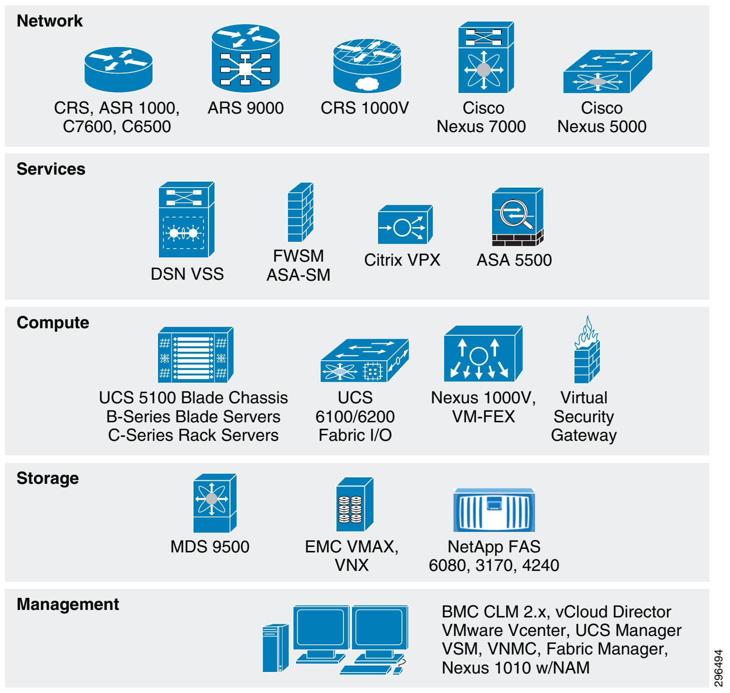
Architecture Configuration

The Cisco DRaaS 2.0 solution is layered on top of the Cisco VMDC VSA 1.0[1] 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.

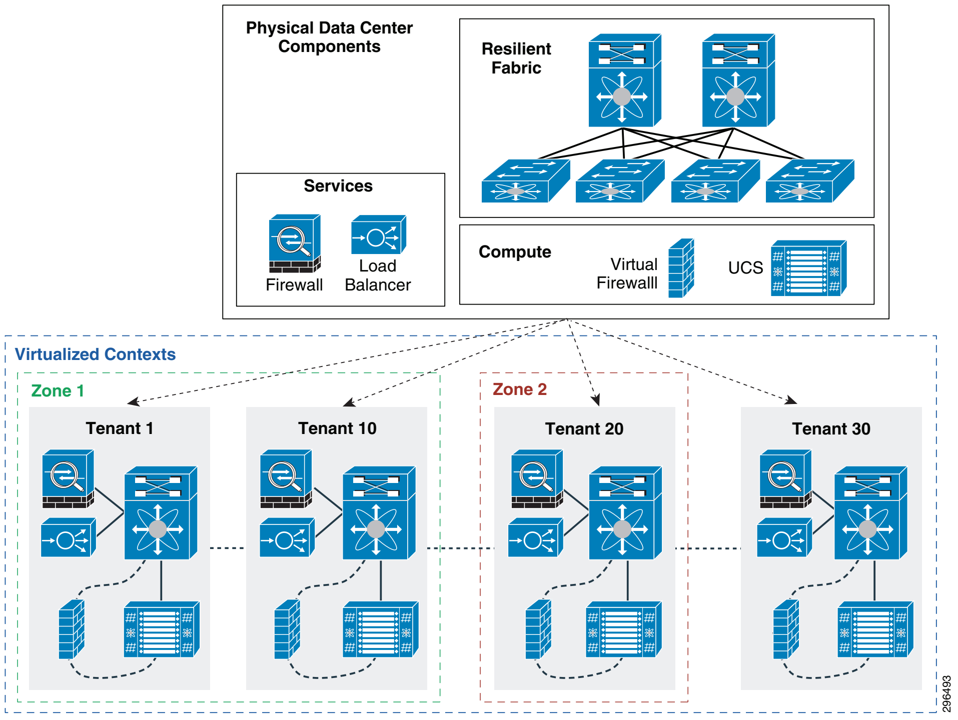
Figure 4-1 Functional Layers Within the VMDC Data Center



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*

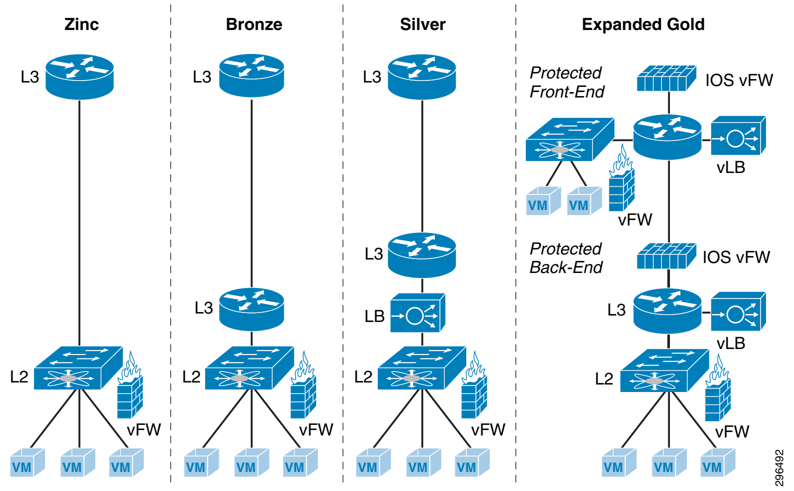
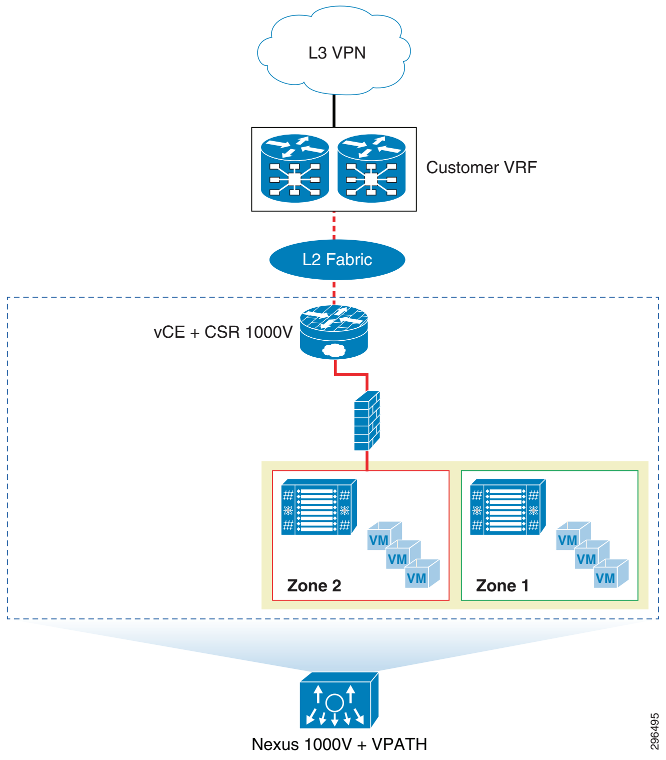


Figure 4-4 *VMDC VSA 1.0 Bronze Container Model*



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 Performamce 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

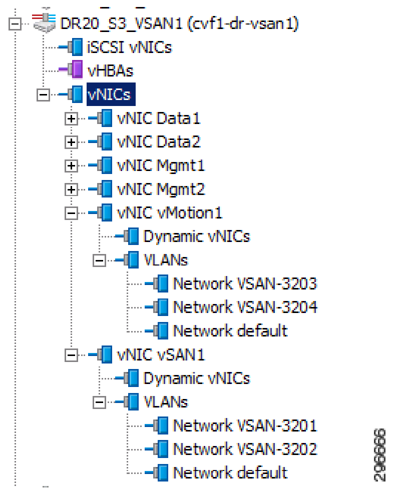
The three UCS C240-M3S hosts were deployed using UCS Manager with six vNICs. The six vNICs were configured to carry various VLANs, as shown in Table 4-4. Figure 4-5 shows the vNICs that were configured to carry Virtual SAN and vMotion traffic.

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.

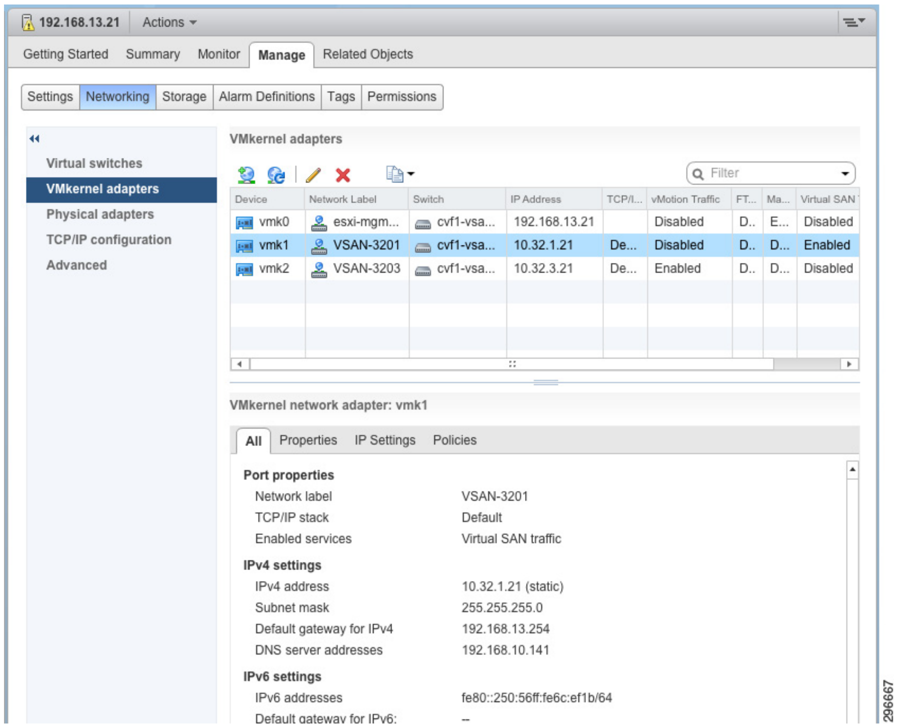
Figure 4-5 UCS Host vNIC VLAN Configuration



```
DR20_S3_VSAN1 (cvf1-dr-vsan1)
  iSCSI vNICs
  vHBAs
  vNICs
    vNIC Data1
    vNIC Data2
    vNIC Mgmt1
    vNIC Mgmt2
    vNIC vMotion1
      Dynamic vNICs
      VLANs
        Network VSAN-3203
        Network VSAN-3204
        Network default
    vNIC vSAN1
      Dynamic vNICs
      VLANs
        Network VSAN-3201
        Network VSAN-3202
        Network default
```

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***



VMkernel adapters

Device	Network Label	Switch	IP Address	TCP/I...	vMotion Traffic	FT...	Ma...	Virtual SAN
vmk0	esxi-mgm...	cvf1-vsa...	192.168.13.21		Disabled	D..	E...	Disabled
vmk1	VSAN-3201	cvf1-vsa...	10.32.1.21	De...	Disabled	D..	D...	Enabled
vmk2	VSAN-3203	cvf1-vsa...	10.32.3.21	De...	Enabled	D..	D...	Disabled

VMkernel network adapter: vmk1

All Properties IP Settings Policies

Port properties

Network label	VSAN-3201
TCP/IP stack	Default
Enabled services	Virtual SAN traffic

IPv4 settings

IPv4 address	10.32.1.21 (static)
Subnet mask	255.255.255.0
Default gateway for IPv4	192.168.13.254
DNS server addresses	192.168.10.141

IPv6 settings

IPv6 addresses	fe80::250:56ff:fe6c:ef1b/64
Default gateway for IPv6:	--

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***

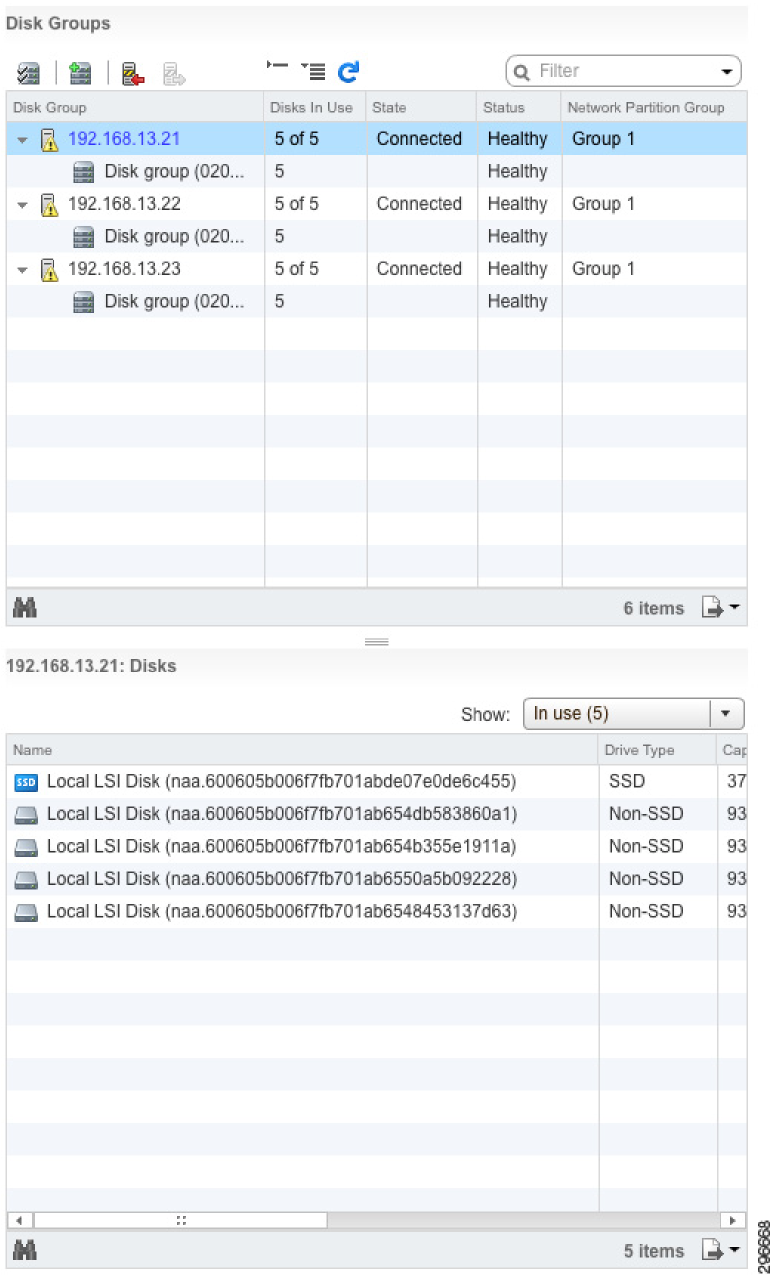
Host	VMKNIC	IP Address	VLAN	Mgmt	Virtual SAN	Vmotion
cvf1-dr-vsan1	vmk0	192.168.13.21	13	X		
	vmk1	10.32.1.21	3201		X	
	vmk2	10.32.3.21	3203			X
cvf1-dr-vsan2	vmk0	192.168.13.22	13	X		
	vmk1	10.32.1.22	3201		X	
	vmk2	10.32.3.22	3203			X
cvf1-dr-vsan3	vmk0	192.168.13.23	13	X		
	vmk1	10.32.1.23	3201		X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



Disk Groups

Disk Group	Disks In Use	State	Status	Network Partition Group
192.168.13.21	5 of 5	Connected	Healthy	Group 1
Disk group (020...	5		Healthy	
192.168.13.22	5 of 5	Connected	Healthy	Group 1
Disk group (020...	5		Healthy	
192.168.13.23	5 of 5	Connected	Healthy	Group 1
Disk group (020...	5		Healthy	

6 items

192.168.13.21: Disks

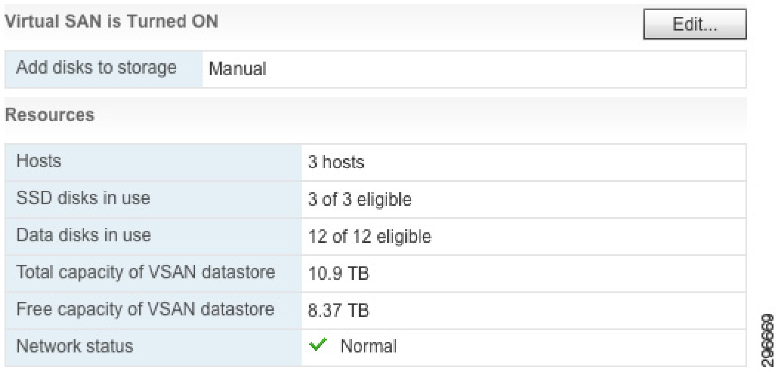
Show: In use (5)

Name	Drive Type	Cap
Local LSI Disk (naa.600605b006f7fb701abde07e0de6c455)	SSD	37
Local LSI Disk (naa.600605b006f7fb701ab654db583860a1)	Non-SSD	93
Local LSI Disk (naa.600605b006f7fb701ab654b355e1911a)	Non-SSD	93
Local LSI Disk (naa.600605b006f7fb701ab6550a5b092228)	Non-SSD	93
Local LSI Disk (naa.600605b006f7fb701ab6548453137d63)	Non-SSD	93

5 items

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

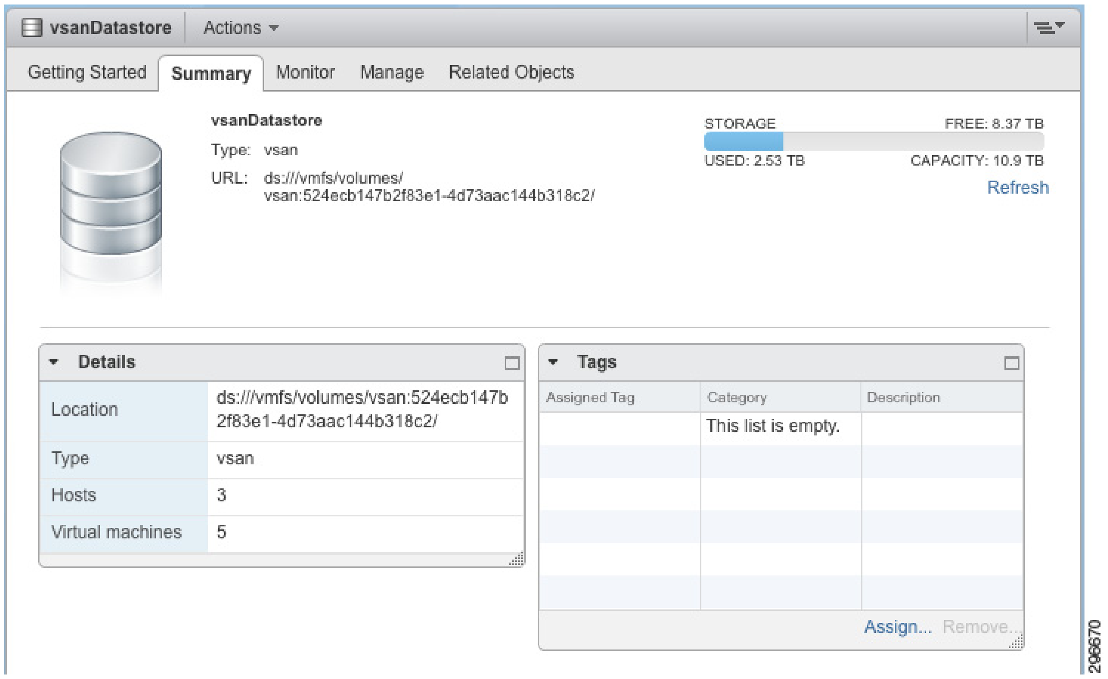


Add disks to storage	Manual

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



vsanDatastore
Type: vsan
URL: ds:///vmfs/volumes/vsan:524ecb147b2f83e1-4d73aac144b318c2/

FREE: 8.37 TB
USED: 2.53 TB
CAPACITY: 10.9 TB

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

Assigned Tag	Category	Description
	This list is empty.	

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.

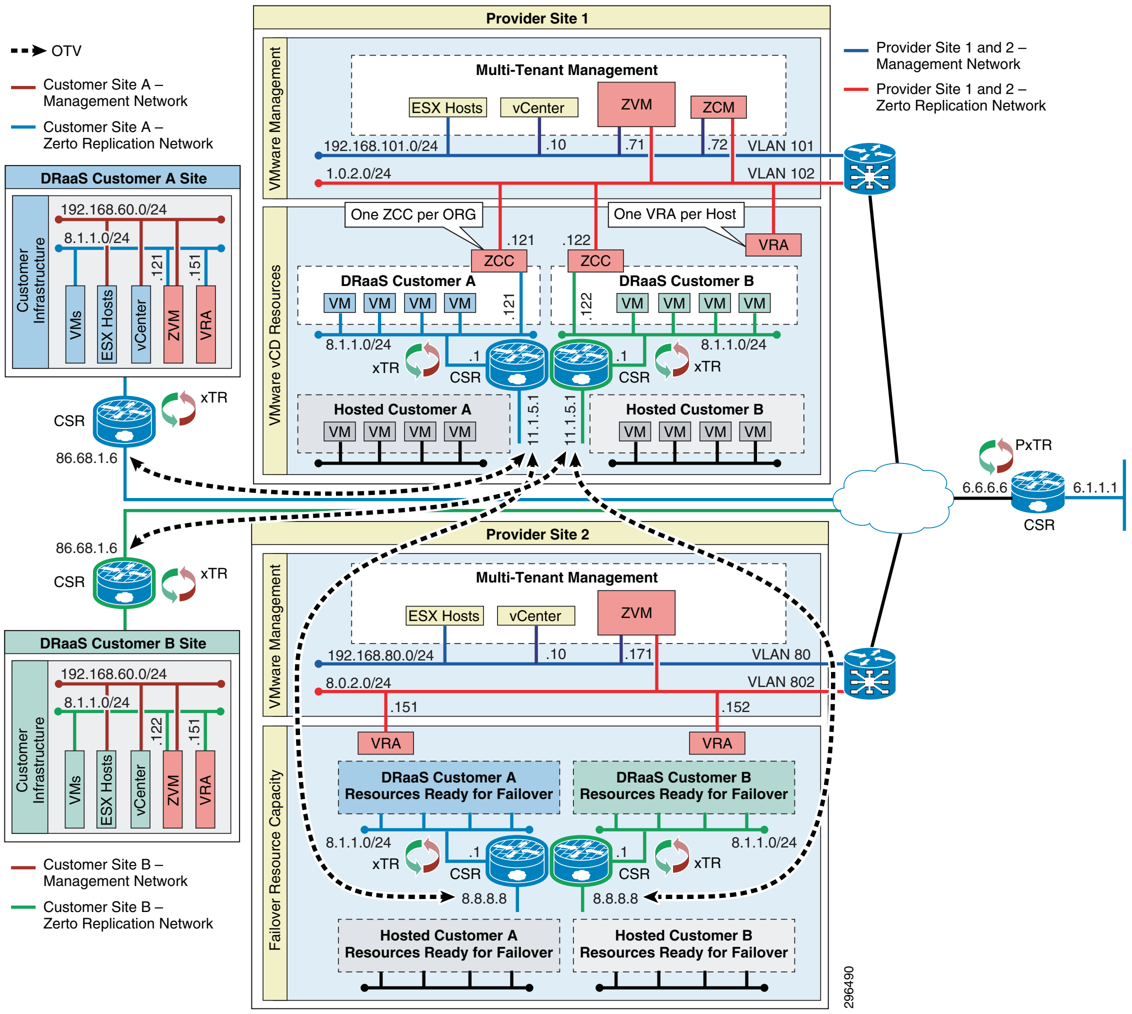
The DRaaS enterprises connected into the first provider site using OTV with CSR 1000V routers at each end. The OTV link carried both tenant traffic and Zerto control plane and replication traffic. The ICDR enterprises connected into the first provider site using OTV and had an OTV link between the provider 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.

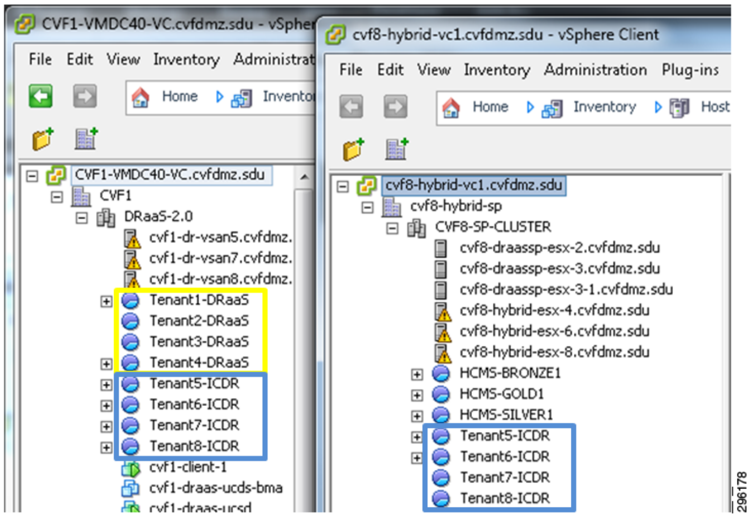
In the lab topology, each enterprise-to-CSP and CSP-to-CSP OTV link carried three VLANs, one for each tier of the three tier applications (e.g., web servers, application servers, database servers) that resided in each tenant. For the enterprise-to-CSP OTV link, the Zerto control and replication traffic 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

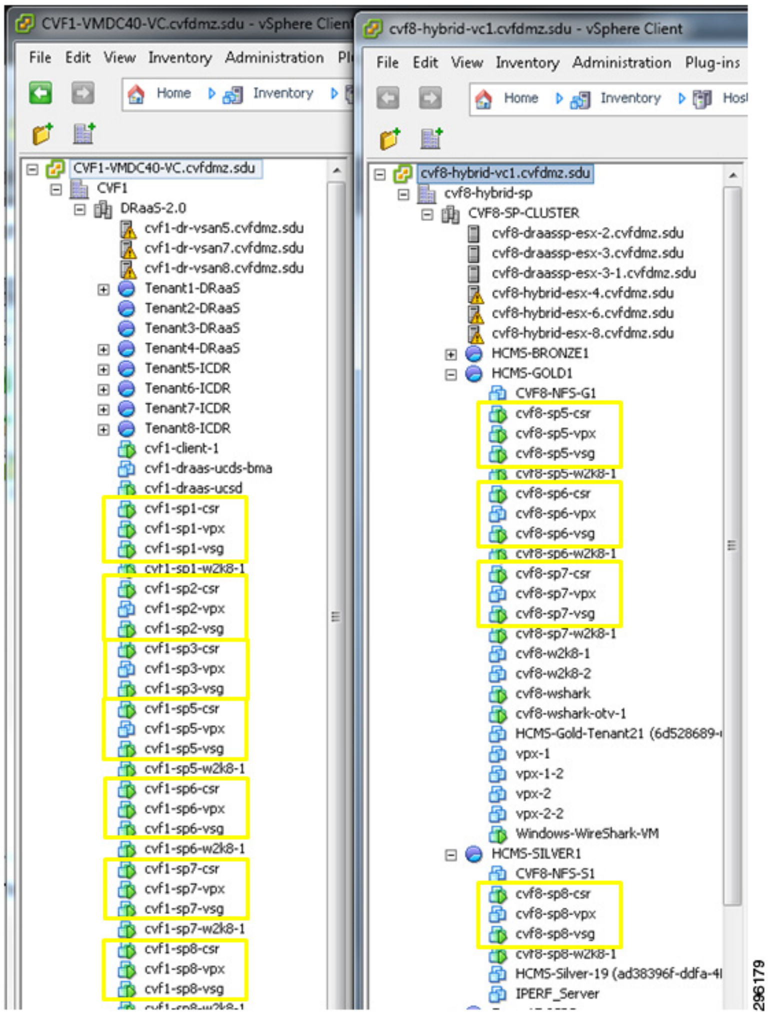

CVF1-VMDC40-VC.cvfdmz.sdu - vSpher
File Edit View Inventory Administrat
Home Invento
cvf8-hybrid-vc1.cvfdmz.sdu - vSphere Client
File Edit View Inventory Administration Plug-ins
Home Inventory Host
296178

```
CVF1-VMDC40-VC.cvfdmz.sdu
  CVF1
    DRaaS-2.0
      cvf1-dr-vsan5.cvfdmz.
      cvf1-dr-vsan7.cvfdmz.
      cvf1-dr-vsan8.cvfdmz.
      Tenant1-DRaaS
      Tenant2-DRaaS
      Tenant3-DRaaS
      Tenant4-DRaaS
      Tenant5-ICDR
      Tenant6-ICDR
      Tenant7-ICDR
      Tenant8-ICDR
      cvf1-client-1
      cvf1-draas-ucds-bma
      cvf1-draas-ucsd
```

```
cvf8-hybrid-vc1.cvfdmz.sdu
  cvf8-hybrid-sp
    CVF8-SP-CLUSTER
      cvf8-draassp-esx-2.cvfdmz.sdu
      cvf8-draassp-esx-3.cvfdmz.sdu
      cvf8-draassp-esx-3-1.cvfdmz.sdu
      cvf8-hybrid-esx-4.cvfdmz.sdu
      cvf8-hybrid-esx-6.cvfdmz.sdu
      cvf8-hybrid-esx-8.cvfdmz.sdu
      HCMS-BRONZE1
      HCMS-GOLD1
      HCMS-SILVER1
      Tenant5-ICDR
      Tenant6-ICDR
      Tenant7-ICDR
      Tenant8-ICDR
```

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

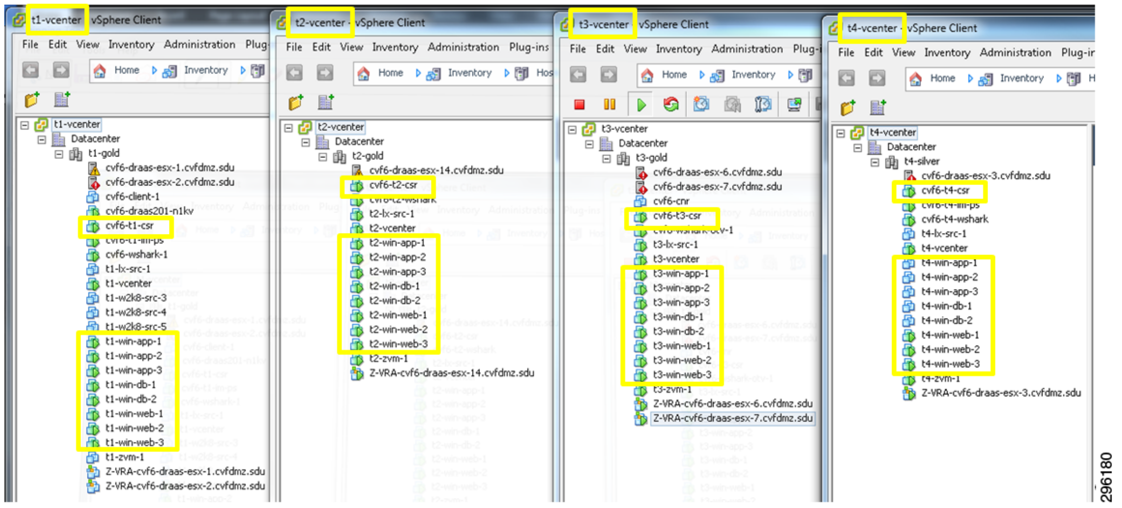

CVF1-VMDC40-VC.cvfdmz.sdu - vSphere Client
File Edit View Inventory Administration Pl
Home Inventory
cvf8-hybrid-vc1.cvfdmz.sdu - vSphere Client
File Edit View Inventory Administration Plug-ins
Home Inventory Host
296179

```
CVF1-VMDC40-VC.cvfdmz.sdu
  CVF1
    DRaaS-2.0
      cvf1-dr-vsan5.cvfdmz.sdu
      cvf1-dr-vsan7.cvfdmz.sdu
      cvf1-dr-vsan8.cvfdmz.sdu
      Tenant1-DRaaS
      Tenant2-DRaaS
      Tenant3-DRaaS
      Tenant4-DRaaS
      Tenant5-ICDR
      Tenant6-ICDR
      Tenant7-ICDR
      Tenant8-ICDR
      cvf1-client-1
      cvf1-draas-ucds-bma
      cvf1-draas-ucsd
      cvf1-sp1-csr
      cvf1-sp1-vpx
      cvf1-sp1-vsg
      cvf1-sp1-w2k8-1
      cvf1-sp2-csr
      cvf1-sp2-vpx
      cvf1-sp2-vsg
      cvf1-sp3-csr
      cvf1-sp3-vpx
      cvf1-sp3-vsg
      cvf1-sp5-csr
      cvf1-sp5-vpx
      cvf1-sp5-vsg
      cvf1-sp5-w2k8-1
      cvf1-sp6-csr
      cvf1-sp6-vpx
      cvf1-sp6-vsg
      cvf1-sp6-w2k8-1
      cvf1-sp7-csr
      cvf1-sp7-vpx
      cvf1-sp7-vsg
      cvf1-sp7-w2k8-1
      cvf1-sp8-csr
      cvf1-sp8-vpx
      cvf1-sp8-vsg
```

```
cvf8-hybrid-vc1.cvfdmz.sdu
  cvf8-hybrid-sp
    CVF8-SP-CLUSTER
      cvf8-draassp-esx-2.cvfdmz.sdu
      cvf8-draassp-esx-3.cvfdmz.sdu
      cvf8-draassp-esx-3-1.cvfdmz.sdu
      cvf8-hybrid-esx-4.cvfdmz.sdu
      cvf8-hybrid-esx-6.cvfdmz.sdu
      cvf8-hybrid-esx-8.cvfdmz.sdu
      HCMS-BRONZE1
      HCMS-GOLD1
        CVF8-NFS-G1
        cvf8-sp5-csr
        cvf8-sp5-vpx
        cvf8-sp5-vsg
        cvf8-sp5-w2k8-1
        cvf8-sp6-csr
        cvf8-sp6-vpx
        cvf8-sp6-vsg
        cvf8-sp6-w2k8-1
        cvf8-sp7-csr
        cvf8-sp7-vpx
        cvf8-sp7-vsg
        cvf8-sp7-w2k8-1
        cvf8-w2k8-1
        cvf8-w2k8-2
        cvf8-wshark
        cvf8-wshark-otv-1
        HCMS-Gold-Tenant21 (6d528689-
        vpx-1
        vpx-1-2
        vpx-2
        vpx-2-2
        Windows-WireShark-VM
      HCMS-SILVER1
        CVF8-NFS-S1
        cvf8-sp8-csr
        cvf8-sp8-vpx
        cvf8-sp8-vsg
        cvf8-sp8-w2k8-1
        HCMS-Silver-19 (ad38396f-ddfa-4
        IPERF_Server
```

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


t1-vcenter - vSphere Client
File Edit View Inventory Administration Plug-ins
Home Inventory
t2-vcenter - vSphere Client
t3-vcenter - vSphere Client
t4-vcenter - vSphere Client
296180

```
t1-vcenter
  Datacenter
    t1-gold
      cvf6-draas-esx-1.cvfdmz.sdu
      cvf6-draas-esx-2.cvfdmz.sdu
      cvf6-client-1
      cvf6-draas201-n1kv
      cvf6-t1-csr
      cvf6-t1-lm-ps
      cvf6-wshark-1
      t1-lx-src-1
      t1-vcenter
      t1-w2k8-src-3
      t1-w2k8-src-4
      t1-w2k8-src-5
      t1-win-app-1
      t1-win-app-2
      t1-win-app-3
      t1-win-db-1
      t1-win-db-2
      t1-win-web-1
      t1-win-web-2
      t1-win-web-3
      t1-zvm-1
      Z-VRA-cvf6-draas-esx-1.cvfdmz.sdu
      Z-VRA-cvf6-draas-esx-2.cvfdmz.sdu

t2-vcenter
  Datacenter
    t2-gold
      cvf6-draas-esx-14.cvfdmz.sdu
      cvf6-t2-csr
      cvf6-t2-wshark
      t2-lx-src-1
      t2-vcenter
      t2-win-app-1
      t2-win-app-2
      t2-win-app-3
      t2-win-db-1
      t2-win-db-2
      t2-win-web-1
      t2-win-web-2
      t2-win-web-3
      t2-zvm-1
      Z-VRA-cvf6-draas-esx-14.cvfdmz.sdu

t3-vcenter
  Datacenter
    t3-gold
      cvf6-draas-esx-6.cvfdmz.sdu
      cvf6-draas-esx-7.cvfdmz.sdu
      cvf6-cnr
      cvf6-t3-csr
      cvf6-wshark-otv-1
      t3-lx-src-1
      t3-vcenter
      t3-win-app-1
      t3-win-app-2
      t3-win-app-3
      t3-win-db-1
      t3-win-db-2
      t3-win-web-1
      t3-win-web-2
      t3-win-web-3
      t3-zvm-1
      Z-VRA-cvf6-draas-esx-6.cvfdmz.sdu
      Z-VRA-cvf6-draas-esx-7.cvfdmz.sdu

t4-vcenter
  Datacenter
    t4-silver
      cvf6-draas-esx-3.cvfdmz.sdu
      cvf6-t4-csr
      cvf6-t4-lm-ps
      cvf6-t4-wshark
      t4-lx-src-1
      t4-vcenter
      t4-win-app-1
      t4-win-app-2
      t4-win-app-3
      t4-win-db-1
      t4-win-db-2
      t4-win-web-1
      t4-win-web-2
      t4-win-web-3
      t4-zvm-1
      Z-VRA-cvf6-draas-esx-3.cvfdmz.sdu
```

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.

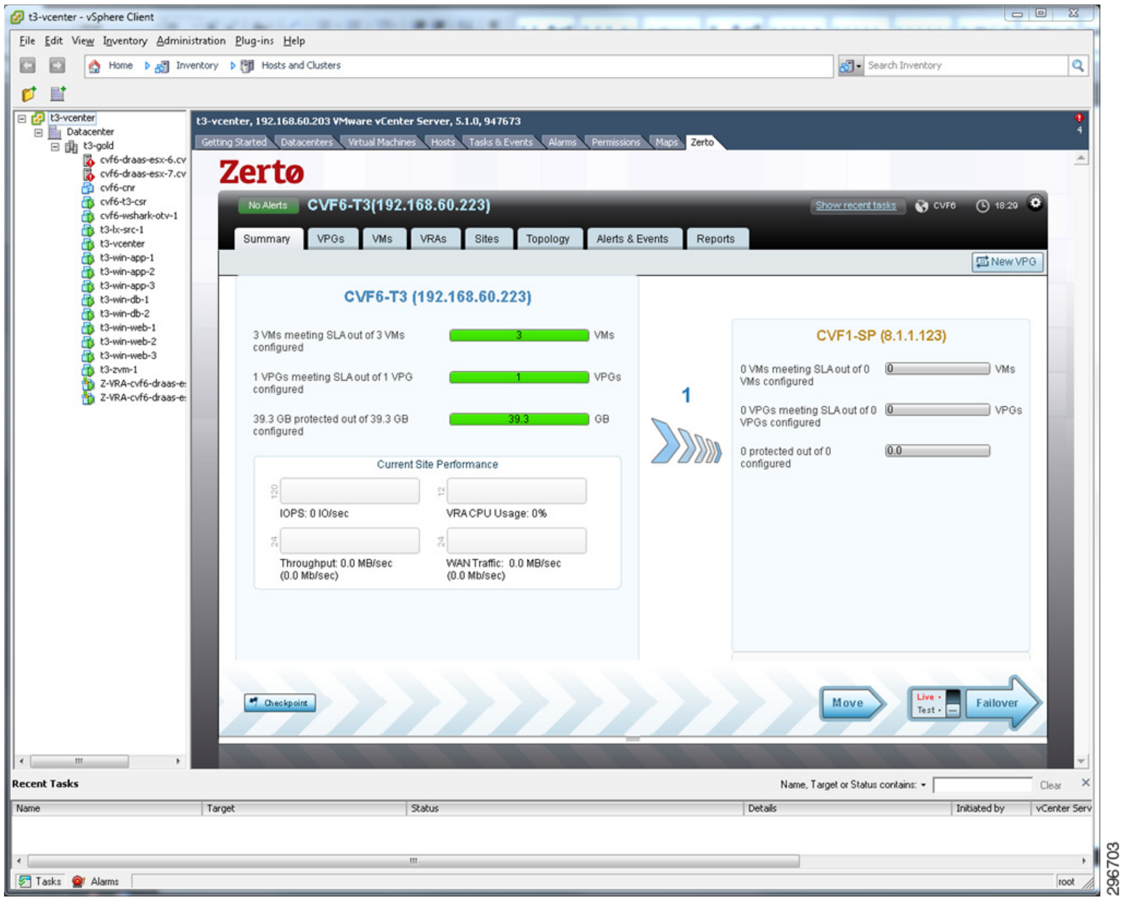
Note VPGs can have dozens of VMs in them, with no actual limit on the number of VMs per VPG, but other factors, such as application affinity grouping requirements, usually keep the number of VMs in a VPG to fewer than 50.

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


t3-vcenter - vSphere Client
File Edit View Inventory Administration Plug-ins Help
Home Inventory Hosts and Clusters
Search Inventory
t3-vcenter, 192.168.60.203 VMware vCenter Server, 5.1.0, 947673
Getting Started Datacenters Virtual Machines Hosts Tasks & Events Alarms Permissions Maps Zerto
Zerto
No Alerts
CVF6-T3(192.168.60.223)
Show recent tasks
Summary VPGs VMs VRAs Sites Topology Alerts & Events Reports
New VPG
Checkpoint
Move
Live Test
Failover
Recent Tasks
Tasks Alarms
296703

CVF6-T3 (192.168.60.223)

3 VMs meeting SLA out of 3 VMs configured — 3 VMs

1 VPGs meeting SLA out of 1 VPG configured — 1 VPGs

39.3 GB protected out of 39.3 GB configured — 39.3 GB

Current Site Performance

IOPS: 0 IO/sec

VRA CPU Usage: 0%

Throughput: 0.0 MB/sec (0.0 Mb/sec)

WAN Traffic: 0.0 MB/sec (0.0 Mb/sec)

CVF1-SP (8.1.1.123)

0 VMs meeting SLA out of 0 VMs configured — 0 VMs

0 VPGs meeting SLA out of 0 VPGs configured — 0 VPGs

0 protected out of 0 configured — 0.0

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:

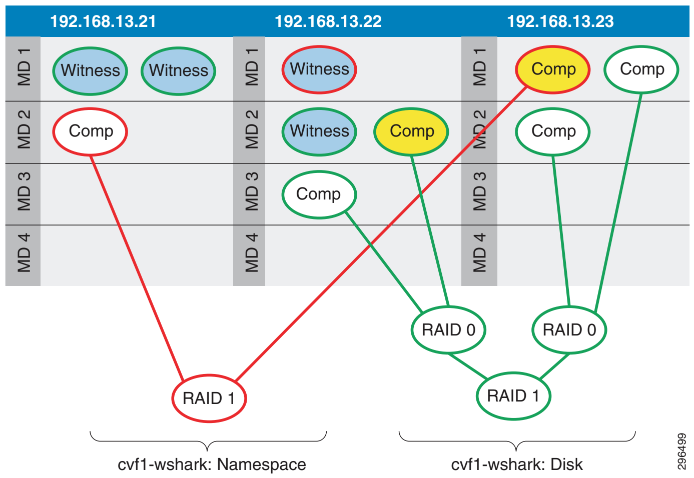
- Three ESXi hosts running version 5.5
- Virtual SAN datastore made of twelve 1 TB spindles and three 400 GB SSDs
- A total datastore capacity of 12.3 TB (raw).
- Virtual SAN default policies set, including *StripeWidth* of 1 and *FailuresToTolerate* of 1.
- VMware Distributed Resource Scheduler (DRS) 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:

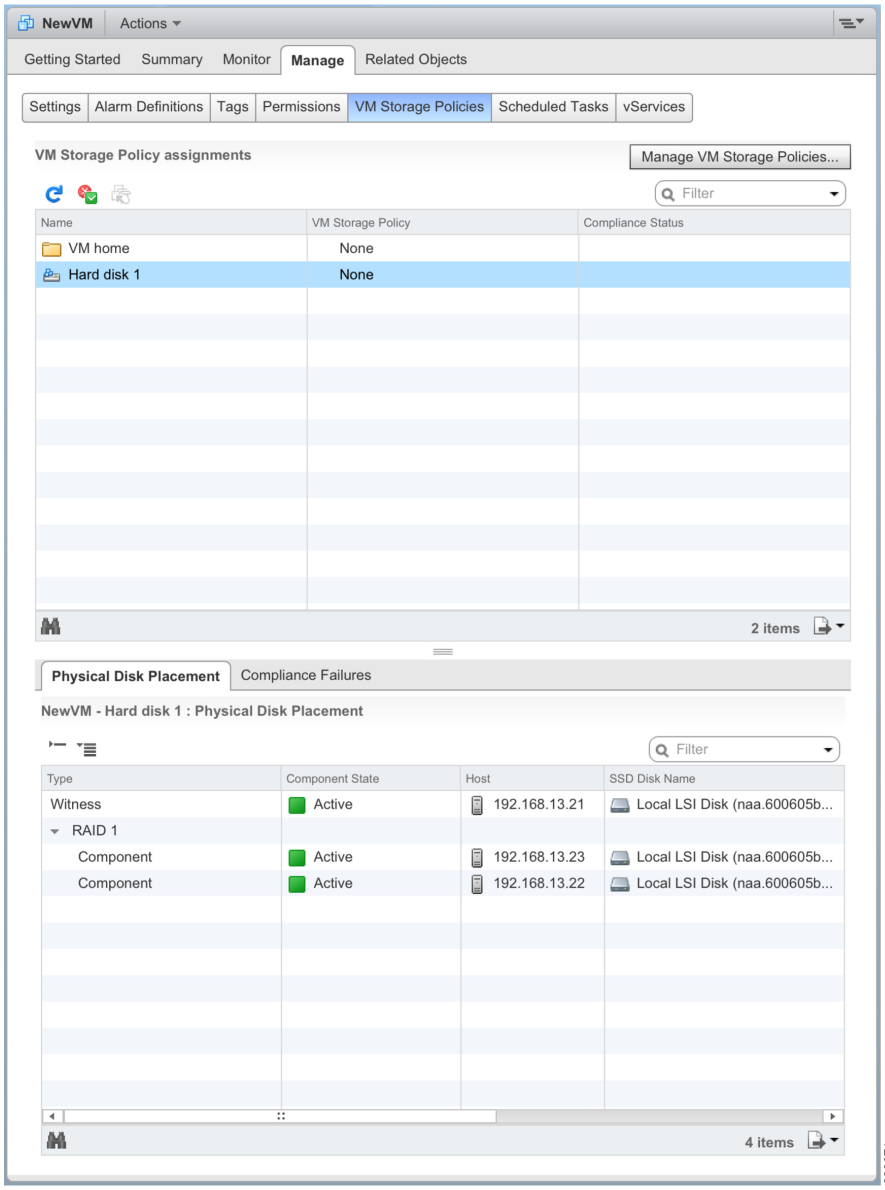
Primary Witnesses—The data components are spread over only 2 nodes (which is not greater than 2*FTT+1), so we need to deploy a primary witness on the excluded node to achieve 2*FTT+1.

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



VM Storage Policy assignments

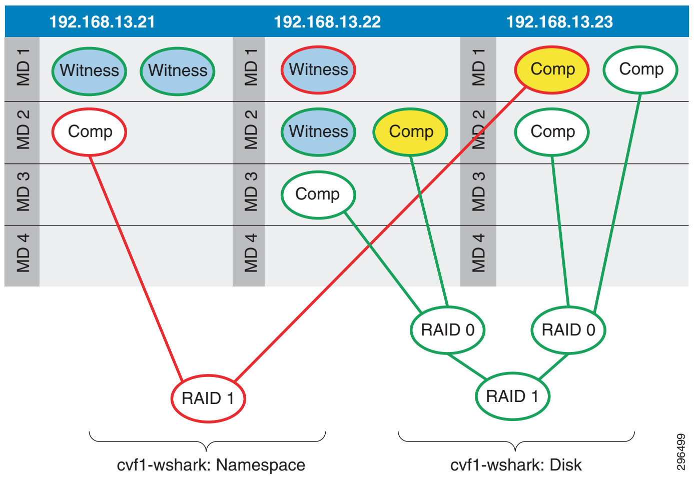
Name	VM Storage Policy	Compliance Status
VM home	None	
Hard disk 1	None	

NewVM - Hard disk 1 : Physical Disk Placement

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

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

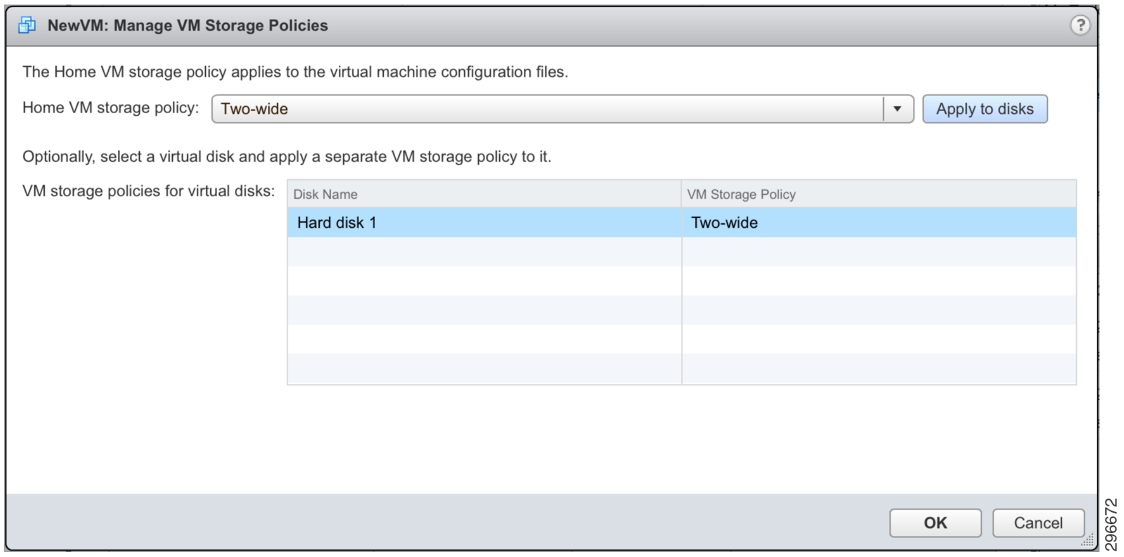


There are several key changes to take notice of, including:

- 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 Home VM storage policy applies to the virtual machine configuration files.

Home VM storage policy: Two-wide

Optionally, select a virtual disk and apply a separate VM storage policy to it.

VM storage policies for virtual disks:

Disk Name	VM Storage Policy
Hard disk 1	Two-wide

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:

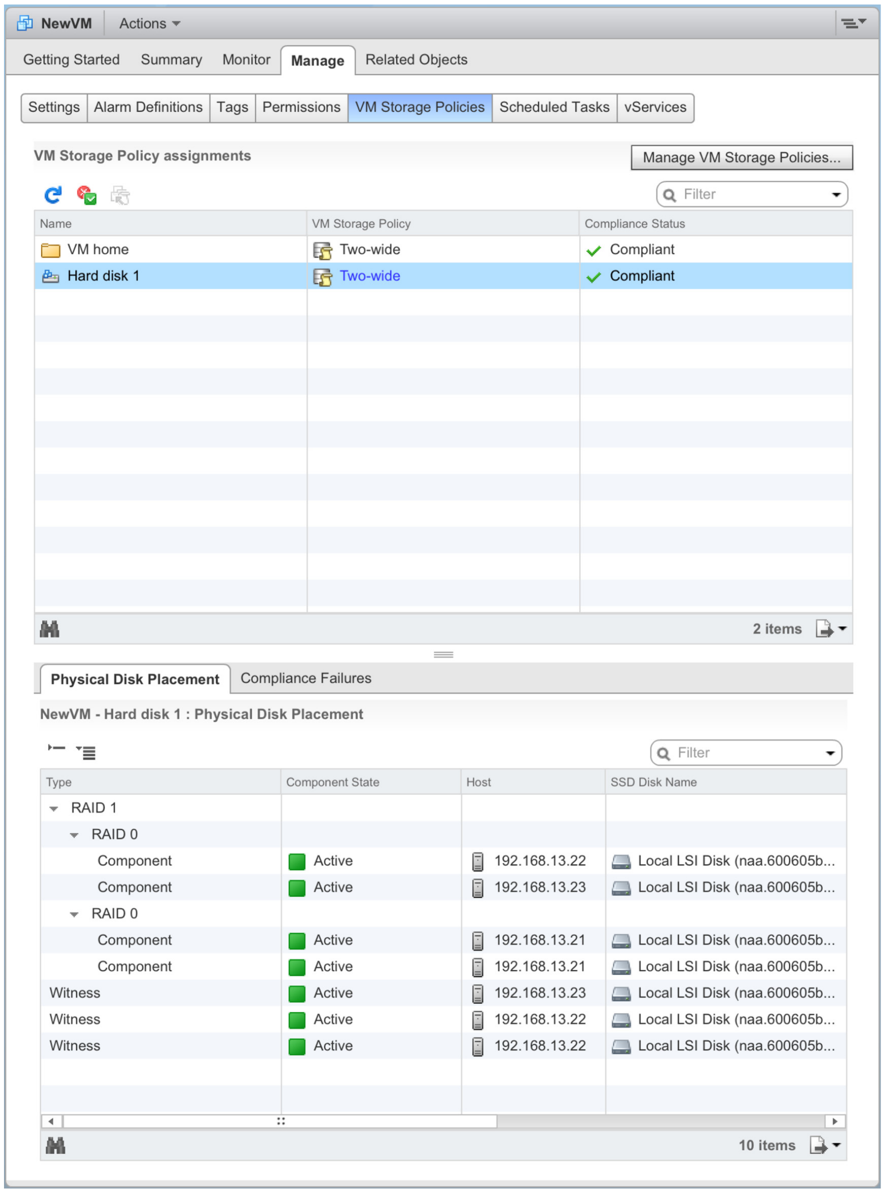
Primary Witnesses—The data components are spread over all 3 nodes (which is equal to 2*FTT+1), so no primary witnesses are needed.

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



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

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...

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
	Component	Active	192.168.13.23
Disk 1	Witness	Active	192.168.13.22
	RAID 1		
	Component	Active	192.168.13.21
	Component	Active	192.168.13.23
Disk 2	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
	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 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 calculate 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.