GUIDE BOOK

RED HAT CLOUD FOUNDATIONS

The Complete Guide to Building Enterprise Private Clouds with OpenShift, OpenStack, and Ceph

BEKROUNDJO AKOLEY ARISTIDE

Red Hat Cloud Foundations: The Complete Guide to Building Enterprise Private Clouds with OpenShift, OpenStack, and Ceph

Published by BEKROUNDJO AKOLEY ARISTIDE

CASABLANCA, Morocco

ISBN: 9798312350395

Printed in the United States of America through Amazon Kindle Direct Publishing.

First Edition, 2025

Cover design and interior formatting by BEKROUNDJO AKOLEY ARISTIDE

Georgia font used throughout.

For more information, updates, or to contribute to the open-source resources accompany-ing this book, visit https://github.com/BAristide/OCPRHOSO2025.

CONTENTS

PREFACE **1**
What You'll Learn 1
Who This Book Is For 2
How to Use This Book 2

ABOUT THE AUTHOR **4**

PART I: GETTING STARTED WITH RED HAT CLOUD **6**

CHAPTER 1: INTRODUCTION TO RED HAT CLOUD FOUNDATIONS **7**
Why On-Premise Cloud Matters 7
The Three Pillars: OpenShift, RHOSO, and Ceph 9
Your Lab Journey: What to Expect 12
What You'll Build: The End Result 14
Summary 16

CHAPTER 2: SETTING UP YOUR LAB ENVIRONMENT **17**
Infrastructure Architecture Overview 18
Key Infrastructure Components 18
Hardware and Software Prerequisites 22
Network Architecture 24

Bridging Virtual and Physical Networks 26
Physical network Topology 26
Lab Environment Network Topology 29
Mapping Virtual to Physical 30

Setting Up the KVM Hypervisor on lab1 31
Installing KVM on Ubuntu 22.04 31
Configuring Network vSwitches 32
Creating Virtual Machines for OpenShift Nodes 35
Implementing Virtual BMC (vBMC) 37

Preparing the Provisioner Node 39
Provisioner Node Overview 39
Customizing the RHEL QCOW2 Image 39
Deploying the Provisioner VM on lab1 40
Initial Provisioner Configuration 41
Setting Up the OpenShift Environment: 43

Configuring Essential Services 45
Summary 49

PART II: DEPLOYING THE CORE STACK 50

CHAPTER 3: DEPLOYING OPENSHIFT ON-PREMISE 51
OpenShift Essentials: Containers and Kubernetes 52
Understanding OpenShift Deployment Methods 55
Setting Up the OpenShift Installer 57
Preparing the Installation Configuration 60
Automating Pre-Deployment Tasks 67
Deploying and Monitoring the Cluster 68
Post-Deployment Validation and Configuration 70
Troubleshooting Common Issues 74
Summary 77
Lab Exercises 78
Lab Exercise Solutions 79

CHAPTER 4: INTEGRATING RED HAT OPENSTACK SERVICES (RHOSO) 86
Understanding OpenStack and RHOSO 87
Deploying RHOSO on OpenShift 89
Lab Environment Architecture Overview 89
Installing RHOSO Operators 91
Setting Up the OpenStack Namespace 101
Securing OpenStack Services 102
Preparing OpenShift for OpenStack Networks 104
Creating Network Attachment Definitions 110
Installing and Configuring MetalLB 113
Creating the data plane network 118
Deploying the RHOSO Control Plane 120
Creating the Data Plane with Unprovisioned Nodes 128
Deploying the data plane. 142
Summary 147

CHAPTER 5: ADDING CEPH STORAGE TO THE MIX 148
Why Ceph for Hybrid Clouds? 149
Deploying Ceph Storage on Three Nodes 150
Integrating Ceph with RHOSO 158
Creating RHOSO Pools and Users 158
Transferring Ceph Credentials to OpenShift 160
Integrating Ceph with the RHOSO Control Plane 161
Configuring the RHOSO Data Plane for Ceph 169
Validating RHOSO Integration: Launch a test VM 174
Configuring Ceph for OpenShift with Dynamic RBDs 180
Persistent Storage with Static Ceph RBDs 189
Summary 194
Lab Exercises 195
Lab Exercise Solutions 197

PART III: SECURING AND OPTIMIZING YOUR CLOUD 201

CHAPTER 6: SECURING YOUR CLOUD INFRASTRUCTURE 202
Configuring LDAP Authentication with Red Hat Identity Management 203
Understanding Identity Management in the Enterprise Context 203
Setting Up Red Hat IdM on the Provisioner Node 203
Managing Users and Groups in IdM 206
Configuring OpenShift with IdM LDAP 207

Configuring HTPasswd Authentication 210
Setting Up HTPasswd Users 210
Integrating HTPasswd with OpenShift 210

Updating the Default OpenShift Certificate 212
Understanding Certificate Types in OpenShift 212
Generating a Custom Root CA and Certificate 212
Deploying the Certificate to OpenShift 213
Updating Client Trust 214

Configuring RHOSO Authentication 216
Understanding Authentication in RHOSO 216
Configuring Keystone for LDAP Authentication 216

Implementing Security Best Practices 219
Summary 221

CHAPTER 7: MONITORING AND PERFORMANCE TUNING 222
OpenShift's Integrated Monitoring Stack 223
Monitoring Core Components 224
Leveraging OpenShift Dashboards 228
Implementing Performance Optimization 231
OpenShift Cluster Optimization 231
RHOSO Performance Optimization 233
Ceph Performance Tuning 234

Building High Availability into Your Hybrid Cloud 235
OpenShift High Availability 235
RHOSO High Availability 236
Ceph High Availability 237

Testing High Availability 238
Control Plane Failover Testing 238
RHOSO Service Failover Testing 238
Ceph Resilience Testing 239

Case Study: Diagnosing a Performance Bottleneck 239
Summary 243

PART IV: ADVANCED OPERATIONS AND APPLICATIONS 244

CHAPTER 8: OPERATING RHOSO IN PRODUCTION 245
RHOSO Service Architecture 246
Managing Multi-Tenancy Workloads 248
Using Horizon and Heat for Orchestration 262

Monitoring Your Clusters 266
Summary 269

CHAPTER 9: SCALING YOUR CLOUD INFRASTRUCTURE 270
Why Extend Your Clusters? 271
Extending OpenShift Clusters 272
Extending the RHOSO Cluster with a New Compute Node 278
Case Study: Scaling for a High-Traffic App 288
Summary 290

CHAPTER 10: DEPLOYING REAL-WORLD APPLICATIONS 292
Manual Multi-Tier App Deployment 293
GitOps-Based Automation with OpenShift and Argo CD 299
OpenShift GitOps Foundation 299
Deploying Red Hat OpenShift GitOps Operator 299
Argo CD Implementation 303
Application Repository Structure 307
Argo CD Application Deployment 317
Continuous Synchronization 326
Summary 332

APPENDICES 333
Appendix A: Command Reference 333
Appendix B: Configuration Snippets 335
Appendix C: Lab Environment Setup 337
Appendix D: Glossary 338
Appendix E: Additional Resources 338

PREFACE

Welcome to Red Hat Cloud Foundations: The Complete Guide to Building Enterprise Private Clouds with OpenShift, OpenStack, and Ceph. This book is your guide to building a powerful, on-premise private cloud using Red Hat's industry-leading technologies. Whether you're new to cloud infrastructure or a seasoned architect looking to master hybrid deployments, these pages will walk you through every step—from setting up a lab environment to deploying real-world applications. My goal is to demystify the process, blending practical, hands-on instructions with the foundational knowledge you need to succeed.

What You'll Learn

In this book, you'll gain the skills to design, deploy, and manage a robust on-premise cloud stack powered by Red Hat OpenShift, Red Hat OpenStack Services on OpenShift (RHOSO), and Red Hat Ceph Storage. Here's what awaits you:

Core Deployment: Step-by-step guidance to install OpenShift for container orchestration, RHOSO for virtualized workloads, and Ceph for scalable storage—all within a KVM-based lab you can replicate.

Integration and Optimization: Techniques to seamlessly integrate these technologies, secure your cluster, tune performance, and ensure high availability for production readiness.

Real-World Applications: Hands-on examples, from launching a simple VM to automating a multi-tier application with GitOps, showing how your cloud supports diverse workloads.

Foundational Knowledge: Clear explanations of why these tools matter, how they work together, and their role in modern private cloud strategies.

By the end, you'll not only have a working hybrid cloud but also the confidence to adapt it to your organization's needs—whether for development, testing, or enterprise-grade deployments.

Who This Book Is For

This book is crafted for a wide range of IT professionals eager to harness Red Hat's cloud technologies on-premise:

- **Beginners and System Administrators**: If you're new to OpenShift, OpenStack, or Ceph, or transitioning from traditional infrastructure, this book starts with the basics—setting up a lab, deploying core components, and validating your work with practical demos. No prior expertise is assumed, though familiarity with Linux and virtualization (e.g., KVM) helps.
- **Cloud Architects and DevOps Engineers**: For those with experience in cloud deployments, you'll find advanced topics like cluster scaling, performance tuning, security with Red Hat IdM, and GitOps automation—plus real-world case studies to inspire your next project.
- **Red Hat Enthusiasts**: If you're already invested in Red Hat's ecosystem (e.g., RHEL, OpenShift), this book bridges your knowledge to a fully integrated private cloud stack.

Whether you're building a proof-of-concept lab or architecting a production environment, this book meets you where you are and equips you for what's next.

How to Use This Book

Red Hat Cloud Foundations is organized into four parts, designed to take you from foundational setup to advanced operations in a logical, hands-on progression:

- **Part I**: **Getting Started with Red Hat Cloud** (Chapters 1-2) introduces the why and how of on-premise cloud, guiding you through lab setup with KVM and networking basics—perfect for beginners or those refreshing their setup.
- **Part II**: **Deploying the Core Stack** (Chapters 3-5) dives into deploying OpenShift, RHOSO, and Ceph, with detailed instructions and early demos to build your confidence.
- **Part III**: **Securing and Optimizing Your Cloud** (Chapters 6-7) focuses on production readiness—security, monitoring, and performance—so your stack is robust and reliable.
- **Part IV**: **Advanced Operations and Applications** (Chapters 8-10) explores operating RHOSO, scaling clusters, and deploying real-world apps, culminating in GitOps automation for expert-level control.

Each chapter blends concepts (the "what" and "why") with practical steps (the "how"), supported by CLI commands, YAML snippets, and validation checks. Case studies—like diagnosing a performance issue or scaling for a high-traffic app—bring theory to life. The appendices offer quick references for commands, configurations, and further learning.

- **For Beginners**: Start at Chapter 1 and follow sequentially, experimenting with the labs on a single host (e.g., a well-equipped workstation). Use the glossary (Appendix D) for unfamiliar terms.

- **For Experts**: Jump to specific chapters (e.g., Chapter 5 for Ceph, Chapter 10 for GitOps) or explore case studies to adapt ideas to your environment. The companion GitHub repo (github.com/BAristide/OCPRHOSO2025) provides all scripts.
- **Hands-On Tip**: Set up the lab in Chapter 2 as your sandbox—every subsequent chapter builds on it.

My hope is that this book becomes both a learning companion and a practical reference, empowering you to master Red Hat's cloud technologies on your terms.

ABOUT THE AUTHOR

Personal Journey in Cloud Infrastructure

Bekroundjo Akoley Aristide is a seasoned cloud architect, educator, and the visionary founder of EC INTELLIGENCE, a premier consulting and training firm dedicated to advancing cloud infrastructure, containerization, and cutting-edge IT solutions. With over a decade of hands-on experience deploying, managing, and scaling complex on-premise and hybrid cloud environments, Aristide has established himself as a trusted leader for organizations embarking on digital transformation journeys. His passion for empowering IT professionals and businesses shines through in his work, blending deep technical expertise with a knack for making intricate concepts accessible.

As a certified Red Hat Ready Partner, Aristide has honed his mastery of Red Hat technologies, including OpenShift Container Platform, Red Hat OpenStack Services on OpenShift (RHOSO), and Red Hat Ceph Storage—the core pillars of this book. His proficiency extends to Kubernetes, Proxmox virtualization, and OpenStack, complemented by a practical, problem-solving approach forged through real-world projects. Notable achievements include:

- **Private Cloud Deployments**: Designed and implemented OpenStack- and OpenShift-based private clouds for Orange Guinea, delivering scalable infrastructure tailored to enterprise needs.
- **Cluster Optimization**: Enhanced Proxmox clusters for international clients, boosting performance and reliability in virtualized environments.
- **Storage Solutions**: Architected self-managed Ceph storage clusters integrated with virtualization platforms, ensuring high availability and flexibility for diverse workloads.

Beyond consulting, Aristide is a prolific content creator and educator, sharing his knowledge through multiple channels. His YouTube channel, *EC INTELLIGENCE*, offers in-depth tutorials on Ceph storage, OpenShift, virtualization, Kubernetes, and more, amassing a global following of IT enthusiasts and professionals seeking practical insights. He also develops e-learning courses and writes technical articles, distilling years of experience into actionable guidance. This book, *Red Hat Cloud Foundations: The Complete Guide to Building Enterprise Private Clouds with OpenShift, OpenStack, and Ceph*, is a culmination of that expertise, offering a step-by-step guide to building a robust hybrid cloud using Red Hat's ecosystem.

Born in Côte d'Ivoire and now based in Morocco, Aristide is driven by a mission to elevate Africa's tech landscape. Through EC INTELLIGENCE, he provides localized expertise, reducing dependency on external vendors and fostering innovation among African businesses. His work bridges technical excellence with community impact, empowering the next generation of IT leaders with accessible training and cutting-edge solutions.

Outside of technology, Aristide is an avid photographer, capturing the beauty of Morocco's landscapes, and a lifelong learner, staying ahead of industry trends to deliver the latest advancements to his clients and readers. In this book, he shares his comprehensive understanding of OpenShift, RHOSO, and Ceph, guiding IT professionals to master the tools and techniques needed to create scalable, secure, and efficient on-premise cloud environments.

Contact the Author

Stay connected with Aristide for updates, insights, and collaboration opportunities:

- LinkedIn: https://www.linkedin.com/in/akoley-bekroundjo/
- EC INTELLIGENCE YouTube Channel: https://www.youtube.com/@ecintelligence
- Email: a.bekroundjo@ecintelligence.ma

PART I: GETTING STARTED WITH RED HAT CLOUD

CHAPTER 1: INTRODUCTION TO RED HAT CLOUD FOUNDATIONS

Welcome to the transformative world of on-premise cloud computing with Red Hat technologies. In this increasingly cloud-centric era, where public cloud providers dominate headlines, the power of building your own cloud infrastructure remains undiminished. This chapter lays the foundation for your journey into creating a robust, scalable, and fully integrated private cloud environment using Red Hat's enterprise-grade suite of technologies: OpenShift, OpenStack Services on OpenShift (RHOSO), and Ceph Storage.

Whether you're a seasoned architect seeking to modernize your data center or an IT professional exploring alternatives to public cloud dependence, this chapter will illuminate why an on-premise approach matters and how Red Hat's integrated stack provides unparalleled flexibility for modern workloads.

Why On-Premise Cloud Matters

The relentless marketing from public cloud giants like AWS, Azure, and Google Cloud might make on-premise infrastructure seem like a relic of the past. However, organizations across industries are rediscovering compelling reasons to maintain control of their cloud infrastructure:

1. Control and Security: Taking Ownership of Your Data Destiny

Public clouds operate on a shared responsibility model that often leaves critical security decisions in the hands of vendors. By contrast, on-premise clouds provide:

- **Complete infrastructure visibility**: When sensitive data and mission-critical workloads reside within your physical infrastructure, you gain transparency that public clouds simply cannot match.
- **Regulatory compliance assurance**: Industries governed by stringent regulations like GDPR, HIPAA, PCI-DSS, or FISMA can implement precise controls ensuring data sovereignty and compliance.
- **Defense-in-depth sovereignty**: Security architectures can be tailored precisely to organizational requirements rather than conforming to a provider's predefined models.
- **Physical access governance**: Maintaining physical control over hardware provides an additional security layer that public cloud environments cannot offer.

At a major European financial institution, implementing an on-premise cloud cut compliance verification time by 68% compared to their previous public cloud environment, as they no longer needed to navigate the complexity of shared responsibility models.

2. Cost Predictability: Escaping the Usage-Based Pricing Trap

The often-touted benefits of public cloud's "pay-as-you-go" model can transform into significant financial challenges:

- **Elimination of egress fee surprises**: On-premise infrastructure removes the unpredictable and often substantial costs of data egress that plague public cloud budgets.
- **Long-term planning stability**: Capital expenditure models create predictable cost structures that align perfectly with steady-state workloads and consistent resource needs.
- **Resource optimization control**: Direct visibility into utilization allows for targeted optimizations that public cloud opacity often prevents.
- **Hardware lifecycle management**: Amortize equipment costs over longer periods than public cloud providers typically rotate their hardware.

A mid-sized healthcare provider discovered their on-premise OpenShift and RHOSO deployment cost 42% less over three years compared to equivalent public cloud services for their steady-state workloads, with the gap widening for data-intensive operations.

3. Customization: Escaping the One-Size-Fits-All Paradigm

Public clouds excel at standardization, but many organizations require specialized environments:

- **Hardware specialization freedom**: Deploy GPU clusters for AI workloads, FPGAs for specialized processing, or high-memory configurations without premium pricing tiers.
- **Network architecture flexibility**: Implement intricate security zones, custom routing policies, or specialized VLAN configurations that precisely match your operational requirements.
- **Storage performance tuning**: Optimize storage tiers based on workload-specific requirements rather than choosing from predefined service levels.
- **Application-specific optimizations**: Tune every layer of the stack from bare metal through virtualization for specific application requirements.

A research institution deployed their on-premise cloud with specialized GPU nodes for AI and machine learning that outperformed equivalent public cloud offerings at 65% of the cost, while maintaining complete control over their proprietary algorithms and datasets.

4. Hybrid Flexibility: The Best of All Worlds

Rather than forcing an all-or-nothing choice, an on-premise foundation enables true hybrid architecture:

- **Workload-appropriate placement**: Deploy applications to the environment that best suits their requirements—containers on OpenShift, traditional VMs on RHOSO, burst capacity to public cloud.

- **Unified management plane**: Manage all environments through consistent tooling and interfaces regardless of where workloads reside.
- **Data gravity awareness**: Keep data-intensive processes close to your data stores while allowing lightweight processing in public environments.
- **Disaster recovery options**: Implement cross-environment resilience without vendor lock-in.

Red Hat's integrated ecosystem amplifies these benefits by delivering enterprise-grade, open-source tools with commercial support. Their technologies blend the innovation of open-source communities with the stability organizations require for production environments.

The Three Pillars: OpenShift, RHOSO, and Ceph

Your on-premise cloud's foundation rests on three integrated technologies that together deliver a complete platform for modern applications. Understanding each component's role is essential to appreciating how they form a cohesive whole.

1. Red Hat OpenShift Container Platform (OCP)

What It Is: OpenShift extends Kubernetes, the industry-standard container orchestration platform, with enterprise features that transform it from a powerful yet complex tool into a comprehensive application platform. OpenShift adds integrated developer workflows, robust multi-tenancy, advanced networking, and enterprise-grade security.

Why It Matters: Containers have revolutionized application development and deployment through:

- **Consistency across environments**: Applications packaged with their dependencies function identically from development to production.
- Resource efficiency: Containers share the host operating system kernel, enabling higher density than traditional virtualization.
- **DevOps enablement**: Immutable infrastructure patterns and declarative configuration align perfectly with modern CI/CD pipelines.
- **Microservice architecture support**: Lightweight containers provide the ideal deployment vehicle for decomposed applications.

OpenShift transforms raw Kubernetes into a complete platform by adding:

- **Integrated CI/CD tools**: Built-in image registries, build pipelines, and source-to-image capabilities.
- **Advanced networking**: The OVN-Kubernetes software-defined networking layer provides network policy enforcement, enhanced security, and integration points for complex networking requirements.
- **Automated operations**: Operators automate day-2 operations from scaling to upgrades.
- **Enterprise authentication**: Integration with LDAP, Active Directory, and other identity providers.

In This Book: You'll deploy OCP 4.18 using the Installer-Provided Infrastructure (IPI) method on a KVM-based lab environment. Chapter 3 guides you through this process step by step, creating the foundational layer for the rest of your cloud stack.

2. Red Hat OpenStack Services on OpenShift (RHOSO)

What It Is: RHOSO represents the next evolution of OpenStack—the open-source cloud computing platform that provides Infrastructure-as-a-Service capabilities. RHOSO containerizes traditional OpenStack services and runs them as Kubernetes applications on OpenShift, revolutionizing deployment and operational models.

Key OpenStack services deployed include:

- **Nova**: The compute service that schedules and manages virtual machines.
- **Neutron**: The networking service providing virtual networks, routers, and security groups.
- **Cinder**: The block storage service for persistent VM volumes.
- **Glance**: The image service managing VM templates.
- **Keystone**: The identity service providing authentication and authorization.
- **Heat**: The orchestration service enabling template-based infrastructure deployment.

Why It Matters: Despite the rise of containers, virtual machines remain essential for:

- **Legacy application support**: Monolithic applications not designed for containerization.
- **Operating system diversity**: Workloads requiring specific OS kernels or configurations.
- **Strong workload isolation**: Regulatory requirements mandating complete isolation.
- **Resource-intensive tasks**: Workloads needing dedicated compute resources.

RHOSO brings several transformative benefits:

- **Operational consistency**: Manage OpenStack services using the same Kubernetes-based patterns used for containerized applications.
- **Simplified lifecycle management**: Leverage Kubernetes Operators for upgrades, scaling, and configuration.
- **Resource efficiency**: Deploy the control plane components as right-sized containers rather than dedicated VMs.
- **Integrated security model**: Apply OpenShift's robust security controls to OpenStack components.

In This Book: Chapter 4 guides you through layering RHOSO atop your OpenShift deployment, establishing containerized control plane services on your OpenShift workers while configuring dedicated compute nodes for VM workloads.

3. Red Hat Ceph Storage

What It Is: Ceph is a distributed storage system designed for excellent performance, reliability, and scalability. It provides unified storage interfaces:

- **Block Storage (RBD):** Traditional block devices for VMs and databases.
- **Object Storage (RADOSGW)**: S3-compatible API for cloud-native applications.
- **File Storage (CephFS)**: POSIX-compliant file system for shared data access.

Why It Matters: Modern cloud environments demand storage that is:

- **Horizontally scalable**: Growing effortlessly from terabytes to petabytes.
- **Self-healing:** Automatically recovering from component failures.
- **Multi-protocol**: Supporting diverse application requirements through a single platform.
- **Cost-effective**: Leveraging commodity hardware while ensuring data durability.

Ceph excels by:

- Eliminating single points of failure: Distributing data across multiple storage nodes with configurable replication.
- Providing automated data management: Intelligent data placement algorithms optimize performance and availability.
- Enabling seamless scaling: Add capacity by simply introducing new nodes.
- Unifying storage access: Single system providing block, object, and file interfaces.

In This Book: Chapter 5 walks you through deploying a three-node Ceph cluster and integrating it with both OpenShift (for container persistent storage) and RHOSO (for VM volumes and images). This integration creates a unified storage foundation for your entire cloud stack.

4. How They Work Together: The Integrated Stack

The power of this architecture comes from the seamless integration between components:

- **OpenShift provides**: The container platform hosting both application workloads and the RHOSO control plane.
- **RHOSO delivers**: VM orchestration capabilities, managing compute nodes that host traditional workloads.
- **Ceph supplies**: The storage backbone serving both container volumes and VM disks.

This integration enables:

- **Unified management**: Consistent operational patterns across container and VM environments.
- **Workload flexibility**: Deploy applications in their optimal format—container or VM—without siloed infrastructure.
- **Resource optimization**: Efficiently utilize infrastructure across different workload types.
- **Future-proof architecture**: Support both cloud-native and traditional applications as your organization evolves.

Your Lab Journey: What to Expect

This book transcends theoretical discussions by providing a complete, hands-on implementation path. Using a single KVM host—named lab1 throughout the book—you'll build a fully functional on-premise cloud that demonstrates each concept in practice.

The Path Forward

Your journey unfolds across four carefully structured parts:

Part I: Getting Started with Red Hat Cloud (Chapters 1-2)

- Chapter 1: Introduction to Red Hat Cloud Foundations (this chapter)
- Chapter 2: Setting Up Your Lab Environment
  - Establish your KVM hypervisor on Ubuntu 22.04
  - Configure four virtual switches (management, provisioning, baremetal, isolated01)
  - Deploy the provisioner node that will bootstrap your environment
  - Configure networking, DNS, and other foundational services

Part II: Deploying the Core Stack (Chapters 3-5)

- Chapter 3: Deploying OpenShift On-Premise
  - Install OpenShift 4.18 using the Installer-Provided Infrastructure (IPI) method
  - Configure networking, storage, and monitoring
  - Validate the deployment with comprehensive health checks
- Chapter 4: Integrating Red Hat OpenStack Services (RHOSO)
  - Deploy the RHOSO control plane on OpenShift
  - Configure networking for tenant isolation
  - Prepare compute nodes for VM workloads
  - Test basic VM deployment capabilities
- Chapter 5: Adding Ceph Storage to the Mix
  - Deploy a three-node Ceph cluster
  - Integrate Ceph with RHOSO for VM storage
  - Configure Ceph as persistent storage for OpenShift
  - Validate the complete storage integration

Part III: Securing and Optimizing Your Cloud (Chapters 6-7)

- Chapter 6: Securing Your Cloud Infrastructure
  - Implement identity management with Red Hat IDM
  - Configure certificate-based security
  - Establish proper authentication for both OpenShift and RHOSO
- Chapter 7: Monitoring and Performance Tuning
  - Deploy monitoring tools across the stack
  - Optimize resource allocation and utilization
  - Implement performance tuning best practices
  - Work through a real-world performance bottleneck diagnosis

Part IV: Advanced Operations and Applications (Chapters 8-10)

- Chapter 8: Operating RHOSO in Production
    - Manage multi-tenant VM workloads
    - Implement Heat templates for orchestration
    - Establish operational procedures
- Chapter 9: Scaling Your Cloud Infrastructure
    - Add nodes to both OpenShift and RHOSO environments
    - Scale storage with additional Ceph nodes
    - Address a high-traffic application scenario
- Chapter 10: Deploying Real-World Applications
    - Implement GitOps workflows with Argo CD
    - Deploy a multi-tier application spanning containers and VMs
    - Establish CI/CD pipelines for hybrid applications

Your Lab Environment

Throughout this book, you'll build a complete environment using virtualized infrastructure on a single host:

Physical Host: lab1

- Operating System: Ubuntu 22.04 LTS (Optional)
- Compute: 32 CPU cores
- Memory: 128 GB RAM
- Storage: 3.2 TB SSD
- Virtualization: KVM with Open vSwitch (OVS)

Virtual Nodes:

- OpenShift Control Plane (cp01-03): Three nodes hosting the Kubernetes control plane
- OpenShift Workers (comp01-03): Three nodes for container workloads and RHOSO control plane
- RHOSO Compute (ospcomp01-03): Three nodes dedicated to VM hosting
- Ceph Storage (cephrhocp18-20): Three nodes providing distributed storage
- Provisioner: One utility node for deployment orchestration

While implemented on virtualized infrastructure for accessibility, all concepts and configurations directly apply to bare-metal deployments, making this a perfect learning environment with direct production applicability.

What You'll Build: The End Result

By the conclusion of Chapter 10, you'll have created a production-ready hybrid cloud environment capable of:

- **Hosting containerized applications**: Web services, microservices, and cloud-native applications
- **Running virtual machines**: Traditional applications, databases, and specialized workloads
- **Providing self-service infrastructure**: Developer-friendly interfaces for both container and VM provisioning
- **Enabling automated operations**: GitOps-driven application deployment and infrastructure management
- **Supporting multi-tenancy**: Isolated environments for different teams or applications
- **Ensuring high availability**: Resilient services that withstand component failures

The table below summarizes the key milestones in your journey:

Table 1.1: Your Lab Milestones

Stage	Achievement	Chapter
Lab Setup	Functional KVM host	2
OpenShift Deployed	Running OCP cluster	3
RHOSO Added	VM orchestration ready	4
Ceph Integrated	Unified storage online	5
Secured & Optimized	Production-ready cloud	6-7
Scaled & Applied	Real-world app deployed	8-10

To support your implementation, all configuration files, scripts, and examples are available in the companion GitHub repository at github.com/BAristide/OCPRHOSO2025, ensuring you have concrete references for each step of the process.

By following this journey from start to finish, you'll gain not just theoretical knowledge but practical experience in building, operating, and optimizing a complete on-premise cloud environment based on Red Hat's integrated platform.

Summary

The on-premise cloud approach, powered by Red Hat's integrated technologies, offers a compelling alternative to public cloud exclusivity. By building your own hybrid foundation with OpenShift, RHOSO, and Ceph, you gain control, flexibility, and cost predictability without sacrificing the agility and operational efficiency that modern IT demands.

In the chapters that follow, we'll transform this vision into reality, building each component step by step until you have a complete, functional on-premise cloud. Whether your goal is learning, testing, or production deployment, the knowledge and experience gained will provide a solid foundation for your cloud journey.

Let's begin by setting up the laboratory environment that will host our cloud stack.

CHAPTER 2: SETTING UP YOUR LAB ENVIRONMENT

Building a robust hybrid cloud foundation begins with a proper lab environment. This chapter guides you through creating a comprehensive virtualized infrastructure that mirrors enterprise deployments while remaining accessible on a single physical host. You'll establish a KVM-based environment capable of supporting OpenShift, Red Hat OpenStack Services on OpenShift (RHOSO), and Ceph storage—the three pillars of our on-premise cloud architecture.

Unlike traditional approaches that rely on pre-configured nodes, we'll embrace automation through the Installer-Provided Infrastructure (IPI) method for OpenShift and the Bare Metal Operator (BMO) for RHOSO compute resources. This approach better reflects real-world deployment scenarios and teaches valuable skills for managing infrastructure as code.

By the end of this chapter, you'll have:

- A fully configured KVM hypervisor with performance-optimized settings
- A network topology with four distinct virtual switches supporting VLANs
- A provisioner node serving as your administrative control center
- Unprovisioned virtual machines ready for automated deployment
- Power management capabilities via Virtual BMC (vBMC)

This foundation will serve as the launching pad for OpenShift deployment (Chapter 3) and RHOSO integration (Chapter 5), setting the stage for your journey into the hybrid cloud.

Infrastructure Architecture Overview

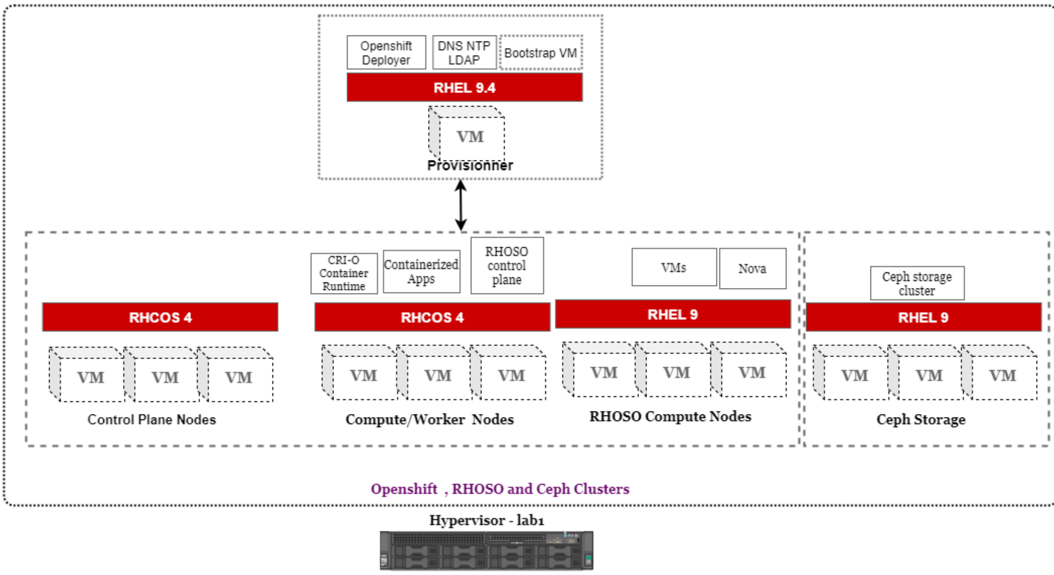

Figure 1 Openshift, RHOSO and Ceph Nodes on a single physical node

Our lab environment places all components—from OpenShift control plane nodes to RHOSO compute resources and Ceph storage—on a single physical server using KVM virtualization. This approach balances accessibility with realism, allowing you to gain hands-on experience with enterprise-grade architectures without requiring multiple physical servers.

The infrastructure consists of six distinct node types, each with specific roles in our hybrid cloud ecosystem:

Node Type	Role	Count
Provisioner	Deployment orchestration and management	1
Bootstrap VM	Temporary node for OpenShift initialization	1(transient)
Control Plane	OpenShift cluster management	3
Compute/Worker	Application workload hosting	3
RHOSO Compute	OpenStack VM hosting	3
Ceph Storage	Distributed storage for cluster services	3

Key Infrastructure Components

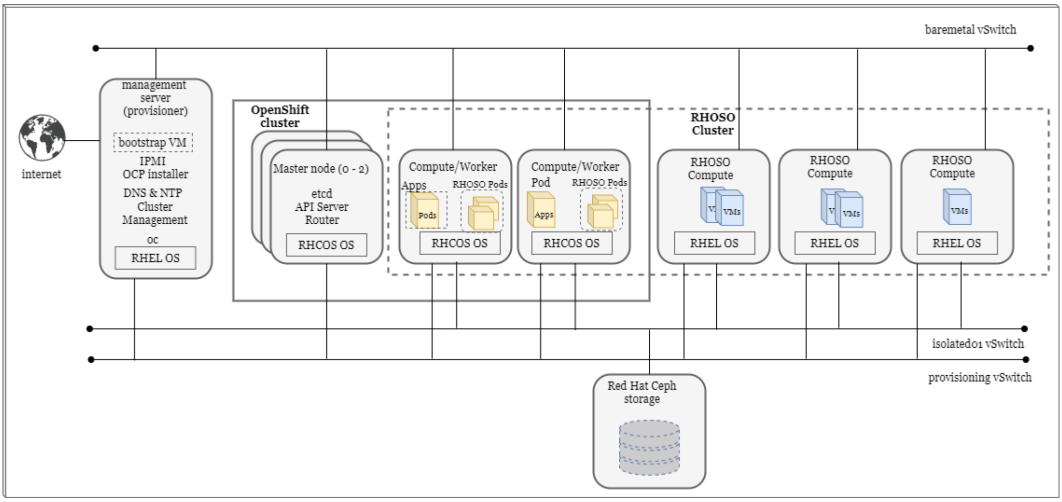

Figure 2 Infrastructure Components

Provisioner Node

The provisioner node (provisioner.ecintelligence.lab) serves as the command center for your entire deployment:

- **Core Function**: Executes the OpenShift installer
- **OS**: Red Hat Enterprise Linux (RHEL) 9.4
- **Key Services:**
  - DNS resolution for cluster domains
  - NTP time synchronization
  - Red Hat IdM for authentication (Chapter 6)
  - KVM for hosting the bootstrap virtual machine
- **Connectivity**: Bridges three networks—provisioning (172.22.0.0/24), baremetal (10.0.0.0/24), and management (10.10.0.0/16)

The provisioner acts as both the deployment engine and gateway for your lab environment, enabling communication between your cluster and external networks.

Bootstrap VM

The bootstrap VM is a critical yet ephemeral component in the OpenShift deployment process:

- **Purpose**: Initializes the control plane by providing Ignition configuration files
- **Lifecycle**: Automatically created and removed by the OpenShift installer
- **OS**: Red Hat CoreOS (RHCOS) 4.18
- **Deployment**: Created on the provisioner node during the initial phase of OpenShift installation

This temporary node handles the "chicken-and-egg" problem of establishing the first control plane nodes before the cluster is operational.

Control Plane Nodes

The three control plane nodes (`cp01-03`) form the management backbone of your OpenShift cluster:

- **Core Function**: Host the Kubernetes API server, scheduler, controller manager, and etcd database
- **Hardware**: Each VM configured with 16GB RAM, 8 vCPUs, and 150GB storage
- **Networking**: Dual NICs bonded to the baremetal network (10.0.0.60-62)
- **OS**: Red Hat CoreOS (RHCOS) 4.18, provisioned via IPI

These nodes maintain the cluster's state, handle API requests, and coordinate scheduling decisions—demonstrating the high-availability principles critical for production environments.

Compute/Worker Nodes

Compute nodes (`comp01-03`) run your containerized applications and the RHOSO control plane:

- **Core Function**: Host application workloads and platform services
- **Hardware**: Each VM equipped with 32GB RAM, 16 vCPUs, and 150GB storage
- **Networking**: Four NICs arranged in two bonds:
  - Bond 1: Connected to baremetal (10.0.0.63-65)
  - Bond 2: Connected to isolated01 for tenant, API, and storage traffic
- **OS**: Red Hat CoreOS (RHCOS) 4.18, provisioned through the OpenShift Machine API

These robust nodes handle the computational workload of both your applications and the OpenStack control plane components.

RHOSO Compute Nodes

RHOSO compute nodes (ospcomp01-03) specifically host OpenStack virtual machines:

Core Function: Run Nova compute instances for tenant workloads

- **Hardware**: 16GB RAM, 8 vCPUs, and 150GB storage per VM
- **Networking**: Four NICs in two bonds, mirroring the worker node configuration but with dedicated addresses
- **OS**: RHEL 9.4, deployed via the OpenStack DataPlane Operator

These nodes represent the OpenStack hypervisor layer, where tenant VMs will be instantiated in later chapters.

Ceph Storage Nodes

Ceph storage nodes (cephrhocp18-20) provide distributed storage services:

- **Core Function**: Host Ceph Monitor, OSD, and Manager daemons
- **Hardware**: 16GB RAM, 8 vCPUs, and 250GB storage per VM
- **Networking**: Dual NICs—one to baremetal (10.0.0.115-117) and one to isolated01 (172.18.0.115-117)
- **OS**: RHEL 9.4, pre-provisioned with QCOW2 images

These nodes deliver the storage foundation for both OpenShift (via ODF) and OpenStack (via Cinder/Glance integration).

Hardware and Software Prerequisites

1. Physical Host Requirements

Your physical server (lab1) must meet specific requirements to support this virtualized environment effectively:

Component	Minimum Specification	Recommended Specification	Purpose
CPU	16 cores	32 cores	VM execution and hypervisor overhead
RAM	96 GB	128 GB	VM memory allocation and host operations
Storage	1 TB SSD	3.2 TB SSD	VM images, snapshots, and ephemeral storage
Network	1 GbE NIC	10 GbE NIC	External connectivity
Virtualization	VT-x/AMD-V support	VT-x/AMD-V with nested virtualization	Hardware acceleration

The recommended specifications ensure optimal performance when running all 13 virtual machines simultaneously. While the minimum specifications may work, you may experience resource contention during intensive operations.

2. Software Requirements

The physical host should run one of these operating systems:

- **Ubuntu 22.04 LTS (used in our examples)**
- CentOS Stream 9
- Red Hat Enterprise Linux (RHEL) 9

Additionally, the CPU must support hardware virtualization extensions (Intel VT-x or AMD-V), which must be enabled in the BIOS. You can verify this with:

```
lscpu | grep Virtualization
```

Look for "VT-x" (Intel) or "AMD-V" (AMD) in the output. If these are absent, you'll need to enable them in your system BIOS before proceeding.

3. Hardware Specifications for Virtual Machines

Each virtual machine has specific resource requirements to ensure proper functionality:

Table 1 Hardware requirements of machines

Server	Role	Hardware				IP adress		Virtual/Physical	Operating system
		CPU Core	RAM (GB)	Network interface	Storage (GB) - SSD	DNS/NTP, Private address	vIPMIwith admin access		
Lab1	On-premise HW	32	128	1	3200	No	No	Physical	Ubuntu 22.04
provisioner	DNS, NTP, router	8	16	3	250	Yes	No	Virtual	RHEL 9.2
cp01	OCP - Master	8	16	4	150	Yes	Yes	Virtual	RHCOS 9.16
cp02	OCP - Master	8	16	4	150	Yes	Yes	Virtual	RHCOS 9.16
cp03	OCP - Master	8	16	4	150	Yes	Yes	Virtual	RHCOS 9.16
comp01	Compute/worker	16	32	4	150	Yes	Yes	Virtual	RHCOS 9.16
comp02	Compute/worker	16	32	4	150	Yes	Yes	Virtual	RHCOS 9.16
comp03	Compute/worker	16	32	4	150	Yes	Yes	Virtual	RHCOS 9.16
ospcomp01	RHOSO compute	8	16	4	150	Yes	Yes	Virtual	RHEL 9.4
ospcomp02	RHOSO compute	8	16	4	150	Yes	Yes	Virtual	RHEL 9.4
ospcomp03	RHOSO compute	8	16	4	150	Yes	Yes	Virtual	RHEL 9.4
cephrhocp18	Ceph Storage	8	16	4	250	Yes	No	Virtual	RHEL 9.2
cephrhocp19	Ceph Storage	8	16	4	250	Yes	No	Virtual	RHEL 9.2
cephrhocp20	Ceph Storage	8	16	4	250	Yes	No	Virtual	RHEL 9.2

This allocation ensures adequate resources for each node while maintaining a realistic representation of production environments.

Network Architecture

1. Network Topology

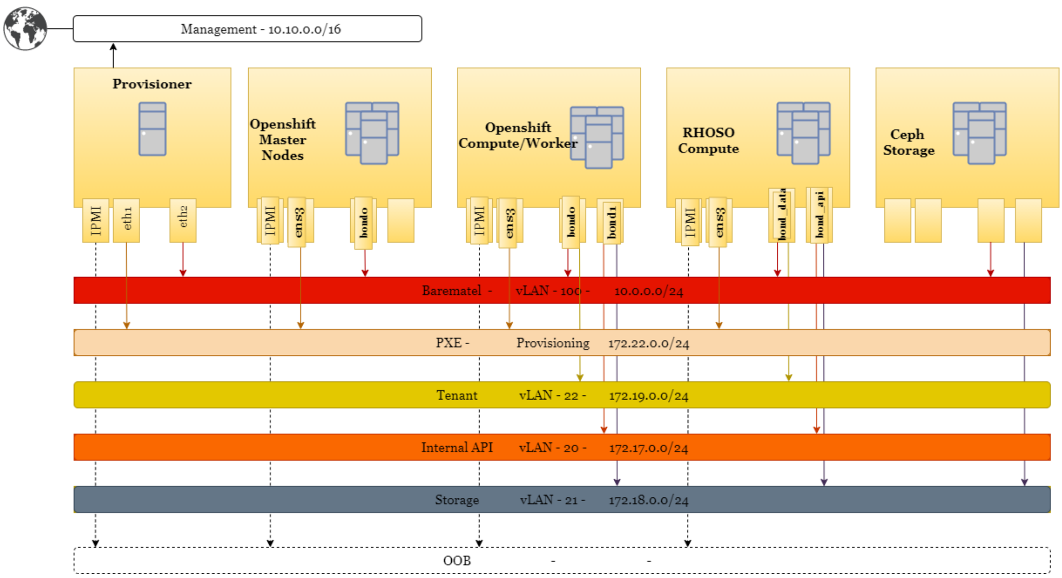

Figure 3 Global lab Network topology

Our lab environment uses four distinct networks, each serving a specific purpose in the hybrid cloud architecture:

Table 2 Lab network addressing plan

Network name	VLAN ID	Subnet	Purpose	Associated vSwitch
Management	-	10.10.0.0/16	Out-of-band access, DNS, and vBMC	management
Provisioning	-	172.22.0.0/24	PXE boot and automated deployment	provisionnet
Baremetal	100	10.0.0.0/24	Cluster traffic	baremetal
Tenant	22	172.19.0.0/24	Isolated tenant workloads	baremetal
Internal API	20	172.17.0.0/24	RHOSO API communication	isolated01
Storage	21	172.18.0.0/24	Ceph storage replication	isolated01

This network design isolates different traffic types while providing connectivity where needed, reflecting enterprise networking best practices.

2. Detailed Network Planning

OpenShift Cluster Network Plan

Table 3 OCP Cluster network planning

Device name	IPMI	provisioning	Public IP	Gateway	Netmask	DNS IP
cp01.ocp4.ecintelligence.lab	10.10.0.1: 60079	172.22.0.XX/24	10.0.0.60	10.0.0.2	255.255.255.0	10.0.0.2
cp02.ocp4.ecintelligence.lab	10.10.0.1: 60080	172.22.0.XX/24	10.0.0.61	10.0.0.2	255.255.255.0	10.0.0.2
cp03.ocp4.ecintelligence.lab	10.10.0.1: 60081	172.22.0.XX/24	10.0.0.62	10.0.0.2	255.255.255.0	10.0.0.2
comp01.ocp4.ecintelligence.lab	10.10.0.1: 60078	172.22.0.XX/24	10.0.0.63	10.0.0.2	255.255.255.0	10.0.0.2
comp02.ocp4.ecintelligence.lab	10.10.0.1: 60077	172.22.0.XX/24	10.0.0.64	10.0.0.2	255.255.255.0	10.0.0.2
comp03.ocp4.ecintelligence.lab	10.10.0.1: 60076	172.22.0.XX/24	10.0.0.68	10.0.0.2	255.255.255.0	10.0.0.2

RHOSO Cluster Network Plan

This extended network configuration addresses the additional network requirements for OpenStack services.

Table 4 RHOSO cluster network planning

Device name	IPMI	Provisioning /24	Public IP (vlan 100) /24	Tenant Vlan 22 /24	Internalapi Vlan 20 /24	Storage Vlan 21 /24
		ens3	**bond0**	**bon0**	**bond1**	**bond1**
comp01.ocp4.ecintelligence.lab	10.10.0.1: 60079	172.22.0.x	10.0.0.63	172.19.0.10	172.17.0.10	172.18.0.10
comp02.ocp4.ecintelligence.lab	10.10.0.1: 60080	172.22.0.x	10.0.0.64	172.19.0.11	172.17.0.11	172.18.0.11
comp03.ocp4.ecintelligence.lab	10.10.0.1: 60081	172.22.0.x	10.0.0.68	172.19.0.12	172.17.0.12	172.18.0.12
ospcomp01.ocp4.ecintelligence.lab	10.10.0.1: 60078	172.22.0.110	10.0.0.80	172.19.0.71	172.17.0.229	172.18.0.91
ospcomp02.ocp4.ecintelligence.lab	10.10.0.1: 60077	172.22.0.111	10.0.0.81	172.19.0.72	172.17.0.230	172.18.0.92
ospcomp03.ocp4.ecintelligence.lab	10.10.0.1: 60076	172.22.0.112	10.0.0.82	172.19.0.73	172.17.0.231	172.18.0.93

Provisioner

Table 5 Provisioner node addressing

Device name	management	Provisioning	baremetal	Gateway	DNS IP
provisioner.ocp4.ecintelligence.lab	10.10.111.50	172.22.0.5/24	10.0.0.2	10.10.0.1/16	10.0.0.2

The provisioner node acts as the gateway for all cluster nodes, bridging the internal networks with external connectivity.

Bridging Virtual and Physical Networks

Understanding how our virtualized lab environment maps to physical infrastructure is essential for applying these skills in production environments.

Physical network Topology

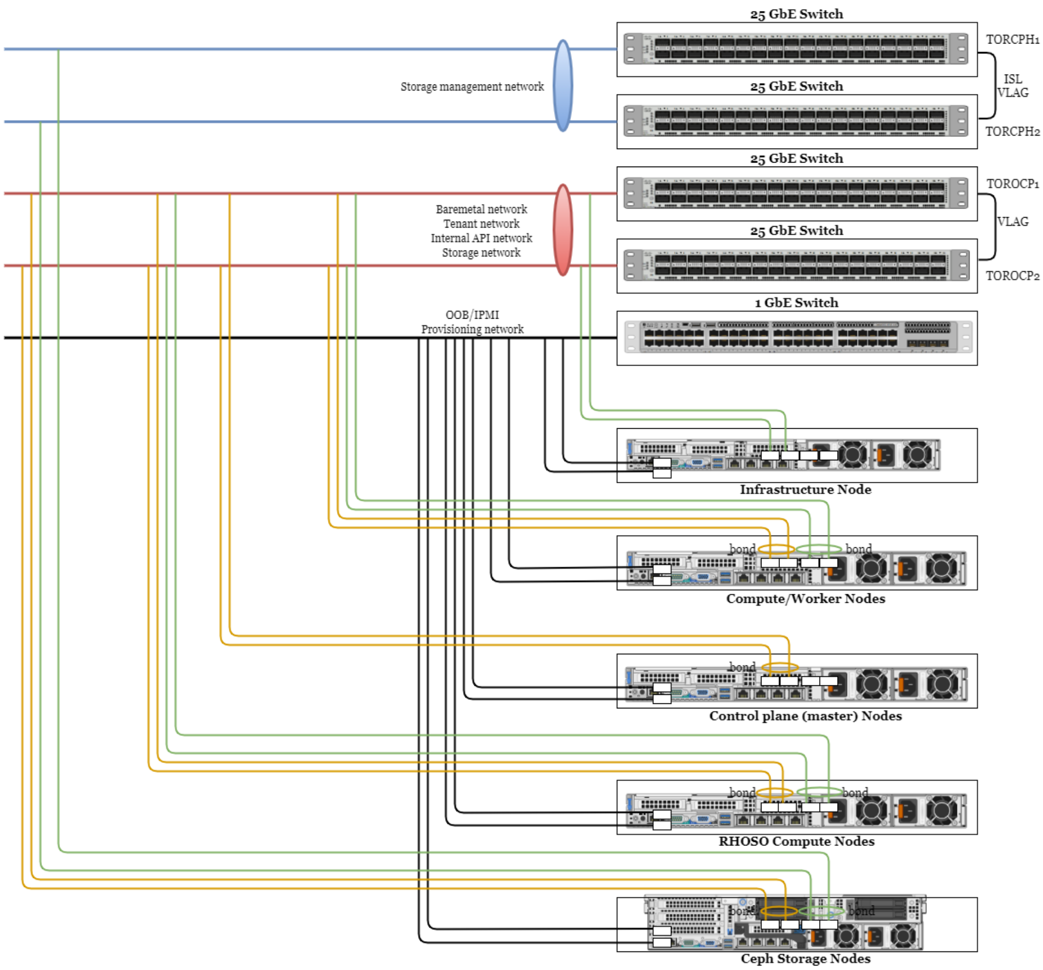

Figure 4 Physical Topology

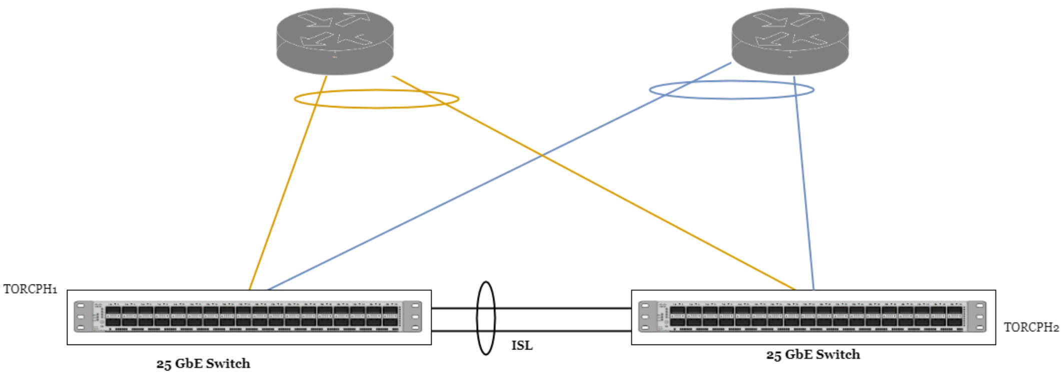

Figure 5 Redundant network architecture

In a production environment, your hybrid cloud leverages dedicated switches and bonded NICs for redundancy and performance. Here's the design:

Ceph Storage Switches:

- Two Top of Rack (TOR) switches, TORCPH1 and TORCPH2 (25 GbE), dedicated to Ceph's management network (e.g., OSD replication, heartbeat).
- Configured in a cluster with Virtual Link Aggregation Group (VLAG) for high availability (HA).

Cluster Traffic Switches:

- Two TOR switches, TOROCP1 and TOROCP2 (25 GbE), handling all isolated traffic:
    - o *Baremetal Network*: Cluster communication (10.0.0.0/24, VLAN 100).
    - o *Tenant Network*: RHOSO tenant VMs (e.g., 172.19.0.0/24, VLAN 22).
    - o *Internal API Network*: Service interactions (172.17.0.0/24, VLAN 20).
    - o *Storage Network*: Ceph access (172.18.0.0/24, VLAN 21).
- Clustered with VLAG for fault tolerance.

Management Switch:

- One 1 GbE switch for provisioning (172.22.0.0/24) and out-of-band (OOB) IPMI management.

Node Connectivity:

- All Nodes: Dual 1 GbE connections to the management switch: one for IPMI (e.g., 10.10.0.x), one for provisioning.
- RHOSO Compute and OCP Worker Nodes (ospcomp01-03, comp01-03):
    - o Four 25 GbE NICs in two bonds (LACP or active-backup):
    - o Bond 1 (NIC1 to TOROCP1, NIC2 to TOROCP2): `Baremetal` and `tenant` traffic.
    - o Bond 2 (NIC3 to TOROCP1, NIC4 to TOROCP2): `Internal API` and `storage` traffic.
- OCP Control Plane Nodes (cp01-03):

  - Two 25 GbE NICs in one bond:
  - NIC1 to TOROCP1, NIC2 to TOROCP2 for baremetal traffic.
- *Ceph Storage Nodes (cephrhocp18-20):*
  - Four 25 GbE NICs in two bonds:
  - Bond 1 (NIC1 to TOROCP1, NIC2 to TOROCP2): Cluster Storage traffic.
  - Bond 2 (NIC3 to TORCPH1, NIC4 to TORCPH2): Ceph management.
- Infrastructure Node (provisioner): Connected to TOROCP1 and management switch.

Lab Environment Network Topology

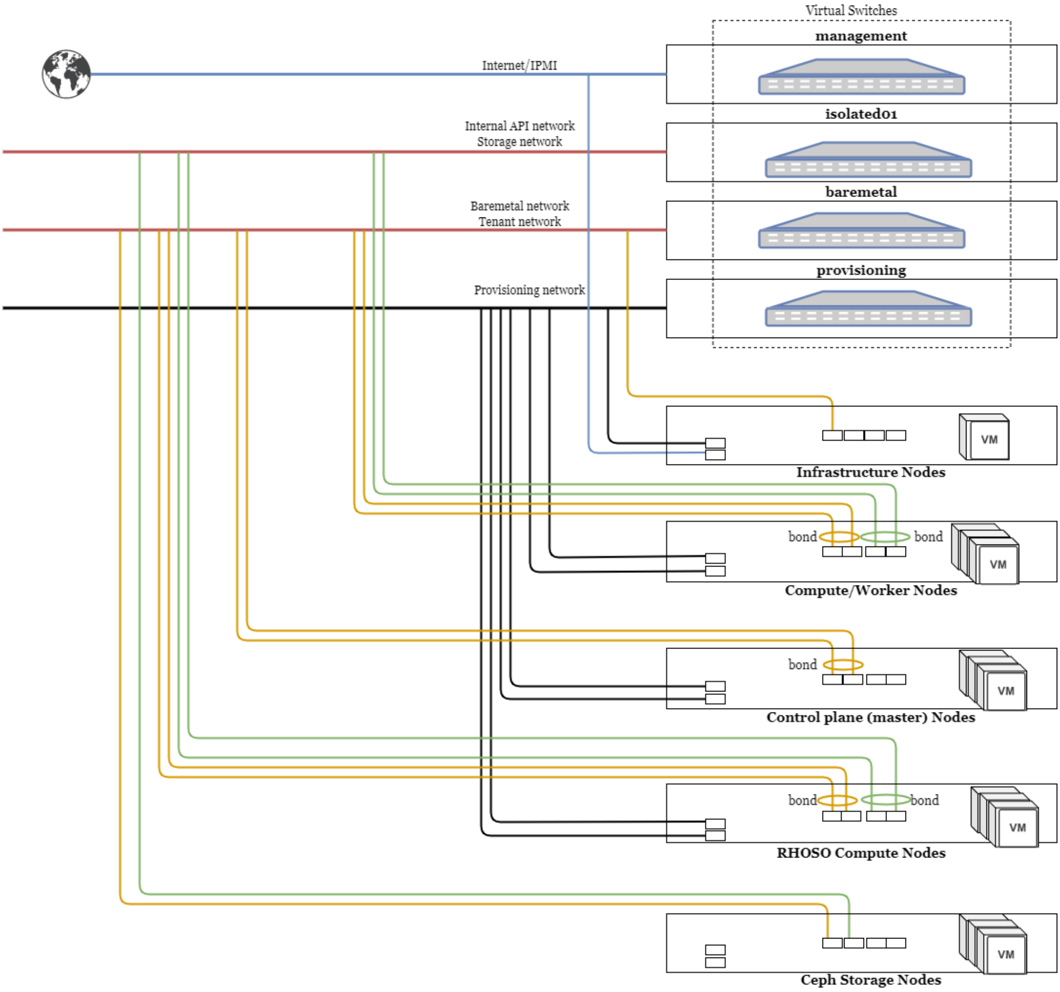

Figure 6 Lab Environment Network Topology

In our KVM lab on lab1, all nodes are VMs, simplifying setup while mirroring physical principles:

Virtual Switches:

- management: 1 GbE for OOB (10.10.0.0/16).
- provisionnet: Provisioning (172.22.0.0/24).
- baremetal: Cluster traffic (10.0.0.0/24, VLAN 100).
- isolated01: Isolated traffic (tenant, internal API, storage; VLANs 20-22).

Node Connectivity:

Provisioner and Ceph Nodes: Pre-provisioned with RHEL 9.4 QCOW2 images.

Other Nodes (cp01-03, comp01-03, ospcomp01-03): Unprovisioned VMs, deployed via IPI (Ch. 3-4).

RHOSO Compute and OCP Worker Nodes:

- Four NICs in two bonds:
- Bond 1 (NIC1-2 to baremetal): Cluster traffic.
- Bond 2 (NIC3-4 to isolated01): Isolated traffic.

OCP Control Plane Nodes:

- Two NICs in one bond (NIC1-2 to baremetal).

Ceph Storage Nodes:

- Two NICs: one to baremetal, one to isolated01.

Provisioner: Connected to provisionnet, baremetal, and management, acting as the gateway.

Mapping Virtual to Physical

- provisionnet → Management Switch: Both handle provisioning and OOB.
- baremetal → TOROCP1-2 Bond 1: Cluster traffic aligns with physical baremetal network.
- isolated01 → TOROCP1-2 Bond 2 / TORCPH1-2: Isolated traffic splits into tenant/API/storage (TOROCP) and Ceph management (TORCPH) in physical setups.
- Bonding: Virtual bonds simulate physical LACP or redundancy.

This mapping ensures your lab skills translate directly to physical deployments, whether on a single host or a rack of servers.

Setting Up the KVM Hypervisor on lab1

Understanding KVM

Kernel-based Virtual Machine (KVM) transforms a standard Linux system into a Type-1 hypervisor by utilizing hardware virtualization extensions. Unlike standalone hypervisors, KVM integrates directly with the Linux kernel, benefiting from its mature memory management, I/O handling, and scheduling capabilities.

KVM works in concert with QEMU, which provides hardware emulation for virtual devices. This KVM-QEMU combination delivers near-native performance while offering the flexibility needed for our complex lab environment.

Installing KVM on Ubuntu 22.04

Begin by updating your system and installing the required packages:

```
sudo apt update
sudo apt upgrade -y
sudo apt install -y qemu-kvm libvirt-daemon bridge-utils \
virtinst libvirt-daemon-system openvswitch-switch openvswitch-common
```

- **qemu-kvm**: Provides the KVM hypervisor and emulator.
- **libvirt-daemon**: Manages virtualization capabilities.
- **bridge-utils**: Tools for configuring network bridges.
- **virtinst**: Utilities for creating and managing VMs.

Verify that the host CPU supports hardware virtualization:

```
cat /proc/cpuinfo | grep -E "vmx|svm"
```

You should see "vmx" (Intel) or "svm" (AMD) in the output. If not, you'll need to enable virtualization in your system's BIOS.

Next, ensure the libvirt service is active and set to start automatically:

```
sudo systemctl enable libvirtd openvswitch-switch
sudo systemctl start libvirtd openvswitch-switch
sudo systemctl status libvirtd openvswitch-switch
```

A “running” status confirms readiness. Configure a default storage pool for VM disks:

```
# sudo virsh pool-define-as --name default \
--type dir --target /var/lib/libvirt/images
# sudo virsh pool-start default
# sudo virsh pool-autostart default
```

This pool uses `/var/lib/libvirt/images`, leveraging the 3.2 TB SSD for VM storage. Check with sudo `virsh pool-list --all`—“default” should be “active” and “autostart.”

Configuring Network vSwitches

Our lab requires four virtual switches, each serving a specific purpose in the network topology. Two of these will be standard libvirt bridges, while the other two will use Open vSwitch (OVS) for VLAN support.

1. Installing Open vSwitch (OVS)

Open vSwitch (OVS) provides advanced networking capabilities, including VLAN tagging and software-defined networking features:

```
sudo apt install -y openvswitch-switch openvswitch-common
sudo systemctl start openvswitch-switch
sudo systemctl enable openvswitch-switch
```

2. Creating the Management vSwitch

The management network provides out-of-band access and Internet connectivity:

```
cat > management.xml << EOF
<network>
  <name>management</name>
  <bridge name="management"/>
  <forward mode="nat" dev="eno1"/>
  <ip address="10.10.0.1" netmask="255.255.0.0"/>
</network>
EOF

sudo virsh net-define management.xml
sudo virsh net-start management
sudo virsh net-autostart management
```

eno1 is your physical NIC interface on **lab1** (connected to internet)

3. Creating the Provisioning vSwitch

The provisioning network handles PXE booting and automated deployment:

```
cat > provision-network02.xml << EOF
<network>
  <name>provisionnet</name>
  <bridge name="provisionnet"/>
</network>
EOF

sudo virsh net-define provision-network02.xml
sudo virsh net-start provisionnet
sudo virsh net-autostart provisionnet
```

4. Creating the Baremetal vSwitch with OVS

The baremetal network carries OpenShift cluster traffic and requires VLAN support:

```
sudo ovs-vsctl add-br baremetal

cat > baremetal-network.xml << EOF
<network>
  <name>baremetal</name>
  <forward mode="bridge"/>
  <bridge name="baremetal"/>
  <virtualport type="openvswitch"/>
</network>
EOF

sudo virsh net-define baremetal-network.xml
sudo virsh net-start baremetal
sudo virsh net-autostart baremetal
```

5. Creating the Isolated01 vSwitch with OVS

The isolated01 network handles tenant, API, and storage traffic with VLAN isolation:

```
sudo ovs-vsctl add-br isolated01

cat > isolated01-network.xml << EOF
<network>
  <name>isolated01</name>
  <forward mode="bridge"/>
  <bridge name="isolated01"/>
  <virtualport type="openvswitch"/>
</network>
EOF

sudo virsh net-define isolated01-network.xml
sudo virsh net-start isolated01
sudo virsh net-autostart isolated01
```

Verify all virtual networks are properly configured:

```
sudo virsh net-list --all
sudo ovs-vsctl list-br
```

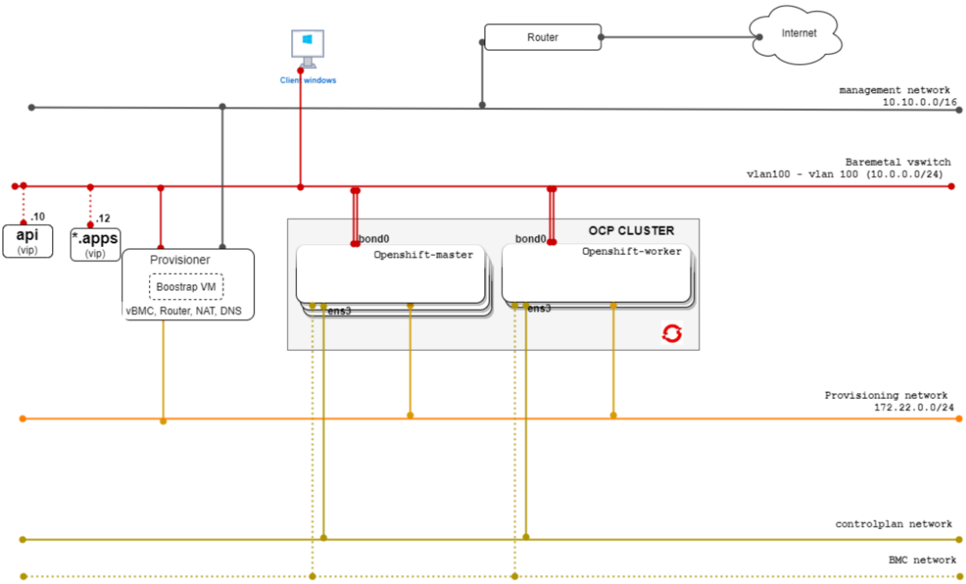

Figure 7 Low-Level Lab Architecture for OCP cluster