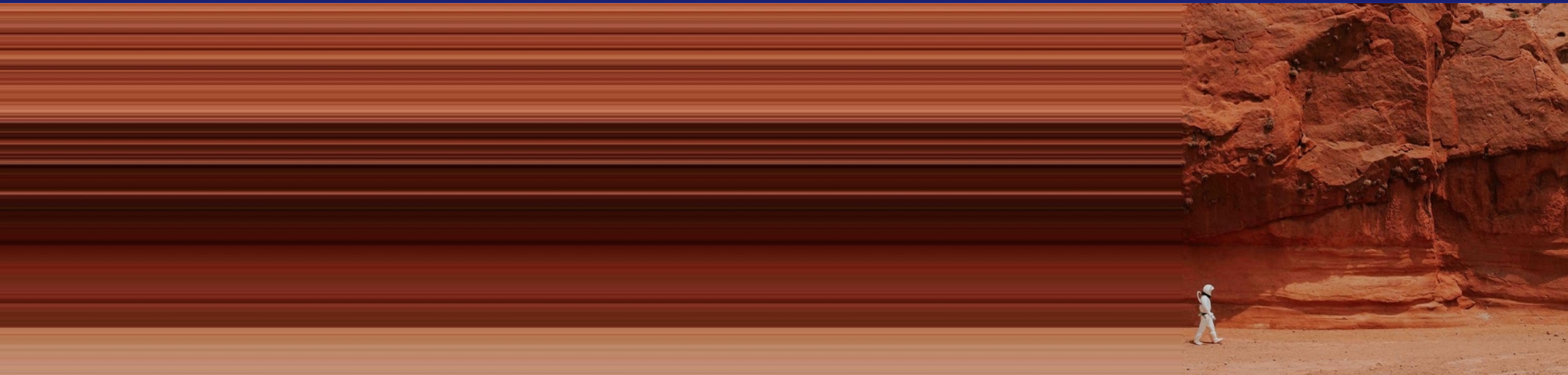




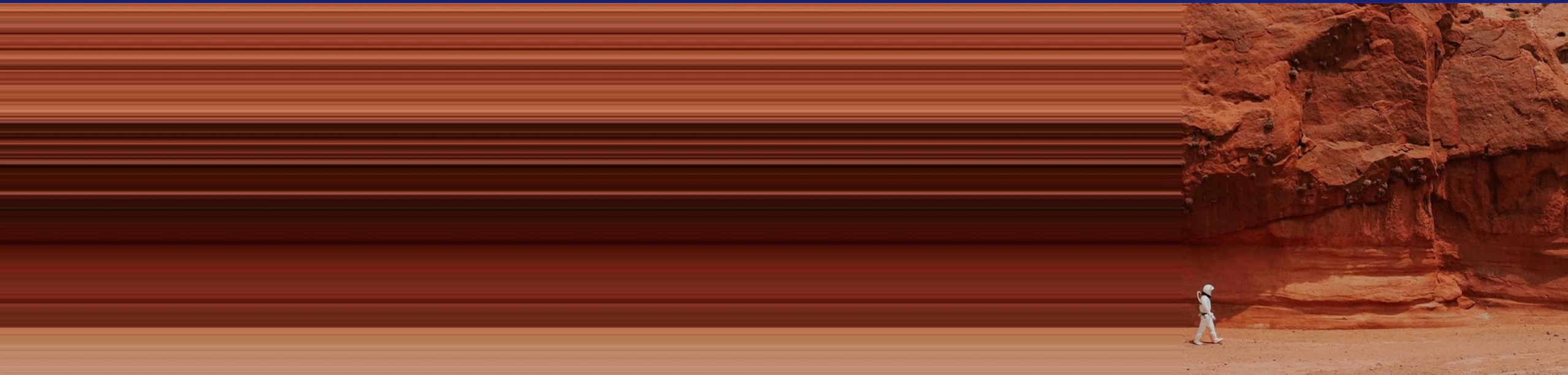
eHRV101: Introduction to Harvester





Section 1:

Course Introduction



Topics

- 1 Course Objectives and Audience
- 2 Course Lab Environment Overview
- 3 Additional SUSE Training



Course Objectives and Audience



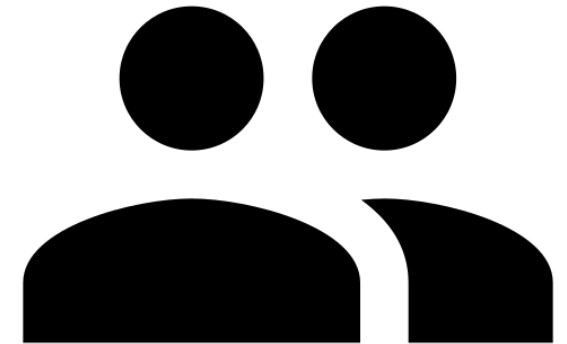
Course Overview

- 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

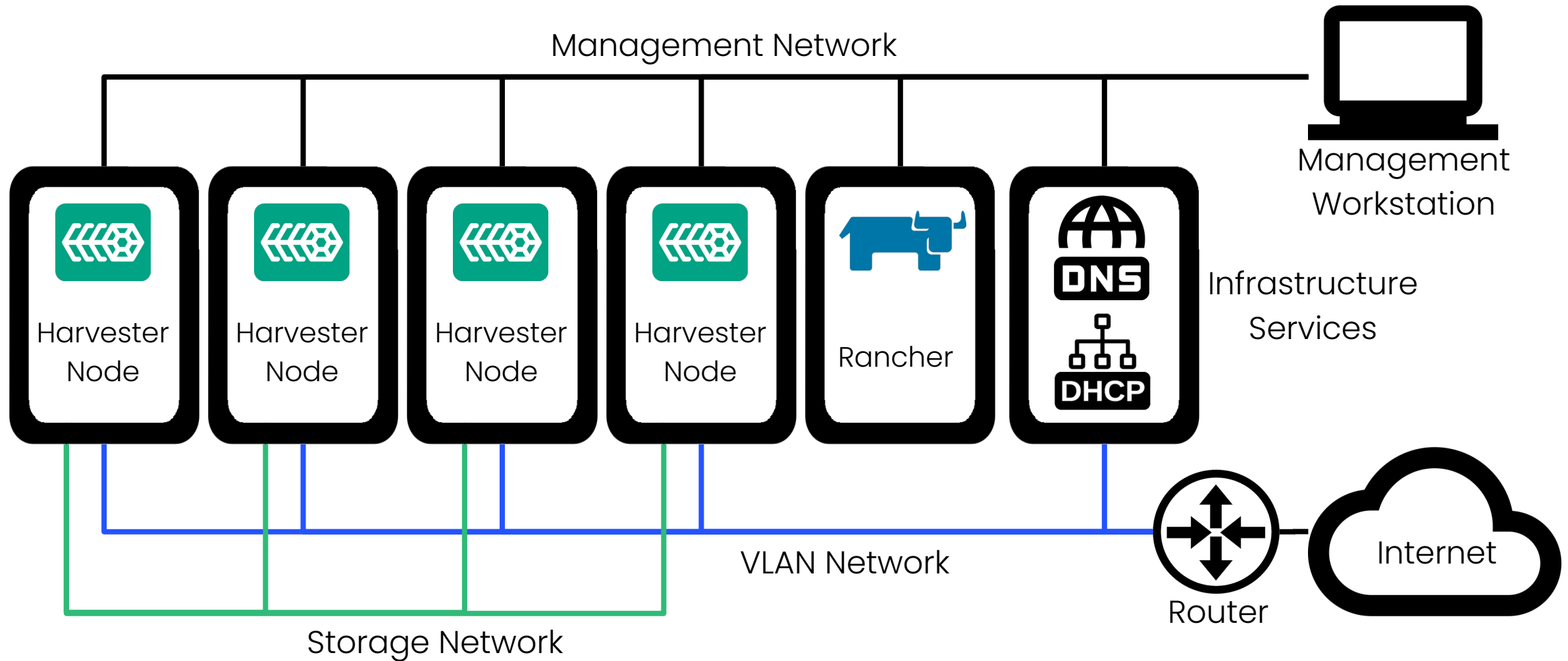
- Section 1: Course Introduction
- Section 2: Introduction to Harvester
- Section 3: Harvester Cluster Deployment
- Section 4: General Harvester Configuration
- Section 5: Networking in Harvester
- Section 6: Storage in Harvester
- Section 7: Virtualization Management with Harvester
- Section 8: Harvester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



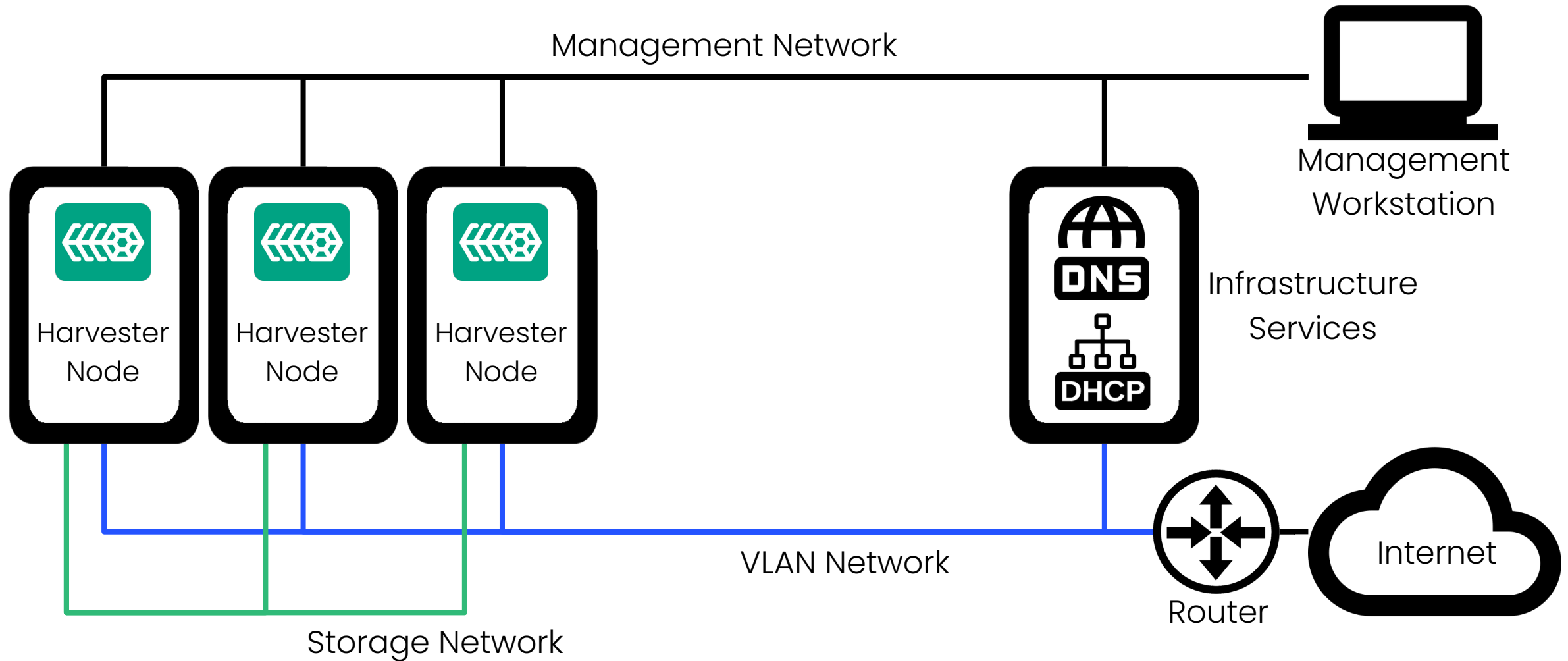
Course Lab Environment Overview



Lab Environment Diagram - Main



Lab Environment Diagram - Deployment/Upgrade

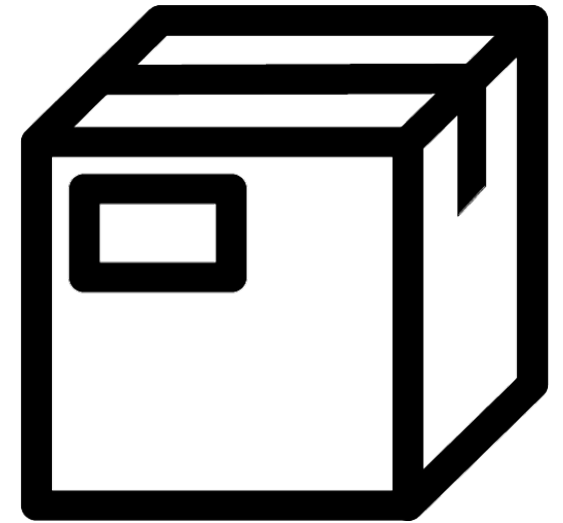


Required Minimum Product Version

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)

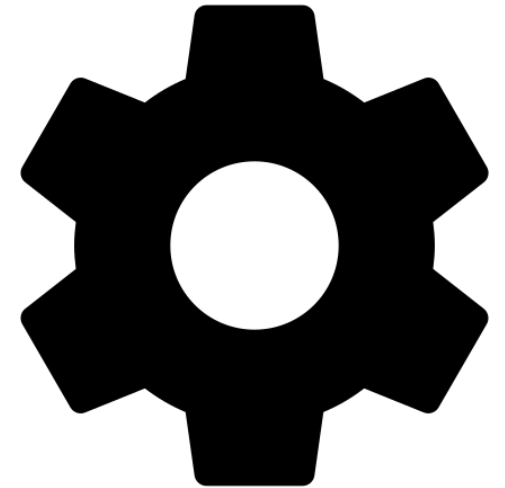
VM Host (Running Management and Rancher VMs)

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

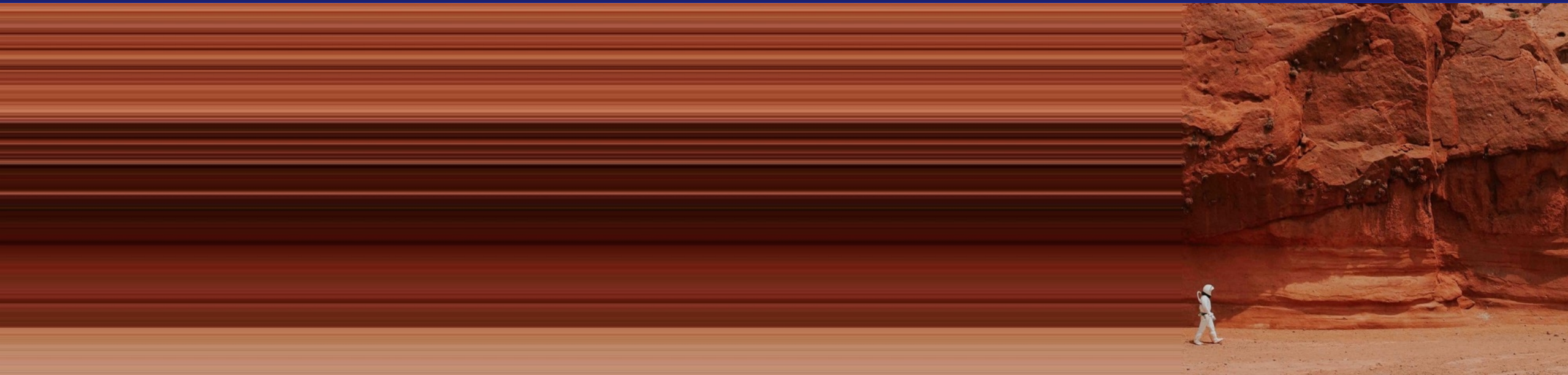
- SUSE Training is available across a wide range of products across our Business Critical Linux, Enterprise Container Management and Edge solutions.
- For more information:
 - <https://www.suse.com/training>





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

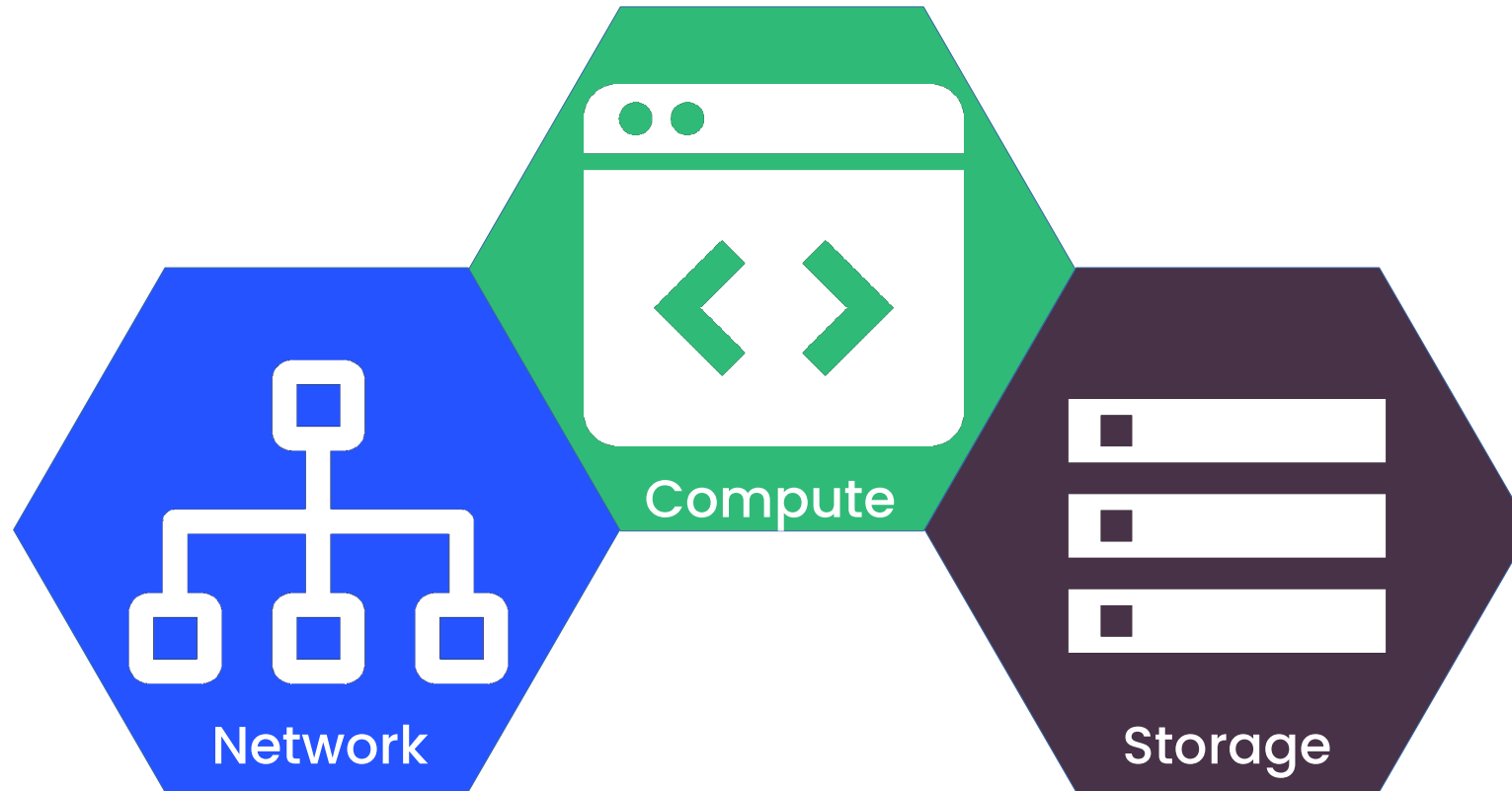
- 1 Hyperconverged Infrastructure Concepts
- 2 Harvester Features and Functionality



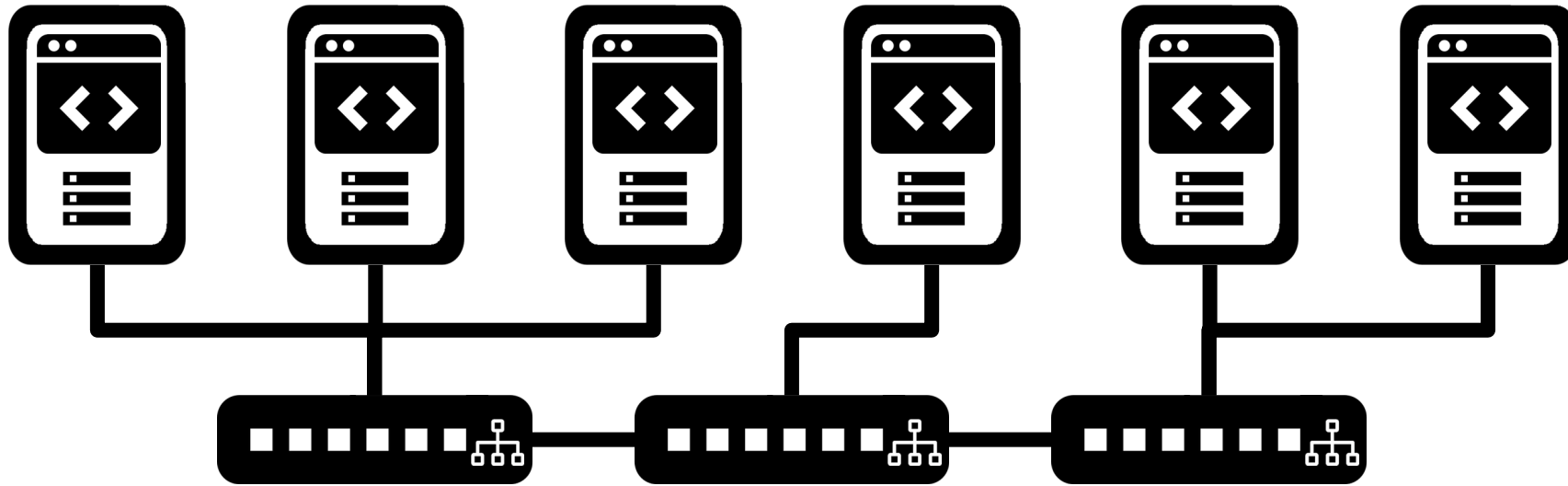
Hyperconverged Infrastructure Concepts



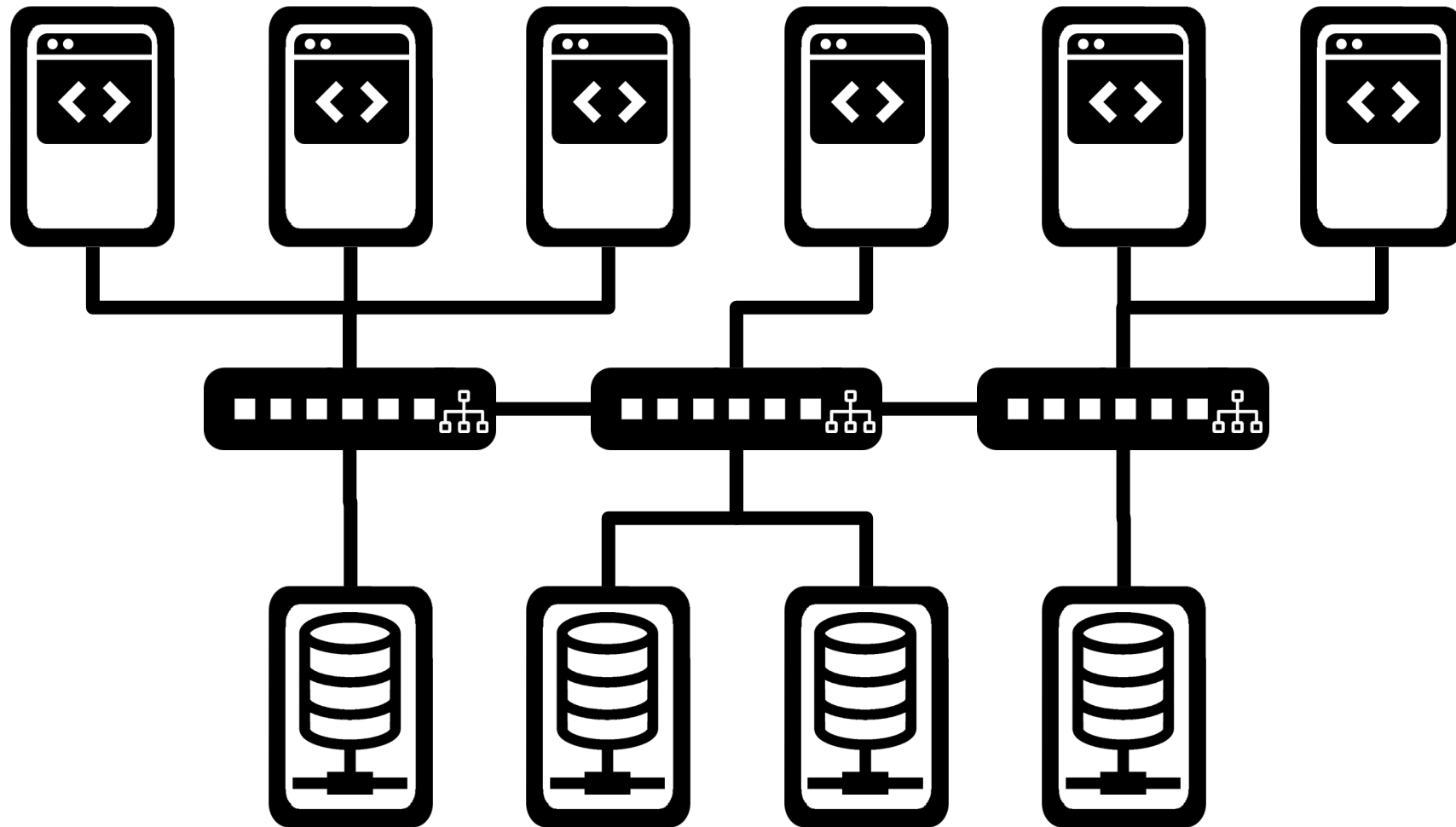
Fundamental Areas of Computing



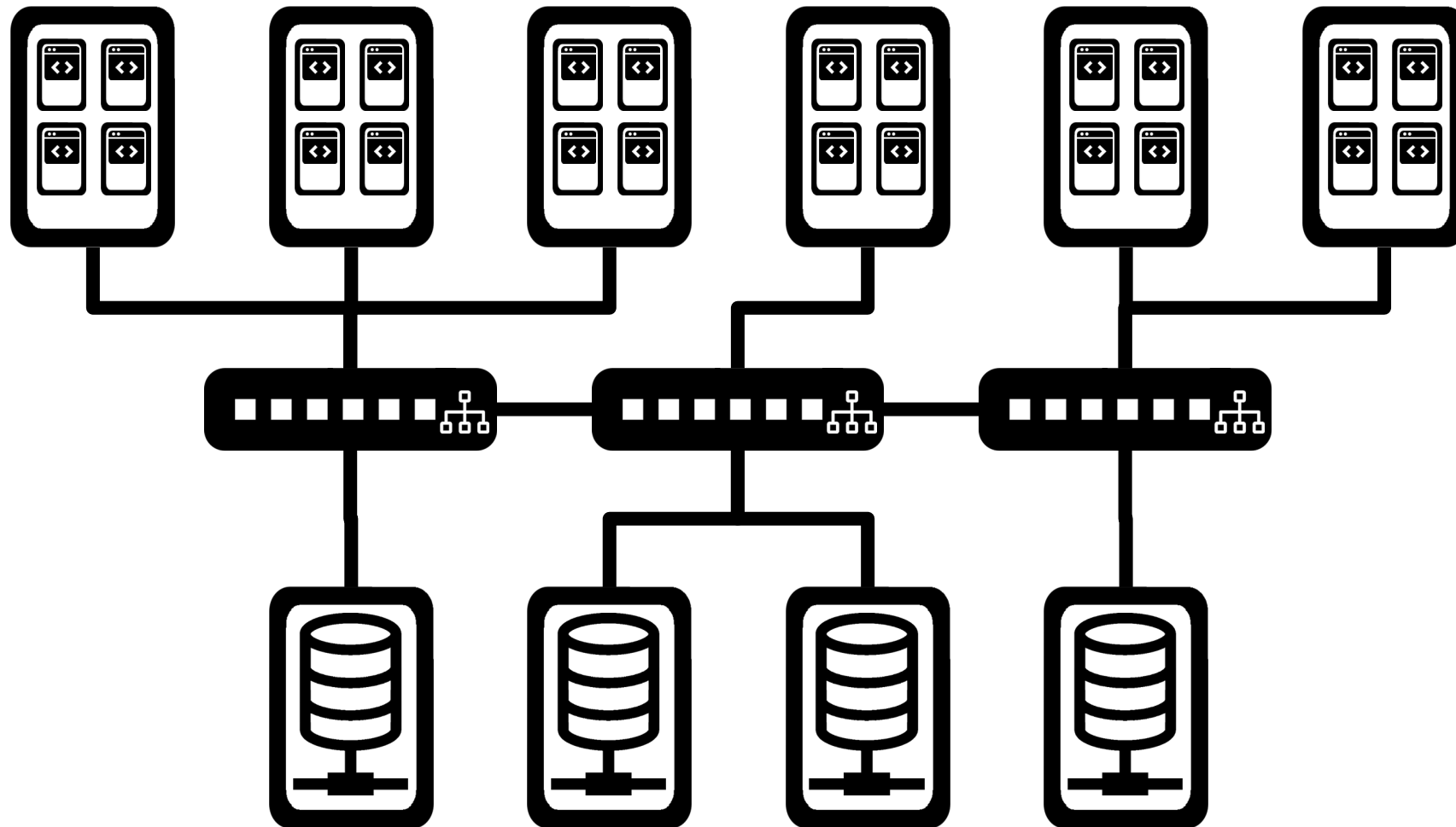
Traditional Architecture



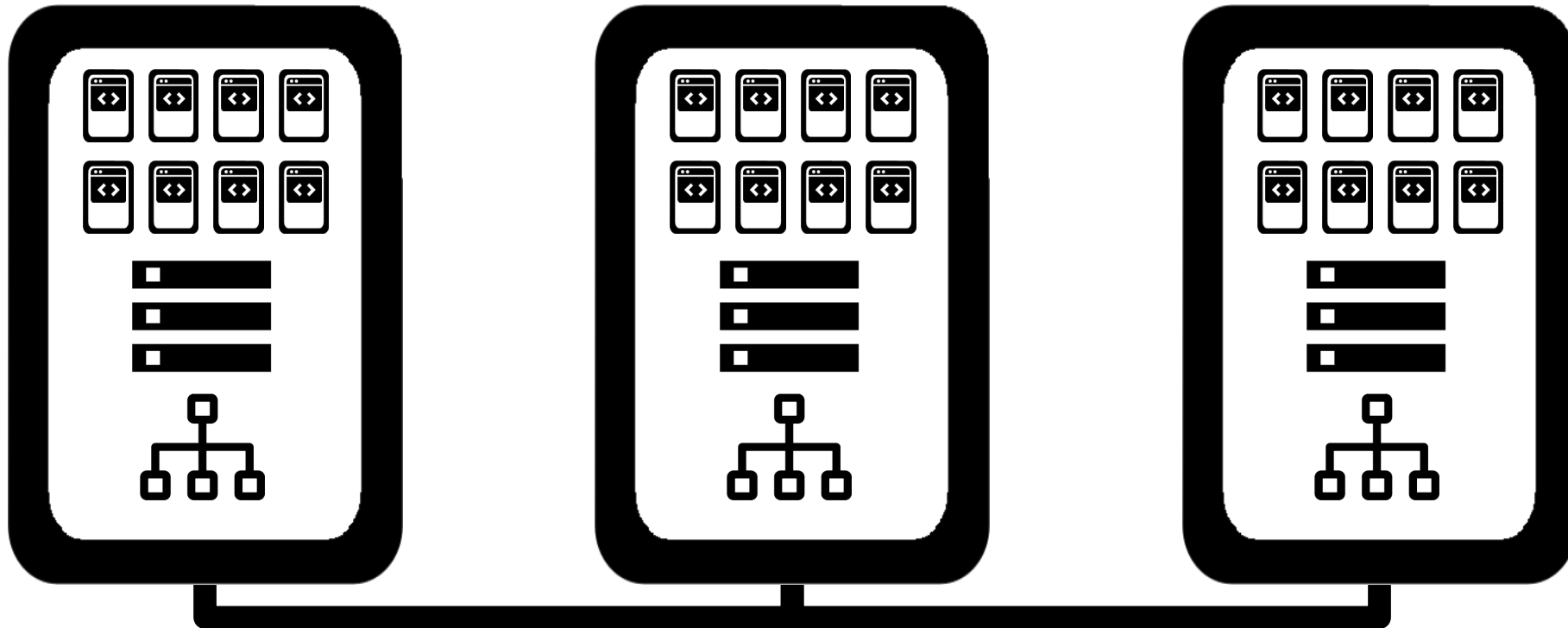
Traditional Architecture with Network Storage



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.

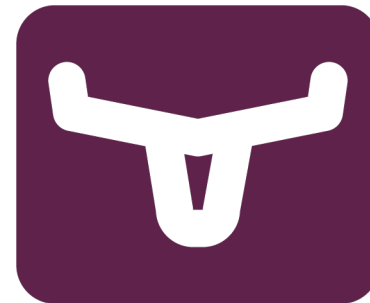


Harvester Features and Functionality



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

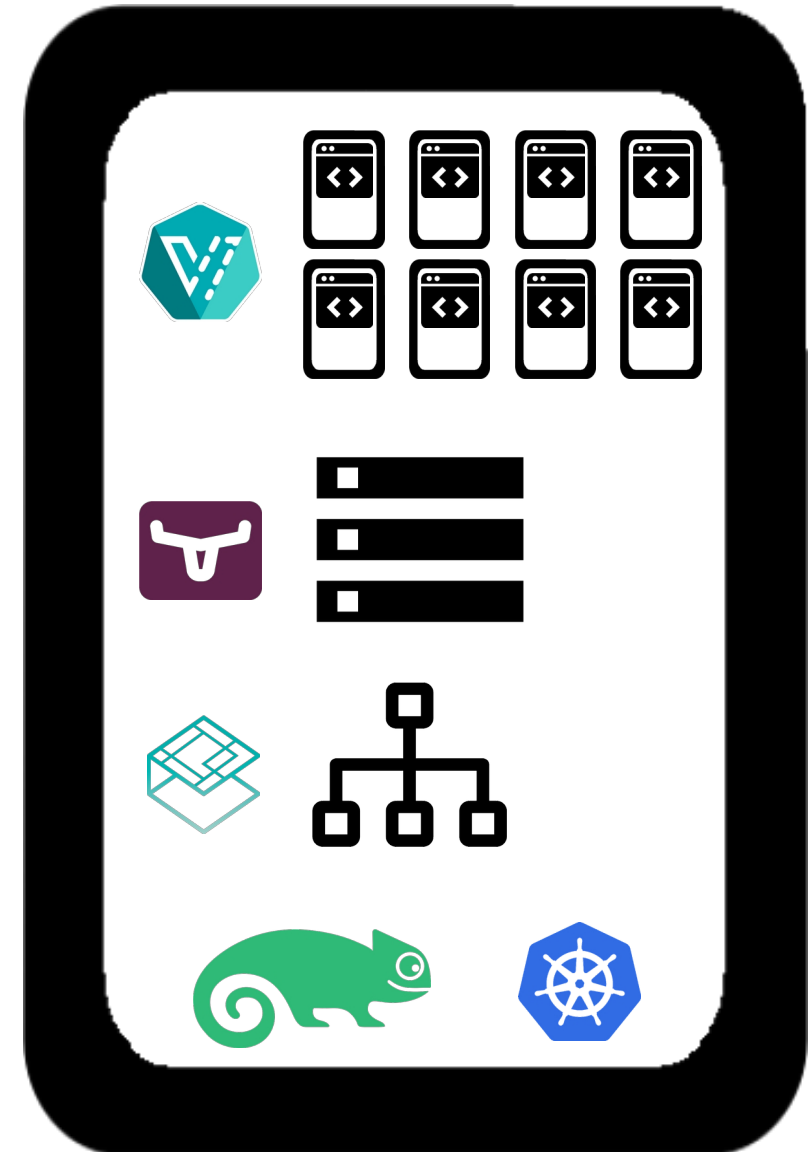
Virtualization Layer: KubeVirt

Storage Layer: Longhorn

Network Layer: Kubernetes CNIs
(Container Network Interfaces)

Orchestration Layer: Kubernetes

Operating System: SLE-Micro



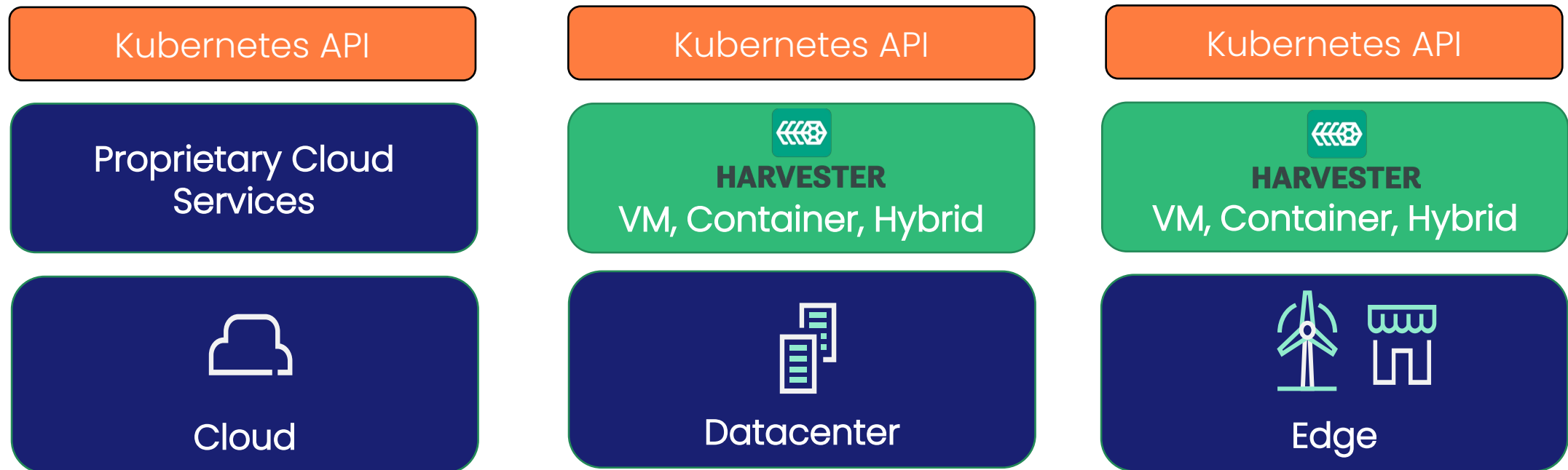
Hardware and Infrastructure Requirements

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

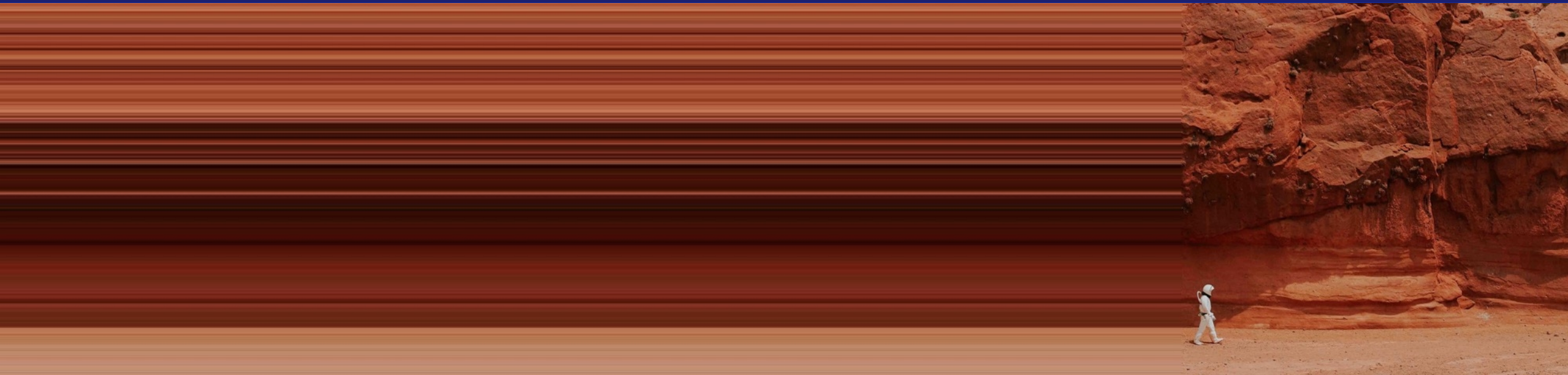


- Describe Hyperconverged Infrastructure Concepts
 - Hyperconverged Infrastructure (HCI) is a platform that combines compute/virtualization, network and storage components onto a single infrastructure node.
 - The capacity of HCI platforms is extended by adding additional nodes.
 - HCI uses software 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.



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

- 1 The Harvester Installation Process
- 2 Harvester Cluster Deployment Preparation
- 3 Harvester Cluster Manual Deployment
- 4 Harvester Cluster Automated Deployment

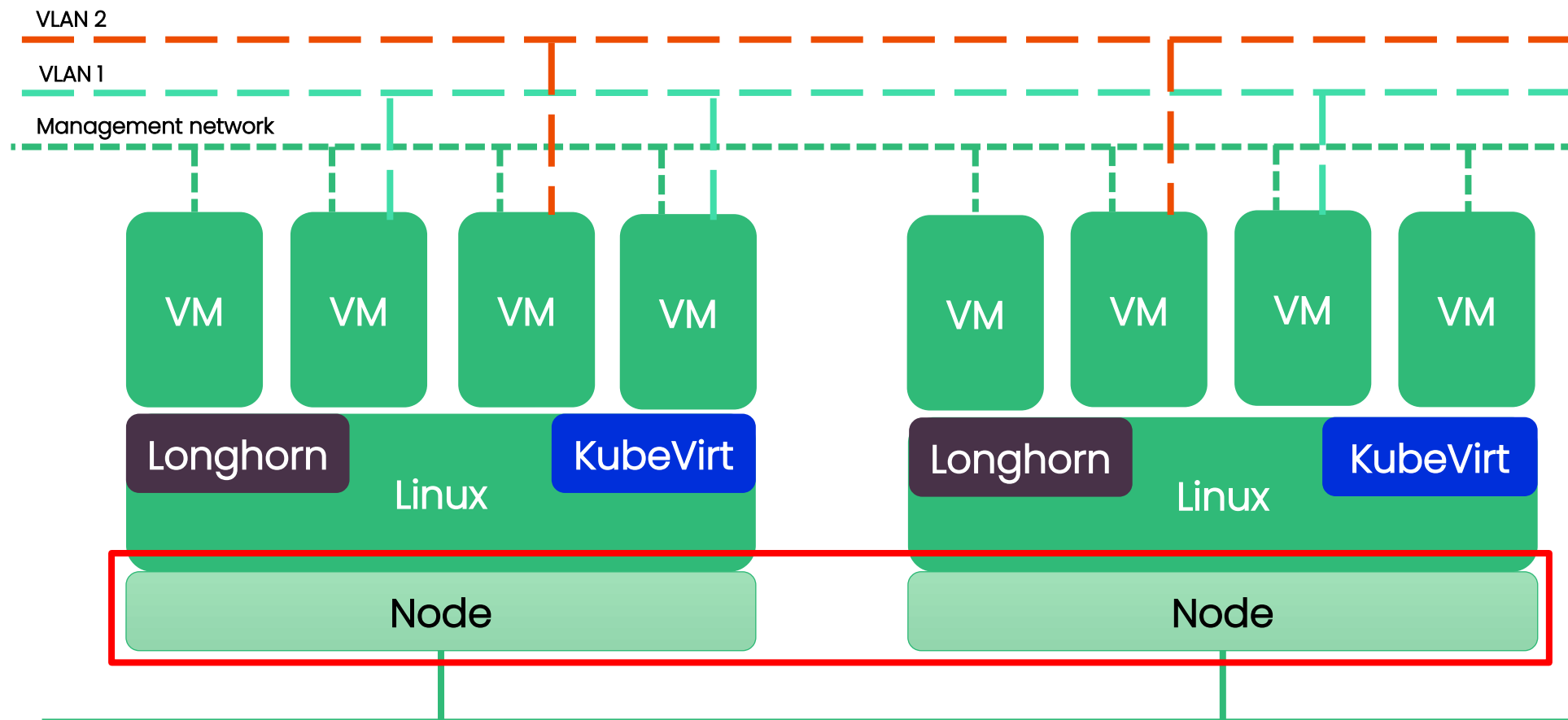


The Harvester Installation Process



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.

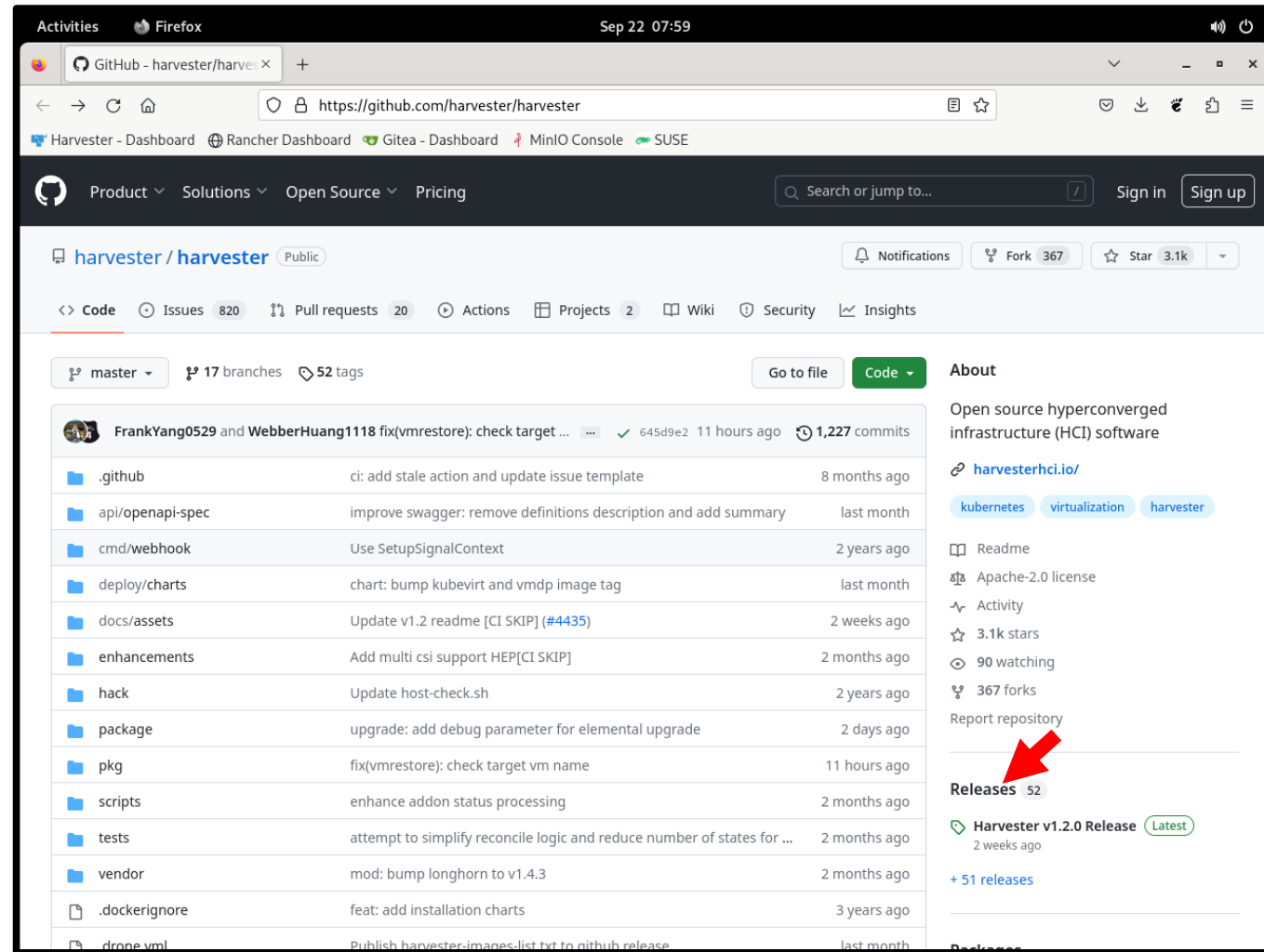


Harvester Cluster Deployment Preparation



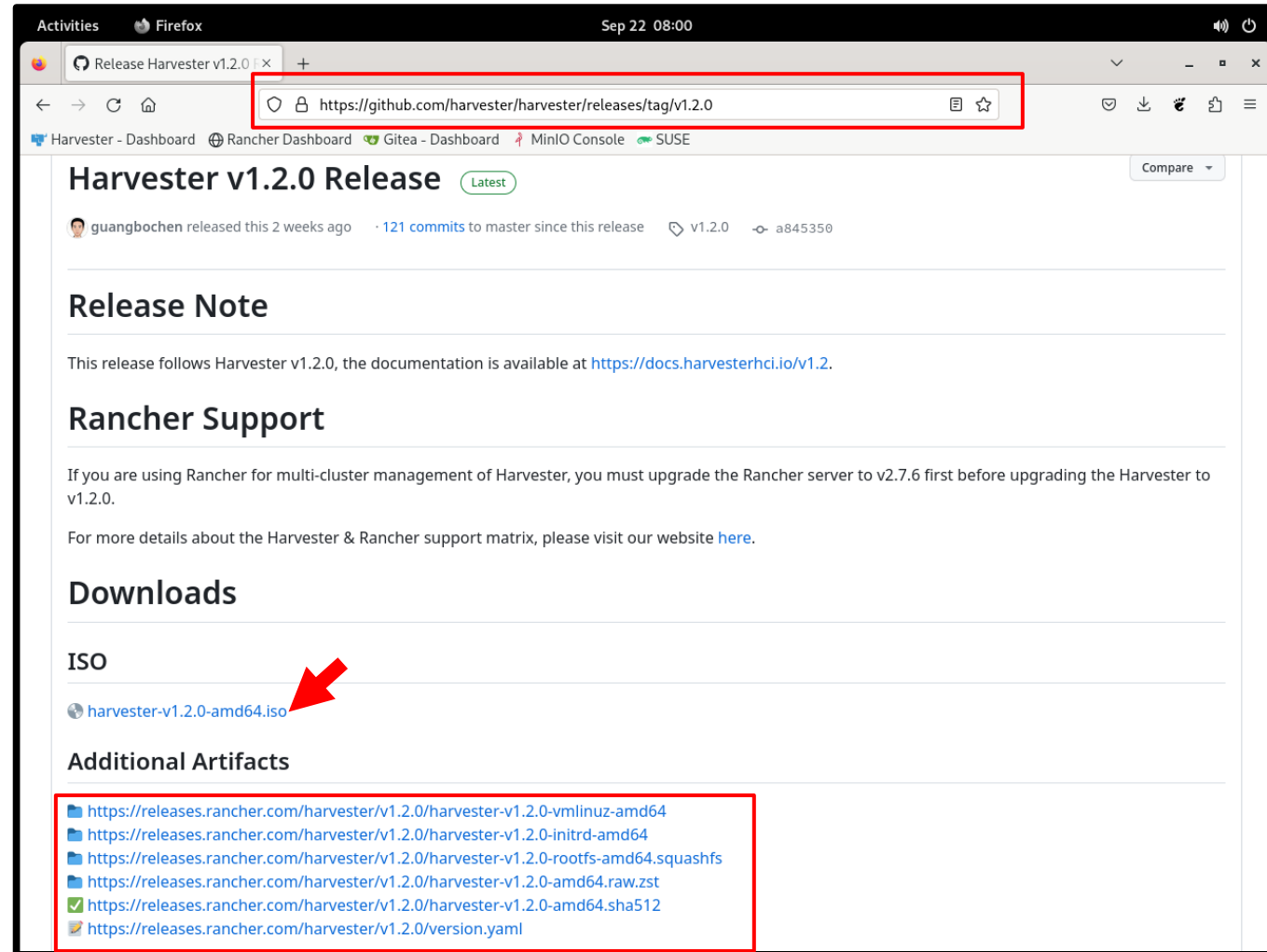
Download Installation Media

- 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.



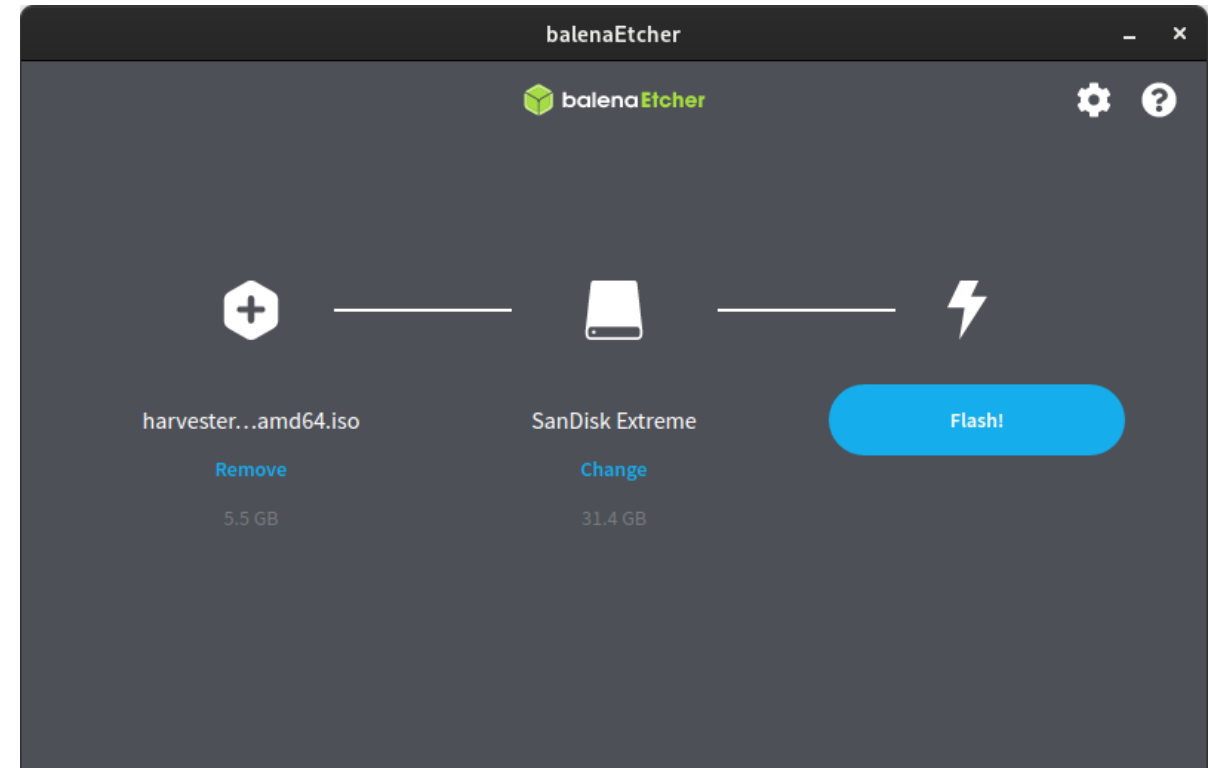
Download Installation Media

- 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.

```
/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-vmlinux-amd64
```

```
| ipxe/  
| -ipxe-boot  
| -ipxe-boot-efi  
| -ipxe-x86_64.efi
```

iPXE:

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



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 **dhcpcd.conf** file to add a group for the iPXE booting.
- 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";
        }
    }
    elif 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";
        }
    } elseif 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.

Optional: remote Harvester config

HTTP URL

<Use ESC to go back to previous page>



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
version: v1.2.0

Harvester Cluster
* Management URL:
  https://172.31.9.30
* Status: Ready

Node
* Hostname: node01
* IP Address: 172.31.9.122
* Status: Ready

<Use F12 to switch between Harvester console and Shell>
```

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
|   |-ipxe-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

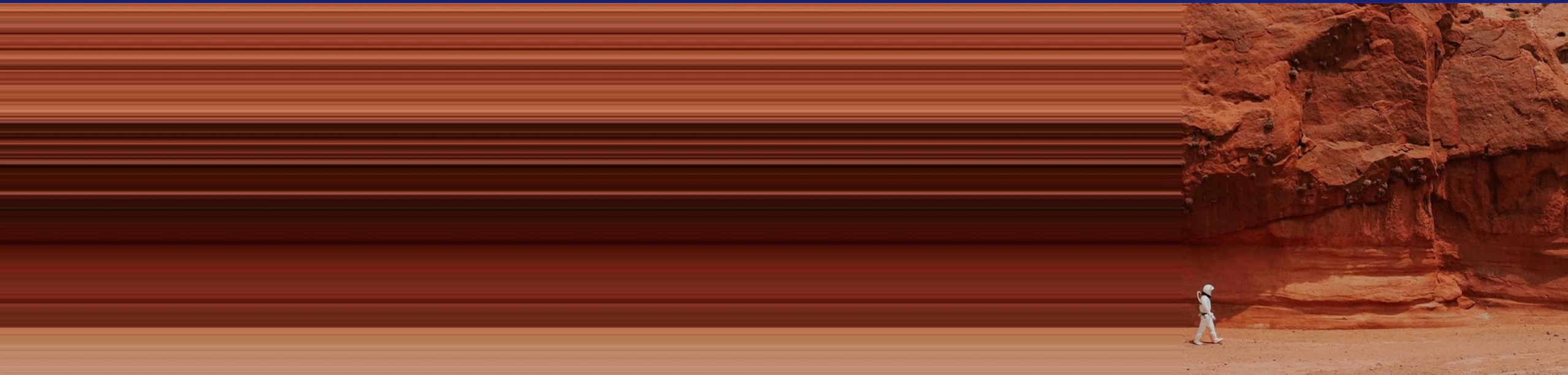


- Manually deploy a Harvester cluster
 - Harvester can be deployed from ISO, USB or via the Network.
 - Installing the first node of a cluster creates the cluster.
 - Additional nodes must join the cluster created on the first node when they are installed.
- Automate the deployment of a Harvester cluster
 - 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.



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

- 1 Harvester Cluster Node Configuration
- 2 Harvester Addons
- 3 Cluster Monitoring and Metrics
- 4 VM Backup Target Configuration

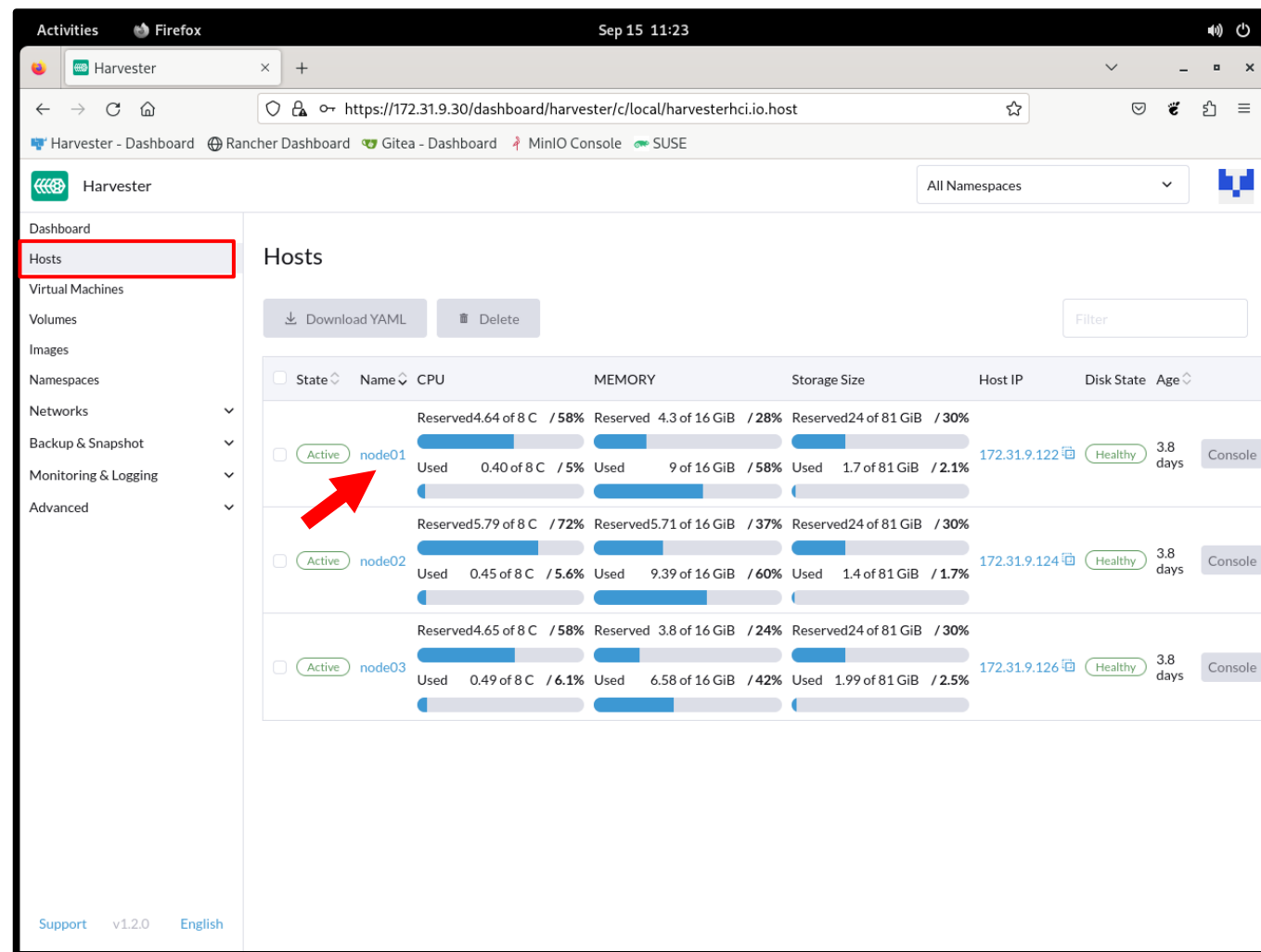


Harvester Cluster Node Configuration



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.



The screenshot shows the Harvester web interface in a Firefox browser. The left sidebar contains a navigation menu with items: Dashboard, Hosts (highlighted with a red box), Virtual Machines, Volumes, Images, Namespaces, Networks, Backup & Snapshot, Monitoring & Logging, and Advanced. The main content area is titled 'Hosts' and features a table of cluster nodes. Above the table are buttons for 'Download YAML' and 'Delete', and a 'Filter' input field. The table has columns for State, Name, CPU, MEMORY, Storage Size, Host IP, Disk State, and Age. Three hosts are listed: node01, node02, and node03, all with an 'Active' state and 'Healthy' disk status. A red arrow points to the 'node01' link in the first row.

State	Name	CPU	MEMORY	Storage Size	Host IP	Disk State	Age	
Active	node01	Reserved 4.64 of 8 C / 58% Used 0.40 of 8 C / 5%	Reserved 4.3 of 16 GiB / 28% Used 9 of 16 GiB / 58%	Reserved 24 of 81 GiB / 30% Used 1.7 of 81 GiB / 2.1%	172.31.9.122	Healthy	3.8 days	Console
Active	node02	Reserved 5.79 of 8 C / 72% Used 0.45 of 8 C / 5.6%	Reserved 5.71 of 16 GiB / 37% Used 9.39 of 16 GiB / 60%	Reserved 24 of 81 GiB / 30% Used 1.4 of 81 GiB / 1.7%	172.31.9.124	Healthy	3.8 days	Console
Active	node03	Reserved 4.65 of 8 C / 58% Used 0.49 of 8 C / 6.1%	Reserved 3.8 of 16 GiB / 24% Used 6.58 of 16 GiB / 42%	Reserved 24 of 81 GiB / 30% Used 1.99 of 81 GiB / 2.5%	172.31.9.126	Healthy	3.8 days	Console



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

The screenshot displays the Harvester web interface in a Firefox browser window. The address bar shows the URL `https://172.31.9.30/dashboard/harvester/c/local/harvesterhci.io/host/node01#basics`. The left sidebar contains a navigation menu with items: Dashboard, Hosts, Virtual Machines, Volumes, Images, Namespaces, Networks, Backup & Snapshot, Monitoring & Logging, and Advanced. A red bracket highlights the 'Networks' section, which includes sub-items: Basics, Virtual Machines, Network, Storage, Ksmtuned, Out-of-Band Access, and Events. The main content area is titled 'Host: node01' with an 'Active' status. It shows the host's age as 3.8 days, labels (cpumanager: false), and annotations (21 annotations). The 'Basics' tab is selected, displaying an overview of the host's configuration, including the custom name, operating system (Harvester v1.2.0), creation time, and console URL. Below the overview, there is a 'Monitor Data' section with three horizontal bar charts for CPU, MEMORY, and Storage, showing reserved and used resources. The CPU chart shows 4.64 of 8 C reserved (58%) and 0.41 of 8 C used (5.1%). The MEMORY chart shows 4.3 of 16 GiB reserved (28%) and 8.98 of 16 GiB used (58%). The Storage chart shows 24 of 81 GiB reserved (30%) and 1.7 of 81 GiB used (2.1%). At the bottom, the 'More Information' section displays the UUID, kernel version, and container runtime.

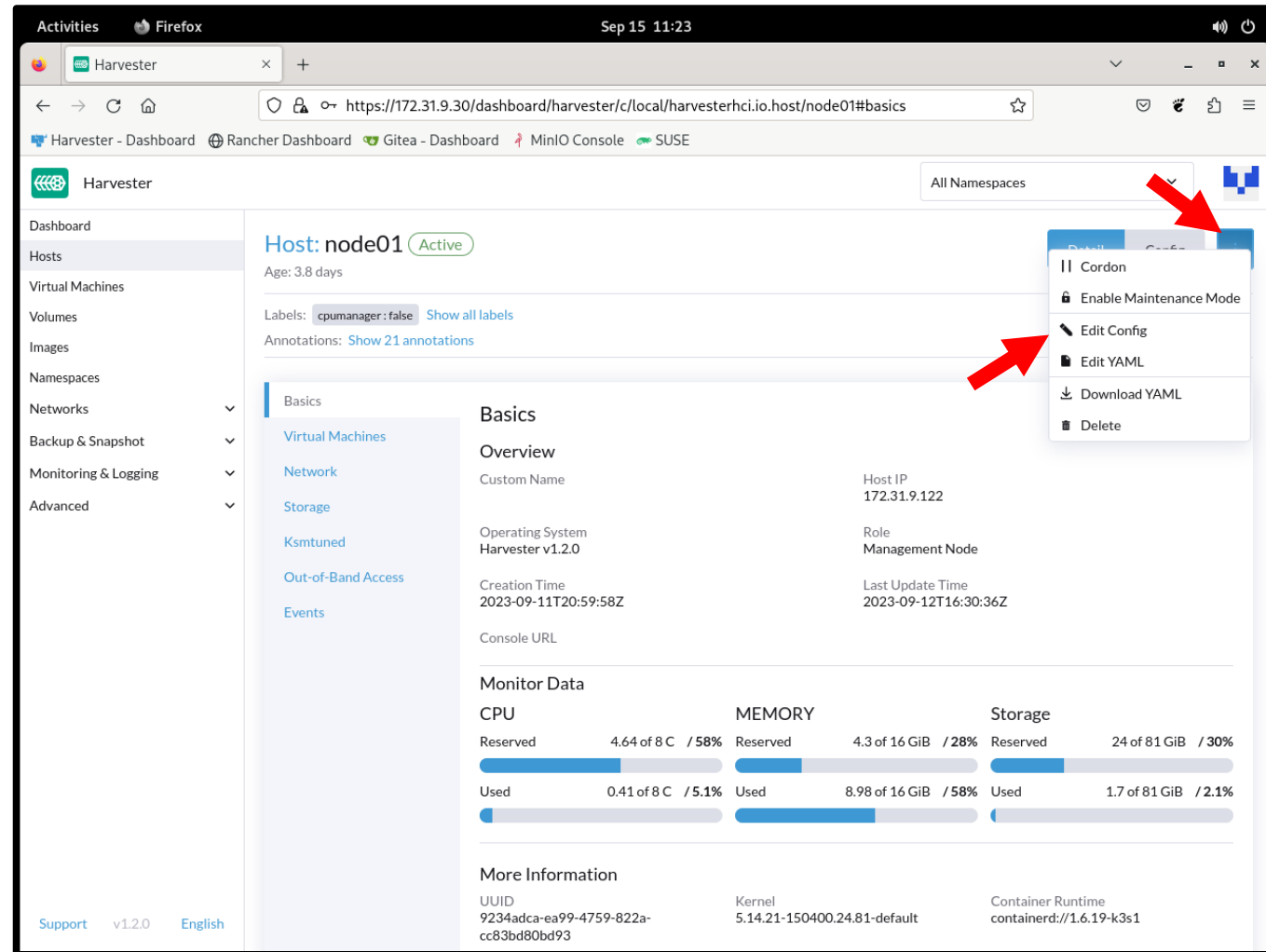
Monitor Data		
CPU	MEMORY	Storage
Reserved: 4.64 of 8 C / 58%	Reserved: 4.3 of 16 GiB / 28%	Reserved: 24 of 81 GiB / 30%
Used: 0.41 of 8 C / 5.1%	Used: 8.98 of 16 GiB / 58%	Used: 1.7 of 81 GiB / 2.1%

More Information		
UUID: 9234adca-ea99-4759-822a-cc83bd80bd93	Kernel: 5.14.21-150400.24.81-default	Container Runtime: containerd://1.6.19-k3s1



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**



The screenshot shows the Harvester web interface in a Firefox browser. The URL is `https://172.31.9.30/dashboard/harvester/c/local/harvesterhci.io/host/node01#basics`. The left sidebar contains navigation links: Dashboard, Hosts, Virtual Machines, Volumes, Images, Namespaces, Networks, Backup & Snapshot, Monitoring & Logging, and Advanced. The main content area displays details for 'Host: node01' (Active), including its age (3.8 days), labels, and annotations. A three-dot menu is open, showing options: Cordon, Enable Maintenance Mode, Edit Config (highlighted with a red arrow), Edit YAML, Download YAML, and Delete. Below the menu, the 'Basics' tab is selected, showing an overview of the host's status, including Host IP (172.31.9.122), Operating System (Harvester v1.2.0), Role (Management Node), and creation/last update times. A 'Monitor Data' section displays CPU, MEMORY, and Storage usage with progress bars. The bottom section shows 'More Information' including UUID, Kernel, and Container Runtime details.

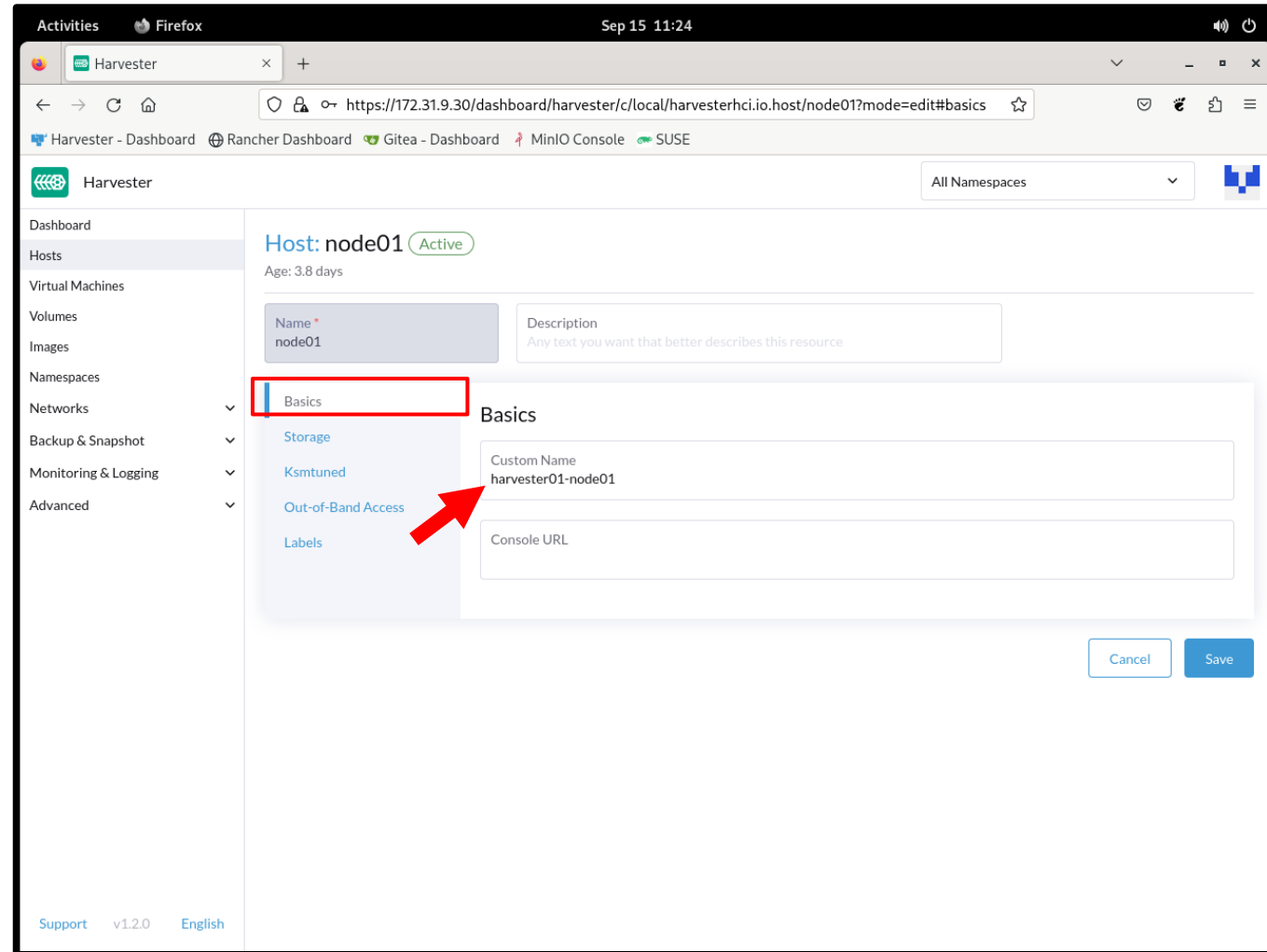
CPU		MEMORY		Storage	
Reserved	4.64 of 8 C / 58%	Reserved	4.3 of 16 GiB / 28%	Reserved	24 of 81 GiB / 30%
Used	0.41 of 8 C / 5.1%	Used	8.98 of 16 GiB / 58%	Used	1.7 of 81 GiB / 2.1%

UUID	Kernel	Container Runtime
9234adca-ea99-4759-822a-cc83bd80bd93	5.14.21-150400.24.81-default	containerd://1.6.19-k3s1



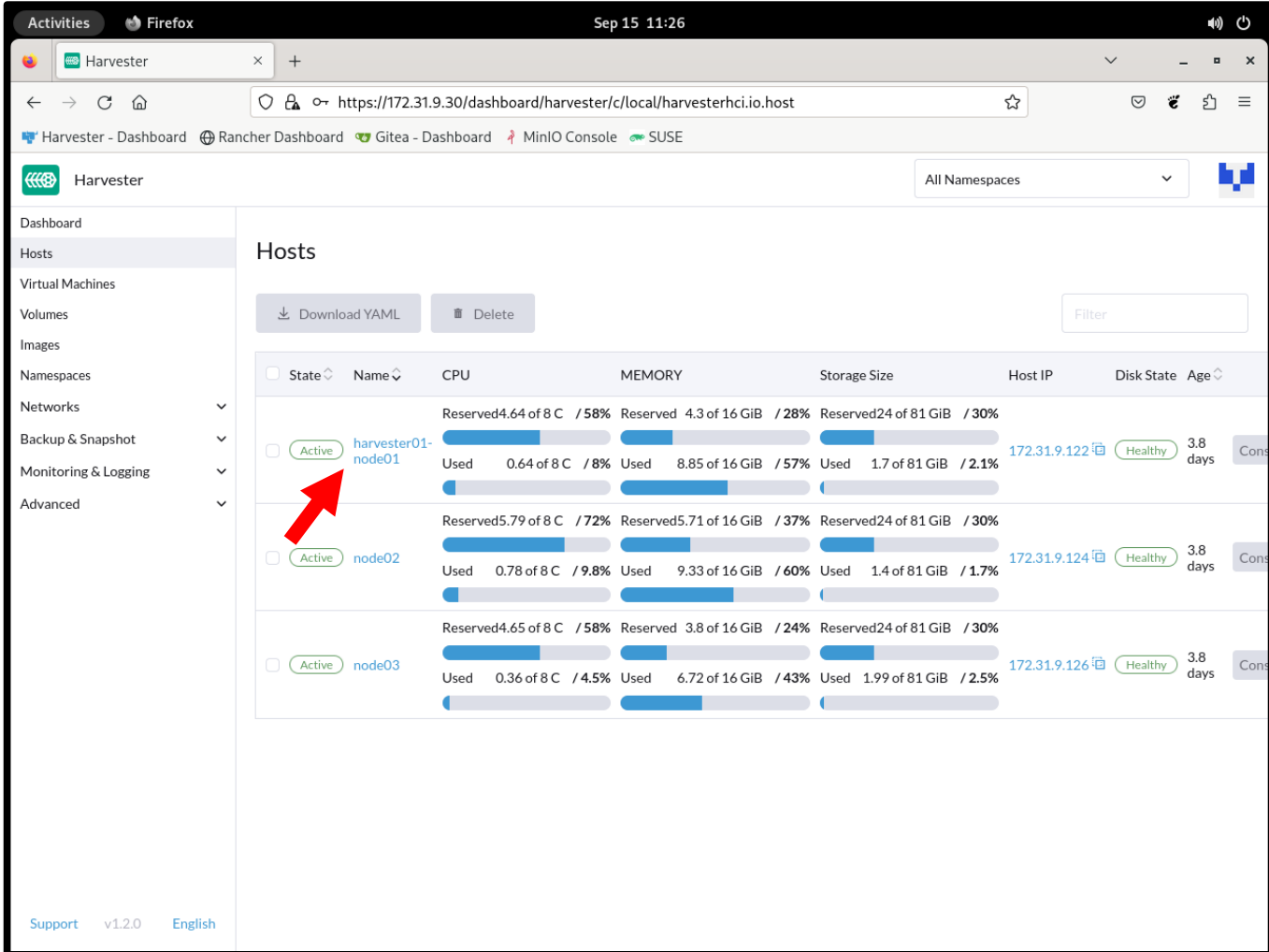
Cluster Host – Edit Config – Example: Basic

- 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.



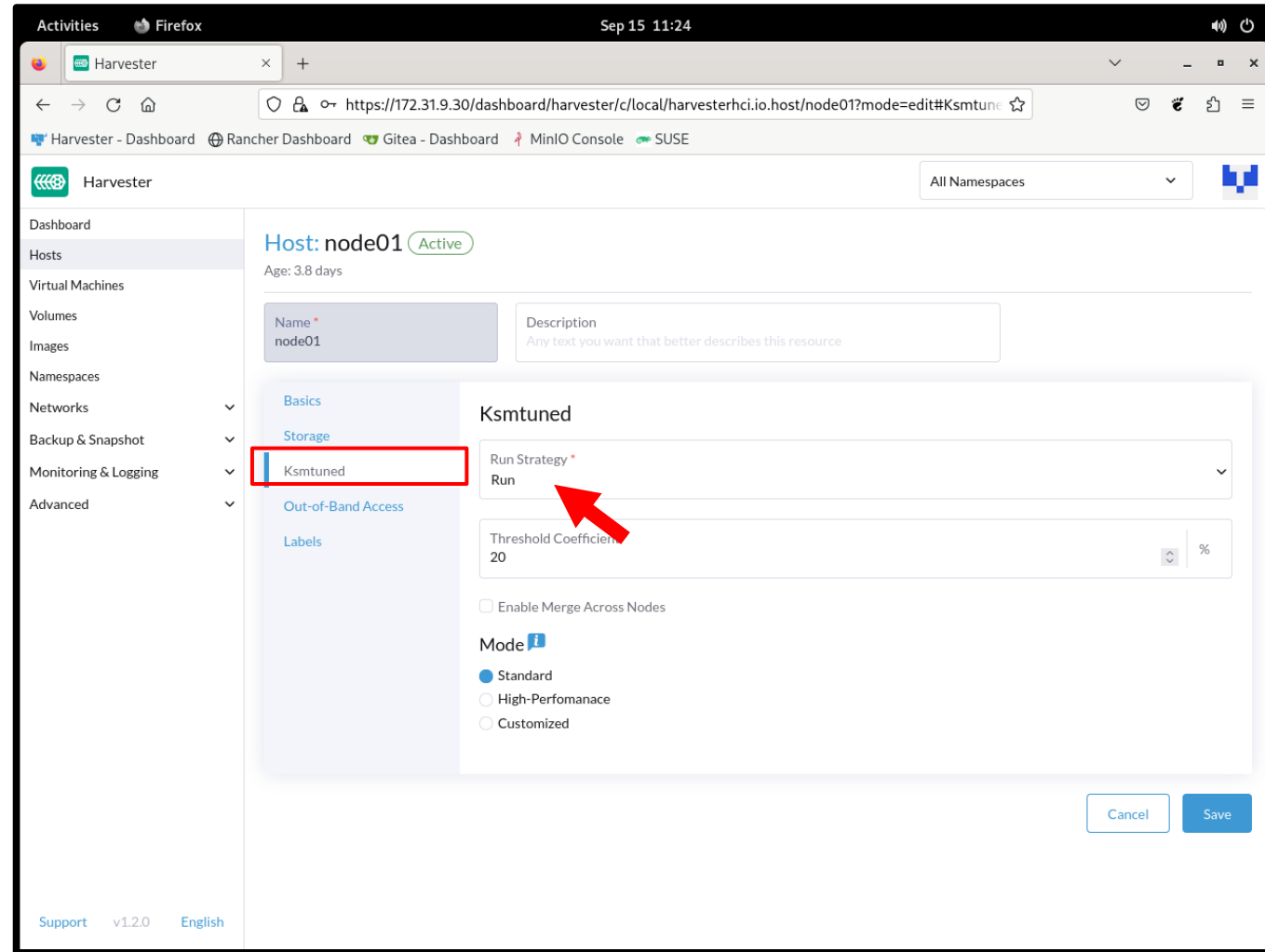
The screenshot shows the Harvester web interface in a Firefox browser. The left sidebar contains navigation links: Dashboard, Hosts, Virtual Machines, Volumes, Images, Namespaces, Networks, Backup & Snapshot, Monitoring & Logging, and Advanced. The main content area is titled 'Hosts' and displays a table of hosts. A red arrow points to the 'harvester01-node01' host name in the first row, indicating that a custom name has been assigned. The table columns include State, Name, CPU, MEMORY, Storage Size, Host IP, Disk State, and Age.

State	Name	CPU	MEMORY	Storage Size	Host IP	Disk State	Age
Active	harvester01-node01	Reserved 4.64 of 8 C / 58% Used 0.64 of 8 C / 8%	Reserved 4.3 of 16 GiB / 28% Used 8.85 of 16 GiB / 57%	Reserved 24 of 81 GiB / 30% Used 1.7 of 81 GiB / 2.1%	172.31.9.122	Healthy	3.8 days
Active	node02	Reserved 5.79 of 8 C / 72% Used 0.78 of 8 C / 9.8%	Reserved 5.71 of 16 GiB / 37% Used 9.33 of 16 GiB / 60%	Reserved 24 of 81 GiB / 30% Used 1.4 of 81 GiB / 1.7%	172.31.9.124	Healthy	3.8 days
Active	node03	Reserved 4.65 of 8 C / 58% Used 0.36 of 8 C / 4.5%	Reserved 3.8 of 16 GiB / 24% Used 6.72 of 16 GiB / 43%	Reserved 24 of 81 GiB / 30% Used 1.99 of 81 GiB / 2.5%	172.31.9.126	Healthy	3.8 days



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

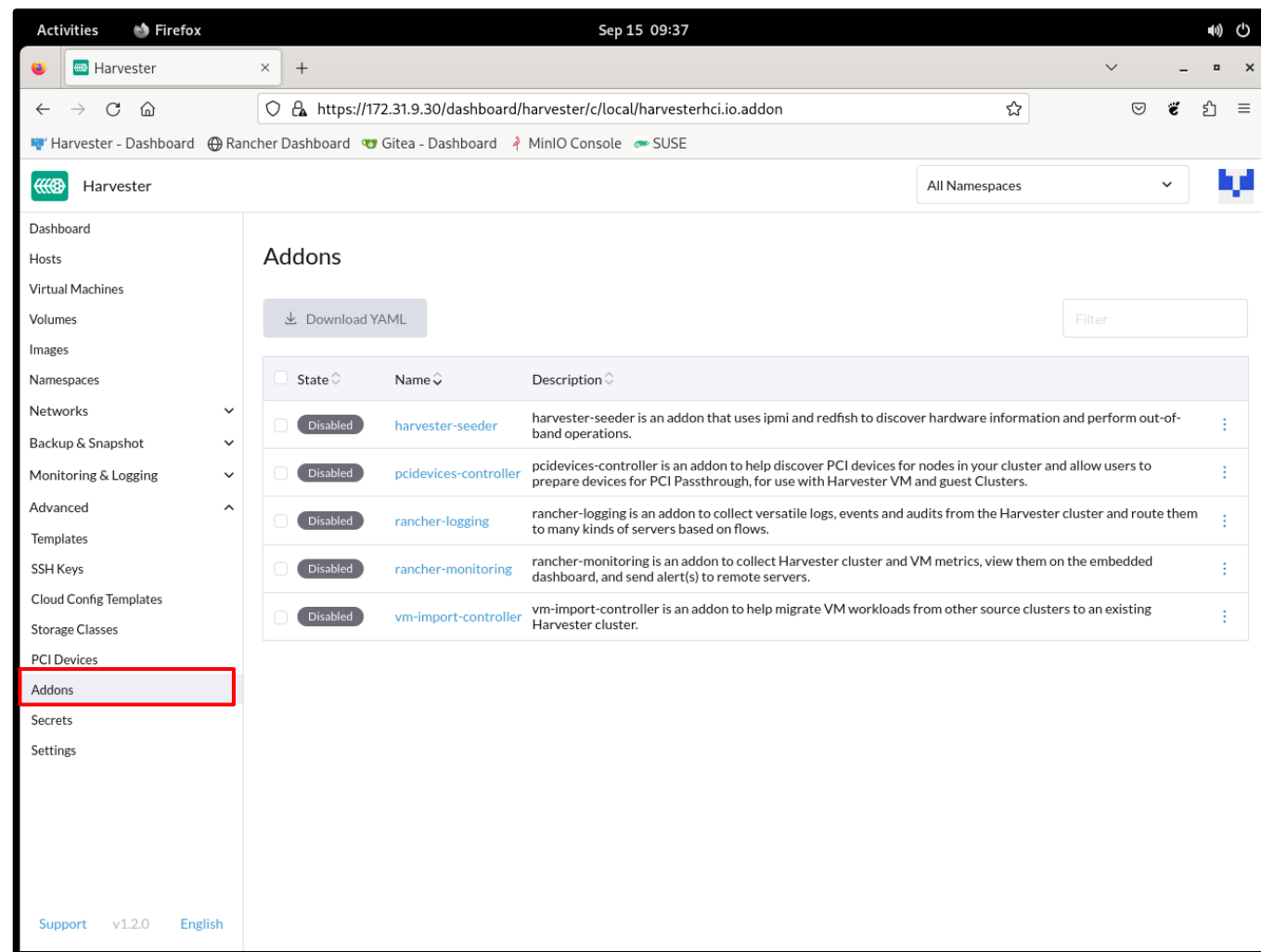


Harvester Addons



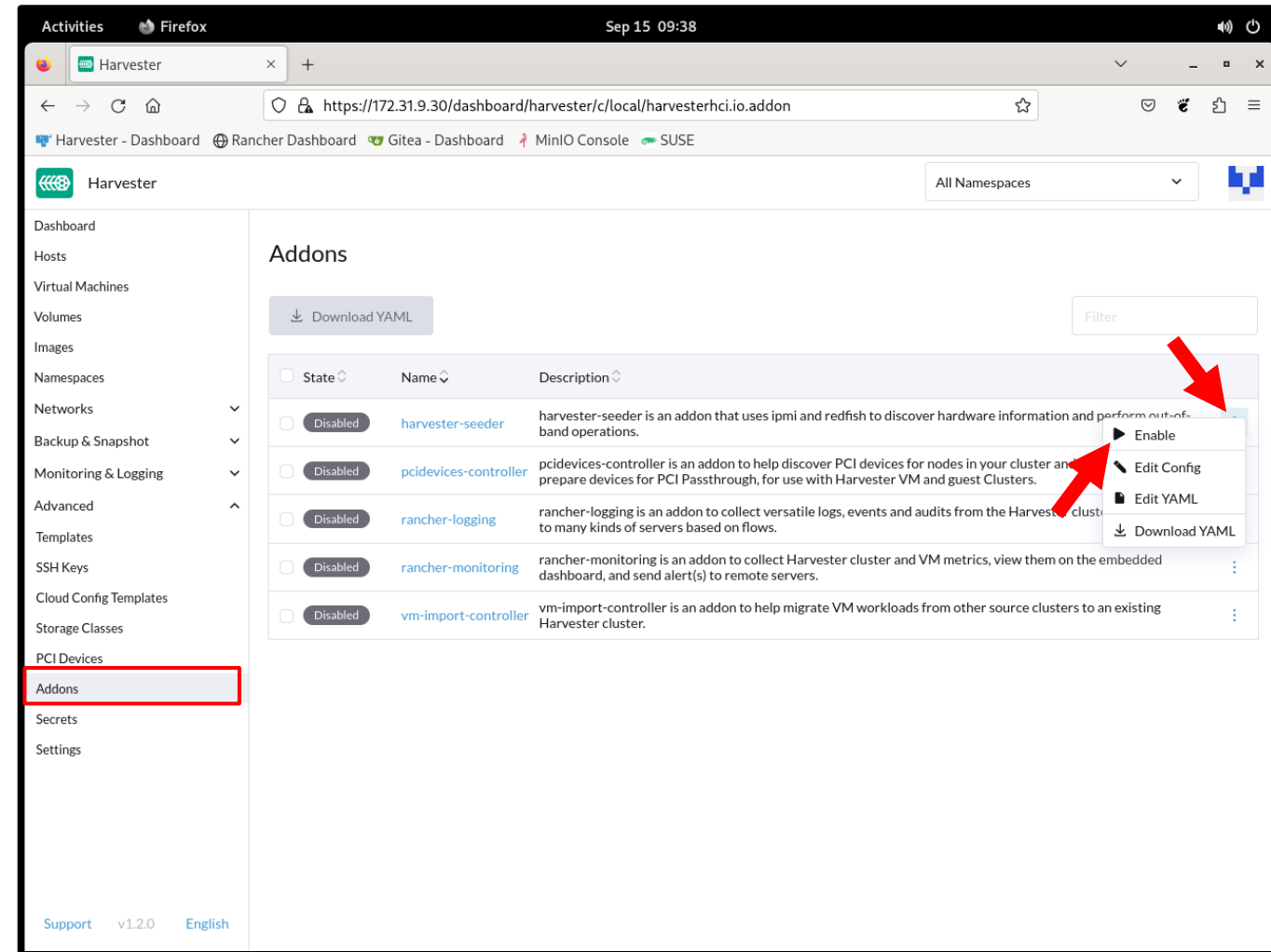
Harvester Addons

- 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.



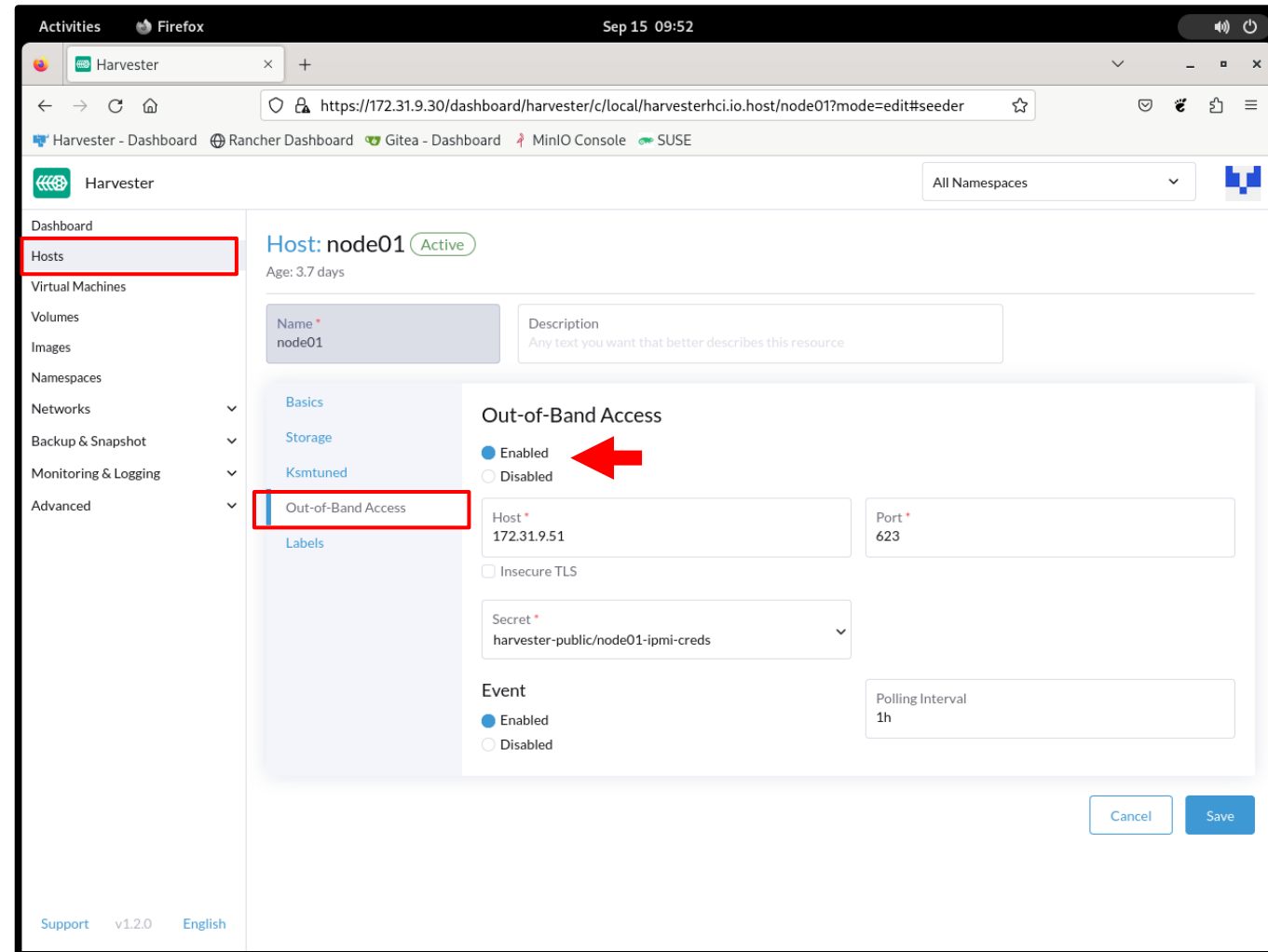
Enable/Disable Addons

- 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 be 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.

Activities Firefox Sep 15 09:49

Harvester

https://172.31.9.30/dashboard/harvester/c/local/harvesterhci.io.dashboard#vm

Harvester - Dashboard Rancher Dashboard Gitea - Dashboard MinIO Console SUSE

Harvester

All Namespaces

Dashboard

Hosts

Virtual Machines

Volumes

Images

Namespaces

Networks

Backup & Snapshot

Monitoring & Logging

Advanced

Harvester Cluster: local

Version: v1.2.0 Created: 3.7 days ago

Monitoring Addon is disabled now, click [here](#) to enable it.

3 Hosts

0 Virtual Machines

1 VM Network

2 Images

0 Volumes

3 Disks

Capacity

CPU

Reserved 12.81 / 24 53.35%

Used 11.11 / 24 46.28%

Memory

Reserved 11 / 47 GiB 23.40%

Used 23 / 47 GiB 48.94%

Storage

Reserved 73 / 243 GiB 30.00%

Used 3.92 / 243 GiB 1.61%

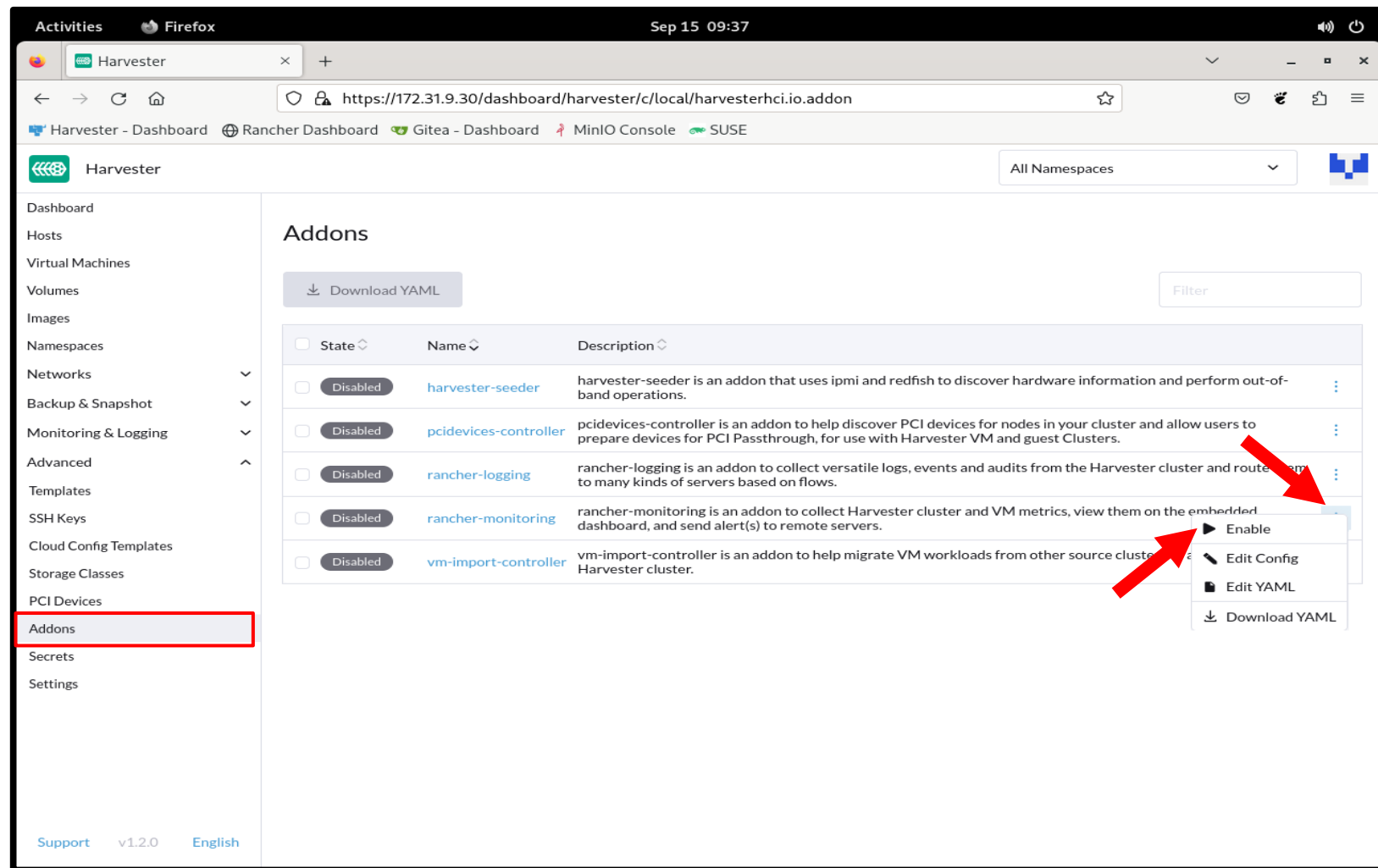
Support v1.2.0 English

Events



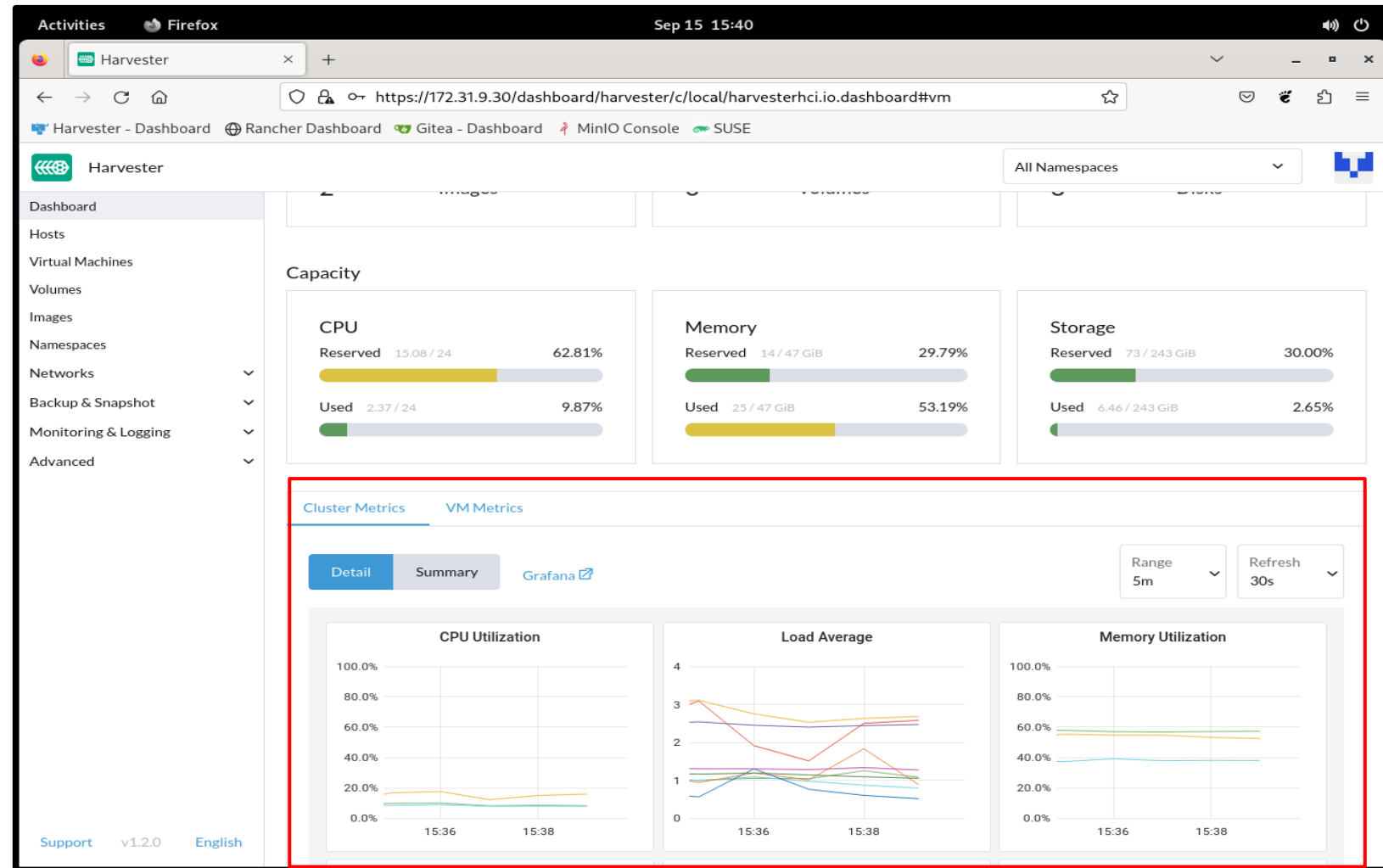
Enable the rancher-monitoring Addon

- 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**



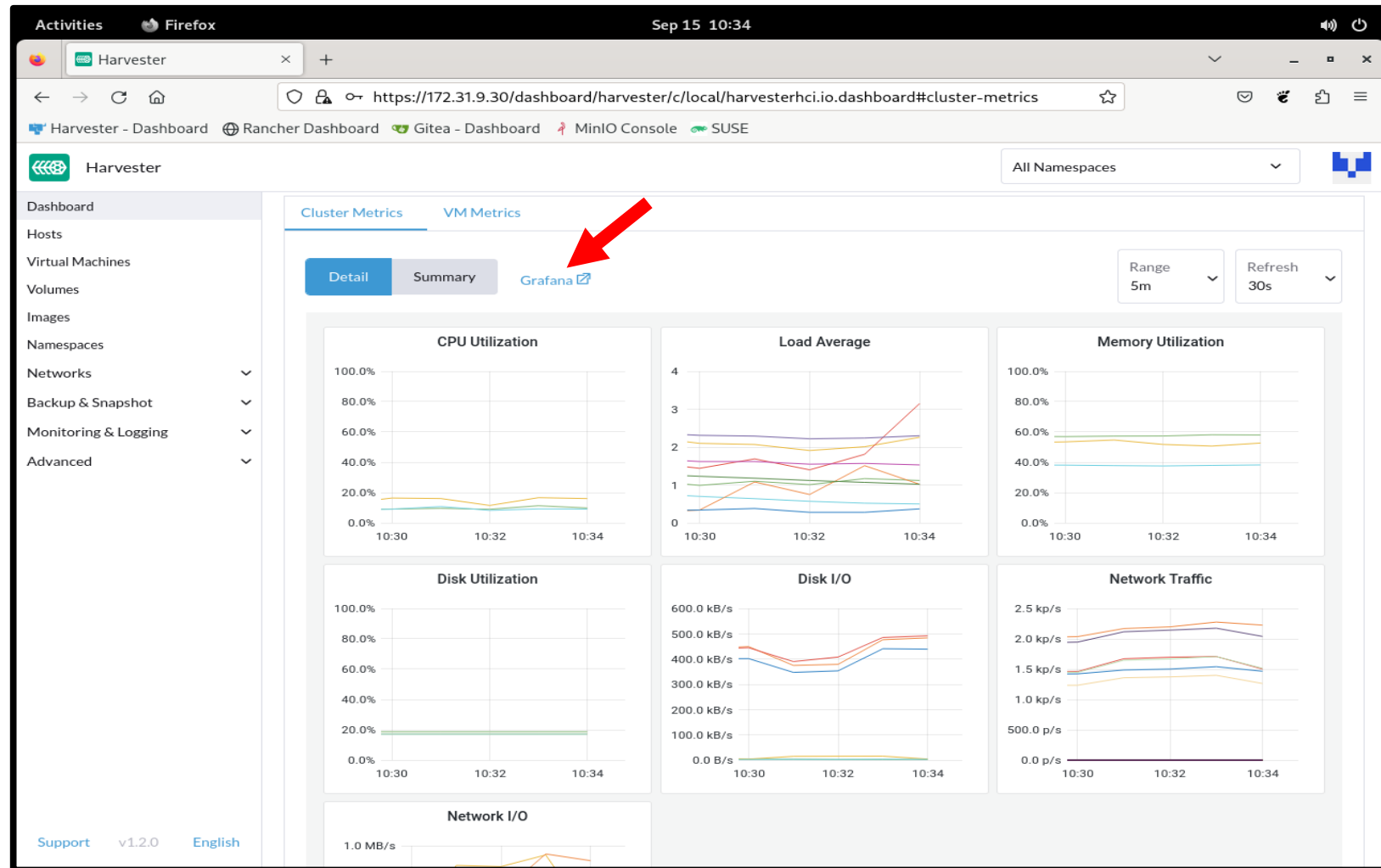
Cluster Metrics Displayed

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



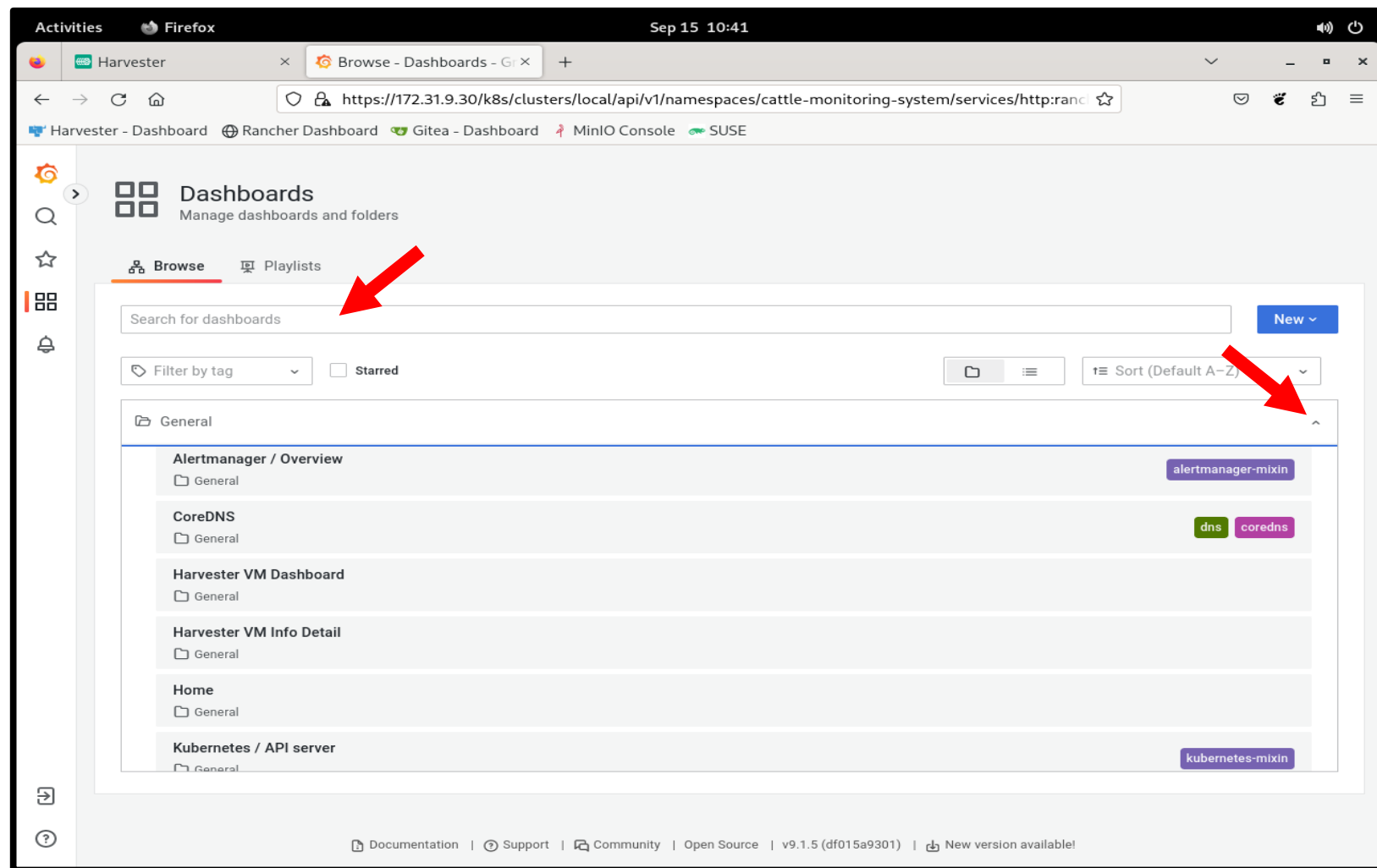
Cluster Metrics

- 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



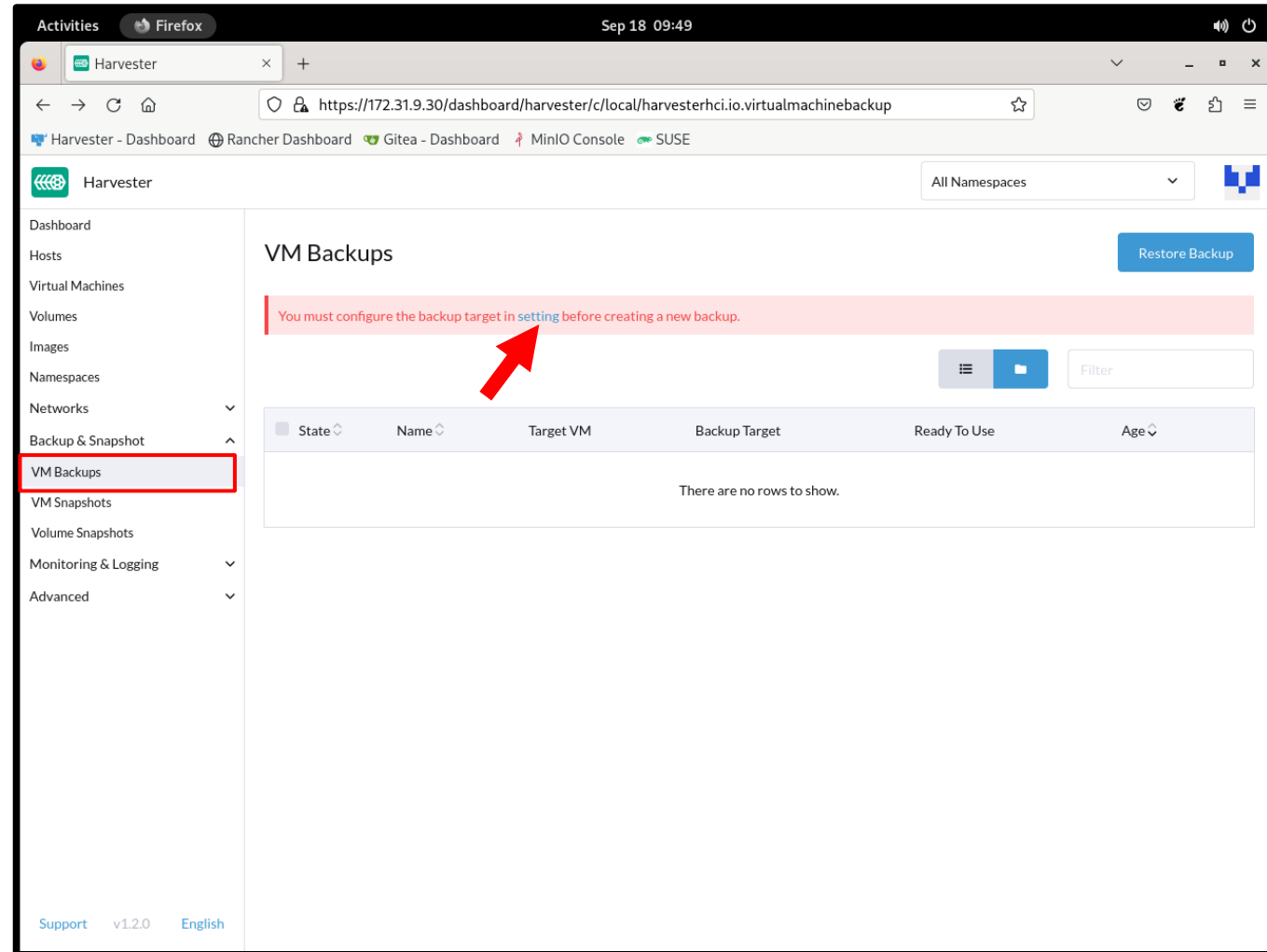
Backup Target

- 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.



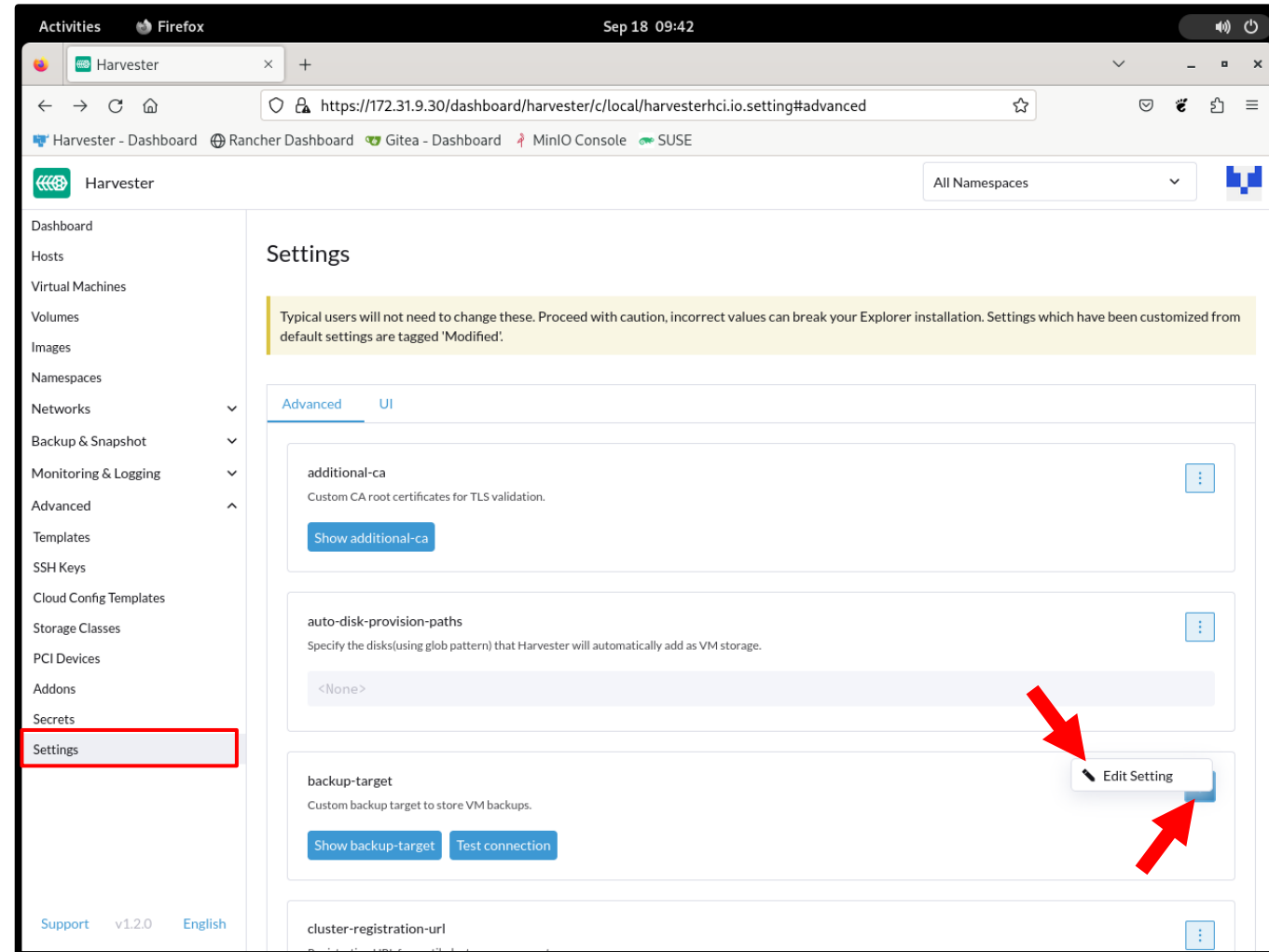
Backups

- 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

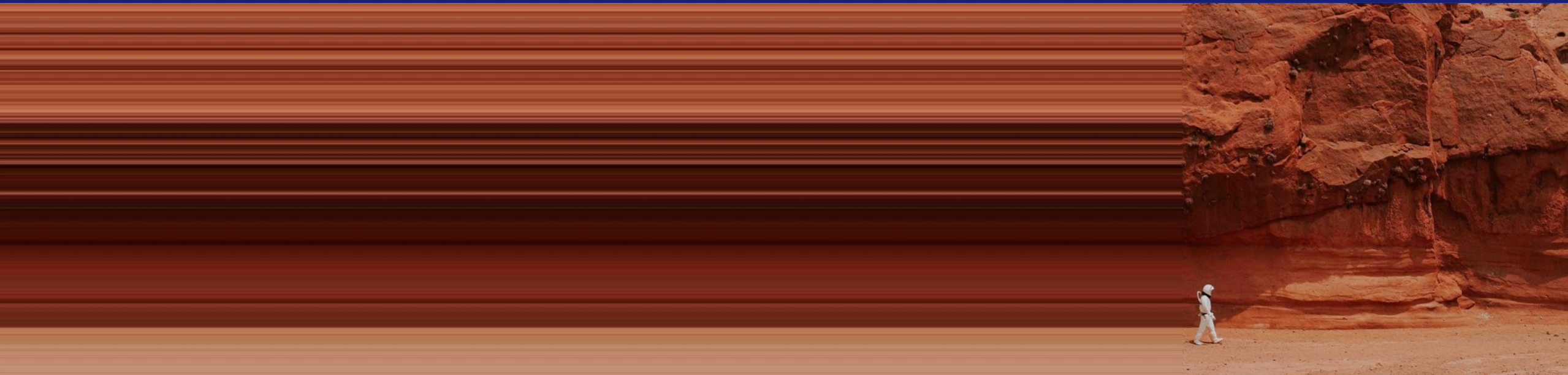


- Perform basic configuration of a Harvester cluster
 - Cluster nodes can be individually configured from the Hosts screen for things such as their custom name, Ksmtuned and other host specific configuration.
 - Harvester addons can be enabled to extend the functionality of the cluster.
 - Cluster metrics 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.



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

- 1 Overview of Networking in a Harvester Cluster
- 2 Cluster Network Definitions
- 3 Network Configs
- 4 VM Networks



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

Cluster Network

Network Config

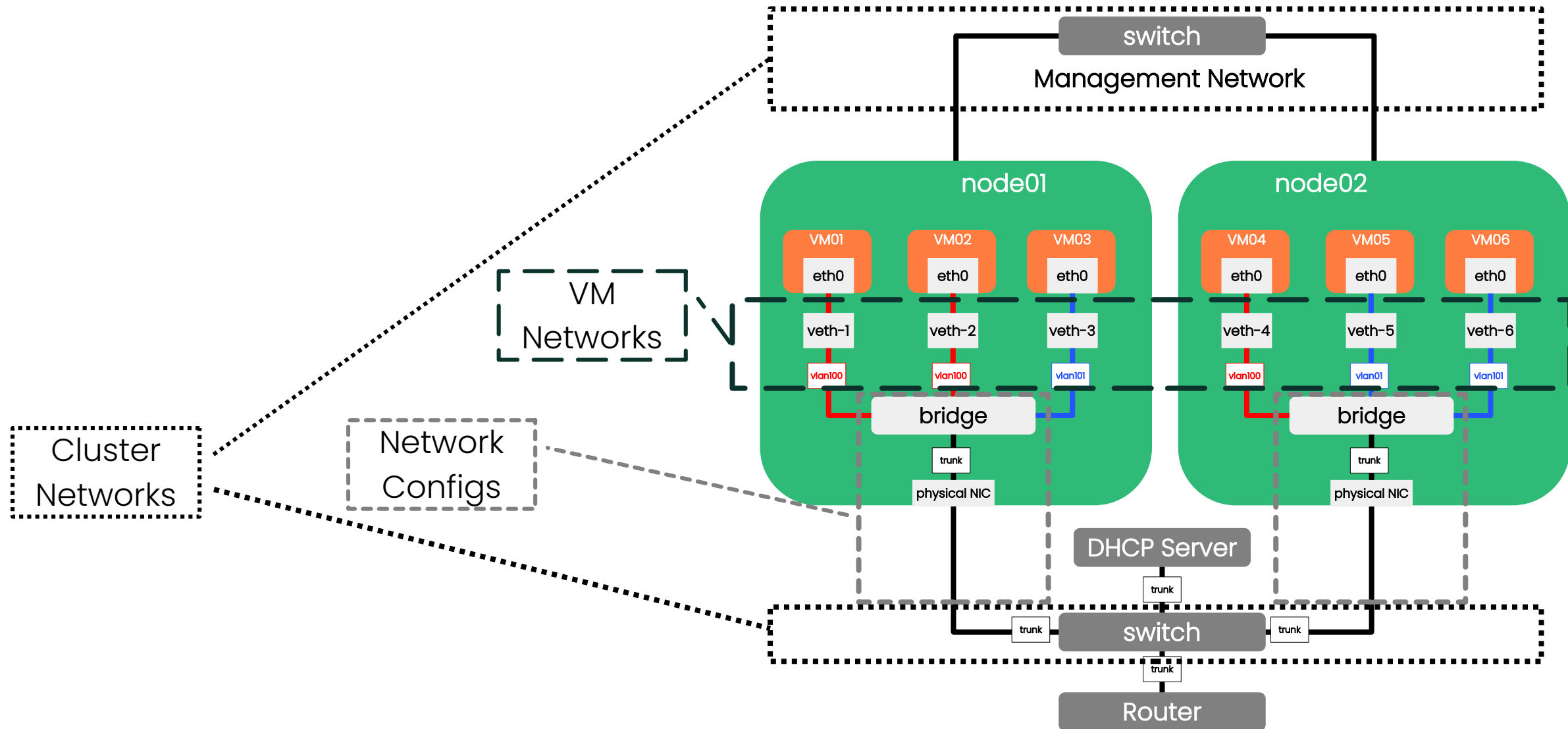
VM Network

Description

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

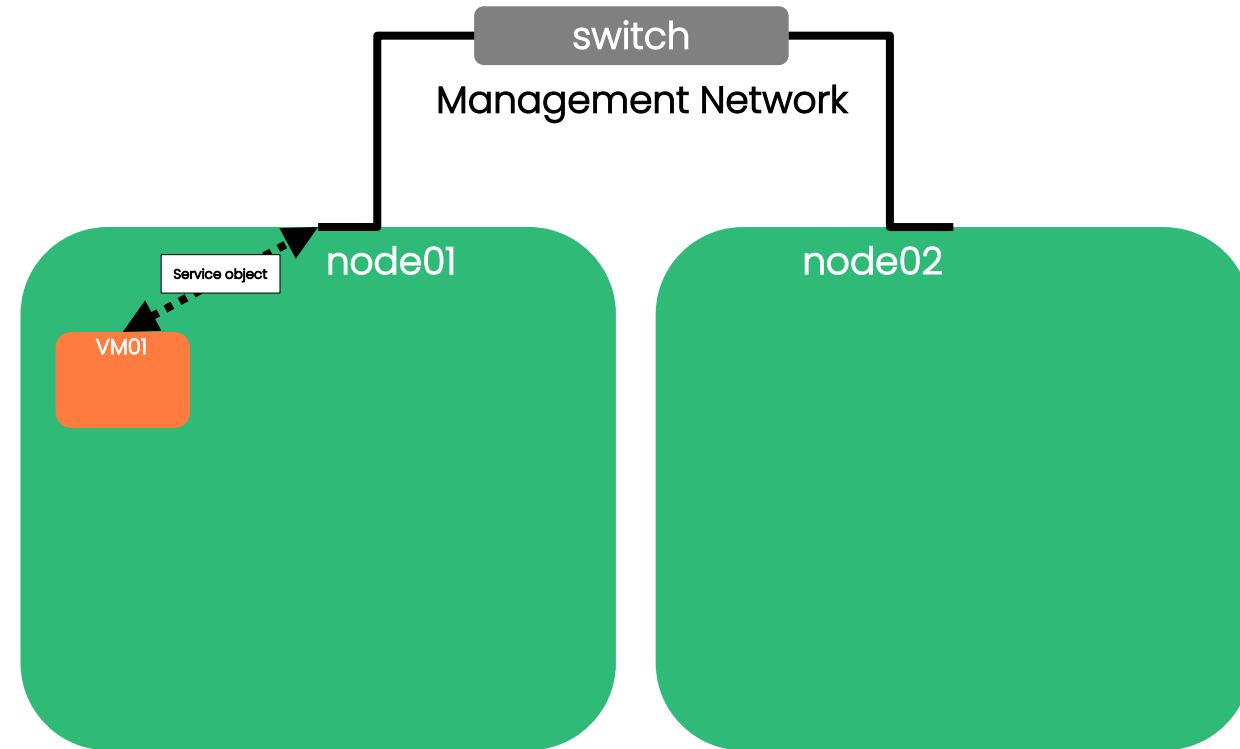


Harvester Network Examples Diagram



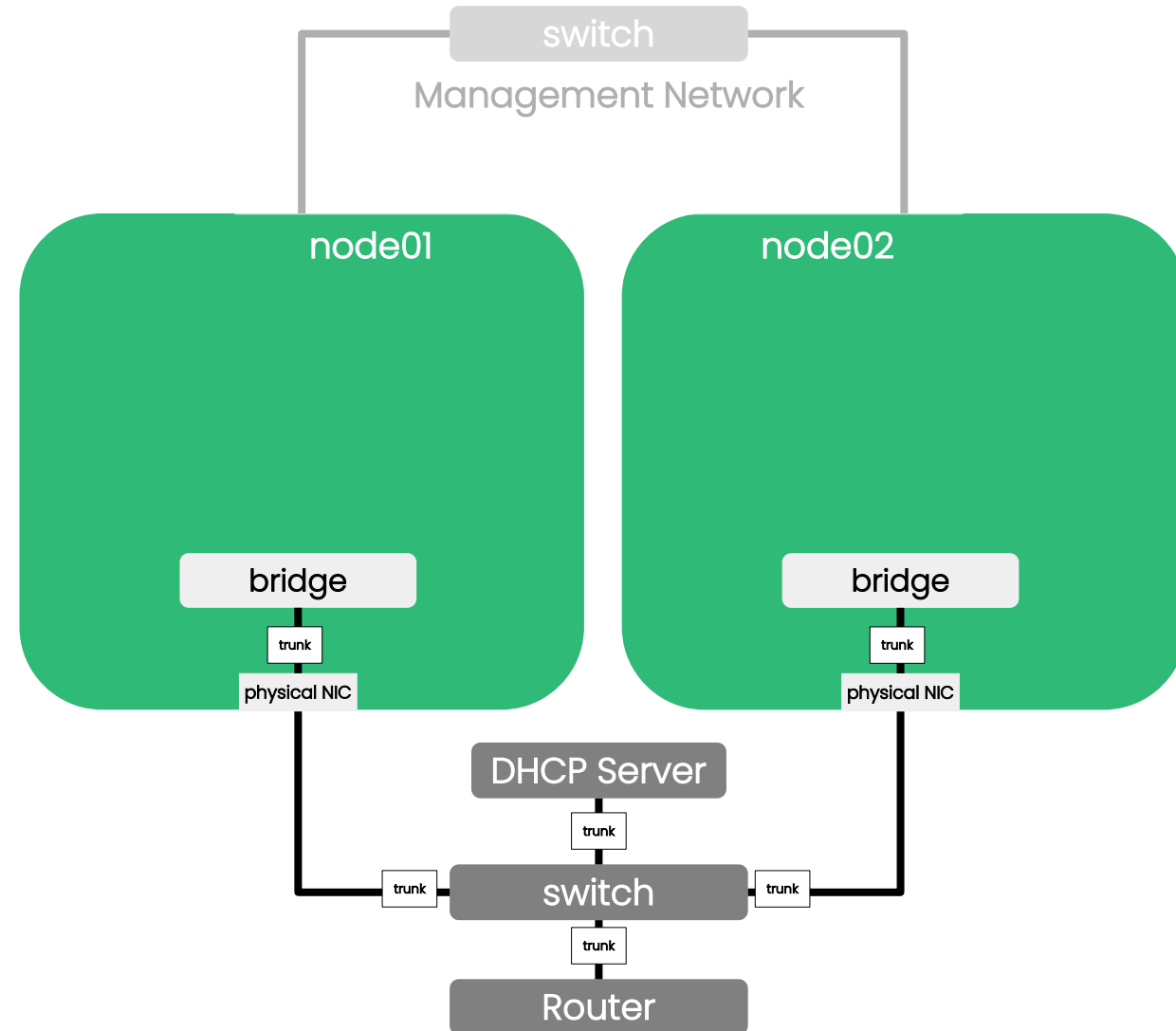
Cluster Network – Management Network

- 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.



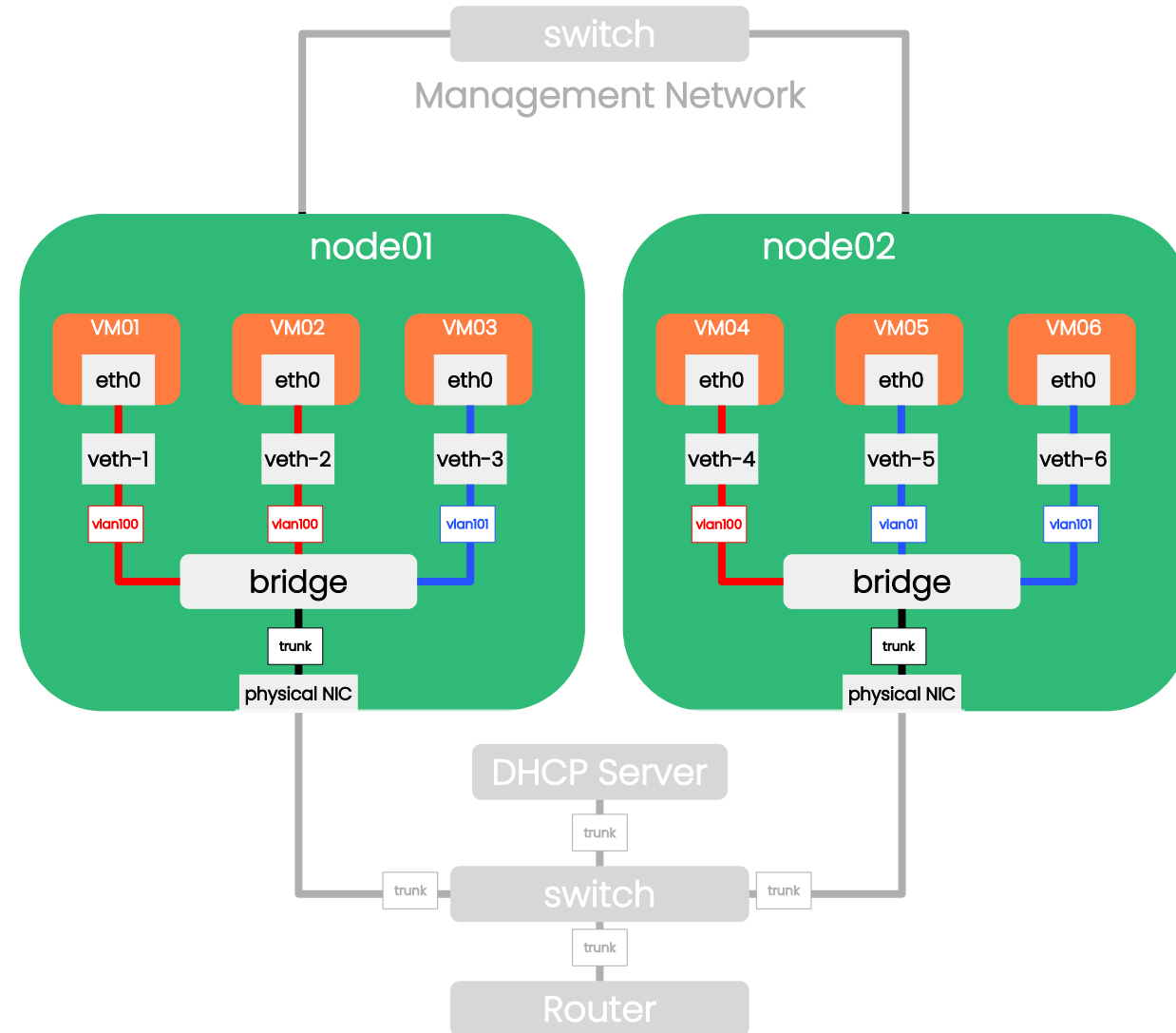
Cluster Network – VLAN Trunk Network

- 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

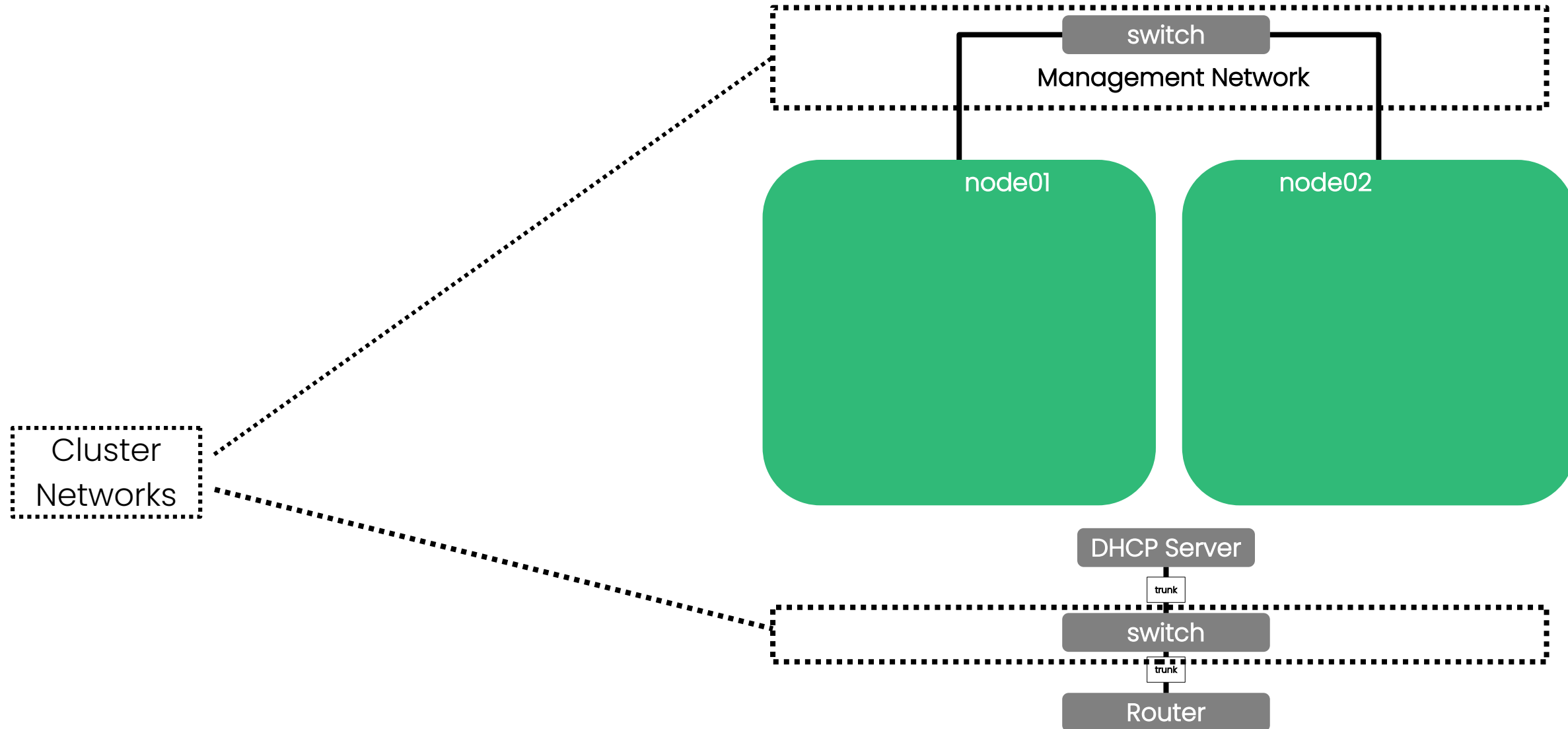
- VM Networks are implemented as VLANs or as untagged networks.
- VM Networks can be added to any Cluster Network.



Cluster Network Definitions

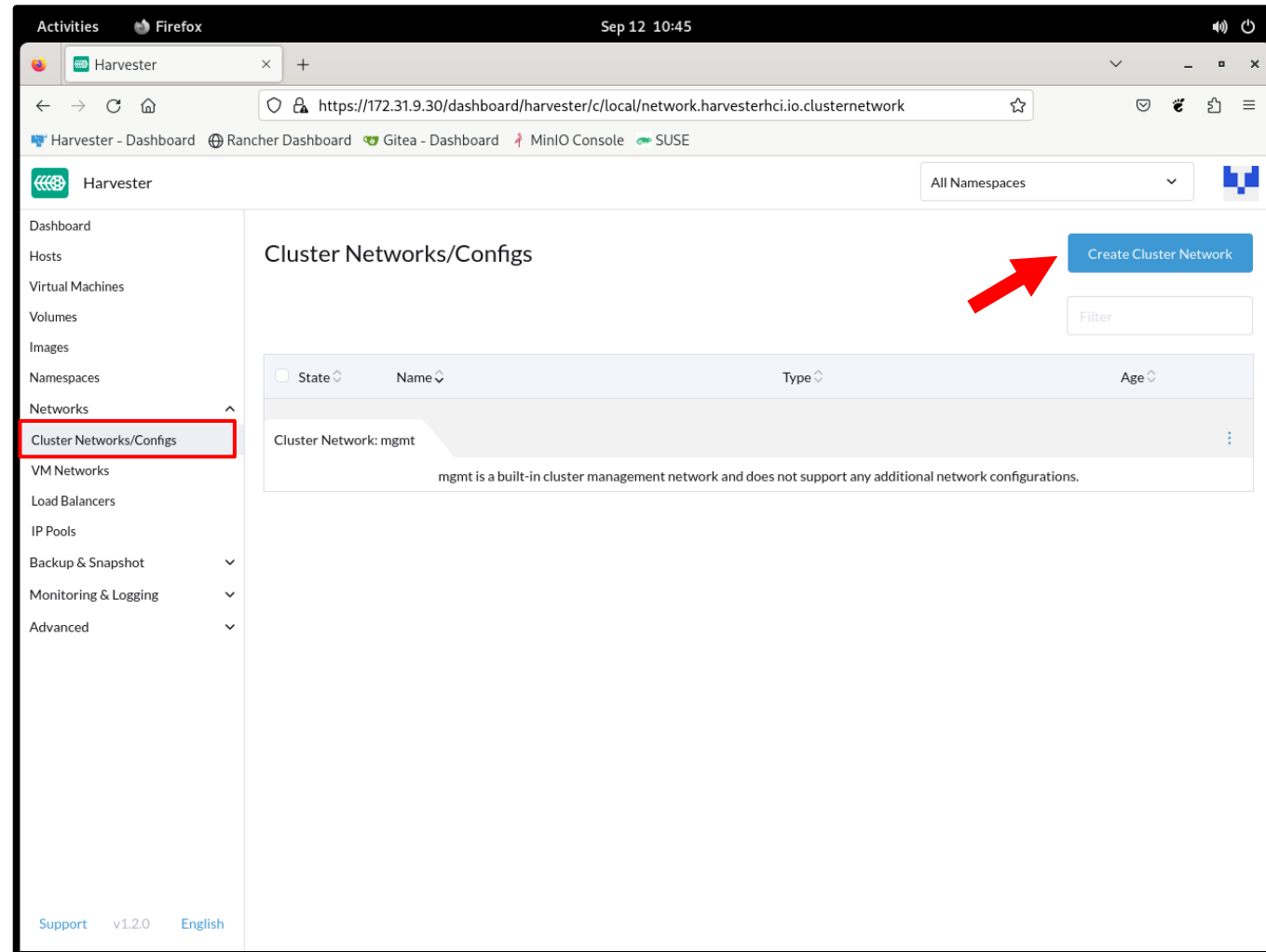


Cluster Networks



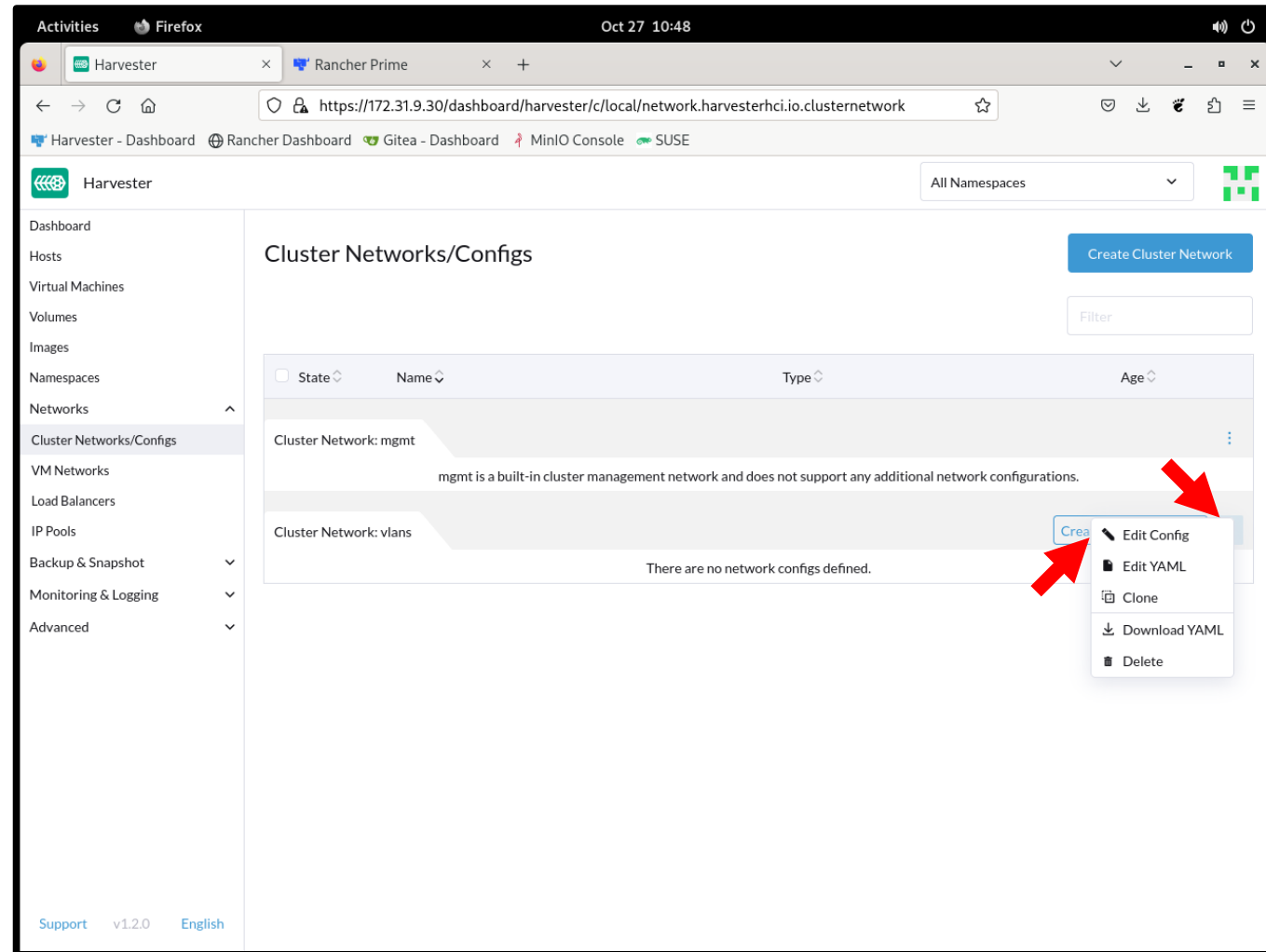
Create a Cluster Network

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



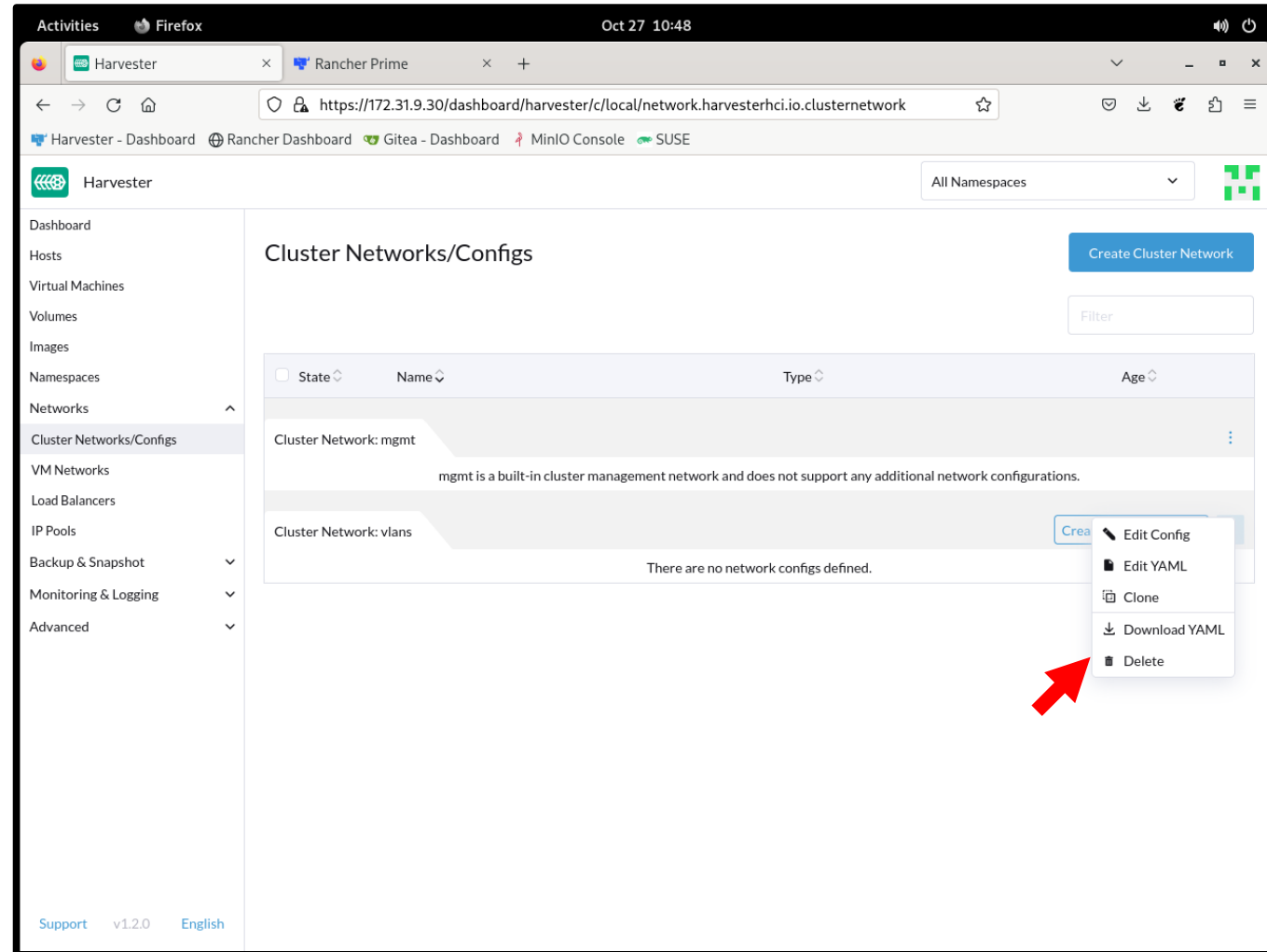
Modify Cluster Networks

- 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

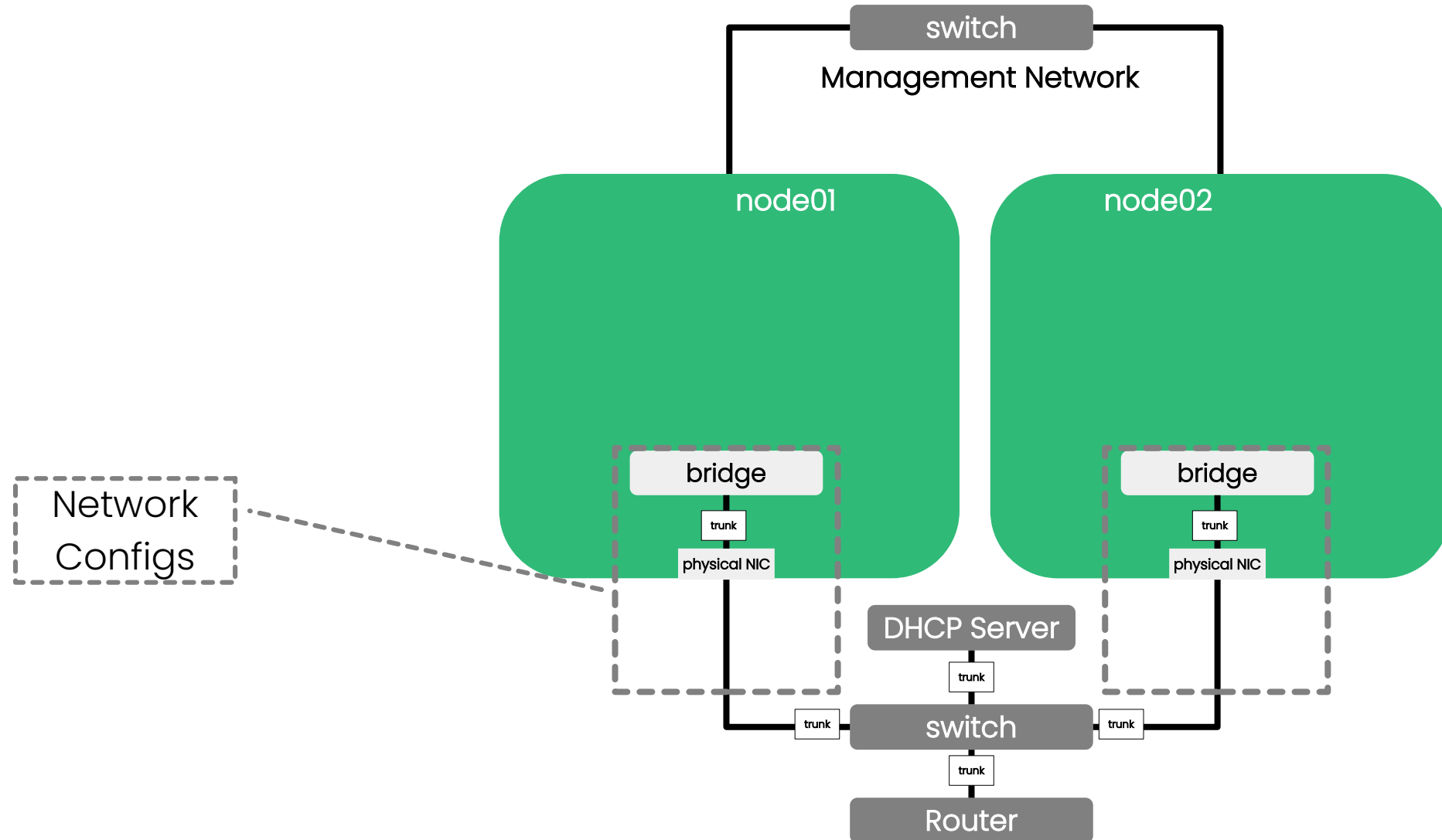
- Cluster Networks, other than the management (mgmt) Cluster Network, can also be deleted.
- Cluster Networks cannot be deleted if they have Network Configs created on them, the Network Configs must be deleted first.



Network Configs

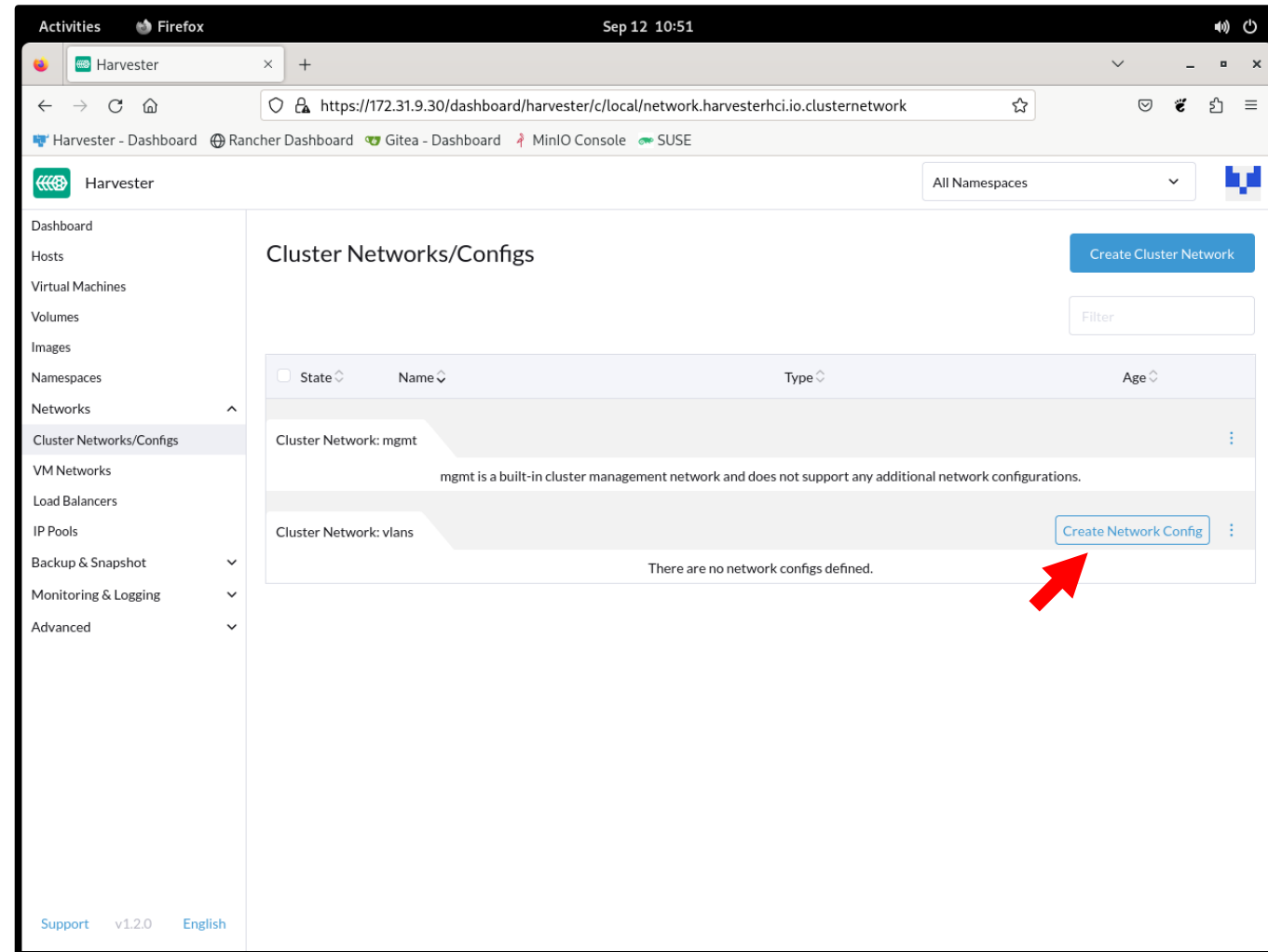


Network Configs



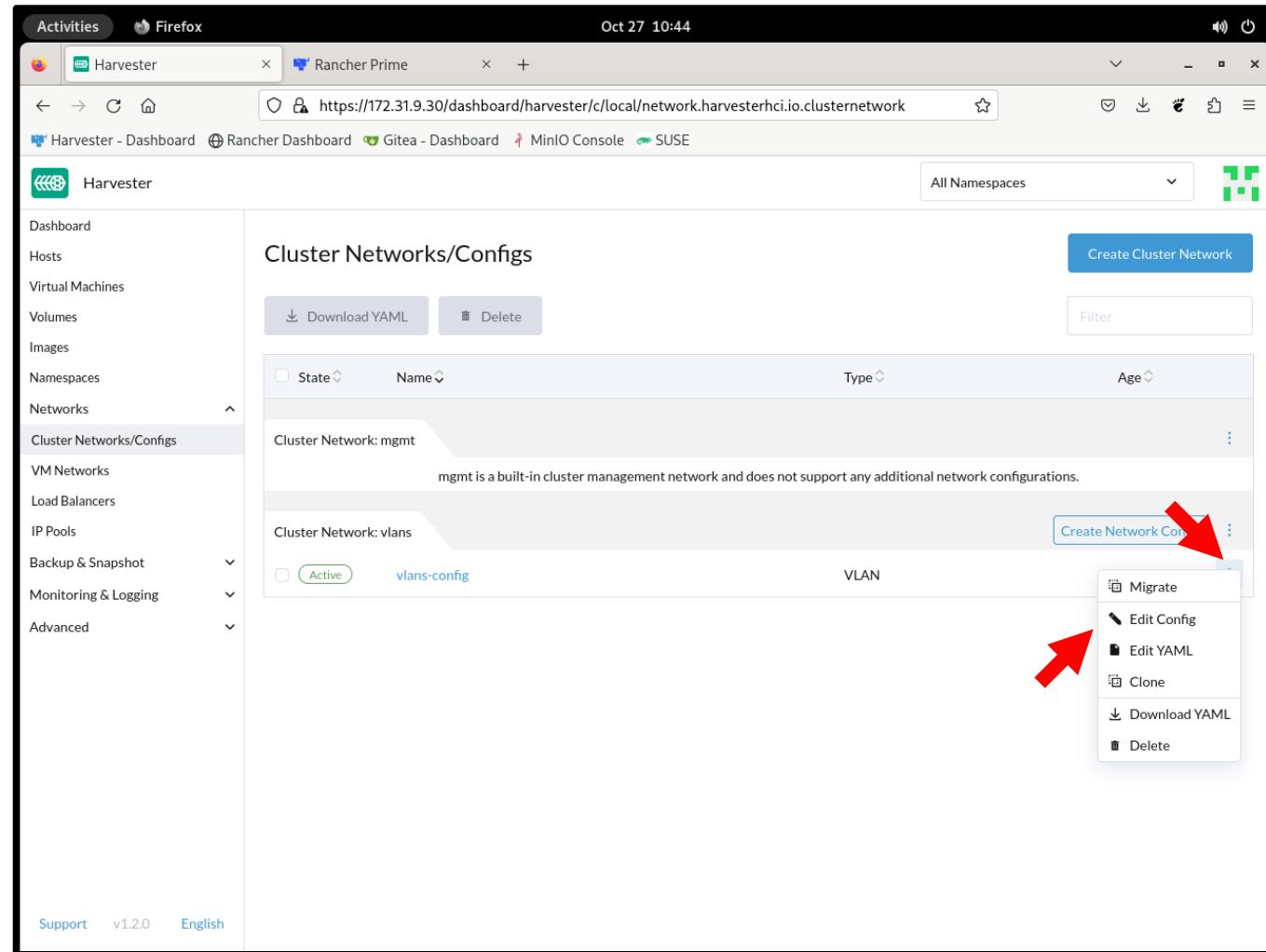
Create a Network Config

- 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**



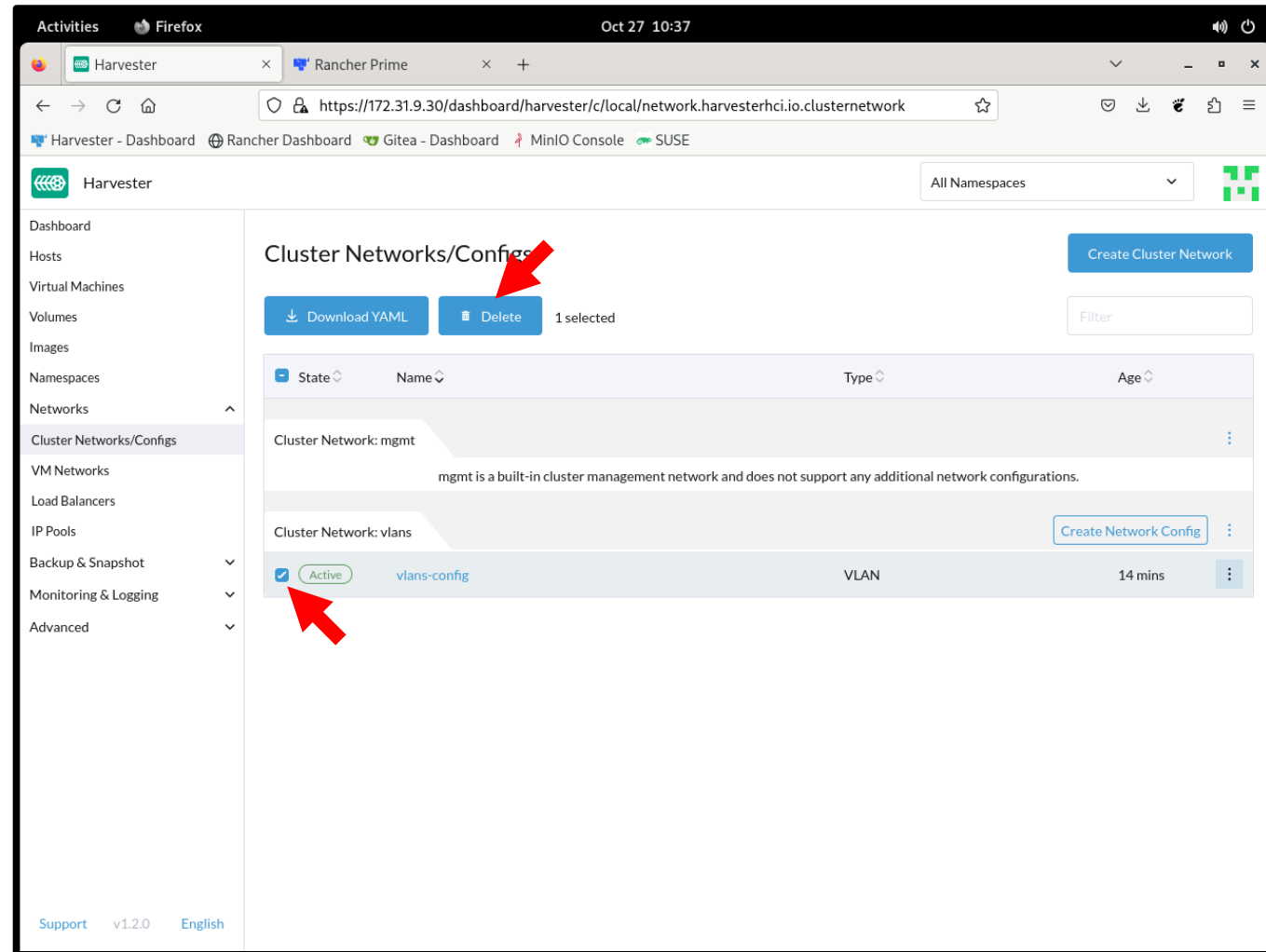
Modify Network Config

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



Delete Network Config

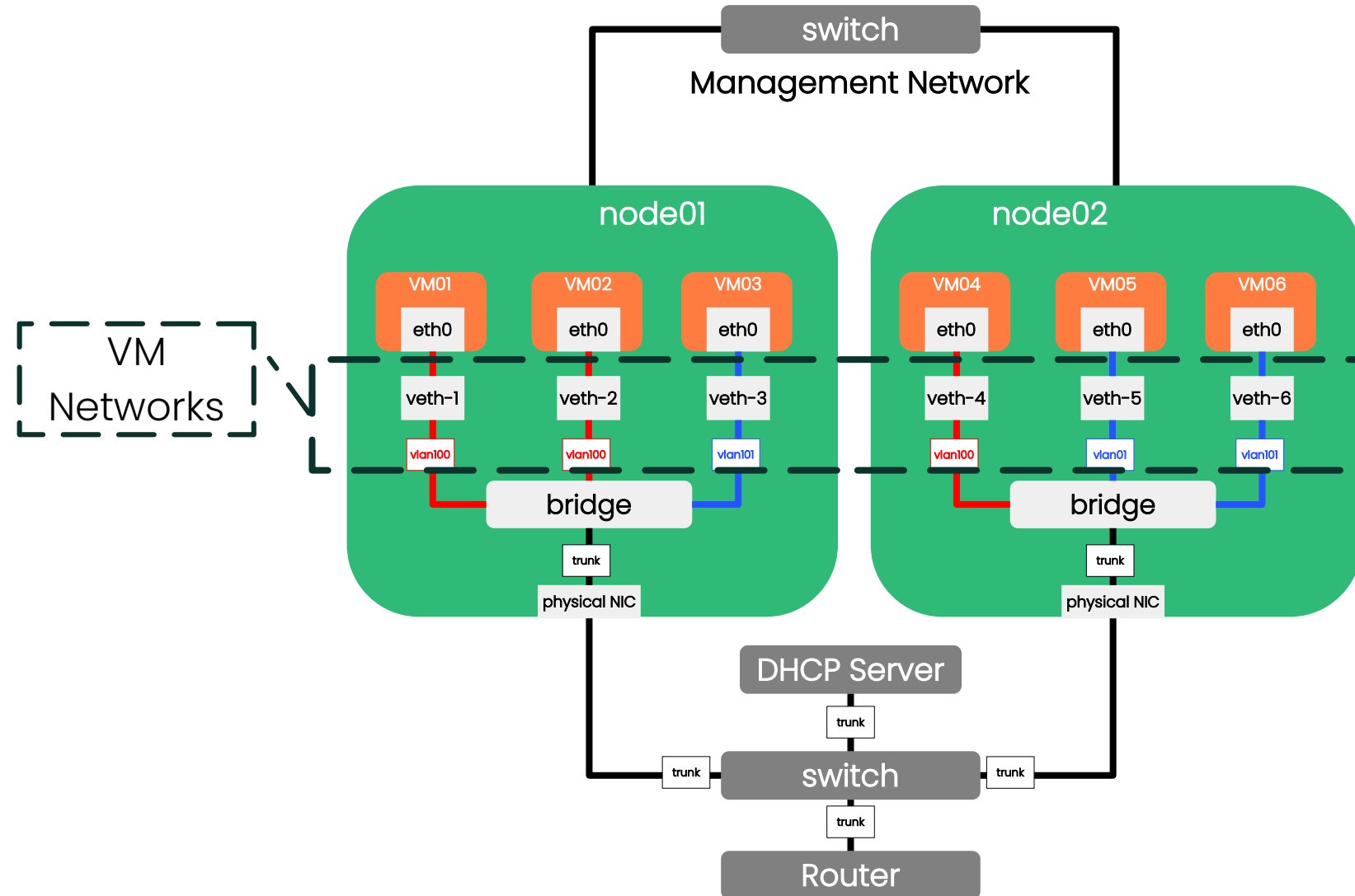
- Network Configs can also be deleted.
- Select the **check box** next to the Network Config and click: **Delete**



VM Networks

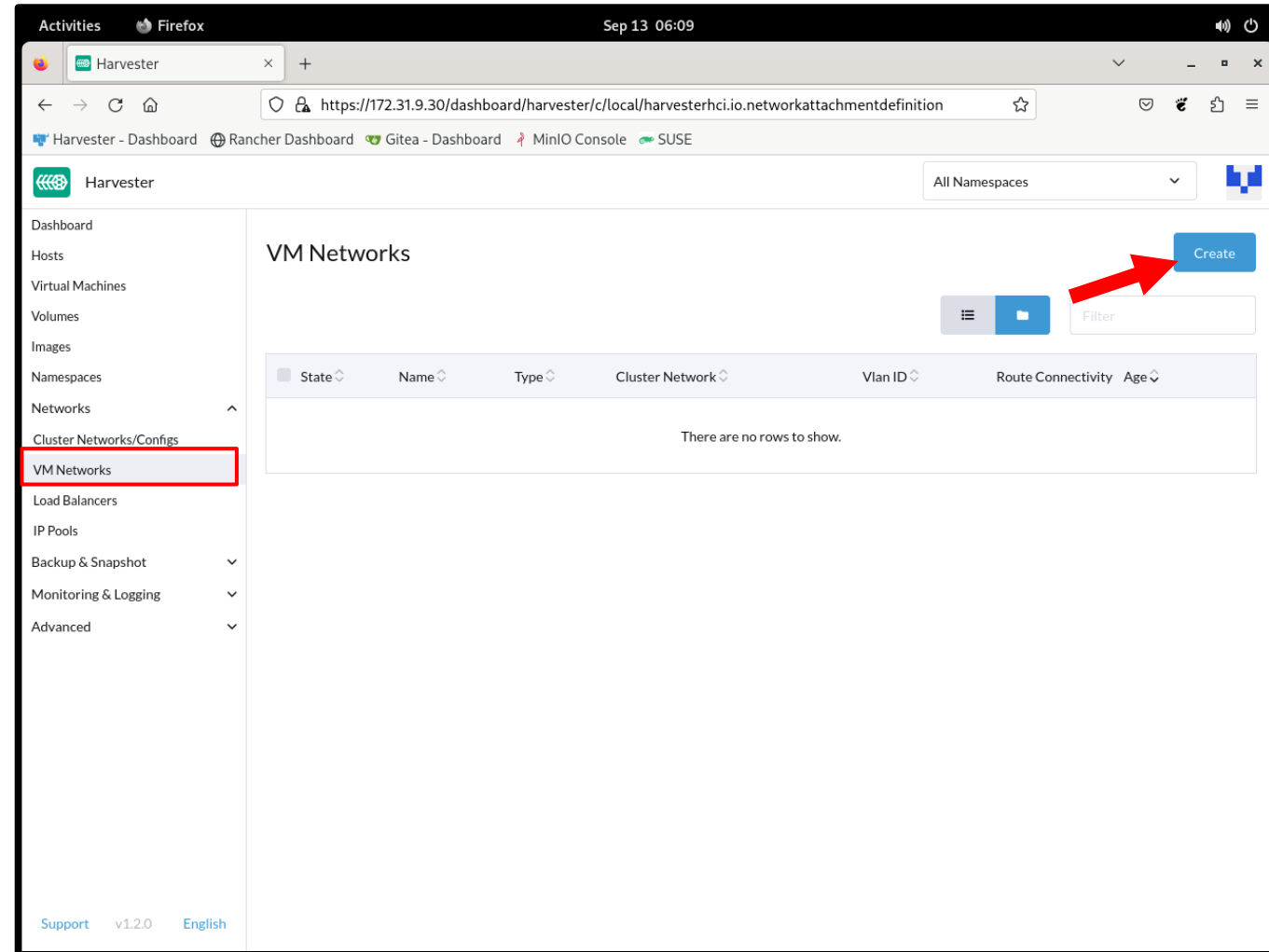


VM Networks



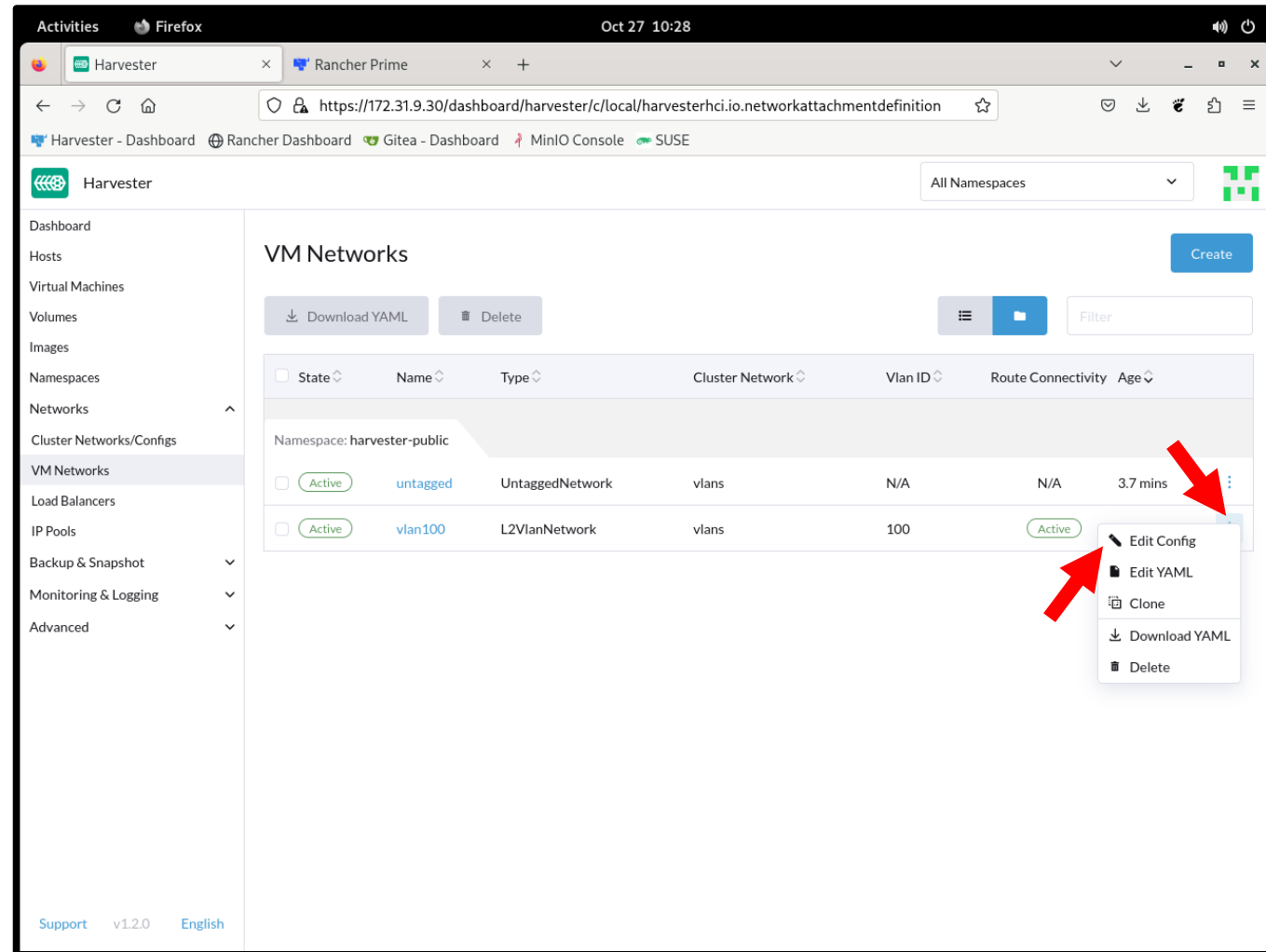
View/Create Networks

- 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.



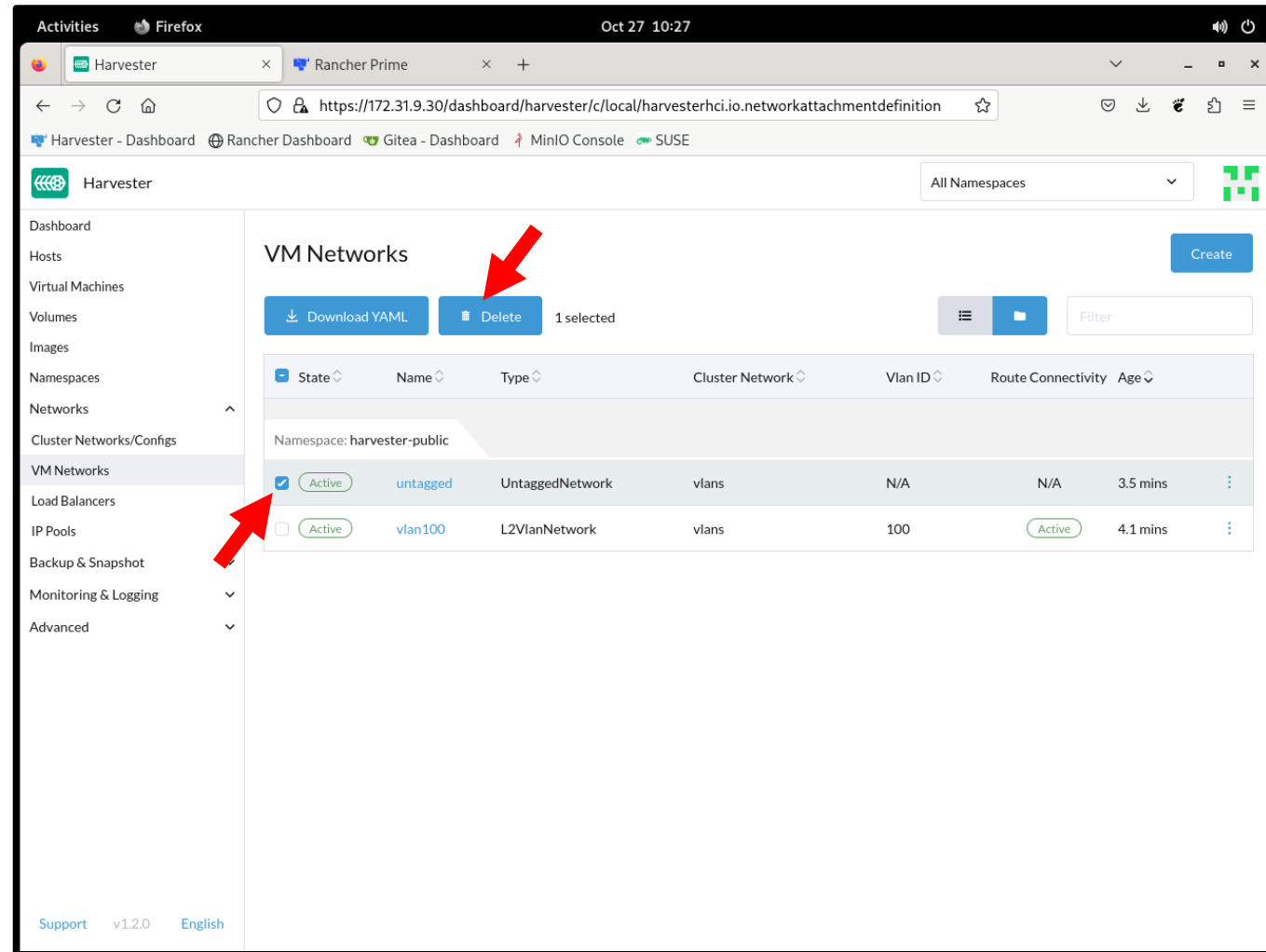
The screenshot shows the Harvester web interface in a Firefox browser. The left sidebar contains a navigation menu with options: Dashboard, Hosts, Virtual Machines, Volumes, Images, Namespaces, Networks (selected), Cluster Networks/Configs, VM Networks, Load Balancers, IP Pools, Backup & Snapshot, Monitoring & Logging, and Advanced. The main content area is titled 'VM Networks' and displays a table of networks for the 'harvester-public' namespace. The table has columns for State, Name, Type, Cluster Network, Vlan ID, Route Connectivity, and Age. Two networks are listed: 'untagged' (UntaggedNetwork) and 'vlan100' (L2VlanNetwork). A context menu is open for the 'vlan100' network, showing options: Edit Config, Edit YAML, Clone, Download YAML, and Delete. Red arrows point to the '3 dot' menu icon and the 'Edit Config' option.

State	Name	Type	Cluster Network	Vlan ID	Route Connectivity	Age
Active	untagged	UntaggedNetwork	vlan	N/A	N/A	3.7 mins
Active	vlan100	L2VlanNetwork	vlan	100	Active	



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.



The screenshot shows the Harvester web interface in a Firefox browser. The left sidebar contains a navigation menu with options: Dashboard, Hosts, Virtual Machines, Volumes, Images, Namespaces, Networks, Cluster Networks/Configs, VM Networks, Load Balancers, IP Pools, Backup & Snapshot, Monitoring & Logging, and Advanced. The 'VM Networks' option is selected and highlighted. The main content area is titled 'VM Networks' and includes a 'Create' button in the top right. Below the title are two buttons: 'Download YAML' and 'Delete'. A red arrow points to the 'Delete' button, and a status indicator shows '1 selected'. A table lists the VM Networks in the 'harvester-public' namespace. The first row, 'untagged' (Type: UntaggedNetwork), has its selection checkbox checked (indicated by a red arrow) and is marked as 'Active'. The second row, 'vlan100' (Type: L2VlanNetwork), has its selection checkbox unchecked. The table columns are: State, Name, Type, Cluster Network, Vlan ID, Route Connectivity, and Age.

State	Name	Type	Cluster Network	Vlan ID	Route Connectivity	Age
Active	untagged	UntaggedNetwork	vlans	N/A	N/A	3.5 mins
Active	vlan100	L2VlanNetwork	vlans	100	Active	4.1 mins



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

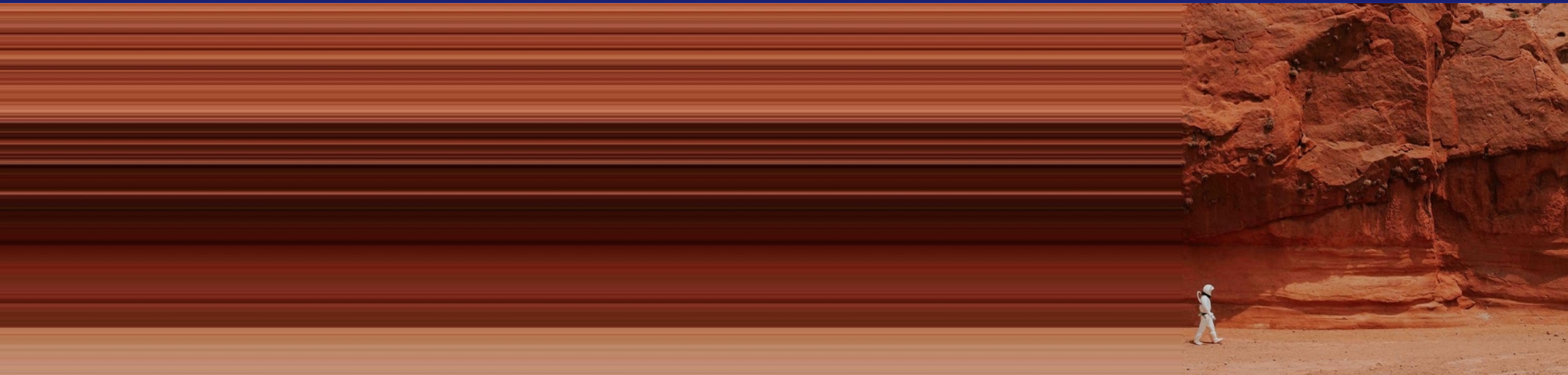


- Create/modify/delete VM networks
 - VM Networks can either be VLANs or untagged networks.
 - VM Networks are associated with Cluster Networks.
 - VM Networks define the type of network (VLAN or untagged) along with the IP address, gateway and, in the case of VLANs, the VLAN ID used by the network.
 - 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.



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 and use storage volumes.
- Manage the amount of storage available in a Harvester cluster.



Topics

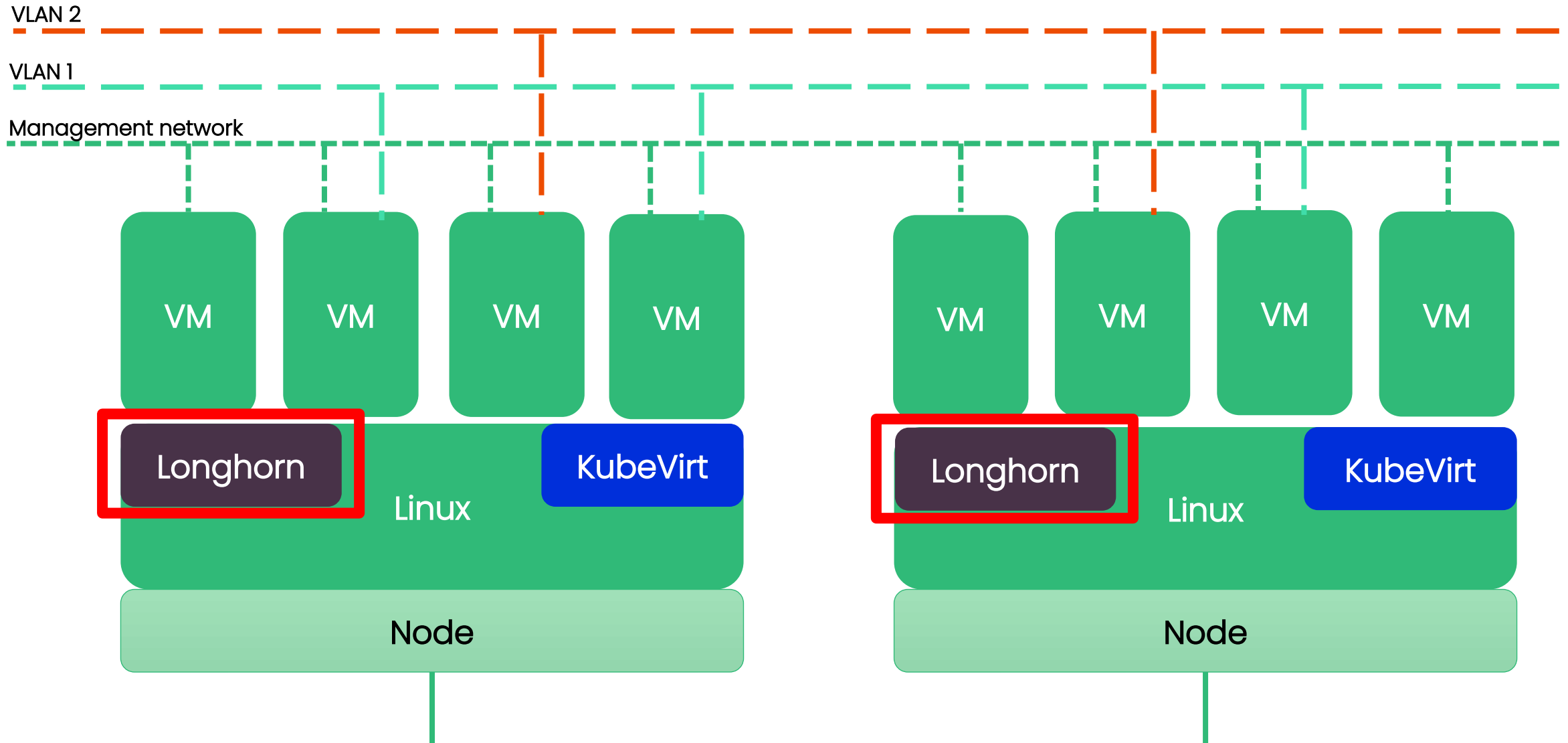
- 1 Longhorn Basics
- 2 Dedicated Storage Network
- 3 Storage Volumes
- 4 Storage Expansion
- 5 Storage Classes



Longhorn Basics



Harvester Components: Longhorn



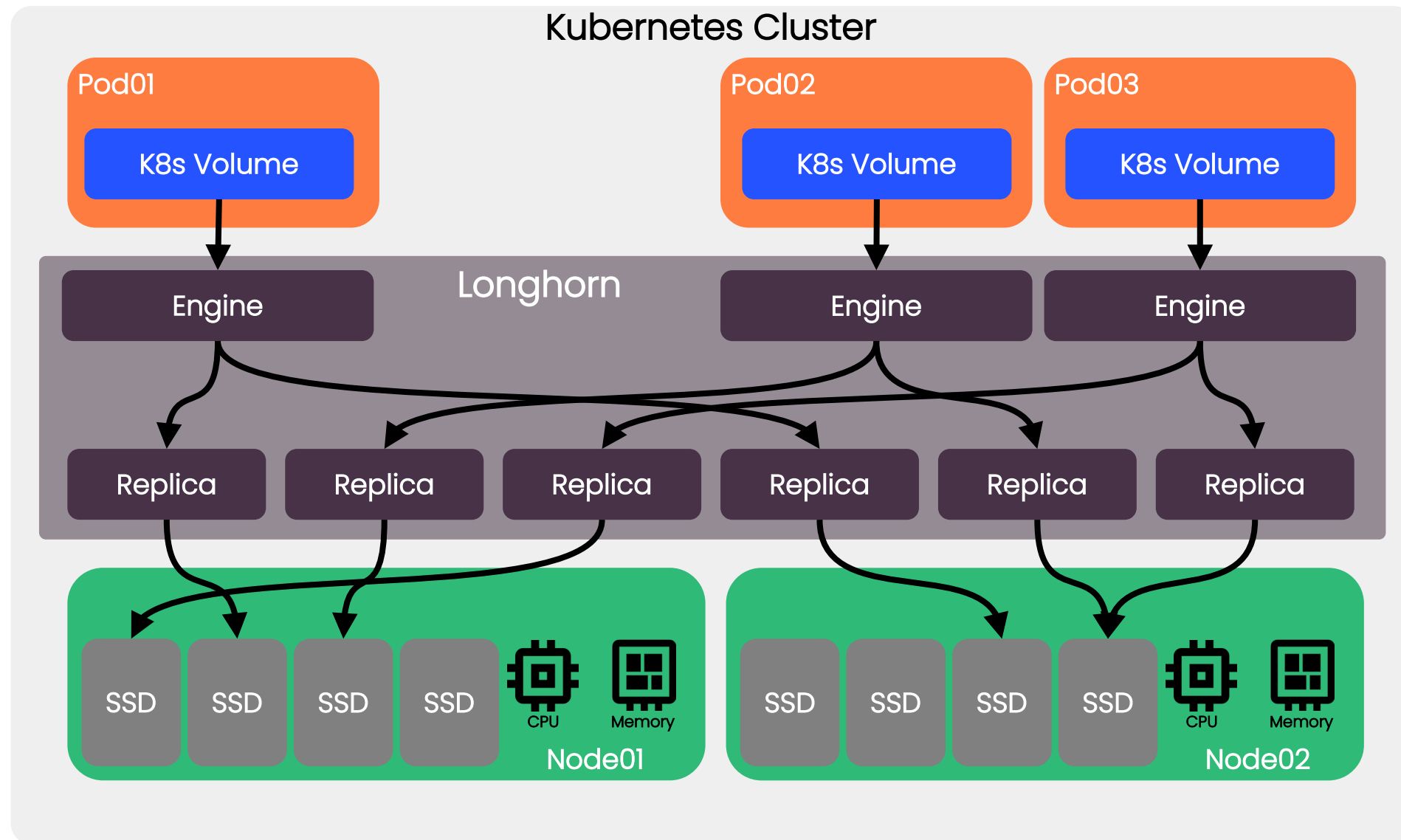
What is Longhorn?

- 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

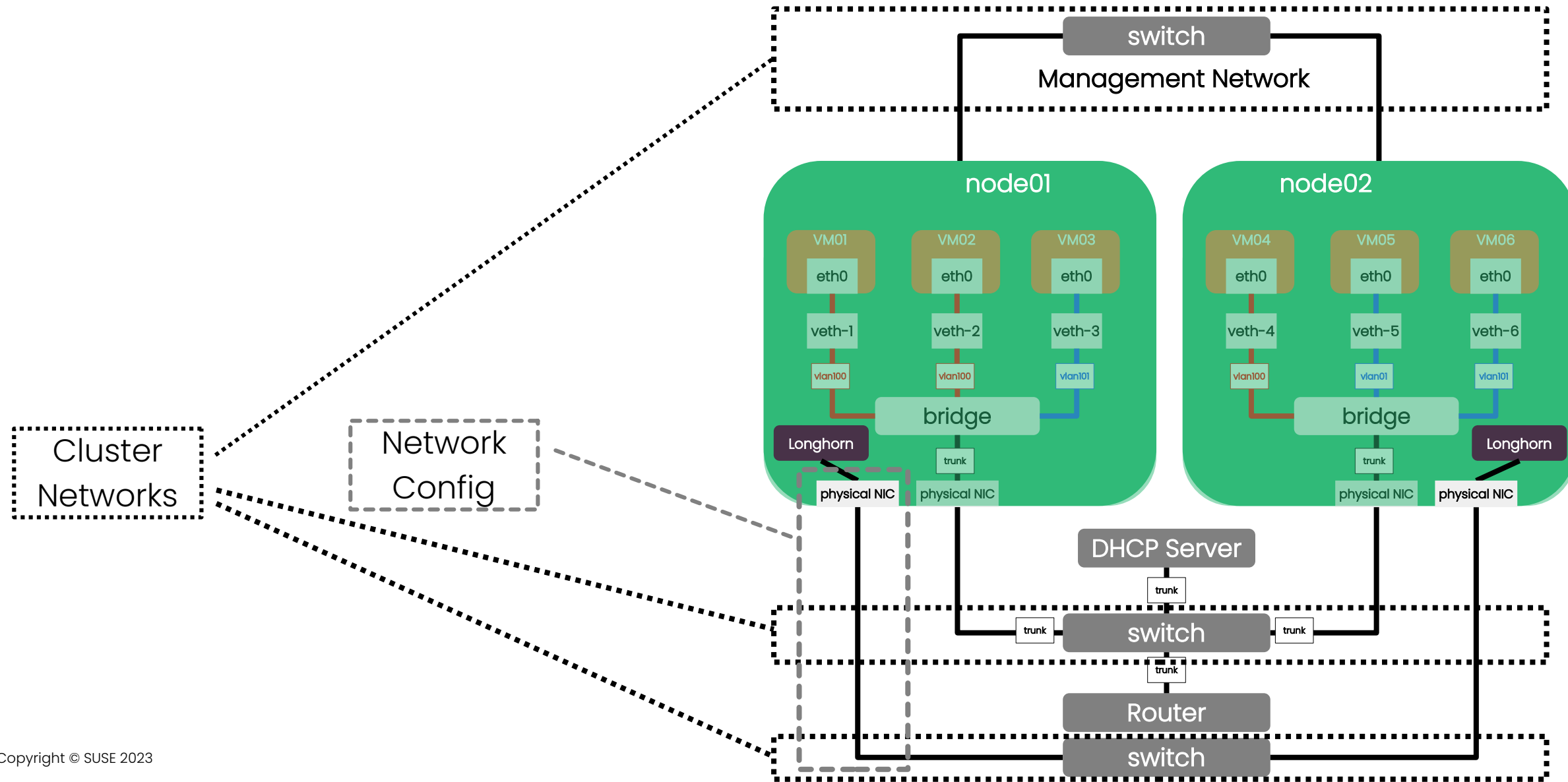
- Persistent **volumes** created by Longhorn are attached to **pods**.
- The Longhorn engine creates **replicas** of the volumes.
- The replicas are spread across the **nodes** and **disks**.



Dedicated Storage Network

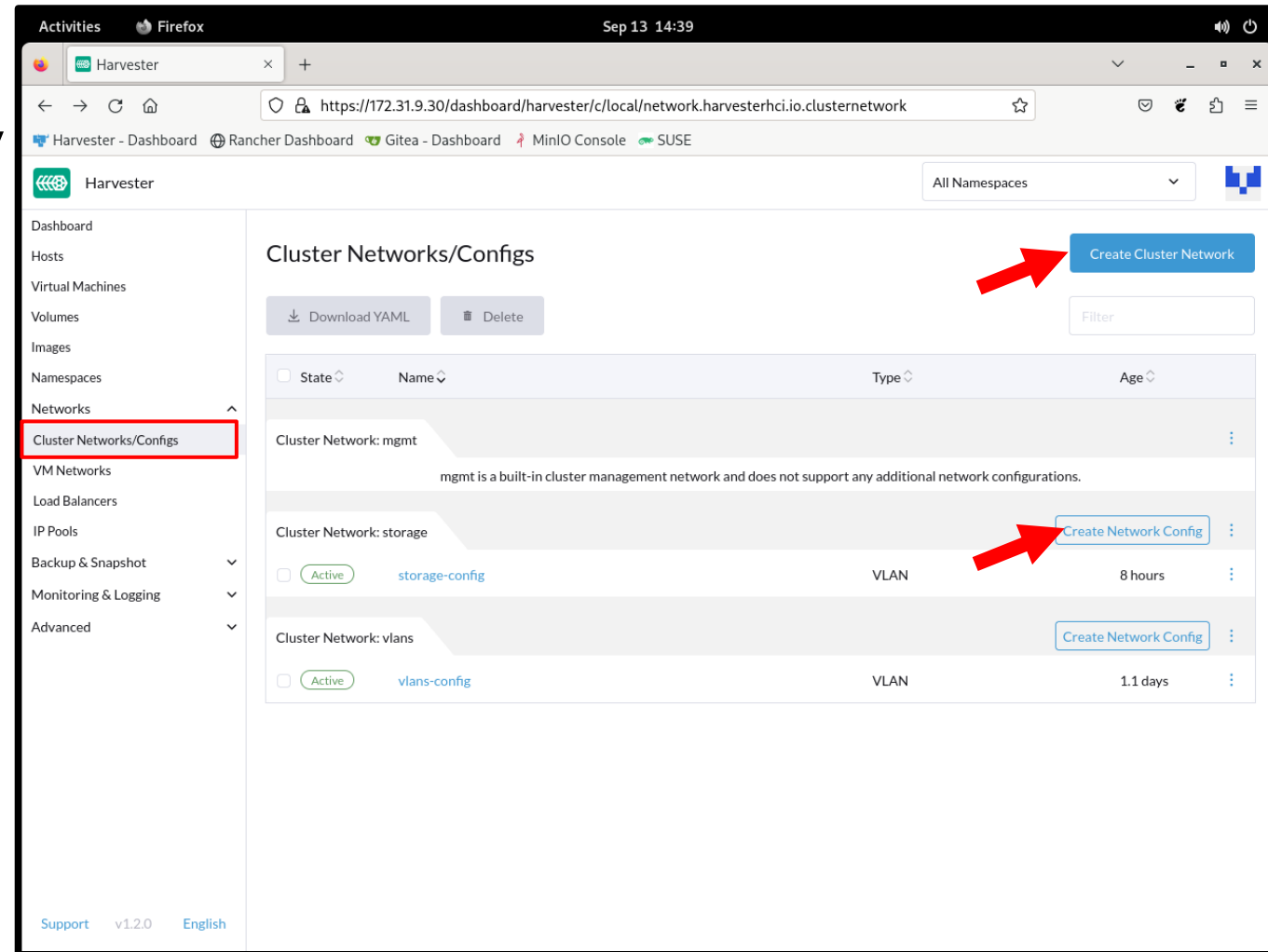


Dedicated Storage Network



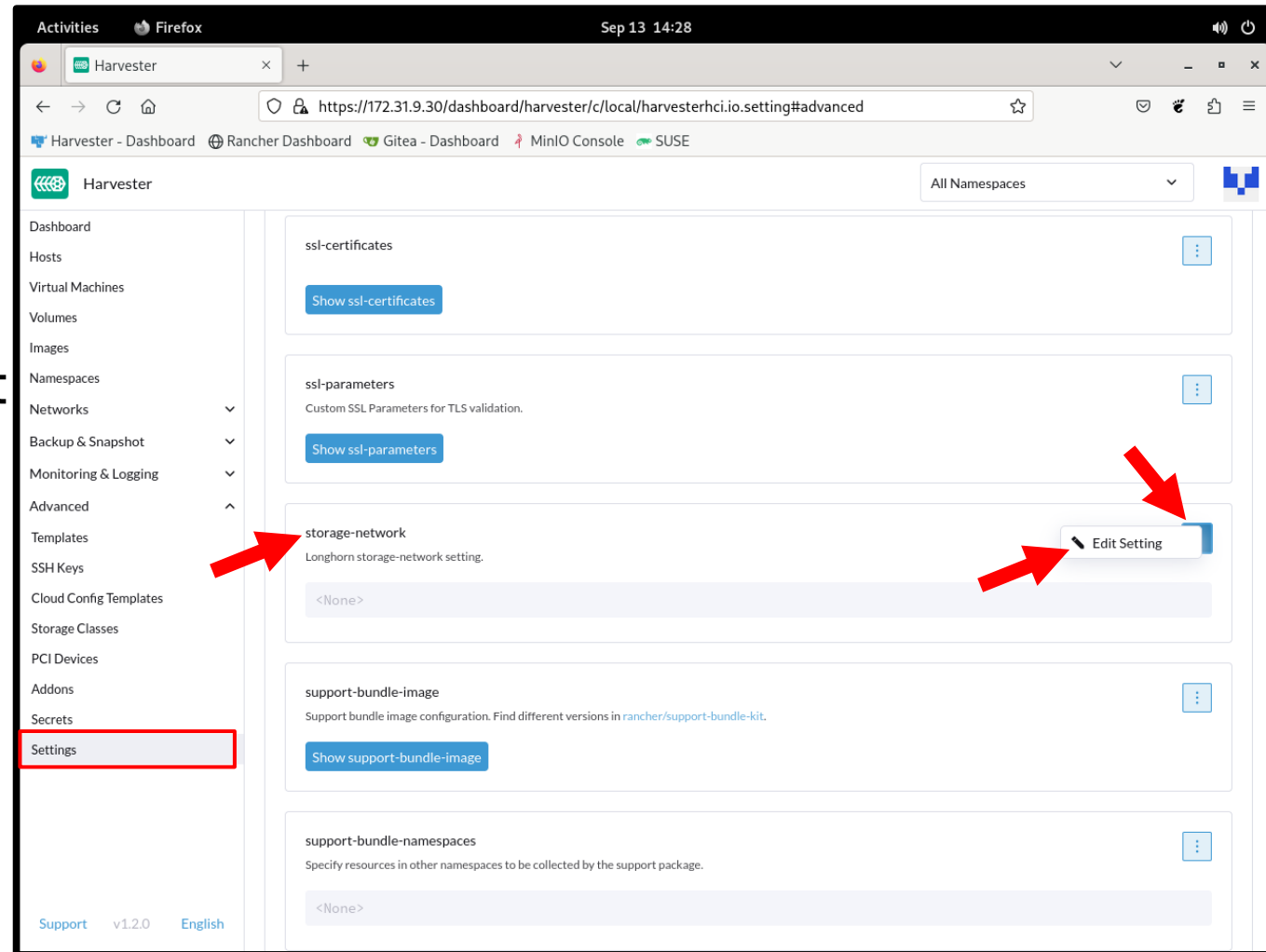
Create Cluster Network for Storage

- 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

Activities Firefox Sep 13 14:36

Harvester

https://172.31.9.30/dashboard/harvester/c/local/harvesterhci.io.setting/storage-network?mode=edit

Harvester - Dashboard Rancher Dashboard Gitea - Dashboard MinIO Console SUSE

Harvester All Namespaces

Dashboard
Hosts
Virtual Machines
Volumes
Images
Namespaces
Networks
Backup & Snapshot
Monitoring & Logging
Advanced
Templates
SSH Keys
Cloud Config Templates
Storage Classes
PCI Devices
Addons
Secrets
Settings

Setting: storage-network **Active**
Age: 1.9 days

Longhorn storage-network setting.

Change Setting: [Use the default value](#)

WARNING:
Any change to storage-network requires shutting down all VMs before applying this setting.
Users have to ensure the Cluster Network is configured and VLAN Config will cover all nodes and ensure the network connectivity is working and expected in all nodes.

☒ Enabled ☐ Disabled

VLAN ID *
99

Cluster Network *
storage

IP Range *
10.0.99.0/24

IP Range should be in IPV4 format.
Number of IPs Required = Number of Nodes * 4 + Number of Disks * 2 + Number of Images to Download/Upload. See doc for more details.

Support v1.2.0 English

Cancel Save



Storage Volumes



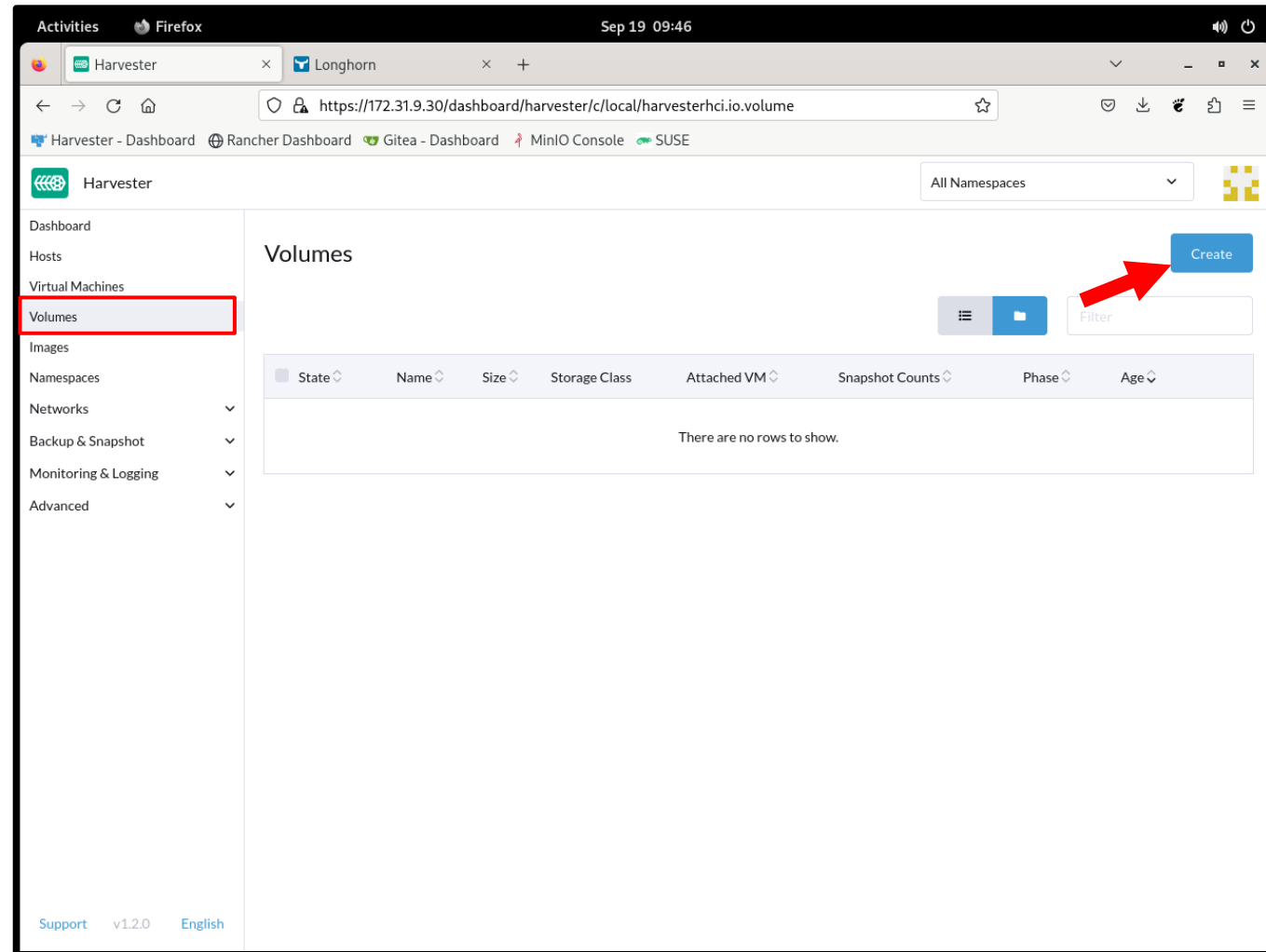
Storage Volumes in Harvester

- 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.



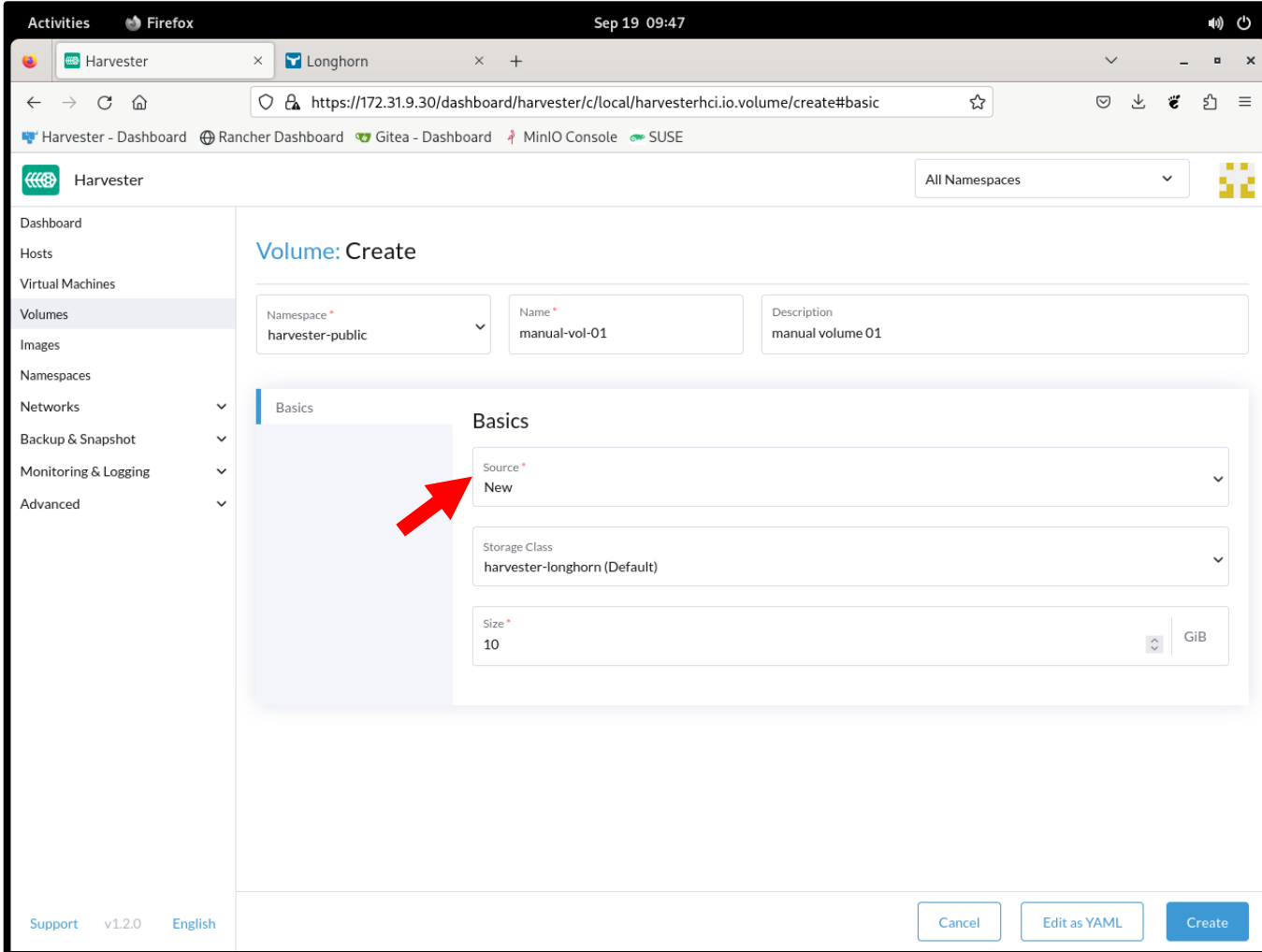
View and Manually Create Volumes

- 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**.



The screenshot shows the Harvester web interface in a Firefox browser window. The page is titled "Volume: Create" and is for the "harvester-public" namespace. The "Name" field is set to "manual-vol-01" and the "Description" is "manual volume 01". In the "Basics" section, the "Source" dropdown menu is open, and a red arrow points to the "New" option. The "Storage Class" is set to "harvester-longhorn (Default)" and the "Size" is 10 GiB. The bottom of the page has "Support", "v1.2.0", and "English" links, and "Cancel", "Edit as YAML", and "Create" buttons.

Activities Firefox Sep 19 09:47

Harvester Longhorn

https://172.31.9.30/dashboard/harvester/c/local/harvesterhci.io.volume/create#basic

Harvester - Dashboard Rancher Dashboard Gitea - Dashboard MinIO Console SUSE

Harvester All Namespaces

Dashboard
Hosts
Virtual Machines
Volumes
Images
Namespaces
Networks
Backup & Snapshot
Monitoring & Logging
Advanced

Volume: Create

Namespace * harvester-public Name * manual-vol-01 Description manual volume 01

Basics

Source * New

Storage Class harvester-longhorn (Default)

Size * 10 GiB

Support v1.2.0 English

Cancel Edit as YAML Create



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**.

The screenshot shows the Harvester web interface in a Firefox browser. The URL is `https://172.31.9.30/dashboard/harvester/c/local/harvesterhci.io.volume/create#basic`. The left sidebar shows the navigation menu with 'Volumes' selected. The main content area is titled 'Volume: Create' and contains the following fields:

- Namespace:** harvester-public
- Name:** leap-15.5
- Description:** Any text you want that better describes this resource
- Basics tab:**
 - Source:** VM Image
 - Image:** harvester-public/leap-15.5-nocloud
 - Size:** 10 GiB

Red arrows point to the 'Source' and 'Image' fields. At the bottom right, there are three buttons: 'Cancel', 'Edit as YAML', and 'Create'.



Storage Expansion



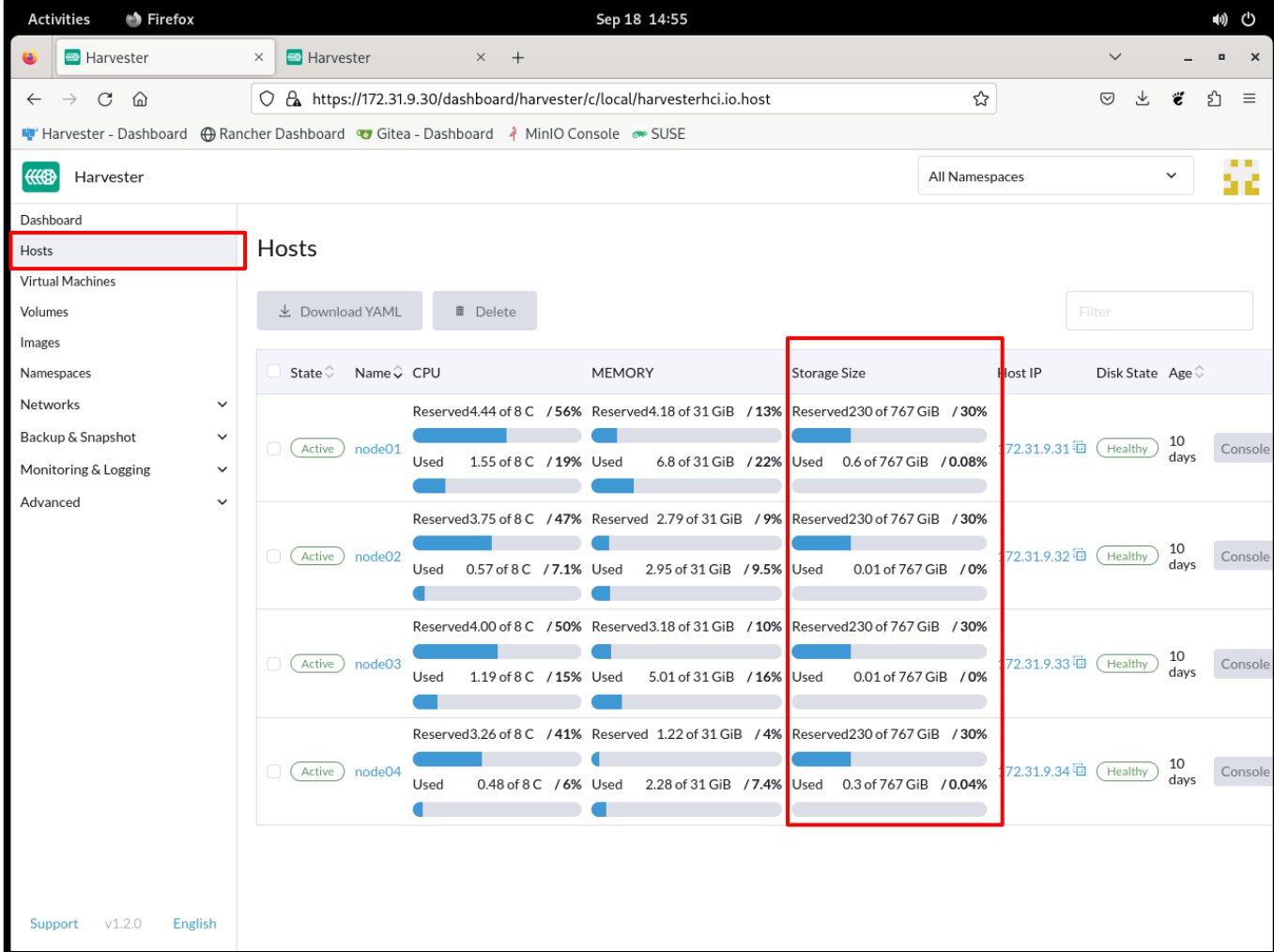
Add Additional Storage

- 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.



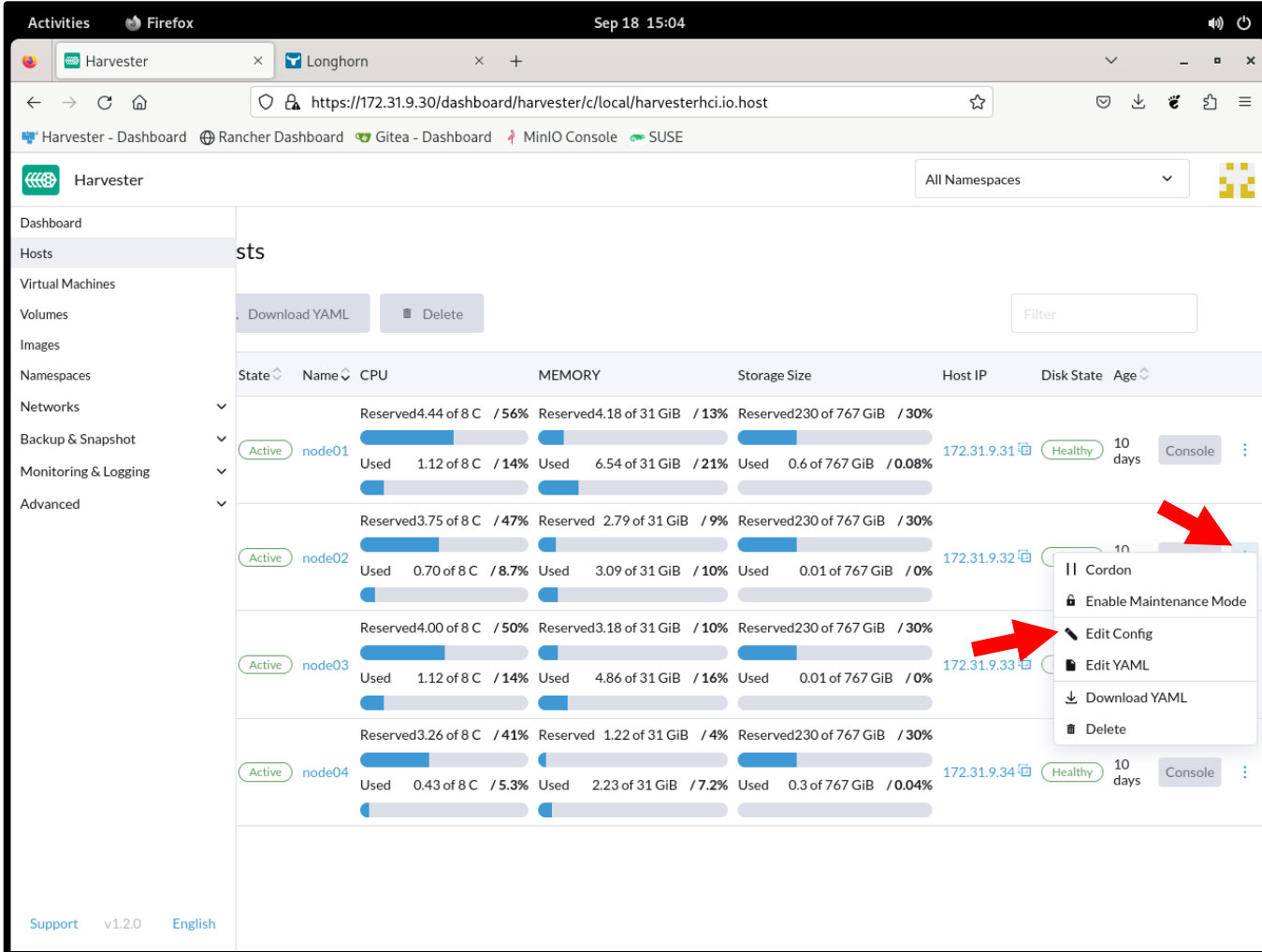
The screenshot shows the Harvester web interface in a Firefox browser. The 'Hosts' tab is selected in the left sidebar. The main content area displays a table of host information for four nodes (node01 to node04). The 'Storage Size' column is highlighted with a red box, showing reserved and used storage for each node. The table also includes columns for State, Name, CPU, MEMORY, Host IP, Disk State, and Age.

State	Name	CPU	MEMORY	Storage Size	Host IP	Disk State	Age
Active	node01	Reserved 4.44 of 8 C / 56% Used 1.55 of 8 C / 19%	Reserved 4.18 of 31 GiB / 13% Used 6.8 of 31 GiB / 22%	Reserved 230 of 767 GiB / 30% Used 0.6 of 767 GiB / 0.08%	72.31.9.31	Healthy	10 days
Active	node02	Reserved 3.75 of 8 C / 47% Used 0.57 of 8 C / 7.1%	Reserved 2.79 of 31 GiB / 9% Used 2.95 of 31 GiB / 9.5%	Reserved 230 of 767 GiB / 30% Used 0.01 of 767 GiB / 0%	72.31.9.32	Healthy	10 days
Active	node03	Reserved 4.00 of 8 C / 50% Used 1.19 of 8 C / 15%	Reserved 3.18 of 31 GiB / 10% Used 5.01 of 31 GiB / 16%	Reserved 230 of 767 GiB / 30% Used 0.01 of 767 GiB / 0%	72.31.9.33	Healthy	10 days
Active	node04	Reserved 3.26 of 8 C / 41% Used 0.48 of 8 C / 6%	Reserved 1.22 of 31 GiB / 4% Used 2.28 of 31 GiB / 7.4%	Reserved 230 of 767 GiB / 30% Used 0.3 of 767 GiB / 0.04%	72.31.9.34	Healthy	10 days



Hosts: Edit Host Config

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



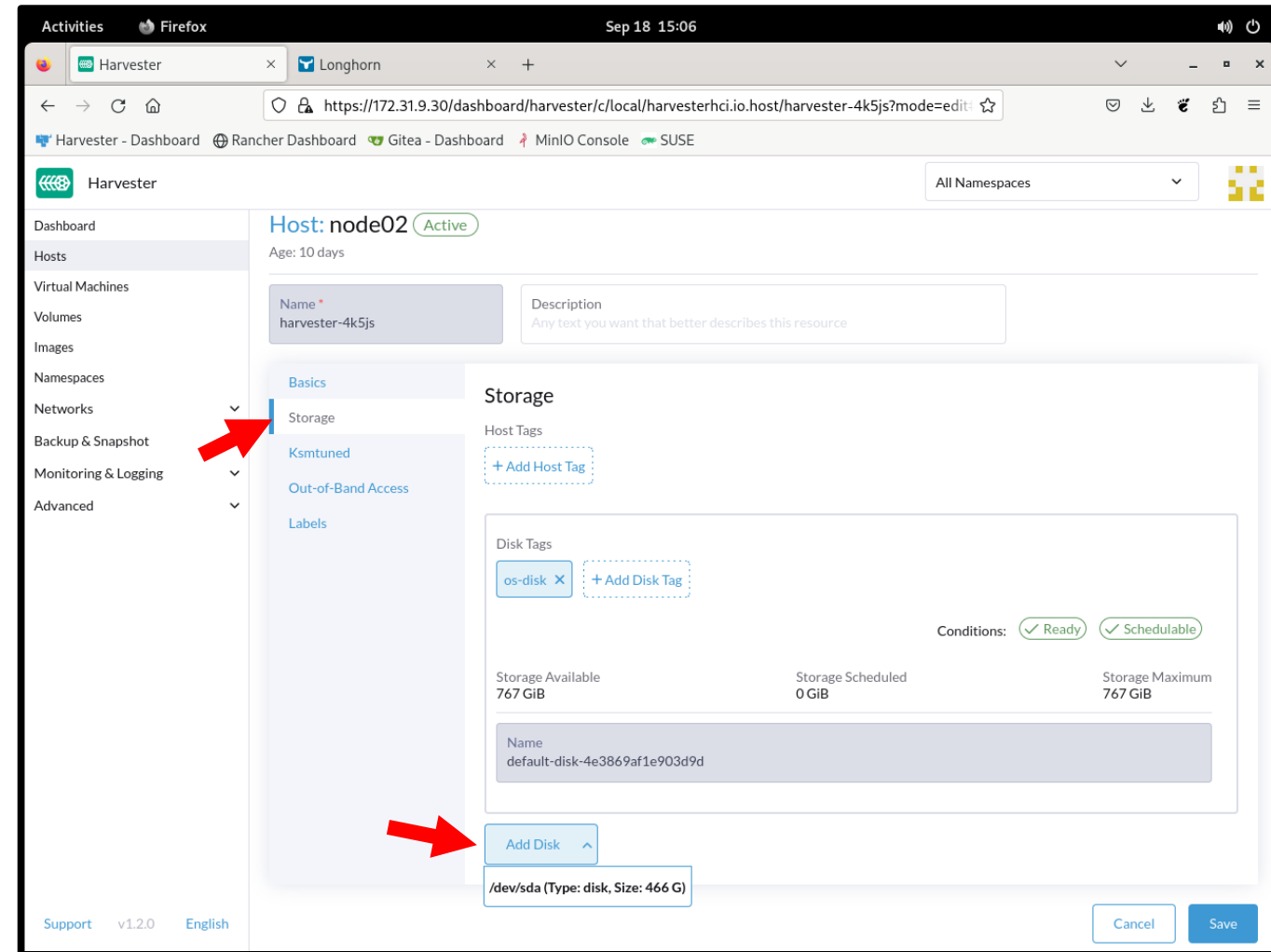
The screenshot shows the Harvester dashboard in a Firefox browser window. The left sidebar contains navigation links: Dashboard, Hosts, Virtual Machines, Volumes, Images, Namespaces, Networks, Backup & Snapshot, Monitoring & Logging, and Advanced. The main content area displays a table of hosts with columns for State, Name, CPU, MEMORY, Storage Size, Host IP, Disk State, and Age. Four hosts are listed: node01, node02, node03, and node04, all in an 'Active' state. A red arrow points to the three-dot menu next to node03, and another red arrow points to the 'Edit Config' option in the dropdown menu that appears.

State	Name	CPU	MEMORY	Storage Size	Host IP	Disk State	Age
Active	node01	Reserved 4.44 of 8 C / 56% Used 1.12 of 8 C / 14%	Reserved 4.18 of 31 GiB / 13% Used 6.54 of 31 GiB / 21%	Reserved 230 of 767 GiB / 30% Used 0.6 of 767 GiB / 0.08%	172.31.9.31	Healthy	10 days
Active	node02	Reserved 3.75 of 8 C / 47% Used 0.70 of 8 C / 8.7%	Reserved 2.79 of 31 GiB / 9% Used 3.09 of 31 GiB / 10%	Reserved 230 of 767 GiB / 30% Used 0.01 of 767 GiB / 0%	172.31.9.32	Healthy	10 days
Active	node03	Reserved 4.00 of 8 C / 50% Used 1.12 of 8 C / 14%	Reserved 3.18 of 31 GiB / 10% Used 4.86 of 31 GiB / 16%	Reserved 230 of 767 GiB / 30% Used 0.01 of 767 GiB / 0%	172.31.9.33	Healthy	10 days
Active	node04	Reserved 3.26 of 8 C / 41% Used 0.43 of 8 C / 5.3%	Reserved 1.22 of 31 GiB / 4% Used 2.23 of 31 GiB / 7.2%	Reserved 230 of 767 GiB / 30% Used 0.3 of 767 GiB / 0.04%	172.31.9.34	Healthy	10 days



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



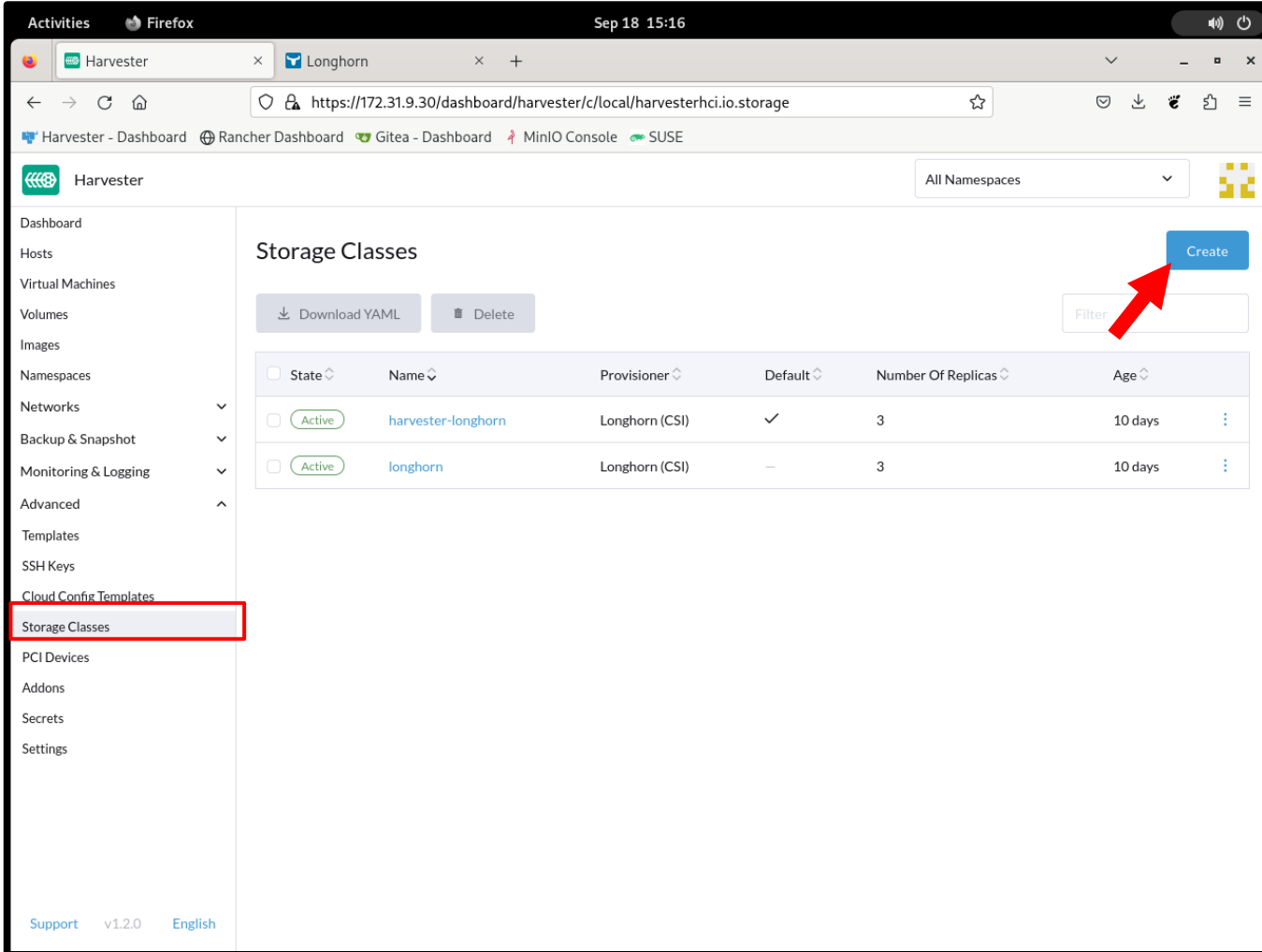
Storage Classes in Harvester

- 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**



The screenshot shows the Harvester web interface in a Firefox browser. The left sidebar contains a navigation menu with the following items: Dashboard, Hosts, Virtual Machines, Volumes, Images, Namespaces, Networks, Backup & Snapshot, Monitoring & Logging, Advanced, Templates, SSH Keys, Cloud Config Templates, Storage Classes (highlighted with a red box), PCI Devices, Addons, Secrets, and Settings. The main content area is titled 'Storage Classes' and includes a 'Create' button (highlighted with a red arrow) and a 'Filter' input field. Below these are two buttons: 'Download YAML' and 'Delete'. A table lists the existing storage classes:

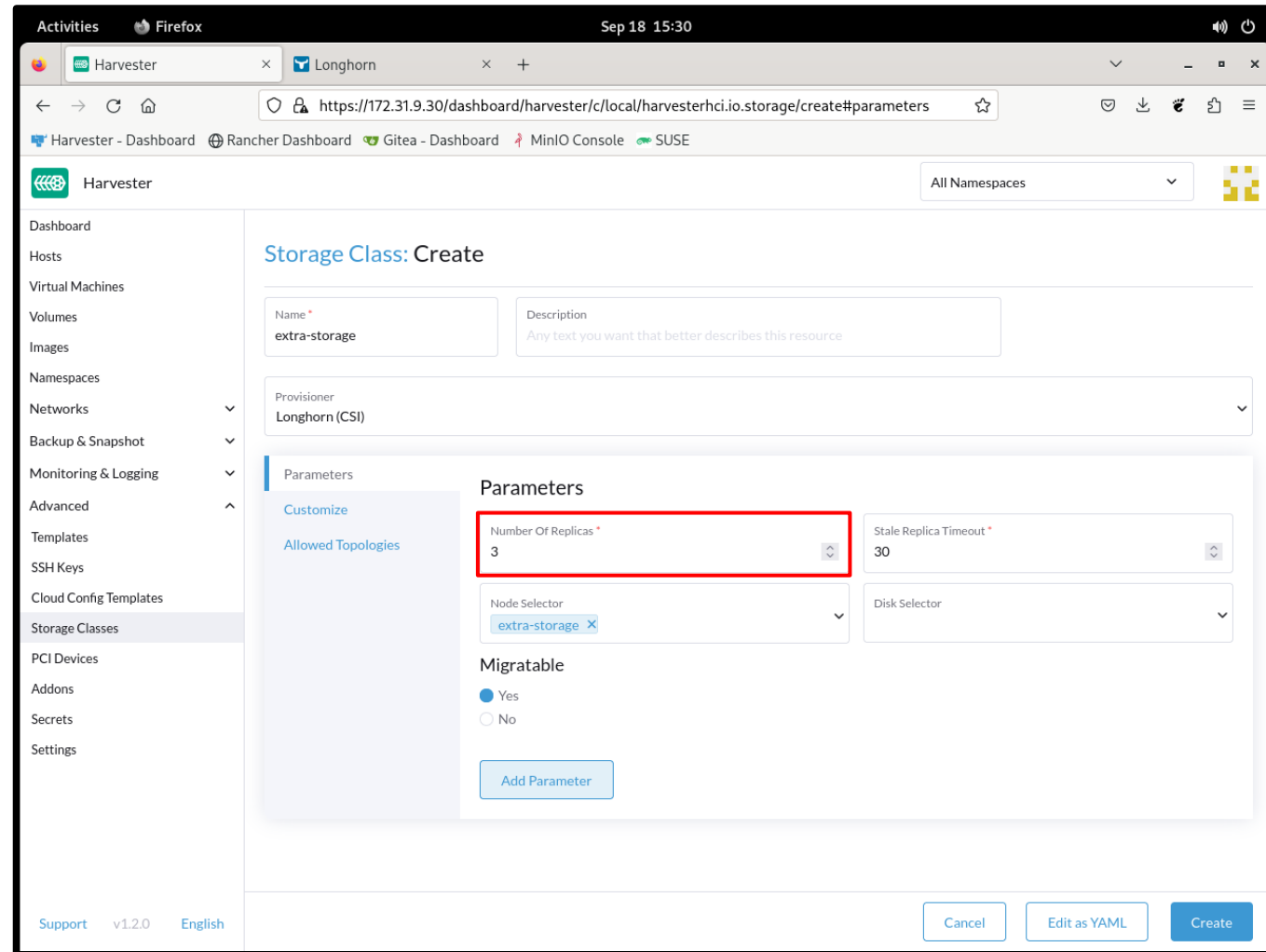
<input type="checkbox"/>	State	Name	Provisioner	Default	Number Of Replicas	Age	
<input type="checkbox"/>	Active	harvester-longhorn	Longhorn (CSI)	✓	3	10 days	⋮
<input type="checkbox"/>	Active	longhorn	Longhorn (CSI)	—	3	10 days	⋮

At the bottom of the interface, there are links for 'Support', 'v1.2.0', and 'English'.



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**.

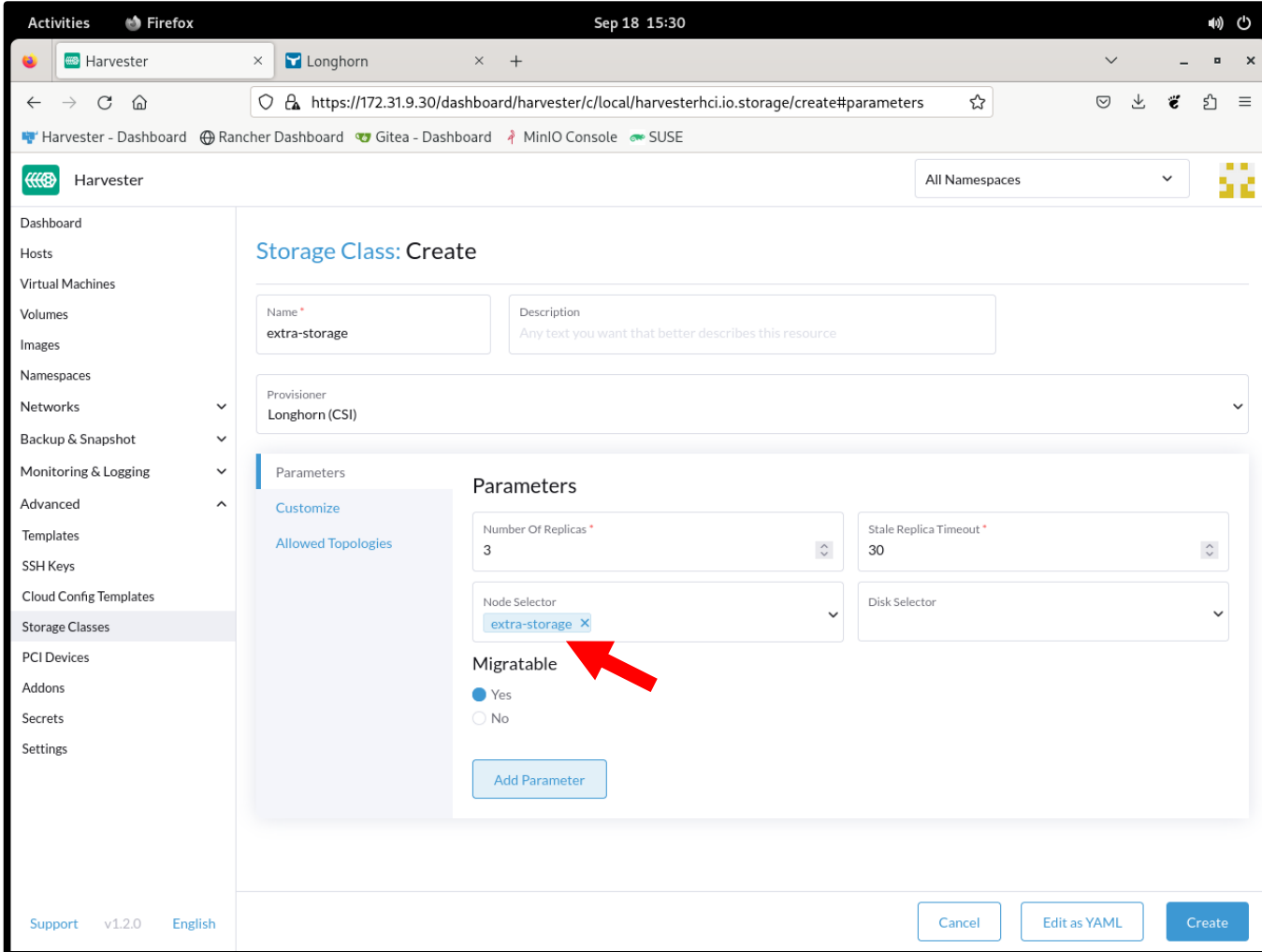


The screenshot shows the Harvester web interface in a Firefox browser window. The page is titled "Storage Class: Create". On the left is a sidebar menu with options like Dashboard, Hosts, Virtual Machines, Volumes, Images, Namespaces, Networks, Backup & Snapshot, Monitoring & Logging, Advanced, Templates, SSH Keys, Cloud Config Templates, Storage Classes (selected), PCI Devices, Addons, Secrets, and Settings. The main content area has a form for creating a storage class. The "Name" field is filled with "extra-storage". The "Provisioner" is set to "Longhorn (CSI)". Under the "Parameters" section, the "Number Of Replicas" is set to 3 (highlighted with a red box), "Stale Replica Timeout" is 30, "Node Selector" is "extra-storage", and "Disk Selector" is empty. The "Migratable" option is set to "Yes". At the bottom right are buttons for "Cancel", "Edit as YAML", and "Create".



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.



The screenshot shows the Harvester web interface in a Firefox browser window. The address bar indicates the URL: `https://172.31.9.30/dashboard/harvester/c/local/harvesterhci.io.storage/create#parameters`. The left sidebar contains a navigation menu with options like Dashboard, Hosts, Virtual Machines, Volumes, Images, Namespaces, Networks, Backup & Snapshot, Monitoring & Logging, Advanced, Templates, SSH Keys, Cloud Config Templates, Storage Classes, PCI Devices, Addons, Secrets, and Settings. The 'Storage Classes' option is selected. The main content area is titled 'Storage Class: Create' and contains the following fields:

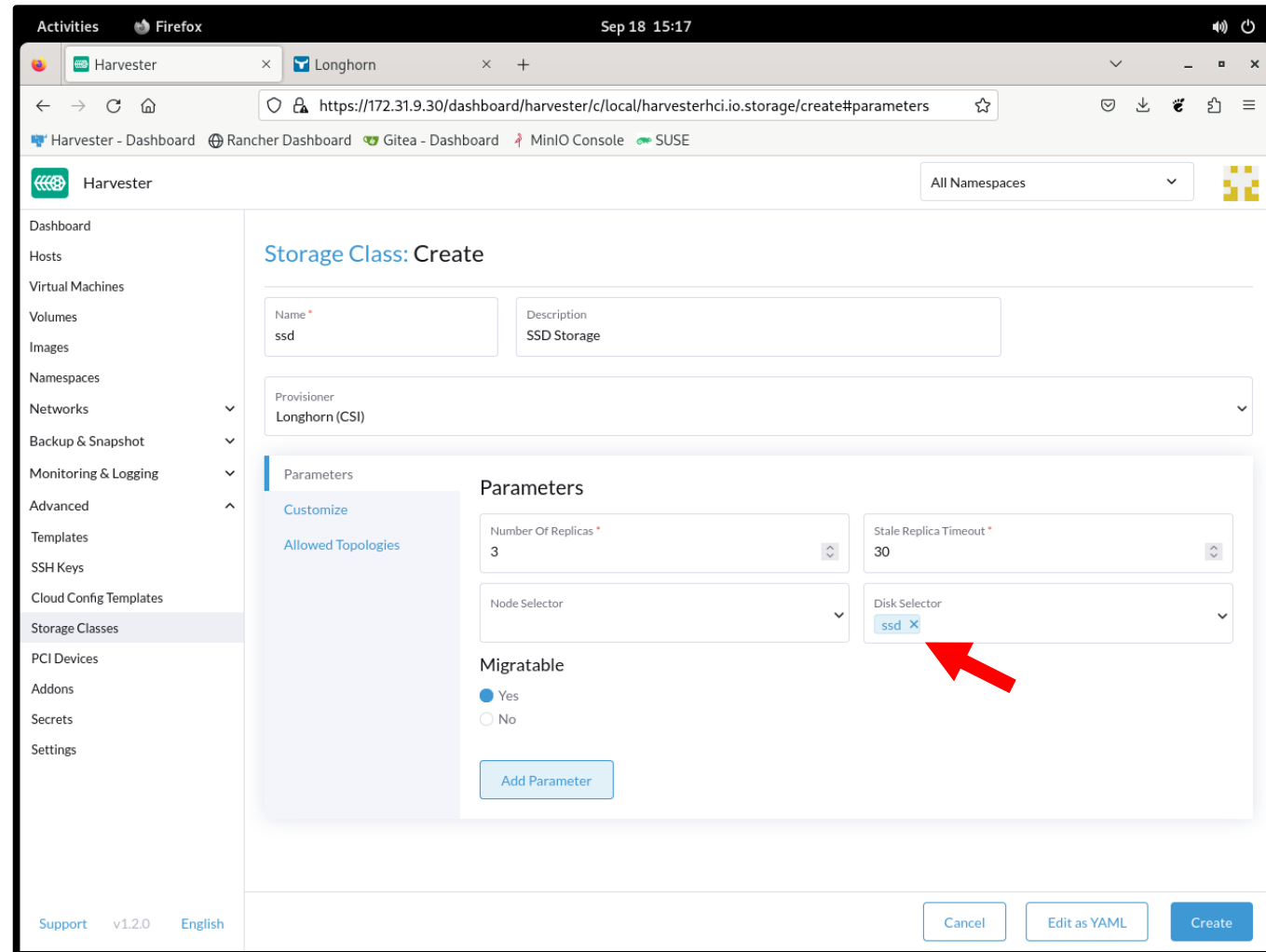
- Name:** extra-storage
- Description:** Any text you want that better describes this resource
- Provisioner:** Longhorn (CSI)
- Parameters:**
 - Number Of Replicas:** 3
 - Stale Replica Timeout:** 30
 - Node Selector:** extra-storage (indicated by a red arrow)
 - Disk Selector:**
 - Migratable:** Yes (selected)
 - Add Parameter:** Button

At the bottom of the form, there are three buttons: 'Cancel', 'Edit as YAML', and 'Create'.



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.



The screenshot shows the Harvester web interface in a Firefox browser window. The page is titled "Storage Class: Create". On the left is a sidebar menu with options like Dashboard, Hosts, Virtual Machines, Volumes, Images, Namespaces, Networks, Backup & Snapshot, Monitoring & Logging, Advanced, Templates, SSH Keys, Cloud Config Templates, Storage Classes (highlighted), PCI Devices, Addons, Secrets, and Settings. The main content area has a form for creating a storage class. The "Name" field is filled with "ssd" and the "Description" is "SSD Storage". The "Provisioner" is set to "Longhorn (CSI)". Under the "Parameters" section, "Number Of Replicas" is 3, "Stale Replica Timeout" is 30, "Node Selector" is empty, and "Disk Selector" is set to "ssd". A red arrow points to the "ssd" value in the "Disk Selector" field. Below the parameters, the "Migratable" option is set to "Yes". At the bottom right are "Cancel", "Edit as YAML", and "Create" buttons.

Activities Firefox Sep 18 15:17

Harvester Longhorn

https://172.31.9.30/dashboard/harvester/c/local/harvesterhci.io.storage/create#parameters

Harvester - Dashboard Rancher Dashboard Gitea - Dashboard MinIO Console SUSE

Harvester All Namespaces

Dashboard

Hosts

Virtual Machines

Volumes

Images

Namespaces

Networks

Backup & Snapshot

Monitoring & Logging

Advanced

Templates

SSH Keys

Cloud Config Templates

Storage Classes

PCI Devices

Addons

Secrets

Settings

Storage Class: Create

Name *
ssd

Description
SSD Storage

Provisioner
Longhorn (CSI)

Parameters

Customize

Allowed Topologies

Number Of Replicas *
3

Stale Replica Timeout *
30

Node Selector

Disk Selector
ssd

Migratable
☒ Yes
☐ No

Add Parameter

Support v1.2.0 English

Cancel Edit as YAML Create



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

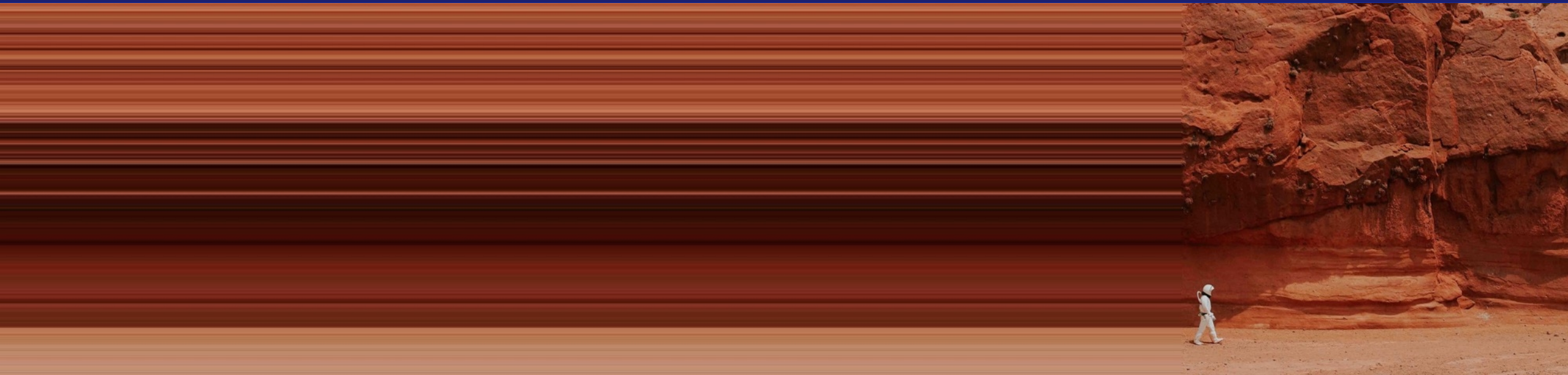


- Manage the amount of storage available in a Harvester cluster
 - Additional storage can be added to the embedded Longhorn cluster by adding disks/partitions/logical volumes attached to the cluster nodes.



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

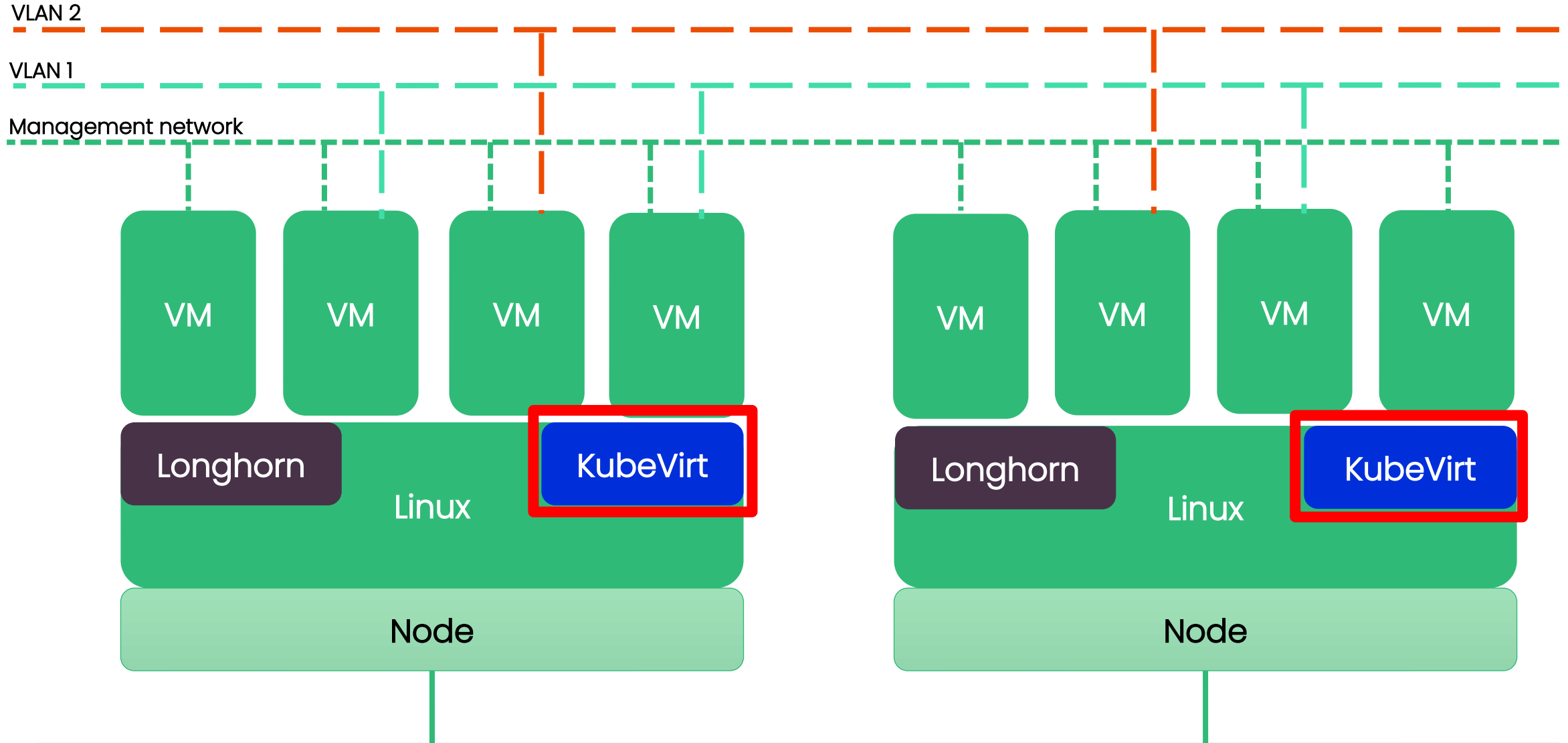
- 1 Virtual Machine Management Concepts in Harvester
- 2 Virtual Machine Images
- 3 SSH Keys
- 4 Virtual Machine Creation
- 5 VM Cloning
- 6 Cloud Config Templates
- 7 VM Templates
- 8 Live Migration
- 9 Hot-plug Volumes
- 10 Virtual Machine Snapshots
- 11 Virtual Machine Backups



Virtual Machine Management Concepts in Harvester



Virtualization in Kubernetes - KubeVirt



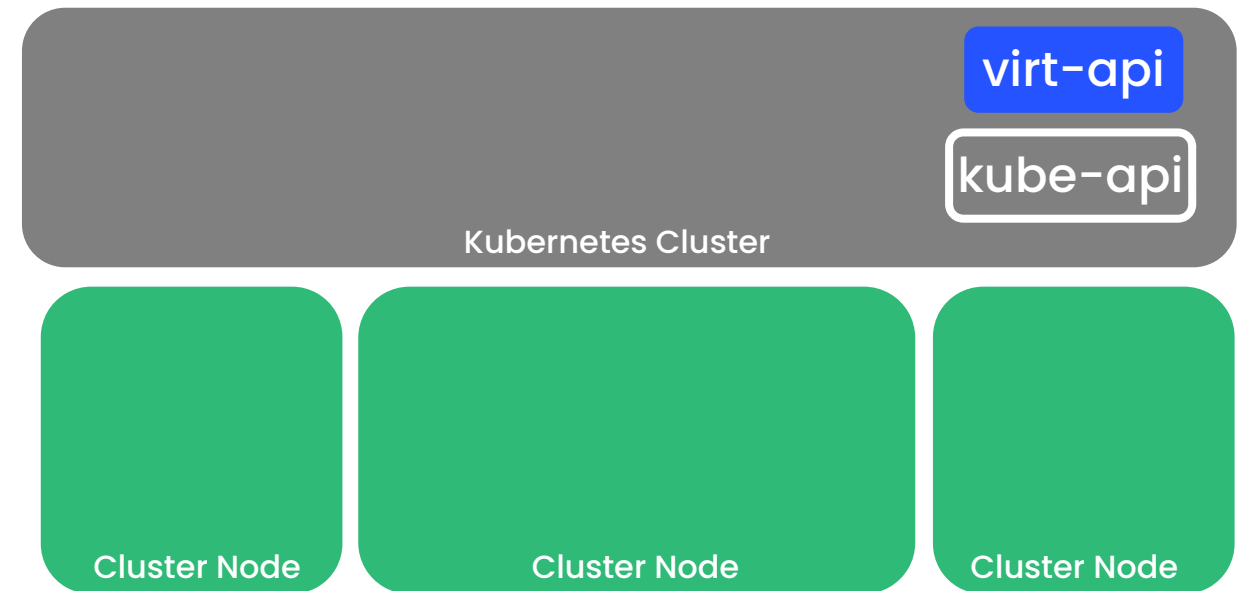
KubeVirt in Harvester

- 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)



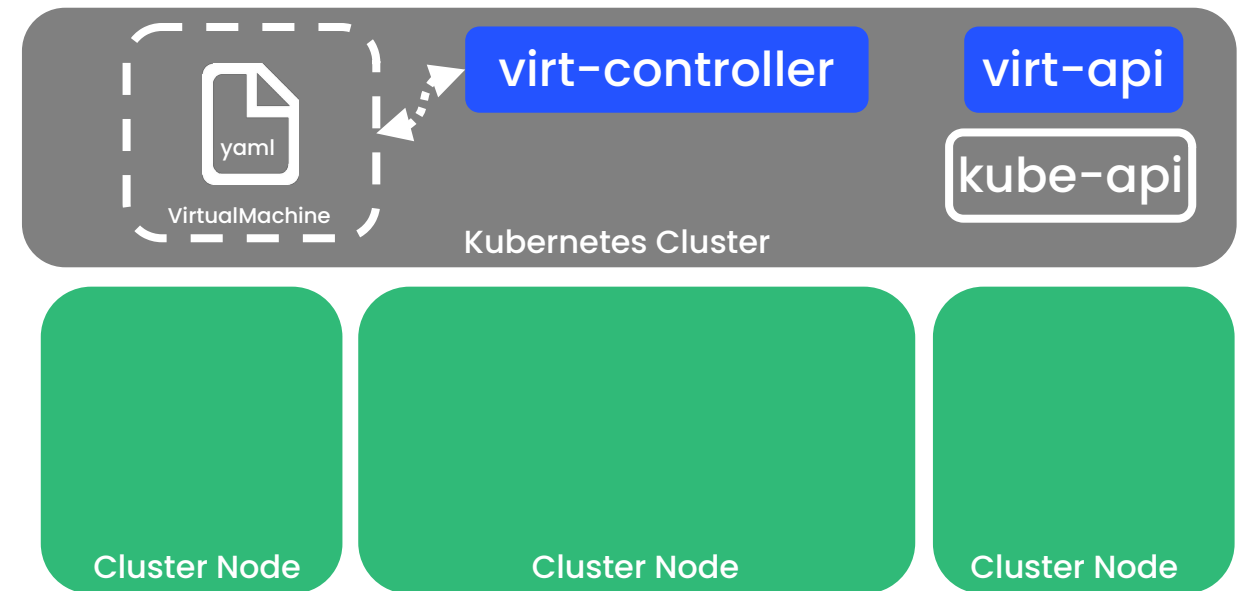
KubeVirt: virt-api

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



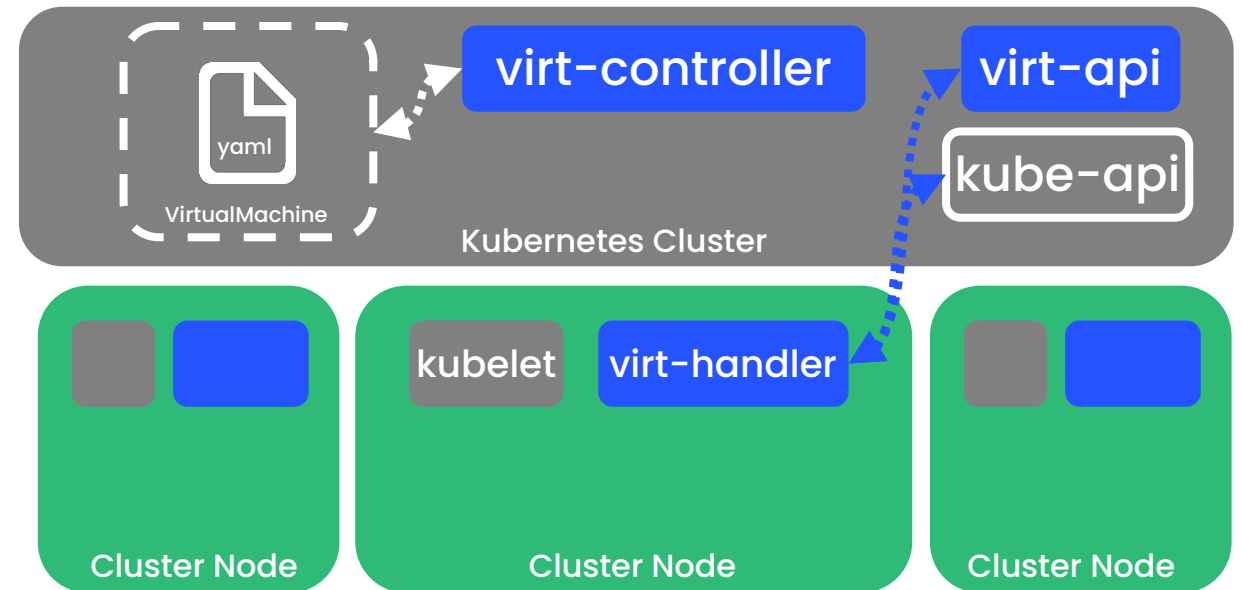
KubeVirt: virt-controller

- 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.



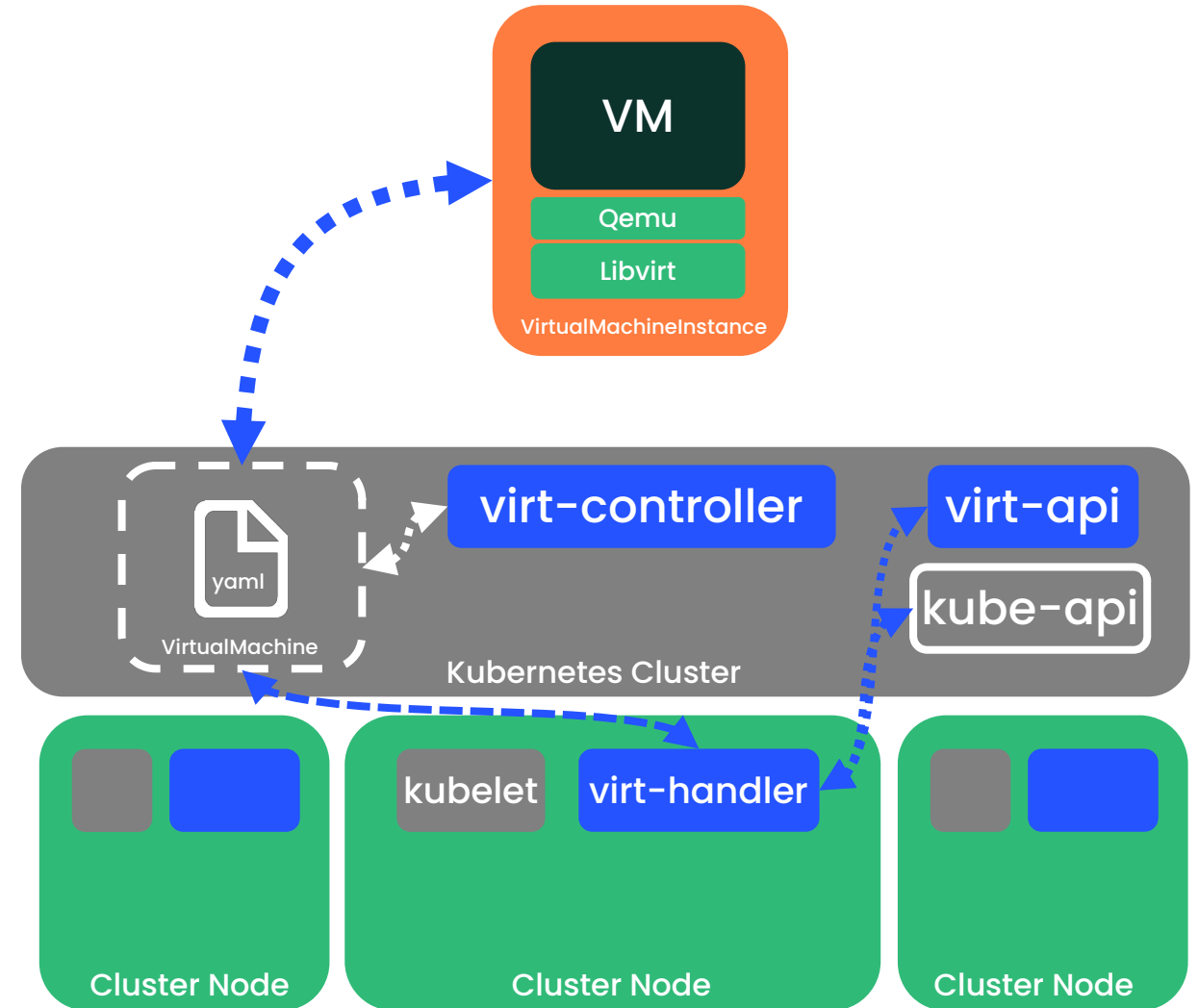
KubeVirt: virt-handler

- 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.).



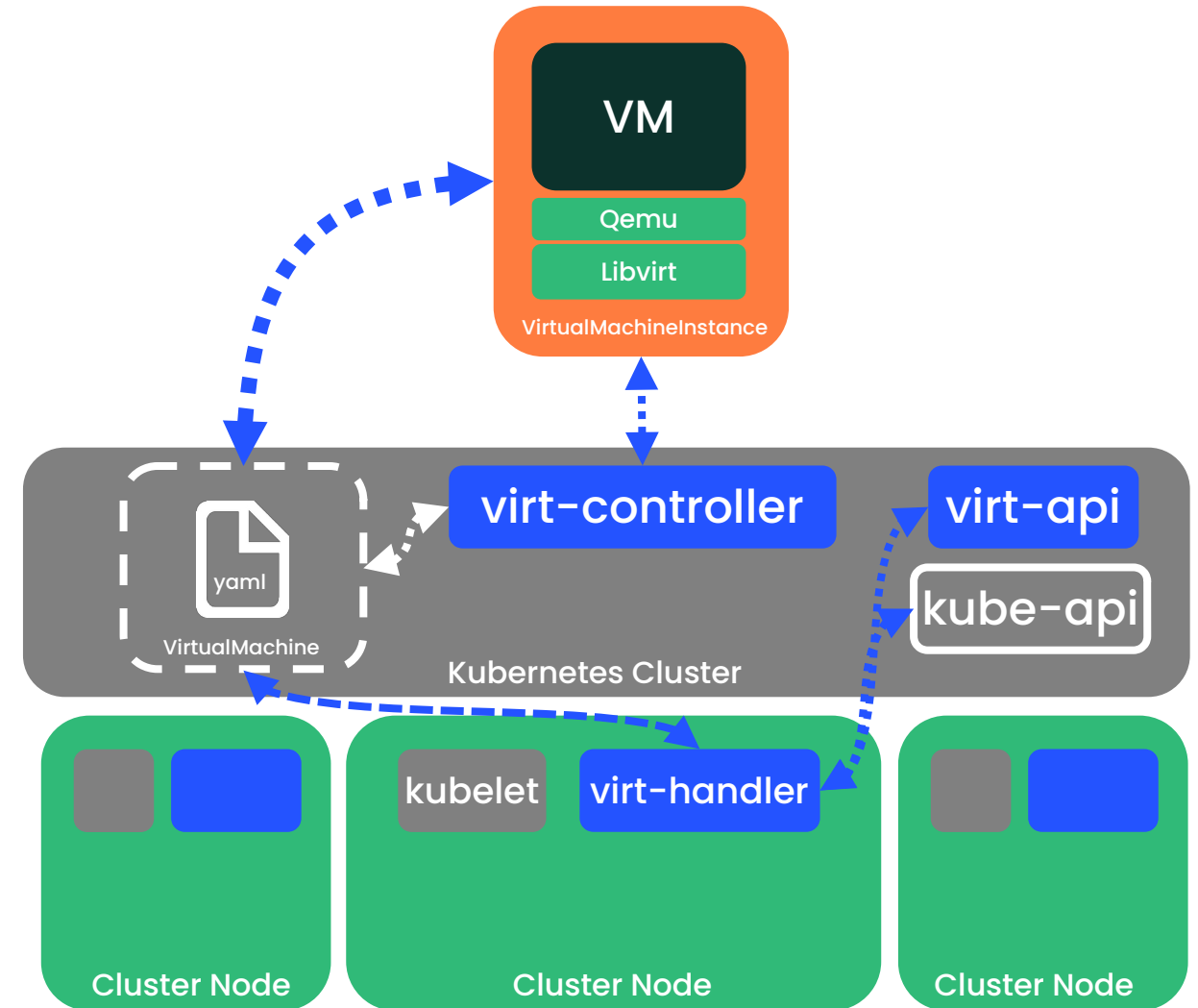
KubeVirt: VM Instantiation

- 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



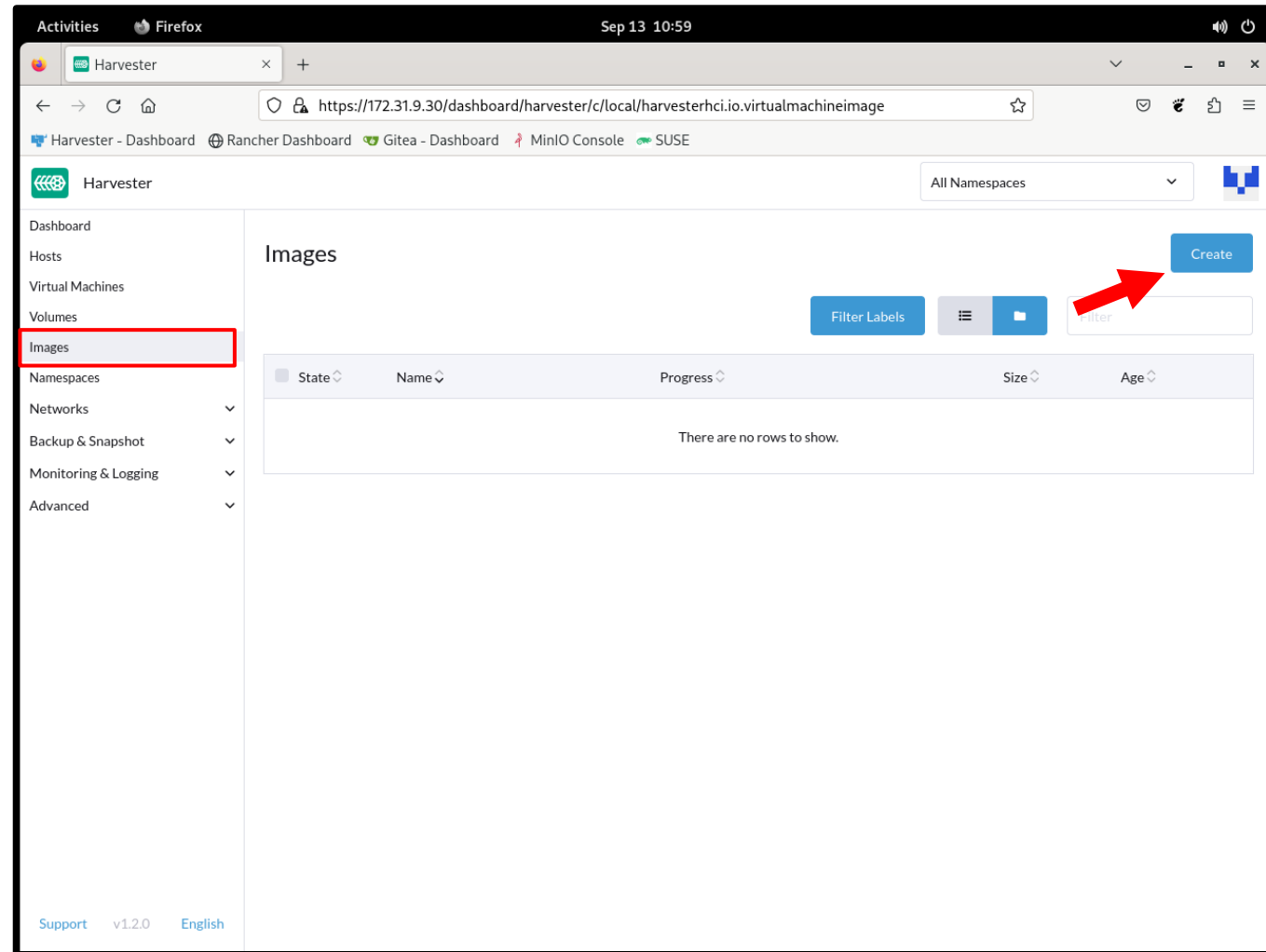
Virtual Machine Image Requirements

- 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.



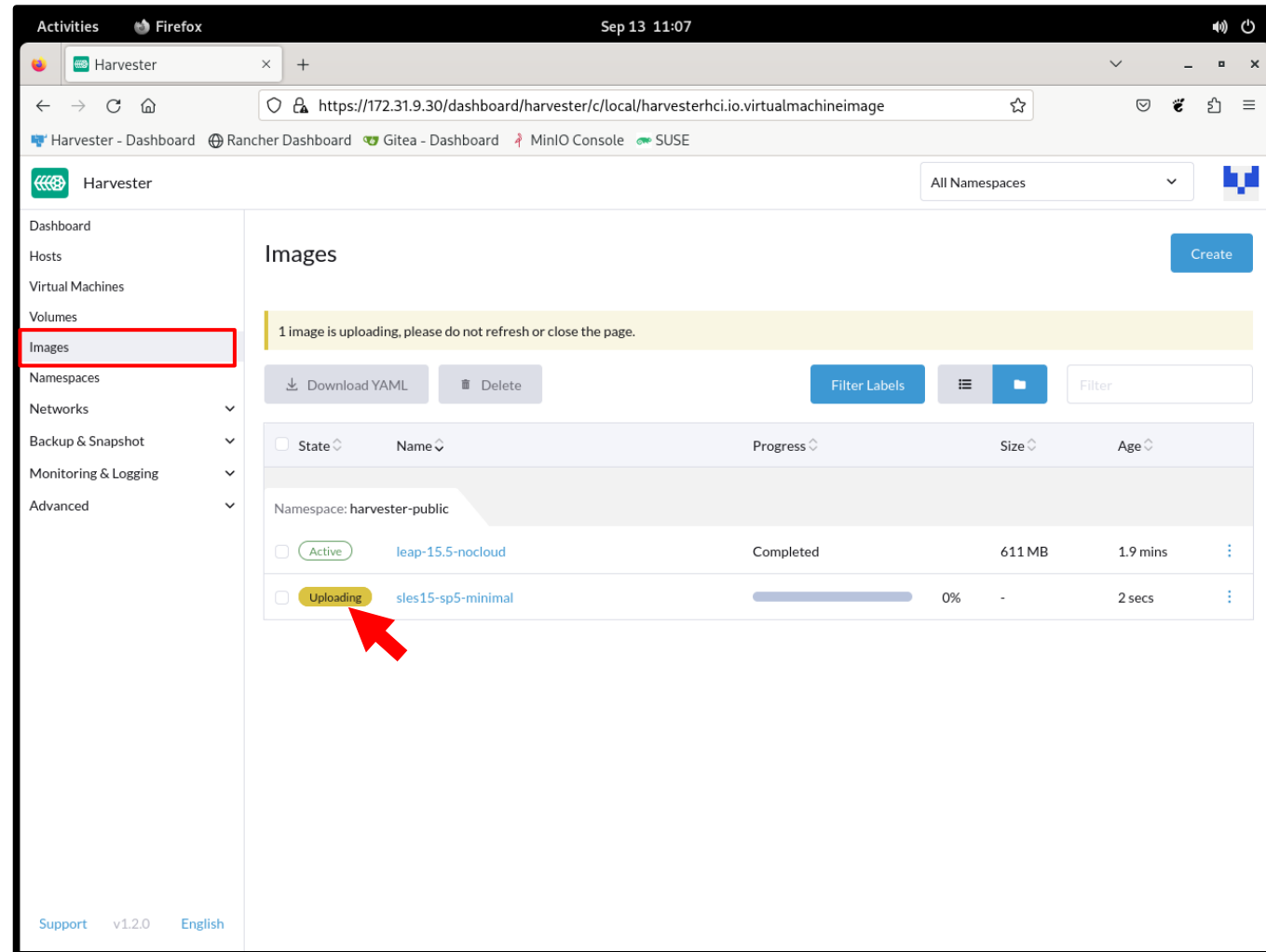
View/Create Images

- 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



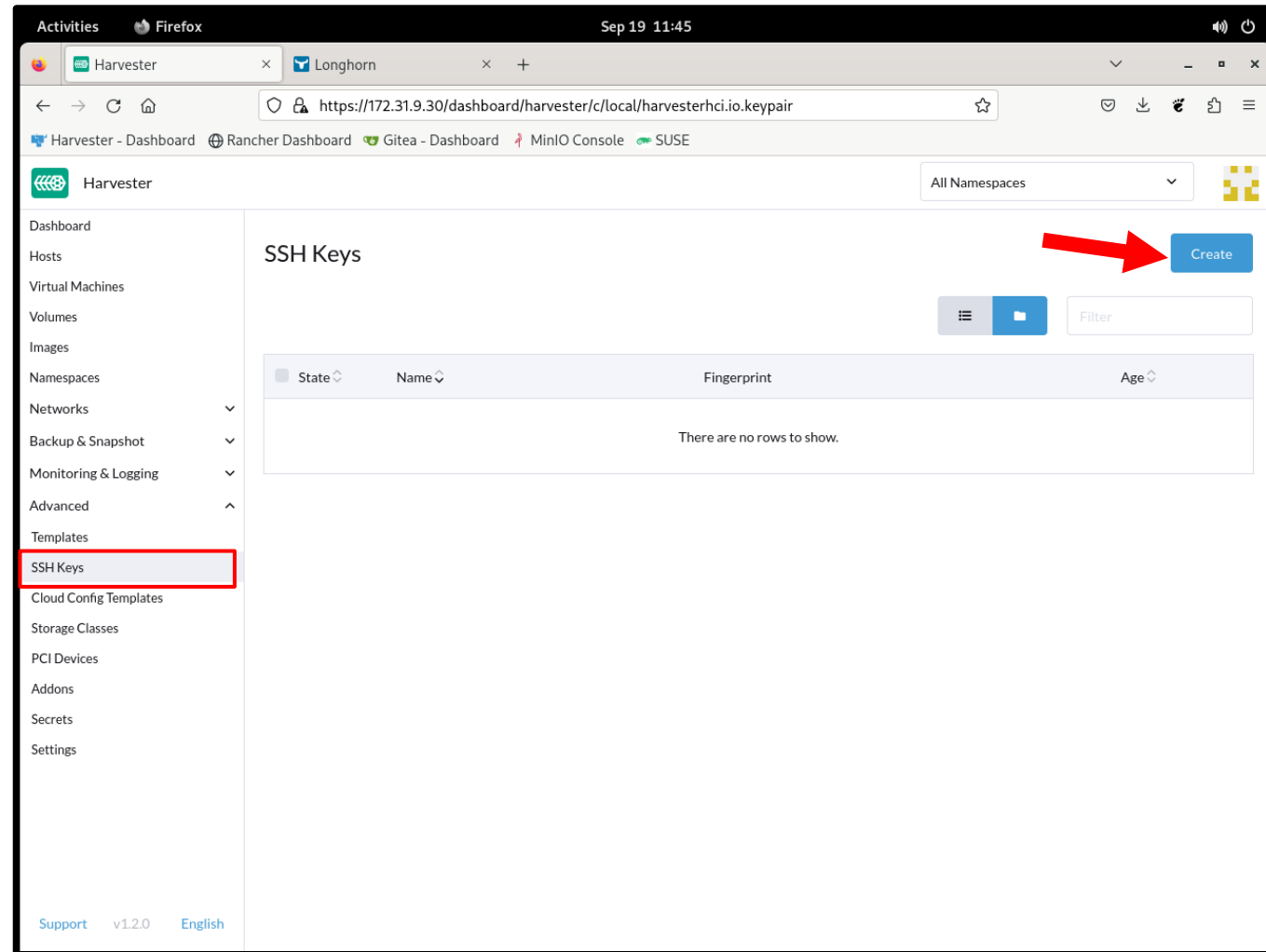
SSH Keys

- 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.



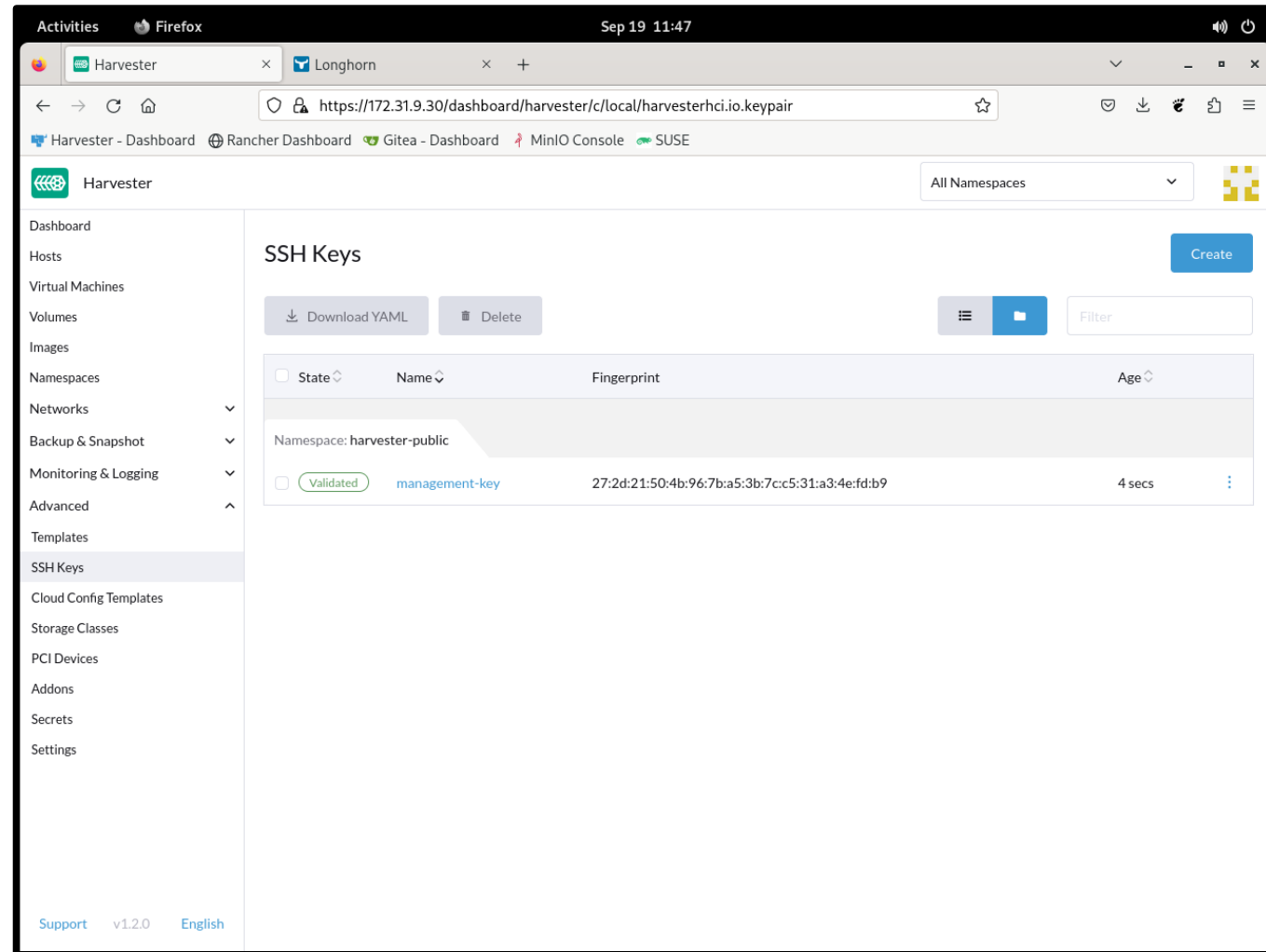
View and Create SSH Keys

- 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



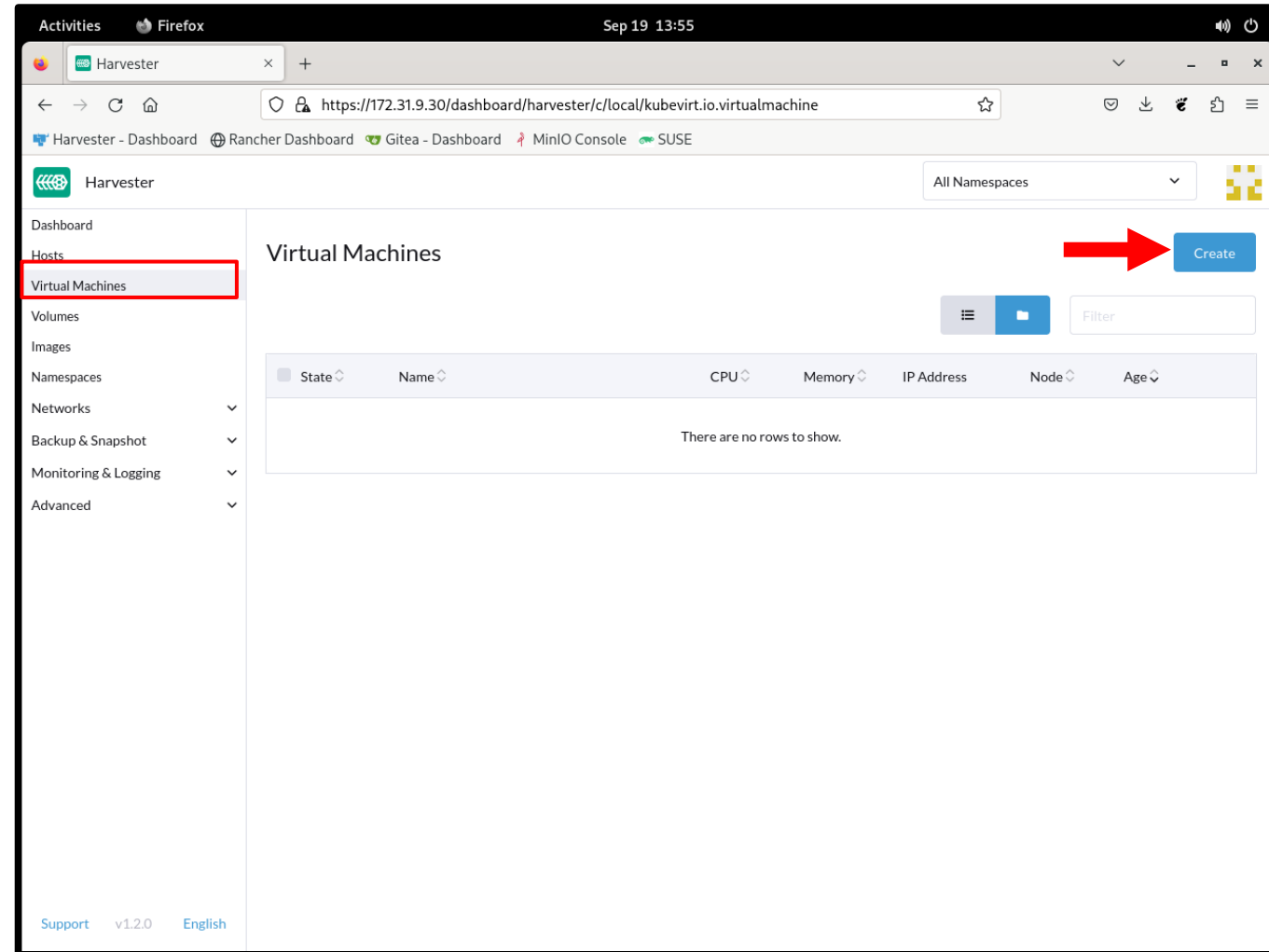
Virtual Machine Creation Workflow Options

- 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**

Activities Firefox Sep 19 13:58

Harvester

https://172.31.9.30/dashboard/harvester/c/local/kubevirt.io.virtualmachine/create#basics

Harvester - Dashboard Rancher Dashboard Gitea - Dashboard MinIO Console SUSE

Harvester All Namespaces

Dashboard
Hosts
Virtual Machines
Volumes
Images
Namespaces
Networks
Backup & Snapshot
Monitoring & Logging
Advanced

Virtual Machine: Create

☒ Single Instance ☐ Multiple Instance

Namespace * harvester-public

Name * vm01

Description
Any text you want that better describes this resource

☐ Use VM Template:

Basics
Volumes
Networks
Node Scheduling
VM Scheduling
Advanced Options
Instance Labels

Basics

CPU * 2 C

Memory * 4 GiB

SSHKey
harvester-public/management-key

Support v1.2.0 English

Cancel Edit as YAML Create



Virtual Machine: Create Multiple Instances

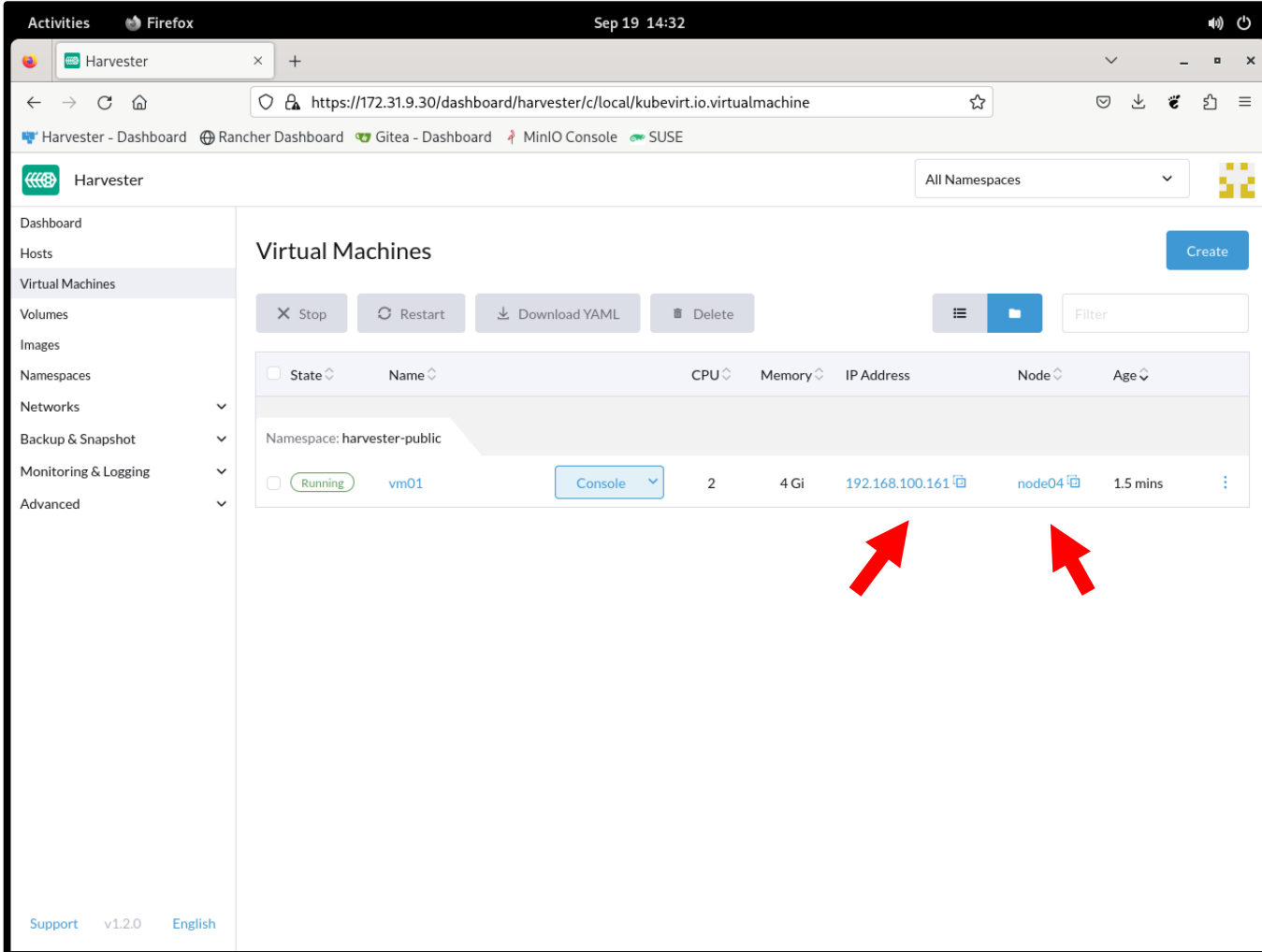
- To create multiple unique instances of a VM from a single set of configuration options select: **Multiple Instances**
- Enter a **Name Prefix**.
 - This is a text string to be prepended to the VM names when they are created.
- Enter the **Count**
 - To specify the number of instances to create.
- All 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.

The screenshot shows the Harvester web interface in a Firefox browser window. The URL is `https://172.31.9.30/dashboard/harvester/c/local/kubevirt.io.virtualmachine/create#basics`. The left sidebar contains a navigation menu with options: Dashboard, Hosts, Virtual Machines (selected), Volumes, Images, Namespaces, Networks, Backup & Snapshot, Monitoring & Logging, and Advanced. The main content area is titled 'Virtual Machine: Create' and features two radio buttons: 'Single Instance' and 'Multiple Instance' (which is selected). Below these are four input fields: 'Namespace' (set to 'harvester-public'), 'Name Prefix' (set to 'set01'), 'Description' (placeholder: 'Any text you want that better descri...'), and 'Count' (set to '4'). A checkbox 'Use VM Template:' is present but unchecked. Below the main form is a 'Basics' section with a sidebar menu containing 'Volumes', 'Networks', 'Node Scheduling', 'VM Scheduling', 'Advanced Options', and 'Instance Labels'. The 'Basics' section includes fields for 'CPU' (set to '2'), 'Memory' (set to '4 GiB'), and 'SSHKey' (set to 'harvester-public/management-key'). At the bottom right, there are 'Cancel' and 'Create' buttons. Three red arrows point to the 'Multiple Instance' radio button, the 'Name Prefix' field, and the 'Count' field.



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.



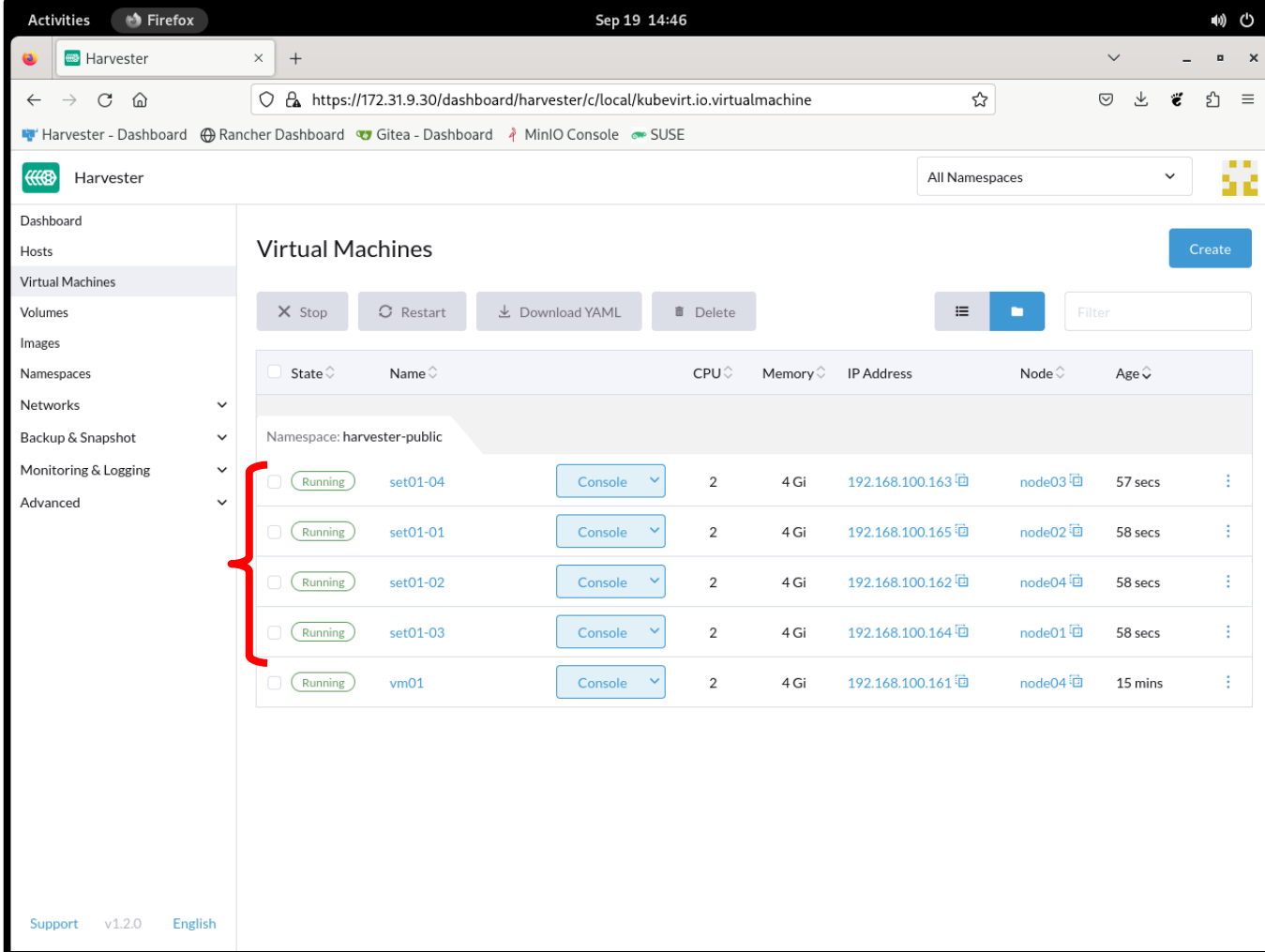
The screenshot shows the Harvester web interface in a Firefox browser. The URL is `https://172.31.9.30/dashboard/harvester/c/local/kubevirt.io.virtualmachine`. The left sidebar contains navigation links: Dashboard, Hosts, Virtual Machines (selected), Volumes, Images, Namespaces, Networks, Backup & Snapshot, Monitoring & Logging, and Advanced. The main panel is titled "Virtual Machines" and includes a "Create" button and action buttons (Stop, Restart, Download YAML, Delete). A table lists VMs in the "harvester-public" namespace. One VM, "vm01", is in a "Running" state. It has 2 CPUs, 4 Gi of memory, IP address 192.168.100.161, and is running on node04. Two red arrows point to the IP address and the node name.

State	Name	CPU	Memory	IP Address	Node	Age
Running	vm01	2	4 Gi	192.168.100.161	node04	1.5 mins



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.



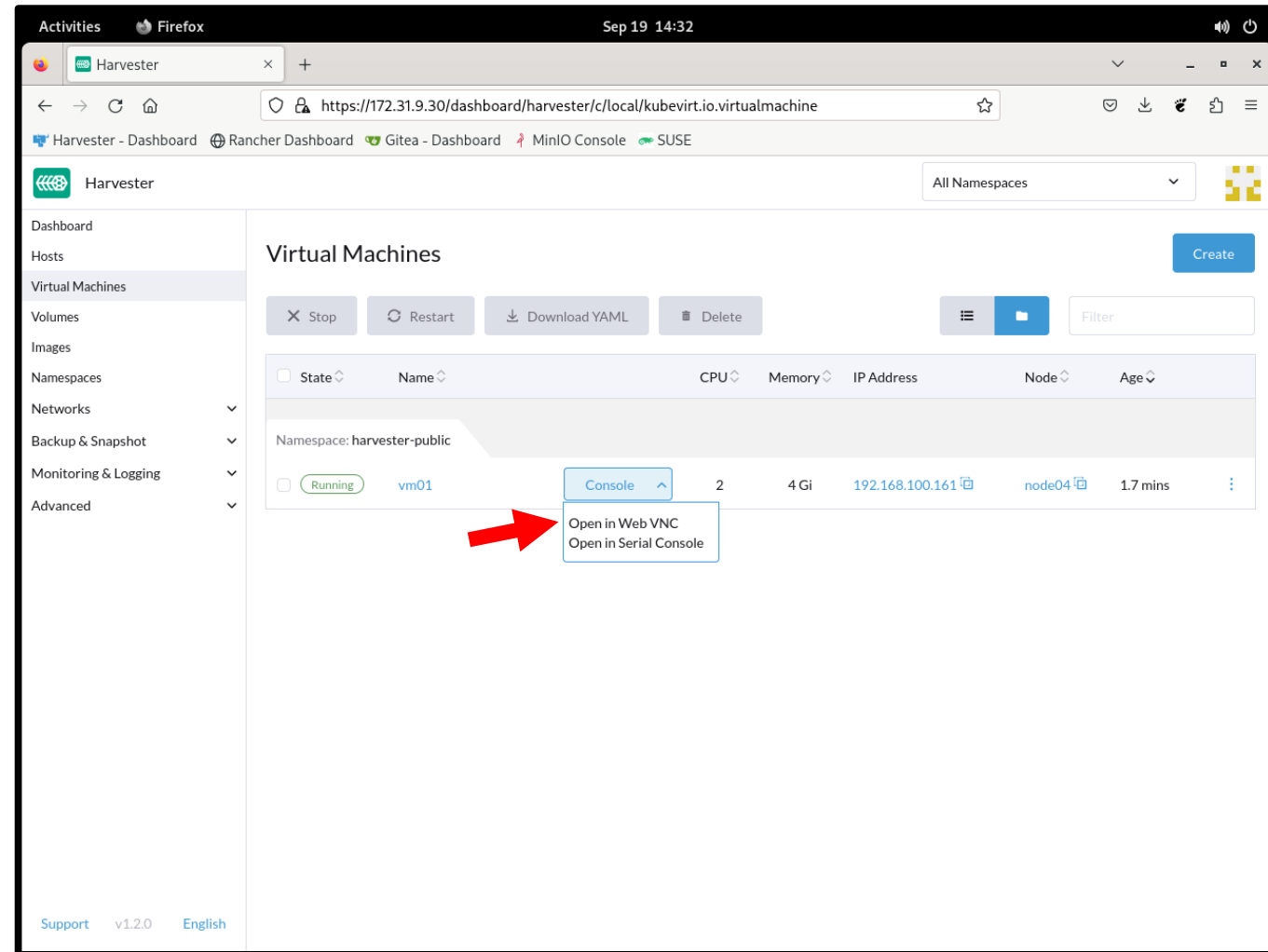
The screenshot shows the Harvester web interface in a Firefox browser. The URL is `https://172.31.9.30/dashboard/harvester/c/local/kubevirt.io.virtualmachine`. The left sidebar contains navigation links: Dashboard, Hosts, Virtual Machines (selected), Volumes, Images, Namespaces, Networks, Backup & Snapshot, Monitoring & Logging, and Advanced. The main panel is titled "Virtual Machines" and includes action buttons: Stop, Restart, Download YAML, and Delete. A table lists the VMs in the "harvester-public" namespace. A red bracket highlights the first four rows, which all have a "Running" status and names starting with "set01-".

State	Name	CPU	Memory	IP Address	Node	Age
Running	set01-04	2	4 Gi	192.168.100.163	node03	57 secs
Running	set01-01	2	4 Gi	192.168.100.165	node02	58 secs
Running	set01-02	2	4 Gi	192.168.100.162	node04	58 secs
Running	set01-03	2	4 Gi	192.168.100.164	node01	58 secs
Running	vm01	2	4 Gi	192.168.100.161	node04	15 mins



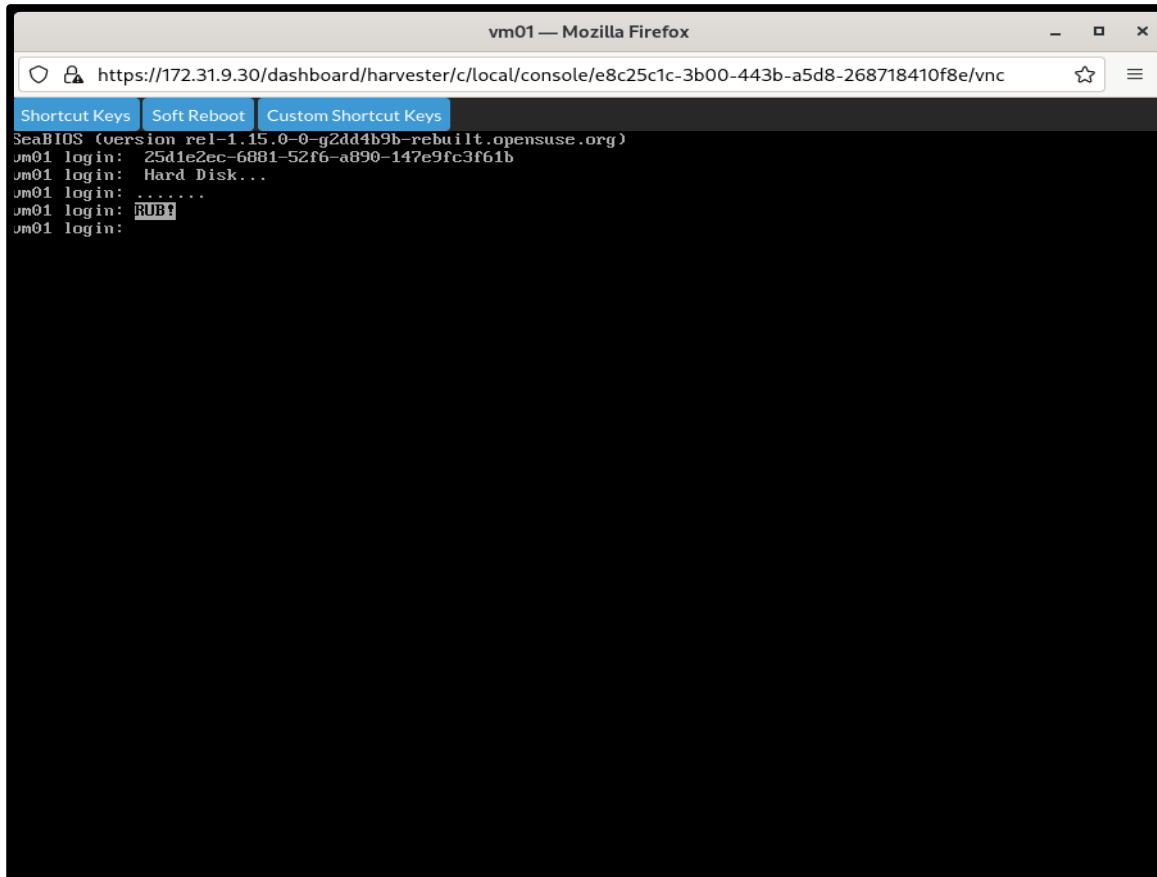
Virtual Machine – Access VM Console

- 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.



vm01 — Mozilla Firefox

https://172.31.9.30/dashboard/harvester/c/local/console/e8c25c1c-3b00-443b-a5d8-268718410f8e/vnc

Shortcut Keys Soft Reboot Custom Shortcut Keys

SeaBIOS (version rel-1.15.0-0-g2dd4b9b-rebuilt.opensuse.org)

vm01 login: 25d1e2ec-6881-52f6-a890-147e9fc3f61b

vm01 login: Hard Disk...

vm01 login:

vm01 login: RUBI

vm01 login:



vm01 — Mozilla Firefox

https://172.31.9.30/dashboard/harvester/c/local/console/e8c25c1c-3b00-443b-a5d8-268718410f8e/serial

vm01 login: █



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.



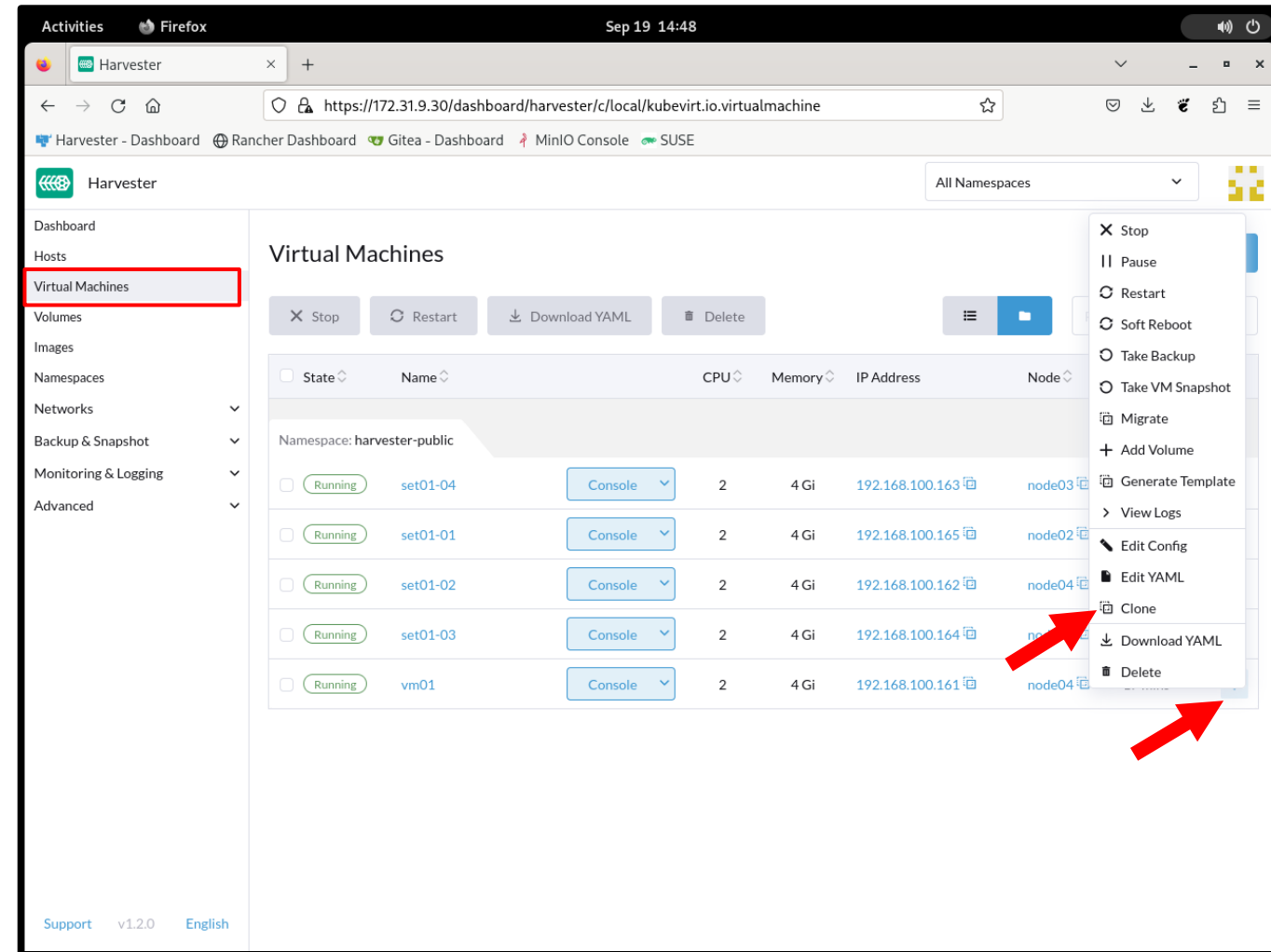
Existing Configuration as a Starting Point

- **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.



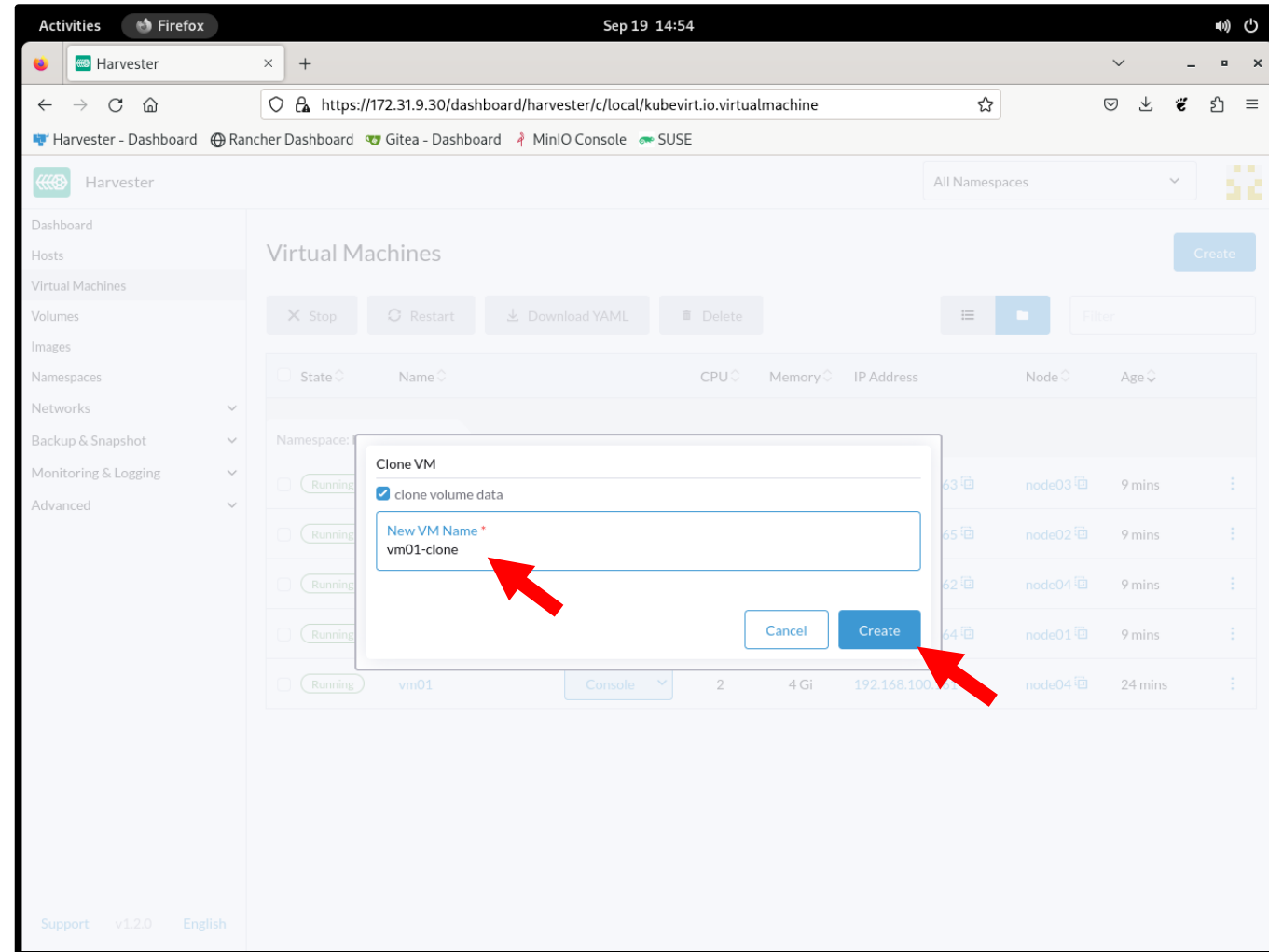
Virtual Machine: Clone a VM

- 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.



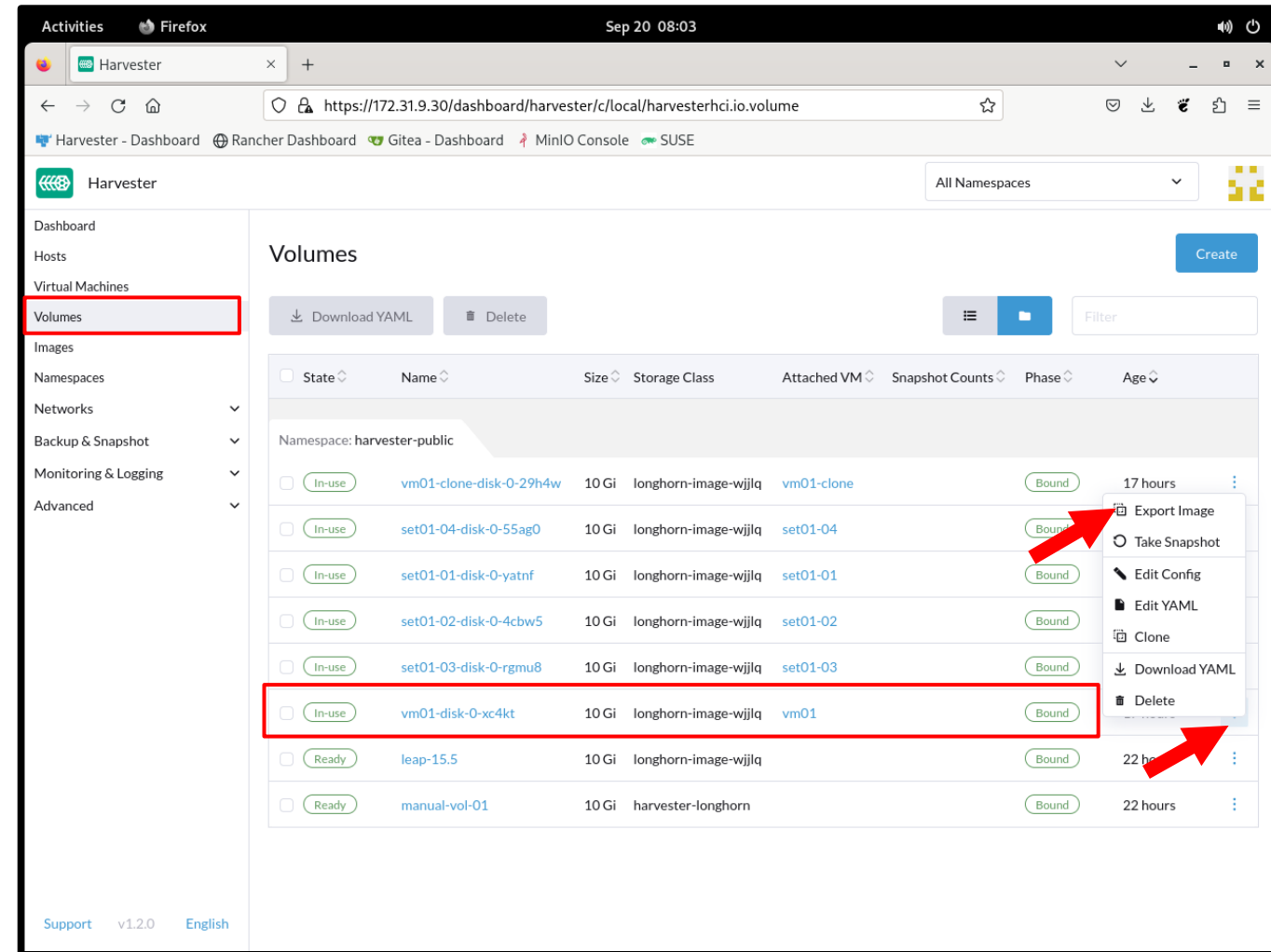
Only Disk Image as a Starting Point

- **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

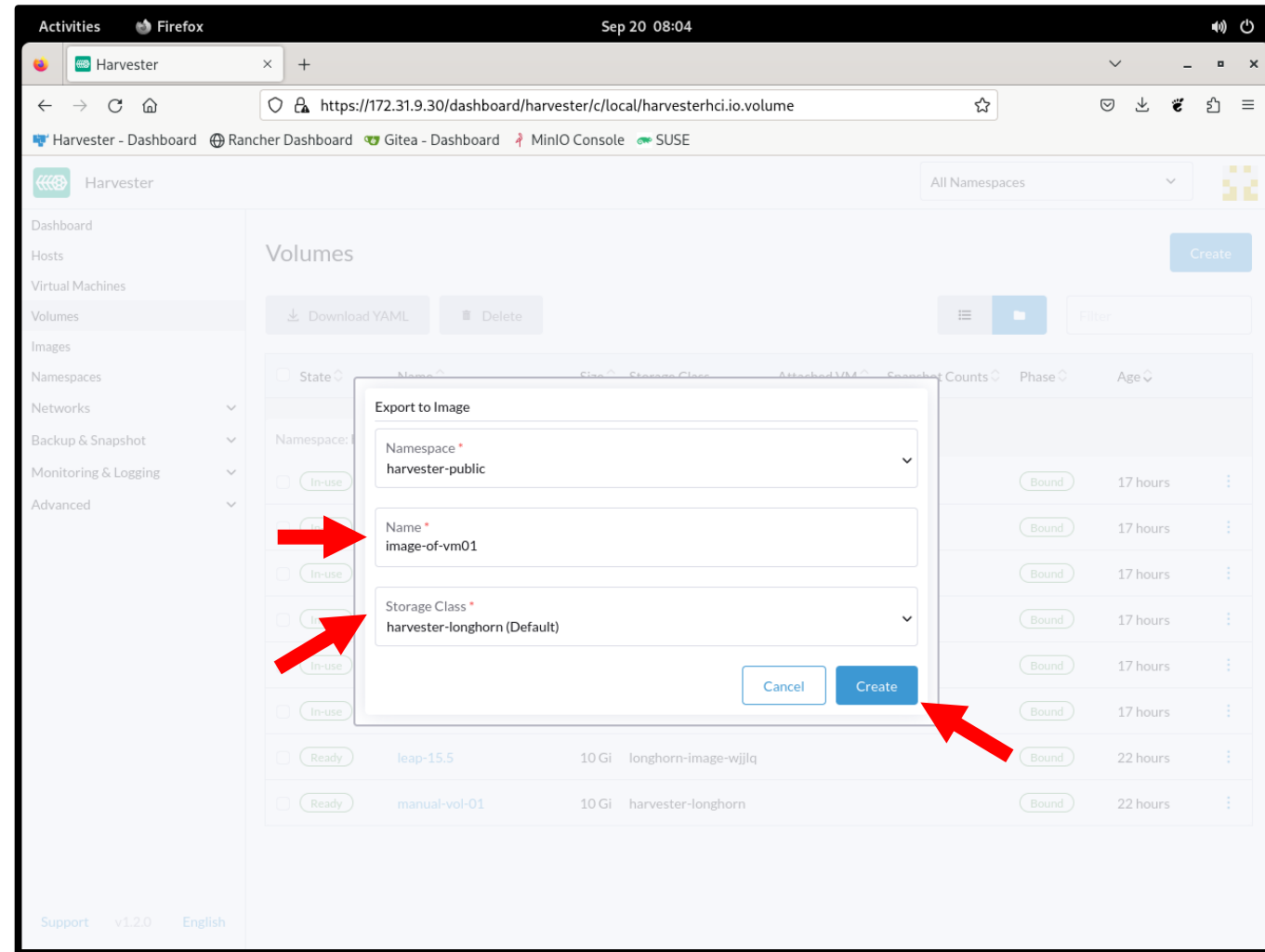
- On the **Volumes** screen, locate the volume use as the root disk of the VM you want to use as an image.
- From the **3 dot** menu on the right of the volume select **Export Image**.



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:  
$6$Sa1TsaLt$47wJg0F2AZFKYiSxDKM8beRyIM5JYg281g  
Eiry25V5FbC4nDuk90HpxC0XtsJkY10YZNYHWMnTLhXsgP  
wL5SS.  
    sudo: ALL=(ALL) NOPASSWD:ALL
```



Cloud Config: User Data – Create User with SSH Key

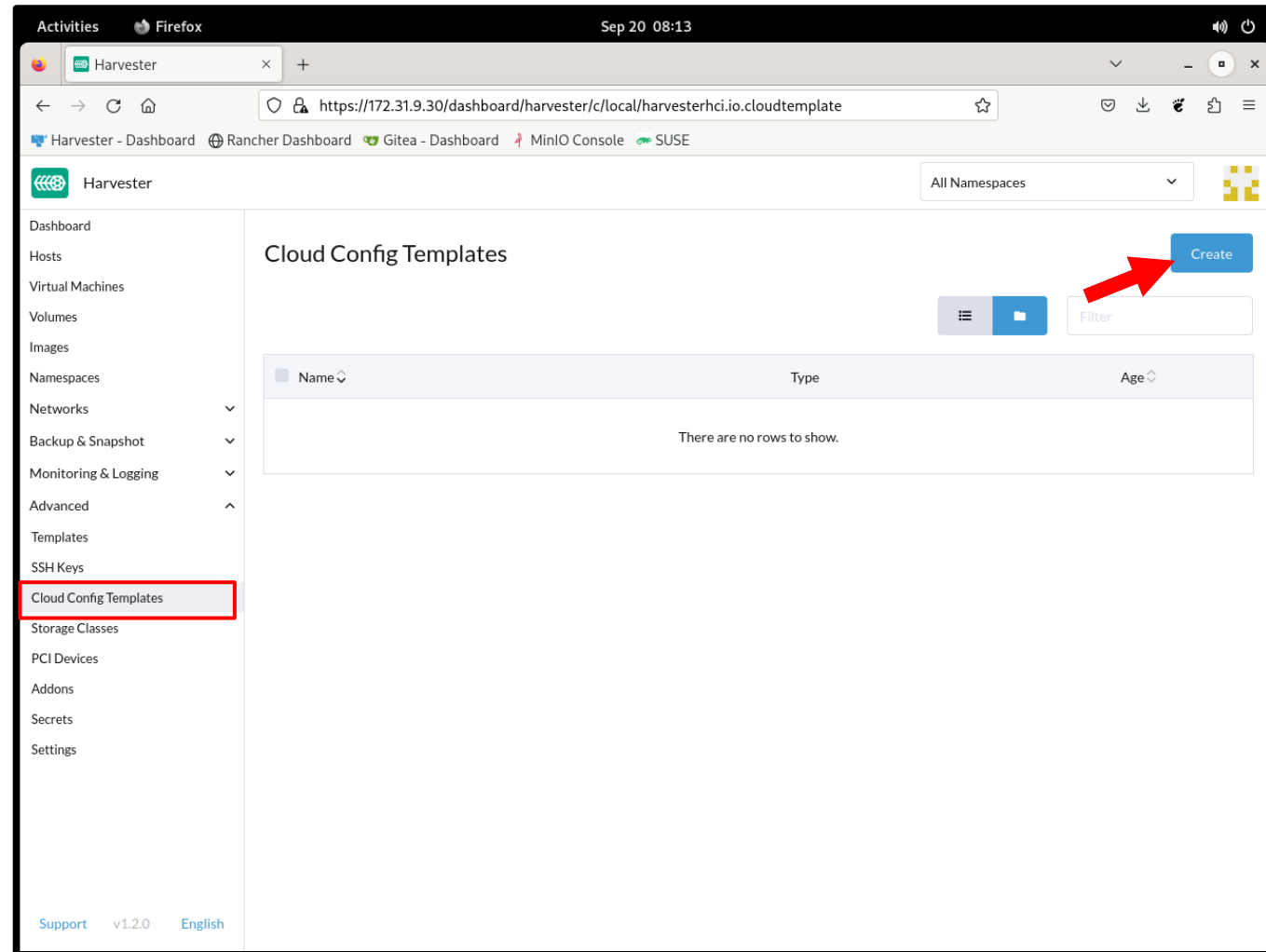
- Create a user named **tux** with a primary group of **users**.
- Disable the password for the **tux** user.
- Enable **tux** to use sudo without requiring a password.
- Add two SSH keys to the **tux** users `authorized_keys` file.

```
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**

The screenshot shows the Harvester web interface in a Firefox browser window. The address bar displays the URL: `https://172.31.9.30/dashboard/harvester/c/local/harvesterhci.io.virtualmachinetemplateversion`. The left sidebar contains a navigation menu with the following items: Dashboard, Hosts, Virtual Machines, Volumes, Images, Namespaces, Networks, Backup & Snapshot, Monitoring & Logging, Advanced, Templates (highlighted with a red box), SSH Keys, Cloud Config Templates, Storage Classes, PCI Devices, Addons, Secrets, and Settings. The main content area is titled 'Templates' and features a 'Download YAML' button and a 'Filter' input field. A table lists the available templates:

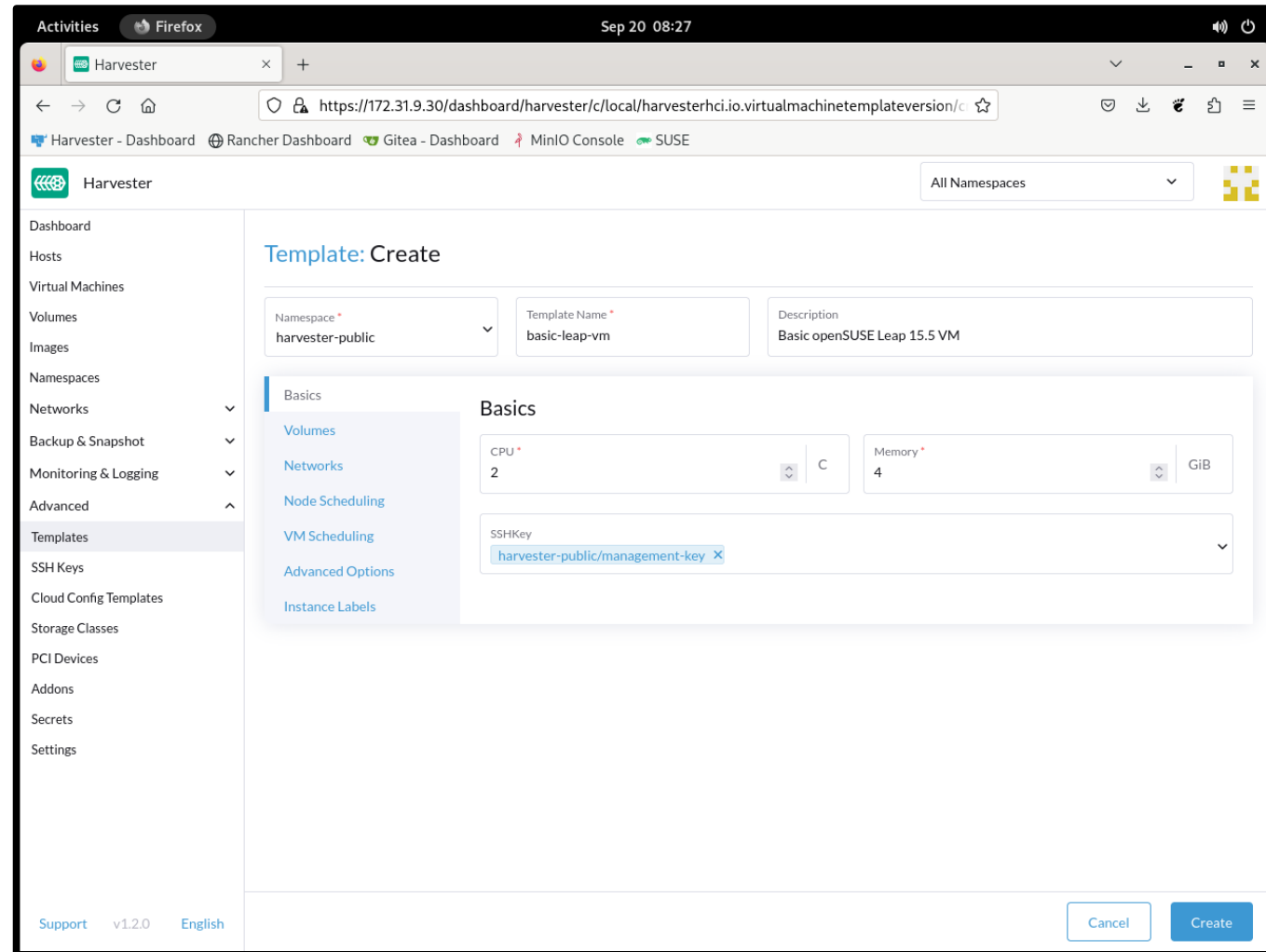
State	Name	Namespace	Default version	Age
Active	iso-image-base-version	harvester-public	✓	11 days
Active	raw-image-base-version	harvester-public	✓	11 days
Active	windows-iso-image-base-version	harvester-public	✓	11 days
Active	windows-raw-image-base-version	harvester-public	✓	11 days

A red arrow points to the 'Create' button in the top right corner of the Templates section.



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.

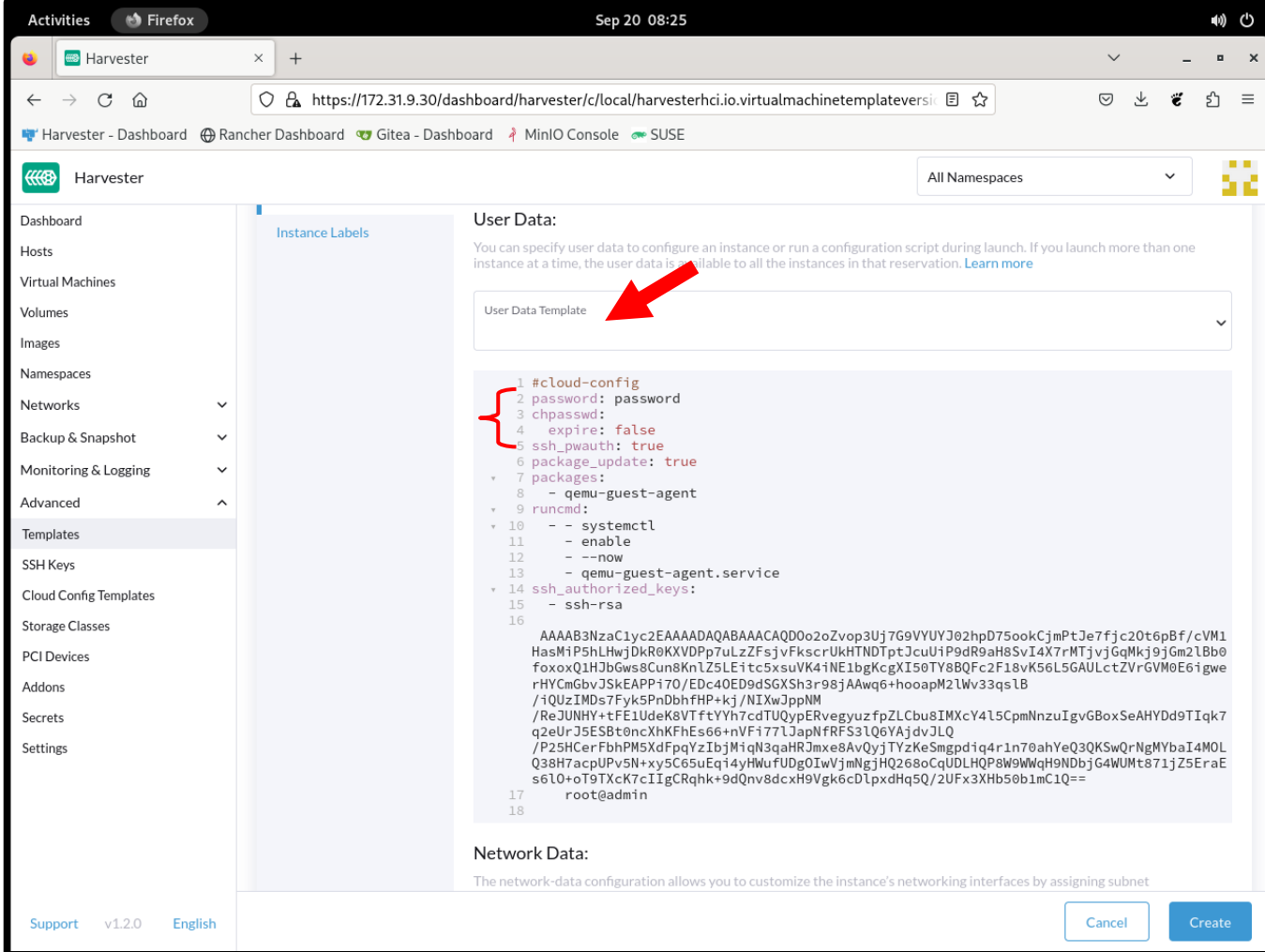


The screenshot shows the 'Template: Create' page in the Harvester web interface. The browser window has a single tab titled 'Harvester' and the address bar shows the URL `https://172.31.9.30/dashboard/harvester/c/local/harvesterhci.io.virtualmachinetemplateversion/c`. The page header includes the Harvester logo, a dropdown for 'All Namespaces', and a user icon. A left sidebar contains a navigation menu with items: Dashboard, Hosts, Virtual Machines, Volumes, Images, Namespaces, Networks, Backup & Snapshot, Monitoring & Logging, Advanced, Templates (highlighted), SSH Keys, Cloud Config Templates, Storage Classes, PCI Devices, Addons, Secrets, and Settings. The main content area is titled 'Template: Create' and features three input fields at the top: 'Namespace' (set to 'harvester-public'), 'Template Name' (set to 'basic-leap-vm'), and 'Description' (set to 'Basic openSUSE Leap 15.5 VM'). Below these is a 'Basics' section with a sub-menu on the left containing 'Volumes', 'Networks', 'Node Scheduling', 'VM Scheduling', 'Advanced Options', and 'Instance Labels'. The 'Basics' form includes 'CPU' (set to 2), 'Memory' (set to 4 GiB), and an 'SSHKey' dropdown (set to 'harvester-public/management-key'). At the bottom right are 'Cancel' and 'Create' buttons. The footer shows 'Support', 'v1.2.0', and 'English'.



VM Templates: Create

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



The screenshot shows the Harvester web interface in a Firefox browser window. The URL is `https://172.31.9.30/dashboard/harvester/c/local/harvesterhci.io.virtualmachinetemplateversions`. The left sidebar contains a navigation menu with options like Dashboard, Hosts, Virtual Machines, Volumes, Images, Namespaces, Networks, Backup & Snapshot, Monitoring & Logging, Advanced, Templates, SSH Keys, Cloud Config Templates, Storage Classes, PCI Devices, Addons, Secrets, and Settings. The 'Templates' section is selected. The main content area is titled 'Instance Labels' and 'User Data:'. The 'User Data' section explains that user data can be used to configure an instance or run a configuration script during launch. A red arrow points to the 'User Data Template' dropdown menu. Below the dropdown is a text area containing a cloud-config script. The script sets a password, enables SSH, and configures the system. The script is as follows:

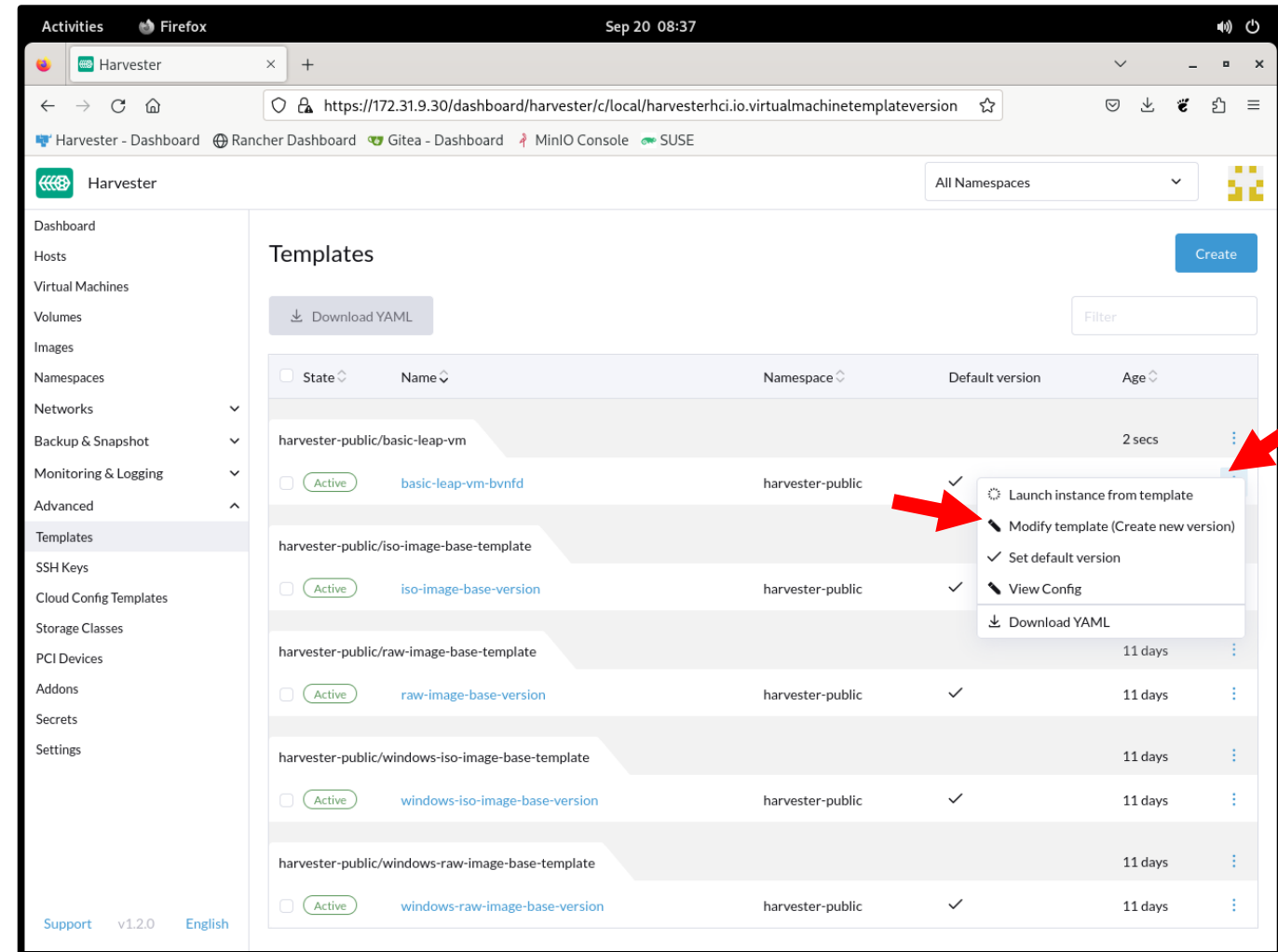
```
1 #cloud-config
2 password: password
3 chpasswd:
4   expire: false
5 ssh_pwauth: true
6 package_update: true
7 packages:
8   - qemu-guest-agent
9 runcmd:
10   - systemctl
11     enable
12     --now
13     qemu-guest-agent.service
14 ssh_authorized_keys:
15   - ssh-rsa
16     AAAAB3NzaC1yc2EAAAADAQABAAQCAQD0o2oZvop3Uj7G9VYUYJ02hpD75ookCjmPtJe7fjc20t6pBf/cVM1
17     HasMiP5hLHwjDkR0KXVDPp7uLzZFsjvFkscrUkHTNDTptJcuUiP9dR9aH8SvI4X7rMTjjvGqMkj9jGm2LBb0
18     foxoxQ1HJbGws8Cun8Kn1Z5LEitc5xsuVK4iNE1bgKcgXI50TY8BQFc2F18vK56L5GAULctZvrGVM0E6igwe
19     rHYCmGbvJSkEAPPi70/EDc40ED9dSGXSh3r98jAAwq6+hoopM2LWv33qslB
20     /iQUzIMDs7Fyk5PnDbhfHP+kj/NIXwJppNM
21     /ReJUNHY+tFE1UdeK8VTftYYh7cdTUQypERvegyuzfpZLCbu8IMXcY4L5CpmNzuIgvGBoxSeAHYDd9TIqk7
22     q2eUrJ5ESBt0ncXhKfHes66+nVfi77LJapNfRFS3LQ6YajdvJLQ
23     /P25HCerFbhPM5XdFpqYzIbjMiQ3qaHRJmxe8AvQyJTYzKeSmgpdq4r1n70ahYeQ3QKSQrNgMYbaI4MOL
24     Q38H7acpUPv5N+xy5C65uEqi4yHWufUDgOIwVjmNgjHQ268oCqUDLHQP8W9WqH9NDbjG4WUMt871jZ5EraE
25     s6LO+oT9TXcK7cIIgCRqhk+9dQnv8dcxH9Vgk6cDlpxdHq5Q/2UFx3XHb50b1mC1Q==
26     root@admin
```

At the bottom of the 'User Data' section, there is a 'Network Data' section with a description: 'The network-data configuration allows you to customize the instance's networking interfaces by assigning subnet'. At the bottom right of the interface, there are 'Cancel' and 'Create' buttons.



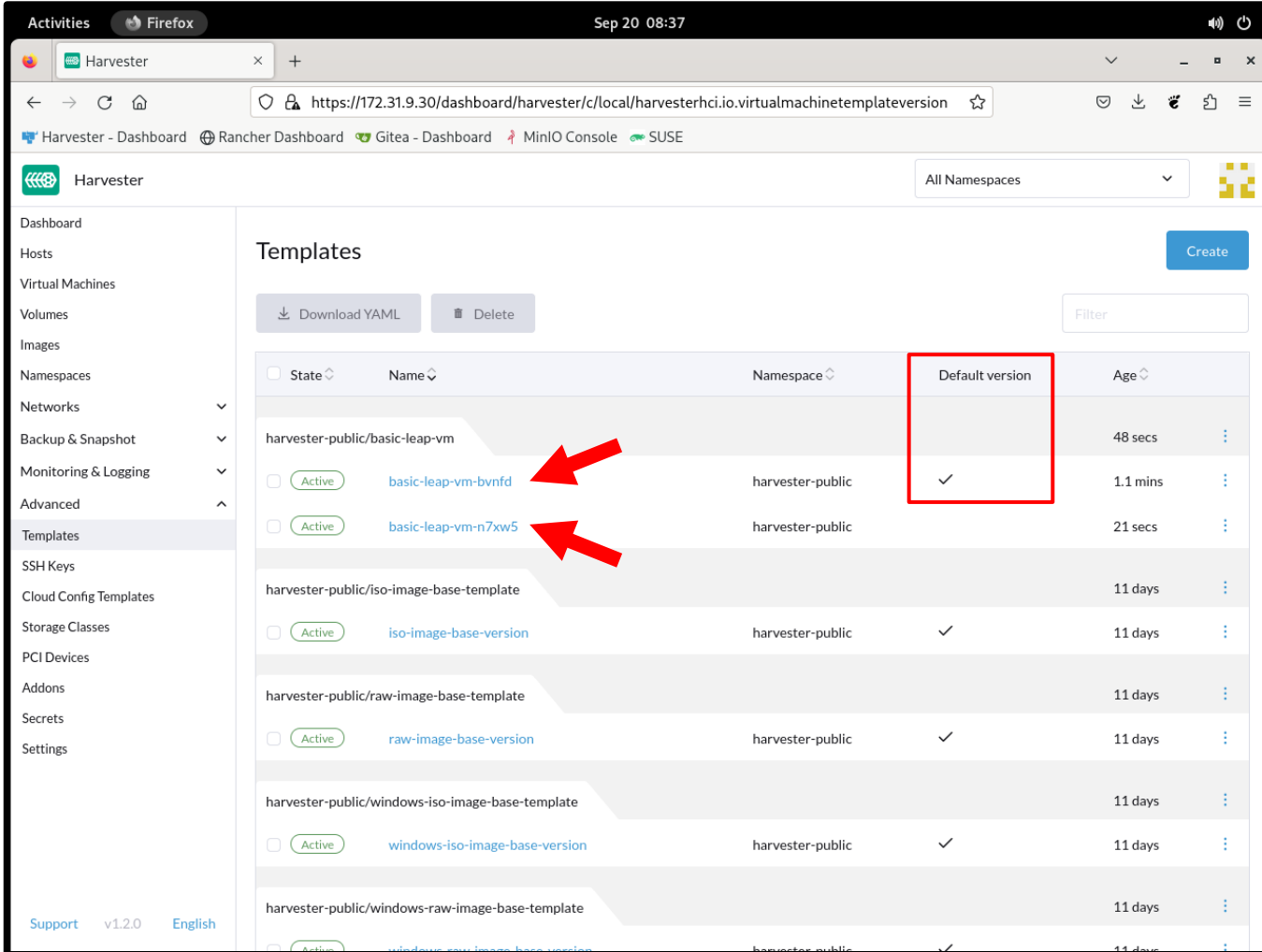
VM Templates: Modify Templates

- 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.



The screenshot shows the Harvester web interface in a Firefox browser window. The URL is <https://172.31.9.30/dashboard/harvester/c/local/harvesterhci.io.virtualmachinetemplateversion>. The left sidebar contains a navigation menu with items like Dashboard, Hosts, Virtual Machines, Volumes, Images, Namespaces, Networks, Backup & Snapshot, Monitoring & Logging, Advanced, Templates (selected), SSH Keys, Cloud Config Templates, Storage Classes, PCI Devices, Addons, Secrets, and Settings. The main content area is titled 'Templates' and features a 'Create' button and a 'Filter' input. Below this is a table of templates:

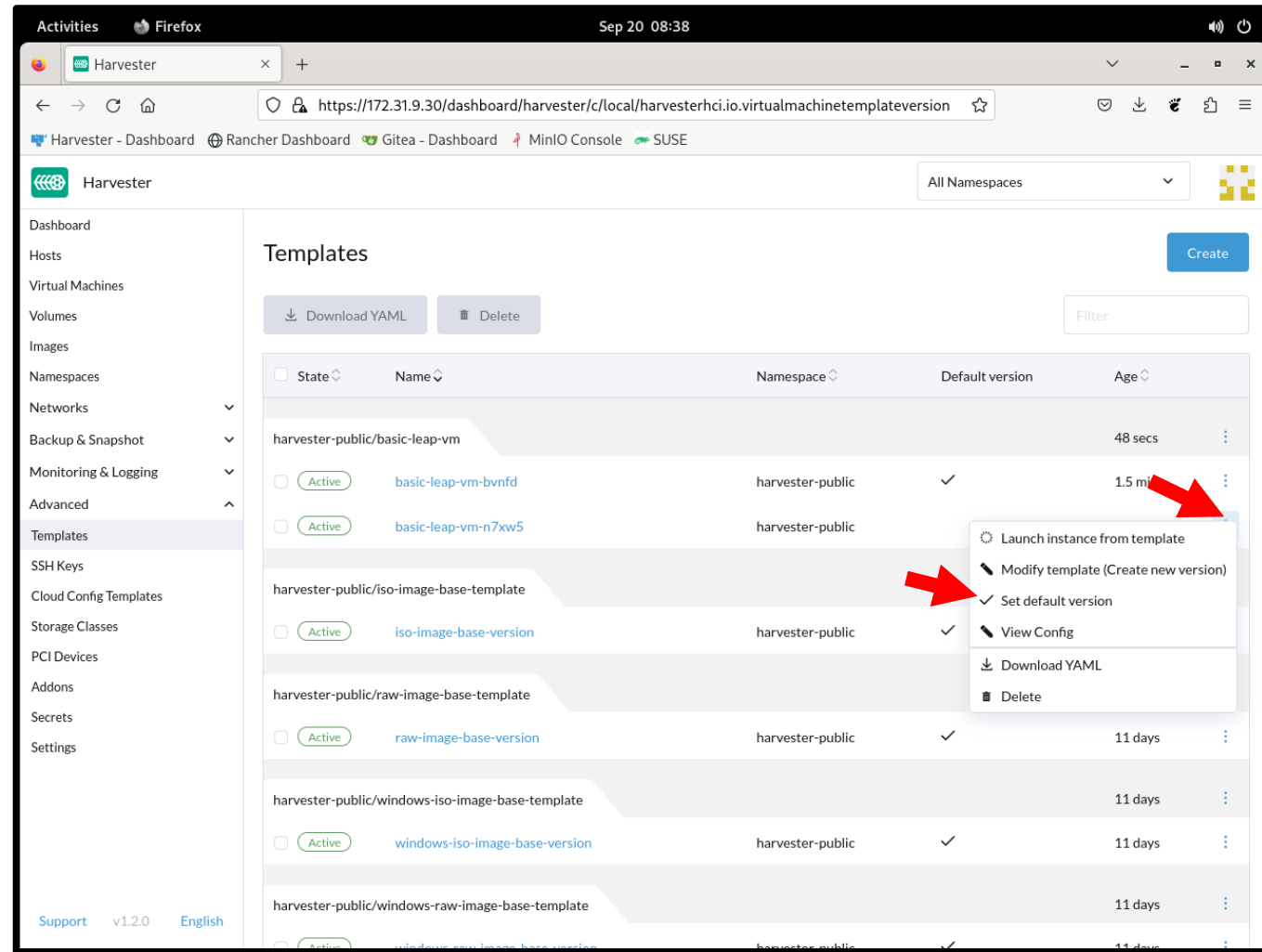
State	Name	Namespace	Default version	Age
harvester-public/basic-leap-vm				
<input type="checkbox"/> Active	basic-leap-vm-bvnfd	harvester-public	✓	48 secs
<input type="checkbox"/> Active	basic-leap-vm-n7xw5	harvester-public		1.1 mins
harvester-public/iso-image-base-template				
<input type="checkbox"/> Active	iso-image-base-version	harvester-public	✓	21 secs
harvester-public/raw-image-base-template				
<input type="checkbox"/> Active	raw-image-base-version	harvester-public	✓	11 days
harvester-public/windows-iso-image-base-template				
<input type="checkbox"/> Active	windows-iso-image-base-version	harvester-public	✓	11 days
harvester-public/windows-raw-image-base-template				
<input type="checkbox"/> Active	windows-raw-image-base-version	harvester-public	✓	11 days

At the bottom of the page, there are links for 'Support', 'v1.2.0', and 'English'.



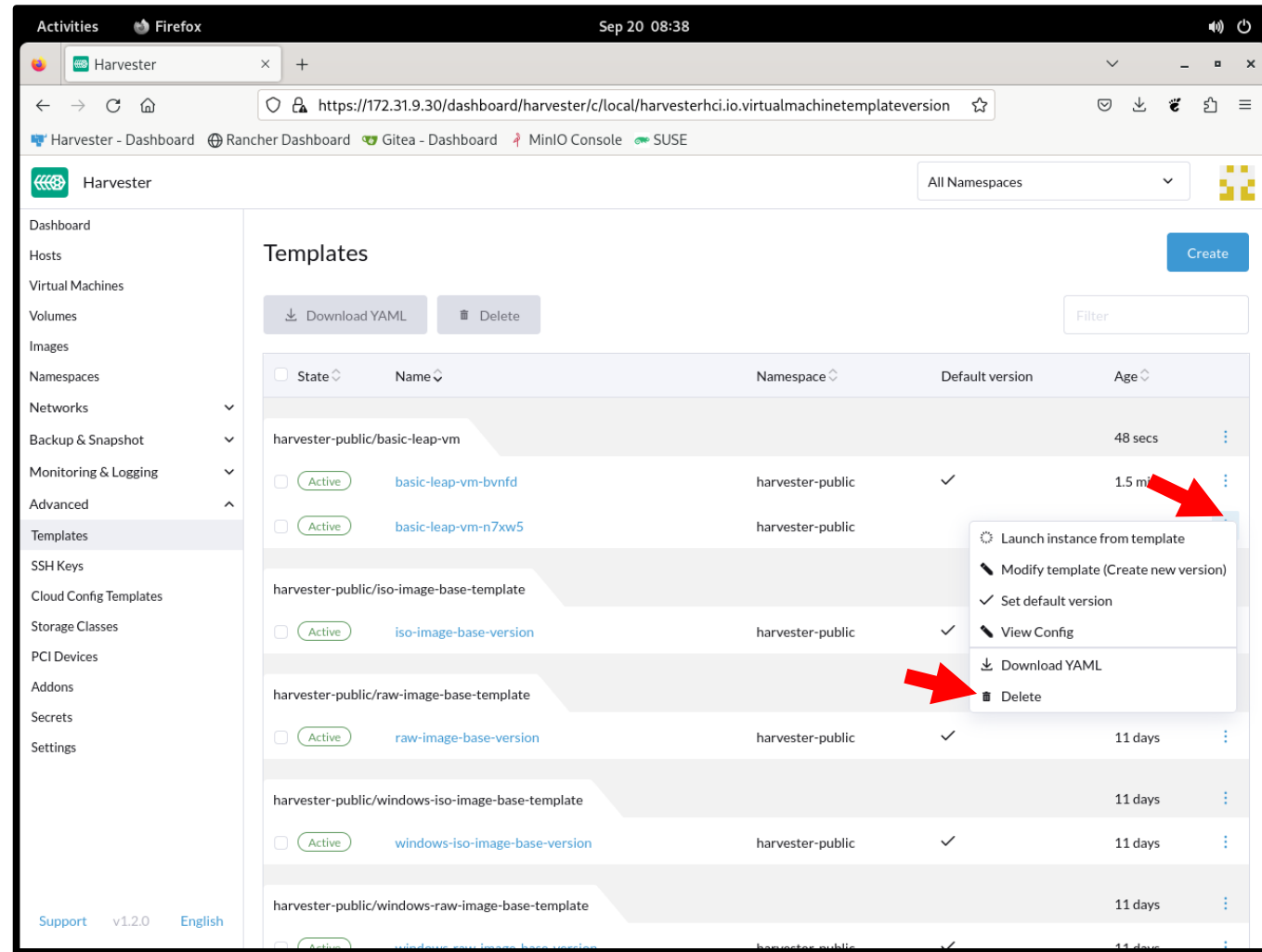
VM Templates: Default Version of a Template

- 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**



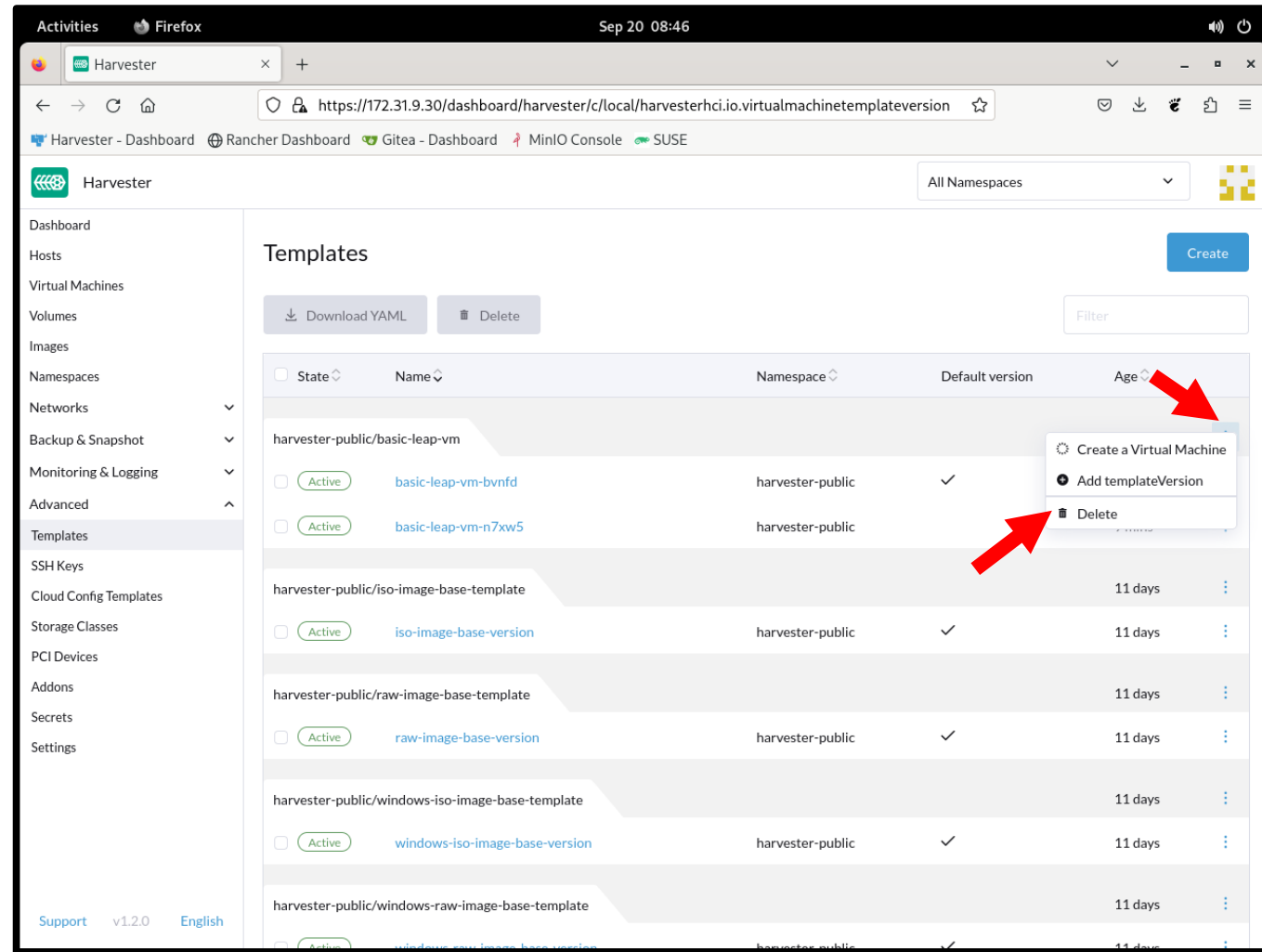
VM Templates: Delete a Version of a Template

- 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.



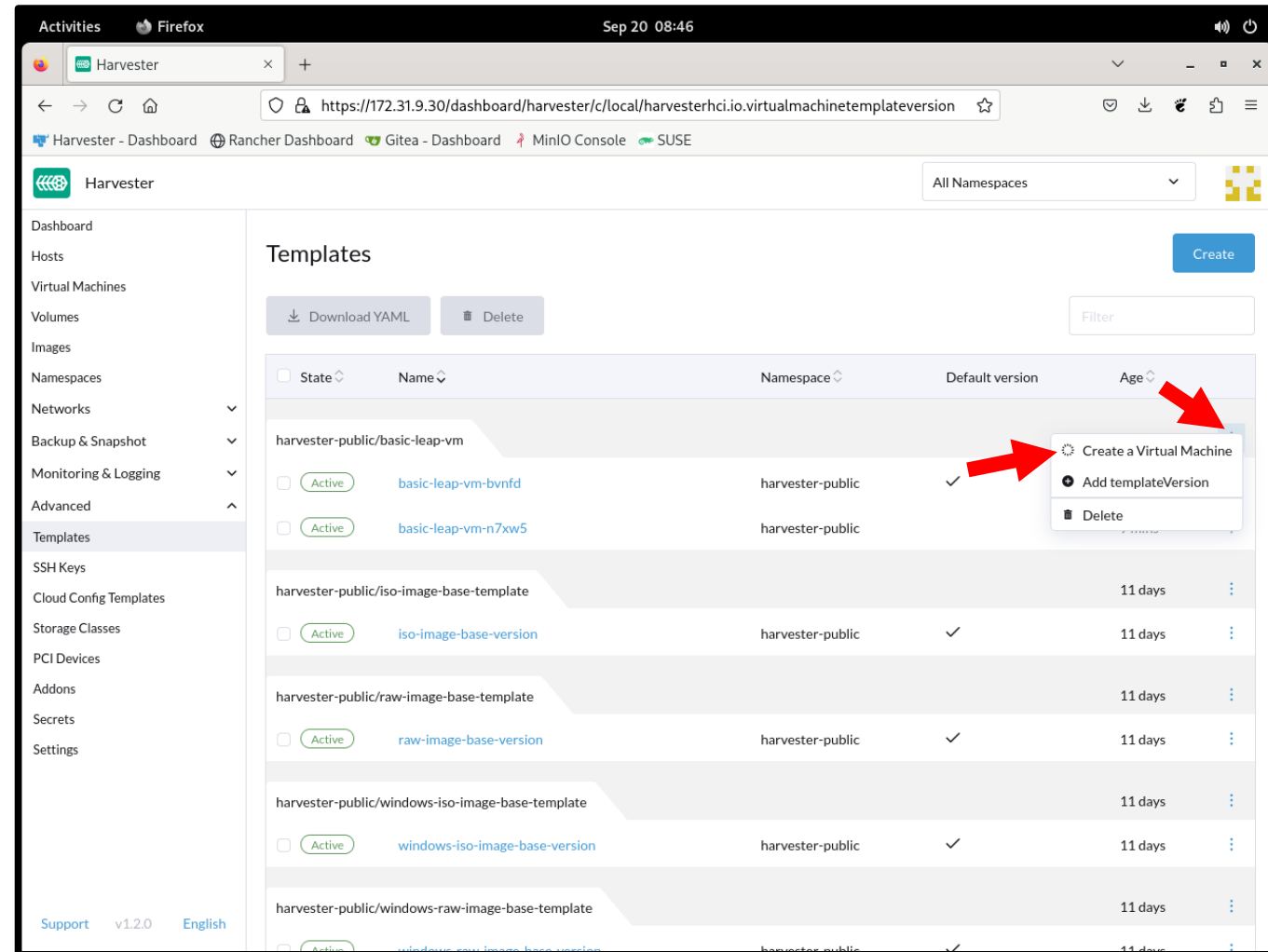
VM Templates: Delete a Template

- 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**



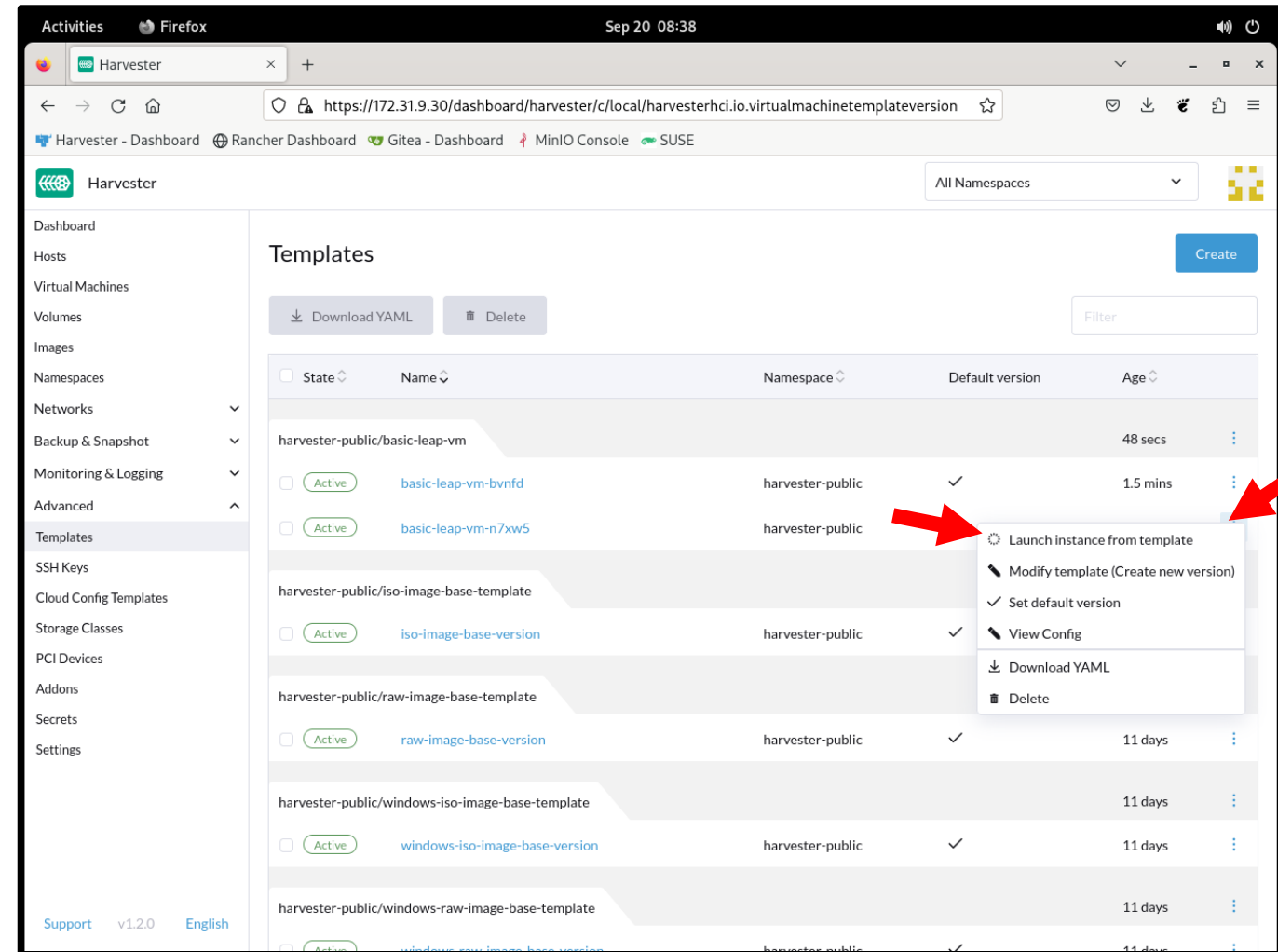
VM Templates: Launch Instance of a Template

- 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.

Activities Firefox Sep 20 08:55

Harvester

https://172.31.9.30/dashboard/harvester/c/local/kubevirt.io.virtualmachine/create?templateId=ha

Harvester - Dashboard Rancher Dashboard Gitea - Dashboard MiniIO Console SUSE

Harvester All Namespaces

Dashboard Hosts Virtual Machines Volumes Images Namespaces Networks Backup & Snapshot Monitoring & Logging Advanced

Virtual Machine: Create

☒ Single Instance ☐ Multiple Instance

Namespace * harvester-public Name * A unique name Description Any text you want that better describes this resource

☒ Use VM Template:

Template harvester-public/basic-leap-vm Version 1 (Default)

Basics

CPU * 2 Memory * 4 GiB

SSHKey harvester-public/management-key

Support v1.2.0 English Cancel Edit as YAML Create



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**

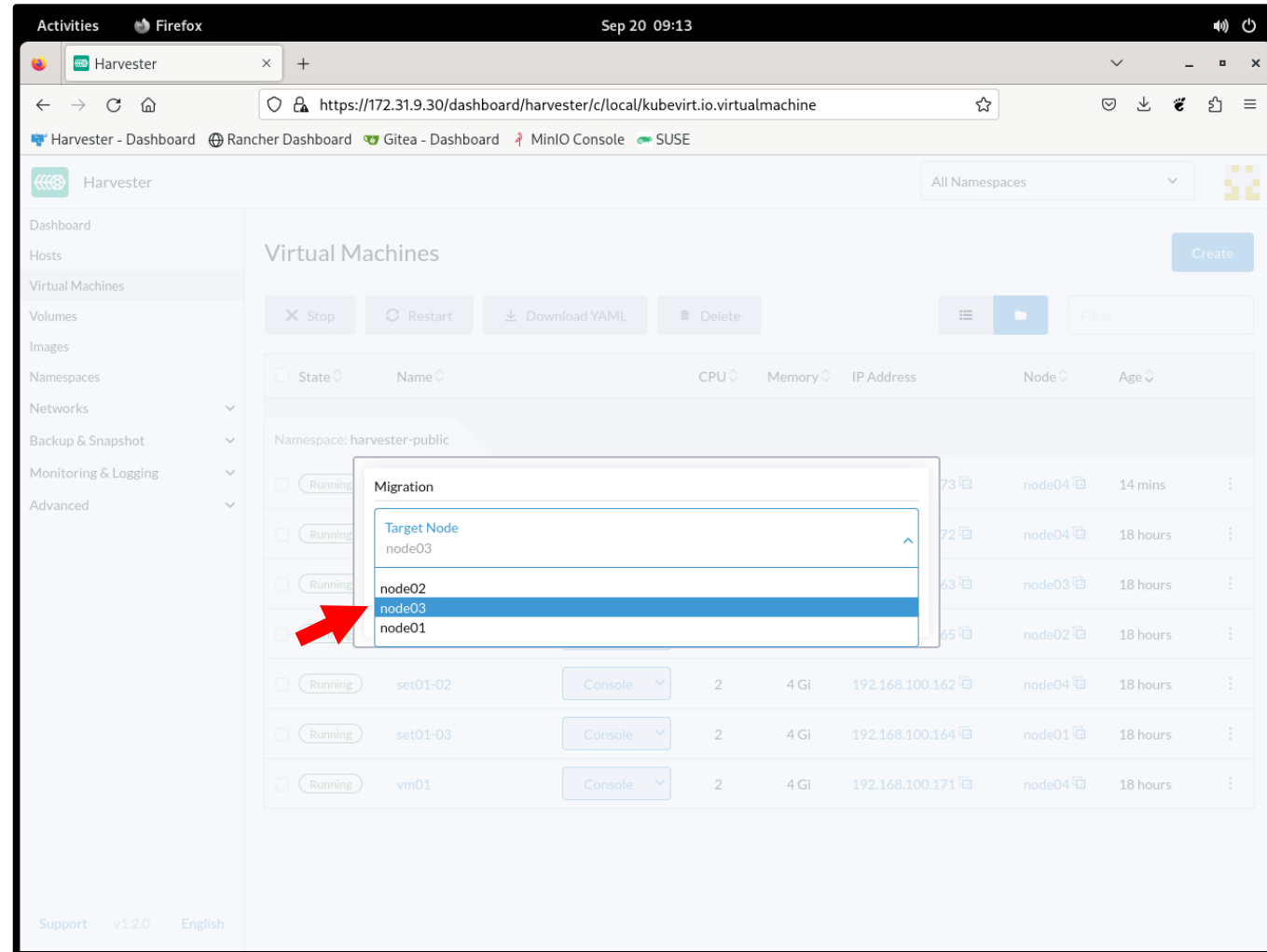
The screenshot shows the Harvester web interface in a Firefox browser. The left sidebar has a red box around the 'Virtual Machines' menu item. The main panel displays a table of Virtual Machines in the 'harvester-public' namespace. A context menu is open for the VM 'vm01', which is running on 'node04'. The 'Migrate' option is highlighted in the menu. Red arrows point to the 'Migrate' option, the 'node04' column, and the bottom right corner of the interface.

State	Name	CPU	Memory	IP Address	Node
Running	vm-from-template	2	4 Gi	192.168.100.173	node04
Running	vm01-clone	2	4 Gi	192.168.100.172	node04
Running	set01-04	2	4 Gi	192.168.100.163	node03
Running	set01-01	2	4 Gi	192.168.100.165	node02
Running	set01-02	2	4 Gi	192.168.100.162	node04
Running	set01-03	2	4 Gi	192.168.100.164	node01
Running	vm01	2	4 Gi	192.168.100.171	node04



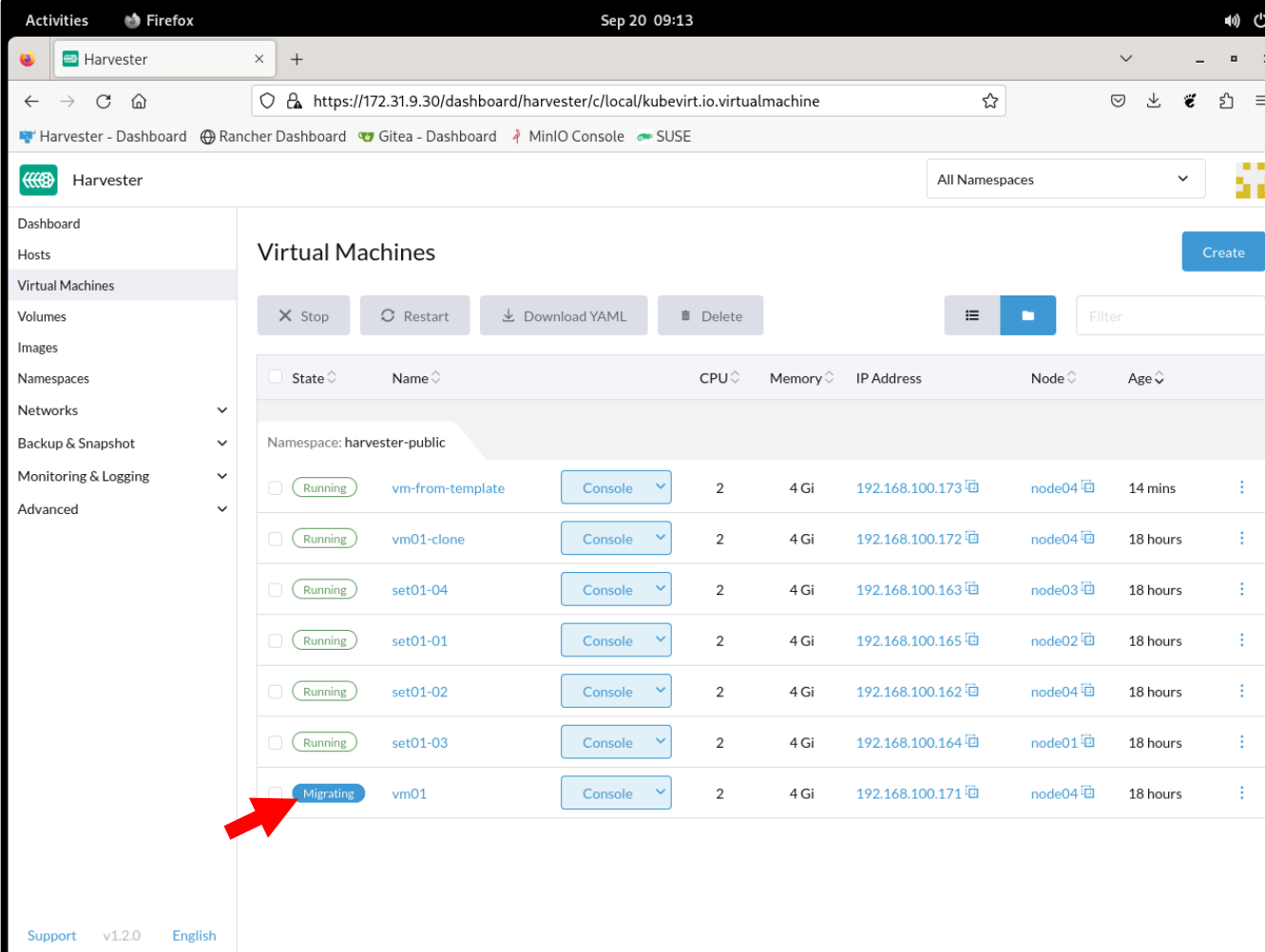
VM Live Migration

- 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**



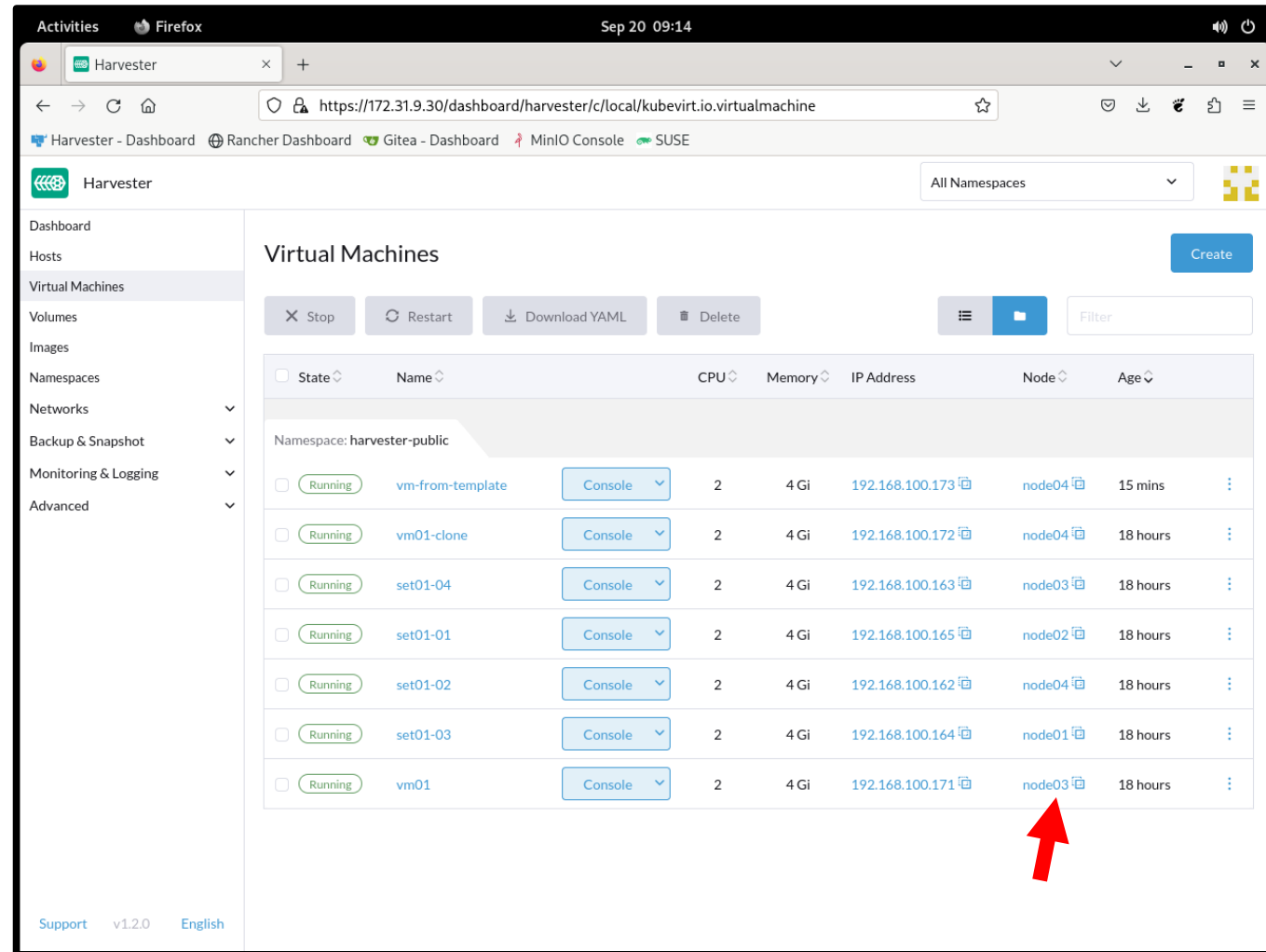
The screenshot shows the Harvester web interface in a Firefox browser window. The left sidebar contains navigation links: Dashboard, Hosts, Virtual Machines (selected), Volumes, Images, Namespaces, Networks, Backup & Snapshot, Monitoring & Logging, and Advanced. The main panel is titled 'Virtual Machines' and includes action buttons (Stop, Restart, Download YAML, Delete) and a 'Create' button. A table lists VMs in the 'harvester-public' namespace. The VM 'vm01' is highlighted with a red arrow pointing to its 'Migrating' status.

State	Name	CPU	Memory	IP Address	Node	Age
Running	vm-from-template	2	4 Gi	192.168.100.173	node04	14 mins
Running	vm01-clone	2	4 Gi	192.168.100.172	node04	18 hours
Running	set01-04	2	4 Gi	192.168.100.163	node03	18 hours
Running	set01-01	2	4 Gi	192.168.100.165	node02	18 hours
Running	set01-02	2	4 Gi	192.168.100.162	node04	18 hours
Running	set01-03	2	4 Gi	192.168.100.164	node01	18 hours
Migrating	vm01	2	4 Gi	192.168.100.171	node04	18 hours



VM Live Migration

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



The screenshot shows the Harvester web interface in a Firefox browser. The left sidebar contains navigation links: Dashboard, Hosts, Virtual Machines (selected), Volumes, Images, Namespaces, Networks, Backup & Snapshot, Monitoring & Logging, and Advanced. The main panel is titled 'Virtual Machines' and includes a 'Create' button and action buttons (Stop, Restart, Download YAML, Delete). A table lists VMs in the 'harvester-public' namespace. The 'Node' column shows the target node for each VM, with a red arrow pointing to 'node03' for the 'vm01' entry.

State	Name	CPU	Memory	IP Address	Node	Age
Running	vm-from-template	2	4 Gi	192.168.100.173	node04	15 mins
Running	vm01-clone	2	4 Gi	192.168.100.172	node04	18 hours
Running	set01-04	2	4 Gi	192.168.100.163	node03	18 hours
Running	set01-01	2	4 Gi	192.168.100.165	node02	18 hours
Running	set01-02	2	4 Gi	192.168.100.162	node04	18 hours
Running	set01-03	2	4 Gi	192.168.100.164	node01	18 hours
Running	vm01	2	4 Gi	192.168.100.171	node03	18 hours



Hot-plug Volumes



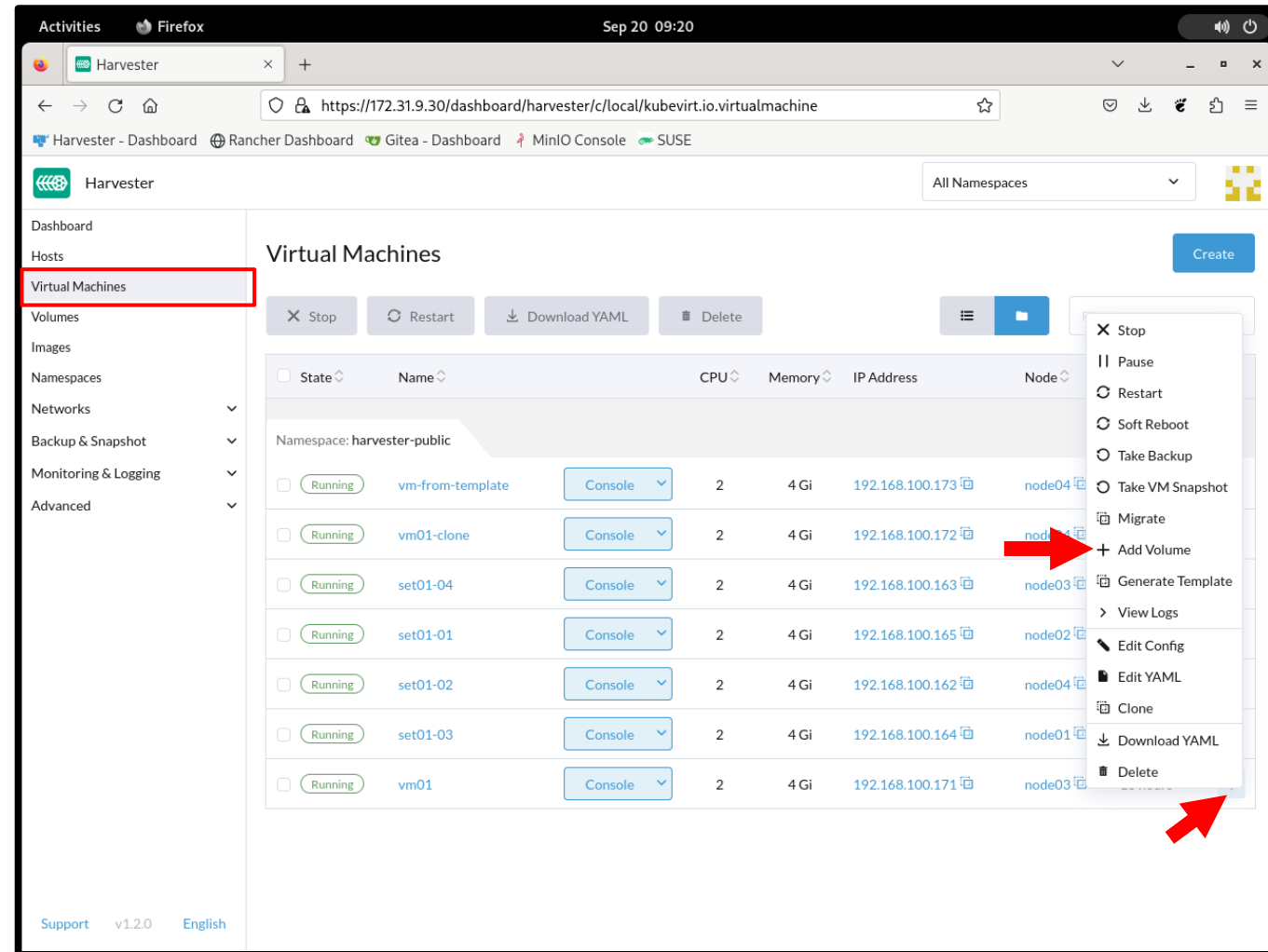
Hot-plug Volumes

- 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.



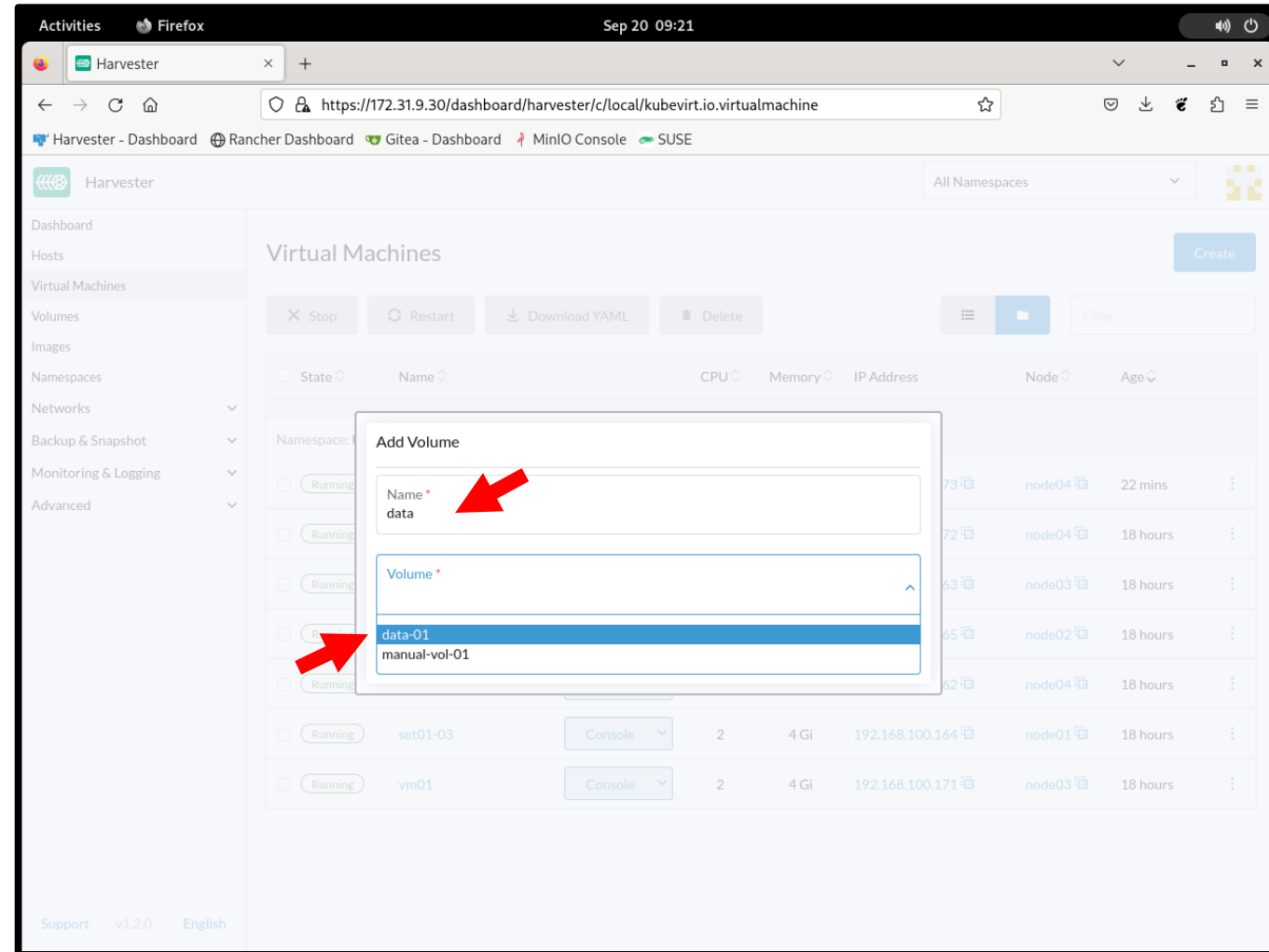
Hot-plug Volume

- Hot-plugging volumes into running VMs is initiated from the **Virtual Machines** screen.
- Click on the **3 dot** menu next to the VM you wish 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.

The screenshot shows the Harvester web interface in a Firefox browser window. The URL is <https://172.31.9.30/dashboard/harvester/c/local/kubevirt.io.virtualmachine/harvester-public/vm01>. The page title is "Virtual Machine: vm01" with a "Running" status. The left sidebar shows the navigation menu with "Volumes" selected. The main content area shows the "Volumes" tab for the VM. It lists two volumes:

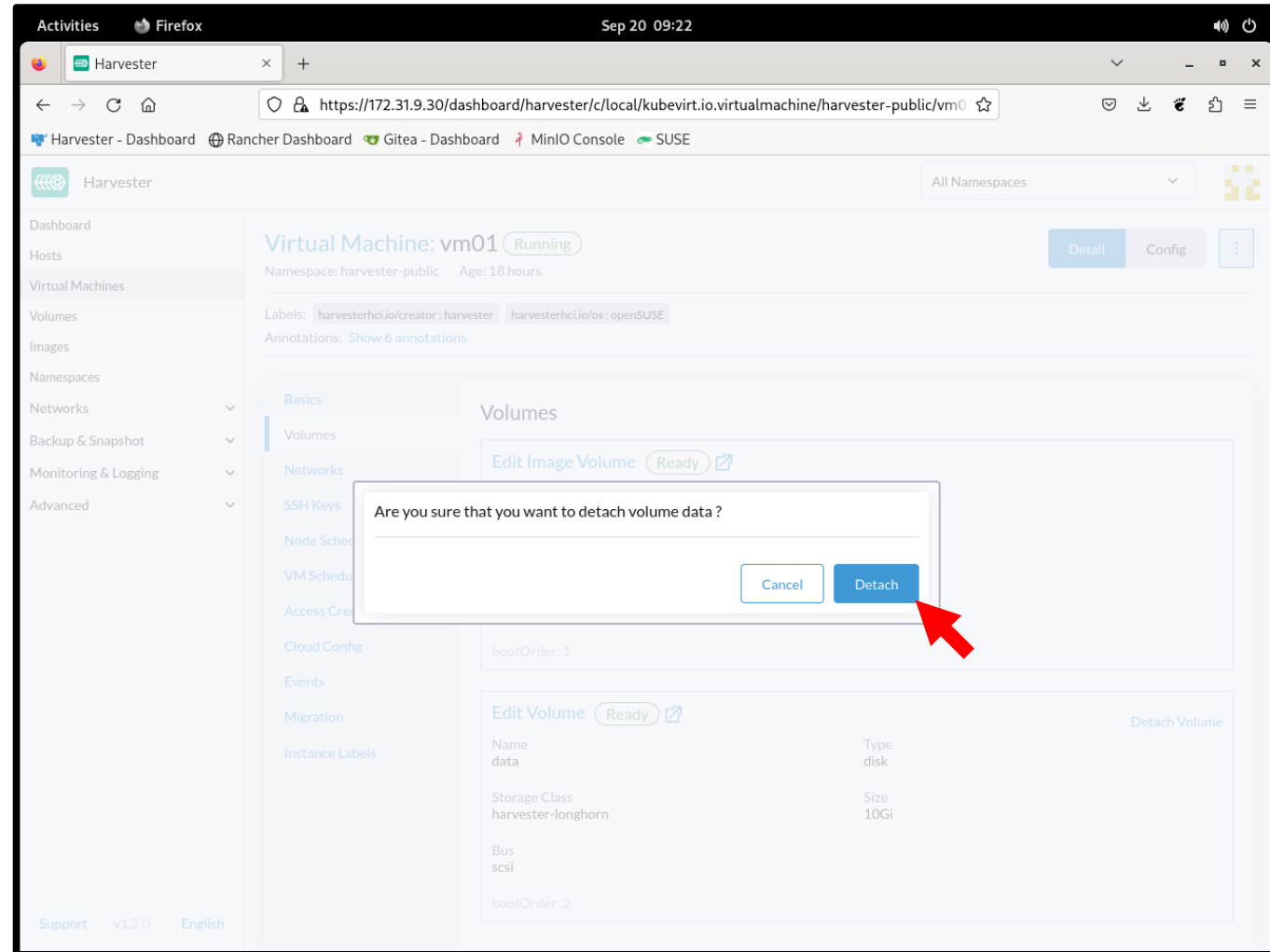
Name	Type	Image	Size	Bus	bootOrder
disk-0	disk	harvester-public/leap-15.5-nocloud	10Gi	virtio	1
data	disk	harvester-longhorn	10Gi	scsi	2

The "data" volume is highlighted with a red box, and a "Detach Volume" link is visible next to it.



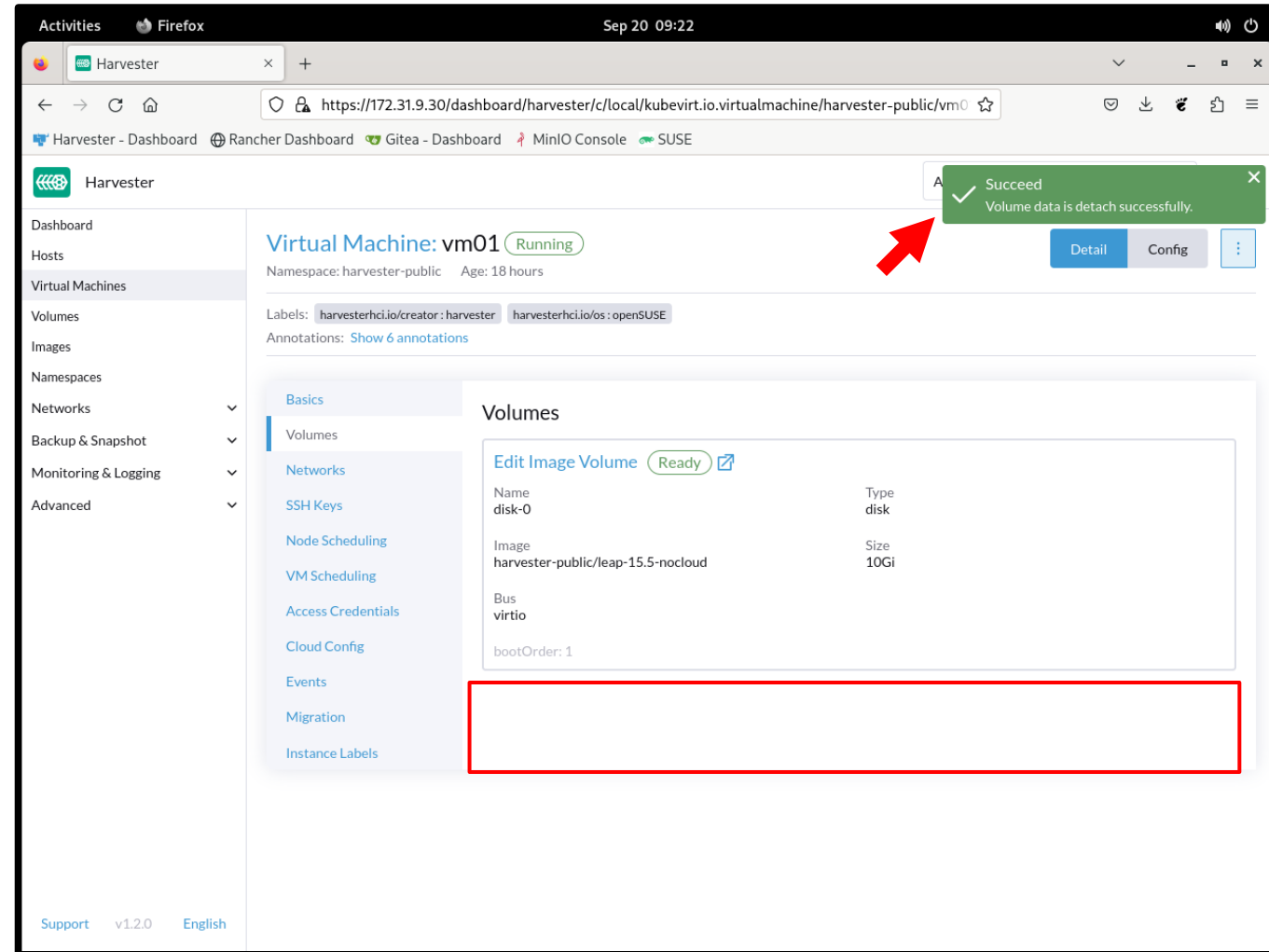
Hot-plug Volume: Detach Volume

- 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.

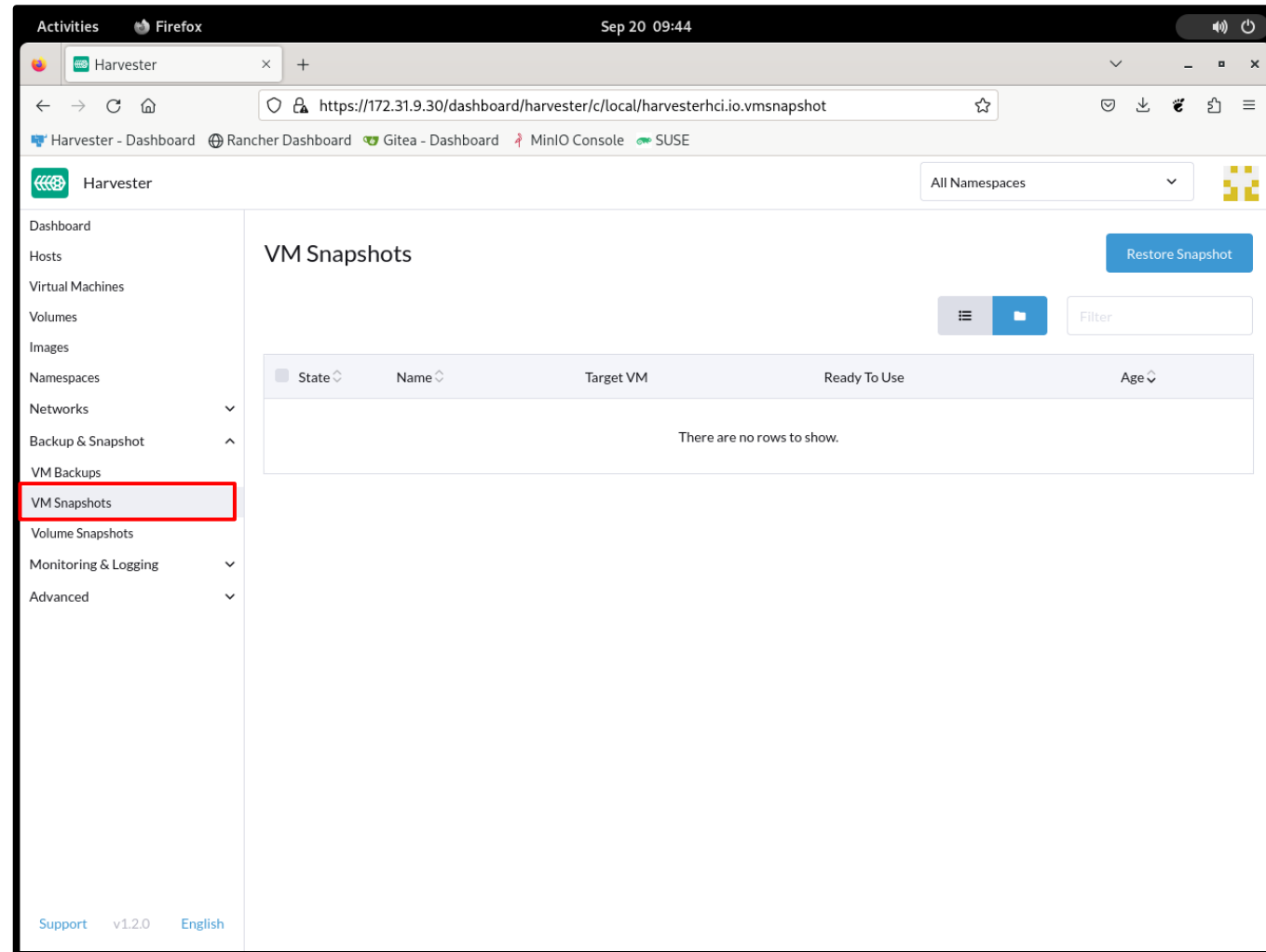
Backups

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



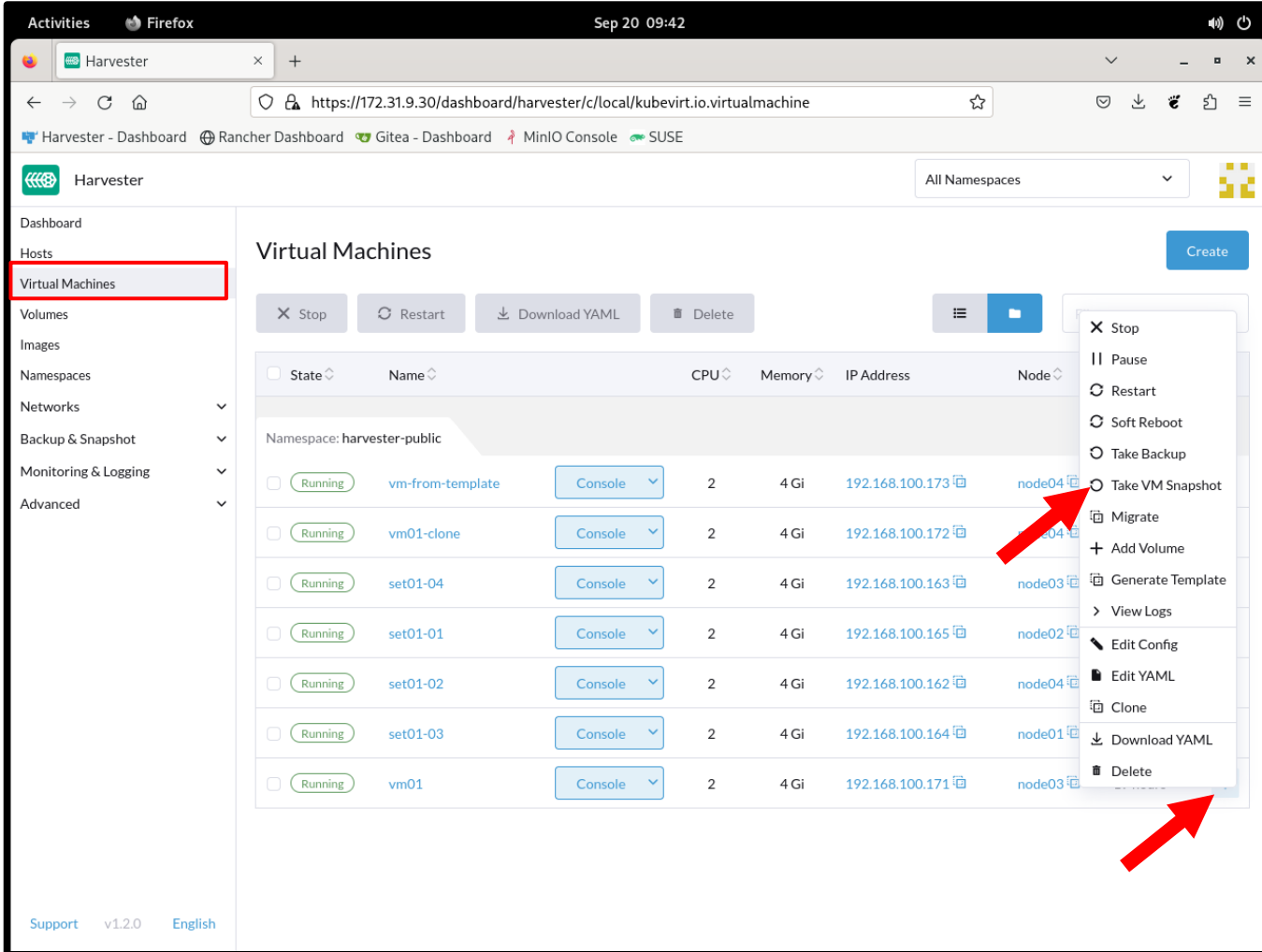
Snapshot a VM

- Virtual Machine snapshots 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**



The screenshot shows the Harvester web interface in a Firefox browser. The left sidebar has a red box around the 'Virtual Machines' link. The main panel displays a table of VMs in the 'harvester-public' namespace. A context menu is open for the 'vm01' VM, with a red arrow pointing to the 'Take VM Snapshot' option.

State	Name	CPU	Memory	IP Address	Node
Running	vm-from-template	2	4 Gi	192.168.100.173	node04
Running	vm01-clone	2	4 Gi	192.168.100.172	node04
Running	set01-04	2	4 Gi	192.168.100.163	node03
Running	set01-01	2	4 Gi	192.168.100.165	node02
Running	set01-02	2	4 Gi	192.168.100.162	node04
Running	set01-03	2	4 Gi	192.168.100.164	node01
Running	vm01	2	4 Gi	192.168.100.171	node03

Context menu options for 'vm01':

- Stop
- Pause
- Restart
- Soft Reboot
- Take Backup
- Take VM Snapshot**
- Migrate
- Add Volume
- Generate Template
- View Logs
- Edit Config
- Edit YAML
- Clone
- Download YAML
- Delete



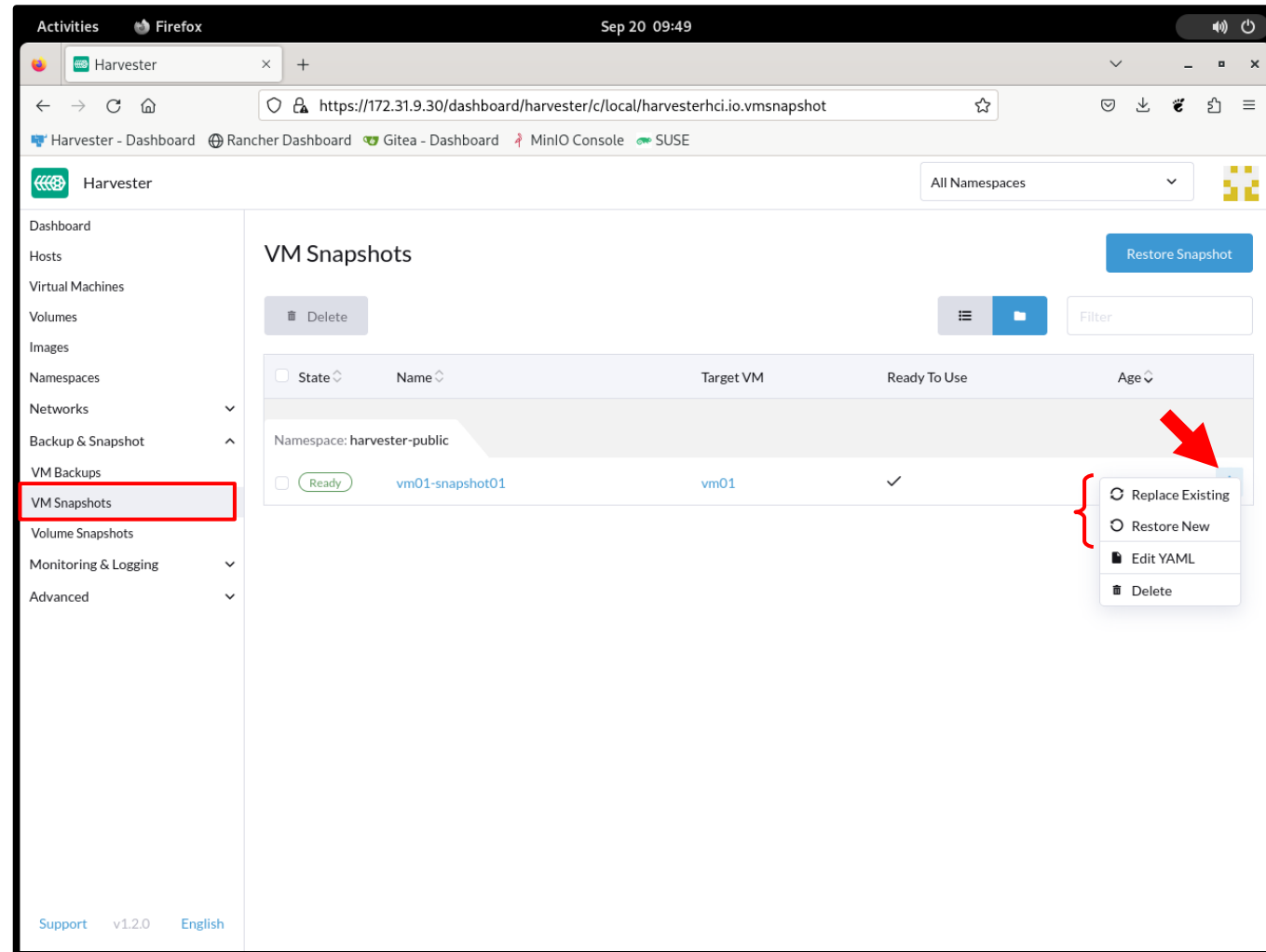
Restore VM Snapshots

- 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 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.

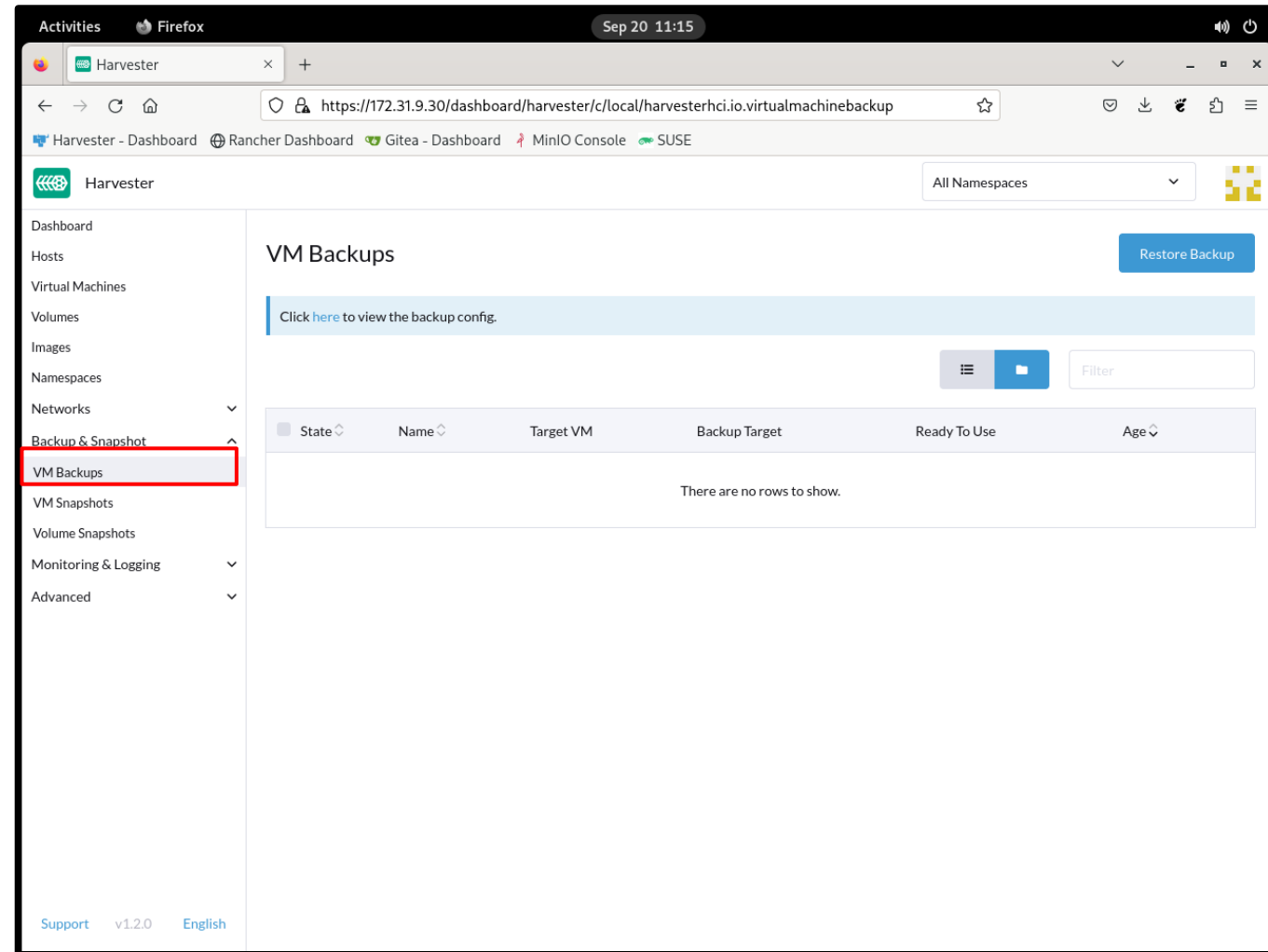
Backups

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



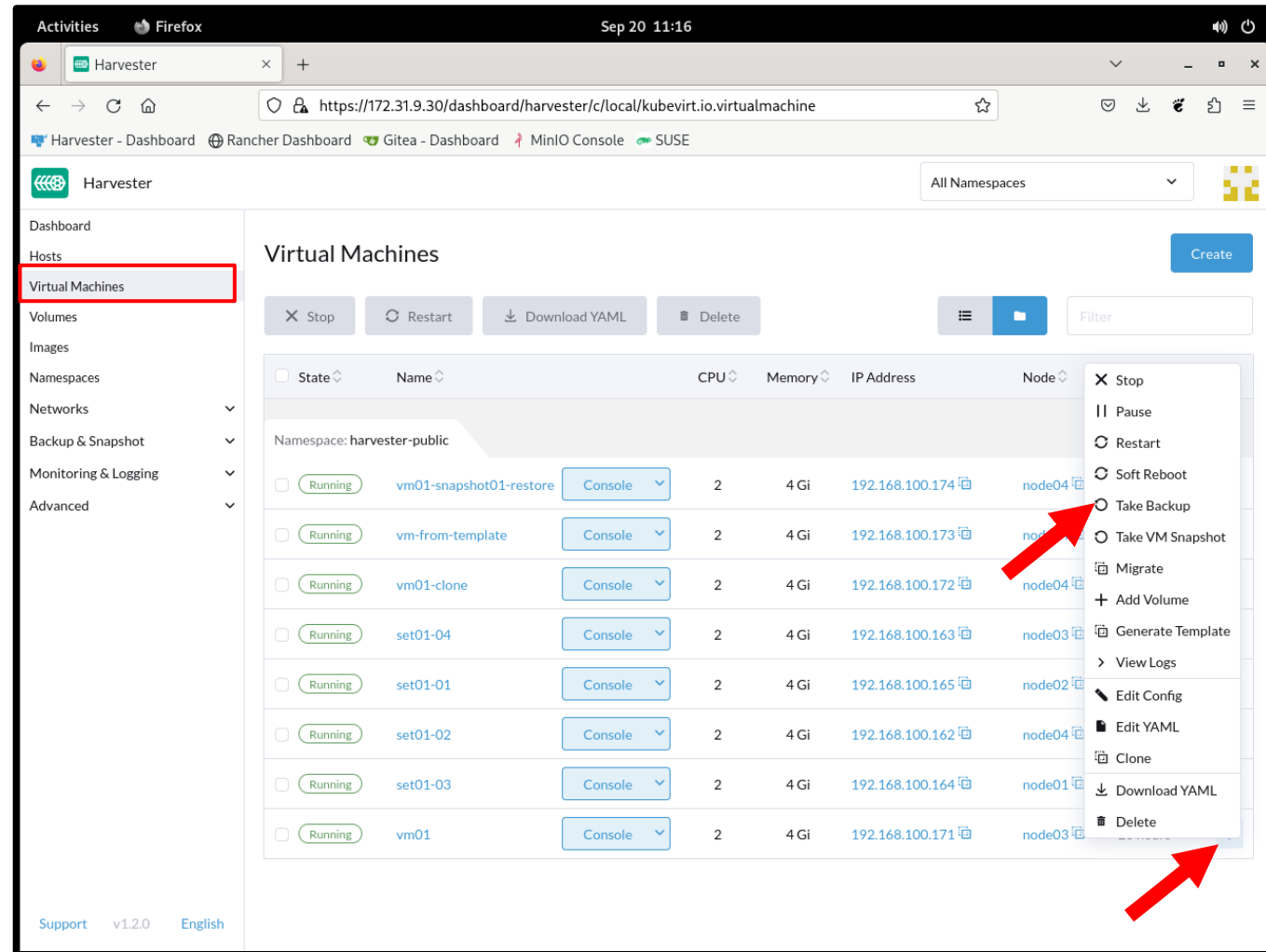
Backup VM

- 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**



The screenshot shows the Harvester web interface in a Firefox browser. The left sidebar has a red box around the 'Virtual Machines' link. The main area displays a table of VMs in the 'harvester-public' namespace. A context menu is open for the VM 'vm01-snapshot01-restore', with a red arrow pointing to the 'Take Backup' option. Another red arrow points to the 'Delete' option at the bottom of the menu.

State	Name	CPU	Memory	IP Address	Node
Running	vm01-snapshot01-restore	2	4 Gi	192.168.100.174	node04
Running	vm-from-template	2	4 Gi	192.168.100.173	node04
Running	vm01-clone	2	4 Gi	192.168.100.172	node04
Running	set01-04	2	4 Gi	192.168.100.163	node03
Running	set01-01	2	4 Gi	192.168.100.165	node02
Running	set01-02	2	4 Gi	192.168.100.162	node04
Running	set01-03	2	4 Gi	192.168.100.164	node01
Running	vm01	2	4 Gi	192.168.100.171	node03



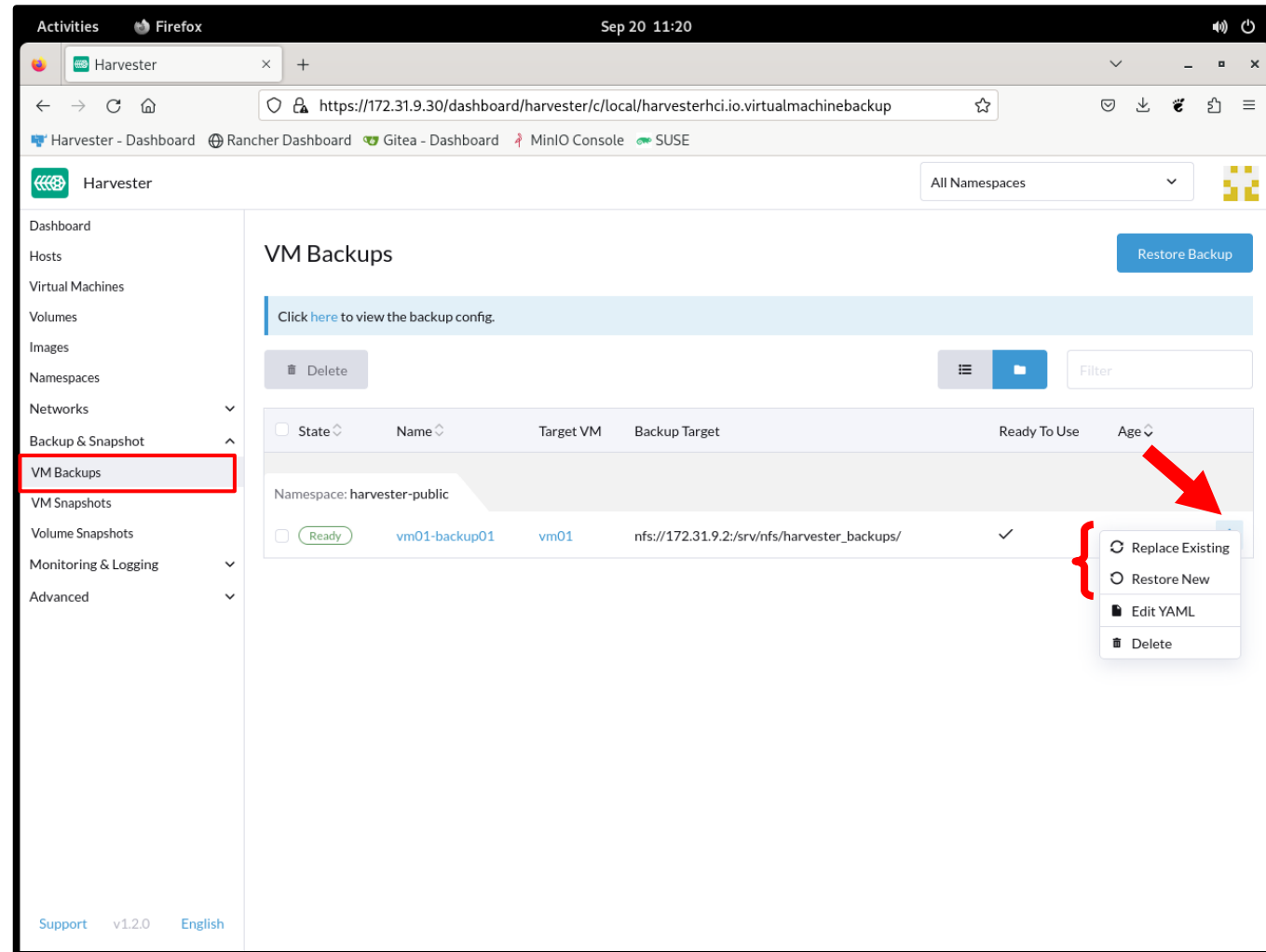
Restore VM Backups

- 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

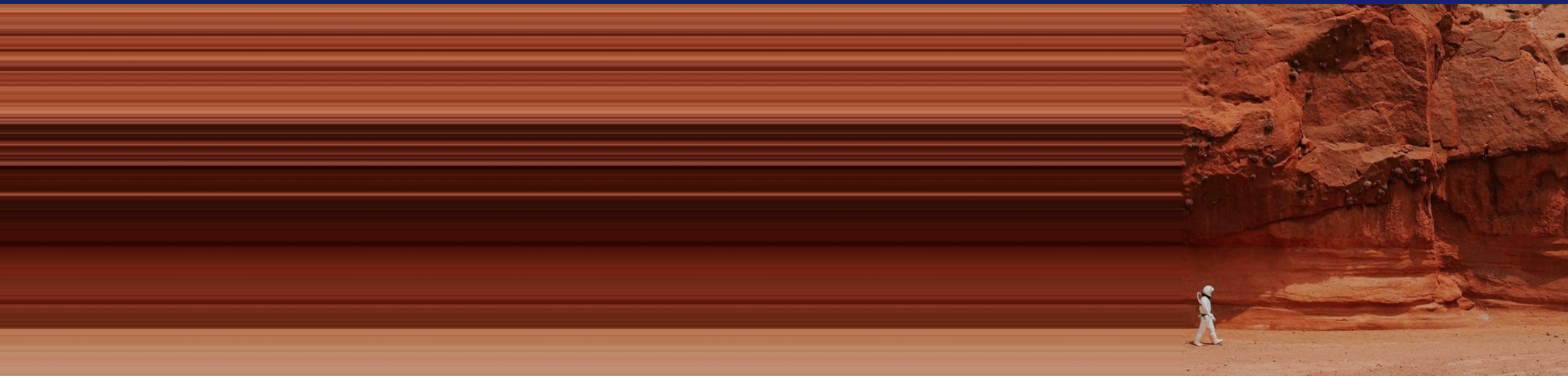


- Backup and Restore Virtual Machines
 - Snapshots of VMs can be created with the data in the snapshots being stored in the local Longhorn storage cluster.
 - Snapshots of VMs can be restored as new VMs or by replacing the original instance the snapshot was created from.
 - Backups of VMs can be created 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.



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**

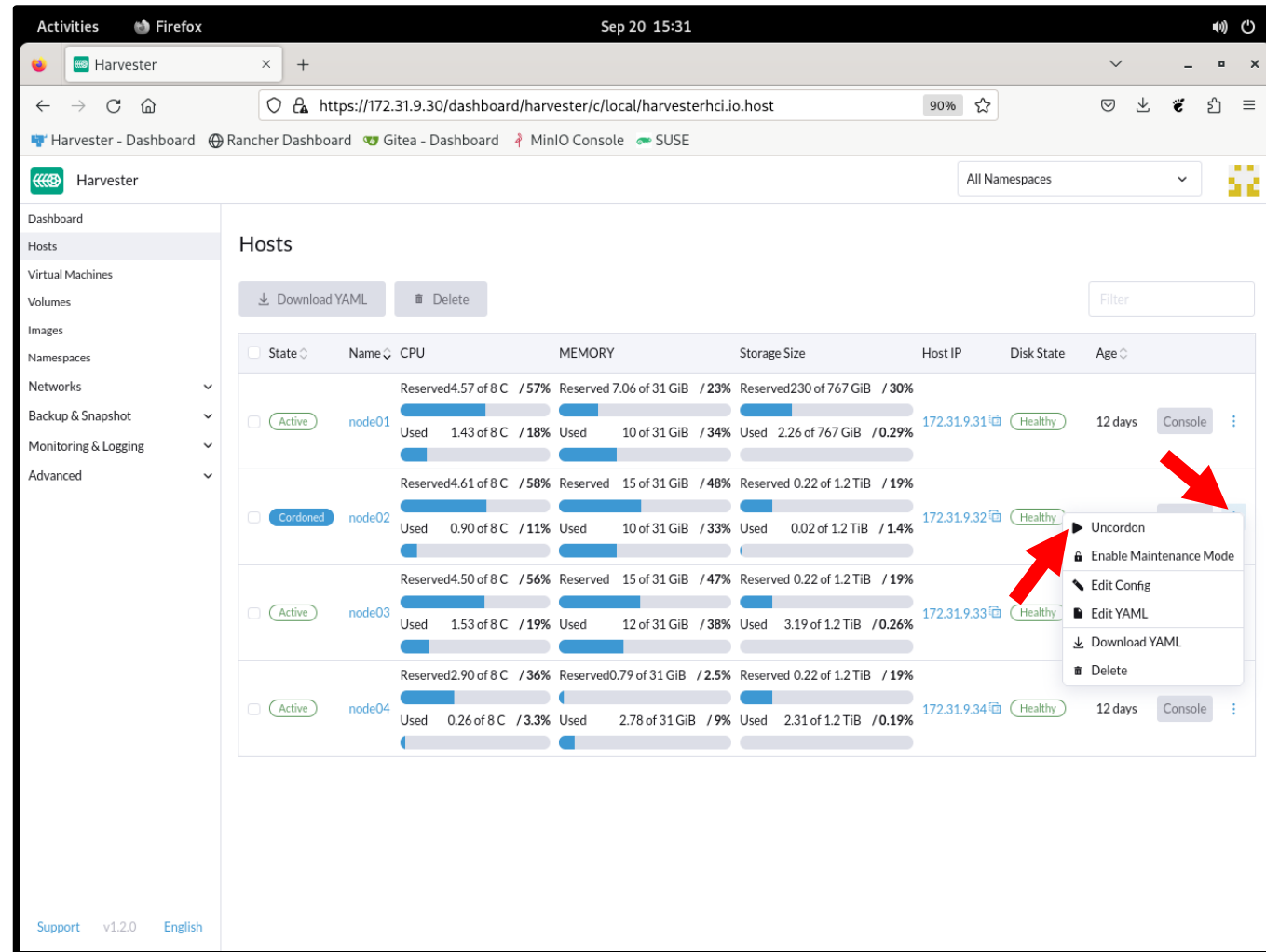
The screenshot shows the Harvester web interface in a Firefox browser. The left sidebar has a red box around the 'Hosts' tab. The main area displays a table of four hosts: node01, node02, node03, and node04. Each host row has a three-dot menu on the right. A red arrow points to the three-dot menu for node02, and another red arrow points to the 'Cordon' option in the dropdown menu that appears.

State	Name	CPU	MEMORY	Storage Size	Host IP	Disk State	Age	Actions
Active	node01	Reserved 4.57 of 8 C / 57% Used 1.24 of 8 C / 15%	Reserved 7.06 of 31 GiB / 23% Used 10 of 31 GiB / 33%	Reserved 230 of 767 GiB / 30% Used 2.26 of 767 GiB / 0.29%	172.31.9.31	Healthy	12 days	Console ⋮
Active	node02	Reserved 4 of 8 C / 50% Used 0.67 of 8 C / 8.4%	Reserved 8.56 of 31 GiB / 28% Used 8.22 of 31 GiB / 27%	Reserved 0.22 of 1.2 TiB / 19% Used 0.02 of 1.2 TiB / 1.4%	172.31.9.32	Healthy		⋮
Active	node03	Reserved 4.25 of 8 C / 53% Used 1.43 of 8 C / 18%	Reserved 8.95 of 31 GiB / 29% Used 9.82 of 31 GiB / 32%	Reserved 0.22 of 1.2 TiB / 19% Used 3.19 of 1.2 TiB / 0.26%	172.31.9.33	Healthy		⋮
Active	node04	Reserved 3.76 of 8 C / 47% Used 0.65 of 8 C / 8.1%	Reserved 13 of 31 GiB / 41% Used 8.99 of 31 GiB / 29%	Reserved 0.22 of 1.2 TiB / 19% Used 2.31 of 1.2 TiB / 0.19%	172.31.9.34	Healthy	12 days	Console ⋮



Node Cordoning

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



The screenshot shows the Harvester dashboard in a Firefox browser window. The left sidebar contains navigation links: Dashboard, Hosts, Virtual Machines, Volumes, Images, Namespaces, Networks, Backup & Snapshot, Monitoring & Logging, and Advanced. The main content area is titled 'Hosts' and displays a table of four hosts. The 'node02' host is highlighted with a blue 'Cordoned' status badge. A context menu is open for 'node02', showing options: Uncordon, Enable Maintenance Mode, Edit Config, Edit YAML, Download YAML, and Delete. A red arrow points to the 'Uncordon' option. The table columns are State, Name, CPU, MEMORY, Storage Size, Host IP, Disk State, and Age.

State	Name	CPU	MEMORY	Storage Size	Host IP	Disk State	Age
Active	node01	Reserved 4.57 of 8 C / 57% Used 1.43 of 8 C / 18%	Reserved 7.06 of 31 GiB / 23% Used 10 of 31 GiB / 34%	Reserved 230 of 767 GiB / 30% Used 2.26 of 767 GiB / 0.29%	172.31.9.31	Healthy	12 days
Cordoned	node02	Reserved 4.61 of 8 C / 58% Used 0.90 of 8 C / 11%	Reserved 15 of 31 GiB / 48% Used 10 of 31 GiB / 33%	Reserved 0.22 of 1.2 TiB / 19% Used 0.02 of 1.2 TiB / 1.4%	172.31.9.32	Healthy	
Active	node03	Reserved 4.50 of 8 C / 56% Used 1.53 of 8 C / 19%	Reserved 15 of 31 GiB / 47% Used 12 of 31 GiB / 38%	Reserved 0.22 of 1.2 TiB / 19% Used 3.19 of 1.2 TiB / 0.26%	172.31.9.33	Healthy	
Active	node04	Reserved 2.90 of 8 C / 36% Used 0.26 of 8 C / 3.3%	Reserved 0.79 of 31 GiB / 2.5% Used 2.78 of 31 GiB / 9%	Reserved 0.22 of 1.2 TiB / 19% Used 2.31 of 1.2 TiB / 0.19%	172.31.9.34	Healthy	12 days



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**

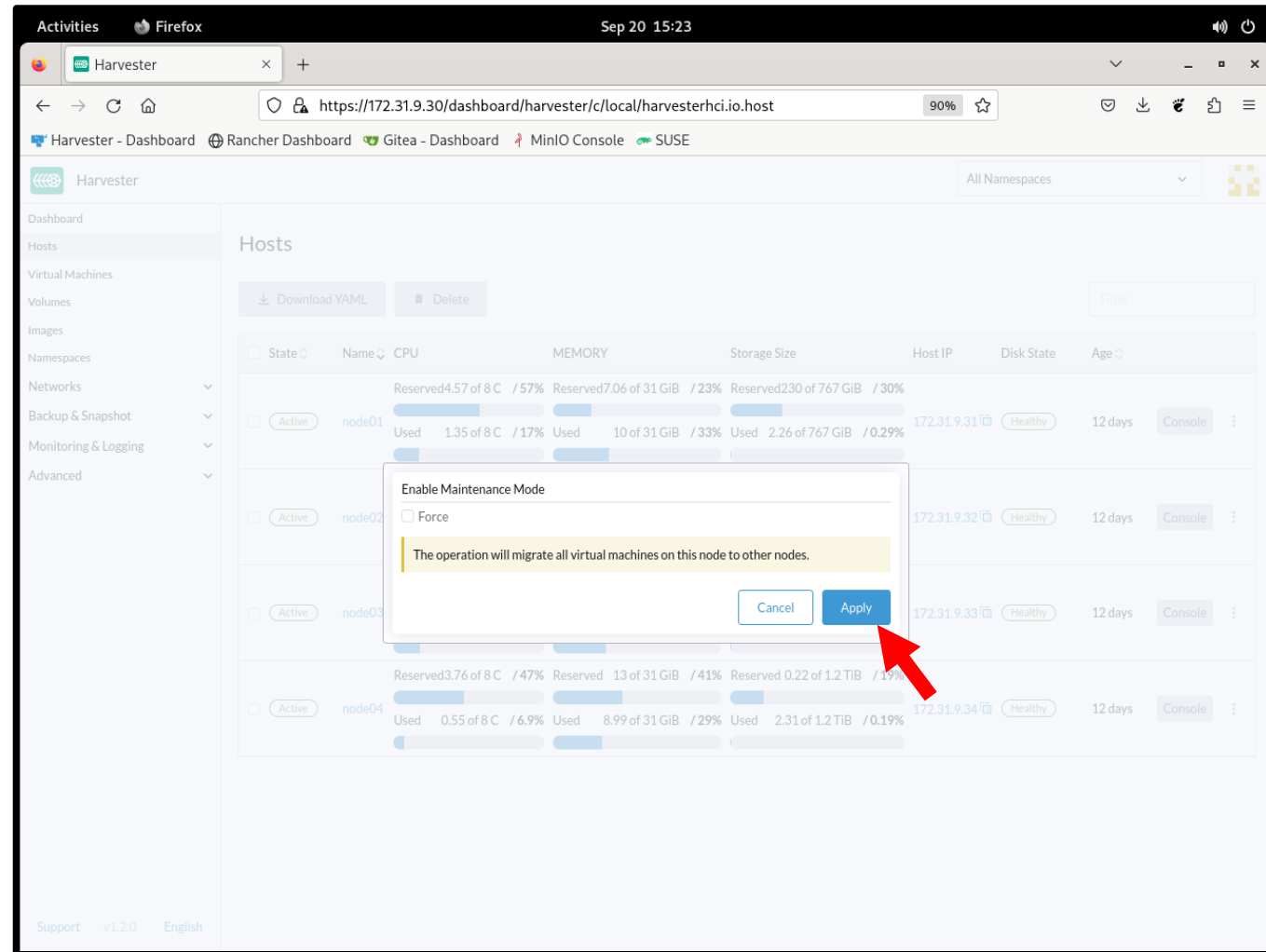
The screenshot shows the Harvester dashboard in a Firefox browser window. The left sidebar has a red box around the 'Hosts' tab. The main content area shows a table of hosts. A context menu is open for 'node03', and a red arrow points to the 'Enable Maintenance Mode' option. Another red arrow points to the '3 dot' menu icon for 'node03'.

State	Name	CPU	MEMORY	Storage Size	Host IP	Disk State	Age	Actions
Active	node01	Reserved 4.57 of 8 C / 57% Used 1.24 of 8 C / 15%	Reserved 7.06 of 31 GiB / 23% Used 10 of 31 GiB / 33%	Reserved 230 of 767 GiB / 30% Used 2.26 of 767 GiB / 0.29%	172.31.9.31	Healthy	12 days	Console
Active	node02	Reserved 4 of 8 C / 50% Used 0.67 of 8 C / 8.4%	Reserved 8.56 of 31 GiB / 28% Used 8.22 of 31 GiB / 27%	Reserved 0.22 of 1.2 TiB / 19% Used 0.02 of 1.2 TiB / 1.4%	172.31.9.32	Healthy		
Active	node03	Reserved 4.25 of 8 C / 53% Used 1.43 of 8 C / 18%	Reserved 8.95 of 31 GiB / 29% Used 9.82 of 31 GiB / 32%	Reserved 0.22 of 1.2 TiB / 19% Used 3.19 of 1.2 TiB / 0.26%	172.31.9.33	Healthy		
Active	node04	Reserved 3.76 of 8 C / 47% Used 0.65 of 8 C / 8.1%	Reserved 13 of 31 GiB / 41% Used 8.99 of 31 GiB / 29%	Reserved 0.22 of 1.2 TiB / 19% Used 2.31 of 1.2 TiB / 0.19%	172.31.9.34	Healthy	12 days	Console



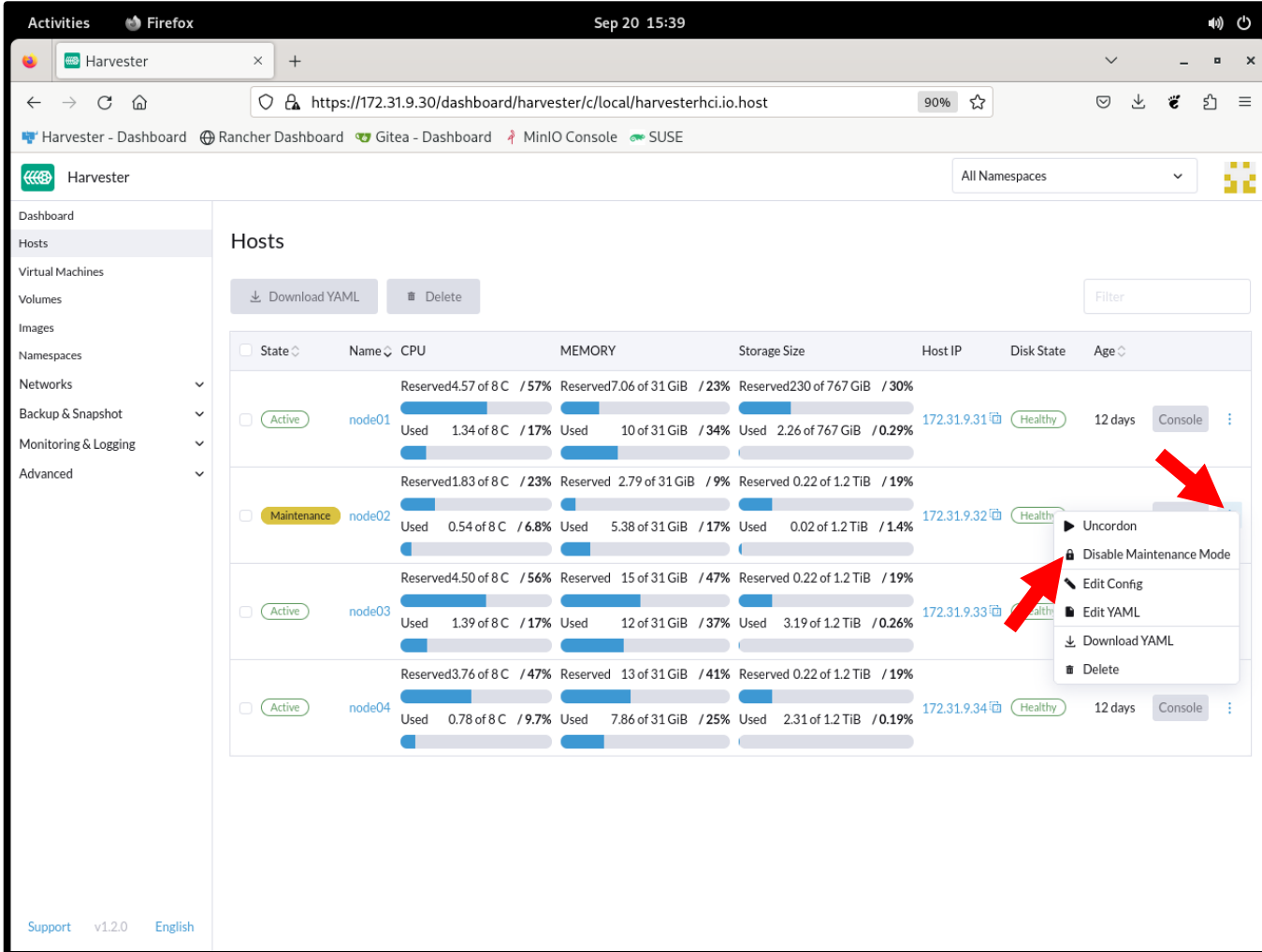
Maintenance Mode

- 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**



The screenshot shows the Harvester dashboard in a Firefox browser window. The 'Hosts' section is active, displaying a table of four nodes: node01, node02, node03, and node04. Node02 is currently in 'Maintenance' mode, indicated by a yellow label. A context menu is open for node02, showing options: 'Uncordon', 'Disable Maintenance Mode', 'Edit Config', 'Edit YAML', 'Download YAML', and 'Delete'. A red arrow points to the 'Disable Maintenance Mode' option. Another red arrow points to the three-dot menu icon on the right of the node02 row. The dashboard includes a sidebar with navigation links like Dashboard, Hosts, Virtual Machines, Volumes, Images, Namespaces, Networks, Backup & Snapshot, Monitoring & Logging, and Advanced. The top of the dashboard shows the Harvester logo and a 'All Namespaces' dropdown.

State	Name	CPU	MEMORY	Storage Size	Host IP	Disk State	Age	Actions
Active	node01	Reserved 4.57 of 8 C / 57% Used 1.34 of 8 C / 17%	Reserved 7.06 of 31 GiB / 23% Used 10 of 31 GiB / 34%	Reserved 230 of 767 GiB / 30% Used 2.26 of 767 GiB / 0.29%	172.31.9.31	Healthy	12 days	Console
Maintenance	node02	Reserved 1.83 of 8 C / 23% Used 0.54 of 8 C / 6.8%	Reserved 2.79 of 31 GiB / 9% Used 5.38 of 31 GiB / 17%	Reserved 0.22 of 1.2 TiB / 19% Used 0.02 of 1.2 TiB / 1.4%	172.31.9.32	Healthy		⋮
Active	node03	Reserved 4.50 of 8 C / 56% Used 1.39 of 8 C / 17%	Reserved 15 of 31 GiB / 47% Used 12 of 31 GiB / 37%	Reserved 0.22 of 1.2 TiB / 19% Used 3.19 of 1.2 TiB / 0.26%	172.31.9.33	Healthy		⋮
Active	node04	Reserved 3.76 of 8 C / 47% Used 0.78 of 8 C / 9.7%	Reserved 13 of 31 GiB / 41% Used 7.86 of 31 GiB / 25%	Reserved 0.22 of 1.2 TiB / 19% Used 2.31 of 1.2 TiB / 0.19%	172.31.9.34	Healthy	12 days	Console



Additional Troubleshooting Tools in Harvester



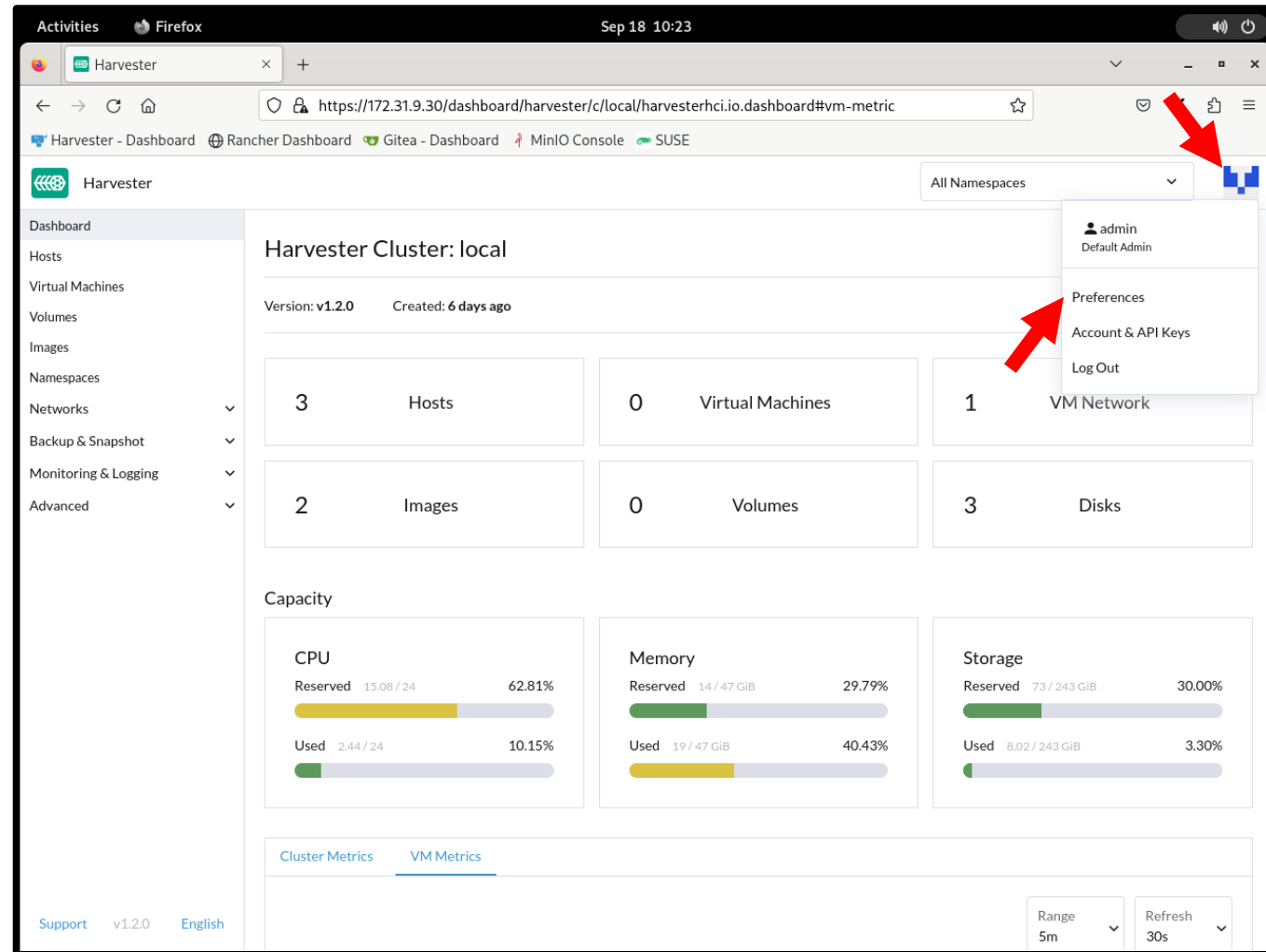
Additional Troubleshooting Tools

- 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.



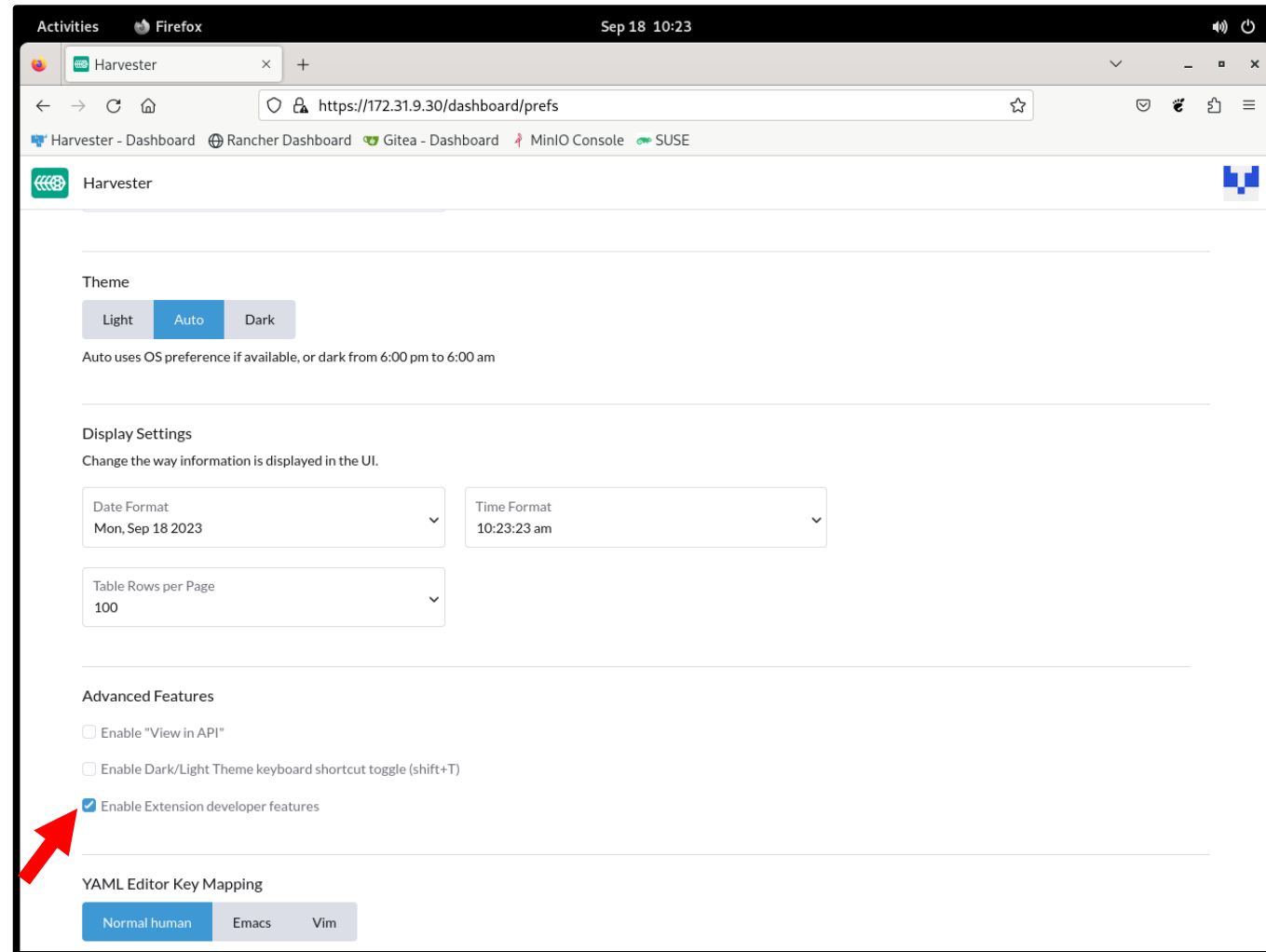
Enable Additional Troubleshooting Tools

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



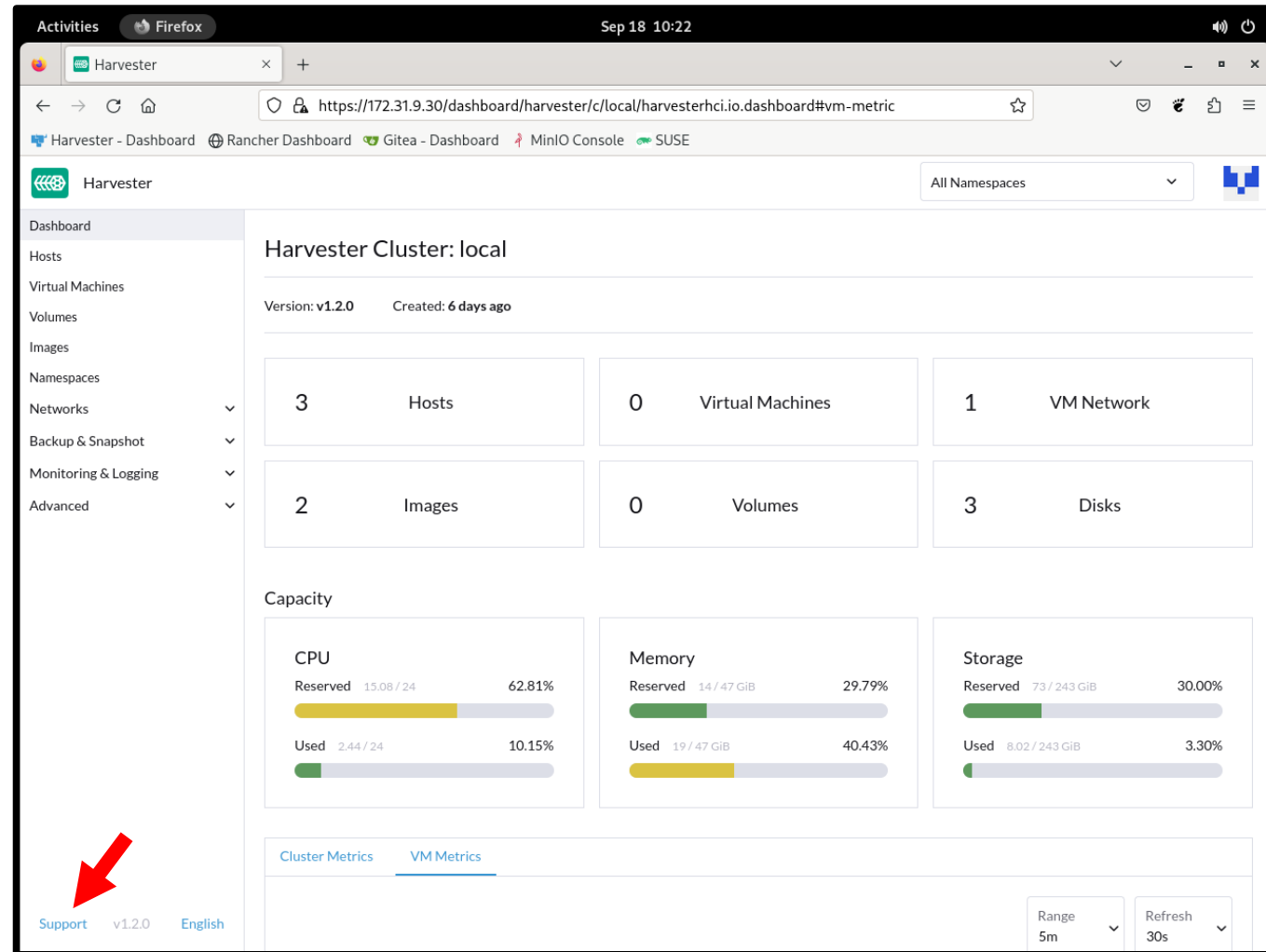
Enable Additional Troubleshooting Tools

- 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**.



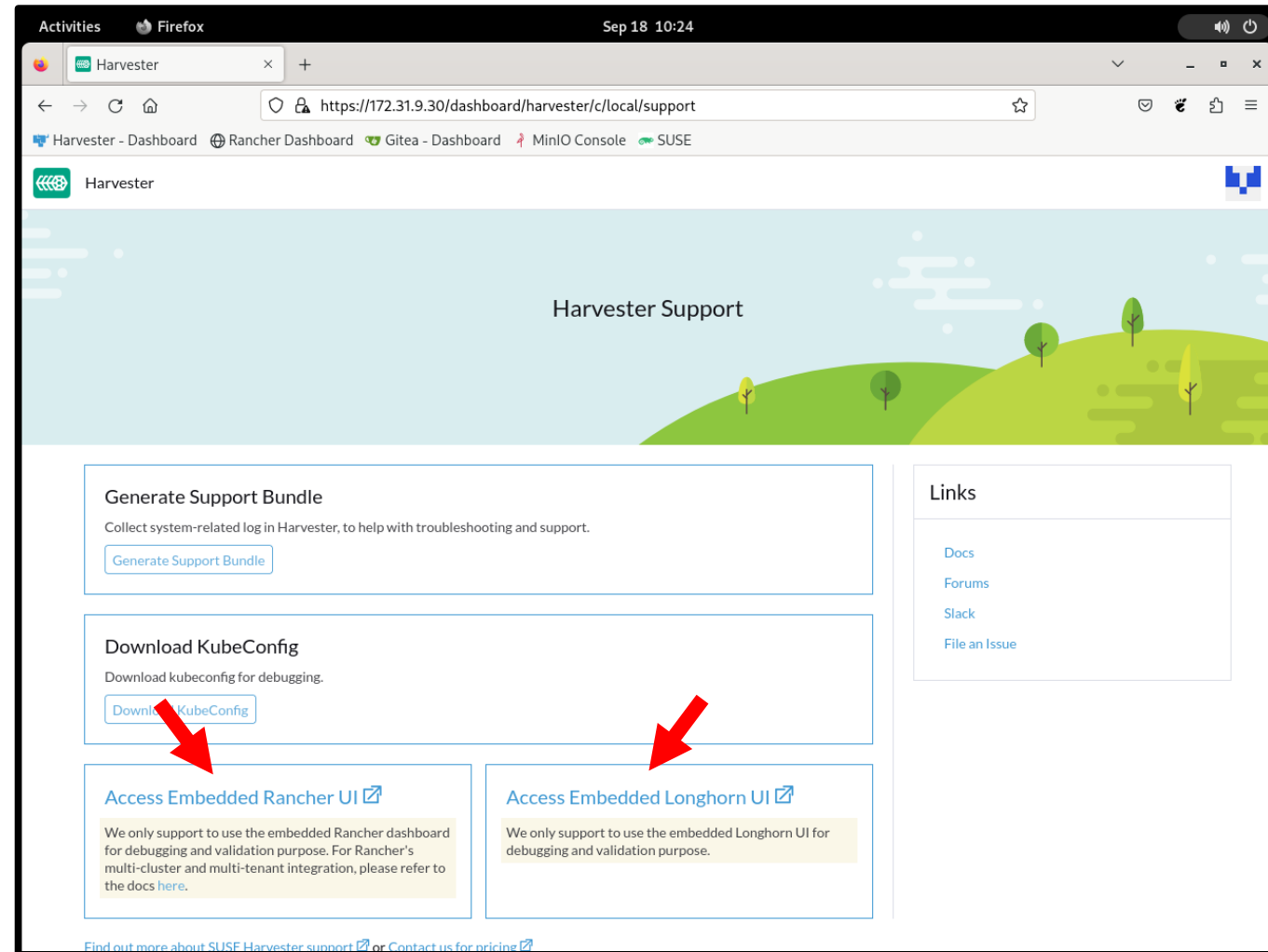
Enable Additional Troubleshooting Tools

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



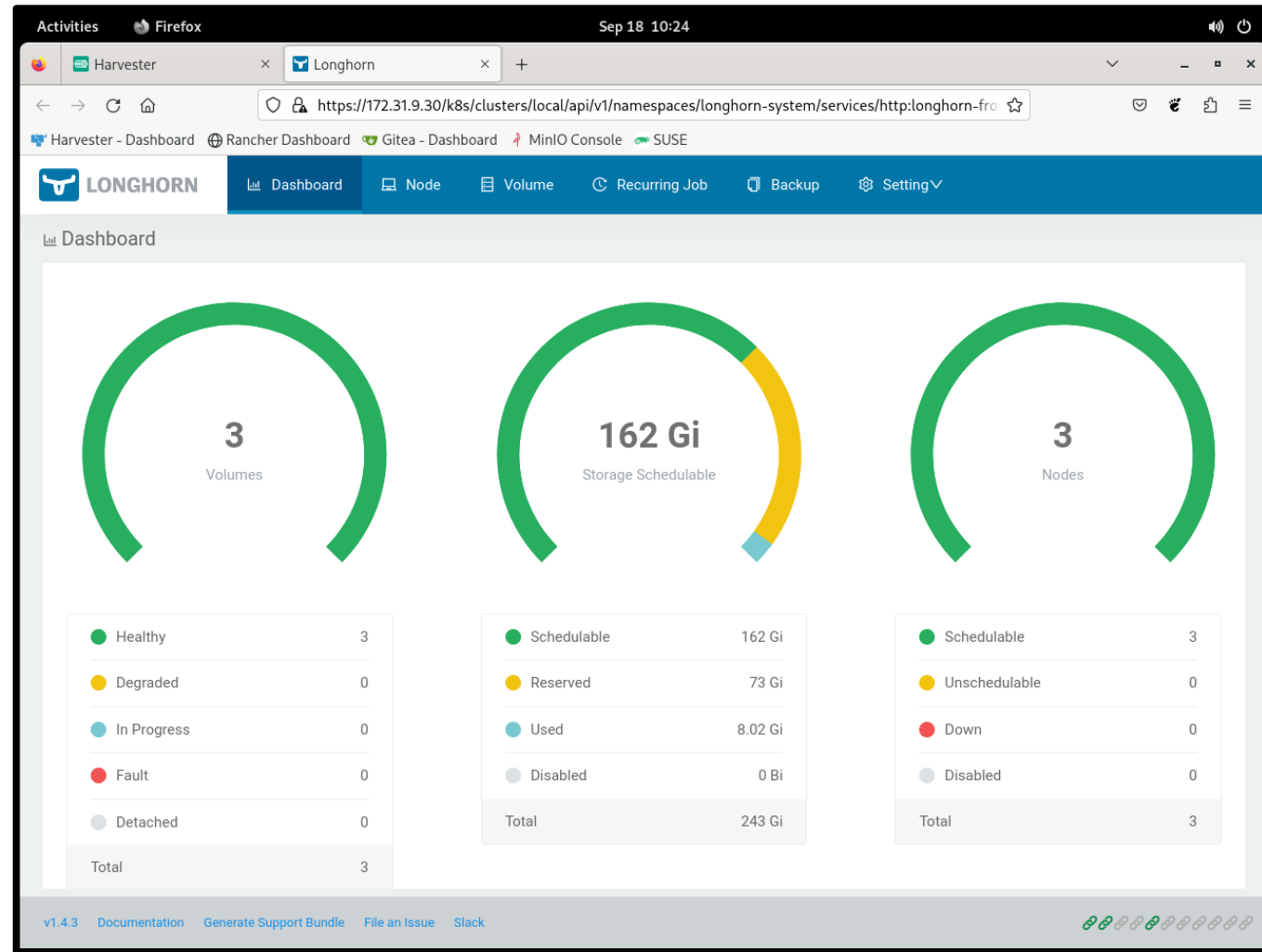
Embedded Longhorn Dashboard

- 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.

The screenshot shows a web browser window displaying the Embedded Rancher Dashboard for a Harvester cluster. The dashboard is titled "Cluster Dashboard" and provides a summary of the cluster's status and resources.

Cluster Summary:

- Provider: Harvester
- Kubernetes Version: —
- Created: 6 days ago
- 1558 Total Resources
- 3 Nodes
- 36 Deployments

Cluster Health:

- Etcd: ✓
- Scheduler: ✓
- Controller Manager: ✓

Events:

Reason	Object	Message	Name	Date
SnapshotError	Snapshot efb2c0bd-cc30-455b-8d82-594436b557ff	lost track of the corresponding snapshot info inside volume engine	efb2c0bd-cc30-455b-8d82-594436b557ff.1786086072661713	Mon, Sep 18 2023 9:38:40 am
SnapshotDelete	Snapshot 45fd9c8a-14ee-4d26-82e0-ca65d0f14686	snapshot is marked as removed	45fd9c8a-14ee-4d26-82e0-ca65d0f14686.1786085f56514c59	Mon, Sep 18 2023 9:38:35 am
SnapshotUpdate	Snapshot 45fd9c8a-14ee-4d26-82e0-ca65d0f14686	snapshot becomes not ready to use	45fd9c8a-14ee-4d26-82e0-ca65d0f14686.1786085f56517333	Mon, Sep 18 2023 9:38:35 am
	Engine pvc-ae1b1c2-	Detected replica pvc-ae1b1c2-	pvc-ae1b1c2-ch7f-4662-84a5-	



Support Bundles in Harvester



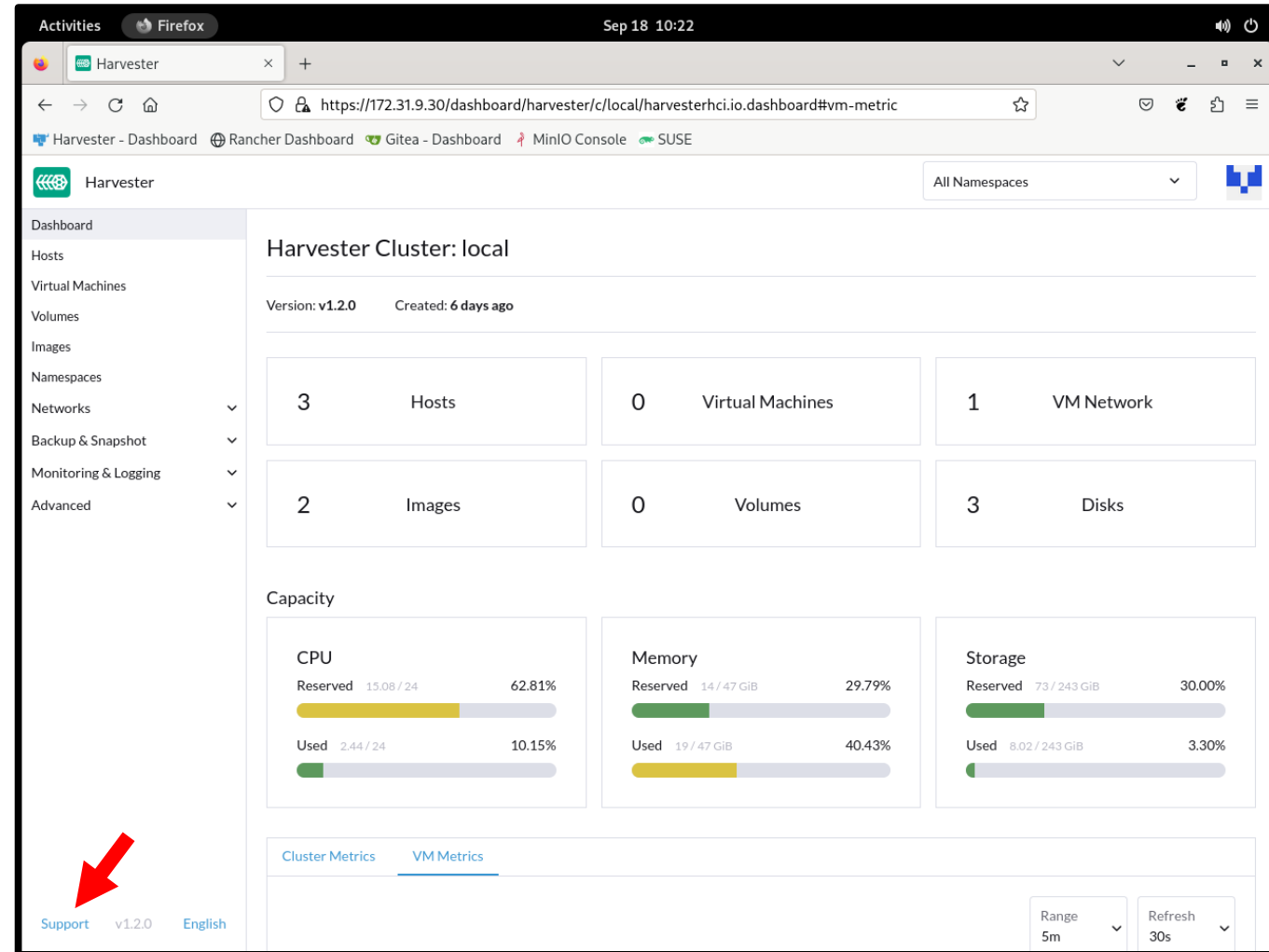
Harvester Support Bundles

- 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.



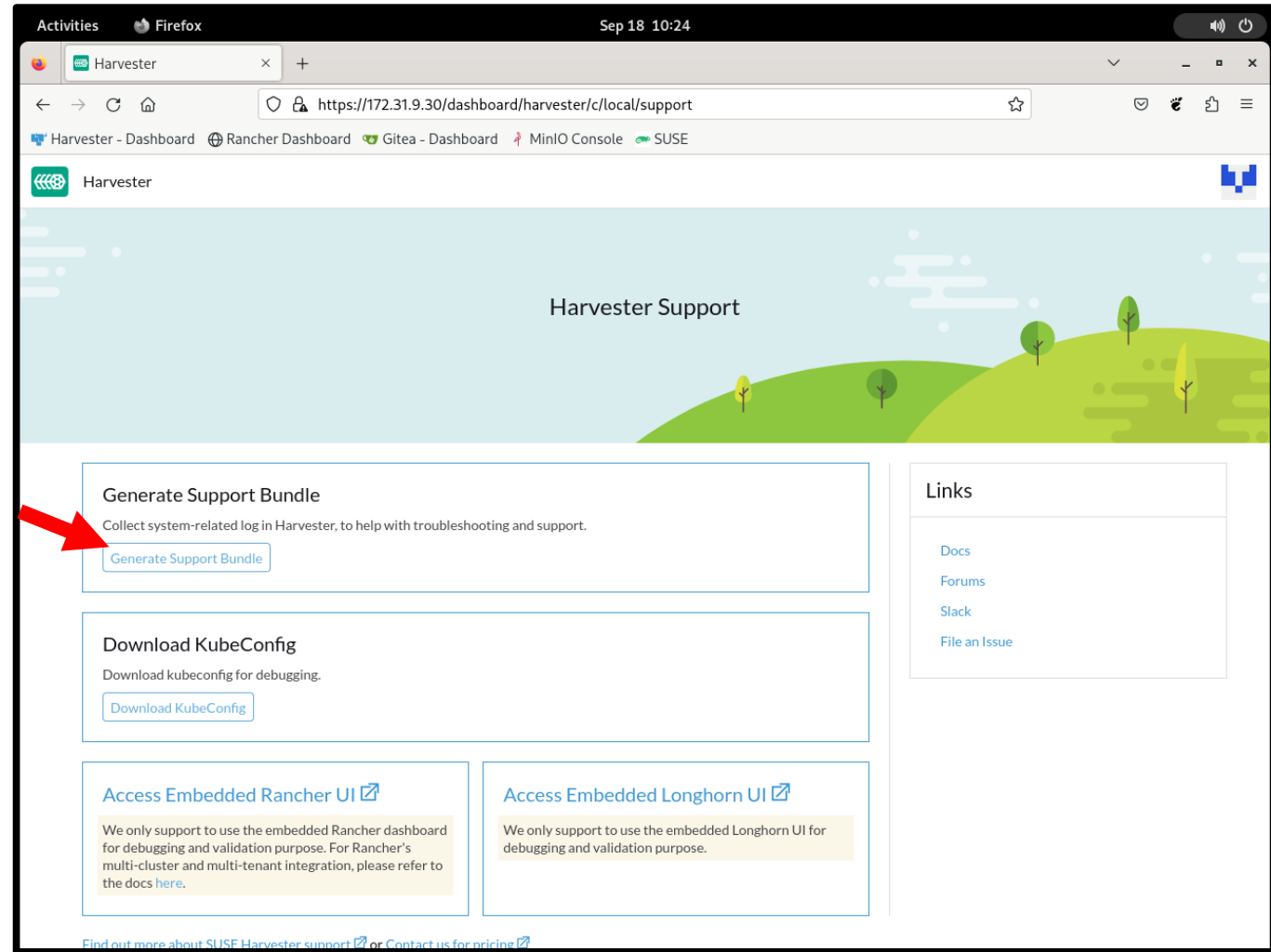
Support Bundle

- 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**.

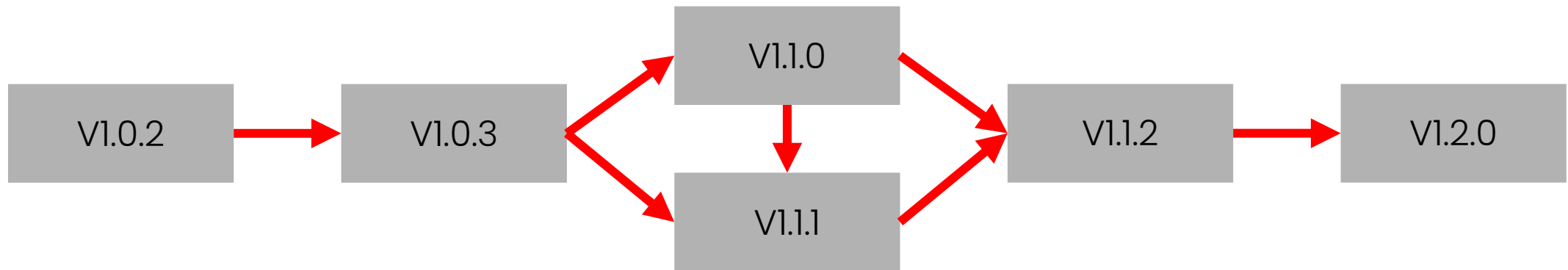


Harvester Cluster Upgrade



Harvester Cluster Upgrade Path

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

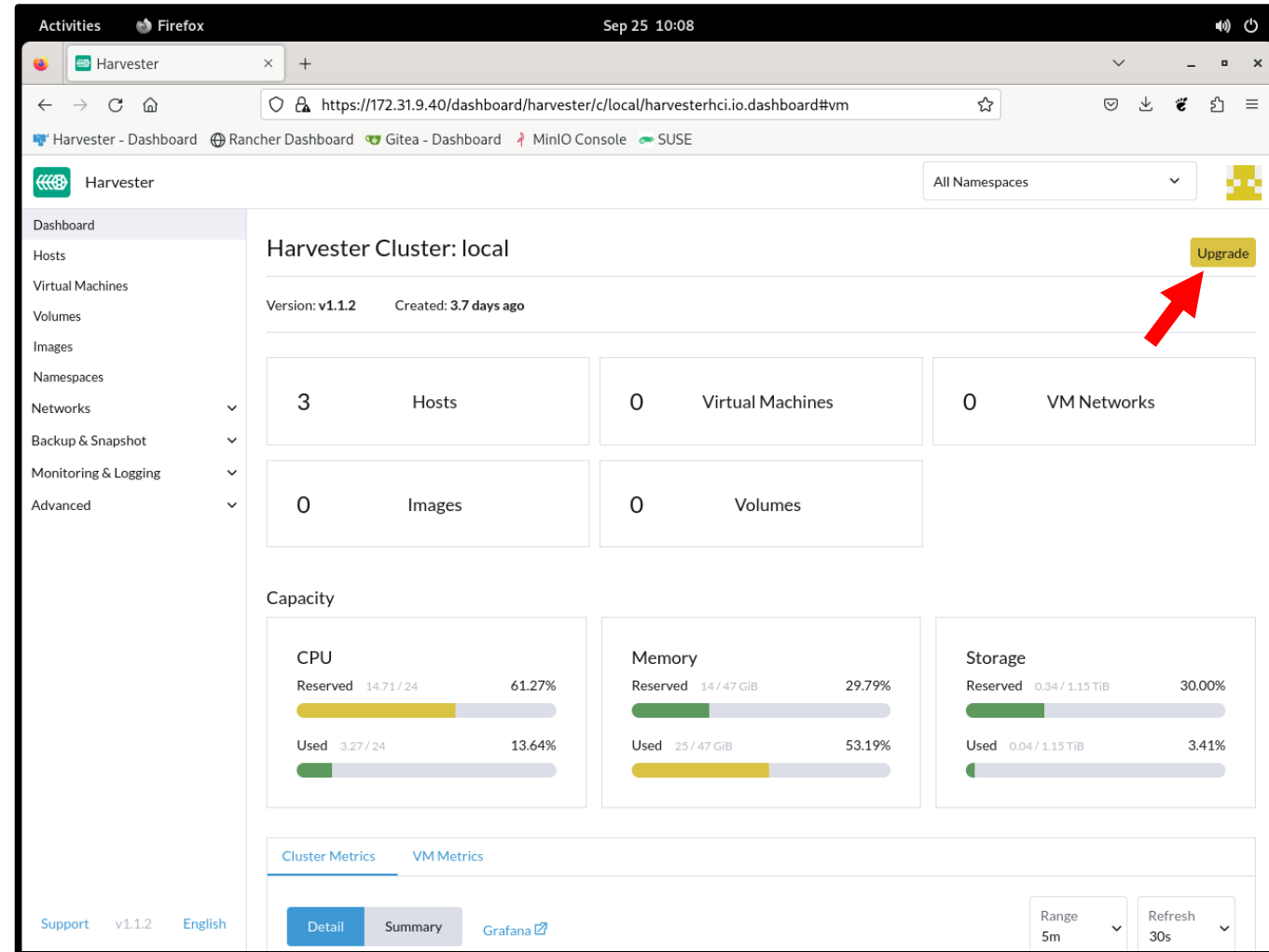


* With the exception that v1.1.0 can be skipped when upgrading from v1.0.3.



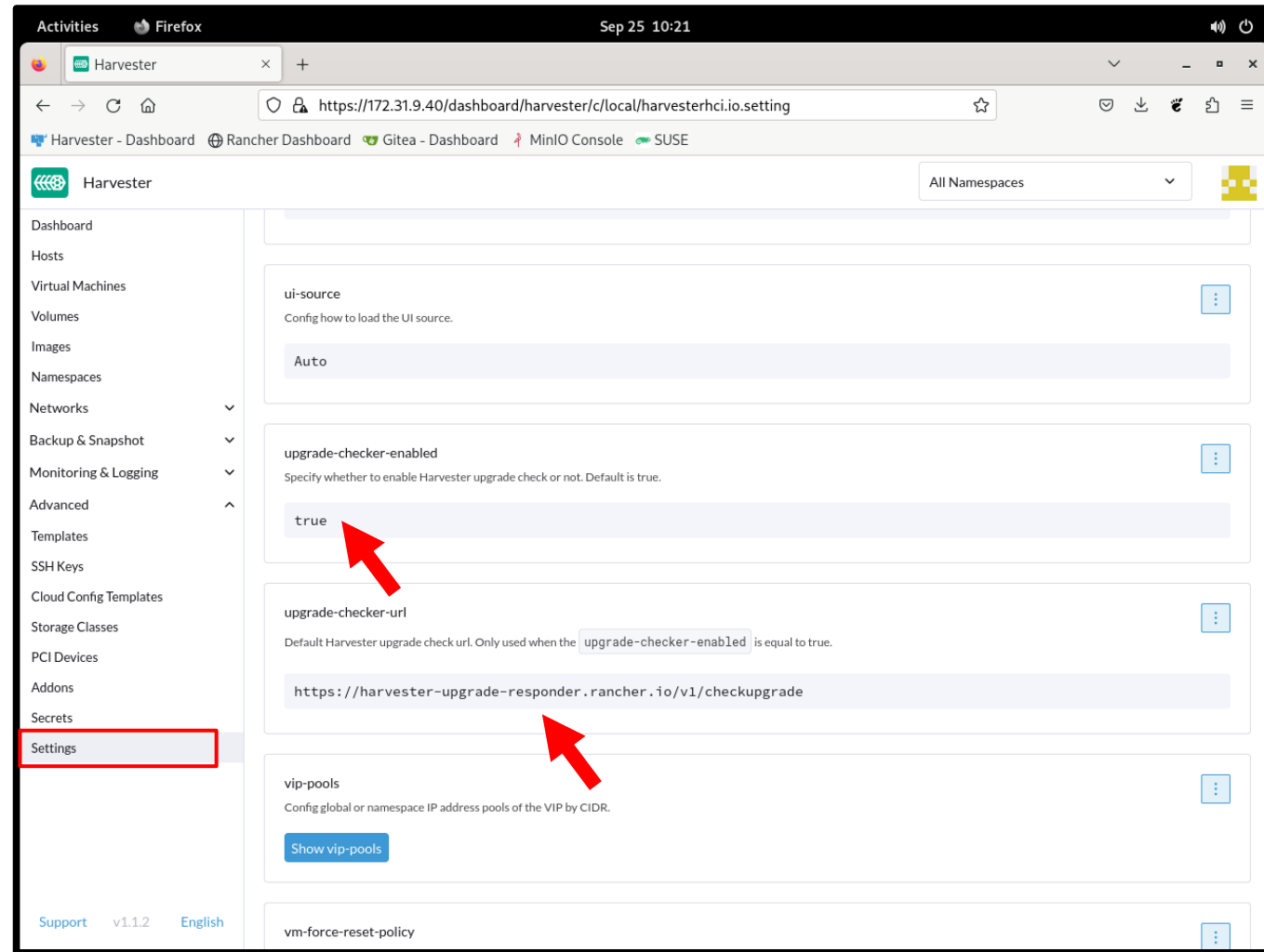
Harvester Cluster Upgrade

- 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, use **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
```

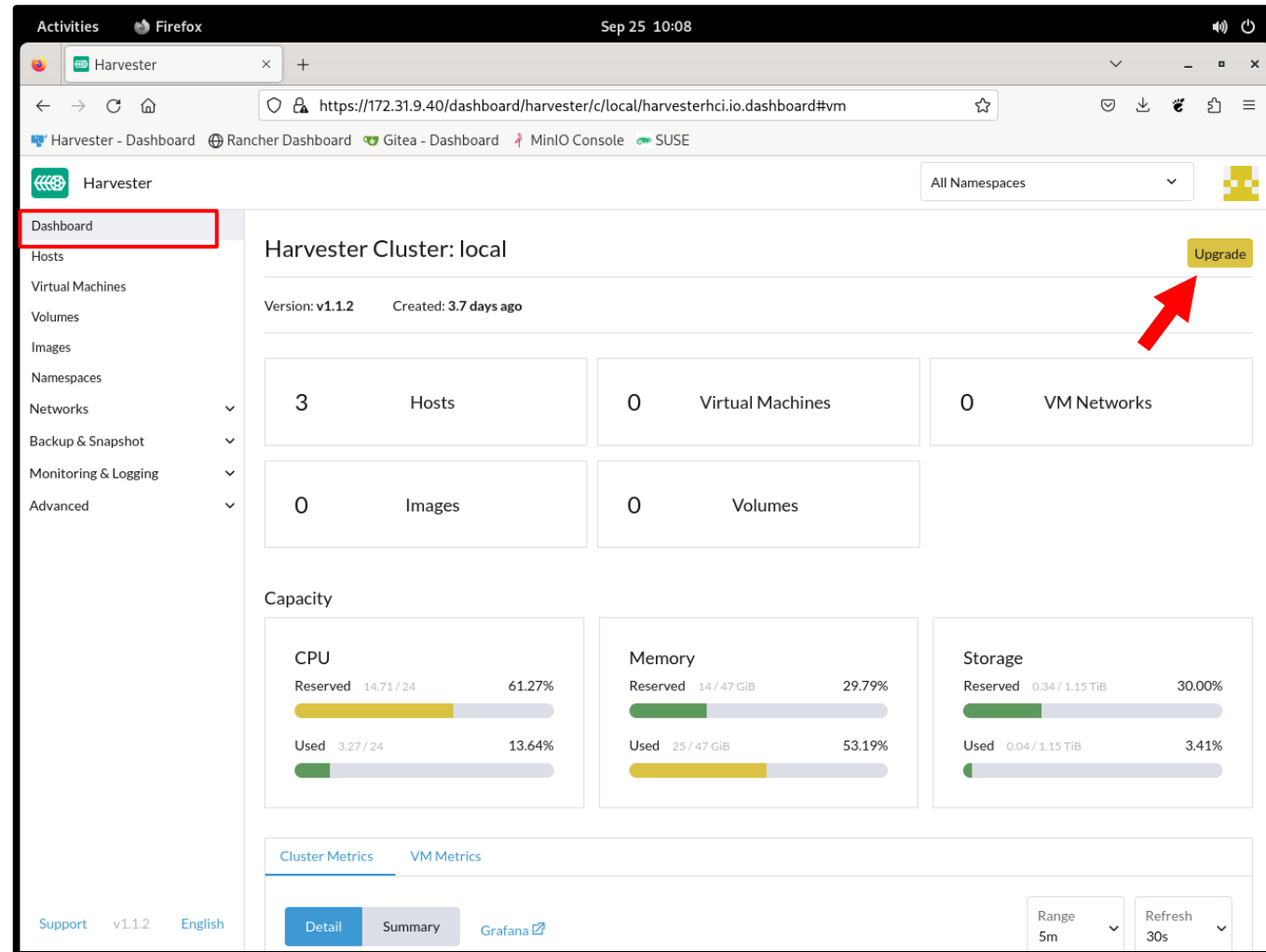
```
>>> Check all bundles ready...
All Helm bundles are ready.
>>> Check the Harvester bundle is ready...
The Harvester bundle is ready.
>>> Check all nodes are ready...
All nodes are ready.
>>> Check the CAPI cluster is provisioned...
The CAPI cluster is provisioned.
>>> Check the CAPI machines count...
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



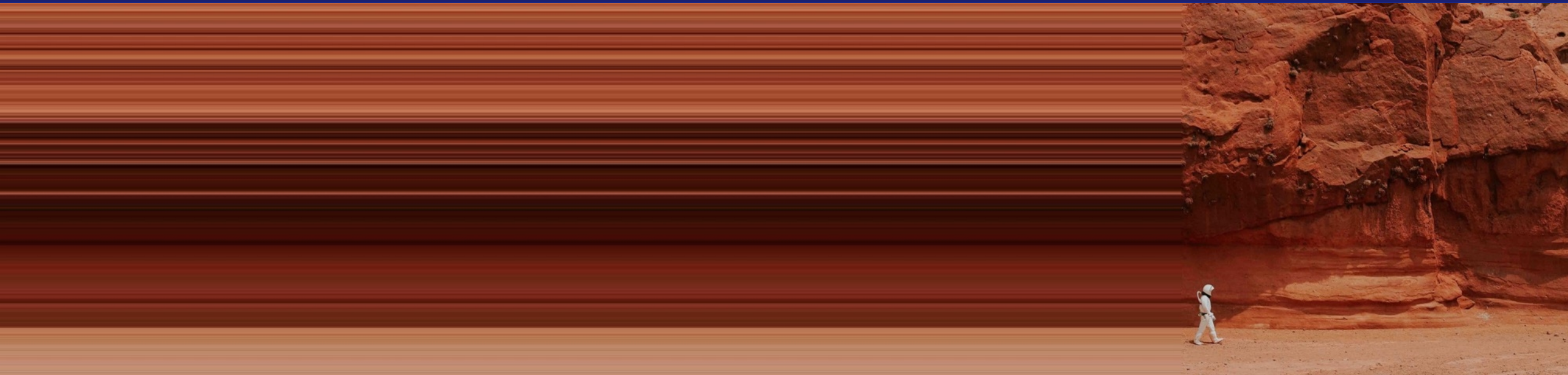
- Upgrade a Harvester cluster
 - Upgrades to a Harvester cluster are enabled when a new version is released.
 - Harvester versions can only be upgraded following a specific upgrade path.
 - A Harvester upgrade can be performed without Internet access by downloading the installation ISO and a **version.yaml**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





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 Kubernetes 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



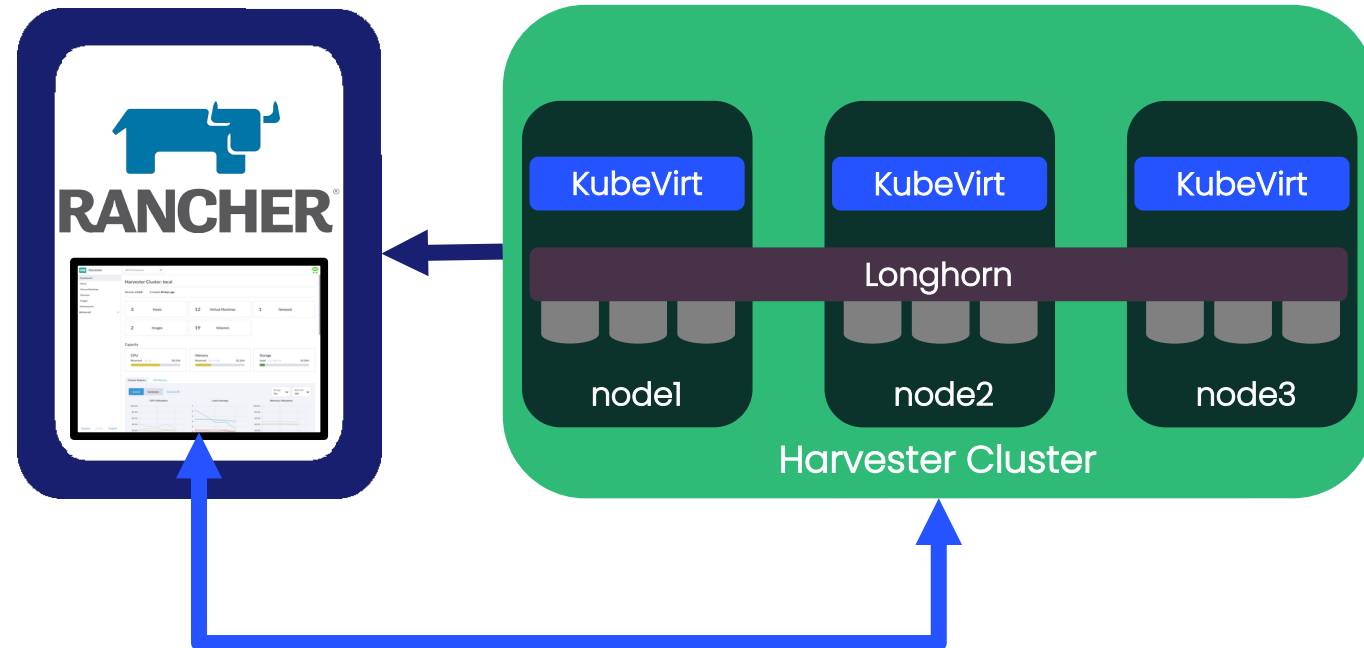
Harvester → Rancher Integration

- 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.



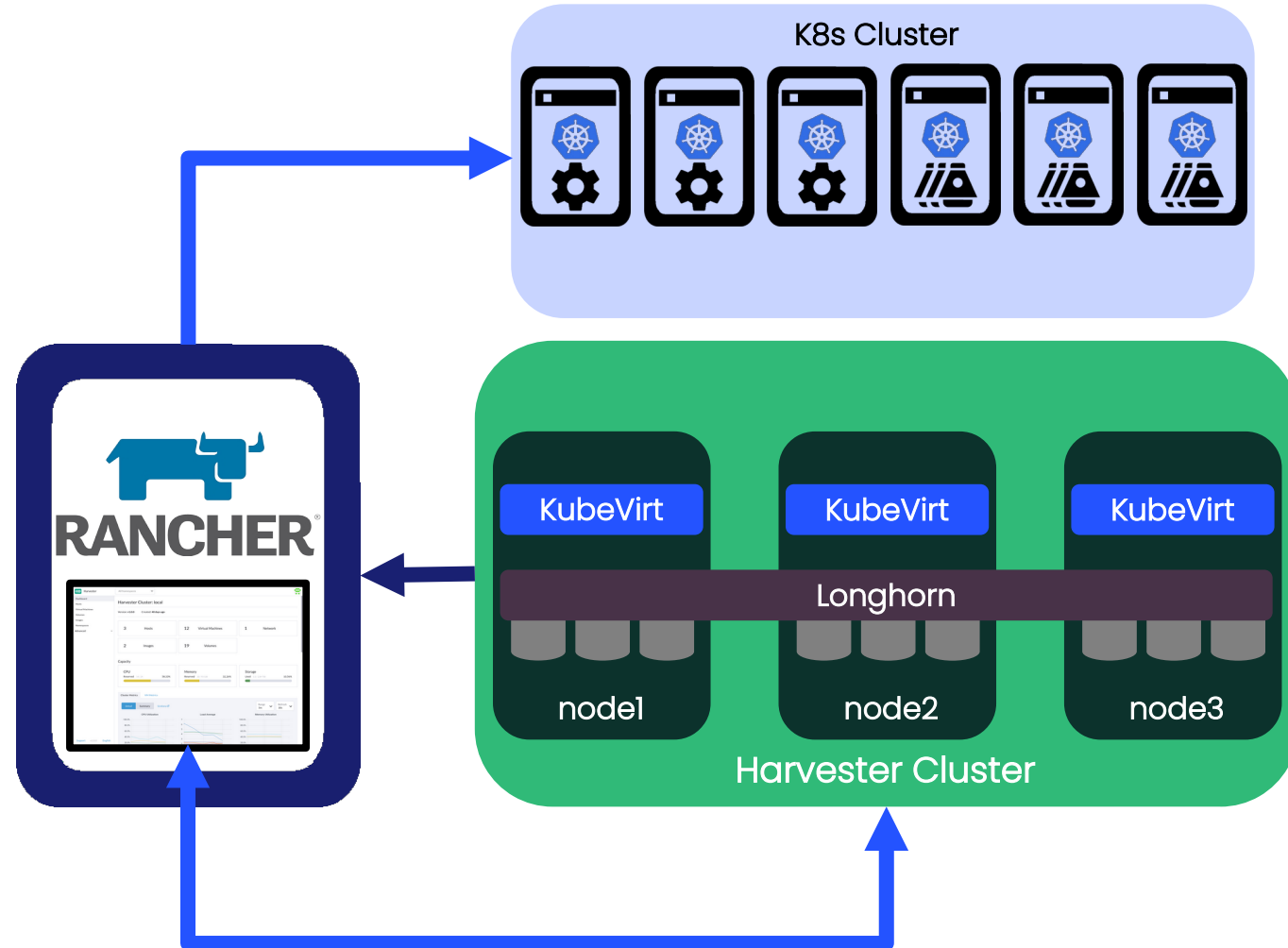
Import Harvester Clusters

- 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.



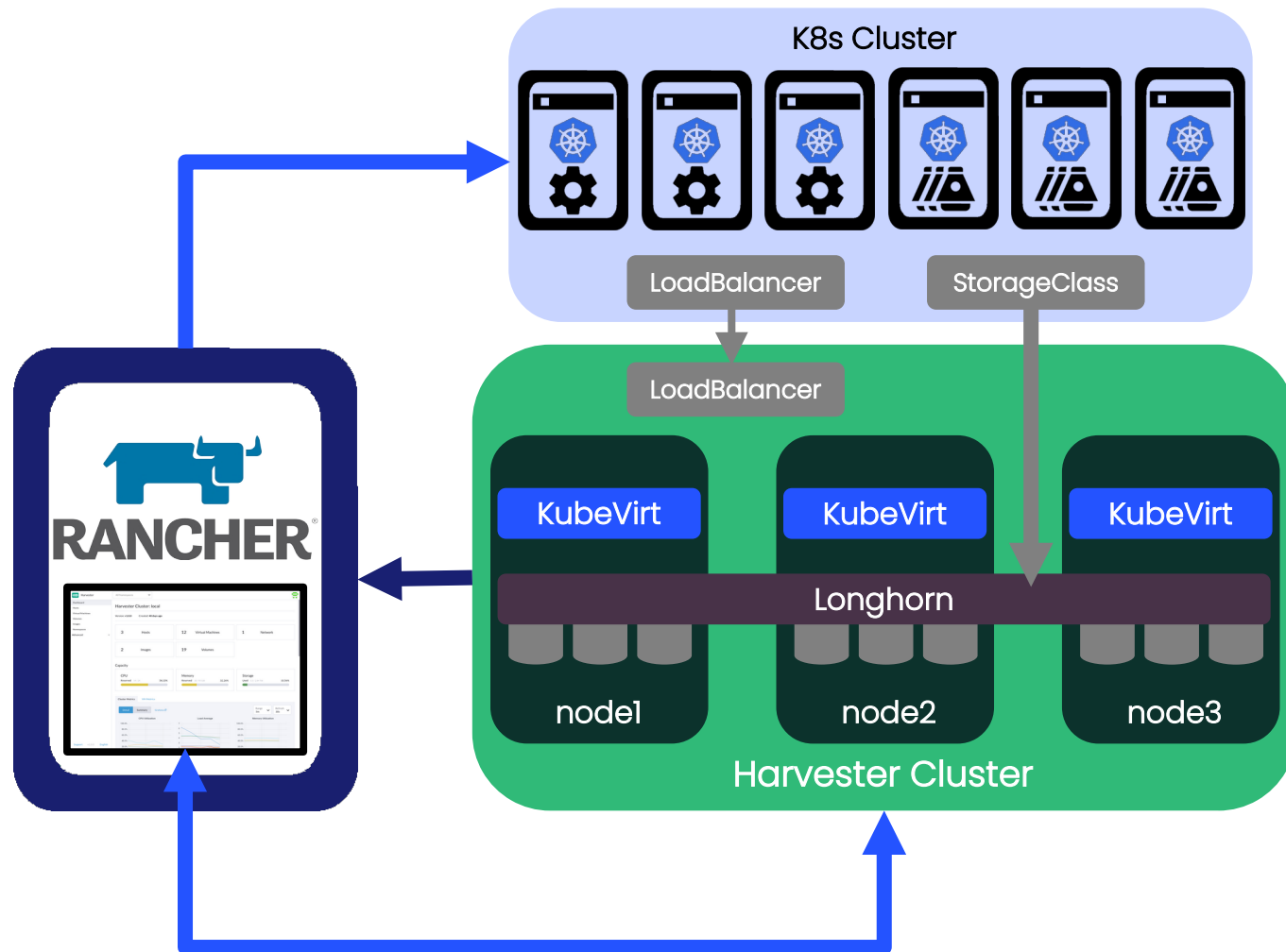
Downstream Clusters on Harvester

- 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



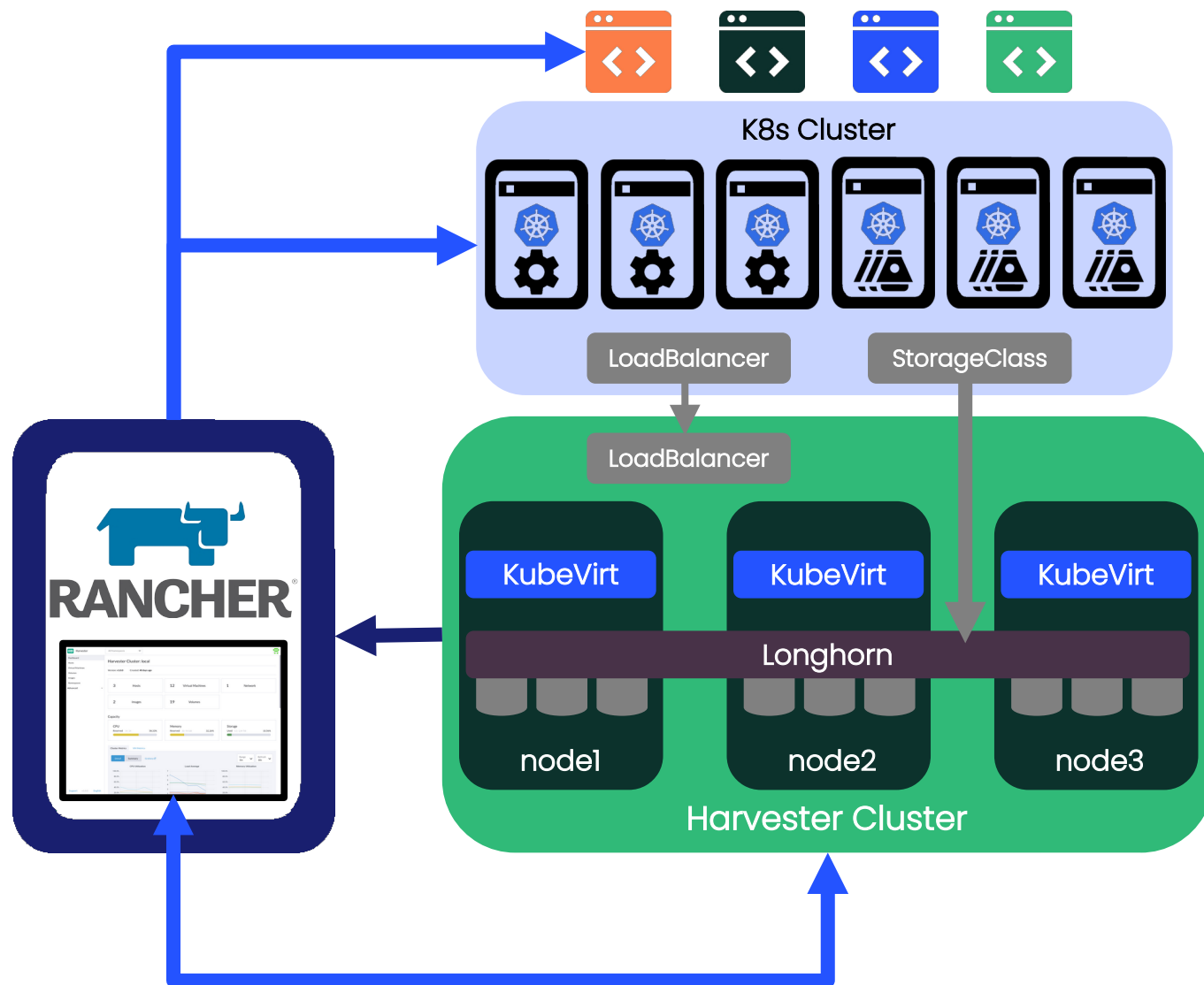
Harvester Cloud Provider Integration

- 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



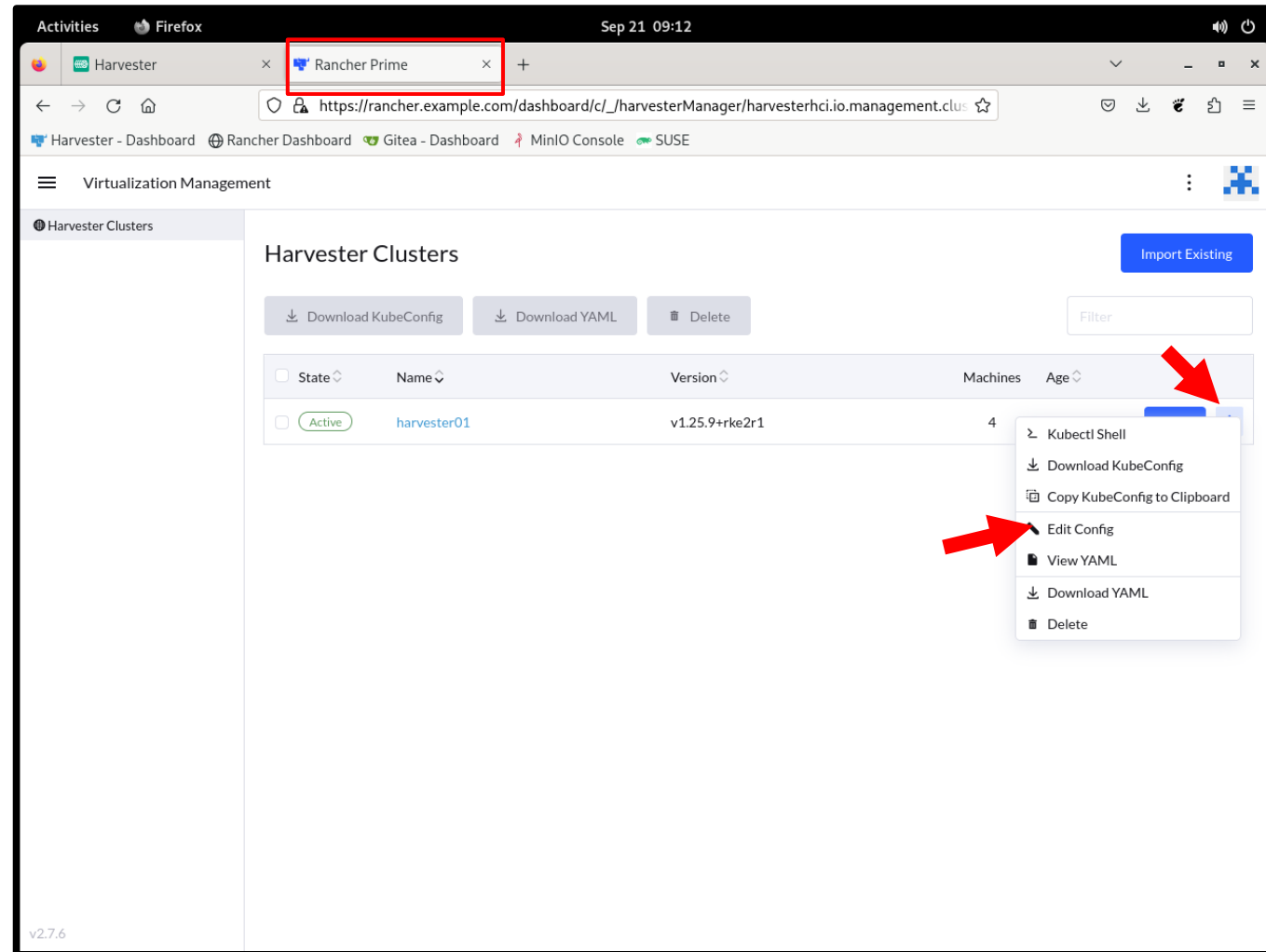
Rancher and Harvester Users

- 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.



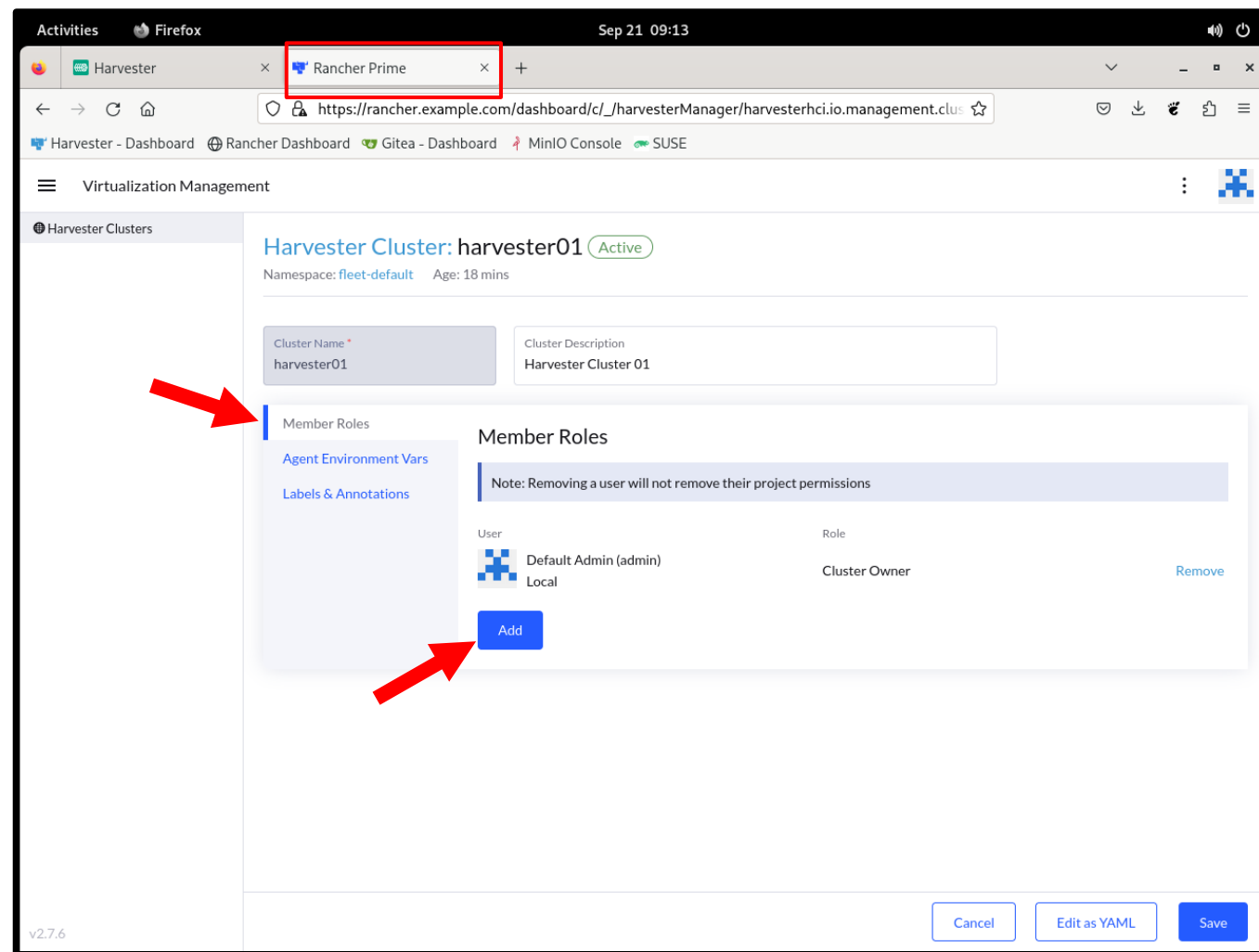
Rancher: Harvester Cluster Config

- 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**.



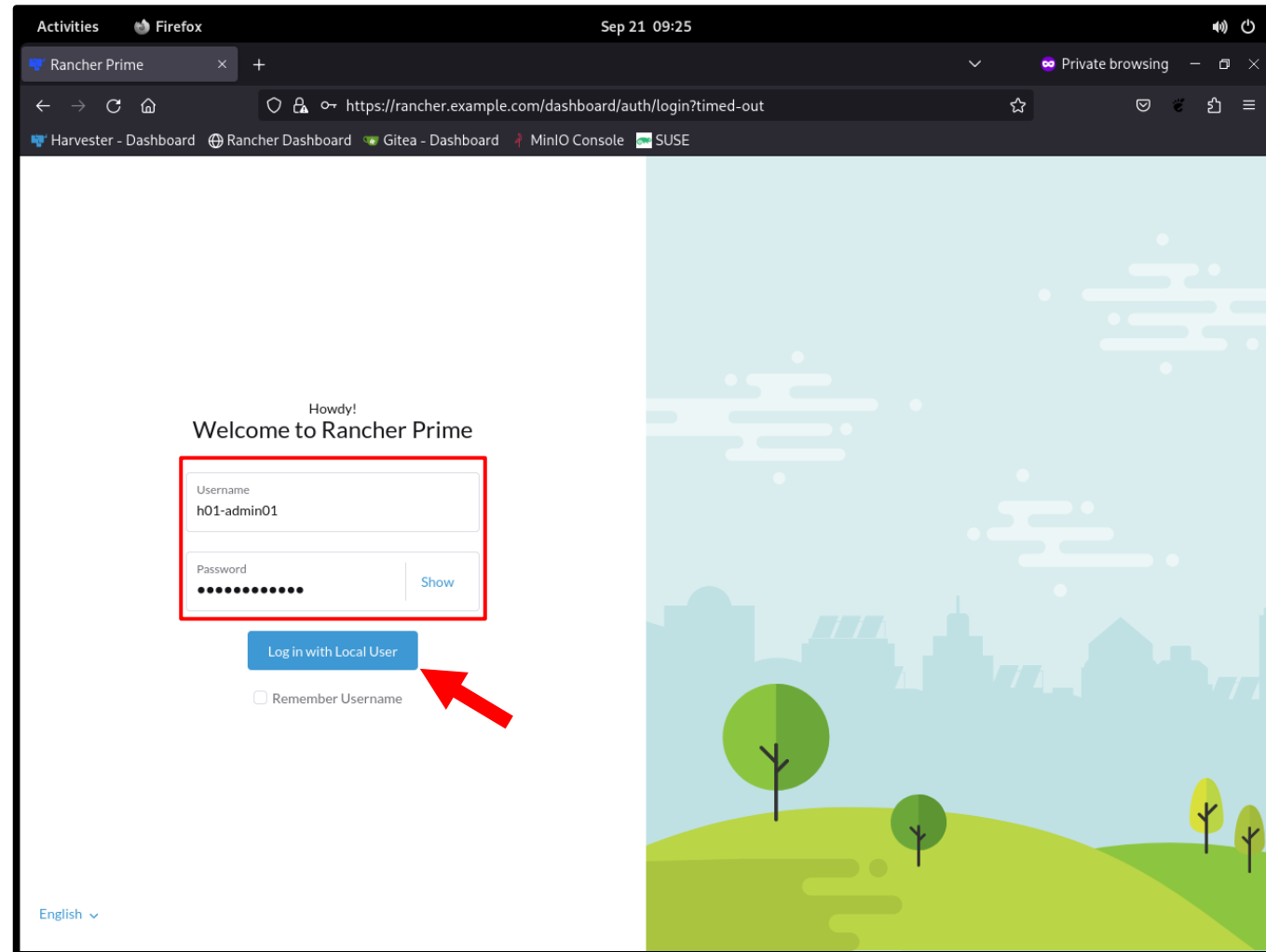
Rancher: Add Rancher Users to a Harvester Cluster

- 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**



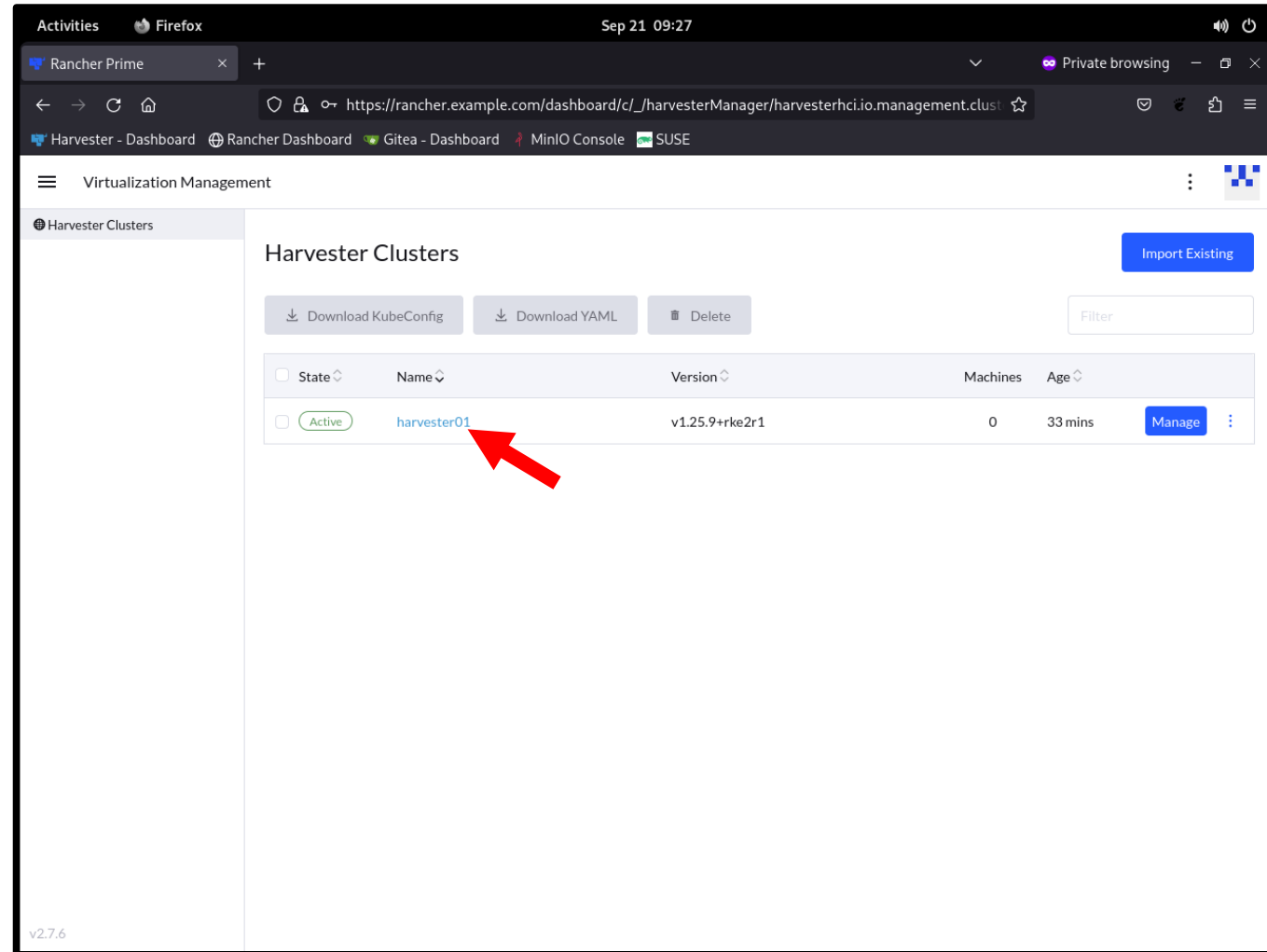
Rancher: Harvester User Login

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



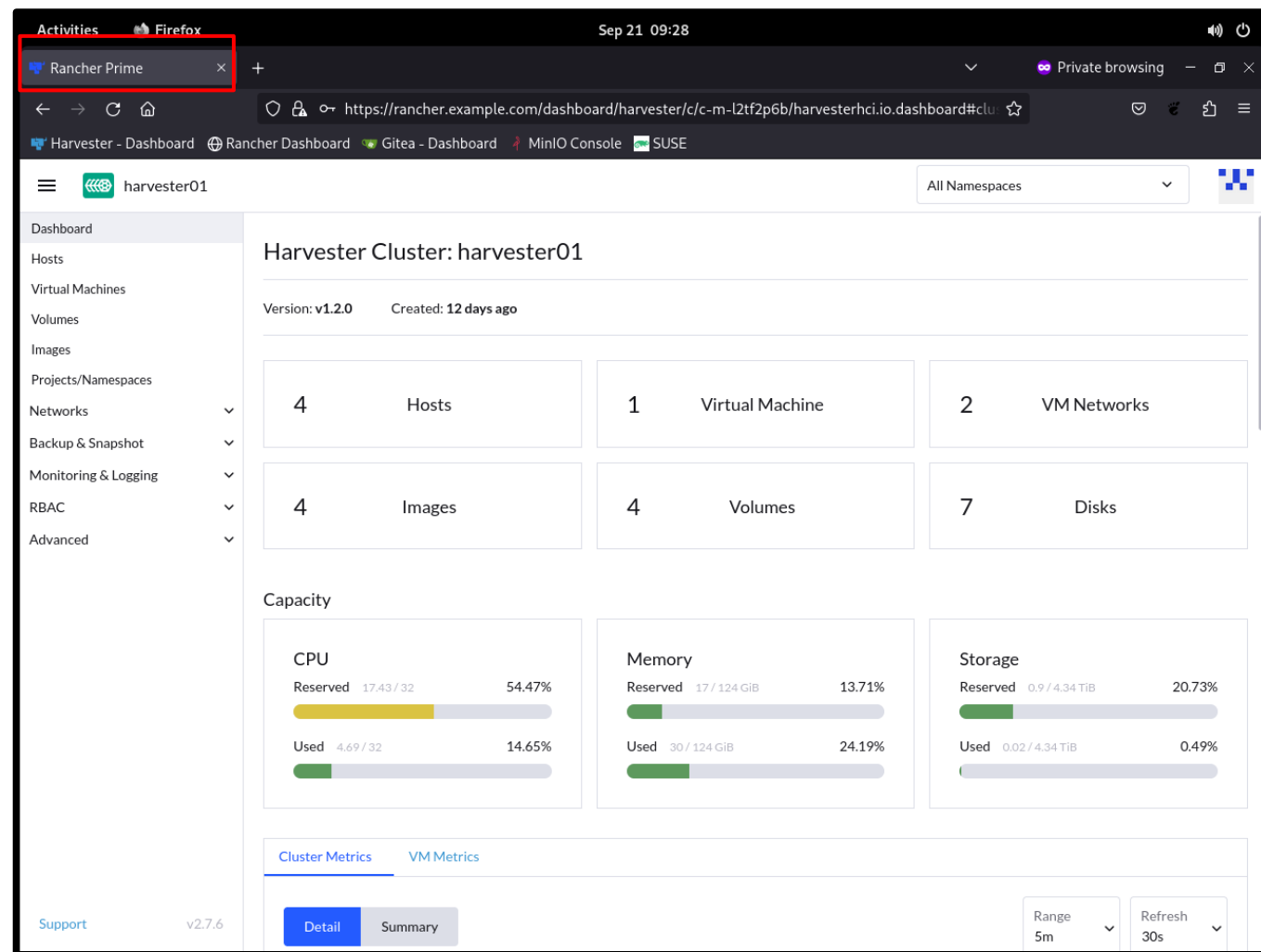
Rancher: Harvester Clusters

- 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).

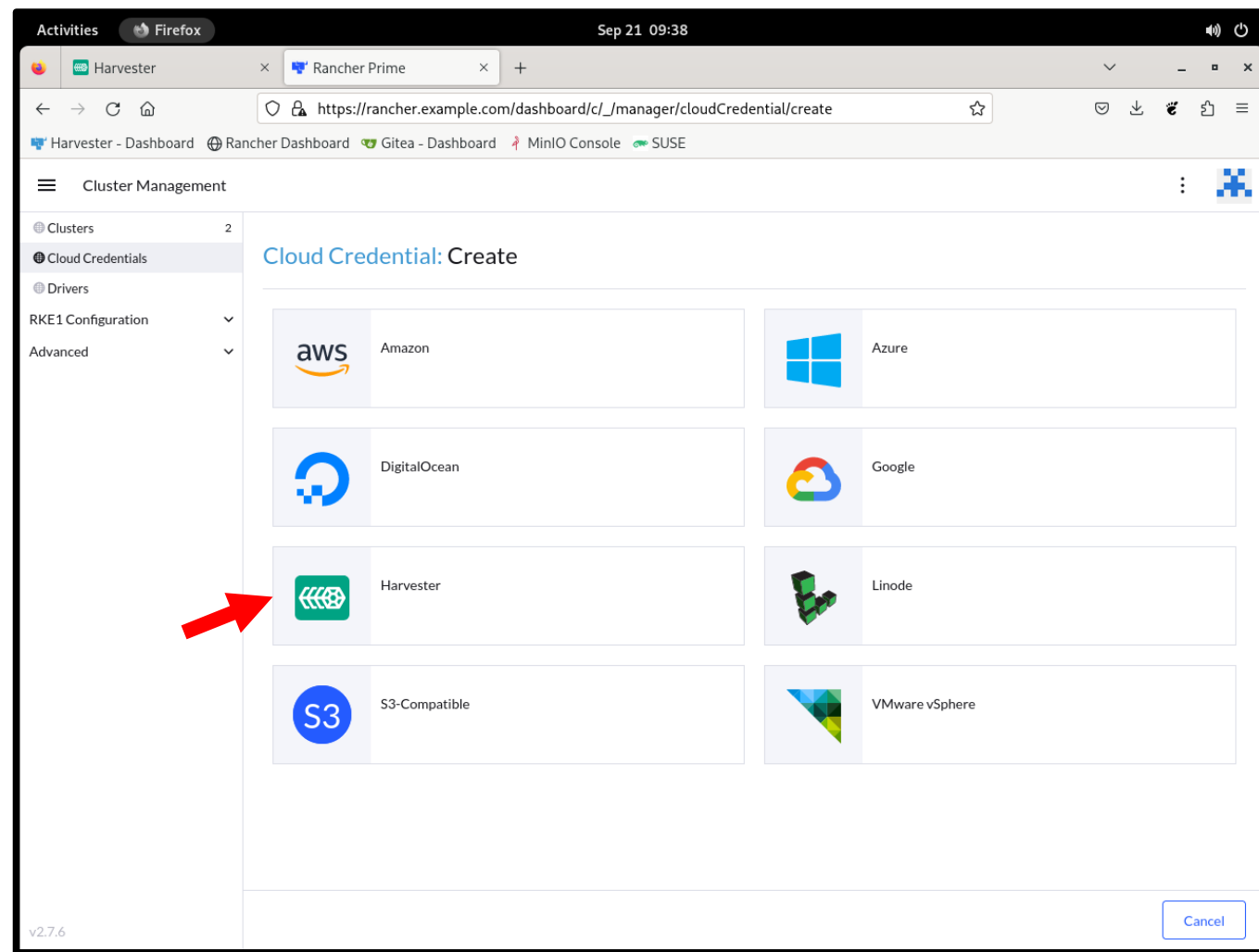


Cloud Credentials



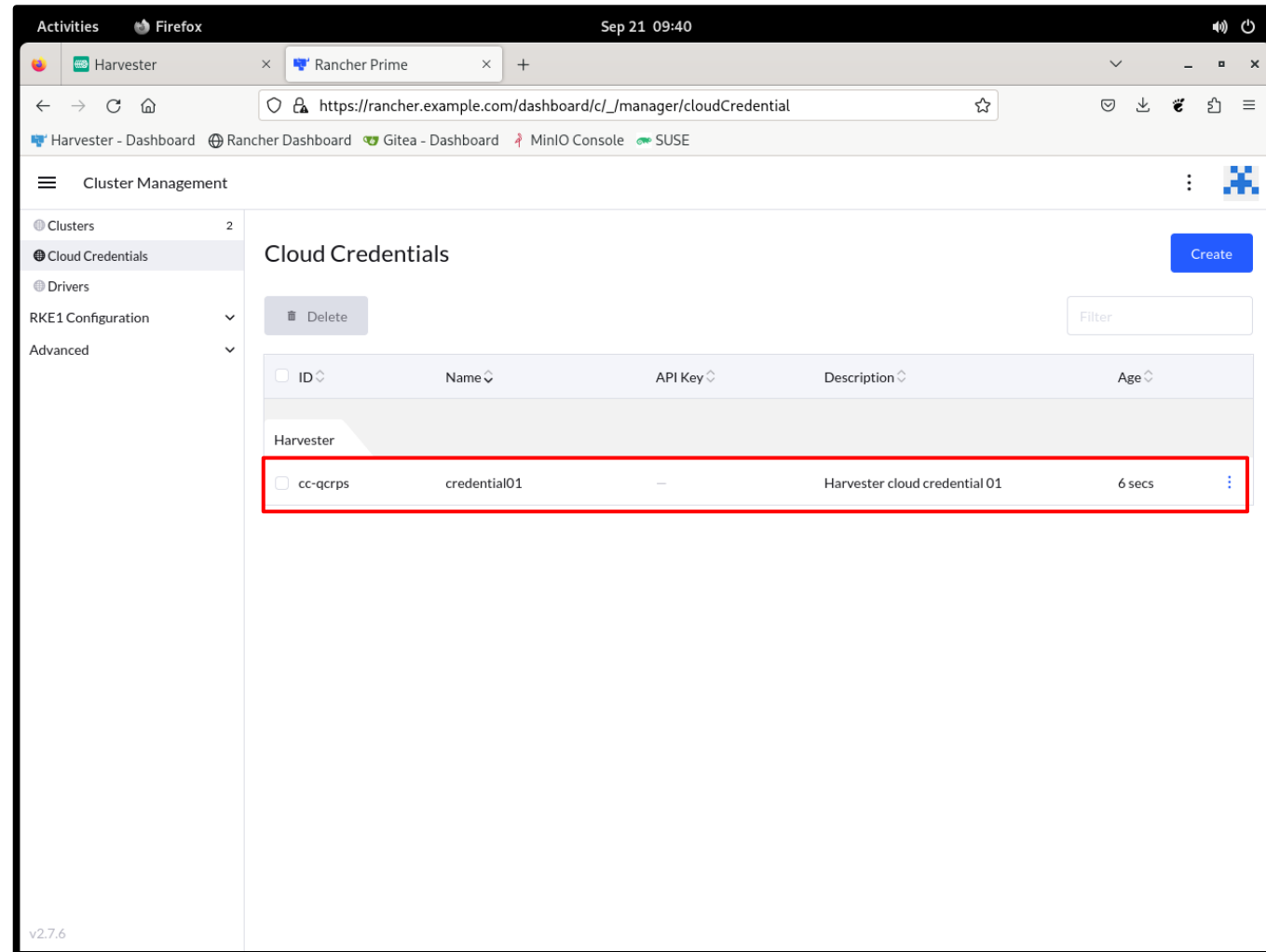
Cloud Credentials

- 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.

Activities Firefox Sep 21 09:48

Harvester x Rancher Prime x +

https://rancher.example.com/dashboard/c/_manager/provisioning.cattle.io/cluster/create 90%

Harvester - Dashboard Rancher Dashboard Gitea - Dashboard MinIO Console SUSE

Cluster Management

Clusters 2

Cloud Credentials

Drivers

RKE1 Configuration

Advanced

Cluster: Create Harvester

Cloud Credential credential01

Cluster Name * rke2-c01

Cluster Description RKE2 Cluster01

Machine Pools 1 etcd 1 Control Plane 1 Worker

pool1

Pool Name * pool1

Machine Count * 1

Roles etcd Control Plane Worker

CPUs * 2 C

Memory * 4 GiB

Namespace * e.g. default

SSH User * e.g. ubuntu

Volumes

Image Volume

Image * Please select a image

Disk * 40 GiB

bootOrder: 1

Cancel Edit as YAML Create

v2.7.6

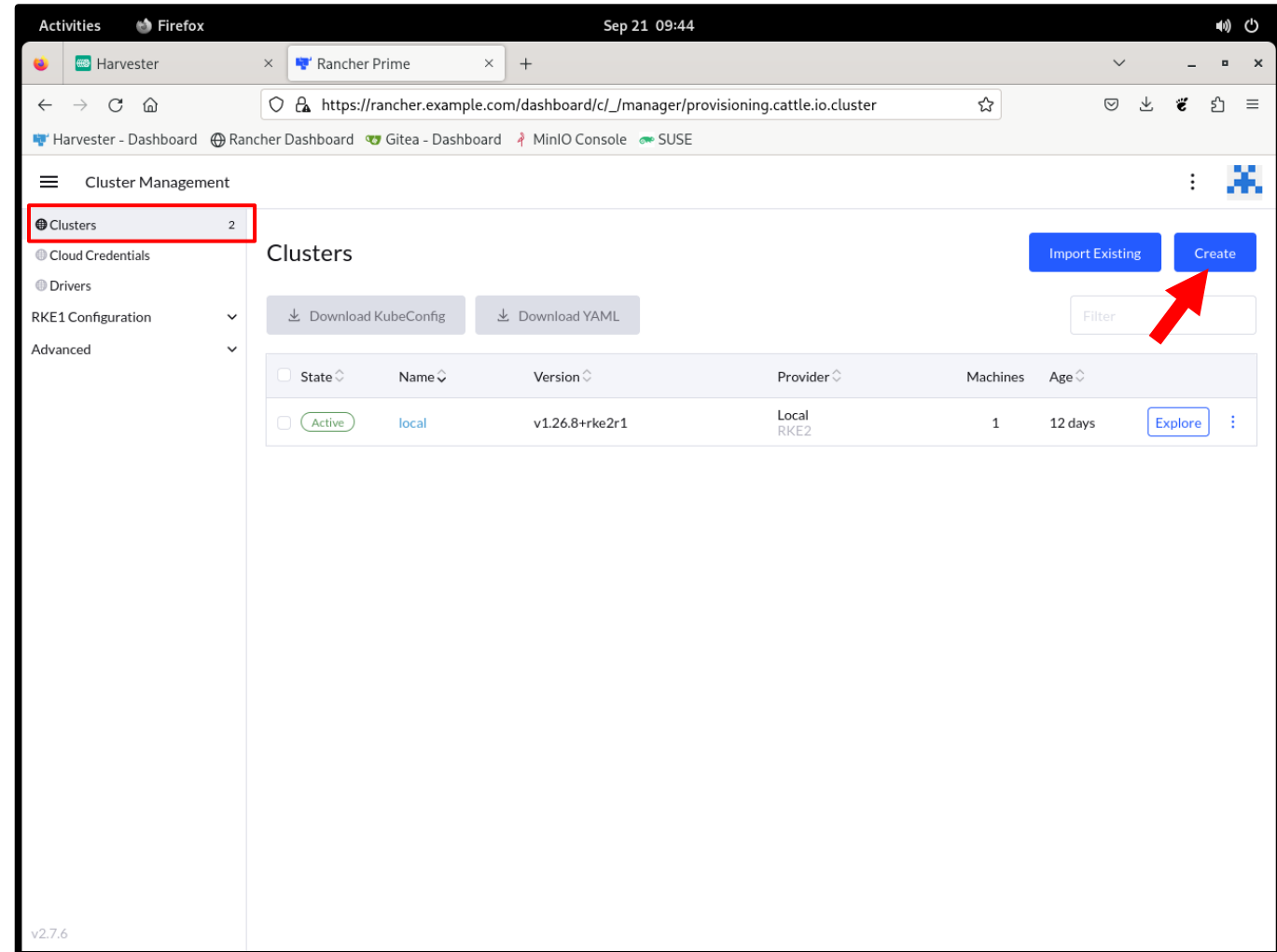


Cluster Deployment with Rancher onto Harvester



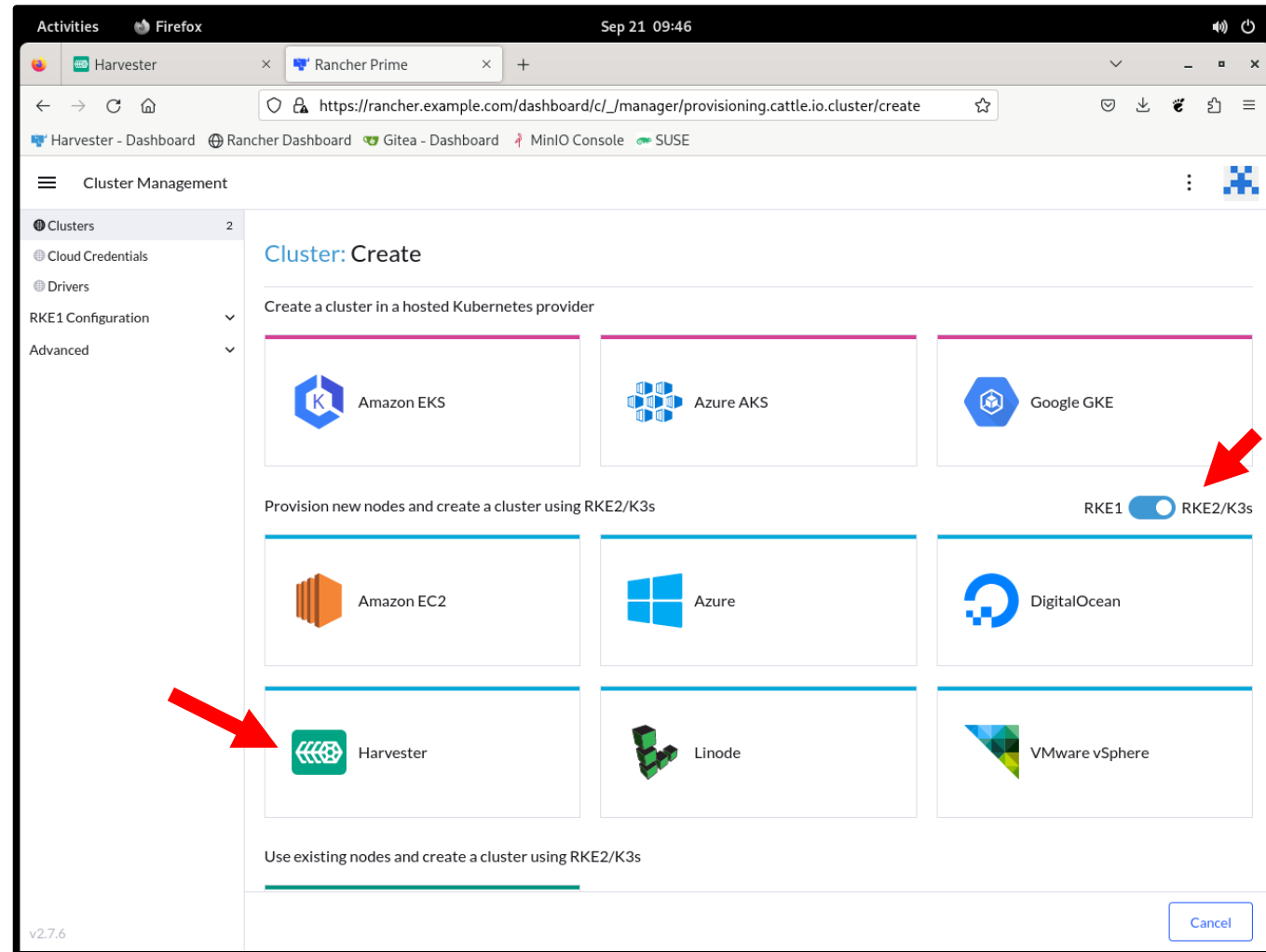
Deploy an RKE2 Cluster

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



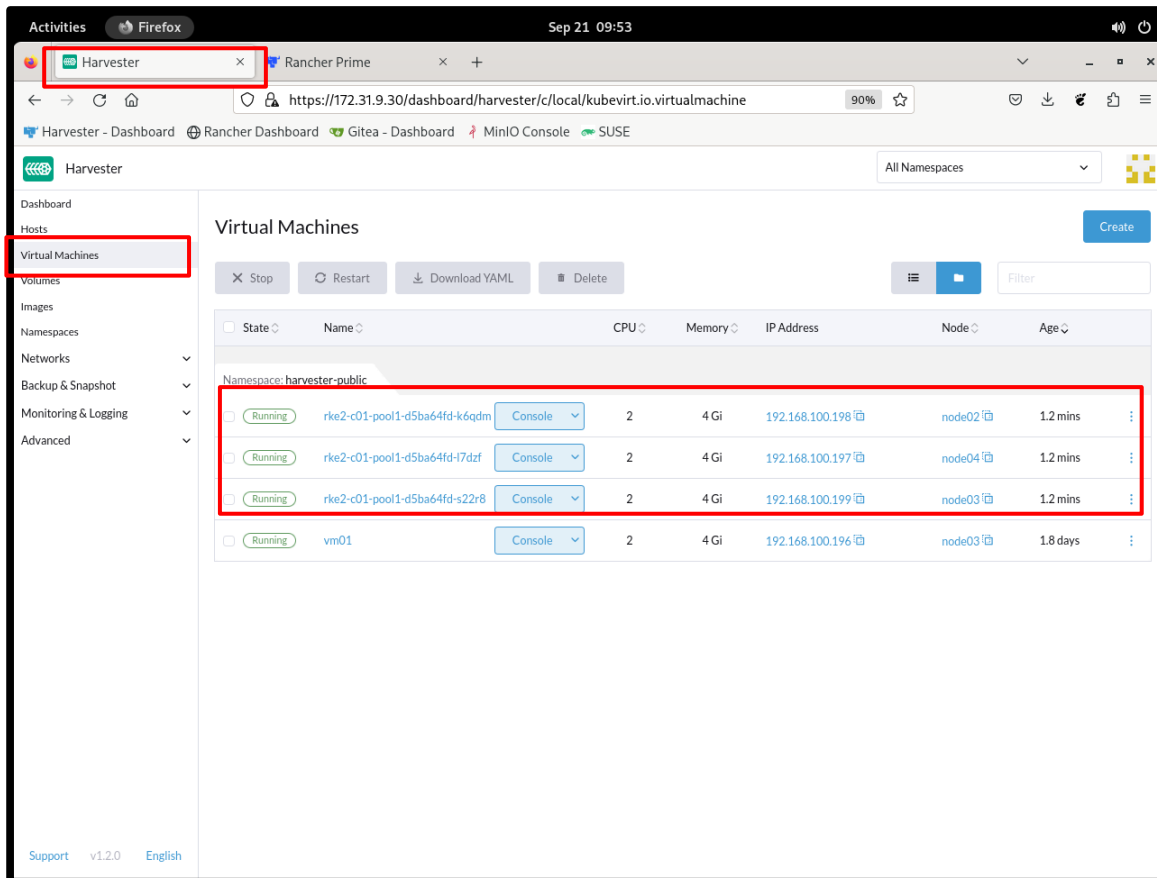
Deploy an RKE2 Cluster

- 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** screen



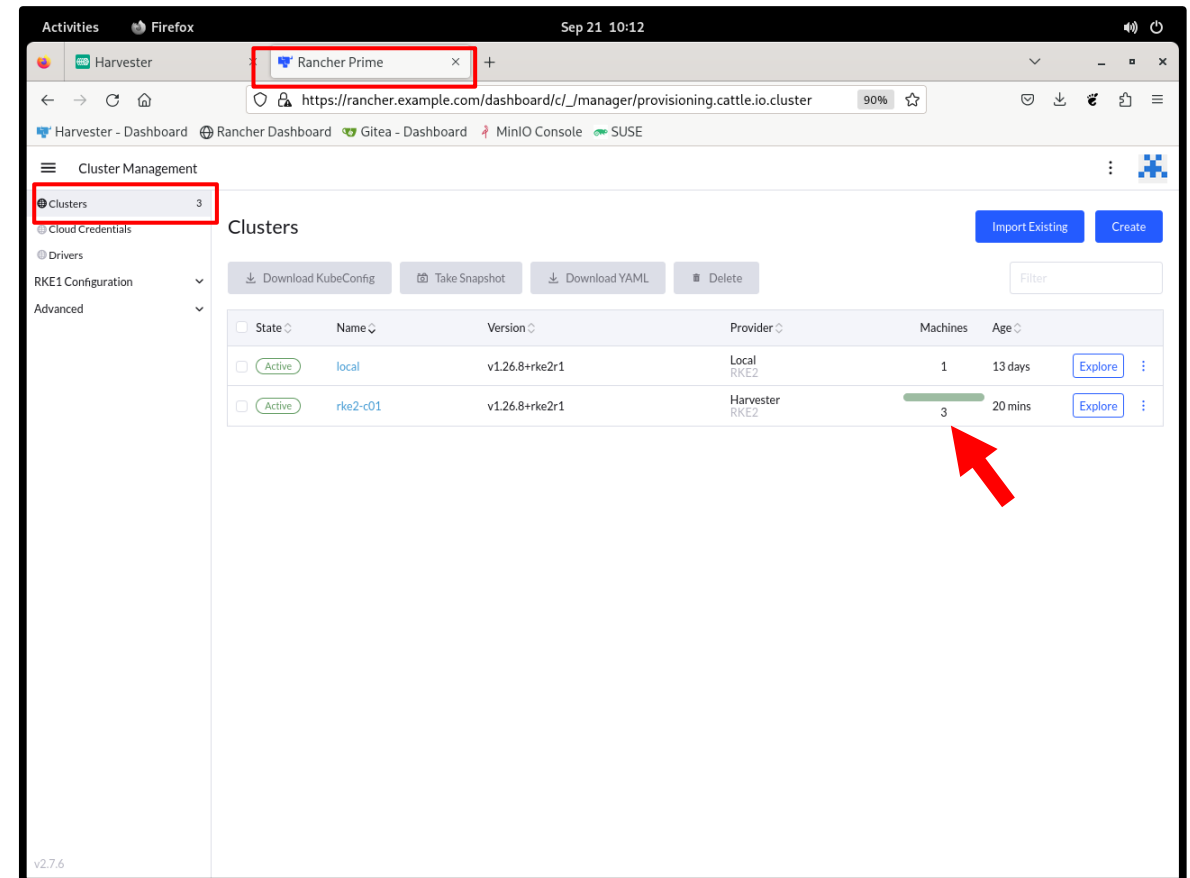
Harvester - Dashboard

Virtual Machines

Create

Stop Restart Download YAML Delete

State	Name	CPU	Memory	IP Address	Node	Age
Running	rke2-c01-pool1-d5ba64fd-k6qdm	2	4 Gi	192.168.100.198	node02	1.2 mins
Running	rke2-c01-pool1-d5ba64fd-l7dzf	2	4 Gi	192.168.100.197	node04	1.2 mins
Running	rke2-c01-pool1-d5ba64fd-s22r8	2	4 Gi	192.168.100.199	node03	1.2 mins
Running	vm01	2	4 Gi	192.168.100.196	node03	1.8 days



Rancher Prime

Cluster Management

Clusters

Download KubeConfig Take Snapshot Download YAML Delete

State	Name	Version	Provider	Machines	Age
Active	local	v1.26.8+rke2r1	Local RKE2	1	13 days
Active	rke2-c01	v1.26.8+rke2r1	Harvester RKE2	3	20 mins



Application Deployment with Rancher onto Clusters Running on Harvester



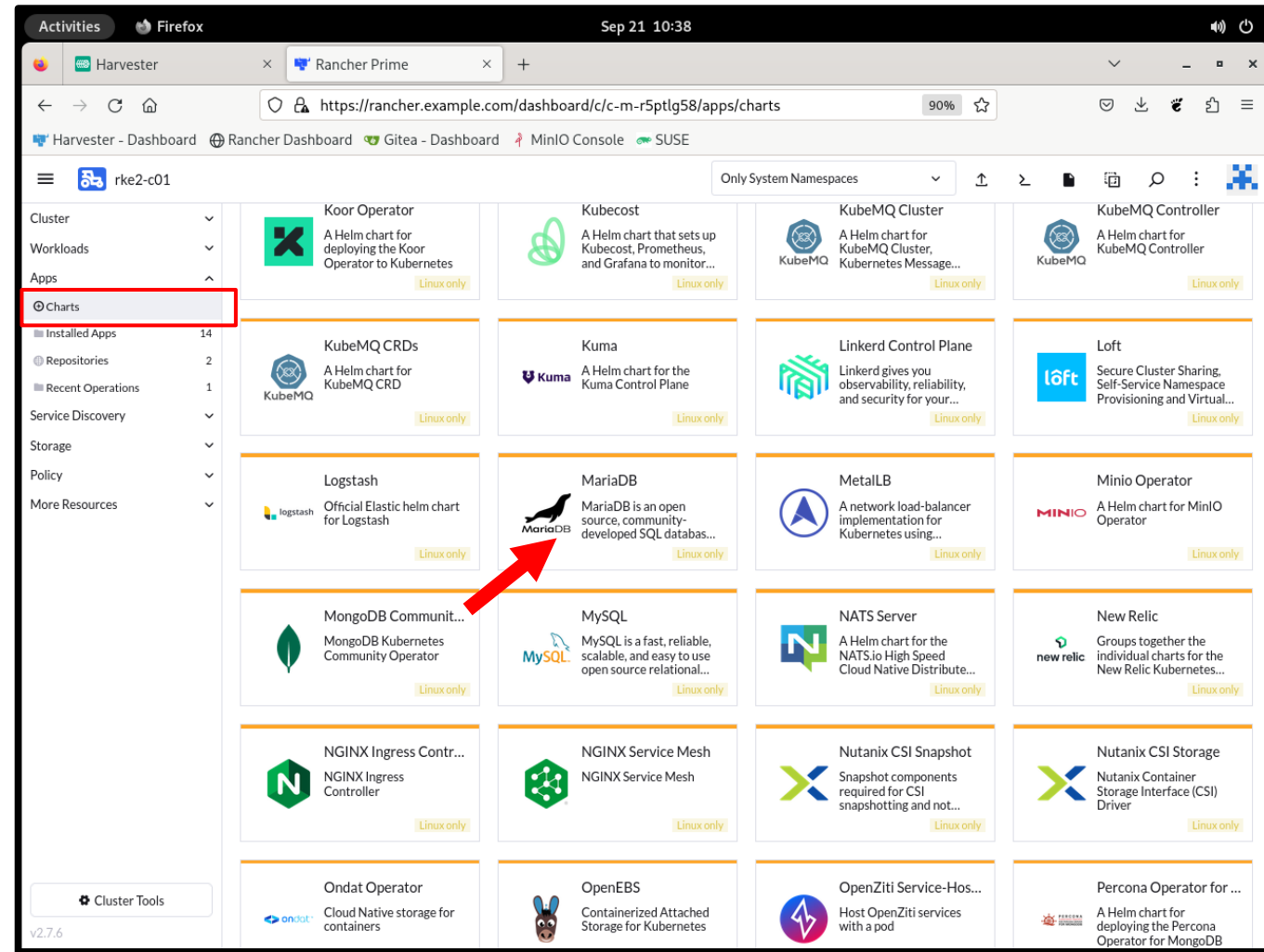
Deploy Applications on a Harvester Hosted RKE2 Cluster

- 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.



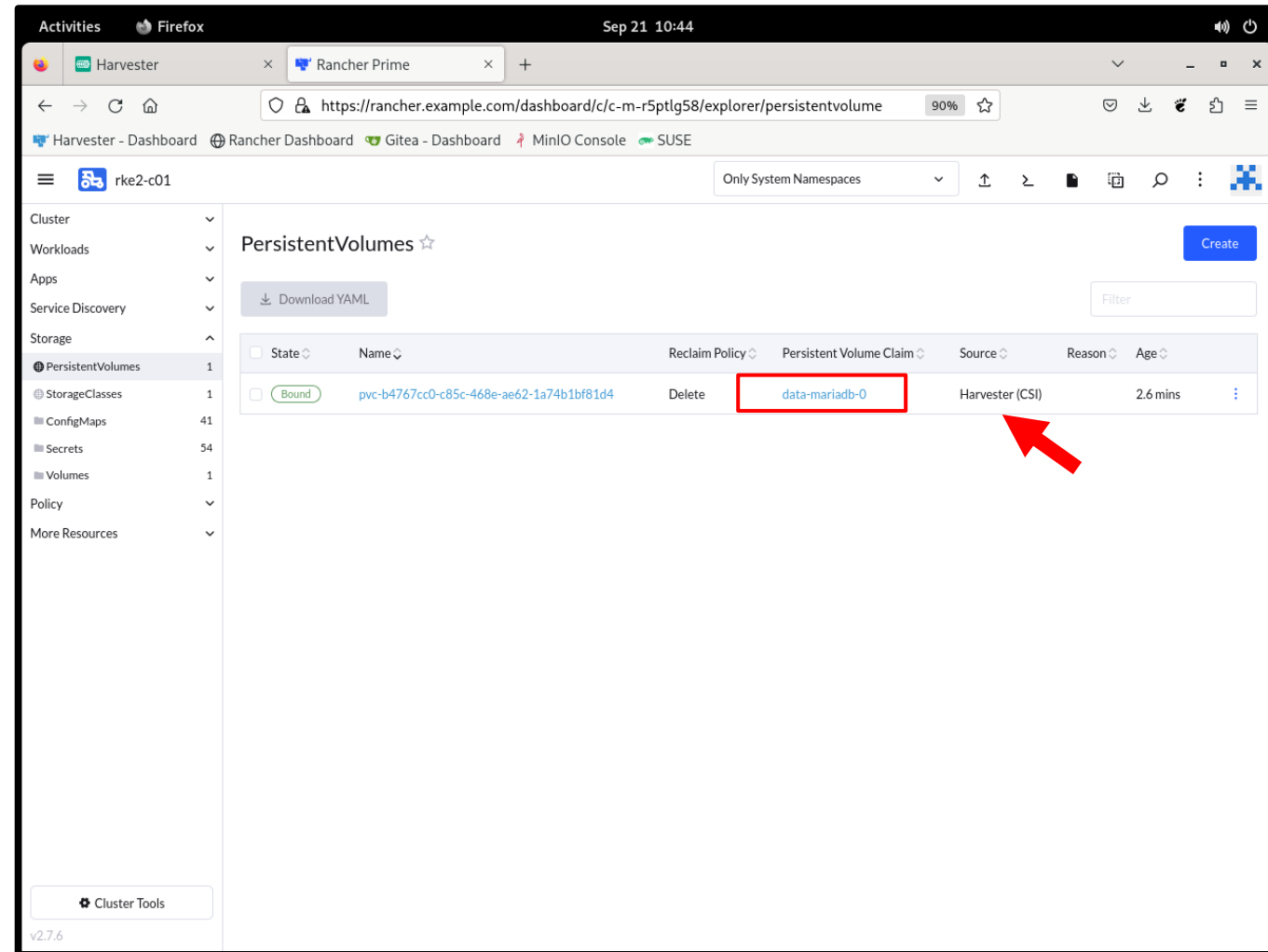
Deploy an Application

- 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.