



Cisco DRaaS with Zerto Virtual Replication and VMware Virtual SAN

July 28, 2014

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco DRaaS with Zerto Virtual Replication and VMware Virtual SAN
© 2014 Cisco Systems, Inc. All rights reserved.



APPENDIX 1**Technology Overview 1-1**

Value of Cisco DRaaS Architecture for Service Providers	1-3
SP Monetization of Cisco DRaaS	1-4
Value of Cisco DRaaS Architecture for Enterprises	1-4
Adoption Challenges to DR and DRaaS	1-5
Cisco/Zerto DRaaS Solution Changes Traditional Capability	1-6
Disparate Hardware Increases Costs	1-6
Complexity	1-6
Zerto Virtual Replication	1-6
Hardware Agnostic	1-7
Simplicity	1-7
Built for Service Providers	1-7
Connect Customers Regardless of their Equipment	1-7
Efficient and Rapid Customer Onboarding	1-7
Centralized Management and Reporting	1-8
Standardization of the Service Provider Infrastructure	1-8
Reduced Costs	1-8
Business Agility	1-8
Simplification	1-8

APPENDIX 2**Technology Overview 2-1**

System Architecture	2-1
Provider Cloud	2-2
Enterprise Data Center	2-2
WAN Connectivity	2-2
Partner Solution for Providing Disaster Recovery	2-2
System Logical Topology	2-3
End-to-End Architecture	2-3
DRaaS Operational Workflows	2-5
Network Deployment Considerations Supporting Recovery Environment	2-6
Cisco UCS	2-7
Cisco UCS C-Series Rack-Mount Servers	2-8
Service Profiles	2-8

- Compute Over-Subscription 2-9
- VMware Virtual SAN 2-9
 - The Virtual SAN Shared Datastore 2-10
 - Read Caching and Write Buffering 2-11
 - Virtual SAN Storage Policy Based Management (SPBM) 2-12
 - Virtual SAN Recommendations and Limits 2-12
 - Virtual SAN Requirements 2-13
 - Defining Virtual Machine Requirements 2-13
 - Distributed RAID 2-14
 - Virtual SAN Storage Objects and Components 2-14
 - Witness Components 2-15
 - Flash-Based Devices in Virtual SAN 2-16
 - Read Cache 2-16
 - Write Cache (Write Buffer) 2-16

APPENDIX 3

DRaaS Application 3-1

- Zerto Virtual Replication Architecture 3-1
 - Helping the CSP Provide a Dynamic DR Platform 3-3
 - Zerto Cloud Manager: Enablement for Cloud DR Resource Management 3-3
 - Service Profiles 3-4
 - Enablement for Cloud DR Resource Consumption: Zerto Self Service Portal 3-4
 - ZSSP Features 3-5
 - Storage 3-5
 - Compression 3-7
 - Dedicated Cisco WAN Optimization Products 3-8
 - Zerto Virtual Replication 3-9
 - Encryption 3-9
 - ZVR Disaster Recovery Workflow 3-9
 - The Move Operation 3-9
 - The Failover Operation 3-11
 - Failback after the Original Site is Operational 3-12
 - Disaster Recovery Workflow 3-12
 - Best Practices 3-13

APPENDIX 4

Architecture Configuration 4-1

- VMDC VSA 1.0 System Architecture 4-2
 - Additional Resources 4-3
- Cisco UCS 4-3
 - Hardware and Software 4-4

Service Profile Configuration	4-5
VMware Virtual SAN	4-6
Host Networking	4-7
Disk Group Creation	4-8
Policy Configuration	4-10
DRaaS Application Suite	4-11
Implementation and Configuration of Zerto Virtual Replication	4-11
Implementing Zerto Virtual Replication at the Cloud Service Provider	4-15
Implementing Zerto Virtual Replication at the Customer Site	4-16
Steps Required for Zerto Virtual Replication (DRaaS Customers)	4-16
Steps Required for Zerto Virtual Replication (ICDR Customers)	4-16
Creating a Virtual Protection Group (VPG)	4-16
Configuring Virtual Protection Groups	4-17
Deploying Application VMs on the Virtual SAN Datastore	4-18
Virtual SAN Component Distribution	4-19
Zerto Application Suite on Virtual SAN at CSP Site	4-25

APPENDIX A**Virtual SAN Command Line Commands** A-1**APPENDIX B****Technical References** B-1

Cisco Technologies	B-1
VMware Technologies	B-2
Zerto Technologies	B-2



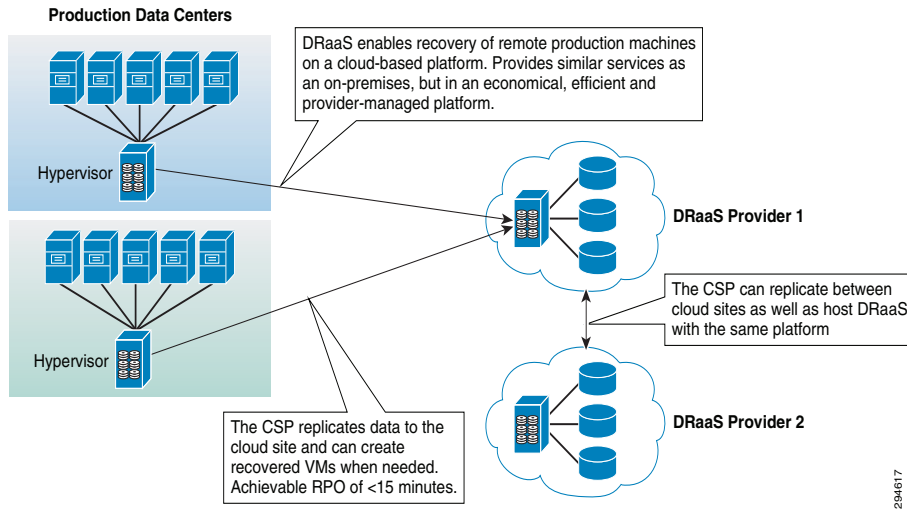
Technology Overview

The Cisco Disaster Recovery as a Service Solution (DRaaS) and In-Cloud Disaster Recovery architectures described in this document are designed to provide a new set of related capabilities allowing Virtualized Multiservice Data Center (VMDC)-based Cloud Service Provider (CSPs) to enhance their addressable market, financial performance, and differentiation vs. commodity cloud solutions (Figure 1-1). Many of Cisco's VMDC-based CSPs seek better monetization of their existing VMDC investments through layered services that are synergistic with the advanced networking capabilities delivered by VMDC. These CSPs demand new, easily deployable services both to keep pace with the innovation of commodity/public cloud providers such as Amazon Web Services (AWS) and to address portions of the market that are not well served by commodity cloud solutions.

The key end-user consumable services being enabled by this system architecture will enable a CSP to offer disaster recovery for both physical and virtual servers from a customer data center to a CSP virtual private cloud (VPC). The DRaaS to Cloud/ICDR system primarily targets SMBs and enterprises. The global DRaaS to Cloud/ICDR and cloud-based business continuity is expected to grow from \$640.84 million in 2013 to \$5.77 billion by 2018, at a CAGR of 55.20%.

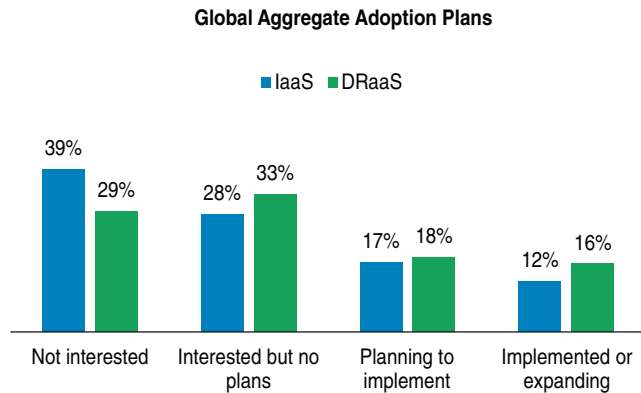
The traditional disaster recovery (DR) system constitutes a substantial portion of expenses annually. With the "pay as you go" model of the cloud-based DR system, the impact of downtime can be minimized through replication. DR can start up applications once the disaster is identified. In addition to recovery, cloud-based DR incorporates business continuity. Implementation of DRaaS to Cloud/ICDR with a virtualized cloud platform can be automated easily and is less expensive, since DR cost varies before and after a disaster occurs. The key requirements for DRaaS to Cloud/ICDR are Recovery Point Objective (RPO), Recovery Time Objective (RTO), performance, consistency, and geographic separation.

Figure 1-1 What is Disaster Recovery as a Service?



The market presents a strong opportunity for the CSPs to take advantage of the demand for DRaaS services as illustrated by Figure 1-2.

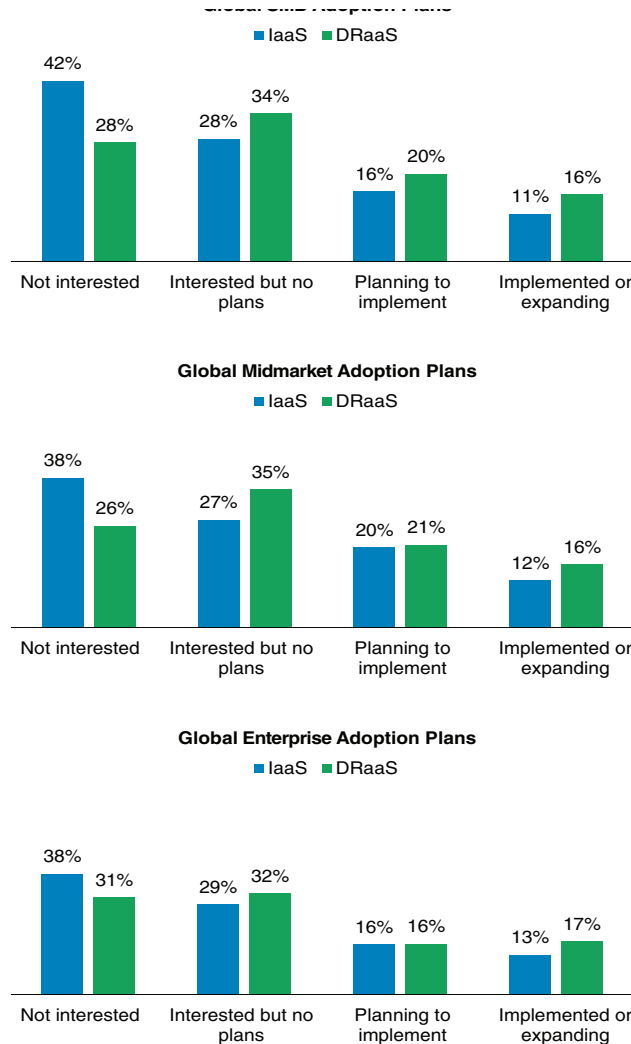
Figure 1-2 Strong Market Demand for DRaaS



• Source: Forrester Budgets and Priorities Tracker Survey Q4 2012

Further investigation of the global demand patterns for DRaaS indicates that the market opportunity and interest is equally spread across the enterprise, mid-market, and SMB segments as summarized in Figure 1-3.

Figure 1-3 Global DRaaS Demand by Segment



• Source: Forrester Budgets and Priorities Tracker Survey Q4 2012

294333

Value of Cisco DRaaS Architecture for Service Providers

DRaaS offers the following value to SPs:

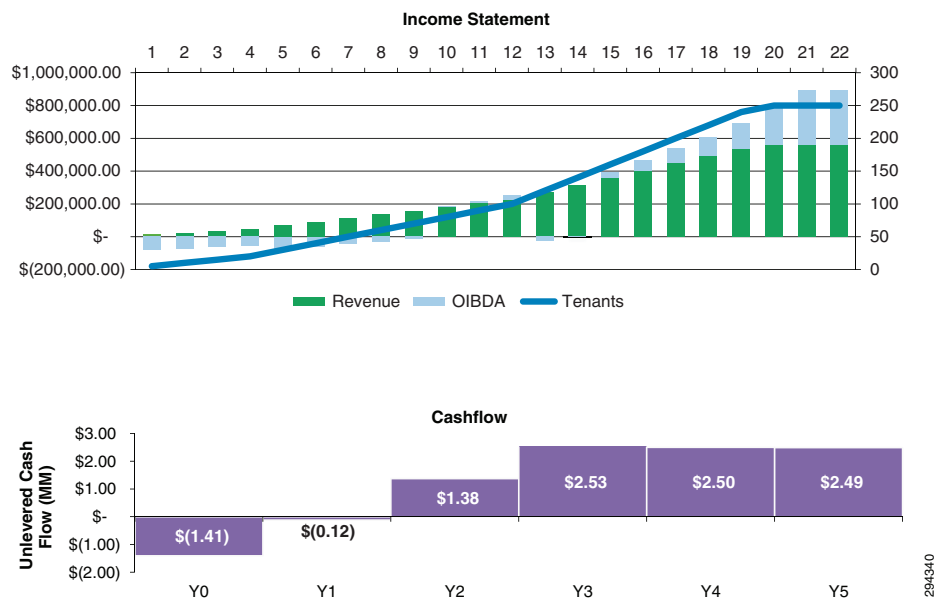
- **Increased Customer Relevance**—Not all of the customers requiring DR services want Infrastructure as a Service Offering (IaaS). Offering DRaaS provides better alignment with a typical IT buyer's focus. Leverage of DRaaS offerings by SPs provide them an opportunity to differentiate from commodity and over-the-top IaaS providers.
- **Bigger, More Profitable Deals**—DR instances command a premium and provide improved margins due to lack of commoditization. DR deals are typically larger compared to IaaS deals for SPs and generate higher margins. DRaaS offerings create reduced capital expenditures on computing resources and lower operating expenses on licensing due to oversubscription opportunities.

- **Strong Services Growth**—DRaaS offerings present a strong ability to attach additional services with the offerings and creates a pipeline of revenue from new and existing customers through new and improved monetization via services growth. Additional monetization opportunities present themselves through possibilities for hybrid services.

SP Monetization of Cisco DRaaS

Figure 1-4 is a financial model that presents the monetization opportunity for SPs associated with the deployment of the Cisco DRaaS system architecture.

Figure 1-4 Monetization Opportunity for SPs



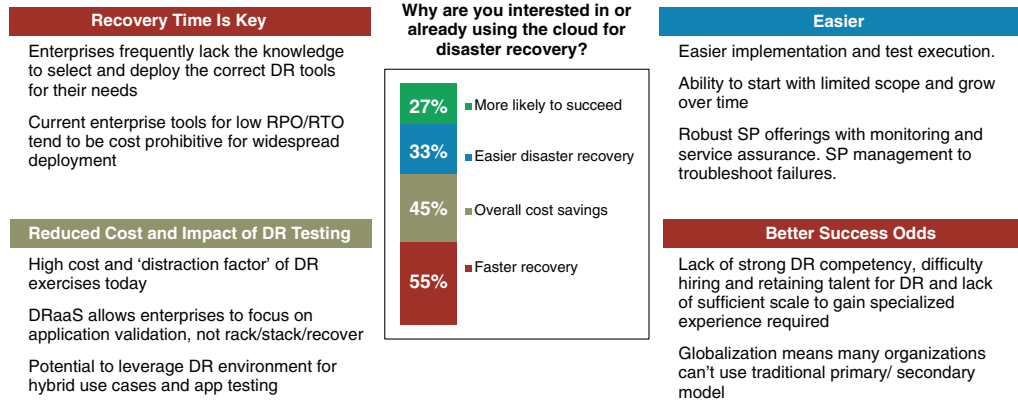
Value of Cisco DRaaS Architecture for Enterprises

DRaaS provides the following value for Enterprises:

- **Recovery Time Is Key**—Enterprises frequently lack the knowledge to select and deploy the optimal DR tools for their needs. Current enterprise tools for low RPO/RTO tend to be cost prohibitive for widespread deployment.
- **Reduced Cost and Impact of Disaster Recovery Testing**—DR exercises present a significantly high cost and are a "distraction factor" to the normal business operation. The use of DRaaS allows enterprises to focus on application validation without being distracted by rack, stack, and recover activities with their infrastructure and IT services. It also presents a potential opportunity to better leverage the DR environment
- **Accelerated Implementation**—The use of DRaaS presents an easier framework for implementation of business continuity plans and test execution and provides end customers with the ability to grow over time from a limited scope. For Enterprises to replace on their own an equivalent DRaaS solution that is provided and managed through an SP's robust offerings would be extremely time consuming. Such a solution would include self-service, monitoring, and service assurance capabilities, which would be part of a holistic DRaaS offering from the SP.

- Better Odds of Success**—Using specialized SP offerings eliminates the need for a strong DR competency and addressed the difficulty associated with hiring and retaining talent for DR. The DRaaS is a niche technology that requires a significantly large scale to gain the required specialized experience. Globalization means many organizations cannot use traditional primary and secondary model of dedicated infrastructures for DR and business continuity operations.

Figure 1-5 Why Enterprises Choose DRaaS

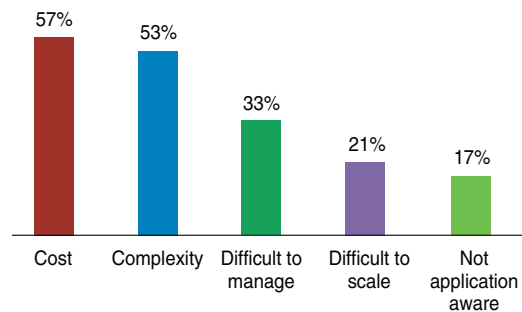


Adoption Challenges to DR and DRaaS

Looking at the Forrester results, the majority of the respondents are either not interested or have no plans for implementing a disaster recovery solution.

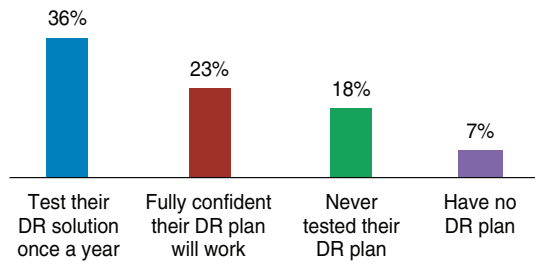
Zerto conducted a survey to gain a better understanding of why organizations are hesitant to implement a disaster recovery solution. To allow for more insightful results, Zerto allowed the respondents to check more than one box and found that cost and complexity overwhelmingly are the biggest obstacles to adopting disaster recovery.

Figure 1-6 Biggest Challenges of Disaster Recovery



• Source: Zerto Survey

To gauge the level of satisfaction with those that have implemented a disaster recovery solution, Zerto asked questions that would provide insight to the actual effectiveness of the disaster recovery solution in place (Figure 1-7).

Figure 1-7 Customer Confidence in Current Disaster Recovery Plans

• Source: Zerto Survey

294619

Cisco/Zerto DRaaS Solution Changes Traditional Capability

Both the Forrester and the Zerto studies indicate that there are barriers that need to be addressed to achieve wide-scale adoption of disaster recovery at the enterprise level and also from a service provider level.

Disparate Hardware Increases Costs

Traditional DR solutions require matching hardware at both the source site in the target site with the replication being performed by a hardware device, usually the storage array. For survey respondents, this created a capital cost barrier for the equipment purchased and significantly increased the administrative overhead to the point that the Forrester survey shows the majority of the respondents had no plan of implementing disaster recovery.

From an SP perspective, not having similar equipment at each customer site made offering a DRaaS solution so expensive that it was not pursued as a feasible service offering.

Complexity

Even if the hardware cost barrier can be overcome, traditional disaster recovery solutions requires a great deal of administrative effort to implement. Implementation usually involves an extended professional services engagement and a significant learning curve for the administrators.

For the service provider, building the core DR infrastructure is only part of the challenge. Creating a multi-tenant capable service offering has traditionally required a significant application development and programming effort.

Zerto Virtual Replication

Zerto Virtual Replication (ZVR) is a hypervisor-based replication and workflow orchestration product. Zerto developed ZVR to specifically address the major barriers to adoption.

Hardware Agnostic

ZVR has no hardware dependencies and enables continuous data protection (CDP) designed to produce production RPOs that are usually in the range of seconds and RTOs that are measured in minutes. ZVR can even support different versions of VMware vSphere and VMware vCloud.

Being hardware agnostic introduces attractive options for enterprises. They may choose to repurpose older hardware and create their own recovery site, but now they can also look at a hybrid cloud solution and choose an SP provider that is running ZVR.

Simplicity

While the underlying components manage a great deal of complexity ensuring replication and workflow orchestration is absolutely correct, the ZVR administrative level of effort is greatly simplified. The user interface is intuitive to an enterprise administrator who can usually learn to manage ZVR in about an hour. The journal in ZVR provides point-in-time recovery for testing and live failovers. The journal history can be as little as one hour, or up to five days' worth of data, with recovery points available every few seconds.

Built for Service Providers

ZVR can be adopted rapidly as a service offering because it has native multi-tenancy capabilities and an out-of-the-box self-service portal that allows customers to perform DR-related activities that are controlled by roles and permissions set by the SP. These built-in features greatly reduce the level of administrative complexity, development time, and time-to-market.

Connect Customers Regardless of their Equipment

A major barrier to DRaaS adoption has been the challenge of the customer equipment being completely different than that of their SP. When the replication between sites is completely dependent upon hardware devices, the devices must match vendor, firmware, and software. Further, all of these must be planned and upgraded at the same time. Hardware-based replication has traditionally been very unforgiving to different versions when site-to-site replication is involved. With ZVR, VMware vSphere is the only requirement, and replication is possible between different versions of VMware vSphere. Replication is possible between vSphere and vCloud environments. This is very important to a service provider because customers update their infrastructure versions on different schedules.

Efficient and Rapid Customer Onboarding

With a hypervisor-based replication solution, customers can be added very quickly. Only VMs and VMDKs are replicated, not LUNs. Regardless of the location of the host or storage, the source VM or group of VMs can be replicated to the CSP data center. This results in reduced customer onboarding time while offering a solution that fully supports the critical VMware features such as DRS, vCloud Director, VMotion, Storage VMotion.

Centralized Management and Reporting

The Zerto Cloud Manager (ZCM) centralizes management of the entire infrastructure. The SP is given a single “pane of glass” from which to view and manage all customers leveraging cloud resources. For example, reports are automatically created showing the usage of customer assets across sites. This dramatically simplifies the relationship between the customer and the CSP. These detailed resource usage reports can be used to generate invoices and can be imported into the SP’s billing system.

Standardization of the Service Provider Infrastructure

Cisco’s DRaaS system architecture is based on the Cisco VMDC cloud architecture and the Cisco Unified Computing System (UCS). VMDC is a reference architecture for building a fabric-based infrastructure that provides design guidelines demonstrating how customers can integrate key Cisco and partner technologies, such as networking, computing, integrated compute stacks, security, load balancing, and system management. Cisco UCS is a next-generation data center platform that unites compute, network, storage access, and virtualization into a cohesive system designed to reduce total cost of ownership (TCO) and increase business agility. By standardizing an infrastructure around these systems, a CSP can realize a number of benefits to reduce costs and complexity, while improving agility.

Reduced Costs

Together, Cisco VMDC and UCS reduce infrastructure expenditures (CAPEX) and operational expenses (OPEX) to increase profitability by reducing the number of devices that must be purchased, cabled, configured, powered, cooled, and secured. The unified architecture uses industry-standard technologies to provide interoperability and investment protection.

Business Agility

Together, Cisco VMDC and UCS enable business agility through faster provisioning of IT infrastructure and delivery of IT-as-a-service (ITaaS). Deployment time and cost is more predictable through the use of an end-to-end validated, scalable, and modular architecture. The unified architecture supports multiple applications, services, and tenants.

Simplification

Cisco VMDC and UCS simplify IT management to support scalability, further control costs, and facilitate automation—keys to delivering ITaaS and cloud applications. The architecture enhances the portability of both physical and virtual machines with server identity, LAN and SAN addressing, I/O configurations, firmware, and network connectivity profiles that dynamically provision and integrate server and network resources.



Technology Overview

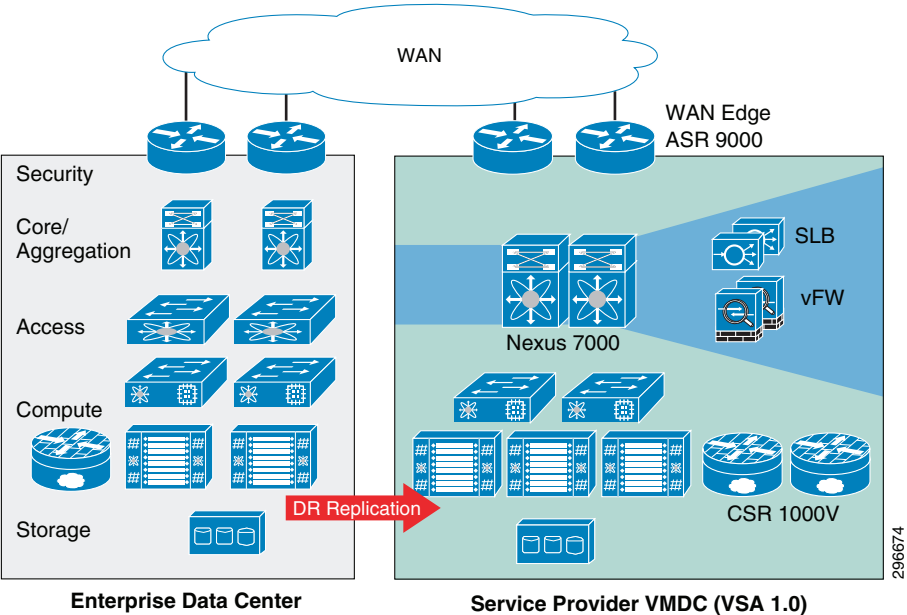
This white paper is focused on the benefits that that Cisco UCS brings to the DRaaS solution when combined with the VMware Virtual SAN hypervisor converged storage offering. These two key technologies are discussed in more detail below.

System Architecture

This section describes the high level architecture of the DRaaS 2.0 system. The system provides disaster recovery for customer physical/virtual servers by deploying recovery VMs in the VMDC VSA 1.0-based container on the provider side.

Figure 2-1 shows the high level architecture of the DRaaS 2.0 system.

Figure 2-1 DRaaS 2.0 High Level Architecture



The physical system architecture consists of the following building blocks:

- Provider Cloud
- Enterprise Data Center

- WAN Connectivity
- Partner Solution for Providing Disaster Recovery

These building blocks are described in further detail below.

Provider Cloud

The provider cloud within the DRaaS 2.0 system is based on VMDC VSA 1.0. The VSA 1.0 design is based on the earlier VMDC 2.2 design, with changes to optimize the design for lower cost, fewer layers, and increased tenancy scale. The VMDC system provides vPC-based L3 hierarchical virtual routing and forwarding (VRF)-Lite DC design, multi-tenancy, secure separation, differentiated service tiers, and high availability in a data center environment. It also provides secure separation between replicated workloads and provides shared network services for customers in DRaaS.

The VMDC VSA 1.0 architecture works with Vblock, FlexPod, or any other integration stack. Integrated stacks can be added as required to scale the CSP cloud environment.

Based on the customer's production environment and needs, a specific tenancy model can be selected to provide similar services in the cloud-matching production environment. VMDC architecture and deployment models will be covered in detail in this chapter.

Enterprise Data Center

The DR solutions should address enterprise customer requirements for various vertical industries and geographies. The enterprise data center design is therefore expected to vary from customer to customer. The intent of the DRaaS 2.0 system is to keep the enterprise DC architecture generic so as to provide the greatest coverage. While the enterprise DC architecture is almost irrelevant and the solution supports heterogeneous replication across any-to-any infrastructure, a typical three tier (core/aggregation and access) DC architecture is suggested in the system.

WAN Connectivity

The WAN connectivity design principles provided by VMDC are maintained and supported without requiring any additional components and technologies. The replicated data between the enterprise and CSP data center can be encrypted with the help of Cisco technologies like IPsec VPN, based on Cisco ASA firewalls.

To support partial failover of a customer's environment, technologies like Overlay Transport Virtualization (OTV) can be used for L2 extension between the customer's data center and the cloud. L2 connectivity allows customers to use the same IP from enterprise network in the cloud without the need to change for accessing workloads in the cloud after recovery.

Partner Solution for Providing Disaster Recovery

ZVR provides a business continuity (BC) and disaster recovery (DR) solution in a virtual environment, enabling the replication of mission-critical applications and data as quickly as possible and with minimal data loss. When devising a recovery solution, these two objectives, minimum time to recover and maximum data to recover, are assigned target values: the RTO and the RPO. ZVR enables a virtual-aware recovery with low values for both the RTO and RPO.

ZVR is installed in both the protected and the DR sites. Administrators can manage the replication from within a standalone UI in a browser, enabling DR management from anywhere or from a vSphere Client console. All recovery that does not rely on native replication functionality can be managed from the

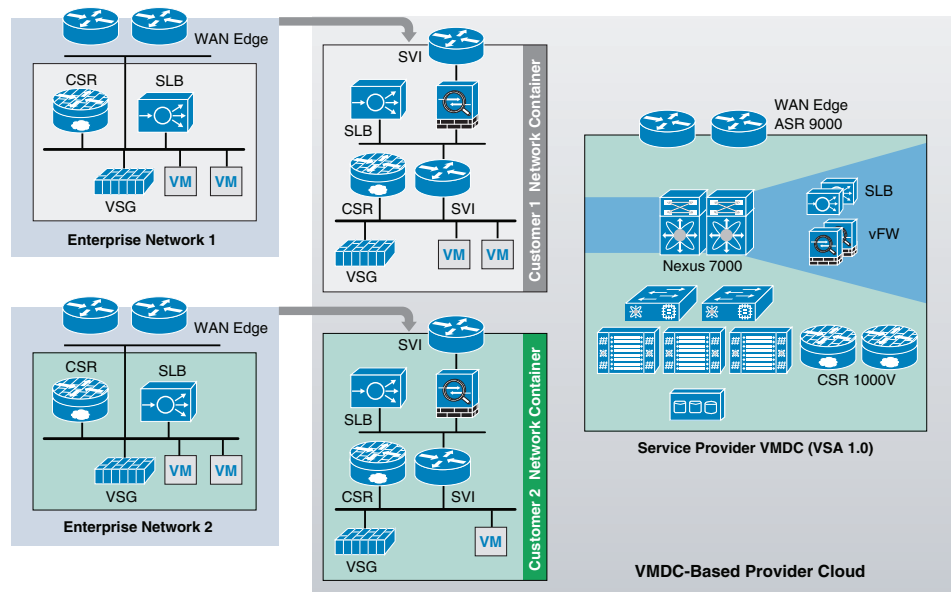
vSphere Client console. Recovery that does rely on native replication functionality, such as recovery available with Microsoft Active Directory or SQL Server, can also be replicated using ZVR, and whether the native replication functionality is used or not is determined by site considerations, such as increased complexity of having multiple points of control and possible additional costs incurred when using vendor native replication.

Replication is configured by first pairing the site with virtual machines to be protected with a recovery site. The administrator then defines the virtual machines that need protection into groups, where the virtual machines in the group comprise the application and data that needs to be recovered together. Different virtual machines can be grouped together or kept separated. Creating more granular replication affinity groups allows for optimal recovery operations.

System Logical Topology

In the logical topology [Figure 2-2](#), each customer will have a dedicated network container created on the CSP VMDC cloud. The network containers will be created based on the necessary security and network services required by the enterprise customers. Any network topology on the customer's data center can be matched on the VMDC cloud using network containers. Pre-defined containers provide examples for different types of deployments. Automated provisioning and management logic for each customer type is pre-defined in the management and orchestration software. Customers can choose from existing models or define their own customized models. The production workloads from each enterprise data center will be replicated to the corresponding network container on the VMDC cloud and will be available for recovery purposes.

Figure 2-2 DRaaS Logical Topology



End-to-End Architecture

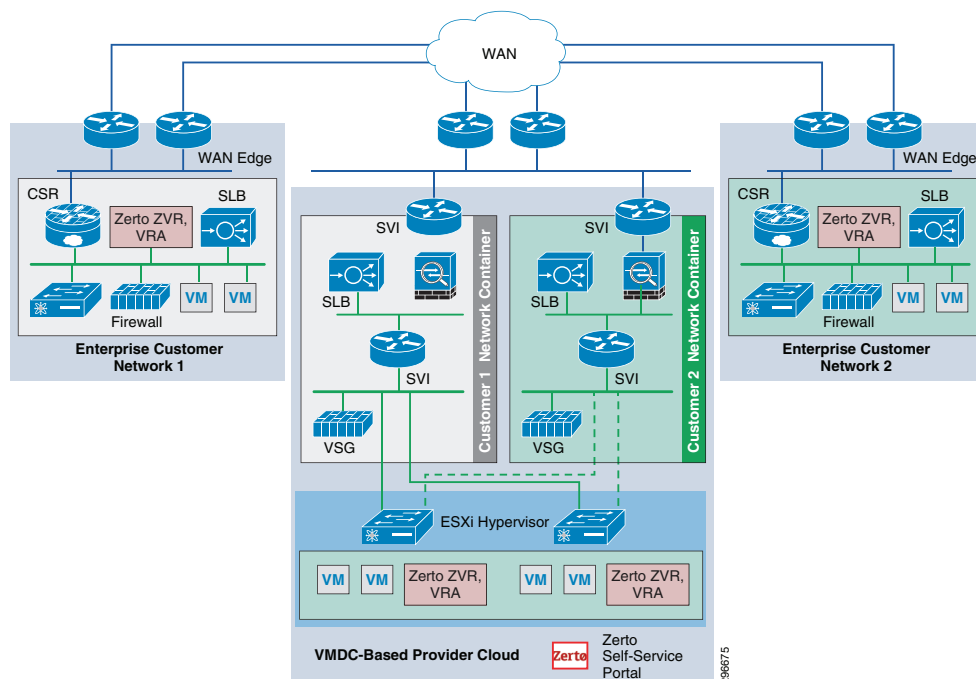
The DRaaS 2.0 system addresses the following design principles and architectural goals:

- Secure multi-tenancy

- Secure, modular, and highly available cloud
- Continuous Data Protection (CDP)
- Physical-to-Virtual (P2V) and Virtual-to-Virtual (V2V) Disaster Recovery
- Near-zero RPO- and RTO-capable DRaaS
- Automated run book automation
- Self-service multi-tenant portal

By utilizing the architecture above, DRaaS in a multi-tenant environment can be supported as shown in Figure 2-3.

Figure 2-3 End-to-End Architecture



In a multi-tenant environment, each customer is mapped as a separate VMDc tenant where the necessary network security is provided and traffic segregation is maintained. Figure 77 depicts the end-to-end architecture of the DRaaS 2.0 system based on VMDc. With the deployment of lightweight components and by utilizing the network security provided by VMDc architecture, customers can replicate their data into a secure cloud environment for recovery.

Data changes are collected from the production servers as they occur, directly in memory before they are written to disk, and sent to a software appliance within an enterprise data center. Because of this approach, absolutely no additional I/O load is induced on production servers due to replication. The appliance is responsible for further offloading compute-intensive tasks from production systems, such as compression, encryption, WAN acceleration, and consolidated bandwidth management.

The system provides the journal for the customer's production servers. The customers will be able to recover their environments to any point in time before the disaster occurred. The servers are not only protected from the physical disasters, but also from logical disasters, due to the journal.

Application consistency is enforced at regular intervals through VSS integration on Windows and native application-specific mechanisms on Linux and Solaris systems. Application consistency is also enforced at the guest level in virtual environments running VMware vSphere. These application-consistent points are tagged by a ZVR checkpoint and included as part of the journal data. They can be leveraged to perform application consistent recoveries within stringent RTOs.

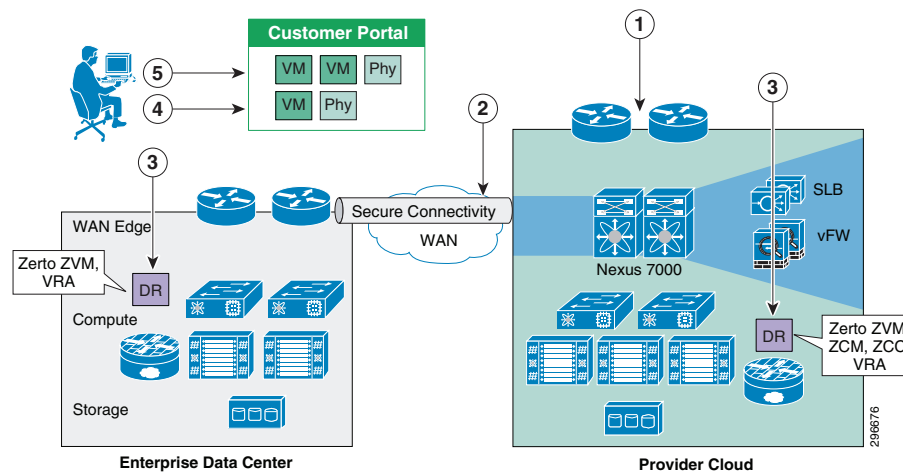
The following use cases are covered as part of the DRaaS 2.0 system and will be discussed in more detail in the following sections.

DRaaS Operational Workflows

Following are the workflows for protecting and recovering the customer's production workloads into the cloud. These workflows describe the process of creating the network containers for customers within the CSP cloud, replication of workloads into the network containers, and recovery of workloads in the event of a disaster.

The workflow in [Figure 2-4](#) is used for protection and failover scenarios.

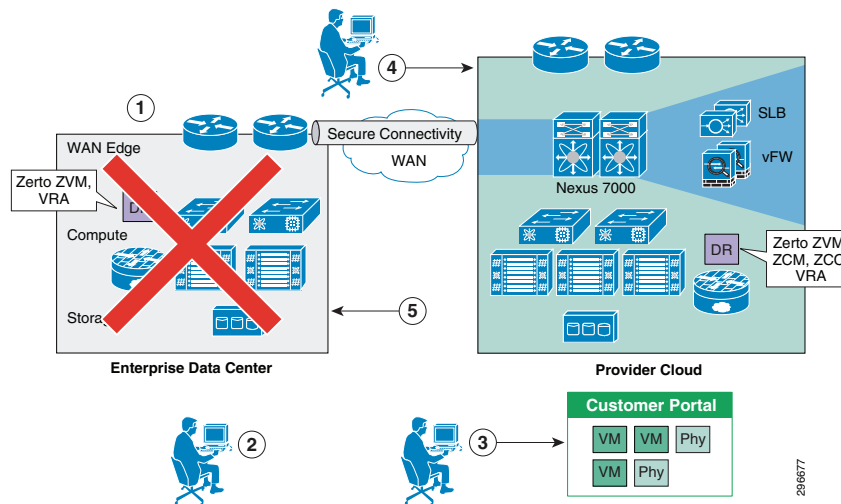
Figure 2-4 New Customer Protection Workflow



- Step 1** Based on the customer requirements, deploy a VMDC network container using BMC.
- Step 2** Secure IPsec connectivity is manually set up between the Enterprise and the VMDC-based cloud provider setup.
- Step 3** At both enterprise and CSP data centers, deploy and configure the necessary DR components.
- Step 4** Use the Zerto UI to select the machines to be protected and set up the recovery plans.
- Step 5** Allow customers to monitor the status of DR and RPO/RTO utilizing the Partner Product portals.

The workflow in case of a failure scenario is shown in [Figure 2-5](#).

Figure 2-5 Failure Scenario



- Step 6** When the customer DC goes down, the customer declares a disaster and communicates to the CSP what VMs to restore and what checkpoints to use. The CSP can use the recovery plan (which could be preconfigured), which details the list of protected VMs, the startup order, and any custom steps.
- Step 7** The CSP logs into the DR product portal and brings up the required VMs in its environment. Customers with self-service capabilities will be able to recover VMs in the cloud themselves using the self-service portal.
- Step 8** The customer works with its DNS provider to direct the client traffic to the CSP DC. If the customer is utilizing a Global Site Selector (GSS)-based DNS solution or has an L2 extension, this step will be automatic or not required.
- Step 9** When the Enterprise DC is back up, the customer works with the CSP during a maintenance window to bring up the VMs in customer DC, failback the VMs from CSP to enterprise, and update the DNS so that the client traffic is re-routed to the customer DC.

Network Deployment Considerations Supporting Recovery Environment

Table 2-1 shows the considerations in matching the networks between the enterprise's and CSP's VPC. Logically, the enterprise network will consist of VLANs and network services, including firewall rules and load balancing. Based on the requirements of the enterprise, which depend on the type of applications that are protected, network containers can be created on the VMDC to meet those requirements.

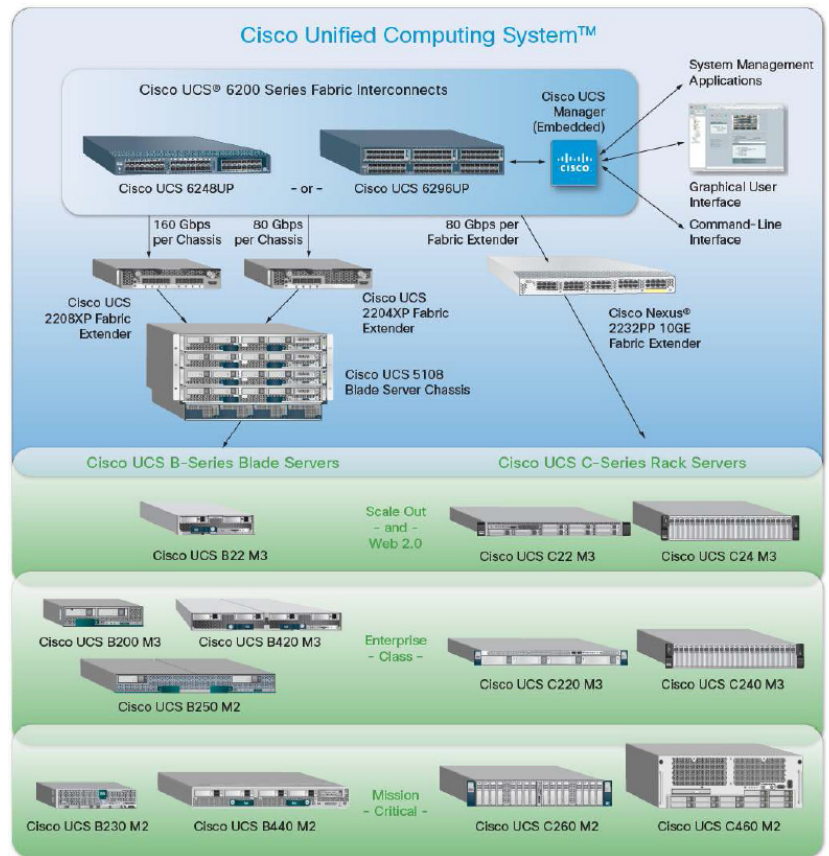
Table 2-1 Network Containers available on VMDC

Container	VLANs	Network Services
Gold	3	Tenant firewall, intra-tenant firewall, and load balancer
Silver	3	Load balancer
Bronze	1	Intra-tenant firewall, load balancer
Copper	1	Intra-tenant firewall

Cisco UCS

The Cisco Unified Computing System (Cisco UCS) is a next-generation data center platform that unites computing, networking, storage access, and virtualization resources into a cohesive system designed to reduce total cost of ownership (TCO) and increase business agility. The system integrates a low-latency, lossless 10 Gigabit Ethernet unified network fabric with enterprise-class, x86-architecture servers. The system is an integrated, scalable, multi-chassis platform in which all resources participate in a unified management domain.

Figure 2-6 Cisco Unified Computing System



Two elements of the Cisco UCS architecture, immediately relevant to the use of Virtual SAN in the DRaaS solution, are highlighted in this white paper: the C-Series rack-mount server and UCS service profiles. Much additional material is already available discussing the capabilities and features of Cisco UCS technologies. Please see the following resources for more information:

<http://cisco.com/go/ucs>

http://cisco.com/c/en/us/td/docs/solutions/Hybrid_Cloud/DRaaS/Service_Template/Service_Templates.pdf

Cisco UCS C-Series Rack-Mount Servers

Cisco UCS C-Series rack-mount servers extend Cisco's Unified Computing to an industry-standard form factor reducing total cost of ownership and increasing agility. A full range of rack servers address workload challenges with a balance of processing, memory, I/O, and internal storage resources. These servers can be managed by the built-in standalone software—called Cisco Integrated Management Controller (CIMC)—or by Cisco UCS Manager when connected through a Cisco Nexus 2232PP Fabric Extender. Cisco UCS Manager, which supplies a totally integrated management process for both rack and blade servers inside a single tool, was used for this white paper. All Cisco UCS servers use leading Intel® Xeon® processors.

There are several UCS C-Series rack-mount server models available, each optimized for particular deployments. For Virtual SAN deployments, disk density drives the model selection, with computing power also being an important consideration. For these reasons, the UCS C240 M3 was chosen for the development of this white paper. The UCS C240 M3 is a 2RU form factor server that comes in two varieties supporting either large form factor (3.5”) or small form factor (2.5”) hard drives. Due to the need for SSDs by the Virtual SAN technology, the small form factor variety is necessary. This provides capacity for up to 24 SAS/SATA/SSD drives and up to 24 TB of capacity (Virtual SAN capacity particulars will be discussed below). The C240 M3 also supports two CPUs and up to 768 GB RAM. Compute performance is important because the VMs leveraging the Virtual SAN datastore will reside on the same hosts that contribute the disk capacity to the datastore.

The following URL provides the complete spec sheet for the UCS C240 M3 small form factor server:

http://www.cisco.com/en/US/prod/collateral/ps10265/ps10493/C240M3_SFF_SpecSheet.pdf

For more details on each of the available models, please visit:

<http://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-c-series-rack-servers/datasheet-listing.html>

For a side-by-side comparison between the different models, please visit:

<http://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-c-series-rack-servers/models-comparison.html>

Service Profiles

In Cisco UCS, a service profile adds a layer of abstraction to the actual physical hardware. The server is defined in a configuration file, which is stored on the UCS 6200 Series Fabric Interconnects and can be associated with physical hardware in a simple operation from the UCS Manager. When the service profile is applied, the UCS Manager configures the server, adaptors, Fabric Extenders, and Fabric Interconnects as specified in the service profile. The service profile makes the physical hardware transparent to the OSs and VMs running on it, enabling stateless computing and maximization of data center resources.

A number of parameters can be defined in the service profile depending on the environment requirements. Administrators can create policies to define specific rules and operating characteristics, and be referenced in the service profiles to ensure consistent configuration across many servers. Updates to a policy can be propagated to all servers that reference that policy in their service profile immediately, or in the case of firmware updates, at the next power cycle event.

The advantages of the service profile can be extended further when server-specific parameters such as UUID, MAC address, and WWN are themselves parameterized and the service profile is converted to a template. The template can be used to rapidly deploy new servers with consistent general parameters and unique server-specific parameters

Service profiles, in use with templates, enable rapid provisioning of servers with consistent operational parameters and high availability functionality. They can be configured in advance and used to move servers to a new blade, chassis, or rack in the event of a failure. The main configurable parameters of a service profile are summarized in [Table 2-2](#).

Table 2-2 Service Profile Parameters

Parameter Type	Parameter	Description
Server Hardware	UUID	Obtained from defined UUID pool
	MAC addresses	Obtained from defined MAC pool
	WWPN/WWNN	Obtained from defined WWPN and WWNN pools
	Boot policy	Boot paths and order
	Disk policy	RAID configuration
Fabric	LAN	vNICs, VLANs, MTU
	SAN	vHBAs, VSANs
	QoS policy	Set CoS for Ethernet uplink traffic
Operational	Firmware policies	Current and backup versions
	BIOS policy	BIOS version and settings
	Stats policy	Controls system data collection
	Power Control policy	Blade server power allotment

Compute Over-Subscription

DRaaS utilizes shared resources on the recovery site. Since resources at the failover site sit idle most of the time, DR enables high over-subscription ratios, making it ideal for cloud environments.

The SP can maintain fewer compute resources compared to the customer's production environments. The compute within the CSP cloud is based on Cisco UCS servers, which can be rapidly deployed with the help of the UCS service profiles to meet any unexpected or rare scenario where all the customers fail over to the cloud. In this scenario, new UCS servers can quickly be deployed and added to the existing compute clusters for additional compute resource needs.

VMware Virtual SAN

Virtual SAN is a new software-defined storage solution that is fully integrated with vSphere. Virtual SAN aggregates locally attached disks in a vSphere cluster to create a storage solution—a shared datastore—that rapidly can be provisioned from VMware vCenter™ during virtual machine provisioning operations. It is an example of a hypervisor-converged platform—that is, a solution in which storage and compute for virtual machines are combined into a single device, with storage's being provided within the hypervisor itself as opposed to via a storage virtual machine running alongside other virtual machines.

Virtual SAN is an object-based storage system designed to provide virtual machine-centric storage services and capabilities through a Storage Based Policy Management (SPBM) platform. SPBM and virtual machine storage policies are solutions designed to simplify virtual machine storage placement decisions for vSphere administrators.

Virtual SAN is fully integrated with core vSphere enterprise features such as VMware vSphere High Availability (vSphere HA), VMware vSphere Distributed Resource Scheduler™ (vSphere DRS), and VMware vSphere vMotion®. Its goal is to provide both high availability and scale-out storage functionality. It also can be considered in the context of quality of service (QoS) because virtual machine storage policies can be created to define the levels of performance and availability required on a per-virtual machine basis.

**Note**

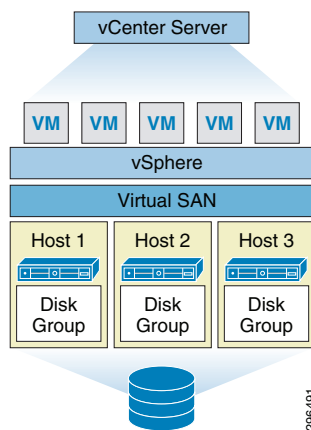
This and the remainder of this technology overview has been compiled, directly and indirectly, from resources available on the Virtual SAN resources website, at the following URL:
<http://www.vmware.com/products/virtual-san/resources.html>.

Another highly recommended resource for Virtual SAN administrators will be the forthcoming *Essential Virtual SAN (VSAN): Administrator's Guide to VMware VSAN* by Cormac Hogan and Duncan Epping.

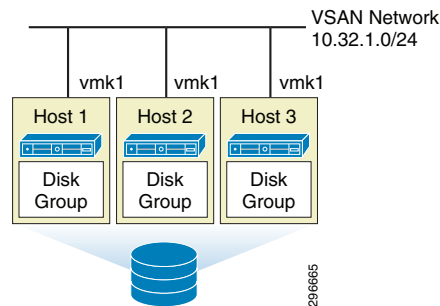
The Virtual SAN Shared Datastore

The Virtual SAN shared datastore is constructed with the minimum three ESXi hosts, each containing at least one SSD and one MD (magnetic drive). Each SSD forms a disk group on the host to which the MD belongs. The VMware virtual machine (VM) files are stored on the MD while the SSD handles the read caching and write buffering. The disk group on each host is joined to a single Network Partition Group, shared and controlled between the hosts. Figure 2-7 shows a Virtual SAN cluster with the minimum configuration.

Figure 2-7 VMware Virtual SAN



For this white paper, the base Virtual SAN cluster was built with three hosts, each having one disk group comprised of one 400GB SSD and four 1TB MDs, controlled by a RAID controller. Each host had a single VMkernel NIC (vmk1), on the 10.32.1.0/24 network, for Virtual SAN communication on a 10Gb physical NIC. Multicast was enabled as required for Virtual SAN control and data traffic. Figure 2-8 illustrates the particular environment built for this white paper. Details of the physical configuration are in Chapter 4, “Architecture Configuration.”

Figure 2-8 Virtual SAN White Paper Environment

The size and capacity of the Virtual SAN shared datastore are dictated by the number of magnetic disks per disk group in a vSphere host and by the number of vSphere hosts in the cluster. For example, using the configuration of this white paper environment, the cluster is composed of three vSphere hosts, where each host contains one disk group composed of four magnetic disks of 1TB in size each, the total raw capacity of the Virtual SAN shared datastore is 11.9TB after subtracting the metadata overhead capacity.

Formulae

- One (1) disk group x four (4) magnetic disks x 1TB x three (3) hosts = 11.9TB raw capacity
- 12TB raw capacity – 21GB metadata overhead = 11.9TB usable raw capacity

With the Cisco UCS C240-M3 rack-mount servers being used to build the Virtual SAN cluster, the theoretical maximum datastore capacity is roughly 168TB, according to the following formula:

- Three (3) disk groups x seven (7) magnetic disks x 1TB x 32 hosts = 672TB raw capacity

After the Virtual SAN shared datastore has been formed, a number of datastore capabilities are surfaced up to vCenter Server. These capabilities are based on storage capacity, performance, and availability requirements and are discussed in greater detail in the “Storage Policy Based Management” section of this paper. The essential point is that they can be used to create a policy that defines the storage requirements of a virtual machine.

These storage capabilities enable the vSphere administrator to create virtual machine storage policies that specify storage service requirements that must be satisfied by the storage system during virtual machine provisioning operations. This simplifies the virtual machine provisioning operations process by empowering the vSphere administrator to easily select the correct storage for virtual machines.

Read Caching and Write Buffering

The flash-based device (e.g. SSD) in the Virtual SAN host serves two purposes: caching the reads and buffering the writes coming from the resident VMs. The read cache keeps a cache of commonly accessed disk blocks. This reduces the I/O read latency in the event of a cache hit. The actual block that is read by the application running in the virtual machine might not be on the same vSphere host on which the virtual machine is running.

To handle this behavior, Virtual SAN distributes a directory of cached blocks between the vSphere hosts in the cluster. This enables a vSphere host to determine whether a remote host has data cached that is not in a local cache. If that is the case, the vSphere host can retrieve cached blocks from a remote host in the cluster over the Virtual SAN network. If the block is not in the cache on any Virtual SAN host, it is retrieved directly from the magnetic disks.

The write cache performs as a nonvolatile write buffer. The fact that Virtual SAN can use flash-based storage devices for writes also reduces the latency for write operations.

Because all the write operations go to flash storage, Virtual SAN ensures that there is a copy of the data elsewhere in the cluster. All virtual machines deployed onto Virtual SAN inherit the default availability policy settings, ensuring that at least one additional copy of the virtual machine data is available. This includes the write cache contents.

After writes have been initiated by the application running inside of the guest operating system (OS), they are sent in parallel to both the local write cache on the owning host and the write cache on the remote hosts. The write must be committed to the flash storage on both hosts before it is acknowledged.

This means that in the event of a host failure, a copy of the data exists on another flash device in the Virtual SAN cluster and no data loss will occur. The virtual machine accesses the replicated copy of the data on another host in the cluster via the Virtual SAN network.

Virtual SAN Storage Policy Based Management (SPBM)

All VMs deployed on a Virtual SAN cluster must use a VM Storage Policy and, in the case where there is none administratively defined, the default is applied. VM Storage Policies define the requirements of the application running in the VM from an availability, sizing, and performance perspective. There are five VM Storage Policy requirements in Virtual SAN, presented in [Table 2-3](#).

Table 2-3 VM Storage Policy Requirements

Policy	Definition	Default	Maximum
Number of Disk Stripes Per Object	The number of MDs across which each replica of a storage object is distributed.	1	12
Flash Read Cache Reservation	Flash capacity reserved as read cache for the storage object	0%	100%
Number of Failures to Tolerate	The number of host, disk, or network failures a storage object can tolerate. For n failures tolerated, $n+1$ copies of the object are created and $2n+1$ hosts contributing storage are required.	1	3 (in an 8-host cluster)
Force Provisioning	Determines whether the object will be provisioned even if currently available resources do not satisfy the VM Storage Policy requirements.	Disabled	[Enabled]
Object Space Reservation	The percentage of the logical size of the storage object that should be reserved (thick provisioned) upon VM provisioning. (The remainder of the storage object will be thin provisioned.)	0%	100%

Virtual SAN Recommendations and Limits

The following are the limits and recommendations for Virtual SAN at this white paper's publication date.

Limits:

- Maximum 32 hosts per Virtual SAN cluster
- Maximum 5 disk groups per host

- Maximum 7 MDs per disk group
- Maximum 1 SSD per disk group

Recommendations:

- Each cluster host shares identical hardware configuration
- Each cluster host has like number of disk groups
- SSD-to-MD capacity ratio of 1:10 of the anticipated consumed storage capacity before the Number of Failures to Tolerate (FTT) is considered
- Each cluster host has a single Virtual SAN-enabled VMkernel NIC

Virtual SAN Requirements

An abbreviated listing of the requirements needed for running a Virtual SAN virtual storage environment follows:

- vCenter Server: Minimum version 5.5 Update 1
- vSphere: Minimum version 5.5
- Hosts: Minimum three (3) ESXi hosts
- Disk Controller:
- SAS or SATA HBA *or*
- RAID controller
- Must function in either pass-through (preferred) or RAID 0 modes
- Hard Disk Drives: Minimum one (1) SAS, NL-SAS, or SATA magnetic hard drive per host
- Flash-Based Devices: Minimum one (1) SAS, SATA, or PCI-E SSD per host
- Network Interface Cards: Minimum one (1) 1Gb or 10Gb (recommended) network adapter per host.
- Virtual Switch: VMware VDS or VSS, or Cisco Nexus 1000v
- VMkernel Network: VMkernel port per host for Virtual SAN communication

Defining Virtual Machine Requirements

When the Virtual SAN cluster is created, the shared Virtual SAN datastore—which has a set of capabilities that are surfaced up to vCenter—is also created.

When a vSphere administrator begins to design a virtual machine, that design is influenced by the application it will be hosting. This application might potentially have many sets of requirements, including storage requirements.

The vSphere administrator uses a virtual machine storage policy to specify and contain the application's storage requirements in the form of storage capabilities that will be attached to the virtual machine hosting the application; the specific storage requirements will be based on capabilities surfaced by the storage system. In effect, the storage system provides the capabilities, and virtual machines consume them via requirements placed in the virtual machine storage policy.

Distributed RAID

In additional storage environments, redundant array of independent disks (RAID) refers to disk redundancy inside the storage chassis to withstand the failure of one or more disk drives.

Virtual SAN uses the concept of distributed RAID, by which a vSphere cluster can contend with the failure of a vSphere host, or of a component within a host—for example, magnetic disks, flash-based devices, and network interfaces—and continue to provide complete functionality for all virtual machines. Availability is defined on a per-virtual machine basis through the use of virtual machine storage policies.

vSphere administrators can specify the number of host component failures that a virtual machine can tolerate within the Virtual SAN cluster. If a vSphere administrator sets zero as the number of failures to tolerate in the virtual machine storage policy, one host or disk failure can impact the availability of the virtual machine.

Using virtual machine storage policies along with Virtual SAN distributed RAID architecture, virtual machines and copies of their contents are distributed across multiple vSphere hosts in the cluster. In this case, it is not necessary to migrate data from a failed node to a surviving host in the cluster in the event of a failure.

Virtual SAN Storage Objects and Components

While the traditional understanding of a virtual machine is that it is a set of files (.vmx, .vmdk, etc.), because the Virtual SAN datastore is an object datastore, a VM on a Virtual SAN is now made up of a set of objects. For VMs on Virtual SAN there are four kinds of Virtual SAN objects:

- The VM home or “namespace” directory
- A swap object (if the VM is powered on)
- Virtual disks/VMDKs
- Delta-disks created for snapshots (each delta-disk is an object)



Note

The VM namespace directory holds all VM files (.vmx, log files, etc.), excluding VM disks, deltas, and swap, all of which are maintained as separate objects.



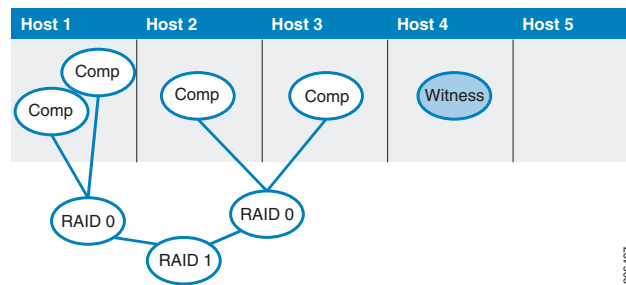
Note

It is important to understand how objects and components are built and distributed in Virtual SAN because there are soft limitations and exceeding those limitations may impact performance.

Each object is deployed on Virtual SAN as a distributed RAID tree and each leaf of the tree is said to be a component. The policies relevant to Virtual SAN object and component count and limitations include the Failures-to-Tolerate (FTT) policy and the Stripe-Width policy. If, for example, deploying a VM with a Stripe Width of two means that a RAID-0 stripe would be configured across two magnetic disks for the VM disk. Similarly, if the FTT policy for that VM is configured as one, a RAID-1 mirror of the VM components would be set up across hosts.

Figure 2-9 represents a possible layout for the components in the above scenario. The stripe components form a RAID-0 configuration, which is then mirrored across hosts using a RAID-1 configuration.

Figure 2-9 Sample component layout for VM on Virtual SAN



Following are some considerations to keep in mind when working with objects and components:

- Each VM has, potentially, four kinds of objects: Namespace; VMDK; Swap; Snapshot delta-disk
 - Namespace: Every VM has a namespace object, and only one
 - VMDK: Every VM will have one VMDK object for each attached virtual disk
 - Swap: Every powered-on VM will have a swap object
 - Delta-disk: Every VM will have one delta-disk object for each snapshot created
- Of the four families of objects, only the VMDKs and delta-disks will inherit the *StripeWidth* policy administratively applied to the VM. Because performance is not a major requirement for the namespace or swap objects, the *StripeWidth* will always be set to 1.
- Witness components will be created to arbitrate between remaining copies should a failure occur so that two identical copies of data are not activated at the same time. Witnesses are not objects but are components within each object RAID tree. More information on witnesses is provided below.



Note

VMware recommends the default settings for *NumberOfFailuresToTolerate* and *StripeWidth*.

Witness Components

As mentioned above, witnesses are components that are deployed to arbitrate between the remaining copies of data should a failure occur within the Virtual SAN cluster, ensuring no split-brain scenarios occur. At first glance, the way witnesses are deployed seems to have no rhyme or reason, but the algorithm governing this mechanism is not very complex and is worth mentioning here.

Witness deployment is not predicated on any *FailuresToTolerate* or *StripeWidth* policy setting. Rather, witness components are defined by three names (Primary, Secondary, and Tiebreaker) and are deployed based on the following three rules.

Primary Witnesses

Need at least $(2 * \text{FTT}) + 1$ nodes in a cluster to be able to tolerate FTT number of node / disk failures. If after placing all the data components, we do not have the required number of nodes in the configuration, primary witnesses are on exclusive nodes until there are $(2 * \text{FTT}) + 1$ nodes in the configuration.

Secondary Witnesses

Secondary witnesses are created to make sure that every node has equal voting power towards quorum. This is important because every node failure should affect the quorum equally. Secondary witnesses are added so that every node gets equal number of component, this includes the nodes that only hold primary witnesses. So the total count of data component + witnesses on each node are equalized in this step.

Tiebreaker Witnesses

If after adding primary and secondary witnesses we end up with even number of total components (data + witnesses) in the configuration then we add one tiebreaker witnesses to make the total component count odd.

**Note**

This is all that will be said about witness functionality here, though some Chapter 4 below demonstrates these three rules in action in deployment examples for this project. This paper is indebted to Rawlinson's blog post on this topic, from which the three rules were quoted verbatim and to which the reader is encouraged to go to gain an even better understanding.

<http://www.punchingclouds.com/2014/04/01/vmware-virtual-san-witness-component-deployment-logic/>

Flash-Based Devices in Virtual SAN

Flash-based devices serve two purposes in Virtual SAN. They are used to build the flash tier in the form of a read cache and a write buffer, which dramatically improves the performance of virtual machines. In some respects, Virtual SAN can be compared to a number of “hybrid” storage solutions on the market that also use a combination of flash-based devices and magnetic disk storage to boost the performance of the I/O and that have the ability to scale out based on low-cost magnetic disk storage.

Read Cache

The read cache keeps a cache of commonly accessed disk blocks. This reduces the I/O read latency in the event of a cache hit. The actual block that is read by the application running in the virtual machine might not be on the same vSphere host on which the virtual machine is running.

To handle this behavior, Virtual SAN distributes a directory of cached blocks between the vSphere hosts in the cluster. This enables a vSphere host to determine whether a remote host has data cached that is not in a local cache. If that is the case, the vSphere host can retrieve cached blocks from a remote host in the cluster over the Virtual SAN network. If the block is not in the cache on any Virtual SAN host, it is retrieved directly from the magnetic disks.

Write Cache (Write Buffer)

The write cache performs as a nonvolatile write buffer. The fact that Virtual SAN can use flash-based storage devices for writes also reduces the latency for write operations.

Because all the write operations go to flash storage, Virtual SAN ensures that there is a copy of the data elsewhere in the cluster. All virtual machines deployed onto Virtual SAN inherit the default availability policy settings, ensuring that at least one additional copy of the virtual machine data is available. This includes the write cache contents.

After writes have been initiated by the application running inside of the guest operating system (OS), they are sent in parallel to both the local write cache on the owning host and the write cache on the remote hosts. The write must be committed to the flash storage on both hosts before it is acknowledged.

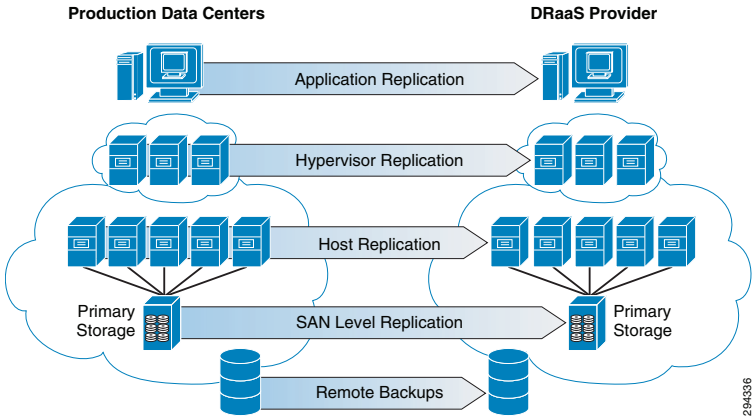
This means that in the event of a host failure, a copy of the data exists on another flash device in the Virtual SAN cluster and no data loss will occur. The virtual machine accesses the replicated copy of the data on another host in the cluster via the Virtual SAN network.



DRaaS Application

Several options exist in the choice of technology for the implementation of DRaaS, which is associated with varying levels of cost, complexity, and operational models. A summary of technology options for the implementation is presented in [Figure 3-1](#).

Figure 3-1 Many Approaches to DRaaS



The hypervisor-based replication technology is one of the recommended implementations for Cisco's DRaaS System architecture. It is delivered in partnership with The Zerto Virtual Replication (ZVR) product offering because of the value and the differentiation ZVR provides delivering DR services for virtual-to-virtual (V2V) workloads.

Layer 2 Extensions and IP mobility using Overlay Transport Virtualization (OTV) and Locator/ID Separation Protocol (LISP) to support partial failovers and active-active scenarios are part of the VMDC Virtual Services Architecture (VSA) 1.0 solution. The solution presents heterogeneous, storage, and infrastructure-agnostic data replication capabilities for the creation and offer of DR solution offerings. The system offers continuous data protection (CDP)-based recovery with the ability to roll back to any point in time. The system provides guaranteed application consistency for most of the widely-used industry applications.

Zerto Virtual Replication Architecture

ZVR and workflow orchestration is a powerful DR solution for organizations that have virtualized environments. ZVR functions at the hypervisor layer, replicating the changes made on the servers at the production site to one or more recovery locations, including the CSP sites. ZVR provides robust workflow orchestration of the failover, migration, and failback operations while allowing complete

failover testing that is not disruptive to the production environment. For the CSP, ZVR is an important technological advance that opens up a whole new set of DRaaS and in-the-cloud cost-effective service offerings.

Since ZVR is "VM-aware," it is possible to select only the VMs that need to be protected which in turn saves storage space at the secondary site as well as network bandwidth between sites. Further, ZVR does not require similar storage between sites, which allows for cheaper or repurposed storage to be used at the target site. The CSP site can be added as a target site as well since ZVR has no hardware dependencies. This presents compelling options to the customer in using one solution for protecting all of their servers, including lower-tier VM protection to any site, public or private.

For the CSP, having the same data protection platform that the customer is using simplifies and accelerates the sales and on-boarding process by removing the barriers to adoption. Additionally, ZVR is natively multi-tenant, so the internal deployment on the CSP infrastructure is non-disruptive.

ZVR allows for very granular protection since the VMware VM VMDKs are being replicated. For application protection, multiple VMs can be put into application affinity groupings called Virtual Protection Groups (VPGs). Virtual machines that are in a VPG have write-order fidelity, which means that the recovery points-in-time are consistent across all the VMs in the VPG.

A hypervisor-based replication solution aligns with the capabilities of the hypervisor, extending the flexibility, agility and benefits of virtualization to BC/DR.

In summary, Zerto Virtual Replication:

- Removes deployment barriers with a storage agnostic solution that installs seamlessly into the existing infrastructure.
- Supports multiple VMware vSphere versions, mixed VMware licensing levels, VMware vCloud environments.
- Provides a centralized DR management solution, regardless of the VM placement.
- Is completely virtual-aware so the customer can make changes to the production environment without impacting existing BC/DR processes.
- Enables hybrid cloud services. VM portability between private and public clouds is simple with very low recovery times when using ZVR.
- Provides the technical infrastructure for secure and segmented multi-tenant DR access

However, providing DR services is different from providing other cloud-based services:

- In a DRaaS scenario, the customer may manage and have complete control over the production data, or the CSP may provide a partial or complete managed service. In either case, the CSP must ensure the availability of the data and adapt as the customer's infrastructure changes.
- When customers leverage an ICDR service, the CSP manages the production and DR sites. The VMs are typically replicated from one CSP data center to another CSP data center as a managed service or as managed co-located data centers. The customers have the ability to interact with their applications as if they were locally hosted.

What is consistent in both scenarios is that the customers have deeper ties to their data when compared to other cloud-based services because they often need to access the actual VMs running the applications.

CSPs are challenged to provide a multi-tenant service that bridges together and connects dissimilar data centers from customers to the CSP's cloud as well as having customer-initiated tests and failovers.

Helping the CSP Provide a Dynamic DR Platform

At the core of the Zerto design philosophy is to simplify DR while providing powerful replication, recovery, and testing with no impact on the environment. ZVR makes VMs more geographically portable and simplifies the technology behind the DR that the CSP provides to customers. With ZVR 3.0, Zerto improves the management experience by adding multi-tenant cloud management and customer-initiated enablement technologies with Zerto Cloud Manager (ZCM) and the Zerto Self Service Portal (ZSSP).

The ZCM allows the CSP to provide resources from multiple CSP data centers and define service level templates called Service Profiles to multiple customers via a unified administrative interface. From the customer perspective, the CSP provides the ZSSP, which is a web-based portal that enables self-initiated provisioning, testing, and failover capability through a private, intuitive administration interface.

By making DR easier to provide and consume, Zerto helps the CSP reach the enterprise IT Manager better by offering DR options that were previously unfeasible or cost-prohibitive. The CSP can offer services ranging from fully managed DR to providing DR for only a portion of the Enterprise’s VMs where a hybrid cloud-based DR approach is a better solution.

Zerto helps drive new service offering innovation for the CSPs. For example, a growing service offering from CSPs using ZVR is “reverse DR.” This configuration uses the CSP’s cloud as the primary site while the customer’s site or sites serve as the DR targets. This is an attractive option to many customers because it allows the customer to use less or older hardware for their DR locally and leverage the power and availability of the CSP’s equipment.

Zerto Cloud Manager: Enablement for Cloud DR Resource Management

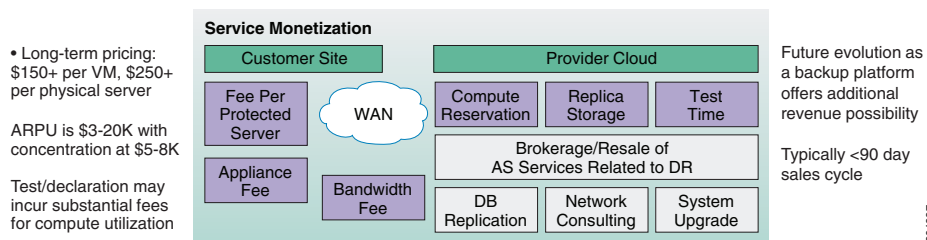
CSPs regularly host the same customer in multiple global locations. ZVR's unique architecture can easily support replication between sites around the world.

While ZVR creates an advantage for CSPs by enabling them to replicate to and from anywhere, it introduces the need for a centralized interface that consolidates information from multiple sites to make management and reporting easier and more accurate.

Zerto has created the Zerto Cloud Manager (ZCM) to deliver centralized management for DR in the cloud. The ZCM consolidates and streamlines resource information into a single interface to make multi-site, multi-tenant, dynamic DR environments easier to manage. The automated consolidation and reporting on cloud usage increases the confidence of customers that they will be billed accurately on their infrastructure usage.

As shown in [Figure 3-2](#), the ZCM manages all of the information from the ZVM at each location in a central user interface.

Figure 3-2 An Example ZVR Deployment



ZCM is the "manager of managers," which is to say the ZCM interfaces with each site's ZVM and allows the CSP administrator to have a single point of management. The administrator can view all of the individual site configurations and statuses, create and manage VPGs, conduct failover tests, migrations or actual failovers, and generate reports, alerts and billing information.

Service Profiles

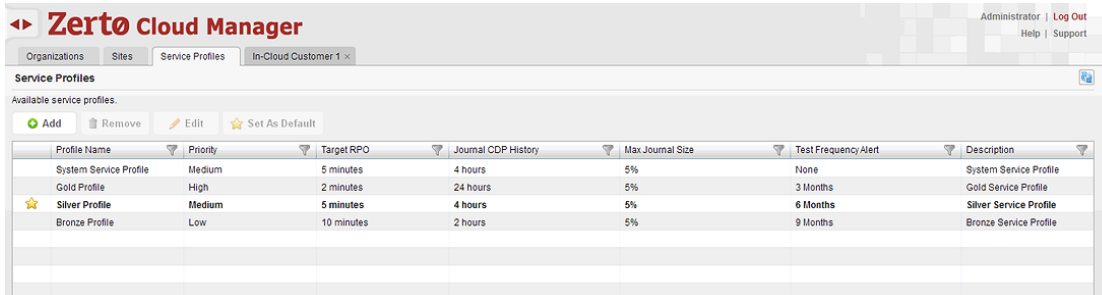
As self-service customers grow to be a larger percentage of the CSP customer base, streamlined workflows and repeatable processes are needed to better meet customer expectations and keep CSP costs low.

Service Profiles give policy-based management and automation capabilities to CSPs to ensure SLAs and service offerings are always consistent. Service Profiles reduce the administrative effort of the CSP by providing a customer-initiated capability to protect VMs.

Service Profiles enable a CSP to define structured service offerings with specific SLA parameters, including RPO, journal maximum size, history, and service level expectations. New Service Profiles can be added to a pool of existing Service Profiles that the CSP has predefined. These profiles make self-service much simpler, decreasing the learning curve for CSP customers, who simply select a profile from a drop-down list. Customers are also able to choose or update the Service Profile selection at the VPG creation stage if they have the proper permissions.

As seen in [Figure 3-3](#), a CSP may have three Service Profiles: Gold, Silver and Bronze. These Service Profiles are created in the ZCM and can be presented to multi-tenant customers. Service Profiles are controlled with permissions set by the CSP to limit customer profile selections to only predefined profiles, or create their own custom Service Profile.

Figure 3-3 Service Profiles



Profile Name	Priority	Target RPO	Journal CDP History	Max Journal Size	Test Frequency/Alert	Description
System Service Profile	Medium	5 minutes	4 hours	5%	None	System Service Profile
Gold Profile	High	2 minutes	24 hours	5%	3 Months	Gold Service Profile
Silver Profile	Medium	5 minutes	4 hours	5%	6 Months	Silver Service Profile
Bronze Profile	Low	10 minutes	2 hours	5%	9 Months	Bronze Service Profile

Enablement for Cloud DR Resource Consumption: Zerto Self Service Portal

DR requires an infrastructure level of integration between CSPs and customers. Depending on the service level requirements, cloud-based DR presents a unique challenge for CSPs because it often requires a two-way interaction that most cloud providers are not prepared to provide.

When customers want a fully managed service, the CSP manages both sides of the DR as their own administrative resources can readily meet that need. However, when customers want a more interactive hybrid DR service that requires that both CSP and the customer have infrastructure level administrative access, the CSP often has to create a customized DR portal to meet the customer access needs.

To help CSPs overcome the challenge of having to develop a custom portal just for DR, Zerto created the Zerto Self Service Portal (ZSSP). The ZSSP gives customers streamlined access to administrative functions and provides CSPs a way to quickly deploy a complete cloud-based DR solution.

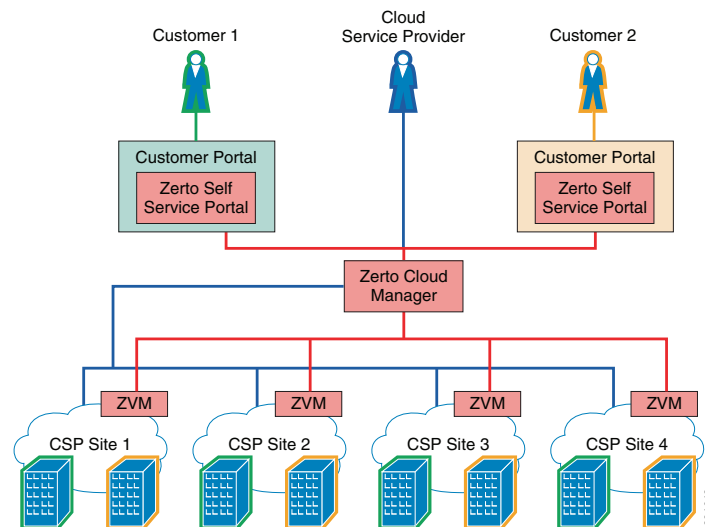
The ZSSP is designed to be an out-of-the-box DR portal solution. Having a fully functioning browser-based service portal available without a great deal of coding or scripting enables CSPs to quickly introduce DR as part of their existing portal or as a stand-alone portal. CSPs are able to quickly offer a robust DR service for faster ROI.

ZSSP Features

The ZSSP incorporates all of the APIs that were commonly requested by CSPs in production. Providing these APIs enables the CSP to rapidly roll out a more automated client experience.

ZCM enables the ZSSP by providing CSPs with the capability to offer a single point-of-view portal for their customers to view the status of their SLAs and manage the DR or migration status of their VMs regardless of the actual location of those VMs.

Figure 3-4 ZVM and the Zerto Self Service Portal



Being browser-based, the ZSSP enables unprecedented management of business continuity and disaster recovery. Administrators can monitor service levels, perform non-disruptive tests, and perform actual failovers from many different devices, including many mobile devices.

Storage

Storage is the main component in the Cisco DRaaS 2.0 system. Proper storage sizing and deployment is critical for delivering an optimized service to customers. The following storage efficiency feature is recommended at the CSP recovery site:

- **Thin Provisioning**—Thin provisioning is a good method for optimizing utilization of available storage. It relies on an on-demand allocation of blocks of data versus the traditional method of allocating all the blocks up front. This method eliminates all the unused space, which increases utilization efficiencies. The best practice is to enable thin provisioning at the storage level or at the hypervisor level to avoid management challenges.

**Note**

In the DRaaS 2.0 system, because Zerto is capable of creating VMs using thin provisioning in the cloud, it is recommended to implement thin provisioning at the hypervisor layer.

The following storage efficiency features are specific to EMC VNX when using vBlock as the Integrated Compute Stack (ICS) architecture:

- **FAST Cache**—EMC FAST Cache technology is an extension of existing DRAM cache in that it allocates certain flash drives to serve as FAST Cache. The benefit is that hotter data from applications running inside the VM will be copied to FAST Cache. Hence, these applications will see improved response time and throughput since the I/O is now serviced from flash drives. In DRaaS environments, FAST Cache is useful during concurrent customer site failovers and during the onboarding of new customers. In general, FAST Cache should be used in cases where storage performance needs to improve immediately for I/O that is burst-prone.
- **FAST VP**—Data has a lifecycle and as data progresses through this lifecycle, it experiences varying levels of activity. When data is created, typically it is accessed very frequently. As it ages, it is accessed less often. This is often referred to as data being *temporal* in nature. EMC FAST VP is a simple and elegant solution for dynamically matching storage requirements with changes in the frequency of data access. FAST VP segregates disk drives into the following three tiers: Extreme Performance Tier (flash drives); Performance Tier (Serial Attached SCSI (SAS) drives for EMC VNX); and Capacity Tier (Near-Line SAS (NL-SAS) drives for EMC VNX platforms).
 - You can use FAST VP to reduce TCO and/or to increase performance. A target workload requiring a large number of Performance Tier drives can be serviced with a mix of tiers and a much lower drive count. In some cases, nearly two-thirds reduction in drive count is achieved. In other cases, performance throughput can double simply by adding no more than 10% of a pool's total capacity in flash drives.
 - FAST VP and FAST Cache can be used together to improve storage system performance.
 - Customers with a limited number of flash drives can create FAST Cache and storage pools consisting of performance and capacity drives. For performance, FAST Cache will provide immediate benefits for any burst-prone data, while FAST VP will move warmer data to performance drives and colder data to capacity drives.
 - FAST Cache is storage system-aware so that storage system resources are not wasted by unnecessarily copying data to FAST Cache if it already exists on flash drives. If FAST VP moves a slice of data to the Extreme Performance Tier, FAST Cache will not promote that slice into FAST Cache - even if it has met the FAST Cache criteria for promotion.
 - When initially deploying flash drives in a storage system, the recommendation is to use them for FAST Cache. FAST Cache will track I/Os smaller than 128 KB and requires multiple cache hits to 64 KB chunks. This will initiate promotions from performance or capacity drives to Flash Cache and, as a result, I/O profiles that do not meet this criteria are better served by flash drives in a pool or RAID group.

The following storage efficiency features are specific to NetApp when using FlexPod as an integrated stack within VMDC:

- **Flash Cache**—NetApp Flash Cache speeds access to data through real-time intelligent caching of recently read user data and NetApp metadata. It is effective for random read-intensive workloads, including database, e-mail, and file services. The combination of intelligent caching and NetApp data storage efficiency technologies enables the virtual storage tier, which promotes hot data to performance media in real time without moving the data, allowing you to scale performance and capacity while achieving the highest level of storage efficiency in the industry.

- **Flash Pool**—Flash Pool is a technology that allows flash technology in the form of solid-state disks (SSDs) and traditional hard disk drives (HDDs) to be combined to form a single Data onTap aggregate. When SSD and HDD technologies are combined in a Data onTap aggregate, the NetApp storage system takes advantage of the latency and throughput benefits of SSD while maintaining the mass storage capacity of the HDD.
 - A Flash Pool is built from a Data onTap aggregate in a two-step process. Essentially, it is the addition of SSDs into an aggregate that provides a high-bandwidth, low-latency location that is capable of caching random reads and random overwrites.

**Note**

This feature does not require a license and works with any NetApp SSDs and a consistent type of HDD per Flash Pool. That is, SSD and SAS performance drives can be combined to make a Flash Pool or SSD and SATA capacity drives can be combined to make a Flash Pool. You cannot combine SSD, SAS, and SATA into a single Flash Pool.

- As a key component of the NetApp Virtual Storage Tier, Flash Pool offers a real-time, highly efficient implementation of automated storage tiering. Fine-grain promotion of hot data elements, combined with data deduplication and thin cloning, enables optimal performance and optimal use of flash storage technology.
- **De-duplication**—NetApp de-duplication is an integral part of the NetApp Data onTap operating environment and the WAFL file system, which manages all data on NetApp storage systems. De-duplication works "behind the scenes," regardless of what applications you run or how you access data, and its overhead is low.
 - NetApp de-duplication is a key component of NetApp's storage efficiency technologies, which enable users to store the maximum amount of data for the lowest possible cost.
 - NetApp de-duplication is a process that can be triggered when a threshold is reached, scheduled to run when it is most convenient, or run as part of an application. It will remove duplicate blocks in a volume or LUN.

In summary, steady-state storage considerations include:

- FAST VP from EMC.
- Flash Pool from NetApp.
- During the steady state replication, the target storage will have the information about the I/ O characteristics and data blocks.
- NetApp Flash Cache and EMC FAST Cache are useful in dealing with unpredicted I/O needs that can be observed during the recovery of multiple customer environments during a disaster.
- NetApp Flash Pool and EMC FAST VP are useful efficiency features that help the CSP to use storage space more efficiently during a steady-state replication scenario. Warmer data gets moved to the faster drives and cold data gets moved to the capacity disks automatically.
- NetApp de-duplication and storage thin provisioning reduces the total storage footprint required to support customer workloads.

Compression

To ensure efficient use of the WAN between sites, replication data sent from one site to another should be compressed before it is sent. This helps to reduce the WAN bandwidth required for data replication. This can be accomplished by using a dedicated external device or by using technologies that are incorporated in the DRaaS 2.0 solution, such as the integrated compression capability available in ZVR.

ZVR can perform data compression, which is a good option for customers who do not want to have a dedicated device for this functionality. It is an ideal choice for customers who have fewer servers being protected.

However, there are advantages of going with an external dedicated compression appliance, including:

- Better handling of data compression and management as the dedicated hardware will be used only for this functionality. This offloads the processing load from DR component that does the compression.
- Compression of non-DR related traffic, optimizing the overall WAN bandwidth usage.
- Easier troubleshooting of contention issues.

Dedicated Cisco WAN Optimization Products

Network links and WAN circuits are sometimes characterized by high latency, packet loss, and limited capacity. WAN optimization devices can be used to maximize the amount of replicated data that can be transmitted over a link.

A WAN Optimization Controller (WOC) is an appliance that can be placed in-line or out-of-path to reduce and optimize the data that is to be transmitted over the WAN. These devices are designed to help mitigate the effects of packet loss, network congestion, and latency while reducing the overall amount of data transmitted over the network. In general, the technologies utilized in accomplishing this are TCP acceleration, data deduplication, and compression. WAN and data optimization can occur at varying layers of the OSI stack, whether they be at the Network and Transport Layers, the Session, Presentation, and Application layers, or just to the data (payload) itself.

Cisco Wide Area Application Services (WAAS) devices can be used for data optimization. The WAAS system consists of a set of devices called Wide Area Application Engines (WAE) that work together to optimize TCP traffic over the network. Cisco WAAS uses a variety of transport flow optimization (TFO) features to optimize TCP traffic intercepted by the WAAS devices. TFO protects communicating devices from negative WAN conditions, such as bandwidth constraints, packet loss, congestion, and retransmission. TFO includes optimization features such as compression, windows scaling, Selective ACK, increased buffering, BIC TCP, and TCP Initial Window Size Maximization.

Cisco WAAS uses Data Redundancy Elimination (DRE) and LZ compression technologies to help reduce the size of data transmitted over the WAN. These compression technologies reduce the size of transmitted data by removing redundant information before sending a shortened data stream over the WAN. By reducing the amount of transferred data, WAAS compression reduces network utilization and application response times.

When a WAE uses compression to optimize TCP traffic, it replaces repeated data in the stream with a much shorter reference and then sends the shortened data stream out across the WAN. The receiving WAE uses its local redundancy library to reconstruct the data stream before passing it along to the destination. The WAAS compression scheme is based on a shared cache architecture in which each WAE involved in compression and decompression shares the same redundancy library. When the cache that stores the redundancy library on a WAE becomes full, WAAS uses a FIFO algorithm (first in, first out) to discard old data and make room for new.



Note

For more information about Cisco WAAS technologies, visit: <http://www.cisco.com/go/waas>.

Zerto Virtual Replication

Compression within ZVR is enabled by a simple checkbox when configuring the VPG. Zerto and Cisco tested the Zerto compression capability and the results exceeded an average of 50% bandwidth savings between sites, depending on the compressibility of the data. Each VRA that operates on each host in the VMware cluster is responsible for the compression. Having this distributed model of compression minimizes the CPU and RAM impact on the host system.

Encryption

Encryption of data-in-transit and data-at-rest is the best method of enforcing the security and privacy of data, regardless of where it resides. Data-in-transit encryption is necessary to keep the data secure while in transit. The network connection between sites must be secure and the data must be protected. The use of IPsec or SSL to encrypt WAN connections ensures that no visibility occurs at the packet level if any of the datagrams are intercepted in transit.

ZVR does not support encryption natively. Encryption of data-in-transit between the sites can be accomplished using an external device, such as the Cisco Adaptive Security Appliance (ASA). The Cisco ASA 55xx Series is a purpose-built platform that combines superior security and VPN services for enterprise applications. The Cisco ASA 55xx Series enables customization for specific deployment environments and options, with special product editions for secure remote access (SSL/IPSec VPN).

The Cisco ASA 55xx Series SSL/IPsec VPN Edition uses network-aware IPsec site-to-site VPN capabilities. This allows customers to extend their networks securely across low-cost Internet connections to the CSP site.

Encryption of data-at-rest can add further security to the storage environment on the CSP's data center. Any external key manager can be used in conjunction with SAN fabrics and storage arrays to secure data-at-rest.

In the control plane, ZVR uses HTTPS to encrypt communications with other components in the system, including:

- Access to the ZVR management UI via the vSphere Client console.
- Communication between the Zerto Virtual Manager and the vCenter Server.
- Communication between the Zerto Virtual Manager and vCloud Connector.
- Communication between the Zerto Virtual Manager and the ESX/ESXi hosts.

ZVR Disaster Recovery Workflow

Zerto Virtual Replication provides a number of operations to recover VMs at the peer site, as shown in the following sections.

The Move Operation

Use the Move operation to migrate protected VMs from the protected (source) site to the recovery (target) site in a planned migration.

When you perform a planned migration of the VMs to the recovery site, Zerto Virtual Replication assumes that both sites are healthy and that you planned to relocate the VMs in an orderly fashion without data loss.

The Move operation follows these basic steps:

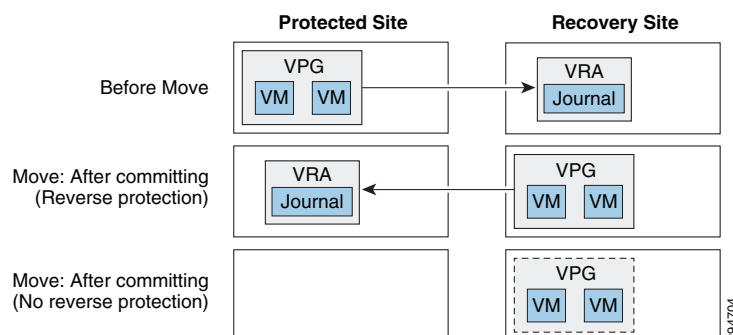
-
- Step 1** Gracefully shut down the protected VMs to ensure data integrity. If the machines cannot be gracefully shut down, for example, when VMware Tools is not available, an Administrator can manually shut down the machines before starting the Move operation or specify as part of the operation a forced power off the VMs. If the machines cannot be gracefully shut down automatically and are not manually shut down and the Move operation is not set to forcibly power them off, the Move operation stops and ZVR rolls back the VMs to their original status.
- Step 2** Insert a clean checkpoint. This avoids potential data-loss since the VMs are not powered on and the new checkpoint is created after all I/Os have been written to disk.
- Step 3** Transfer to the recovery site all the latest changes that are still being queued to pass to the recovery site, including the new checkpoint.
- Step 4** Create the VMs at the remote site in the production network and attach each VM to its relevant disks, based on the checkpoint inserted in Step 2.
- Step 5** Power on the VMs in the recovery site making them available to the user. If applicable, the boot order defined in the VPG settings is used to power on the machines in a specified order.
- Step 6** Run basic tests on the machines to ensure their validity to the specified checkpoint. Depending on the commit/rollback policy that was specified for the operation after testing, either the operation is committed—finalizing the Move—or rolled back, aborting the operation. You can also configure the move operation to automatically commit the move, without testing.
- Step 7** The source VMs are removed from the inventory.
- Step 8** The data from the journal is promoted to the machines. The machines can be used during the promotion and ZVR ensures that the user sees the latest image, even if this is partially data from the journal. That is, when accessing the migrated VM, ZVR can present data both from the disks and from the journal, to ensure that information is current.
-

If reverse replication was specified—the disks used by the VMs in the source site are used for the reverse protection. A Delta Sync is performed to make sure that the two copies—the new target site disks and the original source site disks—are consistent.

If reverse replication was not specified—the VPG definition is saved but the state is left at “Needs Configuration” and the disks used by the VMs in the source site are deleted. Thus, if reverse protection is not set the original disks are not available and a full synchronization will be required.

Figure 3-5 shows the positioning of the VMs before and after the completion of a Move operation.

Figure 3-5 ZVR Move Operation



**Note**

The Move operation without reverse protection does not remove the VPG definition but leaves it in a “Needs Configuration” state.

The Failover Operation

Use the Failover operation following a disaster to recover protected VMs to the recovery site. A failover assumes that connectivity between the sites might be down, and thus the source VMs and disks are not removed, as they are in a planned Move operation.

When you set up a failover, you always specify a checkpoint to which you want to recover the VMs. When you select a checkpoint—either the latest auto-generated checkpoint, an earlier checkpoint, or a user-defined checkpoint—ZVR makes sure that VMs at the remote site are recovered to this specified point-in-time.

**Note**

To identify the checkpoint to use, you can perform a number of consecutive test failovers, each to a different checkpoint until the desired checkpoint for recovery is determined.

The Failover operation has the following basic steps:

Step 1 Create the VMs at the remote site in the production network and attach each VM to its relevant disks, configured to the checkpoint specified for the recovery.

**Note**

The source VMs are not touched since the assumption is that the production site is down.

Step 2 Power on the VMs, making them available to the user. If applicable, the boot order, defined in the VPG settings to power on the machines in a specified order, is used.

Step 3 Run basic tests on the machines to ensure their validity to the specified checkpoint. Depending on the commit/rollback policy that was specified for the operation after testing, either the operation is committed—finalizing the Move—or rolled back, aborting the operation. You can also configure the failover operation to automatically commit the move, without testing.

Step 4 If the source site is still available, for example after a partial disaster, and reverse protection is possible and specified for the failover operation, the source VMs are powered off and removed from the inventory. The disks used by the VMs in the source site are then used for the reverse protection. A Delta Sync is performed to make sure that the two copies, the new target site disks and the original source site disks, are consistent.

**Note**

If reverse protection is not possible, or reverse protection is configured to not use the original disks, the source site VMs are not powered off and are instead removed. In the latter case, if possible, the VMs should be shut down manually before starting the failover.

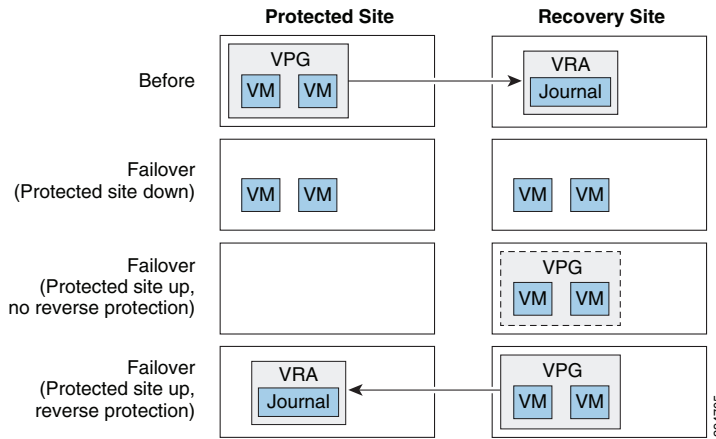
Step 5 The data from the journal is promoted to the machines. The machines can be used during the promotion and ZVR ensures that the user sees the latest image, even if this is partially data from the journal.

Failback after the Original Site is Operational

To perform a failback to the source site, the VPG that is now protecting the VMs on the target site has to be configured. A Delta Sync is then performed with the disks in the source site. Once the VPG is in a protecting state the VMs can be moved back to the source site.

Figure 3-6 shows the positioning of the VMs before and after the completion of a Failover operation

Figure 3-6 ZVR Failback Operation



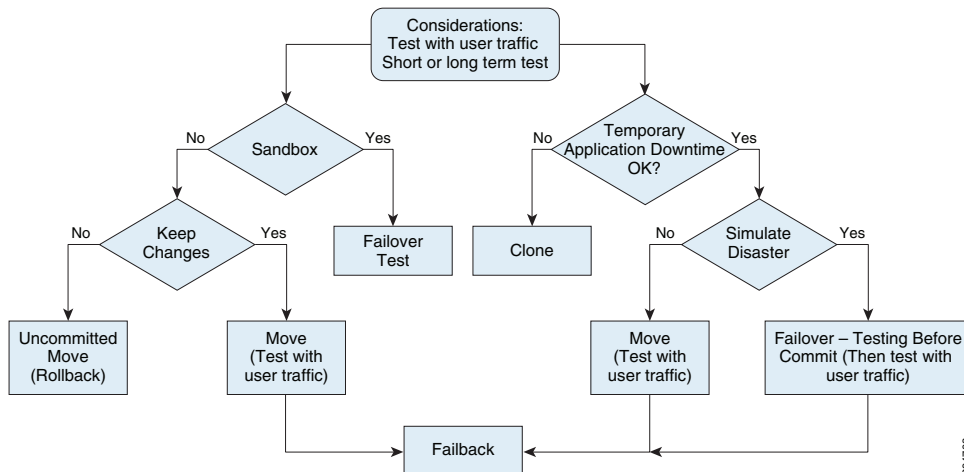
Note

The Failover operation without reverse protection does not remove the VPG definition but leaves it in a “Needs Configuration” state.

Disaster Recovery Workflow

Figure 3-7 shows the disaster recovery testing workflow.

Figure 3-7 Testing Disaster Recovery Workflow



During any live test, it is recommended not to maintain two working versions of the same VMs. Thus, the first step in any test, except for a Failover Test or Clone, is to make sure that the production virtual machines are shut down before starting to test recovered machines. During a Zerto Virtual Replication

Move operation, the first step Zerto Virtual Replication performs is to shut down the protected machines, to ensure data integrity. However, a Zerto Virtual Replication Failover operation assumes that the production VMs are no longer accessible (the total site disaster scenario) and does not attempt by default to shut them down at the beginning of the operation.

In a live test using a failover operation, you have to specify that you want to shut down the VMs to be tested at the beginning of the test to prevent potential split-brain situations where two instances of the same applications are live at the same time.

If you want to perform a live DR test that includes a simulated disaster you can simulate the disaster by, for example, disconnecting the network between the two sites. In this type of test, once the disaster is simulated a Move operation cannot be used, since it requires both sites to be healthy, while a Failover operation can be used.

Best Practices

The following best practices are recommended:

- Prepare an administrator account for the machine where ZVR is installed.
- Install ZVR on a dedicated VM with a dedicated administrator account and with VMware High Availability (HA) enabled and no other applications installed on the VM. If other applications are installed, the Zerto Virtual Manager service must receive enough resources and HA must remain enabled.
- Install a VRA on every host in a cluster so that if protected VMs are moved from one host to another, there is always a VRA to protect the moved VMs. When protecting a vApp, you must install a VRA on every host in the cluster on both the protected and recovery sites and ensure that DRS is enabled for the clusters.
- Install VRAs using static IP addresses and not DHCP.
- Set any antivirus software not to scan the folder where ZVR is installed.
- Ensure the clocks on the machines where ZVR is installed are synchronized using NTP.



Note

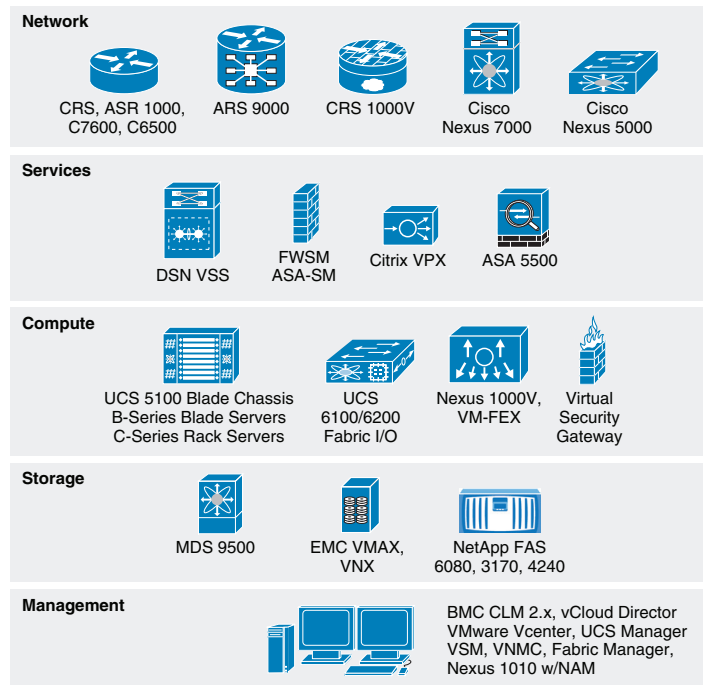
There is much more information available regarding Zerto operation, implementation, and monitoring in the Cisco DRaaS 2.0 Implementation Guide, available at: <http://www.cisco.com/go/draas>.



Architecture Configuration

The Cisco DRaaS 2.0 solution is layered on top of the Cisco VMDC VSA 1.0¹ cloud reference architecture, which provides the basic framework and technologies required to support multi-tenant cloud implementations. The basic functional layers consistent to the VMDC architectures are shown in [Figure 4-1](#). Though all of these functional layers were necessarily a part of the architecture for this project, not all of the components shown were in play.

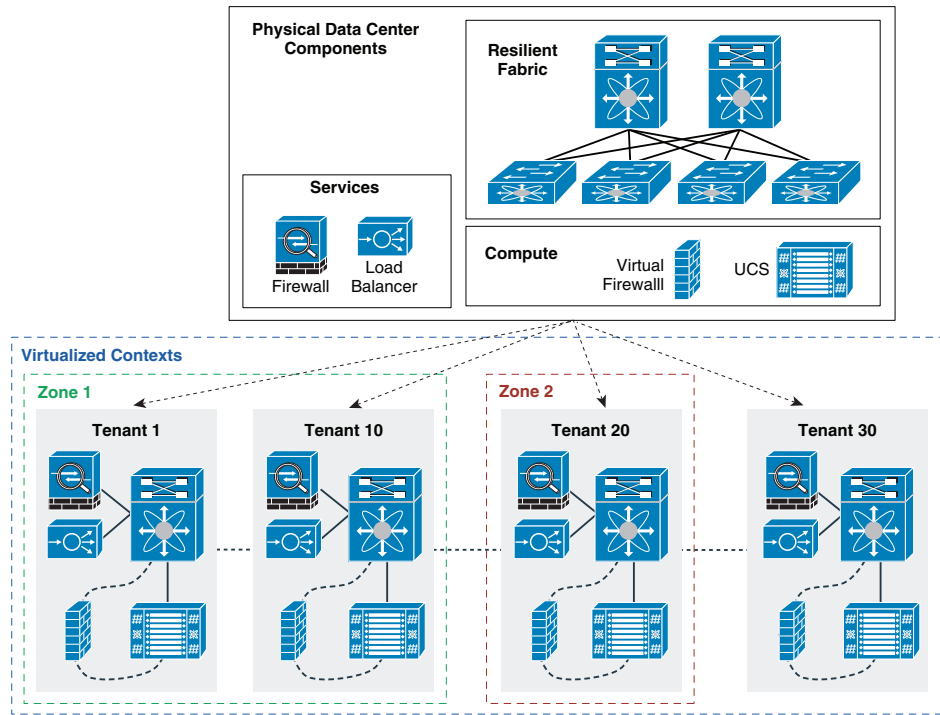
Figure 4-1 Functional Layers Within the VMDC Data Center



The basic tenancy model is illustrated in [Figure 4-2](#). It consists of a network fabric (Layer 2 & 3 switching), Layer 4-7 services (firewall and load balancing, e.g.), and compute. From this infrastructure, multiple virtualized contexts—tenants—are logically created.

1. Virtualized Multiservice Data Center Virtual Services Architecture (version) 1.0

Figure 4-2 Multitenancy Design



VMDC VSA 1.0 System Architecture

There are various tenant container models available, validated, and supported in the VSA 1.0 architecture, which are differentiated by the services that they provide to the individual tenants. [Figure 4-3](#) provides an overview of these tenant container models. For this project, the Bronze Container model was used at the SP site to provide tenancy to the Enterprise customer. The details of the Bronze Container model are illustrated in [Figure 4-4](#).

Figure 4-3 VMDC VSA 1.0 Container Models

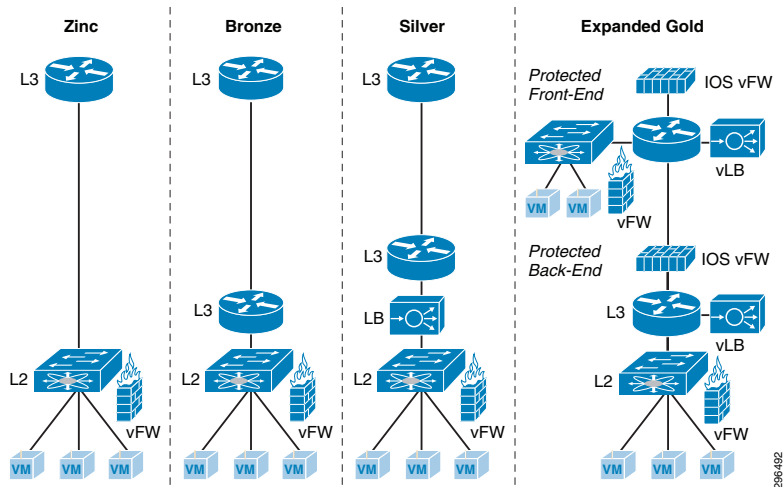
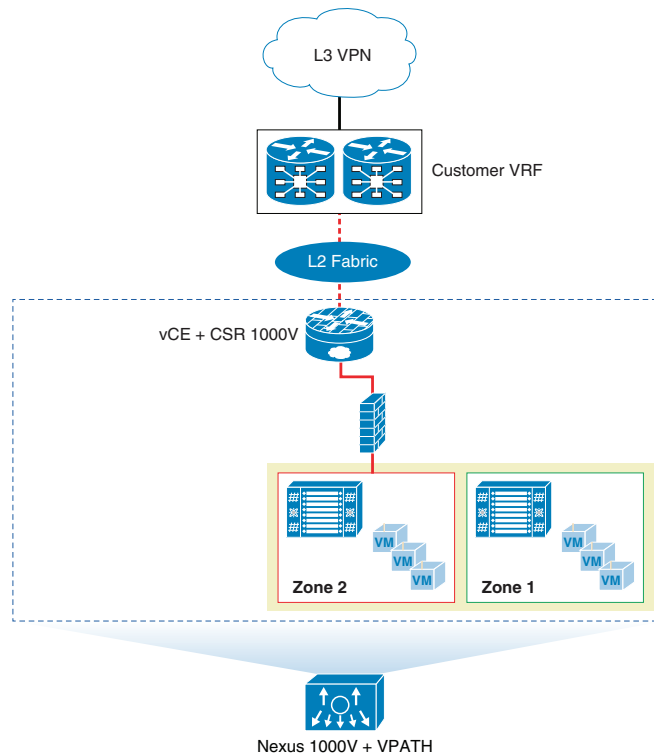


Figure 4-4 VMDC VSA 1.0 Bronze Container Model



Additional Resources

For additional information on the Cisco VMDC program, including summaries, overviews, and white papers, see the following link:

- <http://www.cisco.com/go/vmdc>

For information on the Cisco VMDC VSA 1.0 architecture design specifically, see this link:

- http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Data_Center/VMDC/DCI/1-0/DG/DCI.html

Cisco UCS

Each of the three C240-M3S hosts used in this project as Virtual SAN cluster nodes were configured identically. Each was equipped with the following hardware:

- Dual Intel 2.60 GHz 6-core processors
- 256GB RAM
- Four 1 TB SATA 7200 RPM magnetic drives
- One 400 GB enterprise-class SAS SSD
- Cisco VIC 1225 PCI-E adapter
- MegaRAID controller

Each UCS C240-M3S was connected to upstream to a pair of UCS 6248UP fabric interconnects which were connected, in turn, to a pair of Nexus 5548UP switches. Six vNICs were configured for each host, as follows:

- **Data/Inband**—Two vNICs were configured to carry the data VLANs necessary for this project
- **Management/OOB**—Two vNICs were configured to carry management traffic, not including vMotion or Virtual SAN control/data traffic
- **vMotion**—One vNIC was dedicated to vMotion operations, with a corresponding VMkernel NIC configured as well
- **Virtual SAN**—One vNIC was dedicated to Virtual SAN control and data traffic, with a VMkernel NIC to match

Hardware and Software

Table 4-1 provides a list of the hardware used for this project. Each of the three Virtual SAN cluster nodes was identical to this configuration.

Table 4-1 UCS Hardware Configuration

Component	Quantity	Cisco Part Number	Description
Chassis	1	UCSC-C240-M3S	UCS C240 M3 SFF w/o CPU, mem, HD, PCIe, w/ rail kit, expdr
Processor	2	UCS-CPU-E52630B	2.60 GHz E5-2630 v2/80W 6C/15MB Cache/DDR3 1600MHz
Memory	16	UCS-MR-1X162RY-A	16GB DDR3-1600-MHz RDIMM/PC3-12800/dual rank/1.35v
Hard Drives (Magnetic)	4	A03-D1TBSATA	1TB 6Gb SATA 7.2K RPM SFF HDD/hot plug/drive sled mounted
Hard Drives (Flash)	1	UCS-SD400G0KS2-EP	400GB 2.5 inch Enterprise Performance SAS SSD
RAID Controller	1	UCS-RAID9271CV-8I	MegaRAID 9271CV with 8 internal SAS/SATA ports with Supercap
PCI Express Card	1	UCSC-PCIE-CSC-02	Cisco VIC 1225 Dual Port 10Gb SFP+ CAN

Table 4-2 provides a summary of UCS components with the software versions in use during this project:

Table 4-2 UCS Software Versions

Component	Version
UCS Manager	2.2(1b)
Cisco UCS 6248UP	5.2(3)N2(2.21b)
Cisco UCS VIC 1225 (Adapter)	2.2(1b)
CIMC Controller	1.5(4)

Service Profile Configuration

As discussed elsewhere in this paper, service profile templates assist the compute administrator by allowing for the deployment of multiple physical servers from a single configuration file. A single service profile template was built and used to deploy the three UCS C240 Virtual SAN cluster nodes used in this project. Some of the select settings used to build the service profile are summarized in [Table 4-3](#).

Table 4-3 UCS Service Profile Settings

Configuration Tab	Element	Setting	Purpose
Storage	Local Disk Configuration Policy		
	Mode	Any Configuration	No RAID configuration on local disks (MD & SSD)
	Mode: FlexFlash State	Enabled	Enable the use of SD FlexFlash card
	Mode: SAN Connectivity Policy	<not set>	
Network	vNIC #1: Name	Data 1	Carry inband data VLANs
	vNIC #1: MAC Address	<from pool>	
	vNIC #1: Fabric ID	Fabric A	
	vNIC #1: Enable Failover	No	
	vNIC #1: VLANs	Default (Native), 102, 944, 2484, 2840-2843	
	vNIC #1: MTU	1500	
	vNIC #2: Name	Data 2	Carry inband data VLANs
	vNIC #2: MAC Address	<from pool>	
	vNIC #2: Fabric ID	Fabric B	
	vNIC #2: Enable Failover	No	
	vNIC #2: VLANs	Default (Native), 102, 944, 2484, 2840-2843	
	vNIC #2: MTU	1500	
	vNIC #3: Name	Mgmt 1	Carry OOB management VLANs
	vNIC #3: MAC Address	<from pool>	
vNIC #3: Fabric ID	Fabric A		
vNIC #3: Enable Failover	No		
vNIC #3: VLANs	Default (Native), 101, 13, [kvm-vlan]		
vNIC #3: MTU	1500		

Table 4-3 UCS Service Profile Settings (continued)

Configuration Tab	Element	Setting	Purpose
	vNIC #4: Name	Mgmt 2	Carry OOB management VLANs
	vNIC #4: MAC Address	<from pool>	
	vNIC #4: Fabric ID	Fabric B	
	vNIC #4: Enable Failover	No	
	vNIC #4: VLANs	Default (Native), 101, 13, [kvm-vlan]	
	vNIC #4: MTU	1500	
	vNIC #5: Name	vMotion	VLAN used for vMotion VM migration
	vNIC #5: MAC Address	<from pool>	
	vNIC #5: Fabric ID	Fabric A	
	vNIC #5: Enable Failover	No	
	vNIC #5: VLANs	Default (Native), 3203	
	vNIC #5: MTU	1500	
	vNIC #6: Name	VSAN	VLAN used for Virtual SAN control packets
	vNIC #6: MAC Address	<from pool>	
	vNIC #6: Fabric ID	Fabric A	
	vNIC #6: Enable Failover	No	
	vNIC #6: VLANs	Default (Native), 3201	
	vNIC #6: MTU	1500	
Boot Order	Boot Policy	Boot Order	1: Local CD/DVD 2: SD Card ¹

1. For this project, the VMware ESXi hypervisor image was booted from the on-board SD card. For information on SAN booting the UCS C240, see the Cisco DRaaS 2.0 Implementation Guide.

VMware Virtual SAN

The documentation below covers the administratively-defined aspects of deploying a Virtual SAN datastore.

Host Networking

The three UCS C240-M3S hosts were deployed using UCS Manager with six vNICs. The six vNICs were configured to carry various VLANs, as shown in [Table 4-4](#). [Figure 4-5](#) shows the vNICs that were configured to carry Virtual SAN and vMotion traffic.

Table 4-4 UCS Host vNIC VLAN Configuration

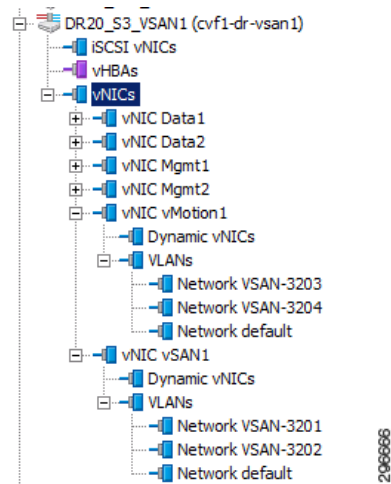
Name	vNIC	VLANs
NIC 1	Mgmt1	1, 13, 101
NIC 2	Mgmt2	1, 13, 101
NIC 3	Data1	1, 102, 254, 944, 2840-2842
NIC 4	Data2	1, 102, 254, 944, 2840-2842
NIC 5	vMotion1	1, 3203
NIC 6	vSAN1	1, 3201



Note

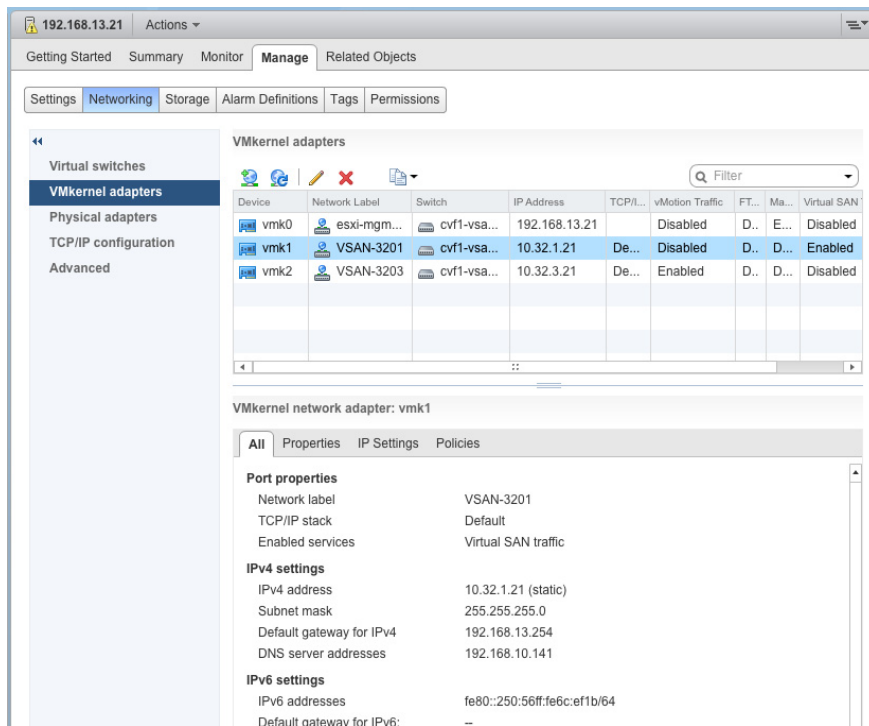
The vNICs in listed in the table carried, in practice, many more VLANs than are shown as the hosts were deployed in a test environment with the potential to be used for other testing. The VLANs shown in [Table 4-4](#) are limited to those relevant to this project.

Figure 4-5 UCS Host vNIC VLAN Configuration



For Virtual SAN networking to be established, Virtual SAN needs to be enabled on an existing or a new VMkernel NIC (VMKNIC). To keep the Virtual SAN service separate from other services, a dedicated VMKNIC was configured with only Virtual SAN enabled, as shown in [Figure 4-6](#).

Figure 4-6 Virtual SAN VMkernel Configuration



The subnet chosen for Virtual SAN communication was 10.32.1.0/24, and the ESXi VMKNICs were assigned IP addresses according to [Table 4-5](#).

Table 4-5 Host Virtual SAN VMkernel NIC IP Addressing

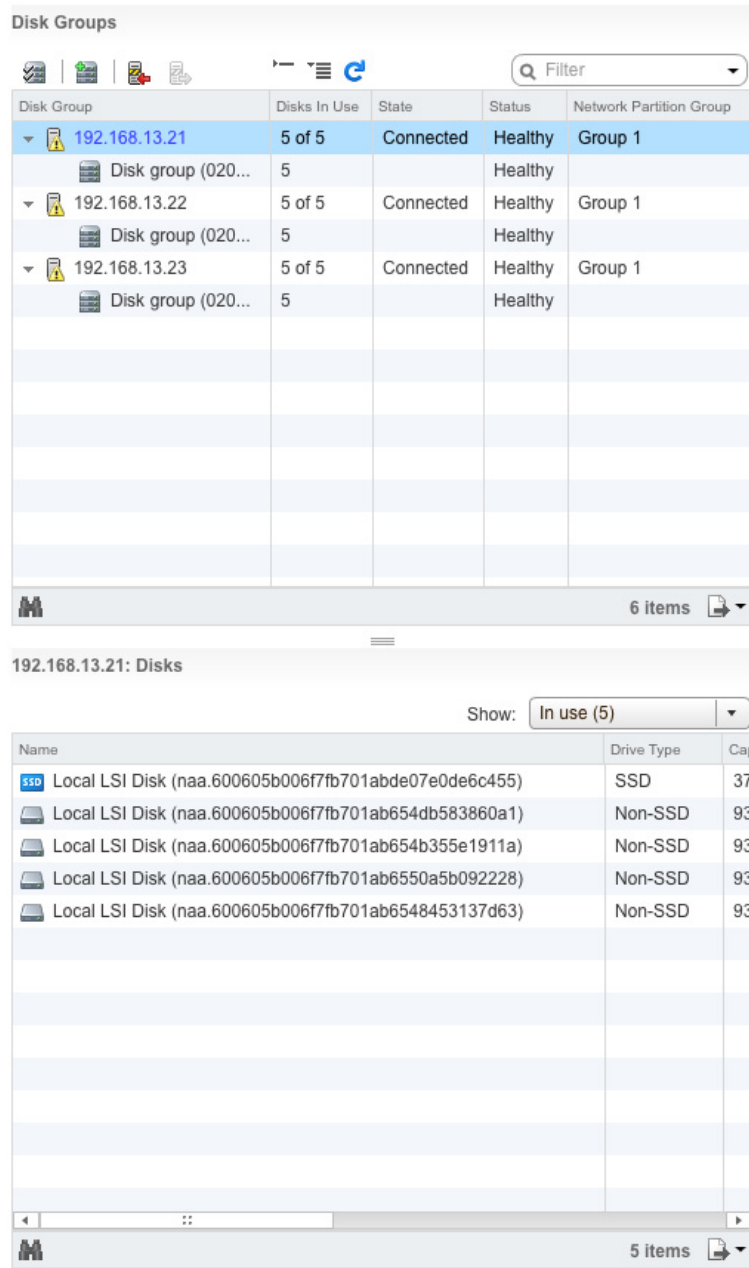
Host	VMKNIC	IP Address	VLAN	Mgmt	Virtual SAN	Vmotion
cvf1-dr-vsan1	vmk0	192.168.13.21	13	X		
	vmk1	10.32.1.21	3201		X	
	vmk2	10.32.3.21	3203			X
cvf1-dr-vsan2	vmk0	192.168.13.22	13	X		
	vmk1	10.32.1.22	3201		X	
	vmk2	10.32.3.22	3203			X
cvf1-dr-vsan3	vmk0	192.168.13.23	13	X		
	vmk1	10.32.1.23	3201		X	
	vmk2	10.32.3.23	3203			X

Disk Group Creation

Each of the three UCS C240-M3S hosts used in this Virtual SAN cluster had one SSD and four MDs (Magnetic Disks) and each host was configured with a single disk group. The SSD was 400GB while the MDs were 1TB each. Thus the total MD data capacity was 4TB. The ratio of SSD:MD capacity was 1:10, which is aligned with VMware Virtual SAN recommendations.

Figure 4-7 is a screenshot of the Virtual SAN disk group configuration. We can see from the vSphere web client that each ESXi host has one disk group and five disks. Each host belongs to Network Partition Group 1, indicating that all hosts are communicating on the Virtual SAN network. We can also see that in one of those disk groups (host 192.168.13.21) there are indeed four MDs for data and one SSD for cache.

Figure 4-7 Virtual SAN Disk Group Verification



In the summary view in Figure 4-8, we can see that all 12 data disks are in use (four from each host) and the total capacity of the datastore is 10.9 TB.

Figure 4-8 Virtual SAN Resources Summary

Virtual SAN is Turned ON Edit...	
Add disks to storage	Manual
Resources	
Hosts	3 hosts
SSD disks in use	3 of 3 eligible
Data disks in use	12 of 12 eligible
Total capacity of VSAN datastore	10.9 TB
Free capacity of VSAN datastore	8.37 TB
Network status	✓ Normal

From the Summary view of the Virtual SAN datastore, we can also see the 10.9 TB capacity available as well as other specifics for the datastore (Figure 4-9).

Figure 4-9 Virtual SAN Datastore Summary

The screenshot displays the VMware vSAN Datastore Summary view. At the top, there is a navigation bar with 'vsanDatastore' and 'Actions'. Below this, there are tabs for 'Getting Started', 'Summary', 'Monitor', 'Manage', and 'Related Objects'. The main content area shows a storage bar with 'STORAGE' and 'FREE: 8.37 TB' labels, and 'USED: 2.53 TB' and 'CAPACITY: 10.9 TB' labels. A 'Refresh' button is visible. Below the storage bar, there is a 'Details' pane with the following information:

Details	
Location	ds://vmfs/volumes/vsan:524ecb147b2f83e1-4d73aac144b318c2/
Type	vsan
Hosts	3
Virtual machines	5

Below the details pane, there is a 'Tags' pane with the following information:

Assigned Tag	Category	Description
This list is empty.		

At the bottom of the tags pane, there are 'Assign...' and 'Remove...' buttons.

Policy Configuration

For this project, all of the default VM Storage Policy settings described in Table 4-5 were used. For clarification, Table 4-6 summarizes these settings. These settings are also the VMware recommendations as of this writing.

Table 4-6 VM Storage Policy Settings

Policy	Value
Number of Disk Stripes Per Object	1
Flash Read Cache Reservation	0%
Number of Failures to Tolerate	1
Force Provisioning	Disabled
Object Space Reservation	0%

DRaaS Application Suite

The DRaaS application suite entails VM deployment on a virtual SAN data store for enterprise and SP data center configurations.

Implementation and Configuration of Zerto Virtual Replication

To verify the DRaaS 2.0 system architecture, a lab topology with two CSP sites and eight enterprise customer sites was configured. DRaaS was tested with the first four enterprises and ICDR was tested with the remaining four enterprises.

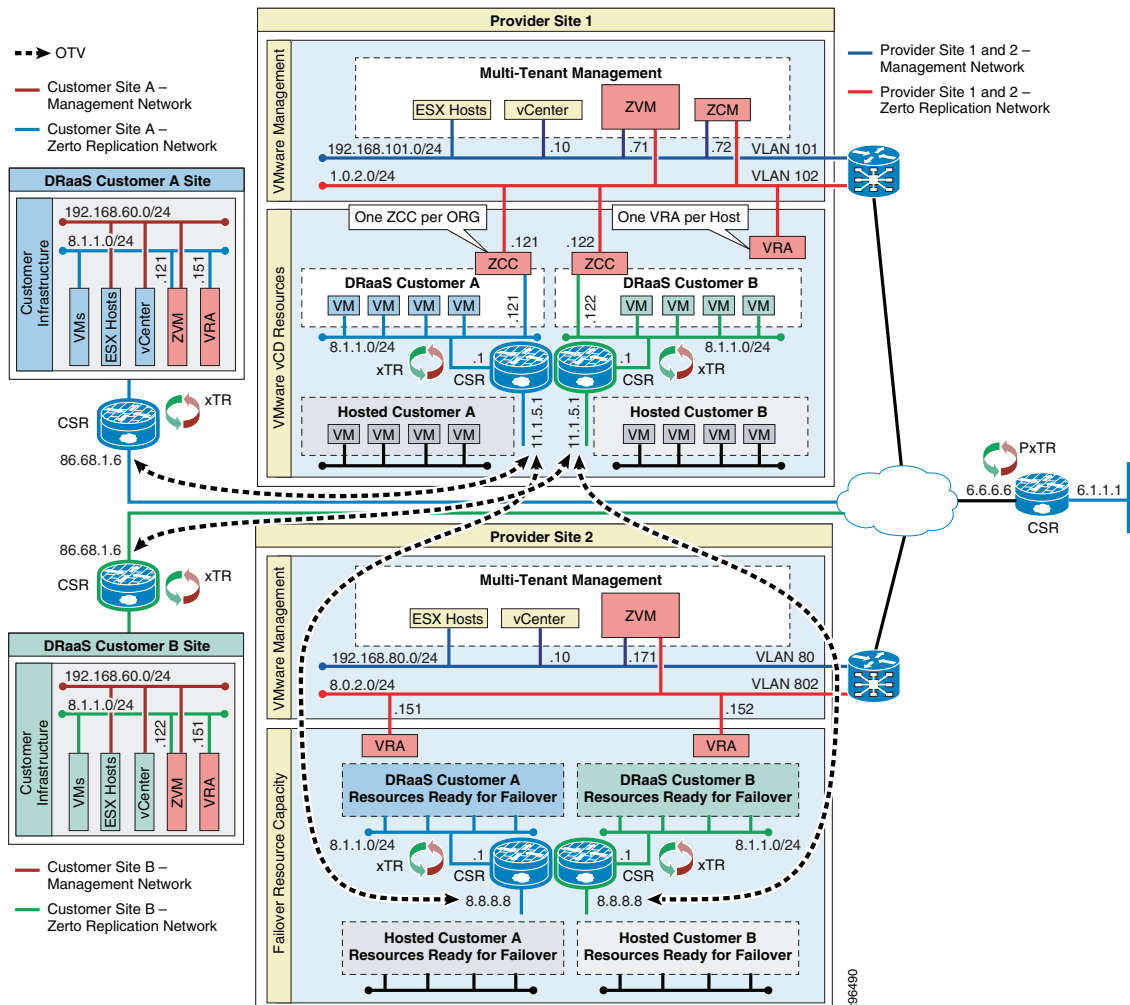
The DRaaS enterprises connected into the first provider site using OTV with CSR 1000V routers at each end. The OTV link carried both tenant traffic and Zerto control plane and replication traffic. The ICDR enterprises connected into the first provider site using OTV and had an OTV link between the provider sites for their tenant networks. For Zerto control plane and replication traffic between provider sites, native IP links and L3 routing were configured. These links allowed the following communication:

- CSP ZCM ↔ CSP/enterprise ZVMs
- CSP ZVM ↔ CSP/enterprise ZVM
- CSP VRA ↔ CSP/enterprise VRA (replication traffic)

Figure 4-10 shows the complete installation where the CSP is providing both DRaaS and ICDR within the same core ZVR environment. The CSP has two data centers with available resources. Each site has a Zerto Virtual Manager (ZVM) installed and VRAs for every ESX/ESXi host and the ZVMs are connected to the ZCM.

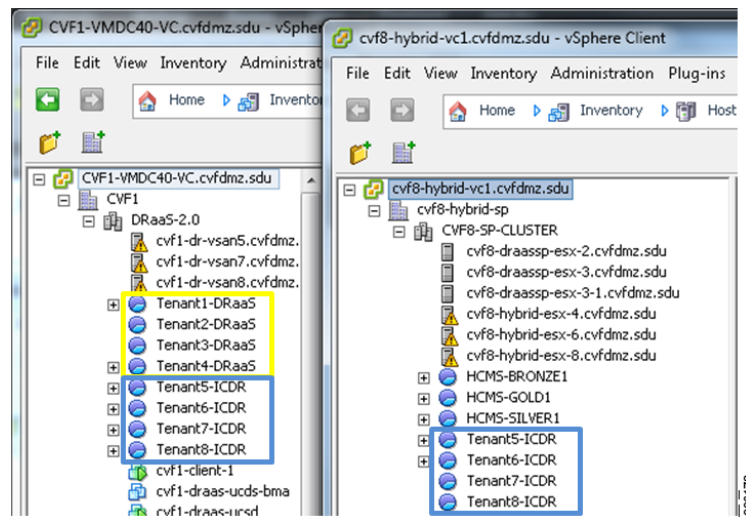
In the lab topology, each enterprise-to-CSP and CSP-to-CSP OTV link carried three VLANs, one for each tier of the three tier applications (e.g., web servers, application servers, database servers) that resided in each tenant. For the enterprise-to-CSP OTV link, the Zerto control and replication traffic was sent across the first VLAN, which was for the web servers. For the CSP-to-CSP link, the Zerto control and replication traffic was sent across a native IP link between sites that was outside of the customer OTV links. The OTV links between the CSP sites carried customer three-tier traffic only.

Figure 4-10 DRaaS 2.0 Test Topology—Diagram (DRAFT ILLO)



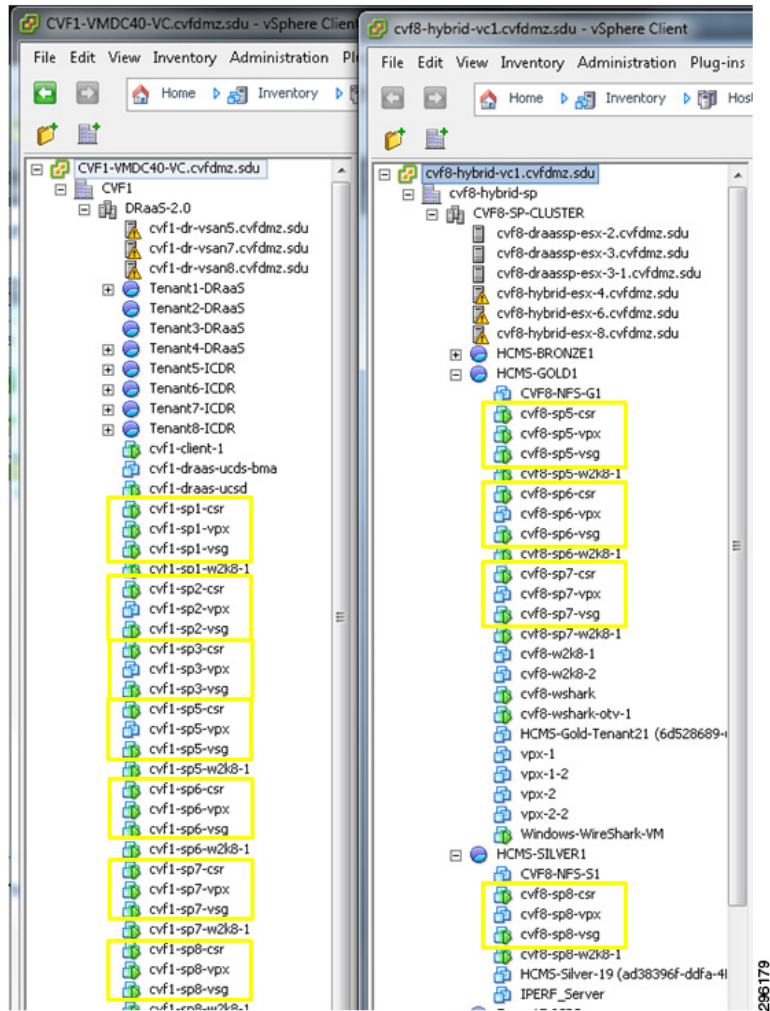
In the lab topology, each customer was assigned a unique resource pool in the CSP sites to help organize the customers' VMs under the data center. The DRaaS customers only require a single resource pool in the main provider site (e.g. CVF1), since they will only be using that site, while the ICDR customers require a resource pool in each site, since the protected VMs hosted in the primary site (e.g. CVF1) will be recovered to the second provider site (e.g. CVF8).

Figure 4-11 DRaaS 2.0 Test Topology - CSP Resource Pools



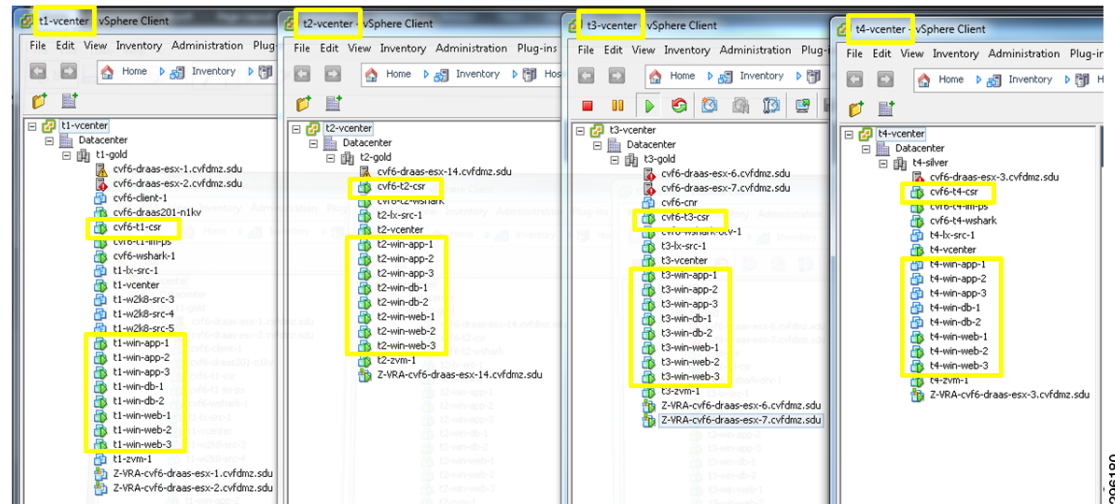
At the primary CSP site (left half of [Figure 4-11](#)), a CSR 1000V, Virtual Service Gateway (VSG), and Citrix VPX were deployed for each customer. At the secondary CSP site (right half of [Figure 4-11](#)), these components were only deployed for the ICDR customers.

Figure 4-12 DRaaS 2.0 Test Topology—CSP Infrastructure



Each customer site included one or two ESXi hosts, a CSR 1000V, and eight servers for a three-tier application. The three-tier application consisted of three web servers, three application servers, and two database servers. These were the VMs that were protected by Zerto into the primary CSP site.

Figure 4-13 DRaaS 2.0 Test Topology—DRaaS Customer Sites



The lab topology was built using Zerto Virtual Replication release 3.1 Update 2 (Build 031027034).



Note

Each Zerto Virtual Replication release includes a number of guides, pre-installation checklists, and tools to help with the planning, installation, and initial configuration, such as:

- Design considerations for DRaaS/ICDR.
- Resource requirements for the various components (e.g., ZVM, ZCM, ZCC, VRA).
- Network considerations (e.g., types of networks, ports used, and WAN sizing tool).
- Recommended best practices.
- Install/configuration/administrative guides for ZVM, ZCM, ZCC, and VRA.

The information in this section includes a subset of that content with some additional discussions pertaining to the specific architecture used in the lab topology. For general installation, configuration, and administration information, the Zerto documentation should be referenced.

Implementing Zerto Virtual Replication at the Cloud Service Provider

A CSP can offer both DRaaS and ICDR with the same Zerto components. The initial configuration of the ZVR involves the following tasks at the cloud sites:

- Licensing the use of Zerto Virtual Replication.
- Install Zerto Virtual Manager at each site.
- Install Zerto Virtual Replication Appliances (VRA) on each ESXi host.
- Set up vCloud Director, if it is being used.
- Set up static routes, if needed to enable connectivity between sites or components.
- Install Zerto Cloud Manager and configure the following:
 - Cloud sites providing DR capabilities.

- Organizations using the cloud DR services, either DRaaS or ICDR. For DRaaS organizations, deploy Zerto Cloud Connector.
- Service profiles, which are templates for protection.



Note A detailed review of the Zerto Virtual Replication installation is covered in Appendix D of the Cisco DRaaS 2.0 Implementation Guide, available at the following URL: <http://www.cisco.com/go/draas>.

Implementing Zerto Virtual Replication at the Customer Site

Once the core service environment is built, DRaaS and ICDR customer organizations can set up their sites. The customer setup involves installing only a ZVM and VRAs and then pairing to the CSP. The Zerto components used and the on-boarding processes are the same for either DRaaS or ICDR, therefore reducing the administrative overhead for the CSP.

DRaaS organizations can manage their disaster recovery via the Zerto UI, either via vSphere Client console or the Zerto standalone web UI. ICDR organizations can use the Zerto Self Service Portal (ZSSP), a stand-alone, limited access customer portal with limited functionality enabled by the CSP. In both cases the CSP can restrict the operations available to the organization, such as whether the organization can initiate a failover or test of protected virtual machines, by setting permissions for the organization in ZCM.

Steps Required for Zerto Virtual Replication (DRaaS Customers)

The initial configuration of the ZVR for DRaaS customers requires execution of some of the same tasks already executed at the CSP sites. The following tasks must be executed at the customer sites:

- Install Zerto Virtual Manager at each site.
- Licensing the use of Zerto Virtual Replication by pointing to CSP ZVM (via ZCC).
- Install Zerto Virtual Replication Appliances (VRA) on each ESXi host.
- If required, modify the MTU and disable TSO/LRO on the VRA.
- Create Virtual Protection Groups (VPGs) to establish protection via local ZVM UI.

Steps Required for Zerto Virtual Replication (ICDR Customers)

Since the CSP has already installed and configured all of the Zerto components required for In-Cloud Disaster Recovery, the ICDR customers only have to execute the following task:

- Create Virtual Protection Groups (VPGs) to establish protection via the Zerto Self Service Portal (ZSSP).

Creating a Virtual Protection Group (VPG)

Virtual machines are protected in virtual protection groups (VPG). A VPG is a group of VMs that are grouped together for replication purposes. For example, the VMs that comprise an application like Microsoft Exchange, where one VMs is used for the software, one for the database and a third for the web server, require that all three VMs are replicated to maintain data integrity.

Once a VM is protected, all changes made on the machine are replicated in the remote site. The initial synchronization between the protected site and remote site takes time, depending on the size of the VM, but after this only the writes to disk from the VM in the protected site are sent to the remote site. These writes are stored by the VRA in the remote site in a journal for a specified period, after which, the old writes are promoted to the mirror virtual disk managed by the VRA.

The CSP should determine the RPO and RTO goals from the customer. While ZVR will meet even the most aggressive requirements, the CSP should be aware that the customer needs to specify which one is more important, in relative terms.

The number of VPGs directly impacts the RTO due to a feature in ZVR that protects the vCenter from being overwhelmed in case of a complete site failover. If the customer needs to fail over several VPGs at once, ZVR considers it a bulk operation and reduces the number of simultaneous volumes per VPG created. While this protects the vCenter from being overloaded, it adds to the recovery time. It is important to note that every environment is unique and there are tools available from Zerto and Cisco that will assist in the design planning, but as a general guideline, there are expected characteristics for failover times.

If the customer requirement has an RPO priority:

- VPGs can be created to reflect the application affinity groupings.
- Larger VPGs can be created for administrative ease of use.

If the customer requirement has an RTO priority:

- Fewer VPGs will recover relatively quicker than more VPGs in the case of a simultaneous failover of multiple VPGs.
- There are other considerations to keep in mind when designing VPGs, such as application affinity grouping write-order fidelity.

**Note**

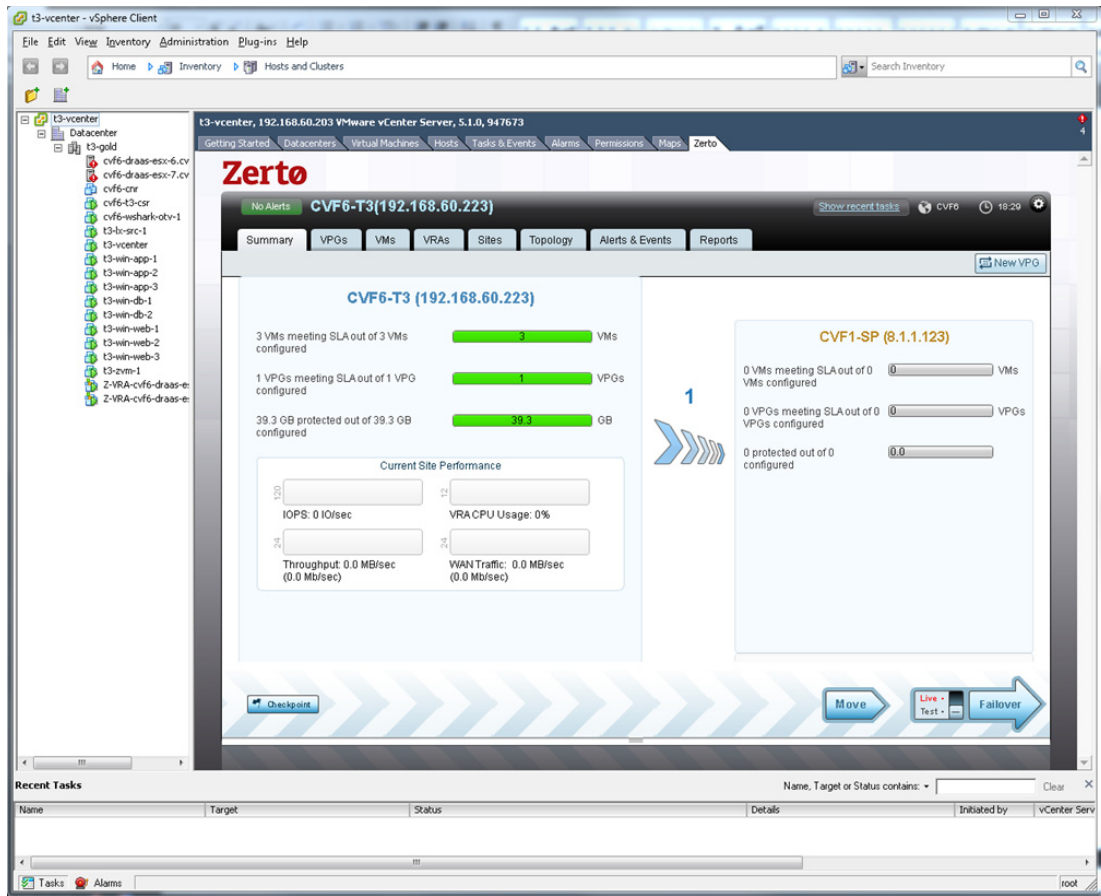
VPGs can have dozens of VMs in them, with no actual limit on the number of VMs per VPG, but other factors, such as application affinity grouping requirements, usually keep the number of VMs in a VPG to fewer than 50.

Configuring Virtual Protection Groups

The VPG must include at least one VM. After creating a VPG, you can add more VMs later. The protected VMs can be defined under a single ESX/ESXi host or under multiple ESX/ESXi hosts. The recovery VMs can also be on a single ESX/ESXi host or multiple ESX/ESXi hosts. The VMs are also recovered with the same configuration as the protected machines. For example, if a VM in the protected site is configured so that space is allocated on demand (thin provisioning) and this machine is protected in a VPG, then during recovery the machine is defined in the recovery site as thin provisioned.

For DRaaS, a new VPG can be created from the ZVM UI on the enterprise or CSP sites. For ICDR, a new VPG can be created from the ZSSP UI or ZVM UI in the CSP site. From the ZVM UI, the **New VPG** button is located in the upper right corner of most tabs in the UI.

Figure 4-14 New VPG Button in ZVM UI

**Note**

A detailed review of the procedure for implementing Zerto Virtual Replication at the customer site—including the Virtual Protection Groups—is covered in Appendix D of the Cisco DRaaS 2.0 Implementation Guide, available at the following URL: <http://www.cisco.com/go/draas>.

Deploying Application VMs on the Virtual SAN Datastore

In the CSP site, with VMware Virtual SAN providing the storage, all of the VMs needed for the Zerto application were deployed on the Virtual SAN cluster. For the sake of summary, here are the specifics of the Virtual SAN configuration once again:

- Three ESXi hosts running version 5.5
- Virtual SAN datastore made of twelve 1 TB spindles and three 400 GB SSDs
- A total datastore capacity of 12.3 TB (raw).
- Virtual SAN default policies set, including *StripeWidth* of 1 and *FailuresToTolerate* of 1.
- VMware Distributed Resource Scheduler (DRS) functionality enabled and configured for Automatic.
- VMware High Availability (HA) feature disabled.

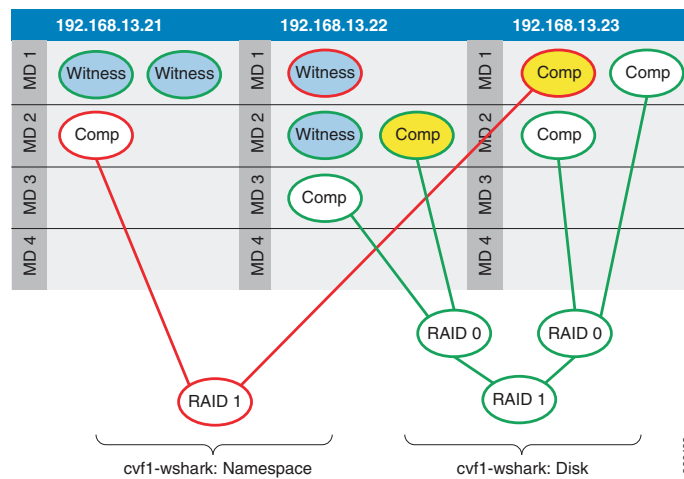
**Note**

VMware HA is certainly an option for Virtual SAN deployments. It was not used in this project due to the inherent resiliency that Virtual SAN provides and the fact that the project environment was limited to three nodes. In a four-node Virtual SAN cluster, HA would be a suitable choice.

Virtual SAN Component Distribution

In Chapter 2 we looked at how Virtual SAN created distributed RAID trees for the VM storage objects that it manages. Figure 4-15 provided a hypothetical look at how those objects and their subcomponents might be distributed on a 5-node Virtual SAN cluster. Figure 4-12 below looks at one of the VMs deployed on the 3-node Virtual SAN cluster used at the CSP site in this project. Because this VM was powered down at the time of this test, there is no SWAP file. Nor had any snapshots been taken of this VM. Therefore, the only storage objects belonging to this VM were the Namespace and the VMDK Disk.

Figure 4-15 Virtual SAN Distributed RAID Tree, *FTT=1* and *StripeWidth=1*



With *StripeWidth=1*, there is no striping across spindles; Each duplicate of the Virtual SAN object is written to a single MD. In the illustration above, then, each object has a single RAID-1 group, consisting of two copies of the object data, the Comp(onent) in the illustration. One of those data copies is active, depicted in yellow in the illustration. Each object has a single witness in this scenario, to arbitrate potential active-active conflicts.

If this VM were powered on, there would be an additional RAID-1 object for swap. If snapshots had been taken of this VM, there would be additional RAID-1 objects for each of the delta disks, one per snapshot.

Using the Ruby vSphere Console (RVC) Virtual SAN commands, we can see the structure of the RAID tree and the distribution of these components across hosts and MDs. The `vsan.vm_object_info` command was used to explore the structure of the storage objects belonging to a VM called “NewVM” which was powered off with no snapshots.

```
/localhost/VSAN-DC/computers/VSAN Cluster> vsan.vm_object_info ../../vms/NewVM/
2014-04-23 14:49:46 +0000: Fetching VSAN disk info from 192.168.13.23 (may take a
moment) ...
2014-04-23 14:49:46 +0000: Fetching VSAN disk info from 192.168.13.21 (may take a
moment) ...
2014-04-23 14:49:46 +0000: Fetching VSAN disk info from 192.168.13.22 (may take a
moment) ...
2014-04-23 14:49:47 +0000: Done fetching VSAN disk infos
```

```

VM NewVM:
  Namespace directory
    DOM Object: b3d15753-4cc3-db86-45ad-0025b5c262cc (owner: 192.168.13.21, policy:
hostFailuresToTolerate = 1)
    Witness: b4d15753-6e46-06d2-568f-0025b5c262cc (state: ACTIVE (5), host:
192.168.13.22, md: naa.600605b006f7dae01ab65f7317f942d1, ssd:
naa.600605b006f7dae01abde7e20e633fb2)
      RAID_1
        Component: b4d15753-52a2-05d2-0475-0025b5c262cc (state: ACTIVE (5), host:
192.168.13.21, md: naa.600605b006f7fb701ab654db583860a1, ssd:
naa.600605b006f7fb701abde07e0de6c455)
        Component: b4d15753-3a65-04d2-a98d-0025b5c262cc (state: ACTIVE (5), host:
192.168.13.23, md: naa.600605b006f7f9701ab603b21e8eff9d, ssd:
naa.600605b006f7f9701abe00d3102940d7)
    Disk backing: [vsanDatastore] b3d15753-4cc3-db86-45ad-0025b5c262cc/NewVM.vmdk
    DOM Object: b9d15753-8a7a-33af-324a-0025b5c262cc (owner: 192.168.13.23, policy:
hostFailuresToTolerate = 1, proportionalCapacity = 100)
    Witness: b9d15753-20da-4dbc-d7c0-0025b5c262cc (state: ACTIVE (5), host:
192.168.13.21, md: naa.600605b006f7fb701ab654db583860a1, ssd:
naa.600605b006f7fb701abde07e0de6c455)
      RAID_1
        Component: b9d15753-b4ad-4cbc-70be-0025b5c262cc (state: ACTIVE (5), host:
192.168.13.23, md: naa.600605b006f7f9701ab603db21047437, ssd:
naa.600605b006f7f9701abe00d3102940d7)
        Component: b9d15753-04d4-4abc-3237-0025b5c262cc (state: ACTIVE (5), host:
192.168.13.22, md: naa.600605b006f7dae01ab65f7317f942d1, ssd:
naa.600605b006f7dae01abde7e20e633fb2)

```

The Namespace and Disk objects have been highlighted (boldface) for clarity. Each of these objects has a RAID tree consisting of RAID-1 duplication of data as components. The output from this RVC command provides information on which MDs the components are written to, which host “owns” the storage object, where the witnesses exist, what the relevant policies are (notice **hostFailuresToTolerate** values in the output, and the SSDs that are front-ending the I/O for each disk for a particular component.

**Note**

One piece of information this command does *not* reveal is which of the object’s duplicates are actually active and used by the VM. For this information, use the **vsan.disk_object_info disk_uuid** command, which is referenced below in the Virtual SAN Troubleshooting section. This command will return all VMs and objects, using asterisks (**) to signify the active components.

Another way to view this output is to place it in a table showing the components, the RAID sets, and the hosts on which they reside. [Table 4-7](#) gives such a view for the VMDK Disk object.

Table 4-7 VM Storage Object View in Table, FTT=1 and StripeWidth=1

Type	Component State	Host	SSD Disk Name	MD Disk Name
Witness	Active	192.168.13.21	naa.*6c455	naa.*860a1
RAID 1				
Component	Active	192.168.13.23	naa.*940d7	naa.*47437
Component	Active	192.168.13.22	naa.*33fb2	naa.*942d1

Using [Table 4-7](#), we can evaluate the witness component placement in light of the witness deployment rules covered in Chapter 2. When the witness calculation—based on those three rules—is performed in this scenario, the witness logic comes into play as below:

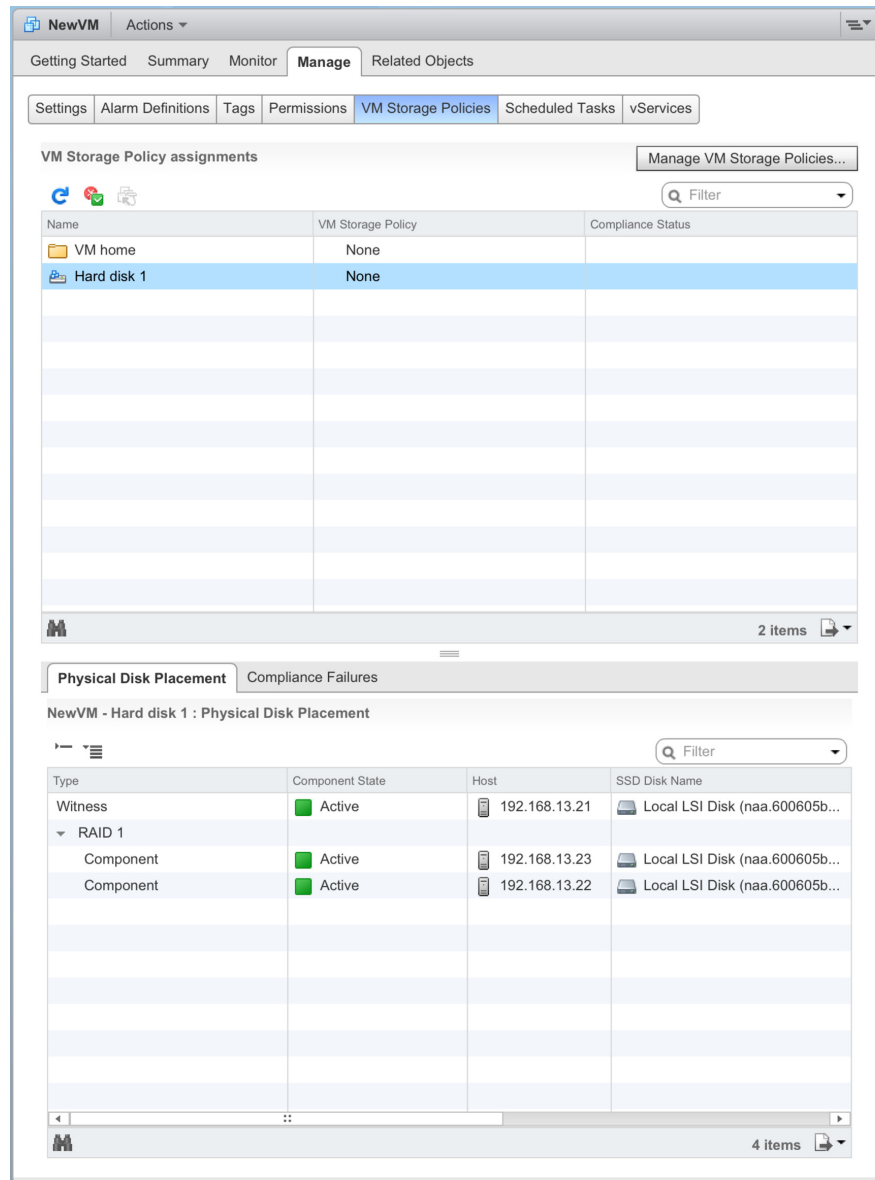
Primary Witnesses—The data components are spread over only 2 nodes (which is not greater than $2*FTT+1$), so we need to deploy a primary witness on the excluded node to achieve $2*FTT+1$.

Secondary Witnesses—With the addition of the Primary Witness, each node now has one component and equal voting power, so no Secondary Witnesses are needed.

Tiebreaker Witness—With the addition of the Primary Witness, there already an odd number of total components, so no Tiebreaker Witness is needed.

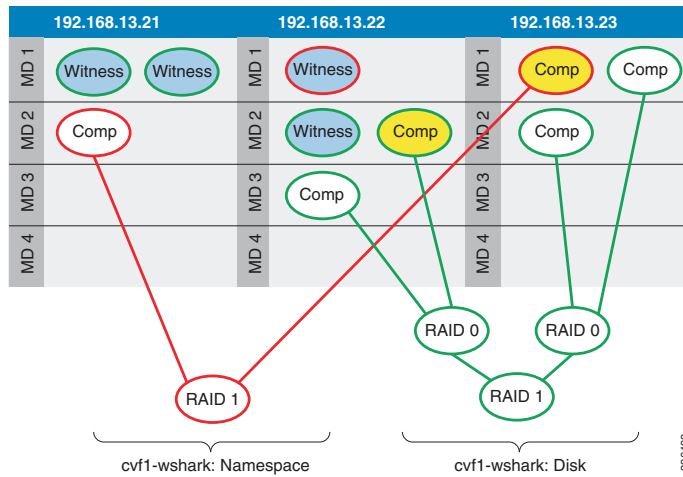
Through the vSphere web client, we can see the RAID tree by exploring the VM, and drilling down into **Manage > VM Storage Policies**, as in Figure 4-16. Here, for the VMDK Disk object, we see the RAID tree with the data object duplicated across two hosts (written to a single disk in each).

Figure 4-16 vSphere Web Client VM Storage Object View



For the sake of further explanation and demonstration, the *StripeWidth* policy was set to “2” for this VM. The object scheme shown in Figure 4-17 is the resultant layout for the new RAID tree.

Figure 4-17 Virtual SAN Distributed RAID Tree, *FTT=1* and *StripeWidth=2*

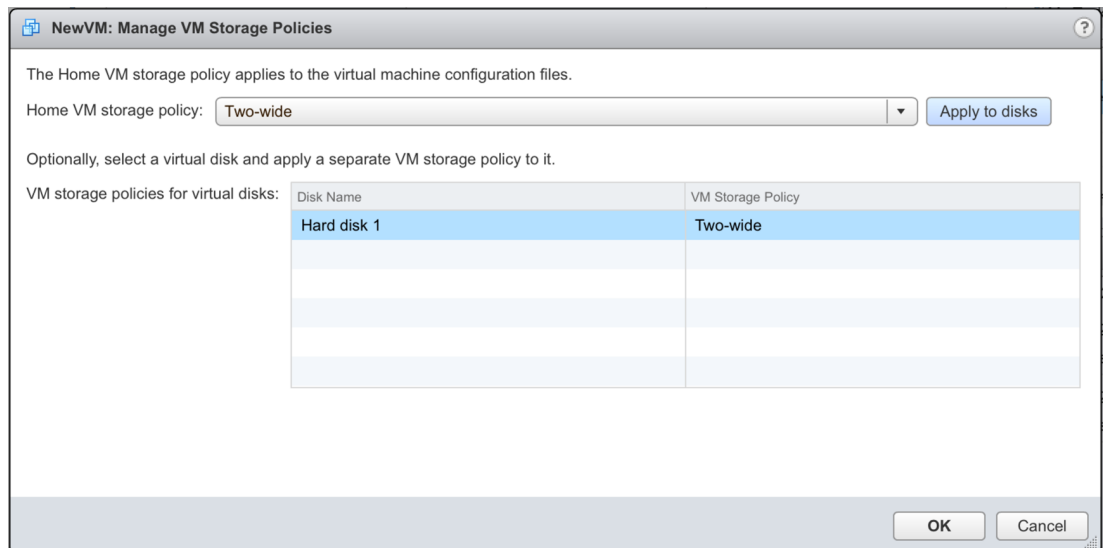


There are several key changes to take notice of, including:

- The Namespace RAID tree has not changed in character. It is still being handled as if *StripeWidth=1*. As mentioned in Chapter 2 above, the namespace and delta-disk objects will retain this striping policy regardless of what the administrator configures. This is because these objects are not performance limited and do not really benefit from having data striped across multiple spindles.
- The Disk (VMDK) RAID tree now includes two RAID-0 “branches” to handle the component striping across the spindles. Each component—the duplicate of the VMDK disk data—that had previously been written to a single MD is now written across two MDs.
- The VMDK Disk data is striped across MDs belonging to the same host. It is possible, with Virtual SAN, for striping to happen between MDs on different hosts as well.
- There are now three witnesses for the VMDK Disk object, due to the striping and MD distribution.

For our “NewVM,” we created a new storage policy with *FailuresToTolerate=1* and *StripeWidth=2* and applied it as shown in Figure 4-18.

Figure 4-18 Applying a *StripeWidth=2* Policy to an Existing VM



The RVC command output following this change is below, for comparison to the output above where the striping was set to 1. Notice that the RAID tree hasn't changed shape for the Namespace object, for the reasons explained above. Notice also that the VMDK Disk object now has the RAID-1 group split into two RAID-0 branches, with the components striped across multiple spindles (across multiple hosts in one case).

```
/localhost/VSAN-DC/computers/VSAN Cluster> vsan.vm_object_info ../../vms/NewVM/
VM NewVM:
  Namespace directory
    DOM Object: b3d15753-4cc3-db86-45ad-0025b5c262cc (owner: 192.168.13.21, policy:
hostFailuresToTolerate = 1, stripeWidth = 1, spbmProfileId =
84fb64de-2308-4214-b4d5-6024e6f3b523, proportionalCapacity = [0, 100],
spbmProfileGenerationNumber = 0)
    Witness: 80d95753-a67e-6264-5291-0025b5c262cc (state: ACTIVE (5), host:
192.168.13.22, md: naa.600605b006f7dae01ab65f7317f942d1, ssd:
naa.600605b006f7dae01abde7e20e633fb2)
      RAID_1
        Component: b4d15753-52a2-05d2-0475-0025b5c262cc (state: ACTIVE (5), host:
192.168.13.21, md: naa.600605b006f7fb701ab654db583860a1, ssd:
naa.600605b006f7fb701abde07e0de6c455)
        Component: b4d15753-3a65-04d2-a98d-0025b5c262cc (state: ACTIVE (5), host:
192.168.13.23, md: naa.600605b006f7f9701ab603b21e8eff9d, ssd:
naa.600605b006f7f9701abe00d3102940d7)
      Disk backing: [vsanDatastore] b3d15753-4cc3-db86-45ad-0025b5c262cc/NewVM.vmdk
      DOM Object: b9d15753-8a7a-33af-324a-0025b5c262cc (owner: 192.168.13.23, policy:
spbmProfileGenerationNumber = 0, stripeWidth = 2, spbmProfileId =
84fb64de-2308-4214-b4d5-6024e6f3b523, hostFailuresToTolerate = 1)
      Witness: e1d95753-424a-6464-6924-0025b5c2622c (state: ACTIVE (5), host:
192.168.13.22, md: naa.600605b006f7dae01ab65f4b159bff8c, ssd:
naa.600605b006f7dae01abde7e20e633fb2)
      Witness: e1d95753-f278-5e64-b8c8-0025b5c2622c (state: ACTIVE (5), host:
192.168.13.23, md: naa.600605b006f7f9701ab603db21047437, ssd:
naa.600605b006f7f9701abe00d3102940d7)
      Witness: e1d95753-5c28-6264-2cb6-0025b5c2622c (state: ACTIVE (5), host:
192.168.13.22, md: naa.600605b006f7dae01ab65f7317f942d1, ssd:
naa.600605b006f7dae01abde7e20e633fb2)
        RAID_1
          RAID_0
            Component: e1d95753-26e0-0448-6233-0025b5c2622c (state: ACTIVE (5), host:
192.168.13.21, md: naa.600605b006f7fb701ab654db583860a1, ssd:
naa.600605b006f7fb701abde07e0de6c455,
dataToSync: 0.00 GB)
            Component: e1d95753-6028-0248-bb0b-0025b5c2622c (state: ACTIVE (5), host:
192.168.13.21, md: naa.600605b006f7fb701ab6548453137d63, ssd:
naa.600605b006f7fb701abde07e0de6c455,
dataToSync: 0.00 GB)
          RAID_0
            Component: e1d95753-622e-ff47-dca8-0025b5c2622c (state: ACTIVE (5), host:
192.168.13.23, md: naa.600605b006f7f9701ab603b21e8eff9d, ssd:
naa.600605b006f7f9701abe00d3102940d7,
dataToSync: 0.00 GB)
            Component: e1d95753-be88-f947-1472-0025b5c2622c (state: ACTIVE (5), host:
192.168.13.22, md: naa.600605b006f7dae01ab65f4b159bff8c, ssd:
naa.600605b006f7dae01abde7e20e633fb2,
dataToSync: 0.00 GB)
```

Again, populating a table with the output data for the VMDK Disk object can provide greater clarity, as in [Table 4-8](#).

Table 4-8 VM Storage Object View in Table, FTT=1, and StripeWidth=2

Type	Component State	Host	SSD Disk Name	MD Disk Name	
Witness	Active	192.168.13.22	naa.*33fb2	naa.*bff8c	
Witness	Active	192.168.13.23	naa.*940d7	naa.*47437	
Witness	Active	192.168.13.22	naa.*33fb2	naa.*942d1	
RAID 1					
RAID 0					
	Component	Active	192.168.13.21	naa.*6c455	naa.*860a1
	Component	Active	192.168.13.21	naa.*6c455	naa.*37d63
RAID 0					
	Component	Active	192.168.13.23	naa.*940d7	naa.*eff9d
	Component	Active	192.168.13.22	naa.*33fb2	naa.*bff8c

Using [Table 4-8](#), we can again evaluate the witness component placement in light of the witness deployment rules covered in Chapter 2. When the witness calculation—based on those three rules—is performed in this scenario, the witness logic comes into play as below:

Primary Witnesses—The data components are spread over all 3 nodes (which is equal to $2*FTT+1$), so no primary witnesses are needed.

Secondary Witnesses—Since one node (.21) has two votes and the other two nodes only have one vote each, we need to add a Secondary Witness on each of the other two nodes (.22 and .23).

Tiebreaker Witness—With the addition of the two Secondary Witnesses, the total component is 6 (4 data + 2 witnesses). There is an even number of total components, so a Tiebreaker Witness is needed and added on the .22 host.

The vSphere web client reflects this new storage policy, visualizing the new RAID tree for the VMDK Disk object, as shown in [Figure 4-19](#).

Figure 4-19 vSphere Web Client VM Storage Object View StripeWidth=2

The screenshot displays the vSphere Web Client interface for a new VM. The top navigation bar includes 'NewVM', 'Actions', and tabs for 'Getting Started', 'Summary', 'Monitor', 'Manage', and 'Related Objects'. The 'Manage' tab is active, showing sub-tabs for 'Settings', 'Alarm Definitions', 'Tags', 'Permissions', 'VM Storage Policies', 'Scheduled Tasks', and 'vServices'. The 'VM Storage Policies' sub-tab is selected, showing a table of VM Storage Policy assignments.

Name	VM Storage Policy	Compliance Status
VM home	Two-wide	✓ Compliant
Hard disk 1	Two-wide	✓ Compliant

Below the table, the 'Physical Disk Placement' section is visible, showing details for 'NewVM - Hard disk 1 : Physical Disk Placement'. This section includes a table with columns for Type, Component State, Host, and SSD Disk Name.

Type	Component State	Host	SSD Disk Name
RAID 1			
RAID 0			
Component	Active	192.168.13.22	Local LSI Disk (naa.600605b...)
Component	Active	192.168.13.23	Local LSI Disk (naa.600605b...)
RAID 0			
Component	Active	192.168.13.21	Local LSI Disk (naa.600605b...)
Component	Active	192.168.13.21	Local LSI Disk (naa.600605b...)
Witness	Active	192.168.13.23	Local LSI Disk (naa.600605b...)
Witness	Active	192.168.13.22	Local LSI Disk (naa.600605b...)
Witness	Active	192.168.13.22	Local LSI Disk (naa.600605b...)

295673

Zerto Application Suite on Virtual SAN at CSP Site

As discussed in Chapter 2, for a complete Zerto deployment, several VMs will be built and configured at the CSP data center site, including the Zerto Virtual Manager (ZVM), Zerto Cloud Manager (ZCM), Zerto Cloud Connector (ZCC), and Zerto Virtual Replication Appliance (VRA). The ZCC is deployed on a per-tenant basis, which will factor into Virtual SAN scalability. The VRA is deployed on a per-host basis, so there will be one VRA for each ESXi host in the Virtual SAN cluster.

We also mentioned Virtual SAN scalability in Chapter 2, specifically the soft limits on the number of storage objects per ESXi host. At the time of this writing, with the initial release of Virtual SAN running, there is a soft limit of 3000 storage objects per ESXi host. Table 4-9 summarizes the Zerto application

pieces deployed for this project, and the number of storage object components each subsists of. An explanation of the component counts, as well as further information on Zerto scalability within Virtual SAN, is provided below.

Table 4-9 Zerto CSP Deployment for Test Project

Zerto Component	Components	Configuration Notes
Zerto Cloud Controller	12 per tenant	Two virtual disk attached; One ZCC per tenant
Zerto Cloud Manager	9	One virtual disk attached
Zerto Virtual Manager	9	One virtual disk attached
Zerto Virtual Replication Appliance	9 per host	Variable disks attached; One VRA per Virtual SAN ESXi node



Note

The reader should not use this summary as predictive of every production deployment, as the number of servers and the configuration of those servers may change from one deployment to the next. Note that there are no snapshots for any of these VMs included in component count.

Component count directly influenced by the storage policy settings *FailuresToTolerate* and *StripeWidth*. The table above represents a deployment in which both were left as default (1). Here is how these component counts are accounted for.

Zerto Cloud Connector

The Zerto Cloud Connector was a VM configured with two virtual disk. Four objects, therefore, subsisted the ZCC: (1) Namespace; (2) VMDK Disks; (1) Swap. Since the ZCC has no need of VMware snapshots, there were no delta disk objects. With *FailuresToTolerate*=1, there was one RAID-1 set per object with two components, each on a different host. With *StripeWidth*=1, each component was written to a single spindle, so there were no RAID-0 sets. To satisfy the witness creation rules (described above in Chapter 2), one witness was created per object; Thus, the total number of components per object was three. [Table 4-10](#) provides a theoretical look at the component map for the ZCC.



Note

Remember, ZCC is deployed on a per-tenant basis, which should factor into any scalability calculations.

Table 4-10 Hypothetical VM Storage Object View For ZCC, FTT=1 and StripeWidth=1

Object	Type	Component State	Host
Namespace	Witness	Active	192.168.13.21
	RAID 1		
	Component	Active	192.168.13.22
Disk 1	Component	Active	192.168.13.23
	Witness	Active	192.168.13.22
	RAID 1		
Disk 2	Component	Active	192.168.13.21
	Component	Active	192.168.13.23
	Witness	Active	192.168.13.21

Table 4-10 Hypothetical VM Storage Object View For ZCC, FTT=1 and StripeWidth=1 (continued)

Object	Type	Component State	Host
	RAID 1		
	Component	Active	192.168.13.22
	Component	Active	192.168.13.23
Disk 3	Witness	Active	192.168.13.23
	RAID 1		
	Component	Active	192.168.13.22
	Component	Active	192.168.13.21
Swap	Witness	Active	192.168.13.22
	RAID 1		
	Component	Active	192.168.13.23
	Component	Active	192.168.13.21

Zerto Cloud Manager and Virtual Manager

Because both the ZCM and the ZVM were configured similarly, with a single virtual disk, the following explanation applies equally. Each of these servers had three objects: (1) Namespace; (1) VMDK Disks; (1) Swap. There were no VMware snapshots, so there were no delta disk objects. With *FailuresToTolerate*=1, there was one RAID-1 set per object with two components, each on a different host. With *StripeWidth*=1, each component was written to a single spindle, so there were no RAID-0 sets. To satisfy the witness creation rules (described above in Chapter 2), one witness was created per object; Thus, the total number of components per object was three. Table 4-11 provides a theoretical look at the component map for the ZCM and ZVM.

Table 4-11 Hypothetical VM Storage Object View For ZCM & ZVM, FTT=1, and StripeWidth=1

Object	Type	Component State	Host
Namespace	Witness	Active	192.168.13.21
	RAID 1		
	Component	Active	192.168.13.22
	Component	Active	192.168.13.23
Disk	Witness	Active	192.168.13.22
	RAID 1		
	Component	Active	192.168.13.21
	Component	Active	192.168.13.23
Swap	Witness	Active	192.168.13.22
	RAID 1		
	Component	Active	192.168.13.23
	Component	Active	192.168.13.21

Zerto Virtual Replication Appliance

The Zerto Virtual Replication Appliance was a VM configured with one base virtual disk. Three objects, therefore, subsisted the VRA upon its creation: (1) Namespace; (1) VMDK Disks; (1) Swap. There were no VMware snapshots, so there were no delta disk objects. With *FailuresToTolerate*=1, there was one RAID-1 set per object with two components, each on a different host. With *StripeWidth*=1, each component was written to a single spindle, so there were no RAID-0 sets. To satisfy the witness creation rules (described above in Chapter 2), one witness was created per object. Thus, the total number of components per object was three. Table 4-12 provides a theoretical look at the component map for the VRA.



Note

Remember, VRA is deployed in the Virtual SAN cluster on a per-host basis, which should factor into any scalability calculations.

Table 4-12 Hypothetical VM Storage Object View For VRA, FTT=1 and StripeWidth=1

Object	Type	Component State	Host
Namespace	Witness	Active	192.168.13.21
	RAID 1		
	Component	Active	192.168.13.22
Disk	Component	Active	192.168.13.23
	Witness	Active	192.168.13.22
	RAID 1		
Swap	Component	Active	192.168.13.21
	Component	Active	192.168.13.23
	Witness	Active	192.168.13.22
	RAID 1		
	Component	Active	192.168.13.23
	Component	Active	192.168.13.21

Zerto Component Scalability on Virtual SAN

As mentioned above, the scalability of a given Zerto DRaaS deployment will be directly related to the configuration of the Zerto environment and the underlying virtual infrastructure. The other factor governing scalability is servers that are being protected by the Zerto DRaaS service running at the CSP. Just as with the Zerto nodes themselves, the configuration of the hosts that are protected by Zerto will determine scalability within Virtual SAN.

This paper cannot predict absolute scalability and make definitive statements regarding the number of hosts that can be protected by Zerto backed by a given Virtual SAN deployment. Instead, we will strive to provide the guidance needed for the administrators to predict scalability in their own, unique deployments. We therefore, offer the following guidelines.

Protection Plan Creation

There are several factors involved that impact how many volumes will be attached to a VRA.

When a Zerto VPG is created, the target site VRA has ownership of the replicated VMDKs for the virtual machines included in the VPG and they are attached to the VRA as volumes.

As failover testing is performed, temporary volumes are added and then removed upon failover test completion. Additionally, the CSP can have multiple customers utilizing the same ESXi host and VRAs.

The VRA automatically adds volumes to accommodate the point-in-time journals of the VPGs. Depending on the size of the journals, the VRA will create additional volumes as needed. Zerto has a journal sizing tool available for planning the expected size of the journals.

Table 4-12 shows the impact of the journal sizes to the number of disks added to the VRA.

Table 4-13 Journal Size Relation to Number of Disks Added to VRA

Journal Size (GB)	Number of Disks
<16	1
16-48	2
48-100	3-4
100-200	4-5
200-500	5-8
500-1000	8-10

- **Guideline #1**—There will be one Namespace object per protected machine.
- **Guideline #2**—Regardless of the storage policy being used, the Namespace object will have *StripeWidth*=1 and *FailuresToTolerate*=1.
- **Guideline #3**—With *StripeWidth*=1, two components will be created for the Namespace object.
- **Guideline #4**—Following the rules for witness creation in Chapter 2, one witness will be created.
- **Guideline #5**—There will be one object created per virtual disk on the machine.
- **Guideline #6**—The number of mirror copies created of each Disk object will be determined by the *FailuresToTolerate* policy, using the formula [#Copies = FTT + 1].
- **Guideline #7**—The number of components created for each of the Disk object copies will be governed by the *StripeWidth* policy, using the formula [#Components = #Copies * #Stripes].
- **Guideline #8**—Follow the rules for witness creation to determine the number of witness components that will be created.
- **Guideline #9**—There will be no swap objects created until the machine is powered on, following the DR event.
- **Guideline #10**—None of the snapshots for the protected machine will be propagated to the Zerto replicate, so there will be no delta disk objects.
- **Summary**—On the CSP side, the machine that is protected will have, at this point, objects for its disks and namespace.

Recover Event

When a failover is invoked to recover customer machines in the CSP, ZVR creates and adds the virtual machines to the vCenter inventory, detaches the replicated volumes from the VRA ownership and attaches them to the appropriate virtual machines, then completes the failover process workflow. This may include changing IP addresses or running startup scripts.

- **Guideline #1**—All Disk and Namespace objects and components will remain the same as when the machine was initially protected.
- **Guideline #2**—A Swap object will be created for the recovered VM.

- **Guideline #3**—Regardless of the storage policy being used, the Namespace object will have *StripeWidth=1* and *FailuresToTolerate=1*.
- **Guideline #4**—With *StripeWidth=1*, two components will be created for the Namespace object.
- **Guideline #5**—Following the rules for witness creation in Chapter 2, one witness will be created.
- **Summary**—The protected machine will add to its existing count three components for the Swap object when it is powered on.

Failover and Reverse Protection Operations

Some of the core operations of ZVR include:

- Move
- Failover Testing
- Failover
- Reverse Protection

Depending on the operation, ZVR will execute different workflows.

When a customer performs a Move operation, they are transferring the workload from being hosted locally to being permanently hosted at the CSP site. The workflow powers off the source virtual machine, removes it from inventory and powers it on at the CSP site.

A failover test operation is completely non-disruptive to the production environment. It allows the administrator to determine boot up order and timing, script execution and generally perfecting the failover event in a safe environment to produce the lowest recovery time possible. During the testing operation, ZVR creates a disposable temporary disk from the source protected replicated VMDK and all changes are temporarily made to the disposable disk. Once testing is complete, the disposable disk is deleted. Depending on the Virtual SAN *StripeWidth* and *FailuresToTolerate* settings, three or more Virtual SAN components will be created with this disposable disk and destroyed with its deletion.

A failover operation fails over the virtual machine to the CSP due to an event at the source site. The source site may or may not be communicable with the CSP site, so the customer can access and manage the operations in the Zerto Self Service Portal, presented as a standalone ZSSP interface or embedded in the CSP's customer portal. In the failover operation, the VRA releases control of the replicated VMDKs and adds the virtual machine to the vCenter inventory and powers on the VM at the CSP site.

During the failover workflow, the Reverse Protection operation is configured. Reverse Protection streamlines the immediate need to replicate from the CSP back to the customer site as soon as the customer site and the CSP site resume communications. Upon initiation of the Reverse Protection operation, ZVR performs a delta sync of the data between the sites.

Zerto Storage Scalability on Virtual SAN

We must mention in brief how storage consumption on the CSP Virtual SAN datastore is scaled. This, too, is governed by the VM storage policies being employed on the Virtual SAN cluster, specifically the *FailuresToTolerate* policy. When FTT=1, for example, two copies of the data are created. For a crude estimation, determine storage consumption using the following formula: [StorageUsed = DiskSize * (FTT + 1)]. For example, given a protected machine having one 20GB disk and FTT=1, 40GB of storage will be used. If a protected machine has three 100 GB disks and FTT=2, 900GB of storage will be used.



Note

For a finer calculation, please use the Virtual SAN datastore calculator on Duncan Epping's blog: <http://www.yellow-bricks.com/2014/01/17/virtual-san-datastore-calculator/>

The storage consumed for non-Disk objects (including witnesses) is negligible, unless at very large scales. For example, a witness object consumes 2 MB of disk. This is not a big deal for a few VMs, but for hundreds of VMs and storage policies set to non-default values, it could add up.



Virtual SAN Command Line Commands

There are many CLI commands available to manipulate VMware Virtual SAN—both to configure it and to get status information about it. The two platforms on which these CLI commands can be used are RVC on the Vcenter server and ESXCLI on the ESXi host.

[Table A-1](#) summarizes those commands, providing info about what platform they are executed on as well as brief a brief list of what you can get from or do with them.



Note

Regarding RVC—To initiate an RVC session, you need to first SSH into the Vcenter host (assuming Linux-based) and enter the following command: `rvc root@localhost`.



Note

Regarding ESXCLI—You will first need to SSH into the ESXi host. If you are unable to connect via SSH, ensure you have enabled SSH on the ESXi host.

Table A-1 Useful Virtual SAN CLI Commands

Command	Platform	Information
<code>vsan.apply_license_to_cluster</code>	RVC	(Unknown. Suspected use is to allow user to apply a Virtual SAN license to the cluster.)
<code>vsan.check_limits</code>	RVC	Gives per-host info on how much of the stated Virtual SAN limits are being consumed, including: <ul style="list-style-type: none"> • # Assocs (20,000 per host limit) • # Sockets (10,000 per host limit) • # Components (3000 per host limit) • # Disk utilization per disk
<code>vsan.check_state</code>	RVC	Runs through three steps to verify the current health of the VMs living on the Virtual SAN datastore.
<code>vsan.cluster_change_autoclaim</code>	RVC	Toggles disk autoclaim on the specified Virtual SAN cluster.

Table A-1 Useful Virtual SAN CLI Commands (continued)

Command	Platform	Information
vsan.cluster_info	RVC	Gives info on each host in the Virtual SAN cluster, including: <ul style="list-style-type: none"> • Virtual SAN status • Node and cluster UUIDs • The host disks that are mapped as part of the cluster • The vmkernel NIC Virtual SAN networking information
vsan.cluster_set_default_policy	RVC	Allows setting the default FailuresToTolerate (FTT) policy
vsan.cmmnds_find	RVC	Gives detailed information on all the disks in the cluster. Output is tabular and includes the following info: <ul style="list-style-type: none"> • The Virtual SAN-assigned UUID of the disk • The cluster host that owns the disk • Health status • Detailed content of the disk, including capacity, IOPS info, whether an SSD, etc.
vsan.disable_vsan_on_cluster	RVC	Disables Virtual SAN completely on the cluster
vsan.disk_object_info	RVC	Uses the disk UUID from the vsan.cmmnds_find command (above) to examine the details of the VM components on a particular disk. Note that the components will only be on magnetic disks. The details given include: <ul style="list-style-type: none"> • The UUID and IP address of the host that the VM actually lives on • The "context" which is often the name of the parent VM that owns the component • The UUID of the RAID1 witness • The hosts that the RAID1 components live on and the info for the disk holding those components • State of the various elements (witness and components)

Table A-1 Useful Virtual SAN CLI Commands (continued)

Command	Platform	Information
vsan.disks_info	RVC	Gives information about the disks living on a particular Virtual SAN host, whether they belong to the Virtual SAN or not, including: <ul style="list-style-type: none"> • Disk type and ID • SSD or MD • Capacity • State ("inUse" by Virtual SAN, or other)
vsan.disks_stats	RVC	Gives detailed info on each disk in the Virtual SAN cluster. The output is formatted as a table with each disk group represented as a separate row. The info provided includes: <ul style="list-style-type: none"> • Disk ID (e.g., naa.500a0751038f25d9) • Hostname or IP of the host the disk group is on • Whether the disk is SSD or MD (magnetic disk) • The number of VM components living on the disk • Disk capacity?% of disk used and reserved • Current IO state • Current Device Latencies
vsan.enable_vsan_on_cluster	RVC	Enables Virtual SAN completely on the cluster
vsan.enter_maintenance_mode	RVC	Puts specified host into maintenance mode
vsan.fix_renamed_vms	RVC	(Unknown)
vsan.host_consume_disks	RVC	(Unknown. Suspected function is to trigger a certain host in the cluster to claim all eligible disks for use in Virtual SAN cluster.)
vsan.host_info	RVC	Gives the same information as the vsan.cluster_info command (above), but for a single Virtual SAN cluster host
vsan.host_wipe_vsan_disks	RVC	Completely wipes out the data on all disks on a single host – USE WITH CAUTION!
vsan.lldpnetmap	RVC	(unknown)
vsan.obj_status_report	RVC	Queries all VMs, objects, disks, and components in a cluster and provides data on healthy and unhealthy objects.

Table A-1 Useful Virtual SAN CLI Commands (continued)

Command	Platform	Information
vsan.object_info	RVC	For a particular object UUID, provides the Virtual SAN storage info, including: <ul style="list-style-type: none"> • The components and witnesses • The hosts they live on • The MD disk IDs they live on • The SSDs for the disk group they are on.
vsan.object_reconfigure	RVC	(Unknown)
vsan.observer	RVC	Initiates the Virtual SAN Observer process, so that real-time metrics about the Virtual SAN (IOPs, etc.) can be viewed in browser.
vsan.reapply_vsan_vmknics_config	RVC	(Unknown)
vsan.recover_spbm	RVC	(Unknown)
vsan.resync_dashboard	RVC	When sync'ing is taking place on the data store, this command can be used to monitor how much data and how many objects remain to be sync'ed.
vsan.vm_object_info	RVC	Provides the same info as the vsan.object_info command (above) except called using a VM name instead of object UUID.
vsan.vm_perf_stats	RVC	Queries twice, at a 20 second interval, and averages performance stats for IOPS, throughput, and latency for a given VM on the Virtual SAN cluster.
vsan.whatif_host_failures	RVC	Displays results of a "what if...?" scenario for a number of host failures (default 1 if FTT is at default of 1). Provides hypothetical resource availability change after failure for the following elements: <ul style="list-style-type: none"> • HDD capacity • Components • RC reservations
esxcli vsan datastore	ESXCLI	Allows user to configure a Virtual SAN datastore name.
esxcli vsan network	ESXCLI	Clear, list, remove, or restore Virtual SAN networking on the host.
esxcli vsan storage	ESXCLI	Add, list, or remove physical storage (MDs or SSDs) from the host. (Note, removing an SSD using this command will remove that SSD's disk group from the Virtual SAN cluster.)
esxcli vsan cluster	ESXCLI	Get the cluster info for the host and Virtual SAN, join the host to a cluster, leave a cluster, or restore the cluster configuration

Table A-1 *Useful Virtual SAN CLI Commands (continued)*

Command	Platform	Information
esxcli vsan maintenancemode	ESXCLI	(Unknown. Suspected use is to take the cluster host in/out of maintenance mode.)
esxcli vsan policy	ESXCLI	Manipulate the default storage policy values. (Unknown whether this applies to <i>FailuresToTolerate</i> , <i>StripeWidth</i> , or both.)
esxcli vsan trace	ESXCLI	(Unknown)





Technical References

Many of the following resources were referenced during the execution of this testing and production of this white paper. They are provided here for the reader's further edification.

Cisco Technologies

For the Cisco SDU Cloud Engineering DRaaS Design and Implementation Guides, including the recent DRaaS 2.0 system to which this white paper is complementary, visit the DRaaS landing page:

- <http://www.cisco.com/go/draas>

For the Cisco SDU Cloud Engineering VMDC Design and Implementation Guides, including the recent VSA 1.0 system, visit the VMDC landing page:

- <http://www.cisco.com/go/vmdc>

For all Cisco SDU Cloud Engineering Data Center architecture documents and white papers:

- <http://www.cisco.com/c/en/us/solutions/enterprise/design-zone-data-centers/index.html>

Cisco UCS landing page:

- <http://www.cisco.com/go/ucs>

Cisco SDU Cloud Engineering UCS Service Templates White Paper:

- http://www.cisco.com/c/en/us/td/docs/solutions/Hybrid_Cloud/DRaaS/Service_Template/Service_Templates.pdf

The following URL provides the complete spec sheet for the UCS C240 M3 small form factor server:

- http://www.cisco.com/en/US/prod/collateral/ps10265/ps10493/C240M3_SFF_SpecSheet.pdf

For more details on each of the available C-Series models, visit:

- <http://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-c-series-rack-servers/datasheet-listing.html>

For a side-by-side comparison between the different UCS C-Series models, visit:

- <http://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-c-series-rack-servers/models-comparison.html>

VMware Technologies

The majority of technical information about Virtual SAN was compiled, directly or indirectly, from resources available on the Virtual SAN resources website:

- <http://www.vmware.com/products/virtual-san/resources.html>

Several VMware technical engineers maintain some very useful Virtual SAN information on their blogs:

- Cormac Hogan: <http://cormachogan.com/vsan/>
- Duncan Epping: <http://www.yellow-bricks.com/virtual-san/>
- Rawlinson: <http://punchingclouds.com/tag/vsan/>

Another highly recommended resource for Virtual SAN administrators is the forthcoming *Essential Virtual SAN (VSAN): Administrator's Guide to VMware VSAN* by Cormac Hogan and Duncan Epping (anticipated publication by VMware Press on August 19, 2014).

Zerto Technologies

For more information about Zerto Virtual Replication:

- <http://www.zerto.com>