ibm.com/redbooks

A Practical Guide to IBM Storage FlashSystem and Red Hat OpenShift Integration

Red Hat OpenShift Container Platform Virtualization Use Case

Vasfi Gucer

Charlotte Despres

Gayathri Gopalakrishnani

Sergei Kubin

Pravin Kudav

Keigo Matsubara

Harald Seipp

Gauthier Siri

Thomas Vogel

Storage

Infrastructure Solutions

Redpaper

IBM Redbooks

A Practical Guide to IBM Storage FlashSystem and Red Hat OpenShift Integration

August 2025

REDP-5753-00

Note: Before using this information and the product it supports, read the information in “Notices” on page v.

First Edition (August 2025)

This edition applies to IBM Container Storage Interface (CSI) driver 1.12.2, Red Hat OpenShift 4.16.25, and Kubernetes 1.29.10.

© Copyright International Business Machines Corporation 2025. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

© Copyright IBM Corp. 2025.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

© Copyright IBM Corp. 2025.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at https://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

Concert®	IBM Consulting™	Redbooks®
FlashCopy®	IBM FlashSystem®	Redbooks (logo) ®
IBM®	IBM Spectrum®	Turbonomic®
IBM Cloud Pak®	Instana®	

The following terms are trademarks of other companies:

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

ITIL is a Registered Trade Mark of AXELOS Limited.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Red Hat, Fedora, OpenShift, are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

VMware, and the VMware logo are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.

Other company, product, or service names may be trademarks or service marks of others.

Preface

Red Hat OpenShift Container Platform is a leading enterprise-grade container orchestration platform that is designed for the rapid deployment of web applications, databases, and microservices. As a Container Orchestration platform as a service (PaaS), it uses open industry standards like CRI-O and Kubernetes, you can focus on code while the platform manages complex IT operations.

This IBM Redpaper publication guides you in understanding, implementing, and managing IBM FlashSystem® storage with Red Hat OpenShift, specifically for Red Hat OpenShift Container Platform Virtualization, by using a real-life case study.

This paper is for storage administrators, IT architects, Red Hat OpenShift technical specialists, and cloud architects like you who deploy and manage containerized applications on IBM FlashSystem storage in a Red Hat OpenShift environment.

Authors

This IBM Redpaper publication was created by a team of specialists from around the world.

Vasfi Gucer leads projects for the IBM Redbooks® team, using his 30 years of experience in systems management, networking, and software. He is a prolific writer and global IBM® instructor whose focus shifted to storage and cloud computing over the past 10 years. Vasfi holds multiple certifications, including IBM Certified Senior IT Specialist, PMP, ITIL V2 Manager, and ITIL V3 Expert.

Charlotte Despres is an IT specialist with over 12 years of experience at IBM France. She currently works in IBM Client Engineering, where she focuses on demonstrating cutting-edge solutions to clients through proofs of concept (PoCs) and Minimum Viable Products (MVPs). Her expertise includes Red Hat OpenShift and virtualization technologies such as VMware, PowerVC, and more recently, Red Hat OpenShift Virtualization. In her previous role as an IT Team Leader in IBM Consulting™, she helped create and operate a private cloud solution that is tailored for IBM Consulting teams, providing on-demand environments to support client-focused development initiatives.

Gayathri Gopalakrishnani is an IT architect at IBM India with over 23 years of experience as a technical solution specialist, primarily in consulting. A results-driven leader, she has extensive experience in managing, designing, developing, implementing, and testing IT solutions. She is recognized for applying high-impact technical solutions to major business objectives, and her capabilities transcend boundaries. She is adept at collaborating with management to prioritize activities, achieve defined project objectives, and convert business requirements into technical solutions.

Sergei Kubin is a senior storage support engineer who is based in GBM Qatar. He holds an electronics engineering degree from Ural Federal University in Russia and has more than 15 years of experience in IT. At GBM, he provides support and guidance for customers that use IBM and multi-vendor storage solutions. His expertise includes file storage, block storage, and storage area networks.

© Copyright IBM Corp. 2025.

Pravin Kudav is an IBM storage solution architect with IBM Systems Labs. He works with various leading clients at IBM India. Pravin has over 15 years of solution architecture and design across diverse storage infrastructures. His current focus is on software-defined storage (SDS) solutions that use IBM Spectrum® Scale. He holds a master's degree in computer applications from Madurai Kamaraj University.

Keigo Matsubara is a technical sales specialist who works with various storage products, including IBM Storage FlashSystem and IBM Storage Insights. He has 33 years of industry experience. He currently focuses on IBM Storage FlashSystem and IBM Storage Insights. He joined IBM Japan in 1992 as a programmer and later migrated to his current role.

Harald Seipp is a senior technical staff member and principal solution architect with IBM Client Engineering Storage EMEA.

He is responsible for creating complex storage cloud and SDS architectures and for providing solutions to meet with challenging client requirements. As a worldwide leader in Kubernetes, Red Hat OpenShift, and object storage solutions, he implements first-of-a-kind solutions in co-creation with clients during proof-of-experience pilot engagements.

In prior job roles, he was the founder and technical leader of the Cloud Storage Center of Excellence within the IBM Systems EMEA Storage Competence Center. He also served as lead architect for IBM's primary host-based tape diagnostic tool, co-inventor of an IBM storage product, contributor to a Cloud Object Storage research project, and software developer for more than 15 years at IBM and two other companies. He regularly speaks at international technical conferences and holds various patents in storage and networking technologies.

Gauthier Siri is a platform engineer at IBM Client Engineering EMEA, based in Montpellier, France. He started his career 18 years ago as part of the IBM Worldwide Education Team, where he led the storage team and automated hardware environments to support global training programs. He then migrated to the storage solutions division, delivering benchmarks and creating storage demonstrations to showcase IBM Storage value to customers. Now part of a pre-sales engineering team, Gauthier focuses on developing assets and performing PoCs and pilots for automation products such as IBM Instana®, IBM Turbonomic®, and IBM Concert®. His natural curiosity drives him to explore new technologies, from automation to containerization, supporting solutions such as IBM Storage for Red Hat OpenShift Container Platform and Virtualization.

Thomas Vogel is a consulting IT specialist with the EMEA Advanced Technical Support organization at IBM Systems Germany. His areas of expertise include solution design, storage virtualization, storage hardware and software, educating technical sales teams and IBM Business Partners, and designing distributed high availability and disaster recovery (HADR) solutions. For the last 16 years, he has been designing and selling solutions for IBM Storage Virtualize and IBM FlashSystem, and assisting with customer performance and problem analysis. He holds a degree in electrical engineering and achieved VMware VCP certification.

Thanks to the following people for their contributions to this project:

Luiz Gabriel Costa
IBM US

Ariel Kass
IBM Israel

Mathias Defiebre, Axel Koester, Klaus Kuehn
IBM Germany

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an IBM Redbooks® residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, IBM Redbooks
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on LinkedIn:

  https://www.linkedin.com/groups/2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/subscribe

- Stay current on recent Redbooks publications with RSS Feeds:

  https://www.redbooks.ibm.com/rss.html

1

Introduction

Modern digital transformation initiatives require robust, scalable, and flexible infrastructure solutions that support containerized applications and ensure enterprise-grade storage capabilities. Integration of Red Hat OpenShift Container Platform with IBM Storage FlashSystem provides you with a robust foundation for building and deploying next-generation applications.

This chapter describes the key components of this integration, with a specific focus on the Red Hat OpenShift Container Platform Virtualization use case. You explore the fundamental concepts, benefits, and technical considerations that IT professionals, storage administrators, DevOps engineers, and cloud architects need to understand when implementing this solution.

This chapter contains the following sections:

© Copyright IBM Corp. 2025.

Important: This Redpaper focuses on virtualization within the Red Hat OpenShift Container Platform, specifically using Red Hat OpenShift Virtualization. You receive guidance on provisioning, configuring, managing, and migrating virtual machines (VMs) within Red Hat OpenShift. Additionally, it covers how you integrate of VMs and containers on a unified platform. This Redpaper helps you as an IT professional, including virtualization administrators and developers, manage both VMs and containers on Red Hat OpenShift.

Using the IBM Block Storage CSI Driver in a Red Hat OpenShift Environment, REDP-5613 primarily discusses how you can integrate IBM block storage with Red Hat OpenShift using the Container Storage Interface (CSI) driver. You explore topics such as container persistent data challenges, Red Hat OpenShift Container Storage, IBM Cloud Pak®, and deployment approaches for the CSI driver. This Redpaper helps you, as a storage administrator, IT architect, or Red Hat OpenShift technical specialist, integrate IBM Enterprise storage with Red Hat OpenShift.

In summary, if you want to integrate IBM Block Storage with Red Hat OpenShift using the CSI driver, see *Using the IBM Block Storage CSI Driver in a Red Hat OpenShift Environment*, REDP-5613 . If you are exploring virtualization within Red Hat OpenShift through Red Hat OpenShift Virtualization, this Redpaper is especially valuable to you.

1.1 Introducing Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform is an enterprise-ready Kubernetes container platform that provides full-stack automated operations for managing hybrid cloud and multi-cloud deployments. You use Red Hat OpenShift Container Platform to manage hybrid cloud, public cloud, and edge deployments on a consistent application platform.

1.1.1 Key features of Red Hat OpenShift

Key features of Red Hat OpenShift include:

- Enterprise Kubernetes: You use enterprise Kubernetes built on the foundation of Kubernetes, with added features for enhanced security, application lifecycle management, and developer productivity.
- Developer-friendly workflows: You benefit from integrated CI/CD pipelines, source-to-image deployment, and developer tools that simplify container development.
- Automated operations: You simplify installation, upgrades, and lifecycle management for both the platform and applications.
- Multi-cluster management: You manage multiple Red Hat OpenShift clusters across various environments.
- Extensible platform: You extend platform capabilities and manage complex stateful applications using the Operator framework.

1.1.2 Red Hat OpenShift Container Platform Virtualization

Red Hat OpenShift Container Platform Virtualization extends Red Hat OpenShift's capabilities so you can run and manage VMs alongside containers. You can migrate and modernize traditional workloads that are not easily containerized, using a unified platform for both container-native and VM-based workloads.

Key capabilities of Red Hat OpenShift Container Platform Virtualization include:

- Run VMs alongside containers: You operate both workloads on the same platform to simplify infrastructure and improve resource utilization.
- Manage containers and VMs with Kubernetes-native tools: You use a unified interface for consistent operations across environments.
- Migrate traditional workloads: You streamline migration using a simplified path for existing applications.
- Apply consistent security and networking policies: You enforce uniform policies across container and VM workloads.
- Manage VM lifecycles: You control VM lifecycles through Kubernetes custom resources (CRs).

1.2 Introducing IBM Storage FlashSystem

IBM Storage FlashSystem is a family of all-flash and hybrid storage solutions that deliver enterprise-grade performance, reliability, and functionality. You use IBM FlashSystem as a foundation for modern data infrastructure that supports the demanding requirements of containerized and virtualized environments.

> **Note:** For more information about IBM Storage FlashSystem, see *Implementation Guide for IBM Storage FlashSystem and IBM SAN Volume*, SG24-8542.

1.2.1 Key features of IBM Storage FlashSystem

Key features of IBM Storage FlashSystem include:

- High performance: You benefit from an all-flash architecture designed for consistent low-latency performance.
- Enterprise reliability: You use IBM Spectrum Virtualize software with advanced availability and data protection features.
- Scalability: You can scale from entry-level to enterprise-class deployments.
- Data efficiency: You achieve inline data reduction through compression, deduplication, and thin provisioning.
- Advanced capabilities: You use replication, snapshots, encryption, and tiering to enhance storage operations.
- Storage virtualization: You virtualize existing storage systems to extend infrastructure flexibility.
- Integration with cloud environments: You support hybrid cloud deployments for modern workloads.

1.2.2 IBM FlashSystem storage for containerized environments

IBM FlashSystem storage offers specific capabilities that make it well-suited for containerized environments:

- CSI support: You use the IBM Block Storage CSI driver to integrate with container orchestration platforms.
- Persistent storage: You provide reliable storage for stateful containerized applications.
- Kubernetes integration: You integrate with Kubernetes storage primitives for seamless orchestration.
- Enterprise-grade data services: You deliver robust data protection and availability for containerized workloads.
- Automation capabilities: You automate operations through RESTful APIs.

1.3 Benefits of IBM FlashSystem and Red Hat OpenShift integration

The integration of IBM FlashSystem storage with Red Hat OpenShift Container Platform offers key benefits when you deploy containerized applications, especially in environments that use Red Hat OpenShift Container Platform Virtualization. You gain enterprise-grade storage performance, reliability, and scalability to support both container-native and VM-based workloads on a unified platform.

1.3.1 Operational benefits

Operational benefits include:

- Unified management: You apply a consistent approach to storage provisioning and management across containers and VMs.
- Simplified infrastructure: You reduce complexity through integrated storage and container orchestration.
- Increased agility: You rapidly provision persistent storage for containerized and virtualized workloads.
- Enhanced automation: You integrate storage operations into CI/CD pipelines and DevOps workflows.
- Improved resource utilization: You use storage resources efficiently across container and VM workloads.

1.3.2 Technical benefits

Technical benefits include:

- Enterprise data services: You use advanced capabilities such as snapshots, cloning, and replication to support containerized applications.
- Performance optimization: You access low-latency persistent storage for performance-sensitive workloads.
- Data protection and availability: You ensure enterprise-grade reliability for mission-critical containerized applications.

- Scalability: You scale storage resources dynamically to meet application demands.
- Security: You apply integrated security features and tenant isolation to protect workloads.

1.3.3 Business benefits

Business benefits include:

- Accelerated application development: You streamline provisioning of development and test environments to speed up delivery cycles.
- Simplified migration path: You transition more easily from traditional to cloud-native applications.
- Cost optimization: You use storage resources more efficiently across environments to reduce operational costs.
- Reduced vendor complexity: You benefit from an integrated solution delivered by trusted enterprise vendors.
- Future-proof architecture: You support evolving application architecture patterns with flexible infrastructure.
- Threat detection capabilities: You enable Workload Anomaly Detection (WAD) or Ransomware Threat Detection (RTD) for volumes used for VMs.

Note: For more information about WAD and RTD, see *Intelligent Storage Management with Artificial Intelligence for IT Operations by using IBM Storage Insights*, REDP-5755*.*

1.4 Introducing Kubernetes storage and the Container Storage Interface

You must understand how Kubernetes manages storage and the role of the CSI to implement IBM FlashSystem storage with Red Hat OpenShift.

1.4.1 Kubernetes storage basics

Kubernetes provides several abstractions for managing storage:

- Persistent volumes (PVs): Cluster-wide storage resources that are provisioned by an administrator or dynamically provisioned using StorageClasses
- Persistent volume claims (PVCs): Requests for storage resources by users or applications
- Storage Classes: Definitions of different classes of storage with specific quality-of-service levels or policies
- Volume Snapshots: Point-in-time copies of volumes that you can use for backups or to create new volumes
- Volume Clones: New volumes created from existing volumes that retain all data

1.4.2 Container Storage Interface

The CSI is a standard that exposes block and file storage systems to containerized workloads in Kubernetes. CSI allows storage vendors to develop plugins that work across container orchestration systems without requiring in-tree code.

Key benefits of CSI include:

- Vendor neutrality: Provides a standardized interface that any storage vendor can implement.
- Plugin isolation: Runs storage plugins run as separate containers to enhance security and stability.
- Feature velocity: Enables new storage features without requiring changes to Kubernetes core code.
- Simplified maintenance: Allows storage plugins to be updated independently of Kubernetes.
- Extended functions: Supports advanced storage operations such as snapshots and cloning.

1.5 RWX and RWO access modes and IBM Block Storage CSI and Red Hat OpenShift Virtualization

When you provision persistent storage on Red Hat OpenShift by requesting a PVC, you must specify an access mode. Red Hat OpenShift ensures that the volume is accessible only with the specified access mode. A ReadWriteOnce (RWO) PVC allows read-write access only to containers running on the same node. In contrast, a ReadWriteMany (RWX) PVC enables simultaneous read-write access by multiple containers running on different nodes.

Characteristics of RWO volumes include:

- Single-node access: Can be mounted as read-write by only one node at a time.
- Application suitability: Ideal for most stateful applications where pods do not need to be scheduled across multiple nodes.
- Storage protocol: Typically implemented using block storage protocols like Fibre Channel (FC) or iSCSI.
- Use case alignment: Well-suited for databases and other applications that require exclusive access to storage.

Characteristics of RWX volumes include:

- Multi-node access: Can be mounted as read-write by multiple nodes simultaneously.
- Scalability: Essential for applications that need to scale horizontally and share data.
- Storage protocol: Traditionally implemented using file storage protocols such as NFS or object storage.
- Use case alignment: Important for workloads such as content management systems, AI/ML training data, and shared application state.

The access mode is closely tied to the volume mode, which determines how the resulting PV is presented to the container. If you specify `volumeMode: Block`, the PV is exposed as a raw block device. If you specify `volumeMode: Filesystem`, the PV is formatted with a file system and mounted to the container after the PVC gets bound to it.

Note: For block storage, various combinations of volumeMode and accessMode are supported, with one exception: you cannot create a PVC with volumeMode: Filesystem and accessMode: ReadWriteMany. The file systems used in this mode, such as `ext4 and XFS`, do not support simultaneous modifications from multiple nodes. Ignoring this limitation can lead to file system data corruption.

To support the RWX file system use case, you must use an NAS share or a distributed file system. IBM provides two solutions for this purpose: IBM Fusion Data Foundation CephFS and IBM Storage Scale Container Native Storage Access (CNSA). Both offer distributed file systems with full support for RWX file system access.

When you use IBM Block Storage CSI as the storage provider for Red Hat OpenShift Container Platform Virtualization VMs, the underlying disks are created as PVCs with the volumeMode set to `Block`.

With IBM Block Storage CSI versions less than 1.12.0, only RWO accessMode was supported for volumes with `volumeMode: Block`. Although you can create VMs, a significant limitation was the lack of support for VM Live Migration. VM Live Migration enables near-zero downtime when moving VMs across nodes in the Red Hat OpenShift cluster.

This capability is important because maintenance operations, such as Red Hat OpenShift upgrades, firmware updates, or hardware replacement, require downtime for a single node. In contrast, a highly available solution is expected to ensure that applications, including VMs, do not experience downtime.

You can achieve zero downtime during maintenance operations by moving all workload from the node under maintenance to adjacent nodes. Red Hat OpenShift performs this operation automatically during updates, using a rolling strategy that processes one node at a time.

IBM Block Storage CSI version 1.12.0 introduced support for RWX accessMode with `volumeMode: Block`, enabling temporary concurrent mounting of a PVC to two nodes. Red Hat OpenShift Container Platform Virtualization ensures that file systems are not written to simultaneously, preventing data corruption. This capability supports VM Live Migration, allowing VMs with operating system (OS) and supplemental disks provisioned by IBM Block Storage CSI to move across hosts without disrupting applications.

1.6 Infrastructure for Red Hat OpenShift with IBM FlashSystem

Implementing a successful Red Hat OpenShift environment with IBM FlashSystem storage requires careful planning of infrastructure components. This section describes configurations and best practices for deploying Red Hat OpenShift Container Platform Virtualization with IBM FlashSystem storage.

1.6.1 Compute infrastructure

Compute infrastructure requirements:

- Master nodes: A minimum of three master nodes is required for production environments to ensure high availability (HA).
- Worker nodes: Must be sized appropriately for the workload profile:
  - CPU: Modern processors with virtualization support (Intel VT-x or AMD-V).
  - Memory: Sufficient RAM to support both container and VM workloads (a best practice is minimum 64 GB per node).
  - Network: Minimum 10 Gbps networking, with 25 Gbps or higher as a best practice for production environments.

1.6.2 Network infrastructure

The network infrastructure must meet the following requirements:

- Cluster network: Must be separated from external network traffic and provide sufficient bandwidth.
- Storage network: Use a dedicated network for storage traffic, ideally separated from cluster communication.
- Network interfaces: Each node must have multiple network interfaces to separate different traffic types.
- Jumbo frames: Configure storage networks to support jumbo frames to optimize performance.
- Latency: Connectivity between compute nodes and storage systems must be low latency.
- Redundancy: Redundant paths between compute nodes and storage are required to ensure HA.

1.6.3 Storage infrastructure

Storage infrastructure requirements:

- IBM FlashSystem sizing: Must be appropriately sized for the workload profile.
  - Performance tier: All-flash storage for performance-sensitive workloads.
  - Capacity tier: Mixed flash and disk for capacity-oriented workloads.
  - Tiering strategy: Multiple tiers configured to support different service levels.
- HA:
  - Redundant storage controllers: Required for HA.
  - Redundant connectivity: Must be established between storage and compute nodes.
- Storage classes: Multiple storage classes defined to match different application requirements.
- Capacity planning: Sufficient capacity headroom for snapshots, clones, and unexpected growth.

1.6.4 Reference architectures

This section provides reference architectures that support different resiliency requirements to ensure high availability and disaster recovery (HADR):

- Single-site HA: Tolerates Red Hat OpenShift node failures.
- Two-site HA: Tolerates datacenter and storage system failures.
- Two-site data resiliency: Tolerate datacenter, storage system, and cluster/software failures.

Single-site HA

The single-site HA architecture is easy to set up and can tolerate the failure of a single worker node and control plane without interrupting the workload, as long as the remaining nodes can support the additional workload. When used with a backup and recovery solution such as IBM Fusion Backup and Restore, the architecture becomes more resilient. IBM Fusion stores backup data at an off-site location, helping protect against local failures. This combination provides a cost-efficient solution with Recovery Point Objective (RPO) and Recovery Time Objective (RTO) typically measured in hours.

Refer to Figure 1-1 for an example of a single-site HA architecture. This configuration includes three worker nodes. which meets the minimum requirement for workloads that require a three-node quorum. These workloads include highly available database deployments.

When a worker node outage occurs, the VMs running on the affected node are re-scheduled to one of the remaining worker nodes. This re-scheduling is successful if sufficient CPU and RAM resources are available to support the additional workload. The RWX access mode for the VM PVs helps reduce transition time. Because the PVs can be immediately bound to the target host, the workload resumes quickly. This happens even if the failing node still retains the volume mapping.

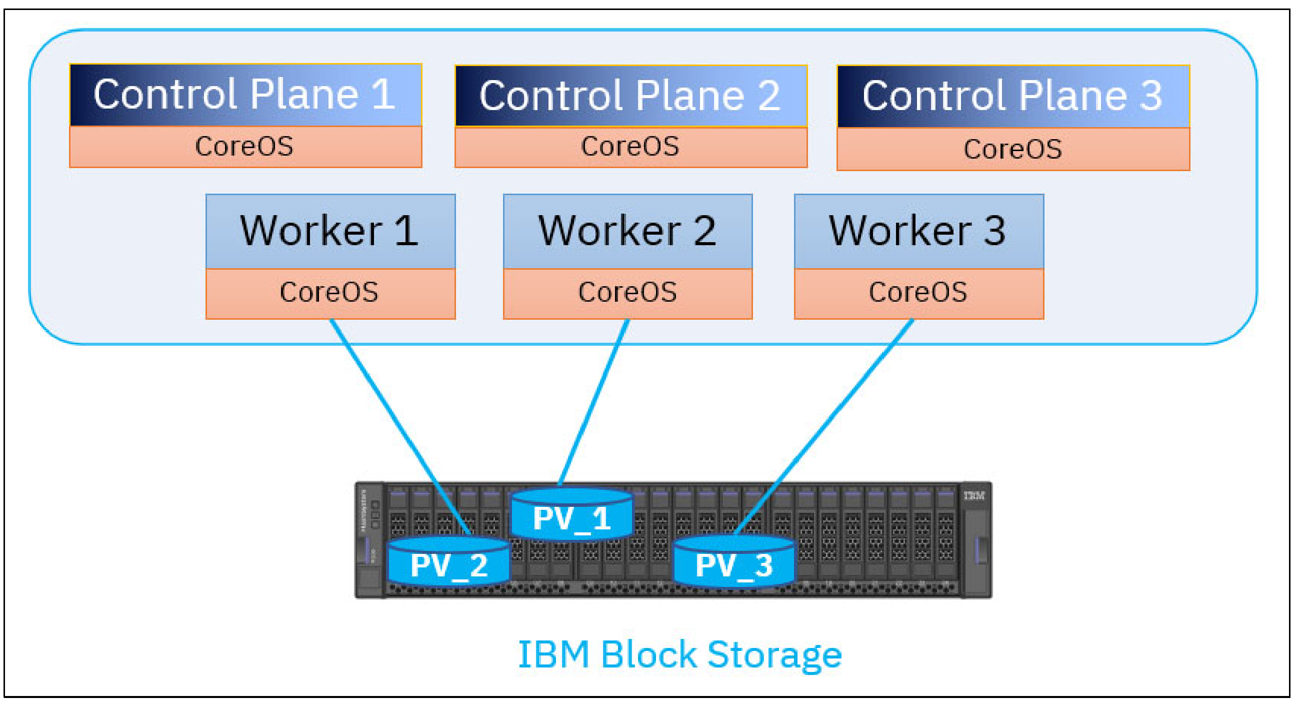

Figure 1-1 Single-site HA architecture

Two-site HA

The two-site HA architecture shown in Figure 1-2 assumes the presence of two datacenters within in metro-area distance or in different buildings on a single campus, labeled "Availability Zone 1" (AZ1) and "Availability Zone 2" (AZ2). A third zone, called "Arbiter Zone" is required for both Block Storage and the Red Hat OpenShift Cluster to reach quorum majority during a site failure or network partition incident. The Arbiter zone is typically a separate building from AZ1 and AZ2, or in a distinct fire zone within AZ1 or AZ2.

Two storage systems form a stretched cluster that uses synchronous replication across AZ1 and AZ2.

In addition to the resiliency of the single-site HA architecture, this design tolerates a complete availability zone outage with an RPO of zero and an RTO near zero. If the Red Hat OpenShift nodes in AZ1 or the storage system in AZ2 fail, VM workloads continue running without significant downtime because the VMs relocate to nodes in AZ2.

Because this solution uses a single Red Hat OpenShift cluster, a Red Hat OpenShift Container Platform cluster outage caused by misconfiguration or software errors poses a risk of significant downtime. To protect against this failure scenario, use a Backup and Recovery solution as described in "Single-site HA" on page 9 for the single-site HA architecture.

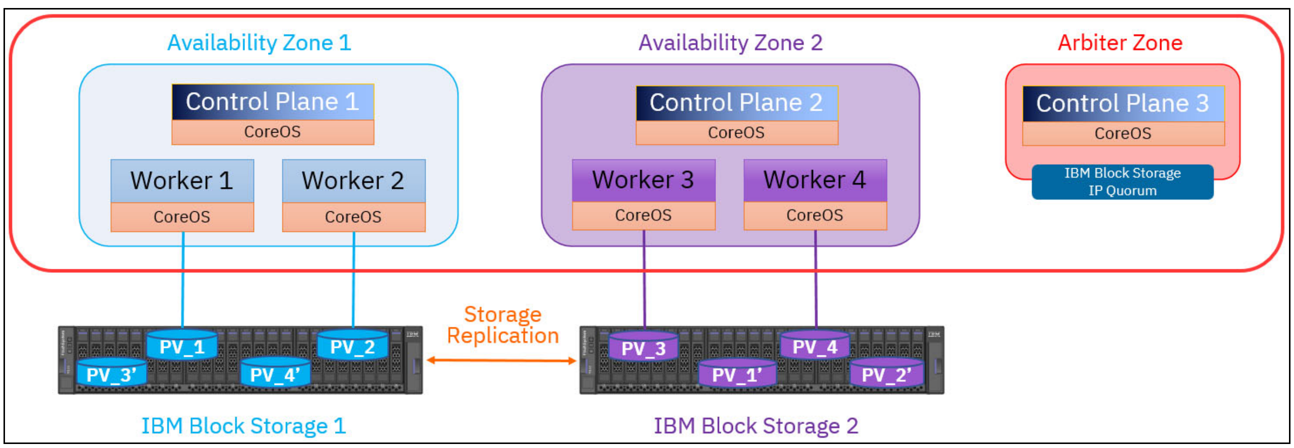

Figure 1-2 Two-site HA architecture

Two-site data resiliency

Figure 1-3 on page 11 illustrates a two-site data resiliency solution architecture with separate Red Hat OpenShift clusters at each datacenter site. To ensure resilience against regional disasters, the sites are located more than 100 km apart, which requires asynchronous storage replication between the storage systems at each site. In addition to storage replication, application metadata for persistent applications must be replicated to the secondary site. You typically deploy static application metadata from PVC upward using GitOps methods. For PV definitions and dynamically generated application metadata, you must establish a manual process to ensure all metadata is available at Site 2 if Site 1 fails. During the failover process, the application is started at Site 2.

This setup tolerates both a complete site failure and a Red Hat OpenShift cluster failure. The RPO and RTO are typically within a few minutes.

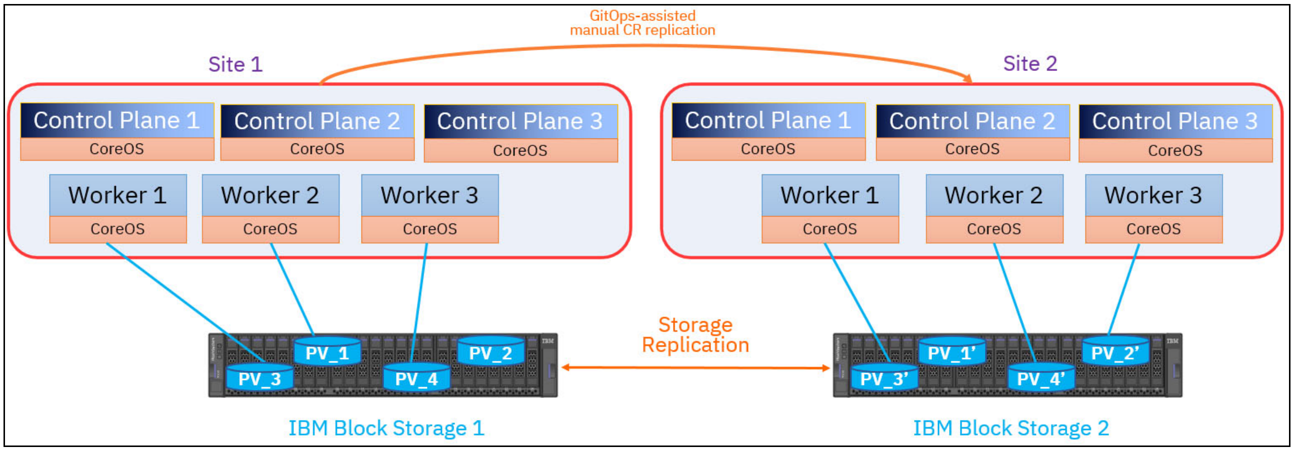

Figure 1-3 Two-site data resiliency architecture

1.7 Considerations

Although the integration of IBM FlashSystem with Red Hat OpenShift Container Platform Virtualization provides powerful capabilities, you must consider important limitations when designing and implementing solutions.

1.7.1 Persistent volume considerations

Key considerations for PVs and VMs in Red Hat OpenShift Container Platform Virtualization:

- PVs per storage system: Each IBM FlashSystem storage system supports a maximum number of volumes. which directly limits the number of available PVs.
- PV to VM mapping: VMs typically require at least one PV for the boot disk, with additional PVs allocated for data disks.
- VM disk sprawl: Unlike traditional virtualization platforms that use datastores, each VM disk in Red Hat OpenShift Container Platform Virtualization corresponds to a separate PV. This design can lead to PV proliferation.

Important: These considerations are especially important for organizations migrating from VMware environments, where the datastore model often contributes to disk sprawl. In Red Hat OpenShift Container Platform Virtualization, each VM disk introduces management overhead at the Kubernetes level, which requires careful planning during solution design.

1.7.2 Scalability considerations

A single IBM Block Storage system has a limitation on the number of volumes it can support. Depending on the system model and configuration, this value typically ranges from 2048 to 15864 volumes[1]. Another relevant limit is the maximum number of volume mappings per host, which ranges from 512 to 2048, as specified in IBM documentation. These limits define the maximum size of a Red Hat OpenShift Container Platform Virtualization cluster and the maximum number of VMs that can be served using a single IBM Block Storage system.

[1] Source:
https://www.ibm.com/support/pages/v871x-configuration-limits-ibm-flashsystem-and-san-volume-controller

For example, a single Red Hat OpenShift cluster might include 100 worker nodes capable of running thousands of VMs, each with at least one boot volume and potentially multiple additional volumes. Although a node cannot realistically support more than 512 VMs due to CPU and memory constraints, each of the 100 nodes might run 20 VMs with two volumes per VM. This configuration requires 4,000 volumes, which exceeds the maximum supported by an IBM FlashSystem 5015.

You can bypass these limitations by using multiple IBM Block Storage systems or by adding a software-defined storage (SDS) layer such as IBM Storage Scale CNSA with a local shared volume configuration. This setup can serve a virtually unlimited number of shared filesystem directories like Red Hat OpenShift PVs with a single IBM Block Storage volume underneath.

1.7.3 Protocol support considerations

IBM Block Storage CSI supports iSCSI and FC protocols. However, it has limitations for NVMe/FC and does not support NVMe/TCP at the time this book was written.

Protocol and platform limitations:

- Supported protocols: IBM Block Storage CSI supports iSCSI and FC protocols.
- Red Hat Enterprise Linux CoreOS (RHCOS) limitations: RHCOS does not currently support NVMe/FC according to the documentation.
- OCP CSI limitations: According to Red Hat documentation, NVMe over TCP is not currently supported by the OCP CSI.
- Depreciation of Red Hat Enterprise Linux (RHEL) worker nodes: RHEL worker nodes are deprecated starting with OCP 4.16, which affects NVMe/FC deployment options.

For more information, see Limitations.

1.7.4 Alternative solutions

Red Hat has explored various approaches to address storage limitations for virtualized workloads:

- KubeSAN: Red Hat developed the KubeSAN project to provide datastore-like functionality for Kubernetes environments, similar to VMware datastore or vSAN. However, the project was eventually suspended. KubeSAN offered a shared block storage solution that enabled dynamic provisioning of volumes backed by a single, cluster-wide shared block device. However, the project was eventually suspended.
- IBM Storage Scale Container Native: Red Hat now favors IBM Storage Scale Container Native (formerly known as CNSA) with local disk attachment, currently in Technology Preview. as an alternative for environments requiring datastore-like functionality. This solution allows you to deploy IBM Storage Scale within a Red Hat OpenShift cluster, providing a persistent data store backed by an IBM Storage Scale remote file system. Applications access this data store through the CSI driver using PVs.

You must understand these limitations and alternatives to design appropriate storage architectures for Red Hat OpenShift Virtualization deployments.

1.8 Summary

This chapter describes the key components, benefits, and considerations for integrating IBM FlashSystem storage with Red Hat OpenShift Container Platform, with a focus on Red Hat OpenShift Container Platform Virtualization.

The combination of IBM FlashSystem enterprise storage capabilities with Red Hat OpenShift's container orchestration platform provides a robust foundation for running both containerized and virtualized workloads. Recent advancements in the IBM Block Storage CSI driver, notably the addition of RWX support for block storage, significantly enhance the capabilities available for Red Hat OpenShift Virtualization deployments.

In Chapter 2, "Integrating IBM Storage FlashSystem with Red Hat OpenShift Container Platform" on page 15, you explore the practical aspects of setting up and configuring the CSI driver for IBM FlashSystem and Red Hat OpenShift Container Platform Virtualization.

In Chapter 3, "Use case example" on page 27, you review detailed examples that demonstrate Day-1 and Day-2 operations for the integrated IBM FlashSystem and Red Hat OpenShift Container Platform Virtualization environment.

2

Integrating IBM Storage FlashSystem with Red Hat OpenShift Container Platform

This chapter describes the steps that are required to integrate IBM Storage FlashSystem with Red Hat OpenShift Container Platform, enabling Red Hat OpenShift Container Platform to use storage from IBM FlashSystem.

This chapter contains the following sections:

© Copyright IBM Corp. 2025.

2.1 Configuration workflow

To achieve a minimal working configuration of IBM FlashSystem with Red Hat OpenShift Container Platform, complete the following steps:

- Verify SAN connectivity between Red Hat OpenShift Container Platform and IBM FlashSystem. Help ensure that Fibre Channel (FC) zoning is configured correctly.
- Configure a management account on IBM FlashSystem for the Container Storage Interface (CSI) driver.
- Create a storage pool or child pool on IBM FlashSystem for the CSI driver to provision volumes.
- Install the IBM block storage CSI Operator on Red Hat OpenShift Container Platform and deploy the CSI driver.
- Create a Secret on Red Hat OpenShift Container Platform that contains the IBM FlashSystem management address and account credentials.
- Create a StorageClass on Red Hat OpenShift Container Platform that specifies the storage pool that the CSI driver uses.

The following sections explain these steps in more detail.

2.2 Setting up the IBM FlashSystem and storage environment

This section describes the required configurations on IBM FlashSystem and SAN for integration with the Red Hat OpenShift Container Platform. It targets storage and infrastructure administrators.

This scenario assumes that the initial setup of IBM FlashSystem is complete, the system is operational, and it is connected to the SAN.

2.2.1 Setting up the data plane connectivity

SAN must be configured to permit traffic for the selected block data access protocol between Red Hat OpenShift Container Platform worker nodes and IBM FlashSystem, which includes helping ensure IP network connectivity for iSCSI and FC connectivity for NVMe/FC and SCSI-FC.

Because the IBM Block Storage CSI driver cannot manage SAN configurations such as zoning, you must configure them manually. IBM documentation outlines the zoning requirements and best practice portset configurations for both SCSI-FC and for NVMe/FC.

A host object must exist in the IBM FlashSystem configuration for a VDisk to be available for a Red Hat OpenShift Container Platform worker node. To define host objects for Red Hat OpenShift Container Platform worker nodes, you can choose one of the following options:

- Automate host definition creation and deletion (preferred): Use the HostDefiner component of the IBM Block Storage CSI driver. HostDefiner runs on Red Hat OpenShift Container Platform and automatically creates and updates host definitions on IBM FlashSystem by using the port addresses of the worker nodes.

- Manual host definition creation: If preferred, you can manually create host definitions for Red Hat OpenShift Container Platform worker nodes. Treat each worker node as a standard Linux host. For each physical system, create a separate host object and assign the NQN, all FC WWPNs, and IQN to it.

2.2.2 Setting up the management plane connectivity

The IBM block storage CSI operator that is running on Red Hat OpenShift Container Platform must connect to the IBM FlashSystem management IP address and run commands over SSH. At the time of writing, it uses the pysvc package to run tasks on IBM FlashSystem.

To help ensure proper communication, you must configure all control plane and worker nodes in the Red Hat OpenShift Container Platform cluster to establish TCP connections to port 22 of the IBM FlashSystem management IP address. Multiple processes in the IBM block storage operator initiate these connections.

To use the IBM block management CSI driver, you must provide a username and password for an IBM FlashSystem user. You can authenticate this user locally or remotely through LDAP. single sign-on (SSO), two-factor authentication (2FA), and key-based authentication are not supported.

To help ensure the CSI driver functions correctly, you must assign the “Administrator” role to the specified IBM FlashSystem user. This role grants permissions to create and delete volumes.

To enhance security and resource isolation, implement the IBM FlashSystem Object-Based Access Control (OBAC) or Ownership Groups feature. These features allow Red Hat OpenShift Container Platform-specific users to manage only the resources within their designated ownership group and restrict their access to other pools, virtual disks, host mappings, and hosts on the IBM FlashSystem.

To implement Ownership Groups in a Red Hat OpenShift Container Platform environment, see 3.5, “Day-2 operations: Part 3” on page 73.

To create a secure management account for Red Hat OpenShift Container Platform, take the following actions:

1. Create Ownership Group: In the IBM FlashSystem GUI, select **Access** → **Ownership Group**, click **Create Ownership Group**, specify the group name, and click **Create**.
2. Create User Group and Assign It: Select **Access** → **Users by Group,** click **Create User Group**, specify the group name, and assign the user group to an Ownership Group. Set the group role to **Administrator**. All users in this group receive Administrator permissions but can act only on objects within their Ownership Group.
3. Create a Local User: Click **Create User**, then specify the name and password for a new local user in the group you created in the previous step.
4. Restrict Access Methods: To restrict the user’s access methods, disable **GUI** and **REST API** access for the new user group, leaving only CLI access, as shown in Example 2-1.

Example 2-1 Disabling GUI and REST access

```
IBM_FlashSystem:FS5200:superuser>chusergrp -disablerest yes OCPAdmins
Modifying the authentication setting for this user group affects logins for all
users in the group. Are you sure you want to continue? (y/yes to confirm)  y
IBM_FlashSystem:FS5200:superuser>chusergrp -disablegui yes OCPAdmins
Modifying the authentication setting for this user group affects logins for all
users in the group. Are you sure you want to continue? (y/yes to confirm)  y
```

```
IBM_FlashSystem:FS5200:superuser>lsusergrp OCPAdmins
id 7
name OCPAdmins
role Administrator
remote no
owner_id 0
owner_name MyOCPGroup
multi_factor no
password_sshkey_required no
gui_disabled yes
cli_disabled no
rest_disabled yes
cim_disabled yes
```

2.2.3 Setting up the storage

To create resources, the IBM block storage CSI operator requires a storage pool (MDisk Group). You can use any pool type, including standard, DRP, parent, or child.

A best practice is to create a child pool for IBM Block Storage CSI driver. This approach enables you to logically group volumes, use Ownership Groups, and set capacity quotas or IOPS and bandwidth throttle.

Red Hat OpenShift Container Platform uses the pool name to address the storage pool. To avoid disruptions, ensure that you do not rename the pool without updating the StorageClass configuration on the Red Hat OpenShift Container Platform side.

As a best practice, assign a *provisioning policy* to the pool. If a provisioning policy is not assigned, the Red Hat OpenShift Container Platform StorageClass defines the capacity savings type for volumes. With a provisioning policy, this setting is determined by the storage-side configuration.

The IBM Block Storage CSI driver deletes volumes that are no longer used by Red Hat OpenShift Container Platform. However, the IBM FlashSystem *Volume Protection* feature can block these deletion requests if the volume recently received host I/O. To prevent conflicts, you must disable Volume Protection for storage that is managed by the IBM Block Storage CSI driver.

Volume Protection is controlled by both system-level and pool-level settings. A best practice is to keep Volume Protection enabled at the system level but disable it for any pool or child pool that is used with the IBM Block Storage CSI driver, as shown in Figure 2-1 on page 19. To make the required changes, select **Settings** → **System** → **Volume Protection** in the IBM FlashSystem GUI.

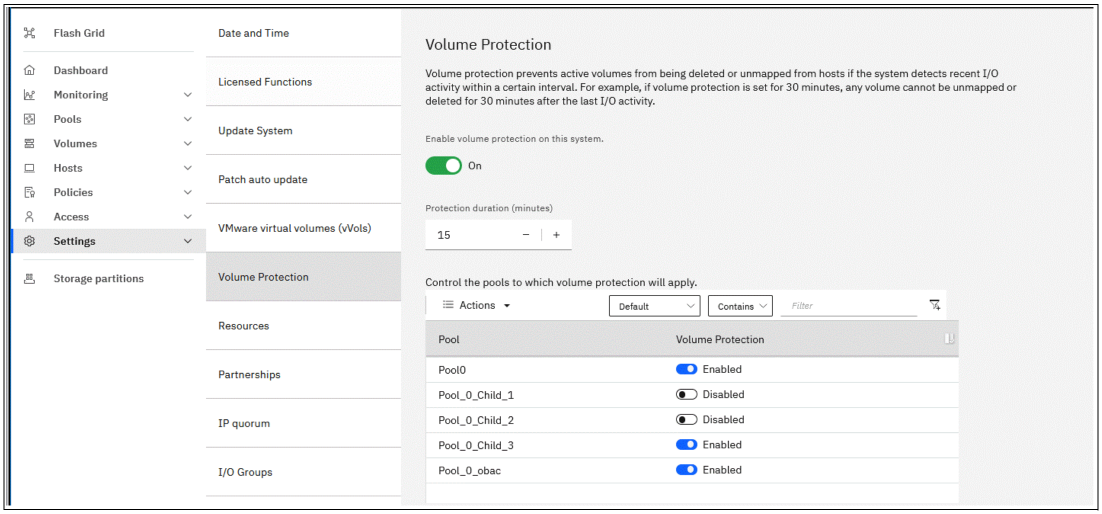

Figure 2-1 Volume Protection

2.3 Setting up the IBM Block Storage CSI driver

This section describes the steps that you must perform within the Red Hat OpenShift Platform management interface to enable IBM FlashSystem block storage for virtual machines (VMs) in Red Hat OpenShift Container Platform Virtualization. This guidance is intended for Red Hat OpenShift administrators. The IBM FlashSystem storage administrator must provide the following information:

- Connection protocol: (NVMe/FC, FC, or iSCSI)
- IBM FlashSystem management IP address
- IBM FlashSystem username and password for the account to be used with Red Hat OpenShift Container Platform
- IBM FlashSystem storage pool name where volumes are created
- Host definition responsibility: Indicate whether IBM FlashSystem or Red Hat OpenShift Container Platform is responsible for creating host definitions on the storage
- Capacity savings type: Confirm whether this item is determined by the policy set on the IBM FlashSystem

2.3.1 Preparing worker nodes

Worker nodes must be able to communicate with the IBM FlashSystem by using one of the supported protocols: NVMe/FC, FC, and iSCSI. If the connection method is not explicitly specified in a HostDefinition, as discussed in 2.3.3, “Connecting IBM FlashSystem storage to IBM Block Storage CSI” on page 23, the system attempts each protocol in the following order: NVMe/FC, FC, and iSCSI.

If a worker node is defined as host NQN, it is considered capable of supporting NVMe attachment. The driver does not perform advanced checks to distinguish between NVMe/FC and NVMe-oTCP. At the time of writing, NVMe-oTCP is not supported by the IBM Block Storage CSI driver. Therefore, it is important to either specify the connection type with HostDefiner or disable NVMe-oTCP to prevent the driver from selecting the NVMe NQN address.

For multipathing between worker nodes and storage to work correctly, IBM provides a MachineConfig YAML file that sets the correct parameters. This file can be found in IBM Documentation. The manifest includes the following components:

- An `/etc/multipath.conf` file with the parameters that are required for IBM FlashSystem and SVC attachment
- An `udev rule` file that sets the device timeout for the IBM Storage Virtualize family as requested
- Enables the multipathd service
- Enables the iSCSI service (this part is commented out in the IBM-provided manifest; uncomment if needed)

After you apply the MachineConfig with the multipathing settings, Red Hat OpenShift Container Platform automatically restarts the worker nodes one by one. You can monitor the progress with the **`oc get mcp`** command.

2.3.2 Installing and running the IBM Block Storage CSI driver operator

You can install the IBM block storage CSI operator in multiple ways, as described in the IBM Block CSI documentation. The simplest way to install the operator is by using the Red Hat OpenShift web console.

To install the IBM block storage CSI operator, complete the following steps:

1. Access the Red Hat OpenShift Container Platform web console:
   a. Log in to the Red Hat OpenShift Container Platform web console by using an account with Operator installation permissions (such as cluster-admin).
   b. Ensure that you are viewing the console from the Administrator perspective.
2. Locate the IBM block storage CSI operator:
   a. Select **Operators** → **OperatorHub** and filter using the keyword “IBM block” to find the IBM Block Storage CSI driver (see Figure 2-2).
   b. Use the Certified channel unless you have specific requirements that are only met by the Community version.

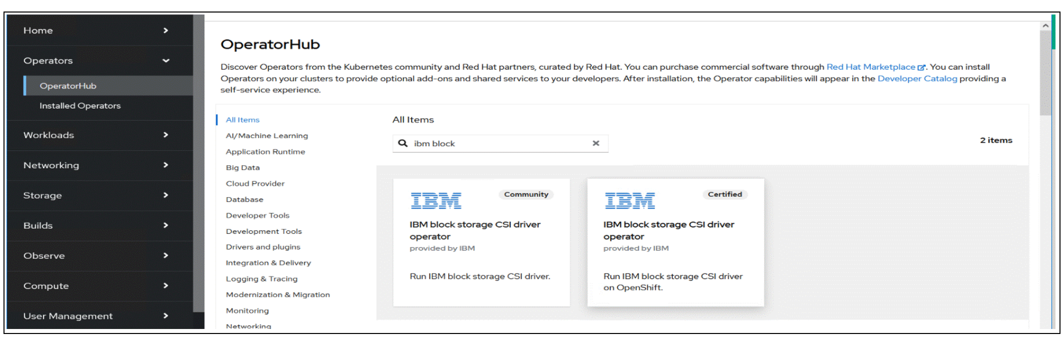

Figure 2-2 IBM Block Storage CSI driver in OperatorHub

   c. Select the latest version in a stable channel and click **Install**.

3. Select the operator namespace:
   a. Choose the namespace for the operator's pods (see Figure 2-3).
   b. Use a separate project, not the default one, or any project used for applications or VMs.
   c. If needed, create a new namespace by clicking the **Installed Namespace** drop-down and selecting **Create Project**.
   d. Keep the other settings at their default values.
   e. Click **Install** after making all selections.

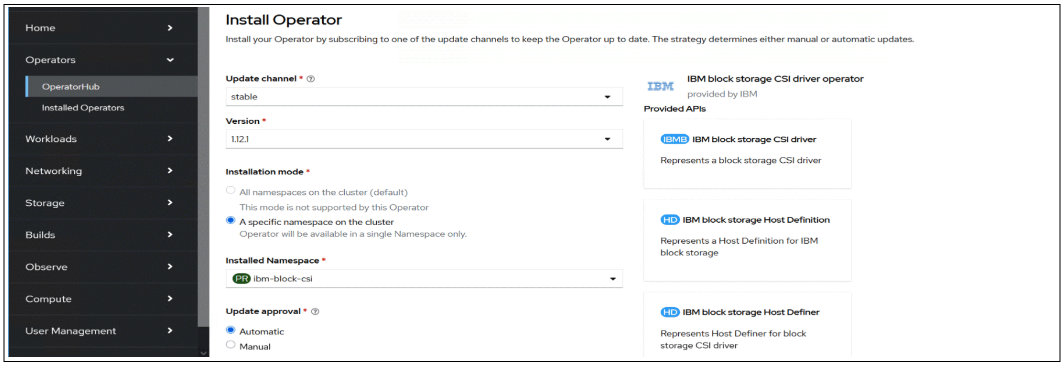

Figure 2-3 Operator installation

4. Verify installation status:
   a. Wait for the message “Installed operator: ready for use”.
   b. verify that the operator is installed and running by checking its status in the **Installed Operators** submenu (see Figure 2-4).

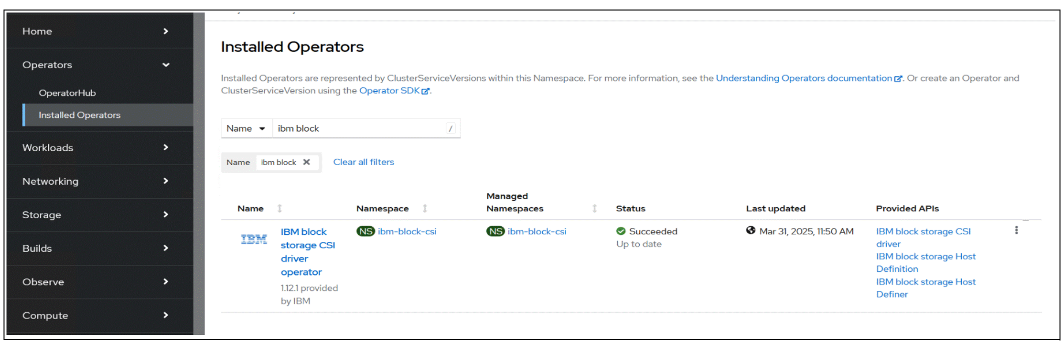

Figure 2-4 Operator installation succeeded

5. Create the IBMBlockCSI Custom Resource Definition (CRD). After the operator is installed, create the IBMBlockCSI CRD to deploy the CSI driver.
   a. Ensure that the worker nodes are ready and have all the necessary MachineConfigs applied. Otherwise, the driver might not be able to pick up endpoint addresses.
   b. To create the CRD, select **Operators** → **Installed Operators**.
   c. Locate the IBM block storage CSI operator.
   d. Click the **IBM Block Storage CSI driver** tab and click **Create IBMBlockCSI** (see Figure 2-5).

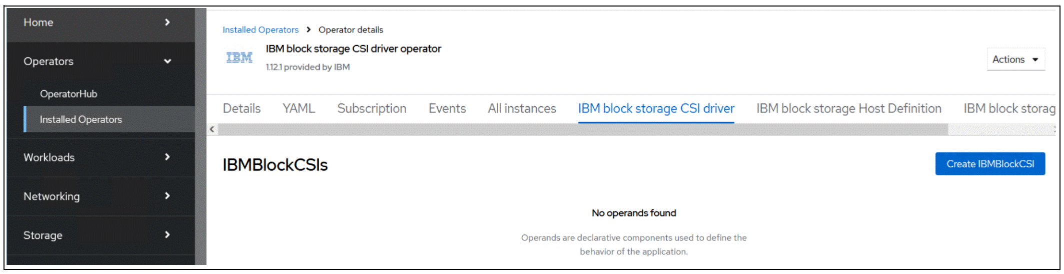

Figure 2-5 Creating IBMBlockCSI

6. Review and verify CRD deployment:
   e. Review the CRD parameters and click **Create**. For most use cases, no modification is required.
   f. To verify deployment:
      i. Check that the IBMBlockCSI status is Running.
      ii. Use the CLI command **`oc get csinodes`** to confirm that the CSI driver is running on worker nodes.
      iii. Use **`oc describe csinodes <node>`** to view the NQN, WWPNs, and IQN that the node will use (see Example 2-2).

Example 2-2 The csinodes resource

```
[ocpadmin@lab ~]$ oc get csinodes
NAME                           DRIVERS   AGE
cp-1.ocp4.lab                  0         10d
cp-2.ocp4.lab                  0         10d
cp-3.ocp4.lab                  0         10d
wk-1.ocp4.lab                  1         10d
wk-2.ocp4.lab                  1         10d
wk-3.ocp4.lab                  1         10d
[ocpadmin@lab ~]$ oc describe csinodes wk-1.ocp4.lab
Name:               wk-1.ocp4.lab
.....
Spec:
  Drivers:
    block.csi.ibm.com:
      Node ID: wk-1.ocp4.lab;
                 nqn.2014-08.org.nvmexpress:uuid:58688...ac4;
                 10000090fa5246d4:10000090fa5246d5;
                 iqn.1994-05.com.redhat:d05c0241d27
```

2.3.3 Connecting IBM FlashSystem storage to IBM Block Storage CSI

Several Red Hat OpenShift Container Platform resources are required to use storage capacity from IBM FlashSystem, so complete the following steps:

1. Create a Secret with the storage system IP address and credentials: Create a Secret that contains the system's management IP address, login credentials, and password.
   - A YAML example is available in the IBM Documentation.
   - Example 2-3 demonstrates the simplest method by using the **oc** CLI.
   - You can create this Secret in any appropriate namespace, such as within the CSI driver project.
   - The storage user account must be prepared and provided by the storage system administrator, as detailed in section 2.2, "Setting up the IBM FlashSystem and storage environment" on page 16.

Example 2-3 Secret with the storage system IP address and credentials

```
$ oc create secret generic flashsystem-secret --from-literal=username=ocpuser
--from-literal=password=ocppassword
--from-literal=management_address=10.20.30.40 -n ibm-block-csi
```

2. Create a StorageClass: Create a StorageClass to define how storage is provisioned.
   - Detailed instructions, including advanced configurations for replicating or multi-IO-group systems, are available in the IBM Documentation.
   - Example 2-4 shows a basic configuration with key parameters.
3. Important parameters:
   - **is-default-virt-class**: Required per Red Hat OpenShift Container Platform documentation. If both Red Hat OpenShift Container Platform Virtualization and Red Hat OpenShift Container Platform default storage classes exist, the virtualization class takes precedence when creating VM disks.
   - **pool**: Specifies the name of the storage pool, provided by the storage system administrator. See section 2.2, "Setting up the IBM FlashSystem and storage environment" on page 16 for details.
   - **SpaceEfficiency**: Defines the capacity savings mode that is used on the storage system.

> **Note:** If a capacity savings policy is already defined on the storage system, omit the **SpaceEfficiency** parameter from the StorageClass manifest.

- **virt_snap_func**: Determines the snapshot technology used.
  - For IBM FlashSystem versions up to Storage Virtualize 8.5.0.x, set **virt_snap_func** to "false" to use the prior IBM FlashCopy® function.
  - For versions starting with Storage Virtualize 8.5.1.x, set **virt_snap_func** to "true" to use the new generation of snapshot functions.

Example 2-4 A basic configuration highlighting important points

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
   name: flashsystem-csi-sc
```

```
    annotations:
      storageclass.kubevirt.io/is-default-virt-class: "true"
provisioner: block.csi.ibm.com
parameters:
  pool: pool2
  SpaceEfficiency: thin            # Optional.
  volume_name_prefix: ocp1  # Optional.
  virt_snap_func: "false"          # Optional. The default is "false".
  csi.storage.k8s.io/fstype: xfs   # Optional. The default is ext4.
  csi.storage.k8s.io/secret-name: flashsystem-secret
  csi.storage.k8s.io/secret-namespace: ibm-block-csi
allowVolumeExpansion: true
```

If host definitions were manually created on the storage system before this step, your Red Hat OpenShift Container Platform cluster configuration is complete, and you can begin using IBM FlashSystem storage.

However, dynamic host definition is the best practice approach. This method automatically creates and removes host definitions when necessary for all or selected nodes within your Red Hat OpenShift Container Platform cluster's IBM FlashSystem configuration. It also updates host definitions to reflect changes in node port addresses. This automated solution significantly reduces administrative processing and improves the IBM Block Storage CSI driver's performance by minimizing management queries to the storage system.

To use dynamic host definition, complete the following steps:

1. Create and start HostDefiner CRD:
   a. In the Red Hat OpenShift Container Platform web console, select **Operators** → **Installed Operators**.
   b. Find the IBM block storage CSI operator, and click the **IBM block storage HostDefiner** tab (see Figure 2-6).
   c. Click **Create HostDefiner**.

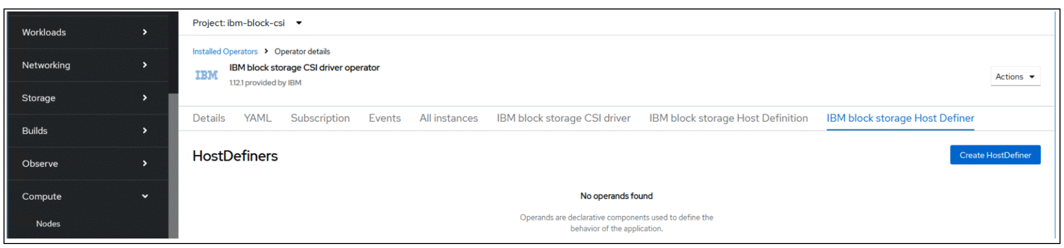

Figure 2-6   Create HostDefiner

2. Adjust the default YAML configuration:
   a. For optimal results, modify the default YAML as shown in Figure 2-7 on page 25:
   – **`dynamicNodeLabeling`**: Set to **“**`true`**”** to automatically create host definitions for all worker nodes in the cluster. This setting fits most use cases. If it remains with the default **“**`false`**”** setting, you must manually label the nodes that will be managed by HostDefiner. This approach is useful when only a subset of nodes use the IBM block storage CSI driver.
   – **`connectivityType`**: Set to the protocol used by the storage system: **“**`fc`**”**, **“**`iscsi`**”**, or **“**`nvmeofc`**”**. If this field is not set, the host definition is created for the first protocol the node supports, which might not be the wanted outcome.

- **`prefix`**: Defines the prefix added to the node name when the host definition is created on the storage system. The prefix is automatically followed by an underscore "_". For example, if the node name is `worker1.local` and the prefix is set to `ocp`, the resulting host object name is `ocp_worker1.local`.

For a complete list of parameters, see IBM Documentation.

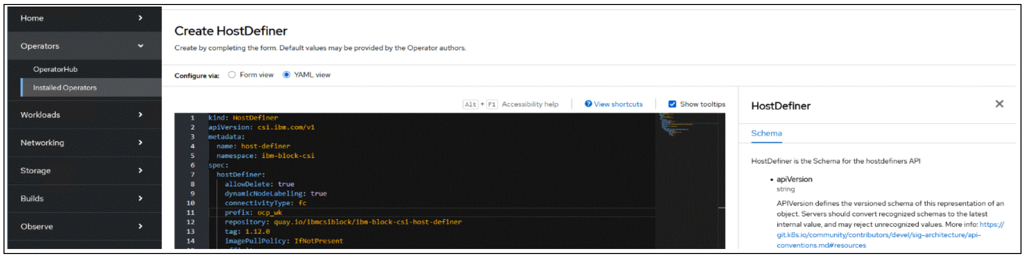

Figure 2-7 HostDefiner YAML

3. Start HostDefiner and verify the host definitions: Upon startup, HostDefiner scans all StorageClasses that use the IBM block CSI provisioner. It retrieves the Secret containing IBM FlashSystem credentials from each StorageClass and creates corresponding host definitions for worker nodes.
   - For each Red Hat OpenShift Container Platform node (or each labeled node, if dynamicNodeLabeling is enabled) and each IBM FlashSystem Secret, a distinct HostDefinition record is generated.
   - You can view HostDefinition records by using the web console or the **oc** CLI, as shown in Example 2-5.

Example 2-5 oc get hostdefinitions

```
[ocpadmin@lab ~]$ oc get hostdefinitions.csi.ibm.com
NAME                     AGE  PHASE    NODE             MANAGEMENT_ADDRESS
wk-1.ocp4.lab-x...t      11d  Ready    wk-1.ocp4        10.20.30.40
wk-2.ocp4.lab-h...f      11d  Ready    wk-2.ocp4        10.20.30.40
wk-3.ocp4.lab-n...d      11d  Ready    wk-3.ocp4        10.20.30.40
[ocpadmin@lab ~]$ oc get hostdefinitions wk-1.ocp4.lab-x...t -o yaml
apiVersion: csi.ibm.com/v1
kind: HostDefinition
...
spec:
  hostDefinition:
    connectivityType: fc                    # host connection protocol
    ioGroups:                               # All IOgroups allowed by default
    - 0
    - 1
    - 2
    - 3
    managementAddress: 10.20.30.40         # FlashSystem management IP
    Node ID: wk-1.ocp4.lab;
               nqn.2014-08.org.nvmexpress:uuid:58688...ac4;
               10000090fa5246d4:10000090fa5246d5;
               iqn.1994-05.com.redhat:d05c0241d27
    nodeName: wk-1.ocp4.lab
    nodeNameOnStorage: ocp_wk_wk-1.ocp4.lab # hostname as set on the storage
    ports:                                 # host ports defined on the storage
    - 10000090FA5246D4
```

```
    - 10000090FA5246D5
    secretName: flashsystem-secret          # secret set in StorageClass
    secretNamespace: ibm-block-csi
status:
  phase: Ready
```

When deployed, HostDefiner identifies and updates existing host definitions. However, it does modify the host object names on the storage system.

2.4 Potential issues

This section lists potential issues that can arise during the integration setup. These issues typically occur when systems that are already configured and running are repurposed or reconfigured to support more workloads.

Watch out for the following potential issues:

- Multiple instances of IBM block storage CSI operator

  Although you can install more than one operator instance in different namespaces, this approach introduces significant risks. This practice can cause conflicts and errors that render all operator instances unusable. These issues can result from overlapping configurations, resource contention, or mismanagement of shared resources. To avoid such complications, carefully plan and manage operator installations. Help ensure that each instance operates within its designated namespace without interference.

- Multiple Secrets referencing the same IBM FlashSystem

  When StorageClasses use Secrets that reference different IBM FlashSystem Ownership Groups or a mix of grouped and ungrouped accounts, errors can occur in host definitions and volume-to-host mappings.

- Incorrect or unset default StorageClass

  Configure the StorageClass that uses the IBM Block Storage CSI driver as the default. This action helps ensure that VM boot devices are created on the intended IBM FlashSystem storage. Set the annotation `storageclass.kubevirt.io/is-default-virt-class=”true”` on the StorageClass to designate it as the default for virtualization.

3

Use case example

This chapter describes a complete lifecycle of deploying the IBM Block Storage Container Storage Interface (CSI) driver in a Red Hat OpenShift Container Platform cluster. The deployment is designed to support workloads running on Red Hat OpenShift Virtualization, which enables you to run and manage virtual machines (VMs) alongside containerized applications in a unified Kubernetes environment.

While the installation part of Day-1 operation was covered in the previous chapter, this chapter focuses on Day-1 configuration and Day-2 lifecycle operations.

This chapter contains the following sections:

© Copyright IBM Corp. 2025.

3.1 Introduction to case study

The case study covers various stages of operating Red Hat OpenShift Virtualization environments that use the IBM Block Storage CSI.

While Chapter 1, “Introduction” on page 1 covers the Day-0 planning phase and the installation portion of Day-1 operations, this chapter focuses on postinstallation Day-1 configurations and Day-2 lifecycle operations.

Following installation, creating a VM verifies a successful setup. For migration scenarios that replace existing virtualization platforms like VMware ESXi with Red Hat OpenShift Container Platform, a common practice is to migrate a VM to Red Hat OpenShift Container Platform and map its underlying storage to a StorageClass provided by the IBM Block Storage CSI driver.

In greenfield deployments, where you set up a new IT infrastructure, system, or application, you can create a new VM to confirm the intended operation of both Red Hat OpenShift Container Platform and the IBM Block Storage CSI driver. After deploying the VM, a typical Red Hat OpenShift operation involves creating a route to expose a service that is offered by the VM and testing its availability from outside the Red Hat OpenShift environment.

The Day-2 lifecycle and maintenance operations can be grouped into the following areas:

- Operational resiliency: Help ensure that VMs can be recovered if there is a misconfiguration or an accidental deletion.
- Concurrency: Use VM live migration to perform updates and maintenance operations while maintaining zero-downtime VM availability.
- Security and Safety: Reduce the attack surface for potential intruders or malware by isolating tenants by using Red Hat OpenShift RBAC and IBM Block Storage Object-Based Access Control (OBAC) together with IBM Block Storage Child Pools.

This chapter also covers advanced topics that might be required for Day-1 and Day-2 operations. Troubleshooting Red Hat OpenShift Container Platform deployments with the IBM Block Storage CSI driver requires an understanding of the following Red Hat OpenShift custom resources (CRs):

- VirtualMachine: Serves as the anchor definition of the VM.
- DataVolume: Connects the VirtualMachine to the source image used to instantiate the VM.
- PersistentVolumeClaim: Binds to a persistent volume (PV) that holds the VM’s data.

3.2 Day-1 operations

This section explains Day-1 operations following the configuration of Red Hat OpenShift Container Platform Virtualization. Day-1 operations include creating VMs and setting up access to VMs or services running inside VMs using the Red Hat OpenShift Route feature. This section also covers the migration from VMware to Red Hat OpenShift Container Platform.

3.2.1 VM creations

Before you create VMs, you must verify the prerequisites. VM creation depends on several required Red Hat OpenShift Operators:

- Red Hat OpenShift Virtualization
- Kubernetes NMState
- MetalLB

Also, install **`virtctl`** to manage VMs from the CLI.

To migrate VMs to Red Hat OpenShift Virtualization, you need the Migration toolkit for virtualization Operator from Red Hat.

VMs are always created within the currently selected namespace. For specific namespace requirements, either use an existing namespace or create a new one. VM creation in this example requires a new namespace.

To create a VM, complete the following steps:

1. In Red Hat OpenShift Container Platform web console, select **Home** → **Projects** and click **Create Project**, as shown in Figure 3-1.

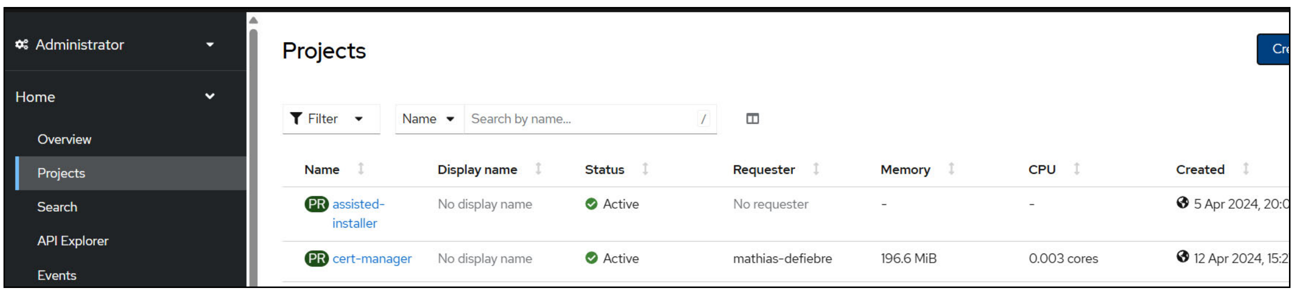

Figure 3-1 Create Project window

2. Enter demo-vm in the Name field, demo-vm in the Display name field and Demo VM Namespace in the Description field, as shown in Figure 3-2. Click **Create**.

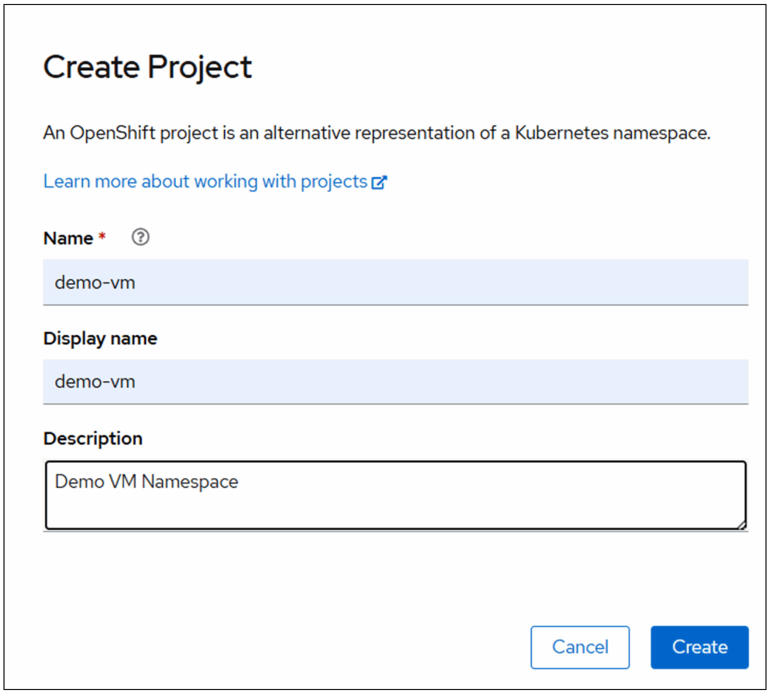

Figure 3-2 Create Project dialog box

3. In the Red Hat OpenShift Container Platform web console, select **Virtualization** → **Catalog**, as shown in Figure 3-4 on page 31.

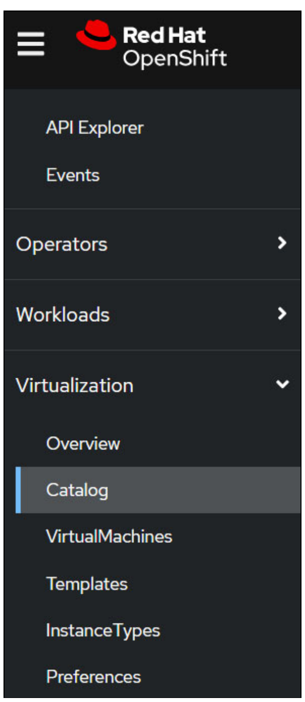

Figure 3-3 Creating a virtual machine by using the Catalog menu

The “Create new VirtualMachine” window opens, as shown in Figure 3-4 on page 31.

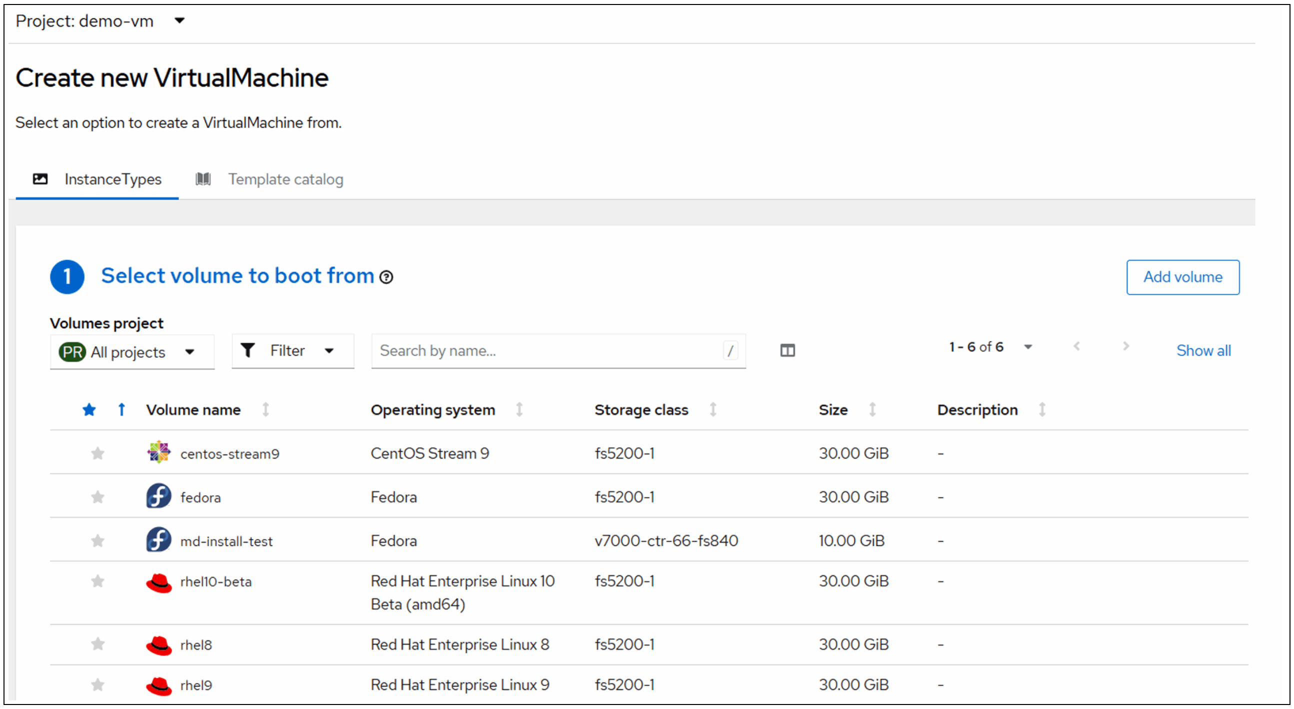

Figure 3-4 Create new VirtualMachine window

4. Select the **demo-vm** project before you proceed with VM creation by using a template:

   You can create VMs by using either the Instance Type or the Template Catalog:

   – Instance Type: Provides predefined VM configurations or allows you to use a user-defined template with predefined resources such as CPU and memory. These templates are designed for common workloads and categorized by resource configuration (for example Small, Medium, or Large).

   – Template Catalog: Offers basic configurations for Red Hat Enterprise Linux (RHEL), CentOS, Fedora, and Windows VMs. The default configuration includes 1 CPU and 2 GB of memory.

5. From the Catalog page, select **Template Catalog** section, as shown in Figure 3-5.

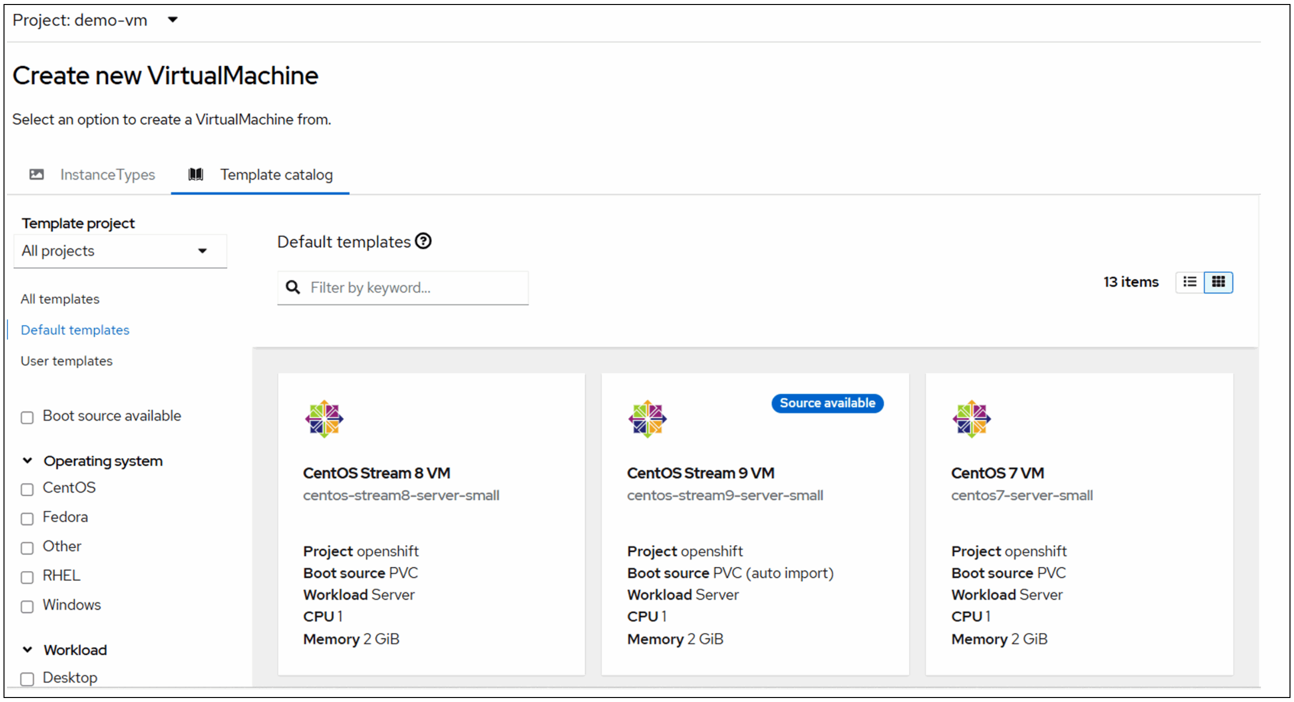

Figure 3-5 Create new VirtualMachine: Template Catalog Selection

6. Under the Operating System filter, select **RHEL**, and then select the **Red Hat Enterprise Linux 9 VM** tile, as shown in Figure 3-6.

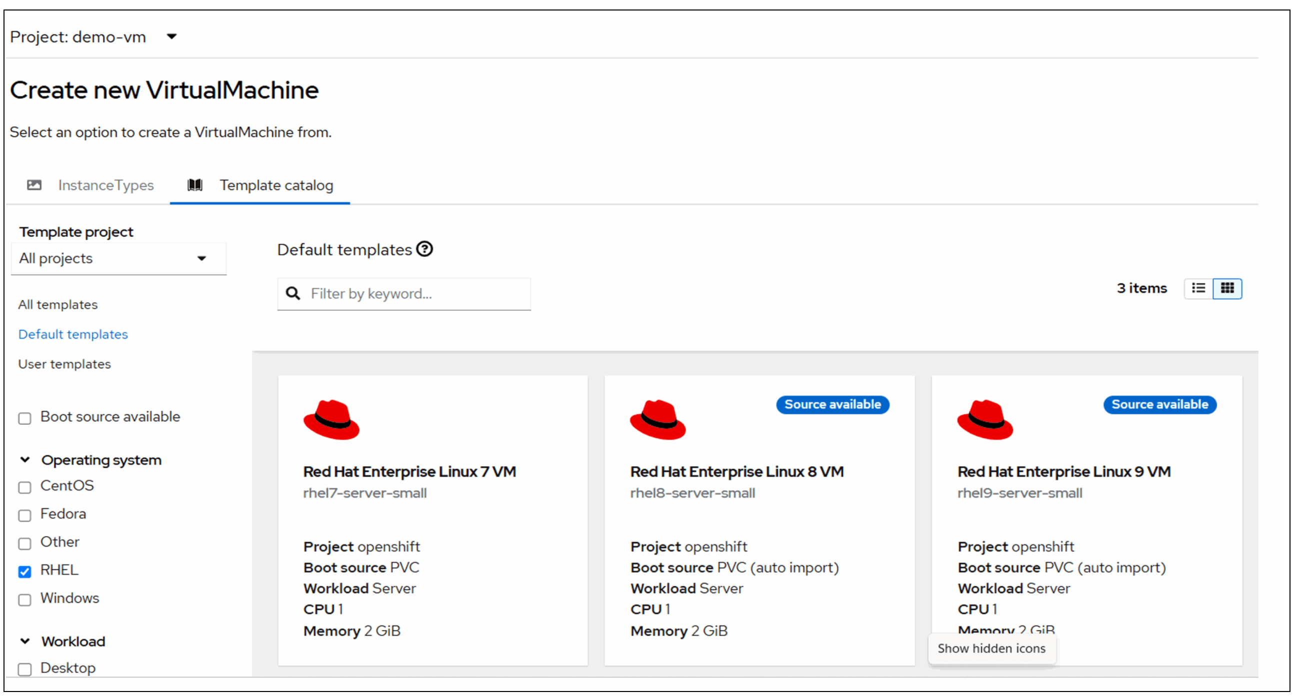

Figure 3-6 Create new VirtualMachine: Operating system selection

7. Enter demo-vm-RHEL 9 in the VirtualMachine name field and click **Quick create VirtualMachine**, as shown in Figure 3-7 on page 33.

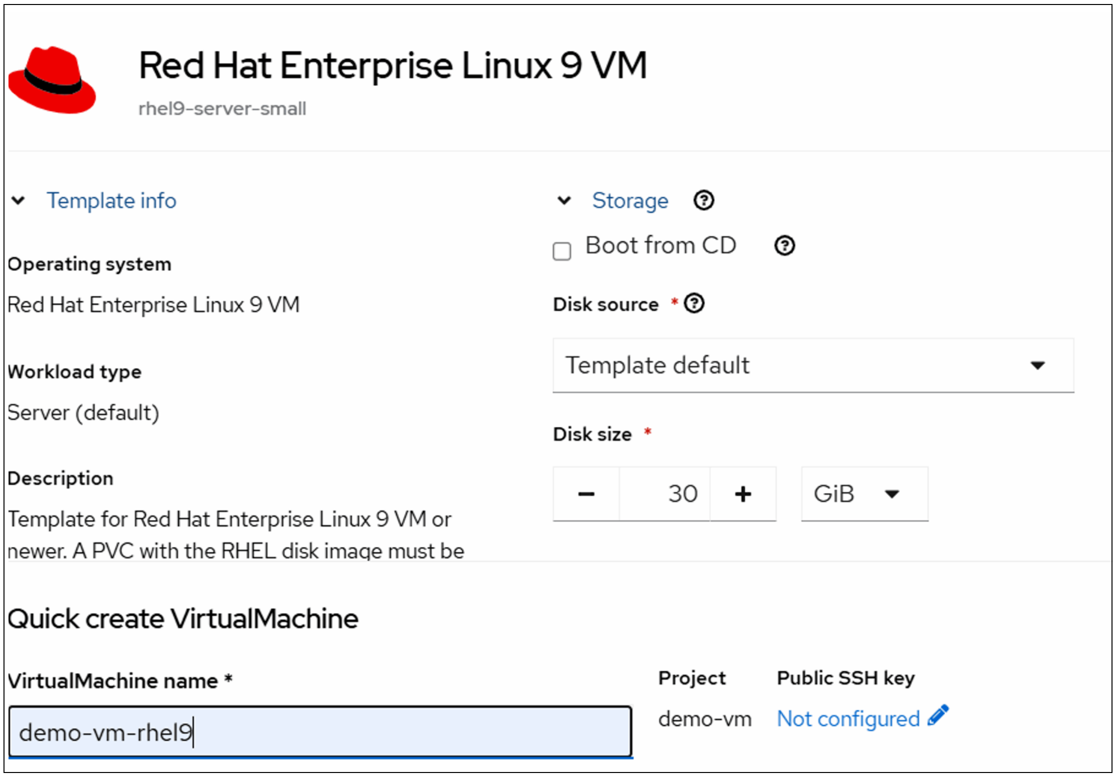

Figure 3-7 Quick create VirtualMachine

This action starts the VM provisioning process, which takes a few minutes. When complete, the VM status displays as Running, indicating the VM is ready for use, as shown in Figure 3-8.

Figure 3-8 Virtual machine: Running status

8. To access the VM, click the **Open new console** link, as shown in Figure 3-9.

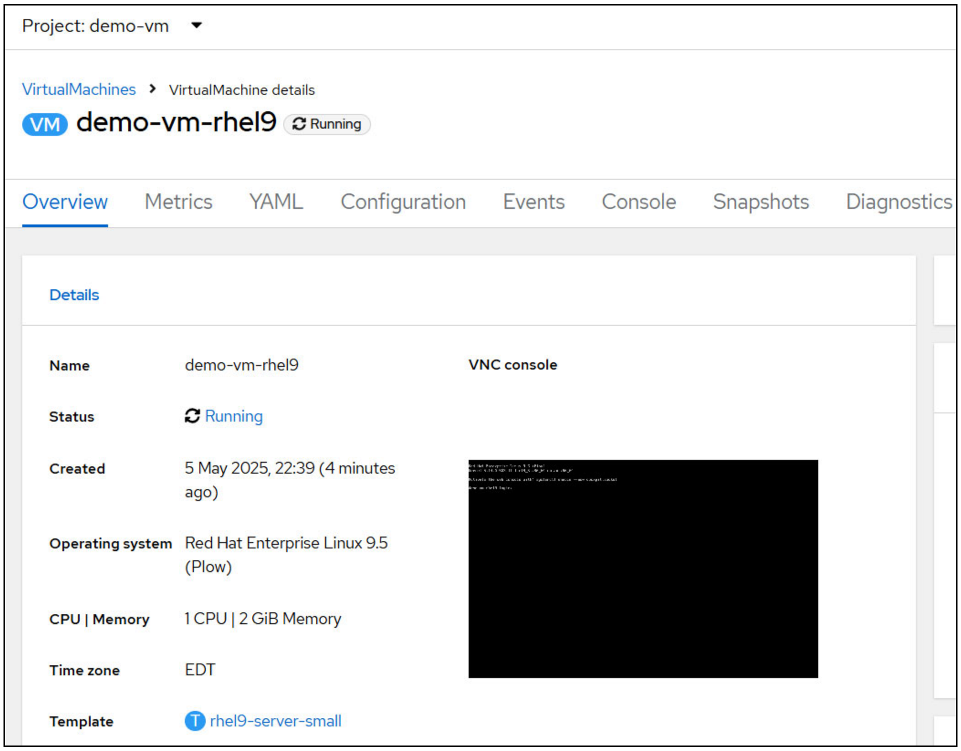

Figure 3-9 Accessing the virtual machine

9. You can also access the VM by clicking the **Console** tab on the VirtualMachine details page, as shown in Figure 3-10.

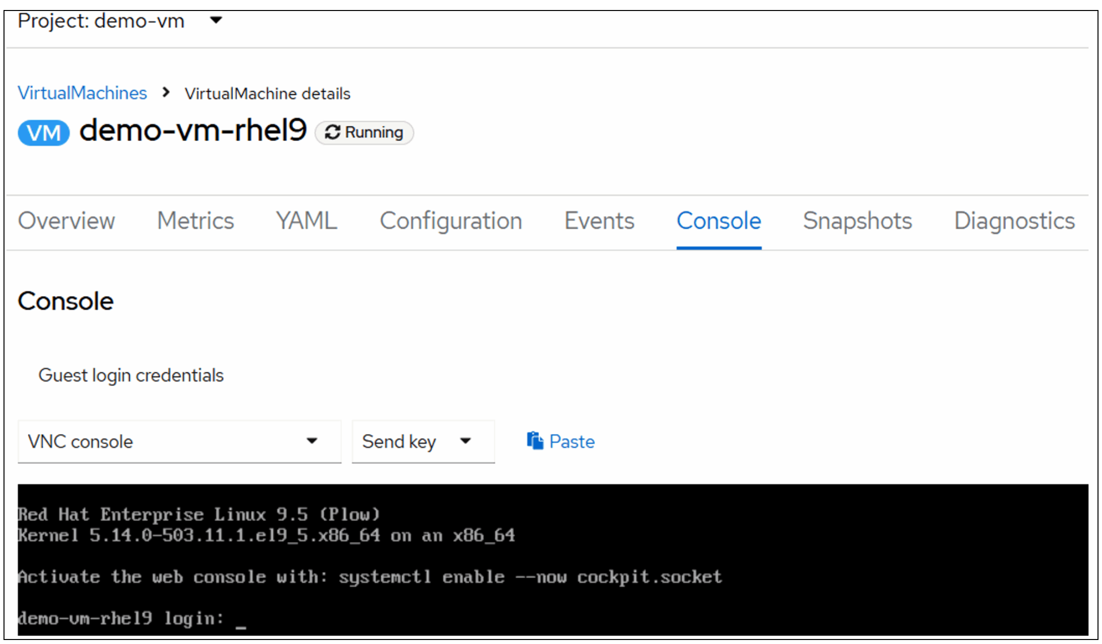

Figure 3-10 Accessing the virtual machine

10. Log in to the VM by using the Guest login credentials section, which you can show or hide by using the carat icon in the upper left (when collapsed), as shown in Figure 3-11.

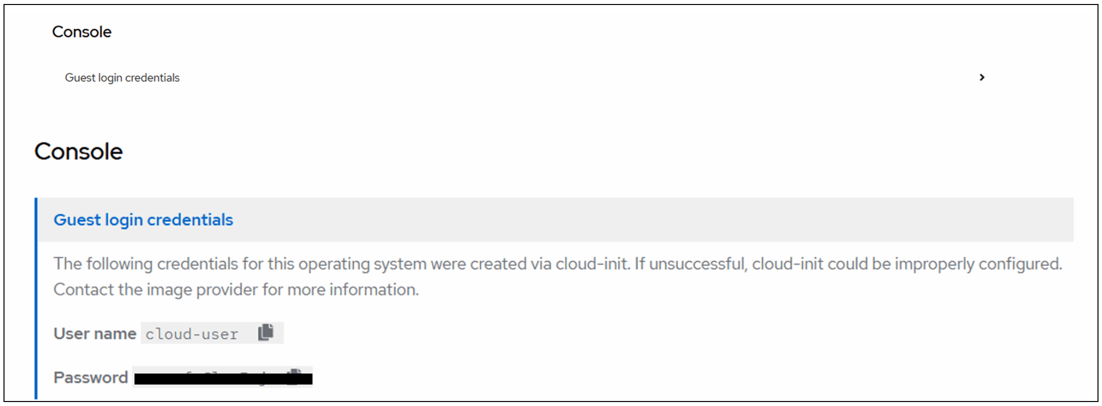

Figure 3-11 Virtual machine initial credentials

11. When you create a VM that uses the Template Catalog, the default user cloud-user is automatically created. Log in using the default username and password that are shown in Figure 3-12.

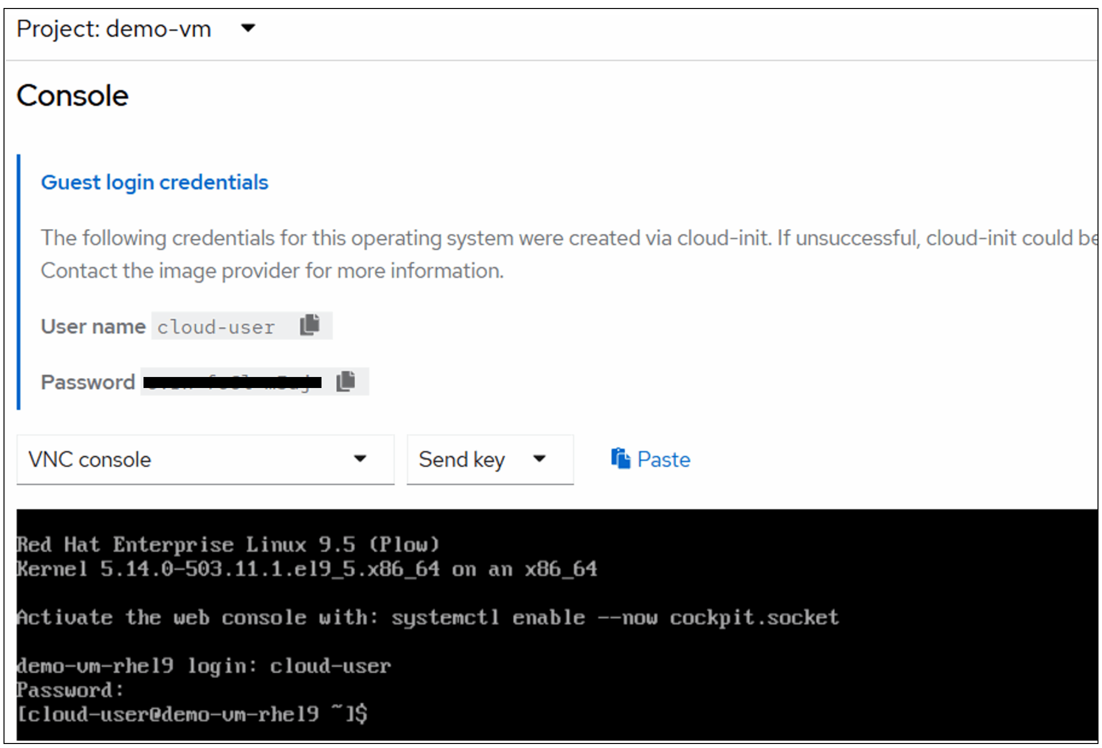

Figure 3-12 Virtual machine access by using the default username and password

VM service access

For more information about creating service access for a VM by using the **virtctl** CLI tool, see IBM Documentation.

Accessing VMs by using a service port

This section demonstrates how to access services running inside a VM by using the Red Hat OpenShift Service feature. Complete the following steps:

1. In Red Hat OpenShift Container Platform web console, select **Networking** → **Service**, as shown in Figure 3-13.

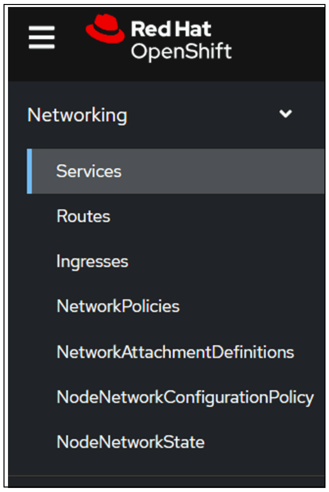

Figure 3-13 Service option from Networking

2. Click **Create Service** from the Service window, as shown in Figure 3-14.

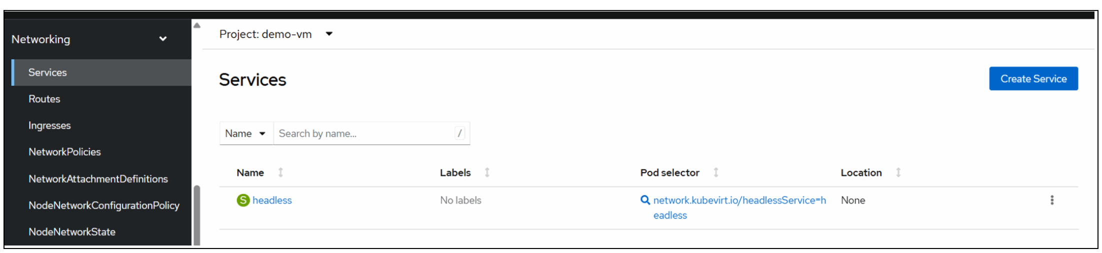

Figure 3-14 Create Service

3. Provide the service name demo-vm-RHEL 9-ssh:
   a. Set the selector name as **service**, and include the service name (demo-vm-RHEL 9-ssh) in the selector.
   b. In the Port section, select the protocol type **(TCP** or **UDP**) based on the service.
   c. Define the port on which the service is running. For SSH, use Port 22 and the Target Port 22.
   d. Define the NodePort, which allows access from outside the Red Hat OpenShift cluster. In this example, use Port 30222 to access SSH externally, as shown in Figure 3-15 on page 37.

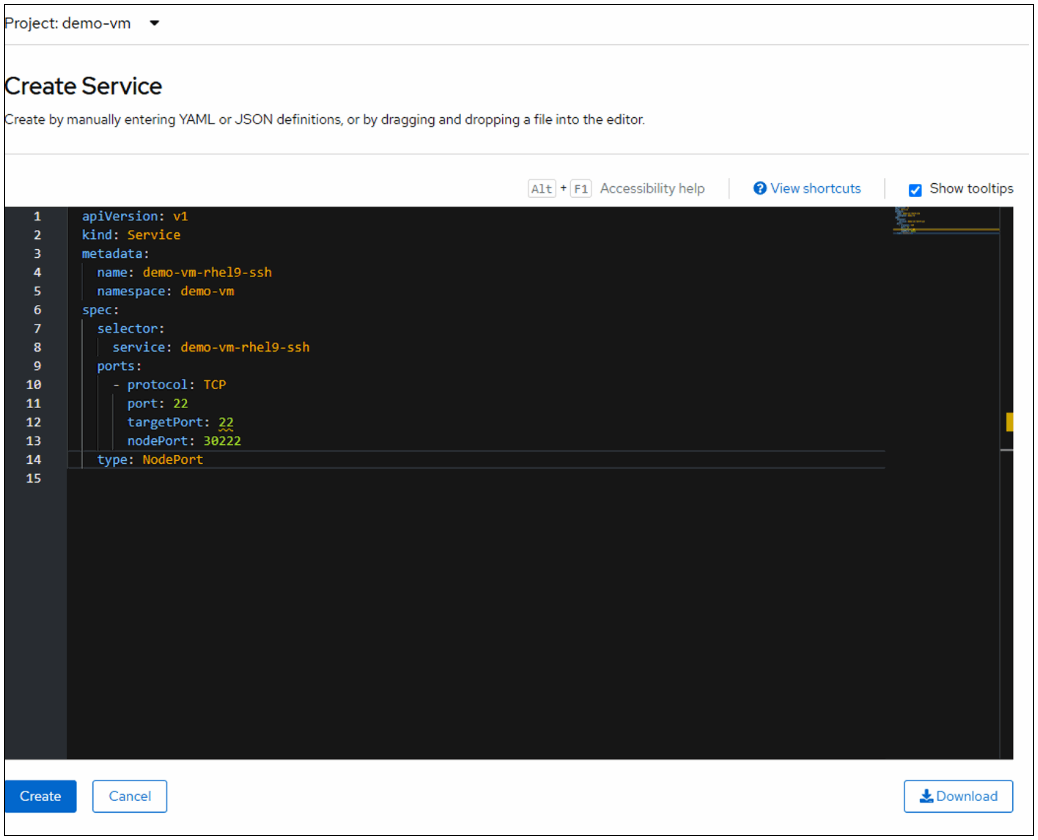

Figure 3-15 Creating a service by using YAML

After you create the service, the Service Details page opens, as shown in Figure 3-16.

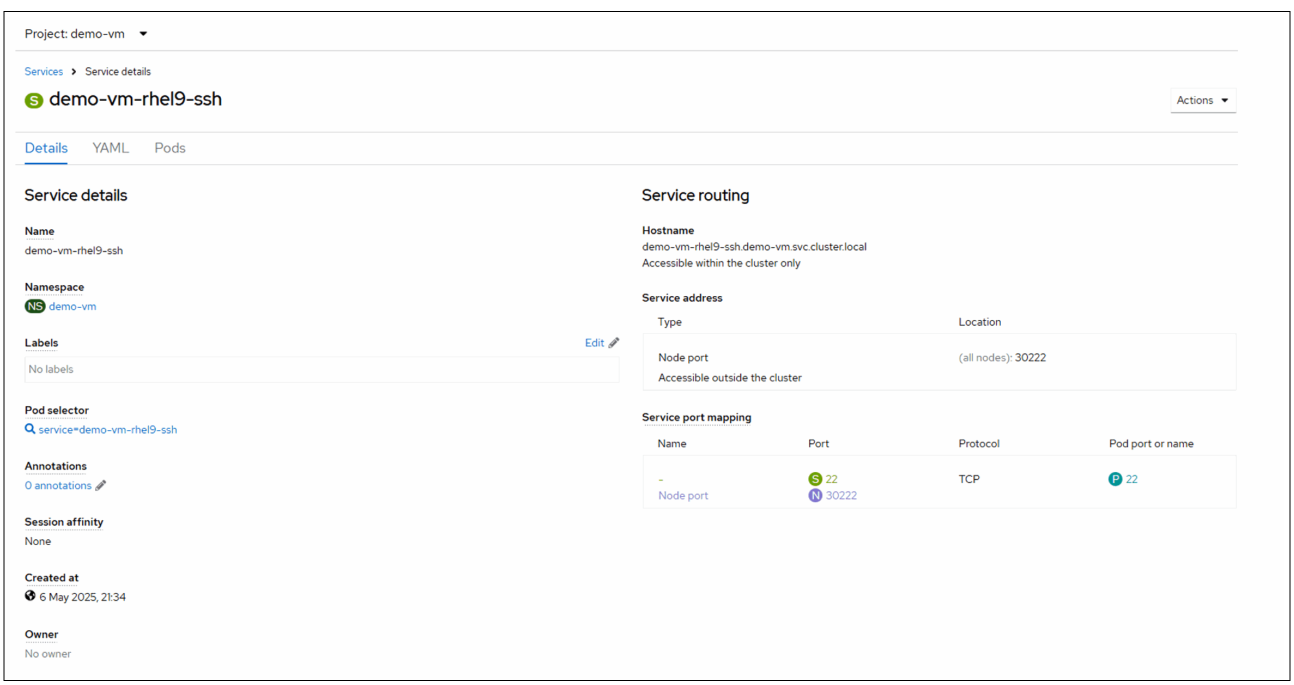

Figure 3-16 Detailed service information

4. In the Red Hat OpenShift Container Platform web console, select **Networking** → **Services** to list the newly created service, as shown in Figure 3-17.

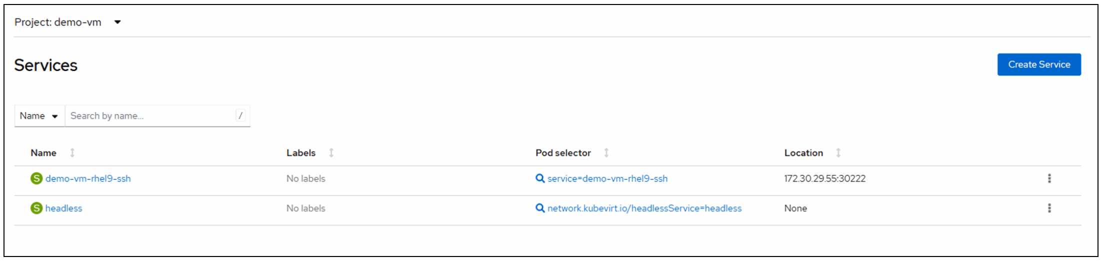

Figure 3-17 Detailed service information

5. To configure service selection within a VM, click the wanted VM in the Red Hat OpenShift Container Platform web console. Select **Virtualization** → **VirtualMachines**, as shown in Figure 3-18.

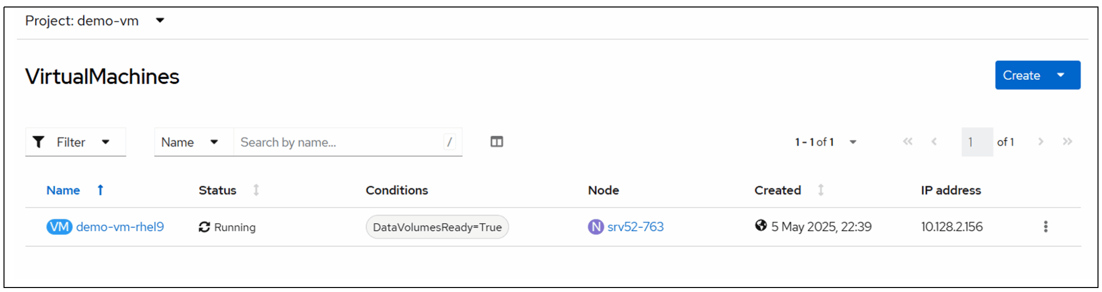

Figure 3-18 Virtual machine listing

6. Select the VM and click the **YAML** section, as shown in Figure 3-19.

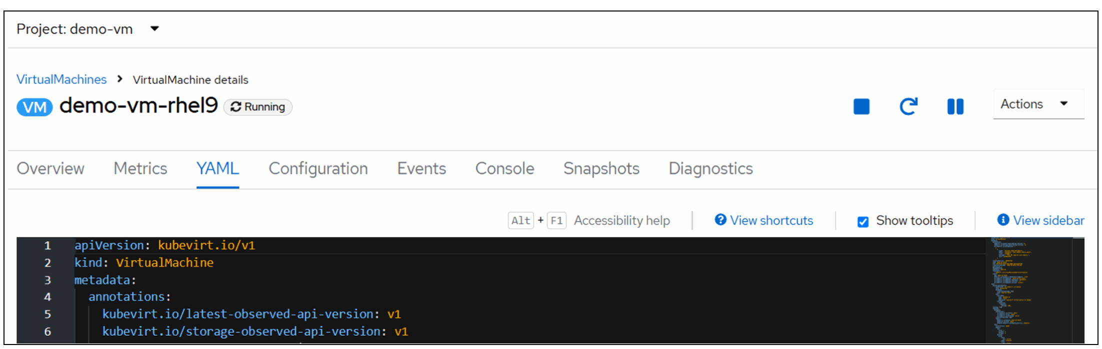

Figure 3-19 YAML section of the virtual machine details page

7. Locate the template: and metadata: within the YAML. Under the `labels:` section, add `service: demo-vm-RHEL 9-ssh`, as shown in Figure 3-20 on page 39.

```
template:
  metadata:
    annotations:
      vm.kubevirt.io/flavor: small
      vm.kubevirt.io/os: rhel9
      vm.kubevirt.io/workload: server
    creationTimestamp: null
    labels:
      kubevirt.io/domain: demo-vm-rhel9
      kubevirt.io/size: small
      network.kubevirt.io/headlessService: headless
      service: demo-vm-rhel9-ssh
```

Figure 3-20 YAML update for the service label

8. Log in to the VM and make the necessary changes to enable SSH login in `/etc/ssh/sshd_config` or other relevant service configuration files, including SSH key exchange.
9. Go to the bastion nodes and log in to the VM by using the `ssh` command. Provide the Red Hat OpenShift node name and Port 30222 (defined during service creation), as shown in Figure 3-21.

```
bash-4.4 ~ $ ssh cloud-user@srv51-763 -p 30222
Register this system with Red Hat Insights: rhc connect

Example:
# rhc connect --activation-key <key> --organization <org>

The rhc client and Red Hat Insights will enable analytics and additional
management capabilities on your system.
View your connected systems at https://console.redhat.com/insights

You can learn more about how to register your system
using rhc at https://red.ht/registration
Last login: Tue May  6 12:43:33 2025 from 100.64.0.4
[cloud-user@demo-vm-rhel9 ~]$
```

Figure 3-21 Log in to the VM by using the service

3.2.2 VM migration

Before you migrate VMs from VMware, you must verify the following prerequisites. You need the Migration Toolkit for Virtualization Operator from Red Hat OpenShift. Also, you must have admin-level access to VMware vCenter or ESXi, along with appropriate network access.

Creating a provider for virtualization

To create a provider for virtualization, complete the following steps:

1. In the Red Hat OpenShift Container Platform web console, select **Migration** → **Providers for virtualization,** as shown in Figure 3-22.

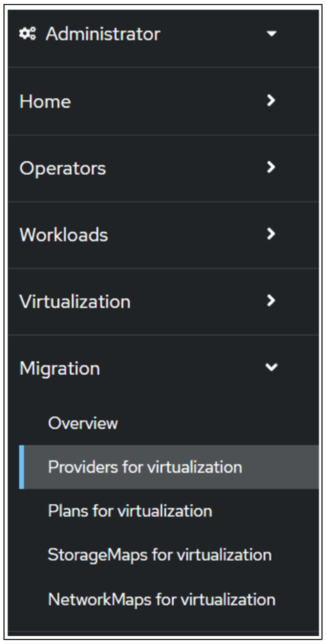

Figure 3-22 Provider for visualization

2. To create a provider, click **Create Provider**, as illustrated in Figure 3-23.

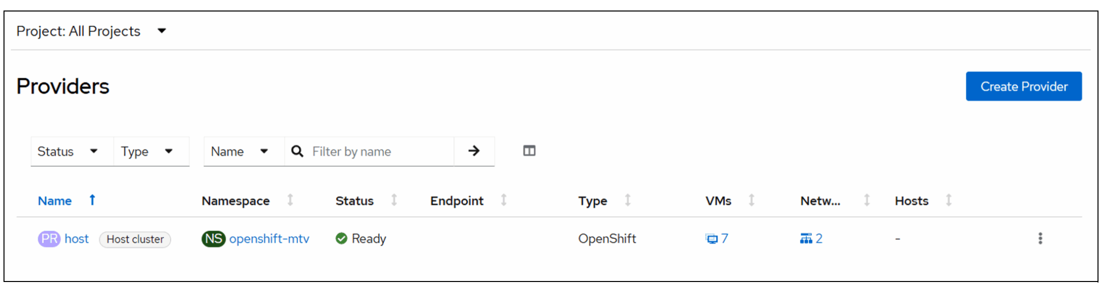

Figure 3-23 Provider for visualization

3. Under the Create Provider section, select the **vm vSphere** tab, as shown in Figure 3-24.

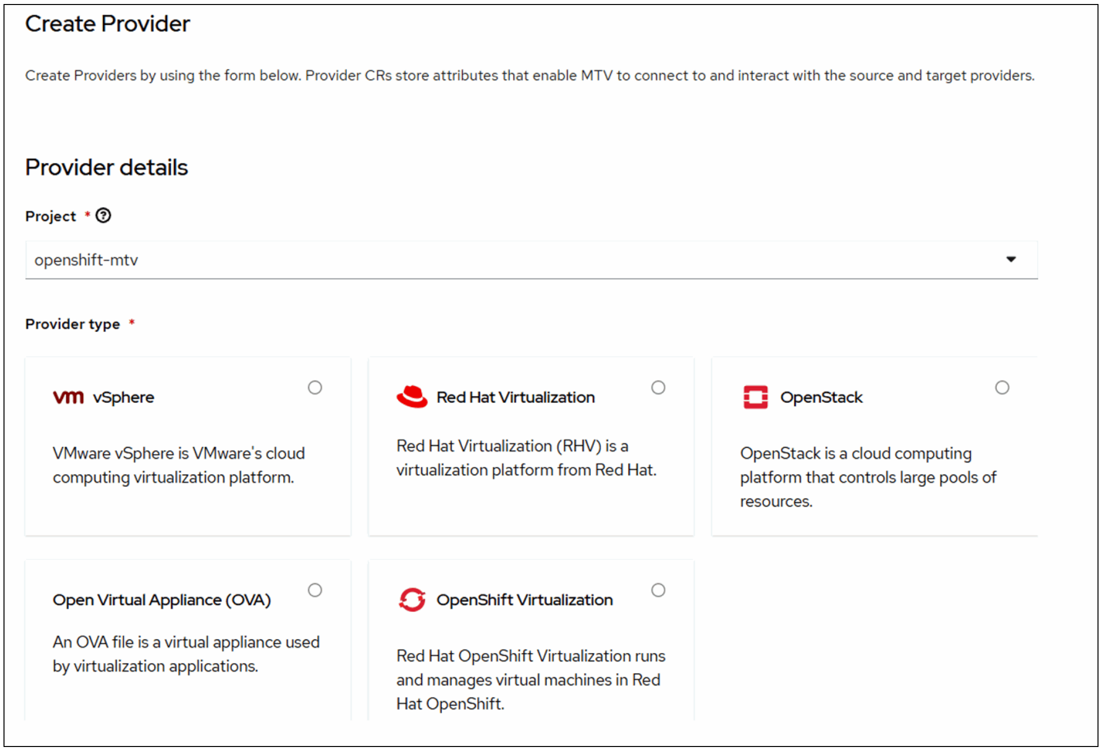

Figure 3-24 Provider details

4. Specify the default namespace as Red Hat OpenShift-mtv. Enter the provider resource name (vcsa). Select the endpoint type (either vCenter or ESXi host). In this example, select **vCenter**, as shown in Figure 3-25.

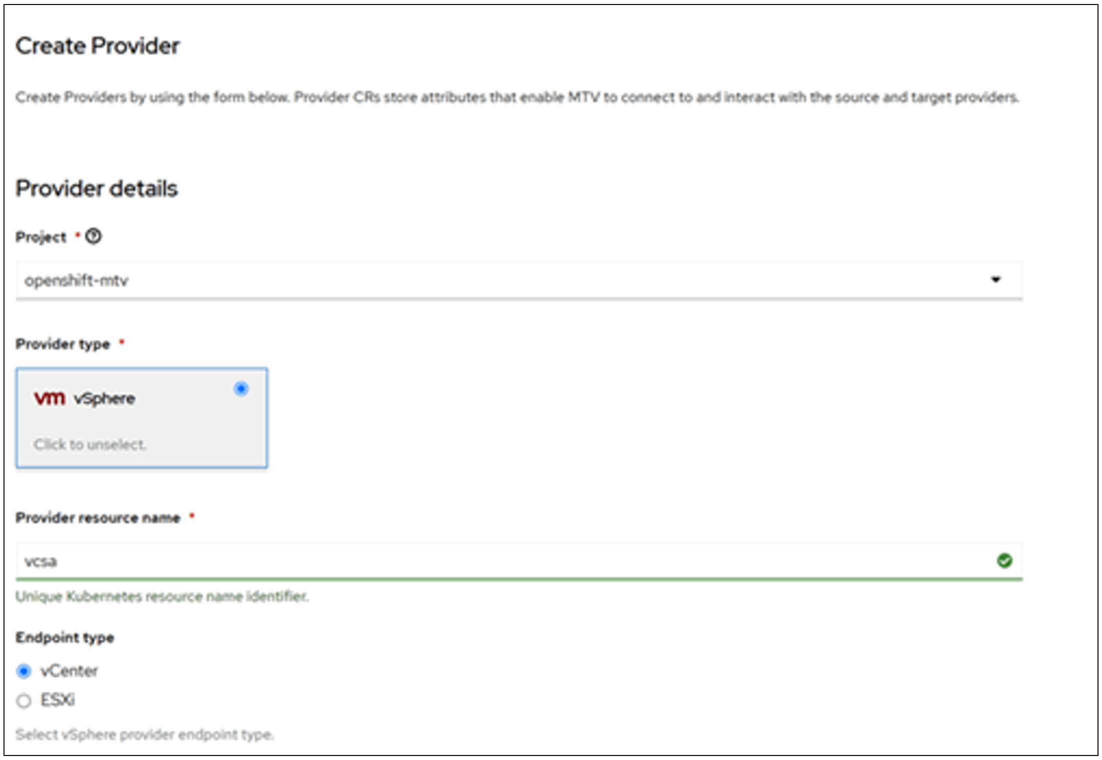

Figure 3-25 Creating a provider for virtualization

5. Provide the URL for vCenter, appending `/sdk` to the end. Then, enter the vCenter username and password, as shown in Figure 3-26.

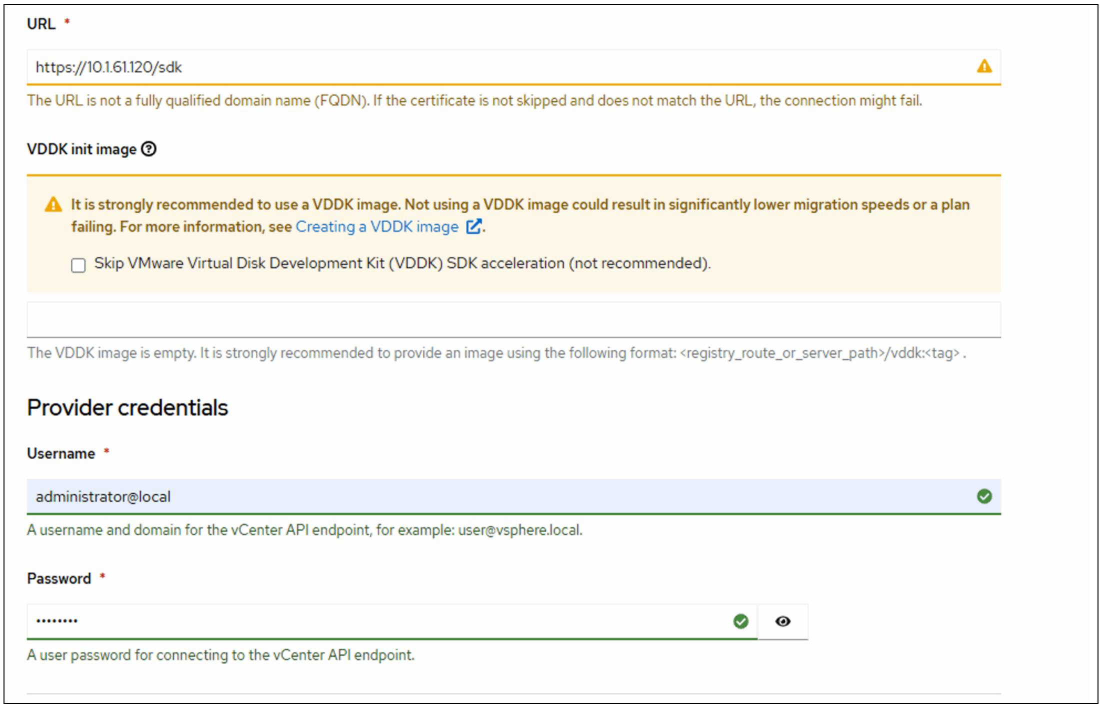

Figure 3-26 Inputs for the provider for visualization

6. In the Skip certificate validation section, skip the step and click **Create provider**, as shown in Figure 3-27.

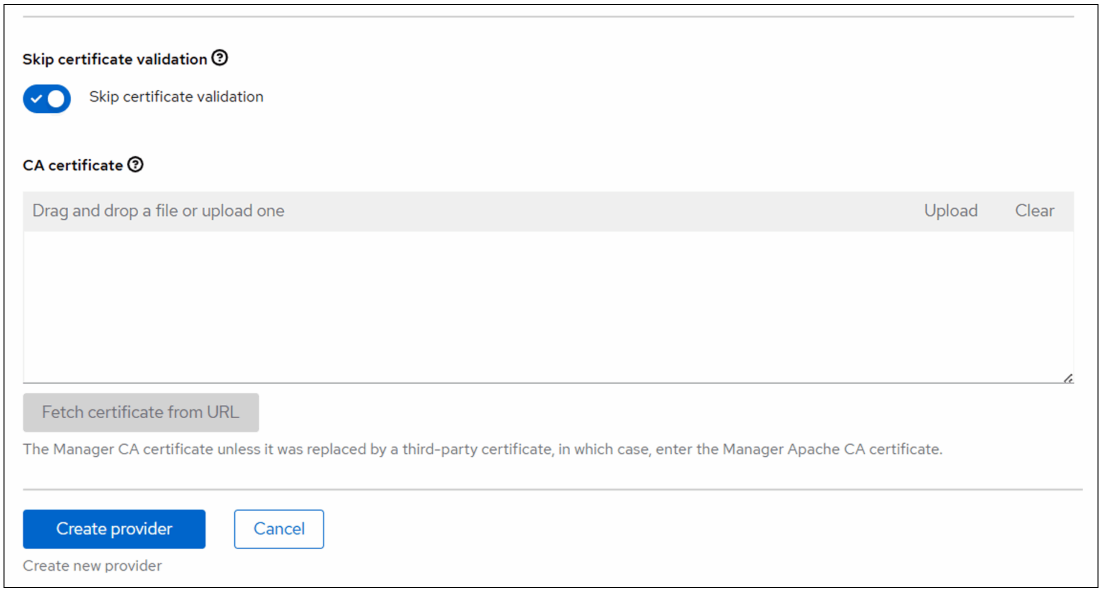

Figure 3-27 Inputs for the provider for visualization

The newly created provider (vcsa) is listed, and the count of ESXi hosts and VMs appears, as shown in Figure 3-28.

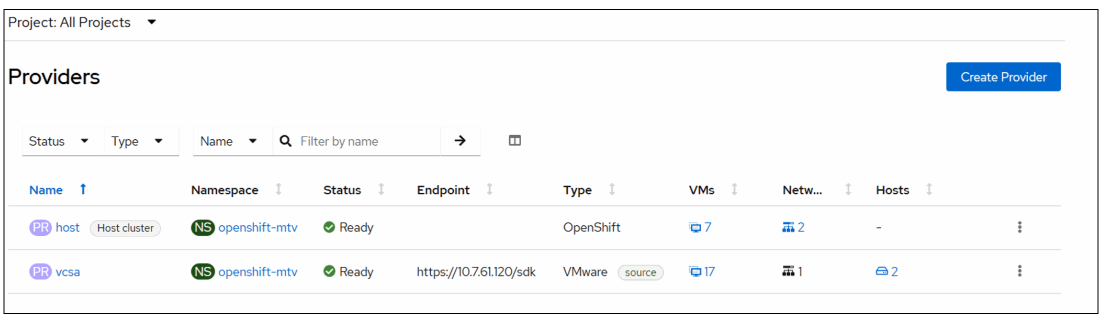

Figure 3-28 Provider for visualization listing

Creating a migration plan

You can migrate VMs into an existing namespace or create a namespace for the migration. In this section, you create a project, as shown in Figure 3-29 on page 45.

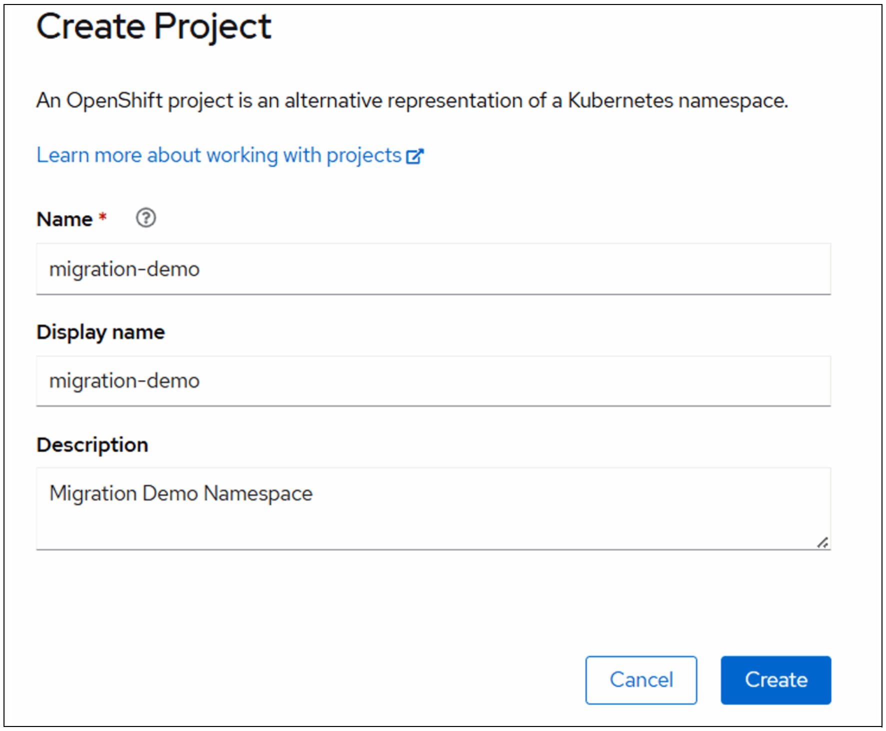

Figure 3-29 Create Project

Complete the following steps:

1. In the Red Hat OpenShift Container Platform web console, select **Migration** > **Plan for virtualization**, as shown in Figure 3-30.

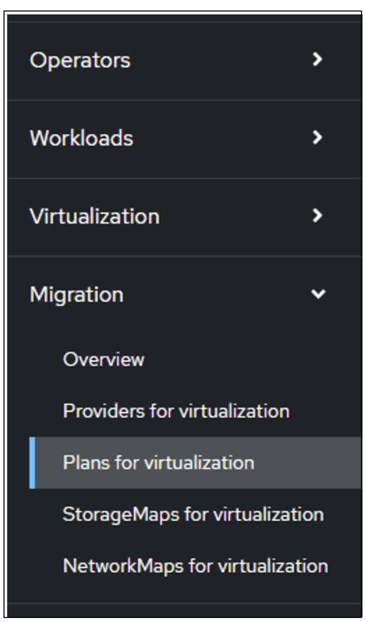

Figure 3-30 Plans for virtualization

2. Provide the plan name migrate-rh-vm, select the project **Red Hat OpenShift-mtv**, and choose the VM **vcsa** (a vSphere-type provider), as shown in Figure 3-31.

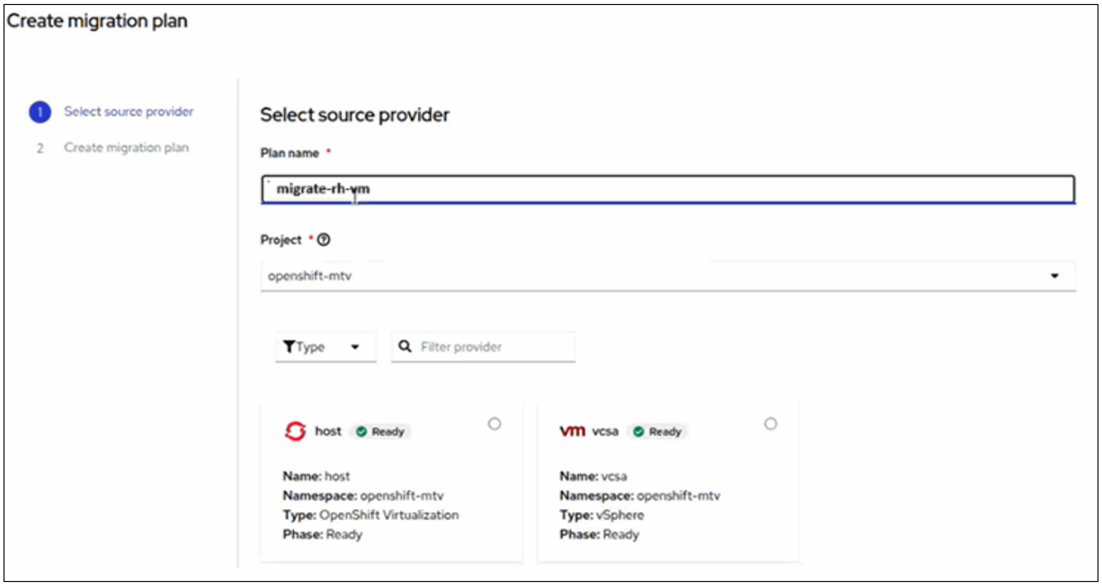

Figure 3-31 Creating a migration plan

3. After you select the VM vcsa (a vSphere-type provider), the **Select virtual machines** option becomes available, as shown in Figure 3-32.

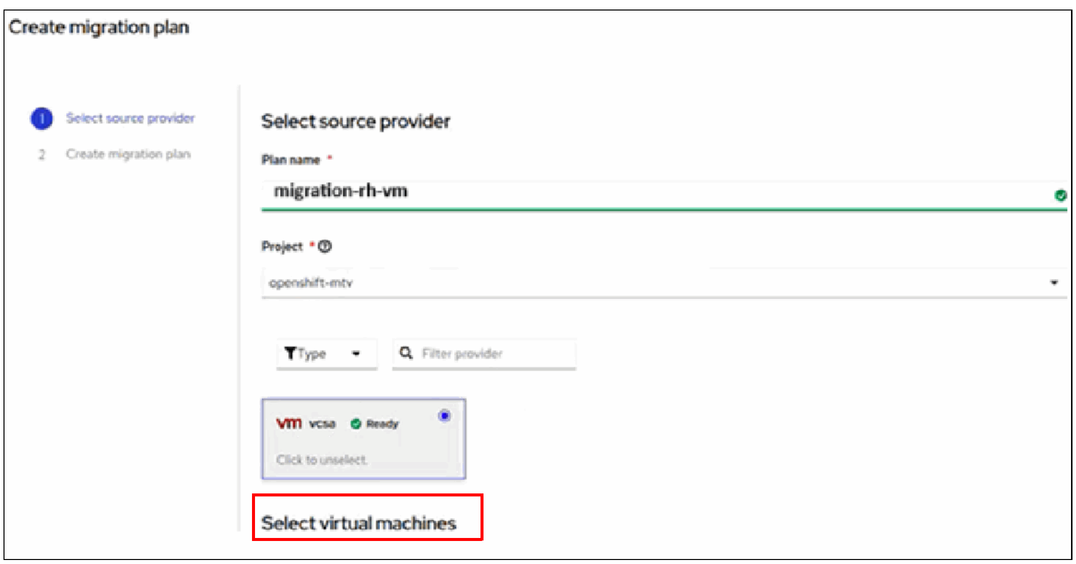

Figure 3-32 Select virtual machines option is enabled

In the next window, you see a list of VMs, as shown in Figure 3-33 on page 47.

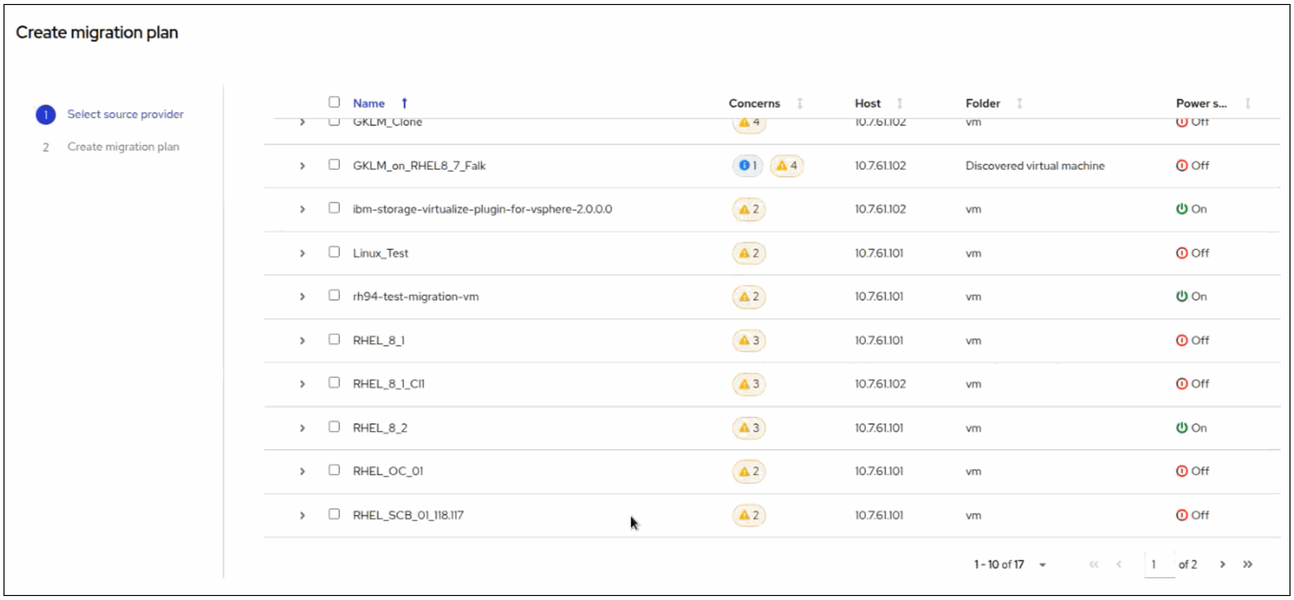

Figure 3-33 List of virtual machines

4. Before you select a VM, examine the concerns that are indicated for that VM. Figure 3-34 shows that the Change Block Tracking (CBT) feature must be enabled on the VM.

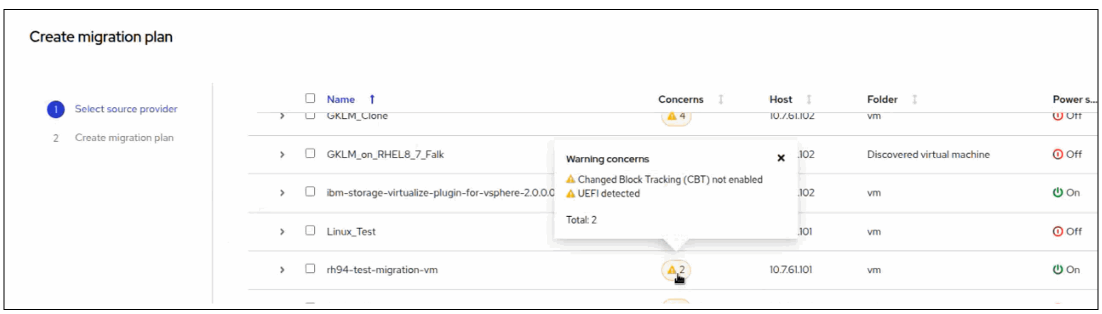

Figure 3-34 Virtual machines “concerns” warning with CBT

5. To enable this feature, you must log in to vCenter, select the VM, and edit its settings. In the settings, select **VM Options** → **Advanced**, and select **Configuration Parameters**. Click **Edit Configuration**, and then click **Add Configuration Parameters**. Enter ctkEnabled with a value of TRUE and save the configuration.

   Delete all snapshots from the targeted VM. After completing these steps, the CBT warning should disappear, as shown in Figure 3-35.

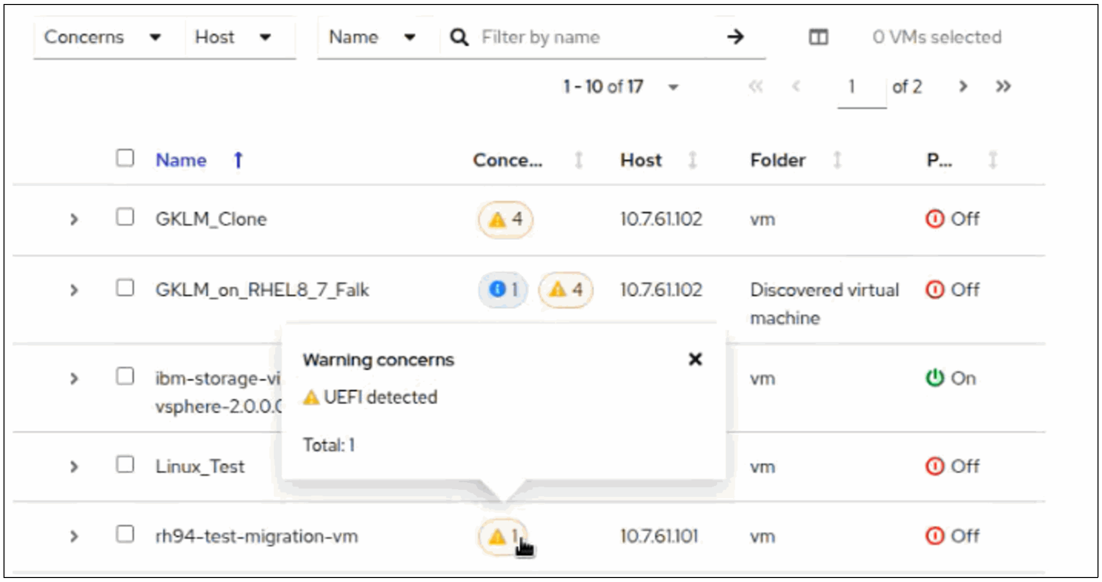

Figure 3-35 Virtual machines “concerns: warning without CBT

6. Select the target VM by selecting the checkbox next to it, and then click **Next**, as shown in Figure 3-36.

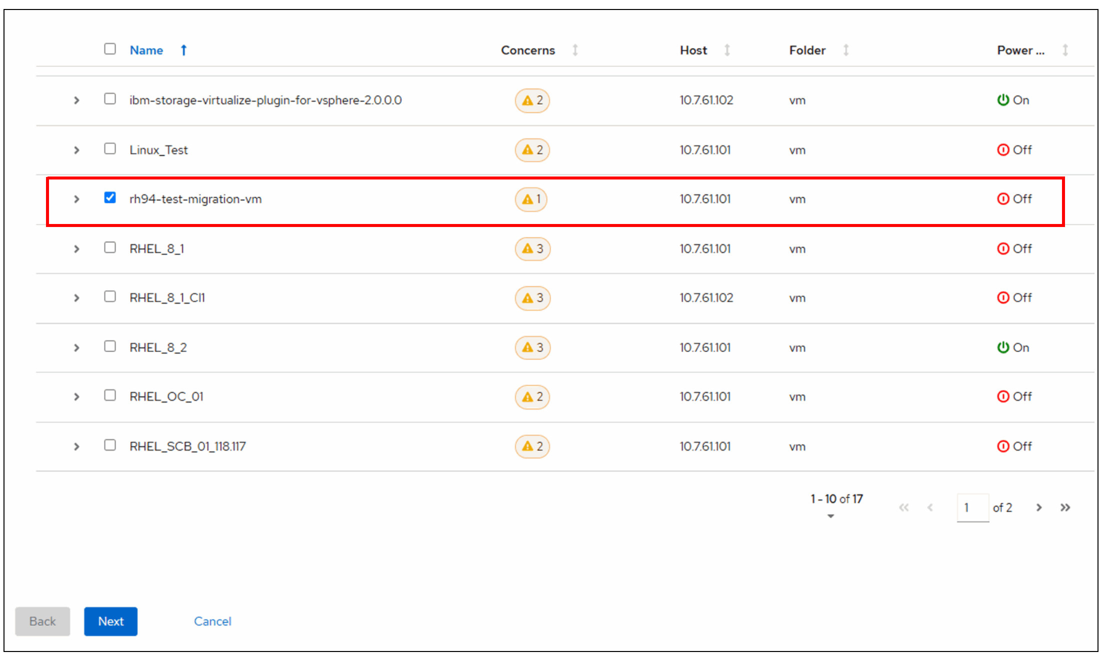

Figure 3-36 Selecting the target VM

7. Choose the Target namespace. For this example, you select **migration-demo**, as illustrated in Figure 3-37 on page 49.

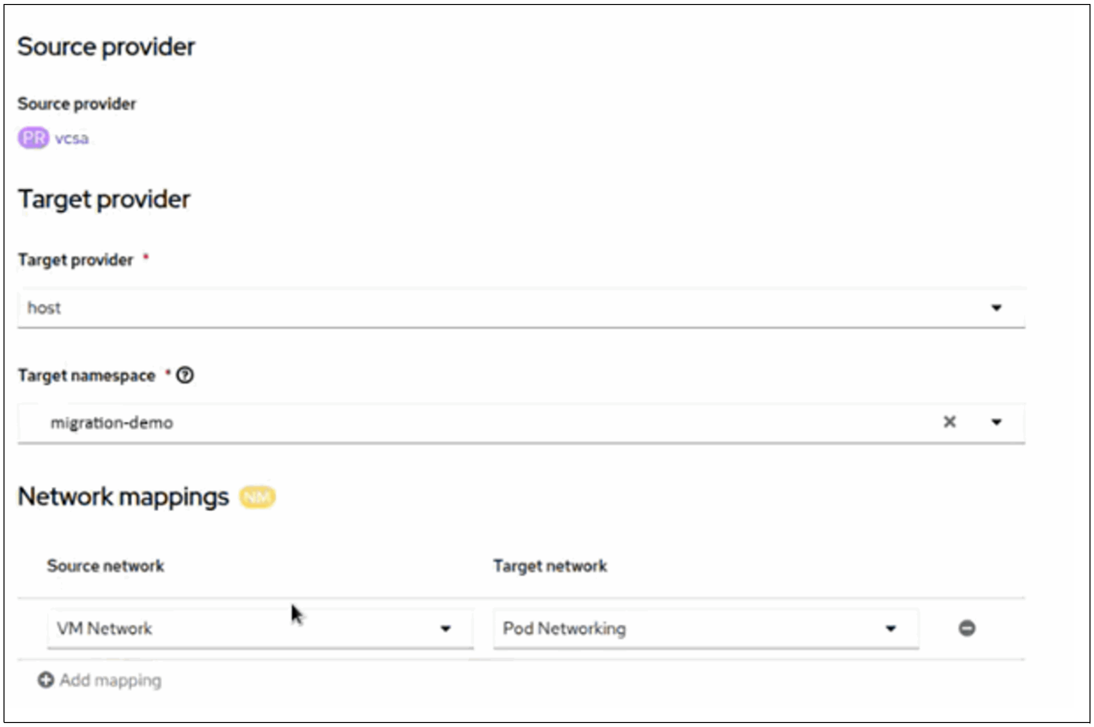

Figure 3-37 Defining the target namespace for migration

8. Choose the Source storage and Target storage, and then proceed by clicking **Create migration plan**, as illustrated in Figure 3-38.

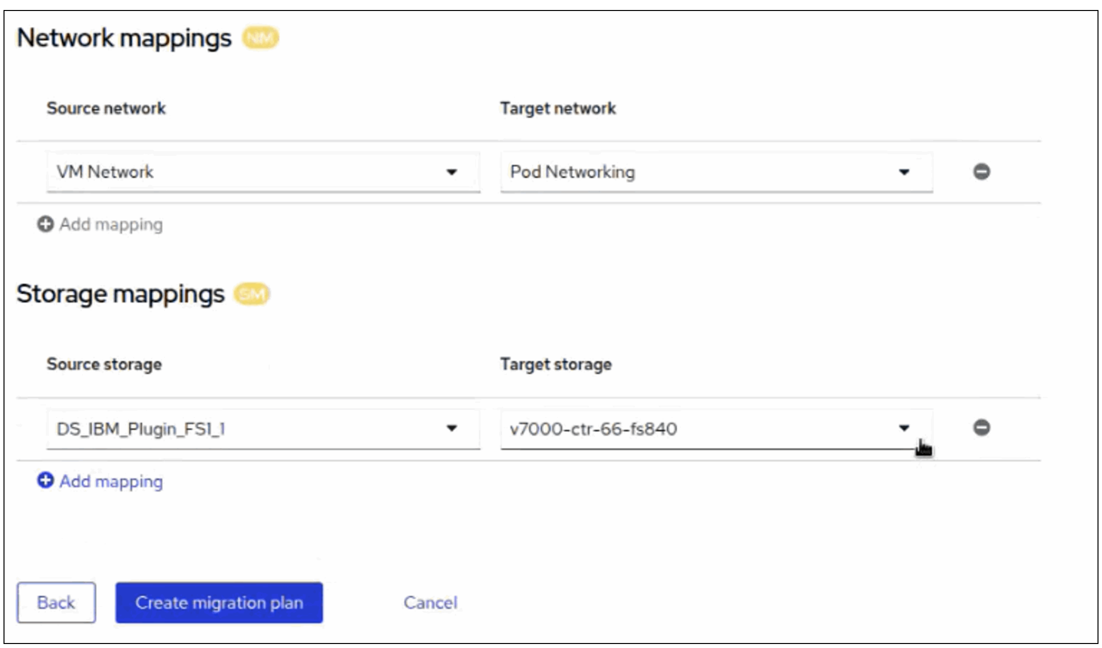

Figure 3-38 Creating the migration plan

The migration plan progress appears in the next window, as shown in Figure 3-39.

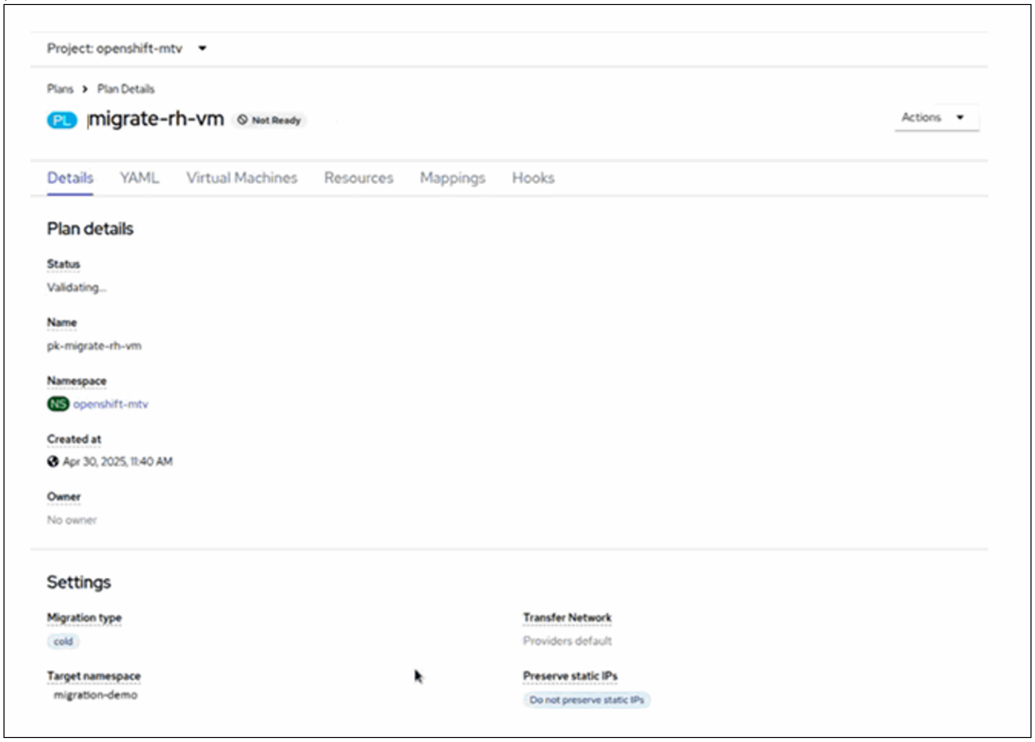

Figure 3-39 Migration plan: Creation status

After validation, the migration plan enters the Ready state, as shown in Figure 3-40.

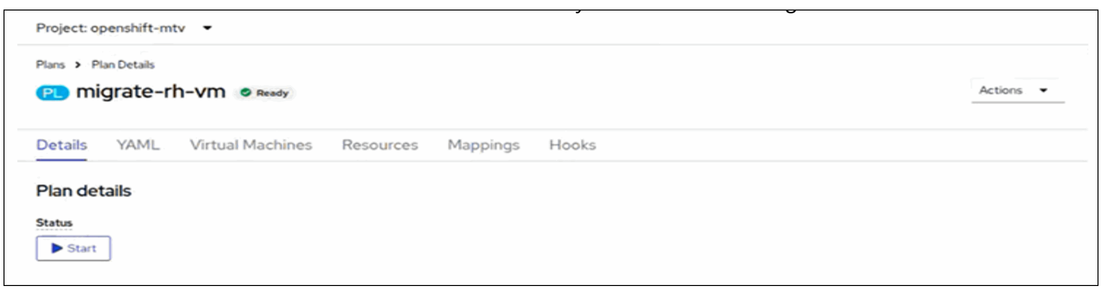

Figure 3-40 Migration plan: Ready state

9. Next, click **Migration plans** in the left menu. All migration plans appear, as shown in Figure 3-41.

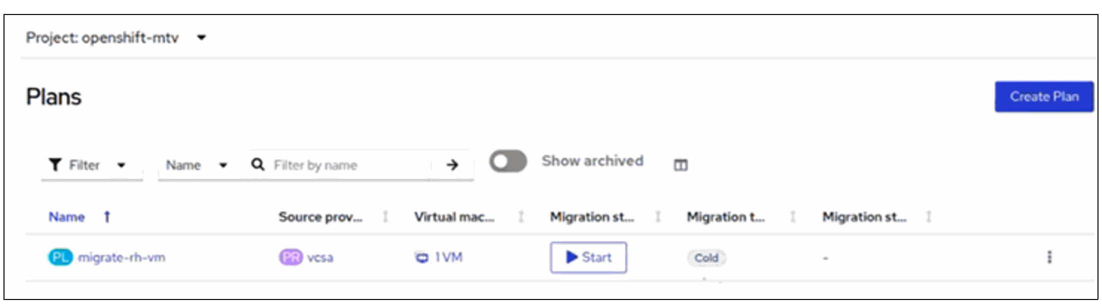

Figure 3-41 Migration plans menu

10. Click **Start** on the migration plan. A Start Confirmation pop-up appears, as shown in Figure 3-42. Click **Start**.

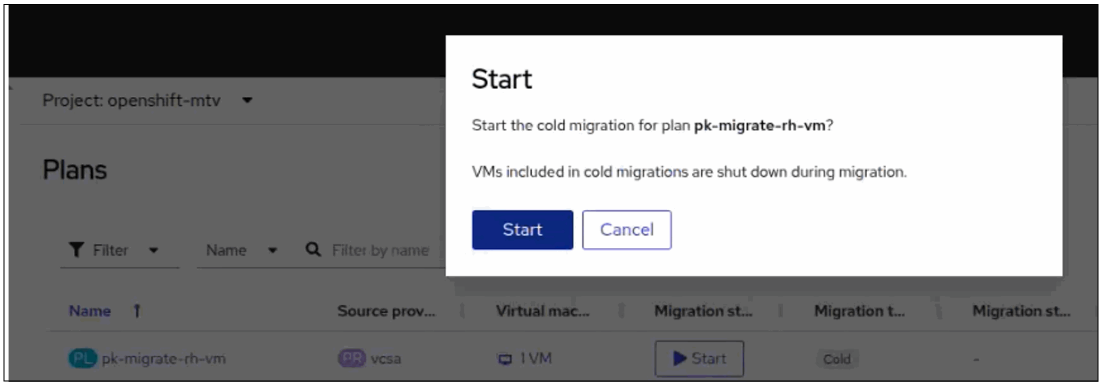

Figure 3-42 Starting the migration plan

Figure 3-43 shows the status of the migration plan that you started.

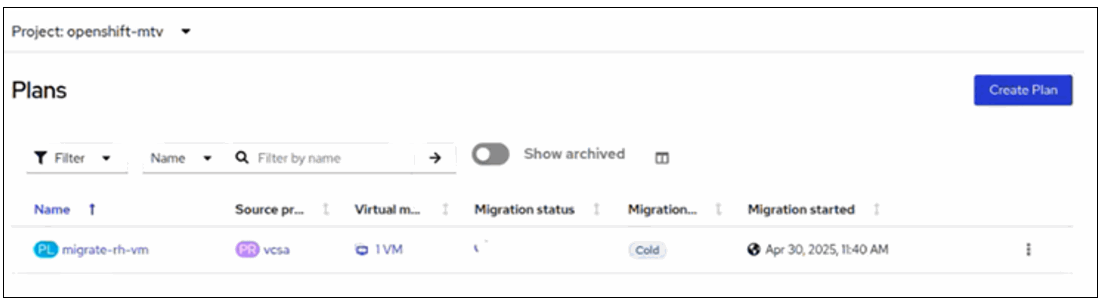

Figure 3-43 The status of the started migration plan

11.To view the detailed migration plan status, click the migration plan. The details are shown in Figure 3-44.

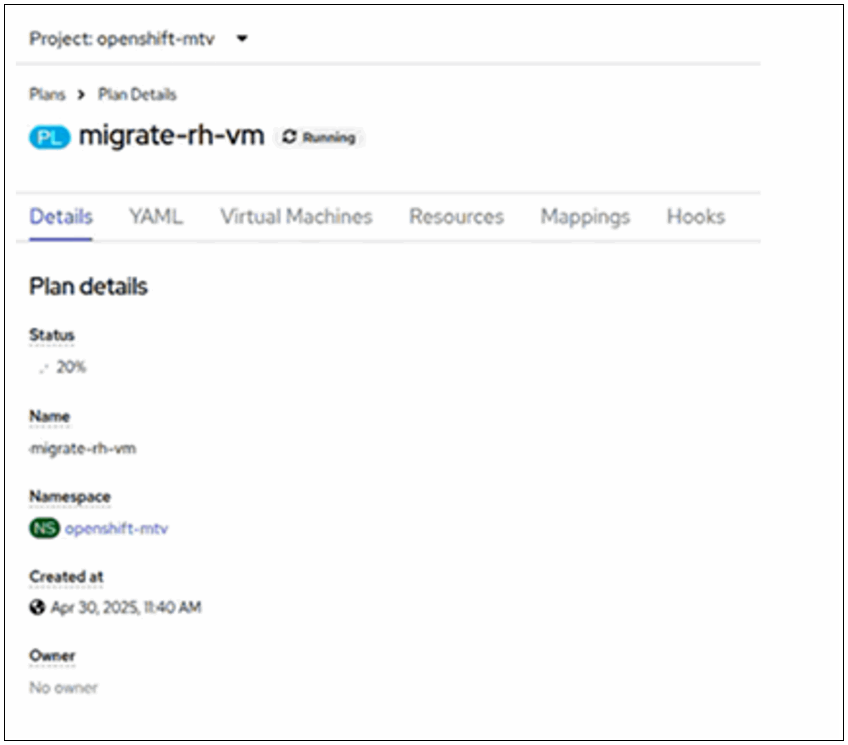

Figure 3-44 Detailed migration plan information

After the migration completes, the status changes from Running to Succeeded, as shown in Figure 3-45 on page 53.

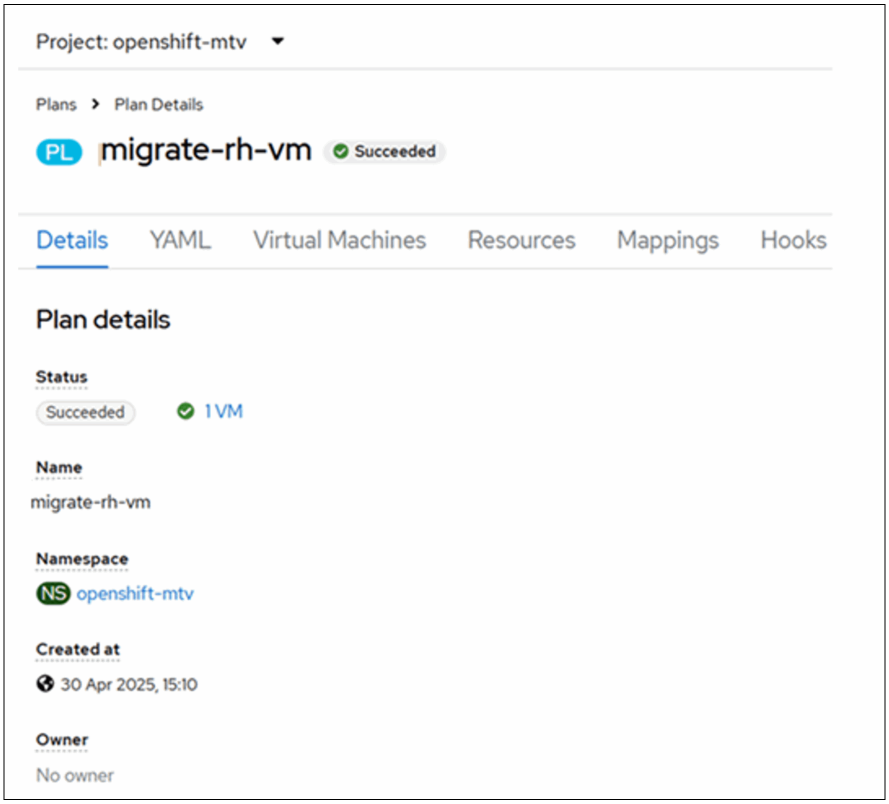

Figure 3-45 Success status of the migration plan

12. In the Red Hat OpenShift Container Platform web console, select **Virtualization** → **VirtualMachines** to check for the migrated VM in Red Hat OpenShift. Then, click the three dots next to the VM and select **Start**, as shown in Figure 3-46.

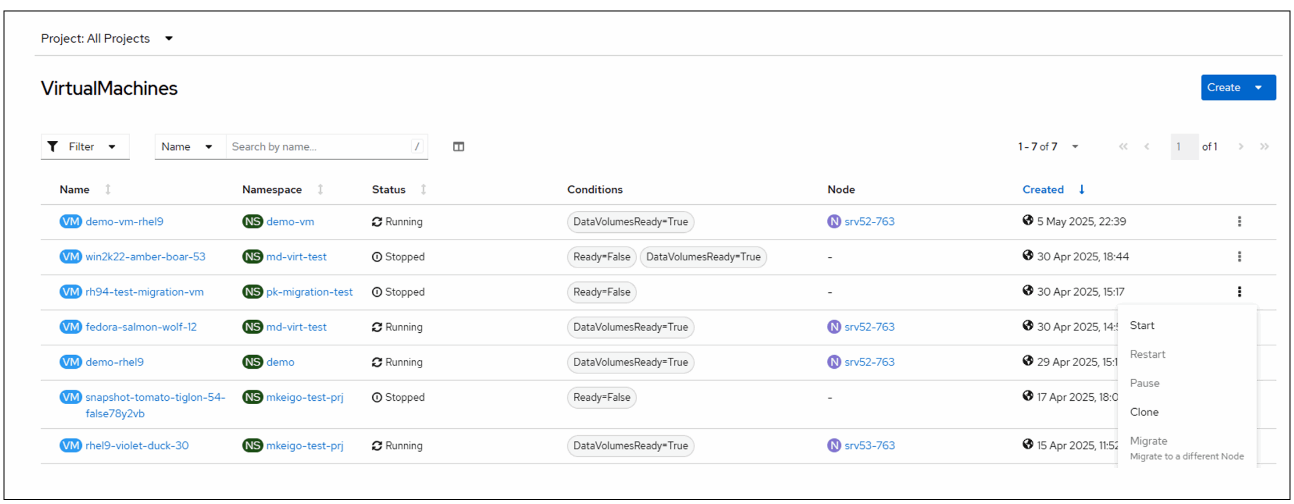

Figure 3-46 Starting the migrated VM

13.Click the VM. The VM details appear, and the status shows Running, as shown in Figure 3-47.

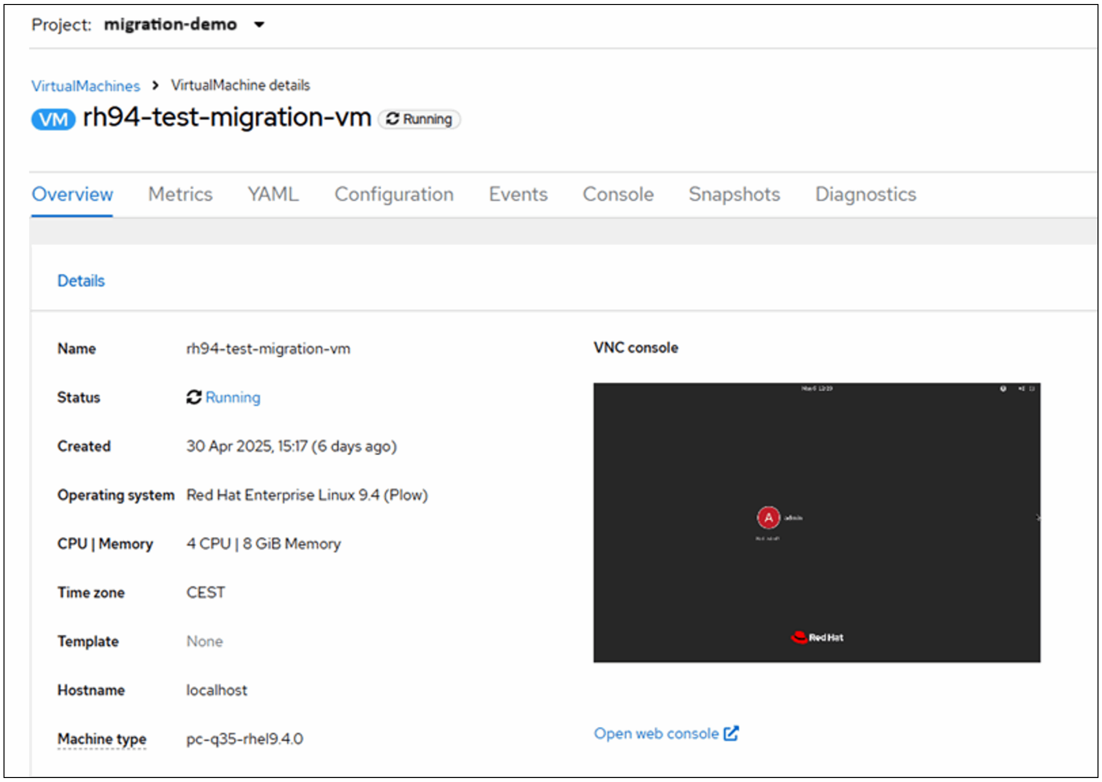

Figure 3-47 The virtual machine is running successfully after the migration

3.3 Day-2 operations: Part 1

This section explains typical Day-2 operations that support operational resiliency, such as taking VM snapshots, restoring a VM from a snapshot, deleting snapshots, cloning a VM, and creating a VM from another VM's snapshot.

3.3.1 Provisioning a VM

Figure 3-48 on page 55 shows a VM running a copy of the RHEL 9 operating system (OS) created on Red Hat OpenShift Virtualization.

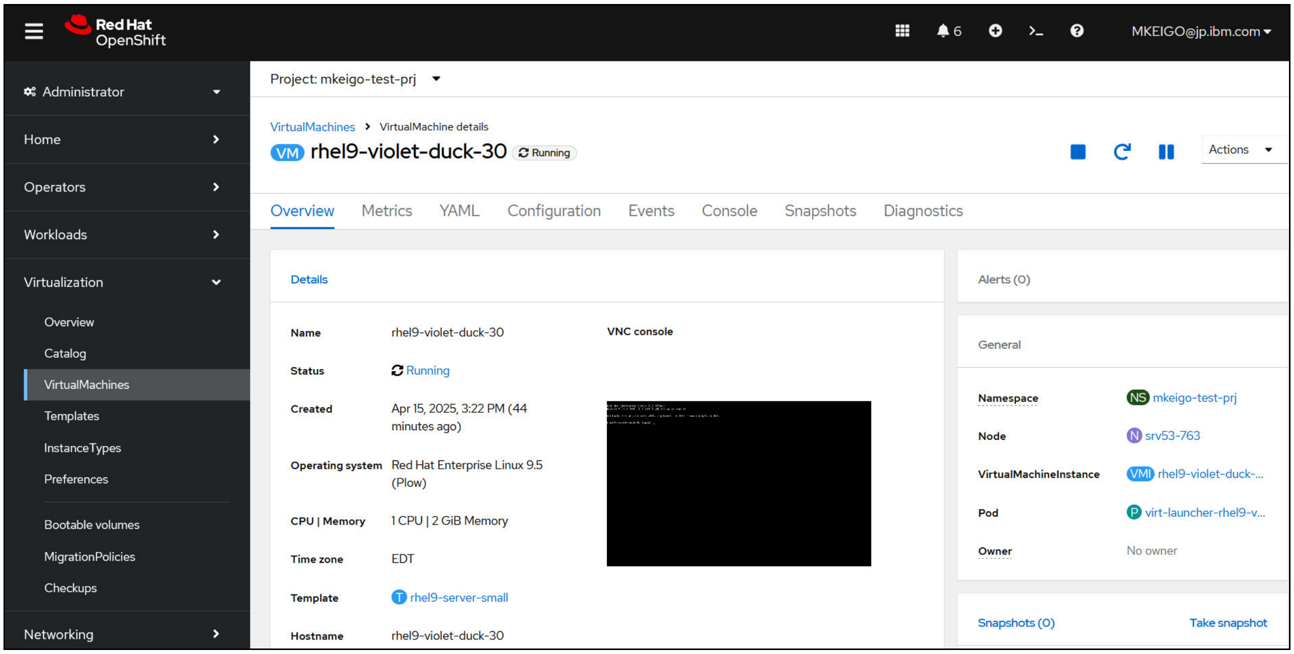

Figure 3-48 A VM is newly created on Red Hat OpenShift Virtualization

This VM instance, named RHEL 9-violet-duck-30 (see the Name field in Figure 3-48) runs in a pod within the project mkeigo-test-prj (Namespace). The pod named virt-launcher-RHEL 9-violet-duck-30-4zz86 (Pod) is provisioned on the server srv53 (Node). The VM is created from the template RHEL 9 server small (Template), and a virtual disk is allocated (not shown in the figure, but visible in the Storage field).

Example 3-1 shows the excerpt of the YAML definition for this VM, including its virtual disk name and UID (unique ID), retrieved by using the **oc** command after setting the project to the one where the VM was created. You can also confirm this information by clicking the **YAML** tab.

Example 3-1 Excerpt of the YAML definition for the created VM (kind, name, and uid)

```
$ oc project mkeigo-test-prj
$ oc get virtualmachine RHEL 9-violet-duck-30
NAME                      AGE   STATUS    READY
RHEL 9-violet-duck-30     10d   Running   True
$ oc get virtualmachine RHEL 9-violet-duck-30 -o yaml | egrep '(kind|name)'
kind: VirtualMachine
          "name": "minimal-required-memory",
    vm.kubevirt.io/template.namespace: Red Hat OpenShift
  name: RHEL 9-violet-duck-30
  namespace: mkeigo-test-prj
  uid: fd0aa886-cc2d-4f36-9aee-5c8f27620a4e
    kind: DataVolume
      name: restore-286cb8f6-ca17-44a9-b4b5-db2316ab4268-rootdisk
        kind: DataSource
        name: RHEL 9
        namespace: Red Hat OpenShift-virtualization-os-images
            name: rootdisk
            name: cloudinitdisk
            name: default
      - name: default
```

```
          name: restore-286cb8f6-ca17-44a9-b4b5-db2316ab4268-rootdisk
        name: rootdisk
        name: cloudinitdisk
    name: rootdisk
    name: cloudinitdisk
```

In the Red Hat OpenShift management GUI, select **Storage** → **PersistentVolumeClaims**. Enter the created VM name in the Name field. Figure 3-49 shows the bound status between the PVC that is named RHEL 9-voilet-duck-30 and the PV that is named pvc-8e7c845b-aa42-4557-af8c-2b873f8cebbd, which is created in the StorageClass that is named fs5200-1.

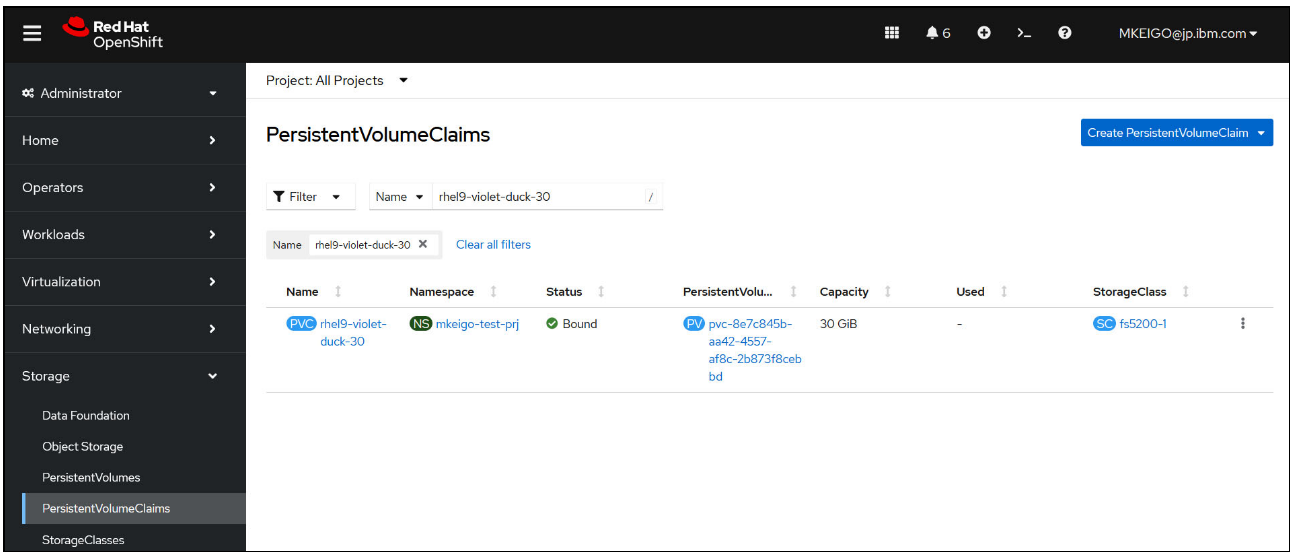

Figure 3-49 PVC for the newly created VM

Example 3-2 shows the volume name, which matches the PV name, created on the IBM FlashSystem 5200-1 for this VM.

Example 3-2 Showing the volume that is named after the PV name for this VM that was created on IBM FlashSystem 5200-1

```
IBM_FlashSystem:FS5200-1:superuser>lsvdisk | head -1; lsvdisk | grep
'pvc-8e7c845b'
id name                                    IO_group_id IO_group_name status
mdisk_grp_id mdisk_grp_name capacity type  FC_id FC_name RC_id RC_name vdisk_UID
fc_map_count copy_count fast_write_state se_copy_count RC_change
compressed_copy_count parent_mdisk_grp_id parent_mdisk_grp_name owner_id
owner_name formatting encrypt volume_id volume_name
function volume_group_id volume_group_name  protocol is_snapshot snapshot_count
volume_type replication_mode is_safeguarded_snapshot safeguarded_snapshot_count
restore_in_progress
13 CSI_pvc-8e7c845b-aa42-4557-af8c-2b873f8cebbd 0          io_grp0       online
1            Pool_0_Child_1 30.00 GB  striped
600507680B8607FE4000000000000995 0           1         not_empty       1
no       0                   0                 Pool0
no     no    13    CSI_pvc-8e7c845b-aa42-4557-af8c-2b873f8cebbd
scsi     no        0                                         no
0                           no
```

Figure 3-50 illustrates the relationship between the provisioned VM, its PVC, and the bound PV created on the IBM FlashSystem 5200-1.

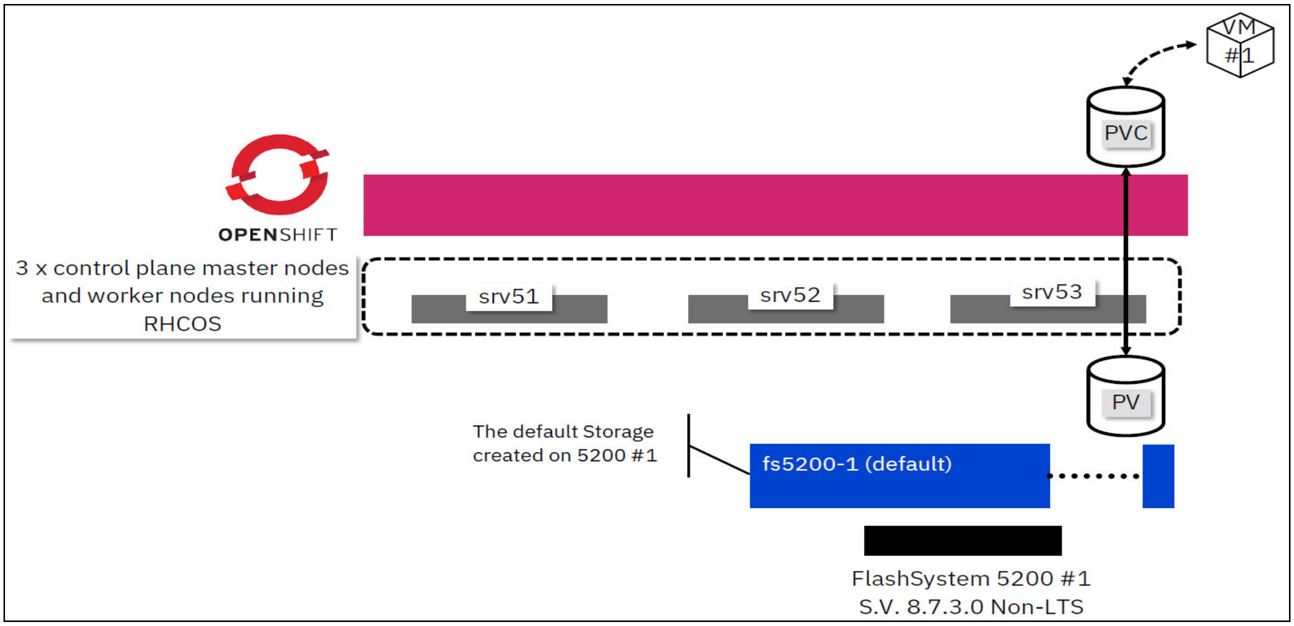

Figure 3-50 Provisioned VM, its PVC, and bounded PV that was created on IBM FlashSystem 5200-1

When you log in to this VM with valid user credentials through the console, the system identifies itself as follows (Example 3-3).

Example 3-3 The system details

```
$ uname -a
RHEL 9-violet-duck-30 5.14.0-503.11.1.el9_5.x86_64 #1 SMP PREEMPT_DYNAMIC Mon Sep
30 11:54:45 EDT 2024 x86_64 x86_64 x86_64 GNU/Linux
$ cat /etc/redhat-release
Red Hat Enterprise Linux release 9.5 (Plow)
```

3.3.2 Taking a VM snapshot

To take a snapshot of a VM in the Red Hat OpenShift management GUI, complete the following steps:

1. Select the VM.
2. Click the **Snapshots** tab.
3. Click **Take Snapshot**. A dialog window opens (see Figure 3-51).
4. Enter a name for the snapshot.
5. Click **Save**.

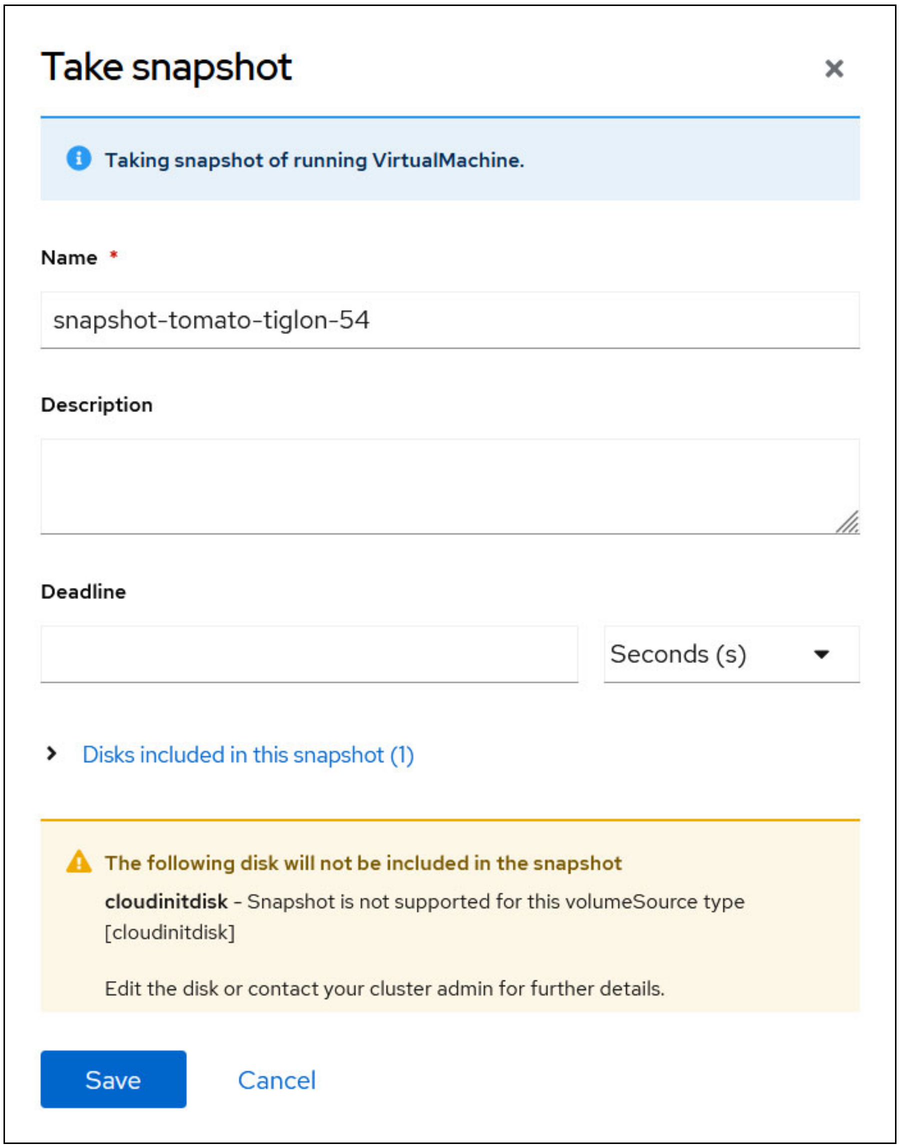

Figure 3-51 Taking a snapshot of a VM

Snapshots are typically created within seconds. After creation, they appear in the **Snapshots** tab, as shown in Figure 3-52 on page 59.

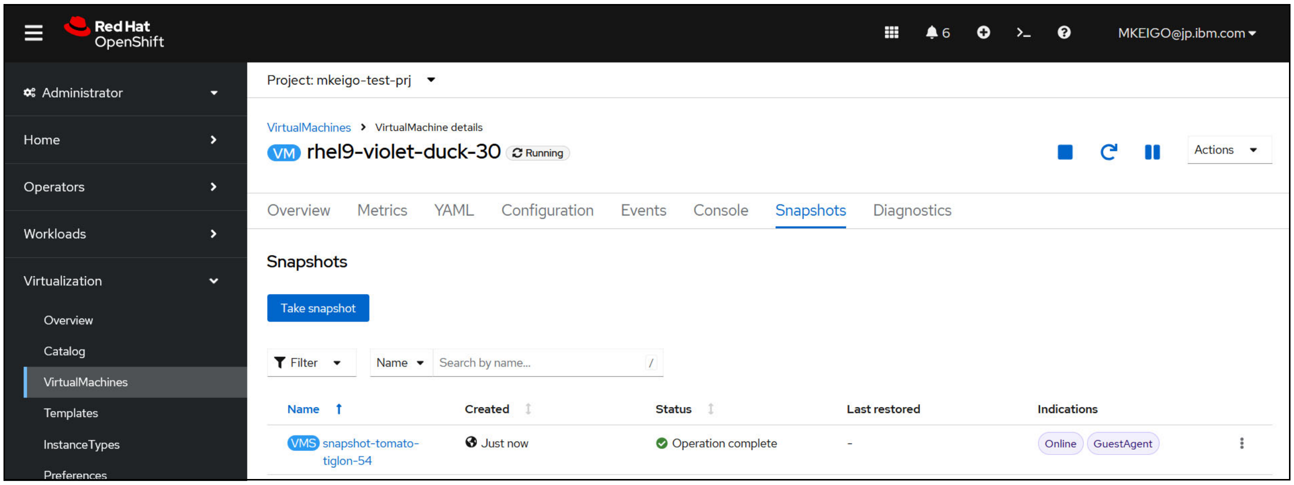

Figure 3-52 List of snapshots for a VM

Example 3-4 shows the command output that extracts relevant information for this snapshot.

Example 3-4 Information about the snapshot

```
$ oc get virtualmachinesnapshot snapshot-tomato-tiglon-54 -o yaml | egrep -i
'(name|uid)'
  name: snapshot-tomato-tiglon-54
  namespace: mkeigo-test-prj
    name: RHEL 9-violet-duck-30
    uid: fd0aa886-cc2d-4f36-9aee-5c8f27620a4e
  uid: a1eb5d37-5eec-421f-a366-c7509bbd7920
    name: RHEL 9-violet-duck-30
  sourceUID: fd0aa886-cc2d-4f36-9aee-5c8f27620a4e
  virtualMachineSnapshotContentName:
vmsnapshot-content-a1eb5d37-5eec-421f-a366-c7509bbd7920
```

This request uses IBM Block Storage CSI to initiate a Volume Group Snapshot (VGS) on the IBM FlashSystem 5200-1. Example 3-5 illustrates the mapping for this VGS. Because the `copy_rate` value is set to zero, the operation never completes. You can identify the `snapshot_id` as *1* and the `mapping_id` as 15863, based on the source volume name matching the rootdisk volume name that is shown in Example 3-2 on page 56.

Example 3-5 VGS mapping on IBM FlashSystem 5200-1 when taking a VM snapshot

```
IBM_FlashSystem:FS5200-1:superuser>lsvolumesnapshot | head -1; lsvolumesnapshot|
tail -1
snapshot_id snapshot_name                                     volume_id
volume_name                                  volume_group_id volume_group_name
parent_uid time         expiration_time state  safeguarded volume_size_mismatch
mirrored written_capacity
1           CSI_snapshot-d6d0502b-0b0c-49fb-a95e-4c46916ca53a 13
CSI_pvc-8e7c845b-aa42-4557-af8c-2b873f8cebbd                           41
250417094719                 active no          no                   no
21.50 MB
IBM_FlashSystem:FS5200-1:superuser>lsfcmap -showhidden | head -1; lsfcmap
-showhidden | grep 8e7c845b
id     name    source_vdisk_id source_vdisk_name
target_vdisk_id target_vdisk_name group_id group_name status  progress copy_rate
```

```
clean_progress incremental partner_FC_id partner_FC_name restoring start_time
rc_controlled size_mismatch is_snapshot snapshot_id
15863 fcmap0 13          CSI_pvc-8e7c845b-aa42-4557-af8c-2b873f8cebbd 15863
vdisk3                                      copying 0        0          100
off                                       no       250417094719 no          no
yes         1
IBM_FlashSystem:FS5200-1:superuser>lsfcmap -showhidden 15863 | grep status
status copying
IBM_FlashSystem:FS5200-1:superuser>lsfcmap -showhidden 15863 | grep progress
progress 0
clean_progress 100
restore_progress 0
IBM_FlashSystem:FS5200-1:superuser>lsfcmap -showhidden 15863 | grep copy_rate
copy_rate 0
IBM_FlashSystem:FS5200-1:superuser>lsfcmap -showhidden 15863 | grep keep_target
keep_target yes
```

Note: Specify the `-showhidden` option when you run the `lsfcmap` command to display maps for VGS.

Figure 3-53 illustrates the relationship between the PVC and its snapshot, along with the volume bound to the PV and the snapshot created by the VGS on the IBM FlashSystem 5200-1.

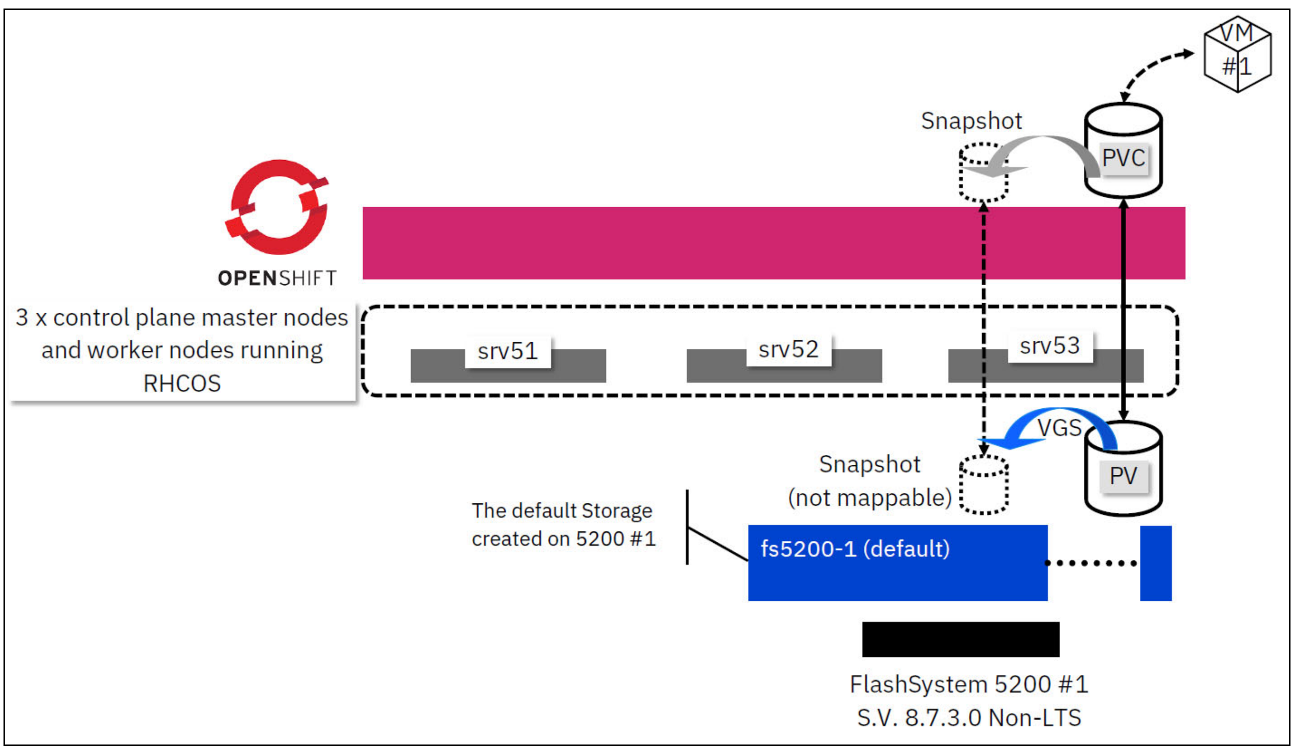

Figure 3-53 A VM's PVC and its bounded PV, snapshot, and snapshot that is created by VGS on IBM FlashSystem 5200-1

3.3.3 Deleting a VM snapshot

To delete a snapshot, complete the following steps:

1. Click the three vertical dots next to the snapshot (see Figure 3-54).
2. Select **Delete Snapshot** from the menu. A confirmation window appears.
3. Click **Delete Snapshot** to confirm and delete the snapshot.

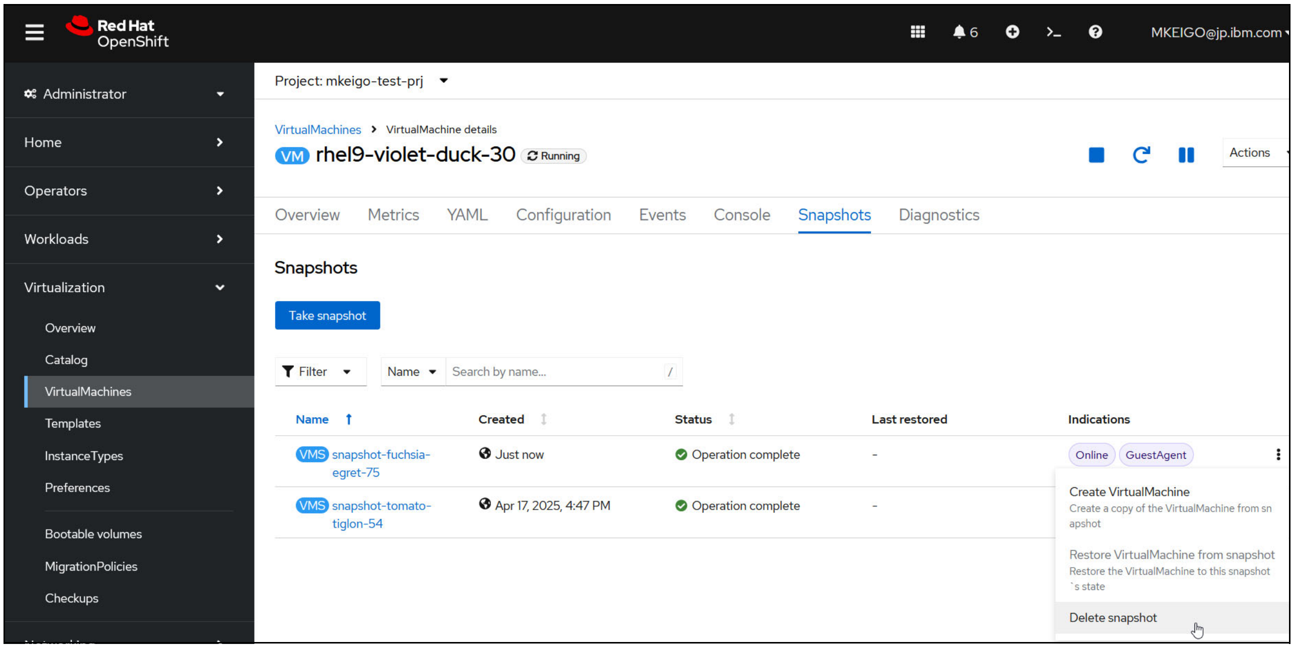

Figure 3-54 Deleting a snapshot for a VM

3.3.4 Restoring a VM from a snapshot

Before you restore a snapshot, you must shut down the VM. In the Red Hat OpenShift management GUI, complete the following steps:

1. Select the specific VM.
2. From the **Actions** menu, click **Stop**.
3. After you shut down the VM, click the **Snapshots** tab.
4. Select the snapshot that you want to restore.
5. Click the three vertical dots next to the snapshot.

6. Select **Restore Virtual Machine from Snapshot**.
7. After you click **Restore**, a confirmation window appears (see Figure 3-55). The system then restores the VM from the selected snapshot.

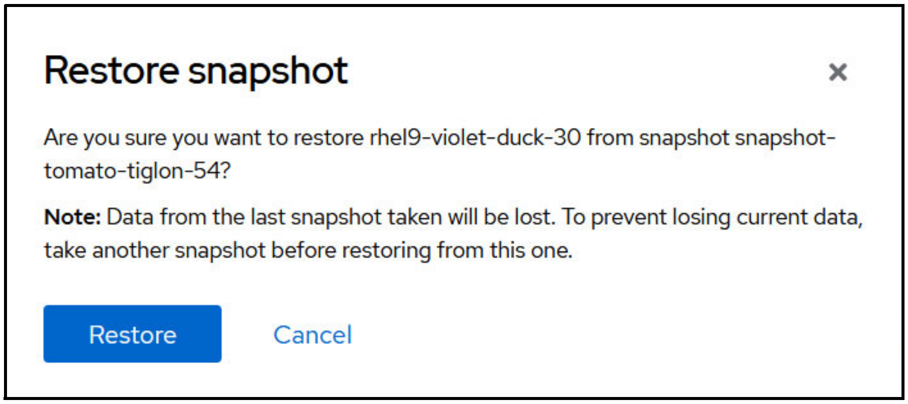

Figure 3-55 Restore snapshot

> **Note:** Restoring the VM from a snapshot reverts the data to the state that it was in when the snapshot was taken. Any changes made to the VM after that point are lost.

This request uses IBM Block Storage CSI to initiate VGS on the IBM FlashSystem 5200-1. Example 3-6 shows the reverse mapping 15862, which the system creates to support this restore-from-snapshot request.

Example 3-6 Showing VGS mappings on IBM FlashSystem 5200-1 when restoring a VM from snapshot

```
IBM_FlashSystem:FS5200-1:superuser>lsfcmap -showhidden | head -1; lsfcmap
-showhidden | tail -2
id    name   source_vdisk_id source_vdisk_name
target_vdisk_id target_vdisk_name                            group_id group_name
status  progress copy_rate clean_progress incremental partner_FC_id
partner_FC_name restoring start_time   rc_controlled size_mismatch is_snapshot
snapshot_id
15862 fcmap1 15863         vdisk3                                  9
CSI_pvc-48625bb3-1b3e-4238-a6cb-50e97276856b                    copying 4
50        100          off                                        no
250417122126 no           no            no
15863 fcmap0 13          CSI_pvc-8e7c845b-aa42-4557-af8c-2b873f8cebbd 15863
vdisk3                                                          copying 0
0         100          off                                        no
250417094719 no           no            yes          1
```

The system deletes the mapping after the background copy completes. This action occurs when the progress value reaches 100.

Example 3-7 The progress value reaches 100

```
IBM_FlashSystem:FS5200-1:superuser>lsfcmap -showhidden 15862 | grep progress
progress 27
clean_progress 100
restore_progress 0
```

3.3.5 Cloning a VM

To clone a VM in the Red Hat OpenShift management GUI, complete the following steps:

1. Select the source VM.
2. From the **Actions** drop-down menu, choose **Clone**. A dialog window appears (see Figure 3-56).
3. Enter a name for the new VM, if needed.
4. Click **Clone**.

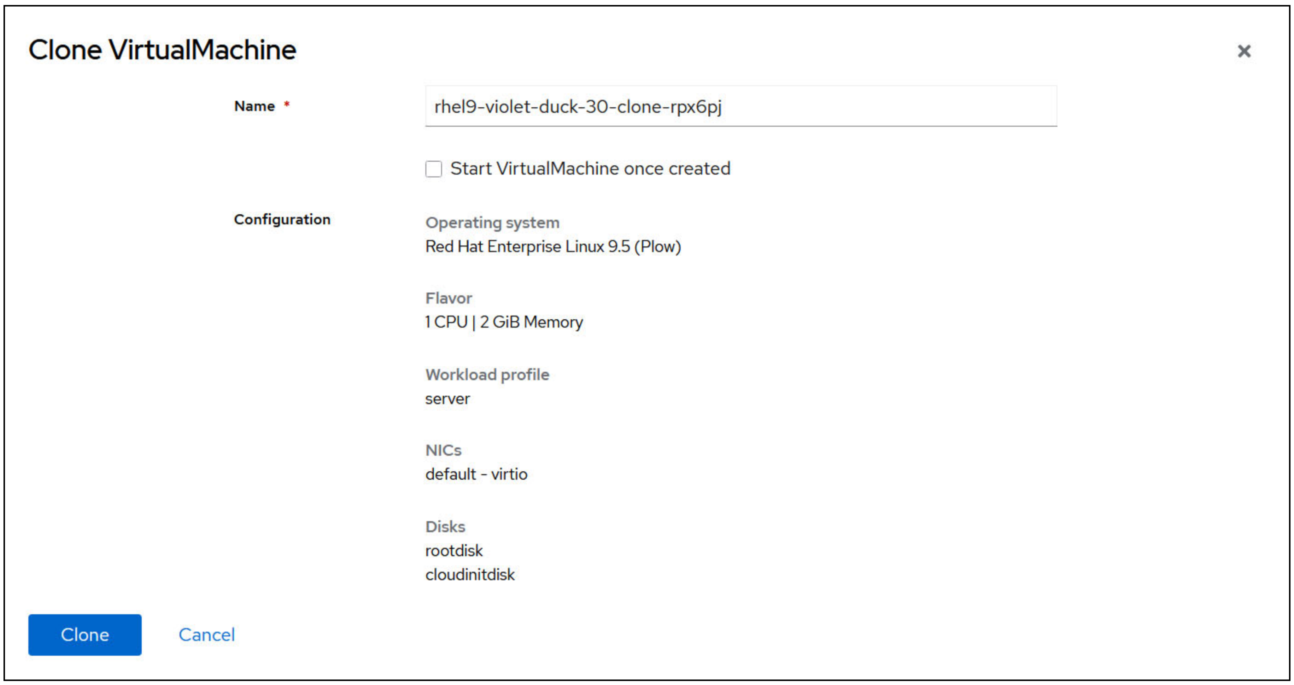

Figure 3-56 Cloning a VM

> **Note:** Select the **Start VirtualMachine once created** checkbox if you want to start the cloned VM immediately after creation.

3.3.6 Create a VM from another VM’s snapshot

To create a VM from another VM’s snapshot, complete the following steps in the Red Hat OpenShift management GUI:

1. Click the **Snapshots** tab.
2. Choose the snapshot that you want to restore.
3. Click the three vertical dots at the far right of the snapshot.
4. Select **Create VirtualMachine** from the drop-down menu.

5. Ensure that the source VM is selected.
6. A dialog window appears (see Figure 3-57). Enter a name for the new VM, if needed, and then click **Create**.

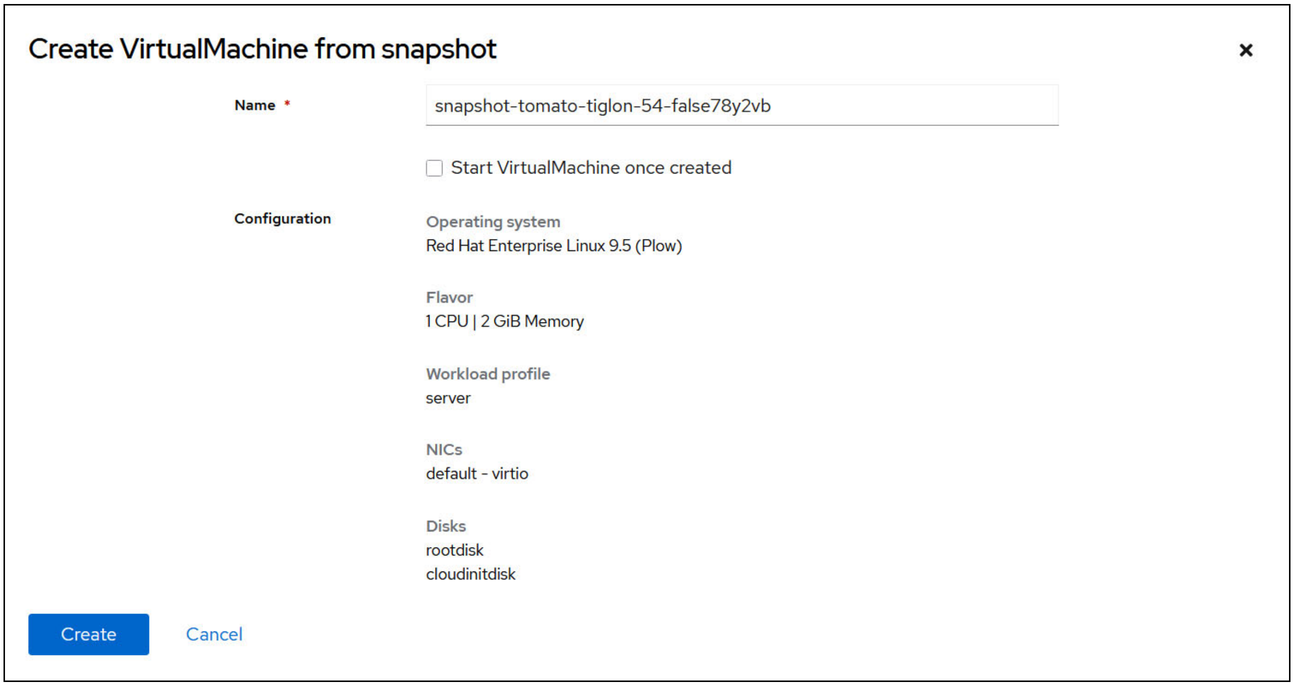

Figure 3-57 Create VirtualMachine from snapshot window

> **Note:** Select the **Start VirtualMachine once created** checkbox if you want to start the cloned VM immediately after it is created.

This request uses IBM Block Storage CSI to initiate VGS on IBM FlashSystem 5200-1. Example 3-8 shows mapping 15982, which the system creates to support the request to create a VM from another VM's snapshot.

Example 3-8 VGS mappings on IBM FlashSystem 5200-1 when creating a VM from another VM's snapshot

```
IBM_FlashSystem:FS5200-1:superuser>lsfcmap -showhidden | head -1; lsfcmap
-showhidden | tail -2
id    name   source_vdisk_id source_vdisk_name
target_vdisk_id target_vdisk_name                            group_id group_name
status  progress copy_rate clean_progress incremental partner_FC_id
partner_FC_name restoring start_time   rc_controlled size_mismatch is_snapshot
snapshot_id
15982 fcmap1 15863         vdisk3                                   16
CSI_pvc-1ef7e868-1f54-4f20-b7ea-a1eda1486a74                     copying 4
50       100          off                                         no
250417143230 no           no            no
15863 fcmap0 13         CSI_pvc-8e7c845b-aa42-4557-af8c-2b873f8cebbd 15863
vdisk3                                                           copying 0
0        100          off                                         no
250417094719 no           no            yes          1
```

The system deletes the mapping after the background copy completes. This action occurs when the progress value reaches 100 (see Example 3-9).

Example 3-9 The progress value reaches 100

```
IBM_FlashSystem:FS5200-1:superuser>lsfcmap -showhidden 15982 | grep progress
progress 67
clean_progress 100
restore_progress 0
```

Example 3-10 shows the source volume name for the newly created VM's volume ID 16. This volume name is the same one that is used by the source VM, as shown in Example 3-2 on page 56.

Example 3-10 The source volume for a newly created VM's volume

```
IBM_FlashSystem:FS5200-1:superuser>lsvdisk 16 | grep volume_name
volume_name CSI_pvc-1ef7e868-1f54-4f20-b7ea-a1eda1486a74
source_volume_name CSI_pvc-8e7c845b-aa42-4557-af8c-2b873f8cebbd
```

Figure 3-58 illustrates the relationship between the source and created VMs' PVCs and bounded PVs, showing how they are snapshotted by VGS and cloned on IBM FlashSystem 5200-1.

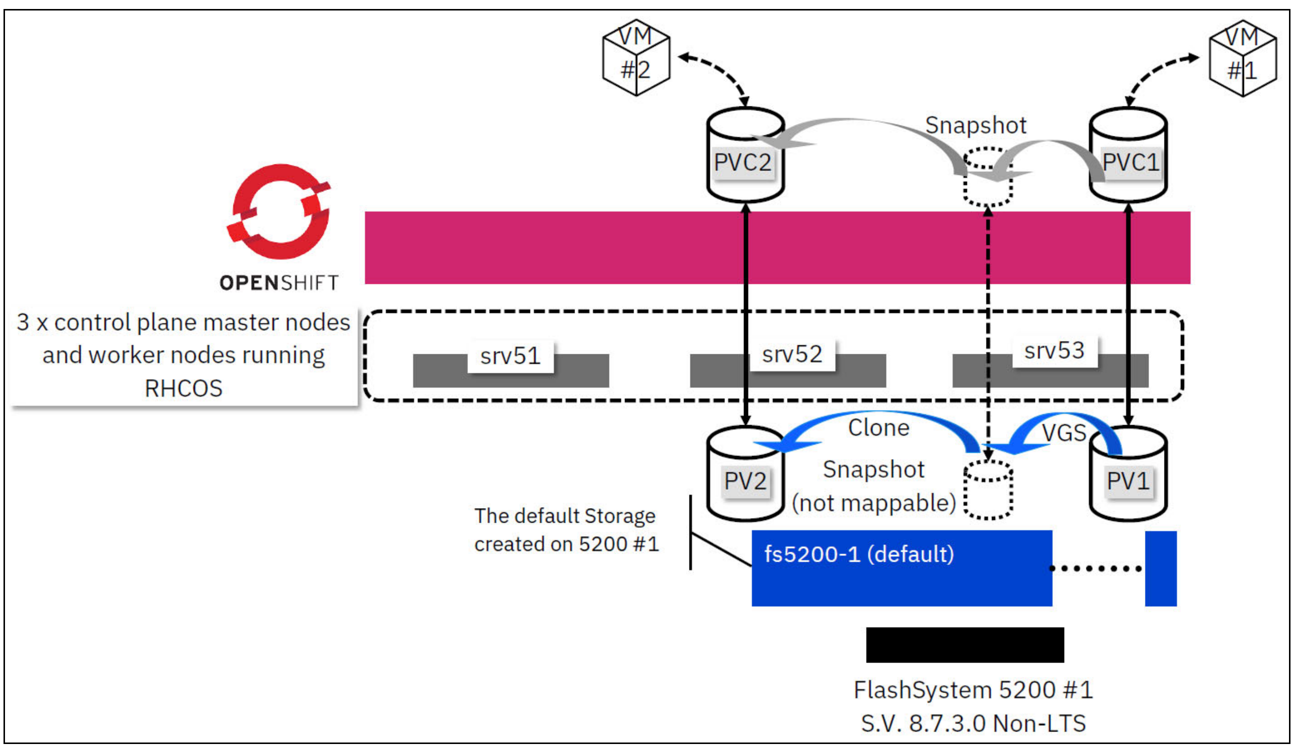

Figure 3-58 Creating a VM from another VM's snapshot

This method of volume snapshotting and cloning, which uses IBM Block Storage CSI, is also used when provisioning a VM from a template. Example 3-11 shows the source volume name for the source VM's volume ID 13. This volume contains the image of a template that is named RHEL 9 server small, which is used to provision the source VM.

Example 3-11 The source volume for the source VM's volume

```
IBM_FlashSystem:FS5200-1:superuser>lsvdisk 13 | grep volume_name
volume_name CSI_pvc-8e7c845b-aa42-4557-af8c-2b873f8cebbd
source_volume_name CSI_pvc-11972854-72fa-48f8-84d0-7b3f8164c395
```

3.4 Day-2 operations: Part 2

During the lifecycle of the Red Hat OpenShift Virtualization environment, system administrators must update or maintain the underlying infrastructure. live migration helps ensure that this process is seamless for VM users.

Compute live migration moves a running VM from one node to another without interrupting its operation. This action enables users to continue accessing the VM without downtime.

You can use live migration for the following tasks:

- Upgrading clusters and firmware
- Performing maintenance on individual servers
- Balancing workloads across servers
- Modifying the underlying infrastructure

Live migration enables system administrators to manage and maintain their Red Hat OpenShift Virtualization environments while helping ensure continuous service availability during updates or maintenance.

3.4.1 Requirements

Live migration requires that the following conditions to be met:

- VM disks must use ReadWriteMany (RWX):

  PVCs so they can be mounted on both the source and destination nodes simultaneously.

  Starting with version 1.12, the IBM Block Storage CSI driver enables RWX support specifically for this Red Hat OpenShift Virtualization live migration use case. However, version 1.12 removes a lock only to allow the volume to be mapped to multiple hosts at the same time.

  The IBM Block Storage CSI driver supports RWX volumes for multi-host attachment to IBM FlashSystem storage. However, it does not manage *concurrent write operations.* While multiple hosts can connect simultaneously, Red Hat OpenShift must enforce strict access controls to prevent parallel write I/O operations on the same volume.

- No pass-through features (such as GPUs) can be present on the VM.
- You must have cluster administrator privileges to perform live migration.
- If you plan to drain a node for maintenance, sufficient resources must be available in the cluster to support workload migration.
- The VM must run on a CPU model that is available on both the source and destination hosts.

By default, a VM running on a host uses the host's latest available CPU features. If a Red Hat OpenShift cluster includes worker nodes with different processor generations, system administrators must help ensure CPU compatibility across nodes.

You can configure VMs to run on a CPU model shared across all nodes, or set the HyperConverged CR to apply a default CPU model globally in the Red Hat OpenShift Virtualization cluster.

Red Hat OpenShift Virtualization sets labels on nodes based on the CPU features it discovers during node scanning.

To list CPU features shared across all worker nodes, you can use the command that is shown in Example 3-12. This command retrieves all CPU feature labels from each node and returns only the labels that are common to all worker nodes in your cluster.

Example 3-12 Listing the CPU models that are shared across all workers nodes

```
oc get nodes -l node-role.kubernetes.io/worker -o json | \
  jq -r '
    def count: [.items[].metadata.name] | length;
    . as $data |
    [.items[].metadata.labels | to_entries[]
    | select(.key | startswith("cpu-model-")) | .key]
    | group_by(.)
    | map(select(length == ($data | count)) | .[0])
    | unique[]'
```

The command output appears as shown in Example 3-13.

Example 3-13 Command output

```
cpu-model-migration.node.kubevirt.io/Haswell-noTSX
cpu-model-migration.node.kubevirt.io/Haswell-noTSX-IBRS
cpu-model-migration.node.kubevirt.io/IvyBridge
cpu-model-migration.node.kubevirt.io/IvyBridge-IBRS
cpu-model-migration.node.kubevirt.io/Nehalem
cpu-model-migration.node.kubevirt.io/Nehalem-IBRS
cpu-model-migration.node.kubevirt.io/Penryn
cpu-model-migration.node.kubevirt.io/SandyBridge
cpu-model-migration.node.kubevirt.io/SandyBridge-IBRS
cpu-model-migration.node.kubevirt.io/Westmere
cpu-model-migration.node.kubevirt.io/Westmere-IBRS
```

This example shows that you can provision VMs to use Westmere CPU features during start if those features are available on all worker nodes in the cluster.

Defining this CPU model in the VM's YAML specification enables live migration of that VM between nodes, as shown in Example 3-14.

Example 3-14 VirtualMachine YAML definition to enforce the Westmere CPU model

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
[…]
spec:
  template:
    spec:
```

```
      architecture: amd64
      domain:
        cpu:
          cores: 4
          model: Westmere
[…]
```

Tip: You can use extra labels on each node to identify which CPU features to apply across your cluster for migration.

Virt-handler pods create these labels as part of Red Hat OpenShift Virtualization. Table 3-1 explains the node CPU labels.

Table 3-1 Explanation of node CPU labels

Node label prefix	Purpose
`cpu-model.node.kubevirt.io/`	Lists CPU models that the node supports natively.
`cpu-model-migration.node.kubevirt.io/`	Indicates which CPU models are compatible for live migration between nodes.
`host-model-cpu.node.kubevirt.io/`	Shows the primary CPU model of the physical host.

Instead of configuring the CPU model for each VM individually, you can define it globally in the HyperConverged CR in Red Hat OpenShift Virtualization. This setting applies to the specified CPU model to all VMs that you start in the Red Hat OpenShift Virtualization cluster.

To enable the CPU model globally, you add `spec.defaultCPUModel` in the HyperConverged CR.

For example, if you want to use the "Westmere" CPU Model, you can run the following command:

```
oc patch hyperconverged kubevirt-hyperconverged -n Red Hat OpenShift-cnv
--type=merge -p '{"spec":{"defaultCPUModel":"Westmere"}}'
```

Note: If you specify a CPU Model in a VM YAML file, it takes precedence over the default CPU model that is defined at the cluster level.

For more information about requirements and configuration steps for live migration, see the Red Hat OpenShift documentation.

3.4.2 Initiating a compute live migration

There are two different ways to start a VM Compute live migration:

- From the Red Hat OpenShift Virtualization web console

  As a cluster administrator, select **Virtualization** → **VirtualMachines**.

  If you are using Red Hat OpenShift 4.18 or later, click the three dots at the end of the VM line and select **Migration** → **Compute** (Figure 3-59 on page 69).

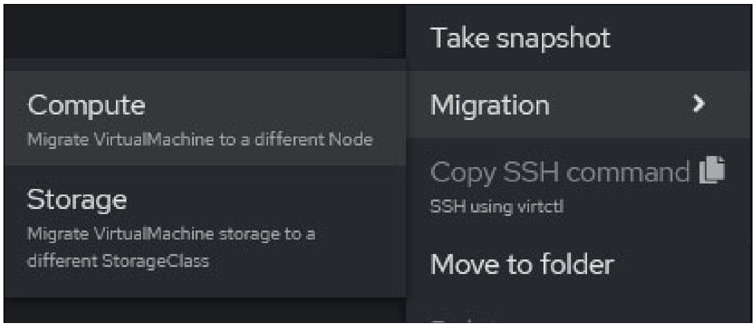

Figure 3-59 Initiating a Compute live migration from the Red Hat OpenShift GUI

If you are using Red Hat OpenShift 4.17 or earlier, click the three dots at the end of the VM line and select **Migrate**.

- From the CLI

  As a cluster administrator, create a VirtualMachineInstanceMigration object in the cluster to start the migration, as shown in Example 3-15.

Example 3-15 Creating a VirtualMachineInstanceMigration object in the cluster to start a migration

```
apiVersion: kubevirt.io/v1
kind: VirtualMachineInstanceMigration
metadata:
  name: <migration_name>
  namespace: <vm_namespace>
spec:
  vmiName: <vm_name>
```

When you apply the YAML to the cluster, it initiates the migration of the VM.

You can use the **`virtctl`** command to start a migration. It creates a VirtualMachineInstanceMigration object similar to Example 3-15.

```
oc project <vm_namespace>
virtctl migrate <vm_name>
```

Note: You can download the **`virtctl`** CLI tool from your Red Hat OpenShift cluster web console by clicking the **?** icon in the upper right and selecting **Command Line Tools**.

3.4.3 Understanding the live migration mechanism

To preserve the VM's compute context during a compute live migration, Red Hat OpenShift Virtualization orchestrates changes across pods, storage attachments, and network configurations, and manage live state transfers. This process remains consistent even when multiple PVCs are attached to the VM. A detailed breakdown of the live migration process follows (Figure 3-60).

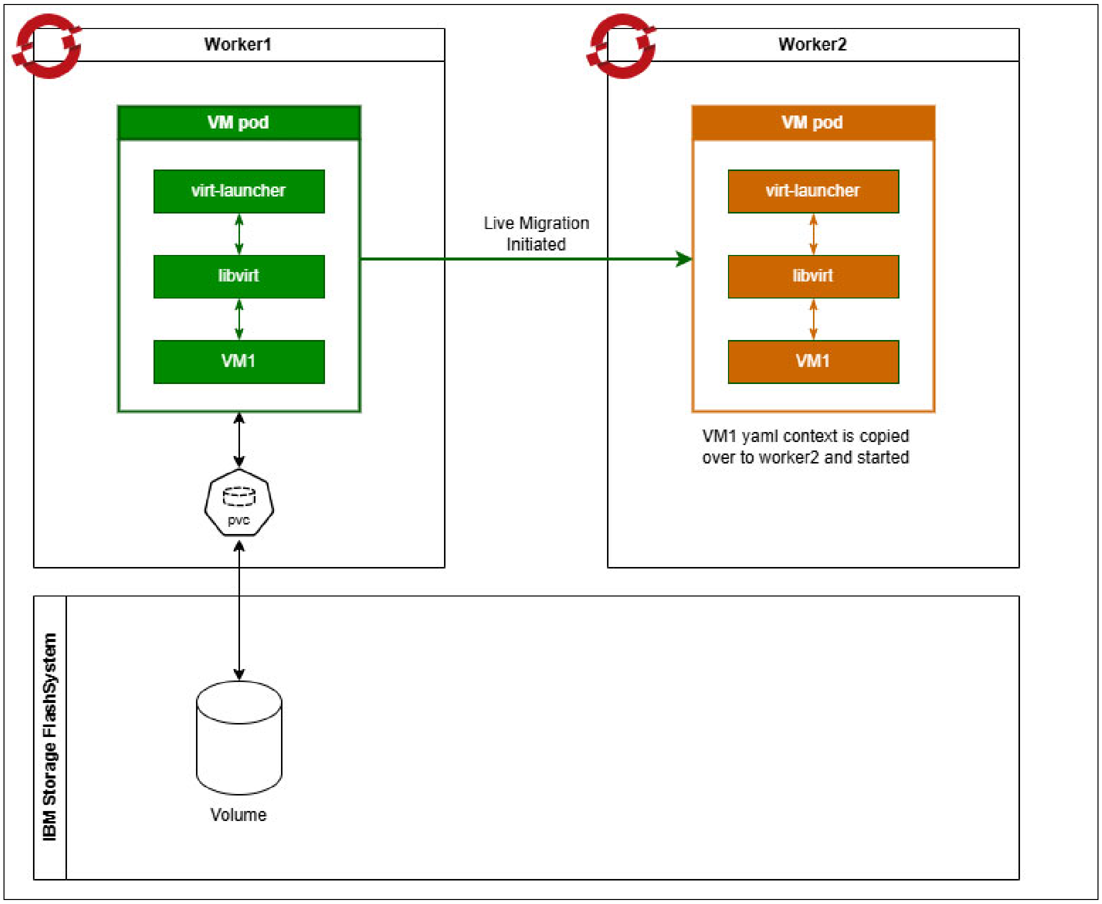

Figure 3-60 Live migration initial state

On the source node (worker1), the virt-launcher pod of the VM continues to run.

On target node (worker2), a new virt-launcher pod is created for the VM by using the same configuration, as shown in Figure 3-61.

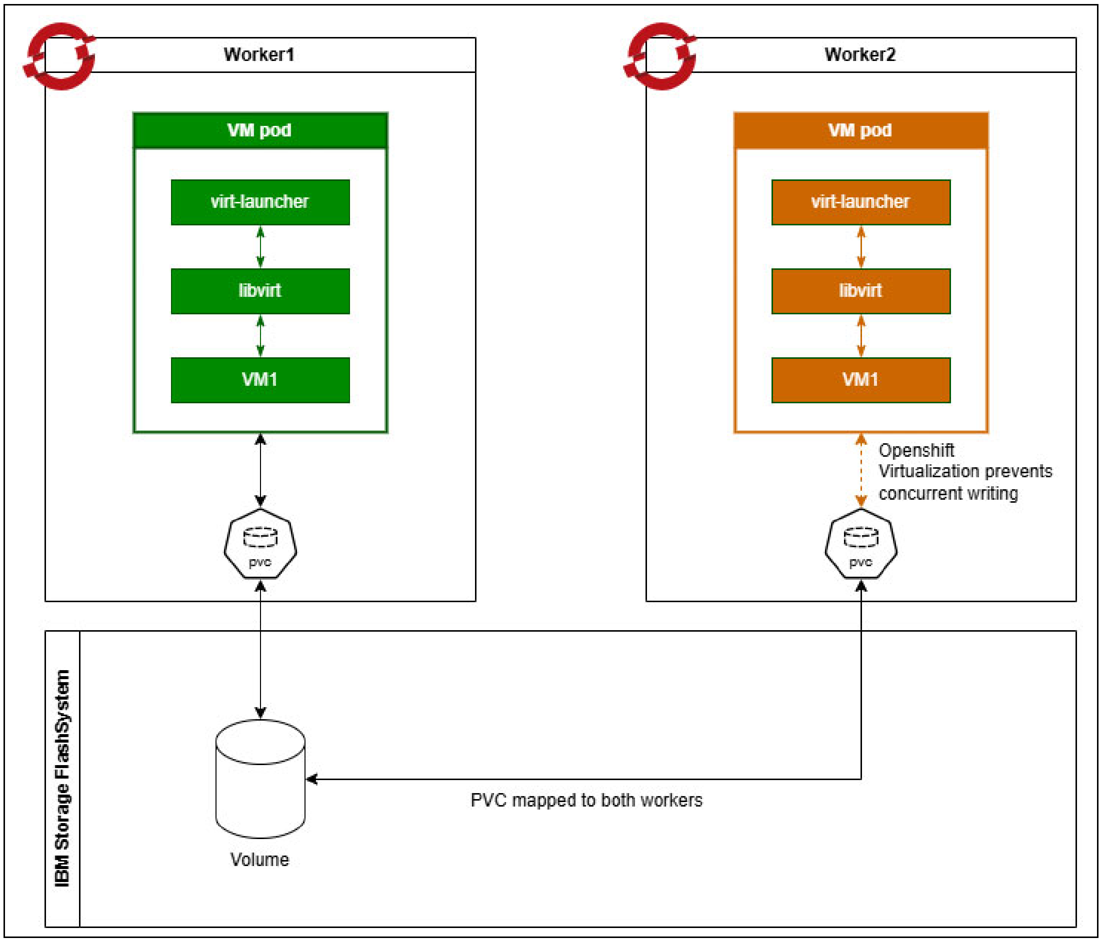

Figure 3-61 Persistent volume mapped to the target host

On the storage back end, the IBM FlashSystem Volume corresponding to the PVC of the VM mapped to worker1 is also mapped to worker2, as shown in Figure 3-62.

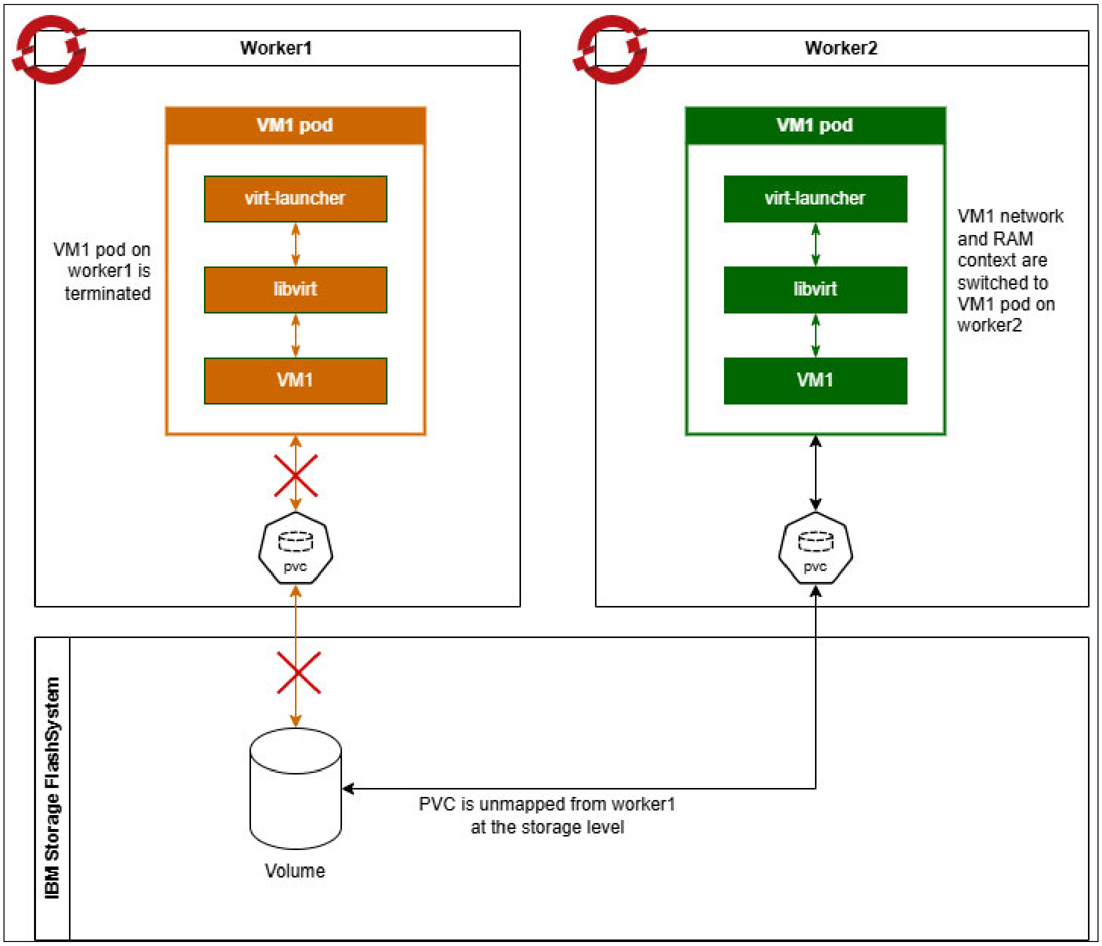

Figure 3-62 Persistent volume unmapped from the source host

An important aspect of live migration is that concurrent writing is not supported. Although IBM Block Storage CSI driver (Version 1.12 and later) allows a PVC to be mounted on multiple nodes. Red Hat OpenShift Virtualization helps ensure that only one node can write to the PVC at any time. The IBM Block Storage CSI driver does not prevent concurrent writes across nodes.

Regarding other VM resources:

- Memory Transfer: The active RAM state is copied over to the new VM.
- Network Redirection: The ARP tables are updated cluster-wide, both existing and new connections are rerouted to the target node (node2).

On completion of the live migration, the source VM pod stops and migrates to the “Completed” state, while the target VM pod initiates with:

- The same network configuration (IP, MAC addresses, and others).
- Identical PCI devices assignments.
- Preserved CPU affinity.

Finally, the IBM FlashSystem volume corresponding to the PVC is unmapped from source node (worker1) on the back-end storage.

For example, when you check the IBM Storage FlashSystem logs, you can see the commands that are run by the IBM Block Storage CSI driver when a VM migration is initiated, as shown in Example 3-16.

Example 3-16 IBM Storage FlashSystem logs

```
4/4/25 10:17:55 AM
svctask mkvdiskhostmap -force -host x3650-758-node2.ocpvirt.mop.ibm -scsi 31
ocpvirt__pvc-8ecd581e-aec7-4902-b029-622f3a3f18f8
4/4/25 10:18:30 AM
svctask rmvdiskhostmap -host x3650-756-node1.ocpvirt.mop.ibm
ocpvirt__pvc-8ecd581e-aec7-4902-b029-622f3a3f18f8
where:
x3650-758-node2.ocpvirt.mop.ibm is the target migration node.
x3650-756-node1.ocpvirt.mop.ibm is the source migration node.
ocpvirt__pvc-8ecd581e-aec7-4902-b029-622f3a3f18f8 is the PVC of the VM.
```

No specific network latency occurs during this process. For a small RHEL 9 server with no workload running, the total live migration process takes approximately 20 seconds and includes the phases that are shown in Example 3-17.

Example 3-17 Phases of the migration

```
oc get vmim <vmim_name> -n <namespace> -o json | jq
'.status.phaseTransitionTimestamps[] | "\(.phase): \(.phaseTransitionTimestamp)"'
    Pending '2025-04-04T13:47:13Z'
    Scheduling '2025-04-04T13:47:14Z'
    PreparingTarget '2025-04-04T13:47:29Z'
    TargetReady '2025-04-04T13:47:29Z'
    Running '2025-04-04T13:47:29Z'
    Succeeded '2025-04-04T13:47:35Z'
```

This information appears in the status of a VirtualMachineInstanceMigration object.

3.5 Day-2 operations: Part 3

To help ensure the stability of the Red Hat OpenShift cluster and IBM FlashSystem, and to avoid resource exhaustion, you need to implement controls on VM storage usage.

To do that, you can act at two different levels:

- At the IBM FlashSystem level, use OBAC and child pools.
- At the Red Hat OpenShift level, use resource quotas.

To keep your Red Hat OpenShift cluster and IBM FlashSystem stable, you should control VM storage usage. You can do this task at two levels:

- IBM FlashSystem:
  - You use OBAC to manage user permissions.
  - You use child pools to manage capacity.
- Red Hat OpenShift

  You implement resource quotas to limit storage usage.

This approach helps you prevent resource exhaustion and maintain operational flexibility.

3.5.1 Limiting storage resources with IBM FlashSystem

By default, storage classes that you create within Red Hat OpenShift are accessible to all users. If you do not apply appropriate safeguards, you can quickly exhaust storage capacity growth on IBM FlashSystem.

To address this problem, you can use OBAC to limit the scope of resources available to Red Hat OpenShift users.

With this feature, you can create administrative groups that have privileges on:

- Pre-defined child pools.
- Subset of hosts and host clusters.
- Predefined FlashCopy consistency groups.

> **Note:** You can also create your own consistency groups.

This section shows you how to restrict the IBM Block Storage CSI driver privileges to prevent adverse impacts on other production systems that are hosted by the Storage Virtualize.

To set up OBAC, you typically:

- Create one or more child pools.
- Create an ownership group.
- Create a user group associated with the ownership group.
- Add new or existing users to the user group.

After you set up OBAC, you configure storage classes and map them to the allocated pools.

Creating child pools

Child pools are subsets of a pool with fully allocated physical capacity. Ensure that their capacity is smaller than the free capacity available to the parent pool.

> **Note:** You can create child pools within data reduction pools. The usable capacity that you allocate to a child pool comes from the entire usable capacity of the parent data reduction pool, effectively without a predefined limit.

Within your Red Hat OpenShift environment, there are several compelling reasons to define multiple child pools instead of a single one:

- You create dedicated child pools based on the service-level agreement (SLA) defined for each project, service, or customer.
- You create dedicated pools per project to meet compliance requirements.

You can create a child pool through the IBM FlashSystem GUI by completing the following steps:

1. After you connect to the GUI as an administrator, select **Pools** → **Pools**.
2. On the selected pool, click **Actions** (the three-dot menu) and select **Create Child Pool**, as shown in Figure 3-63 on page 75.

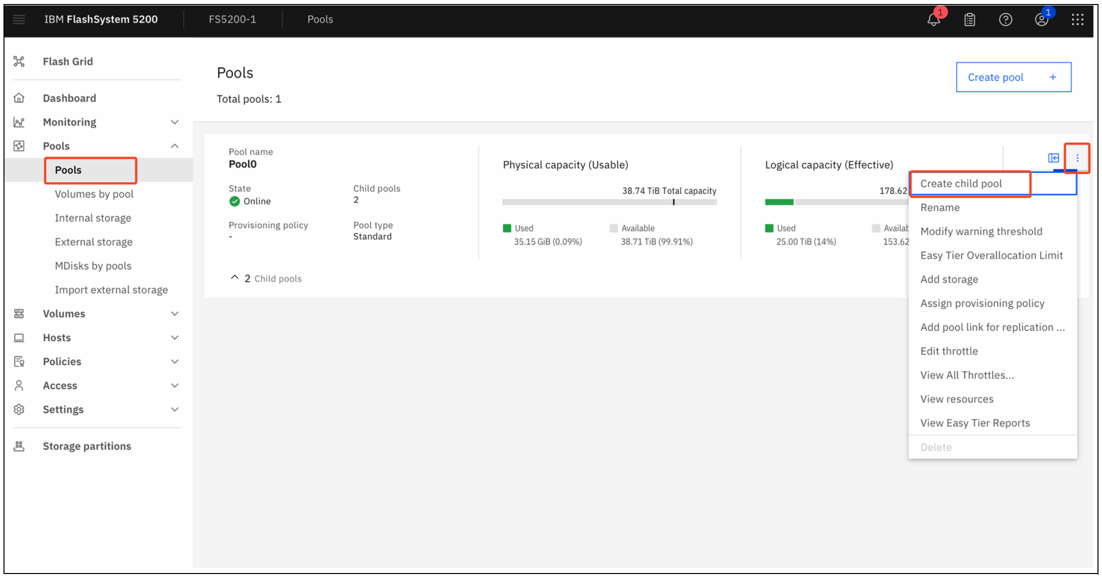

Figure 3-63 Create Child Pool menu

3. Select **Child pool**, and then click **Next**, as shown in Figure 3-64.

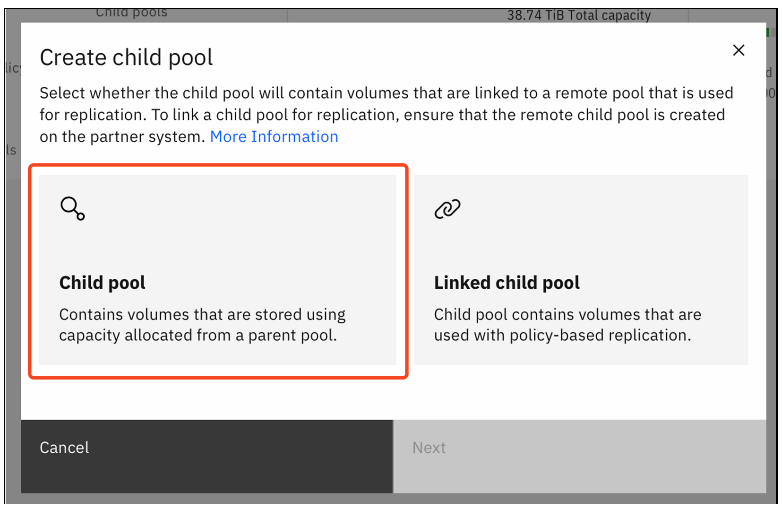

Figure 3-64 Child pool type selection

4. In the next window (Figure 3-65), complete the following steps:
   a. Enter a name for the child pool.
   b. Clear the **Share capacity with parent pool** checkbox to enable the child pool capacity settings.

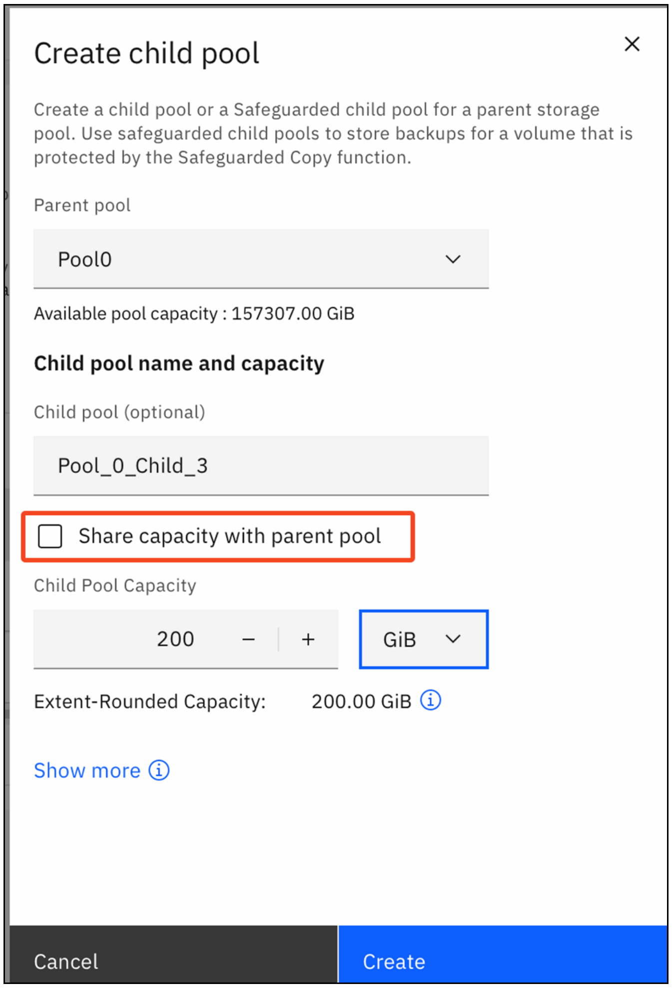

Figure 3-65 Create child pool window

   c. Click **Create**.
5. Expand the parent pool to view its child pools and their usage, as shown in Figure 3-66 on page 77.

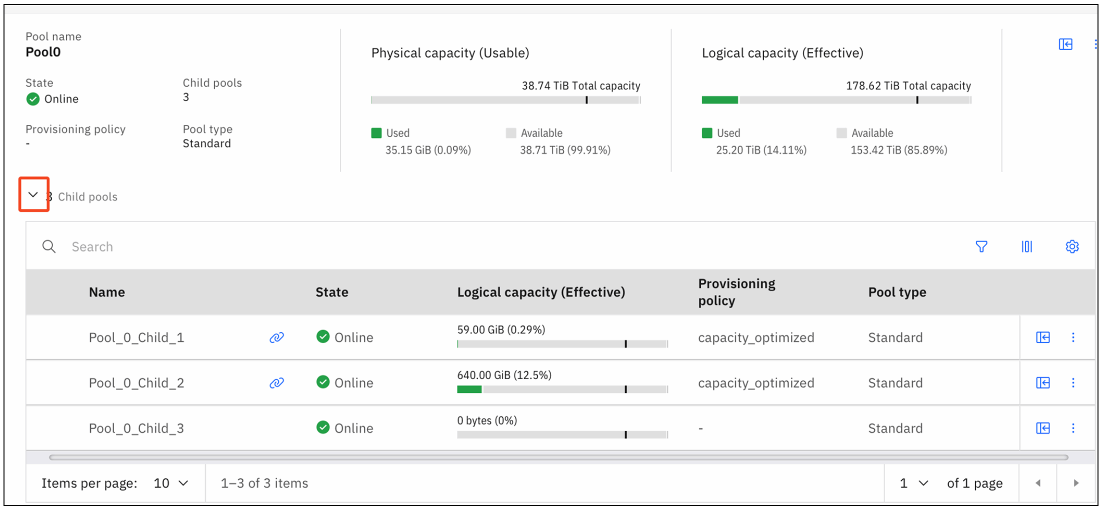

Figure 3-66 Reviewing the created child pools

Defining throttling for the child pools

You can use IBM FlashSystem throttling capabilities to define storage classes with varying performance levels that meet diverse requirements.

According to IBM Documentation:

“Throttles are mechanisms that you use to control the amount of resources consumed when processing I/Os on supported objects. You can apply throttles to hosts, host clusters, volumes, copy offload operations, and storage pools. If you define a throttle limit, the system either processes the I/O or delays the processing of the I/O to free resources for more critical I/O operations.”

You can use throttling to limit IOPS, bandwidth, or both.

To define throttling for the child pools, complete the following steps:

1. In the IBM FlashSystem GUI, select **Pools** → **Pools**.
2. From the **Actions** menu of the selected child pool, click **Edit throttle**, as shown in Figure 3-67.

Figure 3-67 Edit throttle menu

3. In the window that opens, configure the throttle settings, and then click **Create** to apply them, as shown in Figure 3-68.

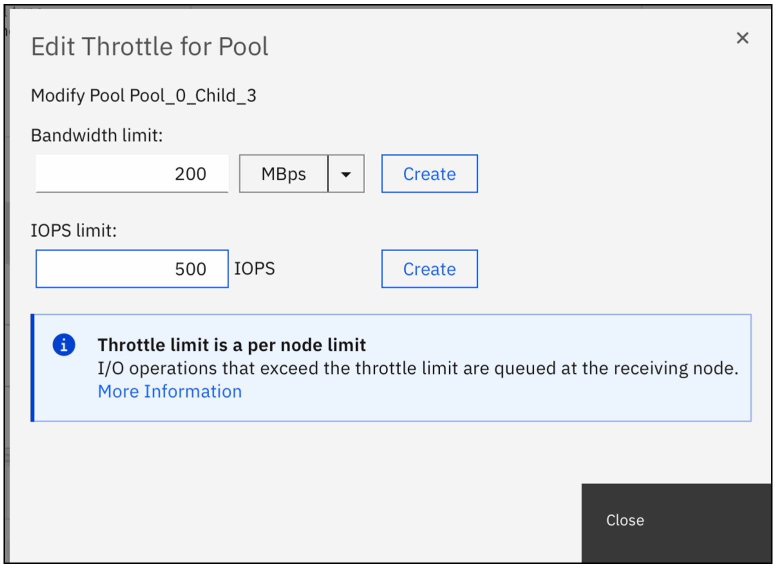

Figure 3-68 Edit Throttle for Pool window

Creating an ownership group

An ownership group represents a specific subset of users and objects within the system. You can create ownership groups to impose extra restrictions on access to specific resources within the group.

> **Important:** You must have Security Administrator privileges to configure and manage ownership groups.

Ownership groups limit access to only the objects that are explicitly assigned to them. Each object can belong to only one ownership group, and you can view and manage the objects that are associated with the group you belong to.

Within Red Hat OpenShift, you establish an ownership group specifically for your Red Hat OpenShift Container Platform cluster. This group manages the child pools that you created, along with the Red Hat OpenShift Container Platform worker nodes.

The IBM Block Storage CSI driver enables dynamic mapping and unmapping of provisioned volumes between Storage Virtualize and Red Hat OpenShift Container Platform worker nodes, as needed.

Complete the following steps:

1. In the IBM FlashSystem GUI, select **Access** → **Ownership groups**, as shown in Figure 3-69 on page 79.

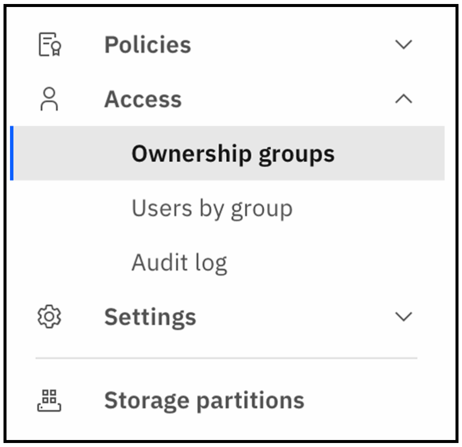

Figure 3-69 Ownerships groups menu

2. Enter a name for the ownership group, then click **Create ownership group**, as shown in Figure 3-70.

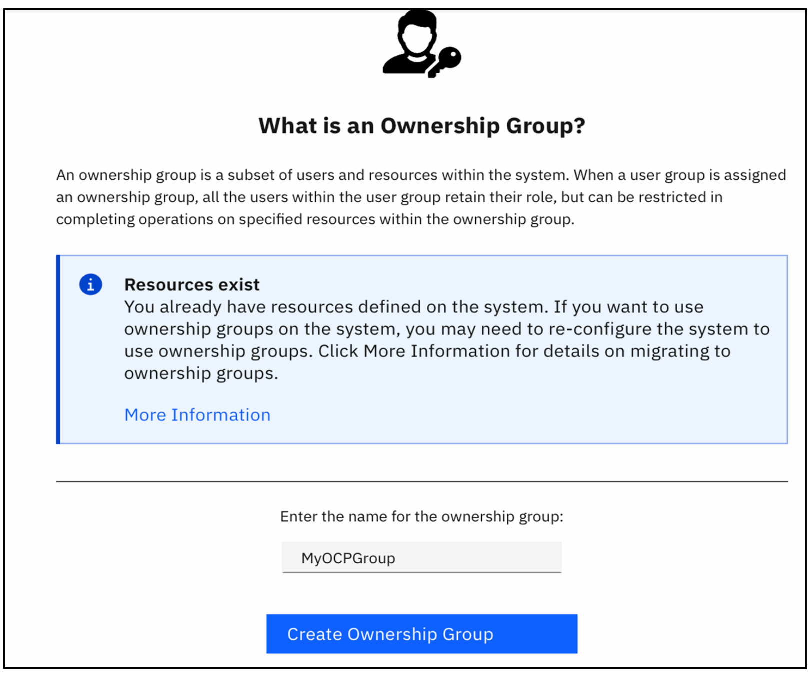

Figure 3-70 Create ownership group window

3. In the next window, which lists resources by ownership group, add the child pools that you created earlier.
4. In the left column, select your newly created group, and then click **Assign Child Pool**.

5. Choose the correct child pool and click **Assign**, as shown in Figure 3-71.

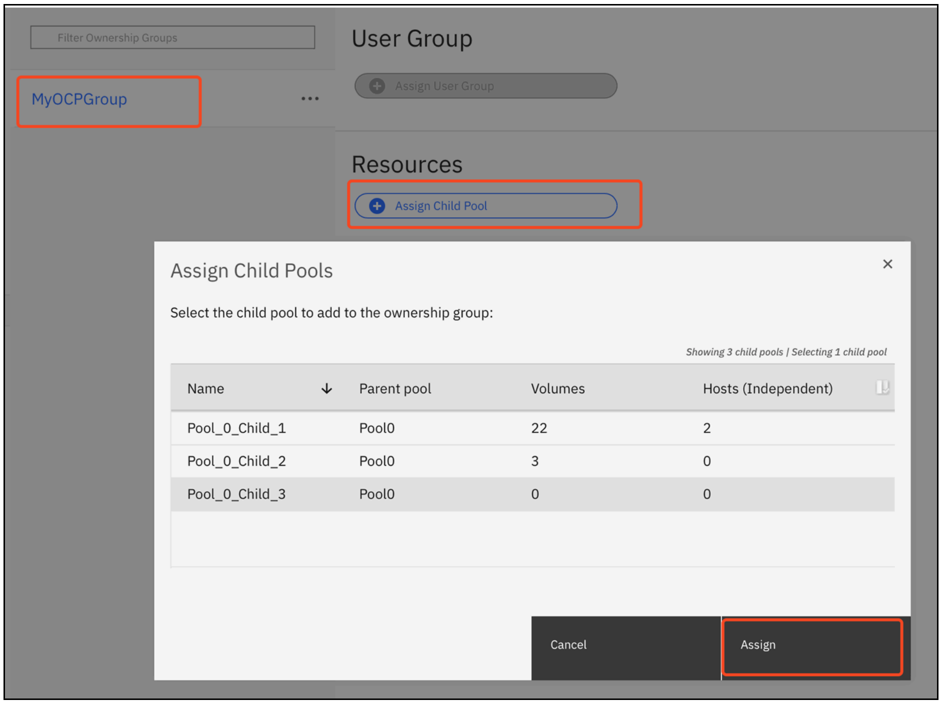

Figure 3-71 Assigning a child pool to an ownership group

6. After assigning the child pool, you can add other objects to the ownership group, such as hosts. For example, select **Independent Hosts**, and then click **Assign**.
7. In the new window, select the hosts that you want to add to the group, and then click **Assign**, as shown in Figure 3-72 on page 81.

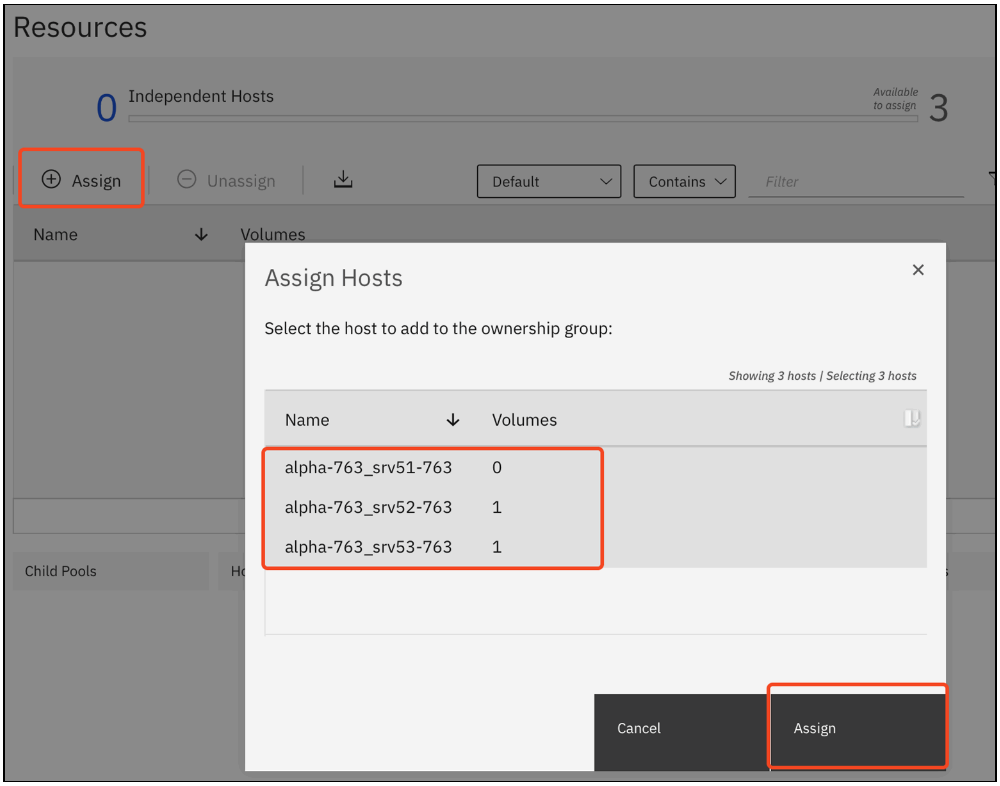

Figure 3-72 Assigning hosts to an ownership group

You can add as many objects as necessary to support your specific use case.

Creating a user group

In IBM FlashSystem software, user roles are determined by group assignments.

Predefined global groups, such as SecurityAdmin and Administrator, provide varying levels of privileges and access to all objects in the system.

Also, you can create custom groups with specific roles that are tied to particular ownership groups. Users assigned to these custom groups inherit the designated role within the corresponding ownership group.

Notes:

- A user can belong to only one user group at a time.
- You cannot assign the Security Administrator role to custom user groups.
- To create a user group, you log in as a Security Administrator on IBM Storage Virtualize.

In this section, you create a user group with administrator privileges for the ownership group by completing the following steps:

1. From the IBM FlashSystem GUI, select **Access** → **Users by group**, as shown in Figure 3-73.

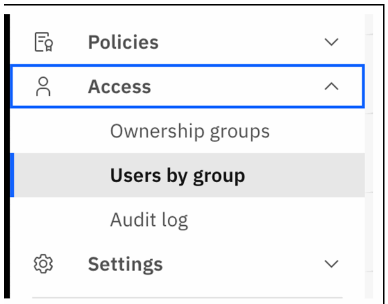

Figure 3-73 User by group menu

2. At the bottom of the left column, click **Create User Group**.
3. Provide a name, select the previously created ownership group, and set **Administrator** as the role. Click **Create**, as shown in Figure 3-74.

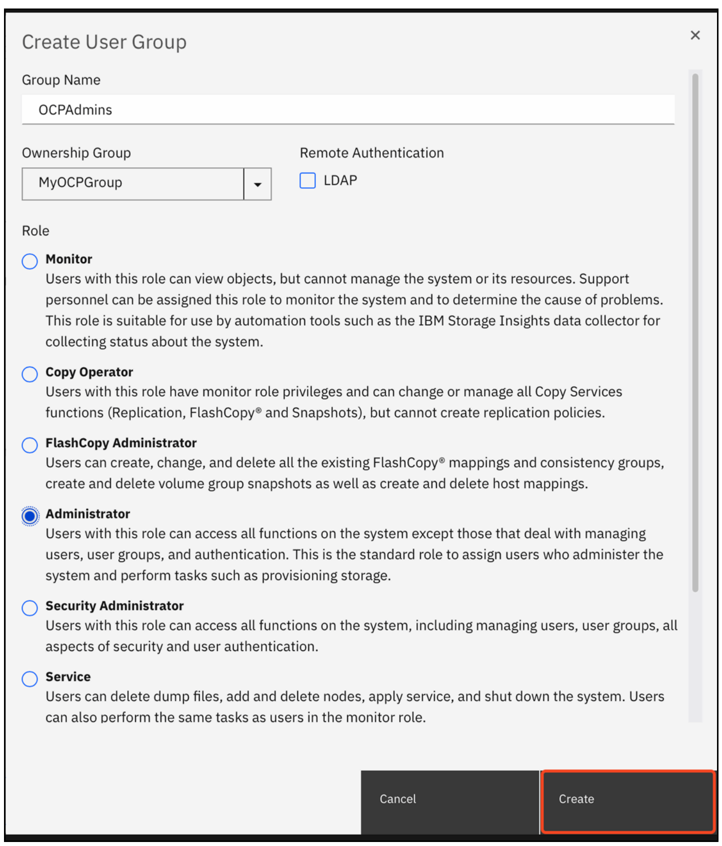

Figure 3-74 Create User Group window

In this example, all users in the OCPAdmins group act as administrators of the ownership group MyOCPGroup.

Include a user from this group to use the IBM Block Storage CSI driver.

Adding users to the group

In this Red Hat OpenShift Container Platform environment, you must use a user account that belongs to the user group that was configured in “Creating child pools” on page 74. This account is required by the IBM Block Storage CSI driver for storage provisioning and usage.

This measure provides granular access control for Red Hat OpenShift Container Platform administrators who interact with IBM FlashSystem, reducing the risk of unintended disruptions to other production systems.

In this example, the IBM Block Storage CSI driver uses the user account named `ocpadmin`. This account has Administrator privileges, which is the minimum level that is required for volume creation, mapping, FlashCopy operations, and related tasks.

Note: You must log in as a Security Administrator on IBM Storage Virtualize to create or modify user accounts.

Complete the following tasks:

1. Select **Access** → **Users by Groups** and click **All Users** (see Figure 3-75).

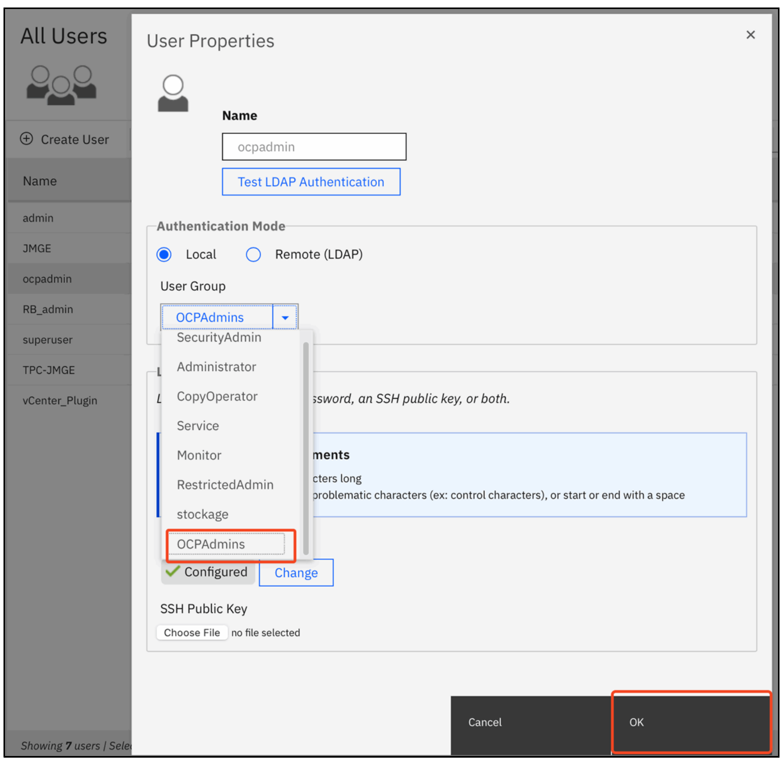

Figure 3-75 Assigning a user to an ownership group

2. Find the user account that is used by the IBM Block Storage CSI driver (ocpadmin), right-click the user, and select **Properties**.
3. In the Properties window, change the user's group from **Administrator** to **OCPAdmin**.
4. Click **OK** to save the changes.

Creating StorageClasses on Red Hat OpenShift Container Platform

After you modify IBM Block Storage CSI driver user privileges to restrict access to specific child pools, you must create corresponding StorageClasses in the Red Hat OpenShift cluster to reflect those changes.

> **Note:** StorageClasses simplify storage management in Kubernetes by defining storage policies and enabling users to request storage dynamically without manual provisioning.

For this demonstration, you create two child pools: Pool_0_Child_2 and Pool_0_Child_3. You then define matching StorageClasses on the Red Hat OpenShift Container Platform cluster to reflect these pools.

Complete the following steps:

1. Log in to your Red Hat OpenShift Container Platform cluster with sufficient privileges to create StorageClasses (for example, cluster-admin).
2. Define YAML files that describe the StorageClasses, helping ensure that they use the IBM Block Storage CSI driver and target the appropriate child pool, as shown in Example 3-18.

Example 3-18 YAML files

```
$ cat sc1.yml
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: fs5200-pool-0-child-2
  namespace: ibm-block-csi
provisioner: block.csi.ibm.com
parameters:
  pool: Pool_0_Child_2
  csi.storage.k8s.io/provisioner-secret-name: fs5200-secret
  csi.storage.k8s.io/provisioner-secret-namespace: ibm-block-csi
  csi.storage.k8s.io/controller-publish-secret-name: fs5200-secret
  csi.storage.k8s.io/controller-publish-secret-namespace: ibm-block-csi
  csi.storage.k8s.io/fstype: ext4
$
$ cat sc2.yml
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: fs5200-pool-0-child-3
  namespace: ibm-block-csi
provisioner: block.csi.ibm.com
parameters:
  pool: Pool_0_Child_3
  SpaceEfficiency: thin
  csi.storage.k8s.io/provisioner-secret-name: fs5200-secret
  csi.storage.k8s.io/provisioner-secret-namespace: ibm-block-csi
  csi.storage.k8s.io/controller-publish-secret-name: fs5200-secret
  csi.storage.k8s.io/controller-publish-secret-namespace: ibm-block-csi
  csi.storage.k8s.io/fstype: ext4
$
```

Each YAML file corresponds to a specific child pool. The fs5200-pool-0-child-3 StorageClass includes the parameter **`SpaceEfficiency: thin`**, which configures volumes that are created with this class as thin-provisioned.

For more information about creating StorageClasses with the IBM Block Storage CSI driver, see the IBM Block Storage CSI driver documentation.

3.5.2 Limiting storage resources with Red Hat OpenShift Container Platform

Previously, you established controls and limitations for the entire Red Hat OpenShift cluster.

In this section, you configure rules to govern and restrict storage consumption by Red Hat OpenShift Container Platform users.

In Red Hat OpenShift Container Platform, Resource Quotas and Cluster Resource Quotas help you manage resource allocation, such as CPU, memory, and the number of pods. Resource Quotas enforce these controls at the project level, while Cluster Resource Quotas extend control across multiple projects. These quotas also regulate storage capacity and the number of volumes, enable you to control the following storage elements:

- `requests.storage`: The cumulative storage requested by all PVCs, regardless of their state, cannot exceed this limit.
- `persistentvolumeclaims`: The total number of PVCs within the project is limited to this value.
- `<storage-class-name>.storageclass.storage.k8s.io/requests.storage`: The total storage that is requested by all PVCs with the specified StorageClass cannot exceed this limit.
- `<storage-class-name>.storageclass.storage.k8s.io/persistentvolumeclaims`: The total number of PVCs with the specified StorageClass is limited to this value.

Global values (`requests.storage` and `persistentvolumeclaims`) take precedence over values that are defined for specific StorageClasses.

> **Important:** To create update or delete a resource quota, you must have the cluster-admin role. It is advisable to define a global quota for each project. Without a global quota, any unspecified StorageClass remains uncontrolled, which can allow unlimited storage provisioning.

Examples

Consider a Red Hat OpenShift Container Platform cluster with four StorageClasses: fs5200-sc1, fs5200-sc2, fs5200-sc3, and fs5200-sc4.

Case 1

Define the Resource Quota rules, as shown in Example 3-19.

Example 3-19 Resource Quota rules

```
persistentvolumeclaims: "10"
requests.storage: "7Gi"
fs5200-sc1.storageclass.storage.k8s.io/requests.storage: "4Gi"
fs5200-sc1.storageclass.storage.k8s.io/persistentvolumeclaims: "5"
fs5200-sc2.storageclass.storage.k8s.io/requests.storage: "5Gi"
fs5200-sc3.storageclass.storage.k8s.io/requests.storage: "0"
fs5200-sc3.storageclass.storage.k8s.io/persistentvolumeclaims: "0"
```

This configuration implies:

- You cannot request storage that uses the fs5200-sc3 StorageClass (explicitly set to 0).
- You cannot create PVCs that uses the fs5200-sc3 StorageClass (explicitly set to 0).
- You can request up to 4Gi of storage that uses the fs5200-sc1 StorageClass.
- You can request up to 5Gi of storage that uses the fs5200-sc2 StorageClass.
- The total storage that is requested across all PVCs cannot exceed 7Gi.
- You can create up to five PVCs that use the fs5200-sc1 StorageClass.
- There is no specific limit on the number of PVCs that you can create when you use the fs5200-sc2 StorageClass.
- The total number of PVCs cannot exceed 10.
- The fs5200-sc4 StorageClass does not have specific limits and is governed by the global quotas.

Case 2

Example 3-20 defines the Resource Quota rules.

Example 3-20 Resource Quota rules

```
fs5200-sc1.storageclass.storage.k8s.io/requests.storage:
"0"fs5200-sc1.storageclass.storage.k8s.io/persistentvolumeclaims:"0"
```

This configuration implies:

- You cannot use the fs5200-sc1 StorageClass to request storage because its quota is explicitly set to 0.
- You cannot use the fs5200-sc1 StorageClass to create PVCs because its quota is explicitly set to 0.
- You can use all other StorageClasses without restriction because they do not have global or specific limits.

> **Important:** If two quotas govern the same resource within a project, the more restrictive quota takes precedence. If both a Resource Quota and a Cluster Resource Quota govern the same resource, the system enforces the lower quota.

3.5.3 Managing storage resources per projects

Projects create and enforce Resource Quotas rules. As mentioned 3.5.2, "Limiting storage resources with Red Hat OpenShift Container Platform" on page 85, you must have the cluster-admin role to create, update, or delete a Resource Quota.

Using storage quotas

Define storage quotas for your project in a YAML file, such as `storage-consumption.yaml`, as shown in Example 3-21.

Example 3-21 Defining storage quotas for your project

```
apiVersion: v1
kind: ResourceQuota
metadata:
  name: storage-consumption
```

```
spec:
  hard:
    persistentvolumeclaims: "10"
    requests.storage: "7Gi"
    fs5200-sc1.storageclass.storage.k8s.io/requests.storage: "3Gi"
    fs5200-sc1.storageclass.storage.k8s.io/persistentvolumeclaims: "5"
    fs5200-sc2.storageclass.storage.k8s.io/requests.storage: "5Gi"
    fs5200-sc2.storageclass.storage.k8s.io/persistentvolumeclaims: "10"
Deploy the quota to your Red Hat OpenShift project (my-project):
$ oc apply -f storage-consumption.yaml -n my-project
```

In this example, the StorageClasses fs5200-sc1 and fs5200-sc2 are governed by both global and class-specific quotas, whereas all other StorageClasses are restricted only by global quotas.

When you create a volume and a quota is reached, the system returns an error, as shown in Example 3-22.

Example 3-22 Error message

```
$ oc apply -f pvc-mypvc-fs5200-sc1-3gb.yml
Error from server (Forbidden): error when creating "pvc-mypvc-fs5200-sc1-3gb.yml":
persistentvolumeclaims "mypvc3" is forbidden: exceeded quota: storage-consumption,
requested: fs5200-sc1.storageclass.storage.k8s.io/requests.storage=3Gi,
used: fs5200-sc1.storageclass.storage.k8s.io/requests.storage=2Gi,
limited: fs5200-sc1.storageclass.storage.k8s.io/requests.storage=3Gi
```

As your project evolves, monitoring storage consumption becomes essential. You can track usage by inspecting the project overview or the ResourceQuota object.

To review storage consumption for your project, use the following command:

```
$ oc describe projects my-project
```

The output includes general project details such as creation time, labels, and annotations. It also includes a Quota section that shows storage usage against configured limits, as shown in Example 3-23.

Example 3-23 The output of the oc command describes my-project

```
Quota:
    Name:                                                storage-consumption
    Resource                                                   Used    Hard
   --------                                                   ----    ----
    persistentvolumeclaims                                      3       10
    requests.storage                                            3Gi     7Gi
    fs5200-sc1.storageclass.st.../persistentvolumeclaims        2       5
    fs5200-sc1.storageclass.st.../requests.storage              2Gi     3Gi
    fs5200-sc2.storageclass.st.../persistentvolumeclaims        1       10
    fs5200-sc2.storageclass.st.../requests.storage              1Gi     5Gi
```

To inspect storage consumption that directly uses ResourceQuota, run the following command:

```
$ oc describe resourcequotas storage-consumption -n my-project
```

The output matches the format that is shown in Example 3-23.

3.5.4 Avoiding storage overallocation on new projects

By default, new projects do not include quotas, which enables users to allocate unlimited storage. Any logged-in user can create a new project. As a result, any authenticated user can create a project without resource limits and consume storage freely.

To avoid this issue, consider the following solutions:

- Remove the default ability for users to create projects. Instead, projects are created by:
  - A Red Hat OpenShift administrator, who also sets the quota for the requester.
  - Users who are authorized to create projects.
- Define a default quota that applies to all new projects that use project templates. When a user creates a new project, the system automatically creates a Resource Quota. A Red Hat OpenShift Container Platform administrator can then adjust the quota based on the agreed-upon SLA.

This section focuses on the second option, which is more practical, and less restrictive.

Creating a custom Project Template that enforces resource quotas

By default, new projects do not include resource quotas, which allow users to provision unlimited storage. To address this limitation, implement a *Project Template* that automatically enforces quotas for all new projects across the cluster.

Project Templates provide a centralized way to enforce default resource quotas across all projects. When configured:

- The template applies to all users and projects.
- Each new project automatically includes a ResourceQuota object.
- You can restrict specific storage classes.

To create the custom Project Template, complete the following steps:

1. Generate the base template by running the following command:

   ```
   $ oc adm create-bootstrap-project-template -o yaml > template.yaml
   ```

2. Edit `template.YAML` to include quota restrictions, as shown in Example 3-24.

Example 3-24 Editing template.YAML

```
apiVersion: template.Red Hat OpenShift.io/v1
kind: Template
metadata:
  name: project-request
objects:
- apiVersion: project.Red Hat OpenShift.io/v1
  kind: Project
  metadata:
    annotations:
      Red Hat OpenShift.io/description: ${PROJECT_DESCRIPTION}
      Red Hat OpenShift.io/display-name: ${PROJECT_DISPLAYNAME}
      Red Hat OpenShift.io/requester: ${PROJECT_REQUESTING_USER}
    name: ${PROJECT_NAME}
  spec: {}
  status: {}
- apiVersion: rbac.authorization.k8s.io/v1
  kind: RoleBinding
  metadata:
```

```
    name: admin
    namespace: ${PROJECT_NAME}
  roleRef:
    apiGroup: rbac.authorization.k8s.io
    kind: ClusterRole
    name: admin
  subjects:
  - apiGroup: rbac.authorization.k8s.io
    kind: User
    name: ${PROJECT_ADMIN_USER}
- apiVersion: v1
  kind: ResourceQuota
  metadata:
    name: storage-consumption-${PROJECT_NAME}
  spec:
    hard:
      persistentvolumeclaims: "5"
      requests.storage: 10Gi
      fs5200-sc2.storageclass.storage.k8s.io/persistentvolumeclaims: "5"
      fs5200-sc2.storageclass.storage.k8s.io/requests.storage: 5Gi
      fs5200-sc1.storageclass.storage.k8s.io/persistentvolumeclaims: "0"
      fs5200-sc1.storageclass.storage.k8s.io/requests.storage: "0"
parameters:
- name: PROJECT_NAME
- name: PROJECT_DISPLAYNAME
- name: PROJECT_DESCRIPTION
- name: PROJECT_ADMIN_USER
- name: PROJECT_REQUESTING_USER
```

In this example, the project template is named project-request, and it adds a ResourceQuota to each new project, which:

- Allows project users to create five volumes with a combined maximum of 10Gi across any storage class.
- Limits usage to 5Gi in the fs5200-sc2 storage class.
- Prevents use of the fs5200-sc1 storage class because its quota is explicitly set to 0.

3. Deploy the template to your Red Hat OpenShift cluster by running the following command:

```
$ oc apply -f template.yaml -n Red Hat OpenShift-config
```

To use this template for all new projects, the cluster must be configured by editing the `project.config.Red Hat OpenShift.io/cluster` object, which can be done through either interface:

- Use the CLI and run the following command:

```
oc edit project.config.Red Hat OpenShift.io/cluster
```

- Use the Web Console and perform the following steps:
  a. Select **Administration** → **Cluster Settings**.
  b. Click the **Global Configuration** tab.
  c. Locate the **Project** resource and click **Edit**, as shown in Figure 3-76.

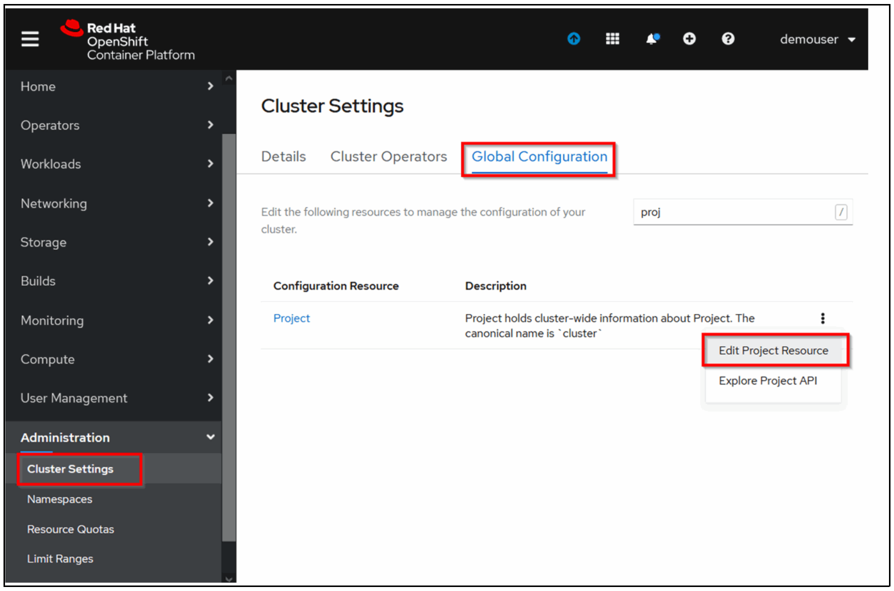

Figure 3-76 Editing the Project Template by using the Red Hat OpenShift GUI

In the configuration, add the `projectRequestTemplate` section under `spec` and specify the template name (default: `project-request`), as shown in Example 3-25. You can use a different name if the template is customized.

Example 3-25 Adding the projectRequestTemplate section

```
spec:
  projectRequestTemplate:
    name: project-request
```

After validation, the system automatically generates and enforces a Resource Quota, as shown in Example 3-26.

Example 3-26 New projects automatically generate and enforce a resource quota

```
$ oc new-project the-new-project
Now using project "the-new-project" on server "https://api.cluster311.ocpstorage.icc:6443".
```

You can add applications to the project by using the **`new-app`** command. For example, run the following command to build a sample Ruby application:

```
$ oc new-app rails-postgresql-example
```

Alternatively, you can use **`kubectl`** to deploy a simple Kubernetes application, as shown in Example 3-27 on page 91.

Example 3-27 Deploying a simple Kubernetes application

```
kubectl create deployment hello-node
--image=registry.k8s.io/e2e-test-images/agnhost:2.43 -- /agnhost serve-hostname

$ oc get resourcequota
NAME                                  CREATED AT
storage-consumption-the-new-project   2020-08-24T12:20:02Z
$ oc describe resourcequota storge-consumption-the-new-project
Name:                                 storage-consumption-the-new-project
Namesspace:                                                the-new-project
Resource                                                      Used    Hard
--------                                                      ----    ----
persistentvolumeclaims                                           0       5
requests.storage                                                 0    10Gi
fs5200-sc1.storageclass.storage.k8s.io/persistentvolumeclaims   0       0
fs5200-sc1.storageclass.storage.k8s.io/requests.storage         0       0
fs5200-sc2.storageclass.storage.k8s.io/persistentvolumeclaims   0       5
fs5200-sc2.storageclass.storage.k8s.io/requests.storage         0     5Gi
```

Modifying an existing the resource quota

The custom project template automatically enforces default quotas for all new projects. However, users might need more storage capacity or access to different storage classes. Administrators can modify existing quotas through these methods:

- Use the CLI and run the following command:

  ```
  oc edit resourcequota quota-to-modify -n project-name
  ```

- Use the Web Console and perform the following steps:

  a. Select **Administration** → **Resource Quotas**.
  b. Select the appropriate quota and click **Edit Resource Quota**.

When you edit the `spec.hard` section of the ResourceQuota object, the changes take effect immediately.

In Example 3-28, we allowed the requester to use 5Gi of fs5200-sc1 storage with a maximum of two PVCs.

Example 3-28 Editing the spec.hard section of the ResourceQuota object

```
apiVersion: v1
kind: ResourceQuota
metadata:
  name: storage-consumption-the-new-project
  …
spec:
  hard:
    persistentvolumeclaims: "5"
    requests.storage: 10Gi
    fs5200-sc1.storageclass.storage.k8s.io/persistentvolumeclaims: "2"
    fs5200-sc1.storageclass.storage.k8s.io/requests.storage: 5Gi
    fs5200-sc2.storageclass.storage.k8s.io/persistentvolumeclaims: "5"
    fs5200-sc2.storageclass.storage.k8s.io/requests.storage: 5Gi
…
```

3.5.5 Managing storage resources per projects

Cluster resource quotas offer an alternative method for controlling storage consumption across multiple projects. They enforce shared resource limits by aggregating usage across all projects that match specified criteria.

You apply ClusterResourceQuota to projects by using label selectors and annotation selectors, or both.

ClusterResourceQuota works alongside project-specific quotas to provide layered governance, as shown in Example 3-29 and Example 3-30.

Example 3-29 Example 1: Restricting storage usage for all projects that are created by my-user-name

```
apiVersion: v1
kind: ClusterResourceQuota
metadata:
  name: storage-consumption-for-my-user-name
spec:
  quota:
    hard:
      persistentvolumeclaims: "10"
      requests.storage: "7Gi"
      fs5200-sc1.storageclass.storage.k8s.io/requests.storage: "3Gi"
      fs5200-sc1.storageclass.storage.k8s.io/persistentvolumeclaims: "5"
      fs5200-sc2.storageclass.storage.k8s.io/requests.storage: "3Gi"
      fs5200-sc2.storageclass.storage.k8s.io/persistentvolumeclaims: "5"
  selector:
    annotations:
      Red Hat OpenShift.io/requester: "my-user-name"
```

Example 3-30 Example 2: Restricting storage usage for all development projects

```
apiVersion: v1
kind: ClusterResourceQuota
metadata:
  name: storage-consumption-for-dev-projects
spec:
  quota:
    hard:
      persistentvolumeclaims: "10"
      requests.storage: "7Gi"
      fs5200-sc1.storageclass.storage.k8s.io/requests.storage: "3Gi"
      fs5200-sc1.storageclass.storage.k8s.io/persistentvolumeclaims: "5"
      fs5200-sc2.storageclass.storage.k8s.io/requests.storage: "3Gi"
      fs5200-sc2.storageclass.storage.k8s.io/persistentvolumeclaims: "5"
  selector:
    labels:
      matchLabels:
        environment: "dev"
```

When you create a cluster resource quota, it takes effect immediately, just like a resource quota.

To view the resource consumption of a cluster resource quota:

- If you are a cluster administrator user, run the following command:

```
oc describe clusterresourcequota storage-consumption-for-my-user-name
```

- If you are not a cluster administrator, run the following command:

```
oc describe AppliedClusterResourceQuota
```

In both cases, you see output similar to Example 3-31.

Example 3-31 The output of the commands

```
$ oc describe AppliedClusterResourceQuota
Name:           storage-consumption-for-my-user-name
Created:        10 minutes ago
Labels:         <none>
Annotations:
kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"quota.Red Hat
OpenShift.io/v1","kind":"ClusterResourceQuota","metadata":{"annotations":{},"name"
:"storage-consumption-for-test"},"spec":{"quota":{"hard":{"persistentvolumeclaims"
:"10","requests.storage":"7Gi","v7k-gold.storageclass.storage.k8s.io/persistentvol
umeclaims":"5","v7k-gold.storageclass.storage.k8s.io/requests.storage":"3Gi","v7k-
silver.storageclass.storage.k8s.io/persistentvolumeclaims":"5","v7k-silver.storage
class.storage.k8s.io/requests.storage":"3Gi"}},"selector":{"annotations":{"Red Hat
OpenShift.io/requester":"test"}}}}
Namespace Selector: ["test-quota2" "the-new-project" "my-project-name"
"test-project2"]
Label Selector:
AnnotationSelector: map[Red Hat OpenShift.io/requester:test]
Resource                                                      Used    Hard
--------                                                      ----    ----
persistentvolumeclaims                                        2       10
requests.storage                                              2Gi     7Gi
v7k-gold.storageclass.stora....io/persistentvolumeclaims      1       5
v7k-gold.storageclass.stora....io/requests.storage            1Gi     3Gi
v7k-silver.storageclass.stora....io/persistentvolumeclaims    1       5
v7k-silver.storageclass.stora....io/requests.storage          1Gi     3Gi
```

3.6 Advanced topics

This section covers advanced topics that you might occasionally need during Day-1 and Day-2 operations, but are not in every use case.

3.6.1 Custom resources that interact with storage

In Red Hat OpenShift, every object or service that extends the Kubernetes API is defined through a CR.

For persistent storage with VMs, the most important CRs are VirtualMachine, VirtualMachineInstance, DataVolume, PVC, and PV. Other CRs are VirtualMachineSnapshot and VirtualMachineSnapshotContents, along with related storage CRs, such as VolumeSnapshot and VolumeSnapshotContents, which are created when you take VM snapshots.

Figure 3-77 shows the relationship between CRs in Red Hat OpenShift Virtualization relevant for storage. When you create a VM from a template, the resulting VirtualMachine CR owns a DataVolume for the OS. This DataVolume is provisioned by using the OS image source PVC defined in the selected template. The PVC associated with the DataVolume uses the strategy that is defined by the StorageProfile of the default StorageClass for virtualization.

The OS image source PVCs are regularly updated through the associated DataVolume and the DataImportCron it references. The DataImportCron specifies the update schedule.

When you trigger a VM snapshot, Red Hat OpenShift creates a VirtualMachineSnapshot CR, along with a VirtualMachineSnapshotContents CR. The latter references the VolumeSnapshot requested from the underlying CSI provider, which in turn references the actual snapshot contents in the VolumeSnapshotContents CR.

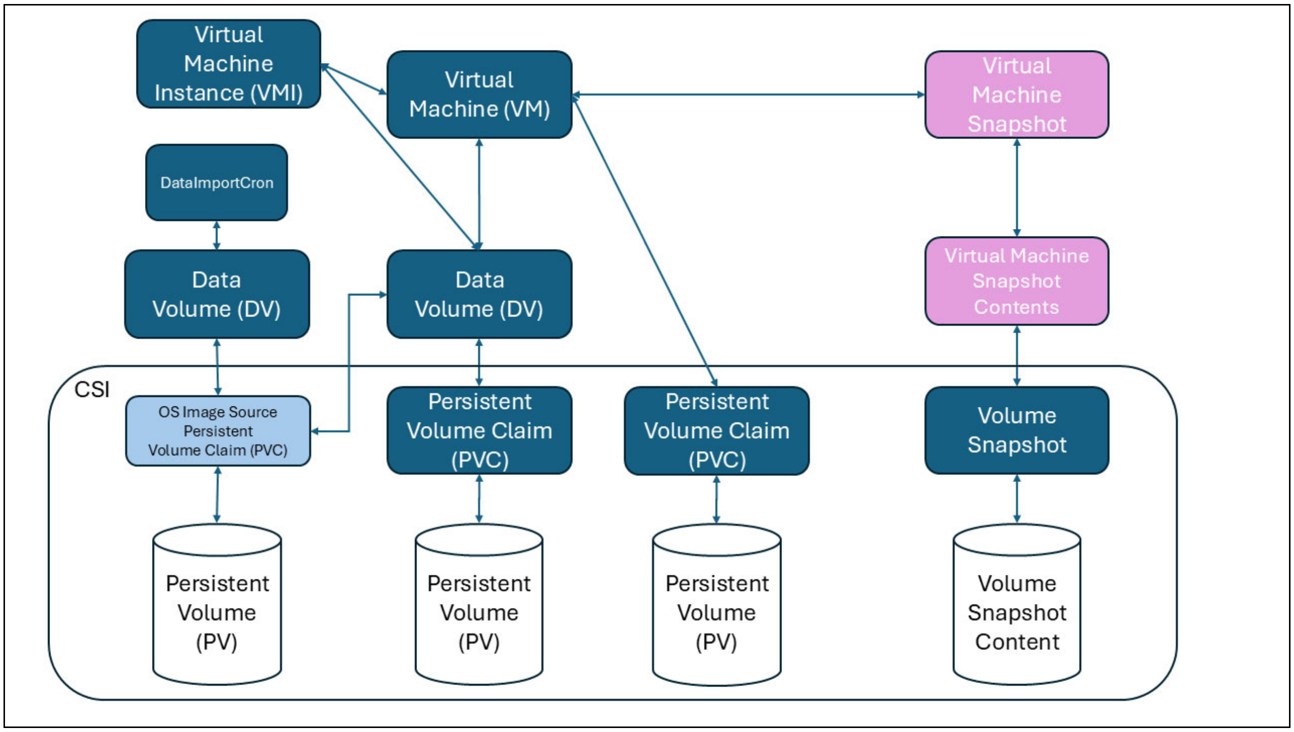

Figure 3-77 Relationship between custom resources in Red Hat OpenShift Virtualization relevant for storage

VirtualMachine

The VirtualMachine CR is the top-level resource that represents a VM, including its compute, memory, and persistent storage settings. A VM can reference DataVolumes for representation of persistent storage, but it can also directly reference PVCs.

For more information, see v1.VirturalMachine.

VirtualMachineInstance

The VirtualMachineInstance CR represents a running VirtualMachine. The VirtualMachineInstance CR operates under its corresponding VirtualMachine CR.

For more information, see v1.VirtualMachineInstance.

DataVolume

The DataVolume CR connects between the VM and PVClaim. It also maintains a reference to the source OS image used to instantiate the PVC, along with the status of that operation.

For more information, see v1beta1.DataVolume.

StorageProfile

You use the StorageProfile CR to define best practice parameters for creating a PVC for a DataVolume.

For more information, see this GitHub repository.

VirtualMachineSnapshot

The VirtualMachineSnapshot CR represents the snapshot creation operation for a VM.

For more information, see v1beta1.VirtualMachineSnapshot.

VirtualMachineSnapshotContent

The VirtualMachineSnapshotContent CR contains the details of the VM snapshot including the captured VM configuration and the reference to the associated persistent storage. Storage-based snapshots use the CSI driver and its snapshot function.

For more information, see v1beta1.VirtualMachineSnapshotContent.

3.6.2 Red Hat OpenShift Container Platform 4.18 VM storage live migration

Red Hat OpenShift Virtualization 4.18 introduces select migration as a *technology preview*, enabling users to migrate running or stopped VMs between different StorageClasses.

With this feature, you can modify the underlying storage infrastructure, such as migrating to new storage systems or rebalancing workloads across different storage systems, without disrupting VM operations.

The following explanation and tests were done with select migration in Red Hat OpenShift 4.18 as of April 2025.

To enable the technology preview select migration feature in Red Hat OpenShift 4.18, the HyperConverged CR needs to be modified to enable VolumeMigration, as shown in Example 3-32.

Example 3-32 Enabling the technology preview select migration feature

```
oc annotate --overwrite -n Red Hat OpenShift-cnv hco kubevirt-hyperconverged
kubevirt.kubevirt.io/jsonpatch='[ {"op": "add", "path":
"/spec/configuration/developerConfiguration/featureGates/-", "value":
"VolumesUpdateStrategy"}, {"op": "add", "path":
"/spec/configuration/developerConfiguration/featureGates/-", "value":
"VolumeMigration"} ]'
```

Note: Red Hat does not support enabling features in KubeVirt by using **`oc annotate`**. When you enable features this way, your cluster displays a warning message about UnsupportedHCOModification.

To start a storage migration from the Red Hat OpenShift UI, click a VM and select **Actions** → **Migration** → **Storage**, as shown in Figure 3-78.

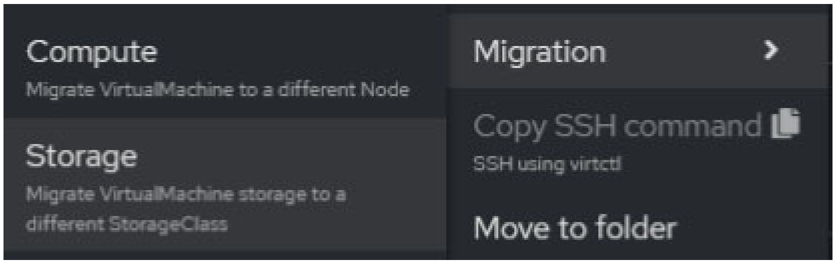

Figure 3-78 Initiating a storage live migration from Red Hat OpenShift GUI

When you start a storage live migration, the system also initiates a compute live migration.

Storage live migration requirements

The target and source StorageClasses object must allow the same claimPropertySets in their StorageProfile.

A StorageProfile is a CR that defines optimized storage configurations for VM disks based on the capabilities of the associated StorageClass. It helps ensure that VMs use appropriate storage settings, such as access modes and volume mode combinations, for features like storage live migration.

Red Hat OpenShift Virtualization automatically creates StorageProfiles for each StorageClass in the cluster.

As an example, see the StorageProfile for the IBM Block Storage CSI driver StorageClass in Example 3-33.

Example 3-33 StorageProfile for the IBM Block Storage CSI driver StorageClass

```
oc get storageprofile/svc-sc -o yaml
apiVersion: cdi.kubevirt.io/v1beta1
kind: StorageProfile
metadata:
  labels:
    app: containerized-data-importer
    app.kubernetes.io/component: storage
    app.kubernetes.io/managed-by: cdi-controller
    app.kubernetes.io/part-of: hyperconverged-cluster
    app.kubernetes.io/version: 4.18.1
    cdi.kubevirt.io: ""
  name: svc-sc
  ownerReferences:
  - apiVersion: cdi.kubevirt.io/v1beta1
    blockOwnerDeletion: true
    controller: true
    kind: CDI
    name: cdi-kubevirt-hyperconverged
spec: {}
status:
  claimPropertySets:
  - accessModes:
    - ReadWriteMany
    volumeMode: Block
```

```
  - accessModes:
    - ReadWriteOnce
    volumeMode: Block
  - accessModes:
    - ReadWriteOnce
    volumeMode: Filesystem
  - accessModes:
    - ReadWriteMany
    volumeMode: Filesystem
  cloneStrategy: snapshot
  dataImportCronSourceFormat: pvc
  provisioner: block.csi.ibm.com
  snapshotClass: svc-volumesnapshotclass
  storageClass: svc-sc
```

You can view the available claimPropertySets in the StorageProfile, such as ReadWrtieMany/Filesystem or ReadWriteMany/Block.

To migrate a VM that uses ReadWrtieMany/Filesystem disks from one StorageClass to another, you must help ensure that the claimPropertySet is available in the target StorageClass.

If the target StorageProfile does not include the required claimPropertySet, you must add it manually.

For example, to add ReadWrtieMany/Filesystem to a target StorageProfile, you can run the following command:

```
oc edit storageprofile/<target-sc-profile-name> -o yaml
```

Then add a claimPropertySets specification, as shown in Example 3-34.

Example 3-34 Adding a claimPropertySets spec

```
spec:
  claimPropertySets:
    - accessModes:
        - ReadWriteMany
      volumeMode: Filesystem
```

Storage live migration process as of technology preview April 2025

This section describes the storage live migration process as of technology preview April 2025:

1. When migrating begins, KubeVirt (through virt-handler pod) creates a new PVC in the target StorageClass and copies the content from the source PVC.
2. KubeVirt creates a DataVolume based on the new PVC.
3. A new virt-launcher pod starts for the VM *on a new node* with an updated YAML that points to the new DataVolume. When this pod is running, the source virt-launcher pod for the VM migrates to the Completed state.

You can perform the migration on a running VM. During the migration, you might observe a brief latency when pods or PVCs are switched, as shown in Example 3-35.

Example 3-35 Small latency during migration when pods/PVCs are switched

```
64 bytes from 10.3.59.201: icmp_seq=5569 ttl=61 time=27.4 ms
64 bytes from 10.3.59.201: icmp_seq=5570 ttl=61 time=27.5 ms
64 bytes from 10.3.59.201: icmp_seq=5578 ttl=61 time=13083 ms
64 bytes from 10.3.59.201: icmp_seq=5579 ttl=61 time=12044 ms
64 bytes from 10.3.59.201: icmp_seq=5580 ttl=61 time=11004 ms
64 bytes from 10.3.59.201: icmp_seq=5581 ttl=61 time=9964 ms
64 bytes from 10.3.59.201: icmp_seq=5582 ttl=61 time=8924 ms
64 bytes from 10.3.59.201: icmp_seq=5583 ttl=61 time=7884 ms
64 bytes from 10.3.59.201: icmp_seq=5584 ttl=61 time=6834 ms
64 bytes from 10.3.59.201: icmp_seq=5585 ttl=61 time=5804 ms
64 bytes from 10.3.59.201: icmp_seq=5587 ttl=61 time=3724 ms
64 bytes from 10.3.59.201: icmp_seq=5588 ttl=61 time=2684 ms
64 bytes from 10.3.59.201: icmp_seq=5586 ttl=61 time=4764 ms
64 bytes from 10.3.59.201: icmp_seq=5589 ttl=61 time=1644 ms
64 bytes from 10.3.59.201: icmp_seq=5590 ttl=61 time=604 ms
64 bytes from 10.3.59.201: icmp_seq=5591 ttl=61 time=27.9 ms
64 bytes from 10.3.59.201: icmp_seq=5592 ttl=61 time=27.7 ms
```

Abbreviations and acronyms

AIOps	Artificial Intelligence for IT Operations
CBT	Change Block Tracking
CNSA	Container Native Storage Access
CR	custom resource
CRD	Custom Resource Definition
CSI	Container Storage Interface
FC	Fibre Channel
HADR	high availability and disaster recovery
HA	high availability
IBM	International Business Machines Corporation
MVP	Minimum Viable Product
OBAC	Object-Based Access Control
OS	operating system
PoCs	Proofs of Concept
PV	persistent volume
PVC	persistent volume claim
RHCOS	Red Hat Enterprise Linux CoreOS
RHEL	Red Hat Enterprise Linux
RPO	Recovery Point Objective
RTD	Ransomware Threat Detection
RTO	Recovery Time Objective
RWO	ReadWriteOnce
RWX	ReadWriteMany
SDS	software-defined storage
SLA	service-level agreement
SSO	single sign-on
VGS	Volume Group Snapshot
VM	virtual machine
WAD	Workload Anomaly Detection

© Copyright IBM Corp. 2025.

Related publications

The publications that are listed in this section are considered suitable for a more detailed description of the topics that are covered in this paper.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topics in this document. Some publications that are referenced in this list might be available in softcopy only.

- *Implementation Guide for IBM Storage FlashSystem and IBM SAN Volume*, SG24-8542
- *Intelligent Storage Management with Artificial Intelligence for IT Operations by using IBM Storage Insights*, REDP-5755
- *Storage Multi-tenancy for Red Hat OpenShift Container Platform with IBM Storage*, REDP-5638
- *Using the IBM Block Storage CSI Driver in a Red Hat OpenShift Environment*, REDP-5613

You can search for, view, download, or order these documents and other Redbooks, Redpapers, web docs, drafts, and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- IBM Storage FlashSystem

  https://www.ibm.com/flashsystem
- kuvervirt.io API definitions

  https://kubevirt.io/cdi-api-reference/main/definitions.html#_top_level_api_objects
- Red Hat OpenShift Container Platform

  https://www.redhat.com/en/technologies/cloud-computing/openshift/container-platform
- Red Hat OpenShift Virtualization

  https://www.redhat.com/en/technologies/cloud-computing/openshift/virtualization

© Copyright IBM Corp. 2025.

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

REDP-5753-00

ISBN 0738462233

Printed in U.S.A.

Get connected

ibm.com/redbooks