Deployment on ROSA with FSxN

NetApp Solutions

NetApp
December 19, 2024

Table of Contents

Deployment on ROSA with FSxN

Deploy Red Hat OpenShift Virtualization with FSxN on ROSA

Overview

This section provides details for setting up FSx for NetApp ONTAP as the default Storage Class for the ROSA cluster, and then create a Virtual Machine that will leverage FSx ONTAP storage for its volumes.
We will also look into connecting to the Virtual Machine using the guest credentials, and restarting the VM. And finally, we will perform a live migration of the Virtual Machine from the current node to a new node. We will examine the contents of the disk storage after a VM restart and the live migration .

Prerequisites

- AWS account
- A Red Hat account
- IAM user with appropriate permissions to create and access ROSA cluster
- AWS CLI
- ROSA CLI
- OpenShift command-line interface (oc)
- Helm 3 documentation
- A HCP ROSA cluster (with at least 3 bare-metal worker nodes)
- OpenShift Virtualization installed on ROSA Cluster
- Access to Red Hat OpenShift web console

Initial Setup

This section shows how to set up the default storage class to be trident-csi and the default VolumeSnapshotClass to be the FSx Volume Snapshot class. Then it shows how to create a VM from a template and then connect and login to it using the guest credentials.

Ensure default Storage Class is set to trident-csi

Ensure default VolumeSnapShotClasses is set as shown

If the defaults are not set, you can set it up either from the console or from command line

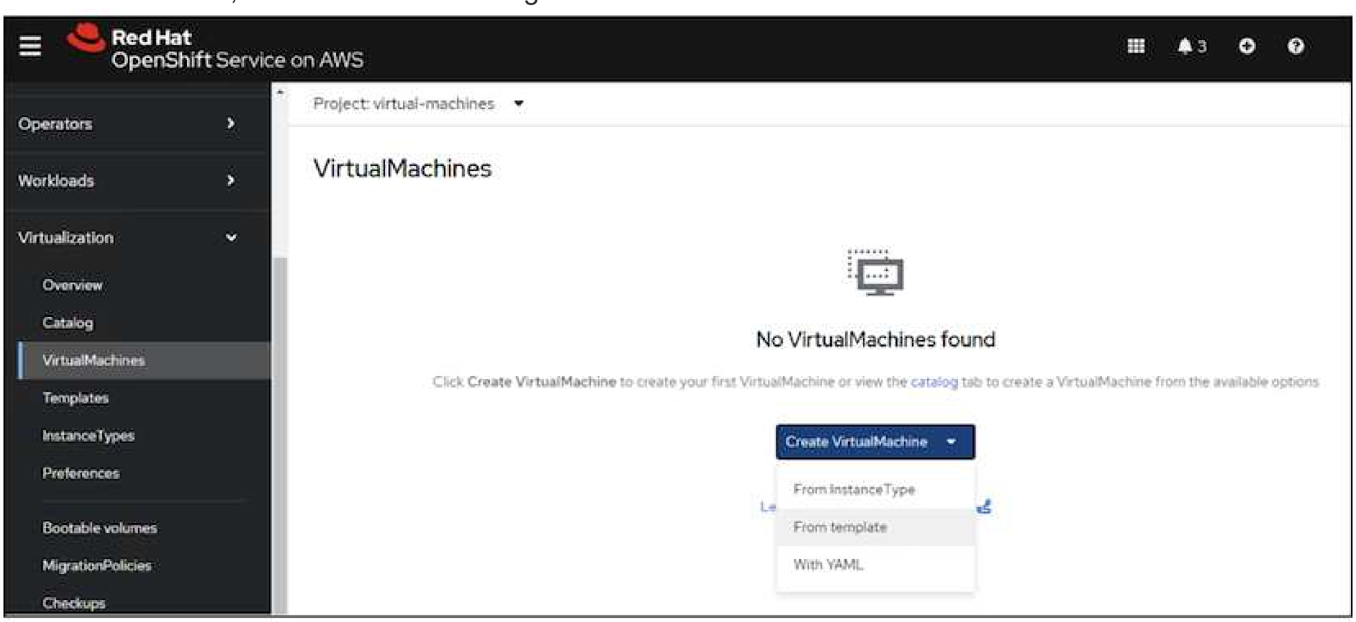
```
$ oc patch storageclass trident-csi -p '{"metadata": {"annotations":
{"storageclass.kubernetes.io/is-default-class": "true"}}}'
```

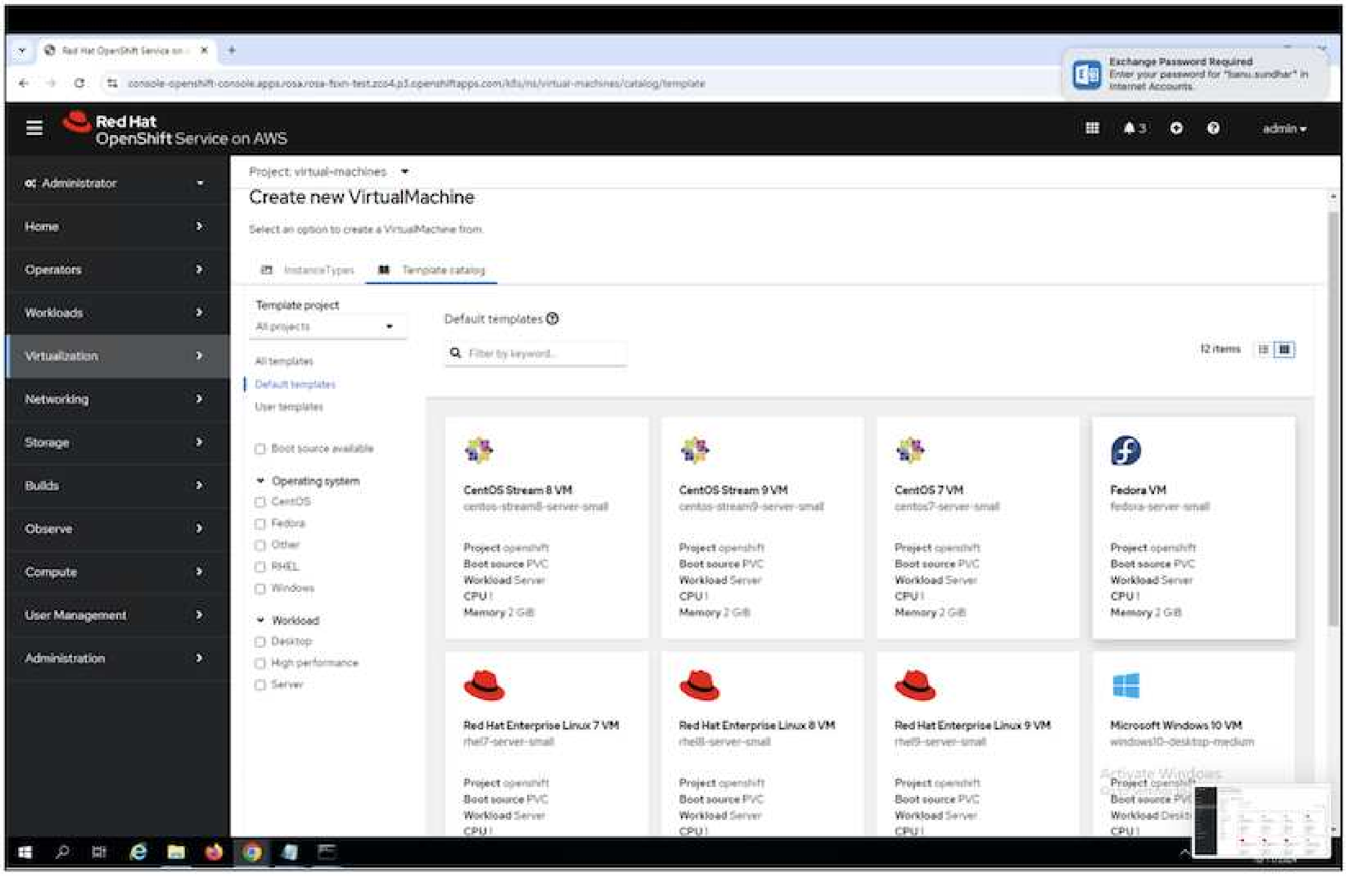
```
$ oc patch VolumeSnapshotClasses fsx-snapclass -p '{"metadata":
{"annotations": {"snapshot.storage.kubernetes.io/is-default-class":
"true"}}}'
```

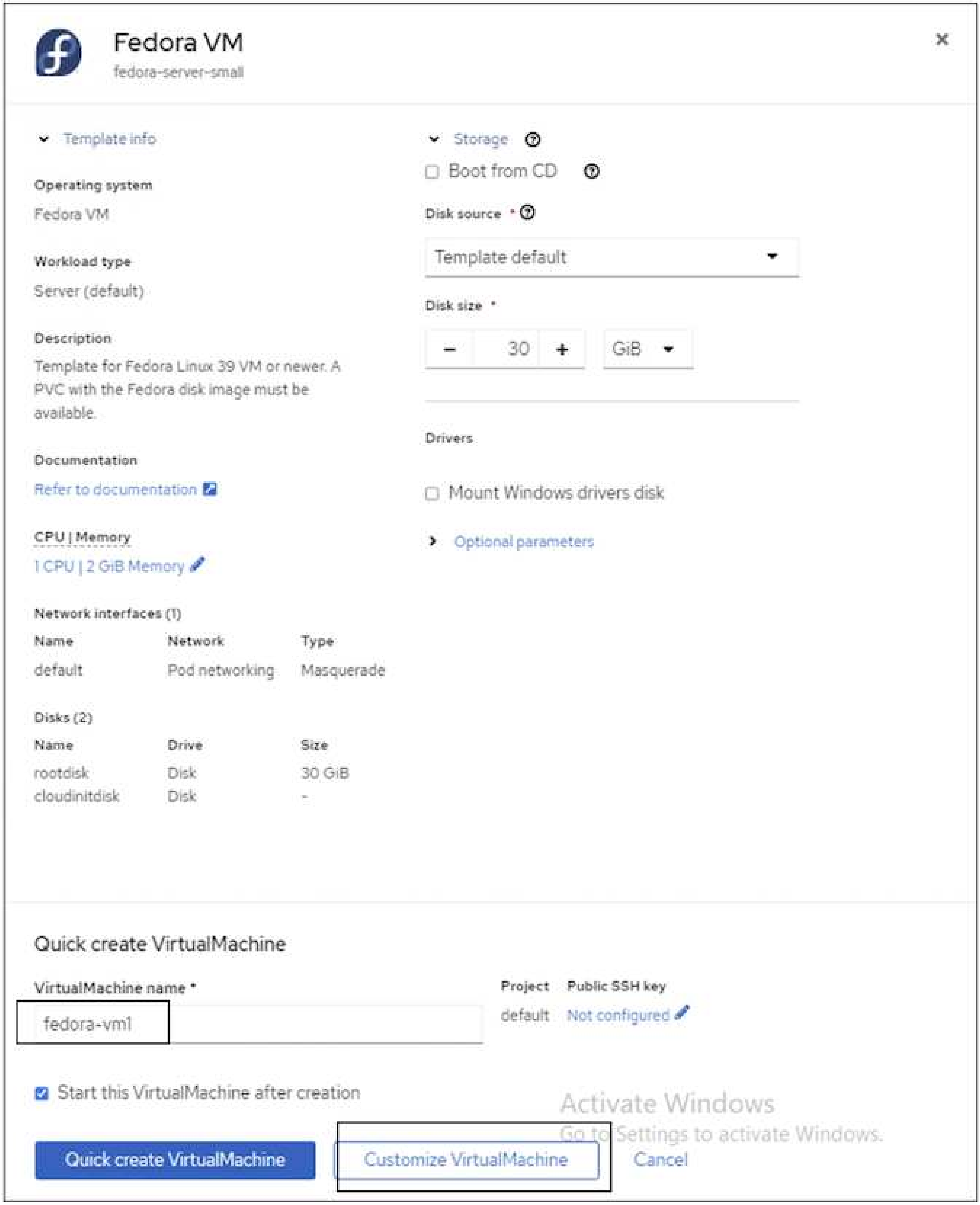
Create a VM from the template

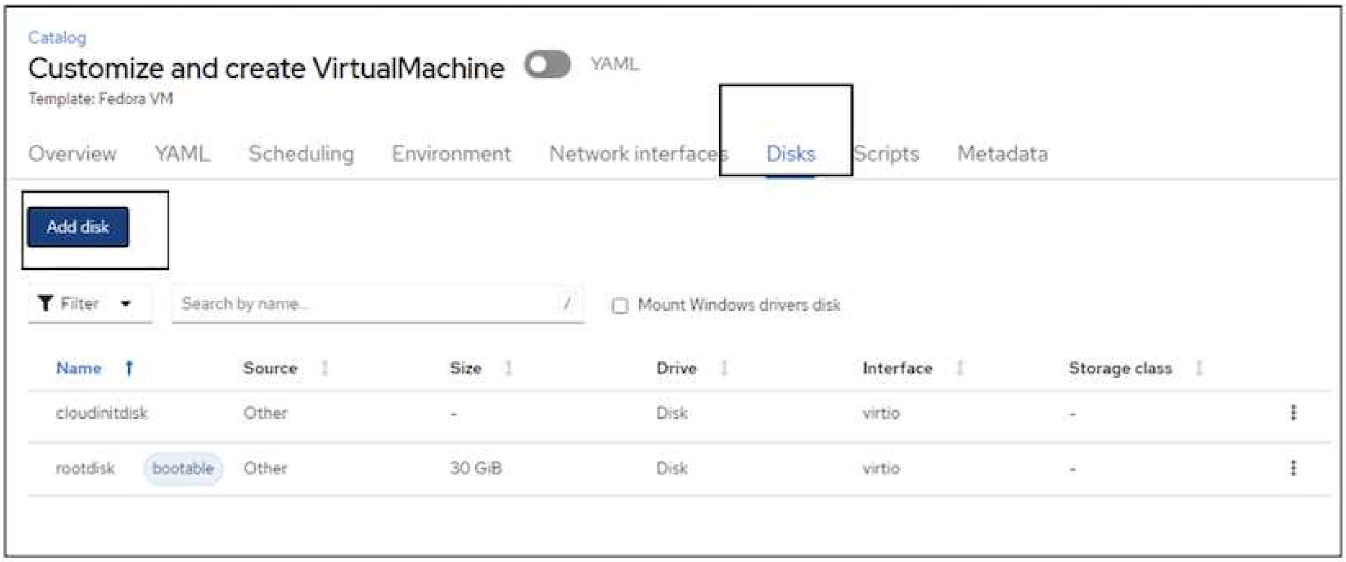
Use the web console to create a VM from a template.
From the RedHat OpenShiftService on AWS console, create a virtual machine. There are templates available on the cluster that can be used to create the VM.
In the screenshot below, we choose fedora VM from this list. Give the VM a name,and then click on **Customize Virtual Machine**. Select the **Disks** tab and click on **Add disks**.
Change the name of the disk preferably to something meaningful, ensure that **trident-csi** is selected for storage class. Click on **Save**. Click on **Create VirtualMachine**

After a few minutes, the VM is in the running state

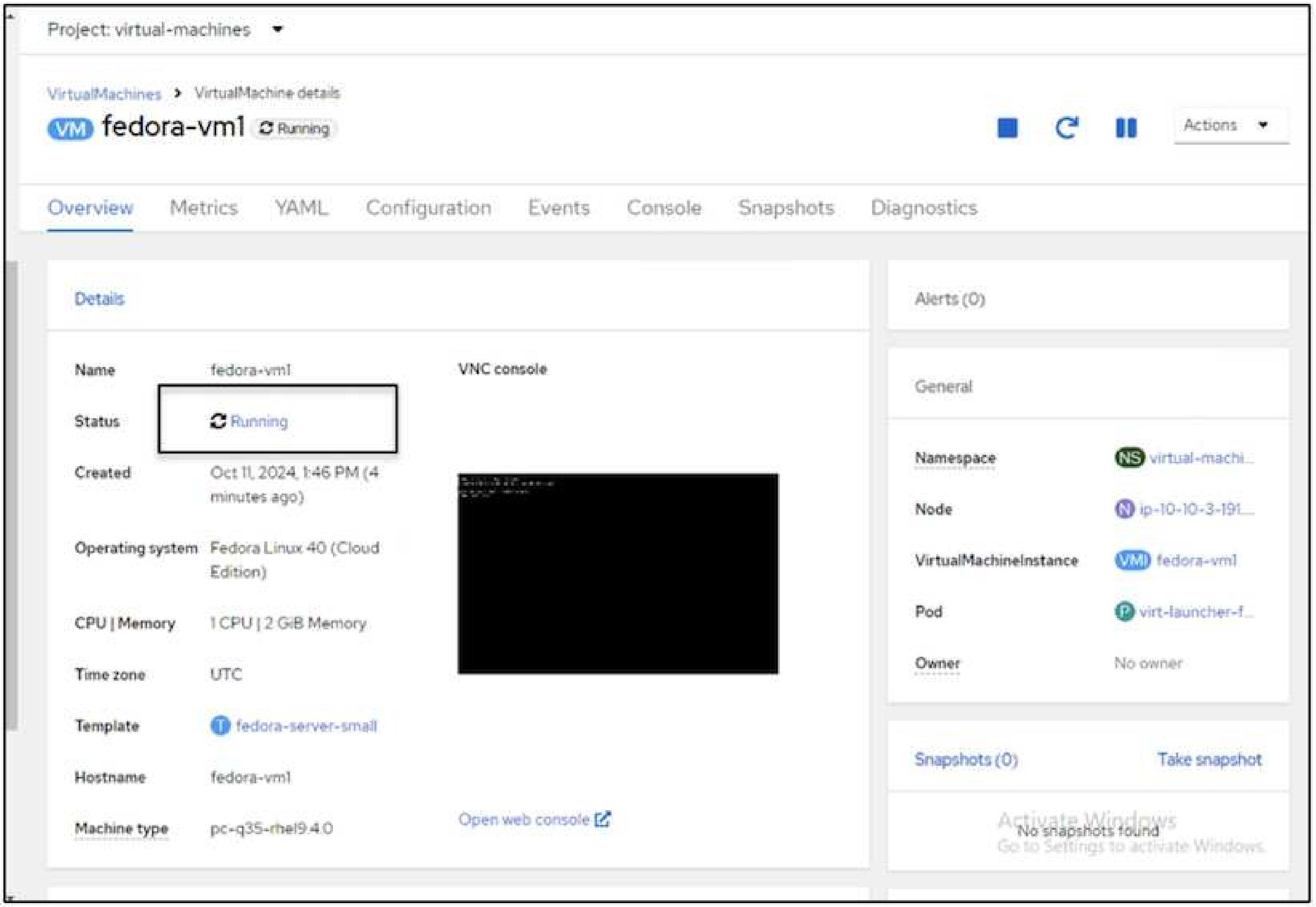



Red Hat
OpenShift Service on AWS
Exchange Password Required
Administrator
Home
Operators
Workloads
Virtualization
Networking
Storage
Builds
Observe
Compute
User Management
Administration
Project: virtual-machines
Create new VirtualMachine
Select an option to create a VirtualMachine from
InstanceTypes
Template catalog
Template project
All projects
All templates
Default templates
User templates
Boot source available
Operating system
CentOS
Fedora
Other
RHEL
Windows
Workload
Desktop
High performance
Server
Default templates
Filter by keyword...
12 items
CentOS Stream 8 VM
centos-stream8-server-small
CentOS Stream 9 VM
centos-stream9-server-small
CentOS 7 VM
centos7-server-small
Fedora VM
fedora-server-small
Red Hat Enterprise Linux 7 VM
rhel7-server-small
Red Hat Enterprise Linux 8 VM
rhel8-server-small
Red Hat Enterprise Linux 9 VM
rhel9-server-small
Microsoft Windows 10 VM
windows10-desktop-medium


Fedora VM
fedora-server-small
Template info
Operating system
Fedora VM
Workload type
Server (default)
Description
Template for Fedora Linux 39 VM or newer. A PVC with the Fedora disk image must be available.
Documentation
Refer to documentation
CPU | Memory
1 CPU | 2 GiB Memory
Network interfaces (1)
Name
Network
Type
default
Pod networking
Masquerade
Disks (2)
Name
Drive
Size
rootdisk
Disk
30 GiB
cloudinitdisk
Disk
-
Storage
Boot from CD
Disk source
Template default
Disk size
30
GiB
Drivers
Mount Windows drivers disk
Optional parameters
Quick create VirtualMachine
VirtualMachine name *
fedora-vm1
Project
default
Public SSH key
Not configured
Start this VirtualMachine after creation
Activate Windows
Go to Settings to activate Windows.
Quick create VirtualMachine
Customize VirtualMachine
Cancel


Catalog
Customize and create VirtualMachine
YAML
Template: Fedora VM
Overview
YAML
Scheduling
Environment
Network interfaces
Disks
Scripts
Metadata
Add disk
Filter
Search by name...
Mount Windows drivers disk
Name
Source
Size
Drive
Interface
Storage class
cloudinitdisk
Other
-
Disk
virtio
-
rootdisk
bootable
Other
30 GiB
Disk
virtio
-


Add disk
Use this disk as a boot source
Name *
fedora-vm1-disk1
Source *
Empty disk (blank)
PersistentVolumeClaim size *
30
GiB
Type
Disk
Hot plug is enabled only for "Disk" type
Interface *
VirtIO
Hot plug is enabled only for "SCSI" interface
StorageClass
SC trident-csi
Save
Cancel



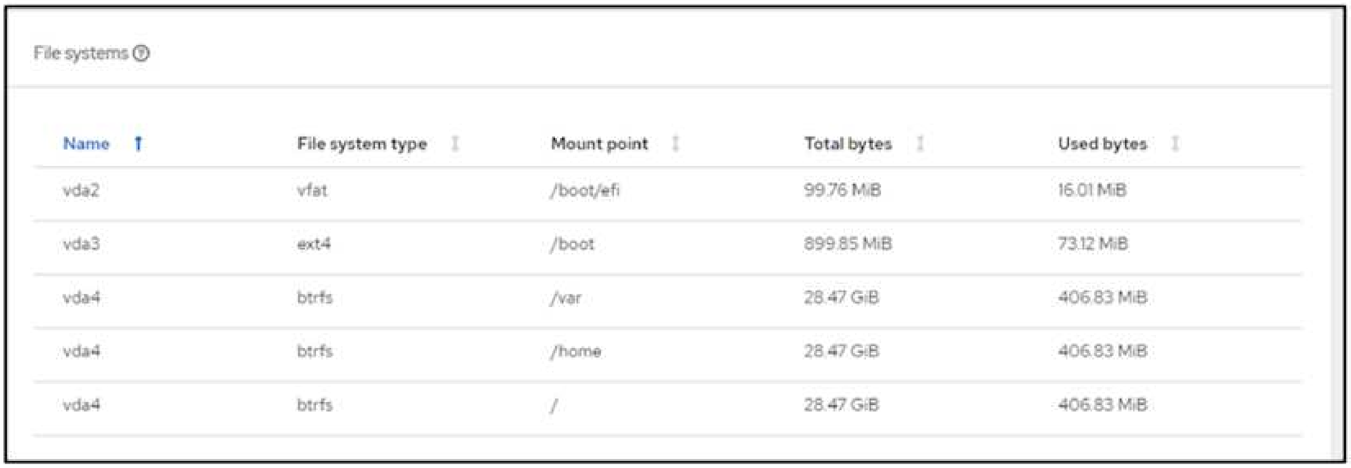
Review all the objects created for the VM

The storage disks.

Storage (3)

Name	Drive	Size	Interface
rootdisk	Disk	31.75 GiB	virtio
cloudinitdisk	Disk	-	virtio
fedora-vm1-disk1	Disk	31.75 GiB	virtio

The file systems of the VM will show the Partitions, File system type and the Mount points.

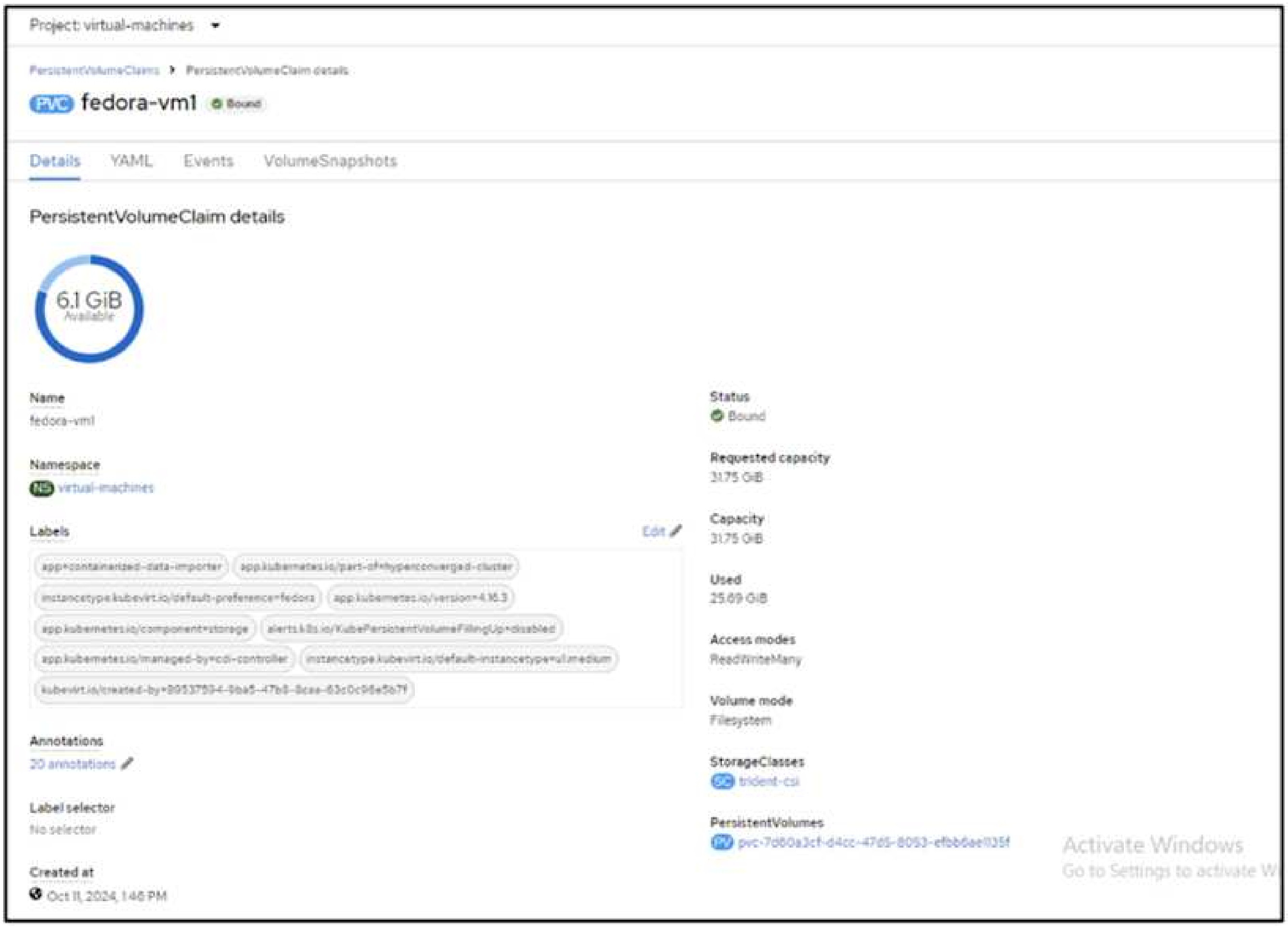

File systems

Name	File system type	Mount point	Total bytes	Used bytes
vda2	vfat	/boot/efi	99.76 MiB	16.01 MiB
vda3	ext4	/boot	899.85 MiB	73.12 MiB
vda4	btrfs	/var	28.47 GiB	406.83 MiB
vda4	btrfs	/home	28.47 GiB	406.83 MiB
vda4	btrfs	/	28.47 GiB	406.83 MiB

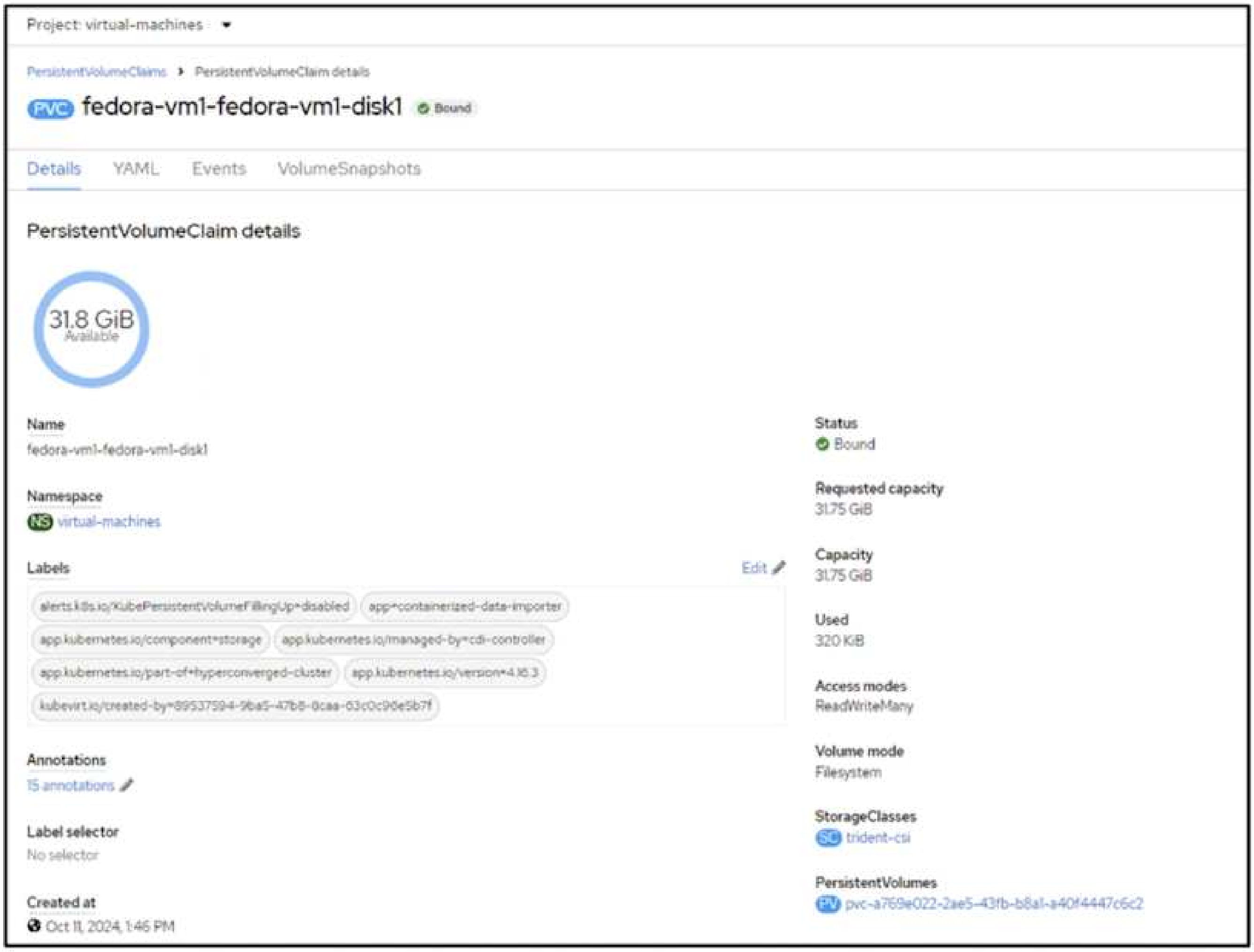
2 PVCs are created for the VM, one from the boot disk and one for the hot plug disk.

Name	Status	PersistentVolumes	Capacity
PVC fedora-vm1	Bound	PV pvc-7d60a3cf-d4cc-47d5-8053-efbb6ae1135f	31.75 GiB
PVC fedora-vm1-fedora-vm1-disk1	Bound	PV pvc-a769e022-2ae5-43fb-b8a1-a40f4447c6c2	31.75 GiB

The PVC for the boot disk shows that the Access mode is ReadWriteMany, and the Storage Class is trident-csi.



Similarly, the PVC for the hot-plug disk shows that the Access mode is ReadWriteMany, and the Storage Class is trident-csi.



In the screenshot below we can see that the pod for the VM has a Status of Running.

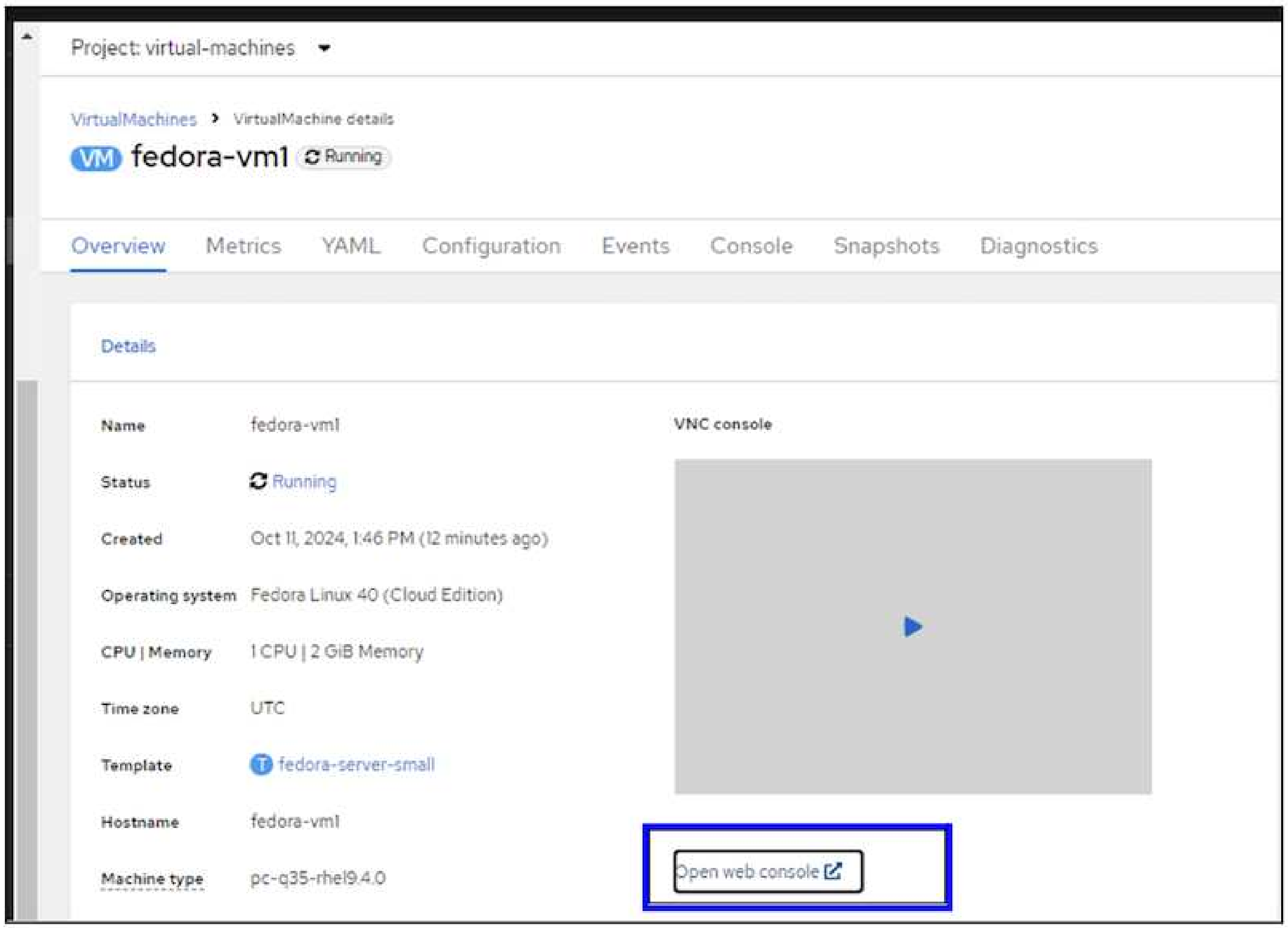
Here we can see the two Volumes associated with the VM pod and the 2 PVCs associated with them.

Volumes

Name	Mount path	SubPath	Type	Permissio...	Utilized by
private	/var/run/kubevirt-private	No subpath		Read/Write	compute
public	/var/run/kubevirt	No subpath		Read/Write	compute
ephemeral-disks	/var/run/kubevirt-ephemeral-disks	No subpath		Read/Write	compute
container-disks	/var/run/kubevirt/container-disks	No subpath		Read/Write	compute
libvirt-runtime	/var/run/libvirt	No subpath		Read/Write	compute
sockets	/var/run/kubevirt/sockets	No subpath		Read/Write	compute
rootdisk	/var/run/kubevirt-private/vmi-disks/rootdisk	No subpath	PVC fedora-vm1	Read/Write	compute
fedora-vm1-disk1	/var/run/kubevirt-private/vmi-disks/fedora-vm1-disk1	No subpath	PVC fedora-vm1-fedora-vm1-disk1	Read/Write	compute
hotplug-disks	/var/run/kubevirt/hotplug-disks	No subpath		Read/Write	compute

Connect to the VM

Click on ‘Open web console’ button, and login using Guest Credentials

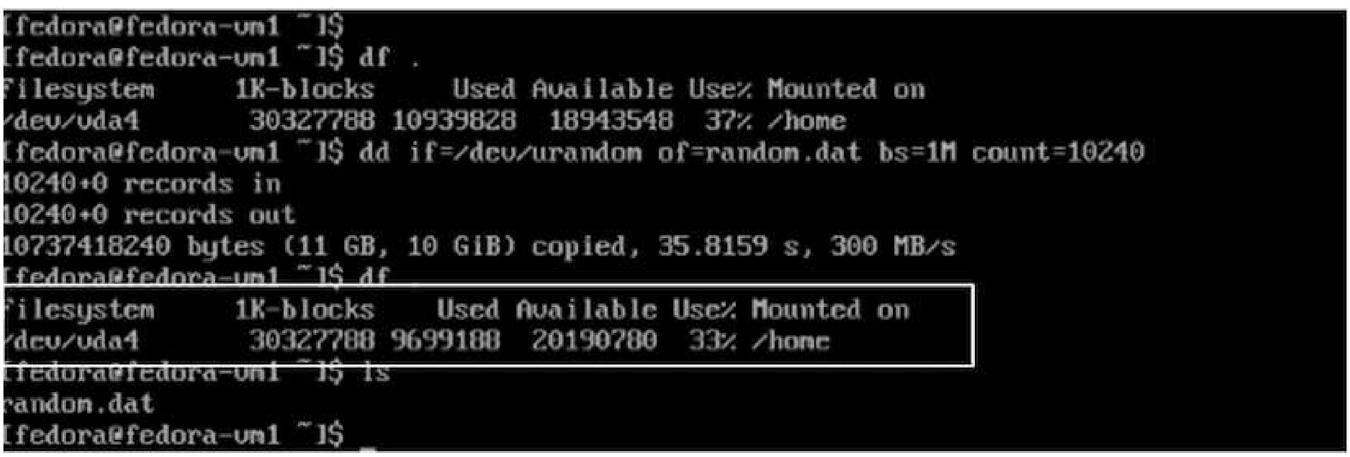


Issue the following commands

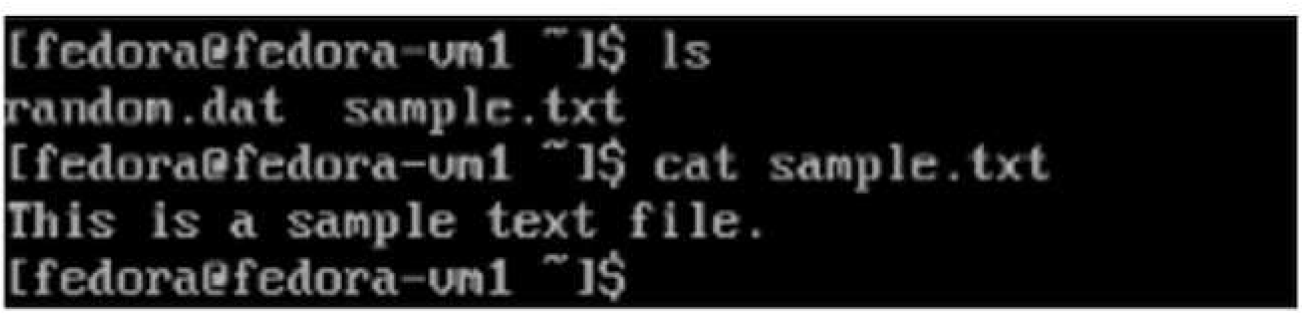
```
$ df (to display information about the disk space usage on a file system).
```

```
$ dd if=/dev/urandom of=random.dat bs=1M count=10240 (to create a file
called random.dat in the home dir and fill it with random data).
```

The disk is filled with 11 GB of data.



Use vi to create a sample text file that we will use to test.

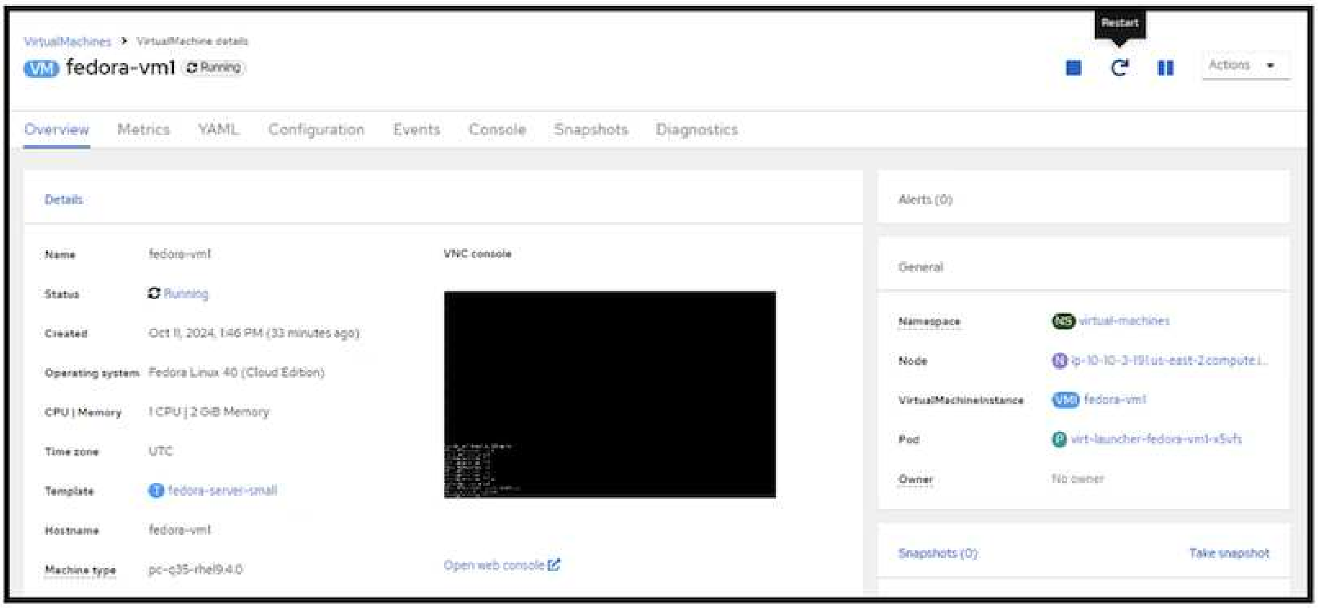


Workflows

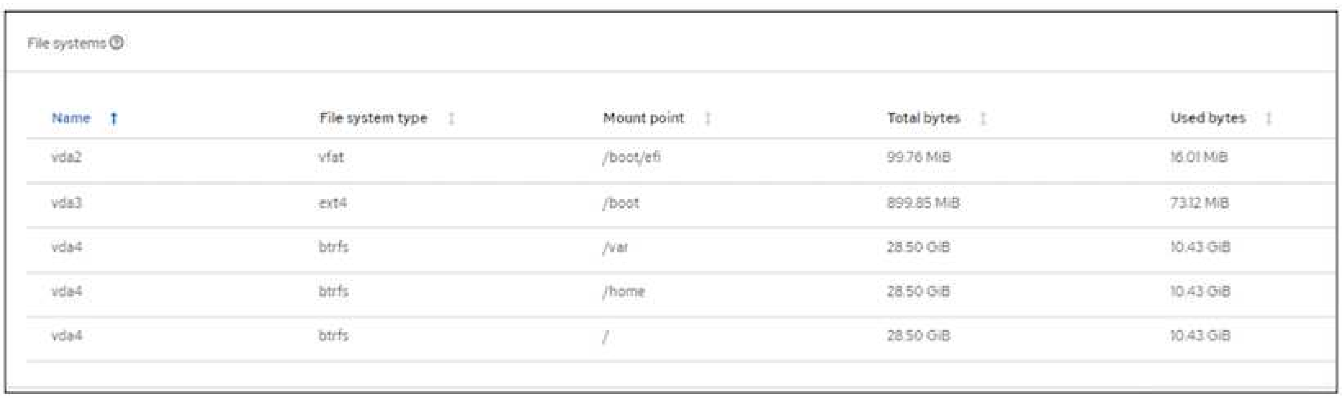
VM Restart

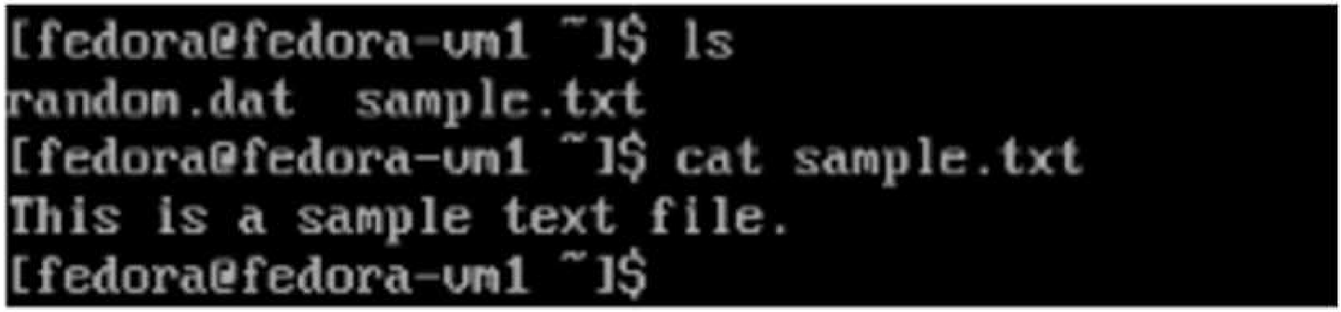
In this sections we will perform a VM restart and then examine the contents of the disks.

Click on the restart button.



The VM comes back to the running state with the exact same filesystems, PVCs and files in the filesystems



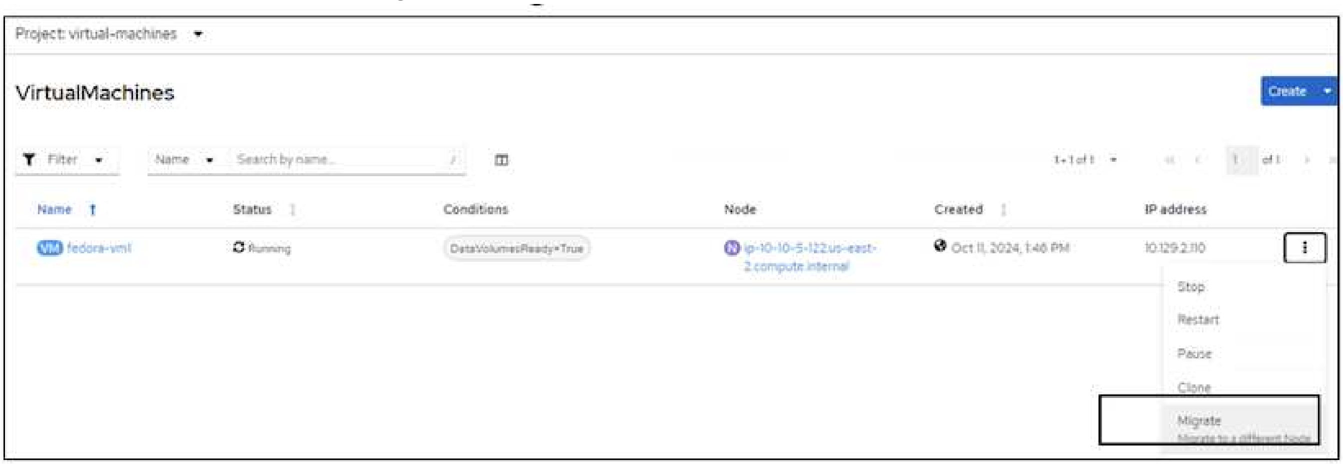


VM live migration

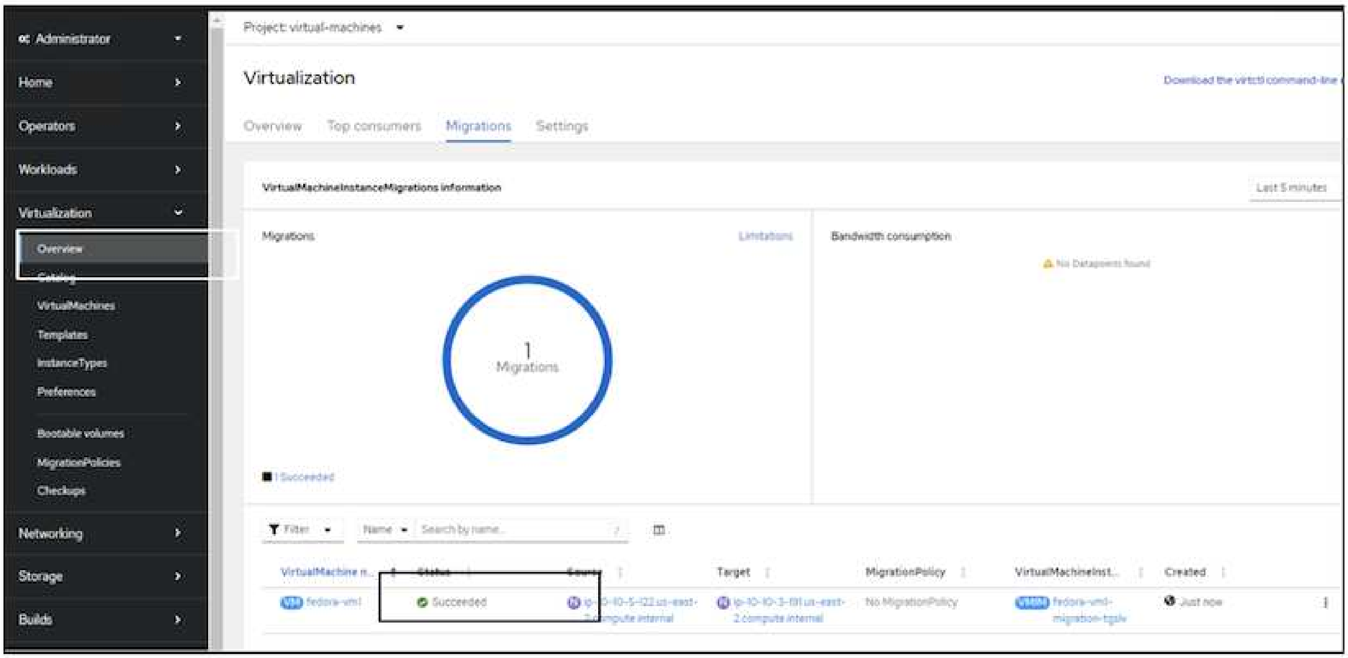
In this sections we will perform a VM live migration and then examine the contents of the disks.
Live migration refers to the process of moving a running Virtual Machine (VM) from one physical host to another host without disrupting normal operations or causing any downtime, or other adverse effects for the end user.
Live migration is considered a major step in Virtualization. It allows an entire VM to be moved with a running operating system (OS), Memory, storage, and network connectivity from their current node to the destination.
Below we will see how to perform a Live Migration of the VM from the current node to a new node.

Note the node on which the VM is running

Click on the 3 dots and select Migrate

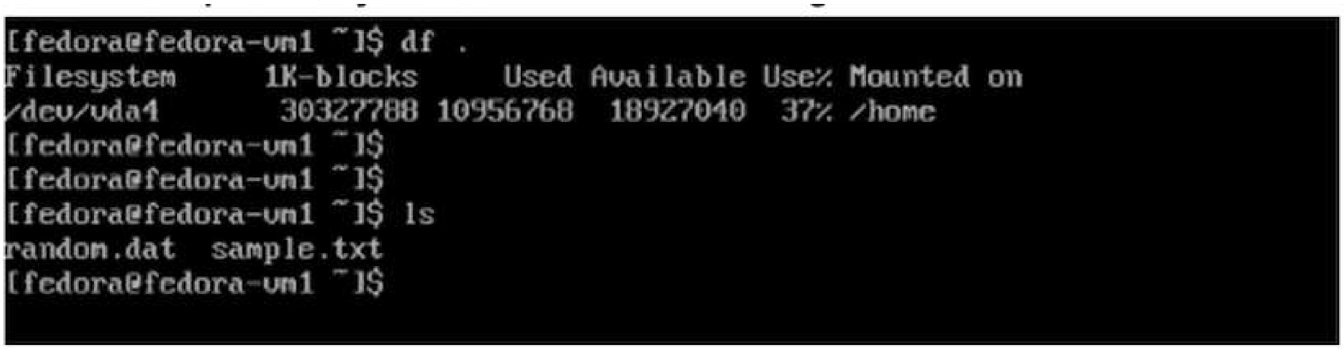


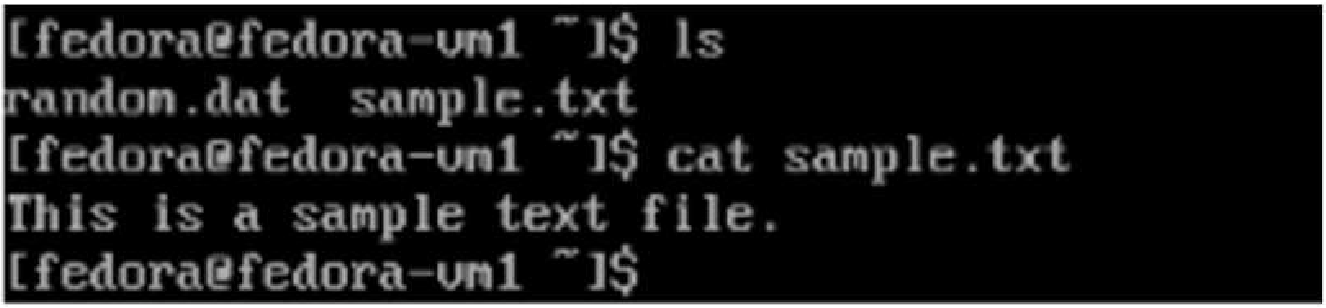
On the Overview page, you can see that the migration has succeeded, and the Status has changed to Succeeded.



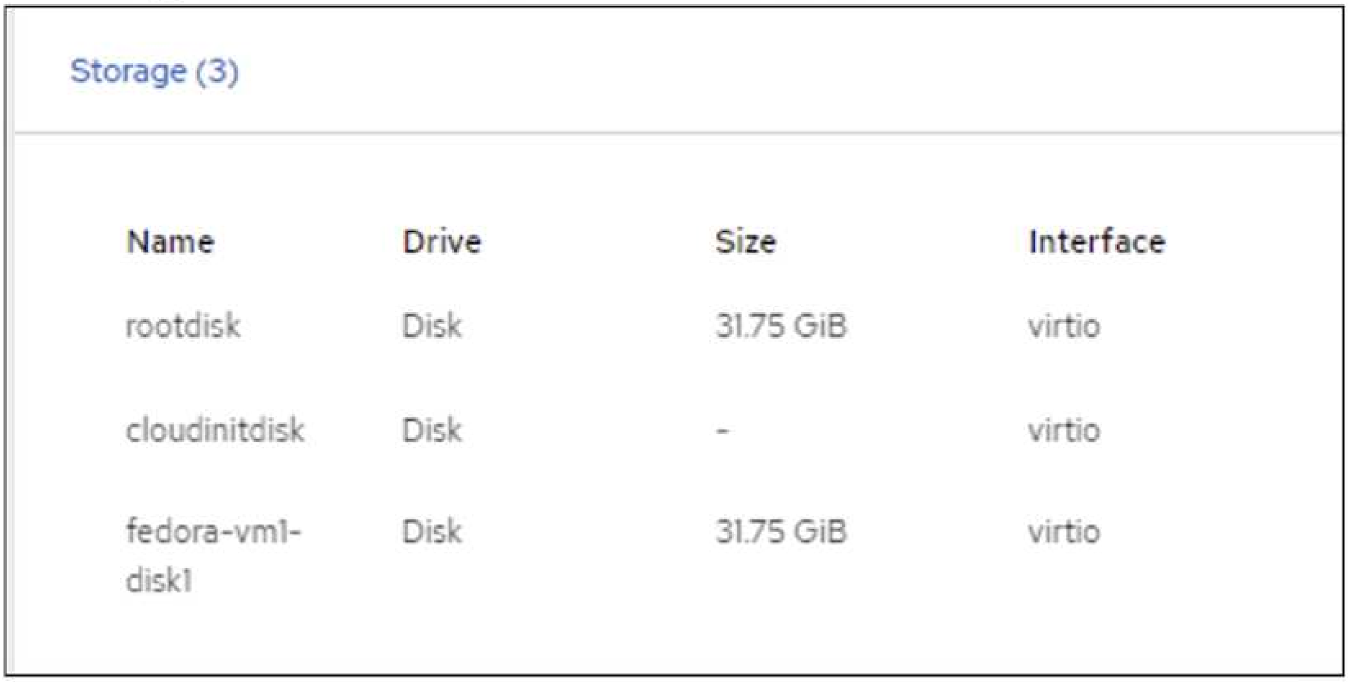
After the completion of the Live Migration, the VM is now on a different node.

Open the web console and view the contents of the disks. It still has the same 2 files that we previously created before the Live Migration.





The storage for the VM on the new node still shows the same disks



Also, the PVCs are the same.

Volumes associated with the VM pod are also the same (2 PVCs) as before.

Volumes

Name	Mount path	SubPath	Type	Permissio...	Utilized by
private	/var/run/kubevirt-private	No subpath		Read/Write	compute
public	/var/run/kubevirt	No subpath		Read/Write	compute
ephemeral-disks	/var/run/kubevirt-ephemeral-disks	No subpath		Read/Write	compute
container-disks	/var/run/kubevirt/container-disks	No subpath		Read/Write	compute
libvirt-runtime	/var/run/libvirt	No subpath		Read/Write	compute
sockets	/var/run/kubevirt/sockets	No subpath		Read/Write	compute
rootdisk	/var/run/kubevirt-private/vmi-disks/rootdisk	No subpath	PVC fedora-vm1	Read/Write	compute
fedora-vm1-disk1	/var/run/kubevirt-private/vmi-disks/fedora-vm1-disk1	No subpath	PVC fedora-vm1-fedora-vm1-disk1	Read/Write	compute
hotplug-disks	/var/run/kubevirt/hotplug-disks	No subpath		Read/Write	compute

Demo video

Live migration of virtual machines in OpenShift Virtualization on ROSA with Amazon FSx for NetApp ONTAP

More videos on Red Hat OpenShift and OpenShift Virtualization solutions can be found here.