

vSphere Storage

Update 2

VMware vSphere 6.7

VMware ESXi 6.7

vCenter Server 6.7

You can find the most up-to-date technical documentation on the VMware by Broadcom website at:

<https://docs.vmware.com/>

VMware by Broadcom
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Copyright © 2009-2023 Broadcom. All Rights Reserved. The term “Broadcom” refers to Broadcom Inc. and/or its subsidiaries. For more information, go to <https://www.broadcom.com>. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Contents

| | |
|---|-----------|
| About vSphere Storage | 14 |
| Updated Information | 15 |
| 1 Introduction to Storage | 16 |
| Traditional Storage Virtualization Models | 16 |
| Software-Defined Storage Models | 18 |
| vSphere Storage APIs | 19 |
| 2 Getting Started with a Traditional Storage Model | 20 |
| Types of Physical Storage | 20 |
| Local Storage | 20 |
| Networked Storage | 21 |
| Target and Device Representations | 25 |
| How Virtual Machines Access Storage | 26 |
| Storage Device Characteristics | 27 |
| Comparing Types of Storage | 30 |
| Supported Storage Adapters | 31 |
| View Storage Adapters Information | 31 |
| Datastore Characteristics | 32 |
| Display Datastore Information | 34 |
| Using Persistent Memory | 35 |
| Monitor PMem Datastore Statistics | 37 |
| 3 Overview of Using ESXi with a SAN | 39 |
| ESXi and SAN Use Cases | 40 |
| Specifics of Using SAN Storage with ESXi | 41 |
| ESXi Hosts and Multiple Storage Arrays | 41 |
| Making LUN Decisions | 41 |
| Use the Predictive Scheme to Make LUN Decisions | 42 |
| Use the Adaptive Scheme to Make LUN Decisions | 42 |
| Selecting Virtual Machine Locations | 43 |
| Third-Party Management Applications | 44 |
| SAN Storage Backup Considerations | 44 |
| Using Third-Party Backup Packages | 45 |
| 4 Using ESXi with Fibre Channel SAN | 46 |
| Fibre Channel SAN Concepts | 46 |

| | |
|---|-----------|
| Ports in Fibre Channel SAN | 47 |
| Fibre Channel Storage Array Types | 47 |
| Using Zoning with Fibre Channel SANs | 48 |
| How Virtual Machines Access Data on a Fibre Channel SAN | 48 |
| 5 Configuring Fibre Channel Storage | 50 |
| ESXi Fibre Channel SAN Requirements | 50 |
| ESXi Fibre Channel SAN Restrictions | 51 |
| Setting LUN Allocations | 51 |
| Setting Fibre Channel HBAs | 51 |
| Installation and Setup Steps | 52 |
| N-Port ID Virtualization | 52 |
| How NPIV-Based LUN Access Works | 53 |
| Requirements for Using NPIV | 53 |
| NPIV Capabilities and Limitations | 54 |
| Configure or Modify WWN Assignments | 54 |
| 6 Configuring Fibre Channel over Ethernet | 56 |
| Fibre Channel over Ethernet Adapters | 56 |
| Configuration Guidelines for Software FCoE | 57 |
| Set Up Networking for Software FCoE | 58 |
| Add Software FCoE Adapters | 59 |
| 7 Booting ESXi from Fibre Channel SAN | 60 |
| Boot from SAN Benefits | 60 |
| Requirements and Considerations when Booting from Fibre Channel SAN | 61 |
| Getting Ready for Boot from SAN | 61 |
| Configure SAN Components and Storage System | 62 |
| Configure Storage Adapter to Boot from SAN | 62 |
| Set Up Your System to Boot from Installation Media | 63 |
| Configure Emulex HBA to Boot from SAN | 63 |
| Enable the BootBIOS Prompt | 63 |
| Enable the BIOS | 64 |
| Configure QLogic HBA to Boot from SAN | 64 |
| 8 Booting ESXi with Software FCoE | 67 |
| Requirements and Considerations for Software FCoE Boot | 67 |
| Set Up Software FCoE Boot | 68 |
| Configure Software FCoE Boot Parameters | 69 |
| Install and Boot ESXi from Software FCoE LUN | 69 |
| Troubleshooting Boot from Software FCoE for an ESXi Host | 70 |

9 Best Practices for Fibre Channel Storage 71

- Preventing Fibre Channel SAN Problems 71
- Disable Automatic Host Registration 72
- Optimizing Fibre Channel SAN Storage Performance 72
 - Storage Array Performance 73
 - Server Performance with Fibre Channel 73

10 Using ESXi with iSCSI SAN 75

- About iSCSI SAN 75
 - iSCSI Multipathing 76
 - Nodes and Ports in the iSCSI SAN 77
 - iSCSI Naming Conventions 77
 - iSCSI Initiators 78
 - About the VMware iSER Adapter 79
 - Establishing iSCSI Connections 79
 - iSCSI Storage System Types 80
 - Discovery, Authentication, and Access Control 81
 - Error Correction 82
 - How Virtual Machines Access Data on an iSCSI SAN 82

11 Configuring iSCSI Adapters and Storage 84

- ESXi iSCSI SAN Recommendations and Restrictions 85
- Configuring iSCSI Parameters for Adapters 85
- Set Up Independent Hardware iSCSI Adapters 87
 - View Independent Hardware iSCSI Adapters 87
 - Edit Network Settings for Hardware iSCSI 88
- Configure Dependent Hardware iSCSI Adapters 89
 - Dependent Hardware iSCSI Considerations 90
 - View Dependent Hardware iSCSI Adapters 90
 - Determine Association Between iSCSI and Network Adapters 91
- Configure the Software iSCSI Adapter 92
 - Activate or Disable the Software iSCSI Adapter 92
- Configure iSER Adapters 93
 - Enable the VMware iSER Adapter 94
- Modify General Properties for iSCSI or iSER Adapters 95
- Setting Up Network for iSCSI and iSER 95
 - Multiple Network Adapters in iSCSI or iSER Configuration 96
 - Best Practices for Configuring Networking with Software iSCSI 98
 - Configure Port Binding for iSCSI or iSER 101
 - Managing iSCSI Network 106
 - iSCSI Network Troubleshooting 107

| | |
|---|------------|
| Using Jumbo Frames with iSCSI | 107 |
| Enable Jumbo Frames for Networking | 108 |
| Enable Jumbo Frames for Independent Hardware iSCSI | 108 |
| Configuring Discovery Addresses for iSCSI Adapters | 109 |
| Configure Dynamic or Static Discovery for iSCSI and iSER | 109 |
| Remove Dynamic or Static iSCSI Targets | 110 |
| Configuring CHAP Parameters for iSCSI Adapters | 110 |
| Selecting CHAP Authentication Method | 111 |
| Set Up CHAP for iSCSI or iSER Adapter | 112 |
| Set Up CHAP for Target | 113 |
| Configuring Advanced Parameters for iSCSI | 114 |
| Configure Advanced Parameters for iSCSI | 116 |
| iSCSI Session Management | 116 |
| Review iSCSI Sessions | 117 |
| Add iSCSI Sessions | 117 |
| Remove iSCSI Sessions | 118 |
| 12 Booting from iSCSI SAN | 119 |
| General Recommendations for Boot from iSCSI SAN | 119 |
| Prepare the iSCSI SAN | 120 |
| Configure Independent Hardware iSCSI Adapter for SAN Boot | 121 |
| Configure iSCSI Boot Settings | 121 |
| iBFT iSCSI Boot Overview | 122 |
| iBFT iSCSI Boot Considerations | 123 |
| Configuring iBFT Boot from SAN | 123 |
| Networking Best Practices | 126 |
| Change iBFT iSCSI Boot Settings | 126 |
| Troubleshooting iBFT iSCSI Boot | 127 |
| 13 Best Practices for iSCSI Storage | 129 |
| Preventing iSCSI SAN Problems | 129 |
| Optimizing iSCSI SAN Storage Performance | 130 |
| Storage System Performance | 130 |
| Server Performance with iSCSI | 131 |
| Network Performance | 131 |
| Checking Ethernet Switch Statistics | 134 |
| 14 Managing Storage Devices | 135 |
| Storage Device Characteristics | 135 |
| Display Storage Devices for a Host | 136 |
| Display Storage Devices for an Adapter | 137 |

| | |
|---|------------|
| Device Sector Formats | 138 |
| Storage Device Names and Identifiers | 140 |
| NVMe Devices with NGUID Device Identifiers | 141 |
| Upgrade Stateless ESXi Hosts with NGUID-only NVMe Devices to 6.7 Update 2 | 142 |
| Rename Storage Devices | 144 |
| Storage Rescan Operations | 145 |
| Perform Storage Rescan | 145 |
| Perform Adapter Rescan | 146 |
| Change the Number of Scanned Storage Devices | 146 |
| Identifying Device Connectivity Problems | 147 |
| Detecting PDL Conditions | 147 |
| Performing Planned Storage Device Removal | 148 |
| Recovering from PDL Conditions | 150 |
| Handling Transient APD Conditions | 150 |
| Verify the Connection Status of a Storage Device | 152 |
| Device Connectivity Problems and High Availability | 153 |
| Enable or Disable the Locator LED on Storage Devices | 153 |
| Erase Storage Devices | 153 |
| 15 Working with Flash Devices | 155 |
| Using Flash Devices with ESXi | 155 |
| Identifying Flash Virtual Disks | 156 |
| Marking Storage Devices | 156 |
| Mark Storage Devices as Flash | 157 |
| Mark Storage Devices as Local | 157 |
| Monitor Flash Devices | 158 |
| Best Practices for Flash Devices | 158 |
| Estimate Lifetime of Flash Devices | 158 |
| About Virtual Flash Resource | 159 |
| Considerations for Virtual Flash Resource | 160 |
| Set Up Virtual Flash Resource | 160 |
| Remove Virtual Flash Resource | 161 |
| Virtual Flash Advanced Settings | 162 |
| Configuring Host Swap Cache | 162 |
| Configure Host Cache with VMFS Datastore | 163 |
| Configure Host Swap Cache with Virtual Flash Resource | 163 |
| 16 About VMware vSphere Flash Read Cache | 165 |
| DRS Support for Flash Read Cache | 166 |
| vSphere High Availability Support for Flash Read Cache | 166 |
| Configure Flash Read Cache for a Virtual Machine | 166 |

Migrate Virtual Machines with Flash Read Cache 167

17 Working with Datastores 169

Types of Datastores 169

Understanding VMFS Datastores 170

Versions of VMFS Datastores 171

VMFS Datastores as Repositories 173

Sharing a VMFS Datastore Across Hosts 173

VMFS Metadata Updates 174

VMFS Locking Mechanisms 175

Snapshot Formats on VMFS 179

Upgrading VMFS Datastores 180

Resolve VMFS3 Upgrade Failures 181

Upgrade a VMFS3 Datastore to VMFS5 181

Understanding Network File System Datastores 182

NFS Protocols and ESXi 182

NFS Storage Guidelines and Requirements 184

Firewall Configurations for NFS Storage 187

Using Layer 3 Routed Connections to Access NFS Storage 189

Using Kerberos for NFS 4.1 189

Set Up NFS Storage Environment 190

Configure ESXi Hosts for Kerberos Authentication 191

Creating Datastores 194

Create a VMFS Datastore 194

Create an NFS Datastore 195

Create a Virtual Volumes Datastore 196

Managing Duplicate VMFS Datastores 197

Mount a VMFS Datastore Copy 198

Increase VMFS Datastore Capacity 199

Administrative Operations for Datastores 201

Change Datastore Name 201

Unmount Datastores 202

Mount Datastores 203

Remove VMFS Datastores 203

Use Datastore Browser 204

Turn Off Storage Filters 208

Set Up Dynamic Disk Mirroring 209

Collecting Diagnostic Information for ESXi Hosts on a Storage Device 210

Set Up a Device Partition as Core Dump Location 211

Set Up a File as Core Dump Location 212

Checking Metadata Consistency with VOMA 214

| | |
|--|------------|
| Use VOMA to Check Metadata Consistency | 217 |
| Configuring VMFS Pointer Block Cache | 217 |
| Obtain Information for VMFS Pointer Block Cache | 218 |
| Change the Size of the Pointer Block Cache | 219 |
| 18 Understanding Multipathing and Failover | 221 |
| Failovers with Fibre Channel | 221 |
| Host-Based Failover with iSCSI | 222 |
| Array-Based Failover with iSCSI | 224 |
| Path Failover and Virtual Machines | 225 |
| Set Timeout on Windows Guest OS | 225 |
| Pluggable Storage Architecture and Path Management | 226 |
| About Pluggable Storage Architecture | 227 |
| VMware Native Multipathing Plug-In | 228 |
| Path Selection Plug-Ins and Policies | 231 |
| VMware SATPs | 232 |
| VMware High Performance Plug-In | 234 |
| Viewing and Managing Paths | 237 |
| View Storage Device Paths | 238 |
| View Datastore Paths | 238 |
| Change the Path Selection Policy | 239 |
| Change Default Parameters for Latency Round Robin | 240 |
| Disable Storage Paths | 240 |
| Using Claim Rules | 241 |
| Multipathing Considerations | 242 |
| List Multipathing Claim Rules for the Host | 242 |
| Add Multipathing Claim Rules | 244 |
| Delete Multipathing Claim Rules | 247 |
| Mask Paths | 248 |
| Unmask Paths | 249 |
| Define NMP SATP Rules | 250 |
| Scheduling Queues for Virtual Machine I/Os | 252 |
| Edit Per File I/O Scheduling | 252 |
| Use esxcli Commands to Enable or Disable Per File I/O Scheduling | 252 |
| 19 Raw Device Mapping | 254 |
| About Raw Device Mapping | 254 |
| Benefits of Raw Device Mapping | 255 |
| RDM Considerations and Limitations | 258 |
| Raw Device Mapping Characteristics | 258 |
| RDM Virtual and Physical Compatibility Modes | 258 |

- Dynamic Name Resolution 259
- Raw Device Mapping with Virtual Machine Clusters 259
- Comparing Available SCSI Device Access Modes 260
- Create Virtual Machines with RDMs 260
- Manage Paths for a Mapped LUN 262
- Virtual Machines with RDMs Must Ignore SCSI INQUIRY Cache 263

20 Storage Policy Based Management 265

- Virtual Machine Storage Policies 266
- Workflow for Virtual Machine Storage Policies 266
- Populating the VM Storage Policies Interface 267
 - Use Storage Providers to Populate the VM Storage Policies Interface 268
 - Assign Tags to Datastores 269
- About Rules and Rule Sets 271
- Creating and Managing VM Storage Policies 273
 - Define a VM Storage Policy in the vSphere Web Client 273
 - Create a VM Storage Policy for Host-Based Data Services 278
 - Create a VM Storage Policy for Virtual Volumes 279
 - Create a VM Storage Policy for Tag-Based Placement 281
 - Edit or Clone a VM Storage Policy 282
- About Storage Policy Components 283
 - Create Storage Policy Components 284
 - Edit or Clone Storage Policy Components 285
- Storage Policies and Virtual Machines 286
 - Assign Storage Policies to Virtual Machines 287
 - Change Storage Policy Assignment for Virtual Machine Files and Disks 288
 - Check Compliance for a VM Storage Policy 290
 - Find Compatible Storage Resource for Noncompliant Virtual Machine 290
 - Reapply Virtual Machine Storage Policy 291
- Default Storage Policies 292
 - Change the Default Storage Policy for a Datastore 293

21 Using Storage Providers 294

- About Storage Providers 294
- Storage Providers and Data Representation 295
- Storage Provider Requirements and Considerations 296
- Register Storage Providers 297
- View Storage Provider Information 298
- Manage Storage Providers 298

22 Working with Virtual Volumes 300

| | |
|---|------------|
| About Virtual Volumes | 300 |
| Virtual Volumes Concepts | 301 |
| Virtual Volumes | 302 |
| Virtual Volumes and Storage Providers | 304 |
| Storage Containers | 304 |
| Protocol Endpoints | 305 |
| Binding and Unbinding Virtual Volumes to Protocol Endpoints | 306 |
| Virtual Volumes Datastores | 306 |
| Virtual Volumes and VM Storage Policies | 307 |
| Virtual Volumes and Storage Protocols | 307 |
| Virtual Volumes Architecture | 308 |
| Virtual Volumes and VMware Certificate Authority | 310 |
| Snapshots and Virtual Volumes | 311 |
| Before You Enable Virtual Volumes | 311 |
| Synchronize vSphere Storage Environment with a Network Time Server | 312 |
| Configure Virtual Volumes | 313 |
| Register Storage Providers for Virtual Volumes | 314 |
| Create a Virtual Volumes Datastore | 315 |
| Review and Manage Protocol Endpoints | 315 |
| Change the Path Selection Policy for a Protocol Endpoint | 316 |
| Provision Virtual Machines on Virtual Volumes Datastores | 316 |
| Virtual Volumes and Replication | 317 |
| Requirements for Replication with Virtual Volumes | 318 |
| Virtual Volumes and Replication Groups | 319 |
| Virtual Volumes and Fault Domains | 319 |
| Virtual Volumes Replication Workflow | 321 |
| Replication Guidelines and Considerations | 321 |
| Best Practices for Working with vSphere Virtual Volumes | 322 |
| Guidelines and Limitations in Using vSphere Virtual Volumes | 323 |
| Best Practices for Storage Container Provisioning | 324 |
| Best Practices for vSphere Virtual Volumes Performance | 325 |
| Troubleshooting Virtual Volumes | 326 |
| Virtual Volumes and esxcli Commands | 326 |
| Virtual Datastore Is Inaccessible | 327 |
| Failures When Migrating VMs or Deploying VM OVFes to Virtual Volumes Datastores | 327 |
| Failed Attempts to Migrate VMs with Memory Snapshots to and from Virtual Datastores | 328 |
| 23 Filtering Virtual Machine I/O | 330 |
| About I/O Filters | 330 |
| Types of I/O Filters | 331 |
| I/O Filtering Components | 331 |

| | |
|--|------------|
| Storage Providers for I/O Filters | 333 |
| Using Flash Storage Devices with Cache I/O Filters | 333 |
| System Requirements for I/O Filters | 334 |
| Configure I/O Filters in the vSphere Environment | 335 |
| Install I/O Filters in a Cluster | 335 |
| View I/O Filters and Storage Providers | 336 |
| Configure Virtual Flash Resource for Caching I/O Filters | 337 |
| Enable I/O Filter Data Services on Virtual Disks | 337 |
| Assign the I/O Filter Policy to Virtual Machines | 338 |
| Managing I/O Filters | 339 |
| Uninstall I/O Filters from a Cluster | 339 |
| Upgrade I/O Filters in a Cluster | 340 |
| I/O Filter Guidelines and Best Practices | 340 |
| Migrating Virtual Machines with I/O Filters | 341 |
| Handling I/O Filter Installation Failures | 342 |
| Install I/O Filters on a Single ESXi Host | 342 |
| 24 Storage Hardware Acceleration | 343 |
| Hardware Acceleration Benefits | 343 |
| Hardware Acceleration Requirements | 344 |
| Hardware Acceleration Support Status | 344 |
| Hardware Acceleration for Block Storage Devices | 344 |
| Disable Hardware Acceleration for Block Storage Devices | 345 |
| Managing Hardware Acceleration on Block Storage Devices | 345 |
| Hardware Acceleration on NAS Devices | 351 |
| Install NAS Plug-In | 352 |
| Uninstall NAS Plug-Ins | 353 |
| Update NAS Plug-Ins | 353 |
| Verify Hardware Acceleration Status for NAS | 354 |
| Hardware Acceleration Considerations | 354 |
| 25 Storage Provisioning and Space Reclamation | 355 |
| Virtual Disk Thin Provisioning | 355 |
| About Virtual Disk Provisioning Policies | 356 |
| Create Thin Provisioned Virtual Disks | 357 |
| View Virtual Machine Storage Resources | 358 |
| Determine the Disk Format of a Virtual Machine | 358 |
| Inflate Thin Virtual Disks | 359 |
| Handling Datastore Over-Subscription | 359 |
| ESXi and Array Thin Provisioning | 360 |
| Monitoring Space Use | 361 |

| | |
|---|------------|
| Identify Thin-Provisioned Storage Devices | 361 |
| Storage Space Reclamation | 362 |
| Space Reclamation Requests from VMFS Datastores | 364 |
| Space Reclamation Requests from Guest Operating Systems | 369 |
| 26 Using vmkfstools | 371 |
| vmkfstools Command Syntax | 371 |
| The vmkfstools Command Options | 372 |
| -v Suboption | 372 |
| File System Options | 373 |
| Virtual Disk Options | 375 |
| Storage Device Options | 382 |

About vSphere Storage

vSphere Storage describes virtualized and software-defined storage technologies that VMware ESXi™ and VMware vCenter Server® offer, and explains how to configure and use these technologies.

Intended Audience

This information is for experienced system administrators who are familiar with the virtual machine and storage virtualization technologies, data center operations, and SAN storage concepts.

vSphere Client and vSphere Web Client

Instructions in this guide reflect the vSphere Client (an HTML5-based GUI). You can also use the instructions to perform the tasks by using the vSphere Web Client (a Flex-based GUI).

Tasks for which the workflow differs significantly between the vSphere Client and the vSphere Web Client have duplicate procedures that provide steps according to the respective client interface. The procedures that relate to the vSphere Web Client, contain vSphere Web Client in the title.

Updated Information

This *vSphere Storage* is updated with each release of the product or when necessary.

This table provides the update history of the *vSphere Storage*.

| Revision | Description |
|-------------|--|
| 28 JUN 2023 | Updated statements about the SCSI <code>TimeoutValue</code> parameter in ESXi Fibre Channel SAN Requirements and ESXi iSCSI SAN Recommendations and Restrictions . |
| 31 JAN 2023 | Clarified information about replication groups in the following topics: <ul style="list-style-type: none">■ Virtual Volumes and Replication Groups■ Virtual Volumes and Fault Domains |
| 19 APR 2022 | Minor revisions. |
| 25 JUN 2021 | Updated Best Practices for Configuring Networking with Software iSCSI to clarify that unbound VMkernel ports are not used for software iSCSI traffic. |
| 04 JAN 2021 | Setting Fibre Channel HBAs has been updated to indicate that ESXi supports 32 Gbps Fibre Channel. |
| 15 AUG 2020 | At VMware, we value inclusion. To foster this principle within our customer, partner, and internal community, we are replacing some of the terminology in our content. We have updated this guide to remove instances of non-inclusive language. |
| 27 JUN 2019 | Minor revisions. |
| 09 MAY 2019 | Minor revisions. |
| 11 APR 2019 | Initial release. |

Introduction to Storage

1

vSphere supports various storage options and functionalities in traditional and software-defined storage environments. A high-level overview of vSphere storage elements and aspects helps you plan a proper storage strategy for your virtual data center.

Read the following topics next:

- [Traditional Storage Virtualization Models](#)
- [Software-Defined Storage Models](#)
- [vSphere Storage APIs](#)

Traditional Storage Virtualization Models

Generally, storage virtualization refers to a logical abstraction of physical storage resources and capacities from virtual machines and their applications. ESXi provides host-level storage virtualization.

In vSphere environment, a traditional model is built around the following storage technologies and ESXi and vCenter Server virtualization functionalities.

Local and Networked Storage

In traditional storage environments, the ESXi storage management process starts with storage space that your storage administrator preallocates on different storage systems. ESXi supports local storage and networked storage.

See [Types of Physical Storage](#).

Storage Area Networks

A storage area network (SAN) is a specialized high-speed network that connects computer systems, or ESXi hosts, to high-performance storage systems. ESXi can use Fibre Channel or iSCSI protocols to connect to storage systems.

See [Chapter 3 Overview of Using ESXi with a SAN](#).

Fibre Channel

Fibre Channel (FC) is a storage protocol that the SAN uses to transfer data traffic from ESXi host servers to shared storage. The protocol packages SCSI commands into FC frames. To connect to the FC SAN, your host uses Fibre Channel host bus adapters (HBAs).

See [Chapter 4 Using ESXi with Fibre Channel SAN](#).

Internet SCSI

Internet iSCSI (iSCSI) is a SAN transport that can use Ethernet connections between computer systems, or ESXi hosts, and high-performance storage systems. To connect to the storage systems, your hosts use hardware iSCSI adapters or software iSCSI initiators with standard network adapters.

See [Chapter 10 Using ESXi with iSCSI SAN](#).

Storage Device or LUN

In the ESXi context, the terms device and LUN are used interchangeably. Typically, both terms mean a storage volume that is presented to the host from a block storage system and is available for formatting.

See [Target and Device Representations](#) and [Chapter 14 Managing Storage Devices](#).

Virtual Disks

A virtual machine on an ESXi host uses a virtual disk to store its operating system, application files, and other data associated with its activities. Virtual disks are large physical files, or sets of files, that can be copied, moved, archived, and backed up as any other files. You can configure virtual machines with multiple virtual disks.

To access virtual disks, a virtual machine uses virtual SCSI controllers. These virtual controllers include BusLogic Parallel, LSI Logic Parallel, LSI Logic SAS, and VMware Paravirtual. These controllers are the only types of SCSI controllers that a virtual machine can see and access.

Each virtual disk resides on a datastore that is deployed on physical storage. From the standpoint of the virtual machine, each virtual disk appears as if it were a SCSI drive connected to a SCSI controller. Whether the physical storage is accessed through storage or network adapters on the host is typically transparent to the VM guest operating system and applications.

VMware vSphere® VMFS

The datastores that you deploy on block storage devices use the native vSphere Virtual Machine File System (VMFS) format. It is a special high-performance file system format that is optimized for storing virtual machines.

See [Understanding VMFS Datastores](#).

NFS

An NFS client built into ESXi uses the Network File System (NFS) protocol over TCP/IP to access an NFS volume that is located on a NAS server. The ESXi host can mount the volume and use it as an NFS datastore.

See [Understanding Network File System Datastores](#).

Raw Device Mapping

In addition to virtual disks, vSphere offers a mechanism called raw device mapping (RDM). RDM is useful when a guest operating system inside a virtual machine requires direct access to a storage device. For information about RDMs, see [Chapter 19 Raw Device Mapping](#).

Software-Defined Storage Models

In addition to abstracting underlying storage capacities from VMs, as traditional storage models do, software-defined storage abstracts storage capabilities.

With the software-defined storage model, a virtual machine becomes a unit of storage provisioning and can be managed through a flexible policy-based mechanism. The model involves the following vSphere technologies.

Storage Policy Based Management

Storage Policy Based Management (SPBM) is a framework that provides a single control panel across various data services and storage solutions, including vSAN and Virtual Volumes. Using storage policies, the framework aligns application demands of your virtual machines with capabilities provided by storage entities.

See [Chapter 20 Storage Policy Based Management](#).

VMware vSphere® Virtual Volumes™

The Virtual Volumes functionality changes the storage management paradigm from managing space inside datastores to managing abstract storage objects handled by storage arrays. With Virtual Volumes, an individual virtual machine, not the datastore, becomes a unit of storage management. And storage hardware gains complete control over virtual disk content, layout, and management.

See [Chapter 22 Working with Virtual Volumes](#).

VMware vSAN

vSAN is a distributed layer of software that runs natively as a part of the hypervisor. vSAN aggregates local or direct-attached capacity devices of an ESXi host cluster and creates a single storage pool shared across all hosts in the vSAN cluster.

See *Administering VMware vSAN*.

I/O Filters

I/O filters are software components that can be installed on ESXi hosts and can offer additional data services to virtual machines. Depending on implementation, the services might include replication, encryption, caching, and so on.

See [Chapter 23 Filtering Virtual Machine I/O](#).

vSphere Storage APIs

Storage APIs is a family of APIs used by third-party hardware, software, and storage providers to develop components that enhance several vSphere features and solutions.

This Storage publication describes several Storage APIs that contribute to your storage environment. For information about other APIs from this family, including vSphere APIs - Data Protection, see the VMware website.

vSphere APIs for Storage Awareness

Also known as VASA, these APIs, either supplied by third-party vendors or offered by VMware, enable communications between vCenter Server and underlying storage. Through VASA, storage entities can inform vCenter Server about their configurations, capabilities, and storage health and events. In return, VASA can deliver VM storage requirements from vCenter Server to a storage entity and ensure that the storage layer meets the requirements.

VASA becomes essential when you work with Virtual Volumes, vSAN, vSphere APIs for I/O Filtering (VAIO), and storage VM policies. See [Chapter 21 Using Storage Providers](#).

vSphere APIs for Array Integration

These APIs, also known as VAAI, include the following components:

- Hardware Acceleration APIs. Help arrays to integrate with vSphere, so that vSphere can offload certain storage operations to the array. This integration significantly reduces CPU overhead on the host. See [Chapter 24 Storage Hardware Acceleration](#).
- Array Thin Provisioning APIs. Help to monitor space use on thin-provisioned storage arrays to prevent out-of-space conditions, and to perform space reclamation. See [ESXi and Array Thin Provisioning](#).

vSphere APIs for Multipathing

Known as the Pluggable Storage Architecture (PSA), these APIs allow storage partners to create and deliver multipathing and load-balancing plug-ins that are optimized for each array. Plug-ins communicate with storage arrays and determine the best path selection strategy to increase I/O performance and reliability from the ESXi host to the storage array. For more information, see [Pluggable Storage Architecture and Path Management](#).

Getting Started with a Traditional Storage Model

2

Setting up your ESXi storage in traditional environments, includes configuring your storage systems and devices, enabling storage adapters, and creating datastores.

Read the following topics next:

- [Types of Physical Storage](#)
- [Supported Storage Adapters](#)
- [Datastore Characteristics](#)
- [Using Persistent Memory](#)

Types of Physical Storage

In traditional storage environments, the ESXi storage management process starts with storage space that your storage administrator preallocates on different storage systems. ESXi supports local storage and networked storage.

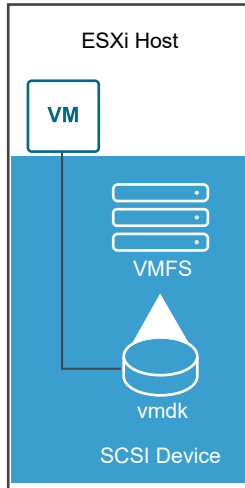
Local Storage

Local storage can be internal hard disks located inside your ESXi host. It can also include external storage systems located outside and connected to the host directly through protocols such as SAS or SATA.

Local storage does not require a storage network to communicate with your host. You need a cable connected to the storage unit and, when required, a compatible HBA in your host.

The following illustration depicts a virtual machine using local SCSI storage.

Figure 2-1. Local Storage



In this example of a local storage topology, the ESXi host uses a single connection to a storage device. On that device, you can create a VMFS datastore, which you use to store virtual machine disk files.

Although this storage configuration is possible, it is not a best practice. Using single connections between storage devices and hosts creates single points of failure (SPOF) that can cause interruptions when a connection becomes unreliable or fails. However, because most of local storage devices do not support multiple connections, you cannot use multiple paths to access local storage.

ESXi supports various local storage devices, including SCSI, IDE, SATA, USB, SAS, flash, and NVMe devices.

Note You cannot use IDE/ATA or USB drives to store virtual machines.

Local storage does not support sharing across multiple hosts. Only one host has access to a datastore on a local storage device. As a result, although you can use local storage to create VMs, you cannot use VMware features that require shared storage, such as HA and vMotion.

However, if you use a cluster of hosts that have just local storage devices, you can implement vSAN. vSAN transforms local storage resources into software-defined shared storage. With vSAN, you can use features that require shared storage. For details, see the *Administering VMware vSAN* documentation.

Networked Storage

Networked storage consists of external storage systems that your ESXi host uses to store virtual machine files remotely. Typically, the host accesses these systems over a high-speed storage network.

Networked storage devices are shared. Datastores on networked storage devices can be accessed by multiple hosts concurrently. ESXi supports multiple networked storage technologies.

In addition to traditional networked storage that this topic covers, VMware supports virtualized shared storage, such as vSAN. vSAN transforms internal storage resources of your ESXi hosts into shared storage that provides such capabilities as High Availability and vMotion for virtual machines. For details, see the *Administering VMware vSAN* documentation.

Note The same LUN cannot be presented to an ESXi host or multiple hosts through different storage protocols. To access the LUN, hosts must always use a single protocol, for example, either Fibre Channel only or iSCSI only.

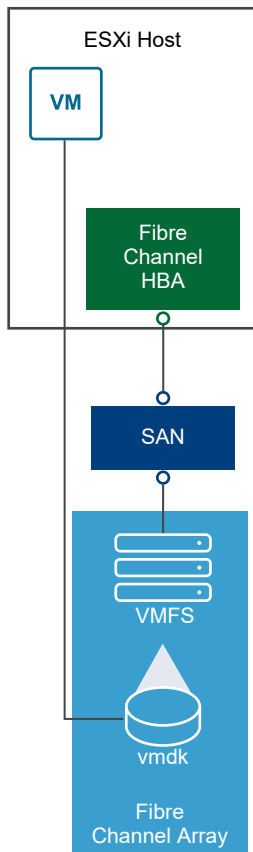
Fibre Channel (FC)

Stores virtual machine files remotely on an FC storage area network (SAN). FC SAN is a specialized high-speed network that connects your hosts to high-performance storage devices. The network uses Fibre Channel protocol to transport SCSI traffic from virtual machines to the FC SAN devices.

To connect to the FC SAN, your host should be equipped with Fibre Channel host bus adapters (HBAs). Unless you use Fibre Channel direct connect storage, you need Fibre Channel switches to route storage traffic. If your host contains FCoE (Fibre Channel over Ethernet) adapters, you can connect to your shared Fibre Channel devices by using an Ethernet network.

Fibre Channel Storage depicts virtual machines using Fibre Channel storage.

Figure 2-2. Fibre Channel Storage



In this configuration, a host connects to a SAN fabric, which consists of Fibre Channel switches and storage arrays, using a Fibre Channel adapter. LUNs from a storage array become available to the host. You can access the LUNs and create datastores for your storage needs. The datastores use the VMFS format.

For specific information on setting up the Fibre Channel SAN, see [Chapter 4 Using ESXi with Fibre Channel SAN](#).

Internet SCSI (iSCSI)

Stores virtual machine files on remote iSCSI storage devices. iSCSI packages SCSI storage traffic into the TCP/IP protocol, so that it can travel through standard TCP/IP networks instead of the specialized FC network. With an iSCSI connection, your host serves as the initiator that communicates with a target, located in remote iSCSI storage systems.

ESXi offers the following types of iSCSI connections:

Hardware iSCSI

Your host connects to storage through a third-party adapter capable of offloading the iSCSI and network processing. Hardware adapters can be dependent and independent.

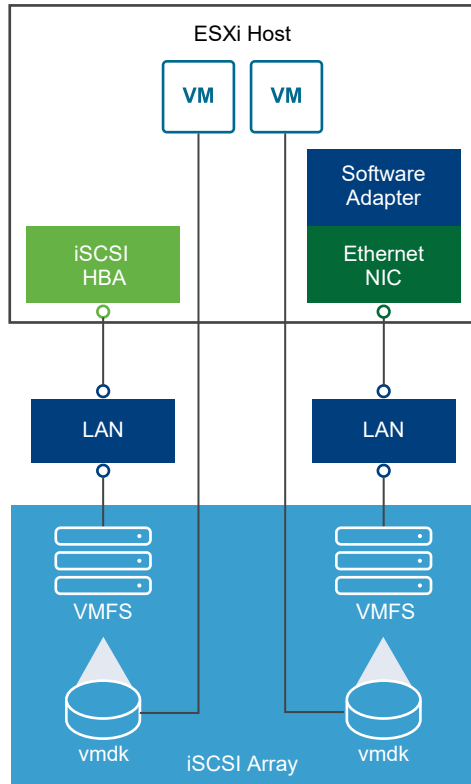
Software iSCSI

Your host uses a software-based iSCSI initiator in the VMkernel to connect to storage. With this type of iSCSI connection, your host needs only a standard network adapter for network connectivity.

You must configure iSCSI initiators for the host to access and display iSCSI storage devices.

iSCSI Storage depicts different types of iSCSI initiators.

Figure 2-3. iSCSI Storage



In the left example, the host uses the hardware iSCSI adapter to connect to the iSCSI storage system.

In the right example, the host uses a software iSCSI adapter and an Ethernet NIC to connect to the iSCSI storage.

iSCSI storage devices from the storage system become available to the host. You can access the storage devices and create VMFS datastores for your storage needs.

For specific information on setting up the iSCSI SAN, see [Chapter 10 Using ESXi with iSCSI SAN](#).

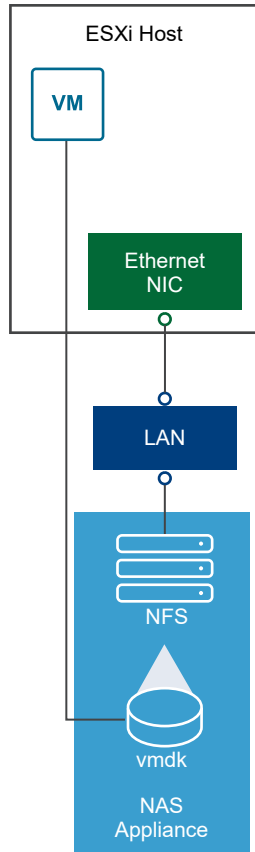
Network-attached Storage (NAS)

Stores virtual machine files on remote file servers accessed over a standard TCP/IP network. The NFS client built into ESXi uses Network File System (NFS) protocol version 3 and 4.1 to communicate with the NAS/NFS servers. For network connectivity, the host requires a standard network adapter.

You can mount an NFS volume directly on the ESXi host. You then use the NFS datastore to store and manage virtual machines in the same way that you use the VMFS datastores.

NFS Storage depicts a virtual machine using the NFS datastore to store its files. In this configuration, the host connects to the NAS server, which stores the virtual disk files, through a regular network adapter.

Figure 2-4. NFS Storage



For specific information on setting up NFS storage, see [Understanding Network File System Datastores](#).

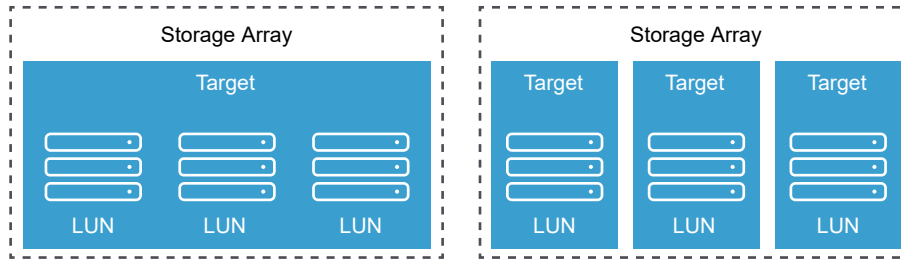
Shared Serial Attached SCSI (SAS)

Stores virtual machines on direct-attached SAS storage systems that offer shared access to multiple hosts. This type of access permits multiple hosts to access the same VMFS datastore on a LUN.

Target and Device Representations

In the ESXi context, the term target identifies a single storage unit that the host can access. The terms storage device and LUN describe a logical volume that represents storage space on a target. In the ESXi context, both terms also mean a storage volume that is presented to the host from a storage target and is available for formatting. Storage device and LUN are often used interchangeably.

Different storage vendors present the storage systems to ESXi hosts in different ways. Some vendors present a single target with multiple storage devices or LUNs on it, while others present multiple targets with one LUN each.

Figure 2-5. Target and LUN Representations

In this illustration, three LUNs are available in each configuration. In one case, the host connects to one target, but that target has three LUNs that can be used. Each LUN represents an individual storage volume. In the other example, the host detects three different targets, each having one LUN.

Targets that are accessed through the network have unique names that are provided by the storage systems. The iSCSI targets use iSCSI names. Fibre Channel targets use World Wide Names (WWNs).

Note ESXi does not support accessing the same LUN through different transport protocols, such as iSCSI and Fibre Channel.

A device, or LUN, is identified by its UUID name. If a LUN is shared by multiple hosts, it must be presented to all hosts with the same UUID.

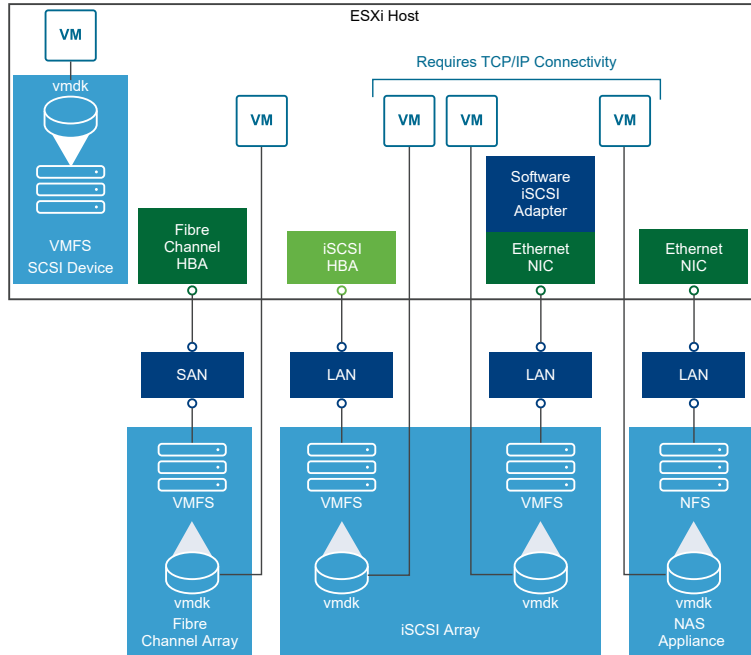
How Virtual Machines Access Storage

When a virtual machine communicates with its virtual disk stored on a datastore, it issues SCSI commands. Because datastores can exist on various types of physical storage, these commands are encapsulated into other forms, depending on the protocol that the ESXi host uses to connect to a storage device.

ESXi supports Fibre Channel (FC), Internet SCSI (iSCSI), Fibre Channel over Ethernet (FCoE), and NFS protocols. Regardless of the type of storage device your host uses, the virtual disk always appears to the virtual machine as a mounted SCSI device. The virtual disk hides a physical storage layer from the virtual machine's operating system. This allows you to run operating systems that are not certified for specific storage equipment, such as SAN, inside the virtual machine.

The following graphic depicts five virtual machines using different types of storage to illustrate the differences between each type.

Figure 2-6. Virtual machines accessing different types of storage



Note This diagram is for conceptual purposes only. It is not a recommended configuration.

Storage Device Characteristics

When your ESXi host connects to block-based storage systems, LUNs or storage devices that support ESXi become available to the host.

After the devices get registered with your host, you can display all available local and networked devices and review their information. If you use third-party multipathing plug-ins, the storage devices available through the plug-ins also appear on the list.

Note If an array supports implicit asymmetric logical unit access (ALUA) and has only standby paths, the registration of the device fails. The device can register with the host after the target activates a standby path and the host detects it as active. The advanced system `/Disk/FailDiskRegistration` parameter controls this behavior of the host.

For each storage adapter, you can display a separate list of storage devices available for this adapter.

Generally, when you review storage devices, you see the following information.

Table 2-1. Storage Device Information

| Storage Device Information | Description |
|----------------------------|--|
| Name | Also called Display Name. It is a name that the ESXi host assigns to the device based on the storage type and manufacturer. Generally, you can change this name to a name of your choice. See Rename Storage Devices . |
| Identifier | A universally unique identifier that is intrinsic to the device. See Storage Device Names and Identifiers . |
| Operational State | Indicates whether the device is attached or detached. See Detach Storage Devices . |
| LUN | Logical Unit Number (LUN) within the SCSI target. The LUN number is provided by the storage system. If a target has only one LUN, the LUN number is always zero (0). |
| Type | Type of device, for example, disk or CD-ROM. |
| Drive Type | Information about whether the device is a flash drive or a regular HDD drive. For information about flash drives and NVMe devices, see Chapter 15 Working with Flash Devices . |
| Transport | Transportation protocol your host uses to access the device. The protocol depends on the type of storage being used. See Types of Physical Storage . |
| Capacity | Total capacity of the storage device. |
| Owner | The plug-in, such as the NMP or a third-party plug-in, that the host uses to manage paths to the storage device. See Pluggable Storage Architecture and Path Management . |
| Hardware Acceleration | Information about whether the storage device assists the host with virtual machine management operations. The status can be Supported, Not Supported, or Unknown. See Chapter 24 Storage Hardware Acceleration . |
| Sector Format | Indicates whether the device uses a traditional, 512n, or advanced sector format, such as 512e or 4Kn. See Device Sector Formats . |
| Location | A path to the storage device in the <code>/vmfs/devices/</code> directory. |
| Partition Format | A partition scheme used by the storage device. It can be of a master boot record (MBR) or GUID partition table (GPT) format. The GPT devices can support datastores greater than 2 TB. See Device Sector Formats . |
| Partitions | Primary and logical partitions, including a VMFS datastore, if configured. |
| Multipathing Policies | Path Selection Policy and Storage Array Type Policy the host uses to manage paths to storage. See Chapter 18 Understanding Multipathing and Failover . |
| Paths | Paths used to access storage and their status. See Disable Storage Paths . |

Display Storage Devices for a Host

Display all storage devices available to a host. If you use any third-party multipathing plug-ins, the storage devices available through the plug-ins also appear on the list.

The Storage Devices view allows you to list the hosts' storage devices, analyze their information, and modify properties.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Devices**.

All storage devices available to the host are listed in the Storage Devices table.

- 4 To view details for a specific device, select the device from the list.
- 5 Use the icons to perform basic storage management tasks.

Availability of specific icons depends on the device type and configuration.

| Icon | Description |
|--------------------|---|
| Refresh | Refresh information about storage adapters, topology, and file systems. |
| Detach | Detach the selected device from the host. |
| Attach | Attach the selected device to the host. |
| Rename | Change the display name of the selected device. |
| Turn On LED | Turn on the locator LED for the selected devices. |
| Turn Off LED | Turn off the locator LED for the selected devices. |
| Mark as Flash Disk | Mark the selected devices as flash disks. |
| Mark as HDD Disk | Mark the selected devices as HDD disks. |
| Mark as Local | Mark the selected devices as local for the host. |
| Mark as Remote | Mark the selected devices as remote for the host. |
| Erase Partitions | Erase partitions on the selected devices. |

- 6 Use the following tabs to access additional information and modify properties for the selected device.

| Tab | Description |
|--------------------------|---|
| Properties | View device properties and characteristics. View and modify multipathing policies for the device. |
| Paths | Display paths available for the device. Disable or enable a selected path. |
| Partition Details | Displays information about partitions and their formats. |

Display Storage Devices for an Adapter

Display a list of storage devices accessible through a specific storage adapter on the host.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Adapters**.

All storage adapters installed on the host are listed in the Storage Adapters table.

4 Select the adapter from the list and click the **Devices** tab.

Storage devices that the host can access through the adapter are displayed.

5 Use the icons to perform basic storage management tasks.

Availability of specific icons depends on the device type and configuration.

| Icon | Description |
|--------------------|---|
| Refresh | Refresh information about storage adapters, topology, and file systems. |
| Detach | Detach the selected device from the host. |
| Attach | Attach the selected device to the host. |
| Rename | Change the display name of the selected device. |
| Turn On LED | Turn on the locator LED for the selected devices. |
| Turn Off LED | Turn off the locator LED for the selected devices. |
| Mark as Flash Disk | Mark the selected devices as flash disks. |
| Mark as HDD Disk | Mark the selected devices as HDD disks. |
| Mark as Local | Mark the selected devices as local for the host. |
| Mark as Remote | Mark the selected devices as remote for the host. |
| Erase Partitions | Erase partitions on the selected devices. |

Comparing Types of Storage

Whether certain vSphere functionality is supported might depend on the storage technology that you use.

The following table compares networked storage technologies that ESXi supports.

Table 2-2. Networked Storage that ESXi Supports

| Technology | Protocols | Transfers | Interface |
|-----------------------------|-----------|-----------------------------|---|
| Fibre Channel | FC/SCSI | Block access of data/LUN | FC HBA |
| Fibre Channel over Ethernet | FCoE/SCSI | Block access of data/LUN | <ul style="list-style-type: none"> ■ Converged Network Adapter (hardware FCoE) ■ NIC with FCoE support (software FCoE) |
| iSCSI | IP/SCSI | Block access of data/LUN | <ul style="list-style-type: none"> ■ iSCSI HBA or iSCSI-enabled NIC (hardware iSCSI) ■ Network adapter (software iSCSI) |
| NAS | IP/NFS | File (no direct LUN access) | Network adapter |

The following table compares the vSphere features that different types of storage support.

Table 2-3. vSphere Features Supported by Storage

| Storage Type | Boot VM | vMotion | Datastore | RDM | VM Cluster | VMware HA and DRS | Storage APIs - Data Protection |
|---------------|---------|---------|-------------------|-----|------------|-------------------|--------------------------------|
| Local Storage | Yes | No | VMFS | No | Yes | No | Yes |
| Fibre Channel | Yes | Yes | VMFS | Yes | Yes | Yes | Yes |
| iSCSI | Yes | Yes | VMFS | Yes | Yes | Yes | Yes |
| NAS over NFS | Yes | Yes | NFS 3 and NFS 4.1 | No | No | Yes | Yes |

Note Local storage supports a cluster of virtual machines on a single host (also known as a cluster in a box). A shared virtual disk is required. For more information about this configuration, see the *vSphere Resource Management* documentation.

Supported Storage Adapters

Storage adapters provide connectivity for your ESXi host to a specific storage unit or network.

ESXi supports different classes of adapters, including SCSI, iSCSI, RAID, Fibre Channel, Fibre Channel over Ethernet (FCoE), and Ethernet. ESXi accesses the adapters directly through device drivers in the VMkernel.

Depending on the type of storage you use, you might need to enable and configure a storage adapter on your host.

For information on setting up software FCoE adapters, see [Chapter 6 Configuring Fibre Channel over Ethernet](#).

For information on configuring different types of iSCSI adapters, see [Chapter 11 Configuring iSCSI Adapters and Storage](#).

View Storage Adapters Information

The host uses storage adapters to access different storage devices. You can display details for the available storage adapters and review their information.

Prerequisites

You must enable certain adapters, for example software iSCSI or FCoE, before you can view their information. To configure adapters, see the following:

- [Chapter 11 Configuring iSCSI Adapters and Storage](#)
- [Chapter 6 Configuring Fibre Channel over Ethernet](#)

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Adapters**.
- 4 Use the icons to perform storage adapter tasks.

Availability of specific icons depends on the storage configuration.

| Icon | Description |
|----------------------|---|
| Add Software Adapter | Add a storage adapter. Applies to software iSCSI and software FCoE. |
| Refresh | Refresh information about storage adapters, topology, and file systems on the host. |
| Rescan Storage | Rescan all storage adapters on the host to discover newly added storage devices or VMFS datastores. |
| Rescan Adapter | Rescan the selected adapter to discover newly added storage devices. |

- 5 To view details for a specific adapter, select the adapter from the list.
- 6 Use tabs under Adapter Details to access additional information and modify properties for the selected adapter.

| Tab | Description |
|--|--|
| Properties | Review general adapter properties that typically include a name and model of the adapter and unique identifiers formed according to specific storage standards. For iSCSI and FCoE adapters, use this tab to configure additional properties, for example, authentication. |
| Devices | View storage devices the adapter can access. Use the tab to perform basic device management tasks. See Display Storage Devices for an Adapter . |
| Paths | List and manage all paths the adapter uses to access storage devices. |
| Targets (Fibre Channel and iSCSI) | Review and manage targets accessed through the adapter. |
| Network Port Binding (iSCSI only) | Configure port binding for software and dependent hardware iSCSI adapters. |
| Advanced Options (iSCSI only) | Configure advanced parameters for iSCSI. |

Datastore Characteristics

Datastores are logical containers, analogous to file systems, that hide specifics of each storage device and provide a uniform model for storing virtual machine files. You can display all datastores available to your hosts and analyze their properties.

Datastores are added to vCenter Server in the following ways:

- You can create a VMFS datastore, an NFS version 3 or 4.1 datastore, or a Virtual Volumes datastore using the New Datastore wizard. A vSAN datastore is automatically created when you enable vSAN.
- When you add an ESXi host to vCenter Server, all datastores on the host are added to vCenter Server.

The following table describes datastore details that you can see when you review datastores through the vSphere Client. Certain characteristic might not be available or applicable to all types of datastores.

Table 2-4. Datastore Information

| Datastore Information | Applicable Datastore Type | Description |
|-----------------------|--|---|
| Name | VMFS NFS vSAN Virtual Volumes | Editable name that you assign to a datastore. For information on renaming a datastore, see Change Datastore Name . |
| Type | VMFS NFS vSAN Virtual Volumes | File system that the datastore uses. For information about VMFS and NFS datastores and how to manage them, see Chapter 17 Working with Datastores . For information about vSAN datastores, see the <i>Administering VMware vSAN</i> documentation. For information about Virtual Volumes, see Chapter 22 Working with Virtual Volumes . |
| Device Backing | VMFS NFS vSAN | Information about underlying storage, such as a storage device on which the datastore is deployed (VMFS), server and folder (NFS), or disk groups (vSAN). |
| Protocol Endpoints | Virtual Volumes | Information about corresponding protocol endpoints. See Protocol Endpoints . |
| Extents | VMFS | Individual extents that the datastore spans and their capacity. |
| Drive Type | VMFS | Type of the underlying storage device, such as a flash drive or a regular HDD drive. For details, see Chapter 15 Working with Flash Devices . |
| Capacity | VMFS NFS vSAN Virtual Volumes | Includes total capacity, provisioned space, and free space. |
| Mount Point | VMFS NFS vSAN Virtual Volumes | A path to the datastore in the <code>/vmfs/volumes/</code> directory of the host. |

Table 2-4. Datastore Information (continued)

| Datastore Information | Applicable Datastore Type | Description |
|-------------------------|--|--|
| Capability Sets | VMFS | Information about storage data services that the underlying storage entity provides. You cannot modify them. |
| | Note A multi-extent VMFS datastore assumes capabilities of only one of its extents. | |
| | NFS | |
| | vSAN Virtual Volumes | |
| Storage I/O Control | VMFS | Information on whether cluster-wide storage I/O prioritization is enabled. See the <i>vSphere Resource Management</i> documentation. |
| | NFS | |
| Hardware Acceleration | VMFS | Information on whether the underlying storage entity supports hardware acceleration. The status can be Supported, Not Supported, or Unknown. For details, see Chapter 24 Storage Hardware Acceleration . |
| | NFS | |
| | vSAN | |
| | Virtual Volumes | |
| | | Note NFS 4.1 does not support Hardware Acceleration. |
| Tags | VMFS | Datastore capabilities that you define and associate with datastores in a form of tags. For information, see Assign Tags to Datastores . |
| | NFS | |
| | vSAN | |
| | Virtual Volumes | |
| Connectivity with Hosts | VMFS | Hosts where the datastore is mounted. |
| | NFS | |
| | Virtual Volumes | |
| Multipathing | VMFS | Path selection policy the host uses to access storage. For more information, see Chapter 18 Understanding Multipathing and Failover . |
| | Virtual Volumes | |

Display Datastore Information

Access the Datastores view with the vSphere Client navigator.

Use the Datastores view to list all datastores available in the vSphere infrastructure inventory, analyze the information, and modify properties.

Procedure

- 1 Navigate to any inventory object that is a valid parent object of a datastore, such as a host, a cluster, or a data center, and click the **Datastores** tab.

Datastores that are available in the inventory appear in the center panel.

- 2 Use the options from a datastore right-click menu to perform basic tasks for a selected datastore.

Availability of specific options depends on the type of the datastore and its configuration.

| Option | Description |
|---|---|
| Register VM | Register an existing virtual machine in the inventory. See the <i>vSphere Virtual Machine Administration</i> documentation. |
| Increase Datastore Capacity | Increase the capacity of the VMFS datastore or add an extent. See Increase VMFS Datastore Capacity . |
| Browse Files | Navigate to the datastore file browser. See Use Datastore Browser . |
| Rename | Change the datastore name. See Change Datastore Name . |
| Mount Datastore | Mount the datastore to certain hosts. See Mount Datastores . |
| Unmount Datastore | Unmount the datastore from certain hosts. See Unmount Datastores . |
| Maintenance Mode | Use datastore maintenance mode. See the <i>vSphere Resource Management</i> documentation. |
| Configure Storage I/O Control (VMFS) | Enable Storage I/O Control for the VMFS datastore. See the <i>vSphere Resource Management</i> documentation. |
| Edit Space Reclamation (VMFS) | Change space reclamation settings for the VMFS datastore. See Change Space Reclamation Settings . |
| Delete Datastore (VMFS) | Remove the VMFS datastore. See Remove VMFS Datastores . |
| Tags & Custom Attributes | Use tags to encode information about the datastore. See Assign Tags to Datastores . |

- 3 To view specific datastore details, click a selected datastore.
- 4 Use tabs to access additional information and modify datastore properties.

| Tab | Description |
|--------------------|---|
| Summary | View statistics and configuration for the selected datastore. |
| Monitor | View alarms, performance data, resource allocation, events, and other status information for the datastore. |
| Configure | View and modify datastore properties. Menu items that you can see depend on the datastore type. |
| Permissions | Assign or edit permissions for the selected datastore. |
| Files | Navigate to the datastore file browser. |
| Hosts | View hosts where the datastore is mounted. |
| VMs | View virtual machines that reside on the datastore. |

Using Persistent Memory

ESXi supports next generation persistent memory devices, also known as Non-Volatile Memory (NVM) devices. These devices combine performance and speed of memory with the persistence of traditional storage. They can retain stored data through reboots or power source failures.

Virtual machines that require high bandwidth, low latency, and persistence can benefit from this technology. Examples include VMs with acceleration databases and analytics workload.

To use persistent memory with your ESXi host, you must be familiar with the following concepts.

PMem Datastore

After you add persistent memory to your ESXi host, the host detects the hardware, and then formats and mounts it as a local PMem datastore. ESXi uses VMFS-L as a file system format. Only one local PMem datastore per host is supported.

Note When you manage physical persistent memory, make sure to evacuate all VMs from the host and place the host into maintenance mode.

To reduce administrative overhead, the PMem datastore offers a simplified management model. Traditional datastore tasks do not generally apply to the datastore because the host automatically performs all the required operations on the background. As an administrator, you cannot display the datastore in the Datastores view of the vSphere Client, or perform other regular datastore actions. The only operation available to you is monitoring statistics for the PMem datastore.

The PMem datastore is used to store virtual NVDIMM devices and traditional virtual disks of a VM. The VM home directory with the `vmx` and `vmware.log` files cannot be placed on the PMem datastore.

PMem Access Modes

ESXi exposes persistent memory to a VM in two different modes. PMem-aware VMs can have direct access to persistent memory. Traditional VMs can use fast virtual disks stored on the PMem datastore.

Direct-Access Mode

In this mode, a PMem region can be presented to a VM as a virtual non-volatile dual in-line memory module (NVDIMM) module. The VM uses the NVDIMM module as a standard byte-addressable memory that can persist across power cycles.

You can add one or several NVDIMM modules when provisioning the VM.

The VMs must be of the hardware version ESXi 6.7 or later and have a PMem-aware guest OS. The NVDIMM device is compatible with latest guest OSes that support persistent memory, for example, Windows 2016.

Each NVDIMM device is automatically stored on the PMem datastore.

Virtual Disk Mode

This mode is available to any traditional VM and supports any hardware version, including all legacy versions. VMs are not required to be PMem-aware. When you use this mode, you create a regular SCSI virtual disk and attach a PMem VM storage policy to the disk. The policy automatically places the disk on the PMem datastore.

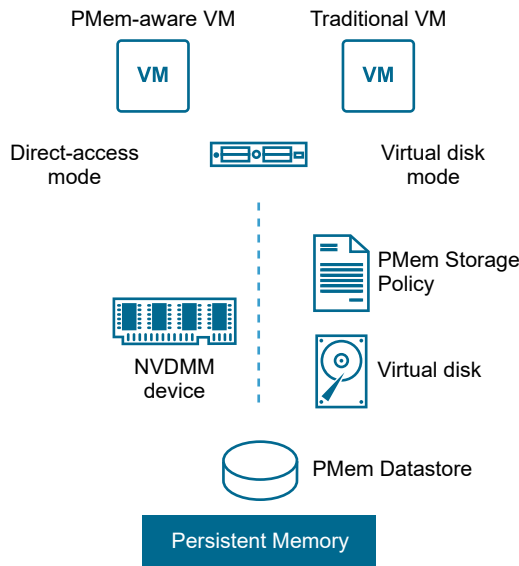
PMem Storage Policy

To place the virtual disk on the PMem datastore, you must apply the host-local PMem default storage policy to the disk. The policy is not editable.

The policy can be applied only to virtual disks. Because the VM home directory does not reside on the PMem datastore, make sure to place it on any standard datastore.

After you assign the PMem storage policy to the virtual disk, you cannot change the policy through the **VM Edit Setting** dialog box. To change the policy, migrate or clone the VM.

The following graphic illustrates how the persistent memory components interact.



For information about how to configure and manage VMs with NVDIMMs or virtual persistent memory disks, see the *vSphere Resource Management* documentation.

Monitor PMem Datastore Statistics

You can use the vSphere Client and the `esxcli` command to review the capacity of the PMem datastore and some of its other attributes.

However, unlike regular datastores, such as VMFS or VVols, the PMem datastore does not appear in the Datastores view of the vSphere Client. Regular datastore administrative tasks do not apply to it.

Procedure

- ◆ Review PMem datastore information.

| Option | Description |
|----------------|---|
| vSphere Client | <ol style="list-style-type: none"> Navigate to the ESXi host and click Summary. In the Hardware panel, verify that Persistent Memory is displayed and review its capacity. |
| esxcli command | Use the <code>esxcli storage filesystem list</code> to list the PMem datastore. |

Example: Viewing the PMem Datastore

The following sample output appears when you use the `esxcli storage filesystem list` command to list the datastore.

```
# esxcli storage filesystem list
Mount Point          Volume Name          UUID                Mounted  Type      Size
Free
-----
-----

/vmfs/volumes/5xxx...  ds01-102             5xxx...            true     VMFS-6    14227079168
12718178304
/vmfs/volumes/59ex...  ds02-102             59ex...            true     VMFS-6    21206401024
19697500160
/vmfs/volumes/59bx...  59bx...              true               vfat     4293591040
4274847744
/vmfs/volumes/pmem:5ax... PMemDS-56ax...       pmem:5a0x...       true     PMEM      12880707584
11504975872
```

Overview of Using ESXi with a SAN

3

Using ESXi with a SAN improves flexibility, efficiency, and reliability. Using ESXi with a SAN also supports centralized management, failover, and load balancing technologies.

The following are benefits of using ESXi with a SAN:

- You can store data securely and configure multiple paths to your storage, eliminating a single point of failure.
- Using a SAN with ESXi systems extends failure resistance to the server. When you use SAN storage, all applications can instantly be restarted on another host after the failure of the original host.
- You can perform live migration of virtual machines using VMware vMotion.
- Use VMware High Availability (HA) in conjunction with a SAN to restart virtual machines in their last known state on a different server if their host fails.
- Use VMware Fault Tolerance (FT) to replicate protected virtual machines on two different hosts. Virtual machines continue to function without interruption on the secondary host if the primary one fails.
- Use VMware Distributed Resource Scheduler (DRS) to migrate virtual machines from one host to another for load balancing. Because storage is on a shared SAN array, applications continue running seamlessly.
- If you use VMware DRS clusters, put an ESXi host into maintenance mode to have the system migrate all running virtual machines to other ESXi hosts. You can then perform upgrades or other maintenance operations on the original host.

The portability and encapsulation of VMware virtual machines complements the shared nature of this storage. When virtual machines are located on SAN-based storage, you can quickly shut down a virtual machine on one server and power it up on another server, or suspend it on one server and resume operation on another server on the same network. This ability allows you to migrate computing resources while maintaining consistent shared access.

Read the following topics next:

- [ESXi and SAN Use Cases](#)
- [Specifics of Using SAN Storage with ESXi](#)
- [ESXi Hosts and Multiple Storage Arrays](#)

- [Making LUN Decisions](#)
- [Selecting Virtual Machine Locations](#)
- [Third-Party Management Applications](#)
- [SAN Storage Backup Considerations](#)

ESXi and SAN Use Cases

When used with a SAN, ESXi can benefit from multiple vSphere features, including Storage vMotion, Distributed Resource Scheduler (DRS), High Availability, and so on.

Using ESXi with a SAN is effective for the following tasks:

Storage consolidation and simplification of storage layout

If you are working with multiple hosts, and each host is running multiple virtual machines, the storage on the hosts is no longer sufficient. You might need to use external storage. The SAN can provide a simple system architecture and other benefits.

Maintenance with zero downtime

When performing ESXi host or infrastructure maintenance, use vMotion to migrate virtual machines to other host. If shared storage is on the SAN, you can perform maintenance without interruptions to the users of the virtual machines. Virtual machine working processes continue throughout a migration.

Load balancing

You can add a host to a DRS cluster, and the host's resources become part of the cluster's resources. The distribution and use of CPU and memory resources for all hosts and virtual machines in the cluster are continuously monitored. DRS compares these metrics to an ideal resource use. The ideal use considers the attributes of the cluster's resource pools and virtual machines, the current demand, and the imbalance target. If needed, DRS performs or recommends virtual machine migrations.

Disaster recovery

You can use VMware High Availability to configure multiple ESXi hosts as a cluster. The cluster provides rapid recovery from outages and cost-effective high availability for applications running in virtual machines.

Simplified array migrations and storage upgrades

When you purchase new storage systems, use Storage vMotion to perform live migrations of virtual machines from existing storage to their new destinations. You can perform the migrations without interruptions of the virtual machines.

Specifics of Using SAN Storage with ESXi

Using a SAN with an ESXi host differs from a traditional SAN use in several ways.

When you use SAN storage with ESXi, the following considerations apply:

- You cannot use SAN administration tools to access operating systems of virtual machines that reside on the storage. With traditional tools, you can monitor only the VMware ESXi operating system. You use the vSphere Client to monitor virtual machines.
- The HBA visible to the SAN administration tools is part of the ESXi system, not part of the virtual machine.
- Typically, your ESXi system performs multipathing for you.

ESXi Hosts and Multiple Storage Arrays

An ESXi host can access storage devices presented from multiple storage arrays, including arrays from different vendors.

When you use multiple arrays from different vendors, the following considerations apply:

- If your host uses the same SATP for multiple arrays, be careful when you change the default PSP for that SATP. The change applies to all arrays. For information on SATPs and PSPs, see [Chapter 18 Understanding Multipathing and Failover](#).
- Some storage arrays make recommendations on queue depth and other settings. Typically, these settings are configured globally at the ESXi host level. Changing settings for one array impacts other arrays that present LUNs to the host. For information on changing queue depth, see the VMware knowledge base article at <http://kb.vmware.com/kb/1267>.
- Use single-initiator-single-target zoning when zoning ESXi hosts to Fibre Channel arrays. With this type of configuration, fabric-related events that occur on one array do not impact other arrays. For more information about zoning, see [Using Zoning with Fibre Channel SANs](#).

Making LUN Decisions

You must plan how to set up storage for your ESXi systems before you format LUNs with VMFS datastores.

When you make your LUN decision, the following considerations apply:

- Each LUN must have the correct RAID level and storage characteristic for the applications running in virtual machines that use the LUN.
- Each LUN must contain only one VMFS datastore.
- If multiple virtual machines access the same VMFS, use disk shares to prioritize virtual machines.

You might want fewer, larger LUNs for the following reasons:

- More flexibility to create virtual machines without asking the storage administrator for more space.
- More flexibility for resizing virtual disks, doing snapshots, and so on.
- Fewer VMFS datastores to manage.

You might want more, smaller LUNs for the following reasons:

- Less wasted storage space.
- Different applications might need different RAID characteristics.
- More flexibility, as the multipathing policy and disk shares are set per LUN.
- Use of Microsoft Cluster Service requires that each cluster disk resource is in its own LUN.
- Better performance because there is less contention for a single volume.

When the storage characterization for a virtual machine is unavailable, it might not be easy to determine the number and size of LUNs to provision. You can experiment using either a predictive or adaptive scheme.

Use the Predictive Scheme to Make LUN Decisions

When setting up storage for ESXi systems, before creating VMFS datastores, you must decide on the size and number of LUNs to provision. You can experiment using the predictive scheme.

Procedure

- 1 Provision several LUNs with different storage characteristics.
- 2 Create a VMFS datastore on each LUN, labeling each datastore according to its characteristics.
- 3 Create virtual disks to contain the data for virtual machine applications in the VMFS datastores created on LUNs with the appropriate RAID level for the applications' requirements.
- 4 Use disk shares to distinguish high-priority from low-priority virtual machines.

Note Disk shares are relevant only within a given host. The shares assigned to virtual machines on one host have no effect on virtual machines on other hosts.

- 5 Run the applications to determine whether virtual machine performance is acceptable.

Use the Adaptive Scheme to Make LUN Decisions

When setting up storage for ESXi hosts, before creating VMFS datastores, you must decide on the number and size of LUNs to provision. You can experiment using the adaptive scheme.

Procedure

- 1 Provision a large LUN (RAID 1+0 or RAID 5), with write caching enabled.
- 2 Create a VMFS on that LUN.
- 3 Create four or five virtual disks on the VMFS.
- 4 Run the applications to determine whether disk performance is acceptable.

Results

If performance is acceptable, you can place additional virtual disks on the VMFS. If performance is not acceptable, create a new, large LUN, possibly with a different RAID level, and repeat the process. Use migration so that you do not lose virtual machines data when you recreate the LUN.

Selecting Virtual Machine Locations

When you are working on optimizing performance for your virtual machines, storage location is an important factor. Depending on your storage needs, you might select storage with high performance and high availability, or storage with lower performance.

Storage can be divided into different tiers depending on several factors:

- High Tier. Offers high performance and high availability. Might offer built-in snapshots to facilitate backups and point-in-time (PiT) restorations. Supports replication, full storage processor redundancy, and SAS drives. Uses high-cost spindles.
- Mid Tier. Offers mid-range performance, lower availability, some storage processor redundancy, and SCSI or SAS drives. Might offer snapshots. Uses medium-cost spindles.
- Lower Tier. Offers low performance, little internal storage redundancy. Uses low-end SCSI drives or SATA.

Not all VMs must be on the highest-performance and most-available storage throughout their entire life cycle.

When you decide where to place a virtual machine, the following considerations apply:

- Criticality of the VM
- Performance and availability requirements
- PiT restoration requirements
- Backup and replication requirements

A virtual machine might change tiers throughout its life cycle because of changes in criticality or changes in technology. Criticality is relative and might change for various reasons, including changes in the organization, operational processes, regulatory requirements, disaster planning, and so on.

Third-Party Management Applications

You can use third-party management applications with your ESXi host.

Most SAN hardware is packaged with storage management software. In many cases, this software is a Web application that can be used with any Web browser connected to your network. In other cases, this software typically runs on the storage system or on a single server, independent of the servers that use the SAN for storage.

Use this third-party management software for the following tasks:

- Storage array management, including LUN creation, array cache management, LUN mapping, and LUN security.
- Setting up replication, check points, snapshots, or mirroring.

If you run the SAN management software on a virtual machine, you gain the benefits of a virtual machine, including failover with vMotion and VMware HA. Because of the additional level of indirection, however, the management software might not see the SAN. In this case, you can use an RDM.

Note Whether a virtual machine can run management software successfully depends on the particular storage system.

SAN Storage Backup Considerations

Having a proper backup strategy is one of the most important aspects of SAN management. In the SAN environment, backups have two goals. The first goal is to archive online data to offline media. This process is repeated periodically for all online data on a time schedule. The second goal is to provide access to offline data for recovery from a problem. For example, database recovery often requires a retrieval of archived log files that are not currently online.

Scheduling a backup depends on several factors:

- Identification of critical applications that require more frequent backup cycles within a given period.
- Recovery point and recovery time goals. Consider how precise your recovery point must be, and how long you are willing to wait for it.
- The rate of change (RoC) associated with the data. For example, if you are using synchronous/asynchronous replication, the RoC affects the amount of bandwidth required between the primary and secondary storage devices.
- Overall impact on a SAN environment, storage performance, and other applications.
- Identification of peak traffic periods on the SAN. Backups scheduled during those peak periods can slow the applications and the backup process.
- Time to schedule all backups within the data center.
- Time it takes to back up an individual application.

- Resource availability for archiving data, such as offline media access.

Include a recovery-time objective for each application when you design your backup strategy. That is, consider the time and resources necessary to perform a backup. For example, if a scheduled backup stores so much data that recovery requires a considerable amount of time, examine the scheduled backup. Perform the backup more frequently, so that less data is backed up at a time and the recovery time decreases.

If an application requires recovery within a certain time frame, the backup process must provide a time schedule and specific data processing to meet the requirement. Fast recovery can require the use of recovery volumes that reside on online storage. This process helps to minimize or eliminate the need to access slow offline media for missing data components.

Using Third-Party Backup Packages

You can use third-party backup solutions to protect system, application, and user data in your virtual machines.

The Storage APIs - Data Protection that VMware offers can work with third-party products. When using the APIs, third-party software can perform backups without loading ESXi hosts with the processing of backup tasks.

The third-party products using the Storage APIs - Data Protection can perform the following backup tasks:

- Perform a full, differential, and incremental image backup and restore of virtual machines.
- Perform a file-level backup of virtual machines that use supported Windows and Linux operating systems.
- Ensure data consistency by using Microsoft Volume Shadow Copy Services (VSS) for virtual machines that run supported Microsoft Windows operating systems.

Because the Storage APIs - Data Protection use the snapshot capabilities of VMFS, backups do not require that you stop virtual machines. These backups are nondisruptive, can be performed at any time, and do not need extended backup windows.

For information about the Storage APIs - Data Protection and integration with backup products, see the VMware website or contact your vendor.

Using ESXi with Fibre Channel SAN

4

When you set up ESXi hosts to use FC SAN storage arrays, special considerations are necessary. This section provides introductory information about how to use ESXi with an FC SAN array.

Read the following topics next:

- [Fibre Channel SAN Concepts](#)
- [Using Zoning with Fibre Channel SANs](#)
- [How Virtual Machines Access Data on a Fibre Channel SAN](#)

Fibre Channel SAN Concepts

If you are an ESXi administrator planning to set up hosts to work with SANs, you must have a working knowledge of SAN concepts. You can find information about SANs in print and on the Internet. Because this industry changes constantly, check these resources frequently.

If you are new to SAN technology, familiarize yourself with the basic terminology.

A storage area network (SAN) is a specialized high-speed network that connects host servers to high-performance storage subsystems. The SAN components include host bus adapters (HBAs) in the host servers, switches that help route storage traffic, cables, storage processors (SPs), and storage disk arrays.

A SAN topology with at least one switch present on the network forms a SAN fabric.

To transfer traffic from host servers to shared storage, the SAN uses the Fibre Channel (FC) protocol that packages SCSI commands into Fibre Channel frames.

To restrict server access to storage arrays not allocated to that server, the SAN uses zoning. Typically, zones are created for each group of servers that access a shared group of storage devices and LUNs. Zones define which HBAs can connect to which SPs. Devices outside a zone are not visible to the devices inside the zone.

Zoning is similar to LUN masking, which is commonly used for permission management. LUN masking is a process that makes a LUN available to some hosts and unavailable to other hosts.

When transferring data between the host server and storage, the SAN uses a technique known as multipathing. Multipathing allows you to have more than one physical path from the ESXi host to a LUN on a storage system.

Generally, a single path from a host to a LUN consists of an HBA, switch ports, connecting cables, and the storage controller port. If any component of the path fails, the host selects another available path for I/O. The process of detecting a failed path and switching to another is called path failover.

Ports in Fibre Channel SAN

In the context of this document, a port is the connection from a device into the SAN. Each node in the SAN, such as a host, a storage device, or a fabric component has one or more ports that connect it to the SAN. Ports are identified in a number of ways.

WWPN (World Wide Port Name)

A globally unique identifier for a port that allows certain applications to access the port. The FC switches discover the WWPN of a device or host and assign a port address to the device.

Port_ID (or port address)

Within a SAN, each port has a unique port ID that serves as the FC address for the port. This unique ID enables routing of data through the SAN to that port. The FC switches assign the port ID when the device logs in to the fabric. The port ID is valid only while the device is logged on.

When N-Port ID Virtualization (NPIV) is used, a single FC HBA port (N-port) can register with the fabric by using several WWPNs. This method allows an N-port to claim multiple fabric addresses, each of which appears as a unique entity. When ESXi hosts use a SAN, these multiple, unique identifiers allow the assignment of WWNs to individual virtual machines as part of their configuration.

Fibre Channel Storage Array Types

ESXi supports different storage systems and arrays.

The types of storage that your host supports include active-active, active-passive, and ALUA-compliant.

Active-active storage system

Supports access to the LUNs simultaneously through all the storage ports that are available without significant performance degradation. All the paths are active, unless a path fails.

Active-passive storage system

A system in which one storage processor is actively providing access to a given LUN. The other processors act as a backup for the LUN and can be actively providing access to other LUN I/O. I/O can be successfully sent only to an active port for a given LUN. If access through the active storage port fails, one of the passive storage processors can be activated by the servers accessing it.

Asymmetrical storage system

Supports Asymmetric Logical Unit Access (ALUA). ALUA-compliant storage systems provide different levels of access per port. With ALUA, the host can determine the states of target ports and prioritize paths. The host uses some of the active paths as primary, and uses others as secondary.

Using Zoning with Fibre Channel SANs

Zoning provides access control in the SAN topology. Zoning defines which HBAs can connect to which targets. When you configure a SAN by using zoning, the devices outside a zone are not visible to the devices inside the zone.

Zoning has the following effects:

- Reduces the number of targets and LUNs presented to a host.
- Controls and isolates paths in a fabric.
- Can prevent non-ESXi systems from accessing a particular storage system, and from possibly destroying VMFS data.
- Can be used to separate different environments, for example, a test from a production environment.

With ESXi hosts, use a single-initiator zoning or a single-initiator-single-target zoning. The latter is a preferred zoning practice. Using the more restrictive zoning prevents problems and misconfigurations that can occur on the SAN.

For detailed instructions and best zoning practices, contact storage array or switch vendors.

How Virtual Machines Access Data on a Fibre Channel SAN

ESXi stores a virtual machine's disk files within a VMFS datastore that resides on a SAN storage device. When virtual machine guest operating systems send SCSI commands to their virtual disks, the SCSI virtualization layer translates these commands to VMFS file operations.

When a virtual machine interacts with its virtual disk stored on a SAN, the following process takes place:

- 1 When the guest operating system in a virtual machine reads or writes to a SCSI disk, it sends SCSI commands to the virtual disk.
- 2 Device drivers in the virtual machine's operating system communicate with the virtual SCSI controllers.
- 3 The virtual SCSI controller forwards the command to the VMkernel.
- 4 The VMkernel performs the following tasks.
 - a Locates the appropriate virtual disk file in the VMFS volume.
 - b Maps the requests for the blocks on the virtual disk to blocks on the appropriate physical device.

- c Sends the modified I/O request from the device driver in the VMkernel to the physical HBA.
- 5 The physical HBA performs the following tasks.
 - a Packages the I/O request according to the rules of the FC protocol.
 - b Transmits the request to the SAN.
- 6 Depending on a port the HBA uses to connect to the fabric, one of the SAN switches receives the request. The switch routes the request to the appropriate storage device.

Configuring Fibre Channel Storage

5

When you use ESXi systems with SAN storage, specific hardware and system requirements exist.

Read the following topics next:

- [ESXi Fibre Channel SAN Requirements](#)
- [Installation and Setup Steps](#)
- [N-Port ID Virtualization](#)

ESXi Fibre Channel SAN Requirements

In preparation for configuring your SAN and setting up your ESXi system to use SAN storage, review the requirements and recommendations.

- Make sure that ESXi systems support the SAN storage hardware and firmware combinations you use. For an up-to-date list, see the *VMware Compatibility Guide*.
- Configure your system to have only one VMFS volume per LUN.
- Unless you are using diskless servers, do not set up the diagnostic partition on a SAN LUN.
If you use diskless servers that boot from a SAN, a shared diagnostic partition is appropriate.
- Use RDMs to access raw disks. For information, see [Chapter 19 Raw Device Mapping](#).
- For multipathing to work properly, each LUN must present the same LUN ID number to all ESXi hosts.
- Make sure that the storage device driver specifies a large enough queue. You can set the queue depth for the physical HBA during a system setup.
- VMtools increases the default disk timeout SCSI `TimeoutValue` parameter to 180 seconds. On virtual machines without VMtools, increase the value of the SCSI `TimeoutValue` parameter. With this parameter set up, the VMs can better tolerate delayed I/O resulting from a path failover. For information, see [Set Timeout on Windows Guest OS](#) or the [KB 1009465](#).

ESXi Fibre Channel SAN Restrictions

When you use ESXi with a SAN, certain restrictions apply.

- ESXi does not support FC connected tape devices.
- You cannot use multipathing software inside a virtual machine to perform I/O load balancing to a single physical LUN. However, when your Microsoft Windows virtual machine uses dynamic disks, this restriction does not apply. For information about configuring dynamic disks, see [Set Up Dynamic Disk Mirroring](#).

Setting LUN Allocations

This topic provides general information about how to allocate LUNs when your ESXi works with SAN.

When you set LUN allocations, be aware of the following points:

Storage provisioning

To ensure that the ESXi system recognizes the LUNs at startup time, provision all LUNs to the appropriate HBAs before you connect the SAN to the ESXi system.

Provision all LUNs to all ESXi HBAs at the same time. HBA failover works only if all HBAs see the same LUNs.

For LUNs that are shared among multiple hosts, make sure that LUN IDs are consistent across all hosts.

vMotion and VMware DRS

When you use vCenter Server and vMotion or DRS, make sure that the LUNs for the virtual machines are provisioned to all ESXi hosts. This action provides the most ability to move virtual machines.

Active-active compared to active-passive arrays

When you use vMotion or DRS with an active-passive SAN storage device, make sure that all ESXi systems have consistent paths to all storage processors. Not doing so can cause path thrashing when a vMotion migration occurs.

For active-passive storage arrays not listed in Storage/SAN Compatibility, VMware does not support storage port failover. In those cases, you must connect the server to the active port on the storage array. This configuration ensures that the LUNs are presented to the ESXi host.

Setting Fibre Channel HBAs

Typically, FC HBAs that you use on your ESXi host work correctly with the default configuration settings.

You should follow the configuration guidelines provided by your storage array vendor. During FC HBA setup, consider the following issues.

- Do not mix FC HBAs from different vendors in a single host. Having different models of the same HBA is supported, but a single LUN cannot be accessed through two different HBA types, only through the same type.
- Ensure that the firmware level on each HBA is the same.
- Set the timeout value for detecting a failover. To ensure optimal performance, do not change the default value.
- ESXi supports 32 Gbps end-to-end Fibre Channel connectivity.

Installation and Setup Steps

This topic provides an overview of installation and setup steps that you need to follow when configuring your SAN environment to work with ESXi.

Follow these steps to configure your ESXi SAN environment.

- 1 Design your SAN if it is not already configured. Most existing SANs require only minor modification to work with ESXi.
- 2 Check that all SAN components meet requirements.
- 3 Perform any necessary storage array modification.

Most vendors have vendor-specific documentation for setting up a SAN to work with VMware ESXi.

- 4 Set up the HBAs for the hosts you have connected to the SAN.
- 5 Install ESXi on the hosts.
- 6 Create virtual machines and install guest operating systems.
- 7 (Optional) Set up your system for VMware HA failover or for using Microsoft Clustering Services.
- 8 Upgrade or modify your environment as needed.

N-Port ID Virtualization

N-Port ID Virtualization (NPIV) is an ANSI T11 standard that describes how a single Fibre Channel HBA port can register with the fabric using several worldwide port names (WWPNs). This allows a fabric-attached N-port to claim multiple fabric addresses. Each address appears as a unique entity on the Fibre Channel fabric.

How NPIV-Based LUN Access Works

NPIV enables a single FC HBA port to register several unique World Wide Name (WWN) identifiers with the fabric, each of which can be assigned to an individual virtual machine. When using NPIV, a SAN administrator can monitor and route storage access per a virtual machine.

Only virtual machines with RDMs can have WWN assignments, and they use these assignments for all RDM traffic.

When a virtual machine has a WWN assigned to it, the virtual machine's configuration file (.vmx) is updated to include a WWN pair. The WWN pair consists of a World Wide Port Name (WWPN) and a World Wide Node Name (WWNN). When that virtual machine is powered on, the VMkernel creates a virtual port (VPORT) on the physical HBA which is used to access the LUN. The VPORT is a virtual HBA that appears to the FC fabric as a physical HBA. As its unique identifier, the VPORT uses the WWN pair that was assigned to the virtual machine.

Each VPORT is specific to the virtual machine. The VPORT is destroyed on the host and no longer appears to the FC fabric when the virtual machine is powered off. When a virtual machine is migrated from one host to another, the VPORT closes on the first host and opens on the destination host.

When virtual machines do not have WWN assignments, they access storage LUNs with the WWNs of their host's physical HBAs.

Requirements for Using NPIV

If you plan to enable NPIV on your virtual machines, you should be aware of certain requirements.

- NPIV can be used only for virtual machines with RDM disks. Virtual machines with regular virtual disks use the WWNs of the host's physical HBAs.
- HBAs on your host must support NPIV.

For information, see the *VMware Compatibility Guide* and refer to your vendor documentation.

- Use HBAs of the same type. VMware does not support heterogeneous HBAs on the same host accessing the same LUNs.
- If a host uses multiple physical HBAs as paths to the storage, zone all physical paths to the virtual machine. This is required to support multipathing even though only one path at a time will be active.
- Make sure that physical HBAs on the host can detect all LUNs that are to be accessed by NPIV-enabled virtual machines running on that host.
- The switches in the fabric must be NPIV-aware.
- When configuring a LUN for NPIV access at the storage level, make sure that the NPIV LUN number and NPIV target ID match the physical LUN and Target ID.

- Zone the NPIV WWPNs so that they connect to all storage systems the cluster hosts can access, even if the VM does not use the storage. If you add any new storage systems to a cluster with one or more NPIV-enabled VMs, add the new zones, so the NPIV WWPNs can detect the new storage system target ports.

NPIV Capabilities and Limitations

Learn about specific capabilities and limitations of the use of NPIV with ESXi.

ESXi with NPIV supports the following items:

- NPIV supports vMotion. When you use vMotion to migrate a virtual machine it retains the assigned WWN.

If you migrate an NPIV-enabled virtual machine to a host that does not support NPIV, VMkernel reverts to using a physical HBA to route the I/O.
- If your FC SAN environment supports concurrent I/O on the disks from an active-active array, the concurrent I/O to two different NPIV ports is also supported.

When you use ESXi with NPIV, the following limitations apply:

- Because the NPIV technology is an extension to the FC protocol, it requires an FC switch and does not work on the direct attached FC disks.
- When you clone a virtual machine or template with a WWN assigned to it, the clones do not retain the WWN.
- NPIV does not support Storage vMotion.
- Disabling and then re-enabling the NPIV capability on an FC switch while virtual machines are running can cause an FC link to fail and I/O to stop.

Configure or Modify WWN Assignments

Assign WWN settings to virtual machine. You can later modify the WWN assignments.

You can create from 1 to 16 WWN pairs, which can be mapped to the first 1 to 16 physical FC HBAs on the host.

Typically, you do not need to change existing WWN assignments on your virtual machine. In certain circumstances, for example, when manually assigned WWNs are causing conflicts on the SAN, you might need to change or remove WWNs.

Prerequisites

- Before configuring WWN, ensure that the ESXi host can access the storage LUN access control list (ACL) configured on the array side.
- If you want to edit the existing WWNs, power off the virtual machine.

Procedure

- 1 Right-click the virtual machine in the inventory and select **Edit Settings**.

- 2 Click **VM Options** and expand **Fibre Channel NPIV**.
- 3 Create or edit the WWN assignments by selecting one of the following options:

| Option | Description |
|---|--|
| Temporarily disable NPIV for this virtual machine | Disable but do not remove the existing WWN assignments for the virtual machine. |
| Leave unchanged | Retain the existing WWN assignments. The read-only WWN assignments section displays the node and port values of any existing WWN assignments. |
| Generate new WWNs | Generate new WWNs, overwriting any existing WWNs. The WWNs of the HBA are not affected. Specify the number of WWNNs and WWPNS. A minimum of two WWPNS are required to support failover with NPIV. Typically only one WWNN is created for each virtual machine. |
| Remove WWN assignment | Remove the WWNs assigned to the virtual machine. The virtual machine uses the HBA WWNs to access the storage LUN. |

- 4 Click **OK** to save your changes.

What to do next

Register newly created WWNs in the fabric.

Configuring Fibre Channel over Ethernet

6

To access Fibre Channel storage, an ESXi host can use the Fibre Channel over Ethernet (FCoE) protocol.

The FCoE protocol encapsulates Fibre Channel frames into Ethernet frames. As a result, your host does not need special Fibre Channel links to connect to Fibre Channel storage. The host can use 10 Gbit lossless Ethernet to deliver Fibre Channel traffic.

Read the following topics next:

- [Fibre Channel over Ethernet Adapters](#)
- [Configuration Guidelines for Software FCoE](#)
- [Set Up Networking for Software FCoE](#)
- [Add Software FCoE Adapters](#)

Fibre Channel over Ethernet Adapters

To use Fibre Channel over Ethernet (FCoE), configure appropriate adapters on your host.

The adapters that VMware supports generally fall into two categories, hardware FCoE adapters and software FCoE adapters that use the native FCoE stack in ESXi.

For information on adapters that can be used with VMware FCoE, see the *VMware Compatibility Guide*

Hardware FCoE Adapters

This category includes a specialized offloaded Converged Network Adapters (CNAs) that contain network and Fibre Channel functionalities on the same card.

When such adapter is installed, your host detects and can use both CNA components. In the vSphere Client, the networking component appears as a standard network adapter (vmnic) and the Fibre Channel component as a FCoE adapter (vmhba). You do not need to configure the hardware FCoE adapter to use it.

Software FCoE Adapters

A software FCoE adapter uses the native FCoE protocol stack in ESXi to perform some of the FCoE processing. You must use the software FCoE adapter with a compatible NIC.

VMware supports two categories of NICs with the software FCoE adapters.

NICs With Partial FCoE Offload

The extent of the offload capabilities might depend on the type of the NIC. Generally, the NICs offer Data Center Bridging (DCB) and I/O offload capabilities.

NICs Without FCoE Offload

Any NICs that offer Data Center Bridging (DCB) and have a minimum speed of 10 Gbps. The network adapters are not required to support any FCoE offload capabilities.

Unlike the hardware FCoE adapter, the software adapter must be activated. Before you activate the adapter, you must properly configure networking.

Note The number of software FCoE adapters you activate corresponds to the number of physical NIC ports. ESXi supports a maximum of four software FCoE adapters on one host.

Configuration Guidelines for Software FCoE

When setting up your network environment to work with ESXi software FCoE, follow the guidelines and best practices that VMware offers.

Network Switch Guidelines

Follow these guidelines when you configure a network switch for software FCoE environment:

- On the ports that communicate with your ESXi host, disable the Spanning Tree Protocol (STP). Having the STP enabled might delay the FCoE Initialization Protocol (FIP) response at the switch and cause an all paths down (APD) condition.

The FIP is a protocol that FCoE uses to discover and initialize FCoE entities on the Ethernet.

- Turn on Priority-based Flow Control (PFC) and set it to AUTO.
- Make sure that you have a compatible firmware version on the FCoE switch.
- Set the MTU on the vSwitch to 2500 or more.

Network Adapter Guidelines and Best Practices

If you plan to enable software FCoE adapters to work with network adapters, specific considerations apply.

- Whether you use a partially offloaded NIC or a non-FCoE capable NIC, make sure that the latest microcode is installed on the network adapter.
- If you use the non-FCoE capable NIC, make sure that it has the DCB capability for software FCoE enablement.
- If the network adapter has multiple ports, when configuring networking, add each port to a separate vSwitch. This practice helps you to avoid an APD condition when a disruptive event, such as an MTU change, occurs.

- Do not move a network adapter port from one vSwitch to another when FCoE traffic is active. If you make this change, reboot your host afterwards.
- If you changed the vSwitch for a network adapter port and caused a failure, moving the port back to the original vSwitch resolves the problem.

Set Up Networking for Software FCoE

Before you activate the software FCoE adapters, create VMkernel network adapters for all physical FCoE NICs installed on your host.

This procedure explains how to create a single VMkernel network adapter connected to a single FCoE physical network adapter through a vSphere Standard switch. If your host has multiple network adapters or multiple ports on the adapter, connect each FCoE NIC to a separate standard switch. For more information, see the *vSphere Networking* documentation.

Procedure

- 1 Navigate to the host.
- 2 Click **Actions > Add Networking**.
- 3 Select **VMkernel Network Adapter**, and click **Next**.
- 4 Select **New standard switch** to create a vSphere Standard switch.
- 5 To enable Jumbo Frames, change **MTU (Bytes)** to the value of 2500 or more, and click **Next**.
- 6 Click the **Add adapters** icon, and select the network adapter (vmnic#) that supports FCoE.
Make sure to assign the adapter to Active Adapters.
- 7 Enter a network label.

Network label is a friendly name that identifies the VMkernel adapter that you are creating, for example, FCoE.
- 8 Specify a VLAN ID and click **Next**.

FCoE traffic requires an isolated network. Make sure that the VLAN ID you enter is different from the one used for regular networking on your host. For more information, see the *vSphere Networking* documentation.
- 9 After you finish configuration, review the information and click **Finish**.

Results

You have created the virtual VMkernel adapter for the physical FCoE network adapter installed on your host.

Note To avoid FCoE traffic disruptions, do not remove the FCoE network adapter (vmnic#) from the vSphere Standard switch after you set up FCoE networking.

Add Software FCoE Adapters

You must activate software FCoE adapters so that your host can use them to access Fibre Channel storage.

The number of software FCoE adapters you can activate corresponds to the number of physical FCoE NIC ports on your host. ESXi supports the maximum of four software FCoE adapters on one host.

Prerequisites

Set up networking for the software FCoE adapter.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Adapters**, and click the **Add** icon (+).
- 4 Select **Software FCoE Adapter**.
- 5 On the Add Software FCoE Adapter dialog box, select an appropriate vmnic from the drop-down list of physical network adapters.

Only those adapters that are not yet used for FCoE traffic are listed.

- 6 Click **OK**.

The software FCoE adapter appears on the list of storage adapters.

Results

After you activate the software FCoE adapter, you can view its properties. If you do not use the adapter, you can remove it from the list of adapters.

Booting ESXi from Fibre Channel SAN

7

When you set up your host to boot from a SAN, your host's boot image is stored on one or more LUNs in the SAN storage system. When the host starts, it boots from the LUN on the SAN rather than from its local disk.

ESXi supports booting through a Fibre Channel host bus adapter (HBA) or a Fibre Channel over Ethernet (FCoE) converged network adapter (CNA).

Read the following topics next:

- [Boot from SAN Benefits](#)
- [Requirements and Considerations when Booting from Fibre Channel SAN](#)
- [Getting Ready for Boot from SAN](#)
- [Configure Emulex HBA to Boot from SAN](#)
- [Configure QLogic HBA to Boot from SAN](#)

Boot from SAN Benefits

Boot from SAN can provide numerous benefits to your ESXi environment. However, in certain cases, boot from SAN is not compatible with your hosts. Before you set up your system for boot from SAN, decide whether it is appropriate for your environment.

Caution When you use boot from SAN with multiple ESXi hosts, each host must have its own boot LUN. If you configure multiple hosts to share the boot LUN, ESXi image corruption might occur.

If you use boot from SAN, the benefits for your environment include the following:

- Cheaper servers. Servers can be more dense and run cooler without internal storage.
- Easier server replacement. You can replace servers and have the new server point to the old boot location.
- Less wasted space. Servers without local disks often take up less space.
- Easier backup processes. You can back up the system boot images in the SAN as part of the overall SAN backup procedures. Also, you can use advanced array features such as snapshots on the boot image.

- Improved management. Creating and managing the operating system image is easier and more efficient.
- Better reliability. You can access the boot disk through multiple paths, which protects the disk from being a single point of failure.

Requirements and Considerations when Booting from Fibre Channel SAN

Your ESXi boot configuration must meet specific requirements.

Table 7-1. Boot from SAN Requirements

| Requirement | Description |
|-----------------------------------|--|
| ESXi system requirements | Follow vendor recommendations for the server booting from a SAN. |
| Adapter requirements | Configure the adapter, so it can access the boot LUN. See your vendor documentation. |
| Access control | <ul style="list-style-type: none"> ■ Each host must have access to its own boot LUN only, not the boot LUNs of other hosts. Use storage system software to make sure that the host accesses only the designated LUNs. ■ Multiple servers can share a diagnostic partition. You can use array-specific LUN masking to achieve this configuration. |
| Multipathing support | Multipathing to a boot LUN on active-passive arrays is not supported because the BIOS does not support multipathing and is unable to activate a standby path. |
| SAN considerations | If the array is not certified for a direct connect topology, the SAN connections must be through a switched topology. If the array is certified for the direct connect topology, the SAN connections can be made directly to the array. Boot from SAN is supported for both switched topology and direct connect topology. |
| Hardware- specific considerations | If you are running an IBM eServer BladeCenter and use boot from SAN, you must disable IDE drives on the blades. |

Getting Ready for Boot from SAN

When you prepare your ESXi host to boot from a SAN, you perform several tasks.

This section describes the generic boot-from-SAN enablement process on the rack-mounted servers. For information on enabling the boot from SAN option on Cisco Unified Computing System FCoE blade servers, refer to Cisco documentation.

What to read next

Procedure

1 [Configure SAN Components and Storage System](#)

Before you set up your ESXi host to boot from a SAN LUN, configure SAN components and a storage system.

2 Configure Storage Adapter to Boot from SAN

When you set up your host to boot from SAN, you enable the boot adapter in the host BIOS. You then configure the boot adapter to initiate a primitive connection to the target boot LUN.

3 Set Up Your System to Boot from Installation Media

When setting up your host to boot from SAN, you first boot the host from the VMware installation media. To boot from the installation media, change the system boot sequence in the BIOS setup.

Configure SAN Components and Storage System

Before you set up your ESXi host to boot from a SAN LUN, configure SAN components and a storage system.

Because configuring the SAN components is vendor-specific, refer to the product documentation for each item.

Procedure

- 1 Connect network cable, referring to any cabling guide that applies to your setup.
Check the switch wiring, if there is any.
- 2 Configure the storage array.
 - a From the SAN storage array, make the ESXi host visible to the SAN. This process is often called creating an object.
 - b From the SAN storage array, set up the host to have the WWPNs of the host's adapters as port names or node names.
 - c Create LUNs.
 - d Assign LUNs.
 - e Record the IP addresses of the switches and storage arrays.
 - f Record the WWPN for each SP.

Caution If you use a scripted installation process to install ESXi in boot from SAN mode, take special steps to avoid unintended data loss.

Configure Storage Adapter to Boot from SAN

When you set up your host to boot from SAN, you enable the boot adapter in the host BIOS. You then configure the boot adapter to initiate a primitive connection to the target boot LUN.

Prerequisites

Determine the WWPN for the storage adapter.

Procedure

- ◆ Configure the storage adapter to boot from SAN.

Because configuring boot adapters is vendor-specific, consult your vendor documentation.

Set Up Your System to Boot from Installation Media

When setting up your host to boot from SAN, you first boot the host from the VMware installation media. To boot from the installation media, change the system boot sequence in the BIOS setup.

Because changing the boot sequence in the BIOS is vendor-specific, refer to vendor documentation for instructions. The following procedure explains how to change the boot sequence on an IBM host.

Procedure

- 1 Power on your system and enter the system BIOS Configuration/Setup Utility.
- 2 Select **Startup Options** and press Enter.
- 3 Select **Startup Sequence Options** and press Enter.
- 4 Change the **First Startup Device** to [CD-ROM].

Results

You can now install ESXi.

Configure Emulex HBA to Boot from SAN

Configuring the Emulex HBA BIOS to boot from SAN includes enabling the BootBIOS prompt and enabling BIOS.

Procedure

- 1 [Enable the BootBIOS Prompt](#)

When you configure the Emulex HBA BIOS to boot ESXi from SAN, you need to enable the BootBIOS prompt.

- 2 [Enable the BIOS](#)

When you configure the Emulex HBA BIOS to boot ESXi from SAN, you must enable BIOS.

Enable the BootBIOS Prompt

When you configure the Emulex HBA BIOS to boot ESXi from SAN, you need to enable the BootBIOS prompt.

Procedure

- 1 Run `lputil`.

- 2 Select **3. Firmware Maintenance**.
- 3 Select an adapter.
- 4 Select **6. Boot BIOS Maintenance**.
- 5 Select **1. Enable Boot BIOS**.

Enable the BIOS

When you configure the Emulex HBA BIOS to boot ESXi from SAN, you must enable BIOS.

Procedure

- 1 Reboot the host.
- 2 To configure the adapter parameters, press ALT+E at the Emulex prompt and follow these steps.
 - a Select an adapter (with BIOS support).
 - b Select **2. Configure This Adapter's Parameters**.
 - c Select **1. Enable or Disable BIOS**.
 - d Select **1** to enable BIOS.
 - e Select **x** to exit and **Esc** to return to the previous menu.
- 3 To configure the boot device, follow these steps from the Emulex main menu.
 - a Select the same adapter.
 - b Select **1. Configure Boot Devices**.
 - c Select the location for the Boot Entry.
 - d Enter the two-digit boot device.
 - e Enter the two-digit (HEX) starting LUN (for example, **08**).
 - f Select the boot LUN.
 - g Select **1. WWPN**. (Boot this device using WWPN, not DID).
 - h Select **x** to exit and **Y** to reboot.
- 4 Boot into the system BIOS and move Emulex first in the boot controller sequence.
- 5 Reboot and install on a SAN LUN.

Configure QLogic HBA to Boot from SAN

This sample procedure explains how to configure the QLogic HBA to boot ESXi from SAN. The procedure involves enabling the QLogic HBA BIOS, enabling the selectable boot, and selecting the boot LUN.

Procedure

- 1 While booting the server, press **Ctrl+Q** to enter the Fast!UTIL configuration utility.
- 2 Perform the appropriate action depending on the number of HBAs.

| Option | Description |
|---------------|---|
| One HBA | If you have only one HBA, the Fast!UTIL Options page appears. Skip to Step 3 . |
| Multiple HBAs | <p>If you have more than one HBA, select the HBA manually.</p> <ol style="list-style-type: none"> a In the Select Host Adapter page, use the arrow keys to position the pointer on the appropriate HBA. b Press Enter. |

- 3 In the Fast!UTIL Options page, select **Configuration Settings** and press **Enter**.
- 4 In the Configuration Settings page, select **Adapter Settings** and press **Enter**.
- 5 Set the BIOS to search for SCSI devices.
 - a In the Host Adapter Settings page, select **Host Adapter BIOS**.
 - b Press **Enter** to toggle the value to **Enabled**.
 - c Press **Esc** to exit.
- 6 Enable the selectable boot.
 - a Select **Selectable Boot Settings** and press **Enter**.
 - b In the Selectable Boot Settings page, select **Selectable Boot**.
 - c Press **Enter** to toggle the value to **Enabled**.
- 7 Select the Boot Port Name entry in the list of storage processors (SPs) and press **Enter**.
The Select Fibre Channel Device page opens.
- 8 Select the specific SP and press **Enter**.

If you are using an active-passive storage array, the selected SP must be on the preferred (active) path to the boot LUN. If you are not sure which SP is on the active path, use your storage array management software to find out. The target IDs are created by the BIOS and might change with each reboot.

- 9 Perform the appropriate action depending on the number of LUNs attached to the SP.

| Option | Description |
|---------------|--|
| One LUN | The LUN is selected as the boot LUN. You do not need to enter the Select LUN page. |
| Multiple LUNs | The Select LUN page opens. Use the pointer to select the boot LUN, then press Enter . |

- 10 If any remaining storage processors show in the list, press **C** to clear the data.

- 11 Press **Esc** twice to exit and press **Enter** to save the setting.

Booting ESXi with Software FCoE



ESXi supports booting from FCoE capable network adapters.

Only NICs with partial FCoE offload support the boot capabilities with the software FCoE. If you use the NICs without FCoE offload, the software FCoE boot is not supported.

When you install and boot ESXi from an FCoE LUN, the host can use a VMware software FCoE adapter and a network adapter with FCoE capabilities. The host does not require a dedicated FCoE HBA.

You perform most configurations through the `option ROM` of your network adapter. The network adapters must support one of the following formats, which communicate parameters about an FCoE boot device to VMkernel.

- FCoE Boot Firmware Table (FBFT). FBFT is Intel propriety.
- FCoE Boot Parameter Table (FBPT). FBPT is defined by VMware for third-party vendors to implement a software FCoE boot.

The configuration parameters are set in the `option ROM` of your adapter. During an ESXi installation or a subsequent boot, these parameters are exported in to system memory in either FBFT format or FBPT format. The VMkernel can read the configuration settings and use them to access the boot LUN.

Read the following topics next:

- [Requirements and Considerations for Software FCoE Boot](#)
- [Set Up Software FCoE Boot](#)
- [Troubleshooting Boot from Software FCoE for an ESXi Host](#)

Requirements and Considerations for Software FCoE Boot

When you boot the ESXi host from SAN using software FCoE, certain requirements and considerations apply.

Requirements

- Use ESXi of a compatible version.

- The network adapter must have the following capabilities:
 - Be FCoE capable.
 - Support ESXi open FCoE stack.
 - Contain FCoE boot firmware which can export boot information in FBFT format or FBPT format.

Considerations

- You cannot change software FCoE boot configuration from within ESXi.
- Coredump is not supported on any software FCoE LUNs, including the boot LUN.
- Multipathing is not supported at pre-boot.
- Boot LUN cannot be shared with other hosts even on shared storage. Make sure that the host has access to the entire boot LUN.

Set Up Software FCoE Boot

Your ESXi host can boot from a FCoE LUN using the software FCoE adapter and a network adapter.

When you configure your host for a software FCoE boot, you perform several tasks.

Prerequisites

The network adapter has the following capabilities:

- Support partial FCoE offloads (software FCoE).
- Contain either a FCoE Boot Firmware Table (FBFT) or a FCoE Boot Parameter Table (FBPT).

For information about network adapters that support software FCoE boot, see the *VMware Compatibility Guide*.

Procedure

1 [Configure Software FCoE Boot Parameters](#)

To support a software FCoE boot process, a network adapter on your host must have a specially configured FCoE boot firmware. When you configure the firmware, you enable the adapter for the software FCoE boot and specify the boot LUN parameters.

2 [Install and Boot ESXi from Software FCoE LUN](#)

When you set up your system to boot from a software FCoE LUN, you install the ESXi image to the target LUN. You can then boot your host from that LUN.

Configure Software FCoE Boot Parameters

To support a software FCoE boot process, a network adapter on your host must have a specially configured FCoE boot firmware. When you configure the firmware, you enable the adapter for the software FCoE boot and specify the boot LUN parameters.

Procedure

- ◆ In the option ROM of the network adapter, specify software FCoE boot parameters.

These parameters include a boot target, boot LUN, VLAN ID, and so on.

Because configuring the network adapter is vendor-specific, review your vendor documentation for instructions.

Install and Boot ESXi from Software FCoE LUN

When you set up your system to boot from a software FCoE LUN, you install the ESXi image to the target LUN. You can then boot your host from that LUN.

Prerequisites

- Configure the `option ROM` of the network adapter, so that it points to a target boot LUN. Make sure that you have information about the bootable LUN.
- Change the boot order in the system BIOS to the following sequence:
 - a The network adapter that you use for the software FCoE boot.
 - b The ESXi installation media.

See the vendor documentation for your system.

Procedure

- 1 Start an interactive installation from the ESXi installation CD/DVD.

The ESXi installer verifies that FCoE boot is enabled in the BIOS and, if needed, creates a standard virtual switch for the FCoE capable network adapter. The name of the vSwitch is `VMware_FCoE_vSwitch`. The installer then uses preconfigured FCoE boot parameters to discover and display all available FCoE LUNs.

- 2 On the **Select a Disk** page, select the software FCoE LUN that you specified in the boot parameter setting.

If the boot LUN does not appear in this menu, make sure that you correctly configured boot parameters in the `option ROM` of the network adapter.

- 3 Complete the installation by following the prompts.
- 4 Reboot the host.

- 5 Change the boot order in the system BIOS so that the FCoE boot LUN is the first bootable device.

ESXi continues booting from the software FCoE LUN until it is ready to be used.

What to do next

If needed, you can rename and modify the `VMware_FCoE_vSwitch` that the installer automatically created. Make sure that the Cisco Discovery Protocol (CDP) mode is set to Listen or Both.

Troubleshooting Boot from Software FCoE for an ESXi Host

If the installation or boot of ESXi from a software FCoE LUN fails, you can use several troubleshooting methods.

Problem

When you install or boot ESXi from FCoE storage, the installation or the boot process fails. The FCoE setup that you use includes a VMware software FCoE adapter and a network adapter with partial FCoE offload capabilities.

Solution

- Make sure that you correctly configured boot parameters in the option ROM of the FCoE network adapter.
- During installation, monitor the BIOS of the FCoE network adapter for any errors.
- If possible, check the VMkernel log for errors.
- Use the `esxcli` command to verify whether the boot LUN is present.

```
esxcli conn_options hardware bootdevice list
```

Best Practices for Fibre Channel Storage

9

When using ESXi with Fibre Channel SAN, follow recommendations to avoid performance problems.

The vSphere Client offers extensive facilities for collecting performance information. The information is graphically displayed and frequently updated.

You can also use the `resxtop` or `esxtop` command-line utilities. The utilities provide a detailed look at how ESXi uses resources. For more information, see the *vSphere Resource Management* documentation.

Check with your storage representative if your storage system supports Storage API - Array Integration hardware acceleration features. If it does, refer to your vendor documentation to enable hardware acceleration support on the storage system side. For more information, see [Chapter 24 Storage Hardware Acceleration](#).

Read the following topics next:

- [Preventing Fibre Channel SAN Problems](#)
- [Disable Automatic Host Registration](#)
- [Optimizing Fibre Channel SAN Storage Performance](#)

Preventing Fibre Channel SAN Problems

When you use ESXi with a Fibre Channel SAN, follow specific guidelines to avoid SAN problems.

To prevent problems with your SAN configuration, observe these tips:

- Place only one VMFS datastore on each LUN.
- Do not change the path policy the system sets for you unless you understand the implications of making such a change.
- Document everything. Include information about zoning, access control, storage, switch, server and FC HBA configuration, software and firmware versions, and storage cable plan.
- Plan for failure:
 - Make several copies of your topology maps. For each element, consider what happens to your SAN if the element fails.

- Verify different links, switches, HBAs, and other elements to ensure that you did not miss a critical failure point in your design.
- Ensure that the Fibre Channel HBAs are installed in the correct slots in the host, based on slot and bus speed. Balance PCI bus load among the available buses in the server.
- Become familiar with the various monitor points in your storage network, at all visibility points, including host's performance charts, FC switch statistics, and storage performance statistics.
- Be cautious when changing IDs of the LUNs that have VMFS datastores being used by your ESXi host. If you change the ID, the datastore becomes inactive and its virtual machines fail. Resignature the datastore to make it active again. See [Managing Duplicate VMFS Datastores](#).

After you change the ID of the LUN, rescan the storage to reset the ID on your host. For information on using the rescan, see [Storage Rescan Operations](#).

Disable Automatic Host Registration

Certain storage arrays require that ESXi hosts register with the arrays. ESXi performs automatic host registration by sending the host's name and IP address to the array. If you prefer to perform manual registration using storage management software, disable the ESXi auto-registration feature.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **System**, click **Advanced System Settings**.
- 4 Under Advanced System Settings, select the **Disk.EnableNaviReg** parameter and click the **Edit** icon.
- 5 Change the value to 0.

Results

This operation disables the automatic host registration that is enabled by default.

Optimizing Fibre Channel SAN Storage Performance

Several factors contribute to optimizing a typical SAN environment.

If the environment is properly configured, the SAN fabric components (particularly the SAN switches) are only minor contributors because of their low latencies relative to servers and storage arrays. Make sure that the paths through the switch fabric are not saturated, that is, that the switch fabric is running at the highest throughput.

Storage Array Performance

Storage array performance is one of the major factors contributing to the performance of the entire SAN environment.

If you encounter any problems with storage array performance, consult your storage array vendor documentation for any relevant information.

To improve the array performance in the vSphere environment, follow these general guidelines:

- When assigning LUNs, remember that several hosts might access the LUN, and that several virtual machines can run on each host. One LUN used by a host can service I/O from many different applications running on different operating systems. Because of this diverse workload, the RAID group containing the ESXi LUNs typically does not include LUNs used by other servers that are not running ESXi.
- Make sure that the read/write caching is available.
- SAN storage arrays require continual redesign and tuning to ensure that I/O is load-balanced across all storage array paths. To meet this requirement, distribute the paths to the LUNs among all the SPs to provide optimal load-balancing. Close monitoring indicates when it is necessary to rebalance the LUN distribution.

Tuning statically balanced storage arrays is a matter of monitoring the specific performance statistics, such as I/O operations per second, blocks per second, and response time. Distributing the LUN workload to spread the workload across all the SPs is also important.

Note Dynamic load-balancing is not currently supported with ESXi.

Server Performance with Fibre Channel

You must consider several factors to ensure optimal server performance.

Each server application must have access to its designated storage with the following conditions:

- High I/O rate (number of I/O operations per second)
- High throughput (megabytes per second)
- Minimal latency (response times)

Because each application has different requirements, you can meet these goals by selecting an appropriate RAID group on the storage array.

To achieve performance goals, follow these guidelines:

- Place each LUN on a RAID group that provides the necessary performance levels. Monitor the activities and resource use of other LUNs in the assigned RAID group. A high-performance RAID group that has too many applications doing I/O to it might not meet performance goals required by an application running on the ESXi host.

- Ensure that each host has enough HBAs to increase throughput for the applications on the host for the peak period. I/O spread across multiple HBAs provides faster throughput and less latency for each application.
- To provide redundancy for a potential HBA failure, make sure that the host is connected to a dual redundant fabric.
- When allocating LUNs or RAID groups for ESXi systems, remember that multiple operating systems use and share that resource. The LUN performance required by the ESXi host might be much higher than when you use regular physical machines. For example, if you expect to run four I/O intensive applications, allocate four times the performance capacity for the ESXi LUNs.
- When you use multiple ESXi systems in with vCenter Server, the performance requirements for the storage subsystem increase correspondingly.
- The number of outstanding I/Os needed by applications running on the ESXi system must match the number of I/Os the HBA and storage array can handle.

Using ESXi with iSCSI SAN

10

ESXi can connect to external SAN storage using the Internet SCSI (iSCSI) protocol. In addition to traditional iSCSI, ESXi also supports iSCSI Extensions for RDMA (iSER).

When the iSER protocol is enabled, the host can use the same iSCSI framework, but replaces the TCP/IP transport with the Remote Direct Memory Access (RDMA) transport.

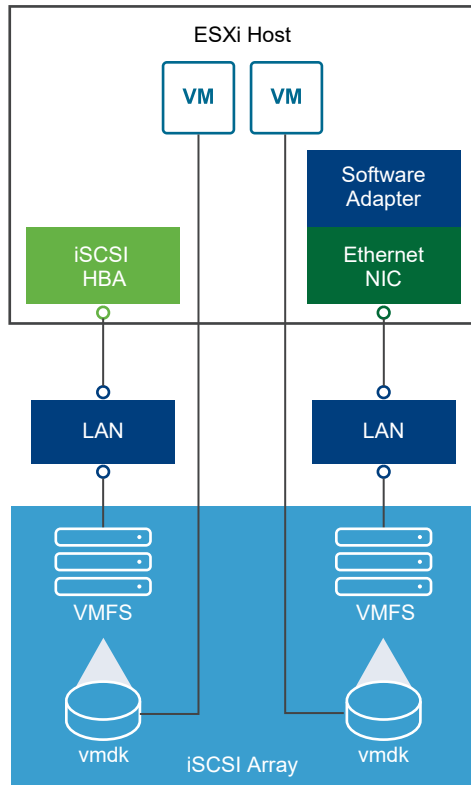
Read the following topics next:

- [About iSCSI SAN](#)

About iSCSI SAN

iSCSI SANs use Ethernet connections between hosts and high-performance storage subsystems.

On the host side, the iSCSI SAN components include iSCSI host bus adapters (HBAs) or Network Interface Cards (NICs). The iSCSI network also includes switches and routers that transport the storage traffic, cables, storage processors (SPs), and storage disk systems.



The iSCSI SAN uses a client-server architecture.

The client, called iSCSI initiator, operates on your ESXi host. It initiates iSCSI sessions by issuing SCSI commands and transmitting them, encapsulated into the iSCSI protocol, to an iSCSI server. The server is known as an iSCSI target. Typically, the iSCSI target represents a physical storage system on the network.

The target can also be a virtual iSCSI SAN, for example, an iSCSI target emulator running in a virtual machine. The iSCSI target responds to the initiator's commands by transmitting required iSCSI data.

iSCSI Multipathing

When transferring data between the host server and storage, the SAN uses a technique known as multipathing. With multipathing, your ESXi host can have more than one physical path to a LUN on a storage system.

Generally, a single path from a host to a LUN consists of an iSCSI adapter or NIC, switch ports, connecting cables, and the storage controller port. If any component of the path fails, the host selects another available path for I/O. The process of detecting a failed path and switching to another is called path failover.

For more information on multipathing, see [Chapter 18 Understanding Multipathing and Failover](#).

Nodes and Ports in the iSCSI SAN

A single discoverable entity on the iSCSI SAN, such as an initiator or a target, represents an iSCSI node.

Each node has a node name. ESXi uses several methods to identify a node.

IP Address

Each iSCSI node can have an IP address associated with it so that routing and switching equipment on your network can establish the connection between the host and storage. This address is like the IP address that you assign to your computer to get access to your company's network or the Internet.

iSCSI Name

A worldwide unique name for identifying the node. iSCSI uses the iSCSI Qualified Name (IQN) and Extended Unique Identifier (EUI).

By default, ESXi generates unique iSCSI names for your iSCSI initiators, for example, `iqn.1998-01.com.vmware:iscsitestox-68158ef2`. Usually, you do not have to change the default value, but if you do, make sure that the new iSCSI name you enter is worldwide unique.

iSCSI Alias

A more manageable name for an iSCSI device or port used instead of the iSCSI name. iSCSI aliases are not unique and are intended to be a friendly name to associate with a port.

Each node has one or more ports that connect it to the SAN. iSCSI ports are end-points of an iSCSI session.

iSCSI Naming Conventions

iSCSI uses a special unique name to identify an iSCSI node, either target or initiator.

iSCSI names are formatted in two different ways. The most common is the IQN format.

For more details on iSCSI naming requirements and string profiles, see RFC 3721 and RFC 3722 on the IETF website.

iSCSI Qualified Name Format

The iSCSI Qualified Name (IQN) format takes the form `iqn.yyyy-mm.naming-authority:unique name`, where:

- `yyyy-mm` is the year and month when the naming authority was established.
- `naming-authority` is the reverse syntax of the Internet domain name of the naming authority. For example, the `iscsi.vmware.com` naming authority can have the iSCSI qualified name form of `iqn.1998-01.com.vmware.iscsi`. The name indicates that the `vmware.com` domain name was registered in January of 1998, and `iscsi` is a subdomain, maintained by `vmware.com`.

- *unique name* is any name you want to use, for example, the name of your host. The naming authority must make sure that any names assigned following the colon are unique, such as:
 - `iqn.1998-01.com.vmware.iscsi:name1`
 - `iqn.1998-01.com.vmware.iscsi:name2`
 - `iqn.1998-01.com.vmware.iscsi:name999`

Enterprise Unique Identifier Format

The Enterprise Unique Identifier (EUI) format takes the form `eui.16_hex_digits`.

For example, `eui.0123456789ABCDEF`.

The 16-hexadecimal digits are text representations of a 64-bit number of an IEEE EUI (extended unique identifier) format. The top 24 bits are a company ID that IEEE registers with a particular company. The remaining 40 bits are assigned by the entity holding that company ID and must be unique.

iSCSI Initiators

To access iSCSI targets, your ESXi host uses iSCSI initiators.

The initiator is a software or hardware installed on your ESXi host. The iSCSI initiator originates communication between your host and an external iSCSI storage system and sends data to the storage system.

In the ESXi environment, iSCSI adapters configured on your host play the role of initiators. ESXi supports several types of iSCSI adapters.

For information on configuring and using iSCSI adapters, see [Chapter 11 Configuring iSCSI Adapters and Storage](#).

Software iSCSI Adapter

A software iSCSI adapter is a VMware code built into the VMkernel. Using the software iSCSI adapter, your host can connect to the iSCSI storage device through standard network adapters. The software iSCSI adapter handles iSCSI processing while communicating with the network adapter. With the software iSCSI adapter, you can use iSCSI technology without purchasing specialized hardware.

Hardware iSCSI Adapter

A hardware iSCSI adapter is a third-party adapter that offloads iSCSI and network processing from your host. Hardware iSCSI adapters are divided into categories.

Dependent Hardware iSCSI Adapter

Depends on VMware networking, and iSCSI configuration and management interfaces provided by VMware.

This type of adapter can be a card that presents a standard network adapter and iSCSI offload functionality for the same port. The iSCSI offload functionality depends on the host's network configuration to obtain the IP, MAC, and other parameters used for iSCSI sessions. An example of a dependent adapter is the iSCSI licensed Broadcom 5709 NIC.

Independent Hardware iSCSI Adapter

Implements its own networking and iSCSI configuration and management interfaces.

Typically, an independent hardware iSCSI adapter is a card that either presents only iSCSI offload functionality or iSCSI offload functionality and standard NIC functionality. The iSCSI offload functionality has independent configuration management that assigns the IP, MAC, and other parameters used for the iSCSI sessions. An example of an independent adapter is the QLogic QLA4052 adapter.

Hardware iSCSI adapters might need to be licensed. Otherwise, they might not appear in the client or vSphere CLI. Contact your vendor for licensing information.

About the VMware iSER Adapter

In addition to traditional iSCSI, ESXi supports the iSCSI Extensions for RDMA (iSER) protocol. When the iSER protocol is enabled, the iSCSI framework on the ESXi host can use the Remote Direct Memory Access (RDMA) transport instead of TCP/IP.

The traditional iSCSI protocol carries SCSI commands over a TCP/IP network between an iSCSI initiator on a host and an iSCSI target on a storage device. The iSCSI protocol encapsulates the commands and assembles that data in packets for the TCP/IP layer. When the data arrives, the iSCSI protocol disassembles the TCP/IP packets, so that the SCSI commands can be differentiated and delivered to the storage device.

iSER differs from traditional iSCSI as it replaces the TCP/IP data transfer model with the Remote Direct Memory Access (RDMA) transport. Using the direct data placement technology of the RDMA, the iSER protocol can transfer data directly between the memory buffers of the ESXi host and storage devices. This method eliminates unnecessary TCP/IP processing and data copying, and can also reduce latency and the CPU load on the storage device.

In the iSER environment, iSCSI works exactly as before, but uses an underlying RDMA fabric interface instead of the TCP/IP-based interface.

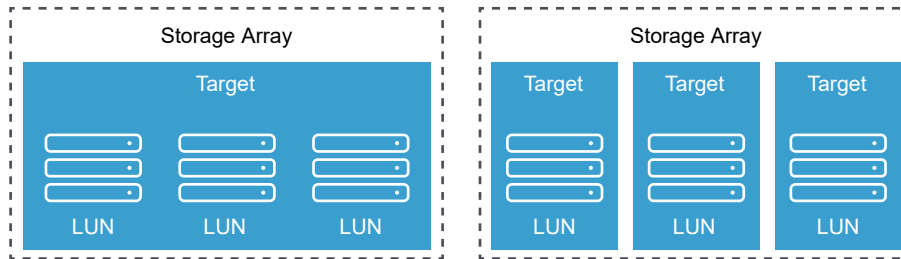
Because the iSER protocol preserves the compatibility with iSCSI infrastructure, the process of enabling iSER on the ESXi host is similar to the iSCSI process. See [Configure iSER Adapters](#).

Establishing iSCSI Connections

In the ESXi context, the term target identifies a single storage unit that your host can access. The terms storage device and LUN describe a logical volume that represents storage space on a target. Typically, the terms device and LUN, in the ESXi context, mean a SCSI volume presented to your host from a storage target and available for formatting.

Different iSCSI storage vendors present storage to hosts in different ways. Some vendors present multiple LUNs on a single target. Others present multiple targets with one LUN each.

Figure 10-1. Target Compared to LUN Representations



In these examples, three LUNs are available in each of these configurations. In the first case, the host detects one target but that target has three LUNs that can be used. Each of the LUNs represents individual storage volume. In the second case, the host detects three different targets, each having one LUN.

Host-based iSCSI initiators establish connections to each target. Storage systems with a single target containing multiple LUNs have traffic to all the LUNs on a single connection. With a system that has three targets with one LUN each, the host uses separate connections to the three LUNs.

This information is useful when you are trying to aggregate storage traffic on multiple connections from the host with multiple iSCSI adapters. You can set the traffic for one target to a particular adapter, and use a different adapter for the traffic to another target.

iSCSI Storage System Types

ESXi supports different storage systems and arrays.

The types of storage that your host supports include active-active, active-passive, and ALUA-compliant.

Active-active storage system

Supports access to the LUNs simultaneously through all the storage ports that are available without significant performance degradation. All the paths are always active, unless a path fails.

Active-passive storage system

A system in which one storage processor is actively providing access to a given LUN. The other processors act as a backup for the LUN and can be actively providing access to other LUN I/O. I/O can be successfully sent only to an active port for a given LUN. If access through the active storage port fails, one of the passive storage processors can be activated by the servers accessing it.

Asymmetrical storage system

Supports Asymmetric Logical Unit Access (ALUA). ALUA-compliant storage systems provide different levels of access per port. With ALUA, hosts can determine the states of target

ports and prioritize paths. The host uses some of the active paths as primary and others as secondary.

Virtual port storage system

Supports access to all available LUNs through a single virtual port. Virtual port storage systems are active-active storage devices, but hide their multiple connections through a single port. ESXi multipathing does not make multiple connections from a specific port to the storage by default. Some storage vendors supply session managers to establish and manage multiple connections to their storage. These storage systems handle port failovers and connection balancing transparently. This capability is often called transparent failover.

Discovery, Authentication, and Access Control

You can use several mechanisms to discover your storage and to limit access to it.

You must configure your host and the iSCSI storage system to support your storage access control policy.

Discovery

A discovery session is part of the iSCSI protocol. It returns the set of targets you can access on an iSCSI storage system. The two types of discovery available on ESXi are dynamic and static. Dynamic discovery obtains a list of accessible targets from the iSCSI storage system. Static discovery can access only a particular target by target name and address.

For more information, see [Configuring Discovery Addresses for iSCSI Adapters](#).

Authentication

iSCSI storage systems authenticate an initiator by a name and key pair. ESXi supports the CHAP authentication protocol. To use CHAP authentication, the ESXi host and the iSCSI storage system must have CHAP enabled and have common credentials.

For information on enabling CHAP, see [Configuring CHAP Parameters for iSCSI Adapters](#).

Access Control

Access control is a policy set up on the iSCSI storage system. Most implementations support one or more of three types of access control:

- By initiator name
- By IP address
- By the CHAP protocol

Only initiators that meet all rules can access the iSCSI volume.

Using only CHAP for access control can slow down rescans because the ESXi host can discover all targets, but then fails at the authentication step. iSCSI rescans work faster if the host discovers only the targets it can authenticate.

Error Correction

To protect the integrity of iSCSI headers and data, the iSCSI protocol defines error correction methods known as header digests and data digests.

Both parameters are disabled by default, but you can enable them. These digests pertain to, respectively, the header and SCSI data being transferred between iSCSI initiators and targets, in both directions.

Header and data digests check the noncryptographic data integrity beyond the integrity checks that other networking layers provide, such as TCP and Ethernet. They check the entire communication path, including all elements that can change the network-level traffic, such as routers, switches, and proxies.

The existence and type of the digests are negotiated when an iSCSI connection is established. When the initiator and target agree on a digest configuration, this digest must be used for all traffic between them.

Enabling header and data digests does require additional processing for both the initiator and the target and can affect throughput and CPU use performance.

Note Systems that use the Intel Nehalem processors offload the iSCSI digest calculations, as a result, reducing the impact on performance.

For information on enabling header and data digests, see [Configuring Advanced Parameters for iSCSI](#).

How Virtual Machines Access Data on an iSCSI SAN

ESXi stores a virtual machine's disk files within a VMFS datastore that resides on a SAN storage device. When virtual machine guest operating systems send SCSI commands to their virtual disks, the SCSI virtualization layer translates these commands to VMFS file operations.

When a virtual machine interacts with its virtual disk stored on a SAN, the following process takes place:

- 1 When the guest operating system in a virtual machine reads or writes to SCSI disk, it sends SCSI commands to the virtual disk.
- 2 Device drivers in the virtual machine's operating system communicate with the virtual SCSI controllers.
- 3 The virtual SCSI controller forwards the commands to the VMkernel.
- 4 The VMkernel performs the following tasks.
 - a Locates an appropriate virtual disk file in the VMFS volume.
 - b Maps the requests for the blocks on the virtual disk to blocks on the appropriate physical device.
 - c Sends the modified I/O request from the device driver in the VMkernel to the iSCSI initiator, hardware or software.

- 5 If the iSCSI initiator is a hardware iSCSI adapter, independent or dependent, the adapter performs the following tasks.
 - a Encapsulates I/O requests into iSCSI Protocol Data Units (PDUs).
 - b Encapsulates iSCSI PDUs into TCP/IP packets.
 - c Sends IP packets over Ethernet to the iSCSI storage system.
- 6 If the iSCSI initiator is a software iSCSI adapter, the following takes place.
 - a The iSCSI initiator encapsulates I/O requests into iSCSI PDUs.
 - b The initiator sends iSCSI PDUs through TCP/IP connections.
 - c The VMkernel TCP/IP stack relays TCP/IP packets to a physical NIC.
 - d The physical NIC sends IP packets over Ethernet to the iSCSI storage system.
- 7 Ethernet switches and routers on the network carry the request to the appropriate storage device.

Configuring iSCSI Adapters and Storage

11

Before ESXi can work with iSCSI SAN, you must set up your iSCSI environment.

The process of preparing your iSCSI environment involves the following steps:

| Step | Details |
|-------------------------------------|--|
| Set up iSCSI storage | For information, see your storage vendor documentation. In addition, follow these recommendations: <ul style="list-style-type: none">■ ESXi iSCSI SAN Recommendations and Restrictions■ Chapter 13 Best Practices for iSCSI Storage |
| Configure iSCSI adapters | Use an appropriate workflow to configure your adapter: <ul style="list-style-type: none">■ Set Up Independent Hardware iSCSI Adapters■ Configure Dependent Hardware iSCSI Adapters■ Configure the Software iSCSI Adapter■ Configure iSER Adapters |
| Create a datastore on iSCSI storage | Creating Datastores |

Read the following topics next:

- [ESXi iSCSI SAN Recommendations and Restrictions](#)
- [Configuring iSCSI Parameters for Adapters](#)
- [Set Up Independent Hardware iSCSI Adapters](#)
- [Configure Dependent Hardware iSCSI Adapters](#)
- [Configure the Software iSCSI Adapter](#)
- [Configure iSER Adapters](#)
- [Modify General Properties for iSCSI or iSER Adapters](#)
- [Setting Up Network for iSCSI and iSER](#)
- [Using Jumbo Frames with iSCSI](#)
- [Configuring Discovery Addresses for iSCSI Adapters](#)
- [Configuring CHAP Parameters for iSCSI Adapters](#)
- [Configuring Advanced Parameters for iSCSI](#)

- [iSCSI Session Management](#)

ESXi iSCSI SAN Recommendations and Restrictions

To work properly with iSCSI SAN, your ESXi environment must follow specific recommendations. In addition, several restrictions exist when you use ESXi with iSCSI SAN.

For additional information about using vSphere with iSCSI, see [Best Practices For Running VMware vSphere On iSCSI](#) blog.

iSCSI Storage Recommendations

- Verify that your ESXi host supports the iSCSI SAN storage hardware and firmware. For an up-to-date list, see *VMware Compatibility Guide*.
- To ensure that the host recognizes LUNs at startup time, configure all iSCSI storage targets so that your host can access them and use them. Configure your host so that it can discover all available iSCSI targets.
- Unless you are using diskless servers, set up a diagnostic partition on local storage. If you have diskless servers that boot from iSCSI SAN, see [General Recommendations for Boot from iSCSI SAN](#) for information about diagnostic partitions with iSCSI.
- Set the SCSI controller driver in the guest operating system to a large enough queue.
- VMtools increases the default disk timeout SCSI `TimeoutValue` parameter to 180 seconds. On virtual machines without VMtools, increase the value of the SCSI `TimeoutValue` parameter. With this parameter set up, the VMs can better tolerate delayed I/O resulting from a path failover. For information, see [Set Timeout on Windows Guest OS](#) or the [KB 1009465](#).
- Configure your environment to have only one VMFS datastore for each LUN.

iSCSI Storage Restrictions

- ESXi does not support iSCSI-connected tape devices.
- You cannot use virtual-machine multipathing software to perform I/O load balancing to a single physical LUN.
- ESXi does not support multipathing when you combine independent hardware adapters with either software or dependent hardware adapters.

Configuring iSCSI Parameters for Adapters

Before your ESXi host can discover iSCSI storage, you must configure your iSCSI adapters. When you configure the adapters, you set several iSCSI parameters.

iSCSI Networking

For certain types of iSCSI adapters, you must configure VMkernel networking.

You can verify the network configuration by using the `vmkping` utility.

The independent hardware iSCSI adapter does not require VMkernel networking. You can configure network parameters, such as an IP address, subnet mask, and default gateway on the independent hardware iSCSI adapter.

All types of iSCSI adapters support IPv4 and IPv6 protocols.

| iSCSI Adapter (vmhba) | Description | VMkernel Networking | Adapter Network Settings |
|------------------------------------|--|---|---|
| Independent Hardware iSCSI Adapter | Third-party adapter that offloads the iSCSI and network processing and management from your host. | Not required. | For information, see Edit Network Settings for Hardware iSCSI . |
| Software iSCSI Adapter | Uses standard NICs to connect your host to a remote iSCSI target on the IP network. | Required. For information, see Setting Up Network for iSCSI and iSER . | N/A |
| Dependent Hardware iSCSI Adapter | Third-party adapter that depends on VMware networking and iSCSI configuration and management interfaces. | Required For information, see Setting Up Network for iSCSI and iSER . | N/A |
| VMware iSER Adapter | Uses an RDMA-capable network adapter to connect your host to a remote iSCSI target. | Required For information, see Setting Up Network for iSCSI and iSER . | N/A |

Discovery Methods

For all types of iSCSI adapters, you must set the dynamic discovery address or static discovery address. In addition, you must provide a target name of the storage system. For software iSCSI and dependent hardware iSCSI, the address is pingable using `vmkping`.

See [Configuring Discovery Addresses for iSCSI Adapters](#).

CHAP Authentication

Enable the CHAP parameter on the initiator and the storage system side. After authentication is enabled, it applies to all targets that are not yet discovered. It does not apply to targets that are already discovered.

See [Configuring CHAP Parameters for iSCSI Adapters](#).

Set Up Independent Hardware iSCSI Adapters

An independent hardware iSCSI adapter is a specialized third-party adapter capable of accessing iSCSI storage over TCP/IP. This iSCSI adapter handles all iSCSI and network processing and management for your ESXi system.

Prerequisites

- Verify whether the adapter must be licensed.
- Install the adapter on your ESXi host.

For information about licensing, installation, and firmware updates, see vendor documentation.

The process of setting up the independent hardware iSCSI adapter includes these steps.

| Step | Description |
|--|---|
| View Independent Hardware iSCSI Adapters | View an independent hardware iSCSI adapter and verify that it is correctly installed and ready for configuration. |
| Modify General Properties for iSCSI or iSER Adapters | If needed, change the default iSCSI name and alias assigned to your iSCSI adapters. For the independent hardware iSCSI adapters, you can also change the default IP settings. |
| Edit Network Settings for Hardware iSCSI | Change default network settings so that the adapter is configured properly for the iSCSI SAN. |
| Configure Dynamic or Static Discovery for iSCSI and iSER | Set up dynamic discovery. With dynamic discovery, each time the initiator contacts a specified iSCSI storage system, it sends the SendTargets request to the system. The iSCSI system responds by supplying a list of available targets to the initiator. In addition to the dynamic discovery method, you can use static discovery and manually enter information for the targets. |
| Set Up CHAP for iSCSI or iSER Adapter | If your iSCSI environment uses the Challenge Handshake Authentication Protocol (CHAP), configure it for your adapter. |
| Enable Jumbo Frames for Independent Hardware iSCSI | If your iSCSI environment supports Jumbo Frames, enable them for the adapter. |

View Independent Hardware iSCSI Adapters

View an independent hardware iSCSI adapter and verify that it is correctly installed and ready for configuration.

After you install an independent hardware iSCSI adapter on a host, it appears on the list of storage adapters available for configuration. You can view its properties.

Prerequisites

Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Adapters**.

If installed, the hardware iSCSI adapter appears on the list of storage adapters.

4 Select the adapter to view.

The default details for the adapter appear.

| Adapter Information | Description |
|---------------------|---|
| Model | Model of the adapter. |
| iSCSI Name | Unique name formed according to iSCSI standards that identifies the iSCSI adapter. You can edit the iSCSI name. |
| iSCSI Alias | A friendly name used instead of the iSCSI name. You can edit the iSCSI alias. |
| IP Address | Address assigned to the iSCSI HBA. |
| Targets | Number of targets accessed through the adapter. |
| Devices | All storage devices or LUNs the adapter can access. |
| Paths | All paths the adapter uses to access storage devices. |

Edit Network Settings for Hardware iSCSI

After you install an independent hardware iSCSI adapter, you might need to change its default network settings so that the adapter is configured properly for the iSCSI SAN.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Adapters**, and select the adapter (vmhba#) to configure.
- 4 Click the **Network Settings** tab and click **Edit**.
- 5 In the IPv4 settings section, disable IPv6 or select the method to obtain IP addresses.

Note The automatic DHCP option and static option are mutually exclusive.

| Option | Description |
|------------------------------------|--|
| No IPv4 settings | Disable IPv4. |
| Obtain IPv4 settings automatically | Use DHCP to obtain IP settings. |
| Use static IPv4 settings | Enter the IPv4 IP address, subnet mask, and default gateway for the iSCSI adapter. |

- 6 In the IPv6 settings section, disable IPv6 or select an appropriate option for obtaining IPv6 addresses.

Note Automatic options and the static option are mutually exclusive.

| Option | Description |
|--|--|
| No IPv6 settings | Disable IPv6. |
| Enable IPv6 | Select an option for obtaining IPv6 addresses. |
| Obtain IPv6 addresses automatically through DHCP | Use DHCP to obtain IPv6 addresses. |
| Obtain IPv6 addresses automatically through Router Advertisement | Use router advertisement to obtain IPv6 addresses. |
| Override Link-local address for IPv6 | Override the link-local IP address by configuring a static IP address. |
| Static IPv6 addresses | <ol style="list-style-type: none"> a Click Add to add a new IPv6 address. b Enter the IPv6 address and subnet prefix length, and click OK. |

- 7 In the DNS settings section, provide IP addresses for a preferred DNS server and an alternate DNS server.

You must provide both values.

Configure Dependent Hardware iSCSI Adapters

A dependent hardware iSCSI adapter is a third-party adapter that depends on VMware networking, and iSCSI configuration and management interfaces provided by VMware.

An example of a dependent iSCSI adapter is a Broadcom 5709 NIC. When installed on a host, it presents its two components, a standard network adapter and an iSCSI engine, to the same port. The iSCSI engine appears on the list of storage adapters as an iSCSI adapter (vmhba).

The iSCSI adapter is enabled by default. To make it functional, you must connect it, through a virtual VMkernel adapter (vmk), to a physical network adapter (vmnic) associated with it. You can then configure the iSCSI adapter.

After you configure the dependent hardware iSCSI adapter, the discovery and authentication data is passed through the network connection. The iSCSI traffic goes through the iSCSI engine, bypassing the network.

The entire setup and configuration process for the dependent hardware iSCSI adapters involves several steps.

| Step | Description |
|--|--|
| View Dependent Hardware iSCSI Adapters | View a dependent hardware iSCSI adapter to verify that it is correctly loaded. |
| Modify General Properties for iSCSI or iSER Adapters | If needed, change the default iSCSI name and alias assigned to your adapter. |

| Step | Description |
|--|---|
| Determine Association Between iSCSI and Network Adapters | You must create network connections to bind dependent iSCSI and physical network adapters. To create the connections correctly, determine the name of the physical NIC with which the dependent hardware iSCSI adapter is associated. |
| Configure Port Binding for iSCSI or iSER | Configure connections for the traffic between the iSCSI component and the physical network adapters. The process of configuring these connections is called port binding. |
| Configure Dynamic or Static Discovery for iSCSI and iSER | Set up dynamic discovery. With dynamic discovery, each time the initiator contacts a specified iSCSI storage system, it sends the SendTargets request to the system. The iSCSI system responds by supplying a list of available targets to the initiator. In addition to the dynamic discovery method, you can use static discovery and manually enter information for the targets. |
| Set Up CHAP for iSCSI or iSER Adapter | If your iSCSI environment uses the Challenge Handshake Authentication Protocol (CHAP), configure it for your adapter. |
| Set Up CHAP for Target | You can also configure different CHAP credentials for each discovery address or static target. |
| Enable Jumbo Frames for Networking | If your iSCSI environment supports Jumbo Frames, enable them for the adapter. |

Dependent Hardware iSCSI Considerations

When you use dependent hardware iSCSI adapters with ESXi, certain considerations apply.

- When you use any dependent hardware iSCSI adapter, performance reporting for a NIC associated with the adapter might show little or no activity, even when iSCSI traffic is heavy. This behavior occurs because the iSCSI traffic bypasses the regular networking stack.
- If you use a third-party virtual switch, for example Cisco Nexus 1000V DVS, disable automatic pinning. Use manual pinning instead, making sure to connect a VMkernel adapter (vmk) to an appropriate physical NIC (vmnic). For information, refer to your virtual switch vendor documentation.
- The Broadcom iSCSI adapter performs data reassembly in hardware, which has a limited buffer space. When you use the Broadcom iSCSI adapter in a congested network or under heavy load, enable flow control to avoid performance degradation.

Flow control manages the rate of data transmission between two nodes to prevent a fast sender from overrunning a slow receiver. For best results, enable flow control at the end points of the I/O path, at the hosts and iSCSI storage systems.

To enable flow control for the host, use the `esxcli system module parameters` command. For details, see the VMware knowledge base article at <http://kb.vmware.com/kb/1013413>

- Dependent hardware adapters support IPv4 and IPv6.

View Dependent Hardware iSCSI Adapters

View a dependent hardware iSCSI adapter to verify that it is correctly loaded.

If installed, the dependent hardware iSCSI adapter (vmhba#) appears on the list of storage adapters under such category as, for example, Broadcom iSCSI Adapter. If the dependent hardware adapter does not appear on the list of storage adapters, check whether it needs to be licensed. See your vendor documentation.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Adapters**.
- 4 Select the adapter (vmhba#) to view.

The default details for the adapter appear, including the iSCSI name, iSCSI alias, and the status.

What to do next

Although the dependent iSCSI adapter is enabled by default, to make it functional, you must set up networking for the iSCSI traffic and bind the adapter to the appropriate VMkernel iSCSI port. You then configure discovery addresses and CHAP parameters.

Determine Association Between iSCSI and Network Adapters

You create network connections to bind dependent iSCSI and physical network adapters. To create the connections correctly, you must determine the name of the physical NIC with which the dependent hardware iSCSI adapter is associated.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Adapters**.
- 4 Select the iSCSI adapter (vmhba#) and click the **Network Port Binding** tab under adapter details.
- 5 Click **Add**.

The network adapter (vmnic#) that corresponds to the dependent iSCSI adapter is listed in the Physical Network Adapter column.

What to do next

If the VMkernel Adapter column is empty, create a VMkernel adapter (vmk#) for the physical network adapter (vmnic#) and then bind them to the associated dependent hardware iSCSI. See [Setting Up Network for iSCSI and iSER](#).

Configure the Software iSCSI Adapter

With the software-based iSCSI implementation, you can use standard NICs to connect your host to a remote iSCSI target on the IP network. The software iSCSI adapter that is built into ESXi facilitates this connection by communicating with the physical NICs through the network stack.

When you use the software iSCSI adapters, consider the following:

- Designate a separate network adapter for iSCSI. Do not use iSCSI on 100 Mbps or slower adapters.
- Avoid hard coding the name of the software adapter, `vmhbaXX`, in the scripts. It is possible for the name to change from one ESXi release to another. The change might cause failures of your existing scripts if they use the hardcoded old name. The name change does not affect the behavior of the iSCSI software adapter.

The process of configuring the software iSCSI adapter involves several steps.

| Step | Description |
|--|---|
| Activate or Disable the Software iSCSI Adapter | Activate your software iSCSI adapter so that your host can use it to access iSCSI storage. |
| Modify General Properties for iSCSI or iSER Adapters | If needed, change the default iSCSI name and alias assigned to your adapter. |
| Configure Port Binding for iSCSI or iSER | Configure connections for the traffic between the iSCSI component and the physical network adapters. The process of configuring these connections is called port binding. |
| Configure Dynamic or Static Discovery for iSCSI and iSER | Set up dynamic discovery. With dynamic discovery, each time the initiator contacts a specified iSCSI storage system, it sends the SendTargets request to the system. The iSCSI system responds by supplying a list of available targets to the initiator. In addition to the dynamic discovery method, you can use static discovery and manually enter information for the targets. |
| Set Up CHAP for iSCSI or iSER Adapter | If your iSCSI environment uses the Challenge Handshake Authentication Protocol (CHAP), configure it for your adapter. |
| Set Up CHAP for Target | You can also configure different CHAP credentials for each discovery address or static target. |
| Enable Jumbo Frames for Networking | If your iSCSI environment supports Jumbo Frames, enable them for the adapter. |

Activate or Disable the Software iSCSI Adapter

You must activate your software iSCSI adapter so that your host can use it to access iSCSI storage. If you do not need the software iSCSI adapter, you can disable it.

You can activate only one software iSCSI adapter.

Prerequisites

Required privilege: **Host.Configuration.Storage Partition Configuration**

Note If you boot from iSCSI using the software iSCSI adapter, the adapter is enabled and the network configuration is created at the first boot. If you disable the adapter, it is reenabled each time you boot the host.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Enable or disable the adapter.

| Option | Description |
|------------------------------------|---|
| Enable the software iSCSI adapter | <ol style="list-style-type: none"> a Under Storage, click Storage Adapters, and click the Add icon. b Select Software iSCSI Adapter and confirm that you want to add the adapter. <p>The software iSCSI adapter (vmhba#) is enabled and appears on the list of storage adapters. After enabling the adapter, the host assigns the default iSCSI name to it. You can now complete the adapter configuration.</p> |
| Disable the software iSCSI adapter | <ol style="list-style-type: none"> a Under Storage, click Storage Adapters, and select the adapter (vmhba#) to disable. b Click the Properties tab. c Click Disable and confirm that you want to disable the adapter. d Reboot the host. <p>The status indicates that the adapter is disabled.</p> <p>After the reboot, the adapter no longer appears on the list of storage adapters. The storage devices associated with the adapter become inaccessible. You can later activate the adapter.</p> |

Configure iSER Adapters

Configure iSER on your ESXi host, so that the iSCSI framework on the host can use the Remote Direct Memory Access (RDMA) transport instead of TCP/IP.

When installed on the host, the RDMA-capable adapter appears in vCenter Server as a network adapter (vmnic).

To make the adapter functional, you must enable the VMware iSER component, and then connect the iSER adapter to the RDMA-capable vmnic. You can then configure typical properties, such as targets and CHAP, for the iSER adapter.

The entire setup and configuration process for the iSER adapters involves several steps.

| Step | Description |
|--|---|
| Enable the VMware iSER Adapter | Use the <code>esxcli</code> command to enable the VMware iSER adapter. |
| Modify General Properties for iSCSI or iSER Adapters | If needed, change the default name and alias assigned to your adapter. |
| Configure Port Binding for iSCSI or iSER | <p>You must create network connections to bind the iSER engine and the RDMA-capable network adapter. The process of configuring these connections is called port binding.</p> <p>Note iSER does not support NIC teaming. When configuring port binding, use only one RDMA adapter per vSwitch.</p> |
| Configure Dynamic or Static Discovery for iSCSI and iSER | Set up the dynamic discovery. With the dynamic discovery, each time the initiator contacts a specified iSER storage system, it sends the SendTargets request to the system. The iSER system responds by supplying a list of available targets to the initiator. In addition to the dynamic discovery method, you can use the static discovery and manually enter information for the targets. |
| Set Up CHAP for iSCSI or iSER Adapter | If your environment uses the Challenge Handshake Authentication Protocol (CHAP), configure it for your adapter. |
| Set Up CHAP for Target | You can also configure different CHAP credentials for each discovery address or static target. |
| Enable Jumbo Frames for Networking | If your environment supports Jumbo Frames, enable them for the adapter. |

Enable the VMware iSER Adapter

Use the `esxcli` command to enable the VMware iSER adapter.

Prerequisites

- Make sure that your iSCSI storage supports the iSER protocol.
- Install the RDMA-capable adapter on your ESXi host.
- Use the RDMA-capable switch.
- Enable flow control on the ESXi host. To enable flow control for the host, use the `esxcli system module parameters` command. For details, see the VMware knowledge base article at <http://kb.vmware.com/kb/1013413>.
- Make sure to configure RDMA switch ports to create lossless connections between the iSER initiator and target.

Procedure

- 1 Use the ESXi Shell or vSphere CLI to enable the iSER adapter:

```
esxcli rdma iser add
```

2 Verify that the VMware iSER adapter has been added:

- a Navigate to the host.
- b Click the **Configure** tab.
- c Under **Storage**, click **Storage Adapters**, and review the list of adapters.

If you enabled the adapter, the adapter (vmhba#) appears on the list under the VMware iSCSI over RDMA (iSER) Adapter category.

Modify General Properties for iSCSI or iSER Adapters

You can change the default name and alias assigned to your iSCSI or iSER adapters. For the independent hardware iSCSI adapters, you can also change the default IP settings.

Important When you modify any default properties for your adapters, make sure to use correct formats for their names and IP addresses.

Prerequisites

Required privilege: **Host .Configuration.Storage Partition Configuration**

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Adapters**, and select the adapter (vmhba#) to configure.
- 4 Click the **Properties** tab, and click **Edit** in the General panel.
- 5 (Optional) Modify the following general properties.

| Option | Description |
|-------------|--|
| iSCSI Name | Unique name formed according to iSCSI standards that identifies the iSCSI adapter. If you change the name, make sure that the name you enter is worldwide unique and properly formatted. Otherwise, certain storage devices might not recognize the iSCSI adapter. |
| iSCSI Alias | A friendly name you use instead of the iSCSI name. |

Results

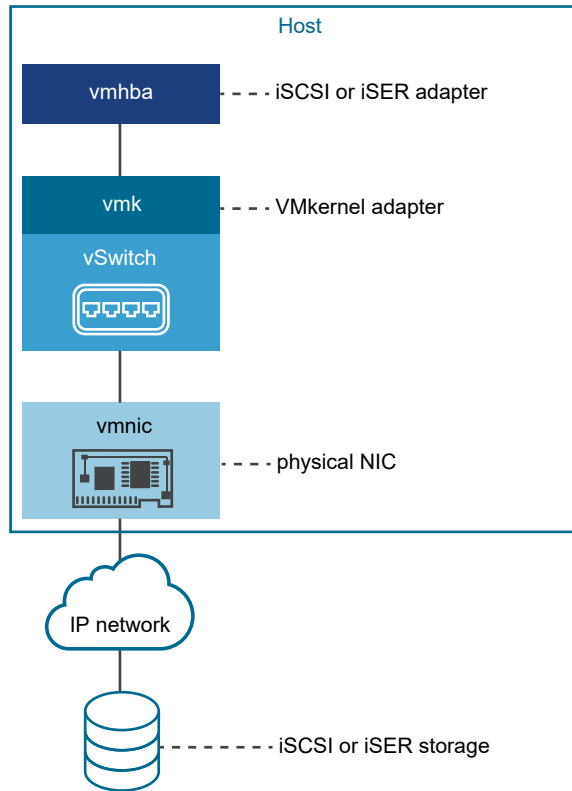
If you change the iSCSI name, it is used for new iSCSI sessions. For existing sessions, the new settings are not used until you log out and log in again.

Setting Up Network for iSCSI and iSER

Certain types of iSCSI adapters depend on the VMkernel networking. These adapters include the software or dependent hardware iSCSI adapters, and the VMware iSCSI over RDMA (iSER)

adapter. If your environment includes any of these adapters, you must configure connections for the traffic between the iSCSI or iSER component and the physical network adapters.

Configuring the network connection involves creating a virtual VMkernel adapter for each physical network adapter. You use 1:1 mapping between each virtual and physical network adapter. You then associate the VMkernel adapter with an appropriate iSCSI or iSER adapter. This process is called port binding.



Follow these rules when configuring the port binding:

- You can connect the software iSCSI adapter with any physical NICs available on your host.
- The dependent iSCSI adapters must be connected only to their own physical NICs.
- You must connect the iSER adapter only to the RDMA-capable network adapter.

For specific considerations on when and how to use network connections with software iSCSI, see the VMware knowledge base article at <http://kb.vmware.com/kb/2038869>.

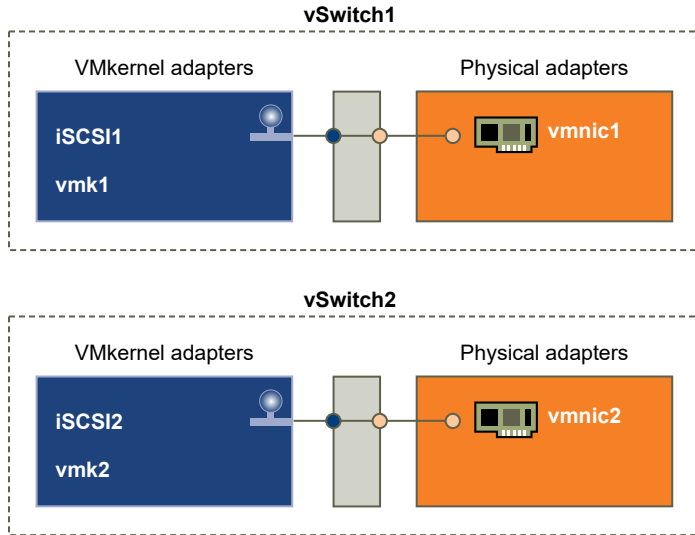
Multiple Network Adapters in iSCSI or iSER Configuration

If your host has more than one physical network adapter for iSCSI or iSER, you can use the adapters for multipathing.

You can use multiple physical adapters in a single or multiple switch configurations.

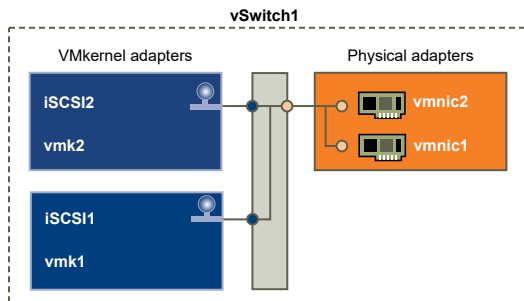
In the multiple switch configuration, you designate a separate vSphere switch for each virtual-to-physical adapter pair.

Figure 11-1. 1:1 Adapter Mapping on Separate vSphere Standard Switches



An alternative is to add all NICs and VMkernel adapters to the single vSphere switch. In this case, you must override the default network setup and make sure that each VMkernel adapter maps to only one corresponding active physical adapter.

Figure 11-2. 1:1 Adapter Mapping on a Single vSphere Standard Switch



The examples show configurations that use vSphere standard switches, but you can use distributed switches as well. For more information about vSphere distributed switches, see the *vSphere Networking* documentation.

The following considerations apply when you use multiple physical adapters:

- Physical network adapters must be on the same subnet as the storage system they connect to.
- (Applies only to iSCSI and not to iSER) If you use separate vSphere switches, you must connect them to different IP subnets. Otherwise, VMkernel adapters might experience connectivity problems and the host fails to discover the LUNs.
- The single switch configuration is not appropriate for iSER because iSER does not support NIC teaming.

Do not use port binding when any of the following conditions exist:

- Array target iSCSI ports are in a different broadcast domain and IP subnet.
- VMkernel adapters used for iSCSI connectivity exist in different broadcast domains, IP subnets, or use different virtual switches.

Note In iSER configurations, the VMkernel adapters used for iSER connectivity cannot be used for converged traffic. The VMkernel adapters that you created to enable connectivity between the ESXi host with iSER and the iSER target must be used only for iSER traffic.

Best Practices for Configuring Networking with Software iSCSI

When you configure networking with software iSCSI, consider several best practices.

Software iSCSI Port Binding

You can bind the software iSCSI initiator on the ESXi host to a single or multiple VMkernel ports, so that iSCSI traffic flows only through the bound ports. Unbound ports are not used for iSCSI traffic.

When port binding is configured, the iSCSI initiator creates iSCSI sessions from all bound ports to all configured target portals.

See the following examples.

| VMkernel Ports | Target Portals | iSCSI Sessions |
|------------------------|------------------|--------------------|
| 2 bound VMkernel ports | 2 target portals | 4 sessions (2 x 2) |
| 4 bound VMkernel ports | 1 target portal | 4 sessions (4 x 1) |
| 2 bound VMkernel ports | 4 target portals | 8 sessions (2 x 4) |

Note Make sure that all target portals are reachable from all VMkernel ports when port binding is used. Otherwise, iSCSI sessions might fail to create. As a result, the rescan operation might take longer than expected.

No Port Binding

If you do not use port binding, the ESXi networking layer selects the best VMkernel port based on its routing table. The host uses the port to create an iSCSI session with the target portal. Without the port binding, only one session per each target portal is created.

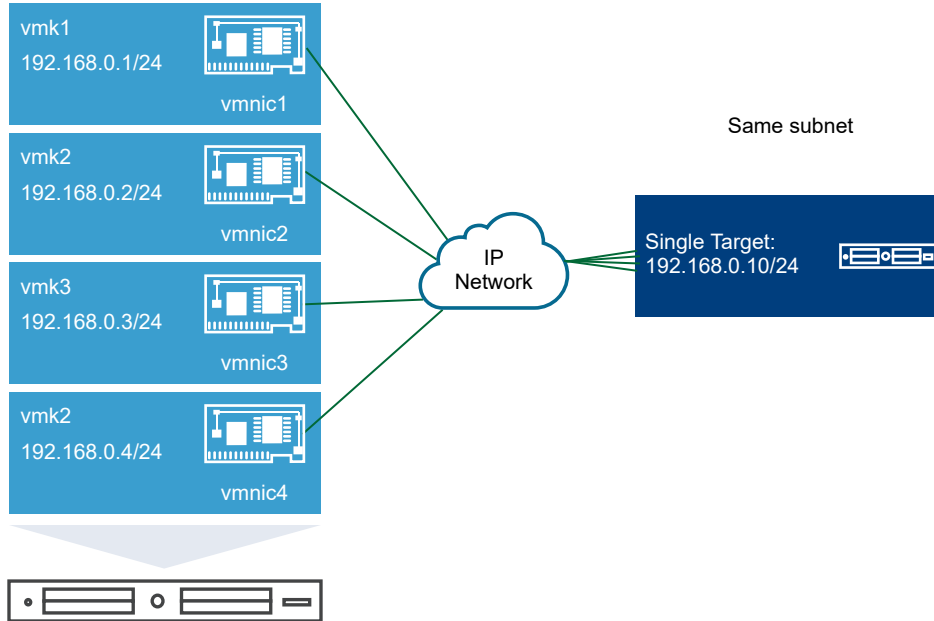
See the following examples.

| VMkernel Ports | Target Portals | iSCSI Sessions |
|--------------------------|------------------|----------------|
| 2 unbound VMkernel ports | 2 target portals | 2 sessions |
| 4 unbound VMkernel ports | 1 target portal | 1 session |
| 2 unbound VMkernel ports | 4 target portals | 4 sessions |

Software iSCSI Multipathing

Example 1. Multiple paths for an iSCSI target with a single network portal

If your target has only one network portal, you can create multiple paths to the target by adding multiple VMkernel ports on your ESXi host and binding them to the iSCSI initiator.

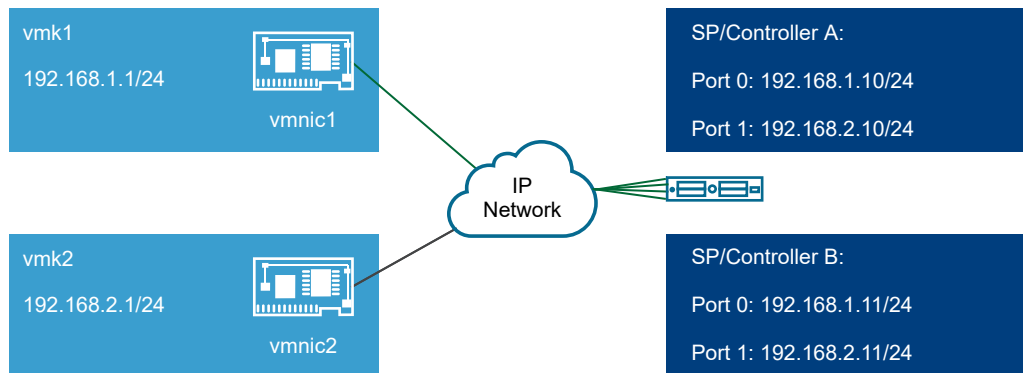


In this example, all initiator ports and the target portal are configured in the same subnet. The target is reachable through all bound ports. You have four VMkernel ports and one target portal, so total of four paths are created.

Without the port binding, only one path is created.

Example 2. Multiple paths with VMkernel ports in different subnets

You can create multiple paths by configuring multiple ports and target portals on different IP subnets. By keeping initiator and target ports in different subnets, you can force ESXi to create paths through specific ports. In this configuration, you do not use port binding because port binding requires that all initiator and target ports are on the same subnet.



ESXi selects vmk1 when connecting to Port 0 of Controller A and Controller B because all three ports are on the same subnet. Similarly, vmk2 is selected when connecting to Port 1 of Controller A and B. You can use NIC teaming in this configuration.

Total of four paths are created.

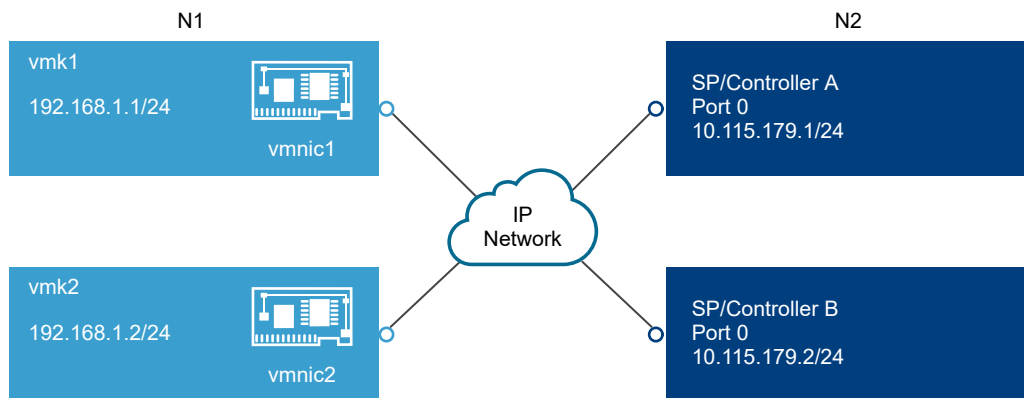
| Paths | Description |
|--------|--------------------------------|
| Path 1 | vmk1 and Port0 of Controller A |
| Path 2 | vmk1 and Port0 of Controller B |
| Path 3 | vmk2 and Port1 of Controller A |
| Path 4 | vmk2 and Port2 of Controller B |

Routing with Software iSCSI

You can use the `esxcli` command to add static routes for your iSCSI traffic. After you configure static routes, initiator and target ports in different subnets can communicate with each other.

Example 1. Using static routes with port binding

In this example, you keep all bound vmkernel ports in one subnet (N1) and configure all target portals in another subnet (N2). You can then add a static route for the target subnet (N2).

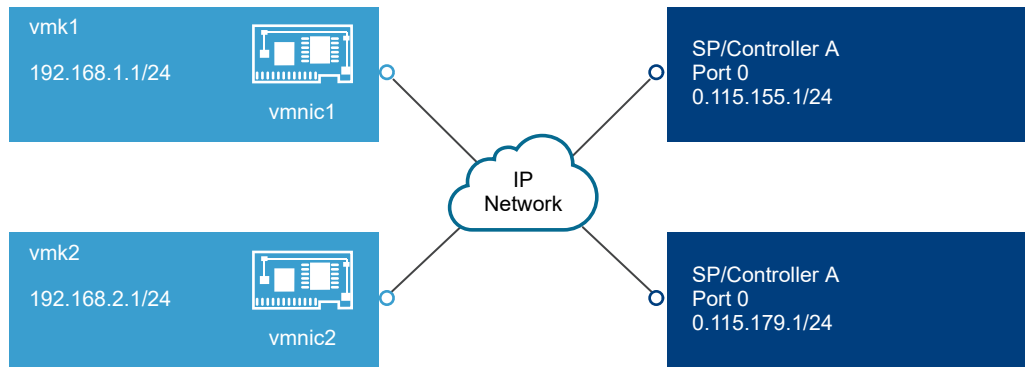


Use the following command:

```
# esxcli network ip route ipv4 add -gateway 192.168.1.253 -network 10.115.179.0/24
```

Example 2. Using static routes to create multiple paths

In this configuration, you use static routing when using different subnets. You cannot use the port binding with this configuration.



You configure vmk1 and vmk2 in separate subnets, 192.168.1.0 and 192.168.2.0. Your target portals are also in separate subnets, 10.115.155.0 and 10.115.179.0.

You can add the static route for 10.115.155.0 from vmk1. Make sure that the gateway is reachable from vmk1.

```
# esxcli network ip route ipv4 add -gateway 192.168.1.253 -network 10.115.155.0/24
```

You then add static route for 10.115.179.0 from vmk2. Make sure that the gateway is reachable from vmk2.

```
# esxcli network ip route ipv4 add -gateway 192.168.2.253 -network 10.115.179.0/24
```

When connecting with Port 0 of Controller A, vmk1 is used.

When connecting with Port 0 of Controller B, vmk2 is used.

Example 3. Routing with a separate gateway per vmkernel port

Starting with vSphere 6.5, you can configure a separate gateway per VMkernel port. If you use DHCP to obtain IP configuration for a VMkernel port, gateway information can also be obtained using DHCP.

To see gateway information per VMkernel port, use the following command:

```
# esxcli network ip interface ipv4 address list
```

| Name | IPv4 Address | IPv4 Netmask | IPv4 Broadcast | Address Type | Gateway | DHCP | DNS |
|------|----------------|---------------|----------------|--------------|----------------|------|-----|
| vmk0 | 10.115.155.122 | 255.255.252.0 | 10.115.155.255 | DHCP | 10.115.155.253 | true | |
| vmk1 | 10.115.179.209 | 255.255.252.0 | 10.115.179.255 | DHCP | 10.115.179.253 | true | |
| vmk2 | 10.115.179.146 | 255.255.252.0 | 10.115.179.255 | DHCP | 10.115.179.253 | true | |

With separate gateways per VMkernel port, you use port binding to reach targets in different subnets.

Configure Port Binding for iSCSI or iSER

The port binding creates connections for the traffic between certain types of iSCSI and iSER adapters and the physical network adapters.

The following types of adapters require the port binding:

- Software iSCSI adapter
- Dependent hardware iSCSI adapter
- VMware iSCSI over RDMA (iSER) adapter

The following tasks discuss the network configuration with a vSphere Standard switch.

You can also use the VMware vSphere[®] Distributed Switch[™] and VMware NSX[®] Virtual Switch[™] in the port binding configuration. For information about NSX virtual switches, see the *VMware NSX* documentation.

Note iSER does not support NIC teaming. When configuring port binding for iSER, use only one RDMA adapter per vSwitch.

If you use a vSphere distributed switch with multiple uplink ports, for port binding, create a separate distributed port group per each physical NIC. Then set the team policy so that each distributed port group has only one active uplink port. For detailed information on distributed switches, see the *vSphere Networking* documentation.

Procedure

1 Create a Single VMkernel Adapter for iSCSI

Connect the VMkernel, which runs services for iSCSI storage, to a physical network adapter.

2 Create Additional VMkernel Adapters for iSCSI

Use this task if you have two or more physical network adapters for iSCSI. And you want to connect all your physical adapters to a single vSphere standard switch. In this task, you add the physical adapters and VMkernel adapters to an existing vSphere standard switch.

3 Change Network Policy for iSCSI

If you use a single vSphere Standard switch to connect multiple VMkernel adapters to multiple network adapters, set up the network policy for this configuration. Only one physical network adapter must be active for each VMkernel adapter.

4 Bind iSCSI and VMkernel Adapters

Bind an iSCSI adapter with a VMkernel adapter.

5 Review Port Binding Details

Review networking details of the VMkernel adapter that is bound to the iSCSI adapter.

Create a Single VMkernel Adapter for iSCSI

Connect the VMkernel, which runs services for iSCSI storage, to a physical network adapter.

Procedure

- 1 Navigate to the host.
- 2 Select **Add Networking** from the right-click menu.

- 3 Select **VMkernel Network Adapter**, and click **Next**.
- 4 Select **New standard switch** to create a vSphere Standard switch.
- 5 Click the **Add adapters** icon, and select the network adapter (vmnic#) to use for iSCSI.
Make sure to assign the adapter to Active Adapters.

Important If you are creating a VMkernel adapter for dependent hardware iSCSI, select the vmnic that corresponds to the iSCSI component. See [Determine Association Between iSCSI and Network Adapters](#). With the iSER adapter, make sure to use an appropriate RDMA-capable vmnic.

- 6 Enter a network label.

A network label is a friendly name that identifies the VMkernel adapter that you are creating, for example, iSCSI.
- 7 Specify the IP settings.
- 8 Review the information and click **Finish**.

Results

You created the virtual VMkernel adapter (vmk#) for a physical network adapter (vmnic#) on your host.

What to do next

If your host has one physical network adapter for iSCSI traffic, you must bind the virtual adapter that you created to the iSCSI adapter.

If you have multiple network adapters, create additional VMkernel adapters and then perform iSCSI binding. The number of virtual adapters must correspond to the number of physical adapters on the host.

Create Additional VMkernel Adapters for iSCSI

Use this task if you have two or more physical network adapters for iSCSI. And you want to connect all your physical adapters to a single vSphere standard switch. In this task, you add the physical adapters and VMkernel adapters to an existing vSphere standard switch.

Prerequisites

Create a vSphere standard switch that maps an iSCSI VMkernel adapter to a single physical network adapter designated for iSCSI traffic.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Networking**, click **Virtual switches**, and select the vSphere switch that you want to modify from the list.

4 Connect additional network adapters to the switch.

- a Click the **Add host networking** icon.
- b Select **Physical Network Adapters**, and click **Next**.
- c Make sure that you are using the existing switch, and click **Next**.
- d Click the **Add adapters** icon, and select one or more network adapters (vmnic#) to use for iSCSI.

With the dependent hardware iSCSI adapter, select the vmnic that has a corresponding iSCSI component. With the iSER adapter, make sure to use an appropriate RDMA-capable vmnic.

- e Complete configuration, and click **Finish**.

5 Create iSCSI VMkernel adapters for all physical network adapters that you added.

The number of VMkernel interfaces must correspond to the number of physical network adapters on the vSphere standard switch.

- a Click the **Add host networking** icon.
- b Select **VMkernel Network Adapter**, and click **Next**.
- c Make sure that you are using the existing switch, and click **Next**.
- d Complete configuration, and click **Finish**.

What to do next

Change the network policy for all VMkernel adapters, so that only one physical network adapter is active for each VMkernel adapter. You can then bind the VMkernel adapters to the appropriate iSCSI adapters.

Change Network Policy for iSCSI

If you use a single vSphere Standard switch to connect multiple VMkernel adapters to multiple network adapters, set up the network policy for this configuration. Only one physical network adapter must be active for each VMkernel adapter.

By default, for each VMkernel adapter on the vSphere Standard switch, all network adapters appear as active. You must override this setup, so that each VMkernel adapter maps to only one corresponding active physical. For example, vmk1 maps to vmnic1, vmk2 maps to vmnic2, and so on.

Prerequisites

Create a vSphere Standard switch that connects VMkernel with physical network adapters designated for iSCSI traffic. The number of VMkernel adapters must correspond to the number of physical adapters on the vSphere Standard switch.

Procedure

- 1 Navigate to the host.

- 2 Click the **Configure** tab.
- 3 Under **Networking**, click **Virtual switches**, and select the vSphere switch that you want to modify from the list.
- 4 On the Port Groups tab, select the iSCSI VMkernel adapter and click the **Edit settings** icon.
- 5 On the **Edit Settings** wizard, click **Teaming and Failover** and select **Override** under Failover Order.
- 6 Designate only one physical adapter as active and move all remaining adapters to the **Unused Adapters** category.
- 7 Repeat [Step 4](#) through [Step 6](#) for each iSCSI VMkernel interface on the vSphere Standard switch.

Example: iSCSI Network Policy

The following table illustrates the proper iSCSI mapping where only one physical network adapter is active for each VMkernel adapter.

| VMkernel Adapter (vmk#) | Physical Network Adapter (vmnic#) |
|-------------------------|-----------------------------------|
| vmk1 (iSCSI1) | Active Adapters |
| | vmnic1 |
| | Unused Adapters |
| | vmnic2 |
| vmk2 (iSCSI2) | Active Adapters |
| | vmnic2 |
| | Unused Adapters |
| | vmnic1 |

What to do next

After you perform this task, bind the VMkernel adapters to the appropriate iSCSI adapters.

Bind iSCSI and VMkernel Adapters

Bind an iSCSI adapter with a VMkernel adapter.

Prerequisites

Create a virtual VMkernel adapter for each physical network adapter on your host. If you use multiple VMkernel adapters, set up the correct network policy.

Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Adapters**, and select the appropriate iSCSI adapter from the list.

- 4 Click the **Network Port Binding** tab and click the **Add** icon.
- 5 Select a VMkernel adapter to bind with the iSCSI adapter.

Note Make sure that the network policy for the VMkernel adapter is compliant with the binding requirements.

You can bind the software iSCSI adapter to one or more VMkernel adapters. For a dependent hardware iSCSI adapter or the iSER adapter, only one VMkernel adapter associated with the correct physical NIC is available.

- 6 Click **OK**.

Results

The network connection appears on the list of VMkernel port bindings for the iSCSI adapter.

Review Port Binding Details

Review networking details of the VMkernel adapter that is bound to the iSCSI adapter.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Adapters**, and select the appropriate iSCSI adapter from the list.
- 4 Click the **Network Port Binding** tab and click the **View Details** icon.
- 5 Review the VMkernel adapter information by switching between available tabs.

Managing iSCSI Network

Special considerations apply to network adapters, both physical and VMkernel, that are associated with an iSCSI adapter.

After you create network connections for iSCSI, an iSCSI indicator becomes enabled in the vSphere Client. The indicator shows that a particular virtual or physical network adapter is iSCSI-bound. To avoid disruptions in iSCSI traffic, follow these guidelines and considerations when managing iSCSI-bound virtual and physical network adapters:

- Make sure that the VMkernel network adapters are assigned addresses on the same subnet as the iSCSI storage portal they connect to.
- iSCSI adapters using VMkernel adapters cannot connect to iSCSI ports on different subnets, even if the iSCSI adapters discover those ports.
- When using separate vSphere switches to connect physical network adapters and VMkernel adapters, make sure that the vSphere switches connect to different IP subnets.
- If VMkernel adapters are on the same subnet, they must connect to a single vSwitch.

- If you migrate VMkernel adapters to a different vSphere switch, move associated physical adapters.
- Do not make configuration changes to iSCSI-bound VMkernel adapters or physical network adapters.
- Do not make changes that might break association of VMkernel adapters and physical network adapters. You can break the association if you remove one of the adapters or the vSphere switch that connects them. Or if you change the 1:1 network policy for their connection.

iSCSI Network Troubleshooting

A warning sign indicates non-compliant port group policy for an iSCSI-bound VMkernel adapter.

Problem

The VMkernel adapter's port group policy is considered non-compliant in the following cases:

- The VMkernel adapter is not connected to an active physical network adapter.
- The VMkernel adapter is connected to more than one physical network adapter.
- The VMkernel adapter is connected to one or more standby physical adapters.
- The active physical adapter is changed.

Solution

Follow the steps in [Change Network Policy for iSCSI](#) to set up the correct network policy for the iSCSI-bound VMkernel adapter.

Using Jumbo Frames with iSCSI

ESXi supports the use of Jumbo Frames with iSCSI.

Jumbo Frames are Ethernet frames with the size that exceeds 1500 Bytes. The maximum transmission unit (MTU) parameter is typically used to measure the size of Jumbo Frames.

When you use Jumbo Frames for iSCSI traffic, the following considerations apply:

- All network components must support Jumbo Frames.
- Check with your vendors to ensure your physical NICs and iSCSI adapters support Jumbo Frames.
- To set up and verify physical network switches for Jumbo Frames, consult your vendor documentation.

The following table explains the level of support that ESXi provides to Jumbo Frames.

Table 11-1. Support of Jumbo Frames

| Type of iSCSI Adapters | Jumbo Frames Support |
|----------------------------|-------------------------------|
| Software iSCSI | Supported |
| Dependent Hardware iSCSI | Supported. Check with vendor. |
| Independent Hardware iSCSI | Supported. Check with vendor. |
| VMware iSER | Supported. Check with vendor. |

Enable Jumbo Frames for Networking

You can enable Jumbo Frames for iSCSI or iSER adapters that use VMkernel networking for their traffic. These adapters include software iSCSI adapters, dependent hardware iSCSI adapters, and VMware iSER adapters.

To enable Jumbo Frames, change the default value of the maximum transmission units (MTU) parameter. You change the MTU parameter on the vSphere switch that you use for iSCSI traffic. For more information, see the *vSphere Networking* documentation.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Networking**, click **Virtual switches**, and select the vSphere switch that you want to modify from the list.
- 4 Click the **Edit settings** icon.
- 5 On the Properties page, change the MTU parameter.

This step sets the MTU for all physical NICs on that standard switch. Set the MTU value to the largest MTU size among all NICs connected to the standard switch. ESXi supports the MTU size of up to 9000 Bytes.

Enable Jumbo Frames for Independent Hardware iSCSI

To enable Jumbo Frames for independent hardware iSCSI adapters in the vSphere Client, change the default value of the maximum transmission units (MTU) parameter.

Use the Advanced Options settings to change the MTU parameter for the iSCSI HBA.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Adapters**, and select the independent hardware iSCSI adapter from the list of adapters.

- 4 Click the **Advanced Options** tab and click **Edit**.
- 5 Change the value of the MTU parameter.

ESXi supports the MTU size up to 9000 Bytes.

Configuring Discovery Addresses for iSCSI Adapters

You need to set up target discovery addresses, so that the iSCSI adapter can determine which storage resource on the network is available for access.

The ESXi system supports these discovery methods:

Dynamic Discovery

Also known as SendTargets discovery. Each time the initiator contacts a specified iSCSI server, the initiator sends the SendTargets request to the server. The server responds by supplying a list of available targets to the initiator. The names and IP addresses of these targets appear on the **Static Discovery** tab. If you remove a static target added by dynamic discovery, the target might be returned to the list the next time a rescan happens, the iSCSI adapter is reset, or the host is rebooted.

Note With software and dependent hardware iSCSI, ESXi filters target addresses based on the IP family of the iSCSI server address specified. If the address is IPv4, IPv6 addresses that might come in the SendTargets response from the iSCSI server are filtered out. When DNS names are used to specify an iSCSI server, or when the SendTargets response from the iSCSI server has DNS names, ESXi relies on the IP family of the first resolved entry from DNS lookup.

Static Discovery

In addition to the dynamic discovery method, you can use static discovery and manually enter information for the targets. The iSCSI adapter uses a list of targets that you provide to contact and communicate with the iSCSI servers.

Configure Dynamic or Static Discovery for iSCSI and iSER

With dynamic discovery, each time the initiator contacts a specified iSCSI storage system, it sends the SendTargets request to the system. The iSCSI system responds by supplying a list of available targets to the initiator. In addition to the dynamic discovery method, you can use static discovery and manually enter information for the targets.

When you set up static or dynamic discovery, you can only add new iSCSI targets. You cannot change any parameters of an existing target. To make changes, remove the existing target and add a new one.

Prerequisites

Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Adapters**, and select the adapter (vmhba#) to configure.
- 4 Configure the discovery method.

| Discovery Method | Description |
|-------------------|--|
| Dynamic Discovery | <ol style="list-style-type: none"> a Click Dynamic Discovery and click Add. b Enter the IP address or DNS name of the storage system and click OK. c Rescan the iSCSI adapter. <p>After establishing the SendTargets session with the iSCSI system, your host populates the Static Discovery list with all newly discovered targets.</p> |
| Static Discovery | <ol style="list-style-type: none"> a Click Static Discovery and click Add. b Enter the target's information and click OK c Rescan the iSCSI adapter. |

Remove Dynamic or Static iSCSI Targets

Remove iSCSI servers that appear on the list of targets.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Adapters**, and select the iSCSI adapter to modify from the list.
- 4 Switch between **Dynamic Discovery** and **Static Discovery**.
- 5 Select an iSCSI server to remove and click **Remove**.
- 6 Rescan the iSCSI adapter.

If you are removing the static target that was dynamically discovered, you need to remove it from the storage system before performing the rescan. Otherwise, your host will automatically discover and add the target to the list of static targets when you rescan the adapter.

Configuring CHAP Parameters for iSCSI Adapters

Because the IP networks that the iSCSI technology uses to connect to remote targets do not protect the data they transport, you must ensure security of the connection. One of the protocols that iSCSI implements is the Challenge Handshake Authentication Protocol (CHAP), which verifies the legitimacy of initiators that access targets on the network.

CHAP uses a three-way handshake algorithm to verify the identity of your host and, if applicable, of the iSCSI target when the host and target establish a connection. The verification is based on a predefined private value, or CHAP secret, that the initiator and target share.

ESXi supports CHAP authentication at the adapter level. In this case, all targets receive the same CHAP name and secret from the iSCSI initiator. For software and dependent hardware iSCSI adapters, ESXi also supports per-target CHAP authentication, which allows you to configure different credentials for each target to achieve greater level of security.

Selecting CHAP Authentication Method

ESXi supports unidirectional CHAP for all types of iSCSI initiators, and bidirectional CHAP for software and dependent hardware iSCSI.

Before configuring CHAP, check whether CHAP is enabled at the iSCSI storage system. Also, obtain information about the CHAP authentication method the system supports. If CHAP is enabled, configure it for your initiators, making sure that the CHAP authentication credentials match the credentials on the iSCSI storage.

ESXi supports the following CHAP authentication methods:

Unidirectional CHAP

In unidirectional CHAP authentication, the target authenticates the initiator, but the initiator does not authenticate the target.

Bidirectional CHAP

The bidirectional CHAP authentication adds an extra level of security. With this method, the initiator can also authenticate the target. VMware supports this method for software and dependent hardware iSCSI adapters only.

For software and dependent hardware iSCSI adapters, you can set unidirectional CHAP and bidirectional CHAP for each adapter or at the target level. Independent hardware iSCSI supports CHAP only at the adapter level.

When you set the CHAP parameters, specify a security level for CHAP.

Note When you specify the CHAP security level, how the storage array responds depends on the array's CHAP implementation and is vendor-specific. For information on CHAP authentication behavior in different initiator and target configurations, consult the array documentation.

Table 11-2. CHAP Security Level

| CHAP Security Level | Description | Supported |
|---|---|--|
| None | The host does not use CHAP authentication. If authentication is enabled, use this option to disable it. | Software iSCSI Dependent hardware iSCSI Independent hardware iSCSI |
| Use unidirectional CHAP if required by target | The host prefers a non-CHAP connection, but can use a CHAP connection if required by the target. | Software iSCSI Dependent hardware iSCSI |

Table 11-2. CHAP Security Level (continued)

| CHAP Security Level | Description | Supported |
|---|---|--|
| Use unidirectional CHAP unless prohibited by target | The host prefers CHAP, but can use non-CHAP connections if the target does not support CHAP. | Software iSCSI Dependent hardware iSCSI Independent hardware iSCSI |
| Use unidirectional CHAP | The host requires successful CHAP authentication. The connection fails if CHAP negotiation fails. | Software iSCSI Dependent hardware iSCSI Independent hardware iSCSI |
| Use bidirectional CHAP | The host and the target support bidirectional CHAP. | Software iSCSI Dependent hardware iSCSI |

Set Up CHAP for iSCSI or iSER Adapter

When you set up CHAP name and secret at the iSCSI adapter level, all targets receive the same parameters from the adapter. By default, all discovery addresses or static targets inherit CHAP parameters that you set up at the adapter level.

The CHAP name cannot exceed 511 alphanumeric characters and the CHAP secret cannot exceed 255 alphanumeric characters. Some adapters, for example the QLogic adapter, might have lower limits, 255 for the CHAP name and 100 for the CHAP secret.

Prerequisites

- Before setting up CHAP parameters for software or dependent hardware iSCSI, determine whether to configure unidirectional or bidirectional CHAP. Independent hardware iSCSI adapters do not support bidirectional CHAP.
- Verify CHAP parameters configured on the storage side. Parameters that you configure must match the ones on the storage side.
- Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

- 1 Display storage adapters and select the iSCSI adapter to configure.
- 2 Click the **Properties** tab and click **Edit** in the Authentication panel.
- 3 Specify authentication method.
 - **None**
 - **Use unidirectional CHAP if required by target**
 - **Use unidirectional CHAP unless prohibited by target**
 - **Use unidirectional CHAP**
 - **Use bidirectional CHAP.** To configure bidirectional CHAP, you must select this option.

4 Specify the outgoing CHAP name.

Make sure that the name you specify matches the name configured on the storage side.

- To set the CHAP name to the iSCSI adapter name, select **Use initiator name**.
- To set the CHAP name to anything other than the iSCSI initiator name, deselect **Use initiator name** and enter a name in the **Name** text box.

5 Enter an outgoing CHAP secret to be used as part of authentication. Use the same secret that you enter on the storage side.

6 If configuring bidirectional CHAP, specify incoming CHAP credentials.

Make sure to use different secrets for the outgoing and incoming CHAP.

7 Click **OK**.

8 Rescan the iSCSI adapter.

Results

If you change the CHAP parameters, they are used for new iSCSI sessions. For existing sessions, new settings are not used until you log out and log in again.

Set Up CHAP for Target

If you use software and dependent hardware iSCSI adapters, you can configure different CHAP credentials for each discovery address or static target.

The CHAP name cannot exceed 511 and the CHAP secret 255 alphanumeric characters.

Prerequisites

- Before setting up CHAP parameters for software or dependent hardware iSCSI, determine whether to configure unidirectional or bidirectional CHAP.
- Verify CHAP parameters configured on the storage side. Parameters that you configure must match the ones on the storage side.
- Access storage adapters.
- Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

- 1 Select the iSCSI adapter to configure.
- 2 Click either **Dynamic Discovery** or **Static Discovery**.
- 3 From the list of available targets, select a target to configure and click **Authentication**.
- 4 Deselect **Inherit settings from parent** and specify authentication method.
 - **None**
 - **Use unidirectional CHAP if required by target**

- **Use unidirectional CHAP unless prohibited by target**
 - **Use unidirectional CHAP**
 - **Use bidirectional CHAP.** To configure bidirectional CHAP, you must select this option.
- 5 Specify the outgoing CHAP name.
- Make sure that the name you specify matches the name configured on the storage side.
- To set the CHAP name to the iSCSI adapter name, select **Use initiator name**.
 - To set the CHAP name to anything other than the iSCSI initiator name, deselect **Use initiator name** and enter a name in the **Name** text box.
- 6 Enter an outgoing CHAP secret to be used as part of authentication. Use the same secret that you enter on the storage side.
- 7 If configuring bidirectional CHAP, specify incoming CHAP credentials.
- Make sure to use different secrets for the outgoing and incoming CHAP.
- 8 Click **OK**.
- 9 Rescan the iSCSI adapter.

Results

If you change the CHAP parameters, they are used for new iSCSI sessions. For existing sessions, new settings are not used until you log out and login again.

Configuring Advanced Parameters for iSCSI

You might need to configure additional parameters for your iSCSI initiators. For example, some iSCSI storage systems require ARP (Address Resolution Protocol) redirection to move iSCSI traffic dynamically from one port to another. In this case, you must activate the ARP redirection on your host.

The following table lists advanced iSCSI parameters that you can configure using the vSphere Client. In addition, you can use the vSphere CLI commands to configure some of the advanced parameters. For information, see the *Getting Started with vSphere Command-Line Interfaces* documentation.

Depending on the type of your adapters, certain parameters might not be available.

Important Do not change the advanced iSCSI settings unless VMware support or Storage Vendors direct you to change them.

Table 11-3. Additional Parameters for iSCSI Initiators

| Advanced Parameter | Description |
|----------------------|--|
| Header Digest | Increases data integrity. When the header digest parameter is enabled, the system performs a checksum over each header part of the iSCSI Protocol Data Unit (PDU). The system verifies the data using the CRC32C algorithm. |
| Data Digest | Increases data integrity. When the data digest parameter is enabled, the system performs a checksum over each PDU data part. The system verifies the data using the CRC32C algorithm. Note Systems that use the Intel Nehalem processors offload the iSCSI digest calculations for software iSCSI. This offload helps to reduce the impact on performance. |
| ErrorRecoveryLevel | iSCSI Error Recovery Level (ERL) value that the iSCSI initiator on the host negotiates during a login. |
| LoginRetryMax | Maximum number of times the ESXi iSCSI initiator attempts to log into a target before ending the attempts. |
| MaxOutstandingR2T | Defines the R2T (Ready to Transfer) PDUs that can be in transition before an acknowledge PDU is received. |
| FirstBurstLength | Specifies the maximum amount of unsolicited data an iSCSI initiator can send to the target during the execution of a single SCSI command, in bytes. |
| MaxBurstLength | Maximum SCSI data payload in a Data-In or a solicited Data-Out iSCSI sequence, in bytes. |
| MaxRecvDataSegLength | Maximum data segment length, in bytes, that can be received in an iSCSI PDU. |
| MaxCommands | Maximum SCSI commands that can be queued on the iSCSI adapter. |
| DefaultTimeToWait | Minimum time in seconds to wait before attempting a logout or an active task reassignment after an unexpected connection termination or reset. |
| DefaultTimeToRetain | Maximum time in seconds, during which reassigning the active task is still possible after a connection termination or reset. |
| LoginTimeout | Time in seconds the initiator will wait for the login response to finish. |
| LogoutTimeout | Time in seconds initiator will wait to get a response for Logout request PDU. |
| RecoveryTimeout | Specifies the amount of time, in seconds, that can lapse while a session recovery is performed. If the timeout exceeds its limit, the iSCSI initiator ends the session. |
| No-Op Interval | Specifies the time interval, in seconds, between NOP-Out requests sent from your iSCSI initiator to an iSCSI target. The NOP-Out requests serve as the ping mechanism to verify that a connection between the iSCSI initiator and the iSCSI target is active. |
| No-Op Timeout | Specifies the amount of time, in seconds, that can lapse before your host receives a NOP-In message. The iSCSI target sends the message in response to the NOP-Out request. When the no-op timeout limit is exceeded, the initiator ends the current session and starts a new one. |
| ARP Redirect | With this parameter enabled, storage systems can move iSCSI traffic dynamically from one port to another. Storage systems that perform array-based failovers require the ARP parameter. |
| Delayed ACK | With this parameter enabled, storage systems can delay an acknowledgment of received data packets. |

Configure Advanced Parameters for iSCSI

The advanced iSCSI settings control such parameters as header and data digest, ARP redirection, delayed ACK, and so on.

Caution Do not make any changes to the advanced iSCSI settings unless you are working with the VMware support team or otherwise have thorough information about the values to provide for the settings.

Prerequisites

Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Adapters**, and select the adapter (vmhba#) to configure.
- 4 Configure advanced parameters.

| Option | Description |
|----------------------|--|
| At the adapter level | Click the Advanced Options tab and click Edit . |
| At the target level | <ol style="list-style-type: none"> a Click either Dynamic Discovery or Static Discovery. b From the list of available targets, select a target to configure and click Advanced. |

- 5 Enter any required values for the advanced parameters you want to modify.

iSCSI Session Management

To communicate with each other, iSCSI initiators and targets establish iSCSI sessions. You can review and manage iSCSI sessions using vSphere CLI.

By default, software iSCSI and dependent hardware iSCSI initiators start one iSCSI session between each initiator port and each target port. If your iSCSI initiator or target has more than one port, your host can have multiple sessions established. The default number of sessions for each target equals the number of ports on the iSCSI adapter times the number of target ports.

Using vSphere CLI, you can display all current sessions to analyze and debug them. To create more paths to storage systems, you can increase the default number of sessions by duplicating existing sessions between the iSCSI adapter and target ports.

You can also establish a session to a specific target port. This capability is useful if your host connects to a single-port storage system that presents only one target port to your initiator. The system then redirects additional sessions to a different target port. Establishing a new session between your iSCSI initiator and another target port creates an additional path to the storage system.

The following considerations apply to iSCSI session management:

- Some storage systems do not support multiple sessions from the same initiator name or endpoint. Attempts to create multiple sessions to such targets can result in an unpredictable behavior of your iSCSI environment.
- Storage vendors can provide automatic session managers. Using the automatic session manages to add or delete sessions, does not guarantee lasting results and can interfere with the storage performance.

Review iSCSI Sessions

Use the vCLI command to display iSCSI sessions between an iSCSI adapter and a storage system.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ To list iSCSI sessions, run the following command:

```
esxcli iscsi session list
```

The command takes these options:

| Option | Description |
|-------------------------------|---|
| <code>-A --adapter=str</code> | The iSCSI adapter name, for example, vmhba34. |
| <code>-s --isid=str</code> | The iSCSI session identifier. |
| <code>-n --name=str</code> | The iSCSI target name, for example, iqn.X. |

Add iSCSI Sessions

Use the vCLI to add an iSCSI session for a target you specify or to duplicate an existing session. By duplicating sessions, you increase the default number of sessions and create additional paths to storage systems.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ To add or duplicate an iSCSI session, run the following command:

```
esxcli iscsi session add
```

The command takes these options:

| Option | Description |
|---------------------------------------|--|
| <code>-A --adapter=</code> <i>str</i> | The iSCSI adapter name, for example, vmhba34. This option is required. |
| <code>-s --isid=</code> <i>str</i> | The ISID of a session to duplicate. You can find it by listing all sessions. |
| <code>-n --name=</code> <i>str</i> | The iSCSI target name, for example, iqn.X. |

What to do next

Rescan the iSCSI adapter.

Remove iSCSI Sessions

Use the vCLI command to remove an iSCSI session between an iSCSI adapter and a target.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ To remove a session, run the following command:

```
esxcli iscsi session remove
```

The command takes these options:

| Option | Description |
|---------------------------------------|--|
| <code>-A --adapter=</code> <i>str</i> | The iSCSI adapter name, for example, vmhba34. This option is required. |
| <code>-s --isid=</code> <i>str</i> | The ISID of a session to remove. You can find it by listing all session. |
| <code>-n --name=</code> <i>str</i> | The iSCSI target name, for example, iqn.X. |

What to do next

Rescan the iSCSI adapter.

Booting from iSCSI SAN

12

When you set up your host to boot from a SAN, your host's boot image is stored on one or more LUNs in the SAN storage system. When the host starts, it boots from the LUN on the SAN rather than from its local disk.

You can use boot from the SAN if you do not want to handle maintenance of local storage or have diskless hardware configurations, such as blade systems.

ESXi supports different methods of booting from the iSCSI SAN.

Table 12-1. Boot from iSCSI SAN support

| Independent Hardware iSCSI | Software iSCSI and Dependent Hardware iSCSI |
|---|---|
| Configure the iSCSI HBA to boot from the SAN. For information on configuring the HBA, see Configure Independent Hardware iSCSI Adapter for SAN Boot | Use the network adapter that supports the iBFT. For information, see iBFT iSCSI Boot Overview . |

Read the following topics next:

- [General Recommendations for Boot from iSCSI SAN](#)
- [Prepare the iSCSI SAN](#)
- [Configure Independent Hardware iSCSI Adapter for SAN Boot](#)
- [iBFT iSCSI Boot Overview](#)

General Recommendations for Boot from iSCSI SAN

If you plan to set up and use an iSCSI LUN as the boot device for your host, follow certain general guidelines.

The following guidelines apply to booting from the independent hardware iSCSI and iBFT.

- Review any vendor recommendations for the hardware you use in your boot configuration.
- For installation prerequisites and requirements, review *vSphere Installation and Setup*.
- Use static IP addresses to reduce the chances of DHCP conflicts.
- Use different LUNs for VMFS datastores and boot partitions.

- Configure proper ACLs on your storage system.
 - The boot LUN must be visible only to the host that uses the LUN. No other host on the SAN is permitted to see that boot LUN.
 - If a LUN is used for a VMFS datastore, multiple hosts can share the LUN.
- Configure a diagnostic partition.
 - With the independent hardware iSCSI only, you can place the diagnostic partition on the boot LUN. If you configure the diagnostic partition in the boot LUN, this LUN cannot be shared across multiple hosts. If a separate LUN is used for the diagnostic partition, multiple hosts can share the LUN.
 - If you boot from SAN using iBFT, you cannot set up a diagnostic partition on a SAN LUN. To collect your host's diagnostic information, use the vSphere ESXi Dump Collector on a remote server. For information about the ESXi Dump Collector, see *vCenter Server Installation and Setup* and *vSphere Networking*.

Prepare the iSCSI SAN

Before you configure your host to boot from an iSCSI LUN, prepare and configure your storage area network.

Caution If you use scripted installation to install ESXi when booting from a SAN, you must take special steps to avoid unintended data loss.

Procedure

- 1 Connect network cables, referring to any cabling guide that applies to your setup.
- 2 Ensure IP connectivity between your storage system and server.

Verify configuration of any routers or switches on your storage network. Storage systems must be able to ping the iSCSI adapters in your hosts.
- 3 Configure the storage system.
 - a Create a volume (or LUN) on the storage system for your host to boot from.
 - b Configure the storage system so that your host has access to the assigned LUN.

This step might involve updating ACLs with the IP addresses, iSCSI names, and the CHAP authentication parameter you use on your host. On some storage systems, in addition to providing access information for the ESXi host, you must also explicitly associate the assigned LUN with the host.
 - c Ensure that the LUN is presented to the host correctly.
 - d Ensure that no other system has access to the configured LUN.
 - e Record the iSCSI name and IP addresses of the targets assigned to the host.

You must have this information to configure your iSCSI adapters.

Configure Independent Hardware iSCSI Adapter for SAN Boot

If your ESXi host uses an independent hardware iSCSI adapter, such as QLogic HBA, you can configure the adapter to boot from the SAN.

This procedure discusses how to enable the QLogic iSCSI HBA to boot from the SAN. For more information and more up-to-date details about QLogic adapter configuration settings, see the QLogic website.

Prerequisites

Because you start with booting from the VMware installation media, set up your host to boot from CD/DVD-ROM.

Procedure

- 1 Insert the installation CD/DVD in the CD/DVD-ROM drive and reboot the host.
- 2 Use the BIOS to set the host to boot from the CD/DVD-ROM drive first.
- 3 During server POST, press Ctrl+q to enter the QLogic iSCSI HBA configuration menu.
- 4 Select the I/O port to configure.

By default, the Adapter Boot mode is set to Disable.
- 5 Configure the HBA.
 - a From the **Fast!UTIL Options** menu, select **Configuration Settings > Host Adapter Settings**.
 - b (Optional) Configure the following settings for your host adapter: initiator IP address, subnet mask, gateway, initiator iSCSI name, and CHAP.
- 6 Configure iSCSI settings.

See [Configure iSCSI Boot Settings](#).
- 7 Save your changes and restart the system.

Configure iSCSI Boot Settings

Configure iSCSI boot parameters, so that your ESXi host can boot from an iSCSI LUN.

Procedure

- 1 From the **Fast!UTIL Options** menu, select **Configuration Settings > iSCSI Boot Settings**.
- 2 Before you can set SendTargets, set Adapter Boot mode to **Manual**.

3 Select **Primary Boot Device Settings**.

- a Enter the discovery **Target IP** and **Target Port**.
- b Configure the **Boot LUN** and **iSCSI Name** parameters.
 - If only one iSCSI target and one LUN are available at the target address, leave **Boot LUN** and **iSCSI Name** blank.

After your host reaches the target storage system, these text boxes are populated with appropriate information.

- If more than one iSCSI target and LUN are available, supply values for **Boot LUN** and **iSCSI Name**.
- c Save changes.

4 From the **iSCSI Boot Settings** menu, select the primary boot device.

An auto rescan of the HBA finds new target LUNs.

5 Select the iSCSI target.

If more than one LUN exists within the target, you can select a specific LUN ID by pressing **Enter** after you locate the iSCSI device.

6 Return to the **Primary Boot Device Setting** menu. After the rescan, **Boot LUN** and **iSCSI Name** are populated. Change the value of **Boot LUN** to the appropriate LUN ID.

iBFT iSCSI Boot Overview

ESXi hosts can boot from an iSCSI SAN using the software or dependent hardware iSCSI adapters and network adapters.

To deploy ESXi and boot from the iSCSI SAN, the host must have an iSCSI boot capable network adapter. The adapter must support the iSCSI Boot Firmware Table (iBFT) format, a method of communicating parameters about the iSCSI boot device to an operating system.

Before installing ESXi and booting from the iSCSI SAN, configure the networking and iSCSI boot parameters on the network adapter. Because configuring the network adapter is vendor-specific, review your vendor documentation for instructions.

When you boot from iSCSI for the first time, the iSCSI boot firmware on your system connects to an iSCSI target. If a login is successful, the firmware saves the networking and iSCSI boot parameters in the iBFT and stores the table in the system's memory. The system uses this table to configure its own iSCSI connection and networking and to start up.

The following list describes the iBFT iSCSI boot sequence.

- 1 When restarted, the system BIOS detects the iSCSI boot firmware on the network adapter.
- 2 The iSCSI boot firmware uses the preconfigured boot parameters to connect with the specified iSCSI target.

- 3 After the successful connection, the iSCSI boot firmware writes the networking and iSCSI boot parameters in to the iBFT. The firmware stores the table in the system memory.

Note The system uses this table to configure its own iSCSI connection and networking and to start up.

- 4 The BIOS boots the boot device.
- 5 The VMkernel starts loading and takes over the boot operation.
- 6 Using the boot parameters from the iBFT, the VMkernel connects to the iSCSI target.
- 7 After the iSCSI connection is established, the system boots.

iBFT iSCSI Boot Considerations

When you boot the ESXi host from iSCSI using iBFT-enabled network adapters, certain considerations apply.

- Update your NIC's boot code and iBFT firmware using vendor supplied tools before trying to install and boot VMware ESXi. Consult vendor documentation and VMware HCL for supported boot code and iBFT firmware versions for VMware ESXi iBFT boot.
- The iBFT iSCSI boot does not support failover for the iBFT-enabled network adapters.
- After you set up your host to boot from iBFT iSCSI, the following restrictions apply:
 - You cannot disable the software iSCSI adapter. If the iBFT configuration is present in the BIOS, the host re-enables the software iSCSI adapter during each reboot.

Note If you do not use the iBFT-enabled network adapter for the iSCSI boot and do not want the software iSCSI adapter to be always enabled, remove the iBFT configuration from the network adapter.

- You cannot remove the iBFT iSCSI boot target using the vSphere Client. The target appears on the list of adapter static targets.

Configuring iBFT Boot from SAN

You can boot from the iSCSI SAN using the software iSCSI adapter or a dependent hardware iSCSI adapter and a network adapter. The network adapter must support iBFT.

When you set up your host to boot with iBFT, you perform a number of tasks.

What to read next

Procedure

1 [Configure iSCSI Boot Parameters](#)

To begin an iSCSI boot process, a network adapter on your host must have a specially configured iSCSI boot firmware. When you configure the firmware, you specify the networking and iSCSI parameters and enable the adapter for the iSCSI boot.

2 Change Boot Sequence in BIOS

When setting up your host to boot from iBFT iSCSI, change the boot sequence to force your host to boot in an appropriate order.

3 Install ESXi to iSCSI Target

When setting up your host to boot from iBFT iSCSI, install the ESXi image to the target LUN.

4 Boot ESXi from iSCSI Target

After preparing the host for an iBFT iSCSI boot and copying the ESXi image to the iSCSI target, perform the actual boot.

Configure iSCSI Boot Parameters

To begin an iSCSI boot process, a network adapter on your host must have a specially configured iSCSI boot firmware. When you configure the firmware, you specify the networking and iSCSI parameters and enable the adapter for the iSCSI boot.

Configuration on the network adapter can be dynamic or static. If you use the dynamic configuration, you indicate that all target and initiator boot parameters are acquired using DHCP. For the static configuration, you manually enter data that includes your host's IP address and initiator IQN, and the target parameters.

Procedure

- ◆ On the network adapter that you use for the boot from iSCSI, specify networking and iSCSI parameters.

Because configuring the network adapter is vendor-specific, review your vendor documentation for instructions.

Change Boot Sequence in BIOS

When setting up your host to boot from iBFT iSCSI, change the boot sequence to force your host to boot in an appropriate order.

Change the BIOS boot sequence to the following sequence:

- iSCSI
- DVD-ROM

Because changing the boot sequence in the BIOS is vendor-specific, refer to vendor documentation for instructions. The following sample procedure explains how to change the boot sequence on a Dell host with a Broadcom network adapter.

Procedure

- 1 Turn on the host.
- 2 During Power-On Self-Test (POST), press F2 to enter the BIOS Setup.
- 3 In the BIOS Setup, select **Boot Sequence** and press Enter.

- 4 In the Boot Sequence menu, arrange the bootable items so that iSCSI precedes the DVD-ROM.
- 5 Press Esc to exit the Boot Sequence menu.
- 6 Press Esc to exit the BIOS Setup.
- 7 Select **Save Changes** and click **Exit** to exit the BIOS Setup menu.

Install ESXi to iSCSI Target

When setting up your host to boot from iBFT iSCSI, install the ESXi image to the target LUN.

Prerequisites

- Configure iSCSI boot firmware on your boot NIC to point to the target LUN that you want to use as the boot LUN.
- Change the boot sequence in the BIOS so that iSCSI precedes the DVD-ROM.
- If you use Broadcom adapters, set **Boot to iSCSI target** to **Disabled**.

Procedure

- 1 Insert the installation media in the CD/DVD-ROM drive and restart the host.
- 2 When the installer starts, follow the typical installation procedure.
- 3 When prompted, select the iSCSI LUN as the installation target.
The installer copies the ESXi boot image to the iSCSI LUN.
- 4 After the system restarts, remove the installation DVD.

Boot ESXi from iSCSI Target

After preparing the host for an iBFT iSCSI boot and copying the ESXi image to the iSCSI target, perform the actual boot.

Prerequisites

- Configure the iSCSI boot firmware on your boot NIC to point to the boot LUN.
- Change the boot sequence in the BIOS so that iSCSI precedes the boot device.
- If you use Broadcom adapters, set **Boot to iSCSI target** to **Enabled**

Procedure

- 1 Restart the host.
The host boots from the iSCSI LUN using iBFT data. During the first boot, the iSCSI initialization script sets up default networking. The network setup is persistent after subsequent reboots.
- 2 (Optional) Adjust networking configuration using the vSphere Client.

Networking Best Practices

To boot the ESXi host from iSCSI using iBFT, you must properly configure networking.

To achieve greater security and better performance, have redundant network adapters on the host.

How you set up all the network adapters depends on whether your environment uses shared or isolated networks for the iSCSI traffic and host management traffic.

Shared iSCSI and Management Networks

Configure the networking and iSCSI parameters on the first network adapter on the host. After the host boots, you can add secondary network adapters to the default port group.

Isolated iSCSI and Management Networks

When you configure isolated iSCSI and management networks, follow these guidelines to avoid bandwidth problems.

- Your isolated networks must be on different subnets.
- If you use VLANs to isolate the networks, they must have different subnets to ensure that routing tables are properly set up.
- VMware recommends that you configure the iSCSI adapter and target to be on the same subnet. If you set up the iSCSI adapter and target on different subnets, the following restrictions apply:
 - The default VMkernel gateway must be able to route both the management and iSCSI traffic.
 - After you boot your host, you can use the iBFT-enabled network adapter only for iBFT. You cannot use the adapter for other iSCSI traffic.
- Use the first physical network adapter for the management network.
- Use the second physical network adapter for the iSCSI network. Make sure to configure the iBFT.
- After the host boots, you can add secondary network adapters to both the management and iSCSI networks.

Change iBFT iSCSI Boot Settings

If settings, such as the IQN name, IP address, and so on, change on the iSCSI storage or your host, update the iBFT. This task assumes that the boot LUN and the data stored on the LUN remain intact.

Procedure

- 1 Shut down the ESXi host.
- 2 Change iSCSI storage settings.

- 3 Update the iBFT on the host with the new settings.
- 4 Restart the host.

The host boots using the new information stored in the iBFT.

Troubleshooting iBFT iSCSI Boot

The topics in this section help you to identify and solve problems you might encounter when using iBFT iSCSI boot.

Loss of System's Gateway Causes Loss of Network Connectivity

You lose network connectivity when you delete a port group associated with the iBFT network adapter.

Problem

A loss of network connectivity occurs after you delete a port group.

Cause

When you specify a gateway in the iBFT-enabled network adapter during ESXi installation, this gateway becomes the system's default gateway. If you delete the port group associated with the network adapter, the system's default gateway is lost. This action causes the loss of network connectivity.

Solution

Do not set an iBFT gateway unless it is required. If the gateway is required, after installation, manually set the system's default gateway to the one that the management network uses.

Changing iSCSI Boot Parameters Causes ESXi to Boot in Stateless Mode

Changing iSCSI boot parameters on the network adapter after the first boot does not update the iSCSI and networking configuration on the ESXi host.

Problem

If you change the iSCSI boot parameters on the network adapter after the first ESXi boot from iSCSI, the host will boot in a stateless mode.

Cause

The firmware uses the updated boot configuration to connect to the iSCSI target and load the ESXi image. However, when loaded, the system does not pick up the new parameters, but continues to use persistent networking and iSCSI parameters from the previous boot. As a result, the host cannot connect to the target and boots in the stateless mode.

Solution

- 1 Navigate to the host.

- 2 Reconfigure the iSCSI and networking parameters on the host, so that they match the iBFT parameters.
- 3 Perform a rescan.

Best Practices for iSCSI Storage

13

When using ESXi with the iSCSI SAN, follow recommendations that VMware offers to avoid problems.

Check with your storage representative if your storage system supports Storage API - Array Integration hardware acceleration features. If it does, refer to your vendor documentation to enable hardware acceleration support on the storage system side. For more information, see [Chapter 24 Storage Hardware Acceleration](#).

Read the following topics next:

- [Preventing iSCSI SAN Problems](#)
- [Optimizing iSCSI SAN Storage Performance](#)
- [Checking Ethernet Switch Statistics](#)

Preventing iSCSI SAN Problems

When using ESXi with a SAN, you must follow specific guidelines to avoid SAN problems.

Observe the following tips:

- Place only one VMFS datastore on each LUN.
- Do not change the path policy the system sets for you unless you understand the implications of making such a change.
- Document everything. Include information about configuration, access control, storage, switch, server and iSCSI HBA configuration, software and firmware versions, and storage cable plan.
- Plan for failure:
 - Make several copies of your topology maps. For each element, consider what happens to your SAN if the element fails.
 - Cross off different links, switches, HBAs, and other elements to ensure that you did not miss a critical failure point in your design.
- Ensure that the iSCSI HBAs are installed in the correct slots in the ESXi host, based on slot and bus speed. Balance PCI bus load among the available buses in the server.

- Become familiar with the various monitor points in your storage network, at all visibility points, including ESXi performance charts, Ethernet switch statistics, and storage performance statistics.
- Change LUN IDs only when VMFS datastores deployed on the LUNs have no running virtual machines. If you change the ID, virtual machines running on the VMFS datastore might fail. After you change the ID of the LUN, you must rescan your storage to reset the ID on your host. For information on using the rescan, see [Storage Rescan Operations](#).
- If you change the default iSCSI name of your iSCSI adapter, make sure that the name you enter is worldwide unique and properly formatted. To avoid storage access problems, never assign the same iSCSI name to different adapters, even on different hosts.

Optimizing iSCSI SAN Storage Performance

Several factors contribute to optimizing a typical SAN environment.

If the network environment is properly configured, the iSCSI components provide adequate throughput and low enough latency for iSCSI initiators and targets. If the network is congested and links, switches or routers are saturated, iSCSI performance suffers and might not be adequate for ESXi environments.

Storage System Performance

Storage system performance is one of the major factors contributing to the performance of the entire iSCSI environment.

If issues occur with storage system performance, consult your storage system vendor's documentation for any relevant information.

When you assign LUNs, remember that you can access each shared LUN through a number of hosts, and that a number of virtual machines can run on each host. One LUN used by the ESXi host can service I/O from many different applications running on different operating systems. Because of this diverse workload, the RAID group that contains the ESXi LUNs should not include LUNs that other hosts use that are not running ESXi for I/O intensive applications.

Enable read caching and write caching.

Load balancing is the process of spreading server I/O requests across all available SPs and their associated host server paths. The goal is to optimize performance in terms of throughput (I/O per second, megabytes per second, or response times).

SAN storage systems require continual redesign and tuning to ensure that I/O is load balanced across all storage system paths. To meet this requirement, distribute the paths to the LUNs among all the SPs to provide optimal load balancing. Close monitoring indicates when it is necessary to manually rebalance the LUN distribution.

Tuning statically balanced storage systems is a matter of monitoring the specific performance statistics (such as I/O operations per second, blocks per second, and response time) and distributing the LUN workload to spread the workload across all the SPs.

Server Performance with iSCSI

To ensure optimal ESXi host performance, consider several factors.

Each server application must have access to its designated storage with the following conditions:

- High I/O rate (number of I/O operations per second)
- High throughput (megabytes per second)
- Minimal latency (response times)

Because each application has different requirements, you can meet these goals by selecting an appropriate RAID group on the storage system.

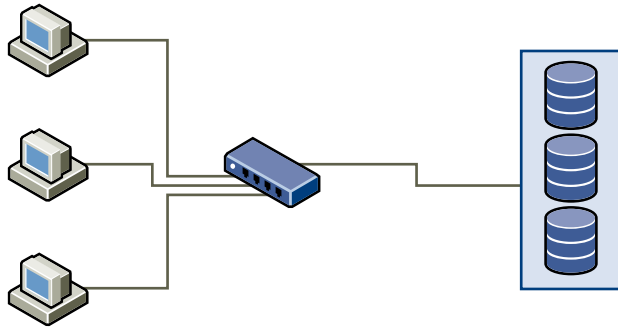
To achieve performance goals, follow these guidelines:

- Place each LUN on a RAID group that provides the necessary performance levels. Monitor the activities and resource use of other LUNS in the assigned RAID group. A high-performance RAID group that has too many applications doing I/O to it might not meet performance goals required by an application running on the ESXi host.
- To achieve maximum throughput for all the applications on the host during the peak period, install enough network adapters or iSCSI hardware adapters. I/O spread across multiple ports provides faster throughput and less latency for each application.
- To provide redundancy for software iSCSI, make sure that the initiator is connected to all network adapters used for iSCSI connectivity.
- When allocating LUNs or RAID groups for ESXi systems, remember that multiple operating systems use and share that resource. The LUN performance required by the ESXi host might be much higher than when you use regular physical machines. For example, if you expect to run four I/O intensive applications, allocate four times the performance capacity for the ESXi LUNs.
- When you use multiple ESXi systems with vCenter Server, the storage performance requirements increase.
- The number of outstanding I/Os needed by applications running on an ESXi system must match the number of I/Os the SAN can handle.

Network Performance

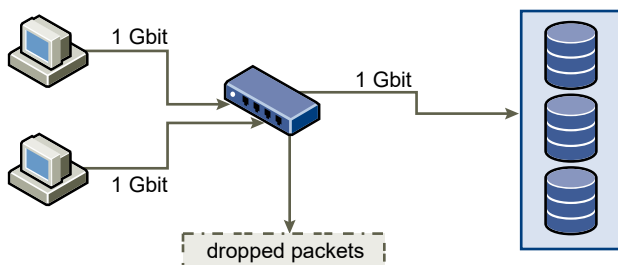
A typical SAN consists of a collection of computers connected to a collection of storage systems through a network of switches. Several computers often access the same storage.

The following graphic shows several computer systems connected to a storage system through an Ethernet switch. In this configuration, each system is connected through a single Ethernet link to the switch. The switch is connected to the storage system through a single Ethernet link.

Figure 13-1. Single Ethernet Link Connection to Storage

When systems read data from storage, the storage responds with sending enough data to fill the link between the storage systems and the Ethernet switch. It is unlikely that any single system or virtual machine gets full use of the network speed. However, this situation can be expected when many systems share one storage device.

When writing data to storage, multiple systems or virtual machines might attempt to fill their links. As a result, the switch between the systems and the storage system might drop network packets. The data drop might occur because the switch has more traffic to send to the storage system than a single link can carry. The amount of data the switch can transmit is limited by the speed of the link between it and the storage system.

Figure 13-2. Dropped Packets

Recovering from dropped network packets results in large performance degradation. In addition to time spent determining that data was dropped, the retransmission uses network bandwidth that can otherwise be used for current transactions.

iSCSI traffic is carried on the network by the Transmission Control Protocol (TCP). TCP is a reliable transmission protocol that ensures that dropped packets are retried and eventually reach their destination. TCP is designed to recover from dropped packets and retransmits them quickly and seamlessly. However, when the switch discards packets with any regularity, network throughput suffers. The network becomes congested with requests to resend data and with the resent packets. Less data is transferred than in a network without congestion.

Most Ethernet switches can buffer, or store, data. This technique gives every device attempting to send data an equal chance to get to the destination. The ability to buffer some transmissions, combined with many systems limiting the number of outstanding commands, reduces transmissions to small bursts. The bursts from several systems can be sent to a storage system in turn.

If the transactions are large and multiple servers are sending data through a single switch port, an ability to buffer can be exceeded. In this case, the switch drops the data it cannot send, and the storage system must request a retransmission of the dropped packet. For example, if an Ethernet switch can buffer 32 KB, but the server sends 256 KB to the storage device, some of the data is dropped.

Most managed switches provide information on dropped packets, similar to the following:

```
*: interface is up
IHQ: pkts in input hold queue      IQD: pkts dropped from input queue
OHQ: pkts in output hold queue    OQD: pkts dropped from output queue
RXBS: rx rate (bits/sec)          RXPS: rx rate (pkts/sec)
TXBS: tx rate (bits/sec)          TXPS: tx rate (pkts/sec)
TRTL: throttle count
```

Table 13-1. Sample Switch Information

| Interface | IHQ | IQD | OHQ | OQD | RXBS | RXPS | TXBS | TXPS | TRTL |
|-------------------------|-----|------|-----|-----|-----------|-------|---------|-------|------|
| * GigabitEthernet0/1 | 3 | 9922 | 0 | 0 | 476303000 | 62273 | 4778400 | 63677 | 0 |

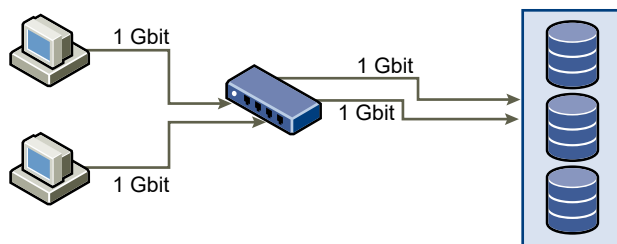
In this example from a Cisco switch, the bandwidth used is 476303000 bits/second, which is less than half of wire speed. The port is buffering incoming packets, but has dropped several packets. The final line of this interface summary indicates that this port has already dropped almost 10,000 inbound packets in the IQD column.

Configuration changes to avoid this problem involve making sure several input Ethernet links are not funneled into one output link, resulting in an oversubscribed link. When several links transmitting near capacity are switched to a smaller number of links, oversubscription becomes possible.

Generally, applications or systems that write much data to storage must avoid sharing Ethernet links to a storage device. These types of applications perform best with multiple connections to storage devices.

Multiple Connections from Switch to Storage shows multiple connections from the switch to the storage.

Figure 13-3. Multiple Connections from Switch to Storage



Using VLANs or VPNs does not provide a suitable solution to the problem of link oversubscription in shared configurations. VLANs and other virtual partitioning of a network provide a way of logically designing a network. However, they do not change the physical capabilities of links and trunks between switches. When storage traffic and other network traffic share physical connections, oversubscription and lost packets might become possible. The same is true of VLANs that share interswitch trunks. Performance design for a SAN must consider the physical limitations of the network, not logical allocations.

Checking Ethernet Switch Statistics

Many Ethernet switches provide different methods for monitoring switch health.

Switches that have ports operating near maximum throughput much of the time do not provide optimum performance. If you have ports in your iSCSI SAN running near the maximum, reduce the load. If the port is connected to an ESXi system or iSCSI storage, you can reduce the load by using manual load balancing.

If the port is connected between multiple switches or routers, consider installing additional links between these components to handle more load. Ethernet switches also commonly provide information about transmission errors, queued packets, and dropped Ethernet packets. If the switch regularly reports any of these conditions on ports being used for iSCSI traffic, performance of the iSCSI SAN will be poor.

Managing Storage Devices

14

Manage local and networked storage device that your ESXi host has access to.

Read the following topics next:

- [Storage Device Characteristics](#)
- [Storage Device Names and Identifiers](#)
- [Storage Rescan Operations](#)
- [Identifying Device Connectivity Problems](#)
- [Enable or Disable the Locator LED on Storage Devices](#)
- [Erase Storage Devices](#)

Storage Device Characteristics

When your ESXi host connects to block-based storage systems, LUNs or storage devices that support ESXi become available to the host.

After the devices get registered with your host, you can display all available local and networked devices and review their information. If you use third-party multipathing plug-ins, the storage devices available through the plug-ins also appear on the list.

Note If an array supports implicit asymmetric logical unit access (ALUA) and has only standby paths, the registration of the device fails. The device can register with the host after the target activates a standby path and the host detects it as active. The advanced system `/Disk/FailDiskRegistration` parameter controls this behavior of the host.

For each storage adapter, you can display a separate list of storage devices available for this adapter.

Generally, when you review storage devices, you see the following information.

Table 14-1. Storage Device Information

| Storage Device Information | Description |
|----------------------------|--|
| Name | Also called Display Name. It is a name that the ESXi host assigns to the device based on the storage type and manufacturer. Generally, you can change this name to a name of your choice. See Rename Storage Devices . |
| Identifier | A universally unique identifier that is intrinsic to the device. See Storage Device Names and Identifiers . |
| Operational State | Indicates whether the device is attached or detached. See Detach Storage Devices . |
| LUN | Logical Unit Number (LUN) within the SCSI target. The LUN number is provided by the storage system. If a target has only one LUN, the LUN number is always zero (0). |
| Type | Type of device, for example, disk or CD-ROM. |
| Drive Type | Information about whether the device is a flash drive or a regular HDD drive. For information about flash drives and NVMe devices, see Chapter 15 Working with Flash Devices . |
| Transport | Transportation protocol your host uses to access the device. The protocol depends on the type of storage being used. See Types of Physical Storage . |
| Capacity | Total capacity of the storage device. |
| Owner | The plug-in, such as the NMP or a third-party plug-in, that the host uses to manage paths to the storage device. See Pluggable Storage Architecture and Path Management . |
| Hardware Acceleration | Information about whether the storage device assists the host with virtual machine management operations. The status can be Supported, Not Supported, or Unknown. See Chapter 24 Storage Hardware Acceleration . |
| Sector Format | Indicates whether the device uses a traditional, 512n, or advanced sector format, such as 512e or 4Kn. See Device Sector Formats . |
| Location | A path to the storage device in the <code>/vmfs/devices/</code> directory. |
| Partition Format | A partition scheme used by the storage device. It can be of a master boot record (MBR) or GUID partition table (GPT) format. The GPT devices can support datastores greater than 2 TB. See Device Sector Formats . |
| Partitions | Primary and logical partitions, including a VMFS datastore, if configured. |
| Multipathing Policies | Path Selection Policy and Storage Array Type Policy the host uses to manage paths to storage. See Chapter 18 Understanding Multipathing and Failover . |
| Paths | Paths used to access storage and their status. See Disable Storage Paths . |

Display Storage Devices for a Host

Display all storage devices available to a host. If you use any third-party multipathing plug-ins, the storage devices available through the plug-ins also appear on the list.

The Storage Devices view allows you to list the hosts' storage devices, analyze their information, and modify properties.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Devices**.

All storage devices available to the host are listed in the Storage Devices table.

- 4 To view details for a specific device, select the device from the list.
- 5 Use the icons to perform basic storage management tasks.

Availability of specific icons depends on the device type and configuration.

| Icon | Description |
|--------------------|---|
| Refresh | Refresh information about storage adapters, topology, and file systems. |
| Detach | Detach the selected device from the host. |
| Attach | Attach the selected device to the host. |
| Rename | Change the display name of the selected device. |
| Turn On LED | Turn on the locator LED for the selected devices. |
| Turn Off LED | Turn off the locator LED for the selected devices. |
| Mark as Flash Disk | Mark the selected devices as flash disks. |
| Mark as HDD Disk | Mark the selected devices as HDD disks. |
| Mark as Local | Mark the selected devices as local for the host. |
| Mark as Remote | Mark the selected devices as remote for the host. |
| Erase Partitions | Erase partitions on the selected devices. |

- 6 Use the following tabs to access additional information and modify properties for the selected device.

| Tab | Description |
|--------------------------|---|
| Properties | View device properties and characteristics. View and modify multipathing policies for the device. |
| Paths | Display paths available for the device. Disable or enable a selected path. |
| Partition Details | Displays information about partitions and their formats. |

Display Storage Devices for an Adapter

Display a list of storage devices accessible through a specific storage adapter on the host.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.

3 Under **Storage**, click **Storage Adapters**.

All storage adapters installed on the host are listed in the Storage Adapters table.

4 Select the adapter from the list and click the **Devices** tab.

Storage devices that the host can access through the adapter are displayed.

5 Use the icons to perform basic storage management tasks.

Availability of specific icons depends on the device type and configuration.

| Icon | Description |
|--------------------|---|
| Refresh | Refresh information about storage adapters, topology, and file systems. |
| Detach | Detach the selected device from the host. |
| Attach | Attach the selected device to the host. |
| Rename | Change the display name of the selected device. |
| Turn On LED | Turn on the locator LED for the selected devices. |
| Turn Off LED | Turn off the locator LED for the selected devices. |
| Mark as Flash Disk | Mark the selected devices as flash disks. |
| Mark as HDD Disk | Mark the selected devices as HDD disks. |
| Mark as Local | Mark the selected devices as local for the host. |
| Mark as Remote | Mark the selected devices as remote for the host. |
| Erase Partitions | Erase partitions on the selected devices. |

Device Sector Formats

ESXi supports storage devices with traditional and advanced sector formats. In storage, a sector is a subdivision of a track on a storage disk or device. Each sector stores a fixed amount of data.

This table introduces different storage device formats that ESXi supports.

| Storage Device Format | ESXi Software Emulation | Logical Sector Size | Physical Sector Size | VMFS Datastore |
|--|-------------------------|---------------------|----------------------|---------------------------|
| 512n | N/A | 512 | 512 | VMFS5 and VMFS6 (default) |
| 512e | N/A | 512 | 4096 | VMFS5 and VMFS6 (default) |
| Note Local 512e storage devices do not support VMFS5. | | | | |
| 4Kn | 512 | 4096 | 4096 | VMFS6 |

512-Byte Native Format

ESXi supports traditional 512n storage devices that use a native 512-bytes sector size.

512-Byte Emulation Format

Due to the increasing demand for larger capacities, the storage industry has introduced advanced formats, such as 512-byte emulation, or 512e. 512e is the advanced format in which the physical sector size is 4096 bytes, but the logical sector emulates 512-bytes sector size. Storage devices that use the 512e format can support legacy applications and guest operating systems. These devices serve as an intermediate step to 4Kn sector drives.

4K Native Format with Software Emulation

Another advanced format that ESXi supports is the 4Kn sector technology. In the 4Kn devices, both physical and logical sectors are 4096 bytes (4 KiB) in length. The device does not have an emulation layer, but exposes its 4Kn physical sector size directly to ESXi.

ESXi detects and registers the 4Kn devices and automatically emulates them as 512e. The device is presented to upper layers in ESXi as 512e. But the guest operating systems always see it as a 512n device. You can continue using existing VMs with legacy guest OSes and applications on the host with the 4Kn devices.

When you use 4Kn devices, the following considerations apply:

- ESXi supports only local 4Kn SAS and SATA HDDs.
- ESXi does not support 4Kn SSD and NVMe devices, or 4Kn devices as RDMs.
- ESXi can boot only from a 4Kn device with UEFI.
- You can use the 4Kn device to configure a coredump partition and coredump file.
- Only the NMP plug-in can claim the 4Kn devices. You cannot use the HPP to claim these devices.
- With vSAN, you can use only the 4Kn capacity HDDs for vSAN Hybrid Arrays. For information, see the *Administering VMware vSAN* documentation.
- Due to the software emulation layer, the performance of the 4Kn devices depends on the alignment of the I/Os. For best performance, run workloads that issue mostly 4K aligned I/Os.
- Workloads accessing the emulated 4Kn device directly using scatter-gather I/O (SGIO) must issue I/Os compatible with the 512e disk.

Example: Determine Device Format

To determine whether the device uses the 512n, 512e, or 4Kn format, run the following command.

```
esxcli storage core device capacity list
```

The following sample output shows the format type.

| Device | Physical Blocksize | Logical Blocksize | Logical Block Count |
|--------------------|--------------------|-------------------|----------------------|
| Size Format Type | | | |
| naa.5000xxxxxxx36f | 512 | 512 | 2344225968 1144641 |

| | | | | |
|---------------------|------|-----|------------|---------|
| MiB 512n | | | | |
| naa.5000xxxxxxxx030 | 4096 | 512 | 3516328368 | 1716957 |
| MiB 4Kn SWE | | | | |
| naa.5000xxxxxxxx8df | 512 | 512 | 2344225968 | 1144641 |
| MiB 512n | | | | |
| naa.5000xxxxxxxx4f4 | 4096 | 512 | 3516328368 | 1716957 |
| MiB 4Kn SWE | | | | |

Storage Device Names and Identifiers

In the ESXi environment, each storage device is identified by several names.

Device Identifiers

Depending on the type of storage, the ESXi host uses different algorithms and conventions to generate an identifier for each storage device.

Storage-provided identifiers

The ESXi host queries a target storage device for the device name. From the returned metadata, the host extracts or generates a unique identifier for the device. The identifier is based on specific storage standards, is unique and persistent across all hosts, and has one of the following formats:

- naa.xxx
- eui.xxx
- t10.xxx

Path-based identifier

When the device does not provide an identifier, the host generates an `mpx.path` name, where *path* represents the first path to the device, for example, `mpx.vmhba1:C0:T1:L3`. This identifier can be used in the same way as the storage-provided identifiers.

The `mpx.path` identifier is created for local devices on the assumption that their path names are unique. However, this identifier is not unique or persistent, and can change after every system restart.

Typically, the path to the device has the following format:

vmhbaAdapter:CChannel:TTarget:LLUN

- *vmhbaAdapter* is the name of the storage adapter. The name refers to the physical adapter on the host, not to the SCSI controller used by the virtual machines.
- *CChannel* is the storage channel number.

Software iSCSI adapters and dependent hardware adapters use the channel number to show multiple paths to the same target.

- *T* *Target* is the target number. Target numbering is determined by the host and might change when the mappings of targets visible to the host change. Targets that are shared by different hosts might not have the same target number.
- *LLUN* is the LUN number that shows the position of the LUN within the target. The LUN number is provided by the storage system. If a target has only one LUN, the LUN number is always zero (0).

For example, `vmhba1:C0:T3:L1` represents LUN1 on target 3 accessed through the storage adapter `vmhba1` and channel 0.

Legacy identifier

In addition to the device-provided identifiers or `mpx.path` identifiers, ESXi generates an alternative legacy name for each device. The identifier has the following format:

`vml.number`

The legacy identifier includes a series of digits that are unique to the device. The identifier can be derived in part from the metadata obtained through the SCSI INQUIRY command. For nonlocal devices that do not provide SCSI INQUIRY identifiers, the `vml.number` identifier is used as the only available unique identifier.

Example: Displaying Device Names in the vSphere CLI

You can use the `esxcli storage core device list` command to display all device names in the vSphere CLI. The output is similar to the following example:

```
# esxcli storage core device list
naa.XXX
    Display Name: DGC Fibre Channel Disk(naa.XXX)
    ...
    Other UIDs: vml.000XXX
mpx.vmhba1:C0:T0:L0
    Display Name: Local VMware Disk (mpx.vmhba1:C0:T0:L0)
    ...
    Other UIDs: vml.0000000000XYZ
```

NVMe Devices with NGUID Device Identifiers

For NVMe devices, ESXi generates device identifiers based on the information it retrieves from the devices. Generally, the NVMe devices support identifiers in EUI64 or NGUID formats, or use both formats. NGUID is a Namespace Globally Unique Identifier that uses the EUI64 16-byte designator format.

For the devices that support only NGUID format, the host-generated device identifier changes depending on the version of ESXi. The ESXi host of version 6.7 and earlier created the `t10.xxx_controller_serial_number` identifier. Starting with 6.7 Update 1, the host creates two identifiers: `eui.xxx (NGUID)` as primary and `t10.xxx_controller_serial_number` as alternative primary.

| ID Formats Supported by Device | | Device Identifier Generated by Host | |
|--------------------------------|-----------------|-------------------------------------|---|
| EUI64 ID Format | NGUID ID Format | ESXi 6.7 and earlier | ESXi 6.7 Update 2 |
| yes | yes | t10.xxx_EUI64 | t10.xxx_EUI64 |
| yes | no | t10.xxx_EUI64 | t10.xxx_EUI64 |
| no | yes | t10.xxx_controller_serial_number | eui.xxx (NGUID) as primary ID t10.xxx_controller_serial_number as alternative primary ID |

Note If your host has NGUID-only devices and you upgrade the host to ESXi 6.7 Update 2 from an earlier version, the device identifier changes from `t10.xxx_controller_serial_number` to `eui.xxx (NGUID)` in the entire ESXi environment. If you use the device identifier in any of your customer scripts, you must reflect this format change.

Verify Mapping Between Primary and Alternative Device Identifiers

Use the `esxcli storage core device uidmap list` command to verify the device identifiers. The output is similar to the following:

```
esxcli storage core device uidmap list
eui.0000xyz.....
  Primary UID: eui.0000xyz.....
  Alternative Primary UIDs: t10.0000abc....
  Legacy UID: vml.00000000000766d68....
  Alternative Legacy UIDs: vml.0000000000080906....
```

Upgrade Stateless ESXi Hosts with NGUID-only NVMe Devices to 6.7 Update 2

If your environment contains stateless ESXi hosts of version 6.7 and earlier and includes NVMe devices that support only NGUID format, you use the present workflow to upgrade the hosts to version 6.7 Update 2.

When upgrading your stateless hosts from version 6.7 and earlier to version 6.7 Update 2, perform these steps to retain the storage configuration. If you perform the upgrade without following the instructions, all storage configurations captured in host profiles might not be retained across the upgrade. As a result, you might encounter host profile compliance failures after the upgrade.

Prerequisites

- Your environment contains stateless ESXi 6.7 or earlier hosts.
- The environment includes NVMe devices that support only NGUID format.

Procedure

1 Determine whether the host contains NGUID-only NVMe devices.

- a Verify whether the vendor of the device is NVMe.

Use the following command as an example.

```
# esxcli storage core device list -d eui.f04xxxxxxxx0000000100000001
eui.f04xxxxxxxx0000000100000001
Display Name: Local NVMe Disk (eui.f04xxxxxxxx0000000100000001)
Has Settable Display Name: true
Devfs Path: /vmfs/devices/disks/eui.f04bxxxxxxxx0000000100000001
Vendor: NVMe
```

The `Vendor: NVMe` line indicates that the device is NVMe.

- b Determine which HBA is connected to the NVMe device.

```
# esxcli storage core adapter device list
HBA      Device UID
-----
vmhba2   eui.f04xxxxxxxx0000000100000001
```

- c Obtain the namespace information for the NVMe device using the HBA and the namespace ID.

```
# esxcli nvme device namespace get -A vmhba2 -n 1
Namespace Identify Info:
Namespace Size: 0xe8e088b0 Logical Blocks
Namespace Capacity: 0xe8e088b0 Logical Blocks
. . .
NVM Capacity: 0x1d1c1116000
Namespace Globally Unique Identifier: 0xf04xxxxxxxx0000000100000001
IEEE Extended Unique Identifier: 0x0
```

In the output, for an NGUID-only NVMe device, the field `IEEE Extended Unique Identifier` contains 0 and `Namespace Globally Unique Identifier` contains a non-zero value.

- 2 To retain storage configurations captured in the host profile, perform these steps when upgrading a stateless host to 6.7 Update 2.

- a Before the upgrade, store `esx.conf` in a persistent location.

For example, you can copy the `esx.conf` file to a VMFS datastore.

```
# cp /etc/vmware/esx.conf /vmfs/volumes/datastore1/
```

- b Upgrade the host.

After the upgrade, the host is not compliant with the profile and might remain in maintenance mode.

- c Apply the device settings for NGUID-only NVMe devices using new ID formats.

Run the following command from the host indicating the location of the `esx.conf` file.

```
# python ./usr/lib/vmware/nvme-nguid-support/bin/nguidApplySettings.pyc -l /vmfs/volumes/datastore1/
```

- 3 Copy the settings from the host and reset host customizations.

- a In vCenter Server, click **Home > Policies and Profiles > Host Profiles**, and click the profile attached to the host.
 - b Click **Configure tab > Copy Setting from Host** and select the host.
 - c To reset customizations, navigate to the host in the vCenter Server browser, and select **Host Profiles > Reset Host Customizations** from the right-click menu.

- 4 From the host's right-click menu, select **Host Profiles > Remediate**.

The host becomes compliant.

- 5 Reboot the host and exit the maintenance mode.

- 6 If the host is still not compliant, repeat [Step 4](#).

Example: Upgrade the ESXi Host Without Retaining Storage Configurations

If you don't need to retain the storage configurations captured in the host profile, after you upgrade the host, you might encounter some compliance failures on the host. In this case, copy the settings from the host and reset host customizations. See [Step 3](#).

Rename Storage Devices

The ESXi host assigns a display name to a storage device based on the storage type and manufacturer. You can change the display name of the device.

You cannot rename certain types of local devices.

Procedure

- 1 Navigate to the host.

- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Devices**.
- 4 Select the device to rename and click **Rename**.
- 5 Change the device name to a friendly name.

Storage Rescan Operations

When you perform storage management tasks or make changes in the SAN configuration, you might need to rescan your storage.

When you perform VMFS datastore management operations, such as creating a VMFS datastore or RDM, adding an extent, and increasing or deleting a VMFS datastore, your host or the vCenter Server automatically rescans and updates your storage. You can disable the automatic rescan feature by turning off the Host Rescan Filter. See [Turn Off Storage Filters](#).

In certain cases, you need to perform a manual rescan. You can rescan all storage available to your host or to all hosts in a folder, cluster, and data center.

If the changes you make are isolated to storage connected through a specific adapter, perform a rescan for this adapter.

Perform the manual rescan each time you make one of the following changes.

- Zone a new disk array on a SAN.
- Create new LUNs on a SAN.
- Change the path masking on a host.
- Reconnect a cable.
- Change CHAP settings (iSCSI only).
- Add or remove discovery or static addresses (iSCSI only).
- Add a single host to the vCenter Server after you have edited or removed from the vCenter Server a datastore shared by the vCenter Server hosts and the single host.

Important If you rescan when a path is unavailable, the host removes the path from the list of paths to the device. The path reappears on the list as soon as it becomes available and starts working again.

Perform Storage Rescan

When you make changes in your SAN configuration, you might need to rescan your storage. You can rescan all storage available to your host, cluster, or data center. If the changes you make are isolated to storage accessed through a specific host, perform the rescan for only this host.

Procedure

- 1 In the vSphere Client object navigator, browse to a host, a cluster, a data center, or a folder that contains hosts.
- 2 From the right-click menu, select **Storage > Rescan Storage**.
- 3 Specify extent of rescan.

| Option | Description |
|------------------------------|--|
| Scan for New Storage Devices | Rescan all adapters to discover new storage devices. If new devices are discovered, they appear in the device list. |
| Scan for New VMFS Volumes | Rescan all storage devices to discover new datastores that have been added since the last scan. Any new datastores appear in the datastore list. |

Perform Adapter Rescan

When you make changes in your SAN configuration and these changes are isolated to storage accessed through a specific adapter, perform rescan for only this adapter.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Adapters**, and select the adapter to rescan from the list.
- 4 Click the **Rescan Adapter** icon.

Change the Number of Scanned Storage Devices

The range of scanned LUN IDs for an ESXi host can be from 0 to 16,383. ESXi ignores LUN IDs greater than 16,383. The configurable `Disk.MaxLUN` parameter controls the range of scanned LUN ID range. The parameter has a default value of 1024.

The `Disk.MaxLUN` parameter also determines how many LUNs the SCSI scan code attempts to discover using individual INQUIRY commands if the SCSI target does not support direct discovery using REPORT_LUNS.

You can modify the `Disk.MaxLUN` parameter depending on your needs. For example, if your environment has a smaller number of storage devices with LUN IDs from 1 through 100, set the value to 101. As a result, you can improve device discovery speed on targets that do not support REPORT_LUNS. Lowering the value can shorten the rescan time and boot time. However, the time to rescan storage devices might also depend on other factors, including the type of the storage system and the load on the storage system.

In other cases, you might need to increase the value if your environment uses LUN IDs that are greater than 1023.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **System**, click **Advanced System Settings**.
- 4 In the Advanced System Settings table, select **Disk.MaxLUN** and click the **Edit** icon.
- 5 Change the existing value to the value of your choice, and click **OK**.

The value you enter specifies the LUN ID that is after the last one you want to discover.

For example, to discover LUN IDs from 1 through 100, set **Disk.MaxLUN** to 101.

Identifying Device Connectivity Problems

When your ESXi host experiences a problem while connecting to a storage device, the host treats the problem as permanent or temporary depending on certain factors.

Storage connectivity problems are caused by a variety of reasons. Although ESXi cannot always determine the reason for a storage device or its paths being unavailable, the host differentiates between a permanent device loss (PDL) state of the device and a transient all paths down (APD) state of storage.

Permanent Device Loss (PDL)

A condition that occurs when a storage device permanently fails or is administratively removed or excluded. It is not expected to become available. When the device becomes permanently unavailable, ESXi receives appropriate sense codes or a login rejection from storage arrays, and is able to recognize that the device is permanently lost.

All Paths Down (APD)

A condition that occurs when a storage device becomes inaccessible to the host and no paths to the device are available. ESXi treats this as a transient condition because typically the problems with the device are temporary and the device is expected to become available again.

Detecting PDL Conditions

A storage device is considered to be in the permanent device loss (PDL) state when it becomes permanently unavailable to your ESXi host.

Typically, the PDL condition occurs when a device is unintentionally removed, or its unique ID changes, or when the device experiences an unrecoverable hardware error.

When the storage array determines that the device is permanently unavailable, it sends SCSI sense codes to the ESXi host. After receiving the sense codes, your host recognizes the device as failed and registers the state of the device as PDL. For the device to be considered permanently lost, the sense codes must be received on all its paths.

After registering the PDL state of the device, the host stops attempts to reestablish connectivity or to send commands to the device.

The vSphere Client displays the following information for the device:

- The operational state of the device changes to `Lost Communication`.
- All paths are shown as `Dead`.
- Datastores on the device are not available.

If no open connections to the device exist, or after the last connection closes, the host removes the PDL device and all paths to the device. You can disable the automatic removal of paths by setting the advanced host parameter `Disk.AutoremoveOnPDL` to 0.

If the device returns from the PDL condition, the host can discover it, but treats it as a new device. Data consistency for virtual machines on the recovered device is not guaranteed.

Note When a device fails without sending appropriate SCSI sense codes or an iSCSI login rejection, the host cannot detect PDL conditions. In this case, the host continues to treat the device connectivity problems as APD even when the device fails permanently.

Permanent Device Loss and SCSI Sense Codes

The following VMkernel log example of a SCSI sense code indicates that the device is in the PDL state.

```
H:0x0 D:0x2 P:0x0 Valid sense data: 0x5 0x25 0x0 or Logical Unit Not Supported
```

Permanent Device Loss and iSCSI

On iSCSI arrays with a single LUN per target, PDL is detected through iSCSI login failure. An iSCSI storage array rejects your host's attempts to start an iSCSI session with a reason `Target Unavailable`. As with the sense codes, this response must be received on all paths for the device to be considered permanently lost.

Permanent Device Loss and Virtual Machines

After registering the PDL state of the device, the host closes all I/O from virtual machines. vSphere HA can detect PDL and restart failed virtual machines. For more information, see [Device Connectivity Problems and High Availability](#).

Performing Planned Storage Device Removal

When a storage device is malfunctioning, you can avoid permanent device loss (PDL) conditions or all paths down (APD) conditions. Perform a planned removal and reconnection of a storage device.

Planned device removal is an intentional disconnection of a storage device. You might also plan to remove a device for such reasons as upgrading your hardware or reconfiguring your storage devices. When you perform an orderly removal and reconnection of a storage device, you complete a number of tasks.

| Task | Description |
|---|--|
| Migrate virtual machines from the device you plan to detach. | <i>vCenter Server and Host Management</i> |
| Unmount the datastore deployed on the device. | See Unmount Datastores . |
| Detach the storage device. | See Detach Storage Devices . |
| For an iSCSI device with a single LUN per target, delete the static target entry from each iSCSI HBA that has a path to the storage device. | See Remove Dynamic or Static iSCSI Targets . |
| Perform any necessary reconfiguration of the storage device by using the array console. | See your vendor documentation. |
| Reattach the storage device. | See Attach Storage Devices . |
| Mount the datastore and restart the virtual machines. | See Mount Datastores . |

Detach Storage Devices

Safely detach a storage device from your host.

You might need to detach the device to make it inaccessible to your host, when, for example, you perform a hardware upgrade on the storage side.

Prerequisites

- The device does not contain any datastores.
- No virtual machines use the device as an RDM disk.
- The device does not contain a diagnostic partition or a scratch partition.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Devices**.
- 4 Select the device to detach and click the **Detach** icon.

Results

The device becomes inaccessible. The operational state of the device changes to Unmounted.

What to do next

If multiple hosts share the device, detach the device from each host.

Attach Storage Devices

Reattach a storage device that you previously detached.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Devices**.
- 4 Select the detached storage device and click the **Attach** icon.

Results

The device becomes accessible.

Recovering from PDL Conditions

An unplanned permanent device loss (PDL) condition occurs when a storage device becomes permanently unavailable without being properly detached from the ESXi host.

The following items in the vSphere Client indicate that the device is in the PDL state:

- The datastore deployed on the device is unavailable.
- Operational state of the device changes to Lost Communication.
- All paths are shown as Dead.
- A warning about the device being permanently inaccessible appears in the VMkernel log file.

To recover from the unplanned PDL condition and remove the unavailable device from the host, perform the following tasks.

| Task | Description |
|--|---|
| Power off and unregister all virtual machines that are running on the datastore affected by the PDL condition. | See <i>vSphere Virtual Machine Administration</i> . |
| Unmount the datastore. | See Unmount Datastores . |
| Rescan all ESXi hosts that had access to the device. | See Perform Storage Rescan . |
| Note If the rescan is not successful and the host continues to list the device, some pending I/O or active references to the device might still exist. Check for any items that might still have an active reference to the device or datastore. The items include virtual machines, templates, ISO images, raw device mappings, and so on. | |

Handling Transient APD Conditions

A storage device is considered to be in the all paths down (APD) state when it becomes unavailable to your ESXi host for an unspecified time period.

The reasons for an APD state can be, for example, a failed switch or a disconnected storage cable.

In contrast with the permanent device loss (PDL) state, the host treats the APD state as transient and expects the device to be available again.

The host continues to retry issued commands in an attempt to reestablish connectivity with the device. If the host's commands fail the retries for a prolonged period, the host might be at risk of having performance problems. Potentially, the host and its virtual machines might become unresponsive.

To avoid these problems, your host uses a default APD handling feature. When a device enters the APD state, the host turns on a timer. With the timer on, the host continues to retry non-virtual machine commands for a limited time period only.

By default, the APD timeout is set to 140 seconds. This value is typically longer than most devices require to recover from a connection loss. If the device becomes available within this time, the host and its virtual machine continue to run without experiencing any problems.

If the device does not recover and the timeout ends, the host stops its attempts at retries and stops any non-virtual machine I/O. Virtual machine I/O continues retrying. The vSphere Client displays the following information for the device with the expired APD timeout:

- The operational state of the device changes to `Dead` or `Error`.
- All paths are shown as `Dead`.
- Datastores on the device are dimmed.

Even though the device and datastores are unavailable, virtual machines remain responsive. You can power off the virtual machines or migrate them to a different datastore or host.

If later the device paths become operational, the host can resume I/O to the device and end the special APD treatment.

Disable Storage APD Handling

The storage all paths down (APD) handling on your ESXi host is enabled by default. When it is enabled, the host continues to retry nonvirtual machine I/O commands to a storage device in the APD state for a limited time period. When the time period expires, the host stops its retry attempts and terminates any nonvirtual machine I/O. You can disable the APD handling feature on your host.

If you disable the APD handling, the host will indefinitely continue to retry issued commands in an attempt to reconnect to the APD device. This behavior might cause virtual machines on the host to exceed their internal I/O timeout and become unresponsive or fail. The host might become disconnected from vCenter Server.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **System**, click **Advanced System Settings**.
- 4 In the Advanced System Settings table, select the **Misc.APDHandlingEnable** parameter and click the `Edit` icon.

- 5 Change the value to 0.

Results

If you disabled the APD handling, you can reenable it and set its value to 1 when a device enters the APD state. The internal APD handling feature turns on immediately and the timer starts with the current timeout value for each device in APD.

Change Timeout Limits for Storage APD

The timeout parameter controls how many seconds the ESXi host must retry the I/O commands to a storage device in an all paths down (APD) state. You can change the default timeout value.

The timeout period begins immediately after the device enters the APD state. After the timeout ends, the host marks the APD device as unreachable. The host stops its attempts to retry any I/O that is not coming from virtual machines. The host continues to retry virtual machine I/O.

By default, the timeout parameter on your host is set to 140 seconds. You can increase the value of the timeout if, for example, storage devices connected to your ESXi host take longer than 140 seconds to recover from a connection loss.

Note If you change the timeout parameter after the device becomes unavailable, the change does not take effect for that particular APD incident.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **System**, click **Advanced System Settings**.
- 4 In the Advanced System Settings table, select the **Misc.APDTimeout** parameter and click the **Edit** icon.
- 5 Change the default value.

You can enter a value between 20 and 99999 seconds.

Verify the Connection Status of a Storage Device

Use the `esxcli` command to verify the connection status of a particular storage device.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 Run the `esxcli storage core device list -d=device_ID` command.

2 Review the connection status in the `Status:` area.

- `on` - Device is connected.
- `dead` - Device has entered the APD state. The APD timer starts.
- `dead timeout` - The APD timeout has expired.
- `not connected` - Device is in the PDL state.

Device Connectivity Problems and High Availability

When a device enters a Permanent Device Loss (PDL) or an All Paths Down (APD) state, vSphere High Availability (HA) can detect connectivity problems and provide automated recovery for affected virtual machines.

vSphere HA uses VM Component Protection (VMCP) to protect virtual machines running on a host in a vSphere HA cluster against accessibility failures. For more information about VMCP and how to configure responses for datastores and virtual machines when an APD or PDL condition occurs, see the *vSphere Availability* documentation.

Enable or Disable the Locator LED on Storage Devices

Use the locator LED to identify specific storage devices, so that you can locate them among other devices. You can turn the locator LED on or off.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Devices**.
- 4 From the list of storage devices, select one or more disks and enable or disable the locator LED indicator.

| Option | Description |
|---------|-------------------------------------|
| Enable | Click the Turn On LED icon. |
| Disable | Click the Turn Off LED icon. |

Erase Storage Devices

Certain functionalities, such as vSAN or virtual flash resource, require that you use clean devices. You can erase an HHD or flash device and remove all existing data.

Prerequisites

- Make sure that the host is in the connected state.
- Verify that the devices you plan to erase are not in use.

- Required privilege: **Host.Config.Storage**

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Devices**.
- 4 Select one or more devices and click the **Erase Partitions** icon.
- 5 Verify that the partition information you are erasing is not critical.
- 6 Click **OK** to confirm your change.

Working with Flash Devices

15

In addition to the regular storage hard disk drives (HDDs), ESXi supports flash storage devices.

Unlike the regular HDDs that are electromechanical devices containing moving parts, the flash devices use semiconductors as their storage medium and have no moving parts. Typically, the flash devices are resilient and provide faster access to data.

To detect flash devices, ESXi uses an inquiry mechanism based on T10 standards. Check with your vendor whether your storage array supports the ESXi mechanism of flash device detection.

After the host detects the flash devices, you can use them for several tasks and functionalities.

If you use NVMe local flash storage, enable the high-performance plug-in (HPP) to improve your storage performance. See [VMware High Performance Plug-In](#).

Read the following topics next:

- [Using Flash Devices with ESXi](#)
- [Marking Storage Devices](#)
- [Monitor Flash Devices](#)
- [Best Practices for Flash Devices](#)
- [About Virtual Flash Resource](#)
- [Configuring Host Swap Cache](#)

Using Flash Devices with ESXi

In your ESXi environment, you can use flash devices with several functionalities.

Table 15-1. Using Flash Devices with ESXi

| Functionality | Description |
|-------------------------------|---|
| vSAN | vSAN requires flash devices. For more information, see the <i>Administering VMware vSAN</i> documentation. |
| VMFS Datastores | <p>You can create VMFS datastores on flash devices. Use the datastores for the following purposes:</p> <ul style="list-style-type: none"> ■ Store virtual machines. Certain guest operating systems can identify virtual disks stored on these datastores as flash virtual disks. See Identifying Flash Virtual Disks. ■ Allocate datastore space for the ESXi host swap cache. See Configuring Host Swap Cache. |
| Virtual Flash Resource (VFFS) | <p>Set up a virtual flash resource and use it for the following functionalities:</p> <ul style="list-style-type: none"> ■ Use as Virtual Flash Read Cache for your virtual machines. See Chapter 16 About VMware vSphere Flash Read Cache. ■ Allocate the virtual flash resource for the ESXi host swap cache. This method is an alternative way of host cache configuration that uses VFFS volumes instead of VMFS datastores. See Configure Host Swap Cache with Virtual Flash Resource. ■ If required by your vendor, use the virtual flash resource for I/O caching filters. See Chapter 23 Filtering Virtual Machine I/O. |

Identifying Flash Virtual Disks

Guest operating systems can identify virtual disks that reside on flash-based datastores as flash virtual disks.

To verify if this feature is enabled, guest operating systems can use standard inquiry commands such as SCSI VPD Page (B1h) for SCSI devices and ATA IDENTIFY DEVICE (Word 217) for IDE devices.

For linked clones, native snapshots, and delta-disks, the inquiry commands report the virtual flash status of the base disk.

Operating systems can detect that a virtual disk is a flash disk under the following conditions:

- Detection of flash virtual disks is supported on ESXi 5.x and later hosts and virtual hardware version 8 or later.
- Detection of flash virtual disks is supported only on VMFS5 or later.
- If virtual disks are located on shared VMFS datastores with flash device extents, the device must be marked as flash on all hosts.
- For a virtual disk to be identified as virtual flash, all underlying physical extents should be flash-backed.

Marking Storage Devices

You can mark storage devices, which are not properly detected, as local flash devices.

When you configure vSAN or set up a virtual flash resource, your storage environment must include local flash devices.

However, ESXi might not recognize certain storage devices as flash devices when their vendors do not support automatic flash device detection. In other cases, certain devices might not be detected as local, and ESXi marks them as remote. When devices are not recognized as the local flash devices, they are excluded from the list of devices offered for vSAN or virtual flash resource. Marking these devices as local flash makes them available for vSAN and virtual flash resource.

Mark Storage Devices as Flash

If ESXi does not recognize its devices as flash, mark them as flash devices.

ESXi does not recognize certain devices as flash when their vendors do not support automatic flash disk detection. The Drive Type column for the devices shows HDD as their type.

Caution Marking the HDD devices as flash might deteriorate the performance of datastores and services that use them. Mark the devices only if you are certain that they are flash devices.

Prerequisites

Verify that the device is not in use.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Devices**.
- 4 From the list of storage devices, select one or several HDD devices and click the **Mark as Flash Disk** (F) icon.
- 5 Click **Yes** to save your changes.

Results

The type of the devices changes to flash.

What to do next

If the flash device that you mark is shared among multiple hosts, make sure that you mark the device from all hosts that share the device.

Mark Storage Devices as Local

ESXi enables you to mark devices as local. This action is useful in cases when ESXi is unable to determine whether certain devices are local.

Prerequisites

- Make sure that the device is not shared.
- Power off virtual machines that reside on the device and unmount an associated datastore.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Devices**.
- 4 From the list of storage devices, select one or several remote devices and click the **Mark as Local** icon.
- 5 Click **Yes** to save your changes.

Monitor Flash Devices

You can monitor certain critical flash device parameters, including `Media Wearout Indicator`, `Temperature`, and `Reallocated Sector Count`, from an ESXi host.

Use the `esxcli` command to monitor flash devices.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ Display the flash device statistics by running the following command:

```
esxcli storage core device smart get -d=flash_device_ID
```

Best Practices for Flash Devices

Follow these best practices when you use flash devices in vSphere environment.

- Make sure to use the latest firmware with flash devices. Frequently check with your storage vendors for any updates.
- Carefully monitor how intensively you use the flash device and calculate its estimated lifetime. The lifetime expectancy depends on how actively you continue to use the flash device.

Estimate Lifetime of Flash Devices

When working with flash devices, monitor how actively you use them and calculate their estimated lifetime.

Typically, storage vendors provide reliable lifetime estimates for a flash device under ideal conditions. For example, a vendor might guarantee a lifetime of 5 years under the condition of 20 GB writes per day. However, the more realistic life expectancy of the device depends on how many writes per day your ESXi host actually generates. Follow these steps to calculate the lifetime of the flash device.

Prerequisites

Note the number of days passed since the last reboot of your ESXi host. For example, ten days.

Procedure

- 1 Obtain the total number of blocks written to the flash device since the last reboot.

Run the `esxcli storage core device stats get -d=device_ID` command. For example:

```
~ # esxcli storage core device stats get -d t10.aaaaaaaaaaaaaaaa
Device: t10.aaaaaaaaaaaaaaaa
Successful Commands: xxxxxxxx
Blocks Read: xxxxxxxx
Blocks Written: 629145600
Read Operations: xxxxxxxx
```

The Blocks Written item in the output shows the number of blocks written to the device since the last reboot. In this example, the value is 629,145,600. After each reboot, it resets to 0.

- 2** Calculate the total number of writes and convert to GB.

One block is 512 bytes. To calculate the total number of writes, multiply the Blocks Written value by 512, and convert the resulting value to GB.

In this example, the total number of writes since the last reboot is approximately 322 GB.

- 3** Estimate the average number of writes per day in GB.

Divide the total number of writes by the number of days since the last reboot.

If the last reboot was ten days ago, you get 32 GB of writes per day. You can average this number over the time period.

- 4** Estimate lifetime of your device by using the following formula:

vendor provided number of writes per day times vendor provided life span divided by actual average number of writes per day

For example, if your vendor guarantees a lifetime of 5 years under the condition of 20 GB writes per day, and the actual number of writes per day is 30 GB, the life span of your flash device will be approximately 3.3 years.

About Virtual Flash Resource

You can aggregate local flash devices on an ESXi host into a single virtualized caching layer called virtual flash resource.

When you set up the virtual flash resource, you create a new file system, Virtual Flash File System (VFFS). VFFS is a derivative of VMFS, which is optimized for flash devices and is used to group the physical flash devices into a single caching resource pool. As a non-persistent resource, it cannot be used to store virtual machines.

The following vSphere functionalities require the virtual flash resource:

- Virtual machine read cache. See [Chapter 16 About VMware vSphere Flash Read Cache](#).
- Host swap cache. See [Configure Host Swap Cache with Virtual Flash Resource](#).
- I/O caching filters, if required by your vendors. See [Chapter 23 Filtering Virtual Machine I/O](#).

Before setting up the virtual flash resource, make sure that you use devices approved by the *VMware Compatibility Guide*.

Considerations for Virtual Flash Resource

When you configure a virtual flash resource to be used by ESXi hosts and virtual machines, several considerations apply.

- You can have only one virtual flash resource, also called a VFFS volume, on a single ESXi host. The virtual flash resource is managed only at the host's level.
- You cannot use the virtual flash resource to store virtual machines. Virtual flash resource is a caching layer only.
- You can use only local flash devices for the virtual flash resource.
- You can create the virtual flash resource from mixed flash devices. All device types are treated the same and no distinction is made between SAS, SATA, or PCI express connectivity. When creating the resource from mixed flash devices, make sure to group similar performing devices together to maximize performance.
- You cannot use the same flash devices for the virtual flash resource and vSAN. Each requires its own exclusive and dedicated flash device.
- The total available capacity of the virtual flash resource can be used by ESXi hosts as host swap cache and by virtual machines as read cache.
- You cannot select individual flash devices for swap cache or read cache. All flash devices are combined into a single flash resource entity.

Set Up Virtual Flash Resource

You can set up a virtual flash resource or add capacity to existing virtual flash resource.

To set up a virtual flash resource, you use local flash devices connected to your host. To increase the capacity of your virtual flash resource, you can add more devices, up to the maximum number indicated in the *Configuration Maximums* documentation. An individual flash device must be exclusively allocated to the virtual flash resource. No other vSphere service, such as vSAN or VMFS, can share the device with the virtual flash resource.

Note This task is available only in the vSphere Web Client.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under Virtual Flash, select **Virtual Flash Resource Management** and click **Add Capacity**.
- 4 From the list of available flash devices, select one or more devices to use for the virtual flash resource and click **OK**.

Under certain circumstances, you might not be able to see flash devices on the list. For more information, see [Marking Storage Devices](#).

Results

The virtual flash resource is created. The Device Backing area lists all devices that you use for the virtual flash resource.

What to do next

You can use the virtual flash resource for cache configuration on the host and Flash Read Cache configuration on virtual disks. In addition, I/O caching filters developed through vSphere APIs for I/O Filtering might require the virtual flash resource.

You can increase the capacity by adding more flash devices to the virtual flash resource.

Remove Virtual Flash Resource

You might need to remove a virtual flash resource deployed on local flash devices to free the devices for other services.

Note This task is available only in the vSphere Web Client.

Prerequisites

- Verify that the virtual flash resource is not configured with host swap cache.
- Verify that no virtual machines configured with Flash Read Cache are powered on the host.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.

- 3 Under Virtual Flash, select **Virtual Flash Resource Management** and click **Remove All**.

Results

After you remove the virtual flash resource and erase the flash device, the device is available for other operations.

Virtual Flash Advanced Settings

You can change advanced parameters for a virtual flash resource.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **System**, click **Advanced System Settings**.
- 4 Select the setting to change and click the **Edit** button.

| Parameter | Description |
|---------------------------------------|---|
| VFLASH.ResourceUsageThreshold | The system triggers the Host vFlash resource usage alarm when a virtual flash resource use exceeds the threshold. The default threshold is 80%. You can change the threshold to an appropriate value. The alarm is cleared when the virtual flash resource use drops below the threshold. |
| VFLASH.MaxResourceGBForVmCache | An ESXi host stores Flash Read Cache metadata in RAM. The default limit of total virtual machine cache size on the host is 2 TB. You can reconfigure this setting. You must restart the host for the new setting to take effect. |

- 5 Click **OK**.

Configuring Host Swap Cache

Your ESXi hosts can use a portion of a flash-backed storage entity as a swap cache shared by all virtual machines.

The host-level cache is made up of files on a low-latency disk that ESXi uses as a write-back cache for virtual machine swap files. All virtual machines running on the host share the cache. Host-level swapping of virtual machine pages makes the best use of potentially limited flash device space.

Depending on your environment and licensing package, the following methods of configuring the host-level swap cache are available. Both methods provide similar results.

- You can create a VMFS datastore on a flash device, and then use the datastore to allocate space for the host cache. The host reserves a certain amount of space for swapping to the host cache.

- If you have a vSphere license to set up a virtual flash resource, you can use the resource to configure the swap cache on the host. The host swap cache is allocated from a portion of the virtual flash resource.

Configure Host Cache with VMFS Datastore

Enable your ESXi host to swap to the host cache. You can also change the amount of space allocated for the host cache.

Use this task if you do not have an appropriate license that allows you to set up and manage a virtual flash resource. If you have the license, use the virtual flash resource for host cache configuration.

Prerequisites

Create a flash-backed VMFS datastore. See [Create a VMFS Datastore](#).

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Host Cache Configuration**.
- 4 Select the flash datastore in the list and click the **Edit** icon.
- 5 Allocate appropriate space for host cache.
- 6 Click **OK**.

Configure Host Swap Cache with Virtual Flash Resource

You can reserve a certain amount of virtual flash resource for host swap cache.

Note This task is available only in the vSphere Web Client.

Prerequisites

- 1 Set up a virtual flash resource. [Set Up Virtual Flash Resource](#).
- 2 If your host is in maintenance mode, exit the maintenance mode before you configure a host swap cache

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under Virtual Flash, select **Virtual Flash Host Swap Cache Configuration** and click **Edit**.
- 4 Select the **Enable virtual flash host swap cache** check box.
- 5 Specify the amount of virtual flash resource to reserve for host swap cache.

6 Click **OK**.

About VMware vSphere Flash Read Cache

16

Flash Read Cache™ can accelerate virtual machine performance by using host-resident flash devices as a cache.

You can reserve a Flash Read Cache for any individual virtual disk. The Flash Read Cache is created only when a virtual machine is powered on. It is discarded when a virtual machine is suspended or powered off.

When you migrate a virtual machine, you can migrate the cache. By default the cache is migrated if the virtual flash modules on the source and destination hosts are compatible. If you do not migrate the cache, the cache is rewarmed on the destination host.

You can change the size of the cache while a virtual machine is powered on. In this instance, the existing cache is discarded and a new write-through cache is created, which results in a cache warm-up period. The advantage of creating a cache is that the cache size can better match the application's active data.

Flash Read Cache supports write-through or read caching. Write-back or write caching are not supported. Data reads are satisfied from the cache, if present. Data writes are dispatched to the backing storage, such as a SAN or NAS. All data that is read from or written to the backing storage is unconditionally stored in the cache.

Flash Read Cache does not support RDMs in physical compatibility. Virtual compatibility RDMs are supported with Flash Read Cache.

Note Not all workloads benefit with a Flash Read Cache. The performance boost depends on your workload pattern and working set size. Read-intensive workloads with working sets that fit into the cache can benefit from a Flash Read Cache configuration. By configuring Flash Read Cache for your read-intensive workloads, additional I/O resources become available on your shared storage. As a result, performance might increase for other workloads even though they are not configured to use Flash Read Cache.

Read the following topics next:

- [DRS Support for Flash Read Cache](#)
- [vSphere High Availability Support for Flash Read Cache](#)
- [Configure Flash Read Cache for a Virtual Machine](#)
- [Migrate Virtual Machines with Flash Read Cache](#)

DRS Support for Flash Read Cache

DRS supports virtual flash as a resource.

DRS manages virtual machines with Flash Read Cache reservations. Every time DRS runs, it displays the available virtual flash capacity reported by the ESXi host. Each host supports one virtual flash resource. DRS selects a host that has sufficient available virtual flash capacity to start a virtual machine. DRS treats powered-on virtual machines with a Flash Read Cache as soft affinity to their current host and moves them only for mandatory reasons or if necessary to correct host over-utilization.

vSphere High Availability Support for Flash Read Cache

Flash Read Cache is supported by High Availability (HA).

vSphere HA restarts a virtual machine with Flash Read Cache on a target host that meets the Flash Read Cache, CPU, Memory, and overhead reservations. If unreserved flash is insufficient to meet the virtual flash reservation, vSphere HA does not restart a virtual machine. If the target host does not have sufficient virtual flash resource available, reconfigure the virtual machine to reduce or drop the Flash Read Cache.

Configure Flash Read Cache for a Virtual Machine

You can configure a Flash Read Cache for your virtual machine. When you enable Flash Read Cache, you can specify the block size and cache size reservation.

- **Block size** is the minimum number of contiguous bytes that can be stored in the cache. This block size can be larger than the nominal disk block size of 512 bytes, between 4 KB and 1024 KB. If a guest operating system writes a single 512-byte disk block, the surrounding cache block size bytes are cached. Do not confuse the cache block size with the disk block size.
- **Reservation** is a reservation size for cache blocks. There is a minimum number of 256 cache blocks. If the cache block size is 1 MB, then the minimum cache size is 256 MB. If the cache block size is 4 K, then the minimum cache size is 1 MB.

For more information about sizing guidelines, search for the *Performance of vSphere Flash Read Cache in VMware vSphere* white paper on the VMware website.

Prerequisites

Set up a virtual flash resource on your client computer.

Procedure

- 1 Right-click a virtual machine in the inventory and select **Edit Settings**.
- 2 On the **Virtual Hardware** tab, expand **Hard disk** to view the disk menu items.
- 3 Enter a value in the **Virtual Flash Read Cache** text box.

- 4 (Optional) Select a block size and a cache size reservation.

| Client | Steps |
|--------------------|--|
| vSphere Client | You cannot specify block size and cache size reservation in the vSphere Client. |
| vSphere Web Client | <ol style="list-style-type: none"> Next to the Virtual flash read cache text box, click Advanced. The Virtual Flash Read Cache Settings dialog box opens. Click Enable virtual Flash Read Cache. In the Reservation text box, enter the cache size reservation, and select the units from the drop-down menu. From the Block Size drop-down menu, select the block size. Click OK. |

- 5 Click **OK**.

Migrate Virtual Machines with Flash Read Cache

When you migrate a powered on virtual machine from one host to another, you can specify whether or not to migrate Flash Read Cache contents with the virtual disks.

Prerequisites

If you plan to migrate Flash Read Cache contents, configure a sufficient virtual flash resource on the destination host.

Procedure

- Right-click the running virtual machine and select **Migrate**.
- Specify the migration type.

| Option | Description |
|---|--|
| Change compute resource only | Migrate the virtual machines to another host or cluster. |
| Change both compute resource and storage | Migrate the virtual machines to a specific host or cluster and their storage to a specific datastore or datastore cluster. |

- Select the destination host and click **Next**.

- 4 Specify a migration setting for all virtual disks configured with virtual Flash Read Cache. This migration parameter does not appear when you do not change the host, but only change the datastore.

| Flash Read Cache Migration Settings | Description |
|--|--|
| Always migrate the cache contents | Virtual machine migration proceeds only if all of the cache contents can be migrated to the destination host. This option is useful when the cache is small or the cache size closely matches the application's active data. |
| Do not migrate the cache contents | Deletes the write-through cache. Cache is recreated on the destination host. This option is useful when the cache size is large or the cache size is larger than the application's active data. |

- 5 If you have multiple virtual disks with Flash Read Cache, you can adjust the migration setting for each individual disk.
 - a Click **Advanced**.
 - b Select a virtual disk for which you want to modify the migration setting.
 - c From the drop-down menu in the **Virtual Flash Read Cache Migration Setting** column, select an appropriate option.
- 6 Complete your migration configuration and click **Finish**.

What to do next

Verify the successful migration by looking at the **Summary** tab of the virtual machine:

- Make sure that the tab displays the correct IP address of the destination host.
- Make sure that the VM Hardware panel displays correct Virtual Flash Read Cache information for each virtual disk.

Working with Datastores

17

Datastores are logical containers, analogous to file systems, that hide specifics of physical storage and provide a uniform model for storing virtual machine files. Datastores can also be used for storing ISO images, virtual machine templates, and floppy images.

Read the following topics next:

- [Types of Datastores](#)
- [Understanding VMFS Datastores](#)
- [Upgrading VMFS Datastores](#)
- [Understanding Network File System Datastores](#)
- [Creating Datastores](#)
- [Managing Duplicate VMFS Datastores](#)
- [Increase VMFS Datastore Capacity](#)
- [Administrative Operations for Datastores](#)
- [Set Up Dynamic Disk Mirroring](#)
- [Collecting Diagnostic Information for ESXi Hosts on a Storage Device](#)
- [Checking Metadata Consistency with VOMA](#)
- [Configuring VMFS Pointer Block Cache](#)

Types of Datastores

Depending on the storage you use, datastores can be of different types.

vCenter Server and ESXi support the following types of datastores.

Table 17-1. Types of Datastores

| Datastore Type | Description |
|-------------------------|--|
| VMFS (version 5 and 6) | Datastores that you deploy on block storage devices use the vSphere Virtual Machine File System (VMFS) format. VMFS is a special high-performance file system format that is optimized for storing virtual machines. See Understanding VMFS Datastores . |
| NFS (version 3 and 4.1) | An NFS client built into ESXi uses the Network File System (NFS) protocol over TCP/IP to access a designated NFS volume. The volume is located on a NAS server. The ESXi host mounts the volume as an NFS datastore, and uses it for storage needs. ESXi supports versions 3 and 4.1 of the NFS protocol. See Understanding Network File System Datastores . |
| vSAN | vSAN aggregates all local capacity devices available on the hosts into a single datastore shared by all hosts in the vSAN cluster. See the <i>Administering VMware vSAN</i> documentation. |
| Virtual Volumes | Virtual Volumes datastore represents a storage container in vCenter Server and vSphere Client. See Chapter 22 Working with Virtual Volumes . |

Depending on your storage type, some of the following tasks are available for the datastores.

- Create datastores. You can use the vSphere Client to create certain types of datastores.
- Perform administrative operations on the datastores. Several operations, such as renaming a datastore, are available for all types of datastores. Others apply to specific types of datastores.
- Organize the datastores. For example, you can group them into folders according to business practices. After you group the datastores, you can assign the same permissions and alarms on the datastores in the group at one time.
- Add the datastores to datastore clusters. A datastore cluster is a collection of datastores with shared resources and a shared management interface. When you create the datastore cluster, you can use Storage DRS to manage storage resources. For information about datastore clusters, see the *vSphere Resource Management* documentation.

Understanding VMFS Datastores

To store virtual disks, ESXi uses datastores. The datastores are logical containers that hide specifics of physical storage from virtual machines and provide a uniform model for storing the virtual machine files. The datastores that you deploy on block storage devices use the native vSphere Virtual Machine File System (VMFS) format. It is a special high-performance file system format that is optimized for storing virtual machines.

Use the vSphere Client to set up the VMFS datastore in advance on the block-based storage device that your ESXi host discovers. The VMFS datastore can be extended to span over several physical storage devices that include SAN LUNs and local storage. This feature allows you to pool storage and gives you flexibility in creating the datastore necessary for your virtual machines.

You can increase the capacity of the datastore while the virtual machines are running on the datastore. This ability lets you add new space to your VMFS datastores as your virtual machine requires it. VMFS is designed for concurrent access from multiple physical machines and enforces the appropriate access controls on the virtual machine files.

Versions of VMFS Datastores

Several versions of the VMFS file system have been released since its introduction. Currently, ESXi supports VMFS5 and VMFS6.

For all supported VMFS version, ESXi offers complete read and write support. On the supported VMFS datastores, you can create and power on virtual machines.

Table 17-2. Host Access to VMFS Versions

| VMFS | ESXi |
|-------|---|
| VMFS6 | Read and write |
| VMFS5 | Read and write |
| VMFS3 | Not supported. For information about upgrades to VMFS5, see Upgrading VMFS Datastores . |

The following table compares major characteristics of VMFS5 and VMFS6. For additional information, see *Configuration Maximums*.

Table 17-3. Comparing VMFS5 and VMFS6

| Features and Functionalities | VMFS5 | VMFS6 |
|--|---|---------------|
| Access for ESXi hosts version 6.5 and later | Yes | Yes |
| Access for ESXi hosts version 6.0 and earlier | Yes | No |
| Datastores per host | 512 | 512 |
| 512n storage devices | Yes | Yes (default) |
| 512e storage devices | Yes. Not supported on local 512e devices. | Yes (default) |
| 4Kn storage devices | No | Yes |
| Automatic space reclamation | No | Yes |
| Manual space reclamation through the <code>esxcli</code> command. See Manually Reclaim Accumulated Storage Space . | Yes | Yes |
| Space reclamation from guest OS | Limited | Yes |

Table 17-3. Comparing VMFS5 and VMFS6 (continued)

| Features and Functionalities | VMFS5 | VMFS6 |
|---|---|---------------|
| GPT storage device partitioning | Yes | Yes |
| MBR storage device partitioning | Yes For a VMFS5 datastore that has been previously upgraded from VMFS3. | No |
| Storage devices greater than 2 TB for each VMFS extent | Yes | Yes |
| Support for virtual machines with large capacity virtual disks, or disks greater than 2 TB | Yes | Yes |
| Support of small files of 1 KB | Yes | Yes |
| Default use of ATS-only locking mechanisms on storage devices that support ATS. See VMFS Locking Mechanisms . | Yes | Yes |
| Block size | Standard 1 MB | Standard 1 MB |
| Default snapshots | VMFSsparse for virtual disks smaller than 2 TB. SEsparse for virtual disks larger than 2 TB. | SEsparse |
| Virtual disk emulation type | 512n | 512n |
| vMotion | Yes | Yes |
| Storage vMotion across different datastore types | Yes | Yes |
| High Availability and Fault Tolerance | Yes | Yes |
| DRS and Storage DRS | Yes | Yes |
| RDM | Yes | Yes |

When you work with VMFS datastores, consider the following:

- **Datastore Extents.** A spanned VMFS datastore must use only homogeneous storage devices, either 512n, 512e, or 4Kn. The spanned datastore cannot extend over devices of different formats.
- **Block Size.** The block size on a VMFS datastore defines the maximum file size and the amount of space a file occupies. VMFS5 and VMFS6 datastores support the block size of 1 MB.
- **Storage vMotion.** Storage vMotion supports migration across VMFS, vSAN, and Virtual Volumes datastores. vCenter Server performs compatibility checks to validate Storage vMotion across different types of datastores.
- **Storage DRS.** VMFS5 and VMFS6 can coexist in the same datastore cluster. However, all datastores in the cluster must use homogeneous storage devices. Do not mix devices of different formats within the same datastore cluster.

- **Device Partition Formats.** Any new VMFS5 or VMFS6 datastore uses GUID partition table (GPT) to format the storage device. The GPT format enables you to create datastores larger than 2 TB. If your VMFS5 datastore has been previously upgraded from VMFS3, it continues to use the master boot record (MBR) partition format, which is characteristic for VMFS3. Conversion to GPT happens only after you expand the datastore to a size larger than 2 TB.

VMFS Datastores as Repositories

ESXi can format SCSI-based storage devices as VMFS datastores. VMFS datastores primarily serve as repositories for virtual machines.

Note Always have only one VMFS datastore for each LUN.

You can store multiple virtual machines on the same VMFS datastore. Each virtual machine, encapsulated in a set of files, occupies a separate single directory. For the operating system inside the virtual machine, VMFS preserves the internal file system semantics, which ensures correct application behavior and data integrity for applications running in virtual machines.

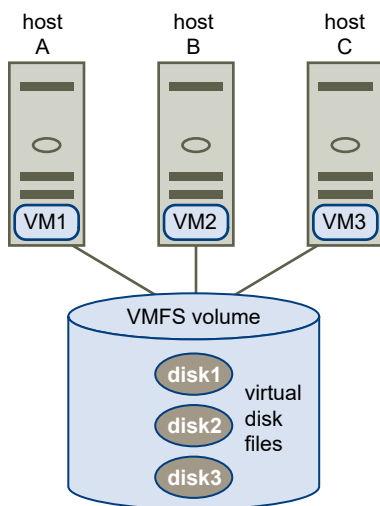
When you run multiple virtual machines, VMFS provides specific locking mechanisms for the virtual machine files. As a result, the virtual machines can operate safely in a SAN environment where multiple ESXi hosts share the same VMFS datastore.

In addition to the virtual machines, the VMFS datastores can store other files, such as the virtual machine templates and ISO images.

Sharing a VMFS Datastore Across Hosts

As a cluster file system, VMFS lets multiple ESXi hosts access the same VMFS datastore concurrently.

Figure 17-1. Sharing a VMFS Datastore Across Hosts



For information on the maximum number of hosts that can connect to a single VMFS datastore, see the *Configuration Maximums* document.

To ensure that multiple hosts do not access the same virtual machine at the same time, VMFS provides on-disk locking.

Sharing the VMFS volume across multiple hosts offers several advantages, for example, the following:

- You can use VMware Distributed Resource Scheduling (DRS) and VMware High Availability (HA).

You can distribute virtual machines across different physical servers. That means you run a mix of virtual machines on each server, so that not all experience high demand in the same area at the same time. If a server fails, you can restart virtual machines on another physical server. If the failure occurs, the on-disk lock for each virtual machine is released. For more information about VMware DRS, see the *vSphere Resource Management* documentation. For information about VMware HA, see the *vSphere Availability* documentation.

- You can use vMotion to migrate running virtual machines from one physical server to another. For information about migrating virtual machines, see the *vCenter Server and Host Management* documentation.

To create a shared datastore, mount the datastore on those ESXi hosts that require the datastore access. See [Mount Datastores](#).

VMFS Metadata Updates

A VMFS datastore holds virtual machine files, directories, symbolic links, RDM descriptor files, and so on. The datastore also maintains a consistent view of all the mapping information for these objects. This mapping information is called metadata.

Metadata is updated each time you perform datastore or virtual machine management operations. Examples of operations requiring metadata updates include the following:

- Creating, growing, or locking a virtual machine file
- Changing attributes of a file
- Powering a virtual machine on or off
- Creating or deleting a VMFS datastore
- Expanding a VMFS datastore
- Creating a template
- Deploying a virtual machine from a template
- Migrating a virtual machine with vMotion

When metadata changes are made in a shared storage environment, VMFS uses special locking mechanisms to protect its data and prevent multiple hosts from concurrently writing to the metadata.

VMFS Locking Mechanisms

In a shared storage environment, when multiple hosts access the same VMFS datastore, specific locking mechanisms are used. These locking mechanisms prevent multiple hosts from concurrently writing to the metadata and ensure that no data corruption occurs.

Depending on its configuration and the type of underlying storage, a VMFS datastore can use different types of locking mechanisms. It can exclusively use the atomic test and set locking mechanism (ATS-only), or use a combination of ATS and SCSI reservations (ATS+SCSI).

ATS-Only Mechanism

For storage devices that support T10 standard-based VAAI specifications, VMFS provides ATS locking, also called hardware assisted locking. The ATS algorithm supports discrete locking per disk sector. All newly formatted VMFS5 and VMFS6 datastores use the ATS-only mechanism if the underlying storage supports it, and never use SCSI reservations.

When you create a multi-extent datastore where ATS is used, vCenter Server filters out non-ATS devices. This filtering allows you to use only those devices that support the ATS primitive.

In certain cases, you might need to turn off the ATS-only setting for a VMFS5 or VMFS6 datastore. For information, see [Change Locking Mechanism to ATS+SCSI](#).

ATS+SCSI Mechanism

A VMFS datastore that supports the ATS+SCSI mechanism is configured to use ATS and attempts to use it when possible. If ATS fails, the VMFS datastore reverts to SCSI reservations. In contrast with the ATS locking, the SCSI reservations lock an entire storage device while an operation that requires metadata protection is performed. After the operation completes, VMFS releases the reservation and other operations can continue.

Datastores that use the ATS+SCSI mechanism include VMFS5 datastores that were upgraded from VMFS3. In addition, new VMFS5 or VMFS6 datastores on storage devices that do not support ATS use the ATS+SCSI mechanism.

If the VMFS datastore reverts to SCSI reservations, you might notice performance degradation caused by excessive SCSI reservations.

Display VMFS Locking Information

Use the `esxcli` command to obtain information about the locking mechanism that a VMFS datastore uses.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ To display information related to VMFS locking mechanisms, run the following command:

```
esxcli storage vmfs lockmode list
```

Results

The table lists items that the output of the command might include.

Table 17-4. VMFS Locking Information

| Fields | Values | Descriptions |
|----------------------------|-----------------------|--|
| Locking Modes | | Indicates the locking configuration of the datastore. |
| | ATS-only | The datastore is configured to use the ATS-only locking mode. |
| | ATS+SCSI | The datastore is configured to use the ATS mode. If ATS fails or is not supported, the datastore can revert to SCSI. |
| | ATS upgrade pending | The datastore is in the process of an online upgrade to the ATS-only mode. |
| | ATS downgrade pending | The datastore is in the process of an online downgrade to the ATS+SCSI mode. |
| ATS Compatible | | Indicates whether the datastore can be or cannot be configured for the ATS-only mode. |
| ATS Upgrade Modes | | Indicates the type of upgrade that the datastore supports. |
| | None | The datastore is not ATS-only compatible. |
| | Online | The datastore can be used during its upgrade to the ATS-only mode. |
| | Offline | The datastore cannot be used during its upgrade to the ATS-only mode. |
| ATS Incompatibility Reason | | If the datastore is not compatible with ATS-only, the item indicates the reason for the incompatibility. |

Change VMFS Locking to ATS-Only

If your VMFS datastore uses the ATS+SCSI locking mechanism, you can change to ATS-only locking.

Typically, VMFS5 datastores that were previously upgraded from VMFS3 continue using the ATS+SCSI locking mechanism. If the datastores are deployed on ATS-enabled hardware, they are eligible for an upgrade to ATS-only locking. Depending on your vSphere environment, you can use one of the following upgrade modes:

- The online upgrade to the ATS-only mechanism is available for most single-extent VMFS5 datastores. While you perform the online upgrade on one of the hosts, other hosts can continue using the datastore.
- The offline upgrade to ATS-only must be used for VMFS5 datastores that span multiple physical extents. Datastores composed of multiple extents are not eligible for the online upgrade. These datastores require that no hosts actively use the datastores at the time of the upgrade request.

Procedure

1 Prepare for an Upgrade to ATS-Only Locking

You must perform several steps to prepare your environment for an online or offline upgrade to ATS-only locking.

2 Upgrade Locking Mechanism to the ATS-Only Type

If a VMFS datastore is ATS-only compatible, you can upgrade its locking mechanism from ATS+SCSI to ATS-only.

Prepare for an Upgrade to ATS-Only Locking

You must perform several steps to prepare your environment for an online or offline upgrade to ATS-only locking.

Procedure

- 1 Upgrade all hosts that access the VMFS5 datastore to the newest version of vSphere.
- 2 Determine whether the datastore is eligible for an upgrade of its current locking mechanism by running the `esxcli storage vmfs lockmode list` command.

The following sample output indicates that the datastore is eligible for an upgrade. It also shows the current locking mechanism and the upgrade mode available for the datastore.

| Locking Mode | ATS Compatible | ATS Upgrade Modes |
|--------------|----------------|-------------------|
| ----- | ----- | ----- |
| ATS+SCSI | true | Online or Offline |

- 3 Depending on the upgrade mode available for the datastore, perform one of the following actions:

| Upgrade Mode | Action |
|--------------|---|
| Online | Verify that all hosts have consistent storage connectivity to the VMFS datastore. |
| Offline | Verify that no hosts are actively using the datastore. |

Upgrade Locking Mechanism to the ATS-Only Type

If a VMFS datastore is ATS-only compatible, you can upgrade its locking mechanism from ATS+SCSI to ATS-only.

Most datastores that do not span multiple extents are eligible for an online upgrade. While you perform the online upgrade on one of the ESXi hosts, other hosts can continue using the datastore. The online upgrade completes only after all hosts have closed the datastore.

Prerequisites

If you plan to complete the upgrade of the locking mechanism by putting the datastore into maintenance mode, disable Storage DRS. This prerequisite applies only to an online upgrade.

Procedure

- 1 Perform an upgrade of the locking mechanism by running the following command:

```
esxcli storage vmfs lockmode set -a|--ats -l|--volume-label= VMFS label -u|--volume-uuid= VMFS UUID.
```

- 2 For an online upgrade, perform additional steps.

- a Close the datastore on all hosts that have access to the datastore, so that the hosts can recognize the change.

You can use one of the following methods:

- Unmount and mount the datastore.
- Put the datastore into maintenance mode and exit maintenance mode.

- b Verify that the Locking Mode status for the datastore changed to ATS-only by running:

```
esxcli storage vmfs lockmode list
```

- c If the Locking Mode displays any other status, for example ATS UPGRADE PENDING, check which host has not yet processed the upgrade by running:

```
esxcli storage vmfs host list
```

Change Locking Mechanism to ATS+SCSI

When you create a VMFS5 datastore on a device that supports the atomic test and set (ATS) locking, the datastore uses the ATS-only locking mechanism. In certain circumstances, you might need to downgrade the ATS-only locking to ATS+SCSI.

You might need to switch to the ATS+SCSI locking mechanism when, for example, your storage device is downgraded. Or when firmware updates fail and the device no longer supports ATS.

The downgrade process is similar to the ATS-only upgrade. As with the upgrade, depending on your storage configuration, you can perform the downgrade in online or offline mode.

Procedure

- 1 Change the locking mechanism to ATS+SCSI by running the following command:

```
esxcli storage vmfs lockmode set -s|--scsi -l|--volume-label= VMFS label
-u|--volume-uuid= VMFS UUID.
```

- 2 For an online mode, close the datastore on all hosts that have access to the datastore, so that the hosts can recognize the change.

Snapshot Formats on VMFS

When you take a snapshot, the state of the virtual disk is preserved, which prevents the guest operating system from writing to it. A delta or child disk is created. The delta represents the difference between the current state of the VM disk and the state that existed when you took the previous snapshot. On the VMFS datastore, the delta disk is a sparse disk.

Sparse disks use the copy-on-write mechanism, in which the virtual disk contains no data, until the data is copied there by a write operation. This optimization saves storage space.

Depending on the type of your datastore, delta disks use different sparse formats.

| Snapshot Formats | VMFS5 | VMFS6 |
|------------------|--------------------------------------|----------------|
| VMFSsparse | For virtual disks smaller than 2 TB. | N/A |
| SEsparse | For virtual disks larger than 2 TB. | For all disks. |

VMFSsparse

VMFS5 uses the VMFSsparse format for virtual disks smaller than 2 TB.

VMFSsparse is implemented on top of VMFS. The VMFSsparse layer processes I/Os issued to a snapshot VM. Technically, VMFSsparse is a redo-log that starts empty, immediately after a VM snapshot is taken. The redo-log expands to the size of its base vmdk, when the entire vmdk is rewritten with new data after the VM snapshotting. This redo-log is a file in the VMFS datastore. Upon snapshot creation, the base vmdk attached to the VM is changed to the newly created sparse vmdk.

SEsparse

SEsparse is a default format for all delta disks on the VMFS6 datastores. On VMFS5, SEsparse is used for virtual disks of the size 2 TB and larger.

SEsparse is a format similar to VMFSsparse with some enhancements. This format is space efficient and supports the space reclamation technique. With space reclamation, blocks that the guest OS deletes are marked. The system sends commands to the SEsparse layer in the hypervisor to unmap those blocks. The unmapping helps to reclaim space allocated by SEsparse once the guest operating system has deleted that data. For more information about space reclamation, see [Storage Space Reclamation](#).

Snapshot Migration

You can migrate VMs with snapshots across different datastores. The following considerations apply:

- If you migrate a VM with the VMFSsparse snapshot to VMFS6, the snapshot format changes to SEsparse.
- When a VM with a vmdk of the size smaller than 2 TB is migrated to VMFS5, the snapshot format changes to VMFSsparse.
- You cannot mix VMFSsparse redo-logs with SEsparse redo-logs in the same hierarchy.

Upgrading VMFS Datastores

ESXi uses different approaches to VMFS5 and VMFS3 upgrades.

VMFS5 Datastores

You cannot upgrade a VMFS5 datastore to VMFS6. If you have a VMFS5 datastore in your environment, create a VMFS6 datastore and migrate virtual machines from the VMFS5 datastore to VMFS6.

VMFS3 Datastores

ESXi no longer supports VMFS3 datastores. The ESXi host automatically upgrades VMFS3 to VMFS5 when mounting existing datastores. The host performs the upgrade operation in the following circumstances:

- At the first boot after an upgrade to ESXi 6.7 or later, when the host mounts all discovered VMFS3 datastores.
- When you manually mount the VMFS3 datastores that are discovered after the boot, or mount persistently unmounted datastores.

When you have VMFS3 datastores in your environment, the following considerations apply:

- If you use an ESXi .iso image to upgrade your legacy host through vSphere Update Manager, and the upgrade is not successful, the VMFS3 datastore is upgraded to VMFS5 if the installation process passes the mount phase.

- In the mixed environment with the ESXi hosts of the 6.5 release and 6.7 and later, the VMFS3 datastore upgrades when the 6.7 host attempts to mount it. The 6.5 host can continue to access the datastore even when the upgrade is unsuccessful.
- When you mount the VMFS3 datastore after its resignaturing, it does not upgrade to VMFS5. You must perform the upgrade manually.

Resolve VMFS3 Upgrade Failures

If the VMFS3 datastore fails to upgrade, it remains unmounted, and its status changes to Failed to Upgrade. Use the following steps to resolve the failure.

Note If the upgrade fails in the mixed environment of the 6.5 and 6.7 and later ESXi hosts, the VMFS3 datastore remains unmounted in the Failed to Upgrade status on the 6.7 host. However, the datastore is accessible from the 6.5 host.

Procedure

- 1 Manually mount the VMFS3 datastore.

See [Mount Datastores](#).

Because the datastore is in the Failed to Upgrade state, the ESXi host does not attempt an upgrade. After a successful mount, the Failed to Upgrade status is cleared.

- 2 Take any necessary corrective actions on the datastore.

For example, if the previous upgrade failed due to lack of space, delete the files from the datastore to make space.

- 3 Repeat the upgrade operation.

See [Upgrade a VMFS3 Datastore to VMFS5](#).

Upgrade a VMFS3 Datastore to VMFS5

After you mount the VMFS3 datastore that failed to upgrade automatically, upgrade it to VMFS5.

Procedure

- 1 Navigate to the VMFS3 datastore.
- 2 Select **Upgrade to VMFS5** from the right-click menu.
- 3 Verify that the hosts accessing the datastore support VMFS5.
- 4 Click **OK** to start the upgrade.
- 5 Perform a rescan on all hosts that are associated with the datastore.

Results

The datastore is upgraded to VMFS5 and is available to all hosts that are associated with the datastore.

Understanding Network File System Datastores

An NFS client built into ESXi uses the Network File System (NFS) protocol over TCP/IP to access a designated NFS volume that is located on a NAS server. The ESXi host can mount the volume and use it for its storage needs. vSphere supports versions 3 and 4.1 of the NFS protocol.

Typically, the NFS volume or directory is created by a storage administrator and is exported from the NFS server. You do not need to format the NFS volume with a local file system, such as VMFS. Instead, you mount the volume directly on the ESXi hosts and use it to store and boot virtual machines in the same way that you use the VMFS datastores.

In addition to storing virtual disks on NFS datastores, you can use NFS as a central repository for ISO images, virtual machine templates, and so on. If you use the datastore for the ISO images, you can connect the CD-ROM device of the virtual machine to an ISO file on the datastore. You then can install a guest operating system from the ISO file.

NFS Protocols and ESXi

ESXi supports NFS protocols version 3 and 4.1. To support both versions, ESXi uses two different NFS clients.

Comparing Versions of NFS Clients

The following table lists capabilities that the NFS version 3 and 4.1 support.

| Characteristics | NFS version 3 | NFS version 4.1 |
|---|-------------------------------|---|
| Security mechanisms | AUTH_SYS | AUTH_SYS and Kerberos (krb5 and krb5i) |
| Encryption algorithms with Kerberos | N/A | AES256-CTS-HMAC-SHA1-96 and AES128-CTS-HMAC-SHA1-96 |
| Multipathing | Not supported | Supported through the session trunking |
| Locking mechanisms | Propriety client-side locking | Server-side locking |
| Hardware acceleration | Supported | Supported |
| Thick virtual disks | Supported | Supported |
| IPv6 | Supported | Supported for AUTH_SYS and Kerberos |
| ISO images presented as CD-ROMs to virtual machines | Supported | Supported |
| Virtual machine snapshots | Supported | Supported |
| Virtual machines with virtual disks greater than 2 TB | Supported | Supported |

NFS Protocols and vSphere Solutions

The following table lists major vSphere solutions that NFS versions support.

| vSphere Features | NFS version 3 | NFS version 4.1 |
|--------------------------------------|---------------|-----------------|
| vMotion and Storage vMotion | Yes | Yes |
| High Availability (HA) | Yes | Yes |
| Fault Tolerance (FT) | Yes | Yes |
| Distributed Resource Scheduler (DRS) | Yes | Yes |
| Host Profiles | Yes | Yes |
| Storage DRS | Yes | No |
| Storage I/O Control | Yes | No |
| Site Recovery Manager | Yes | No |
| Virtual Volumes | Yes | Yes |
| vSphere Replication | Yes | Yes |
| vRealize Operations Manager | Yes | Yes |

NFS 4.1 and Fault Tolerance

Virtual machines on NFS v4.1 support the new Fault Tolerance mechanism introduced in vSphere 6.0.

Virtual machines on NFS v4.1 do not support the old, legacy Fault Tolerance mechanism.

In vSphere 6.0, the newer Fault Tolerance mechanism can accommodate symmetric multiprocessor (SMP) virtual machines with up to four vCPUs. Earlier versions of vSphere used a different technology for Fault Tolerance, with different requirements and characteristics.

NFS Upgrades

When you upgrade ESXi from a version earlier than 6.5, existing NFS 4.1 datastores automatically begin supporting functionalities that were not available in the previous ESXi release. These functionalities include Virtual Volumes, hardware acceleration, and so on.

ESXi does not support automatic datastore conversions from NFS version 3 to NFS 4.1.

If you want to upgrade your NFS 3 datastore, the following options are available:

- Create the NFS 4.1 datastore, and then use Storage vMotion to migrate virtual machines from the old datastore to the new one.
- Use conversion methods provided by your NFS storage server. For more information, contact your storage vendor.

- Unmount the NFS 3 datastore, and then mount as NFS 4.1 datastore.

Caution If you use this option, make sure to unmount the datastore from all hosts that have access to the datastore. The datastore can never be mounted by using both protocols at the same time.

NFS Storage Guidelines and Requirements

When you use NFS storage, follow specific guidelines related to NFS server configuration, networking, NFS datastores, and so on.

What to read next

- [NFS Server Configuration](#)

When you configure NFS servers to work with ESXi, follow recommendation of your storage vendor. In addition to these general recommendations, use specific guidelines that apply to NFS in vSphere environment.

- [NFS Networking](#)

An ESXi host uses TCP/IP network connection to access a remote NAS server. Certain guidelines and best practices exist for configuring the networking when you use NFS storage.

- [NFS File Locking](#)

File locking mechanisms are used to restrict access to data stored on a server to only one user or process at a time. NFS 3 and NFS 4.1 use incompatible file locking mechanisms.

- [NFS Security](#)

With NFS 3 and NFS 4.1, ESXi supports the AUTH_SYS security. In addition, for NFS 4.1, the Kerberos security mechanism is supported.

- [NFS Multipathing](#)

While NFS 3 with ESXi does not provide multipathing support, NFS 4.1 supports multiple paths.

- [NFS and Hardware Acceleration](#)

Virtual disks created on NFS datastores are thin-provisioned by default. To be able to create thick-provisioned virtual disks, you must use hardware acceleration that supports the Reserve Space operation.

- [NFS Datastores](#)

When you create an NFS datastore, make sure to follow specific guidelines.

NFS Server Configuration

When you configure NFS servers to work with ESXi, follow recommendation of your storage vendor. In addition to these general recommendations, use specific guidelines that apply to NFS in vSphere environment.

The guidelines include the following items.

- Make sure that the NAS servers you use are listed in the *VMware HCL*. Use the correct version for the server firmware.
- Ensure that the NFS volume is exported using NFS over TCP.
- Make sure that the NAS server exports a particular share as either NFS 3 or NFS 4.1. The NAS server must not provide both protocol versions for the same share. The NAS server must enforce this policy because ESXi does not prevent mounting the same share through different NFS versions.
- NFS 3 and non-Kerberos (AUTH_SYS) NFS 4.1 do not support the delegate user functionality that enables access to NFS volumes using nonroot credentials. If you use NFS 3 or non-Kerberos NFS 4.1, ensure that each host has root access to the volume. Different storage vendors have different methods of enabling this functionality, but typically the NAS servers use the `no_root_squash` option. If the NAS server does not grant root access, you can still mount the NFS datastore on the host. However, you cannot create any virtual machines on the datastore.
- If the underlying NFS volume is read-only, make sure that the volume is exported as a read-only share by the NFS server. Or mount the volume as a read-only datastore on the ESXi host. Otherwise, the host considers the datastore to be read-write and might not open the files.

NFS Networking

An ESXi host uses TCP/IP network connection to access a remote NAS server. Certain guidelines and best practices exist for configuring the networking when you use NFS storage.

For more information, see the *vSphere Networking* documentation.

- For network connectivity, use a standard network adapter in your ESXi host.
- ESXi supports Layer 2 and Layer 3 Network switches. If you use Layer 3 switches, ESXi hosts and NFS storage arrays must be on different subnets and the network switch must handle the routing information.
- Configure a VMkernel port group for NFS storage. You can create the VMkernel port group for IP storage on an existing virtual switch (vSwitch) or on a new vSwitch. The vSwitch can be a vSphere Standard Switch (VSS) or a vSphere Distributed Switch (VDS).
- If you use multiple ports for NFS traffic, make sure that you correctly configure your virtual switches and physical switches.
- NFS 3 and NFS 4.1 support IPv6.

NFS File Locking

File locking mechanisms are used to restrict access to data stored on a server to only one user or process at a time. NFS 3 and NFS 4.1 use incompatible file locking mechanisms.

NFS 3 locking on ESXi does not use the Network Lock Manager (NLM) protocol. Instead, VMware provides its own locking protocol. NFS 3 locks are implemented by creating lock files on the NFS server. Lock files are named `.lck-file_id..`

NFS 4.1 uses share reservations as a locking mechanism.

Because NFS 3 and NFS 4.1 clients do not use the same locking protocol, you cannot use different NFS versions to mount the same datastore on multiple hosts. Accessing the same virtual disks from two incompatible clients might result in incorrect behavior and cause data corruption.

NFS Security

With NFS 3 and NFS 4.1, ESXi supports the AUTH_SYS security. In addition, for NFS 4.1, the Kerberos security mechanism is supported.

NFS 3 supports the AUTH_SYS security mechanism. With this mechanism, storage traffic is transmitted in an unencrypted format across the LAN. Because of this limited security, use NFS storage on trusted networks only and isolate the traffic on separate physical switches. You can also use a private VLAN.

NFS 4.1 supports the Kerberos authentication protocol to secure communications with the NFS server. Nonroot users can access files when Kerberos is used. For more information, see [Using Kerberos for NFS 4.1](#).

In addition to Kerberos, NFS 4.1 supports traditional non-Kerberos mounts with the AUTH_SYS security. In this case, use root access guidelines for NFS version 3.

Note You cannot use two security mechanisms, AUTH_SYS and Kerberos, for the same NFS 4.1 datastore shared by multiple hosts.

NFS Multipathing

While NFS 3 with ESXi does not provide multipathing support, NFS 4.1 supports multiple paths.

NFS 3 uses one TCP connection for I/O. As a result, ESXi supports I/O on only one IP address or hostname for the NFS server, and does not support multiple paths. Depending on your network infrastructure and configuration, you can use the network stack to configure multiple connections to the storage targets. In this case, you must have multiple datastores, each datastore using separate network connections between the host and the storage.

NFS 4.1 provides multipathing for servers that support the session trunking. When the trunking is available, you can use multiple IP addresses to access a single NFS volume. Client ID trunking is not supported.

NFS and Hardware Acceleration

Virtual disks created on NFS datastores are thin-provisioned by default. To be able to create thick-provisioned virtual disks, you must use hardware acceleration that supports the Reserve Space operation.

NFS 3 and NFS 4.1 support hardware acceleration that allows your host to integrate with NAS devices and use several hardware operations that NAS storage provides. For more information, see [Hardware Acceleration on NAS Devices](#).

NFS Datastores

When you create an NFS datastore, make sure to follow specific guidelines.

The NFS datastore guidelines and best practices include the following items:

- You cannot use different NFS versions to mount the same datastore on different hosts. NFS 3 and NFS 4.1 clients are not compatible and do not use the same locking protocol. As a result, accessing the same virtual disks from two incompatible clients might result in incorrect behavior and cause data corruption.
- NFS 3 and NFS 4.1 datastores can coexist on the same host.
- ESXi cannot automatically upgrade NFS version 3 to version 4.1, but you can use other conversion methods. For information, see [NFS Protocols and ESXi](#).
- When you mount the same NFS 3 volume on different hosts, make sure that the server and folder names are identical across the hosts. If the names do not match, the hosts see the same NFS version 3 volume as two different datastores. This error might result in a failure of such features as vMotion. An example of such discrepancy is entering `filer` as the server name on one host and `filer.domain.com` on the other. This guideline does not apply to NFS version 4.1.
- If you use non-ASCII characters to name datastores and virtual machines, make sure that the underlying NFS server offers internationalization support. If the server does not support international characters, use only ASCII characters, or unpredictable failures might occur.

Firewall Configurations for NFS Storage

ESXi includes a firewall between the management interface and the network. The firewall is enabled by default. At installation time, the ESXi firewall is configured to block incoming and outgoing traffic, except traffic for the default services, such as NFS.

Supported services, including NFS, are described in a rule set configuration file in the ESXi firewall directory `/etc/vmware/firewall/`. The file contains firewall rules and their relationships with ports and protocols.

The behavior of the NFS Client rule set (`nfsClient`) is different from other rule sets.

For more information about firewall configurations, see the *vSphere Security* documentation.

NFS Client Firewall Behavior

The NFS Client firewall rule set behaves differently than other ESXi firewall rule sets. ESXi configures NFS Client settings when you mount or unmount an NFS datastore. The behavior differs for different versions of NFS.

When you add, mount, or unmount an NFS datastore, the resulting behavior depends on the version of NFS.

NFS v3 Firewall Behavior

When you add or mount an NFS v3 datastore, ESXi checks the state of the NFS Client (`nfsClient`) firewall rule set.

- If the `nfsClient` rule set is disabled, ESXi enables the rule set and disables the Allow All IP Addresses policy by setting the `allowedAll` flag to `FALSE`. The IP address of the NFS server is added to the allowed list of outgoing IP addresses.
- If the `nfsClient` rule set is enabled, the state of the rule set and the allowed IP address policy are not changed. The IP address of the NFS server is added to the allowed list of outgoing IP addresses.

Note If you manually enable the `nfsClient` rule set or manually set the Allow All IP Addresses policy, either before or after you add an NFS v3 datastore to the system, your settings are overridden when the last NFS v3 datastore is unmounted. The `nfsClient` rule set is disabled when all NFS v3 datastores are unmounted.

When you remove or unmount an NFS v3 datastore, ESXi performs one of the following actions.

- If none of the remaining NFS v3 datastores are mounted from the server of the datastore being unmounted, ESXi removes the server's IP address from the list of outgoing IP addresses.
- If no mounted NFS v3 datastores remain after the unmount operation, ESXi disables the `nfsClient` firewall rule set.

NFS v4.1 Firewall Behavior

When you mount the first NFS v4.1 datastore, ESXi enables the `nfs41client` rule set and sets its `allowedAll` flag to `TRUE`. This action opens port 2049 for all IP addresses. Unmounting an NFS v4.1 datastore does not affect the firewall state. That is, the first NFS v4.1 mount opens port 2049 and that port remains enabled unless you close it explicitly.

Verify Firewall Ports for NFS Clients

To enable access to NFS storage, ESXi automatically opens firewall ports for the NFS clients when you mount an NFS datastore. For troubleshooting reasons, you might need to verify that the ports are open.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **System**, click **Firewall**, and click **Edit**.
- 4 Scroll down to an appropriate version of NFS to make sure that the port is open.

Using Layer 3 Routed Connections to Access NFS Storage

When you use Layer 3 (L3) routed connections to access NFS storage, consider certain requirements and restrictions.

Ensure that your environment meets the following requirements:

- Use Cisco's Hot Standby Router Protocol (HSRP) in IP Router. If you are using a non-Cisco router, use Virtual Router Redundancy Protocol (VRRP) instead.
- To prioritize NFS L3 traffic on networks with limited bandwidths, or on networks that experience congestion, use Quality of Service (QoS). See your router documentation for details.
- Follow Routed NFS L3 recommendations offered by storage vendor. Contact your storage vendor for details.
- Disable Network I/O Resource Management (NetIORM).
- If you are planning to use systems with top-of-rack switches or switch-dependent I/O device partitioning, contact your system vendor for compatibility and support.

In an L3 environment, the following restrictions apply:

- The environment does not support VMware Site Recovery Manager.
- The environment supports only the NFS protocol. Do not use other storage protocols such as FCoE over the same physical network.
- The NFS traffic in this environment does not support IPv6.
- The NFS traffic in this environment can be routed only over a LAN. Other environments such as WAN are not supported.

Using Kerberos for NFS 4.1

With NFS version 4.1, ESXi supports the Kerberos authentication mechanism.

The RPCSEC_GSS Kerberos mechanism is an authentication service. It allows an NFS 4.1 client installed on ESXi to prove its identity to an NFS server before mounting an NFS share. The Kerberos security uses cryptography to work across an insecure network connection.

The ESXi implementation of Kerberos for NFS 4.1 provides two security models, krb5 and krb5i, that offer different levels of security.

- Kerberos for authentication only (krb5) supports identity verification.
- Kerberos for authentication and data integrity (krb5i), in addition to identity verification, provides data integrity services. These services help to protect the NFS traffic from tampering by checking data packets for any potential modifications.

Kerberos supports cryptographic algorithms that prevent unauthorized users from gaining access to NFS traffic. The NFS 4.1 client on ESXi attempts to use either the AES256-CTS-HMAC-SHA1-96 or AES128-CTS-HMAC-SHA1-96 algorithm to access a share on the NAS server. Before using your NFS 4.1 datastores, make sure that AES256-CTS-HMAC-SHA1-96 or AES128-CTS-HMAC-SHA1-96 are enabled on the NAS server.

The following table compares Kerberos security levels that ESXi supports.

Table 17-5. Types of Kerberos Security

| | | ESXi 6.0 | ESXi 6.5 and later |
|--|-----------------------------------|--------------|--------------------|
| Kerberos for authentication only (krb5) | Integrity checksum for RPC header | Yes with DES | Yes with AES |
| | Integrity checksum for RPC data | No | No |
| Kerberos for authentication and data integrity (krb5i) | Integrity checksum for RPC header | No krb5i | Yes with AES |
| | Integrity checksum for RPC data | | Yes with AES |

When you use Kerberos authentication, the following considerations apply:

- ESXi uses Kerberos with the Active Directory domain.
- As a vSphere administrator, you specify Active Directory credentials to provide access to NFS 4.1 Kerberos datastores for an NFS user. A single set of credentials is used to access all Kerberos datastores mounted on that host.
- When multiple ESXi hosts share the NFS 4.1 datastore, you must use the same Active Directory credentials for all hosts that access the shared datastore. To automate the assignment process, set the user in host profiles and apply the profile to all ESXi hosts.
- You cannot use two security mechanisms, AUTH_SYS and Kerberos, for the same NFS 4.1 datastore shared by multiple hosts.

Set Up NFS Storage Environment

You must perform several configuration steps before you mount an NFS datastore in vSphere.

Prerequisites

- Familiarize yourself with the guidelines in [NFS Storage Guidelines and Requirements](#).
- For details on configuring NFS storage, consult your storage vendor documentation.
- If you use Kerberos, make sure that AES256-CTS-HMAC-SHA1-96 or AES128-CTS-HMAC-SHA1-96 are enabled on the NAS server.

Procedure

- 1 On the NFS server, configure an NFS volume and export it to be mounted on the ESXi hosts.
 - a Note the IP address or the DNS name of the NFS server and the full path, or folder name, for the NFS share.

For NFS 4.1, you can collect multiple IP addresses or DNS names to use the multipathing support that the NFS 4.1 datastore provides.

- b If you plan to use Kerberos authentication with NFS 4.1, specify the Kerberos credentials to be used by ESXi for authentication.
- 2 On each ESXi host, configure a VMkernel Network port for NFS traffic.
- 3 If you plan to use Kerberos authentication with the NFS 4.1 datastore, configure the ESXi hosts for Kerberos authentication.

See [Configure ESXi Hosts for Kerberos Authentication](#).

What to do next

You can now create an NFS datastore on the ESXi hosts.

Configure ESXi Hosts for Kerberos Authentication

If you use NFS 4.1 with Kerberos, you must perform several tasks to set up your hosts for Kerberos authentication.

When multiple ESXi hosts share the NFS 4.1 datastore, you must use the same Active Directory credentials for all hosts that access the shared datastore. You can automate the assignment process by setting the user in host profiles and applying the profile to all ESXi hosts.

Prerequisites

- Make sure that Microsoft Active Directory (AD) and NFS servers are configured to use Kerberos.
- Enable AES256-CTS-HMAC-SHA1-96 or AES128-CTS-HMAC-SHA1-96 encryption modes on AD. The NFS 4.1 client does not support the DES-CBC-MD5 encryption mode.
- Make sure that the NFS server exports are configured to grant full access to the Kerberos user.

Procedure

- 1 [Configure DNS for NFS 4.1 with Kerberos](#)

When you use NFS 4.1 with Kerberos, you must change the DNS settings on ESXi hosts. The settings must point to the DNS server that is configured to hand out DNS records for the Kerberos Key Distribution Center (KDC). For example, use the Active Directory server address if AD is used as a DNS server.

2 Configure Network Time Protocol for NFS 4.1 with Kerberos

If you use NFS 4.1 with Kerberos, configure Network Time Protocol (NTP) to make sure all ESXi hosts on the vSphere network are synchronized.

3 Enable Kerberos Authentication in Active Directory

If you use NFS 4.1 storage with Kerberos, you must add each ESXi host to an Active Directory domain and enable Kerberos authentication. Kerberos integrates with Active Directory to enable single sign-on and provides an extra layer of security when used across an insecure network connection.

What to do next

After you configure your host for Kerberos, you can create an NFS 4.1 datastore with Kerberos enabled.

Configure DNS for NFS 4.1 with Kerberos

When you use NFS 4.1 with Kerberos, you must change the DNS settings on ESXi hosts. The settings must point to the DNS server that is configured to hand out DNS records for the Kerberos Key Distribution Center (KDC). For example, use the Active Directory server address if AD is used as a DNS server.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Networking**, click **TCP/IP configuration**.
- 4 Select **Default** and click the **Edit** icon.
- 5 Manually enter the DNS settings.

| Option | Description |
|----------------------|-----------------------|
| Domain | <i>AD Domain Name</i> |
| Preferred DNS server | <i>AD Server IP</i> |
| Search domains | <i>AD Domain Name</i> |

Configure Network Time Protocol for NFS 4.1 with Kerberos

If you use NFS 4.1 with Kerberos, configure Network Time Protocol (NTP) to make sure all ESXi hosts on the vSphere network are synchronized.

The best practice is to use the Active Domain server as the NTP server.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.

- 3 Under **System**, select **Time Configuration**.
- 4 Click **Edit** and set up the NTP server.
 - a Select **Use Network Time Protocol (Enable NTP client)**.
 - b To synchronize with the NTP server, enter its IP addresses.
 - c Select **Start NTP Service**.
 - d Set the NTP Service Startup Policy.
- 5 Click **OK**.

The host synchronizes with the NTP server.

Enable Kerberos Authentication in Active Directory

If you use NFS 4.1 storage with Kerberos, you must add each ESXi host to an Active Directory domain and enable Kerberos authentication. Kerberos integrates with Active Directory to enable single sign-on and provides an extra layer of security when used across an insecure network connection.

Prerequisites

Set up an AD domain and a domain administrator account with the rights to add hosts to the domain.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **System**, click **Authentication Services**.
- 4 Add the ESXi host to an Active Directory domain.
 - a In the Authentication Services pane, click **Join Domain**.
 - b Supply the domain settings, and click **OK**.

The directory services type changes to Active Directory.

- 5 Configure or edit credentials for an NFS Kerberos user.
 - a In the NFS Kerberos Credentials pane, click **Edit**.
 - b Enter a user name and password.

Files stored in all Kerberos datastores are accessed using these credentials.

The state for NFS Kerberos credentials changes to Enabled.

Creating Datastores

You use the New Datastore wizard to create your datastores. Depending on the type of your storage and storage needs, you can create a VMFS, NFS, or Virtual Volumes datastore.

A vSAN datastore is automatically created when you enable vSAN. For information, see the *Administering VMware vSAN* documentation.

You can also use the New Datastore wizard to manage VMFS datastore copies.

What to read next

- [Create a VMFS Datastore](#)

VMFS datastores serve as repositories for virtual machines. You can set up VMFS datastores on any SCSI-based storage devices that the host discovers, including Fibre Channel, iSCSI, and local storage devices.

- [Create an NFS Datastore](#)

You can use the **New Datastore** wizard to mount an NFS volume.

- [Create a Virtual Volumes Datastore](#)

You use the **New Datastore** wizard to create a Virtual Volumes datastore.

Create a VMFS Datastore

VMFS datastores serve as repositories for virtual machines. You can set up VMFS datastores on any SCSI-based storage devices that the host discovers, including Fibre Channel, iSCSI, and local storage devices.

Prerequisites

- 1 Install and configure any adapters that your storage requires.
- 2 To discover newly added storage devices, perform a rescan. See [Storage Rescan Operations](#).
- 3 Verify that storage devices you are planning to use for your datastores are available. See [Storage Device Characteristics](#).

Procedure

- 1 In the vSphere Client object navigator, browse to a host, a cluster, or a data center.
- 2 From the right-click menu, select **Storage > New Datastore**.
- 3 Select VMFS as the datastore type.
- 4 Enter the datastore name and if necessary, select the placement location for the datastore.
The system enforces a 42 character limit for the datastore name.

- 5 Select the device to use for your datastore.

Important The device you select must not have any values displayed in the Snapshot Volume column. If a value is present, the device contains a copy of an existing VMFS datastore. For information on managing datastore copies, see [Managing Duplicate VMFS Datastores](#).

- 6 Specify the datastore version.

| Option | Description |
|--------------|--|
| VMFS6 | Default format on all hosts that support VMFS6. The ESXi hosts of version 6.0 or earlier cannot recognize the VMFS6 datastore. |
| VMFS5 | VMFS5 datastore supports access by the ESXi hosts of version 6.7 or earlier. |

- 7 Define configuration details for the datastore.

Note The required minimum size for a VMFS6 datastore is 2 GB.

- a Specify partition configuration.

| Option | Description |
|-------------------------------------|---|
| Use all available partitions | Dedicates the entire disk to a single VMFS datastore. If you select this option, all file systems and data currently stored on this device are destroyed. |
| Use free space | Deploys a VMFS datastore in the remaining free space of the disk. |

- b If the space allocated for the datastore is excessive for your purposes, adjust the capacity values in the Datastore Size field.

By default, the entire free space on the storage device is allocated.

- c For VMFS6, specify the block size and define space reclamation parameters. See [Space Reclamation Requests from VMFS Datastores](#).

- 8 In the Ready to Complete page, review the datastore configuration information and click **Finish**.

Results

The datastore on the SCSI-based storage device is created. It is available to all hosts that have access to the device.

Create an NFS Datastore

You can use the **New Datastore** wizard to mount an NFS volume.

Prerequisites

- Set up NFS storage environment.

- If you plan to use Kerberos authentication with the NFS 4.1 datastore, make sure to configure the ESXi hosts for Kerberos authentication.

Procedure

- 1 In the vSphere Client object navigator, browse to a host, a cluster, or a data center.
- 2 From the right-click menu, select **Storage > New Datastore**.
- 3 Select NFS as the datastore type and specify an NFS version.
 - NFS 3
 - NFS 4.1

Important If multiple hosts access the same datastore, you must use the same protocol on all hosts.

- 4 Enter the datastore parameters.

| Option | Description |
|----------------|---|
| Datastore name | The system enforces a 42 character limit for the datastore name. |
| Folder | The mount point folder name |
| Server | The server name or IP address. You can use IPv6 or IPv4 formats. With NFS 4.1, you can add multiple IP addresses or server names if the NFS server supports trunking. The ESXi host uses these values to achieve multipathing to the NFS server mount point. |

- 5 Select **Mount NFS read only** if the volume is exported as read-only by the NFS server.
- 6 To use Kerberos security with NFS 4.1, enable Kerberos and select an appropriate Kerberos model.

| Option | Description |
|--|---|
| Use Kerberos for authentication only (krb5) | Supports identity verification |
| Use Kerberos for authentication and data integrity (krb5i) | In addition to identity verification, provides data integrity services. These services help to protect the NFS traffic from tampering by checking data packets for any potential modifications. |

If you do not enable Kerberos, the datastore uses the default AUTH_SYS security.

- 7 If you are creating a datastore at the data center or cluster level, select hosts that mount the datastore.
- 8 Review the configuration options and click **Finish**.

Create a Virtual Volumes Datastore

You use the **New Datastore** wizard to create a Virtual Volumes datastore.

Procedure

- 1 In the vSphere Client object navigator, browse to a host, a cluster, or a data center.
- 2 From the right-click menu, select **Storage > New Datastore**.
- 3 Select **VVol** as the datastore type.
- 4 Enter the datastore name and select a backing storage container from the list of storage containers.

Make sure to use the name that does not duplicate another datastore name in your data center environment.

If you mount the same Virtual Volumes datastore to several hosts, the name of the datastore must be consistent across all hosts.

- 5 Select the hosts that require access to the datastore.
- 6 Review the configuration options and click **Finish**.

What to do next

After you create the Virtual Volumes datastore, you can perform such datastore operations as renaming the datastore, browsing datastore files, unmounting the datastore, and so on.

You cannot add the Virtual Volumes datastore to a datastore cluster.

Managing Duplicate VMFS Datastores

When a storage device contains a VMFS datastore copy, you can mount the datastore with the existing signature or assign a new signature.

Each VMFS datastore created on a storage device has a unique signature, also called UUID, that is stored in the file system superblock. When the storage device is replicated or its snapshot is taken on the array side, the resulting device copy is identical, byte-for-byte, with the original device. For example, if the original storage device contains a VMFS datastore with UUIDX, the copy appears to contain a datastore copy with the same UUIDX.

In addition to LUN snapshots and replications, certain device operations, such as LUN ID changes, might produce a copy of the original datastore.

ESXi can detect the VMFS datastore copy. You can mount the datastore copy with its original UUID or change the UUID. The process of changing the UUID is called the datastore resignaturing.

Whether you select resignaturing or mounting without resignaturing depends on how the LUNs are masked in the storage environment. If your hosts can see both copies of the LUN, then resignaturing is the optimal method.

Keeping Existing Datastore Signature

If you do not need to resignature a VMFS datastore copy, you can mount it without changing its signature.

You can keep the signature if, for example, you maintain synchronized copies of virtual machines at a secondary site as part of a disaster recovery plan. In the event of a disaster at the primary site, you mount the datastore copy and power on the virtual machines at the secondary site.

Resignaturing a VMFS Datastore Copy

Use datastore resignaturing if you want to retain the data stored on the VMFS datastore copy.

When resignaturing a VMFS copy, ESXi assigns a new signature (UUID) to the copy, and mounts the copy as a datastore distinct from the original. All references to the original signature in virtual machine configuration files are updated.

When you perform datastore resignaturing, consider the following points:

- Datastore resignaturing is irreversible.
- After resignaturing, the storage device replica that contained the VMFS copy is no longer treated as a replica.
- A spanned datastore can be resignatured only if all its extents are online.
- The resignaturing process is fault tolerant. If the process is interrupted, you can resume it later.
- You can mount the new VMFS datastore without a risk of its UUID conflicting with UUIDs of any other datastore from the hierarchy of device snapshots.

Mount a VMFS Datastore Copy

Use datastore resignaturing if you want to retain the data stored on the VMFS datastore copy. If you do not need to resignature the VMFS datastore copy, you can mount it without changing its signature.

Prerequisites

- Perform a storage rescan on your host to update the view of storage devices presented to the host.
- Unmount the original VMFS datastore that has the same UUID as the copy you plan to mount. You can mount the VMFS datastore copy only if it does not collide with the original VMFS datastore.

Procedure

- 1 In the vSphere Client object navigator, browse to a host, a cluster, or a data center.
- 2 From the right-click menu, select **Storage > New Datastore**.
- 3 Select VMFS as the datastore type.

- 4 Enter the datastore name and if necessary, select the placement location for the datastore.
- 5 From the list of storage devices, select the device that has a specific value displayed in the Snapshot Volume column.

The value present in the Snapshot Volume column indicates that the device is a copy that contains a copy of an existing VMFS datastore.

- 6 Mount the datastore.

| Option | Description |
|-----------------------------|---|
| Mount with resignaturing | Under Mount Options , select Assign a New Signature and click Next . |
| Mount without resignaturing | Under Mount Options , select Keep Existing Signature . |

- 7 Review the datastore configuration information and click **Finish**.

Increase VMFS Datastore Capacity

You can increase the capacity of a VMFS datastore. Additional capacity might be required when you add virtual machines to a datastore, or when the virtual machines running on a datastore require more space.

If a shared datastore has powered on virtual machines and becomes 100% full, you can increase the datastore capacity. You can perform this action only from the host where the powered on virtual machines are registered.

Depending on your storage configuration, you can use one of the following methods to increase the datastore capacity. You do not need to power off virtual machines when using either method of increasing the datastore capacity.

Expand an Existing Datastore

Increase the size of an expandable datastore. The datastore is considered expandable when the backing storage device has free space immediately after the datastore extent.

Add an Extent

Increase the capacity of an existing VMFS datastore by adding new storage devices to the datastore. The datastore can span over multiple storage devices, yet appear as a single volume.

The spanned VMFS datastore can use any or all its extents at any time. It does not need to fill up a particular extent before using the next one.

Note Datastores that support only the hardware assisted locking, also called the atomic test and set (ATS) mechanism, cannot span over non-ATS devices. For more information, see [VMFS Locking Mechanisms](#).

Prerequisites

You can increase the datastore capacity if the host storage meets one of the following conditions:

- The backing device for the existing datastore has enough free space.
- You added new storage devices to the host.

Procedure

- 1 Navigate to the datastore.
- 2 Select **Increase Datastore Capacity** from the datastore right-click menu.
- 3 Select a device from the list of storage devices.

Your selection depends on whether an expandable storage device is available.

| Option | Description |
|--|--|
| To expand an existing datastore extent | Select the device for which the Expandable column reads YES. |
| To add an extent | Select the device for which the Expandable column reads NO. |

- 4 Review the **Partition Layout** to see the available configurations.
- 5 Select a configuration option from the bottom panel.

Depending on the current layout of the disk and on your previous selections, the menu items you see might vary.

| Menu Item | Description |
|--|---|
| Use free space to expand the datastore | Expands an existing extent to a required capacity. |
| Use free space | Deploys an extent in the remaining free space of the disk. This menu item is available only when you are adding an extent. |
| Use all available partitions | Dedicates the entire disk to a single extent. This menu item is available only when you are adding an extent and when the disk you are formatting is not blank. The disk is reformatted, and the datastores and any data that it contains are erased. |

- 6 Set the capacity for the extent.

The minimum extent size is 1.3 GB. By default, the entire free space on the storage device is available.

- 7 Click **Next**.
- 8 Review the proposed layout and the new configuration of your datastore, and click **Finish**.

Administrative Operations for Datastores

After creating datastores, you can perform several administrative operations on the datastores. Certain operations, such as renaming a datastore, are available for all types of datastores. Others apply to specific types of datastores.

- **Change Datastore Name**

Use the vSphere Client to change the name of an existing datastore. You can rename the datastore that has virtual machines running on it without any negative impact.

- **Unmount Datastores**

When you unmount a datastore, it remains intact, but can no longer be seen from the hosts that you specify. The datastore continues to appear on other hosts, where it remains mounted.

- **Mount Datastores**

You can mount a datastore you previously unmounted. You can also mount a datastore on additional hosts, so that it becomes a shared datastore.

- **Remove VMFS Datastores**

You can delete any type of VMFS datastore, including copies that you have mounted without resignaturing. When you delete a datastore, it is destroyed and disappears from all hosts that have access to the datastore.

- **Use Datastore Browser**

Use the datastore file browser to manage contents of your datastores. You can browse folders and files that are stored on the datastore. You can also use the browser to upload files and perform administrative tasks on your folders and files.

- **Turn Off Storage Filters**

When you perform VMFS datastore management operations, vCenter Server uses default storage protection filters. The filters help you to avoid storage corruption by retrieving only the storage devices that can be used for a particular operation. Unsuitable devices are not displayed for selection. You can turn off the filters to view all devices.

Change Datastore Name

Use the vSphere Client to change the name of an existing datastore. You can rename the datastore that has virtual machines running on it without any negative impact.

Note If the host is managed by vCenter Server, you cannot rename the datastore by directly accessing the host from the VMware Host Client. You must rename the datastore from vCenter Server.

Procedure

- 1 Navigate to the datastore.

- 2 Right-click the datastore to rename, and select **Rename**.
- 3 Enter a new datastore name.

The system enforces a 42 character limit for the datastore name.

Results

The new name appears on all hosts that have access to the datastore.

Unmount Datastores

When you unmount a datastore, it remains intact, but can no longer be seen from the hosts that you specify. The datastore continues to appear on other hosts, where it remains mounted.

Do not perform any configuration operations that might result in I/O to the datastore while the unmounting is in progress.

Note Make sure that the datastore is not used by vSphere HA Heartbeating. vSphere HA Heartbeating does not prevent you from unmounting the datastore. However, if the datastore is used for heartbeating, unmounting it might cause the host to fail and restart any active virtual machine.

Prerequisites

When appropriate, before unmounting datastores, make sure that the following prerequisites are met:

- No virtual machines reside on the datastore.
- Storage DRS does not manage the datastore.
- Storage I/O Control is disabled for this datastore.

Procedure

- 1 Navigate to the datastore to unmount.
- 2 Right-click the datastore and select **Unmount Datastore**.
- 3 If the datastore is shared, select the hosts from which to unmount the datastore.
- 4 Confirm that you want to unmount the datastore.

Results

After you unmount a VMFS datastore from all hosts, the datastore is marked as inactive. If you unmount an NFS or a virtual volumes datastore from all hosts, the datastore disappears from the inventory. You can mount the unmounted VMFS datastore. To mount the NFS or virtual volumes datastore that has been removed from the inventory, use the New Datastore wizard.

What to do next

If you unmounted the VMFS datastore as a part of a storage removal procedure, you can now detach the storage device that is backing the datastore. See [Detach Storage Devices](#).

Mount Datastores

You can mount a datastore you previously unmounted. You can also mount a datastore on additional hosts, so that it becomes a shared datastore.

A VMFS datastore that has been unmounted from all hosts remains in inventory, but is marked as inaccessible. You can use this task to mount the VMFS datastore to a specified host or multiple hosts.

If you have unmounted an NFS or a Virtual Volumes datastore from all hosts, the datastore disappears from the inventory. To mount the NFS or Virtual Volumes datastore that has been removed from the inventory, use the New Datastore wizard.

A datastore of any type that is unmounted from some hosts while being mounted on others, is shown as active in the inventory.

Procedure

- 1 Navigate to the datastore.
- 2 Right-click the datastore to mount and select one of the following options:
 - **Mount Datastore**
 - **Mount Datastore on Additional Hosts**Whether you see one or another option depends on the type of datastore you use.
- 3 Select the hosts that should access the datastore and click **OK**.
- 4 To list all hosts that share the datastore, navigate to the datastore, and click the **Hosts** tab.

Remove VMFS Datastores

You can delete any type of VMFS datastore, including copies that you have mounted without resignaturing. When you delete a datastore, it is destroyed and disappears from all hosts that have access to the datastore.

Note The delete operation for the datastore permanently deletes all files associated with virtual machines on the datastore. Although you can delete the datastore without unmounting, it is preferable that you unmount the datastore first.

Prerequisites

- Remove or migrate all virtual machines from the datastore.
- Unmount the datastore from all hosts.
- Disable Storage DRS for the datastore.
- Disable Storage I/O Control for the datastore.
- Make sure that the datastore is not used for vSphere HA heartbeating.

Procedure

- 1 Navigate to the datastore.
- 2 Right-click the datastore to remove, and select **Delete Datastore**.
- 3 Confirm that you want to remove the datastore.

Use Datastore Browser

Use the datastore file browser to manage contents of your datastores. You can browse folders and files that are stored on the datastore. You can also use the browser to upload files and perform administrative tasks on your folders and files.

Procedure

- 1 Open the datastore browser.
 - a Display the datastore in the inventory.
 - b Right-click the datastore and select **Browse Files**.
- 2 Explore the contents of the datastore by navigating to existing folders and files.
- 3 Perform administrative tasks by using the icons and options.

| Icons and Options | Descriptions |
|--|---|
| Upload Files | Upload a file to the datastore. |
| Upload Folder (Available only in the vSphere Client) | Upload a folder to the datastore. |
| Download | Download from the datastore. |
| New Folder | Create a folder on the datastore. |
| Copy to | Copy selected folders or files to a new location, either on the same datastore or on a different datastore. |
| Move to | Move selected folders or files to a new location, either on the same datastore or on a different datastore. |
| Rename to | Rename selected files. |
| Delete | Delete selected folders or files. |
| Inflate | Convert a selected thin virtual disk to thick. This option applies only to thin-provisioned disks. |

Upload Files or Folders to Datastores

Use the datastore file browser to upload files to datastores on the ESXi host. If you use the vSphere Client, you can also upload folders.

In addition to their traditional use as storage for virtual machines files, datastores can serve to store data or files related to virtual machines. For example, you can upload ISO images of operating systems from a local computer to a datastore on the host. You then use these images to install guest operating systems on the new virtual machines.

Note You cannot upload files directly to the Virtual Volumes datastores. You must first create a folder on the Virtual Volumes datastore, and then upload the files into the folder. The created folders in Virtual Volume datastores for block storage have a limited storage capacity space of 4GB. The VVols datastore supports direct uploads of folders.

Prerequisites

Required privilege: **Datastore.Browse Datastore**

Procedure

- 1 Open the datastore browser.
 - a Display the datastore in the inventory.
 - b Right-click the datastore and select **Browse Files**.
- 2 (Optional) Create a folder to store the file or folder.
- 3 Upload the file or folder.

| Option | Description |
|--|--|
| Upload a file | <ol style="list-style-type: none"> a Select the target folder and click Upload Files. b Locate the item to upload on the local computer and click Open. |
| Upload a folder (available only in the vSphere Client) | <ol style="list-style-type: none"> a Select the datastore or the target folder and click Upload Folders. b Locate the item to upload on the local computer and click Ok. |

- 4 Refresh the datastore file browser to see the uploaded files or folders on the list.

What to do next

You might experience problems when deploying an OVF template that you previously exported and then uploaded to a datastore. For details and a workaround, see the VMware Knowledge Base article [2117310](#).

Download Files from Datastores

Use the datastore file browser to download files from the datastore available on your ESXi host to your local computer.

Prerequisites

Required privilege: **Datastore.Browse Datastore**

Procedure

- 1 Open the datastore browser.
 - a Display the datastore in the inventory.
 - b Right-click the datastore and select **Browse Files**.
- 2 Navigate to the file to download and click **Download**.
- 3 Follow the prompts to save the file to your local computer.

Move or Copy Datastore Folders or Files

Use the datastore browser to move or copy folders or files to a new location, either on the same datastore or on a different datastore.

Note Virtual disk files are moved or copied without format conversion. If you move a virtual disk to a datastore that belongs to a host different from the source host, you might need to convert the virtual disk. Otherwise, you might not be able to use the disk.

You cannot copy VM files across vCenter Servers.

Prerequisites

Required privilege: **Datastore.Browse Datastore**

Procedure

- 1 Open the datastore browser.
 - a Display the datastore in the inventory.
 - b Right-click the datastore and select **Browse Files**.
- 2 Browse to an object you want to move or copy, either a folder or a file.
- 3 Select the object and click **Move to** or **Copy to**.
- 4 Specify the destination location.
- 5 (Optional) Select **Overwrite files and folders with matching names at the destination**.
- 6 Click **OK**.

Rename Datastore Files

Use the datastore browser to rename files.

Prerequisites

Required privilege: **Datastore.Browse Datastore**

Procedure

- 1 Open the datastore browser.
 - a Display the datastore in the inventory.
 - b Right-click the datastore and select **Browse Files**.
- 2 Browse to a file you want to rename.
- 3 Select the file and click **Rename to**.
- 4 Specify the new name and click **OK**.

Inflate Thin Virtual Disks


If you created a virtual disk in the thin format, you can change the format to thick.

You use the datastore browser to inflate the thin virtual disk.

Prerequisites

- Make sure that the datastore where the virtual machine resides has enough space.
- Make sure that the virtual disk is thin.
- Remove snapshots.
- Power off your virtual machine.

Procedure

- 1 Navigate to the folder of the virtual disk you want to inflate.
 - a Navigate to the virtual machine.
 - b Click the **Datastores** tab.
The datastore that stores the virtual machine files is listed.
 - c Right-click the datastore and select **Browse Files**.
The datastore browser displays contents of the datastore.
- 2 Expand the virtual machine folder and browse to the virtual disk file that you want to convert.
The file has the `.vmdk` extension and is marked with the virtual disk  icon.
- 3 Select the virtual disk file and click **Inflate**.

Note The option might not be available if the virtual disk is thick or when the virtual machine is running.

Results

The inflated virtual disk occupies the entire datastore space originally provisioned to it.

Turn Off Storage Filters

When you perform VMFS datastore management operations, vCenter Server uses default storage protection filters. The filters help you to avoid storage corruption by retrieving only the storage devices that can be used for a particular operation. Unsuitable devices are not displayed for selection. You can turn off the filters to view all devices.

Note This task is available only in the vSphere Web Client.

Prerequisites

Before you change the device filters, consult with the VMware support team. You can turn off the filters only if you have other methods to prevent device corruption.

Procedure

- 1 Browse to the vCenter Server instance.
- 2 Click the **Configure** tab.
- 3 Under **Settings**, click **Advanced Settings**, and click **Edit**.
- 4 Specify the filter to turn off.

In the **Name** and Value text boxes at the bottom of the page, enter appropriate information.

| Name | Value |
|--|-------|
| config.vpxd.filter.vmfsFilter | False |
| config.vpxd.filter.rdmFilter | False |
| config.vpxd.filter.sameHostsAndTransportFilter | False |
| config.vpxd.filter.hostRescanFilter | False |

Note If you turn off this filter, your hosts continue to perform a rescan each time you present a new LUN to a host or a cluster.

- 5 Click **Add**, and click **OK** to save your changes.

You are not required to restart the vCenter Server system.

Storage Filtering

vCenter Server provides storage filters to help you avoid storage device corruption or performance degradation that might be caused by an unsupported use of storage devices. These filters are available by default.

Table 17-6. Storage Filters

| Filter Name | Description |
|--|---|
| config.vpxd.filter.vmfsFilter (VMFS Filter) | Filters out storage devices, or LUNs, that are already used by a VMFS datastore on any host managed by vCenter Server. The LUNs do not show up as candidates to be formatted with another VMFS datastore or to be used as an RDM. |
| config.vpxd.filter.rdmFilter (RDM Filter) | Filters out LUNs that are already referenced by an RDM on any host managed by vCenter Server. The LUNs do not show up as candidates to be formatted with VMFS or to be used by a different RDM. For your virtual machines to access the same LUN, the virtual machines must share the same RDM mapping file. For information about this type of configuration, see the <i>vSphere Resource Management</i> documentation. |
| config.vpxd.filter.sameHostsAndTransportsFilter (Same Hosts and Transports Filter) | Filters out LUNs ineligible for use as VMFS datastore extents because of host or storage type incompatibility. Prevents you from adding the following LUNs as extents: <ul style="list-style-type: none"> ■ LUNs not exposed to all hosts that share the original VMFS datastore. ■ LUNs that use a storage type different from the one the original VMFS datastore uses. For example, you cannot add a Fibre Channel extent to a VMFS datastore on a local storage device. |
| config.vpxd.filter.hostRescanFilter (Host Rescan Filter) | Automatically rescans and updates VMFS datastores after you perform datastore management operations. The filter helps provide a consistent view of all VMFS datastores on all hosts managed by vCenter Server. |
| Note If you present a new LUN to a host or a cluster, the hosts automatically perform a rescan no matter whether you have the Host Rescan Filter on or off. | |

Set Up Dynamic Disk Mirroring

Typically, you cannot use LUN manager software on virtual machines to mirror virtual disks. However, if your Microsoft Windows virtual machines support dynamic disks, you can mirror virtual disks across two SAN LUNs. Mirroring helps you to protect the virtual machines from an unplanned storage device loss.

Prerequisites

- Use a Windows virtual machine that supports dynamic disks.
- Required privilege: **Advanced**

Procedure

- 1 Create a virtual machine with two virtual disks.
Place the disks on different datastores.
- 2 Log in to your virtual machine and configure the disks as dynamic mirrored disks.
See Microsoft documentation.
- 3 After the disks synchronize, power off the virtual machine.

4 Change virtual machine settings to enable the dynamic disk mirroring.

- a Right-click the virtual machine and select **Edit Settings**.
- b Click the **VM Options** tab and expand the **Advanced** menu.
- c Click **Edit Configuration** next to Configuration Parameters.
- d Click **Add Row** and add the following parameters:

| Name | Value |
|--|-------|
| <code>scsi#.returnNoConnectDuringAPD</code> | True |
| <code>scsi#.returnBusyOnNoConnectStatus</code> | False |

- e If you use ESXi version 6.7 or later, include an additional parameter for each virtual disk participating in the software RAID-1 configuration.

The parameter prevents guest OS I/O failures when a storage device fails.

| Name | Value |
|--|-------|
| <code>scsi#:1.passthruTransientErrors</code> | True |
| <code>scsi#:2.passthruTransientErrors</code> | True |

- f Click **OK**.

Collecting Diagnostic Information for ESXi Hosts on a Storage Device

During a host failure, ESXi must be able to save diagnostic information to a preconfigured location for diagnostic and technical support purposes.

Typically, a partition to collect diagnostic information, also called VMkernel core dump, is created on a local storage device during ESXi installation. You can override this default behavior if, for example, you use shared storage devices instead of local storage. To prevent automatic formatting of local devices, detach the devices from the host before you install ESXi and power on the host for the first time. You can later set up a location for collecting diagnostic information on a local or remote storage device.

When you use storage devices, you can select between two options of setting up core dump collection. You can use a preconfigured diagnostic partition on a storage device or use a file on a VMFS datastore.

What to read next

■ [Set Up a Device Partition as Core Dump Location](#)

Create a diagnostic partition for your ESXi host.

■ [Set Up a File as Core Dump Location](#)

If the size of your available core dump partition is insufficient, you can configure ESXi to use a file for diagnostic information.

Set Up a Device Partition as Core Dump Location

Create a diagnostic partition for your ESXi host.

Note This task is available only in the vSphere Web Client.

When you create a diagnostic partition, the following considerations apply:

- You cannot create a diagnostic partition on an iSCSI LUN accessed through the software iSCSI or dependent hardware iSCSI adapter. For more information about diagnostic partitions with iSCSI, see [General Recommendations for Boot from iSCSI SAN](#).
- You cannot create a diagnostic partition on a LUN accessed through software FCoE.
- Unless you are using diskless servers, set up a diagnostic partition on local storage.
- Each host must have a diagnostic partition of 2.5 GB. If multiple hosts share a diagnostic partition on a SAN LUN, the partition must be large enough to accommodate core dumps of all hosts.
- If a host that uses a shared diagnostic partition fails, reboot the host and extract log files immediately after the failure. Otherwise, the second host that fails before you collect the diagnostic data of the first host might save the core dump.

Procedure

- 1 Navigate to the host.
- 2 Right-click the host, and select **Add Diagnostic Partition**.

If you do not see this menu item, the host already has a diagnostic partition.

- 3 Specify the type of diagnostic partition.

| Menu Item | Description |
|----------------------------|--|
| Private local | Creates the diagnostic partition on a local disk. This partition stores fault information only for your host. |
| Private SAN storage | Creates the diagnostic partition on a non-shared SAN LUN. This partition stores fault information only for your host. |
| Shared SAN storage | Creates the diagnostic partition on a shared SAN LUN. Multiple hosts can access this partition. It can store fault information for more than one host. |

- 4 Click **Next**.
- 5 Select the device to use for the diagnostic partition and click **Next**.
- 6 Review the partition configuration information and click **Finish**.

Verify a Diagnostic Partition

Use the `esxcli` command to verify whether a diagnostic partition is set.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ List partitions to verify that a diagnostic partition is set.

```
esxcli system coredump partition list
```

Results

If a diagnostic partition is set, the command displays information about it. Otherwise, the command shows that no partition is activated and configured.

What to do next

To manage the host's diagnostic partition, use the vCLI commands. See *vSphere Command-Line Interface Concepts and Examples*.

Set Up a File as Core Dump Location

If the size of your available core dump partition is insufficient, you can configure ESXi to use a file for diagnostic information.

Typically, a core dump partition of 2.5 GB is created during ESXi installation. For upgrades from ESXi 5.0 and earlier, the core dump partition is limited to 100 MB. For this type of upgrade, during the boot process the system might create a core dump file on a VMFS datastore. If it does not create a core dump file, you can manually create the file.

Note Software iSCSI and software FCoE are not supported for core dump file locations.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 Create a VMFS datastore core dump file by running the following command:

```
esxcli system coredump file add
```

The command takes the following options, but they are not required and can be omitted:

| Option | Description |
|--|--|
| <code>--datastore -d datastore_UUID or datastore_name</code> | If not provided, the system selects a datastore of sufficient size. |
| <code>--file -f file_name</code> | If not provided, the system specifies a unique name for the core dump file. |
| <code>--size -s file_size_MB</code> | If not provided, the system creates a file of the size appropriate for the memory installed in the host. |

- 2 Verify that the file has been created:

esxcli system coredump file list

You can see the output similar to the following:

| Path | Active | Configured | Size |
|--|--------|------------|-----------|
| ----- | ----- | ----- | ----- |
| /vmfs/volumes/52b021c3-.../vmkdump/test.dumpfile | false | false | 104857600 |

- 3 Activate the core dump file for the host:

esxcli system coredump file set

The command takes the following options:

| Option | Description |
|---------------------------|--|
| <code>--path -p</code> | The path of the core dump file to use. The file must be pre-allocated. |
| <code>--smart -s</code> | This flag can be used only with <code>--enable -e=true</code> . It causes the file to be selected using the smart selection algorithm. For example, esxcli system coredump file set --smart --enable true |

- 4 Verify that the core dump file is active and configured:

esxcli system coredump file list

The output similar to the following indicates that the core dump file is active and configured:

| Path | Active | Configured | Size |
|--|--------|------------|-----------|
| ----- | ----- | ----- | ----- |
| /vmfs/volumes/52b021c3-.../vmkdump/test.dumpfile | True | True | 104857600 |

What to do next

For information about other commands you can use to manage the core dump files, see the *vSphere Command-Line Interface Reference* documentation.

Deactivate and Delete a Core Dump File

Deactivate a configured core dump file and, if needed, remove it from the VMFS datastore.

You can temporarily deactivate the core dump file. If you do not plan to use the deactivated file, you can remove it from the VMFS datastore. To remove the file that has not been deactivated, you can use the `esxcli system coredump file remove` command with the `--force | -F` parameter.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 List the core dump files:

```
esxcli system coredump file list
```

- 2 Deactivate the core dump file by running the following command:

```
esxcli system coredump file set --unconfigure | -u
```

- 3 Remove the file from the VMFS datastore:

```
esxcli system coredump file remove --file | -f file_name
```

The command takes the following options:

| Option | Description |
|---------------------------|--|
| <code>--file -f</code> | Enter the name of the dump file to be removed. If you do not enter the name, the command removes the default configured core dump file. |
| <code>--force -F</code> | Deactivate and unconfigure the dump file being removed. This option is required if the file has not been previously deactivated and is active. |

Results

The core dump file becomes disabled and is removed from the VMFS datastore.

Checking Metadata Consistency with VOMA

Use vSphere On-disk Metadata Analyzer (VOMA) to identify and fix incidents of metadata corruption that affect file systems or underlying logical volumes.

Problem

You can check metadata consistency when you experience problems with a VMFS datastore or a virtual flash resource. For example, perform a metadata check if one of the following occurs:

- You experience storage outages.
- After you rebuild RAID or perform a disk replacement.

- You see metadata errors in the `vmkernel.log` file similar to the following:

```
cpu11:268057)WARNING: HBX: 599: Volume 50fd60a3-3aae1ae2-3347-0017a4770402
("<Datastore_name>") may be damaged on disk. Corrupt heartbeat detected at offset 3305472:
[HB state 0 offset 6052837899185946624 gen 15439450 stampUS 5 $
```

- You are unable to access files on a VMFS.
- You see corruption being reported for a datastore in events tabs of vCenter Server.

Solution

To check metadata consistency, run VOMA from the CLI of an ESXi host. VOMA can be used to check and fix minor inconsistency issues for a VMFS datastore or logical volumes that back the VMFS datastore.

VOMA can check and fix the following items.

Table 17-7. VOMA Functions

| VOMA Functions | Description |
|---------------------------------|--|
| Metadata check and fix | <p>Examples of metadata check and fix include, but are not limited to, the following:</p> <ul style="list-style-type: none"> ■ Validation of VMFS volume header for basic metadata consistency. ■ Checking consistency of VMFS resource files (system file). ■ Checking the pathname and connectivity of all files. |
| Affinity metadata check and fix | <p>To enable the affinity check for VMFS6, use the <code>-a --affinityChk</code> option.</p> <p>Several examples of affinity metadata check and fix include the following:</p> <ul style="list-style-type: none"> ■ Affinity flags in resource types and <code>FS3_ResFileMetadata</code>. ■ Validation of affinity flags in SFB RC meta (<code>FS3_ResourceClusterMDVMFS6</code>). ■ Validation of all entries in the <code>affinityInfo</code> entries in <code>rcMeta</code> of RC, including the overflow key, to make sure no invalid entries exist. Checking for missing entries. |
| Directory validation | <p>VOMA can detect and correct the following errors:</p> <ul style="list-style-type: none"> ■ Directory hash block corruption. ■ Alloc map corruption. ■ Link blocks corruptions. ■ Directory entry block corruptions. <p>Based on the nature of the corruption, VOMA can either fix only the corrupted entries or fully reconstruct the hash block, alloc map blocks and link blocks.</p> |
| Lost and found files | <p>During a filesystem check, VOMA can find files that are not referenced anywhere in the filesystem. These orphaned files are valid and complete, but do not have a name or directory entry on the system.</p> <p>If VOMA encounters orphaned files during scanning, it creates a directory named <code>lost+found</code> at the root of the volume to store the orphaned files. The names of the files use the <code>Filesequence-number</code> format.</p> |

Command options that the VOMA tool takes include the following.

Table 17-8. VOMA Command Options

| Command Option | Description | | | | | | | | |
|-------------------------------|---|--------------------|--|--------------------|--|-------------------|--|-------------------|------------------------|
| <code>-m --module</code> | <p>The modules to run include the following:</p> <table> <tr> <td><code>vmfs</code></td><td>If you do not specify the name of the module, this option is used by default. You can check the VMFS file systems and the file systems that back virtual flash resources. If you specify this module, minimal checks are performed for LVM as well.</td></tr> <tr> <td><code>lvm</code></td><td>Check logical volumes that back the VMFS datastores.</td></tr> <tr> <td><code>ptbl</code></td><td>Check and validate VMFS partitions, such as MBR or GPT. If no partition exists, determine whether partitions should exist.</td></tr> </table> | <code>vmfs</code> | If you do not specify the name of the module, this option is used by default. You can check the VMFS file systems and the file systems that back virtual flash resources. If you specify this module, minimal checks are performed for LVM as well. | <code>lvm</code> | Check logical volumes that back the VMFS datastores. | <code>ptbl</code> | Check and validate VMFS partitions, such as MBR or GPT. If no partition exists, determine whether partitions should exist. | | |
| <code>vmfs</code> | If you do not specify the name of the module, this option is used by default. You can check the VMFS file systems and the file systems that back virtual flash resources. If you specify this module, minimal checks are performed for LVM as well. | | | | | | | | |
| <code>lvm</code> | Check logical volumes that back the VMFS datastores. | | | | | | | | |
| <code>ptbl</code> | Check and validate VMFS partitions, such as MBR or GPT. If no partition exists, determine whether partitions should exist. | | | | | | | | |
| <code>-f --func</code> | <p>Functions to be performed include the following:</p> <table> <tr> <td><code>query</code></td><td>List functions supported by module.</td></tr> <tr> <td><code>check</code></td><td>Check for errors.</td></tr> <tr> <td><code>fix</code></td><td>Check and fix errors.</td></tr> <tr> <td><code>dump</code></td><td>Collect metadata dump.</td></tr> </table> | <code>query</code> | List functions supported by module. | <code>check</code> | Check for errors. | <code>fix</code> | Check and fix errors. | <code>dump</code> | Collect metadata dump. |
| <code>query</code> | List functions supported by module. | | | | | | | | |
| <code>check</code> | Check for errors. | | | | | | | | |
| <code>fix</code> | Check and fix errors. | | | | | | | | |
| <code>dump</code> | Collect metadata dump. | | | | | | | | |
| <code>-a --affinityChk</code> | Include affinity related check and fix for VMFS6. | | | | | | | | |
| <code>-d --device</code> | Device or disk to be inspected. Make sure to provide the absolute path to the device partition backing the VMFS datastore. For example, <code>/vmfs/devices/disks/naa.00000000000000000000000000000000:1</code> . | | | | | | | | |
| <code>-s --logfile</code> | Specify the log file to output the results. | | | | | | | | |
| <code>-x --extractDump</code> | Extract the collected dump using VOMA. | | | | | | | | |
| <code>-D --dumpfile</code> | Dump file to save the collected metadata dump. | | | | | | | | |
| <code>-v --version</code> | Display the version of VOMA. | | | | | | | | |
| <code>-h --help</code> | Display the help message for the VOMA command. | | | | | | | | |

Example

```
voma -m vmfs -f check -d /vmfs/devices/disks/naa.xxxx:x
```

```
voma -m vmfs -f dump -d /vmfs/devices/disks/naa.xxxx:x -D dumpfilename
```


Use VOMA to Check Metadata Consistency

The task demonstrates how to use VOMA to check VMFS metadata consistency. VOMA can be used to check and fix minor inconsistency issues for a VMFS datastore or a virtual flash resource. Run VOMA from the CLI of an ESXi host.

Prerequisites

- Make sure that the VMFS datastore you analyze does not span multiple extents. You can run VOMA only against a single-extent datastore.
- Power off any virtual machines that are running or migrate them to a different datastore.

Procedure

- 1 Obtain the name and partition number of the device that backs the VMFS datastore that you want to check.

```
#esxcli storage vmfs extent list
```

The Device Name and Partition columns in the output identify the device. For example:

| Volume Name | Device Name | Partition |
|-------------|-------------|-----------|
| 1TB_VMFS6 | naa.xxxx | 3 |

- 2 Check for VMFS errors.

Provide the absolute path to the device partition that backs the VMFS datastore, and provide a partition number with the device name. For example:

```
# voma -m vmfs -f check -d /vmfs/devices/disks/naa.xxxx:3
```

The output lists possible errors. For example, the following output indicates that the heartbeat address is invalid.

```
XXXXXXXXXXXXXXXXXXXXXXXXX
Phase 2: Checking VMFS heartbeat region
ON-DISK ERROR: Invalid HB address
Phase 3: Checking all file descriptors.
Phase 4: Checking pathname and connectivity.
Phase 5: Checking resource reference counts.

Total Errors Found:          1
```

Configuring VMFS Pointer Block Cache

Pointer blocks, also called indirection blocks, are file system resources that contain addresses to VMFS file blocks. When you open a vmdk file on an ESXi host, pointer blocks related to that file are stored in the pointer block cache. The size of the pointer block cache is a configurable parameter.

The pointer block cache is a host-wide cache that is independent from VMFS. The cache is shared across all datastores that are accessed from the same ESXi host.

The size of the pointer block cache is controlled by `/VMFS3/MinAddressableSpaceTB` and `/VMFS3/MaxAddressableSpaceTB`. You can configure the minimum and maximum sizes on each ESXi host.

`/VMFS3/MinAddressableSpaceTB`

The minimum value is minimum amount of memory that the system guarantees to the pointer block cache. For example, 1 TB of open file space requires approximately 4 MB of memory. Default value is 10 TB.

`/VMFS3/MaxAddressableSpaceTB`

The parameter defines the maximum limit of pointer blocks that can be cached in memory. Default value is 32 TB. Maximum value is 128 TB. Typically, the default value of the `/VMFS3/MaxAddressableSpaceTB` parameter is adequate.

However, as the size of the open vmdk files increases, the number of pointer blocks related to those files also increases. If the increase causes any performance degradation, you can adjust the parameter to its maximum value to provide more space for the pointer block cache. Base the maximum size of the pointer block cache on the working set, or the active pointer blocks required.

Pointer Block Eviction

The `/VMFS3/MaxAddressableSpaceTB` parameter also controls the growth of the pointer block cache. When the size of the pointer block cache approaches the configured maximum size, a pointer block eviction process starts. The mechanism leaves active pointer blocks, but removes non-active or less active blocks from the cache, so that space can be reused.

To change the values for the pointer block cache, use the **Advanced System Settings** dialog box of the vSphere Client or the `esxcli system settings advanced set -o` command.

You can use the `esxcli storage vmfs pbcache` command to obtain information about the size of the pointer block cache and other statistics. This information assists you in adjusting minimum and maximum sizes of the pointer block cache, so that you can get maximum performance.

Obtain Information for VMFS Pointer Block Cache

You can get information about VMFS pointer block cache use. This information helps you understand how much space the pointer block cache consumes. You can also determine whether you must adjust the minimum and maximum sizes of the pointer block cache.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ To obtain or reset the pointer block cache statistics, use the following command:

```
esxcli storage vmfs pbcache
```

| Option | Description |
|--------------|--|
| get | Get VMFS pointer block cache statistics. |
| reset | Reset the VMFS pointer block cache statistics. |

Example: Getting Statistics for Pointer Block Cache

```
#esxcli storage vmfs pbcache get
Cache Capacity Miss Ratio: 0 %
Cache Size: 0 MiB
Cache Size Max: 132 MiB
Cache Usage: 0 %
Cache Working Set: 0 TiB
Cache Working Set Max: 32 TiB
Vmfs Heap Overhead: 0 KiB
Vmfs Heap Size: 23 MiB
Vmfs Heap Size Max: 256 MiB
```

Change the Size of the Pointer Block Cache

You can adjust the minimum and maximum sizes of the pointer block cache.

Caution Changing advanced options is considered unsupported. Typically, the default settings produce the optimum result. Change the advanced options only when you get specific instructions from VMware technical support or a knowledge base article.

Procedure

- 1 Browse to the host.
- 2 Click the **Configure** tab.
- 3 Under **System**, click **Advanced System Settings**.
- 4 In Advanced System Settings, select the appropriate item.

| Option | Description |
|------------------------------------|---|
| VMFS3.MinAddressableSpaceTB | Minimum size of all open files that VMFS cache guarantees to support. |
| VMFS3.MaxAddressableSpaceTB | Maximum size of all open files that VMFS cache supports before eviction starts. |

- 5 Click the **Edit** button and change the value.
- 6 Click **OK**.

Example: Use the esxcli Command to Change the Pointer Block Cache

You can also use the `esxcli system settings advanced set -o` to modify the size of the pointer block cache. The following example describes how to set the size to its maximum value of 128 TB.

- 1 To change the value of `/VMFS3/MaxAddressableSpaceTB` to 128 TB, enter the following command:

```
# esxcli system settings advanced set -i 128 -o /VMFS3/
MaxAddressableSpaceTB
```

- 2 To confirm that the value is set correctly, enter this command:

```
# esxcli system settings advanced list -o /VMFS3/MaxAddressableSpaceTB
```

Understanding Multipathing and Failover

18

To maintain a constant connection between a host and its storage, ESXi supports multipathing. With multipathing, you can use more than one physical path that transfers data between the host and an external storage device.

If a failure of any element in the SAN network, such as an adapter, switch, or cable, occurs, ESXi can switch to another viable physical path. This process of path switching to avoid failed components is known as path failover.

In addition to path failover, multipathing provides load balancing. Load balancing is the process of distributing I/O loads across multiple physical paths. Load balancing reduces or removes potential bottlenecks.

Note Virtual machine I/O might be delayed for up to 60 seconds while path failover takes place. With these delays, the SAN can stabilize its configuration after topology changes. In general, the I/O delays might be longer on active-passive arrays and shorter on active-active arrays.

Read the following topics next:

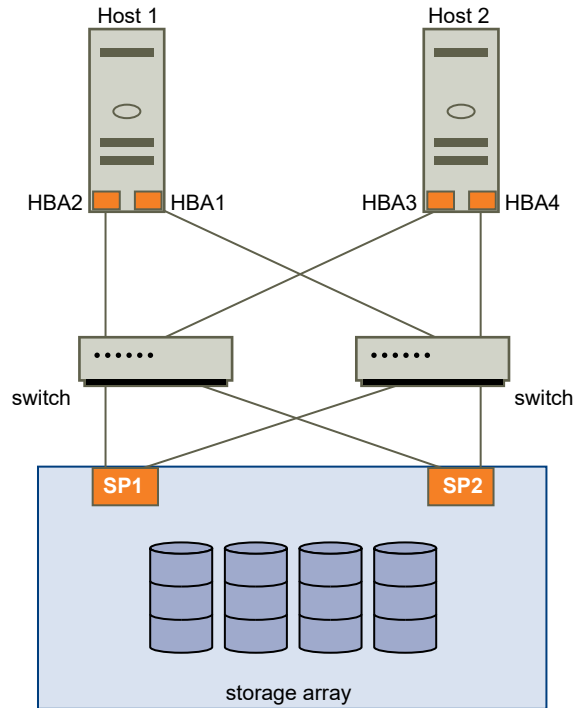
- [Failovers with Fibre Channel](#)
- [Host-Based Failover with iSCSI](#)
- [Array-Based Failover with iSCSI](#)
- [Path Failover and Virtual Machines](#)
- [Pluggable Storage Architecture and Path Management](#)
- [Viewing and Managing Paths](#)
- [Using Claim Rules](#)
- [Scheduling Queues for Virtual Machine I/Os](#)

Failovers with Fibre Channel

To support multipathing, your host typically has two or more HBAs available. This configuration supplements the SAN multipathing configuration. Generally, the SAN multipathing provides one or more switches in the SAN fabric and one or more storage processors on the storage array device itself.

In the following illustration, multiple physical paths connect each server with the storage device. For example, if HBA1 or the link between HBA1 and the FC switch fails, HBA2 takes over and provides the connection. The process of one HBA taking over for another is called HBA failover.

Figure 18-1. Multipathing and Failover with Fibre Channel



Similarly, if SP1 fails or the links between SP1 and the switches breaks, SP2 takes over. SP2 provides the connection between the switch and the storage device. This process is called SP failover. VMware ESXi supports both HBA and SP failovers.

Host-Based Failover with iSCSI

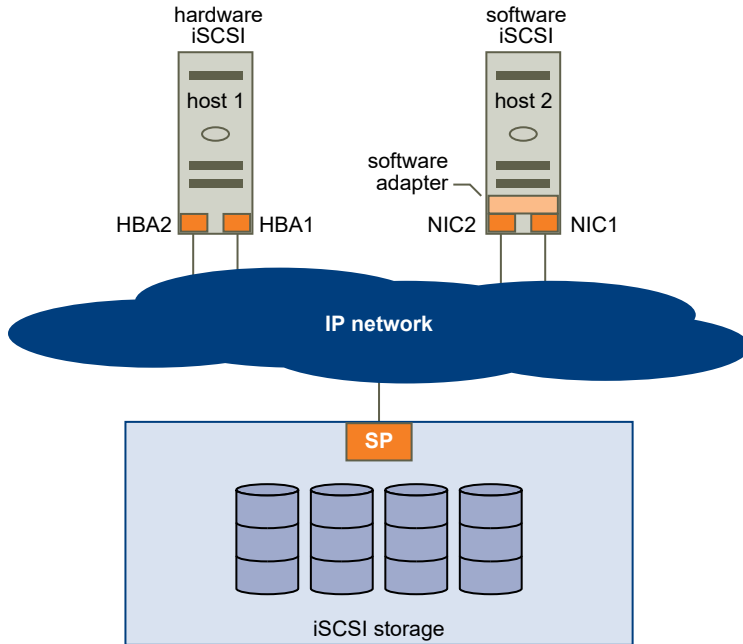
When setting up your ESXi host for multipathing and failover, you can use multiple iSCSI HBAs or combine multiple NICs with the software iSCSI adapter.

For information on different types of iSCSI adapters, see [iSCSI Initiators](#).

When you use multipathing, specific considerations apply.

- ESXi does not support multipathing when you combine an independent hardware adapter with software iSCSI or dependent iSCSI adapters in the same host.
- Multipathing between software and dependent adapters within the same host is supported.
- On different hosts, you can mix both dependent and independent adapters.

The following illustration shows multipathing setups possible with different types of iSCSI initiators.

Figure 18-2. Host-Based Path Failover

Hardware iSCSI and Failover

With hardware iSCSI, the host typically has two or more hardware iSCSI adapters. The host uses the adapters to reach the storage system through one or more switches. Alternatively, the setup might include one adapter and two storage processors, so that the adapter can use different paths to reach the storage system.

On the illustration, Host1 has two hardware iSCSI adapters, HBA1 and HBA2, that provide two physical paths to the storage system. Multipathing plug-ins on your host, whether the VMkernel NMP or any third-party MPPs, have access to the paths by default. The plug-ins can monitor health of each physical path. If, for example, HBA1 or the link between HBA1 and the network fails, the multipathing plug-ins can switch the path over to HBA2.

Software iSCSI and Failover

With software iSCSI, as shown on Host 2 of the illustration, you can use multiple NICs that provide failover and load balancing capabilities for iSCSI connections.

Multipathing plug-ins do not have direct access to physical NICs on your host. As a result, for this setup, you first must connect each physical NIC to a separate VMkernel port. You then associate all VMkernel ports with the software iSCSI initiator using a port binding technique. Each VMkernel port connected to a separate NIC becomes a different path that the iSCSI storage stack and its storage-aware multipathing plug-ins can use.

For information about configuring multipathing for software iSCSI, see [Setting Up Network for iSCSI and iSER](#).

Array-Based Failover with iSCSI

Some iSCSI storage systems manage path use of their ports automatically and transparently to ESXi.

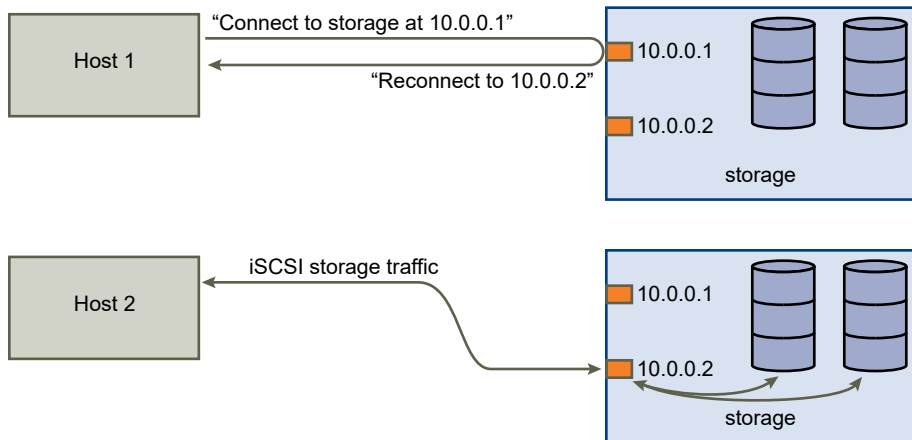
When using one of these storage systems, your host does not see multiple ports on the storage and cannot choose the storage port it connects to. These systems have a single virtual port address that your host uses to initially communicate. During this initial communication, the storage system can redirect the host to communicate with another port on the storage system. The iSCSI initiators in the host obey this reconnection request and connect with a different port on the system. The storage system uses this technique to spread the load across available ports.

If the ESXi host loses connection to one of these ports, it automatically attempts to reconnect with the virtual port of the storage system, and should be redirected to an active, usable port. This reconnection and redirection happens quickly and generally does not disrupt running virtual machines. These storage systems can also request that iSCSI initiators reconnect to the system, to change which storage port they are connected to. This allows the most effective use of the multiple ports.

The Port Redirection illustration shows an example of port redirection. The host attempts to connect to the 10.0.0.1 virtual port. The storage system redirects this request to 10.0.0.2. The host connects with 10.0.0.2 and uses this port for I/O communication.

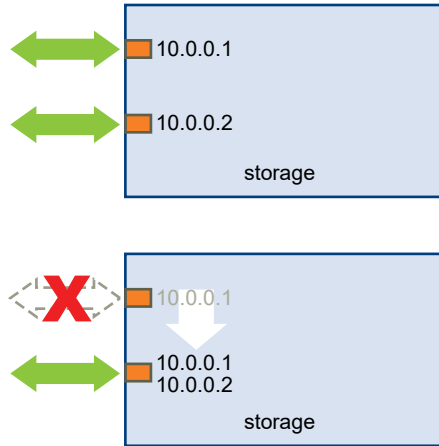
Note The storage system does not always redirect connections. The port at 10.0.0.1 could be used for traffic, also.

Figure 18-3. Port Redirection



If the port on the storage system that is acting as the virtual port becomes unavailable, the storage system reassigns the address of the virtual port to another port on the system. Port Reassignment shows an example of this type of port reassignment. In this case, the virtual port 10.0.0.1 becomes unavailable and the storage system reassigns the virtual port IP address to a different port. The second port responds to both addresses.

Figure 18-4. Port Reassignment



With this form of array-based failover, you can have multiple paths to the storage only if you use multiple ports on the ESXi host. These paths are active-active. For additional information, see [iSCSI Session Management](#).

Path Failover and Virtual Machines

A path failover occurs when the active path to a LUN is changed from one path to another. Typically, the path failover occurs as a result of a SAN component failure along the current path.

When a path fails, storage I/O might pause for 30-60 seconds until your host determines that the link is unavailable and performs the failover. If you attempt to display the host, its storage devices, or its adapters, the operation might appear to stall. Virtual machines with their disks installed on the SAN can appear unresponsive. After the failover, I/O resumes normally and the virtual machines continue to run.

A Windows virtual machine might interrupt the I/O and eventually fail when failovers take too long. To avoid the failure, set the disk timeout value for the Windows virtual machine to at least 60 seconds.

Set Timeout on Windows Guest OS

To avoid disruptions during a path failover, increase the standard disk timeout value on a Windows guest operating system.

This procedure explains how to change the timeout value by using the Windows registry.

Prerequisites

Back up the Windows registry.

Procedure

- 1 Select **Start > Run**.
- 2 Type **regedit.exe**, and click **OK**.

3 In the left-panel hierarchy view, double-click **HKEY_LOCAL_MACHINE > System > CurrentControlSet > Services > Disk**.

4 Double-click **TimeOutValue**.

5 Set the value data to 0x3c (hexadecimal) or 60 (decimal) and click **OK**.

After you make this change, Windows waits at least 60 seconds for delayed disk operations to finish before it generates errors.

6 Reboot guest OS for the change to take effect.

Pluggable Storage Architecture and Path Management

This topic introduces the key concepts behind the ESXi storage multipathing.

Pluggable Storage Architecture (PSA)

To manage multipathing, ESXi uses a special VMkernel layer, Pluggable Storage Architecture (PSA). The PSA is an open and modular framework that coordinates various software modules responsible for multipathing operations. These modules include generic multipathing modules that VMware provides, NMP and HPP, and third-party MPPs.

Native Multipathing Plug-in (NMP)

The NMP is the VMkernel multipathing module that ESXi provides by default. The NMP associates physical paths with a specific storage device and provides a default path selection algorithm based on the array type. The NMP is extensible and manages additional submodules, called Path Selection Policies (PSPs) and Storage Array Type Policies (SATPs). PSPs and SATPs can be provided by VMware, or by a third party.

Path Selection Plug-ins (PSPs)

The PSPs are submodules of the VMware NMP. PSPs are responsible for selecting a physical path for I/O requests.

Storage Array Type Plug-ins (SATPs)

The SATPs are submodules of the VMware NMP. SATPs are responsible for array-specific operations. The SATP can determine the state of a particular array-specific path, perform a path activation, and detect any path errors.

Multipathing Plug-ins (MPPs)

The PSA offers a collection of VMkernel APIs that third parties can use to create their own multipathing plug-ins (MPPs). The modules provide specific load balancing and failover functionalities for a particular storage array. The MPPs can be installed on the ESXi host. They can run in addition to the VMware native modules, or as their replacement.

VMware High-Performance Plug-in (HPP)

The HPP replaces the NMP for high-speed devices, such as NVMe PCIe flash. The HPP improves the performance of ultra-fast flash devices that are installed locally on your ESXi host. The plug-in supports only single-pathed devices.

Claim Rules

The PSA uses claim rules to determine whether an MPP or NMP owns the paths to a particular storage device. The NMP has its own set of claim rules. These claim rules match the device with a specific SATP and PSP.

The MPP claim rules are ordered. Lower rule numbers have preference over higher rule numbers. The NMP claim rules are not ordered.

Table 18-1. Multipathing Acronyms

| Acronym | Definition |
|-------------------|---|
| PSA | Pluggable Storage Architecture |
| NMP | Native Multipathing Plug-in. Generic VMware multipathing module. |
| PSP | Path Selection Plug-in. Handles path selection for a given device. |
| SATP | Storage Array Type Plug-in. Handles path failover for a given storage array. |
| MPP (third-party) | Multipathing Plug-in. A multipathing module developed and provided by a third party. |
| HPP | Native High-Performance Plug-in provided by VMware. It is used with ultra-fast local flash devices. |

About Pluggable Storage Architecture

Pluggable Storage Architecture (PSA) is an open and modular framework that coordinates various software modules responsible for multipathing operations.

VMware provides generic native multipathing modules, called VMware NMP and VMware HPP. In addition, the PSA offers a collection of VMkernel APIs that third-party developers can use. The software developers can create their own load balancing and failover modules for a particular storage array. These third-party multipathing modules (MPPs) can be installed on the ESXi host and run in addition to the VMware native modules, or as their replacement.

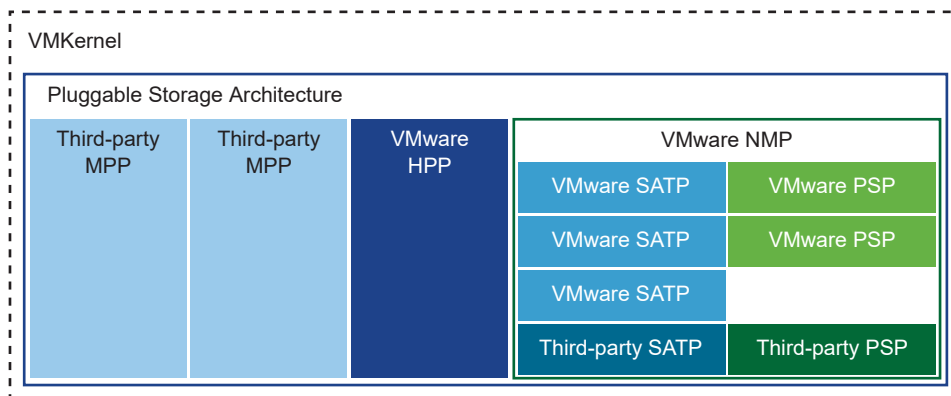
When coordinating the VMware native modules and any installed third-party MPPs, the PSA performs the following tasks:

- Loads and unloads multipathing plug-ins.
- Hides virtual machine specifics from a particular plug-in.
- Routes I/O requests for a specific logical device to the MPP managing that device.
- Handles I/O queueing to the logical devices.

- Implements logical device bandwidth sharing between virtual machines.
- Handles I/O queueing to the physical storage HBAs.
- Handles physical path discovery and removal.
- Provides logical device and physical path I/O statistics.

As the Pluggable Storage Architecture illustration shows, multiple third-party MPPs can run in parallel with the VMware NMP or HPP. When installed, the third-party MPPs can replace the behavior of the native modules. The MPPs can take control of the path failover and the load-balancing operations for the specified storage devices.

Figure 18-5. Pluggable Storage Architecture



VMware Native Multipathing Plug-In

By default, ESXi provides an extensible multipathing module called Native Multipathing Plug-In (NMP).

Generally, the VMware NMP supports all storage arrays listed on the VMware storage HCL and provides a default path selection algorithm based on the array type. The NMP associates a set of physical paths with a specific storage device, or LUN.

For additional multipathing operations, the NMP uses submodules, called SATPs and PSPs. The NMP delegates to the SATP the specific details of handling path failover for the device. The PSP handles path selection for the device.

Typically, the NMP performs the following operations:

- Manages physical path claiming and unclaiming.
- Registers and de-registers logical devices.
- Associates physical paths with logical devices.
- Supports path failure detection and remediation.
- Processes I/O requests to logical devices:
 - Selects an optimal physical path for the request.

- Performs actions necessary to handle path failures and I/O command retries.
- Supports management tasks, such as reset of logical devices.

ESXi automatically installs an appropriate SATP for an array you use. You do not need to obtain or download any SATPs.

VMware NMP Flow of I/O

When a virtual machine issues an I/O request to a storage device managed by the NMP, the following process takes place.

- 1 The NMP calls the PSP assigned to this storage device.
- 2 The PSP selects an appropriate physical path on which to issue the I/O.
- 3 The NMP issues the I/O request on the path selected by the PSP.
- 4 If the I/O operation is successful, the NMP reports its completion.
- 5 If the I/O operation reports an error, the NMP calls the appropriate SATP.
- 6 The SATP interprets the I/O command errors and, when appropriate, activates the inactive paths.
- 7 The PSP is called to select a new path on which to issue the I/O.

Display Multipathing Modules

Use the `esxcli` command to list all multipathing modules loaded into the system. Multipathing modules manage physical paths that connect your host with storage. The modules include VMware native NMP and HPP, and any third-party MPPs.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ To list multipathing modules, run the following command:
`esxcli storage core plugin list --plugin-class=MP`

Results

This command typically shows the NMP and, if loaded, the HPP and the MASK_PATH module. If any third-party MPPs have been loaded, they are listed as well.

```
Plugin name  Plugin class
-----
NMP         MP
```

For more information about the command, see the *vSphere Command-Line Interface Concepts and Examples* and *vSphere Command-Line Interface Reference* documentation.

Display NMP Storage Devices

Use the `esxcli` command to list all storage devices controlled by the VMware NMP and display SATP and PSP information associated with each device.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ To list all storage devices, run the following command:

```
esxcli storage nmp device list
```

Use the `--device | -d=device_ID` parameter to filter the output of this command to show a single device.

Example: Displaying NMP Storage Devices

```
# esxcli storage nmp device list
mpx.vmhba1:C0:T2:L0
  Device Display Name: Local VMware Disk (mpx.vmhba1:C0:T2:L0)
  Storage Array Type: VMW_SATP_LOCAL
  Storage Array Type Device Config: SATP VMW_SATP_LOCAL does not support device
configuration.
  Path Selection Policy: VMW_PSP_FIXED
  Path Selection Policy Device Config: {preferred=vmhba1:C0:T2:L0;current=vmhba1:C0:T2:L0}
  Path Selection Policy Device Custom Config:
  Working Paths: vmhba1:C0:T2:L0
  Is USB: false

.....

eui.6238666462643332
  Device Display Name: SCST_BIO iSCSI Disk (eui.6238666462643332)
  Storage Array Type: VMW_SATP_DEFAULT_AA
  Storage Array Type Device Config: {action_OnRetryErrors=off}
  Path Selection Policy: VMW_PSP_FIXED
  Path Selection Policy Device Config: {preferred=vmhba65:C0:T0:L0;current=vmhba65:C0:T0:L0}
  Path Selection Policy Device Custom Config:
  Working Paths: vmhba65:C0:T0:L0
  Is USB: false
```

For more information about the command, see the *vSphere Command-Line Interface Concepts and Examples* and *vSphere Command-Line Interface Reference* documentation.

Path Selection Plug-Ins and Policies

VMware Path Selection Plug-ins (PSPs) are responsible for selecting a physical path for I/O requests.

The plug-ins are submodules of the VMware NMP. The NMP assigns a default PSP for each logical device based on the device type. You can override the default PSP. For more information, see [Change the Path Selection Policy](#).

Each PSP enables and enforces a corresponding path selection policy.

VMW_PSP_MRU - Most Recently Used (VMware)

The Most Recently Used (VMware) policy is enforced by VMW_PSP_MRU. It selects the first working path discovered at system boot time. When the path becomes unavailable, the host selects an alternative path. The host does not revert to the original path when that path becomes available. The Most Recently Used policy does not use the preferred path setting. This policy is default for most active-passive storage devices.

The VMW_PSP_MRU supports path ranking. To set ranks to individual paths, use the `esxcli storage nmp psp generic pathconfig set` command. For details, see the VMware knowledge base article at <http://kb.vmware.com/kb/2003468> and the *vSphere Command-Line Interface Reference* documentation.

VMW_PSP_FIXED - Fixed (VMware)

This Fixed (VMware) policy is implemented by VMW_PSP_FIXED. The policy uses the designated preferred path. If the preferred path is not assigned, the policy selects the first working path discovered at system boot time. If the preferred path becomes unavailable, the host selects an alternative available path. The host returns to the previously defined preferred path when it becomes available again.

Fixed is the default policy for most active-active storage devices.

VMW_PSP_RR - Round Robin (VMware)

VMW_PSP_RR enables the Round Robin (VMware) policy. Round Robin is the default policy for many arrays. It uses an automatic path selection algorithm rotating through the configured paths.

Both active-active and active-passive arrays use the policy to implement load balancing across paths for different LUNs. With active-passive arrays, the policy uses active paths. With active-active arrays, the policy uses available paths.

The latency mechanism that is activated for the policy by default makes it more adaptive. To achieve better load balancing results, the mechanism dynamically selects an optimal path by considering the following path characteristics:

- I/O bandwidth
- Path latency

To change the default parameters for the adaptive latency Round Robin policy or to disable the latency mechanism, see the [Change Default Parameters for Latency Round Robin](#).

To set other configurable parameters for VMW_PSP_RR, use the `esxcli storage nmp psp roundrobin` command. For details, see the *vSphere Command-Line Interface Reference* documentation.

VMware SATPs

Storage Array Type Plug-ins (SATPs) are responsible for array-specific operations. The SATPs are submodules of the VMware NMP.

ESXi offers an SATP for every type of array that VMware supports. ESXi also provides default SATPs that support non-specific active-active, active-passive, ALUA, and local devices.

Each SATP accommodates special characteristics of a certain class of storage arrays. The SATP can perform the array-specific operations required to detect path state and to activate an inactive path. As a result, the NMP module itself can work with multiple storage arrays without having to be aware of the storage device specifics.

Generally, the NMP determines which SATP to use for a specific storage device and associates the SATP with the physical paths for that storage device. The SATP implements the tasks that include the following:

- Monitors the health of each physical path.
- Reports changes in the state of each physical path.
- Performs array-specific actions necessary for storage fail-over. For example, for active-passive devices, it can activate passive paths.

ESXi includes several generic SATP modules for storage arrays.

VMW_SATP_LOCAL

SATP for local direct-attached devices.

As of vSphere 6.5 Update 2 release, VMW_SATP_LOCAL provides multipathing support for the local devices, except the devices in 4K native format. To claim multiple paths to the local devices, you are no longer required to use other SATPs as you were in the earlier vSphere releases.

VMW_SATP_LOCAL supports the VMW_PSP_MRU and VMW_PSP_FIXED path selection plug-ins, but does not support VMW_PSP_RR.

VMW_SATP_DEFAULT_AA

Generic SATP for active-active arrays.

VMW_SATP_DEFAULT_AP

Generic SATP for active-passive arrays.

VMW_SATP_ALUA

SATP for ALUA-compliant arrays.

For more information, see the *VMware Compatibility Guide* and the *vSphere Command-Line Interface Reference* documentation.

Display SATPs for the Host

Use the `esxcli` command to list VMware NMP SATPs loaded into the system. Display information about the SATPs.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ To list VMware SATPs, run the following command:

```
esxcli storage nmp satp list
```

Results

For each SATP, the output displays information that shows the type of storage array or system the SATP supports. The output also shows the default PSP for any LUNs that use this SATP.

Placeholder (plugin not loaded) in the Description column indicates that the SATP is not loaded.

Example: Displaying SATPs for the Host

```
# esxcli storage nmp satp list
Name                Default PSP      Description
VMW_SATP_MSA        VMW_PSP_MRU     Placeholder (plugin not loaded)
VMW_SATP_ALUA        VMW_PSP_MRU     Placeholder (plugin not loaded)
VMW_SATP_DEFAULT_AP VMW_PSP_MRU     Placeholder (plugin not loaded)
VMW_SATP_SVC         VMW_PSP_FIXED   Placeholder (plugin not loaded)
VMW_SATP_EQL         VMW_PSP_FIXED   Placeholder (plugin not loaded)
VMW_SATP_INV         VMW_PSP_FIXED   Placeholder (plugin not loaded)
VMW_SATP_EVA         VMW_PSP_FIXED   Placeholder (plugin not loaded)
VMW_SATP_ALUA_CX     VMW_PSP_RR      Placeholder (plugin not loaded)
VMW_SATP_SYMM        VMW_PSP_RR      Placeholder (plugin not loaded)
VMW_SATP_CX          VMW_PSP_MRU     Placeholder (plugin not loaded)
VMW_SATP_LSI         VMW_PSP_MRU     Placeholder (plugin not loaded)
VMW_SATP_DEFAULT_AA VMW_PSP_FIXED   Supports non-specific active/active arrays
VMW_SATP_LOCAL       VMW_PSP_FIXED   Supports direct attached devices
```

For more information about the command, see the *vSphere Command-Line Interface Concepts and Examples* and *vSphere Command-Line Interface Reference* documentation.

VMware High Performance Plug-In

VMware provides the High-Performance Plug-in (HPP) to improve the performance of ultra-fast local flash devices on your ESXi host.

The HPP replaces the NMP for high-speed devices, such as NVMe PCIe flash, installed on your host.

The HPP uses a direct I/O submission model, also called fast path, and does not require SATPs or PSPs. The plug-in directly submits I/Os to the local device using one path. Only single-pathed devices are supported.

The HPP is included with vSphere. The direct submission APIs can also be included in the multipathing plug-ins (MPPs) that third parties provide.

Any standalone ESXi hosts that require faster storage performance can benefit from the HPP.

HPP Requirements

The HPP requires the following infrastructure.

- You use vSphere 6.7 or later.
- Your ESXi host uses high-speed local flash devices for storage.

HPP Limitations

HPP does not support the following items that NMP typically supports.

- Multipathing. The HPP claims the first path to a device and rejects the rest of the paths.
- Second-level plug-ins, such as PSP and SATP.
- SCSI-3 persistent reservations or any shared devices.
- 4Kn devices with software emulation. You cannot use the HPP to claim these devices.

vSAN does not support HPP.

HPP Best Practices

To achieve the fastest throughput from a high-speed storage device, follow these recommendations.

- Use the vSphere version that supports the HPP, such as vSphere 6.7 or later.
- Use the HPP for high-speed local flash devices.
- Do not activate the HPP for HDDs, slower flash devices, or remote storage. The HPP is not expected to provide any performance benefits with devices incapable of at least 200 000 IOPS.
- Because ESXi is not expected to provide built-in claim rules for the HPP, enable the HPP using the `esxcli` command.

- Configure your VMs to use VMware Paravirtual controllers. See the *vSphere Virtual Machine Administration* documentation.
- Set the latency sensitive threshold.
- If a single VM drives a significant share of the device's I/O workload, consider spreading the I/O across multiple virtual disks. Attach the disks to separate virtual controllers in the VM.

Otherwise, I/O throughput might be limited due to saturation of the CPU core responsible for processing I/Os on a particular virtual storage controller.

For information about device identifiers for NVMe devices that support only NGUID ID format, see [NVMe Devices with NGUID Device Identifiers](#).

Enable the High-Performance Plug-In

Use claim rules to enable the high-performance plug-in (HPP) on your ESXi host.

You can use the ESXi Shell or vSphere CLI to configure the HPP claim rules. For more information, see *Getting Started with vSphere Command-Line Interfaces* and *vSphere Command-Line Interface Reference*.

Note Enabling the HPP is not supported on PXE booted ESXi hosts.

Prerequisites

- Upgrade your vSphere environment to version 6.7 or later.
- Have high-speed local flash devices installed on your ESXi host.

Procedure

- 1 Create an HPP claim rule by running the following sample command.

```
esxcli storage core claimrule add -r 10 -t vendor -V=NVMe -M=* -P HPP --force-reserved
```

This sample command instructs the HPP to claim all devices with the vendor NVMe. Modify this rule to claim the devices you specify. Make sure to follow these recommendations:

- For the rule ID parameter, use the number within the 1–49 range to make sure that the HPP claim rule precedes the build-in NMP rules. The default NMP rules 50–54 are reserved for locally attached storage devices.
 - Use the `--force-reserved` option. With this option, you can add a rule into the range 0–100 that is reserved for internal VMware use.
- 2 Reboot your host for your changes to take effect.
 - 3 Verify that the HPP claimed the appropriate device.

```
esxcli storage core device list
```

```
mpx.vmhba2:C0:T0:L0
```

```

Display Name: Local NVMe Disk (mpx.vmhba2:C0:T0:L0)
...
Multipath Plugin: HPP
...

```

Set Latency Sensitive Threshold

When you use the HPP for your storage devices, set the latency sensitive threshold for the device, so that I/O can avoid the I/O scheduler.

By default, ESXi passes every I/O through the I/O scheduler. However, using the scheduler might create internal queuing, which is not efficient with the high-speed storage devices.

You can configure the latency sensitive threshold and enable the direct submission mechanism that helps I/O to bypass the scheduler. With this mechanism enabled, the I/O passes directly from PSA through the HPP to the device driver.

For the direct submission to work properly, the observed average I/O latency must be lower than the latency threshold you specify. If the I/O latency exceeds the latency threshold, the system stops the direct submission and temporarily reverts to using the I/O scheduler. The direct submission is resumed when the average I/O latency drops below the latency threshold again.

Procedure

- 1 Set the latency sensitive threshold for the device by running the following command:

```
esxcli storage core device latencythreshold set --device=device name --latency-sensitive-threshold=value in milliseconds
```

- 2 Verify that the latency threshold is set:

```
esxcli storage core device latencythreshold list
```

| Device | Latency Sensitive Threshold |
|----------------------|-----------------------------|
| ----- | ----- |
| naa.55cd2e404c1728aa | 0 milliseconds |
| naa.500056b34036cdfd | 0 milliseconds |
| naa.55cd2e404c172bd6 | 50 milliseconds |

- 3 Monitor the status of the latency sensitive threshold. Check VMkernel logs for the following entries:

- Latency Sensitive Gatekeeper turned on for device *device*. Threshold of *XX* msec is larger than max completion time of *YYY* msec
- Latency Sensitive Gatekeeper turned off for device *device*. Threshold of *XX* msec is exceeded by command completed in *YYY* msec

ESXi ESXCLI HPP Commands

You can use the ESXi Shell or vSphere CLI commands to configure and monitor the high-performance plug-in.

See *Getting Started with vSphere Command-Line Interfaces* for an introduction, and *vSphere Command-Line Interface Reference* details of `esxcli` command use.

| Command | Description | Options |
|---|--|---|
| <code>esxcli storage hpp path list</code> | List the paths currently claimed by the high-performance plug-in. | <code>-d --device=device</code> Display information for a specific device. <code>-p --path=path</code> Limit the output to a specific path. |
| <code>esxcli storage hpp device list</code> | List the devices currently controlled by the high-performance plug-in. | <code>-d --device=device</code> Show a specific device. |
| <code>esxcli storage hpp device set</code> | Configure settings for an HPP device. | <code>-d --device=device</code> The HPP device upon which to operate. Use any of the UIDs that the device reports. Required. <code>-M --mark-device-ssd=bool</code> Specify whether the HPP treats the device as an SSD. |
| <code>esxcli storage hpp device usermarkedssd list</code> | List the devices that were marked as SSD by user. | <code>-d --device=device</code> Limit the output to a specific device. |

Viewing and Managing Paths

When you start your ESXi host or rescan your storage adapter, the host discovers all physical paths to storage devices available to the host. Based on a set of claim rules, the host determines which multipathing module, the NMP, HPP, or an MPP, owns the paths to a particular device.

The module that owns the device becomes responsible for managing the multipathing support for the device. By default, the host performs a periodic path evaluation every five minutes and assigns unclaimed paths to the appropriate module.

For the paths managed by the NMP module, a second set of claim rules is used. These rules assign an SATP and PSP modules to each storage device and determine which Storage Array Type Policy and Path Selection Policy to apply.

Use the vSphere Client to view the Storage Array Type Policy and Path Selection Policy assigned to a specific storage device. You can also check the status of all available paths for this storage device. If needed, you can change the default Path Selection Policy using the client.

To change the default multipathing module or SATP, modify claim rules using the vSphere CLI.

You can find some information about modifying claim rules in [Using Claim Rules](#).

View Storage Device Paths

View which multipathing policies the host uses for a specific storage device and the status of all available paths for this storage device.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Devices**.
- 4 Select the storage device whose paths you want to view.
- 5 Click the **Properties** tab and review the module that owns the device, for example NMP.
Under Multipathing Policies, you can also see the Path Selection Policy and Storage Array Type Policy assigned to the device.
- 6 Click the **Paths** tab to review all paths available for the storage device and the status of each path. The following path status information can appear:

| Status | Description |
|--------------|---|
| Active (I/O) | Working path or multiple paths that currently transfer data. |
| Standby | Paths that are inactive. If the active path fails, they can become operational and start transferring I/O. |
| Disabled | Paths that are disabled by the administrator. |
| Dead | Paths that are no longer available for processing I/O. A physical medium failure or array misconfiguration can cause this status. |

If you are using the **Fixed** path policy, you can see which path is the preferred path. The preferred path is marked with an asterisk (*) in the Preferred column.

View Datastore Paths

Review the paths that connect to storage devices backing your VMFS datastores.

Procedure

- 1 Navigate to the VMFS datastore.
- 2 Click the **Configure** tab.
- 3 Click **Connectivity and Multipathing**.
- 4 Select a host to view multipathing details for its devices.
- 5 Under Multipathing Policies, review the module that owns the device, such as NMP. You can also see the Path Selection Policy and Storage Array Type Policy assigned to the device.

For example, you might see the following:

| | |
|---------------------------|----------------|
| Path Selection Policy | Preferred Path |
| Storage Array Type Policy | VMW_SATP_LOCAL |
| Owner Plugin | NMP |

- 6 Under Paths, review the device paths and the status of each path. The following path status information can appear:

| Status | Description |
|---------------------|---|
| Active (I/O) | Working path or multiple paths that currently transfer data. |
| Standby | Paths that are inactive. If the active path fails, they can become operational and start transferring I/O. |
| Disabled | Paths that are disabled by the administrator. |
| Dead | Paths that are no longer available for processing I/O. A physical medium failure or array misconfiguration can cause this status. |

If you are using the **Fixed** path policy, you can see which path is the preferred path. The preferred path is marked with an asterisk (*) in the Preferred column.

Change the Path Selection Policy

Generally, you do not need to change the default multipathing settings your host uses for a specific storage device. If you want to make any changes, you can use the **Edit Multipathing Policies** dialog box to modify a path selection policy. You can also use this dialog box to change multipathing for SCSI-based protocol endpoints.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Storage Devices** or **Protocol Endpoints**.
- 4 Select the item whose paths you want to change and click the **Properties** tab.
- 5 Under Multipathing Policies, click **Edit Multipathing**.
- 6 Select a path policy.

By default, VMware supports the following path selection policies. If you have a third-party PSP installed on your host, its policy also appears on the list.

- Fixed (VMware)
- Most Recently Used (VMware)
- Round Robin (VMware)

- 7 For the fixed policy, select the preferred path from the list of available paths.
- 8 To save your settings and exit the dialog box, click **OK**.

Change Default Parameters for Latency Round Robin

The latency mechanism is activated for the Round Robin path selection policy by default. The mechanism considers I/O bandwidth and path latency to select an optimal path for I/O. When using the latency mechanism, the Round Robin policy can dynamically select the optimal path and achieve better load balancing results.

You can change the default parameters of the latency mechanism, or disable the mechanism.

Prerequisites

Set the path selection policy to Round Robin. See [Change the Path Selection Policy](#).

Procedure

- 1 Configure the latency mechanism by using the following command.

```
esxcli storage nmp psp roundrobin deviceconfig set --type=latency --device=device ID
```

The command takes the following parameters:

| Parameter | Description |
|---|--|
| -S --num-sampling-cycles=<i>sampling value</i> | When <code>--type</code> is set to <code>latency</code> , this parameter controls how many I/Os to use to calculate the average latency of each path. The default value of this parameter is 16. |
| -T --latency-eval-time=<i>time in ms</i> | When <code>--type</code> is set to <code>latency</code> , this parameter controls the frequency at which the latency of paths is updated. Default is 3 minutes. |

- 2 Verify whether the latency Round Robin and its parameters are configured correctly.

```
esxcli storage nmp psp roundrobin deviceconfig get --device=device ID
```

or

```
esxcli storage nmp device list --device=device ID
```

The following sample output shows the path's configuration:

```
Path Selection Policy: VMW_PSP_RR
Path Selection Policy Device Config:
{policy=latency,latencyEvalTime=180000,samplingCycles=16,curSamplingCycle=16,useANO=0;
CurrentPath=vmhba1:C0:T0:L0: NumIOsPending=0,latency=0}
```

What to do next

To disable the latency mechanism, in the Advanced System Settings for your host, change the `Misc.EnablePSPLatencyPolicy` parameter to 0.

Disable Storage Paths

You can temporarily disable paths for maintenance or other reasons.

You disable a path using the Paths panel. You have several ways to access the Paths panel, from a datastore, a storage device, an adapter, or a VVols Protocol Endpoint view.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click one of the following items:
 - **Storage Adapters**
 - **Storage Devices**
 - **Protocol Endpoints**
- 4 In the right pane, select the item whose paths you want to disable, an adapter, storage device, or Protocol Endpoint, and click the **Paths** tab.
- 5 Select the path to disable and click **Disable**.

The path's status changes to Disabled.

Using Claim Rules

Claim rules determine which multipathing module owns the paths to a particular storage device. They also define the type of multipathing support that the host provides to the device.

The claim rules are listed in the host's `/etc/vmware/esx.conf` file.

The rules fall into these categories:

Core Claim Rules

These claim rules determine which multipathing module, the NMP, HPP, or a third-party MPP, claims the specific device.

SATP Claim Rules

Depending on the device type, these rules assign a particular SATP submodule that provides vendor-specific multipathing management to the device.

You can use the `esxcli` commands to add or change the core and SATP claim rules. Typically, you add the claim rules to load a third-party MPP or to hide a LUN from your host. Changing claim rules might be necessary when default settings for a specific device are not sufficient.

For more information about commands available to manage PSA claim rules, see the *Getting Started with vSphere Command-Line Interfaces*.

For a list of storage arrays and corresponding SATPs and PSPs, see the Storage/SAN section of the *vSphere Compatibility Guide*.

Multipathing Considerations

Specific considerations apply when you manage storage multipathing plug-ins and claim rules.

The following considerations help you with multipathing:

- If no SATP is assigned to the device by the claim rules, the default SATP for iSCSI or FC devices is VMW_SATP_DEFAULT_AA. The default PSP is VMW_PSP_FIXED.
- When the system searches the SATP rules to locate a SATP for a given device, it searches the driver rules first. If there is no match, the vendor/model rules are searched, and finally the transport rules are searched. If no match occurs, NMP selects a default SATP for the device.
- If VMW_SATP_ALUA is assigned to a specific storage device, but the device is not ALUA-aware, no claim rule match occurs for this device. The device is claimed by the default SATP based on the device's transport type.
- The default PSP for all devices claimed by VMW_SATP_ALUA is VMW_PSP_MRU. The VMW_PSP_MRU selects an active/optimized path as reported by the VMW_SATP_ALUA, or an active/unoptimized path if there is no active/optimized path. This path is used until a better path is available (MRU). For example, if the VMW_PSP_MRU is currently using an active/unoptimized path and an active/optimized path becomes available, the VMW_PSP_MRU will switch the current path to the active/optimized one.
- While VMW_PSP_MRU is typically selected for ALUA arrays by default, certain ALUA storage arrays need to use VMW_PSP_FIXED. To check whether your storage array requires VMW_PSP_FIXED, see the *VMware Compatibility Guide* or contact your storage vendor. When using VMW_PSP_FIXED with ALUA arrays, unless you explicitly specify a preferred path, the ESXi host selects the most optimal working path and designates it as the default preferred path. If the host selected path becomes unavailable, the host selects an alternative available path. However, if you explicitly designate the preferred path, it will remain preferred no matter what its status is.
- By default, the PSA claim rule 101 masks Dell array pseudo devices. Do not delete this rule, unless you want to unmask these devices.

List Multipathing Claim Rules for the Host

Use the `esxcli` command to list available multipathing claim rules.

Claim rules indicate whether the NMP, HPP, or a third-party MPP manages a given physical path. Each claim rule identifies a set of paths based on the following parameters:

- Vendor/model strings
- Transportation, such as SATA, IDE, Fibre Channel
- Adapter, target, or LUN location
- Device driver, for example, Mega-RAID

Procedure

- ◆ List the multipathing claim rules by running the **esxcli storage core claimrule list --claimrule-class=MP** command.

If you do not use the `claimrule-class` option, the MP rule class is implied.

Example: Sample Output of the `esxcli storage core claimrule list` Command

| Rule | Class | Rule | Class | Type | Plugin | Matches |
|------|-------|-------|---------|-----------|-----------|--|
| MP | | 10 | runtime | vendor | HPP | vendor=NVMe model=* |
| MP | | 10 | file | vendor | HPP | vendor=NVMe model=* |
| MP | | 50 | runtime | transport | NMP | transport=usb |
| MP | | 51 | runtime | transport | NMP | transport=sata |
| MP | | 52 | runtime | transport | NMP | transport=ide |
| MP | | 53 | runtime | transport | NMP | transport=block |
| MP | | 54 | runtime | transport | NMP | transport=unknown |
| MP | | 101 | runtime | vendor | MASK_PATH | vendor=DELL model=Universal Xport |
| MP | | 101 | file | vendor | MASK_PATH | vendor=DELL model=Universal Xport |
| MP | | 200 | runtime | vendor | MPP_1 | vendor=NewVend model=* |
| MP | | 200 | file | vendor | MPP_1 | vendor=NewVend model=* |
| MP | | 201 | runtime | location | MPP_2 | adapter=vmhba41 channel=* target=* lun=* |
| MP | | 201 | file | location | MPP_2 | adapter=vmhba41 channel=* target=* lun=* |
| MP | | 202 | runtime | driver | MPP_3 | driver=megaraid |
| MP | | 202 | file | driver | MPP_3 | driver=megaraid |
| MP | | 65535 | runtime | vendor | NMP | vendor=* model=* |

This example indicates the following:

- The NMP claims all paths connected to storage devices that use the USB, SATA, IDE, and Block SCSI transportation.
- The rules for HPP, MPP_1, MPP_2, and MPP_3 have been added, so that the modules can claim specified devices. For example, the HPP claims all devices with vendor NVMe. All devices handled by the inbox nvme driver are claimed regardless of the actual vendor. The MPP_1 module claims all paths connected to any model of the NewVend storage array.
- You can use the MASK_PATH module to hide unused devices from your host. By default, the PSA claim rule 101 masks Dell array pseudo devices with a vendor string `DELL` and a model string `Universal Xport`.
- The Rule Class column in the output describes the category of a claim rule. It can be MP (multipathing plug-in), Filter, or VAAI.
- The Class column shows which rules are defined and which are loaded. The `file` parameter in the Class column indicates that the rule is defined. The `runtime` parameter indicates that the rule has been loaded into your system. For a user-defined claim rule to be active, two lines with the same rule number must exist, one line for the rule with the `file` parameter and another line with `runtime`. Several default system-defined claim rules have only one line with the Class of `runtime`. You cannot modify these rules.
- The default rule 65535 assigns all unclaimed paths to the NMP. Do not delete this rule.

Add Multipathing Claim Rules

Use the `esxcli` commands to add a multipathing PSA claim rule to the set of claim rules on the system. For the new claim rule to be active, you first define the rule and then load it into your system.

Examples when you add a PSA claim rule include the following:

- You load a new third-party MPP and must define the paths that this module claims.
- You must enable the native HPP.

Warning You cannot create rules where two different plug-ins claim paths to the same device. Your attempts to create these claim rules fail with a warning in `vmkernel.log`.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 To define a new claim rule, use the following command:

```
esxcli storage core claimrule add
```

The command takes the following options:

| Option | Description |
|--|---|
| <code>-A --adapter=<adapter></code> | Adapter of the paths to use. Valid only if <code>--type</code> is <code>location</code> . |
| <code>-u --autoassign</code> | Adds a claim rule based on its characteristics. The rule number is not required. |
| <code>-C --channel=<channel></code> | Channel of the paths to use. Valid only if <code>--type</code> is <code>location</code> . |
| <code>-c --claimrule-class=<cl></code> | Claim rule class to use in this operation. You can specify <code>MP</code> (default), <code>Filter</code> , or <code>VAAI</code> . To configure hardware acceleration for a new array, add two claim rules, one for the VAAI filter and another for the VAAI plug-in. See Add Hardware Acceleration Claim Rules for detailed instructions. |
| <code>-d --device=<device_uid></code> | UID of the device. Valid only when <code>--type</code> is <code>device</code> . |
| <code>-D --driver=<driver></code> | Driver for the HBA of the paths to use. Valid only if <code>--type</code> is <code>driver</code> . |
| <code>-f --force</code> | Force claim rules to ignore validity checks and install the rule anyway. |
| <code>--force-reserved</code> | Override protection of reserved rule ID ranges. Reserved claim rules are the rules with an ID below 100. You can use them to reassign local devices to specific plug-ins, for example, the NVMe device to HPP. |
| <code>--if-unset=<str></code> | Run this command if this advanced user variable is not set to 1. |
| <code>-i --iqn=<iscsi_name></code> | iSCSI Qualified Name for the target. Valid only when <code>--type</code> is <code>target</code> . |

| Option | Description |
|---|---|
| -L --lun=<lun_id> | LUN of the paths. Valid only if <code>--type</code> is <code>location</code> . LUN ID must not be higher than the value of the advanced configuration option <code>/Disk/MaxLUN</code> . |
| -M --model=<model> | Model of the paths to use. Valid only if <code>--type</code> is <code>vendor</code> . Valid values are values of the Model string from the SCSI inquiry string. Run <code>vicfg-scsidevs <conn_options> -l</code> on each device to see model string values. |
| -P --plugin=<plugin> | PSA plug-in to use. The values are <code>NMP</code> , <code>MASK_PATH</code> , or <code>HPP</code> . Third parties can also provide their own PSA plug-ins. Required. |
| -r --rule=<rule_ID> | Rule ID to use. The rule ID indicates the order in which the claim rule is to be evaluated. User-defined claim rules are evaluated in numeric order starting with 101. You can run <code>esxcli storage core claimrule list</code> to determine which rule IDs are available. |
| -T --target=<target> | Target of the paths to use. Valid only if <code>--type</code> is <code>location</code> . |
| -R --transport=<transport> | Transport of the paths to use. Valid only if <code>--type</code> is <code>transport</code> . The following values are supported. <ul style="list-style-type: none"> ■ <code>block</code> — block storage ■ <code>fc</code> — Fibre Channel ■ <code>iscsivendor</code> — iSCSI ■ <code>iscsi</code> — not currently used ■ <code>ide</code> — IDE storage ■ <code>sas</code> — SAS storage ■ <code>sata</code> — SATA storage ■ <code>usb</code> — USB storage ■ <code>parallel</code> — parallel ■ <code>fcoe</code> — FCoE ■ <code>unknown</code> |
| -t --type=<type> | Type of matching to use for the operation. Valid values are the following. Required. <ul style="list-style-type: none"> ■ <code>vendor</code> ■ <code>location</code> ■ <code>driver</code> ■ <code>transport</code> ■ <code>device</code> ■ <code>target</code> |
| -V --vendor=<vendor> | Vendor of the paths to use. Valid only if <code>--type</code> is <code>vendor</code> . Valid values are values of the vendor string from the SCSI inquiry string. Run <code>vicfg-scsidevs <conn_options> -l</code> on each device to see vendor string values. |
| --wwnn=<wwnn> | World-Wide Node Number for the target. |
| --wwpn=<wwpn> | World-Wide Port Number for the target. |

| Option | Description |
|---|---|
| -a --xcopy-use-array-values | Use the array reported values to construct the XCOPY command to be sent to the storage array. This applies to VAAI claim rules only. |
| -s --xcopy-use-multi-segs | Use multiple segments when issuing an XCOPY request. Valid only if <code>--xcopy-use-array-values</code> is specified. |
| -m --xcopy-max-transfer-size | Maximum data transfer size in MB when you use a transfer size different than array reported. Valid only if <code>--xcopy-use-array-values</code> is specified. |
| -k --xcopy-max-transfer-size-kib | Maximum transfer size in KiB for the XCOPY commands when you use a transfer size different than array reported. Valid only if <code>--xcopy-use-array-values</code> is specified. |

- 2 To load the new claim rule into your system, use the following command:

```
esxcli storage core claimrule load
```

This command loads all newly created multipathing claim rules from the `esx.conf` configuration file into the VMkernel. The command has no options.

- 3 To apply claim rules that are loaded, use the following command:

```
esxcli storage core claimrule run
```

The command takes the following options:

| Option | Description |
|--|--|
| -A --adapter=<adapter> | If <code>--type</code> is <code>location</code> , name of the HBA for the paths to run the claim rules on. To run claim rules on paths from all adapters, omit this option. |
| -C --channel=<channel> | If <code>--type</code> is <code>location</code> , value of the SCSI channel number for the paths to run the claim rules on. To run claim rules on paths with any channel number, omit this option. |
| -c --claimrule-class=<cl> | Claim rule class to use in this operation. |
| -d --device=<device_uid> | UID of the device. |
| -L --lun=<lun_id> | If <code>--type</code> is <code>location</code> , value of the SCSI LUN for the paths to run claim rules on. To run claim rules on paths with any LUN, omit this option. |
| -p --path=<path_uid> | If <code>--type</code> is <code>path</code> , this option indicates the unique path identifier (UID) or the runtime name of a path to run claim rules on. |
| -T --target=<target> | If <code>--type</code> is <code>location</code> , value of the SCSI target number for the paths to run claim rules on. To run claim rules on paths with any target number, omit this option. |

| Option | Description |
|--|--|
| <code>-t --type=<location path all></code> | Type of claim to perform. By default, uses <code>all</code> , which means claim rules run without restriction to specific paths or SCSI addresses. Valid values are <code>location</code> , <code>path</code> , and <code>all</code> . |
| <code>-w --wait</code> | <p>You can use this option only if you also use <code>--type all</code>.</p> <p>If the option is included, the claim waits for paths to settle before running the claim operation. In that case, the system does not start the claiming process until it is likely that all paths on the system have appeared before starting the claim process.</p> <p>After the claiming process has started, the command does not return until device registration has completed.</p> <p>If you add or remove paths during the claiming or the discovery process, this option might not work correctly.</p> |

Example: Defining Multipathing Claim Rules

In the following example, you add and load rule # 500. The rule claims all paths with the `NewMod` model string and the `NewVend` vendor string for the `NMP` plug-in.

```
# esxcli storage core claimrule add -r 500 -t vendor -V NewVend -M NewMod -P NMP
```

```
# esxcli storage core claimrule load
```

After you run the `esxcli storage core claimrule list` command, you can see the new claim rule appearing on the list.

The following output indicates that the claim rule 500 has been loaded into the system and is active.

| Rule | Class | Rule | Class | Type | Plugin | Matches |
|------|-------|------|---------|--------|--------|-----------------------------|
| ... | | ... | ... | ... | ... | ... |
| MP | | 500 | runtime | vendor | NMP | vendor=NewVend model=NewMod |
| MP | | 500 | file | vendor | NMP | vendor=NewVend model=NewMod |

Delete Multipathing Claim Rules

Use the `esxcli` commands to remove a multipathing PSA claim rule from the set of claim rules on the system.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 Delete a claim rule from the set of claim rules.

```
esxcli storage core claimrule remove
```

Note By default, the PSA claim rule 101 masks Dell array pseudo devices. Do not delete this rule, unless you want to unmask these devices.

The command takes the following options:

| Option | Description |
|---|---|
| -c --claimrule-class=<str> | Indicate the claim rule class (MP, Filter, VAAI). |
| -P --plugin=<str> | Indicate the plug-in. |
| -r --rule=<long> | Indicate the rule ID. |

This step removes the claim rule from the File class.

- 2 Remove the claim rule from the system.

```
esxcli storage core claimrule load
```

This step removes the claim rule from the Runtime class.

Mask Paths

You can prevent the host from accessing storage devices or LUNs or from using individual paths to a LUN. Use the `esxcli` commands to mask the paths. When you mask paths, you create claim rules that assign the MASK_PATH plug-in to the specified paths.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 Check what the next available rule ID is.

```
esxcli storage core claimrule list
```

The claim rules that you use to mask paths have rule IDs in the range from 101 through 200. If this command shows that rules 101 and 102 exist, you can specify 103 for the rule to add.

- 2 Assign the MASK_PATH plug-in to a path by creating a new claim rule for the plug-in.

```
esxcli storage core claimrule add -P MASK_PATH
```

- 3 Load the MASK_PATH claim rule into your system.

```
esxcli storage core claimrule load
```


- 4 Verify that the MASK_PATH claim rule was added correctly.

```
esxcli storage core claimrule list
```

- 5 If a claim rule for the masked path exists, remove the rule.

```
esxcli storage core claiming unclaim
```

- 6 Run the path claiming rules.

```
esxcli storage core claimrule run
```

Results

After you assign the MASK_PATH plug-in to a path, the path state becomes irrelevant and is no longer maintained by the host. As a result, commands that display the masked path's information might show the path state as dead.

Example: Masking a LUN

In this example, you mask the LUN 20 on targets T1 and T2 accessed through storage adapters vmhba2 and vmhba3.

```
1 #esxcli storage core claimrule list

2 #esxcli storage core claimrule add -P MASK_PATH -r 109 -t location -A vmhba2 -C 0 -T 1 -L 20
  #esxcli storage core claimrule add -P MASK_PATH -r 110 -t location -A vmhba3 -C 0 -T 1 -L 20
  #esxcli storage core claimrule add -P MASK_PATH -r 111 -t location -A vmhba2 -C 0 -T 2 -L 20
  #esxcli storage core claimrule add -P MASK_PATH -r 112 -t location -A vmhba3 -C 0 -T 2 -L 20

3 #esxcli storage core claimrule load

4 #esxcli storage core claimrule list

5 #esxcli storage core claiming unclaim -t location -A vmhba2
  #esxcli storage core claiming unclaim -t location -A vmhba3

6 #esxcli storage core claimrule run
```

Unmask Paths

When you need the host to access the masked storage device, unmask the paths to the device.

Note When you run an unclaim operation using a device property, for example, device ID or vendor, the paths claimed by the MASK_PATH plug-in are not unclaimed. The MASK_PATH plug-in does not track any device property of the paths that it claims.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 Delete the MASK_PATH claim rule.

```
esxcli storage core claimrule remove -r rule#
```

- 2 Verify that the claim rule was deleted correctly.

```
esxcli storage core claimrule list
```

- 3 Reload the path claiming rules from the configuration file into the VMkernel.

```
esxcli storage core claimrule load
```

- 4 Run the `esxcli storage core claiming unclaim` command for each path to the masked storage device.

For example:

```
esxcli storage core claiming unclaim -t location -A vmhba0 -C 0 -T 0 -L 149
```

- 5 Run the path claiming rules.

```
esxcli storage core claimrule run
```

Results

Your host can now access the previously masked storage device.

Define NMP SATP Rules

The NMP SATP claim rules define which SATP manages a storage device. Usually, you can use the default SATPs provided for storage devices. If default settings are not sufficient, use the `esxcli` commands to change the SATP for a specific device.

You might need to create an SATP rule when you install a third-party SATP for a specific storage array.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 To add a claim rule for a specific SATP, run the **esxcli storage nmp satp rule add** command. The command takes the following options.

| Option | Description |
|---------------------------------|--|
| -b --boot | This rule is a system default rule added at boot time. Do not modify <code>esx.conf</code> or add to a host profile. |
| -c --claim-option=string | Set the claim option string when adding a SATP claim rule. |
| -e --description=string | Set the claim rule description when adding a SATP claim rule. |
| -d --device=string | Set the device when adding SATP claim rules. Device rules are mutually exclusive with vendor/model and driver rules. |
| -D --driver=string | Set the driver string when adding a SATP claim rule. Driver rules are mutually exclusive with vendor/model rules. |
| -f --force | Force claim rules to ignore validity checks and install the rule anyway. |
| -h --help | Show the help message. |
| -M --model=string | Set the model string when adding SATP a claim rule. Vendor/Model rules are mutually exclusive with driver rules. |
| -o --option=string | Set the option string when adding a SATP claim rule. |
| -P --psp=string | Set the default PSP for the SATP claim rule. |
| -O --psp-option=string | Set the PSP options for the SATP claim rule. |
| -s --satp=string | The SATP for which a new rule is added. |
| -R --transport=string | Set the claim transport type string when adding a SATP claim rule. |
| -t --type=string | Set the claim type when adding a SATP claim rule. |
| -V --vendor=string | Set the vendor string when adding SATP claim rules. Vendor/Model rules are mutually exclusive with driver rules. |

Note When searching the SATP rules to locate a SATP for a given device, the NMP searches the driver rules first. If there is no match, the vendor/model rules are searched, and finally the transport rules. If there is still no match, NMP selects a default SATP for the device.

- 2 Reboot your host.

Example: Defining an NMP SATP Rule

The following sample command assigns the VMW_SATP_INV plug-in to manage storage arrays with vendor string NewVend and model string NewMod.

```
# esxcli storage nmp satp rule add -V NewVend -M NewMod -s VMW_SATP_INV
```

When you run the **esxcli storage nmp satp list -s VMW_SATP_INV** command, you can see the new rule on the list of VMW_SATP_INV rules.

Scheduling Queues for Virtual Machine I/Os

By default, vSphere provides a mechanism that creates scheduling queues for every virtual machine file. Each file, for example `.vmdk`, gets its own bandwidth controls.

This mechanism ensures that I/O for a particular virtual machine file goes into its own separate queue and avoids interfering with I/Os from other files.

This capability is enabled by default. To turn it off, adjust the `VMkernel.Boot.isPerFileSchedModelActive` parameter in the advanced system settings.

Edit Per File I/O Scheduling

The advanced `VMkernel.Boot.isPerFileSchedModelActive` parameter controls the per file I/O scheduling mechanism. The mechanism is enabled by default.

If you turn off the per file I/O scheduling model, your host reverts to a legacy scheduling mechanism. The legacy scheduling maintains only one I/O queue for each virtual machine and storage device pair. All I/Os between the virtual machine and its virtual disks are moved into this queue. As a result, I/Os from different virtual disks might interfere with each other in sharing the bandwidth and affect each other's performance.

Note Do not disable per file scheduling if you have the HPP plug-in and the latency sensitive threshold parameter configured for high-speed local devices. Disabling per file scheduling might cause unpredictable behavior.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **System**, click **Advanced System Settings**.
- 4 Edit the value of the `VMkernel.Boot.isPerFileSchedModelActive` parameter.

| Option | Description |
|----------------|---|
| False | Disable the per file scheduling mechanism. |
| True (default) | Reenable the per file scheduling mechanism. |

- 5 Reboot the host for the changes to take effect.

Use esxcli Commands to Enable or Disable Per File I/O Scheduling

You can use the `esxcli` commands to change the I/O scheduling capability. The capability is enabled by default.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ To enable or disable per file I/O scheduling, run the following commands:

| Option | Description |
|--|---------------------------------|
| <code>esxcli system settings kernel set -s isPerFileSchedModelActive -v FALSE</code> | Disable per file I/O scheduling |
| <code>esxcli system settings kernel set -s isPerFileSchedModelActive -v TRUE</code> | Enable per file I/O scheduling |

Raw Device Mapping

19

Raw device mapping (RDM) provides a mechanism for a virtual machine to have direct access to a LUN on the physical storage subsystem.

The following topics contain information about RDMs and provide instructions on how to create and manage RDMs.

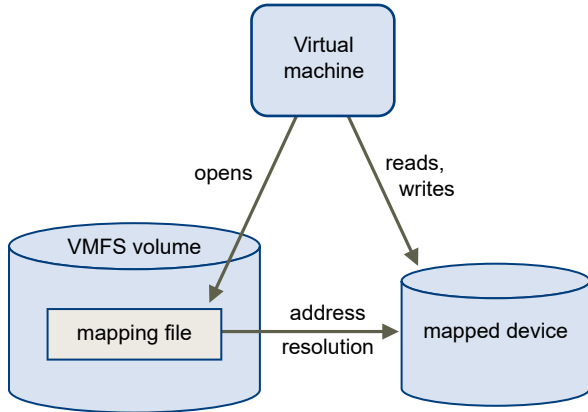
Read the following topics next:

- [About Raw Device Mapping](#)
- [Raw Device Mapping Characteristics](#)
- [Create Virtual Machines with RDMs](#)
- [Manage Paths for a Mapped LUN](#)
- [Virtual Machines with RDMs Must Ignore SCSI INQUIRY Cache](#)

About Raw Device Mapping

An RDM is a mapping file in a separate VMFS volume that acts as a proxy for a raw physical storage device. With the RDM, a virtual machine can access and use the storage device directly. The RDM contains metadata for managing and redirecting disk access to the physical device.

The file gives you some of the advantages of direct access to a physical device, but keeps some advantages of a virtual disk in VMFS. As a result, it merges the VMFS manageability with the raw device access.

Figure 19-1. Raw Device Mapping

Typically, you use VMFS datastores for most virtual disk storage. On certain occasions, you might use raw LUNs or logical disks located in a SAN.

For example, you might use raw LUNs with RDMs in the following situations:

- When SAN snapshot or other layered applications run in the virtual machine. The RDM enables backup offloading systems by using features inherent to the SAN.
- In any MSCS clustering scenario that spans physical hosts, such as virtual-to-virtual clusters and physical-to-virtual clusters. In this case, cluster data and quorum disks are configured as RDMs rather than as virtual disks on a shared VMFS.

Think of an RDM as a symbolic link from a VMFS volume to a raw LUN. The mapping makes LUNs appear as files in a VMFS volume. The RDM, not the raw LUN, is referenced in the virtual machine configuration. The RDM contains a reference to the raw LUN.

Two compatibility modes are available for RDMs:

- In the virtual compatibility mode, the RDM acts like a virtual disk file. The RDM can use snapshots.
- In the physical compatibility mode, the RDM offers direct access to the SCSI device for those applications that require lower-level control.

Benefits of Raw Device Mapping

An RDM provides a number of benefits, but it should not be used in every situation. In general, virtual disk files are preferable to RDMs for manageability. However, when you need raw devices, you must use the RDM.

RDM offers several benefits.

User-Friendly Persistent Names

Provides a user-friendly name for a mapped device. When you use an RDM, you do not need to refer to the device by its device name. You refer to it by the name of the mapping file, for example:

```
/vmfs/volumes/myVolume/myVMDirectory/myRawDisk.vmdk
```

Dynamic Name Resolution

Stores unique identification information for each mapped device. VMFS associates each RDM with its current SCSI device, regardless of changes in the physical configuration of the server because of adapter hardware changes, path changes, device relocation, and so on.

Distributed File Locking

Makes it possible to use VMFS distributed locking for raw SCSI devices. Distributed locking on an RDM makes it safe to use a shared raw LUN without losing data when two virtual machines on different servers try to access the same LUN.

File Permissions

Makes file permissions possible. The permissions of the mapping file are enforced at file-open time to protect the mapped volume.

File System Operations

Makes it possible to use file system utilities to work with a mapped volume, using the mapping file as a proxy. Most operations that are valid for an ordinary file can be applied to the mapping file and are redirected to operate on the mapped device.

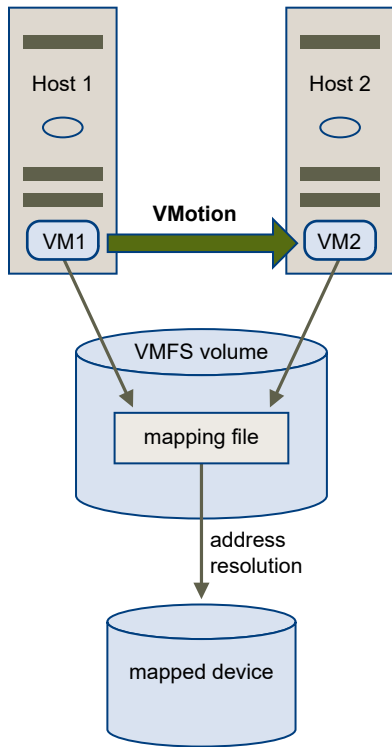
Snapshots

Makes it possible to use virtual machine snapshots on a mapped volume. Snapshots are not available when the RDM is used in physical compatibility mode.

vMotion

Lets you migrate a virtual machine with vMotion. The mapping file acts as a proxy to allow vCenter Server to migrate the virtual machine by using the same mechanism that exists for migrating virtual disk files.

Figure 19-2. vMotion of a Virtual Machine Using Raw Device Mapping



SAN Management Agents

Makes it possible to run some SAN management agents inside a virtual machine. Similarly, any software that needs to access a device by using hardware-specific SCSI commands can be run in a virtual machine. This kind of software is called SCSI target-based software. When you use SAN management agents, select a physical compatibility mode for the RDM.

N-Port ID Virtualization (NPIV)

Makes it possible to use the NPIV technology that allows a single Fibre Channel HBA port to register with the Fibre Channel fabric using several worldwide port names (WWPNs). This ability makes the HBA port appear as multiple virtual ports, each having its own ID and virtual port name. Virtual machines can then claim each of these virtual ports and use them for all RDM traffic.

Note You can use NPIV only for virtual machines with RDM disks.

VMware works with vendors of storage management software to ensure that their software functions correctly in environments that include ESXi. Some applications of this kind are:

- SAN management software
- Storage resource management (SRM) software
- Snapshot software
- Replication software

Such software uses a physical compatibility mode for RDMs so that the software can access SCSI devices directly.

Various management products are best run centrally (not on the ESXi machine), while others run well on the virtual machines. VMware does not certify these applications or provide a compatibility matrix. To find out whether a SAN management application is supported in an ESXi environment, contact the SAN management software provider.

RDM Considerations and Limitations

Certain considerations and limitations exist when you use RDMs.

- The RDM is not available for direct-attached block devices or certain RAID devices. The RDM uses a SCSI serial number to identify the mapped device. Because block devices and some direct-attach RAID devices do not export serial numbers, they cannot be used with RDMs.
- If you are using the RDM in physical compatibility mode, you cannot use a snapshot with the disk. Physical compatibility mode allows the virtual machine to manage its own, storage-based, snapshot or mirroring operations.

Virtual machine snapshots are available for RDMs with virtual compatibility mode.

- You cannot map to a disk partition. RDMs require the mapped device to be a whole LUN.
- If you use vMotion to migrate virtual machines with RDMs, make sure to maintain consistent LUN IDs for RDMs across all participating ESXi hosts.
- Flash Read Cache does not support RDMs in physical compatibility. Virtual compatibility RDMs are supported with Flash Read Cache.

Raw Device Mapping Characteristics

An RDM is a special mapping file in a VMFS volume that manages metadata for its mapped device. The mapping file is presented to the management software as an ordinary disk file, available for the usual file-system operations. To the virtual machine, the storage virtualization layer presents the mapped device as a virtual SCSI device.

Key contents of the metadata in the mapping file include the location of the mapped device (name resolution), the locking state of the mapped device, permissions, and so on.

RDM Virtual and Physical Compatibility Modes

You can use RDMs in virtual compatibility or physical compatibility modes. Virtual mode specifies full virtualization of the mapped device. Physical mode specifies minimal SCSI virtualization of the mapped device, allowing the greatest flexibility for SAN management software.

In virtual mode, the VMkernel sends only READ and WRITE to the mapped device. The mapped device appears to the guest operating system exactly the same as a virtual disk file in a VMFS volume. The real hardware characteristics are hidden. If you are using a raw disk in virtual mode, you can realize the benefits of VMFS such as advanced file locking for data protection and snapshots for streamlining development processes. Virtual mode is also more portable across storage hardware than physical mode, presenting the same behavior as a virtual disk file.

In physical mode, the VMkernel passes all SCSI commands to the device, with one exception: the REPORT LUNs command is virtualized so that the VMkernel can isolate the LUN to the owning virtual machine. Otherwise, all physical characteristics of the underlying hardware are exposed. Physical mode is useful to run SAN management agents or other SCSI target-based software in the virtual machine. Physical mode also allows virtual-to-physical clustering for cost-effective high availability.

VMFS5 and VMFS6 support greater than 2 TB disk size for RDMs in virtual and physical modes.

Dynamic Name Resolution

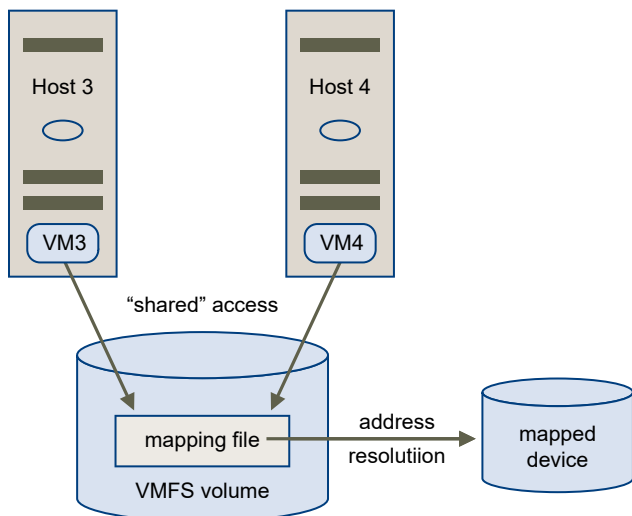
The RDM file supports dynamic name resolution when a path to a raw device changes.

VMFS uniquely identifies all mapped storage devices, and the identification is stored in its internal data structures. Any change in the path to a raw device, such as a Fibre Channel switch failure or the addition of a new HBA, can change the device name. Dynamic name resolution resolves these changes and automatically associates the original device with its new name.

Raw Device Mapping with Virtual Machine Clusters

Use an RDM with virtual machine clusters that require access to the same raw LUN for failover scenarios. The setup is similar to that of a virtual machine cluster that accesses the same virtual disk file, but an RDM replaces the virtual disk file.

Figure 19-3. Access from Clustered Virtual Machines



Comparing Available SCSI Device Access Modes

The ways of accessing a SCSI-based storage device include a virtual disk file on a VMFS datastore, virtual mode RDM, and physical mode RDM.

The following table provides a comparison of features available with the different modes.

Table 19-1. Features Available with Virtual Disks and Raw Device Mappings

| ESXi Features | Virtual Disk File | Virtual Mode RDM | Physical Mode RDM |
|------------------------------|-----------------------|--|--|
| SCSI Commands Passed Through | No | No | Yes <i>REPORT LUNs</i> is not passed through |
| vCenter Server Support | Yes | Yes | Yes |
| Snapshots | Yes | Yes | No |
| Distributed Locking | Yes | Yes | Yes |
| Clustering | Cluster-in-a-box only | Cluster-in-a-box cluster-across-boxes | Physical-to-virtual clustering cluster-across-boxes |
| SCSI Target-Based Software | No | No | Yes |

Use virtual disk files for the cluster-in-a-box type of clustering. If you plan to reconfigure your cluster-in-a-box clusters as cluster-across-boxes clusters, use virtual mode RDMs for the cluster-in-a-box clusters.

Create Virtual Machines with RDMs

When you give your virtual machine direct access to a raw SAN LUN, you create an RDM disk that resides on a VMFS datastore and points to the LUN. You can create the RDM as an initial disk for a new virtual machine or add it to an existing virtual machine. When creating the RDM, you specify the LUN to be mapped and the datastore on which to put the RDM.

Although the RDM disk file has the same `.vmdk` extension as a regular virtual disk file, the RDM contains only mapping information. The actual virtual disk data is stored directly on the LUN.

This procedure assumes that you are creating a new virtual machine. For information, see the *vSphere Virtual Machine Administration* documentation.

Procedure

- 1 Create a virtual machine.
 - a Right-click any inventory object that is a valid parent object of a virtual machine, such as a data center, folder, cluster, resource pool, or host, and select **New Virtual Machine**.
 - b Select **Create a new virtual machine** and click **Next**.
 - c Follow the steps required to create a virtual machine.
- 2 On the Customize Hardware page, click the **Virtual Hardware** tab.

3 (Optional) To delete the default virtual hard disk that the system created for your virtual machine, move your cursor over the disk and click the **Remove** icon.

4 Add an RDM disk.

- a Click **Add New Devices** and select **RDM Disk** from the list.
- b From the list of LUNs, select a target raw LUN and click **OK**.

The system creates an RDM disk that maps your virtual machine to the target LUN. The RDM disk is shown on the list of virtual devices as a new hard disk.

5 Configure the RDM disk.

- a Click the **New Hard Disk** triangle to expand the properties for the RDM disk.
- b Select a location for the RDM.

You can place the RDM on the same datastore where your virtual machine configuration files reside, or select a different datastore.

Note To use vMotion for virtual machines with enabled NPIV, make sure that the RDM files and the virtual machine files are located on the same datastore. You cannot perform Storage vMotion when NPIV is enabled.

- c Select a compatibility mode.

| Option | Description |
|-----------------|--|
| Physical | Allows the guest operating system to access the hardware directly. Physical compatibility is useful if you are using SAN-aware applications on the virtual machine. However, a virtual machine with a physical compatibility RDM cannot be cloned, made into a template, or migrated if the migration involves copying the disk. |
| Virtual | Allows the RDM to behave as if it were a virtual disk, so you can use such features as taking snapshots, cloning, and so on. When you clone the disk or make a template out of it, the contents of the LUN are copied into a .vmdk virtual disk file. When you migrate a virtual compatibility mode RDM, you can migrate the mapping file or copy the contents of the LUN into a virtual disk. |

- d If you selected virtual compatibility mode, select a disk mode.

Disk modes are not available for RDM disks using physical compatibility mode.

| Option | Description |
|------------------------------------|--|
| Dependent | Dependent disks are included in snapshots. |
| Independent - Persistent | Disks in persistent mode behave like conventional disks on your physical computer. All data written to a disk in persistent mode are written permanently to the disk. |
| Independent - Nonpersistent | Changes to disks in nonpersistent mode are discarded when you power off or reset the virtual machine. With nonpersistent mode, you can restart the virtual machine with a virtual disk in the same state every time. Changes to the disk are written to and read from a redo log file that is deleted when you power off or reset. |

- 6 Complete your virtual machine configuration.

Manage Paths for a Mapped LUN

When you use virtual machines with RDMs, you can manage paths for mapped raw LUNs.

Note This task is available only in the vSphere Web Client.

Procedure

- 1 Right-click the virtual machine and select **Edit Settings**.
- 2 Click the **Virtual Hardware** tab and click **Hard Disk** to expand the disk options menu.
- 3 Click **Manage Paths** in the Multipathing field.
- 4 Use the Edit Multipathing Policies dialog box to enable or disable paths, set multipathing policy, and specify the preferred path.

For information on managing paths, see [Chapter 18 Understanding Multipathing and Failover](#).

Virtual Machines with RDMs Must Ignore SCSI INQUIRY Cache

Certain virtual machines with RDMs must obtain the SCSI INQUIRY information from the LUN instead of using SCSI INQUIRY data cached by ESXi.

Problem

Certain guest operating systems or applications that run in the virtual machines with the RDMs display unpredictable behavior.

Cause

This behavior might be caused by cached SCSI INQUIRY data that interferes with specific guest operating systems and applications.

When the ESXi host first connects to a target storage device, it issues the SCSI INQUIRY command to obtain basic identification data from the device. By default, ESXi caches the received SCSI INQUIRY data (Standard, page 80, and page 83), and the data remains unchanged afterwards. Responses for subsequent SCSI INQUIRY commands are returned from the cache.

However, specific guest operating systems running in virtual machines with RDMs must query the LUN instead of using SCSI INQUIRY data cached by ESXi. In these cases, you can configure the VM to ignore the SCSI INQUIRY cache.

Solution

- ◆ Use one of the following methods.

Make the changes only when your storage vendor recommends that you do so.

| Option | Description |
|---|--|
| Modify the .vmx file of the virtual machine with the RDM | <p>Use this method for the VMs with hardware version 8 or later.</p> <p>a Add the following parameter to the file:</p> <pre>scsix:y.ignoreDeviceInquiryCache = "true"</pre> <p>where <i>x</i> is the SCSI controller number and <i>y</i> is the SCSI target number of the RDM.</p> <p>b Reboot the VM.</p> |
| Use the <code>esxcli</code> command | <p>Because you configure the setting at a host level, no VM hardware version limitations apply.</p> <pre>esxcli storage core device inquirycache set --device device id --ignore true</pre> <p>No VM reboot is required.</p> |

No matter which method you use to set the SCSI INQUIRY cache parameter to true, the VM starts contacting the LUN directly for the SCSI INQUIRY data.

| ignoreDeviceInquiryCache parameter in vmx | ignore inquirycache parameter in esxcli | Inquiry request serviced from |
|--|--|--------------------------------------|
| True | True | LUN |
| False (default if the parameter is not present) | True | LUN |
| True | False | LUN |
| False (default if the parameter is not present) | False | Cache |

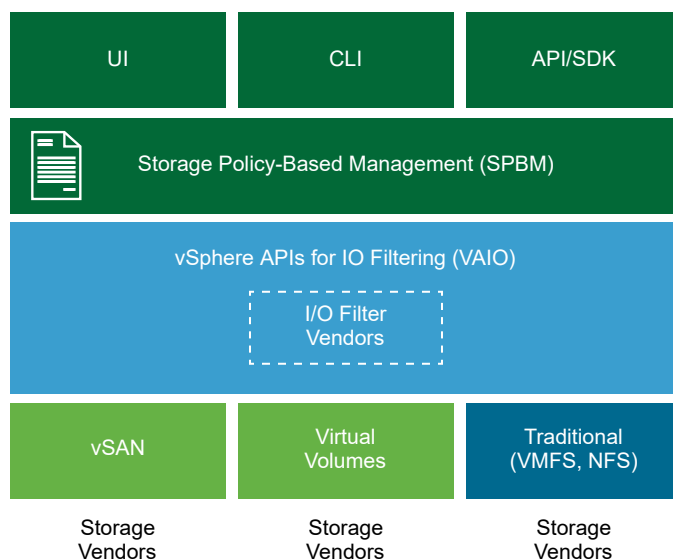
Storage Policy Based Management

20

Within a software-defined data center, Storage Policy Based Management (SPBM) plays a major role by helping to align storage with application demands of your virtual machines. It provides a storage policy framework that serves as a single unified control panel across a broad range of data services and storage solutions.

As an abstraction layer, SPBM abstracts storage services delivered by Virtual Volumes, vSAN, I/O filters, or other storage entities.

Rather than integrating with each individual type of storage and data services, SPBM provides a universal framework for different types of storage entities.



SPBM offers the following mechanisms:

- Advertisement of storage capabilities and data services that storage arrays and other entities, such as I/O filters, offer.
- Bidirectional communications between ESXi and vCenter Server on one side, and storage arrays and entities on the other.
- Virtual machine provisioning based on VM storage policies.

Read the following topics next:

- [Virtual Machine Storage Policies](#)

- [Workflow for Virtual Machine Storage Policies](#)
- [Populating the VM Storage Policies Interface](#)
- [About Rules and Rule Sets](#)
- [Creating and Managing VM Storage Policies](#)
- [About Storage Policy Components](#)
- [Storage Policies and Virtual Machines](#)
- [Default Storage Policies](#)

Virtual Machine Storage Policies

Virtual machine storage policies are essential to virtual machine provisioning through SPBM. The policies control which type of storage is provided for the virtual machine and how the virtual machine is placed within storage. They also determine data services that the virtual machine can use.

vSphere offers default storage policies. In addition, you can define policies and assign them to the virtual machines.

You use the VM Storage Policies interface to create a storage policy. When you define the policy, you specify various storage requirements for applications that run on the virtual machines. You can also use storage policies to request specific data services, such as caching or replication, for virtual disks.

You apply the storage policy when you create, clone, or migrate the virtual machine. After you apply the storage policy, the SPBM mechanism assists you with placing the virtual machine in a matching datastore. In certain storage environments, SPBM determines how the virtual machine storage objects are provisioned and allocated within the storage resource to guarantee the required level of service. The SPBM also enables requested data services for the virtual machine and helps you to monitor policy compliance.

Workflow for Virtual Machine Storage Policies

The entire process of creating and managing storage policies typically includes several steps.

Whether you must perform a specific step might depend on the type of storage or data services that your environment offers.

| Step | Description |
|---|---|
| Populate the VM Storage Policies interface with appropriate data. | <p>The VM Storage Policies interface is populated with information about datastores and data services that are available in your storage environment. This information is obtained from storage providers and datastore tags.</p> <ul style="list-style-type: none"> ■ For entities represented by storage providers, verify that an appropriate provider is registered. <p>Entities that use the storage provider include vSAN, Virtual Volumes, and I/O filters. Depending on the type of storage entity, some providers are self-registered. Other providers must be manually registered.</p> <p>See Use Storage Providers to Populate the VM Storage Policies Interface and Register Storage Providers for Virtual Volumes.</p> <ul style="list-style-type: none"> ■ Tag datastores that are not represented by storage providers. You can also use tags to indicate a property that is not communicated through the storage provider, such as geographical location or administrative group. <p>See Assign Tags to Datastores.</p> |
| Create predefined storage policy components. | <p>A storage policy component describes a single data service, such as replication, that must be provided for the virtual machine. You can define the component in advance and associate it with multiple VM storage policies. The components are reusable and interchangeable.</p> <p>See Create Storage Policy Components.</p> |
| Create VM storage policies. | <p>When you define storage policies for virtual machines, you specify storage requirements for applications that run on the virtual machines.</p> <p>See Creating and Managing VM Storage Policies.</p> |
| Apply the VM storage policy to the virtual machine. | <p>You can apply the storage policy when deploying the virtual machine or configuring its virtual disks.</p> <p>See Assign Storage Policies to Virtual Machines.</p> |
| Check compliance for the VM storage policy. | <p>Verify that the virtual machine uses the datastore that is compliant with the assigned storage policy.</p> <p>See Check Compliance for a VM Storage Policy.</p> |

To create and manage your storage policies, you use the VM Storage Policy interface of the vSphere Client.

Populating the VM Storage Policies Interface

Before you start creating VM storage policies, you must populate the VM Storage Policy interface with information about storage entities and data services that are available in your storage environment.

This information is obtained from storage providers, also called VASA providers. Another source is datastore tags.

Storage Capabilities and Services

Certain datastores, for example, Virtual Volumes and vSAN, are represented by the storage providers. Through the storage providers, the datastores can advertise their capabilities in the VM Storage Policy interface. These datastore capabilities, data services, and other characteristics with ranges of values populate the VM Storage Policy interface.

You use these characteristics when you define datastore-based placement and service rules for your storage policy.

Data Services

I/O filters on your hosts are also represented by the storage providers. The storage provider delivers information about the data services of the filters to the VM Storage Policy interface. You use this information when defining the rules for host-based data services, also called common rules. Unlike the datastore-specific rules, these rules do not define storage placement and storage requirements for the virtual machine. Instead, they activate the requested I/O filter data services for the virtual machine.

Tags

Generally, VMFS and NFS datastores are not represented by a storage provider. They do not display their capabilities and data services in the VM Storage Policies interface. You can use tags to encode information about these datastores. For example, you can tag your VMFS datastores as VMFS-Gold and VMFS-Silver to represent different levels of service.

For VVols and vSAN datastores, you can use tags to encode information that is not advertised by the storage provider, such as geographical location (Palo Alto), or administrative group (Accounting).

Similar to the storage capabilities and characteristics, all tags associated with the datastores appear in the VM Storage Policies interface. You can use the tags when you define the tag-based placement rules.

Use Storage Providers to Populate the VM Storage Policies Interface

For entities represented by storage (VASA) providers, verify that an appropriate provider is registered. After the storage providers are registered, the VM Storage Policies interface becomes populated with information about datastores and data services that the providers represent.

Entities that use the storage provider include vSAN, Virtual Volumes, and I/O filters. Depending on the type of the entity, some providers are self-registered. Other providers, for example, the Virtual Volumes storage provider, must be manually registered. After the storage providers are registered, they deliver the following data to the VM Storage Policies interface:

- Storage capabilities and characteristics for such datastores as Virtual Volumes and vSAN.
- Data services the I/O filters provide.

Prerequisites

Register the storage providers that require manual registration. For more information, see the appropriate documentation:

- *Administering VMware vSAN*
- [Chapter 22 Working with Virtual Volumes](#)

■ [Chapter 23 Filtering Virtual Machine I/O](#)

Procedure

- 1 Browse to the vCenter Server instance.
- 2 Click the **Configure** tab, and click **Storage Providers**.
- 3 In the Storage Providers list, view the storage providers registered with vCenter Server.

The list shows general information including the name of the storage provider, its URL and status, storage entities that the provider represents, and so on.

- 4 To display more details, select a specific storage provider or its component from the list.

Assign Tags to Datastores

Use tags to encode information about a datastore. The tags are helpful when your datastore is not represented by a storage provider and does not advertise its services in the VM Storage Policies interface. You can also use the tags to indicate a property that is not communicated through a storage provider, such as a geographical location or administrative group.

You can apply a new tag that contains general storage information to a datastore. For more details about the tags, their categories, and how to manage the tags, see the *vCenter Server and Host Management* documentation.

Prerequisites

Required privileges:

- **vSphere Tagging.Create vSphere Tag** on the root vCenter Server instance
- **vSphere Tagging.Create vSphere Tag Category** on the root vCenter Server instance
- **vSphere Tagging.Assign or Unassign vSphere Tag** on the root vCenter Server instance

Procedure

- 1 Create a category for storage tags.
 - a From the Home menu, click **Tags & Custom Attributes**.
 - b Click the **Tags** tab and click **Categories**.
 - c Click the **Add Category** icon.

- d Specify the category properties. See the following example.

| Category Property | Example |
|-------------------------|--|
| Category Name | Storage Location |
| Description | Category for tags related to location of storage |
| Tags Per Object | Many tags |
| Associable Object Types | Datastore and Datastore Cluster |

- e Click **OK**.

2 Create a storage tag.

- a On the **Tags** tab, click **Tags**.
- b Click the **Add Tag** icon.
- c Specify the properties for the tag. See the following example.

| Tag Property | Example |
|--------------|----------------------------|
| Name | Texas |
| Description | Datastore located in Texas |
| Category | Storage Location |

- d Click **OK**.

3 Apply the tag to the datastore.

- a Navigate to the datastore.
- b Right-click the datastore, and select **Tags & Custom Attributes > Assign Tag**.
- c From the list of tags, select an appropriate tag, for example, Texas in the Storage Location category, and click **Assign**.

Results

The new tag is assigned to the datastore and appears on the datastore **Summary** tab in the **Tags** pane.

What to do next

When creating a VM storage policy, you can reference the tag to include the tagged datastore in the list of compatible storage resources. See [Create Storage-Specific Rules for a VM Storage Policy](#).

Or you can exclude the tagged datastore from the VM storage policy. For example, your VM storage policy can include Virtual Volumes datastores located in Texas and California, but exclude datastores located in Nevada.

To learn more about how to use tags in VM storage policies, watch the following video.



(Using Tags in Storage Policies)

About Rules and Rule Sets

After the VM Storage Policies interface is populated with the appropriate data, you can start creating your storage policies. Creating a policy involves defining specific storage placement rules and rules to configure data services.

Rules

The rule is a basic element of the VM storage policy. Each individual rule is a statement that describes a single requirement for virtual machine storage and data services.

Rule Sets

Within a storage policy, individual rules are organized into collections of rules, or rule sets. Typically, the rule sets can be in one of the following categories: rules for host-based services and datastore-specific rules.

Datastore-Specific Rule Sets

Each rule set must include placement rules that describe requirements for virtual machine storage resources. All placement rules within a single rule set represent a single storage entity. These rules can be based on storage capabilities or tags.

In addition, the datastore-specific rule set can include optional rules or storage policy components that describe data services to provide for the virtual machine. Generally, these rules request such services as caching, replication, other services provided by storage systems.

To define the storage policy, one datastore-specific set is required. Additional rule sets are optional. A single policy can use multiple sets of rules to define alternative storage placement parameters, often from several storage providers.

Placement Rules: Capability-Based

Placement rules specify a particular storage requirement for the VM and enable SPBM to distinguish compatible datastores among all datastores in the inventory. These rules also describe how the virtual machine storage objects are allocated within the datastore to receive the required level of service. For example, the rules can list Virtual Volumes as a destination and define the maximum recovery point objective (RPO) for the Virtual Volumes objects.

When you provision the virtual machine, these rules guide the decision that SPBM makes about the virtual machine placement. SPBM finds the Virtual Volumes datastores that can match the rules and satisfy the storage requirements of the virtual machine. See [Create a VM Storage Policy for Virtual Volumes](#).

Placement Rules: Tag-Based

Tag-based rules reference datastore tags. These rules can define the VM placement, for example, request as a target all datastores with the VMFS-Gold tag. You can also use the tag-based rules to fine-tune your VM placement request further. For example, exclude datastores with the Palo Alto tag from the list of your Virtual Volumes datastores. See [Create a VM Storage Policy for Tag-Based Placement](#).

Rules for Host-Based Services

This rule set activates data services provided by the host. The set for host-based services can include rules or storage policy components that describe particular data services, such as encryption or replication.

Unlike datastore-specific rules, this set does not include placement rules. Rules for host-based services are generic for all types of storage and do not depend on the datastore. See [Create a VM Storage Policy for Host-Based Data Services](#).

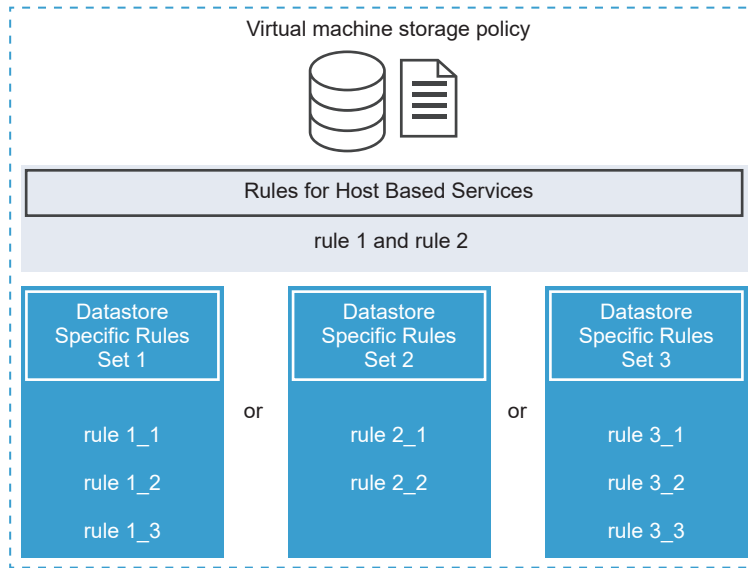
Table 20-1. Structure of a VM Storage Policy

| Rules for Host-Based Services | Datastore-Specific Rule Sets |
|---|---|
| Rules or predefined storage policy components to activate data services installed on ESXi hosts. For example, replication by I/O filters. | Capability-based or tag-based placement rules that describe requirements for virtual machine storage resources. For example, Virtual Volumes placement. |
| | Rules or predefined storage policy components that activate data services provided by storage. For example, caching by Virtual Volumes. |

Relationships Between Rules and Rule Sets

The boolean operator **OR** defines the relationship between the datastore-specific rule sets within the policy. The **AND** operator defines the relationship between all rules within a single rule set. The policy can contain only a rule set for host-based services, or only a datastore-specific rule set, or both.

If the rule set for host-based services is not present, meeting all the rules of a single datastore-specific rule set is sufficient to satisfy the entire policy. If the rule set for host-based services is present, the policy matches the datastore that satisfies the host services rules and all rules in one of the datastore-specific sets.



Creating and Managing VM Storage Policies

To create and manage storage policies for your virtual machines, you use the VM Storage Policies interface.

Depending on whether you use the vSphere Web Client or the vSphere Client, the appearance of the VM Storage Policy interface and its options might change.

Define a VM Storage Policy in the vSphere Web Client

When you define storage policies for virtual machines, you specify storage requirements for applications that run on the virtual machines.

A storage policy can reference storage capabilities that are advertised by a storage entity. Or it can reference datastore tags. The policy can include components that enable data services, such as replication or caching, provided by I/O filters, storage systems, or other entities.

Prerequisites

- Make sure that the VM Storage Policies interface is populated with information about storage entities and data services that are available in your storage environment. See [Populating the VM Storage Policies Interface](#).
- Define appropriate storage policy components. See [Create Storage Policy Components](#).
- Required privileges: **VM storage policies.Update** and **VM storage policies.View**.

Procedure

1 Start VM Storage Policy Creation Process

To define a virtual machine storage policy, use the **Create New VM Storage Policy** wizard.

2 Define Common Rules for a VM Storage Policy

On the **Common rules** page, specify which data services to include in the VM storage policy. The data services are provided by software components that are installed on your ESXi hosts and vCenter Server. The VM storage policy that includes common rules activates specified data services for the virtual machine.

3 Create Storage-Specific Rules for a VM Storage Policy

Use the **Rule Set** page to define storage placement rules. If your storage provides additional data services, such as replication, use the page to specify which data services to include in the VM storage policy.

4 Finish VM Storage Policy Creation

You can review the list of datastores that are compatible with the VM storage policy and change any storage policy settings.

What to do next

You can apply this storage policy to virtual machines. If you use object-based storage, such as vSAN and Virtual Volumes, you can designate this storage policy as the default.

Start VM Storage Policy Creation Process

To define a virtual machine storage policy, use the **Create New VM Storage Policy** wizard.

Procedure

- 1 From the vSphere Web Client Home, click **Policies and Profiles > VM Storage Policies**.
- 2 Click the **VM Storage Policies** tab.
- 3 Click the **Create a New VM Storage Policy** icon.
- 4 Select the vCenter Server instance.
- 5 Type a name and a description for the storage policy.

Define Common Rules for a VM Storage Policy

On the **Common rules** page, specify which data services to include in the VM storage policy. The data services are provided by software components that are installed on your ESXi hosts and vCenter Server. The VM storage policy that includes common rules activates specified data services for the virtual machine.

The data services are generic for all types of storage and do not depend on a datastore. Depending on your environment, the data services can belong to various categories, including encryption, caching, replication, and so on. Certain data services, such as encryption, are provided by VMware. Others are offered by third-party I/O filters.


Prerequisites

- For information about encrypting your virtual machines, see the *vSphere Security* documentation.

- For information about I/O filters, see [Chapter 23 Filtering Virtual Machine I/O](#).
- For information about storage policy components, see [About Storage Policy Components](#).

Procedure

- 1 Enable common rules by selecting **Use common rules in the VM storage policy**.
- 2 Click the **Add component** (+) icon and select a data service category from the drop-down menu, for example, Replication.
- 3 Define rules for the data service category by specifying an appropriate provider and values for the rules. Or select the data service from the list of predefined components.

| Option | Description |
|---|--|
|  Component Name | This option is available if you have predefined storage policy components in your database. If you know which component to use, select it from the list to add to the VM storage policy. |
| See all | Review all component available for the category. To include a specific component, select it from the list and click OK . |
| Custom | Define custom rules for the data service category by specifying an appropriate provider and values for the rules. |

- 4 Add more components to request other data services.

You can use only one component from the same category, for example caching, per a set of common or regular rules.

- 5 Click **Next**.

Create Storage-Specific Rules for a VM Storage Policy

Use the **Rule Set** page to define storage placement rules. If your storage provides additional data services, such as replication, use the page to specify which data services to include in the VM storage policy.

Prerequisites

- If your environment includes storage entities such as vSAN or Virtual Volumes, review these functionalities. For information, see the *Administering VMware vSAN* documentation and [Chapter 22 Working with Virtual Volumes](#).
- To configure predefined storage policy components, see [About Storage Policy Components](#).

Procedure

- 1 Make sure that the **Use rule-sets in the storage policy** check box is selected.

2 Define placement rules.


Placement rules request a specific storage entity as a destination for the virtual machine. They can be capability-based or tag-based. Capability-based rules are based on data services that storage entities such as vSAN and Virtual Volumes advertise through storage (VASA) providers. Tag-based rules reference tags that you assign to datastores.

| Option | Description |
|--|---|
| Placement based on storage capabilities | <ol style="list-style-type: none"> From the Storage Type drop-down menu, select a target storage entity, for example, Virtual Volumes. From the Add rule drop-down menu, select a capability and specify its value. For example, you can specify the number of read operations per second for Virtual Volumes objects. You can include as many rules as you need for the selected storage entity. Verify that the values you provide are within the range of values that the storage resource advertises. If you need to fine-tune your placement request further, add a tag-based rule. Tag-based rules can include or exclude specific placement criteria. For example, you can exclude datastores with the Palo Alto tag from the list of your target Virtual Volumes datastores. |
| Placement based on tags | <ol style="list-style-type: none"> From the Storage Type drop-down menu, select Tags based placement. From the Add rule drop-down menu, select Tags from category. Define tag-based placement criteria. For example, you can request as a target all datastores with the VMFS-Gold tag. |

3 (Optional) Select data services to include in the VM storage policy.

The data services that you reference on the Rule Set page are provided by the storage. The VM storage policy that references the data services, requests them for the virtual machine.

- a Click the **Add component** (+) icon and select a data service category from the drop-down menu, for example, Replication.
- b Define rules for the data service category by specifying an appropriate provider and values for the rules. Or select the data service from the list of predefined components.

| Option | Description |
|---|--|
|  Component Name | This option is available if you have predefined storage policy components in your database. If you know which component to use, select it from the list to add to the VM storage policy. |
| See all | Review all component available for the category. To include a specific component, select it from the list and click OK . |
| Custom | Define custom rules for the data service category by specifying an appropriate provider and values for the rules. |

- c Add more components to request other data services.

You can use only one component from the same category, for example caching, per a set of common or regular rules.

4 (Optional) To define another rule set, click **Add another rule set** and repeat Step 2 through Step 3.

Multiple rule sets allow a single policy to define alternative storage placement parameters, often from several storage providers.

5 Click **Next**.

Finish VM Storage Policy Creation

You can review the list of datastores that are compatible with the VM storage policy and change any storage policy settings.

Procedure

- 1 On the **Storage compatibility** page, review the list of datastores that match this policy and click **Next**.

To be eligible, the datastore must satisfy at least one rule set and all rules within this set.

- 2 On the **Ready to complete** page, review the storage policy settings.

If you need to change any settings, click **Back** to go back to the relevant page.

- 3 Click **Finish**.

Results

The VM storage policy appears in the list.

Create a VM Storage Policy for Host-Based Data Services

To define the VM storage policy in the vSphere Client, use the **Create VM Storage Policy** wizard. In this task, you create rules for data services offered by ESXi hosts. The VM storage policy that includes these rules activates specified data services for the virtual machine.

Available data services include encryption, I/O control, caching, and so on. Certain data services, such as encryption, are provided by VMware. Others can be offered by third-party I/O filters that you install on your host.

The data services are usually generic for all types of storage and do not depend on a datastore. Adding datastore-specific rules to the storage policy is optional.

If you add datastore-specific rules, and both the I/O filters on the host and storage offer the same type of service, for example, encryption, your policy can request this service from both providers. As a result, the virtual machine data is encrypted twice, by the I/O filter and your storage. However, replication provided by Virtual Volumes and replication provided by the I/O filter cannot coexist in the same storage policy.

Prerequisites

- For information about encrypting your virtual machines, see the *vSphere Security* documentation.
- For information about I/O filters, see [Chapter 23 Filtering Virtual Machine I/O](#).
- For information about storage policy components, see [About Storage Policy Components](#).
- Required privileges: **VM storage policies.Update** and **VM storage policies.View**.

Procedure

- 1 Open the **Create VM Storage Policy** wizard.
 - a Click **Menu > Policies and Profiles**.
 - b Under **Policies and Profiles**, click **VM Storage Policies**
 - c Click **Create VM Storage Policy**.
- 2 Enter the policy name and description, and click **Next**.

| Option | Action |
|----------------|--|
| vCenter Server | Select the vCenter Server instance. |
| Name | Enter the name of the storage policy. |
| Description | Enter the description of the storage policy. |

- 3 On the **Policy structure** page under **Host based services**, enable host-based rules.

- 4 On the **Host based services** page, define rules to enable and configure data services provided by your host.
 - a Click the tab for the data service category, for example, **Replication**.
 - b Define custom rules for the data service category or use predefined components.

| Option | Description |
|------------------------------|--|
| Disabled | Host-based services are disabled by default. |
| Use storage policy component | Select a storage policy component from the drop-down menu. This option is available only if you have predefined components in your database. |
| Custom | Define custom rules for the data service category by specifying an appropriate provider and values for the rules. |

Note You can enable several data services. If you use encryption with other data services, set the **Allow I/O filters before encryption** parameter to **True**, so that other services, such as replication, can analyze clear text data before it is encrypted.

- 5 On the **Storage compatibility** page, review the list of datastores that match this policy.
 To be compatible with the policy for host-based services, datastores must be connected to the host that provides these services. If you add datastore-specific rule sets to the policy, the compatible datastores must also satisfy storage requirements of the policy.
- 6 On the **Review and finish** page, review the storage policy settings and click **Finish**.
 To change any settings, click **Back** to go to the relevant page.

Results

The new VM storage policy for host-based data services appears on the list.

Create a VM Storage Policy for Virtual Volumes

To define the VM storage policy in the vSphere Client, use the **Create VM Storage Policy** wizard. In this task, you create a custom storage policy compatible with Virtual Volumes. When you define the VM storage policy for Virtual Volumes, you create rules to configure storage and data services provided by the VVols datastore. The rules are applied when the VM is placed on the VVols datastore. The custom storage policy can replace the default No Requirements storage policy for Virtual Volumes that VMware provides.

The procedure assumes that you are creating the VM storage policy for Virtual Volumes. For information about the vSAN storage policy, see the *Administering VMware vSAN* documentation.

Prerequisites

- Verify that the Virtual Volumes storage provider is available and active. See [Register Storage Providers for Virtual Volumes](#).

- Make sure that the VM Storage Policies interface is populated with information about storage entities and data services that are available in your storage environment. See [Populating the VM Storage Policies Interface](#).
- Define appropriate storage policy components. See [Create Storage Policy Components](#).
- Required privileges: **VM storage policies.Update** and **VM storage policies.View**.

Procedure

- 1 Open the **Create VM Storage Policy** wizard.
 - a Click **Menu > Policies and Profiles**.
 - b Under **Policies and Profiles**, click **VM Storage Policies**.
 - c Click **Create VM Storage Policy**.
- 2 Enter the policy name and description, and click **Next**.

| Option | Action |
|-----------------------|---|
| vCenter Server | Select the vCenter Server instance. |
| Name | Enter the name of the storage policy, for example VVols Storage Policy. |
| Description | Enter the description of the storage policy. |

- 3 On the **Policy structure** page under Datastore specific rules, enable rules for a target storage entity, such as Virtual Volumes storage.

You can enable rules for several datastores. Multiple rule sets allow a single policy to define alternative storage placement parameters, often from several storage providers.

- 4 On the *Virtual Volumes* rules page, define storage placement rules for the target VVols datastore.

- a Click the **Placement** tab and click **Add Rule**.
- b From the Add Rule drop-down menu, select available capability and specify its value.

For example, you can specify the number of read operations per second for the Virtual Volumes objects.

You can include as many rules as you need for the selected storage entity. Verify that the values you provide are within the range of values that the VVols datastore advertises.

- c To fine-tune your placement request further, click the **Tags** tab and add a tag-based rule.

Tag-based rules can filter datastores by including or excluding specific placement criteria. For example, your VM storage policy can include Virtual Volumes datastores located in Texas and California, but exclude datastores located in Nevada.

5 (Optional) Define rules to configure datastore-specific services.

The data services, such as encryption, caching, or replication, are offered by the storage. The VM storage policy that references data services, requests these services for the VM when the VM is placed to the VVols datastore.

- a Click the tab for the data service category, for example, **Replication**.
- b Define custom rules for the data service category or use predefined components.

| Option | Description |
|------------------------------|--|
| Disabled | Datastore-specific services are disabled by default. |
| Use storage policy component | Select a storage policy component from the drop-down menu. This option is available only if you have predefined components in your database. |
| Custom | Define custom rules for the data service category by specifying an appropriate provider and values for the rules. |

6 On the **Storage compatibility** page, review the list of datastores that match this policy.

If the policy includes several rule sets, the datastore must satisfy at least one rule set and all rules within this set.

7 On the **Review and finish** page, review the storage policy settings and click **Finish**.

To change any settings, click **Back** to go to the relevant page.

Results

The new VM storage policy compatible with Virtual Volumes appears on the list.

What to do next

You can now associate this policy with a virtual machine, or designate the policy as default.

Create a VM Storage Policy for Tag-Based Placement

Tag-based rules reference the tags that you assign to the datastores and can filter the datastores to be used for placement of the VMs. To define tag-based placement in the vSphere Client, use the **Create VM Storage Policy** wizard.

Prerequisites

- Make sure that the VM Storage Policies interface is populated with information about storage entities and data services that are available in your storage environment. See [Populating the VM Storage Policies Interface](#).
- Required privileges: **VM storage policies.Update** and **VM storage policies.View**.

Procedure

- 1 Open the **Create VM Storage Policy** wizard.
 - a Click **Menu > Policies and Profiles**.
 - b Under **Policies and Profiles**, click **VM Storage Policies**.
 - c Click **Create VM Storage Policy**.
- 2 Enter the policy name and description, and click **Next**.

| Option | Action |
|-----------------------|--|
| vCenter Server | Select the vCenter Server instance. |
| Name | Enter the name of the storage policy. |
| Description | Enter the description of the storage policy. |

- 3 On the **Policy structure** page under Datastore specific rules, enable tag-based placement rules.
- 4 On the **Tag based placement** page, create the tag rules.
 - a Click **Add Tag Rule** and define tag-based placement criteria. Use the following as an example.

| Option | Example |
|---------------------|-------------------------|
| Tag category | Level of Service |
| Usage option | Use storage tagged with |
| Tags | Gold |

All datastores with the Gold tag become compatible as the storage placement target.

- b (Optional) Add more tag-based rules.
- 5 On the **Storage compatibility** page, review the list of datastores that match this policy.
- 6 On the **Review and finish** page, review the storage policy settings and click **Finish**.
To change any settings, click **Back** to go to the relevant page.

Results

The new VM storage policy compatible with tagged datastores appears on the list.

Edit or Clone a VM Storage Policy

If storage requirements for virtual machines and virtual disks change, you can modify the existing storage policy. You can also create a copy of the existing VM storage policy by cloning it. While cloning, you can optionally select to customize the original storage policy.

Prerequisites

Required privilege: **StorageProfile.View**

Procedure

- 1 Navigate to the storage policy.

| Option | Description |
|---------------------------|---|
| In the vSphere Web Client | <ol style="list-style-type: none"> a From the Home menu, click Policies and Profiles > VM Storage Policies. b Click the VM Storage Policies tab. |
| In the vSphere Client | <ol style="list-style-type: none"> a Click Menu > Policies and Profiles. b Under Policies and Profiles, click VM Storage Policies. |

- 2 Select the storage policy, and click one of the following icons:

- **Edit Settings**
- **Clone**

- 3 (Optional) Modify the policy and click **OK**.

- 4 If editing the storage policy that is used by a virtual machine, reapply the policy to the virtual machine.

| Option | Description |
|----------------|---|
| Manually later | If you select this option, the compliance status for all virtual disks and virtual machine home objects associated with the storage policy changes to Out of Date. To update configuration and compliance, manually reapply the storage policy to all associated entities. See Reapply Virtual Machine Storage Policy . |
| Now | Update virtual machine and compliance status immediately after editing the storage policy. |

About Storage Policy Components

A VM storage policy can include one or several reusable and interchangeable building blocks, called storage policy components. Each component describes a particular data service to be provided for the virtual machine. You can define the policy components in advance and associate them with multiple VM storage policies.

You cannot assign the predefined component directly to a virtual machine or virtual disk. Instead, you must add the component to the VM storage policy, and assign the policy to the virtual machine.

The component describes one type of service from one service provider. The services can vary depending on the providers that you use, but generally belong in one of the following categories.

- Compression
- Caching
- Encryption
- Replication

When you create the storage policy component, you define the rules for one specific type and grade of service.

The following example shows that virtual machines VM1 and VM2 have identical placement requirements, but must have different grades of replication services. You can create the storage policy components with different replication parameters and add these components to the related storage policies.

Table 20-2. Storage Policy Components

| Virtual Machine | Placement Rules | Storage Policy Component |
|--|-----------------|--------------------------|
| VM1 requires replication every 2 hours | VVols Datastore | 2-hour Replication |
| VM2 requires replication every 4 hours | VVols Datastore | 4-hour Replication |

The provider of the service can be a storage system, an I/O filter, or another entity. If the component references an I/O filter, the component is added to the host-based rules of the storage policy. Components that reference entities other than the I/O filters, for example, a storage system, are added to the datastore-specific rule sets.

When you work with the components, follow these guidelines:

- Each component can include only one set of rules. All characteristics in this rule set belong to a single provider of the data services.
- If the component is referenced in the VM storage policy, you cannot delete the component. Before deleting the component, you must remove it from the storage policy or delete the storage policy.
- When you add components to the policy, you can use only one component from the same category, for example caching, per a set of rules.

Create Storage Policy Components

A storage policy component describes a single data service, such as replication, that must be provided for the virtual machine. You can define the component in advance and associate it with multiple VM storage policies. The components are reusable and interchangeable.

Procedure

- 1 Open the **New Storage Policy Component** dialog box.

| Option | Description |
|---------------------------|--|
| In the vSphere Web Client | <ol style="list-style-type: none"> a From the vSphere Web Client Home, click Policies and Profiles > VM Storage Policies. b Click the Storage Policy Component tab. |
| In the vSphere Client | <ol style="list-style-type: none"> a Click Menu > Policies and Profiles. b Under Policies and Profiles, click Storage Policy Components. |

2 Click **Create Storage Policy Component**.

3 Select the vCenter Server instance.

4 Enter a name, for example, 4-hour Replication, and a description for the policy component.
Make sure that the name does not conflict with the names of other components or storage policies.

5 Select the category of the service, for example, **Replication**.

6 Select the service provider.

7 Define rules for the selected category.

For example, if you are configuring 4-hour replication, set the Recovery Point Objective (RPO) value to 4.

For encryption based on I/O filters, set the **Allow I/O filters before encryption** parameter. Encryption provided by storage does not require this parameter.

| Option | Description |
|-----------------|---|
| False (default) | Does not allow the use of other I/O filters before the encryption filter. |
| True | Allows the use of other I/O filters before the encryption filter. Other filters, such as replication, can analyze clear text data before it is encrypted. |

8 Click **OK**.

Results

The new component appears in the list of storage policy components.

What to do next

You can add the component to the VM storage policy. If the data service that the component references is provided by the I/O filters, you add the component to the host-based rules of the storage policy. Components that reference entities other than the I/O filters, for example, a storage system, are added to the datastore-specific rule sets.

Edit or Clone Storage Policy Components

You can modify the existing storage policy components. You can also create a copy of the existing component by cloning it.

Procedure

- 1 Navigate to the storage policy component to edit or clone.

| Option | Description |
|---------------------------|---|
| In the vSphere Web Client | <ol style="list-style-type: none"> a From the vSphere Web Client Home, click Policies and Profiles > VM Storage Policies. b Click the Storage Policy Components tab. |
| In the vSphere Client | <ol style="list-style-type: none"> a Click Menu > Policies and Profiles. b Under Policies and Profiles, click Storage Policy Components. |

- 2 Select the component and click one of the following icons.

| Option | Description |
|---------------|---|
| Edit Settings | When editing, you cannot change the category of the data service and the provider. For example, if the original component references replication provided by I/O filters, these settings must remain unchanged. |
| Clone | When cloning, you can customize any settings of the original component. |

- 3 Modify appropriate values, and click **OK**.
- 4 If a VM storage policy that is assigned to a virtual machine references the policy component you edit, reapply the storage policy to the virtual machine.

| Menu Item | Description |
|----------------|---|
| Manually later | If you select this option, the compliance status for all virtual disks and virtual machine home objects associated with the storage policy changes to Out of Date. To update configuration and compliance, manually reapply the storage policy to all associated entities. See Reapply Virtual Machine Storage Policy . |
| Now | Update virtual machine and compliance status immediately after editing the storage policy. |

Storage Policies and Virtual Machines

After you define a VM storage policy, you can apply it to a virtual machine. You apply the storage policy when provisioning the virtual machine or configuring its virtual disks. Depending on its type and configuration, the policy might serve different purposes. The policy can select the most appropriate datastore for the virtual machine and enforce the required level of service. Or it can enable specific data services for the virtual machine and its disks.

If you do not specify the storage policy, the system uses a default storage policy that is associated with the datastore. If your storage requirements for the applications on the virtual machine change, you can modify the storage policy that was originally applied to the virtual machine.

Assign Storage Policies to Virtual Machines

You can assign a VM storage policy in an initial deployment of a virtual machine or when you perform other virtual machine operations, such as cloning or migrating.

This topic describes how to assign the VM storage policy when you create a virtual machine. For information about other deployment methods that include cloning, deployment from a template, and so on, see the *vSphere Virtual Machine Administration* documentation.

You can apply the same storage policy to the virtual machine configuration file and all its virtual disks. If storage requirements for your virtual disks and the configuration file are different, you can associate different storage policies with the VM configuration file and the selected virtual disks.

Procedure

- 1 Start the virtual machine provisioning process and follow the appropriate steps.
- 2 Assign the same storage policy to all virtual machine files and disks.
 - a On the **Select storage** page, select a storage policy from the **VM Storage Policy** drop-down menu.

Based on its configuration, the storage policy separates all datastores into compatible and incompatible. If the policy references data services offered by a specific storage entity, for example, Virtual Volumes, the compatible list includes datastores that represent only that type of storage.

- b Select an appropriate datastore from the list of compatible datastores.

The datastore becomes the destination storage resource for the virtual machine configuration file and all virtual disks.

- c If you use the replication service with Virtual Volumes, specify the replication group.

Replication groups indicate which VMs and virtual disks must be replicated together to a target site.

| Option | Description |
|--|---|
| Preconfigured replication group | Replication groups that are configured in advance on the storage side. vCenter Server and ESXi discover the replication groups, but do not manage their life cycle. |
| Automatic replication group | Virtual Volumes creates a replication group and assigns all VM objects to this group. |

3 Change the VM storage policy for the virtual disk.

Use this option if requirements for storage placement are different for virtual disks. You can also use this option to enable I/O filter services, such as caching and replication, for your virtual disks.

- a On the **Customize hardware** page, expand the **New hard disk** pane.
- b From the **VM storage policy** drop-down menu, select the storage policy to assign to the virtual disk.
- c (Optional) Change the storage location of the virtual disk.

Use this option to store the virtual disk on a datastore other than the datastore where the VM configuration file resides.

4 Complete the virtual machine provisioning process.

Results

After you create the virtual machine, the **Summary** tab displays the assigned storage policies and their compliance status.

What to do next

If storage placement requirements for the configuration file or the virtual disks change, you can later modify the virtual policy assignment.

Change Storage Policy Assignment for Virtual Machine Files and Disks

If your storage requirements for the applications on the virtual machine change, you can edit the storage policy that was originally applied to the virtual machine.

You can edit the storage policy for a powered-off or powered-on virtual machine.

When changing the VM storage policy assignment, you can apply the same storage policy to the virtual machine configuration file and all its virtual disks. You can also associate different storage policies with the VM configuration file and the virtual disks. You might apply different policies when, for example, storage requirements for your virtual disks and the configuration file are different.

Procedure

- 1 Browse to the virtual machine.

| Option | Actions |
|---------------------------|--|
| In the vSphere Web Client | <ol style="list-style-type: none"> a From the vSphere Web Client Home, click Policies and Profiles > VM Storage Policies. b On the VM Storage Policies tab, click the storage policy you want to change. c Click the VMs tab and click Virtual Machines. You can see the list of virtual machines that use this storage policy. d Click the virtual machine whose policy you want to modify. |
| In the vSphere Client | <ol style="list-style-type: none"> a Click Menu > Policies and Profiles. b Under Policies and Profiles, click VM Storage Policies. c Click the storage policy you want to change, and click VM Compliance. You can see the list of virtual machines that use this storage policy. d Click the virtual machine whose policy you want to modify. |

- 2 Click the **Configure** tab and click **Policies**.
- 3 Click **Edit VM Storage Policies**.
- 4 Specify the VM storage policy for your virtual machine.

| Option | Actions |
|--|--|
| Apply the same storage policy to all virtual machine objects (in the vSphere Web Client) | <ol style="list-style-type: none"> a Select the policy from the VM storage policy drop-down menu. b Click Apply to all. |
| Apply different storage policies to the VM home object and virtual disks (in the vSphere Web Client) | <ol style="list-style-type: none"> a Select the object, for example, VM home. b In the VM Storage Policy column, select the policy from the drop-down menu. |
| Apply the same storage policy to all virtual machine objects (in the vSphere Client) | Select the policy from the VM storage policy drop-down menu. |
| Apply different storage policies to the VM home object and virtual disks (in the vSphere Client) | <ol style="list-style-type: none"> a Turn on the Configure per disk option. b Select the object, for example, VM home. c In the VM Storage Policy column, select the policy from the drop-down menu. |

- 5 If you use Virtual Volumes policy with replication, configure the replication group.

Replication groups indicate which VMs and virtual disks must be replicated together to a target site.

You can select a common replication group for all objects or select different replication groups for each storage object.

- 6 Click **OK** to save the VM storage policy changes.

Results

The storage policy is assigned to the virtual machine and its disks.

Check Compliance for a VM Storage Policy

You can check whether a virtual machine uses a datastore that is compatible with the storage requirements specified in the VM storage policy.

Prerequisites

Verify that the virtual machine has a storage policy that is associated with it.

Procedure

- 1 Navigate to the virtual machine.
- 2 Click the **Configure** tab and click **Policies**.
- 3 Click **Check VM Storage Policy Compliance**.

The system verifies the compliance.

- 4 View the compliance status.

| Compliance Status | Description |
|-------------------|--|
| Compliant | The datastore that the virtual machine or virtual disk uses has the storage capabilities compatible with the policy requirements. |
| Noncompliant | The datastore that the virtual machine or virtual disk uses does not have the storage capabilities compatible with the policy requirements. You can migrate the virtual machine files and virtual disks to compliant datastores. |
| Out of Date | The status indicates that the policy has been edited, but the new requirements have not been communicated to the datastore where the virtual machine objects reside. To communicate the changes, reapply the policy to the objects that are out of date. |
| Not Applicable | This storage policy references datastore capabilities that are not supported by the datastore where virtual machine resides. |

What to do next

When you cannot bring the noncompliant datastore into compliance, migrate the files or virtual disks to a compatible datastore. See [Find Compatible Storage Resource for Noncompliant Virtual Machine](#).

If the status is Out of Date, reapply the policy to the objects. See [Reapply Virtual Machine Storage Policy](#).

Find Compatible Storage Resource for Noncompliant Virtual Machine

Determine which datastore is compatible with the storage policy that is associated with your virtual machine.

Occasionally, a storage policy that is assigned to a virtual machine can be in the noncompliant status. This status indicates that the virtual machine or its disks use datastores that are incompatible with the policy. You can migrate the virtual machine files and virtual disks to compatible datastores.

Use this task to determine which datastores satisfy the requirements of the policy.

Procedure

- 1 Verify that the storage policy for the virtual machine is in the noncompliant state.

- a Navigate to the virtual machine.
- b Click the **Summary** tab.

The VM Storage Policy Compliance panel on the VM Storage Policies pane shows the Noncompliant status.

- 2 Navigate to the noncompliant storage policy.

| Option | Description |
|---------------------------|--|
| In the vSphere Web Client | <ol style="list-style-type: none"> a From the vSphere Web Client Home, click Policies and Profiles > VM Storage Policies. b Click the Storage Policy tab. |
| In the vSphere Client | <ol style="list-style-type: none"> a Click Menu > Policies and Profiles. b Under Policies and Profiles, click VM Storage Policies. |

- 3 Display the list of compatible datastores for the noncompliant storage policy.

| Option | Description |
|---------------------------|---|
| In the vSphere Web Client | <ol style="list-style-type: none"> a Click the storage policy. b Click the Monitor tab and click Storage Compatibility. |
| In the vSphere Client | <ol style="list-style-type: none"> a Click the storage policy. b Click Storage Compatibility. |

The list of datastores that match the requirements of the policy appears.

What to do next

You can migrate the virtual machine or its disks to one of the datastores in the list.

Reapply Virtual Machine Storage Policy

After you edit a storage policy that is already associated with a virtual machine object, you must reapply the policy. By reapplying the policy, you communicate new storage requirements to the datastore where the virtual machine object resides.

Prerequisites

The compliance status for a virtual machine is Out of Date. The status indicates that the policy has been edited, but the new requirements have not been communicated to the datastore.

Procedure

- 1 Navigate to the virtual machine.
- 2 Click the **Configure** tab and click **Policies**.
- 3 Verify that the compliance status is Out of Date.
- 4 Click **Reapply VM Storage Policy**.
- 5 Check the compliance status.

| Compliance Status | Description |
|-------------------|---|
| Compliant | The datastore that the virtual machine or virtual disk uses has the storage capabilities that the policy requires. |
| Noncompliant | <p>The datastore that the virtual machine or virtual disk uses does not have the storage capabilities that the policy requires.</p> <p>When you cannot bring the noncompliant datastore into compliance, migrate the files or virtual disks to a compatible datastore. See Find Compatible Storage Resource for Noncompliant Virtual Machine.</p> |
| Not Applicable | This storage service level references datastore capabilities that are not supported by the datastore where the virtual machine resides. |

Default Storage Policies

When you provision a virtual machine on a datastore, you must assign to the virtual machine a compatible VM storage policy. If you do not configure and explicitly assign the storage policy to the virtual machine, the system uses a default storage policy.

VMware-Provided Default Storage Policy

The generic default storage policy that ESXi provides applies to all datastores and does not include rules specific to any storage type.

In addition, ESXi offers the default storage policies for object-based datastores, vSAN or Virtual Volumes. These policies guarantee the optimum placement for the virtual machine objects within the object-based storage.

For information about the default storage policy for VVols, see [Virtual Volumes and VM Storage Policies](#).

VMFS and NFS datastores do not have specific default policies and can use the generic default policy or a custom policy you define for them.

User-Defined Default Storage Policies

You can create a VM storage policy that is compatible with vSAN or Virtual Volumes. You can then designate this policy as the default for vSAN and Virtual Volumes datastores. The user-defined default policy replaces the default storage policy that VMware provides.

Each vSAN and Virtual Volumes datastore can have only one default policy at a time. However, you can create a single storage policy with multiple placement rule sets, so that it matches multiple vSAN and Virtual Volumes datastores. You can designate this policy as the default policy for all datastores.

When the VM storage policy becomes the default policy for a datastore, you cannot delete the policy unless you disassociate it from the datastore.

Change the Default Storage Policy for a Datastore

For Virtual Volumes and vSAN datastores, VMware provides storage policies that are used as the default during the virtual machine provisioning. You can change the default storage policy for a selected Virtual Volumes or vSAN datastore.

Note Do not designate a storage policy with replication rules as a default storage policy. Otherwise, the policy prevents you from selecting replication groups.

Prerequisites

Create a storage policy that is compatible with Virtual Volumes or vSAN. You can create a policy that matches both types of storage.

Procedure

- 1 Navigate to the datastore.
- 2 Click the **Configure** tab, and click **General**.
- 3 In the Default Storage Policy pane, click **Edit**.
- 4 From the list of available storage policies, select a policy to designate as the default and click **OK**.

Results

The selected storage policy becomes the default policy for the datastore. The system assigns this policy to any virtual machine objects that you provision on the datastore when no other policy is selected.

Using Storage Providers

21

A storage provider is a software component that is offered by VMware or developed by a third party through vSphere APIs for Storage Awareness (VASA). The storage provider can also be called VASA provider. The storage providers integrate with various storage entities that include external physical storage and storage abstractions, such as vSAN and Virtual Volumes. Storage providers can also support software solutions, for example, I/O filters.

Read the following topics next:

- [About Storage Providers](#)
- [Storage Providers and Data Representation](#)
- [Storage Provider Requirements and Considerations](#)
- [Register Storage Providers](#)
- [View Storage Provider Information](#)
- [Manage Storage Providers](#)

About Storage Providers

Generally, vCenter Server and ESXi use the storage providers to obtain information about storage configuration, status, and storage data services offered in your environment. This information appears in the vSphere Client. The information helps you to make appropriate decisions about virtual machine placement, to set storage requirements, and to monitor your storage environment.

Persistence Storage Providers

Storage providers that manage arrays and storage abstractions, are called persistence storage providers. Providers that support Virtual Volumes or vSAN belong to this category. In addition to storage, persistence providers can provide other data services, such as replication.

Data Service Providers

Another category of providers is I/O filter storage providers, or data service providers. These providers offer data services that include host-based caching, compression, and encryption.

Both persistence storage and data service providers can belong to one of these categories.

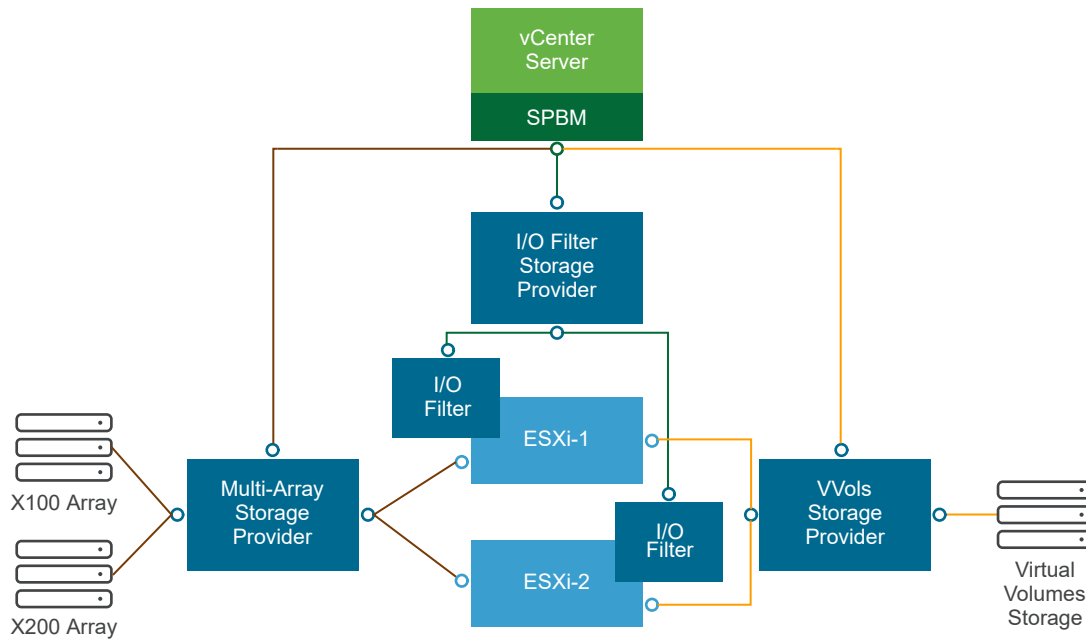
Built-in Storage Providers

Built-in storage providers are offered by VMware. Typically, they do not require registration. For example, the storage providers that support vSAN or I/O filters are built-in and become registered automatically.

Third-Party Storage Providers

When a third party offers a storage provider, you typically must register the provider. An example of such a provider is the Virtual Volumes provider. You use the vSphere Client to register and manage each storage provider component.

The following graphic illustrates how different types of storage providers facilitate communications between vCenter Server and ESXi and other components of your storage environment. For example, the components might include storage arrays, Virtual Volumes storage, and I/O filters.



Storage Providers and Data Representation

vCenter Server and ESXi communicate with the storage provider to obtain information that the storage provider collects from underlying physical and software-defined storage, or from available I/O filters. vCenter Server can then display the storage data in the vSphere Client.

Information that the storage provider supplies can be divided into the following categories:

- Storage data services and capabilities. This type of information is essential for such functionalities as vSAN, Virtual Volumes, and I/O filters. The storage provider that represents these functionalities integrates with the Storage Policy Based Management (SPBM) mechanism. The storage provider collects information about data services that are offered by underlying storage entities or available I/O filters.

You reference these data services when you define storage requirements for virtual machines and virtual disks in a storage policy. Depending on your environment, the SPBM mechanism ensures appropriate storage placement for a virtual machine or enables specific data services for virtual disks. For details, see [Creating and Managing VM Storage Policies](#).

- Storage status. This category includes reporting about status of various storage entities. It also includes alarms and events for notifying about configuration changes.

This type of information can help you troubleshoot storage connectivity and performance problems. It can also help you to correlate array-generated events and alarms to corresponding performance and load changes on the array.

- Storage DRS information for the distributed resource scheduling on block devices or file systems. This information helps to ensure that decisions made by Storage DRS are compatible with resource management decisions internal to the storage systems.

Storage Provider Requirements and Considerations

When you use the third-party storage providers, certain requirements and considerations apply.

Typically, vendors are responsible for supplying storage providers. The VMware VASA program defines an architecture that integrates third-party storage providers into the vSphere environment, so that vCenter Server and ESXi hosts can communicate with the storage providers.

To use storage providers, follow these requirements:

- Make sure that every storage provider you use is certified by VMware and properly deployed. For information about deploying the storage providers, contact your storage vendor.
- Make sure that the storage provider is compatible with the vCenter Server and ESXi versions. See *VMware Compatibility Guide*.
- Do not install the VASA provider on the same system as vCenter Server.
- If your environment contains older versions of storage providers, existing functionality continues to work. However, to use new features, upgrade your storage provider to a new version.
- When you upgrade a storage provider to a later VASA version, you must unregister and reregister the provider. After registration, vCenter Server can detect and use the functionality of the new VASA version.

Register Storage Providers

To establish a connection between vCenter Server and a storage provider, you must register the storage provider. Make sure to register a separate storage provider for each host in a cluster.

When you upgrade a storage provider to a later VASA version, you must unregister and reregister the provider. After registration, vCenter Server can detect and use the functionality of the later VASA version.

Note If you use vSAN, the storage providers for vSAN are registered and appear on the list of storage providers automatically. vSAN does not support manual registration of storage providers. See the *Administering VMware vSAN* documentation.

Prerequisites

Verify that the storage provider component is installed on the storage side and obtain its credentials from your storage administrator.

Procedure

- 1 Navigate to vCenter Server.
- 2 Click the **Configure** tab, and click **Storage Providers**.
- 3 Click the **Add** icon.
- 4 Enter connection information for the storage provider, including the name, URL, and credentials.
- 5 Specify the security method.

| Action | Description |
|---|---|
| Direct vCenter Server to the storage provider certificate | Select the Use storage provider certificate option and specify the certificate's location. |
| Use a thumbprint of the storage provider certificate | If you do not guide vCenter Server to the provider certificate, the certificate thumbprint is displayed. You can check the thumbprint and approve it. vCenter Server adds the certificate to the truststore and proceeds with the connection. |

The storage provider adds the vCenter Server certificate to its truststore when vCenter Server first connects to the provider.

- 6 Click **OK**.

Results

vCenter Server registers the storage provider and establishes a secure SSL connection with it.

What to do next

To troubleshoot registration of your storage provider, see the VMware Knowledge Base article <https://kb.vmware.com/s/article/49798>.

View Storage Provider Information

After you register a storage provider component with vCenter Server, the storage provider appears on the storage providers list. Certain storage providers are self-registered and automatically appear on the list after you set up the entity they represent, for example, vSAN or I/O filters.

View general storage provider information and details for each storage component.

Procedure

- 1 Navigate to vCenter Server.
- 2 Click the **Configure** tab, and click **Storage Providers**.
- 3 In the Storage Providers list, view the storage providers registered with vCenter Server.
The list shows general information including the name of the storage provider, its URL and status, version of VASA APIs, storage entities the provider represents, and so on.
- 4 To display additional details, select a specific storage provider or its component from the list.

Note A single storage provider can support storage systems from multiple different vendors.

Manage Storage Providers

You can perform several management operations on the registered storage providers.

Procedure

- 1 Navigate to vCenter Server.
- 2 Click the **Configure** tab, and click **Storage Providers**.
- 3 From the list of storage providers, select a storage provider and click one of the following icons.

| Option | Description |
|--------------------------------------|--|
| Synchronize Storage Providers | Synchronize all storage providers with the current state of the environment. |
| Rescan | Update storage data for the provider. vCenter Server periodically updates storage data in its database. The updates are partial and reflect only those changes that storage providers communicate to vCenter Server at that moment. When needed, you can perform a full database synchronization for the selected storage provider. |

| Option | Description |
|---------------------|---|
| Remove | <p>Unregister storage providers that you do not use. After this operation, vCenter Server closes the connection and removes the storage provider from its configuration.</p> <hr/> <p>Note You cannot manually unregister certain storage providers supplied by VMware, for example, vSAN storage providers.</p> <hr/> <p>This option is also useful when you upgrade a storage provider to a later VASA version. In this case, you must unregister and then reregister the provider. After registration, vCenter Server can detect and use the functionality of the later VASA version.</p> |
| Refresh certificate | <p>vCenter Server warns you when a certificate assigned to a storage provider is about to expire. You can refresh the certificate to continue using the provider.</p> <p>If you fail to refresh the certificate before it expires, vCenter Server discontinues using the provider.</p> |

Results

vCenter Server closes the connection and removes the storage provider from its configuration.

Virtual Volumes virtualizes SAN and NAS devices by abstracting physical hardware resources into logical pools of capacity. The Virtual Volumes functionality changes the storage management paradigm from managing space inside datastores to managing abstract storage objects handled by storage arrays.

Read the following topics next:

- [About Virtual Volumes](#)
- [Virtual Volumes Concepts](#)
- [Virtual Volumes and Storage Protocols](#)
- [Virtual Volumes Architecture](#)
- [Virtual Volumes and VMware Certificate Authority](#)
- [Snapshots and Virtual Volumes](#)
- [Before You Enable Virtual Volumes](#)
- [Configure Virtual Volumes](#)
- [Provision Virtual Machines on Virtual Volumes Datastores](#)
- [Virtual Volumes and Replication](#)
- [Best Practices for Working with vSphere Virtual Volumes](#)
- [Troubleshooting Virtual Volumes](#)

About Virtual Volumes

With Virtual Volumes, an individual virtual machine, not the datastore, becomes a unit of storage management, while storage hardware gains complete control over virtual disk content, layout, and management.

Historically, vSphere storage management used a datastore-centric approach. With this approach, storage administrators and vSphere administrators discuss in advance the underlying storage requirements for virtual machines. The storage administrator then sets up LUNs or NFS shares and presents them to ESXi hosts. The vSphere administrator creates datastores based on LUNs or NFS, and uses these datastores as virtual machine storage. Typically, the

datastore is the lowest granularity level at which data management occurs from a storage perspective. However, a single datastore contains multiple virtual machines, which might have different requirements. With the traditional approach, it is difficult to meet the requirements of an individual virtual machine.

The Virtual Volumes functionality helps to improve granularity. It helps you to differentiate virtual machine services on a per application level by offering a new approach to storage management. Rather than arranging storage around features of a storage system, Virtual Volumes arranges storage around the needs of individual virtual machines, making storage virtual-machine centric.

Virtual Volumes maps virtual disks and their derivatives, clones, snapshots, and replicas, directly to objects, called virtual volumes, on a storage system. This mapping allows vSphere to offload intensive storage operations such as snapshot, cloning, and replication to the storage system.

By creating a volume for each virtual disk, you can set policies at the optimum level. You can decide in advance what the storage requirements of an application are, and communicate these requirements to the storage system. The storage system creates an appropriate virtual disk based on these requirements. For example, if your virtual machine requires an active-active storage array, you no longer must select a datastore that supports the active-active model. Instead, you create an individual virtual volume that is automatically placed to the active-active array.

Virtual Volumes Concepts

With Virtual Volumes, abstract storage containers replace traditional storage volumes based on LUNs or NFS shares. In vCenter Server, the storage containers are represented by Virtual Volumes datastores. Virtual Volumes datastores store virtual volumes, objects that encapsulate virtual machine files.

Watch the video to learn more about different components of the Virtual Volumes functionality.



([Virtual Volumes Part 1: Concepts](#))

What to read next

- [Virtual Volumes](#)

Virtual volumes are encapsulations of virtual machine files, virtual disks, and their derivatives.

- [Virtual Volumes and Storage Providers](#)

A Virtual Volumes storage provider, also called a VASA provider, is a software component that acts as a storage awareness service for vSphere. The provider mediates out-of-band communication between vCenter Server and ESXi hosts on one side and a storage system on the other.

■ Storage Containers

Unlike traditional LUN and NFS-based storage, the Virtual Volumes functionality does not require preconfigured volumes on a storage side. Instead, Virtual Volumes uses a storage container. It is a pool of raw storage capacity or an aggregation of storage capabilities that a storage system can provide to virtual volumes.

■ Protocol Endpoints

Although storage systems manage all aspects of virtual volumes, ESXi hosts have no direct access to virtual volumes on the storage side. Instead, ESXi hosts use a logical I/O proxy, called the protocol endpoint, to communicate with virtual volumes and virtual disk files that virtual volumes encapsulate. ESXi uses protocol endpoints to establish a data path on demand from virtual machines to their respective virtual volumes.

■ Binding and Unbinding Virtual Volumes to Protocol Endpoints

At the time of creation, a virtual volume is a passive entity and is not immediately ready for I/O. To access the virtual volume, ESXi or vCenter Server send a bind request.

■ Virtual Volumes Datastores

A Virtual Volumes (VVol) datastore represents a storage container in vCenter Server and the vSphere Client.

■ Virtual Volumes and VM Storage Policies

A virtual machine that runs on a Virtual Volumes datastore requires a VM storage policy.

Virtual Volumes

Virtual volumes are encapsulations of virtual machine files, virtual disks, and their derivatives.

Virtual volumes are stored natively inside a storage system that is connected to your ESXi hosts through Ethernet or SAN. They are exported as objects by a compliant storage system and are managed entirely by hardware on the storage side. Typically, a unique GUID identifies a virtual volume. Virtual volumes are not preprovisioned, but created automatically when you perform virtual machine management operations. These operations include a VM creation, cloning, and snapshotting. ESXi and vCenter Server associate one or more virtual volumes to a virtual machine.

Types of Virtual Volumes

The system creates the following types of virtual volumes for the core elements that make up the virtual machine:

Data-VVol

A data virtual volume that corresponds directly to each virtual disk `.vmdk` file. As virtual disk files on traditional datastores, virtual volumes are presented to virtual machines as SCSI disks. Data-VVols can be either thick or thin-provisioned.

Config-VVol

A configuration virtual volume, or a home directory, represents a small directory that contains metadata files for a virtual machine. The files include a `.vmtx` file, descriptor files for virtual disks, log files, and so forth. The configuration virtual volume is formatted with a file system. When ESXi uses the SCSI protocol to connect to storage, configuration virtual volumes are formatted with VMFS. With NFS protocol, configuration virtual volumes are presented as an NFS directory. Typically, it is thin-provisioned.

Swap-VVol

Created when a VM is first powered on. It is a virtual volume to hold copies of VM memory pages that cannot be retained in memory. Its size is determined by the VM's memory size. It is thick-provisioned by default.

Snapshot-VVol

A virtual memory volume to hold the contents of virtual machine memory for a snapshot. Thick-provisioned.

Other

A virtual volume for specific features. For example, a digest virtual volume is created for Content-Based Read Cache (CBRC).

Typically, a VM creates a minimum of three virtual volumes, data-VVol, config-VVol, and swap-VVol. The maximum depends on how many virtual disks and snapshots reside on the VM.

For example, the following SQL server has six virtual volumes:

- Config-VVol
- Data-VVol for the operating system
- Data-VVol for the database
- Data-VVol for the log
- Swap-VVol when powered on
- Snapshot-VVol

By using different virtual volumes for different VM components, you can apply and manipulate storage policies at the finest granularity level. For example, a virtual volume that contains a virtual disk can have a richer set of services than the virtual volume for the VM boot disk. Similarly, a snapshot virtual volume can use a different storage tier compared to a current virtual volume.

Virtual Volumes and Disk Provisioning

The Virtual Volumes functionality supports a concept of thin and thick-provisioned virtual disks. However, from the I/O prospective, implementation and management of thin or thick provisioning by the arrays is transparent to the ESXi host. ESXi offloads to the storage arrays any functions related to thin provisioning. In the data path, ESXi does not treat the thin or thick virtual volumes differently.

You select the thin or thick type for your virtual disk at the VM creation time. If your disk is thin and resides on a VVols datastore, you cannot change its type later by inflating the disk.

Virtual Volumes and MSCS Support

You can place a shared disk on a Virtual Volumes storage that supports SCSI Persistent Reservations for VVols. You can use this disk as a quorum disk and eliminate RDMs in the MSCS clusters. For more information, see the *vSphere Resource Management* documentation.

Virtual Volumes and Storage Providers

A Virtual Volumes storage provider, also called a VASA provider, is a software component that acts as a storage awareness service for vSphere. The provider mediates out-of-band communication between vCenter Server and ESXi hosts on one side and a storage system on the other.

The storage provider is implemented through VMware APIs for Storage Awareness (VASA) and is used to manage all aspects of Virtual Volumes storage. The storage provider integrates with the Storage Monitoring Service (SMS), shipped with vSphere, to communicate with vCenter Server and ESXi hosts.

The storage provider delivers information from the underlying storage container. The storage container capabilities appear in vCenter Server and the vSphere Client. Then, in turn, the storage provider communicates virtual machine storage requirements, which you can define in the form of a storage policy, to the storage layer. This integration process ensures that a virtual volume created in the storage layer meets the requirements outlined in the policy.

Typically, vendors are responsible for supplying storage providers that can integrate with vSphere and provide support to Virtual Volumes. Every storage provider must be certified by VMware and properly deployed. For information about deploying and upgrading the Virtual Volumes storage provider to a version compatible with current ESXi release, contact your storage vendor.

After you deploy the storage provider, you must register it in vCenter Server, so that it can communicate with vSphere through the SMS.

Storage Containers

Unlike traditional LUN and NFS-based storage, the Virtual Volumes functionality does not require preconfigured volumes on a storage side. Instead, Virtual Volumes uses a storage container. It is a pool of raw storage capacity or an aggregation of storage capabilities that a storage system can provide to virtual volumes.

A storage container is a part of the logical storage fabric and is a logical unit of the underlying hardware. The storage container logically groups virtual volumes based on management and administrative needs. For example, the storage container can contain all virtual volumes created for a tenant in a multitenant deployment, or a department in an enterprise deployment. Each storage container serves as a virtual volume store and virtual volumes are allocated out of the storage container capacity.

Typically, a storage administrator on the storage side defines storage containers. The number of storage containers, their capacity, and their size depend on a vendor-specific implementation. At least one container for each storage system is required.

Note A single storage container cannot span different physical arrays.

After you register a storage provider associated with the storage system, vCenter Server discovers all configured storage containers along with their storage capability profiles, protocol endpoints, and other attributes. A single storage container can export multiple capability profiles. As a result, virtual machines with diverse needs and different storage policy settings can be a part of the same storage container.

Initially, all discovered storage containers are not connected to any specific host, and you cannot see them in the vSphere Client. To mount a storage container, you must map it to a Virtual Volumes datastore.

Protocol Endpoints

Although storage systems manage all aspects of virtual volumes, ESXi hosts have no direct access to virtual volumes on the storage side. Instead, ESXi hosts use a logical I/O proxy, called the protocol endpoint, to communicate with virtual volumes and virtual disk files that virtual volumes encapsulate. ESXi uses protocol endpoints to establish a data path on demand from virtual machines to their respective virtual volumes.

Each virtual volume is bound to a specific protocol endpoint. When a virtual machine on the host performs an I/O operation, the protocol endpoint directs the I/O to the appropriate virtual volume. Typically, a storage system requires just a few protocol endpoints. A single protocol endpoint can connect to hundreds or thousands of virtual volumes.

On the storage side, a storage administrator configures protocol endpoints, one or several per storage container. The protocol endpoints are a part of the physical storage fabric. The storage system exports the protocol endpoints with associated storage containers through the storage provider. After you map the storage container to a Virtual Volumes datastore, the ESXi host discovers the protocol endpoints and they become visible in the vSphere Client. The protocol endpoints can also be discovered during a storage rescan. Multiple hosts can discover and mount the protocol endpoints.

In the vSphere Client, the list of available protocol endpoints looks similar to the host storage devices list. Different storage transports can be used to expose the protocol endpoints to ESXi. When the SCSI-based transport is used, the protocol endpoint represents a proxy LUN defined by a T10-based LUN WWN. For the NFS protocol, the protocol endpoint is a mount point, such as an IP address and a share name. You can configure multipathing on the SCSI-based protocol endpoint, but not on the NFS-based protocol endpoint. No matter which protocol you use, the storage array can provide multiple protocol endpoints for availability purposes.

Protocol endpoints are managed per array. ESXi and vCenter Server assume that all protocol endpoints reported for an array are associated with all containers on that array. For example, if an array has two containers and three protocol endpoints, ESXi assumes that virtual volumes on both containers can be bound to all three protocol points.

Binding and Unbinding Virtual Volumes to Protocol Endpoints

At the time of creation, a virtual volume is a passive entity and is not immediately ready for I/O. To access the virtual volume, ESXi or vCenter Server send a bind request.

The storage system replies with a protocol endpoint ID that becomes an access point to the virtual volume. The protocol endpoint accepts all I/O requests to the virtual volume. This binding exists until ESXi sends an unbind request for the virtual volume.

For later bind requests on the same virtual volume, the storage system can return different protocol endpoint IDs.

When receiving concurrent bind requests to a virtual volume from multiple ESXi hosts, the storage system can return the same or different endpoint bindings to each requesting ESXi host. In other words, the storage system can bind different concurrent hosts to the same virtual volume through different endpoints.

The unbind operation removes the I/O access point for the virtual volume. The storage system might unbind the virtual volume from its protocol endpoint immediately, or after a delay, or take some other action. A bound virtual volume cannot be deleted until it is unbound.

Virtual Volumes Datastores

A Virtual Volumes (VVol) datastore represents a storage container in vCenter Server and the vSphere Client.

After vCenter Server discovers storage containers exported by storage systems, you must mount them as Virtual Volumes datastores. The Virtual Volumes datastores are not formatted in a traditional way like, for example, VMFS datastores. You must still create them because all vSphere functionalities, including FT, HA, DRS, and so on, require the datastore construct to function properly.

You use the datastore creation wizard in the vSphere Client to map a storage container to a Virtual Volumes datastore. The Virtual Volumes datastore that you create corresponds directly to the specific storage container.

From a vSphere administrator prospective, the Virtual Volumes datastore is similar to any other datastore and is used to hold virtual machines. Like other datastores, the Virtual Volumes datastore can be browsed and lists virtual volumes by virtual machine name. Like traditional datastores, the Virtual Volumes datastore supports unmounting and mounting. However, such operations as upgrade and resize are not applicable to the Virtual Volumes datastore. The Virtual Volumes datastore capacity is configurable by the storage administrator outside of vSphere.

You can use the Virtual Volumes datastores with traditional VMFS and NFS datastores and with vSAN.

Note The size of a virtual volume must be a multiple of 1 MB, with a minimum size of 1 MB. As a result, all virtual disks that you provision on a Virtual Volumes datastore must be an even multiple of 1 MB. If the virtual disk you migrate to the Virtual Volumes datastore is not an even multiple of 1 MB, extend the disk to the nearest even multiple of 1 MB.

Virtual Volumes and VM Storage Policies

A virtual machine that runs on a Virtual Volumes datastore requires a VM storage policy.

A VM storage policy is a set of rules that contains placement and quality-of-service requirements for a virtual machine. The policy enforces appropriate placement of the virtual machine within Virtual Volumes storage and guarantees that storage can satisfy virtual machine requirements.

You use the VM Storage Policies interface to create a Virtual Volumes storage policy. When you assign the new policy to the virtual machine, the policy enforces that the Virtual Volumes storage meets the requirements.

Virtual Volumes Default Storage Policy

For Virtual Volumes, VMware provides a default storage policy that contains no rules or storage requirements, called VVol No Requirements Policy. This policy is applied to the VM objects when you do not specify another policy for the virtual machine on the Virtual Volumes datastore. With the No Requirements policy, storage arrays can determine the optimum placement for the VM objects.

The default No Requirements policy that VMware provides has the following characteristics:

- You cannot delete, edit, or clone this policy.
- The policy is compatible only with the Virtual Volumes datastores.
- You can create a VM storage policy for Virtual Volumes and designate it as the default.

Virtual Volumes and Storage Protocols

A virtual volumes-based storage system provides protocol endpoints that are discoverable on the physical storage fabric. ESXi hosts use the protocol endpoints to connect to virtual volumes on the storage. Operation of the protocol endpoints depends on storage protocols that expose the endpoints to ESXi hosts.

Virtual Volumes supports NFS version 3 and 4.1, iSCSI, Fibre Channel, and FCoE.

No matter which storage protocol is used, protocol endpoints provide uniform access to both SAN and NAS storage. A virtual volume, like a file on other traditional datastore, is presented to a virtual machine as a SCSI disk.

Note A storage container is dedicated to SCSI or NAS and cannot be shared across those protocol types. An array can present one storage container with SCSI protocol endpoints and a different container with NFS protocol endpoints. The container cannot use a combination of SCSI and NFS protocol endpoints.

Virtual Volumes and SCSI-Based Transports

On disk arrays, virtual volumes support Fibre Channel, FCoE, and iSCSI protocols.

When the SCSI-based protocol is used, the protocol endpoint represents a proxy LUN defined by a T10-based LUN WWN.

As any block-based LUNs, the protocol endpoints are discovered using standard LUN discovery commands. The ESXi host periodically rescans for new devices and asynchronously discovers block-based protocol endpoints. The protocol endpoint can be accessible by multiple paths. Traffic on these paths follows well-known path selection policies, as is typical for LUNs.

On SCSI-based disk arrays at VM creation time, ESXi makes a virtual volume and formats it as VMFS. This small virtual volume stores all VM metadata files and is called the config-VVol. The config-VVol functions as a VM storage locator for vSphere.

Virtual volumes on disk arrays support the same set of SCSI commands as VMFS and use ATS as a locking mechanism.

Virtual Volumes and NFS Transports

With NAS storage, a protocol endpoint is an NFS share that the ESXi host mounts using IP address or DNS name and a share name. Virtual Volumes supports NFS version 3 and 4.1 to access NAS storage. Both IPv4 and IPv6 formats are supported.

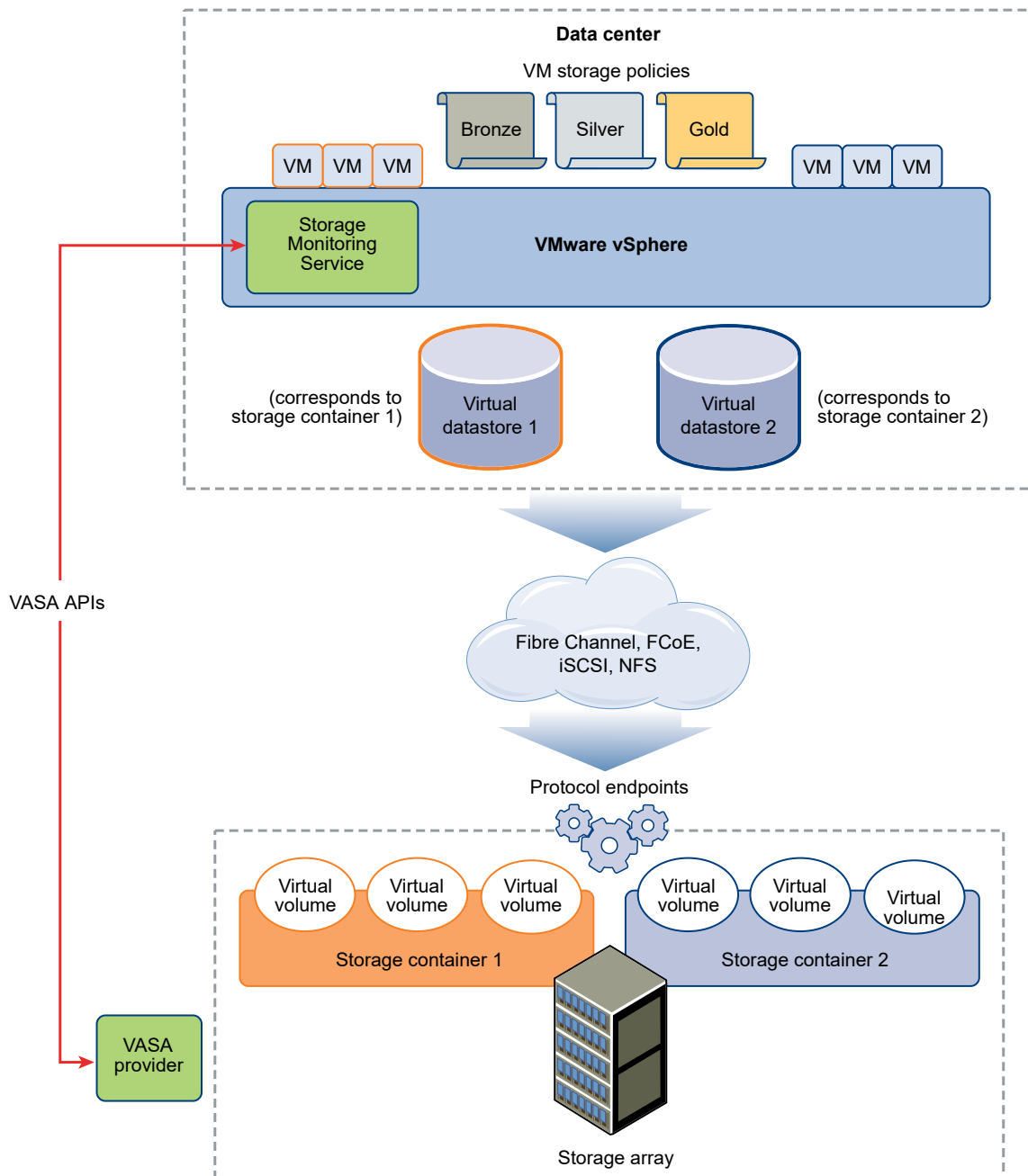
No matter which version you use, a storage array can provide multiple protocol endpoints for availability purposes.

Virtual volumes on NAS devices support the same NFS Remote Procedure Calls (RPCs) that ESXi hosts use when connecting to NFS mount points.

On NAS devices, a config-VVol is a directory subtree that corresponds to a config-VVolID. The config-VVol must support directories and other operations that are necessary for NFS.

Virtual Volumes Architecture

An architectural diagram provides an overview of how all components of the Virtual Volumes functionality interact with each other.



Virtual volumes are objects exported by a compliant storage system and typically correspond one-to-one with a virtual machine disk and other VM-related files. A virtual volume is created and manipulated out-of-band, not in the data path, by a VASA provider.

A VASA provider, or a storage provider, is developed through vSphere APIs for Storage Awareness. The storage provider enables communication between the ESXi hosts, vCenter Server, and the vSphere Client on one side, and the storage system on the other. The VASA provider runs on the storage side and integrates with the vSphere Storage Monitoring Service (SMS) to manage all aspects of Virtual Volumes storage. The VASA provider maps virtual disk objects and their derivatives, such as clones, snapshots, and replicas, directly to the virtual volumes on the storage system.

The ESXi hosts have no direct access to the virtual volumes storage. Instead, the hosts access the virtual volumes through an intermediate point in the data path, called the protocol endpoint. The protocol endpoints establish a data path on demand from the virtual machines to their respective virtual volumes. The protocol endpoints serve as a gateway for direct in-band I/O between ESXi hosts and the storage system. ESXi can use Fibre Channel, FCoE, iSCSI, and NFS protocols for in-band communication.

The virtual volumes reside inside storage containers that logically represent a pool of physical disks on the storage system. On the vCenter Server and ESXi side, storage containers are presented as Virtual Volumes datastores. A single storage container can export multiple storage capability sets and provide different levels of service to different virtual volumes.

Watch the video for information about Virtual Volumes architecture.



([Virtual Volumes Part 2: Architecture](#))

Virtual Volumes and VMware Certificate Authority

vSphere includes the VMware Certificate Authority (VMCA). By default, the VMCA creates all internal certificates used in vSphere environment. It generates certificates for newly added ESXi hosts and storage VASA providers that manage or represent Virtual Volumes storage systems.

Communication with the VASA provider is protected by SSL certificates. These certificates can come from the VASA provider or from the VMCA.

- Certificates can be directly provided by the VASA provider for long-term use. They can be either self-generated and self-signed, or derived from an external Certificate Authority.
- Certificates can be generated by the VMCA for use by the VASA provider.

When a host or VASA provider is registered, VMCA follows these steps automatically, without involvement from the vSphere administrator.

- 1 When a VASA provider is first added to the vCenter Server storage management service (SMS), it produces a self-signed certificate.
- 2 After verifying the certificate, the SMS requests a Certificate Signing Request (CSR) from the VASA provider.
- 3 After receiving and validating the CSR, the SMS presents it to the VMCA on behalf of the VASA provider, requesting a CA signed certificate.

The VMCA can be configured to function as a standalone CA, or as a subordinate to an enterprise CA. If you set up the VMCA as a subordinate CA, the VMCA signs the CSR with the full chain.

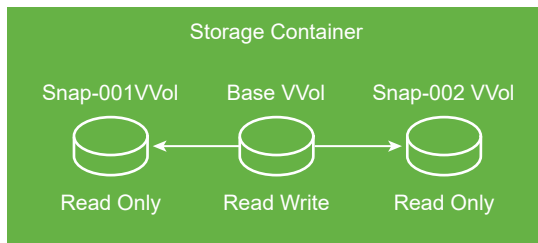
- 4 The signed certificate with the root certificate is passed to the VASA provider. The VASA provider can authenticate all future secure connections originating from the SMS on vCenter Server and on ESXi hosts.

Snapshots and Virtual Volumes

Snapshots preserve the state and data of a virtual machine at the time you take the snapshot. Snapshots are useful when you must revert repeatedly to the same virtual machine state, but you do not want to create multiple virtual machines. Virtual volumes snapshots serve many purposes. You can use them to create a quiesced copy for backup or archival purposes, or to create a test and rollback environment for applications. You can also use them to provision application images instantly.

In Virtual Volumes environment, snapshots are managed by ESXi and vCenter Server, but are performed by the storage array.

Each snapshot creates an extra virtual volume object, snapshot, or memory, virtual volume, that holds the contents of virtual machine memory. Original VM data is copied to this object, and it remains read-only, which prevents the guest operating system from writing to snapshot. You cannot resize the snapshot virtual volume. And it can be read only when the VM is reverted to a snapshot. Typically, when you replicate the VM, its snapshot virtual volume is also replicated.



The base virtual volume remains active, or read-write. When another snapshot is created, it preserves the new state and data of the virtual machine at the time you take the snapshot.

Deleting snapshots leaves the base virtual volume that represents the most current state of the virtual machine. Snapshot virtual volumes are discarded. Unlike snapshots on the traditional datastores, virtual volumes snapshots do not need to commit their contents to the base virtual volume.



For information about creating and managing snapshots, see the *vSphere Virtual Machine Administration* documentation.

Before You Enable Virtual Volumes

To work with Virtual Volumes, you must make sure that your storage and vSphere environment are set up correctly.

Prepare Storage System for Virtual Volumes

To prepare your storage system environment for Virtual Volumes, follow these guidelines. For additional information, contact your storage vendor.

- The storage system or storage array that you use must support Virtual Volumes and integrate with the vSphere components through vSphere APIs for Storage Awareness (VASA). The storage array must support thin provisioning and snapshotting.
- The Virtual Volumes storage provider must be deployed.
- The following components must be configured on the storage side:
 - Protocol endpoints
 - Storage containers
 - Storage profiles
 - Replication configurations if you plan to use Virtual Volumes with replication. See [Requirements for Replication with Virtual Volumes](#).

Prepare vSphere Environment

- Make sure to follow appropriate setup guidelines for the type of storage you use, Fibre Channel, FCoE, iSCSI, or NFS. If necessary, install and configure storage adapters on your ESXi hosts.
 - If you use iSCSI, activate the software iSCSI adapters on your ESXi hosts. Configure Dynamic Discovery and enter the IP address of your Virtual Volumes storage system. See [Configure the Software iSCSI Adapter](#).
- Synchronize all components in the storage array with vCenter Server and all ESXi hosts. Use Network Time Protocol (NTP) to do this synchronization.

For more information, contact your vendor and see *VMware Compatibility Guide*

Synchronize vSphere Storage Environment with a Network Time Server

If you use Virtual Volumes, configure Network Time Protocol (NTP) to make sure all ESXi hosts on the vSphere network are synchronized.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **System**, select **Time Configuration**.

- 4 Click **Edit** and set up the NTP server.
 - a Select **Use Network Time Protocol (Enable NTP client)**.
 - b Set the NTP Service Startup Policy.
 - c Enter the IP addresses of the NTP server to synchronize with.
 - d Click **Start** or **Restart** in the NTP Service Status section.
- 5 Click **OK**.
The host synchronizes with the NTP server.

Configure Virtual Volumes

To configure your Virtual Volumes environment, follow several steps.

Prerequisites

Follow guidelines in [Before You Enable Virtual Volumes](#).

Procedure

1 [Register Storage Providers for Virtual Volumes](#)

Your Virtual Volumes environment must include storage providers, also called VASA providers. Typically, third-party vendors develop storage providers through the VMware APIs for Storage Awareness (VASA). Storage providers facilitate communication between vSphere and the storage side. You must register the storage provider in vCenter Server to be able to work with Virtual Volumes.

2 [Create a Virtual Volumes Datastore](#)

You use the **New Datastore** wizard to create a Virtual Volumes datastore.

3 [Review and Manage Protocol Endpoints](#)

ESXi hosts use a logical I/O proxy, called protocol endpoint, to communicate with virtual volumes and virtual disk files that virtual volumes encapsulate. Protocol endpoints are exported, along with associated storage containers, by the storage system through a storage provider. Protocol endpoints become visible in the vSphere Client after you map a storage container to a Virtual Volumes datastore. You can review properties of protocol endpoints and modify specific settings.

4 [\(Optional\) Change the Path Selection Policy for a Protocol Endpoint](#)

If your ESXi host uses SCSI-based transport to communicate with protocol endpoints representing a storage array, you can modify default multipathing policies assigned to protocol endpoints. Use the **Edit Multipathing Policies** dialog box to change a path selection policy.

What to do next

You can now provision virtual machines on the Virtual Volumes datastore. For information on creating virtual machines, see [Provision Virtual Machines on Virtual Volumes Datastores](#) and the *vSphere Virtual Machine Administration* documentation.

Register Storage Providers for Virtual Volumes

Your Virtual Volumes environment must include storage providers, also called VASA providers. Typically, third-party vendors develop storage providers through the VMware APIs for Storage Awareness (VASA). Storage providers facilitate communication between vSphere and the storage side. You must register the storage provider in vCenter Server to be able to work with Virtual Volumes.

After registration, the Virtual Volumes provider communicates with vCenter Server. The provider reports characteristics of underlying storage and data services, such as replication, that the storage system provides. The characteristics appear in the VM Storage Policies interface and can be used to create a VM storage policy compatible with the Virtual Volumes datastore. After you apply this storage policy to a virtual machine, the policy is pushed to Virtual Volumes storage. The policy enforces optimal placement of the virtual machine within Virtual Volumes storage and guarantees that storage can satisfy virtual machine requirements. If your storage provides extra services, such as caching or replication, the policy enables these services for the virtual machine.

Prerequisites

Verify that an appropriate version of the Virtual Volumes storage provider is installed on the storage side. Obtain credentials of the storage provider.

Procedure

- 1 Navigate to vCenter Server.
- 2 Click the **Configure** tab, and click **Storage Providers**.
- 3 Click the **Add** icon.
- 4 Enter connection information for the storage provider, including the name, URL, and credentials.
- 5 Specify the security method.

| Action | Description |
|--|---|
| Direct vCenter Server to the storage provider certificate | Select the Use storage provider certificate option and specify the certificate's location. |
| Use a thumbprint of the storage provider certificate | If you do not guide vCenter Server to the provider certificate, the certificate thumbprint is displayed. You can check the thumbprint and approve it. vCenter Server adds the certificate to the truststore and proceeds with the connection. |

The storage provider adds the vCenter Server certificate to its truststore when vCenter Server first connects to the provider.

- 6 To complete the registration, click **OK**.

Results

vCenter Server discovers and registers the Virtual Volumes storage provider.

Create a Virtual Volumes Datastore

You use the **New Datastore** wizard to create a Virtual Volumes datastore.

Procedure

- 1 In the vSphere Client object navigator, browse to a host, a cluster, or a data center.
- 2 From the right-click menu, select **Storage > New Datastore**.
- 3 Select **VVOL** as the datastore type.
- 4 Enter the datastore name and select a backing storage container from the list of storage containers.

Make sure to use the name that does not duplicate another datastore name in your data center environment.

If you mount the same Virtual Volumes datastore to several hosts, the name of the datastore must be consistent across all hosts.

- 5 Select the hosts that require access to the datastore.
- 6 Review the configuration options and click **Finish**.

What to do next

After you create the Virtual Volumes datastore, you can perform such datastore operations as renaming the datastore, browsing datastore files, unmounting the datastore, and so on.

You cannot add the Virtual Volumes datastore to a datastore cluster.

Review and Manage Protocol Endpoints

ESXi hosts use a logical I/O proxy, called protocol endpoint, to communicate with virtual volumes and virtual disk files that virtual volumes encapsulate. Protocol endpoints are exported, along with associated storage containers, by the storage system through a storage provider. Protocol endpoints become visible in the vSphere Client after you map a storage container to a Virtual Volumes datastore. You can review properties of protocol endpoints and modify specific settings.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Protocol Endpoints**.

- 4 To view details for a specific item, select this item from the list.
- 5 Use tabs under Protocol Endpoint Details to access additional information and modify properties for the selected protocol endpoint.

| Tab | Description |
|---|---|
| Properties | View the item properties and characteristics. For SCSI (block) items, view and edit multipathing policies. |
| Paths (SCSI protocol endpoints only) | Display paths available for the protocol endpoint. Disable or enable a selected path. Change the Path Selection Policy. |
| Datastores | Display a corresponding Virtual Volumes datastore. Perform datastore management operations. |

Change the Path Selection Policy for a Protocol Endpoint

If your ESXi host uses SCSI-based transport to communicate with protocol endpoints representing a storage array, you can modify default multipathing policies assigned to protocol endpoints. Use the **Edit Multipathing Policies** dialog box to change a path selection policy.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **Storage**, click **Protocol Endpoints**.
- 4 Select the protocol endpoint whose paths you want to change and click the **Properties** tab.
- 5 Under Multipathing Policies, click **Edit Multipathing**.
- 6 Select a path policy.

The path policies available for your selection depend on the storage vendor support.

- Fixed (VMware)
- Most Recently Used (VMware)
- Round Robin (VMware)

- 7 For the fixed policy, select the preferred path from the list of available paths.
- 8 To save your settings and exit the dialog box, click **OK**.

Provision Virtual Machines on Virtual Volumes Datastores

You can provision virtual machines on a Virtual Volumes datastore.

Note All virtual disks that you provision on a Virtual Volumes datastore must be an even multiple of 1 MB.

A virtual machine that runs on a Virtual Volumes datastore requires an appropriate VM storage policy.

After you provision the virtual machine, you can perform typical VM management tasks. For information, see the *vSphere Virtual Machine Administration* documentation.

Procedure

- 1 Define a VM storage policy for Virtual Volumes.

VMware provides a default No Requirements storage policy for Virtual Volumes. If you need, you can create a custom storage policy compatible with Virtual Volumes.

See [Create a VM Storage Policy for Virtual Volumes](#).

- 2 Assign the Virtual Volumes storage policy to the virtual machine.

To guarantee that the Virtual Volumes datastore fulfills specific storage requirements when allocating a virtual machine, associate the Virtual Volumes storage policy with the virtual machine.

See [Assign Storage Policies to Virtual Machines](#).

- 3 Change default storage policy for a Virtual Volumes datastore.

For virtual machines provisioned on Virtual Volumes datastores, VMware provides a default No Requirements policy. You cannot edit this policy, but you can designate a newly created policy as default.

See [Change the Default Storage Policy for a Datastore](#).

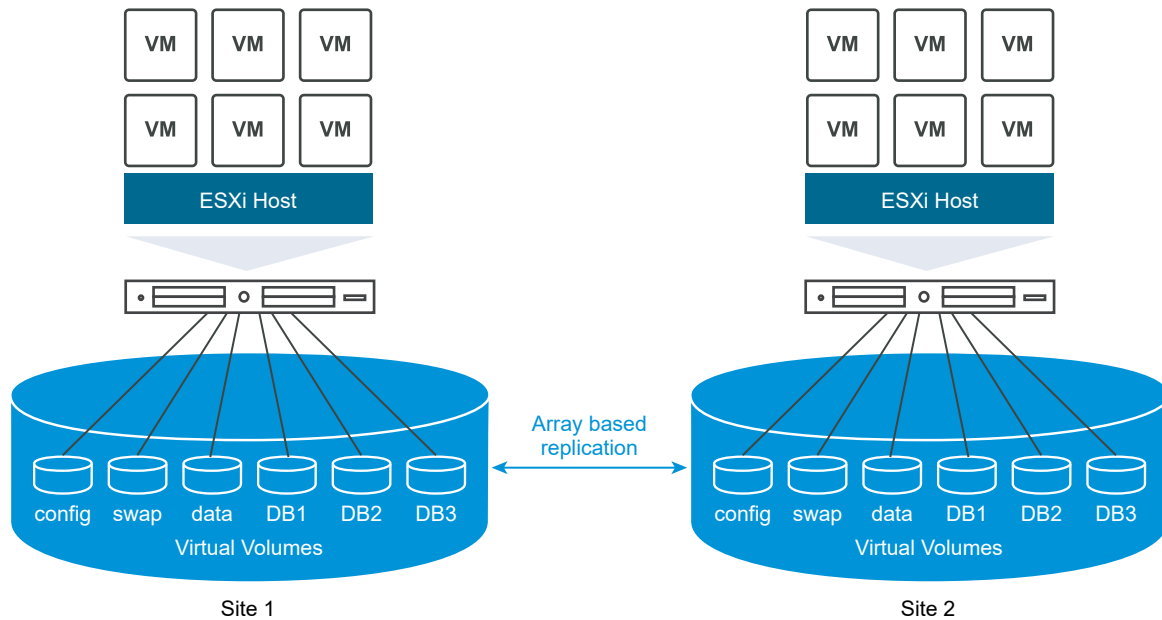
Virtual Volumes and Replication

Virtual Volumes supports replication and disaster recovery. With the array-based replication, you can off-load replication of virtual machines to your storage array and use full replication capabilities of the array. You can replicate a single VM object, such as a virtual disk. You can also group several VM objects or virtual machines to replicate them as a single unit.

Array-based replication is policy driven. After you configure your Virtual Volumes storage for replication, information about replication capabilities and replication groups is delivered from the array by the storage provider. This information shows in the VM Storage Policy interface of vCenter Server.

You use the VM storage policy to describe replication requirements for your virtual machines. The parameters that you specify in the storage policy depend on how your array implements replication. For example, your VM storage policy might include such parameters as the replication schedule, replication frequency, or recovery point objective (RPO). The policy might also indicate the replication target, a secondary site where your virtual machines are replicated, or specify whether replicas must be deleted.

By assigning the replication policy during VM provisioning, you request replication services for your virtual machine. After that, the array takes over the management of all replication schedules and processes.



Requirements for Replication with Virtual Volumes

When you enable Virtual Volumes with replication, in addition to general Virtual Volumes requirements, your environment must satisfy several specific prerequisites.

For general Virtual Volumes requirements, see [Before You Enable Virtual Volumes](#).

Storage Requirements

Implementation of Virtual Volumes replication depends on your array and might be different for storage vendors. Generally, the following requirements apply to all vendors.

- The storage arrays that you use to implement replication must be compatible with Virtual Volumes.
- The arrays must integrate with the version of the storage (VASA) provider compatible with Virtual Volumes replication.
- The storage arrays must be replication capable and configured to use vendor-provided replication mechanisms. Typical configurations usually involve one or two replication targets. Any required configurations, such as pairing of the replicated site and the target site, must be also performed on the storage side.
- When applicable, replication groups and fault domains for Virtual Volumes must be preconfigured on the storage side.

For more information, contact your vendor and see *VMware Compatibility Guide*.

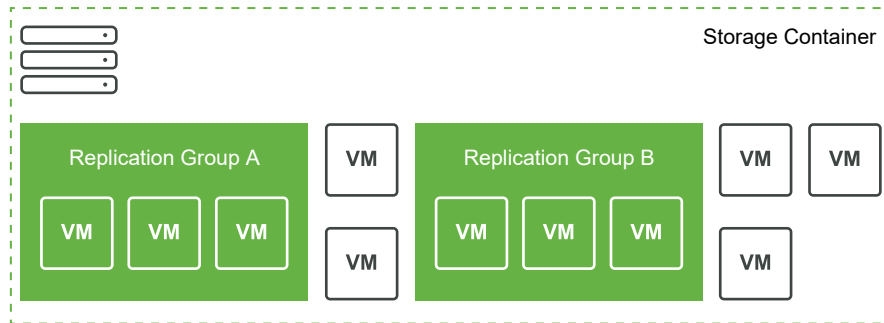
vSphere Requirements

- Use the vCenter Server and ESXi versions that support Virtual Volumes storage replication. vCenter Server and ESXi hosts that are older than 6.5 release do not support replicated Virtual Volumes storage. Any attempts to create a replicated VM on an incompatible host fail with an error. For information, see *VMware Compatibility Guide*.
- If you plan to migrate a virtual machine, make sure that target resources, such as the ESXi hosts and Virtual Volumes datastores, support storage replication.

Virtual Volumes and Replication Groups

When your storage offers replication services, in addition to storage containers and protocol endpoints, your storage administrator can configure replication groups on the storage side.

vCenter Server and ESXi can discover replication groups, but do not manage their life cycle. Replication groups, also called consistency groups, indicate which VMs and virtual disks must be replicated together to a target site. A single VM cannot span multiple replication groups.



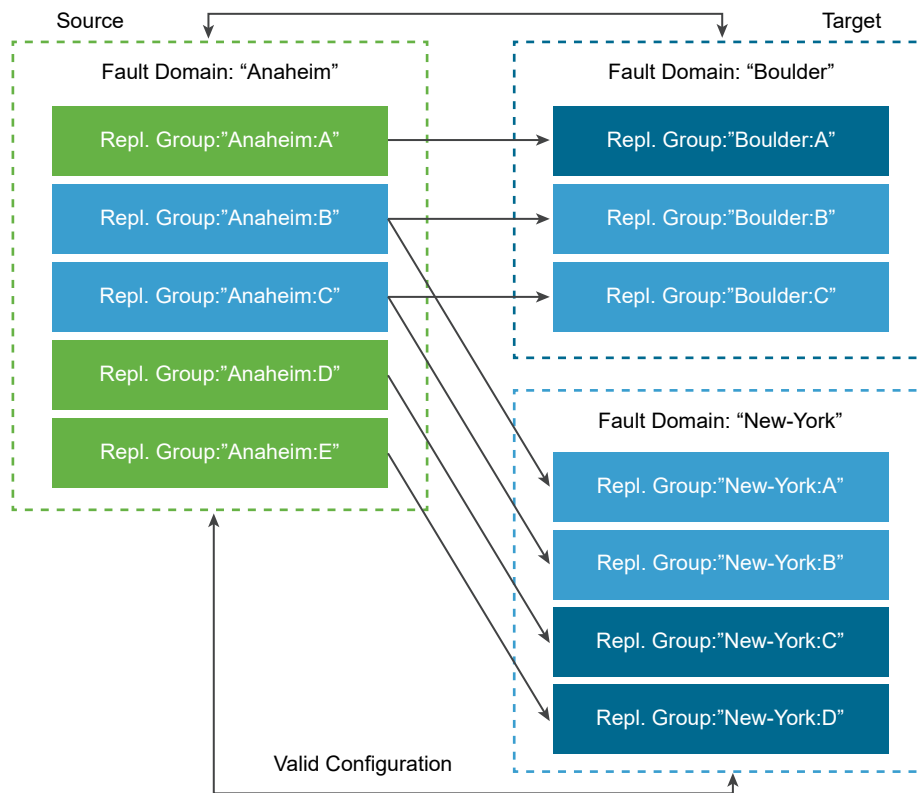
If no preconfigured groups are available, Virtual Volumes can use an automatic method. With the automatic method, Virtual Volumes creates a replication group on demand and associates this group with a Virtual Volumes object being provisioned. If you use the automatic replication group, all components of a virtual machine are assigned to the group. You cannot mix preconfigured and automatic replication groups for components of the same virtual machine.

Virtual Volumes and Fault Domains

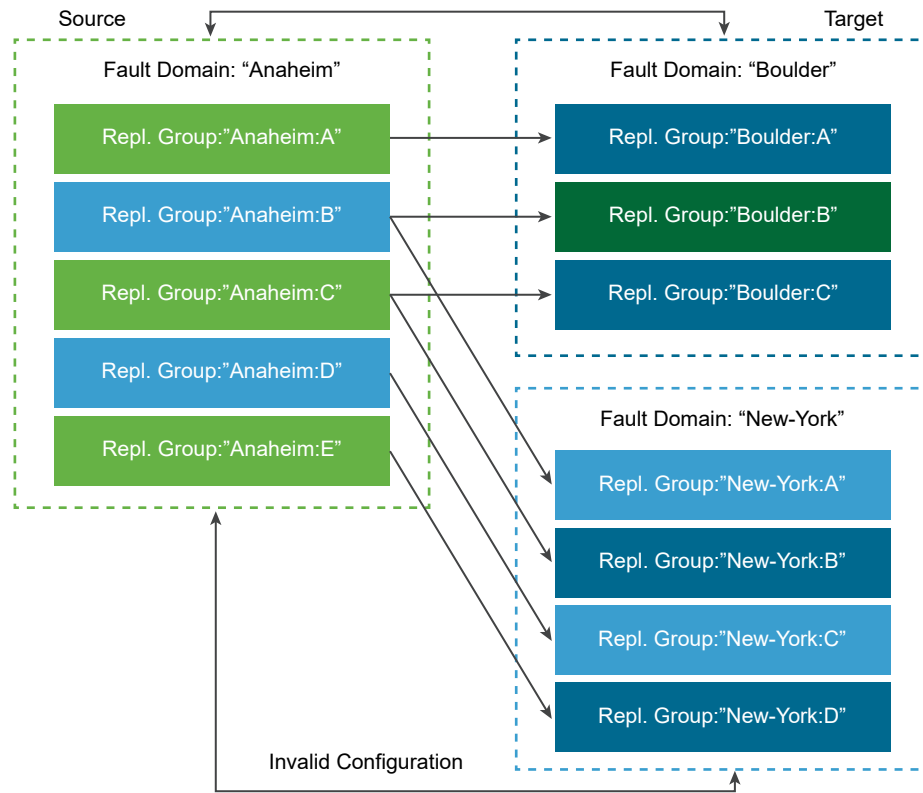
In the Virtual Volumes environment, fault domains define how specific replication groups must be combined when being replicated from a source to a target site.

Fault domains are configured and reported by the storage array, and are not exposed in the vSphere Client. The Storage Policy Based Management (SPBM) mechanism discovers fault domains and uses them for validation purposes during a virtual machine creation.

For example, provision two virtual machines, one associated with replication group Anaheim: B, the second associated with replication group Anaheim: C. SPBM validates the provisioning because each VM is in its own replication group with corresponding target fault domains.



Now provision two VMs, one associated with replication group Anaheim: B, the second associated with replication group Anaheim: D. This configuration is invalid. Both replication groups replicate to the New-York fault domain, however, only one replicates to the Boulder fault domain.



Virtual Volumes Replication Workflow

If information about replication capabilities of the Virtual Volumes storage array shows in vCenter Server, you can activate replication for your virtual machines.

The workflow to activate replication for your virtual machines includes steps typical for the virtual machine provisioning on Virtual Volumes storage.

- 1 Define the VM storage policy compatible with replication storage. The datastore-based rules of the policy must include the replication component. See [Create a VM Storage Policy for Virtual Volumes](#).

After you configure the storage policy that includes replication, vCenter Server discovers available replication groups.

- 2 Assign the replication policy to your virtual machine. If configured, select a compatible replication group, or use the automatic assignment. See [Assign Storage Policies to Virtual Machines](#).

Replication Guidelines and Considerations

When you use replication with Virtual Volumes, specific considerations apply.

- You can apply the replication storage policy only to a configuration virtual volume and a data virtual volume. Other VM objects inherit the replication policy in the following way:
 - The memory virtual volume inherits the policy of the configuration virtual volume.

- The digest virtual volume inherits the policy of the data virtual volume.
- The swap virtual volume, which exists while a virtual machine is powered on, is excluded from replication.
- If you do not apply the replication policy to a VM disk, the disk is not replicated.
- The replication storage policy should not be used as a default storage policy for a datastore. Otherwise, the policy prevents you from selecting replication groups.
- Replication preserves snapshot history. If a snapshot was created and replicated, you can recover to the application consistent snapshot.
- You can replicate a linked clone. If a linked clone is replicated without its parent, it becomes a full clone.
- If a descriptor file belongs to a virtual disk of one VM, but resides in the VM home of another VM, both VMs must be in the same replication group. If the VMs are located in different replication groups, both of these replication groups must be failed over at the same time. Otherwise, the descriptor might become unavailable after the failover. As a result, the VM might fail to power on.
- In your Virtual Volumes with replication environment, you might periodically run a test failover workflow to ensure that the recovered workloads are functional after a failover.

The resulting test VMs that are created during the test failover are fully functional and suitable for general administrative operations. However, certain considerations apply:

- All VMs created during the test failover must be deleted before the test failover stops. The deletion ensures that any snapshots or snapshot-related virtual volumes that are part of the VM, such as the snapshot virtual volume, do not interfere with stopping of the test failover.
- You can create full clones of the test VMs.
- You can create fast clones only if the policy applied to the new VM contains the same replication group ID as the VM being cloned. Attempts to place the child VM outside of the replication group of the parent VM fail.

Best Practices for Working with vSphere Virtual Volumes

Observe the following recommendations when you use Virtual Volumes with ESXi and vCenter Server.

What to read next

- [Guidelines and Limitations in Using vSphere Virtual Volumes](#)

For the best experience with vSphere Virtual Volumes functionality, you must follow specific guidelines.

- [Best Practices for Storage Container Provisioning](#)

Follow these best practices when provisioning storage containers on the vSphere Virtual Volumes array side.

- [Best Practices for vSphere Virtual Volumes Performance](#)

To ensure optimal vSphere Virtual Volumes performance results, follow these recommendations.

Guidelines and Limitations in Using vSphere Virtual Volumes

For the best experience with vSphere Virtual Volumes functionality, you must follow specific guidelines.

Virtual Volumes supports the following capabilities, features, and VMware products:

- With Virtual Volumes, you can use advanced storage services that include replication, encryption, deduplication, and compression on individual virtual disks. Contact your storage vendor for information about services they support with Virtual Volumes.
- Virtual Volumes functionality supports backup software that uses vSphere APIs - Data Protection. Virtual volumes are modeled on virtual disks. Backup products that use vSphere APIs - Data Protection are as fully supported on virtual volumes as they are on VMDK files on a LUN. Snapshots that the backup software creates using vSphere APIs - Data Protection look as non-VVols snapshots to vSphere and the backup software.

Note vSphere Virtual Volumes does not support SAN transport mode. vSphere APIs - Data Protection automatically selects an alternative data transfer method.

For more information about integration with the vSphere Storage APIs - Data Protection, consult your backup software vendor.

- Virtual Volumes supports such vSphere features as vSphere vMotion, Storage vMotion, snapshots, linked clones, Flash Read Cache, and DRS.
- You can use clustering products, such as Oracle Real Application Clusters, with Virtual Volumes. To use these products, you activate the multiwrite setting for a virtual disk stored on the VVol datastore.

For more details, see the knowledge base article at <http://kb.vmware.com/kb/2112039>. For a list of features and products that Virtual Volumes functionality supports, see *VMware Product Interoperability Matrixes*.

vSphere Virtual Volumes Limitations

Improve your experience with vSphere Virtual Volumes by knowing the following limitations:

- Because the Virtual Volumes environment requires vCenter Server, you cannot use Virtual Volumes with a standalone host.
- Virtual Volumes functionality does not support RDMS.

- A Virtual Volumes storage container cannot span multiple physical arrays. Some vendors present multiple physical arrays as a single array. In such cases, you still technically use one logical array.
- Host profiles that contain Virtual Volumes datastores are vCenter Server specific. After you extract this type of host profile, you can attach it only to hosts and clusters managed by the same vCenter Server as the reference host.

Best Practices for Storage Container Provisioning

Follow these best practices when provisioning storage containers on the vSphere Virtual Volumes array side.

Creating Containers Based on Your Limits

Because storage containers apply logical limits when grouping virtual volumes, the container must match the boundaries that you want to apply.

Examples might include a container created for a tenant in a multitenant deployment, or a container for a department in an enterprise deployment.

- Organizations or departments, for example, Human Resources and Finance
- Groups or projects, for example, Team A and Red Team
- Customers

Putting All Storage Capabilities in a Single Container

Storage containers are individual datastores. A single storage container can export multiple storage capability profiles. As a result, virtual machines with diverse needs and different storage policy settings can be a part of the same storage container.

Changing storage profiles must be an array-side operation, not a storage migration to another container.

Avoiding Over-Provisioning Your Storage Containers

When you provision a storage container, the space limits that you apply as part of the container configuration are only logical limits. Do not provision the container larger than necessary for the anticipated use. If you later increase the size of the container, you do not need to reformat or repartition it.

Using Storage-Specific Management UI to Provision Protocol Endpoints

Every storage container needs protocol endpoints (PEs) that are accessible to ESXi hosts.

When you use block storage, the PE represents a proxy LUN defined by a T10-based LUN WWN. For NFS storage, the PE is a mount point, such as an IP address or DNS name, and a share name.

Typically, configuration of PEs is array-specific. When you configure PEs, you might need to associate them with specific storage processors, or with certain hosts. To avoid errors when creating PEs, do not configure them manually. Instead, when possible, use storage-specific management tools.

No Assignment of IDs Above Disk.MaxLUN to Protocol Endpoint LUNs

By default, an ESXi host can access LUN IDs that are within the range of 0 to 1023. If the ID of the protocol endpoint LUN that you configure is 1024 or greater, the host might ignore the PE.

If your environment uses LUN IDs that are greater than 1023, change the number of scanned LUNs through the `Disk.MaxLUN` parameter. See [Change the Number of Scanned Storage Devices](#).

Best Practices for vSphere Virtual Volumes Performance

To ensure optimal vSphere Virtual Volumes performance results, follow these recommendations.

Using Different VM Storage Policies for Individual Virtual Volumes

By default, all components of a virtual machine in the Virtual Volumes environment get a single VM storage policy. However, different components might have different performance characteristics, for example, a database virtual disk and a corresponding log virtual disk. Depending on performance requirements, you can assign different VM storage policies to individual virtual disks and to the VM home file, or config-VVol.

When you use vSphere Client, you cannot change the VM storage policy assignment for swap-VVol, memory-VVol, or snapshot-VVol.

See [Create a VM Storage Policy for Virtual Volumes](#).

Getting a Host Profile with Virtual Volumes

The best way to get a host profile with Virtual Volumes is to configure a reference host and extract its profile. If you manually edit an existing host profile in the vSphere Client and attach the edited profile to a new host, you might trigger compliance errors. Other unpredictable problems might occur. For more details, see the [VMware Knowledge Base article 2146394](#).

Monitoring I/O Load on Individual Protocol Endpoint

- All virtual volume I/O goes through protocol endpoints (PEs). Arrays select protocol endpoints from several PEs that are accessible to an ESXi host. Arrays can do load balancing and change the binding path that connects the virtual volume and the PE. See [Binding and Unbinding Virtual Volumes to Protocol Endpoints](#).
- On block storage, ESXi gives a large queue depth to I/O because of a potentially high number of virtual volumes. The `Scsi.ScsiVVolPESNRO` parameter controls the number of I/O that can be queued for PEs. You can configure the parameter on the Advanced System Settings page of the vSphere Client.

Monitoring Array Limitations

A single VM might occupy multiple virtual volumes. See [Virtual Volumes](#).

Suppose that your VM has two virtual disks, and you take two snapshots with memory. Your VM might occupy up to 10 VVol objects: a config-VVol, a swap-VVol, two data-VVols, four snapshot-VVols, and two memory snapshot-VVols.

Ensuring that Storage Provider Is Available

To access vSphere Virtual Volumes storage, your ESXi host requires a storage provider (VASA provider). To ensure that the storage provider is always available, follow these guidelines:

- Do not migrate a storage provider VM to Virtual Volumes storage.
- Back up your storage provider VM.
- When appropriate, use vSphere HA or Site Recovery Manager to protect the storage provider VM.

Troubleshooting Virtual Volumes

The troubleshooting topics provide solutions to problems that you might encounter when using Virtual Volumes.

- [Virtual Volumes and esxcli Commands](#)

You can use the `esxcli storage vvol` commands to troubleshoot your Virtual Volumes environment.

- [Virtual Datastore Is Inaccessible](#)

After you create a virtual datastore, it remains inaccessible.

- [Failures When Migrating VMs or Deploying VM OVF to Virtual Volumes Datastores](#)

Your attempts to migrate a virtual machine or to deploy a VM OVF to virtual datastores fail.

- [Failed Attempts to Migrate VMs with Memory Snapshots to and from Virtual Datastores](#)

When you attempt to migrate a VM with hardware version 10 or earlier to and from a vSphere Virtual Volumes datastore, failures occur if the VM has memory snapshots.

Virtual Volumes and esxcli Commands

You can use the `esxcli storage vvol` commands to troubleshoot your Virtual Volumes environment.

The following command options are available:

Table 22-1. esxcli storage vvol commands

| Namespace | Command Option | Description |
|--------------------------------------|----------------------------|---|
| esxcli storage core device | list | Identify protocol endpoints. The output entry <code>Is VVOL PE: true</code> indicates that the storage device is a protocol endpoint. |
| esxcli storage vvol daemon | unbindall | Unbind all virtual volumes from all VASA providers known to the ESXi host. |
| esxcli storage vvol protocolendpoint | list | List all protocol endpoints that your host can access. |
| esxcli storage vvol storagecontainer | list abandonedvvol scan | List all available storage containers. Scan the specified storage container for abandoned VVols. |
| esxcli storage vvol vasacontext | get | Show the VASA context (VC UUID) associated with the host. |
| esxcli storage vvol vasaprovider | list | List all storage (VASA) providers associated with the host. |

Virtual Datastore Is Inaccessible

After you create a virtual datastore, it remains inaccessible.

Problem

The vSphere Client shows the datastore as inaccessible. You cannot use the datastore for virtual machine provisioning.

Cause

This problem might occur when you fail to configure protocol endpoints for the SCSI-based storage container that is mapped to the virtual datastore. Like traditional LUNs, SCSI protocol endpoints need to be configured so that an ESXi host can detect them.

Solution

Before creating virtual datastores for SCSI-based containers, make sure to configure protocol endpoints on the storage side.

Failures When Migrating VMs or Deploying VM OVF to Virtual Volumes Datastores

Your attempts to migrate a virtual machine or to deploy a VM OVF to virtual datastores fail.

Problem

An OVF template or a VM being migrated from a nonvirtual datastore might include additional large files, such as ISO disk images, DVD images, and image files. If these additional files cause the configuration virtual volume to exceed its 4-GB limit, migration or deployment to a virtual datastore fails.

Cause

The configuration virtual volume, or config-VVol, contains various VM-related files. On traditional nonvirtual datastores, these files are stored in the VM home directory. Similar to the VM home directory, the config-VVol typically includes the VM configuration file, virtual disk and snapshot descriptor files, log files, lock files, and so on.

On virtual datastores, all other large-sized files, such as virtual disks, memory snapshots, swap, and digest, are stored as separate virtual volumes.

Config-VVols are created as 4-GB virtual volumes. Generic content of the config-VVol usually consumes only a fraction of this 4-GB allocation, so config-VVols are typically thin-provisioned to conserve backing space. Any additional large files, such as ISO disk images, DVD images, and image files, might cause the config-VVol to exceed its 4-GB limit. If such files are included in an OVF template, deployment of the VM OVF to vSphere Virtual Volumes storage fails. If these files are part of an existing VM, migration of that VM from a traditional datastore to vSphere Virtual Volumes storage also fails.

Solution

- For VM migration. Before migrating a VM from a traditional datastore to a virtual datastore, remove excess content from the VM home directory to keep the config-VVol under the 4-GB limit.
- For OVF deployment. Because you cannot deploy an OVF template that contains excess files directly to a virtual datastore, first deploy the VM to a nonvirtual datastore. Remove any excess content from the VM home directory, and migrate the resulting VM to vSphere Virtual Volumes storage.

Failed Attempts to Migrate VMs with Memory Snapshots to and from Virtual Datastores

When you attempt to migrate a VM with hardware version 10 or earlier to and from a vSphere Virtual Volumes datastore, failures occur if the VM has memory snapshots.

Problem

The following problems occur when you migrate a version 10 or earlier VM with memory snapshots:

- Migration of a version 10 or earlier VM with memory snapshots to a virtual datastore is not supported and causes a failure.

- Migration of a version 10 or earlier VM with memory snapshots from a virtual datastore to a nonvirtual datastore, such as VMFS, can succeed. If you later make additional snapshots and attempt to migrate this VM back to vSphere Virtual Volumes storage, your attempt fails.

Cause

vSphere Virtual Volumes storage does not require that you use a particular hardware version for your virtual machines. Typically, you can move a virtual machine with any hardware version to vSphere Virtual Volumes storage. However, if you have a VM with memory snapshots, and plan to migrate this VM between a virtual datastore and a nonvirtual datastore, use the VM of hardware version 11.

Non-VVols virtual machines of hardware version 11 or later use separate files to store their memory snapshots. This usage is consistent with VMs on vSphere Virtual Volumes storage, where memory snapshots are created as separate VVols instead of being stored as part of a `.vmsn` file in the VM home directory. In contrast, non-VVols VMs with hardware version 10 continue to store their memory snapshots as part of the `.vmsn` file in the VM home directory. As a result, you might experience problems or failures when attempting to migrate these VMs between virtual and nonvirtual datastores.

Solution

To avoid problems when migrating VMs with memory snapshots across virtual and nonvirtual datastores, use hardware version 11. Follow these guidelines when migrating version 10 or earlier VMs with memory snapshots:

- Migrating a version 10 or earlier VM with memory snapshots to a virtual datastore is not supported. The only workaround is to remove all snapshots. Upgrading the hardware version does not solve this problem.
- Migrating a version 10 or earlier VM with memory snapshots from a virtual datastore to a nonvirtual datastore, such as VMFS, can succeed. However, the migration might put the VM in an inconsistent state. The snapshots that were taken on the virtual datastore use the `vmem` object. Any memory snapshots taken after migrating to VMFS are stored in the `.vmsn` file. If you later attempt to migrate this VM back to vSphere Virtual Volumes storage, your attempt fails. As with the previous case, remove all snapshots to work around this problem.

Filtering Virtual Machine I/O

23

I/O filters are software components that can be installed on ESXi hosts and can offer additional data services to virtual machines. The filters process I/O requests, which move between the guest operating system of a virtual machine and virtual disks.

The I/O filters can be offered by VMware or created by third parties through vSphere APIs for I/O Filtering (VAIO).

Read the following topics next:

- [About I/O Filters](#)
- [Using Flash Storage Devices with Cache I/O Filters](#)
- [System Requirements for I/O Filters](#)
- [Configure I/O Filters in the vSphere Environment](#)
- [Enable I/O Filter Data Services on Virtual Disks](#)
- [Managing I/O Filters](#)
- [I/O Filter Guidelines and Best Practices](#)
- [Handling I/O Filter Installation Failures](#)

About I/O Filters

I/O filters can gain direct access to the virtual machine I/O path. You can enable the I/O filter for an individual virtual disk level. The I/O filters are independent of the storage topology.

VMware offers certain categories of I/O filters. In addition, third-party vendors can create the I/O filters. Typically, they are distributed as packages that provide an installer to deploy the filter components on vCenter Server and ESXi host clusters.

After the I/O filters are deployed, vCenter Server configures and registers an I/O filter storage provider, also called a VASA provider, for each host in the cluster. The storage providers communicate with vCenter Server and make data services offered by the I/O filter visible in the VM Storage Policies interface. You can reference these data services when defining common rules for a VM policy. After you associate virtual disks with this policy, the I/O filters are enabled on the virtual disks.

Datastore Support

I/O filters can support all datastore types including the following:

- VMFS
- NFS 3
- NFS 4.1
- Virtual Volumes (VVol)
- vSAN

Types of I/O Filters

VMware provides certain categories of I/O filters that are installed on your ESXi hosts. In addition, VMware partners can create the I/O filters through the vSphere APIs for I/O Filtering (VAIO) developer program. The I/O filters can serve multiple purposes.

The supported types of filters include the following:

- Replication. Replicates all write I/O operations to an external target location, such as another host or cluster.
- Encryption. Offered by VMware. Provides encryption mechanisms for virtual machines. For more information, see the *vSphere Security* documentation.
- Caching. Implements a cache for virtual disk data. The filter can use a local flash storage device to cache the data and increase the IOPS and hardware utilization rates for the virtual disk. If you use the caching filter, you might need to configure a Virtual Flash Resource.
- Storage I/O control. Offered by VMware. Throttles the I/O load towards a datastore and controls the amount of storage I/O that is allocated to virtual machines during periods of I/O congestion. For more information, see the *vSphere Resource Management* documentation.

Note You can install several filters from the same category, such as caching, on your ESXi host. However, you can have only one filter from the same category per virtual disk.

I/O Filtering Components

Several components are involved in the I/O filtering process.

The basic components of I/O filtering include the following:

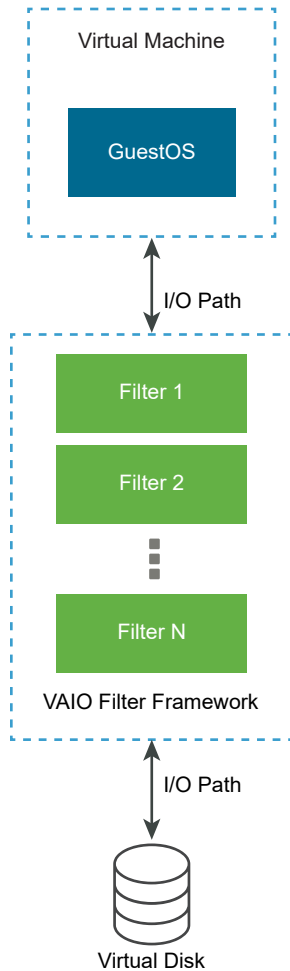
VAIO Filter Framework

A combination of user world and VMkernel infrastructure provided by ESXi. With the framework, you can add filter plug-ins to the I/O path to and from virtual disks. The infrastructure includes an I/O filter storage provider (VASA provider). The provider integrates with the Storage Policy Based Management (SPBM) system and exports filter capabilities to vCenter Server.

I/O Filter Plug-In

A software component provided by VMware or developed by VMware partners that intercepts and filters I/O data in transit between virtual disks and guest operating systems. If VMware partners develop the I/O filter, the filter might include additional optional components that help in its configuration and management.

The following figure illustrates the components of I/O filtering and the flow of I/O between the guest OS and the virtual disk.



Each Virtual Machine Executable (VMX) component of a virtual machine contains a Filter Framework that manages the I/O filter plug-ins attached to the virtual disk. The Filter Framework invokes filters when the I/O requests move between the guest operating system and the virtual disk. Also, the filter intercepts any I/O access towards the virtual disk that happens outside of a running VM.

The filters run sequentially in a specific order. For example, a replication filter executes before a cache filter. More than one filter can operate on the virtual disk, but only one for each category.

Once all filters for the particular disk verify the I/O request, the request moves to its destination, either the VM or the virtual disk.

Because the filters run in user space, any filter failures impact only the VM, but do not affect the ESXi host.

Storage Providers for I/O Filters

When I/O filters are installed on ESXi hosts, the I/O filter framework configures and registers a storage provider, also called a VASA provider, for each host in the cluster.

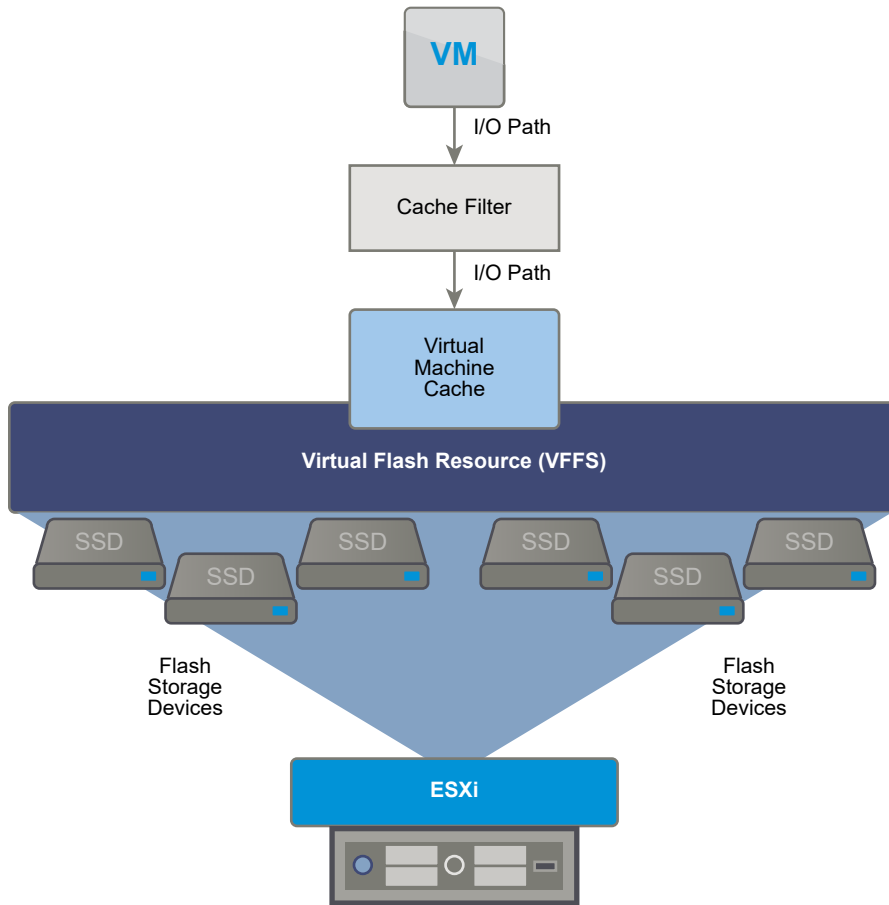
Storage providers for I/O filtering are software components that are offered by vSphere. They integrate with I/O filters and report data service capabilities that I/O filters support to vCenter Server.

The capabilities populate the VM Storage Policies interface and can be referenced in a VM storage policy. You then apply this policy to virtual disks, so that the I/O filters can process I/O for the disks.

Using Flash Storage Devices with Cache I/O Filters

A cache I/O filter can use a local flash device to cache virtual machine data.

If your caching I/O filter uses local flash devices, you need to configure a virtual flash resource, also known as VFFS volume. You configure the resource on your ESXi host before activating the filter. While processing the virtual machine read I/Os, the filter creates a virtual machine cache and places it on the VFFS volume.



To set up a virtual flash resource, you use flash devices that are connected to your host. To increase the capacity of your virtual flash resource, you can add more flash drives. An individual flash drive must be exclusively allocated to a virtual flash resource and cannot be shared with any other vSphere service, such as vSAN or VMFS.

Flash Read Cache and caching I/O filters are mutually exclusive because both functionalities use the virtual flash resource on the host. You cannot enable Flash Read Cache on a virtual disk with the cache I/O filters. Similarly, if a virtual machine has Flash Read Cache configured, it cannot use the cache I/O filters.

System Requirements for I/O Filters

To be able to use I/O filters in your environment, you must follow specific requirements.

The following requirements apply.

- Use the latest version of ESXi and vCenter Server compatible with I/O filters. Older versions might not support I/O filters, or provide only partial support.
- Check for any additional requirements that individual partner solutions might have. In specific cases, your environment might need flash devices, extra physical memory, or network connectivity and bandwidth. For information, contact your vendor or your VMware representative.

- Web server to host partner packages for filter installation. The server must remain available after initial installation. When a new host joins the cluster, the server pushes appropriate I/O filter components to the host.

Configure I/O Filters in the vSphere Environment

To set up data services that the I/O filters provide for your virtual machines, follow several steps.

Prerequisites

- Create a cluster that includes at least one ESXi host.
- For information about I/O filters provided by third parties, contact your vendor or your VMware representative.

Procedure

1 [Install I/O Filters in a Cluster](#)

If you use I/O filters provided by third parties, install the I/O filters in an ESXi host cluster.

2 [View I/O Filters and Storage Providers](#)

You can review I/O filters available in your environment and verify that the I/O filter providers appear as expected and are active.

3 [Configure Virtual Flash Resource for Caching I/O Filters](#)

If your caching I/O filter uses local flash devices, configure a virtual flash resource, also known as VFFS volume. You configure the resource on your ESXi host before activating the filter.

Install I/O Filters in a Cluster

If you use I/O filters provided by third parties, install the I/O filters in an ESXi host cluster.

VMware partners create I/O filters through the vSphere APIs for I/O Filtering (VAIO) developer program. The filter packages are typically distributed as vSphere Installation Bundles (VIBs). The VIB package can include I/O filter daemons, CIM providers, and other associated components.

Typically, to deploy the filters, you run installers provided by vendors. Installation is performed at an ESXi cluster level. You cannot install the filters on selected hosts.

Prerequisites

- Required privileges: **Host.Configuration.Query patch**.
- Verify that the I/O filter solution integrates with vSphere ESX Agent Manager and is certified by VMware.

Procedure

- ◆ Run the installer that the vendor provided.

The installer deploys the appropriate I/O filter extension on vCenter Server and the filter components on all hosts within a cluster.

A storage provider, also called a VASA provider, is automatically registered for every ESXi host in the cluster. Successful auto-registration of the I/O filter storage providers triggers an event at the host level. If the storage providers fail to auto-register, the system raises alarms on the hosts.

View I/O Filters and Storage Providers

You can review I/O filters available in your environment and verify that the I/O filter providers appear as expected and are active.

When you install a third-party I/O filter, a storage provider, also called VASA provider, is automatically registered for every ESXi host in the cluster. Successful auto-registration of the I/O filter storage providers triggers an event at the host level. If the storage providers fail to auto-register, the system raises alarms on the hosts.

Procedure

- 1 Verify that the I/O filter storage providers appear as expected and are active.

- a Navigate to vCenter Server.
- b Click the **Configure** tab, and click **Storage Providers**.
- c Review the storage providers for I/O filters.

When the I/O filter providers are properly registered, capabilities and data services that the filters offer populate the VM Storage Policies interface.

- 2 Verify that the I/O filter components are listed on your cluster and ESXi hosts.

| Option | Actions |
|-------------------------------|---|
| View I/O filters on a cluster | <ol style="list-style-type: none"> a Navigate to the cluster. b Click the Configure tab. c Under Configuration, click I/O Filters to review the filters installed in the cluster. |
| View I/O filters on a host | <ol style="list-style-type: none"> a Navigate to the host. b Click the Configure tab. c Under Storage, click I/O Filters to review the filters installed on the host. |

Configure Virtual Flash Resource for Caching I/O Filters

If your caching I/O filter uses local flash devices, configure a virtual flash resource, also known as VFFS volume. You configure the resource on your ESXi host before activating the filter.

Note This task is available only in the vSphere Web Client.

Prerequisites

To determine whether the virtual flash resource must be enabled, check with your I/O filter vendor.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under Virtual Flash, select **Virtual Flash Resource Management** and click **Add Capacity**.
- 4 From the list of available flash drives, select one or more drives to use for the virtual flash resource and click **OK**.

Results

The virtual flash resource is created. The Device Backing area lists all the drives that you use for the virtual flash resource.

Enable I/O Filter Data Services on Virtual Disks

Enabling data services that I/O filters provide is a two-step process. You create a virtual machine policy based on data services that the I/O filters provide, and then attach this policy to a virtual machine.

Prerequisites

For the caching I/O filters, configure the virtual flash resource on your ESXi host.

Procedure

- 1 Define a VM policy based on I/O filter services.

You must first create a virtual machine policy that lists data services provided by the I/O filters.

See [Create a VM Storage Policy for Host-Based Data Services](#).

- 2 Assign the I/O filter policy to a virtual machine.

To activate data services that the I/O filter provides, associate the I/O filter policy with virtual disks. You can assign the policy when you provision the virtual machine.

What to do next

If you later want to disable the I/O filter for a virtual machine, you can remove the filter rules from the VM storage policy and re-apply the policy. See [Edit or Clone a VM Storage Policy](#). Or you can edit the settings of the virtual machine and select a different storage policy that does not include the filter.

Assign the I/O Filter Policy to Virtual Machines

To activate data services that I/O filters provide, associate the I/O filter policy with virtual disks. You can assign the policy when you create or edit a virtual machine.

You can assign the I/O filter policy during an initial deployment of a virtual machine. This topic describes how to assign the policy when you create a new virtual machine. For information about other deployment methods, see the *vSphere Virtual Machine Administration* documentation.

Note You cannot change or assign the I/O filter policy when migrating or cloning a virtual machine.

Prerequisites

Verify that the I/O filter is installed on the ESXi host where the virtual machine runs.

Procedure

- 1 Start the virtual machine provisioning process and follow the appropriate steps.
- 2 Assign the same storage policy to all virtual machine files and disks.
 - a On the **Select storage** page, select a storage policy from the **VM Storage Policy** drop-down menu.
 - b Select the datastore from the list of compatible datastores and click **Next**.

The datastore becomes the destination storage resource for the virtual machine configuration file and all virtual disks. The policy also activates I/O filter services for the virtual disks.
- 3 Change the VM storage policy for the virtual disk.
Use this option to enable I/O filters just for your virtual disks.
 - a On the **Customize hardware** page, expand the **New hard disk** pane.
 - b From the **VM storage policy** drop-down menu, select the storage policy to assign to the virtual disk.
 - c (Optional) Change the storage location of the virtual disk.

Use this option to store the virtual disk on a datastore other than the datastore where the VM configuration file resides.
- 4 Complete the virtual machine provisioning process.

Results

After you create the virtual machine, the **Summary** tab displays the assigned storage policies and their compliance status.

What to do next

You can later change the virtual policy assignment. See [Change Storage Policy Assignment for Virtual Machine Files and Disks](#).

Managing I/O Filters

You can run the installer provided by your vendor to install, uninstall, or upgrade I/O filters.

When you work with I/O filters, the following considerations apply:

- vCenter Server uses ESX Agent Manager (EAM) to install and uninstall I/O filters. As an administrator, never invoke EAM APIs directly for EAM agencies that are created or used by vCenter Server. All operations related to I/O filters must go through VIM APIs. If you accidentally modify an EAM agency that was created by vCenter Server, you must revert the changes. If you accidentally destroy an EAM agency that is used by I/O filters, you must call `Vim.IoFilterManager#uninstallIoFilter` to uninstall the affected I/O filters. After uninstalling, perform a fresh reinstall.
- When a new host joins the cluster that has I/O filters, the filters installed on the cluster are deployed on the host. vCenter Server registers the I/O filter storage provider for the host. Any cluster changes become visible in the VM Storage Policies interface of the vSphere Client.
- When you move a host out of a cluster or remove it from vCenter Server, the I/O filters are uninstalled from the host. vCenter Server unregisters the I/O filter storage provider.
- If you use a stateless ESXi host, it might lose its I/O filter VIBs during a reboot. vCenter Server checks the bundles installed on the host after it reboots, and pushes the I/O filter VIBs to the host if necessary.

Uninstall I/O Filters from a Cluster

You can uninstall I/O filters deployed in an ESXi host cluster.

Prerequisites

- Required privileges: **Host.Config.Patch**.

Procedure

- 1 Uninstall the I/O filter by running the installer that your vendor provides.

During uninstallation, vSphere ESX Agent Manager automatically places the hosts into maintenance mode.

If the uninstallation is successful, the filter and any related components are removed from the hosts.

- 2 Verify that the I/O filter components are properly uninstalled from your ESXi hosts:

```
esxcli software vib list
```

The uninstalled filter no longer appears on the list.

Upgrade I/O Filters in a Cluster

Use installers provided by I/O filter vendors to upgrade I/O filters deployed in an ESXi host cluster.

An upgrade consists of uninstalling the old filter components and replacing them with the new filter components. To determine whether an installation is an upgrade, vCenter Server checks the names and versions of existing filters. If the existing filter names match the names of the new filters but have different versions, the installation is considered an upgrade.

Prerequisites

- Required privileges: **Host.Config.Patch**.

Procedure

- 1 To upgrade the filter, run the vendor-provided installer.

During the upgrade, vSphere ESX Agent Manager automatically places the hosts into maintenance mode.

The installer identifies any existing filter components and removes them before installing the new filter components.

- 2 Verify that the I/O filter components are properly uninstalled from your ESXi hosts:

```
esxcli software vib list
```

Results

After the upgrade, vSphere ESX Agent Manager places the hosts back into operational mode.

I/O Filter Guidelines and Best Practices

When you use I/O filters in your environment, follow specific guidelines and best practices.

- Because I/O filters are datastore-agnostic, all types of datastores, including VMFS, NFS, Virtual Volumes, and vSAN, are compatible with I/O filters.
- I/O filters support RDMs in virtual compatibility mode. No support is provided to RDMs in physical compatibility mode.

- Flash Read Cache and caching I/O filters are mutually exclusive because both functionalities use the virtual flash resource on the host. You cannot enable Flash Read Cache on a virtual disk with the cache I/O filters. Similarly, if a virtual machine has Flash Read Cache configured, it cannot use the cache I/O filters.
- You cannot change or assign the I/O filter policy while migrating or cloning a virtual machine. You can change the policy after you complete the migration or cloning.
- When you clone or migrate a virtual machine with I/O filter policy from one host to another, make sure that the destination host has a compatible filter installed. This requirement applies to migrations initiated by an administrator or by such functionalities as HA or DRS.
- When you convert a template to a virtual machine, and the template is configured with I/O filter policy, the destination host must have the compatible I/O filter installed.
- If you use vCenter Site Recovery Manager to replicate virtual disks, the resulting disks on the recovery site do not have the I/O filter policies. You must create the I/O filter policies in the recovery site and reattach them to the replicated disks.
- You can attach an encryption I/O filter to a new virtual disk when you create a virtual machine. You cannot attach the encryption filter to an existing virtual disk.
- If your virtual machine has a snapshot tree associated with it, you cannot add, change, or remove the I/O filter policy for the virtual machine.

Migrating Virtual Machines with I/O Filters

When you migrate a virtual machine with I/O filters, specific considerations apply.

If you use Storage vMotion to migrate a virtual machine with I/O filters, a destination datastore must be connected to hosts with compatible I/O filters installed.

You might need to migrate a virtual machine with I/O filters across different types of datastores, for example between VMFS and Virtual Volumes. If you do so, make sure that the VM storage policy includes rule sets for every type of datastore you are planning to use. For example, if you migrate your virtual machine between the VMFS and Virtual Volumes datastores, create a mixed VM storage policy that includes the following rules:

- Common Rules for the I/O filters
- Rule Set 1 for the VMFS datastore. Because Storage Policy Based Management does not offer an explicit VMFS policy, the rule set must include tag-based rules for the VMFS datastore.
- Rule Set 2 for the Virtual Volumes datastore

When Storage vMotion migrates the virtual machine, the correct rule set that corresponds to the target datastore is selected. The I/O filter rules remain unchanged.

If you do not specify rules for datastores and define only Common Rules for the I/O filters, the system applies default storage policies for the datastores.

Handling I/O Filter Installation Failures

Typically, all ESXi hosts in a cluster have the same set of I/O filters installed. Occasionally, failures might happen during installation.

If an I/O filter installation fails on a host, the system generates events that report the failure. In addition, an alarm on the host shows the reason for the failure. Examples of failures include the following:

- The VIB URL is not accessible from the host.
- The VIB has an invalid format.
- The VIB requires the host to be in maintenance mode for an upgrade or uninstallation.
- The VIB requires the host to reboot after the installation or uninstallation.
- Attempts to put the host in maintenance mode fail because the virtual machine cannot be evacuated from the host.
- The VIB requires manual installation or uninstallation.

vCenter Server can resolve some failures. You might have to intervene for other failures. For example, you might need to edit the VIB URL, manually evacuate or power off virtual machines, or manually install or uninstall VIBs.

Install I/O Filters on a Single ESXi Host

For troubleshooting purposes, you can download an ESXi component of the I/O filter, packaged as a VIB file, and install it on the ESXi host. Use the `esxcli` command to install the VIB file.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 Install the VIBs by running the following command:

```
esxcli software vib install --depot path_to_VMware_vib_ZIP_file
```

Options for the `install` command allow you to perform a dry run, specify a specific VIB, bypass acceptance-level verification, and so on. Do not bypass verification on production systems. See the *vSphere Command-Line Interface Reference* documentation.

- 2 Verify that the VIBs are installed on your ESXi host.

```
esxcli software vib list
```

Storage Hardware Acceleration

24

The hardware acceleration functionality enables the ESXi host to integrate with compliant storage systems. The host can offload certain virtual machine and storage management operations to the storage systems. With the storage hardware assistance, your host performs these operations faster and consumes less CPU, memory, and storage fabric bandwidth.

Block storage devices, Fibre Channel and iSCSI, and NAS devices support the hardware acceleration.

For additional details, see the VMware knowledge base article at <http://kb.vmware.com/kb/1021976>.

Read the following topics next:

- [Hardware Acceleration Benefits](#)
- [Hardware Acceleration Requirements](#)
- [Hardware Acceleration Support Status](#)
- [Hardware Acceleration for Block Storage Devices](#)
- [Hardware Acceleration on NAS Devices](#)
- [Hardware Acceleration Considerations](#)

Hardware Acceleration Benefits

When the hardware acceleration functionality is supported, the host can get hardware assistance and perform several tasks faster and more efficiently.

The host can get assistance with the following activities:

- Migrating virtual machines with Storage vMotion
- Deploying virtual machines from templates
- Cloning virtual machines or templates
- VMFS clustered locking and metadata operations for virtual machine files
- Provisioning thick virtual disks
- Creating fault-tolerant virtual machines

- Creating and cloning thick disks on NFS datastores

Hardware Acceleration Requirements

The hardware acceleration functionality works only if you use an appropriate host and storage array combination.

Table 24-1. Hardware Acceleration Storage Requirements

| ESXi | Block Storage Devices | NAS Devices |
|------|---|--|
| ESXi | Support T10 SCSI standard, or block storage plug-ins for array integration (VAAI) | Support NAS plug-ins for array integration |

Note If your SAN or NAS storage fabric uses an intermediate appliance in front of a storage system that supports hardware acceleration, the intermediate appliance must also support hardware acceleration and be properly certified. The intermediate appliance might be a storage virtualization appliance, I/O acceleration appliance, encryption appliance, and so on.

Hardware Acceleration Support Status

For each storage device and datastore, the vSphere Client display the hardware acceleration support status.

The status values are Unknown, Supported, and Not Supported. The initial value is Unknown.

For block devices, the status changes to Supported after the host successfully performs the offload operation. If the offload operation fails, the status changes to Not Supported. The status remains Unknown if the device provides partial hardware acceleration support.

With NAS, the status becomes Supported when the storage can perform at least one hardware offload operation.

When storage devices do not support or provide partial support for the host operations, your host reverts to its native methods to perform unsupported operations.

Hardware Acceleration for Block Storage Devices

With hardware acceleration, your host can integrate with block storage devices, Fibre Channel or iSCSI, and use certain storage array operations.

ESXi hardware acceleration supports the following array operations:

- Full copy, also called clone blocks or copy offload. Enables the storage arrays to make full copies of data within the array without having the host read and write the data. This operation reduces the time and network load when cloning virtual machines, provisioning from a template, or migrating with vMotion.

- Block zeroing, also called write same. Enables storage arrays to zero out a large number of blocks to provide newly allocated storage, free of previously written data. This operation reduces the time and network load when creating virtual machines and formatting virtual disks.
- Hardware assisted locking, also called atomic test and set (ATS). Supports discrete virtual machine locking without use of SCSI reservations. This operation allows disk locking per sector, instead of the entire LUN as with SCSI reservations.

Check with your vendor for the hardware acceleration support. Certain storage arrays require that you activate the support on the storage side.

On your host, the hardware acceleration is enabled by default. If your storage does not support the hardware acceleration, you can disable it.

In addition to hardware acceleration support, ESXi includes support for array thin provisioning. For information, see [ESXi and Array Thin Provisioning](#).

Disable Hardware Acceleration for Block Storage Devices

On your host, the hardware acceleration for block storage devices is enabled by default. You can use the vSphere Client advanced settings to disable the hardware acceleration operations.

As with any advanced settings, before you disable the hardware acceleration, consult with the VMware support team.

Procedure

- 1 Navigate to the host.
- 2 Click the **Configure** tab.
- 3 Under **System**, click **Advanced System Settings**.
- 4 Change the value for any of the options to 0 (disabled):
 - VMFS3.HardwareAcceleratedLocking
 - DataMover.HardwareAcceleratedMove
 - DataMover.HardwareAcceleratedInit

Managing Hardware Acceleration on Block Storage Devices

To integrate with the block storage arrays, vSphere uses the ESXi extensions called Storage APIs - Array Integration (VAAI). With this integration, vSphere can use the array hardware operations.

In the vSphere 5.x and later releases, these extensions are implemented as the T10 SCSI commands. As a result, with the devices that support the T10 SCSI standard, your ESXi host can communicate directly and does not require the VAAI plug-ins.

If the device does not support T10 SCSI or provides partial support, ESXi reverts to using the VAAI plug-ins, installed on your host. The host can also use a combination of the T10 SCSI commands and plug-ins. The VAAI plug-ins are vendor-specific and can be either VMware or partner developed. To manage the VAAI capable device, your host attaches the VAAI filter and vendor-specific VAAI plug-in to the device.

For information about whether your storage requires VAAI plug-ins or supports hardware acceleration through T10 SCSI commands, see the *VMware Compatibility Guide* or contact your storage vendor.

You can use several `esxcli` commands to query storage devices for the hardware acceleration support information. For the devices that require the VAAI plug-ins, the claim rule commands are also available. For information about `esxcli` commands, see *Getting Started with vSphere Command-Line Interfaces*.

Display Hardware Acceleration Plug-Ins and Filter

To communicate with the devices that do not support the T10 SCSI standard, your host uses a single VAAI filter and a vendor-specific VAAI plug-in. Use the `esxcli` command to view the hardware acceleration filter and plug-ins currently loaded into your system.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ Run the **`esxcli storage core plugin list --plugin-class=value`** command.

For *value*, enter one of the following parameters:

- Type `VAAI` to display plug-ins.

The output of this command is similar to the following example:

```
#esxcli storage core plugin list --plugin-class=VAAI
Plugin name      Plugin class
VMW_VAAIP_EQL    VAAI
VMW_VAAIP_NETAPP VAAI
VMW_VAAIP_CX     VAAI
```

- Type `Filter` to display the Filter.

The output of this command is similar to the following example:

```
esxcli storage core plugin list --plugin-class=Filter
Plugin name  Plugin class
VAAI_FILTER  Filter
```

Verify Hardware Acceleration Support Status

Use the `esxcli` command to verify the hardware acceleration support status of a particular storage device.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ Run the `esxcli storage core device list -d=device_ID` command.

The output shows the hardware acceleration, or VAAI, status that can be unknown, supported, or unsupported.

```
# esxcli storage core device list -d naa.XXXXXXXXXXXXX4c
naa.XXXXXXXXXXXXX4c
Display Name: XXXX Fibre Channel Disk(naa.XXXXXXXXXXXXX4c)
Size: 20480
Device Type: Direct-Access
Multipath Plugin: NMP
XXXXXXXXXXXXXXXXXX
Attached Filters: VAAI_FILTER
VAAI Status: supported
XXXXXXXXXXXXXXXXXX
```

Verify Hardware Acceleration Support Details

Use the `esxcli` command to query whether the block storage device provides the hardware acceleration support.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ Run the **esxcli storage core device vaa1 status get -d=device_ID** command.

If a VAAI plug-in manages the device, the output shows the name of the plug-in attached to the device. The output also shows the support status for each T10 SCSI based primitive, if available. Output appears in the following example:

```
# esxcli storage core device vaa1 status get -d naa.XXXXXXXXXXX4c
naa.XXXXXXXXXXX4c
VAAI Plugin Name: VMW_VAAIP_SYMM
ATS Status: supported
Clone Status: supported
Zero Status: supported
Delete Status: unsupported
```

List Hardware Acceleration Claim Rules

Each block storage device managed by a VAAI plug-in needs two claim rules. One claim rule specifies the hardware acceleration filter and another specifies the hardware acceleration plug-in for the device. You can use the **esxcli** commands to list the hardware acceleration filter and plug-in claim rules.

Procedure

- 1 To list the filter claim rules, run the **esxcli storage core claimrule list --claimrule-class=Filter** command.

In this example, the filter claim rules specify devices that the VAAI_FILTER filter claims.

```
# esxcli storage core claimrule list --claimrule-class=Filter
Rule Class Rule Class Type Plugin Matches XCOPY Use Array
Reported Values XCOPY Use Multiple Segments XCOPY Max Transfer Size KiB
Filter 65430 runtime vendor VAAI_FILTER
vendor=EMC model=SYMMETRIX False
False 0
Filter 65430 file vendor VAAI_FILTER
vendor=EMC model=SYMMETRIX False
False 0
Filter 65431 runtime vendor VAAI_FILTER
vendor=DGC model=* False
False 0
Filter 65431 file vendor VAAI_FILTER
vendor=DGC model=* False
False 0
```

- 2 To list the VAAI plug-in claim rules, run the **esxcli storage core claimrule list --claimrule-class=VAAI** command.

In this example, the VAAI claim rules specify devices that the VAAI plug-in claims.

```
esxcli storage core claimrule list --claimrule-class=VAAI
Rule Class Rule Class Type Plugin Matches XCOPY Use
```

| Array | Reported Values | XCOPY Use | Multiple Segments | XCOPY Max Transfer Size KiB |
|------------|----------------------|----------------|-------------------|-----------------------------|
| VAAI | 65430 runtime vendor | VMW_VAAIP_SYMM | | |
| vendor=EMC | model=SYMMETRIX | | False | |
| False | | 0 | | |
| VAAI | 65430 file vendor | VMW_VAAIP_SYMM | | |
| vendor=EMC | model=SYMMETRIX | | False | |
| False | | 0 | | |
| VAAI | 65431 runtime vendor | VMW_VAAIP_CX | | |
| vendor=DGC | model=* | | False | |
| False | | 0 | | |
| VAAI | 65431 file vendor | VMW_VAAIP_CX | | |
| vendor=DGC | model=* | | False | |
| False | | 0 | | |

Add Hardware Acceleration Claim Rules

To configure the hardware acceleration for a new array, add two claim rules, one for the VAAI filter and another for the VAAI plug-in. For the new claim rules to be active, you first define the rules and then load them into your system.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 Define a new claim rule for the VAAI filter by running the **`esxcli storage core claimrule add --claimrule-class=Filter --plugin=VAAI_FILTER`** command.
- 2 Define a new claim rule for the VAAI plug-in by running the **`esxcli storage core claimrule add --claimrule-class=VAAI`** command.
- 3 Load both claim rules by running the following commands:
`esxcli storage core claimrule load --claimrule-class=Filter`
`esxcli storage core claimrule load --claimrule-class=VAAI`
- 4 Run the VAAI filter claim rule by using the **`esxcli storage core claimrule run --claimrule-class=Filter`** command.

Note Only the filter-class rules must be run. When the VAAI filter claims a device, it automatically finds the proper VAAI plug-in to attach.

Example: Defining Hardware Acceleration Claim Rules

This example shows how to configure the hardware acceleration for IBM arrays using the VMW_VAAIP_T10 plug-in. Use the following sequence of commands. For information about the options that the command takes, see [Add Multipathing Claim Rules](#).

```
# esxcli storage core claimrule add --claimrule-class=Filter --
plugin=VAAI_FILTER --type=vendor --vendor=IBM --autoassign

# esxcli storage core claimrule add --claimrule-class=VAAI --
plugin=VMW_VAAIP_T10 --type=vendor --vendor=IBM --autoassign

# esxcli storage core claimrule load --claimrule-class=Filter

# esxcli storage core claimrule load --claimrule-class=VAAI

# esxcli storage core claimrule run --claimrule-class=Filter
```

Configure XCOPY Parameters

XCOPY is one of the VAAI primitives that is used for offloading tasks to the storage array. For example, you can use XCOPY to offload such operations as migration or cloning of virtual machines to the array instead of consuming vSphere resources to perform these tasks.

You can use the XCOPY mechanism with all storage arrays that support the SCSI T10 based VMW_VAAIP_T10 plug-in developed by VMware. To enable the XCOPY mechanism, create a claim rule of the VAAI class.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ Use the following command and enter the XCOPY options:

```
esxcli storage core claimrule add --claimrule-class=VAAI
```

For information about the options that the command takes, see [Add Multipathing Claim Rules](#).

| Option | Description |
|---|--|
| <code>-a --xcopy-use-array-values</code> | Use array reported values for XCOPY commands. |
| <code>-s --xcopy-use-multi-segs</code> | Use multiple segments for XCOPY commands. Valid only when <code>--xcopy-use-array-values</code> is specified. |
| <code>-m --xcopy-max-transfer-size</code> | Maximum transfer size in MB for the XCOPY commands when you use a transfer size different than array reported. Valid only when <code>--xcopy-use-array-values</code> is specified. |
| <code>-k --xcopy-max-transfer-size-kib</code> | Maximum transfer size in KiB for the XCOPY commands when you use a transfer size different than array reported. Valid only if <code>--xcopy-use-array-values</code> is specified. |

Example: Configuring XCOPY

```
■ # esxcli storage core claimrule add -r 914 -t vendor -V XtremIO -M XtremApp -P
VMW_VAAIP_T10 -c VAAI -a -s -k 64
```

- ```
esxcli storage core claimrule add -r 65430 -t vendor -V EMC -M SYMMETRIX -P
VMW_VAAIP_SYMM -c VAAI -a -s -m 200
```

## Delete Hardware Acceleration Claim Rules

Use the `esxcli` command to delete existing hardware acceleration claim rules.

### Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

### Procedure

- ◆ Run the following commands:

```
esxcli storage core claimrule remove -r claimrule_ID --claimrule-
class=Filter
```

```
esxcli storage core claimrule remove -r claimrule_ID --claimrule-
class=VAAI
```

## Hardware Acceleration on NAS Devices

With the hardware acceleration, ESXi hosts can integrate with NAS devices and use several hardware operations that NAS storage provides. The hardware acceleration uses vSphere APIs for Array Integration (VAAI) to facilitate communications between the hosts and storage devices.

The VAAI NAS framework supports both versions of NFS storage, NFS 3 and NFS 4.1.

The VAAI NAS uses a set of storage primitives to offload storage operations from the host to the array. The following list shows the supported NAS operations:

- **Full File Clone.** Supports an ability of NAS device to clone virtual disk files. This operation is similar to the VMFS block cloning, except that NAS devices clone entire files instead of file segments.
- **Reserve Space.** Supports an ability of storage arrays to allocate space for a virtual disk file in the thick format.

Typically, when you create a virtual disk on an NFS datastore, the NAS server determines the allocation policy. The default allocation policy on most NAS servers is thin and does not guarantee backing storage to the file. However, the reserve space operation can instruct the NAS device to use vendor-specific mechanisms to reserve space for a virtual disk. As a result, you can create thick virtual disks on the NFS datastore.

- Array-based Snapshots. Creation of virtual machine snapshots can be offloaded to the array.

---

**Note** Storage DRS does not detect array-based snapshots on NFS datastores. As a result, when you perform such operations as cloning a VM with an array-based snapshot, Storage DRS does not provide any recommendations.

---

- Extended Statistics. Supports visibility to space use on NAS devices. This functionality is useful for thin provisioning.

With NAS storage devices, the hardware acceleration integration is implemented through vendor-specific NAS plug-ins. These plug-ins are typically created by vendors and are distributed as VIB packages through a website. No claim rules are required for the NAS plug-ins to function.

Several tools for installing and upgrading VIB packages are available. They include the `esxcli` commands and vSphere Update Manager. For more information, see the *vSphere Upgrade and Installing and Administering VMware vSphere Update Manager* documentation.

## Install NAS Plug-In

Install vendor-distributed hardware acceleration NAS plug-ins on your host.

This topic provides an example for a VIB package installation using the `esxcli` command. For more details, see the *vSphere Upgrade* documentation.

### Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

### Procedure

- 1 Place your host into the maintenance mode.
- 2 Set the host acceptance level:

```
esxcli software acceptance set --level=value
```

The host acceptance level must be the same or less restrictive than the acceptance level of any VIB you want to add to the host. The *value* can be one of the following:

- VMwareCertified
- VMwareAccepted
- PartnerSupported
- CommunitySupported

- 3 Install the VIB package:

```
esxcli software vib install -v|--viburl=URL
```

The *URL* specifies the URL to the VIB package to install. `http:`, `https:`, `ftp:`, and `file:` are supported.



- 4 Verify that the plug-in is installed:

```
esxcli software vib list
```

- 5 Restart your host for the installation to take effect.

## Uninstall NAS Plug-Ins

To uninstall a NAS plug-in, remove the VIB package from your host.

This topic discusses how to uninstall a VIB package using the `esxcli` command. For more details, see the *vSphere Upgrade* documentation.

### Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

### Procedure

- 1 Uninstall the plug-in:

```
esxcli software vib remove -n|--vibName=name
```

The *name* is the name of the VIB package to remove.

- 2 Verify that the plug-in is removed:

```
esxcli software vib list
```

- 3 Reboot your host for the change to take effect.

## Update NAS Plug-Ins

Upgrade hardware acceleration NAS plug-ins on your host when a storage vendor releases a new plug-in version.

### Prerequisites

This topic discusses how to update a VIB package using the `esxcli` command. For more details, see the *vSphere Upgrade* documentation.

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

### Procedure

- 1 Upgrade to a new plug-in version:

```
esxcli software vib update -v|--viburl=URL
```

The *URL* specifies the URL to the VIB package to update. `http:`, `https:`, `ftp:`, and `file:` are supported.

- 2 Verify that the correct version is installed:

```
esxcli software vib list
```

- 3 Reboot the host.

## Verify Hardware Acceleration Status for NAS

In addition to the client, you can use the `esxcli` command to verify the hardware acceleration status of the NAS device.

### Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

### Procedure

- ◆ Run the `esxcli storage nfs list` command.

The Hardware Acceleration column in the output shows whether hardware acceleration is supported.

## Hardware Acceleration Considerations

When you use the hardware acceleration functionality, certain considerations apply.

Several reasons might cause a hardware-accelerated operation to fail.

For any primitive that the array does not implement, the array returns an error. The error triggers the ESXi host to attempt the operation using its native methods.

The VMFS data mover does not leverage hardware offloads and instead uses software data movement when one of the following occurs:

- The source and destination VMFS datastores have different block sizes.
- The source file type is RDM and the destination file type is non-RDM (regular file).
- The source VMDK type is eagerzeroedthick and the destination VMDK type is thin.
- The source or destination VMDK is in sparse or hosted format.
- The source virtual machine has a snapshot.
- The logical address and transfer length in the requested operation are not aligned to the minimum alignment required by the storage device. All datastores created with the vSphere Client are aligned automatically.
- The VMFS has multiple LUNs or extents, and they are on different arrays.

Hardware cloning between arrays, even within the same VMFS datastore, does not work.

# Storage Provisioning and Space Reclamation

# 25

vSphere supports two models of storage provisioning, thick provisioning and thin provisioning.

## Thick provisioning

It is a traditional model of the storage provisioning. With the thick provisioning, large amount of storage space is provided in advance in anticipation of future storage needs. However, the space might remain unused causing underutilization of storage capacity.

## Thin provisioning

This method contrast with thick provisioning and helps you eliminate storage underutilization problems by allocating storage space in a flexible on-demand manner. With ESXi, you can use two models of thin provisioning, array-level and virtual disk-level.

Thin provisioning allows you to report more virtual storage space than there is real physical capacity. This discrepancy can lead to storage over-subscription, also called over-provisioning. When you use thin provisioning, monitor actual storage usage to avoid conditions when you run out of physical storage space.

Read the following topics next:

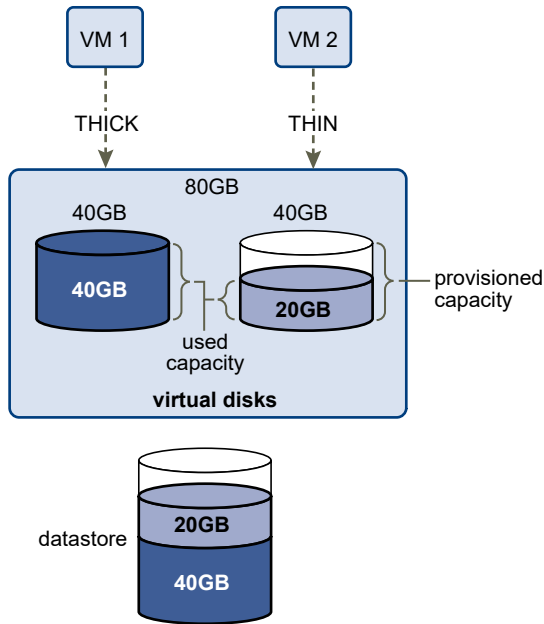
- [Virtual Disk Thin Provisioning](#)
- [ESXi and Array Thin Provisioning](#)
- [Storage Space Reclamation](#)

## Virtual Disk Thin Provisioning

When you create a virtual machine, a certain amount of storage space on a datastore is provisioned to virtual disk files.

By default, ESXi offers a traditional storage provisioning method for virtual machines. With this method, you first estimate how much storage the virtual machine might need for its entire life cycle. You then provision a fixed amount of storage space to the VM virtual disk in advance, for example, 40 GB. The entire provisioned space is committed to the virtual disk. A virtual disk that immediately occupies the entire provisioned space is a thick disk.

ESXi supports thin provisioning for virtual disks. With the disk-level thin provisioning feature, you can create virtual disks in a thin format. For a thin virtual disk, ESXi provisions the entire space required for the disk's current and future activities, for example 40 GB. However, the thin disk uses only as much storage space as the disk needs for its initial operations. In this example, the thin-provisioned disk occupies only 20 GB of storage. If the disk requires more space, it can expand into its entire 40 GB of provisioned space.



## About Virtual Disk Provisioning Policies

When you perform certain virtual machine management operations, you can specify a provisioning policy for the virtual disk file. The operations include creating a virtual disk, cloning a virtual machine to a template, or migrating a virtual machine.

NFS datastores with Hardware Acceleration and VMFS datastores support the following disk provisioning policies. On NFS datastores that do not support Hardware Acceleration, only thin format is available.

You can use Storage vMotion or cross-host Storage vMotion to transform virtual disks from one format to another.

### Thick Provision Lazy Zeroed

Creates a virtual disk in a default thick format. Space required for the virtual disk is allocated when the disk is created. Data remaining on the physical device is not erased during creation, but is zeroed out on demand later on first write from the virtual machine. Virtual machines do not read stale data from the physical device.

### Thick Provision Eager Zeroed

A type of thick virtual disk that supports clustering features such as Fault Tolerance. Space required for the virtual disk is allocated at creation time. In contrast to the thick provision lazy zeroed format, the data remaining on the physical device is zeroed out when the virtual disk is created. It might take longer to create virtual disks in this format than to create other types of disks. Increasing the size of an Eager Zeroed Thick virtual disk causes a significant stall time for the virtual machine.

### Thin Provision

Use this format to save storage space. For the thin disk, you provision as much datastore space as the disk would require based on the value that you enter for the virtual disk size. However, the thin disk starts small and at first, uses only as much datastore space as the disk needs for its initial operations. If the thin disk needs more space later, it can grow to its maximum capacity and occupy the entire datastore space provisioned to it.

Thin provisioning is the fastest method to create a virtual disk because it creates a disk with just the header information. It does not allocate or zero out storage blocks. Storage blocks are allocated and zeroed out when they are first accessed.

---

**Note** If a virtual disk supports clustering solutions such as Fault Tolerance, do not make the disk thin.

---

## Create Thin Provisioned Virtual Disks

To save storage space, you can create a virtual disk in a thin provisioned format. The thin provisioned virtual disk starts small and expands as more disk space is required. You can create thin disks only on the datastores that support disk-level thin provisioning.

This procedure assumes that you are creating a new virtual machine. For information, see the *vSphere Virtual Machine Administration* documentation.

### Procedure

- 1 Create a virtual machine.
  - a Right-click any inventory object that is a valid parent object of a virtual machine, such as a data center, folder, cluster, resource pool, or host, and select **New Virtual Machine**.
  - b Select **Create a new virtual machine** and click **Next**.
  - c Follow the steps required to create a virtual machine.
- 2 Configure the thin virtual disk.
  - a On the Customize Hardware page, click the **Virtual Hardware** tab.
  - b Click the **New Hard disk** triangle to expand the hard disk options.

- c (Optional) Adjust the default disk size.

With a thin virtual disk, the disk size value shows how much space is provisioned and guaranteed to the disk. At the beginning, the virtual disk might not use the entire provisioned space. The actual storage use value can be less than the size of the virtual disk.

- d Select **Thin Provision** for Disk Provisioning.

- 3 Finish the virtual machine creation.

#### Results

You created a virtual machine with a disk in the thin format.

#### What to do next

If you created a virtual disk in the thin format, you can later inflate it to its full size.

## View Virtual Machine Storage Resources

You can view how datastore storage space is allocated for your virtual machines.

#### Procedure

- 1 Browse to the virtual machine.
- 2 Double-click the virtual machine and click the **Summary** tab.
- 3 Review the storage use information in the upper right area of the **Summary** tab.

#### Results

**Storage Usage** shows how much datastore space is occupied by virtual machine files, including configuration and log files, snapshots, virtual disks, and so on. When the virtual machine is running, the used storage space also includes swap files.

For virtual machines with thin disks, the actual storage use value might be less than the size of the virtual disk.

## Determine the Disk Format of a Virtual Machine

You can determine whether your virtual disk is in thick or thin format.

#### Procedure

- 1 Right-click the virtual machine and select **Edit Settings**.
- 2 Click the **Virtual Hardware** tab.
- 3 Click the **Hard Disk** triangle to expand the hard disk options.

The **Type** text box shows the format of your virtual disk.

## What to do next

If your virtual disk is in the thin format, you can inflate it to its full size.

## Inflate Thin Virtual Disks

If you created a virtual disk in the thin format, you can change the format to thick.

You use the datastore browser to inflate the thin virtual disk.

### Prerequisites


- Make sure that the datastore where the virtual machine resides has enough space.
- Make sure that the virtual disk is thin.
- Remove snapshots.
- Power off your virtual machine.

### Procedure

- 1 Navigate to the folder of the virtual disk you want to inflate.
  - a Navigate to the virtual machine.
  - b Click the **Datastores** tab.

The datastore that stores the virtual machine files is listed.
  - c Right-click the datastore and select **Browse Files**.

The datastore browser displays contents of the datastore.
- 2 Expand the virtual machine folder and browse to the virtual disk file that you want to convert.

The file has the `.vmdk` extension and is marked with the virtual disk  icon.
- 3 Select the virtual disk file and click **Inflate**.

---

**Note** The option might not be available if the virtual disk is thick or when the virtual machine is running.

---

### Results

The inflated virtual disk occupies the entire datastore space originally provisioned to it.

## Handling Datastore Over-Subscription

Because the provisioned space for thin disks can be greater than the committed space, a datastore over-subscription can occur, which results in the total provisioned space for the virtual machine disks on the datastore being greater than the actual capacity.

Over-subscription can be possible because usually not all virtual machines with thin disks need the entire provisioned datastore space simultaneously. However, if you want to avoid over-subscribing the datastore, you can set up an alarm that notifies you when the provisioned space reaches a certain threshold.

For information on setting alarms, see the *vCenter Server and Host Management* documentation.

If your virtual machines require more space, the datastore space is allocated on a first come first served basis. When the datastore runs out of space, you can add more physical storage and increase the datastore.

See [Increase VMFS Datastore Capacity](#).

## ESXi and Array Thin Provisioning

You can use thin-provisioned storage arrays with ESXi.

The ESXi host integrates with block-based storage and performs these tasks:

- The host can recognize underlying thin-provisioned LUNs and monitor their space use to avoid running out of physical space. The LUN space might change if, for example, your VMFS datastore expands or if you use Storage vMotion to migrate virtual machines to the thin-provisioned LUN. The host warns you about breaches in physical LUN space and about out-of-space conditions.
- The host can run the automatic T10 `unmap` command from VMFS6 and VM guest operating systems to reclaim unused space from the array. VMFS5 supports a manual space reclamation method.

---

**Note** ESXi does not support enabling and disabling of thin provisioning on a storage device.

---

## Requirements

To use the thin provisioning reporting and space reclamation features, follow these requirements:

- Use an appropriate ESXi version.

**Table 25-1. ESXi Versions and Thin Provisioning Support**

| Supported thin provisioning components  | ESXi 6.0 and earlier                                           | ESXi 6.5 and later  |
|-----------------------------------------|----------------------------------------------------------------|---------------------|
| Thin provisioning                       | Yes                                                            | Yes                 |
| Unmap command originating from VMFS     | Manual for VMFS5. Use <code>esxcli storage vmfs unmap</code> . | Automatic for VMFS6 |
| Unmap command originating from guest OS | Yes. Limited support.                                          | Yes (VMFS6)         |

- Use storage systems that support T10-based vSphere Storage APIs - Array Integration (VAAI), including thin provisioning and space reclamation. For information, contact your storage provider and check the *VMware Compatibility Guide*.



## Monitoring Space Use

The thin provision integration functionality helps you to monitor the use of space on thin-provisioned LUNs and to avoid running out of space.

The following sample flow demonstrates how the ESXi host and the storage array interact to generate breach of space and out-of-space warnings for a thin-provisioned LUN. The same mechanism applies when you use Storage vMotion to migrate virtual machines to the thin-provisioned LUN.

- 1 Using storage-specific tools, your storage administrator provisions a thin LUN and sets a soft threshold limit that, when reached, triggers an alert. This step is vendor-specific.
- 2 Using the vSphere Client, you create a VMFS datastore on the thin-provisioned LUN. The datastore spans the entire logical size that the LUN reports.
- 3 As the space used by the datastore increases and reaches the set soft threshold, the following actions take place:

- a The storage array reports the breach to your host.
- b Your host triggers a warning alarm for the datastore.

You can contact the storage administrator to request more physical space. Alternatively, you can use Storage vMotion to evacuate your virtual machines before the LUN runs out of capacity.

- 4 If no space is left to allocate to the thin-provisioned LUN, the following actions take place:
- a The storage array reports out-of-space condition to your host.

---

**Caution** In certain cases, when a LUN becomes full, it might go offline or get unmapped from the host.

---

- b The host pauses virtual machines and generates an out-of-space alarm.

You can resolve the permanent out-of-space condition by requesting more physical space from the storage administrator.

## Identify Thin-Provisioned Storage Devices

Use the `esxcli` command to verify whether a particular storage device is thin-provisioned.

### Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

### Procedure

- ◆ Run the `esxcli storage core device list -d=device_ID` command.

## Results

The following thin provisioning status indicates that the storage device is thin-provisioned.

```
esxcli storage core device list -d naa.XXXXXXXXXXXXX4c
naa.XXXXXXXXXXXXX4c
 Display Name: XXXX Fibre Channel Disk(naa.XXXXXXXXXXXXX4c)
 Size: 20480
 Device Type: Direct-Access
 Multipath Plugin: NMP

 Thin Provisioning Status: yes

```

An unknown status indicates that a storage device is thick.

---

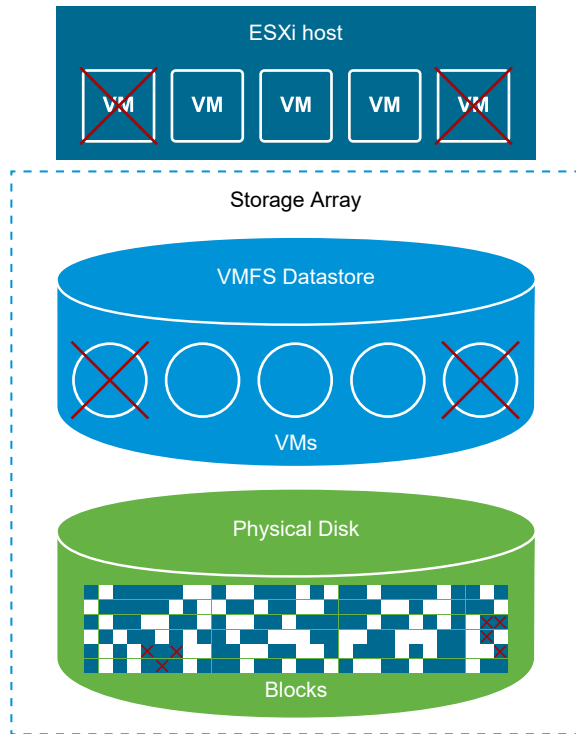
**Note** Some storage systems present all devices as thin-provisioned no matter whether the devices are thin or thick. Their thin provisioning status is always `yes`. For details, check with your storage vendor.

---

## Storage Space Reclamation

ESXi supports the space reclamation command, also called SCSI unmap command, that originates from a VMFS datastore or a VM guest operating system. The command helps thin-provisioned storage arrays to reclaim unused space from the VMFS datastore and thin virtual disks on the datastore. The VMFS6 datastore can send the space reclamation command automatically. With the VMFS5 datastore, you can manually reclaim storage space.

You free storage space inside the VMFS datastore when you delete or migrate the VM, consolidate a snapshot, and so on. Inside the virtual machine, storage space is freed when you delete files on the thin virtual disk. These operations leave blocks of unused space on the storage array. However, when the array is not aware that the data was deleted from the blocks, the blocks remain allocated by the array until the datastore releases them. VMFS uses the SCSI unmap command to indicate to the array that the storage blocks contain deleted data, so that the array can unallocate these blocks.



The command can also originate directly from the guest operating system. Both VMFS5 and VMFS6 datastores can provide support to the unmap command that proceeds from the guest operating system. However, the level of support is limited on VMFS5.

Depending on the type of your VMFS datastore, you use different methods to configure space reclamation for the datastore and your virtual machines.

Watch the following video to learn more about how space reclamation works.



(Space Reclamation with VMFS )

### What to read next

- [Space Reclamation Requests from VMFS Datastores](#)

Deleting or removing files from a VMFS datastore frees space within the file system. This free space is mapped to a storage device until the file system releases or unmaps it. ESXi supports reclamation of free space, which is also called the unmap operation.

- [Space Reclamation Requests from Guest Operating Systems](#)

ESXi supports the unmap commands issued directly from a guest operating system to reclaim storage space. The level of support and requirements depend on the type of datastore where your virtual machine resides.

## Space Reclamation Requests from VMFS Datastores

Deleting or removing files from a VMFS datastore frees space within the file system. This free space is mapped to a storage device until the file system releases or unmaps it. ESXi supports reclamation of free space, which is also called the unmap operation.

The operation helps the storage array to reclaim unused free space. Unmapped space can be then used for other storage allocation requests and needs.

### Asynchronous Reclamation of Free Space on VMFS6 Datastore

On VMFS6 datastores, ESXi supports the automatic asynchronous reclamation of free space. VMFS6 can run the unmap command to release free storage space in the background on thin-provisioned storage arrays that support unmap operations.

Asynchronous unmap processing has several advantages:

- Unmap requests are sent at a constant rate, which helps to avoid any instant load on the backing array.
- Freed regions are batched and unmapped together.
- I/O performance of other workloads are not impacted by the unmap command.

For VMFS6 datastores, you can configure the following space reclamation parameters.

#### Space reclamation granularity

Granularity defines the minimum size of a released space sector that underlying storage can reclaim. Storage cannot reclaim those sectors that are smaller in size than the specified granularity.

For VMFS6, reclamation granularity equals the block size. When you specify the block size as 1 MB, the granularity is also 1 MB. Storage sectors of the size smaller than 1 MB are not reclaimed.

---

**Note** Certain storage arrays recommend an optimal unmap granularity. ESXi supports automatic unmap processing on arrays with the recommended unmap granularity of 1 MB or greater, for example 16 MB,. On the arrays with the optimal granularity of 1 MB and less, the unmap operation is supported if the granularity is a factor of 1 MB. For example, 1 MB is divisible by 512 bytes, 4 KB, 64 KB, and so on.

---

#### Space reclamation method

The method can be either priority or fixed. When the method you use is priority, you configure the priority rate. For the fixed method, you must indicate the bandwidth in MB per second.

#### Space reclamation priority

This parameter defines the rate at which the space reclamation operation is performed when you use the priority reclamation method. Typically, VMFS6 can send the unmap commands either in bursts or sporadically depending on the workload and configuration. For VMFS6, you can specify one of the following options.

| Space Reclamation Priority | Description                                                                               | Configuration                                 |
|----------------------------|-------------------------------------------------------------------------------------------|-----------------------------------------------|
| None                       | Disables the unmap operations for the datastore.                                          | vSphere Client<br><code>esxcli</code> command |
| Low (default)              | Sends the unmap command at a less frequent rate, 25–50 MB per second.                     | vSphere Client<br><code>esxcli</code> command |
| Medium                     | Sends the command at a rate twice faster than the low rate, 50–100 MB per second.         | <code>esxcli</code> command                   |
| High                       | Sends the command at a rate three times faster than the low rate, over 100 MB per second. | <code>esxcli</code> command                   |

**Note** The ESXi host of version 6.5 does not recognize the medium and high priority rates. If you migrate the VMs to the host version 6.5, the rate defaults to low.

After you enable space reclamation, the VMFS6 datastore can start releasing the blocks of unused space only when it has at least one open file. This condition can be fulfilled when, for example, you power on one of the VMs on the datastore.

## Manual Reclamation of Free Space on VMFS5 Datastore

VMFS5 and earlier file systems do not unmap free space automatically, but you can use the `esxcli storage vmfs unmap` command to reclaim space manually. When you use the command, be aware that it might send many unmap requests at a time. This action can lock some of the resources during the operation.

## Configure Space Reclamation for a VMFS6 Datastore

When you create a VMFS6 datastore, you can modify the default parameters for automatic space reclamation.

At the VMFS6 datastore creation time, the only available method for the space reclamation is priority. To use the fixed method, edit the space reclamation settings of the existing datastore.

### Procedure

- 1 In the vSphere Client object navigator, browse to a host, a cluster, or a data center.
- 2 From the right-click menu, select **Storage > New Datastore**.
- 3 Follow the steps required to create a VMFS6 datastore.

- 4 On the **Partition configuration** page, specify the space reclamation parameters.

The parameters define granularity and the priority rate at which space reclamation operations are performed. You can also use this page to disable space reclamation for the datastore.

| Option                               | Description                                                                                                                                                                                                                                                                                                              |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Block size</b>                    | The block size on a VMFS datastore defines the maximum file size and the amount of space the file occupies. VMFS6 supports the block size of 1 MB.                                                                                                                                                                       |
| <b>Space reclamation granularity</b> | Specify granularity for the unmap operation. Unmap granularity equals the block size, which is 1 MB.<br>Storage sectors of the size smaller than 1 MB are not reclaimed.                                                                                                                                                 |
| <b>Space reclamation priority</b>    | Select one of the following options. <ul style="list-style-type: none"> <li>■ Low (default). Use the priority method for space reclamation. Enable the unmap operation at a low priority rate.</li> <li>■ None. Select this option if you want to disable the space reclamation operations for the datastore.</li> </ul> |

**Note** In the vSphere Client, the only available settings for the space reclamation priority are Low and None. To change the settings to Medium or High, use the `esxcli` command. See [Use the ESXCLI Command to Change Space Reclamation Parameters](#).

- 5 Finish the datastore creation process.

## Results

After you enable space reclamation, the VMFS6 datastore can start releasing the blocks of unused space only when it has at least one open file. This condition can be fulfilled when, for example, you power on one of the VMs on the datastore.

## Change Space Reclamation Settings

When you create a VMFS6 datastore in the vSphere Client, the only method for space reclamation you can specify is the priority method. To enable the fixed method, modify the space reclamation settings for the existing datastore.

### Procedure

- 1 Navigate to the datastore.
- 2 Select **Edit Space Reclamation** from the right-click menu.
- 3 Specify the space reclamation setting.

| Option                                                  | Description                                                                                 |
|---------------------------------------------------------|---------------------------------------------------------------------------------------------|
| <b>Enable automatic space reclamation at fixed rate</b> | Use the fixed method for space reclamation. Specify reclamation bandwidth in MB per second. |
| <b>Disable automatic space reclamation</b>              | Deleted or unmapped blocks are not reclaimed.                                               |

4 Click **OK** to save the new settings.

## Results

The modified value for the space reclamation priority appears on the **General** page for the datastore.

## Use the ESXCLI Command to Change Space Reclamation Parameters

You can change the default space reclamation priority, granularity, and other parameters.

### Procedure

- ◆ Use the following command to set space reclamation parameters.

```
esxcli storage vmfs reclaim config set
```

The command takes these options:

| Option                          | Description                                                                                                                                                           |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-b --reclaim-bandwidth</b>   | Space reclamation fixed bandwidth in MB per second.                                                                                                                   |
| <b>-g --reclaim-granularity</b> | Minimum granularity of automatic space reclamation in bytes.                                                                                                          |
| <b>-m --reclaim-method</b>      | Method of automatic space reclamation. Supported options: <ul style="list-style-type: none"> <li>■ priority</li> <li>■ fixed</li> </ul>                               |
| <b>-p --reclaim-priority</b>    | Priority of automatic space reclamation. Supported options: <ul style="list-style-type: none"> <li>■ none</li> <li>■ low</li> <li>■ medium</li> <li>■ high</li> </ul> |
| <b>-l --volume-label</b>        | The label of the target VMFS volume.                                                                                                                                  |
| <b>-u --volume-uuid</b>         | The uuid of the target VMFS volume.                                                                                                                                   |

### Example: Setting the Reclamation Method to Fixed

Use the following example to set the reclamation method to fixed and the rate to 100 MB per second.

```
esxcli storage vmfs reclaim config set --volume-label datastore_name --reclaim-method fixed
-b 100
```

## Verify Settings for Automatic Space Reclamation

After you configure or edit space reclamation parameters for a VMFS6 datastore, you can review your settings.

### Procedure

- 1 In the vSphere Client, navigate to the datastore.

- 2 Click the **Configure** tab.
- 3 Click **General**, and verify the space reclamation settings.
  - a Under Properties, expand **File system** and review the value for the space reclamation granularity.
  - b Under Space Reclamation, review the setting for the space reclamation priority.

If you configured any values through the `esxcli` command, for example, Medium or High for the space reclamation priority, these values also appear in the vSphere Client.

### Example: Obtaining Parameters for VMFS6 Space Reclamation

You can also use the `esxcli storage vmfs reclaim config get -l=VMFS_label|-u=VMFS_uuid` command to obtain information for the space reclamation configuration.

```
esxcli storage vmfs reclaim config get -l my_datastore
Reclaim Granularity: 1048576 Bytes
Reclaim Priority: low
```

## Manually Reclaim Accumulated Storage Space

On VMFS datastores that do not support automatic space reclamation, you can use the `esxcli` command to reclaim unused storage space manually.

### Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

### Procedure

- 1 To reclaim unused storage blocks on the thin-provisioned device, run the following command:

```
esxcli storage vmfs unmap
```

The command takes these options:

| Option                                      | Description                                                                                                                                        |
|---------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-l --volume-label=volume_label</code> | The label of the VMFS volume to unmap. A mandatory argument. If you specify this argument, do not use <code>-u --volume-uuid=volume_uuid</code> .  |
| <code>-u --volume-uuid=volume_uuid</code>   | The UUID of the VMFS volume to unmap. A mandatory argument. If you specify this argument, do not use <code>-l --volume-label=volume_label</code> . |
| <code>-n --reclaim-unit=number</code>       | Number of VMFS blocks to unmap per iteration. An optional argument. If it is not specified, the command uses the default value of 200.             |

- 2 To verify whether the unmap process has finished, search for `unmap` in the `vmkernel.log` file.



## Space Reclamation Requests from Guest Operating Systems

ESXi supports the unmap commands issued directly from a guest operating system to reclaim storage space. The level of support and requirements depend on the type of datastore where your virtual machine resides.

Inside a virtual machine, storage space is freed when, for example, you delete files on the thin virtual disk. The guest operating system notifies VMFS about freed space by sending the unmap command. The unmap command sent from the guest operating system releases space within the VMFS datastore. The command then proceeds to the array, so that the array can reclaim the freed blocks of space.

### Space Reclamation for VMFS6 Virtual Machines

VMFS6 generally supports automatic space reclamation requests that generate from the guest operating systems, and passes these requests to the array. Many guest operating systems can send the unmap command and do not require any additional configuration. The guest operating systems that do not support the automatic unmaps might require user intervention. For information about guest operating systems that support the automatic space reclamation on VMFS6, contact your vendor.

Generally, the guest operating systems send the unmap commands based on the unmap granularity they advertise. For details, see documentation provided with your guest operating system.

The following considerations apply when you use space reclamation with VMFS6:

- VMFS6 processes the unmap request from the guest OS only when the space to reclaim equals 1 MB or is a multiple of 1 MB. If the space is less than 1 MB or is not aligned to 1 MB, the unmap requests are not processed.
- For VMs with snapshots in the default SEsparse format, VMFS6 supports the automatic space reclamation only on ESXi hosts version 6.7 or later. If you migrate VMs to ESXi hosts version 6.5 or earlier, the automatic space reclamation stops working for the VMs with snapshots.

Space reclamation affects only the top snapshot and works when the VM is powered on.

### Space Reclamation for VMFS5 Virtual Machines

Typically, the unmap command that generates from the guest operation system on VMFS5 cannot be passed directly to the array. You must run the `esxcli storage vmfs unmap` command to trigger unmaps for the array.

However, for a limited number of the guest operating systems, VMFS5 supports the automatic space reclamation requests.

To send the unmap requests from the guest operating system to the array, the virtual machine must meet the following prerequisites:

- The virtual disk must be thin-provisioned.
- Virtual machine hardware must be of version 11 (ESXi 6.0) or later.

- The advanced setting EnableBlockDelete must be set to 1.
- The guest operating system must be able to identify the virtual disk as thin.

`vmkfstools` is one of the ESXi Shell commands for managing VMFS volumes, storage devices, and virtual disks. You can perform many storage operations using the `vmkfstools` command. For example, you can create and manage VMFS datastores on a physical partition, or manipulate virtual disk files, stored on VMFS or NFS datastores.

**Note** After you make a change using the `vmkfstools`, the vSphere Client might not be updated immediately. Use a refresh or rescan operation from the client.

For more information on the ESXi Shell, see *Getting Started with vSphere Command-Line Interfaces*.

Read the following topics next:

- [vmkfstools Command Syntax](#)
- [The vmkfstools Command Options](#)

## vmkfstools Command Syntax

Generally, you do not need to log in as the root user to run the `vmkfstools` commands. However, some commands, such as the file system commands, might require the root user login.

The `vmkfstools` command supports the following command syntax:

`vmkfstools options target.`

Target specifies a partition, device, or path to apply the command option to.

**Table 26-1.** `vmkfstools` Command Arguments

| Argument  | Description                                                                                                                                                                                                                                                                                                                                         |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| options   | One or more command-line options and associated arguments that you use to specify the activity for <code>vmkfstools</code> to perform. For example, selecting the disk format when creating a new virtual disk.<br>After entering the option, specify a target on which to perform the operation. Target can indicate a partition, device, or path. |
| partition | Specifies disk partitions. This argument uses a <code>disk_ID:P</code> format, where <code>disk_ID</code> is the device ID returned by the storage array and <code>P</code> is an integer that represents the partition number. The partition digit must be greater than zero (0) and must correspond to a valid VMFS partition.                    |

**Table 26-1. vmkfstools Command Arguments (continued)**

| Argument | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| device   | <p>Specifies devices or logical volumes. This argument uses a path name in the ESXi device file system. The path name begins with <code>/vmfs/devices</code>, which is the mount point of the device file system.</p> <p>Use the following formats when you specify different types of devices:</p> <ul style="list-style-type: none"> <li>■ <code>/vmfs/devices/disks</code> for local or SAN-based disks.</li> <li>■ <code>/vmfs/devices/lvm</code> for ESXi logical volumes.</li> <li>■ <code>/vmfs/devices/generic</code> for generic SCSI devices.</li> </ul>                                                                                                                                                                                                                                                                                              |
| path     | <p>Specifies a VMFS file system or file. This argument is an absolute or relative path that names a directory symbolic link, a raw device mapping, or a file under <code>/vmfs</code>.</p> <ul style="list-style-type: none"> <li>■ To specify a VMFS file system, use this format: <div data-bbox="644 701 1042 730" data-label="Text"> <pre>/vmfs/volumes/file_system_UUID</pre> </div> <p>or</p> <div data-bbox="644 823 1053 852" data-label="Text"> <pre>/vmfs/volumes/file_system_label</pre> </div> </li> <li>■ To specify a file on a VMFS datastore, use this format: <div data-bbox="644 934 1367 989" data-label="Text"> <pre>/vmfs/volumes/file_system_label file_system_UUID/[dir]/myDisk.vmdk</pre> </div> </li> </ul> <p>If the current working directory is the parent directory of <code>myDisk.vmdk</code>, do not enter the entire path.</p> |

## The vmkfstools Command Options

The `vmkfstools` command has several options. Some of the options are suggested for advanced users only.

The long and single-letter forms of the options are equivalent. For example, the following commands are identical.

```
vmkfstools --createfs vmfs6 --blocksize 1m disk_ID:P
vmkfstools -C vmfs6 -b 1m disk_ID:P
```

### -v Suboption

The `-v` suboption indicates the verbosity level of the command output.

The format for this suboption is as follows:

```
-v --verbose number
```

You specify the *number* value as an integer from 1 through 10.

You can specify the `-v` suboption with any `vmkfstools` option. If the output of the option is not suitable for use with the `-v` suboption, `vmkfstools` ignores `-v`.

---

**Note** Because you can include the `-v` suboption in any `vmkfstools` command line, `-v` is not included as a suboption in the option descriptions.

---

## File System Options

File system options allow you to create and manage VMFS datastores. These options do not apply to NFS. You can perform many of these tasks through the vSphere Client.

### Listing Attributes of a VMFS Datastore

Use the `vmkfstools` command to list attributes of a VMFS datastore.

```
-P|--queryfs
 -h|--humanreadable
```

When you use this option on any file or directory that resides on a VMFS datastore, the option lists the attributes of the specified datastore. The listed attributes typically include the file system label, the number of extents for the datastore, the UUID, and a list of the devices where each extent resides.

---

**Note** If any device backing VMFS file system goes offline, the number of extents and available space change accordingly.

---

You can specify the `-h|--humanreadable` suboption with the `-P` option. If you do so, `vmkfstools` lists the capacity of the volume in a more readable form.

#### Example: Example of Listing VMFS Attributes

```
~ vmkfstools -P -h /vmfs/volumes/my_vmfs
VMFS-5.81 (Raw Major Version: 14) file system spanning 1 partitions.
File system label (if any): my_vmfs
Mode: public
Capacity 99.8 GB, 97.5 GB available, file block size 1 MB, max supported file size 62.9 TB
UUID: 571fe2fb-ec4b8d6c-d375-XXXXXXXXXXXX
Partitions spanned (on "lvm"):
 eui.3863316131XXXXXX:1
Is Native Snapshot Capable: YES
```

### Creating a VMFS Datastore or a Scratch Partition

Use the `vmkfstools` command to create a VMFS datastore or a scratch partition.

```
-C|--createfs [vmfs5|vmfs6|vfat]
```

This option creates the VMFS datastore on the specified SCSI partition, such as `disk_ID:P`. The partition becomes the head partition of the datastore. For VMFS5 and VMFS6, the only available block size is 1 MB.

You can specify the following suboptions with the `-c` option.

- `-S|--setfsname` - Define the volume label of the VMFS datastore you are creating. Use this suboption only with the `-c` option. The label you specify can be up to 128 characters long and cannot contain any leading or trailing blank spaces.

---

**Note** vCenter Server supports the 80 character limit for all its entities. If a datastore name exceeds this limit, the name gets shortened when you add this datastore to vCenter Server.

---

After you define a volume label, you can use it whenever you specify the VMFS datastore for the `vmkfstools` command. The volume label appears in listings generated for the `ls -l` command and as a symbolic link to the VMFS volume under the `/vmfs/volumes` directory.

To change the VMFS volume label, use the `ln -sf` command. Use the following as an example:

```
ln -sf /vmfs/volumes/UUID /vmfs/volumes/datastore
```

*datastore* is the new volume label to use for the *UUID* VMFS.

---

**Note** If your host is registered with vCenter Server, any changes you make to the VMFS volume label get overwritten by vCenter Server. This operation guarantees that the VMFS label is consistent across all vCenter Server hosts.

---

- `-Y|--unmapGranularity #[bBsSkKmMgGtT]` - This suboption applies to VMFS6 only. Define granularity for the unmap operation. The default granularity is 1 MB. As with the block size, enter the unit type.
- `-O|--unmapPriority <none|low|medium|high>` - This suboption applies to VMFS6 only. Define the priority for the unmap operation.

### Example: Example for Creating a VMFS File System

This example illustrates creating a VMFS6 datastore named `my_vmfs` on the `naa.ID:1` partition.

```
~ vmkfstools -C vmfs6 -S my_vmfs /vmfs/devices/disks/naa.ID:1
```

### Adding an Extent to a VMFS Datastore

Use the `vmkfstools` command to add an extent to a VMFS datastore.

When you add an extent, you span the VMFS datastore from the head partition across the partition specified by *span\_partition*.

```
-Z|--spanfs span_partition head_partition
```

You must specify the full path name for the head and span partitions, for example `/vmfs/devices/disks/disk_ID:1`. Each time you use this option, you add an extent to the VMFS datastore, so that the datastore spans multiple partitions.

---

**Caution** When you run this option, you lose all data that previously existed on the SCSI device you specified in *span\_partition*.

---

### Example: Example for Extending a VMFS Datastore

In this example, you extend the existing head partition of the VMFS datastore over a new partition.

```
~ vmkfstools -Z /vmfs/devices/disks/naa.disk_ID_2:1 /vmfs/devices/disks/naa.disk_ID_1:1
```

The extended datastore spans two partitions, `naa.disk_ID_1:1` and `naa.disk_ID_2:1`. In this example, `naa.disk_ID_1:1` is the name of the head partition.

### Expanding a VMFS Datastore

Instead of adding an extent to a VMFS datastore, you can increase the size of an existing datastore. Use the `vmkfstools -G` command.

You might increase the datastore size after the underlying storage had its capacity increased.

The command uses the following option:

```
-G|--growfs device device
```

This option expands the VMFS datastore or its specific extent. For example,

```
vmkfstools --growfs /vmfs/devices/disks/disk_ID:1 /vmfs/devices/disks/disk_ID:1
```

### Upgrading a VMFS Datastore

If you use a VMFS3 datastore, you must upgrade it to VMFS5.

When upgrading the datastore, use the following option:

```
-T|--upgradenvmfs /vmfs/volumes/UUID
```

The upgrade is a one-way process. After you have converted a VMFS3 datastore to VMFS5, you cannot revert it back.

All hosts accessing the datastore must support VMFS5.

### Virtual Disk Options

Virtual disk options allow you to set up, migrate, and manage virtual disks stored on your datastores. You can also perform most of these tasks through the vSphere Client.

## Supported Disk Formats

When you create or clone a virtual disk, you can use the `-d|--diskformat` suboption to specify the format for the disk.

Choose from the following formats:

- `zeroedthick` (default) – Space required for the virtual disk is allocated during creation. Any data remaining on the physical device is not erased during creation, but is zeroed out on demand on first write from the virtual machine. The virtual machine does not read stale data from disk.
- `eagerzeroedthick` – Space required for the virtual disk is allocated at creation time. In contrast to `zeroedthick` format, the data remaining on the physical device is zeroed out during creation. It might take much longer to create disks in this format than to create other types of disks.
- `thin` – Thin-provisioned virtual disk. Unlike with the `thick` format, space required for the virtual disk is not allocated during creation, but is supplied, zeroed out, on demand.
- `rdm:device` – Virtual compatibility mode raw disk mapping.
- `rdmp:device` – Physical compatibility mode (pass-through) raw disk mapping.
- `2gbsparse` – A sparse disk with the maximum extent size of 2 GB. You can use disks in this format with hosted VMware products, such as VMware Fusion. However, you cannot power on the sparse disk on an ESXi host unless you first re-import the disk with `vmkfstools` in a compatible format, such as `thick` or `thin`.

### Disk Formats on NFS Datastores

The only disk formats you can use for NFS are `thin`, `thick`, `zeroedthick`, and `2gbsparse`.

`Thick`, `zeroedthick`, and `thin` formats usually behave the same because the NFS server and not the ESXi host determines the allocation policy. The default allocation policy on most NFS servers is `thin`. However, on NFS servers that support Storage APIs - Array Integration, you can create virtual disks in `zeroedthick` format. The reserve space operation enables NFS servers to allocate and guarantee space.

For more information on array integration APIs, see [Chapter 24 Storage Hardware Acceleration](#).

## Creating a Virtual Disk

Use the `vmkfstools` command to create a virtual disk.

```
-c|--createvirtualdisk size[bB|sS|kK|mM|gG]
-d|--diskformat [thin|zeroedthick|eagerzeroedthick]
-W|--objecttype [file|vsan|vvol]
--policyFile fileName
```



This option creates a virtual disk at the specified path on a datastore. Specify the size of the virtual disk. When you enter the value for *size*, you can indicate the unit type by adding a suffix of *k* (kilobytes), *m* (megabytes), or *g* (gigabytes). The unit type is not case-sensitive. `vmkfstools` interprets either *k* or *K* to mean kilobytes. If you do not specify a unit type, `vmkfstools` defaults to bytes.

You can specify the following suboptions with the `-c` option.

- `-d|--diskformat` specifies disk formats.
- `-W|--objecttype` specifies whether the virtual disk is a file on a VMFS or NFS datastore, or an object on a vSAN or Virtual Volumes datastore.
- `--policyFile fileName` specifies VM storage policy for the disk.

### Example: Example for Creating a Virtual Disk

This example shows how to create a two-gigabyte virtual disk file named `disk.vmdk`. You create the disk on the VMFS datastore named `myVMFS`. The disk file represents an empty virtual disk that virtual machines can access.

```
vmkfstools -c 2048m /vmfs/volumes/myVMFS/disk.vmdk
```

### Initializing a Virtual Disk

Use the `vmkfstools` command to initialize a virtual disk.

```
-w|--writezeros
```

This option cleans the virtual disk by writing zeros over all its data. Depending on the size of your virtual disk and the I/O bandwidth to the device hosting the virtual disk, completing this command might take a long time.

---

**Caution** When you use this command, you lose any existing data on the virtual disk.

---

### Inflating a Thin Virtual Disk

Use the `vmkfstools` command to inflate a thin virtual disk.

```
-j|--inflatedisk
```

This option converts a `thin` virtual disk to `eagerzeroedthick`, preserving all existing data. The option allocates and zeroes out any blocks that are not already allocated.

### Converting a Zeroedthick Virtual Disk to an Eagerzeroedthick Disk

Use the `vmkfstools` command to convert any zeroedthick virtual disk to an eagerzeroedthick disk.

```
-k|--eagerzero
```

While performing the conversion, this option preserves any data on the virtual disk.

Follow this example:

```
vmkfstools --eagerzero /vmfs/volumes/myVMFS/VMName/disk.vmdk
```

## Removing Zeroed Blocks

Use the `vmkfstools` command to remove zeroed blocks.

```
-K|--punchzero
```

This option deallocates all zeroed out blocks and leaves only those blocks that were allocated previously and contain valid data. The resulting virtual disk is in thin format.

## Deleting a Virtual Disk

Use the `vmkfstools` command to delete a virtual disk file at the specified path on the VMFS volume.

Use the following option:

```
-U|--deletevirtualdisk
```

## Renaming a Virtual Disk

Use the `vmkfstools` command to rename a virtual disk file at the specified path on the VMFS volume.

You must specify the original filename or file path *oldName* and the new filename or file path *newName*.

```
-E|--renamevirtualdisk oldName newName
```

## Cloning or Converting a Virtual Disk or RDM

Use the `vmkfstools` command to create a copy of a virtual disk or raw disk you specify.

A non-root user cannot clone a virtual disk or an RDM. You must specify the original filename or file path *oldName* and the new filename or file path *newName*.

```
-i|--clonevirtualdisk oldName newName
 -d|--diskformat [thin|zeroedthick|eagerzeroedthick|rdm:device|rdmp:device|2gbsparse]
 -W|--objecttype [file|vsan|vvol]
 --policyFile fileName
 -N|--avoidnativeclone
```

Use the following suboptions to change corresponding parameters for the copy you create.

- `-d|--diskformat` specifies disk formats.

- `-W|--objecttype` specifies whether the virtual disk is a file on a VMFS or NFS datastore, or an object on a vSAN or Virtual Volumes datastore.
- `--policyFile fileName` specifies VM storage policy for the disk.

By default, ESXi uses its native methods to perform the cloning operations. If your array supports the cloning technologies, you can off-load the operations to the array. To avoid the ESXi native cloning, specify the `-N|--avoidnativeclone` option.

### Example: Example for Cloning or Converting a Virtual Disk

This example illustrates cloning the contents of a primary virtual disk from the `templates` repository to a virtual disk file named `myOS.vmdk` on the `myVMFS` file system.

```
vmkfstools -i /vmfs/volumes/myVMFS/templates/gold-primary.vmdk /vmfs/volumes/myVMFS/myOS.vmdk
```

You can configure a virtual machine to use this virtual disk by adding lines to the virtual machine configuration file, as in the following example:

```
scsi0:0.present = TRUE
scsi0:0.fileName = /vmfs/volumes/myVMFS/myOS.vmdk
```

If you want to convert the format of the disk, use the `-d|--diskformat` suboption.

This suboption is useful when you import virtual disks in a format not compatible with ESXi, for example 2gbsparse format. After you convert the disk, you can attach this disk to a new virtual machine you create in ESXi.

For example:

```
vmkfstools -i /vmfs/volumes/myVMFS/templates/gold-primary.vmdk /vmfs/volumes/myVMFS/
myOS.vmdk -d thin
```

### Extending a Virtual Disk

After you create a virtual machine, you can use the `vmkfstools` command to extend the size of a disk allocated to the virtual machine.

```
-X|--extendvirtualdisk newSize[bBsSkKmMgGtT]
```

Specify the `newSize` parameter adding an appropriate unit suffix. The unit type is not case-sensitive. `vmkfstools` interprets either `k` or `K` to mean kilobytes. If you do not specify the unit type, `vmkfstools` defaults to kilobytes.

The `newSize` parameter defines the entire new size, not just the increment you add to the disk.

For example, to extend a 4-g virtual disk by 1 g, enter: `vmkfstools -X 5g disk name`.

You can extend the virtual disk to the `eagerzeroedthick` format by using the `-d eagerzeroedthick` option.

When you use the `-x` option, the following considerations apply:

- Do not extend the base disk of a virtual machine that has snapshots associated with it. If you do, you can no longer commit the snapshot or revert the base disk to its original size.
- After you extend the disk, you might need to update the file system on the disk. As a result, the guest operating system recognizes the new size of the disk and can use it.

## Upgrading Virtual Disks

This option converts the specified virtual disk file from ESX Server 2 formats to the ESXi format.

Use this option to convert virtual disks of type LEGACYSPARSE, LEGACYPLAIN, LEGACYVMFS, LEGACYVMFS\_SPARSE, and LEGACYVMFS\_RDM.

```
-M|--migratevirtualdisk
```

## Creating a Virtual Compatibility Mode Raw Device Mapping

Use the `vmkfstools` command to create a Raw Device Mapping (RDM) file on a VMFS volume and map a raw LUN to this file. After this mapping is established, you can access the LUN as you would a normal VMFS virtual disk. The file length of the mapping is the same as the size of the raw LUN it points to.

```
-r|--createrdm device
```

When specifying the *device* parameter, use the following format:

```
/vmfs/devices/disks/disk_ID:P
```

### Example: Example for Creating a Virtual Compatibility Mode RDM

In this example, you create an RDM file named *my\_rdm.vmdk* and map the *disk\_ID* raw disk to that file.

```
vmkfstools -r /vmfs/devices/disks/disk_ID my_rdm.vmdk
```

You can configure a virtual machine to use the *my\_rdm.vmdk* mapping file by adding the following lines to the virtual machine configuration file:

```
scsi0:0.present = TRUE
scsi0:0.fileName = /vmfs/volumes/myVMFS/my_rdm.vmdk
```

## Creating a Physical Compatibility Mode Raw Device Mapping

Use the `vmkfstools` command to map a pass-through raw device to a file on a VMFS volume. With the mapping, a virtual machine can bypass ESXi SCSI command filtering when accessing its virtual disk. This type of mapping is useful when the virtual machine must send proprietary SCSI commands, for example, when SAN-aware software runs on the virtual machine.

```
-z|--createrdmpassthru device example.vmdk
```

After you establish this type of mapping, you can use it to access the raw disk as you access any other VMFS virtual disk.

When specifying the *device* path, use the following format:

```
/vmfs/devices/disks/device_ID
```

For the .vmdk name, use this format. Make sure to create the datastore before using the command.

```
/vmfs/volumes/datastore_name/example.vmdk
```

For example,

```
vmkfstools -z /vmfs/devices/disks/naa.600a000000000000... /vmfs/volumes/datastore1/
mydisk.vmdk
```

## Listing Attributes of an RDM

Use the `vmkfstools` command to list the attributes of a raw disk mapping. The attributes help you identify the storage device to which your RDM files maps.

```
-q|--queryrdm my_rdm.vmdk
```

This option prints the name of the raw disk RDM. The option also prints other identification information, like the disk ID, for the raw disk.

### Example: Example of Listing RDM Attributes

```
vmkfstools -q /vmfs/volumes/VMFS/my_vm/my_rdm.vmdk

Disk /vmfs/volumes/VMFS/my_vm/my_rdm.vmdk is a Passthrough Raw Device Mapping

Maps to: vml.02000000006005076801900207700000000000005323134352020
```

## Displaying Virtual Disk Geometry

Use the `vmkfstools` command to get information about the geometry of a virtual disk.

```
-g|--geometry
```

The output is in the form: `Geometry information C/H/S`, where `C` represents the number of cylinders, `H` represents the number of heads, and `S` represents the number of sectors.

---

**Note** When you import virtual disks from hosted VMware products to the ESXi host, you might see a disk geometry mismatch error message. A disk geometry mismatch might also trigger problems when you load a guest operating system or run a newly created virtual machine.

---

## Checking and Repairing Virtual Disks

Use the `vmkfstools` command to check or repair a virtual disk if it gets corrupted.

```
-x|--fix [check|repair]
```

For example,

```
vmkfstools -x check /vmfs/volumes/my_datastore/my_disk.vmdk
```

## Checking Disk Chain for Consistency

Use the `vmkfstools` command to check the entire snapshot chain. You can determine if any of the links in the chain are corrupted or any invalid parent-child relationships exist.

```
-e|--chainConsistent
```

## Storage Device Options

You can use the device options of the `vmkfstools` command to perform administrative task for physical storage devices.

## Managing SCSI Reservations of LUNs

Use the `vmkfstools` command to reserve a SCSI LUN for exclusive use by the ESXi host. You can also release a reservation so that other hosts can access the LUN, and reset a reservation, forcing all reservations from the target to be released.

```
-L|--lock [reserve|release|lunreset|targetreset|busreset|readkeys|readresv] device
```

---

**Caution** Using the `-L` option can interrupt the operations of other servers on a SAN. Use the `-L` option only when troubleshooting clustering setups.

---

Unless advised by VMware, never use this option on a LUN hosting a VMFS volume.

You can specify the `-L` option in several ways:

- `-L reserve` – Reserves the specified LUN. After the reservation, only the server that reserved that LUN can access it. If other servers attempt to access that LUN, a reservation error appears.
- `-L release` – Releases the reservation on the specified LUN. Other servers can access the LUN again.
- `-L lunreset` – Resets the specified LUN by clearing any reservation on the LUN and making the LUN available to all servers again. The reset does not affect any of the other LUNs on the device. If another LUN on the device is reserved, it remains reserved.
- `-L targetreset` – Resets the entire target. The reset clears any reservations on all the LUNs associated with that target and makes the LUNs available to all servers again.

- `-L busreset` – Resets all accessible targets on the bus. The reset clears any reservation on all the LUNs accessible through the bus and makes them available to all servers again.
- `-L readkeys` – Reads the reservation keys registered with a LUN. Applies to SCSI-III persistent group reservation functionality.
- `-L readresv` – Reads the reservation state on a LUN. Applies to SCSI-III persistent group reservation functionality.

When entering the *device* parameter, use the following format:

```
/vmfs/devices/disks/disk_ID:P
```

## Breaking Device Locks

Use the `vmkfstools` command to break the device lock on a particular partition.

```
-B|--breaklock device
```

When entering the *device* parameter, use the following format:

```
/vmfs/devices/disks/disk_ID:P
```

You can use this command when a host fails in the middle of a datastore operation, such as expand the datastore, add an extent, or resignature. When you run this command, make sure that no other host is holding the lock.