



Title
Author



Title
Author

VMware Certified Professional - Data Center Virtualization (VCP-DCV) Certification Guide

*Master data center virtualization with
expert VMware vSphere guidance*

Dinesh Shaw



www.bpbonline.com

First Edition 2026

Copyright © BPB Publications, India

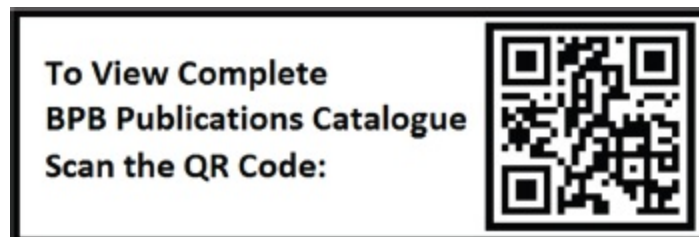
ISBN: 978-93-65897-340

All Rights Reserved. No part of this publication may be reproduced, distributed or transmitted in any form or by any means or stored in a database or retrieval system, without the prior written permission of the publisher with the exception to the program listings which may be entered, stored and executed in a computer system, but they can not be reproduced by the means of publication, photocopy, recording, or by any electronic and mechanical means.

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true and correct to the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but the publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.



www.bpbonline.com

Dedicated to

My beloved wife:

Shila

and

*My daughters ***Samriddhi*** and ***Shanvi****

About the Author

Dinesh Shaw is a very experienced senior principal system development engineer and VMware expert with over 16 years of IT infrastructure experience. He has experience in the areas of compute, virtualization, networking, and storage solutions and has spent over a decade designing, implementing, and optimizing VMware vSphere infrastructures.

Over the years in his career, Dinesh has worked with major global organizations such as Dell Technologies and Cognizant Technology Solutions and has developed/deployed converged and hyper-converged infrastructure solutions that boosted scalability, reduced provisioning times, and enhanced system performance for critical business functions. His real-world experience includes VMware vSphere, ESXi, vCenter, NSX, vSAN, VCF, SRM, HCX, and lifecycle management tools and automations using PowerShell, Python, and VMware SDKs.

Aside from work related to technology, Dinesh is a committed leader and mentor. He authored technical articles, conducted workshops, and taught engineers VMware best practices and shaped the future leaders of the field of virtualization professionals.

He also possesses various industry certifications, such as the **VMware Certified Professional - Data Center Virtualization (VCP-DCV)**, **VMware Certified Professional - Network Virtualization -(VCP-NV)**, CCNA/CCNP Data Center, and Dell Gen AI Foundation.

In this book, the author incorporates both long-time technical experience and real-world experience to help IT professionals successfully learn VMware vSphere and achieve success at the VCP-DCV certification.

About the Reviewer

Mohamed Sokarno is a senior software and data centre engineer with a strong passion for technology and a focus on virtualization and modern infrastructure solutions. He specializes in VMware technologies and holds multiple VMware certifications, enabling him to design and implement scalable, reliable, and efficient data-centre environments.

He thrives on solving complex technical challenges and helping organizations unlock the full value of their infrastructure. Combining deep technical expertise with a strong understanding of business needs, he ensures that every solution aligns with industry best practices and delivers meaningful, measurable outcomes.

Acknowledgement

I want to express my deepest gratitude to my family and friends for their unwavering support and encouragement throughout this book's writing, especially my wife Shila and my daughters Samriddhi and Shanvi.

I am also grateful to BPB Publications for their guidance and expertise in bringing this book to fruition. It was a long journey of revising this book, with valuable participation and collaboration of reviewers, technical experts, and editors.

I would also like to acknowledge the valuable contributions of my colleagues and co-worker during many years working in the tech industry, who have taught me so much and provided valuable feedback on my work.

Finally, I would like to thank all the readers who have taken an interest in my book and for their support in making it a reality. Your encouragement has been invaluable.

Preface

Virtualization has transformed today's IT by altering the way organizations develop, utilize, and maintain systems. The key to such transformation is VMware vSphere, the industry-leading virtualization platform that enables businesses to reduce expenses, enhance flexibility, and simplifies operations. For IT specialists, understanding vSphere goes beyond a technical expertise, such a knowledge is a valuable skill that brings new opportunities in the field of cloud computing and data center administration.

This book, *VMware Certified Professional - Data Center Virtualization (VCP-DCV) Certification Guide*, is reader's complete guide to the VCP-DCV certification. It belongs to those: if they are new to virtualization or if they want to enhance their skills. This guide provides readers with structured learning, hands-on labs, exercises grounded in real-world situations, and exam tips.

Readers will move step by step - from introductory coverage of concepts of virtualization through intermediate and advanced coverage to vSphere clusters, life cycle management, and best practices. The chapters pair the VCP-DCV exam objectives so the study meets certification needs while earning real-world skills.

At the end of this book, readers will have the confidence to pass the VCP-DCV exam and be able to confidently deploy, manage, and enhance VMware environments in organizations. This book allows readers to learn about virtualization and enhance their IT career. I believe readers will find this book informative and helpful.

Chapter 1: Overview and Aim - In this chapter, the structure of the book is outlined, the goals defined, and the immediate alignment to the VCP-DCV exam criteria. It outlines the progression through fundamental principles to advanced vSphere functionalities and provides readers with a clear educational path and outcomes leading to certification.

Chapter 2: Understanding Virtualization and vSphere - Establishes a foundation by examining fundamental virtualization principles and the structural design of VMware vSphere. This exploration allows readers to comprehend the roles of ESXi, vCenter, vSAN, and NSX as essential components of contemporary data centers, facilitating scalability, operational efficiency, and security within virtualized settings.

Chapter 3: Installing and Setting Up ESXi - Guiding readers through the process of installing ESXi on bare-metal servers, the book takes readers through prerequisites to host configurations. Hands-on exercises using the DCUI and the Host Client enable readers to build a secure and optimized base for vSphere infrastructures.

Chapter 4: vCenter Deployment and Configuration - Explains the installation of the **vCenter Server Appliance (vCSA)** and settings of major services. Includes license administration, inventory objects, permissions, and troubleshooting by means of logs and events.

Chapter 5: Networking in vSphere - Explores concepts of virtual networking utilizing standard switches and distributed switches. The reader learns how to configure NIC teaming, load balancing, and VMkernel adapters for vMotion and HA services. Hands-on comparison and design considerations dictate the selection of a proper methodology.

Chapter 6: Managing Storage in vSphere - Explains various storage technologies such as VMFS, NFS, Fibre Channel, and iSCSI. The reader will learn how to set up datastores, multipath policies, and storage solutions that offer maximum performance, resiliency, and scalability in virtual systems.

Chapter 7: Virtual Machine Deployment - Virtual machines explains the construction and provisioning of VMs, VMware Tools functionality and the use of templates, clones, and content libraries. Step-by-step instructions walk the end user through deploying VMs in rapid and highly functional ways.

Chapter 8: Virtual Machines Management - Concerned mostly with daily VM activities such as vMotion, Storage vMotion, snapshots, and resource allocation (shares, reservations, and limits). As readers become experts at VM migrations, they will discover how to optimize CPU and memory performance on real-world platforms.

Chapter 9: vSphere Clusters Management - Get hands-on experience with

the advanced cluster features including vSphere DRS, HA, and fault tolerance. Readers will configure clusters for resiliency, optimize workload deployment, and learn methods to provide network redundancy and admission control and high availability.

Chapter 10: Lifecycle Management - Introduces vSphere Lifecycle Manager for automating upgrades and patch management. Includes using baselines and images, staying in compliance, updating ESXi hosts and VMware Tools and VM hardware to a secure, current state-of-technology infrastructure.

Chapter 11: Virtualization Best Practices - Integrates insights into implementable best practices encompassing installation, networking, storage, security, and resource allocation. Readers will acquire practical approaches for constructing effective, secure, and high-performance vSphere environments in accordance with industry benchmarks.

APPENDIX: Mock Exams and Preparation - Wraps up with specialized examination preparation techniques such as mock questions, examination-day advice, and key objective at-a-glance. Readers will also look at what they can do after certification and beyond VMware advanced certifications, laying the groundwork for the long-term progress.

Coloured Images

Please follow the link to download the
Coloured Images of the book:

<https://rebrand.ly/41b19c>

We have code bundles from our rich catalogue of books and videos available at <https://github.com/bpbpublications>. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

At www.bpbonline.com, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on BPB books and eBooks. You can check our social media handles below:



Instagram



Facebook



Linkedin



YouTube

Get in touch with us at: business@bpbonline.com for more details.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at business@bpbonline.com with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit www.bpbonline.com. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit www.bpbonline.com.

Join our Discord space

Join our Discord workspace for latest updates, offers, tech happenings around the world, new releases, and sessions with the authors:

<https://discord.bpbonline.com>



Table of Contents

1. Overview and Aim

- Introduction
- Structure
- Objectives
- Understanding VMware vSphere
- Importance of data center virtualization
- Certification overview
- Course objectives and learning outcomes
- Conclusion
- Exercises
- References

2. Understanding Virtualization and vSphere

- Introduction
- Structure
- Objectives
- Basic virtualization concepts
 - Brief introduction to virtual machine*
 - Benefits of using virtual machines*
 - Brief introduction to vSphere*
 - Types of virtualizations*
 - vSphere's interaction with hardware*
 - Guest of ESXi host*
 - Comprehending virtual and physical architectures*

- Physical resource sharing*
- Sharing of physical resources*
- CPU virtualization*
- Memory usage of physical and virtualised hosts*
- Physical and virtual networking*
- Physical file systems and datastores*
- Virtual discs and data stores*
- vSphere in the software-defined data center*
- vSphere in the cloud infrastructure*
- User interfaces for vSphere*
 - VMware Host Client*
 - vSphere client*
 - ESXCLI vs. PowerCLI*

Conclusion

Points to remember

Exercises

Lab exercises

3. Installing and Setting Up ESXi

Introduction

Structure

Objectives

Installing an ESXi host

- ESXi installation requirements*

- Interactive ESXi installation*

Configuring an ESXi host

- Configuring ESXi host settings*

- Configuring an ESXi host with management network*

 - Configuring an ESXi host with root access*

 - Configuring an ESXi host with troubleshooting options*

 - Time synchronization for the ESXi host*

Configuring NTP

Configuring PTP

Controlling remote access to an ESXi host

Recognizing ESXi user account best practices

Conclusion

Points to remember

Exercises

Lab exercises

4. vCenter Deployment and Configuration

Introduction

Structure

Objectives

Understanding ESXi hosts communication with vCenter

vCenter services

vCenter architecture

vCenter Single Sign-On

Enhanced Linked Mode

Communication between vCenter and ESXi

vCenter scalability

Deployment of vCenter Server Appliance

Installer for vCenter Server Appliance Native GUI

vCenter Server Appliance installation

Deployment automation

vCenter Server Appliance installation

vCenter Server Appliance installation

Getting started with vCenter

Configuring essential vCenter settings

vCenter Management Interface

Multi-homing the vCSA

Managing license keys with the vSphere Client

vSphere License Service

Adding license keys to vCenter

Viewing licensed features

Organizing vCenter inventory objects

Navigating the inventory

Viewing object information

About data center objects

Sorting inventory objects into folders

Adding a data centre and organizational objects to vCenter

Add ESXi hosts to vCenter

Creating custom tags for inventory objects

Explaining vCenter permissions

Understanding roles in vCenter

Understanding objects in vCenter

Permissions on objects

Assigning permissions in vCenter

Viewing roles and user assignments in vCenter

Understanding permission propagation and combination in vCenter

Creating a role in vCenter

About global permissions

Gaining insights from vCenter logs and events

vCenter log levels

Configuring log levels

Forwarding vCenter log files to a central log server

Forwarding ESXi host log files to a central log server

Conclusion

Points to remember

Exercises

Lab exercises

5. Networking in vSphere

Introduction

Structure

Objectives

Unleash the potential of standard switches

Virtual switch connections and examples

VLANs and virtual switch tagging

Viewing and adding standard switches

VMkernel adapter properties

Enabled services on VMkernel adaptors

Mastering distributed switches

Distributed switches topology

Understanding and configuring discovery protocols

Port binding overview

Inbound traffic shaping

Physical NIC Load balancing

Standard vs. distributed switches

Sculpting network policies with finesse

Security policies

Traffic shaping policies

Outbound traffic shaping

NIC teaming and failover

Load balancing methods for virtual switches

Detecting and handling network failure

Conclusion

Points to remember

Exercises

Lab exercises

6. Managing Storage in vSphere

Introduction

Structure

Objectives

Exploring the landscape of vSphere storage technologies

Improving the performance of storage

vSphere storage device naming conventions

Storage protocol overview in vSphere

Navigating the world of vSphere datastores

Methods to access datastore

Datastore contents

vSphere VMFS overview

Deployment options

Network file system overview

Compatibility considerations

vSAN overview

vSphere Virtual Volumes overview

Key features and capabilities

Raw device mapping overview

Compatibility modes

Considerations for physical storage

Fibre Channel components and addressing

Connectivity overview

Components of Fibre Channel SANs

Access control and Fibre Channel addressing

Fibre Channel multipathing

Disk array configurations

Real-world use

Mastering iSCSI components and addressing

iSCSI addressing

iSCSI adaptors

Configuring the ESXi network for software iSCSI

Activating the software iSCSI adapter

iSCSI target discovery

Target-discovery techniques for iSCSI

iSCSI security, CHAP authentication

Using iSCSI software for multipathing

Using dependent hardware iSCSI for multipathing

Using independent hardware iSCSI for multipathing

Utilizing iSCSI initiator to bind VMkernel ports

VMFS datastores creation and management

Viewing datastore contents

Expanding the VMFS datastores size

Datastore maintenance mode

Deleting or unmounting a VMFS datastore

Configuring storage load balancing and path selection policies

Path selection policies

Visualizing multipathing in vSphere

NFS datastores configuration and administration

NFS datastores configuration

Setup ESXi host authentication and NFS Kerberos credentials

Setting up Kerberos authentication in NFS datastore

Unmounting an NFS datastore

Multipathing and NFS storage

Multipathing configuration for NFS 4.1

VMkernel binding in NFSv3

Conclusion

Points to remember

Exercises

Lab exercises

7. Virtual Machine Deployment

Introduction

Structure

Objectives

Creating and provisioning virtual machines

Provisioning methods

New Virtual Machine wizard overview

Guest operating system installation

OVF templates deployment

Removing virtual machines

VMware Tools

Installing VMware Tools

Downloading VMware Tools

Virtual machines components

Virtual machine encapsulation

Virtual machine files

Understanding virtual machine virtual hardware

Virtual hardware versions and compatibility

Optimizing CPU and memory in VMs

Understanding virtual storage in vSphere

Thick provisioned virtual disks

Thin provisioned virtual disks

Thin provisioned disks datastore management

Understanding virtual networks in vSphere

Virtual network adapters in vSphere

PCI passthrough devices in vSphere

Other virtual devices within vSphere

Navigating vSphere client for VM management

Understanding the VM Console

Accessing the VM Console

Common use cases for the VM Console

Virtual machines resources optimization

Hot-pluggable devices overview

Increasing virtual disk size dynamically

Inflating thin-provisioned disks

General settings in VM Options

VMware tools settings in VM Options

Boot settings in VM Options

Harnessing the efficiency of templates

Creating a template by cloning a VM

Creating a template by converting a VM

Updating templates

Deploying VMs from a template

Cloning virtual machines

Guest operating system customization

Customization specifications

Customizing the guest operating system

Content libraries for VM resources

Advantages of content libraries

Types of content libraries

Interface of content library

Creating a local content library

Filling the content library with templates

Adding VM or OVF templates into content library

Adding OVF templates into a content library

Viewing content library items

Deploying VMs from a content library

Content library integration

Publishing a content library

Subscribing to a content library

Viewing content libraries

Viewing subscribed content library templates

Publishing a subscription to a shared VM templates

Synchronizing libraries with or without Enhanced Linked Mode

Simple versioning in content libraries

Advanced configuration

Content library maximums

Managing VM template versions

Template versioning process overview

Viewing template versions

Deleting and reverting to template versions

Conclusion

Points to remember

Exercises

Lab exercises

8. Virtual Machines Management

Introduction

Structure

Objectives

VM migrations types and strategies

VM migration options

Understanding vSphere vMotion

Configuring vSphere vMotion networks

Best practices for vSphere vMotion networking

vSphere vMotion migration workflow

VM requirements for vSphere vMotion migration

Host requirements for vSphere vMotion migration

Performing a vSphere vMotion migration

Migration errors validation

Migrating encrypted VMs

Migration compatibility with EVC

Understanding EVC

EVC clusters requirements for CPU mode

Setting the EVC CPU mode on an existing cluster

Changing the EVC CPU mode for a cluster

Virtual machine EVC CPU mode

EVC for vSGA GPUs

EVC cluster requirements for graphics mode

Configuring EVC graphics mode on an existing cluster

Virtual machine EVC graphics mode

VM migration with vSphere Storage vMotion

vSphere Storage vMotion deployment

Identifying storage arrays that support VAAI

vSphere Storage vMotion guidelines and limitations

Changing compute resource and storage during migration

Use case for moving both compute resource and storage

VM migration across vCenter

Cross vCenter migration requirements

Executing cross vCenter vMotion in same SSO domain

Performing cross vCenter vMotion in different SSO domain

Network compatibility checks during cross vCenter migrations

Understanding VMkernel networking layer and TCP/IP stacks

Best practices for network security and performance

vSphere vMotion TCP/IP stack

Understanding Long Distance vSphere vMotion migration

Networking requirements for Long Distance vSphere vMotion

Snapshot management

Taking snapshots

Types of snapshots

VM snapshot files

Managing snapshots

VM snapshot deletion scenarios

Understanding snapshot consolidation

CPU and memory concepts and considerations

VM memory overcommitment

Memory overcommits techniques

Configuring multicore VMs

Understanding hyperthreading

CPU load balancing in ESXi

VM resource allocation

RAM reservations for VMs

CPU reservations for VMs

Controlling resource utilization

Resource allocation shares

Configuring resource allocation settings for a VM

Conclusion

Points to remember

Exercises

Lab exercises

9. vSphere Clusters Management

Introduction

Structure

Objectives

vSphere cluster insights

Creating a vSphere cluster

Cluster Quickstart overview

Cluster Quickstart for activating services

Cluster Quickstart for adding hosts

Cluster Quickstart for configuring the cluster

Setting up a cluster for distributed switches

Setting up a cluster for vSAN and vMotion traffic

Setting up a cluster for advanced features

Cluster summary information

Observing cluster resources

vSphere Cluster Services VMs

vSphere DRS overview

vSphere DRS with a VM-centric approach

Viewing VM DRS scores in the monitor tab

Requirements for a vSphere DRS cluster

vSphere DRS settings for automation levels

vSphere DRS settings for migration threshold

vSphere DRS settings for predictive DRS

Checking vSphere DRS settings

Configurations of vSphere DRS for VMs automation

vSphere DRS settings for VM swap file location

vSphere DRS configurations for affinity of VMs

vSphere DRS configurations for DRS groups

vSphere DRS configurations for VM-Host affinity rules

Viewing vSphere DRS recommendations

Maintenance mode vs. standby mode

vSphere HA overview

vSphere HA scenario for ESXi host failure

Importance of heartbeat networks in vSphere HA

vSphere HA scenario for protecting VMs against network isolation

Heartbeat network redundancy using NIC teaming

Heartbeat network redundancy using additional networks

vSphere HA designing and configuration

vSphere HA designing for network heartbeats

vSphere HA designing for datastore heartbeats

vSphere HA failure scenarios

Failed secondary hosts

Failed primary hosts

Isolated hosts

VM storage failures

Protecting against storage failures with VMCP

Additional vSphere HA design approaches

- vSphere HA configuration requirements*
- Configuring vSphere HA settings*
- vSphere HA settings: failures and responses*
- vSphere HA settings for default VM restart priority*
- About vSphere HA orchestrated restart*
- Configuring orchestrated restart*
- vSphere HA settings for VM monitoring*
- vSphere HA settings for admission control*
- Admission control using cluster resource percentage*
- Admission control using slots*
- vSphere HA settings for performance degradation VM toleration*
- vSphere HA heartbeat datastore settings*
- vSphere HA advanced options settings*
- Network configuration and maintenance*
- Monitoring vSphere HA cluster status*
- Using vSphere HA with vSphere DRS*

Embracing vSphere Fault Tolerance

- vSphere Fault Tolerance with vSphere HA and DRS*

- Configuring vSphere FT on a VM*

Conclusion

Points to remember

Exercises

Lab exercises

10. Lifecycle Management

Introduction

Structure

Objectives

vCenter lifecycle management

- Preparing the interoperability report*

- Upgrading and patching vCenter*

- Overview of upgrading vSphere*
- Exploring vSphere lifecycle manager
 - Understanding ESXi images*
 - About image depot*
 - Importing content to image depot from online sources*
 - Specifying the online source for download*
 - Importing content to image depot from offline sources*
- ESXi host and cluster lifecycle management
 - Managing clusters with vSphere lifecycle manager*
 - Verification of host compliance with a cluster image*
 - Performing a remediation pre-check*
 - Staging the cluster*
 - Remediation of a cluster against an image*
 - Reviewing remediation impact*
 - Parallel remediation*
 - Recommended images in vSphere lifecycle manager*
 - Viewing recommended images in vSphere lifecycle manager*
 - Customizing cluster images using vSphere lifecycle manager*
 - Managing vSphere configuration profiles*
 - Configuration documentation*
 - Using vSphere configuration profiles*
 - vSphere lifecycle manager for standalone hosts*
- Managing VMware tools and virtual hardware
 - Keeping VM hardware up to date*
- Conclusion
- Points to remember
- Exercises
- Lab exercises

11. Virtualization Best Practices

- Introduction

Structure

Objectives

Applying best practices from installation to management

Real-world tip

Networking best practices in vSphere environment

Real-world tip

Storage best practices in vSphere environment

Real-world tip

Virtual machine best practices

Real-world tip

CPU and memory best practices

Real-world tip

Best practices for vSphere clusters management

Real-world tip

Best practices for lifecycle management

Real-world tip

Security best practices in vSphere environment

Real-world tip

Best practice for a healthy vSphere environment

Real-world tip

Best practices for vSphere HA and Fault Tolerance

Real-world tip

Conclusion

APPENDIX: Mock Exams and Preparation

Introduction

Practice questions

Answers

Index

CHAPTER 1

Overview and Aim

Introduction

This chapter introduces the overall structure and key objectives of this book.

This book is designed to equip the readers with a clear roadmap of what will be discussed throughout the book, guiding you toward achieving the *VMware Certified Professional - Data Center Virtualization (VCP-DCV)* certification. Whether you are new to VMware or seeking to expand your existing knowledge, this book will help you with the skills needed to manage and optimize virtual environments in the VMware ecosystem.

Note: VMware is now part of Broadcom and is known as VMware by Broadcom. All references to VMware in this book reflect this change.

Structure

In this chapter, we will cover the following topics:

- Understanding VMware vSphere
- Importance of data center virtualization
- Certification overview (VCP-DCV)
- Course objectives and learning outcomes

Objectives

By the end of this chapter, the readers would recognize how the course is organized and how each module advances our knowledge of VMware vSphere and data center virtualization. We would also understand the fundamental elements and function of VMware vSphere, which lay the groundwork for data center virtualization, and acknowledge the advantages of virtualization in modern IT infrastructure, such as cost-effectiveness, scalability, and resource optimization.

Additionally, we will understand the main competencies and subjects included in the VCP-DCV certification test. We would also be able to determine the knowledge and abilities that readers would acquire from this course and how they relate to certification criteria and industry standards.

Understanding VMware vSphere

With VMware vSphere, the industry's top virtualization platform, businesses can use virtualization to simplify and improve their IT infrastructure. By combining workloads from several servers onto a smaller number of actual hardware resources, vSphere enables companies to increase productivity, scale as needed, and drastically save operating expenses. A variety of parts make up the vSphere ecosystem, which collectively offers a virtual infrastructure that is adaptable, safe, and extremely managed.

Let us examine the following fundamental components:

- **ESXi hosts:** The hypervisor of VMware vSphere is called ESXi. Multiple **virtual machines (VMs)** can operate independently on a single host thanks to ESXi, which runs directly on bare-metal hardware and abstracts the real hardware into virtual resources. ESXi allows for flexible resource allocation, secure virtual machine isolation and effective resource utilization.
- **vCenter Server:** For the management of numerous ESXi hosts and the virtual machines that operate on them, vCenter Server is an essential centralized management solution. A single interface for automating, configuring, and monitoring the vSphere environment is offered by

vCenter Server. Administrators may simply manage resource pools, clusters, and virtual machine templates with vCenter. They can also carry out necessary operations like load balancing across hosts and live VM migrations.

- **vSAN:** An integrated, software-defined storage system called vSAN creates a high-performance, scalable, and resilient storage layer inside the vSphere environment by pooling local storage from ESXi hosts into a shared data store. vSAN is perfect for supporting cloud-native apps and virtualized workloads since it reduces reliance on conventional storage infrastructure, enhances storage performance, and streamlines storage administration.
- **NSX:** NSX extends virtualization to the network layer by offering **software-defined networking (SDN)** capabilities. In the vSphere environment, this platform makes network abstraction, automation, and improved security possible. Without the need for actual hardware, NSX enables the development of intricate network topologies, micro-segmentation for security, and sophisticated routing and load-balancing choices. Additionally, it simplifies network administration, which facilitates the deployment and security of apps across clouds and data centers.

These elements work together to provide a unified and strong virtualization platform that helps businesses efficiently install, manage, and safeguard their IT assets. Because of its versatility, vSphere is a crucial component of contemporary data centers, enabling a wide range of use cases from containerized workloads and next-generation cloud to classic business applications.

Importance of data center virtualization

The foundation of contemporary IT infrastructures is data center virtualization, which turns physical gear into virtual assets to help businesses optimize and expedite resource management. Organizations can get several important advantages by virtualizing their physical resources:

- **Increased resource utilization:** By enabling the execution of numerous

workloads on a single physical server, VMs greatly increase hardware utilization. By ensuring that hardware is utilized to its maximum capacity, this consolidation enables businesses to get the most out of their physical resource investments.

- **Simplified management:** Virtualization provides a centralized command over networking, storage, and processing power. IT teams may streamline operations and lower administrative complexity by monitoring, configuring, and managing several VMs, storage volumes, and network connections from a single platform using a single management interface.
- **Flexibility and scalability:** Virtualized environments enable IT teams to modify resources in response to demand swiftly. Organizations may more readily adapt to shifting business needs thanks to this scalability, which allows them to add or remove VMs or change resource allocations without requiring extra physical hardware.
- **Cost savings:** By eliminating the requirement for a significant number of physical servers, virtualization reduces operating and capital costs. Having fewer physical equipment results in cheaper energy expenses, simpler maintenance, and lower hardware purchase costs, all of which help businesses save a lot of money.

As data center virtualization offers efficiency, flexibility, and cost-effectiveness that help businesses remain competitive in a digital world, it has become indispensable for contemporary IT settings.

Certification overview

The *VCP-DCV* certification is designed to validate your expertise in deploying, managing, and optimizing VMware vSphere environments.

There is a dedicated chapter towards the end where we will cover:

- Certification prerequisites and requirements.
- The structure of the certification exam, including key topics such as:
 - vSphere architecture
 - Storage and network configuration

- Virtual machine management
- Resource management (e.g., vMotion, DRS)

VMware badges are digital representations of skills and accomplishments. Digital badges include the following features:

- Easy to share on social media (LinkedIn, Twitter, Facebook, blogs, etc.).
- Connected to VMware to validate and verify achievements.

For the complete list of certifications and details about how to attain these certifications, see

<https://www.broadcom.com/support/education/software/certification>.

Course objectives and learning outcomes

This book is designed with the following objectives in mind:

- Building a solid foundation of VMware vSphere and its architecture.
- Gaining the expertise to design, implement, and manage VMware virtual environments.

By the end of this book, the reader should be able to:

- Understand the significance of VMware vSphere and its role as a leading virtualization platform.
- Install and configure VMware ESXi hosts to build a robust virtualization infrastructure.
- Master deploy and configure vCenter Server Appliance for centralized management of virtual environments.
- Create and configure virtual networks using vSphere Standard Switches for optimized networking.
- Understand and implement storage technologies like iSCSI, NFS, and VMFS in vSphere.
- Configure and manage datastores.
- Deploy, manage, and optimize virtual machines using templates, clones, and snapshots.
- Set up and manage vSphere clusters with HA and DRS.
- Migrate virtual machines with vSphere vMotion and Storage.

- Use **vSphere Lifecycle Manager (vLCM)** to perform ESXi host and VM upgrades.
- Apply VMware best practices for fine-tuning, performance, security, and scalability in virtual environments.
- Prepare thoroughly for the VCP-DCV certification, mastering the skills needed for success.

We will also discuss the key learning objectives that map directly to the VCP-DCV exam objectives.

Conclusion

This chapter outlined the book's main goals, structure, and expectations. The fundamental ideas of virtualization and the vital role VMware vSphere plays in updating data centers have been outlined in this chapter.

By converting physical resources into adaptable, scalable, and effective virtual assets, virtualization helps businesses cut expenses, combine tasks, and maximize resource utilization. These efforts are supported by VMware vSphere, the industry-leading platform that offers a solid basis for managing storage, network configurations, and VMs.

In the next chapter, we will understand the architecture, components, and basic operations of VMware vSphere. We will also examine the key elements enabling us to get the most out of this potent platform.

Exercises

1. What is the primary benefit of data center virtualization?
2. Explain the significance of VMware vSphere in IT infrastructure.
3. List three to five essential skills you will gain from this book.
4. How would you apply the benefits of data center virtualization to optimize an organization's infrastructure?

References

To enhance your understanding further, explore VMware's offerings. The following are some useful resources:

- *VMware Documentation for vSphere*: <https://techdocs.broadcom.com/>
- *VMware Communities*: <https://community.broadcom.com/home>
- *VMware Support and Education*: <https://support.broadcom.com/>
- *VMware Hands-on Labs*: <http://hol.vmware.com>

Join our Discord space

Join our Discord workspace for latest updates, offers, tech happenings around the world, new releases, and sessions with the authors:

<https://discord.bpbonline.com>



CHAPTER 2

Understanding Virtualization and vSphere

Introduction

In this engaging chapter, we will discuss the fundamental ideas that form the basis of the virtualization industry and how VMware vSphere blends in with the ever-changing software-defined data center and cloud infrastructure environments. Gain a thorough understanding of the fundamentals of virtualization and learn how they serve as the cornerstone of contemporary IT infrastructures.

As we progress, readers will discover how to use the several user interfaces that provide access to vSphere, guaranteeing that the reader can work with this powerful technology with assurance. We will also untie the complex interactions that vSphere has with key parts like CPUs, memory, networks, storage, and GPUs. You will have a firm grasp of virtualization and an understanding of how vSphere transforms cloud and data center management by the end of this chapter.

Note: VMware is now part of Broadcom and is known as 'VMware by Broadcom.' All references to 'VMware' in this book reflect this change.

The components that vSphere is built upon must be known to the reader as a vSphere administrator. Additionally, readers need to comprehend the

following ideas:

- VMs, virtualization, and the function of the ESXi hypervisor
- Essential vSphere elements and vSphere's application in software-defined data centers
- vSphere environments are managed and administered by vSphere clients

Structure

In this chapter, we will cover the following topics:

- Basic virtualization concepts
- vSphere's interaction with hardware
- vSphere in the software-defined data center
- vSphere in the cloud infrastructure
- User interfaces for vSphere

Objectives

After finishing this chapter, the reader should be able to explain the core ideas of virtualization and have a sound foundation for understanding VMware vSphere. Reader will get insight into proactive vSphere management practices, which are critical for optimizing and maintaining a virtualized infrastructure. This chapter will also teach reader how vSphere works with major hardware components such as CPUs, memory, networks, and storage to properly manage resources in a virtualized data center. In addition, reader will investigate the use of GPUs in virtualization, working alongside vSphere Bitfusion to provide advanced computational workloads. Reader will also learn about VMware vSphere+, a subscription-based solution that allows on-premises workloads to take advantage of cloud benefits, resulting in more flexibility and resource management.

Finally, the reader will become familiar with the numerous user interfaces used to access and manage ESXi hosts as well as the vCenter Server system, giving the reader the tools the reader needs to confidently traverse and handle the vSphere environment.

Basic virtualization concepts

The key VMware vSphere terminologies and definitions are as follows:

Terminology	Definition	Example
Operating system	Software designed to allocate and manage physical resources for applications.	Microsoft Windows, Linux
Application	Software that runs on an operating system, utilizing system resources.	Microsoft Office, Chrome
Virtual machine	Specialized software-based instance that mimics physical hardware functionality.	
Guest	Operating system running within a VM, also known as the guest operating system.	Microsoft Windows, Linux
Hypervisor	A hypervisor is specialized software that creates and manages virtual machines by abstracting physical hardware resources and allocating them to multiple VMs running simultaneously on a single physical host.	Types of hypervisors: <ul style="list-style-type: none">• Type 1 (Bare-metal/Native): Runs directly on the physical hardware, like VMware ESXi or Microsoft Hyper-V.• Type 2 (Hosted): Runs on top of a host operating system, like Oracle VirtualBox or VMware Workstation.
Host	Physical machine that provides resources to run virtual machines via the ESXi hypervisor.	Dell PowerEdge Server, HPE ProLiant blade server, Cisco Rack or Blade Server
vSphere	VMware's virtualization platform, combining ESXi hypervisor and vCenter Server for centralized management.	
Cluster	Group of ESXi hosts with shared resources used collectively by virtual machines.	
vSphere vMotion	Feature that enables the live migration of powered-on VMs from one host to another without downtime.	
vSphere vMotion	Feature that enables the live migration of powered-on VM's storage from one datastore to another without downtime.	

vSphere high availability (HA)	Cluster feature that restarts VMs on other hosts when hardware failures occur to maintain availability.	
vSphere Distributed Resource Scheduler (DRS)	Cluster feature that distributes VMs across hosts for optimized resource allocation using vMotion.	

Table 2.1: vSphere terminologies

Brief introduction to virtual machine

A VM is a software-based representation of a physical computer and its components. Virtualization software creates an abstraction layer between the physical hardware and virtual machines, allowing multiple VMs to share and run on a single host system.

Each VM consists of the following key components:

- **Guest operating system:** The OS (Windows, Linux, etc.) that runs within the VM.
- **Virtual resources**, such as:
 - CPU and memory
 - Network adapters
 - Disks and storage controllers
 - Parallel and serial ports

Every VM has virtual devices that function like physical hardware but are more portable, secure, and easier to manage. Typically, a VM includes an operating system, applications, and the virtual hardware needed to function just like a physical computer.

Guest OS enhancement: VMware Tools is an optional suite of drivers and utilities that can be installed within the guest OS to enhance performance and enable better integration with the virtual environment. While not required for VM operation, VMware Tools improve VM performance by providing drivers that allow the guest OS to communicate efficiently with the virtual hardware, enabling ESXi to manage the VM's use of the host's physical resources more effectively.

Benefits of using virtual machines

The following are some of the benefits of using virtual machines as compared to physical machines:

- **Physical machines:**

- Hard to move or copy, as they are tied to specific hardware.
- Bound to a particular set of hardware components, making upgrades challenging.
- Typically have a limited life cycle.
- Require manual upgrades and often hands-on support for hardware changes.

- **Virtual machines:**

- Easy to move or copy, as they are encapsulated into files.
- Independent of specific physical hardware, allowing flexibility.
- Isolated from other VMs running on the same physical hardware.
- Protected from physical hardware changes, so upgrades do not impact the VM directly.

The following figure illustrates the differences between physical and virtual server:

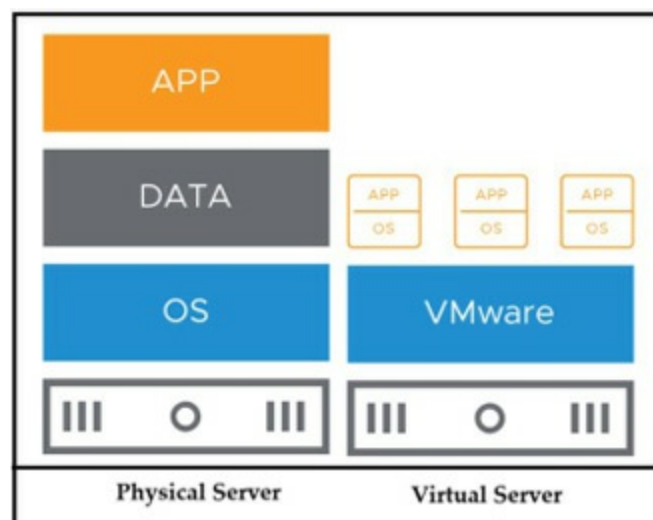


Figure 2.1: Physical vs. virtual server

(Source: VMware)

In a physical setup, the operating system (like Windows or Linux) installs

directly on the hardware and depends on specific drivers to work with particular devices. When new hardware is added, new drivers are often needed. If applications rely directly on these hardware drivers, any hardware or driver upgrade can cause compatibility issues. These challenges mean that IT teams must spend time and money testing hardware upgrades with different operating systems and applications.

By virtualizing, these concerns are reduced. VMs are entirely software-based, meaning you can run multiple VMs (such as a database server and an email server) on the same physical machine without worrying about software dependency conflicts. VMs are isolated from one another, so even if one VM experiences a failure, the others continue to operate smoothly. This isolation also enhances security since users with admin rights on one VM's OS can not access other VMs unless specifically granted permission by the ESXi system admin.

With VMs, server consolidation becomes easier, making hardware use more efficient. Since VMs are just files, you gain access to features that might be limited or unavailable on physical machines:

- **Fast, consistent deployment:** VMs can be provisioned quickly and consistently.
- **High availability and resilience:** Live migration, fault tolerance, high availability, and disaster recovery scenarios are available to help minimize downtime and speed up recovery.
- **Flexible configurations:** You can support multitenancy, such as placing VMs into specialized configurations like a DMZ.
- **Legacy support:** VMs allow you to run older applications and operating systems on modern hardware, even when the original hardware's maintenance has expired.

By using VMs, you streamline infrastructure management, reduce costs, and gain flexibility in handling your workloads.

Brief introduction to vSphere

The virtualization platform vSphere comes with two essential administrative components for managing virtual machines:

- **ESXi:** The hypervisor used to run virtual machines is called ESXi.

- **vCenter:** A platform for central management of networking, storage, virtual machines, and ESXi hosts.

The following figure illustrates the foundation of a vSphere infrastructure:

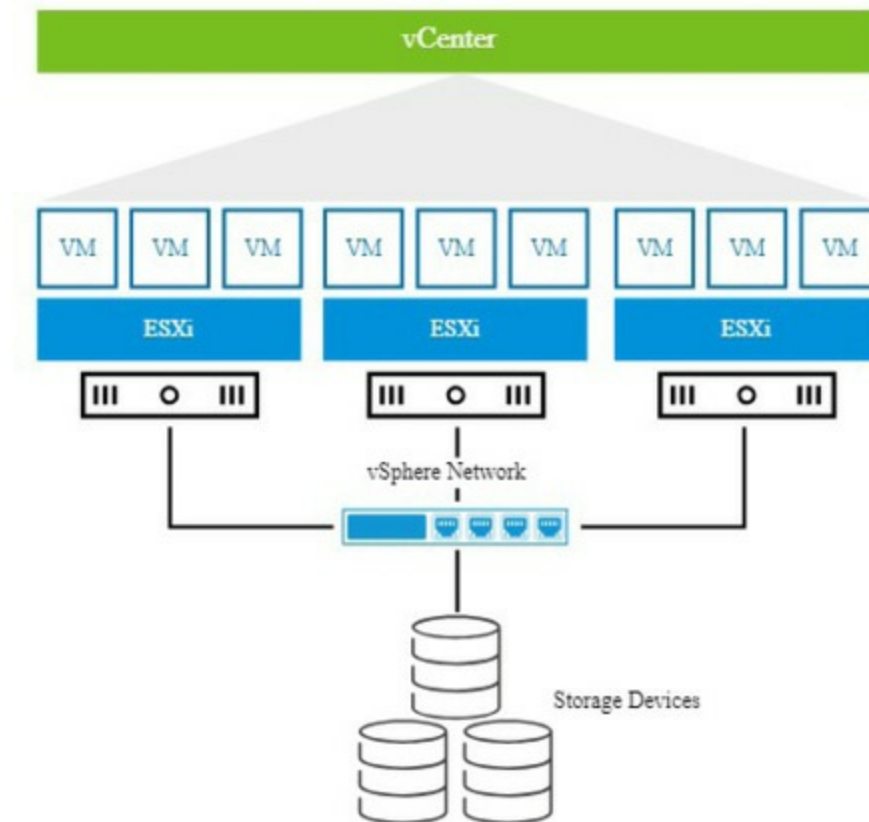


Figure 2.2: vSphere Foundation

(Source: VMware)

The foundation of a vSphere infrastructure is ESXi and vCenter Server.

A virtualization platform called vSphere offers features for resource optimization, application availability, virtualization, management, and operational automation. vSphere gives the data center access to pools of virtual resources by virtualizing and combining the underlying physical hardware resources across several computers.

Furthermore, vSphere offers a collection of dispersed services that facilitate high availability, scalability, and policy-driven resource distribution throughout the virtual data center.

Types of virtualizations

Virtualization involves creating a software-based version of a physical resource, such as a server, desktop, network, or storage device. It is one of the best ways for businesses of all sizes to reduce IT costs while increasing efficiency and flexibility.

The following figure illustrates the types of virtualizations:

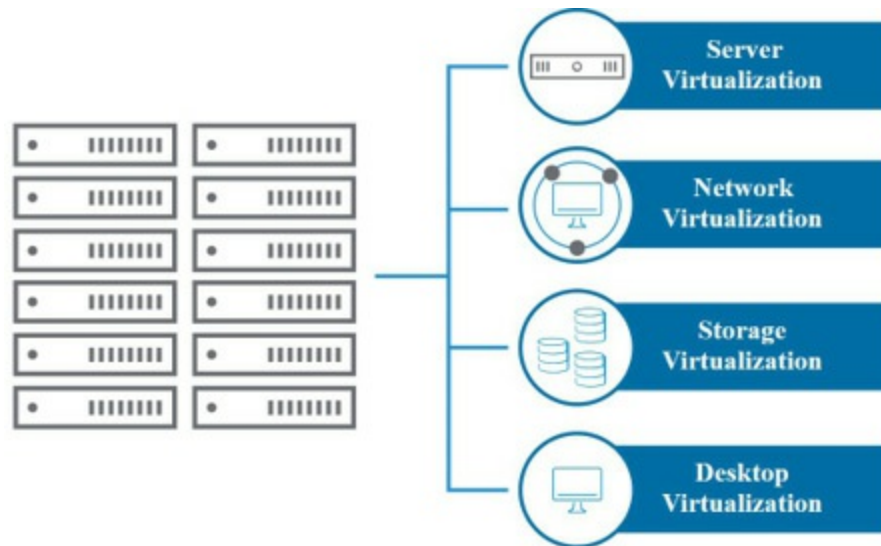


Figure 2.3: Types of virtualizations

(Source: VMware)

The following are some key types of virtualizations:

- **Server virtualization:** This type allows multiple operating systems to run on a single physical server. Each VM gets access to the server's computing resources, helping to make better use of hardware and reduce waste.
- **Network virtualization:** Network virtualization creates a complete software-based copy of a physical network. Applications run on this virtual network just like they would on a real, physical network, but with the added flexibility of software management.
- **Storage virtualization:** Here, software combines multiple network storage devices to appear as one single storage unit. This makes managing storage more straightforward and flexible.
- **Desktop virtualization:** By providing desktops as a managed service, businesses can quickly adapt to changing needs and support remote

work, making it easier to respond to new opportunities.

vSphere's interaction with hardware

To administer and distribute resources to VMs, vSphere communicates directly with physical hardware. Through the virtualization of hardware components, vSphere offers a layer that maximizes speed and flexibility by enabling numerous VMs to share and utilize the physical host's CPU, memory, network, and storage resources.

Guest of ESXi host

A VM is a software-based model of a real computer that uses an ESXi host's CPU, memory, disc, and network resources. It enables programs to use the physical resources of the host while running on compatible operating systems in a *guest* environment. The VMware Compatibility Guide contains a comprehensive list of supported OS systems.

VMware Compatibility Guide can be accessed from this link:

<https://www.vmware.com/resources/compatibility/search.php>

The following figure illustrates the guest OS on ESXi host:

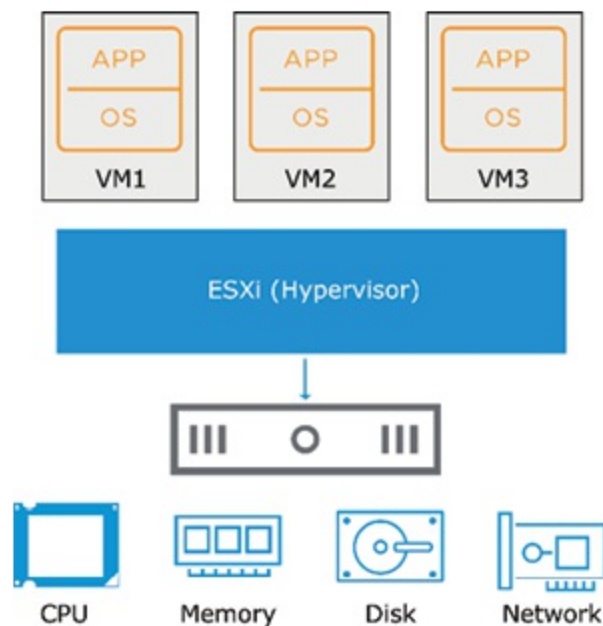


Figure 2.4: Guest OS on ESXi host

(Source: VMware)

Comprehending virtual and physical architectures

Multiple workloads can operate as VMs on a single physical machine thanks to virtualization technology, which builds a bridge between physical and virtual components. This technology is very useful for resolving IT issues because it makes resource sharing possible and maximizes device utilization. The following figure illustrates the differences between physical and virtual architecture:

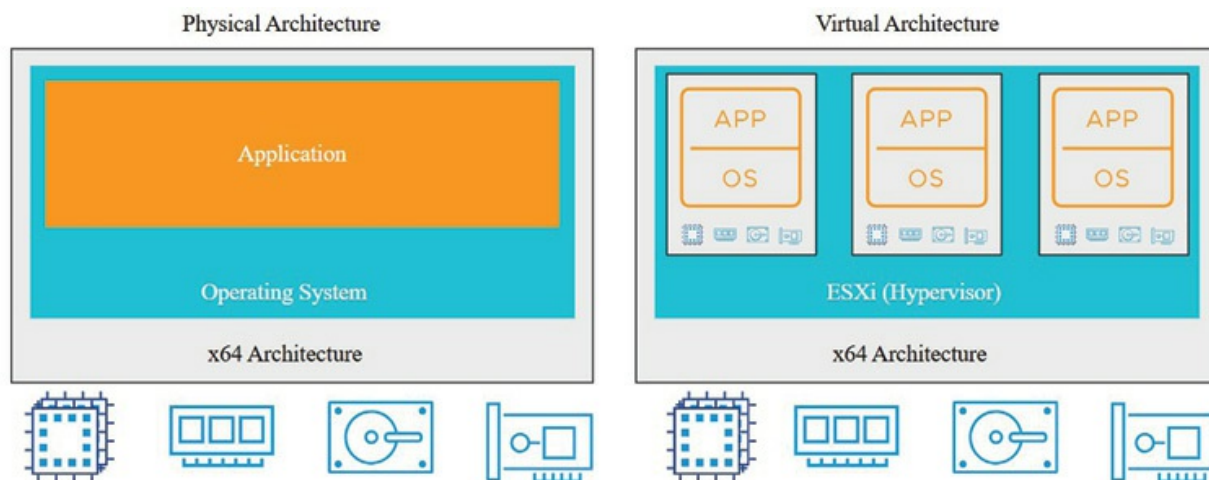


Figure 2.5: Physical vs. virtual architecture

(Source: VMware)

The operating system works directly with the hardware in conventional (non-virtualized) configurations, managing functions including memory allocation, process scheduling, network communication, and data storage. On the other hand, a *virtualization layer* or hypervisor controls the hardware in a virtualized environment. By dynamically allocating physical resources to VMs, this hypervisor allows them to function independently of the underlying hardware.

For example, it is simple to move a VM from one physical host to another or to move its storage without affecting its operation. In a virtualized system, this resource management flexibility enables improved scalability, maintenance, and disaster recovery choices.

Physical resource sharing

In a virtualized environment, multiple VMs operate on a single physical host,

sharing its CPU, memory, network, and storage resources. With virtualization, a single physical server can run multiple VMs, each with access to shared hardware resources, allowing for multiple environments on one physical system.

The following figure illustrates the physical resource sharing within virtual environment:

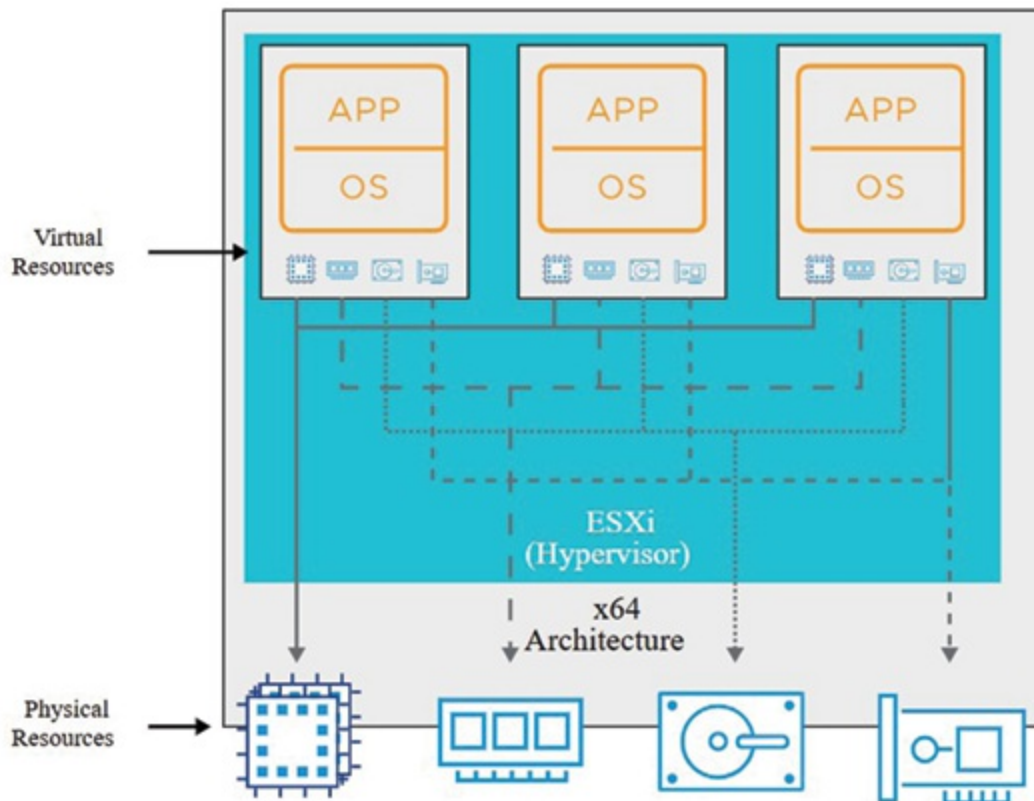


Figure 2.6: Physical resource sharing

(Source: VMware)

Each VM receives an assigned memory segment and shares CPU time, network cards, and storage controllers with other VMs. Through virtualization, different VMs on the same host can run different operating systems and applications. The hypervisor (ESXi) manages this allocation by scheduling VMs to use CPU time and allocating memory, similar to how an operating system manages applications. Additionally, the ESXi hypervisor can overcommit memory, meaning VMs can be allocated more virtual RAM than the physical RAM available on the host.

Each VM uses network and disk bandwidth, which is managed by the hypervisor to balance access fairly among VMs. By default, all VMs on the same ESXi host are given an equal share of resources, though advanced settings allow customization if certain VMs need more priority.

Sharing of physical resources

Several VMs share a single physical host's CPU, memory, network, and storage resources when operating in a virtualized environment. Multiple environments can be run on a single physical server thanks to virtualization, which also gives each VM access to shared hardware resources.

A memory segment is allocated to each VM, and they share storage controllers, network cards, and CPU time. Different VMs on the same host can execute distinct operating systems and applications thanks to virtualization. In the same way that an operating system controls programs, the hypervisor (ESXi) controls this allocation by allocating memory and scheduling virtual machines to use CPU time.

Furthermore, the ESXi hypervisor has the ability to overcommit memory, which allows virtual machines to be assigned more virtual RAM than the host's actual RAM.

The hypervisor controls the network and disc bandwidth used by each VM to equitably distribute access among them. Although advanced options allow customization if specific VMs require additional priority, by default, all virtual machines on the same ESXi host are assigned an equal proportion of resources.

CPU virtualization

In a physical setup, the operating system manages all the available CPUs directly. However, in a virtual environment, CPU virtualization allows each VM to run as though it has its own dedicated processors, maximizing performance by running directly on the physical CPUs.

The following figure illustrates the CPU virtualization:

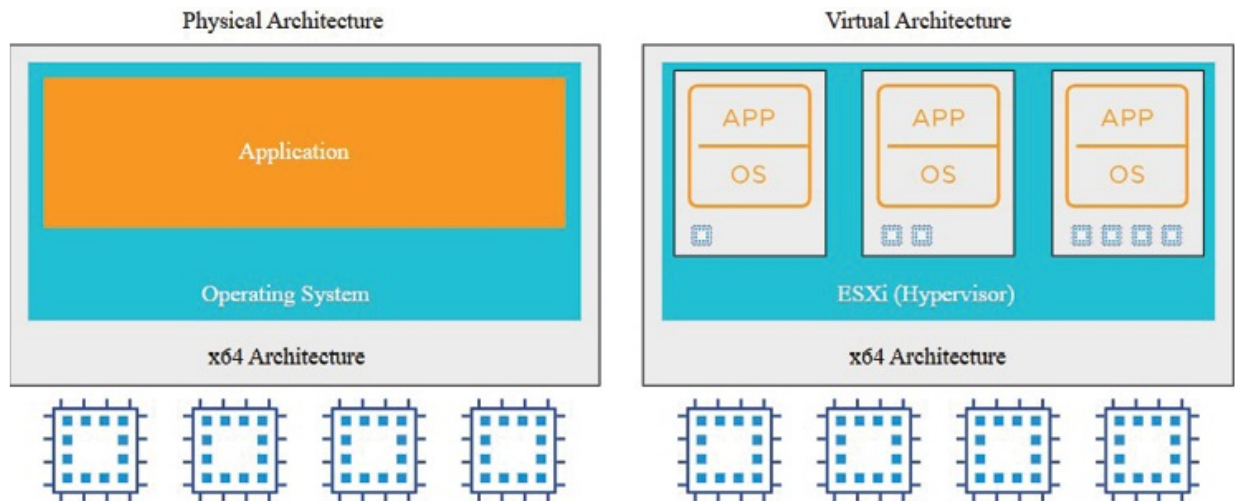


Figure 2.7: CPU virtualization

(Source: VMware)

Unlike software emulation, where programs are translated to run on different types of hardware, CPU virtualization runs on the native x64 processors, providing efficient performance without translation overhead. Emulation can enable portability across systems but often reduces speed, whereas CPU virtualization uses the host's physical CPUs, allowing supported operating systems to run natively.

When multiple VMs are active on an ESXi host, they may sometimes compete for CPU resources. To manage this, the ESXi hypervisor uses time slicing, distributing CPU time across VMs to ensure each VM operates as if it has a dedicated number of virtual processors.

Memory usage of physical and virtualised hosts

All of the system's memory is directly managed by the operating system in a physical system. By establishing a single memory space for every VM without requiring direct access to the physical memory, virtualized environments improve memory management.

The following figure illustrates the memory virtualization:

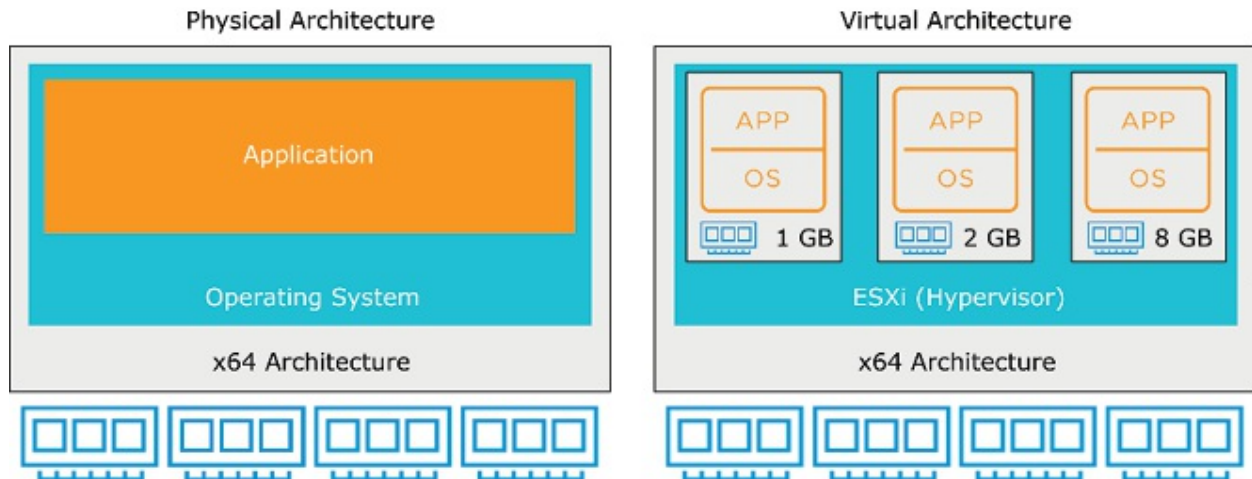


Figure 2.8: Memory virtualization

(Source: VMware)

By mapping out a consistent virtual address space and translating it into physical memory addresses, virtual memory is a popular technique that enables applications to use more memory than is physically available. This is supported by specialized hardware in modern processors, which makes memory swapping, process isolation, and address translation effectively.

Like virtual memory in physical systems, VMware's hypervisor allocates a specific amount of memory to each VM during startup. Multiple VMs can operate on the same host thanks to this configuration, which isolates each VM's memory to stop outside intervention.

Physical and virtual networking

Virtual networking in vSphere relies on virtual Ethernet adapters and virtual switches to manage communication between VMs and external networks.

The following figure illustrates the network virtualization:

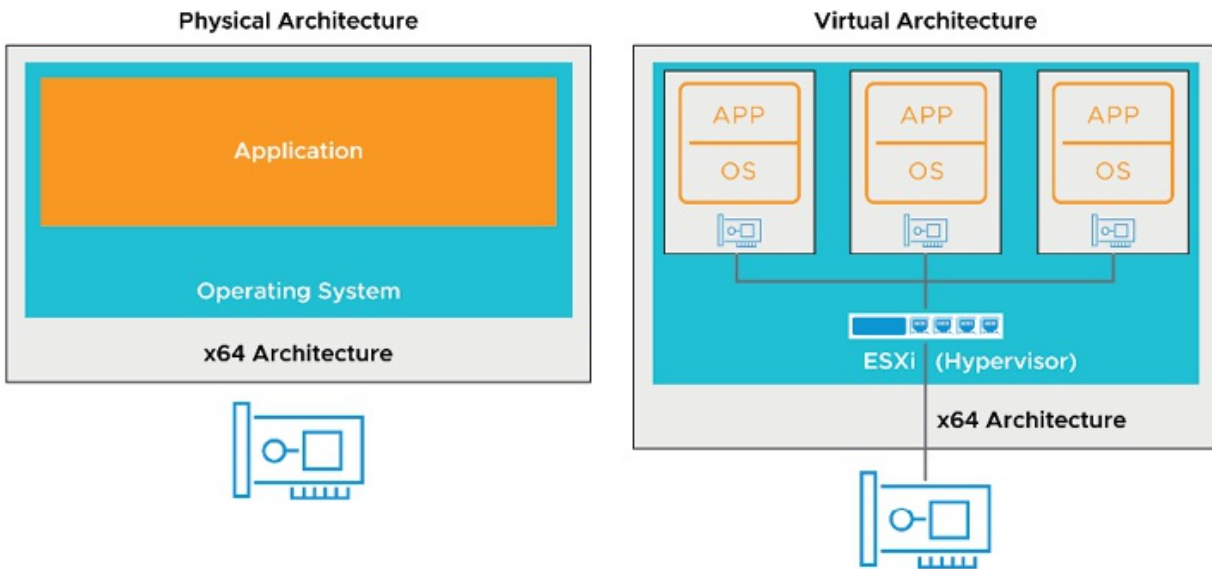


Figure 2.9: Network virtualization

(Source: VMware)

Each VM can be equipped with one or more virtual Ethernet adapters. When VMs need to communicate on the same ESXi host, they can use virtual switches, which mimic physical switches, allowing them to connect using standard network protocols without extra hardware. Virtual switches also support VLANs, making them compatible with VLAN configurations from other networking vendors.

Virtual switches in VMware function similarly to physical Ethernet switches, forwarding network frames on the data link layer. ESXi hosts can have multiple virtual switches, each connected to the physical network via outbound adapters known as vmnics. By teaming multiple vmnics together, virtual switches increase both the network's availability and the bandwidth available to VMs.

In terms of security, each virtual switch is isolated, maintaining its own forwarding table so that data can only be sent to ports on the same switch. VLAN segmentation is also supported, enabling ports to be set as access or trunk ports to accommodate single or multiple VLANs.

Unlike physical switches, virtual switches do not require the **Spanning Tree Protocol (STP)** due to their single-tier structure; traffic between virtual switches on the same host is not possible. This design prevents potential network loops, ensuring that virtual switches remain isolated, stable, and

secure.

Physical file systems and datastores

Shared Storage and vSphere VMFS - VMware Multiple ESXi hosts can access shared storage at once thanks to vSphere's high-performance storage architecture, the **Virtual Machine File System (VMFS)**. VMs can function effectively and use storage resources as needed thanks to shared storage.

The following figure illustrates the storage virtualization:

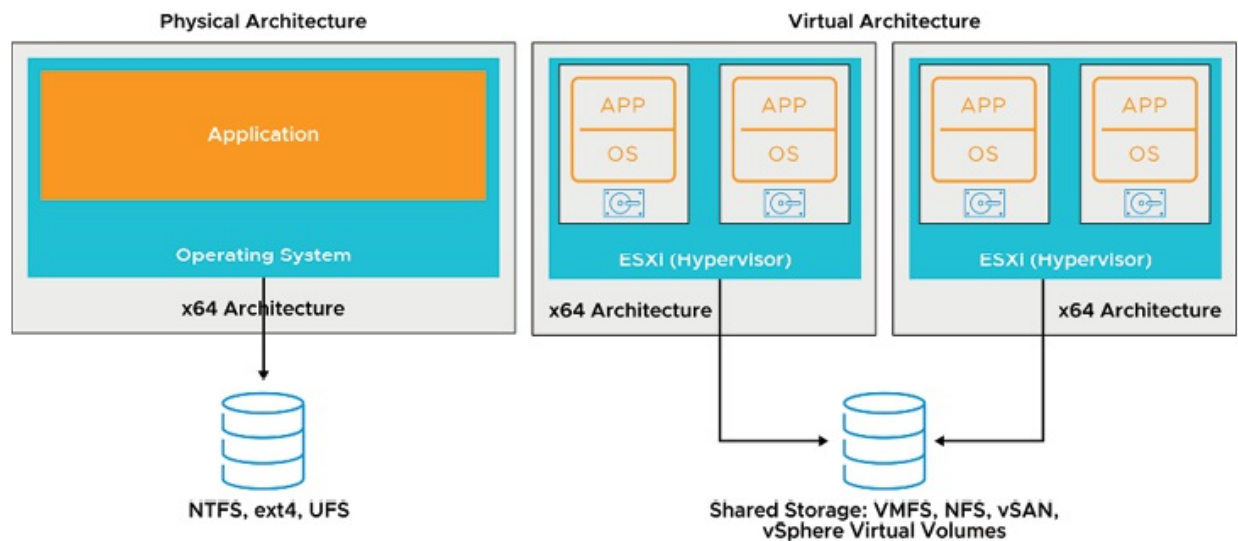


Figure 2.10: Datastore

(Source: VMware)

Virtual discs and data stores

Datastores are used by ESXi to control virtual machine storage. Datastores are logical storage containers that offer a uniform storage paradigm across various hardware types by hiding the actual components of the storage infrastructure. VMFS, which is made especially to manage VM storage in a virtualized environment, is used to format datastores for block storage.

vSphere supports the following types of datastores:

- **VMFS:** Storage virtualization optimized for virtual machines is offered via the VMware VMFS, a clustered file system. The same VMFS datastore can be read and written to by several ESXi hosts at once.
- **NFS:** ESXi hosts connect to a **network-attached storage (NAS)** device

via the file-sharing protocol known as **Network File System (NFS)**.

- **vSAN:** A software-defined storage system, vSAN gives virtual machines access to shared storage. vSAN transforms ESXi hosts in a cluster into a single datastore by virtualizing local physical storage, such as HDD or SSD devices.
- **vSphere Virtual Volumes:** By abstracting physical hardware resources to logical pools of capacity, the datastore virtualizes SAN and NAS devices. Servers or storage arrays are made to handle every facet of vSphere Virtual Volumes. vSphere Virtual Volumes storage is not directly accessible from datastores or ESXi hosts.

Let us understand flexibility in storage and protocol compatibility.

Numerous enterprise storage systems are compatible with VMFS because it integrates easily with a variety of storage protocols, including iSCSI, Fibre Channel, and Fibre Channel over Ethernet. It is because of its dynamic scalability, VMFS datastores can expand and aggregate storage without experiencing any downtime.

Let us understand cluster integration and distributed locking.

VMs can join to a storage cluster with less administrative work thanks to VMFS's distributed locking functionality, which allows VMs to safely access and share storage resources. A crucial component of VMware's virtual architecture, this integration of VM and storage resources simplifies operations across hosts and storage networks.

Let us discuss GPU virtualization.

Graphics processing unit (GPU) devices are designed to handle complex graphics tasks efficiently, taking the load off the CPU. By virtualizing GPUs, vSphere enables VMs to benefit from high-performance graphics processing without requiring dedicated physical GPUs for each VM.

The following are the use cases for **Virtual GPUs (vGPUs)**:

- **High-quality graphics:** vGPUs allow VMs to perform rich 2D and 3D graphics operations, which are essential for users who require detailed visual displays.
- **VMware Horizon Virtual Desktops:** Virtual desktops that require high graphics performance can use vGPUs to deliver a smooth, responsive experience for end-users.

- **Graphics-intensive applications:** Industries like architecture and engineering benefit from virtualized GPU resources to run design, modeling, and simulation applications efficiently.
- **High-performance server applications:** vGPUs support massively parallel tasks, such as scientific computations, which are common in fields that require heavy data analysis or large-scale simulations.

Let us understand configuration and compatibility.

VMs can be configured with up to four vGPU devices, making it possible to use multiple GPU accelerators within a single VM. VMware vSphere supports both AMD and NVIDIA GPUs, giving flexibility in hardware choices based on performance needs and budget.

Let us understand the relevance for servers.

Although servers typically do not have physical displays, GPU support is highly valuable in server virtualization. GPUs are increasingly used in server environments to support applications requiring extensive computation power, which makes GPU virtualization important for developers working with data-intensive applications.

Let us discuss vSphere Bitfusion.

Similar to how vSphere delivers CPU and memory resources, vSphere Bitfusion is a potent technology that allows vSphere to supply virtualized GPU resources to VMs on demand. Applications like **artificial intelligence (AI)**, **machine learning (ML)**, and data analytics that demand **high-performance computing (HPC)** capabilities will find this particularly helpful.

The following figure illustrates the vSphere bitfusion:



Figure 2.11: vSphere Bitfusion

(Source: VMware)

Key features of vSphere Bitfusion:

- **GPU sharing and pooling:** vSphere Bitfusion allows multiple VMs to share GPU resources, increasing efficiency by pooling GPUs from multiple hosts and making them available to VMs as needed.
- **On-demand resource allocation:** With vSphere Bitfusion, applications running in VMs can access GPU resources dynamically, receiving only the GPU power they require. This on-demand approach helps to reduce costs and maximize hardware utilization.
- **Network-based GPU access:** Bitfusion enables GPU access over the network, meaning VMs can leverage GPU resources from other hosts without needing direct access to a physical GPU. This flexibility is particularly advantageous for data centers that require scalability and flexibility in resource allocation.

Use case examples:

- AI and ML model training, which requires intensive compute power.
- Data analytics workloads that benefit from GPU acceleration.
- Computational research tasks that need scalable GPU resources.

vSphere Bitfusion streamlines GPU resource management in vSphere settings, allowing organisations to use GPU resources more efficiently and provide improved performance to applications that require it.

vSphere in the software-defined data center

A **software-defined data center (SDDC)** virtualizes all infrastructure components and automates data center management through software.

VMware vSphere is the cornerstone for creating an SDDC, delivering a versatile and efficient virtualized environment.

The following figure illustrates the software-defined data center:

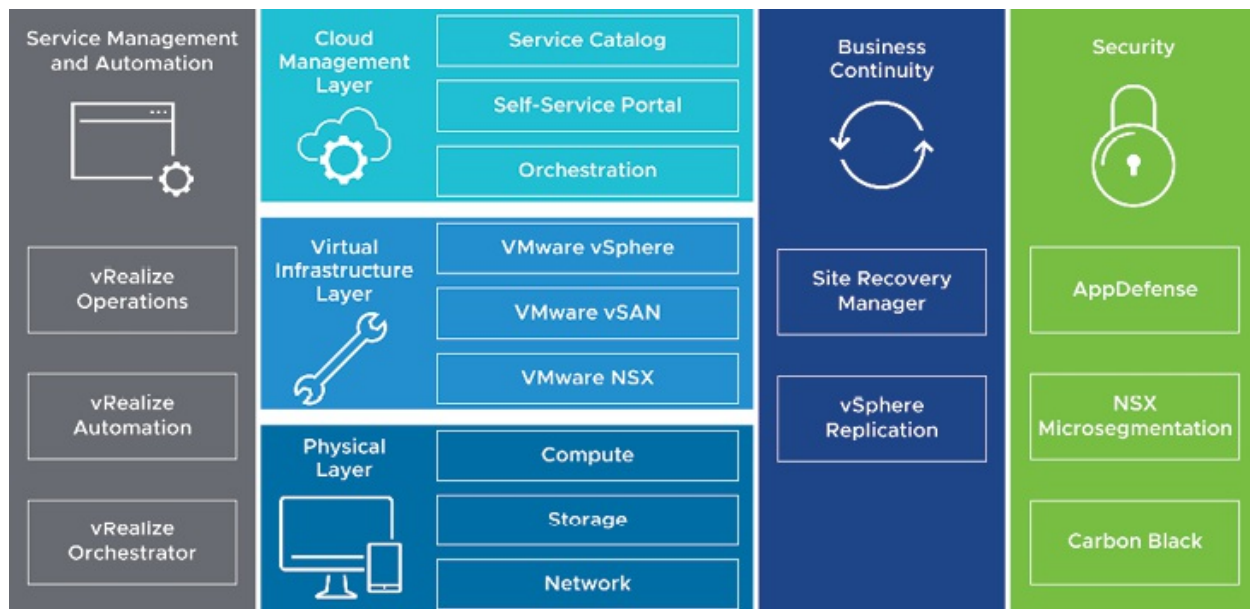


Figure 2.12: vSphere in software-defined data center

(Source: VMware)

An SDDC separates and virtualizes all important resources, including computing, storage, networking, and security, making it faster and more versatile than traditional hardware-based data centers.

These resources, which include CPU, memory, storage, and network, are abstracted and represented as software, resulting in enhanced flexibility, scalability, and ease of management.

The following are the components of an SDDC:

- **Service management and automation:** This part keeps an eye on and controls several data sources spread across different SDDC regions. For consistent availability and real-time insights, tools such as vRealize Log Insight and vRealize Operations Manager are used.
- **Cloud management layer:** This layer consists of a self-service interface for users to access the SDDC, orchestration processes to automate deployments, and a service catalog for resource provisioning.
- **Virtual infrastructure layer:** This layer supports the **infrastructure as a service (IaaS)** and **platform as a service (PaaS)** models and houses the hypervisor, resource pools, and virtualization control.
- **Physical layer:** Networking hardware, storage devices, and physical servers form the SDDC's framework.

- **Security layer:** This layer is made to adhere to strict compliance guidelines, controlling corporate risk and safeguarding virtualized workloads in accordance with legal mandates.

vSphere in the cloud infrastructure

Data center virtualization is the cornerstone of modern IT infrastructures. So, as management of the infrastructure and when vSphere manages the workload from the cloud, it becomes vSphere+. It allows organizations to efficiently manage their on-premises workloads from a cloud console. VMware vSphere+ is a subscription-based solution that gives on-premises workloads access to cloud advantages.

The following figure illustrates the vSphere in the cloud infrastructure:

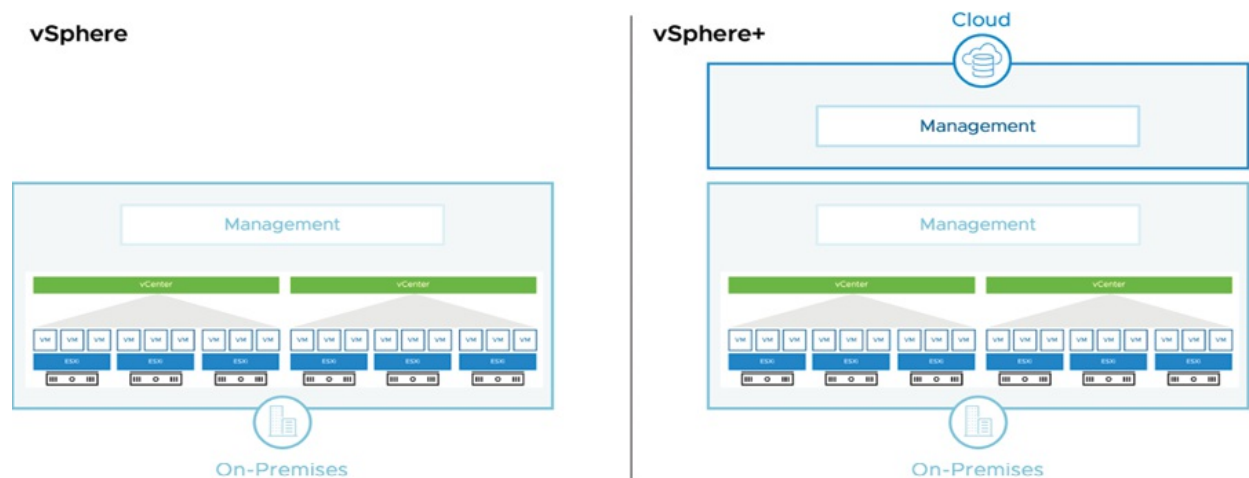


Figure 2.13: vSphere in the cloud infrastructure

(Source: VMware)

Both contemporary workloads based on containers and more conventional workloads based on virtual machines are supported by vSphere+. The kinds of workloads that are supported by vSphere+ and vSphere are the same. Your workloads are hosted on ESXi servers on-site. Your on-premises vSphere infrastructure and workloads are not transferred to the cloud.

Both on-premises and cloud components make up vSphere+:

- **On-premises components:**
 - ESXi hosts and vCenter instances

- cloud gateway that links the VMware Cloud Console to vCenter instances

- **Cloud components:**

- Cloud components include the VMware Cloud Console, which allows you to access cloud services and centrally manage on-premises infrastructure.
- Cloud services that supplement and improve on-premises capabilities for administrators (or IT operations) and developers (or DevOps)

There are no additional hardware requirements for vSphere+ beyond those of vSphere.

vSphere+ lets you access cloud services to augment and enhance on-premises capabilities:

- **Admin services:**

- Inventory management
- Events and alerts management
- VM provisioning
- Lifecycle management
- Configuration management

- **Developer services:**

- Tanzu Kubernetes Grid
- Tanzu integrated services

- **Add-on services:**

- Disaster recovery

The following figure illustrates vSphere+ managing on-premises workloads:

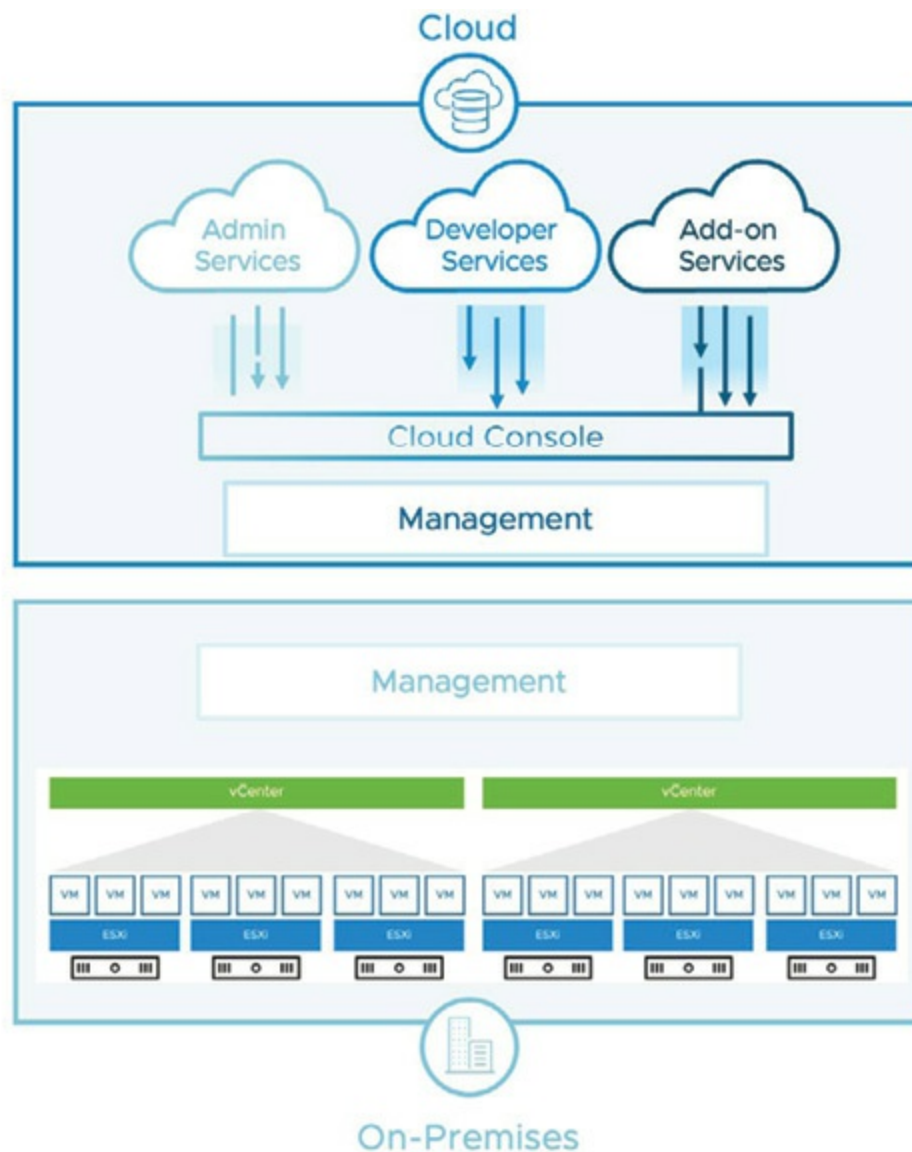


Figure 2.14: vSphere+

(Source: VMware)

VMware is creating vSphere+ add-on services to speed up capacity planning, ransomware prevention, disaster recovery, and more. VMware partner or sales representative can help with the list of add-on services that are either in development or currently for sale.

Numerous developer services are included in vSphere+, including as storage, network, registry, Kubernetes Grid, VM, Tanzu integrated services, Tanzu Mission-Control Essentials, and more. These services are free of charge and come with vSphere+.

Numerous administrative services are included in vSphere+, such as Event View, Global Inventory, Security Health Check, VM provisioning, lifecycle management, configuration management, and more. These services are free of charge and come with vSphere+.

User interfaces for vSphere

You may manage and interact with vSphere resources in the vSphere environment using a variety of interfaces, including the vSphere Client, VMware Host Client, PowerCLI, and ESXCLI. Each interface offers specific features for controlling a single ESXi host or the complete vSphere ecosystem.

For information on ports and protocols, refer to <https://ports.broadcom.com/>.

The following figure illustrates the different vSphere user interfaces:

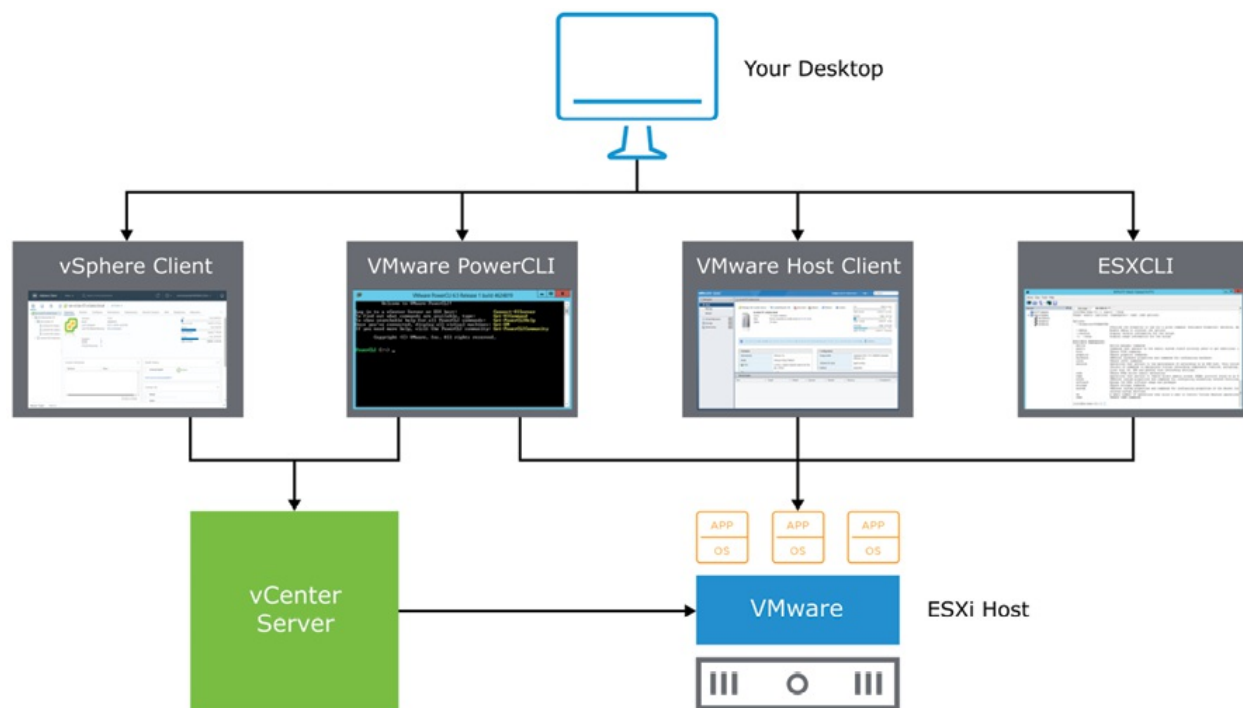


Figure 2.15: vSphere user interface

(Source: VMware)

VMware Host Client

An HTML5-based interface called the VMware Host Client is used to directly administer a single ESXi host, particularly if vCenter Server is unavailable. It can be accessed by going to **https://ESXi_FQDN_or_IP_Address/ui** in a compatible browser. The VMware Host Client is integrated into ESXi itself and features a clean, contemporary user interface that does not require plug-ins.

vSphere client

The vSphere Client offers extensive administrative capabilities for the vCenter Server and the vSphere environment and is likewise HTML5-based. It may be accessed at **https://vCenter_Server_Appliance_FQDN_or_IP_Address/ui**, (which redirects to the standard HTTPS port 443). The vSphere Client has an easy-to-use user interface and is independent of Adobe Flash and browser plug-ins.

ESXCLI vs. PowerCLI

A command-line tool called PowerCLI, which is based on PowerShell, has more than 700 cmdlets for managing and automating vSphere activities. It acts as a strong gateway to the vSphere API, enabling sophisticated automation scripts.

Another command-line utility for remotely managing ESXi hosts is called ESXCLI. Effective remote management is made possible with ESXCLI commands, which are available for Windows and Linux and can target certain ESXi hosts via vCenter.

For more detailed information about PowerCLI and ESXCLI, including downloading and usage, you can visit VMware's {code} page here, **<https://community.broadcom.com/vmware-code/home>**.

Conclusion

The fundamental ideas of virtualization and the critical role VMware vSphere plays in data center modernization have been covered in this chapter. By converting real hardware into adaptable, scalable, and effective virtual resources, enables businesses to minimize expenses, optimize resource usage,

and combine workloads. Leading the way in this technology is VMware vSphere, which offers a stable foundation for easily managing virtual machines, network setups, and storage options.

As we have seen, ESXi, the hypervisor for vSphere, isolates the physical resources such that several virtual machines can run separately but share a host. Centralized management is another benefit of vCenter Server, which provides administrators with the means to easily monitor, set up, and optimize their virtual infrastructure. These elements work together to provide the framework of a software-defined data center, which effectively distributes network, storage, and processing resources to satisfy changing business requirements.

Now that the readers have a firm understanding of these ideas, they can explore the technical details of installing and configuring VMware vSphere. We will start this trip by installing and configuring ESXi, the foundation of the vSphere environment, in [Chapter 3, Installing and Setting Up ESXi](#).

Points to remember

- Virtualization transforms physical resources into adaptive virtual resources, which optimize workloads and reduce costs.
- VMware vSphere is the industry's premier platform for managing virtual computers, networks, and storage in data centers.
- The ESXi hypervisor isolates physical resources, allowing several VMs to run independently on a single host.
- vCenter Server enables centralized management, which simplifies monitoring, configuration, and optimization.
- vSphere serves as the foundation for the SDDC, virtualizing computation, network, and storage to increase flexibility.
- Key vSphere components include ESXi, vCenter Server, vSAN, and NSX, which provide extensive virtualization capabilities.
- VMware vSphere+ is a subscription-based solution that gives on-premises workloads access to cloud advantages.

Exercises

1. What is the primary benefit of virtualization in data centers?
2. Explain the role of the ESXi hypervisor in the vSphere environment.
3. What functions does the vCenter Server provide in a virtualized infrastructure?
4. Define SDDC and list the key components that vSphere provides to support it.
5. Why is vSphere considered foundational for cloud and modern IT environments?

Lab exercises

1. **Access and review the lab environment at VMware HOL or the home lab.**
 - a. Access the jump host or VMware HOL to access the desktop
 - b. Log in to an ESXi host with VMware Host Client
 - c. Log in to the vCenter Server with the vSphere Client
2. **Explore vCenter server features and navigation.**
 - a. Log in to the vSphere Client connected to the vCenter Server.
 - b. Explore features such as the inventory view, resource pools, and VMs.
 - c. Review various tabs (e.g., Summary, Monitor, Configure) to understand the information available for each VM and host.

Join our Discord space

Join our Discord workspace for latest updates, offers, tech happenings around the world, new releases, and sessions with the authors:

<https://discord.bpbonline.com>



CHAPTER 3

Installing and Setting Up ESXi

Introduction

In this chapter, we will look at VMware ESXi, the core hypervisor that powers the vSphere platform, and how it manages virtualized hardware resources and supports virtual machines. ESXi, the foundation of VMware's virtualization platform, allows you to quickly create and run **virtual machines (VMs)**, delivering the dependability and scalability required for modern data centers.

ESXi is more than just a platform for operating VMs; it is also a reliable system that, when properly configured, offers excellent performance, security, and stability for all VMs. Understanding the fundamentals of installing and configuring ESXi lays the framework for a virtualized environment that fulfills your organization's requirements for dependability and resource efficiency. This chapter will help readers understand how to set up a robust ESXi host and prepare them for more sophisticated setups in the vSphere environment.

Note: VMware is now part of Broadcom and is known as 'VMware by Broadcom'. All references to VMware in this book reflect this change.

Structure

In this chapter, we will cover the following topics:

- Installing an ESXi host
- Configuring ESXi host settings
- Recognizing ESXi user account best practices

Objectives

By the end of this chapter, readers should understand the ESXi host architecture, critical hardware, and compatibility requirements for a successful ESXi installation. They will be able to install ESXi and finish its basic configuration, with tools such as the **Direct Console User Interface (DCUI)** and VMware Host Client for initial setup. Additionally, readers will learn how to adopt security best practices for ESXi user accounts, establishing a safe foundation for their virtual environment. Finally, they will be able to link their ESXi host to a vCenter Server for more efficient, centralized control of virtual resources.

Installing an ESXi host

ESXi is VMware's bare-metal hypervisor, which serves as the foundation for vSphere. ESXi is available as a standalone server (with a free version called vSphere Hypervisor) and as part of a licensed vSphere package, allowing organizations to operate numerous VMs on a single physical host while maintaining security, efficiency, and scalability.

The key features of ESXi area as follows:

- **Enhanced security:** ESXi provides strong security features including a host-based firewall, memory hardening, kernel module integrity, and support for **Trusted Platform Module (TPM 2.0)** and **Unified Extensible Firmware Interface (UEFI)** Secure Boot, which ensure a secure operating environment.
- **Small Disc footprint:** The lightweight design reduces resource utilization, resulting in increased stability and faster startup times.
- **ESXi quick boot:** This feature enables ESXi to reboot without entirely

reinitializing the server's BIOS, which speeds up patching and upgrading on supported hardware.

- **Flexible installation options:** ESXi can be installed on hard discs, SAN LUNs, SSDs, and SATADOM, as well as diskless using vSphere Auto Deploy for systems that boot directly from memory.

The VMware compatibility guide provides information on hardware compatibility with ESXi 8.0. Refer to the following link to access the VMware compatibility guide:

<https://www.vmware.com/resources/compatibility/search.php>.

VMware also warns against utilizing USB devices as boot drives (supported until ESXi 8.x Version) owing to reliability concerns, preferring SSDs instead. For further information on ESXi installation and supported storage devices, see **VMware Knowledge Base Article 82515** or access the following link:

<https://knowledge.broadcom.com/external/article?legacyId=82515>.

ESXi has a small disc footprint, which enhances security and dependability. ESXi offers enhanced safety with the following features:

- **Host-based firewall:** Adds an extra layer of security by controlling access between the ESXi administration interface and the network.
- **Memory hardening:** ESXi employs memory randomization and protections to prevent exploits targeting memory vulnerabilities, making it more difficult for malicious code to compromise the system.
- **Kernel module integrity:** ESXi uses digital signatures to verify the validity of kernel modules, drivers, and applications, preventing unauthorized code from loading.
- **Trusted platform module (TPM 2.0):** Creates a trustworthy hardware platform by ensuring that the boot process and loaded drivers are genuine and uncompromised.
- **Unified Extensible Firmware Interface (UEFI) secure boot:** On systems that enable UEFI secure boot, ESXi checks **VMware infrastructure bundles (VIBs)** against a digital certificate in the firmware to assure boot integrity.
- **Lockdown modes:** This vSphere feature deactivates login and API functions from being executed directly on an ESXi host.

Organizations can use ESXi's advanced features to create a secure, scalable, and efficient virtualized environment that matches modern data center needs.

ESXi installation requirements

Check that the host fulfils the minimum hardware requirements supported by the intended ESXi version, as shown:

- Supported server platform
- At least two CPU cores.
- A minimum of 8 GB of physical RAM and 12 GB for a production environment.
- One or more Ethernet controllers with Gigabit or higher speeds
- Boot disc with at least 32 GB of persistent storage.

Interactive ESXi installation

For installations with fewer than five hosts, an interactive installation is suggested. Administrator can boot from the installer and follow the on-screen prompts of the installation process, as follows:

- Begin from the Welcome page.
- Accept the EULA.
- Select a disc.
- Select a keyboard layout.
- Enter the root password.
- Start the installation.

The following image illustrates the interactive ESXi installer steps:



Figure 3.1: ESXi installer

(Source: VMware)

The ESXi installer can be run via a CD or DVD, a bootable USB device, or via PXE booting over the network.

Configuring an ESXi host

Upon successful installation, an ESXi host is assigned an IP address using DHCP. To set up and update critical configurations, such as network settings, use the DCUI, a text-based interface that allows for keyboard-only input, as shown:

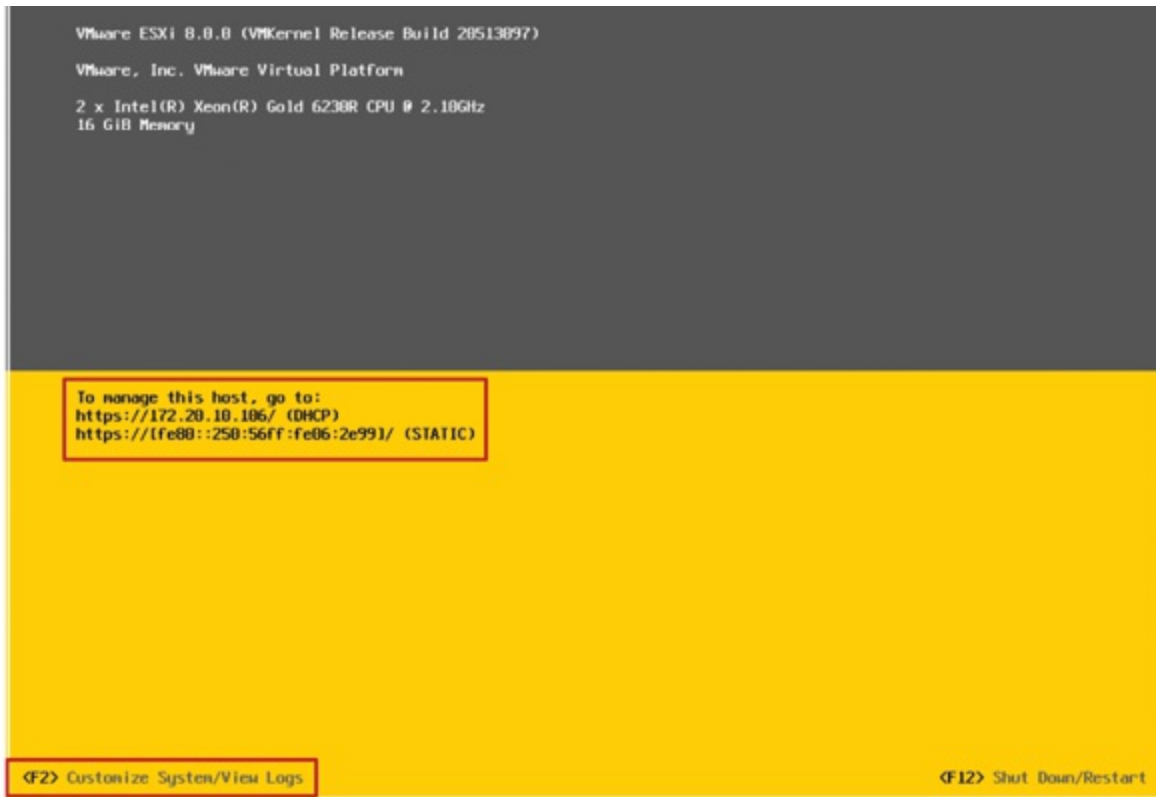


Figure 3.2: ESXi installation completed

(Source: VMware)

The DCUI is a low-level configuration tool that may be accessed directly from the server console. The DCUI is mostly used for initial setup and provides basic ESXi host configuration and management options. To start customizing system settings, press F2 in the DCUI to access a variety of configuration choices required for creating a safe and efficient virtualization environment.

Configuring ESXi host settings

ESXi hosts can be configured using one of the following modes:

- **Direct Console User Interface (DCUI):** DCUI is a text-based interface accessed directly from an ESXi host's physical console which enables initial configuration tasks such as network setup, password resets, and troubleshooting when remote access is not available.
- **vSphere Client:** This mode offers a comprehensive graphical interface for remote management and gives a complete control over host settings including networking, storage, security, and performance optimization.

Configuring an ESXi host with management network

Before an ESXi host can be completely operational, its management network settings must be correctly set up. By default, the host receives a DHCP-assigned IP address. However, setting a static IP address is frequently chosen for stability in administration access.

The DCUI allows you to change important management network settings such as:

- Selection of network adapters
- VLAN ID configuration for traffic segmentation
- IPv4 and IPv6 setups (including IP address, subnet mask, and default gateway)
- Host name assignment
- DNS server and suffix configurations

The following figure illustrates the ESXi management network configuration using DCUI:

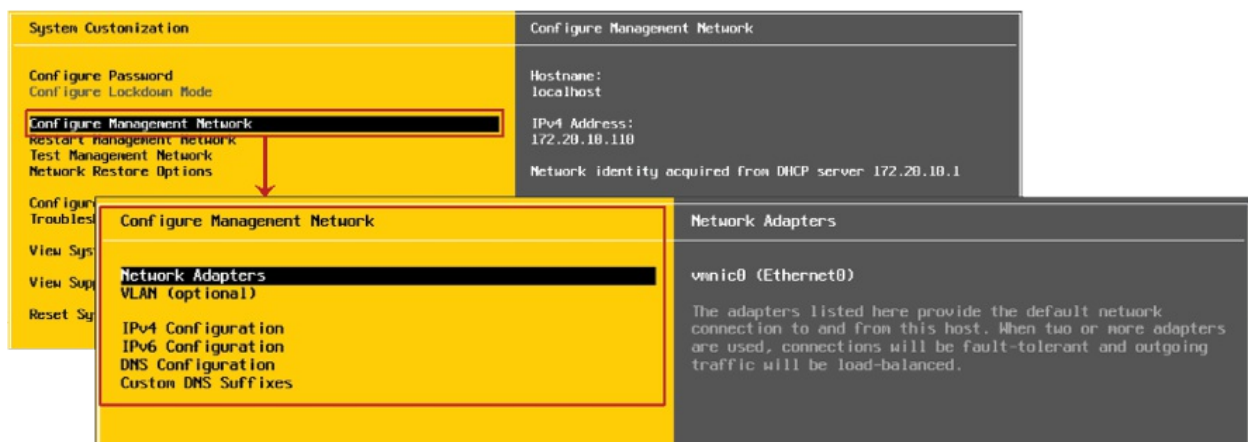


Figure 3.3: ESXi Management Network

(Source: VMware)

The DCUI also allows you to manage and troubleshoot network settings without rebooting the entire system. The DCUI offers the following key network administration tasks:

- Configure VLAN and IP address settings for both IPv4 and IPv6.
- Customizing DNS suffixes for network resolution.
- Restarting the management network (if changes were made, without rebooting the system).
- Test network connectivity (using tools like ping and DNS requests).
- Restoring the network configuration to default is especially useful if settings are mistakenly mismatched.

This configuration creates a solid network foundation, ensuring that your ESXi host is accessible and manageable in your virtualized environment.

Configuring an ESXi host with root access

In ESXi, root access configuration is an essential part of securing the host. Administrators can use the DCUI to manage root settings effectively.

Key tasks include:

- **Setting or changing the root password:** The root account, the primary administrative user for the ESXi host, requires a strong and complex password. This password is initially configured during the ESXi installation, but administrators can update it anytime through the DCUI as needed.
- **Activating or deactivating lockdown mode:** Lockdown mode enhances security by limiting management access to the ESXi host only through vCenter Server. This setting is available exclusively for hosts managed by vCenter and restricts direct access to the host for improved security controls.

The following figure illustrates the ESXi host root access:



Figure 3.4: ESXi host root access

(Source: VMware)

Root is the administrative username for the ESXi host. By configuring these root access settings, administrators establish a robust security framework that protects the ESXi host from unauthorized access and helps ensure that only approved management methods are used.

Configuring an ESXi host with troubleshooting options

The DCUI provides various configuration choices necessary for operating ESXi hosts. The DCUI allows you to configure the keyboard layout, obtain support information, enable troubleshooting services, and check system logs. By default, the keyboard layout is set to US English, although this can be changed if necessary.

In addition to these options, the DCUI gives you access to troubleshooting services, which are disabled by default for security reasons. This includes:

- **vSphere ESXi Shell:** For local troubleshooting on the ESXi host.
- **SSH:** Allows remote troubleshooting using SSH clients such as PuTTY.

The following figure illustrates the troubleshooting options in DCUI:

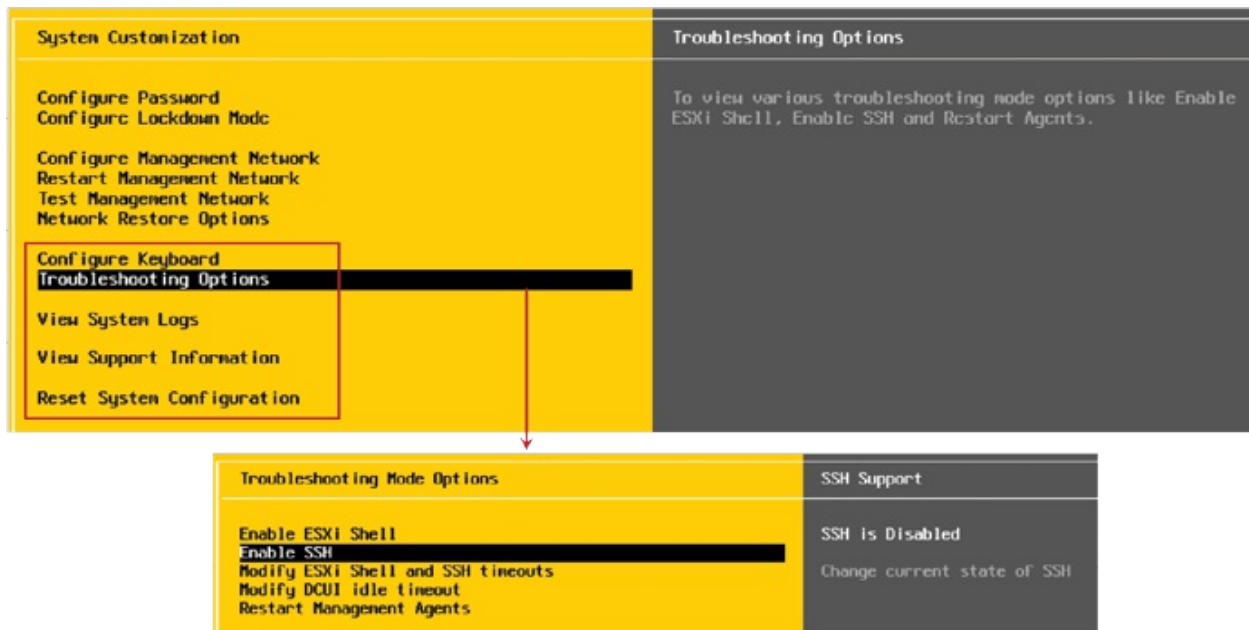


Figure 3.5: ESXi host other settings

(Source: VMware)

Best practices demand that these services be deactivated unless they are necessary, usually while working with VMware technical support to resolve difficulties. The DCUI also includes a Reset System Configuration option, which restores the host's default software settings and removes any custom extensions or packages. When selecting this option, be cautious because it also clears the root password, potentially allowing unauthorized access to the machine.

To maintain a secure ESXi environment, troubleshooting services should only be activated when absolutely necessary and swiftly deactivated when no longer needed.

Time synchronization for the ESXi host

Maintaining synchronization across a vSphere environment requires precise timekeeping. Synchronizing the ESXi host's clock with a reliable time source ensures accurate performance data, precise time stamps in log messages, and consistent time settings for VMs linked to the host.

Time synchronization on ESXi hosts has various benefits:

- **Accurate performance metrics:** With correct time data, performance graphs can be reliably evaluated.

- **Reliable log entries:** Accurate time stamps in log messages improve audit logs and aid in troubleshooting.
- **Consistent VM time:** VMs can synchronize their time with the ESXi host, which is important for time-sensitive applications such as database systems.

The following are the methods of time synchronization:

- **Manual configuration:** Enter the time and date manually.
- **Network Time Protocol (NTP):** Provides millisecond-level time synchronization, making it acceptable for the majority of general-purpose applications.
- **Precision Time Protocol (PTP):** Provides microsecond-level precision, making it excellent for applications that require extremely exact timing.

NTP or PTP can be configured using either the VMware Host Client or the vSphere Client, but they cannot be used simultaneously:

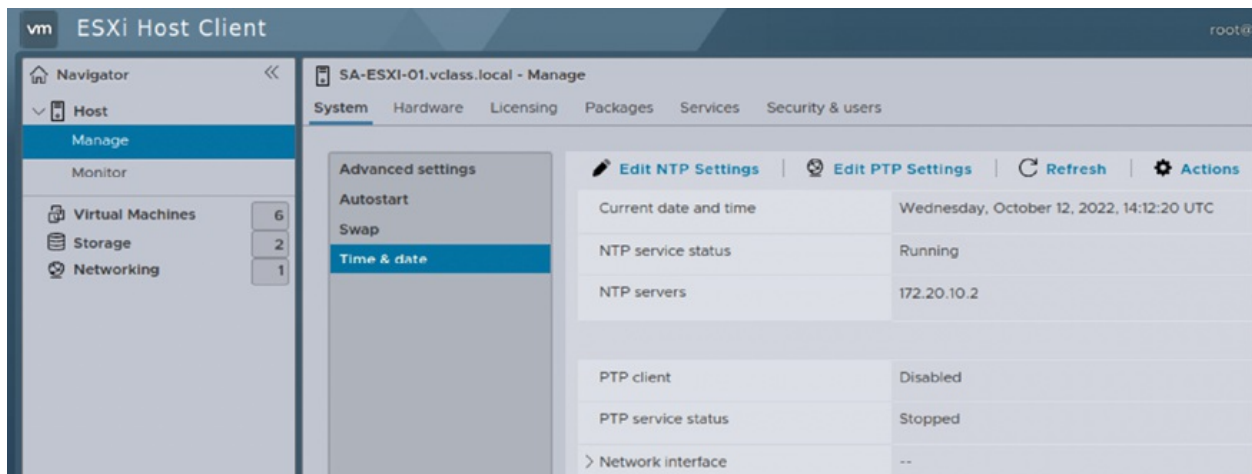


Figure 3.6: Configuring time and date in VMware Host Client

(Source: VMware)

Configuring NTP

An ESXi host can be set up as an NTP client. The NTP client communicates over UDP over port 123:

Edit NTP Settings

Specify how the date and time of this host should be set.

☐ Manually configure the date and time on this host

10/12/2022 7:22 AM

☒ Use Network Time Protocol (enable NTP client)

NTP service startup policy: Start and stop manually

NTP servers: 172.20.10.2

Separate servers with commas, e.g. 10.31.21.2, fe00::2800

CANCEL SAVE

Figure 3.7: Configuring NTP using VMware Host Client

(Source: VMware)

NTP is a client or server protocol. When an administrator configures the ESXi host to be an NTP client, it synchronizes its time with an NTP server, which can be an Internet server or the company's NTP servers.

For more information on NTP, refer <http://www.ntp.org>.

Configuring PTP

PTP provides hardware-based timestamping for virtual machines and hosts on a network. If the PTP service fails, you can fall back to NTP.

PTP client communicates over UDP on ports 319 and 320. The following figure illustrates the PTP configurations:

Precision Time Protocol

Use Precision Time Protocol (PTP) as the primary service to synchronize the system time.

Network adapter type: VMkernel adapter

Device name: vmk0

IPv4 address: 172.20.10.51

Subnet mask: 255.255.255.0

☒ Enable monitoring events

☒ Enable fallback

Fallback NTP servers: 0.vmware.pool.ntp.org, 1.vmware.pool.ntp.org, 2.vmware.pool.ntp.org

If you enter multiple server names and IP addresses, use commas to separate them.

CANCEL OK

Figure 3.8: Configuring PTP using the vSphere Client

(Source: VMware)

Using the vSphere Client to configure PTP allows for precise time synchronization on ESXi hosts, including software and hardware timestamping options:

- To configure hardware timestamping, choose **PCI passthrough** as the network adapter type.
- To configure software timestamping, choose **VMkernel Adapter** as the network adapter type.

Controlling remote access to an ESXi host

Administrator can use the vSphere Client or VMware Host Client to customize crucial security settings for remote access to an ESXi host. By default, the ESXi firewall is active, blocking all incoming and outgoing traffic, excluding for services explicitly allowed. Only users with administrator privileges can manage services such as NTP and SSH. Enabling lockdown mode further restricts remote access, allowing host management

only via the DCUI or through vCenter.

The following image illustrates the ESXi host setting using the vSphere client:

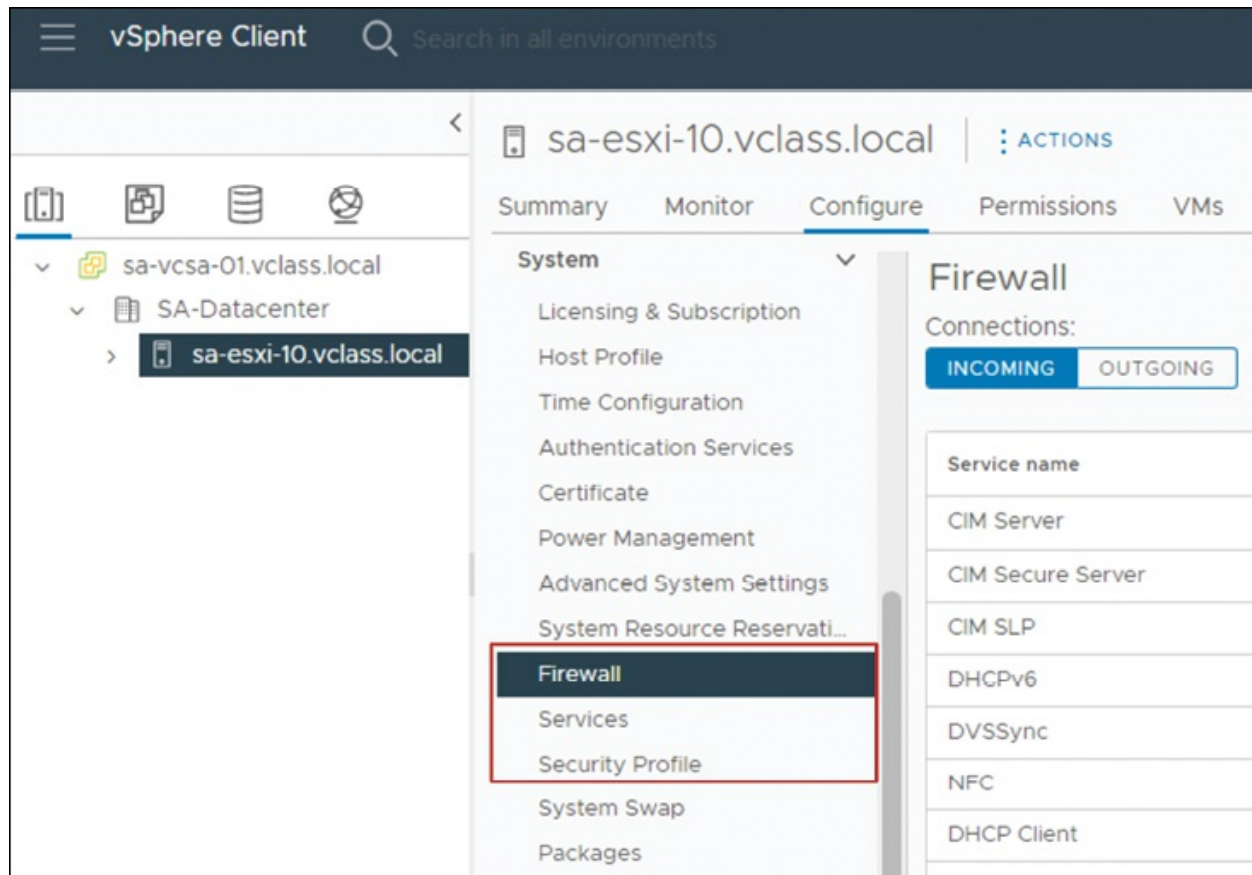


Figure 3.9: Controlling remote access to an ESXi

(Source: VMware)

The ESXi firewall is designed to restrict unauthorized access by preventing open ports and restricting service access. Firewall settings can be modified for both incoming and outgoing connections, allowing administrators to specify allowed IP addresses or ranges for certain services. For each service, you can also temporarily change its status (START, STOP, RESTART) or set it to start automatically with the host.

Recognizing ESXi user account best practices

Effective management of user accounts is essential for securing ESXi hosts and vCenter systems.

The best practices are as follows:

- **Limit root access:** Control and restrict access to the powerful root account on ESXi hosts. Ensure strong passwords are in place, with at least eight characters, including special characters, uppercase and lowercase letters, and numbers. Change root passwords periodically to maintain security.
- **Centralized management via vCenter server:** Where possible, manage ESXi hosts centrally through vCenter Server using the vSphere Client, rather than directly on the host. This approach allows for streamlined access and better security management.
- **Minimize local accounts on hosts:** Avoid creating local user accounts on ESXi hosts. Instead, add ESXi hosts to an Active Directory domain and assign relevant administrators to the ESX Admins domain group, which grants root privileges on the ESXi hosts to domain group members.
- **Use direct host access carefully:** Direct access to ESXi hosts should be used only if the vCenter server is inaccessible. In these instances, use VMware Host Client or, if necessary, the ESXi Shell. Reserve these methods for troubleshooting or configurations that cannot be managed through vCenter.

By adhering to these best practices, administrators can enhance security and restructure user account management across the virtual infrastructure, ensuring only authorized users can access the critical systems and resources.

Conclusion

In this chapter, we covered the essential steps for installing and configuring VMware ESXi, the foundational hypervisor for vSphere environments. Readers explored the hardware and compatibility requirements, installation process, and initial configuration steps for ESXi, ensuring a strong base for building your virtualized environment. We also explored the use of the DCUI and the VMware Host Client for initial setup, highlighting their roles in network and security configuration. Finally, readers learned about best practices for securing user accounts, an essential measure for safeguarding

your ESXi host.

With ESXi now set up and configured, readers are ready to establish centralized management through VMware vCenter Server. Connecting ESXi hosts to a vCenter Server will streamline the management process, allowing for a unified view and enhanced control over your virtual infrastructure.

Now that readers have a firm understanding of the installation and setup of VMware ESXi, readers are ready to explore further into the technical aspects of centralized management. In the next chapter, we will begin our journey into deploying and configuring vCenter Server, the centralized management solution that serves as the backbone of the vSphere environment.

In the next chapter, we will guide the readers through understanding ESXi host communication, deploying the vCenter Server Appliance, and configuring essential vCenter settings to create an organized and manageable virtual infrastructure.

Points to remember

- ESXi is a bare-metal hypervisor that manages virtualized resources on a host.
- The DCUI is used to first configure initial network and security settings.
- The VMware Host Client offers a web-based interface for configuring individual ESXi hosts.
- Following best practices for ESXi user account management is critical for keeping a secure and organized environment.
- Connecting ESXi hosts to vCenter Server provides centralized management and enhanced resource allocation capabilities.

Exercises

1. What are the minimum hardware and compatibility requirements for installing ESXi?
2. Describe the process of installing ESXi on a physical host and outline key configuration steps.

3. What are the primary functions of the DCUI, and when might you use it over other management tools?
4. Explain the importance of connecting an ESXi host to a vCenter Server and the advantages of centralized management.

Lab exercises

1. Install ESXi on a physical or virtual machine:
 - **Objective:** This lab walks the readers through the installation of the ESXi hypervisor on a compatible system, either physical or virtual.
 - **Prepare for installation:**
 - Verify that the target machine meets ESXi hardware requirements, including CPU, memory, and storage.
 - Download the ESXi installer ISO from the VMware website.
 - Burn the ISO to a USB drive (for physical machines) or attach it to the virtual machine as a CD/DVD drive.
 - **Boot from the ESXi installer:**
 - Insert the bootable USB drive or configure the virtual machine to boot from the ISO.
 - Power on the machine and boot from the installer media.
 - **Installation process:**
 - Select the drive where to install ESXi (ensure it is a dedicated or empty drive).
 - Follow the on-screen instructions to accept the **End User License Agreement (EULA)**.
 - Choose the appropriate keyboard layout and language options.
 - **Set root password:** Enter and confirm a secure root password for accessing the ESXi host.
 - **Finalizing installation:**

- Review the installation settings and confirm to begin the installation.
- Once the installation is complete, reboot the host.
- Remove the installation media before the machine reboots to avoid reinstalling.
- **Verify installation:** Once rebooted, verify that ESXi boots up and displays the DCUI on the screen.

2. Configure ESXi network settings using DCUI:

- **Objective:** This lab guides the readers through using the DCUI to configure basic network settings on the ESXi host.
 - **Access the DCUI:**
 - After the ESXi host boots up, the DCUI appears on the console screen.
 - Press F2 to access the customization options, then enter the root credentials set during installation.
 - **Configure network settings:**
 - From the DCUI main menu, select Configure Management Network.
 - Configure the following options as needed:
 - ◆ **IPv4 configuration:** Choose a static IP address or use DHCP.
 - ◆ **IPv6 configuration** (if applicable): Set IPv6 options for management.
 - ◆ **DNS configuration:** Enter DNS server addresses and set a hostname for the ESXi host.
 - ◆ **VLAN ID:** If using VLANs, set the VLAN ID for management traffic.
 - **Test network connectivity:**
 - In the DCUI, select **Test Management Network** to verify

connectivity.

- Check that the ESXi host can successfully ping the gateway, DNS server, and resolve its hostname.

- **Save and exit:**

- Once network settings are configured, press Esc to exit.
- Choose to restart the management network if prompted to apply changes.

3. Log into VMware Host Client and perform basic configurations:

- **Objective:** This lab demonstrates how to access the VMware Host Client and perform basic configurations on the ESXi host.

- **Access the VMware Host Client:**

- From a web browser, enter the IP address of the ESXi host in the following format, **https://<ESXi_IP_Address>/ui**.
- Log in with the root account credentials.

- **Verify host information:** Once logged in, view the Summary tab to confirm host details, including version, hardware, and network configurations.

- Configure time and date settings:
 - ◆ Go to manage | system | time and date.
 - ◆ Set the time zone, configure **Network Time Protocol (NTP)** settings, and ensure that the time and date are correct.

- **Check and configure network adapters:**

- Under networking, review the status and settings of physical network adapters.
- Ensure that the network adapters are connected and have proper speed and duplex settings.

- **Add a datastore (optional):**

- Under storage | datastores, select new datastore.
- Follow the wizard to add a local or shared datastore to the ESXi host, allowing it to store VM files.
- **Verify host health:**
 - Under monitor | hardware and performance, check hardware health and resource usage.
- **Log out:**
 - Once configurations are complete, log out of the VMware Host Client.

Join our Discord space

Join our Discord workspace for latest updates, offers, tech happenings around the world, new releases, and sessions with the authors:

<https://discord.bpbonline.com>



CHAPTER 4

vCenter Deployment and Configuration

Introduction

This chapter explores the critical steps of deploying and configuring vCenter Server, the primary administration hub for the VMware vSphere installation. As the central command center, vCenter allows administrators to efficiently manage many ESXi hosts and virtual machines, resulting in a more simplified approach to managing complex virtualized data centers. Mastering vCenter setup and configuration is more than simply a recommended practice; it is critical to sustaining an optimized, secure, and scalable infrastructure.

vCenter provides comprehensive capabilities for monitoring, configuring, and managing the whole virtualized environment from a single interface. This chapter will walk readers through every step of effectively deploying and configuring vCenter, creating a solid foundation for the virtualized environment. Whether the readers want to boost performance, security, or operational efficiency, a properly configured vCenter can help realize the full potential of the virtual infrastructure.

Let us start by creating a strong management platform that will simplify and enhance VMware operations.



Note: VMware is now part of Broadcom and is known as 'VMware by Broadcom.' All references to 'VMware' in this book reflect this change.

Structure

In this chapter, we will cover the following topics:

- Understanding ESXi hosts communication with vCenter
- Deployment of vCenter Server Appliance
- Configuring essential vCenter settings
- Managing license keys with the vSphere Client
- Organizing vCenter inventory objects
- Explaining vCenter permissions
- Gaining insights from vCenter logs and events

Objectives

By the end of this chapter, readers will understand how ESXi hosts communicate with the vCenter server and appreciate the significance of this interaction for centralized management. Readers will learn how to deploy and configure the **vCenter Server Appliance (vCSA)**, laying the groundwork for controlling the virtualized infrastructure. Readers will also learn how to configure critical vCenter settings that are required for smooth integration and optimal performance in the vSphere installation. This chapter also explains how to manage licensing keys in vCenter using the vSphere Client, which gives centralized control over software entitlements.

Additionally, readers can organize and structure vCenter inventory objects, resulting in a logical hierarchy that facilitates resource allocation and management. Understanding vCenter permissions and properly assigning access privileges are key skills for ensuring safe, role-based resource access. Finally, readers will learn how to access and interpret vCenter logs and events, which are essential for monitoring operations, troubleshooting issues, and keeping accurate records of system performance and changes. These objectives are intended to provide readers with the knowledge required to deploy, configure, and maintain vCenter Server as the primary command

point for the VMware vSphere ecosystem.

Understanding ESXi hosts communication with vCenter

The vCSA is a preconfigured Linux-based virtual computer that is intended to execute vCenter and its associated services quickly. vCenter, which serves as the central administrative point for ESXi hosts and virtual machines in a network, is critical for administering and optimizing virtualized systems.

The following are the key features of the vCSA:

- **Core software components:**
 - **Photon OS:** Photon OS is a lightweight, open-source Linux distribution designed for vCSA.
 - **PostgreSQL database:** An integrated database that stores vSphere data and configurations.
 - **vCenter services:** A set of management tools for streamlining vSphere operations.
- **Deployment options:** During deployment, administrators can customize the appliance size to fit the vSphere environment and allocate sufficient storage for database needs.
- **Centralized management:** vCSA provides centralized management for ESXi hosts and virtual machines, including provisioning, configuration, and monitoring. It offers advanced features such as:
 - **vSphere Distributed Resource Scheduler (DRS):** Automatically balances workloads across hosts.
 - **vSphere High Availability (HA):** Ensures that VMs experience little downtime during host failures.
 - **vSphere Fault Tolerance:** Protects essential workloads by producing duplicate VM instances.
 - **vSphere vMotion and Storage vMotion:** Enables live migration of VMs and storage with no downtime.
- **Deployment method:** The vCSA is deployed as a virtual appliance on an

ESXi host in the infrastructure. Its Linux-based design offers optimal performance when executing vCenter components.

The following figure illustrates the vCenter management platform:

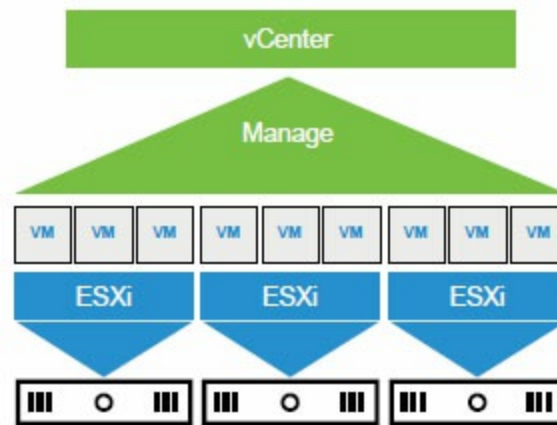


Figure 4.1: vCenter management platform

(Source: VMware)

vCenter enables organizations to easily pool and manage the resources of numerous ESXi hosts, allowing for seamless operations and scalability in virtualized environments.

vCenter services

The vCSA includes a strong array of services that can be preconfigured and configured in a single virtual machine. These services collaborate to provide centralized management and increase the operational effectiveness of the vSphere environment.

The following are the examples of a few services:

- **vCenter server:** Manages ESXi hosts, virtual machines, and other resources in the virtualized environment.
- **vSphere client:** The vSphere Client is a user-friendly web-based interface for managing vCenter and ESXi resources.
- **License service:** Ensures that licenses are properly allocated and compliant throughout the entire vSphere environment.
- **vSphere auto deploy:** Allows for quick provisioning of ESXi hosts via network booting, which is perfect for large-scale deployments.

- **Content Library:** The Content Library centralizes storage for VM templates, ISO images, and other key assets, making it easier to share and deploy.
- **vSphere Lifecycle Manager (vLCM):** Streamlines host and cluster lifecycle processes such as patching, upgrades, and updates.
- **vSphere Trust Authority (vTA):** Improves security by offering attestation and encryption services for sensitive workloads.

The following figure illustrates the vCenter services:

Service	Mode	Health	State
ImageBuilder Service	Manual		Stopped
License Service	Automatic	Healthy	Started
Service Control Agent	Automatic	Healthy	Started
vAPI Endpoint	Automatic	Healthy	Started
vCenter Server Profiles	Automatic	Healthy	Started
VMware Analytics Service	Automatic	Healthy	Started
VMware Appliance Monitoring Service	Automatic	Healthy	Started
VMware Certificate Authority Service	Automatic	Healthy	Started
VMware Certificate Management Service	Automatic	Healthy	Started
VMware ESX Agent Manager	Automatic	Healthy	Started
VMware HTTP Reverse Proxy	Automatic	Healthy	Started
VMware Lookup Service	Automatic	Healthy	Started
VMware Observability Vapi Service	Automatic	Healthy	Started
VMware Performance Charts Service	Automatic	Healthy	Started
VMware Postgres	Automatic	Healthy	Started
VMware Postgres Archiver	Automatic	Healthy	Started

Figure 4.2: vCenter Services

(Source: VMware)

With these integrated services, the vCSA provides a comprehensive and powerful solution for deploying, administering, and optimizing VMware's virtualized infrastructure. Consolidating functions into a single appliance reduces complexity and streamlines management.

vCenter architecture

The *vCenter Server* architecture is designed to provide centralized management and seamless integration across VMware's virtualized infrastructure. Its components work in harmony to ensure the availability, scalability, and integrity of the vSphere environment.

The following are the core components of vCenter architecture:

- The **vSphere Client** serves as the primary user interface, allowing administrators to connect to vCenter and manage their vSphere environments centrally. It simplifies the management of ESXi hosts, virtual machines, and other resources by offering an intuitive, web-based interface. When an ESXi host is managed by vCenter, all administrative tasks should be performed through the vSphere Client for consistency and reliability.
- The **vCenter database** is a critical backbone of the architecture. It stores essential data such as:
 - Inventory objects (e.g., VMs, datastores, clusters).
 - Security roles and permissions.
 - Performance metrics and historical data.
 - Configuration details for hosts and virtualized resources.
 - The database ensures the integrity and availability of this data, supporting smooth operations and robust reporting within the vSphere environment.
- **Managed hosts** are the ESXi servers connected to vCenter. Through vCenter, administrators can monitor, configure, and optimize these hosts and their virtual machines. Centralized management of managed hosts enhances efficiency, reduces administrative overhead, and provides access to advanced vSphere features such as vMotion, DRS, and HA.

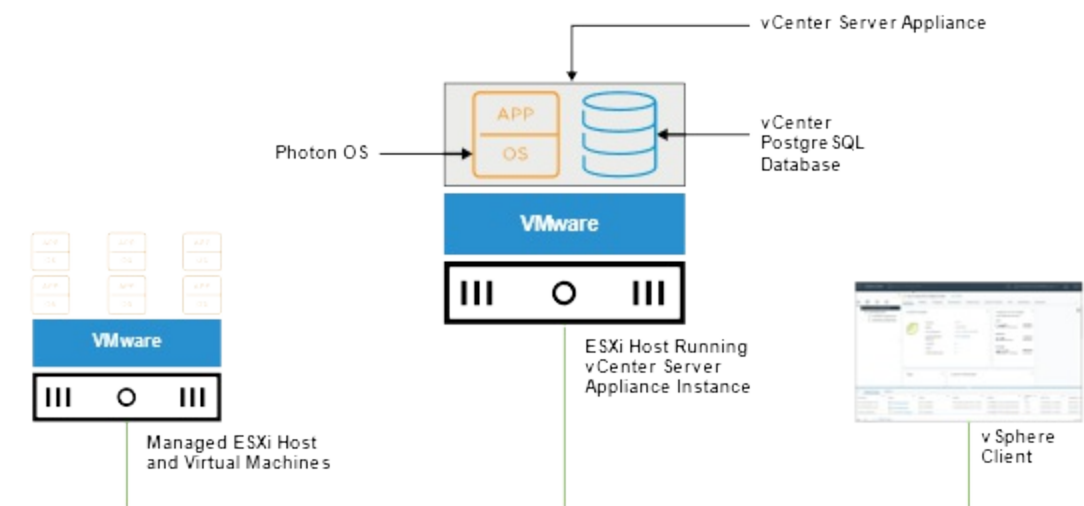


Figure 4.3: vCenter Architecture

(Source: VMware)

The synergy between these components creates a cohesive and highly functional architecture, enabling administrators to manage complex virtualized environments effectively. By unifying management, vCenter provides the foundation for scaling and optimizing IT infrastructure.

vCenter Single Sign-On

vCenter Single Sign-On (SSO) is a foundational security feature that enables seamless and secure communication between vSphere components. By leveraging a token-based authentication mechanism, SSO provides a centralized approach to manage user access and component interoperability.

The following are the key features of vCenter SSO:

- **Token-based authentication:** vSphere components communicate securely using **Security Assertion Markup Language (SAML)** tokens issued by the SSO service, eliminating the need for repetitive authentication between components.
- **Authentication sources:** SSO can authenticate users through built-in or external identity providers:
 - **Built-in identity provider:** By default, vCenter uses the *vsphere.local* domain as the identity source for user authentication.
 - **External identity providers:**
 - **Active Directory (AD)** via LDAP, LDAPS, OpenLDAP, or OpenLDAPS.
 - Federated authentication using **Active Directory Federation Services (AD FS)** for enhanced security.
 - VMware recommends AD over LDAP or Federated Identity with AD FS for vCenter Server and ESXi.
- **Integrated Windows Authentication (IWA):** While IWA is still supported, VMware strongly advises using AD over LDAP or federated solutions for improved security and compliance. For more details, see VMware knowledge base article 78506 at <https://knowledge.broadcom.com/external/article?legacyId=78506>.

Let us look at the user login flow with a built-in identity provider.

When vCenter acts as the identity provider, the login process follows these steps:

1. The user logs in to the vSphere Client.
2. vCenter Single Sign-On authenticates the user's credentials against a directory service (e.g., Active Directory).
3. An SAML token is generated and sent back to the user's browser.
4. The browser forwards the SAML token to vCenter, granting the user access.

The following are the benefits of vCenter SSO:

- Centralized authentication for vSphere components.
- Reduced complexity and enhanced security through federated identity solutions.
- Streamlined user management with integrated identity sources.

The following figure illustrates the vCenter Single-Sign-On login flow:

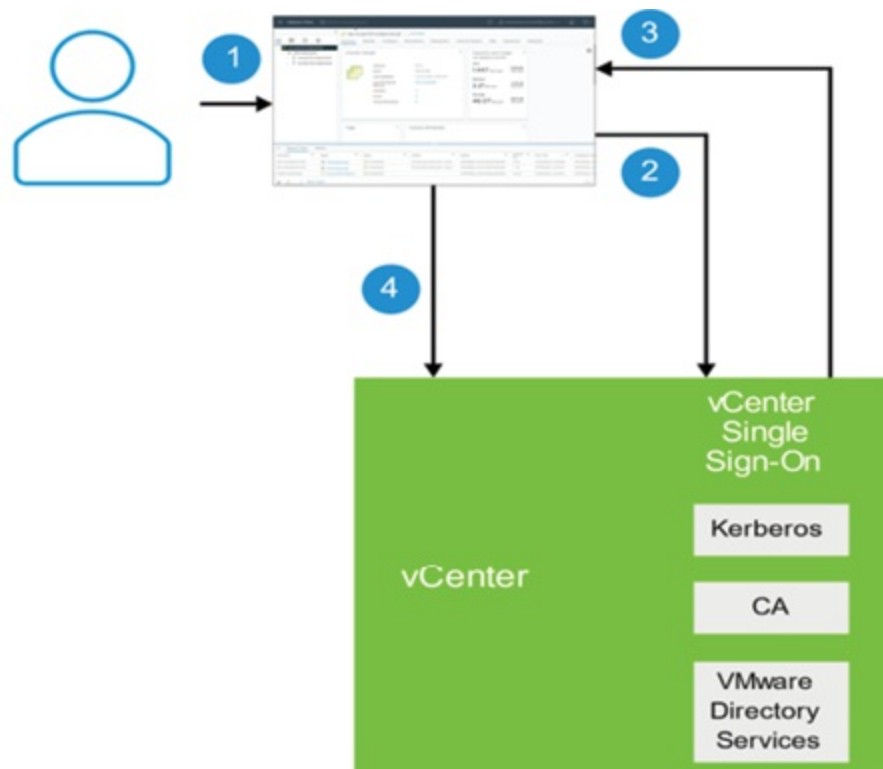


Figure 4.4: vCenter Single Sign-On

(Source: VMware)

For more information about configuring vCenter SSO and supported identity providers, refer to **VMware vSphere Authentication** at <https://techdocs.broadcom.com>.

Enhanced Linked Mode

Enhanced Linked Mode (ELM) allows administrators to manage multiple vCenter instances within a single vSphere environment seamlessly. Up to **15 vCenter instances** can be linked in one vCenter SSO domain; ELM simplifies the administration of distributed infrastructures.

The following are the key features of Enhanced Linked Mode:

- **Centralized login:** Users can log in to all linked vCenter instances simultaneously using a single username and password.
- **Unified inventory management:** Administrators can view, search, and manage the inventories of all linked vCenter instances through the vSphere Client.
- **Replication across instances:** ELM ensures consistent management by replicating roles, permissions, licenses, tags, and policies (e.g., storage policies) across all linked vCenter instances.
- **Flexible deployment options:**
 - Create an Enhanced Linked Mode group during the deployment of the vCSA.
 - Join an existing ELM group by moving or repointing a vCenter instance from one vSphere domain to another.
- **Licensing requirements:** Enhanced Linked Mode requires the *vCenter Standard license* and is not supported for vCenter Foundation or vCenter for Essentials editions.

For details on repointing vCenter instances and setting up Enhanced Linked Mode, refer to VMware vSphere documentation at <https://techdocs.broadcom.com>.

The following figure illustrates the vCenter Enhanced Linked Mode:

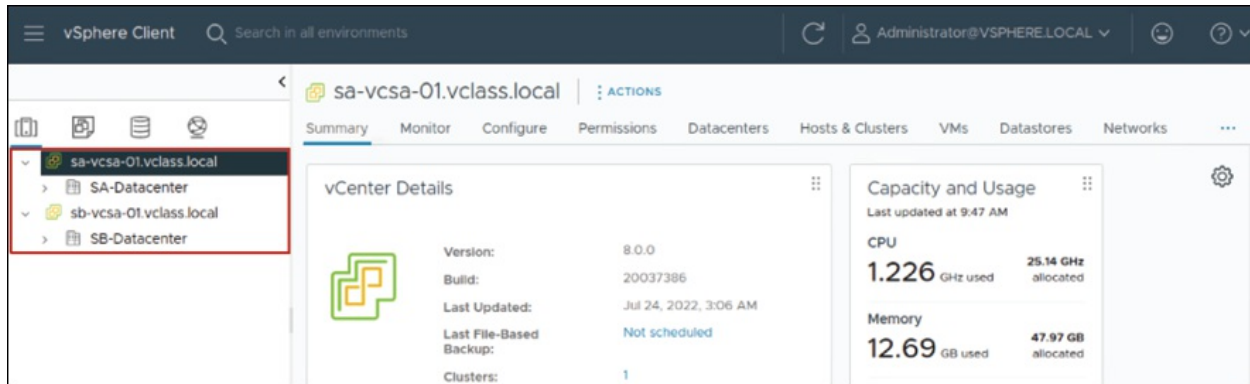


Figure 4.5: vCenter Enhanced Linked Mode

(Source: VMware)

Communication between vCenter and ESXi

Administrators can communicate with vCenter for centralized management through the vSphere Client, which is the main interface for controlling ESXi hosts. If vCenter is unavailable, administrators can connect directly to specific ESXi hosts using the VMware Host Client.

The following is the vCenter and ESXi communication flow:

1. **vCenter Agent (vpxa):** A vCenter agent known as vpxa is deployed and launched on an ESXi host when it is added to vCenter's inventory. The vCenter service (vpxd) and the ESXi host's management daemon (hostd) communicate with each other over vpxa.
2. **Host Management Daemon (hostd):** Running directly on the ESXi host, the *hostd* manages local ESXi operations including virtual machine creation, power state changes, vSphere vMotion migrations, and storage discovery (LUN and VMFS volumes). It also monitors the host's status, registered virtual machines, and visible storage volumes. Direct client connections, such as those from the VMware Host Client, are also handled by hostd.
3. **vpxd process:** Using vpxa, the vCenter's vpxd service transmits commands to the ESXi host. Creating, moving, or turning on virtual machines are a few examples of operations. The inventory is kept current since any modifications made via vCenter are mirrored in the vCenter database.

The following figure illustrates the communication flow between ESXi and

vCenter:

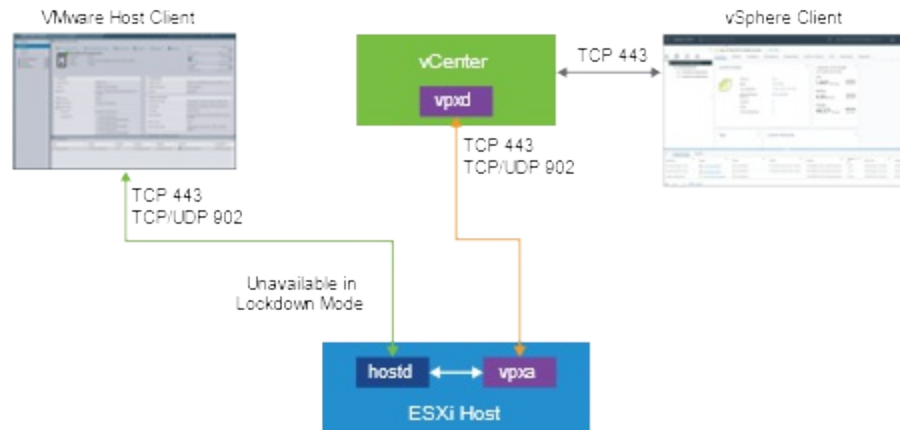


Figure 4.6: ESXi and vCenter Server Communication

(Source: VMware)

This architecture ensures that vCenter can efficiently manage multiple ESXi hosts and their associated resources while providing a fallback mechanism for direct host management when needed.

vCenter scalability

vCenter Server 8.0 is designed to accommodate large-scale enterprise environments, offering robust scalability to manage a wide range of virtualized resources effectively.

The following are the key scalability metrics:

Metric	vCenter 8.0
Hosts per vCenter instance	2,500
Powered-on VMs per instance	40,000
Registered VMs per instance	45,000
Hosts per cluster	96
Virtual machines per host	1024
Virtual machines per cluster	8,000
Linked vCenter Servers	15
	16

Maximum number of shared physical GPUs	
--	--

Table 4.1: Scalability metrics

This scalability enables organizations to efficiently manage their virtual infrastructure, from small-scale deployments to expansive enterprise environments. With these capabilities, vCenter ensures seamless management of even the most complex virtual environments, offering unparalleled performance and flexibility.

For up-to-date configuration limits and recommendations tailored to the specific needs, visit the VMware Configuration Maximums resource at <https://configmax.broadcom.com/home>.

Deployment of vCenter Server Appliance

Deploying the vCSA successfully requires careful planning. Before continuing, make sure the following actions are finished:

- Ensure that the vCSA installer is obtained from a verified and trusted source.
- Check that the vCSA satisfies all system requirements, including software and hardware compatibility.
- For the vCSA installation, get the host computer's static IP address or **fully qualified domain name (FQDN)**.
- Ensure the IP address and FQDN are reserved for the vCSA to deploy.
- Verify that all of the virtual machines in your vSphere environment have the same date and time settings. Time synchronization is essential for dependable operations and smooth communication.

Consult the official VMware guide on vCenter Server Installation and Setup for further information regarding system requirements at <https://techdocs.broadcom.com>.

Installer for vCenter Server Appliance Native GUI

The following features of the vCSA Native GUI installer make deployment easier:

- Offers an interactive deployment experience that walks the administrator

through the installation procedure step-by-step.

- Offering wide platform compatibility, it is a native application for Windows, Linux, and macOS.
- It carries out prechecks and automatic validations before deployment to find possible configuration problems and establish a compliant environment.

These validations and prechecks reduce errors and guarantee a seamless and effective deployment procedure. The GUI installer is the best option for those who want a user-friendly interface and comprehensive deployment instructions.

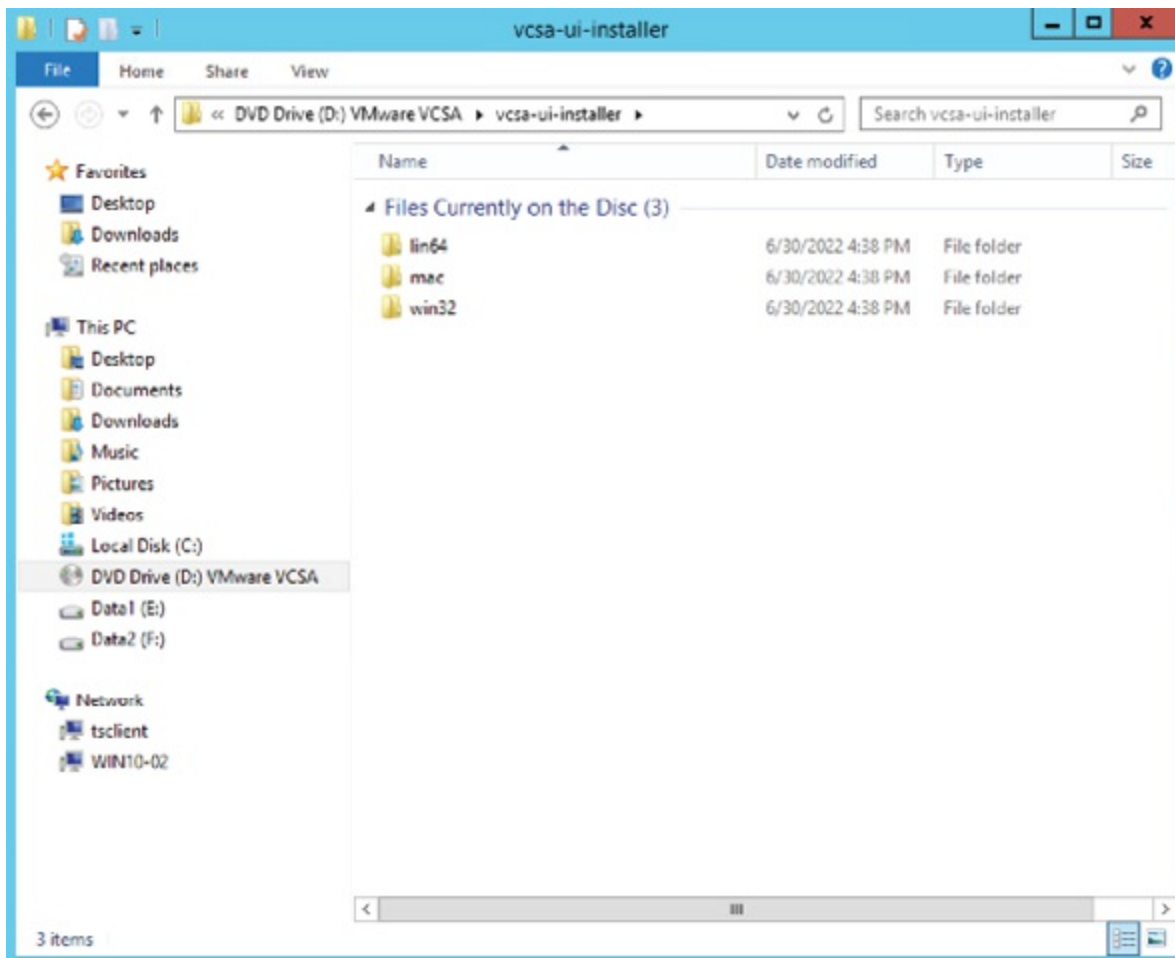


Figure 4.7: vCSA Native GUI installer

(Source: VMware)

vCenter Server Appliance installation

The installation of the vCSA follows a structured two-stage process to ensure proper deployment and configuration, as follows:

1. **Deployment of Open Virtualization Format (OVF):** First stage involves deploying the OVF file for the vCSA to an ESXi host. The deployment prepares the groundwork for the appliance and establishes its virtual machine structure.
2. **Configuration:** After the OVF deployment is finished, the second stage configures the vCSA. This comprises configuring the network, defining the database, and initializing the vCenter services.

Deployment automation

For repeated or large-scale deployments, the installation can be completely automated using JSON templates and the CLI installer, which is compatible with Windows, Linux, and macOS.

The vCSA installer offers the following four options to accommodate different deployment scenarios:

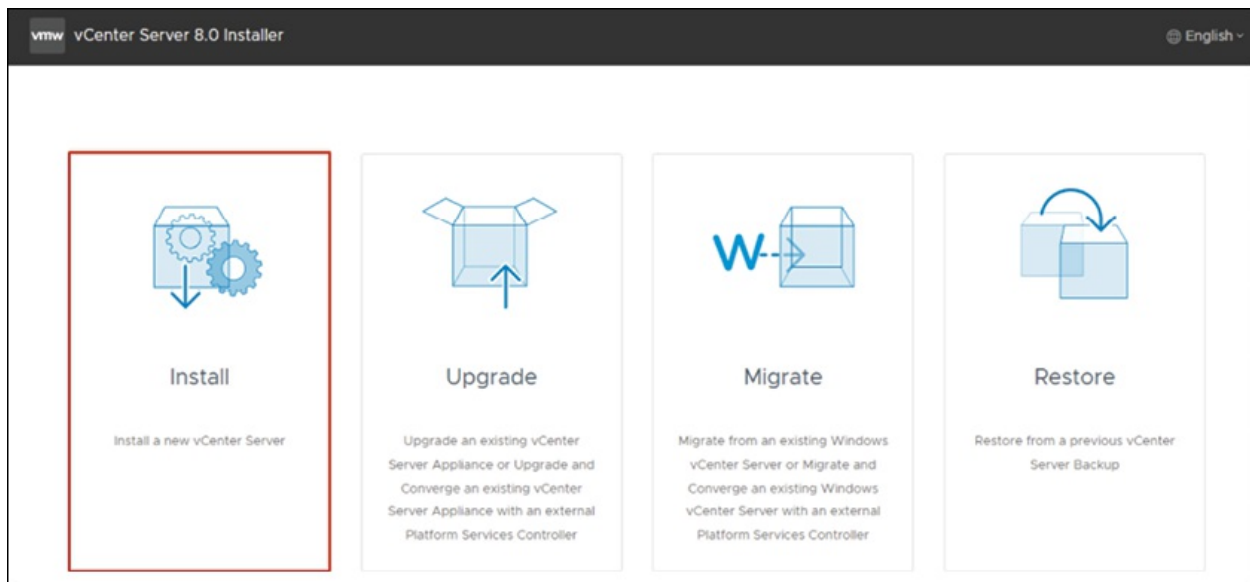


Figure 4.8: vCSA installation

(Source: VMware)

- **Install:** This option is used to deploy a new vCSA instance.
- **Upgrade:** Refreshes an existing vCSA or incorporates an external **Platform Services Controller (PSC)** into a converged environment.

- **Migrate:** Converts an existing Windows-based vCenter instance to a vCSA, which can optionally integrate with an external PSC.
- **Restore:** Restores a vCSA instance from a backup.

These versatile options make the vCSA installation flexible and responsive to various organizational requirements.

vCenter Server Appliance installation

The first stage of installing the vCSA focuses on deploying the appliance via an interactive UI and following the deployment phase:

- **UI phase:** During this step, the administrator will provide important configuration parameters to prepare the environment for the appliance deployment.
 - **Accept the EULA:** Review and accept the End User License Agreement.
 - **Connect with the Target Host:** Specify the ESXi host or vCenter system where the appliance will be installed.
 - **Define appliance details:** Assign a name to the vCSA and configure the root password for administrative access.
 - **Select resources:** Select the compute size (CPU and RAM), storage size, and datastore location, with the option of thin disc provisioning.
 - **Set the networking settings:** Configure network parameters such as IP address, subnet mask, gateway, and DNS information.
- **Deployment phase:** Once the UI phase is finished, the deployment phase starts:
 - **OVF deployment:** The OVF file is installed on the target ESXi host or vCenter environment.
 - **Disc and network configuration:** The appliance's virtual discs are initialized, and network settings are applied to prepare it for use.

The following figure illustrates the stage 1 of vCSA installation wizard:

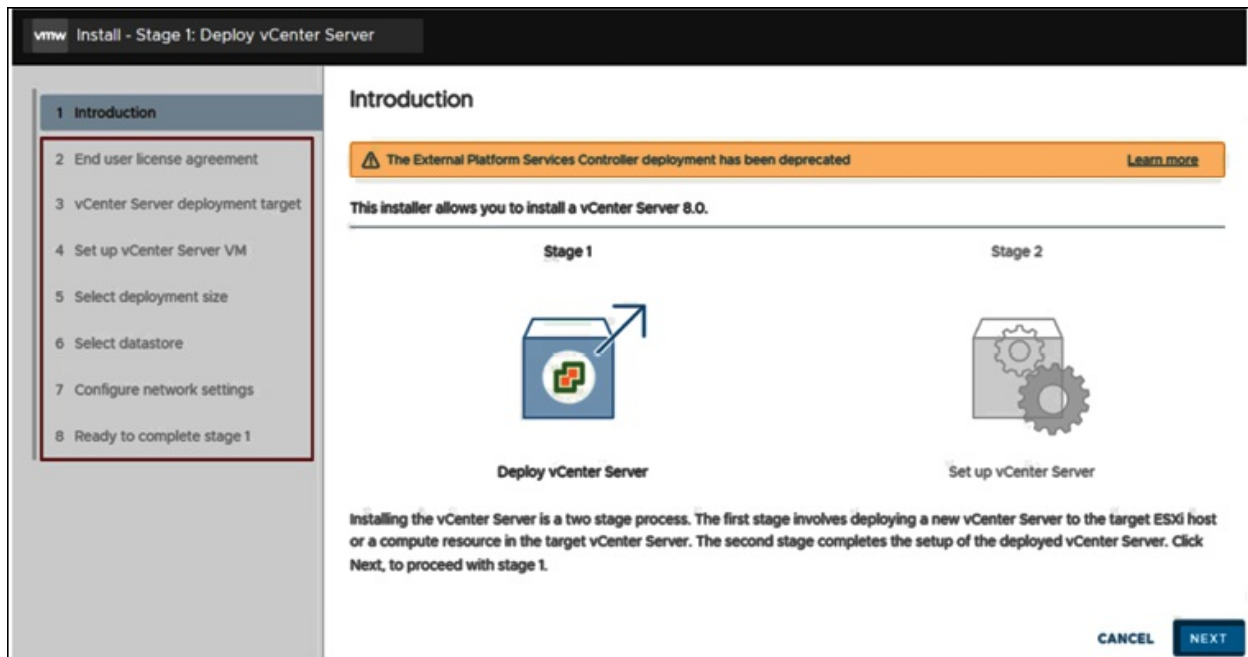


Figure 4.9: vCSA installation stage 1

(Source: VMware)

This stage guarantees that the vCSA is properly installed on your infrastructure and ready for additional setup in stage 2.

vCenter Server Appliance installation

The second stage of the vCSA installation focuses on configuring the appliance to function within your vSphere environment. This *configuration phase* establishes essential settings to prepare the appliance for production use.

The following are the key configuration steps:

1. **Time synchronization:**

- a. Define the source for time synchronization to ensure consistent timestamps across the vSphere environment.
- b. Options include using the ESXi host or specifying NTP servers.
- c. Accurate time synchronization is critical for log management, performance monitoring, and secure communications.

2. **SSH access:**

- a. Choose whether to enable SSH access for secure remote troubleshooting and management.

- b. By default, SSH is disabled to enhance security but can be activated as required.
- 3. **vCenter SSO:**
 - a. Create a new SSO domain (e.g., vsphere.local) to establish centralized authentication for vSphere components.
 - b. Alternatively, join an existing SSO domain to integrate with an existing infrastructure.
- 4. **Customer Experience Improvement Program (CEIP):** Decide whether to participate in the CEIP, which helps VMware improve its products and services by collecting anonymized data.

Once the configuration is complete, the vCSA is fully operational and ready to manage the vSphere environment, providing advanced features such as centralized management, resource pooling, and enhanced scalability. This phase ensures the appliance is tailored to meet the specific needs of your infrastructure.

The following figure illustrates the stage 2 of vCSA installation wizard:

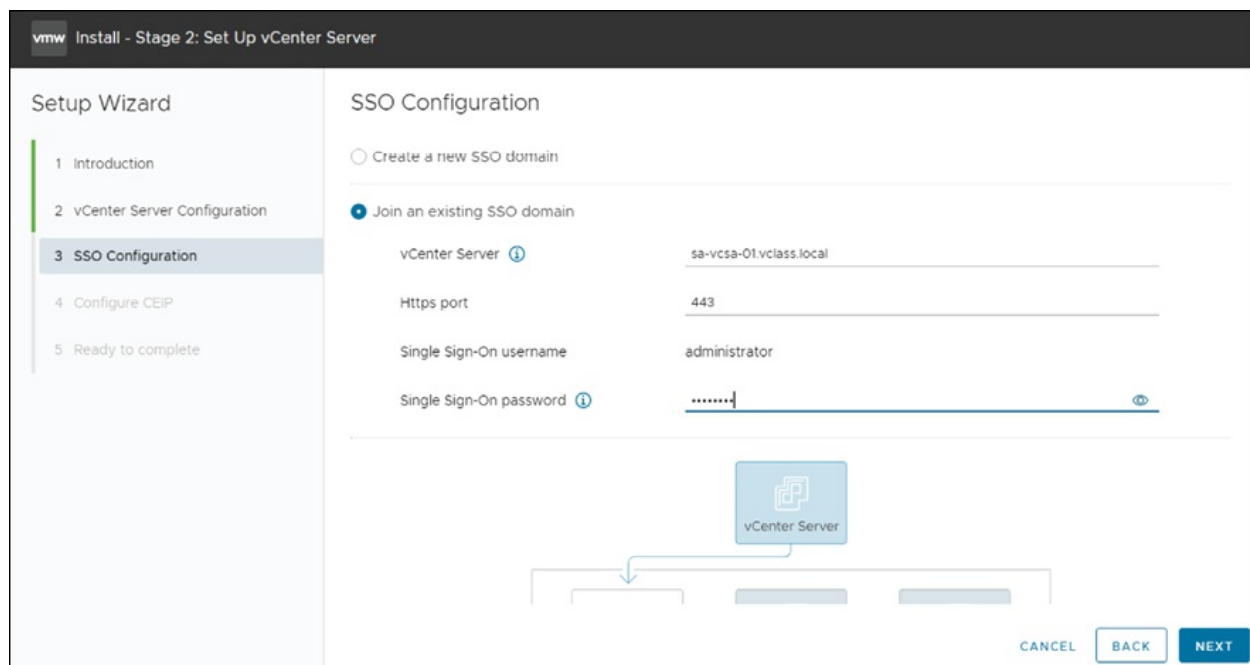


Figure 4.10: vCSA installation stage 2

(Source: VMware)

Getting started with vCenter

Once the vCSA has been successfully deployed and configured, you can begin managing your virtualized infrastructure using the vSphere Client. The vSphere Client is a web-based interface that provides centralized control of vCenter inventory, including ESXi hosts, virtual machines, and other resources.

The following are the steps to Access vCenter:

1. Open a supported web browser.
2. Navigate to the vSphere Client URL, **https://<vCenter_FQDN_or_IP_address>/ui.**
3. Enter your vCenter SSO credentials to log in.

The following figure illustrates the homepage of vCenter Client:

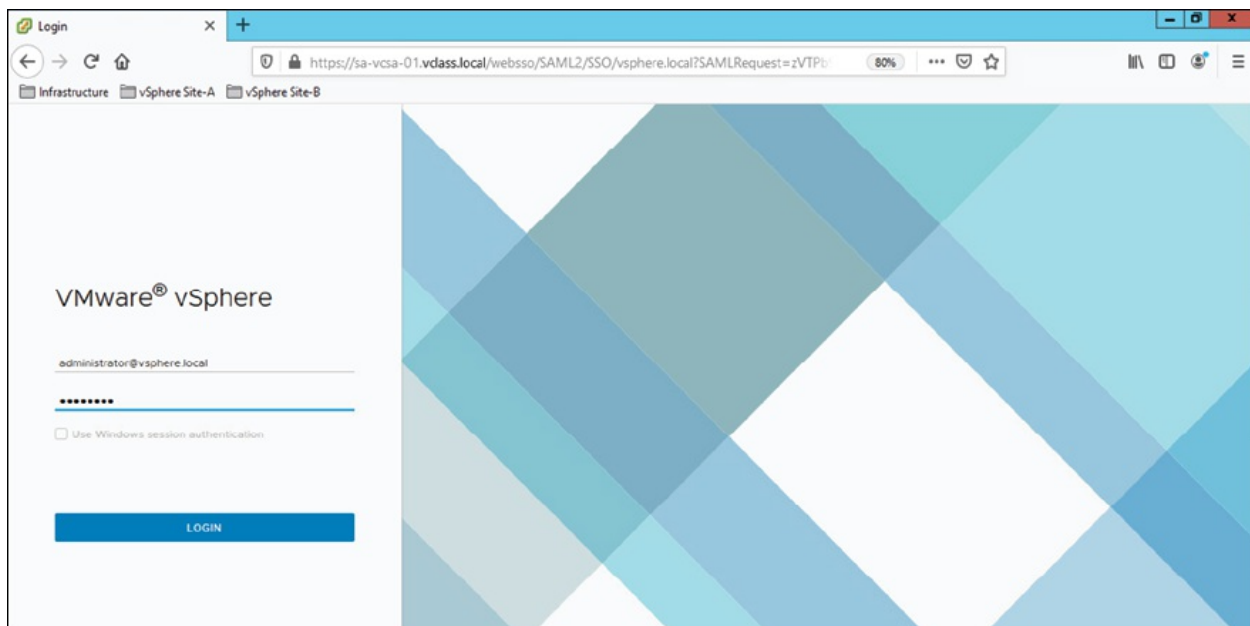


Figure 4.11: Getting started with vCenter Server

(Source: VMware)

Configuring essential vCenter settings

The vSphere Client provides administrators with centralized control to configure and manage key vCenter settings, ensuring an optimized and

efficient virtualized environment.

The following are the key management capabilities:

- **Inventory management:** Organize and manage ESXi hosts, clusters, virtual machines, datastores, and networks, ensuring streamlined resource allocation and organization.
- **Resource monitoring:** Monitor infrastructure performance and utilization metrics to maintain efficiency and prevent resource bottlenecks.
- **Task automation:** Automate essential operations such as creating, cloning, or migrating virtual machines to save time and reduce manual effort.
- **Centralized configuration:** Set up licenses, manage permissions, and apply policies consistently across your environment.

Using the vSphere Client, administrators can fine-tune vCenter settings, such as licensing, statistics collection, and logging.

The following are the steps to access vCenter system settings:

1. Open the **vSphere Client** and log in with your vCenter SSO credentials.
2. In the navigation pane, select the **vCenter Server system**.
3. Click the **Configure** tab in the main interface.
4. Expand the **Settings** section to view and modify system configurations.

Through these capabilities, administrators can leverage the full potential of vCenter to simplify infrastructure management while enhancing performance and reliability, as shown:

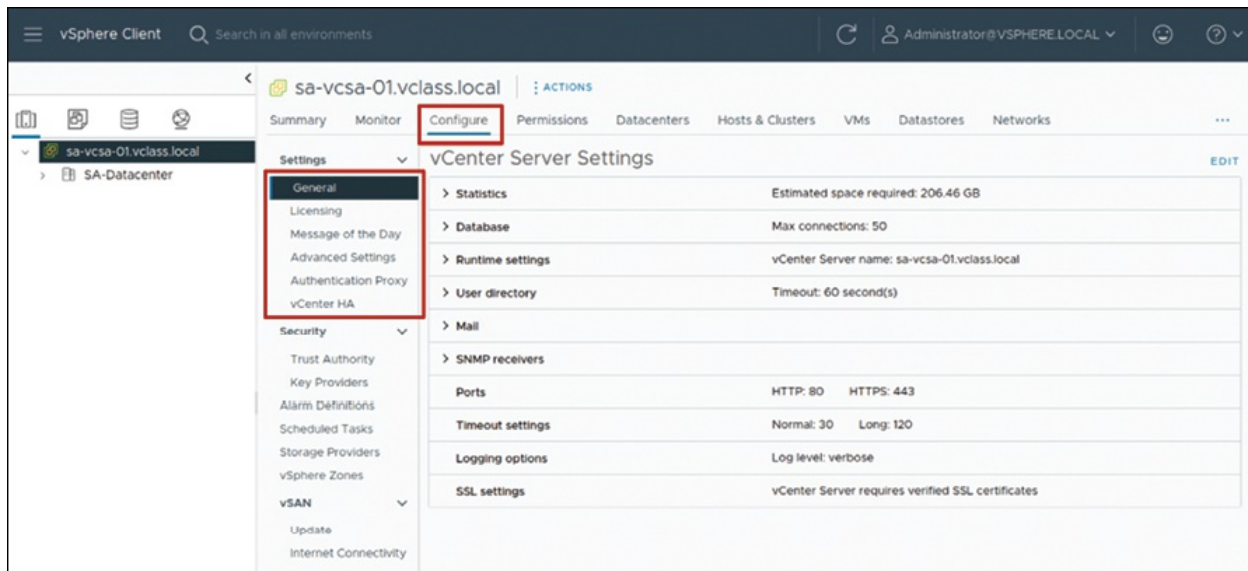


Figure 4.12: Configure vCenter using vSphere Client

(Source: VMware)

vCenter Management Interface

The **vCenter Management Interface (VAMI)** is an HTML-based client designed to help administrators configure and monitor their vCSA.

The following are the key tasks administrator can perform with VAMI:

- **Monitoring appliance resources:** Track the CPU, memory, and storage usage of your vCSA to ensure optimal performance.
- **Backing up the appliance:** Configure and manage backups to safeguard your vCenter configurations and data against potential failures.
- **Monitoring vCenter services:** Check the status of vCenter services and restart them as needed to maintain operational stability.
- **Adding network adapters:** Add and configure additional network adapters to extend the appliance's connectivity or improve redundancy.
- **Updating vCSA:** Manage updates and patches directly from the VAMI to keep your appliance secure and up-to-date.
- **Configuring time settings:** Synchronize the appliance's time settings using NTP servers to maintain consistency across your vSphere environment.

Access the vCenter Management Interface as follows:

- **URL:** Use **https://<FQDN_or_IP_address>:5480** to access the VAMI.
- **Port:** The interface connects directly to port **5480** on the vCSA.

By using the vCenter Management Interface, administrators gain a centralized and efficient way to monitor and maintain vCenter instances and ensure a robust and well-performing virtualized infrastructure.

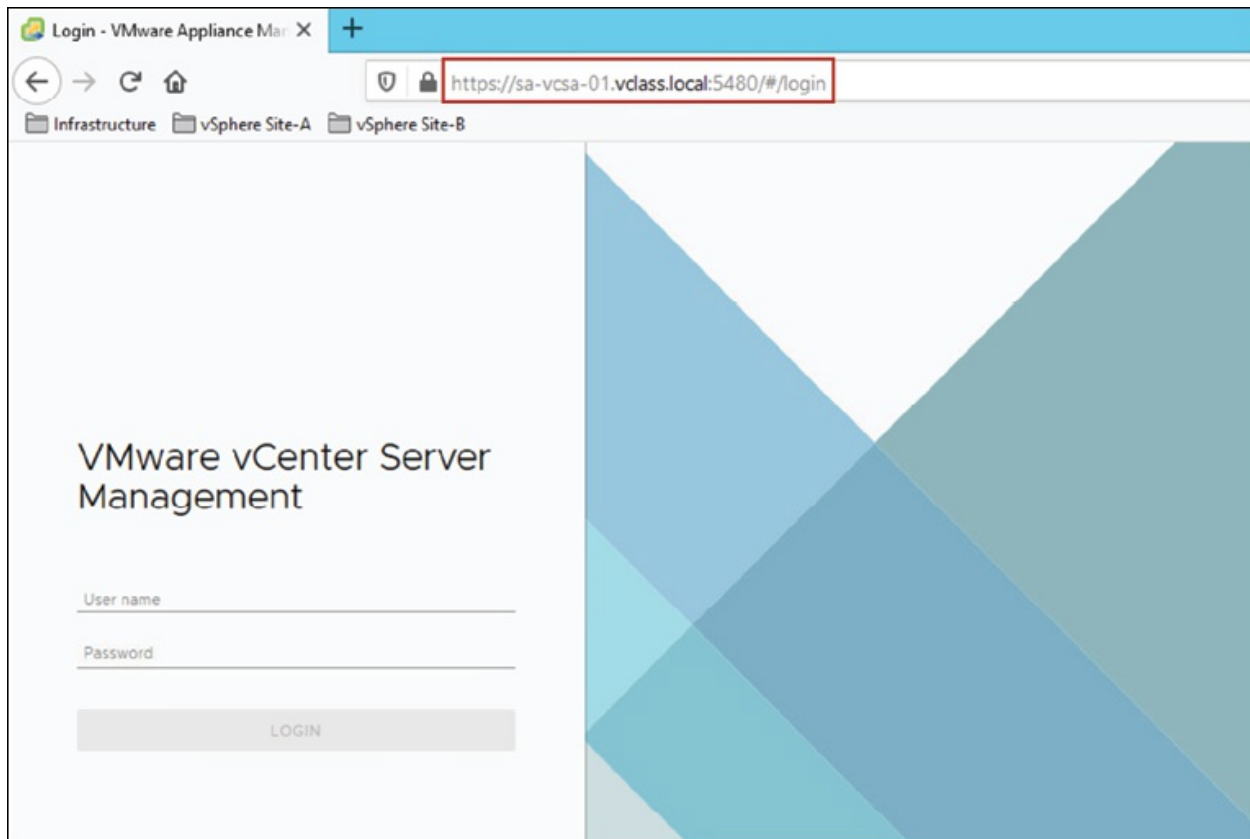


Figure 4.13: vCSA Management Interface

(Source: VMware)

Multi-homing the vCSA

Multi-homing in vCSA allows you to configure multiple NICs to manage distinct types of network traffic, such as management, storage, or backup. This feature enhances the flexibility and efficiency of your network infrastructure while maintaining a clear separation between different traffic types.

The following are the key features:

- **Support for multiple NICs:** The vCSA supports up to four NICs for

handling diverse network requirements.

- **Preservation during maintenance:** NIC configurations are preserved across upgrade, backup, and restore processes, ensuring seamless operation without requiring additional reconfiguration.

The following are the benefits:

- **Traffic segregation:** Multi-homing allows for logical separation of network traffic, ensuring that management traffic does not interfere with storage or backup operations.
- **Improved network performance:** Multi-homing prevents bottlenecks and ensures optimal resource usage by distributing network loads across multiple NICs.
- **Enhanced security:** Isolating traffic types reduces the risk of unauthorized access, helping meet stringent compliance and security policies.

Multi-homing configurations are particularly beneficial for organizations seeking to optimize and scale their virtualized network environment. By effectively managing network segmentation and traffic distribution, this capability ensures a stable and secure vSphere environment.

The following figure illustrates the vCenter multihoming feature:

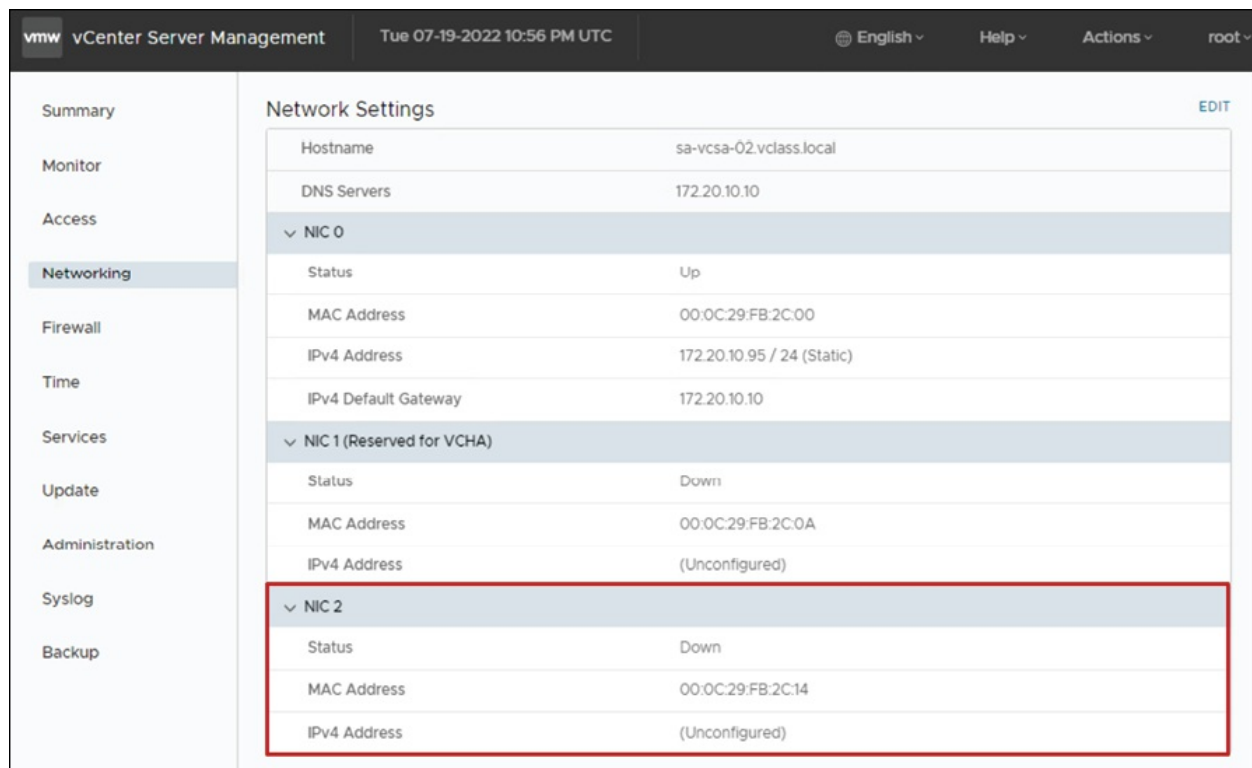


Figure 4.14: vCenter Appliance Multihoming

(Source: VMware)

Managing license keys with the vSphere Client

VMware offers many vSphere licensing models designed specifically for use by organizations, ranging from small to large enterprise setups. The knowledge about the existing licensing models is essential for efficient deployment and management of vSphere.

VMware has provided various licensing models with vSphere 8.0 to suit various organizational requirements:

- **Licensing for ESXi hosts:** ESXi hosts require vSphere licenses, with each license providing specific capacity for licensing physical CPUs on ESXi hosts. Broadcom offers four main licensing modes:
- **Per core licensing model (Subscription-Based):**
 - **Minimum requirement:** 16 cores licensed per CPU
 - **Calculation:** Based on total physical CPU cores across all ESXi hosts

- **Example:** 1 ESXi host with 1 CPU having 8 cores requires purchasing 16 core licenses (minimum)
- **Per virtual machine licensing model:**
 - **Application:** Primarily for **Virtual Desktop Infrastructure (VDI)** environments
 - **Calculation:** Based on total number of powered-on desktop virtual machines
 - **Use case:** Organizations running virtual desktop workloads
- **vSphere+ subscription capacity-based licensing:**
 - **Model:** Cloud-integrated subscription service
 - **Features:** Combines on-premises vSphere with cloud capabilities
 - **Management:** Centralized through VMware Cloud Console
- **Per CPU licensing model (Legacy):**
 - **Coverage:** One CPU license covers up to 32 cores per CPU
 - **Additional licenses:** Required for CPUs exceeding 32 cores
 - **Example:** 1 CPU with 40 cores requires 2 CPU licenses

License assignment requirements

To properly license an ESXi host using vSphere license:

- **Sufficient capacity:** License must have adequate capacity based on the licensing model
- **Feature support:** License must support all features the host uses. For **instance**, vSphere Distributed Switch demands proper license tier
- **Compatibility:** License version must match or exceed vSphere version
- **Licensing for vCenter server:** Administrators have two options for licensing vCenter Server systems, either use a vCenter Server license with per-instance capacity or apply a Solution License that covers multiple components together.

For detailed licensing information and calculations, go to the official site at:

<https://techdocs.broadcom.com/us/en/vmware-cis/vsphere/vsphere/8-0/vcenter-and-host-management-8-0/license-management-host-management/licensing-for-products-in-vsphere-host-management.html>

Let us look at the licensing of VMware components.

Licensing VMware vSphere is a straightforward two-step process that assures your components are active and ready to go:

1. **Add a license to the vCenter License Service:** First, add vSphere licensing keys to the vCenter Licensing Service. This centralizes license management, making it easy to manage throughout your vSphere environment.
2. **Assign the license to the components:** Once license is in place, the next step is to assign it to the required components, which include:
 - a. **ESXi hosts:** This grants hosts access to features such as vSphere HA and vMotion.
 - b. **vCenter instances:** This is where administrators enable the central administration functions for your infrastructure.
 - c. **Other vSphere components:** This includes components like vSAN and vSphere Trust Authority.

Following these procedures ensures that everything is properly licensed and running as intended. Check that the licenses are compatible with the features and capacity that environment requires.

vSphere License Service

The **vSphere License Service** manages all VMware vSphere licenses centrally. It provides an inventory of vSphere licenses and assigns them to components like **ESXi hosts**, **vCenter systems**, and **vSAN clusters**. It also handles license management for integrated products such as **Site Recovery Manager**.

You can monitor the health and status of the vSphere License Service using the *vCenter Management Interface* for a streamlined overview of license performance and compliance.

vmw vCenter Server Management

Fri 07-08-2022 11:07 PM UTC

English

Help

Actions

root

Summary

Monitor

Access

Networking

Firewall

Time

Services

Update

Administration

Syslog

Backup

RESTARTSTARTSTOPSET STARTUP TYPE

	Name	Startup Type	Health	State
<input type="radio"/>	Appliance Management Service	Automatic	Healthy	Started
<input type="radio"/>	Auto Deploy	Manual		Stopped
<input type="radio"/>	Content Library Service	Automatic	Healthy	Started
<input type="radio"/>	Envoy Host Gateway	Automatic	Healthy	Started
<input type="radio"/>	Envoy Sidecar Proxy	Automatic	Healthy	Started
<input type="radio"/>	Hybrid vCenter Service	Automatic	Healthy	Started
<input type="radio"/>	ImageBuilder Service	Manual		Stopped
<input type="radio"/>	License Service	Automatic	Healthy	Started
<input type="radio"/>	Service Control Agent	Automatic	Healthy	Started
<input type="radio"/>	vAPI Endpoint	Automatic	Healthy	Started
<input type="radio"/>	vCenter Server Profiles	Automatic	Healthy	Started
<input type="radio"/>	VMware Analytics Service	Automatic	Healthy	Started

Figure 4.15: vSphere License Service

(Source: VMware)

Adding license keys to vCenter

Before the 60-day evaluation period expires, you need to assign a license to the vCenter. To do this, go to the *vSphere Client* and select **Administration | Licenses** from the main menu to open the **Licenses** pane.

vSphere centralizes license management, where each product and feature is associated with a 25-character license key. Readers can view the license details in terms of product, license key, or asset, as follows:

- **Product:** Refers to the vSphere software component or feature, such as **vCenter** or **vSphere Enterprise Plus**.
- **License key:** The serial number tied to the product.
- **Asset:** The component assigned a product license to run legally.

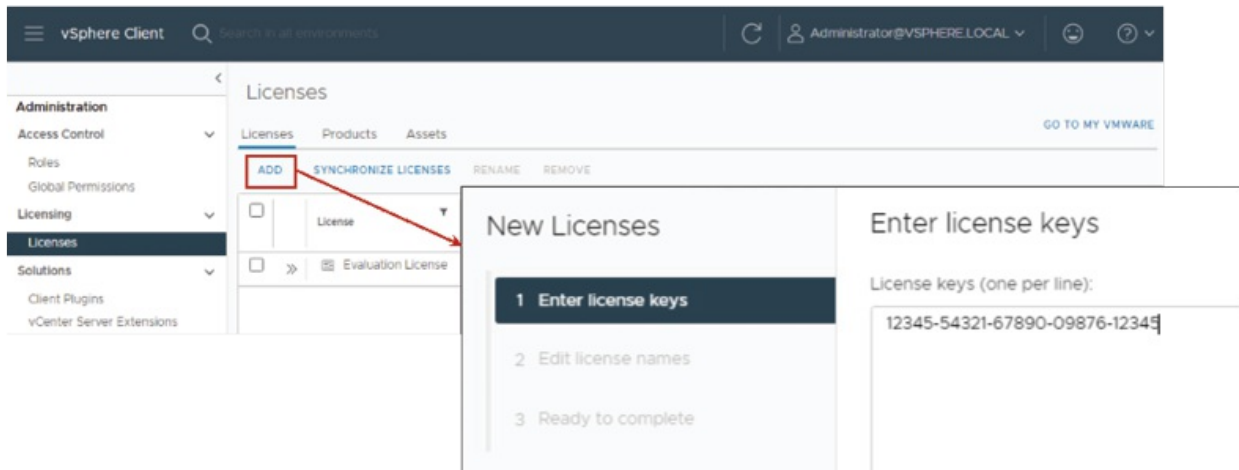


Figure 4.16: Adding License to vCenter

(Source: VMware)

The following is the process to assign a license to an asset:

- **Navigate to licensing:** From the main menu, select **Administration | Licenses | Assets**.
- **Assign license:** Click on **Assign License**. Enter the 25-character license key for the desired product or feature and click **OK**.
- **Verify license:** After assigning the license, check the **Licenses** pane to make sure the licensing information is correct. Readers may see the product, licensing key, and assigned asset.

This procedure guarantees that the vCenter (or any other asset) is appropriately licensed, as shown:

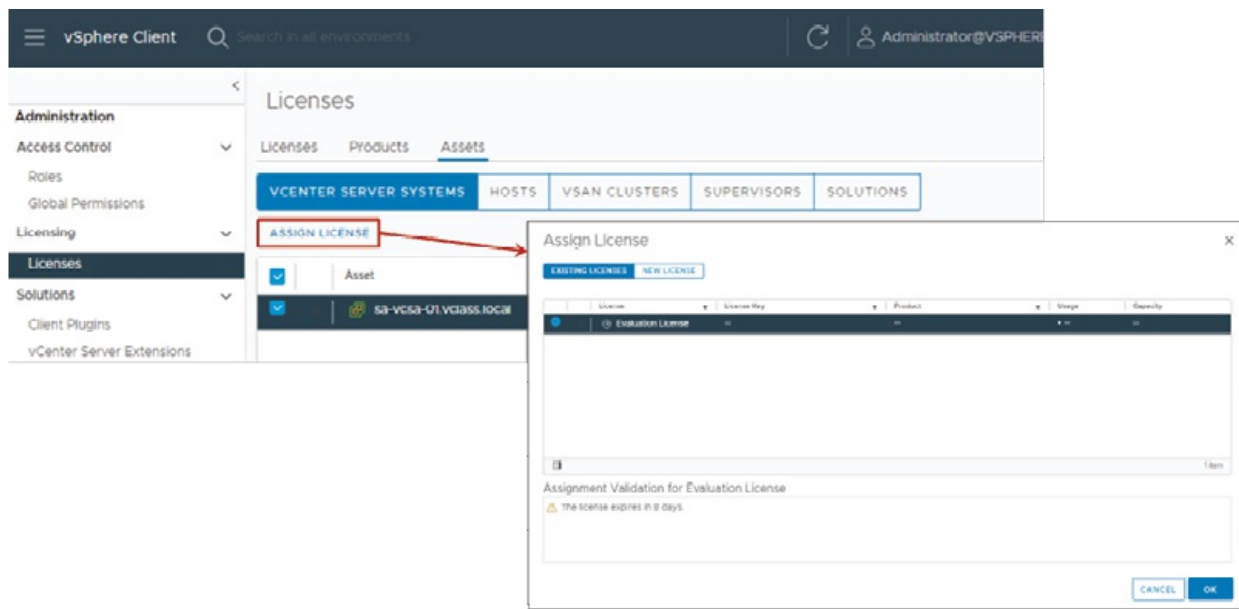


Figure 4.17: Assigning License to vCenter

(Source: VMware)

Viewing licensed features

To manage licenses in vCenter, select the License pane from the **Configure** menu. This section contains thorough information on the license type and the features available under it.

When you install ESXi or vCenter, the software starts in evaluation mode by default. The evaluation mode allows you to test the software's full set of features and functionality. It is useful for showcasing or testing the program before making a purchase.

The evaluation period lasts 60 days following the initial installation. During this time, the software functions normally, but it will periodically alert you of the remaining time before the evaluation period expires. Keep in mind that the 60-day period cannot be interrupted or restarted.

If the evaluation term expires without a valid license, important operations in ESXi and vCenter will be unavailable. For example, you will be unable to turn on or reset virtual machines, and all ESXi hosts will be disconnected from the vCenter system. To assure continued access to all features and operations, purchase, install, and assign valid license keys before the evaluation time expires, as follows:

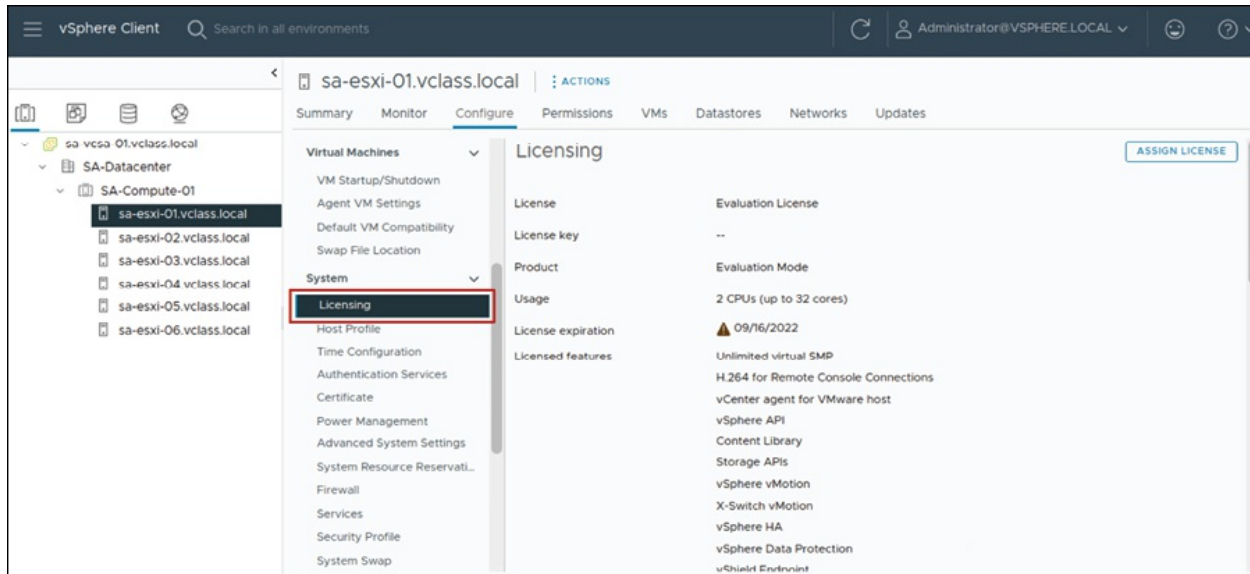


Figure 4.18: Viewing Licensed Features

(Source: VMware)

Organizing vCenter inventory objects

The **vSphere Client** main menu serves as the central hub for managing your vCenter system inventory, overseeing the infrastructure environment, and performing administrative tasks.

You can access the main menu by clicking the *three-lined icon* (often referred to as the *hamburger menu*) located in the upper-left corner of the vSphere Client window. From this menu, administrators can easily navigate to essential management and configuration sections, streamlining the operation of the virtualized environment, as follows:

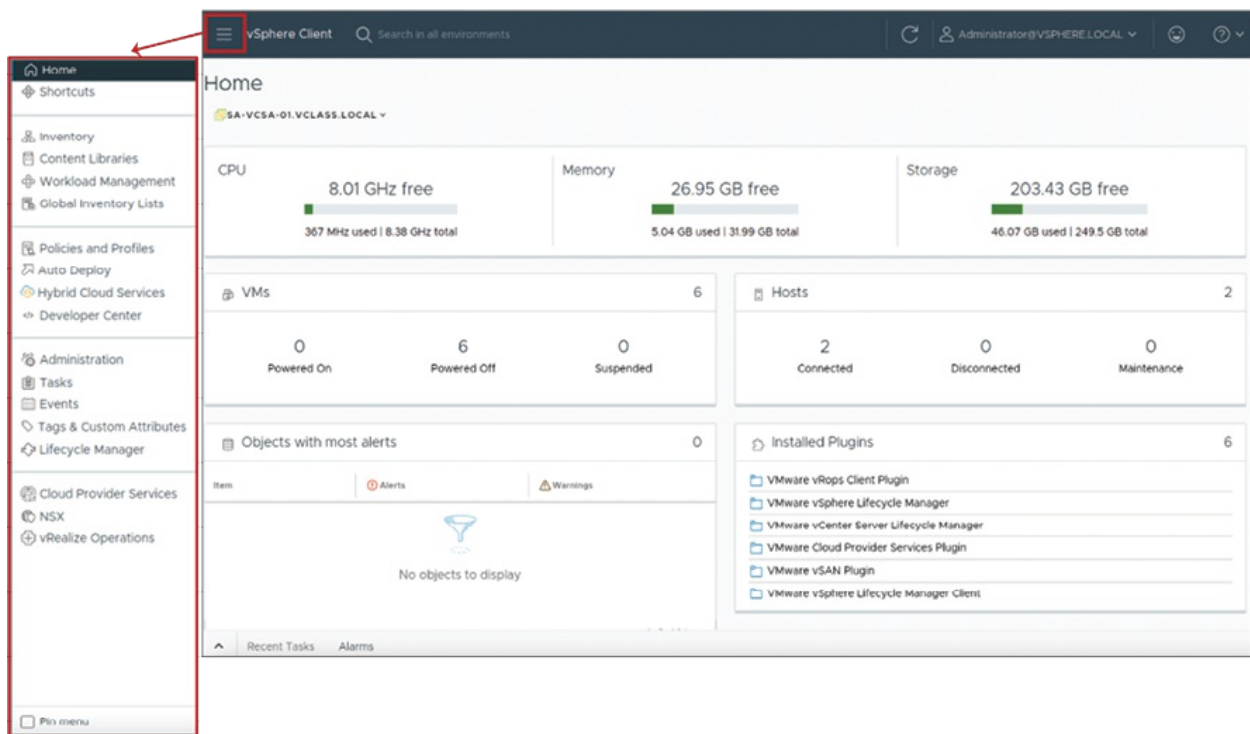


Figure 4.19: vSphere Client main menu

(Source: VMware)

Navigating the inventory

The vSphere Client's **navigation pane** allows you to easily browse and manage objects in vCenter inventory. Administrators can rapidly access specific configurations and perform administrative activities by selecting entities like data centers, clusters, hosts, or virtual machines from the UI. This improved navigation allows for efficient oversight of the virtualized environment, as follows:

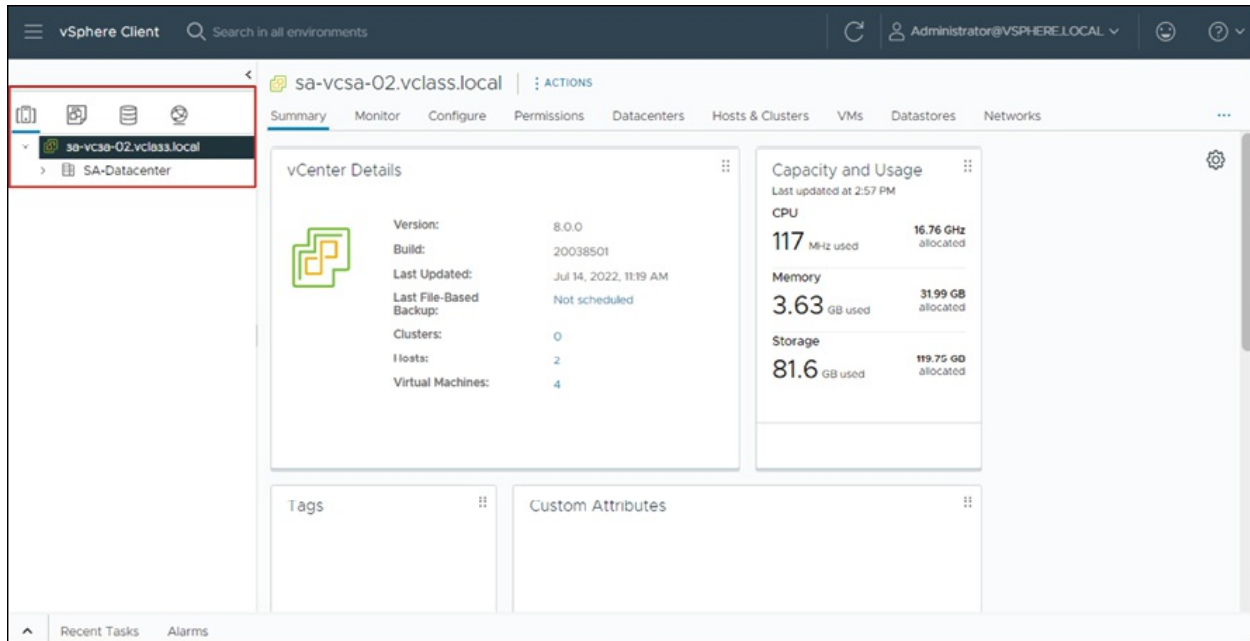


Figure 4.20: Navigating the Main Menu

(Source: VMware)

The vSphere Client provides distinct inventory views of Hosts, Clusters, VMs, and Templates for efficient management:

- **Hosts and clusters view:** Displays all host and cluster objects in a data center. Administrators can organize these objects into folders for better structure and ease of management.
- **VMs and templates view:** Showcases all virtual machine and template objects in a data center. Like hosts and clusters, VMs and templates can also be grouped into folders for streamlined organization and quick access.

These views help administrators efficiently navigate and manage their virtualized infrastructure.

The following figure illustrates the inventory view of vCenter:

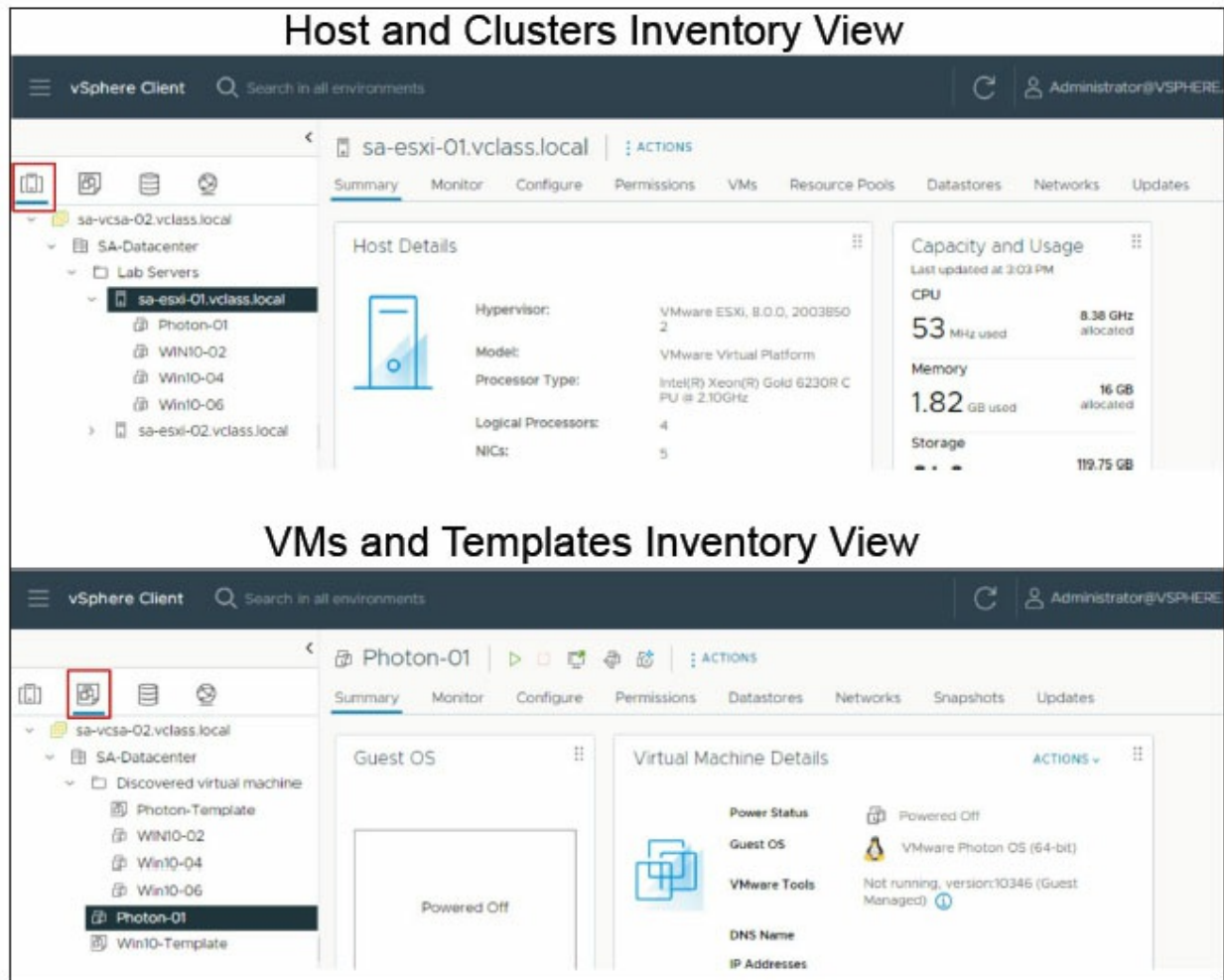


Figure 4.21: Inventory View of Host/Cluster/VMs/Templates

(Source: VMware)

The following are the views for storage and networks:

- **Storage inventory view:** Displays all datastores within the data center, providing detailed information about storage resources.
- **Networking inventory view:** Lists all port groups associated with standard and distributed switches in the data center.

Like other inventory views, datastore and network objects can be grouped into folders for better organization and simplified management. These views enhance visibility and control over storage and network configurations in your virtualized environment.

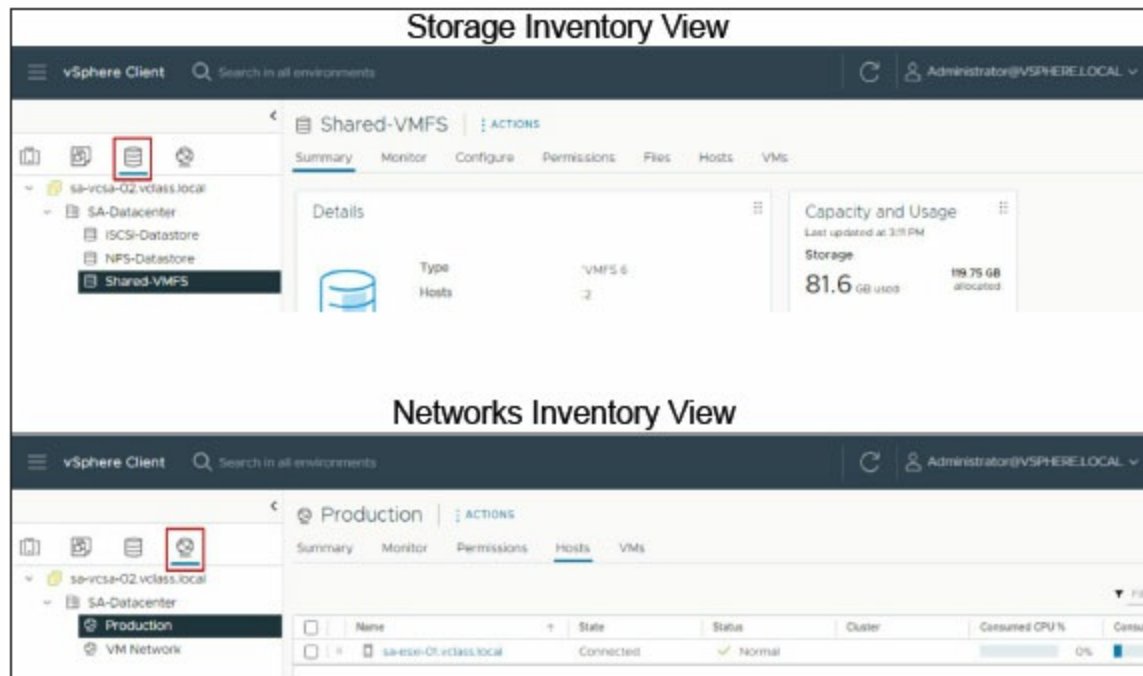


Figure 4.22: Inventory View Storage and Network

(Source: VMware)

Viewing object information

The vSphere Client simplifies monitoring and managing object properties by allowing readers to examine comprehensive information and access associated objects. This streamlined interface allows administrators to easily track and adjust item properties as needed, hence improving overall management capabilities, as shown:

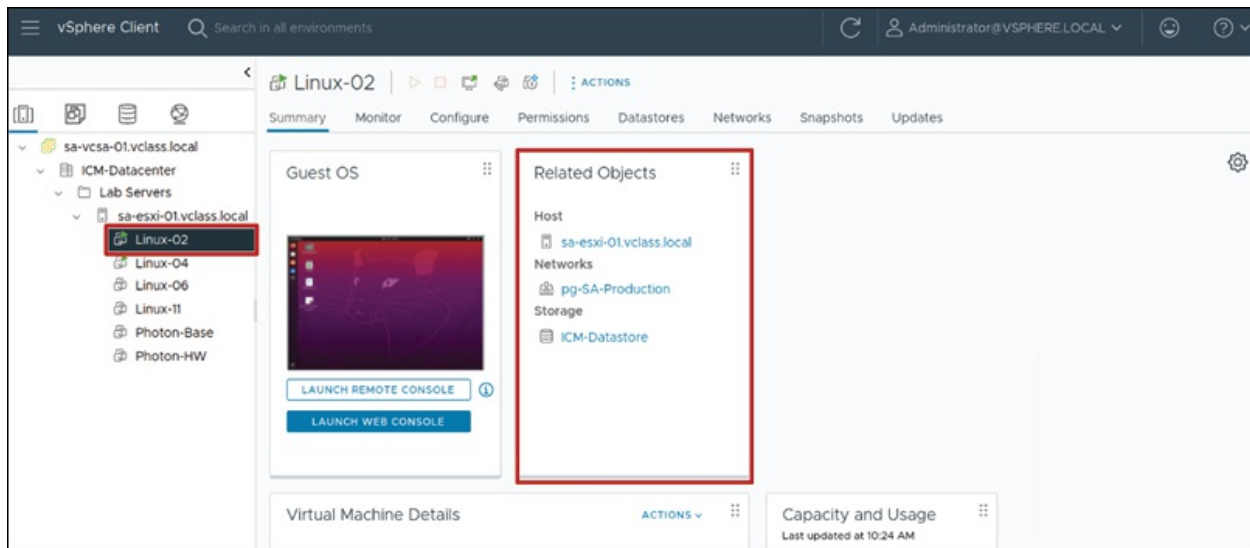


Figure 4.23: Viewing object information

(Source: VMware)

About data center objects

A virtual data center serves as a logical container for organizing all inventory objects essential to creating a fully operational environment for virtual machines. These objects include hosts, VMs, templates, datastores, and networks.

Administrators can create multiple data center objects to manage and segment environments effectively. For instance, readers might create a data center object based on geographical locations or align it with organizational units within your enterprise. Each data center operates independently, allowing for tailored configurations and resource management as follows:

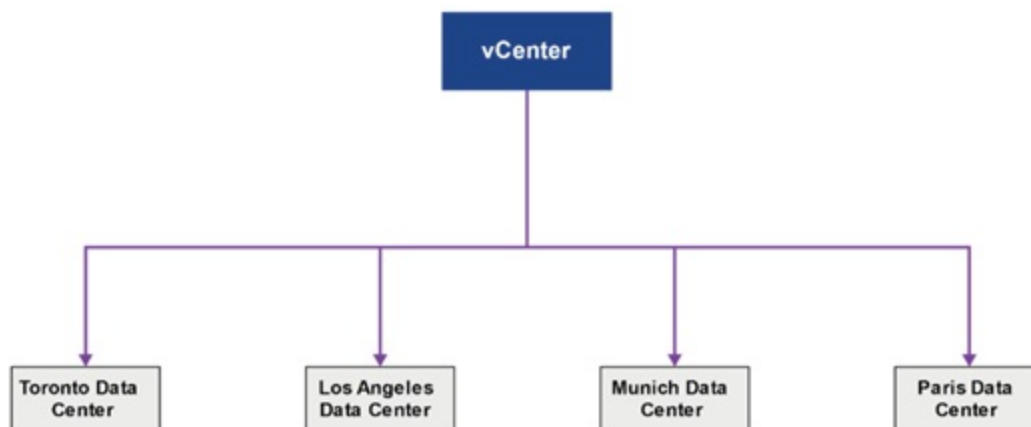


Figure 4.24: About data center objects

(Source: VMware)

Sorting inventory objects into folders

vSphere allows the administrators to use folders to organize inventory objects within a data center. This hierarchical structure enables effective resource management and grouping. Each inventory view has its own folder structure, allowing for the personalized organization of each resource category.

The setup and organization of the virtual environment depends on its size and complexity:

- *Large implementations* may comprise several virtual data centers, clusters, and a diverse set of hosts, resource pools, and networks. These configurations may span many vCenter systems and necessitate careful planning for scalability and performance.
- *Smaller implementations* often use a single virtual data center with a simpler topology and fewer resources to maintain.

Regardless of the implementation size, the following tasks are required for successfully organizing and managing the inventory:

- **Creating data centers:** Logical groups of hosts, virtual machines, and associated resources.
- **Building clusters:** Combine the resources of several hosts and virtual machines.
- **Adding hosts:** Install ESXi hosts in clusters or directly into data centers.
- **Folder organization:** Group inventory items into folders and subfolders for easier administration.
- **Network configuration:** For reliable networking, use vSphere standard or distributed switches.
- **Configuring storage:** Use datastore inventory objects to manage and deploy storage devices more efficiently.

By carefully organizing inventory objects, you ensure that your virtualized environment runs more smoothly, has greater scalability, and performs optimally:

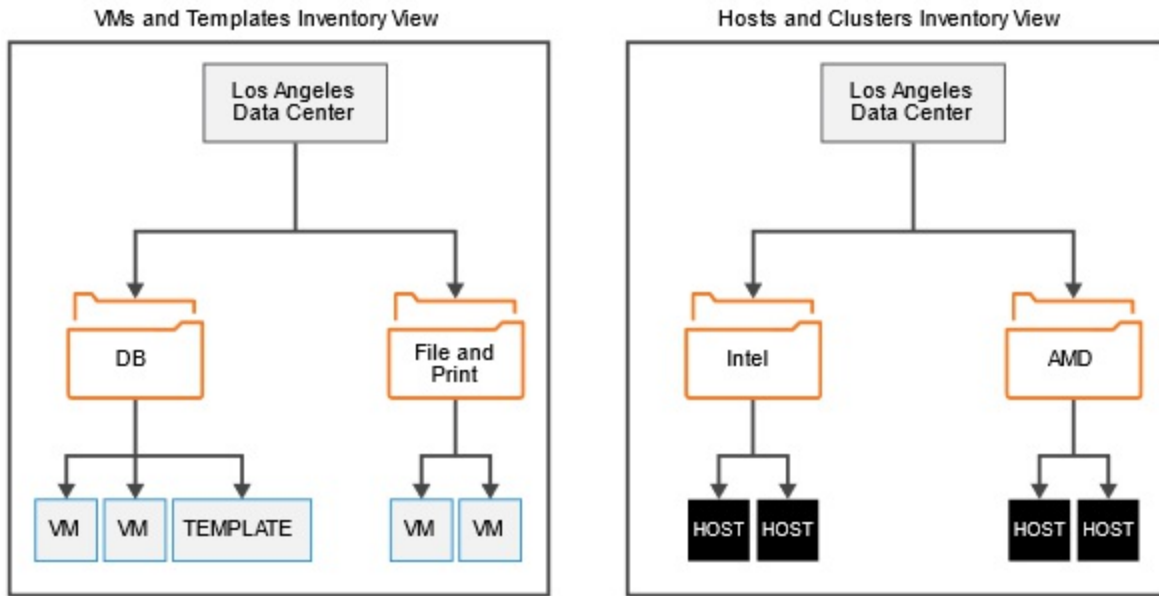


Figure 4.25: Organizing inventory objects

(Source: VMware)

Adding a data centre and organizational objects to vCenter

In vCenter, you can add essential organizational objects such as data centers, hosts, clusters, and folders to structure your virtual environment effectively:

- **Data center:** Represents a logical container for organizing all inventory objects, including hosts, VMs, networks, and storage.
- **Host:** Adds physical compute resources to your virtual infrastructure.
- **Cluster:** Groups multiple hosts to consolidate resources and enable advanced features like HA and DRS.
- **Folders:** Used to organize objects of the same type (e.g., VMs, templates, datastores) into logical groups for easier management.

These organizational objects allow you to design and manage your virtual infrastructure efficiently, tailoring the setup to fit your operational and administrative requirements.

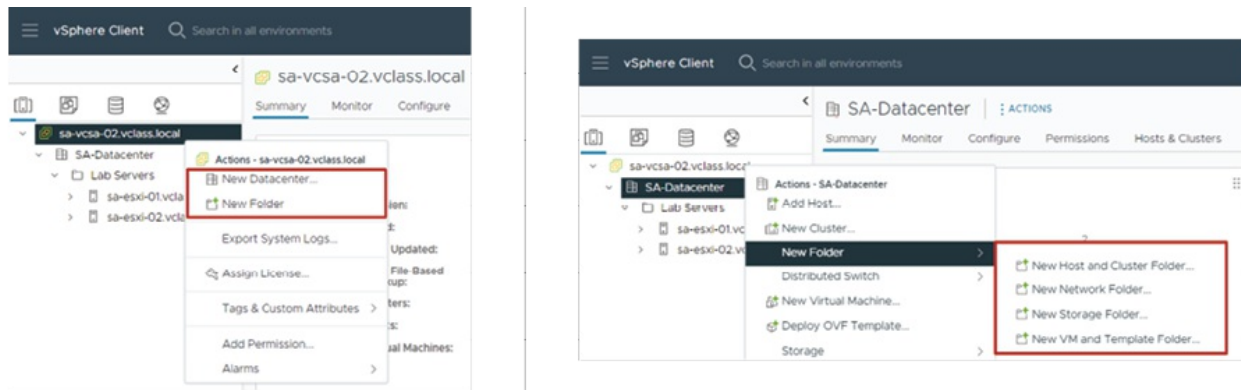


Figure 4.26: Adding objects to vCenter

(Source: VMware)

Add ESXi hosts to vCenter

The vSphere Client allows you to seamlessly integrate ESXi hosts into your vCenter system. This procedure allows for centralized control, monitoring, and utilization of advanced vSphere features like HA and DRS.

The following are the steps for adding an ESXi host in the vCenter:

1. **Log into the vSphere Client:** The Administrator can access the vCenter instance using the vSphere Client.
2. **Navigate to hosts and clusters:** In the inventory view, right-click the data center or cluster to which the ESXi host needs to be added.
3. **Select Add Host:** Use the wizard to enter the host's IP address or FQDN.
4. **Authenticate:** Enter the administrator credentials for the ESXi host.
5. **Assign a license:** If needed, administrator will be prompted to assign a license to the ESXi host. Administrator can either enter a license key or select an existing one.
6. **Review and complete:** Verify the host information and click **Finish** to add it to vCenter.

Once the process is completed, the ESXi host will be successfully added to the vCenter inventory, which allows you to better manage resources, monitor performance, and set up virtual machines, as follows:

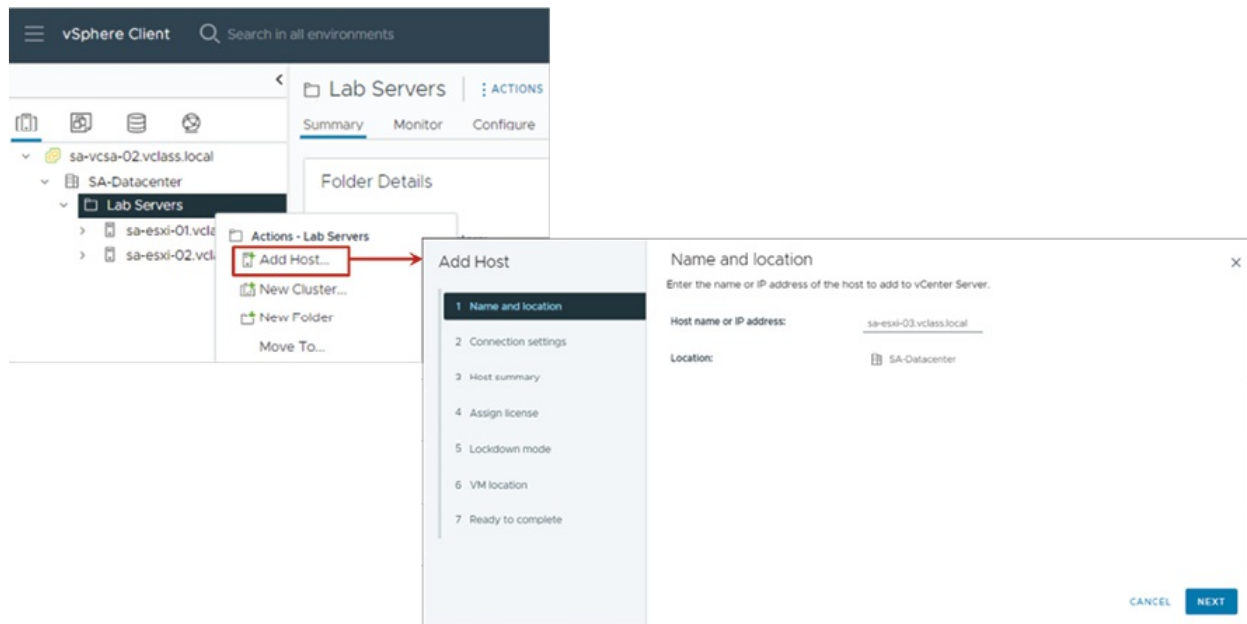


Figure 4.27: Adding ESXi host to vCenter

(Source: VMware)

Creating custom tags for inventory objects

Custom tags allow administrators to add metadata to inventory objects in vCenter, helping them categorize and manage resources effectively. Tags provide a simple way to sort, group, and locate objects based on specific attributes or purposes.

For example:

- Tag VMs running production workloads for easy identification.
- Tag VMs by guest operating system to streamline management and reporting.
- Tag clusters and datastores to organize resources based on usage or location.

The following are the benefits:

- **Simplified search:** Search inventory objects quickly based on tags.
- **Enhanced organization:** Group-related objects for better management.
- **Custom metadata:** Attach specific details, such as usage, location, or environment, to streamline operations.

Tags are especially useful in large environments where numerous inventory

objects exist, allowing for efficient categorization and streamlined workflows, as follows:

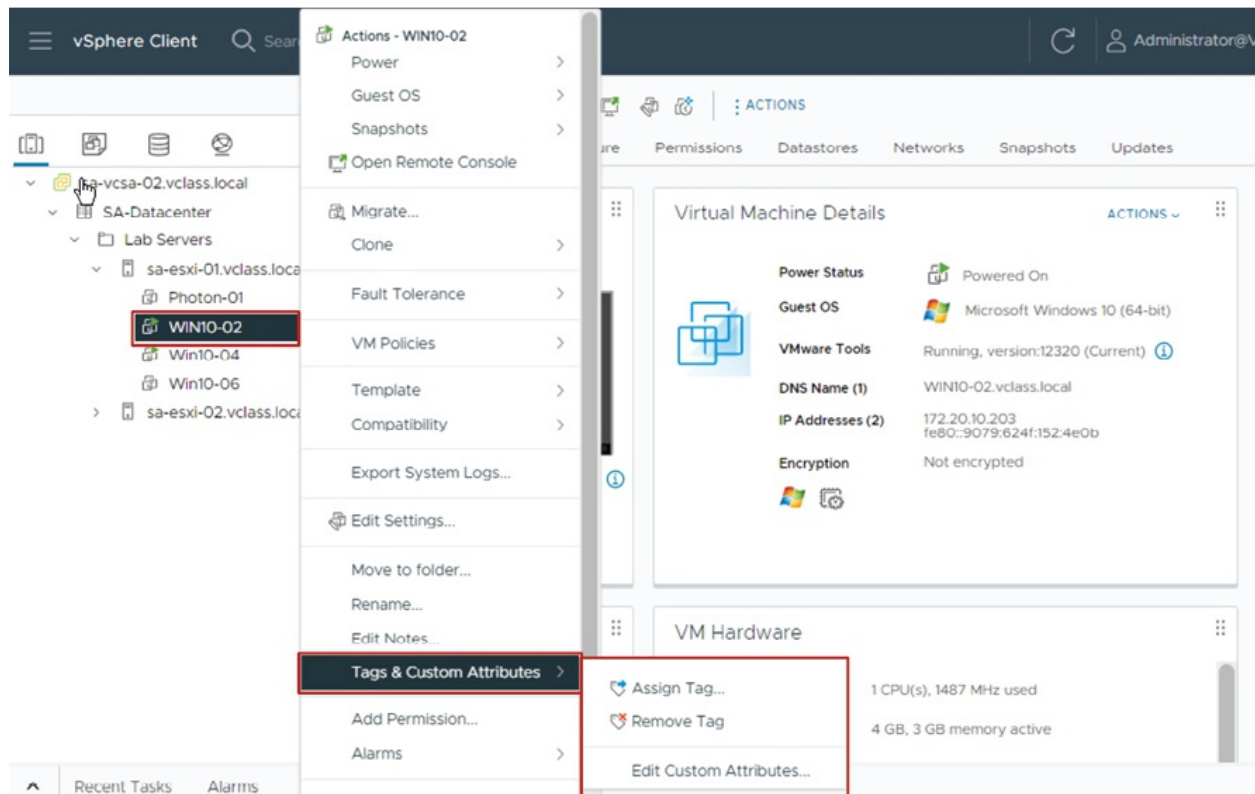


Figure 4.28: Custom Tags

(Source: VMware)

Explaining vCenter permissions

In vCenter, permissions are governed by an access control system, allowing administrators to define user privileges for accessing various objects within the inventory. Here is a breakdown of the key concepts:

- **Privilege:** An action or task that can be performed, such as creating a VM or configuring a host.
- **Role:** A set of privileges grouped together. Roles define what actions a user or group can perform.
- **Object:** The entity upon which the action is performed. For instance, this could be a VM, host, datastore, or network.

- **User or group:** The individual or group that is granted permission to perform actions.
- **Permission:** The assignment that ties a user or group to a specific role for a particular object.

This is how Permissions work.

Permissions in vCenter are assigned to objects within the inventory. When administrators assign permission, they are associating an object with a user or group and granting them a specific role. For example:

- The administrator can assign the **Read-only** role to a group for a VM object, which limits that group to viewing the VM without making changes.
- Administrators could assign the **Administrator** role to a specific user for the same VM, allowing them to perform all actions on it.

By setting permissions on different objects, administrators control what users or groups can do within the vSphere environment. For example, if the administrator wants a group to have the ability to power on or off virtual machines, you would:

- **Select the VM object:** Navigate to the specific virtual machine within the vCenter inventory.
- **Add a permission:** Assign a permission to the group, granting them a role with the **Virtual Machine.Power On** and **Virtual Machine.Power Off** privileges.

This flexibility enables granular control over who can perform which tasks on various objects in your virtual environment:

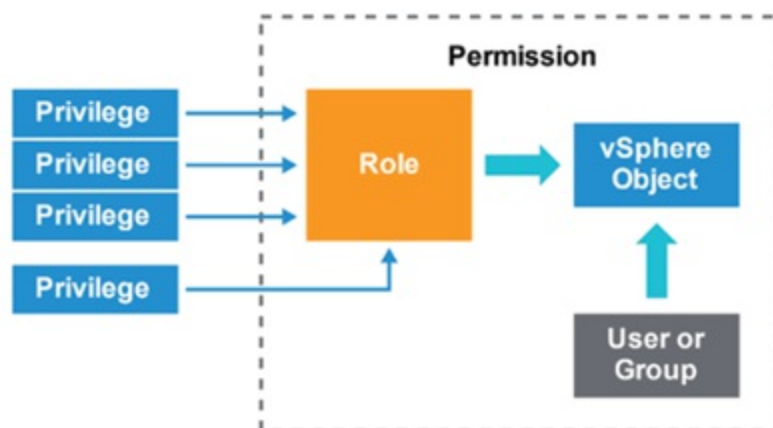


Figure 4.29: About vCenter Server Permission

(Source: VMware)

Understanding roles in vCenter

Roles in vCenter are collections of privileges that determine the level of access a user or group has to a specific object or task within the vSphere environment.

- *Privileges* are individual actions that can be performed, such as creating a virtual machine or managing host configurations.
- *Roles* are groups of related privileges that enable users or groups to perform certain tasks.

vCenter provides several *predefined system roles* that cannot be modified. However, an administrator can *clone* these roles to create custom roles that suit the needs.

The following are the key system roles in vCenter:

- **Administrator role:** Users with this role can view and perform all actions on an object. They have full access to modify settings and configurations.
- **Read-only role:** Users can only view the state and details of the object, but they cannot make any changes.
- **No access role:** Users cannot view or modify the object in any way. This role essentially restricts all access to the object.
- **No cryptography administrator role:** Users with this role have the same privileges as the Administrator role, except they do not have access to privileges in the Cryptographic operations category.
- **Custom roles:** While system roles are predefined, you can build custom roles by cloning existing sample roles. For example, if you want a role for a user who needs to manage virtual machines but not change their configurations, you could clone the Virtual Machine Power User role, which provides privileges for operations such as turning on and off VMs but does not grant full administrator powers.

It is vital to remember that positions are independent of one another, with no inheritance or hierarchy between them. This means that each role has its own

set of rights that can be adjusted to match the unique requirements of your environment, as shown:

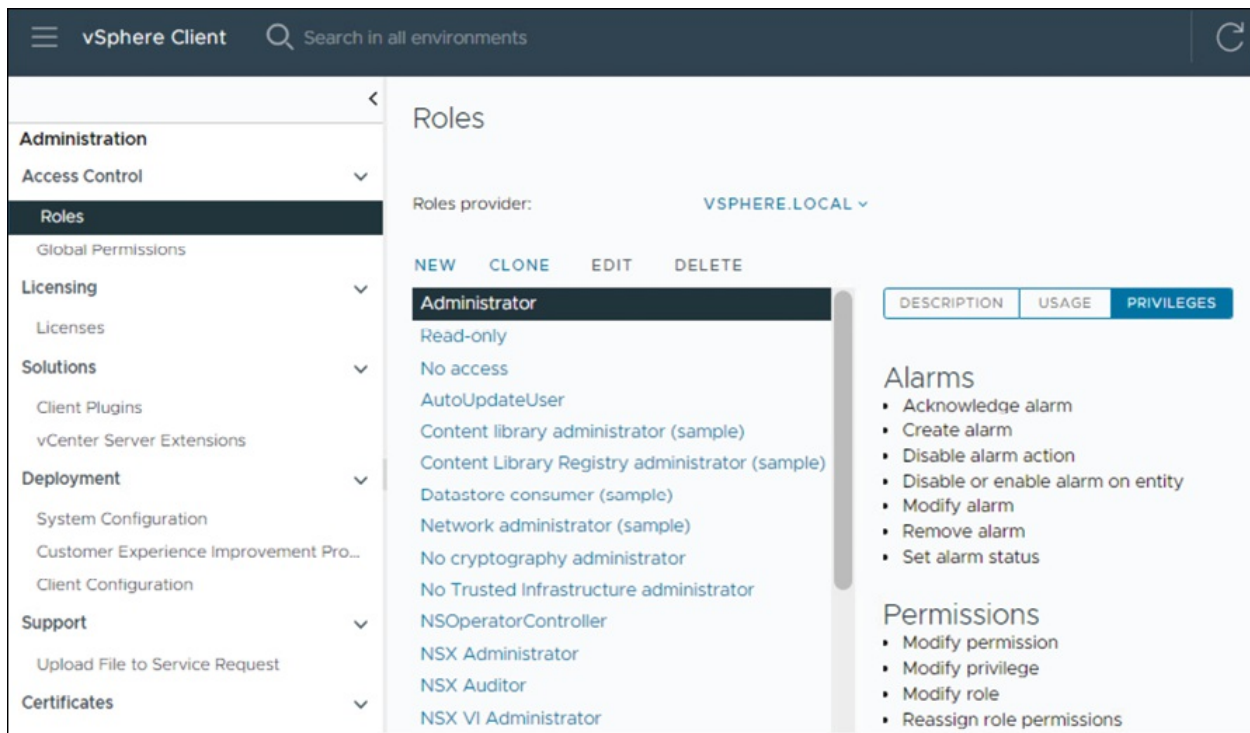


Figure 4.30: About Roles

(Source: VMware)

Understanding objects in vCenter

In vCenter, objects are the entities on which actions are carried out. These objects are the foundation of the virtualized environment, and administrator may setup and manage them with the vSphere Client.

Common vCenter objects include data centers, folders, clusters, hosts, datastores, networks, and VMs.

Permissions on objects

Each object in vCenter has a **Permissions** tab, which displays the allocated roles and users for each item. This enables administrators to specify which users or groups can conduct operations on each object.

For example:

- Administrator can provide a user the Administrator role for a datastore

object, giving them complete responsibility for configuring and managing the datastore.

- Alternatively, the administrator may give a user the Read-Only role for a host object, which allows them to see the host's status and data but prevents them from making any changes.

By managing permissions effectively, administrators can ensure that only authorized users or groups have access to certain objects in your vSphere environment, as follows:

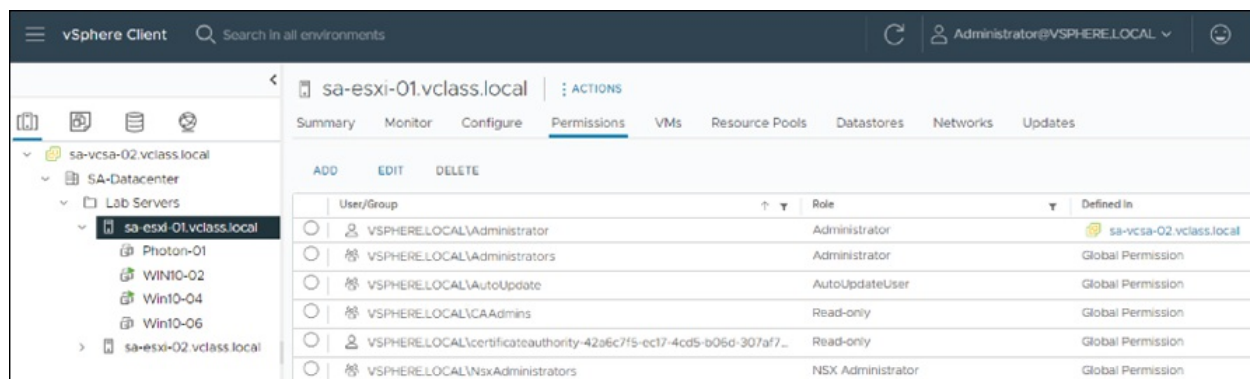


Figure 4.31: About objects

(Source: VMware)

Assigning permissions in vCenter

To grant permissions to objects in vCenter, perform these steps:

1. **Select an object:** Navigate to the object (e.g., host, datastore, VM, folder) to which permissions need to be applied. Right-click the object and choose Permissions.
2. **Select a domain:** Select the domain that the user or group belongs to. The domain may be the Active Directory domain or a local vCenter domain.
3. **Select a user or group:** Choose the user or group where authorization need to be granted from the domains available. You can search for a user or group by name.
4. **Select a role:** Assign a role to the specified user or group. The role specifies the user or group's privileges on the object. Administrators can select from existing roles such as Administrator and Read-Only or build

their own positions with particular permissions.

5. **Propagate the permission:** To ensure that the permission is passed down to child objects (such as files, clusters, or VMs), select the propagate permission option. This will provide permission to all objects in the selected object's hierarchy, simplifying permission administration.

Permissions can be assigned at various levels of the object hierarchy. For example:

- **Host object:** Permissions granted to a host will only affect that host.
- **Folder object:** Permissions set to a folder can be applied to all objects within it, including VMs and hosts.
- **Global root object:** Permissions established at the global root level (the top of the inventory) apply to all objects in the vCenter environment, ensuring consistent access control.

The vSphere Security Documentation at <https://techdocs.broadcom.com> contains more information on permission hierarchies and how to manage global permissions.

By carefully assigning roles and permissions, administrators can enforce security and access control throughout your vSphere environment, ensuring that individuals and groups only have the privileges required for their jobs, as shown:

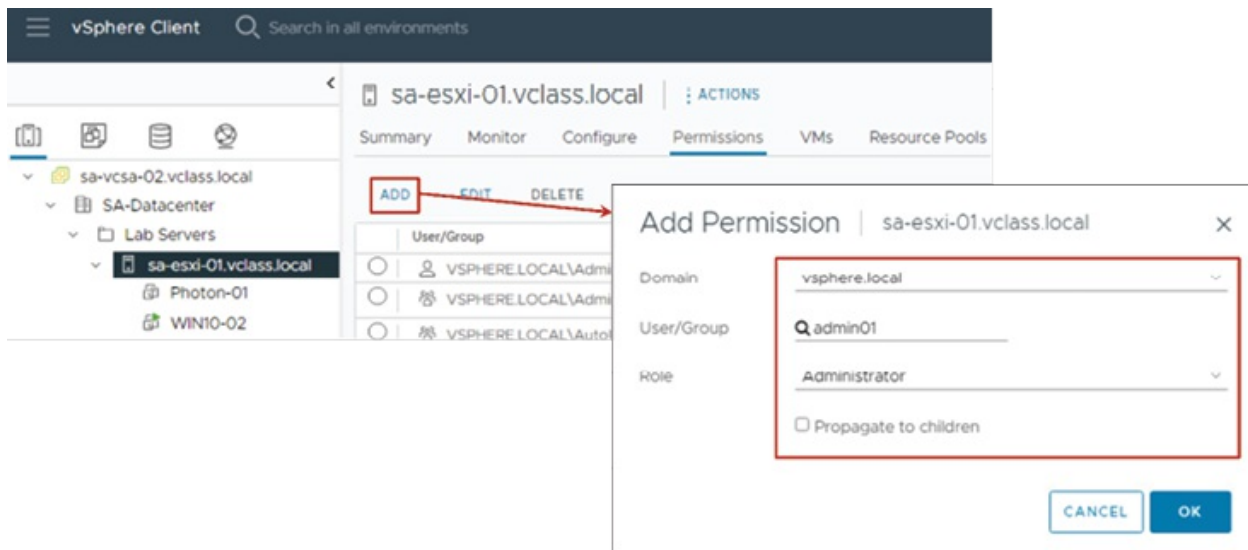


Figure 4.32: Assigning Permissions

(Source: VMware)

Viewing roles and user assignments in vCenter

Administrators can view the details of roles and their user assignments in the vSphere Client's Roles pane. This page provides visibility into how roles are distributed across objects and which users or groups are assigned specific roles.

The following are the steps to view roles and user assignments:

1. **Open the roles Pane:** Log in to the vSphere Client and navigate to the **Roles** section from the menu.
2. **Select a role:** Select the role to inspect in the Roles list. This could be a system-defined role (e.g., Administrator, Read-only) or a custom role.
3. **Click on usage:** In the **Roles** pane, click on the **Usage** tab to access detailed information about the selected role.
4. **View role information:** On the right-hand side, you will see the following:
 - a. **Objects:** A list of objects (e.g., VMs, clusters, datastores, etc.) where the role is applied.
 - b. **Users and groups:** The users or groups that have been granted the selected role on these objects.

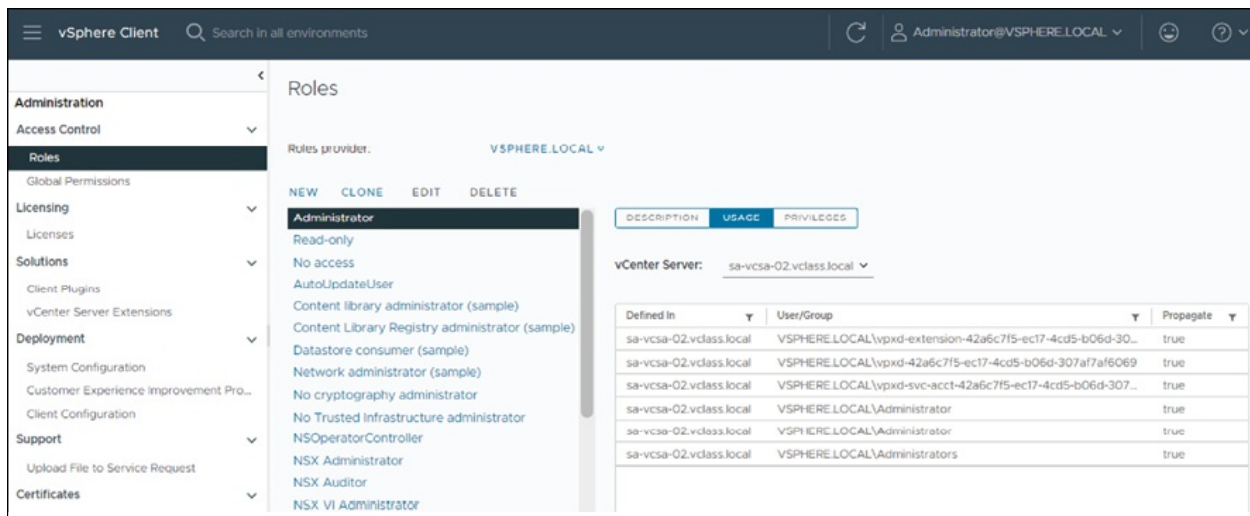


Figure 4.33: Viewing Roles and User Assignment

(Source: VMware)

Let us consider the following use case example.

For instance, if they regularly review roles and user assignments, administrators can ensure that access control policies are enforced effectively and that permissions are properly aligned with operational requirements.

Understanding permission propagation and combination in vCenter

Permissions in vCenter can either propagate down the object hierarchy to all sub-objects or apply specifically to a single object. Administrators can override inherited permissions by explicitly setting new permissions for a lower-level object, as shown:

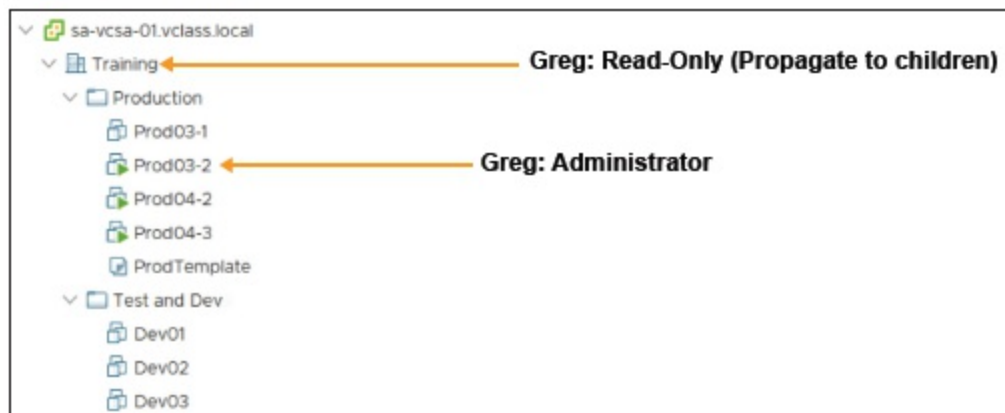


Figure 4.34: Permission propagation Scenario 1

(Source: VMware)

Additionally, when a user belongs to multiple groups with permissions on the same object, the user receives the combined privileges of all assigned roles across those groups, as shown:

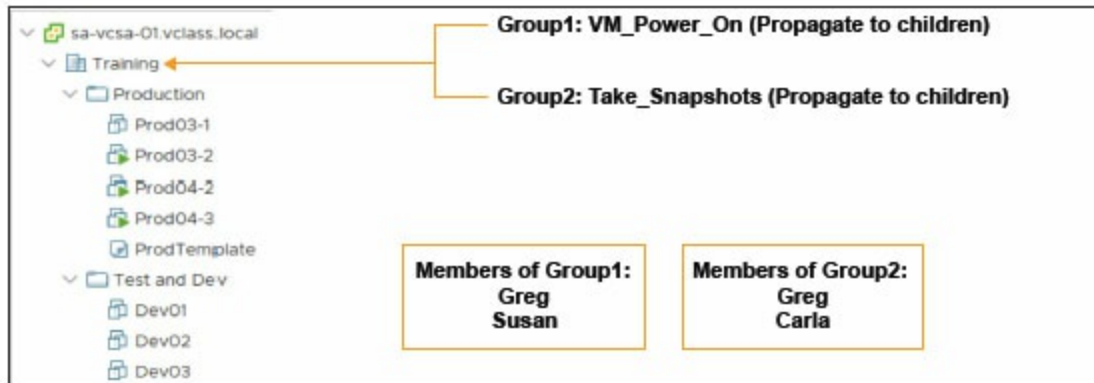


Figure 4.35: Permission propagation Scenario 2

(Source: VMware)

Creating a role in vCenter

To preserve security and simplify administration, roles must only have the privileges that are required. To enable a user, like **abc@company.com**, to deploy virtual machines from a template, for example, a Provision VMs role can be made. This role can be given explicitly to a folder, like Production VMs, to establish appropriate access control and restrict its reach.

Define roles with few privileges and give them names that express their function to improve usability and clarity. This strategy guarantees efficient role administration and contributes to the upkeep of a safe and orderly virtual environment, as follows:

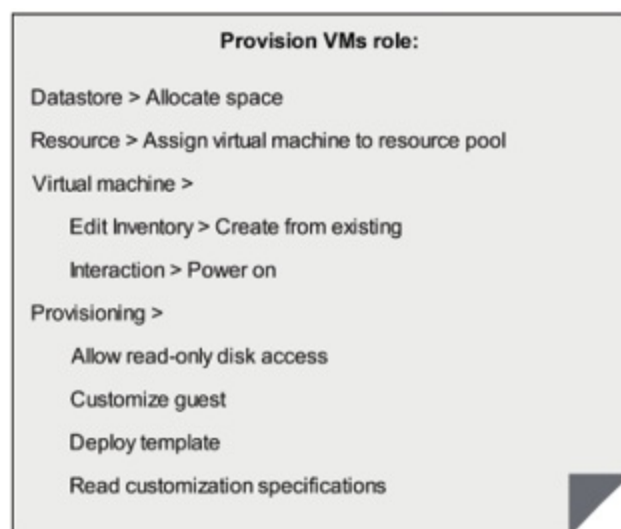


Figure 4.36: Creating a role

(Source: VMware)

About global permissions

Global permissions provide a way to assign privileges across all vSphere solutions and multiple vCenter instances from the global root object. These permissions span solutions like vRealize Orchestrator and grant access to all objects across multiple vCenter hierarchies.

Typically, permissions are assigned to specific vCenter inventory objects, such as ESXi hosts or VMs, by granting a role (a set of privileges) to a user or group for that object. In contrast, global permissions extend this capability to encompass all objects in every inventory hierarchy within the deployment.

For example, a global root object can include permissions that apply to content libraries, vCenter instances, and tags, enabling consistent access and management across all linked vCenter instances. However, global permissions remain effective only within the context of a specific vCenter instance, as follows:

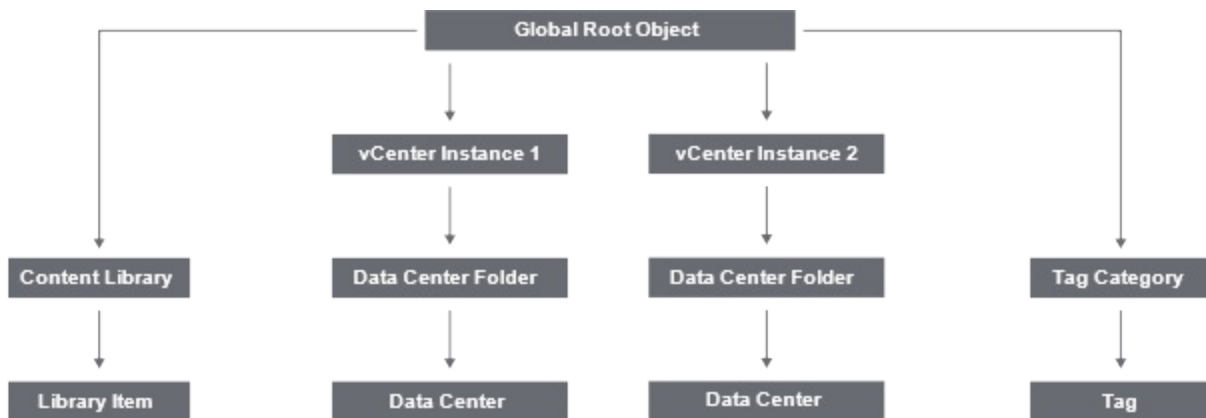


Figure 4.37: About global permissions

(Source: VMware)

Gaining insights from vCenter logs and events

In vSphere, any operation or action performed as part of daily administration is referred to as a *task*. Examples of tasks include:

- Adding or removing a datastore

- Creating or modifying resource pools
- Deploying a virtual machine from a template
- Cloning or migrating a virtual machine
- Configuring vSphere Distributed Switch settings
- Assigning or updating permissions for users and groups

The vSphere Client allows administrators to monitor tasks and identify who initiated them. This task information is valuable for troubleshooting, as it provides a detailed record of actions performed within the vSphere environment.

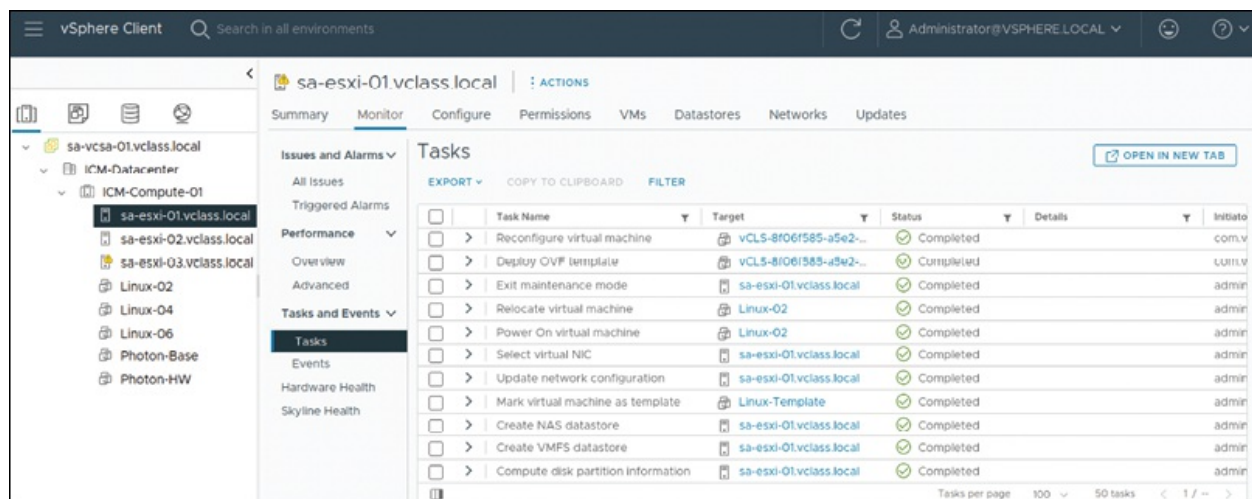


Figure 4.38: vSphere Tasks

(Source: VMware)

vSphere events provide a detailed record of actions or activities that take place within the vCenter inventory, capturing both user and system-generated actions. These events include information such as the user's account, the specific action performed, and the associated object.

Key details captured in vSphere events include:

- **Event severity:** Indicates whether the event is informational, warning, or critical, helping administrators prioritize their responses.
- **Event source:** The component or service that generated the event (e.g., ESXi host, vCenter, vSphere Client).
- **Affected entity:** The specific object, such as a virtual machine, host, datastore, or network, that the event pertains to.

- **User permissions:** The level of access the user had when performing the event action (e.g., admin, read-only).
- **Event ID:** A unique identifier for each event, which can be useful when searching or filtering events.
- **Associated task:** If the event is related to a task, such as creating a VM or changing a configuration setting, the task ID will also be associated with the event.

In addition to these details, vSphere events and alarms alert administrators about significant changes or failures in the environment, such as issues with service health.

The *Tasks and Events* panes in the vCenter Client allow administrators to monitor events, offering an audit trail that by default maintains a 30-day history.

Some examples of actions that might trigger a recorded event include:

- A virtual machine snapshot is created or deleted.
- A datastore is created, expanded, or removed.
- A user logs into the vSphere Client or vCenter Server.
- A host enters maintenance mode or exits maintenance mode.
- An alarm triggers due to performance degradation or capacity overuse.
- A license key is added or updated.
- Network configuration changes are applied (e.g., network adapter added or modified).
- A VM's resource allocation (CPU, memory) is modified.

These events provide vital information for troubleshooting, monitoring, and auditing activities in a virtualized environment.

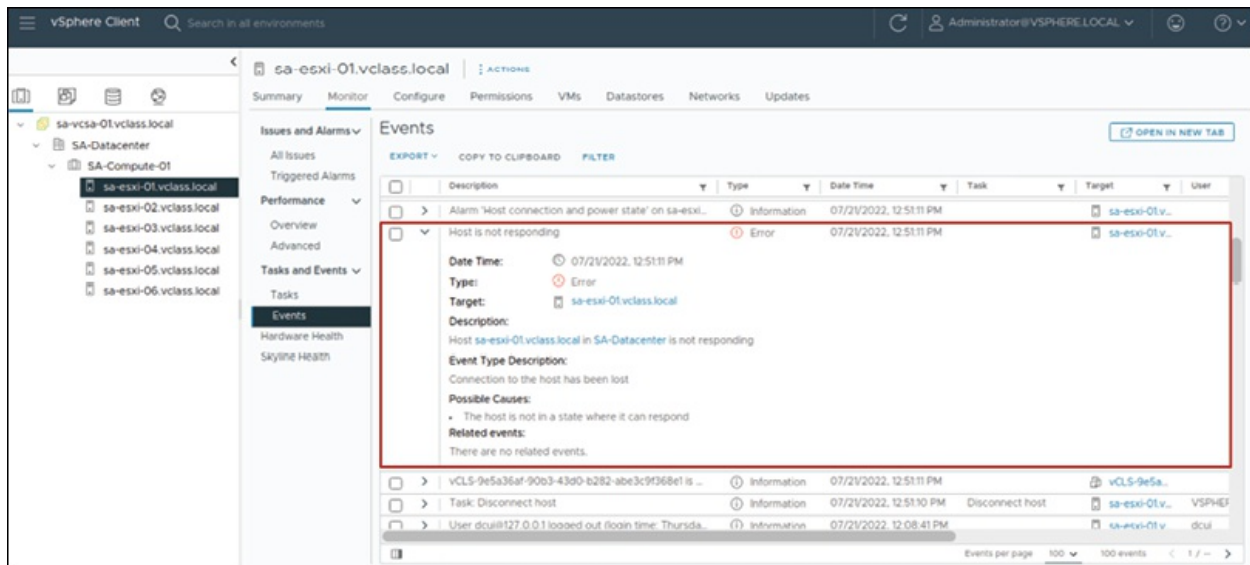


Figure 4.39: vCenter Events

(Source: VMware)

vCenter log levels

vCenter services generate logs that can be instrumental for troubleshooting and tracking various events. To manage the volume and specificity of these logs, you can adjust the log levels, which control the amount and type of data logged by vCenter.

When to adjust log levels:

- **Verbose or Trivia level:** For detailed logs, especially when troubleshooting complex issues.
- **Info or Error level:** For regular operation and managing system performance without overwhelming the log storage.

Available log levels and descriptions:

- **None:** Disables all logging, resulting in no logs being stored.
- **Error:** Records only error-level log entries, which capture critical issues or failures that impact the functionality of vCenter or its services.
- **Warning:** Logs errors and warnings, capturing potential issues that may not yet be critical.
- **Info:** Logs general information, errors, and warnings; standard setting for everyday operation.
- **Verbose:** Includes detailed information along with errors, warnings, and

normal logging; useful for in-depth troubleshooting.

- **Trivia:** Captures all details, including information, errors, warnings, verbose logs, and additional minute details for extensive diagnostics.

Changes made to log settings take effect immediately without the need to restart the vCenter system. However, it is crucial to reset the log level to *Info* after troubleshooting to avoid overwhelming the system with excessive log data.

Configuring log levels

Administrators have the liberty to control the level of detail vCenter collects in log files. More detailed logging can be useful for troubleshooting but requires additional storage space on your vCenter system.

To configure log levels in the vSphere Client, follow these steps:

1. In the vSphere Client, select the vCenter instance from the navigation pane.
2. Click the **Configure** tab.
3. Under **Settings**, click **General**.
4. Click **EDIT**.
5. In the left pane, select **Logging settings**.
6. From the **Log level** drop-down menu, choose the desired log level.

This allows you to adjust the amount of log information based on your troubleshooting needs, ensuring you capture the right level of detail for your specific situation, as shown:

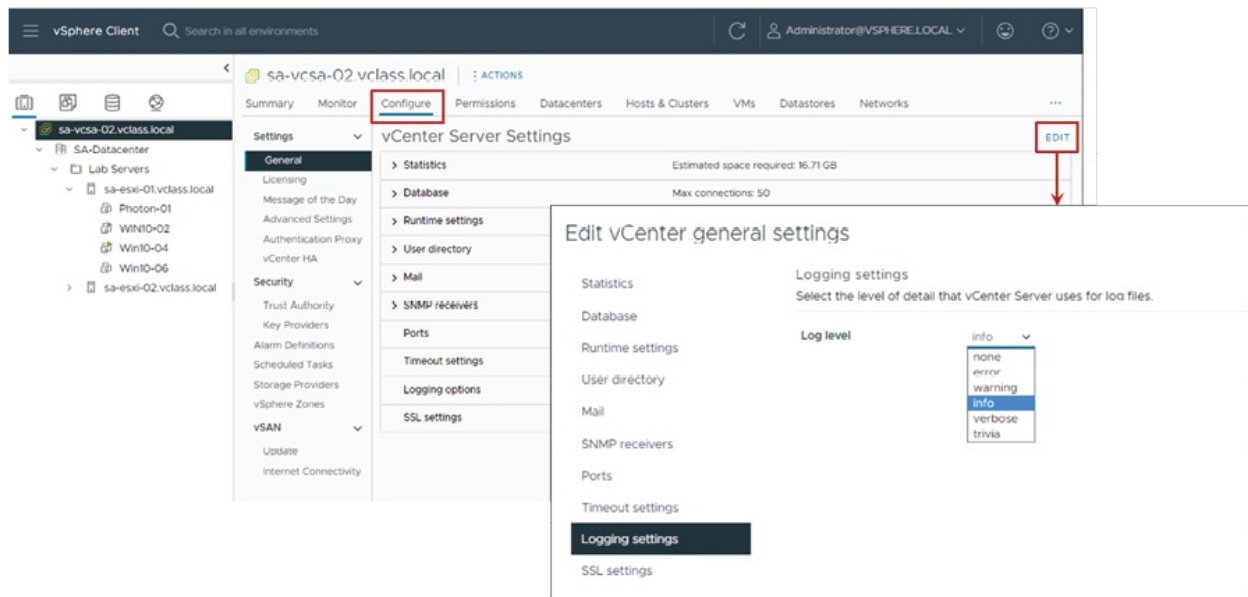


Figure 4.40: Setting log levels

(Source: VMware)

Forwarding vCenter log files to a central log server

vCenter provides the capability to forward its log data to a remote Syslog server, enabling centralized logging and more efficient monitoring of events throughout your environment. This feature is particularly beneficial in large-scale environments where managing logs locally may not be practical.

To enable log forwarding, follow these steps:

1. Log in to the **vCenter Management Interface**.
2. Navigate to the **Syslog** settings.
3. Click **Configure** under the forwarding configuration section and enter the IP address or hostname of the remote Syslog server, along with the protocol and port number.
4. Save the configuration.

By forwarding logs to a remote Syslog server, you streamline log management, making monitoring and troubleshooting issues across your environment easier with a centralized log repository.

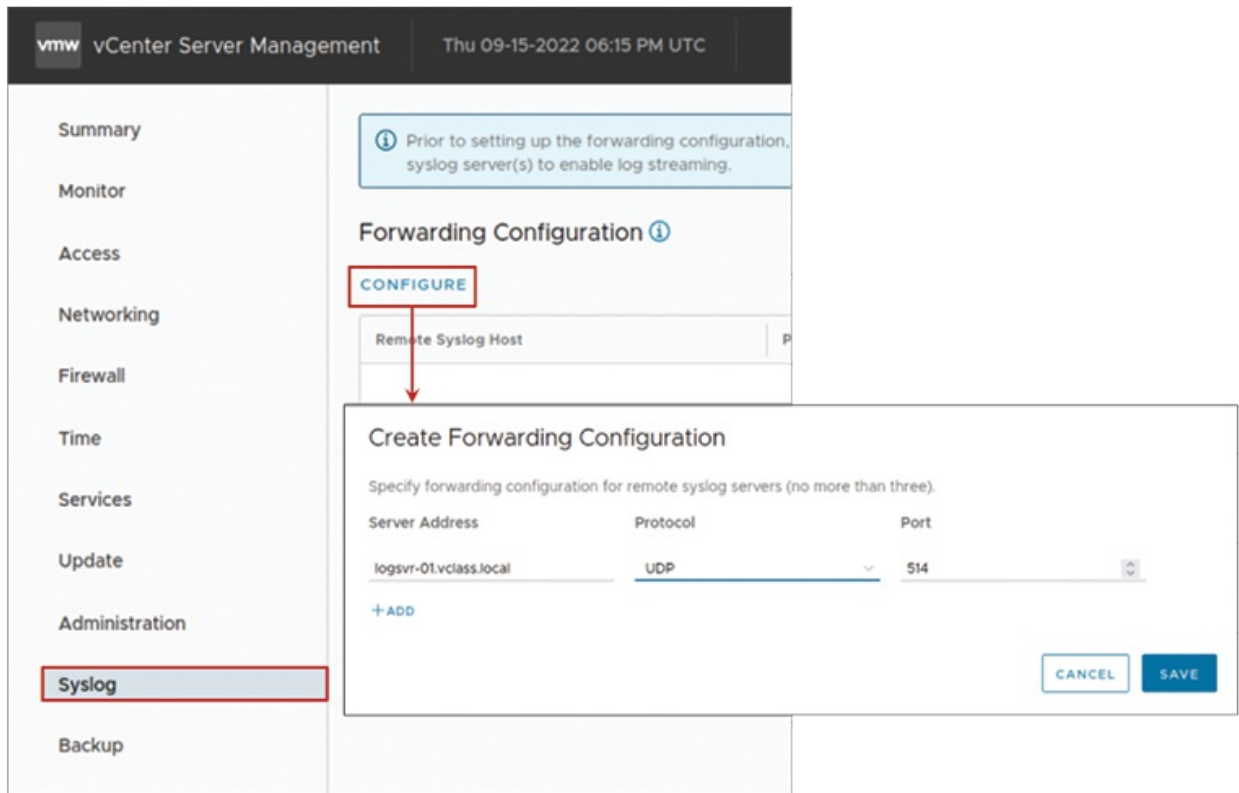


Figure 4.41: Forwarding vCenter Log Files

(Source: VMware)

Forwarding ESXi host log files to a central log server

To forward log files from an ESXi host to a remote Syslog server, specify the server's address in the *Advanced System Settings* section of the *vSphere Client*. This enables centralized log management and enhances the ability to monitor and troubleshoot the host environment.

Additionally, you can analyze the forwarded vCenter/ESXi logs using log analysis tools like **vRealize Log Insight**, which provides deeper insights and streamlined log management for better operational efficiency, as follows:

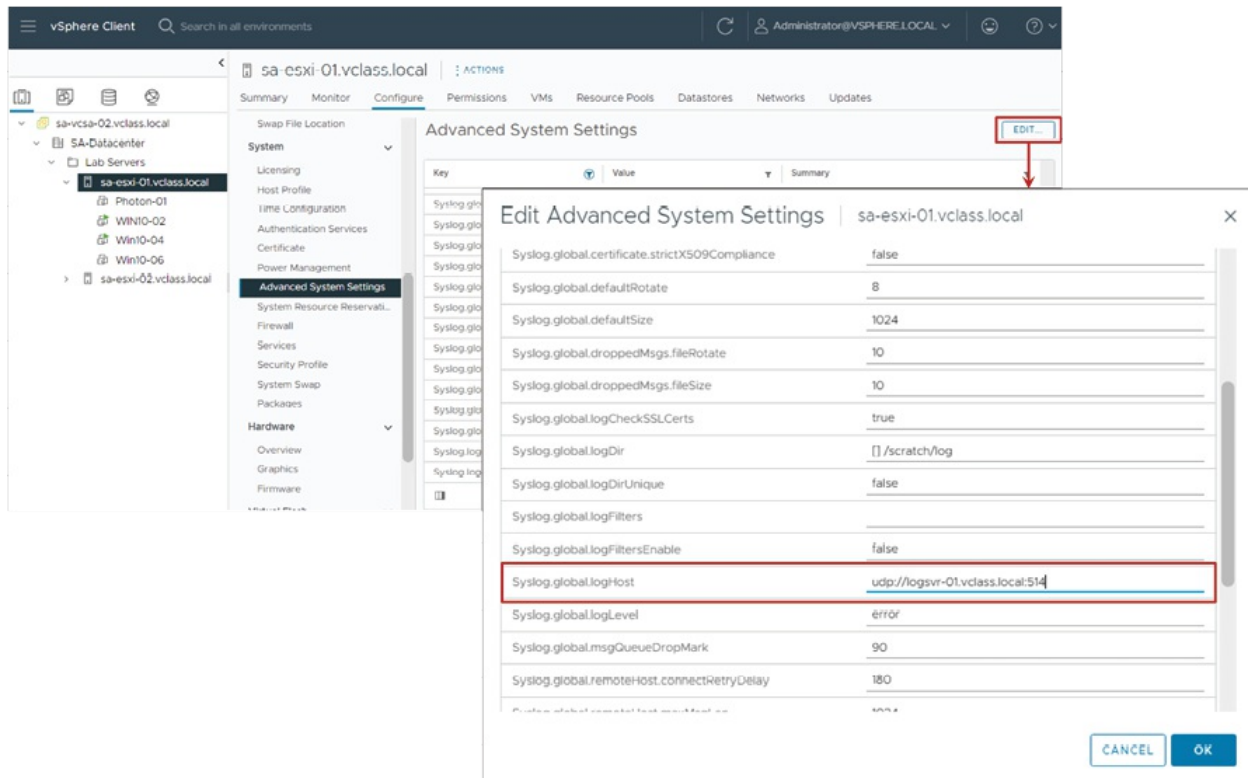


Figure 4.42: Forwarding ESXi host log files

(Source: VMware)

Conclusion

In this chapter, we looked at the key features and deployment options of VMware vCenter, the foundation of a well-managed virtualized environment. Readers learned how the vCenter server connects with ESXi hosts, enabling centralized control of the virtualized environment. We addressed the deployment and configuration of the vCSA, as well as critical vCenter settings that improve speed, scalability, and organization. We also talked about managing license keys with the vSphere Client, organizing inventory objects to make resource management easier, giving permissions to regulate access securely, and using logs and events for monitoring and troubleshooting purposes. Mastering these abilities guarantees that readers have the foundation for managing a streamlined, efficient virtualized infrastructure.

Now that readers understand vCenter deployment and configuration, we will

move on to a critical part of the vSphere environment, networking.

In [Chapter 5, Networking in vSphere](#), we will look at how to configure and optimize virtual networks, which are a key component of any virtualization strategy. Readers will learn about the distinctions between standard and distributed switches, how to implement network policies, and how to ensure that the virtual infrastructure communicates smoothly. This chapter will teach readers how to build a resilient, flexible network that supports virtualized workloads efficiently and securely.

Points to remember

- The vCSA, running on Photon OS with a PostgreSQL database, is central to managing ESXi hosts and virtual machines across the infrastructure.
- The **vCenter Management Interface (VAMI)** provides controls for vCenter networking and services, allowing configuration and updates within a centralized console.
- The vCenter service (vpxd) interacts with the ESXi host daemon (hostd) via the vCenter agent (vpxa).
- vCenter communicates directly with ESXi hosts, coordinating services and settings to ensure streamlined operations within the virtual environment.
- The vSphere Client connects users to vCenter, where they can manage inventory objects, create custom tags, and organize data centers and organizational objects.
- Licensing in vCenter allows administrators to view licensed features, add new license keys, and manage licensing for vCenter and ESXi hosts.
- Permissions in vCenter assign roles (collections of privileges) to users for specific objects, with global permissions allowing broad access across all vCenter objects.
- Inventory management within vCenter includes adding data centers, ESXi hosts, and organizational objects and using custom tags for the organization.
- vCenter's logging levels, adjustable for various data collection needs,

can impact filesystem usage based on the amount of data collected.

Exercises

1. What is the role of the vCenter Server Appliance, and which operating system and database does it use?
2. How does vCenter communicate with ESXi hosts, and why is this connection important?
3. What is the purpose of the vSphere Client, and what types of objects can you manage with it?
4. What is the difference between standard permissions and global permissions in vCenter?
5. Why would an administrator adjust the vCenter logging level, and what is the impact?
6. How do custom tags and organizational objects help in managing vCenter inventory?
7. What are the steps to add license keys in vCenter, and how does licensing affect ESXi hosts?

Lab exercises

1. **Deploying the vCenter Server Appliance:** The Objective is to learn how to deploy the vCSA into the virtual infrastructure.
 - a. Access the vSphere Client and log in to an ESXi host with administrative privileges, or jump to a host/Windows client with sufficient permissions.
 - b. Open the vCSA installer from the installation media or downloaded file.
 - c. Choose the Deploy vCenter Server option and follow the setup wizard.
 - d. Select the target ESXi host, provide a name for the vCenter Server VM, and configure its size and storage.
 - e. Specify network settings (IP address, DNS, gateway, and subnet

mask).

- f. Complete the deployment process and verify that the vCSA is running.
2. **Adding vSphere licenses:** The objective is to learn how to add and assign vSphere licenses to vCenter and ESXi using the vSphere client.
 - a. Log into vSphere Client to *Access License Management | Navigate to Administration | Licensing | Select License Management tab*
 - i. Add vSphere Licenses to vCenter and assign it to the vCenter Instance
 - ii. Add ESXi Licenses to vCenter and assign it to the ESXi host(s)
3. **Adding an identity source:** The objective is to learn how to add an identity source to the vCenter Server for centralized user authentication and management.
 - a. Access the vSphere Client with administrative privileges.
 - b. Navigate to SSO settings under the administration section and click Configuration.
 - c. Under the Identity Sources tab, click Add.
 - d. Select the type of identity source (e.g., Active Directory (Integrated Windows Authentication), LDAP, or OpenLDAP).
 - e. Provide Configuration Details:
 - For Active Directory, enter the domain name and provide administrative credentials.
 - For LDAP, enter the LDAP server URL, base DN, and authentication details.
 - f. Click *Test Connection* to verify that vCenter can communicate with the identity source.
 - g. After the test succeeds, click OK to add the identity source.
 - h. Finally, navigate to the Users and Groups tab under SSO and confirm that users and groups from the identity source are listed.
4. **Configuring vCenter server settings:** The objective is to learn how to configure essential settings for vCenter Server, including network and service configurations.
 - a. Log in to the VAMI at **<https://<vCenter-IP>:5480>**.

- b. Navigate to the Network Settings section and verify or update the DNS, hostname, and network adapter details.
 - c. Configure the NTP settings for time synchronization under the Time Settings section.
 - d. Restart the necessary services from the Services tab if required.
 - e. Log in to the vSphere Client and verify that the changes are reflected in the vCenter inventory.
5. **Managing inventory objects in vCenter:** The objective is to learn to create and organize inventory objects like data centers, clusters, and hosts.
- a. Log in to the vSphere Client and navigate to the Inventory tab.
 - b. Create a new data center by right-clicking the vCenter object and selecting New Data Center.
 - c. Add ESXi hosts to the data center by providing their IP addresses or FQDNs and credentials.
 - d. Create and configure a new cluster within the data center.
 - e. Explore creating custom tags and assigning them to inventory objects.
6. **Configuring Permissions in vCenter:** The objective is to understand how to create roles and assign permissions to users and groups.
- a. Log in to the vSphere Client and navigate to the Administration tab.
 - b. Create a custom role by selecting Roles and defining privileges.
 - c. Assign the role to a user or group for a specific inventory object.
 - d. Set a global permission to allow a user access across multiple vCenter instances.
 - e. Verify the permissions by logging in as the user and checking the allowed actions.

Join our Discord space

Join our Discord workspace for latest updates, offers, tech happenings around the world, new releases, and sessions with the authors:

<https://discord.bpbonline.com>



CHAPTER 5

Networking in vSphere

Introduction

In this chapter, we will explore the revolutionary potential of virtual networking and guarantee smooth communication inside the virtual infrastructure. When ESXi networking is set up properly, virtual machines may easily communicate with both virtual and physical machines.

Furthermore, the VMkernel can execute IP-based storage and remote host management tasks with accuracy and efficiency when the network is properly configured. vSphere Standard Switches offer dependable and simple networking options for modest organizations. The sophisticated features of vSphere Distributed Switches, however, are invaluable as the infrastructure grows, providing improved networking capabilities and simplified management for bigger, more complicated situations.

Let us embark on this journey to build a robust networking foundation that simplifies and elevates VMware operations.

Note: VMware is now part of Broadcom and is known as 'VMware by Broadcom'. All references to VMware in this book reflect this change.

Structure

In this chapter, we will cover the following topics:

- Unleash the potential of standard switches
- Mastering distributed switches
- Standard vs. distributed switches
- Sculpting network policies with finesse

Objectives

By the end of this chapter, readers will be able to confidently navigate the details of virtual networking within VMware vSphere. Readers will learn to identify various virtual switch connection types and configure standard switch settings while mastering the ability to view and manage these configurations. This chapter will also guide readers in setting security, traffic shaping, and NIC teaming policies to ensure optimal performance and failover management for virtual switches. Additionally, readers will explore the key differences between standard and distributed switches, recognizing the advanced features and benefits that distributed switches bring to larger, more complex environments. Finally, readers will acquire the skills needed to create and manage distributed switches effectively, empowering readers to streamline and enhance the virtualized networking infrastructure.

Unleash the potential of standard switches

Virtual switches are essential networking components within VMware vSphere, enabling seamless connectivity for virtual machines and services. They serve as virtualized versions of actual network switches and offer the following essential features:

- **Connecting virtual machines:** Virtual switches allow VMs to communicate with one another whether they are located on the same ESXi host or on other hosts within the infrastructure.
- **Supporting VMkernel services:** These switches also support VMkernel services, including iSCSI for storage access, NFS for file-sharing storage, vSphere vMotion for live migrations, and management network access for host administration.

Virtual switches ensure that the virtualized environment runs effectively, is

adaptable, and is scalable by bridging the gap between virtual and real networks.

VMware vSphere supports two types of virtual switches, which are designed for flexibility. These switches automatically create and remove ports as needed to accommodate network traffic, as follows:

- **Standard switch:** A standard switch is configured individually for a single ESXi host. It provides basic networking capabilities and is ideal for smaller or standalone environments.
- **Distributed switch:** A distributed switch extends networking functionality across an entire data center. It allows up to 2,000 hosts to connect to the same switch while maintaining consistent configuration across all attached hosts. Distributed switches simplify network management in large-scale environments but require specific licensing:
 - Hosts must have an Enterprise Plus license, or
 - Be part of a vSAN cluster.

These two types of switches provide scalable and tailored solutions to meet the networking needs of both small-scale and enterprise-level infrastructures.

Virtual switch connections and examples

Virtual switches in vSphere support several connection types, each serving a distinct role within the virtualized network infrastructure. Port groups act as templates that store the configuration settings for creating virtual switch ports.

They ensure consistent networking properties across connected devices, as follows:

- **VM ports:** Connect virtual machines to the virtual switch for communication within and outside the ESXi host.
- **VMkernel ports:** Enable communication for critical VMkernel services, including:
 - IP storage (iSCSI or NFS)
 - vSphere vMotion migration
 - vSphere Fault Tolerance

- vSAN
- vSphere replication
- ESXi management network
- **Uplink ports:** Connect the virtual switch and the physical network by utilizing physical Ethernet adapters. Both VM and VMkernel ports rely on *uplink ports* to connect to the physical network, enabling seamless communication between virtual and physical environments.

Refer to the following figure:

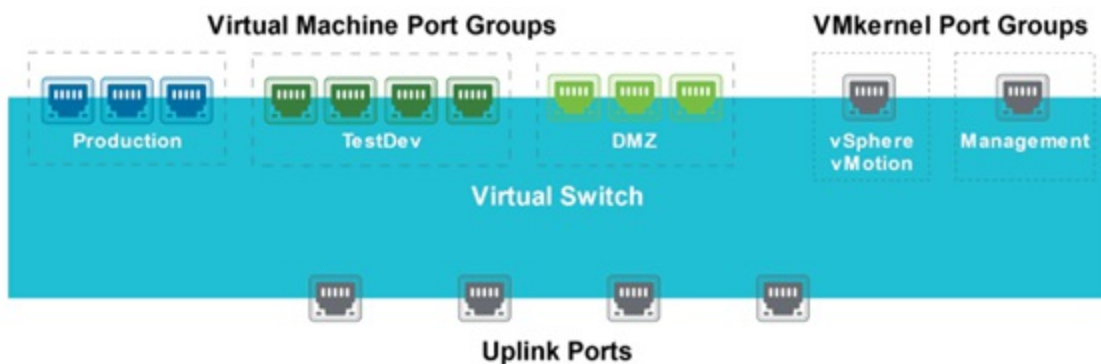


Figure 5.1: Type of virtual switch connections

(Source: VMware)

In a virtualized environment, networks (port groups) can either coexist on the same virtual switch or be distributed across multiple virtual switches, depending on your design preferences and physical network layout.

When designing your virtual networking environment, consider the following approaches:

- **Single virtual switch for all networks:**
 - Group all networks on a single virtual switch.
 - Use VLANs to isolate traffic and segregate networks.
 - Ideal when the number of available physical network adapters (NICs) is limited.
- **Separate virtual switches for each network:**
 - Assign a dedicated virtual switch to each network for better isolation.
 - Requires sufficient physical NICs to support multiple virtual

switches.

The following figures illustrate the virtual switch connection examples:

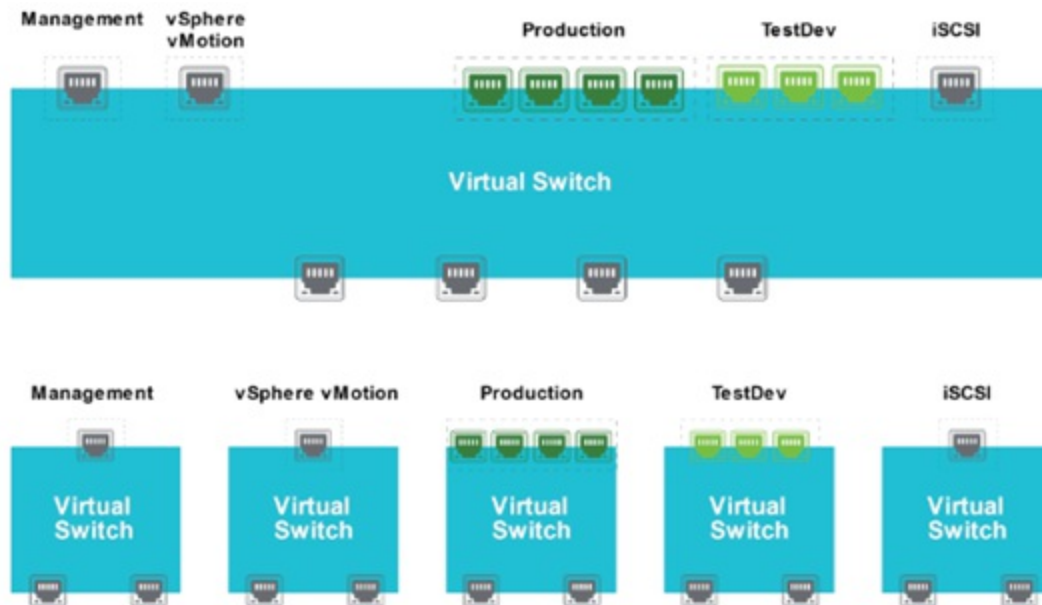


Figure 5.2: Virtual switch connection examples

(Source: VMware)

VLANs and virtual switch tagging

Virtual Local Area Networks (VLANs) segment networks logically, allowing virtual machines or ports in a VLAN to communicate as if they are on the same physical LAN. They provide logical networks independent of physical topology, improve performance by limiting broadcast traffic, and reduce costs by avoiding new routers.

ESXi VLAN implementation is done as follows:

- VLANs are configured at the *port group level* with a VLAN ID (default 0).
- ESXi supports 802.1Q VLAN tagging:
 - *Outgoing traffic* is tagged as it leaves the virtual switch.
 - *Incoming traffic* is untagged before reaching the VM.
- The *VMkernel* manages all tagging and untagging seamlessly, with minimal performance impact.

- **Physical switch setup:**

- Physical switch ports connected to ESXi hosts must be trunk ports, enabling tagged packet transmission.
- No VLAN settings are needed on the VM.

For details, see VMware KB 1003806 at

<https://knowledge.broadcom.com/external/article?legacyId=1003806>

The following figures illustrate the VLAN tagging:

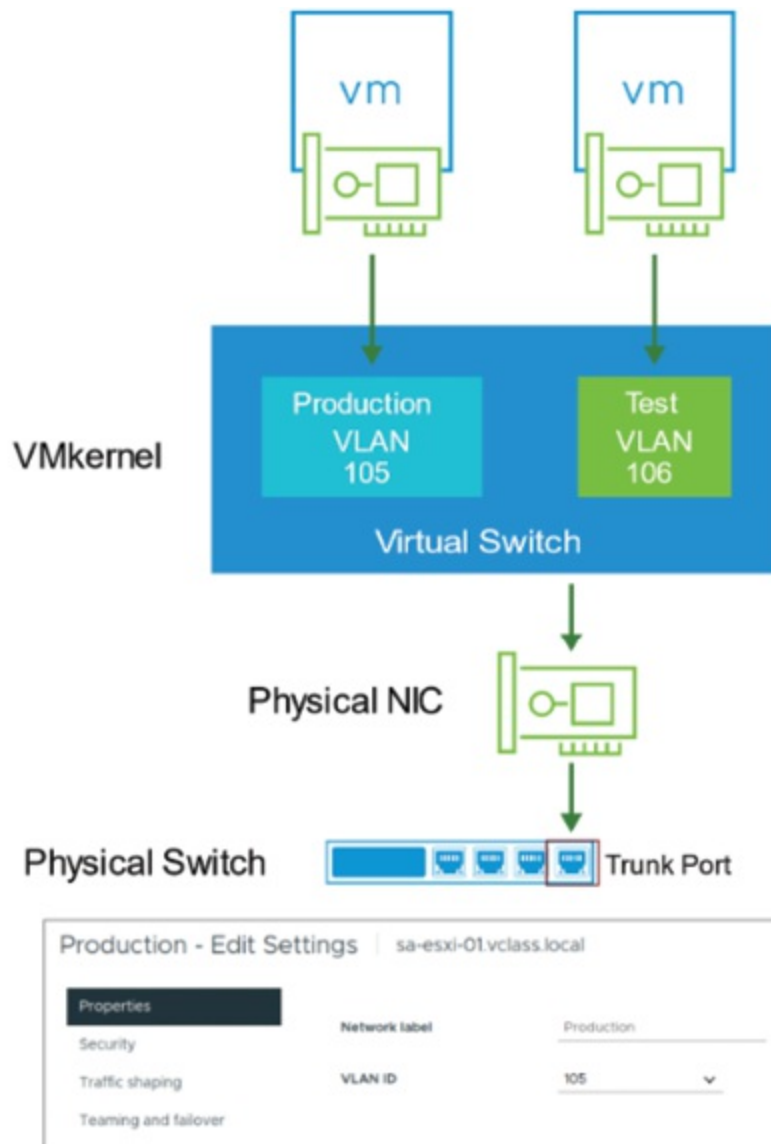


Figure 5.3: VLAN tagging

(Source: VMware)

Viewing and adding standard switches

In the *vSphere Client*, the administrator can view a host's standard switch configuration by navigating to the **Configure** tab and selecting **Virtual switches**.

By default, the ESXi installation creates a standard switch, such as *vSwitch0*, which includes the following:

- **VM network:** A default virtual machine port group for VM traffic.
- **Management network:** A port group containing a VMkernel port for management traffic.

Administrators can create additional port groups to meet specific requirements. For example, an administrator might create an IP Storage port group with a VMkernel port for iSCSI storage access.

The following is the best practice:

Remove the default **VM Network** port group and separate virtual machine and management traffic onto different physical networks or VLANs for better performance and security.

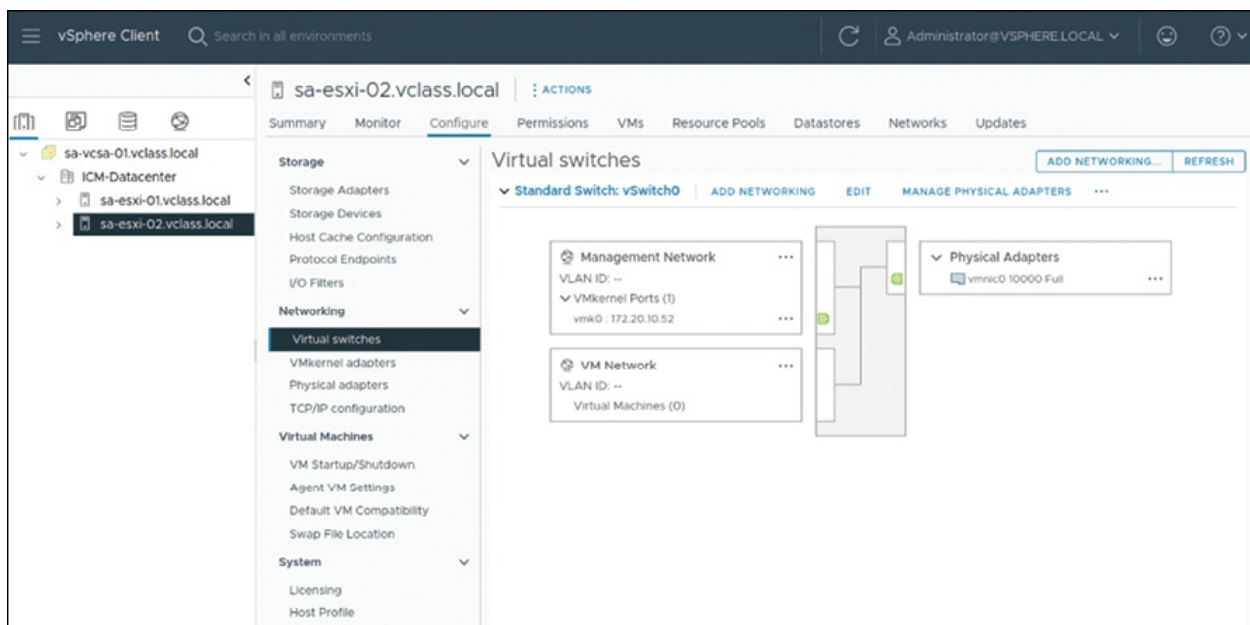


Figure 5.4: Viewing standard switches

(Source: VMware)

Administrators can create new standard switches or modify existing ones on

an ESXi host using the **vSphere Client** or **VMware Host Client**.

With these tools, the administrator can:

- Add uplinks (physical network adapters) to the standard switch.
- Define new port groups for virtual machine or VMkernel traffic.
- Configure the switch or port groups' security, traffic shaping, and failover settings.

Adding standard switches provides flexibility in tailoring the network setup for specific workloads, ensuring optimized connectivity and performance for VMs and management operations.

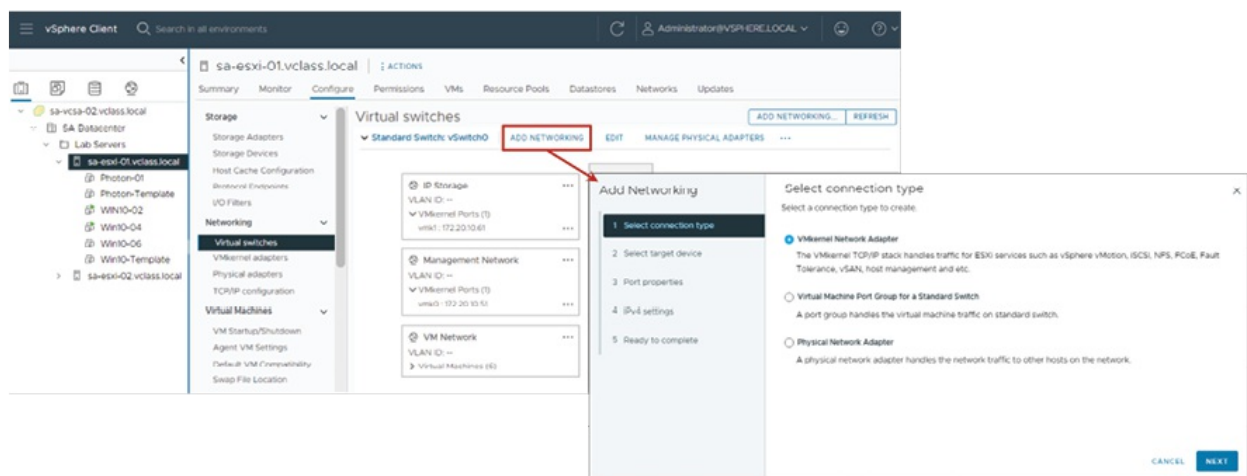


Figure 5.5: Adding standard switches

(Source: VMware)

VMkernel adapter properties

The **VMkernel adapters** pane provides essential information about each VMkernel interface, including:

- The *name* of the adapter.
- The *virtual switch* it resides on.
- The *IP address* assigned to it.
- The *services* enabled on the adapter.

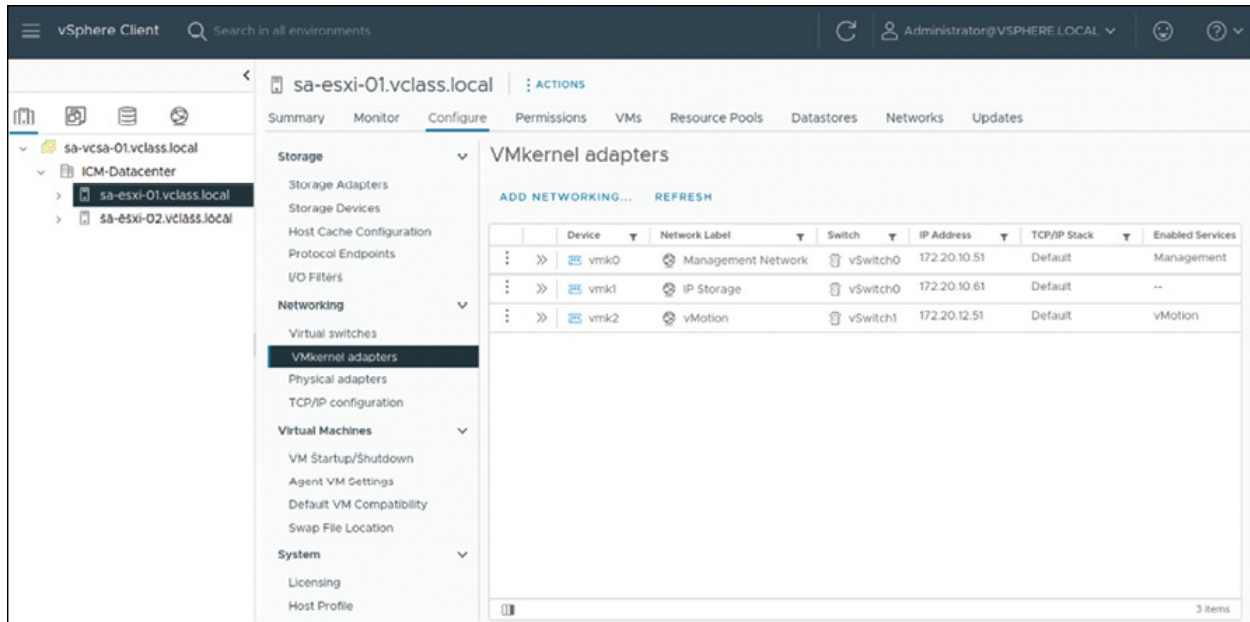


Figure 5.6: VMkernel adapters

(Source: VMware)

Enabled services on VMkernel adapters

You can activate specific services for each VMkernel adapter based on its role in the network infrastructure:

- **vMotion:** Enables the adapter to handle vSphere vMotion traffic, allowing live migrations of virtual machines between hosts.
- **Provisioning:** Supports data transfers for tasks such as VM cold migrations, cloning, and snapshot migrations.
- **Fault tolerance logging:** Manages traffic for vSphere Fault Tolerance to ensure synchronized operation of primary and secondary VMs.
- **Management:** Handles management traffic between the host and vCenter Server.
- **vSphere Replication:** Manages outgoing replication data sent from the source ESXi host to the vSphere Replication server.
- **vSphere Replication NFC:** Handles incoming replication data on the target site during replication processes.
- **vSAN:** Supports vSAN traffic for distributed storage operations.
- **vSphere Backup NFC:** Dedicated to backup traffic using the NFC protocol.

- **NVMe over TCP:** Manages NVMe over TCP storage traffic when the adapter is enabled.
- **NVMe over RDMA:** Manages NVMe over RDMA storage traffic, ensuring high-speed connectivity for RDMA-enabled storage solutions.

These properties and configurations ensure efficient resource allocation and proper traffic routing for various network operations in your vSphere environment, as follows:

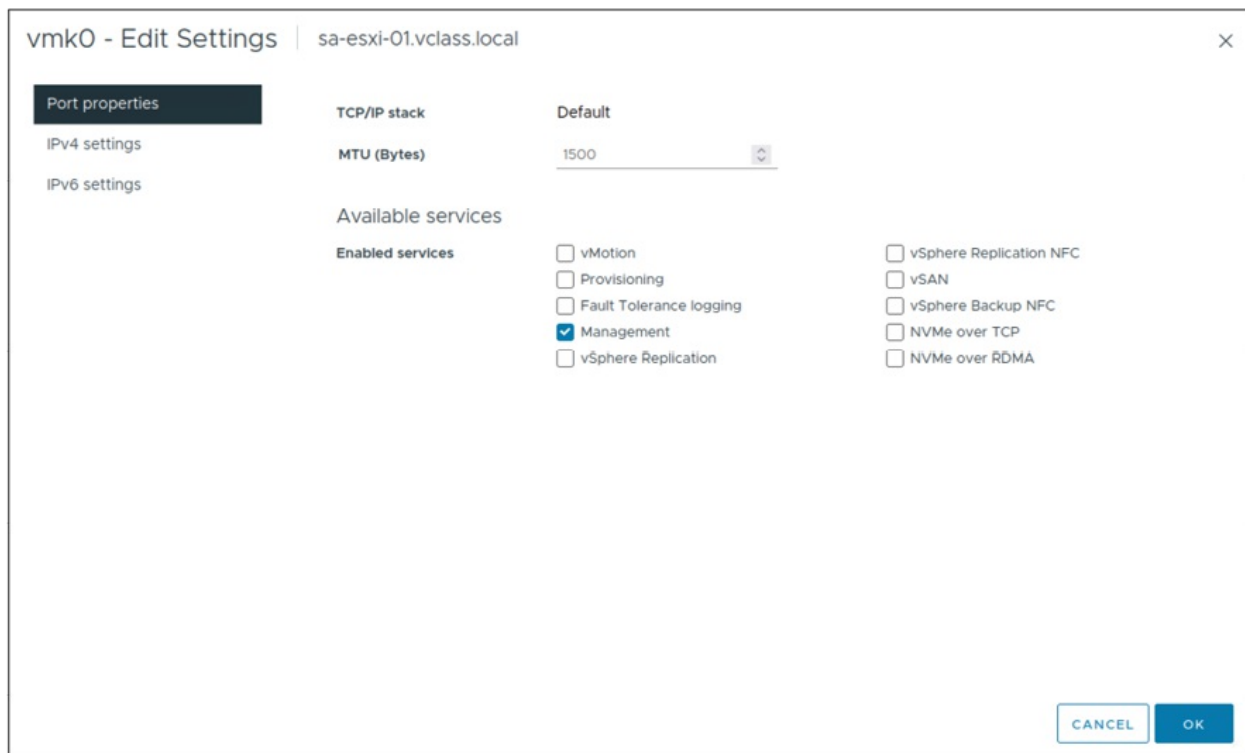


Figure 5.7: vmk0 - Edit Settings

(Source: VMware)

The **Physical adapters** pane provides key details about each network adapter on an ESXi host, including:

- **peed:** The data transfer rate of the adapter.
- **Duplex:** Whether the adapter operates in full-duplex (simultaneous send/receive) or half-duplex mode.
- **Connected Networks:** Displays the networks associated with the adapter.

Let us discuss the best practices.

While it is possible to manually configure the speed and duplex settings, *auto-negotiation* is the recommended practice. This ensures that the adapter dynamically aligns with the optimal settings for the connected switch, minimizing the chances of misconfiguration or performance bottlenecks.

If specific traffic rate compliance is required, you can manually adjust the speed and duplex settings, although this should be done with caution and in alignment with network requirements, as shown:

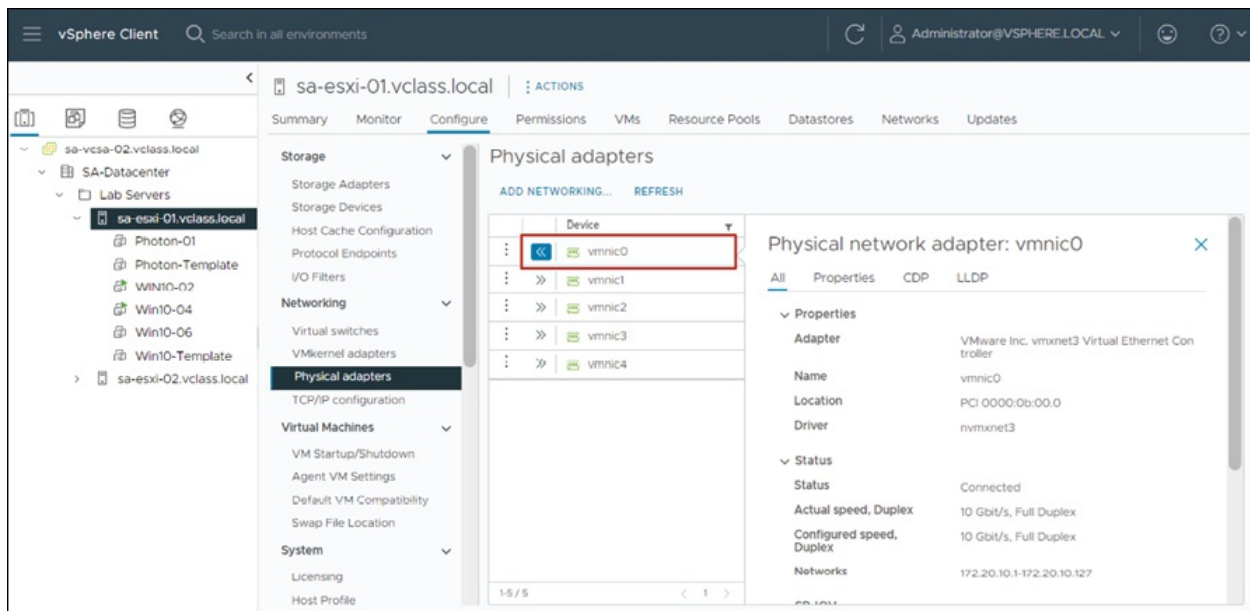


Figure 5.8: Network adaptor properties

(Source: VMware)

Mastering distributed switches

A distributed switch offers a centralized method of controlling virtual networks by functioning as a single virtual switch across several connected hosts. Distributed switches have several significant benefits over standard switches, including:

- **Centralized network administration:** Using vCenter to enable centralized configuration and control streamlines data center management.
- **Granular policy and statistics control:** vCenter assigns distributed

switch ports statically, enabling more accurate network policy and performance metric monitoring and adjustment.

Distributed switches are an effective tool for simplifying and improving networking in virtualized systems because of these advantages.

The following figures illustrate the vSphere distributed switch:

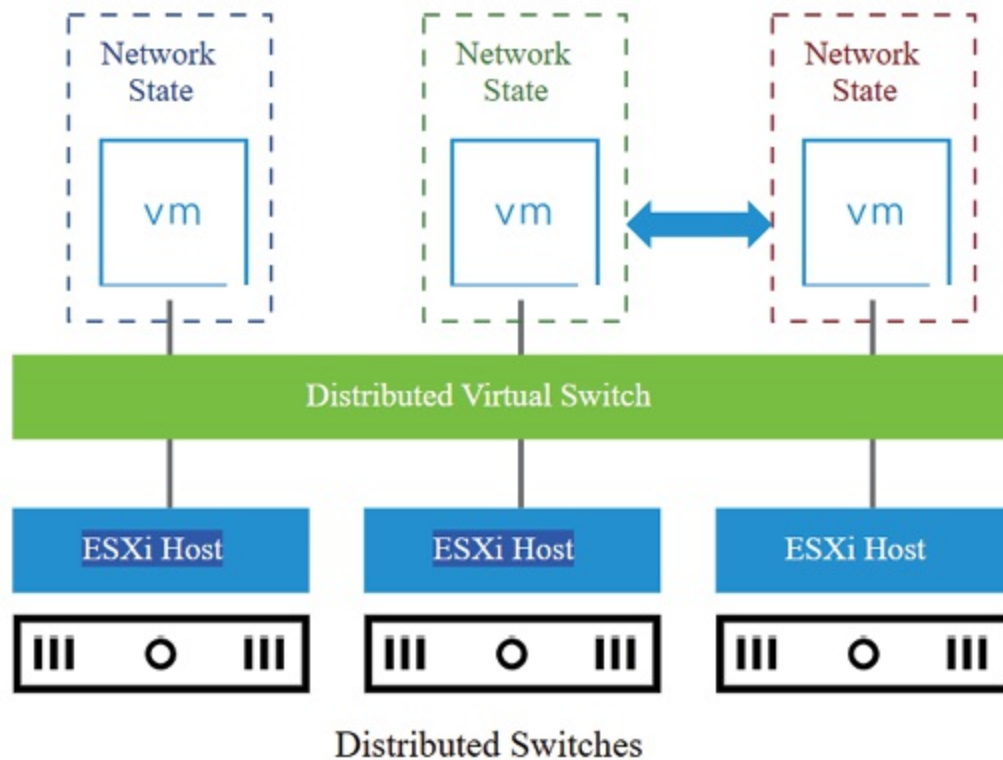


Figure 5.9: Distributed switches

(Source: VMware)

A *distributed switch*, managed by vCenter, provides a unified and consistent virtual networking configuration across all ESXi hosts in a data center.

The architecture consists of two planes:

- **Control plane** (resides in vCenter):
 - Configures distributed switches, distributed port groups, distributed ports, uplinks, NIC teaming, and switch settings.
 - Coordinates the migration of ports and manages switch configurations.
- **I/O plane** (resides on ESXi hosts):

- Implemented as a hidden virtual switch in the VMkernel.
- Manages the I/O hardware and forwards packets on the host.
- The hidden virtual switches are created under vCenter's oversight.

Key components of distributed switch architecture:

- **Distributed ports:** Logical ports to connect networking entities such as VMs or VMkernel interfaces. The state is stored in the vCenter database.
- **Distributed port group:** Groups distributed ports to simplify configuration. Each port can also have unique settings.
- **Uplinks:** Abstractions of physical NICs (vmnics) on multiple hosts to a single distributed switch. Comparable to vmnics on standard switches.

For communication between VMs on different hosts, their uplinks must exist within the same *broadcast domain*.

The *distributed switch architecture* centralizes network management, simplifies configuration, and ensures consistency across the data center:

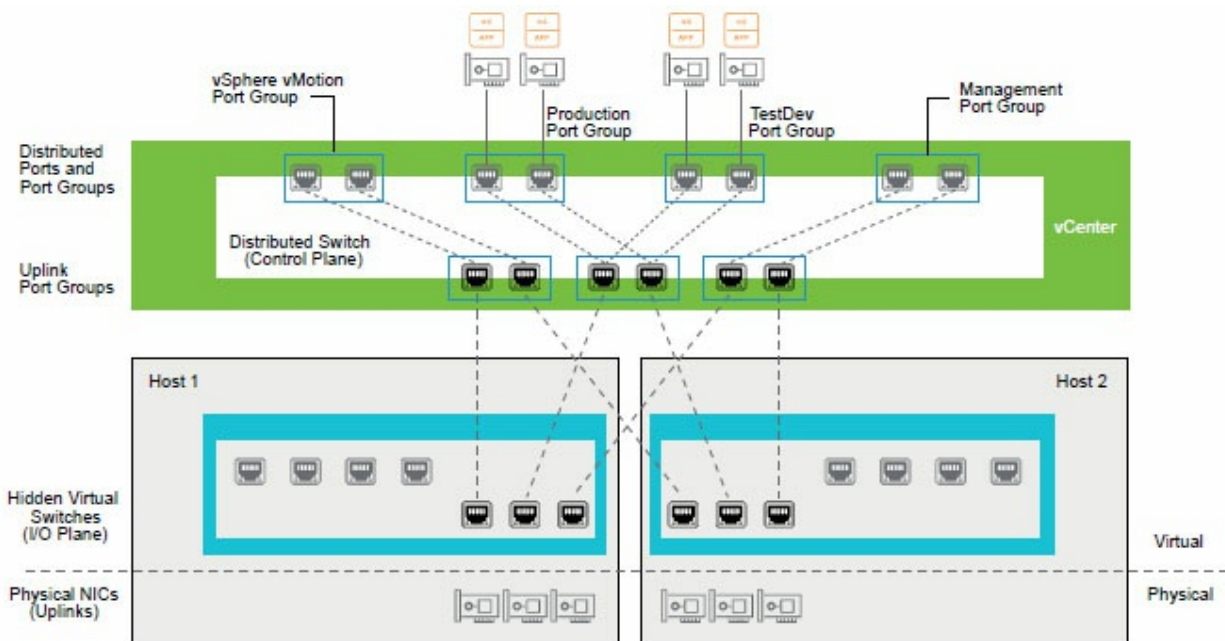


Figure 5.10: Distributed switches architecture

(Source: VMware)

Distributed switches topology

In the vSphere Client, readers can view the distributed switch configuration

by accessing the **Topology pane** under the **Configure** tab. This view provides a visual representation of the distributed switch setup, allowing readers to see the configuration and structure of the distributed switch, including its connections to ESXi hosts, port groups, and uplinks. This helps in managing and monitoring the virtual network infrastructure more effectively at the data center level.

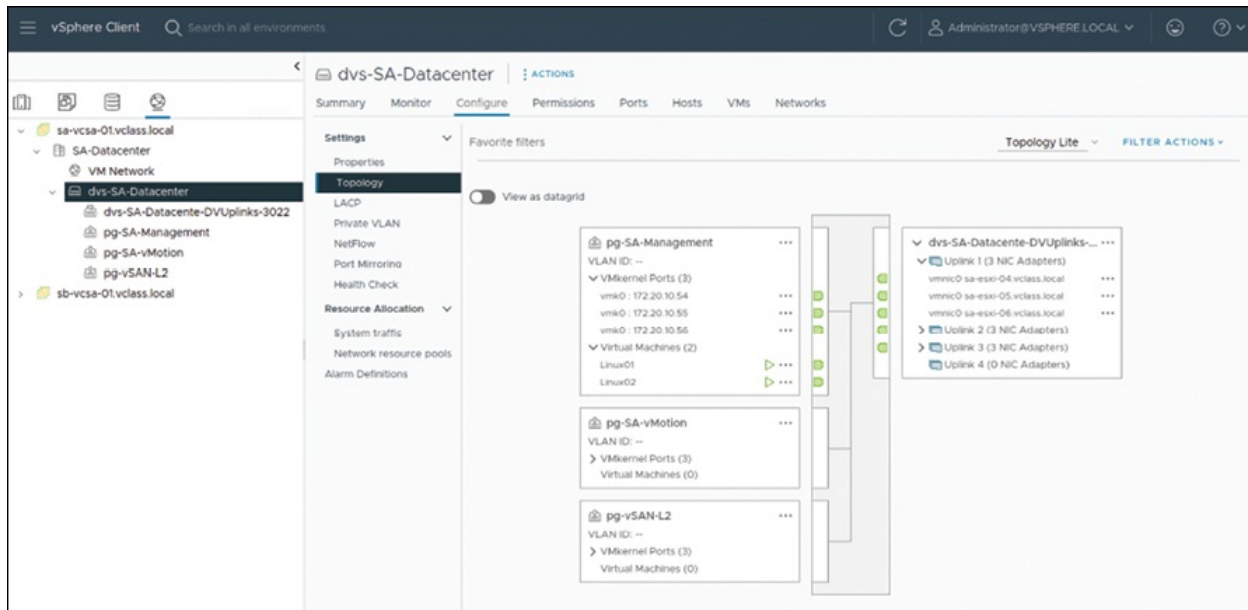


Figure 5.11: Viewing distributed switches

(Source: VMware)

Understanding and configuring discovery protocols

Discovery protocols help network administrators gather information about physical or virtual switches. vSphere supports two protocols:

- **Cisco Discovery Protocol (CDP):** Used with vSphere standard and distributed switches connected to Cisco physical switches. It broadcasts connected device information at Layer 2 and has been supported in vSphere since version 4.0.
- **Link Layer Discovery Protocol (LLDP):** A vendor-neutral protocol supported only on distributed switches. It follows the IEEE 802.1AB standard.

Standard switches can use CDP, while distributed switches can use either CDP or LLDP. These protocols help network administrators discover device

capabilities and neighbors, which is useful for troubleshooting network issues.

With CDP or LLDP enabled, a virtual switch can be configured in the following modes:

- **Listen:** Receives information from the physical switches.
- **Advertise:** Sends information to the physical switches.
- **Both:** Sends and receives information between the virtual and physical switches.

These protocols allow the vSphere Client to identify properties of physical switches, such as switch name, port number, speed, and duplex settings. They also enable the sharing of information about physical adapters and ESXi host names with compatible switches.

CDP can be enabled on a standard switch using the **esxcli** command, as follows:

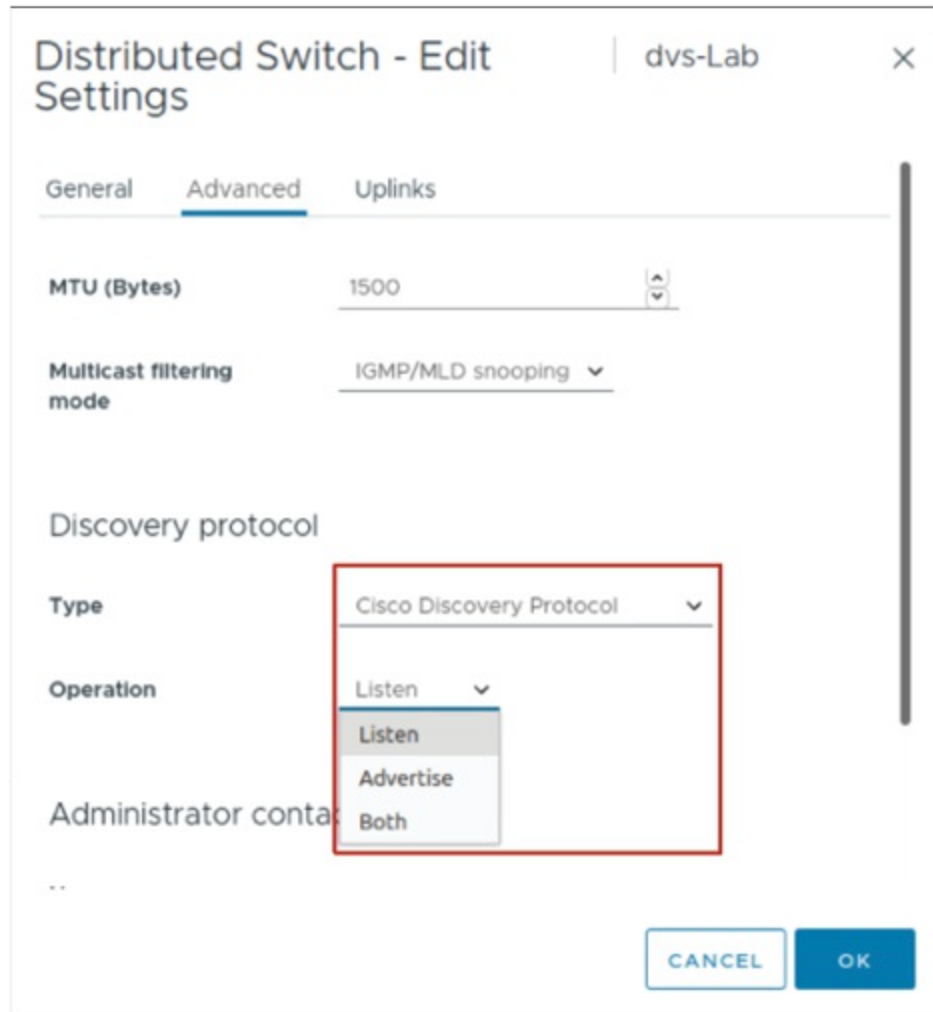


Figure 5.12: Configuring CDP or LLDP

(Source: VMware)

Port binding overview

Port binding determines how and when a **virtual NIC (vNIC)** is assigned to a port on a virtual switch. This configuration is done at the distributed port group level and includes two main types of port binding:

- **Static binding (default):**
 - vCenter assigns a permanent port to the VM or VMkernel interface.
 - The assigned port remains reserved and guarantees connectivity even if the VM reboots.
 - The port is only disconnected when the VM is removed from the port group.

- **Port allocation options for static binding:**

- **Elastic (default):** A new set of eight ports is created when all ports are assigned.
- **Fixed:** No additional ports are created once all ports are assigned.

Static binding is the recommended method for general use as it always ensures a dedicated port for the VM.

- **Ephemeral binding:**

- The ESXi host (not vCenter) assigns a port to the VM when powered on and connected, and the port is deleted when the VM is powered off or its NIC is disconnected.
- Ephemeral binding is flexible as it allows you to modify VM network connections when vCenter is unavailable, making it useful for recovery scenarios.
- However, **ephemeral** binding should be used only for recovery purposes when you need to provision ports directly on an ESXi host, bypassing vCenter. It is not recommended for regular operations, as it lacks the permanency of static binding.

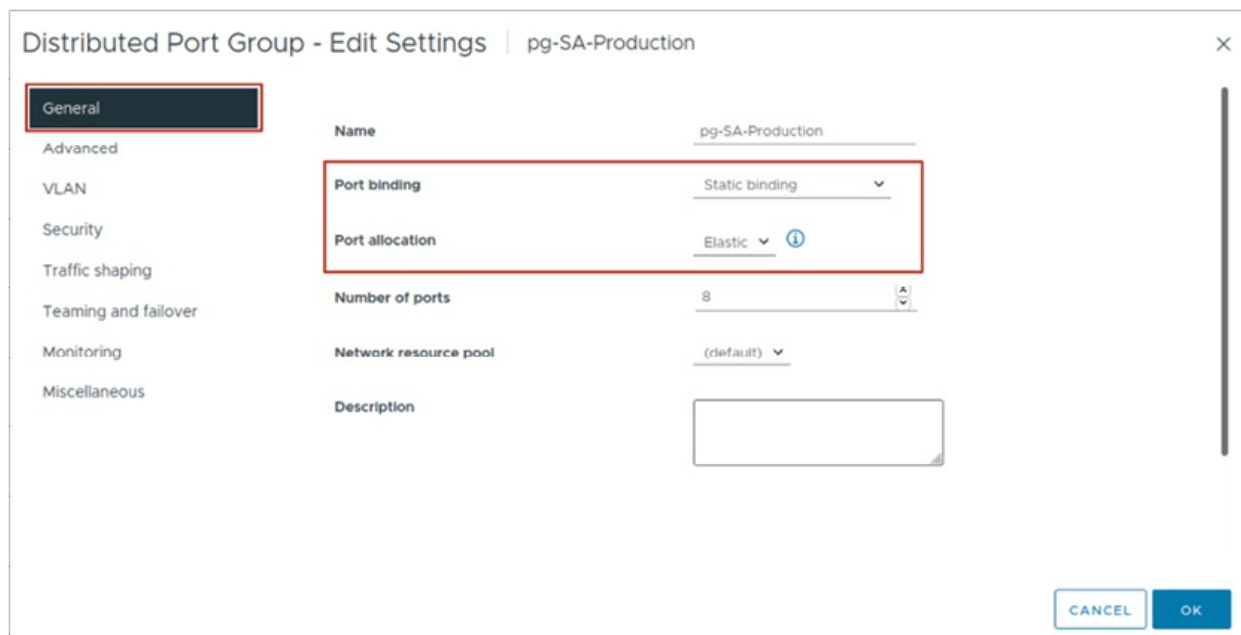


Figure 5.13: About port binding

(Source: VMware)

Inbound traffic shaping

Inbound traffic shaping, supported only by *distributed switches*, controls the flow of traffic from the physical network to the virtual switch and, ultimately, to the **virtual machines (VMs)**. It allows administrators to manage the amount of incoming data (ingress) that flows into the virtual switch, providing finer control over network traffic and ensuring the network is not overwhelmed.

While *outbound (egress) traffic shaping* is available on both standard and distributed switches, *inbound (ingress) traffic shaping* is exclusive to distributed switches. By configuring inbound traffic shaping, you can apply limits on the rate at which data is received from the physical network, improving the overall performance and stability of the network infrastructure. In essence, inbound traffic shaping helps prevent congestion and ensures smooth traffic flow from external networks into the virtual environment, enhancing VM performance and maintaining efficient resource utilization.

Distributed Port Group - Edit Settings | pg-SA-Production

General
Advanced
VLAN
Security
Traffic shaping
Teaming and failover
Monitoring
Miscellaneous

Ingress traffic shaping ⓘ

Status	Enabled ▾
Average bandwidth (kbit/s)	102400 ▴ ▾
Peak bandwidth (kbit/s)	204800 ▴ ▾
Burst size	102400 ▴ ▾

Egress traffic shaping ⓘ

Status	Disabled ▾
Average bandwidth (kbit/s)	100000 ▴ ▾
Peak bandwidth (kbit/s)	100000 ▴ ▾
Burst size (KB)	102400 ▴ ▾

Figure 5.14: Configuring traffic shaping

(Source: VMware)

Physical NIC Load balancing

The *Physical NIC Load* method is a load balancing policy available only on *distributed switches* and is the recommended policy for distributed port groups. It optimizes physical NIC capacity in a NIC team by distributing I/O flows based on the load of the physical network interface cards (NICs).

This method works as follows:

- It monitors the *mean send/receive utilization* of each uplink.
- If the utilization of an uplink exceeds 75% over a 30-second period, the method moves the I/O flow to another uplink in the team.
- This is done to prevent any single NIC from becoming overloaded, helping to balance the load efficiently across available uplinks.

To enable this method, you must select **Route based on physical NIC load** in the distributed port group settings. This method is not enabled by default, so it requires manual configuration.

The following are the advantages:

- **Low algorithm overhead:** There is very little effect on system performance because the distributed switch only computes uplinks once per VM.
- **Load reduction:** By automatically rerouting I/O flows, it effectively lessens the load on highly used uplinks.
- **No adjustments to the hardware switches are necessary:** There is no need to change the physical switch configurations because this load balancing method is fully controlled within the distributed switch.

Let us discuss the drawback.

Bandwidth limitation: The uplinks linked to the distributed switch limit the amount of bandwidth that virtual machines can use. Insufficient uplinks may cause bandwidth to become a constraint.

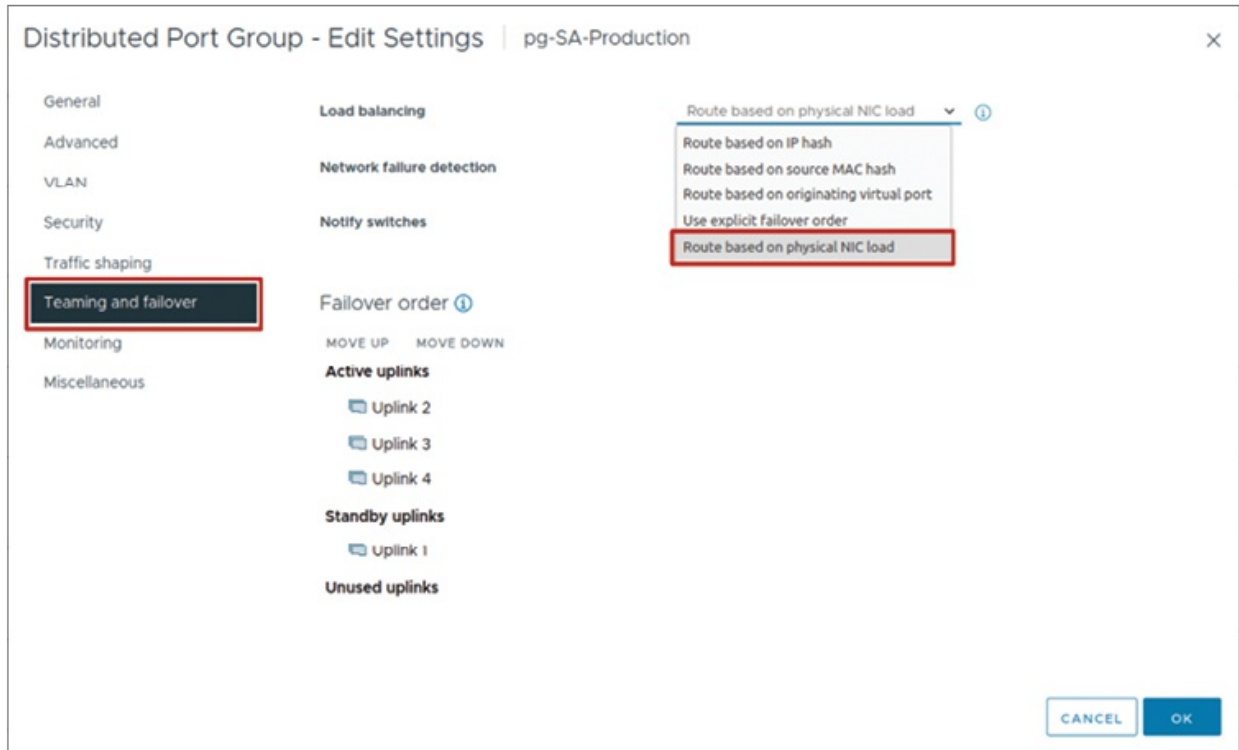


Figure 5.15: Load balancing physical NIC load

(Source: VMware)

Standard vs. distributed switches

Standard switches are manually configured on each ESXi host and are configured separately at the host level. However, distributed switches are configured centrally at the data center level in the vCenter Server, providing uniform control for all related hosts. Particularly in larger settings, distributed switches offer substantial advantages over standard switches due to their centralized management and extra functions.

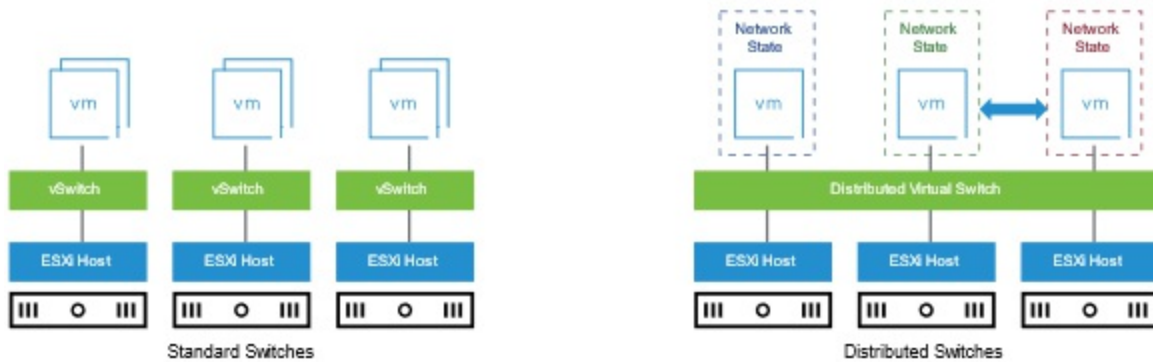


Figure 5.16: Standard vs. distributed switches

(Source: VMware)

The following table highlights how distributed switches streamline network management, improve scalability, and simplify operations compared to standard switches:

Feature	Standard switch	Distributed switch
Configuration level	Configured at the host level.	Configured at the data center level via vCenter.
Management	Managed individually per ESXi host.	Centrally managed for all associated hosts.
Port management	Ports are local to each host.	Ports are centrally managed and migrate with VMs.
vSphere vMotion	Requires manual network consistency across hosts.	Simplified as port statistics/policies migrate with VMs.
Administration	Time-consuming for large environments.	Simplifies data center administration.
Troubleshooting	Requires checking each host's configuration.	Centralized monitoring and management simplify debugging.

Table 5.1 : Standard vs. distributed switches

The following are the common features available in both *standard* and *distributed switches*:

- Layer 2 switch
- VLAN segmentation
- 802.1Q tagging
- IPv4 and IPv6 support

- NIC teaming
- Outbound traffic shaping

The following are the exclusive features of **distributed virtual switch (DVS)** that are not available in standard switches:

- Inbound traffic shaping
- Configuration backup and restore
- Private VLANs
- **Link Aggregation Control Protocol (LACP)**
- Data center-level management
- vSphere vMotion migration of network state
- **Network I/O Control (NIOC)**
- Per-port policy settings
- Port state monitoring
- NetFlow
- Port mirroring
- Support for NSX

Key highlights of distributed switch features:

- **vSphere vMotion migration of network state:** Distributed switches track the virtual network state, including counters and port statistics, during VM migration. This ensures a *consistent view* of the network interface across hosts.
- **Simplified monitoring and troubleshooting:** Tracking the network state during vMotion migration simplifies *network monitoring* and *troubleshooting* when VMs move between hosts.

Sculpting network policies with finesse

Networking policies on virtual switches allow administrators to configure key virtual network properties, including *security*, *performance*, and *availability*. These policies ensure the efficient and secure operation of virtualized environments.

- **Policy application levels:** Networking policies vary based on the virtual

switch type and can be applied as follows:

Virtual switch type	Default policy level	Override level
vSphere Standard Switch	Standard switch level	Port group level
vSphere Distributed Switch	Distributed port group level	Individual port level

Table 5.2: Policy application levels

The following are the key networking policies:

- **Security policies:** Protect against network vulnerabilities, such as *MAC address impersonation and unwanted port scanning*, ensuring safe and controlled communication.
- **Traffic shaping policies:** Enable administrators to **regulate the traffic rate** to and from VMs or groups of VMs, providing fine control over bandwidth utilization. This is particularly useful for prioritizing critical workloads.
- **Teaming and failover policies:** Define how network traffic is distributed across physical adapters and determine *failover strategies* if an adapter fails. This ensures both *load balancing and resilience* in the event of hardware issues.

By carefully setting and managing these policies, administrators can achieve a balanced, secure, and highly available virtual network environment.

Security policies

Security policies on virtual switches control how network traffic is handled, protecting against potential threats like unauthorized traffic interception or MAC address spoofing. These policies can be configured at the standard switch and port group levels.

The following are the key security policies:

- **Promiscuous mode:**
 - **Default setting:** Reject
 - **Description:** Determines whether a virtual switch or port group can forward all traffic, regardless of the destination.
 - **Usage:** Accept only in specialized cases, such as enabling a network analysis tool (e.g., packet sniffers) in a VM.

- **MAC address changes:**
 - **Default setting:** Reject
 - **Description:** Controls whether inbound traffic is accepted if the guest operating system alters the MAC address of its virtual NIC.
 - **Usage:** Reject to protect against unauthorized MAC address spoofing. This is only acceptable if an application requires MAC address modifications, such as guest OS-based firewalls.
- **Forged transmits:**
 - **Default setting:** Reject
 - **Description:** Determines whether outbound traffic is allowed when the guest operating system alters the source MAC address.
 - **Usage:** Reject to prevent rogue systems from sending unauthorized traffic. Accept only for applications that intentionally modify the source MAC address.

The following are some of the general recommendations:

- For most environments, retain the *default setting (Reject)* for all security policies to minimize vulnerabilities.
- Change to *Accept* only for specific use cases that require these exceptions, such as:
 - Promiscuous mode for VMs running intrusion detection or packet-sniffing applications.
 - MAC address changes and forged transmits for applications or systems that rely on custom MAC address configurations.

These security policies play a vital role in protecting the virtualized network from potential attacks, such as network traffic interception or MAC address spoofing, ensuring a robust and secure infrastructure, as shown:

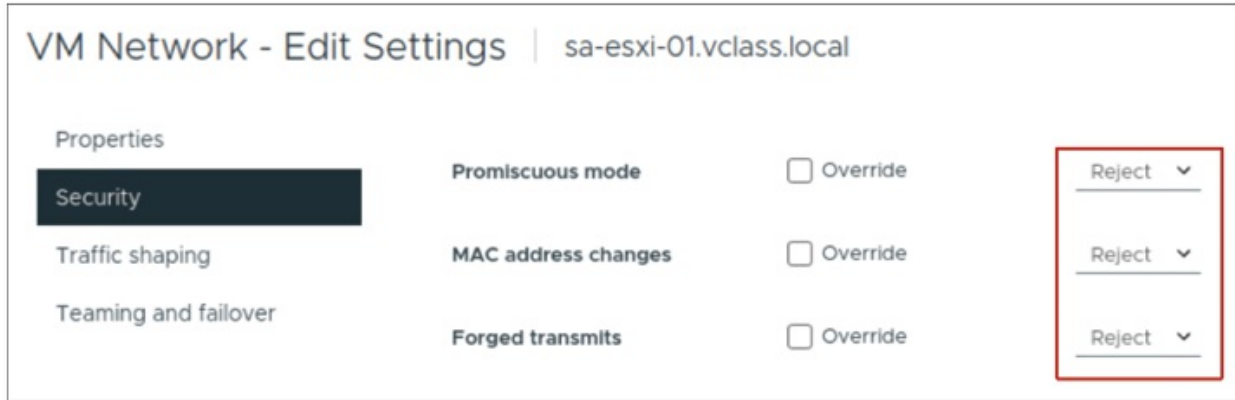


Figure 5.17: Configuring security policies

(Source: VMware)

Traffic shaping policies

Network traffic shaping is a tool for regulating the bandwidth consumption of virtual machines, ensuring fair usage across the network. By configuring parameters like **average rate**, **peak rate**, and **burst size**, administrators can fine-tune network performance based on organizational requirements. By default, traffic shaping is turned off, but it can be turned on for use cases that call for regulated network performance.

The following are some of the key traffic shaping parameters:

- **Average bandwidth (Kbps)**
 - Defines the allowed average data transfer rate in kilobits per second over time.
 - Represents the baseline bandwidth available to the port.
- **Peak bandwidth (Kbps)**
 - Specifies the maximum data transfer rate for a port during traffic bursts.
 - Allows the port to exceed its average bandwidth temporarily when needed, up to this limit.
- **Burst size (KB)**
 - Establishes the maximum data volume that can be sent in a single burst.

- Ports can accumulate unused bandwidth (burst bonus) to transmit at faster rates, when necessary, within this size limit.

Let us discuss a use-case example.

While maintaining a constant average bandwidth to avoid network saturation, managers may set greater peak bandwidth and burst size values for high-priority applications to guarantee excellent performance during traffic spikes. This strategy guarantees a balanced and effective network environment by restricting bandwidth usage and providing flexibility for unexpected spikes in demand.

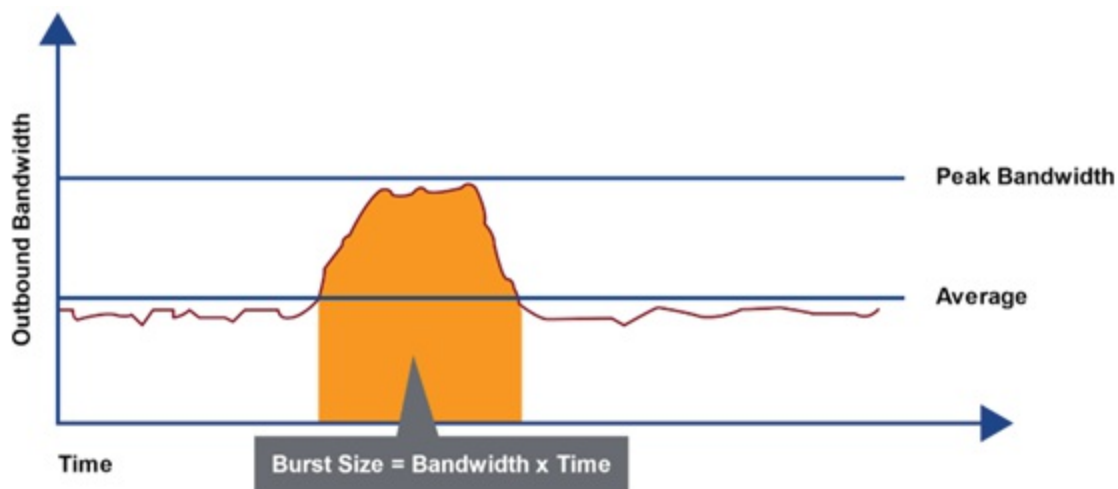


Figure 5.18: Traffic shaping policies

(Source: VMware)

Outbound traffic shaping

Outbound traffic shaping allows administrators to regulate the network bandwidth consumed by virtual machines as they send data through a *standard switch* to the physical network.

- **Parameters:**

- **Average bandwidth:** The consistent rate of data transfer allowed (Kbps).
- **Peak bandwidth:** The maximum rate permitted during bursts of traffic (Kbps).
- **Burst size:** The amount of data (KB) allowed in a single burst when

additional bandwidth is available.

- **Scope:**
 - Applies to each virtual NIC connected to a standard switch.
 - Shapes *outbound traffic only*, from the VM to the physical network.
- **Control mechanism:**
 - Traffic shaping on a standard switch is activated to manage outgoing traffic, ensuring network resources are shared efficiently among VMs.
 - For *inbound traffic shaping*, administrators must use external tools like *load-balancing systems* or rate-limiting features on physical routers.

Let us discuss a use case:

When multiple VMs are transmitting data simultaneously, traffic shaping ensures that no single VM monopolizes network resources, maintaining consistent performance across the environment.

This policy is particularly valuable in *resource-constrained environments*, helping maintain a balanced and predictable network throughput.

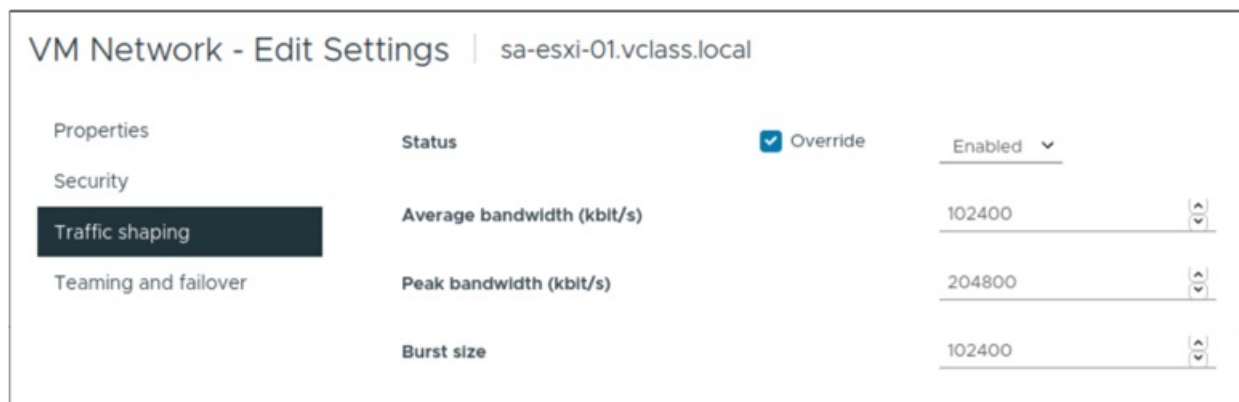


Figure 5.19: Configuring outbound traffic shaping

(Source: VMware)

NIC teaming and failover

NIC teaming allows you to combine multiple physical NICs (uplinks) into a team to enhance network bandwidth and provide redundancy for virtual

switches or port groups.

The following are the key features:

- **Load balancing:**
 - Network traffic is distributed across active NICs in the team according to the selected load-balancing policy.
 - Virtual switches only balance *outbound traffic*; inbound traffic is managed by the physical switch.
- **Redundancy with failover:**
 - If a NIC fails, traffic is rerouted to standby NICs based on the failover order.
 - This setup ensures uninterrupted network connectivity.
- **Configurable failover order options:**
 - **Active:** NICs in this group handle network traffic during normal operations.
 - **Standby:** Used only if an active NIC fails, ensuring redundancy.
 - **Unused:** NICs in this group are excluded from the failover process but can be reconfigured as needed.

Let us discuss an example of a use case:

A team of three NICs can be configured where two NICs are *Active* to handle traffic, and the third is *Standby* for redundancy. If one active NIC fails, the standby NIC takes over, preventing network disruption.

Administrators can ensure high availability, optimized performance, and fault tolerance for virtualized network environments by effectively managing NIC teaming and failover.

The following figures illustrate the NIC teaming and failover configuration:

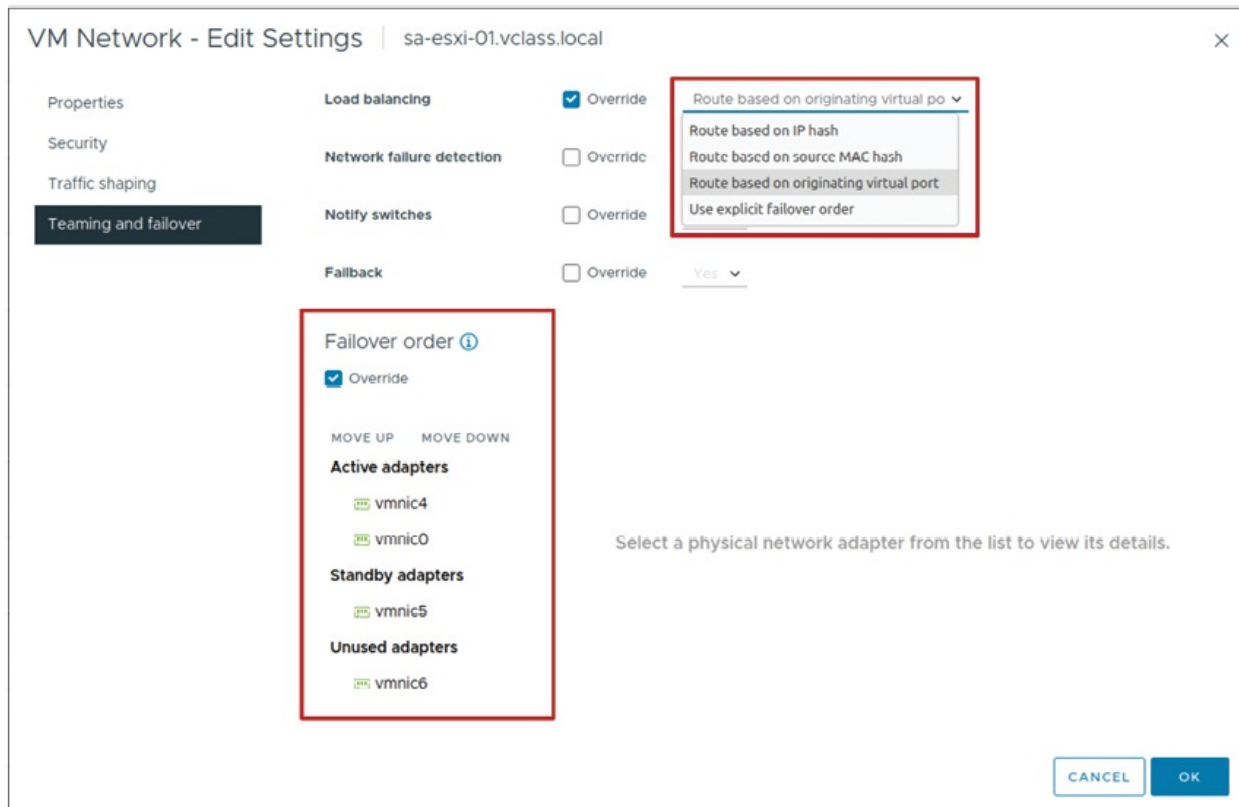


Figure 5.20: Configuring NIC teaming and failover

(Source: VMware)

Load balancing methods for virtual switches

VMware vSphere offers several load balancing methods to optimize outbound traffic from virtual machines to physical NICs. Each method has specific characteristics, advantages, and disadvantages suited to different use cases:

- **Originating virtual port ID:** Maps a VM's outbound traffic to a specific physical NIC based on the virtual port ID.

The following are the advantages:

- Low overhead; virtual switch calculates uplinks only once.
- Even traffic distribution if the number of virtual NICs exceeds the number of physical NICs.
- No physical switch configuration is required.

The following are the disadvantages:

- The traffic load on uplinks is not considered, potentially overloading certain uplinks.
- Bandwidth for a VM is limited to the speed of a single uplink unless multiple virtual NICs are used.
- **Source MAC hash:** Maps a VM's traffic to a physical NIC based on its MAC address.

The following are the advantages:

- Stable uplink assignment, powering a VM on or off does not change the uplink.
- Compatible with all physical switches.
- No changes required on the physical switch.

The following are the disadvantages:

- Limited bandwidth to the speed of a single uplink unless the VM uses multiple MAC addresses.
- Algorithm overhead is higher compared to the originating virtual port ID method.
- Traffic is not evenly balanced across all physical NICs.
- **Source and destination IP hash:** Selects a NIC based on the source and destination IP addresses of each packet.

The following are the advantages:

- Provides better traffic distribution than other methods.
- VMs communicating with multiple IPs can achieve higher throughput.

The following are the disadvantages:

- Requires 802.3ad link aggregation or EtherChannel on physical switches.
- Algorithm overhead is the highest among all methods.
- Complex to troubleshoot, and physical network changes are required.

Detecting and handling network failure

The VMkernel ensures network uptime by monitoring and detecting failures through *link status checks* and *beacon probing* (if enabled). These checks are performed every second. When a network failure occurs, the VMkernel updates the physical switches about MAC address location changes and implements failover based on configurable parameters.

The following are the key features:

- **Failure detection:**
 - **Link status monitoring:** Detects issues like cable disconnections or switch power failures.
 - **Beacon probing:** Sends and listens for probe packets to detect upstream network failures that link-status monitoring alone cannot identify. (Requires specific network topology).
- **Failover management:** The VMkernel uses **explicit failover order**, prioritizing the top uplink in the active adapter list for traffic rerouting.
- **Failback behaviour:**
 - **Failback Yes:** A recovered adapter is restored to active duty immediately, replacing the standby adapter.
 - **Failback No:** A recovered adapter remains inactive until another active adapter fails.
- **Switch notifications:** The VMkernel informs physical switches about virtual NIC connections and failover events, reducing latency during events like failovers and vSphere vMotion operations.

The following are the limitations:

- *Link status monitoring* cannot detect configuration issues like blocked ports due to Spanning Tree Protocol or misconfigured VLANs.
- Avoid enabling switch notifications if VMs in the port group use Microsoft **Network Load Balancing (NLB)** in *unicast mode*. Multicast mode is unaffected.

By leveraging these mechanisms, VMware ensures robust failover capabilities and optimized network performance while minimizing downtime during network issues.

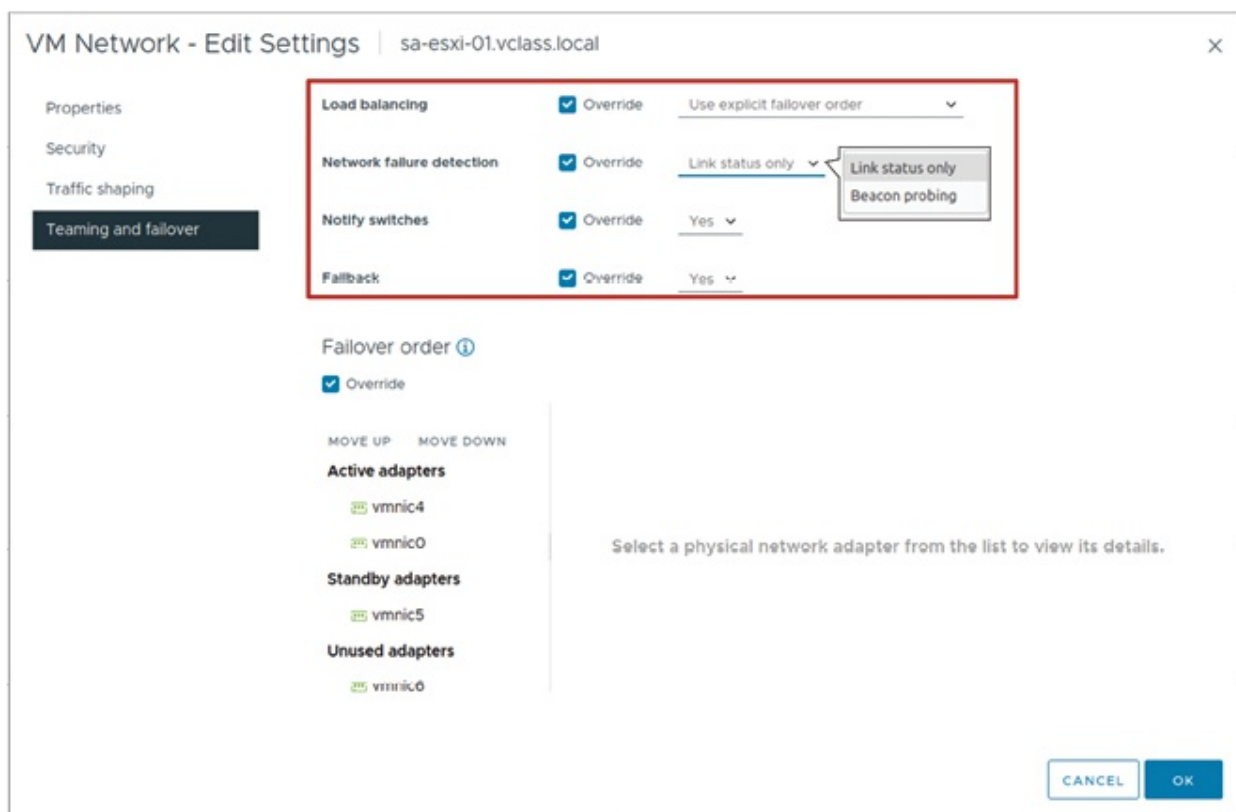


Figure 5.21: Detecting and handling network failure

(Source: VMware)

Conclusion

In this chapter, we explored the foundational and advanced aspects of virtual networking in VMware vSphere, which also highlights the important functions of distributed and standard switches as well as the complex rules governing networking setups. Readers can now develop, implement, and optimize networking solutions for small- and enterprise-level contexts by grasping the ideas and methods covered in this chapter. Readers have established a strong basis for effective network administration in the virtualized infrastructure, from setting up virtual switches to controlling sophisticated features like NIC teaming and traffic shaping.

Now that readers have a solid understanding of networking, it is time to move on to storage, another essential element of VMware systems. We will explore the core storage technologies that support virtualized activities in [Chapter 6](#),

Managing Storage in vSphere. Discover the refinements of datastores, investigate various storage technologies, and develop practical experience implementing *iSCSI* and *Fibre Channel* storage solutions.

In the next chapter, we will enable readers to guarantee the scalability, performance, and dependability of the vSphere storage system, whether readers are managing NFS storage or building VMFS datastores.

Points to remember

- Virtual switches support connections like VM ports for virtual machines, VMkernel ports for host services, and physical uplinks for external network communication.
- A standard switch operates locally on a single ESXi host, providing basic virtual networking capabilities for smaller environments.
- A distributed switch centralizes management, enabling a consistent networking configuration shared across multiple hosts via vCenter. It is ideal for larger, more complex environments.
- Virtual switches allow configuring security, traffic shaping, and NIC teaming policies to enhance performance, reliability, and failover management.
- Policies defined at a standard switch level can be overridden at the port group level, while policies on a distributed switch can be further customized at the individual port level.
- Distributed switches streamline operations and provide advanced networking features, making them suitable for growing infrastructures.
- By default, traffic shaping is turned off, but it can be turned on and network performance can be fine-tuned by configuring parameters like average rate, peak rate and burst size.

Exercises

1. What are the primary connection types supported by virtual switches in vSphere?

2. How does a standard switch differ from a distributed switch in terms of management and scope?
3. What is the role of VMkernel ports in virtual networking, and what tasks do they enable?
4. Explain how networking policies can be overridden at different levels for standard and distributed switches.
5. What are the advantages of using a distributed switch in a large-scale vSphere environment?
6. How can traffic shaping policies improve network performance in a virtualized environment?
7. Describe the process of creating a distributed switch and its benefits for multi-host environments.

Lab exercises

1. Configuring standard switches:

Objective: Learn how to create and configure a standard switch and port groups for virtual machines.

- **View standard switch configuration:**

Log into the **vSphere Client** | Select an **ESXi host** from the inventory | Navigate to the **Configure** tab | Under **Networking**, select **Virtual switches** to view the existing standard switch configuration.

- **Create a standard switch with a Virtual Machine Port Group:**

In the **vSphere Client**, navigate to the **Networking** section of an ESXi host. | Click **Add Networking** and choose **VM Port Group for a standard switch**. | **Create a new standard switch** or attach it to an existing one. | Assign a name to the port group and specify VLAN settings if required. | Complete the wizard to create the standard switch and port group.

- **Attach virtual machines to the virtual machine port group:**

Select a virtual machine from the inventory. | Edit the VM settings and navigate to **Network adapter**. | Assign the **newly created port**

group to the VM's network adapter. | Power on the VM to test connectivity.

2. Configuring vSphere Distributed Switches:

Objective: Understand how to create and configure distributed switches for multi-host environments.

- **Create a distributed switch:**

In the **vSphere Client**, navigate to **Networking**. | Right-click the data center and select **Distributed switch** | **New distributed switch**. | Specify a name for the switch and select the version. | Configure the number of uplinks and enable any optional features, such as Network I/O Control. | Complete the wizard to create the distributed switch.

- **Add ESXi hosts to the distributed switch:**

Select the distributed switch from the inventory. | Click **Add and Manage Hosts** and select **Add Hosts**. | Choose the ESXi hosts to connect to the distributed switch. | Assign uplinks to the physical NICs on the ESXi hosts. | Migrate the host networking to the distributed switch.

- **Verify distributed switch configuration:**

Navigate to the **Monitor** tab of the distributed switch. | View the **Topology** to ensure all connected hosts and uplinks are correctly displayed. | Test connectivity by pinging VMs or using the **Test Management Network** feature on the ESXi host.

3. Configuring networking policies for standard switches:

Objective: Explore networking policies at the standard switch level.

- **Set security policies:**

Navigate to **Networking** | **Virtual Switches** for the ESXi host. | Select the standard switch or port group and click **Edit Settings**. | Modify **MAC Address Changes**, **Forged Transmits**, and **Promiscuous Mode** under the **Security** section.

- **Configure traffic shaping:**

Edit the standard switch settings. | Enable **Traffic Shaping** and specify settings such as average bandwidth, peak bandwidth, and burst size.

- **Set NIC teaming and failover policies:**

Configure **NIC teaming** by adding multiple uplinks to the standard switch. | Define failover policies such as **Route based on IP hash** or **Explicit Failover Order**.

4. **Advanced networking policies with distributed switches:**

Objective: Dive deeper into distributed switch networking policies for large-scale environments.

- **Set security and traffic shaping policies:**

Navigate to the distributed switch in **Networking**. | Select a port group and edit its settings to configure **Security Policies** (Promiscuous Mode, MAC Address Changes, etc.) and **Traffic Shaping**.

- **Configure Port Mirroring:**

Use the **Port Mirroring** feature to monitor traffic. | Create a new port mirroring session and define the source and destination ports.

- **Override distributed switch policies for specific ports:**

Navigate to a specific port on the distributed switch. | Edit the port settings to override default policies for security, shaping, or teaming.

5. **Analyzing networking configurations:**

Objective: Use vSphere tools to analyze and troubleshoot networking configurations.

- **Monitor distributed switch traffic:**

Navigate to the **Monitor** tab of the distributed switch. | Analyse traffic flows using **Network I/O Control** or other monitoring tools.

- **Check VLAN configurations:**

Ensure proper VLAN tagging is configured at the port group or switch level. | Use tools like **ping** to test VLAN communication between VMs.

- **Troubleshoot network connectivity:**

Use the **Test Management Network** feature in the ESXi host settings to identify connectivity issues. | Check logs in **/var/log/vmkernel.log** for detailed error analysis.

CHAPTER 6

Managing Storage in vSphere

Introduction

In this chapter, we explore the critical role of storage in a virtualized environment. A deep understanding of the available storage options empowers administrators to design and manage a storage solution that aligns with the organization's cost, performance, and manageability requirements. The strong storage features of vSphere serve as the cornerstone for high availability, disaster recovery, and the smooth transfer of VMs across hosts.

This chapter explores the essential storage ideas and features that vSphere provides. To create a scalable and effective virtualized environment, readers will learn how to configure and manage various crucial storage options, including **Fibre Channel (FC)**, iSCSI, VMFS, and NFS datastores.

Let us get started on the path to being an expert in storage management and utilizing the potent storage architecture of vSphere.

Note: VMware is now part of Broadcom and is known as 'VMware by Broadcom'. All references to VMware in this book reflect this change.

Structure

In this chapter, we will cover the following topics:

- Exploring the landscape of vSphere storage technologies
- Navigating the world of vSphere datastores
- Fibre channel components and addressing
- Mastering iSCSI components and addressing
- VMFS datastores creation and management
- NFS datastores configuration and administration

Objectives

The readers will have a comprehensive understanding of vSphere storage technologies and how they are used in real-world virtualized environments by the end of this chapter. To ensure clarity and effectiveness in storage management, readers will gain an understanding of the many types of datastores and the naming standards used for storage devices.

Insights into how multipathing guarantees high availability and excellent performance are provided by this chapter's exploration of the elements and addressing mechanisms of Fibre Channel storage. Additionally, it explores the parts and setup of iSCSI storage, giving readers the skills they need to properly configure iSCSI initiators.

Along with learning how to use NFS storage solutions, readers will also learn how to construct, grow, and delete VMFS datastores. This entails knowing how NFS 3 and NFS 4.1 differ from one another and setting up NFS datastores to meet the storage requirements of the organization.

By grasping these ideas and methods, readers will be able to control and enhance vSphere storage infrastructure, guaranteeing their virtualized environments' scalability, dependability, and superior performance.

Exploring the landscape of vSphere storage technologies

Efficient storage configuration is a cornerstone of any robust virtualized environment. In vSphere, ESXi hosts rely on shared access to datastores for seamless operations, including features like **high availability (HA)**, vMotion,

and fault tolerance.

Datastores in vSphere can be formatted with either **virtual machine file system (VMFS)** or **network file system (NFS)**, depending on the type of storage deployed. ESXi hosts support a wide array of storage technologies, each offering unique benefits suited

The following is an overview of the major storage technologies used in the vSphere environment:

- **FC:** A fast protocol called Fibre Channel was created specifically for **storage area networks (SANs)**. For transmission between nodes, like servers, storage systems, or tape drives, it encodes SCSI commands. The switch-based fabric architecture of Fibre Channel networks allows for smooth communication between numerous nodes. This technology is well known for having low latency and fast throughput, which makes it perfect for applications requiring high performance.
- **FCoE, or FC over Ethernet:** By encapsulating Fibre Channel frames into Ethernet packets, FCoE integrates Fibre Channel traffic with Ethernet networks. Because of this convergence, businesses may operate different kinds of traffic over the same high-speed Ethernet infrastructure, which lowers operational complexity and cabling while preserving great performance.
- **Internet Small Computer System Interface (iSCSI):** A transport technology called iSCSI transfers SCSI commands across common TCP/IP networks. Through initiators like an iSCSI **host bus adapter (HBA)**, which interfaces with iSCSI targets in storage arrays, ESXi hosts can establish connections to distant storage devices via iSCSI. This affordable approach offers strong block-level storage functionality while utilising current IP networks.
- **Direct attached storage (DAS):** Internal or external discs or arrays that are directly connected to the ESXi host via a dedicated connection as opposed to a network are referred to as direct-attached storage. Although this configuration is easy to use and reasonably priced, it does not offer the scalability and flexibility of networked storage.
- **Network attached storage (NAS):** NAS uses the NFS protocol and provides file-based storage via TCP/IP networks. Multiple ESXi hosts

may access shared storage at the file level thanks to NFS datastores, which makes virtualisation capabilities like vMotion and snapshots easier. NFS operates differently from other storage protocols since it does not support SCSI commands.

Improving the performance of storage

High-speed networks can support technologies like iSCSI, NAS, and FCoE, guaranteeing enough bandwidth for demanding applications. These protocols make it possible for various kinds of high-bandwidth traffic to coexist peacefully with sufficient network infrastructure.

For additional details about physical NIC support and the maximum number of supported ports, refer to the **VMware Configuration Maximums** at <https://configmax.broadcom.com/home>.

The following figure illustrates the storage overview:

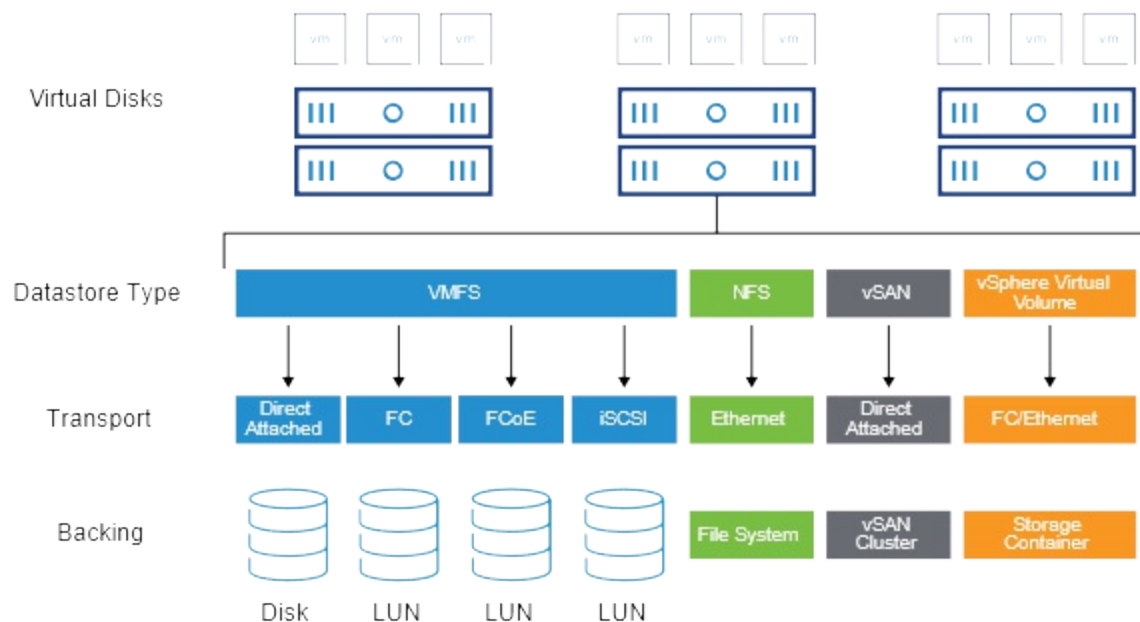


Figure 6.1: Storage overview

(Source: VMware)

vSphere storage device naming conventions

ESXi hosts in vSphere environments use quite a few naming standards to identify storage devices. Administrators can more efficiently control and

communicate with storage systems thanks to these identifiers. Every identifier has a distinct function and offers scalability and flexibility for various storage setups.

The types of storage identifiers are as follows:

- **Runtime name:**
 - **It adheres to the format:** vmhbaN:C:T:L
 - **vmhbaN:** The storage adapter is referred to as.
 - **C:** The channel number.
 - **T:** The desired number.
 - **L:** Logical Unit Number, or LUN.
 - This name is generated at runtime by the host and is easy to use.
 - **Limitations:**
 - The runtime name changes with each reboot.
 - If more HBAs are added to the ESXi host, it can alter.
 - **Use case:** Helpful for managing ESXi host storage with command-line tools or for instant reference.
- **Target:** Represents the storage device's port and address.
 - The target for iSCSI is located using:
 - iSCSI Qualified Name, or IQN: complies with the format iqn.yyyy-mm.Authorisation for naming: uniquename.
 - An additional format for unique identification is EUI (Extended Unique Identifier).
 - This is essential for large-scale or multi-host setups since it guarantees that every storage node (such as an iSCSI target) may be uniquely identified globally.
- **Logical unit number, or LUN:**
 - Discs or groups of storage devices can be uniquely identified by their LUNs.

- Addressed by the SCSI protocol or SAN protocols (such as Fibre Channel and iSCSI) that encapsulate SCSI.
- Guarantees the unique identification of every logical storage unit for effective mapping and administration.

SCSI identifiers that are unique are listed as follows:

- SCSI storage devices on ESXi hosts depend on distinct identities to guarantee appropriate communication and administration.
- For each LUN, the VMkernel needs a distinct identity. The VMkernel creates a unique identity to represent the LUN or disc if the storage device is unable to supply one.

Practical use: Administrators can easily monitor and control storage devices because storage device names are displayed in many vSphere Client panels. Identifiers such as runtime names, targets, and LUNs provide flexibility in various operations, including troubleshooting, performance tuning, and configuring advanced storage features. The following figure illustrates the storage device naming convention:

Adapter	Model	Type	Status	Identifier	Targets
vmhba65	iSCSI Software Adapter	iSCSI	Online	iscsi_vmk(iqn.1998-01.com.v...	1
vmhba1	PIIX4 for 430TX/440BX/MX IDE Controll...	Block SCSI	Unknown	--	1

Runtime Name	Target	LUN	Status
vmhba65:C0:T0:L0	iqn.2005-10.org.freenas.cticmtargets:172.20.10.15:3260	0	Active (I/O)
vmhba65:C0:T0:L2	iqn.2005-10.org.freenas.cticmtargets:172.20.10.15:3260	2	Active (I/O)
vmhba65:C0:T0:L3	iqn.2005-10.org.freenas.cticmtargets:172.20.10.15:3260	3	Active (I/O)
vmhba65:C0:T0:L4	iqn.2005-10.org.freenas.cticmtargets:172.20.10.15:3260	4	Active (I/O)

Figure 6.2: Storage device naming convention

(Source: VMware)

Storage protocol overview in vSphere

In vSphere environments, various storage protocols support different datastore types, each with unique features and compatibility. The choice of protocol impacts functionality such as booting, vSphere vMotion, HA, and **Distributed Resource Scheduler (DRS)**.

The following table illustrates the datastore types and supported features:

Datastore type	Storage protocol	Boot from SAN support	vSphere vMotion support	vSphere HA support	vSphere DRS support
VMFS	Fibre Channel	Yes	Yes	Yes	Yes
	FCoE	Yes	Yes	Yes	Yes
	iSCSI	Yes	Yes	Yes	Yes
	iSER/NVMe-oF (RDMA)	No	Yes	Yes	Yes
	DAS (SAS, SATA, NVMe)	N/A	Yes*	No	No
NFS		No	Yes	Yes	Yes
vSphere Virtual Volumes		No	Yes	Yes	Yes
vSAN Datastore	vSAN	No	Yes	Yes	Yes

Table 6.1 : Datastore types and supported features

Note: Direct-attached storage (DAS) supports vSphere vMotion when used in combination with vSphere Storage vMotion.

Navigating the world of vSphere datastores

A datastore in vSphere is a logical storage unit used to store various data types, including:

- **Virtual machines (VMs)**
- VM templates
- ISO images

The datastores types supported in vSphere are listed as follows:

- **VMFS:** Designed for virtualization, VMFS is a high-performance clustered file system that allows several ESXi hosts to share storage.
- **NFS:** An Ethernet-based file system that provides access to shared storage.
- **vSAN Datastore:** This feature pools local storage devices across ESXi hosts by integrating with VMware vSAN technology. For contemporary applications, it offers scalability and high availability.
- **vSphere Virtual Volumes (vVols):** Offers VM-centric storage management, where storage operations are offloaded to the storage array.

A datastore is a container used to hold files and objects. Datastores are logical containers, like file systems, that provide a standardized architecture for VM file storage while hiding the specifics of the physical storage device. A VM can be kept as a collection of files in a separate directory or as a collection of items in a datastore.

The following figure illustrates the datastores:

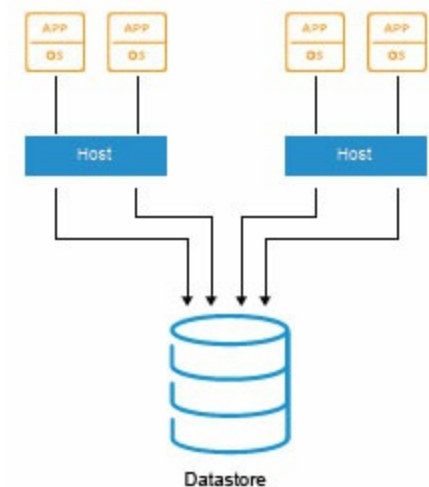


Figure 6.3: About datastores

(Source: VMware)

Methods to access datastore

vSphere datastores can access data as either blocks or files. Block-backed storage organizes data as byte sequences and is used with local storage, SANs (via iSCSI or Fibre Channel), and supports VMFS, vSAN, and vSphere

Virtual Volumes datastores. File-backed storage, on the other hand, organizes data hierarchically in files and folders, typically used on NAS and supported by NFS and vSphere Virtual Volumes datastores.

Datastore contents

Datastores store data as either files or objects, depending on their type.

- **File-based datastores** (VMFS and NFS) store VMs as a set of files within individual directories. These files include the VM configuration file, virtual disk files, swap files, and more.
- **Object-based datastores** (vSAN and vSphere Virtual Volumes) store VMs as data containers called objects. Each VM is represented by a configuration object, virtual disk objects, swap space objects, and others. Objects consist of data, metadata, and a unique identifier.

The following figure illustrates the datastores with object and files:

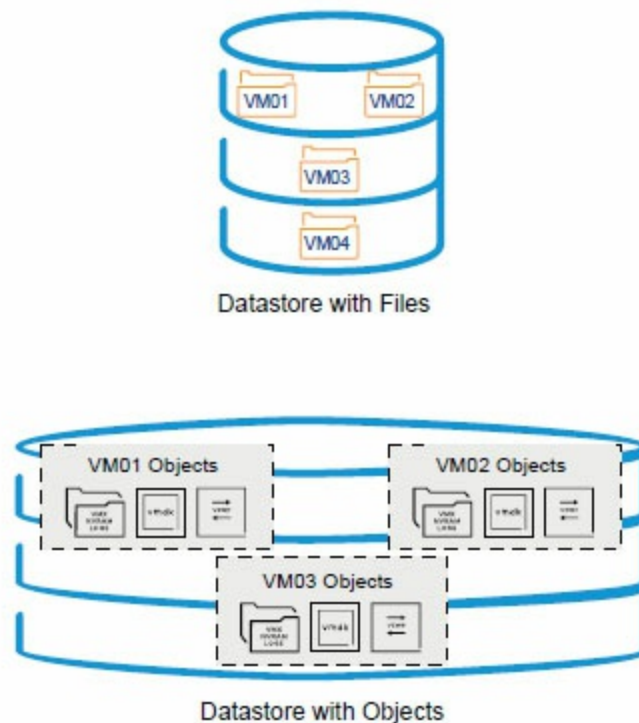


Figure 6.4: Datastores with objects

(Source: VMware)

vSphere VMFS overview

VMFS is a high-performance, clustered file system supported by ESXi hosts, available in versions VMFS5 and VMFS6. It enables multiple ESXi hosts to access shared storage concurrently, offering a variety of features tailored for virtualization.

The key features of VMFS are as follows:

- **Shared access:** Allows many ESXi hosts to read and write to the same storage drive simultaneously.
- **Dynamic scalability:** Datastores can be increased dynamically, even while VMs are running.
- **Virtualization services:** Centralized VM storage enables vSphere vMotion, HA, and clustering capabilities.
- **Efficient storage:** Sub-block addressing optimizes small file storage and supports virtual discs up to 62 TB.
- **On-disk locking:** Prevents several hosts from powering up the same virtual machine at the same time. Locking mechanisms are automatically released when a host fails, allowing VMs to restart on other hosts.

VMFS5 and **VMFS6** provide concurrent access, dynamic expansion, and on-disk locking. However, VMFS6 provides more capabilities, described as follows:

- Support for 4K native storage devices
- Automatic space reclamation
- Scalability to 128 hosts per datastore

Deployment options

VMFS can be implemented on direct-attached, Fibre Channel, and iSCSI storage.

The operational details are as follows:

- A VMFS datastore stores VM data as files on one or more LUNs.
- Virtual discs on VMFS appear as mounted SCSI devices to VMs, hiding the physical storage layer.
- The file system semantics remain native to the VM's guest operating system, ensuring data integrity and correct application behavior.

The following figure illustrates the VMFS datastore:

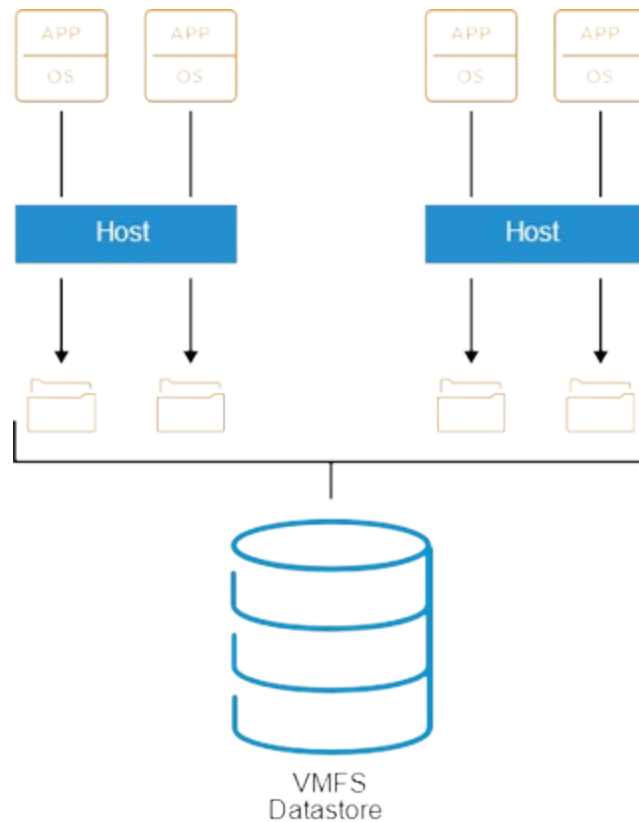


Figure 6.5: VMFS datastore

(Source: VMware)

Network file system overview

NFS is a file-sharing protocol that enables ESXi hosts to communicate with NAS devices over TCP/IP. NAS offers file access services, making it a unique and scalable storage solution for virtualised systems.

The key features of NFS in vSphere are as follows:

- ESXi supports NFS protocol versions 3 and 4.1 for connecting to NAS devices.
- NFS datastores, like VMFS datastores, can store virtual machine files, templates, and ISO images. They also enable vSphere vMotion migration for VMs stored on NFS volumes.
- **Locking mechanisms:**
 - **NFS 3** utilises VMware's unique locking technique, creating lock files on the NFS server instead of the conventional **Network Lock Manager (NLM)** protocol.

- **NFS 4.1** supports server-side file locking.

Compatibility considerations

NFS 3 and NFS 4.1 employ distinct locking methods, therefore administrator cannot mount the same datastore on multiple hosts running different NFS versions. Attempting to do so may result in data corruption or unpredictable behavior when accessing virtual discs.

NFS adds flexibility and simplicity to virtualized storage by bridging the gap between file-based storage and VMware's sophisticated capabilities, such as vSphere vMotion and centralized management.

The following figure illustrates the NFS datastore:

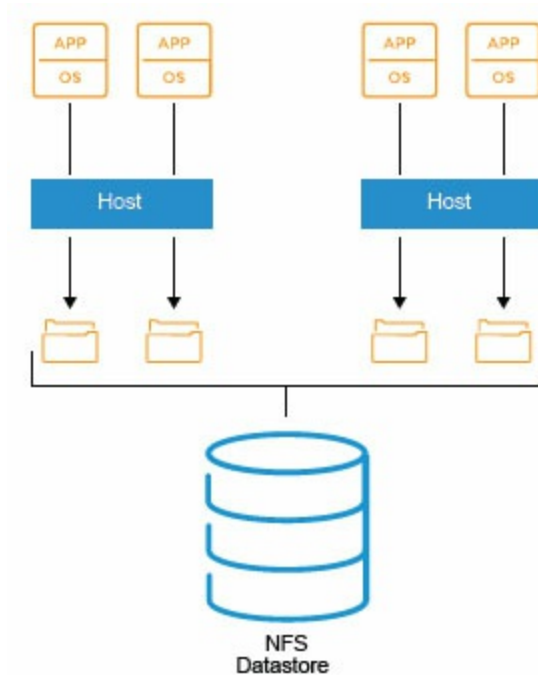


Figure 6.6: NFS datastore

(Source: VMware)

vSAN overview

vSAN is a software-defined storage solution fully integrated into VMware's hypervisor. Instead of relying on traditional external storage systems, vSAN leverages local storage devices, SSDs and HDDs, attached to ESXi hosts to create a shared datastore. This datastore is accessible to all hosts within the

vSAN cluster, enabling streamlined management and efficient resource utilization.

When vSAN is enabled on a cluster, it consolidates the storage devices from each host into a single, unified datastore. This eliminates the need for dedicated storage hardware, reducing complexity and cost.

vSAN offers two deployment models:

- **Hybrid architecture:** In this model, vSAN combines SSDs and HDDs to deliver a balanced storage solution. SSDs are utilized as a read cache and write buffer to enhance performance, while HDDs provide the necessary capacity for persistent data storage. This approach delivers a cost-effective solution suitable for environments with moderate performance requirements.
- **All-flash architecture:** All-flash deployment uses only SSDs for both caching and capacity. Flash devices in the cache tier handle write-intensive tasks, while the capacity tier uses SSDs optimized for read operations. This configuration ensures consistent performance with low latency and is ideal for applications demanding high throughput and reliability.

Whether it is hybrid or all-flash, vSAN provides organizations with a scalable, resilient, and high-performance storage solution, enabling efficient operations in virtualized environments.

The following figure illustrates the vSAN datastore:

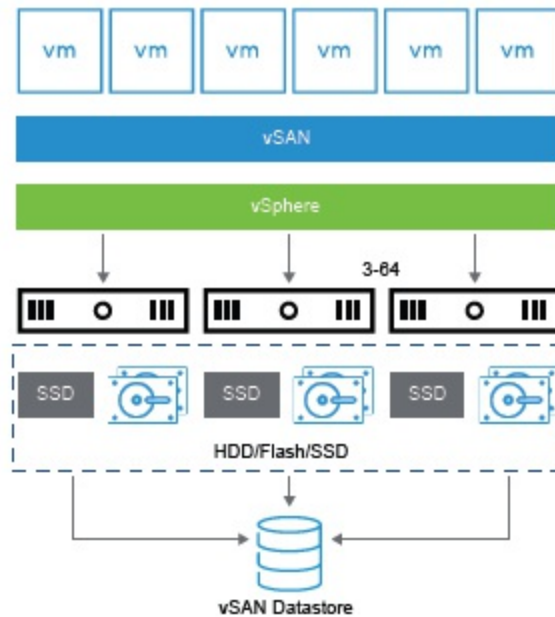


Figure 6.7: vSAN datastore

(Source: VMware)

vSphere Virtual Volumes overview

By converting actual storage hardware into virtual capacity pools, vSphere **Virtual Volumes (vVols)** revolutionise the way SAN and NAS storage are integrated in VMware systems. This method handles storage resources in a more dynamic and effective manner by doing away with the requirement for volume management or conventional LUNs.

Key features and capabilities

By integrating easily with present SAN and NAS infrastructures, vVols allow businesses to provide improved storage capabilities while optimizing current investments. vVols simplify management by doing away with the need for LUNs by natively representing VMDKs on external storage. At the virtual machine level, capabilities like snapshots and replications give administrators more precise control.

Storage containers that span entire arrays improve scalability and flexibility, while policy-driven automation guarantees that storage management complies with performance and availability requirements. vVols lower complexity and expenses by automating per-VM service levels and

eliminating the inefficiencies of conventional storage configurations. They are perfect for expanding settings because of their scalable architecture and direct VM-level operations, which provide outstanding performance for taxing workloads and analytics.

By adopting vVols, organizations can modernize their storage architecture, improve operational agility, and meet evolving business demands with greater efficiency and scalability. This method streamlines the administration of complex storage infrastructures while guaranteeing VMware environments are prepared for future expansion.

The following figure illustrates about vSphere virtual volumes:

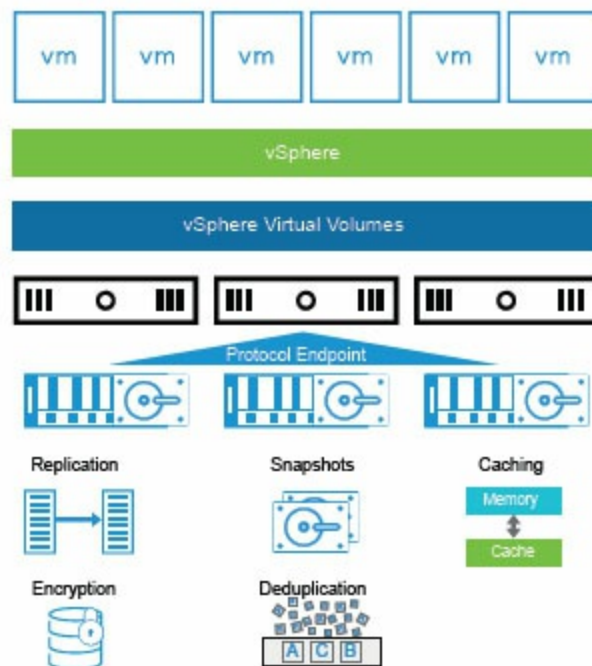


Figure 6.8: About vSphere virtual volumes

(Source: VMware)

Raw device mapping overview

Raw device mapping (RDM) is a unique VMware feature that provides VMs direct access to a physical LUN on a **storage area network (SAN)**. While it is not a traditional datastore, RDM enables advanced storage configurations by bridging the virtual and physical storage layers.

At its core, RDM uses a special mapping file (with the extension -

rdm.vmdk) stored on a VMFS datastore. This file serves as a proxy, pointing the VM to the physical LUN. Instead of storing VM data within a virtual disk file on the datastore, RDM allows the guest operating system to interact directly with the physical storage device.

Compatibility modes

RDMs operate in two distinct compatibility modes, depending on the requirements:

- **Virtual compatibility mode:**
 - In this mode, the RDM behaves like a standard virtual disk.
 - It supports features like snapshots, cloning, and vMotion.
 - The pointer file for this mode uses the extension **-rdm.vmdk**.
- **Physical compatibility mode:**
 - This mode allows the guest operating system to interact with the physical storage hardware directly.
 - It is ideal for SAN-aware applications that require access to underlying hardware features.
 - The pointer file for this mode uses the extension **-rdmp.vmdk**.

The following figure illustrates the raw device mapping:

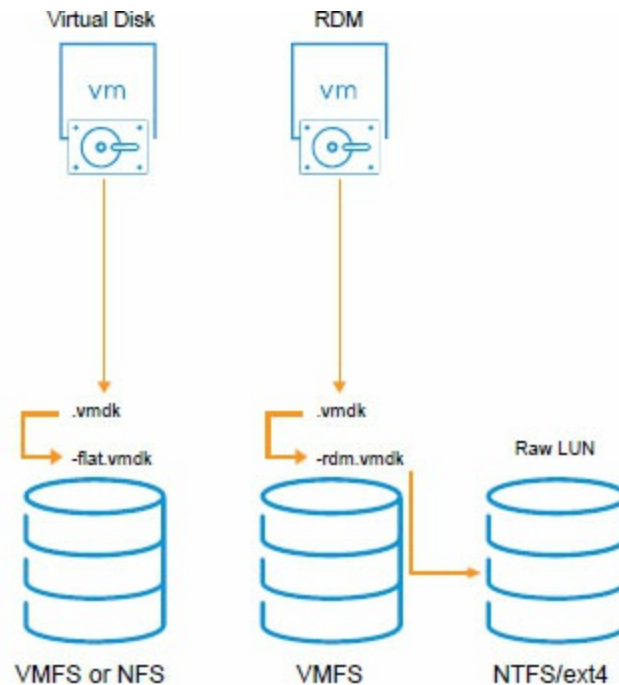


Figure 6.9: About raw device mapping

(Source: VMware)

Considerations for physical storage

One of the most important steps in creating a solid vSphere environment is organizing the storage infrastructure. Early collaboration with the storage administration team guarantees that the storage setup satisfies organizational objectives and application demands. LUN sizes, applications' I/O bandwidth needs, and the LUN's capacity to process I/O requests per second are important considerations. Disc cache parameters, zoning, masking setups, and multipathing settings are also crucial, regardless of whether the storage arrays are in active-active or active-passive mode. To prevent bottlenecks in situations that use NFS datastores, the export properties must also be properly specified.

Refer to the official vSphere Storage Documentation for additional advice on organizing the storage requirements at <https://docs.vmware.com/en/VMware-vSphere/index.html>. To learn more about advanced features and best practices, readers can also study the comprehensive resources on the vSphere Storage page at <https://core.vmware.com/>.

Fibre Channel components and addressing

FC is a high-speed protocol designed to access storage devices over a dedicated network. It is widely used in SAN environments, empowering efficient and reliable transport of SCSI traffic between virtual machines and Fibre Channel storage devices.

The key features and support in ESXi are as follows:

- Supports 32 Gbps Fibre Channel and **FC over Ethernet (FCoE)**.
- Utilizes Fibre Channel **host bus adapters (HBAs)** to connect hosts to Fibre Channel SANs.

Connectivity overview

To connect to a Fibre Channel SAN, ESXi hosts typically use HBAs or FCoE adapters. In a traditional setup:

- The host interfaces with a SAN fabric comprising Fibre Channel switches and storage arrays.
- Storage LUNs presented by the array are accessible to the host for creating datastores formatted with VMFS.

For environments leveraging vSphere Virtual Volumes, Fibre Channel allows direct access to storage containers, enabling the creation of Virtual Volumes datastores for granular VM storage management.

Whether using Fibre Channel switches or FCoE networks, Fibre Channel guarantees robust, high-performance storage connectivity, making it a vital component for enterprise-grade virtualized environments.

The following figure illustrates the VMFS/vVols datastore on Fibre Channel:

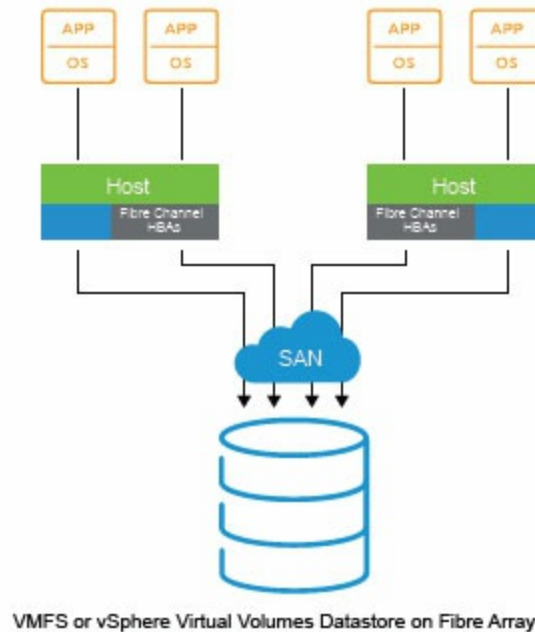


Figure 6.10: VMFS/vVols datastore Fibre Channel

(Source: VMware)

Components of Fibre Channel SANs

A Fibre Channel SAN is a high-performance, specialized storage network designed to connect multiple servers to a shared storage array. Its architecture ensures scalability, redundancy, and reliability for enterprise applications.

[Figure 6.11](#) highlights the following components:

- **Disk array:** The disc array, which houses several storage discs arranged into storage processors, is the central component of the SAN. These processors control the flow of data between the SAN fabric and the storage array.
- **Storage processors:** Storage processors, which facilitate effective data routing to the linked hosts via Fibre Channel switches, are the interface between the disc array and the SAN fabric.
- **Fibre Channel SAN switches:** These switches act as the hubs connecting storage arrays and hosts. The switches allow path redundancy by linking hosts to storage devices, guaranteeing that data is still available even if one or more paths fail.
- **Hosts (Servers):** To access storage resources, hosts that are outfitted with HBAs connect to the SAN fabric. Multiple applications that require

dedicated storage access and excellent performance can run on each host.

- **The fabric:** The network infrastructure that links switches, hosts, and storage arrays is made up of the SAN fabric. Fibre Channel protocol facilitates communication over the fabric. To guarantee fault tolerance, redundancy is frequently incorporated into the fabric using numerous interconnected switches.
- **Host bus adapters:** Installing HBAs on host servers allows them to connect to the SAN fabric. These adapters guarantee smooth communication between the fabric's storage devices and the server.

The following figure illustrates the Fibre Channel components:

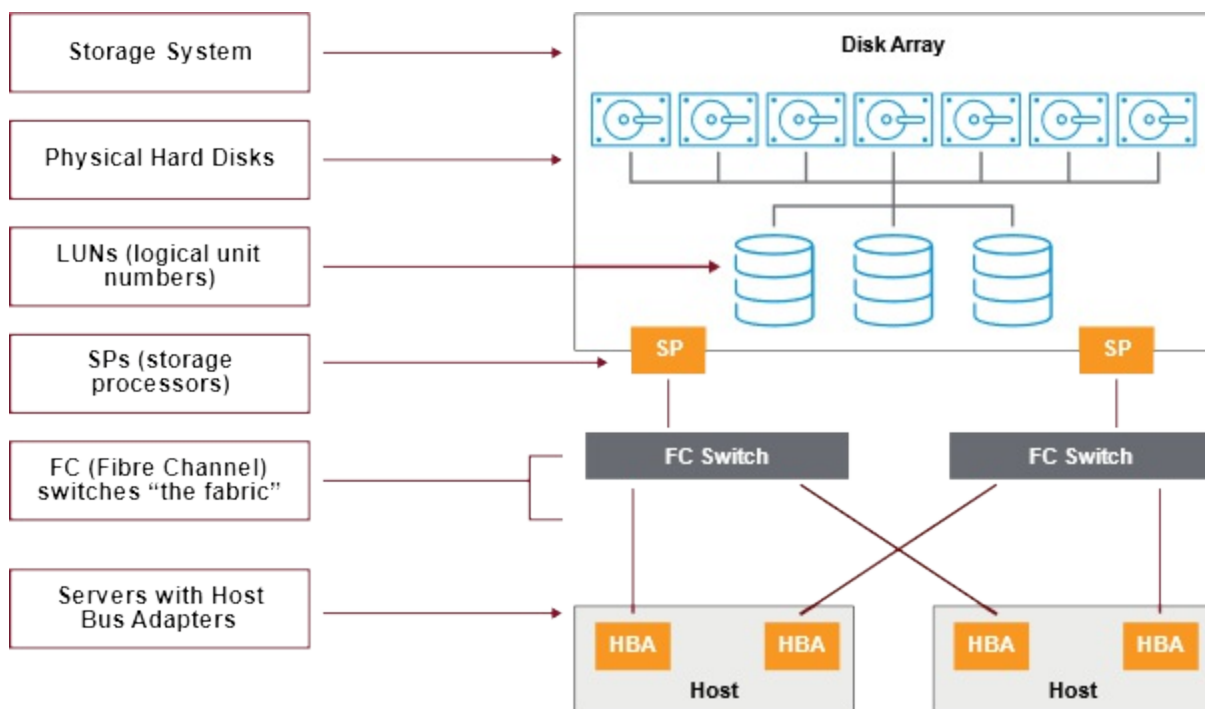


Figure 6.11: Fibre Channel components

(Source: VMware)

Access control and Fibre Channel addressing

Addressing and access control systems are essential to a Fibre Channel SAN because they guarantee secure storage resource access and effective communication. To keep the SAN functioning and organized, all of its components [hosts, storage devices, and fabric components like switches] are connected by ports that are uniquely recognized.

The key concepts are as follows:

- **World Wide Port Name (WWPN) and ports:** Every SAN node, such as switches, hosts, and storage array, has one or more ports that allow it to connect to the SAN fabric. These ports have a distinct WWPN, which makes them worldwide traceable. Applications may identify and access particular ports within the SAN fabric thanks to the WWPN.
- **Switches assign addresses:** Fibre Channel switches assign a port address dynamically after learning the WWPN of linked hosts or devices. For traffic routing and SAN communication integrity, this addressing method is essential.
- **Zoning for access control:** Within the SAN, devices are logically grouped using the zoning technique. Administrators can regulate which devices (such as hosts and storage arrays) are able to communicate with one another by establishing zones. Zones can be set up to separate:
 - **Environments for production and testing:** Make sure that testing does not conflict with workloads for production.
 - **Departmental workloads:** For increased security and resource allocation, establish distinct zones for different teams or departments.
- **LUN masking for storage separation:** By limiting which devices can view or access particular storage volumes, LUN masking improves access control even more. Only authorised devices will be able to access sensitive data thanks to this separation.
- **Integration of zoning and masking:** In vSphere environments, zoning, and LUN masking work together to provide a strong security layer that safeguards storage. SAN devices' host groups can also help with effective access control management.
- **Vendor-specific tools:** Storage arrays and SAN switches have hardware-specific zoning and masking features. To comprehend the tools available for configuration and management, it is imperative to refer to vendor-specific documentation.

For further details, refer to VMware's official documentation at <https://docs.vmware.com/en/VMware-vSphere/index.html>.

The following figure illustrates the Fibre Channel addressing and access

control:

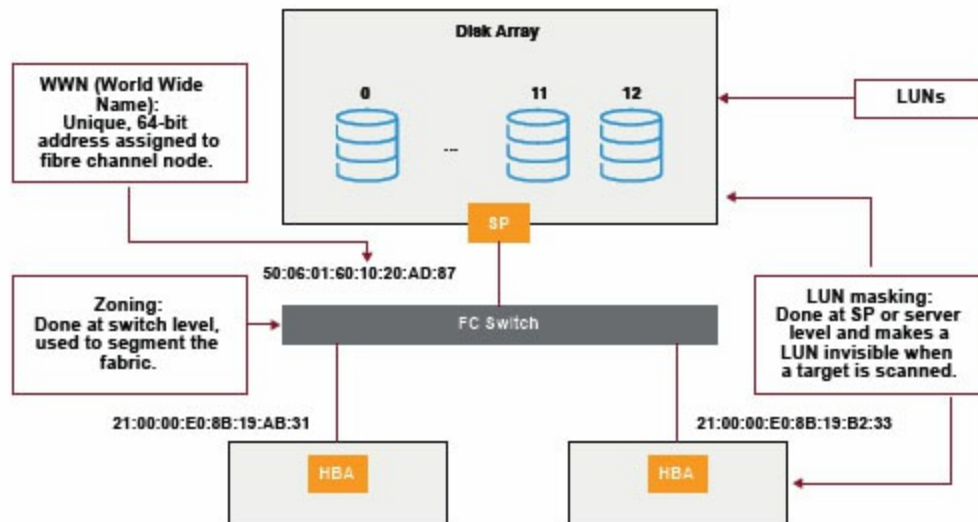


Figure 6.12: Fibre Channel addressing and access control

(Source: VMware)

Fibre Channel multipathing

Multipathing states to the configuration of multiple physical paths between a host and a storage LUN. It improves reliability, availability, and performance in a storage environment.

The key benefits of multipathing are as follows:

- **Fault tolerance:** Multipathing ensures uninterrupted access to SAN LUNs even in the event of hardware failure along a path (e.g., HBA, switch, or storage port).
- **Load balancing:** By distributing I/O traffic across multiple available paths, multipathing can avoid bottlenecks and enhance storage performance.

Fibre Channel path overview is provided as follows:

- **HBA port:** The host's interface connecting to the SAN fabric.
- **Switch fabric:** The intermediary SAN switches directing traffic.
- **Storage port:** The port on the storage array providing access to the LUN.

Path management in ESXi hosts includes the following:

- **Default behavior:** By default, ESXi hosts use a single active path to communicate with a specific LUN. If this path fails, the ESXi host automatically switches to an alternate path.
- **Path failover:** The process of detecting a failed path and seamlessly switching to another available path is called **path failover**. A failure could occur due to issues in any component of the path, such as:
 - HBA failure
 - Faulty cables
 - Switch port or fabric disruption
 - Storage processor malfunction

Disk array configurations

Understanding the disk array type is crucial for designing an efficient multipathing setup:

- **Active-active disk arrays:**
 - These arrays allow concurrent access to LUNs through multiple storage processors without lowering performance.
 - All paths remain active, confirming high availability and balanced I/O operations.
- **Active-passive disk arrays:**
 - In these arrays, only one storage processor actively services a given LUN, while the secondary processor acts as a standby.
 - I/O operations can only be engaged to the active processor. If it fails, the standby processor takes over, either automatically or through manual involvement.

Real-world use

Multipathing configurations are vital for mission-critical workloads that require consistent uptime, scenarios requiring performance optimization through load balancing, and environments where high reliability is essential, leveraging fault-tolerant storage designs to ensure continuous access to critical resources.

The following figure illustrates the multipathing with Fibre Channel:

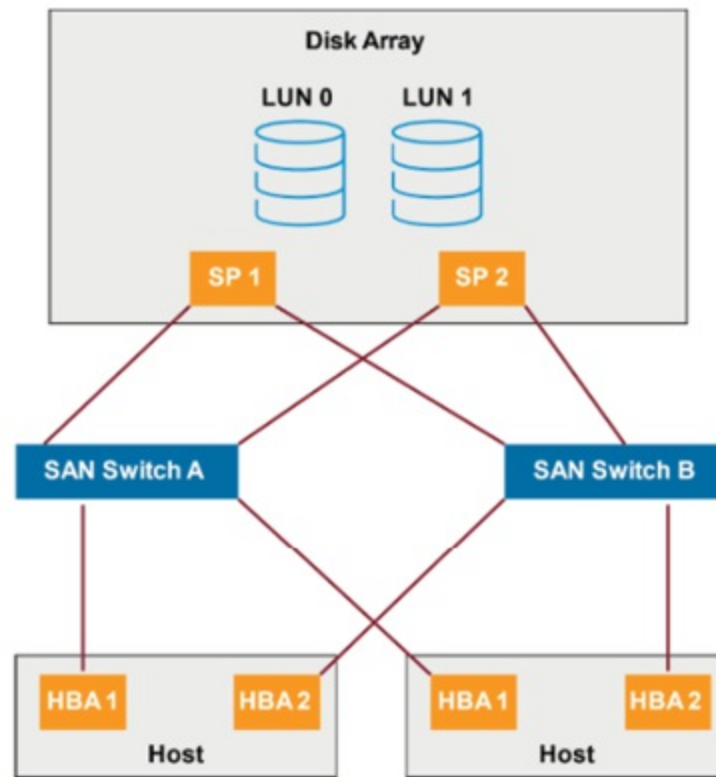


Figure 6.13: Multipathing with Fibre Channel

(Source: VMware)

Mastering iSCSI components and addressing

An iSCSI SAN embraces an iSCSI storage system that consists of one or more LUNs and storage processors. Communication between the host and the storage array occurs across a TCP/IP network, which utilizes the Ethernet infrastructure to provide SAN capability.

In an iSCSI SAN, the ESXi host contains an iSCSI initiator, which can be hardware-based (using an iSCSI HBA) or software-based (known as the iSCSI software initiator). The initiator sends SCSI commands over the IP network, while the target, located in the storage array, receives and processes these commands. A single iSCSI network can support numerous initiators and destinations, providing scalability and flexibility in complicated storage settings.

iSCSI is SAN-oriented for the resulting interactions:

- One or more targets are found by the initiator.
- The initiator receives LUNs from targets.
- To enable data interchange, the initiator gives the target SCSI commands.

The targets are found in the storage arrays that the host supports, while the initiator is found in the ESXi host. iSCSI arrays use a few techniques to guarantee safe access and prevent unauthorized hosts from getting to the targets, such as subnet limitations, IP address filtering, and authentication procedures.

By limiting access to specified targets and LUNs to authorized hosts, these safeguards improve security and preserve data integrity.

The following figure illustrates the iSCSI components:

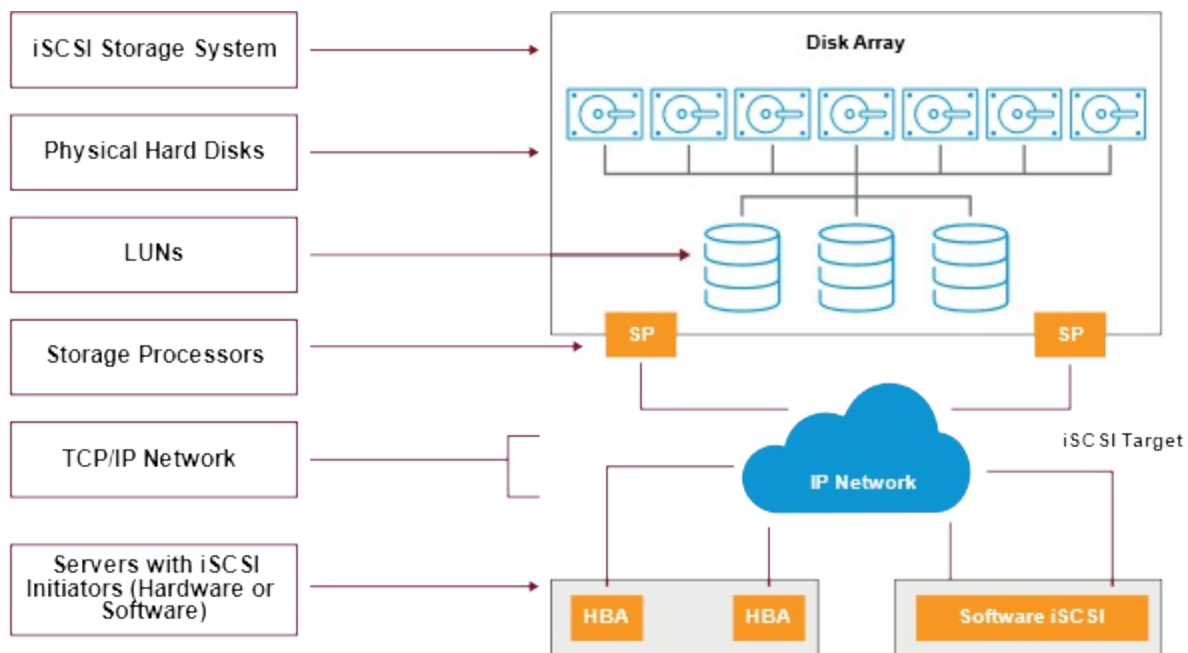


Figure 6.14: iSCSI components

(Source: VMware)

iSCSI addressing

In an iSCSI system, the principal accessible and discoverable entity is the iSCSI node, which can act as both an initiator and a target. Each iSCSI node

is given a name to ensure that storage resources may be managed efficiently, regardless of physical or network addresses. iSCSI nodes use one of the following naming conventions:

- **iSCSI Qualified Name (IQN)**
- **Extended Unique Identifier (EUI)**

The **IQN** format supports names up to 255 characters long and follows a systematic pattern. It starts with the prefix **iqn**, followed by a date code that indicates the year and month the organization registered its domain or subdomain. This is followed by the organizational naming authority string, which contains the domain or subdomain name in reverse. To ensure that each name is unique, a colon (:) can be added, followed by a custom string established by the organization.

For example, an IQN could be: **iqn.2025-01.com.example:storage.target1**.

On the other side, the **EUI** format has a more compact structure. It begins with the prefix **EUI** and is followed by a 16-character identification. This identity consists of 24 bits for the firm name, as assigned by IEEE, and 40 bits for a unique device ID, such as a serial number.

An example of an EUI name is: **eui.0123456789ABCDEF**.

Both the IQN and EUI naming formats guarantee exact and reliable identification of iSCSI nodes in a SAN environment, allowing for greater storage management flexibility and scalability.

The following figure illustrates the iSCSI addressing:

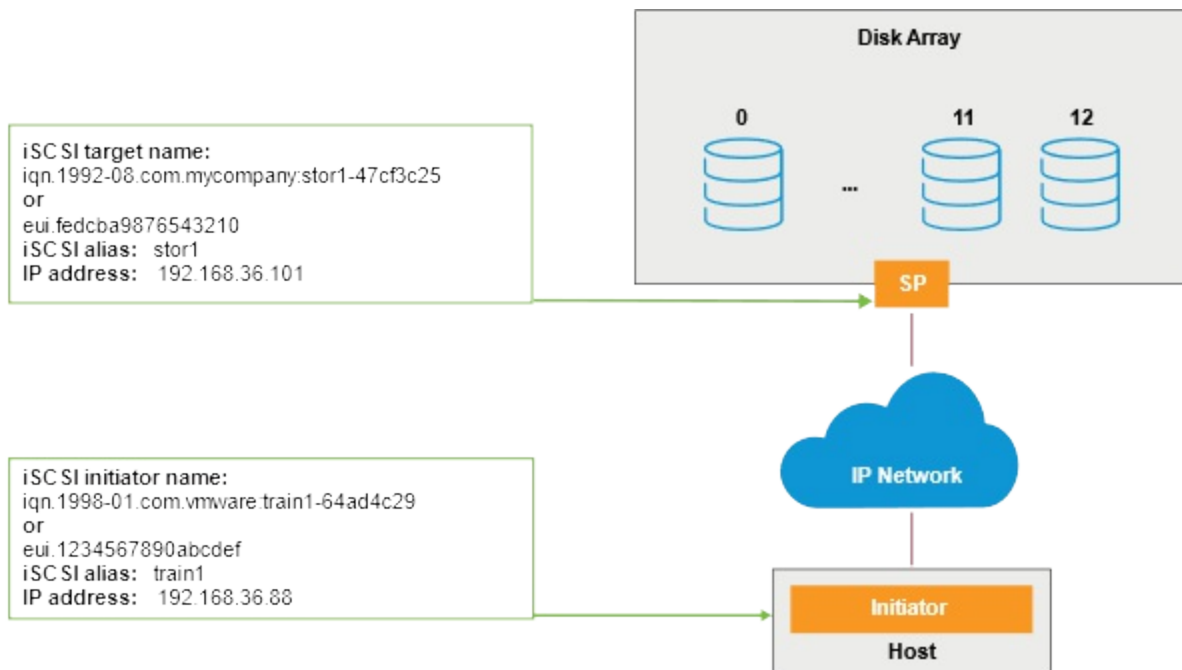


Figure 6.15: iSCSI addressing

(Source: VMware)

iSCSI adaptors

To allow an ESXi host to access iSCSI storage, iSCSI adapters must be configured first. These adapters function as initiators, sending SCSI requests and responses encapsulated in the iSCSI protocol between the host and the iSCSI target. There are two types of iSCSI initiators available:

- Software iSCSI initiators
- Hardware iSCSI initiators

A *software iSCSI initiator* is included in the VMware VMkernel, allowing the ESXi host to connect to iSCSI storage devices via standard network adapters. This software-based initiator manages all iSCSI processing while interacting with the network adapter, providing the benefits of iSCSI communication without the need for specialist hardware.

Hardware iSCSI initiators, on the other hand, are third-party adapters that use TCP/IP to connect to iSCSI storage devices. These are further classified as dependent and independent hardware iSCSI adapters:

- *Dependent hardware iSCSI adapters* combine iSCSI offload capabilities with the functionality of a typical network adapter. They need to be

connected to a VMkernel iSCSI port and require networking configurations for iSCSI communication.

- *Independent hardware iSCSI adapters* are standalone units that separately control all network and iSCSI operations. A VMkernel iSCSI port is not necessary for these adapters to function.

For detailed setup instructions, refer to vSphere Storage at: <https://docs.vmware.com/en/VMware-vSphere/index.html>.

The following figure illustrates the iSCSI adaptors:

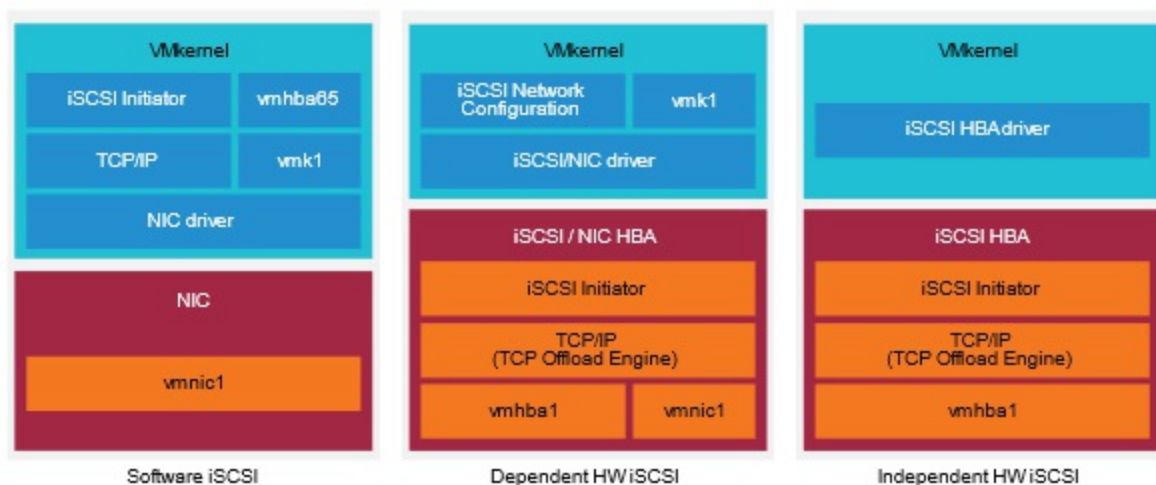


Figure 6.16: iSCSI adaptors

(Source: VMware)

Configuring the ESXi network for software iSCSI

Before turning on the iSCSI initiator, a VMkernel port must be created to configure ESXi to use software iSCSI. When using software iSCSI, proper networking configuration is essential for maximizing speed and guaranteeing security. The following are important suggestions:

- **Network segregation:**
 - **Physical segregation:** iSCSI traffic ought to ideally be managed on a distinct physical network that is isolated from other traffic, such as NAS or NFS.
 - **VLAN separation:** Use VLANs to logically segregate iSCSI traffic on a shared virtual switch if physical separation is not practical.

- **Setting up a network:**
 - **Single network adapter:** To manage iSCSI traffic when utilizing a single physical network adapter, set up a virtual switch's VMkernel port.
 - **Multiple network adapters:** To increase performance and redundancy in setups with two or more physical adapters, set up host-based multipathing.
- **VMkernel ports:**
 - These ports are essential for controlling iSCSI traffic and connections to hardware iSCSI adapters that depend on it.
 - Make sure the VMkernel ports assigned to iSCSI have the correct IP addressing and isolation.
- **Performance and security:**
 - To improve security and performance, separate iSCSI traffic from conventional network traffic.
 - In the event that physical network isolation is not possible, configure distinct VLANs on the virtual switch.

Figure 6.17 illustrates a standard ESXi network setup with iSCSI, showing the VMkernel ports corresponding to IP Storage 1 and IP Storage 2, together with the corresponding physical adapters (vmnic5 and vmnic6). Each port is set up according to iSCSI best practices on distinct subnets for isolation and redundancy.

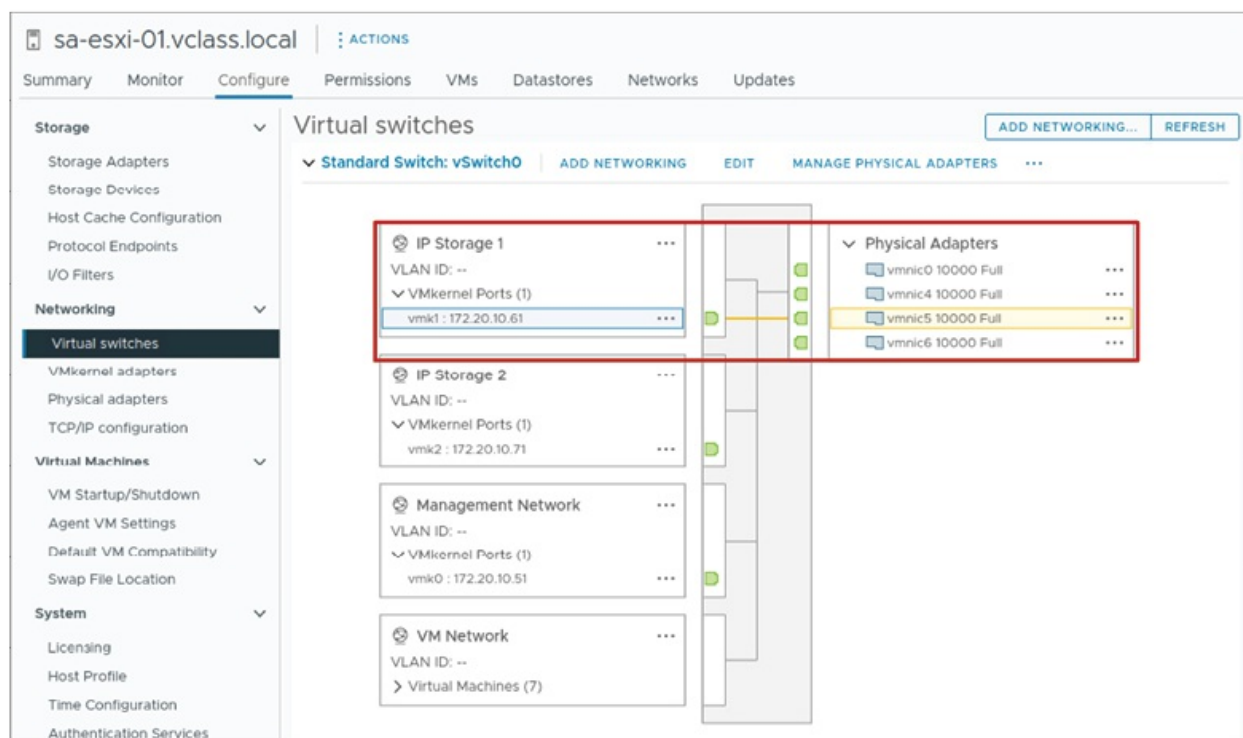


Figure 6.17: ESXi network config for software

(Source: VMware)

Activating the software iSCSI adapter

To use iSCSI storage with an ESXi host, the administrator first needs to activate the software iSCSI adapter. This process ensures that the adapter is functional and ready for storage configurations.

The steps to add and activate the software iSCSI adapter is as follows:

1. Select the ESXi host from the vSphere Client inventory.
2. Click the **Configure** tab.
3. Under the **Storage** section, select **Storage Adapters**.
4. Click **ADD SOFTWARE ADAPTER**.
5. From the dropdown menu, select **Add iSCSI adapter** (as shown in [Figure 6.18](#)).
6. The software iSCSI adapter (e.g., vmhba65) will appear in the list of storage adapters.

Some important points to note are as follows:

- Only one software iSCSI adapter can be activated per host.

- If the host boots from iSCSI using the software iSCSI adapter, the adapter is automatically activated during the first boot, and the required network configuration is created.
- If administrator deactivates the adapter, it will be reactivated every time the host is rebooted.

The following figure illustrates the activation of the software iSCSI adaptor:

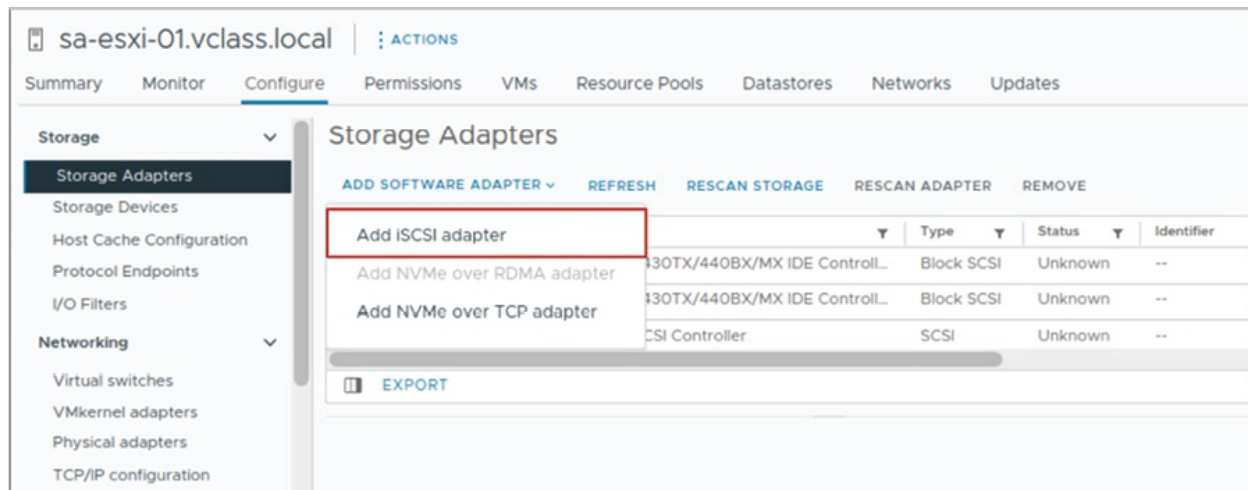


Figure 6.18: Activating the software iSCSI adaptor

(Source: VMware)

iSCSI target discovery

ESXi hosts can find and access network storage resources thanks to the iSCSI adaptor. It finds storage targets and their availability using discovery techniques.

Target-discovery techniques for iSCSI

The following techniques are supported by an ESXi host to find iSCSI targets:

- **Static discovery:**
 - Active discovery is not carried out by the initiator (ESXi host).
 - All the iSCSI targets that the initiator must contact have their IP addresses or domain names pre-configured.
 - Although this approach is simple, each target must be manually

configured.

- **Dynamic discovery (SendTargets discovery):**

- Using a SendTargets request, the initiator actively interacts with an iSCSI server.
- A list of all available targets, along with their IP addresses and IQNs (iSCSI Qualified Names), is returned by the iSCSI server.
- The vSphere Client automatically displays these targets as static targets.

The key considerations are as follows:

- The storage array's storage processor IP addresses must be provided by the ESXi host for it to communicate with the array and discover iSCSI LUNs.
- It is possible to manually eliminate static targets that are found dynamically. They might, however, resurface in the following occurrences:
 - A rescan.
 - A reset of the HBA.
 - A host restart.

While static discovery offers greater flexibility in some situations, dynamic discovery streamlines management by automatically updating the target list. The storage architecture and operational needs of the network will determine which approach is best.

The following figure illustrates the iSCSI targets discovery:

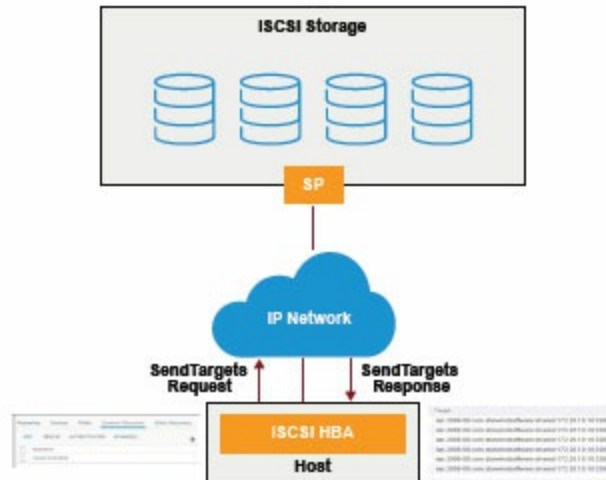


Figure 6.19: Discovering iSCSI targets

(Source: VMware)

iSCSI security, CHAP authentication

iSCSI initiators use **Challenge-Handshake Authentication Protocol (CHAP)** to enhance security by authenticating communication between the initiator (ESXi host) and iSCSI targets. By default, CHAP is not configured, but it can be implemented for both unidirectional and bidirectional authentication.

The types of CHAP authentication are as follows:

- **Unidirectional CHAP (One-way CHAP):**
 - The iSCSI target authenticates the initiator.
 - The initiator does not authenticate the target.
 - A CHAP secret (predefined private value) must be specified for the initiator to access the target.
- **Bidirectional CHAP (Mutual CHAP):**
 - Adds an extra layer of security by enabling mutual authentication between the initiator and the target.
 - Both the target and initiator must authenticate each other using separate CHAP secrets.

CHAP implementation includes the following:

- CHAP uses a *three-way handshake algorithm* to verify identities during

connection establishment. The process ensures that only trusted hosts and storage systems can interact on the iSCSI SAN.

- ESXi implements CHAP as defined in *RFC 1994*.
- **Adapter-level CHAP:**
 - The iSCSI adapter applies the same CHAP secret to all targets.
 - This is supported by both software iSCSI adapters and dependent hardware iSCSI adapters.
- **Per-Target CHAP:** For enhanced security, ESXi supports unique CHAP credentials for individual targets.

The steps to configure CHAP are as follows:

1. Verify CHAP activation on the iSCSI storage system.
2. Confirm the CHAP authentication method (Unidirectional or Bidirectional) supported by the storage system.
3. Activate CHAP on the ESXi host.
4. Ensure that the CHAP credentials (CHAP secret) match between the initiator and the storage system.

The best practices for iSCSI security are as follows:

- **Use CHAP:**
 - CHAP authentication is recommended to secure communication in iSCSI SAN implementations.
 - Consult the storage vendor for best practices and guidance specific to the storage architecture.
- **Segregate the iSCSI SAN:**
 - Use a dedicated **standard switch** for the iSCSI network.
 - Configure the iSCSI SAN on a separate **VLAN** to improve both performance and security.
- **Additional security measures:**
 - Deploy inline network devices for encryption to further protect data in transit.
 - Use access control lists and firewalls to restrict unauthorized access

to iSCSI targets.

By implementing CHAP and additional security measures, administrator can significantly enhance the protection of their iSCSI SAN environment. Ensure thorough validation of the configuration with the storage vendor to align with industry best practices.

The following figure illustrates the iSCSI security CHAP:

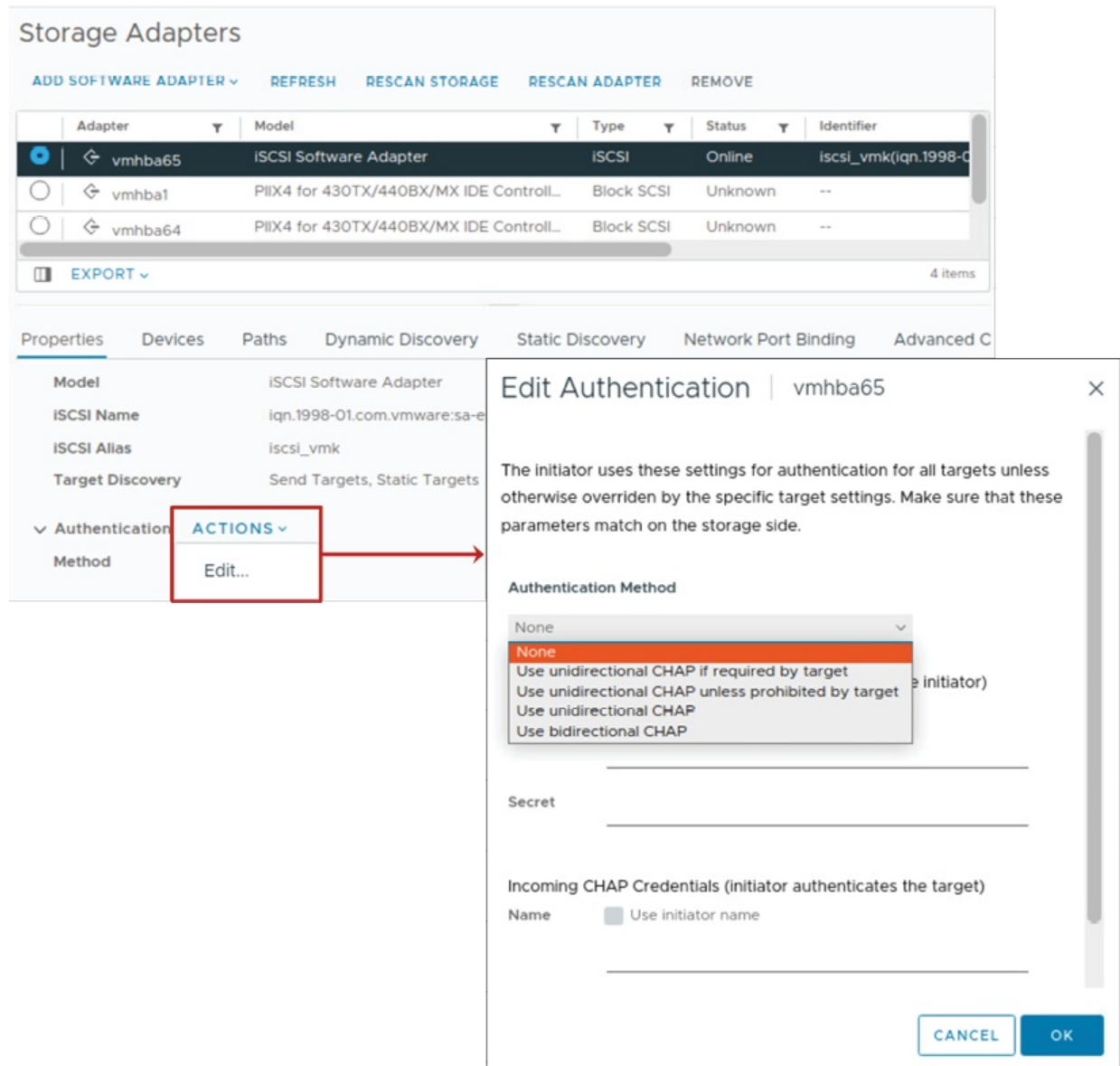


Figure 6.20: iSCSI Security CHAP

(Source: VMware)

Using iSCSI software for multipathing

By using numerous pathways between the ESXi host and iSCSI storage systems, multipathing is a crucial feature in iSCSI storage environments which ensures excellent availability and load balancing. Several NICs are employed for redundancy and performance optimization while multipathing with software iSCSI. The following are the important concepts of software iSCSI multipathing:

- **Using multiple NICs:**
 - A distinct VMkernel port is linked to each NIC.
 - The identical iSCSI initiator is connected to every VMkernel port.
 - This configuration offers numerous pathways to the iSCSI target and guarantees failover.
- **Initiator configuration:**
 - The iSCSI initiator IQN (iSCSI Qualified Name) is shared by all VMkernel ports.
 - Multiple pathways to the storage system are established by assigning a distinct IP address to each VMkernel port.
- **Path management:**
 - When deciding which NIC to use for iSCSI traffic, the ESXi host does not consult the VMkernel routing table.
 - vSphere multipathing modules manage traffic distribution and path selection, optimizing traffic flow and offering failover capabilities.
- **Preventing latency with routing:**
 - It is because of the possibility of higher latency, it is not recommended to route iSCSI communication via the VMkernel.
 - For best results, direct connection between the storage system and the ESXi host is recommended.
- **iSCSI multipathing advantages:**
 - **High availability:** In the event of a hardware or network failure, redundant paths avoid service interruptions.

- **Load balancing:** By dividing traffic over several routes, throughput and performance are increased.
- **Resilience:** Even during maintenance or unforeseen malfunctions, multipathing guarantees uninterrupted access to storage.

By carefully configuring iSCSI multipathing, administrator can achieve an efficient, reliable, and highly available storage network. Always follow vendor-specific recommendations to ensure compatibility and optimal performance.

The following figure illustrates the multipathing software iSCSI:

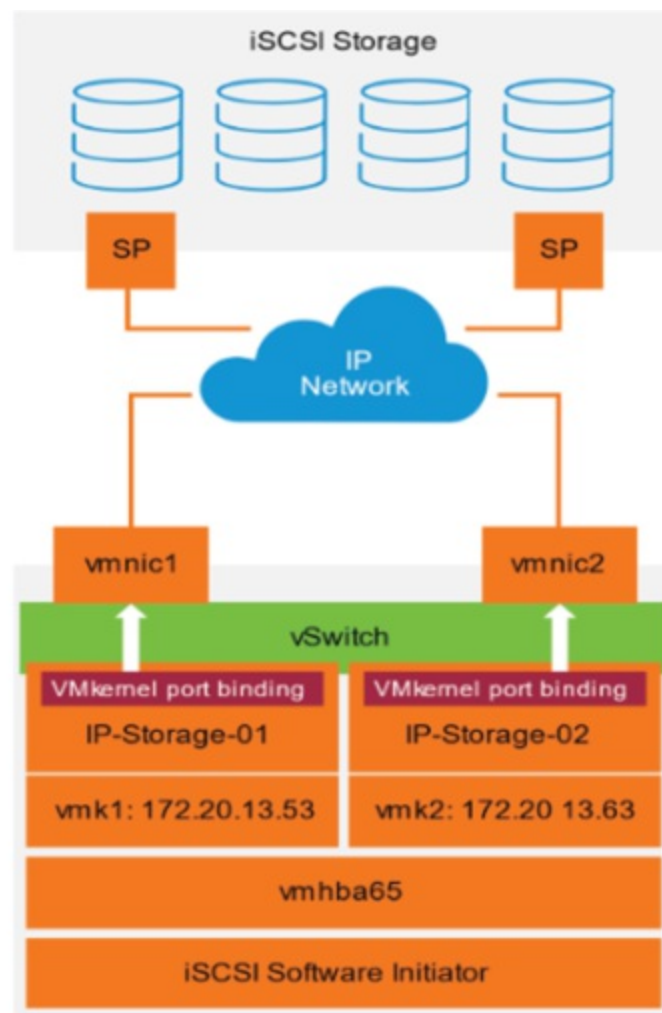


Figure 6.21: Multipathing software iSCSI

(Source: VMware)

Using dependent hardware iSCSI for multipathing

By using several network connections between ESXi hosts and storage systems, multipathing with dependent hardware iSCSI improves speed and redundancy. This method integrates vSphere networking and iSCSI settings with NICs and hardware iSCSI HBAs.

A normal network adapter (vmnic) and an iSCSI engine (vmhba), which show up as a single unit in the vSphere Client, are combined to form a dependent hardware iSCSI adapter. Despite being enabled by default, the iSCSI adapter needs to be set up to work properly. This entails attaching the adapter to a physical NIC and VMkernel port (vmk).

Use recommended practices, like employing separate physical connections for NICs and checking path redundancy on a regular basis, to guarantee optimal performance and dependability. Administrator may create a storage network for the ESXi hosts that is robust, efficient, and simple to administer by utilizing this configuration.

The following figure illustrates the multipathing dependent hardware iSCSI:

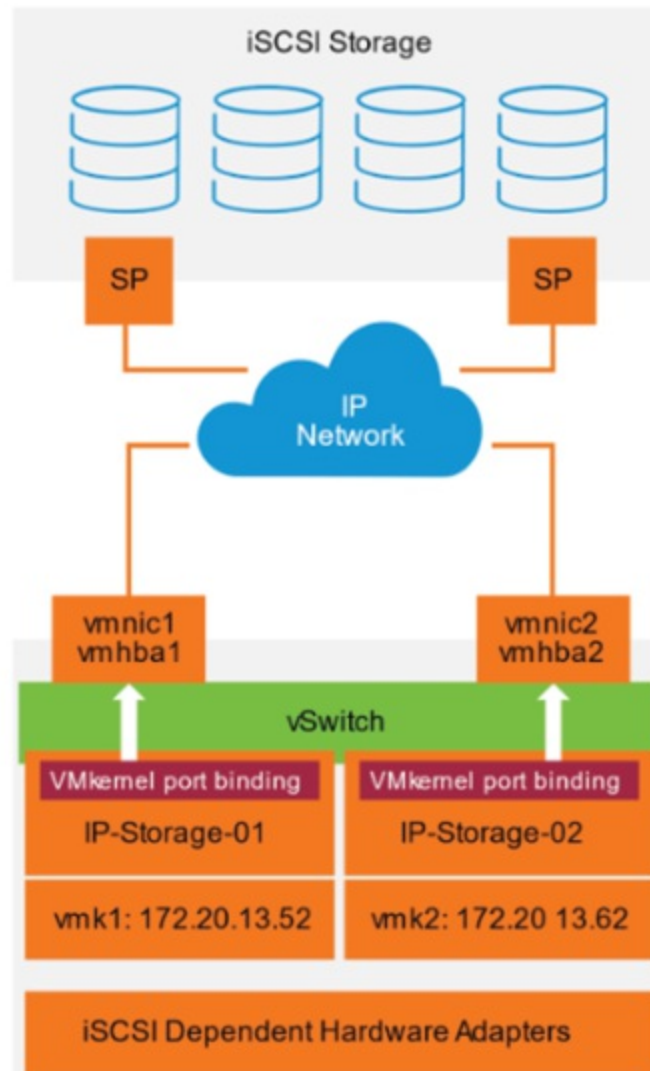


Figure 6.22: Multipathing dependent hardware iSCSI

(Source: VMware)

Using independent hardware iSCSI for multipathing

By using two or more hardware iSCSI adapters on the ESXi host, multipathing with independent hardware iSCSI improves redundancy and fault tolerance. Multiple pathways, via one or more switches, are available for accessing the storage system.

As an alternative, the configuration can use two storage processors and a single hardware iSCSI adapter. In this situation, the adapter can reach the storage system via a variety of routes, guaranteeing increased performance and constant availability.

When fault tolerance and high availability are crucial, this method offers a dependable and effective way to manage storage in virtualized settings.

The following figure illustrates the multipathing independent hardware iSCSI:

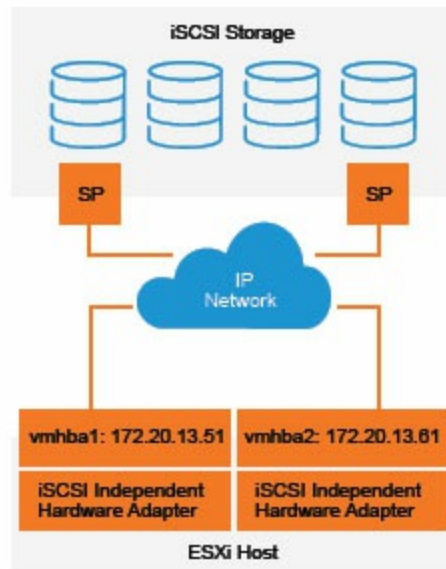


Figure 6.23: Multipathing independent hardware iSCSI

(Source: VMware)

Utilizing iSCSI initiator to bind VMkernel ports

Port binding lets each VMkernel port connected to a separate NIC act as an independent path for the iSCSI storage stack. This setup ensures well-organized utilization of multiple paths, enhancing redundancy and performance.

In software iSCSI and dependent hardware iSCSI configurations, multipathing plug-ins do not directly access physical NICs. Thus, each physical NIC must first be linked to a dedicated VMkernel port. The port-binding process then associates these VMkernel ports with the iSCSI initiator, enabling effective path management.

For dependent hardware iSCSI, ensure that the physical network card is properly installed. It should be visible in the **Configure** tab under the **Virtual Switches** view of the ESXi host.

The following figure illustrates the VMkernel ports binding with iSCSI

adaptor:

Storage Adapters								
ADD SOFTWARE ADAPTER ▾ REFRESH RESCAN STORAGE RESCAN ADAPTER REMOVE								
Adapter	Model	Type	Status	Identifier	Targets	Devices	Path	
vmhba65	ISCSI Software Adapter	iSCSI	Online	iscsi_vmk(iqn.1998-01.com.v...	1	4	8	
vmhba1	PIIX4 for 430TX/440BX/MX IDE Controll...	Block SCSI	Unknown	--	1	1	1	
vmhba64	PIIX4 for 430TX/440BX/MX IDE Controll...	Block SCSI	Unknown	--	0	0	0	
vmhba0	PVSCSI SCSI Controller	SCSI	Unknown	--	1	1	1	

Properties Devices Paths Dynamic Discovery Static Discovery Network Port Binding Advanced Options					
ADD REMOVE VIEW DETAILS					
Port Group	VMkernel Adapter	Port Group Policy	Path Status	Physical Network Adapter	
IP Storage 1 (vSwitch0)	vmk1	Compliant	Active	vmnic5 (10 Gbit/s, Full)	
IP Storage 2 (vSwitch0)	vmk2	Compliant	Active	vmnic6 (10 Gbit/s, Full)	

Figure 6.24: Binding VMkernel Ports with iSCSI Adaptor

(Source: VMware)

VMFS datastores creation and management

VMFS is a high-performance cluster file system used to store and handle big files, including virtual machine files, templates, and ISO images. It is designed to handle virtual disks and memory images from paused VMs, ensuring efficient storage operations.

A VMFS datastore has a maximum volume size of 64 TB and can be built on any SCSI-based storage device identified by the host, such as Fibre Channel, iSCSI, or local storage.

This flexibility allows organizations to leverage diverse storage technologies while maintaining consistent performance and reliability.

The following figure illustrates the VMFS datastore creation:

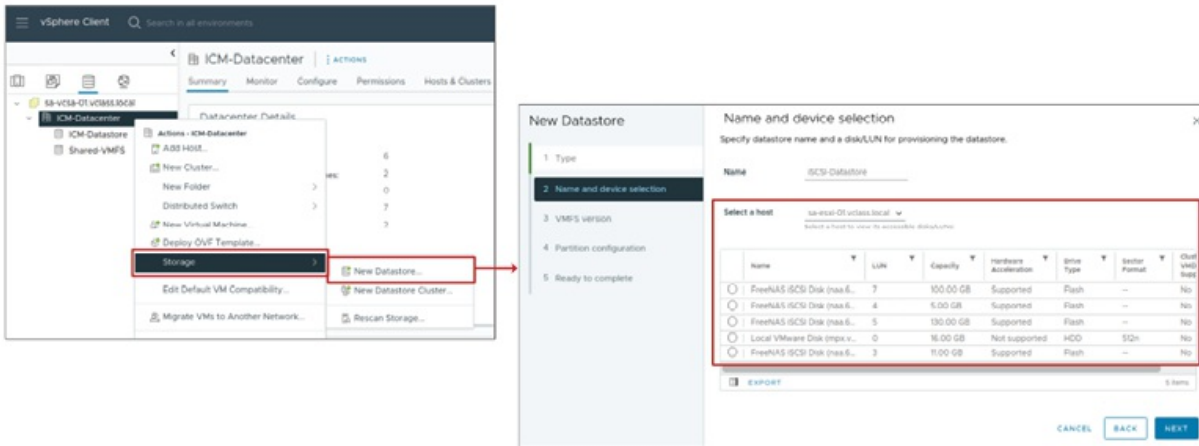


Figure 6.25: Creating VMFS datastore

(Source: VMware)

Viewing datastore contents

The *Datastore File Browser* in vSphere is an important tool for managing the contents of the datastores. Within the datastores pane, all configured datastores are displayed for easy navigation and organization across the managed ESXi hosts.

For example, [Figure 6.26](#) displays the ICM-Datastore that contains several folders and files. In the middle of the datastore pane, Linux-04 is selected and it displays their associated files, including:

- **vmx files:** Configuration files for virtual machines.
- **vmdk files:** Virtual disk files storing the data of the virtual machine.
- **nvram files:** Non-volatile memory files for storing VM BIOS information.
- **Log files:** Providing diagnostic and operational insights.

These folders and files allow administrators to proficiently manage virtual machines, templates, and related resources stored in the datastore. This feature also allows *uploading files*, *creating folders*, and *renaming or deleting files*, ensuring comprehensive datastore management.

The following figure illustrates the datastore content browsing:

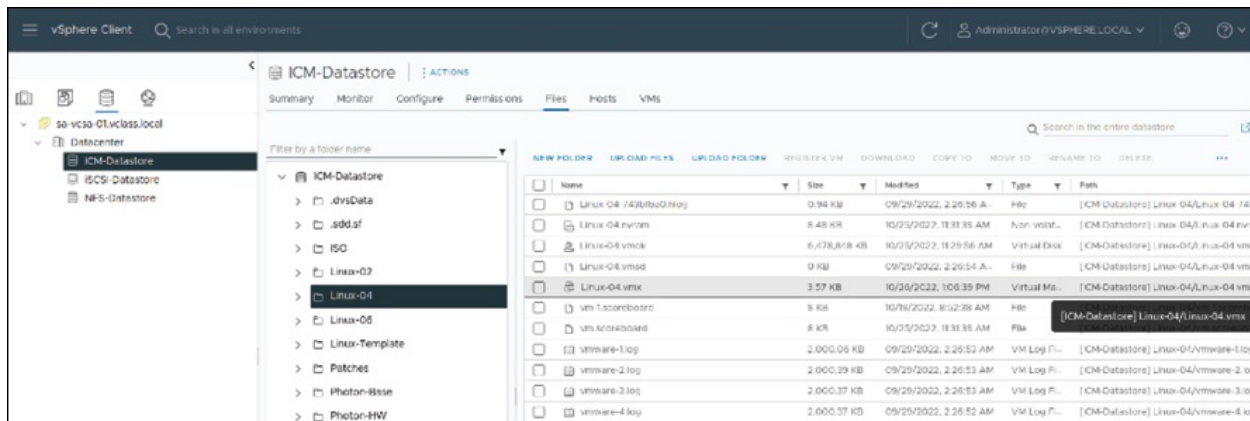


Figure 6.26: Browsing datastore content

(Source: VMware)

Expanding the VMFS datastores size

Expanding a VMFS datastore becomes necessary in situations such as insufficient disk space when creating or modifying virtual machines or when additional storage is required for existing virtual machines. Using the unique identifier (e.g., the NAA ID) to identify the datastore ensures accurate targeting of the volume, enabling effective and secure capacity management. Before modifying the storage allocation, it is highly recommended to perform below tasks:

- First, **perform a storage rescan** to confirm that all hosts can detect the most current storage configuration.
- Secondly, **record the unique identifier** of the volume that needs to be expanded (e.g., the NAA ID).

VMFS datastores can be expanded using the following approaches:

- **Adding an extent (LUN):** An extent is a partition on a LUN. By adding an extent to a VMFS datastore, the datastore can span multiple extents, up to a maximum of **32 extents**. This method allows the datastore to aggregate additional storage capacity from other devices.
- **Expanding the datastore within its existing extent:** This method involves increasing the size of the underlying extent first and then expanding the datastore to use the newly available space.

The following figure illustrates the VMFS datastore size increment:

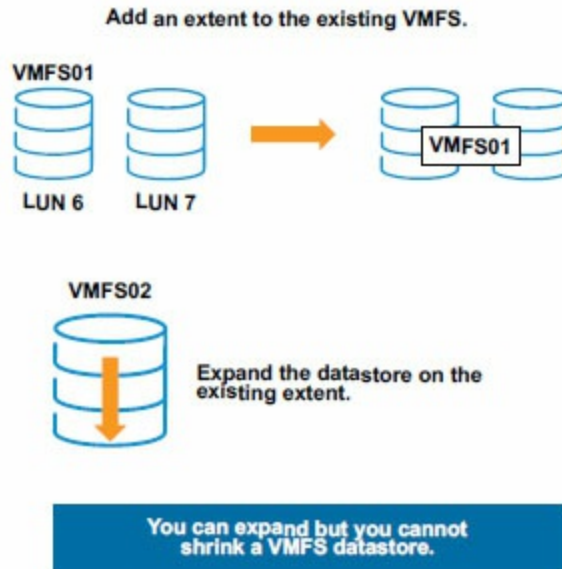


Figure 6.27: Increasing the size of the VMFS datastore

(Source: VMware)

Datastore maintenance mode

Before decommissioning a datastore, it must be placed into maintenance mode. This entails transferring all powered-on and powered-off virtual machines, as well as templates, to a different datastore. The datastore only enters maintenance mode once all VMs and templates have been transferred.

By selecting the option *Let me migrate storage for all virtual machines and continue entering maintenance mode after migration*, the migrate wizard makes VM migrations easier by allowing the administrator to migrate storage for all virtual machines while activating maintenance mode. If templates exist, they can be changed to VMs, moved, and then returned to templates.

The vSphere Storage DRS functionality includes datastore maintenance mode, however, it can also be used separately. VM migrations are entirely automated with vSphere Storage DRS.

The following figure illustrates the datastore maintenance mode:

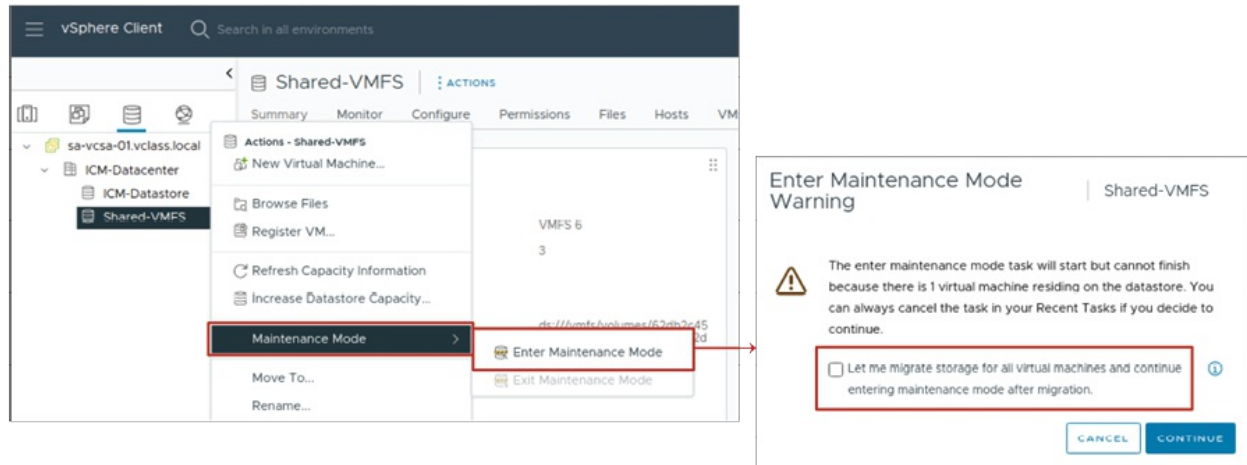


Figure 6.28: Datastore maintenance mode

(Source: VMware)

Deleting or unmounting a VMFS datastore

Unmounting a VMFS datastore preserves its files, but it is not accessible to the specified ESXi hosts, while a deleted datastore permanently erases all files and removes it from all connected hosts. During unmounting, avoid any operations that could generate I/O to the datastore.

Important points to consider before unmounting a datastore:

- No VMs reside on the datastore.
- It is not part of a datastore cluster.
- It is not managed by vSphere Storage DRS.
- Storage I/O Control is disabled.
- It is not used for vSphere HA heartbeat.

Always power off VMs using the datastore before unmounting or deleting to avoid errors. Deleting a datastore should be preceded by unmounting it, and backups are recommended to safeguard data.

The following figure illustrates the deleting unmounted VMFS datastore:

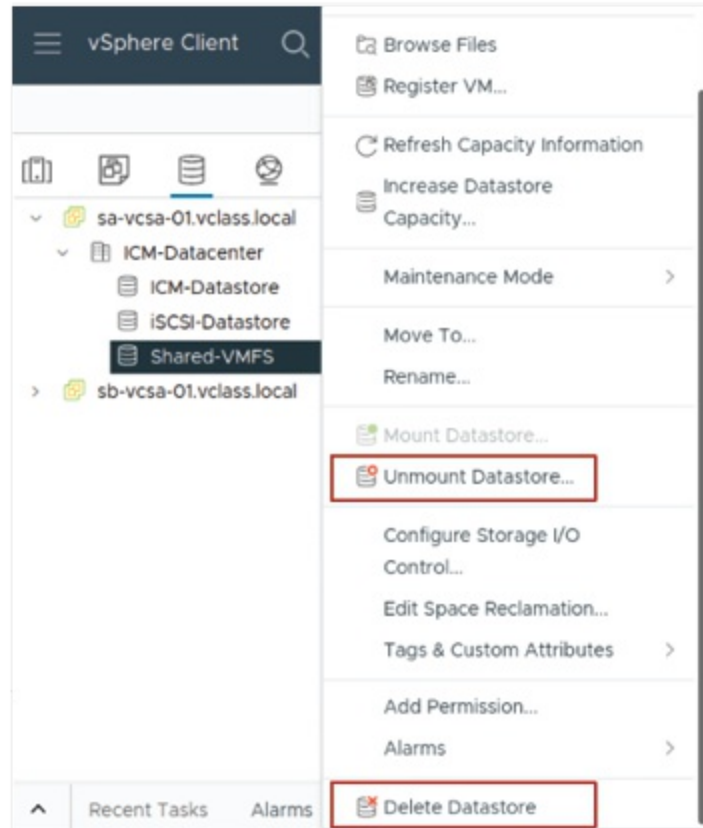


Figure 6.29: Deleting unmounted VMFS datastore

(Source: VMware)

Configuring storage load balancing and path selection policies

In VMware vSphere, multipathing allows multiple paths between an ESXi host and a storage device, which ensures high availability, scalability, and load balancing.

Path selection policies

Path selection policies determine how an ESXi host accesses storage devices. They provide key benefits, including *scalability*, *availability*, and *performance optimization*. Some of the policies are provided as follows:

- **Round Robin (RR):**
 - In addition to failover, the *Round Robin* policy provides *load balancing* by distributing I/O operations across all available paths.
 - The policy dynamically selects the optimal path by considering I/O

bandwidth and *latency* metrics.

- This helps in achieving better performance and scalability.
- *Latency-based optimization* is enabled by default on ESXi hosts.
- Before configuring this policy, consult the storage vendor to verify compatibility.

- **The Most Recently Used (MRU):**

- This policy directs the host to use the *first available path* discovered during boot.
- When the active path becomes unavailable, the host selects another functional path.
- It does *not revert* to the original path even after it becomes available.
- The *Most Recently Used* policy is typically the default for *active-passive storage arrays* and is mandatory for these types of devices.

- **Fixed:**

- The *Fixed* policy always uses a *preferred path* to access the disk when available.
- If the preferred path becomes unavailable, the host automatically switches to an alternative path.
- Once the preferred path is restored, the host reverts to it.
- This policy is ideal for *active-active storage arrays* and is set as the default policy for such devices.

In addition to VMware's native policies, third-party vendors can develop custom path selection algorithms tailored to their specific storage arrays. These algorithms provide enhanced load balancing and failover capabilities, allowing for optimized performance and seamless integration without requiring the vendors to disclose proprietary details about their storage systems to VMware.

By offering native and customizable multipathing solutions, VMware ensures efficient storage path management, improved reliability, and enhanced performance in diverse enterprise environments.

The following figure illustrates the multipathing algorithm:

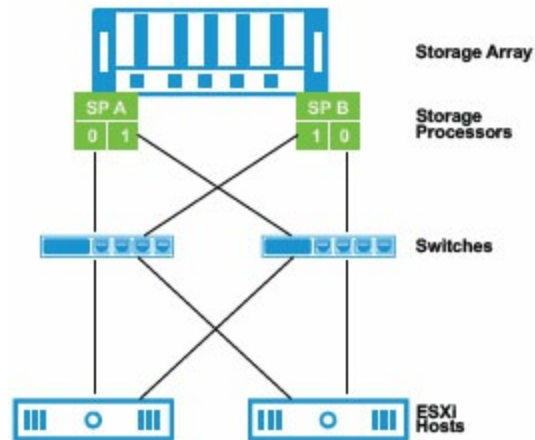


Figure 6.30: Multipathing algorithm

(Source: VMware)

Configuring path selection policies includes the following steps:

1. Navigate to the **Configure** tab in the vSphere Client.
2. Under **Connectivity and Multipathing**, select the datastore.
3. Click **Actions** and choose **Edit Multipathing Policies**.
4. From the **Path Selection Policy** dropdown, select one of the available options:
 - Fixed
 - Most Recently Used
 - Round Robin
5. Configure additional settings, such as the preferred path, and click **OK** to save changes.

Visualizing multipathing in vSphere

Figure 6.31 displays the configuration process for multipathing policies:

- The **Connectivity and Multipathing** tab display active paths to the datastore.
- The **Edit Multipathing Policies** window allows administrators to select a path selection policy and specify a preferred path.

By properly configuring these policies, organizations can ensure high availability, optimal performance, and efficient resource utilization for their

storage infrastructure.

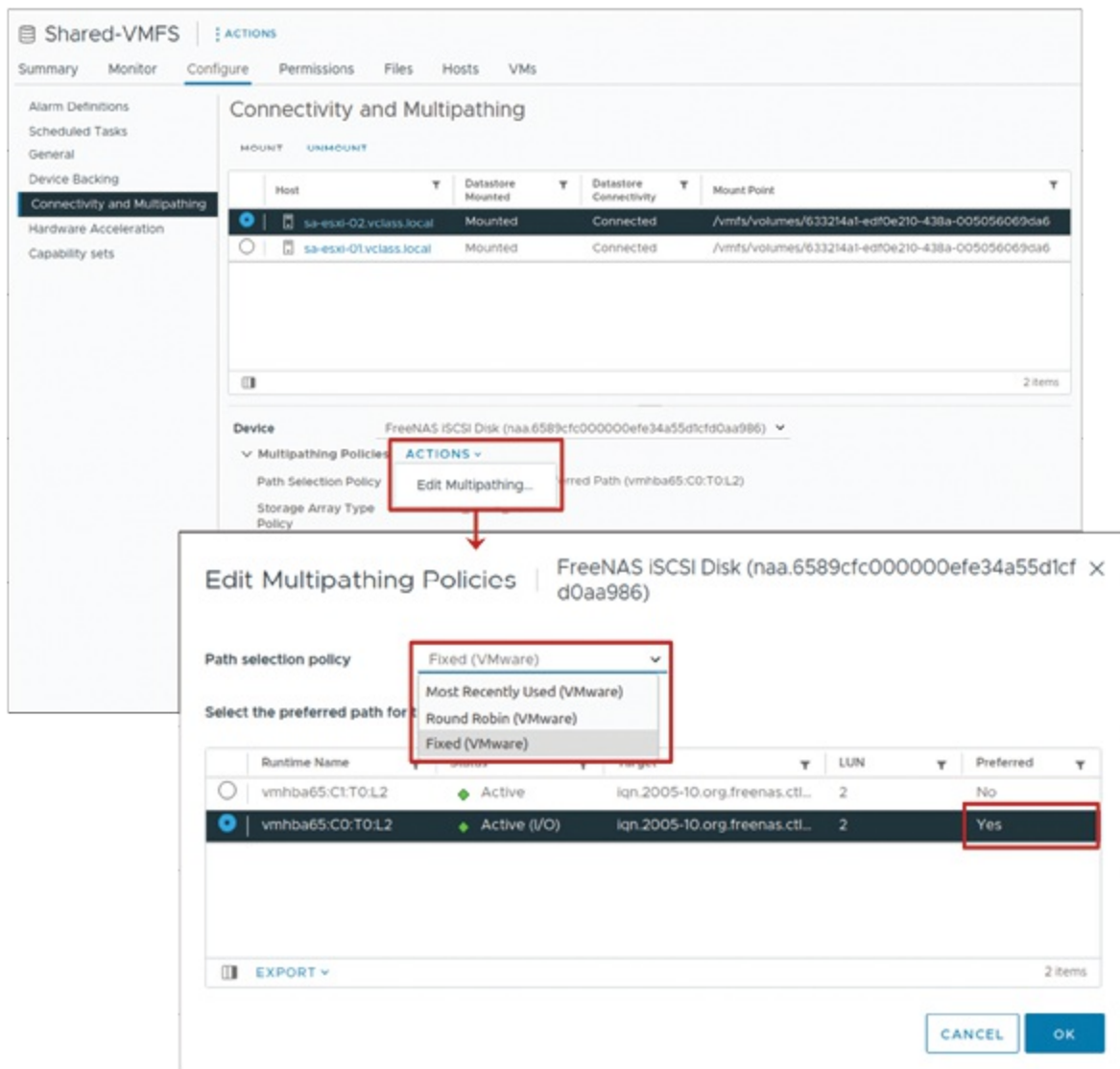


Figure 6.31: Configuring storage load balancing

(Source: VMware)

NFS datastores configuration and administration

NFS datastores provide shared storage in VMware environments by utilizing **network attached storage (NAS)** devices. The ESXi host uses a **VMkernel port** to communicate with the NFS server over a **TCP/IP network**, which simplifies the storage management for virtual environment.

The following figure illustrates the NFS components:

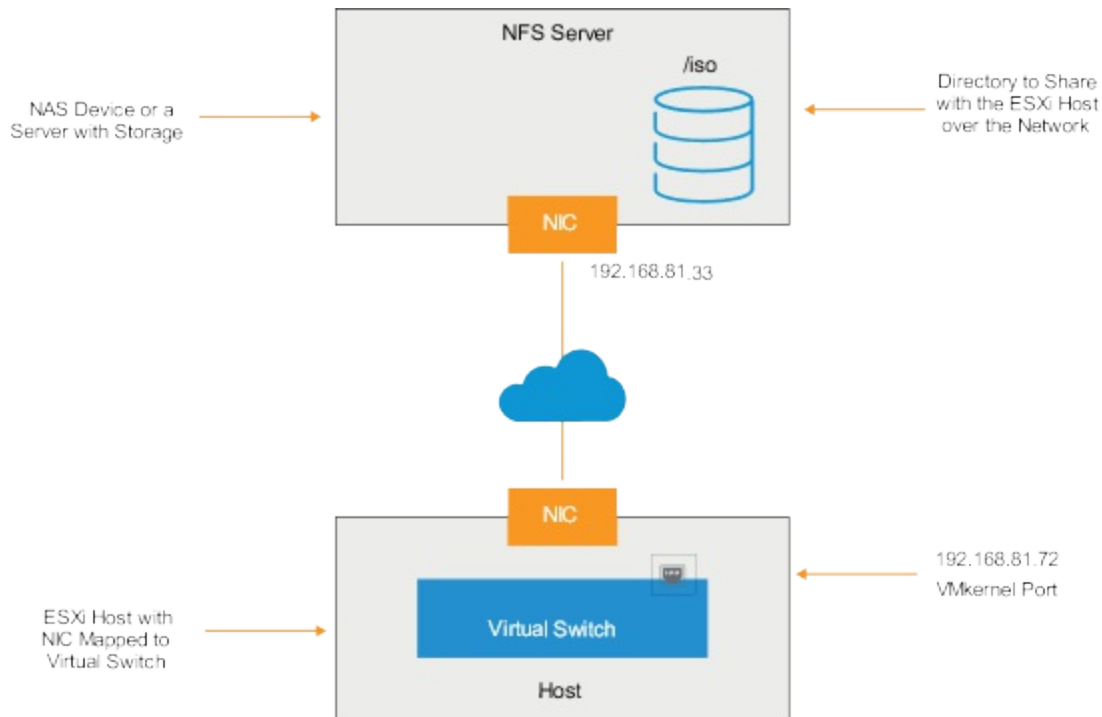


Figure 6.32: NFS components

(Source: VMware)

There are two types of NFS protocol, well known as NFSv3 and NFSv4.1 in VMware environments. The following are the distinct differences between the two protocols:

Feature	NFS 3	NFS 4.1
Multipathing	Managed by ESXi	Native multipathing and session trunking supported.
Authentication	Uses AUTH_SYS (root authentication).	Supports optional Kerberos authentication for enhanced security.
File Locking	Relies on VMware proprietary client-side file locking .	Uses server-side file locking , managed by the NFS server.
Error Tracking	Errors are tracked on the client side (ESXi host).	Errors are tracked on the server side (NFS server).
Performance	Generally lightweight and performs well in basic setups.	Used for advanced features, like authentication and locking.

Table 6.2: Comparison: NFS 3 vs. NFS 4.1

Due to compatibility apprehensions between the two NFS versions, access to datastores using both protocols from multiple hosts is not possible simultaneously. If a datastore is set up for NFS 4.1, all hosts that access it must be mounted with the share as NFS 4.1. Data corruption can occur if hosts access a datastore using the incorrect NFS version.

vSphere supports NFS 4.1, addresses several problems of NFS 3. Both NFS 3 and NFS 4.1 shares can be used, but administrators must be aware of critical limitations and choose wisely one over the other. Please visit <https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.storage.doc/GUID-8A929FE4-1207-4CC5-A086-7016D73C328F.html> for further information.

NFS datastores configuration

To configure an NFS datastore in VMware environment, administrator first **Create a VMkernel Port** on a virtual switch for each ESXi host that needs to access the NFS datastore. Assign a name to the VMkernel port for easy identification and management.

During the configuration wizard, the following details are needed:

- Specify the NFS version to be used (either **NFS 3** or **NFS 4.1**).
- Assign a **datastore name** for easy identification.
- Enter the **names or IP addresses** of the NFS server(s).
- Specify the **directory path** to be used, such as **/templates** or **/nfs_share**.
- Indicate whether the NFS file system should be mounted as **read-only**.
- Define which **ESXi hosts** will mount the datastore.
- Configure **authentication settings**, such as Kerberos or **AUTH_SYS** (depending on the NFS version).

Note: Separate the NFS network from other networks, such as the iSCSI network and virtual machine networks, to enhance performance and ensure security.

Setup ESXi host authentication and NFS Kerberos credentials

To enable **Kerberos authentication** for NFS datastores, confirm that all participating components, including ESXi hosts, NFS servers, and **Active**

Directory (AD), are properly configured and time must be synchronized using common NTP server. Precise time synchronization is highly important as Kerberos authentication might fail if there is a substantial time drift between the nodes.

Reduce the possibility of manual configuration errors by automating the NFS Kerberos setup with Host Profiles.

As we have already discussed, administrator should use either NFS 3 or NFS4.1 but not both. It is highly recommended to use NFS 3 or NFS 4.1 solely to avoid different authentication mechanisms, else this may lead to potential incompatibilities in **user identifier (UID)** and **group identifier (GID)** mappings.

The following figure illustrates the configuration of ESXi and NFS Kerberos credentials:

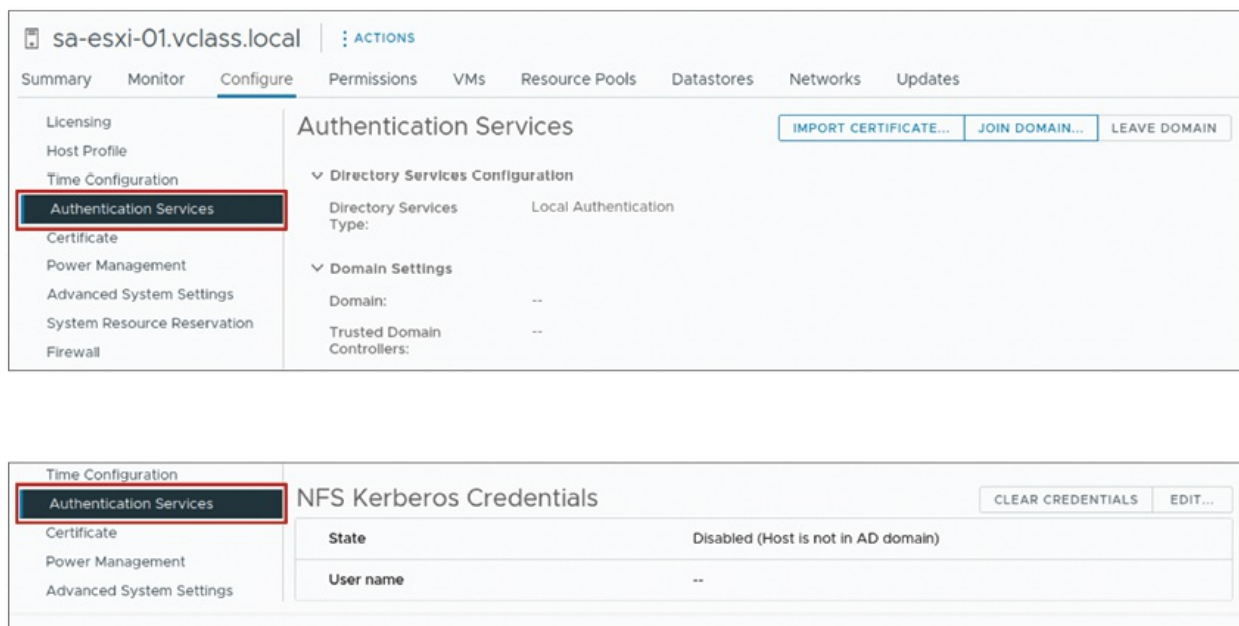


Figure 6.33: Configuring ESXi and NFS Kerberos credentials

(Source: VMware)

For detailed instructions on setting up Kerberos authentication for ESXi hosts, refer vSphere Storage at <https://docs.vmware.com/en/VMware-vSphere/index.html.vSphere>.

Setting up Kerberos authentication in NFS datastore

Administrators can improve security when configuring an NFS 4.1 datastore by turning on Kerberos authentication. One of the following security modes can enable this:

- Solely validated users and systems can access the NFS datastore thanks to **Kerberos5 (krb5)**, which solely offers authentication. It does not, however, provide protection against possible data manipulation while in transit.
- **Kerberos5i (krb5i)** provides data integrity, adding another layer of protection. Man-in-the-middle attacks, which aim to alter data while it is being transmitted, are identified, and countered by it.

Kerberos authentication can be enabled for the datastore after finishing the necessary initial setups, such as linking the ESXi host to the Active Directory domain and configuring Kerberos credentials.

For detailed instructions on setting up Kerberos authentication for ESXi hosts, refer to vSphere Storage at <https://docs.vmware.com/en/VMware-vSphere/index.html.vsphere>.

The following figure illustrates the NFS configuration to use Kerberos:

The screenshot displays the 'New Datastore' configuration wizard in vSphere. The left sidebar shows a progress list with six steps: 1 Type, 2 NFS version, 3 Name and configuration, 4 Kerberos authentication (currently selected), 5 Hosts accessibility, and 6 Ready to complete. The main panel is titled 'Kerberos authentication' and contains the following text: 'The NFS 4.1 client can secure NFS messages using Kerberos. You can enable the requisite security level below.' Below this is a yellow warning box with a triangle icon and the text: 'To use Kerberos authentication, each host that mounts this datastore has to be a part of an Active Directory domain and its NFS authentication credentials need to be set. This is done on the Authentication Services page on each host.' At the bottom of the main panel are three radio button options: 'Don't use Kerberos authentication', 'Use Kerberos for authentication only (krb5)' (which is selected), and 'Use Kerberos for authentication and data integrity (krb5i)'. At the bottom right of the wizard are three buttons: 'CANCEL', 'BACK', and 'NEXT'.

Figure 6.34: Configuring NFS to use Kerberos

(Source: VMware)

Unmounting an NFS datastore

When an NFS datastore is unmounted, the ESXi hosts from which it is unmounted are unable to access the files stored on it. If there are any VMs with disks on the NFS datastore, make sure those are powered off before proceeding with the unmounting operation.

The steps to unmount an NFS datastore are as follows:

1. **Locate the datastore:** Navigate to the **vSphere Client** and select the datastore that need to be unmounted (e.g., **NFS-Datastore** in the [Figure 6.35](#)).
2. **Access actions menu:** Right-click on the selected datastore and open the **Actions** menu.
3. **Select unmount datastore:** From the drop-down list, choose the option **Unmount Datastore...** to begin the unmounting process.

This process ensures safe unmounting while protecting data integrity:

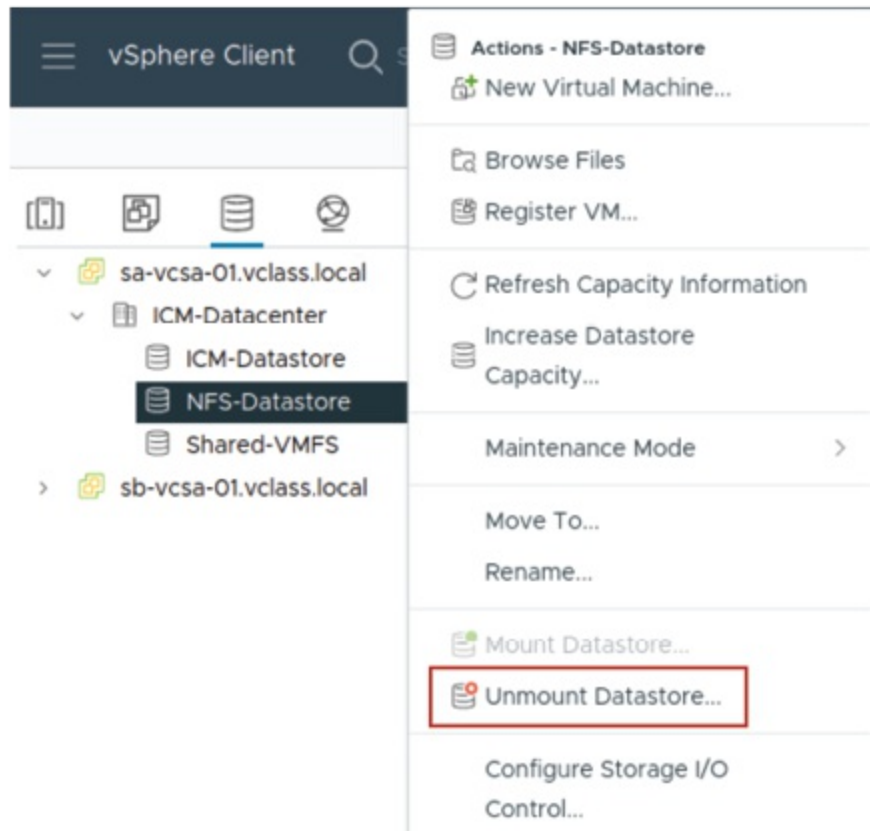


Figure 6.35: Unmounting NFS datastore

(Source: VMware)

Multipathing and NFS storage

To ensure a highly available NAS architecture and avoid single points of failure, configure NFS multipathing with the following best practices:

- **Avoid single points of failure:** Use redundant NICs and physical switches. For resilience, configure NIC teams across separate external switches.
- **NIC teaming and load balancing:** Attach NICs to different physical switches or the same switch (depending on switch capability). Use IP hash load-balancing policies to optimize bandwidth and performance.
- **VMkernel ports and NFS server configuration:** Use multiple VMkernel ports and configure the NFS server with multiple IP addresses (same or different subnets). Based on whether the switches support Cross-Stack EtherChannel:

- **With support:** Single VMkernel port and NIC teaming on separate switches.
- **Without support:** Multiple VMkernel ports on different subnets and virtual switches.
- **Bandwidth and performance optimization:** Use multiple datastores, each with separate connections, to maximize available bandwidth.

This streamlined approach improves both high availability and storage performance for ESXi hosts.

The following figure illustrates the multipathing and NFS storage:

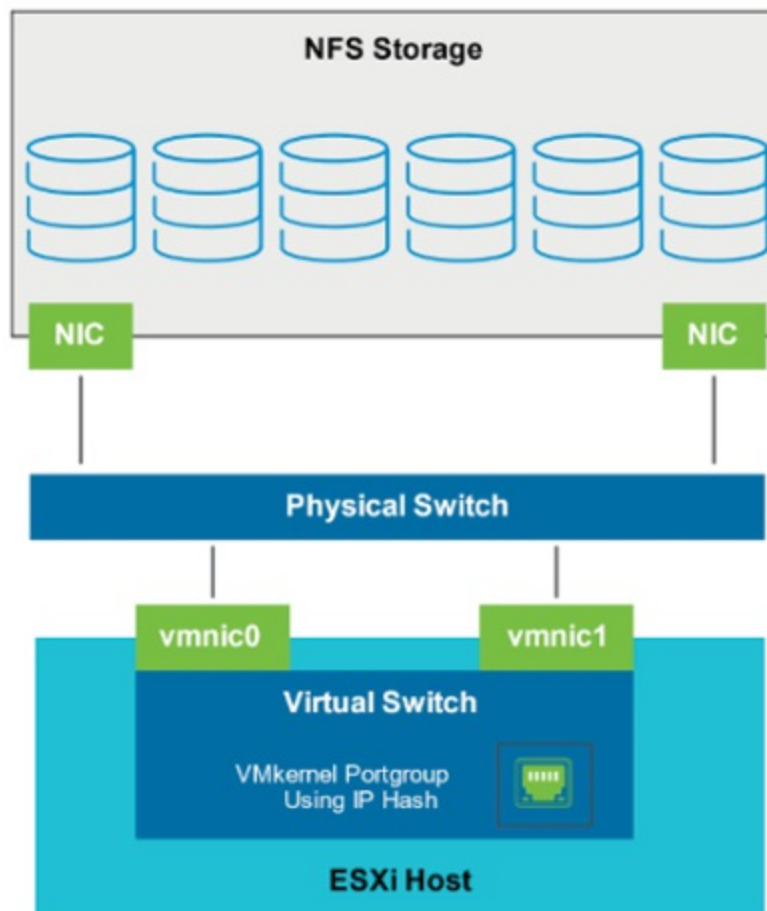


Figure 6.36: Multipathing and NFS storage

(Source: VMware)

Multipathing configuration for NFS 4.1

NFS 4.1 comes with some great built-in features, including native

multipathing and session trunking. As shown in [Figure 6.37](#), multiple server hostnames or IP addresses can be specified in the datastore configuration wizard. Session Trunking allows multiple IP addresses to access a single NFS volume, which means better performance and reliability.

Notes:

- Client ID trunking is not supported in NFS 4.1
- All specified server endpoints must have access to the same NFS share
- The multipathing configuration affects both performance and availability
- Proper network configuration is essential for optimal multipathing performance

The following figure illustrates the multipathing NFS configuration:

The screenshot shows the 'New Datastore' wizard in vSphere, specifically the 'Name and configuration' step. The left sidebar lists the steps: 1 Type, 2 NFS version, 3 Name and configuration (selected), 4 Kerberos authentication, 5 Hosts accessibility, and 6 Ready to complete. The main panel is titled 'Name and configuration' and includes a warning box: 'If you plan to configure an existing datastore on new hosts in the datacenter, it is recommended to use the "Mount to additional hosts" action from the datastore instead.' Below this, the 'NFS Share Details' section contains fields for 'Name' (NFS-ISO), 'Folder' (/iso), and 'Server' (E.g. nas, nas.it.com or 192.168.0.1) with an 'ADD' button. A table titled 'Servers to be added' lists two servers: nas01.vclass.local and nas05.vclass.local, which are highlighted with a red box. At the bottom, the 'Access Mode' section has a checkbox for 'Mount NFS as read-only'. Navigation buttons 'CANCEL', 'BACK', and 'NEXT' are at the bottom right.

Figure 6.37: Configuring multipathing NFS

(Source: VMware)

VMkernel binding in NFSv3

VMware has introduced NFSv3 VMkernel binding in vSphere 8 Update 1, allows administrator to direct NFS storage traffic to specific VMkernel port.

This feature enhances network security and provides improved control over the NFS storage connections.

As shown in [Figure 6.38](#), set up is quite straightforward in the New Datastore wizard:

- Look for the **VMkernel binding** option
- Check **Bind to vmknic** to enable the feature
- Select the preferred VMkernel port for NFS traffic

Let us look at enhancement in ESXCLI and vSphere API:

- The ESXCLI command now includes a **--connections (-c)** argument to specify the number of connections for an NFS datastore.
 - **ESXCLI:** `esxcli storage nfs add -H [host] -s [share] -v NFS -c [connections]`
 - **Example:** `esxcli storage nfs add -H 192.168.30.6 -s /mnt/nfs -v NFS -c 4`
- The vSphere API `createNasDatastore()` has been updated to include a connections property for defining the number of connections.
 - **vSphere API:** `/NFS/MaxConnectionsPerDatastore`

The following figure illustrates the NFSv3 VMK port binding:

The screenshot shows the 'New Datastore' wizard with the 'Name and configuration' step selected. The 'NFS Share Details' section includes fields for Name, Folder, and Server. The 'Vmknics binding' section is highlighted with a red box, showing the 'Bind to vmknic' checkbox checked. The 'Access Mode' section shows 'Mount NFS as read-only' unchecked. Navigation buttons 'CANCEL', 'BACK', and 'NEXT' are at the bottom right.

Figure 6.38: NFSv3 VMK port binding

(Source: VMware)

Conclusion

In this chapter, we looked at the whole landscape of storage management within VMware vSphere, exploring the critical technologies and configurations that support a strong virtualized environment. We started by learning about several storage systems, such as VMFS, NFS, and vSAN, as well as their specific features and applications.

The readers obtained a solid basis for installing and maintaining datastores, multipathing, and security after learning about Fibre Channel, iSCSI, and NFS storage solutions in depth. The introduction of sophisticated capabilities such as NFSv3 VMkernel binding and multipathing setups highlighted VMware's emphasis on performance, security, and scalability.

With this understanding, readers are now prepared to perform key storage jobs such as datastore expansion, multipathing to provide high availability, and secure storage traffic management. The ideas covered here lay the groundwork for developing efficient, high-performance, and robust virtual environments. Now that readers have a solid understanding of storage and datacentre, it is time to move on to VMs, another essential element of VMware systems.

As we move on to the next chapter, we will explore how to create, provision, and manage VMs within the vSphere environment in [Chapter 7, Virtual Machine Deployment](#). This chapter is going to be a comprehensive guide to mastering the intricacies of VM deployment and management, a vital skill for any vSphere professional.

Points to remember

- ESXi hosts support a wide range of storage technologies, which include **direct-attached storage (DAS)**, Fibre Channel, FCoE, iSCSI, and NAS.
- VMFS and NFS datastores store virtual machine files, but vSAN and vSphere Virtual Volumes handle VM objects, providing flexibility in storage architecture.

- Only one software iSCSI adapter can be activated per ESXi host.
- VMware's path selection policies, Fixed, **Most Recently Used (MRU)**, and Round Robin, optimize load balancing and failover, depending on the storage configuration.
- Leverage NFS 4.1's session trunking, Kerberos authentication, and advanced file locking for secure and scalable storage management.
- Use multipathing and NIC teaming to enhance storage performance and ensure high availability.
- Client ID Trunking is not supported in NFS 4.1.

Exercises

1. What are the primary differences between VMFS and NFS datastores in terms of file storage and access methods?
2. How does multipathing improve storage availability and performance in an ESXi environment?
3. What are the key features of NFS 4.1 compared to NFS 3?
4. How do VMware path selection policies like Round Robin, MRU, and Fixed differ in handling I/O traffic?
5. Describe the role of Kerberos in NFS authentication. What precautions must be taken for its configuration?
6. What is the maximum volume size supported by VMFS datastores, and what storage protocols can be used with them?
7. How does vSAN leverage local storage to create a unified datastore accessible to all hosts in the cluster?
8. When would you use VMkernel Binding with NFSv3, and how does it improve storage traffic management?
9. Discuss the steps involved in expanding a VMFS datastore and the potential scenarios where this might be necessary.

Lab exercises

1. **Lab, managing VMFS datastores:** Create, expand, and unmount

VMFS datastores.

a. Create VMFS datastores for the ESXi hosts:

- Select an ESXi host and go to Storage | Datastores.
- Use the New Datastore Wizard to create a VMFS datastore.
- Assign an appropriate name, select a storage device, and format it with VMFS.

b. Expand a VMFS datastore to Consume Unused Space on a LUN:

- **Log in to the vSphere Client:** Access the vSphere Client and navigate to the ESXi host where the VMFS datastore is mounted.
- **Identify the target datastore:**
 - Under Storage, select Datastores.
 - Locate and click on the VMFS datastore that needs to be expanded.
- **Check available LUN capacity:** In the Datastore Summary tab, view the datastore's current capacity and check if the underlying LUN has free, unallocated space.
- **Expand the datastore:**
 - Click Actions | Increase Datastore Capacity.
 - Select the existing LUN that has unused space available.
- **Allocate unused space to the datastore:**
 - Specify the additional space to allocate to the datastore. Administrators can use the maximum available space or specify a custom size.
 - Confirm the new capacity and proceed.
- **Verify expansion:**
 - After the operation completes, return to the Datastore Summary tab.
 - Verify that the datastore capacity reflects the expanded size.
- **Rescan the storage devices:** Go to Storage Adapters and rescan the devices to ensure all hosts in the cluster recognize the updated datastore size.

- **Test datastore functionality:** Create or expand a virtual disk on the datastore to confirm that the additional capacity is functional and accessible.

c. Unmount a VMFS datastore:

- First, migrate or delete the data from the datastore if available.
- Navigate to Menu | Storage in the vSphere Client and select the datastore to unmount.
- Right-click the datastore and choose Unmount Datastore.
- A dialog box appears listing all hosts connected to the datastore.
- Select the hosts from which the datastore should be unmounted and click OK.
- Check the status of the datastore. It should now appear as Unmounted in the Datastore Overview pane.

2. Lab, accessing iSCSI storage: Configure and verify access to an iSCSI datastore.

a. View an existing ESXi host iSCSI configuration:

- Log in to the vSphere Client and navigate to the ESXi host.
- View the iSCSI adapter under Configure | Storage Adapters to verify existing settings.

b. Add a VMkernel port for IP storage:

- Add a VMkernel port on a virtual switch dedicated to iSCSI traffic.
- Assign an appropriate IP address and subnet for the iSCSI network.

c. Add a second VMkernel port for IP storage:

- Create another VMkernel port to set up multipathing for iSCSI.
- Bind the VMkernel ports to the iSCSI adapter.

d. Add the iSCSI software adapter to an ESXi host:

- Activate the software iSCSI adapter via Storage Adapters | Add Adapter | Add iSCSI Adapter.
- Configure dynamic discovery by adding the IP address of the iSCSI target.

- e. **Discover LUNs on the iSCSI Target Server:**
 - Rescan the iSCSI adapter to detect available LUNs.
 - Verify the LUNs are visible under Storage Devices.
- 3. **Lab, accessing NFS storage:** Configure and verify access to NFS storage.
 - a. **Configure access to an NFS datastore:**
 - Add a new datastore and select NFS as the storage type.
 - Provide the NFS server details, including the export path and access mode.
 - b. **View NFS storage information:**
 - Verify the datastore details under datastores to confirm the capacity, protocol, and mount status.
- 4. **Lab, viewing a vSAN datastore configuration:** Understand vSAN datastore architecture and settings.
 - a. Navigate to the vSAN cluster and view the vSAN datastore under datastores.
 - b. Examine the storage policies, capacity distribution, and fault domain configuration.

Join our Discord space

Join our Discord workspace for latest updates, offers, tech happenings around the world, new releases, and sessions with the authors:

<https://discord.bpbonline.com>



CHAPTER 7

Virtual Machine Deployment

Introduction

Virtual machines (VMs) form the cornerstone of any virtualized infrastructure, providing the level of flexibility, scalability, and efficiency required by today's IT environments. Thus, mastering VM deployment extends beyond mere creation into the realms of understanding the different types of virtual hardware and optimizing configurations to fit into the workload. This chapter will take you through some of the critical processes for creating, cloning, and managing VMs and templates, enabling readers to edit already existing VMs and update existing templates to fit the changing needs with ease. Readers will have the necessary knowledge to build and maintain a dynamic virtual environment by the time they finish.

Note: VMware is now part of Broadcom and is known as 'VMware by Broadcom'. All references to VMware in this book reflect this change.

Structure

In this chapter, we will cover the following topics:

- Creating and provisioning virtual machines
- VMware Tools

- Virtual machines components
- Navigating vSphere client for VM management
- Virtual machines resources optimization
- Harnessing the efficiency of templates
- Cloning virtual machines
- Content libraries for VM resources
- Content library integration
- Managing VM template versions

Objectives

In this chapter, readers learn how to create, provision, and manage virtual machines in a vSphere environment; deeply understand the VMware tools, including their importance and the process of installation; determine the files that make up a virtual machine and compare hardware versions; assess the vSphere Client to get updated on the settings of VMs, get access to the VM Consoles, and dynamically configure hardware components such as virtual disks and hot-pluggable devices. It also allows them to learn how to develop templates, deploy VMs from templates, clone virtual machines, and customize guest operating systems; learn the different ways to use Content Libraries: creation, publishing, subscription, and management of the various versions of a VM Template. With all this information, the reader should be able to competently deploy, optimize, and manage virtualized environments.

Creating and provisioning virtual machines

There are several methods for creating and provisioning VMs in vSphere environment, each method is tailored to certain requirements and scenarios. The best method relies on the infrastructure's size, purpose, and requirements.

Provisioning methods

VMs can be provisioned using either the vSphere Client (vCenter) or the VMware Host Client (ESXi host):

- **New Virtual Machine wizard:** Both the vSphere Client and the VMware Host Client provide the option for creating single virtual machines.
- **Templates or clones:** The vSphere Client exclusively allows administrators to deploy VMs from existing templates or clones, which simplifies the process for environments that require several identical machines. This is not available in the VMware Host Client.
- **Open Virtual Machine Format (OVF) templates:** With any client, an administrator can deploy VMs, virtual appliances, or vApps stored in OVF.

Be aware of limitations compared to the vSphere Client when using the VMware Host Client to deploy OVF files. Certain limitations might impact the deployment of complex configurations. We highly recommend to consult VMware documentation for specifics on OVF/OVA support within the Host Client at <https://techdocs.broadcom.com/>.

New Virtual Machine wizard overview

Figure 7.1 displays how to use the New Virtual Machine wizard to create a VM in the vSphere Client. Right click on the host and select **New Virtual Machine** and follow the onscreen instructions as mentioned here:

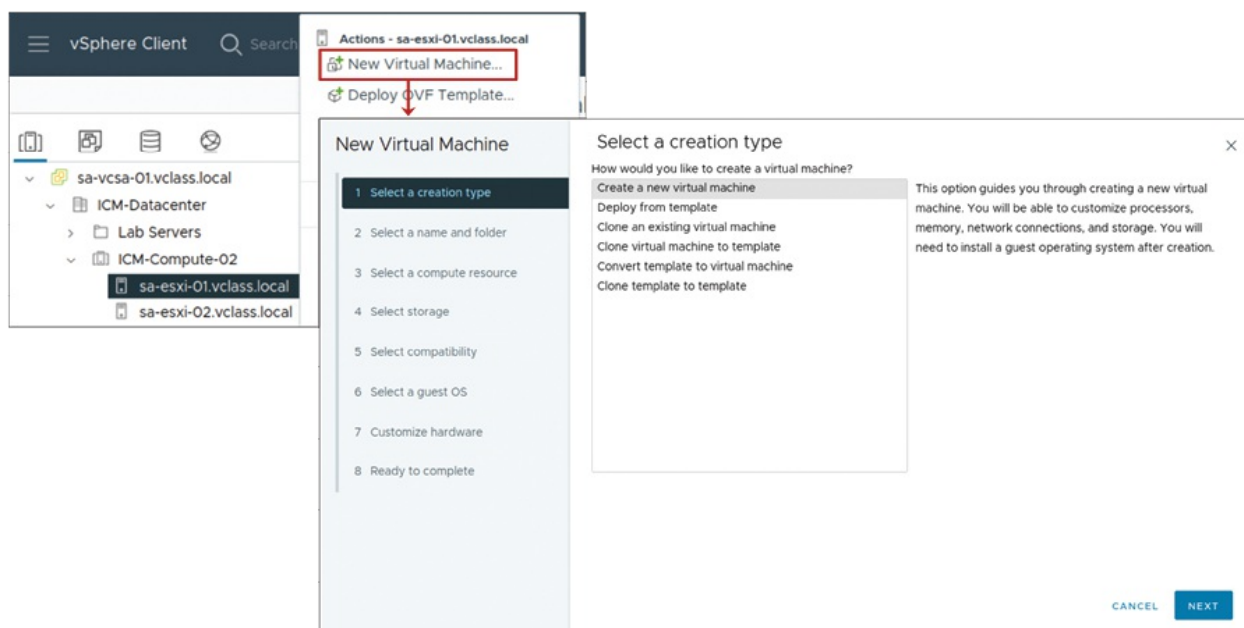


Figure 7.1: New VM wizard

(Source: VMware)

The steps are as follows:

1. **VM name and folder placement:**

- a. **Assigning a name:** Provide a unique name for the VM to facilitate easy identification and management across the vSphere environment.
- b. **Folder organization:** Specify the folder where the VM will reside. Using folders helps streamline VM management, particularly in environments with a large number of VMs.

2. **Compute resource selection:** Choose where the VM will run. Available options include:

- a. **Host:** Assign the VM to a specific ESXi host.
- b. **Cluster:** Utilize cluster-level features like **high availability (HA)** or **distributed resource scheduler (DRS)** for resource pooling and fault tolerance.
- c. **vApp:** Group VMs logically with shared policies and settings.
- d. **Resource pool:** Assign the VM to a resource pool for better control and prioritization of resources.

The following figure illustrates the new virtual machine wizard:

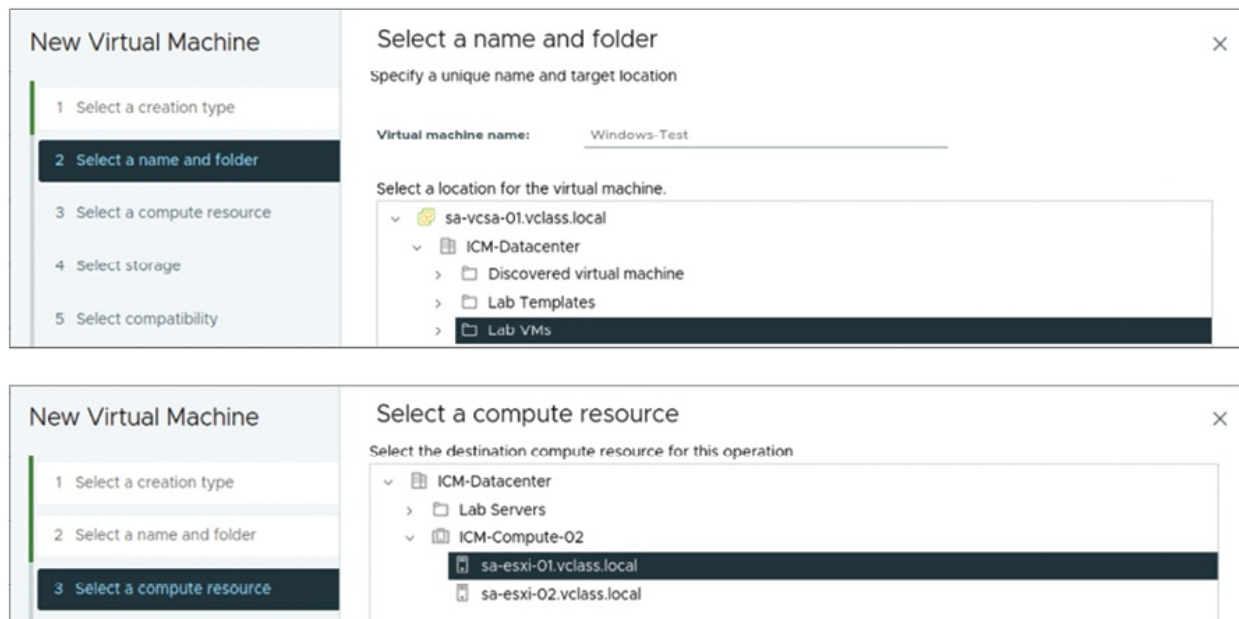


Figure 7.2: New Virtual Machine wizard: Name | Folder | Compute

(Source: VMware)

3. **Storage selection:** Define where the VM's files will be stored, ensuring alignment with performance and capacity requirements:
 - a. **Datastore selection:** Choose a datastore that matches the needs for size, speed, and redundancy.
 - b. **Storage policy:** If applicable, apply a policy to enforce requirements such as encryption or replication. The datastores presented will depend on the compute resource chosen earlier.
4. **Compatibility selection:**
 - a. **Specify the ESXi version:** Select the ESXi version the VM will be compatible with.
 - b. **Hardware features:** Choosing the latest version unlocks the most recent virtual hardware capabilities.
 - c. **Flexibility:** Ensures the VM can be migrated between hosts running different ESXi versions without compatibility issues.

The following figure illustrates the storage selection in new VM wizard:

The figure consists of two screenshots of the VMware New Virtual Machine wizard, showing the 'Select storage' and 'Select compatibility' steps.

Top Screenshot: Select storage

The wizard is titled 'New Virtual Machine' and shows a progress bar with six steps: 1. Select a creation type, 2. Select a name and folder, 3. Select a compute resource, 4. Select storage (current step), 5. Select compatibility, and 6. Select a guest OS.

The 'Select storage' step includes the following options:

- ☐ Encrypt this virtual machine (Requires Key Management Server)
- VM Storage Policy: Datastore Default
- ☐ Disable Storage DRS for this virtual machine

A table lists available datastores:

	Name	Storage Compatibility	Capacity	Provisioned	Free	Type	Cluster
<input type="radio"/>	ICM-Datastore	--	119.75 GB	136.83 GB	47.32 GB	VMFS 6	
<input checked="" type="radio"/>	ISCSI-Datasto...	--	129.75 GB	112.71 GB	83.51 GB	VMFS 6	
<input type="radio"/>	NFS-Datastore	--	7.26 GB	4 GB	6.98 GB	NFS v4.1	
<input type="radio"/>	Shared-VMFS	--	17.5 GB	1.66 GB	15.84 GB	VMFS 6	

Items per page: 10, 4 items

Bottom Screenshot: Select compatibility

The wizard is titled 'New Virtual Machine' and shows a progress bar with five steps: 1. Select a creation type, 2. Select a name and folder, 3. Select a compute resource, 4. Select storage, and 5. Select compatibility (current step).

The 'Select compatibility' step includes the following options:

- Select compatibility for this virtual machine depending on the hosts in your environment
- The host or cluster supports more than one VMware virtual machine version. Select a compatibility for the virtual machine.
- Compatible with: ESXi 8.0 and later
- Virtual machines using hardware version 20 provide the best performance and latest features available in ESXi 8.0.

Figure 7.3: New Virtual Machine wizard: Storage and compatibility

(Source: VMware)

5. Guest operating system selection:

- a. **Guest OS family:** Select the OS to be installed on the VM to ensure compatibility with the virtual hardware.
- b. **Guest OS version:** The system defaults to BIOS or EFI based on the OS.
- c. If the OS supports both, you can modify this setting after creating the VM but before installing the OS.

Note: Selecting EFI prevents booting OS versions that support only BIOS, and vice versa.

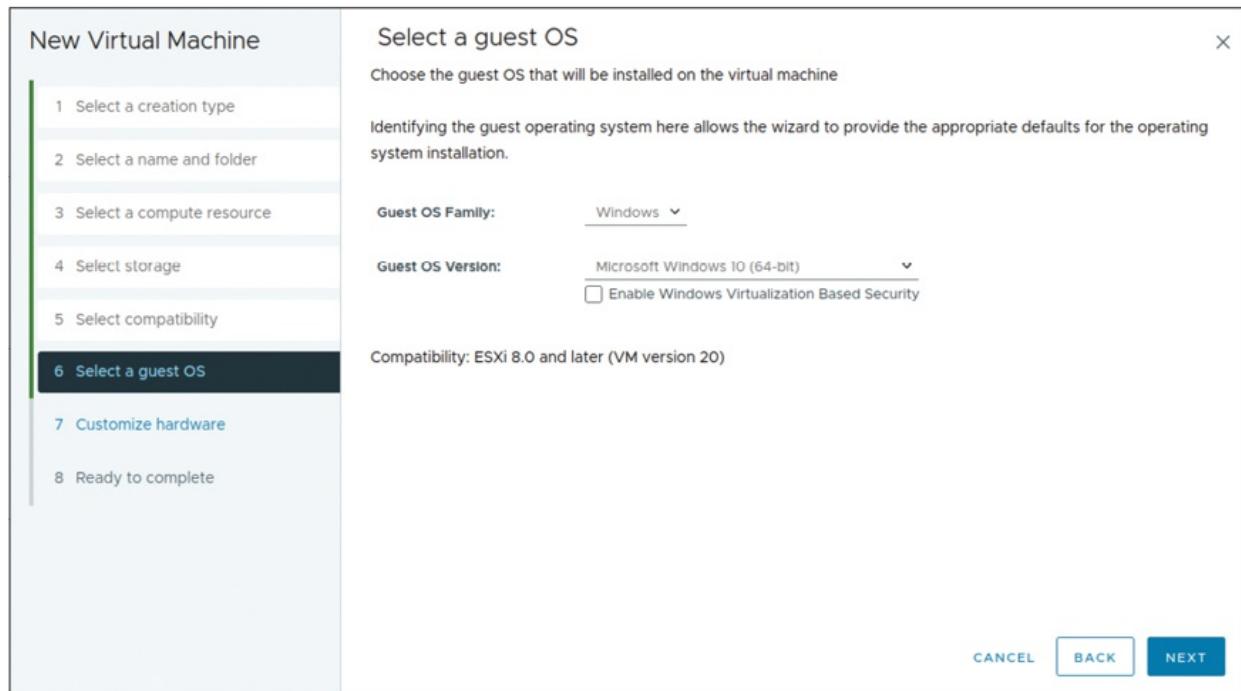


Figure 7.4: New Virtual Machine wizard: Guest OS

(Source: VMware)

6. Customize hardware configuration:

- a. **Default settings:** The system pre-configures CPU, memory, and disk size based on the selected OS.
- b. **Custom adjustments:** Tailor CPU cores, memory allocation, disk size, network adapters, and other resources to meet workload requirements.

- c. **Installation media:** Attach an ISO image containing the OS installation files, either from local storage or a content library, to initiate the OS installation.

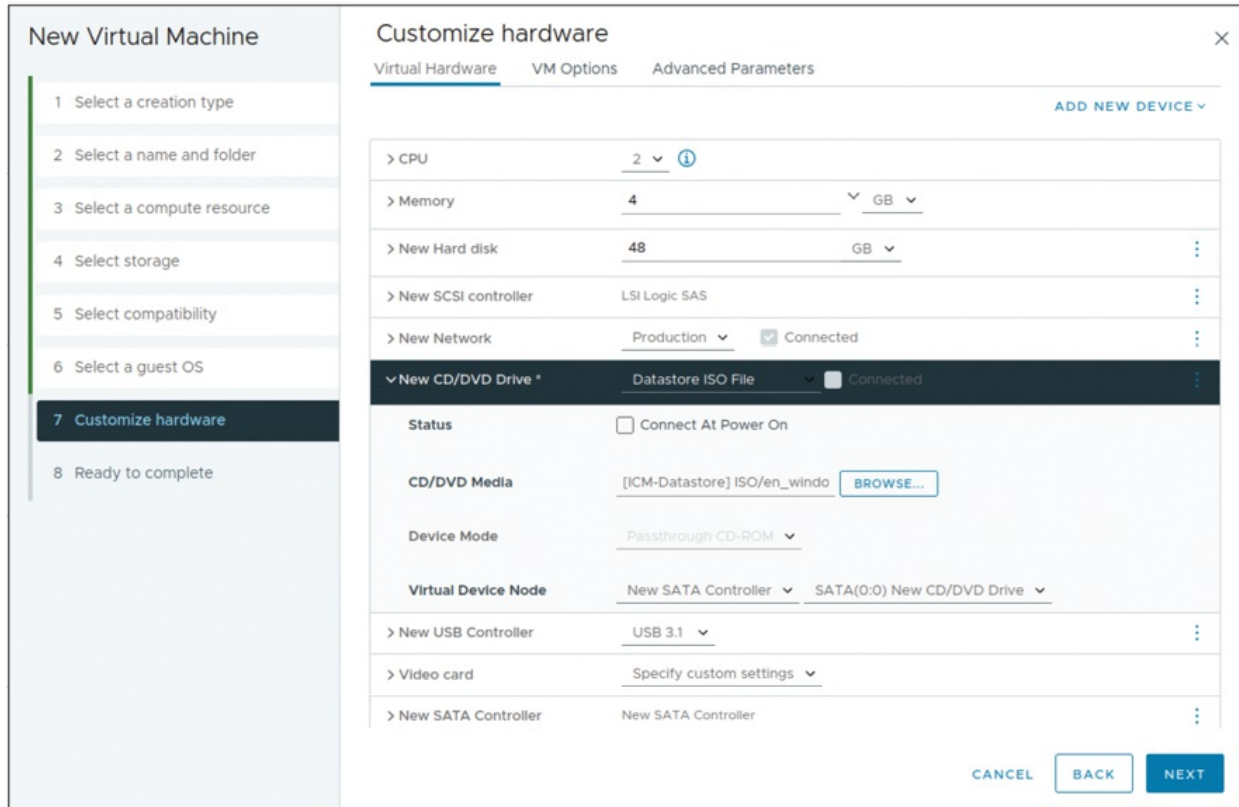


Figure 7.5: New Virtual Machine wizard: Virtual hardware

(Source: VMware)

By carefully configuring these steps, administrators ensure that the VM operates efficiently, meets workload demands, and integrates seamlessly into the virtualized infrastructure.

Guest operating system installation

The process of guest **operating system (OS)** installation on a virtual machine is exactly the same as on a physical computer. To begin the installation, interact with the VM through the **VM Console** in the **vSphere Client**. Administrators need to attach an installation media to the virtual CD/DVD drive or an **ISO image**, which is often faster and more convenient.

Figure 7.6 displays an example of **Windows Server 2008** installation; the

administrator can attach the ISO image or CD to the virtual machine and proceed with the OS installation through the VM Console. The installation process is very simple and follows the same prompts and configurations as a traditional physical server.

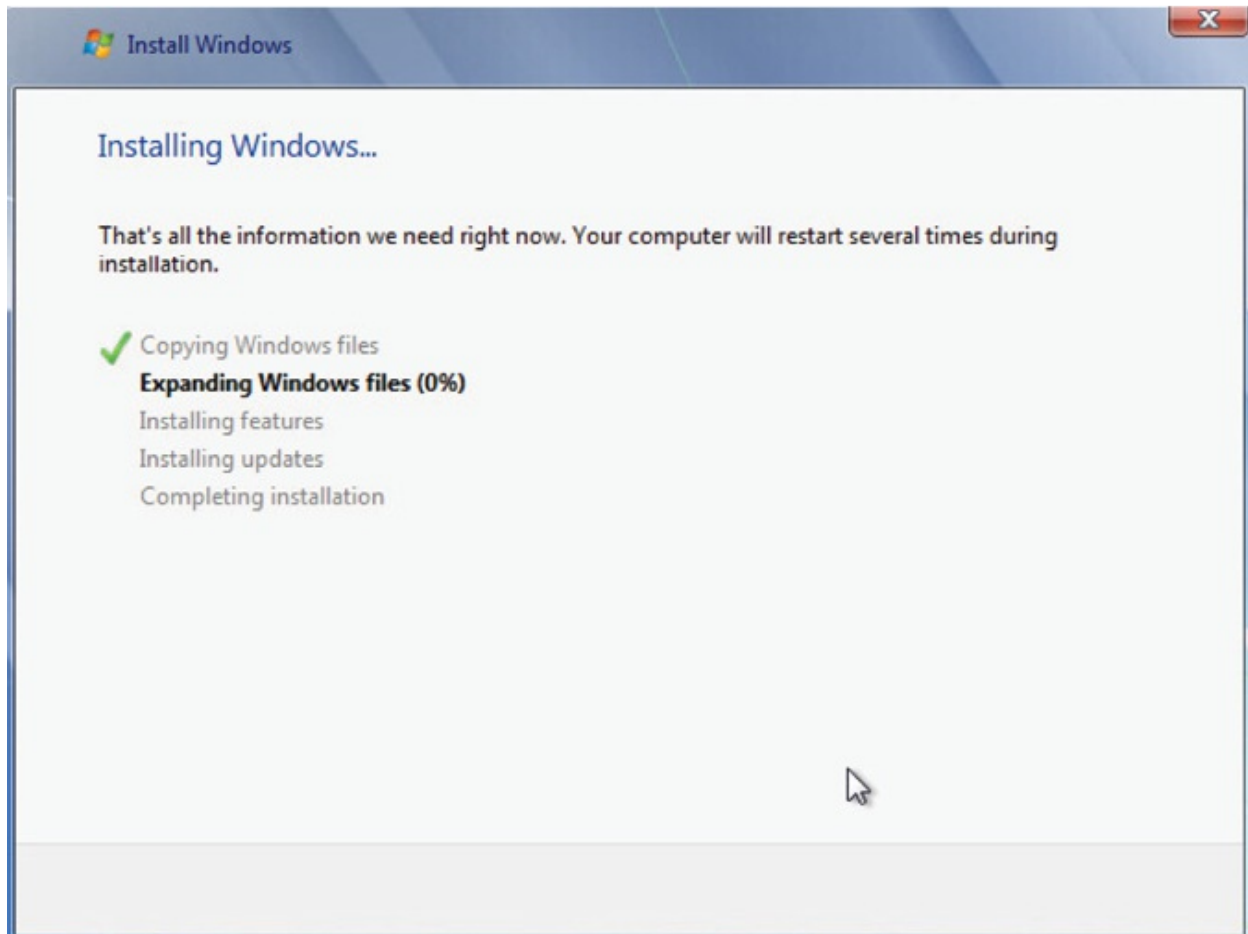


Figure 7.6: New Virtual Machine wizard: Guest OS installation

(Source: VMware)

OVF templates deployment

Open Virtualization Format (OVF) templates provide an efficient way to deploy virtual machines or virtual appliances in vSphere. A virtual appliance is a preconfigured VM designed for a specific function, such as security, backup, or network management.

Administrators can import the Virtual appliances directly into vCenter Server or an ESXi host from sources like VMware Marketplace at

<https://vcf.broadcom.com/vsc/>. Since OVF templates are compressed, they allow for faster downloads and deployments.

To deploy an OVF template using the **vSphere Client**, follow these steps:

1. Right-click a data center or ESXi host and select Deploy OVF Template.
2. In the Deploy OVF Template wizard, click Upload Files to select the template files.
3. The vSphere Client will validate the template and ensure compatibility with the target environment before deployment.

The following figure illustrates the OVF template deployment:

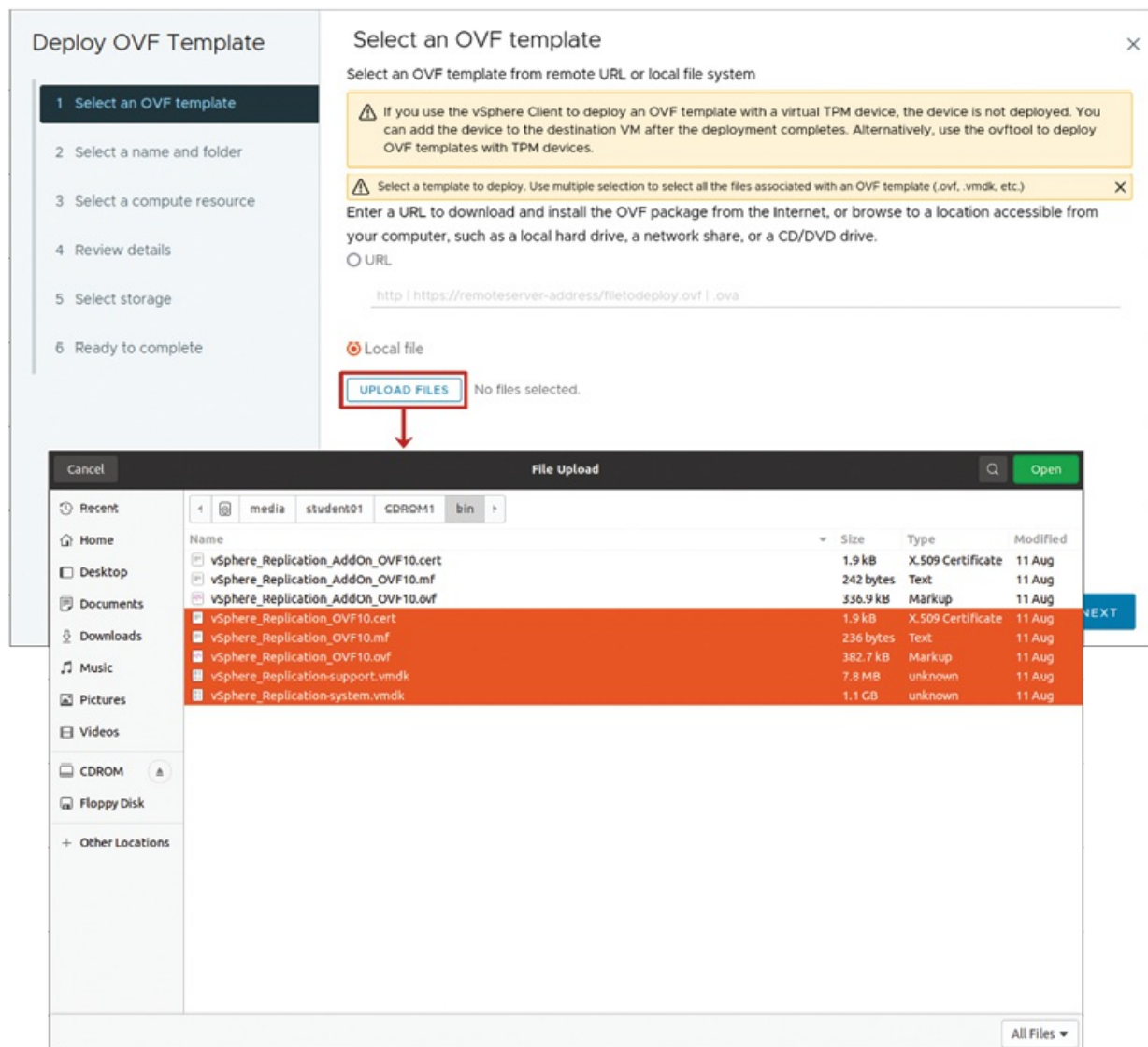


Figure 7.7: Deploying OVF Templates

(Source: VMware)

Removing virtual machines

Following are two options for removing a VM, depending on whether to retain the VM's files for future use or permanently delete them:

- **Remove from inventory:** This option unregisters the VM from the ESXi host and vCenter Server and keeps its files intact on the datastore. If needed, the VM can be re-registered using the datastore browser.
- **Delete from disk:** This permanently deletes all VM-related files from the datastore, including configuration, virtual disks, and snapshots. The VM is also unregistered from vCenter and the ESXi host, making recovery impossible unless a backup exists.

Always maintain a proper method for flexible lifecycle management, holding data for future use, or freeing storage resources.

The following figure illustrates how to remove VM from inventory or delete from disk:

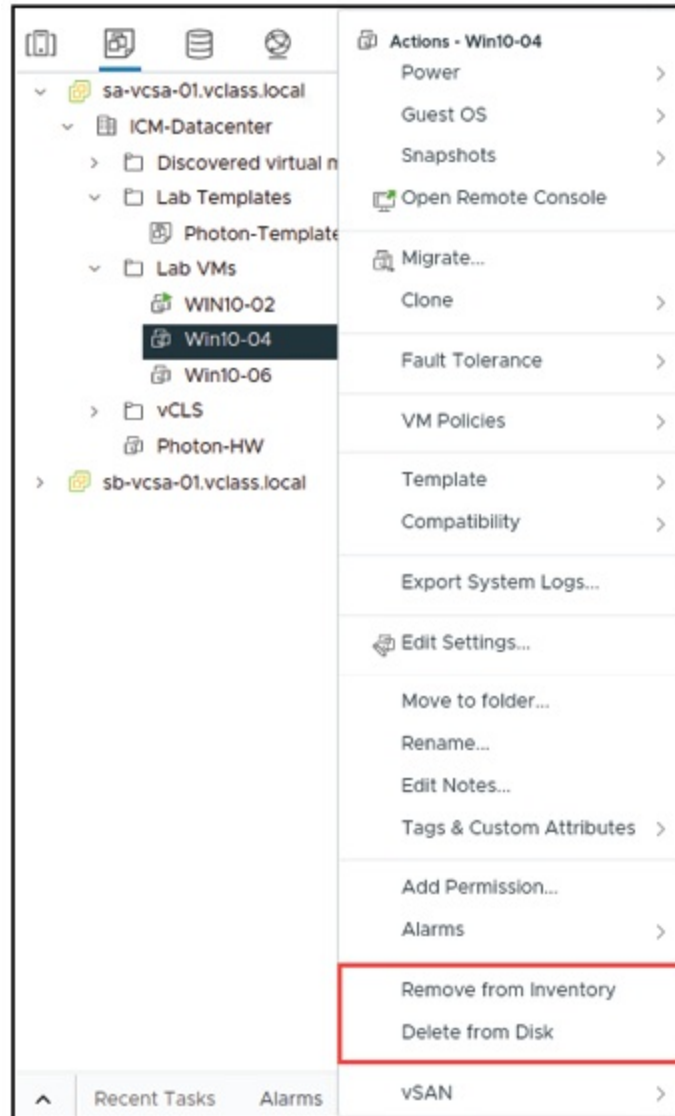


Figure 7.8: Removing VMs

(Source: VMware)

VMware Tools

VMware Tools is an important utility that improves the performance, usability, and manageability of virtual machines. It replaces generic operating system drivers with VMware-optimized drivers, ensuring seamless integration with virtual hardware.

The key features and benefits are as follows:

- **Optimized drivers:** Includes SVGA display, Balloon driver for memory

management, network (VMXNET/VMXNET3), and storage (paravirtual SCSI) drivers for improved performance.

- **Enhanced usability:** Boosts graphics and mouse performance, enables shared folders, and supports copy-paste between guest and host systems.
- **Time synchronization:** Keeps the guest OS clock aligned with the host system.
- **Remote management:** Allows remote shutdown and restart of VMs.
- **Guest OS monitoring:** Provides heartbeat services for better monitoring and automation.

While VMware Tools is optional, it is highly recommended for optimal performance and full functionality.

Installing VMware Tools

For optimal performance of the virtual machines, always install the latest version of VMware Tools supported by the guest operating system.

The key consideration is as follows:

- It is highly recommended to refer to the respective vSphere version's Release Notes to confirm that the VMware Tools ISO images included in the vSphere version are being used in the environment.

The installation methods based on OS is as follows:

- **Windows:** Install using windows.iso for Windows Vista and later.
- **Linux:**
 - Install from **linux.iso**.
 - Alternatively, use **open-vm-tools**, which can be retrieved using standard Linux package managers such as **yum**, **apt**, or **rpm**.

By choosing the right method based on the OS, administrators can have a trouble-free VMware Tools installation experience.

Downloading VMware Tools

The following figure illustrates how to access the VMware Tools product download page to download a specific version of VMware Tools:

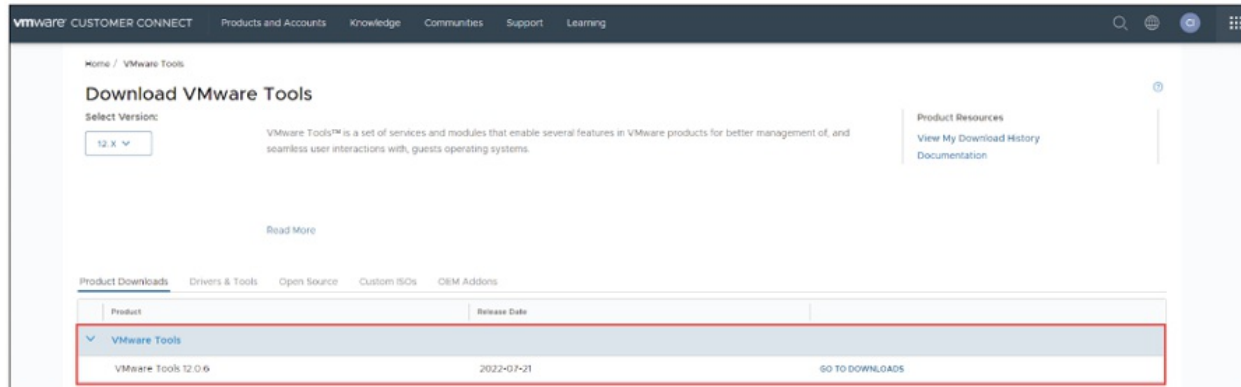


Figure 7.9: Downloading VMware Tools

(Source: VMware)

For step-by-step installation and further details on Open VM Tools, see the VMware Tools Administration guide at <https://techdocs.broadcom.com/>.

Virtual machines components

A VM consists of three core parts: the hypervisor, virtual hardware, and the guest operating system. Virtual hardware mimics real physical components so that the guest operating system can function as if it were running on dedicated hardware. The guest OS is the system installed and executing within that virtual environment, managing applications and user interactions just like on a physical machine.

Virtual machine encapsulation

Every virtual machine in the vSphere universe consists of files or objects, depending on the storage. Virtual machine objects in VMFS or NFS storage systems are files in a special directory, whereas vSAN and vSphere **Virtual Volumes (vVols)** refer to them as objects. Every virtual disk is an independent file or object, which is easier to move and manage. Encapsulation facilitates easier movement, backup, and replication of virtual machines, and it maintains storage systems synchronized.

The following figure illustrates the virtual machine encapsulation:

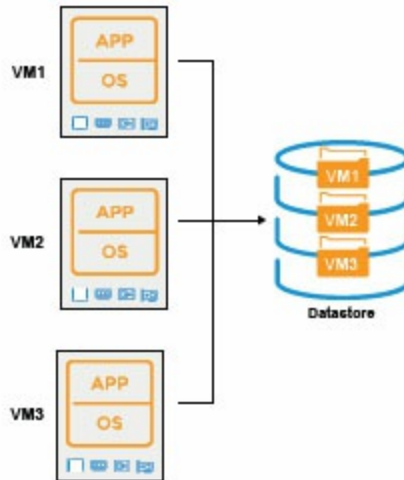


Figure 7.10: Virtual machine encapsulation

(Source: VMware)

Virtual machine files

A vSphere virtual machine is made up of many configuration, storage, and run-state files that make it up. The files are stored in a specific directory in a datastore.

The following are common VM file types:

- **Configuration file (.vmx):** Saves the system and hardware configurations of the VM. This file is overwritten by a **.vmtx** template configuration file when the VM is converted into a template.
- **Swap file (.vswp):** The file is created every time a VM is booted. It assists in memory utilization when there are insufficient resources.
- **BIOS/EFI file (.nvram):** Serves to store the VM's firmware settings, just like BIOS or EFI settings in physical systems.
- **Log files (.log):** VM activity and errors are logged in **vmware.log**, and previous logs are **-1.log** through **-6.log**. Logs are reset when the VM is restarted, and older files are removed.
- **Virtual Disk Files (.vmdk and -flat.vmdk):** A virtual disk consists of a descriptor file (**.vmdk**) and a flat file (**-flat.vmdk**) with the actual data.

Figure 7.11 displays couple of key VM files, vSphere also generates other files depending on features like snapshots, suspend states, and replication:

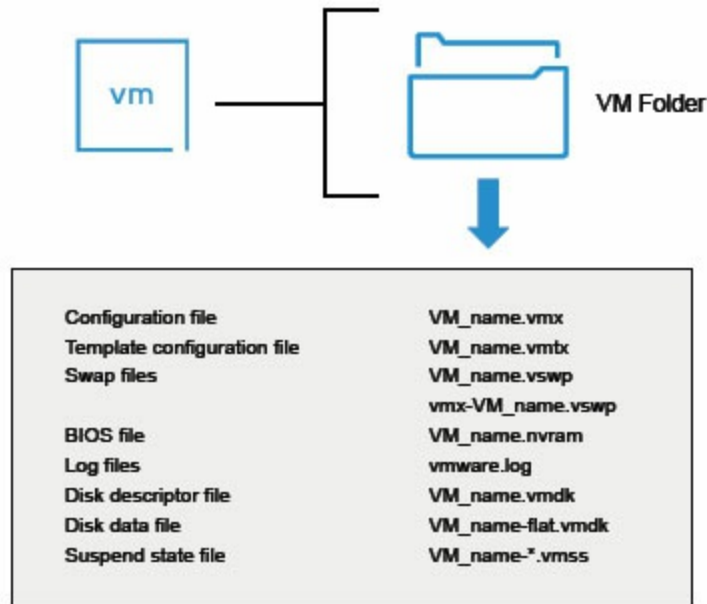


Figure 7.11: Virtual machine files

(Source: VMware)

Understanding virtual machine virtual hardware

A virtual machine exposes a consistent hardware set to the guest operating system, so VMs are relocatable from one VMware environment to another. The guest OS sees virtual hardware in the same way it sees physical hardware without realizing that devices are virtualized.

The configurable virtual hardware components are as follows:

- **CPU and memory:** CPU and memory configurations can be set to optimize VM performance. Highly advanced CPU features, including virtual performance counters, are supported.
- **Storage devices:** VMs provide several virtual hard disks, and multiple controllers, including SCSI, SATA, and NVMe, are supported. PVSCSI adapters support up to 64 virtual SCSI targets, but other controllers support up to 15 targets. The SATA controller is exposed as an AHCI SATA controller to the guest OS.
- **Network and USB devices:** Virtual NICs can be added to provide access to VMs onto networks, while USB devices (such as security dongles and mass storage) can be inserted into a VM but are accessible only from the host on which they are locally plugged in. Only one VM is allowed to

access a USB device at a time.

- **Virtual Machine Communication Interface (VMCI):** VMCI provides improved communication between the VM and hypervisor by minimizing network overhead. It provides socket APIs to enable datagram-based (such as UDP) as well as connection-oriented (such as TCP) communications. VMCI is disabled by default.

Figure 7.12 displays virtual hardware of VMware exhibits flexibility as well as scalability with being workload-compatible with a very wide range of workloads:

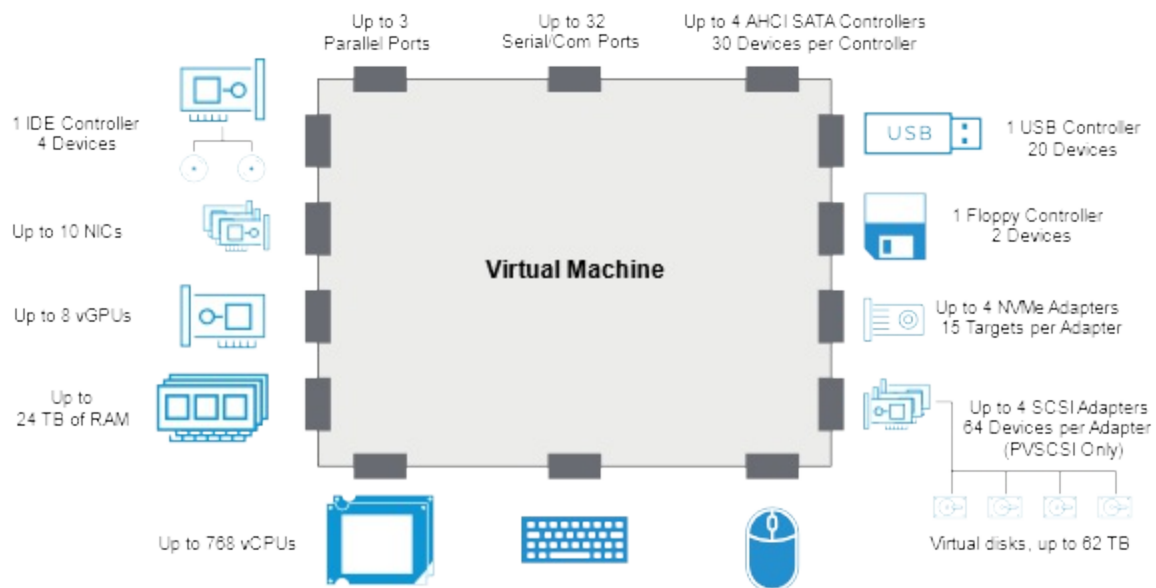


Figure 7.12: About VM virtual hardware

(Source: VMware)

Virtual hardware versions and compatibility

The virtual hardware version, or VM compatibility level, determines what a VM can do. This version and a particular ESXi release can be combined to offer improved hardware and guest operating system compatibility.

To have optimal performance and compatibility, always select a virtual hardware version compatible with the vSphere environment.

The compatibility matrix is shown in the following table:

ESXi version	Supported virtual hardware version

ESXi 8.0 U2 (8.0.2)	21
ESXi 8.0	20
ESXi 7.0 U2 and later	19
ESXi 7.0 U1 and later	18
ESXi 7.0 and later	17
ESXi 6.7 U2 and later	15
ESXi 6.7 and later	14

Table 7.1: Supported virtual hardware version

Note: VMware Workstation and Fusion Pro use distinct hardware versions, such as version 16, which is exclusive to those platforms.

To learn more, refer to VMware's Virtual Machine Hardware Versions Knowledge Base at <https://knowledge.broadcom.com/external/article?legacyId=1003746>.

Optimizing CPU and memory in VMs

Administrators can add, change, or configure CPU and Memory to ensure optimal VM performance in vSphere.

The CPU allocation is as follows:

- vCPUs depend on the host's logical CPUs, guest OS, and hyperthreading.
- **ESXi 8.0** supports up to **768 vCPUs per VM**, but a VM cannot exceed the host's logical CPU count.
- **Multicore vCPU configuration** helps OS environments with socket-based licensing optimize CPU usage.

The memory allocation is as follows:

- A VM's memory is limited by the ESXi version, host capacity, and guest OS.
- ESXi 8.0 allows up to 24 TB RAM per VM.
- Memory changes may require powering off the VM.
- The virtual hardware memory limit determines the max usable memory for applications.

By properly sizing CPU and memory, administrators can maximize performance while ensuring efficient resource utilization.

The compute maximums are provided in the following table:

Resource	vSphere 8
Virtual CPU per VM	768
Memory per VM	24 TB
CPU per host	896
Memory per host	24 TB
Hosts per cluster	96

Table 7.2: Compute maximums

For latest and greatest information on configuration maximums, always refer <https://configmax.broadcom.com/>.

Understanding virtual storage in vSphere

Virtual machines access storage through virtual storage adapters, which serve as a middleman between the VM and the storage resources of the ESXi host. ESXi offers several virtual storage adapter choices, each designed for various workloads and performance requirements.

The types of virtual storage adapters are as follows:

- **BusLogic Parallel:** An older SCSI adapter with minimal functionality.
- **LSI Logic Parallel:** An Ultra320 SCSI controller, often used for legacy systems.
- **LSI logic SAS:** A serial interface adapter with enhanced performance and compatibility.
- **VMware Paravirtual SCSI (PVSCSI):** Built for high-latency and high-throughput workloads.
- **AHCI SATA Controller:** Supports virtual disk, CD/DVD device, and other SATA access.
- **Virtual NVMe:** Optimized for high-speed flash storage by using PCIe to access data quicker.

Thick provisioned virtual disks

Thick provisioning allocates all space allocated when it is provisioned, irrespective of how much the guest OS will consume. Utilization of the storage is guaranteed to be predictable but may have some effect on available capacity.

The types of thick-provisioned disks are as follows:

- **Lazy-zeroed thick disk (Default):** Zeros ahead of time, but does not zero existing data. Blocks will be zeroed on first write, which creates some performance overhead.
- **Eager-zeroed thick disk:** Provisions space and fills all the blocks with zeroes at the time of creation. This provides faster write performance and is necessary for features such as fault tolerance.

Choosing between these types depends on performance needs and storage space efficiency. Eager-zeroed disks provide better performance, but lazy-zeroed disks conserve setup time.

The following figure illustrates the thick provisioned virtual disks:

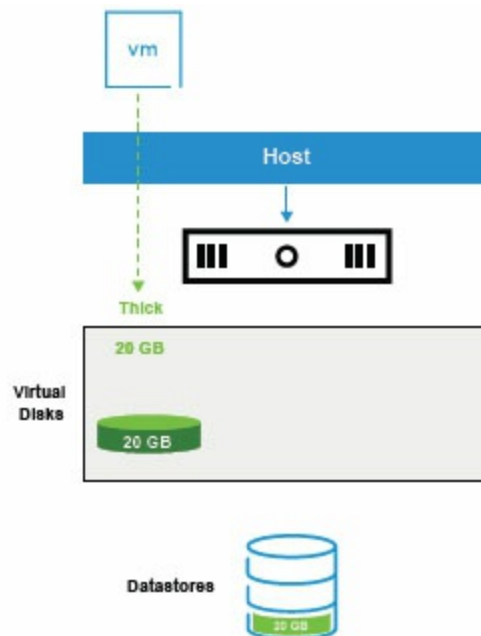


Figure 7.13: Thick provisioned virtual disks

(Source: VMware)

Thin provisioned virtual disks

Thin provisioning allows VMs to use storage dynamically, consuming only the space needed for current data while appearing as fully allocated to the guest OS. This approach optimizes storage utilization and enables overallocation, provided monitoring and alerts are in place.

The key features are as follows:

- **Space-efficient:** Virtual disks grow as needed, reducing initial storage consumption.
- **Full capacity visibility:** The guest OS always sees the full allocated disk size.
- **Storage reclamation:** Use the **unmap** command to reclaim unused space.
- **Monitoring and alerts:** Helps track storage allocations and prevent overcommitment risks.
- **Flexible storage management:** Thick and thin disks can coexist on the same datastore.

Figure 7.11 displays an example, if **140 GB** is allocated across virtual disks but only **80 GB** is actively used, a datastore with **100 GB** of capacity can still support the workload efficiently. Thin provisioning is particularly useful in environments where storage optimization and flexibility are key concerns.

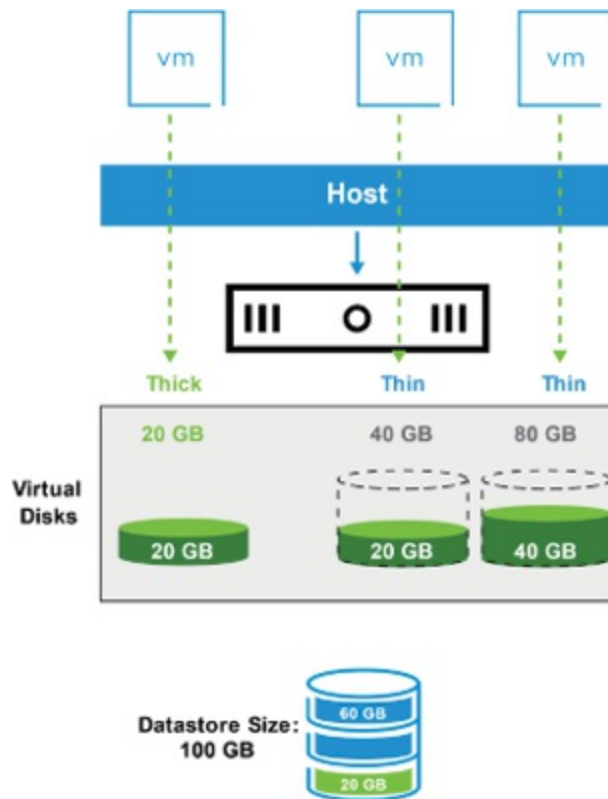


Figure 7.14: Thin provisioned virtual disks

(Source: VMware)

Thin provisioned disks datastore management

Thin provisioning maximizes storage utilization but must be closely monitored to avoid overcommitment. A datastore is overcommitted when the combined provisioned size of thin-provisioned disks is greater than its capacity. When the datastore is full, VMs will stall when they try to write new data.

The best practices for monitoring and managing capacity are as follows:

- **Set alarms and reports:** Set up alarms for datastore disk overallocation and VM disk usage to catch issues early.
- **Expand storage when needed:** Dynamically expand datastore capacity to prevent service interruptions.
- **Use vSphere Storage vMotion:** Move VMs between datastores to rebalance space usage, such as to convert thick-provisioned disks to thin-provisioned disks if necessary.

By proactively tracking and controlling datastore capacity, administrators can achieve optimal storage efficiency while providing consistent VM performance.

Understanding virtual networks in vSphere

Virtual networks allow VMs and physical machines to communicate. In setting up networking for a VM, administrators need to choose the right network adapter type, assign the VM to a port group, and specify the network connection state. They can also set up whether the VM will automatically connect to the network when it starts.

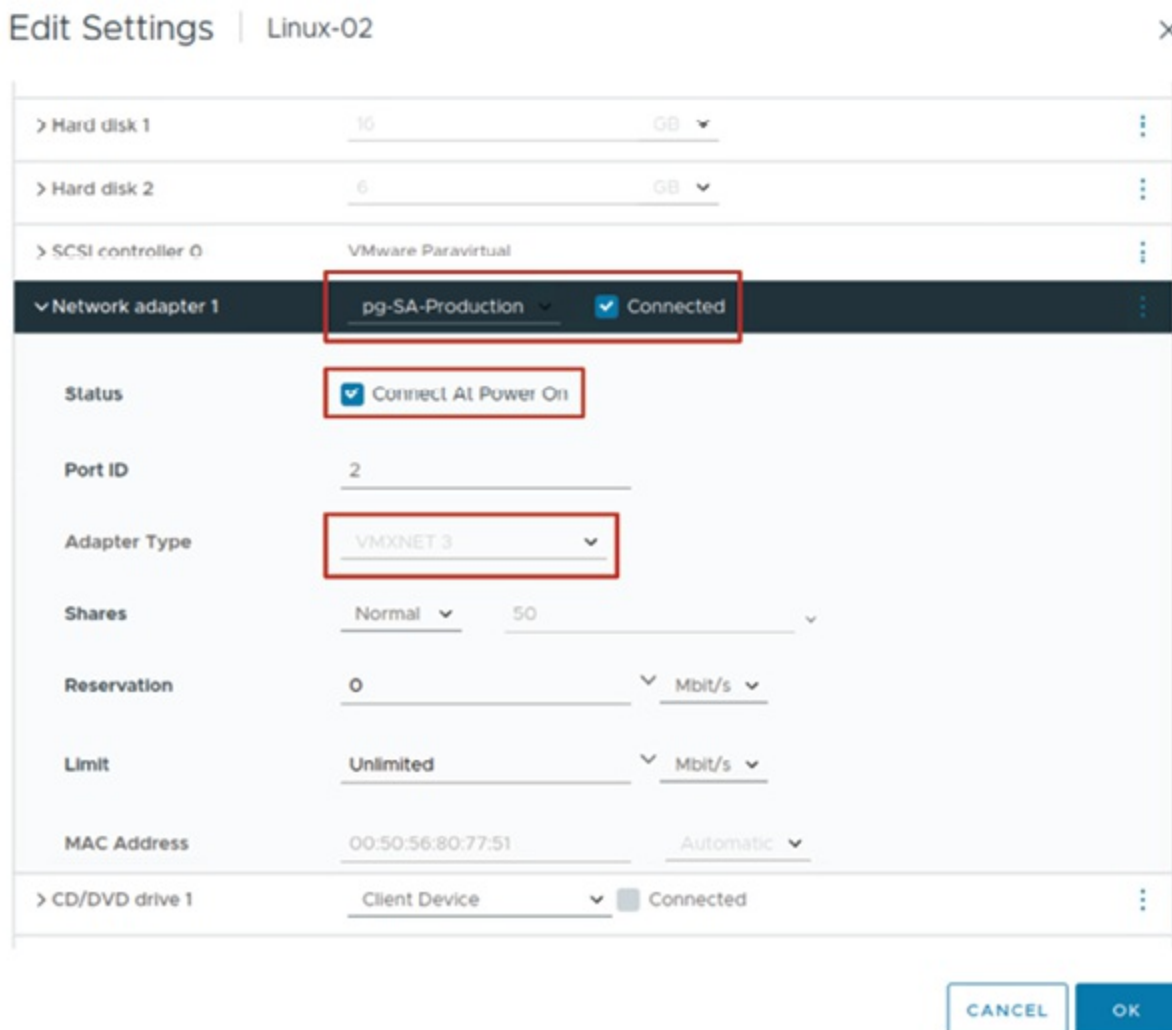


Figure 7.15: About virtual networks

(Source: VMware)

Virtual network adapters in vSphere

When setting up a VM in vSphere, administrators can install network adapters (NICs) and select an appropriate adapter type depending on compatibility, performance, and guest OS needs. The VMXNET3 adapter should be the priority because of its improved performance and support for advanced features.

The supported NIC types vary based on hardware version, VM compatibility level, and guest OS support. The choices are E1000/E1000E, which simulate Intel Gigabit Ethernet adapters, Flexible, which can be used as either Vlance or VMXNET, and VMXNET3, a paravirtualized, high-performance NIC that needs VMware Tools.

For robust workloads, PVRDMA supports RDMA-like communication between VMs with low latency and high bandwidth, and SR-IOV pass-through enables direct VM-to-physical adapter data transfer with lower network latency.

PCI passthrough devices in vSphere

PCI passthrough provides physical direct access to devices from a virtual machine with increased performance and improved resource use. VMware can implement various kinds of passthrough configurations, all offering different strengths.

vSphere DirectPath I/O provides access by a VM of a reserved PCI or PCIe device directly connected to a single ESXi server. The VM cannot be accessed, though, other than that server, via restrictions on vMotion, snapshot, and suspend/resumptions.

vSphere Dynamic DirectPath I/O provides more flexibility by dynamically mapping PCI passthrough devices within a cluster. This enables vSphere DRS to put VMs on any host in the cluster that offers an equivalent device, providing more balanced workload distribution.

NVIDIA GRID vGPU technology accelerates graphics performance through the sharing of a single GPU by multiple VMs, optimizing workloads like AI, 3D rendering, and **virtual desktop infrastructures (VDI)**. It minimizes CPU overhead with near-native performance for graphics-intensive workloads.

Other virtual devices within vSphere

Though each VM needs a vCPU and virtual memory, some other virtual devices add to its capabilities:

- **CD/DVD drive:** Provides support for physical or virtual media like ISO images to install software.
- **USB 3.0 and 3.1:** Accommodates both host-connected and client-connected USB devices with fast data transfer rates and peripheral support.
- **Floppy drive:** While seldom used now, a VM can also be attached to a virtual or physical floppy drive for older software.
- **Generic SCSI devices:** Enabling VMs to attach to more SCSI adapters, adding storage possibilities.
- **nvGPUs (Virtual GPUs):** Uses physical GPU capacity for computational-intensive workloads like AI, machine learning, or graphics processing.
- **Precision clock:** Keeps the VM system time in synchronization with the lead ESXi host, enhancing accurate time in applications that are highly sensitive.
- **Virtual Trusted Platform Module (TPM) 2.0:** Implements hardware-supported security features, such as encryption and secure boot, to meet standards of contemporary security.

Navigating vSphere client for VM management

To control VMs through the vSphere client, administrators can make use of the integrated VM Console or **VMware Remote Console (VMRC)**. The vSphere Client provides a user interface for virtual machine access and management, such as opening the console for direct access to the guest operating system.

Understanding the VM Console

The VM Console in vSphere provides administrators direct access to the keyboard, mouse, and display of a VM so they can have complete control

over its activities. This is particularly helpful in initial setup, troubleshooting, and maintenance when other methods of accessing the VM remotely are not possible.

Accessing the VM Console

Administrators may launch the VM Console from the vSphere Client in two methods:

- **Web console:**
 - Opens the VM Console in a new **browser tab**.
 - Highly convenient when quick retrieval is required without using other software.
- **VMware Remote Console (VMRC):**
 - An independent **application** that runs in its own window.
 - Provides greater control by allowing you to install local or remote devices such as USB drives.
 - Suitable for extended sessions or where you need to utilize local facilities.

The following figure illustrates the VM Console:

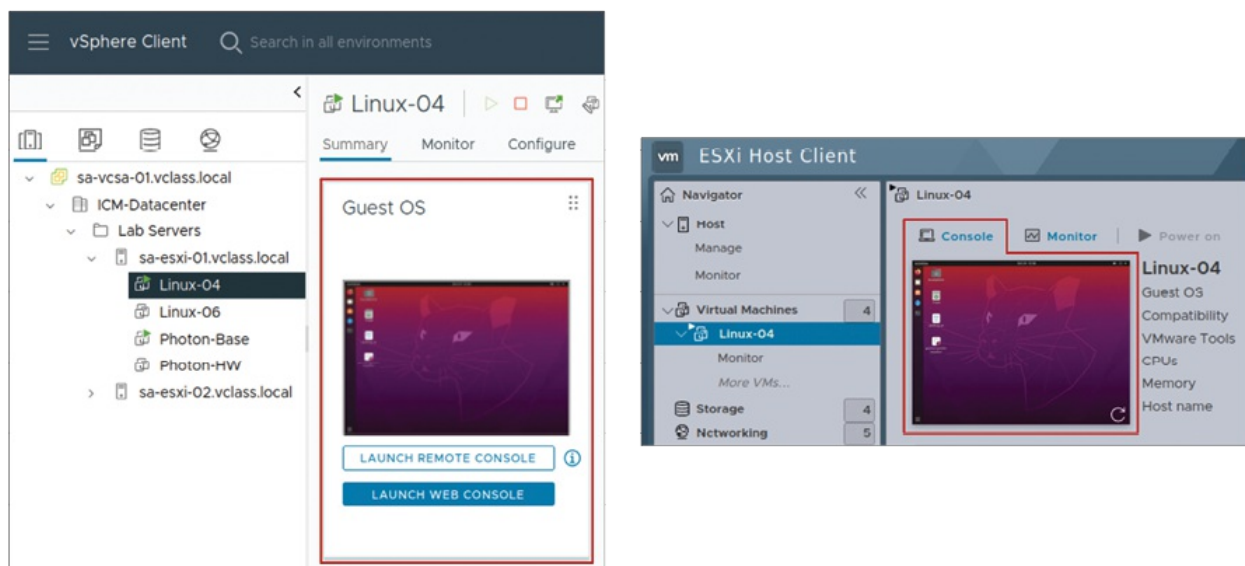


Figure 7.16: About VM Console

(Source: VMware)

Common use cases for the VM Console

Unlike tools like **Remote Desktop Connection (RDP)** or SSH, which provide *network-based access* to VMs, the VM Console remains functional *even when the network is down*. It is used for:

The VM Console functions even without a network, in contrast to tools such as **Remote Desktop Connection (RDP)** or SSH, which require a network in order to connect to VMs. It is utilized for:

- **BIOS or EFI access:** Change firmware settings, change boot orders, or repair boot issues.
- **Operating system installation:** Connect ISO images, install the OS software, or fix problems with the OS.
- **Power operations:** Restart, shutdown, suspend, or resume VMs, particularly when they fail to respond.
- **Troubleshooting:** Detect and fix problems when a VM becomes disconnected from the network, crashes, or needs offline diagnostics.

The key considerations for using the VM Console are as follows:

- **Performance constraints:** VM Console is not designed to be used on a daily basis such as running programs within the VM. It is less responsive compared to RDP, SSH, or VNC and therefore not appropriate for intensive use.
- **Security concerns:** Since it grants direct access to the console, provide it to only authorized administrators. Regularly check the audit logs to keep track of the access.
- **Multi-monitor support:** **VMware Remote Console (VMRC)** supports multi-monitors, which are easier to use for graphical workloads.

When to use the VM Console vs. other remote tools is shown in the following table:

Scenario	Recommended access method
VM is unresponsive or has no network connectivity	VM Console (Web or VMRC)
Installing or repairing the OS	VM Console (Attach ISO)
Running daily administrative tasks	RDP (Windows) or SSH (Linux)
Mounting local devices (USB, CD/DVD)	VMRC

Checking system logs on a frozen VM	VM Console
Managing multiple VMs remotely	vSphere Client or VMRC

Table 7.3: *VM Console recommended access method*

Virtual machines resources optimization

As shown in [Figure 7.17](#), **ADD NEW DEVICE** under VM edit settings allows administrators to add resources from the list for better performance and hardware control.

Some of these changes can be made while the VM is operational to accommodate the needs of workload, but some must be made while it is off, such as disks and network cards, to avoid system issues.

Administrators need to consider compatibility, workload effect, and resource availability for a stable virtual environment.

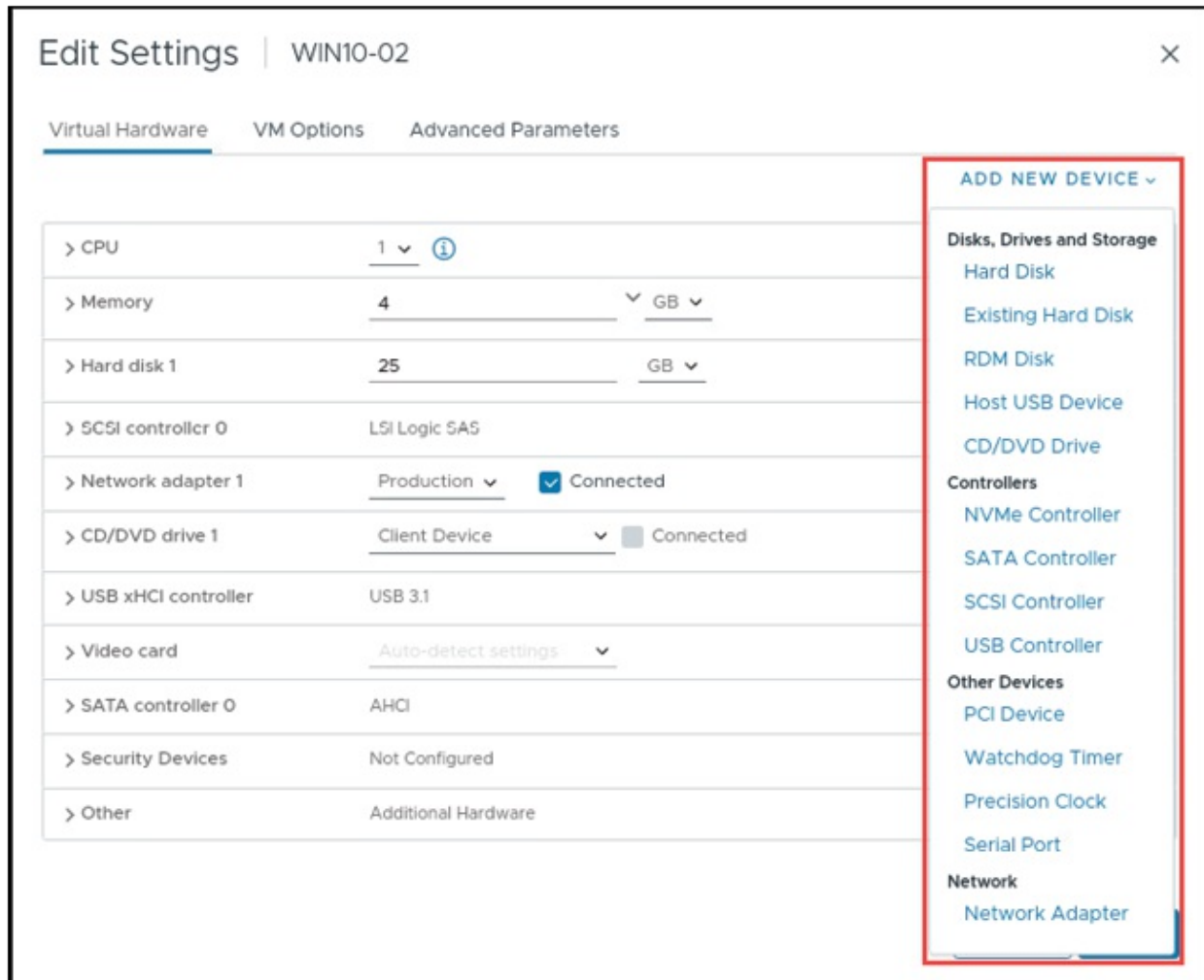


Figure 7.17: Modifying virtual machine settings

(Source: VMware)

Hot-pluggable devices overview

Hot-pluggable devices enable administrators to add resources to an existing VM in running state without downtime, enhancing flexibility and efficiency in a vSphere environment. Hot-pluggable devices supported include USB controllers, Ethernet adapters, and hard disks.

For supported guest operating systems, vSphere is also capable of CPU Hot Add and Memory Hot Plug, allowing extra CPU cores and memory to be dynamically allocated, provided that the following requirements are met:

- VMware Tools must be installed.
- Hardware version 11 or later on the VM.

- Guest OS support for hot-plug capability.
- Hot-plug options are enabled in the VM settings, as highlighted in [Figure 7.18](#):

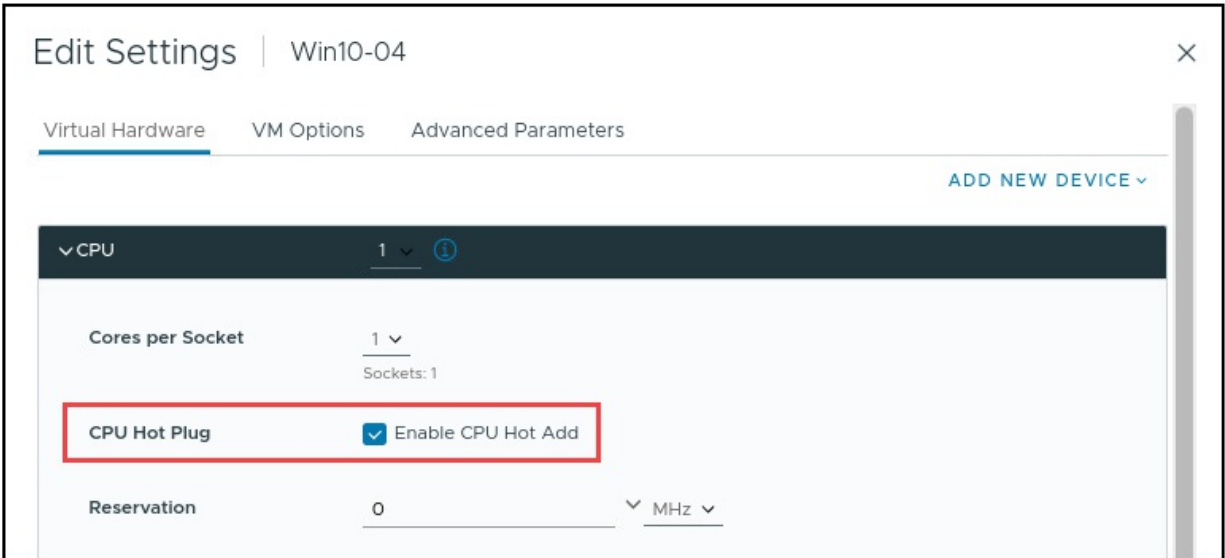


Figure 7.18: Hot-pluggable devices

(Source: VMware)

Increasing virtual disk size dynamically

vSphere enables administrators to expand a virtual disk during VM operation with no downtime, but the operation does have several important considerations:

- The VM cannot have attached snapshots.
- The added disk space could be unavailable and may need to be resized inside the guest OS.

Once the disk is expanded, utilize the necessary system tools from within the guest operating system to enlarge the file system and make use of the additional allocated space.

This capability allows for transparent expansion of storage without downtime, enhancing VM scalability and storage management.

The following figure illustrates how to increase virtual disk size dynamically:

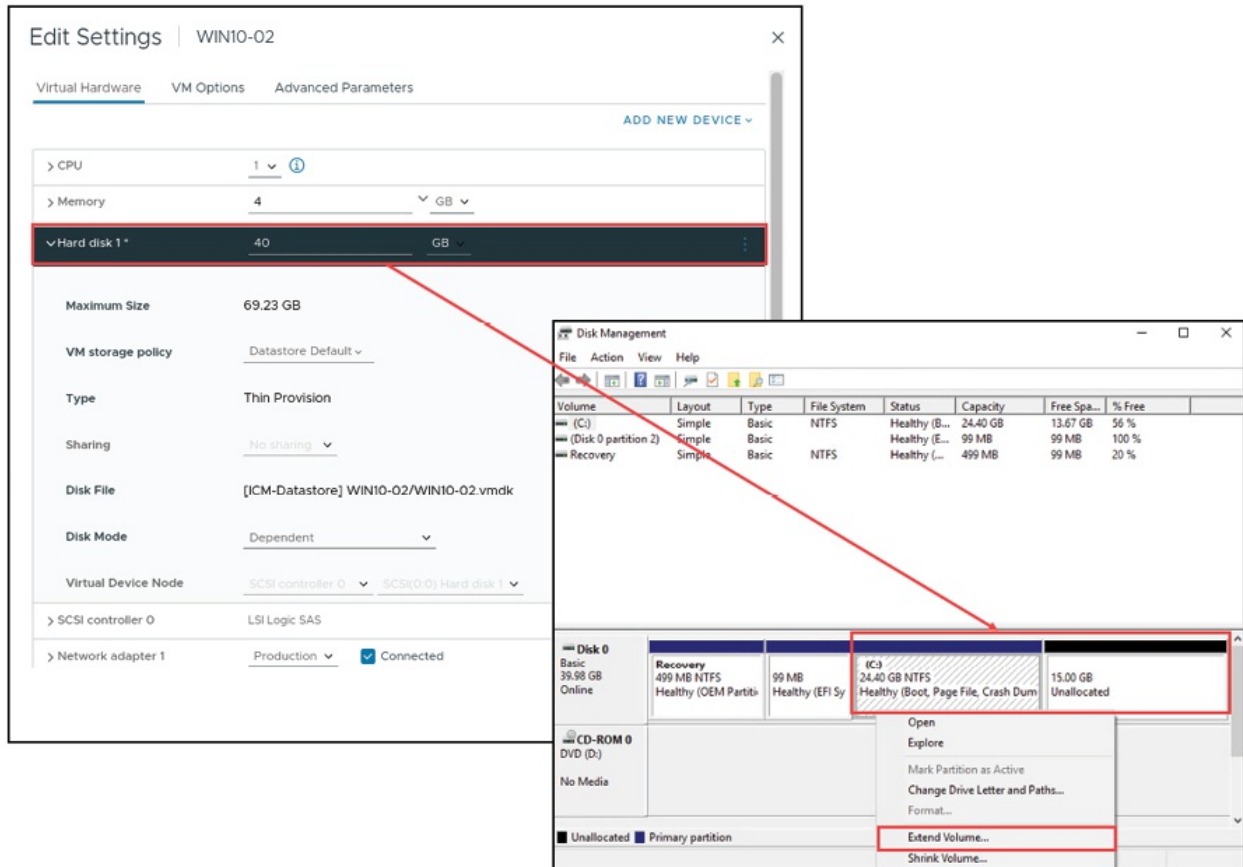


Figure 7.19: Dynamically increasing virtual disk size

(Source: VMware)

Inflating thin provisioned disks

Thin provisioned virtual disks can be converted to thick-provisioned, eager-zeroed disks, reserving space allocated all at once. It can be performed by either of the following:

- Inflate the disk by selecting the.vmdk file and selecting the Inflate option, as depicted in [Figure 7.20](#).
- Utilize vSphere Storage vMotion and select thick-provisioned for the destination disk format when moving.

Once inflated, the virtual disk occupies all provisioned datastore capacity, except for overcommitment, but limiting storage flexibility.

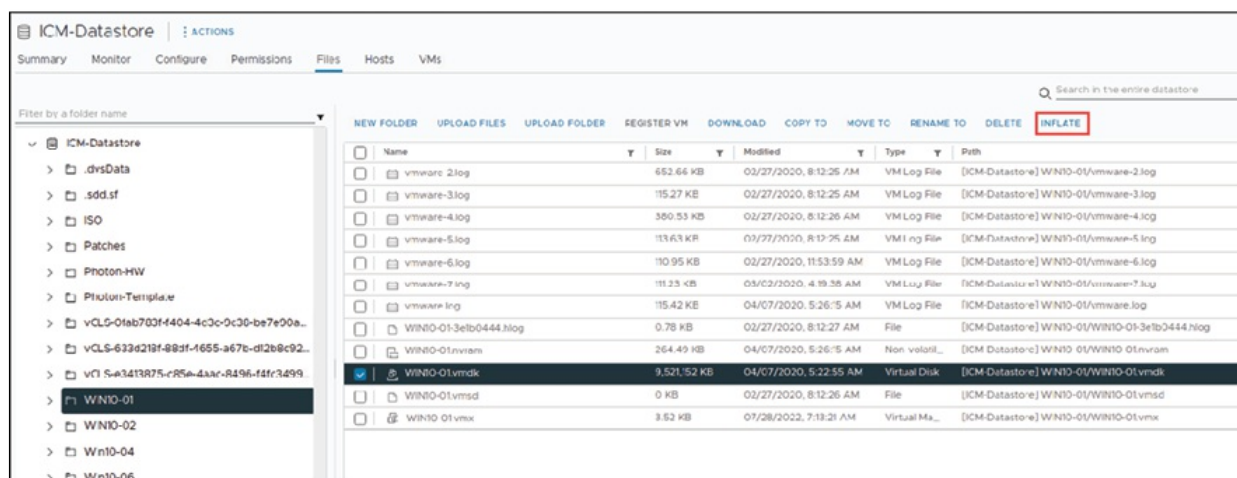


Figure 7.20: Inflating thin provisioned disks

(Source: VMware)

General settings in VM Options

The VM Options tab allows administrators to modify critical information, such as the VM display name and guest operating system:

- Administrators can see the location of the configuration file (**.vmx**) and the VM directory path under **General Options**.
- Administrators may change the display name and guest OS type, but this will not rename the VM files or folders that are underneath.
- Administrators can copy the configuration file path and working location to save for notes or reference.

VM can be renamed, but its associated files and folder names remain the same. This remains consistent with the original name assigned when it was created.

The following figure illustrates the general settings in VM Options:



Figure 7.21: VM Options, general settings

(Source: VMware)

VMware tools settings in VM Options

VMware Tools configuration lets administrators personalize power button action and automate VMware Tools update:

- Customizing the power button needs the VM to be powered off to install changes.
- Activation of *check and upgrade VMware Tools before each power on* ensures that the VM will automatically update VMware Tools in case a new version is out.
- Choosing *synchronize guest time with host* maintains the VM clock synchronized with the ESXi host, avoiding time drift.

The following figure illustrates the VMware tool settings in VM Options:

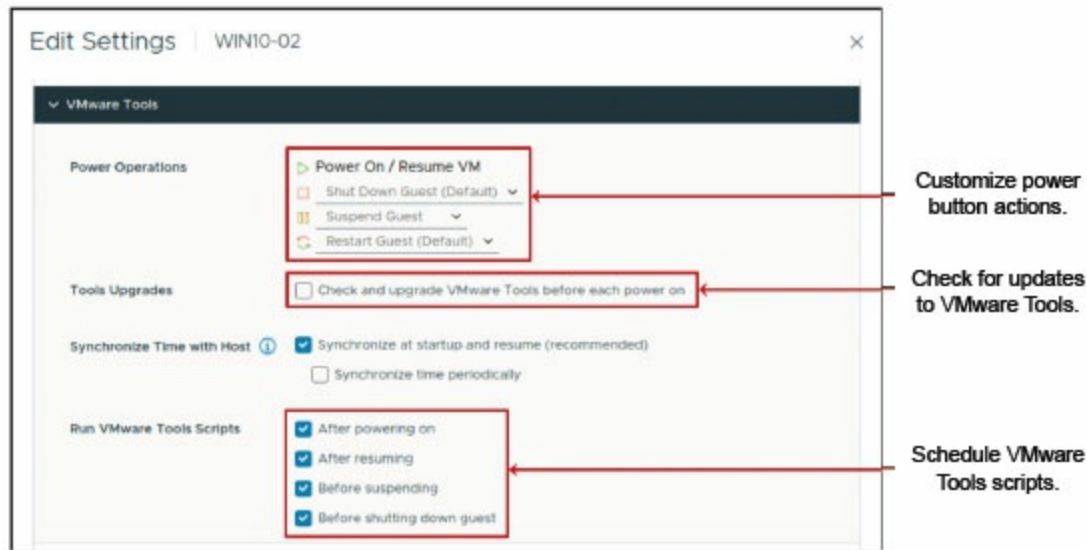


Figure 7.22: VM Options: VMware tools settings

(Source: VMware)

These configurations further automate, improve VM performance, and support newer VMware functionality.

For Timekeeping best practices for Linux guests, refer to the VMware knowledge base articles at <https://knowledge.broadcom.com/external/article?legacyId=1006427>.

Boot settings in VM Options

VM boot settings offer management over the way in which a virtual machine boots up, enabling modification for firmware type, boot delay, and recovery options. The boot options that can be changed are:

- **Firmware selection:** During VM creation, it chooses BIOS or EFI automatically, depending on the guest OS. Both may be supported, but you will need to alter this prior to installing the OS.
- **UEFI secure boot:** Prevents any non-manufacturer-approved software from booting, and all parts, bootloader, OS kernel, and drivers, must be signed.
- **Boot delay:** Inserts a delay prior to the start of the OS, which is handy when starting VMs in sequence or resolving boot problems.
- **Force BIOS/EFI setup:** Allows users to access firmware configuration directly (e.g., boot off a CD/DVD) without having to manually press

keystrokes during boot.

- **Failed boot recovery:** If a VM cannot locate a boot device, it will automatically retry after 10 seconds (default).

These options increase flexibility in VM deployment, security, and troubleshooting.

The following figure illustrates the VM boot settings in VM Options:

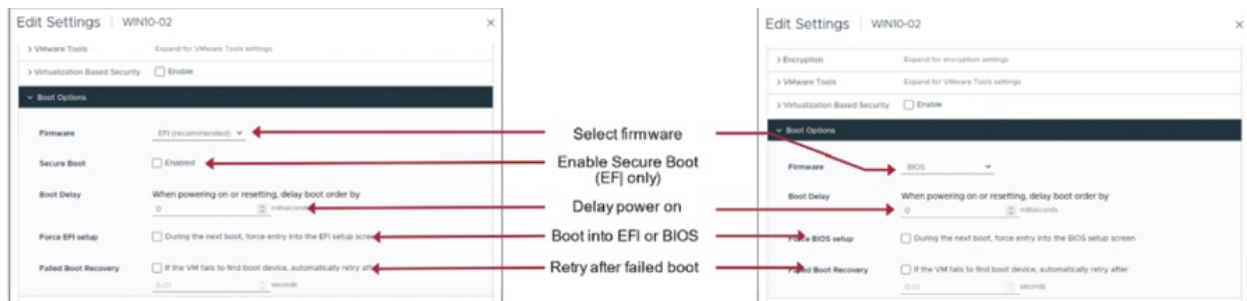


Figure 7.23: VM Option VM boot settings

(Source: VMware)

Harnessing the efficiency of templates

A VM template is a preconfigured virtual machine image that simplifies VM deployment by offering a repeatable and consistent configuration.

A template contains the following:

- A guest operating system with preinstalled software.
- A preconfigured virtual hardware setup, guaranteeing compatibility.
- VMware Tools enhancing VM performance and integration.

The major advantages are as follows:

- **Quick deployment:** Avoids incessant setup procedures.
- **Consistent configurations:** Keeps all VMs having identical settings.
- **Optimized resource use:** Saves storage and administrative overhead.

VM templates are stored together with VMs within the vSphere inventory, where administrators can reconfigure VMs into templates without copying files. When a new VM is provisioned from a template, the new VM is stored in the designated folder, keeping the environment structured and well-

organized.

The following figure illustrates the VM templates:

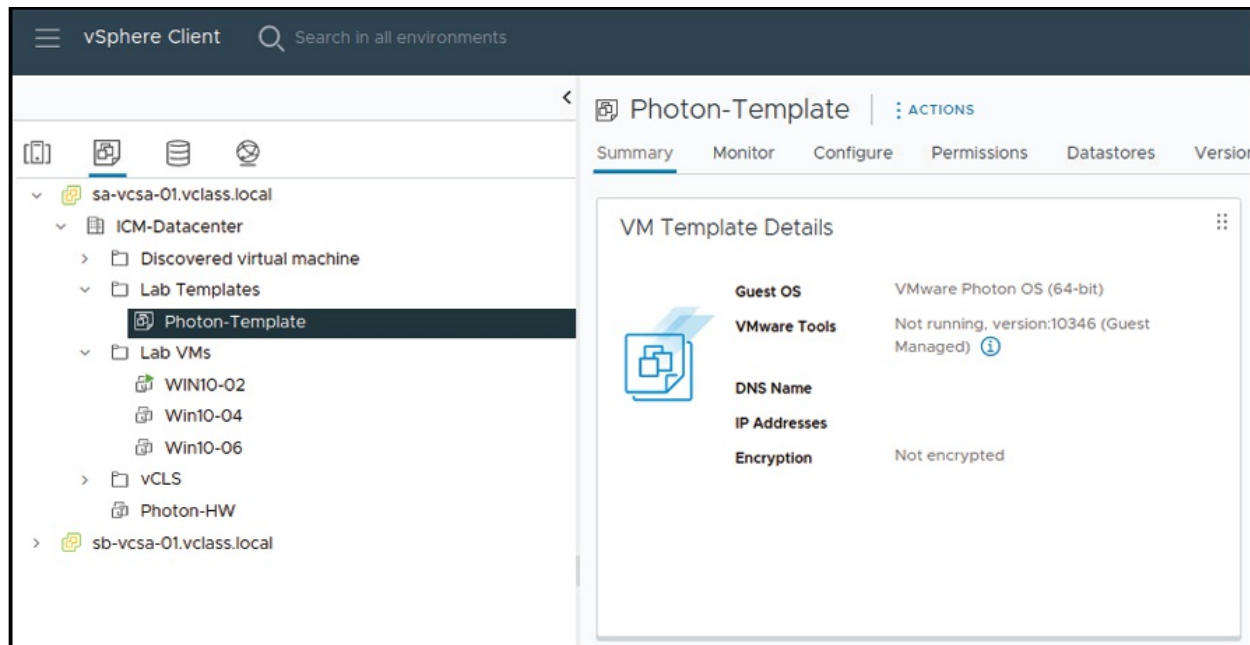


Figure 7.24: About templates

(Source: VMware)

Creating a template by cloning a VM

An effective method of creating a VM template is cloning an existing VM. Cloning a VM takes the existing state of a VM, its configuration, guest OS, and applications, and saves them for future use.

The key considerations are as follows:

- The source VM may be turned on or off while it is being cloned.
- Storage format must be defined for the virtual disk of the template.

The available virtual disk formats are as follows:

- **Same format as source:** Retains the native disk format.
- **Thin provisioned:** Reserves only the needed space and expands on demand.
- **Thick provisioned (Lazy-Zeroed):** Allocates ahead but zeroes out blocks on the first write.
- **Thick provisioned (Eager-Zeroed):** Allocates and zeroes all blocks

ahead, enhancing security and performance for secure workloads.

As shown in [Figure 7.25](#), using cloning, administrators can deploy standardized VMs quickly, creating consistency within the virtual infrastructure:

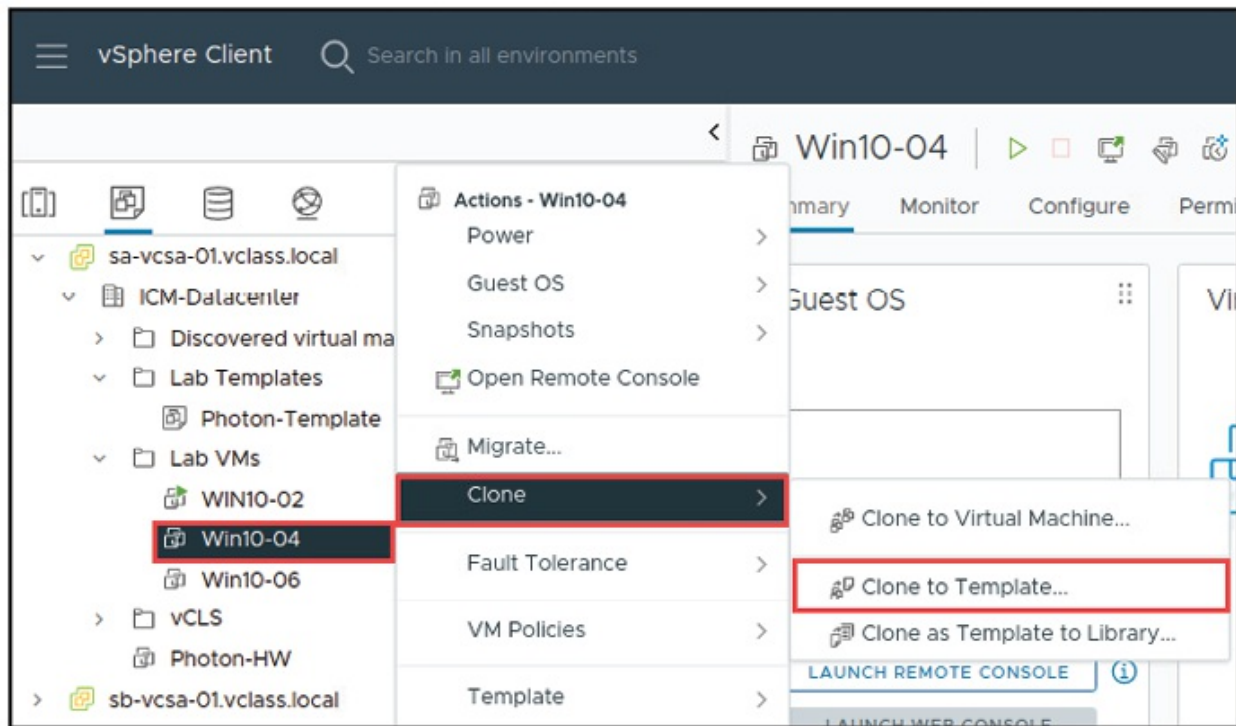


Figure 7.25: Creating a template

(Source: VMware)

Creating a template by converting a VM

Another method to create a **VM template** is by converting an existing VM. Unlike cloning, this process *permanently transforms* the VM into a template, making it *unusable as a regular VM* unless converted back.

The key considerations are as follows:

- The VM **must be powered off** before conversion.
- No option to change the **storage format**—the virtual disk remains unchanged.
- The **VM configuration file (.vmx)** is replaced with a **template configuration file (.vmtx)**.

This method is ideal for **long-term template storage** when you no longer

need the VM in an active state. If future modifications are required, the template must be **converted back into a VM** before making changes.

The following figure illustrates converting a VM to a template:

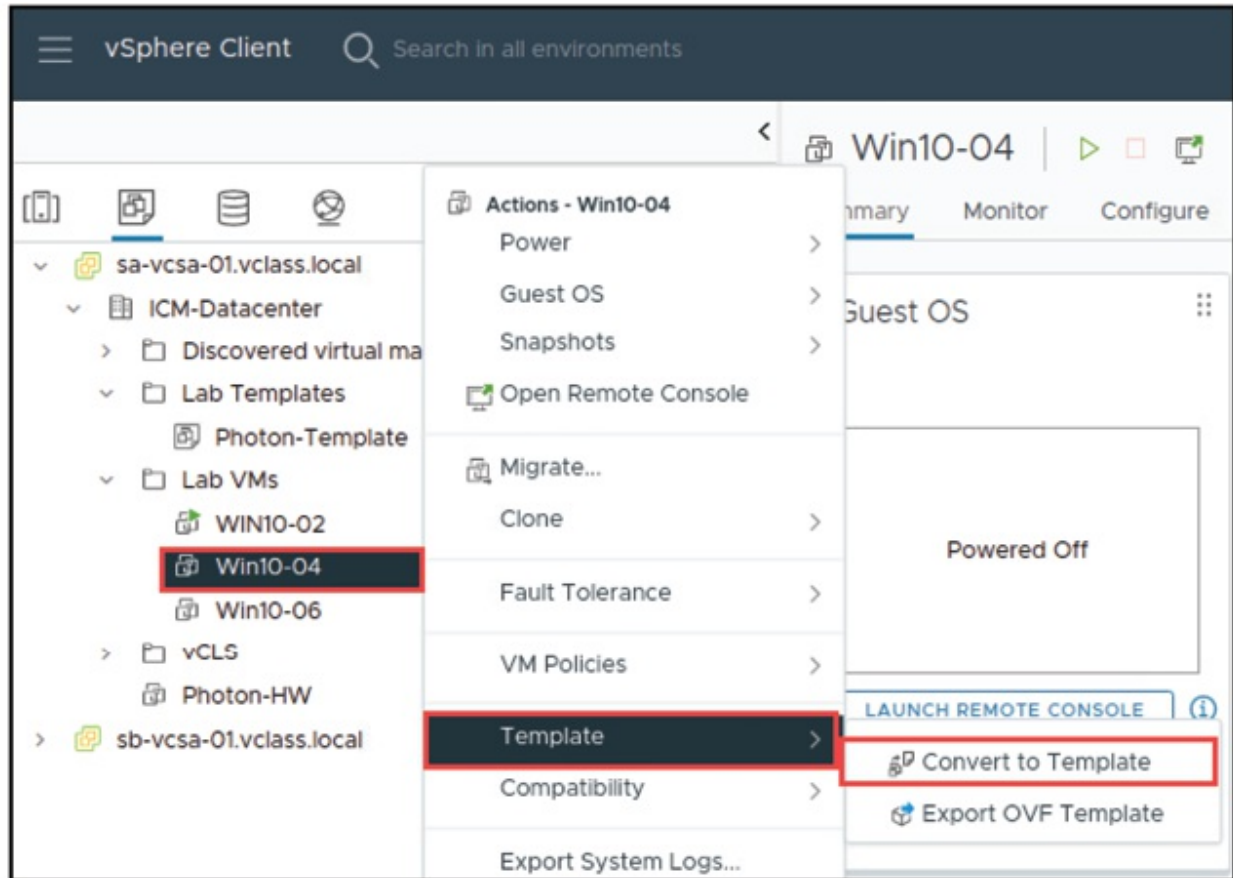


Figure 7.26: Convert VM to a template

(Source: VMware)

The following figure illustrates how to clone a template:

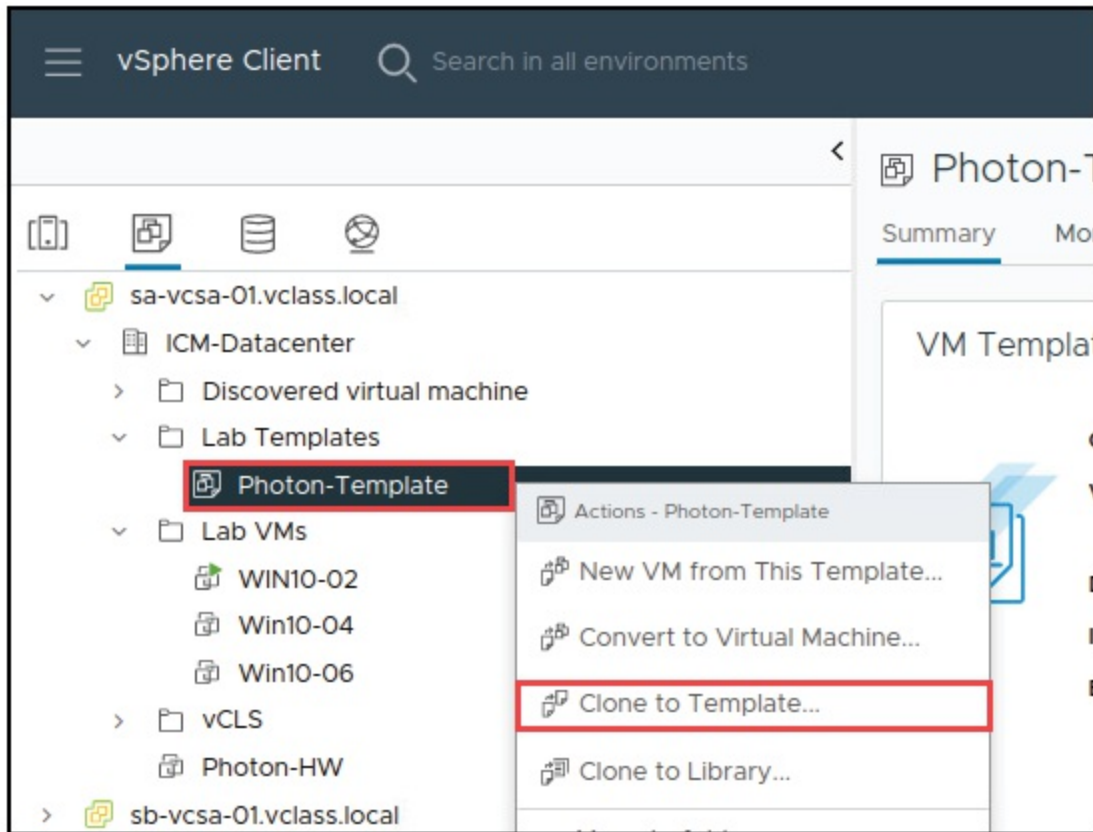


Figure 7.27: Clone a template

(Source: VMware)

Updating templates

Updating VM templates ensures that the newly provisioned VMs will be equipped with the latest patches, software, and hardware configurations. Rather than building a new template from the ground up, administrators can update an existing one by doing the following:

- **Convert the template to a VM:** This temporarily deactivates template-based deployments.
- **Isolate the VM:** Disable user access by disconnecting it from the network or putting it in an isolated environment.
- **Apply updates:** Apply software patches, upgrade VMware Tools, update VM hardware versions, or change virtual hardware.
- **Convert back to a template:** After updates are finished, shut down the VM and convert it back to a template.

The following figure illustrates converting a template to VM for updating:

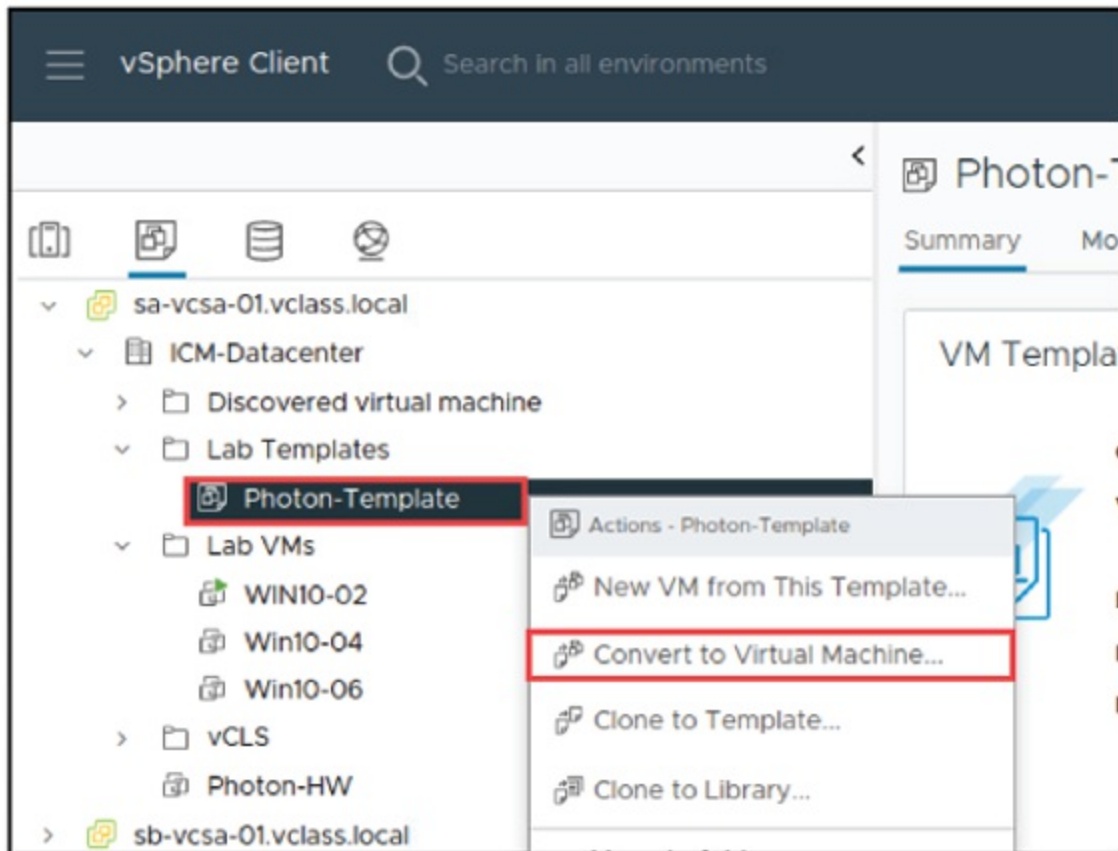


Figure 7.28: Updating a template

(Source: VMware)

Deploying VMs from a template

Whenever a VM is deployed from a template, administrators must enter important information, such as:

- **VM name and inventory location:** Provide a distinctive name and select where the VM will be stored in the vSphere inventory.
- **Compute resource:** Choose an ESXi host, cluster, or resource pool on which to execute the VM.
- **Datastore:** Select a location for the VM files based on performance, capacity, and availability needs.
- **Guest OS customization:** Deploy custom configuration including computer name, network setup, and joining to a domain.

Deploying VMs from templates reduces provisioning by automating, thereby enforcing consistency, and lessening manual configurations.

The following figure illustrates the VM deployment from a template:

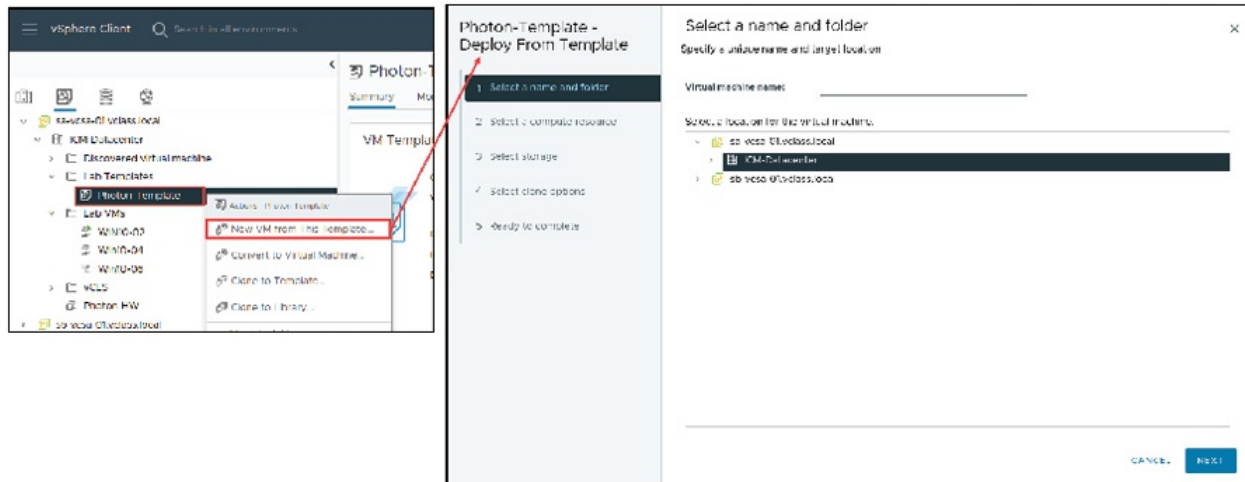


Figure 7.29: Deploying VMs from a template

(Source: VMware)

Cloning virtual machines

Cloning a VM duplicates the source VM exactly, maintaining its configuration, installed software, and data. This is a different approach to deploying a VM from a template and supports cloning when the VM is running or shut down.

Some important points to keep in mind for cloning are as follows:

- **vCenter requirement:** Cloning is supported through vCenter Server and not the VMware Host Client.
- **Powered-on cloning:** When cloning an already running VM, its applications and services are not inherently quiesced, which can create inconsistencies.
- **Template vs. clone:** Templates offer a standardized base image, providing consistency when deploying several VMs, while cloning snapshots the VM in its current state at the time it was created.
- **Storage planning:** VM templates use storage space, thus, disk space

usage needs to be taken into account before creating several clones.

- **Efficiency:** To launch from a template takes less time compared to cloning an already powered-up VM, especially if provisioning numerous VMs.

Cloning proves useful if you want an identical copy of an already provisioned VM without recreating it using the complete provisioning process.

The following figure illustrates the VM cloning:

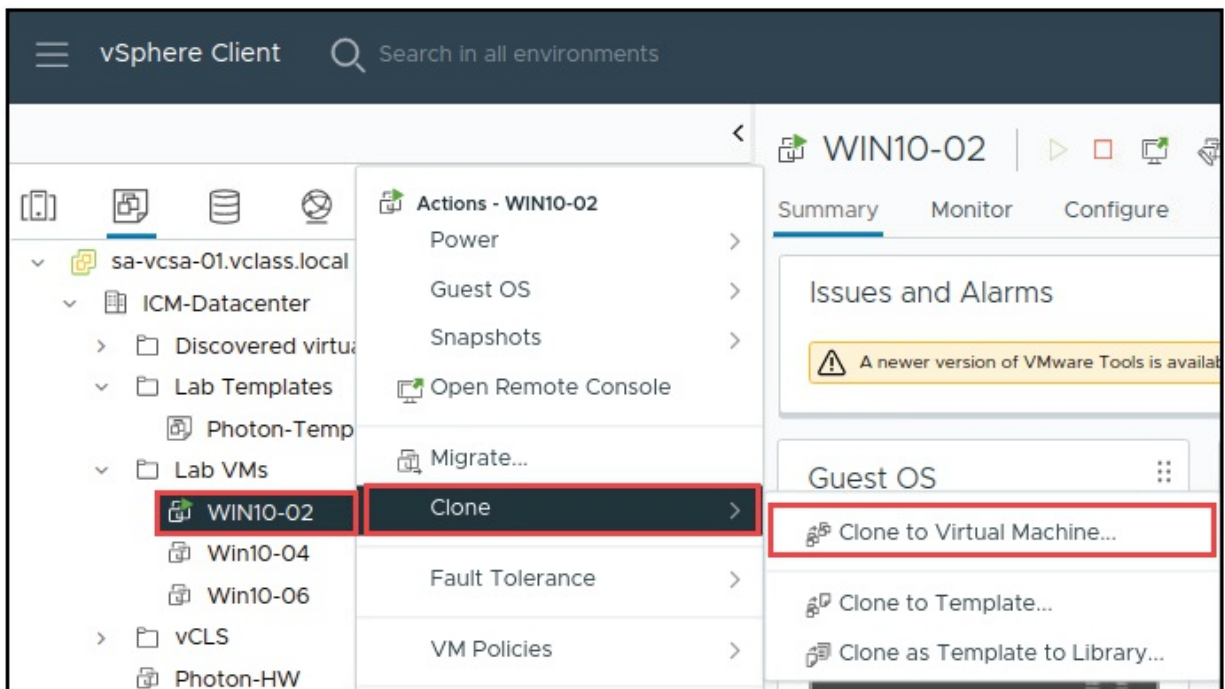


Figure 7.30: Cloning virtual machines

(Source: VMware)

Guest operating system customization

When deploying a VM from a template or clone, customization make sure that each VM has unique system settings, which prevents conflicts caused by duplicate configurations.

The customizable guest OS settings are as follows:

- **Computer name:** Assign a unique hostname to avoid network conflicts.
- **Network settings:** Configure IP addresses, DNS settings, and network adapters.

- **License settings:** Define software activation and product keys for Windows VMs.
- **Time zone:** Set the appropriate time zone for consistency across deployments.
- **Administrator/root password:** Define new credentials for security purposes.
- **Windows Security Identifier (SID):** Generate a unique SID to prevent authentication issues.

Without customization, new VMs *retain the hostname, IP address, and SID* of the source VM, leading to potential *networking and security issues*. VMware's customization options simplify deployment by *automating OS configurations*, ensuring that each VM is exclusively configured and consistency is maintained.

Customization specifications

Customization specifications enable administrators to automate and standardize guest operating system settings when deploying VMs. These specifications are saved in the vCenter database, hence can be reused on numerous deployments.

The key features are as follows:

- Supports Windows and Linux guests to automate OS configuration.
- Centrally stored in vCenter to ensure consistency and avoid manual configuration.
- Quick to manage from the Policies and Profiles menu in the vSphere Client.

Using customization specifications, administrators can effectively set hostnames, network, time zones, and **security identifiers (SIDs)** while saving deployment time and possible errors.

The following figure illustrates the VM customization specifications:

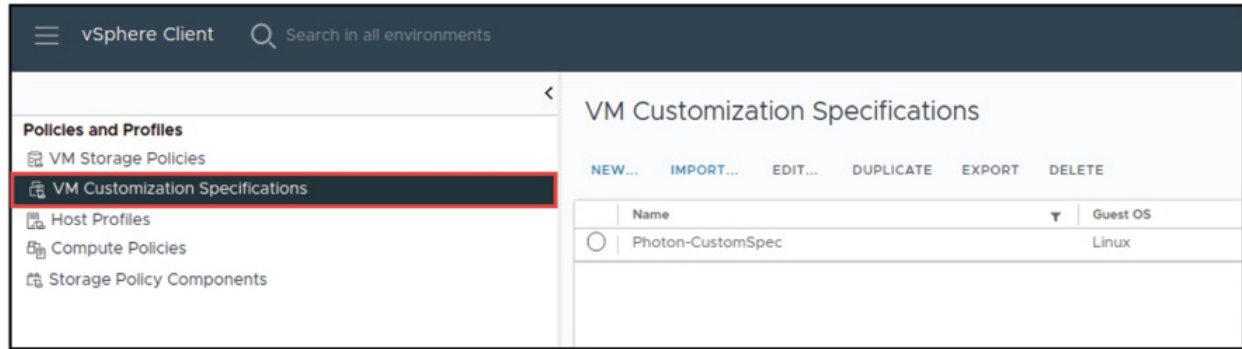


Figure 7.31: About customization specifications

(Source: VMware)

Customizing the guest operating system

Cloning a VM or deploying from a template involves a customization specification that ensures every VM has its own system settings, avoiding conflicts like duplicate hostnames and IP addresses. To use modification, first need to create a specification. When cloning or deploying, administrators can select and apply an existing specification to the new VM.

The key requirements are as follows:

- VMware Tools should be installed on the guest operating system.
- The guest OS disk needs to be attached to SCSI node 0:0 within the VM settings.

Customization specs automate deployment by enabling each VM to have the proper network configurations, host identity, and system settings without any user interaction.

The following figure illustrates the guest OS customization:

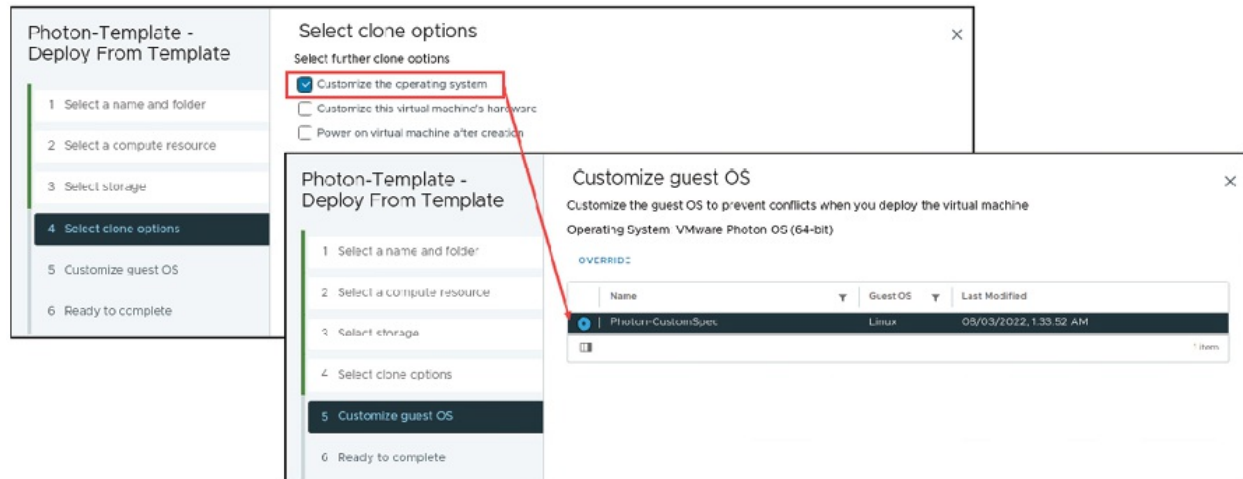


Figure 7.32: Customizing the guest operating system

(Source: VMware)

Content libraries for VM resources

Content libraries provide centralized storage of OVF templates, ISO images, and other content, which makes it possible to share and synchronize them in various vCenter instances worldwide easily.

Multiple data centers in organizations commonly face problems in keeping template and image versions up to date on different vCenter instances. Content libraries alleviate this problem with the provision of a centralized repository, which can be accessed and utilized easily.

The key benefits are as follows:

- **Effective storage management:** Stores and maintains templates, ISO files, and other resources in an organized manner.
- **Version control and synchronization:** Keeps content current by automatically synchronizing with published libraries.
- **Global access and sharing:** Enables vCenters to subscribe to published libraries, making it simple to distribute standardized templates across locations.

By taking advantage of content libraries, organizations simplify VM deployments, are consistent, and minimize administrative costs in managing templates and other fundamental VM resources.

The following figure illustrates the content libraries:



Figure 7.33: About content libraries

(Source: VMware)

Advantages of content libraries

Content libraries improve storage effectiveness, consistency, and simplicity of management in virtual environments. Administrators can store, share, and synchronize important resources such as templates, ISO images, and scripts between multiple vCenter instances.

The primary advantages are as follows:

- **Centralized storage and sharing:** Store all templates, scripts, and ISO images in a single location for quick access.
- **Distributed file management:** Sync content libraries between multiple sites and vCenter instances.
- **Seamless VM deployment:** Directly mount ISO files from the content library for rapid provisioning.
- **Version control:** Keep multiple versions of VM templates with the ability to roll back to earlier versions.
- **Automatic updates and synchronization:** If a published content library is updated, all subscribed libraries will automatically synchronize with the new version.

With vSphere 7 and above, VM templates can be updated even while being utilized for deployments. The system maintains the old and new versions, allowing for a simple rollback if necessary. Content libraries guarantee uniform deployments, minimum manual effort, and consistent availability of resources in distributed environments.

Types of content libraries

Content libraries are divided into three types depending on how content is managed and shared among vCenter environments:

- **Local content library:**
 - An administrator manages a vCenter instance.
 - Templates, ISOs, and other files are safely stored and can be used locally.
 - Unable to share with other vCenter instances unless changed to a published library.
- **Published content library:**
 - Subscriptions allow you to share your local library with other vCenter instances.
 - Traces version changes but does not save previous versions.
 - Administrators may edit, add, or delete content.
- **Subscribed content library:**
 - Synchronizes with a published content library to get automatic updates.
 - Users have read-only access; administrators cannot directly change content.
 - May be configured for immediate or on-demand synchronization with the publisher.
 - Cannot be converted into a published library.

A local content library is best for isolated environments, whereas published and subscribed libraries ensure centralized management and consistency across multiple vCenter instances.

Interface of content library

To create and manage content libraries, navigate to the **vSphere Client**, and select **Content Libraries** from the main menu. This interface allows administrators to do the following:

- Create local, published, and subscribed content libraries.
- Upload, modify, and organize templates, ISOs, and other files.
- Configure synchronization settings for subscribed libraries.
- Track version history for VM templates (starting from vSphere 7).

The **Content Libraries** section provides a centralized view of all available libraries, making it easy to manage and distribute virtual machine resources efficiently across vCenter environments.

The following figure illustrates the content library interface:

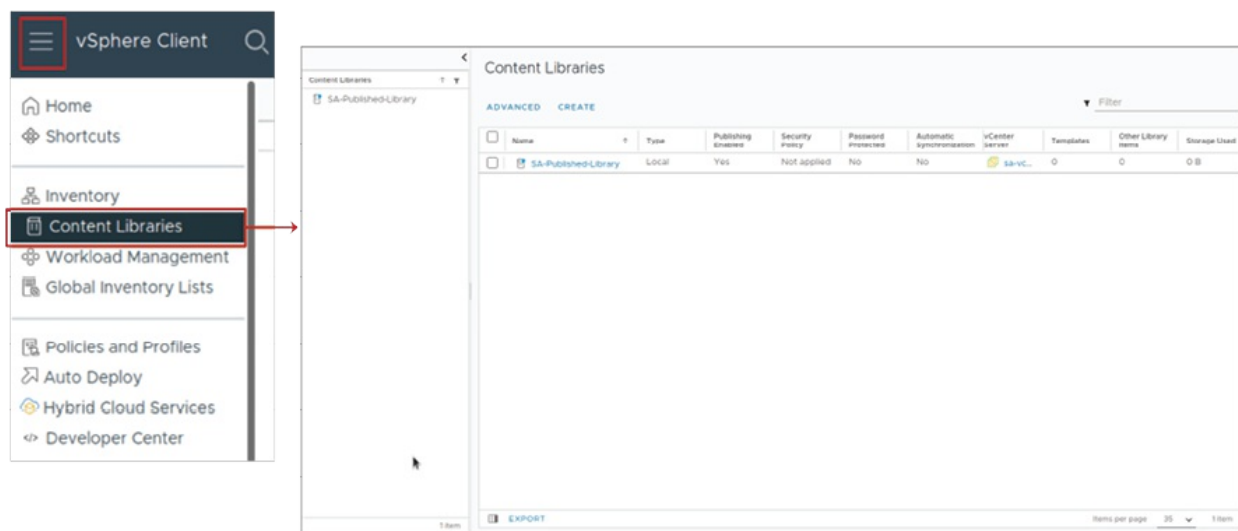


Figure 7.34: Content library interface

(Source: VMware)

Creating a local content library

To create a **local content library**, follow these steps:

1. Open **vSphere Client** | Navigate to the **Content Libraries** section from the main menu.
2. Select **Create Library** | Click **New Content Library** and provide a name.

3. Choose **Library Type** | Select **Local Content Library**, which allows direct management of stored items.
4. Choose a **datastore or remote storage** location to store the content.
5. Review the configuration and click **Finish** to create the library.

This is ideal for maintaining VM templates, ISO images, and scripts within a vCenter instance, ensuring consistency in virtual machine deployments.

The following figure illustrates how to create a local content library:

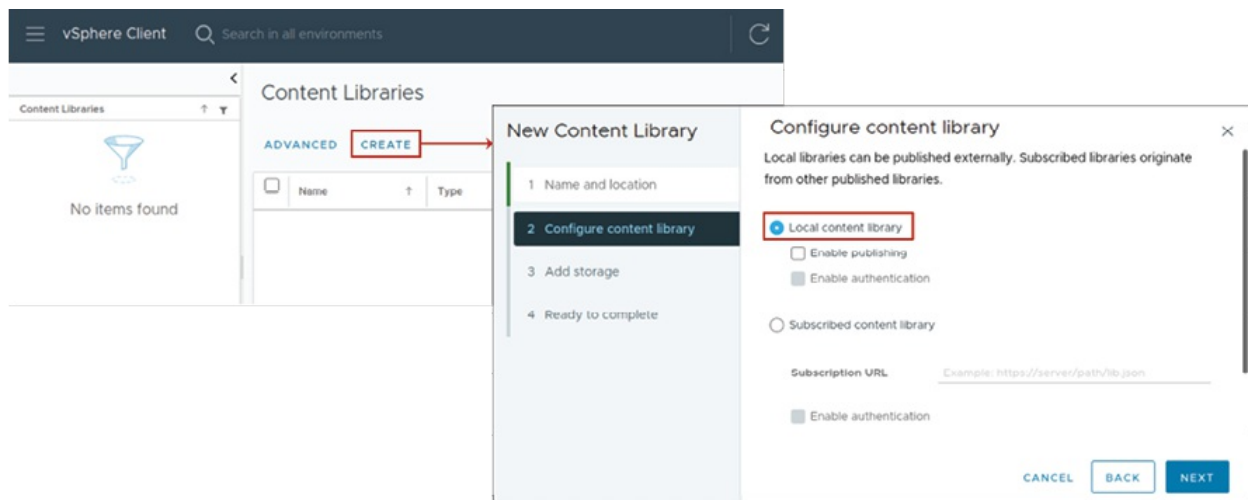


Figure 7.35: Creating a local content library

(Source: VMware)

Filling the content library with templates

The content library can store VM Templates and OVF Templates, both for different purposes as mentioned here:

- **VM templates** are saved in the datastore's default format (e.g., thick-provisioned eager-zeroed), can be on any datastore other than NFS, and show up in the vCenter inventory. They are linked to a particular host.
- **OVF templates** are never stored in thick-provisioned format, are not associated with a particular host, and do not show up in the vCenter inventory. They must be on a datastore associated with the content library.

When a VM Template is stored in the content library, it is still connected to the vCenter inventory. Any changes, like renaming, deleting, or reverting it

back to a VM, impact both the inventory and the library item.

Adding VM or OVF templates into content library

When administrators clone a virtual machine into a template within a content library, the available option to deploy is either using VM templates or OVF templates.

Figure 7.36 displays cloning the Photon-01 VM as a VM template and stores it within the content library while retaining its link with the vCenter inventory. This enables you to easily deploy and manage virtual machines directly from the library.

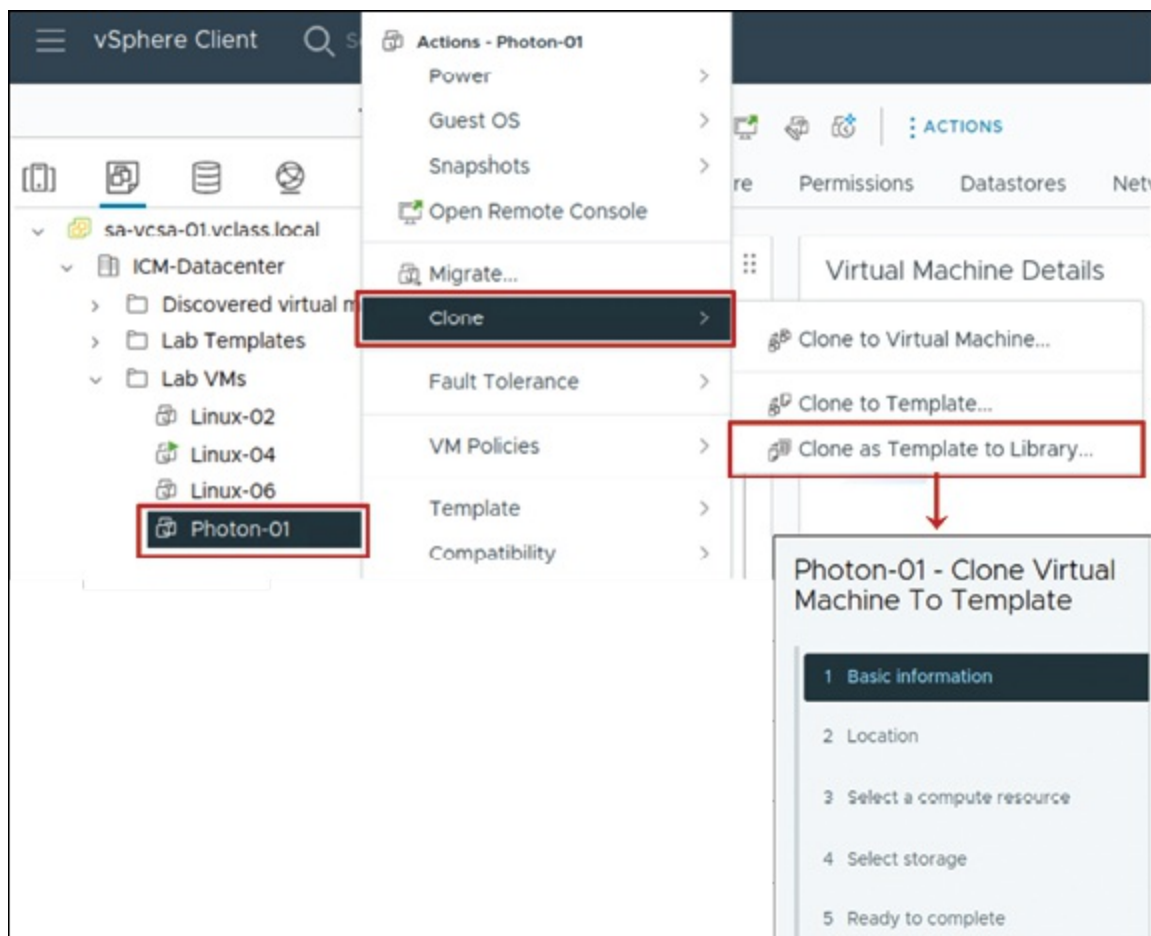


Figure 7.36: Adding VM templates to content library

(Source: VMware)

Adding OVF templates into a content library

When cloning a template from the vCenter inventory to a content library, it is stored as an **OVF template**. Unlike VM templates, OVF templates are **not associated with a host** and do not appear in the vCenter inventory. This format allows for efficient storage, easy sharing, and deployment across multiple environments.

The following figure illustrates the adding OVF templates into a content library:

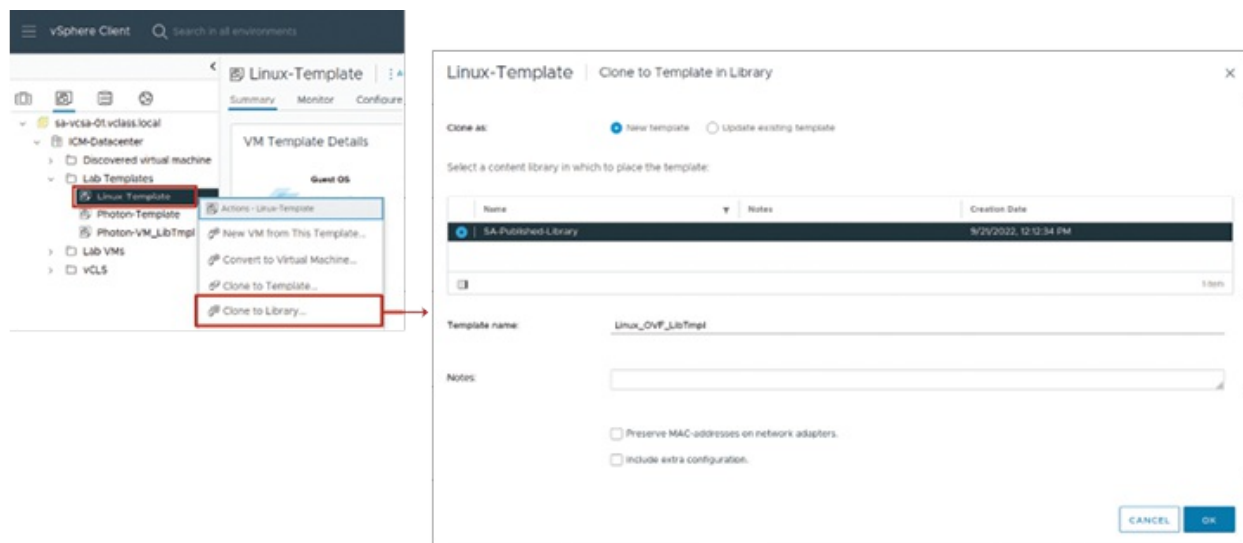


Figure 7.37: Adding OVF templates to the content library

(Source: VMware)

Viewing content library items

The content library structures items into templates (VM templates and OVF templates) and other types (like ISO images and scripts).

ISO files can be mounted by administrators directly from the content library to install software or guest operating systems. ISO files are only available to VMs that are registered on ESXi hosts with access to the datastore of the content library.

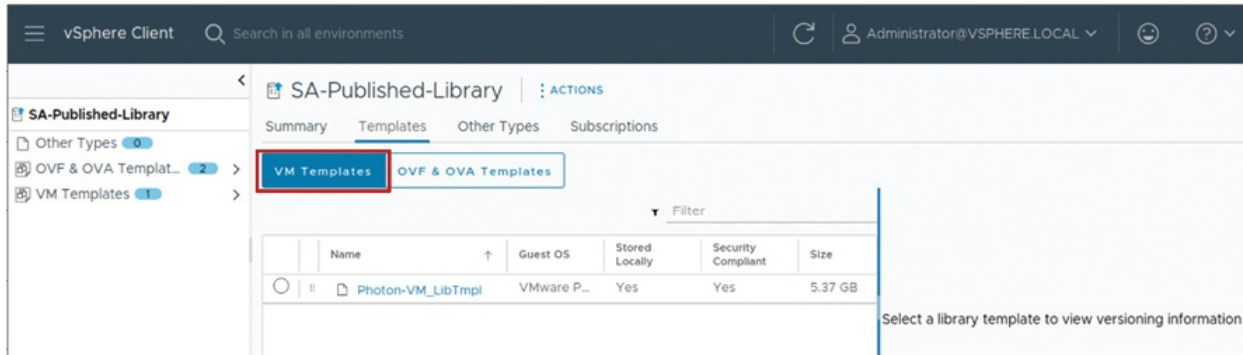


Figure 7.38: Viewing content library templates

(Source: VMware)

Deploying VMs from a content library

Administrators can provision virtual machines from templates within a content library via the New Virtual Machine wizard, as shown in [Figure 7.39](#). When provisioning, the Select a template page offers two choices:

- **Content library tab:** Displays OVF templates within the library.
- **Data center tab:** Shows VM templates, such as those that are added to the content library.

By using the content library, admins can manage VM deployments effectively with consistency across multiple vCenter environments.

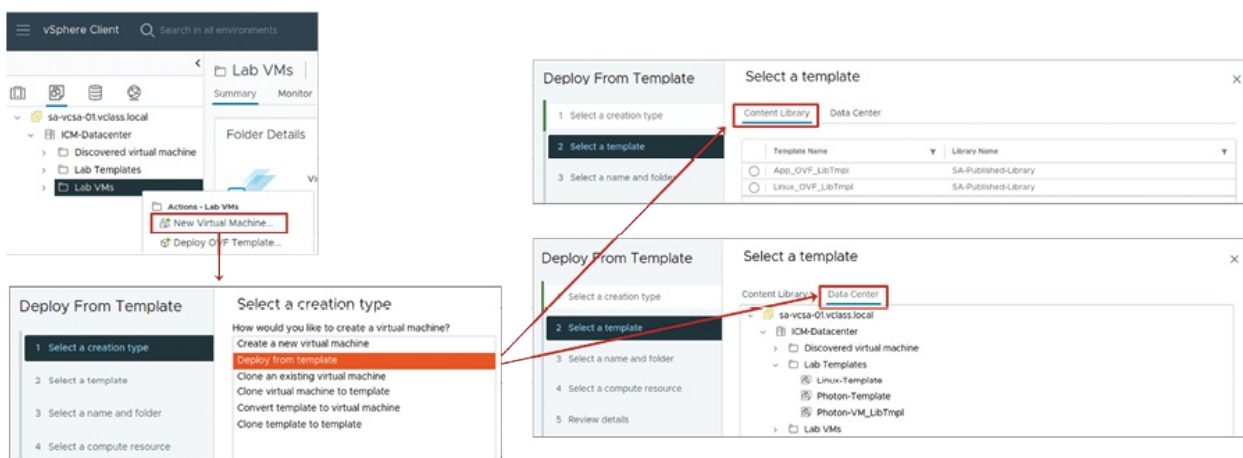


Figure 7.39: Deploying VMs from a content library

(Source: VMware)

Content library integration

The types of content libraries are as follows:

- **Local content library:** A vCenter-specific repository for storing and managing VM templates, ISO images, and other content, requiring manual updates.
- **Published content library:** Extends a local library by allowing other vCenter instances to subscribe and access its content, with optional authentication for controlled access.
- **Subscribed content library:** Syncs with a published library, storing either full copies or just metadata. Updates must be made in the source library, ensuring consistency across vCenter environments.

Content libraries enable version control, streamline VM deployments, and maintain uniformity across distributed virtual infrastructures.

The following figure illustrates the types of content libraries:



Figure 7.40: Types of content libraries

(Source: VMware)

Publishing a content library

To distribute VM templates, ISO images, and other content to various vCenter instances, administrators can publish a local content library by enabling publishing settings. When publishing is enabled, a subscription URL is generated, which allows other vCenter instances to subscribe and sync the content.

For added security, password protection can be enabled, which provides

access to published content only to authenticated subscribers though its optional. This aspect assists in access control while ensuring consistency and efficiency with multiple environments.

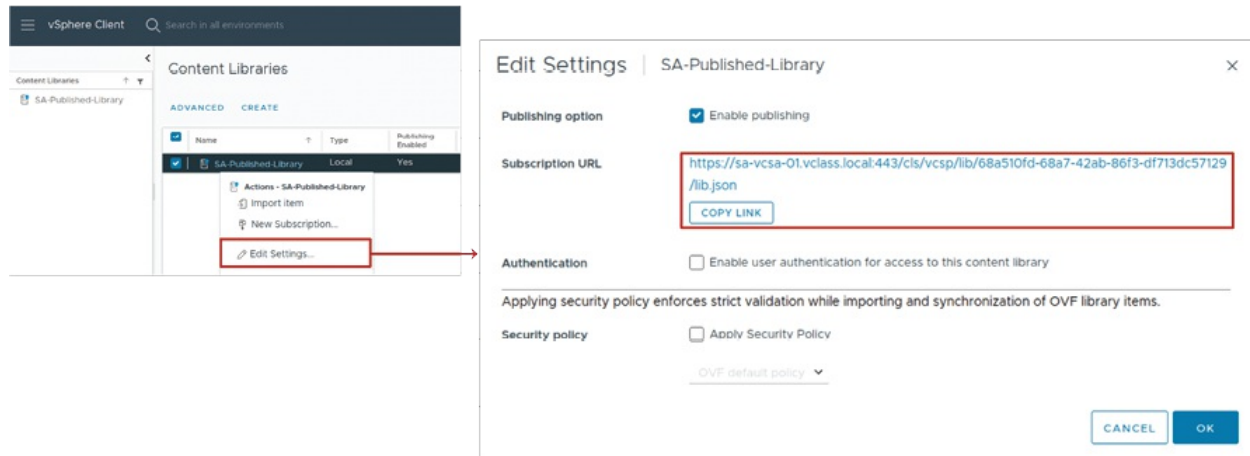


Figure 7.41: Publishing a content library

(Source: VMware)

Subscribing to a content library

To use common templates and files from a published content library, administrators must subscribe and provide the subscription URL.

When administrators subscribe, administrator can select how the content is synchronized:

- **Immediate download:** This option downloads all the content to the designated storage location at once, making everything immediately available.
- **On-demand synchronization:** Basic data is stored only initially in this scenario, and the complete content is downloaded the first time you open it, conserving storage space.

This option allows administrators to control storage efficiency and availability based on what their environment requires.

The following figure illustrates how to subscribe to a content library:

New Content Library

- 1 Name and location
- 2 Configure content library
- 3 Add storage
- 4 Ready to complete

Configure content library

Local libraries can be published externally. Subscribed libraries originate from other published libraries.

☐ Local content library

☐ Enable publishing

☐ Enable authentication

☒ Subscribed content library

Subscription URL

☐ Enable authentication

Download content

☐ Immediately ☒ when needed

CANCEL BACK NEXT

Figure 7.42: Subscribing to a content library

(Source: VMware)

Viewing content libraries

The **Content Libraries** pane provides a centralized overview of all local and subscribed content libraries in vSphere infrastructure. It displays valuable information, including:

- **Library type:** Indicates whether the library is local or subscribed.
- **Publishing status:** Displays whether a local library is published for subscription by other instances of vCenter.
- **Synchronization state:** Displays whether a subscribed library is synchronized with its published source.

This overview helps administrators to manage and monitor content distribution efficiently across several instances of vCenter.

The following figure illustrates how to view a content library:

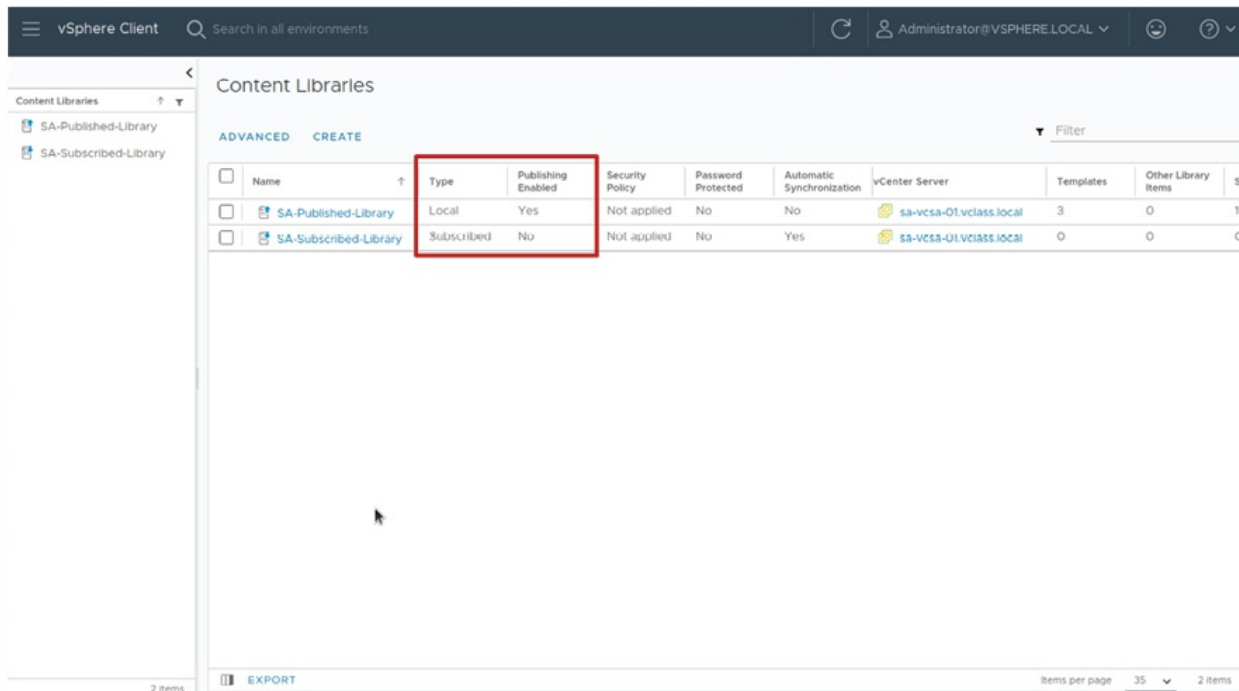


Figure 7.43: Viewing content libraries

(Source: VMware)

Viewing subscribed content library templates

The **OVF & OVA Templates** pane lists OVF templates in a subscribed content library, which can be manually synced via **ACTIONS | Synchronize**.

The **VM Templates** pane displays VM templates, but these sync only when explicitly published by the source library. Unlike OVF templates, VM templates cannot be updated on demand by a subscriber, ensuring controlled distribution across vCenter instances.

The following figure illustrates how to view a subscribed content library template:

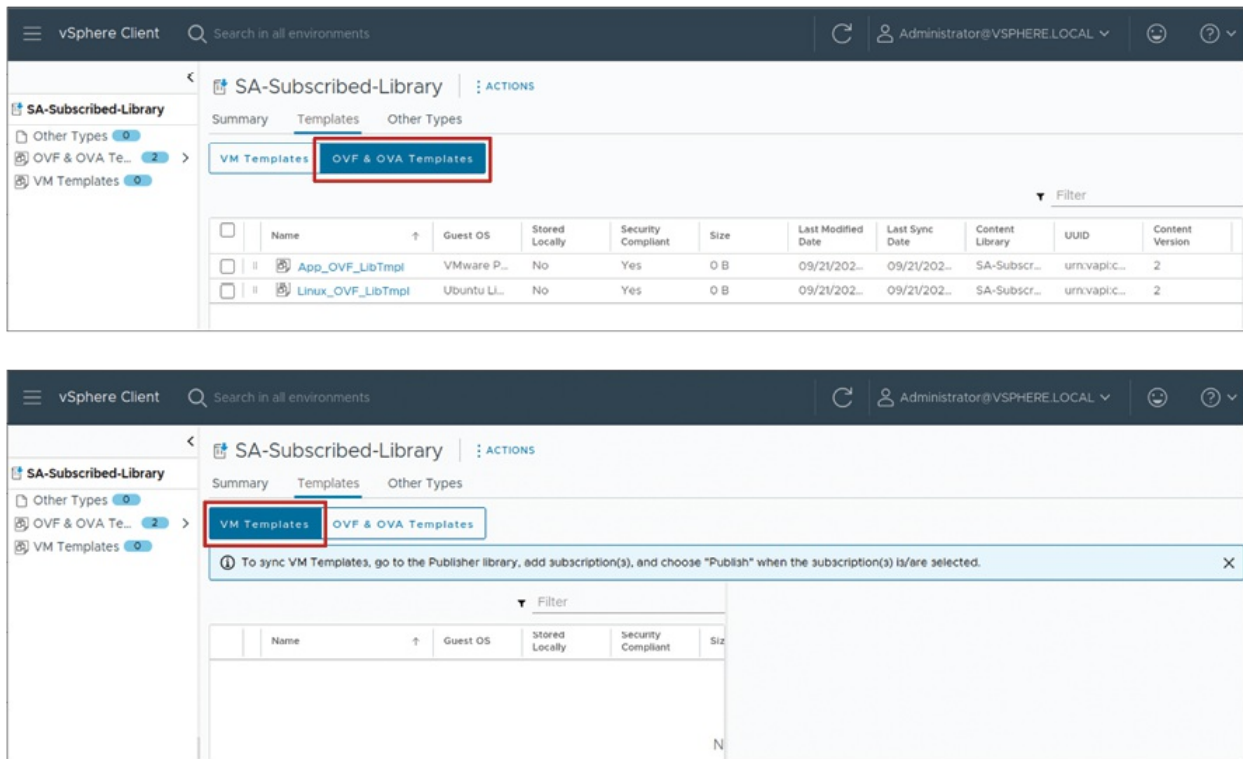


Figure 7.44: Viewing subscribed content libraries templates

(Source: VMware)

Publishing a subscription to a shared VM templates

To publish VM templates to a subscribed content library, administrators first need to create a subscription in the published library by following these steps:

- Go to the published library pane, choose **ACTIONS | New Subscription**, and set up the subscription.
- When the subscription is created, find it in the list and click **PUBLISH** to publish the VM templates to the subscribed library.
- When you add new VM templates in the future, you need to publish them again manually to share them with the subscribed library.

While OVF templates can be synchronized by the subscribers, VM templates need to be published manually. This method allows for more control over when and what content to share between vCenter instances.

The following figure illustrates how to create a subscription:

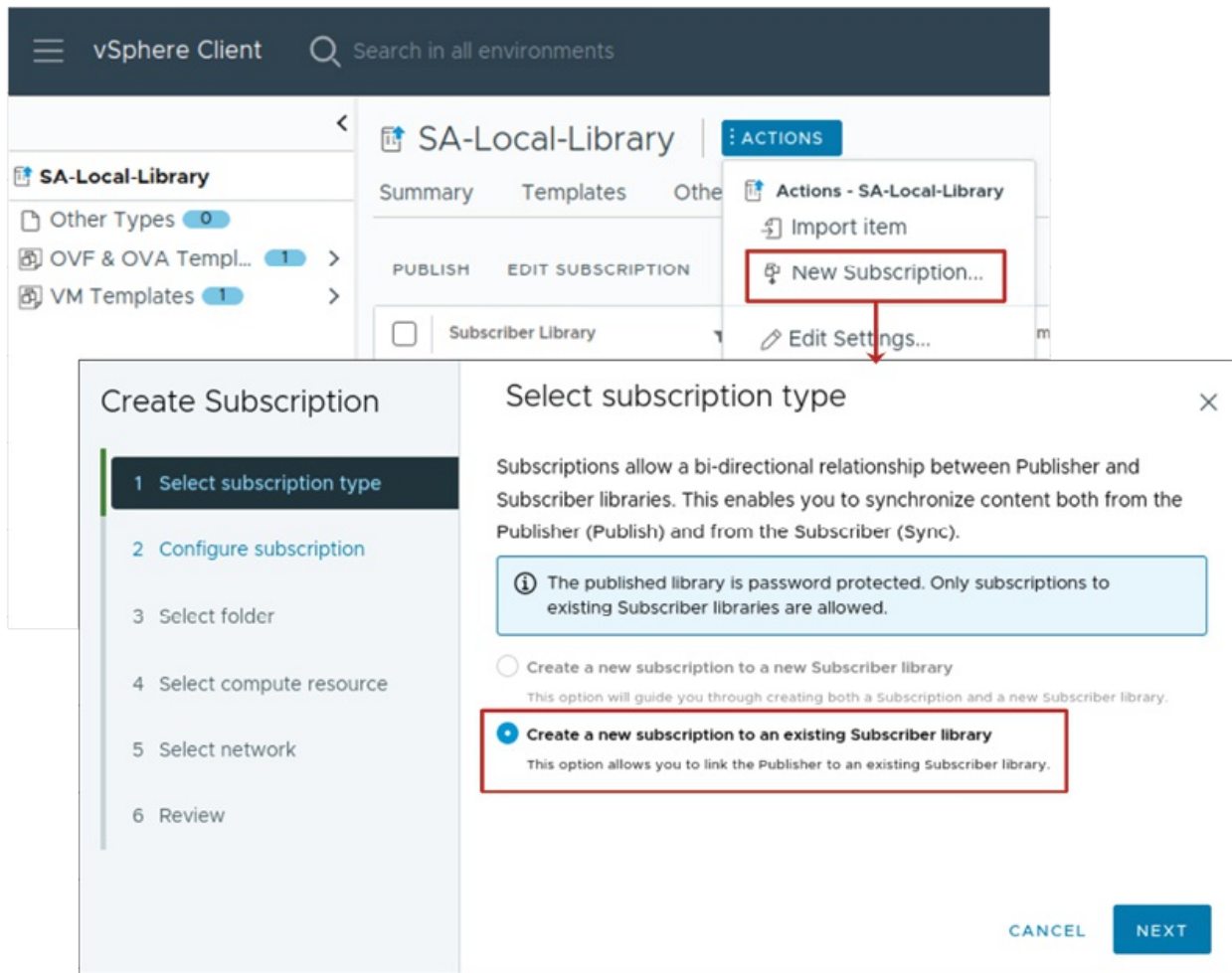


Figure 7.45: Creating a subscription

(Source: VMware)

Synchronizing libraries with or without Enhanced Linked Mode

Synchronization between published and subscribed content libraries is optimized based on the vCenter **Single Sign-On (SSO)** domain and storage configuration as mentioned here:

- **With Enhanced Linked Mode (ELM):** If enabled and the ESXi hosts communicate, the **Network File Copy (NFC)** protocol optimizes data replication. If both libraries exist on the same storage array, **VMware vSphere APIs for Array Integration (VAAI)** further optimizes the transfer.
- **Without Enhanced Linked Mode:** Library Content transfers via

HTTPS through the Transfer Service. Efficiency varies based on storage setup:

- Both published and subscribed libraries reside on a datastore.
- Both libraries are on an NFS file system mounted on vCenter instances.
- Published library on NFS, subscribed library on a datastore.

Simple versioning in content libraries

To track updates, content libraries use simple versioning:

- Each library item has a version number that increments when modified.
- The library itself also has a version number to reflect overall changes.
- During synchronization, only items with version mismatches are updated, reducing unnecessary data transfers.

The following figure illustrates how to synchronize content library:

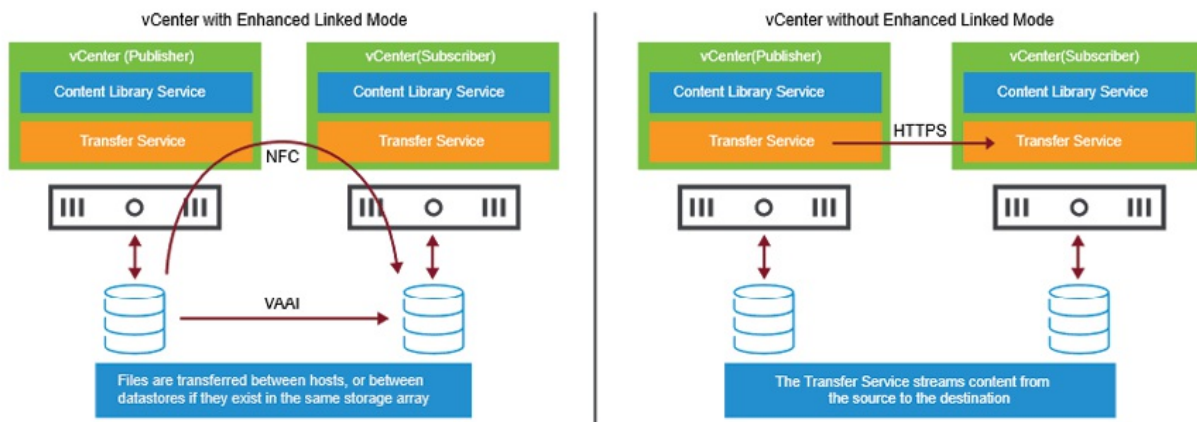


Figure 7.46: Synchronizing content libraries

(Source: VMware)

Advanced configuration

Administrators can use the **Advanced Configuration** page to make content storage and sync enhancements by adjusting storage policies, sync intervals, and transfer settings. Altering the settings makes things work better, reduces wait times, and makes content accessible at all times.

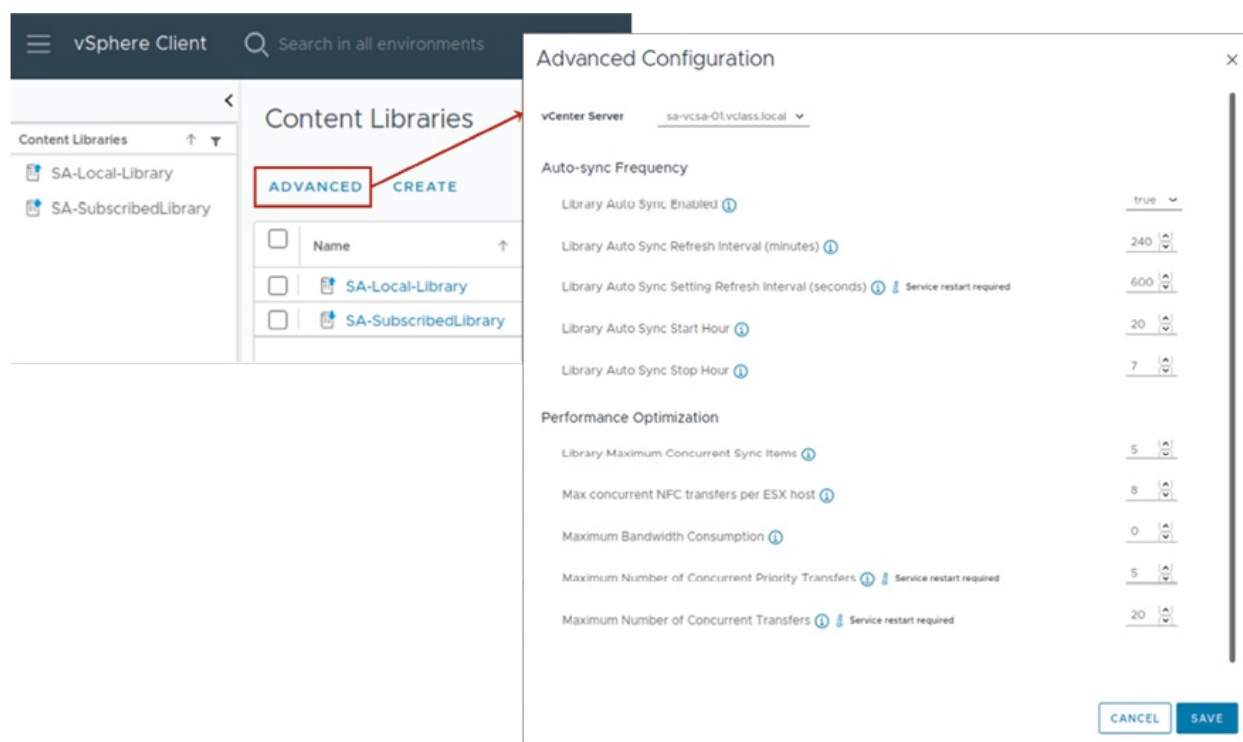


Figure 7.47: Advanced configuration

(Source: VMware)

Content library maximums

The content library has predefined limits to ensure optimal performance and efficient management within a vCenter instance. It can be stored on a datastore or in available vCenter storage, but it must reside within a single file system or datastore.

The following table illustrates the content library maximums:

Configuration items	Maximums
Content library item size	1 TB
Total items per library	1,000
Total library items per vCenter instance (across all libraries)	2,000
Maximum number of concurrent sync operations on the published library's vCenter instance	16
Total number of libraries per vCenter instance	1,000

Table 7.4: Content library maximums

By default, *automatic synchronization* occurs every *240 minutes*, but administrators can modify the frequency in the *advanced configuration settings*.

Managing VM template versions

The administrator can dynamically update templates, deploy patched VMs, and monitor changes through version history. It also maintains current and previous versions to facilitate rapid recovery from compatibility problems.

In [Figure 7.48](#), **App-Lib Template** is managed within the **SA-Local-Library**, allowing for controlled updates and seamless deployments:

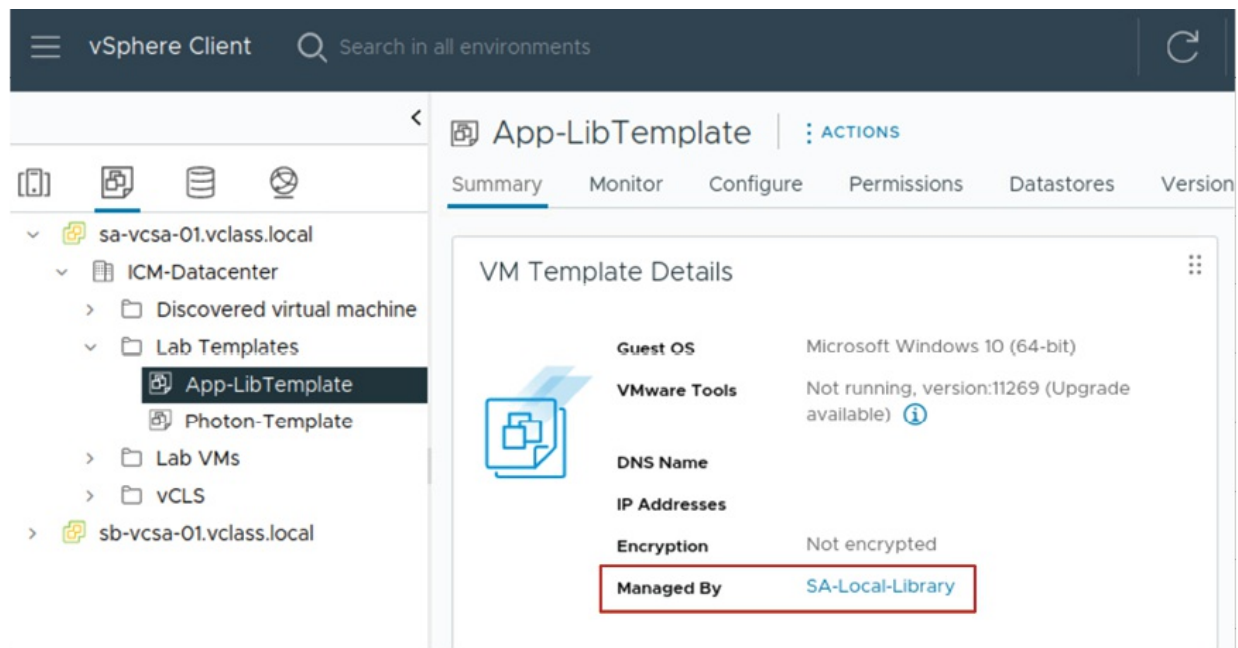


Figure 7.48: Updating VM templates in content library

(Source: VMware)

Template versioning process overview

The template versioning process in a content library allows administrators to update VM templates and maintain version history. The process includes the following:

- **Check out the VM:** A VM is checked out for updates without affecting

ongoing deployments. While checked out, others cannot modify it, and it cannot be converted back or migrated.

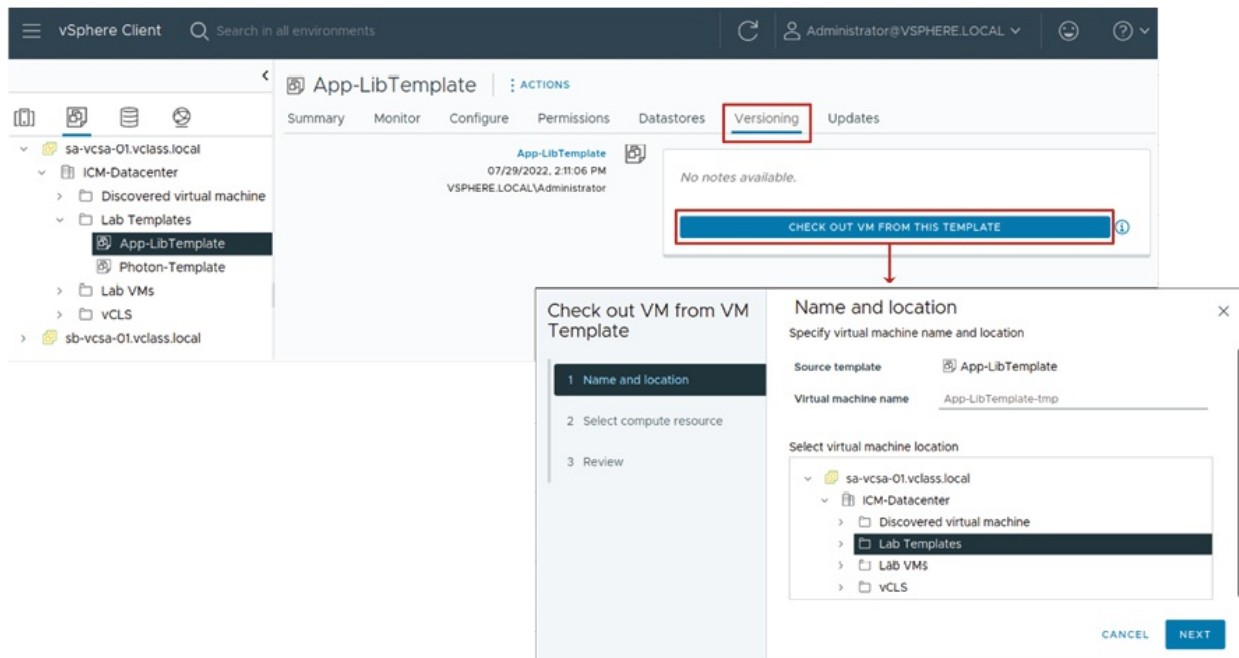


Figure 7.49: Checking out VM from template

(Source: VMware)

- **Apply changes:** Administrators can modify hardware, install updates, or upgrade VMware Tools. vSphere uses linked clone technology to keep the original template intact. Changes can be merged back or discarded if unnecessary.

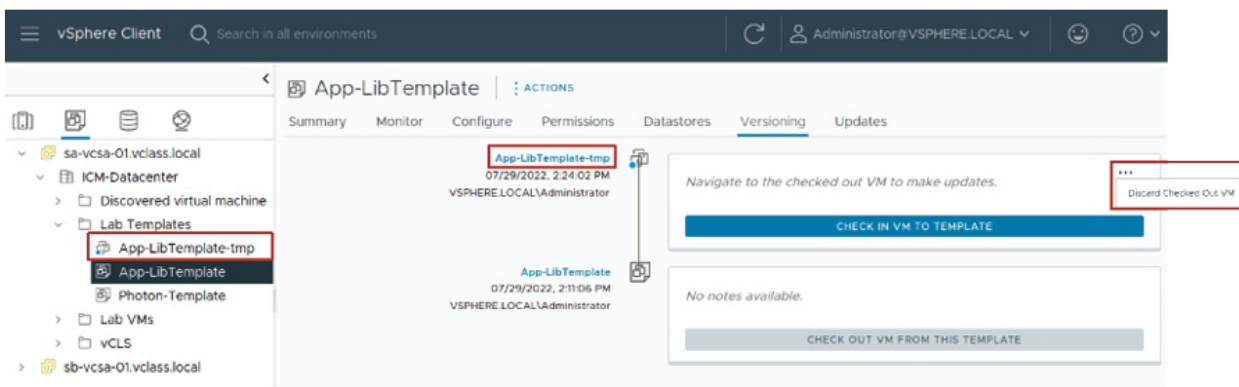


Figure 7.50: Making changes to template

(Source: VMware)

- **Check in the VM:** Once updates are complete, the VM is checked back in, creating a new template version with documented change history.

The following figure illustrates checking in VM to template:

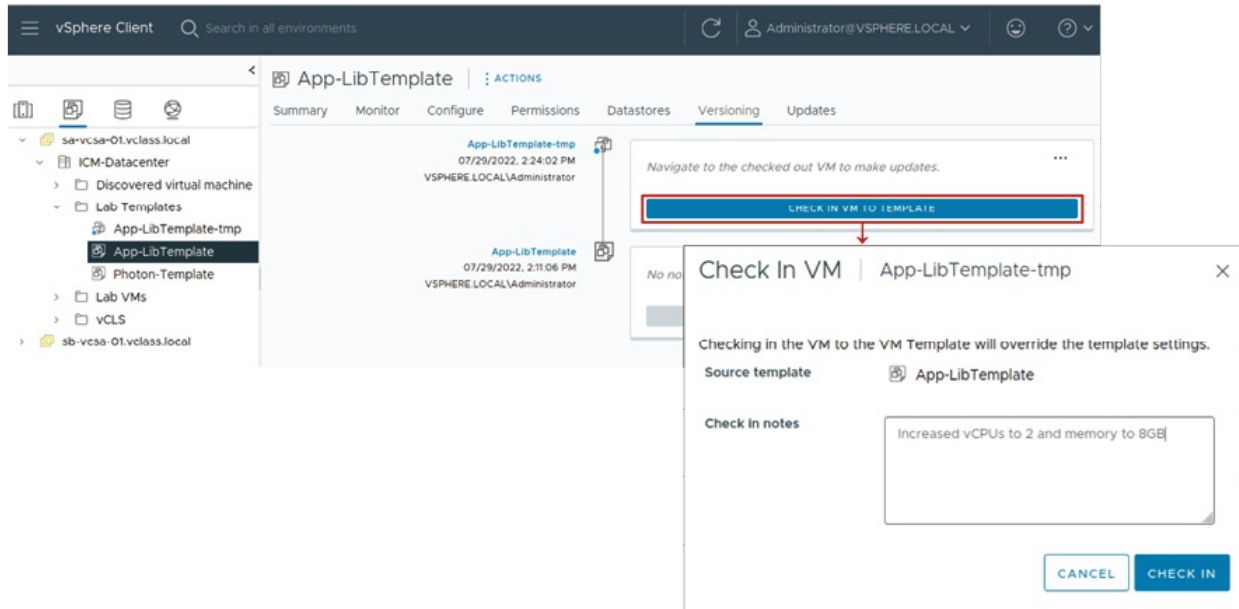


Figure 7.51: Checking in VM to the template

(Source: VMware)

This method ensures controlled updates, seamless deployments, and easy rollback.

Viewing template versions

Every time a VM is checked in, a new version of the template is created and recorded under the **Versioning** tab. The following is the detailed explanation:

- **Tracking updates:** Displays details such as who made the changes and when they were made.
- **Check-in notes:** Provides descriptions of modifications, which are stored in the vCenter database until the template is deleted.
- **Version control:** Ensures historical tracking of template changes while maintaining deployment availability.

Maintaining detailed check-in notes improves documentation and helps track template evolution over time.

The following figure illustrates viewing template versions:

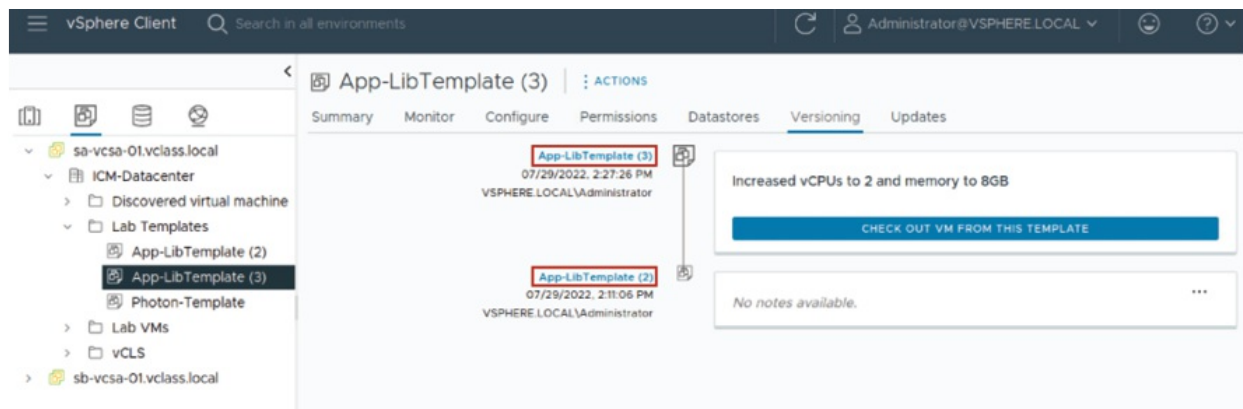


Figure 7.52: Viewing template versions

(Source: VMware)

Deleting and reverting to template versions

An administrator can manage VM template versions by either deleting outdated versions or reverting to a previous version when necessary. Select the required option as shown in [Figure 7.53](#).

- **Deleting version:** Administrator can remove the older version of a VM template to free up storage and maintain a clean inventory.
- **Revert to this version:** Administrator can roll back to an earlier version to restore the previous configuration if needed.

The following figure illustrates managing template versions:

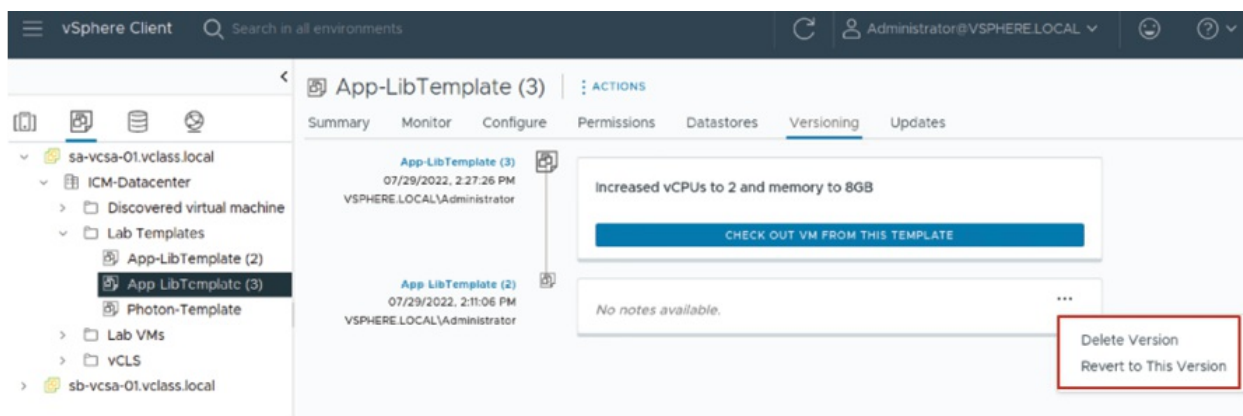


Figure 7.53: Managing template versions

(Source: VMware)

Note: Deleting the template will also remove all the contents from the inventory.

Conclusion

This chapter has taken readers through the heart of virtualization by enabling them to understand how to deploy virtual machines in a vSphere environment. Readers learned how to create and provision virtual machines, and how to install VMware Tools to set up and manage virtual infrastructure. They have learned about VM components, hardware versions, and configuration using the vSphere Client. Advanced topics were covered as well, such as creating and publishing VM templates, cloning virtual machines, and leveraging content libraries to simplify the process of deploying VMs by simplifying and streamlining it.

Having acquired a sound understanding of virtual machine deployment, the reader is adequately prepared to delve into the important concepts of *Virtual Machine Management* in [Chapter 8](#). This chapter goes into depth regarding managing the virtual infrastructure, from the details of VM migrations like vSphere vMotion and Storage vMotion, to understanding Enhanced vMotion Compatibility to ensure seamless migrations. Then, the readers learn to master the snapshot management techniques to maintain data integrity and resource efficiency. Further, the chapter introduces some CPU and memory management concepts related to resource allocation, reservations, and limits in virtualized environments.

Points to remember

- OVF templates simplify virtual machine provisioning, reduce the time to set up, and ensure consistency in deployments.
- Each VM compatibility level has at least five major or minor vSphere releases.
- Run the "unmap" command to free the available space from the virtual disks.
- A VM must be turned off before it is converted into a template.
- A Local Content Library is particularly suited for storing VM templates, ISO images, and scripts in a vCenter instance to provide consistency for virtual machine deployments.

- VM templates allow for faster VM provisioning, reducing errors compared to provisioning each virtual machine manually.
- Guest OS customization settings avoid problems resulting from duplicate configuration when deploying or cloning VMs from templates.

Exercises

1. How does deploying a VM from a template differ from cloning an existing VM?
2. What role does VMware Tools play in optimizing VM performance and management?
3. What are the key differences between an OVF template and a VM template stored in a content library?
4. Explain the difference between a local content library, a published content library, and a subscribed content library in vSphere.
5. What is the purpose of the "Check Out" and "Check In" processes when managing VM templates in the content library?
6. What are the key considerations when selecting a virtual hardware version for a VM?
7. What are hot-pluggable devices in vSphere, and which types of hardware can be added dynamically to a running VM?

Lab exercises

1. Creating, registering, and removing a virtual machine:

- a. **Objective:** Master the process of creating a new VM, unregistering it, and deleting it from the datastore.
- b. **Create a virtual machine:**
 - i. Log into the vSphere Client and select an ESXi host from the inventory.
 - ii. Go to VMs and Templates and select Create/Register VM.
 - iii. Select Create a new virtual machine and proceed through the wizard to enter VM name, compute resource, storage, and guest

OS.

iv. Finish the installation and boot up the VM.

c. Unregister the virtual machine from vCenter inventory:

i. Choose the VM from the inventory.

ii. Click Actions | Remove from Inventory.

iii. Verify the deletion (the VM files are still on the datastore).

d. Register the virtual machine to re-add it to vCenter inventory:

i. Open the datastore that contains VM files.

ii. Right-click the .vmx file and select Register VM.

iii. Install it on a host or resource pool and complete the process.

e. Delete the virtual machine from the datastore:

i. Choose the VM from the inventory.

ii. Click Actions | Delete from Disk to delete the VM and files permanently.

2. Adding and configuring virtual hardware:

a. **Objective:** Modify a virtual machine by adding new hardware components and comparing disk provisioning formats.

b. Examine a virtual machine's hardware configuration:

i. Select a VM from the inventory and click Edit Settings.

ii. Review the virtual CPU, memory, and storage settings.

c. Add virtual hard disks to the virtual machine:

a. In Edit Settings, click Add New Device > Hard Disk.

b. Choose a provisioning format (Thick or Thin) and specify the size.

3. Creating templates and deploying virtual machines:

a. **Objective:** Learn how to create VM templates, apply customization, and deploy VMs.

b. Create a virtual machine template:

i. Right-click a VM, select Clone | Clone to Template.

ii. Choose a destination datastore and complete the process.

c. Deploy virtual machines from a template:

- i. Right-click the template, choose New VM from This Template.
 - ii. Apply a customization specification and complete the wizard.
- 4. **Using subscribed content libraries:**
 - a. **Objective:** Set up a subscribed content library that synchronizes with a published library.
 - b. **Publish a local content library:**
 - i. Open the existing library and enable Publishing in settings.
 - ii. Generate a subscription URL and set authentication if needed.
 - c. **Create a subscribed content library:**
 - i. Navigate to Content Libraries and create a new library.
 - ii. Select Subscribed Library and enter the subscription URL.
 - d. **Deploy a VM from a Subscribed Content Library:**
 - i. Synchronize the subscribed library with the published source.
 - ii. Choose a VM template and deploy a new virtual machine.

CHAPTER 8

Virtual Machines Management

Introduction

In all virtualized infrastructures, **virtual machines (VMs)** are the base, offering applications, services, and workloads to end-users. With growing environments, proper management of VMs is essential to offer performance, availability, and efficient use of resources. VM management is more than provisioning, migrating VMs among hosts or datastores without interruption, relying on snapshots for backup and restore, and assigning CPU and memory resources to maximize use.

Migrations enable workload balancing and maintenance, snapshots offer rapid restore points, and smart resource allocation, through shares, reservations, and limits, offer fairness when several VMs utilize hardware. In short, effective virtual machine management is part of the overall agility, stability, and performance of the virtual infrastructure as a whole. This chapter aims to provide a solid foundation for these core competencies.

Note: VMware is now part of Broadcom and is known as 'VMware by Broadcom'. All references to 'VMware' in this book reflect this change.

Structure

In this chapter, we will cover the following topics:

- VM migrations types and strategies
- Migration compatibility with EVC
- VM migration with vSphere Storage vMotion
- VM migration across vCenter
- Snapshot management
- CPU and memory concepts and considerations
- VM resource allocation

Objectives

By the end of this chapter, the readers will have a thorough understanding of the virtual machine migration process both inside and outside vCenter Server instances. They will be able to identify different types of migration and be aware of the underlying fundamental concepts of vSphere vMotion, vSphere Storage vMotion, and the conditions that need to be fulfilled to deploy them successfully. Readers will be able to relocate virtual machines based on computational resources, storage solutions, or both, depending on operational needs. Additionally, this chapter discusses the role of **Enhanced vMotion Compatibility (EVC)** in facilitating smooth VM migrations between hosts with different CPU capabilities, explaining the configuration of EVC modes at cluster and VM levels, and the use of EVC Graphics mode for applications dependent on GPU resources.

In addition, readers will gain expertise in snapshot management, including the creation, deletion, consolidation, and efficient management of multiple snapshots. A detailed discussion of CPU and memory concepts will further readers' understanding of resource usage and sharing in a virtualized environment. The chapter outlines critical approaches to addressing memory overcommitment, the working of VMware Virtual SMP and hyperthreading, and ways of allocating CPU and memory resources using shares, reservations, and limits. Together, these topics will provide readers with the tools necessary to maintain performance, ensure availability, and optimize resource usage in their vSphere environment.

VM migrations types and strategies

VM migration refers to the ability to relocate a virtual machine from one place to another within the vSphere environment, this could be between hosts, datastores, or between vCenter instances. Such a feature is required to ensure system flexibility, carrying out hardware maintenance, and load distribution without breaking services.

Virtual machine migration can be classified into two broad categories:

- **Cold migration:** Refers to the operation of migrating a powered-off or suspended virtual machine. As the virtual machine is not running, cold migration will not involve a risk of service interruption and is commonly employed for routine resource reassignment or planned maintenance periods.
- **Hot migration:** This process allows active VM migration from one host to the other with zero downtime. It is very useful for load-balancing, maintenance, and addressing dynamic resource needs efficiently in real-time.

Prior to migrating, vCenter Server verifies compatibility to ensure the target has the capability to accommodate the VM. The verifications involve hardware, software, and configuration compatibility to avoid interruptions and workload integrity.

VM migration, whether hot or cold, is a cornerstone of a responsive and stable virtual infrastructure.

VM migration options

The kind of migration administrators do in vSphere depends on two things: the virtual machine's power state at present and the migration option chosen in the migrate wizard. VMware vSphere offers a variety of flexible migration options, every one of which has varying use cases and scenarios for use.

The migrate wizard provides several types of migrations options:

- **Compute resource only:** Moves a virtual machine to a different host except for changing its storage location. For powered-on VMs, vSphere vMotion enables this migration without service disruption.
- **Storage only:** Moves the files of a VM to another datastore without

changing its host. For powered-on VMs, vSphere Storage vMotion enables this migration without service disruption.

- **Both computing capacity and storage space:** It moves a VM to a new host and a new datastore. In cases of powered-on virtual machine, vSphere vMotion along with Storage vMotion work in tandem to allow live migration.
- **Cross vCenter server export:** This option migrates a VM to another host and datastore managed by a different vCenter Server instance, even between non-linked SSO domains.

This type of migration helps with infrastructure consolidation or restructuring between sites or vCenters. The migration approach is usually determined by the particular operational requirement. For instance, to get a host ready for maintenance without breaking the continuity of its virtual machines, vSphere vMotion is the best option. However, if the purpose is to rebalance the storage workload or relocate to a new storage array, Storage vMotion offers a non-disruptive option. It should be noted that some migration methods, e.g., vMotion, consider special hardware and network demands (e.g., shared storage, same CPUs, and proper VMkernel configurations). As opposed to those, tools like cold migration (of powered-down or suspended VMs) are less demanding in requirements and do not require special hardware conditions.

The following figure illustrates the VM migration types:

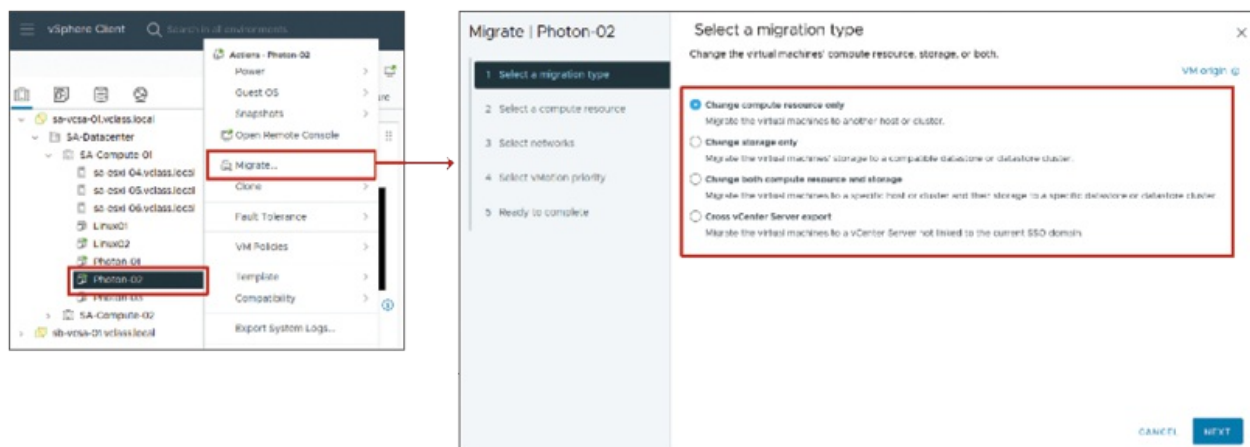


Figure 8.1: VM migration types

(Source: VMware)

Understanding vSphere vMotion

vSphere vMotion is a central part of VMware's virtualization infrastructure, allowing real-time movement of an active VM from one ESXi host to a different host, completed effortlessly and without affecting service. Such support is vital in dynamic environments where maximizing availability and resource usage is of the highest priority.

One of the most significant advantages of vSphere vMotion is that it can redistribute computing loads without negatively affecting end users. During planned maintenance or load balancing situations, virtual machines can be moved from one host to another without being shut down. Such flexibility ensures high availability, makes the operations tasks easier, and provides maximum performance with the workloads being balanced in the cluster.

When migrating a VM between hosts using vSphere vMotion, its global activity state, memory contents in their entirety, and all config settings at a system level are moved to the destination host. The state consists of transaction metadata, OS/app memory, as well as system metadata such as virtual device maps, BIOS/EFI settings, CPU settings, and MACs.

The following figure illustrates vSphere vMotion:

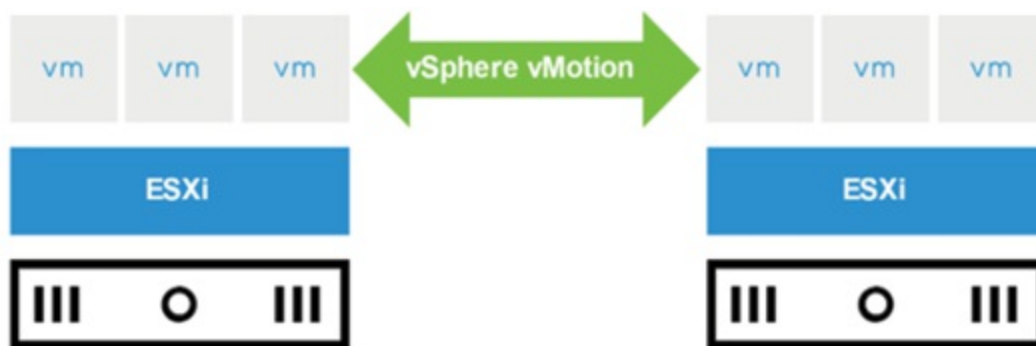


Figure 8.2: vSphere vMotion

(Source: VMware)

Configuring vSphere vMotion networks

To support vSphere vMotion capability, all participating hosts must have a specific VMkernel adapter with the vSphere vMotion service enabled. This special VMkernel interface allows for the transfer of a VM's memory and state data over the network during a migration. Without this setup, vMotion

migrations will not start.

Best practice is to locate the vMotion VMkernel adapters on a separate, high-speed network segment that is isolated from management and virtual machine traffic. Isolating the traffic in this way avoids high rates of migration traffic from polluting other network traffic, allowing vMotion operations to be completed quickly and efficiently.

Correct network configuration is a basic prerequisite for successful vMotion operations and must be properly planned and verified during the infrastructure installation.

The following figure illustrates the vSphere vMotion network configurations:

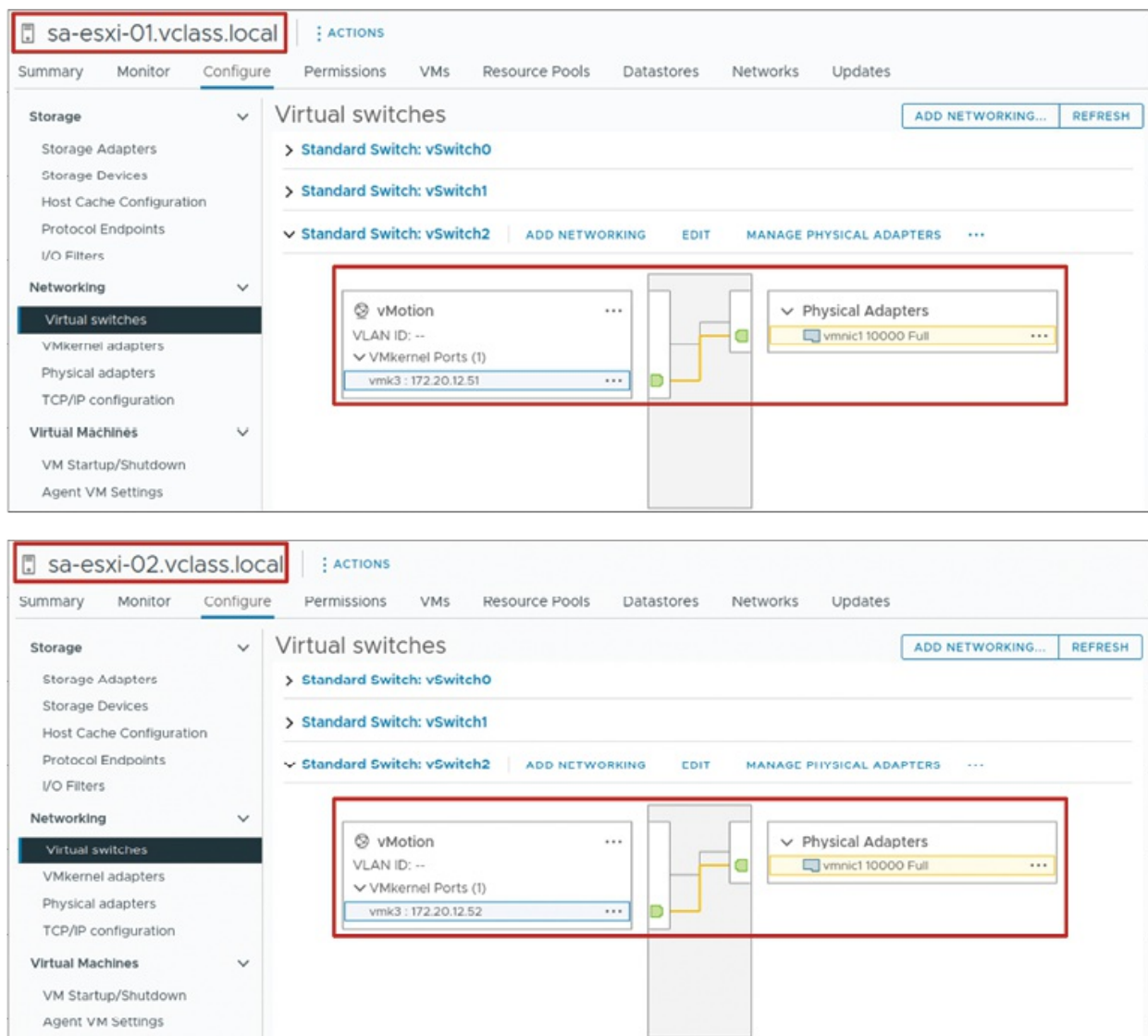


Figure 8.3: Configuring vSphere vMotion networks

(Source: VMware)

Best practices for vSphere vMotion networking

vSphere vMotion configuration tuning is essential to enable efficient, reliable, and disruption-free migration. Administrators must adhere to defined best practices to ensure performance, availability, and compatibility throughout the infrastructure. The best practices are as follows:

- One of the most significant recommendations is having at least one other physical NIC reserved for failover. This provides redundancy in the event of failures with the primary NIC, and the network will function normally with vMotion operations.
- Jumbo frames offer bigger packet sizes, less CPU overhead and memory bandwidth during memory-intensive vMotion transfers. To take full advantage, jumbo frames need to be enabled end-to-end in the vMotion path - starting with the physical NICs on the ESXi host, the virtual switches, and all intermediary physical switches.
- Where vMotion traffic must be routed across different IP subnets, administrators can utilize the vMotion TCP/IP stack. The vMotion TCP/IP stack facilitates the transfer of vMotion traffic through a specific dedicated default gateway regardless of the management network. Isolation is increased through improved efficiency of traffic flow as well as towards greater security and partitioning of networks.

By implementing these best practices, the administrators can ensure that vSphere vMotion operations are scalable, resilient, and efficient in the environment.

vSphere vMotion migration workflow

The vSphere vMotion process enables live migration of a powered-on virtual machine from one ESXi host to another with zero downtime. This is done by a highly coordinated sequence of operations between the target and source hosts while users keep on using the VM as usual.

Let us assume that both the source host (ESXi01) and the destination host (ESXi02) share a common datastore where the VM files are kept. Shared accessibility is required for seamless and uninterrupted VM migration.

The vMotion process goes through several key steps:

1. **Shadow VM creation:** On the target host, a shadow copy of the VM is created. It is a holder for the incoming VM state.
2. **Memory copy initiation:** The source host requests copying the full memory state of the VM over the vMotion network to the target host. The VM is maintained in full active state, and the users are free to interact with it. Memory pages that get changed during this interval are tracked through a memory bitmap.
3. **Iterative memory transfer:** Once the initial copy of the memory is finished, a cycle of iterative passes is initiated wherein only the modified pages of the memory since the previous pass are copied. The process is continued until the count of modified pages is low enough to be copied within a timeframe of less than 500 milliseconds.
4. **Quiescing the VM:** The VM is paused temporarily (quiesced) to achieve data consistency. The latest set of memory pages and VM device state (e.g., virtual CPU, disk, and network settings) are stored on the target host during the quiesce interval.
5. **Transition to Destination Host:** After the device state and memory transfer is successfully completed, the **virtual machine (VM)** then goes on to function normally on the destination host. A **Gratuitous ARP (GARP)** request is sent to inform the network of the VM's MAC address migration to a different switch port to allow transparent redirection of user connections.
6. **Host lock transition:** The source host releases the file locks for the virtual machine's storage, and the destination host then acquires them, thus completing the migration process.
7. **Memory cleanup:** Finally, the source host marks the VM's pre-allocated memory pages as free, thereby releasing them for use by other workloads.

During this process, users feel no noticeable interruption. The virtual machine keeps running as if nothing has changed, while vMotion coordinates the underlying processes with almost perfect precision.

For a detailed review of the internals of vMotion technicalities, VMware's *The vMotion Process Under the Hood* blog post is very informative. It can be

accessed here <https://blogs.vmware.com/vsphere/2019/07/the-vmotion-process-under-the-hood.html>.

VM requirements for vSphere vMotion migration

Before initiating a vSphere vMotion migration, it must be confirmed that the virtual machine has the following prerequisites to allow an uninterrupted and seamless migration process:

- One crucial matter to consider would be **raw device mapping (RDM)**. Assuming a virtual machine is running using an RDM disk, having the RDM mapping file along with the attendant **logical unit number (LUN)** accessible from the destination host is paramount. This ensures uninterrupted availability of raw device data for the virtual machine in the event it is migrated.
- Another requirement is in the case of virtual devices. The VM must not be connected to a virtual device such as CD/DVD or floppy drive whose host-local image is mounted. Host-local media devices can cause a dependency on the source host, and that will prevent the migration process. But vSphere vMotion supports migration of VMs with devices connected via remote console, even physical devices, or remote-accessible disk images.

Satisfying these requirements is important towards guaranteeing continuity of operations and guaranteeing availability of resources, thus vSphere vMotion is an effective and dependable method for live virtual machine migration.

Host requirements for vSphere vMotion migration

To successfully run a vSphere vMotion, the source and target ESXi hosts must fulfil a few important requirements:

- Most importantly, both hosts need to have access to the storage resources of the virtual machine. Two-way access is important, particularly during live migrations. Any one datastore can support up to 128 simultaneous migrations, but performance will suffer if a swap file needs to be relocated, especially when the swap file location is different on the source and target hosts. Here, the file is copied, which can slow down the migration. If the target host does not have access to the provided swap

location, it will write out the swap file and the VM's configuration file.

- Both hosts should be configured with a VMkernel port for vSphere vMotion, and the associated IP address class (IPv4 or IPv6) should be the same on both. For example, migration of a virtual machine from a host that is registered with an IPv4 address to a host with IPv6 is not supported.
- vSphere vMotion requires adequate network throughput to perform best. A minimum of 250 Mbps network bandwidth is required for each uninterrupted migration. For a network of 1 Gbps, four migrations can be supported concurrently, but 10 Gbps and above networks can support eight migrations at the same time. To work best, at least two VMkernel port groups must be allocated for vMotion traffic exclusively.
- Lastly, CPU compatibility between the source and destination hosts is necessary. The CPUs should be capable of supporting a compatible set of features. When the CPUs are different, **Enhanced vMotion Compatibility (EVC)** or CPU compatibility masks can be utilized to mask the differences and facilitate a smooth migration process.

Performing a vSphere vMotion migration

vSphere vMotion allows transparent migration of a running virtual machine from one ESXi host to another with no downtime. Migration moves the processing load and delivers uninterrupted access to the virtual machine's resources and services.

When initiating a migration using the migration wizard, a few choices are available based on what is required:

- **Change compute resources:** This operation moves the virtual machine to a different host without altering its storage on the same datastore.
- **Change both compute and storage:** The virtual machine is moved to another host and a new datastore simultaneously.
- **Cross vCenter server export:** Migrates the VM to a new host and datastore in another vCenter server instance.

In each of these situations, if the VM is online, vSphere vMotion facilitates the migration operation. This amazing feature allows administrators to perform maintenance on the hardware, load balance, or any other operational

functions without affecting VM availability or access by users.

The following figure illustrates the choice of vSphere vMotion migration:

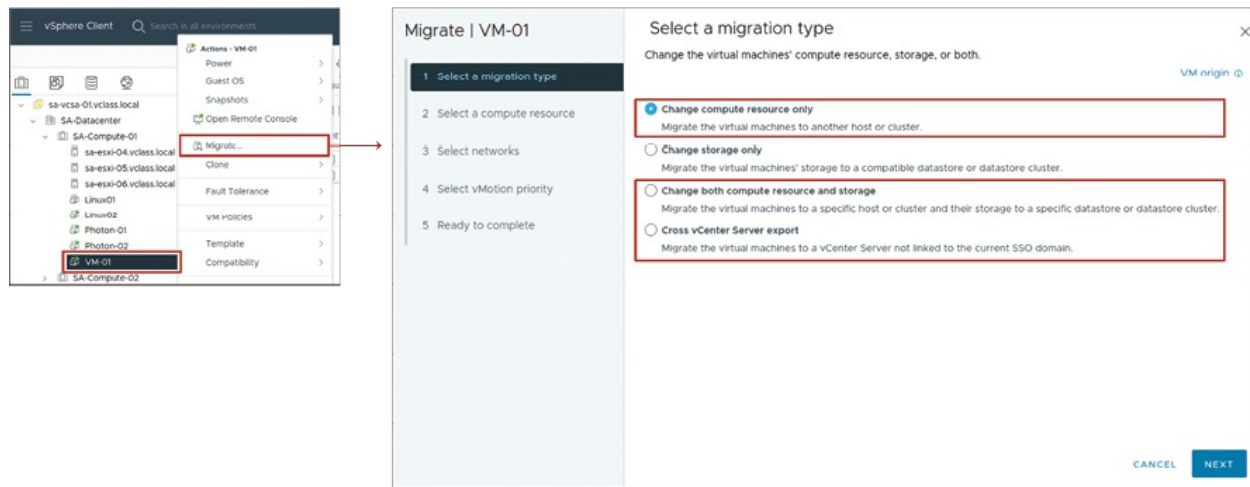


Figure 8.4: Performing a vSphere vMotion migration

(Source: VMware)

Migration errors validation

Before initiating a migration of a vSphere vMotion, the wizard performs automatic validation tests to determine whether the destination host or cluster selected meets all the required specifications. This is crucial to avoid any possible issues during migration.

If the validation is successful, the administrator can go ahead with the migration. But if the system finds any problems, they are shown in the compatibility pane of the wizard as warnings or errors:

- Warnings are alerts of potential problems, but they do not stop the migration from proceeding. It is the administrator's choice to continue.
- Errors, on the other hand, will impede the migration process. It is essential to rectify these issues prior to making another attempt at the migration.

In the case where a migration gets stuck halfway, vSphere vMotion ensures that the virtual machine continues to run securely on the source machine without interfering with its operation.

The following figure illustrates the vSphere vMotion migration errors:

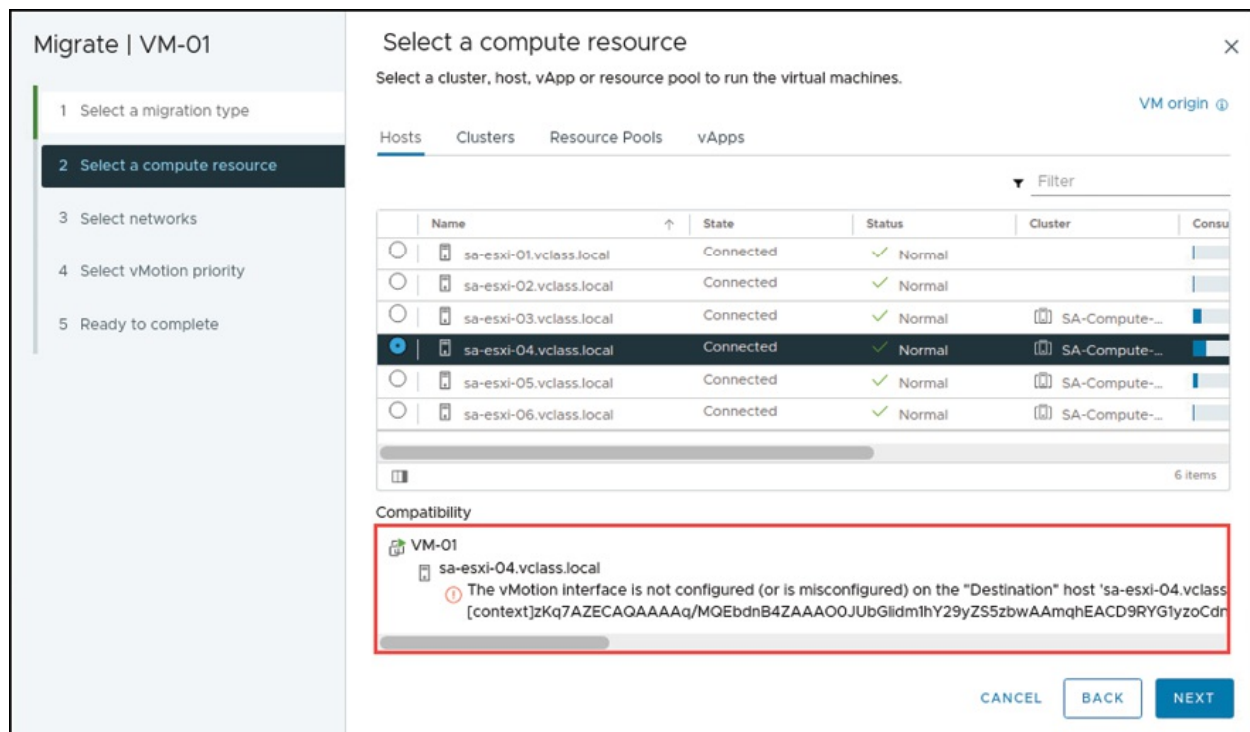


Figure 8.5: Validating migration errors

(Source: VMware)

Migrating encrypted VMs

During powered-on encrypted VM vMotion, encrypted vSphere vMotion is enabled by default. This protects the data in transit during migration, ensuring confidentiality, integrity, and authenticity.

For non-encrypted VMs, readers can choose from the following encrypted vSphere vMotion settings:

- **Disabled:** Encryption is not being used during migration.
- **Opportunistic (default):** Encryption is employed only if both source and destination hosts can support it.
- **Required:** Both the hosts must be configured to support encrypted vSphere vMotion; otherwise, the migration will fail.

Encrypted vSphere vMotion is supported for all vMotion types, ranging from migrations between vCenter instances. Encrypted vSphere Storage vMotion is not supported.

Besides, whenever it is required to work on encrypted VMs, encrypted vSphere vMotion cannot be disabled, as it is a mandatory security feature for

those workloads.

The following figure illustrates encrypted vSphere vMotion:

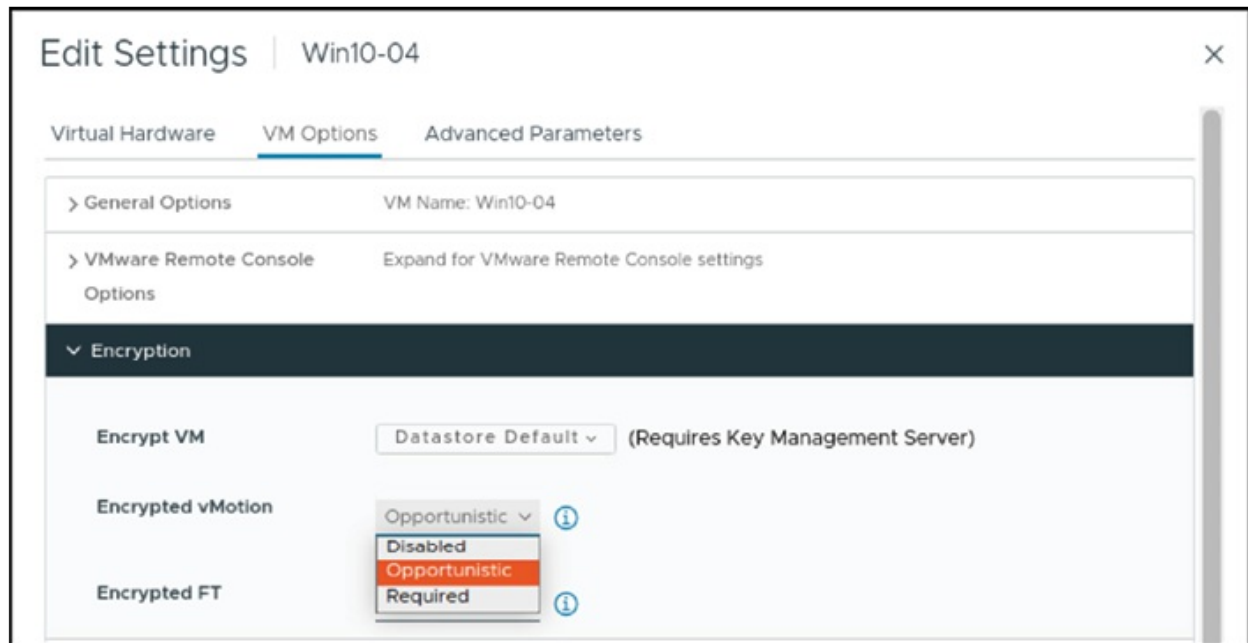


Figure 8.6: Encrypted vSphere vMotion

(Source: VMware)

Migration compatibility with EVC

During a vSphere vMotion migration, CPU compatibility must be set between the source and destination hosts. The vSphere environment performs a compatibility check to ensure that the virtual machine being migrated will operate without issues on the destination host, thereby avoiding CPU instruction mismatches that may lead to errors or system instability.

Some features of CPUs do not require an exact match among hosts; however, others require a perfect match to enable vMotion. For instance, hyperthreading may be varied across different hosts; one host might have it turned on, and another might not - yet the VMkernel adjusts, and the migration proceeds successfully.

If the source host itself has SSE4.1 instruction support but the destination host lacks this, then migration will be unsuccessful. This is because these multimedia instructions run directly within applications, and any

incompatibility will result in application crashes or performance degradation upon migration.

Understanding these constraints is crucial for successful migration deployment and ensuring consistent application performance after virtual machine migration.

Understanding EVC

In a virtual environment, not every physical host contains identical CPUs. That is a problem in terms of compatibility when migrating VMs from one host to another using vSphere vMotion. This is where EVC is intended to make vMotion migrations between different CPU generations within a cluster easier.

EVC normalizes the CPU feature set presented to virtual machines across all hosts in a cluster. It does this by enforcing a shared CPU baseline, essentially hiding more advanced features on newer CPUs so that all hosts present the same features to the VMs. This presentation consistency allows VMs to be moved from host to host without CPU compatibility problems.

With the activation of EVC in a cluster, all the current and prospective hosts must meet the specified baseline. Those hosts that fall below the minimum CPU requirements of the baseline are not eligible to be included in the cluster. Virtual machines operating in the EVC enabled cluster always *perceive* uniform virtual CPU features irrespective of the underlying hardware variations.

The following figure illustrates the enhanced vMotion Compatibility:

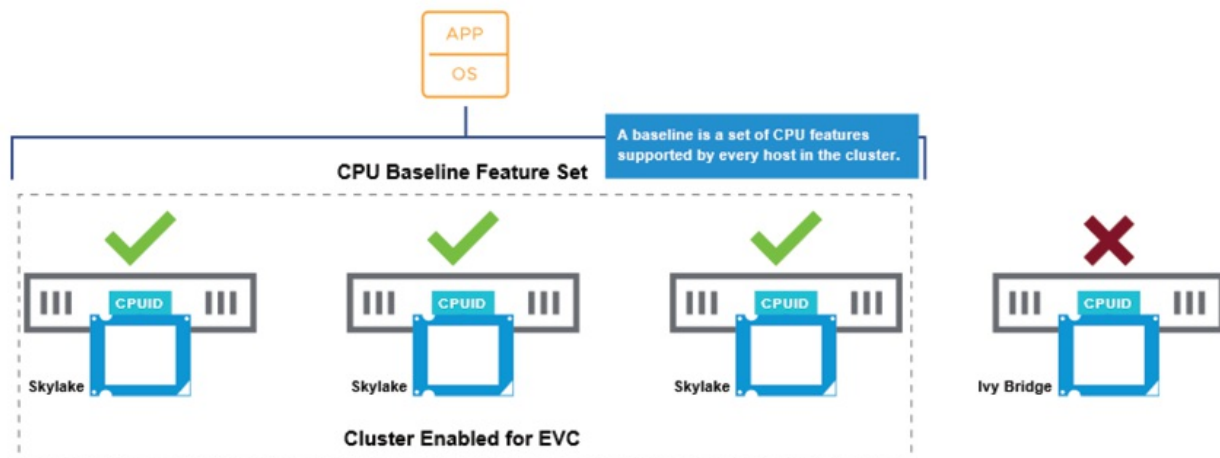


Figure 8.7: Enhanced vMotion Compatibility

(Source: VMware)

EVC clusters requirements for CPU mode

To operate properly, EVC requires all hosts in the cluster to comply with certain CPU-related requirements. These are listed as follows:

- **Same CPU vendor:** All hosts should either use CPUs from AMD or Intel, vendors should not be combined.
- **Hardware virtualization supported:** Intel VT or AMD-V should be enabled within the BIOS.
- **Execution disable support:** Intel XD or AMD NX technology should be enabled for security and compatibility.
- **vSphere vMotion readiness:** Hosts are to be configured to support vMotion migrations.
- **CPU ID compatibility:** The software running in VMs should employ the CPU ID instruction to identify features, since Intel and AMD both recommend this.

Prior to creating an EVC-supported cluster, one must ensure that all targeted hosts to be included in the cluster satisfy these conditions. Intel FlexMigration or AMD-V Extended Migration CPUs assist in ensuring compatibility with previous processors to offer wider support within EVC.

To verify what EVC modes host CPUs support, go to the VMware Compatibility Guide at https://vcg-prod-vip-1.broadcom.com/comp_guide2/search.php. Just search by server model or CPU family and look at the supported EVC modes in the CPU Series section.

Setting the EVC CPU mode on an existing cluster

Enabling EVC in a cluster ensures that all hosts provide VMs with a consistent CPU feature set, thus enabling transparent and compatible vSphere vMotion migrations across multiple generations of hosts.

There are two major methods to allow EVC:

- **Recommended procedure:** Create a new, empty cluster, configure the EVC mode, and subsequently add compatible hosts to it. This does not

disrupt active VMs.

- **On an existing cluster:** The other option is to enable EVC on an existing cluster, but at the price of putting all hosts into maintenance mode, i.e., all VMs would have to be shut down or suspended, causing downtime.

Therefore, the safer and more efficient method is to set up EVC before adding hosts and virtual machines. This method saves system downtime while keeping CPU compatibility intact to support future migrations.

The following figure illustrates EVC CPU mode on an existing cluster:

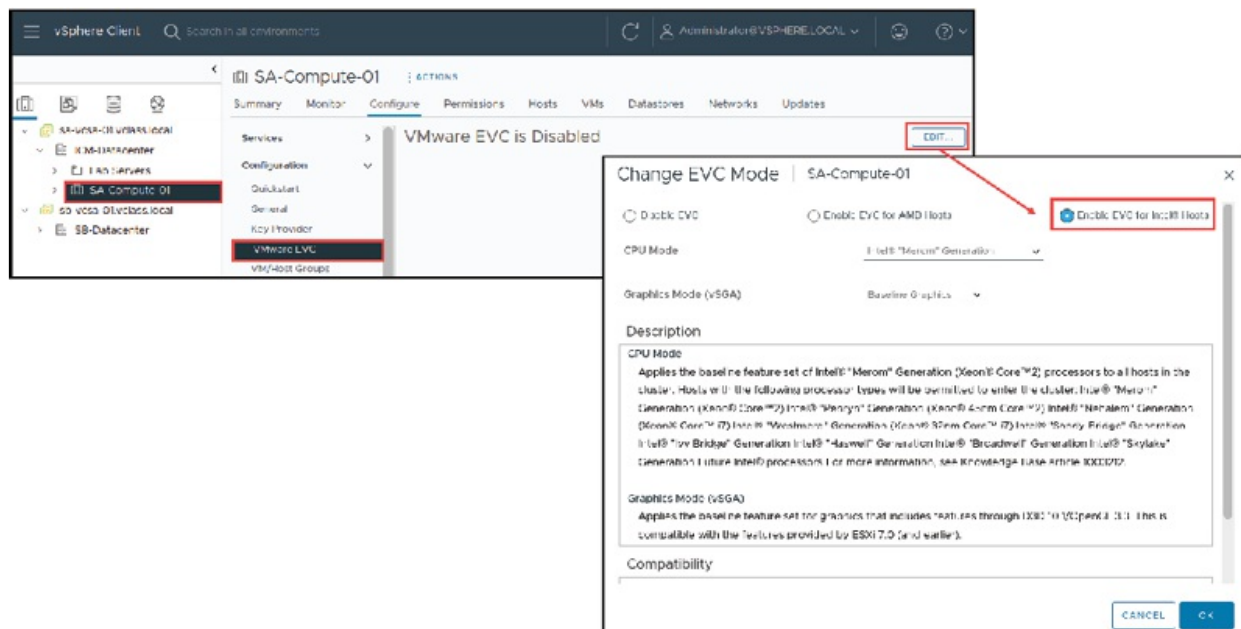


Figure 8.8: Configuring EVC CPU mode on an existing cluster

(Source: VMware)

Changing the EVC CPU mode for a cluster

EVC mode allows for preserving CPU compatibility across hosts within a cluster, and it can be configured to suit changing infrastructure requirements. When all hosts within a cluster are compatible with a later CPU baseline, administrators can have the EVC mode reconfigured, even on clusters that were not initially designed with EVC.

Changes in EVC mode can be in two possible directions:

- **Increasing the EVC mode:** You can increase the EVC mode to support newer CPU features. Currently running VMs can remain powered on, but

the new features will not be available to them yet. VMs need to be powered off and restarted (a suspend/resume cycle is not sufficient).

- **Decreasing the EVC mode:** To change to a smaller baseline, maybe to accommodate legacy hardware, shut down any virtual machine that uses features from the larger EVC mode before you make the switch. The process ensures an identical and compatible CPU feature set for the whole cluster. Such versatility enables administrators to modify their settings to accommodate hardware adjustments while simultaneously enabling vMotion migrations to run smoothly.

Virtual machine EVC CPU mode

EVC is not only cluster-specific; it can even be configured at the VM level. The per-VM EVC configuration is especially useful while performing VM migrations across various clusters or even vCenter instances and data centers with differing CPU generations.

By using per-VM EVC, the compatibility baseline is applied directly to the virtual machine, making the virtual machine portable and consistent throughout the host environment. This makes the virtual machine run on different hardware platforms if the host contains the given EVC mode of the virtual machine.

Unlike conventional cluster-level EVC, which normalizes CPU features across all hosts within a cluster, per-VM EVC is more flexible and more fine-grained. It separates the compatibility requirements of the VM from the underlying cluster configuration and is not lost even in the event of cross vCenter migrations or power cycling.

The following figure illustrates virtual machine EVC CPU mode:

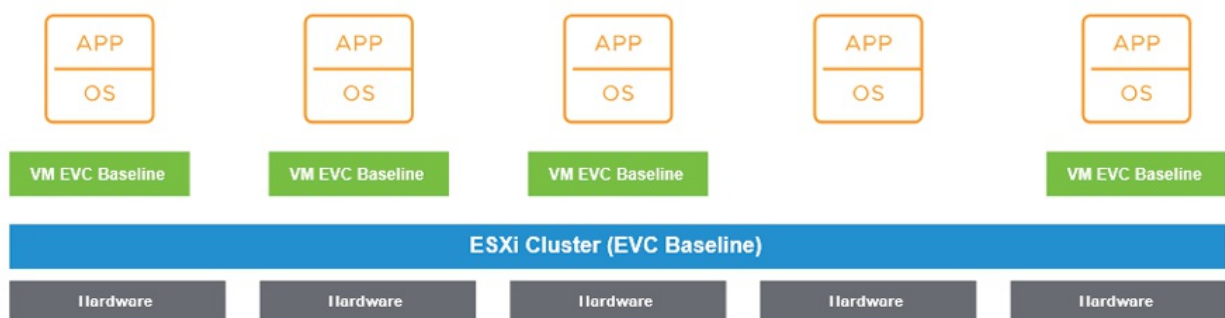


Figure 8.9: Virtual machine EVC CPU mode

(Source: VMware)

EVC for vSGA GPUs

The **vMotion Compatibility (EVC)** is not limited to CPUs alone but can also be used with **virtual Shared Graphics Acceleration (vSGA)** environments. With vSGA-enabled EVC, administrators can enable consistent GPU feature exposure across cluster hosts, simplifying the maintenance of compatibility for vMotion migration of VMs using virtual graphics.

When EVC for vSGA is enabled, a default GPU feature baseline is imposed on the cluster. The non-baseline features are masked, meaning they are not offered to the virtual machines to provide a consistent GPU capability set. This enables VMs on vSGA to migrate from one host to another without any compatibility issues.

The EVC for vSGA suggests the support for both hardware-based and software-based GPU renderers, thus offering flexibility across different deployment scenarios.

The following figure illustrates the EVC for vSGA GPUs:

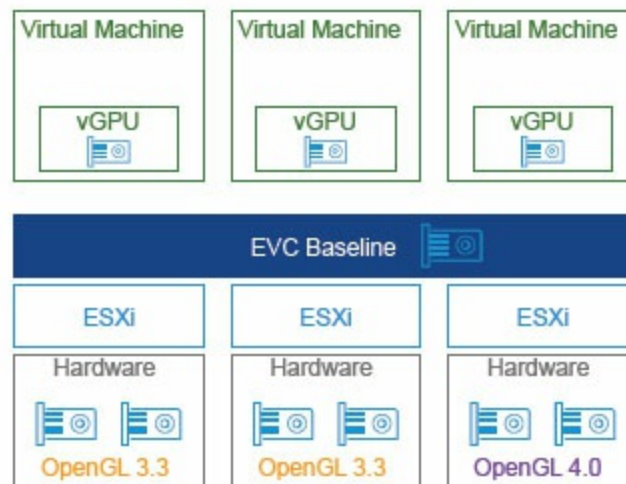


Figure 8.10: EVC for vSGA GPUs

(Source: VMware)

EVC cluster requirements for graphics mode

To enable EVC for vSGA GPUs, some prerequisites must be met at both the cluster and virtual machine level.

At the cluster level, all ESXi hosts should support the GPU feature baseline set for the cluster. If an ESXi host does not meet the baseline requirements, it will not be able to join the cluster. Mixed-version clusters, such as ESXi 6.7 and ESXi 7.0, are allowed when EVC is set at the cluster level, if all hosts meet the GPU baseline requirements.

At the VM level, EVC for vSGA GPUs requires VMs to be ESXi 7.0 Update 1 or later compatible. In particular, the VM hardware version should be version 18 or later.

Configuring EVC graphics mode on an existing cluster

To enable EVC for vSGA graphics, administrators configure it at the cluster level using the same EVC settings interface used for CPU compatibility. This integration allows for consistent GPU feature exposure across all hosts in the cluster, ensuring that VMs using vSGA can seamlessly migrate between hosts without compatibility issues.

The following figure illustrates configuring EVC graphics mode on an existing cluster:

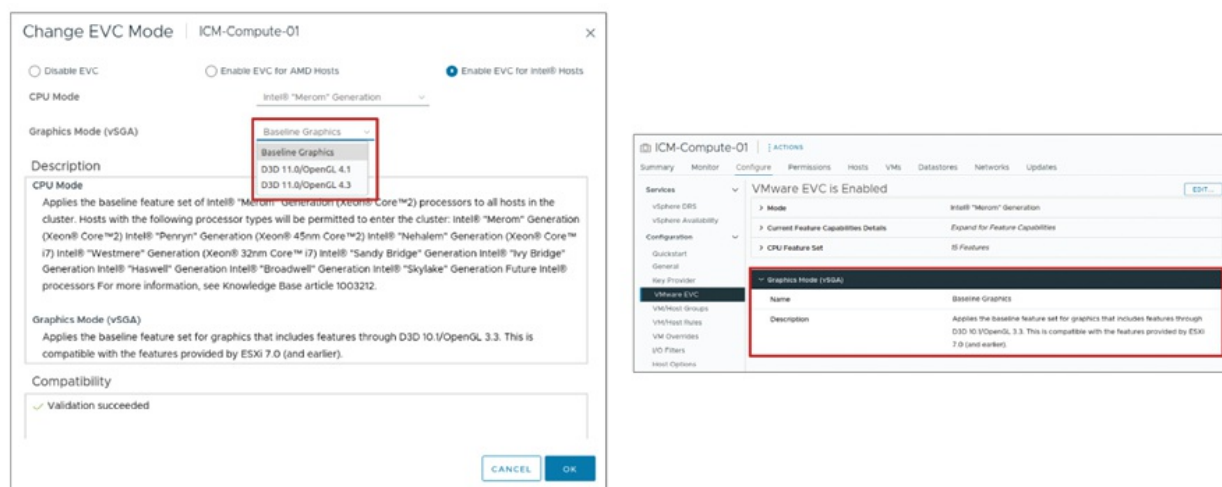


Figure 8.11: Configuring EVC graphics mode on an existing cluster

(Source: VMware)

Virtual machine EVC graphics mode

EVC for vSGA graphics can also be configured at the virtual machine level. This is done through the same VM EVC settings interface used for CPU compatibility. By applying EVC at the VM level, administrators ensure consistent GPU feature exposure for that VM, enabling it to be migrated across clusters or hosts with varying GPU capabilities, if the baseline requirements are met.

The following figure illustrates the VM EVC graphics mode:

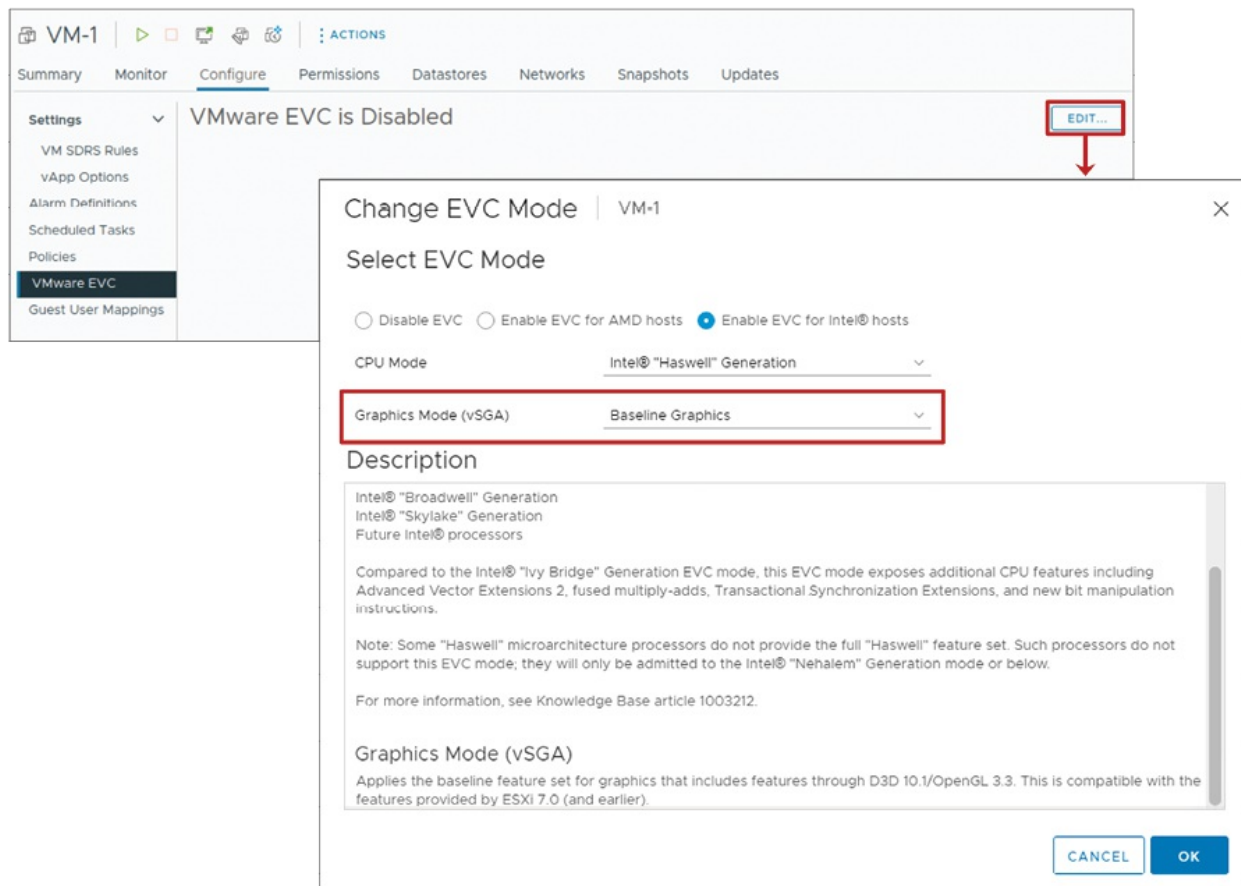


Figure 8.12: VM EVC graphics mode

(Source: VMware)

VM migration with vSphere Storage vMotion

vSphere Storage vMotion enables administrators to move a running virtual machine from one datastore to another, regardless of the type of datastore involved, without any downtime. This capability is particularly valuable in

operations such as relocating virtual machines from storage arrays for maintenance, storage infrastructure upgrades, or distributing input/output load across storage systems to achieve optimal performance and minimize latency.

Storage vMotion also allows administrators to modify disk provisioning types (for example, from thick to thin) and rename VM files to match the current inventory name. Administrators can relocate the entire VM (including configuration files and all virtual disks) to a single datastore or distribute them across multiple datastores.

Notably, this migration affects only the storage location. The VM continues to run on its original host without any interruption. With either Storage vMotion or cold migration, administrators can also choose to rename all VM-related files during migration, providing consistency and simpler management post-migration.

The following figure illustrates the vSphere Storage vMotion:

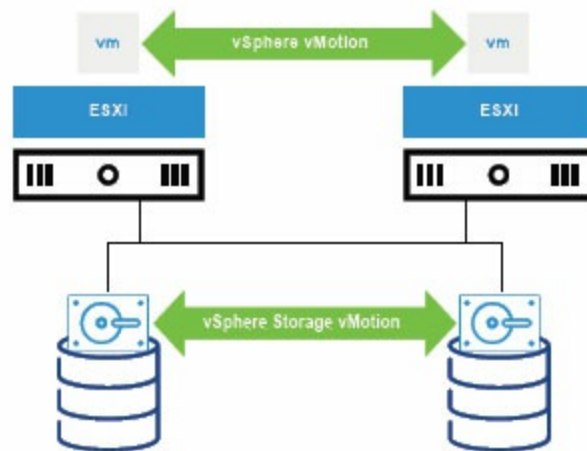


Figure 8.13: About vSphere Storage vMotion

(Source: VMware)

vSphere Storage vMotion deployment

vSphere Storage vMotion employs a robust input/output mirroring method to facilitate the migration of virtual machine disk data between datastores when the virtual machine is in an active running state. This method provides a non-disruptive and transparent migration experience for end users. The process is

done in several vital steps:

1. **Initiating migration:** The administrator initiates the migration of storage.
2. **Data copy begins:** The VMkernel's data mover, also known as the **vSphere Storage APIs for Array Integration (VAAI)**, commands the disk block movement to the target datastore.
3. **Launching a new process:** A new virtual machine process is initiated to facilitate the switch.
4. **I/O mirroring:** While the initial disk blocks are duplicated, any subsequent write operations are simultaneously replicated to both the source disk and the destination disk.
5. **Final transition:** When disk copy is complete and synchronization of the VM state is achieved, the virtual machine is moved to the target datastore and continues in normal operation from there.

This single-pass copying technique maintains data consistency without resorting to time-consuming recursive scans. The mirror driver in the VMkernel performs the synchronization in the background, allowing end users to work without interruption.

Depending on the storage environment, vSphere Storage vMotion can migrate internally on the ESXi host or offload to the storage array, if the array is VAAI-capable. Offloading to the array provides improved performance using hardware acceleration.

Identifying storage arrays that support VAAI

vSphere Storage vMotion can delegate migration work to the storage array itself, achieving faster and more efficient data movement, if specific requirements are fulfilled. This offloading relies on **vSphere Storage APIs for Array Integration (VAAI)**, also known as *hardware acceleration*.

For Storage vMotion to take advantage of hardware acceleration:

- The storage array must be VAAI-capable.
- Both the target and source datastores must be located on the same storage array.

To verify if the storage array supports VAAI, utilize the vSphere client and

sa-esxi-01.vclax.local
ACTIONS

Summary
Monitor
Configure
Permissions
VMs
Resource Pools
Datastores
Networks
Updates

Storage
Storage Adapters
Storage Devices
Host Cache Configuration
Printout Endpoints
I/O Hitters
Networking
Virtual switches
VMkernel adapters
Physical adapters
TCP/IP configuration
Virtual Machines
VM Startup/Shutdown
Agent VM Settings
Default VM Compatibility
Snap Extension

Storage Devices

REFRESH
ATTACH
DETACH
RENAME

<input type="checkbox"/>	Name	Y	LLN	Y	Type	Y	Capacity	Y	Datastore	Y	Operational State	Y	Hardware Acceleration	Y	Drive Type	Y	Transport	Y
<input type="checkbox"/>	FreeNAS iSCSI Disk (naa.659b9cfc:0000007f96b39ca20ef9a3)		3		disk		11.00 GB		Not Consumed		Attached		Supported		Flash		iSCSI	
<input type="checkbox"/>	FreeNAS iSCSI Disk (naa.659b9cfc:000000b363d62135d550a7e9)		7		disk		100.00 GB		Not Consumed		Attached		Supported		Flash		iSCSI	
<input type="checkbox"/>	Local VMware Disk (mpx.vmhba0:C0:T0:L0)		0		disk		16.00 GB		Not Consumed		Attached		Not supported		HDD		Parallel SCSI	
<input type="checkbox"/>	FreeNAS iSCSI Disk (naa.659b9cfc:000000550b10c0c06eef5c7...)		5		disk		130.00 GB		iSCSI Data...		Attached		Supported		Flash		iSCSI	
<input type="checkbox"/>	FreeNAS iSCSI Disk (naa.659b9cfc:000000f3d0d0a0b0b0b0d0d0)		0		disk		120.00 GB		ICM Dela...		Attached		Supported		Flash		iSCSI	
<input type="checkbox"/>	FreeNAS iSCSI Disk (naa.659b9cfc:000000550b10c0c06eef5c7...)		6		disk		7.00 GB		Shared-V...		Attached		Supported		HASH		iSCSI	
<input type="checkbox"/>	Local BECVVWar CD-ROM (mpx.vmhba0:C0:T0:L0)		0		cdrom				Not Consumed		Attached		Not supported		HDD		Disk Adapter	
<input type="checkbox"/>	FreeNAS iSCSI Disk (naa.659b9cfc:000000e3b3455c1c30aa986)		2		disk		11.00 GB		Shared-V...		Attached		Supported		Flash		iSCSI	
<input type="checkbox"/>	FreeNAS iSCSI Disk (naa.659b9cfc:000000d0c52f67d254b0dc)		4		disk		5.00 GB		Not Consumed		Attached		Supported		Flash		iSCSI	

EXPORT
9 items

(Source: VMware)

Before initiating a vSphere Storage vMotion migration, it is important to consider a few best practices and understand key limitations for a smooth and successful migration experience.

- Plan and coordinate with relevant stakeholders to avoid service disruption.
- Schedule migrations during off-peak hours to minimize the impact on performance and ensure resource availability.

- Independent disks must be in persistent mode to be eligible for migration.
- The VM and its host must meet configuration and resource requirements, including access to both the source and destination datastores.
- During the migration, administrator can change the disk provisioning type (for example, from thick to thin).
- The VM's files are renamed on the destination datastore to match the inventory name. This includes virtual disk files, configuration files, snapshots, and **.nvram** files.

- If any of the renamed files exceed the maximum allowed filename length, the migration will fail.

Changing compute resource and storage during migration

Sometimes, it will be necessary to migrate a virtual machine to another host and another datastore at the same time. This task is an integration of vSphere vMotion and Storage vMotion, allowing the change of both the virtual machine's processing resources and its storage location in a single unified process.

This kind of migration is useful when relocating VMs between clusters, data centers, or vCenter instances. It provides workload flexibility for hardware upgrades, resource balancing throughout the infrastructure without downtime or human intervention.

The following figure illustrates the compute and storage migration:

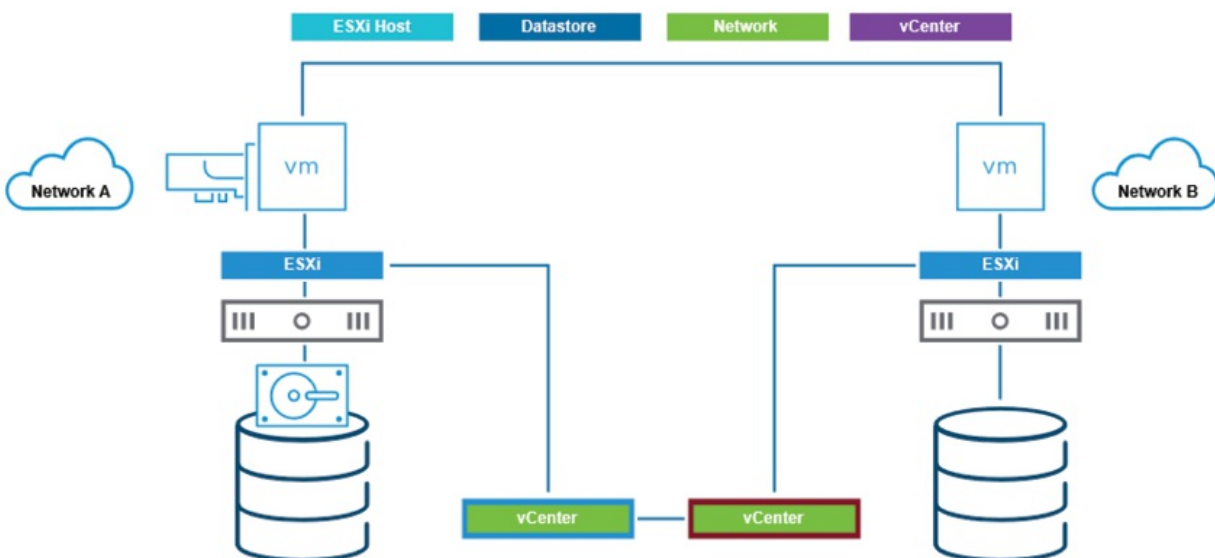


Figure 8.15: Compute and storage migration using Storage vMotion

(Source: VMware)

Use case for moving both compute resource and storage

Migrating both compute and storage together is best in situations where the target environment does not share storage with the source. Typical use cases are:

- Migration of a VM from a locally stored host to a shared storage host.
- Migration of a VM to a new cluster that is not able to access the original datastore.
- Migrating a VM to a different data center with its own storage.
- Transitioning a VM between different vCenter instances that do not share storage resources.
- This two-way migration provides flexibility in infrastructure management without downtime.
- This combined migration ensures flexibility in managing infrastructure while avoiding downtime.

VM migration across vCenter

vSphere vMotion enables the migration of virtual machines among different vCenter Server instances, regardless of whether they are part of an **Enhanced Linked Mode (ELM)** group. vSphere vMotion is a highly flexible and scalable feature, beneficial in complex or dispersed environments.

Cross vCenter migrations are particularly useful in scenarios such as:

- Uniform distribution of workloads among clusters or instances of vCenter, both site-based and geographically dispersed data centers.
- Promoting VMs from one environment to another, for instance, moving a virtual machine from development to production.
- Meeting various **service-level agreements (SLAs)** by relocating virtual machines to storage or compute resources that are closer to the required performance or capacity demands.
- Migrating workloads to the cloud, for example, moving virtual machines from a local data center to a cloud like VMware Cloud on AWS.

Cross vCenter migration requirements

To successfully perform a cross vCenter migration, a few key requirements must be met to ensure security, compatibility, and performance:

- **Time synchronization:** The two vCenter server instances must be time-synchronized. This is necessary for vCenter **Single Sign-On (SSO)**

token validation during the migration process.

- **vCenter SSO domain:** The vCenter instances can either be within the same SSO domain or different ones. Cross vCenter migrations do not require ELM.
- **Storage access:** In case only the compute resource is moving (the host), both vCenter instances should be able to access the shared storage where the VM is located.
- **Version compatibility:** vCenter migrations among different versions of vCenter are supported, as long as they are compatible based on interoperability guidelines.

These specifications ensure that migrations between vCenters are performed in a manner that is secure, efficient, and appropriate for various deployment environments.

The following figure illustrates the cross vCenter migration:

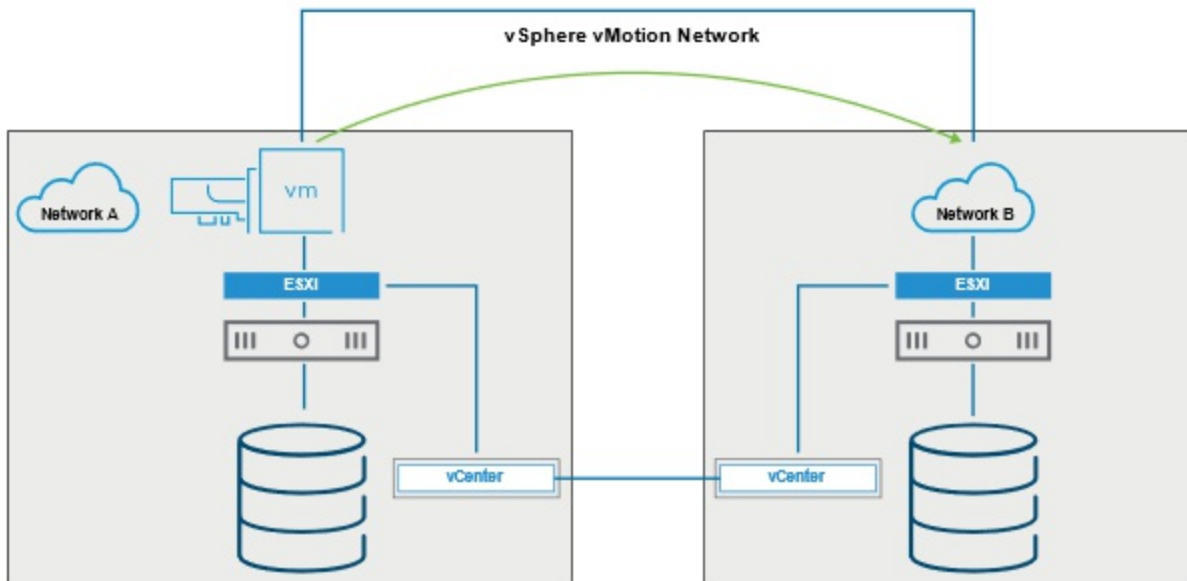


Figure 8.16: Cross vCenter migration

(Source: VMware)

Executing cross vCenter vMotion in same SSO domain

To start the migration process, enable the migrate wizard using the vSphere Client. During the process, it is essential to choose a compute resource, such as a host or cluster, of the target vCenter instance. Since the two vCenter

instances are linked through SSO, the resources of both environments are available within the interface, making it an easy process of migration without further manual authentication.

It is beneficial in the context of big environments where numerous vCenter instances are centrally managed within one authentication domain, which is easier to manage and migrate between sites or clusters.

The following figure illustrates the cross vCenter vMotion in same SSO domain:

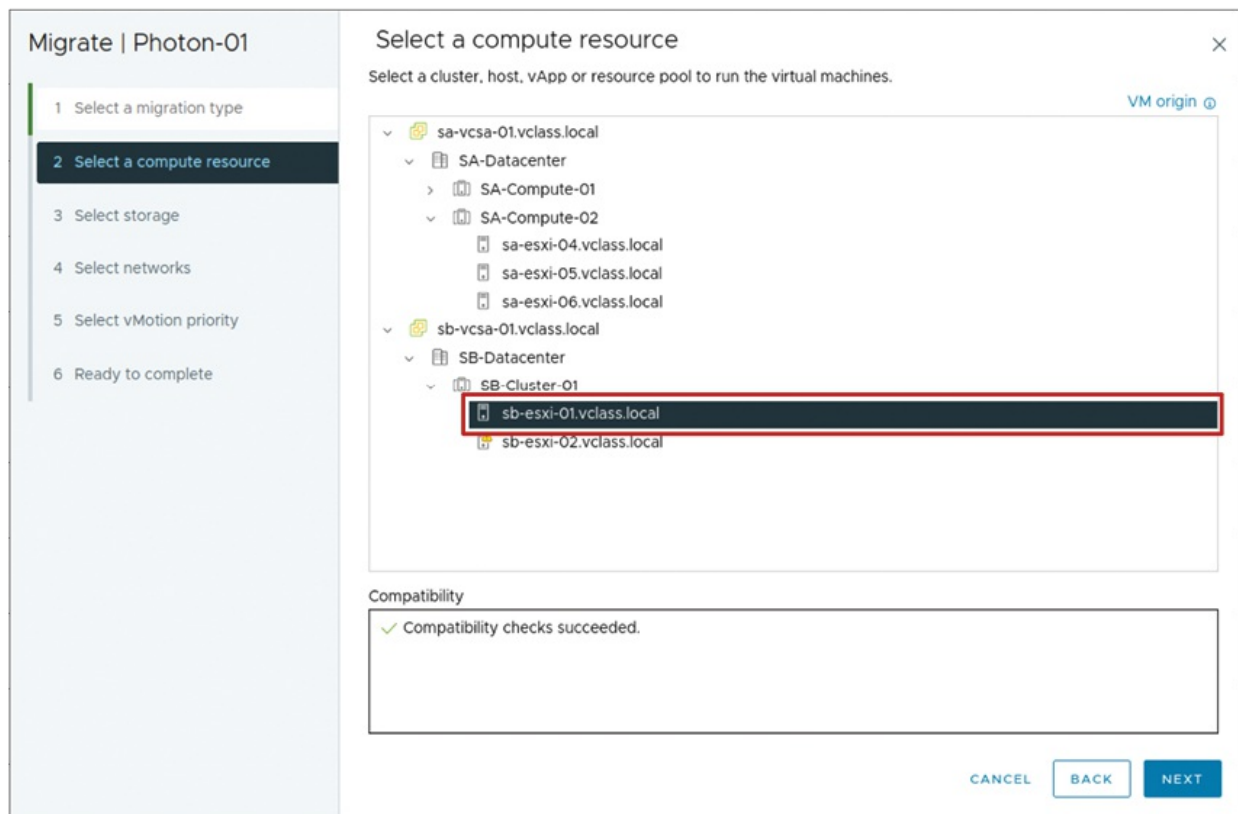


Figure 8.17: Performing cross vCenter vMotion in same SSO domain

(Source: VMware)

Performing cross vCenter vMotion in different SSO domain

When migrating a virtual machine across vCenter instances that are part of different SSO domains, the process requires a different method and includes the manual input of the target credentials. To begin, launch the migrate wizard and choose **Cross vCenter Server export** as the migration type.

There is also an option to clone the VM instead of moving it, which is helpful for testing or environment duplication.

Then input the destination vCenter instance FQDN or IP address and input valid vCenter credentials to authenticate. The rest of the process is a standard migration such as choose the destination compute resource, datastore, and VM network in the destination vCenter infrastructure.

This cross-domain migration helps achieve dynamic workload mobility between environments not connected with SSO, like from on-premises data centers and remote or cloud-hosted vCenter instances.

Figure 8.18 illustrates the cross vCenter vMotion in different SSO domain:

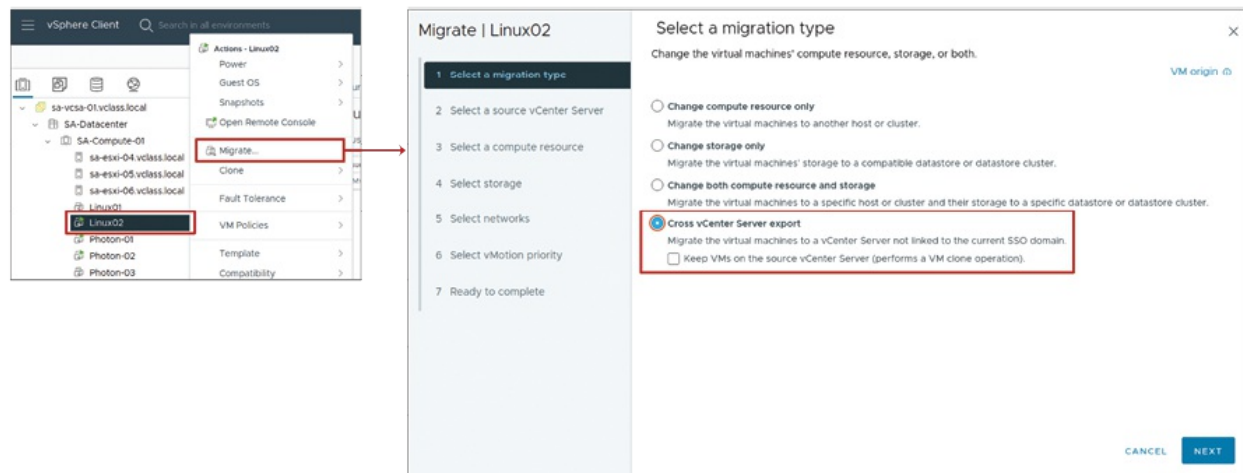


Figure 8.18: Performing cross vCenter vMotion in different SSO domain (1)

(Source: VMware)

In continuation, *Figure 8.19* shows the cross vCenter vMotion in different SSO domain:

Migrate | Linux02

1 Select a migration type

2 Select a source vCenter Server

3 Select a compute resource

4 Select storage

5 Select networks

6 Select vMotion priority

7 Ready to complete

Select a source vCenter Server

Import Virtual Machines from the selected source vCenter Server.

SAVED VCENTER SERVERS NEW VCENTER SERVER

✓ Successfully connected to sb-vcsa-01.vclass.local

vCenter Server address sb-vcsa-01.vclass.local
vCenter Server FQDN or IP address

Username administrator@vsphere.local
example@domain.local

Password *****
Password

Save vCenter Server address ⓘ ☒

LOGIN

CANCEL BACK NEXT

Figure 8.19: Performing cross vCenter vMotion in different SSO domain (2)

(Source: VMware)

Network compatibility checks during cross vCenter migrations

In cross vCenter migrations, vCenter itself performs a series of network compatibility tests to prevent problems that may impact the VM after migration. The tests ensure a seamless migration by checking the following:

- **MAC address compatibility:** It verifies MAC address incompatibility on the destination host.
- **Switch type consistency:** It prevents vSphere vMotion migration from a distributed switch to a standard switch.
- **Switch version compatibility:** It prevents vSphere vMotion migration between distributed switches of different versions.

Understanding VMkernel networking layer and TCP/IP stacks

For a vSphere infrastructure, the VMkernel's networking layer is necessary to enable connectivity in the form of system-level traffic, including vSphere

vMotion, IP storage, vSAN, and Fault Tolerance. To manage different types of network traffic, vSphere provides support for several TCP/IP stacks at the VMkernel level as mentioned:

- **Default TCP/IP stacks in vSphere:**

- **Default TCP/IP stack:** This setting controls vCenter to ESXi host traffic, together with system traffic like vMotion, IP storage, and Fault Tolerance.
- **vSphere vMotion TCP/IP stack:** Specifically designed to handle the network traffic for hot VM migrations, allowing them to be smooth and isolated transfers.
- **Provisioning of the TCP/IP stack:** Provisioning of the TCP/IP stack is needed for cold migrations, cloning, and snapshot operations. The stack is critical for enabling Long Distance vSphere vMotion because it handles the **Network File Copy (NFC)** traffic required for replication of virtual disks during migration.

Once a VMkernel adapter is dedicated to the provisioning stack, provisioning traffic is handled through it only, and the default TCP/IP stack is not utilized for this traffic anymore.

- **Custom TCP/IP stacks:** Administrators can also install custom TCP/IP stacks to manage some traffic types or to support proprietary applications. To install a custom stack, SSH to the host and run the following command:

```
esxcli network ip netstack add -N="stack_name"
```

Best practices for network security and performance

To maintain a secure and effective setting:

- Segregate vMotion traffic on a separate physical or logical network that is accessible only to the participating ESXi hosts.
- Limit access to management traffic to only authorized network and security administrators.

With proper utilization of these TCP/IP stacks, administrators can enhance traffic control, performance, and security in the vSphere infrastructure.

vSphere vMotion TCP/IP stack

Each ESXi host includes a dedicated TCP/IP stack specifically for vSphere vMotion, designed to handle the network traffic associated with hot migrations of virtual machines.

By assigning vMotion traffic to this dedicated stack, administrators can isolate and optimize migration performance without interfering with other types of network traffic. Once a VMkernel adapter is created on the vSphere vMotion TCP/IP stack, that stack becomes the exclusive path for vMotion traffic on the host.

At that point, any VMkernel adapters previously configured for vMotion on the default TCP/IP stack are automatically disabled for future vMotion operations. While vMotion tasks using the default stack may still complete successfully during transition, going forward, only the dedicated vMotion TCP/IP stack is used.

This setup ensures clean traffic separation, improved network efficiency, and easier troubleshooting in complex virtual environments.

The following figure illustrates the vSphere vMotion TCP/IP stacks:

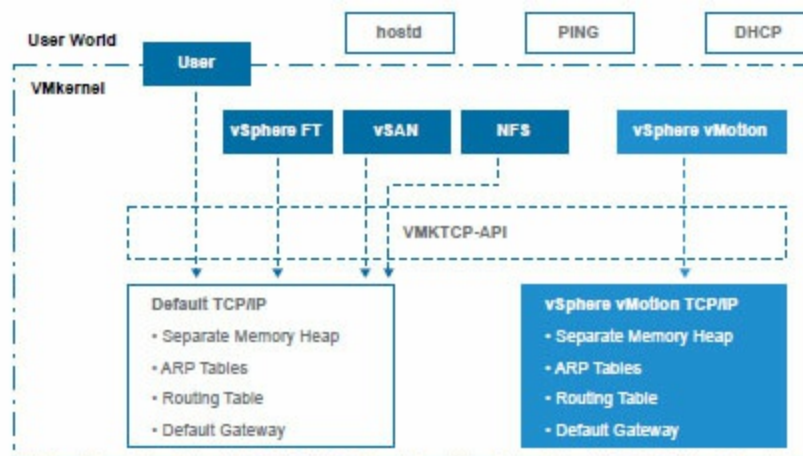


Figure 8.20: vSphere vMotion TCP/IP stacks

(Source: VMware)

Understanding Long Distance vSphere vMotion migration

Long-distance vSphere vMotion enhances the feature of cross vCenter migrations by allowing virtual machine migration between geographically

disparate data centers, even where network latency is considerably higher.

This capability opens opportunities for numerous applications, such as:

- Permanent VM relocation to another site
- Disaster avoidance during planned outages or threats
- Assessing disaster recovery processes with solutions like Site Recovery Manager.
- Equally distributing workloads across several locations
- Follow-the-sun support, where VMs are relocated to the region currently servicing operations

For example, in a follow-the-sun model, as one worldwide support team is closing their shift, the VMs they are servicing can move near the next team beginning their workday supporting local access and better performance. This versatility enables organizations to ensure business continuity, maximize the use of resources, and facilitate globally distributed operations.

The following figure illustrates the Long Distance vSphere vMotion migration:

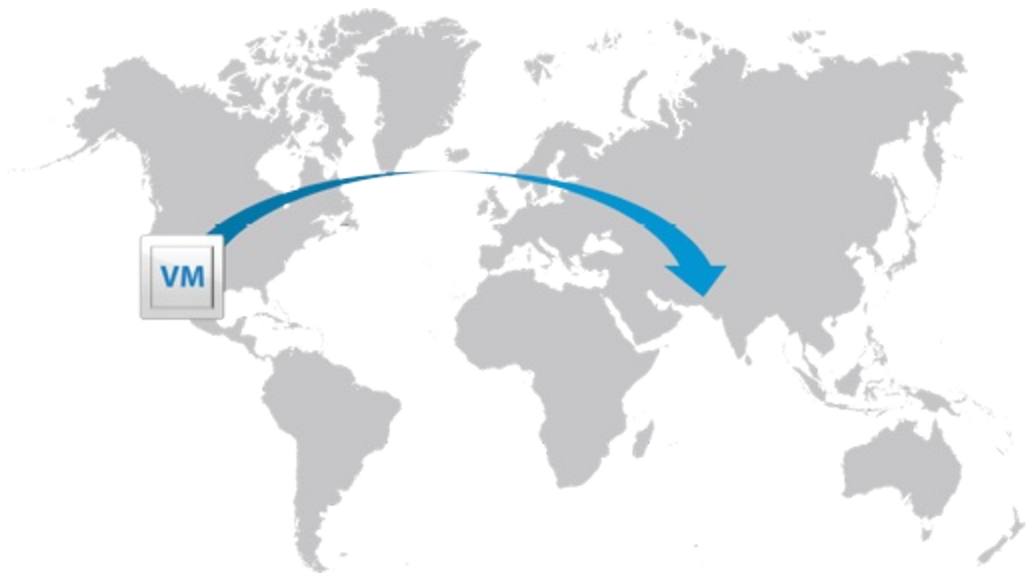


Figure 8.21: Long-distance vSphere vMotion migration

(Source: VMware)

Networking requirements for Long Distance vSphere vMotion

To support vSphere vMotion migrations over long distances, there are requirements that must be met on the VM and the vSphere vMotion network:

- **Virtual machine network:** It requires an L2 link to ensure the validity of the virtual machine's IP address at the target site. This requirement helps ensure the normal functioning of programs and services without the necessity of re-configuration.
- **vSphere vMotion network:** Functions on a *Layer 3 (L3)* connection and requires a minimum of 250 Mbps of bandwidth for each vMotion operation. The round-trip latency from source to destination hosts can be as high as 150 milliseconds and still provide migration safety over many geographic distances.

Snapshot management

VM snapshots are a helpful way to capture the current state of a virtual machine, allowing administrators to return to that exact point if needed. This is particularly useful during operations like software updates or patch installations, if something goes wrong, reverting to the previous snapshot can quickly restore the VM to a stable state.

However, snapshots are not designed to be a full backup solution. They should be used for short-term recovery, not long-term protection or archival.

Technically, snapshots are organized in a hierarchical *tree* structure. Each snapshot connects to a parent snapshot, forming a chain of restore points. The final snapshot in a branch does not have any children, making it the latest state in that line of progression.

The following figure illustrates the VM snapshots hierarchy:

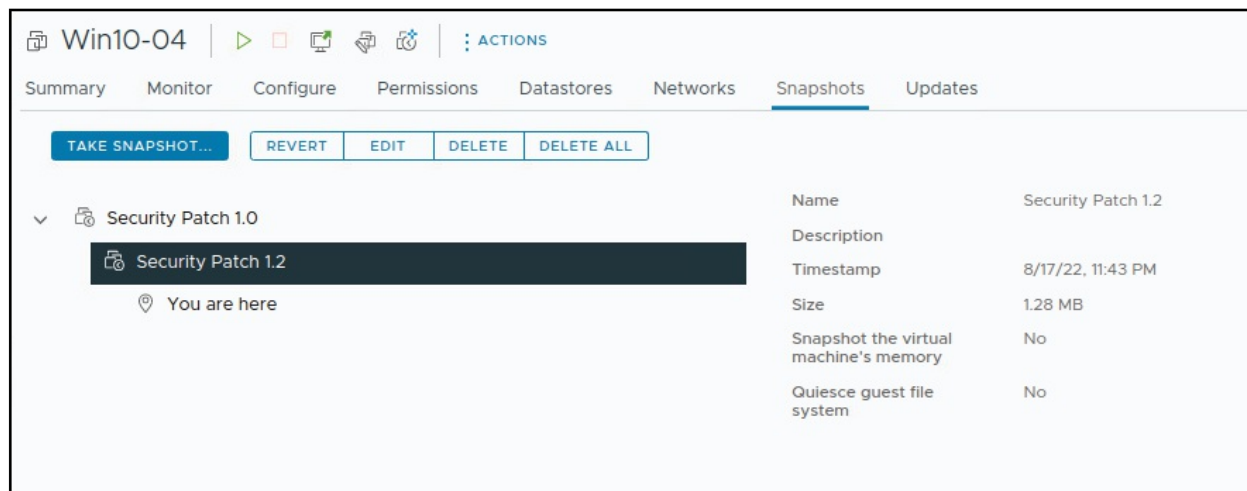


Figure 8.22: VM snapshots

(Source: VMware)

Taking snapshots

Snapshots can be taken when a VM is powered on, powered off, or suspended, depending on the scenario and what readers aim to capture. When taking a snapshot, VMware records a complete picture of the VM's state at that moment, including its settings, memory (optional), and disk data.

Here is what gets included in a snapshot:

- **VM configuration:** All the settings that define how the VM is set up.
- **Memory state (optional):** If the VM is powered on and administrators choose to include the memory state (which is selected by default), the snapshot will save everything in the VM's memory. This allows the VM to resume exactly where it left off after a revert.
- **Virtual disk state:** The exact condition of all the VM's virtual disks at the time of the snapshot.

However, it is worth noting that *independent virtual disks*, whether persistent or nonpersistent are not included in the snapshot.

When memory is not captured, readers have the option to *quiesce* the guest operating system. This process pauses or flushes disk activity, ensuring that the file system is in a consistent state before the snapshot is taken. This is particularly useful for applications that require data consistency.

The following figure illustrates how to take snapshot:

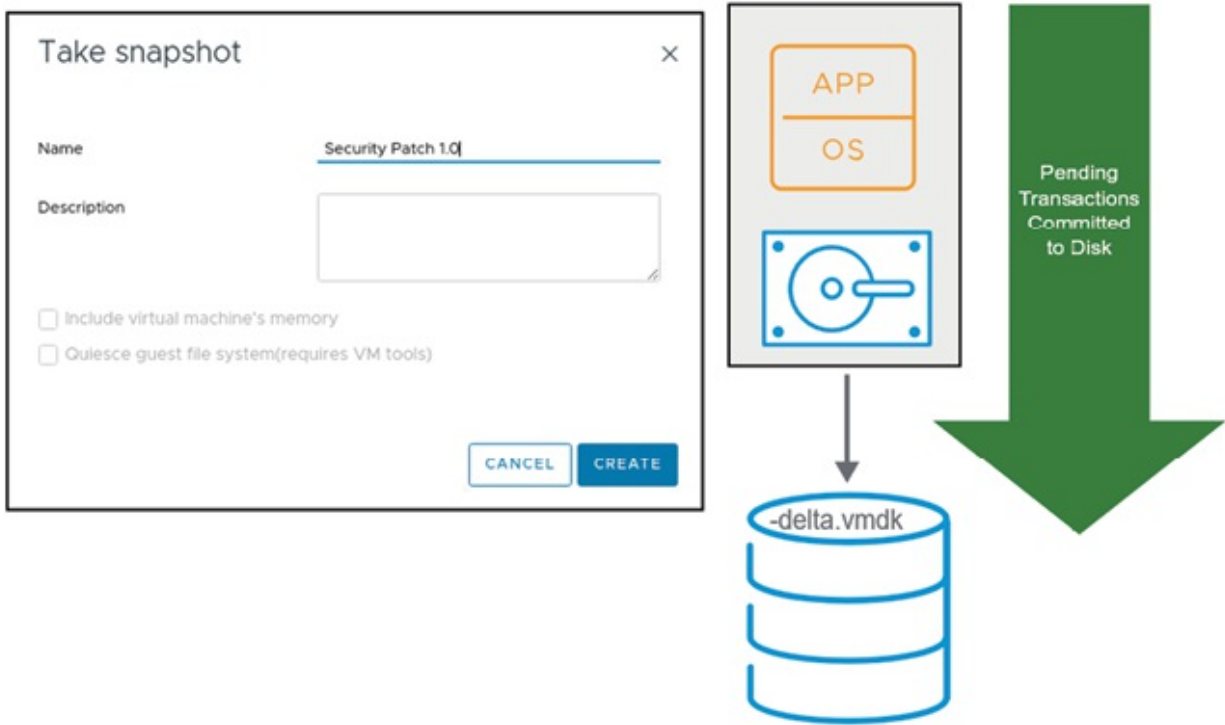


Figure 8.23: Taking snapshot

(Source: VMware)

Types of snapshots

When a snapshot is created in vSphere, a delta disk, also known as a child disk, is automatically generated. This delta disk records all changes made to the VM's virtual disk after the snapshot is taken, while preserving the original disk in its current state.

On the datastore, these delta disks are saved in a sparse format, which varies depending on the type of datastore in use. The sparse format determines how data is written and managed for the snapshot.

The following table outlines the different snapshot types based on the datastore format:

Snapshot type	Datastore type	Filename	Block size
VMFSsparse	VMFS5 with disks < 2 TB	#-delta.vmdk	512 bytes
SEsparse	VMFS6, or VMFS5 with disks \geq 2 TB	#-sesparse.vmdk	4 KB
	vSAN Original Storage Architecture		

vsanSparse	(OSA)	Delta object	4 MB
------------	-------	--------------	------

Table 8.1: Types of snapshots

Note: In environments using vSphere Virtual Volumes (vVols), snapshots are managed directly by the storage array and not through vSphere's traditional snapshot formats.

VM snapshot files

When snapshot is taken for a virtual machine, several files are produced to preserve the VM state. These files coexist to save the configuration state, memory, and disk state at the point that the snapshot operation is being performed.

The following is the list of snapshot files:

- **Delta disk file (-00000#-delta.vmdk or -00000#-sesparse.vmdk):** This file will hold changes of the VM's disk since snapshot creation. New writes are diverted here but not to the base disk (-flat.vmdk). With each change made, the size of the delta file increases.
- **Disk descriptor file (-00000#.vmdk):** A small metadata file that describes the corresponding delta disk file and specifies its dependency on the base disk.
- **Configuration state file (snapshot#.vmsn):** Stores the VM's power state and configuration at the snapshot creation time, including virtual hardware configuration and VM tools data.
- **Memory state file (snapshot#.vmem):** This is an optional file that captures the contents of the VM's RAM if the snapshot is created when the VM is running and the memory capture feature is on.
- **Snapshot metadata file (.vmsd):** Stores a record of all snapshots for the VM. It contains the names, descriptions, and parent/child relationships of a snapshot. It is generated when the VM is created and updated when snapshots are created or removed.

Snapshots taken with the memory state capture both. vmem and .vmsn files, making them larger in size but more inclusive in scope because they enable full *live* recovery to the exact working place of the virtual machine.

Let us take a closer look at how VM snapshot files evolve by examining three common scenarios: when a VM has no snapshot, one snapshot, and two

snapshots. The following figures illustrate how the associated files are created and grow with each snapshot.

The following figure illustrates the files of a VM with no snapshots:



Figure 8.24: VM snapshot files (1)

(Source: VMware)

The following figure illustrates the files of a VM with one snapshot with memory:



Figure 8.25: VM snapshot files (2)

(Source: VMware)

The following figure illustrates the files of a VM with second snapshot without memory:

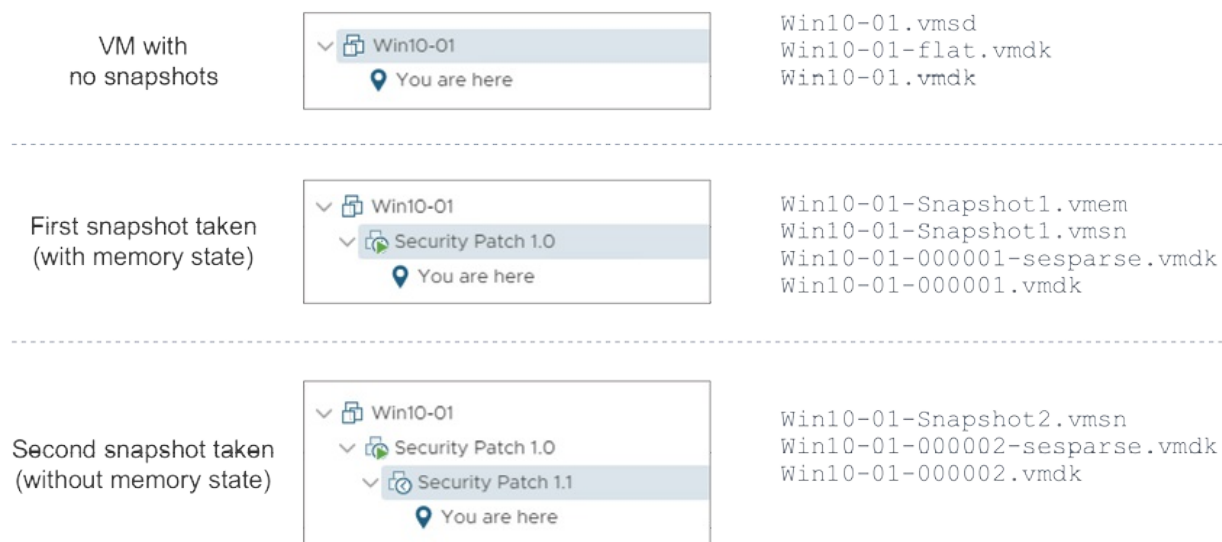


Figure 8.26: VM snapshot files (3)

(Source: VMware)

Managing snapshots

The **Manage Snapshots** pane in the vSphere client provides one location to track and manage snapshots for a given virtual machine. Administrators can utilize this pane to make significant changes or revert the virtual machine to a given point in time.

Available options are listed as follows:

- **Revert:** Roll the VM back to an earlier snapshot, returning the VM to the same power state and configuration it was in when the snapshot was made.
- **Edit:** Rename and update the name and details of an existing snapshot to enhance accuracy and improve documentation.
- **Delete:** Erase a given snapshot, merge its data into the parent drive, and merge the changes into the initial disk.
- **Delete All:** Publish all intermediate changes to the base disk and remove all the snapshots of the VM.

During snapshot revert, make sure the VM is in the desired state (powered on, off, or suspended) before the snapshot is taken, to pick up where it left off. Deleting or reverting snapshots initiates a consolidation operation, which combines the snapshot data with the base **virtual disk file (VMDK)** so that no changes are lost.

Appropriate snapshot management is necessary to ensure storage efficiency and VM stability in the long run.

The following figure illustrates the option for snapshot management:

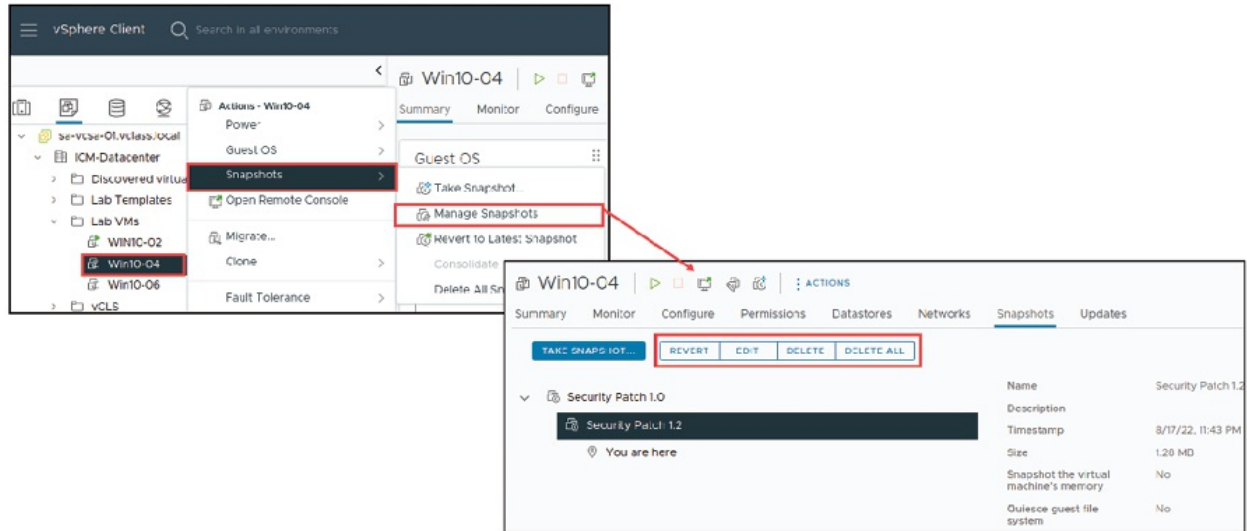


Figure 8.27: Managing snapshots

(Source: VMware)

VM snapshot deletion scenarios

As we deal with handling snapshots, understanding the mechanisms in place when we delete a snapshot is important. The effects may differ depending on the particular snapshot being deleted and its position in the snapshot hierarchy. We will discuss the details here:

- **Deleting a snapshot above the You are here level:** When an administrator removes an older snapshot (one or several levels above the current active level), its changes are written to the next available snapshot in sequence. For example, if snap01 is deleted, its content becomes part of the base disk, but the foundation for snap02 is preserved, ensuring continuity of the VM.
- **Deleting the latest snapshot:** The deletion of the latest snapshot merges its changes into the previous snapshot. In the case where snap02 is the latest, deleting it merges all of the changes into snap01, and the related delta file is removed.
- **Deleting snapshots below the You are here level:** When the previous

states are deleted in snapshots, i.e., a return to them has not yet occurred, these future states are irretrievably lost. This approach is helpful when removing unwanted or unnecessary snapshot branches, but it results in the loss of the ability to revert to such states.

- **Deleting all snapshots:** This operation effectively merges all modifications from the snapshots into the master disk. All snapshots prior to the present active state are checked, and any snapshots subsequent to the present point are deleted. This housekeeping operation not only makes it easy to manage snapshots but also assists in storage space recovery.

Understanding snapshot consolidation

Snapshot consolidation is a cleaning process applied when snapshot delta disk files remain in a VM's datastore directory, even though no snapshots are visible in the Snapshot Manager. This may occur if a deletion of a snapshot fails or goes halfway, leaving orphaned delta files (such as `-delta.vmdk` or `-sesparse.vmdk`).

Snapshot consolidation combines the data from all delta disks into the base disk and removes unnecessary snapshot files. It is critical to virtual machine performance, disk integrity, and avoiding the datastore from running out of storage space. It is a vital part of virtual machine maintenance, especially in environments where snapshot operations fail or are interrupted. When a mismatch occurs between a VM's snapshot descriptor file and the actual delta disk files, vCenter will display a warning under the **Monitor | All Issues** tab for that VM, indicating that *consolidation is required*. After the warning displays, administrators can use the vSphere Client to commit the snapshots.

The following figure illustrates when to consolidate snapshots:

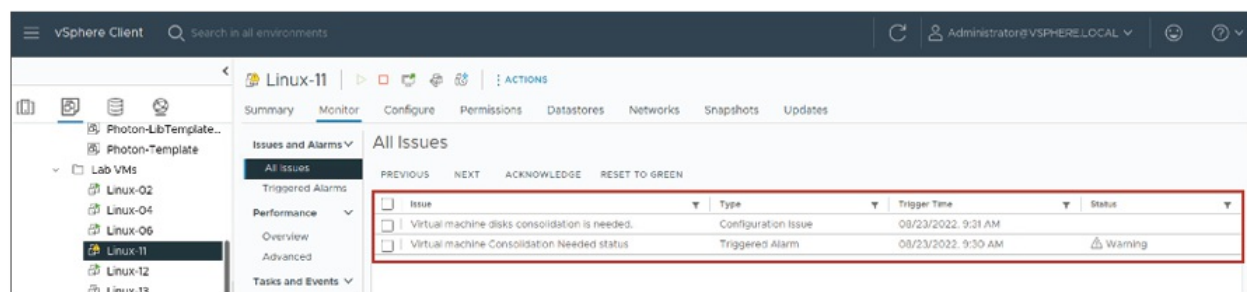


Figure 8.28: Discovering when to consolidate snapshot

(Source: VMware)

With the snapshot consolidation warning in the vSphere Client, administrators can initiate the consolidation from the console. This consolidates all the changes that are maintained in the snapshot delta disks (**-delta.vmdk** or **-sesparse.vmdk**) to the base virtual disk files.

Consolidation is a significant maintenance process to prevent datastore space from being used by unlinked or orphaned snapshot files, specifically when snapshots were deleted improperly or in the case of backup failures. It maintains the VM disk structure tidy and clean.

For VMware's official recommendations and best practices for the use of snapshots, see VMware KB Article 1025279:

<https://knowledge.broadcom.com/external/article?legacyId=1025279>.

The following figure illustrates how to consolidate snapshots:

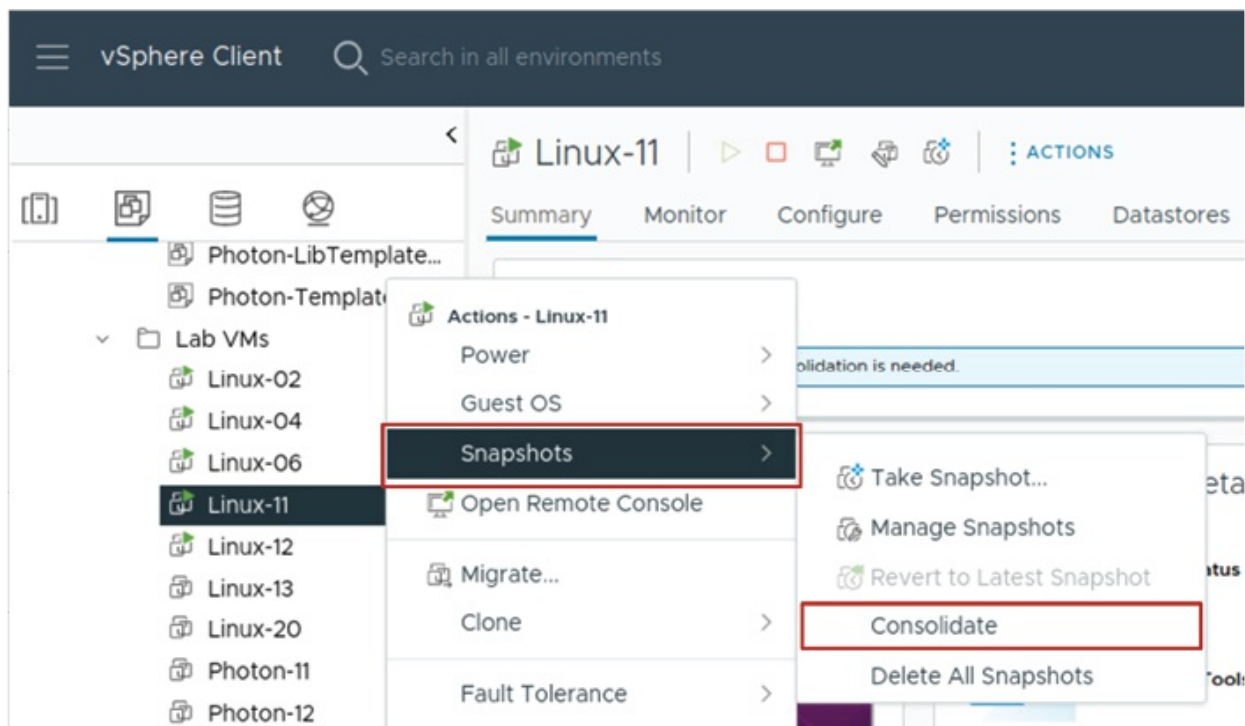


Figure 8.29: Consolidating snapshots

(Source: VMware)

By understanding these behaviors, readers can confidently manage VM snapshots and maintain a clean and efficient snapshot tree, ensuring that only meaningful restore points are kept while unnecessary clutter is removed.

CPU and memory concepts and considerations

Memory virtualization in a vSphere environment is a multi-layered operation where multiple layers collaborate to ensure efficient and isolated memory access to virtual machines:

- At the top level, applications in a VM communicate with guest OS virtual memory, just as they would on a physical machine.
- The physical memory of the guest OS is, in turn, made available by the VMkernel to the virtual machine.
- Finally, the host physical memory (the VMkernel-controlled one) includes the actual hardware layer.

When a VM runs, the VMkernel assigns a unique and contiguous block of addressable memory to it. This space simulates the physical memory topology of physical memory for the VM and enables several VMs to run concurrently without interfering with one another.

For ensuring security as well as efficiency, VMkernel implements an additional level of address translation where guest physical memory addresses are translated into host equivalent physical addresses. This process plays a very vital role in achieving memory isolation while not harming the performance of other virtual machines.

The following figure illustrates the basics of memory virtualization:

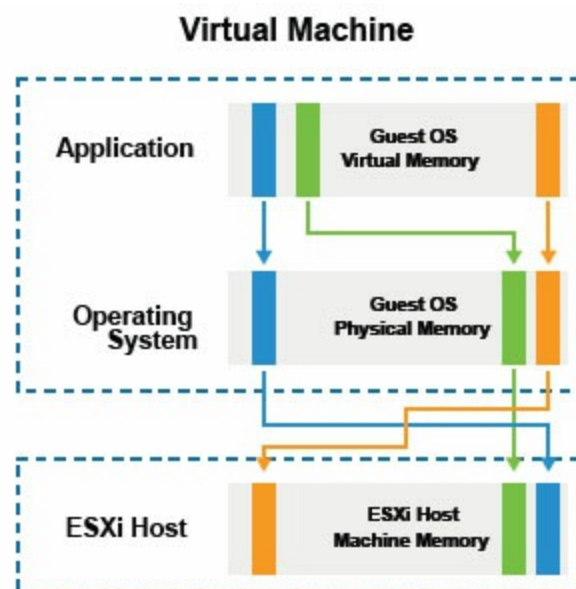


Figure 8.30: Memory virtualization

(Source: VMware)

VM memory overcommitment

Memory overcommitment happens when the aggregate memory committed to all running virtual machines is more than the physical memory available on the ESXi host. Overcommitment, however, does not necessarily lead to problems since most virtual machines do not always utilize their entire committed memory.

To solve this issue, ESXi uses several memory reclamation techniques. It can reclaim unused memory from idle VMs and reallocate it to those that are running low on memory. ESXi can also move VM memory to disk through swap files, for example, **.vswp** for the guest memory and **vmx-*.vswp** for the overhead memory consumed by the VM's executable process.

For instance, a host with 32 GB of RAM might have three VMs with each of them assigned 12 GB. Although this totals 36 GB total, if all VMs are not using their full memory, the system will still function fine. But if all VMs need their full allocation at the same time, then the host is overcommitted. In this case, even though existing VMs might still be running fine, new VMs will not be started due to insufficient memory.

Memory overcommitment is effective in dynamic systems where not all VMs are heavily utilized simultaneously.

The following figure illustrates the VM memory overcommitment:

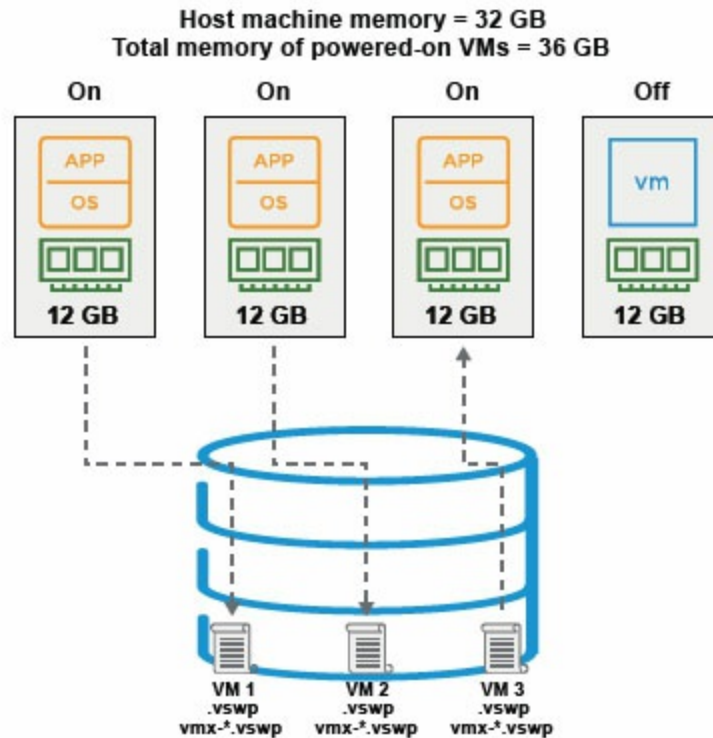


Figure 8.31: VM memory overcommitment

(Source: VMware)

Memory overcommits techniques

When memory requirement surpasses the physically available memory on an ESXi host, vSphere employs several intelligent methods to effectively optimize memory management without resorting to disk swapping, a process that can adversely affect performance, immediately. These methods are implemented in a particular sequence to recover memory with minimal disruption:

- **Transparent page sharing (TPS):** This is a method that identifies and keeps only distinct memory pages, thus removing redundant memory usage. While TPS continues to run in isolated VMs, the inter-VM page sharing functionality is turned off by default for security reasons starting with vSphere 6.0.
- **Ballooning:** It is because of the VMware Tools balloon driver, ESXi can ask VMs to release unused memory. This memory is reclaimed and can be utilized by other VMs. Ballooning works if the guest OS has sufficient swap space and can handle its own memory.

- **Memory compression:** When memory pressure continues to rise, ESXi compresses memory pages rather than swapping them out to disk. Since decompression of memory is faster than reading from disk, this feature is employed to maintain performance.
- **Host-level SSD swapping:** In case of insufficient compression, ESXi can also use locally attached SSD drives as a memory page cache for VMs. This provides faster access than traditional disk swapping and helps maintain the system responsive.
- **VM memory paging to disk:** VM swapping to disk is a last option. After all other mechanisms have been used up, ESXi resorts to using VMkernel swap space to swap out memory pages to disk. Although this action releases physical memory, it will severely impact performance.

By layering these methods in a strategic way, ESXi keeps VMs operating at peak levels, even under high-usage conditions and without losing as much performance and stability as possible.

Configuring multicore VMs

Configuring VMs with multiple **virtual CPUs (vCPUs)** for intensive workloads is typical in today's virtualized environment. The number of vCPUs that a VM can accommodate depends on several factors, including the physical CPU architecture of the ESXi host, the guest operating system's capabilities, the extent to which the applications in the VM are scalable, and how the VM is intended to be used.

Behind the scenes, the VMkernel possesses a high-level CPU scheduler that allocates these vCPUs to the host CPUs. It knows the overall hardware topology, such as sockets, cores, and threads. For instance, a CPU socket may have several cores, and each core may support several logical CPUs (or threads) due to technologies such as hyper-threading.

When a vCPU requires processing time, the VMkernel seeks out an idle logical processor on the host and allocates it. Scheduling assists in maximizing the host's CPU while evenly processing numerous multicore VMs. VMs are thus able to scale with new hardware and deliver performance not much different from physical systems yet take advantage of the efficiency and flexibility of virtualization.

The following figure illustrates the multicore virtual machines:

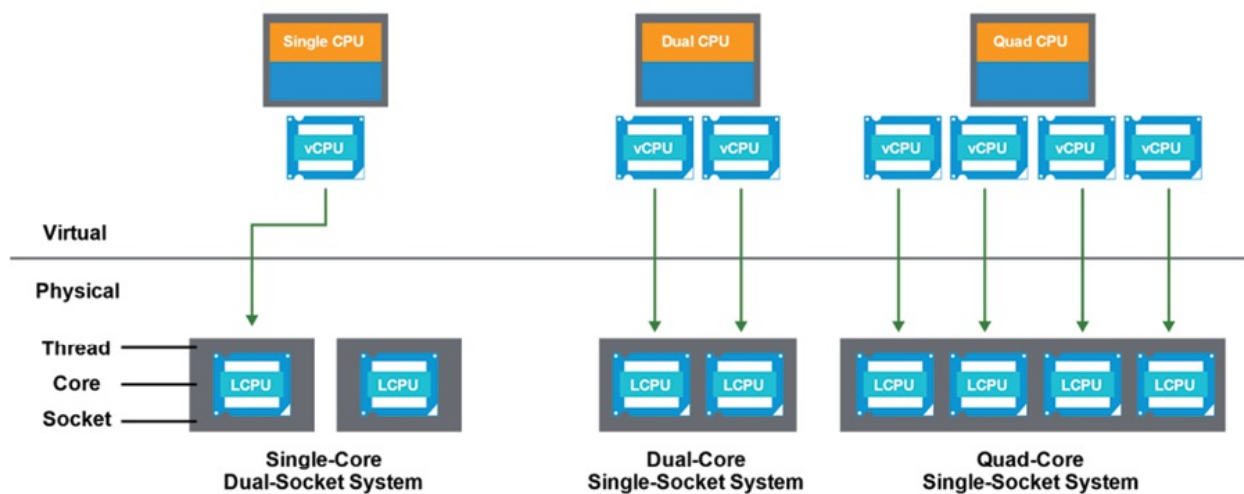


Figure 8.32: Configuring multicore VMs

(Source: VMware)

Understanding hyperthreading

Hyperthreading is a technology that enables a physical CPU core to run two threads concurrently. This effectively doubles the number of logical CPUs on a host, giving greater flexibility to vSphere when scheduling vCPU workloads.

Hyperthreading is automatically turned on in ESXi hosts if it is also turned on in the server BIOS. The feature may be referred to as *Hyperthreading* or *Logical Processor*, depending on the hardware manufacturer. After being turned on, the ESXi scheduler can schedule vCPUs on these logical CPUs, and in most situations, this results in better performance.

To check if hyperthreading is enabled or not, go to the Summary page of the host in the vSphere Client. Simply go to the Hardware category and click on CPUs to ensure the feature is active and enabled.

The following figure illustrates about the hyperthreading:

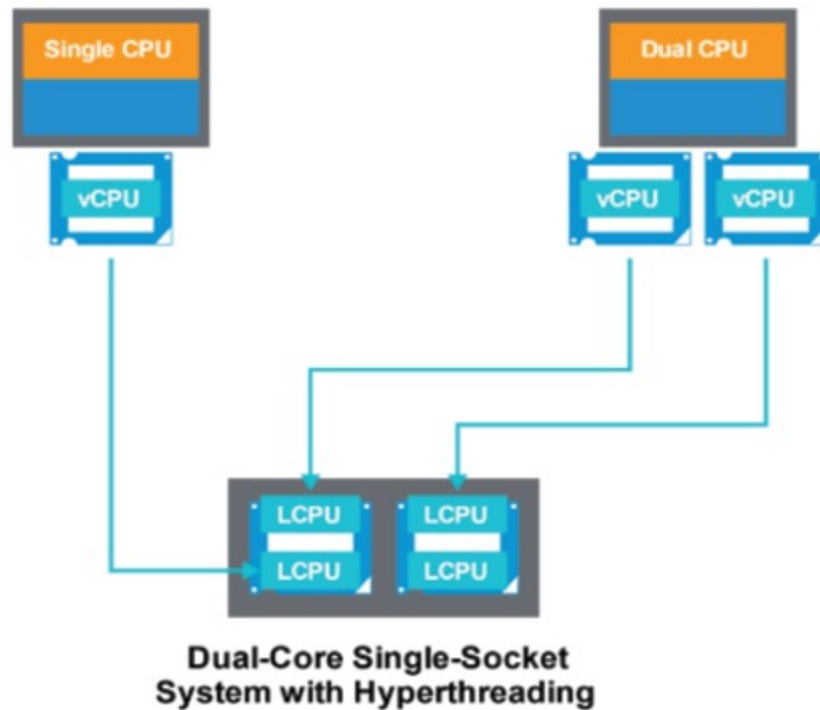


Figure 8.33: About hyperthreading

(Source: VMware)

CPU load balancing in ESXi

VMkernel is at the center of optimal CPU resource distribution among all processor cores within a host system. Its CPU scheduler is optimized to enable fair distribution of processing workloads, thereby minimizing bottlenecks, and maximizing overall performance.

Every few milliseconds, usually between 2 to 40 milliseconds depending on the host CPU architecture, the VMkernel examines the load and can migrate **virtual CPUs (vCPUs)** from one logical processor to another to distribute the load. Dynamic migration prevents CPU contention and keeps virtual machines running.

Wherever possible, the scheduler assigns multi-vCPU VM vCPUs to various physical cores, rather than to equivalent logical processors of the same core. However, when resources fall short, it can assign the vCPUs to the same physical core logical threads.

When a logical processor is not being used, it enters halt state. This enables the other thread on the same core to use the resources of the core to their

maximum capacity. The VMkernel observes this effect by re-calibrating the usage count of the resources involved accordingly, ensuring fair allocation and compliance with resource allocation policy.

The following figure illustrates the CPU load balancing:

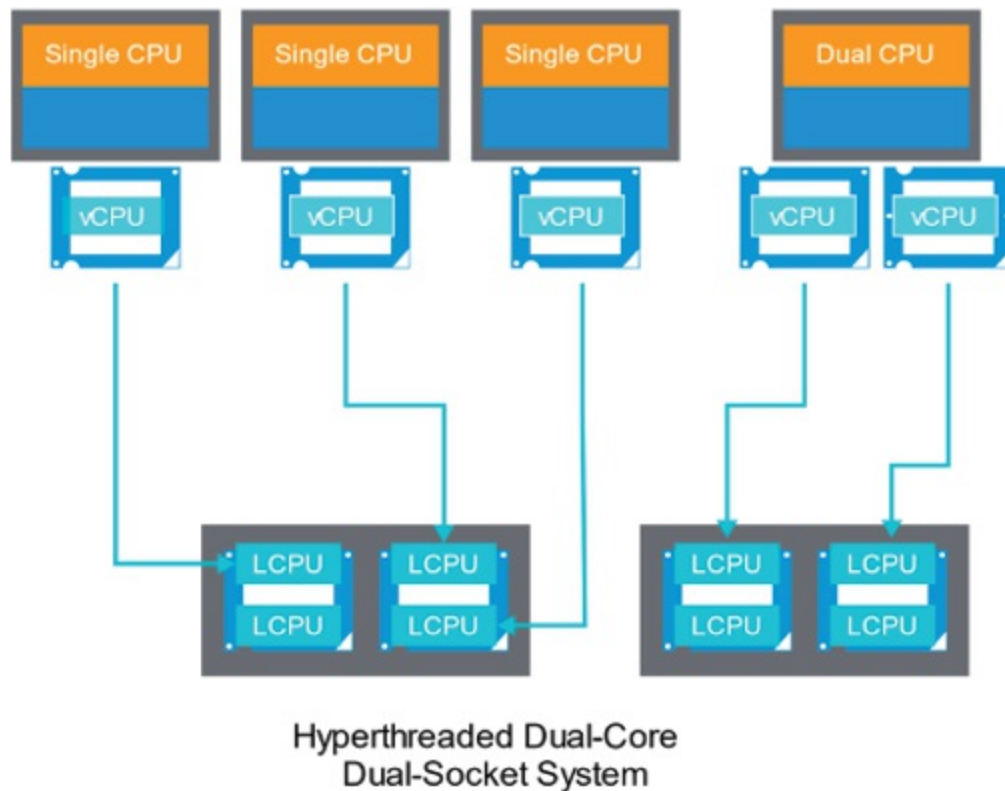


Figure 8.34: CPU load balancing

(Source: VMware)

VM resource allocation

In a virtualized environment, multiple VMs often compete for the same pool of CPU and memory resources on an ESXi host. To manage this shared usage efficiently and fairly, vSphere provides three key resource control mechanisms:

- **Reservations:** It guarantees a minimum amount of CPU or memory to a VM. This ensures that the VM always has the resources it needs to function properly, even under high system load.

- **Limits:** It set an upper boundary on how much of a resource a VM can consume, regardless of how much is available on the host.
- **Shares:** It determines a VM's priority relative to others when resources are overcommitted. VMs with higher shares receive more resources compared to others with lower shares when demand exceeds supply.

These settings come into play especially when resource contention occurs. For instance, if CPU or memory is overcommitted, the VM's actual allocation will fall somewhere between its reservation and limit, guided by the relative shares it holds. This dynamic balancing ensures that critical VMs get the performance they need while still maximizing overall efficiency across the host.

By using reservations, limits, and shares wisely, administrators can fine-tune performance, prioritize key workloads, and avoid resource contention in their virtual environment.

The following figure illustrates the limits, shares, and reservations:

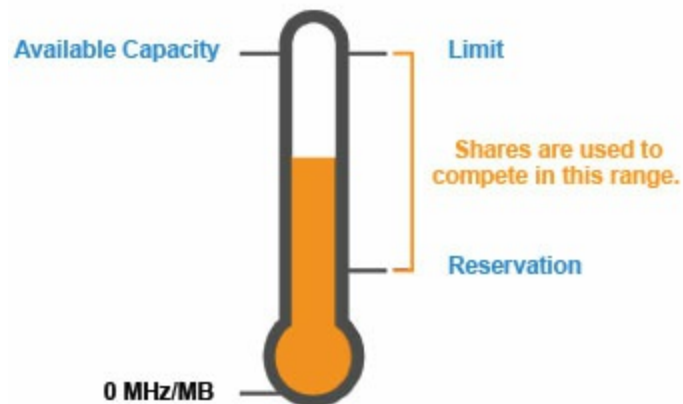


Figure 8.35: Limits, shares, and reservations

(Source: VMware)

RAM reservations for VMs

In a virtualized environment, memory can be overcommitted, meaning more memory is allocated to VMs than the host physically has. However, what if there is a critical VM that always needs consistent performance? That is where *memory reservations* come in.

A memory reservation guarantees that a VM will get a specific amount of

physical RAM from the host, no swapping, no ballooning. This means the reserved memory is always available to that VM, helping ensure low latency and high performance, even under heavy load.

Let us say there is a VM configured with 4 GB of memory, and administrators want to guarantee all of that. Administrators can simply set a reservation for 4 GB. This tells the host, *Make sure this VM always has 4 GB available - no matter what.*

One important thing to keep in mind that ESXi host must have not just the reserved memory available, but also enough *overhead memory* to power on the VM. For example, a VM with 4 GB of RAM and two vCPUs may require an additional 53 MB or so in overhead just to run.

If the host does not have enough unreserved memory to meet the reservation and overhead, the VM will not power on. So, while memory reservations are powerful, they should be used thoughtfully, especially when dealing with a host that is already heavily utilized.

The following figure illustrates the resource allocation reservations for RAM:

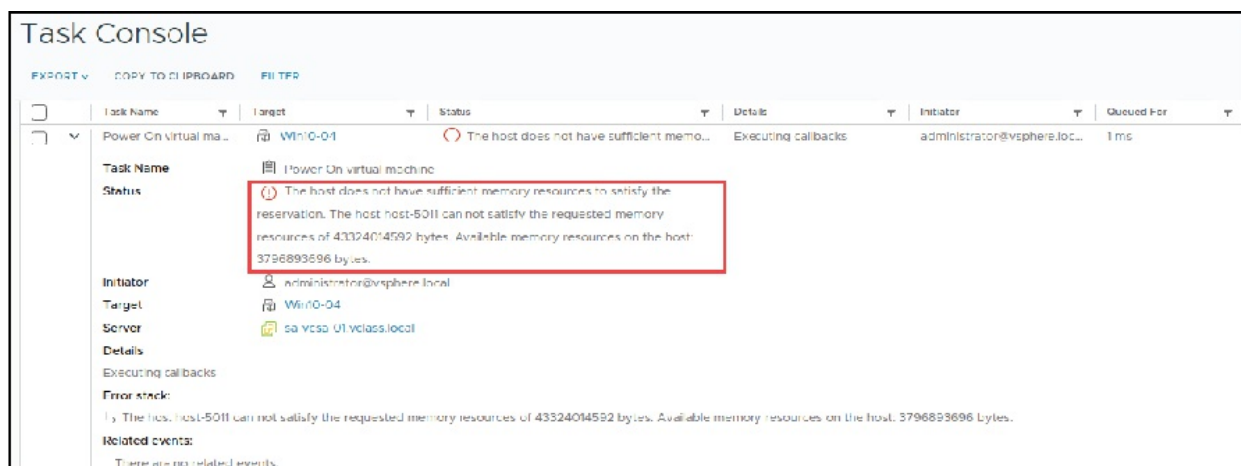


Figure 8.36: Resource allocation reservations for RAM

(Source: VMware)

CPU reservations for VMs

CPU reservations ensure a minimum amount of processing capacity (in MHz or GHz) is always reserved for a VM. The feature helps to overcome delays due to CPU contention and ensures predictability of performance, especially

for business-critical workloads. When there is a reservation of CPU, the VM has a guarantee of immediate access to physical CPU capacity, so it will never be in the ready state. However, if the host does not have sufficient unreserved CPU capacity, the VM will not boot.

CPU reservations are set to 0 MHz by default. It is recommended to utilize them sparingly on high-priority VMs that need guaranteed performance.

Controlling resource utilization

Resource constraints specify the maximum level of CPU or memory a VM can consume, independent of what is available on the host.

For example:

- For RAM, the VM will never use more physical memory than the limit. When the guest OS attempts to use more, the system swaps memory to the .vswp file.
- For CPU, the VM will not exceed the CPU limits, and any further processing demand will result in the threads of the VM being lined up in a ready state.

Though constraints are helpful in a controlled setting, such as resource simulation or initial user expectations, there are also disadvantages. When a host has spare capacity, such constraints still deny a virtual machine from accessing it, which may result in suboptimal utilization.

In general, it is more sensible to avoid putting restrictions unless there is a clear reason to do so.

The following figure illustrates the limits, shares, and reservations:

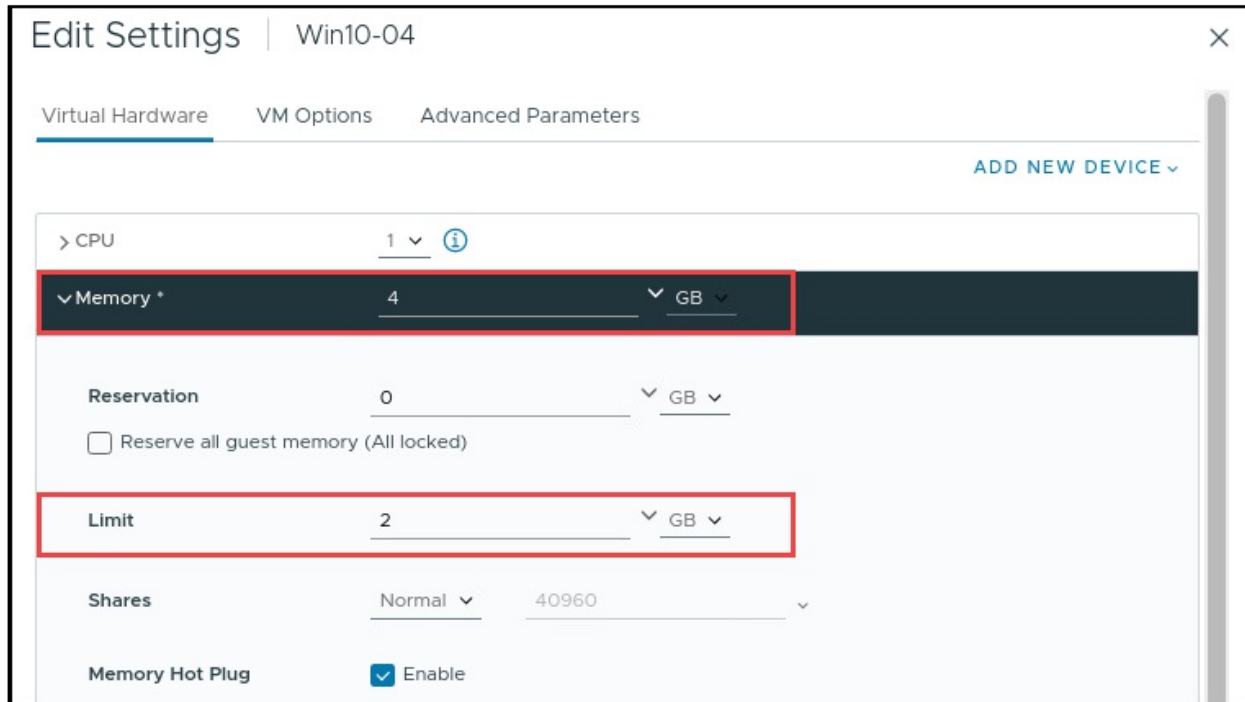


Figure 8.37: Resource allocation limits

(Source: VMware)

Resource allocation shares

Shares regulate how resources are distributed among VMs when there is a resource contention. A VM with more shares has a higher priority to access CPU or memory during resource contention.

For example, if one VM has twice the number of CPU shares as another, it will receive twice the CPU time when both are competing for resources. Shares levels can be assigned as high, normal, or low, or set a custom number to fine-tune resource prioritization.

Here is how the default settings work:

- **CPU shares:**
 - High = 2,000 shares per vCPU
 - Normal = 1,000 shares per vCPU
 - Low = 500 shares per vCPU
- **Memory shares:**
 - High = 20 shares per MB of memory

- Normal = 10 shares per MB
- Low = 5 shares per MB

These options follow a 4:2:1 ratio, giving administrators flexible control over VM resource priorities.

Shares come into play only when resources are contested. They define how much of a resource (such as CPU or memory) a VM gets compared to others.

Let us understand resource shared through an example:

- Administrators can adjust shares even while a VM is running, giving it more priority if needed as shown in the second row in [Figure 8.38](#).
- Adding a VM increases the total share pool, so each VM's portion adjusts automatically. For instance, consider the third row in [Figure 8.38](#): if a VM is powered on with 1,000 shares and the total becomes 6,000 shares across all VMs, that VM is entitled to one-sixth of the contested resource.
- If a VM is powered off or deleted as shown in the last row in [Figure 8.38](#), the remaining VMs get a larger slice of the resource pie.

The following figure illustrates the resource shares examples:

Number of shares	1,000 VM A	1,000 VM B	1,000 VM C	
Change Number of shares	1,000 VM A	3,000 VM B	1,000 VM C	
Power on virtual machine	1,000 VM A	3,000 VM B	1,000 VM C	1,000 VM D
Power off virtual machine	1,000 VM A	3,000 VM B		1,000 VM D

Figure 8.38: Resource shares examples

(Source: VMware)

Configuring resource allocation settings for a VM

VM resource allocation can be optimized by changing its CPU and memory configurations. For memory, administrator can choose to **Reserve all guest**

memory, which locks the full configured amount. If the VM's memory is later changed, the reservation adjusts automatically, ensuring all of it remains reserved without manual updates.

The following figure illustrates the resource allocation setting for a VM:

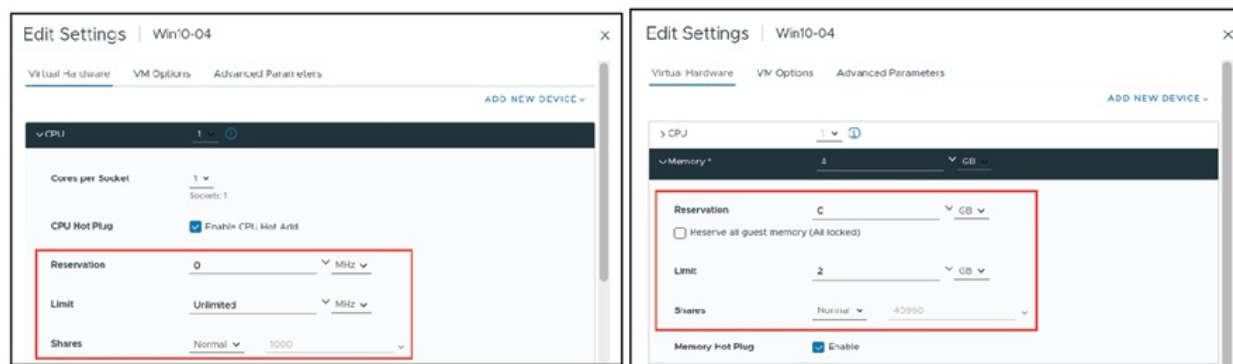


Figure 8.39: Defining resource allocation settings for a VM

(Source: VMware)

Administrators can easily view each VM's reservations, limits, and shares across the entire cluster to monitor and manage resource distribution effectively.

The following figure illustrates the view of VM resource allocation settings:

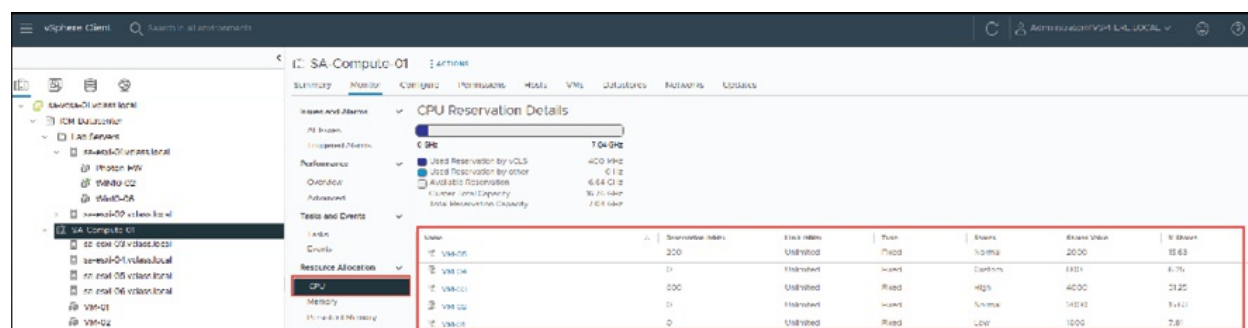


Figure 8.40: Viewing VM resource allocation settings

(Source: VMware)

Conclusion

Virtual machine management is at the heart of any virtualized infrastructure, enabling administrators to control efficiency, availability, and performance

for different workloads. The chapter provided a glimpse of the critical competencies involved in managing virtual machines in a vSphere environment, starting with conducting transparent live migrations with vSphere vMotion and Storage vMotion to using Enhanced vMotion Compatibility to enable support for cross-generational CPUs. These features enable continuity of operations, load balancing, and storage optimization while maintaining active workloads.

The readers were also informed about effective VM snapshot management, taking VM snapshots for system states, undoing changes, and understanding the underlying snapshot structure. The chapter explored memory and CPU-related concepts, such as resource reservation, limits, and shares, thus making readers more aware of VM-to-VM resource competition for fair and efficient allocation principles. With information on overcommitment methods, hyperthreading, and multicore scheduling, readers are now equipped to maximize VM performance in high numbers.

With these capabilities, readers are in a better position to manage VMs confidently in dynamic and challenging environments. This is succeeded by the next chapter, [Chapter 9](#), *vSphere Clusters Management*, which discusses the design and optimization of fault-tolerant clusters. Readers will be able to maximize vSphere DRS and HA to smartly load balance workloads and provide failover protection. From cluster design and health monitoring to fault tolerance implementation, the next chapter is centered on designing highly available, self-healing vSphere infrastructures for business continuity.

Points to remember

1. VM migrations can be compute-only, storage-only, or both, and can include cross-vCenter migrations.
2. vSphere vMotion is capable of live migrating the running VMs with zero downtime, but it requires an appropriate CPU as well as a network.
3. EVC offers secure migration through an official CPU baseline across hosts.
4. Snapshots are not backups and should not be kept for longer than 72 hours to prevent performance and storage problems.

5. Snapshot consolidation is necessary when delta files persist after being removed to avoid datastore bloat and free up space.
6. Overcommitment of memory is controlled through mechanisms like ballooning, page sharing, compression, and swapping.
7. CPU and memory resources can be managed with reservations, limits, and shares, particularly in overcommitted environments.

Exercises

1. What are the key differences between vSphere vMotion and vSphere Storage vMotion, and in what scenarios would each be used?
2. How does EVC ensure seamless VM migration across different CPU generations?
3. Why are snapshots not recommended as a long-term backup solution, and what are the risks of keeping them for extended periods?
4. Describe how memory overcommitment works in vSphere and list two techniques the VMkernel uses to manage overcommitted memory.
5. What is the purpose of setting CPU and memory reservations, limits, and shares for a virtual machine?
6. How does vSphere handle CPU load balancing across multiple virtual CPUs and physical cores?
7. What happens during a snapshot consolidation process, and when should it be performed?

Lab exercises

1. **Performing vSphere vMotion migrations:** Configure vSphere vMotion networking and migrate powered-on virtual machines between hosts with no downtime.
 - a. Configure vSphere vMotion networking on the source host:
 - i. Navigate to the host in the vSphere Client
 - ii. Go to *Configure | Networking | VMkernel adapters*
 - iii. Add a new VMkernel adapter and enable the vMotion traffic

checkbox

- iv. Assign a unique IP in the vMotion subnet
 - b. Configure vSphere vMotion networking on the destination host:
 - i. Repeat the steps above to configure the VMkernel adapter for vMotion on this host
 - c. Verify VM and host compatibility for vMotion:
 - i. Power on a VM on the source host
 - ii. Right-click the VM | *Migrate*
 - iii. Select “Change compute resource only.”
 - iv. Check compatibility; ensure no blocking errors
 - d. Migrate a running VM using vSphere vMotion:
 - i. Follow the Migrate wizard to move the VM to the target host
 - ii. Verify the VM continues running without interruption on the destination host
2. **Executing vSphere Storage vMotion migrations:** Migrate a VM's storage files from one datastore to another with no downtime.
- a. Initiate a storage migration:
 - i. In the vSphere Client, right-click a running VM | *Migrate*
 - ii. Select Change storage only
 - b. Select the target datastore:
 - i. Choose a destination datastore different from the current one
 - ii. (Optional) Change the disk format (e.g., thin to thick provisioning)
 - c. Complete the migration:
 - i. Proceed thro.M stays powered on during the operation
3. **Full compute and storage migration of a VM:** Relocate both the compute and storage of a VM using combined vMotion and Storage vMotion.
- a. Select the VM to Migrate:
 - i. Choose a powered-on VM with access to multiple hosts and datastores
 - b. Start the migration:

- i. Right-click VM | Migrate
 - ii. Select Change both compute resource and storage
 - c. Choose the target host and datastore:
 - i. Select a valid destination host and a different datastore
 - ii. Ensure compatibility is green-checked
 - d. Validate post-migration:
 - i. Confirm the VM runs as expected on the new host
 - ii. Verify that storage location and compute resource have both changed
- 4. **Managing VM snapshots:** Perform snapshot operations for backup, restore, and maintenance scenarios.
 - a. Take a snapshot:
 - i. Right-click a VM | Snapshots | Take Snapshot
 - ii. Provide a name and description
 - iii. Leave memory and quiesce options as needed
 - b. Create another snapshot after changes:
 - i. Make a visible change (e.g., create a file on the VM)
 - ii. Take another snapshot
 - c. Revert to a previous snapshot:
 - i. From the Snapshot Manager, select the first snapshot
 - ii. Click Revert To and confirm
 - d. Delete a specific snapshot:
 - i. From Snapshot Manager, delete the second snapshot
 - ii. Confirm that changes are committed
 - e. Delete all snapshots:
 - i. Select Delete All to merge all deltas back to the base disk
- 5. **Controlling VM resources using shares, limits, and reservations:** Observe how resource allocations affect VM performance during contention.
 - a. Create a contention scenario:
 - i. Deploy 2–3 VMs on a host with limited CPU resources

- ii. Run CPU-intensive workloads on each
- b. Modify CPU shares:
 - i. Right-click a VM | Edit Settings | VM Options | Resources
 - ii. Adjust CPU share levels (e.g., set one to High, others to Low)
- c. Set reservations and limits:
 - i. Apply a CPU reservation on one VM
 - ii. Apply a memory limit to another VM
 - iii. Observe performance in real-time from the Performance tab
- 6. **Snapshot consolidation and cleanup:** Consolidate orphaned delta files from snapshot operations.
 - a. Simulate snapshot sprawl:
 - i. Take multiple snapshots, then manually delete them from the disk
 - ii. Snapshot Manager may show no snapshots, but delta files remain
 - b. Trigger consolidation:
 - i. Right-click VM | Snapshots | Consolidate
 - ii. Confirm the action and monitor progress
 - c. Review datastore space:
 - i. Check the VM's datastore usage before and after consolidation
- 7. **(Optional): Enabling Enhanced vMotion Compatibility (EVC):** Configure EVC mode to enable cross-host migration in a CPU-heterogeneous environment.
 - a. Create a new cluster with EVC enabled:
 - i. Go to Hosts and Clusters | New Cluster
 - ii. Enable Enhanced vMotion Compatibility and choose appropriate CPU baseline
 - b. Add compatible hosts:
 - i. Add hosts with supported CPUs to the cluster
 - ii. Verify that EVC mode is active and hosts are compatible
 - c. Test VM migrations:
 - i. Deploy or migrate VMs within the EVC-enabled cluster
 - ii. Ensure that vMotion operations complete successfully

Join our Discord space

Join our Discord workspace for latest updates, offers, tech happenings around the world, new releases, and sessions with the authors:

[**https://discord.bpbonline.com**](https://discord.bpbonline.com)



CHAPTER 9

vSphere Clusters Management

Introduction

Today's organizations rely significantly on computer-based services such as email and databases to mission-critical web applications. Any disruption of these services will result in huge losses, impacting productivity as well as revenues.

vSphere clusters provide robust capabilities to ensure service continuity and performance. With vSphere **high availability (HA)**, clusters can be set up to reduce downtime and recover automatically in the event of host failure. This feature is essential to ensure continuous operation and competitive advantage. Additionally, the vSphere **Distributed Resource Scheduler (DRS)** optimizes resource utilization within the cluster. When well-configured, DRS ensures **virtual machines (VMs)** receive the computing resources they need, thus optimizing overall service quality and workload balancing.

Note: VMware is now part of Broadcom and is known as VMware by Broadcom. All references to VMware in this book reflect this change.

Structure

In this chapter, we will cover the following topics:

- vSphere cluster insights

- vSphere DRS overview
- vSphere HA overview
- vSphere HA designing and configuration
- Embracing vSphere Fault Tolerance

Objectives

By the end of this chapter, readers will be well-versed in managing and optimizing vSphere VMware clusters both for resource effectiveness and high availability. The chapter will guide readers through the building of a vSphere cluster and discuss the utilization of Cluster Quickstart to automate the configuration process. The chapter will guide readers through identifying important cluster features such as the DRS and HA, and explain how the technologies support workload and service availability. Step-by-step, readers will witness the vSphere DRS approach used to achieve the best VM placement based on resource needs, as well as how to set up and monitor DRS behaviour in real-world environments. In addition, the chapter will explore vSphere HA, its reactions to all kinds of failure types, i.e., host failure, VM failure, and network isolation incidents. Readers will learn the HA mechanisms used, such as heartbeats and isolation responses, and will discover how to create fault-tolerant cluster environments. Finally, this chapter will introduce vSphere Fault Tolerance as a mechanism of uninterrupted operation for mission-critical workloads. Through mastery of these tools and topics, readers can build and maintain reliable, efficient, and highly available virtual infrastructure clusters.

vSphere cluster insights

Within the framework of VMware vSphere, a cluster is an abstract group of **ESXi hosts** that provides management of pooled resources and guarantees high availability. As hosts are added to a cluster, vCenter Server combines their individual compute and memory capacities into one large resource pool. Abstraction makes it easier to manage processes and allows advanced functions such as load balancing, automatic resource allocation, and

protection from failure.

Clusters can be created based on their assigned roles in the virtual world. Some of the common uses are as follows:

- **Management clusters:** Used to manage infrastructure services such as vCenter Server and monitoring tools
- **Production clusters:** Designed to support business-critical workloads
- **Compute clusters:** For high-performance computing or general resource-intensive use

All vSphere clusters are scalable up to 96 ESXi hosts, provided the environment is at vSphere 7 Update 1 or newer. This type of high scalability provides flexibility for small-scale and enterprise-scale deployments.

The following figure illustrates the vSphere clusters:

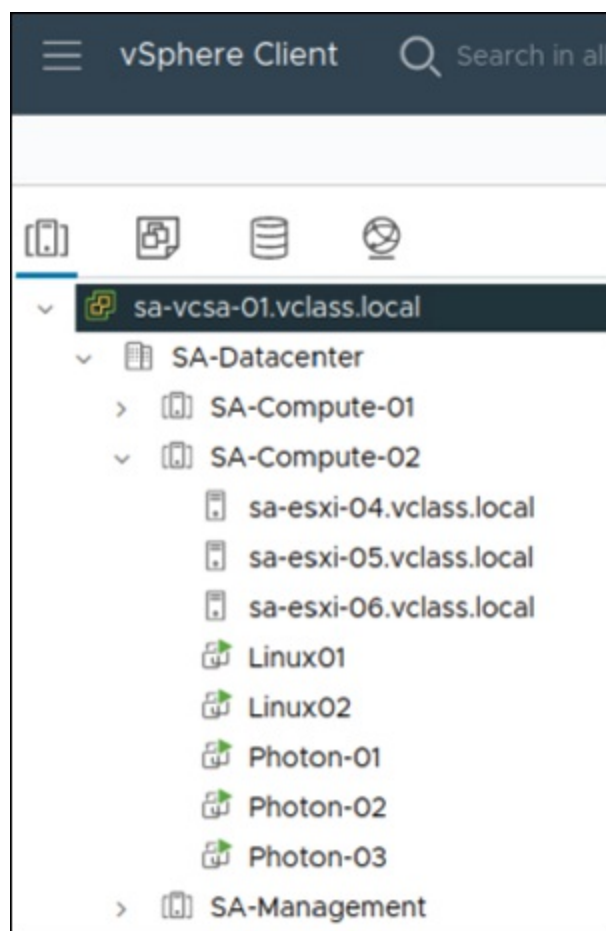


Figure 9.1: About vSphere clusters

Source: VMware

Creating a vSphere cluster

Creating a vSphere cluster starts with providing a name to the cluster and enabling the services required to sustain the environment. The cluster is implemented with multiple ESXi hosts to streamline the management of resources by pooling them together.

The following list outlines a few core services that can be configured during the creation of the cluster:

- The high availability feature of vSphere (vSphere HA) enables workload availability by providing the required restarting of VM resources on alternate hosts in the event of a failure.
- To ensure optimal performance, vSphere DRS automates workload balancing across hosts and optimizes resource utilization.
- Shared storage is made possible by **Virtual Storage Area Network (vSAN)** through the aggregation of local disks from every host in the cluster into a single software-defined datastore.

For ease of ongoing maintenance, vSphere Lifecycle Manager can be utilized. It streamlines the update and patching process by enabling the administrator to update all cluster hosts at once through a single ESXi image. This adds reliability and consistency to the update process.

The following figure illustrates how to create a vSphere cluster:

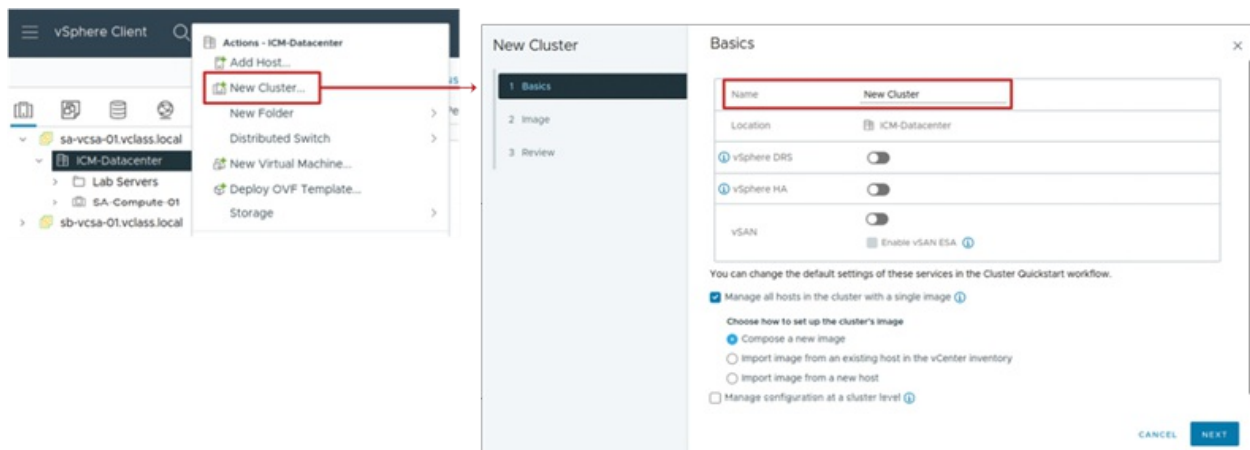


Figure 9.2: Creating vSphere clusters

Source: VMware

Cluster Quickstart overview

Upon cluster creation, the subsequent step in the workflow offers a way to configure the cluster, efficiently customized to specific business needs. This setup wizard guides users through the initial configuration steps to ensure that all critical services and settings are implemented correctly across the entire environment.

The Cluster Quickstart is designed to omit auxiliary configurations in favour of guiding users through core task deployment, thereby increasing the ease and speed of completing a deployment by clustering essential steps into one workflow. It includes service enablement and verification of service readiness, as well as networking parameters and inter-service policies configurations.

Among other key steps, the following are included in the Cluster Quickstart workflow:

- Activating essential services: vSphere HA, vSphere DRS, and vSAN.
- Validating hardware and software compatibility across the hosts.
- Deploying and configuring vSphere Distributed Switches.
- Configuring vSphere vMotion and vSAN traffic networks.
- Creating vSAN Stretched Clusters or vSAN Fault Domains to increase availability.
- Configuring the **Network Time Protocol (NTP)** service for all hosts in the cluster to ensure uniformity in time reference.

Cluster Quickstart also aids in scaling clusters, as it allows new hosts to be integrated into the cluster at a faster pace from a configuration standpoint by seamlessly synchronizing them with the current configuration of the cluster's existing members.

For more information about creating and managing clusters, consult the *vCenter Server and Host Management* documentation at <https://techdocs.broadcom.com/>.

The following figure illustrates the Cluster Quickstart:

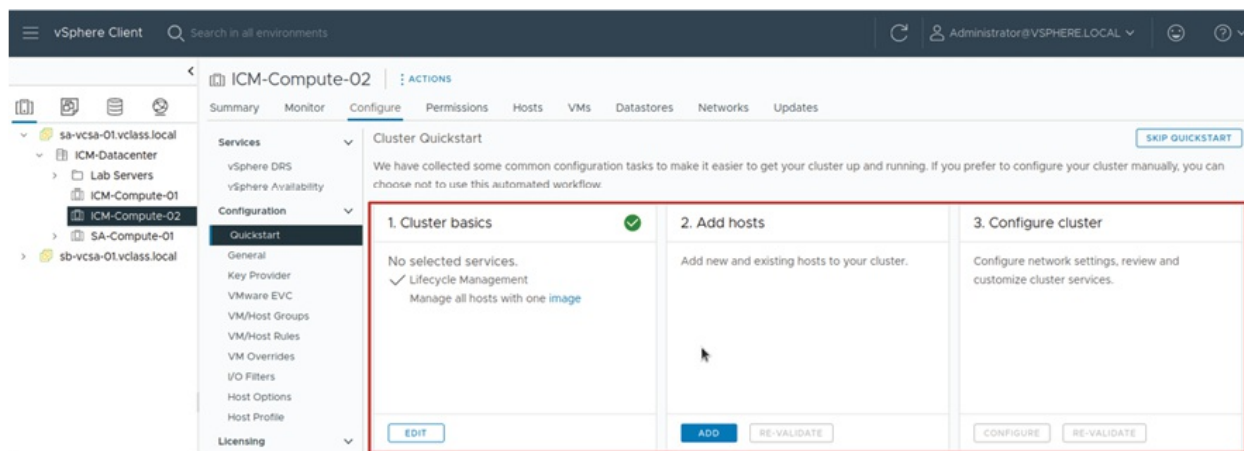


Figure 9.3: About Cluster Quickstart

Source: VMware

Cluster Quickstart for activating services

The first step in the **Cluster Quickstart** workflow is selecting and confirming the services that will be enabled for the cluster. This foundational step ensures that the cluster is prepared to deliver the functionality needed for high availability, resource distribution, and storage management.

Within the Cluster Basics pane, the following options are available:

- *Edit the cluster name* to reflect its role or purpose clearly.
- Enable or disable key services, including:
 - *vSphere DRS* for automated workload balancing
 - *vSphere HA* for VM availability during host failures
 - *vSAN* for integrated, software-defined storage
- Choose an image for vSphere Lifecycle Manager, which allows centralized lifecycle management and ensures that all ESXi hosts within the cluster stay consistent and compliant.

This step lays the groundwork for building a robust and flexible vSphere cluster environment.

The following figure illustrates the Cluster Quickstart wizard:

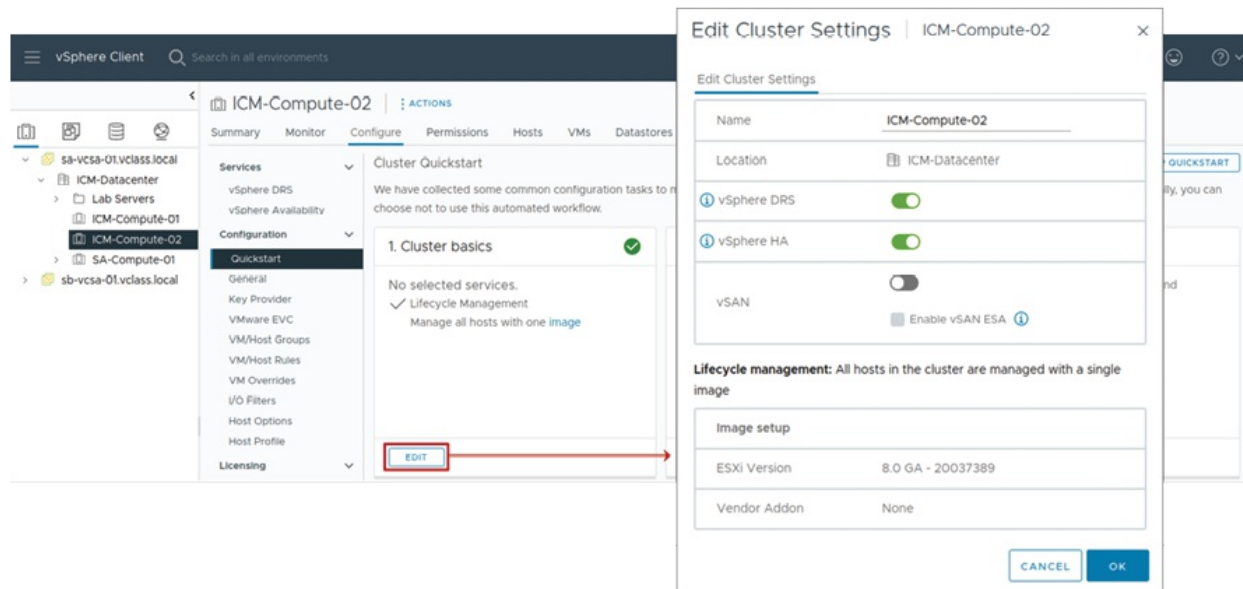


Figure 9.4: Cluster Quickstart step 1, activate service

(Source: VMware)

Cluster Quickstart for adding hosts

The next stage in the **Cluster Quickstart** workflow involves adding ESXi hosts to the cluster.

As part of the **Add Hosts** pane, the following two options are available:

- **New hosts:** This feature enables administrators to conveniently add hosts to the vCenter inventory and cluster simultaneously. Administrators can add a host by specifying either its **Internet Protocol (IP)** address or **Fully Qualified Domain Name (FQDN)**.
- **Existing hosts:** This option is applicable when the hosts are already available in the inventory but need to be linked to the cluster.

After the hosts are added, the workflow is refreshed to show the full count of hosts across the cluster. Additionally, it performs a health check validation, which ensures that all required compatibility criteria of the hosts are met before proceeding.

The following figure illustrates adding hosts:

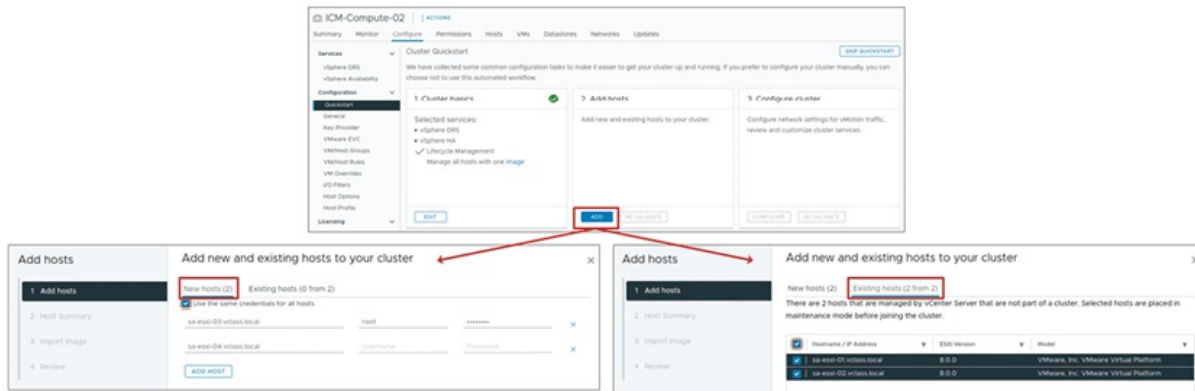


Figure 9.5: Cluster Quickstart step 2, add hosts

Source: VMware

Cluster Quickstart for configuring the cluster

The final configuration step of the Cluster Quickstart workflow focuses on customizing host networking configuration and adjusting cluster services settings.

This task allows administrators to configure the integration of vSphere components, such as vMotion, vSAN, and management traffic, from the respective physical network interface card on each host in the cluster. It ensures reliable connectivity within the cluster environment.

Remember, once the **Skip Quickstart** is clicked, it will leave the guided setup. Everything after that must be done manually from different menus in the vSphere Client. That workflow cannot be reinitiated for the same cluster, which means any additional hosts that are added later will require manual configuration and will not have automatic configuration.

The following figure illustrates the cluster configuration:

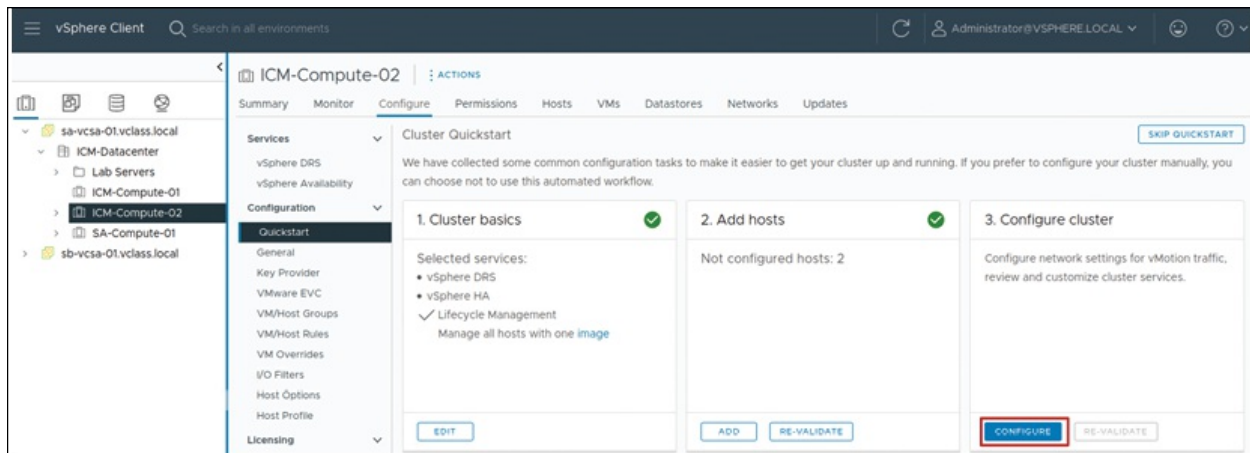


Figure 9.6: Cluster Quickstart step 3, configure cluster

(Source: VMware)

Setting up a cluster for distributed switches

With respect to the configuration of clusters, the role of Cluster Quickstart in networking (especially for vMotion and vSAN) is critical. The automation features provided by vSphere help administrators simplify this setup through **vSphere Distributed Switches (VDS)**.

In this step, decision-making is focused mainly on the switch configuration, and the specific tasks are to:

- Pick a maximum of three distributed switches for the cluster.
- Select a network for the vSphere vMotion, which provides support for the remote migration of VMs.
- Assign at least one Physical Adapter (Network Interface Card) for the cluster, which guarantees transfer access to the cluster.

There are also configuration options like postponing the setup of detailed network configuration. Checking the **Configure networking settings later** option allows users to skip all advanced networking steps. The configuration applies only the default configurations required for cluster services. It is important to note that after selecting this option, there is no possibility to change the network configuration in the **Configure cluster** wizard. Any subsequent changes will have to be made using the vSphere Client independently.

The following figure illustrates the cluster DVS configuration:

Configure cluster

1 Distributed switches
2 vMotion traffic
3 Advanced options
4 Review

Distributed switches

Configure the distributed switches

☐ Configure networking settings later ⓘ

Distributed switches

Number of distributed switches: 1 ⓘ

Configure the following distributed switches, based on the port group and uplink options you select on this page. There may be additional port groups created if existing VM networks are migrated to these switches. VMkernel adapters for management network will be migrated with the physical adapters assigned to distributed switches.

Name	Port groups	Uplinks
DSwitch	USE EXISTING	1

Port groups

The following default port groups will be assigned to the distributed switch.

vMotion network: DSwitch
DSwitch-vMotion

Physical adapters

One uplink port group will be created on each switch containing all the specified physical adapters.

CANCEL NEXT

Figure 9.7: Cluster Quickstart step 3.1, configure cluster DVS

(Source: VMware)

Setting up a cluster for vSAN and vMotion traffic

Additional network configuration is necessary for specific cluster services during the Cluster Quickstart process, as outlined:

- The *vSphere DRS* option will have the workflow ask for an address for the vMotion network, which is needed for vMotion to occur seamlessly.
- The *vSAN* checkbox will have the wizard ask for the vSAN IP configuration for the vSAN network, which is used for storage traffic among ESXi hosts.

If the hosts have the same subnet settings, the cluster can be set up more quickly. In such cases, the **AUTOFILL** option is available to set up the cluster in record time. Provide the correct network parameters for the first host and then enable the autofill option for the other hosts in the cluster.

The following figure illustrates the cluster vMotion configuration:

Configure cluster

- 1 Distributed switches
- 2 vMotion traffic**
- 3 Advanced options
- 4 Review

vMotion traffic

Specify the IP addresses for the vMotion traffic

Distributed switch: DSwitch

Distributed port group name: DSwitch-vMotion

☒ Use VLAN: 12

Protocol: IPv4

IPv4 configuration

IP type: Static IPs

Each host is configured automatically based on the input below. Empty gateway might result in a segmented network.

Host	IP Address	Subnet Mask	Gateway
sa-esxi-01.vclass.l...	172.20.12.51	255.255.255.0	Gateway
sa-esxi-02.vclass.l...	172.20.12.52	255.255.255.0	Gateway

AUTOFILL

CANCEL BACK NEXT

Figure 9.8: Cluster Quickstart step 3.2, configure cluster vMotion

Source: VMware

Setting up a cluster for advanced features

As the last step in the Cluster Quickstart workflow, additional configuration settings become available with the selected services on the cluster.

The selected adjustments could include the following:

- **HA:** Grouping optional customizations that manage the handling of host failures and VM recovery within the cluster.
- **DRS:** Customization of VM load balancing and automation tier, if any, is defined in these optional settings.
- **Host options:** Lockdown mode and NTP server can be configured in participating hosts in the cluster.
- **Enhanced vMotion Compatibility (EVC):** Settings governing uniform migration of VMs across diverse processor features on different hosts and ensuring flawless vMotion are enabled.

These advanced features grant users the ability to tailor operational procedures of a cluster, improve overall functionality, and ensure consistent

performance within mixed hardware environments.

The following figure illustrates the cluster advance configuration:

The screenshot shows the 'Configure cluster' wizard in the vSphere Client. The left sidebar lists four steps: 1 Distributed switches, 2 vMotion traffic, 3 Advanced options (selected), and 4 Review. The main panel is titled 'Advanced options' and contains the instruction 'Customize the cluster settings.' Below this, there are expandable sections: 'vSphere HA', 'vSphere DRS', 'Host Options' (which is expanded), and 'Enhanced vMotion Compatibility'. The 'Host Options' section includes a 'Lockdown mode' dropdown set to 'Disabled' and an 'NTP server' field with a placeholder 'Optional IP Address or FQDN' and a note 'Separate servers with commas, e.g. 10.31.212, fe00:2800'. At the bottom right, there are 'CANCEL', 'BACK', and 'NEXT' buttons.

Figure 9.9: Cluster Quickstart step 3.3, configure cluster advanced options

Source: VMware

Cluster summary information

The summary tab in the vSphere Client provides a consolidated snapshot of the cluster's current state. It provides critical information about the cluster's resources, such as CPU, memory, and storage, as well as how VMs and services use them.

The following figure illustrates the cluster summary information:

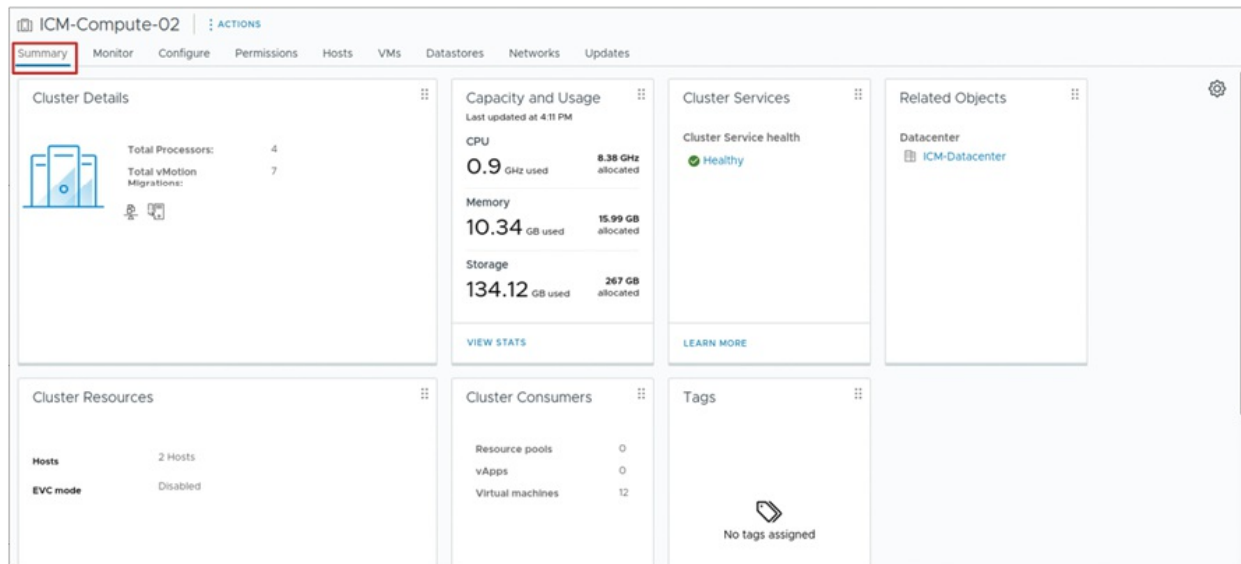


Figure 9.10: Viewing cluster summary information

(Source: VMware)

Observing cluster resources

Using vSphere, one can view in-depth how resources within a cluster are allocated and used. It is possible for the administrator to track the memory and CPU usage at the cluster level, which helps ensure that resources are optimally utilized and distributed.

The platform, under the memory statistics section, provides an exhaustive overview that contains the following:

- The total memory and CPU capacity of each host that is part of the cluster.
- The memory overhead for virtualization.
- The overall storage is still available for use.
- The capacity reserved for use by VMs.
- The overall unused free capacity that is available for use.

This level of detail provides the necessary information to enable proper decisions around the placement of workloads, scaling, and general cluster health management.

The following figure illustrates the cluster resources:

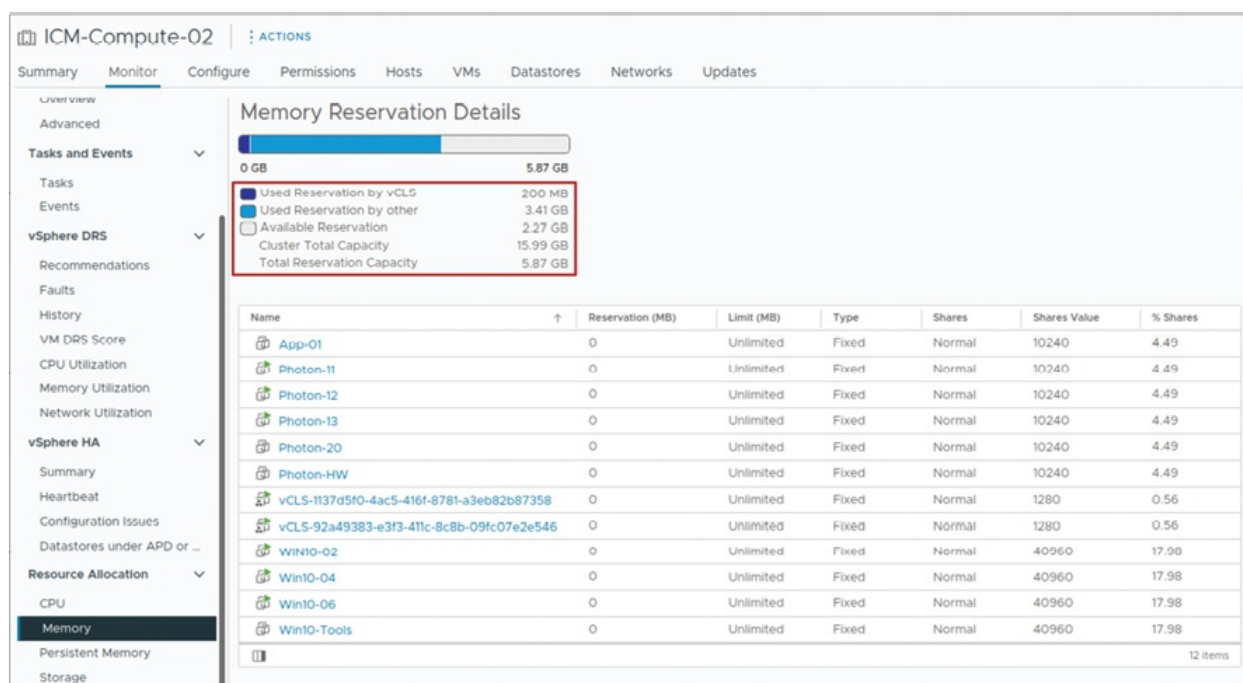


Figure 9.11: Monitoring cluster resources

(Source: VMware)

vSphere Cluster Services VMs

All vSphere clusters contain up to three **vSphere Cluster Services (vCLS)** VMs, which are essential for maintaining the operational health of core cluster features such as vSphere DRS and vSphere HA.

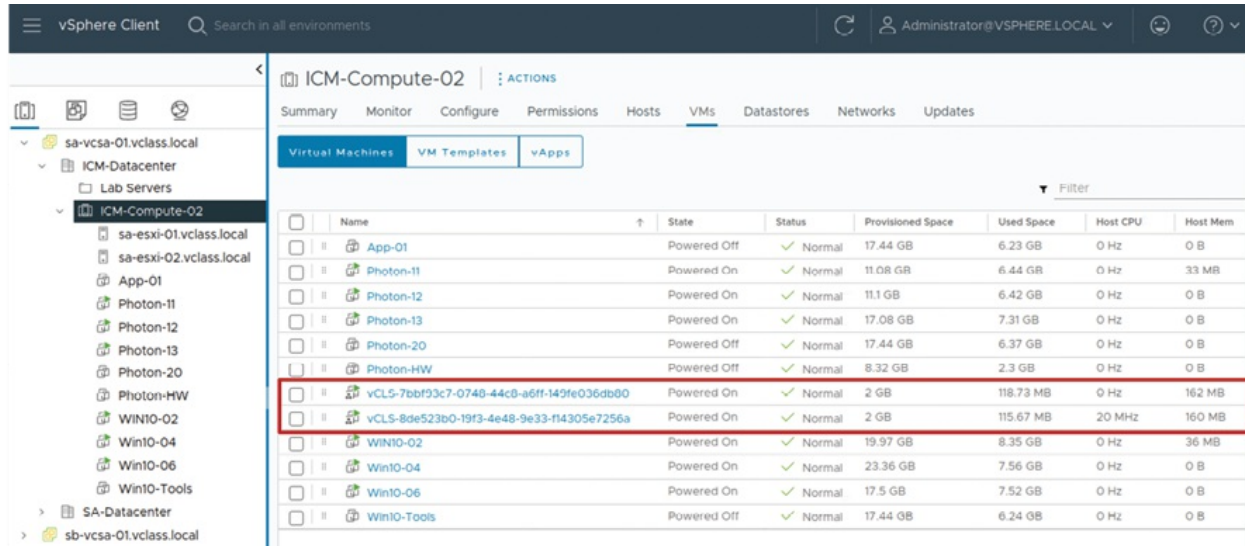
With the addition of new hosts or the creation of new clusters, these lightweight VMs are automatically provisioned. Each vCLS VM runs under a Photon **Operating System (OS)** profile and is provisioned from an OVA. A vCenter component, the vCLS manager, manages these VMs' lifecycle, including power and placement, availability, and overall resource provisioning.

Do not modify the resources or power state of these VMs, because doing so will prevent the cluster from sustaining its health, thus stopping vital operations like DRS from functioning properly.

These VMs are not shown in the **Hosts** and **Clusters** inventory view, but they can be found in the VMs section of the cluster and the VMs and Templates inventory. The cluster will issue a vSphere Client alert notification when these VMs become unhealthy or are deactivated. vCenter will automatically

attempt to activate these VMs, overriding any manual shutdowns to ensure service continuity.

The following figure illustrates the vCLS VMs:



Name	State	Status	Provisioned Space	Used Space	Host CPU	Host Mem
App-01	Powered Off	✓ Normal	17.44 GB	6.23 GB	0 Hz	0 B
Photon-11	Powered On	✓ Normal	11.08 GB	6.44 GB	0 Hz	33 MB
Photon-12	Powered On	✓ Normal	11.1 GB	6.42 GB	0 Hz	0 B
Photon-13	Powered On	✓ Normal	17.08 GB	7.31 GB	0 Hz	0 B
Photon-20	Powered Off	✓ Normal	17.44 GB	6.37 GB	0 Hz	0 B
Photon-HW	Powered Off	✓ Normal	8.32 GB	2.3 GB	0 Hz	0 B
vCLS-7bbf93c7-0748-44c8-a6ff-149fe036db00	Powered On	✓ Normal	2 GB	118.73 MB	0 Hz	162 MB
vCLS-8de523b0-19f3-4e48-9e33-f4305e7256a	Powered On	✓ Normal	2 GB	115.67 MB	20 MHz	160 MB
Win10-02	Powered On	✓ Normal	19.97 GB	8.35 GB	0 Hz	36 MB
Win10-04	Powered On	✓ Normal	23.36 GB	7.56 GB	0 Hz	0 B
Win10-06	Powered On	✓ Normal	17.5 GB	7.52 GB	0 Hz	0 B
Win10-Tools	Powered Off	✓ Normal	17.44 GB	6.24 GB	0 Hz	0 B

Figure 9.12: vCLS VMs

(Source: VMware)

vSphere DRS overview

The vSphere **Distributed Resource Scheduler (DRS)** is geared towards maximizing resource allocation efficiency for all hosts in a vSphere cluster. It ensures that all VMs are allocated the required CPU and memory resources as workload balancing occurs.

Some common uses for DRS include the following:

- Initial VM placement while powering on
- Load balancing across hosts
- VM migration when a host is in maintenance mode

DRS takes into consideration the entire computational capabilities of the hosts in a cluster as one unified resource pool. At the time of powering on a VM, DRS checks for the host whose resources can best serve the VM's requirements, details as follows:

- In fully automated mode, DRS places the VM on the best-optimized host

without any user input.

- In partially automated or manual mode, DRS provides suggestions to the user for validation or endorsement.

DRS continuously monitors the cluster for balance and performance and runs vSphere vMotion when necessary to shift VMs from or to hosts as needed. Balance and performance checks are continuous; thus, when a host is placed in maintenance mode, DRS automation (with VM control spun off) will migrate the VMs automatically, while offering suggestions with automation turned off.

The following figure illustrates the vSphere DRS:

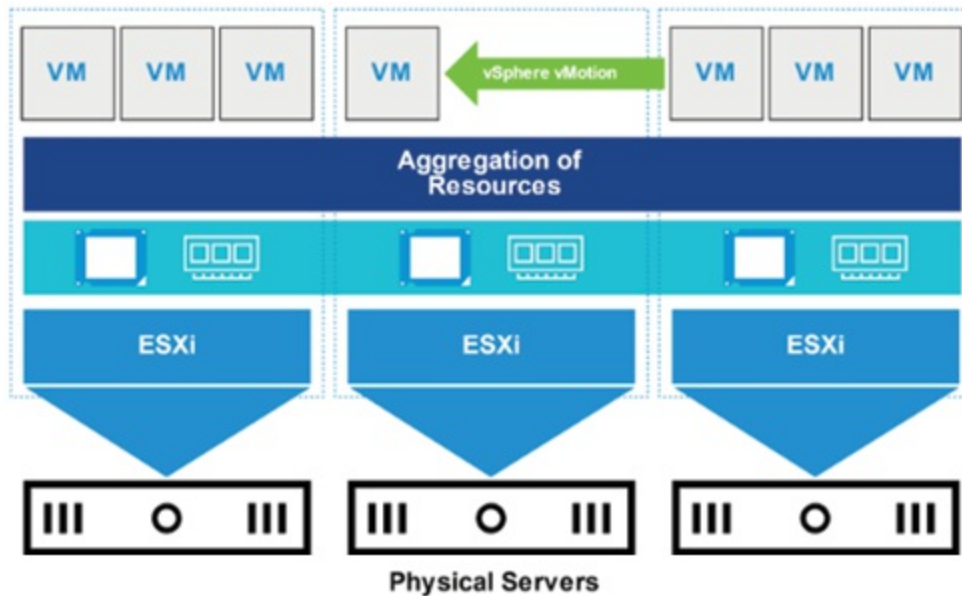


Figure 9.13: About vSphere DRS

(Source: VMware)

vSphere DRS with a VM-centric approach

vSphere DRS takes a VM-centric approach for coordinating the resources within the cluster. Instead of viewing hosts from a high level as an aggregated unit, DRS drills down to each VM level in the hierarchy and ensures it gets the resources required for its smooth functioning.

The following list outlines the way the process works:

- While a VM is powered on, DRS continuously monitors its performance

and scores resource (such as CPU and memory) consumption through a defined threshold for given resources:

- According to these scores, DRS makes recommendations or, in fully automated mode, automatically moves VMs to more appropriately suited hosts during migration cycles.
- The decisions are made every minute to ensure that workloads are perfectly balanced across the cluster, on the hosts.

Grasping the VM DRS score helps in understanding what factors are affecting a virtual or physical machine's load and how efficiently resources are distributed for the machine's workload. In general, the greater resource availability, the better the performance of operations performed by the VM, and the higher the score the VM gets.

Rightfully, if the score is near to 100%, the VM is in a much better state when it comes to the usage of CPU, memory, and network resources. On the flip side, if the score is closer to 0%, it means the VM is poor, which is not a desirable outcome. Not only does the VM execute poorly and face fierce competition for resources, but it also faces contention for them, and it will not work efficiently.

This score is determined from live interval measurements of the VM. There is no estimation, only real data obtained from the CPU, memory, and network activity.

In *vSphere 7* and later, DRS recalculates this score *every minute*. It also provides a **Cluster DRS Score**, which reflects the overall *health* of the cluster in terms of resource distribution. This score falls into one of five buckets:

- 0–20%
- 20–40%
- 40–60%
- 60–80%
- 80–100%

These buckets help administrators quickly assess how well the cluster is supporting all the VMs within it.

The following figure illustrates the VM DRS score:

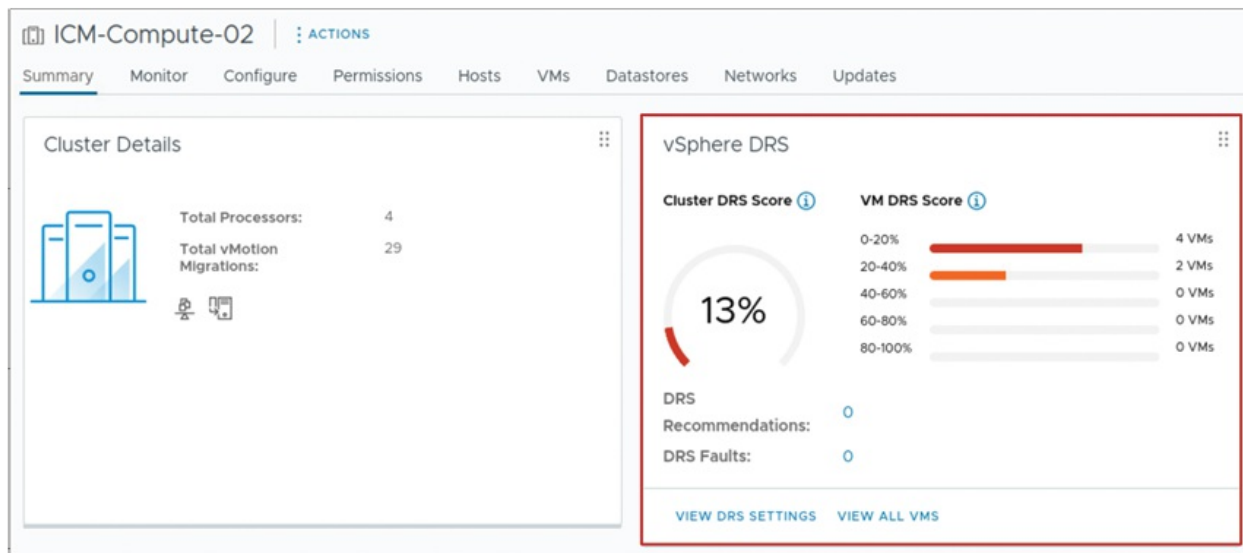


Figure 9.14: About the VM DRS score

Source: VMware

Viewing VM DRS scores in the monitor tab

To monitor VM performance within the cluster, navigate to the **Monitor** tab of the cluster in the vSphere Client. Review the *VM DRS Score* under vSphere DRS, which provides insight into how effectively each VM's resource requirements are being satisfied.

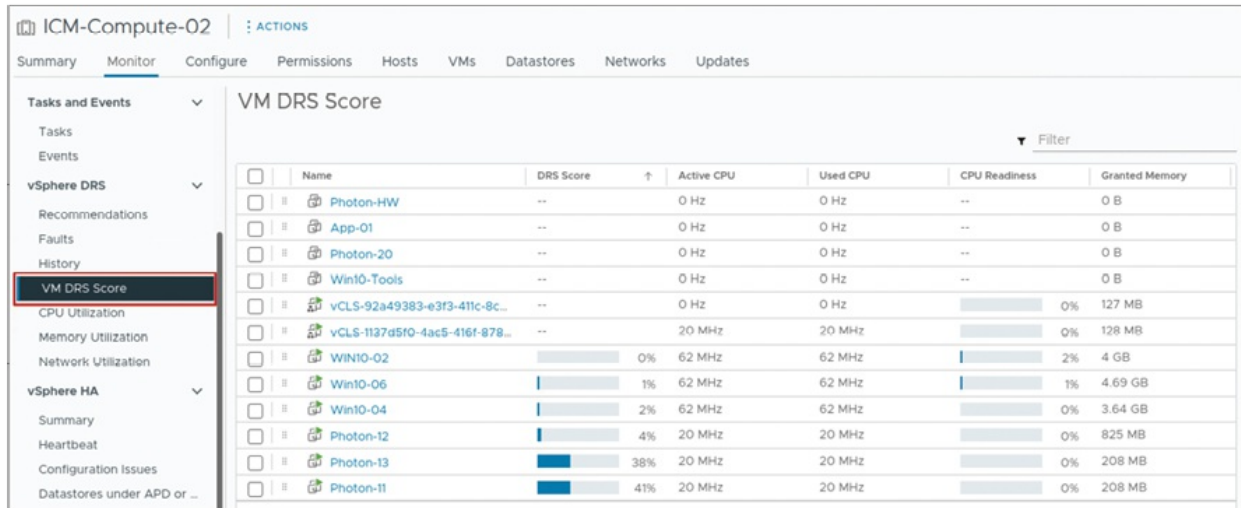
For *powered-on VMs*, the *VM DRS Score page* shows the following key metrics:

- **DRS score:** Indicates how well the VM is performing based on its current access to resources.
- **Active CPU:** The percentage of CPU actively being used by the VM.
- **Used CPU:** The total amount of CPU cycles consumed by the VM.
- **CPU readiness:** How long the VM had to wait to access CPU resources (lower is better).
- **Granted memory:** The amount of memory the VM has been allocated.
- **Swapped memory:** The amount of memory moved to disk due to memory pressure.
- **Ballooned memory:** Memory reclaimed from the VM by the balloon driver (usually during resource contention).

These metrics provide clarity on whether a VM is thriving or struggling and

can help administrators make better decisions about resource allocation and load balancing.

The following figure illustrates the VM DRS score list:



Name	DRS Score	Active CPU	Used CPU	CPU Readiness	Granted Memory
Photon-HW	--	0 Hz	0 Hz	--	0 B
App-01	--	0 Hz	0 Hz	--	0 B
Photon-20	--	0 Hz	0 Hz	--	0 B
Win10-Tools	--	0 Hz	0 Hz	--	0 B
vCLS-92a49383-e3f3-411c-8c...	--	0 Hz	0 Hz	0%	127 MB
vCLS-1137d5f0-4ac5-416f-878...	--	20 MHz	20 MHz	0%	128 MB
Win10-02	0%	62 MHz	62 MHz	2%	4 GB
Win10-06	1%	62 MHz	62 MHz	1%	4.69 GB
Win10-04	2%	62 MHz	62 MHz	0%	3.64 GB
Photon-12	4%	20 MHz	20 MHz	0%	825 MB
Photon-13	38%	20 MHz	20 MHz	0%	208 MB
Photon-11	41%	20 MHz	20 MHz	0%	208 MB

Figure 9.15: VM DRS score list

Source: VMware

Requirements for a vSphere DRS cluster

There are several prerequisites to successful load balancing and VM migrations on vSphere DRS clusters. First, to add ESXi hosts to a vSphere DRS cluster, the following requirements must be fulfilled:

- **vMotion network:** All hosts in the cluster should be part of a vMotion network for vSphere DRS to perform live migrations (vSphere vMotion). Clusters lacking a vMotion network cannot automatically balance VMs during execution, although they can suggest initial placement.
- **Compatibility for vSphere vMotion:** To allow seamless migration of VMs between hosts using DRS without any downtime, the prerequisites of vSphere vMotion must be met.
- **Storage sharing:** Incorporate all ESXi hosts under a cluster shared storage. This can either be vSAN, **Network File System (NFS)**, or **Internet Small Computer System Interface (iSCSI)**. This permits workloads, in this situation the VMs, to be used interchangeably with any ESXi host within a cluster mounted in shared mode.

Adhering to these simple principles permits vSphere DRS to execute load balancing across the servers dynamically while monitoring resource consumption, thereby maintaining high availability, optimal performance for VMs, and ensuring the seamless operation of the cluster.

vSphere DRS settings for automation levels

In vSphere DRS, the automation level determines the level of control a user has over host selection for VMs, VM workload balancing, and dynamic resource allocation. This configuration can be set for both the initial scenario when a VM is initialized and for subsequent dynamic balancing scenarios.

The automation levels are as follows:

- **Manual:**
 - When a VM is powered on:
 - vSphere DRS provides a list of suitable hosts for selection.
 - **During runtime:**
 - Proposed actions for VM migrations to enhance cluster balance are considered, but no automatic actions are implemented.
 - Any changes driven by automation must be manually implemented.
- **Partially automated:**
 - When a VM is powered on:
 - The best-fitting host selection is automatically performed by DRS for optimization.
 - **During runtime:**
 - Recommended migrations must be approved before implementation.
- **Fully automated:**
 - When a VM is powered on:

- In the best suitable host, the machine will automatically be placed virtually.
- **During runtime:**
 - Proactive physical VM migrations are performed to maintain balance and performance requirements.

The most hands-off experience will be the one given when *fully automated* is selected. In contrast to other options, the fully automated mode will provide maximum relief for the operation. In production environments that require critical decision-making support and reliable performance, fully automated mode optimizes operations.

The following figure illustrates the automation level in vSphere DRS settings:

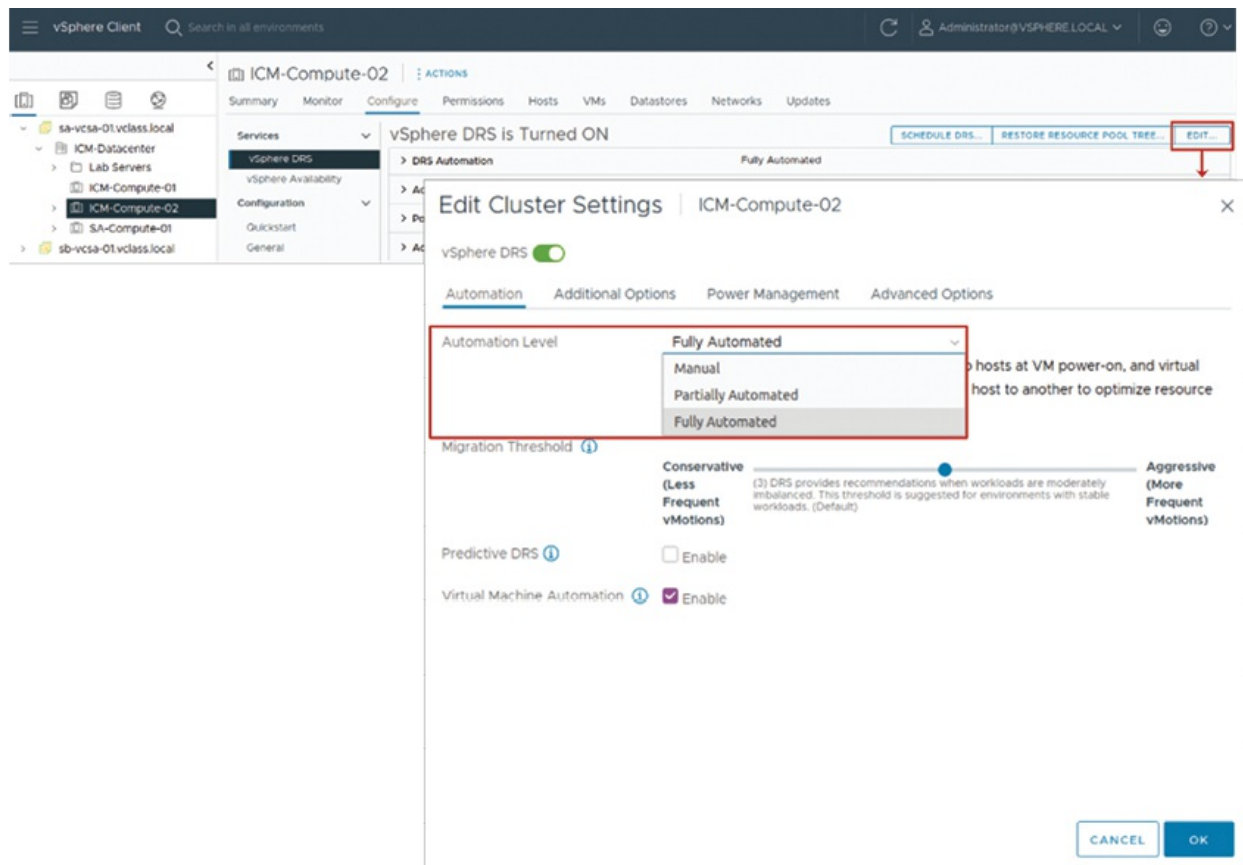


Figure 9.16: vSphere DRS settings, automation level

Source: VMware

vSphere DRS settings for migration threshold

The threshold for migration determines the level of aggressiveness DRS uses to balance the cluster by migrating VMs. It determines what recommendations will be followed in terms of migration, and each recommendation is expected to improve cluster performance differently, as outlined:

- **Level 1 (Conservative):** This mode only applies the Priority 1 recommendations. Only the bare minimum is done to help and attempt to balance the system:
 - Host maintenance
 - Affinity rule enforcement
 - Fewer than the necessary migrations, attending to the necessary required actions.
- **Level 2:** A vast improvement rests on the score; this level includes significant improvements to the cluster's DRS score. Moves VMs only when necessary, aiming to avoid unnecessary VM movement.
- **Level 3 (Default):**
 - This level applies the Priority 1, 2, and 3 recommendations.
 - A balanced approach is taken, providing a decent improvement versus the frequency of migration balance. It is usually recommended for most environments.
- **Level 4:** This level applies Priority 1 through 4 recommendations. It includes moderate improvements in cluster balance, targeting more aggressive migration behaviour than Level 3.
- **Level 5 (Aggressive):** This level uses every recommendation from Priority 1 to 5. The baseline for responsiveness is set, but frequent moves of VMs are likely to occur, applying all motion-triggered even on baseline adjustments to the DRS score, with every move integrating the Level 5 setting.

The following figure illustrates the migration threshold in vSphere DRS settings:

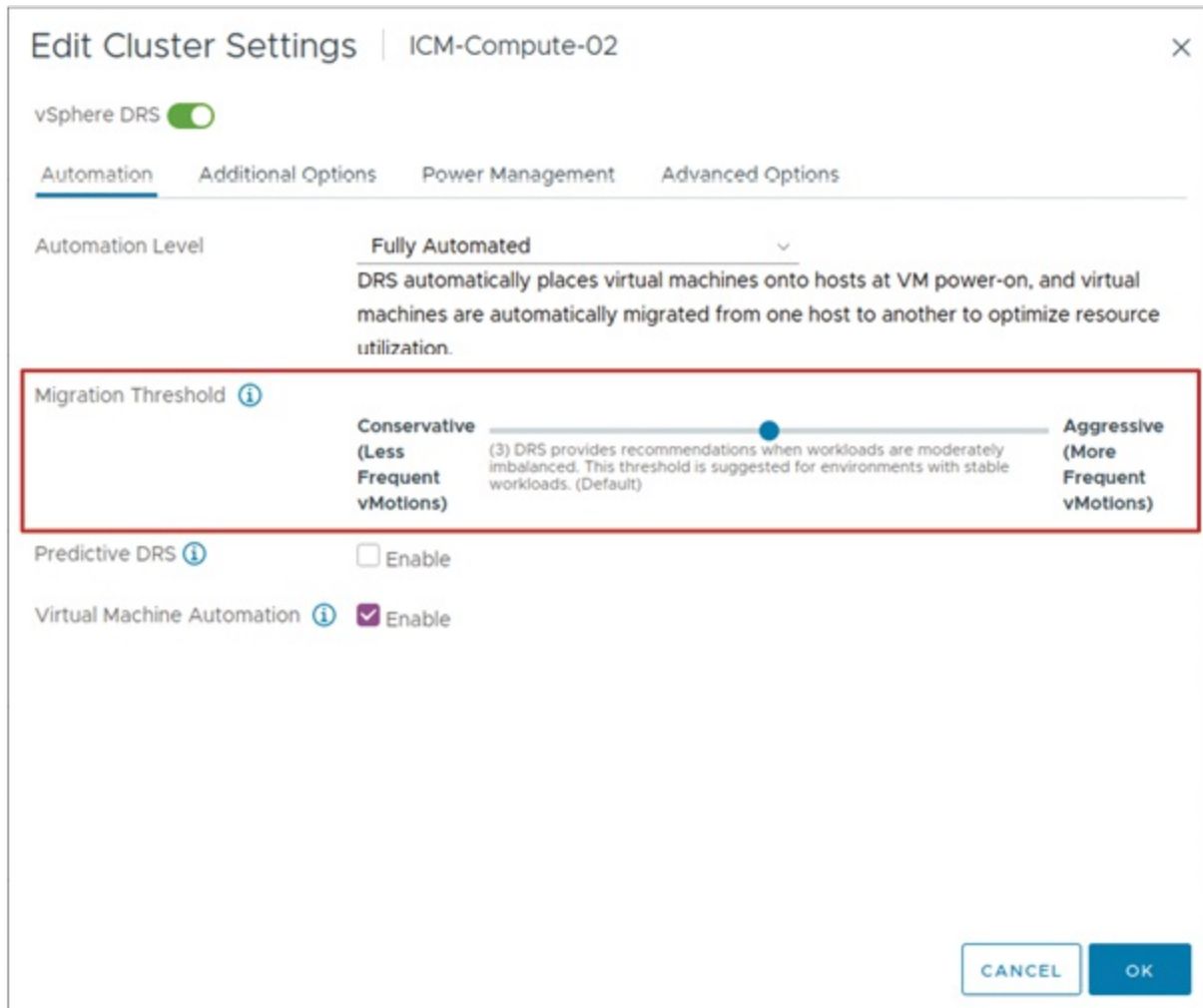


Figure 9.17: vSphere DRS settings, migration threshold

(Source: VMware)

vSphere DRS settings for predictive DRS

Predictive DRS utilizes past patterns of data collection to make proactive resource allocation decisions instead of waiting for a resource to be under or overutilized and reacting thereafter. This allows for a smoother performance for workloads without any mid-execution drops in performance.

For these decisions to be reliable, the vSphere DRS data collector must retrieve the following:

- Current resource consumption from ESXi hosts.
- Predicted consumption from VMware Aria Operations.

The main objectives of predictive DRS are:

- To prevent performance degradation of VMs, which could cause greater damage.
- To ensure resource contention is avoided, which fosters a responsive environment by taking measures early on.

An important point to remember, according to a vSphere DRS operational policy, predicted data overrides current usage data without exception. That is, in vSphere, it is more reliable and always performs better when planned rather than relying on current conditions for resources.

The following figure illustrates the predictive DRS in vSphere DRS settings:

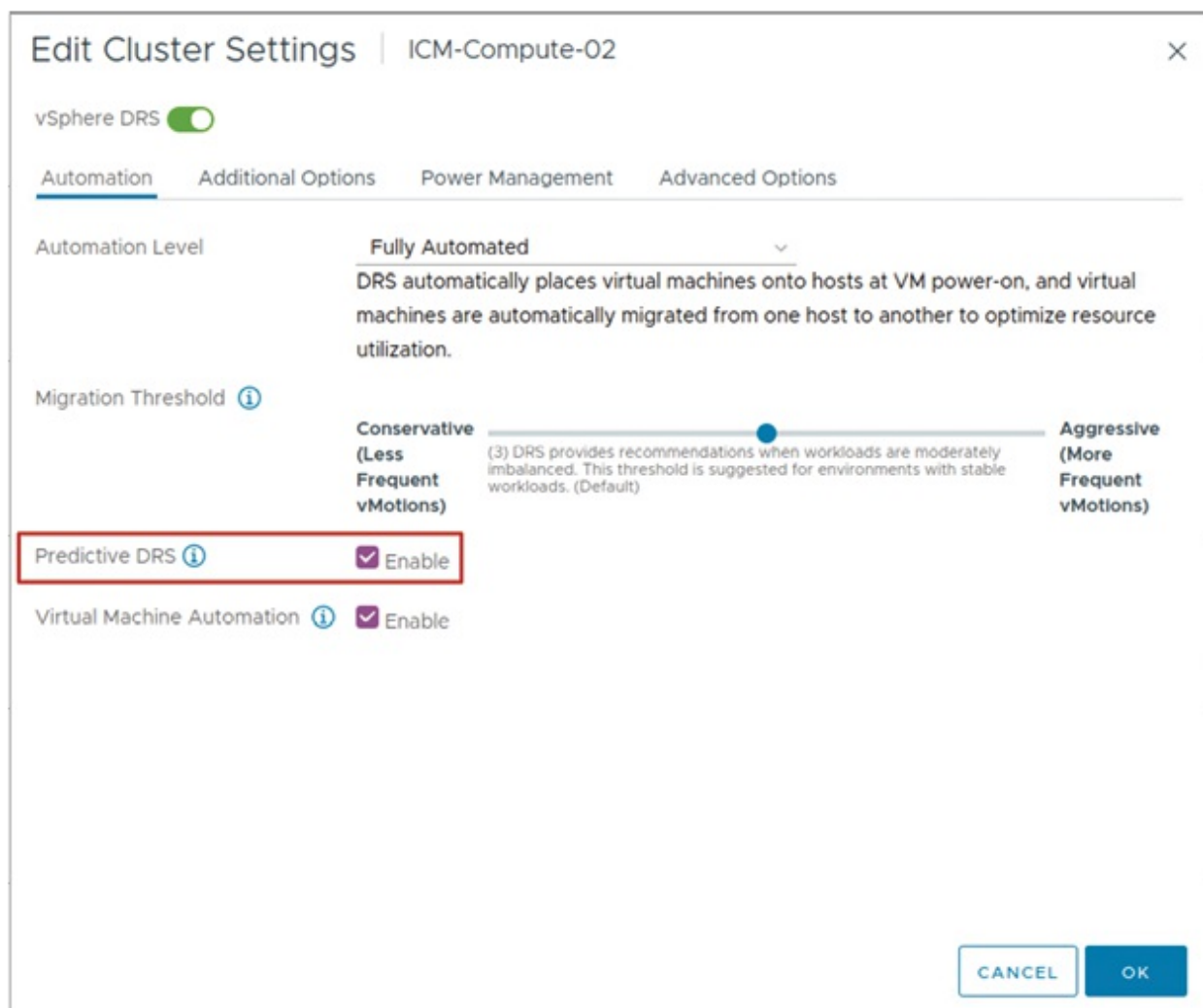


Figure 9.18: vSphere DRS settings—Predictive DRS

(Source: VMware)

Checking vSphere DRS settings

To check the configuration of the vSphere DRS, administrators need to click on **VIEW DRS SETTINGS**. A window pops up and displays all the important values and their active states.

The following list outlines some of the main settings:

- Automation level, which determines how DRS manages VM placement and migrations.
- Migration threshold, which indicates how aggressively DRS balances and loads moves the VMs in the cluster to the other available resources.

Administrators can find this information from the Summary tab of the cluster in the vSphere Client.

The following figure illustrates the vSphere DRS settings:

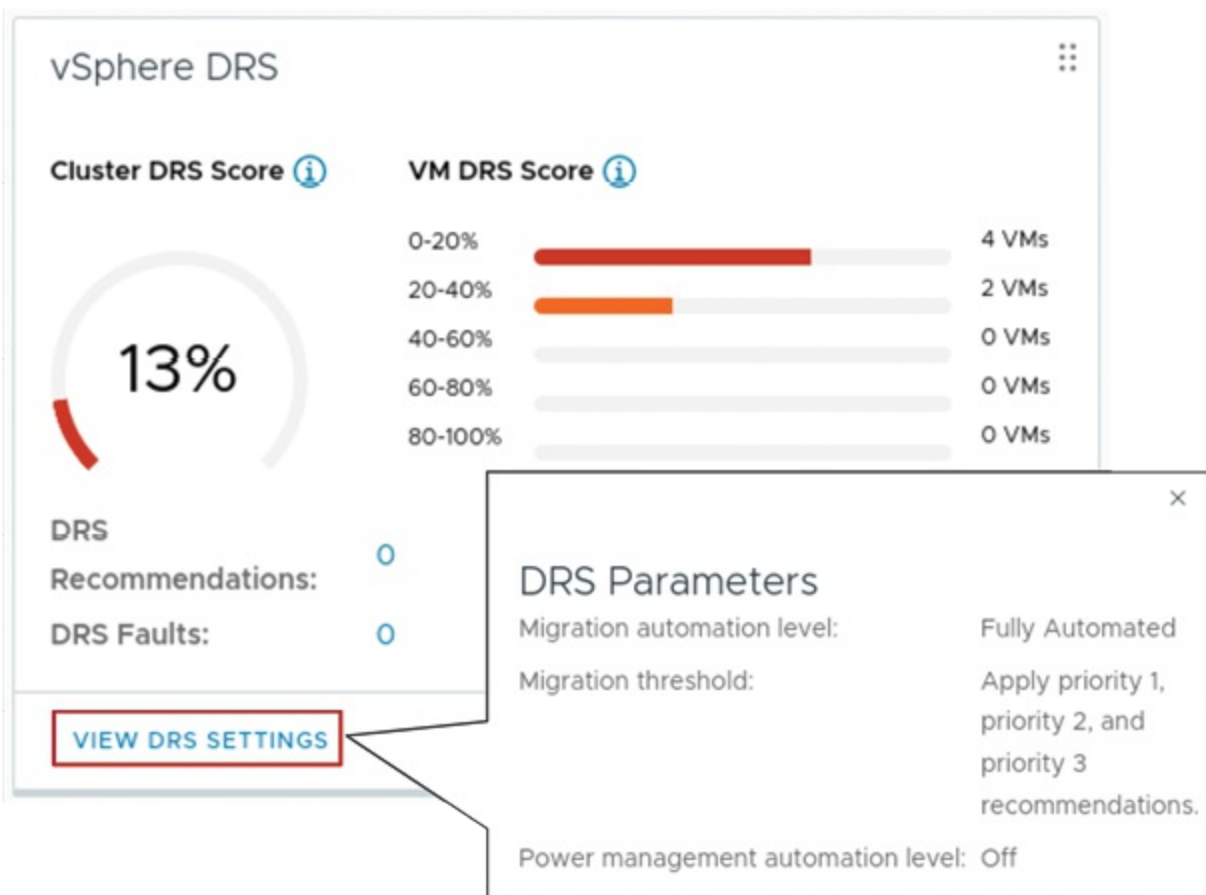


Figure 9.19: Viewing vSphere DRS settings

(Source: VMware)

Configurations of vSphere DRS for VMs automation

Besides having configured cluster-wide policies, administrators can adjust how DRS controls specific VMs. This gives more control on a per-VM basis, allowing administrators to circumvent some of the automated control at the cluster level.

Consider a VM that is critical to the operations. Administrators might want very stringent control over its containment and movement, so in such situations, they can set that VM's automation level to manual.

The following are the VM-level automation options:

- **Fully automated:** DRS assumes control of placing and migrating the VM completely.
- **Partially automated:** DRS performs the initial placement but only recommends migrations.
- **Manual:** Placement and migration suggestions will be offered, but administrators take the actions.
- **Disable:** DRS does nothing in regards to the placement and/or removal of the VM after initially stating its intentions to recommend.

This adaptability is beneficial for both novice DRS users and those maintaining a complex ecosystem.

The following figure illustrates the VM-level automation in vSphere DRS settings:

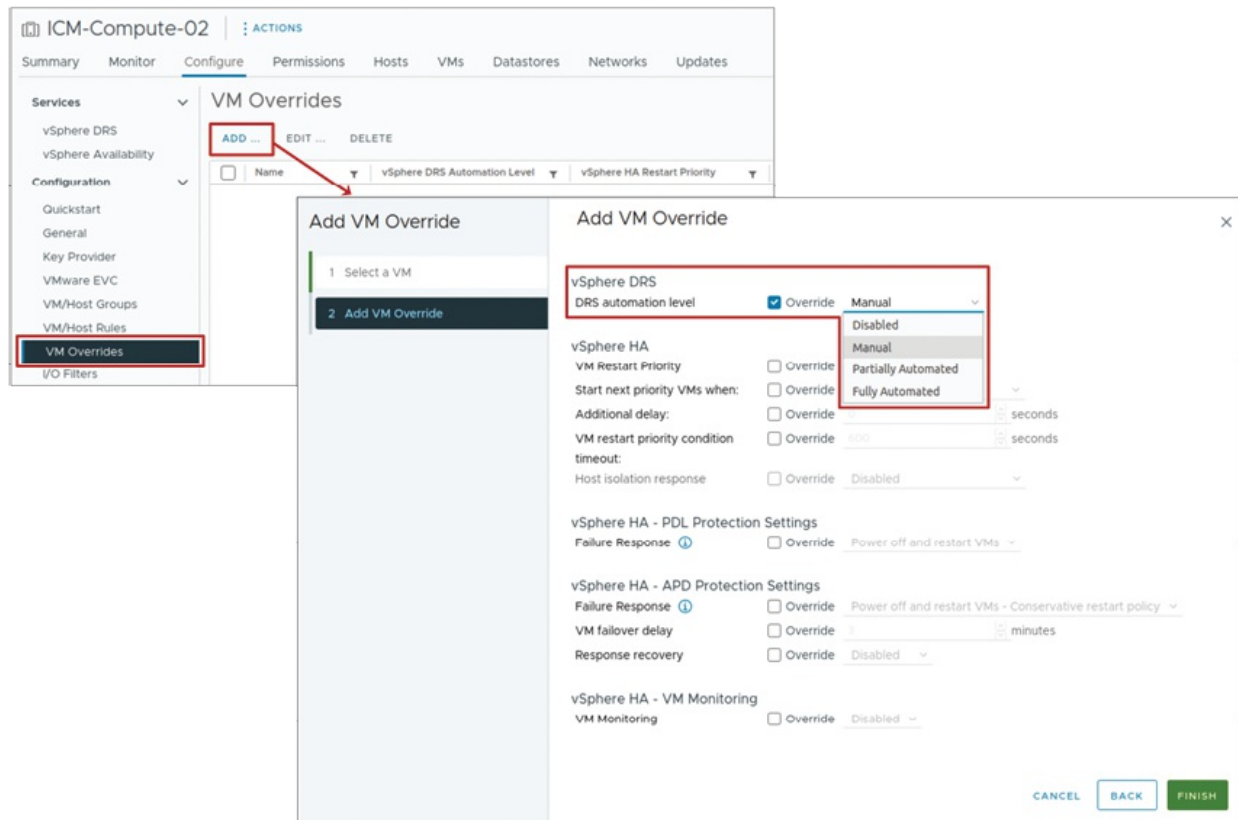


Figure 9.20: vSphere DRS settings—VM-level automation

(Source: VMware)

vSphere DRS settings for VM swap file location

With the powering of a VM, ESXi must create two swap files. These files are critical for the machine's starting process; hence, their creation is mandatory. In the case of a default setting, these swap files will usually exist alongside the other files residing in the datastores of the VMs. However, a change in this default logic should be triggered in certain cases.

Alternative available options are as follows:

- **Host-local swap:** This configures a local datastore attached to each host, enabling per-host swap. This setup is useful in certain aspects, although performance issues may arise during vMotion. Furthermore, vSAN and vSphere Virtual Volumes cannot be utilized in host-local swaps.
- **Per-VM configuration:** As required, administrators can even change the storage locations for swap files on each VM, allowing customized options for certain workloads.

The following figure illustrates the VM swap file location in vSphere DRS settings:

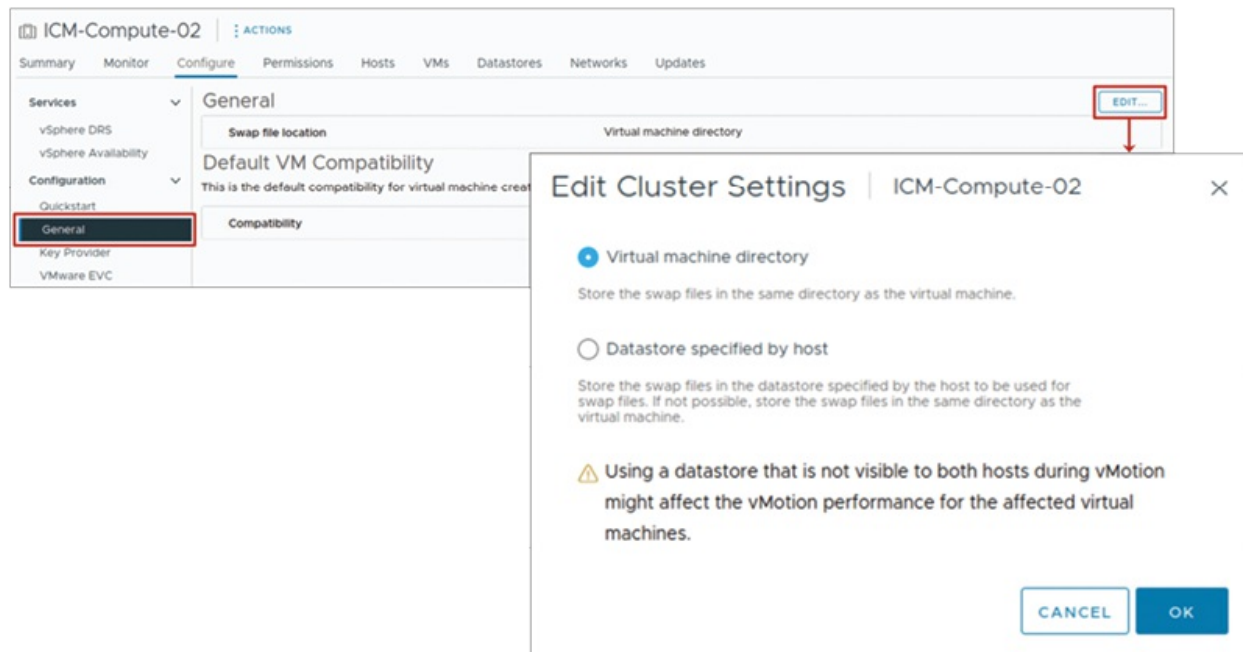


Figure 9.21: vSphere DRS settings—VM swap file location

(Source: VMware)

vSphere DRS configurations for affinity of VMs

vSphere DRS enables administrators to set VM affinity policies that determine how VMs are distributed across hosts in a cluster. These policies are particularly helpful for optimizing performance or guaranteeing availability, depending on the specific workload requirements.

Two main forms of VM policies exist, as follows:

- **VM affinity rules:** They ensure that selected VMs are contained to a single host. Most appropriate for VMs that communicate with each other and require a common host, such as multi-tier application components.
- **Anti-affinity rules,** which ensure that selected VMs are maintained on different hosts. Employed when high availability is desired, for instance, to avoid all domain controllers being placed on a single host.

Building and administering are based on vSphere DRS clusters:

- After configuring a vSphere DRS cluster, the administrator has the

option to edit the properties to create and manage the rules.

- If conflicting rules are set, vSphere DRS ensures that it will not enable both rules to maintain cluster uptime.
- A rule may be added, and the cluster may be in violation already (for example, the VMs are in an opposite placement to the enabling rule); in this case, vSphere DRS does not forcibly change the state of current operations.

Considerations based on automation level are as follows:

- In manual or partially automated DRS modes, migration recommendations are generated by considering both rule compliance and cluster load balancing.
- In fully automated mode, DRS handles VM placement and migrations automatically, while still honouring the defined affinity and anti-affinity rules.

The following figure illustrates the VM affinity in vSphere DRS settings:

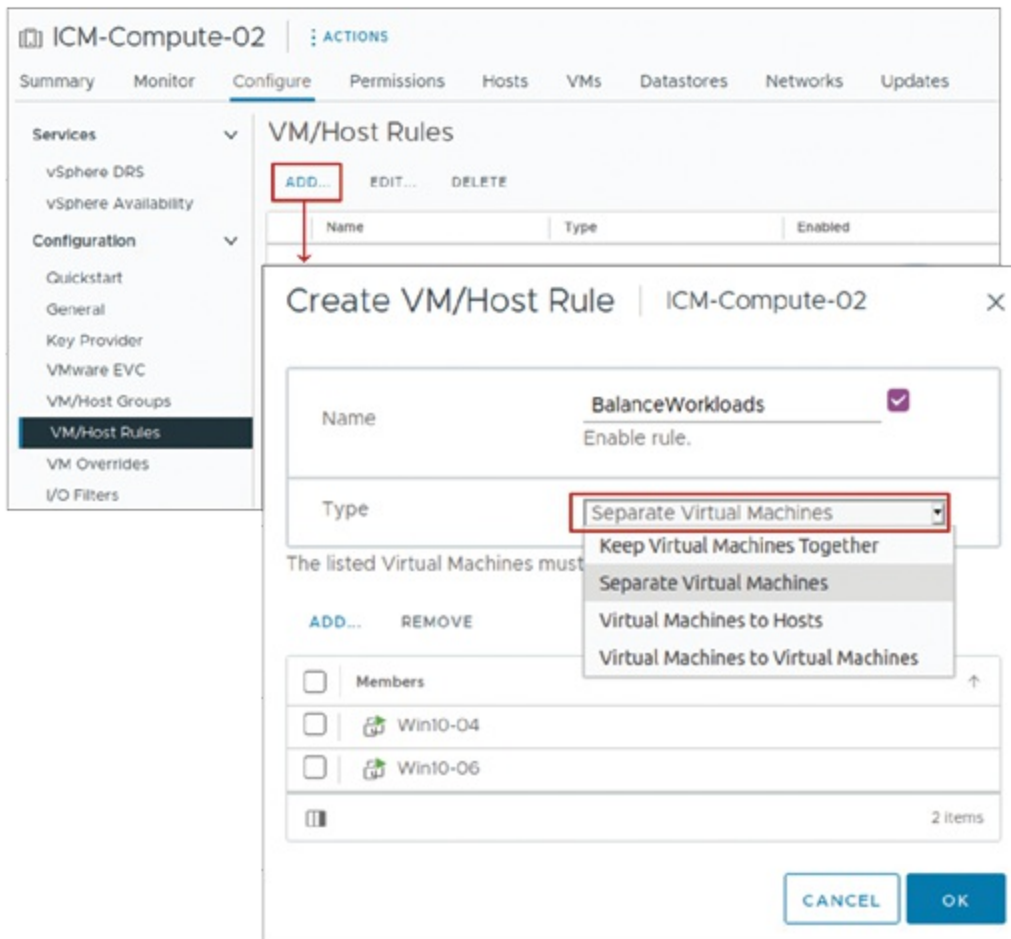


Figure 9.22: vSphere DRS settings—VM affinity

(Source: VMware)

vSphere DRS configurations for DRS groups

VM groups and host groups play a crucial role in defining VM-Host affinity rules within a vSphere DRS cluster. These groups enable administrators to define which VMs can or cannot run on specific hosts based on operational or compliance requirements.

The following outlines the structure and characteristics of these groups:

- **Group types:**
 - **VM group:** It is a single unit that may contain multiple VMs.
 - **Host group:** It is also a unit that has one or more ESXi hosts.
- **Characteristics:**

- A single VM can be part of multiple VM groups.
- A single host can be included in multiple host groups.

The primary function of VM and host groups is to aid VM-Host affinity rules, which require a certain placement type or define restrictions for the placement of VMs to hosts, and vice versa.

These groups enable more efficient cluster management because rules can be set at the group level, alleviating the need to create them per individual VM or host.

The following figure illustrates the DRS groups in vSphere DRS settings:

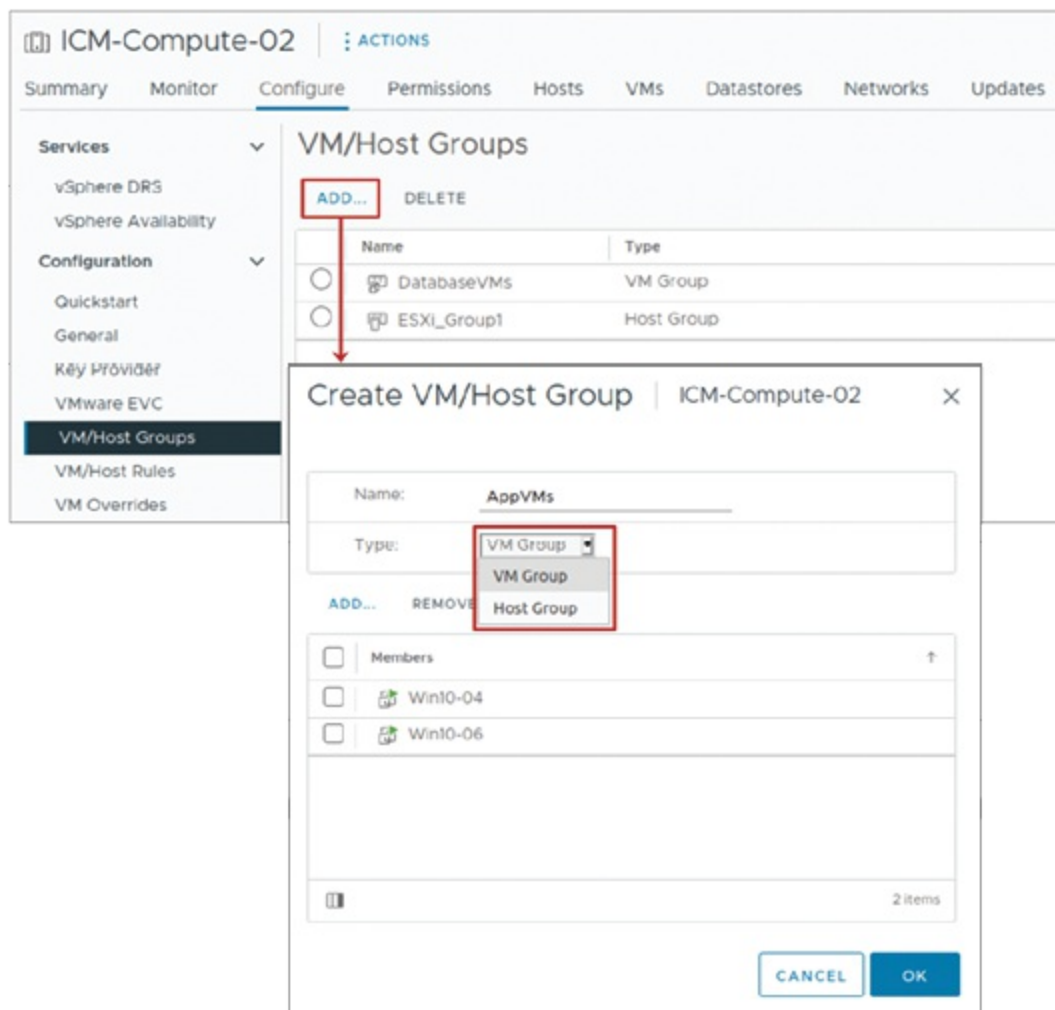


Figure 9.23: vSphere DRS settings—DRS groups

(Source: VMware)

vSphere DRS configurations for VM-Host affinity rules

VM-Host affinity rules specify both addition and removal placement constraints for machines relating them to groups of VMs and groups of ESXi hosts within a single vSphere DRS cluster. These rules are bound by criteria considered necessary for enforcing policies of an organization, ensuring appropriate licensing use, or backlog balancing.

Important affinity features to consider:

- A VM-Host affinity rule establishes either an affinity or anti-affinity relationship between:
 - A VM group (one or more VMs)
 - A host group (one or more ESXi hosts)
- Rules can be categorized as either:
 - **Required:** Must be strictly enforced.
 - **Preferential:** vSphere DRS tries to honour the rule but may violate it for overall balance.
- Available rule options:
 - Must run on hosts in a group
 - Should run on hosts in a group
 - Must not run on hosts in the group
 - Should not run on hosts in the group
- Additional considerations:
 - VM-Host affinity rules apply only within the same cluster. Both the VMs and the hosts must reside in the same vSphere DRS cluster.
 - If a VM is removed from the cluster, its membership in any VM groups is lost, even if it is re-added later.

The following figure illustrates the VM-Host affinity rules in vSphere DRS settings:

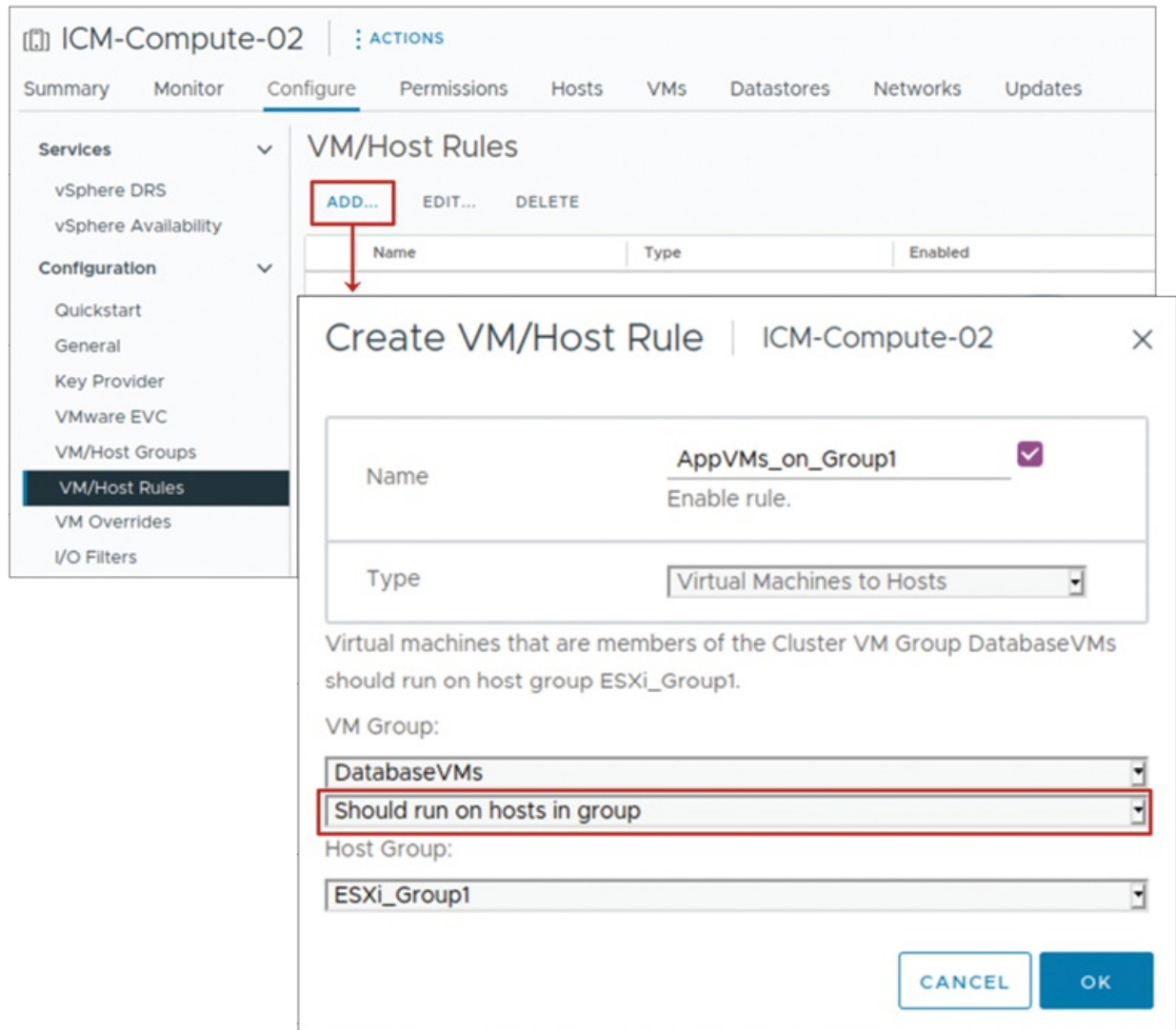


Figure 9.24: vSphere DRS settings—VM-Host affinity rules

(Source: VMware)

- **VM-Host affinity rules:** There are two types of VM-Host affinity rules in vSphere DRS: preferential and required. Details as follows:
 - Soft rules that provide guidance on using specific hosts to place VMs are called **preferential rules**. By default, they have some flexibility. For instance, if some blade systems are assigned to certain groups of VMs to mitigate performance issues, these rules assist. However, due to a host failure, maintenance, or resource constraints, vSphere DRS or HA can override them to ensure the VM is available.
 - Required or adopted rules are a class of norms that are custom

tailored basic rules of behaviour; these are strict and must be always followed. Mandatory guidelines like these are useful when specific licensing gaps or compliance gaps are present, such as the need to ensure that certain VMs do not operate on hosts other than those containing ISV licenses. In the absence of the stipulated hosts, vSphere HA is unable to reboot the VMs at other locations because DRS will not relocate the VMs to unapproved hosts.

In brief, preferential rules allow the user to bend the rules, whereas required rules restrict the user and impose firm controls based on identity policies or licensing.

- **Analysing vSphere DRS cluster resource usage:** The Monitor section of a vSphere DRS cluster provides a comprehensive overview of network traffic, CPU, and memory utilization for each host:
 - **CPU utilization:** Displays the distribution of a host's CPU resources to the VMs running on it. VMs are visualized as coloured boxes. Green indicates that the VM has been allocated all its entitled CPU resources. Other colours suggest that there are resource caps being placed somewhere and require investigation, such as unapplied DRS recommendations.
 - **Memory utilization:** Displays the amount of memory assigned to each VM as a bar, but unlike the previous metric, lacks colour coding, as the relationship between memory consumed and memory entitled is not very straightforward.
 - **Network utilization:** Measures the aggregate amount of traffic that a physical network interface card on the host server is receiving and gives an idea of the distribution of the load on the network.

The following figure illustrates the resource utilization of a vSphere DRS cluster:

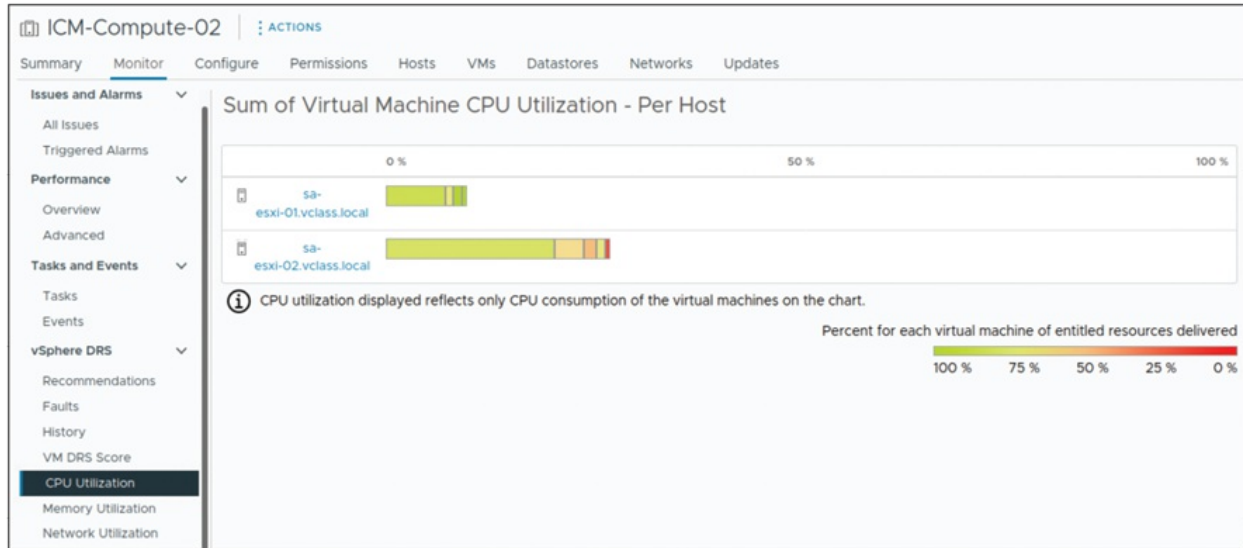


Figure 9.25: Viewing vSphere DRS cluster resource utilization

(Source: VMware)

Viewing vSphere DRS recommendations

The options available under recommendations in the monitor tab of the cluster allow the administrator to view and control the vSphere DRS recommendations. These recommendations are intended to balance resource allocation by means of VM migrations or power management, particularly when DRS is set to manual or partially automated modes.

Important actions are as follows:

- **RUN DRS NOW:** Refreshes the recommendations given.
- **SELECT ALL + APPLY RECOMMENDATIONS:** Applies all recommendations listed.
- **Selecting specific checkboxes + APPLY RECOMMENDATIONS:** Will issue only the selected recommendations.

The following are the further clarifications:

- **Faults tab:** Indicates problems encountered while trying to apply the recommendation.
- **History tab:** Keeps a record of DRS actions taken for auditing and review purposes.

The following figure illustrates the vSphere DRS recommendation:

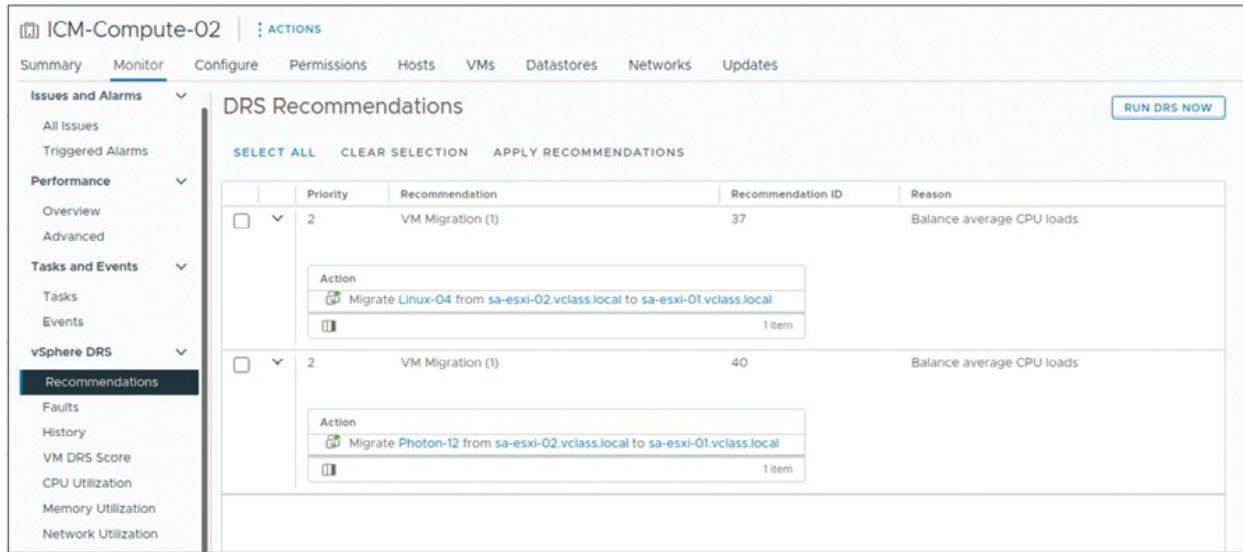


Figure 9.26: Viewing vSphere DRS recommendations

(Source: VMware)

Maintenance mode vs. standby mode

ESXi hosts can be placed in different operational states to manage workloads and system maintenance effectively. The two primary modes are:

- **Maintenance mode:**
 - This mode is used when servicing a host, such as in the case of hardware upgrades or removing the host from a cluster.
 - All host resources become unavailable.
 - VMs need to be migrated, shut down, or suspended before the host can enter maintenance mode.
 - This action is done manually.
 - No powering on or deploying of VMs can be done in this mode.
- **Standby mode:**
 - This mode is invoked by vSphere **Distributed Power Management (DPM)** to minimize power usage.
 - The host is powered down, apart from the baseboard management controller.
 - DPM automation or manual activation puts the system into this mode.

- This mode may be overridden by DRS in the next evaluation cycle.

To keep the host in a power-off state without any external override, use maintenance mode and turn off the host.

The following figure illustrates the maintenance and standby mode:

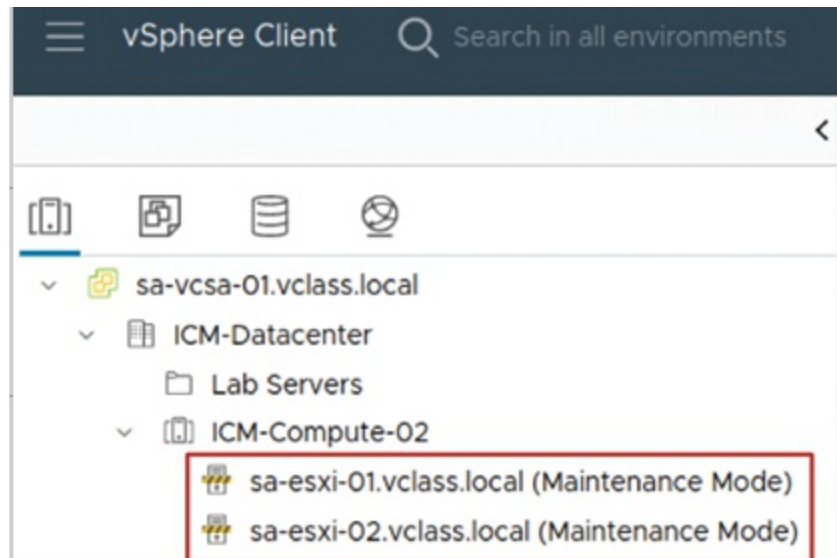


Figure 9.27: Maintenance mode and standby mode

(Source: VMware)

Steps to remove a host from a vSphere DRS cluster are as follows:

1. Place the host into maintenance mode.
 - a. All running VMs need to be shut down, suspended, or migrated (vMotion).
 - b. If DRS is automated, powered-on VMs are migrated automatically.
 - c. VMs on local storage need to be manually turned off or relocated.
2. Remove the host.
 - a. Drag the host to a different inventory location (data centre or another cluster).
 - b. The host, along with its VMs, will be removed from the cluster.
 - c. The available resources in the cluster decrease.

DRS recalculates how to distribute the resources to the VMs to be serviced with reasonable resources and decides how to allocate them.

vSphere HA overview

Providing affordable safeguard options for applications operating on VMs using vSphere HA is possible due to rapid recovery from hardware and software failures. After configuring vSphere HA, it automatically protects all VMs after adding a new workload on a cluster, thereby reducing manual work.

Types of failures that vSphere HA protects are as follows:

- **Failure of an ESXi host:** In the event of an unexpected host shutdown, vSphere HA will restart all VMs running on the remaining hosts of the cluster.
- **Failure of VM:** In the event a VM does not send heartbeat signals via VMware Tools within the set time, vSphere HA will restart that particular VM on either the same or a different host.
- **Failure of an application:** Inability of an application running on a VM to send heartbeat signals within the given time-period results in the VM being restarted to bring it back to a functional state.
- **Failure of accessibility to datastore:** vSphere HA will restart the affected VMs on other hosts that can access the datastore in case a host loses access to it.
- **Isolation of a network:** Irrespective of cluster membership to the heartbeat network, vSphere HA will restart the VMs to ensure availability from network partition.

Unlike traditional clustering solutions, vSphere HA integrates directly into the virtualization platform. Once enabled, it automatically protects all workloads without requiring VM-specific configuration by offering a simple, scalable, and infrastructure-driven approach to high availability.

Organizations can reduce both planned and unplanned downtime with rapid recovery from outages using VMware vSphere. Regardless of the cause, downtime can lead to losses, both operationally and financially. Traditionally, achieving **high availability (HA)** required servers configured and connected with advanced, expensive, complicated hardware. With vSphere, this need is now addressed with a simplified and economical software solution for delivering enterprise-grade availability across all

applications.

The following are the key benefits of vSphere for availability:

- **Hardware-independent protection:** The virtualization layer has been abstracted from underlying hardware, operating systems, and applications. This independence from the hardware stack permits vSphere to guarantee availability, continuously ensure cross-platform virtualization, and mitigate management risks across multi-domain environments.
- **Reduction in planned downtime:** Most routine maintenance work, such as storage updates, firmware patches, and server upgrades, often mandates bringing resources offline and requires long operational downtimes. vSphere mitigates the impact during these tasks through:
 - **vSphere vMotion:** Enables live migration of VMs from one host to another with zero downtime.
 - **vSphere Storage vMotion:** Facilitates the movement of VM disk files across datastores without interrupting the operation of the VM.
- **Unplanned downtime recovery:** vSphere provides features for the automatic recovery from unpredictable failures:
 - **HA of vSphere:** Supervises the hosts and VMs in a cluster and restarts VMs on other hosts in case of a host failure.
 - **Fault tolerance (FT) of vSphere:** Ensures that there is no downtime or loss of data by running a secondary VM along with the primary VM. If the primary host fails, FT can seamlessly hand over to the secondary VM without any service interruption.
- **Replicating data and disaster recovery:** To restrict site-wide failures and loss of data:
 - **vSphere Replication:** Facilitates native replication of a VM at a remote site by copying the disk files of the VM. The VM can be recovered at a certain point and is independent of storage.
 - **Site Recovery Manager (SRM):** An integrated automation solution for enforcing policies of disaster recovery with vCenter that uses:
 - vSphere Replication

- Array-based replication (provided by supported storage vendors)
- Failover, failback, and DR testing are managed and automated by SRM, enabling continued business with minimal effort and improved **recovery time objectives (RTOs)**.
- **Integration of backup and recovery:**
 - In vSphere, backup and restore operations are simplified as VMs are stored as files.
 - The VMware Storage API's third-party access data protection allows image-level and file-level backups to be performed efficiently.
 - These are very helpful for recovering from data corruption, malware attacks, or deletion of files from the guest OS.
 - The encapsulation model and API features enable efficient, advanced backup solutions while managing complexity in data protection.

Data availability is protected on multiple levels in vSphere, starting from live migration during maintenance, automated failure recovery, backup replication, and dynamic disaster replication recovery. The system employs software-defined solutions, resulting in lowered costs and complexities associated with traditional high-availability systems. In return, strengthening the system's robustness, resilience, and business continuity in modern IT frameworks.

The following figure illustrates the protection at every level:

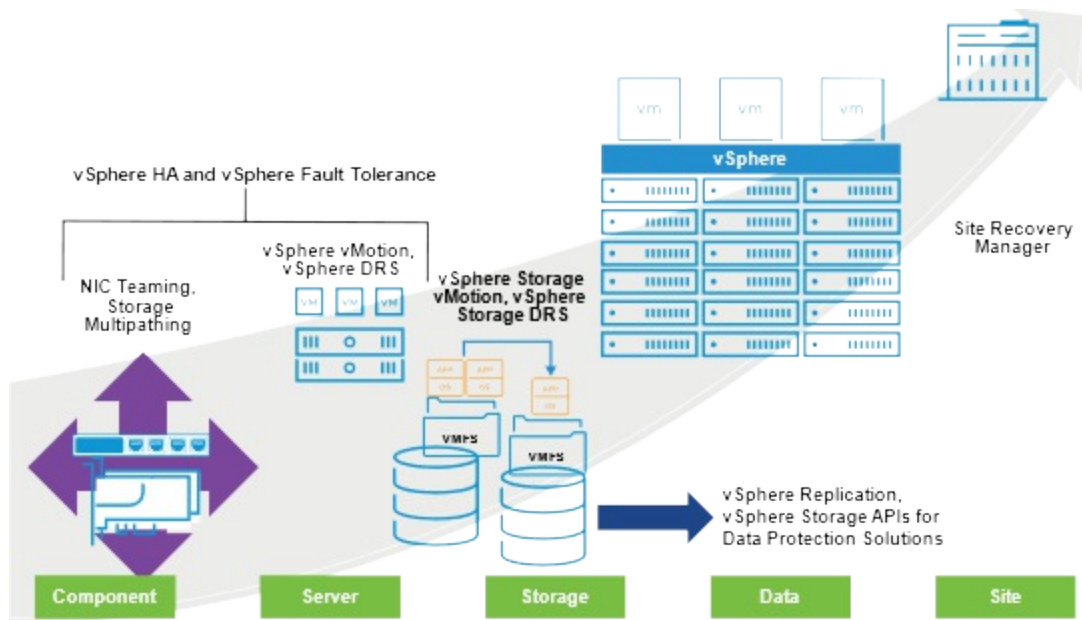


Figure 9.28: Protection at every level

(Source: VMware)

vSphere HA scenario for ESXi host failure

In the event of an ESXi host failure, vSphere HA ensures continued service availability by automatically restarting the affected VMs on other functioning hosts within the cluster. vSphere HA is designed to distinguish between a host that is isolated from the network and one that has completely failed. When a host failure is detected, HA immediately initiates the VM failover process using the remaining cluster resources.

The total recovery time in this scenario depends largely on how quickly the guest operating systems and applications inside the VMs can boot and resume operations.

vSphere HA addresses various failure scenarios beyond host failures, each requiring specific response mechanisms:

- **vSphere HA scenario—guest operating system failure:** If a VM stops responding, either due to a failure in the guest operating system or a crash of the VM process (vmx), vSphere HA can detect the issue and restart the VM on the same host. This functionality requires VM monitoring to be enabled.

With VM monitoring active, the vSphere HA agent keeps track of

VMware Tools heartbeat signals for each VM. If heartbeats stop for a predefined duration, HA considers the VM to have failed and initiates a restart to restore services quickly.

- **vSphere HA scenario—application failure:** Application failures in vSphere HA clusters can also be addressed using application monitoring. If a VM's application stops functioning, it will fail to send heartbeat signals. In such cases, vSphere HA will attempt to restart the VM on the same host to restore the application service.

vSphere HA can only use the application monitoring feature if the appropriate SDK and VMware's Application Monitoring are deployed in the network, using an application that is compatible with VMware Application Monitoring. During operation, the application will send custom heartbeat pulses that are monitored by the HA agent on each host.

- **vSphere HA scenario: datastore accessibility failures:** vSphere HA can detect and correct problems related to datastore availability, thereby guaranteeing long-term VM availability despite interruptions to the underlying storage paths. vSphere HA calls a broad variety of recovery methods based on the type and severity of failure experienced.
- **HA response to types of datastore failure:**
 - **All-Paths-Down (APD)—An event that can be recovered:** This state occurs when a host loses all its access pathways to a datastore temporarily. The reason can either be a temporary storage problem or an unknown problem. vSphere HA can provide the following reactions:
 - Issue events only.
 - Power off and restart VMs—Conservative policy.
 - Power off and restart VMs—Aggressive policy.
 - **Permanent Device Loss (PDL)—An event that cannot be recovered:** This state occurs when the storage device clearly indicates that the datastore is permanently unavailable. The possible responses are:
 - Issue events only

- Power off and restart VMs
- **Reset policy options:**
 - **Conservative restart policy:** This choice permits vSphere HA to restart and power off VMs only if it finds that another host within the cluster can successfully restart the VMs, and then vSphere will continue. The APD-affected host will query the primary HA host to check for resource availability. If it is not possible to communicate or capacity is unknown, nothing is attempted, minimizing the risk of VM loss.
 - **Aggressive restart policy:** The major position here is quick recovery, with it having the highest priority, though it cannot ensure other hosts can recover them. The protocol is extremely appropriate for split-brain situations where a network that is split cannot access the main HA node. In situations where there is insufficient capacity within the cluster, recovery of certain VMs will not be feasible. Through its deliberate failure on the datastore level, vSphere HA reduces downtime and optimizes workload reliability even when utilized during intricate storage failure.

Importance of heartbeat networks in vSphere HA

For vSphere High Availability, heartbeat networks are of utmost importance for accurate failure detection and reliable operation. These networks use VMkernel ports configured for either management traffic or vSAN traffic to transmit heartbeat signals.

Key characteristics of heartbeats are as follows:

- Exchanged between the primary HA host and secondary hosts in the cluster.
- Used to identify whether a host becomes unresponsive, falls under isolation, or suffers a state of failure.
- Transmitted over the designated heartbeat network.
- If both vSAN and vSphere HA are enabled, the vSAN network takes precedence as the heartbeat network over the management network.

The advantages of having duplicate heartbeat networks are:

- **Enhanced reliability:** The backup does not rely on a single point of failure, and the reduction of the chance of failure due to redundancy is achieved, thanks to the aid of multiple networks.
- **Mitigates partition or isolation scenario:** The assumption stands that each network can cater to one network route that gets blocked, and it is accessible through other non-blocked routes for heartbeats.
- **Reduces the single point of failure:** Standby segments make it possible to guarantee that the hosts are still connected, which is crucial to sustaining uninterrupted cluster functionality as well as fail-over decisions.

The following figure illustrates the heartbeat networks:

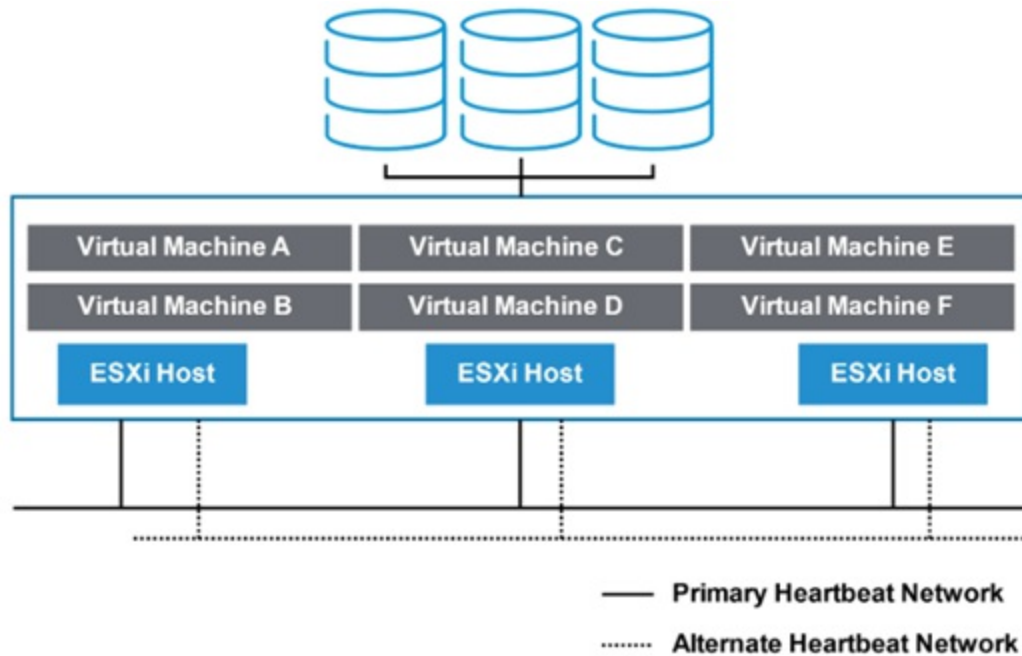


Figure 9.29: Importance of heartbeat networks

(Source: VMware)

vSphere HA scenario for protecting VMs against network isolation

vSphere HA safeguards VMs from network isolation by monitoring the heartbeat communication between ESXi hosts. If a host becomes isolated, meaning it remains powered on but loses all connectivity on the management

or vSAN network, vSphere HA takes action to maintain VM availability.

Refer to the following list to understand host network isolation:

- Host network isolation occurs when an ESXi host loses network connectivity with other HA-enabled hosts.
- The host is still running but can no longer detect heartbeat traffic from peer hosts over the heartbeat network.
- In response, vSphere HA restarts the affected VMs on other healthy hosts in the cluster.

Note: If network infrastructure has sufficient redundancy, network isolation conditions can usually be avoided.

The following figure illustrates the vSphere HA scenario protecting VMs against network isolation:

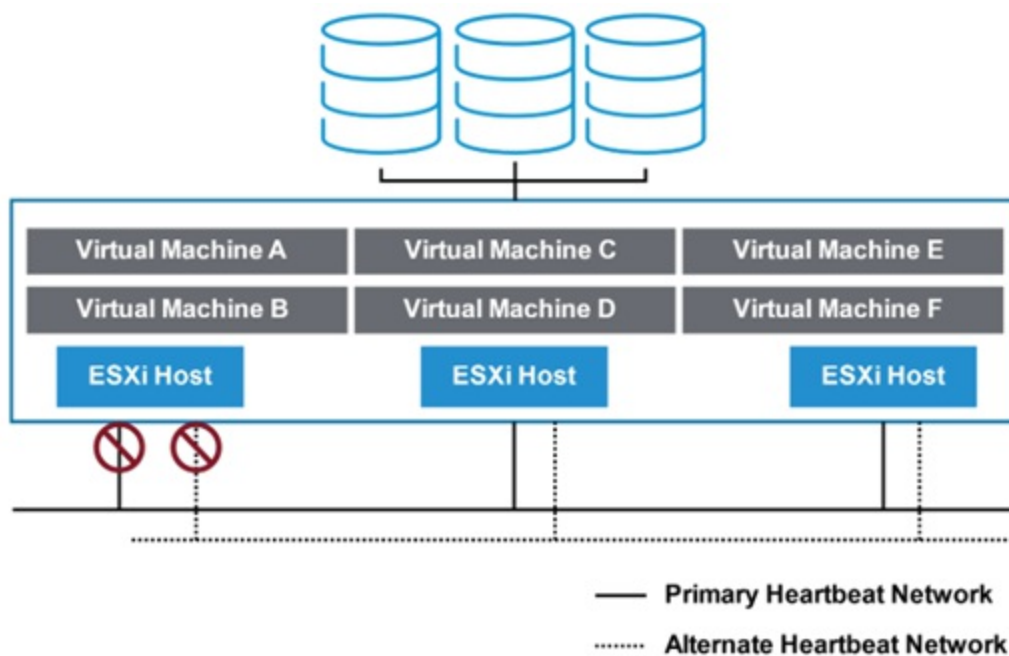


Figure 9.30: vSphere HA scenario protecting VMs against network isolation

(Source: VMware)

Heartbeat network redundancy using NIC teaming

To enhance reliability and prevent false isolation events, VMware recommends NIC teaming for heartbeat network redundancy.

Refer to the following list to understand NIC teaming:

- NIC teaming involves combining multiple physical NICs (e.g., vmnic0 and vmnic4) into a logical group to provide fault tolerance and load balancing.
- When used in the Management network, it ensures that the VMkernel port vmk0 (which has the Management service enabled), remains connected even if one NIC fails.

The benefits of NIC teaming are as follows:

- Redundant heartbeat paths reduce the risk of false isolation scenarios.
- NIC teaming enhances network resiliency for both management and HA heartbeat traffic.

It ensures that VM restart decisions are based on actual host failure, not on temporary network disruptions.

Note: Always implement NIC teaming on the VMkernel ports used for management or vSAN traffic in a vSphere HA-enabled cluster.

The following figure illustrates the redundancy using NIC teaming:

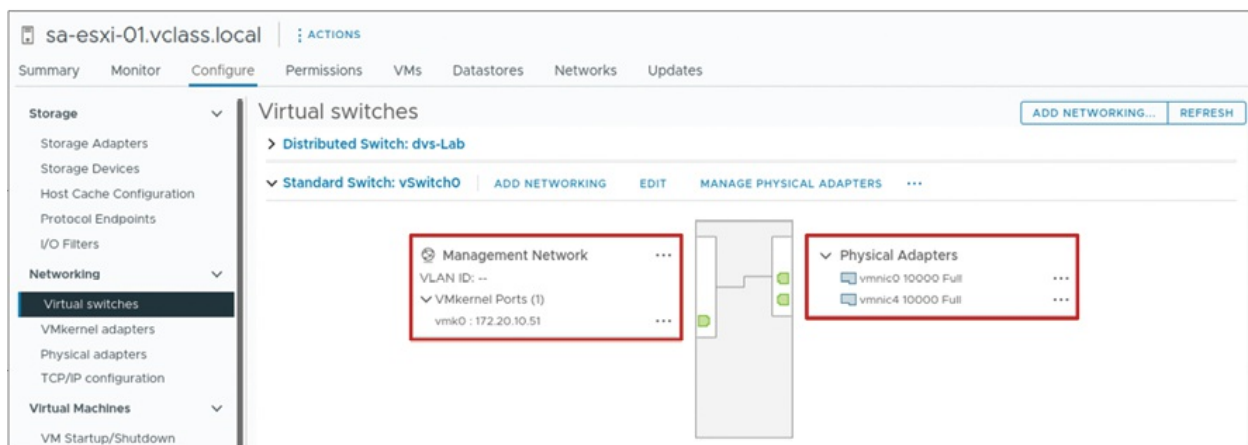


Figure 9.31: Redundancy using NIC teaming

(Source: VMware)

Heartbeat network redundancy using additional networks

Administrators can enhance vSphere HA reliability by configuring additional heartbeat networks. This is done by creating a second VMkernel port, either on a separate virtual switch with its own physical adapter or on the same switch but in a different port group using a different adapter. While NIC

teaming usually provides enough redundancy, adding VMkernel ports offers extra protection. If one path fails, heartbeat traffic continues over the other, reducing the risk of false host isolation.

The following figure illustrates the redundancy using additional networks:

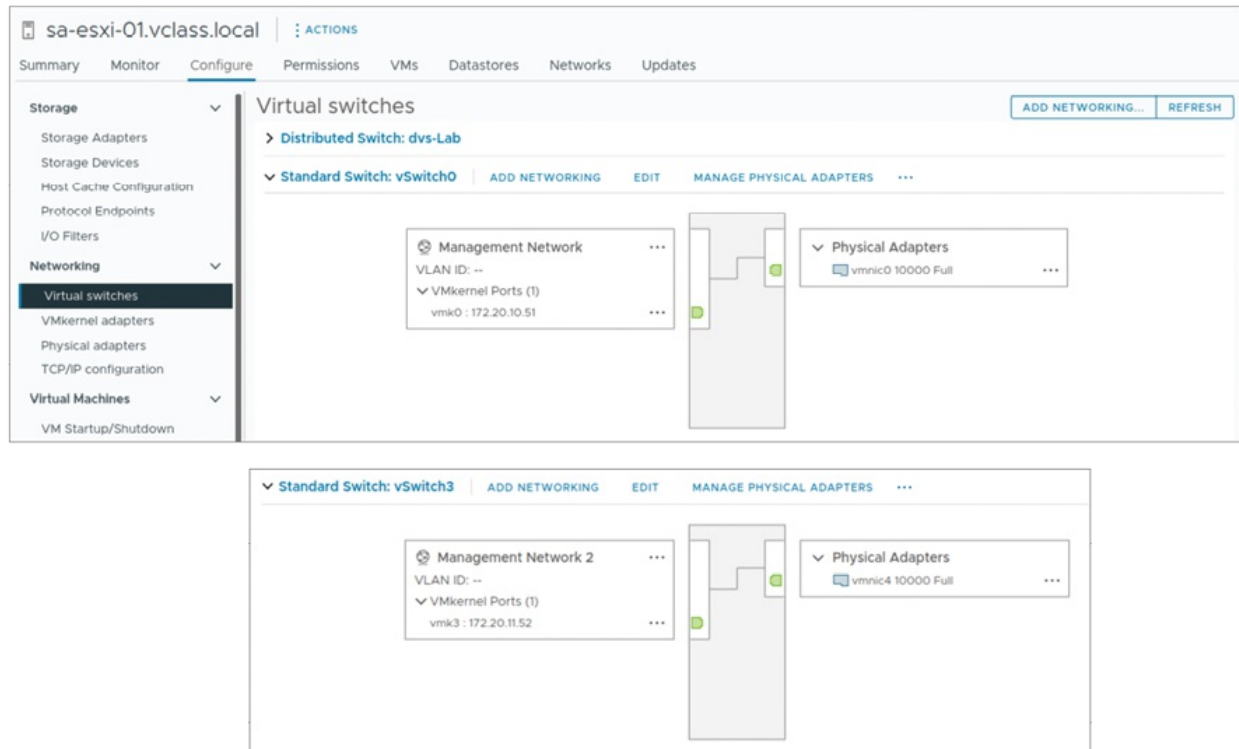


Figure 9.32: Redundancy using additional networks

Source: VMware

vSphere HA designing and configuration

On every host in a vSphere HA cluster, the **Fault Domain Manager (FDM)** service, known as the vSphere HA agent, is active. One host is elected as the primary, while others serve as secondary hosts. The primary host handles cluster configuration, monitors the secondary hosts, and coordinates VM protection and recovery.

To determine the primary host, an election procedure is followed, which takes approximately 15 seconds. The host with access to the maximum number of datastores is elected. If several hosts have access to an equal number of datastores, the host given the lowest **Managed Object ID**

Protection and configuration information are retained locally on each host and duplicated during cluster reconfiguration. A VM is protected when powered on (including memory-state snapshot revert) and is unprotected when powered off or reverted to a snapshot without memory state.

The diagram illustrates a three-node PostgreSQL High Availability (HA) architecture. At the top, three 'Datastore' icons represent the PostgreSQL databases. Below them, three nodes are shown, each containing an 'FDM' (Failover Manager) process. The nodes are labeled 'ECX Host (Secondary)' for the first two and 'ECX Host (Primary)' for the third. Each node also contains 'vpxa' and 'hostd' components. Arrows show data flow from the Datastores to the FDMs and between the FDMs. A 'vCenter' component at the bottom manages the nodes via 'vpxd' agents. A legend indicates that orange lines represent the 'Heartbeat Network'.

(Source: VMware)

In a vSphere HA cluster, the primary host broadcasts network heartbeats to all secondary hosts periodically to monitor their health. Secondary hosts pick up these heartbeats and respond via one of the configured heartbeat networks. When the active heartbeat path is dropped, the secondary host automatically

fails over to another operational heartbeat network.

If the secondary host fails to respond within the specified timeout period, the main host declares it unreachable and begins diagnostic tests to determine if the issue is due to a system crash, a network break, or agent communication failure.

The following figure illustrates the vSphere HA architecture, networks heartbeats:

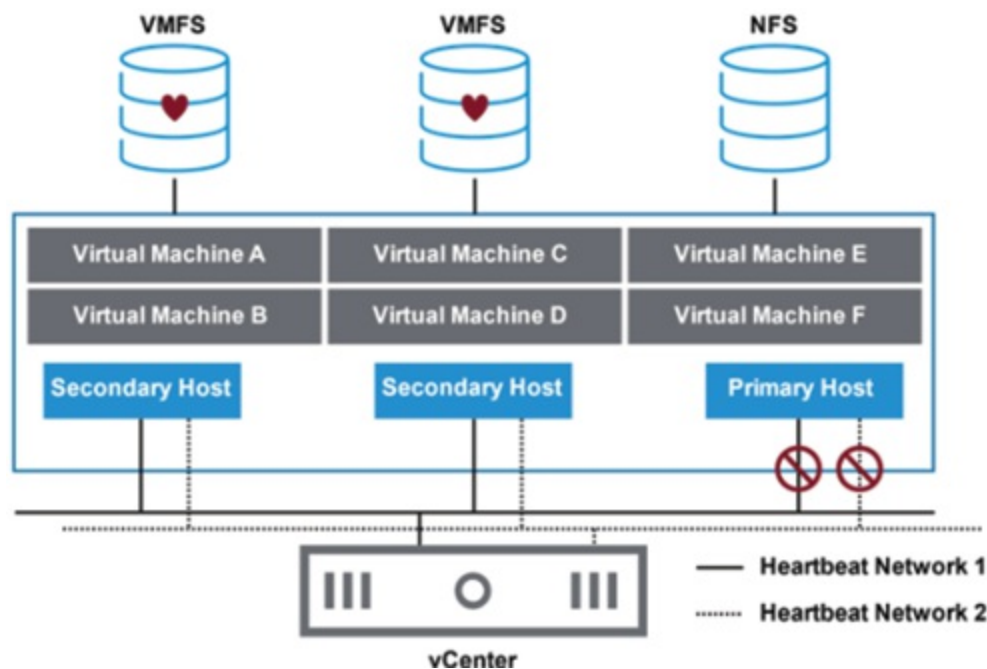


Figure 9.34: vSphere HA architecture networks heartbeats

(Source: VMware)

vSphere HA designing for datastore heartbeats

When the primary host loses connection with a secondary host via the heartbeat network, it uses datastore heartbeat to find the root cause, which could be a host failure, network partition, or network isolation. If the host's datastore heartbeats stop, the host is marked unsuccessful, and its VMs are restarted on another host in the vSphere HA cluster.

The following figure illustrates the vSphere HA architecture datastore heartbeats:

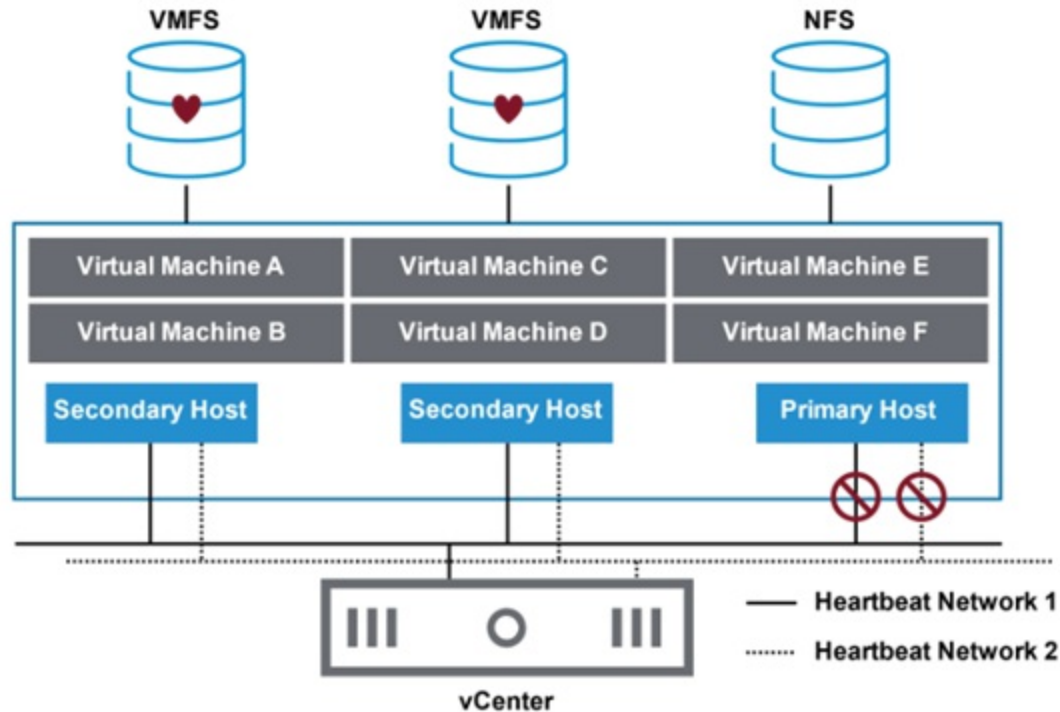


Figure 9.35: vSphere HA architecture datastore heartbeats

(Source: VMware)

vSphere HA failure scenarios

vSphere HA can detect and respond to a range of failure conditions that may impact availability within a cluster. These scenarios include the following:

- **Secondary host failure:** A condition where a secondary ESXi host becomes unresponsive or fails.
- **Primary host failure:** A failure of the elected primary ESXi host responsible for managing HA communication and coordination.
- **Network failure (Host isolation):** This occurs when a host cannot communicate with other cluster members over the heartbeat network and fails to ping its designated isolation address.
- **Datastore accessibility failures:**
 - **All-Paths-Down (APD):** Temporary loss of access to a datastore with no path available.
 - **Permanent Device Loss (PDL):** Permanent unavailability of a datastore due to storage device failure.

vSphere HA distinguishes between host failure and host isolation, details as follows:

- Host Isolation occurs when an ESXi host cannot detect network traffic from other hosts and fails to reach its configured isolation address.
- In the event of host failure, vSphere HA attempts to restart the affected VMs on other functional hosts in the cluster.
- If a host is isolated, it initiates the host isolation response, which may include powering off or leaving VMs powered on, depending on the configured policy.

Failed secondary hosts

When a secondary host does not respond to network heartbeats from the primary host, the primary host investigates the problem, which could be isolation, misconfiguration (for example, firewall rules), or hardware failure.

If network communication fails, the primary host uses datastore heartbeats to determine the host's state, details as follows:

- If the secondary host stops sending heartbeats to the datastore, it is marked as unsuccessful, and its VMs are restarted on different hosts.
- To confirm host activity, VMFS datastores read a heartbeat area.
- In NFS datastores, a locked host, **hb** file indicates an active heartbeat. The timestamp of the lock file indicates whether the host is isolated or has failed.
- To decrease overhead, vCenter picks up to two shared datastores per host, depending on their accessibility across hosts for datastore heartbeating.

The following figure illustrates the failed secondary host:

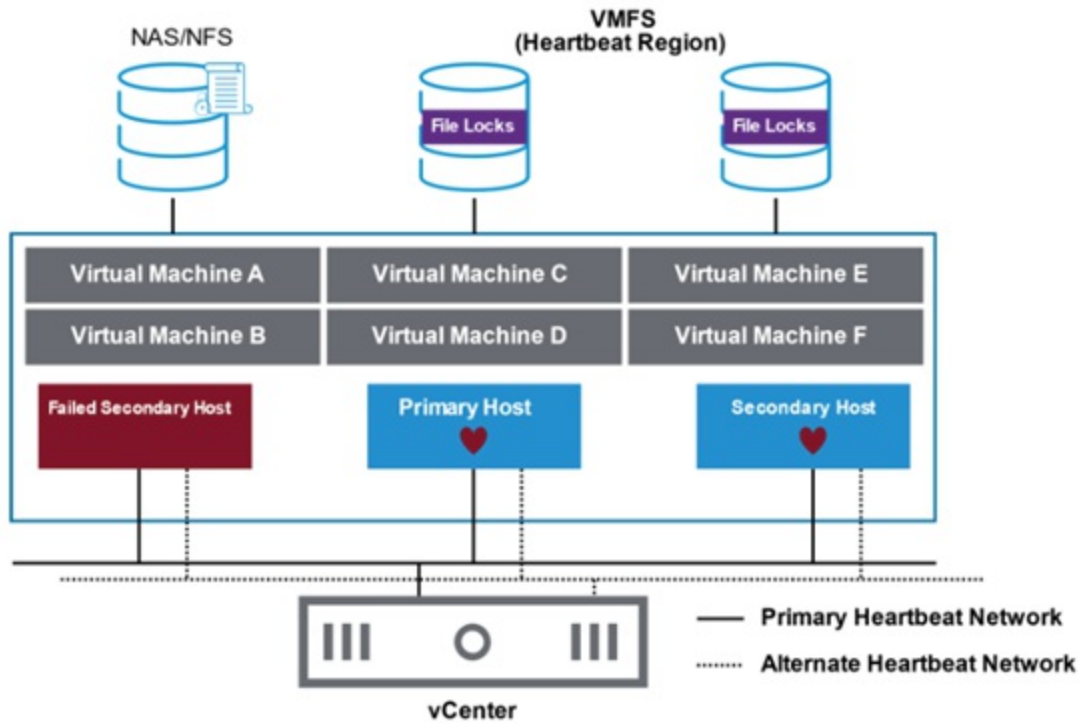


Figure 9.36: Failed secondary host

Source: VMware

Failed primary hosts

If the primary host fails, is shut down, or enters maintenance mode, secondary hosts detect the absence of heartbeats and trigger a new election to choose a replacement.

The election selects the host that:

- Has access to the greatest number of datastores.
- If there is a tie, the host with the lowest MOID assigned by vCenter is selected.

This ensures uninterrupted vSphere HA cluster management by promoting a new primary host quickly when needed.

The following figure illustrates the failed primary host:

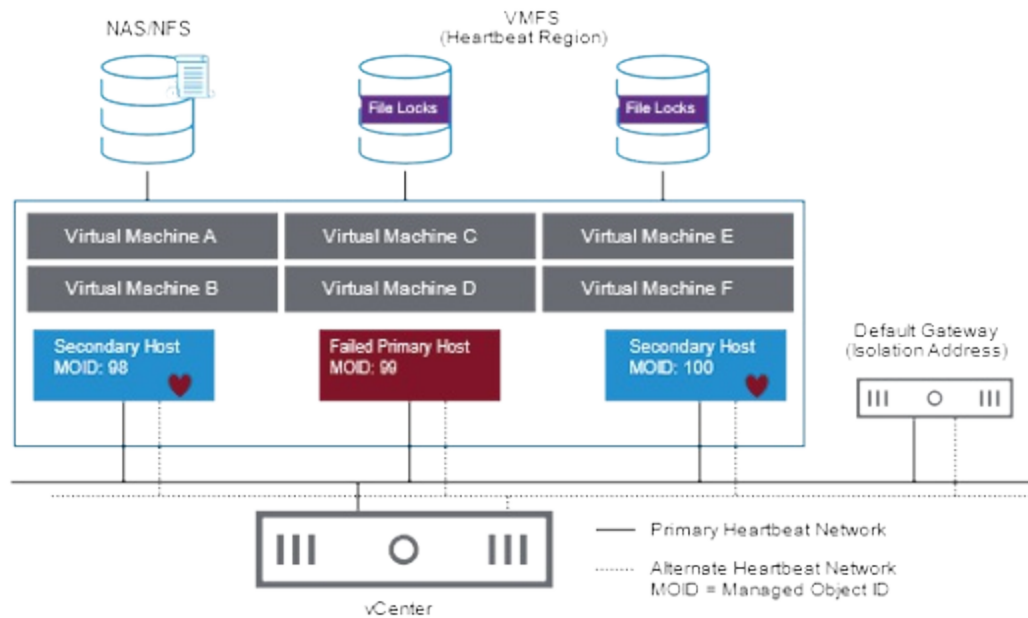


Figure 9.37: Failed primary host

(Source: VMware)

Isolated hosts

A host is considered isolated when:

- It no longer receives network heartbeats from other hosts.
- It cannot ping its configured isolation address (the default is the default gateway).

In this state, the host triggers the vSphere HA isolation response, which may power off VMs on the isolated host so they can be restarted on a healthy host. The primary host uses datastore heartbeats to confirm if the isolated host is still active. This helps differentiate between a host that has failed and one that is merely isolated or partitioned. Datastore heartbeats are checked only during such failure conditions.

The following figure illustrates the isolated host:

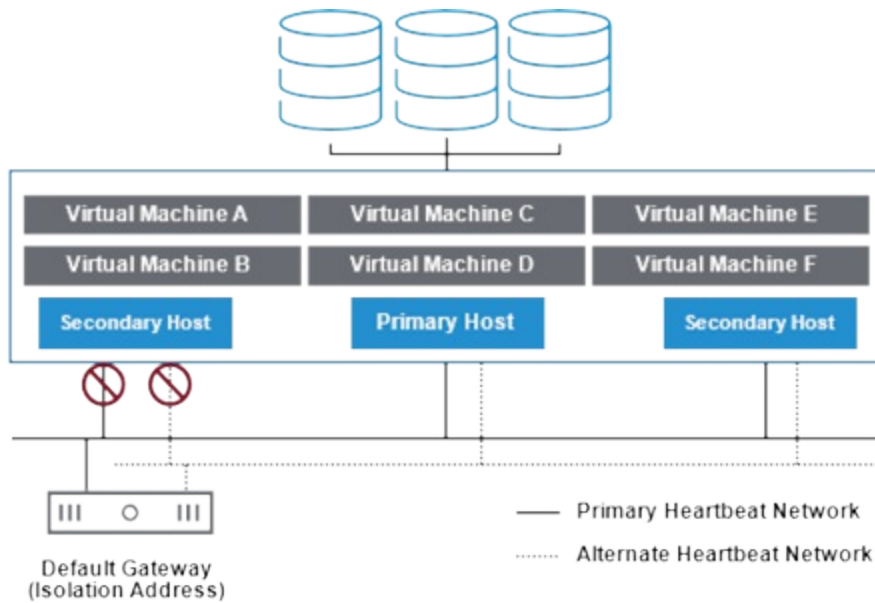


Figure 9.38: *Isolated host*

(Source: VMware)

VM storage failures

Storage connectivity difficulties can arise due to:

- Network or switch failures
- Storage array misconfiguration
- Power outages.

These issues have an impact on the availability of VMs, making affected VMs harder to administer and causing errors in apps with associated virtual drives.

The following figure illustrates the VM storage failure:

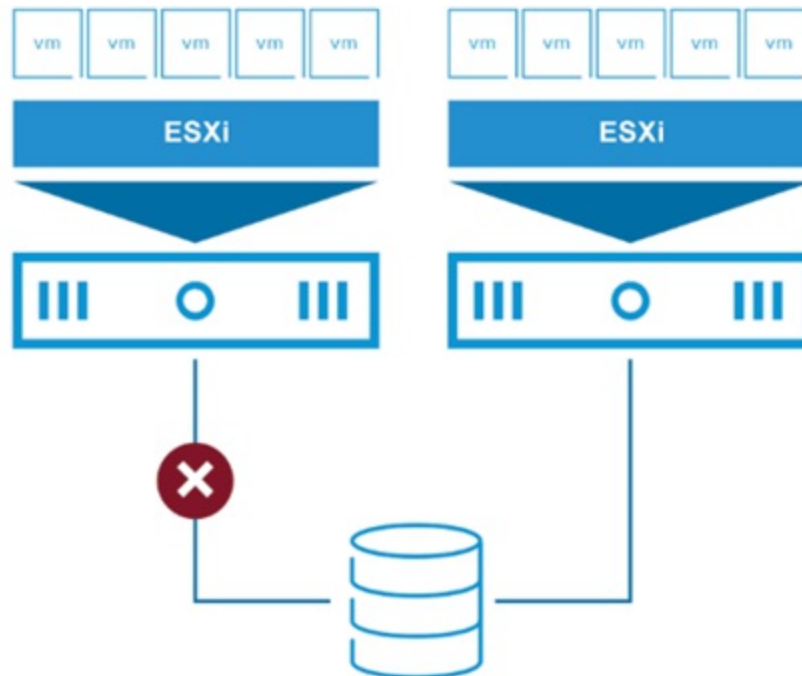


Figure 9.39: VM storage failure

(Source: VMware)

Protecting against storage failures with VMCP

VMCP safeguards granular virtual workloads, in conjunction with vSphere HA, overrides storage-based weaknesses, and ensures logical workload accessibility in a vCenter region.

Automated responses on datastore accessibility failure include the following:

- Detection of an outage through VMCP.
- Enabling automated actions such as event alarming.
- Executing VM restarts on operational and unaffected hosts.

VMCP guarantees consistent endurance of fundamental VM services despite interruptions in storage paths.

The following figure illustrates the VMCP:

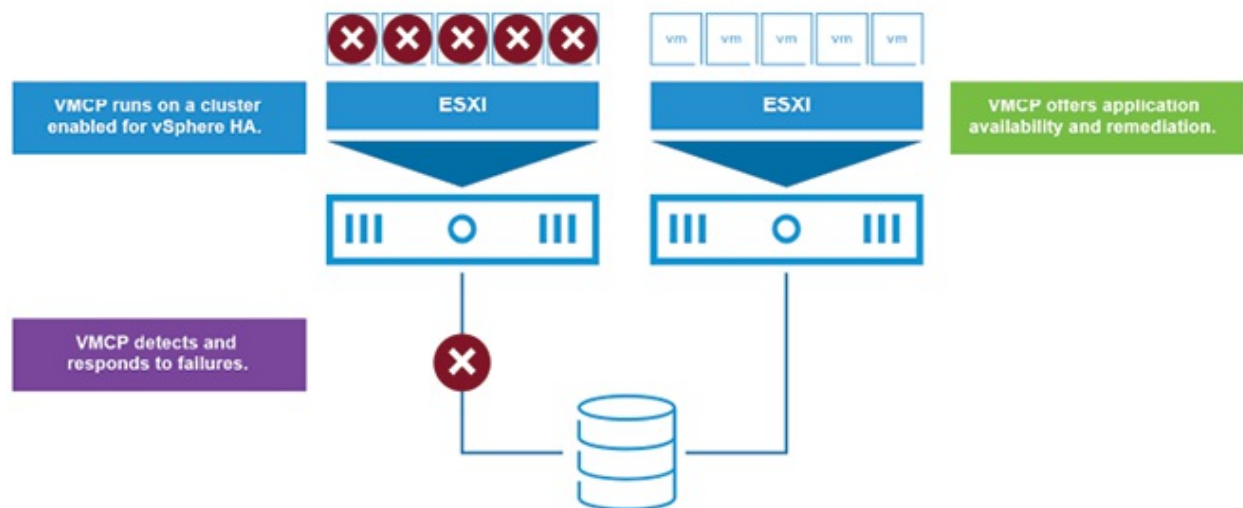


Figure 9.40: About VMCP

(Source: VMware)

Additional vSphere HA design approaches

While extending high availability and reducing co-located false isolation scenarios in a vSphere HA cluster, consider the following design pointers:

- Make use of redundant heartbeat networks with isolation addresses.
- Physically segment VM networks from quiet heartbeat networks.
- Elevate separation from services and enhance visible issues, also known as fault isolation.
- Incorporate redundant intelligent storage area network attachments to mitigate Ethernet disruption loopholes.
 - If utilizing NFS, iSCSI, or **Fibre Channel over Ethernet (FCoE)**, ensure the IP storage network is spatially separated from the HA heartbeat network.

Adhering to these standards improves the dependability and fault tolerance of the vSphere HA ecosystem.

vSphere HA configuration requirements

Ensure that the following prerequisites are met before configuring a vSphere HA cluster:

- **Standardized IP allocation:** Each host must have a static IP address or

an IP address assigned by **Dynamic Host Configuration Protocol (DHCP)** that remains the same after rebooting.

- **Shared common networks for heartbeat:** Every host should have at least one common shared heartbeat network.
- **VMware Tools is installed:** VM monitoring functionality requires VMware Tools to be installed on all VMs.
- **Cluster size limits:** The number of hosts should not be exceeded beyond the supported maximums.

Check <https://configmax.broadcom.com/home> for the most recent limits on cluster size and configuration metrics.

Configuring vSphere HA settings

When setting up a vSphere HA cluster in the vSphere Client, configure the following:

- **Failure and responses:** Set responses for host failure, isolation, VM monitoring actions, and **VM Component Protection (VMCP)** provisioning.
- **Admission control:** Activate or deactivate admission control and set the rule under which resources will be reserved from the cluster for failover.
- **Heartbeat datastores:** Set preferences for datastores used in the datastore heartbeat.
- **Advanced options:** Enable additional configurations to adjust the behavior of HA on vSphere.

The following figure illustrates the vSphere HA settings configuration:

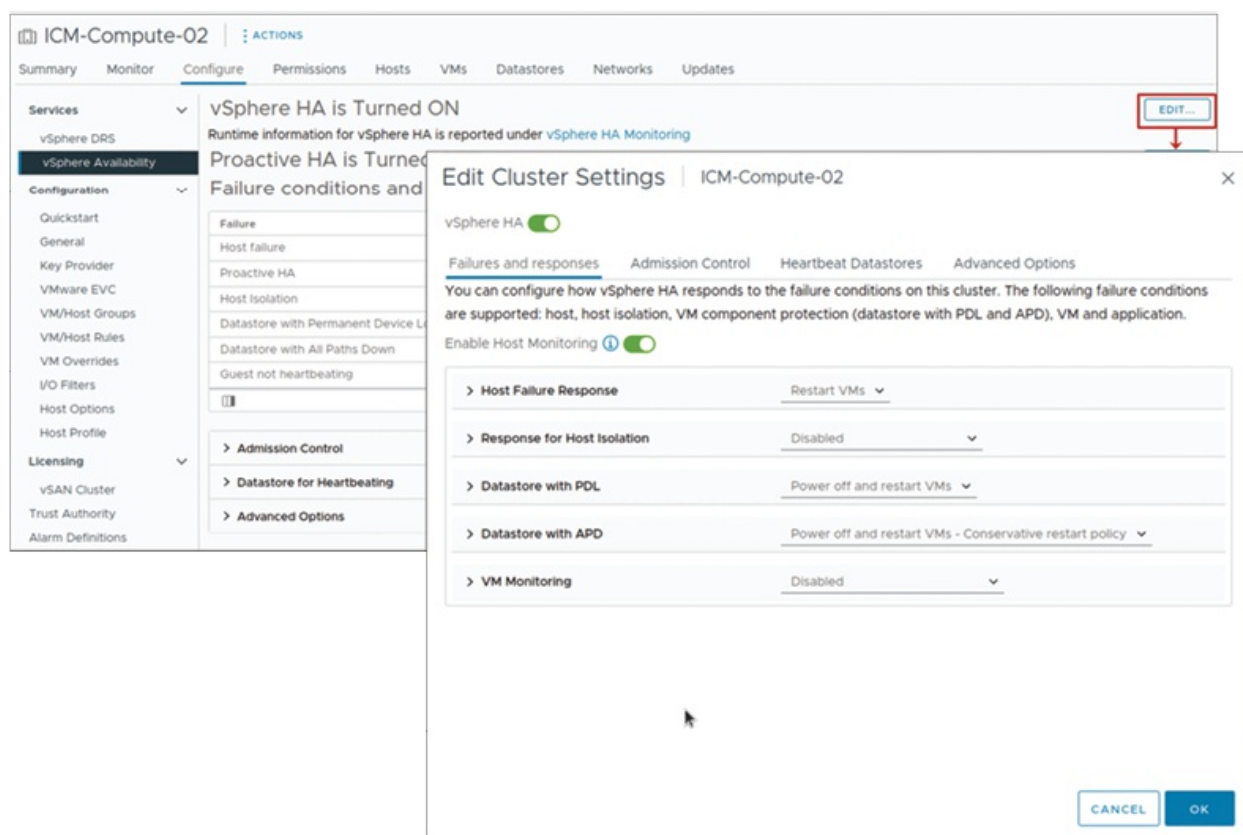


Figure 9.41: Configuring vSphere HA settings

(Source: VMware)

vSphere HA settings: failures and responses

Use the **Failures and Responses** pane in vSphere HA to configure how the cluster reacts to various failure scenarios:

- **Host failure responses:**
 - **Disabled:** Host monitoring is off; VMs are not restarted.
 - **Restart VMs:** Restarts VMs based on their restart priority.
- **Host isolation responses:**
 - Disabled
 - Power Off and Restart VMs
 - Shut Down and Restart VMs
- **Permanent Device Loss (PDL):**
 - Disabled

- **Issue events:** Admin is notified; no action taken.
- Power Off and Restart VMs
- **All-Paths-Down (APD):**
 - Disabled
 - **Issue events:** Admin is notified.
 - **Power Off and Restart VMs (Conservative):** Only restarts if another host has confirmed capacity.
 - **Power Off and Restart VMs (Aggressive):** Stops VMs regardless of known restart capacity, risking data loss in partitioned clusters.
- **VM monitoring options:**
 - VM monitoring only
 - VM and application monitoring

The following figure illustrates the failure and responses in vSphere HA settings:

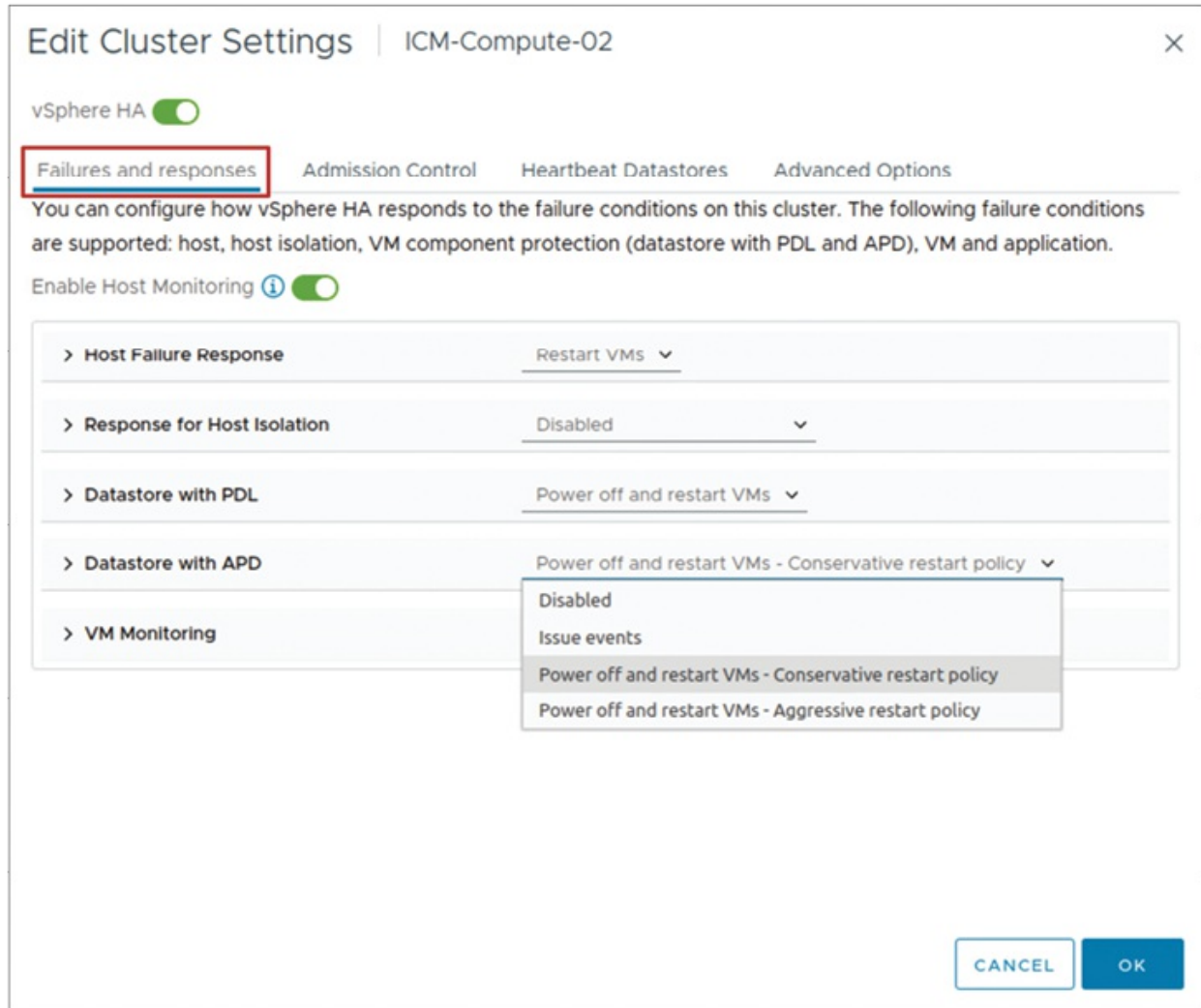


Figure 9.42: vSphere HA settings, failure and responses

(Source: VMware)

vSphere HA settings for default VM restart priority

The restart priority of a VM determines how vSphere HA will restart VMs on a host based on failure types:

- **Default level:** All VMs configured are given a medium priority as the default level.
- **Custom level overrides:** This can be overridden by setting user-defined policies.

When a host is down, VMs get assigned to other hosts that have resources, but whose resources are not currently reserved. The servicing order is first

served to high-priority VMs, then to low-priority VMs. If the available capacity in the cluster is low, then vSphere HA must wait for some time until the capacity becomes available, such as when the host recovers, and then it can be placed properly.

To ensure availability, use admission control policies by reserving some of the failover resources. Reconfigure restart conditions, such as managing:

- Resource allocation
- Power on state
- Heartbeat detection from VMware tools or application.

The following figure illustrates the default VM restart priority in vSphere HA settings:

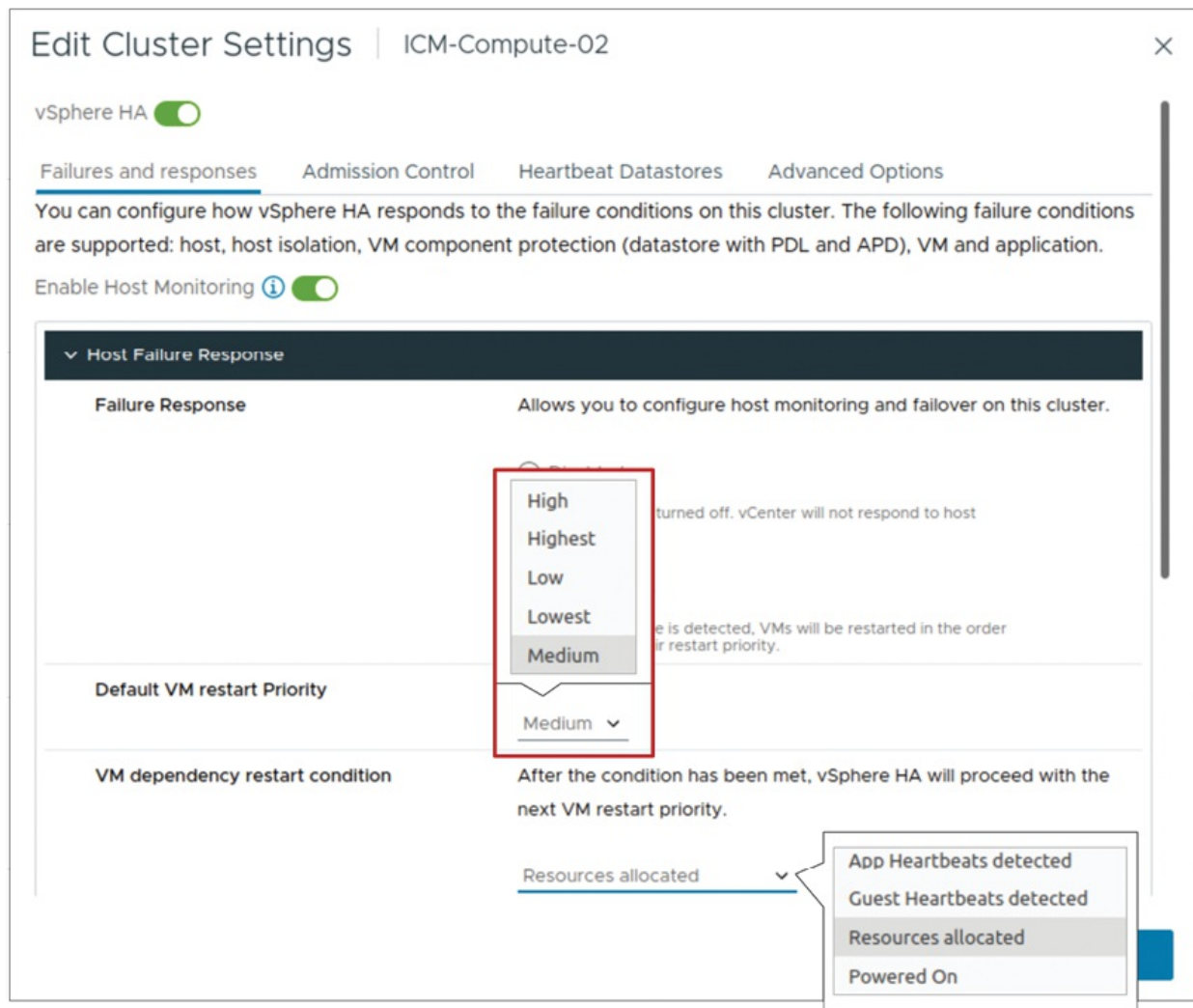


Figure 9.43: vSphere HA settings—default VM restart priority

(Source: VMware)

Individual VMs' restart priority can be customized to override the default level set for the cluster.

The following figure illustrates the VM level settings in vSphere HA settings:

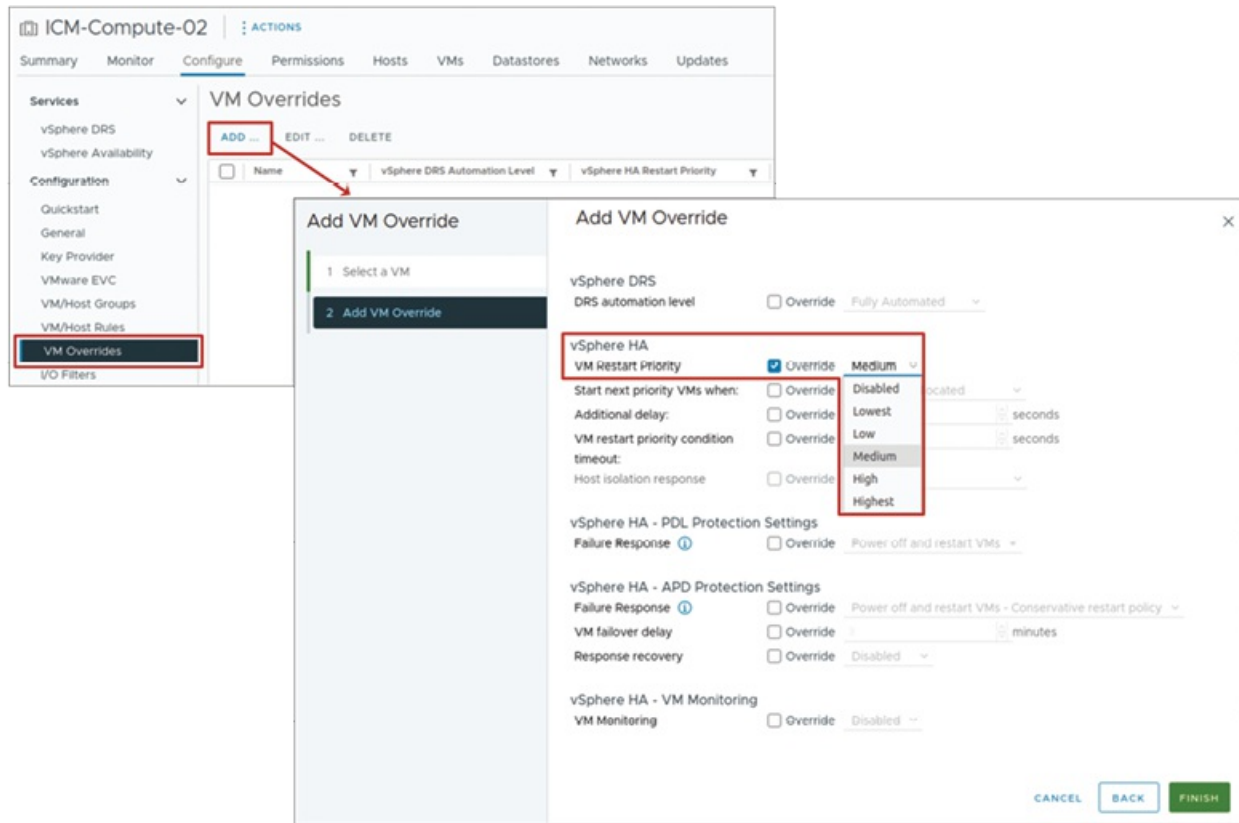


Figure 9.44: vSphere HA settings—VM level settings

(Source: VMware)

About vSphere HA orchestrated restart

Orchestrated restart is an alternative to using basic VM restart priority settings in vSphere HA. It enables administrators to control the restart sequence of VMs based on application or service dependencies.

The following details explain the functionality and implementation of orchestrated restart:

- **Key features:**
 - It ensures that dependent services start in the correct order, for example, database | application | and web tier.

- It is especially useful for multi-tier applications, such as three-tier architectures.
- **Way it works:**
 - Administrators define explicit dependencies between VMs.
 - vSphere HA restarts VMs in the specified order, based on those dependencies.
- **Important constraints:**
 - Only direct dependencies are allowed.
 - Cyclical dependencies, where VMs depend on each other in a loop, can cause restart failures.

The following figure illustrates the orchestrated restart three tier:

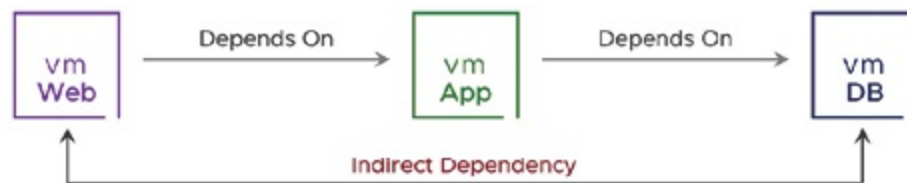


Figure 9.45: Orchestrated restart three-tier

(Source: VMware)

Configuring orchestrated restart

To ensure VMs restart in a specific order (e.g., DB | App | and Web), perform the following steps:

1. **Create VM groups:**
 - a. DB-VMs | contains DB-01
 - b. App-VMs | contains App-01
 - c. Web-VMs | contains Web-01, Web-02
2. **Create restart dependency rules (VM to VM):**
 - a. Use the VM/Host Rules section under the cluster's Configure tab.
 - **Rule 1: DB VMs before App VMs:**
 - **Name:** DB VMs | App VMs
 - **Type:** VMs to VMs

- **First group:** DB-VMs
- **Second group:** App-VMs
- **Meaning:** DB-VMs must start before App-VMs.
- **Rule 2: App VMs before Web VMs:**
 - **Name:** App VMs | Web VMs
 - **Type:** VMs to VMs
 - **First group:** App-VMs
 - **Second group:** Web-VMs
 - **Meaning:** App-VMs must start before Web-VMs.

Note: Only direct dependencies are supported. Avoid circular dependencies, which VMs from restarting.

The following figure illustrates the orchestrated restart configuration:

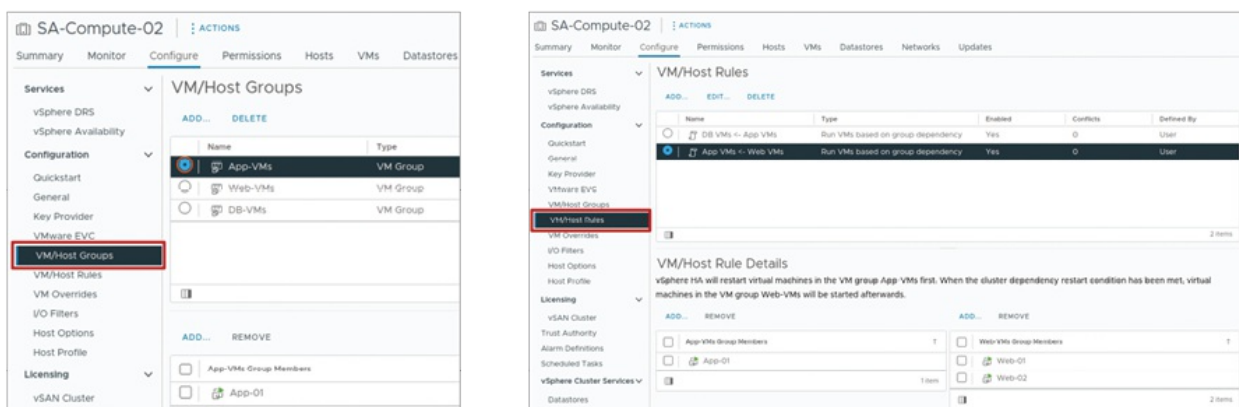


Figure 9.46: Configuring orchestrated restart

(Source: VMware)

vSphere HA settings for VM monitoring

VM monitoring in vSphere HA enables automated recovery of VMs when system or application-level failures are detected. By default, this setting is disabled, but it can be configured based on workload sensitivity.

vSphere HA considers a VM to have failed under the following conditions:

- No VMware Tools heartbeats are detected.
- No I/O activity is observed for a default period of 2 minutes.

When a failure is detected, vSphere HA resets the affected VM in an attempt to restore services.

The monitoring sensitivity level can be tuned as shown in the following list:

- **High sensitivity:** Faster failure detection and response but may lead to false positives if heartbeats are missed temporarily due to resource constraints.
- **Low sensitivity:** Reduces the chance of false alarms but may delay recovery from actual failures.

Administrators can choose a setting that best fits their workload needs and tolerance for brief interruptions.

If VM and application monitoring are selected, the system also monitors application-level heartbeats (via VMware Tools integration or supported SDKs). If an application fails to send its heartbeat, vSphere HA triggers a VM reset to recover services.

The following figure illustrates the **VM Monitoring** in vSphere HA settings:

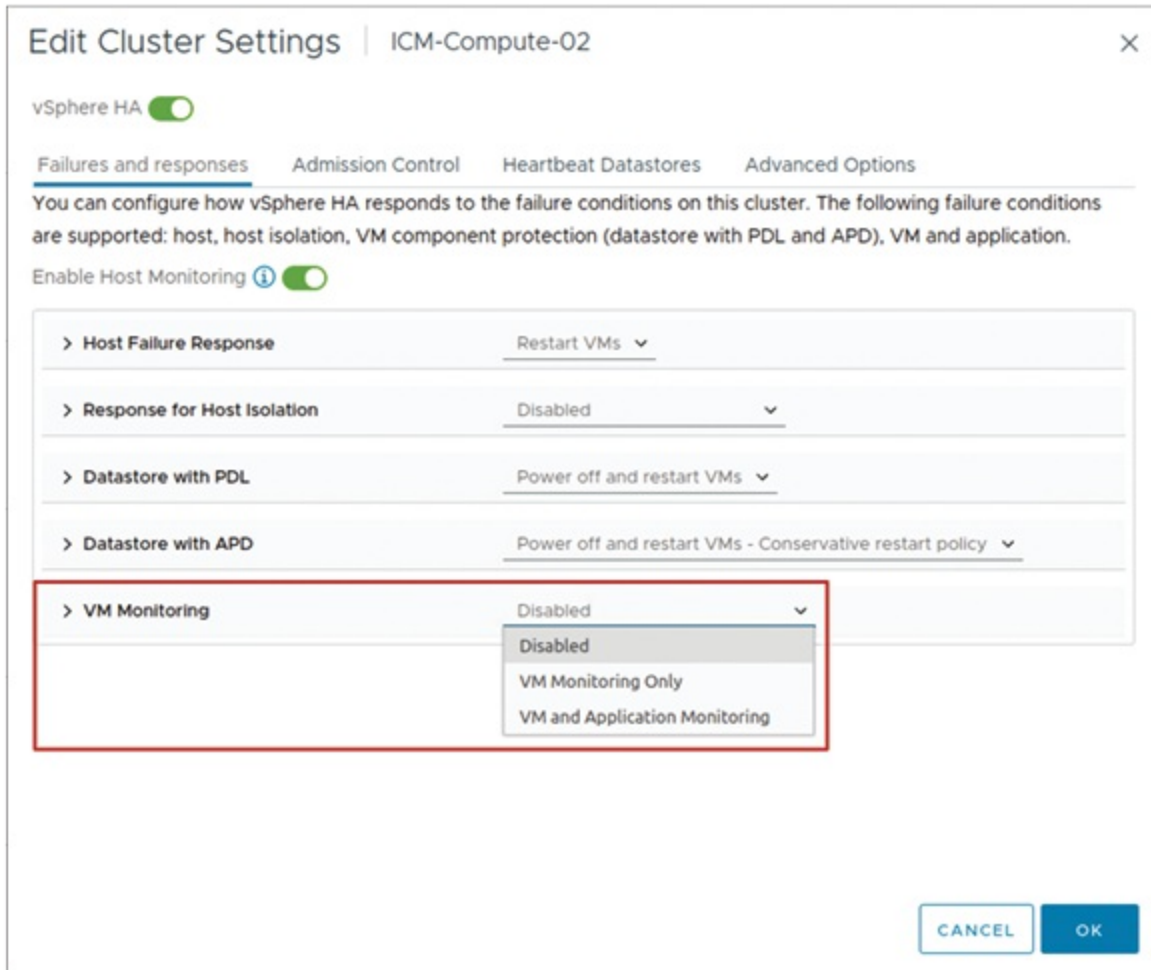


Figure 9.47: vSphere HA settings, VM monitoring

(Source: VMware)

vSphere HA settings for admission control

Within vSphere HA, admission control is an essential mechanism that ensures the cluster can meet the availability goals during host failures. It prevents the powering on of VMs if doing so would compromise the cluster's capability to recover from failures.

With admission control enabled, vCenter guarantees the following:

- There are sufficient resources available for all VMs to be restarted during host outages.
- All protected VMs receive their reserved (CPU and memory) allocation.

Admission control settings are as follows:

- **Disabled:** Completely removes the admission control policy. While all

VMs can be powered on at the same time, it is allowed and controlled irrespective of the cluster's capacity. This option does remove guaranteed failover protection. It is not advisable for production settings.

- **Slot policy:** Determines the upper limit of resources assigned on a per-VM basis and uses slots as memory and CPU resources deducted from the cluster. The host's dominant support determines the cluster's failover ability based on the cluster's designated number of slots. This is a more rigid and less flexible policy for mixed VM sizes.
- **Cluster resource percentage (Default):** Allocates a set percentage of CPU and memory resources for the entire cluster to be set aside for failover within the cluster. It is the default and most flexible approach that adapts to the needs and demands of different VM resources and varying workloads.
- **Dedicated failover hosts:** Designates one or more specific hosts solely for VM recovery during failures. These hosts do not run regular workloads under normal operations. If the designated hosts lack sufficient resources, failovers may still occur on other available hosts.

The following figure illustrates the admission control in vSphere HA settings:

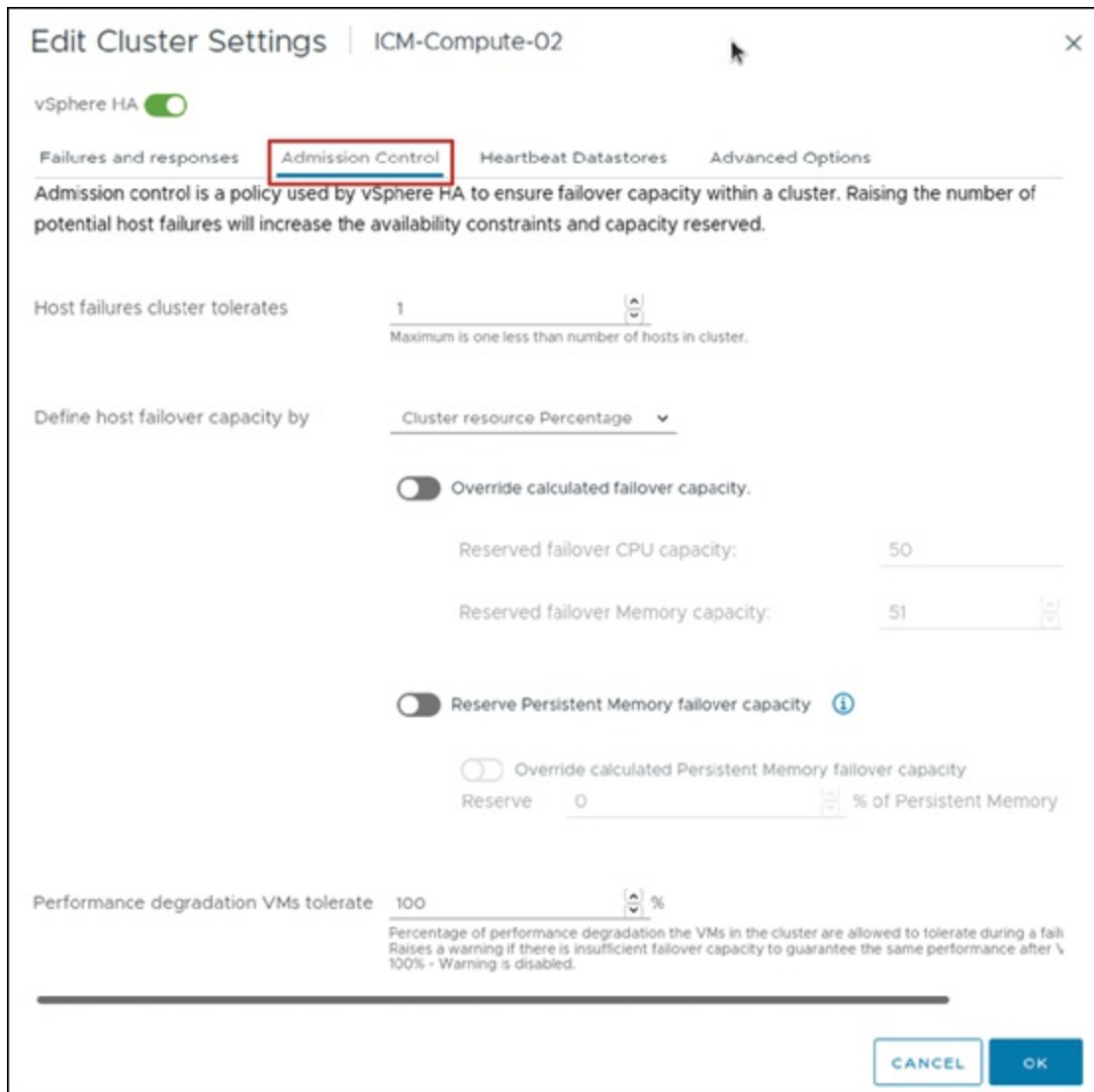


Figure 9.48: vSphere HA settings—Admission control

(Source: VMware)

Admission control using cluster resource percentage

When the default cluster resource percentage admission control policy is applied, vSphere HA continuously checks how much failover capacity there is in terms of actual-time resource utilization.

Let us take an example.

- **Total cluster capacity:**
 - **CPU:** 18 GHz
 - **RAM:** 24 GB

- **Total VM reservations:**
 - **CPU:** 7 GHz
 - **Memory:** 6 GB
- **Failover capacity calculation:**
 - **CPU failover capacity:**
 - $((18 \text{ GHz} - 7 \text{ GHz})/18 \text{ GHz}) = 61\%$
 - **Memory failover capacity:**
 - $((24 \text{ GB} - 6 \text{ GB})/24 \text{ GB}) = 75\%$

This implies that, from existing reservations, the cluster has 61% CPU and 75% memory capacity available to allow for failover scenarios. Such percentages are dynamically tuned as hosts or VMs are added or removed from the cluster, thereby ensuring consistent meet of failover protection needs.

The following figure illustrates the admission control using cluster resource percentage:

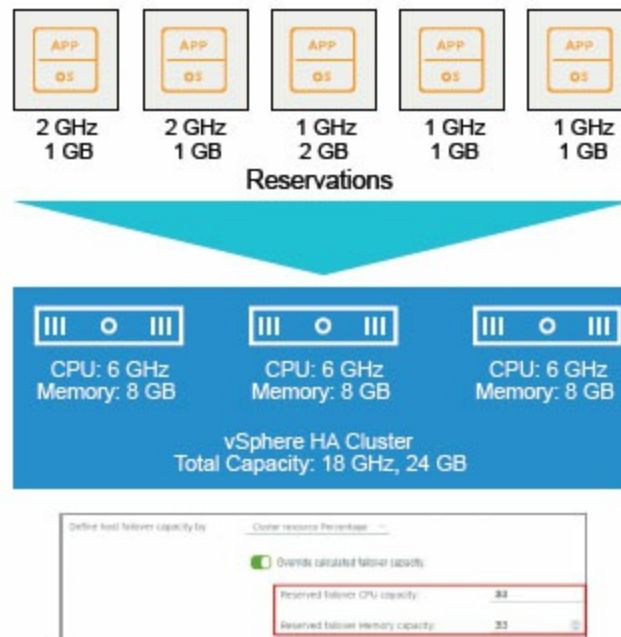


Figure 9.49: Admission control using cluster resource percentage

(Source: VMware)

Admission control using slots

A slot is determined by aggregating the maximum memory and CPU reservations of any operating VM in the cluster as shown in [Figure 9.50](#).

vSphere HA handles admission control by calculating the following values.

Slot size: In this case, the slot capacity is 2 GHz CPU and 2 GB memory.

- Cluster hosts can contain up to three slots each: The cluster consists of nine slots (3 + 3 + 3).

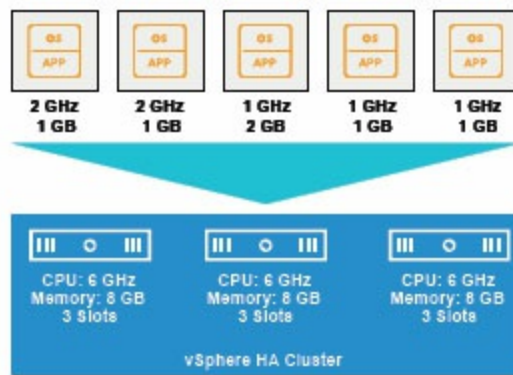


Figure 9.50 Admission control using slots

(Source: VMware)

vSphere HA also determines the current failover capacity. In this case, the failover capacity is just one host:

- If the first host fails, the cluster still has six slots available, which is enough to accommodate all five powered-on VMs.
- If the first and second hosts fail, there are only three slots available, which is inadequate to accommodate all five VMs.
- If the current failover capacity is less than the configured failover capacity, vSphere HA does not enable any additional VMs to boot up.

vSphere HA settings for performance degradation VM toleration

This configuration enables an administrator to set boundaries on the performance degradation that can be tolerated by the cluster's VMs in the event of a host failure in the system.

Following are the key aspects:

- **Key points:**
 - **Purpose:** This setting assists vSphere HA policies in defining the net resource consumption with respect to the failover capacity and alerts if set thresholds are crossed.
 - **Threshold (%):** This determines the amount of permitted degradation in performance.
 - **Default:** 100%—No warning is issued.
 - **0%:** Excessive use will result in a warning being generated.
 - **20%:** Actual usage that is 20% less than the available failover capacity is exceeded.
- **Example:** If cluster usage is at 10 GHz and the threshold is set to 20%, then:
 - Performance decrease can be calculated by: $10 \text{ GHz} \times 20\% = 2 \text{ GHz}$
 - If the available capacity is less than $(10 \text{ GHz} - 2 \text{ GHz} = 8 \text{ GHz})$, a configuration notice is issued.

This option becomes available when vSphere DRS has been enabled.

The following figure illustrates the performance degradation VMs tolerate in vSphere HA settings:

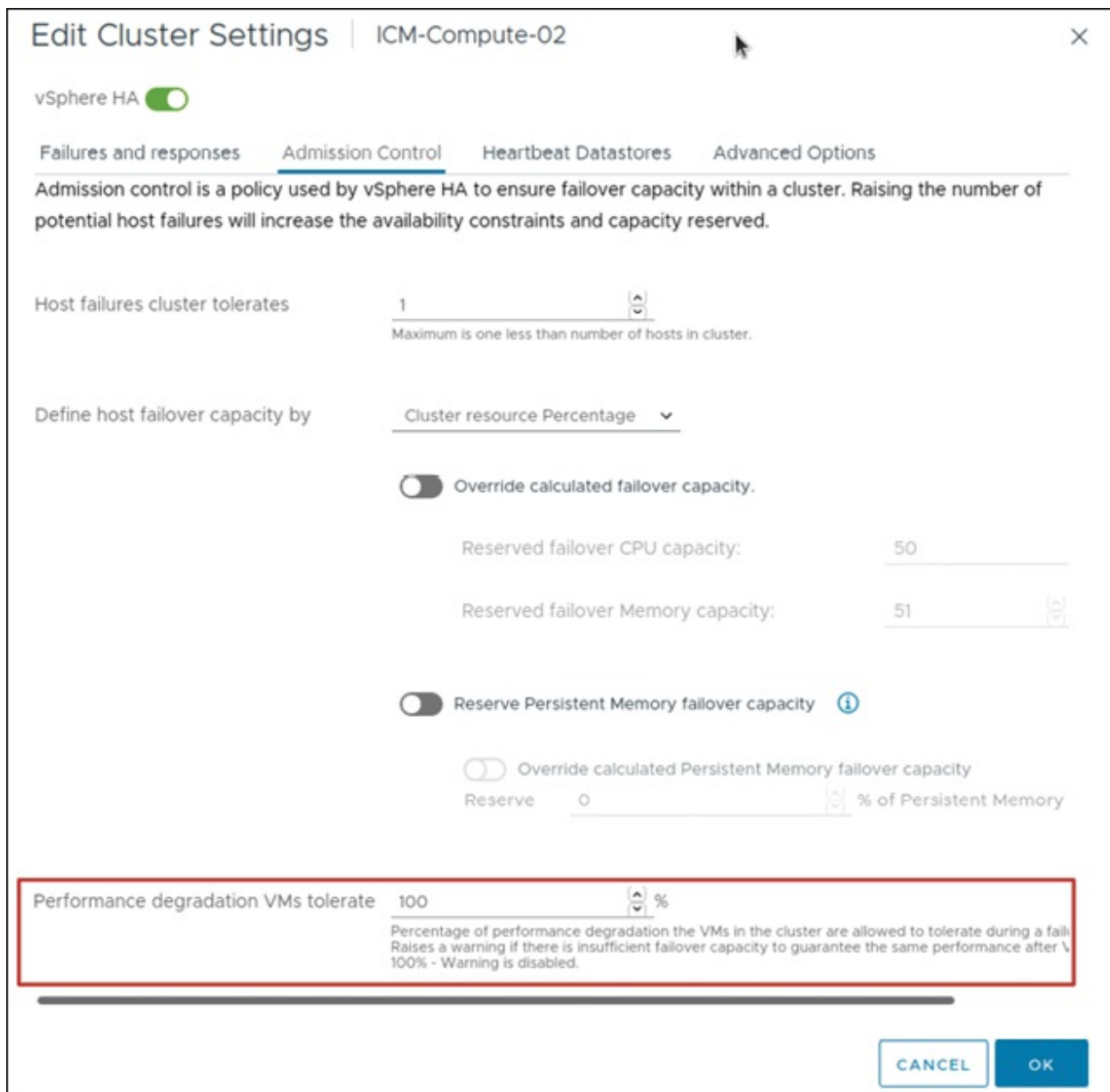


Figure 9.51: vSphere HA settings for performance degradation VMs tolerate

(Source: VMware)

vSphere HA heartbeat datastores settings

Datastore heartbeating provides an additional mechanism for vSphere HA to determine whether a host is isolated or has failed, especially when network heartbeat communication is lost.

The following explains the operational details of datastore heartbeating:

- **Way it works:**

- vSphere HA writes heartbeat files to shared datastores.
- If a host loses network connectivity but still updates the heartbeat file, it's considered alive.
- If both network and datastore heartbeats are missed, the host is considered failed, and VMs are restarted on other hosts.
- **Supported datastore types:**
 - VMFS
 - NFS
 - **vSphere Virtual Volumes (vVols)**

Note: vSAN datastores is not supported for datastore heartbeating.

vSphere HA increases host health monitoring by looking at more than just the heartbeat network. Administrators can specify the datastores to utilize for datastore heartbeating or let vSphere HA decide.

The following figure illustrates the heartbeat datastore in vSphere HA settings:

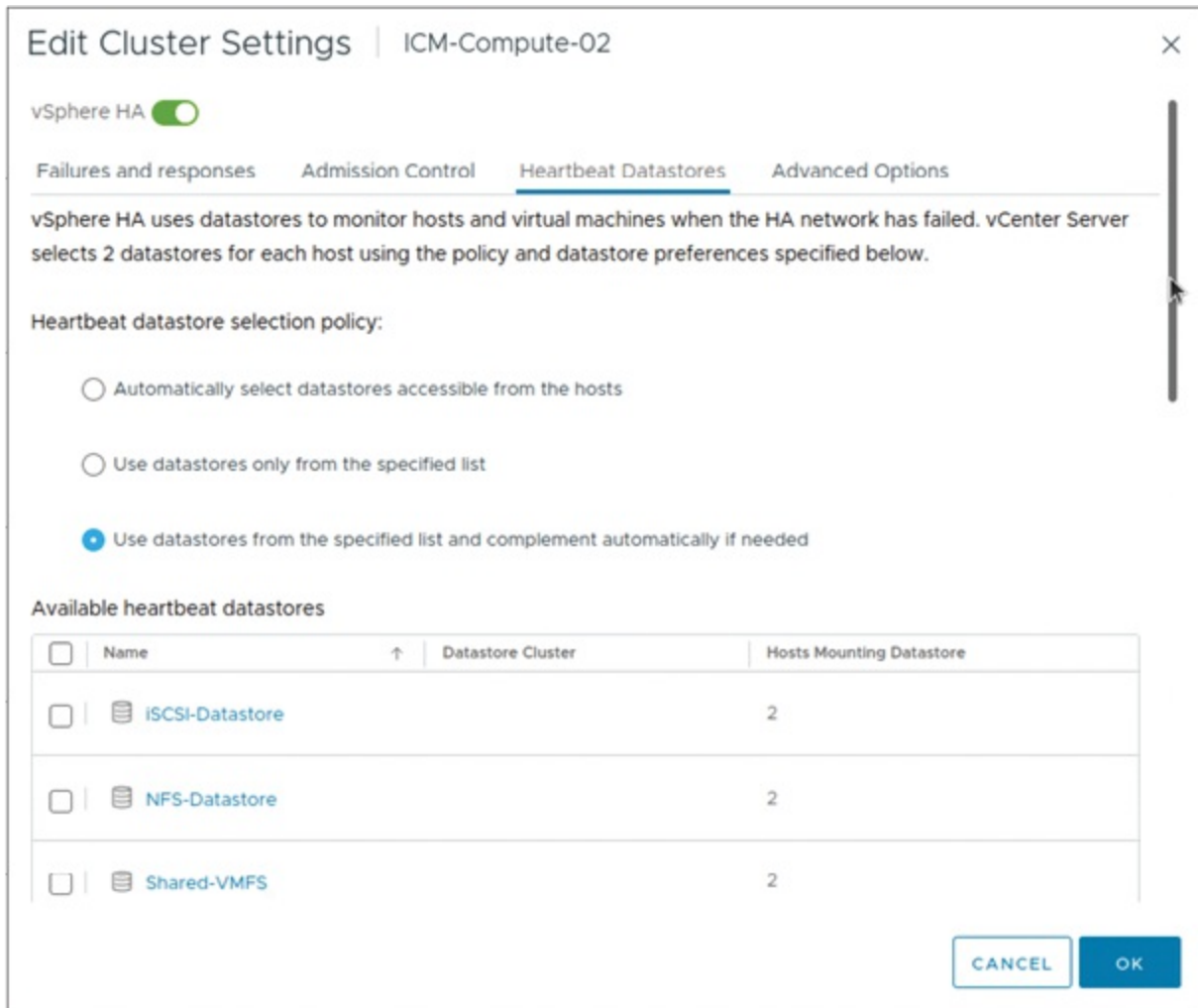


Figure 9.52: vSphere HA settings, heartbeat datastore

(Source: VMware)

vSphere HA advanced options settings

Administrator can tweak advance vSphere HA options to customize vSphere HA behaviour. Few examples are as follows:

- **Force a cluster not to use the default isolation address (default gateway):**
 - **Option:** das.usedefaultisolationaddress
 - **Value:** False
- **Force a cluster to ping alternate isolation addresses:**
 - **Option:** das.isolationaddressX (where X is a number, e.g., 1, 2...)

- **Value:** An IP address or FQDN (e.g., das.isolationaddress1 = 192.168.1.1)
- **Force a cluster to wait beyond the default 30-second isolation action window:**
 - **Option:** das.config.fdm.isolationPolicyDelaySec
 - **Value:** Any number greater than or equal to 30 (e.g., 60 seconds)
- **Force maximum bound on the memory slot size:**
 - **Option:** das.slotmeminmb
 - **Value:** Minimum memory size in MB (e.g., 100)
- **Force maximum bound on the CPU slot size:**
 - **Option:** das.slotcpuinmhz
 - **Value:** Minimum CPU size in MHz (e.g., 32)

Network configuration and maintenance

Deactivate host monitoring before modifying VMkernel ports used for management or vSAN traffic avoids unintended VM failovers due to false heartbeat loss.

- Perform these steps when changing network hardware or settings:
 - Suspend host monitoring.
 - Place the host into maintenance mode.
 - These steps help prevent false detection of host failure due to dropped HA heartbeats.
- Deactivating host monitoring is necessary only when modifying virtual networking components involving:
 - **VMkernel ports configured for:**
 - Management traffic
 - vSAN traffic
- Reconfigure vSphere HA on all hosts in the cluster if any of the following is performed:

- Change the networking configuration on ESXi hosts
- Add port groups
- Remove virtual switches
- Suspend host monitoring
- After completing network changes:
 - vSphere HA reconfiguration causes network information to be reinspected.
 - Then, must reactivate host monitoring.

The following figure illustrates the network configuration and maintenance:

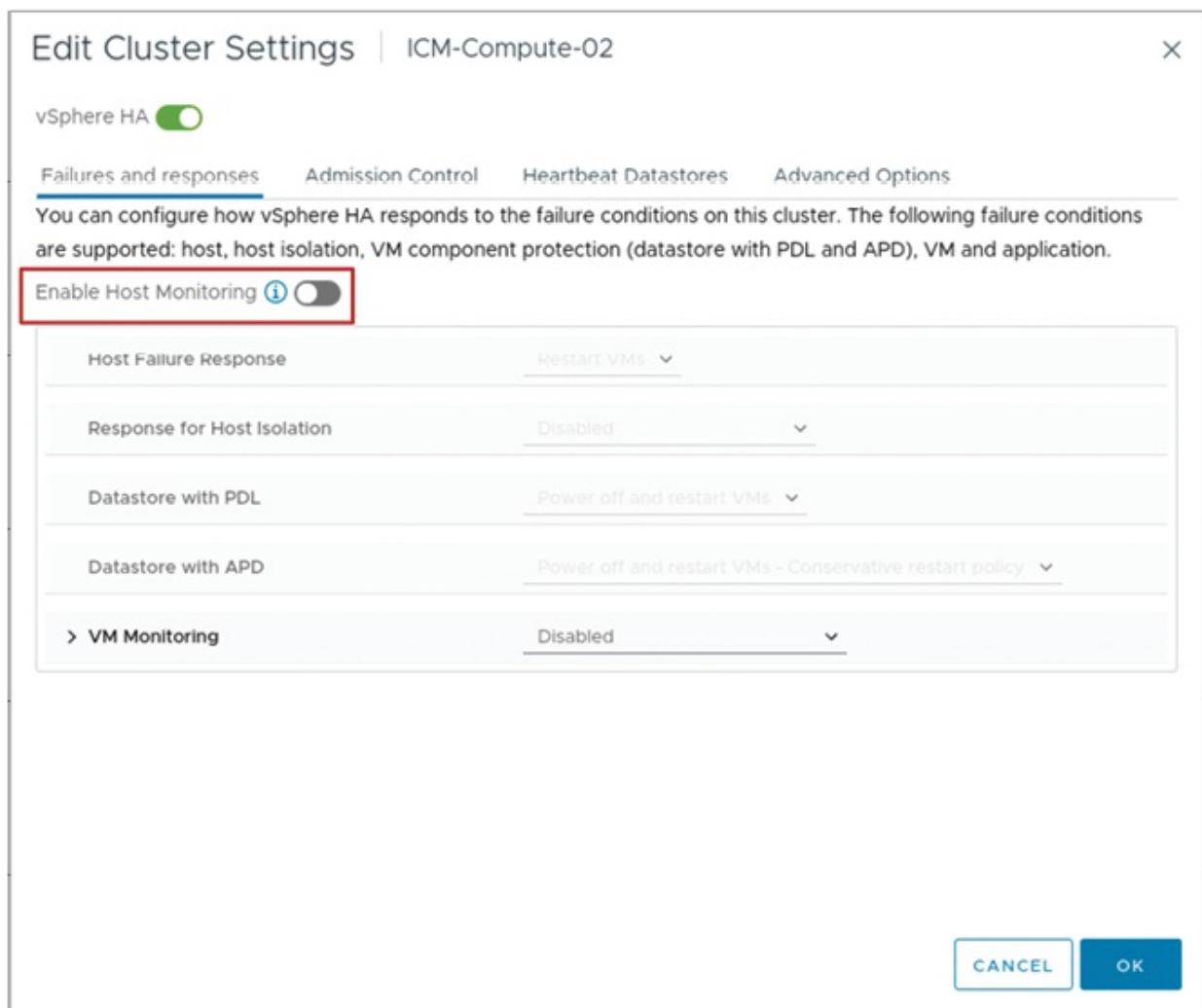


Figure 9.53: Network configuration and maintenance

(Source: VMware)

Monitoring vSphere HA cluster status

The status of a vSphere HA cluster can be monitored from the **Summary** page under the **Monitor** tab in the vSphere Client.

The following figure illustrates the vSphere HA cluster status:

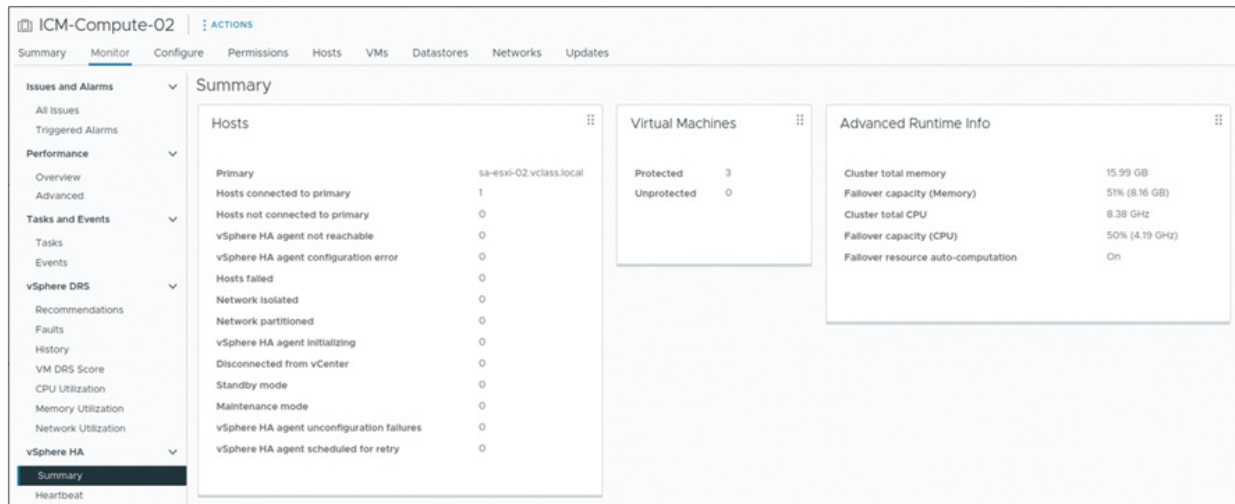


Figure 9.54: Monitoring vSphere HA cluster status

(Source: VMware)

Using vSphere HA with vSphere DRS

vSphere HA integrates with vSphere DRS to improve failover efficiency and resource utilization. During a failover, vSphere HA verifies resource availability on hosts; if insufficient, it requests DRS to assist with VM placement. Failover may not occur if:

- Admission control is disabled
- Remaining hosts lack sufficient resources to power on all failed VMs

After HA restarts VMs on available hosts (often overloading them), DRS automatically balances the load across the cluster by migrating VMs to optimize performance.

Embracing vSphere Fault Tolerance

vSphere **Fault Tolerance (FT)** provides continuous availability for mission-critical VMs by ensuring:

- Zero downtime
- Zero data loss
- Uninterrupted network connectivity

Built on the ESXi host platform, vSphere FT creates a secondary VM that mirrors the primary VM's execution in real time on a different host. The primary and secondary VMs continuously monitor each other's status to ensure protection is maintained. If the host running the primary VM fails, the secondary VM takes over instantaneously and transparently, ensuring uninterrupted service. A new secondary VM is then automatically created on another host to restore fault tolerance redundancy. Similarly, if the host running the secondary VM fails, a replacement secondary is instantiated without affecting the primary VM.

The following figure illustrates the vSphere Fault Tolerance:

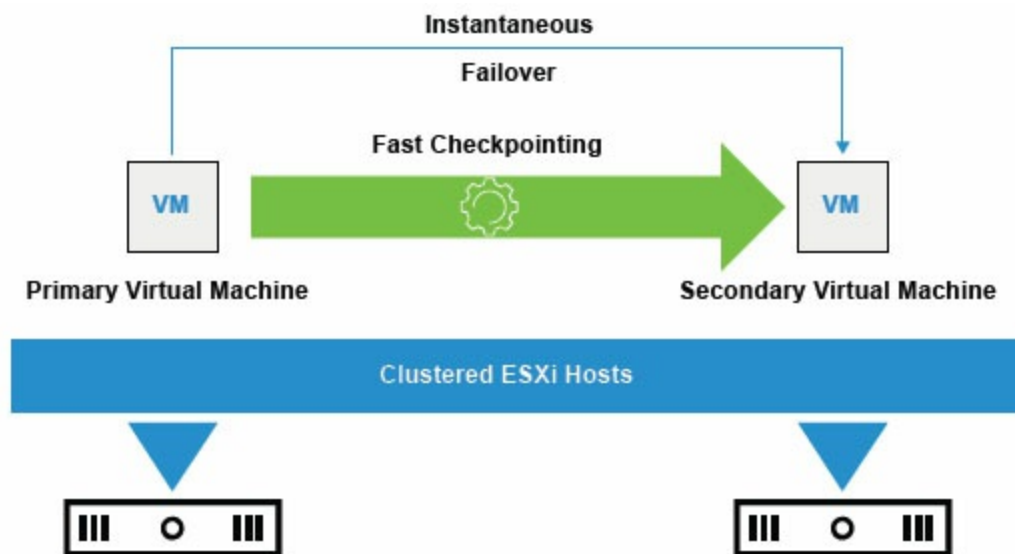


Figure 9.55: About vSphere Fault Tolerance

(Source: VMware)

vSphere Fault Tolerance with vSphere HA and DRS

Both vSphere HA and vSphere DRS are integrated with vSphere FT, where DRS has an auxiliary role. vSphere HA is essential to FT features, while DRS has a passive role, being informed about FT, deciding which hosts will run, and powering on the primary and secondary VMs. Initially, vSphere DRS

does not actively perform the migration of fault-tolerant VMs to designated hosts. VMs are preserved in a static position to protect the integrity of the failover configuration in place. vSphere FT enforces the primary and secondary VMs to exist on different hosts for added redundancy in the event of a failure. The other limitation is that vSphere vMotion is not permitted to place both FT VMs on the same host to enable seamless availability.

The protection within vSphere FT is unparalleled when it comes to fundamental workloads. It supports VMs with up to 128 GB of memory and 8 vCPUs. It permits up to four fault-tolerant VMs per host, if the cumulative vCPUs do not exceed eight. The solution also supports vSphere vMotion, allowing for the uninterrupted migration of both primary and secondary VMs. To guarantee uninterrupted access, vSphere FT continuously maintains a backup of all VM files and disks. FT can support multiple types of disks, including thin provision, thick provision, lazy-zeroed, and thick provision eager-zeroed, which improves storage flexibility. Additionally, FT works with vSAN environments, allowing the use of fast checkpoint copying, which can always keep the primary and secondary VMS in tight synchronization.

Configuring vSphere FT on a VM

Before activating vSphere FT on a VM, it is essential to first verify the prerequisites at the cluster level. These prerequisites enable the use of FT on a per-VM basis through the vSphere Client. During the enabling process, multiple compatibility validation checks are executed. FT activates within vCenter, which auto-configures a VM's memory limit to *unlimited* and a reservation equal to the assigned memory. While FT is enabled, vCenter does not permit modifications to memory allocation terms, virtual CPU count, or bit-level granularity resolution of these parameters—disk number also cannot be adjusted. Furthermore, the addition or deletion of disks becomes static. The parameters do not revert upon the deactivation of FT.

The following list describes the additional options for a VM with FT:

- **Turn off fault tolerance:** Removes the secondary VM alongside secondary configuration details and history.
- **Suspend fault tolerance:** Stops FT temporarily while retaining the secondary VM in standby for future resumption.

- **Migrate secondary:** Relocates the secondary VM into another designated host.
- **Test failover:** Executes a primary VM failover simulation to assess FT features (permitted only when the VM is online).
- **Test restart secondary:** When the VM is p, the system simulates the failure of a secondary VM to verify the behaviour of **fault tolerance (FT)**.

The following figure illustrates the vSphere Fault Tolerance configuration on VM:

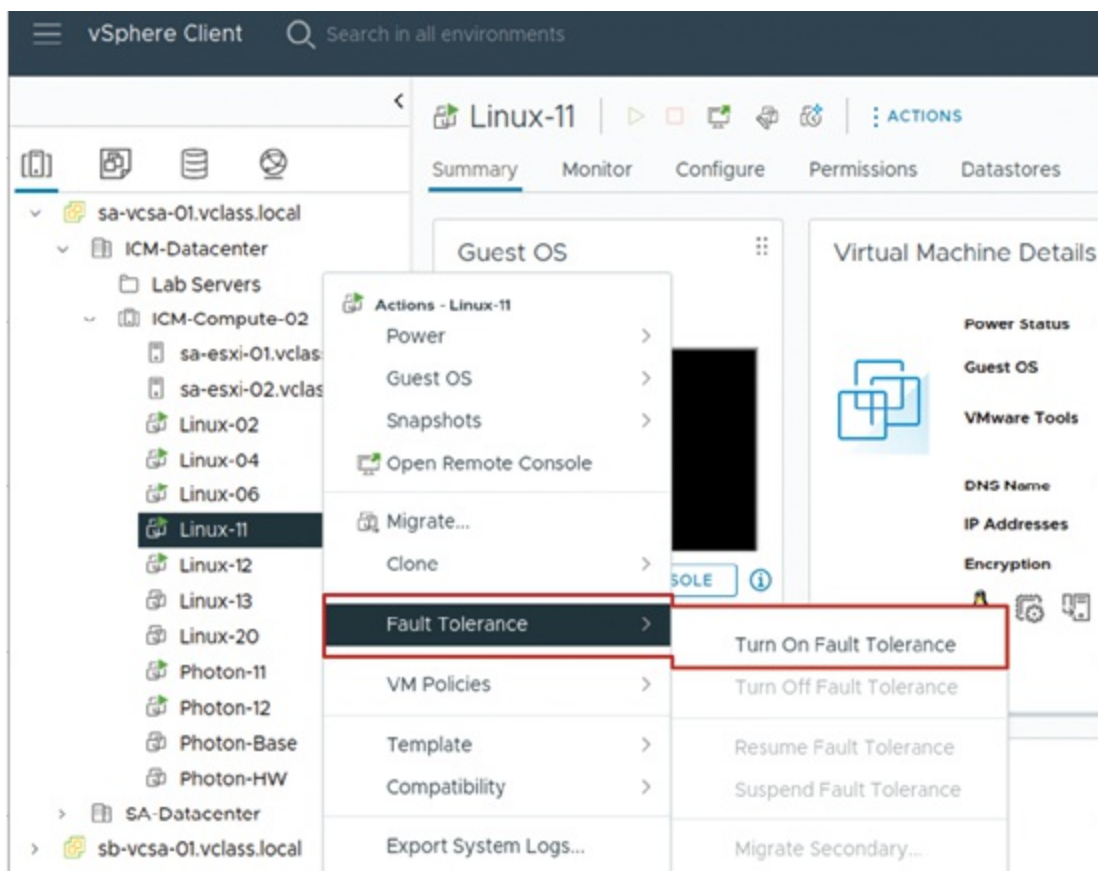


Figure 9.56: Configuring vSphere Fault Tolerance on VM

(Source: VMware)

Conclusion

In this chapter, we discussed the implementation of vSphere clusters and how

they contribute to the maintenance of service availability and performance in virtualized environments. As a starting point, we examined the guides provided through Cluster Quickstart and how to implement better feature deployment and cluster building automation.

We discussed further the basis of vSphere DRS, which balances workloads according to available resources, and vSphere HA, which detects and resolves host and VM failures to minimize downtime and facilitate recovery. Concepts such as heartbeats, isolation responses, and admission control were also discussed as important tools for building robust HA configuration frameworks.

The chapter concluded with an overview of vSphere Fault Tolerance, which provides non-stop outage defense for critical workloads using live VM duplication.

The readers now have the tools of DRS, HA, and FT to create systems that are resource-efficient and available.

In [Chapter 10, Lifecycle Management](#), we will discuss how to manage lifecycle security and updates in vSphere environment. As part of this, we will cover updating vCenter, patching ESXi hosts, and using vLCM to manage VMware Tools and virtual hardware as image-based lifecycle operations.

Points to remember

- When creating a vSphere cluster, administrators can configure vSphere DRS, vSphere HA, vSAN, and image-based lifecycle management for all hosts.
- vSphere DRS functions optimally when VMs meet vMotion requirements, such as shared storage and compatible CPU features.
- The performance degradation VMs tolerate setting is only available when vSphere DRS is enabled.
- vSphere HA protects against multiple failure types, including host failure, guest OS crash, application failure, datastore inaccessibility, and network isolation.
- Redundant HA heartbeat networks can be implemented using NIC

teaming or by adding secondary heartbeat networks.

- When both vSAN and HA are enabled, the vSAN network is used for heartbeat communication instead of the management network.
- Supported datastores for HA heartbeating include VMFS, NFS, and vSphere **Virtual Volumes (vVols)**; vSAN datastores are not supported for heartbeat.
- The HA primary host election process typically completes within 15 seconds.
- vSphere FT offers continuous availability and zero downtime for critical workloads by creating a live shadow VM.
- All vSphere clusters are scalable up to 96 ESXi hosts with vSphere 7 Update 1 or later.

Exercises

1. What are the key features that can be configured when creating a vSphere cluster?
2. How does vSphere DRS help optimize VM performance within a cluster?
3. What are the vMotion requirements that must be met for vSphere DRS to function effectively?
4. List the different types of failures that vSphere HA can protect against.
5. How does vSphere HA use heartbeat networks, and what happens when vSAN is enabled?
6. What is the role of the primary host in a vSphere HA cluster, and how long does the election process typically take?
7. What is vSphere FT, and in what scenarios should it be used over traditional HA?

Lab exercises

1. **Implementing vSphere DRS clusters:** Create a DRS-enabled cluster, configure basic settings using Cluster Quickstart, and verify resource

balancing through vSphere DRS recommendations.

- a. Create a cluster that is configured for vSphere DRS:
 - i. Log into the vSphere Client and navigate to Hosts and Clusters.
 - ii. Right-click the datacenter object and select New Cluster.
 - iii. Enter a name, enable vSphere DRS, and complete the wizard.
 - iv. Use Cluster Quickstart to configure the cluster with desired services.
- b. Verify vSphere vMotion Configuration on the ESXi hosts:
 - i. Go to each ESXi host's configuration tab.
 - ii. Ensure vMotion is enabled on at least one VMkernel adapter.
 - iii. Confirm all hosts use the same network label and IP subnet.
- c. Add ESXi hosts to the cluster:
 - i. Repeat the steps above to configure VMkernel adapter for vMotion on this host
 - ii. Right-click the new cluster and select Add Hosts.
 - iii. Enter the host IPs, credentials, and complete the wizard.
 - iv. Verify that hosts appear in the cluster with no configuration issues.
- d. Modify vSphere DRS settings:
 - i. Select the cluster and navigate to the Configure tab.
 - ii. Go to Services | vSphere DRS and edit the settings.
 - iii. Change the automation level or migration threshold as needed.
- e. Power On VMs and Review vSphere DRS Recommendations:
 - i. Power on multiple VMs in the cluster.
 - ii. Monitor the DRS Recommendations panel under cluster Summary.
 - iii. Review any automatic or manual migration suggestions.
- f. Review vSphere DRS Recommendations When the Cluster Is Imbalanced:
 - i. Introduce a workload imbalance by powering on large VMs on one host.

- ii. Observe how DRS identifies the imbalance and recommends migrations.
- 2. **Configuring vSphere HA:** Enable vSphere High Availability in a cluster, configure redundancy, and test the automated failover response.
 - a. Configure vSphere HA in a cluster:
 - i. Select DRS-enabled cluster.
 - ii. Go to Configure | Services | vSphere Availability.
 - iii. Click Edit, enable vSphere HA, and save the configuration.
 - b. View information about the vSphere HA cluster:
 - i. Under Monitor | vSphere HA, observe HA status and recent events.
 - ii. Note the primary and secondary hosts in the cluster.
 - c. Configure network management redundancy:
 - i. Select each ESXi host and navigate to Configure | VMkernel adapters.
 - ii. Add an additional VMkernel adapter for management traffic.
 - iii. Ensure it is on a different NIC or subnet to provide redundancy.
 - d. Test the vSphere HA functionality:
 - i. Power off or disconnect a host that is running active VMs.
 - ii. Watch as HA detects the failure and restarts VMs on other hosts.
 - e. View the vSphere HA cluster resource usage:
 - i. Go to Monitor | vSphere HA on the cluster.
 - ii. View the Failover Capacity and Cluster Resource Allocation panels.
 - f. Configure the percentage of resource degradation to tolerate:
 - i. Edit vSphere HA settings and go to Admission Control.
 - ii. Select Percentage of cluster resources reserved as failover spare capacity.
 - iii. Set an appropriate CPU and memory reservation threshold.
- 3. **Configuring datastore heartbeating for HA:** Configure additional datastores for heartbeating to improve HA reliability in the event of management network failure.

- a. Review available datastores for heartbeating:
 - i. Go to Cluster | Configure | vSphere Availability.
 - ii. Scroll to the Datastore Heartbeating section.
 - iii. Click Edit to view datastores eligible for heartbeating.
 - b. Add supported datastores to heartbeating:
 - i. Choose two or more supported datastores (VMFS, NFS, or vVols).
 - ii. Ensure vSAN datastores are excluded, as they are not supported for heartbeating.
 - iii. Save the settings and allow changes to propagate.
 - c. Simulate a network isolation scenario:
 - i. Disconnect the management network from one host.
 - ii. Observe HA's behavior using datastore heartbeats to determine host status.
 - d. Review cluster logs and response:
 - i. View recent HA events in Monitor | Events.
 - ii. Confirm that the host was correctly isolated and handled as per configuration.
4. **Implementing vSphere Fault Tolerance (FT):** Enable vSphere Fault Tolerance on a critical VM and validate uninterrupted operation during host failure.
- a. Check prerequisites for vSphere FT:
 - i. Identify a VM with 1 vCPU and no snapshots.
 - ii. Ensure FT-compatible hardware and shared storage across hosts.
 - iii. Verify that the hosts are connected with low-latency FT-compatible networking.
 - b. Enable FT on the selected VM:
 - i. Right-click the VM and select Fault Tolerance | Turn On Fault Tolerance.
 - ii. Choose a compatible host for the secondary VM.
 - iii. Monitor the configuration tasks and validate completion.
 - c. Simulate a host failure to trigger FT response:
 - i. Disconnect or shut down the host running the primary VM.

- ii. Observe automatic failover to the secondary VM without disruption.
- d. Verify continuous availability and logs:
 - i. Access the VM console to verify the workload continued.
 - ii. Go to Monitor | Events and review FT activity logs.

Join our Discord space

Join our Discord workspace for latest updates, offers, tech happenings around the world, new releases, and sessions with the authors:

<https://discord.bpbonline.com>



CHAPTER 10

Lifecycle Management

Introduction

Efficient vSphere lifecycle management ensures stability, security, and optimal performance across virtualized environments. This includes updating the vCenter Server, ESXi hosts, VMware Tools, and virtual hardware while maintaining smooth interoperability with VMware and third-party solutions. **vSphere Lifecycle Manager (vLCM)** simplifies and centralizes the process, allowing for cluster-level control of ESXi hosts via patching, upgrades, and image-based customizations.

Administrators may prevent downtime and assure compliance with the most recent upgrades by knowing vCenter update methods and utilizing vSphere lifecycle manager. This chapter looks at best practices for updating vCenter, managing ESXi hosts, and guaranteeing VM compatibility using structured lifecycle management.

Note: VMware is now part of Broadcom and is known as 'VMware by Broadcom.' All references to 'VMware' in this book reflect this change.

Structure

In this chapter, we will cover the following topics:

- vCenter lifecycle management

- Exploring vSphere lifecycle manager
- ESXi host and cluster lifecycle management
- Managing VMware tools and virtual hardware

Objectives

By the end of this chapter, readers will be able to plan vCenter updates and upgrades using interoperability reports to ensure compatibility with VMware and third-party solutions. They will learn how to apply patches, updates, and upgrades to vCenter while exploring the key features of vLCM for efficient ESXi host and cluster management.

Readers will understand how to import ESXi images into the vLCM image depot, configure patch download sources, activate vLCM in a cluster, and define a cluster image to standardize host configurations. The chapter also covers validating compliance, remediating hosts, and applying recommended cluster images to maintain stability and security.

Additionally, readers will learn how to use vLCM to upgrade VMware Tools and virtual hardware, ensuring VMs benefit from the latest features and performance improvements. By mastering these practices, they will be equipped to maintain a secure, consistent, and optimized vSphere environment with minimal effort.

vCenter lifecycle management

Let us get an overview of upgrades, updates, and patches.

Three methods are employed in updating vSphere software versions. All the methods, upgrades, patches, and updates are uniquely utilized in different functions for maintaining and improving the virtual infrastructure, as follows:

- Upgrades are used to introduce major improvements to programs, generally consisting of new features and architectural enhancements. Upgrades are indicated by a major or minor version number, such as updates from vSphere 6.7 to 7.0 or 7.0 to 8.0.
- Updates and patches are incremental improvements, security patches,

and bug fixes. They are marked by an add-on release number, e.g., vSphere 8.0 Update 3, to deliver ongoing stability and performance improvements.

Understanding these differences enables administrators to plan and execute lifecycle management appropriately. Successful vCenter updates and upgrades entail compatibility testing with other VMware solutions through interoperability reports. Interoperability reports would allow administrators to check how VMware solutions to be installed will behave with both the target and current vCenter versions, guaranteeing a smooth upgrade process.

To create these reports, admins must first sign up for the VMware **Customer Experience Improvement Program (CEIP)**. CEIP offers proactive monitoring and management features, such as the vCenter Server Update Planner, which conducts version compatibility analysis automatically.

Administrators can join or leave CEIP by choosing **Learn more and enable CEIP** in the vSphere Client or by going to **Main Menu | Administration | Customer Experience Improvement Program** under **Deployment** and clicking **JOIN** or **LEAVE PROGRAM**.

The following figure illustrate the planning of vCenter updates and upgrades:

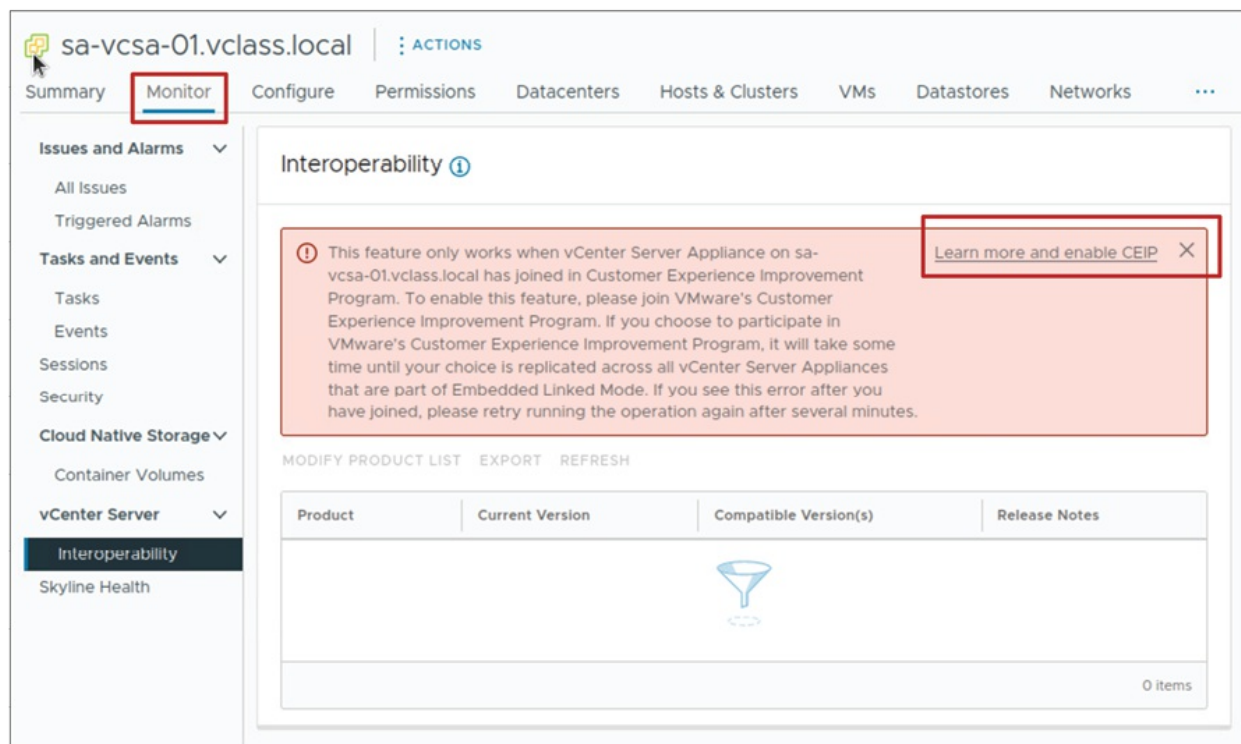


Figure 10.1: Planning for vCenter updates and upgrades

(Source: VMware)

Preparing the interoperability report

The interoperability report within the vSphere Client indicates the overall status of VMware products registered with vCenter and compatible with the target and current vCenter versions. It allows administrators to identify probable upgrade conflict locations and take action on them in advance.

Available through the **Monitor** tab of vCenter, the report presents product names, their installation versions, and the compatible versions once the upgrade has occurred. In these products, if they are not automatically discovered, administrators can manually edit the list of products and re-run the report.

The interoperability report is also exportable as a CSV file, allowing documentation or further analysis and serving as a handy reference for planning upgrades, as shown:

Product	Current Version	Compatible Version(s)	Release Notes
❗ VMware vRealize Automation	8.8.1	8.9.0	Not Available
❗ VMware vRealize Orchestrator	8.8.1	8.9.0	Not Available
✅ VMware vRealize Log Insight	8.6	8.8.0	Not Available
❗ VMware Horizon	2203.0	2206.0.0	Not Available
✅ VMware vSphere Hypervisor (ESXi) (2)	8.0.0	8.0.0	Not Available

Figure 10.2: Generating interoperability report

(Source: VMware)

Upgrading and patching vCenter

To ensure vCenter security and current, administrators use **vCenter Management Interface (VAMI)** to apply updates and patches. The **Update** pane displays a list of available vCenter updates with the release date, version, severity, and type of update (update, upgrade, or patch). In the event of multiple versions, vCenter automatically preselects the most recent recommended version.

vCenter, as a choice, receives updates automatically from the VMware repository but can be set up to receive updates from a custom repository URL, i.e., an internally hosted update server. This option gives organizations control over the distribution of updates and ensures internal security policy compliance.

The following figure illustrates the upgrading and patching vCenter from VAMI:

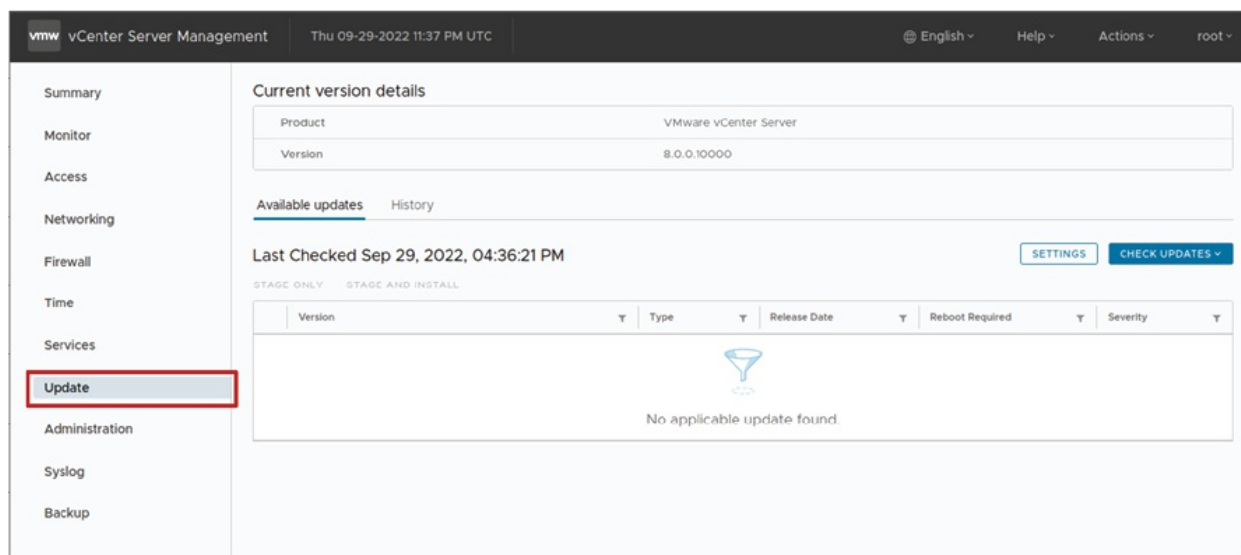


Figure 10.3: Updating and patching vCenter

(Source: VMware)

To upgrade the **vCenter Server Appliance (vCSA)** to a newer version, administrators use the **vCenter installer**, which provides an efficient process for upgrading while preserving existing configurations.

The following figure illustrates the upgrading vCenter from the vCenter installer:

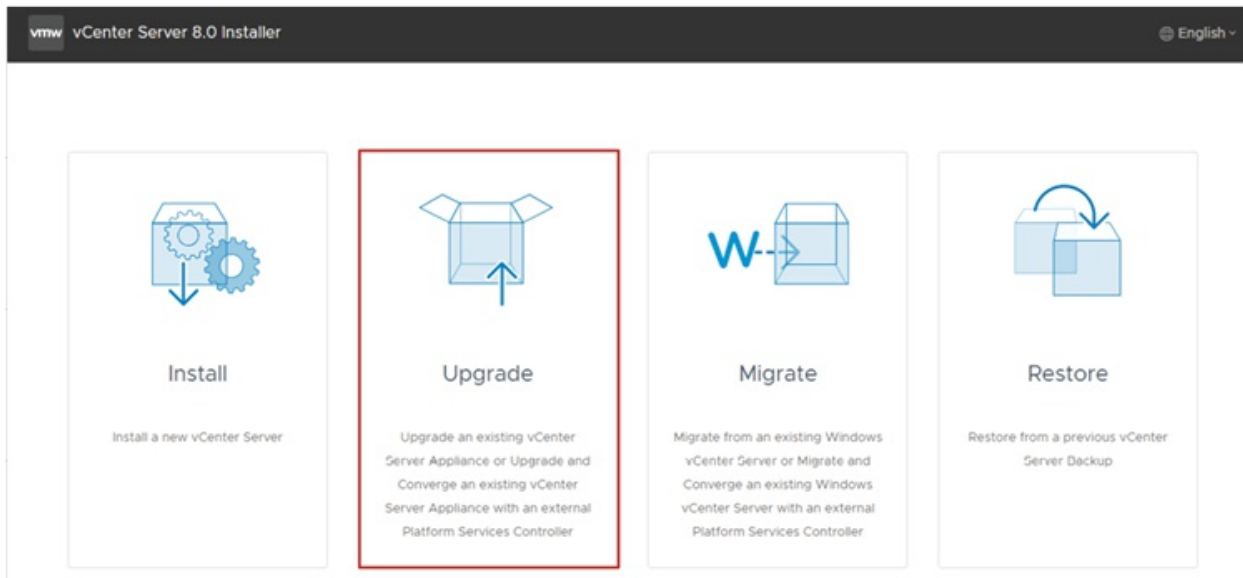


Figure 10.4: Upgrading vCenter appliance

(Source: VMware)

For detailed steps and best practices on upgrading vCenter, refer to <https://docs.vmware.com/en/VMware-vSphere/index.html> on VMware's official documentation.

Overview of upgrading vSphere

There are numerous components to upgrade vSphere, and every component requires a planned approach to upgrade. Having knowledge about the proper sequence of upgrades helps to minimize downtime and maintain the system stable.

The upgrading of vSphere entails the following vital steps:

- Review the vSphere release notes to find out about new features, patches, and potential issues.
- Backup vSphere infrastructure such as vCenter, ESXi hosts, and essential VM settings.
- Ensure that VMware solutions and plug-ins are supported for the vCenter version you are running.
- First, update the **vCenter Server Appliance (vCSA)** because it controls all the ESXi hosts and clusters.
- Upgrade the ESXi hosts across the infrastructure.

- Upgrade **virtual machines (VMs)** manually or using vLCM to obtain the latest hardware capabilities.

The following figure illustrates the vSphere upgrade process:



Figure 10.5: vSphere upgrade process

(Source: VMware)

For step-by-step upgrade instructions, refer to VMware's official documentation at <https://techdocs.broadcom.com/>.

Exploring vSphere lifecycle manager

vLCM based on cluster-wide images is critical for ensuring consistency and minimizing administrative overhead. This allows administrators to consolidate configurations and efficiently apply changes to a group of ESXi hosts, thereby leading to a simpler and more consistent infrastructure.

The core components of vSphere lifecycle manager are:

- VMware Tool and VM Hardware Upgrade Management to maintain virtual machines in top condition.
- Update and patch the ESXi hosts for improved security, stability, and performance.
- The installation and updating of third-party software to introduce new functionality.
- Firmware and hardware driver management of ESXi ensures the hardware compatibility. Standardizing ESXi images across clusters to provide consistent deployments.

To start vSphere Lifecycle Manager, navigate to **Lifecycle Manager** from the top menu of the vSphere Client, as shown in [Figure 10.6](#):

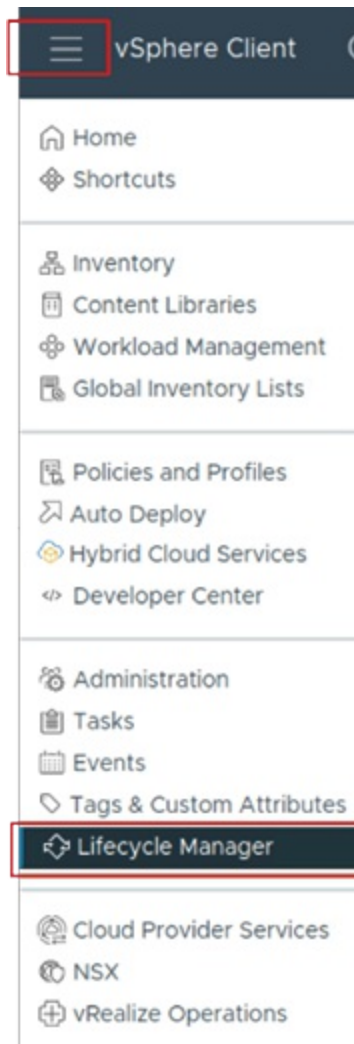


Figure 10.6: Main menu lifecycle manager

(Source: VMware)

Understanding ESXi images

The use of ESXi hosts with images facilitates maintaining the software configuration standardized and uniform across clusters. An ESXi image consists of a number of required components, as shown in [Figure 10.7](#):

- **Firmware and driver's add-on:** Packages that include firmware and driver updates for specific server hardware, and require Hardware Support Manager plug-in installation.
- **Vendor add-ons:** OEM vendor-specific sets of components designed to customize ESXi images for particular hardware platforms.
- **Components:** A logical groupings of one or more **vSphere Installation**

Bundles (VIBs) that deliver specific features or functionalities within the ESXi environment.

- **ESXi base image:** This is the base update package that provides software fixes and enhancements. VMware only produces and distributes these images.

The following figure illustrates the elements of ESXi image:

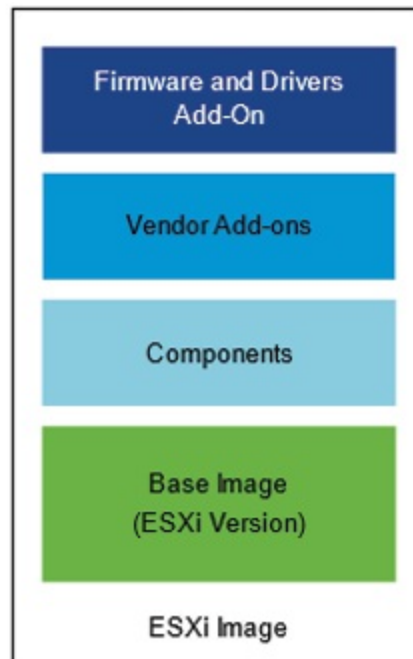


Figure 10.7: Elements of ESXi image

(Source: VMware)

Using a single ESXi image on all hosts in a cluster allows administrators to have consistency, simplify maintenance procedures, and reduce configuration variation. Moreover, components and vendor-specific add-ons allow customization without sacrificing compatibility with the base image.

About image depot

The vLCM image depot is a centralized repository for ESXi images, vendor extensions, and external components. The depot in vCenter gives administrators access to a complete set of software upgrades required for ESXi host lifecycle management.

Administrators can view and manage the following under the **Image Depot**

tab:

- **ESXi base images:** Provides information on the version, build identification number, and contents of the base image.
- **Vendor add-ons:** Provides OEM-specific add-ons, as well as information on the version, release date, and component changes.
- **Third-party components:** Shows component metadata like publisher, version, severity, and included VIBs.

By default, vLCM is configured to gather host updates and metadata automatically during the vCenter deployment process and then repeat on a predetermined period. This process ensures that administrators always have access to the latest patches and updates for maintaining a secure and optimized ESXi environment, as shown:

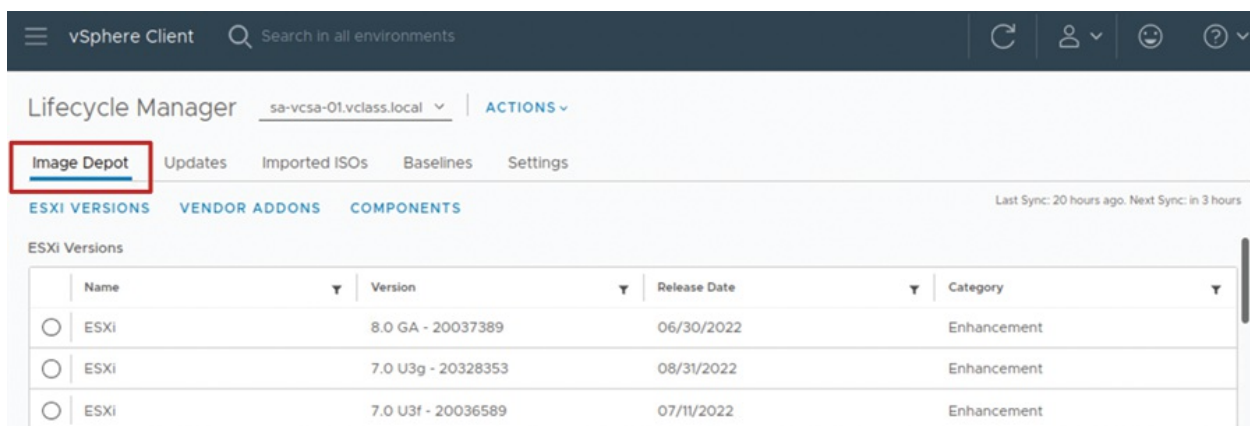


Figure 10.8: About image depot

(Source: VMware)

Importing content to image depot from online sources

vLCM keeps ESXi hosts up to date by automatically downloading updates from defined internet sources at predetermined intervals. Administrators can manually trigger synchronization, when necessary, by navigating to the **Lifecycle Manager** pane in the vSphere Client.

In the lifecycle manager pane, click **ACTIONS** and then **Sync Updates** to update the image depot with the most recent fixes.

An Internet connection is required to access the online depot and the relevant URL that allows vLCM to retrieve updates. This ensures administrators

always have the most recent ESXi images, vendor add-ons, and software components ready to deploy.

The following figure illustrates how to sync updates from online:

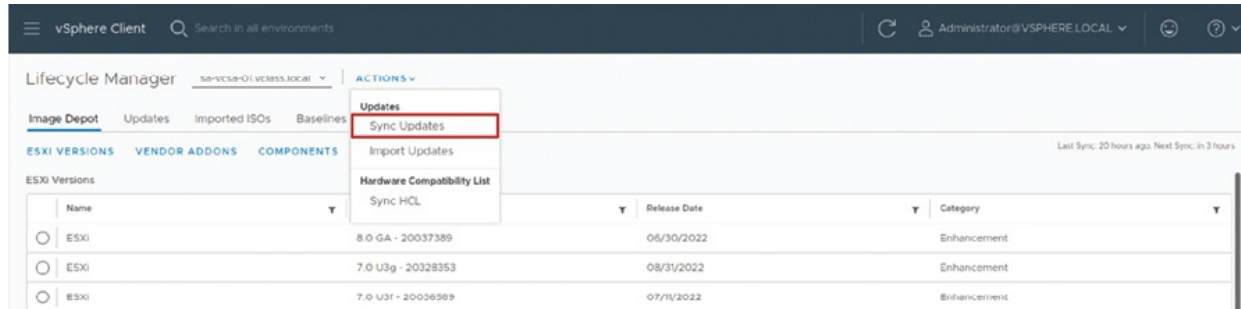


Figure 10.9: Importing online content into the image depot

(Source: VMware)

Specifying the online source for download

Administrators can opt to set vLCM to download updates from VMware's default repository or a custom alternate source. To display or change the download source, go to **Settings | Patch Setup** in the vSphere Client to display or change the download source.

To add a new source, choose **NEW** and enter the URL of a custom repository. While the VMware base depot accommodates the majority of the updates, third-party suppliers can offer more repositories with bespoke pieces, such as CIM modules. Such additional resources, however, are not usually necessary because vendor extensions in the VMware depot usually deliver full OEM customization for ESXi. All depots chosen synchronize updates to the local vLCM image depot, thereby allowing efficient lifecycle management.

The following figure illustrates how to set up download from an online source:

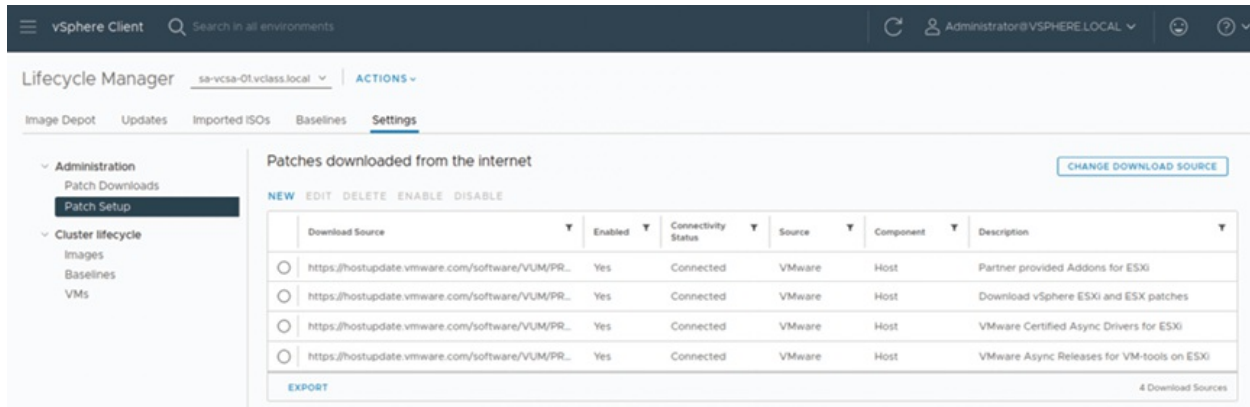


Figure 10.10: Specifying the download source

(Source: VMware)

Importing content to image depot from offline sources

Offline bundles are beneficial where internet access is not available or where updates must be preserved under controlled conditions in secure environments. Offline bundles can consist of an ESXi base image, vendor extensions, third-party components, or asynchronous drivers that are specific to OEM hardware.

Administrators can upload updates to vLCM manually by navigating to **Lifecycle Manager | Actions | Import Updates**. Select a ZIP file containing the offline package or give a URL where the update is available.

The following figure illustrates how to set up download from offline source:

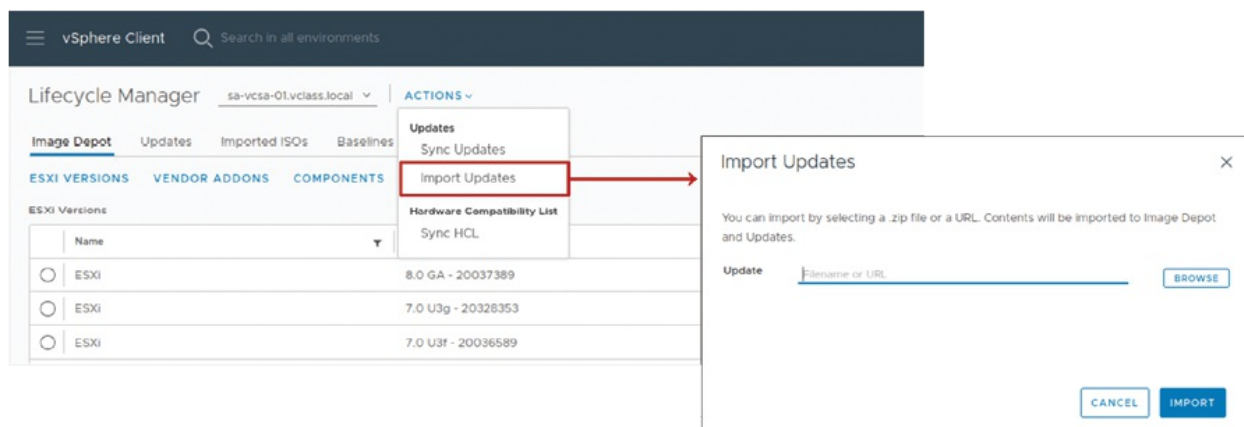


Figure 10.11: Importing offline content into the image depot

(Source: VMware)

ESXi host and cluster lifecycle management

A vSphere Lifecycle Manager cluster enables administrators to manage several ESXi hosts through the utilization of a single image, thereby ensuring consistency throughout the environment. When establishing a new cluster, utilize the following steps to deploy a cluster image:

- **Create a cluster:** Go to the cluster creation wizard in the vSphere Client.
- **Enable image-based management:** Choose **Manage all hosts in the cluster using a single image**.
- **Choose an ESXi image:** Import or upload an existing one from a host system.

The ESXi image can be imported from:

- **Import image from an existing host in the vCenter inventory:** Best to standardize existing hosts in the same vCenter.
- **Import image from a new host:** Supports the import operation from a different vCenter instance or an independent ESXi host, with the option of adding the host to the cluster.

The following figure illustrates creating a cluster with a single image:

New Cluster

1 Basics
2 Image
3 Review

Basics

Name: ICM-Compute-02
Location: ICM-Datacenter
vSphere DRS: ☒
vSphere HA: ☒
vSAN: ☐ Enable vSAN ESA ⓘ

You can change the default settings of these services in the Cluster Quickstart workflow.

☒ Manage all hosts in the cluster with a single image ⓘ

Choose how to set up the cluster's image

- ☒ Compose a new image
- ☐ Import image from an existing host in the vCenter inventory
- ☐ Import image from a new host

☐ Manage configuration at a cluster level ⓘ

CANCEL NEXT

New Cluster

1 Basics
2 Image
3 Review

Image

Compose a new image

Image setup

ESXi Version: 8.0 GA - 20201225
Vendor Addon (optional): None

The cluster image can be further customized later.

Figure 10.12: Creating a cluster with a single image

(Source: VMware)

By employing a cluster image, administrators lock down software versions, drivers, and firmware for all ESXi hosts, minimizing drift and making lifecycle management easier.

The following figure illustrates the image for the cluster:

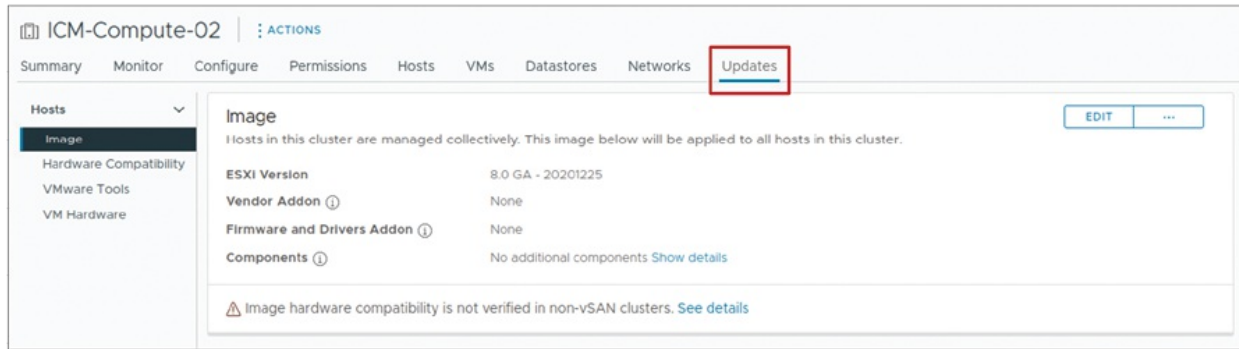


Figure 10.13: Image pane in the Updates tab

(Source: VMware)

Managing clusters with vSphere lifecycle manager

When new software updates are available in the image depot, administrators can use vLCM to maintain ESXi hosts efficiently. The update process follows the following key steps:

1. **Check host compliance:** It verifies that ESXi hosts in the cluster are compliant with the cluster image.
2. **Run a pre-check:** Before applying updates, it performs a remediation pre-check to assess software and hardware compatibility.
3. **Remediate non-compliant hosts:** Apply the necessary updates to bring hosts into compliance state with the cluster image.

By leveraging image-based management, administrators can streamline updates, ensure consistency, and minimize configuration drift across ESXi clusters.

Verification of host compliance with a cluster image

Once a cluster image is established, administrators can run a compliance check that confirms the image against the software and firmware on the ESXi hosts within the cluster, as follows:

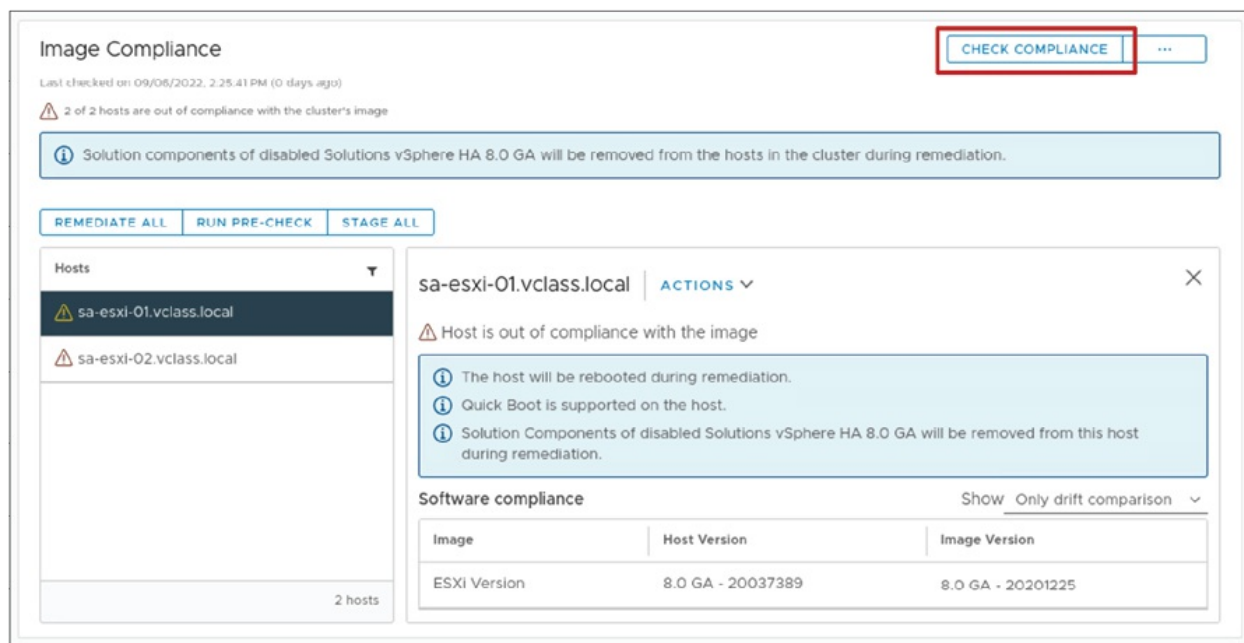


Figure 10.14: Checking host compliance

(Source: VMware)

The following is the host compliance status:

- **Unknown:** The status is unknown prior to the compliance test being conducted.
- **Compliant:** The host's installed software and firmware are identical to the cluster image, with no individual VIBs or conflicting elements.
- **Out of compliance:** The host requires corrective action, which can involve restarting the system or activating maintenance mode.
- **Not compatible:** The host has newer software/firmware than the cluster image or is not compatible with vSphere build specifications.

Administrators can ensure compliance at several vCenter objects, as follows:

- **Host level:** Used on a particular ESXi host.
- **Cluster level:** This is for all ESXi hosts within a cluster.
- **Data center level:** For every host and cluster in a data center.
- **vCenter level:** Refers to the management of all ESXi hosts in multiple clusters and data centers.

Administrators should track this compliance status on a regular basis to maintain consistent and standardized software versions within their vSphere environment.

Performing a remediation pre-check

A remediation pre-check is performed before updating the ESXi hosts. It verifies compliance, identifies conflicts in software and hardware, and checks whether host updates can be conducted. It identifies the necessary maintenance steps for minimum downtime, such as VM evacuations. Running the check prevents update failure and guarantees the success of remediation.

The following figure illustrates the remediation pre-check for image compliance:

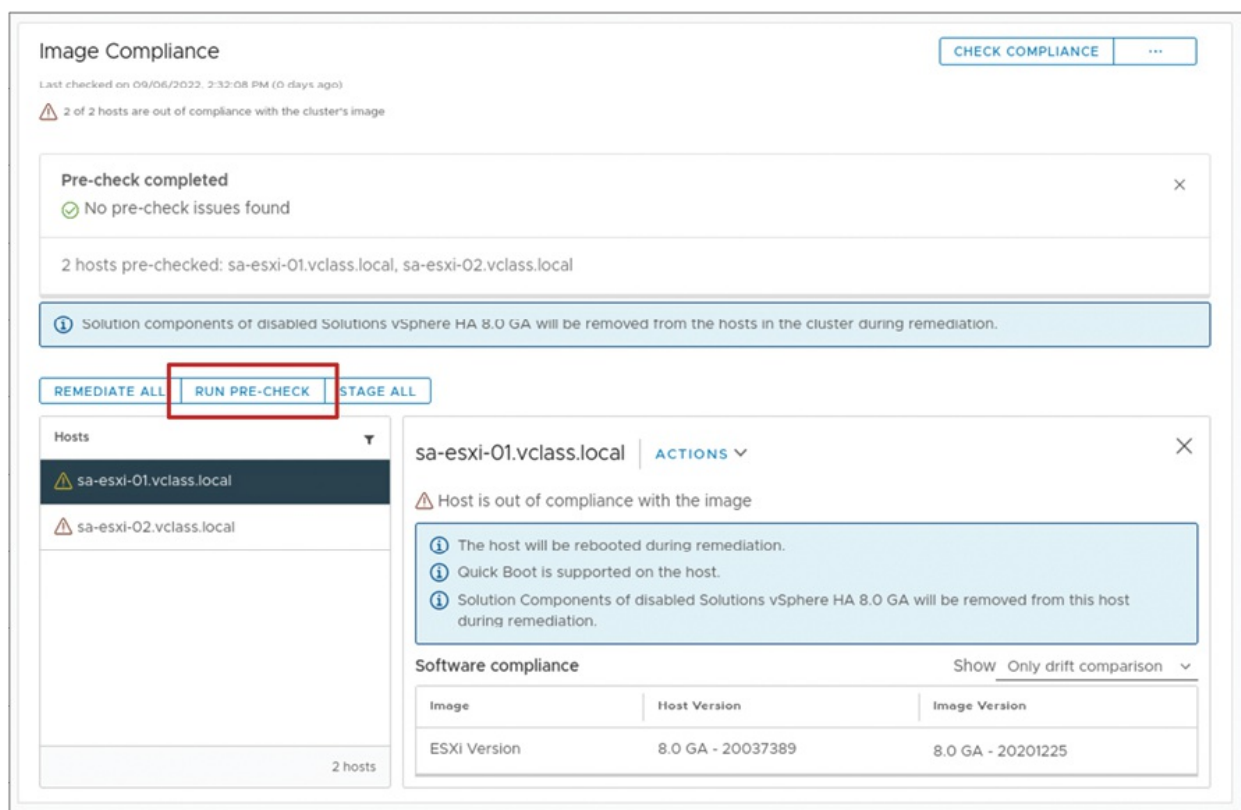


Figure 10.15: Remediation pre-check

(Source: VMware)

Staging the cluster

Organizing the cluster staging is the advance loading of image depot patches and extensions onto ESXi hosts before installation. Staging does not update, but it shortens remediation time by having updates ready on the hosts. A

green icon indicates the successful staging of an image. This method offers the quickest maintenance mode time, ensuring that the hosts are more available to workloads.

The following figure illustrates the staging of a cluster:

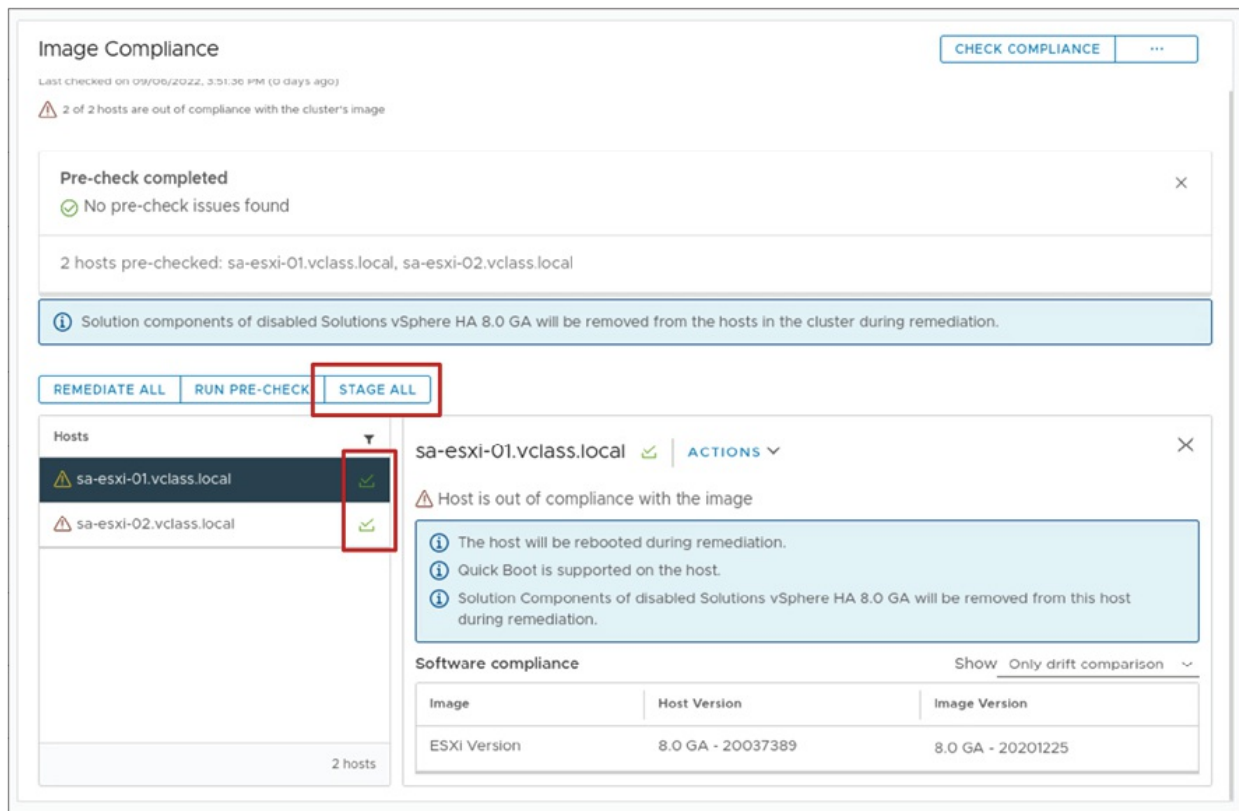


Figure 10.16: Staging the cluster

(Source: VMware)

Remediation of a cluster against an image

Remediation ensures ESXi hosts conform to the defined cluster image. Administrators can remediate all clusters and individual hosts or perform a pre-check but not apply any updates. If the pre-check is successful, vSphere Lifecycle Manager applies the image.

vSphere Lifecycle Manager checks during remediation whether hosts can enter or leave maintenance mode and be restarted.

The procedure is applied to:

- ESXi image version

- Vendor add-ons (Optional)
- Firmware and driver enhancements (Optional)
- User-defined elements (Optional)

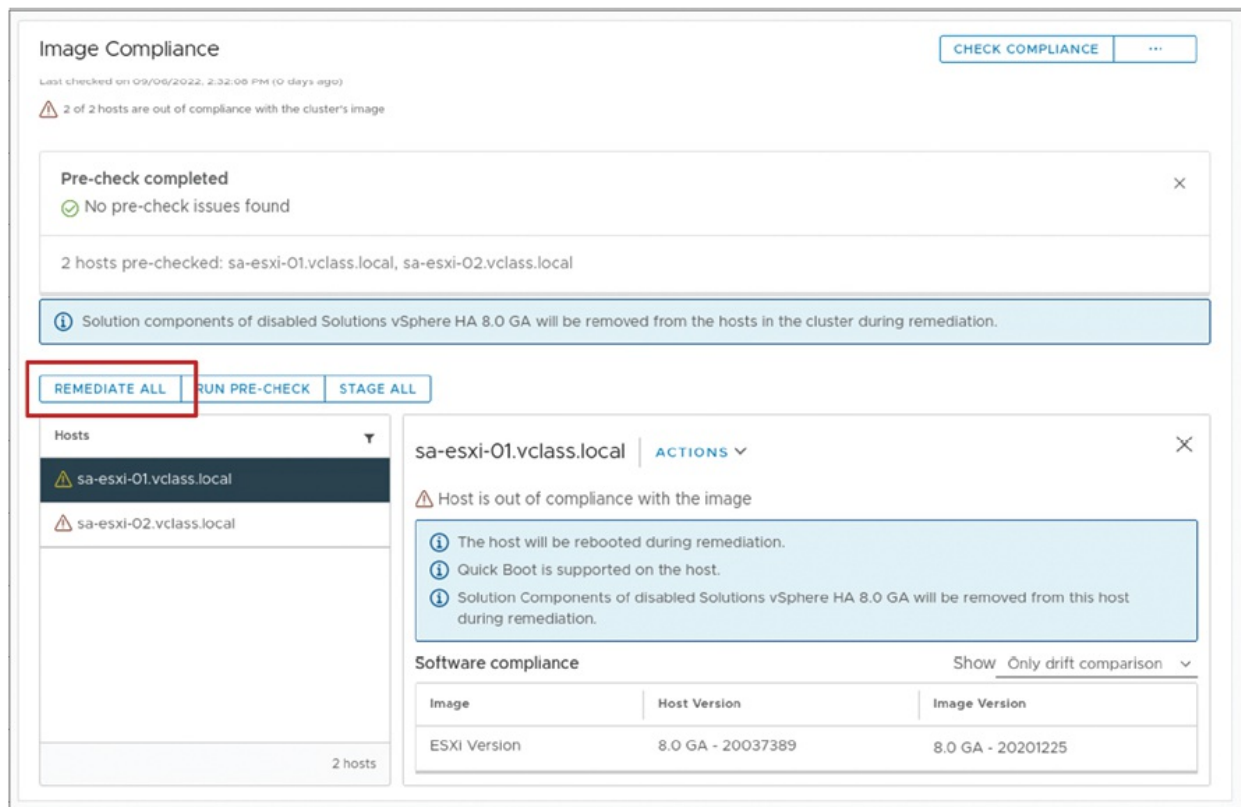


Figure 10.17: Remediating a cluster

(Source: VMware)

This methodical approach guarantees uniformity and reduces interruptions in a vSphere infrastructure.

Reviewing remediation impact

Before applying updates, the vSphere Lifecycle Manager conducts a remediation pre-check to assess the impact on ESXi hosts. If successful, remediation proceeds with applying the cluster image.

The **Review Remediation Impact** dialog box provides critical insights, including:

- **Impact summary:** Overview of changes affecting the hosts.
- **Applicable remediation settings:** Configurations that influence the

remediation process.

- **End User License Agreement (EULA):** Licensing terms for updates.
- **Impact on specific hosts:** Details on reboot requirements, maintenance mode, and configuration changes.

The following image illustrates the remediation impact:

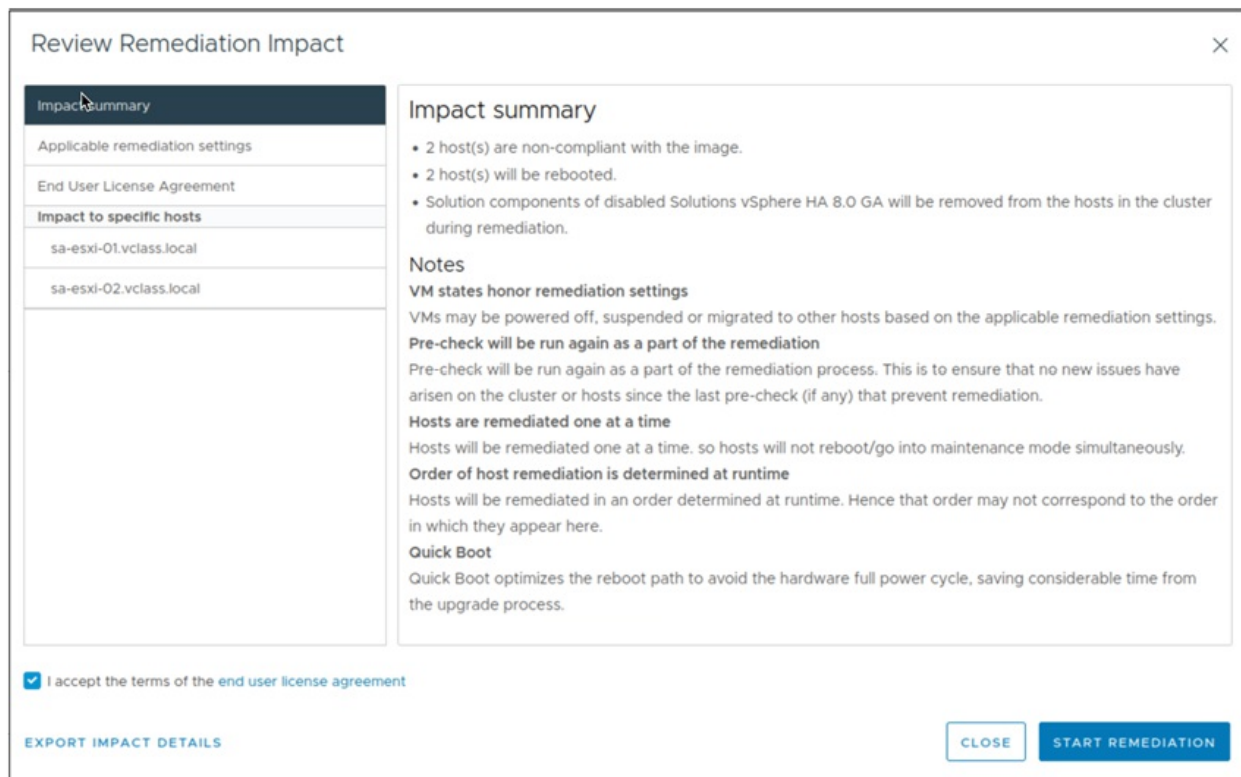


Figure 10.18: Reviewing remediation impact

(Source: VMware)

Parallel remediation

vSphere Lifecycle Manager also features parallel remediation to accelerate the upgrade of a cluster by remediating ESXi hosts in parallel instead of one at a time. The administrators can configure the maximum number of concurrent remediations to balance speed against availability.

Hosts need to be in maintenance mode to allow parallel remediation to continue. All non-maintenance-mode hosts are skipped but can be manually remediated afterward. This feature saves considerable downtime and improves efficiency while deploying cluster-wide updates.

The following figure illustrates the parallel remediation page:

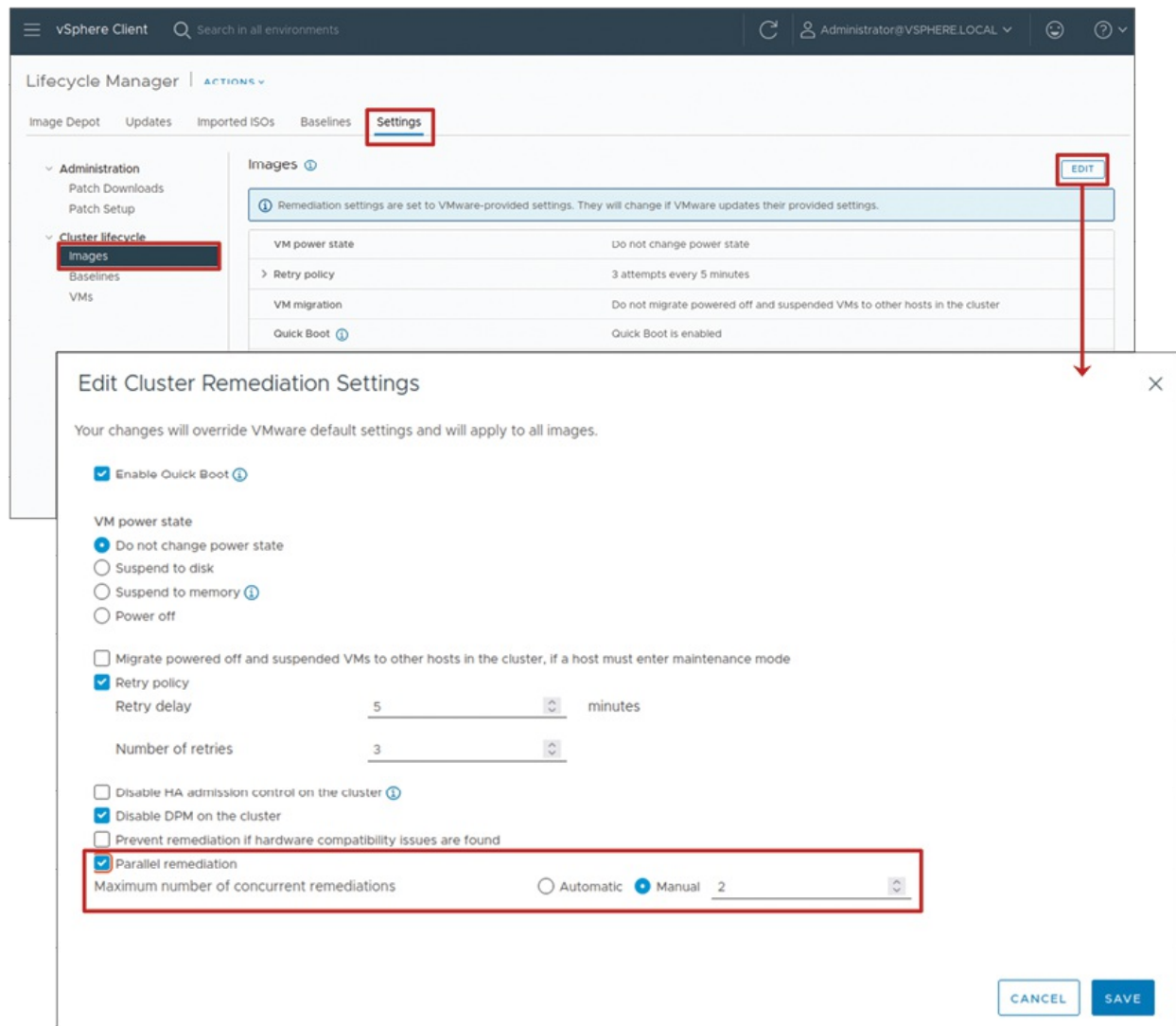


Figure 10.19: Parallel remediation

(Source: VMware)

When remediating a vSphere DRS cluster, vLCM will automatically collaborate with vSphere DRS to automate remediation, as follows:

- **Host evacuation:** Before remediation, vSphere DRS evacuates VMs from the ESXi host to reduce the impact.
- **Maintenance mode tests:** vLCM tests the capability of a host to enter maintenance mode and identifies any configuration issues that would prevent this from occurring.

- **Automated recovery:** vLCM is patched and is restarted, exiting maintenance mode, and vSphere DRS redistributes workloads to the host. This integration provides an efficient and smooth remediation cycle with reduced manual intervention and cluster stability.

Recommended images in vSphere lifecycle manager

Administrators can successfully maintain ESXi hosts updated by leveraging vLCM-suggested images for image-managed clusters to ensure compatibility and optimal performance, thus maintaining stability in the vSphere environment.

- **Compatibility tests:** vLCM ensures that the recommended image is free from any missing dependencies or malicious items.
- **Source validation:** The procedure verifies the vLCM repository for available software and firmware enhancements, including hardware support manager support.
- **Enhanced selection:** More hardware compatibility tests are executed for vSAN clusters to make sure the newest and greatest image is recommended.

By following these guidelines, administrators can successfully update ESXi hosts and maintain stability in the vSphere environment.

The following figure illustrates the option for image recommendation:

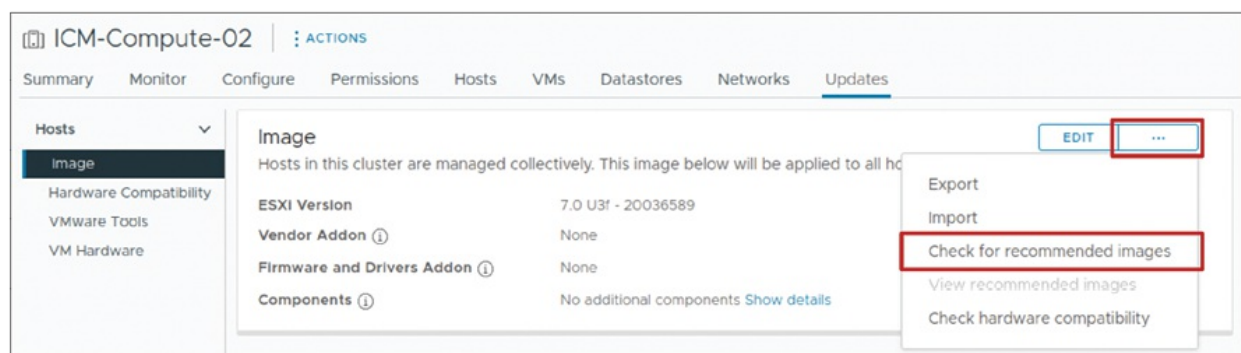


Figure 10.20: Recommended image

(Source: VMware)

Viewing recommended images in vSphere lifecycle manager

To view recommended images, navigate to the **Recommendations** tab or select View recommended images from the drop-down menu as shown in *Figure 10.21*:

- **Major release series:** Recommendations are based on the same major ESXi version currently used in the cluster.
- **Compatibility-driven suggestions:** vSphere lifecycle manager ensures that updates do not introduce hardware compatibility issues or regressions.
- **Example:** If a cluster is running **ESXi 7.0 U3f** and **7.0 U3g** and **8.0** are available, **7.0 U3g** is recommended, as 8.0 belongs to a new major release series.

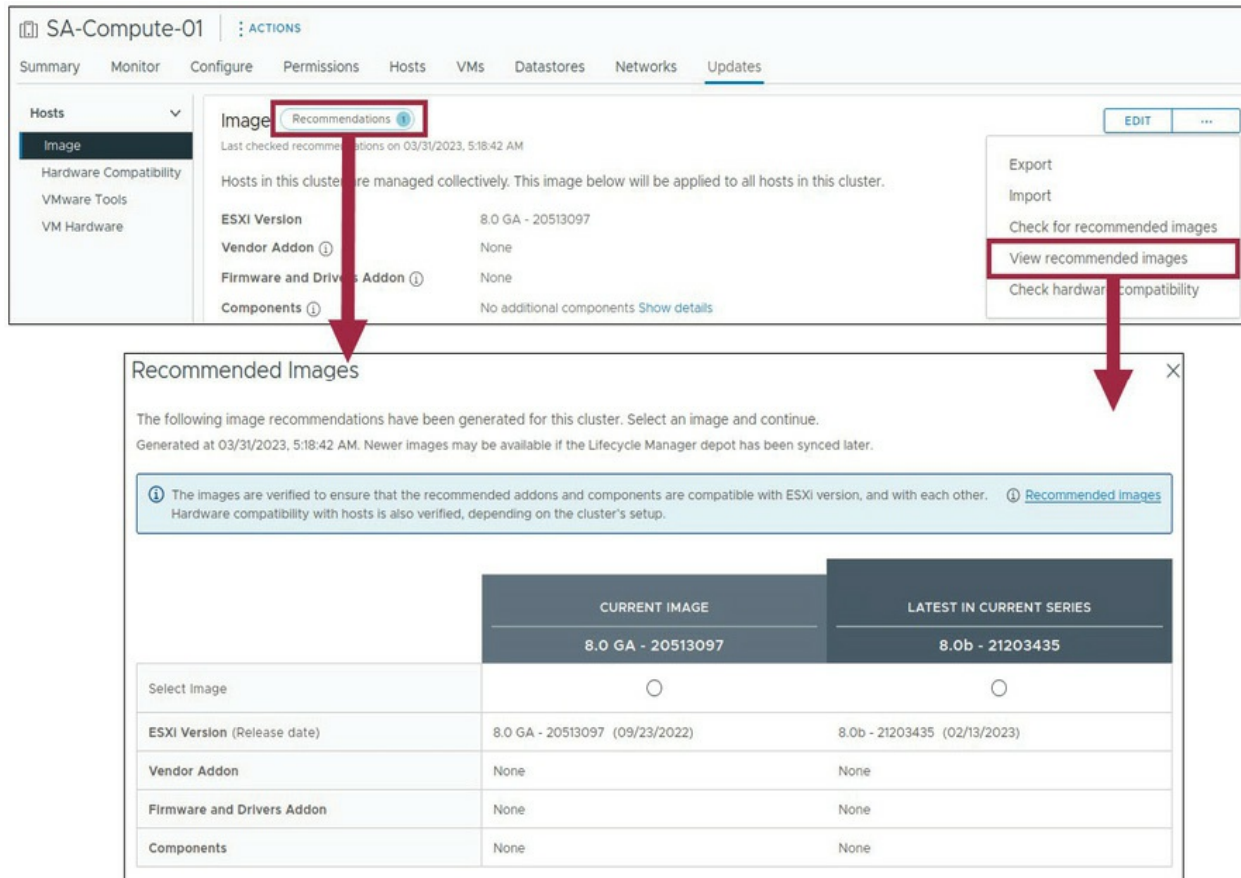


Figure 10.21: Viewing recommended image

(Source: VMware)

The following are the types of recommended images:

- **Current image:** The image currently running on the cluster.
- **Latest in the current series:** It is recommended that a newer version within the same release series be used.
- **No recommendation:** No further recommendations appear if the cluster is already running the latest available version.

This approach ensures that updates are applied safely and efficiently without unnecessary changes that could impact stability.

Customizing cluster images using vSphere lifecycle manager

Administrators can alter an image once it has been used to operate a cluster to meet changing infrastructure requirements:

- **Change the ESXi base image:** Upgrade to a newer ESXi version or rollback to an older version.
- **Modify components:** Add, remove, or update vendor add-ons, firmware, drivers, and other components.

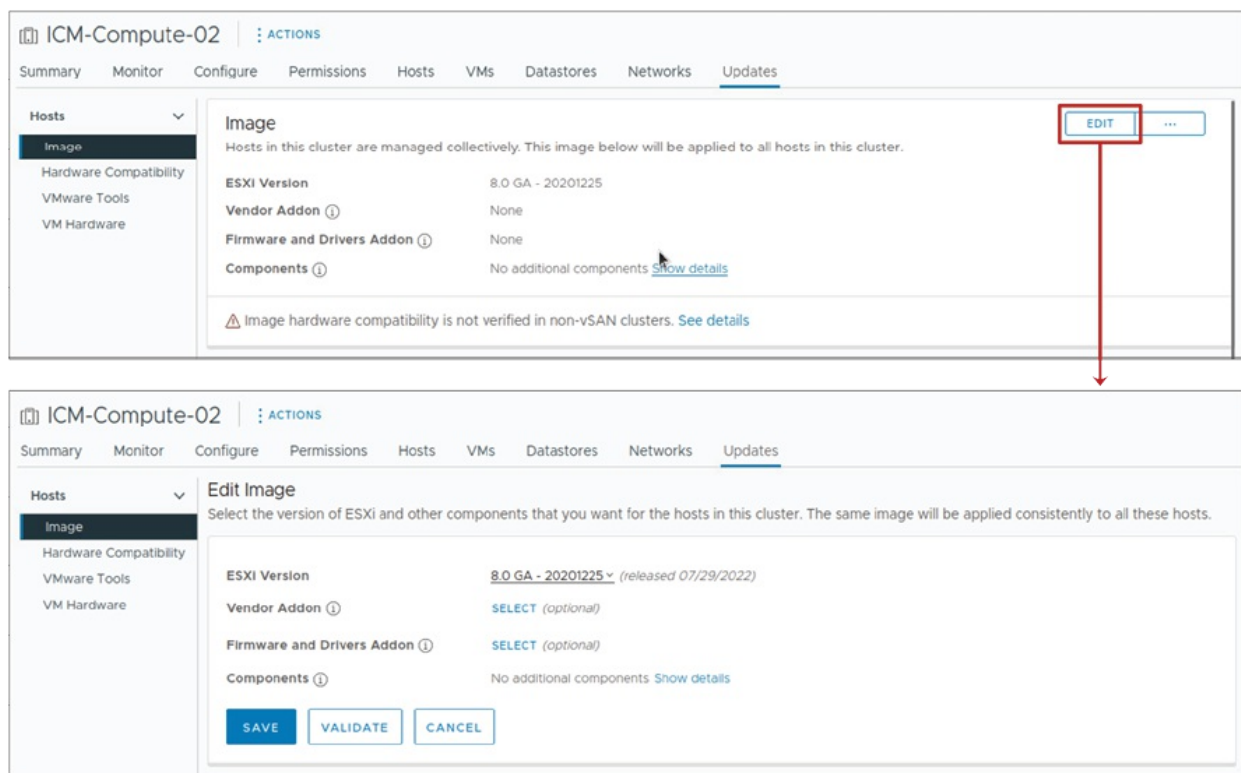


Figure 10.22: Customizing cluster images

(Source: VMware)

Before making changes, vSphere lifecycle manager checks that the modified image meets compatibility and dependency requirements:

- **Completeness check:** Ensures all necessary components are included.
- **Dependency validation:** Ensures that no required components are missing.
- **Conflict detection:** Prevents mismatched or incompatible components from being used.

Hardware compatibility is important when operating ESXi hosts and clusters in vSphere. vSphere lifecycle manager validates hardware against VMware's released compatibility lists to avoid unsupported configurations.

The following is a VMware compatibility guide:

- Lists vSphere-certified server models and I/O devices by vSphere version.
- Helps verify hardware support before upgrading ESXi hosts.
- Accessible via the VMware compatibility guide.

vSAN hardware compatibility list (HCL) is exclusive to vSAN clusters to provide compatibility for:

- Input/output and networking controllers
- Supported disk drives
- Approved firmware versions

Hardware compatibility checks allow administrators to prevent upgrade issues and have a solid, supported vSphere platform.

The following image illustrates the hardware compatibility:

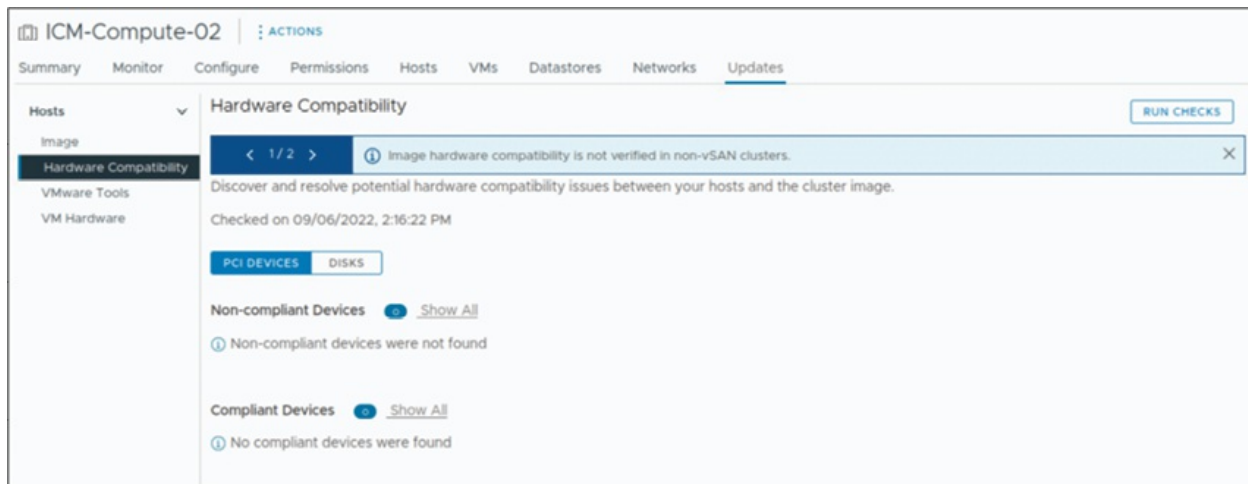


Figure 10.23: Hardware compatibility

(Source: VMware)

Managing vSphere configuration profiles

Introduced in vSphere 8 Update 1, vSphere Configuration Profiles allow administrators to define and enforce host configurations at the cluster level centrally.

The following are the critical competencies:

- A desired cluster configuration should be specified in a JSON file.
- Confirm host compliance with the provided configuration.
- Remediate automatically any non-compliant hosts to match the desired state.

The following are the requirements and considerations:

- Needs single-image cluster management.
- All ESXi hosts should be vSphere 8.0 or higher.
- VMware NSX environments are not supported. vSphere 8 Update 1 introduced the vSphere Distributed Switch configuration feature.

This facility automates host management, providing consistency and minimizing the efforts required for manual configuration across clusters.

The following figure illustrates the vSphere configuration profiles at the cluster level:

New Cluster

Basics

1 Basics
2 Image
3 Configuration
4 Review

Name: New Cluster

Location: SA-Datacenter

vSphere DRS: ☐

vSphere HA: ☐

vSAN: ☐ Enable vSAN ESA ⓘ

☒ Manage all hosts in the cluster with a single image ⓘ

Choose how to set up the cluster's image

☒ Compose a new image

☐ Import image from an existing host in the vCenter inventory

☐ Import image from a new host

☒ Manage configuration at a cluster level ⓘ

CANCEL NEXT

Figure 10.24: Managing vSphere configuration profiles

(Source: VMware)

Configuration documentation

The configuration file vSphere Configuration Profiles uses is a JSON file consisting of the desired state of all hosts in a cluster. A schema controls it, and hence, it is readable and editable in any JSON editor, as shown in [Figure 10.25](#):

Draft the configuration document as follows:

- **Pull from a reference host:** Create a JSON file by pulling the configuration information of a properly configured ESXi host.
- **Build from scratch:** Manually specify the target cluster-wide configuration in JSON format.


```

{
  "profile": {
    "esx": {
      "network": {
        "firewall": {
          "enabled" : true
        }
      }
    }
  },
  "host-override" : {
    "4201db82-b62e-1a56-462c-2648158ac1f2" : {
      "esx": {
        "network": {
          "firewall": {
            "enabled" : false
          }
        }
      }
    }
  },
  "host-specific" : {
    "4201db82-b62e-1a56-462c-2648158ac1f2" : {
      "esx": {
        "advanced_options": {
          "misc": {
            "host_name" : "host1.vmware.com"
          }
        }
      }
    }
  },
  "metadata": {
    "reference_host": {
      "uuid": "42011225-0b2c-b321-f586-b0e32581ba3e",
      "build": "BETAbuild-58249482",
      "patch": "0",
      "update": "0",
      "version": "8.0.0"
    }
  }
}

```

Figure 10.25: Configuration document

(Source: VMware)

This paper acts as a host configuration template to provide uniformity and ease of compliance management across the cluster.

Using vSphere configuration profiles

vSphere configuration profiles automate cluster-wide configuration management. The entire process involves:

- **Reaching the desired state:** Go to the **Configuration** tab under **Desired States**.
- **Compliance verification:** Enumerate any hosts that fail to comply with the provided configuration requirements.
- **Do a pre-check:** Perform a remediation pre-check to search for possible problems.
- **Assessing the impacts:** Test changes before adopting them.
- **Remediate the cluster:** Push the desired configuration to get all the hosts in compliance.

This approach ensures the even application of configuration to the cluster.

vSphere lifecycle manager for standalone hosts

From vSphere 8 onwards, vLCM adds support for vCenter-managed standalone ESXi hosts, which includes the following:

- **vSphere client support:** With vSphere 8 Update 1, administrators are now able to utilize the vSphere Client to:
 - Designate a preferred depiction of single hosts.
 - Remediate hosts against the said image.
 - Check compliance for standalone hosts.
- **API-based management:** The vLCM for a single host uses vSphere APIs to enable lifecycle operations.
- **Custom image depots:** Standalone hosts can make use of custom image depots in vSphere 8 Update 1. It is convenient for the following:
 - Edge environments with remote hosts
 - Hosts with limited connectivity to vCenter
 - Reducing latency issues during remediation

The following figure illustrates the vLCM for standalone hosts:

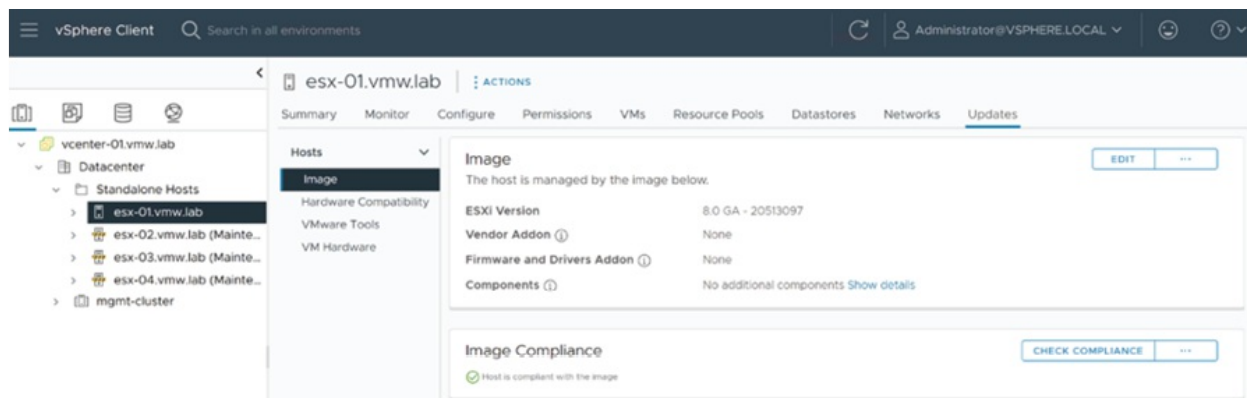


Figure 10.26: vSphere lifecycle manager for standalone hosts

(Source: VMware)

This enhancement ensures better update management and flexibility for standalone ESXi deployments.

Managing VMware tools and virtual hardware

VMware Tools and virtual hardware are crucial for optimizing virtual machine performance and functionality within a VMware environment, providing features like improved graphics, mouse, and network performance, as well as enabling features like time synchronization and guest OS management, as follows:

- **Keeping VMware Tools up to date:** VMware Tools is updated with ESXi releases to add bug fixes, security patches, improved driver support, and performance enhancements for virtual devices. Keeping VMware Tools up to date ensures maximum compatibility, stability, and security in the virtualized environment. As part of regular data center maintenance, keeping VMware Tools up to date improves overall system reliability and efficiency.
- **Upgrading VMware Tools:** To keep VMware Tools current, go to a host or cluster's Updates page and choose VMware Tools. From here, the administrator can see the status of VMware Tools and updates needed on virtual machines, as shown in [Figure 10.27](#):

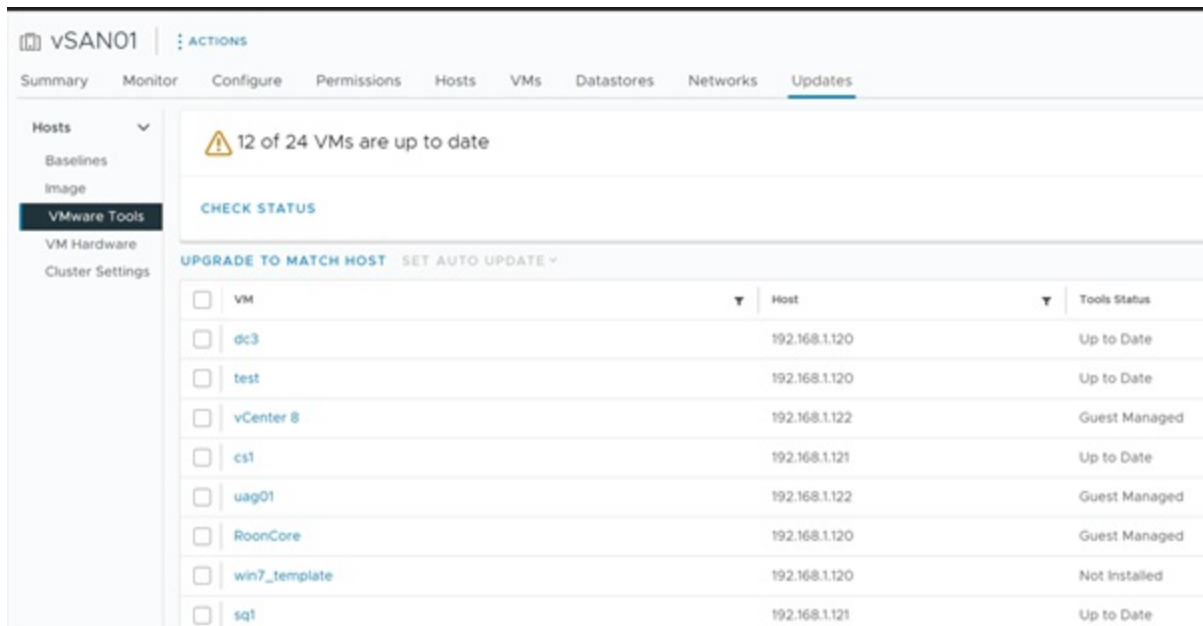


Figure 10.27: Upgrading VMware Tools (1)

(Source: VMware)

A VM can have one of the following VMware Tools status:

- **Current state:** The installed version is supported and compliant with the ESXi host.
- **Upgrade available:** A newer version is available on the ESXi host, and an upgrade is required.
- **Version incompatible:** The existing version is outdated, has known issues, or is too advanced for adequate compatibility. A priority update may be required.
- **Not installed:** VMware Tools is not installed on the VM and must be installed for best performance.
- **Guest managed:** VMware Tools are externally managed (e.g., OpenVMTools in Linux). Updates have to be addressed via native OS package management.
- **Unknown:** We are unable to confirm the VM status. Make sure the VM is on and press CHECK STATUS to get the current data. Monitoring VMware Tools status on a regular basis ensures performance, security, and compatibility within your vSphere environment.

After reviewing the VMware Tools status, proceed with the upgrade by

choosing the VMs that require an update and clicking *UPGRADE TO MATCH HOST*, as follows:

- Select the VMs that require an update.
- Schedule the upgrade to coincide with maintenance periods.
- Plan for any necessary reboots following the update.
- Choose rollback options to reverse changes if problems develop.

The following figure illustrates the VMware Tools upgrade:

Upgrade VMware Tools to Match Host | 192.168.1.120

Upgrading a virtual machine might require that it is powered on, powered off, or rebooted multiple times. Only 5 virtual machines can be updated per host at one time.

1 VM will upgrade

VMware does not recommend upgrading VMware Tools on virtual appliances (VAs) from here. Consider de-selecting any VAs in the following table.

<input type="checkbox"/>	VM	Tools Status	Auto Update
<input checked="" type="checkbox"/>	DEMProfiler	Upgrade Available	Off
<input type="checkbox"/>	Server2022 (template)	Unknown	Off

1 2 VMs

Scheduling Options: All VMs will upgrade immediately

Scheduled Task Name: 192.168.1.120 - Scheduled Upgrade

Scheduled Task Description:

Powered ON VMs: Immediately

Powered OFF VMs: Immediately

Suspended VMs: Immediately

Rollback Options: VM snapshots are enabled

Rollback will take a snapshot of the VMs before upgrading.

☒ Take snapshot of VMs

Snapshots reduce performance of VMs. Delete the snapshots as soon as you have validated the upgrade.

☒ Do not delete snapshots

☐ Keep snapshots for 1 hours

Snapshot Name:

Snapshot Description:

☐ Include the virtual machine memory in the snapshot

CANCEL UPGRADE TO MATCH HOST

Figure 10.28: Upgrading VMware Tools (2)

(Source: VMware)

To avoid unexpected downtime, schedule VMware Tools upgrades during scheduled maintenance.

Keeping VM hardware up to date

VMware updates VM hardware versions with each ESXi update, including increased configuration maximums and support for new virtual hardware like vGPU, vNVMe, vSGX, and vTPM.

When upgrading VMware Tools, it is important to prioritize compatibility and must be upgraded before VM hardware. Upgrade VM hardware only when absolutely necessary, such as when new features or optimizations are needed.

Regularly reviewing hardware requirements helps maintain maximum performance while guaranteeing compliance with the most recent ESXi upgrades.

To keep VMware Hardware current, go to a host or cluster's Updates page and choose VMware Hardware. From here, the administrator can see the status of VMware Hardware and updates needed on virtual machines, as shown in [Figure 10.29](#).

The following steps outline the methods for upgrading the VMware hardware version:

1. **Verify VM hardware status:** Go to the VM Hardware page to see the present hardware version of the virtual machines. The status will be displayed as follows:
 - a. **Upgrade available:** The VM hardware version is outdated and can be upgraded to be in line with the ESXi host.
 - b. **Up to date:** The VM hardware version is currently the latest version supported on the ESXi host.

The following figure illustrates the VMware hardware status:

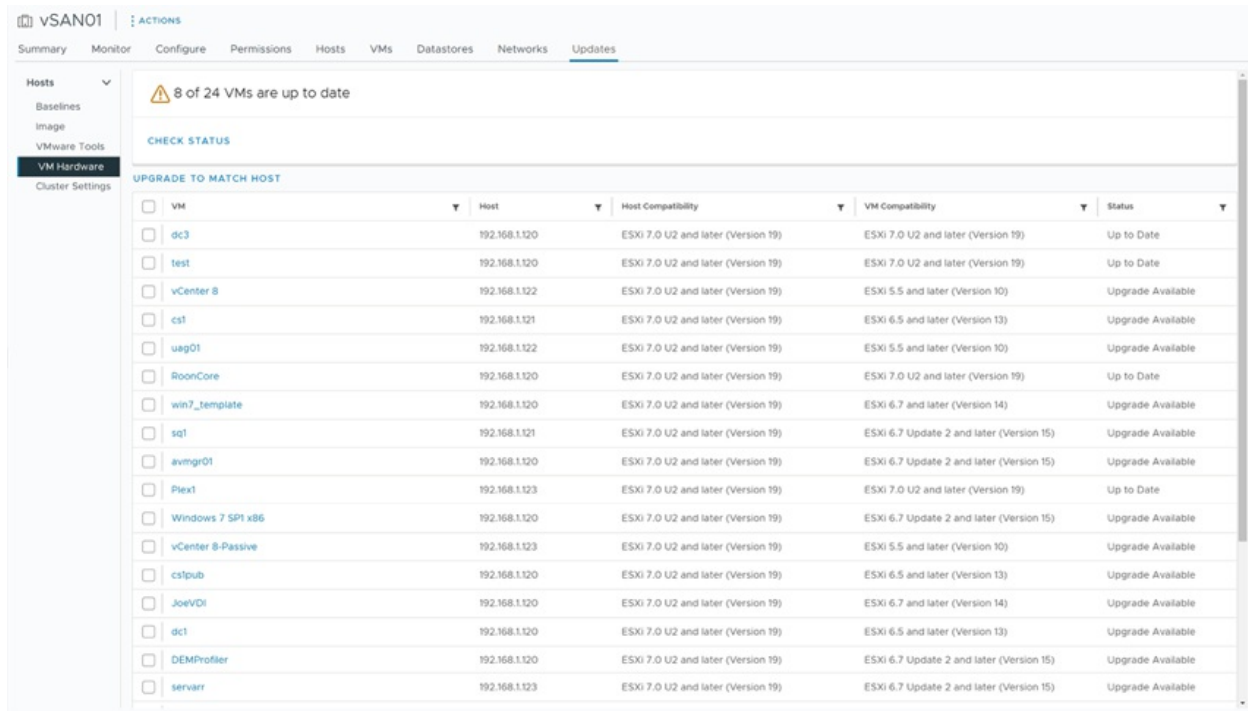


Figure 10.29: Upgrading VMware hardware (1)

(Source: VMware)

2. **Select the VMs:** Select the VMs that require a hardware upgrade.
3. **Upgrade the VM hardware version:** Perform the Upgrade Process by clicking **UPGRADE TO MATCH HOST**.

Upgrading a virtual machine might require that it is powered on, powered off, or rebooted multiple times. Only 5 virtual machines can be updated per host at one time.

2 VMs will upgrade

VMware does not recommend upgrading hardware on virtual appliances (VAs) from here. Consider de-selecting any VAs in the following table.

<input checked="" type="checkbox"/>	VM	VM Compatibility	Status
<input checked="" type="checkbox"/>	cs1	ESXi 6.5 and later (Version 13)	Upgrade Available
<input checked="" type="checkbox"/>	sq1	ESXi 6.7 Update 2 and later (Version 15)	Upgrade Available

☒ 2 VMs

Scheduling Options: All VMs will upgrade immediately

Scheduled Task Name

192.168.1.121 - Scheduled Upgrade

Scheduled Task Description

Powered ON VMs

Immediately

Powered OFF VMs

Immediately

Suspended VMs

Immediately

Rollback Options: VM snapshots are enabled

Rollback will take a snapshot of the VMs before upgrading.

☒ Take snapshot of VMs

Snapshots reduce performance of VMs. Delete the snapshots as soon as you have validated the upgrade.

☒ Do not delete snapshots

☐ Keep snapshots for 1 hours

Snapshot Name

Snapshot Description

☐ Include the virtual machine memory in the snapshot

CANCEL

UPGRADE TO MATCH HOST

Figure 10.30: Upgrading VMware hardware (2)

(Source: VMware)

Upgrade within a maintenance window to experience the minimum possible disruptions. Set up rollback options to roll back if there are compatibility problems. Maintaining VMware Tools up to date prior to upgrading VM hardware prevents compatibility problems and surprise reboots.

Note: VM hardware upgrades often require a reboot. However, if no major changes are

introduced, some upgrades may not need downtime.

Conclusion

Lifecycle management is essential to maintaining a stable, secure, and high-performing vSphere environment. In this chapter, we discussed the essential role of vLCM in automating and simplifying the upgrade and patching of vCenter, ESXi hosts, VMware Tools, and virtual hardware. By using interoperability reports and the vCenter Update Planner, administrators can pre-test for compatibility before upgrades, reducing risks and making the transition seamless.

Readers also learned about the importance of lifecycle management between clusters using images to maintain consistency among ESXi hosts with minimal administrative burden. The image depot and its ability to import updates from online and offline repositories provide flexibility in software update management, and compliance checks guarantee hosts are in sync with the defined cluster image. This chapter also emphasized the importance of upgrading VMware Tools and VM hardware to guarantee that virtual machines benefit from enhanced security, performance improvements, and compatibility with new releases of ESXi.

With the knowledge of these lifecycle management methods, readers are now ready to have a well-tuned, efficient, and secure vSphere infrastructure with minimal or no manual effort.

After a strong lifecycle management foundation, [Chapter 11, Virtualization Best Practices](#) seeks to integrate needed knowledge throughout the book, presenting proven methods, expert advice, and real-world optimizations for streamlined virtualized environment management. Networking, storage, resource allocation, security controls, monitoring practices, and high availability best practices will be learned by readers, thus ensuring their vSphere infrastructure runs at peak efficiency.

Points to remember

- Interoperability reports allow administrators to confirm that the vCenter

upgrade is VMware- and third-party solution-compatible to avoid potential problems.

- vLCM provides a single approach to managing patches, updates, and upgrades for clusters, ESXi hosts, drivers, firmware, VMware Tools, and virtual machine hardware.
- vLCM supports both online and offline depots for importing patches, updates, and add-ons, which is convenient in terms of update management.
- VMware Tools updates need to be carried out prior to VM hardware upgrades for compatibility purposes and to prevent surprise restarts.
- Upgrades to VM hardware should only be done, if necessary, to leverage new functionality since they can require VM downtime.
- Rollback mechanisms must always be considered before any lifecycle updates are rolled out to mitigate risks and prevent disruptions.
- Maintenance and updates enhance security, stability, and performance throughout the vSphere infrastructure.

Exercises

1. What is the sole function of vLCM?
2. In what ways does vCenter Update Planner help plan updates and upgrades, and why should the interoperability report be created prior to upgrading?
3. What are the major parts of an ESXi image in vSphere Lifecycle?
4. Why is VMware Tools upgraded prior to upgrading VM hardware?

Lab exercises

1. **Updating ESXi hosts using the vSphere lifecycle manager:** The objective is to learn how to manage and apply updates to ESXi hosts using vLCM.
 - a. **Create a cluster and select an image:**
 - i. In the vSphere Client, navigate to Hosts and Clusters.

- ii. Create a new cluster and enable vSphere lifecycle manager.
 - iii. Select and configure a cluster image that includes the ESXi base image, vendor add-ons, and components.
 - b. **Add ESXi hosts to the cluster:**
 - i. Add existing ESXi hosts to the newly created cluster.
 - ii. Ensure that all hosts inherit the selected cluster image.
 - c. **Check for host compliance:** From the vLCM interface, run a compliance check to identify non-compliant hosts.
 - d. **Remediate noncompliant hosts:**
 - i. Apply updates to bring non-compliant hosts in line with the cluster image.
 - ii. Verify successful remediation by rechecking compliance.
2. **Upgrading VMware tools and VM hardware using vLCM:** The objective is to use vLCM to upgrade VMware Tools and virtual hardware for virtual machines.
- a. **Check VMware tools status:**
 - i. Navigate to Lifecycle Manager | Updates.
 - ii. Review the VMware Tools compliance report for VMs in your cluster.
 - b. **Upgrade VMware tools for noncompliant VMs:**
 - i. Select VMs with outdated VMware Tools.
 - ii. Click Upgrade to Match Host and confirm the update process.
 - iii. Verify successful upgrades by checking the VMware Tools version.
 - c. **Upgrade VM hardware:**
 - i. In the Updates tab, select VM Hardware.
 - ii. Identify VMs with outdated hardware versions.
 - iii. Schedule the upgrade, ensuring VMware Tools is up to date first.
 - iv. Apply the upgrade and verify successful completion.
3. **Importing and managing ESXi images in vSphere lifecycle manager:** The objective is to learn how to import ESXi images and configure vLCM to use both online and offline sources.

a. Import an ESXi image from an online source:

- i. Navigate to Lifecycle Manager | Image Depot.
- ii. Select Sync Updates to fetch the latest ESXi images from VMware's online depot.
- iii. Verify the availability of downloaded images in the depot.

b. Import an ESXi image from an offline bundle:

- i. Select Import Updates and upload an ESXi offline bundle (.zip) from local storage.
- ii. Confirm that the image appears in the Image Depot.

c. Assign an image to a cluster:

- i. Create a new cluster and assign an ESXi image from the depot.
- ii. Validate cluster compliance against the selected image.

Join our Discord space

Join our Discord workspace for latest updates, offers, tech happenings around the world, new releases, and sessions with the authors:

<https://discord.bpbonline.com>



CHAPTER 11

Virtualization Best Practices

Introduction

In this chapter, we provide a selected set of implementation strategies and insights from industry experts aimed at improving the efficiency, resiliency, and security of VMware vSphere environments. Readers will learn how to convert technical constructs into operational mastery by adhering to well-established lessons learned through prior chapters and subsequent best practices.

Note: VMware is now part of Broadcom and is known as VMware by Broadcom. All references to VMware in this book reflect this change.

Structure

In this chapter, we will cover the following topics:

- Applying best practices from installation to management
- Networking best practices in vSphere environment
- Storage best practices in vSphere environment
- Virtual machine best practices
- CPU and memory best practices
- Best practices for vSphere clusters management

- Best practices for lifecycle management
- Security best practices in vSphere environment
- Best practice for a healthy vSphere environment
- Best practices for vSphere HA and Fault Tolerance

Objectives

By the end of this chapter, readers will gain the ability to implement comprehensive vSphere environment best practices from initial installation and configuration to ongoing maintenance and support. Moreover, readers will learn how to optimize resources regarding data retention, security, and system uptime. Focusing on practical application complements theoretical learning and equips the learner to build and manage virtual infrastructures that are robust, highly available, and enterprise-ready. From designing new environments or fine-tuning existing deployments, this chapter helps readers understand how to elevate their virtualization practices.

Applying best practices from installation to management

Starting off with a properly configured vSphere environment is crucial for ensuring long-term stability, performance, and scalability. Every step, from planning and installation to management, has room for optimization of the environment. The practices suggested below help establish a dependable environment by particularly reducing administrative workload and future operational risks.

The following standards for installation, configuration, and optimization will ensure a more efficient and resilient virtualization platform, irrespective of whether it is a lab or an enterprise data center.

The key best practices are as follows:

- **Installation and configuration:**
 - Plan thoroughly before installation, including network topology, IP schemes, host names, storage requirements, and resource sizing.

- Use the VMware compatibility guide to verify that all hardware and software components are supported.
- Install ESXi on high-performance storage (e.g., local SSD, SAN, or NVMe-based boot devices) for improved host performance.
- Configure hostnames and DNS resolution properly to prevent communication issues between vCenter and ESXi hosts.
- **Standardization:**
 - Use consistent naming conventions for **virtual machines (VMs)**, datastores, port groups, and clusters to simplify management and automation.
 - Enable **Network Time Protocol (NTP)** on all hosts to maintain time synchronization, which is critical for logs, certificates, and **Distributed Resource Scheduler (DRS)/ high availability (HA)**.
 - Centralize logging (e.g., to a syslog server) to improve troubleshooting and auditing across the environment.
- **Management practices:**
 - Document environment including changes, installed versions, and configuration details.
 - Use Host Profiles or templates to maintain configuration consistency across multiple ESXi hosts.
 - Automate regular tasks using vSphere CLI, PowerCLI, or **vSphere Lifecycle Manager (vLCM)** to reduce human error.
- **Resilience and recovery:**
 - Create and test a backup of the vCenter Server regularly, and ensure that backup and recovery procedures are well documented.
 - Deploy vCenter in **Enhanced Linked Mode (ELM)** if managing multiple sites or vCenter instances with due diligence.
 - Use **role-based access control (RBAC)** to ensure security and reduce configuration mistakes.

Real-world tip

In large-scale environments, configuration drift is one of the most common operational challenges. Over time, administrators may make host-level changes manually, such as modifying NTP settings, changing VMkernel **network adapter (NIC)** configurations, or adjusting security policies, which can cause inconsistencies across hosts.

For example, in a 20 host cluster, if NTP is misconfigured on just one ESXi host, it may fail to join the vSphere HA cluster or cause SSL certificate issues due to time skew. Similarly, if the Management VMkernel port is accidentally removed from a host during manual configuration, vCenter may lose connectivity to that host.

To prevent such issues, keep the following in mind:

- Use Host Profiles to apply and enforce uniform configuration across ESXi hosts.
- Combine Host Profiles with vLCM to monitor compliance and remediate drift automatically.
- In environments using tools like *Ansible* or *PowerCLI*, automate repetitive setup tasks (e.g., VMkernel network creation, syslog configuration, NTP assignment) to maintain consistency from day one.

This approach not only minimizes manual errors but also speeds up host deployment and streamlines compliance audits.

Networking best practices in vSphere environment

Networking forms the foundation of any virtualized infrastructure. Virtual switches, port groups, and VMkernel interfaces in VMware vSphere infrastructures need to be properly configured and managed to provide performance, security, and HA. Misconfiguration of a network can impact VM connectivity as well as trigger vSphere services such as vMotion, HA, and replication to fail.

This topic discusses the most critical best practices for building a stable, scalable, and secure virtual network in vSphere for both standard and distributed switches.

The key best practices are as follows:

- **Use vSphere Distributed Switches (VDSs) for enterprise**

deployments: Consolidate network management for hosts, support features such as port mirroring, NetFlow, LACP, and ease the application of consistent policies.

- **Implement NIC teaming and redundancy:** Designate a minimum of two physical NICs per category of (traffic management, vMotion, VM traffic, vSAN) to eliminate single points of failure.
- **Isolate network traffic using VLANs:** Isolate traffic types for performance and security (e.g., Backup, replication, vSAN, vMotion, management).
- **Enable Network I/O Control (NIOC):** Control important traffic such as vMotion and vSAN to enable service continuity in case of network contention.
- **Use VMkernel adapters properly:** Create specialized VMkernel NICs for every service, like vMotion, **fault tolerance (FT)**, vSAN, etc., to enhance performance and visibility.
- **Standardize MTU settings for jumbo frames:** Standardize the configuration of MTU (e.g., 9000 for vMotion or vSAN) on virtual and physical switches for improved throughput.
- **Implement port security policies:** Modify options like promiscuous mode, MAC address changes, and Forged transmits carefully, and disable them unless necessary.

Real-world tip

In the world of production, network redundancy and isolation are often underappreciated until some hardware fails. Consider a common scenario in which a single physical NIC is used for both vMotion traffic and management traffic. In the middle of a scheduled vMotion process, the NIC died, taking the host down and isolating it from vCenter. The following examples illustrate the impact of these practices:

- **Example 1:** A customer with 10G NICs had not segmented vMotion and management traffic. When vMotion was initiated for many large VMs, latency skyrocketed and caused management disconnects. When traffic was segmented with VLANs and NIOC was set up, the issue was resolved.

- **Example 2:** A business-critical VM could not be migrated in a healthcare data center because its VMkernel adapter for vMotion was not configured correctly on all hosts. The team then utilized VDS and Host Profiles to enforce consistency.

These problems underscore the importance of validating virtual network configurations during initial deployment and performing periodic audits thereafter.

Storage best practices in vSphere environment

Storage is a fundamental building block of any virtual infrastructure. The well-architected storage architecture delivers high-performance, scalability, and data availability to VMs and vSphere services, with the variety of storage technologies available, such as VMFS, NFS, vSAN, and vVols. The administrators must choose the most appropriate solutions according to organizational requirements, performance needs, and redundancy needs.

This section provides best practices for storage management in a vSphere environment with an emphasis on availability, optimization, and resiliency in various storage infrastructures.

The key best practices are as follows:

- **Match storage type to use case:** Employ VMFS in block-based SAN environments, NFS for file-based access, and vSAN for hyperconverged. Employ vVols for application-level granularity.
- **Support multipathing for redundant operations:** Storage paths must be configured with failover choices like round robin or **most recently used (MRU)** to enhance availability and load balancing.
- **Use storage policies for vSAN and vVols:** Define and apply storage policies that match VM performance and availability requirements, like **failures to tolerate (FTT)** in vSAN.
- **Align block size and disk provisioning:** Choose appropriate block sizes and disk allocation modes (thin, thick lazy-zeroed, thick eager-zeroed) based on the VM workload type.
- **Monitor datastore space and latency periodically:** Instrument alarms for datastore space utilization and real-time storage latency (read/write)

to avoid performance loss.

- **Enable Storage I/O Control (SIOC):** Utilize SIOC to prioritize I/O of important workloads during contention and prevent VM performance bottlenecks due to noisy neighbors.
- **Avoid oversubscription of thin-provisioned storage:** Monitor actual usage to prevent space exhaustion in environments heavily using thin provisioning.
- **Standardize VM placement strategy:** Avoid unbalanced VM placement across datastores. Use Storage DRS to automate placement and rebalance I/O loads in an intelligent manner.

Real-world tip

Storage mismanagement can lead to performance bottlenecks, service outages, or even data loss. In one financial institution, a VM with critical workloads was running on a thin-provisioned datastore. When multiple VMs simultaneously grew their disks, the datastore filled up and crashed multiple VMs. The following examples illustrate the impact of these practices:

- **Example 1:** In a VMFS data store company, there were persistent latency issues since all VMs were grouped onto a single LUN. The administrator installed Storage DRS, and the load was redistributed automatically between existing datastores; thus, the performance was significantly improved.
- **Example 2:** SIOC was deployed to a shared datastore by one organization with SQL and dev VMs. SQL queries were being delayed in high-demand times because of dev workloads. SIOC ensured SLA by giving I/O priority to the database VMs.

These examples highlight that ideal storage habits go beyond simple capacity issues; they also include handling performance, policy, and protection across the entire environment.

Virtual machine best practices

VMs form the basis of a vSphere infrastructure. Accurate configuration, resource allocation, and VM lifecycle management ensure application

stability, performance, and scalability in the virtual data center. From provisioning to regular maintenance, administrators need to adopt practices that meet business needs, workload demands, and infrastructure capabilities.

This section emphasizes practices that are crucial in ensuring smooth VM functioning, minimizing possible risks, and maintaining consistency in the environment.

The key best practices are as follows:

- **Utilize templates to ensure consistency:** Create VM templates with preinstalled OS, tools, and security settings to ensure consistency across deployments.
- **VM optimization:** Avoid overprovisioning of CPU and memory capacity. Utilize performance monitoring products to monitor actual resource needs before scaling up or down.
- **Enable VMware Tools:** Keep VMware Tools up to date on all VMs for compatibility, guest performance, and to support features like vMotion and VM Monitoring.
- **Isolate high-load workloads:** Place resource-intensive VMs on dedicated hosts or clusters where necessary to avoid contention and maintain predictable performance.
- **Use thin provisioning wisely:** Although thin provisioning conserves storage initially, monitor datastore usage periodically to prevent overcommitment and space depletion.
- **Do not use snapshots as backups:** Snapshots are temporary (preferably not exceeding 72 hours). Utilize appropriate backup techniques for data protection in the long-term.
- **Use VM tags and folders:** Organize and categorize VMs with tags and folders for simpler management, automation, and reporting.
- **Harden the VM configuration:** Disable unused devices (e.g., floppy disk drive), utilize VMX configuration to harden VMs, and limit console access to avoid misuse.

Real-world tip

VM sprawl, where VMs are created without control, can lead to bloated environments, increased license costs, and resource wastage. The following

examples illustrate the impact of these practices:

- **Example 1:** A mid-sized enterprise had hundreds of unused or abandoned VMs still consuming CPU and storage. By implementing a regular VM review and retirement policy, the company reclaimed 20% of its storage and cut licensing costs.
- **Example 2:** An admin created a snapshot for an OS patch and forgot to delete it. Months later, the snapshot grew to 400 GB, consuming datastore space and impacting VM performance. From that point, a policy was enforced to delete snapshots within 72 hours unless explicitly extended with approval.

These examples reinforce the importance of structured VM management practices and regular audits to avoid silent failures and resource wastage.

CPU and memory best practices

Optimal CPU and memory distribution is at the center of a high-performance vSphere. Overcommitment and improper VM configuration result in performance bottlenecks, resource contention, and poor hardware utilization. Performance, scalability, and fairness need to be balanced by administrators in CPU and memory allocation to VMs.

This chapter outlines well-tested methods for efficient distribution and management of computational power in a virtualized environment.

The key best practices are as follows:

- **Avoid overprovisioning resources:** Assign CPU and memory according to workload requirements, not estimates. Overprovisioning reduces consolidation ratios and may lead to unnecessary contention.
- **Use shares, limits, and reservations wisely:** Use shares to define resource priority in conflict; apply reservations to only critical VMs, and reserve limits only when necessary.
- **Monitor CPU ready time:** High CPU ready time indicates that VMs are sitting idle, waiting for CPU time. Monitor this periodically to identify CPU contention.
- **Keep vCPU count in check:** Assign the minimal number of vCPUs required. Unused vCPUs can increase scheduling delays.

- **Use memory hot add with caution:** Memory hot add disables NUMA optimizations. Use it only when dynamic memory addition is essential.
- **Enable memory ballooning and transparent page sharing (TPS):** Allow the ESXi host to reclaim memory efficiently. Check that the balloon driver is enabled via VMware Tools. TPS can be beneficial for optimizing memory usage, but it requires careful consideration from a security perspective. Administrators are recommended to follow the below knowledge base articles and resources to ensure a thorough review of security rules and best practices before implementing TPS:
 - **Broadcom KB:** Article 2080735
 - **Broadcom KB:** Article 2097593
 - **VMware Blogs:** Assessing the performance impact of security changes in TPS behavior
- **Account for overhead memory:** All VMs use some overhead memory to run virtual processes. Consider this when calculating total memory allocation on hosts.
- **Use vSphere DRS and host affinity rules:** For resource balancing across clusters, let DRS distribute VMs intelligently based on CPU/memory availability.

Real-world tip

Misconfigured CPU and memory resources are among the top causes of performance complaints in virtual environments. The following examples illustrate the impact of these practices:

- **Example 1:** A team allocated 8 vCPUs to a lightweight web application VM, thinking *more is better*. The result: increased CPU wait times and degraded performance. After resizing the VM to 2 vCPUs, performance and scheduling efficiency improved dramatically.
- **Example 2:** In a production database environment, administrators failed to reserve memory for a critical VM. During host memory contention, the VM was swapped to disk, severely affecting query performance. Enabling a reservation ensured consistent performance going forward.

These examples emphasize that smarter resource allocation is key to stable

and efficient performance, and not just more resources.

Best practices for vSphere clusters management

Clusters are the foundations of resilience and resource pooling in a vSphere environment. Well-designed and well-configured clusters enable administrators to maximize the benefits of advanced features like HA, DRS, and vSAN. Poorly organized clusters, on the other hand, can result in imbalances, inefficiencies, and potentially even cause services to be disrupted.

This chapter deals with best practices that guarantee safe, reliable, and solid operation of clusters in small-scale and large-scale organizational contexts.

The key best practices are as follows:

- **Standardize host configurations across the cluster:** Ensure all hosts in a cluster have consistent hardware, network, and storage configurations to avoid compatibility issues with vMotion, HA, and DRS.
- **Use the Cluster Quickstart for initial configuration:** Utilize the Cluster Quickstart procedure to simplify initial configuration and impose consistency when turning on features such as HA and DRS.
- **Enable DRS and HA together:** Use vSphere DRS for intelligent VM load distribution and vSphere HA for automated VM restarts during host failures. Together, they create a resilient and balanced infrastructure.
- **Avoid mixing incompatible host CPUs:** Use **Enhanced vMotion Compatibility (EVC)** to mask CPU differences and enable seamless VM migration across hosts with slightly different processor generations.
- **Check cluster health periodically:** Use the vSphere Client's cluster-level monitoring tools to check CPU, memory, storage, and network utilization across all hosts.
- **Capacity overhead strategy:** Reserve resources to accommodate failover scenarios by configuring vSphere HA admission control with appropriate policies (e.g., cluster resource percentage).
- **Use host affinity and anti-affinity rules with care:** Apply VM placement rules only when necessary, and document their impact. Improper rules may conflict with DRS load balancing or HA recovery.

actions.

Real-world tip

An up-to-date cluster makes administration easier and facilitates business continuity through forward-looking planning. The following examples illustrate the impact of these practices:

- **Example 1:** An organization deployed a 10 host cluster but forgot to configure EVC. When they added a newer host model, vMotion Compatibility failed, disrupting DRS operations. Enabling EVC earlier would have prevented this.
- **Example 2:** In another case, an admin applied strict anti-affinity rules to a set of VMs across all hosts, unaware it would block vSphere HA during a host failure. Revisiting the rules and adjusting their scope enabled HA to perform successful recoveries during testing.

These illustrations underscore the value of foresight and documentation in defining cluster-level characteristics.

Best practices for lifecycle management

It is crucial that a vSphere environment remains current and secure, not just recommended. Lifecycle management involves patching, upgrading, and achieving consistency between ESXi hosts, vCenter Server, VMware Tools, and VM hardware. When done properly, it reduces system downtime, lowers vulnerabilities, and improves overall performance.

This section describes best practices that enable administrators to utilize vLCM optimally and maintain long-term infrastructure health.

The key best practices are as follows:

- **Use vLCM for cluster-wide updates:** Use vLCM to control ESXi host versions within clusters based on target images for uniform patching and upgrades.
- **Upgrade vCenter Server first:** Upgrade the vCenter Server before upgrading any of the ESXi hosts. This ensures compatibility and access to new features.

- **Run interoperability checks before upgrades:** Use the VMware Interoperability Matrix and the Update Planner to ensure that third-party software, drivers, and plugins are supported in new releases.
- **Regularly upgrade VMware Tools and VM hardware:** Keep VMware Tools and virtual hardware versions up to date for enhanced stability, performance, and new guest OS features.
- **Use Host Profiles or desired state images:** Create cluster baselines via images or Host Profiles so that repetitive configurations are ensured and drift is avoided.
- **Schedule upgrades during maintenance windows:** Perform host and cluster upgrades during planned downtime to avoid unexpected service disruption.
- **Verify compliance regularly:** Use the compliance check feature in vLCM to identify noncompliant hosts or outdated components.

Real-world tip

A safe and well-maintained environment is only possible when lifecycle tasks are uniform and automated. The following examples illustrate the impact of these practices:

- **Example 1:** An admin once skipped checking interoperability before upgrading vCenter and found that their third-party backup plugin stopped working. Utilizing the Update Planner would have flagged it ahead of time.
- **Example 2:** In another scenario, an organization used vLCM to push a security patch to a 16 host cluster. Through the enforcement of desired images, each host was running the same secure build, avoiding human errors, and improving compliance reporting.

Lifecycle management is not about upgrades. It is about preserving operational continuity, security, and compliance in the future. In today's fast-changing data centers, automation is not a choice; it is a foundation.

Security best practices in vSphere environment

In every virtualized environment, the paramount nature of security cannot be

overemphasized. VMware vSphere has several protection layers that span from secure host setup to RBAC to encrypted data transfer. In poorly secured environments, threats such as data breaches, privilege escalation, and malware infection are pending dangers, issues that can quite seriously impede business continuity and regulatory compliance.

This topic outlines key best practices for administrators to protect their vSphere infrastructure within the compute, network, and storage domains.

The key best practices are as follows:

- **Implement RBAC:** Define roles and assign the least rights required for users or service accounts to perform tasks.
- **Use ESXi lockdown mode:** Restrict access to hosts directly by enabling lockdown mode and managing ESXi hosts using just vCenter.
- **Harden the vCenter Server appliance (vCSA):** Keep the vCSA updated, enforce strong password policies, and monitor logs using the built-in appliance shell tools.
- **Encrypt vMotion and VM data:** Enable encrypted vMotion and VM encryption for sensitive workloads to protect data in transit and at rest.
- **Enable and monitor secure boot for ESXi hosts:** Secure boot ensures that only signed and verified code runs on your ESXi hosts, secure from tampering or injection of malware.
- **Perform regular audits and refresh access credentials:** Remove inactive accounts, regularly review permission assignments, and regularly refresh passwords or certificates.
- **Activate vSphere Native Key Provider (NKP):** Use the NKP feature for managing encryption keys without an external **Key Management Server (KMS)**, simplifying secure deployments.
- **Leverage VMware Tools for guest OS hardening:** Use guest introspection, anti-malware agents, and file integrity tools integrated via VMware Tools to extend protection inside VMs.

Real-world tip

In most breach investigations, misconfigured permissions and stale security settings are the usual perpetrators. The following examples illustrate the impact of these practices:

- **Example 1:** In a test environment, a service account was discovered to have too many admin privileges. RBAC fixed the issue and ensured least-privilege access in the future.
- **Example 2:** A financial institution flagged a compliance gap when encrypted vMotion was missing from its infrastructure. By enabling vSphere NKP and applying VM encryption to its most sensitive workloads, the team quickly met security standards without deploying a third-party KMS.

By adopting layered security and automation, organizations can safeguard their virtual kingdom against growing threats while meeting regulatory and business demands.

Best practice for a healthy vSphere environment

A well-functioning virtual infrastructure depends not only on good design principles but also on ongoing monitoring and real-time troubleshooting. Monitoring provides feedback on system state in operation, allowing administrators to detect anomalies before they escalate into full-blown issues. Troubleshooting, however, proves useful once issues occur, allowing root-cause analysis and corrective measures.

VMware vSphere has native monitoring capabilities through vCenter performance charts, alarms, and the ESXi host logs. For a larger picture and correlation, products like VMware Aria Operations (formerly **vRealize Operations (vROps)**) and VMware Aria Log Insight assist in detecting trends, proactive issue detection, and support for automated remediation. Regular monitoring of performance, capacity, or compliance ensures infrastructure resilience.

The monitoring and troubleshooting are explained as follows:

- **Leverage vSphere alarms proactively:**
 - Configure alarms for CPU, memory, disk, and network usage at both host and VM levels.
 - Set thresholds based on normal patterns of workload.
 - Use email or SNMP alerts to provide timely reactions to alerts or

significant events.

- **Use VMware Aria operations (vROps):**
 - Install vROps to visualize capacity, performance, and health throughout the environment.
 - Utilize predictive analytics to ascertain anomalies prior to any impact on users.
 - Enable automated remediation wherever feasible, for example, recovering idle resources or load balancing.
- **Enable log collection and analysis:**
 - Leverage VMware Aria Log Insight or other SIEM tools to collect log information from vCenter, ESXi, VMs, and network and storage devices.
 - Develop customized dashboards for chronic issues or highlight critical logs.
- **Maintain baseline performance benchmarks:**
 - Capture initial baseline performance data for CPU, memory, disk, and latency.
 - Continuously compare baselines with current measures to identify deviations.
- **Correlate performance with configuration changes:**
 - Utilize vROps change tracking and vSphere events view to identify what changes to the configuration accompany performance degradation or instability.
 - Always update the change logs while performing major changes.
- **Track resource contention metrics:**
 - Highlight CPU ready time, memory ballooning, swapping, and disk latency to identify oversubscription.
 - Use DRS scores and performance charts to highlight any discrepancies.
- **Automate with health dashboards:**

- Utilize vROps or vCenter health dashboards to view clusters, hosts, and datastores at a glance.
- Create executive-level views for senior management as needed.
- **Regularly audit system logs:**
 - Periodically review vCenter, ESXi, and VM logs for hidden problems.
 - Securely store logs and ship to external logging infrastructure for retention and compliance.

Real-world tip

Monitoring is what identifies issues early on, but troubleshooting is what turns transparency into action. The following examples illustrate the impact of these practices:

- **Example 1:** In one production environment, admins noticed intermittent VM slowness. A quick look at vCenter performance charts showed a spike in CPU Ready Time, revealing an overcommitted cluster. Adding a host and adjusting shares instantly improved performance.
- **Example 2:** In another case, Aria Log Insight logs assisted in identifying repeated failed login attempts by a defective backup agent. Without centralized logging, this might have remained undetected for weeks and posed a potential security vulnerability.

Monitoring and troubleshooting are not reactive activities; they are proactive disciplines. By cultivating a culture of observability and correlating numbers to system behavior, admins can transition from firefighting to foresight, not just getting the infrastructure to bounce back quickly but preventing it from crashing in the first place.

Best practices for vSphere HA and Fault Tolerance

HA is no longer an option; it is a necessity for most production workloads today. VMware vSphere provides strong HA in the form of vSphere HA and vSphere FT. They are designed to reduce downtime due to hardware failure, application failure, and unplanned downtime, to provide continuity of service

to business-critical applications.

While vSphere HA provides restart functionality on failure, vSphere FT goes one step further with no downtime and loss of data by creating a live shadow copy of a VM. Spanning across these features, however, requires thoughtful design to balance resource overhead, redundancy, and recovery expectations.

The key best practices are as follows:

- **Enable admission control and set adequate policies:**
 - Set aside adequate capacity for VMs to restart in case of host failure.
 - Use the policy cluster resource percentage as an adjustable and precise failover calculation.
- **Configure redundant heartbeat and isolation settings:**
 - Use NIC teaming or multiple VMkernel ports for redundant heartbeat paths.
 - Provide additional isolation that covers and handles both vSAN and management networks.
- **Implement VM Monitoring and VM Component Protection (VMCP):**
 - Enable VM Monitoring to detect OS or application failures.
 - Use VMCP to recover from **All Paths Down (APD)** or **Permanent Device Loss (PDL)** storage events.
- **Group critical VMs with restart priorities and dependencies:**
 - Establish VM restart priorities (high, medium, low) according to application priority.
 - Use orchestrated restart for multi-tier applications like databases, middleware, and frontend servers.
- **Regularly test HA and FT functionality:**
 - Perform controlled failover and failback tests to validate HA/FT response.
 - Monitor for any configuration drift or misalignment across hosts.
- **Monitor cluster health and utilization:**

- Use the HA Cluster Monitoring pane to view failover readiness.
- Validate that host isolation responses and restart priorities are operating as expected.
- **Use vSphere FT for mission-critical VMs:**
 - Support zero-downtime and zero-data-loss workloads.
 - Ensure adequate host resources and compatible CPU configurations across cluster nodes.
- **Avoid placing FT primary and secondary VMs on the same host:**
 - This violates FT principles. DRS and vMotion rules must enforce separation.
 - Support zero-downtime and zero-data-loss workloads.

Real-world tip

Failover is only as good as the infrastructure that supports it. Plan capacity, simulate failures, and configure wisely. The following examples illustrate the impact of these practices:

- **Example 1:** An admin once configured a cluster without admission control to save resources. When a host failed, the cluster could not restart several VMs due to a lack of capacity. Admission control would have prevented this by refusing VM power-on beyond failover capacity.
- **Example 2:** FT was adopted by a financial institution for its VM that handled transaction processing. When a host failed, the secondary VM took over without loss of data. This safeguard saved costly reconciliation efforts and allowed compliance audits to be done effectively.

Virtual HA is not a coincidence but by design. By best practice deployment of vSphere HA and FT configuration, administrators can design clusters that not only survive failure, but remain capable of maintaining service integrity despite it. Reliability is integrated into the platform, but it has been made accessible by careful design and ongoing testing.

Conclusion

Success in virtualization is much more than deploying technology. It requires careful planning, strict management, and relentless improvement. In this chapter, we have combined important best practices, which have been developed over years of industry experience, and showcased how they can be used to improve performance, security, and stability in VMware vSphere environments.

Virtualized data centers in constructed and proactively monitored environments are enhanced at every operational level, from network design to resource allocation and security. The readers were presented with applied case studies and recommended practices that illustrate how top-tier corporations operate sophisticated virtual infrastructures.

Equipped with these best practices, readers are now prepared to manage a virtualization platform that is responsive and high-performing.

The journey culminates in the *Appendix, Mock Exams and Preparation*, where all the knowledge gathered throughout the book is compiled. Through this, readers will receive essential guidance concerning the VCP-DCV certification, exploring exam preparation strategies and review guides, mock tests, and explaining how this achievement transforms virtualization career pathways.

Join our Discord space

Join our Discord workspace for latest updates, offers, tech happenings around the world, new releases, and sessions with the authors:

<https://discord.bpbonline.com>



APPENDIX

Mock Exams and Preparation

Introduction

Taking mock exams is essential to build confidence and manage time under pressure. They help reinforce memory, improve recall, and reduce test anxiety.

Undertake the following measures:

- Use online platforms with VCP-DCV mock exams.
- Focus on weak areas highlighted in your practice results.
- Review not just correct answers but also why the others are incorrect.
- Simulate a test-like environment, no distractions, timed sessions, and a quiet room.

Practice questions

The following are practice questions for references:

1. **Which feature helps maintain zero loss when a VM on an ESXi server fails unexpectedly?**
 - a. vMotion
 - b. Fault Tolerance

- c. Storage vMotion
 - d. vSphere High Availability
2. **An administrator is responsible for maintaining a single cluster VMware solution with the following characteristics:**
- a. A single VMware vCenter is deployed.
 - b. The solution hosts critical, Network I/O Control intensive workloads.
 - c. At the hardware level, each node is identically configured with two CPUs (16-cores), 512GB RAM and four 10GbE connections.
 - d. Each host currently uses vSphere Standard Switches.

After completing some maintenance tasks requiring the administrator to live migrate workloads onto another ESXi host, the administrator has noticed that the live migration of workloads takes a very long time.

Which three actions should the administrator take to ensure the time required to live migrate workloads between nodes within the cluster is reduced? (Choose three)

- a. Configure a vSphere Distributed Switch and add each ESXi host to it.
 - b. Enable Network I/O Control to ensure that sufficient bandwidth for system traffic is reserved.
 - c. Enable Network I/O Control to ensure that sufficient bandwidth for **virtual machine (VM)** traffic is reserved.
 - d. Configure a new vSphere Standard Switch on each ESXi host.
 - e. Migrate all the workloads and networking to the new vSphere Distributed Switch.
 - f. Migrate some of the workloads to the new vSphere Standard Switch.
3. **An administrator is asked to copy a single hard disk from one datastore to another. Which VM file type should the administrator select?**
- a. vmdk
 - b. vswp
 - c. vmx

d. vmss

4. **An administrator has enabled vSphere Cluster Services (vCLS) Retreat Mode on a cluster with three ESXi hosts. What will be the impact of this change in the event of a host failure?**

- a. vSphere High Availability optimal placement will not be available on the cluster.
- b. vSphere **Distributed Resource Scheduler (DRS)** will be set to disabled on the cluster.
- c. vSphere High Availability will be set to disabled on the cluster.
- d. Enhanced **vMotion Compatibility (EVC)** will not be available on the cluster.

5. **An administrator is tasked with deploying a new software-defined data center (SDDC) that will contain five VMware vCenter instances. The following requirements must be met:**

- **All vCenter instances should be visible in a single vSphere client session.**
- **All vCenter inventory should be searchable from a single vSphere client session.**
- **Any administrator must be able to complete operations on any vCenter instance using a single set of credentials.**

A combination of which two configurations should the administrator use to meet these requirements? (Choose two)

- a. The first vCenter instance should create a new vCenter Single Sign-On domain.
- b. The first vCenter instance should create a remote vCenter Single Sign-On domain.
- c. The first vCenter instance should create a multi-tenant vCenter Single Sign-On domain.
- d. Any additional vCenter instances should join the existing vCenter Single Sign-On domain.
- e. Any additional vCenter instances should create standalone vCenter Single Sign-On domains.

6. **Which vSphere component centralizes automated patch and version**

management for clusters. ESXi. drivers and firmware. Virtual hardware, and VMware Tools?

- a. vSphere DRS
 - b. Content Library
 - c. VMware Cloud Foundation
 - d. vSphere Lifecycle Manager
7. **An administrator configures a vSphere cluster to use vSphere Lifecycle Manager images for managing host version compliance. Which action should the administrator take to find the latest verified software available in the vSphere Lifecycle Manager depot?**
- a. Check hardware compatibility
 - b. Check compliance
 - c. Check for recommended images
 - d. Manage depot overrides
8. **Which file system protocol is supported by vSphere?**
- a. NFS
 - b. iSCSI
 - c. SAN
 - d. vSAN
9. **When configuring vCenter Identity Provider Federation in vSphere, which three pieces of information are required? (Choose three)**
- a. LDAP address
 - b. Client identifier for the application group
 - c. Shared secret for the application group
 - d. Server application name
 - e. One time passcode
 - f. OpenID address
10. **Which two things should an administrator consider when tasked with deploying new encrypted VMs into an existing VMware vSphere environment? (Choose two)**
- a. VM encryption is only supported when a datastore is backed by **self-encrypting drives (SEDs)**.

- b. All VM data (excluding swap files) is encrypted when using VM encryption.
 - c. Once encrypted, the process of unencrypting a VM is destructive.
 - d. VM encryption works uniformly across all supported guest operating systems.
 - e. All VM data (including swap files) is encrypted when using VM encryption.
- 11. You are implementing vCenter High Availability for a vSphere 8.0 environment. Which of the following does not need to be connected to the vCenter High Availability network?**
- a. Active Node
 - b. Witness Mode
 - c. ESXi
 - d. vCenter Server
- 12. You are managing certificates in your vSphere environment. By default, what types of certificates are in VECS? (Choose two)**
- a. ESXi Certificates
 - b. Machine SSL Certificates
 - c. Trusted Root Certificates
 - d. None of the above
- 13. Which policy is used for intelligent optimization of network interface traffic on a vSphere Distributed Switch (VDS)?**
- a. Route Based on Physical NIC Load
 - b. Route Based on Originating Virtual Port
 - c. Route Based on Source MAC Hash
 - d. Route Based on IP Hash
- 14. A vSphere administrator is receiving complaints a database VM is experiencing performance issues. The VM is a member of the high priority resource pool and the cluster has not experienced contention. Which condition should be checked to address immediate performance concerns?**
- a. Resource pool share value

- b. VM snapshot
 - c. Configured CPU shares
 - d. .VMFS version
15. **A vSphere administrator needs to quickly move a critical VM from an AMD-cluster to an Intel-cluster. How can the vSphere administrator move the VM?**
- a. Cold migration
 - b. vSphere encrypted vMotion
 - c. vSphere Storage vMotion
 - d. vSphere vMotion
16. **Which two datastore types store the components of a VM as a set of objects? (Choose two).**
- a. VMware VM File System (VMFS)
 - b. VMware vSAN
 - c. Network File System (NFS) 3
 - d. vSphere Virtual Volumes (vVols)
 - e. NFS 4.1
17. **An administrator needs to consolidate several physical servers by migrating the workloads to a software-defined data center solution. Which VMware solution should the administrator recommend?**
- a. VMware Horizon
 - b. VMware vSAN
 - c. VMware vSphere
 - d. VMware NSX
18. **An administrator is adding a new ESXi host to an existing vSphere cluster. When selecting the cluster, the administrator is unable to use the Cluster Quickstart workflow to add and configure the addition host. What could be the root cause of this issue?**
- a. The administrator has previously dismissed the Cluster Quickstart workflow.
 - b. The administrator must manually add the host to the cluster before using the Cluster Quickstart workflow.

- c. The administrator has not been assigned the required permission to use the Cluster Quickstart workflow.
 - d. The administrator must enable the Cluster Quickstart workflow option in VMware vCenter.
19. **A company has two sites: Site A and Site B. The administrator would like to manage the VMware vCenter inventories in both sites from a single vSphere client session?**
- a. VMware Certificate Authority
 - b. VMware Site Recovery Manager
 - c. vCenter Single Sign-on
 - d. Enhanced Linked Mode
20. **Which four elements can a vSphere Lifecycle Manager image contain? (Choose four)**
- a. ESXi base image
 - b. ESXi configuration
 - c. Vendor agents
 - d. Vendor add-ons
 - e. BIOS updates
 - f. Firmware and drivers add-on
 - g. Independent components
21. **An administrator is investigating reports of users experiencing difficulties logging into a VMware vCenter instance using LDAP account. Which service should the administrator check as part of troubleshooting?**
- a. VMware Authentication Proxy Service
 - b. Lookup Service
 - c. Identity Management Service
 - d. VMware Authentication Framework Daemon
22. **An administrator is tasked with installing VMware vCenter. The vCenter Server Appliance must support an environment of:**
- 400 hosts**
- 4000 VMs**

Which two resources must be allocated, at a minimum, to meet the requirements? (Choose two)

- a. 16 vCPUs
- b. 30GB memory
- c. 4 vCPUs
- d. 8 vCPUs
- e. 20GB memory

23. When configuring vCenter High Availability, which two statements are true regarding the active, passive, and witness nodes? (Choose two)

- a. Network latency must be less than 10 milliseconds.
- b. They must have a supported **wide area network (WAN)**.
- c. They must have a minimum of a 10GBps network adaptor.
- d. They must have a minimum of a 1GBps network adaptor.
- e. Network latency must be more than 10 milliseconds.

24. Which VMware offering will allow an administrator to manage the lifecycle of multiple vCenter Server instances in a single software as a service (SaaS)-based solution to help drive operational efficiency?

- a. VMware vSphere with Tanzu
- b. VMware Cloud Foundation
- c. VMware vSphere+
- d. VMware Aria Suite Lifecycle

Answers

1	b
2	a, b, e
3	a
4	a
5	a, d
6	d

7	c
8	a
9	b, c, f
10	d, e
11	c
12	b, c
13	a
14	b
15	a
16	b ,d
17	c
18	a
19	d
20	a, d, f, g
21	c
22	b, d
23	a, d
24	c

Join our Discord space

Join our Discord workspace for latest updates, offers, tech happenings around the world, new releases, and sessions with the authors:

<https://discord.bpbonline.com>



Index

A

Access Control [141](#)
Access Control, concepts [141](#), [142](#)
Automation Level [301](#)
Automation Level, ensuring [301](#), [302](#)

C

Cloning [210](#)
Cloning, points
 Efficiency [210](#)
 Powered-On [210](#)
 Storage [210](#)
 Template [210](#)
 vCenter Requirement [210](#)
Cloning, terms
 Customization Specifications [212](#)
 Guest Operating System (OS) [211](#)
Cloud Infrastructure [23](#)
Cloud Infrastructure, illustrating [23-25](#)
Cluster Lifecycle Management [365](#)
Cluster Lifecycle Management, approach
 Cluster Image [367](#)
 Cluster Staging [369](#)
 Image Depot [367](#)
 Parallel Remediation [372](#)
 Remediation Pre-Check [369](#)
 vSphere Configuration Profiles [377](#)
Cluster Lifecycle Management, configuring [365](#), [366](#)
Cluster Quickstart [288](#)
Cluster Quickstart, configuring [288](#)
Cluster Quickstart, terms
 ESXi Hosts [290](#)
 Network Configuration [291](#)
 Resource Distribution [289](#)
Cluster Summary Information [295](#)
Connection State [194](#)
Connection State, terms
 Network Adapters [195](#)
 PCI Passthrough [195](#)

- Virtual Devices [196](#)
- Content Libraries [213](#)
- Content Libraries, advantages [214](#)
- Content Libraries, benefits
 - Effective Storage Management [213](#)
 - Global Access [213](#)
 - Version Control [213](#)
- Content Libraries, fundamentals
 - Administrators [224](#)
 - Advanced Configuration [226](#)
 - OVF Templates [223](#)
 - Publishing [221](#)
 - Simple Versioning [226](#)
 - Single Sign-On (SSO) [225](#)
 - Subscribing [221](#)
 - vCenter Instance [227](#)
 - Viewing [222](#)
- Content Libraries, integrating [218](#)
- Content Libraries, steps [216](#)
- Content Libraries, template
 - OVF [217](#)
 - VM [217](#)
- Content Libraries, types
 - Local [215](#)
 - Published [215](#)
 - Subscribed [215](#)
- CPU Load Balancing [273](#)
- Cross vCenter Migration [255](#)
- Cross vCenter Migration, requirements
 - Storage Access [256](#)
 - Time Synchronization [256](#)
 - vCenter SSO Domain [256](#)
 - Version Compatibility [256](#)
- Cross vCenter Migration, terms
 - Executing [256](#)
 - Network Compatibility [259](#)
 - vCenter Instance [257](#)
- Customization Specifications [212](#)

D

- Data Center Virtualization [3](#)
- Data Center Virtualization, advantages
 - Cost Saving [3](#)
 - Flexibility [3](#)
 - Resource Utilization [3](#)
 - Simplified Management [3](#)
- Datastore [131](#)
- Datastore, methods

- Block-Backed Storage [132](#)
- Direct-Attached [133](#)
- VMFS [133](#)
- Datastore, terms
 - Network File System (NFS) [134](#)
 - Raw Device Mapping (RDM) [137](#)
 - Virtual Volumes [136](#)
- Datastore, types
 - NFS [131](#)
 - Virtual Volumes [131](#)
 - VMFS [131](#)
 - vSAN [131](#)
- Discovery Protocols [107](#)
- Discovery Protocols, types
 - CDP [107](#)
 - LLDP [107](#)
- Disk Array [143](#)
- Distributed Switches [104](#)
- Distributed Switches, components [105](#)
- Distributed Switches, planes
 - Control [105](#)
 - I/O [105](#)
- Distributed Switches, topology
 - Discovery Protocols [107](#)
 - Inbound Traffic Shaping [109](#)
 - Physical NIC Load Balancing [110](#)
 - Port Binding [108](#)

E

- ELM, features
 - Centralized Login [52](#)
 - Flexible Deployment [52](#)
 - Licensing Requirements [52](#)
 - Replication Across Instances [52](#)
 - Unified Inventory Management [52](#)
- ELM, flow
 - Host Management Daemon [53](#)
 - vCenter Agent [53](#)
 - vpxd process [53](#)
- Enhanced Linked Mode (ELM) [52](#)
- ESXi [30](#)
- ESXi, configuring [30](#), [31](#)
- ESXi, features
 - Quick Boot [30](#)
 - SAN LUNs [30](#)
 - Security Enhancing [30](#)
 - Small Disc Footprint [30](#)
- ESXi Host [32](#), [33](#)

- ESXi Host, foundation
 - NTP Client Communication [37](#)
 - PTP Client [38](#)
 - Remote Access [39](#)
 - Root Access [34](#)
 - Time Synchronization [36](#)
 - Troubleshoot Services [35](#)
- ESXi Host, modes
 - Direct Console User Interface (DCUI) [33](#)
 - vSphere Client [33](#)
- ESXi Image [362](#)
- ESXi, installing [31](#)
- ESXi, practices [39](#), [40](#)
- EVC [245](#)
- EVC, architecture [245](#), [246](#)
- EVC Cluster, sources
 - CPU Vendor [246](#)
 - Execution Disable [246](#)
 - Hardware Virtualization [246](#)
 - ID Compatibility [246](#)
 - vSphere vMotion [246](#)
- EVC CPU Mode [248](#)
- EVC, directions
 - Decreasing [248](#)
 - Increasing [248](#)
- EVC, illustrating [247](#)

F

- FC SAN [139](#)
- FC SAN, architecture [140](#)
- FC SAN, components [140](#), [141](#)
- FC SAN, terms
 - Access Control [141](#)
 - Disk Array [143](#)
 - Physical Paths [143](#)
- Fibre Channel (FC) [139](#)

G

- GPU, configuring [20](#)
- Graphics Mode [250](#)
- Graphics Mode, illustrating [250](#)
- Graphics Processing Unit (GPU) [20](#)
- Guest Operating System (OS) [183](#), [211](#)
- Guest OS, integrating [212](#)

H

[Hyperthreading 272](#)

I

[Image Depot 362](#)

[Image Depot, ensuring 363](#)

[Image Depot, integrating 364](#)

[Inbound Traffic Shaping 109](#)

[iSCSI 144](#)

[iSCSI, architecture 144, 145](#)

[iSCSI, components](#)

[Adaptors 146](#)

[Addressing 145](#)

[Administrator Functional 149](#)

[CHAP Authentication 151](#)

[Fault Tolerance 156](#)

[Network Configuration 147](#)

[Performance/Redundancy 153](#)

[Speed/Redundancy 155](#)

[Target Discovery 150](#)

[VMkernel Ports 157](#)

L

[Lifecycle Management 356](#)

[Lifecycle Management, tips](#)

[Interperability Report 357](#)

[Patching vCenter 358](#)

[Load Balancing 119](#)

[Load Balancing, characteristics](#)

[Destination IP Hash 119](#)

[Source MAC Hash 119](#)

[Virtual Port ID 119](#)

[Log Levels 87](#)

[Log Levels, configuring 87, 88](#)

[Log Levels, terms](#)

[ESXi Hosts 89](#)

[Remote Syslog Server 88](#)

[Long-Distance vSphere vMotion 261](#)

M

[Memory Overcommitment 270](#)

[Memory Overcommitment, methods](#)

[Ballooning 271](#)

[Host-Level SSD Swapping 271](#)

[Memory Compression 271](#)

[Memory Paging 271](#)

[Transparent Page Sharing \(TPS\) 271](#)

- Memory Virtualization [269](#)
- Memory Virtualization, terms
 - CPU Load Balancing [273](#)
 - Hyperthreading [272](#)
 - Memory Overcommitment [270](#)
 - Multicore VMs [271](#)
- Migration Threshold [302](#)
- Migration Threshold, illustrating [302](#), [303](#)
- Multicore VMs [271](#)
- Multi-Homing [62](#)
- Multi-Homing, benefits
 - Network Performance [63](#)
 - Security, enhancing [63](#)
 - Traffic Segregation [63](#)
- Multi-Homing, features [62](#)
- Multi-Homing, illustrating [63](#)

N

- Network Connections [97](#)
- Network Connections, ports
 - Uplink [97](#)
 - VM [97](#)
 - VMkernel [97](#)
- Network Failure [120](#)
- Network Failure, features
 - Failback Behaviour [120](#)
 - Failover Management [120](#)
 - Failure Detection [120](#)
 - Switch Notification [120](#)
- Network Failure, limitations [120](#)
- Network File System (NFS) [134](#)
- Networking Policies [113](#)
- Networking Policies, section
 - NIC Teaming [117](#)
 - Outbound Traffic Shaping [116](#)
 - Security Policies [114](#)
 - Traffic Shaping Policies [115](#)
- Networking Policies, types
 - Security [114](#)
 - Teaming/Failover [114](#)
 - Traffic Shaping [114](#)
- Network Traffic Shaping [115](#)
- Network Traffic Shaping, parameters
 - Average Bandwidth [115](#)
 - Burst Size [116](#)
 - Peak Bandwidth [116](#)
- NFS Datastores [165](#)
- NFS Datastores, configuring [166](#)

- NFS Datastores, terms
 - ESXi Host [166](#)
 - External Switches [169](#)
 - IP Addresses [170](#)
 - Kerberos Authentication [167](#)
 - Unmount Tasks [168](#)
 - VMkernel Binding [171](#)
- NIC Teaming [117](#)
- NIC Teaming, features
 - Failover Order [118](#)
 - Load Balancing [117](#)
 - Redundancy [118](#)
- NIC Teaming, illustrating [118](#)

O

- Open Virtualization Format (OVF) [184](#)
- Outbound Traffic Shaping [116](#)
- Outbound Traffic Shaping, terms
 - Control Mechanism [117](#)
 - Parameters [116](#)
 - Scope [117](#)
- OVF, steps [184](#)

P

- Parallel Remediation [372](#)
- Parallel Remediation, illustrating [373](#)
- Parallel Remediation, terms
 - Image-Managed Clusters [374](#)
 - View Recommended Images [374](#)
- Physical NIC Load Balancing [110](#)
- Physical Paths [143](#)
- Physical Paths, configuring [143](#)
- Port Binding [108](#)
- Port Binding, types
 - Ephemeral [109](#)
 - Static [108](#)
- Predictive DRS [304](#)
- Predictive DRS, configuring [304](#)
- Provisioning VMs [178](#)
- Provisioning VMs, methods [178](#), [179](#)
- Provisioning VMs, terms
 - Guest Operating System (OS) [183](#)
 - Open Virtualization Format (OVF) [184](#)
 - Remove Inventory [185](#)
 - Virtual Machine Wizard [179](#)

R

- Raw Device Mapping (RDM) [137](#)
- RDM, modes
 - Physical Compatibility [138](#)
 - Virtual Compatibility [138](#)
- Remove Inventory [185](#)
- Resource Allocation [274](#)
- Resource Allocation, illustrating [278](#), [279](#)
- Resource Allocation, mechanisms
 - Limits [274](#)
 - Reservation [274](#)
 - Shares [274](#)
- Resource Allocation, terms
 - CPU Reservation [276](#)
 - RAM Reservation [275](#)
 - Resource Constraints [276](#)
 - Resource Contention [277](#)

S

- SDDC, components
 - Cloud Management [22](#)
 - Physical [22](#)
 - Security [22](#)
 - Service Management [22](#)
 - Virtual Infrastructure [22](#)
- Security Policies [114](#)
- Security Policies, terms
 - Forged Transmits [114](#)
 - MAC Address [114](#)
 - Promiscuous Mode [114](#)
- Snapshot Consolidation [267](#)
- Snapshot Consolidation, illustrating [268](#), [269](#)
- Snapshots [262](#)
- Snapshots, configuring [262](#), [263](#)
- Snapshots, integrating [264](#), [265](#)
- Snapshots, options
 - Delete [266](#)
 - Delete All [266](#)
 - Edit [266](#)
 - Revert [266](#)
- Snapshots, type [263](#)
- Software-Defined Data Center (SDDC) [21](#), [22](#)
- SSO, features
 - Authentication Sources [50](#)
 - Integrated Window Authentication (IWA) [51](#)
 - Token-Based Authentication [50](#)
- SSO, illustrating [51](#)

Standard/Distributed Switches, comparing [112](#)
Standard Switches [111](#)
Storage Migration [254](#)
Storage Migration, use cases [255](#)

T

TCP/IP Stack [260](#)
Template Versioning [228](#)
Template Versions [230](#)

U

Upgrade vSphere [360](#)
Upgrade vSphere, steps [360](#)

V

VAMI, configuring [61](#), [62](#)
vCenter Architecture [49](#)
vCenter Architecture, components [49](#)
vCenter Inventory Objects [68](#)
vCenter Inventory Objects, architecture [72](#)
vCenter Inventory Objects, capabilities
 Custom Tags [76](#)
 ESXI Hosts [75](#)
 Resource Management [73](#), [74](#)
 Virtual Data Center [73](#)
 Virtual Environment [74](#)
vCenter Inventory Objects, illustrating [69-71](#)
vCenter Management Interface (VAMI) [61](#)
vCenter Server Permission [77](#)
vCenter Server Permission, configuring [77](#), [78](#)
vCenter Server Permission, tasks
 Global Permissions [84](#)
 Objects [80](#)
 Permission [80](#)
 Propagation [83](#)
 Roles [78](#)
 User Assignments [82](#)
vCenter Services [48](#)
vCenter Services, illustrating [48](#), [49](#)
vCenter Single Sign-On (SSO) [50](#)
vCLS, architecture [296](#)
VCP-DCV Certification [4](#)
vCSA [46](#)
vCSA, deploying [54](#)
vCSA, features
 Centralized Management [47](#)

- Core Software [47](#)
- Deployment Method [47](#)
- Deployment Options [47](#)
- vCSA, process
 - Configuration [56](#)
 - Open Virtualization Format (OVF) [56](#)
- vCSA, resources
 - Enhanced Linked Mode (ELM) [52](#)
 - vCenter Architecture [49](#)
 - vCenter Services [48](#)
 - vCenter Single Sign-On (SSO) [50](#)
- vCSA, stages
 - Configuration Phase [58](#)
 - Interactive UI [57](#)
 - vSphere Client [59](#)
- Virtualization [12](#)
- Virtualization, types
 - Desktop [13](#)
 - Network [13](#)
 - Server [13](#)
 - Storage [13](#)
- Virtual Local Area Networks (VLANs) [98](#)
- Virtual Machine (VM) [9](#)
- Virtual Machine Wizard [179](#)
- Virtual Machine Wizard, steps [179-182](#)
- Virtual Switches [96](#)
- Virtual Switches, features
 - Virtual Machines [96](#)
 - VMkernel Services [96](#)
- Virtual Switches, solutions
 - Network Connections [97](#)
 - Virtual Local Area Networks (VLANs) [98](#)
 - vSphere Client [100](#)
- Virtual Switches, types
 - Distributed [96](#)
 - Standard [96](#)
- Virtual Volumes [136](#)
- VLANs, illustrating [99](#)
- VM, architecture [10](#)
- VM, benefits [10](#)
- VM, components
 - Connection State [194](#)
 - CPU Performance [191](#)
 - Directory Datastore [188](#)
 - Encapsulations [188](#)
 - Guest OS [193](#)
 - Virtual Disks [192](#)
 - Virtual Hardware [189](#)

- Virtual Hardware Version [190](#)
- Virtual Storage [192](#)
- VM, configuring [11](#)
- VM Console [196](#)
- VM Console, architecture [197](#)
- VM Console, illustrating [197](#)
- VM Console, integrating [198](#)
- VM Console, methods
 - VMware Remote Console (VMRC) [197](#)
 - Web Console [197](#)
- VMFS [158](#)
- VMFS, terms
 - Datastore File Browser [159](#)
 - Datastore Maintenance [160](#)
 - Insufficient Disk [159](#)
 - Load Balancing [162](#)
 - Multipath Policies [164](#)
 - Unmounting Operations [161](#)
- VMkernel Adapter [101](#)
- VMkernel Adapter, roles [102](#)
- VMkernel Networking Layer [259](#)
- VMkernel Networking Layer, levels
 - Custom TCP/IP [259](#)
 - Default TCP/IP [259](#)
- VM Migrations [236](#)
- VM Migrations, categories
 - Cold [236](#)
 - Hot [237](#)
- VM Migrations, options
 - Compute Resource [237](#)
 - Cross vCenter Server [237](#)
 - Storage [237](#)
 - Storage Space [237](#)
- VM, resources
 - Administrators Personalize [203](#)
 - Boot Management [204](#)
 - Guest Operating System [202](#)
 - Hot-Pluggable Devices [199](#)
 - Provisioned Disks [201](#)
 - Virtual Disk Size [200](#)
- VM Template [205](#)
- VM Template, architecture [205](#)
- VM Template, sources
 - Administrators [209](#)
 - Cloning [206](#)
 - Hardware Configuration [208](#)
 - Storage Format [207](#)
- VM Template, terms

- Template Versioning [228](#)
- Template Versions [230](#)
- Version Reverting [231](#)
- VMware Tools [186](#)
- VMware Tools, benefits [186](#)
- VMware Tools, configuring [187](#)
- VMware Tools, demonstrating [380-382](#)
- VMware Tools, integrating [187](#)
- VMware vSphere [2](#)
- VMware vSphere, components
 - ESXi Hosts [2](#)
 - NSX [3](#)
 - vCenter Server [2](#)
 - vSAN [2](#)
- VMware vSphere, objectives [4](#), [5](#)
- VMware vSphere, terminologies [8](#)
- vSAN [135](#)
- vSAN, models
 - All-Flash Architecture [136](#)
 - Hybrid Architecture [135](#)
- vSphere [12](#)
- vSphere, architecture [12](#)
- vSphere Bitfusion [20](#)
- vSphere Bitfusion, features
 - GPU Sharing [21](#)
 - Network-Based GPU Access [21](#)
 - On-Demand Resource Allocation [21](#)
- vSphere Client [60](#)
- vSphere Client, capabilities [60](#)
- vSphere Client, steps [60](#)
- vSphere Cluster [286](#)
- vSphere Cluster, configuring [287](#)
- vSphere Cluster Services (vCLS) [296](#)
- vSphere Cluster, terms
 - Distributed Switches [292](#)
 - Quickstart Workflow [293](#)
 - vMotion Traffic [293](#)
- vSphere Distributed Resource Scheduler (DRS) [297](#)
- vSphere DRS, concepts
 - Automation Level [301](#)
 - Migration Threshold [302](#)
 - Monitor Tab [299](#)
 - Predictive DRS [304](#)
 - VM-Centric [298](#)
 - VM Migrations [300](#)
- vSphere DRS, terms
 - Backlog Balancing [310](#)
 - Operational States [313](#)

- Power Management [313](#)
- Swap File Location [306](#)
- VM Affinity Policies [307](#)
- VM-Host Affinity [309](#)
- VMs Automation [305](#)
- vSphere Environment [390](#)
- vSphere Environment, practices
 - Configuration [390](#)
 - Management [391](#)
 - Resilience/Recovery [391](#)
 - Standardization [391](#)
- vSphere Environment, resources
 - Clusters Management [397](#)
 - Data Breaches [399](#)
 - Fault Tolerance [402](#)
 - Lifecycle Management [398](#)
 - Virtual Infrastructure [400](#)
- vSphere Environment, terms
 - CPU/Memory Distribution [395](#)
 - Networking Forms [392](#)
 - Storage [393](#)
 - Virtual Machine [394](#)
- vSphere Events [85](#)
- vSphere Events, configuring [85](#), [86](#)
- vSphere Fault Tolerance (FT) [346](#)
- vSphere FT, configuring [347](#)
- vSphere FT, illustrating [348](#)
- vSphere HA [315](#)
- vSphere HA, approaches
 - Admission Control [338](#)
 - Cluster Resource Percentage [339](#)
 - Cluster Status [345](#)
 - Configuration Requirements [330](#)
 - CPU Reservations [340](#)
 - Failures/Response [332](#)
 - Heatbeat Datastores [342](#)
 - Networks Configuration [344](#)
 - Orchestrated Restart [335](#)
 - Performance Degradation [341](#)
 - Restart Priority [333](#)
 - VM Monitoring [337](#)
- vSphere Client [331](#)
- vSphere HA, benefits
 - Disaster Recovery [316](#)
 - Downtime Recovery [316](#)
 - Hardware-Independent Protection [316](#)
 - Planned Downtime [316](#)
 - Recovery Integration [317](#)

- vSphere HA, communications
 - Datastore Heartbeats [325](#)
 - Network Heartbeats [324](#)
- vSphere HA, configuring [315](#)
- vSphere HA, illustrating [319](#)
- vSphere HA, integrating [318](#), [319](#)
- vSphere HA, scenarios
 - Failed Primary Hosts [327](#)
 - Failed Secondary Hosts [326](#)
 - Isolated Hosts [328](#)
 - vCenter Region [329](#)
 - VM Storage [329](#)
- vSphere HA, terms
 - Additional Networks [322](#)
 - Network Isolation [320](#)
 - NIC Teaming [321](#)
- vSphere, interface
 - PowerCLI [26](#)
 - VMware Host Client [26](#)
 - vSphere Client [26](#)
- vSphere License Service [64](#)
- vSphere License Service, architecture [65](#)
- vSphere License Service, configuring [64](#), [65](#)
- vSphere License Service, ensuring [66](#), [67](#)
- vSphere Lifecycle Manager [361](#)
- vSphere Lifecycle Manager, configuring [361](#)
- vSphere, resources
 - CPU Utilization [16](#)
 - Datastores [19](#)
 - ESXi Host [13](#)
 - High-Performance Storage [18](#)
 - Memory Segment [16](#)
 - Physical Architecture [14](#)
 - Resource Sharing [15](#)
 - Virtualised Hosts [17](#)
 - Virtual Networking [17](#), [18](#)
- vSphere Storage [126](#)
- vSphere Storage, architecture [127](#)
- vSphere Storage, configuring [130](#)
- vSphere Storage, illustrating [128](#), [129](#)
- vSphere Storage, technologies
 - Direct Attached Storage (DAS) [127](#)
 - FC [127](#)
 - FC over Ethernet [127](#)
 - iSCSI [127](#)
 - Network Attached Storage (NAS) [127](#)
- vSphere Storage vMotion [251](#), [252](#)
- vSphere Storage vMotion, configuring [253](#)

- vSphere Storage vMotion, guidelines [253](#)
- vSphere Storage vMotion, limitations [254](#)
- vSphere Storage vMotion, steps
 - Data Copy [252](#)
 - Final Transition [252](#)
 - Initiating Migration [252](#)
 - I/O Mirroring [252](#)
 - New Process [252](#)
- vSphere vMotion [238](#)
- vSphere vMotion, configuring [239](#)
- vSphere vMotion, illustrating [242](#), [243](#)
- vSphere vMotion, practices [240](#)
- vSphere vMotion, prerequisites [241](#)
- vSphere vMotion, sources [242](#)
- vSphere vMotion, steps [240](#), [241](#)