



VMware PowerCLI 11 Storage Module Reference Poster for vSAN



vSAN Cmdlets

Cluster

Get-VsanSpaceUsage
Get-VsanClusterConfiguration
Set-VsanClusterConfiguration
Start-VsanClusterRebalance
Stop-VsanClusterRebalance
Update-VsanHclDatabase
Get-VsanRuntimeInfo
Get-VsanStat
Get-VsanView
Get-VsanEnterMaintenanceModeReport
Get-VsanEvacuationPlan

Disk Group

Get-VsanDisk
Get-VsanDiskGroup
New-VsanDisk
New-VsanDiskGroup
Remove-VsanDisk
Remove-VsanDiskGroup
Start-VsanClusterDiskUpdate

Testing

Test-VsanClusterHealth
Test-VsanNetworkPerformance
Test-VsanStoragePerformance
Test-VsanVMCreation

vSAN iSCSI Service

Get-VsanIscsiInitiatorGroup
Get-VsanIscsiInitiatorGroupTargetAssociation
New-VsanIscsiInitiatorGroup
New-VsanIscsiInitiatorGroupTargetAssociation
Remove-VsanIscsiInitiatorGroup
Remove-VsanIscsiInitiatorGroupTargetAssociation
Set-VsanIscsiInitiatorGroup

Get-VsanIscsiLun
New-VsanIscsiLun
Remove-VsanIscsiLun
Set-VsanIscsiLun
Get-VsanIscsiTarget
New-VsanIscsiTarget
Remove-VsanIscsiTarget
Set-VsanIscsiTarget

Core Storage & Policy Cmdlets

Export-SpbmStoragePolicy
Get-SpbmCapability
Get-SpbmCompatibleStorage
Get-SpbmEntityConfiguration
Get-SpbmFaultDomain
Get-SpbmPointInTimeReplica
Get-SpbmReplicationGroup
Get-SpbmReplicationPair
Get-SpbmStoragePolicy
Get-SpbmView
Get-VsanView
Import-SpbmStoragePolicy
New-SpbmRule
New-SpbmRuleSet
New-SpbmStoragePolicy
Remove-SpbmStoragePolicy
Set-SpbmEntityConfiguration
Set-SpbmStoragePolicy
Start-SpbmReplicationFailover
Start-SpbmReplicationPrepareFailover
Start-SpbmReplicationPromote
Start-SpbmReplicationReverse
Start-SpbmReplicationTestFailover
Stop-SpbmReplicationTestFailover
Sync-SpbmReplicationGroup

Core

Get-NfsUser
New-NfsUser
Remove-NfsUser
Set-NfsUser
Get-VASIOFilter
Set-VASIOFilter
Get-VasaProvider
New-VasaProvider
Remove-VasaProvider

First Class Disks

Copy-VDisk
Get-VDisk
Move-VDisk
New-VDisk
Remove-VDisk
Set-VDisk

<h3>PowerCLI Info</h3> <p>What is PowerCLI? VMware PowerCLI is a powerful, easy-to-use scripting interface to manage the vSphere platform. Administrators can leverage more than 360 cmdlets to simplify everyday tasks.</p>	<h3>Getting Started</h3> <p>VMware PowerCLI frequently asked questions (FAQs) link: http://communities.vmware.com/docs/DOC-13700 To find out what cmdlets are available: <code>Get-VICmdlet</code> To show documentation for all available cmdlets: <code>Get-PowerCLIDocumentation</code> For help with a cmdlet: <code>Get-Help cmdlet-name -Full</code></p> <h4>How to Connect to vCenter Server or ESXi</h4> <p>To connect to a VMware vCenter server. Start a new session or reestablish a previous session with a VMware vCenter server. <code>\$srv = Connect-VIServer -Server 192.168.0.10 -User Admin -Password Pass01</code> To disconnect from the connected vSphere server: <code>Disconnect-VIServer -Server \$srv -Confirm:\$false</code></p>	<h3>Quick Helpful Commands</h3> <p>Top 5 VM memory allocation: <code>Get-VM Sort-Object -Property MemoryGB -Descending Select -First 5</code> Total memory available on all ESXi: <code>Get-VMHost Measure-Object -Property MemoryTotalGB -Sum Select -ExpandProperty Sum</code> Report on NumCPU and number of VM: <code>Get-VM Group-Object -Property NumCpu Select @(N="NumCpu";E={\$_.Name}).@(N="Number of VM";E={\$_.Count})</code> Most API methods require parameters. You create those with <code>New-Object</code>: <code>\$vm = Get-VM -Name MyVM</code> <code>\$spec = New-Object VMware.Vim.VirtualMachineConfigSpec</code> <code>\$spec.LatencySensitivity = New-Object VMware.Vim.LatencySensitivity</code> <code>\$spec.LatencySensitivity.Level = [VMware.Vim.LatencySensitivityLevel]:high</code> <code>\$vm.ExtensionData.ReconfigVM(\$spec)</code></p>	<h3>Host Static Routing</h3> <p>To list all the host routes: <code>Get-VMHostRoute -VMHost "hostname"</code> To create a new route on a host: <code>New-VMHostRoute -VMHost "hostname" -Destination X.X.X.X -Gateway Y.Y.Y.Y -PrefixLength ZZ -Confirm:\$false</code> To remove a route on a host: <code>Remove-VMHostRoute (Get-VMHostRoute -VMHost "hostname") Where {\$_.Destination -eq "Y.Y.Y.Y"} -Confirm:\$false</code> To list all the host routes for all hosts in a vSAN Fault Domain: <code>Get-Cluster -Name "Cluster" Get-VsanFaultDomain -Name "SiteA" Get-VMHost Get-VMHostRoute</code> To create a new route on all hosts in a vSAN Fault Domain: <code>Get-Cluster -Name "Cluster" Get-VsanFaultDomain -Name "SiteA" Get-VMHost New-VMHostRoute -Destination X.X.X.X -Gateway Y.Y.Y.Y -PrefixLength ZZ</code> To remove a specific route on all hosts in a vSAN Fault Domain: <code>Get-Cluster -Name "Cluster" Get-VsanFaultDomain -Name "SiteA" Get-VMHost Get-VMHostRoute Where-Object {\$_.Destination -eq "Y.Y.Y.Y"} Remove-VMHostRoute</code></p>
<h3>How to Store vCenter Credentials</h3> <p>Method to not have to input credentials every time you connect to vCenter: <code>New-VICredentialStoreItem -Host vcsa.lab.local -User root -Password "VMware!"</code> To remove credentials: <code>Remove-VICredentialStoreItem -Host vcsa.lab.local -Confirm</code></p>	<h3>Enabling / Disabling vSAN</h3> <p>Method to create a cluster, enabling vSAN upon creation: <code>New-Cluster -Name "Cluster" -Location "Datacenter" -VsanEnabled</code> Method to enable vSAN on an existing cluster: <code>Set-Cluster -Cluster "Cluster" -Location "Datacenter" -VsanEnabled \$true</code> Method to disable vSAN on an existing cluster: <code>Set-Cluster -Cluster "Cluster" -Location "Datacenter" -VsanEnabled \$false</code></p>	<h3>vSAN Datastore Policy</h3> <p>Get the assigned Storage Policy for a vSAN Datastore: <code>Get-Cluster -Name "Cluster" Get-Datastore Where-Object {\$_.Type -eq "vSAN"} Get-SpbmEntityConfiguration</code> Set the assigned Storage Policy for a vSAN Datastore: <code>Get-Cluster -Name "Cluster" Get-Datastore Where-Object {\$_.Type -eq "vSAN"} Get-SpbmEntityConfiguration Set-SpbmEntityConfiguration -StoragePolicy (Get-SpbmStoragePolicy -Name "RAID5")</code></p>	<h3>vSAN Disk Group Operations</h3> <p>To list all of the vSAN disks groups for an ESXi host: <code>Get-VsanDiskGroup -VMHost (Get-VMHost Select -First 1)</code> To list all of the disks within a vSAN disk group for an ESXi host: <code>\$vmhost = Get-VMHost Select -First 1</code> <code>\$vsanDiskGroup = Get-VsanDiskGroup -VMHost \$vmhost</code> <code>Get-VsanDisk -VsanDiskGroup \$vsanDiskGroup</code> To create a new vSAN disk group for an ESXi host: <code>\$vmhost = Get-VMHost Select -First 1</code> <code>\$ssd = Get-ScsiLun -VMHost \$vmhost Where {\$_.extensionData.ssd -eq \$true -and \$_.ExtensionData.LocalDisk -eq \$true}</code> <code>\$datadisk = Get-ScsiLun -VMHost \$vmhost Where {\$_.extensionData.ssd -eq \$false -and \$_.ExtensionData.LocalDisk -eq \$true}</code> <code>New-VsanDiskGroup -VMHost \$vmhost -SsdCanonicalName \$ssd.CanonicalName -DataDiskCanonicalName \$datadisk.CanonicalName</code> To remove a disk from a vSAN disk group: <code>\$vmhost = Get-VMHost Select -First 1</code> <code>\$vsanDiskGroup = Get-VsanDiskGroup -VMHost \$vmhost</code> <code>\$vsanDisk = Get-VsanDisk -VsanDiskGroup \$vsanDiskGroup Select -First 1</code> <code>Remove-VsanDisk -VsanDisk \$vsanDisk</code></p>
<h3>Deduplication & Compression</h3> <p>Including DD+C when creating a cluster: <code>New-Cluster -Name "Cluster" -Location "Datacenter" -VsanEnabled \$true -SpaceEfficiencyEnabled \$true</code> Method to enable/disable DD+C on an existing vSAN cluster: <code>Set-Cluster -Cluster "Cluster" -Location "Datacenter" -VsanEnabled \$true/\$false -AllowReducedRedundancy \$true</code></p>	<h3>TRIM/UNMAP Support - vSAN 6.7U1 or higher</h3> <p>To enable TRIM/UNMAP: <code>Get-VsanClusterConfiguration -Cluster (Get-Cluster -Name "Cluster") Set-VsanClusterConfiguration -guestTrimUnmap \$true</code> To disable TRIM/UNMAP: <code>Get-VsanClusterConfiguration -Cluster (Get-Cluster -Name "Cluster") Set-VsanClusterConfiguration -guestTrimUnmap \$false</code></p>	<h3>Maintenance Mode</h3> <p>To place a vSAN Host in Maintenance Mode <code>Set-VMHost -VMHost "hostname" -State Maintenance -VsanDataMigrationMode EnsureAccessibility/Full/NoDataMigration</code> To take a vSAN Host out of Maintenance Mode <code>Set-VMHost -VMHost "hostname" -State Connected</code></p>	<h3>Free vSAN Datastore Capacity</h3> <p>Free Raw Capacity: <code>(Get-VsanSpaceUsage -Cluster "Cluster").FreeSpaceGB</code> Free Capacity Based on a Specific Storage Policy: <code>(Get-VsanSpaceUsage -StoragePolicy (Get-SpbmStoragePolicy -Name "RAID5") -Cluster "Cluster").VsanWhatIfCapacity.FreeWhatIfCapacityGB</code></p>
<h3>vSAN Encryption</h3> <p>Enabling vSAN Encryption on a cluster: <code>Start-VsanEncryptionConfiguration -Cluster "Cluster" -EncryptionEnabled \$true -KmsCluster "KMSCluster" -AllowReducedRedundancy \$true</code> Disabling vSAN Encryption on a cluster: <code>Start-VsanEncryptionConfiguration -Cluster "Cluster" -EncryptionEnabled \$false -AllowReducedRedundancy \$true</code> Perform a Deep Rekey: <code>Start-VsanEncryptionConfiguration -Cluster "Cluster" -DeepRekey -AllowReducedRedundancy \$true</code> Perform a Shallow Rekey: <code>Start-VsanEncryptionConfiguration -Cluster "Cluster" -ShallowRekey</code> Changing the KMS Server used with vSAN: <code>Start-VsanEncryptionConfiguration -Cluster "Cluster" -KmsCluster "KMSCluster2"</code></p>	<h3>Install a new VIB to a vSAN Host</h3> <p>Installing a vSphere installable Bundle (VIB) on a vSphere/vSAN Host <code>\$VIBPATH = "vms/volumes/NFS01/VIB/cisco/iscsi-fnic_1.6.0.24-10EM.600.0.0.2494585.vib"</code> <code>Get-VMHost -Name "hostname" Get-ESXCLI -v2 % { (\$_.software.vib.install.invoke(@{viburi=\$VIBPath;nosigcheck=\$true}))}</code></p>	<h3>Repair Objects Immediately</h3> <p>Repair Objects Immediately: <code>\$Cluster = Get-Cluster -Name "Cluster"</code> <code>\$Vchs = Get-VsanView -Id VsanVcClusterHealthSystem-vsan-cluster-health-system</code> <code>\$Vchs.VsanHealthRepairClusterObjectsImmediate(\$Cluster.ExtensionData.MoRef,\$null)</code></p>	<h3>Update the vSAN HCL Database</h3> <p>From VMware.com: <code>Update-VsanHclDatabase</code> From an offline file: <code>Update-VsanHclDatabase -Filepath "path/to/file/all.json"</code></p>
<h3>vSAN Traffic</h3> <p>Tagging/Untagging the vmk1 VMkernel interface for vSAN Traffic on a host: <code>Get-VMHostNetworkAdapter -Name "vmk1" -VMHost "hostname" Set-VMHostNetworkAdapter -VsanTrafficEnabled \$true/\$false -Confirm \$true</code> Tagging the vmk1 VMkernel interface for vSAN Traffic on every host in a cluster: <code>Get-Cluster -Name "Cluster" Get-VMHost Get-VMHostNetworkAdapter -Name "vmk1" Set-VMHostNetworkAdapter -VsanTrafficEnabled \$true/\$false -Confirm \$false</code></p>	<h3>Create a vSAN Storage Policy</h3> <p>Assign policy rules to variables <code>\$VsanPFTT = Get-SpbmCapability -Name "Vsan.hostFailuresToTolerate"</code> <code>\$VsanFTM = Get-SpbmCapability -Name "Vsan.replicaPreference"</code> <code>\$VsanSFTT = Get-SpbmCapability -Name "Vsan.subFailuresToTolerate"</code> <code>\$VsanOSB = Get-SpbmCapability -Name "Vsan.proportionalCapacity"</code> <code>\$VsanCapRes = Get-SpbmCapability -Name "Vsan.cacheReservation"</code> <code>\$VsanCheckSumOff = Get-SpbmCapability -Name "Vsan.checksumDisabled"</code> <code>\$VsanIopsLimit = Get-SpbmCapability -Name "Vsan.iopsLimit"</code> <code>\$VsanLocality = Get-SpbmCapability -Name "Vsan.locality"</code> <code>\$VsanStripeWidth = Get-SpbmCapability -Name "Vsan.stripeWidth"</code> <code>\$VsanForceProvision = Get-SpbmCapability -Name "Vsan.forceProvisioning"</code> <code>\$FTM = Get-SpbmCapability -Name "Vsan.replicaPreference" Select-Object AllowedValue</code> <code>\$Locality = Get-SpbmCapability -Name "Vsan.locality" Select-Object AllowedValue</code> Create a New Mirroring Policy <code>New-SpbmStoragePolicy -Name "RAID1" -Description "Mirroring Policy" -AnyOfRuleSets (New-SpbmRuleSet (New-SpbmRule -Capability \$VsanPFTT -Value 1),(New-SpbmRule -Capability \$VsanFTM -Value \$FTM.AllowedValue(0)))</code> Create a New RAID5 (Erasure Coding) Policy <code>New-SpbmStoragePolicy -Name "RAID5" -Description "Erasure Coding Policy" -AnyOfRuleSets (New-SpbmRuleSet (New-SpbmRule -Capability \$VsanPFTT -Value 1),(New-SpbmRule -Capability \$VsanFTM -Value \$FTM.AllowedValue(0)))</code></p>	<h3>Assign a Storage Policy to a VM</h3> <p># Get the working SPBM Policy <code>\$Policy = Get-SpbmStoragePolicy -Name "RAID5"</code> # Get the VM to apply the policy to <code>\$VM = Get-VM -Name "APP1"</code> # Get the current SPBM configuration & replace the policy assigned <code>Set-SpbmEntityConfiguration -Configuration (Get-SpbmEntityConfiguration \$VM) -StoragePolicy \$Policy</code> # Get the current SPBM configuration for each hard disk & replace the policy assigned <code>Set-SpbmEntityConfiguration -Configuration (Get-SpbmEntityConfiguration -HardDisk (Get-HardDisk -VM \$VM)) -StoragePolicy \$Policy</code></p>	<h3>Per-VM Storage Policy Compliance</h3> <p># Get all of the VM's <code>\$VMs = Get-Cluster -Name "Cluster" Get-VM Sort-Object Name</code> <code>ForEach (\$VM in \$VMs) {</code> <code>Get-SpbmEntityConfiguration -VM \$VM</code> <code>\$HardDisks = Get-HardDisk -VM \$VM</code> <code>ForEach (\$HardDisk in \$HardDisks) {</code> <code>Get-SpbmEntityConfiguration -HardDisk \$HardDisk</code> } } In a one-liner: <code>Get-Cluster -Name "Cluster" Get-VM Sort-Object Name % {Get-SpbmEntityConfiguration -VM \$_.Name; Get-HardDisk -VM \$_.Name % {Get-SpbmEntityConfiguration -VM \$_.Name}}</code></p>
<h3>PowerCLI Community</h3> 	<h3>PowerCLI Sites</h3> <p>Official PowerCLI Blog - http://blogs.vmware.com/PowerCLI/ Documentation - https://code.vmware.com/web/tool/vmware-powercli VMware PowerCLI Community - http://vmware.com/go/powercli Twitter - https://twitter.com/PowerCLI LinkedIn - http://www.linkedin.com/groups/PowerCLI-Users-162324 Facebook - https://www.facebook.com/vmwarepowercli Slack - https://code.vmware.com/web/code/join PowerCLI Cookbook for vSAN - https://vmware.com/go/powercli4vsan</p>	<h3>PowerCLI Community Repository</h3> <p>The VMware PowerCLI Community Repository is a central location where anyone can contribute their own resources. The repository features a Principles of Operations document to help guide contributors along the process of using, creating, and updating resources which are contained within the repository. Link: https://github.com/vmware/PowerCLI-Example-Scripts</p> <p>Featured Modules:</p> <ul style="list-style-type: none"> - Horizon View Helper - VM Encryption - vCenter High Availability <p>Featured Scripts:</p> <ul style="list-style-type: none"> - Home Lab Deployment - Affinity Rule Creation - Lun Path Report 	<h3>PowerCLI Cookbook for vSAN</h3> 