



HEALTH CHECK HANDBOOK

VIRTUAL CONTROL

VMWARE CLOUD FOUNDATION SOLUTIONS

vSAN Health Check Handbook

vSAN cluster health assessment including disk groups, resync status, capacity, performance, network configuration, and data integrity checks.

VSAN

DISK GROUPS

CAPACITY

RESYNC

PERFORMANCE

VCF 9.0

VMware Cloud Foundation

vSAN Health Check Handbook

Comprehensive Health Verification for vSAN in VMware Cloud Foundation 9

Prepared by: Virtual Control LLC

Date: March 2026

Version: 1.0

Classification: Internal Use

Platform: VMware Cloud Foundation 9.x / vSAN 8.x ESA & OSA

Table of Contents

1. Overview & Purpose

- 1.1 Scope
- 1.2 When to Run Health Checks
- 1.3 Target Audience

2. Prerequisites

- 2.1 Access Requirements
- 2.2 Tools & Utilities
- 2.3 RVC Setup
- 2.4 PowerCLI Setup

3. Quick Reference Summary Table

4. vSAN Cluster Health

- 4.1 Cluster Health List
- 4.2 Proactive Health Tests
- 4.3 vSAN Health Service
- 4.4 Cluster Membership

5. Disk Group Status

- 5.1 Disk Group Listing
- 5.2 Cache & Capacity Tier Status
- 5.3 SMART Data Analysis
- 5.4 vSAN Storage List

6. Capacity Analysis

- 6.1 Capacity Overview
- 6.2 Dedup & Compression Savings
- 6.3 Thin Provisioning
- 6.4 Slack Space Calculation
- 6.5 Capacity Thresholds

7. Resync Status

- 7.1 Active Resyncs
- 7.2 Resync ETA
- 7.3 Performance Impact

8. vSAN Network Health

- 8.1 vSAN VMkernel Adapters
- 8.2 Multicast & Unicast Check
- 8.3 Jumbo Frame Validation
- 8.4 Network Partition Detection
- 8.5 Witness Host Connectivity

9. Performance Health

- 9.1 IOPS & Latency
- 9.2 Congestion
- 9.3 Outstanding IO
- 9.4 vscsiStats
- 9.5 Performance Service

10. Object Health & Compliance

- 10.1 Object Count
- 10.2 Compliance State
- 10.3 Inaccessible Objects
- 10.4 Reduced Redundancy Objects

11. Stretched Cluster Health

- 11.1 Preferred & Secondary Site
- 11.2 Witness Host
- 11.3 Site Affinity Rules
- 11.4 Inter-Site Latency

12. Fault Domains

- 12.1 Fault Domain Configuration
- 12.2 Host Distribution
- 12.3 Policy Compliance with Fault Domains

13. vSAN Health Service Detail

- 13.1 Health Service Status
- 13.2 Test Categories
- 13.3 Silenced Alarms

14. HCL Compliance

- 14.1 Controller, Driver & Firmware
- 14.2 HCL Database Update

15. Port Reference Table

16. Common Issues & Remediation

- 16.1 Disk Failures
- 16.2 Network Partition

- 16.3 Resync Storms
- 16.4 Performance Degradation
- 16.5 CLOM Errors

17. CLI Quick Reference Card

18. PowerCLI Quick Reference

1. Overview & Purpose

This handbook provides a systematic, step-by-step approach to verifying the health and operational readiness of vSAN clusters running within VMware Cloud Foundation (VCF) 9 environments. It covers both vSAN Original Storage Architecture (OSA) and the vSAN Express Storage Architecture (ESA), which is the recommended configuration in VCF 9.

1.1 Scope

This document covers the following health check domains:

Domain	Description
Cluster Health	Overall cluster status, membership, and health service results
Disk Group Status	Physical disk health, SMART data, cache and capacity tiers
Capacity Analysis	Space utilization, deduplication, compression, slack space
Resync Operations	Active resyncs, component movement, ETA, and impact
Network Health	VMkernel configuration, connectivity, jumbo frames, partitions
Performance	IOPS, latency, congestion, outstanding IO metrics
Object Health	Object compliance, accessibility, redundancy state
Stretched Cluster	Site configuration, witness host, inter-site latency
Fault Domains	Domain layout, host distribution, policy interaction
HCL Compliance	Controller, driver, and firmware compatibility verification

VCF 9 Context: In VCF 9, vSAN is the default and only supported principal storage for the management domain. Workload domains may use vSAN, NFS, VMFS on FC, or vVols. This handbook focuses on the vSAN principal storage layer but applies to any vSAN-backed workload domain as well.

1.2 When to Run Health Checks

Health checks should be executed at these critical intervals:

Trigger	Frequency	Checks
Routine maintenance	Weekly	Full suite
Pre-upgrade (VCF lifecycle)	Before each LCM bundle	Full suite
Post-upgrade	Immediately after LCM completes	Full suite
Host addition/removal	After cluster change	Cluster, disk, network, capacity
Disk replacement	After replacement completes	Disk group, resync, object health
Network change	After vDS/vmkernel modification	Network health, connectivity
Performance complaint	On demand	Performance, congestion, resync
After power event	After datacenter power restoration	Full suite
Pre-expansion	Before adding workload domains	Capacity, performance baseline

1.3 Target Audience

This handbook is intended for:

- **VCF Administrators** managing day-to-day operations
- **Storage Engineers** responsible for vSAN health and capacity
- **Network Engineers** validating vSAN transport configuration
- **Operations Teams** performing routine maintenance windows
- **Escalation Engineers** troubleshooting vSAN issues

2. Prerequisites

2.1 Access Requirements

Requirement	Detail
vCenter SSO Admin	<code>administrator@vsphere.local</code> or equivalent role
ESXi Root Access	SSH enabled on target hosts (temporarily, disable after)
SDDC Manager Access	Admin-level access for LCM and inventory queries
vSAN Witness Host	Root access if stretched cluster is deployed
Network Access	Ability to reach vSAN VMkernel IPs on port 2233

Security Note: SSH access to ESXi hosts should be enabled only for the duration of the health check. In VCF 9, SSH is disabled by default and locked down via the Security Configuration Guide. Always disable SSH and re-enable lockdown mode after completing CLI-based checks.

2.2 Tools & Utilities

Required Tools

Tool	Version	Purpose
<code>esxcli</code>	Built into ESXi 8.x	Primary CLI for vSAN health checks
<code>vSAN Health Service</code>	Built into vCenter 8.x	Automated health test framework
PowerCLI	13.3+	Scripted health checks and reporting
RVC (Ruby vSphere Console)	Built into vCenter appliance	Deep vSAN diagnostics
<code>vmkping</code>	Built into ESXi	vSAN network validation
<code>vsanDiskMgmt</code>	Built into ESXi	Disk management and SMART queries
Python (<code>pyVmomi</code>)	8.0+	API-driven automation

Optional Tools

Tool	Purpose
vSAN Observer	Real-time performance monitoring (HTML5 dashboard)
vRealize Operations / Aria Operations	Trending, capacity forecasting
VDT (VMware Diagnostic Tool)	Automated diagnostic collection
SOS Report	Support bundle generation

2.3 RVC Setup

The Ruby vSphere Console is accessed directly from the vCenter Server Appliance (VCSA).

```
# SSH to VCSA
ssh root@vcsa-01.vcf.local

# Launch RVC
rvc administrator@vsphere.local@localhost

# Navigate to the vSAN cluster
cd /localhost/SDDC-Datacenter/computers/SDDC-Cluster1

# Run the vSAN health check
vsan.health.health_summary .
```

Key RVC Commands for vSAN

```
# Full cluster health summary
vsan.health.health_summary /localhost/datacenter/computers/cluster

# Disk balance check
vsan.disks_stats /localhost/datacenter/computers/cluster

# Object placement info
vsan.object_info /localhost/datacenter/computers/cluster

# Network partition check
vsan.cluster_info /localhost/datacenter/computers/cluster

# Resync dashboard
vsan.resync_dashboard /localhost/datacenter/computers/cluster

# Performance diagnostics
vsan.perf.stats_object_list /localhost/datacenter/computers/cluster
```

2.4 PowerCLI Setup

```
# Install or update PowerCLI
Install-Module -Name VMware.PowerCLI -Scope CurrentUser -Force

# Configure certificate handling for lab/internal environments
Set-PowerCLIConfiguration -InvalidCertificateAction Ignore -Confirm:$false

# Connect to vCenter
Connect-VIServer -Server vcsa-01.vcf.local -User administrator@vsphere.local -
Password '<password>'

# Verify vSAN module is loaded
Get-Module VMware.VimAutomation.Storage -ListAvailable
```

PowerCLI 13.3+: VCF 9 ships with vCenter 8.x which requires PowerCLI 13.3 or later for full vSAN ESA cmdlet support. Ensure you are running the latest version before executing vSAN commands.

3. Quick Reference Summary Table

The following table provides a consolidated view of every health check in this handbook with pass/warn/fail criteria.

#	Check	CLI / Method	PASS	WARN	FAIL
1	Cluster Health Status	<code>esxcli vsan health cluster list</code>	All tests green	Any test yellow	Any test red
2	Cluster Membership	<code>esxcli vsan cluster get</code>	All hosts in cluster	Host count mismatch	Partitioned cluster
3	Disk Group Status	<code>esxcli vsan storage list</code>	All disks healthy	Disk degraded	Disk absent/failed
4	SMART Health	<code>esxcli vsan debug disk smart get</code>	All attributes OK	Wear leveling > 80%	Reallocated sectors > 0
5	Capacity Used %	vSAN UI / PowerCLI	< 70%	70-80%	> 80%
6	Slack Space	Calculated	>= 25% of raw	15-25% of raw	< 15% of raw

7	Dedup/Compression Ratio	vSAN UI	> 1.5x	1.0-1.5x	< 1.0x (overhead)
8	Active Resyncs	<code>esxcli vsan debug resync summary</code>	0 active	< 100 components	> 100 components
9	Resync ETA	vSAN UI	< 1 hour	1-8 hours	> 8 hours
10	vSAN VMkernel Config	<code>esxcli network ip interface list</code>	vSAN vmk on each host	MTU mismatch	vmk missing
11	Jumbo Frame Test	<code>vmkping -s 8972 -d</code>	0% packet loss	Intermittent loss	Complete failure
12	Network Partition	Health service	No partition	N/A	Partition detected
13	Read Latency	vSAN perf service	< 1 ms	1-5 ms	> 5 ms
14	Write Latency	vSAN perf service	< 2 ms	2-10 ms	> 10 ms
15	Congestion	<code>esxcli vsan debug controller list</code>	0	1-40	> 40
16	Outstanding IO	<code>vsish</code>	< 32	32-64	> 64
17	Object Health	<code>esxcli vsan debug object health summary</code>	All healthy	Reduced redundancy	Inaccessible objects
18	Policy Compliance	vSAN UI / PowerCLI	All compliant	Non-compliant (rebuilding)	Non-compliant (stuck)
19	Witness Host	<code>esxcli vsan cluster get</code>	Connected	High latency	Disconnected
20	Inter-Site Latency	<code>vmkping</code>	< 5 ms RTT	5-100 ms	> 200 ms / timeout
21	Fault Domain Count	vSAN UI	>= 3 FDs	2 FDs	1 FD (no protection)
22	HCL Controller	Health service	Certified	DB outdated > 90 days	Not on HCL
23	HCL Driver/Firmware	Health service	Matched	Minor mismatch	Critical mismatch
24	Health Service Status	vCenter UI	Running, recent test	Last test > 24h ago	Service not running
25	Silenced Alarms	Health service	0 silenced	1-3 silenced	> 3 silenced

4. vSAN Cluster Health

4.1 Cluster Health List

The primary entry point for vSAN health is the `esxcli vsan health cluster list` command. This queries the vSAN health service and returns the state of every registered health test.

Command

```
esxcli vsan health cluster list
```

Expected Output (Healthy)

Group: Cluster

Overall Health: green

Tests:

vSAN Health Service Up-To-Date:	green
vSAN Build Recommendation Engine Health:	green
vSAN CLOMD Liveness:	green
vSAN Disk Balance:	green
vSAN Object Health:	green
vSAN Cluster Partition:	green

Group: Network

Overall Health: green

Tests:

All Hosts Have a vSAN VMkernel Adapter:	green
All Hosts Have Matching Subnets:	green
vSAN: Basic (Unicast) Connectivity Check:	green
vSAN: MTU Check (Ping with Large Packet Size):	green
vMotion: Basic Connectivity Check:	green

Group: Physical Disk

Overall Health: green

Tests:

vSAN Disk Health:	green
Metadata Health:	green
Component Metadata Health:	green
Congestion:	green
Disk Space Usage:	green

Group: Data

Overall Health: green

Tests:

vSAN Object Health: green

vSAN VM Health: green

Group: Limits

Overall Health: green

Tests:

Current Cluster Situation: green

After 1 Additional Host Failure: green

Host Component Limit: green

Pass / Warn / Fail Criteria

Result	Condition	Action
PASS	All groups show green	No action required
WARN	One or more tests show yellow	Investigate the specific test; see relevant section of this handbook
FAIL	Any test shows red	Immediate investigation required; do NOT proceed with maintenance

Remediation: If any test returns red, drill into the specific group. Run `esxcli vsan health cluster list -t "test name"` to get details on the specific failing test. Cross-reference with the relevant section of this handbook for targeted remediation steps.

4.2 Proactive Health Tests

vSAN proactive tests simulate failure scenarios to predict cluster behavior under stress.

Proactive Rebalance Test

```
esxcli vsan health cluster list -t "vSAN Disk Balance"
```

What It Checks

- Disk usage variance across all capacity disks
- Whether any single disk has > 30% more usage than the average
- Recommendations for proactive rebalancing

Expected Output (Healthy)

```
Health Test: vSAN Disk Balance
Status: green
Description: Disks are well balanced. Max variance: 8%
```

Triggering a Manual Rebalance

```
# From RVC
vsan.proactive_rebalance /localhost/datacenter/computers/cluster --start
```

Caution: Proactive rebalance generates resync traffic. Only run during maintenance windows or periods of low IO activity. The rebalance operation can be throttled or stopped at any time.

4.3 vSAN Health Service

The vSAN Health Service runs within vCenter and executes periodic health tests.

Verify Health Service Status

```
# On any ESXi host in the cluster
esxcli vsan health cluster list -t "vSAN Health Service Up-To-Date"
```

Force a Health Check Run via PowerCLI

```
# Get the vSAN cluster
$cluster = Get-Cluster -Name "SDDC-Cluster1"

# Get vSAN view
$vsanClusterHealthSystem = Get-VsanView -Id "VsanVcClusterHealthSystem-vsan-cluster-health-system"

# Run health check
$vsanClusterHealthSystem.VsanQueryVcClusterHealthSummary(
    $cluster.ExtensionData.MoRef,
    $null, $null, $true, $null, $null, "defaultView"
)
```

Expected Behavior

Attribute	Expected
Service Running	Yes
Last Test Time	Within 60 minutes
Test Result Format	Per-group green/yellow/red
Auto-Run Interval	Every 60 minutes (configurable)

4.4 Cluster Membership

Every host in a vSAN cluster must be an active member. Use `esxcli vsan cluster get` to verify.

Command

```
esxcli vsan cluster get
```

Expected Output (Healthy)

Cluster Information

Enabled: true

Current Local Time: 2026-03-26T14:30:00Z

Local Node UUID: 5f3e8c7a-xxxx-xxxx-xxxx-xxxxxxxxxxxx

Local Node Type: NORMAL

Local Node State: MASTER

Member Count: 4

Sub-Cluster Member UUIDs: 5f3e8c7a-..., 6a4f9d8b-..., 7b5a0e9c-..., 8c6b1fad-...

Sub-Cluster Membership Entry Revision: 12

Sub-Cluster Member Count: 4

Maintenance Mode State: OFF

Verify From All Hosts

```
# Run on each ESXi host to ensure consistent membership
for host in esx01 esx02 esx03 esx04; do
  echo "=== $host ==="
  ssh root@$host 'esxcli vsan cluster get | grep "Member Count"'
done
```

Result	Condition	Action
PASS	Member Count matches expected host count on ALL hosts	Healthy
WARN	A host shows BACKUP instead of MASTER / AGENT	Verify roles; may be transitional
FAIL	Member count differs between hosts (split-brain)	Network partition detected -- see Section 8.4

Remediation (Split-Brain):

1. Check vSAN network connectivity between all hosts: `vmkping -I vmk1 <target_ip>`
2. Verify the vSAN portgroup is consistent across all hosts
3. Check for physical switch issues on the vSAN VLAN
4. If a host is isolated, restart the vSAN CLOMD service: `/etc/init.d/clomd restart`
5. As a last resort, remove and re-add the host to the vSAN cluster

5. Disk Group Status

5.1 Disk Group Listing

In vSAN OSA, storage is organized into disk groups (1 cache SSD + 1-7 capacity disks). In vSAN ESA (VCF 9 default), all NVMe devices participate in a single storage pool without a separate cache tier.

Command (OSA)

```
esxcli vsan storage list
```

Expected Output (OSA Healthy)

```
Device: naa.55cd2e414f5356c0
  Display Name: naa.55cd2e414f5356c0
  Is SSD: true
  In CMMDS: true
  On-disk Format Version: 15
  Is Capacity Tier: false
  Is Cache Tier: true
  RAID Level: NA
  vSAN UUID: 52e9a1f4-xxxx-xxxx-xxxx-xxxxxxxxxxxx
  vSAN Disk Group UUID: 52e9a1f4-xxxx-xxxx-xxxx-xxxxxxxxxxxx
  vSAN Disk Group Name: naa.55cd2e414f5356c0
  Health Status: Healthy

Device: naa.55cd2e414f53789a
  Display Name: naa.55cd2e414f53789a
  Is SSD: true
  In CMMDS: true
  On-disk Format Version: 15
  Is Capacity Tier: true
  Is Cache Tier: false
  RAID Level: NA
  vSAN UUID: 63fa2b05-xxxx-xxxx-xxxx-xxxxxxxxxxxx
  Health Status: Healthy
```

Command (ESA)

```
esxcli vsan storage list
```

Expected Output (ESA Healthy)

```
Device: t10.NVMe____Dell_Ent_NVMe_v2_AGN_RI_U.2_1.6TB
  Display Name: Dell Ent NVMe AGN RI U.2 1.6TB
  Is SSD: true
  In CMMDS: true
  On-disk Format Version: 19
  Is Capacity Tier: true
  Is Cache Tier: false
  ESA Eligible: true
  Storage Pool UUID: 74ab3c16-xxxx-xxxx-xxxx-xxxxxxxxxxxx
  Health Status: Healthy
```

5.2 Cache & Capacity Tier Status

Check Individual Disk Status

```
# List all vSAN disks with health  
esxcli vsan storage list | grep -E "Display Name|Health Status|Is Cache|Is  
Capacity"
```

Result	Condition	Action
PASS	All disks: Health Status: Healthy	No action
WARN	Any disk: Health Status: Degraded	Schedule replacement at next window
FAIL	Any disk: Health Status: Failed or missing	Immediate replacement required

Critical: If a cache tier disk fails in OSA, the ENTIRE disk group goes offline. All components on capacity disks in that group become unavailable until the cache disk is replaced and the disk group is recreated. In ESA, single disk failures are tolerated without disk group loss.

5.3 SMART Data Analysis

Self-Monitoring, Analysis, and Reporting Technology (SMART) provides early warning of disk failure.

Command

```
esxcli vsan debug disk smart get -d naa.55cd2e414f5356c0
```

Expected Output (Healthy)

Parameter	Value	Threshold	Worst	Status
-----	-----	-----	-----	-----
Health Status	OK	N/A	N/A	OK
Media Wearout Indicator	98	0	98	OK
Write Error Count	0	0	0	OK
Read Error Count	0	0	0	OK
Power-on Hours	14820	0	14820	OK
Power Cycle Count	12	0	12	OK
Reallocated Sector Count	0	0	0	OK
Uncorrectable Error Count	0	0	0	OK
Temperature Celsius	34	0	42	OK

Critical SMART Attributes to Monitor

Attribute	PASS	WARN	FAIL
Media Wearout Indicator	> 20% remaining	5-20% remaining	< 5% remaining
Reallocated Sector Count	0	1-10	> 10
Uncorrectable Error Count	0	1-5	> 5
Temperature Celsius	< 50C	50-70C	> 70C
Write Error Count	0	1-10	> 10
Read Error Count	0	1-10	> 10

Remediation (Failing SMART):

1. Open a VMware SR or OEM hardware support case for disk replacement
2. Place the host in maintenance mode (ensure evacuate data): `esxcli system maintenanceMode set -e true -m ensureAccessibility`

3. Remove the disk from the disk group: `esxcli vsan storage remove -d naa.xxxx`
4. Physically replace the disk
5. Add the new disk: `esxcli vsan storage add -d naa.xxxx -s naa.cache_disk`
6. Exit maintenance mode: `esxcli system maintenanceMode set -e false`

5.4 vSAN Storage List -- Complete Output

Full Storage Inventory Command

```
# Comprehensive disk listing with all properties
esxcli vsan storage list --format=xml
```

PowerCLI Alternative

```
# Get all vSAN disk information
$cluster = Get-Cluster "SDDC-Cluster1"
$vsanDisks = Get-VsanDisk -Cluster $cluster

foreach ($disk in $vsanDisks) {
    [PSCustomObject]@{
        Host           = $disk.VsanDiskGroup.VMHost.Name
        DiskGroup       = $disk.VsanDiskGroup.Name
        CanonicalName   = $disk.CanonicalName
        IsSSD           = $disk.IsSsd
        IsCacheDisk     = $disk.IsCacheDisk
        CapacityGB      = [math]::Round($disk.CapacityGB, 2)
    }
} | Format-Table -AutoSize
```

6. Capacity Analysis

6.1 Capacity Overview

Command

```
# From any ESXi host in the cluster
esxcli vsan debug object health summary get
```

PowerCLI Method (Recommended)

```
$cluster = Get-Cluster "SDDC-Cluster1"
$spaceReport = Get-VsanSpaceUsage -Cluster $cluster

# Display capacity summary
[PSCustomObject]@{
    "Total Capacity (TB)"      = [math]::Round($spaceReport.TotalCapacityGB /
1024, 2)
    "Used Capacity (TB)"      = [math]::Round($spaceReport.UsedCapacityGB / 1024,
2)
    "Free Capacity (TB)"      = [math]::Round($spaceReport.FreeCapacityGB / 1024,
2)
    "Used %"                  = [math]::Round(($spaceReport.UsedCapacityGB /
$spaceReport.TotalCapacityGB) * 100, 1)
}
```

Expected Output

```
Total Capacity (TB)      : 23.64
Used Capacity (TB)       : 9.82
Free Capacity (TB)       : 13.82
Used %                   : 41.5
```

6.2 Dedup & Compression Savings

When deduplication and compression are enabled (common in vSAN ESA and optional in OSA all-flash), significant space savings are expected.

Command

```
esxcli vsan debug space show
```

PowerCLI Method

```
$cluster = Get-Cluster "SDDC-Cluster1"
$spaceReport = Get-VsanSpaceUsage -Cluster $cluster

[PSCustomObject]@{
    "Before Dedup & Compression (TB)" =
    [math]::Round($spaceReport.PhysicalUsedCapacityGB / 1024, 2)
    "After Dedup & Compression (TB)" = [math]::Round($spaceReport.UsedCapacityGB
    / 1024, 2)
    "Dedup Ratio" = [math]::Round($spaceReport.DedupRatio, 2)
    "Compression Ratio" =
    [math]::Round($spaceReport.CompressionRatio, 2)
    "Overall Savings Ratio" =
    [math]::Round($spaceReport.DedupCompressionRatio, 2)
}
```

Result	Condition	Action
PASS	Savings ratio > 1.5x	Good efficiency
WARN	Savings ratio 1.0-1.5x	Review workload data characteristics
FAIL	Savings ratio < 1.0x	Dedup/compression overhead exceeds savings; consider disabling

6.3 Thin Provisioning

vSAN uses thin provisioning by default for object storage. The logical provisioned space can far exceed physical capacity.

Check Provisioned vs. Used Space

```
$cluster = Get-Cluster "SDDC-Cluster1"
$vms = Get-VM -Location $cluster

$report = foreach ($vm in $vms) {
    $disks = Get-HardDisk -VM $vm
    foreach ($disk in $disks) {
        [PSCustomObject]@{
            VM          = $vm.Name
            Disk         = $disk.Name
            ProvisionedGB = [math]::Round($disk.CapacityGB, 2)
            UsedGB       = [math]::Round(($disk.CapacityGB -
$disk.FreeSpaceGB), 2) # Approximate
            ThinProvisioned = $disk.StorageFormat -eq "Thin"
        }
    }
}

$report | Format-Table -AutoSize
Write-Host "Total Provisioned: $($([math]::Round(($report | Measure-Object -
Property ProvisionedGB -Sum).Sum, 2)) GB"
Write-Host "Total Used:         $($([math]::Round(($report | Measure-Object -
Property UsedGB -Sum).Sum, 2)) GB"
```

6.4 Slack Space Calculation

vSAN reserves slack space for resyncs, maintenance operations, and failure recovery. The formula depends on the cluster size and policy.

Slack Space Formula

$$\text{Slack Space} = \text{Max}(\text{HostCapacity}, 25\% \text{ of RawCapacity})$$

Where:

$\text{HostCapacity} = \text{Total raw capacity} / \text{Number of hosts}$
(i.e., the capacity of the largest single host)

Example Calculation

Cluster: 4 hosts x 10 TB raw each = 40 TB total raw

$\text{HostCapacity} = 40 \text{ TB} / 4 = 10 \text{ TB}$

$25\% \text{ of Raw} = 40 \text{ TB} \times 0.25 = 10 \text{ TB}$

$\text{Slack Space} = \text{Max}(10 \text{ TB}, 10 \text{ TB}) = 10 \text{ TB}$

$\text{Usable Capacity} = 40 \text{ TB} - 10 \text{ TB} = 30 \text{ TB}$
(Before policy overhead)

With FTT=1, RAID-1 mirroring:

$\text{Effective Usable} = 30 \text{ TB} / 2 = 15 \text{ TB}$

6.5 Capacity Thresholds

Used %	Status	Description	Action
0-70%	PASS	Healthy capacity headroom	Normal operations
70-75%	WARN	Approaching capacity limits	Plan expansion or cleanup
75-80%	WARN	vSAN generates a warning alarm	Active capacity management needed
80-90%	FAIL	vSAN throttles new writes	Immediate expansion or VM migration
90-95%	FAIL	Severe performance impact	Emergency capacity action
>95%	FAIL	Risk of data inaccessibility	Emergency: free space immediately

Critical Threshold - 80%: When vSAN capacity reaches 80%, the CLOM (Cluster Level Object Manager) stops performing automatic rebalancing and repairs. New VM deployments may fail. This is a hard operational limit that must never be sustained.

Remediation (High Capacity):

1. Identify largest consumers: `Get-VsanSpaceUsage -Cluster $cluster | Select -ExpandProperty SpaceDetail`
2. Remove orphaned snapshots and stale VM files
3. Storage vMotion cold VMs to alternative datastores (NFS, VMFS)
4. Enable or verify deduplication and compression
5. Plan cluster expansion (add hosts or disks)
6. Review and right-size VMDK allocations

7. Resync Status

7.1 Active Resyncs

Resyncs occur when vSAN needs to rebuild or move components. They can be triggered by host maintenance, disk failures, policy changes, or rebalancing.

Command

```
esxcli vsan debug resync summary
```

Expected Output (No Active Resyncs)

```
Resync Summary:
  Total Objects Resyncing: 0
  Total Bytes To Resync: 0 B
  Total Bytes Resynced: 0 B
  Total Recoveries: 0
  Total Rebalance: 0
  Total Policy Change: 0
  Total Evacuating: 0
```

Expected Output (Active Resyncs)

```
Resync Summary:
  Total Objects Resyncing: 42
  Total Bytes To Resync: 287.35 GB
  Total Bytes Resynced: 143.67 GB
  Total Recoveries: 38
  Total Rebalance: 4
  Total Policy Change: 0
  Total Evacuating: 0
```

Result	Condition	Action
PASS	0 objects resyncing	Cluster fully converged
WARN	< 100 objects, progress advancing	Monitor progress; expected after maintenance
FAIL	> 100 objects or resync stalled	Investigate root cause; check for disk or network issues

7.2 Resync ETA

Monitoring Resync Progress

```
# Real-time resync monitoring
watch -n 10 'esxcli vsan debug resync summary'
```

PowerCLI Resync Details

```
$cluster = Get-Cluster "SDDC-Cluster1"
$vsanHealthSystem = Get-VsanView -Id "VsanVcClusterHealthSystem-vmware-
health-system"
$resyncStatus = $vsanHealthSystem.VsanQuerySyncingVsanObjects(
    $cluster.ExtensionData.MoRef
)

foreach ($obj in $resyncStatus) {
    [PSCustomObject]@{
        UUID           = $obj.Uuid
        BytesToSync     = [math]::Round($obj.BytesToSync / 1GB, 2)
        BytesSynced     = [math]::Round($obj.RecoveryETA, 0)
        Reason          = $obj.Reason
    }
} | Format-Table -AutoSize
```

RVC Resync Dashboard

```
# Provides a continuously updating view of resync progress
vsan.resync_dashboard /localhost/datacenter/computers/cluster
```

7.3 Performance Impact

Active resyncs consume disk IO and network bandwidth. vSAN uses a throttling mechanism to limit impact on production workloads.

Check Resync Throttle Configuration

```
esxcli system settings advanced list -o /LSOM/lsonResyncThrottleEnabled
esxcli system settings advanced list -o /VSAN/ResyncThrottleAdaptive
```

Resync Traffic Limits

Parameter	Default	Impact
ResyncThrottleAdaptive	1 (enabled)	vSAN automatically reduces resync bandwidth when VM IO is detected
ResyncBandwidthCap	0 (unlimited)	Maximum MB/s for resync traffic per host
lsonResyncThrottleEnabled	1	Enables disk-level resync throttling

Performance Impact: During large-scale resyncs (e.g., after host failure), expect 10-30% reduction in VM IO performance. Avoid scheduling additional maintenance operations or large deployments until resyncs complete.

8. vSAN Network Health

8.1 vSAN VMkernel Adapters

Every host in the vSAN cluster must have a dedicated VMkernel adapter tagged for vSAN traffic.

Command

```
esxcli network ip interface list | grep -A5 "vsan"
```

Alternative: Full VMkernel Listing

```
esxcli network ip interface list
```

Expected Output

```
vmk1
  Name: vmk1
  MAC Address: 00:50:56:6a:xx:xx
  Enabled: true
  Portset: DvsPortset-0
  Portgroup: SDDC-DPortGroup-vSAN
  VDS Name: SDDC-Dswitch-Private
  MTU: 9000
  TSO MSS: 65535
  Port ID: 33554435
  Netstack Instance: defaultTcpipStack
  IPv4 Address: 172.16.10.101
  IPv4 Netmask: 255.255.255.0
  IPv4 Broadcast: 172.16.10.255
  IPv6 Enabled: false
  Tags: VSAN
```

Verify vSAN Tag on VMkernel

```
esxcli vsan network list
```

Expected Output

Interface

VmkNic Name: vmk1
IP Protocol: IP
Interface UUID: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx
Agent Group Multicast Address: 224.2.3.4
Agent Group IPv6 Multicast Address: ff19::2:3:4
Agent Group Multicast Port: 23451
Master Group Multicast Address: 224.1.2.3
Master Group IPv6 Multicast Address: ff19::1:2:3
Master Group Multicast Port: 12345
Host Unicast Channel Bound Port: 12321
Multicast Enabled: true
Traffic Type: vsan

Result	Condition	Action
PASS	All hosts have vmk with Tags: VSAN , MTU 9000	Healthy
WARN	MTU mismatch across hosts	Standardize MTU to 9000
FAIL	Host missing vSAN-tagged vmk adapter	Add vSAN VMkernel adapter immediately

8.2 Multicast & Unicast Check

vSAN can operate in multicast mode (legacy) or unicast mode (default in vSAN 7+/VCF 5+). VCF 9 clusters should use unicast.

Check Current Mode

```
esxcli vsan network list | grep "Multicast Enabled"
```

Mode	VCF 9 Status	Notes
Unicast (Multicast Enabled: false)	Recommended	Default for new VCF 9 clusters
Multicast (Multicast Enabled: true)	Legacy	Requires IGMP snooping on physical switches

Verify Unicast Connectivity

```
# From each host, test connectivity to every other host on vSAN network
vmkping -I vmk1 172.16.10.102
vmkping -I vmk1 172.16.10.103
vmkping -I vmk1 172.16.10.104
```

8.3 Jumbo Frame Validation

Jumbo frames (MTU 9000) are required for optimal vSAN performance. End-to-end validation is critical.

Test Jumbo Frames Between Hosts

```
# From ESXi host, test jumbo frame path to each peer
# -s 8972 = 9000 - 20 (IP header) - 8 (ICMP header) = 8972
# -d = set DF (Don't Fragment) bit

vmkping -I vmk1 -s 8972 -d 172.16.10.102
vmkping -I vmk1 -s 8972 -d 172.16.10.103
vmkping -I vmk1 -s 8972 -d 172.16.10.104
```

Expected Output (Healthy)

```
PING 172.16.10.102 (172.16.10.102): 8972 data bytes
8980 bytes from 172.16.10.102: icmp_seq=0 ttl=64 time=0.254 ms
8980 bytes from 172.16.10.102: icmp_seq=1 ttl=64 time=0.198 ms
8980 bytes from 172.16.10.102: icmp_seq=2 ttl=64 time=0.211 ms

--- 172.16.10.102 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.198/0.221/0.254 ms
```

Expected Output (FAILURE)

```
PING 172.16.10.102 (172.16.10.102): 8972 data bytes
sendto() failed (Message too long)
--- 172.16.10.102 ping statistics ---
3 packets transmitted, 0 packets received, 100% packet loss
```

Result	Condition	Action
PASS	0% packet loss on all hosts with 8972 byte payload	Jumbo frames working end-to-end
WARN	Intermittent packet loss	Check physical switch MTU, NIC firmware
FAIL	100% loss or "Message too long"	MTU mismatch in path -- check vmk, vDS, physical switch

Remediation (Jumbo Frame Failure):

1. Verify VMkernel MTU: `esxcli network ip interface list | grep MTU`
2. Verify vDS MTU: vCenter > Networking > vDS > Settings > MTU = 9000
3. Verify physical switch port MTU: interface-level `mtu 9216` (allows for overhead)
4. Verify physical NIC MTU: `esxcli network nic list`
5. Check for any intermediate firewalls or routers that may reduce MTU
6. After fixing, retest: `vmkping -I vmk1 -s 8972 -d <target_ip>`

8.4 Network Partition Detection

A vSAN network partition occurs when hosts lose connectivity to each other, causing the cluster to split into sub-clusters.

Check via Health Service

```
esxcli vsan health cluster list -t "vSAN Cluster Partition"
```

Check via Cluster Membership

```
# Run on EVERY host and compare Sub-Cluster Member UUIDs
esxcli vsan cluster get
```

Detecting Partition via CMMDS

```
# Check CMMDS master node
esxcli vsan cluster get | grep "Local Node State"
```

If multiple hosts report **MASTER**, a partition exists -- only one host should be MASTER.

Result	Condition	Action
PASS	Single MASTER, all hosts in same sub-cluster	No partition
FAIL	Multiple MASTERS or mismatched sub-cluster membership	Active partition -- URGENT

Network Partition Emergency: A vSAN network partition can cause data inaccessibility, VM failures, and split-brain conditions. This is a P1 severity event.

Remediation (Network Partition):

1. Identify which hosts can communicate with each other
2. Check physical network: `esxcli network nic stats get -n vmnic0` for errors/drops
3. Verify VLAN tagging consistency across all hosts
4. Check physical switch logs for spanning tree topology changes or port flapping
5. Test L2 connectivity: `vmkping -I vmk1 <peer_vsan_ip>` from each host

6. If a single host is isolated, restart its networking: `esxcli network ip interface set -i vmk1 -e false && esxcli network ip interface set -i vmk1 -e true`
7. Monitor CLOMD log for partition events: `tail -f /var/log/clomd.log | grep -i partition`

8.5 Witness Host Connectivity (Stretched Cluster)

If a stretched cluster is deployed, the witness host must be reachable from both sites.

Test Witness Connectivity

```
# From preferred site host
vmkping -I vmk1 <witness_vsan_ip>

# From secondary site host
vmkping -I vmk1 <witness_vsan_ip>
```

Verify Witness in Cluster

```
esxcli vsan cluster get | grep -A2 "Witness"
```

Result	Condition	Action
PASS	Witness reachable from both sites, < 200ms RTT	Healthy
WARN	Witness reachable but RTT > 100ms	Investigate WAN link quality
FAIL	Witness unreachable from either site	Immediate investigation -- quorum at risk

9. Performance Health

9.1 IOPS & Latency

vSAN Performance Service (vCenter UI)

Navigate to: **Cluster** > **Monitor** > **vSAN** > **Performance** > **Virtual Machine Consumption**

CLI: Check Per-Host IO Statistics

```
# Real-time IOPS and latency from ESXi
vsish -e get /vmkModules/lsm/disks/<disk_uuid>/info
```

PowerCLI: Query vSAN Performance Data

```
$cluster = Get-Cluster "SDDC-Cluster1"
$vsanPerfSystem = Get-VsanView -Id "VsanPerformanceManager-vsan-performance-
manager"

# Define time range (last 1 hour)
$endTime = Get-Date
$startTime = $endTime.AddHours(-1)

# Query cluster performance
$spec = New-Object VMware.Vsan.Views.VsanPerfQuerySpec
$spec.EntityRefId = "cluster-domclient:*"
$spec.StartTime = $startTime
$spec.EndTime = $endTime

$perfData = $vsanPerfSystem.VsanPerfQueryPerf(@( $spec ),
$cluster.ExtensionData.MoRef)
```

Latency Thresholds

Metric	PASS	WARN	FAIL
Read Latency (average)	< 1 ms	1-5 ms	> 5 ms
Write Latency (average)	< 2 ms	2-10 ms	> 10 ms
Read IOPS	Per baseline	> 20% below baseline	> 50% below baseline

Write IOPS	Per baseline	> 20% below baseline	> 50% below baseline
Read Cache Hit Ratio (OSA)	> 90%	70-90%	< 70%

9.2 Congestion

vSAN congestion values indicate back-pressure in the IO stack. A non-zero congestion value means vSAN is throttling IO.

Command

```
esxcli vsan debug controller list
```

Expected Output (Healthy)

```
Controller: naa.55cd2e414f5356c0
State: HEALTHY
Congestion Value: 0
Congestion Type: None
Outstanding IO: 0
```

Alternative: VSISH Congestion Query

```
# Get per-disk congestion
vsish -e get /vmkModules/lom/disks/<disk_uuid>/info | grep -i congestion
```

Congestion Value	Status	Description
0	PASS	No congestion
1-20	WARN	Mild congestion -- transient during bursts
21-40	WARN	Moderate congestion -- sustained IO pressure
41-60	FAIL	High congestion -- significant IO throttling
61-100	FAIL	Severe congestion -- critical performance impact

Remediation (High Congestion):

1. Identify top IO consumers: vCenter > Cluster > Monitor > vSAN > Performance > VM Consumption
2. Check for active resyncs: `esxcli vsan debug resync summary`
3. Verify no runaway processes: `esxtop` (press `u` for disk view)
4. Check disk health -- degraded disks cause elevated congestion

5. Consider distributing workloads across more hosts
6. If persistent, add capacity disks or hosts to reduce per-disk load

9.3 Outstanding IO

Outstanding IO counts indicate the number of IO operations queued but not yet completed.

Command

```
esxcli vsan debug controller list | grep "Outstanding IO"
```

VSISH Deep Inspection

```
# Per-device outstanding IO  
vsish -e get /vmkModules/lsom/disks/<disk_uuid>/info | grep outstanding
```

Outstanding IO	Status	Description
0-16	PASS	Normal queue depth
17-32	PASS	Moderate load, acceptable
33-64	WARN	Elevated queue depth
> 64	FAIL	Queue saturation -- investigate

9.4 vscsiStats

vscsiStats provides detailed IO profiling for individual VMs and virtual disks.

Enable vscsiStats Collection

```
# List all virtual SCSI handles
vscsiStats -l

# Start collection for a specific handle
vscsiStats -s -w <world_id> -i <handle_id>

# Wait for collection period (e.g., 60 seconds)
sleep 60

# Retrieve statistics
vscsiStats -p all -w <world_id> -i <handle_id>

# Stop collection
vscsiStats -x -w <world_id> -i <handle_id>
```

Output Metrics

Metric	Description
IO Size Histogram	Distribution of IO sizes (4K, 8K, 16K, etc.)
Seek Distance	Sequential vs. random IO pattern
Outstanding IO	Per-VMDK queue depth
Latency Histogram	Distribution of latency values
IO Type	Read/write ratio

Best Practice: Use **vscsiStats** sparingly in production. It adds minor overhead during collection. Collect for 60-120 seconds to get a representative sample, then stop immediately.

9.5 Performance Service

The vSAN Performance Service must be enabled for historical performance data.

Verify Performance Service Status

```
$cluster = Get-Cluster "SDDC-Cluster1"
$vsanConfig = Get-VsanClusterConfiguration -Cluster $cluster
$vsanConfig.PerformanceServiceEnabled
```

Enable Performance Service

```
Set-VsanClusterConfiguration -Cluster $cluster -PerformanceServiceEnabled $true
```

Performance Service Health Check

```
esxcli vsan health cluster list -t "Performance Service"
```

Result	Condition	Action
PASS	Performance service enabled and collecting data	Healthy
WARN	Service enabled but stats database > 80% full	Archive or increase stats DB size
FAIL	Performance service disabled or not functioning	Enable via PowerCLI or vCenter UI

10. Object Health & Compliance

10.1 Object Count

Command

```
esxcli vsan debug object health summary get
```

Expected Output (Healthy)

```
Object Health Summary:
  Total Objects: 2847
  Healthy: 2847
  Objects with Reduced Redundancy: 0
  Inaccessible Objects: 0
  Non-Compliant Objects: 0
  Quorum Not Satisfied: 0
```

PowerCLI Object Count

```
$cluster = Get-Cluster "SDDC-Cluster1"
$vsanHealth = Get-VsanView -Id "VsanVcClusterHealthSystem-vsan-cluster-health-
system"
$objHealth = $vsanHealth.VsanQueryVcClusterHealthSummary(
    $cluster.ExtensionData.MoRef, $null, $null, $true, $null, $null,
    "objectHealth"
)

$objHealth.ObjectHealth.ObjectHealthDetail | ForEach-Object {
    [PSCustomObject]@{
        Category      = $_.NumObjects
        ObjectCount    = $_.ObjHealthState
    }
}
```

10.2 Compliance State

vSAN object compliance verifies that every object meets its assigned storage policy (FTT, stripe width, etc.).

Check Compliance via PowerCLI

```
$cluster = Get-Cluster "SDDC-Cluster1"
$vms = Get-VM -Location $cluster

foreach ($vm in $vms) {
    $spPolicy = Get-SpbmEntityConfiguration -VM $vm
    foreach ($policy in $spPolicy) {
        if ($policy.ComplianceStatus -ne "compliant") {
            [PSCustomObject]@{
                VM           = $vm.Name
                Entity       = $policy.Entity
                Policy       = $policy.StoragePolicy.Name
                Status       = $policy.ComplianceStatus
            }
        }
    }
}
```

Result	Condition	Action
PASS	All objects compliant	No action
WARN	Objects non-compliant but actively rebuilding	Monitor resync progress
FAIL	Objects persistently non-compliant	Investigate capacity or host availability

10.3 Inaccessible Objects

Inaccessible objects have lost quorum -- they cannot be read or written. This is the most critical vSAN health state.

Command

```
esxcli vsan debug object health summary get | grep "Inaccessible"
```

List Inaccessible Objects

```
esxcli vsan debug object list --type=inaccessible
```

Trace Object to VM

```
# Get the object UUID from the inaccessible list, then:  
esxcli vsan debug object list -u <object_uuid>
```

CRITICAL -- Inaccessible Objects: Any inaccessible object represents potential data loss. This is a P1 severity event requiring immediate action. Do NOT perform any maintenance operations until all objects are accessible.

Remediation (Inaccessible Objects):

1. Identify which hosts own the components: `esxcli vsan debug object list -u <uuid>`
2. Check if hosts are offline or partitioned
3. Verify vSAN network connectivity between all hosts
4. If a host is down, restore it immediately
5. If a disk has failed, initiate replacement
6. Check CLOMD logs: `grep -i "inaccessible" /var/log/clomd.log`
7. If persistent, contact VMware Support with object UUIDs and CLOMD logs

10.4 Reduced Redundancy Objects

Objects with reduced redundancy are accessible but have fewer copies than specified by their policy.

Command

```
esxcli vsan debug object health summary get | grep "Reduced Redundancy"
```

Detailed Listing

```
esxcli vsan debug object list --type=reducedRedundancy
```

Result	Condition	Action
PASS	0 objects with reduced redundancy	Full policy compliance
WARN	Objects in reduced redundancy during resync	Expected after host/disk event; monitor resync
FAIL	Persistent reduced redundancy (no active resync)	Investigate CLOM; check capacity/host availability

Remediation (Persistent Reduced Redundancy):

1. Verify resyncs are not stalled: `esxcli vsan debug resync summary`
2. Check available capacity -- CLOM cannot rebuild if < 20% free
3. Check for component limit violations: `esxcli vsan health cluster list -t "Host Component Limit"`
4. Force a repair on the object: from RVC, `vsan.fix_renamed_objects /path/to/cluster`
5. Restart CLOMD if the service is stuck: `/etc/init.d/clomd restart`

11. Stretched Cluster Health

Applicability: This section applies only to environments with vSAN stretched clusters. If your VCF 9 deployment uses standard (non-stretched) clusters, skip to Section 12.

11.1 Preferred & Secondary Site

In a vSAN stretched cluster, hosts are divided into two fault domains (sites) plus a witness host.

Check Site Configuration

```
esxcli vsan cluster get
```

Look for:

```
Preferred Fault Domain: site-a  
Secondary Fault Domain: site-b
```

PowerCLI Site Verification

```
$cluster = Get-Cluster "SDDC-Cluster1"  
$vsanConfig = Get-VsanClusterConfiguration -Cluster $cluster  
  
[PSCustomObject]@{  
    StretchedCluster = $vsanConfig.StretchedClusterEnabled  
    PreferredSite     = $vsanConfig.PreferredFaultDomain.Name  
    SecondarySite     = ($vsanConfig.FaultDomains | Where-Object { $_.Name -ne  
$vsanConfig.PreferredFaultDomain.Name }).Name  
    WitnessHost      = $vsanConfig.WitnessHost.Name  
}
```

Result	Condition	Action
PASS	Both sites have equal host counts, preferred site set correctly	Healthy
WARN	Uneven host distribution between sites	Rebalance hosts if possible
FAIL	One site has no hosts or stretched cluster misconfigured	Reconfigure stretched cluster

11.2 Witness Host

The witness host provides the tiebreaker vote in a stretched cluster. It must be in a third fault domain.

Verify Witness Host

```
# From any cluster host
esxcli vsan cluster get | grep -i witness
```

Witness Host Health Checks

```
# SSH to witness host
ssh root@witness-host.vcf.local

# Verify vSAN is running
esxcli vsan cluster get

# Check witness disk status
esxcli vsan storage list

# Verify network connectivity to both sites
vmkping -I vmk0 <site-a-host-vsan-ip>
vmkping -I vmk0 <site-b-host-vsan-ip>
```

Witness Appliance Resources

Resource	Minimum	Recommended
vCPUs	2	2
Memory	16 GB (< 750 components)	32 GB (> 750 components)
Witness disk cache	5 GB SSD	10 GB SSD
Witness disk capacity	15 GB	30 GB

11.3 Site Affinity Rules

Site affinity rules ensure that specific VMs prefer to run at a particular site during normal operations.

Check Site Affinity via PowerCLI

```
$cluster = Get-Cluster "SDDC-Cluster1"
$rules = Get-DrsRule -Cluster $cluster | Where-Object { $_.Type -eq "VMAffinity"
}
$rules | Format-Table Name, Type, Enabled, VMIds -AutoSize
```

vSAN Storage Policy Site Affinity

```
# Check vSAN storage policies with site affinity
Get-SpbmStoragePolicy | Where-Object {
    $_.AnyOfRuleSets.AnyOfRules.Capability.Name -match "locality"
} | ForEach-Object {
    [PSCustomObject]@{
        PolicyName = $_.Name
        Locality   = ($_.AnyOfRuleSets.AnyOfRules | Where-Object {
            $_.Capability.Name -match "locality"
        }).Value
    }
}
```

11.4 Inter-Site Latency

Test Inter-Site Latency

```
# From a host at Site A to a host at Site B
vmkping -I vmk1 -c 100 <site-b-host-vsan-ip>
```

Latency Requirements

Link	Maximum RTT	PASS	WARN	FAIL
Site A to Site B	5 ms (data)	< 5 ms	5-100 ms	> 100 ms
Either Site to Witness	200 ms	< 100 ms	100-200 ms	> 200 ms
Bandwidth (data sites)	10 Gbps	>= 10 Gbps	1-10 Gbps	< 1 Gbps

Latency Note: vSAN stretched clusters in VCF 9 support up to 5ms RTT between data sites for write operations (synchronous replication). The witness host can tolerate up to 200ms RTT. Exceeding these limits will cause write performance degradation or cluster instability.

12. Fault Domains

12.1 Fault Domain Configuration

Fault domains define failure boundaries. vSAN places components across fault domains to ensure that a single domain failure does not cause data loss.

View Fault Domains

```
$cluster = Get-Cluster "SDDC-Cluster1"
$faultDomains = Get-VsanFaultDomain -Cluster $cluster

foreach ($fd in $faultDomains) {
    [PSCustomObject]@{
        Name      = $fd.Name
        HostCount = ($fd.VMHost | Measure-Object).Count
        Hosts      = ($fd.VMHost.Name -join ", ")
    }
} | Format-Table -AutoSize
```

esxcli Fault Domain Check

```
esxcli vsan cluster get | grep "Fault Domain"
```

12.2 Host Distribution

For optimal fault tolerance, hosts should be evenly distributed across fault domains.

Configuration	PASS	WARN	FAIL
Fault Domain Count	≥ 3 FDs	2 FDs	1 FD or none configured
Hosts per FD	Equal distribution	+/- 1 host variance	Severe imbalance
FTT=1 compliance	≥ 3 FDs	2 FDs (works but no FD-level protection)	1 FD
FTT=2 compliance	≥ 5 FDs	3-4 FDs	< 3 FDs

Example: Optimal 4-Host, 4-FD Configuration

```
Fault Domain: rack-01 -> esx-01.vcf.local
Fault Domain: rack-02 -> esx-02.vcf.local
Fault Domain: rack-03 -> esx-03.vcf.local
Fault Domain: rack-04 -> esx-04.vcf.local
```

12.3 Policy Compliance with Fault Domains

When fault domains are configured, vSAN places mirrors/parity components in different fault domains. The policy must be compatible with the number of fault domains.

Validate Policy vs. Fault Domain Count

```
$cluster = Get-Cluster "SDDC-Cluster1"
$fds = Get-VsanFaultDomain -Cluster $cluster
$fdCount = ($fds | Measure-Object).Count

$policies = Get-SpbmStoragePolicy | Where-Object { $_.Name -like "*vSAN*" }
foreach ($pol in $policies) {
    $ftt = ($pol.AnyOfRuleSets.AnyOfRules | Where-Object {
        $_.Capability.Name -eq "VSAN.hostFailuresToTolerate"
    }).Value

    $requiredFDs = (2 * $ftt) + 1 # For RAID-1

    [PSCustomObject]@{
        Policy          = $pol.Name
        FTT             = $ftt
        RequiredFDs     = $requiredFDs
        AvailableFDs    = $fdCount
        Compliant       = $fdCount -ge $requiredFDs
    }
} | Format-Table -AutoSize
```

Remediation (Insufficient Fault Domains):

1. If the cluster has fewer fault domains than required, either:
 - a. Add more hosts in new fault domains
 - b. Reduce the FTT level in the storage policy
2. To create a new fault domain: vCenter > Cluster > Configure > vSAN > Fault Domains > Add
3. To move a host to a fault domain via PowerCLI:

```
New-VsanFaultDomain -Name "rack-05" -VMHost (Get-VMHost "esx-05.vcf.local")
```

13. vSAN Health Service Detail

13.1 Health Service Status

Verify Health Service is Running

```
esxcli vsan health cluster list -t "vSAN Health Service Up-To-Date"
```

Expected Output

```
Health Test: vSAN Health Service Up-To-Date
Status: green
Description: vSAN Health Service is up-to-date.
Last Run: 2026-03-26T14:00:00Z
```

Check Health Service Database

```
# On VCSA, check health service status
vmon-cli --status vsanhealth
```

Force Health Check Refresh

```
$cluster = Get-Cluster "SDDC-Cluster1"
$healthSystem = Get-VsanView -Id "VsanVcClusterHealthSystem-vsan-cluster-health-
system"
$healthSystem.VsanQueryVcClusterHealthSummary(
    $cluster.ExtensionData.MoRef,
    $null, $null, $true, $null, $null, "defaultView"
)
```

Result	Condition	Action
PASS	Service running, last test < 1 hour ago	Healthy
WARN	Service running but last test > 24 hours ago	Force a refresh
FAIL	Service not running	Restart: <code>vmon-cli --restart vsanhealth</code> on VCSA

13.2 Test Categories

The vSAN Health Service organizes tests into the following categories:

Category	Tests Included	Frequency
Cluster	Partition, CLOMD liveness, disk balance, member health	Every 60 min
Network	VMkernel config, connectivity, MTU, multicast	Every 60 min
Physical Disk	Disk health, metadata, congestion, capacity	Every 60 min
Data	Object health, VM health, compliance	Every 60 min
Limits	Component limits, host failure simulation	Every 60 min
HCL	Controller, driver, firmware, HCL DB age	Every 24 hours
Performance	Performance service status, stats integrity	Every 60 min
Stretched Cluster	Witness, site configuration, inter-site latency	Every 60 min
Encryption	KMS connectivity, key status, rekey status	Every 60 min

13.3 Silenced Alarms

Silenced alarms are health tests that have been muted by an administrator. Excessive silencing can mask real problems.

Check Silenced Alarms

```
$cluster = Get-Cluster "SDDC-Cluster1"
$healthSystem = Get-VsanView -Id "VsanVcClusterHealthSystem-vmware-vmware-health-system"
$silenced =
$healthSystem.VsanHealthGetVsanClusterSilentChecks($cluster.ExtensionData.MoRef)

Write-Host "Silenced checks count: $($silenced.Count)"
$silenced | ForEach-Object { Write-Host "  - $_" }
```

Unsilence All Alarms

```
$healthSystem.VsanHealthSetVsanClusterSilentChecks(
    $cluster.ExtensionData.MoRef,
    $null # Pass null to clear all silenced checks
)
```

Result	Condition	Action
PASS	0 silenced alarms	Full visibility into health
WARN	1-3 silenced alarms	Review each; unsilence if no longer needed
FAIL	> 3 silenced alarms	Audit all silenced checks; likely masking real issues

14. HCL Compliance

14.1 Controller, Driver & Firmware

HCL (Hardware Compatibility List) compliance ensures that storage controllers, drivers, and firmware are certified for vSAN.

Check HCL Status via Health Service

```
esxcli vsan health cluster list -t "vSAN HCL Health"
```

Detailed HCL Query

```
# Controller model
esxcli storage core adapter list

# Driver version
esxcli storage core adapter stats get -a vmhba0

# Firmware version
esxcli storage core adapter list | grep -i firmware
```

PowerCLI HCL Check

```
$cluster = Get-Cluster "SDDC-Cluster1"
$healthSystem = Get-VsanView -Id "VsanVcClusterHealthSystem-vsan-cluster-health-system"
$hclResult = $healthSystem.VsanQueryVcClusterHealthSummary(
    $cluster.ExtensionData.MoRef, $null, $null, $true, $null, $null, "hclInfo"
)

$hclResult.HclInfo | ForEach-Object {
    [PSCustomObject]@{
        Host      = $_.Hostname
        Controller = $_.ControllerName
        Driver     = $_.DriverVersion
        Firmware   = $_.FirmwareVersion
        HCLStatus  = $_.HclStatus
    }
}
```

```
}
} | Format-Table -AutoSize
```

Result	Condition	Action
PASS	All controllers/drivers/firmware on HCL	Fully certified
WARN	HCL database outdated (> 90 days)	Update HCL DB
FAIL	Controller, driver, or firmware NOT on HCL	Update driver/firmware to certified version

Non-HCL Hardware: Running vSAN on non-HCL certified hardware voids VMware support coverage. Disk failures, data loss, and performance issues on non-HCL configurations will not receive VMware engineering assistance. Always maintain HCL compliance.

14.2 HCL Database Update

The HCL database is bundled with vCenter and should be updated regularly.

Check HCL DB Age

```
esxcli vsan health cluster list -t "vSAN HCL DB Up-To-Date"
```

Update HCL DB Online

```
$healthSystem = Get-VsanView -Id "VsanVcClusterHealthSystem-vsan-cluster-health-system"
$healthSystem.VsanVcUploadHclDb($null) # Downloads latest from VMware
```

Update HCL DB Offline (Air-Gapped Environments)

1. Download the latest HCL JSON from [VMware Partner Connect](#)
2. Upload via vCenter UI: **Cluster > Monitor > vSAN > Health > HCL Database > Upload from file**

Or via PowerCLI:

```
$jsonContent = Get-Content -Path "C:\path\to\all.json" -Raw
$healthSystem.VsanVcUploadHclDb($jsonContent)
```

Remediation (Outdated HCL):

1. For internet-connected vCenter: update is automatic; force via PowerCLI if needed
2. For air-gapped environments: download JSON from VMware, upload via UI or PowerCLI
3. Schedule HCL DB updates quarterly at minimum
4. After updating, re-run health checks to verify compliance

15. Port Reference Table

The following ports must be open for vSAN communication between all participating hosts and vCenter.

Port	Protocol	Direction	Service	Description
2233	TCP/UDP	Host <-> Host	vSAN Transport	Primary vSAN data transport (IO traffic)
12321	UDP	Host <-> Host	vSAN Clustering (Unicast)	Unicast agent-to-agent communication
12345	UDP	Host <-> Host	vSAN Clustering (Multicast)	Multicast master group (legacy)
23451	UDP	Host <-> Host	vSAN Clustering (Multicast)	Multicast agent group (legacy)
8080	TCP	Host -> vCenter	vSAN Health	Health check data upload
6500	TCP	Host -> vCenter	vSAN VASA	VASA provider for storage policies
8006	TCP	vCenter -> Host	vSAN VASA	VASA provider callback
443	TCP	Host <-> vCenter	HTTPS	vSphere API, management
902	TCP/UDP	Host <-> vCenter	NFC/Heartbeat	Network file copy, host heartbeat
8010	TCP	Host -> vCenter	vSAN Performance	Performance data upload
2233	TCP	Host <-> Witness	vSAN Transport	Witness traffic (stretched cluster)
12321	UDP	Host <-> Witness	vSAN Clustering	Witness cluster communication
514	UDP	Host -> Syslog	Syslog	vSAN log forwarding
8100	TCP	Host <-> Host	vSAN RDMA	RDMA transport (ESA with RDMA NICs)
8200	TCP	Host <-> Host	vSAN RDMA	RDMA transport secondary

Firewall Validation Script

```
# Verify vSAN firewall rules on ESXi host
esxcli network firewall ruleset list | grep -i vsan

# Check if vSAN ports are open
esxcli network firewall ruleset rule list -r vsanvp
esxcli network firewall ruleset rule list -r vsanEncryption
esxcli network firewall ruleset rule list -r vsanhealth
```

Port Connectivity Test

```
# From each ESXi host, test TCP 2233 to peers
nc -z -w3 172.16.10.102 2233 && echo "OK" || echo "FAIL"
```

```
nc -z -w3 172.16.10.103 2233 && echo "OK" || echo "FAIL"  
nc -z -w3 172.16.10.104 2233 && echo "OK" || echo "FAIL"
```

VCF 9 Note: In VCF 9, vSAN ESA may use RDMA transport on ports 8100/8200 when supported NICs are present. Ensure these ports are open if RDMA is enabled in your environment.

16. Common Issues & Remediation

16.1 Disk Failures

Symptoms

- Health service shows red for "vSAN Disk Health"
- `esxcli vsan storage list` shows **Health Status: Failed** or disk is missing
- SMART errors in `/var/log/vmkernel.log`
- Components become absent or degraded

Diagnostic Commands

```
# Check disk status
esxcli vsan storage list | grep -E "Display Name|Health Status"

# Check SMART data
esxcli vsan debug disk smart get -d naa.<disk_id>

# Check kernel log for disk errors
grep -i "disk error\|I/O error\|medium error" /var/log/vmkernel.log | tail -20

# Check vSAN trace for disk events
grep -i "disk" /var/log/vsantraced.log | tail -20
```

Remediation (Disk Failure - OSA):

Cache Disk Failure (Entire Disk Group Lost):

1. Identify the failed disk and its disk group UUID
2. All capacity disks in the group are now offline
3. Replace the cache disk physically
4. Recreate the disk group: `esxcli vsan storage add -s naa.new_cache -d naa.cap1 -d naa.cap2`
5. vSAN will automatically rebuild components from surviving copies

Capacity Disk Failure:

1. Remove the failed disk from the disk group: `esxcli vsan storage remove -d naa.failed_disk`
2. Physically replace the disk
3. Add the new disk: `esxcli vsan storage add -d naa.new_disk -s naa.cache_disk`
4. Monitor resync: `esxcli vsan debug resync summary`

Remediation (Disk Failure - ESA):

1. In ESA, individual NVMe disk failure does not cause disk group loss

2. Remove the failed disk: `esxcli vsan storage remove -d naa.failed_nvme`
3. Physically replace the disk
4. Add the new disk to the storage pool: `esxcli vsan storage add -d naa.new_nvme`
5. Monitor resync: `esxcli vsan debug resync summary`

16.2 Network Partition

Symptoms

- Health service shows red for "vSAN Cluster Partition"
- Multiple hosts claim MASTER role
- VMs on isolated hosts may become unresponsive
- `esxcli vsan cluster get` shows different Member Counts on different hosts

Diagnostic Commands

```
# Check cluster membership on each host
esxcli vsan cluster get

# Check network connectivity
vmkping -I vmk1 <peer_vsan_ip>

# Check physical NIC status
esxcli network nic stats get -n vmnic2

# Check for CRC errors, drops, overruns
esxcli network nic stats get -n vmnic2 | grep -i "error\|drop\|overrun"

# Check switch port channel status
esxcli network vswitch dvs vmware lacp status get
```

Remediation (Network Partition):

1. Identify the partition boundary -- which hosts can talk to which
2. Check physical connectivity: cables, switch ports, SFP modules
3. Verify VLAN tags: `esxcli network vswitch dvs vmware list`
4. Check for spanning tree issues on physical switches
5. If LACP is in use, verify LACP negotiation: `esxcli network vswitch dvs vmware lacp status get`
6. Test connectivity: `vmkping -I vmk1 -s 8972 -d <peer>`
7. Restart vSAN networking on the isolated host (last resort):
`esxcli vsan network remove -i vmk1`
`esxcli vsan network ip add -i vmk1`
8. If spanning tree is blocking, enable PortFast on access ports

16.3 Resync Storms

Symptoms

- Hundreds or thousands of components resyncing simultaneously
- Severe performance degradation during resync
- High network utilization on vSAN VMkernel interfaces
- VM IO latency spikes

Diagnostic Commands

```
# Check resync volume
esxcli vsan debug resync summary

# Check network utilization
esxtop # Press 'n' for network view, look at vmk1 throughput

# Check throttle settings
esxcli system settings advanced list -o /VSAN/ResyncThrottleAdaptive
```

Remediation (Resync Storm):

1. Verify adaptive throttle is enabled:
`esxcli system settings advanced set -o /VSAN/ResyncThrottleAdaptive -i 1`
2. If needed, manually cap resync bandwidth (MB/s per host):
`esxcli system settings advanced set -o /VSAN/ResyncBandwidthCap -i 500`
3. Avoid performing multiple maintenance operations simultaneously
4. If a single disk failure triggered the storm, it will self-resolve -- monitor progress
5. After the storm subsides, remove any manual bandwidth cap:
`esxcli system settings advanced set -o /VSAN/ResyncBandwidthCap -i 0`

16.4 Performance Degradation

Symptoms

- High latency (> 5ms read, > 10ms write)
- Non-zero congestion values
- Elevated outstanding IO
- User complaints about slow VMs

Diagnostic Commands

```
# Check congestion
esxcli vsan debug controller list

# Check disk latency
esxcli vsan debug disk latency get

# Check for noisy neighbor VMs
esxtop # Press 'v' for VM disk view, sort by DAVG (device average latency)

# Check if resyncs are causing pressure
esxcli vsan debug resync summary

# Check cache tier utilization (OSA only)
vsish -e get /vmkModules/lsm/disks/<cache_uuid>/info | grep -i cache
```

Remediation (Performance Degradation):

1. Identify the bottleneck: disk, network, or compute
2. **Disk bottleneck:** Check SMART, replace aging disks, add capacity
3. **Network bottleneck:** Verify jumbo frames, check for errors/drops, upgrade to 25GbE
4. **Compute bottleneck:** Check CPU ready time on hosts, redistribute VMs with DRS
5. **Noisy neighbor:** Identify high-IO VMs with esxtop, apply IO shares/limits via SIOC
6. **Cache saturation (OSA):** Increase cache tier size or migrate to ESA
7. Review storage policies -- RAID-5/6 has lower write performance than RAID-1
8. Enable vSAN performance service to establish baselines for trending

16.5 CLOM Errors

CLOM (Cluster Level Object Manager) is the vSAN component responsible for object placement and repair. CLOM errors indicate placement failures.

Symptoms

- Objects stuck in non-compliant state
- New VM provisioning fails with "insufficient resources"
- CLOM log shows placement errors

Diagnostic Commands

```
# Check CLOM log for errors
grep -i "error\|fail\|cannot place" /var/log/clomd.log | tail -30

# Check component limits
esxcli vsan health cluster list -t "Host Component Limit"

# Check CLOM status
/etc/init.d/clomd status

# List objects with placement issues
esxcli vsan debug object list --type=nonCompliant
```

Common CLOM Error Messages

Error	Cause	Fix
Not enough fault domains	FTT > available FDs	Add hosts/FDs or reduce FTT
Not enough disk space	Capacity > 80%	Free space or add capacity
Component limit reached	> 9000 components/host	Reduce FTT, consolidate VMs, or add hosts
Cannot place	Combination of above	Analyze specific constraint from log
Disk group offline	Cache disk failure (OSA)	Replace cache disk, recreate DG

Remediation (CLOM Errors):

1. Restart CLOM if hung: `/etc/init.d/clomd restart`
2. Verify sufficient resources: capacity > 20% free, components < 9000/host
3. Check fault domain count meets policy requirements

4. If component limit is reached, reduce FTT on low-priority VMs
5. Review and consolidate storage policies to reduce component count
6. After resolving constraints, CLOM will automatically retry placement

17. CLI Quick Reference Card

Cluster Operations

```
# Get cluster status
esxcli vsan cluster get

# Join a vSAN cluster
esxcli vsan cluster join -c <cluster-uuid>

# Leave a vSAN cluster
esxcli vsan cluster leave

# Restore cluster from backup
esxcli vsan cluster restore -c <cluster-uuid>
```

Health Commands

```
# List all health checks
esxcli vsan health cluster list

# Run a specific health test
esxcli vsan health cluster list -t "<test name>"

# Get health summary
esxcli vsan health cluster get
```

Storage Commands

```
# List all vSAN disks
esxcli vsan storage list

# Add a disk to vSAN (OSA - with cache disk)
esxcli vsan storage add -d naa.<capacity_disk> -s naa.<cache_disk>

# Add a disk to vSAN (ESA)
esxcli vsan storage add -d naa.<nvme_disk>

# Remove a disk from vSAN
esxcli vsan storage remove -d naa.<disk_id>

# Auto-claim disks
esxcli vsan storage automode set -e true
```

Network Commands

```
# List vSAN network interfaces
esxcli vsan network list

# Add a VMkernel interface to vSAN
esxcli vsan network ip add -i vmk1

# Remove a VMkernel interface from vSAN
esxcli vsan network remove -i vmk1

# Test connectivity with jumbo frames
vmkping -I vmk1 -s 8972 -d <target_ip>

# Test standard connectivity
vmkping -I vmk1 <target_ip>
```

Debug Commands

```
# Resync summary
esxcli vsan debug resync summary

# Object health summary
```

```
esxcli vsan debug object health summary get

# List objects by type
esxcli vsan debug object list --type=inaccessible
esxcli vsan debug object list --type=reducedRedundancy
esxcli vsan debug object list --type=nonCompliant

# Disk SMART data
esxcli vsan debug disk smart get -d naa.<disk_id>

# Controller info (congestion, outstanding IO)
esxcli vsan debug controller list

# Space usage details
esxcli vsan debug space show

# Disk latency
esxcli vsan debug disk latency get
```

Policy Commands

```
# List vSAN storage policies applied to a VM's namespace
esxcli vsan policy getdefault

# Set the default vSAN policy
esxcli vsan policy setdefault -c "proportionalCapacity=0" -p
"hostFailuresToTolerate=1"
```

Maintenance Mode

```
# Enter maintenance mode (ensure accessibility)
esxcli system maintenanceMode set -e true -m ensureAccessibility

# Enter maintenance mode (full data migration)
esxcli system maintenanceMode set -e true -m evacuateAllData

# Enter maintenance mode (no data migration)
esxcli system maintenanceMode set -e true -m noAction
```

```
# Exit maintenance mode
esxcli system maintenanceMode set -e false
```

Advanced Settings

```
# List all vSAN advanced settings
esxcli system settings advanced list -o /VSAN

# Common performance-related settings
esxcli system settings advanced list -o /VSAN/ResyncThrottleAdaptive
esxcli system settings advanced list -o /VSAN/ResyncBandwidthCap
esxcli system settings advanced list -o /LSOM/lSomResyncThrottleEnabled

# Set a vSAN advanced parameter
esxcli system settings advanced set -o /VSAN/ResyncThrottleAdaptive -i 1
```

Log Locations

```
# vSAN trace log
/var/log/vsantraced.log

# CLOMD (object placement) log
/var/log/clomd.log

# vSAN management log (on VCSA)
/var/log/vmware/vpxd/vpxd.log # (vSAN operations logged here)

# vSAN health log (on VCSA)
/var/log/vmware/vsanHealth/vsanhealth.log

# VMkernel log (disk errors, IO errors)
/var/log/vmkernel.log

# Syslog (general ESXi system log)
/var/log/syslog.log

# vSAN observer data (if enabled)
/var/log/vsan/observer/
```

18. PowerCLI Quick Reference

Connection & Setup

```
# Install PowerCLI
Install-Module -Name VMware.PowerCLI -Scope CurrentUser -Force

# Connect to vCenter
Connect-VIServer -Server vcsa-01.vcf.local -User administrator@vsphere.local

# Ignore certificate errors (lab only)
Set-PowerCLIConfiguration -InvalidCertificateAction Ignore -Confirm:$false
```

Cluster Information

```
# Get vSAN cluster configuration
$cluster = Get-Cluster "SDDC-Cluster1"
Get-VsanClusterConfiguration -Cluster $cluster

# Get cluster hosts
Get-VMHost -Location $cluster | Select Name, ConnectionState, PowerState

# Get vSAN datastore
Get-Datastore -RelatedObject $cluster | Where-Object { $_.Type -eq "vsan" }
```

Health Checks

```
# Get vSAN health summary
$cluster = Get-Cluster "SDDC-Cluster1"
$healthSystem = Get-VsanView -Id "VsanVcClusterHealthSystem-vsan-cluster-health-system"
$summary = $healthSystem.VsanQueryVcClusterHealthSummary(
    $cluster.ExtensionData.MoRef,
    $null, $null, $true, $null, $null, "defaultView"
)
```

```
# Display overall health
$summary.OverallHealth
$summary.OverallHealthDescription

# Display per-group health
$summary.Groups | ForEach-Object {
    [PSCustomObject]@{
        Group    = $_.GroupName
        Health    = $_.GroupHealth
    }
} | Format-Table -AutoSize
```

Capacity & Space

```
# Get vSAN space usage
$cluster = Get-Cluster "SDDC-Cluster1"
Get-VsanSpaceUsage -Cluster $cluster

# Detailed space breakdown
$space = Get-VsanSpaceUsage -Cluster $cluster
[PSCustomObject]@{
    "Total (TB)"      = [math]::Round($space.TotalCapacityGB / 1024, 2)
    "Used (TB)"       = [math]::Round($space.UsedCapacityGB / 1024, 2)
    "Free (TB)"       = [math]::Round($space.FreeCapacityGB / 1024, 2)
    "Used %"          = [math]::Round(($space.UsedCapacityGB /
$space.TotalCapacityGB) * 100, 1)
    "Dedup Ratio"     = [math]::Round($space.DedupRatio, 2)
    "Compression Ratio" = [math]::Round($space.CompressionRatio, 2)
}
```

Disk Management

```
# List all vSAN disks
$cluster = Get-Cluster "SDDC-Cluster1"
Get-VsanDisk -Cluster $cluster | Select VsanDiskGroup, CanonicalName,
IsCacheDisk, CapacityGB

# Get disk groups per host
```

```

$hosts = Get-VMHost -Location $cluster
foreach ($vmHost in $hosts) {
    $dgs = Get-VsanDiskGroup -VMHost $vmHost
    foreach ($dg in $dgs) {
        [PSCustomObject]@{
            Host      = $vmHost.Name
            DiskGroup = $dg.Name
            DiskCount = ($dg | Get-VsanDisk).Count
        }
    }
} | Format-Table -AutoSize

```

Storage Policies

```

# List all vSAN storage policies
Get-SpbmStoragePolicy | Where-Object { $_.Name -like "*vSAN*" } |
    Select Name, Description

# Check VM compliance
$vms = Get-VM -Location (Get-Cluster "SDDC-Cluster1")
foreach ($vm in $vms) {
    $compliance = Get-SpbmEntityConfiguration -VM $vm
    foreach ($c in $compliance) {
        if ($c.ComplianceStatus -ne "compliant") {
            [PSCustomObject]@{
                VM      = $vm.Name
                Entity   = $c.Entity
                Status   = $c.ComplianceStatus
                Policy    = $c.StoragePolicy.Name
            }
        }
    }
} | Format-Table -AutoSize

# Create a new vSAN storage policy
New-SpbmStoragePolicy -Name "vSAN-FTT1-RAID1" -Description "FTT=1 RAID-1
Mirroring" -RuleSet (
    New-SpbmRuleSet -Name "vSAN" -AllOfRules @(
        New-SpbmRule -Capability (Get-SpbmCapability -Name
"VSAN.hostFailuresToTolerate") -Value 1,
        New-SpbmRule -Capability (Get-SpbmCapability -Name
"VSAN.replicaPreference") -Value "RAID-1 (Mirroring) - Performance"
    )
)

```

```
)
)
```

Fault Domains

```
# List fault domains
$cluster = Get-Cluster "SDDC-Cluster1"
Get-VsanFaultDomain -Cluster $cluster | ForEach-Object {
    [PSCustomObject]@{
        Name     = $_.Name
        Hosts     = ($_.VMHost.Name -join ", ")
    }
} | Format-Table -AutoSize

# Create a new fault domain
New-VsanFaultDomain -Name "rack-05" -VMHost (Get-VMHost "esx-05.vcf.local")

# Remove a fault domain
Remove-VsanFaultDomain -VsanFaultDomain (Get-VsanFaultDomain -Name "rack-05")
```

Stretched Cluster

```
# Get stretched cluster configuration
$cluster = Get-Cluster "SDDC-Cluster1"
$config = Get-VsanClusterConfiguration -Cluster $cluster
[PSCustomObject]@{
    StretchedCluster = $config.StretchedClusterEnabled
    PreferredSite     = $config.PreferredFaultDomain.Name
    WitnessHost       = $config.WitnessHost.Name
}

# Set preferred fault domain
Set-VsanClusterConfiguration -Cluster $cluster -PreferredFaultDomain (
    Get-VsanFaultDomain -Name "site-a"
)
```

Performance Service

```
# Enable performance service
$cluster = Get-Cluster "SDDC-Cluster1"
Set-VsanClusterConfiguration -Cluster $cluster -PerformanceServiceEnabled $true

# Check performance service status
(Get-VsanClusterConfiguration -Cluster $cluster).PerformanceServiceEnabled

# Query performance data
$vsanPerfMgr = Get-VsanView -Id "VsanPerformanceManager-vsan-performance-manager"
$spec = New-Object VMware.Vsan.Views.VsanPerfQuerySpec
$spec.EntityRefId = "cluster-domclient:*"
$spec.StartTime = (Get-Date).AddHours(-1)
$spec.EndTime = Get-Date
$vsanPerfMgr.VsanPerfQueryPerf(@( $spec), $cluster.ExtensionData.MoRef)
```

Maintenance & Operations

```
# Enter maintenance mode (ensure accessibility)
$vmHost = Get-VMHost "esx-01.vcf.local"
Set-VMHost -VMHost $vmHost -State Maintenance -VsanDataMigrationMode
EnsureAccessibility

# Enter maintenance mode (full evacuation)
Set-VMHost -VMHost $vmHost -State Maintenance -VsanDataMigrationMode Full

# Exit maintenance mode
Set-VMHost -VMHost $vmHost -State Connected

# Pre-check maintenance mode (dry run)
$vsanHealthSystem = Get-VsanView -Id "VsanVcClusterHealthSystem-vsan-cluster-
health-system"
$vsanHealthSystem.VsanQueryVcClusterHealthSummary(
    (Get-Cluster "SDDC-Cluster1").ExtensionData.MoRef,
    $null, $null, $true, $null, $null, "maintenanceMode"
)
```

Comprehensive Health Report Script

```

# Full vSAN Health Report
function Get-VsanHealthReport {
    param(
        [string]$ClusterName = "SDDC-Cluster1"
    )

    $cluster = Get-Cluster $ClusterName
    $config = Get-VsanClusterConfiguration -Cluster $cluster
    $space = Get-VsanSpaceUsage -Cluster $cluster
    $hosts = Get-VMHost -Location $cluster

    Write-Host "===== " -ForegroundColor Cyan
    Write-Host " vSAN Health Report: $ClusterName" -ForegroundColor Cyan
    Write-Host " Generated: $(Get-Date)" -ForegroundColor Cyan
    Write-Host "===== " -ForegroundColor Cyan

    # Cluster Config
    Write-Host "`n--- Cluster Configuration ---" -ForegroundColor Yellow
    Write-Host " Hosts: $($hosts.Count)"
    Write-Host " vSAN Enabled: $($config.VsanEnabled)"
    Write-Host " Stretched: $($config.StretchedClusterEnabled)"
    Write-Host " Perf Service: $($config.PerformanceServiceEnabled)"

    # Capacity
    Write-Host "`n--- Capacity ---" -ForegroundColor Yellow
    $usedPct = [math]::Round(($space.UsedCapacityGB / $space.TotalCapacityGB) *
100, 1)
    Write-Host " Total: $([math]::Round($space.TotalCapacityGB / 1024, 2)) TB"
    Write-Host " Used: $([math]::Round($space.UsedCapacityGB / 1024, 2)) TB
($usedPct%)"
    Write-Host " Free: $([math]::Round($space.FreeCapacityGB / 1024, 2)) TB"

    if ($usedPct -gt 80) {
        Write-Host " STATUS: CRITICAL" -ForegroundColor Red
    } elseif ($usedPct -gt 70) {
        Write-Host " STATUS: WARNING" -ForegroundColor Yellow
    } else {
        Write-Host " STATUS: HEALTHY" -ForegroundColor Green
    }

    # Host Status
    Write-Host "`n--- Host Status ---" -ForegroundColor Yellow
    foreach ($h in $hosts) {
        $state = $h.ConnectionState

```

```

        $color = if ($state -eq "Connected") { "Green" } else { "Red" }
        Write-Host "    $($h.Name): $state" -ForegroundColor $color
    }

# Disk Health
Write-Host "`n--- Disk Health ---" -ForegroundColor Yellow
$d disks = Get-VsanDisk -Cluster $cluster
Write-Host "    Total Disks: $($disks.Count)"

# Policy Compliance
Write-Host "`n--- Policy Compliance ---" -ForegroundColor Yellow
$vm = Get-VM -Location $cluster
$nonCompliant = 0
foreach ($vm in $vms) {
    $compliance = Get-SpbmEntityConfiguration -VM $vm -ErrorAction
SilentlyContinue
    $nonCompliant += ($compliance | Where-Object { $_.ComplianceStatus -ne
"compliant" }).Count
}

if ($nonCompliant -eq 0) {
    Write-Host "    All VMs compliant" -ForegroundColor Green
} else {
    Write-Host "    Non-compliant entities: $nonCompliant" -ForegroundColor Red
}

Write-Host "`n===== " -ForegroundColor Cyan
Write-Host "    Report Complete" -ForegroundColor Cyan
Write-Host "===== " -ForegroundColor Cyan
}

# Execute the report
Get-VsanHealthReport -ClusterName "SDDC-Cluster1"

```

vSAN Health Check Handbook

Version 1.0 -- March 2026

Copyright 2026 Virtual Control LLC. All rights reserved.

This document is for internal use only and may not be distributed without written permission.

VMware, vSAN, vSphere, vCenter, ESXi, and VCF are registered trademarks of Broadcom Inc.