vmware PRESS

Essential Virtual SAN (VSAN)

Administrator's Guide to VMware® Virtual SAN

Cormac Hogan
Duncan Epping

FREE SAMPLE CHAPTER

SHARE WITH OTHERS

Essential Virtual SAN (VSAN)

Administrator's Guide to VMware® Virtual SAN

Second Edition

VMware Press is the official publisher of VMware books and training materials, which provide guidance on the critical topics facing today's technology professionals and students. Enterprises, as well as small- and medium-sized organizations, adopt virtualization as a more agile way of scaling IT to meet business needs. VMware Press provides proven, technically accurate information that will help them meet their goals for customizing, building, and maintaining their virtual environment.

With books, certification and study guides, video training, and learning tools produced by world-class architects and IT experts, VMware Press helps IT professionals master a diverse range of topics on virtualization and cloud computing and is the official source of reference materials for preparing for the VMware Certified Professional examination.

VMware Press is also pleased to have localization partners that can publish its products into more than 42 languages, including, but not limited to, Chinese (Simplified), Chinese (Traditional), French, German, Greek, Hindi, Japanese, Korean, Polish, Russian, and Spanish.

For more information about VMware Press, please visit
http://www.vmwarepress.com.

This page intentionally left blank

Essential Virtual SAN (VSAN)

Administrator's Guide to VMware® Virtual SAN

Second Edition

Cormac Hogan
Duncan Epping

vmware® PRESS

Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

Essential Virtual SAN (VSAN) Second Edition

Published by Pearson Education, Inc.

Publishing as VMware Press

ISBN-10: 0-13-451166-2

ISBN-13: 978-0-13-451166-5

Library of Congress Control Number: 2016903470

Printed in the United States of America

First Printing: May 2016

Warning and Disclaimer

Special Sales

VMWARE PRESS PROGRAM MANAGER
Karl Childs

EXECUTIVE EDITOR
Mary Beth Ray

TECHNICAL EDITORS
Christian Dickmann
John Nicholson

SENIOR DEVELOPMENT EDITOR
Ellie Bru

MANAGING EDITOR
Sandra Schroeder

PROJECT EDITOR
Mandie Frank

COPY EDITOR
Cenveo® Publisher Services

PROOFREADER
Cenveo Publisher Services

INDEXER
Cenveo Publisher Services

EDITORIAL ASSISTANT
Vanessa Evans

DESIGNER
Chuti Prasertsith

COMPOSITOR
Cenveo Publisher Services

"I close my eyes, and think of home. Another city goes by in the night. Ain't it funny how it is, you never miss it 'til it has gone away.
And my heart is lying there and will be 'til my dying day."

We would like to dedicate this book to the greatest band on earth, Iron Maiden.
—Cormac & Duncan

This page intentionally left blank

Contents

This page intentionally left blank

Foreword by Christos Karamanolis

We are in the midst of a major evolution of hardware, especially storage technologies and networking. With NAND-based Flash, a single device can deliver 1M I/O operations a second. That's more than what a high-end disk array could deliver just a few years ago. With 10 Gbps networks becoming commodity, the industry is already eying 40 or even 100 Gbps Ethernet. Efficient protocols like RDMA and NVMe reduce the cycles CPUs need to consume to harness the capabilities of emerging storage and networking technologies. At the same time, CPU densities are doubling every two years.

It was within this technology forecast that I started the Virtual SAN project together with a handful of brilliant engineers back in 2010. We recognized that commodity servers have enough oomph to take over services like storage and networking traditionally offered by dedicated hardware appliances. We argued that software running on general-purpose (commodity) hardware is the way to go. Simply put, the technology evolution allows for the democratization of enterprise storage and networking. A software-defined architecture allows the customer to ride the wave of technology trends and offers enormous benefits—lower costs, flexibility with hardware choices, and, if it comes with the right tools, a unified way to manage one's infrastructure: compute, storage, network.

The operational complexity and scalability challenges of storage management have been amongst the biggest pain points of virtualization adopters. So, we focused on Software-Defined Storage at the same time that a little-known Stanford graduate, named Martin Casado, was making the case for Software-Defined Networks.

Fast forward to March 2016: VMware's Virtual SAN is in its fourth release (version 6.2), the most important release of the product since its debut in March 2014. With a panoply of data integrity, availability and space efficiency features, Virtual SAN stands tall against the most sophisticated storage products in the industry. By combining commodity hardware economics and a flexible, highly efficient software stack, VMware is making all-flash storage affordable and applicable to a broad range of use cases. With thousands of enterprise customers using VSAN for business-critical use cases that range from Hospitals to popular E-commerce platforms, and from oil rigs to aircraft carriers, Virtual SAN has the credentials to be deemed battle tested. And we proudly carry the scars that come with that!

However, if we focus entirely on the (impressive) storage features of Virtual SAN, we are missing an important point: the fundamental shift in the management paradigm that the product introduces. In the current instantiation of the technology, Virtual SAN is designed to be an ideal storage platform for hyper-converged infrastructures (HCI). The key benefit of HCI is its simple operational model that allows customers to manage the entire IT infrastructure with a single set of tools. Virtual SAN is blazing the path for VMware products and for the industry at large.

Since version 6.1, Virtual SAN has been offering a "Health Service" that monitors, probes and assists with remediation of hardware, configuration, and performance issues. With version 6.2, the management toolkit is extended with a "Performance Service", which provides advanced performance monitoring and analysis. Using these tools and data, the user has end-to-end visibility into the state of their infrastructure and the consumption of resources by different VMs and workloads. For example, one can do performance troubleshooting, pinpoint the root cause of any issues (whether compute, network or storage) and decide on remediation actions, while using a single tool set.

All management features of Virtual SAN including enablement, configuration, upgrades, as well as the health and performance services, are built using a scalable distributed architecture. They are fully supported through APIs, which are extensions of the very popular vSphere API. In the same spirit, all VSAN and HCI management features are natively integrated with the vSphere Web Client.

Virtual SAN is natively integrated with ESXi, the best hypervisor in the industry, to offer unmatched HCI efficiency and performance. The product works "out of the box" for vSphere customers. Moreover, the scale-out nature of its control plane opens the path for new use cases and applications in private and public clouds. We are just at the beginning of an exciting journey.

Much (digital) ink has been spilled describing Virtual SAN, its unique architecture, and the operational benefits it brings to customers. Not a day goes by without an article been published, without a friend or a foe expressing their opinion on the product and what it means for the customers. This book, however, is different. It is the most comprehensive and authoritative reference for Virtual SAN, the product and the technology behind it.

Duncan and Cormac have been on board since the early days of the project. They have contributed invaluable input during the design phases based on their personal experience and on customer feedback. Not only they are intimately familiar with the architecture of the product, but they also have extensive operational experience in the lab and with customer deployments around the world. As such, they approach the subject through the eyes of the IT professional.

I hope you find the book as informative as I have.

Christos Karamanolis
VMware Fellow. CTO, Storage and Availability Business Unit.

About the Authors

Cormac Hogan is a Senior Staff Engineer in the Office of the CTO in the Storage and Availability business unit at VMware. Cormac was one of the first VMware employees at the EMEA headquarters in Cork, Ireland, back in 2005, and has previously held roles in VMware's Technical Marketing, Integration Engineering and Support organizations. Cormac has written a number of storage-related white papers and has given numerous presentations on storage best practices and new features. Cormac is the owner of CormacHogan.com, a blog site dedicated to storage and virtualization.

He can be followed on twitter @CormacJHogan.

Duncan Epping is a Chief Technologist working for VMware in the Office of CTO of the Storage and Availability business unit. Duncan is responsible for ensuring VMware's future innovations align with essential customer needs, translating customer problems to opportunities, and function as the global lead evangelist for Storage and Availability. Duncan specializes in Software Defined Storage, hyper-converged infrastructures and business continuity/disaster recovery solutions. He has four patents pending and one granted on the topic of availability, storage and resource management. Duncan is the author/owner of VMware Virtualization blog Yellow-Bricks.com and has various books on the topic of VMware including the "vSphere Clustering Deepdive" series.

He can be followed on twitter @DuncanYB.

This page intentionally left blank

About the Technical Reviewers

Christian Dickmann is a Virtual SAN Architect and Sr. Staff Engineer in the Storage and Availability Business Unit at VMware. Besides system architecture, he currently focuses on management functionality for ease of use, troubleshooting, monitoring, installation, and lifecycle management. Since joining VMware in 2007, he worked on extensibility of the networking stack as well as built the vSphere R&D dev/test cloud from the ground up. The latter experience made him an avid user experience and customer advocate.

John Nicholson is a Senior Technical Marketing Manager in the Storage and Availability Business Unit. He focuses on delivering technical guidance around VMware Virtual SAN solutions. John previously worked for partners and customers in architecting and implementing enterprise storage and VDI solutions. He holds the VCP certification (VCP5-DT, VCP6-DCV) and is a member of the vExpert and Veeam Vanguard programs.

This page intentionally left blank

Acknowledgments

The authors of this book both work for VMware. The opinions expressed in the book are the authors' personal opinions and experience with the product. Statements made throughout the book do not necessarily reflect the views and opinions of VMware.

We would like to thank Christian Dickmann and John Nicholson for keeping us honest as our technical editors. Of course, we also want to thank the Virtual SAN engineering team for their help and patience. In particular, we want to call out a couple of individuals of the Storage and Availability BU, Christian Dickmann, Paudie O'Riordan and Christos Karamanolis, whose deep knowledge and understanding of VSAN, and storage technology in general, was leveraged throughout this book. We also want to thank William Lam for his contributions to the book.

Lastly, we want to thank our VMware management team (Yanbing Li, Christos Karamanolis) for supporting us on this and other projects.

Go VSAN!

Cormac Hogan and Duncan Epping

This page intentionally left blank

We Want to Hear from You!

As the reader of this book, *you* are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

We welcome your comments. You can email or write us directly to let us know what you did or didn't like about this book—as well as what we can do to make our books better.

Please note that we cannot help you with technical problems related to the topic of this book.

When you write, please be sure to include this book's title and author as well as your name, email address, and phone number. We will carefully review your comments and share them with the author and editors who worked on the book.

Email: VMwarePress@vmware.com

Mail: VMware Press

ATTN: Reader Feedback

800 East 96th Street

Indianapolis, IN 46240 USA

Reader Services

Register your copy of *Essential Virtual SAN (VSAN)* at www.informit.com for convenient access to downloads, updates, and corrections as they become available. To start the registration process, go to informit.com/register and log in or create an account*. Enter the product ISBN, 9780133854992, and click Submit. Once the process is complete, you will find any available bonus content under Registered Products.

*Be sure to check the box that you would like to hear from us in order to receive exclusive discounts on future editions of this product.

Introduction

When talking about virtualization and the underlying infrastructure that it runs on, one component that always comes up in conversation is storage. The reason for this is fairly simple: In many environments, storage is a pain point. Although the storage landscape has changed with the introduction of flash technologies that mitigate many of the traditional storage issues, many organizations have not yet adopted these new architectures and are still running into the same challenges.

Storage challenges range from operational effort or complexity to performance problems or even availability constraints. The majority of these problems stem from the same fundamental problem: legacy architecture. The reason is that most storage platform architectures were developed long before virtualization existed, and virtualization changed the way these shared storage platforms were used.

In a way, you could say that virtualization forced the storage industry to look for new ways of building storage systems. Instead of having a single server connect to a single storage device (also known as a logical unit or LUN for short), virtualization typically entails having one (or many) physical server(s) running many virtual machines connecting to one or multiple storage devices. This did not only increase the load on these storage systems, it also changed the workload patterns and increased the total capacity required.

As you can imagine, for most storage administrators, this required a major shift in thinking. What should the size of my LUN be? What are my performance requirements, and how many spindles will that result in? What kind of data services are required on these LUNs, and where will virtual machines be stored? Not only did it require a major shift in thinking, but it also required working in tandem with other IT teams. Whereas in the past server admins and network and storage admins could all live in their own isolated worlds, they now needed to communicate and work together to ensure availability of the platform they were building. Whereas in the past a mistake, such as a misconfiguration or underprovisioning, would only impact a single server, it could now impact many virtual machines.

There was a fundamental shift in how we collectively thought about how to operate and architect IT infrastructures when virtualization was introduced. Now another collective shift is happening all over again. This time it is due to the introduction of software-defined networking and software-defined storage. But let's not let history repeat itself, and let's avoid the mistakes we all made when virtualization first arrived. Let's all have frank and open discussions with our fellow datacenter administrators as we all aim to revolutionize datacenter architecture and operations!

You, the Reader

This book is targeted at IT professionals who are involved in the care and feeding of a VMware vSphere environment. Ideally, you have been working with VMware vSphere for some time and perhaps you have attended an authorized course in vSphere, such as the "Install, Configure, and Manage" class. This book is not a starters guide, but there should be enough in the book for administrators and architects of all levels.

How to Use This Book

This book is split into ten chapters, as described here:

- **Chapter 1, "Introduction to VSAN":** This chapter provides a high-level introduction to software-defined storage and VSAN.
- **Chapter 2, "VSAN Prerequisites and Requirements for Deployment":** This chapter describes the requirements from a physical and virtual perspective to safely implement VSAN.
- **Chapter 3, "VSAN Installation and Configuration":** This chapter goes over the steps needed to install and configure VSAN.
- **Chapter 4, "VM Storage Policies on VSAN":** This chapter explains the concept of storage policy-based management.
- **Chapter 5, "Architectural Details":** This chapter provides in-depth architectural details of VSAN.
- **Chapter 6, "VM Storage Policies and Virtual Machine Provisioning":** This chapter describes how VM storage policies can be used to simplify VM deployment.
- **Chapter 7, "Management and Maintenance":** This chapter describes the steps for most common management and maintenance tasks.
- **Chapter 8, "Stretched Clusters":** This chapter covers the operational and architectural aspects and design decisions around the introduction of VSAN stretched clusters.
- **Chapter 9, "Designing a VSAN Cluster":** This chapter provides various examples around designing a VSAN cluster, including sizing exercises.
- **Chapter 10, "Troubleshooting, Monitoring, and Performance":** This chapter covers the various (command line) tools available to troubleshoot and monitor VSAN.

This page intentionally left blank

Chapter 4

VM Storage Policies on VSAN

In vSphere 5.0, VMware introduced a feature called profile-driven storage. Profile-driven storage is a feature that allows vSphere administrators to easily select the correct datastore on which to deploy virtual machines (VMs). The selection of the datastore is based on the capabilities of that datastore, or to be more specific, the underlying capabilities of the storage array that have been assigned to this datastore. Examples of the capabilities are RAID level, thin provisioning, deduplication, encryption, replication, etc. The capabilities are completely dependent on the storage array.

Throughout the life cycle of the VM, profile-driven storage allows the administrator to check whether its underlying storage is still *compatible*. In other words, does the datastore on which the VM resides still have the correct capabilities for this VM? The reason why this is useful is because if the VM is migrated to a different datastore for whatever reason, the administrator can ensure that it has moved to a datastore that continues to meet its requirements. If the VM is migrated to a datastore without paying attention to the capabilities of the destination storage, the administrator can still check the compliance of the VM storage from the vSphere client at any time and take corrective actions if it no longer resides on a datastore that meets its storage requirements (in other words, move it back to a compliant datastore).

However, VM storage policies and storage policy-based management (SPBM) have taken this a step further. In the previous paragraph, we described a sort of storage quality of service driven by the storage. All VMs residing on the same datastore would inherit the capabilities of the datastore. With VSAN, the storage quality of service no longer resides with the datastore; instead, it resides with the VM and is enforced by the VM storage policy associated with the VM and the VM disks (VMDKs). Once the policy is pushed down to the storage layer, in this case VSAN, the underlying storage is then responsible for creating storage for the VM that meets the requirements placed in the policy.

Introducing Storage Policy-Based Management in a VSAN Environment

VSAN leverages this approach to VM deployment, using an updated method called storage policy-based management (SPBM). All VMs deployed to a VSAN datastore must use a VM storage policy, although if one is not specifically created, a default one that is associated with the datastore is assigned to the VM. The VM storage policy contains one or more VSAN capabilities. This chapter will describe the VSAN capabilities. After the VSAN cluster has been configured and the VSAN datastore has been created, VSAN surfaces up a set of capabilities to the vCenter Server. These capabilities, which are surfaced by the vSphere APIs for Storage Awareness (VASA) storage provider (more on this shortly) when the cluster is configured successfully, are used to set the availability, capacity, and performance policies on a per-VM (and per-VMDK) basis when that VM is deployed on the VSAN datastore.

As previously mentioned, this differs significantly from the previous VM storage profile mechanism that we had in vSphere in the past. With the VM storage profile feature, the capabilities were associated with datastores, and were used for VM placement decisions. Now, through SPBM, administrators create a policy defining the storage requirements for the VM, and this policy is pushed out to the storage, which in turn instantiates per-VM (and per-VMDK) storage for virtual machines. In vSphere 6.0, VMware introduced Virtual Volumes (VVols). Storage policy-based management for VMs using VVols is very similar to storage policy-based management for VMs deployed on VSAN. In other words, administrators no longer need to carve up logical unit numbers (LUNs) or volumes for virtual machine storage. Instead, the underlying storage infrastructure instantiates the virtual machine storage based on the contents of the policy. What we have now with SPBM is a mechanism whereby we can specify the requirements of the VM, and the VMDKs. These requirements are then used to create a policy. This policy is then sent to the storage layer [in the case of VVols, this is a SAN or network-attached storage (NAS) storage array] asking it to build a storage object for this VM that meets these policy requirements. In fact, a VM can have multiple policies associated with it, different policies for different VMDKs.

By way of explaining capabilities, policies, and profiles, capabilities are what the underlying storage is capable of providing by way of availability, performance, and reliability. These capabilities are visible in vCenter Server. The capabilities are then used to create a VM storage policy (or just policy for short). A policy may contain one or more capabilities, and these capabilities reflect the requirements of your VM or application running in a VM. Previous versions of vSphere used the term *profiles*, but these are now known as *policies*.

Deploying VMs on a VSAN datastore is very different from previous approaches in vSphere. In the past, an administrator would present a LUN or volume to a group of ESXi hosts and in the case of block storage partition, format, and build a VMFS file system to create a datastore for storing VM files. In the case of network-attached storage (NAS), a network file system (NFS) volume is mounted to the ESXi host, and once again a VM

is created on the datastore. There is no way to specify a RAID-0 stripe width for these VMDKs, nor is there any way to specify a RAID-1 replica for the VMDK.

In the case of VSAN (and now VVols), the approach to deploying VMs is quite different. Consideration must be given to the availability, performance, and reliability factors of the application running in the VM. Based on these requirements, an appropriate VM storage policy must be created and associated with the VM during deployment.

There were five capabilities in the initial release of VSAN, as illustrated in Figure 4.1.

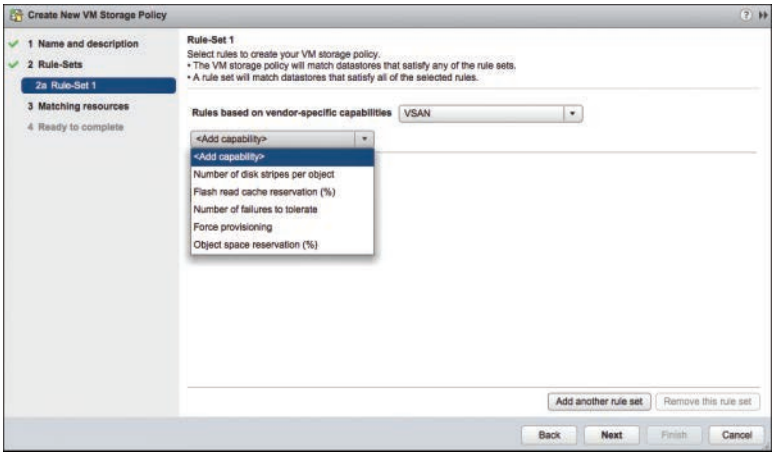

Figure 4.1 VSAN capabilities that can be used for VM storage policies

In VSAN 6.2, the number of capabilities is increased to support a number of new features. These features include the ability to implement RAID-5 and RAID-6 configurations for virtual machine objects deployed on an all-flash VSAN configuration, alongside the existing RAID-0 and RAID-1 configurations. With RAID-5 and RAID-6, it now allows VMs to tolerate one or two failures, but it means that the amount of space consumed is much less than a RAID-1 configuration to tolerate a similar amount of failures. There is also a new policy for software checksum. Checksum is enabled by default, but it can be disabled through policies if an administrator wishes to disable it. The last capability relates to quality of service and provides the ability to limit the number of input/output operations per second (IOPS) for a particular object.

You can select the capabilities when a VM storage policy is created. Note that certain capabilities are applicable to hybrid VSAN configurations (e.g., flash read cache reservation), while other capabilities are applicable to all-flash VSAN configurations (e.g., failure tolerance method set to performance).

VM storage policies are essential in VSAN deployments because they define how a VM is deployed on a VSAN datastore. Using VM storage policies, you can define the capabilities that can provide the number of VMDK RAID-0 stripe components or the number of

RAID-1 mirror copies of a VMDK. If an administrator desires a VM to tolerate one failure but does not want to consume as much capacity as a RAID-1 mirror, a RAID-5 configuration can be used. This requires a minimum of four hosts in the cluster and implements a distributed parity mechanism across the storage of all four hosts. If this configuration would be implemented with RAID-1, the amount of capacity consumed would be 200% the size of the VMDK. If this is implemented with RAID-5, the amount of capacity consumed would be 133% the size of the VMDK.

Similarly, if an administrator desires a VM to tolerate two failures using a RAID-1 mirroring configuration, there would need to be three copies of the VMDK, meaning the amount of capacity consumed would be 300% the size of the VMDK. With a RAID-6 implementation, a double parity is implemented, which is also distributed across all the hosts. For RAID-6, there must be a minimum of six hosts in the cluster. RAID-6 also allows a VM to tolerate two failures, but only consumes capacity equivalent to 150% the size of the VMDK.

Figure 4.2 shows the new policies introduced in VSAN 6.2.

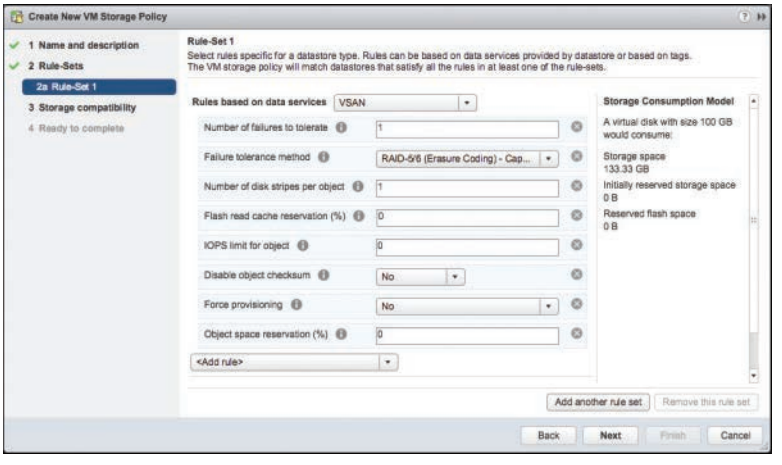

Figure 4.2 New VSAN capabilities

The sections that follow highlight where you should use these capabilities when creating a VM storage policy and when to tune these values to something other than the default. Remember that a VM storage policy will contain one or more capabilities.

In the initial release of VSAN, five capabilities were available for selection to be part of the VM storage policy. In VSAN 6.2, as previously highlighted, additional policies were introduced. As an administrator, you can decide which of these capabilities can be added to the policy, but this is, of course, dependent on the requirements of your VM. For example, what performance and availability requirements does the VM have? The capabilities are as follows:

- Number of failures to tolerate
- Number of disk stripes per object

- Failure tolerance method
- IOPS limit for object
- Disable object checksum
- Flash read cache reservation (hybrid configurations only)
- Object space reservation
- Force provisioning

The sections that follow describe the VSAN capabilities in detail.

Number of Failures to Tolerate

In this section, *number of failures to tolerate* is described having *failure tolerance method* set to its default value that is *Performance*. Later on we will describe a different scenario when *failure tolerance method* is set to *Capacity*.

This capability sets a requirement on the storage object to tolerate at least *n* number of failures in the cluster. This is the number of concurrent host, network, or disk failures that may occur in the cluster and still ensure the availability of the object. When the *failure tolerance method* is set to its default value of *RAID-1*, the VM's storage objects are mirrored; however, the mirroring is done across ESXi hosts, as shown in Figure 4.3.

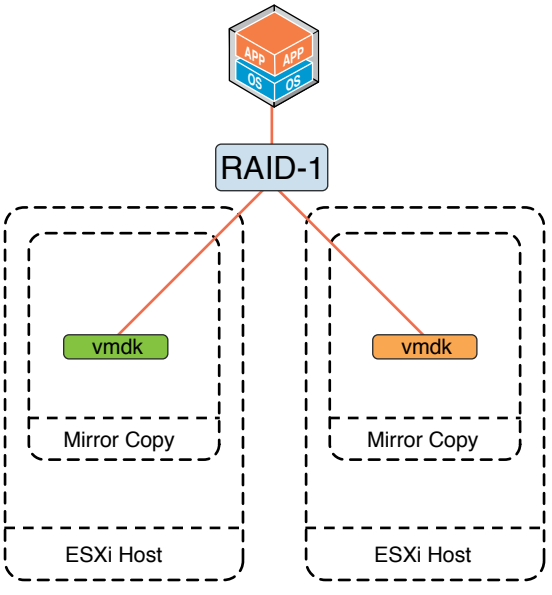

Figure 4.3 Number of failures to tolerate results in a RAID-1 configuration

When this capability is set to a value of *n*, it specifies that the VSAN configuration must contain at least *n* + 1 replicas (copies of the data); this also implies that there are 2*n* + 1 hosts in the cluster.

Note that this requirement will create a configuration for the VM objects that may also contain an additional number of witness components being instantiated to ensure that the VM remains available even in the presence of up to *number of failures to tolerate* concurrent failures (see Table 4.1). Witnesses provide a quorum when failures occur in the cluster or a decision has to be made when a split-brain situation arises. These witnesses will be discussed in much greater detail later in the book, but suffice it to say that witness components play an integral part in maintaining VM availability during failures and maintenance tasks.

One aspect worth noting is that any disk failure on a single host is treated as a "failure" for this metric (although multiple disk failures on the same host are also treated as a single host failure). Therefore, the VM may not persist (remain accessible) if there is a disk failure on host A and a host failure of host B when number of failures to tolerate is set to 1.

Table 4.1 Witness and hosts required to meet number of failures to tolerate requirement

Number of Failures to Tolerate	Mirror Copies/ Replicas	Witness Objects	Minimum Number of ESXi Hosts
0	1	0	1
1	2	1	3
2	3	2	5
3	4	3	7

Table 4.1 is true if the number of disk objects to stripe is set to 1. The behavior is subtly different if there is a stripe width greater than 1. The number of disk stripes per object will be discussed in more detail shortly.

If no policy is chosen when a VM is deployed, the default policy associated with the VSAN datastore is chosen, which in turn sets the number of failures to tolerate to 1. When a new policy is created, the default value of number of failures to tolerate is also 1. This means that even if this capability is not explicitly specified in the policy, it is implied.

Failure Tolerance Method

This is a new capability introduced in VSAN 6.2 and is how administrators can choose either RAID-1 or RAID-5/6 configuration for their virtual machine objects. The *failure tolerance method* is used in conjunction with *number of failures to tolerate*. The purpose of this

setting is to allow administrators to choose between performance and capacity. If performance is the absolute end goal for administrators, then RAID-1 (which is still the default) is the tolerance method that should be used. If administrators do not need maximum performance and are more concerned with capacity usage, then RAID-5/6 is the tolerance method that should be used. The easiest way to explain the behavior is to display the various policy settings and the resulting object configuration, as shown in Table 4.2.

As can be seen from Table 4.2, when the *failure tolerance method* RAID5/6 is selected, either RAID-5 or RAID-6 is implemented depending on the number of failures that you wish to tolerate (although it only supports a maximum setting of two for *number of failures to tolerate*). If performance is still the desired capability, then the traditional RAID-1 configuration is implemented, with the understanding that this uses mirror copies of the objects, and thus consumes significantly more space.

One might ask why RAID-5/6 less performing than RAID-1. The reason lies in I/O amplification. In steady state, where there are no failures in the cluster, there is no read amplification when using RAID-5/6 versus RAID-1. However, there is write amplification. This is because the parity component needs to be updated every time there is a write to the associated data components. So in the case of RAID-5, we need to read the component that is going to be updated with additional write data, read the current parity, merge the new write data with the current data, write this back, calculate the new parity value, and write this back also. In essence, a single write operation can amplify into two reads and two

Table 4.2 Object configuration when number of failures to tolerate and failure tolerance method are set

Number of Failures to Tolerate	Failure Tolerance Method	Object Configuration	Number of ESXi Hosts Required
0	RAID5/6 (Erasure Coding)	RAID-0	1
0	RAID-1 (mirroring)	RAID-0	1
1	RAID5/6 (Erasure Coding)	RAID-5	4
1	RAID-1 (mirroring)	RAID-1	3
2	RAID5/6 (Erasure Coding)	RAID-6	6
2	RAID-1 (mirroring)	RAID-1	5
3	RAID5/6 (Erasure Coding)	N/A	N/A
3	RAID-1 (mirroring)	RAID-1	7

writes. With RAID-6, which has double parity, a single write can amplify into three reads and three writes.

And indeed, when there is a failure of some component in the RAID-5 and RAID-6 objects, and data needs to be determined using parity, then the I/O amplification is even higher. These are the considerations an administrator needs to evaluate when deciding on the *failure tolerance method*.

One item to keep in mind is that even though *failure tolerance method* set to RAID5/6 consumes less capacity, it does require more hosts than the traditional RAID-1 approach and is only supported (in this release) on an all-flash VSAN configuration. When using RAID-1, the rule is that to tolerate n failures, there must be $2n + 1$ hosts for the mirrors/replicas and witness. So to tolerate one failure, there must be three hosts; to tolerate two failures, there must be five hosts; or to tolerate three failures, there must be seven hosts in the cluster, all contributing storage to the VSAN datastore. With *failure tolerance method* set to RAID5/6, four hosts are needed to tolerate one failure and six hosts are needed to tolerate two failures, even though less space is consumed on each host. Figure 4.4 shows an example of a RAID-5 configuration for an object, deployed across four hosts with a distributed parity.

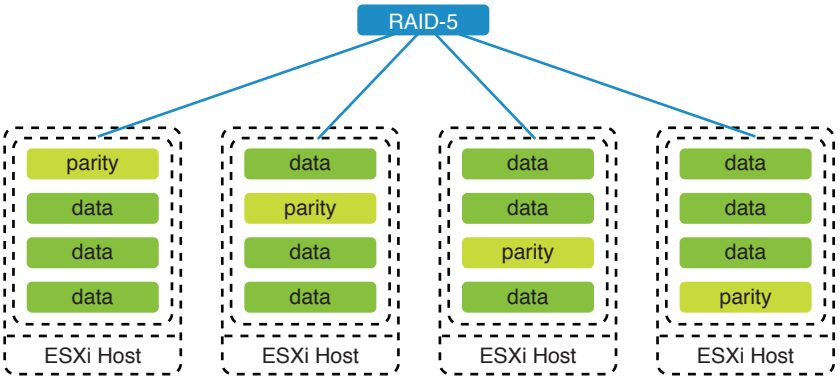

Figure 4.4 RAID-5 configuration, a result of *failure tolerance method* RAID5/6 and number of failures to tolerate set to 1

The RAID-5 or RAID-6 configurations also work with *number of disk stripes per object*. If stripe width is also specified as part of the policy along with *failure tolerance method* set to RAID5/6 each of the components on each host is striped in a RAID-0 configuration, and these are in turn placed in either a RAID-5 or-6 configuration.

One final note is in relation to having a number of failures to tolerate setting of zero or three. If you deploy a VM with this policy setting, which includes a *failure tolerance method*

RAID5/6 setting, the VM provisioning wizard will display a warning stating that this policy setting is only effective when the number of failures to tolerate is set to either one or two. You can still proceed with the deployment, but the object is deployed as a single RAID-0 object.

Number of Disk Stripes Per Object

This capability defines the number of physical disks across which each replica of a storage object (e.g., VMDK) is striped. When *failure tolerance method* is set to performance, this policy setting can be considered in the context of a RAID-0 configuration on each RAID-1 mirror/replica where I/O traverses a number of physical disk spindles. When *failure tolerance method* is set to capacity, each component of the RAID-5 or RAID-6 stripe may also be configured as a RAID-0 stripe. Typically, when the number of disk stripes per object is defined, the number of failures to tolerate is also defined. Figure 4.5 shows what a combination of these two capabilities could result in, once again assuming that the new VSAN 6.2 policy setting of *failure tolerance method* is set to its default value *RAID-1*.

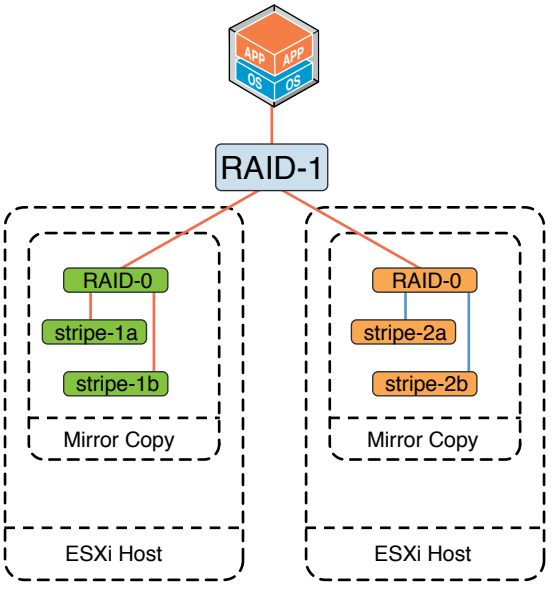

Figure 4.5 Storage object configuration when stripe width set is to 2 and failures to tolerate is set to 1 and replication method optimizes for is not set

To understand the impact of stripe width, let's examine it first in the context of write operations and then in the context of read operations.

Because all writes go to the cache device write buffer, the value of an increased stripe width may or may not improve performance. This is because there is no guarantee that the new stripe will use a different cache device; the new stripe may be placed on a capacity device in the same disk group, and thus the new stripe will use the same cache device. If the new stripe is placed in a different disk group, either on the same host or on a different host, and thus leverages a different cache device, performance might improve. However, you as the vSphere administrator have no control over this behavior. The only occasion where an increased stripe width could definitely add value is when there is a large amount of data to destage from the cache tier to the capacity tier. In this case, having a stripe could improve destage performance.

From a read perspective, an increased stripe width will help when you are experiencing many read cache misses, but note that this is a consideration in hybrid configurations only. All-flash VSAN considerations do not have a read cache. Consider the example of a VM deployed on a hybrid VSAN consuming 2,000 read operations per second and experiencing a hit rate of 90%. In this case, there are still 200 read operations that need to be serviced from magnetic disk in the capacity tier. If we make the assumption that a single magnetic disk can provide 150 input/output operations per second (IOPS), then it is obvious that it is not able to service all of those read operations, so an increase in stripe width would help on this occasion to meet the VM I/O requirements. In an all-flash VSAN, which is extremely read intensive, striping across multiple capacity flash devices can also improve performance.

In general, the default stripe width of 1 should meet most, if not all VM workloads. Stripe width is a capability that should change only when write destaging or read cache misses are identified as a performance constraint.

IOPS Limit for Object

IOPS limit for object is a new Quality of Service (QoS) capability introduced with VSAN 6.2. This allows administrators to ensure that an object, such as a VMDK, does not generate more than a predefined number of I/O operations per second. This is a great way of ensuring that a "noisy neighbor" virtual machine does not impact other virtual machine components in the same disk group by consuming more than its fair share of resources. By default, VSAN uses an I/O size of 32 KB as a base. **This means that a 64 KB I/O will therefore represent two I/O operations in the limits calculation**. I/Os that are less than or equal to 32 KB will be considered single I/O operations. For example, 2 × 4 KB I/Os are considered as two distinct I/Os. It should also be noted that both read and write IOPS are regarded as equivalent. Neither cache hit rate nor sequential I/O are taken into account. If the IOPS limit threshold is passed, the I/O is throttled back to bring the IOPS value back under the threshold. The default value for this capability is 0, meaning that there is no IOPS limit threshold and VMs can consume as many IOPS as they want, subject to available resources.

Flash Read Cache Reservation

This capability is applicable to hybrid VSAN configurations only. It is the amount of flash capacity reserved on the cache tier device as read cache for the storage object. It is specified as a percentage of the logical size of the storage object (i.e., VMDK). This is specified as a percentage value (%), with up to four decimal places. This fine granular unit size is needed so that administrators can express sub 1% units. Take the example of a 1 TB VMDK. If you limited the read cache reservation to 1% increments, this would mean cache reservations in increments of 10 GB, which in most cases is far too much for a single VM.

Note that you do not have to set a reservation to allow a storage object to use cache. All VMs equally share the read cache of cache devices. The reservation should be left unset (default) unless you are trying to solve a real performance problem and you believe dedicating read cache is the solution. If you add this capability to the VM storage policy and set it to a value 0 (zero), however, you will not have any read cache reserved to the VM that uses this policy. In the current version of VSAN, there is no proportional share mechanism for this resource when multiple VMs are consuming read cache, so every VM consuming read cache will share it equally.

Object Space Reservation

All objects deployed on VSAN are thinly provisioned. This means that no space is reserved at VM deployment time, but rather space is consumed as the VM uses storage. The object space reservation capability defines the percentage of the logical size of the VM storage object that may be reserved during initialization. The object space reservation is the amount of space to reserve specified as a percentage of the total object address space. This is a property used for specifying a thick provisioned storage object. If object space reservation is set to 100%, all of the storage capacity requirements of the VM storage are reserved up front (thick). This will be lazy zeroed thick (LZT) format and not eager zeroed thick (EZT). The difference between LZT and EZT is that EZT virtual disks are zeroed out at creation time; LZT virtual disks are zeroed out gradually at first write time.

One thing to bring to the readers' attention is the special case of using object space reservation when deduplication and compression are enabled on the VSAN cluster. When deduplication and compression are enabled, any objects that wish to use object space reservation in a policy must have it set to either 0% (no space reservation) or 100% (fully reserved). Values between 1% and 99% are not allowed. Any existing objects that have object space reservation between 1% and 99% will need to be reconfigured with 0% or 100% prior to enabling deduplication and compression on the cluster.

Force Provisioning

If the force provisioning parameter is set to a nonzero value, the object that has this setting in its policy will be provisioned even if the requirements specified in the VM storage

policy cannot be satisfied by the VSAN datastore. The VM will be shown as noncompliant in the VM summary tab and relevant VM storage policy views in the vSphere client. If there is not enough space in the cluster to satisfy the reservation requirements of at least one replica, however, the provisioning will fail even if force provisioning is turned on. When additional resources become available in the cluster, VSAN will bring this object to a compliant state.

One thing that might not be well understood regarding *force provisioning* is that if a policy cannot be met, it attempts a much simpler placement with requirements which reduces to *number of failures to tolerate* to 0, *number of disk stripes per object* to 1, and *flash read cache reservation* to 0 (on hybrid configurations). This means Virtual SAN will attempt to create an object with just a single copy of data. Any *object space reservation (OSR)* policy setting is still honored. Therefore there is no gradual reduction in capabilities as VSAN tries to find a placement for an object. For example, if policy contains *number of failures to tolerate* = 2, VSAN won't attempt an object placement using *number of failures to tolerate* = 1. Instead, it immediately looks to implement *number of failures to tolerate* = 0.

Similarly, if the requirement was *number of failures to tolerate* = 1, number of *disk stripes per* object = 4, but Virtual SAN doesn't have enough capacity devices to accommodate *number of disk stripes per object* = 4, then it will fall back to *number of failures to tolerate* = 0, *number of disk stripes per object* = 1, even though a policy of *number of failures to tolerate* = 1, *number of disk stripes per object* = 2 or *number of failures to tolerate* = 1, *number of disk stripes per object* = 3 may have succeeded.

Caution should be exercised if this policy setting is implemented. Since this allows VMs to be provisioned with no protection, it can lead to scenarios where VMs and data are at risk.

Administrators who use this option to force provision virtual machines need to be aware that although virtual machine objects may be provisioned with only one replica copy (perhaps due to lack of space), once additional resources become available in the cluster, VSAN may immediately consume these resources to try to satisfy the policy settings of virtual machines.

Some commonly used cases where force provisioning is used are (a) when boot-strapping a VSAN management cluster, starting with a single node that will host the vCenter Server, which is then used to configure a larger VSAN cluster, and (b) when allowing the provisioning of virtual machine/desktops when a cluster is under maintenance, such as a virtual desktop infrastructure (VDI) running on VSAN.

Remember that this parameter should be used only when *absolutely* needed and as an exception. When used by default, this could easily lead to scenarios where VMs, and all data associated with it, are at risk. Use with caution!

Disable Object Checksum

VSAN 6.2 introduced this new capability. This feature, which is enabled by default, is looking for data corruption (bit rot), and if found, automatically corrects it. Checksum is validated on the complete I/O path, which means that when writing data the checksum is calculated and automatically stored. Upon a read the checksum of the data is validated, and if there is a mismatch, the data is repaired. VSAN 6.2 also includes a scrubber mechanism. This mechanism is configured to run once a year (by default) to check all data on the VSAN datastore; however, this value can be changed by setting an advanced host setting. We recommend leaving this configured to the default value of once a year. In some cases you may desire to disable checksums completely. The reason for this could be performance, although the overhead is negligible and most customers prefer data integrity over a 1% to 3% performance increase. However in some cases, this performance increase may be desired. Another reason for disabling checksums is in the situation where the application already provides a checksum mechanism, or the workload does not require checksum. If that is the case, then checksums can be disabled through the "disable object checksum capability," which should be set to "Yes" to disable it.

That completes the capabilities overview. Let's now look at some other aspects of the storage policy-based management mechanism.

VASA Vendor Provider

As part of the VSAN cluster creation step, each ESXi host has a VSAN storage provider registered with vCenter. This uses the vSphere APIs for Storage Awareness (VASA) to surface up the VSAN capabilities to the vCenter Server. The capabilities can then be used to create VM storage policies for the VMs deployed on the VSAN datastore. If you are familiar with VASA and have used it with traditional storage environments, you'll find this functionality familiar; however, with traditional storage environments that leverage VASA, some configuration work needs to be done to add the storage provider for that particular storage. In the context of VSAN, a vSphere administrator does not need to worry about registering these; these are automatically registered when a VSAN cluster is created.

An Introduction to VASA

VASA allows storage vendors to publish the capabilities of their storage to vCenter Server, which in turn can display these capabilities in the vSphere Web Client. VASA may also provide information about storage health status, configuration info, capacity and thin provisioning info, and so on. VASA enables VMware to have an end-to-end story regarding storage. Traditionally, this enabled storage arrays to inform the VASA storage provider of capabilities, and then the storage provider informed vCenter Server,

so now users can see storage array capabilities from vSphere Web Client. Through VM storage policies, these storage capabilities are used in the vSphere Web Client to assist administrators in choosing the right storage in terms of space, performance, and service-level agreement (SLA) requirements. This was true for both traditional storage arrays, and now it is true for VSAN also. Prior to the release of virtual volumes (VVols), there was a notable difference in workflow when using VASA and VM storage policies when comparing traditional storage to VSAN. With traditional storage, VASA historically surfaced information about the datastore capabilities, and a vSphere administrator had to choose the appropriate storage on which to place the VM. With VSAN, and now VVols, you define the capabilities you want to have for your VM storage in a VM storage policy. This policy information is then pushed down to the storage layer, basically informing it that these are the requirements you have for storage. VASA will then tell you whether the underlying storage (e.g., VSAN) can meet these requirements, effectively communicating compliance information on a per-storage object basis. The major difference is that this functionality is now working in a bidirectional mode. Previously, VASA would just surface up capabilities. Now it not only surfaces up capabilities but also verifies whether a VM's storage requirements are being met based on the contents of the policy.

Storage Providers

Figure 4.6 illustrates an example of what the storage provider looks like. When a VSAN cluster is created, the VASA storage provider from every ESXi host in the cluster is registered to the vCenter Server. In a four-node VSAN cluster, the VASA VSAN storage provider configuration would look similar to this.

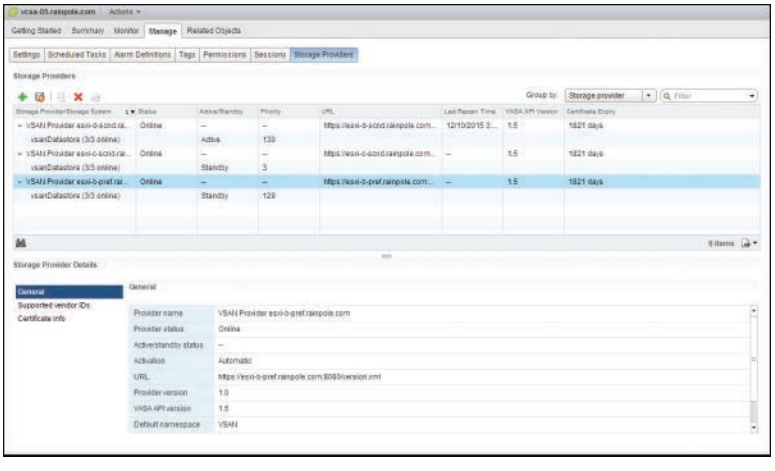

Figure 4.6 VSAN storage providers, added when the VSAN cluster is created

You can always check the status of the storage providers by navigating in the Web Client to the vCenter Server inventory item, selecting the **Manage** tab and then the **Storage Providers** view. One VSAN provider should always be online. The other storage providers should be in standby mode. This is all done automatically by VSAN. There is typically no management of the VASA providers required by administrators.

In VSAN clusters that have more than eight ESXi hosts, and thus more than eight VASA storage providers, the list of storage providers is shortened to eight in the user interface (UI) for display purposes. The number of standby storage providers is still displayed correctly; you simply won't be able to interrogate them.

VSAN Storage Providers: Highly Available

You might ask why every ESXi host registers this storage provider. The reason for this is high availability. Should one ESXi host fail, another ESXi host in the cluster can take over the presentation of these VSAN capabilities. If you examine the storage providers shown in Figure 4.6, you will see that only one of the VSAN providers is online. The other storage providers from the other two ESXi hosts in this three-node cluster are in a standby state. Should the storage provider that is currently active go offline or fail for whatever reason (most likely because of a host failure), one of the standby providers will be promoted to active.

There is very little work that a vSphere administrator needs to do with storage providers to create a VSAN cluster. This is simply for your own reference. However, if you do run into a situation where the VSAN capabilities are not surfacing up in the VM storage policies section, it is worth visiting this part of the configuration and verifying that at least one of the storage providers is active. If you have no active storage providers, you will not discover any VSAN capabilities when trying to build a VM storage policy. At this point, as a troubleshooting step, you could consider doing a refresh of the storage providers by clicking on the refresh icon (orange circular arrows) in the storage provider screen.

What should be noted is that the VASA storage providers do not play any role in the data path for VSAN. If storage providers fail, this has no impact on VMs running on the VSAN datastore. The impact of not having a storage provider is lack of visibility into the underlying capabilities, so you will not be able to create new storage policies. However, already running VMs and policies are unaffected.

Changing VM Storage Policy On-the-Fly

Being able to change a VM storage policy on-the-fly is quite a unique aspect of VSAN. We will use an example to explain the concept of how you can change a VM storage policy

on-the-fly and how it changes the layout of a VM without impacting the application or the guest operating system running in the VM.

Consider the following scenario, briefly mentioned earlier in the context of stripe width. A vSphere administrator has deployed a VM on a hybrid VSAN configuration with the default VM storage policy, which is that the VM storage objects should have no disk striping and should tolerate one failure. The layout of the VM disk file would look something like Figure 4.7.

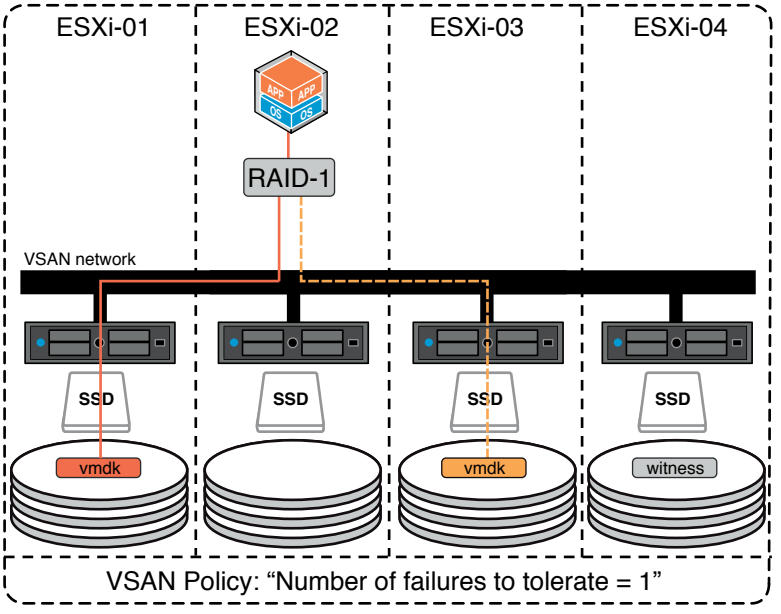

Figure 4.7 VSAN policy with the capability number of failures to tolerate = 1

The VM and its associated applications initially appeared to perform satisfactorily with a 100% cache hit rate; however, over time, an increasing number of VMs were added to the VSAN cluster. The vSphere administrator starts to notice that the VM deployed on VSAN is getting a 90% read cache hit rate. This implies that 10% of reads need to be serviced from magnetic disk/capacity tier. At peak time, this VM is doing 2,000 read operations per second. Therefore, there are 200 reads that need to be serviced from magnetic disk (the 10% of reads that are cache misses). The specifications on the magnetic disks imply that each disk can do 150 IOPS, meaning that a single disk cannot service these additional 200 IOPS. To meet the I/O requirements of the VM, the vSphere administrator correctly decides to create a RAID-0 stripe across two disks.

On VSAN, the vSphere administrator has two options to address this.

The first option is to simply modify the VM storage policy currently associated with the VM and add a stripe width requirement to the policy; however, this would change the storage layout of all the other VMs using this same policy.

Another approach is to create a brand-new policy that is identical to the previous policy but has an additional capability for stripe width. This new policy can then be attached to the VM (and VMDKs) suffering from cache misses. Once the new policy is associated with the VM, the administrator can synchronize the new/updated policy with the VM. This can be done immediately, or can be deferred to a maintenance window if necessary. If it is deferred, the VM is shown as noncompliant with its new policy. When the policy change is implemented, VSAN takes care of changing the underlying VM storage layout required to meet the new policy, *while the VM is still running* without the loss of any failure protection. It does this by mirroring the new storage objects with the additional components (in this case additional RAID-0 stripe width) to the original storage objects.

As seen, the workflow to change the VM storage policy can be done in two ways; either the original current VM storage policy can be edited to include the new capability of a stripe width = 2 or a new VM storage policy can be created that contains the failures to tolerate = 1 and stripe width = 2. The latter is probably more desirable because you may have other VMs using the original policy, and editing that policy will affect all VMs using it. When the new policy is created, this can be associated with the VM and the storage objects in a number of places in the vSphere Web Client. In fact, policies can be changed at the granularity of individual VM storage objects (e.g., VMDK) if necessary.

After making the change, the new components reflecting the new configuration (e.g., a RAID-0 stripe) will enter a state of reconfiguring. This will temporarily build out additional replicas or components, in addition to keeping the original replicas/components, so additional space will be needed on the VSAN datastore to accommodate this on-the-fly change. When the new replicas or components are ready and the configuration is completed, the original replicas/components are discarded.

Note that not all policy changes require the creation of new replicas or components. For example, adding an IOPS limit, or reducing the number of failures to tolerate, or reducing space reservation does not require this. However, in many cases, policy changes will trigger the creation of new replicas or components.

Your VM storage objects may now reflect the changes in the Web Client, for example, a RAID-0 stripe as well as a RAID-1 replica configuration, as shown in Figure 4.8.

Compare this to the tasks you may have to perform on many traditional storage arrays to achieve this. It would involve, at the very least, the following:

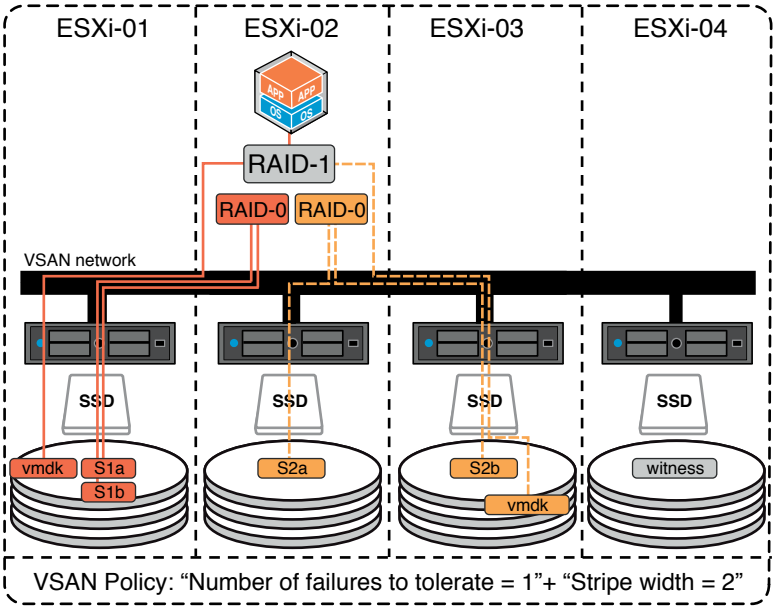

Figure 4.8 VSAN RAID-0 and RAID-1 configuration

- The migration of VMs from the original datastore.
- The decommissioning of the said LUN/volume.
- The creation of a new LUN with the new storage requirements (different RAID level).
- Possibly the reformatting of the LUN with VMFS in the case of block storage.
- Finally, you have to migrate your VMs back to the new datastore.

In the case of VSAN, after the new storage replicas or components have been created and synchronized, the older storage replicas and/or components will be automatically removed. Note that VSAN is capable of striping across disks, disk groups, and hosts when required, as depicted in Figure 4.8, where stripes S1a and S1b are located on the same host but stripes S2a and S2b are located on different hosts. It should also be noted that VSAN can create the new replicas or components without the need to move any data between hosts; in many cases the new components can be instantiated on the same storage on the same host.

We have not shown that there are, of course, additional witness components that could be created with such a change to the configuration. For a VM to continue to access all its components, a full replica copy of the data must be available and more than 50% of the components (votes) of that object must also be available in the cluster. Therefore, changes to the VM storage policy could result in additional witness components being created, or indeed, in the case of introducing a policy with less requirements, there could be fewer witnesses.

You can actually see the configuration changes taking place in the vSphere UI during this process. Select the VM that is being changed, click its **manage** tab, and then choose the **VM storage policies** view, as shown in Figure 4.9. Although this view does not show all the VM storage objects, it does display the VM home namespace, and the VMDKs are visible.

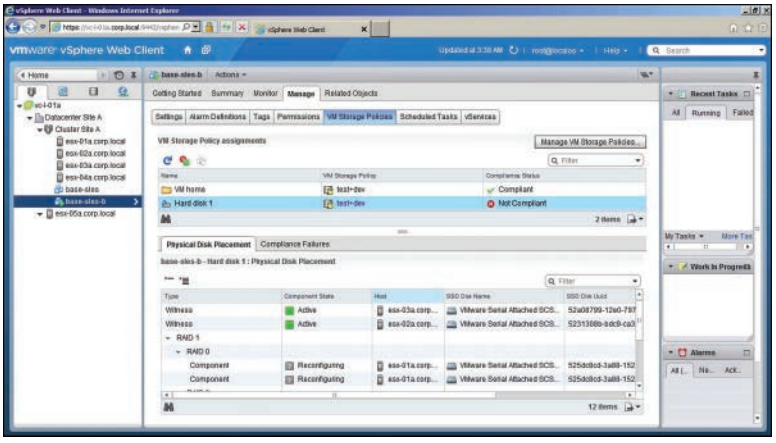

Figure 4.9 VM Storage Policy view in the vSphere client showing component reconfiguration

In VSAN 6.0, there is also a way to examine all resyncing components. Select the VSAN cluster object in the vCenter inventory, then select monitor, Virtual SAN, and finally "resyncing components" in the menu. This will display all components that are currently resyncing/rebuilding. Figure 4.10 shows the resyncing dashboard view, albeit without any resyncing activity taking place.

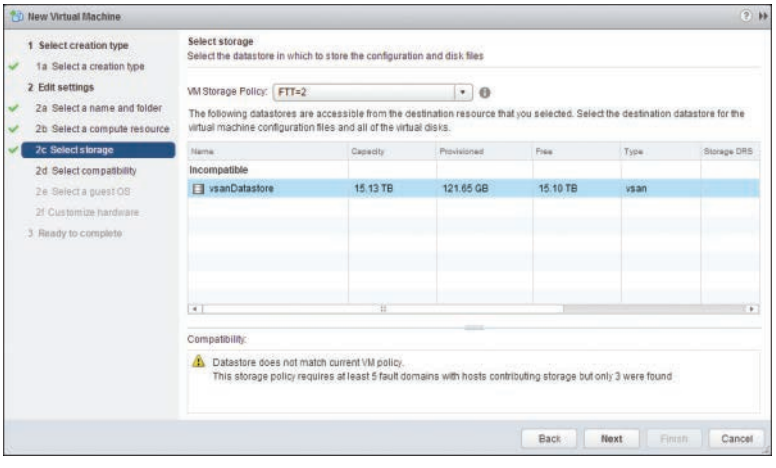

Figure 4.10 Resyncing activity as seen from the vSphere Web Client

Objects, Components, and Witnesses

A number of new concepts have been introduced in this chapter so far, including some new terminology. Chapter 5, "Architectural Details," covers in greater detail objects, components, and indeed witness disks, as well as which VM storage objects are impacted by a particular capability in the VM storage policy. For the moment, it is enough to understand that on VSAN, a VM is no longer represented by a set of files but rather a set of storage objects. There are five types of storage objects:

- VM home namespace
- VMDKs
- VM swap
- Snapshot delta disks
- Snapshot memory

Although the vSphere Web Client displays only the VM home namespace and the VMDKs (hard disks) in the VM > monitor > policies > physical disk placement, snapshot deltas and VM swap can be viewed in the cluster > monitor > Virtual SAN > virtual disks view. We will also show ways of looking at detailed views of all the storage objects, namely delta and VM swap, in Chapter 10, "Troubleshooting, Monitoring, and Performance," when we look at various monitoring tools available to VSAN.

VM Storage Policies

VM storage policies work in an identical fashion to storage profiles introduced in vSphere 5.0, insofar as you simply build a policy containing your VM provisioning requirements. There is a major difference in how storage policies work when compared to the original storage profiles feature. With storage profiles, you simply used the requirements in the policy to select an appropriate datastore when provisioning the VM. The storage policies not only select the appropriate datastore, but also inform the underlying storage layer that there are also certain availability and performance requirements associated with this VM. So while the VSAN datastore may be the destination datastore when the VM is provisioned with a VM storage policy, settings within the policy will stipulate additional requirements. For example, it may state that this VM has a requirement for a number of replica copies of the VM files for availability, a stripe width and read cache requirement for high performance, and a thin provisioning requirement.

VM storage policies are held inside VSAN, as well as being stored in the vCenter inventory database. Every object stores its policy inside its own metadata. This means that

vCenter is not required for VM storage policy enforcement. So if for some reason the vCenter Server is unavailable, policies can continue to be enforced.

Enabling VM Storage Policies

In the initial release of VSAN, VM storage policies could be enabled or disabled via the UI. This option is not available in later releases. However, VM storage policies are automatically enabled on a cluster when VSAN is enabled on the cluster. Although VM storage policies are normally only available with certain vSphere editions, a VSAN license will also provide this feature.

Creating VM Storage Policies

vSphere administrators have the ability to create multiple policies. As already mentioned, a number of VSAN capabilities are surfaced up by VASA related to availability and performance, and it is at this point that the administrator must decide what the requirements are for the applications running inside of the VMs from a performance and availability perspective. For example, how many component failures (hosts, network, and disk drives) does the administrator require this VM to tolerate and continue to function? Also, is the application running in this VM demanding from an IOPS perspective? If so, an adequate read cache should be provided as a possible requirement so that the performance requirement is met. Other considerations include whether the VM should be thinly provisioned or thickly provisioned, if RAID-5 or RAID-6 configurations are desired to save storage space, if checksum should be disabled, or if an IOPS limit is required for a particular VM to avoid a "noisy neighbor" situation.

Another point to note is that since vSphere 5.5, policies also support the use of tags for provisioning. Therefore, instead of using VSAN datastore capabilities for the creation of requirements within a VM storage policy, tag-based policies may also be created. The use of tag-based policies is outside the scope of this book, but further information may be found in the generic vSphere storage documentation.

Assigning a VM Storage Policy During VM Provisioning

The assignment of a VM storage policy is done during the VM provisioning. At the point where the vSphere administrator must select a destination datastore, the appropriate policy is selected from the drop-down menu of the available VM storage policies. The datastores are then separated into compatible and incompatible datastores, allowing the vSphere administrator to make the appropriate and correct choice for VM placement.

This matching of datastores does not necessarily mean that the datastore will meet the requirements in the VM storage policy. What it means is that the datastore understands

the set of requirements placed in the policy. It may still fail to provision this VM if there are not enough resources available to meet the requirements placed in the policy. However, if a policy cannot be met, the compatibility section in the lower part of the screen displays a warning that states why a policy may not be met.

This three-node cluster example shows a policy that contains a *number of failures to tolerate* = 2. A three-node cluster cannot meet this policy, but when the policy was originally created, the VSAN datastore shows up as a matching resource as it understood the contents of the policy. However, on trying to use this policy when deploying a VM, the VSAN datastore shows up as noncompliant, as Figure 4.11 demonstrates.

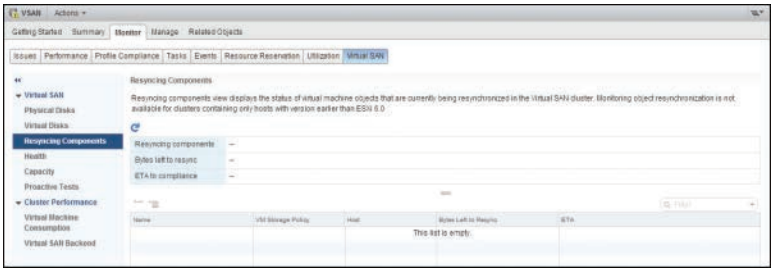

Figure 4.11 The VSAN datastore is shown as noncompliant when a policy cannot be met

This is an important point to keep in mind: Just because VSAN tells you it is compatible with a particular policy when that policy is originally created, this in no way implies that it can deploy a VM that uses the policy.

Summary

You may have used VM storage profiles in the past. VM storage policies differ significantly. Although we continue to use VASA, the vSphere APIs for Storage Awareness, VM storage policies have allowed us to switch the storage characteristics away from datastores and to the VMs. VMs, or more specifically the applications running in VMs, can now specify their own requirements using a policy that contains underlying storage capabilities around performance, reliability, and availability.

Index

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W