



在NetApp存储上部署Microsoft Hyper-V

NetApp Solutions

NetApp
September 26, 2024

目录

在NetApp存储上部署Microsoft Hyper-V	1
在NetApp存储上部署Microsoft Hyper-V	1
在NetApp存储上部署Microsoft Hyper-V: 前提条件	1
在NetApp存储上部署Microsoft Hyper-V: 注意事项	6
在NetApp存储上部署Microsoft Hyper-V: 数据保护	12
在NetApp存储上部署Microsoft Hyper-V: 总结	18
在NetApp存储上部署Microsoft Hyper-V: 迁移脚本	18

在NetApp存储上部署Microsoft Hyper-V

在NetApp存储上部署Microsoft Hyper-V

Windows Server平台使用Hyper-V角色提供虚拟化技术。Hyper-V是Windows Server提供的许多可选角色之一。

概述

通过Hyper-V角色，我们可以使用Windows Server中内置的虚拟化技术创建和管理虚拟化计算环境。Hyper-V技术可对硬件进行虚拟化、从而提供一个环境、在此环境中、您可以在一台物理计算机上同时运行多个操作系统。通过Hyper-V、您可以创建和管理虚拟机及其资源。每个虚拟机都是一个独立的虚拟化计算机系统、可以运行自己的操作系统。Hyper-V提供的基础架构可对应用程序和工作负载进行虚拟化，这些应用程序和工作负载支持旨在提高效率 and 降低成本的各种业务目标，这是VMware®vSphere的完美替代方案，特别是当企业希望在当前市场条件下共存多个虚拟机管理程序时。

audience

本文档介绍了使用NetApp ONTAP系统的Hyper-V集群配置的架构和部署过程。本文档的目标受众包括销售工程师、现场顾问、专业服务人员、IT经理、合作伙伴工程师、以及希望将Hyper-V部署为主虚拟机管理程序或备用虚拟机管理程序的客户。

架构

本文档中介绍的架构专门包括Microsoft®Windows Server®2022和Hyper-V®虚拟化。NetApp强烈建议在每次部署中都使用虚拟化软件和基础架构管理软件。此配置会根据每个组件的最佳实践来实现可靠的企业级基础架构。

使用情形摘要

本文档介绍了在使用NetApp纯闪存FAS和ASA阵列型号的Microsoft Windows Server 2022上设置Hyper-V集群以使其作为工作负载以最佳方式运行的部署过程和最佳实践。服务器操作系统/虚拟机管理程序为Microsoft Windows Server 2022。本指南涵盖了通过存储区域网络(SAN)和网络连接存储(NAS)协议提供数据的NetApp存储系统。

在NetApp存储上部署Microsoft Hyper-V：前提条件

本主题介绍了使用ONTAP存储系统配置和部署双节点故障转移集群和集群模式Hyper-V虚拟机的步骤。

部署操作步骤的前提条件

- 所有硬件都必须经过所运行的Windows Server版本的认证、并且整个故障转移集群解决方案必须通过验证配置向导中的所有测试
- Hyper-V节点已加入域控制器(建议)、并且彼此之间具有适当的连接。
- 每个Hyper-V节点的配置应完全相同。
- 在每个Hyper-V服务器上配置网络适配器和指定虚拟交换机、用于隔离管理、iSCSI、SMB和实时迁移的流

量。

- 故障转移集群功能已在每个Hyper-V服务器上启用。
- SMB共享或LV用作共享存储、用于存储Hyper-V集群中的虚拟机及其磁盘。
- 存储不应在不同集群之间共享。为每个集群规划一个或多个CSV/CIFS共享。
- 如果SMB共享用作共享存储、则必须配置SMB共享上的权限、以授予对集群中所有Hyper-V节点的计算机帐户的访问权限。

有关详细信息，请参见

- ["Windows Server上Hyper-V的系统要求"](#)
- ["验证故障转移集群的硬件"](#)
- ["部署Hyper-V集群"](#)

安装Windows功能

以下步骤介绍如何安装所需的Windows Server 2022功能。

所有主机

1. 在所有指定节点上准备Windows OS 2022以及必要的更新和设备驱动程序。
2. 使用安装期间输入的管理员密码登录到每个Hyper-V节点。
3. 右键单击任务栏中的PowerShell图标并选择以启动PowerShell提示符 Run as Administrator。
4. 添加Hyper-V、MPIO和集群功能。

```
Add-WindowsFeature Hyper-V, Failover-Clustering, Multipath-IO `-  
IncludeManagementTools -Restart
```

正在配置网络

正确的网络规划是实现容错部署的关键。为每种类型的流量设置不同的物理网络适配器是故障转移集群的标准建议。由于能够添加虚拟网络适配器、交换机嵌入式绑定(组)和引入Hyper-V QoS等功能、因此可以在更少的物理适配器上精简网络流量。在设计网络配置时应考虑服务质量、冗余和流量隔离。将VLAN等网络隔离技术与流量隔离技术相结合、可以为流量和服务质量提供冗余、从而提高和提高存储流量性能的一致性。

建议使用多个逻辑和/或物理网络隔离和隔离特定工作负载。通常分为多个网段的典型网络流量示例如下：

- iSCSI存储网络。
- CSV (集群共享卷)或Heartbeat网络。
- 实时迁移
- VM网络
- 管理网络

注意：如果将iSCSI与专用NIC结合使用，则不建议使用任何绑定解决方案，而应使用MPIO/DSM。

注意：Hyper-V网络最佳实践也不建议在Hyper-V环境中对SMB 3.0存储网络使用NIC绑定。

有关追加信息、请参见 ["在Windows Server中规划Hyper-V网络"](#)

确定Hyper-V的存储设计

Hyper-V支持使用NAS (SMB3.0)和块存储(iSCSI/FC)作为虚拟机的后备存储。NetApp支持SMB3.0、iSCSI和FC协议、这些协议可用作VM的本机存储—使用iSCSI/FC和SMB3的集群共享卷(CSV)。对于需要直接访问存储的工作负载、客户还可以使用SMB3和iSCSI作为子系统连接的存储选项。ONTAP为统一存储(全闪存阵列)提供了灵活的选项—适用于需要混合协议访问和SAN优化存储(全SAN阵列)的工作负载(仅适用于SAN配置)。

SMB3与iSCSI/FC的选择是受当前现有基础架构的驱动、SMB3/iSCSI允许客户使用现有网络基础架构。对于已有FC基础架构的客户、可以利用该基础架构并将存储呈现为基于FC的集群共享卷。

*注意：*运行ONTAP软件的NetApp存储控制器可在Hyper-V环境中支持以下工作负载：

- 在持续可用的SMB 3.0共享上托管的VM
- 托管在iSCSI或FC上运行的集群共享卷(CSV) LUN上的VM
- 子系统内存、并将直通磁盘传递到子系统虚拟机

注意：核心ONTAP功能，如精简配置、重复数据删除、数据压缩、数据缩减、灵活克隆、无论使用何种平台或操作系统、快照和复制都可以在后台无缝工作、并为Hyper-V工作负载提供显著价值。这些功能的默认设置最适合Windows Server和Hyper-V

注意：如果VM有多个可用路径、并且安装并配置了多路径I/O功能、则使用子系统内启动程序的子VM上支持MPIO。

注意：ONTAP支持所有主要的行业标准客户端协议：NFS、SMB、FC、FCoE、iSCSI、NVMe/FC和S3。但是、Microsoft不支持NVMe/FC和NVMe/TCP。

安装NetApp Windows iSCSI主机实用程序

下一节介绍如何在无人看管的情况下安装NetApp Windows iSCSI主机实用程序。有关安装的详细信息、请参见 ["安装Windows Unified Host Utilities 7.2 \(或支持的最新版本\)"](#)

所有主机

1. 下载 ["Windows iSCSI Host Utilities"](#)
2. 取消阻止下载的文件。

```
Unblock-file ~\Downloads\netapp_windows_host_utilities_7.2_x64.msi
```

3. 安装Host Utilities。

```
~\Downloads\netapp_windows_host_utilities_7.2_x64.msi /qn  
"MULTIPATHING=1"
```

注：在此过程中，系统将重新启动。

正在配置Windows主机iSCSI启动程序

以下步骤介绍如何配置内置的Microsoft iSCSI启动程序。

所有主机

1. 右键单击任务栏中的PowerShell图标并选择以管理员身份运行、以启动PowerShell提示符。
2. 将iSCSI服务配置为自动启动。

```
Set-Service -Name MSiSCSI -StartupType Automatic
```

3. 启动iSCSI服务。

```
Start-Service -Name MSiSCSI
```

4. 配置MPIO以声明任何iSCSI设备。

```
Enable-MSDSMAutomaticClaim -BusType iSCSI
```

5. 将所有新声明的设备的默认负载平衡策略设置为轮叫。

```
Set-MSDSMGlobalDefaultLoadBalancePolicy -Policy RR
```

6. 为每个控制器配置iSCSI目标。

```
New-IscsiTargetPortal -TargetPortalAddress <<iscsia_lif01_ip>>  
-InitiatorPortalAddress <iscsia_ipaddress>  
  
New-IscsiTargetPortal -TargetPortalAddress <<iscsib_lif01_ip>>  
-InitiatorPortalAddress <iscsib_ipaddress>  
  
New-IscsiTargetPortal -TargetPortalAddress <<iscsia_lif02_ip>>  
-InitiatorPortalAddress <iscsia_ipaddress>  
  
New-IscsiTargetPortal -TargetPortalAddress <<iscsib_lif02_ip>>  
-InitiatorPortalAddress <iscsib_ipaddress>
```

7. 将每个iSCSI网络的会话连接到每个目标。

```
Get-IscsiTarget | Connect-IscsiTarget -IsPersistent $true
-IsMultipathEnabled $true -InitiatorPortAddress <iscsia_ipaddress>

Get-IscsiTarget | Connect-IscsiTarget -IsPersistent $true
-IsMultipathEnabled $true -InitiatorPortAddress <iscsib_ipaddress>
```

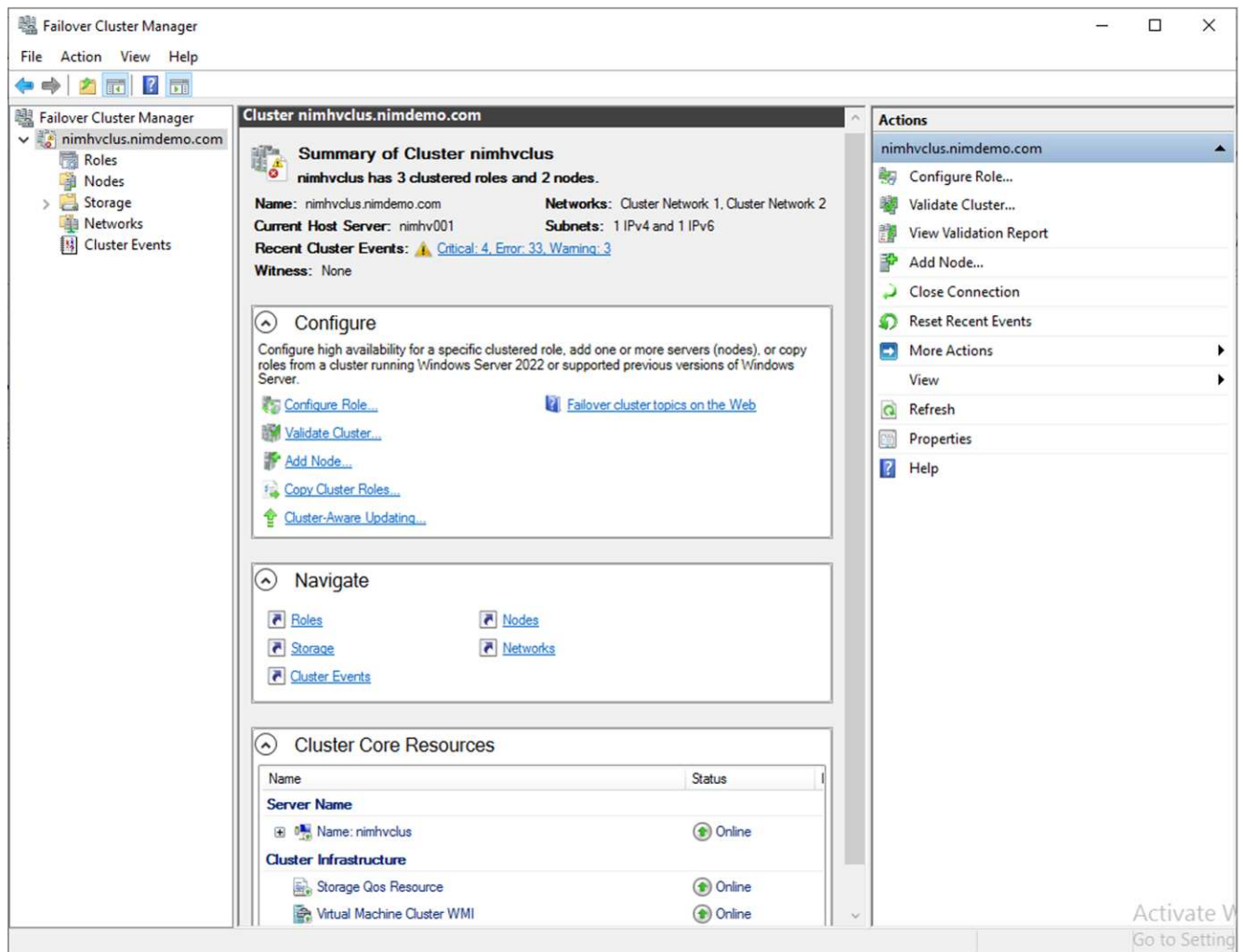
注意：添加多个会话(最小值为5到8)以提高性能并利用带宽。

创建集群

仅限一台服务器

1. 右键单击PowerShell图标并选择、以使用管理权限启动PowerShell提示符 Run as Administrator`。
2. 创建新集群。

```
New-Cluster -Name <cluster_name> -Node <hostnames> -NoStorage
-StaticAddress <cluster_ip_address>
```



3. 为实时迁移选择适当的集群网络。

4. 指定CSV网络。

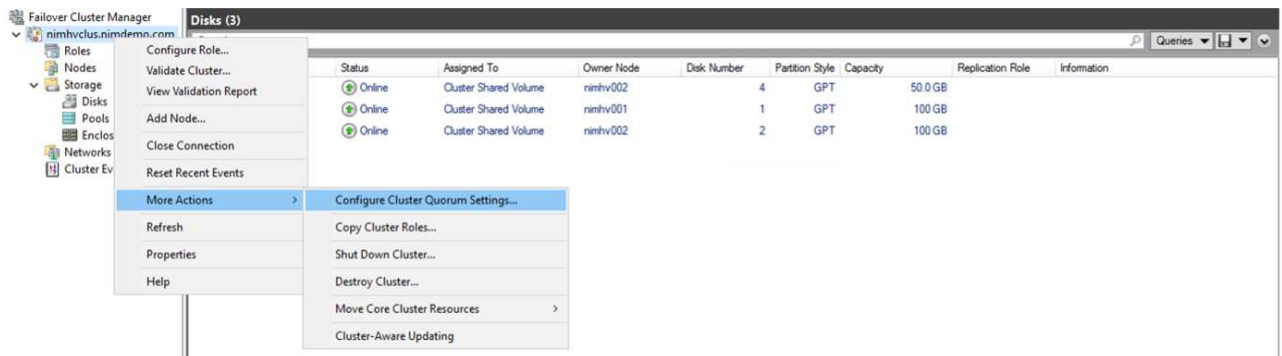
```
(Get-ClusterNetwork -Name Cluster).Metric = 900
```

5. 更改集群以使用仲裁磁盘。

a. 右键单击PowerShell图标并选择"Run as Administrator (以管理员身份运行)"、以管理员权限启动PowerShell提示符。

```
start-ClusterGroup "Available Storage" | Move-ClusterGroup -Node  
$env:COMPUTERNAME
```

b. 在故障转移集群管理器中、选择 Configure Cluster Quorum Settings。



c. 单击欢迎页面中的下一步。

d. 选择仲裁见证、然后单击下一步。

e. 选择Configure a disk witness` (配置磁盘见证)、然后单击Next (下一步)。

f. 从可用存储中选择磁盘W: 、然后单击下一步。

g. 在确认页面中单击下一步、并在摘要页面上单击完成。

有关仲裁和见证的更多详细信息、请参见 ["配置和管理仲裁"](#)

6. 从故障转移集群管理器运行集群验证向导以验证部署。

7. 创建CSV LUN以存储虚拟机数据、并通过故障转移集群管理器中的角色创建高可用性虚拟机。

在NetApp存储上部署Microsoft Hyper-V: 注意事项

要确保应用程序、服务和 workload 能够在Hyper-V环境中有效运行、此步骤至关重要。兼容性检查必须包括操作系统版本、Windows服务器版本、应用程序依赖项、数据库系统以及现有环境中存在的任何特定配置或自定义设置。

正确估算存储规模

在部署工作负载或从现有虚拟机管理程序迁移之前、请确保工作负载的规模已调整为满足所需的性能。为此、可以轻松收集每个虚拟机的性能数据、这些数据会收集CPU (已用/已配置)、内存(已用/已配置)、存储(已配置/已使用)、网络吞吐量和延迟的统计信息、并汇总读/写IOPS、吞吐量和块大小。要成功部署并正确估算存储阵列和工作负载主机的规模、必须使用这些参数。

注意：在估算Hyper-V和相关工作负载的存储规模时、请规划IOPS和容量。

注意：对于I/O密度较高的VM或需要大量资源和容量的VM、请将操作系统和数据磁盘隔离。操作系统和应用程序二进制文件更改频率不高、并且可以接受卷崩溃状态一致性。

注意：与使用VHD相比、使用子系统连接的存储(也称为子系统)来存储高性能数据磁盘。这也有助于简化克隆过程。

提高虚拟机性能

选择适当数量的RAM和vCPU以获得最佳性能、并将多个磁盘连接到一个虚拟SCSI控制器。对于部署中的虚拟磁盘、仍然建议使用固定VHDx作为主要选择、并且使用任何类型的VHDx虚拟磁盘都没有限制。

注意：避免在Windows Server上安装不必要的角色，因为这些角色将不会被利用。

注意：选择Gen2作为能够从SCSI控制器加载VM的虚拟机的代数，并且基于VMBus和VSP/VSC架构进行启动，这将显著提高VM的整体性能。

注意：避免频繁执行检查点操作、因为这会对虚拟机的性能产生负面影响。

SMB3.0设计和注意事项

SMB 3.0文件共享可用作Hyper-V的共享存储ONTAP支持通过适用于Hyper-V的SMB共享实现无中断运行Hyper-V可以使用SMB文件共享来存储虚拟机文件、例如配置、快照和虚拟硬盘(Virtual Hard Disk、VHD)文件。使用专用的ONTAP CIFS SVM实现Hyper-V基于SMB3.0的共享用于存储虚拟机文件的卷必须使用NTFS安全模式卷创建。建议在Hyper-V主机和NetApp阵列之间使用10 GB网络(如果有)进行连接。如果网络连接为1 GB、NetApp建议创建一个由多个1 GB端口组成的接口组。将提供SMB多通道的每个NIC连接到其专用IP子网、以便每个子网在客户端和服务器之间提供一条路径。

要点

- 在ONTAP SVM上启用SMB多通道
- ONTAP CIFS SVM应在集群中的每个节点上至少具有一个数据LIF。
- 使用的共享必须配置为具有持续可用属性集。
- 现在、每个AFF (A系列和C系列)、纯SAN阵列(ASA)和FAS系统都包含ONTAP One。因此、不需要单独的许可证。
- 对于共享VHDx、请使用子系统连接的iSCSI LUN

注意：ODX支持并支持多种协议。在文件共享与iSCSI或FCP连接的LUN之间复制数据也会利用ODX。

注意：应相应地设置集群中节点的时间设置。如果NetApp CIFS服务器必须加入Windows Active Directory (AD)域、则应使用网络时间协议(NTP)。

注意：必须通过CIFS服务器启用较大的MTU值。数据包大小较小可能会导致性能下降。

配置SMB卷

1. 验证是否已在Storage Virtual Machine (SVM)上启用所需的CIFS服务器选项
2. 应将以下选项设置为true：smb2-enabled smb3-enabled copy-offload-enabled scopy-enabled是-多通道-启用是-大型-MTU-enabled

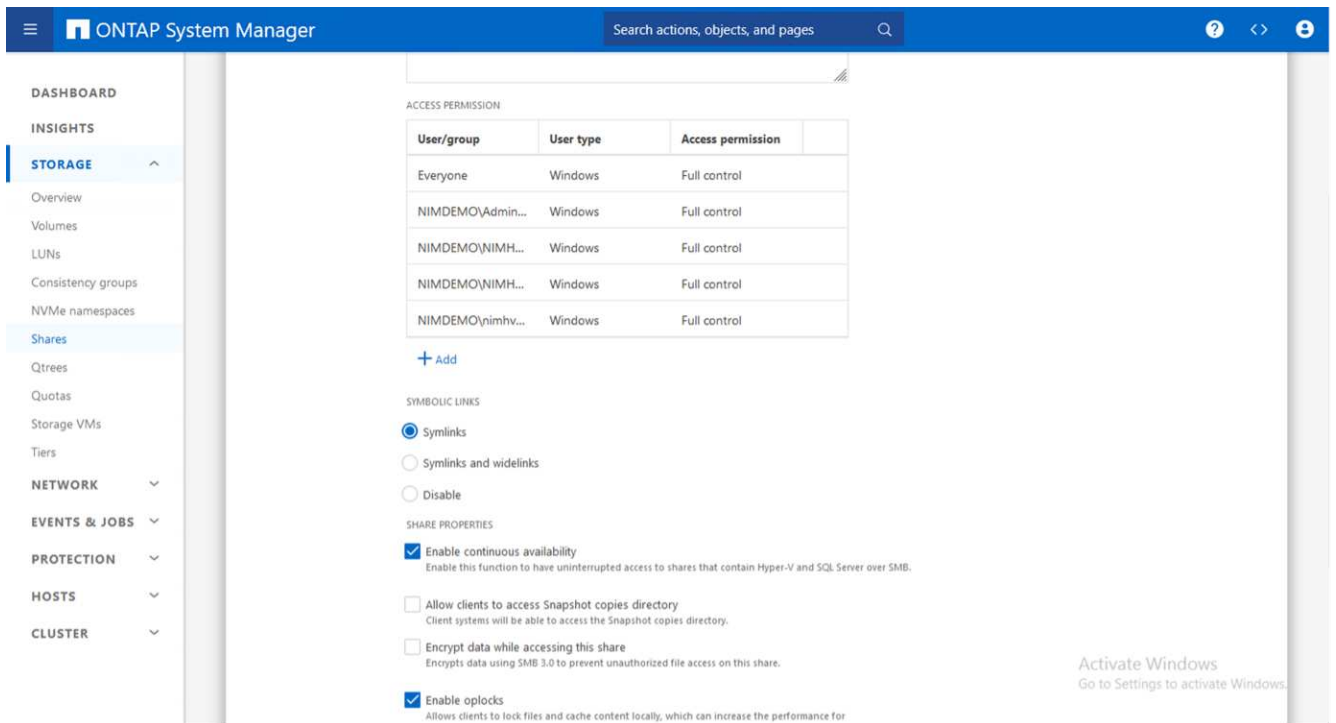
```
HV_NestedCluster:~> vservers cifs options show -vserver NestedHvsvm01 -fields copy-offload-enabled, is-multichannel-enabled, is-large-mtu-enabled, smb2-enabled, smb3-enabled, copy-offload-enabled, shadowcopy-enabled
vservers
-----
NestedHvsvm01 true true true true true true
```

3. 在Storage Virtual Machine (SVM)上创建NTFS数据卷、然后配置持续可用的共享以用于Hyper-V

```
HV_NestedCluster:~> volume create -vserver NestedHvsvm01 -volume hvdemo smb -aggregate HV_NestedCluster_01_VM_DISK_1 -size 500GB -security-style ntfs -function-path /hvdemo smb
[Job 169] Job succeeded: Successful
```

注意：除非将配置中使用的卷创建为NTFS安全模式卷、否则基于SMB的Hyper-V不会正常运行。

4. 在共享上启用持续可用并配置NTFS权限、以包括具有完全控制权的Hyper-V节点。



有关详细的最佳实践指导、请参见 ["Hyper-V部署准则和最佳实践"](#)。

有关追加信息、请参见 ["基于 SMB 的 Hyper-V 的 SMB 服务器和卷要求"](#)。

块协议设计和注意事项

要点

- 在主机上使用多路径(MPIO)管理多个路径。根据需要创建更多路径、以便于数据移动操作或利用更多I/O资

源、但不要超过主机操作系统可支持的最大路径数。

- 在访问LUN的主机上安装Host Utilities Kit。
- 至少创建8个卷。

注意：每个卷使用一个LUN、因此LUN与CSV的比率为1：1映射。

- 在要使用iSCSI或光纤通道提供数据的每个存储控制器上、SVM的每个以太网网络或光纤通道网络都应具有一个LIF。
- 使用FCP或iSCSI提供数据的SVM需要SVM管理接口。

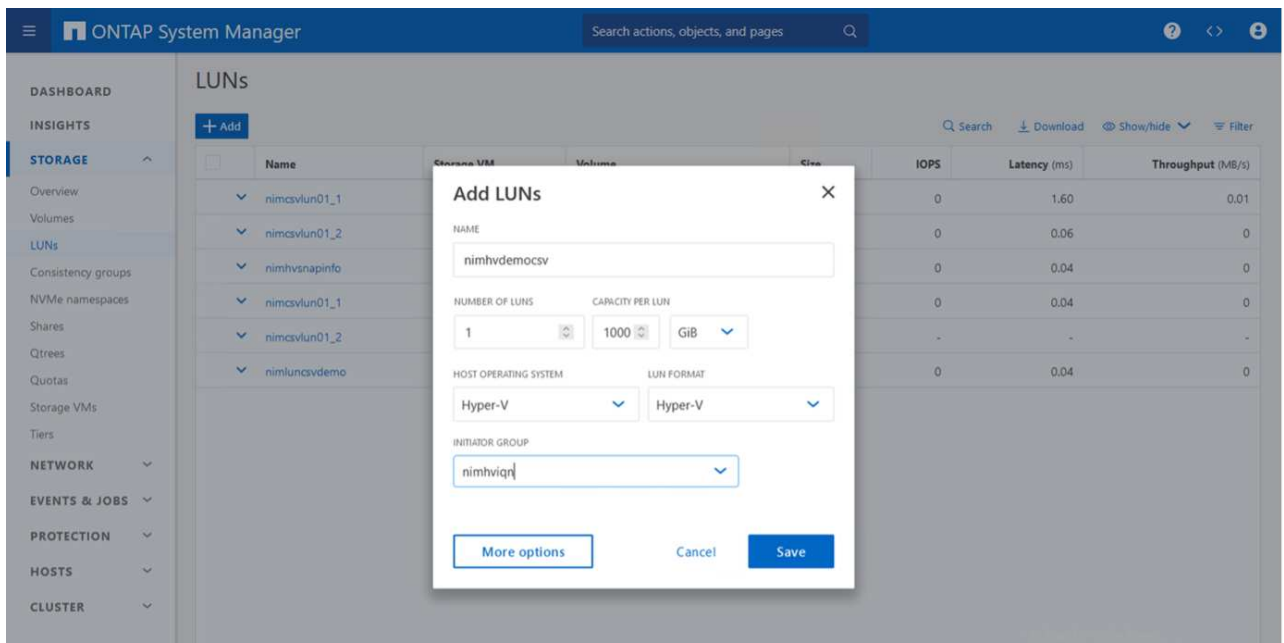
正在配置iSCSI卷

要配置iSCSI卷、请确保满足以下前提条件。

- Storage Virtual Machine (SVM)应启用iSCSI协议并创建相应的逻辑接口(Logical Interface、Logical Interface、Logical Interface、Logical Interface、SVM)。
- 指定聚合必须具有足够的可用空间来容纳LUN。

注意：默认情况下，ONTAP使用选择性LUN映射(选择性LUN Map, SLM)使LUN只能通过LUN所属节点及其高可用性(HA)配对节点上的路径进行访问。

- 配置每个节点上的所有iSCSI LIFs、以便在LUN移动到集群中的另一节点时实现LUN移动性。
- 步骤 *
 1. 使用System Manager导航到LUN窗口(可以使用ONTAP命令行界面执行相同的操作)。
 2. 单击创建。
 3. 浏览并选择要在其中创建LUN的指定SVM、此时将显示创建LUN向导。
 4. 在常规属性页面上、为包含Hyper-V虚拟机虚拟硬盘(Virtual Hard Disk、VHD)的LUN选择Hyper-V。



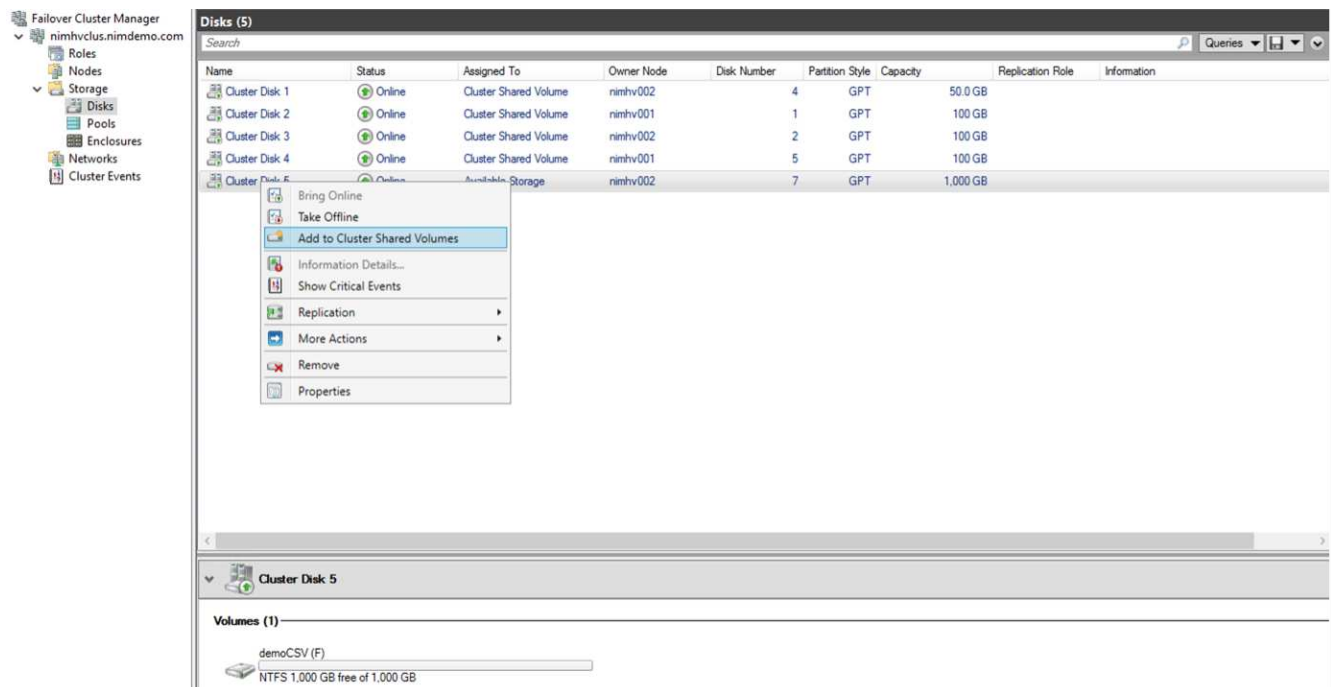
5. <单击更多选项>在"LUN Container"页面上、选择现有FlexVol卷、否则将创建新卷。
6. <单击更多选项>在启动程序映射页面上、单击添加启动程序组、在常规选项卡上输入所需信息、然后在启动程序选项卡上输入主机的iSCSI启动程序节点名称。
7. 确认详细信息、然后单击完成以完成向导。

创建LUN后、转到故障转移集群管理器。要将磁盘添加到CSV、必须将该磁盘添加到集群的可用存储组(如果尚未添加)、然后将该磁盘添加到集群上的CSV。

注意：在故障转移集群中、CSV功能默认处于启用状态。

将磁盘添加到可用存储：

1. 在故障转移集群管理器的控制台树中、展开集群名称、然后展开存储。
2. 右键单击磁盘、然后选择添加磁盘。此时将显示一个列表、其中显示了可添加以在故障转移集群中使用的磁盘。
3. 选择要添加的一个或多个磁盘、然后选择确定。
4. 此时、磁盘将分配给可用存储组。
5. 完成后、选择刚刚分配给可用存储的磁盘、右键单击所选内容、然后选择添加到集群共享卷。



6. 磁盘现在已分配给集群中的集群共享卷组。磁盘作为编号卷(挂载点)公开给每个集群节点、位于%SystemDrive%ClusterStorage文件夹下。这些卷将显示在CSVFS文件系统中。

有关追加信息、请参见 ["在故障转移集群中使用集群共享卷"](#)。

创建高可用性虚拟机：

要创建高可用性虚拟机、请执行以下步骤：

1. 在故障转移集群管理器中、选择或指定所需的集群。确保已展开集群下的控制台树。

2. 单击角色。
3. 在操作窗格中、单击虚拟机、然后单击新建虚拟机。此时将显示"New Virtual Machine Wizard"(新建虚拟机向导)。单击下一步。
4. 在Specify Name and Location页面上、指定虚拟机的名称、例如nimdemO。单击将虚拟机存储在其他位置、然后键入完整路径、或者单击浏览并导航到共享存储。
5. 为与物理网络适配器关联的虚拟交换机分配内存并配置网络适配器。
6. 在连接虚拟硬盘页面上、单击创建虚拟硬盘。
7. 在"Installation Options (安装选项)"页面上、单击"Install an operating system from a boot CD/DVD-ROM (从启动CD/DVD-ROM安装操作系统)"。在介质下、指定介质的位置、然后单击完成。
8. 此时将创建虚拟机。然后、故障转移集群管理器中的高可用性向导会自动配置虚拟机以实现高可用性。

使用ODX功能快速配置虚拟磁盘

通过ONTAP中的ODX功能、只需复制由ONTAP存储系统托管的主VHDX文件、即可创建主VHDX的副本。由于启用了ODX的副本不会将任何数据放在网络线上、因此复制过程会在NetApp存储端进行、因此速度可以提高多达6到8倍。快速配置的一般注意事项包括存储在文件共享上的主系统已准备好的映像以及由Hyper-V主机启动的常规复制过程。

注意：ONTAP支持对SMB和SAN协议使用ODX。

注意：要利用Hyper-V中ODX副本卸载直通的使用情形、子操作系统必须支持ODX、并且子操作系统的磁盘必须是SCSI磁盘、并由支持ODX的存储(SMB或SAN)提供后备支持。子操作系统上的IDE磁盘不支持ODX直通。

性能优化

虽然建议的每个CSV的VM数量是主观的、但许多因素决定了可放置在每个CSV或SMB卷上的最佳VM数量。虽然大多数管理员仅考虑容量、但发送到VHDX的并发I/O量是影响整体性能的最关键因素之一。控制性能最简单的方法是、调节每个CSV或共享上放置的虚拟机数量。如果并发虚拟机I/O模式向CSV或共享发送的流量过多、则磁盘队列会填满、并会产生较高的延迟。

SMB卷和CSV规模估算

确保解决方案具有足够的端到端大小以避免出现瓶颈、在为Hyper-V VM存储创建卷时、最佳实践是创建不超过所需大小的卷。适当的卷规模可防止意外在CSV上放置过多的虚拟机、并降低资源争用的可能性。每个集群共享卷(CSV)都支持一个或多个VM。要放置在CSV上的VM数量取决于工作负载和业务首选项以及如何使用ONTAP存储功能(例如快照和复制)。在大多数部署情形下、将多个VM放置在一个CSV上是一个很好的起点。根据特定使用情形调整此方法、以满足性能和数据保护要求。

由于卷和VHDX大小可以轻松增加、因此、如果虚拟机需要额外容量、则不必将CSV的大小设置为大于所需大小。可以使用Diskpart扩展CSV大小、或者更简单的方法是创建新的CSV并将所需的VM迁移到新的CSV。为了获得最佳性能、最佳做法是增加CSV数量、而不是作为临时措施增加其大小。

migration

在当前市场条件下、最常见的使用情形之一是迁移。客户可以使用VMM Fabric或其他第三方迁移工具来迁移VM。这些工具使用主机级副本将数据从源平台移动到目标平台、这可能会非常耗时、具体取决于迁移范围内的虚拟机数量。

与使用基于主机的迁移过程相比、在这种情况下使用ONTAP可以加快迁移速度。ONTAP还支持将虚拟机从一个

虚拟机管理程序快速迁移到另一个虚拟机管理程序(此处为ESXi到Hyper-V)。在NetApp存储上、任何大小的VMDK都可以在几秒钟内转换为VHDx。这就是我们的PowerShell方式—它利用NetApp FlexClone®技术快速转换VM硬盘。它还负责创建和配置目标和目标VM。

此过程有助于最大限度地减少停机时间并提高业务效率。此外、它还可以降低许可成本、减少受制于单个供应商的情况、从而提供丰富的选择和灵活性。对于希望优化VM许可成本并扩展IT预算的组织来说、这一点也非常有用。

以下视频介绍了将虚拟机从VMware ESX迁移到Hyper-V的过程

[从ESX到Hyper-V的零接触迁移](#)

有关使用FlexClone和PowerShell进行迁移的追加信息、请参见 ["用于迁移的PowerShell脚本"](#)。

在NetApp存储上部署Microsoft Hyper-V：数据保护

数据保护是任何生产工作负载的关键租户。本节介绍如何备份和还原Hyper-V虚拟机。

使用NetApp存储快照还原

备份VM以及快速恢复或克隆VM是ONTAP卷的一大优势。使用Snapshot副本可以快速创建VM甚至整个CSV卷的FlexClone副本、而不会影响性能。这样、在克隆生产数据卷并将其挂载到QA、暂存和开发环境时、可以使用生产数据、而不会面临数据损坏的风险。FlexClone卷可用于创建生产数据的测试副本、而无需将复制数据所需的空间量增加一倍。

请注意、Hyper-V节点会为每个磁盘分配一个唯一ID、为具有相应分区(MBR或GPT)的卷创建快照将具有相同的唯一标识。MBR使用磁盘签名、而GPT使用GUID (全局唯一标识符)。如果是独立Hyper-V主机、则可以轻松挂载FlexClone卷、而不会产生任何冲突。这是因为独立的Hyper-V服务器可以自动检测重复的磁盘ID并动态更改它们、而无需用户干预。可以使用此方法根据情况需要复制vHD来恢复虚拟机。

虽然对于独立Hyper-V主机来说非常简单、但Hyper-V集群的操作步骤有所不同。恢复过程包括将FlexClone卷映射到独立Hyper-V主机、或者使用diskpart通过将FlexClone卷映射到独立Hyper-V主机来手动更改签名(这一点非常重要、因为磁盘ID冲突会导致磁盘无法联机)、完成后、将FlexClone卷映射到集群。

使用第三方解决方案备份和还原

注意：本部分使用Commvault,但这适用于其他第三方解决方案。

CommvaultIntelliSnap®利用ONTAP快照创建基于硬件的快照

Hyper-V可以根据Hyper-V虚拟机管理程序或虚拟机组的配置自动执行备份、也可以手动执行虚拟机组或特定虚拟机的备份。IntelliSnap可以快速保护Hyper-V环境、将生产虚拟化场的负载降到最低。IntelliSnap技术与虚拟服务器代理(VSA)的集成使NetApp ONTAP阵列能够在几分钟内完成对大量虚拟机和数据存储的备份。通过精细访问、可以从二级存储层恢复单个文件和文件夹以及完整的子系统.vhd文件。

在配置虚拟化环境之前、请部署需要与阵列进行快照集成的适当代理。Microsoft Hyper-V虚拟化环境需要以下代理：

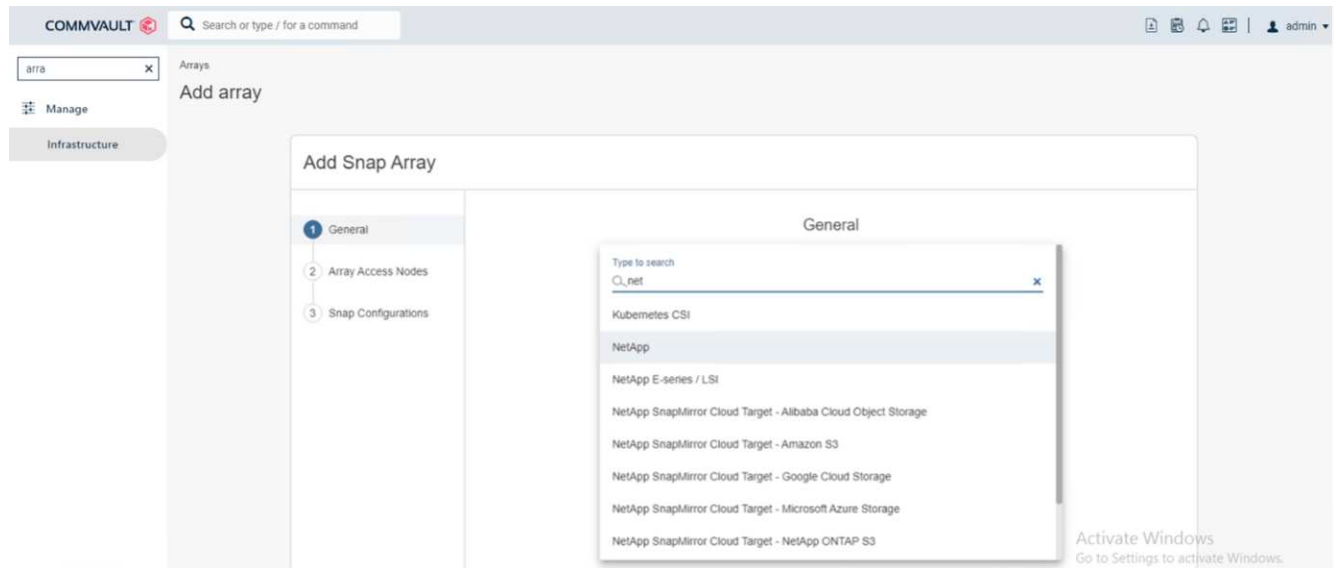
- MediaAgent
- 虚拟服务器代理(VSA)
- VSS硬件提供程序(Windows Server 2012及更高版本的操作系统)

使用阵列管理配置NetApp阵列

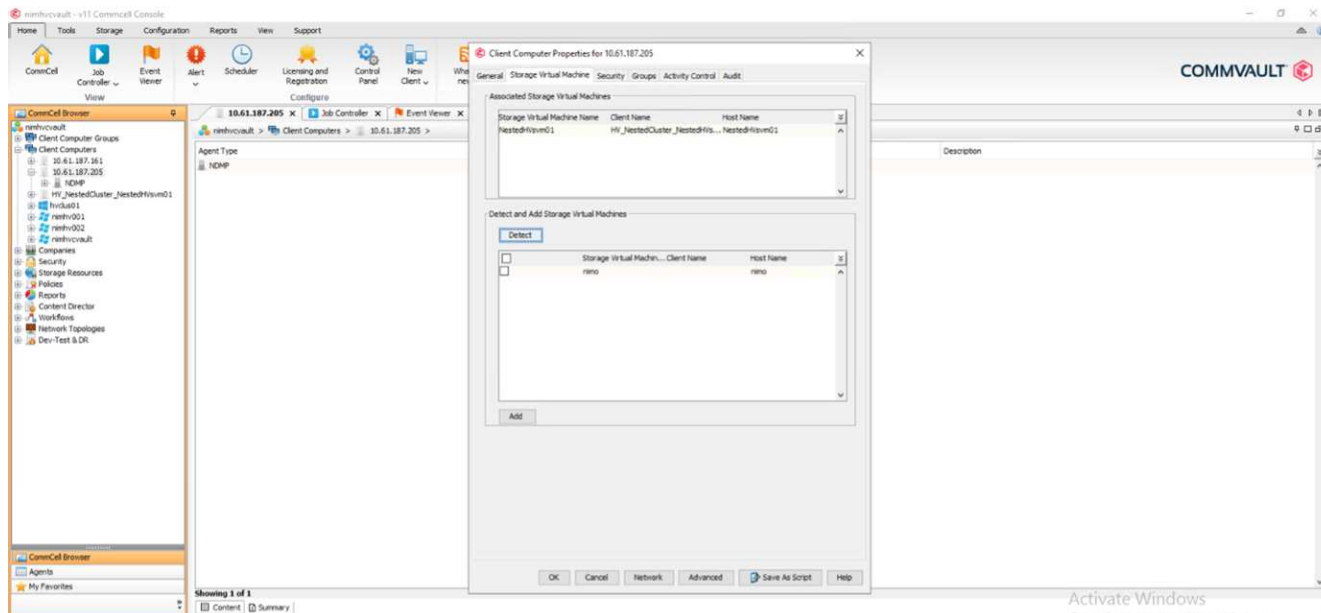
以下步骤显示了如何在使用ONTAP阵列和Hyper-V的环境中配置IntelliSnap虚拟机备份

1. 在"CommCell Console"(CommCell控制台)的功能区上、单击"Storage"(存储)选项卡、然后单击"Array Management"(阵列管理)。
2. 此时将显示"Array Management"(阵列管理)对话框。
3. 单击添加。

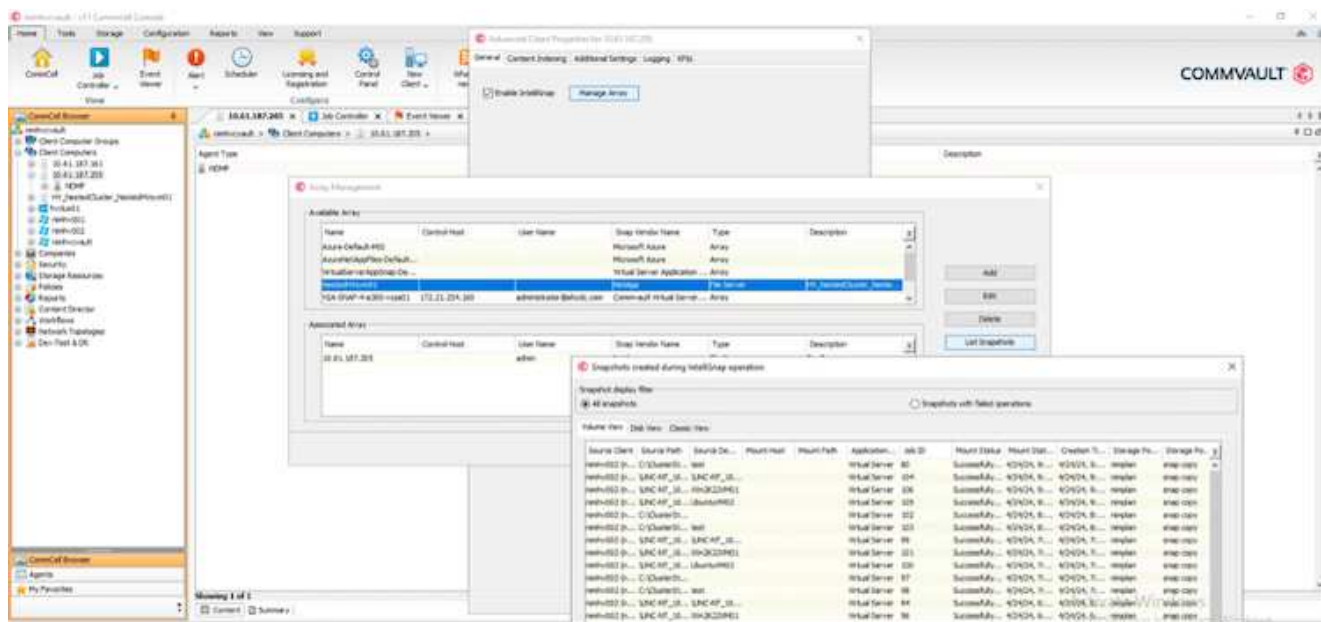
出现"Array Properties (阵列属性)"对话框。



4. 在常规选项卡上、指定以下信息：
5. 从Snap Vendor列表中、选择NetApp。
6. 在名称框中、输入主文件服务器的主机名、完全限定域名(FQDN)或TCP/IP地址。
7. 在阵列访问节点选项卡上、选择可用的介质代理。
8. 在Snap Configuration选项卡上、根据需要配置Snapshot Configuration Properties。
9. 单击确定。
10. <Mandatory step>完成后、还可以在NetApp存储阵列上配置SVM、方法是使用检测选项自动检测Storage Virtual Machine (SVM)、然后选择SVM、并使用添加选项将SVM作为阵列管理条目添加到CommServe数据库中。



11. 单击Advanced (高级)(如下图所示)、然后选中Enable IntelliSnap (启用IntelliSnap)复选框。



有关配置阵列的详细步骤、请参见 ["正在配置NetApp阵列"](#) 和 ["在NetApp阵列上配置Storage Virtual Machine"](#)

添加Hyper-V作为虚拟机管理程序

下一步是添加Hyper-V虚拟机管理程序并添加虚拟机组。

前提条件

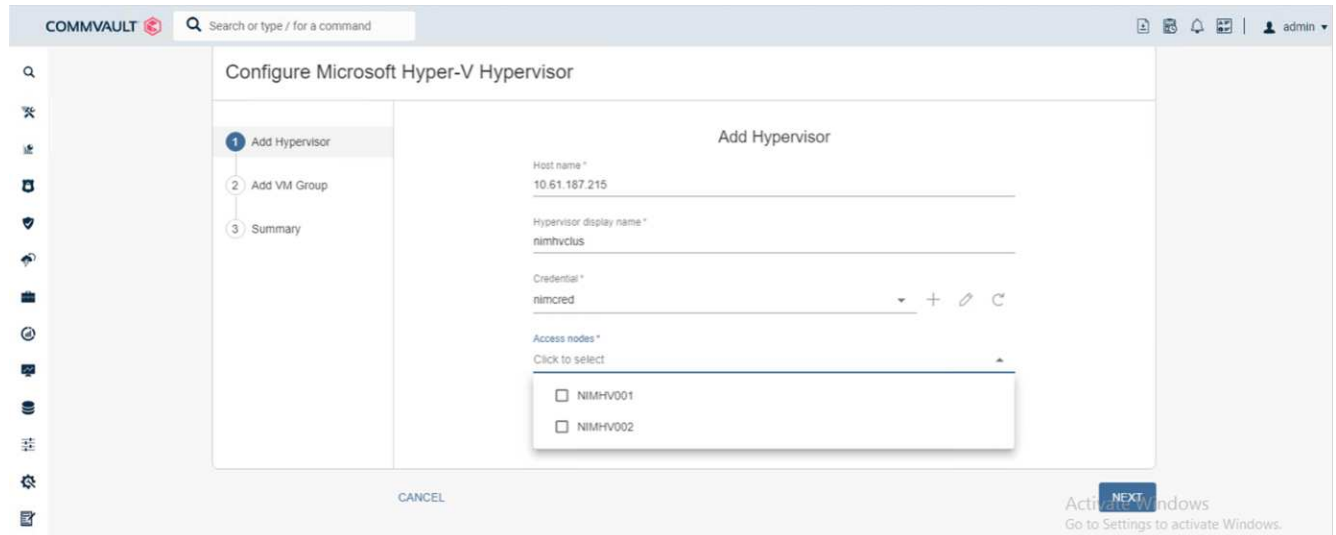
- 虚拟机管理程序可以是Hyper-V集群、集群中的Hyper-V服务器或独立的Hyper-V服务器。
- 对于Hyper-V Server 2012及更高版本、用户必须属于Hyper-V管理员组。对于Hyper-V集群、用户帐户必须具有完全集群权限(读取和完全控制)。
- 确定要安装虚拟服务器代理(VSA)以创建用于备份和还原操作的访问节点(VSA代理)的一个或多个节点。要发

现Hyper-V服务器、必须在CommServe系统上安装VSA。

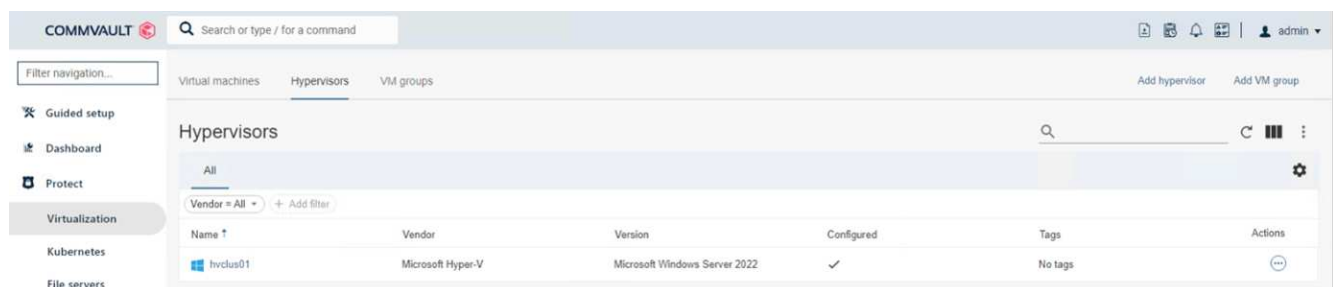
- 要对Hyper-V 2012 R2使用更改的块跟踪、请选择Hyper-V集群中的所有节点。

以下步骤显示了如何将Hyper-V添加为虚拟机管理程序。

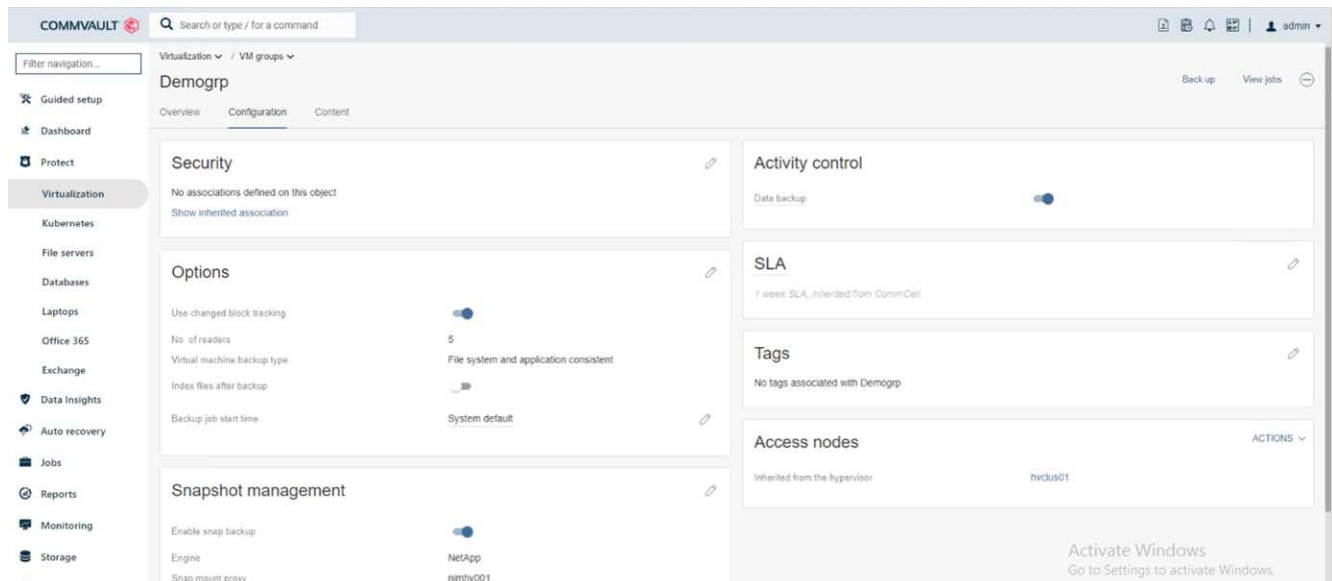
1. 核心设置完成后、在保护选项卡上、单击虚拟化磁贴。
2. 在创建服务器备份计划页面上、键入计划的名称、然后提供有关存储、保留和备份计划的信息。
3. 此时将显示"Add hypervisor"页面>"Select vendor": 选择Hyper-V (输入IP地址或FQDN和用户凭据)
4. 对于Hyper-V服务器、请单击Discover node。填充节点字段后、选择要安装虚拟服务器代理的一个或多个节点。



5. 单击Next (下一步)和Save (保存)。



6. 在"Add VM group"(添加VM组)页面上、选择要保护的虚拟机(Demo是在本例中创建的VM组)、然后启用IntelliSnap选项、如下所示。



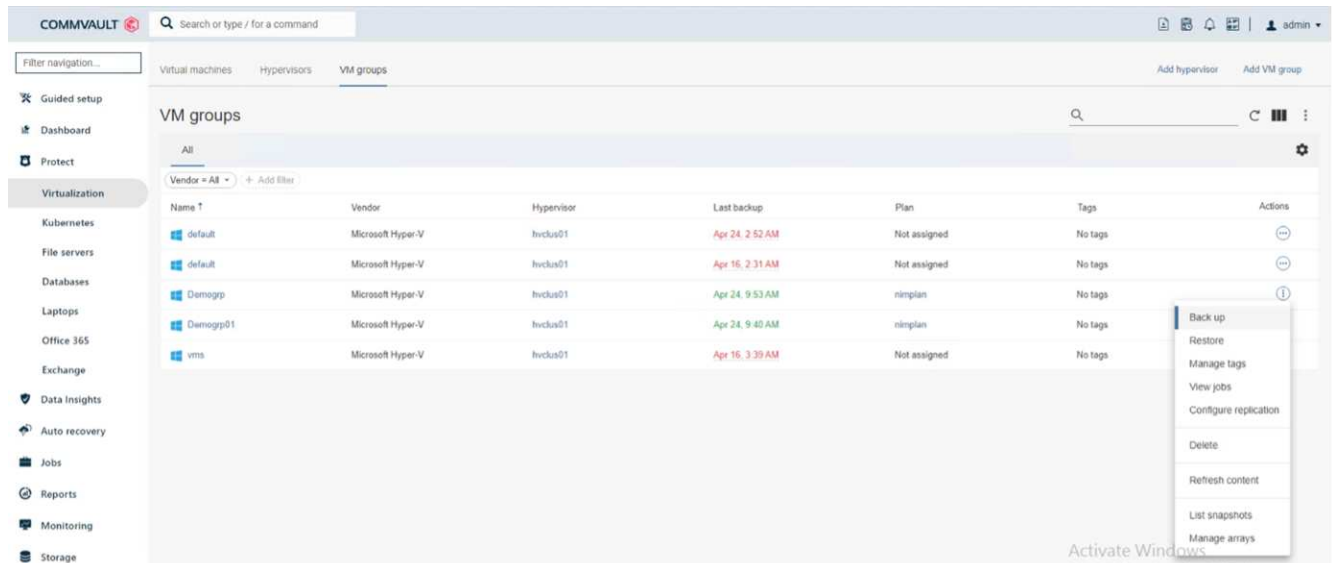
注意：在VM组上启用IntelliSnap后，CommVault会自动为主(Snap)副本和备份副本创建计划策略。

7. 单击保存。

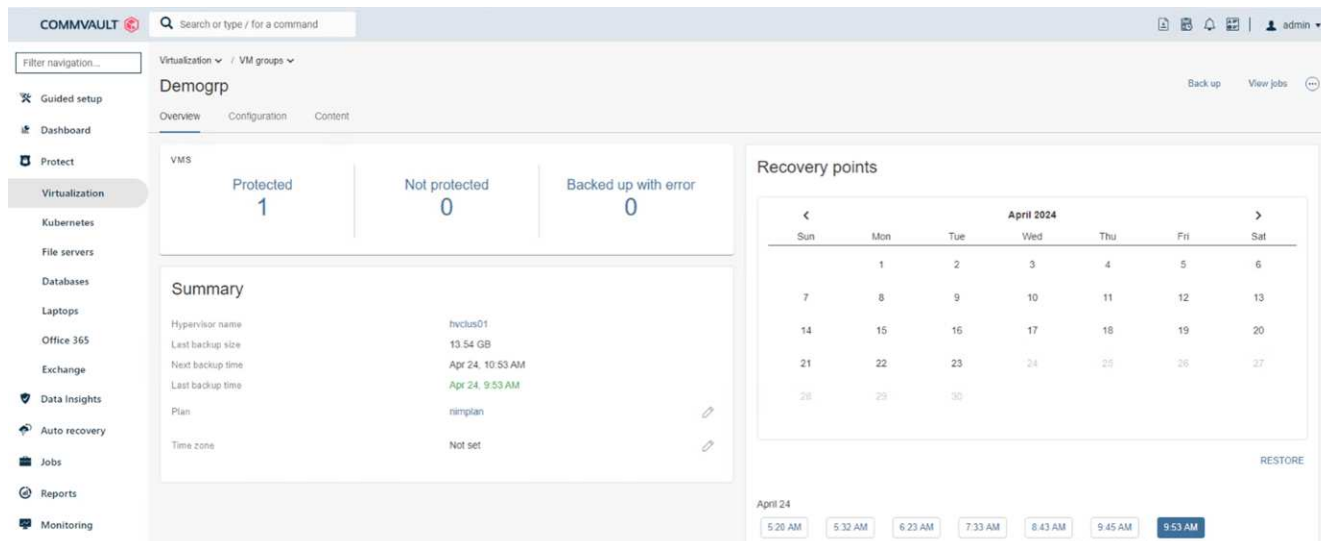
有关配置阵列的详细步骤、请参见 ["添加虚拟机管理程序"](#)。

执行备份：

1. 在导航窗格中、转至"Protect">"Virtualization (保护>虚拟化)"。此时将显示虚拟机页面。
2. 备份虚拟机或虚拟机组。在此演示中、将选择VM组。在VM组对应的行中、单击操作按钮action_button、然后选择备份。在这种情况下、nimplan是与DemOTP和DemOrp01相关的计划。



3. 备份成功后、还原点将可用、如屏幕截图所示。从Snap副本中、可以还原完整的VM以及子系统文件和文件夹。

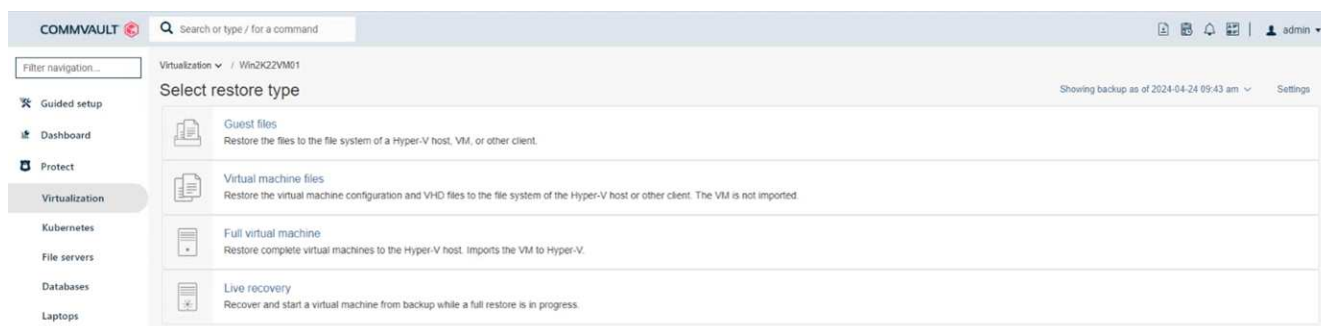


注意：对于利用率较高的关键虚拟机，每个CSV所保留的虚拟机较少

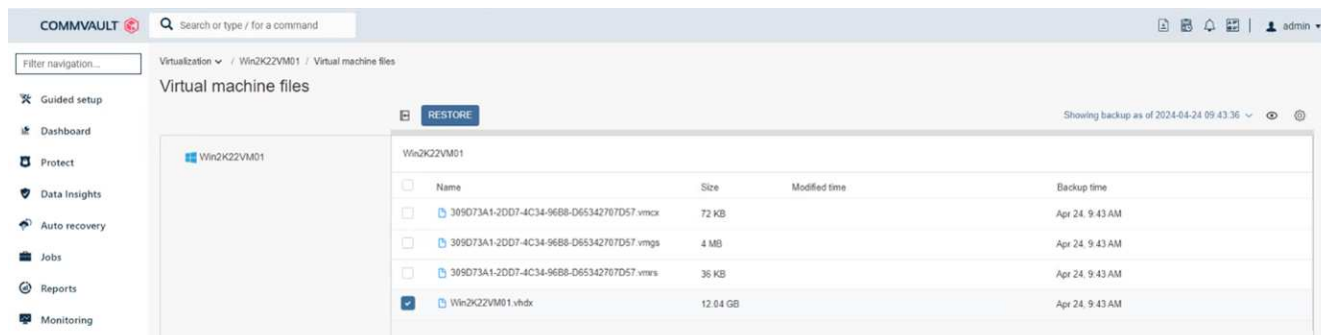
执行还原操作：

通过还原点还原完整的VM、子系统文件和文件夹或虚拟磁盘文件。

1. 在导航窗格中、转至"Protect">"Virtualation"、此时将显示"Virtual Machine"页面。
2. 单击VM Groups选项卡。
3. 此时将显示虚拟机组页面。
4. 在VM Groups区域中、对包含虚拟机的VM组单击Restore。
5. 此时将显示Select restore type页面。



6. 根据所做的选择选择子文件或完整虚拟机、然后触发还原。



有关所有受支持的还原选项的详细步骤、请参见 ["Hyper-V还原"](#)。

高级NetApp ONTAP选项

NetApp SnapMirror可实现高效的站点间存储复制、从而导致灾难快速、可靠且易于管理的恢复、可满足当今全球企业的需求。SnapMirror可通过LAN和广域网高速复制数据、为任务关键型应用程序提供高数据可用性和快速恢复、并具有出色的存储重复数据删除和网络压缩功能。借助NetApp SnapMirror技术、灾难恢复可以保护整个数据中心。卷可以增量备份到异地位置。SnapMirror按照所需RPO的频率执行基于块的增量复制。块级更新可降低带宽和时间要求、并在灾难恢复站点保持数据一致性。

重要的一步是创建整个数据集的一次性基线传输。执行增量更新前需要执行此操作。此操作包括在源上创建Snapshot副本、以及将其引用的所有数据块传输到目标文件系统。初始化完成后、可以按计划或手动触发更新。每次更新仅将新块和更改的块从源文件系统传输到目标文件系统。此操作包括在源卷上创建Snapshot副本、将其与基线副本进行比较、以及仅将发生更改的块传输到目标卷。新副本将成为下次更新的基线副本。由于复制是定期进行的、因此SnapMirror可以整合更改的块并节省网络带宽。对写入吞吐量和写入延迟的影响微乎其微。

要执行恢复、请完成以下步骤：

1. 连接到二级站点上的存储系统。
2. 中断SnapMirror关系。
3. 将SnapMirror卷中的LUN映射到二级站点上Hyper-V服务器的启动程序组(igrop)。
4. 将LUN映射到Hyper-V集群后、使这些磁盘联机。
5. 使用故障转移集群PowerShell cmdlet、将磁盘添加到可用存储并将其转换为CSV。
6. 将CSV中的虚拟机导入到Hyper-V管理器中、使其具有高可用性、然后将其添加到集群中。
7. 打开VM。

在NetApp存储上部署Microsoft Hyper-V：总结

ONTAP是部署各种IT工作负载的最佳共享存储基础。ONTAP AFF或ASA平台既灵活又可扩展、可用于多种使用情形和应用程序。启用了Windows Server 2022和Hyper-V的Windows Server 2022和Hyper-V是虚拟化解决方案的一个常见用例、本文档将对此进行介绍。凭借ONTAP存储的灵活性和可扩展性以及相关功能、客户可以从规模合适的存储层入手、并根据不断变化的业务需求进行扩展。在当前市场环境下、Hyper-V提供了一个完美的虚拟机管理程序替代选项、可提供VMware提供的大多数功能。

在NetApp存储上部署Microsoft Hyper-V：迁移脚本

本节包含可用于使用FlexClone进行迁移的PowerShell脚本。

PowerShell脚本

```
param (  
    [Parameter(Mandatory=$True, HelpMessage="VCenter DNS name or IP  
Address")]
```

```

[String]$VCENTER,
[Parameter(Mandatory=$True, HelpMessage="NetApp ONTAP NFS Datastore
name")] ]
[String]$DATASTORE,
[Parameter(Mandatory=$True, HelpMessage="VCenter credentials")] ]
[System.Management.Automation.PSCredential]$VCENTER_CREDS,
[Parameter(Mandatory=$True, HelpMessage="The IP Address of the ONTAP
Cluster")] ]
[String]$ONTAP_CLUSTER,
[Parameter(Mandatory=$True, HelpMessage="NetApp ONTAP VServer/SVM
name")] ]
[String]$VSERVER,
[Parameter(Mandatory=$True, HelpMessage="NetApp ONTAP NFS,SMB Volume
name")] ]
[String]$ONTAP_VOLUME_NAME,
[Parameter(Mandatory=$True, HelpMessage="ONTAP NFS/CIFS Volume mount
Drive on Hyper-V host")] ]
[String]$ONTAP_NETWORK_SHARE_ADDRESS,
[Parameter(Mandatory=$True, HelpMessage="NetApp ONTAP Volume QTree
folder name")] ]
[String]$VHDX_QTREE_NAME,
[Parameter(Mandatory=$True, HelpMessage="The Credential to connect to
the ONTAP Cluster")] ]
[System.Management.Automation.PSCredential]$ONTAP_CREDS,
[Parameter(Mandatory=$True, HelpMessage="Hyper-V VM switch name")] ]
[String]$HYPERV_VM_SWITCH
)

function main {

    ConnectVCenter

    ConnectONTAP

    GetVMList

    GetVMInfo

    #PowerOffVMs

    CreateOntapVolumeSnapshot

    Shift

    ConfigureVMsOnHyperV
}

```

```

function ConnectVCenter {
    Write-Host
    -----
    ----" -ForegroundColor Cyan
    Write-Host "Connecting to vCenter $VCENTER" -ForegroundColor Magenta
    Write-Host
    -----
    ----`n" -ForegroundColor Cyan

    [string]$vmwareModuleName = "VMware.VimAutomation.Core"

    Write-Host "Importing VMware $vmwareModuleName Powershell module"
    if ((Get-Module|Select-Object -ExpandProperty Name) -notcontains
$vmwareModuleName) {
        Try {
            Import-Module $vmwareModuleName -ErrorAction Stop
            Write-Host "$vmwareModuleName imported successfully"
            -ForegroundColor Green
        } Catch {
            Write-Error "Error: $vmwareMdouleName PowerShell module not
found"
            break;
        }
    }
    else {
        Write-Host "$vmwareModuleName Powershell module already imported"
        -ForegroundColor Green
    }

    Write-Host "`nConnecting to vCenter $VCENTER"
    Try {
        $connect = Connect-VIServer -Server $VCENTER -Protocol https
        -Credential $VCENTER_CREDS -ErrorAction Stop
        Write-Host "Connected to vCenter $VCENTER" -ForegroundColor Green
    } Catch {
        Write-Error "Failed to connect to vCenter $VCENTER. Error : $($_.
.Exception.Message)"
        break;
    }
}

function ConnectONTAP {
    Write-Host "`n
-----
----" -ForegroundColor Cyan
    Write-Host "Connecting to VSerevr $VSERVER at ONTAP Cluster

```

```

$ONTAP_CLUSTER" -ForegroundColor Magenta
Write-Host
-----
----`n" -ForegroundColor Cyan

[string]$ontapModuleName = "NetApp.ONTAP"

Write-Host "Importing NetApp ONTAP $ontapModuleName Powershell module"
if ((Get-Module|Select-Object -ExpandProperty Name) -notcontains
$ontapModuleName) {
    Try {
        Import-Module $ontapModuleName -ErrorAction Stop
        Write-Host "$ontapModuleName imported successfully"
-ForegroundColor Green
    } Catch {
        Write-Error "Error: $vmwareModule Name PowerShell module not
found"
        break;
    }
}
else {
    Write-Host "$ontapModuleName Powershell module already imported"
-ForegroundColor Green
}

Write-Host "`nConnecting to ONTAP Cluster $ONTAP_CLUSTER"
Try {
    $connect = Connect-NcController -Name $ONTAP_CLUSTER -Credential
$ONTAP_CREDS -Vserver $VSERVER
    Write-Host "Connected to ONTAP Cluster $ONTAP_CLUSTER"
-ForegroundColor Green
} Catch {
    Write-Error "Failed to connect to ONTAP Cluster $ONTAP_CLUSTER.
Error : $($_.Exception.Message)"
    break;
}
}

function GetVMList {
    Write-Host "`n
-----
----" -ForegroundColor Cyan
    Write-Host "Fetching powered on VMs list with Datastore $DATASTORE"
-ForegroundColor Magenta
    Write-Host
-----

```

```

-----`n" -ForegroundColor Cyan
    try {
        $vmList = VMware.VimAutomation.Core\Get-VM -Datastore $DATASTORE
-ErrorAction Stop| Where-Object {$_.PowerState -eq "PoweredOn"} | OUT-
GridView -OutputMode Multiple
        # $vmList = Get-VM -Datastore $DATASTORE -ErrorAction Stop| Where-
Object {$_.PowerState -eq "PoweredOn"}

        if($vmList) {
            Write-Host "Selected VMs for Shift" -ForegroundColor Green
            $vmList | Format-Table -Property Name
            $Script:VMList = $vmList
        }
        else {
            Throw "No VMs selected"
        }
    }
    catch {
        Write-Error "Failed to get VM List. Error : $($_.Exception.
Message)"
        Break;
    }
}

function GetVMInfo {
    Write-Host
    -----
    -----" -ForegroundColor Cyan
    Write-Host "VM Information" -ForegroundColor Magenta
    Write-Host
    -----
    -----" -ForegroundColor Cyan
    $vmObjArray = New-Object System.Collections.ArrayList

    if($VMList) {
        foreach($vm in $VMList) {
            $vmObj = New-Object -TypeName System.Object

            $vmObj | Add-Member -MemberType NoteProperty -Name ID -Value
$vm.Id
            $vmObj | Add-Member -MemberType NoteProperty -Name Name -Value
$vm.Name
            $vmObj | Add-Member -MemberType NoteProperty -Name NumCpu
-Value $vm.NumCpu
            $vmObj | Add-Member -MemberType NoteProperty -Name MemoryGB
-Value $vm.MemoryGB

```



```

        $vmObj | Add-Member -MemberType NoteProperty -Name Firmware
-Value $vm.ExtensionData.Config.Firmware

    $vmDiskInfo = $vm | VMware.VimAutomation.Core\Get-HardDisk

    $vmDiskArray = New-Object System.Collections.ArrayList
    foreach($disk in $vmDiskInfo) {
        $diskObj = New-Object -TypeName System.Object

        $diskObj | Add-Member -MemberType NoteProperty -Name Name
-Value $disk.Name

        $fileName = $disk.FileName
        if ($fileName -match '\[(.*?)\]') {
            $dataStoreName = $Matches[1]
        }

        $parts = $fileName -split " "
        $pathParts = $parts[1] -split "/"
        $folderName = $pathParts[0]
        $fileName = $pathParts[1]

        $diskObj | Add-Member -MemberType NoteProperty -Name
DataStore -Value $dataStoreName
        $diskObj | Add-Member -MemberType NoteProperty -Name
Folder -Value $folderName
        $diskObj | Add-Member -MemberType NoteProperty -Name
Filename -Value $fileName
        $diskObj | Add-Member -MemberType NoteProperty -Name
CapacityGB -Value $disk.CapacityGB

        $null = $vmDiskArray.Add($diskObj)
    }

    $vmObj | Add-Member -MemberType NoteProperty -Name
PrimaryHardDisk -Value "[ $($vmDiskArray[0].DataStore) ] $($vmDiskArray[0]
.Folder)/$($vmDiskArray[0].Filename) "
    $vmObj | Add-Member -MemberType NoteProperty -Name HardDisks
-Value $vmDiskArray

    $null = $vmObjArray.Add($vmObj)

    $vmNetworkArray = New-Object System.Collections.ArrayList

    $vm |
    ForEach-Object {
        $VM = $_
    }

```

```

    $VM | VMware.VimAutomation.Core\Get-VMGuest | Select-Object
-ExpandProperty Nics |
    ForEach-Object {
        $Nic = $_
        foreach ($IP in $Nic.IPAddress)
        {
            if ($IP.Contains('.'))
            {
                $networkObj = New-Object -TypeName System.Object

                $vlanId = VMware.VimAutomation.Core\Get-
VirtualPortGroup | Where-Object {$_.Key -eq $Nic.NetworkName}
                $networkObj | Add-Member -MemberType NoteProperty
-Name VlanID -Value $vlanId
                $networkObj | Add-Member -MemberType NoteProperty
-Name IPv4Address -Value $IP

                $null = $vmNetworkArray.Add($networkObj)
            }
        }
    }

    $vmObj | Add-Member -MemberType NoteProperty -Name PrimaryIPv4
-Value $vmNetworkArray[0].IPv4Address
    $vmObj | Add-Member -MemberType NoteProperty -Name
PrimaryVlanID -Value $vmNetworkArray.VlanID
    $vmObj | Add-Member -MemberType NoteProperty -Name Networks
-Value $vmNetworkArray

    $guest = $vm.Guest
    $parts = $guest -split ":"
    $afterColon = $parts[1]

    $osFullName = $afterColon

    $vmObj | Add-Member -MemberType NoteProperty -Name OSFullName
-Value $osFullName
    $vmObj | Add-Member -MemberType NoteProperty -Name GuestID
-Value $vm.GuestId
    }
}

$vmObjArray | Format-Table -Property ID, Name, NumCpu, MemoryGB,
PrimaryHardDisk, PrimaryIPv4, PrimaryVlanID, GuestID, OSFullName, Firmware

```

```

    $Script:VMObjList = $vmObjArray
}

function PowerOffVMs {
    Write-Host "`n"

    -----
    ----" -ForegroundColor Cyan
    Write-Host "Power Off VMs" -ForegroundColor Magenta
    Write-Host

    -----
    ----`n" -ForegroundColor Cyan
    foreach($vm in $VMObjList) {
        try {
            Write-Host "Powering Off VM $($vm.Name) in vCenter $($VCENTER
)"
            $null = VMware.VimAutomation.Core\Stop-VM -VM $vm.Name
            -Confirm:$false -ErrorAction Stop
            Write-Host "Powered Off VM $($vm.Name)" -ForegroundColor Green
        }
        catch {
            Write-Error "Failed to Power Off VM $($vm.Name). Error :
$._Exception.Message"
            Break;
        }
        Write-Host "`n"
    }
}

function CreateOntapVolumeSnapshot {
    Write-Host "`n"

    -----
    ----" -ForegroundColor Cyan
    Write-Host "Taking ONTAP Snapshot for Volume $ONTAP_VOLUME_NAME"
    -ForegroundColor Magenta
    Write-Host

    -----
    ----`n" -ForegroundColor Cyan

    Try {
        Write-Host "Taking snapshot for Volume $ONTAP_VOLUME_NAME"
        $timestamp = Get-Date -Format "yyyy-MM-dd_HH:mm:ss"
        $snapshot = New-NcSnapshot -VserverContext $VSERVER -Volume
$ONTAP_VOLUME_NAME -Snapshot "snap.script-$timestamp"

        if($snapshot) {
            Write-Host "Snapshot ""$($snapshot.Name)"" created for Volume

```

```

$ONTAP_VOLUME_NAME" -ForegroundColor Green
    $Script:OntapVolumeSnapshot = $snapshot
    }
} Catch {
    Write-Error "Failed to create snapshot for Volume
$ONTAP_VOLUME_NAME. Error : $_.Exception.Message"
    Break;
}
}

function Shift {
    Write-Host
    "-----"
    -----" -ForegroundColor Cyan
        Write-Host "VM Shift" -ForegroundColor Magenta
        Write-Host
    "-----"
    -----`n" -ForegroundColor Cyan

    $Script:HypervVMList = New-Object System.Collections.ArrayList
    foreach($vmObj in $VMObjList) {

        Write-Host "*****"
        Write-Host "Performing VM conversion for $($vmObj.Name)"
    -ForegroundColor Blue
        Write-Host "*****"

        $hypervVMObj = New-Object -TypeName System.Object

        $directoryName = "/vol/$( $ONTAP_VOLUME_NAME )/$( $VHDX_QTREE_NAME )
/$($vmObj.HardDisks[0].Folder) "

        try {
            Write-Host "Creating Folder ""$directoryName"" for VM $(
$vmObj.Name) "
            $dir = New-NcDirectory -VserverContext $VSERVER -Path
$directoryName -Permission 0777 -Type directory -ErrorAction Stop
            if($dir) {
                Write-Host "Created folder ""$directoryName"" for VM
$( $vmObj.Name )`n" -ForegroundColor Green
            }
        }
        catch {
            if($_.Exception.Message -eq "[500]: File exists") {
                Write-Warning "Folder ""$directoryName"" already exists!
`n"

```

```

    }
    Else {
        Write-Error "Failed to create folder ""$directoryName""
for VM $($vmObj.Name). Error : $($_.Exception.Message)"
        Break;
    }
}

$vmDiskArray = New-Object System.Collections.ArrayList

foreach($disk in $vmObj.HardDisks) {
    $vmDiskObj = New-Object -TypeName System.Object
    try {
        Write-Host "`nConverting $($disk.Name)"
        Write-Host "-----"

        $vmdkPath = "/vol/$(ONTAP_VOLUME_NAME)/$(disk.Folder) /
$(disk.FileName)"
        $fileName = $disk.FileName -replace '\.vmdk$', ''
        $vhdxPath = "$(directoryName)/$(fileName).vhdx"

        Write-Host "Converting ""$($disk.Name)"" VMDK path ""
$(vmdkPath)"" to VHDX at Path ""$(vhdxPath)"" for VM $($vmObj.Name)"
        $convert = ConvertTo-NcVhdx -SourceVmdk $vmdkPath
-DestinationVhdx $vhdxPath -SnapshotName $OntapVolumeSnapshot
-ErrorAction Stop -WarningAction SilentlyContinue
        if($convert) {
            Write-Host "Successfully converted VM ""$($vmObj.Name
)"" VMDK path ""$(vmdkPath)"" to VHDX at Path ""$(vhdxPath)""
-ForegroundColor Green

            $vmDiskObj | Add-Member -MemberType NoteProperty -Name
Name -Value $disk.Name
            $vmDiskObj | Add-Member -MemberType NoteProperty -Name
VHDXPath -Value $vhdxPath

            $null = $vmDiskArray.Add($vmDiskObj)
        }
    }
    catch {
        Write-Error "Failed to convert ""$($disk.Name)"" VMDK to
VHDX for VM $($vmObj.Name). Error : $($_.Exception.Message)"
        Break;
    }
}

```

```

    $hypervVMObj | Add-Member -MemberType NoteProperty -Name Name
-Value $vmObj.Name
    $hypervVMObj | Add-Member -MemberType NoteProperty -Name HardDisks
-Value $vmDiskArray
    $hypervVMObj | Add-Member -MemberType NoteProperty -Name MemoryGB
-Value $vmObj.MemoryGB
    $hypervVMObj | Add-Member -MemberType NoteProperty -Name Firmware
-Value $vmObj.Firmware
    $hypervVMObj | Add-Member -MemberType NoteProperty -Name GuestID
-Value $vmObj.GuestID

    $null = $HypervVMList.Add($hypervVMObj)
    Write-Host "`n"

}
}

function ConfigureVMsOnHyperV {
    Write-Host
    "-----"
    "-----" -ForegroundColor Cyan
    Write-Host "Configuring VMs on Hyper-V" -ForegroundColor Magenta
    Write-Host
    "-----"
    "-----`n" -ForegroundColor Cyan

    foreach($vm in $HypervVMList) {
        try {

            # Define the original path
            $originalPath = $vm.HardDisks[0].VHDXPath
            # Replace forward slashes with backslashes
            $windowsPath = $originalPath -replace "/", "\"

            # Replace the initial part of the path with the Windows drive
            letter
            $windowsPath = $windowsPath -replace "^\\vol\\", "\\
$( $ONTAP_NETWORK_SHARE_ADDRESS )"

            $vmGeneration = if ($vm.Firmware -eq "bios") {1} else {2};

            Write-Host "*****"
            Write-Host "Creating VM $($vm.Name)" -ForegroundColor Blue
            Write-Host "*****"
            Write-Host "Creating VM $($vm.Name) with Memory $($vm.

```

```

MemoryGB)GB, vSwitch $($HYPERV_VM_SWITCH), $($vm.HardDisks[0].Name) ""
 $($windowsPath)"" , Generation $($vmGeneration) on Hyper-V"

    $createVM = Hyper-V\New-VM -Name $vm.Name -VHDPATH
$windowsPath -SwitchName $HYPERV_VM_SWITCH -MemoryStartupBytes (Invoke-
Expression "$($vm.MemoryGB)GB") -Generation $vmGeneration -ErrorAction
Stop

    if($createVM) {
        Write-Host "VM $($createVM.Name) created on Hyper-V host
`n" -ForegroundColor Green

        $index = 0
        foreach($vmDisk in $vm.HardDisks) {
            $index++
            if ($index -eq 1) {
                continue
            }

            Write-Host "`nAttaching $($vmDisk.Name) for VM $($vm
.Name) "

            Write-Host
"-----"

            $originalPath = $vmDisk.VHDXPath

            # Replace forward slashes with backslashes
            $windowsPath = $originalPath -replace "/", "\"

            # Replace the initial part of the path with the
Windows drive letter
            $windowsPath = $windowsPath -replace "^\\vol\\", "\\
 $($ONTAP_NETWORK_SHARE_ADDRESS)\\"

            try {
                $attachDisk = Hyper-v\Add-VMHardDiskDrive -VMName
$vm.Name -Path $windowsPath -ErrorAction Stop
                Write-Host "Attached $($vmDisk.Name) ""
 $($windowsPath)"" to VM $($vm.Name) " -ForegroundColor Green
            }
            catch {
                Write-Error "Failed to attach $($vmDisk.Name)
 $($windowsPath) to VM $($vm.Name): Error : $($_.Exception.Message) "
                Break;
            }
        }
    }

```

```

        if($vmGeneration -eq 2 -and $vm.GuestID -like "*rhel*") {
            try {
                Write-Host "`nDisabling secure boot"
                Hyper-V\Set-VMFirmware -VMName $createVM.Name
                -EnableSecureBoot Off -ErrorAction Stop
                Write-Host "Secure boot disabled" -ForegroundColor
Green
            }
            catch {
                Write-Error "Failed to disable secure boot for VM
${$createVM.Name}. Error : ${$.Exception.Message}"
            }
        }

        try {
            Write-Host "`nStarting VM ${$createVM.Name}"
            Hyper-v\Start-VM -Name $createVM.Name -ErrorAction
Stop
            Write-Host "Started VM ${$createVM.Name}`n"
            -ForegroundColor Green
        }
        catch {
            Write-Error "Failed to start VM ${$createVM.Name}.
Error : ${$.Exception.Message}"
            Break;
        }
    }
}
catch {
    Write-Error "Failed to create VM ${$vm.Name} on Hyper-V.
Error : ${$.Exception.Message}"
    Break;
}
}
}

main

```


版权信息

版权所有 © 2024 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。