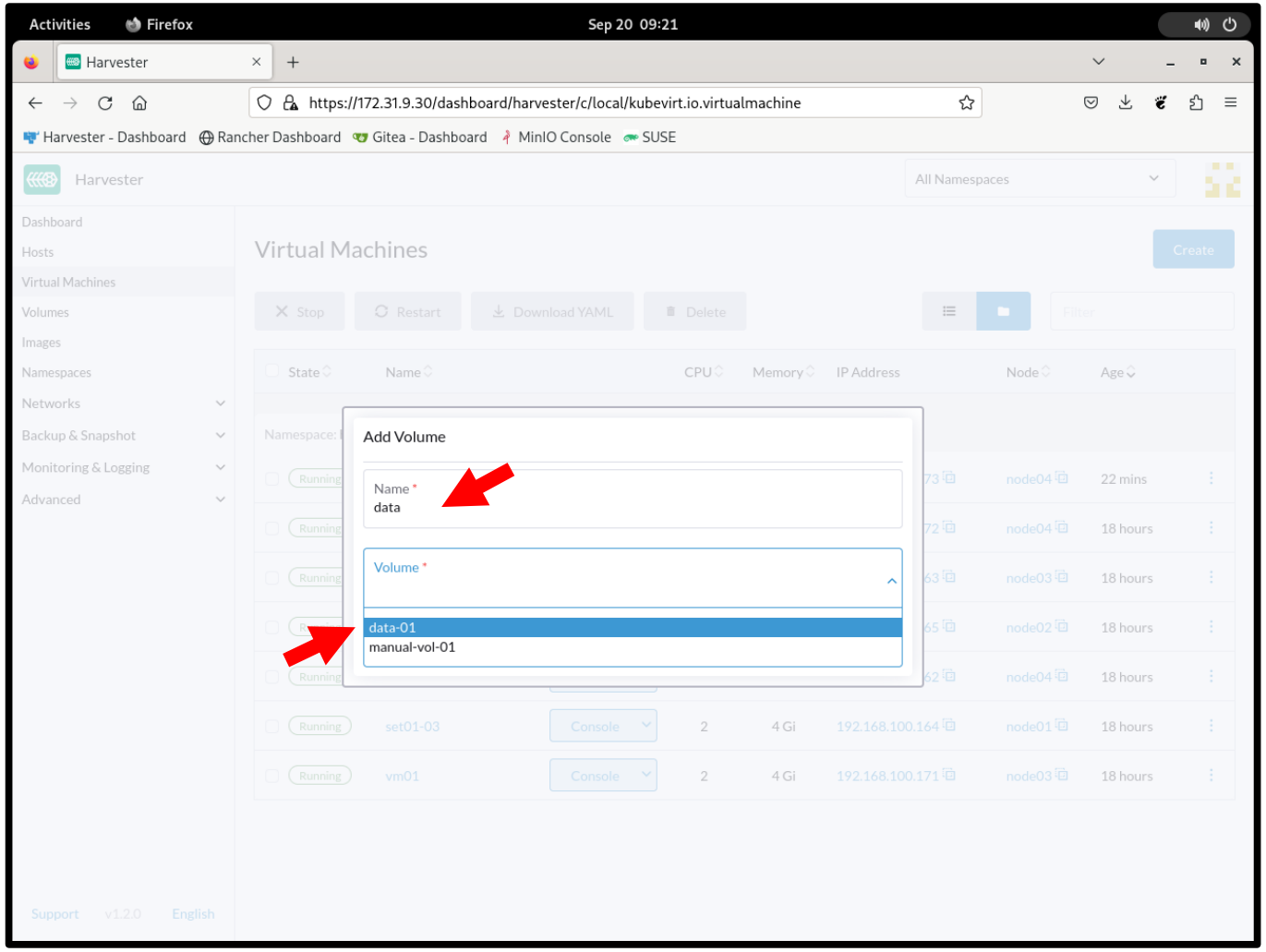
Hot-plug Volume

- Hot-plugging volumes into running VMs is initiated from the **Virtual Machines** screen.
- Click on the **3 dot** menu next to the VM you with to hot-plug a volume into and select: **Add Volume**



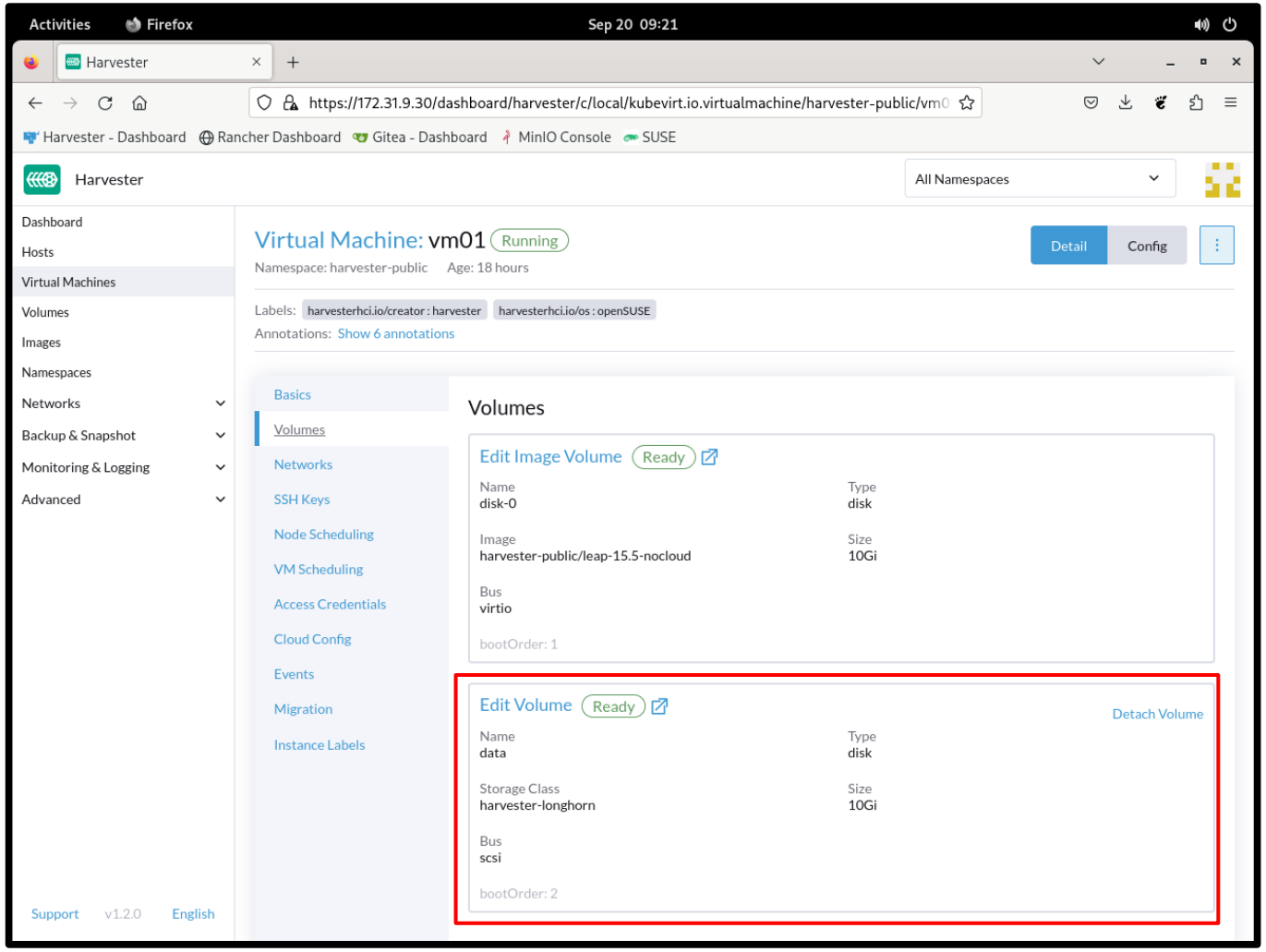
Hot-plug Volume: Attach Volume

- On the **Add Volume** pop-up window give the volume a **Name** and select the **Volume** to attach.



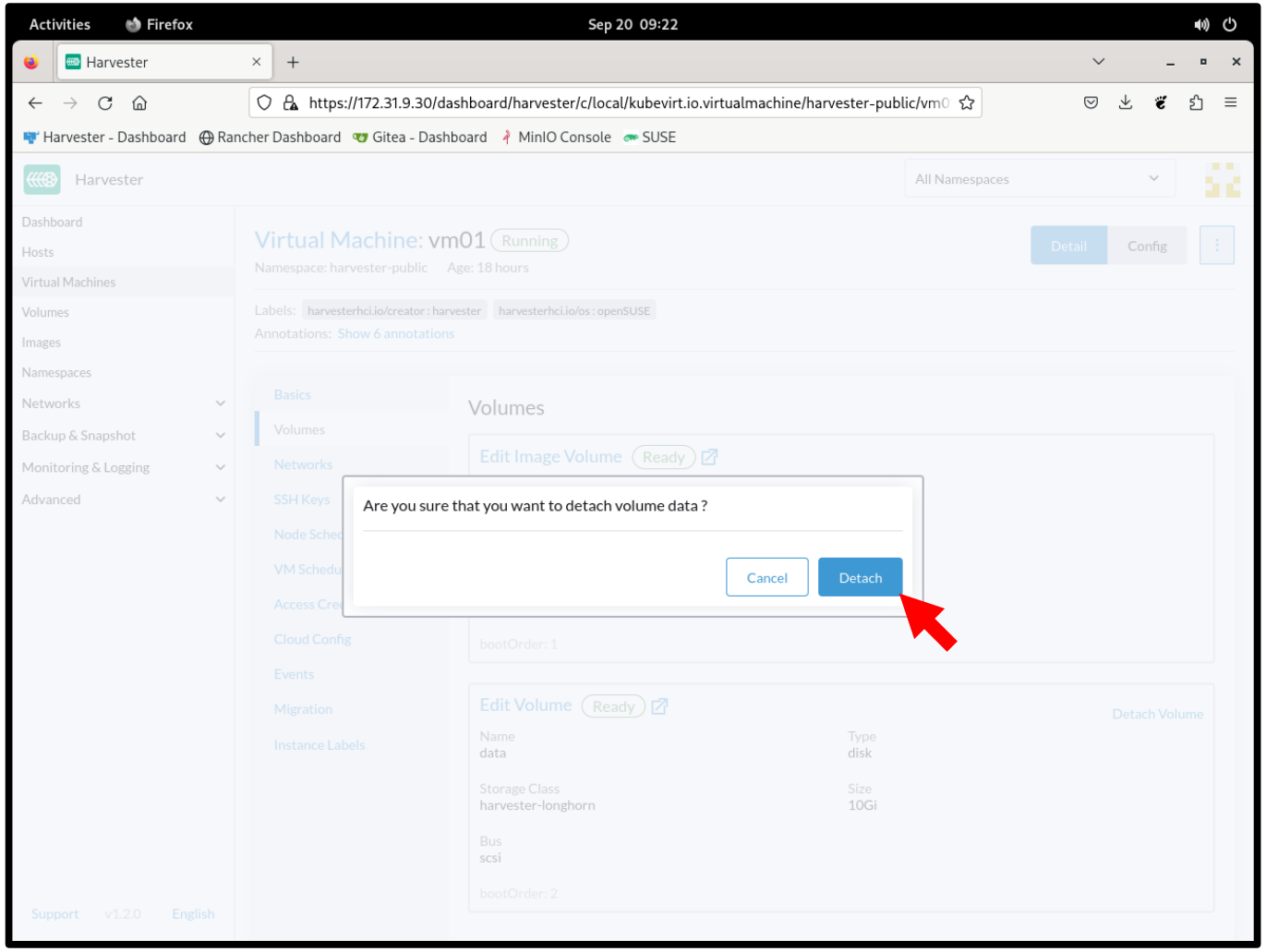
Hot-plug Volume: Detach Volume

- On the VM's info screen, on the Volumes tab, you can see the volumes that are attached.
- From this screen the additional (non root disk) volumes can be detached.



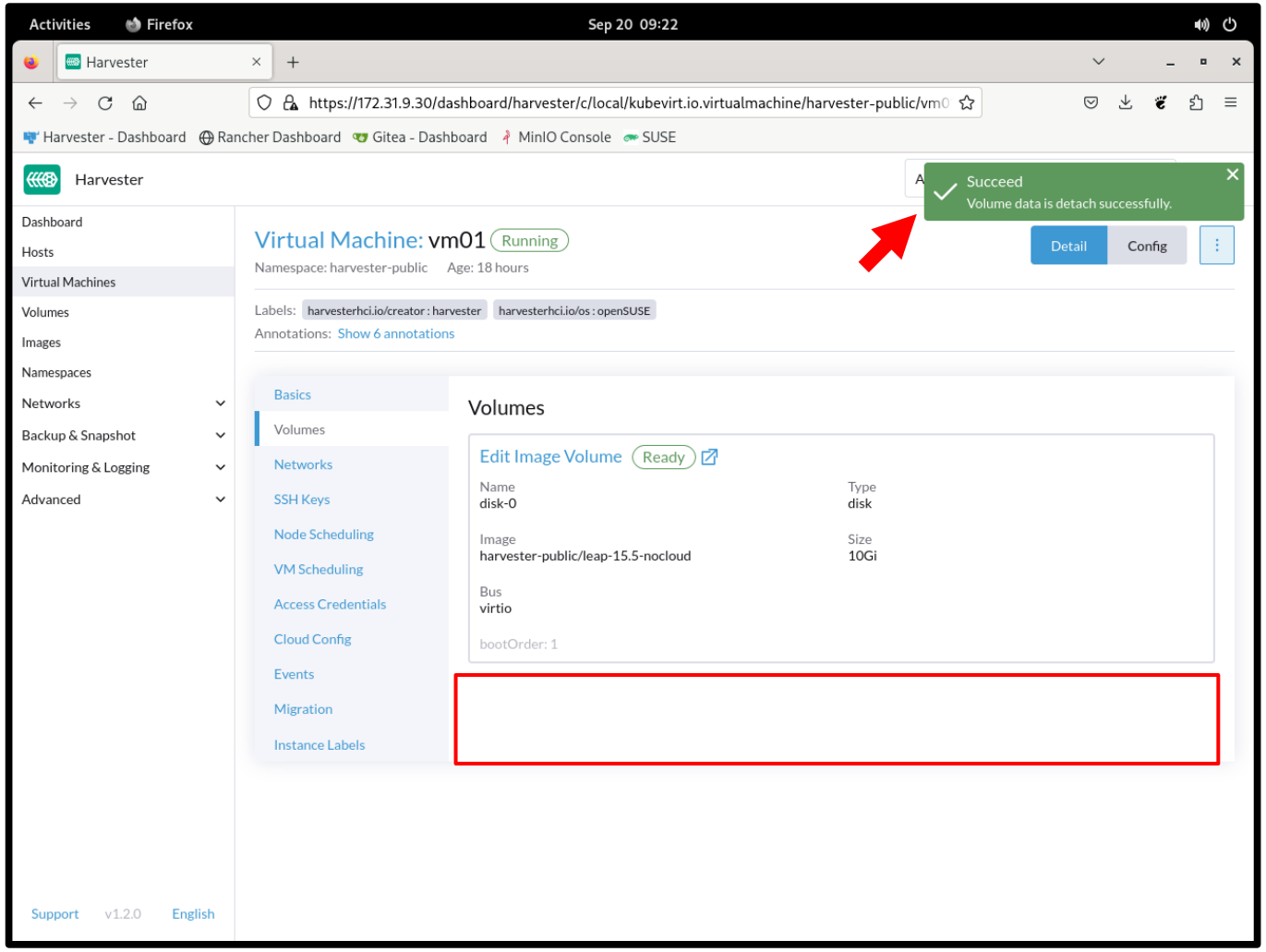
Hot-plug Volume: Detach Volume

- When prompted and if you are sure click: **Detach**



Hot-plug Volume: Detach Volume

- The volume will no longer show as being attached to the VM.



Virtual Machine Snapshots

VM Snapshots vs VM Backups

Snapshots

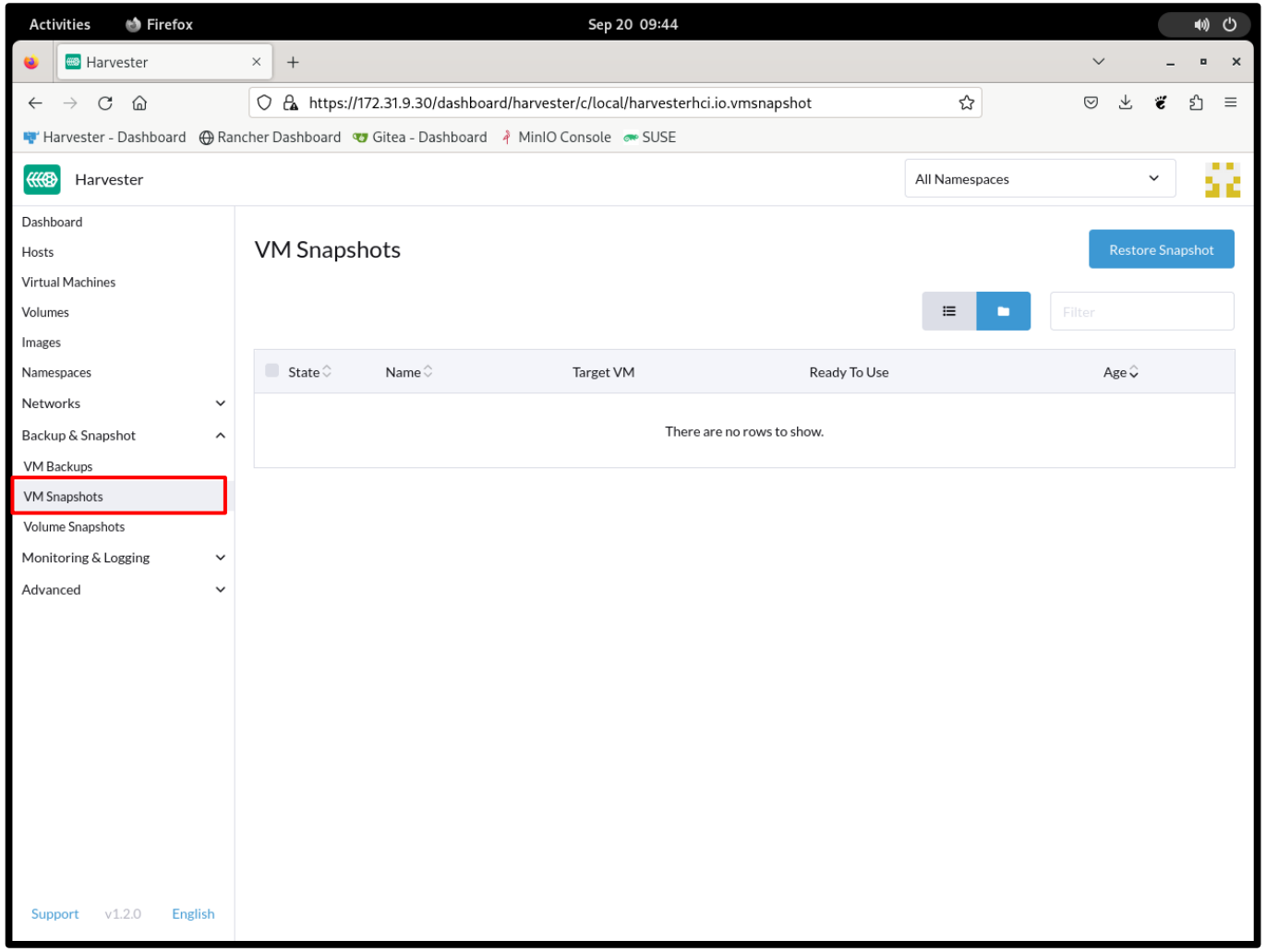
- VMs can be snapshotted using the volume snapshot functionality of the embedded Longhorn.
- VM snapshots are stored internal to the Harvester cluster in Longhorn as separate volumes.
- VM configuration is also included in the VM snapshots.

Backups

- VMs can be backed up in Harvester by leveraging the backup functionality of the embedded Longhorn.
- Backups of volumes ares saved external to the Harvester cluster in S3 or NFS servers.
- VM configuration is also included in the VM backups.

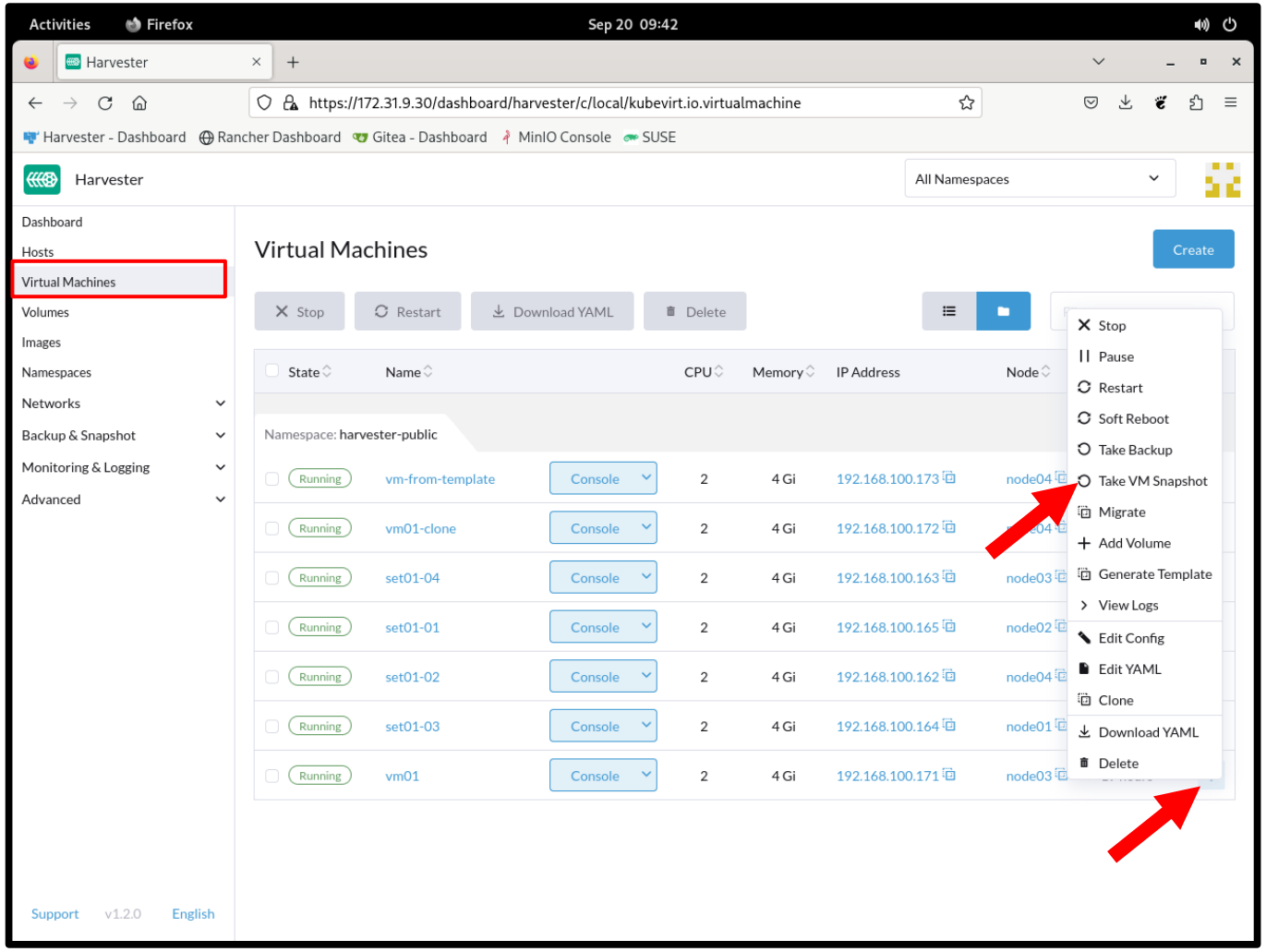
Snapshot a VM

- Virtual Machine shapshots are displayed on the **Backups & Snapshots → VM Snapshots** tab.



Snapshot a VM

- Creating a VM snapshot is initiated from the **Virtual Machines** screen.
- Click on the **3 dot** menu next to the VM you wish to back up and select: **Take VM Snapshot**

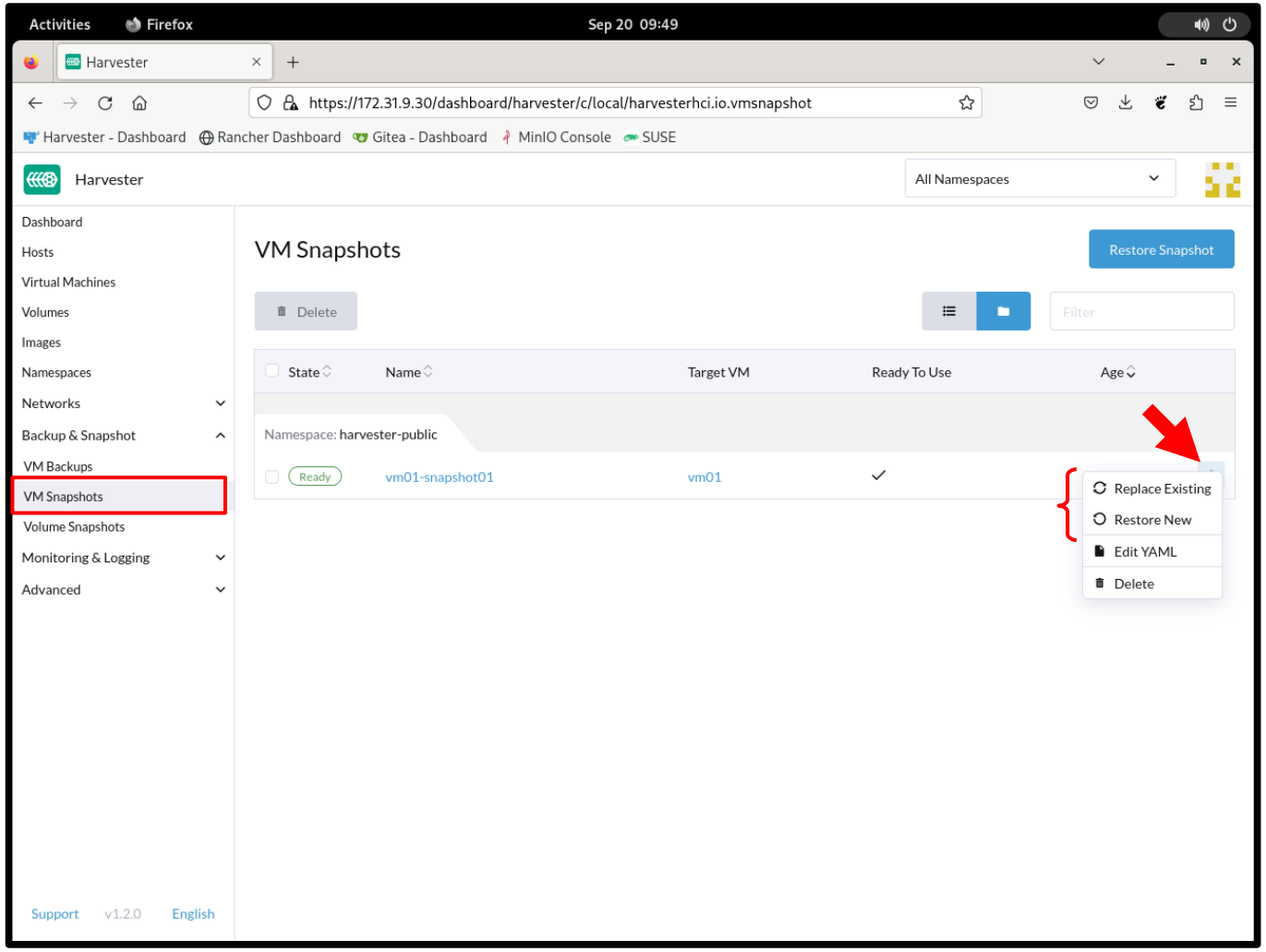


Restore VM Snapshots

- VM snapshots can be restored at any time.
- A snapshot can be restored as:
  - A new VM
  - Replacing the original source VM

Restore a VM Snapshot

- Restoring a backup is initiated from the **Backups & Snapshots → VM Snapshots** screen.
- Locate the snapshot you wish to restore, click on the **3 dot** menu and select either **Restore New** or **Replace Existing**.



Virtual Machine Backups

VM Snapshots vs VM Backups

Snapshots

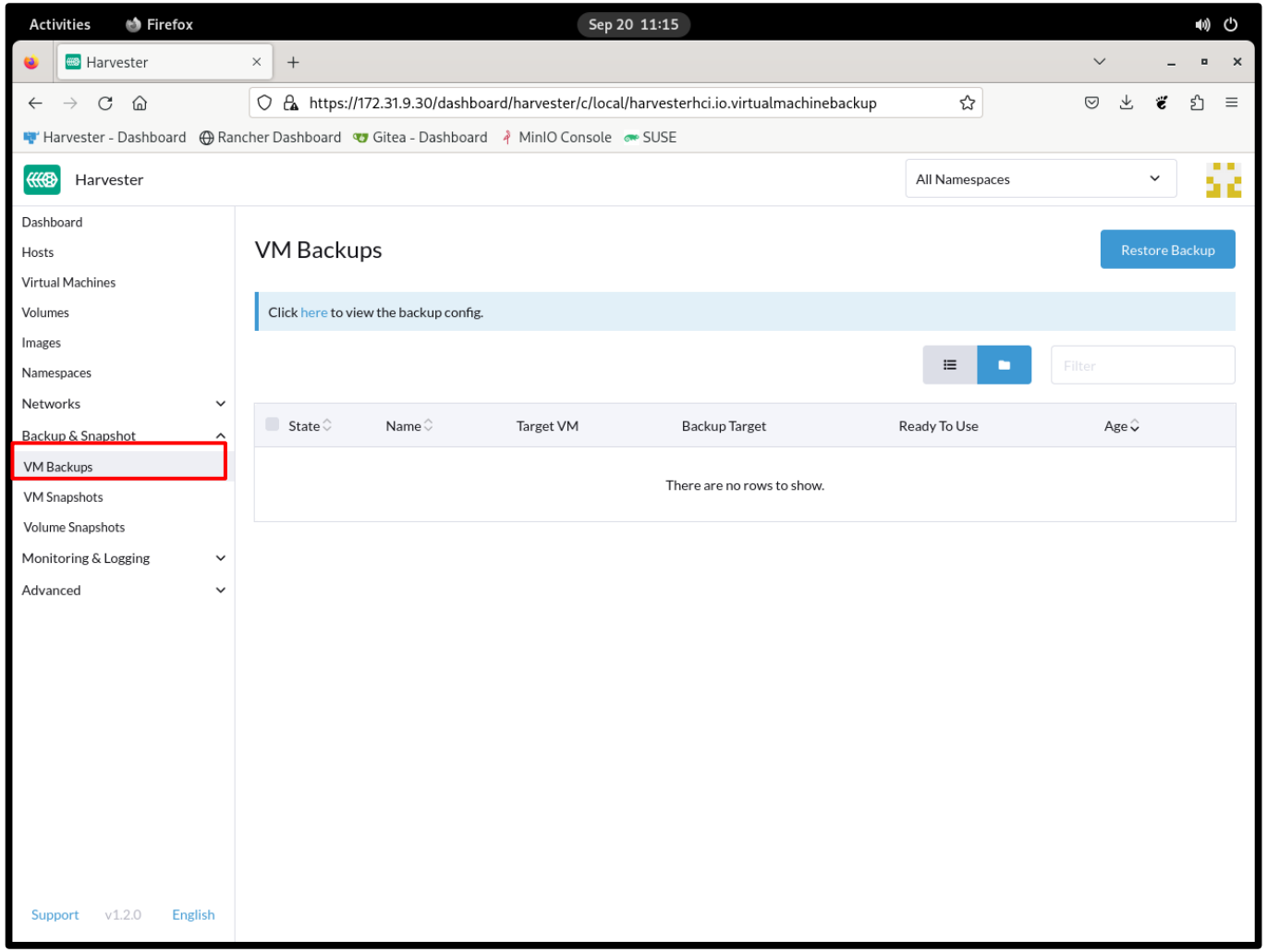
- VMs can be snapshoted using the volume snapshot functionality of the embedded Longhorn.
- VM snapshots are stored internal to the Harvester cluster in Longhorn as separate volumes.
- VM configuration is also included in the VM snapshots.

Backups

- VMs can be backed up in Harvester by leveraging the functionality of the embedded Longhorn.
- Backups of volumes ares saved external to the Harvester cluster in S3 or NFS servers.
- VM configuration is also included in the VM backups.

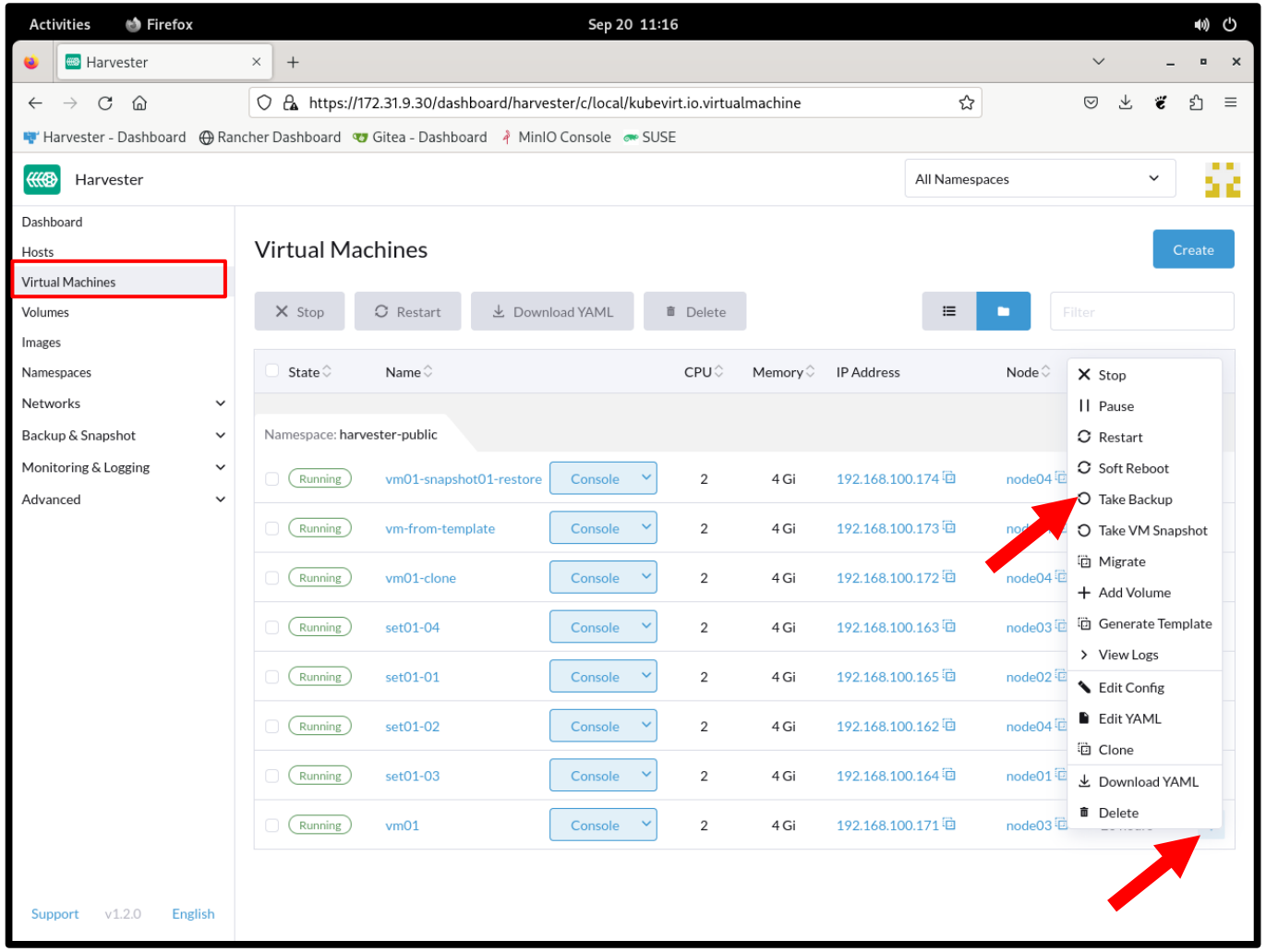
Backup VM

- Virtual Machine backups are displayed on the **Backups & Snapshots→ VM Backups** tab.



Backup VM

- Creating a VM backup is initiated from the **Virtual Machines** screen.
- Click on the **3 dot** menu next to the VM you wish to back up and select: **Take Backup**

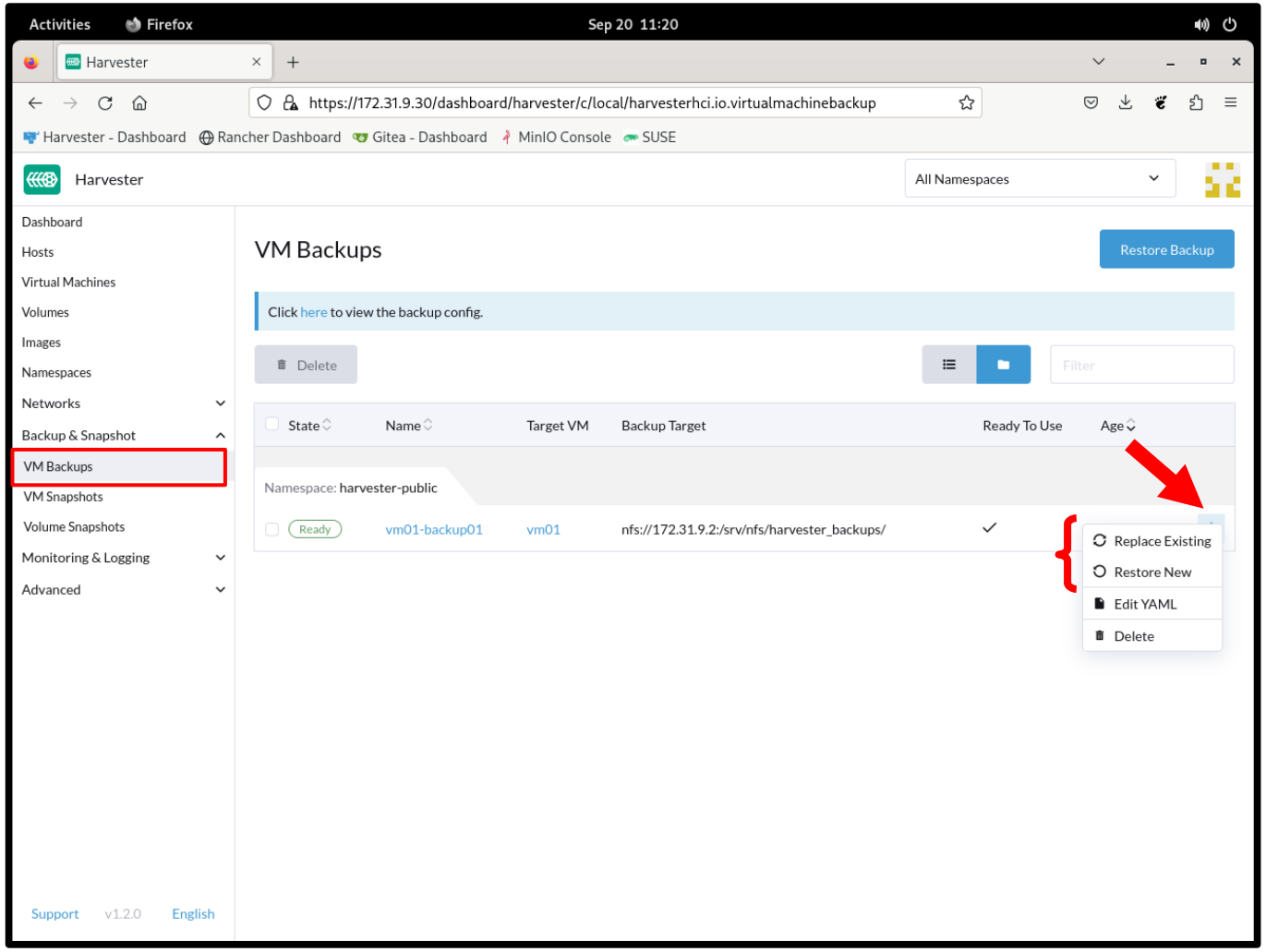


Restore VM Backups

- VM backups can be restored at any time.
- A backup can be restored as:
  - A new VM
  - Replacing the original source VM

Restore a VM Backup

- Restoring a backup is initiated from the **Backups & Snapshots → VM Backups** screen.
- Locate the backup you wish to restore, click on the **3 dot** menu and select either **Restore New** or **Replace Existing**.



Section Summary

Section Summary

- Describe virtualization concepts in Harvester
  - Harvester uses KubeVirt for virtualization management.
  - KubeVirt extends Kubernetes giving it the ability to understand and manage virtualization related objects.
- Upload and use virtual machine images
  - Virtual machine images are uploaded and stored in Harvester and used to launch new VMs.
  - Images can be uploaded from your local machine or can be downloaded from a remote URL.
  - New VM images can be created from existing VM root disks.

Section Summary

- Create and manage the lifecycle of virtual machines
  - VMs can be launched as a single instance or as multiple instances that are unique but otherwise configured identically.
  - Existing VMs can be cloned.
  - Cloud config templates can be created and used to more easily provide cloud-init configuration info to VMs when they are launched.
  - VM templates can be created which contain VM configuration information that can be used to more quickly create new VMs.

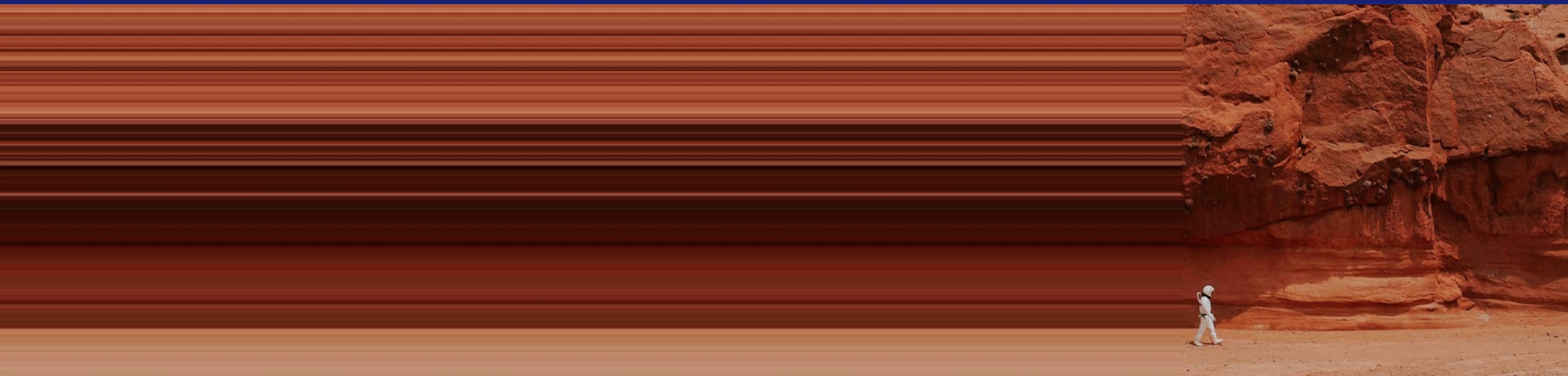
Section Summary

- Backup and Restore Virtual Machines
  - Snapshots of VMs can be created with the data in the snapshots being stored in the local Longhorn storage cluster.
  - Snapshots of VMs can be restored as new VMs or by replacing the original instance the snapshot was created from.
  - Backups of VMs can be created that are stored externally from the cluster in an NFS server or S3 bucket.
  - Like with snapshots, backups of VMs can be restored as new VMs or by replacing the original instance the backup was created from.

SUSE

Section: 8

Harvester Maintenance, Troubleshooting Tools and Cluster Upgrade



Learning Objectives

At the end of this section you will be able to:

- Place cluster nodes in states where maintenance can be performed on the nodes and the cluster.
- Enable and access additional tools used for troubleshooting the cluster.
- Upgrade a Harvester cluster.

Topics

1 Harvester Host Management

2 Additional Troubleshooting Tools in Harvester

3 Support Bundles in Harvester

4 Harvester Cluster Upgrade

Harvester Host Management

Node States for Maintenance and Troubleshooting

- Cluster nodes can be put into special states that can aid in the maintenance and troubleshooting processes:

  States:

  - Cordoned
  - Maintenance Mode

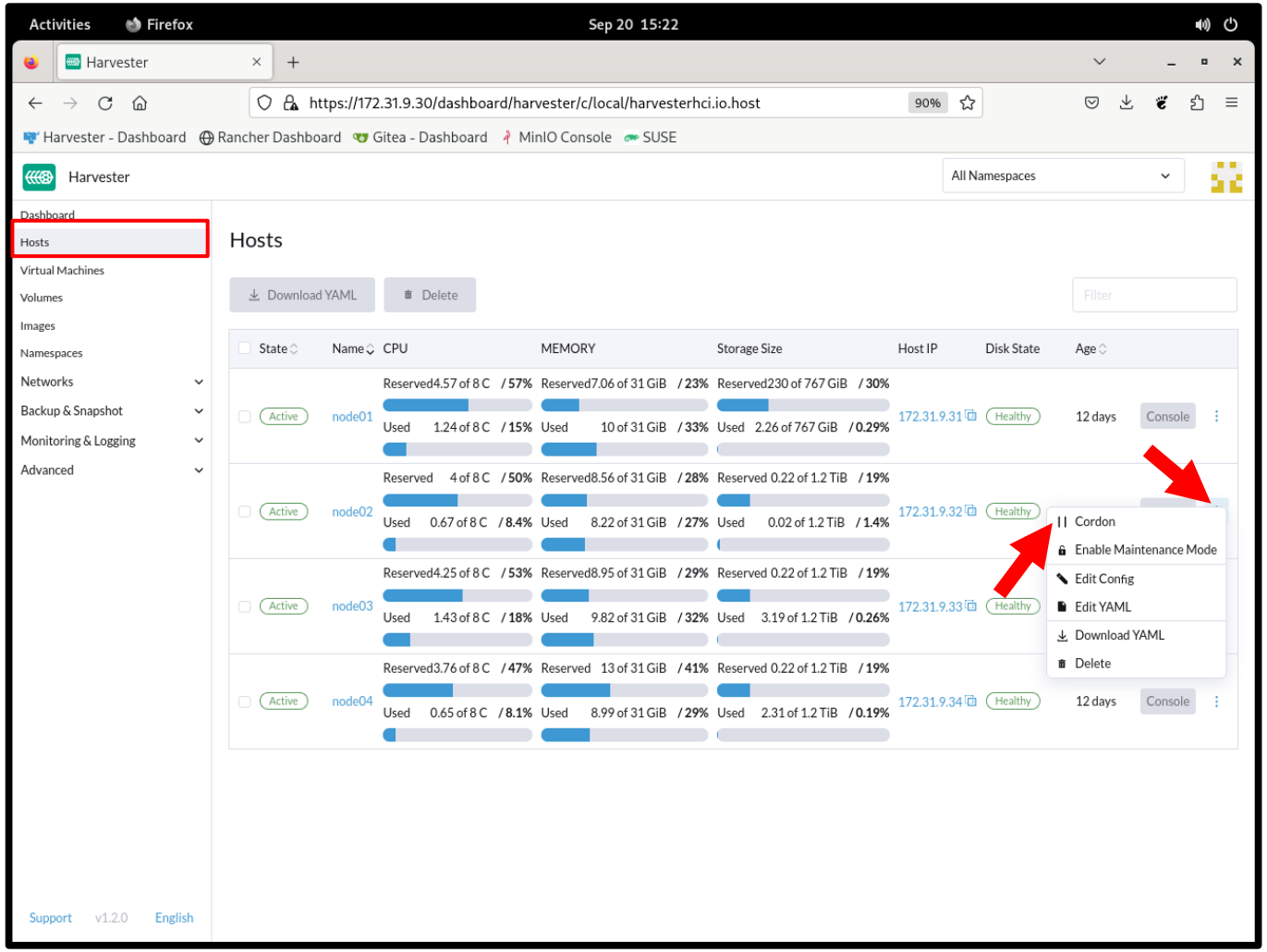
Node Cordoning

- What is node cordoning?
  - Marks a node as "unschedulable".
  - No new VMs will be run on the node.
  - Any VMs currently running on the node are not affected.
  - Node still participates in the cluster.

- Why use node cordoning?
  - Nodes require small maintenance tasks.
  - Nodes may be misbehaving and need to be isolated.

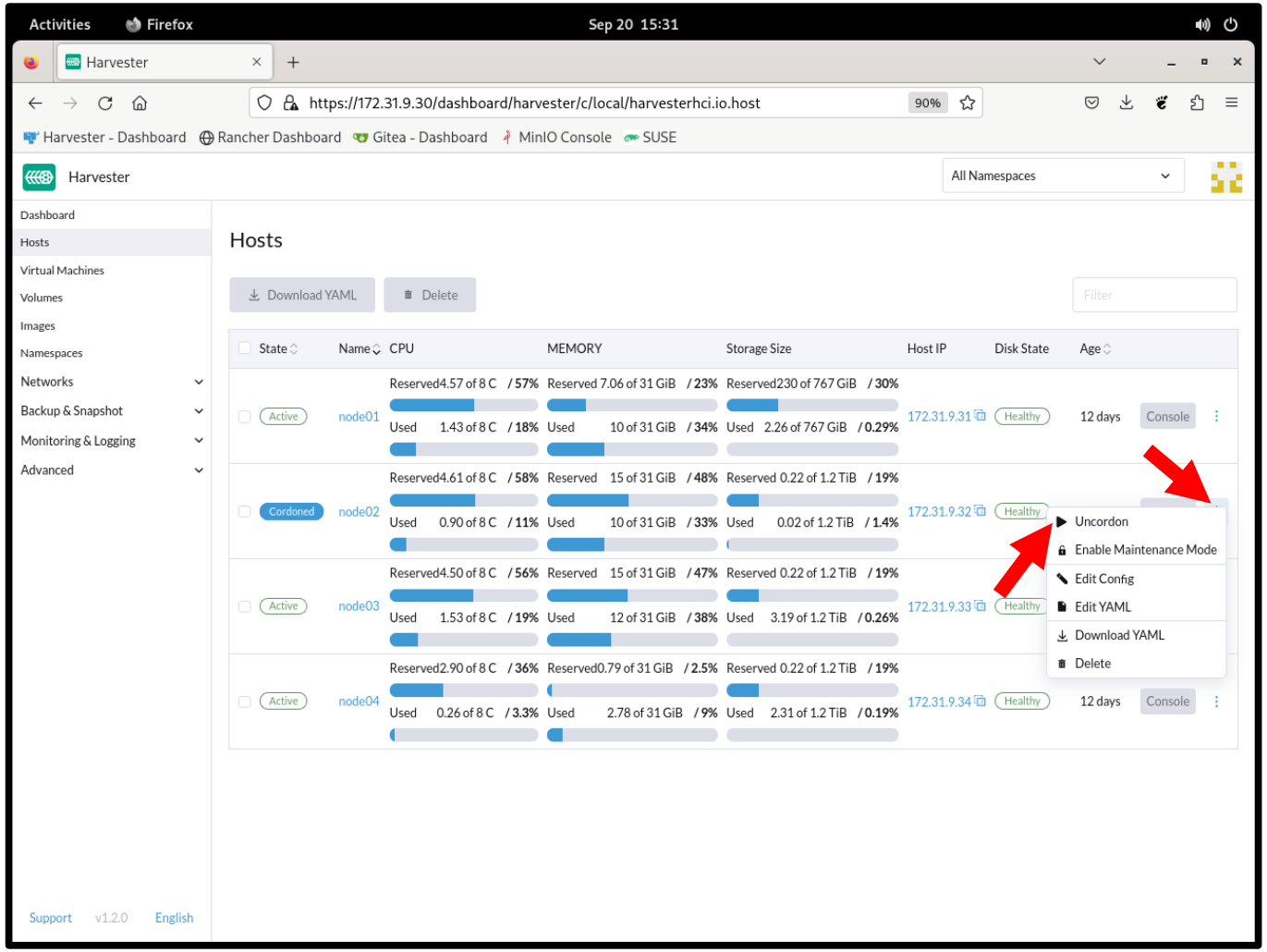
Node Cordoning

- Navigate to the **Hosts** screen.
- Locate the node you wish to cordon and click the **3 dot** menu on the right of the node.
- Select: **Cordon**



Node Cordoning

- To uncordon a node, click the **3 dot** menu on the right of the node.
- Select: **Uncordon**



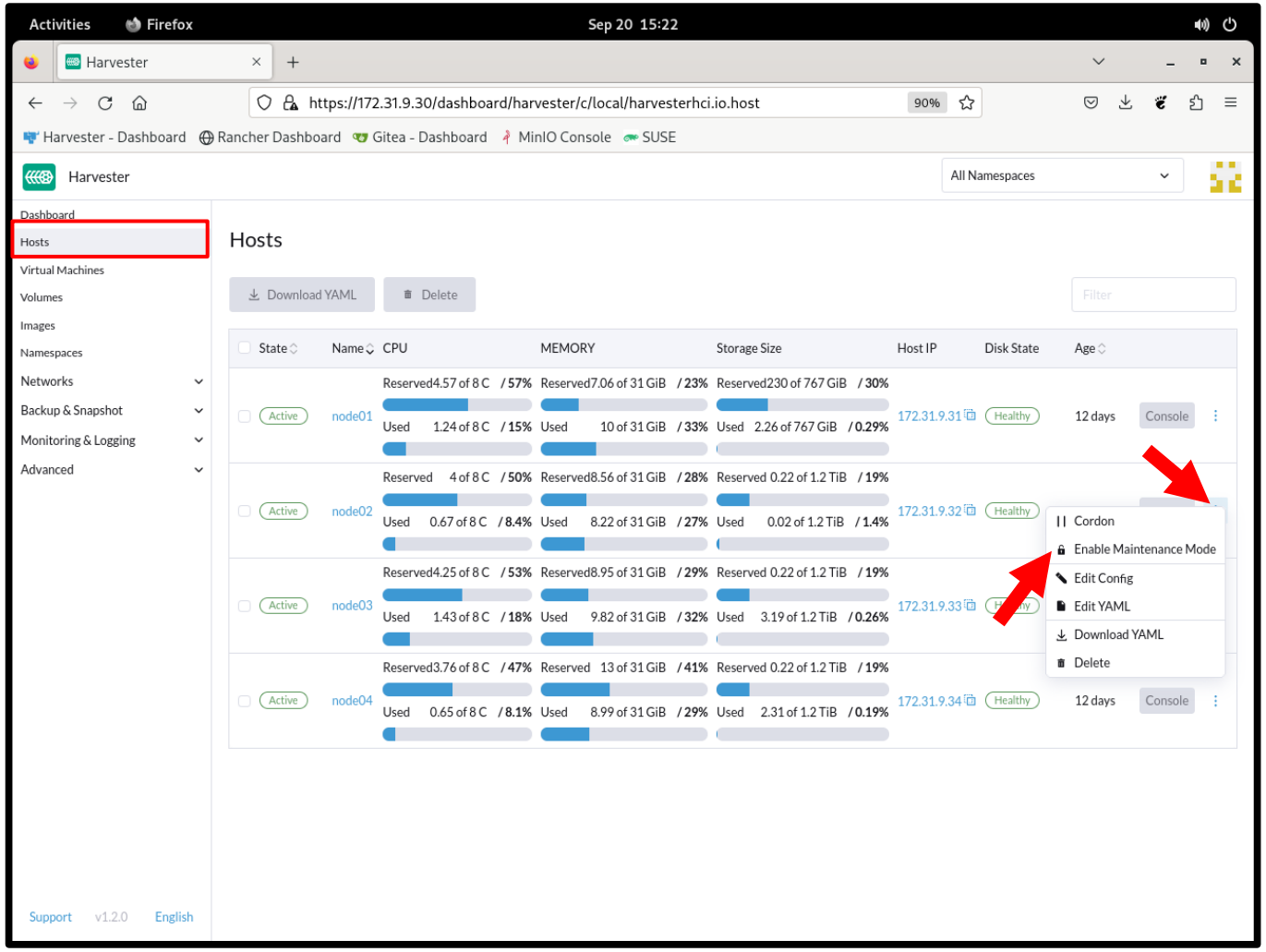
Maintenance Mode

- What is Maintenance Mode?
  - Node is marked as "unschedulable".
  - No new VMs will be run on the node.
  - All VMs currently running on the node will be live migrated to other nodes.

- Why use Maintenance Mode?
  - More in depth maintenance needs to be performed such as patching, reboot or hardware maintenance.

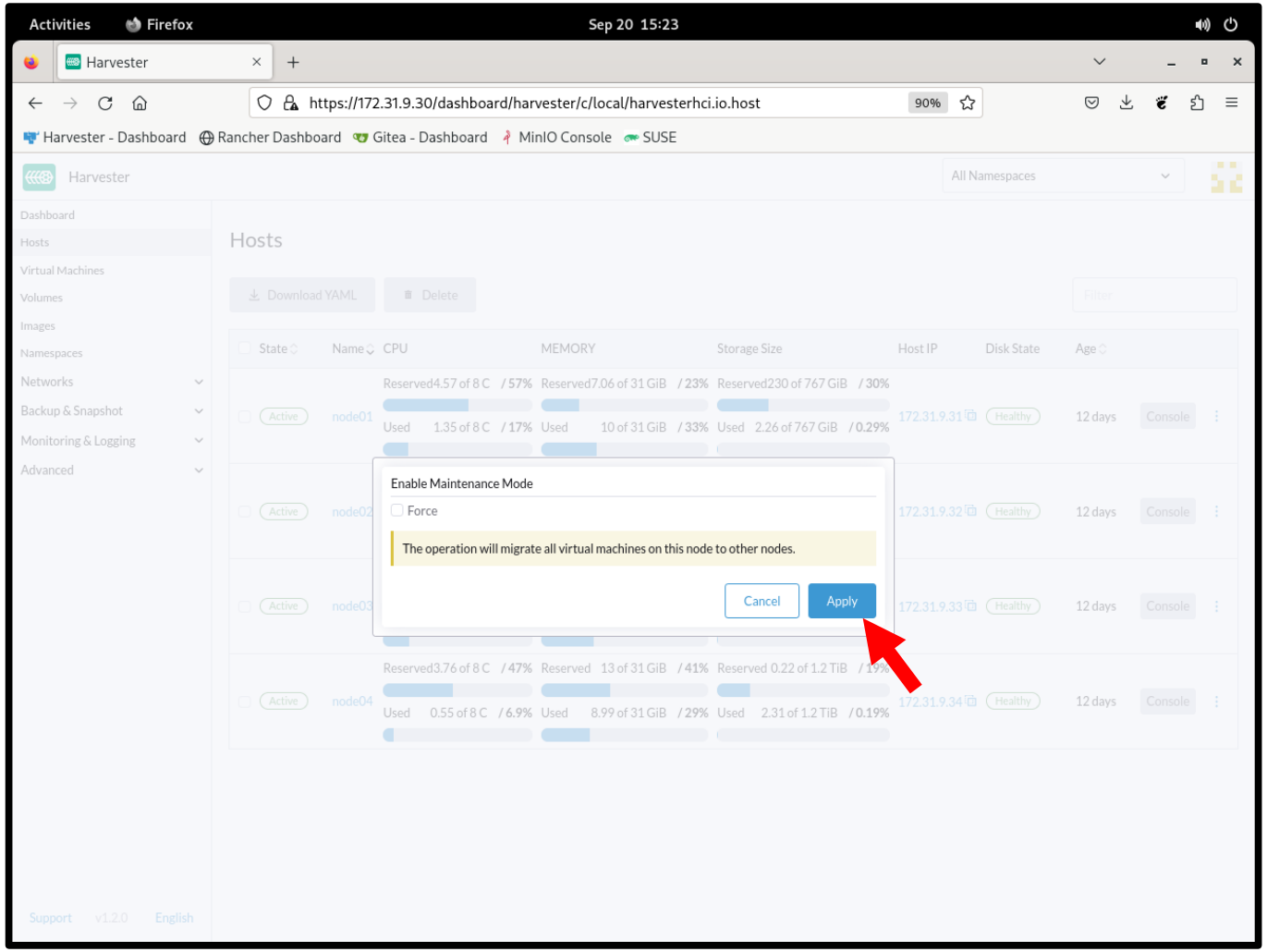
Maintenance Mode

- Navigate to the **Hosts** screen.
- Locate the node you wish to place in maintenance mode click the **3 dot** menu on the right of the node.
- Select: **Enable Maintenance Mode**



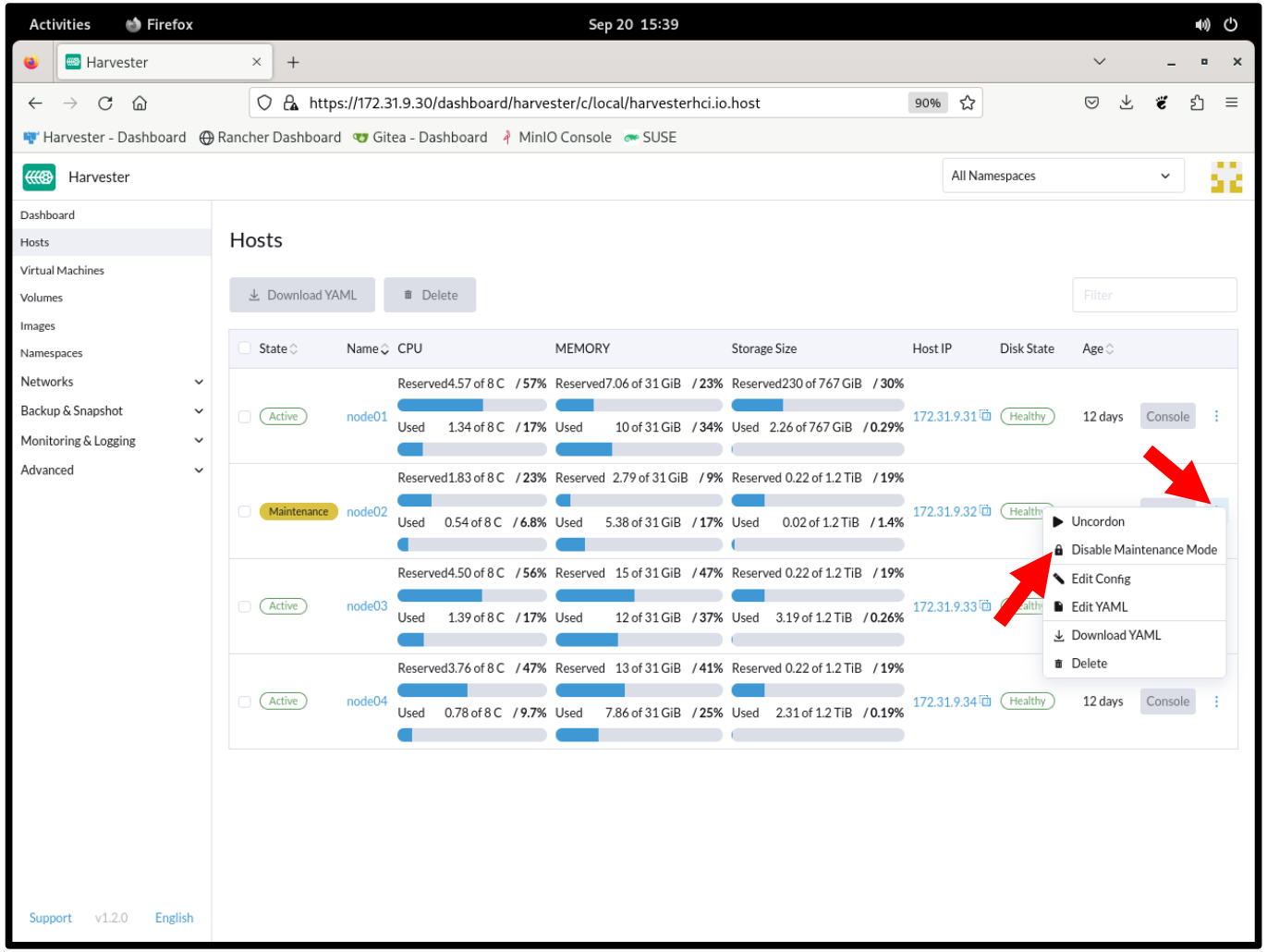
Maintenance Mode

- Confirm you want the node placed into Maintenance Mode.



Maintenance Mode

- To take the node out of maintenance mode, click the **3 dot** menu on the right of the node.
- Select: **Disable Maintenance Mode**



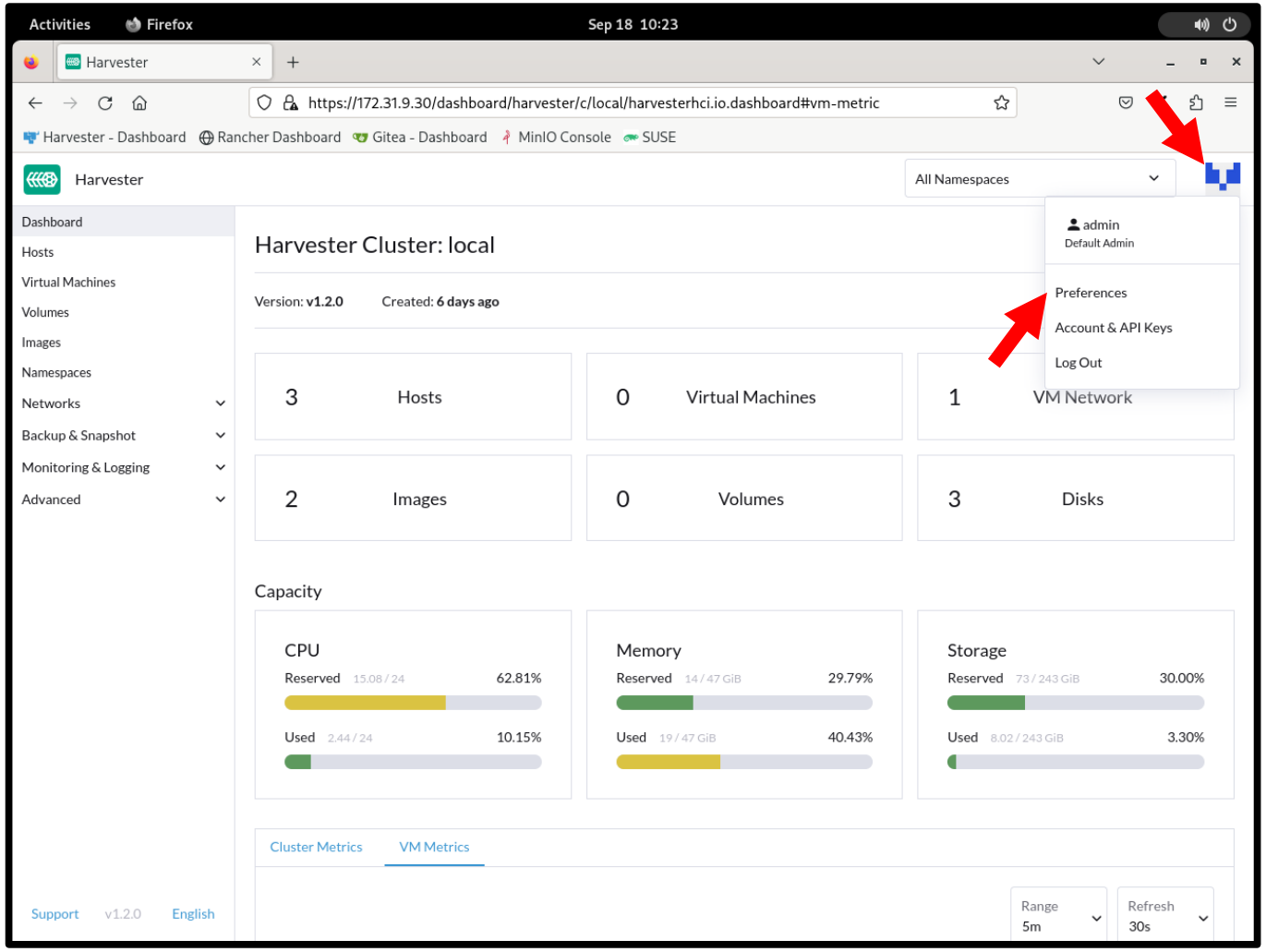
Additional Troubleshooting Tools in Harvester

Additional Troubleshooting Tools

- Additional troubleshooting tools can be enabled in the Harvester UI.
- These tools allow you to access the normally hidden embedded instance of Longhorn and Rancher directly.

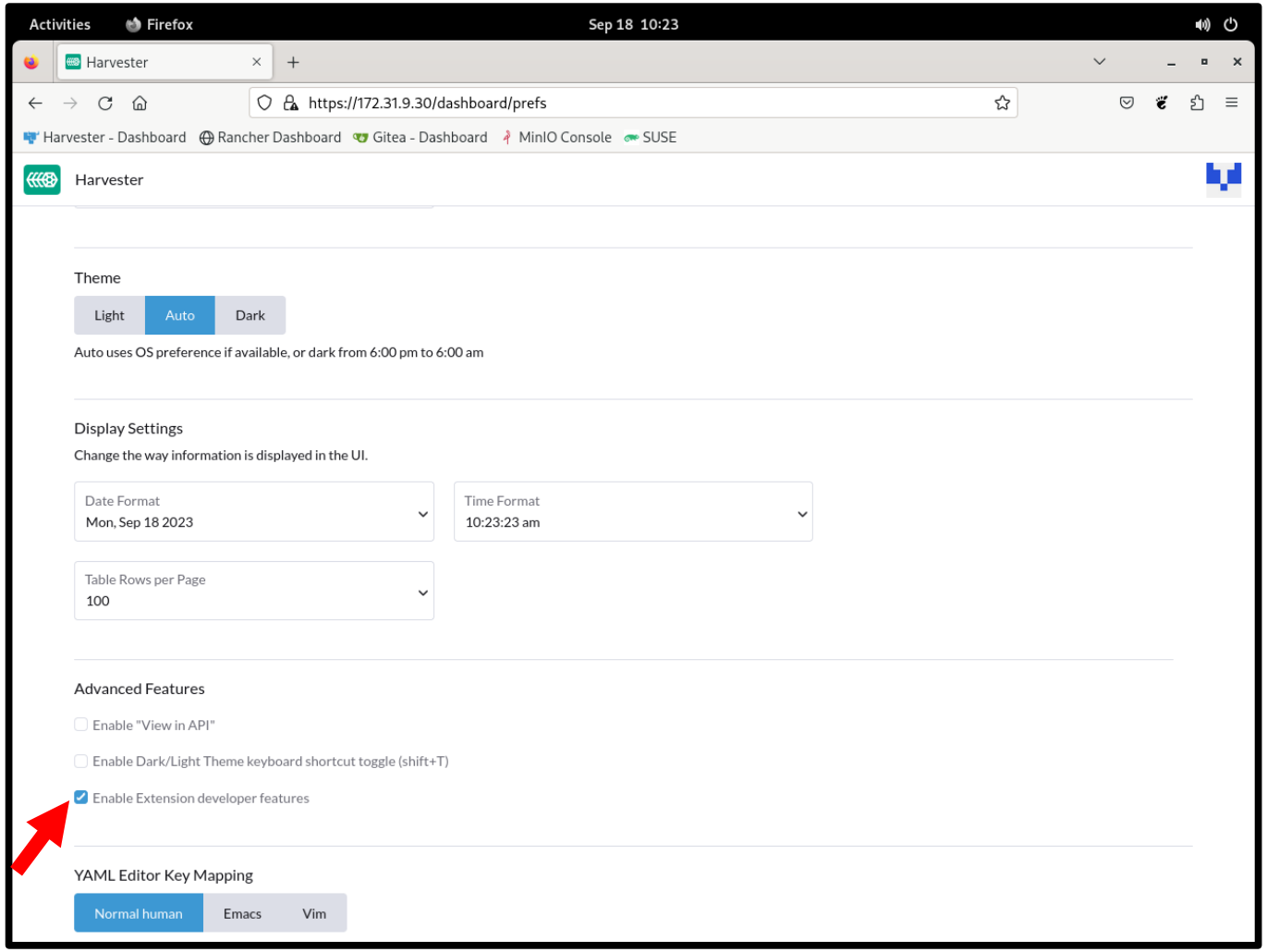
Enable Additional Troubleshooting Tools

- Click on the **User Icon** in the top right of the Harvester web UI and select **Preferences**.



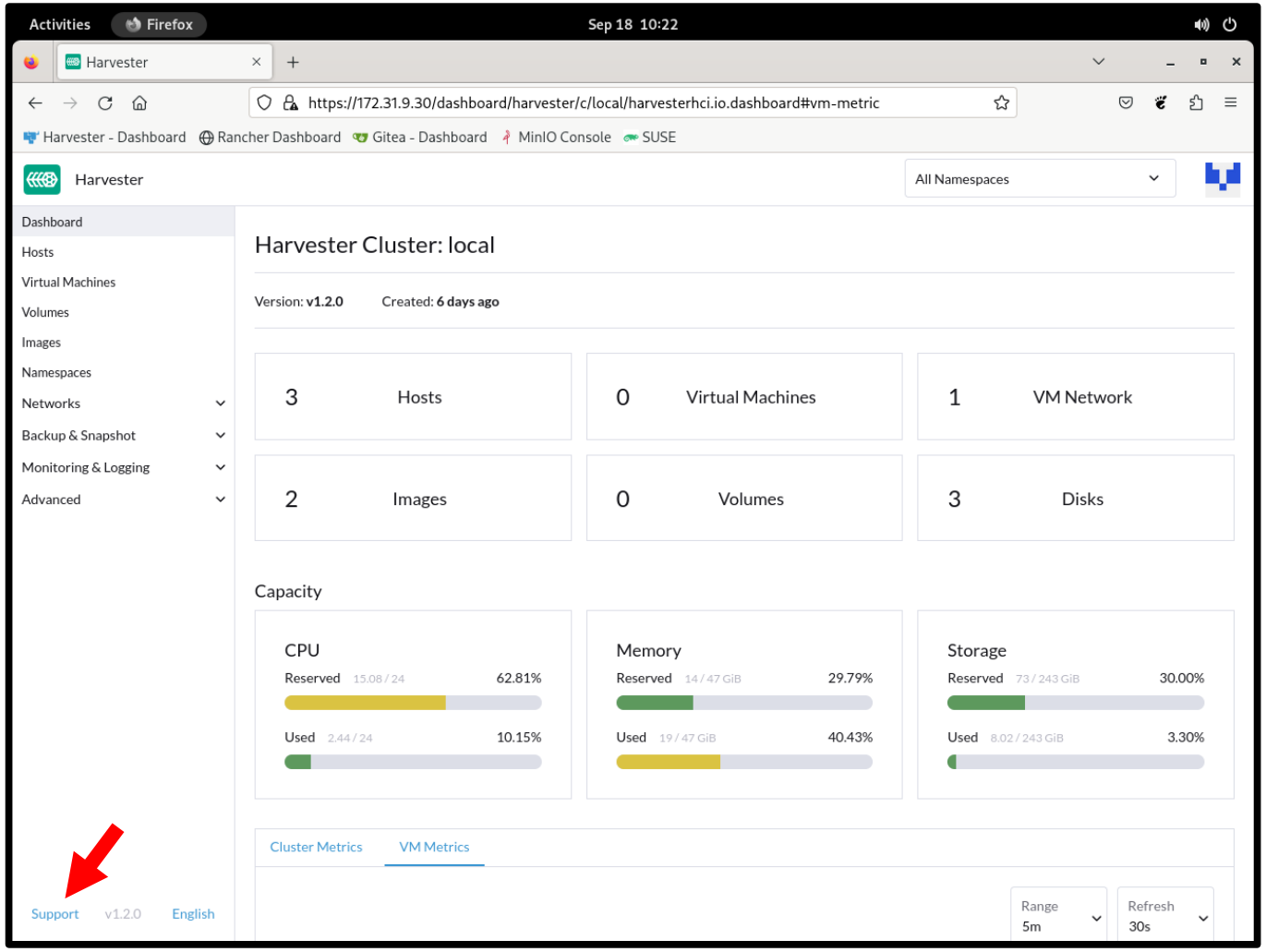
Enable Additional Troubleshooting Tools

- To enable access to the embedded Longhorn and Rancher UIs, on the **Preferences** screen, in the **Advanced Features** section, tick the check box next to **Enable extension developer features**.



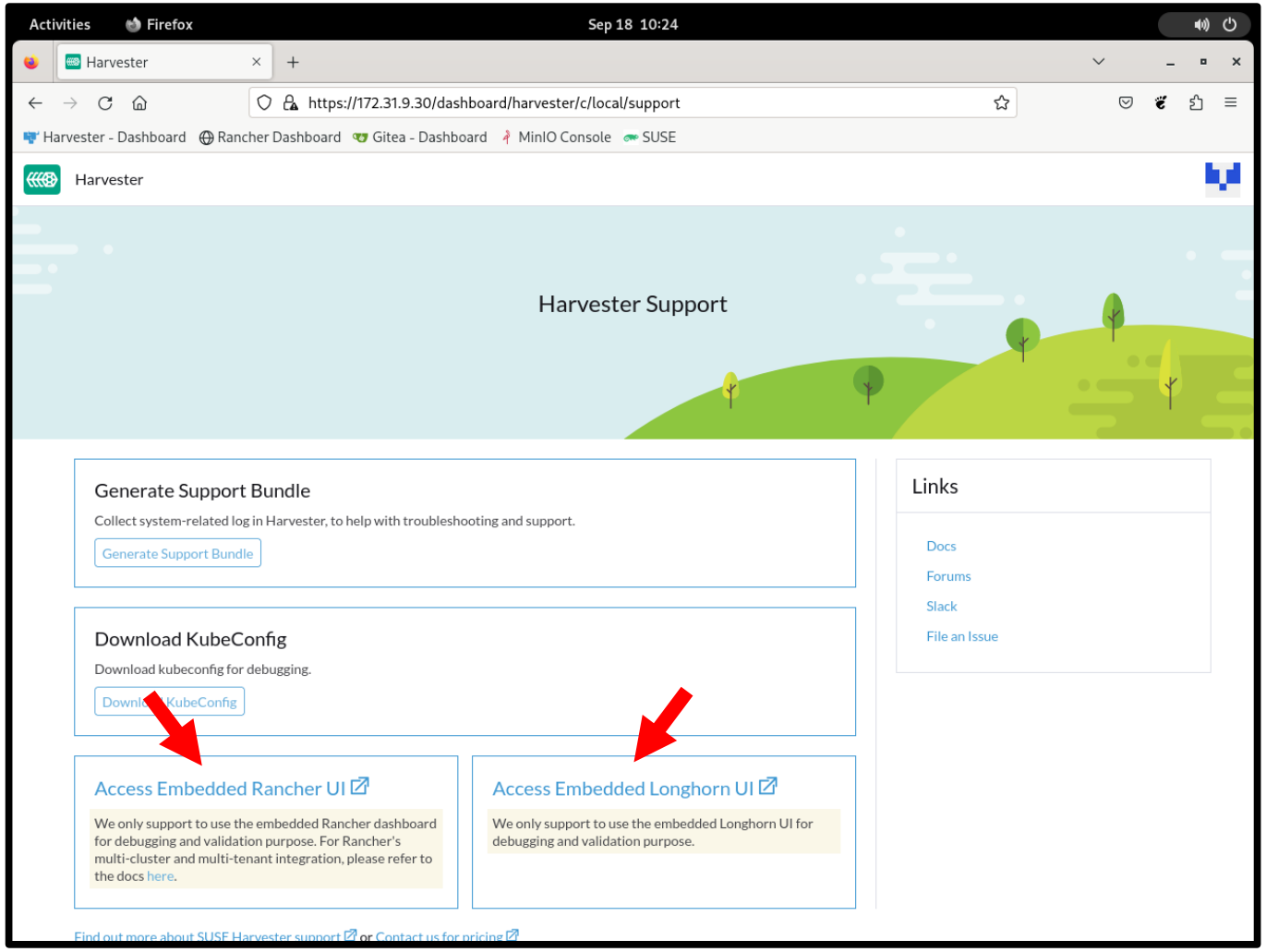
Enable Additional Troubleshooting Tools

- Click on the **Support** link in the bottom left of the Harvester web UI to go to the **Harvester Support** screen.



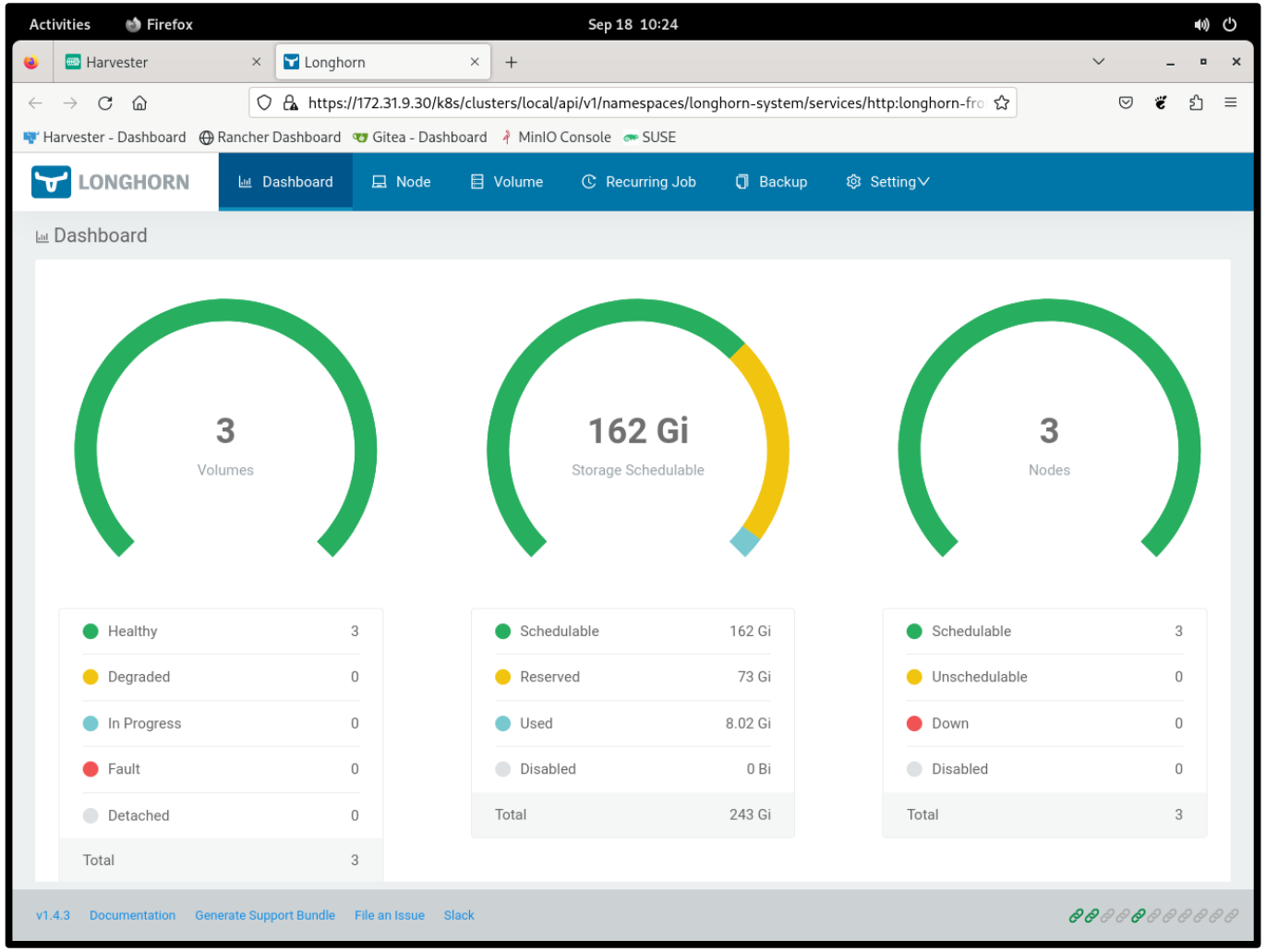
Embedded Longhorn Dashboard

- On the **Harvester Support** screen there are links to open the **Embedded Longhorn UI** and **Embedded Rancher UI**.



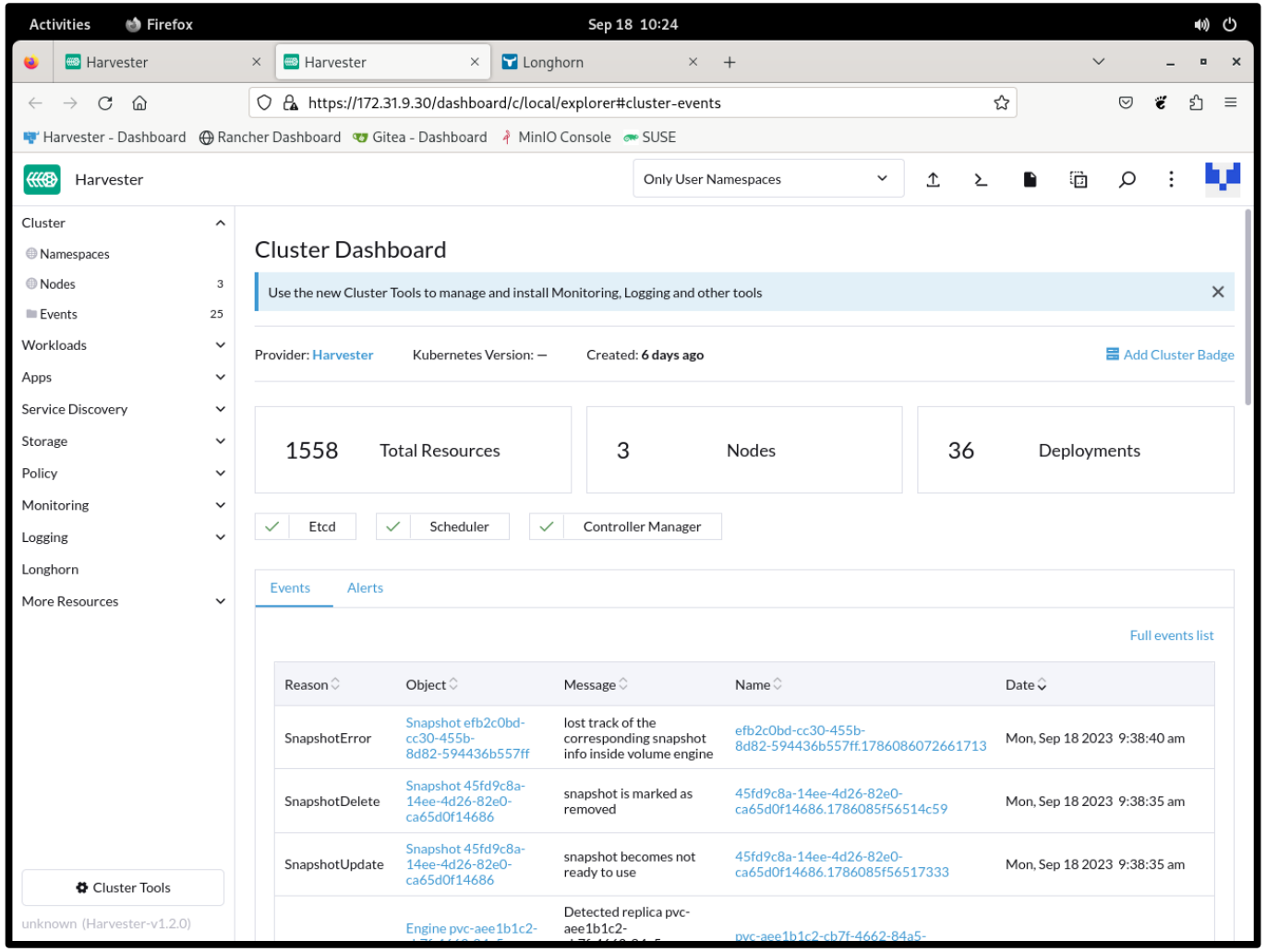
Embedded Longhorn Dashboard

- The Embedded Longhorn Dashboard can be used to further troubleshoot the Longhorn component of the Harvester cluster beyond what is available on the Harvester dashboard.



Embedded Rancher Dashboard

- The Embedded Rancher Dashboard can be used to further troubleshoot cluster resources running on the Harvester cluster from the Rancher perspective.
- This instance of Rancher should only be used to interact with the Harvester cluster itself and should not be used to launch user workloads.



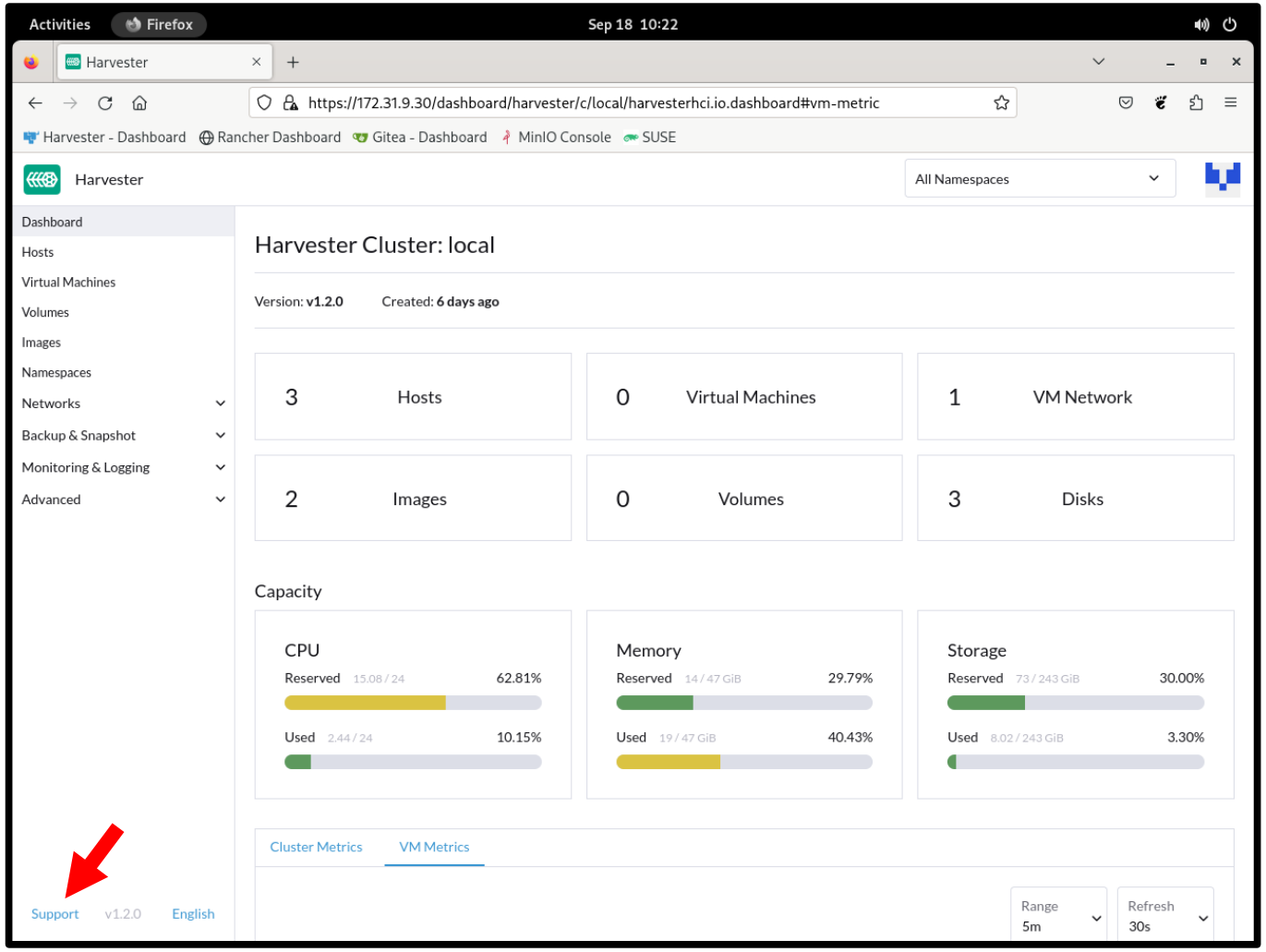
Support Bundles in Harvester

Harvester Support Bundles

- Harvester Support Bundles can be used by SUSE Support to get an understanding of the state of a cluster for troubleshooting.
- They contain logs, YAML files and other information about the state of the cluster.
- They can be saved locally or uploaded to Support for evaluation.

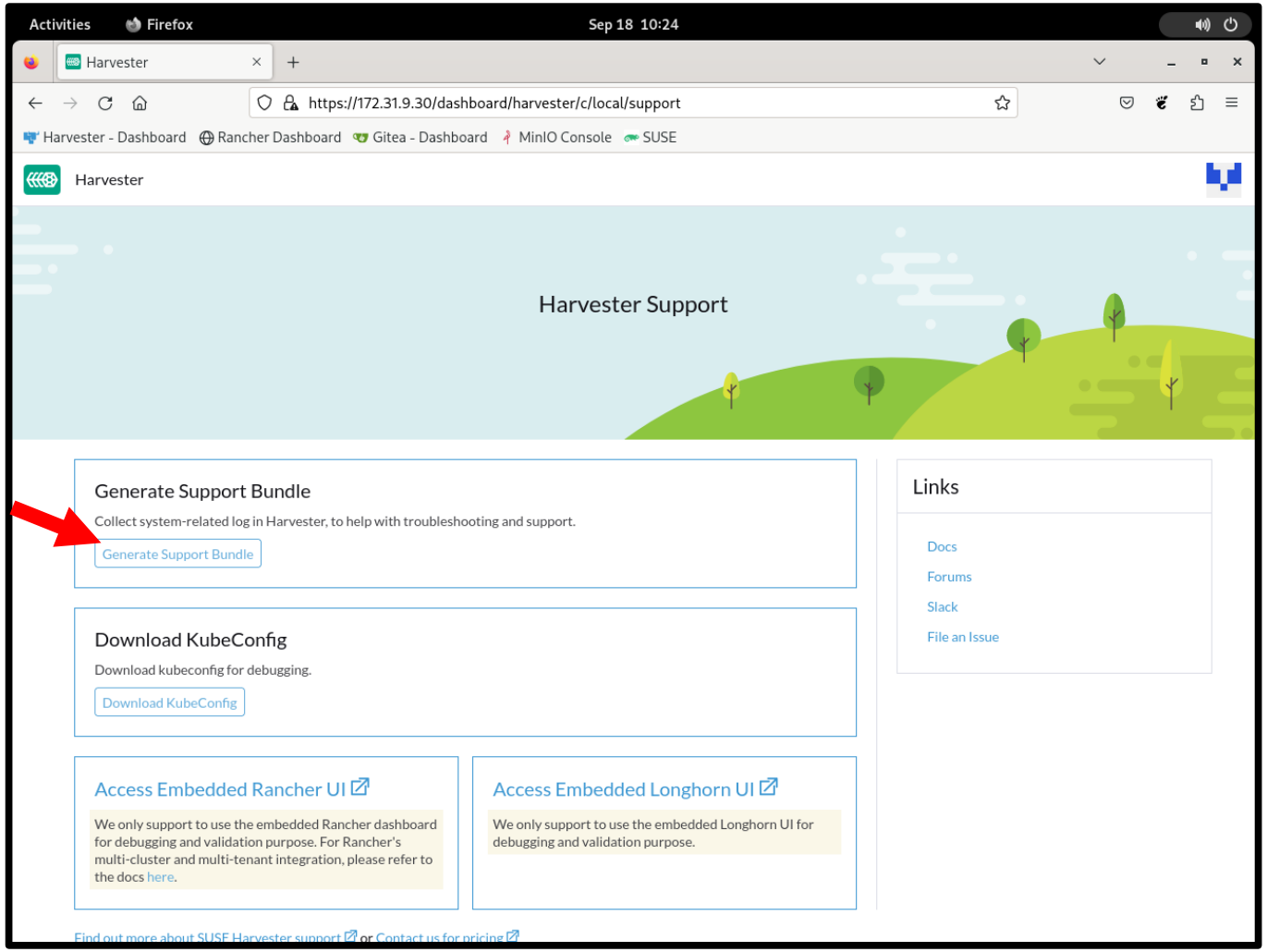
Support Bundle

- Click on the **Support** link in the bottom left of the Harvester web UI.



Support Bundle

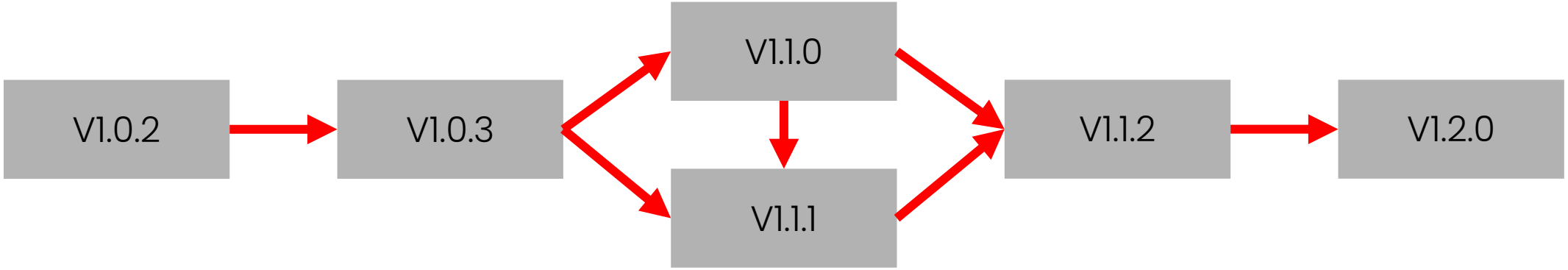
- On the Harvester Support page click on **Generate Support Bundle**.



Harvester Cluster Upgrade

Harvester Cluster Upgrade Path

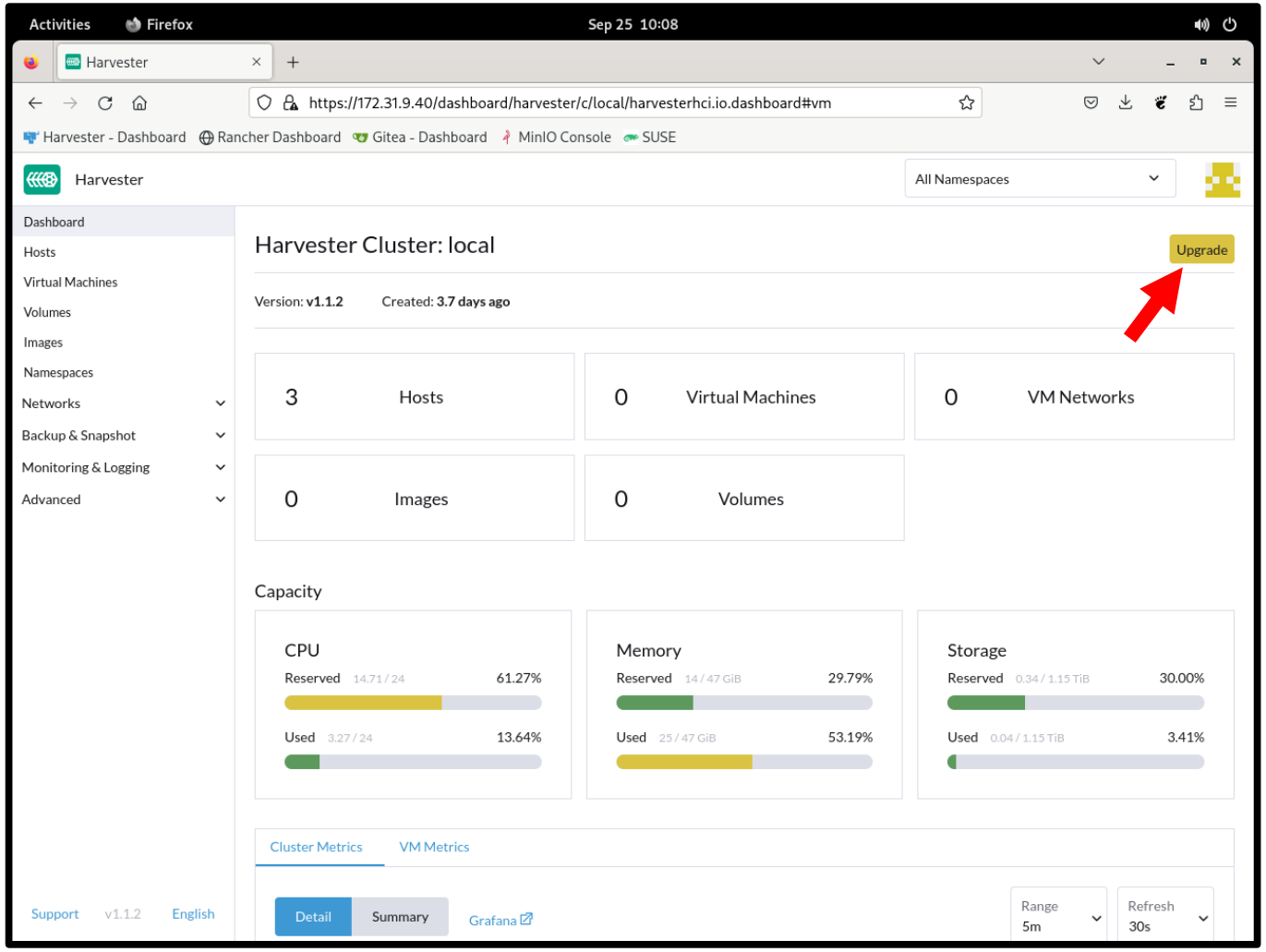
- Harvester versions must follow a specific upgrade path.
- You cannot skip over* Harvester releases when performing an upgrade.



* With the exception that v1.1.0 can be skipped when upgrading from v1.0.3.

Harvester Cluster Upgrade

- By default Harvester will periodically check if there are new compatible upgradable versions available.
- If there are, an **Upgrade** button will be displayed on the dashboard.

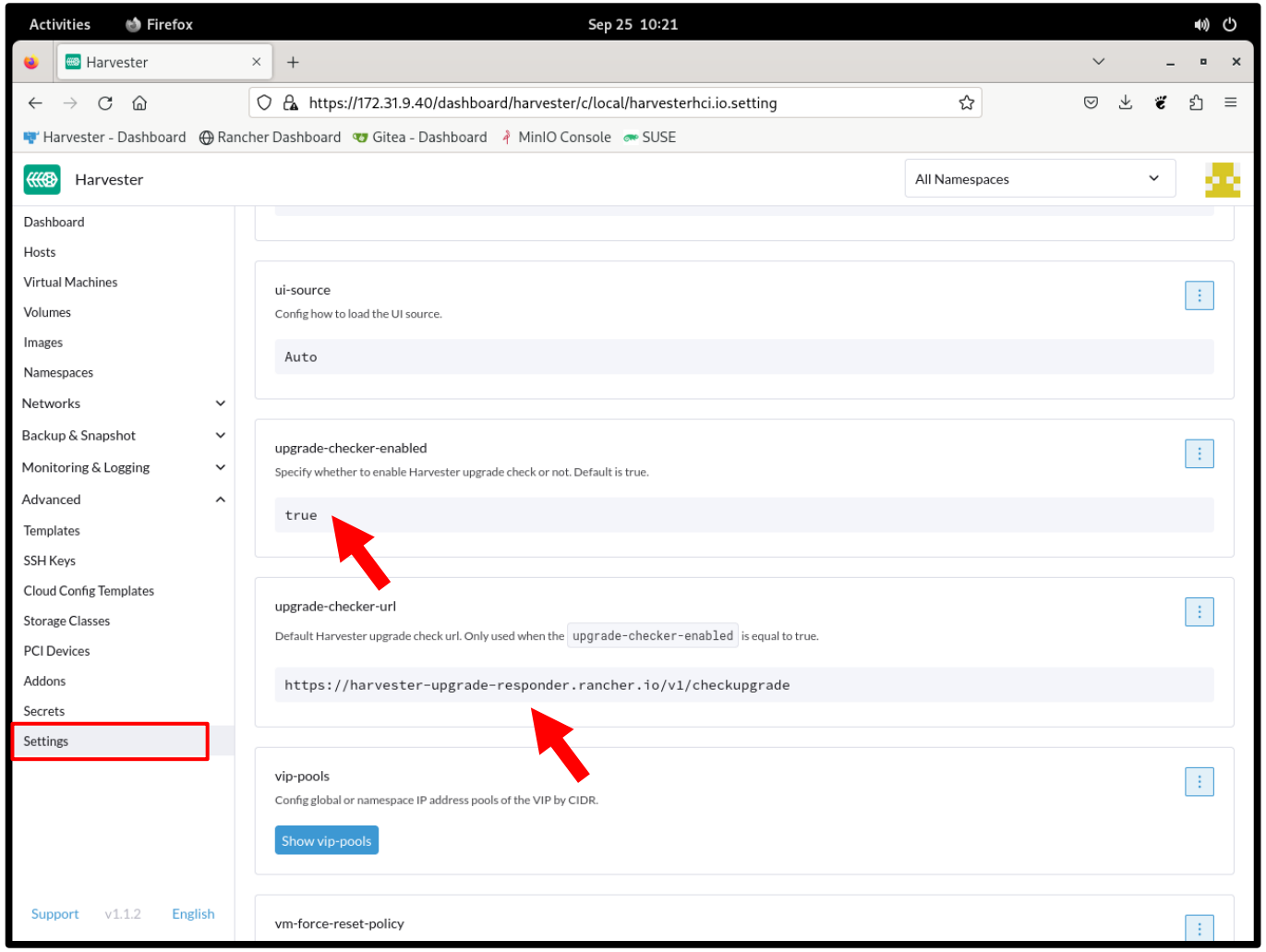


Configure Harvester Cluster Upgrade Checks

- On the **Settings** screen the upgrade checker can be enabled/disabled.

  The default is: enabled=true

- The URL the upgrade checker uses to check for available upgrades can also be updated.



Manually Enable Upgrades

- In air-gapped environments or other environments where access to the Internet is restricted, the upgrade can manually be enabled as follows:
  1) Download the installation ISO for the new version and save it on a local HTTP server.
  2) Download the **`version.yaml`** file for the new release.
  3) Update the **`isoURL`** line in the **`version.yaml`** file to point to the downloaded ISO.
  4) On a Harvester controller node, us **`kubectl`** to apply the **`version.yaml`** file

```
rmt:~ # curl -sLf \
 https://releases.rancher.com/harvester/v1.2.0/version.yaml \
 /var/lib/rmt/public/repo/Install/Harvester/version.yaml
```

version.yaml

```
apiVersion: harvesterhci.io/v1beta1
kind: Version
metadata:
  name: v1.2.0
  namespace: harvester-system
spec:
  isoChecksum: <SHA-512 checksum of the ISO>
  isoURL:
http://172.31.9.2/repo/Install/Harvester/harvester-
v1.2.0-amd64.iso
  releaseDate: '20230908'
```

```
node11:~ # kubectl create -f \
 http://rmt.example.com/repo/Install/Harvester/version.yaml
```

Harvester Cluster Upgrade Requirements

- The cluster should not be operated (create new VMs, upload images, etc.) during the upgrade process.
- Hardware should meet the preferred hardware requirements due to additional intermediate resources consumed during the upgrade.
- Each node must have at least 30GB of free space in the partition where `/usr/local/` resides.
- Cluster nodes must be in time sync.

Upgrade Pre-check Script

- Upgrade pre-check scripts are provided:

  https://github.com/harvester/upgrade-helpers/tree/main/pre-check

- On a Harvester **control node** , as **root**, download the script and make it executable:

  ```
  curl -sLf CHECK_SCRIPT_URL check.sh
  chmod +x check.sh
  ```

- Run the script.

  If all checks pass the upgrade can be performed.

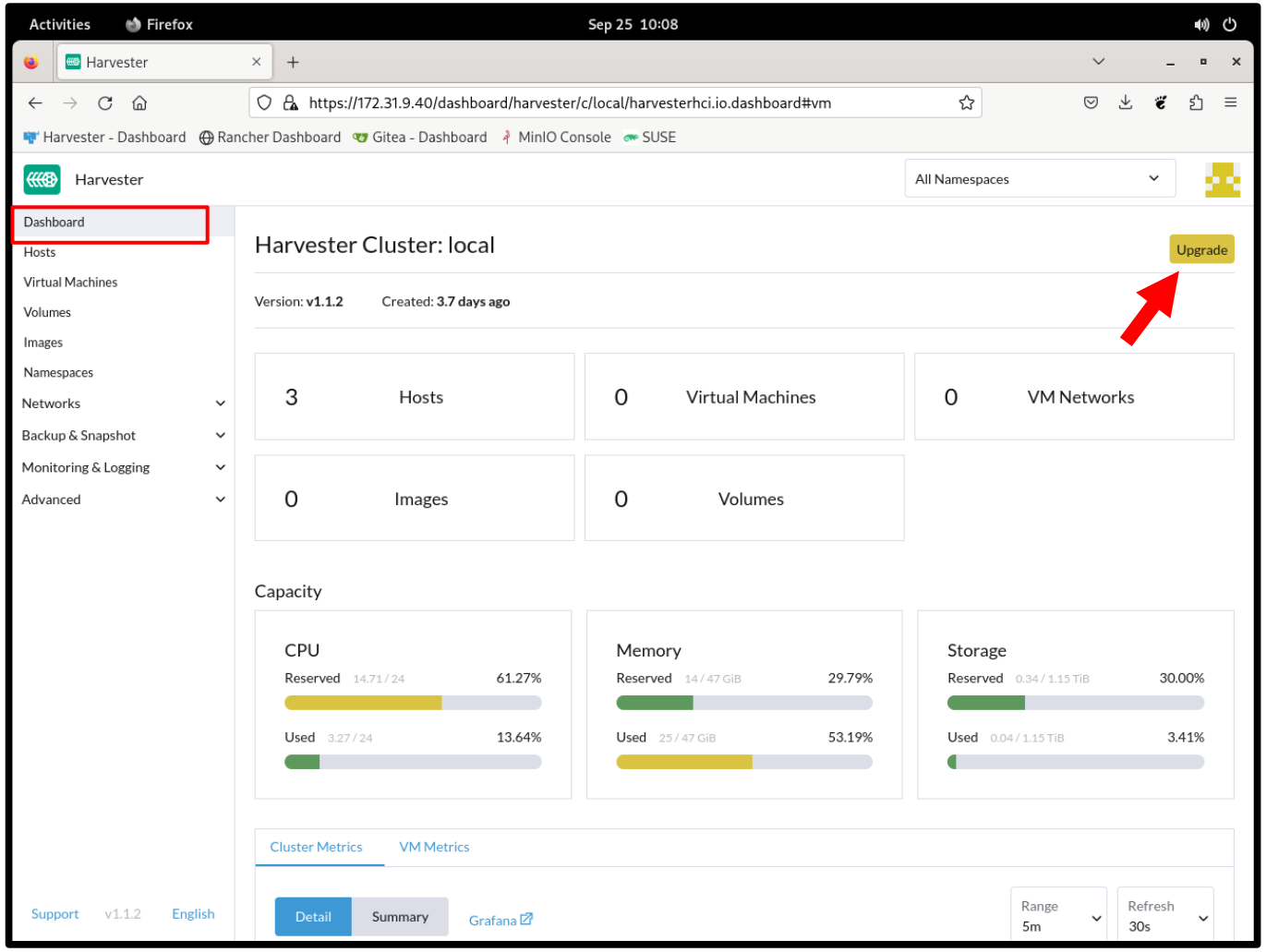
```
node11:~ # ./check.sh

>>> Check all bundles ready...
All Helm bundles are ready.
>>> Check the Harvester bundle is ready...
The Harvester bundle is ready.
>>> Check all nodes are ready...
All nodes are ready.
>>> Check the CAPI cluster is provisioned...
The CAPI cluster is provisioned.
>>> Check the CAPI machines count...
CAPI machine count is equal to node count.
>>> Check the CAPI machines are running...
The CAPI machines are provisioned.
>>> Check Longhorn volumes...
checking running engine pvc-165dc95f-bfb0-4b34-9015-4113e5017d20-e-
568bf087...
Volume pvc-165dc95f-bfb0-4b34-9015-4113e5017d20 is healthy.
checking running engine pvc-50206083-653c-414c-a5fc-1d91618d21e3-e-
c7f0a32c...
Volume pvc-50206083-653c-414c-a5fc-1d91618d21e3 is healthy.
checking running engine pvc-82c1d477-27cb-47dc-bd34-f0038b29ea3e-e-
abb1dcd7...
Volume pvc-82c1d477-27cb-47dc-bd34-f0038b29ea3e is healthy.
All volumes are healthy.
>>> Check stale Longhorn volumes...
Checking volume longhorn-system/pvc-165dc95f-bfb0-4b34-9015-
4113e5017d20...
Checking volume longhorn-system/pvc-50206083-653c-414c-a5fc-
1d91618d21e3...
Checking volume longhorn-system/pvc-82c1d477-27cb-47dc-bd34-
f0038b29ea3e...
There is no stale Longhorn volume.
>>> Check error pods...
All pods are OK.
All nodes have more than 30GB free space.

All checks pass.
```

Perform a Harvester Cluster Upgrade

- On the Dashboard screen click on the **Upgrade** button to start the upgrade.



Section Summary

Section Summary

- Place cluster nodes in to states where maintenance can be performed
  - Node cordoning and maintenance mode can be used to place cluster nodes into a state where maintenance can be performed.
  - Node cordoning prevents new VMs from being scheduled on a node but allows already running VMs to continue running on the node.
  - Maintenance mode both prevents new VMs from being scheduled on a node and clears the node of any currently running VMs by live migrating them to other nodes.

Section Summary

- Enable and access additional tools used for troubleshooting the cluster
  - Access to the dashboard of the embedded Longhorn can be enabled allowing direct access to the Longhorn cluster.
  - Access to the embedded Rancher Manager can be enabled allowing direct access to Rancher Manager running the Harvester cluster components.
  - Support bundles can be created that gather all basic troubleshooting data together enabling easier access to cluster troubleshooting

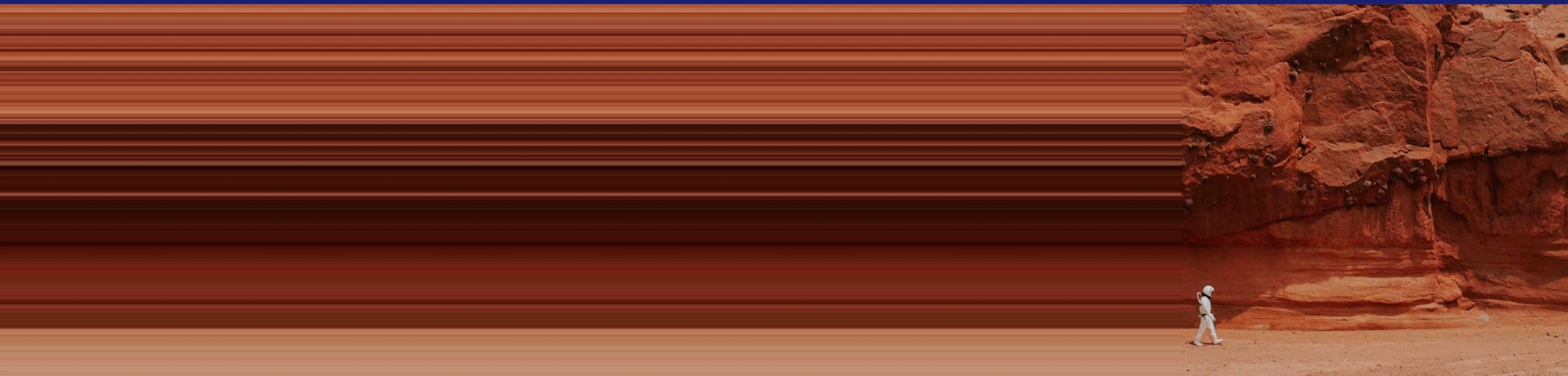
Section Summary

- Upgrade a Harvester cluster
  - Upgrades to a Harvester cluster are enabled when a new version is released.
  - Harvester versions can only be upgraded following a specific upgrade path.
  - A Harvester upgrade can be performed without Internet access by downloading the installation ISO and a `version.yaml` file.
  - The cluster should not be operated (creating new VMs, etc.) during the upgrade process

SUSE

Section: 9

Harvester and Rancher Integration



Learning Objectives

At the end of this section you will be able to:

- Integrate Harvester with Rancher.
- Manage and use Harvester clusters with Rancher users.
- Use Rancher to deploy Kubernetess clusters into VMs running on Harvester.

Topics

1 Harvester and Rancher Integration

2 Rancher and Harvester Users

3 Cloud Credentials

4 Cluster Deployment with Rancher onto Harvester

5 Application Deployment with Rancher onto Clusters Running on Harvester

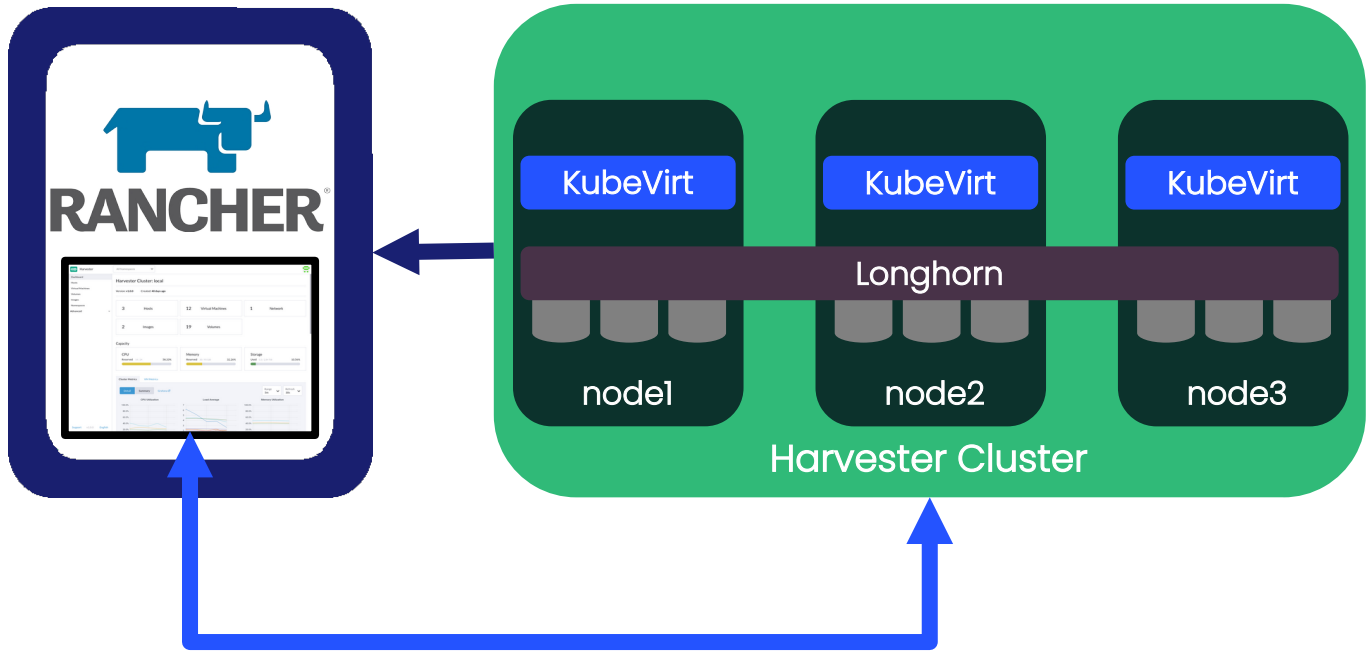
Harvester and Rancher Integration

Harvester → Rancher Integration

- Harvester clusters can be imported into Rancher to provide a virtualization platform for Kubernetes (K8s) cluster deployment.
- This provides an on-prem option for Kubernetes cluster deployment that mirrors functionality available in cloud providers.
- The on-prem option links the flexibility and scalability of the Harvester hyperconverged infrastructure with the power and feature set of Rancher's Kubernetes management.

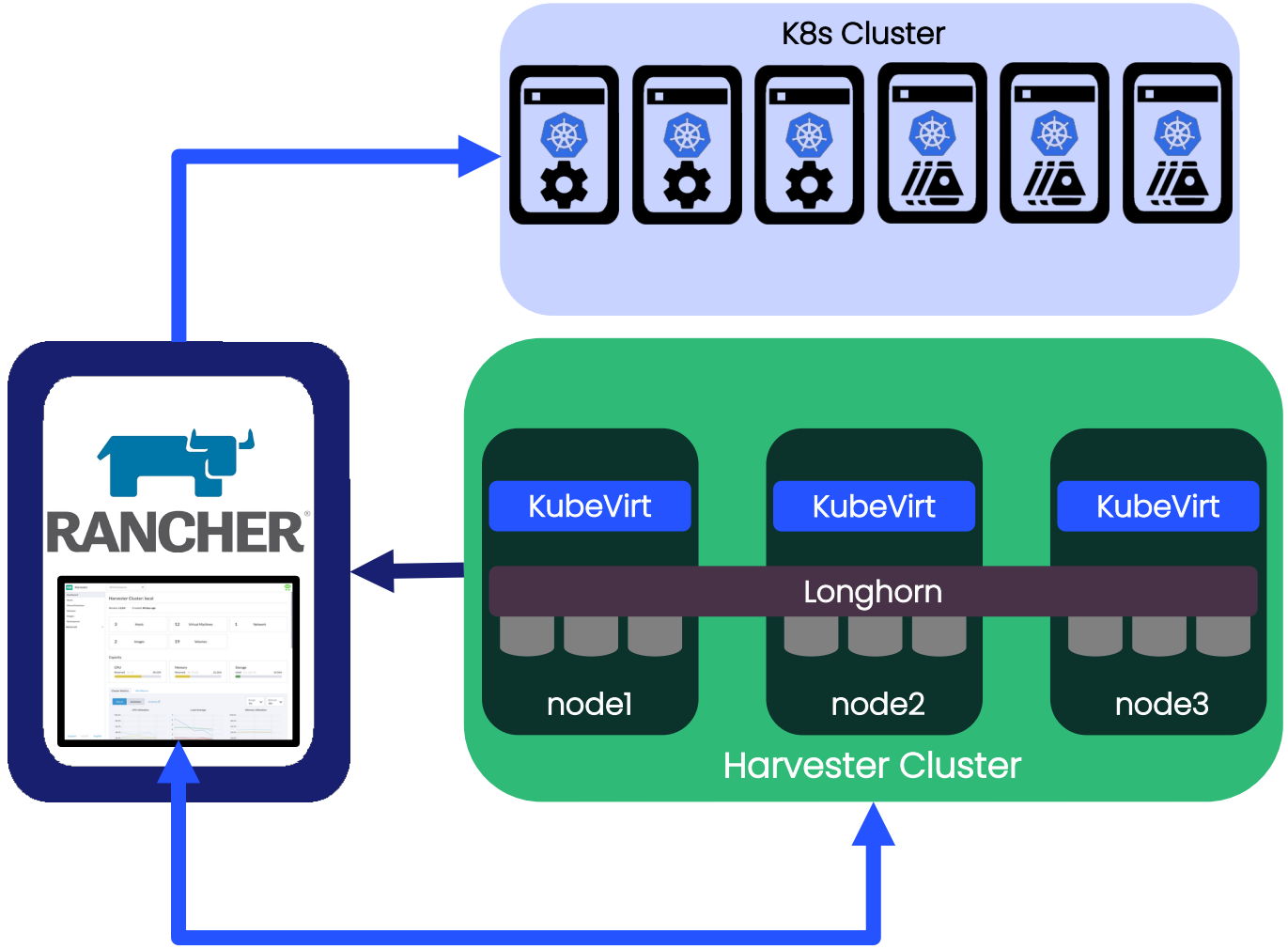
Import Harvester Clusters

- Harvester clusters are imported into Rancher as a special type of cluster, capable of virtualization and appear in the **Virtualization Management** section of the Rancher UI.
- The Harvester cluster web UI can be accessed from within the Rancher web UI.



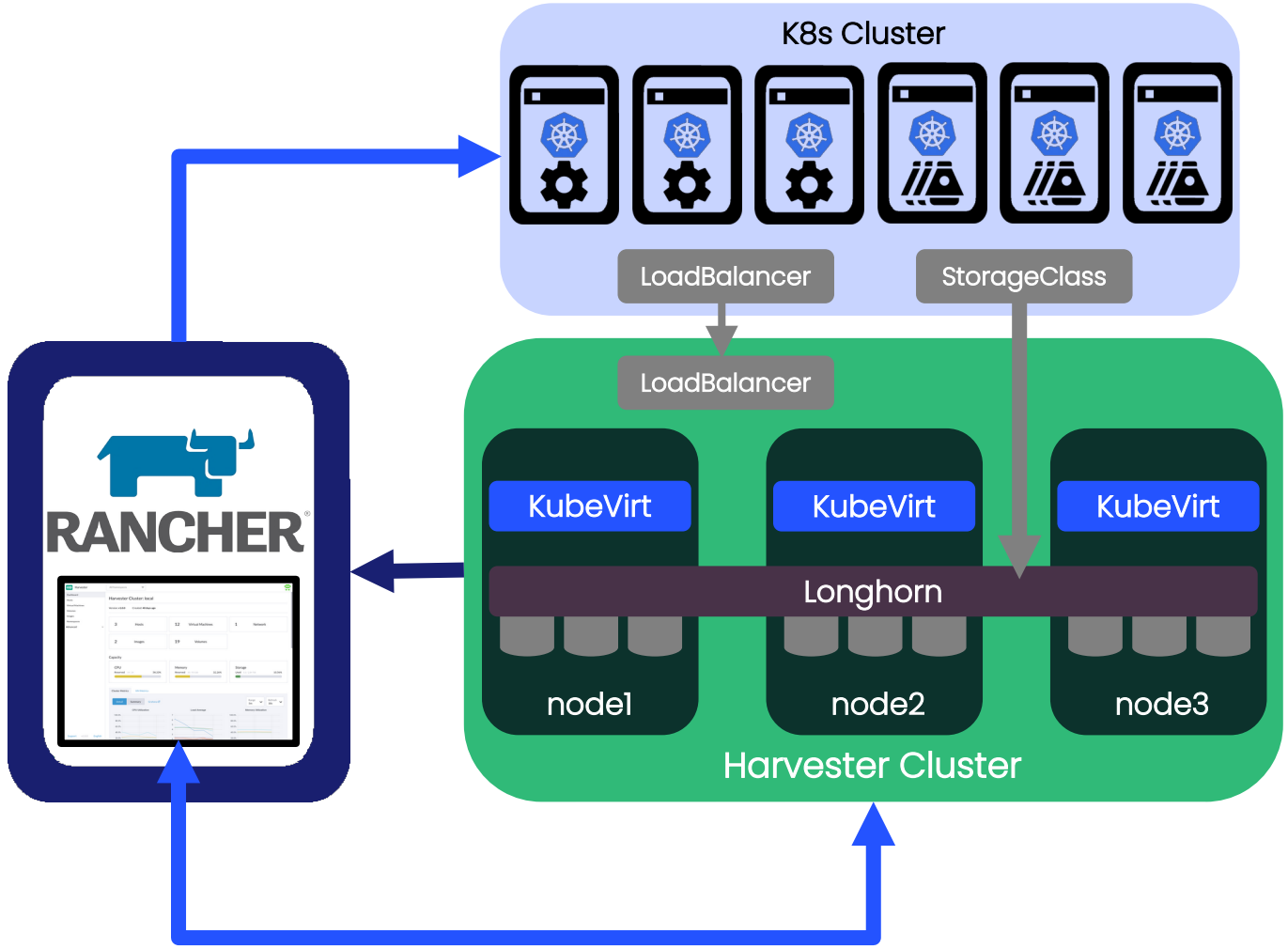
Downstream Clusters on Harvester

- Rancher can launch new downstream K8s clusters on the Harvester clusters similarly to how it can launch downstream K8s clusters on other virtualization and cloud platforms.
  - Provision new VMs as cluster nodes
  - Deploy K8s cluster on the nodes
  - Manage the K8s cluster lifecycle



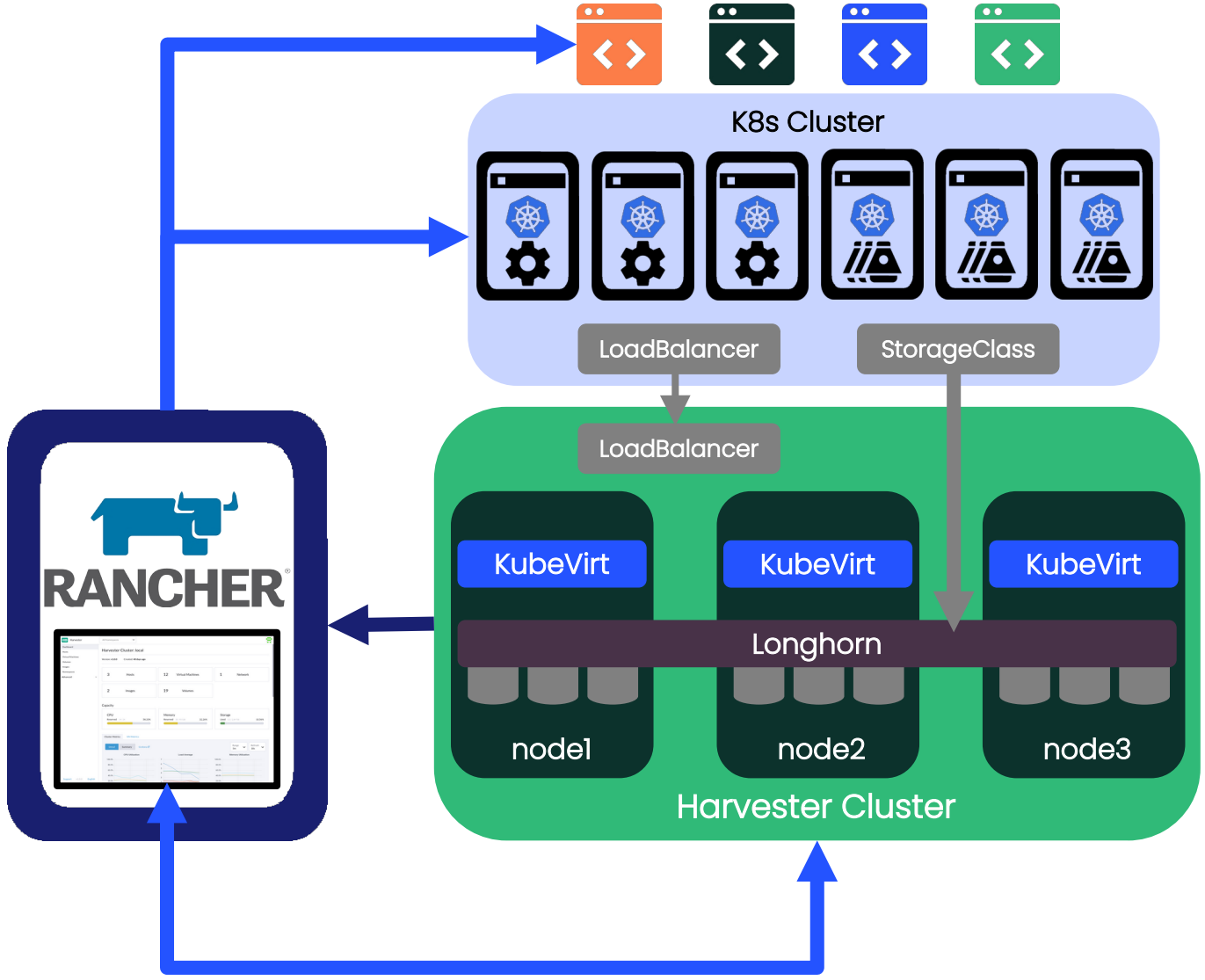
Harvester Cloud Provider Integration

- A **Harvester Cloud Provider** is installed on the downstream K8s clusters running on Harvester to provide further integration such as the ability to use a Harvester **load balancer**.
- A **Harvester CSI Driver** is installed on the downstream K8s clusters that configures a **StorageClass** exposing the **Longhorn** cluster embedded in the Harvester cluster for use by the K8s clusters.



Manage Clusters Running on Harvester

- Rancher can manage K8s clusters running on Harvester in the same manner as any other downstream cluster.
  - Rancher can manage the lifecycle of the downstream K8s clusters.
  - Rancher can launch applications on the downstream K8s clusters.



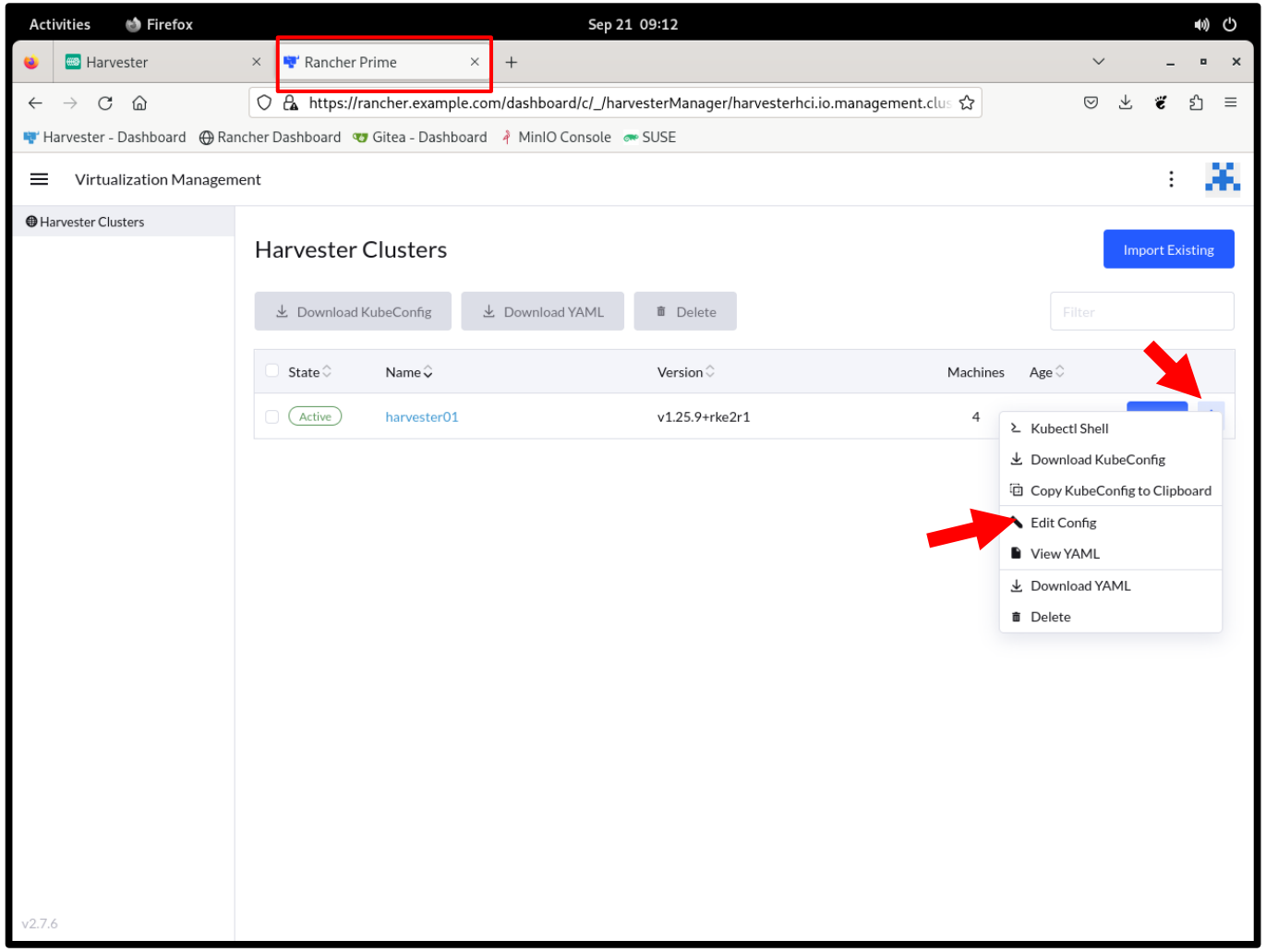
Rancher and Harvester Users

Rancher and Harvester Users

- Additional users for Harvester clusters are defined in Rancher Manager.
- These users are assigned as "Members" of the Harvester cluster(s) with specific roles referred to as "Permissions".
- Users access the Harvester cluster dashboard through the Rancher dashboard.

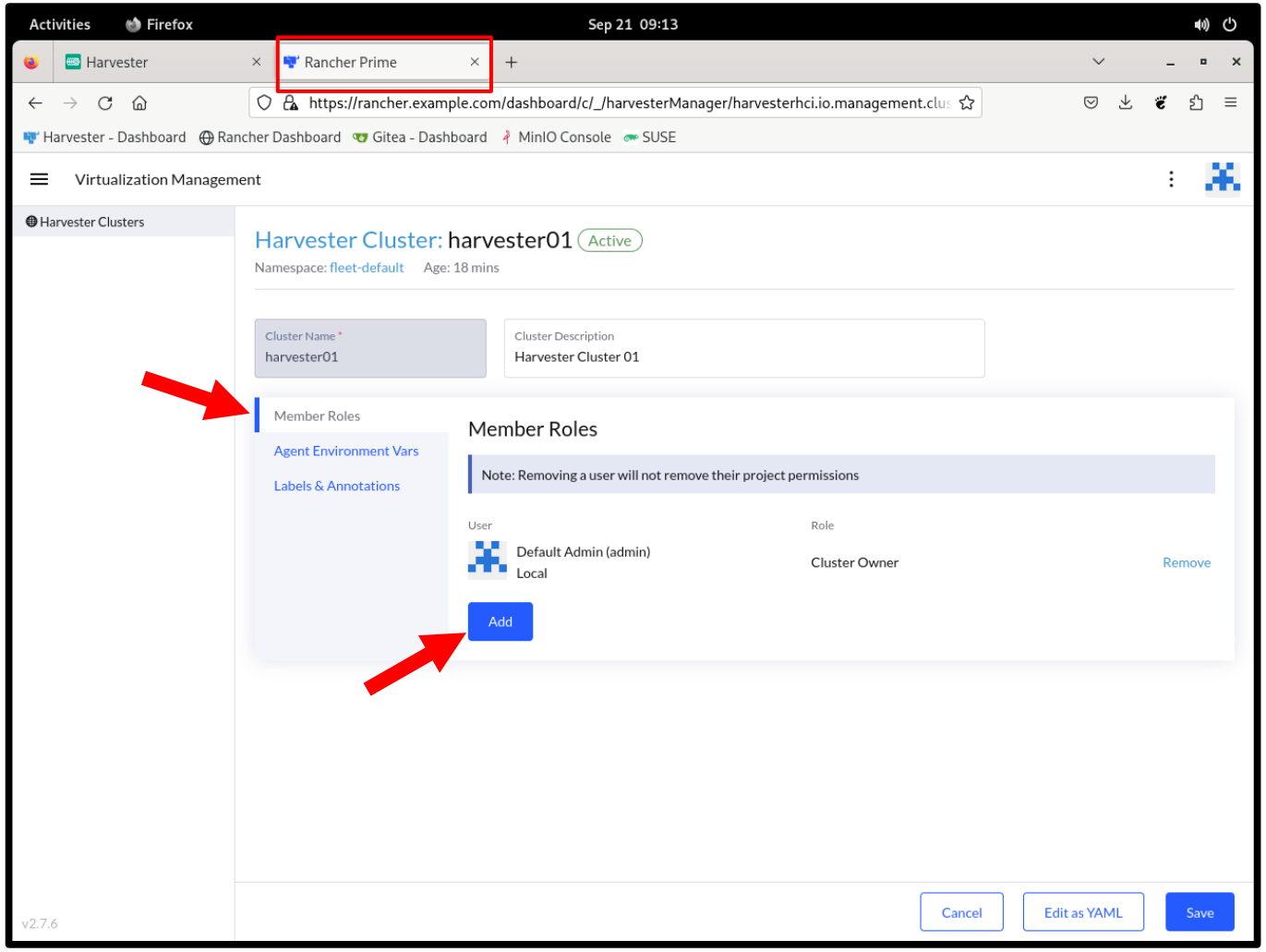
Rancher: Harvester Cluster Config

- Users are added to a Harvester cluster from Rancher starting on the **Virtualization Management → Harvester Clusters** screen.
- Select the **3 dot** menu and then **Edit Config**.



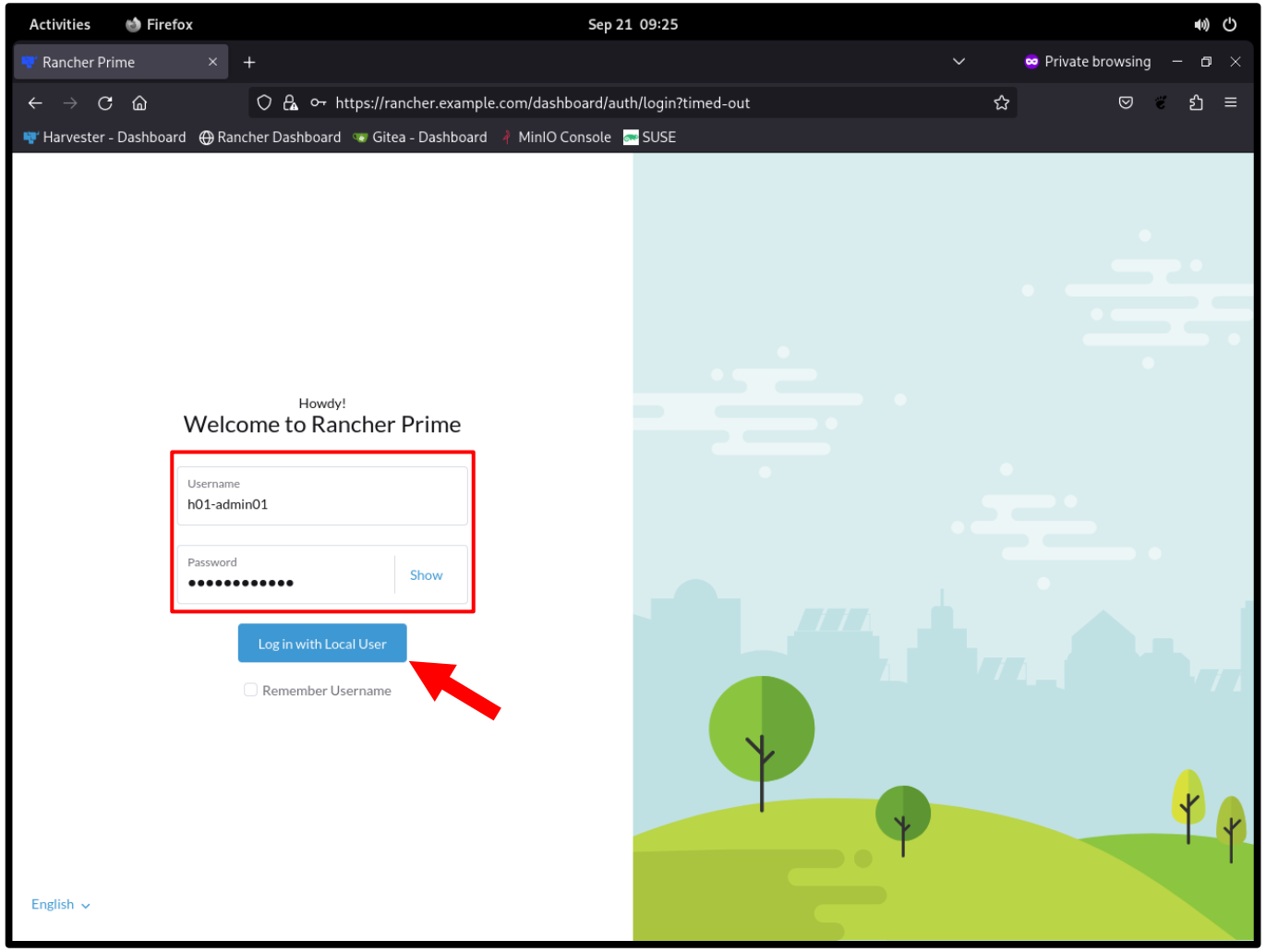
Rancher: Add Rancher Users to a Harvester Cluster

- On the **Harvester Cluster: <cluster_name>** screen the users are added on the **Member Roles** tab.
- To add Rancher users to the Harvester cluster click: **Add**



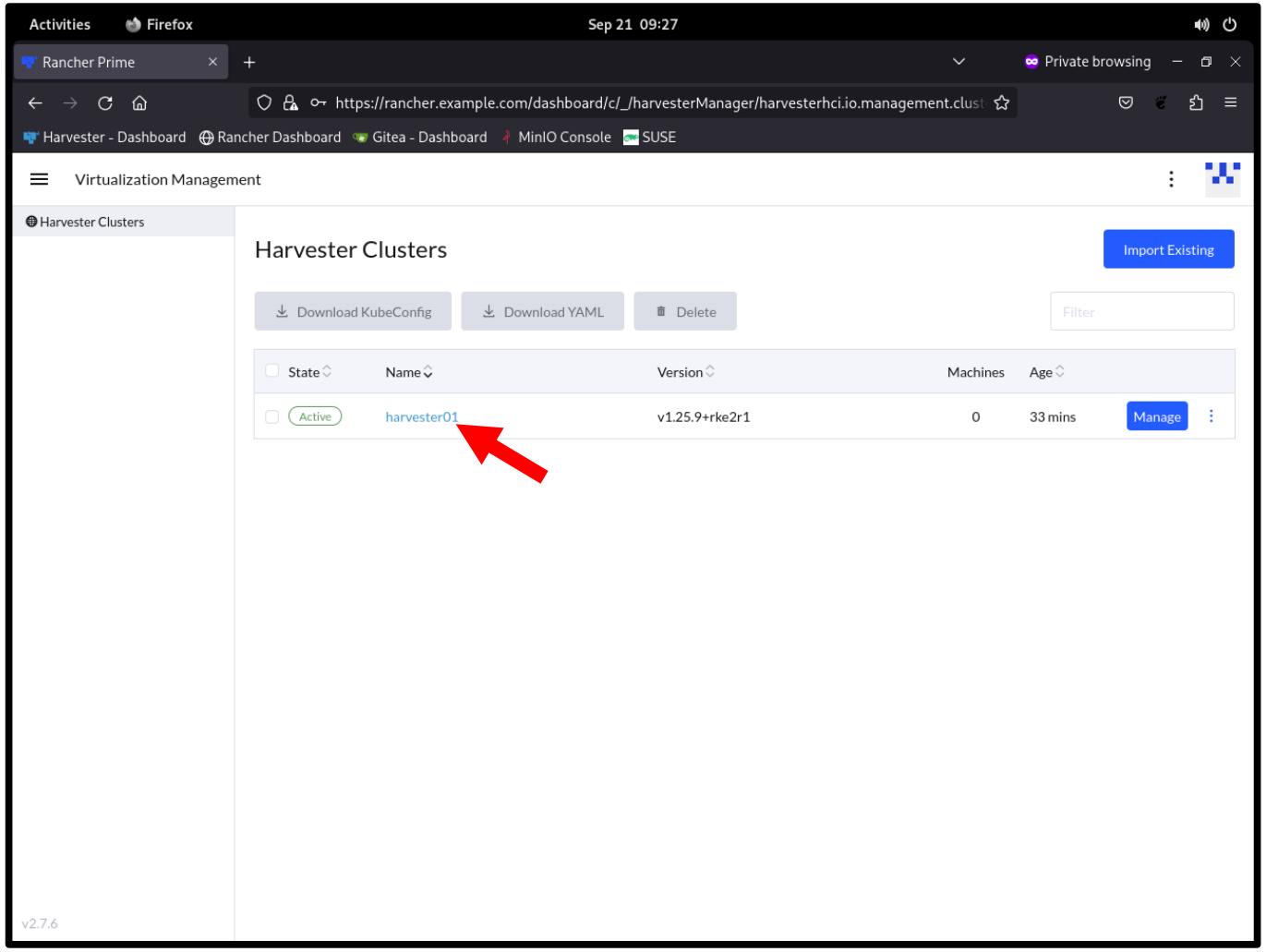
Rancher: Harvester User Login

- When the Rancher users have been added as Members of the Harvester cluster, log into Rancher as one of these Rancher users.



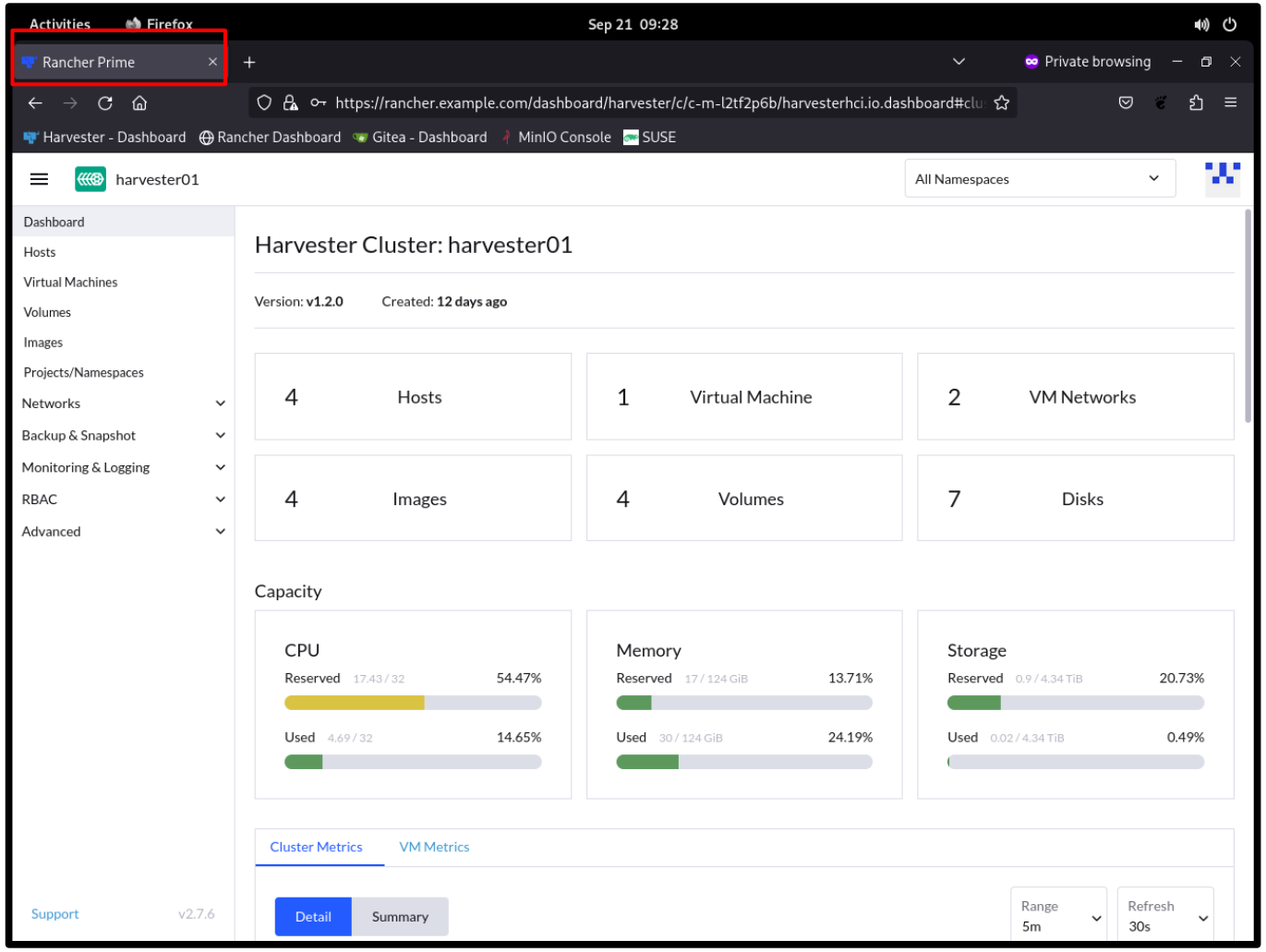
Rancher: Harvester Clusters

- On the **Virtualization Management → Harvester Clusters** screen the list of Harvester clusters that are available to manage are displayed.



Rancher: Harvester Dashboard

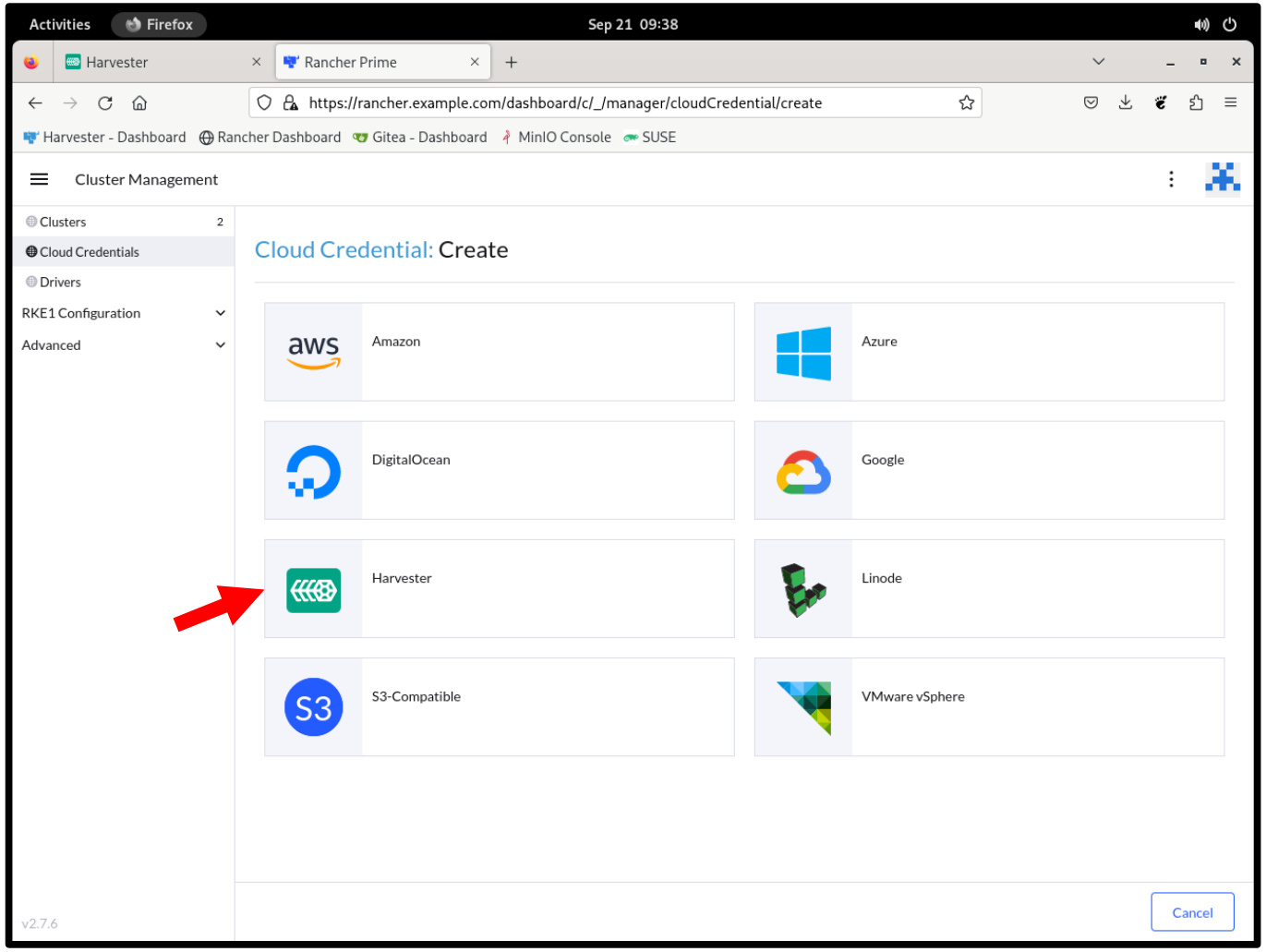
- You will be taken to a Harvester Dashboard.
- Notice that this **Harvester Dashboard** is being accessed through the **Rancher Dashboard** (Rancher is displayed on the tab).



Cloud Credentials

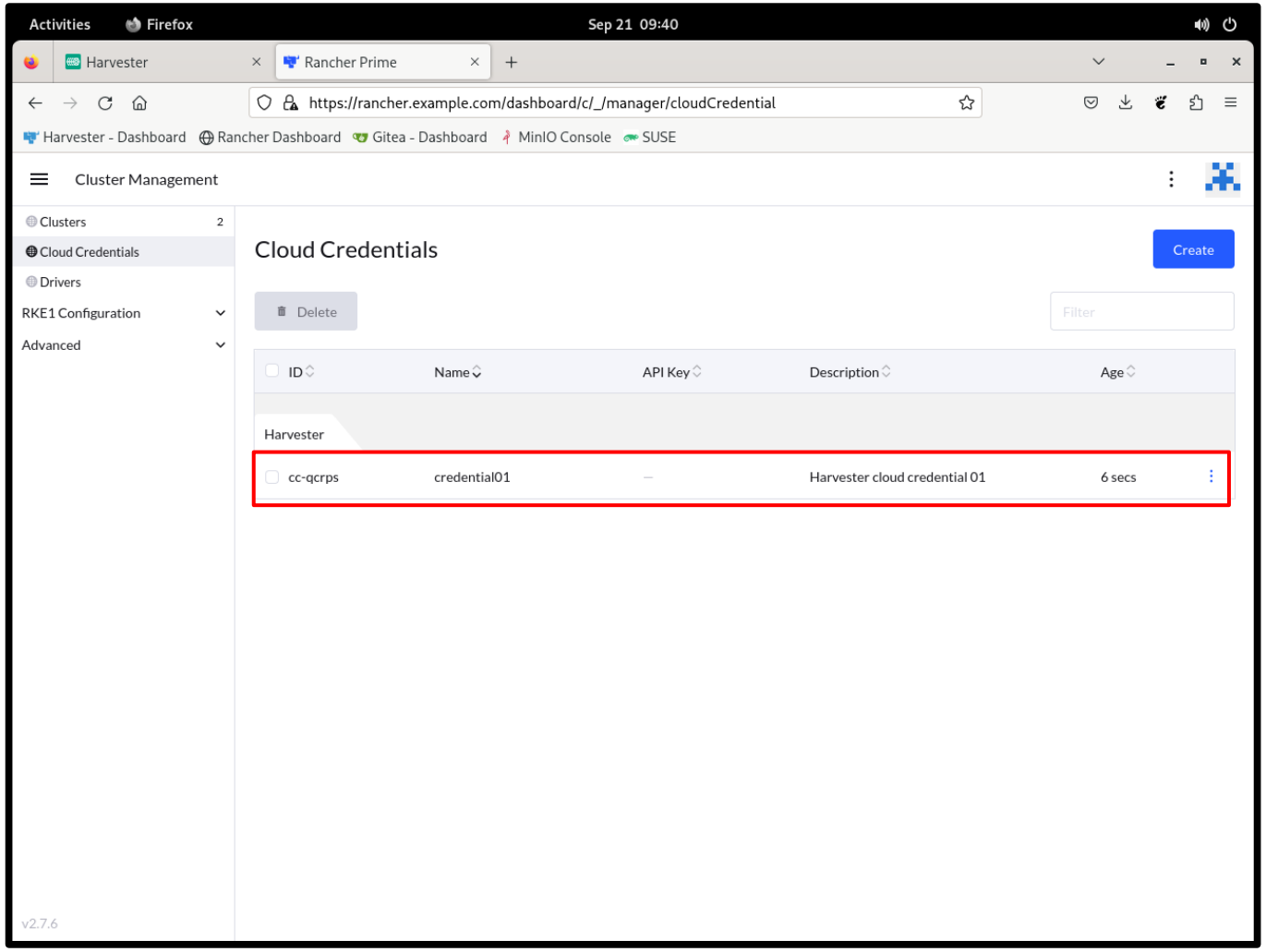
Cloud Credentials

- Cloud Credentials provide a mechanism of authentication and authorization when interacting with a cloud provider.
- Harvester cloud credentials enable this link specifically to Harvester clusters.
- On the **Cloud Credential: Create** screen select the **Harvester** cloud provider for the credentials.



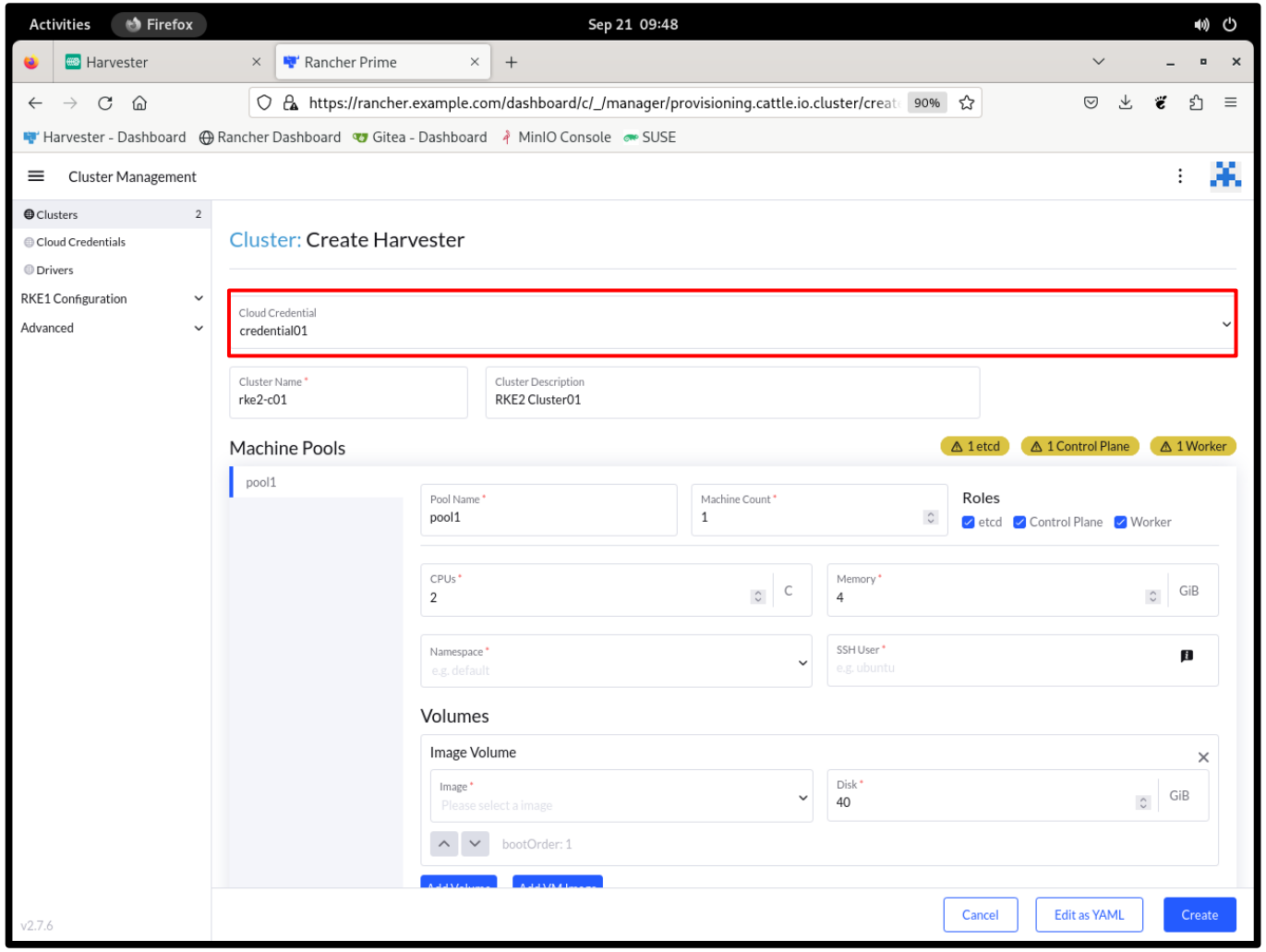
Cloud Credentials Ready for Use

- When created, the cloud credentials are available to be used when deploying RKE, RKE2 and K3S clusters on Harvester.



Cloud Credentials when Deploying a Cluster

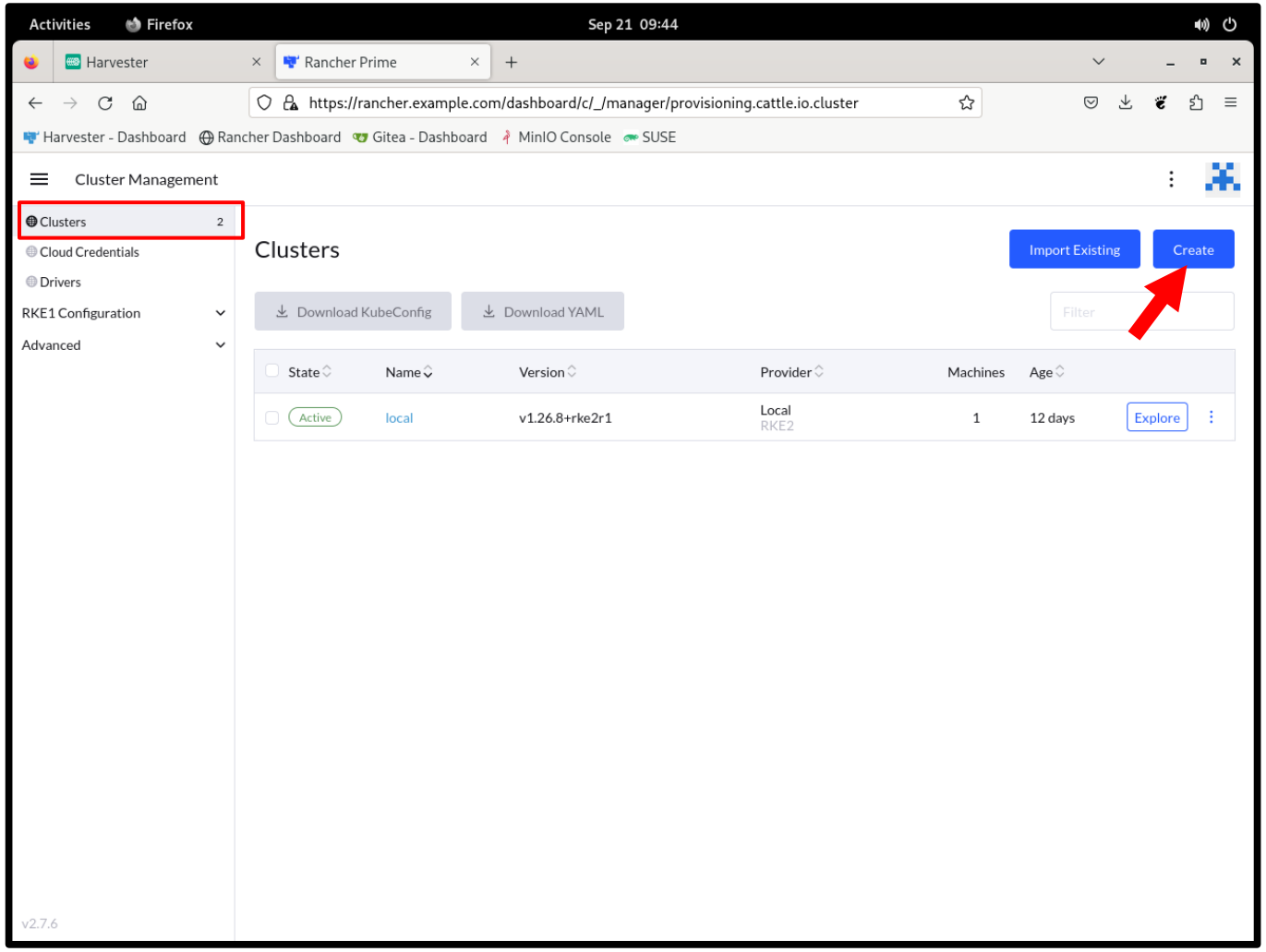
- When creating a new K8s cluster, on the **Cluster Create: Harvester** screen, select the cloud credential to be used.



Cluster Deployment with Rancher onto Harvester

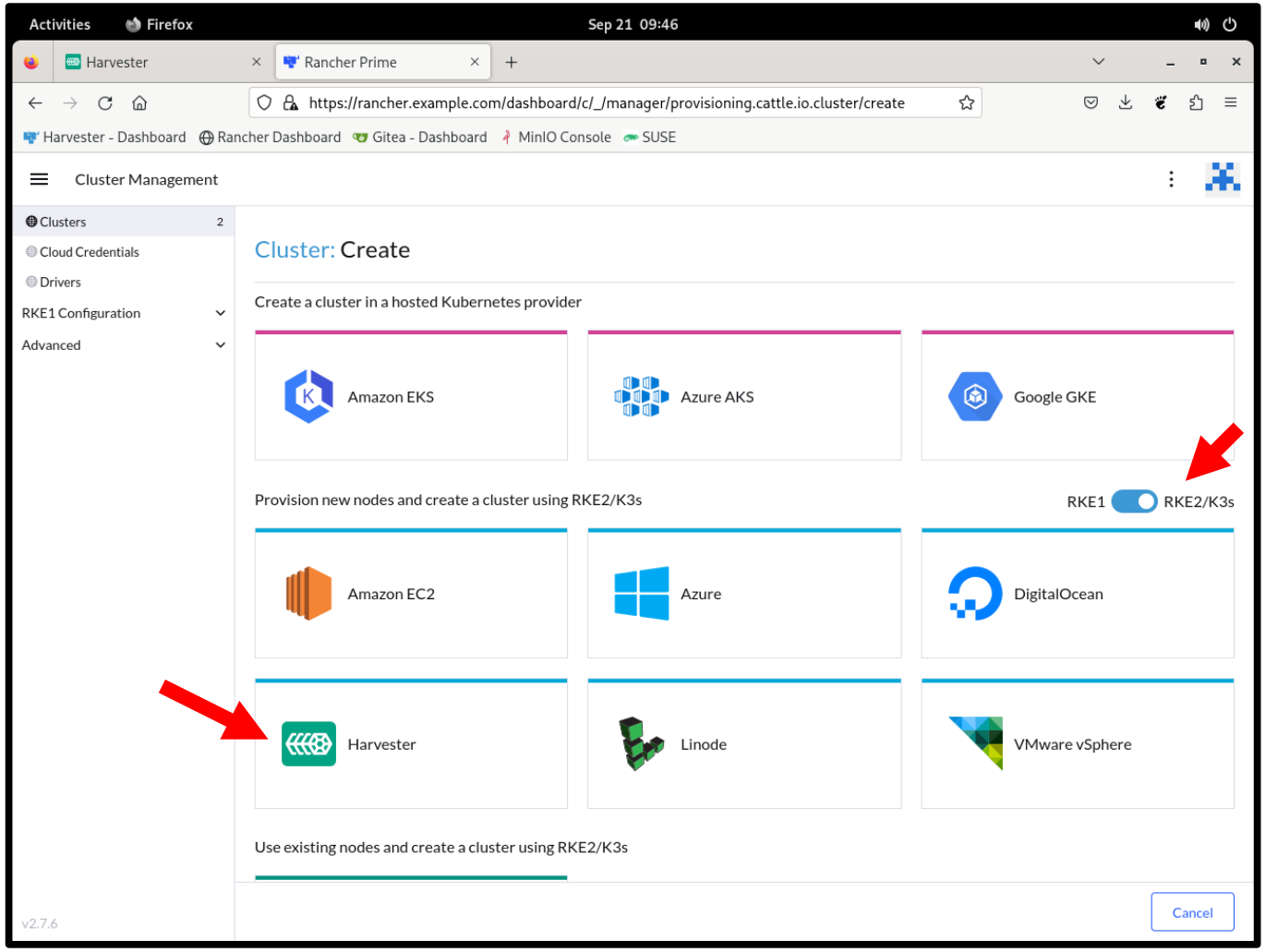
Deploy an RKE2 Cluster

- As usual clusters are deployed and managed on the **Cluster Management → Clusters** screen.
- To create a new cluster click: **Create**



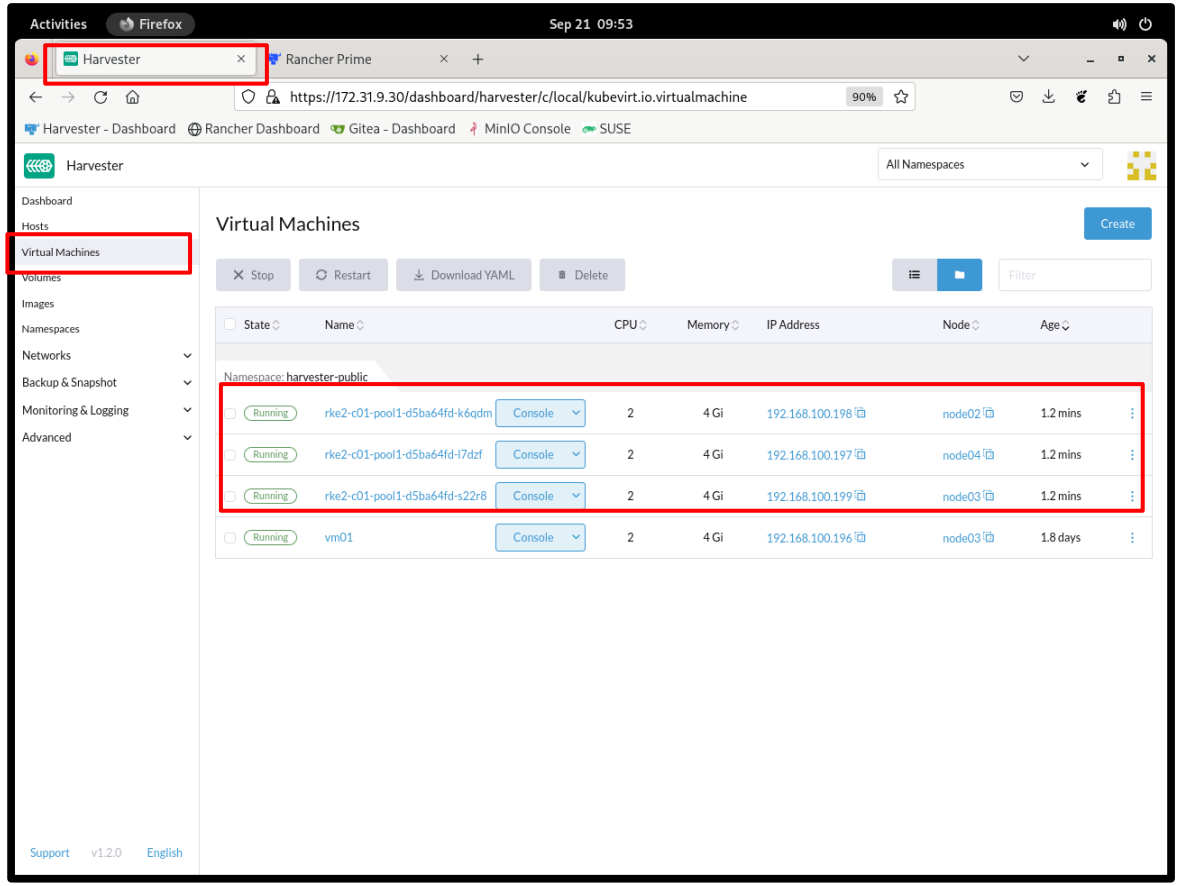
Deploy an RKE2 Cluster

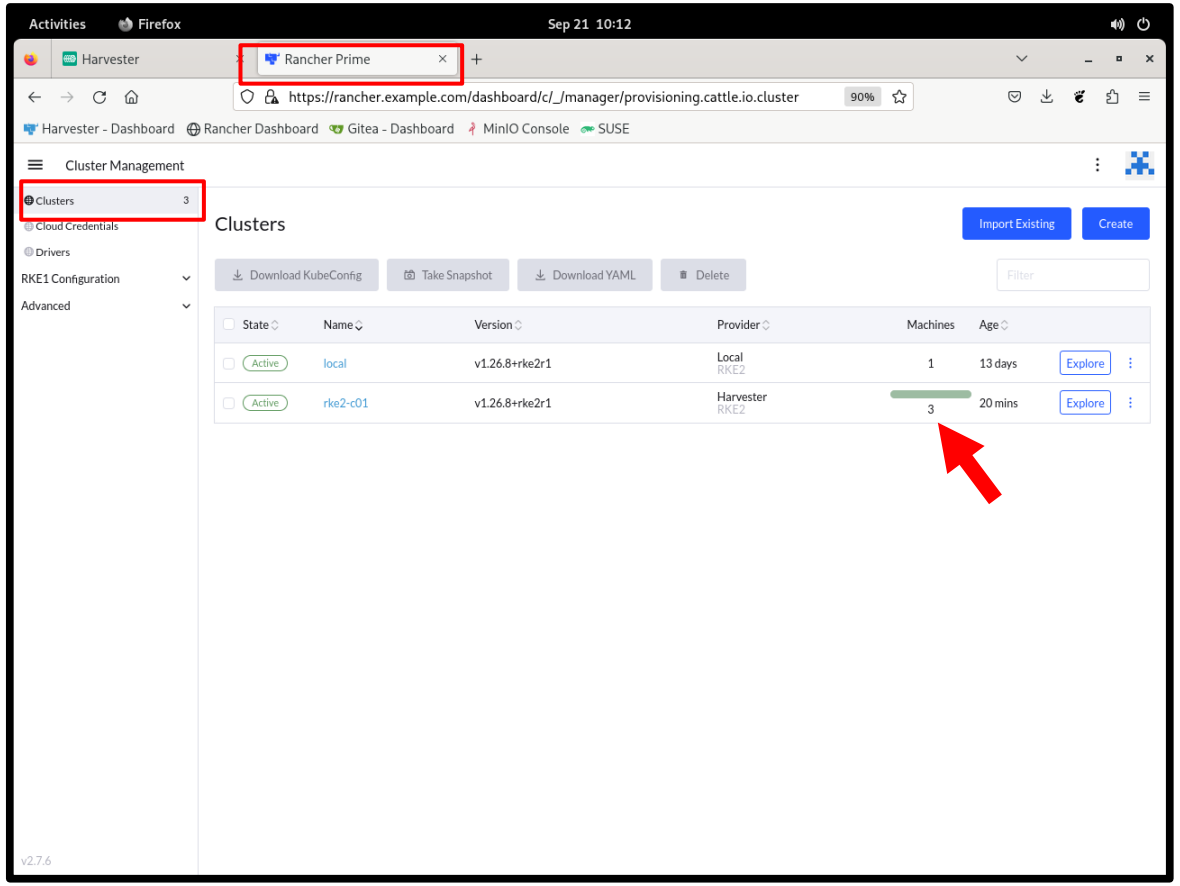
- On the **Cluster: Create** screen set the **RKE1-RKE2/K3s** slider to **RKE2/K3s**.
- Select: **Harvester**



Deploy an RKE2 Cluster

- As the Nodes are provisioned you can see them listed on the **Harvester UI: Virtual Machines** screen.
- The progress is also displayed on the **Rancher UI: Cluster Management → Clusters** scr





Application Deployment with Rancher onto Clusters Running on Harvester

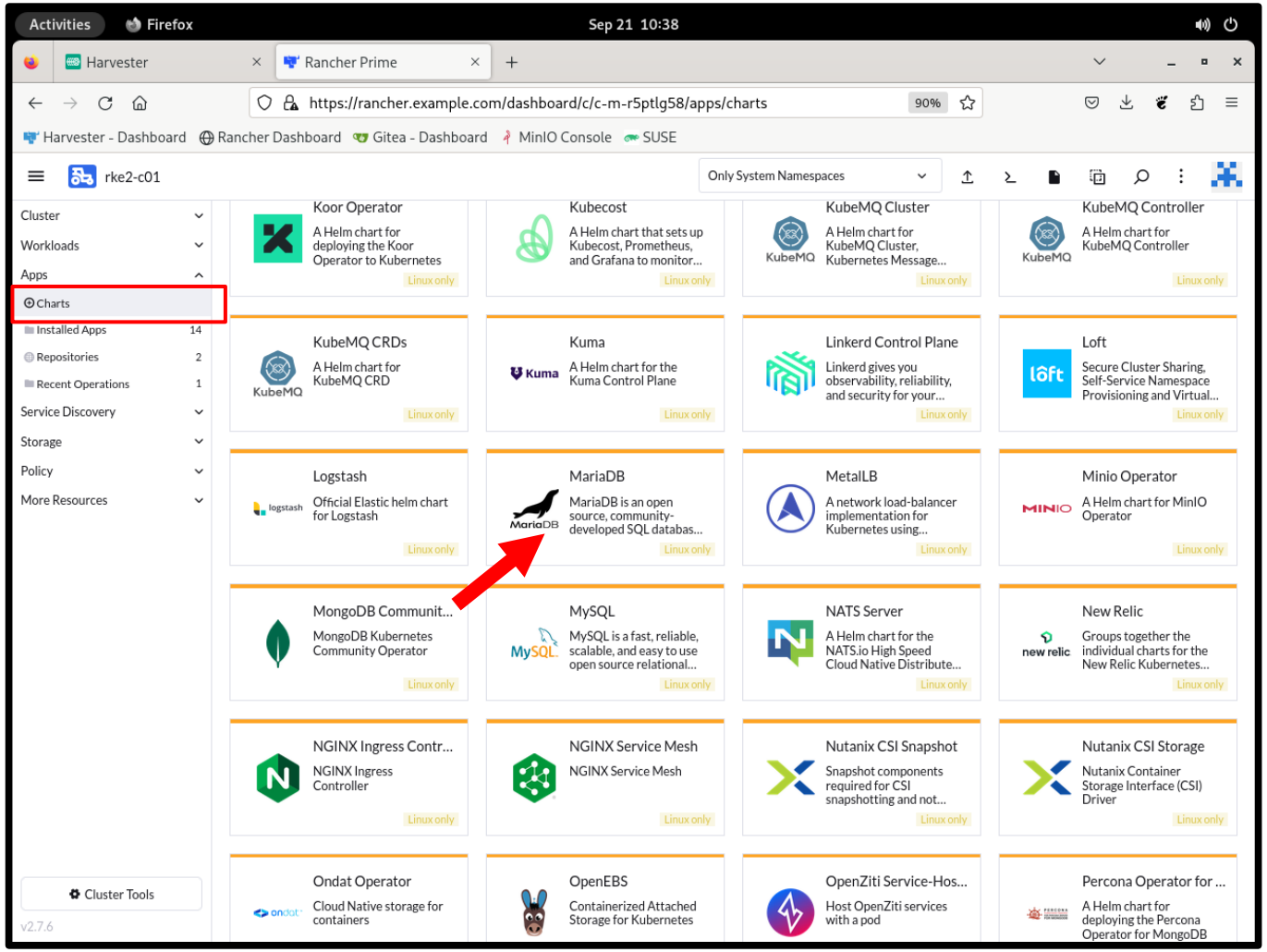
Deploy Applications on a Harvester Hosted RKE2 Cluster

- Applications deployed on a Harvester hosted downstream RKE2 cluster have access to resources such as Longhorn storage and Harvester load balancers.

- Applications are deployed in the same manner as on other RKE2 clusters.

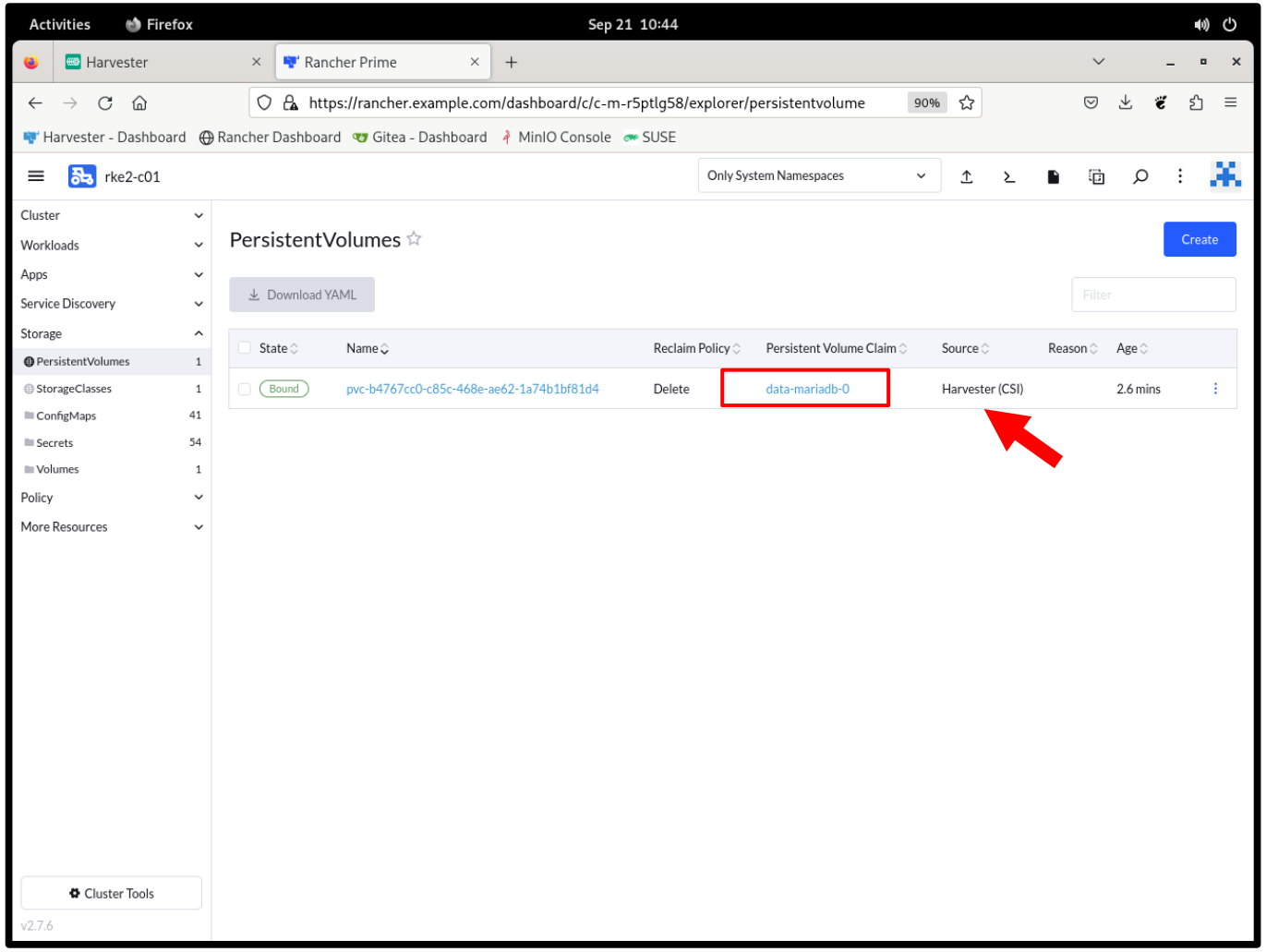
Deploy an Application

- Deploy an application on the Downstream cluster as usual:
  - **Cluster Explorer → <cluster_name> → Apps & Marketplace → Charts**
  - Select an application.
  - Go through the application's configuration and install it.



View Harvester Resources Used by the Application

- After the application is installed select:
  - **Storage → PersistentVolumes** to see the volume used on Harvester.



Section Summary

Section Summary

- Integrate Harvester with Rancher
  - Rancher can import Harvester clusters, which show up differently in Rancher than regular downstream K8s clusters, as they provide virtualization functionality.
  - The Harvester dashboard for the imported Harvester clusters can be accessed from within Rancher.
  - Rancher can be used to deploy downstream K8s clusters onto Harvester.
  - Importing a Harvester cluster into Rancher starts in the Rancher Manager UI.

Section Summary

- Manage and use Harvester clusters with Rancher users
  - Additional users for Harvester are created and managed in Rancher.
  - Additional Harvester users access the Harvester Dashboard by first logging into Rancher.
- Use Rancher to deploy Kubernetes Clusters into VMs running on Harvester
  - Before deploying K8s clusters on Harvester cloud credentials must first be created to grant access to the Harvester cluster.
  - Applications can be deployed from Rancher onto downstream clusters running on Harvester.
  - Applications running on downstream clusters running on Harvester have access to Harvester resources such as Longhorn storage and a Harvester load balancer.