



在NetApp存储上部署 Hyper-V NetApp virtualization solutions

NetApp
February 13, 2026

This PDF was generated from <https://docs.netapp.com/zh-cn/netapp-solutions-virtualization/hyperv/hyperv-deploy.html> on February 13, 2026. Always check docs.netapp.com for the latest.

目录

在NetApp存储上部署 Hyper-V	1
了解如何使用ONTAP存储系统部署 Microsoft Hyper-V	1
概述	1
受众	1
架构	1
使用情形概要	1
准备利用ONTAP存储系统部署 Microsoft Hyper-V	1
部署过程的先决条件	1
适用于 Microsoft Hyper-V 和ONTAP存储系统的部署指南	6
正确调整存储大小	7
增强虚拟机性能	7
SMB3.0的设计与考量	7
配置 SMB 卷	8
区块协议设计与考量	8
配置 iSCSI 卷	9
使用 ODX 功能快速配置虚拟磁盘	11
性能优化	11
SMB 卷和 CSV 大小	11
迁移	11
在NetApp存储上部署 Microsoft Hyper-V	12
使用NetApp存储快照进行恢复	12
使用第三方解决方案备份和恢复	12
高级NetApp ONTAP选项	18
ONTAP存储系统上的 Microsoft Hyper-V 部署摘要	18
使用 PowerShell 脚本将虚拟机迁移到 Microsoft Hyper-V	18
Powershell 脚本	18

在NetApp存储上部署 Hyper-V

了解如何使用ONTAP存储系统部署 Microsoft Hyper-V

通过 Windows Server 的 Hyper-V 角色实现 Microsoft 的计算机虚拟化。了解如何使用ONTAP存储系统和 Windows Server 功能创建和管理虚拟化计算环境。

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

概述

Hyper-V 角色使我们能够使用 Windows Server 内置的虚拟化技术来创建和管理虚拟化计算环境。Hyper-V 技术将硬件虚拟化，从而提供一个可以在一台物理计算机上同时运行多个操作系统的环境。Hyper-V 使您能够创建和管理虚拟机及其资源。每个虚拟机都是一个独立的、虚拟化的计算机系统，可以运行自己的操作系统。Hyper-V 提供虚拟化应用程序和工作负载的基础架构，支持旨在提高效率和降低成本的各种业务目标，是 VMware vSphere 的完美替代方案，尤其是在当前市场条件下组织寻求多个虚拟机管理程序共存的情况下。

受众

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。本文档的目标读者包括销售工程师、现场顾问、专业服务、IT 经理、合作伙伴工程师以及希望将 Hyper-V 部署为主要或备用虚拟机管理程序的客户。

架构

本文档中描述的架构具体包括 Microsoft Windows Server 2022 和 Hyper-V 虚拟化。NetApp强烈建议在每次部署中都采用虚拟化软件和基础设施管理软件。该配置使用每个组件的最佳实践来实现可靠的企业级基础设施。

使用情形概要

本文档介绍了使用NetApp全闪存FAS和ASA阵列模型设置 Hyper-V 集群以便在 Microsoft Windows Server 2022 上作为工作负载实现最佳性能的部署过程和最佳实践。服务器操作系统/虚拟机管理程序是 Microsoft Windows Server 2022。该指南涵盖通过存储区域网络 (SAN) 和网络附加存储 (NAS) 协议提供数据的NetApp存储系统。

准备利用ONTAP存储系统部署 Microsoft Hyper-V

准备您的环境以部署带有ONTAP存储系统的 Microsoft Hyper-V 集群。此过程包括安装 Windows Server 功能、为 Hyper-V 流量配置网络接口、决定适当的存储设计、安装 iSCSI 主机实用程序、配置 Windows iSCSI 启动器以及创建故障转移群集。

部署过程的先决条件

- 所有硬件必须针对您正在运行的 Windows Server 版本进行认证，并且完整的故障转移群集解决方案必须通过验证配置向导中的所有测试
- Hyper-V 节点加入域控制器（推荐）并彼此之间建立适当的连接。

- 每个 Hyper-V 节点都应进行相同的配置。
- 每个 Hyper-V 服务器上配置的网络适配器和指定的虚拟交换机用于隔离管理、iSCSI、SMB、实时迁移的流量。
- 每个 Hyper-V 服务器上都启用了故障转移群集功能。
- SMB 共享或 CSV 用作共享存储，用于存储 Hyper-V 集群的虚拟机及其磁盘。
- 不同集群之间不应共享存储。每个集群规划一个或多个 CSV/CIFS 共享。
- 如果将 SMB 共享用作共享存储，则必须配置 SMB 共享上的权限以授予集群中所有 Hyper-V 节点的计算机帐户访问权限。

有关详细信息，请参阅：For more information, see:

- ["Windows Server 上的 Hyper-V 系统要求"](#)
- ["验证故障转移群集的硬件"](#)
- ["部署 Hyper-V 群集"](#)

安装 Windows 功能

以下步骤介绍如何安装所需的 Windows Server 2022 功能。

所有主持人

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

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

配置网络

适当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群集的标准建议。通过添加虚拟网络适配器、交换机嵌入式组合 (SET) 以及引入 Hyper-V QoS 等功能，可以将网络流量压缩到更少的物理适配器上。设计网络配置时要考虑服务质量、冗余和流量隔离。结合流量隔离技术配置 VLAN 等网络隔离技术可为流量和服务质量提供冗余，从而改善并增加存储流量性能的一致性。

建议使用多个逻辑和/或物理网络来分离和隔离特定的工作负载。通常分为几段的典型网络流量示例如下：

- iSCSI 存储网络。
- CSV（集群共享卷）或心跳网络。
- 实时迁移
- VM 网络
- 管理网络

注意：当 iSCSI 与专用 NIC 一起使用时，不建议使用任何组合解决方案，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 组合。

有关更多信息，请参阅["在 Windows Server 中规划 Hyper-V 网络"](#)

确定 Hyper-V 的存储设计

Hyper-V支持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，可用作虚拟机的本机存储 - 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 作为来宾连接存储选项。ONTAP为需要混合协议访问的工作负载提供了统一存储（全闪存阵列）的灵活选项，并为仅 SAN 配置提供了 SAN 优化存储（全 SAN 阵列）。

使用 SMB3 还是 iSCSI/FC 的决定取决于现有的基础设施，SMB3/iSCSI 允许客户使用现有的网络基础设施。对于拥有现有 FC 基础设施的客户，可以利用该基础设施并将存储作为基于 FC 的集群共享卷呈现。

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

- 托管在持续可用的 SMB 3.0 共享上的虚拟机
- 托管在 iSCSI 或 FC 上运行的群集共享卷 (CSV) LUN 上的虚拟机
- 客户机内存存储并将磁盘传递至客户虚拟机

注：无论平台或操作系统如何，精简配置、重复数据删除、压缩、数据压缩、弹性克隆、快照和复制等核心ONTAP功能均可在后台无缝运行，并为 Hyper-V 工作负载提供巨大价值。这些功能的默认设置对于 Windows Server 和 Hyper-V 来说是最佳的。

注意：如果虚拟机有多条路径可用，并且已安装和配置多路径 I/O 功能，则使用来宾启动器在来宾虚拟机上支持 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，NVMe/FC 和 NVMe/TCP 不受 Microsoft 支持。

安装NetApp Windows iSCSI 主机实用程序

以下部分介绍如何执行NetApp Windows iSCSI Host Utilities 的无人值守安装。有关安装的详细信息，请参阅["安装 Windows Unified Host Utilities 7.2（或最新支持的版本）"](#)

所有主持人

1. 下载["Windows iSCSI 主机实用程序"](#)
2. 解除对下载文件的阻止。

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

3. 安装主机实用程序。

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

注意：在此过程中系统将重新启动。

配置 Windows 主机 iSCSI 启动器

以下步骤介绍如何配置内置的 Microsoft iSCSI 启动器。

所有主持人

1. 右键单击任务栏中的 PowerShell 图标并选择以管理员身份运行，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. 将所有新认领的设备的默认负载平衡策略设置为循环。

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

6. 为每个控制器配置一个 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>
```

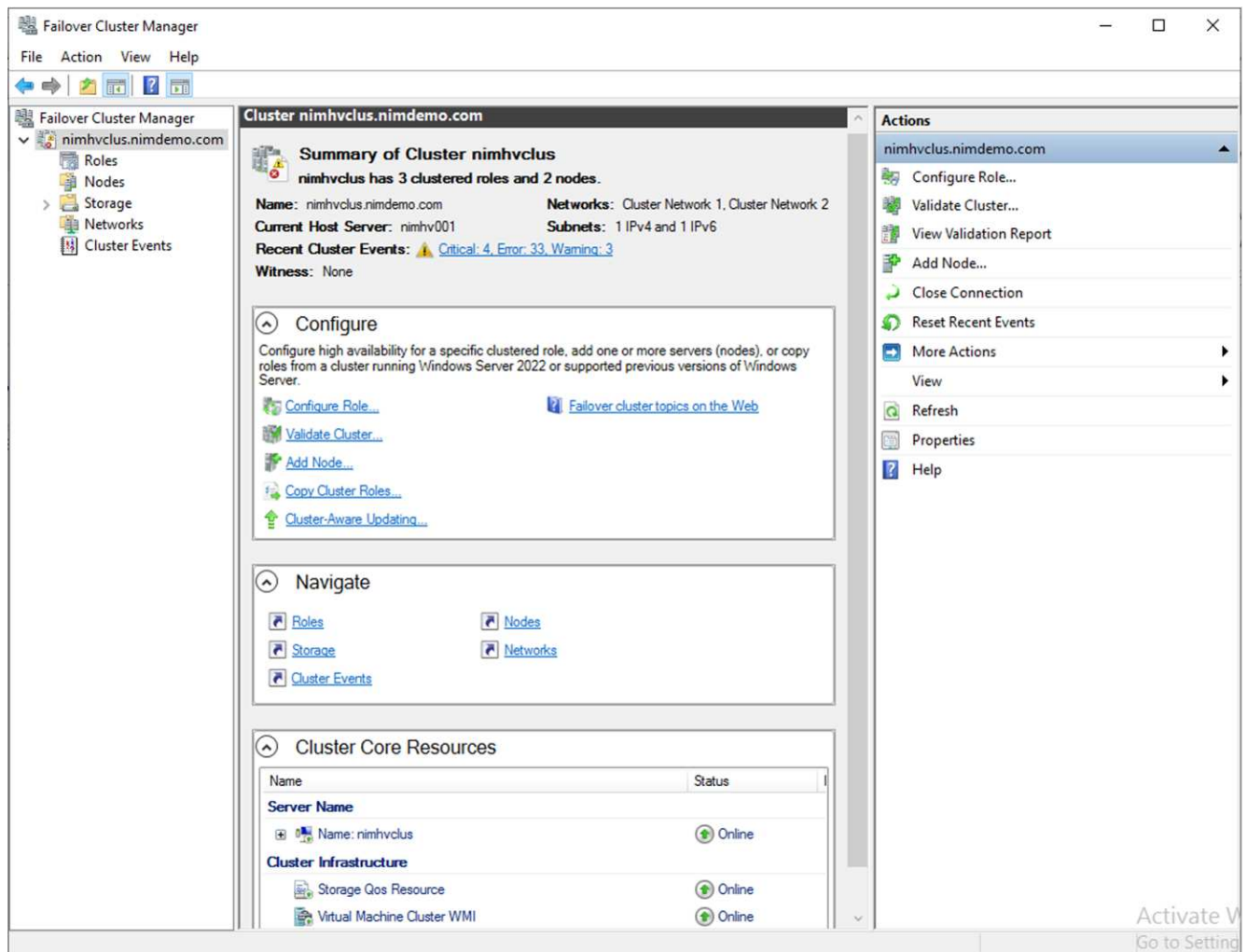
注意：添加多个会话（至少 5-8 个）以提高性能并利用带宽。

创建集群

仅限一台服务器

1. 右键单击 PowerShell 图标并选择，以管理员权限启动 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. 为实时迁移选择合适的集群网络。

4. 指定 CSV 网络。

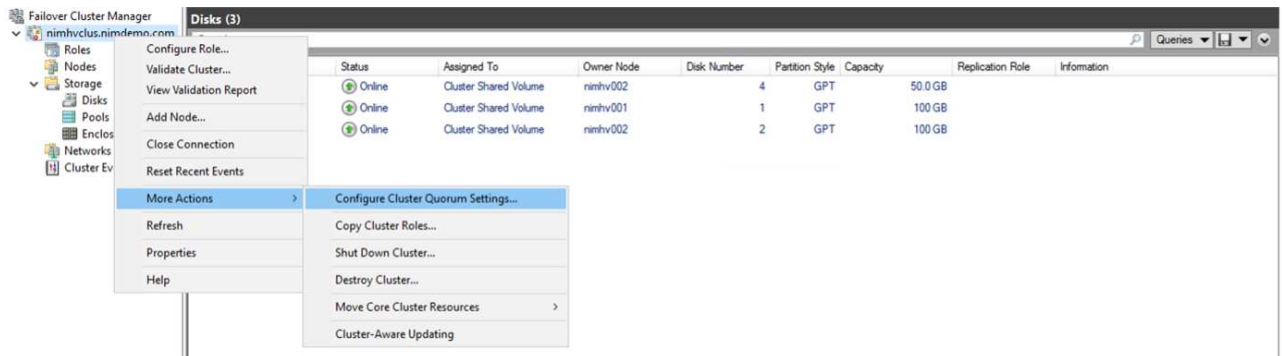
```
(Get-ClusterNetwork -Name Cluster).Metric = 900
```

5. 更改群集以使用仲裁磁盘。

a. 右键单击 PowerShell 图标并选择“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. 在故障转移群集管理器中，选择 Configure Cluster Quorum Settings。



c. 在欢迎页面中单击下一步。

d. 选择仲裁见证并单击“下一步”。

e. 选择“配置磁盘见证”，然后单击“下一步”。

f. 从可用存储中选择磁盘 W：，然后单击下一步。

g. 在确认页面上单击“下一步”，然后在摘要页面上单击“完成”。

有关法定人数和见证人的详细信息，请参阅["配置和管理仲裁"](#)

6. 从故障转移群集管理器运行群集验证向导来验证部署。

7. 创建 CSV LUN 来存储虚拟机数据，并通过故障转移群集管理器中的角色创建高可用性虚拟机。

适用于 Microsoft Hyper-V 和 ONTAP 存储系统的部署指南

为了确保在使用 ONTAP 存储部署 Microsoft Hyper-V 时获得最佳性能和可靠性，请考虑工作负载兼容性、存储大小和 VM 资源分配等因素。兼容性检查应包括操作系统版本、应用程序、数据库和任何现有定制，以确保在 Hyper-V 环境中顺利运行。

正确调整存储大小

在部署工作负载或从现有虚拟机管理程序迁移之前，请确保工作负载的大小能够满足所需的性能。这可以通过收集每个单独的 VM 的性能数据轻松完成，这些数据收集 CPU（已使用/已配置）、内存（已使用/已配置）、存储（已配置/已利用）、网络吞吐量和延迟的统计信息以及读/写 IOP、吞吐量和块大小的聚合。这些参数对于成功部署以及正确确定存储阵列和工作负载主机的大小是必需的。

注意：在为 Hyper-V 和相关工作负载调整存储大小时，请规划 IOPS 和容量。

注意：对于 I/O 密集型虚拟机或需要大量资源和容量的虚拟机，请分离操作系统和数据磁盘。操作系统和应用程序二进制文件很少更改，并且卷崩溃一致性是可以接受的。

注意：使用来宾连接存储（又名来宾内）来获得高性能数据磁盘，而不是使用 VHD。这也有助于简化克隆过程。

增强虚拟机性能

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单个虚拟 SCSI 控制器。仍然建议使用固定 VHDx 作为部署虚拟磁盘的主要选择，并且对使用任何类型的 VHDx 虚拟磁盘没有任何限制。

注意：避免在 Windows Server 上安装不需要的角色。

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 架构作为启动级别，这显著提高了整体 VM 性能。

注意：避免频繁进行检查点，因为这会对虚拟机的性能产生负面影响。

SMB3.0的设计与考量

SMB 3.0 文件共享可用作 Hyper-V 的共享存储。ONTAP通过 SMB 共享为 Hyper-V 进行无中断操作。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 (VHD) 文件。使用专用ONTAP CIFS SVM 为 Hyper-V 提供基于 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。如果有 10GB 网络，建议使用 10GB 网络建立 Hyper-V 主机与NetApp阵列之间的连接。对于 1GB 网络连接，NetApp建议创建一个由多个 1GB 端口组成的接口组。将每个为 SMB 多通道提供服务的 NIC 连接到其专用 IP 子网，以便每个子网在客户端和服务端之间提供单一路径。

要点

- 在ONTAP SVM 上启用 SMB 多通道
- ONTAP CIFS SVM 在集群中的每个节点上应至少有一个数据 LIF。
- 所使用的共享必须配置持续可用的属性集。
- ONTAP One 现在包含在每个AFF（A 系列和 C 系列）、全 SAN 阵列 (ASA) 和FAS系统中。因此不需要单独的许可证。
- 对于共享 VHDx，使用来宾连接的 iSCSI LUN

注意：ODX 受支持并可跨协议工作。在文件共享和 iSCSI 或 FCP 连接的 LUN 之间复制数据也利用 ODX。

注意：集群中节点的时间设置应相应设置。如果NetApp CIFS 服务器必须参与 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。较小的数据包可能会导致性能下降。

配置 SMB 卷

1. 验证存储虚拟机 (SVM) 上是否启用了所需的 CIFS 服务器选项
2. 以下选项应设置为 true：smb2-enabled、smb3-enabled、copy-offload-enabled、shadowcopy-enabled、is-multichannel-enabled、is-large-mtu-enabled

```

HV_NestedCluster::> vserver 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
vserver      smb2-enabled smb3-enabled copy-offload-enabled shadowcopy-enabled is-multichannel-enabled is-large-mtu-enabled
NestedHvsvm01 true         true         true         true         true         true

```

3. 在存储虚拟机 (SVM) 上创建 NTFS 数据卷，然后配置持续可用的共享以供 Hyper-V 使用

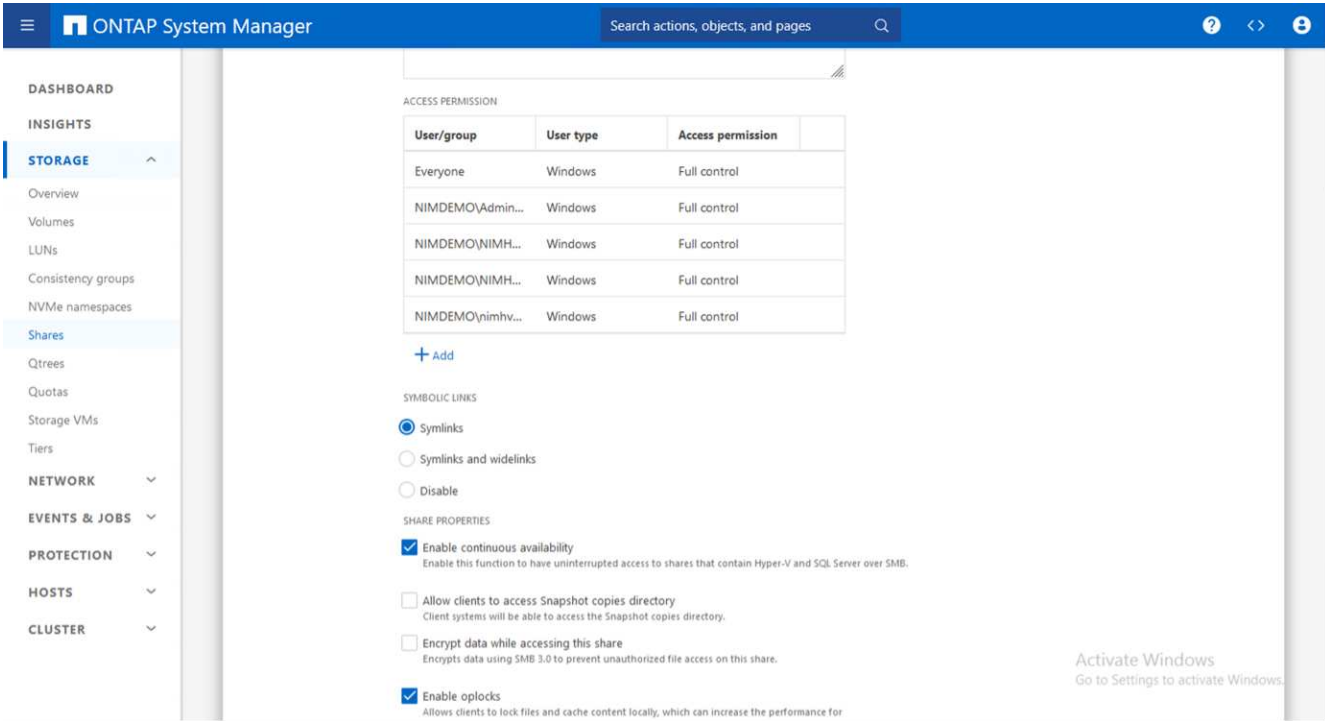
```

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：除非配置中使用的卷创建为 NTFS 安全样式卷，否则 Hyper-V 通过 SMB 的无中断操作无法正常工作。

4. 启用持续可用性并在共享上配置 NTFS 权限以包含具有完全控制权的 Hyper-V 节点。



有关详细的最佳做法指导，请参阅["Hyper-V 的部署指南和最佳实践"](#)。

有关更多信息，请参阅["Hyper-V over SMB 的 SMB 服务器和卷要求"](#)。

区块协议设计与考量

要点

- 在主机上使用多路径（MPIO）来管理多条路径。根据需要创建更多路径，以促进数据移动操作或利用额外

的 I/O 资源，但不要超过主机操作系统可以支持的最大路径数。

- 在访问 LUN 的主机上安装主机实用程序工具包。
- 至少创建 8 个卷。

注意：每个卷使用一个 LUN，因此 LUN 与 CSV 的映射比率为 1:1。

- SVM 应该在每个要使用 iSCSI 或光纤通道提供数据的存储控制器上为每个以太网网络或光纤通道结构配备一个 LIF。
- 使用 FCP 或 iSCSI 提供数据的 SVM 需要 SVM 管理接口。

配置 iSCSI 卷

要配置 iSCSI 卷，请确保满足以下先决条件。

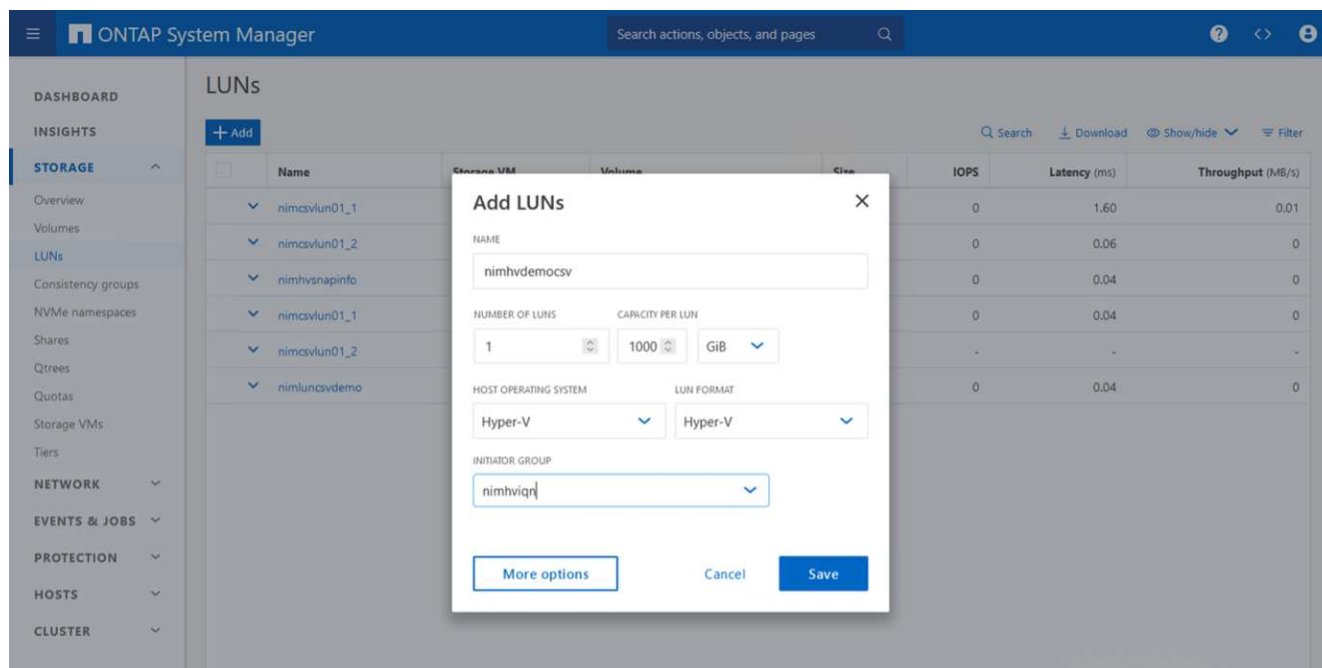
- 存储虚拟机 (SVM) 应启用 iSCSI 协议并创建适当的逻辑接口 (LIF)。
- 指定的聚合必须具有足够的可用空间来包含 LUN。

注：默认情况下，ONTAP 使用选择性 LUN 映射 (SLM) 使 LUN 只能通过拥有该 LUN 的节点及其高可用性 (HA) 伙伴节点上的路径访问。

- 在每个节点上配置所有 iSCSI LIF 以实现 LUN 移动性，以防 LUN 移动到集群中的另一个节点。

步骤

1. 使用系统管理器并导航到 LUN 窗口（ONTAP CLI 可用于相同操作）。
2. 单击“创建”。
3. 浏览并选择要创建 LUN 的指定 SVM，然后显示创建 LUN 向导。
4. 在“常规属性”页面上，为包含 Hyper-V 虚拟机的虚拟硬盘 (VHD) 的 LUN 选择 Hyper-V。



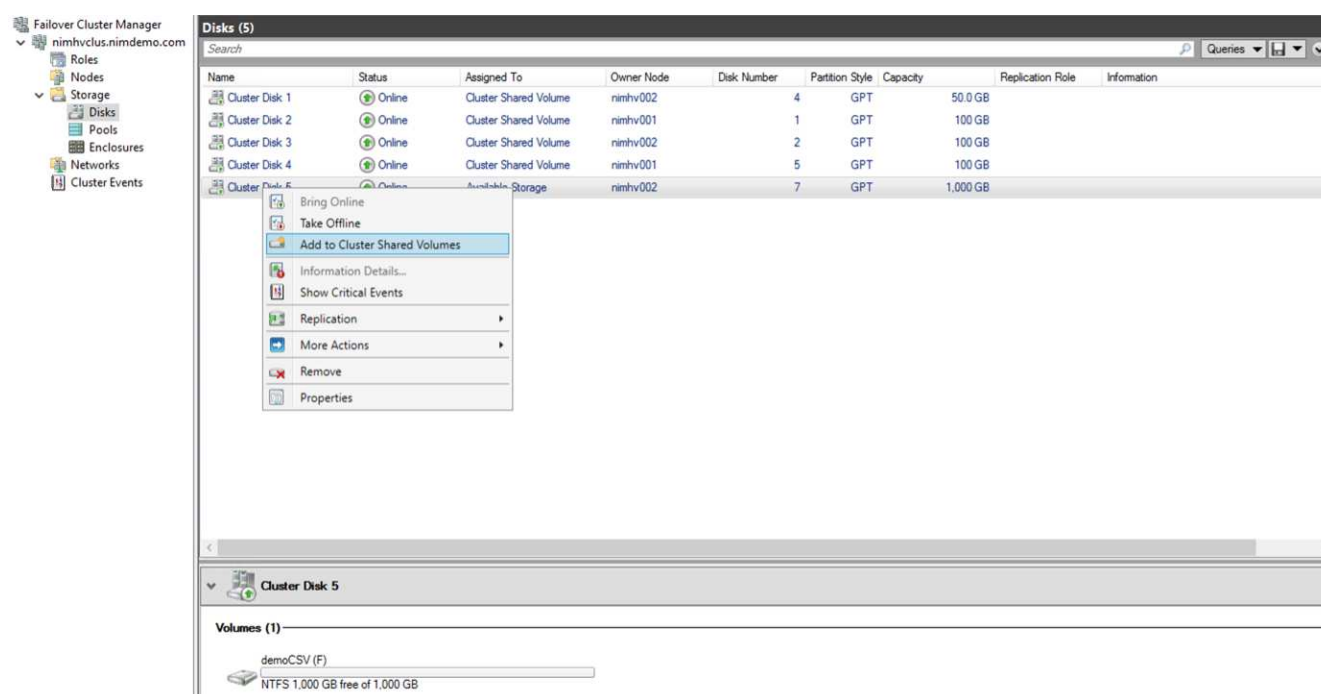
5. <单击更多选项> 在 LUN 容器页面上，选择一个现有的FlexVol volume，否则将创建一个新卷。
6. <单击更多选项> 在启动器映射页面上，单击添加启动器组，在常规选项卡上输入所需信息，然后在启动器选项卡上输入主机的 iSCSI 启动器节点名称。
7. 确认详细信息，然后单击“完成”以完成向导。

创建 LUN 后，转到故障转移群集管理器。要将磁盘添加到 CSV，必须先将该磁盘添加到群集的可用存储组（如果尚未添加），然后将该磁盘添加到群集上的 CSV。

注意：故障转移群集中默认启用 CSV 功能。

将磁盘添加到可用存储空间：

1. 在故障转移群集管理器的控制台树中，展开群集的名称，然后展开“存储”。
2. 右键单击“磁盘”，然后选择“添加磁盘”。出现一个列表，显示可以添加用于故障转移群集的磁盘。
3. 选择要添加的一个或多个磁盘，然后选择“确定”。
4. 磁盘现已分配给可用存储组。
5. 完成后，选择刚刚分配给可用存储的磁盘，右键单击选择，然后选择添加到群集共享卷。



6. 磁盘现已分配给集群中的集群共享卷组。磁盘作为 %SystemDrive%\ClusterStorage 文件夹下的编号卷（挂载点）向每个集群节点公开。这些卷出现在 CSVFS 文件系统中。

有关更多信息，请参阅[“在故障转移群集中使用群集共享卷”](#)。

创建高可用性虚拟机：

要创建高可用性虚拟机，请按照以下步骤操作：

1. 在故障转移群集管理器中，选择或指定所需的群集。确保群集下的控制台树已展开。

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”。出现新的虚拟机向导。单击“下一步”。
4. 在指定名称和位置页面上，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“安装选项”页面上，单击从启动 CD/DVD-ROM 安装操作系统。在“媒体”下，指定媒体的位置，然后单击“完成”。
8. 虚拟机已创建。然后，故障转移群集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

ONTAP 中的 ODX 功能允许通过简单地复制 ONTAP 存储系统托管的主 VHDX 文件来制作主 VHDX 的副本。由于支持 ODX 的副本不会将任何数据放在网络上，因此复制过程发生在 NetApp 存储端，因此速度可提高六到八倍。快速配置的一般考虑包括存储在文件共享上的主系统准备映像和由 Hyper-V 主机启动的常规复制过程。

注意：ONTAP 支持 SMB 和 SAN 协议的 ODX。

注意：要利用 Hyper-V 的 ODX 复制卸载直通用例，来宾操作系统必须支持 ODX，并且来宾操作系统的磁盘必须是由支持 ODX 的存储（SMB 或 SAN）支持的 SCSI 磁盘。客户操作系统上的 IDE 磁盘不支持 ODX 直通。

性能优化

虽然每个 CSV 的推荐虚拟机数量是主观的，但有许多因素决定了可以放置在每个 CSV 或 SMB 卷上的虚拟机的最佳数量。尽管大多数管理员只考虑容量，但发送到 VHDx 的并发 I/O 量是整体性能的最关键因素之一。控制性能的最简单方法是调节每个 CSV 或共享上放置的虚拟机的数量。如果并发虚拟机 I/O 模式向 CSV 或共享发送过多流量，则磁盘队列将被填满，并产生更高的延迟。

SMB 卷和 CSV 大小

确保解决方案端到端大小足够以避免瓶颈，并且在为 Hyper-V VM 存储目的创建卷时，最佳做法是创建不大于所需大小的卷。正确调整卷大小可防止意外在 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) 支持一个或多个 VM。放置在 CSV 上的虚拟机数量取决于工作负载和业务偏好，以及如何使用 ONTAP 存储功能（例如快照和复制）。在大多数部署场景中，将多个虚拟机放置在 CSV 上是一个很好的起点。针对具体用例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 VM 需要额外的容量，则无需将 CSV 的大小设置得大于所需。Diskpart 可用于扩展 CSV 大小，或者更简单的方法是创建一个新的 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 的数量，而不是增加其大小作为临时措施。

迁移

当前市场条件下最常见的用例之一是迁移。客户可以使用 VMM Fabric 或其他第三方迁移工具来迁移虚拟机。这些工具使用主机级复制将数据从源平台移动到目标平台，这可能会很耗时，具体取决于迁移范围内的虚拟机数量。

在这种情况下使用 ONTAP 可以比使用基于主机的迁移过程实现更快的迁移。ONTAP 还支持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（在本例中为 ESXi 到 Hyper-V）。在 NetApp Storage 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。它还处理目标和目标虚拟机的创建和配置。

此过程有助于最大限度地减少停机时间并提高业务生产力。它还通过降低许可成本、锁定和对单一供应商的承诺来提供选择和灵活性。对于希望优化 VM 许可成本和扩展 IT 预算的组织来说，这也大有裨益。

以下视频演示了将虚拟机从 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 进行迁移的更多信息，请参阅["用于迁移的 PowerShell 脚本"](#)。

在NetApp存储上部署 Microsoft Hyper-V

使用基于ONTAP存储的解决方案和第三方备份集成部署 Microsoft Hyper-V 虚拟机。此过程包括使用ONTAP Snapshot 副本和FlexClone技术进行快速备份和恢复操作、配置 CommVault IntelliSnap 进行企业备份管理以及实施SnapMirror复制以进行跨站点的备份和灾难恢复。

学习解决独特的 Hyper-V 备份考虑因素（例如集群环境中的磁盘 ID 冲突）并优化独立主机和 Hyper-V 集群的数据保护。

使用NetApp存储快照进行恢复

备份虚拟机并快速恢复或克隆它们是ONTAP卷的一大优势。使用 Snapshot 副本快速制作虚拟机甚至整个 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安装在 QA、暂存和开发环境中时，就可以处理生产数据，而不会有数据损坏的风险。FlexClone卷可用于制作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。

请记住，Hyper-V 节点为每个磁盘分配一个唯一的 ID，并且对具有相应分区（MBR 或 GPT）的卷进行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）。对于独立的 Hyper-V 主机，可以轻松安装FlexClone卷而不会发生任何冲突。这是因为独立的 Hyper-V 服务器可以自动检测重复的磁盘 ID 并在无需用户干预的情况下动态更改它们。该方法可用于根据场景需求通过复制 VHD 来恢复 VM。

虽然对于独立的 Hyper-V 主机来说很简单，但对于 Hyper-V 集群来说，过程有所不同。恢复过程包括将FlexClone卷映射到独立的 Hyper-V 主机或使用 diskpart 通过将FlexClone卷映射到独立的 Hyper-V 主机来手动更改签名（这很重要，因为磁盘 ID 冲突会导致无法使磁盘联机）完成后，将FlexClone卷映射到集群。

使用第三方解决方案备份和恢复

注意：本节使用 Commvault，但这也适用于其他第三方解决方案。

CommVault IntelliSnap 利用ONTAP快照技术，创建基于硬件的 Hyper-V 快照。备份可以根据 Hyper-V 虚拟机管理程序或虚拟机组的配置自动执行，也可以针对虚拟机组或特定虚拟机手动执行。IntelliSnap 能够快速保护 Hyper-V 环境，最大限度地减少生产虚拟化场的负载。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大量虚拟机和数据存储的备份。细粒度访问提供从存储的二级层恢复单个文件和文件夹以及完整的来宾 .vhd 文件。

在配置虚拟化环境之前，部署需要快照与阵列集成的适当代理。Microsoft Hyper-V 虚拟化环境需要以下代理：

- 媒体代理

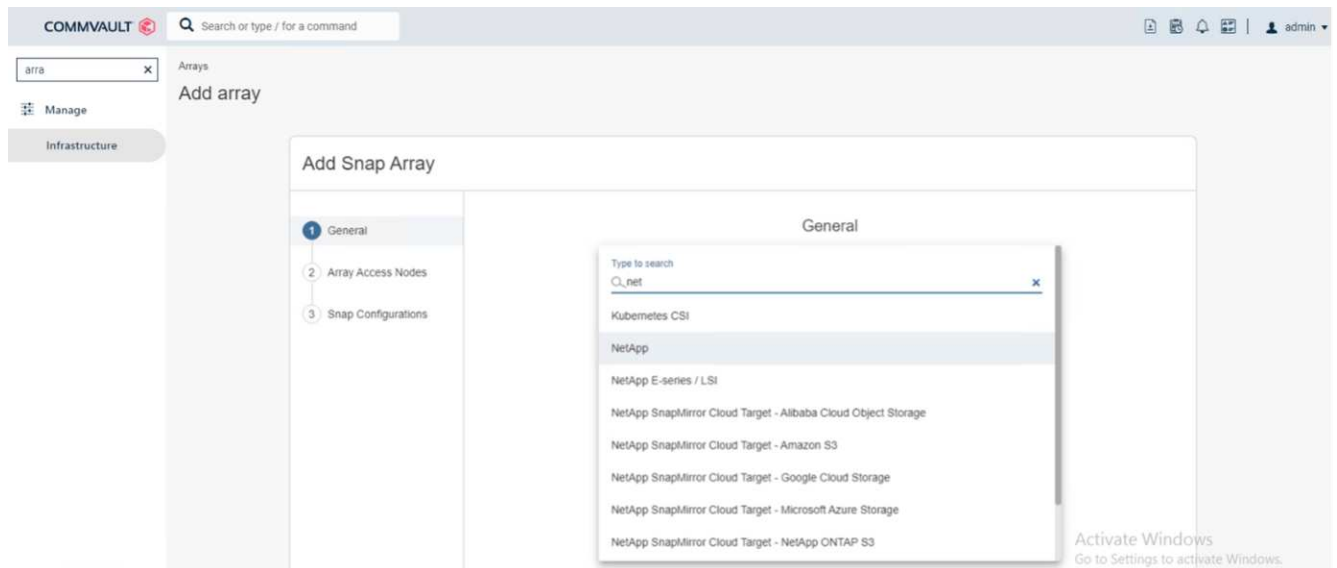
- 虚拟服务器代理 (VSA)
- VSS 硬件提供商 (Windows Server 2012 及更新操作系统)

使用阵列管理配置NetApp阵列

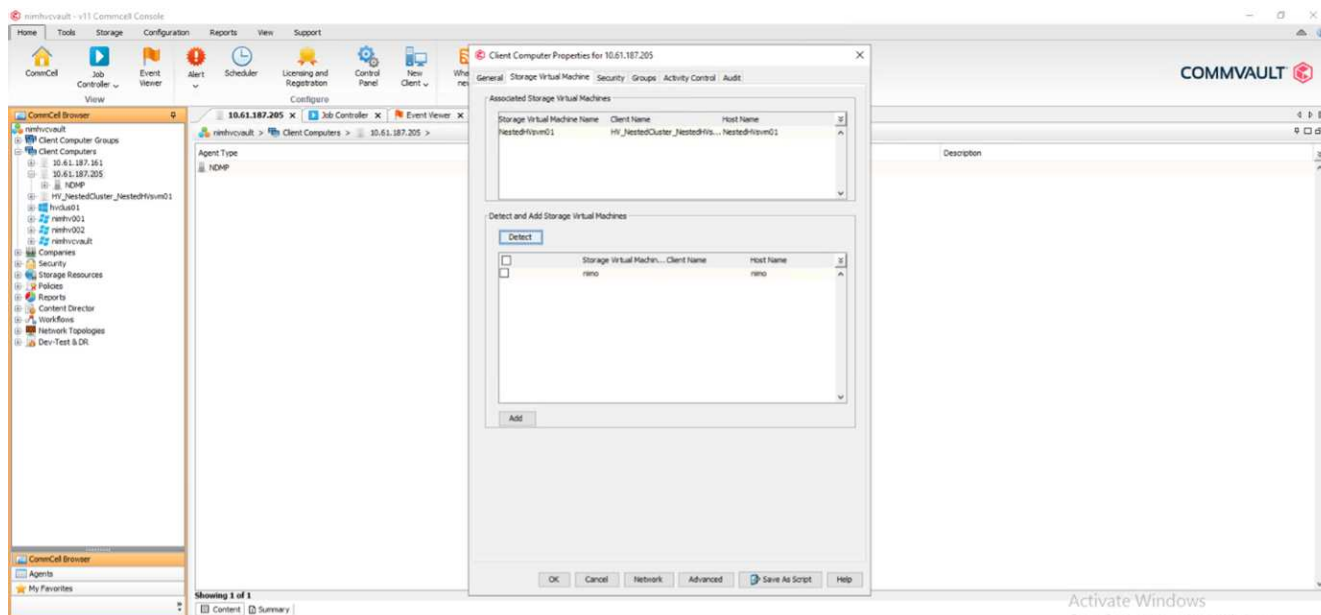
以下步骤显示如何在利用ONTAP阵列和 Hyper-V 的环境中配置 IntelliSnap 虚拟机备份。

1. 在 CommCell Console 的功能区上，单击“存储”选项卡，然后单击“阵列管理”。
2. 出现阵列管理对话框。
3. 单击“添加”。

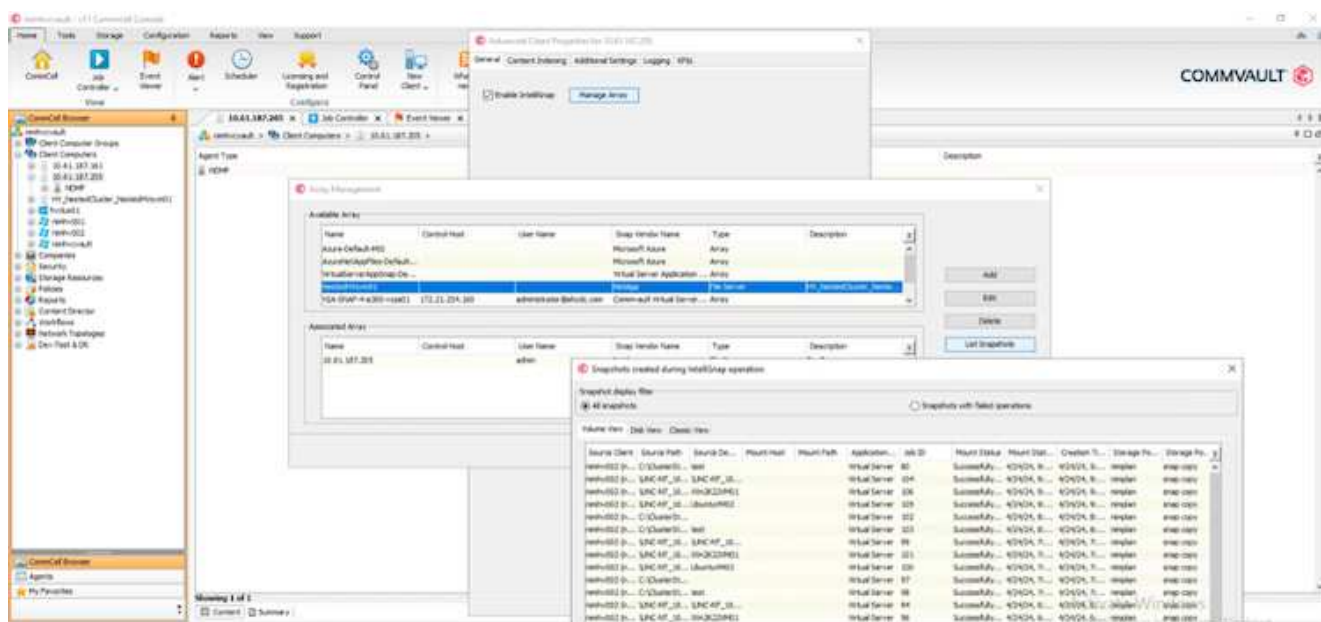
出现阵列属性对话框。



4. 在常规选项卡上，指定以下信息：
5. 从 Snap 供应商列表中，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“OK”。
10. <强制步骤> 完成后，还要在NetApp存储阵列上配置 SVM，使用检测选项自动检测存储虚拟机 (SVM)，然后选择一个 SVM，并使用添加选项将该 SVM 添加到 CommServe 数据库中，作为阵列管理条目。



11. 单击“高级”（如下图所示）并选中“启用 IntelliSnap”复选框。



有关配置数组的详细步骤，请参阅 ["配置NetApp阵列"](#) 和 ["在NetApp阵列上配置存储虚拟机"](#)

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 VM 组。

先决条件

- 虚拟机管理程序可以是 Hyper-V 群集、群集中的 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）。
- 确定将安装虚拟服务器代理 (VSA) 的一个或多个节点，以创建用于备份和还原操作的访问节点（VSA 代理

）。要发现 Hyper-V 服务器，CommServe 系统必须安装 VSA。

- 要对 Hyper-V 2012 R2 使用更改块跟踪，请选择 Hyper-V 群集中的所有节点。

以下步骤显示如何添加 Hyper-V 作为虚拟机管理程序。

1. 核心设置完成后，在“保护”选项卡上，单击“虚拟化”图块。
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. 现在出现添加虚拟机管理程序页面 > 选择供应商：选择 Hyper-V（输入 IP 地址或 FQDN 和用户凭据）
4. 对于 Hyper-V 服务器，单击发现节点。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。

COMMVAULT Search or type / for a command admin

Configure Microsoft Hyper-V Hypervisor

1 Add Hypervisor

2 Add VM Group

3 Summary

Add Hypervisor

Host name *
10.61.167.215

Hypervisor display name *
nimhvclus

Credential *
nimcred

Access nodes *
Click to select

- ☐ NIMHV001
- ☐ NIMHV002

CANCEL

NEXT
Activate Windows
Go to Settings to activate Windows.

5. 单击“下一步”并“保存”。

COMMVAULT Search or type / for a command admin

Filter navigation... Virtual machines **Hypervisors** VM groups

Guided setup Dashboard Protect **Virtualization** Kubernetes File servers

Add hypervisor Add VM group

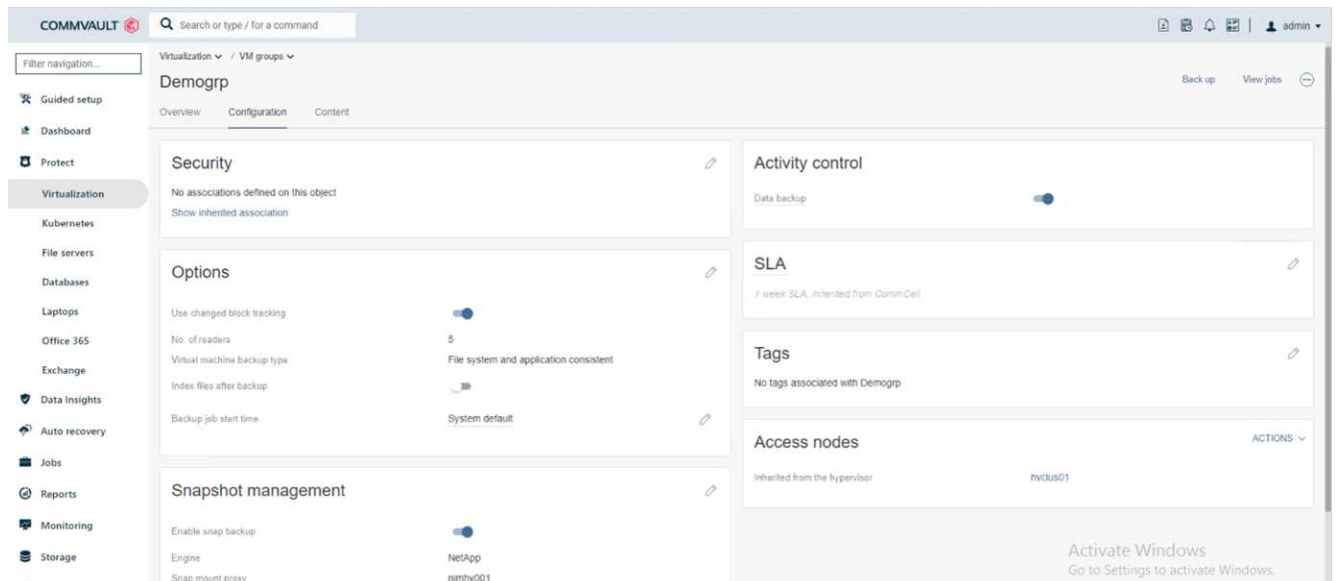
Hypervisors

All

Vendor = All + Add filter

Name ↑	Vendor	Version	Configured	Tags	Actions
hvclus01	Microsoft Hyper-V	Microsoft Windows Server 2022	✓	No tags	⋮

6. 在添加 VM 组页面上，选择要保护的虚拟机（在本例中 Demogrp 是创建的 VM 组）并启用 IntelliSnap 选项，如下所示。



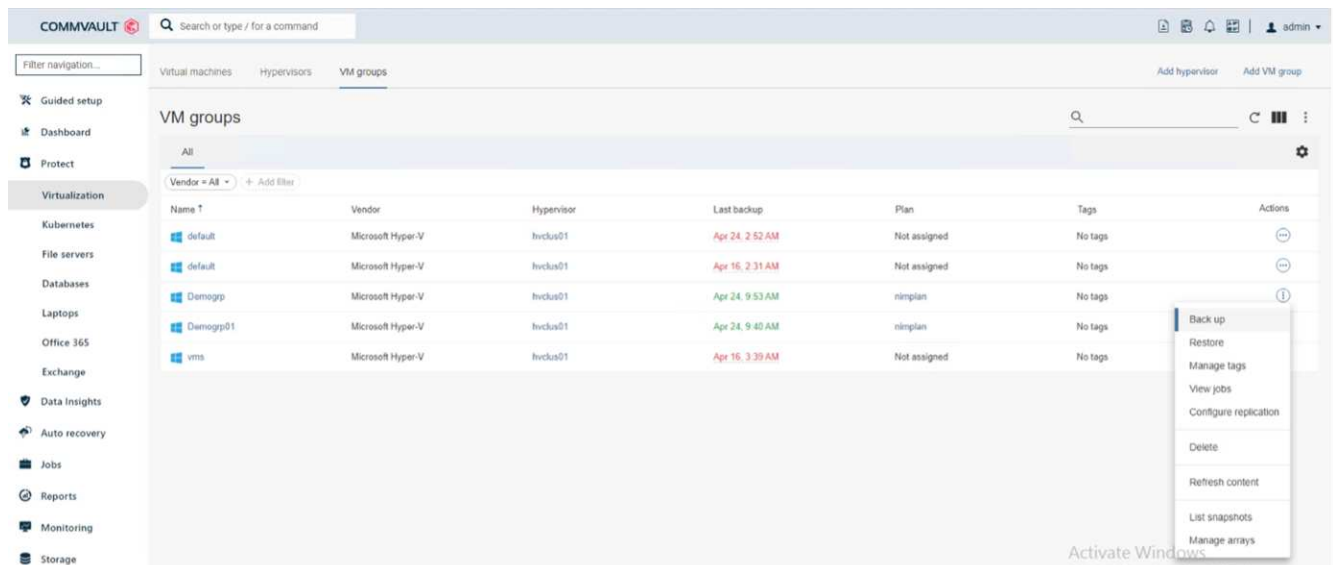
注意：当在 VM 组上启用 IntelliSnap 时，Commvault 会自动为主要（快照）和备份副本创建计划策略。

7. 单击“Save”。

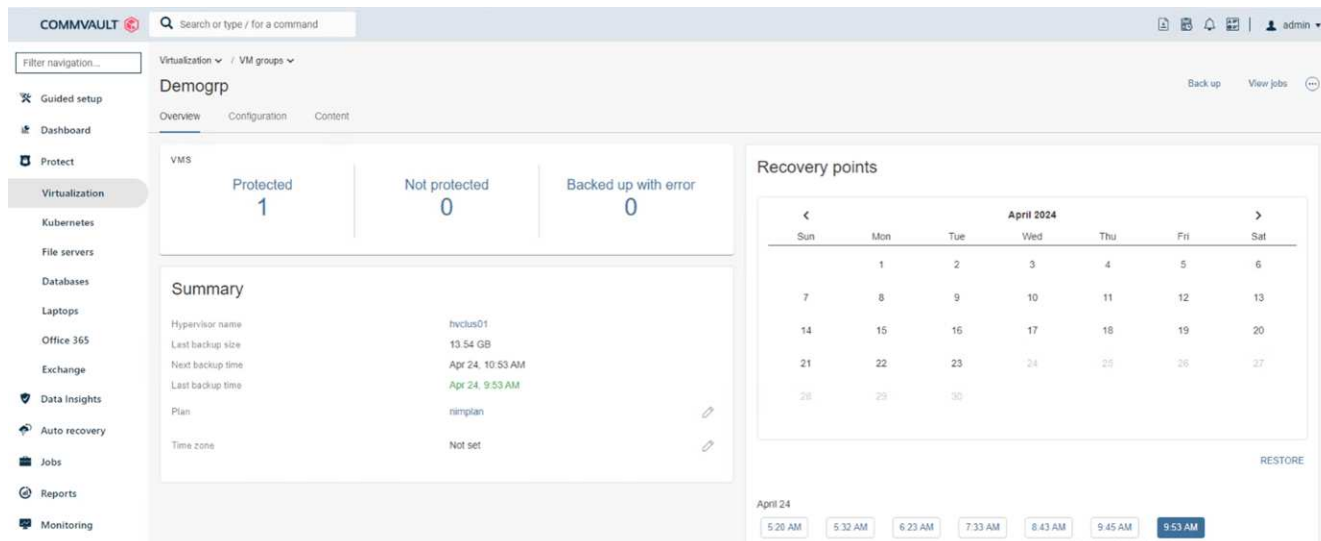
有关配置阵列的详细步骤，请参阅["HyperV 的引导设置"](#)。

执行备份：

1. 从导航窗格中，转到保护 > 虚拟化。出现虚拟机页面。
2. 备份虚拟机或虚拟机组。在这个演示中，选择了VM组。在 VM 组行中，单击操作按钮 action_button，然后选择备份。在这种情况下，nimplan 是与 Demogrp 和 Demogrp01 相关联的计划。



3. 一旦备份成功，还原点即可使用，如屏幕截图所示。从快照副本中，可以执行完整 VM 的还原以及客户文件和文件夹的还原。

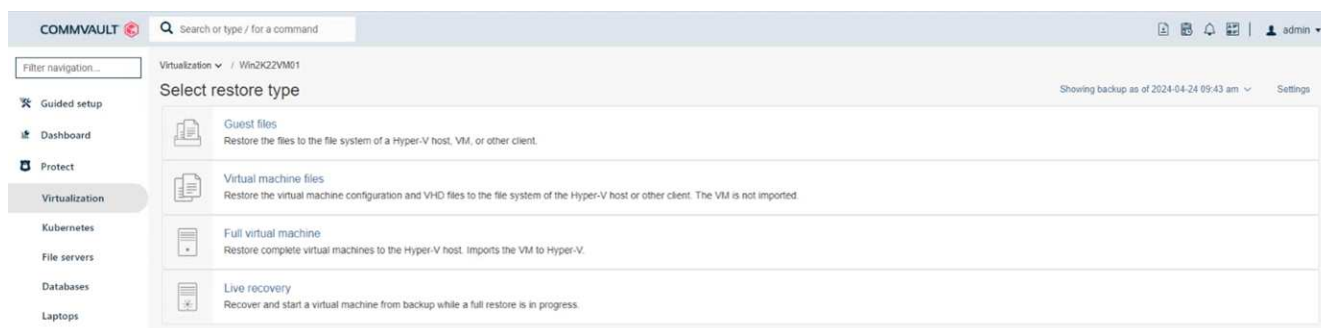


注意：对于关键且使用率较高的虚拟机，每个 CSV 保留较少的虚拟机。

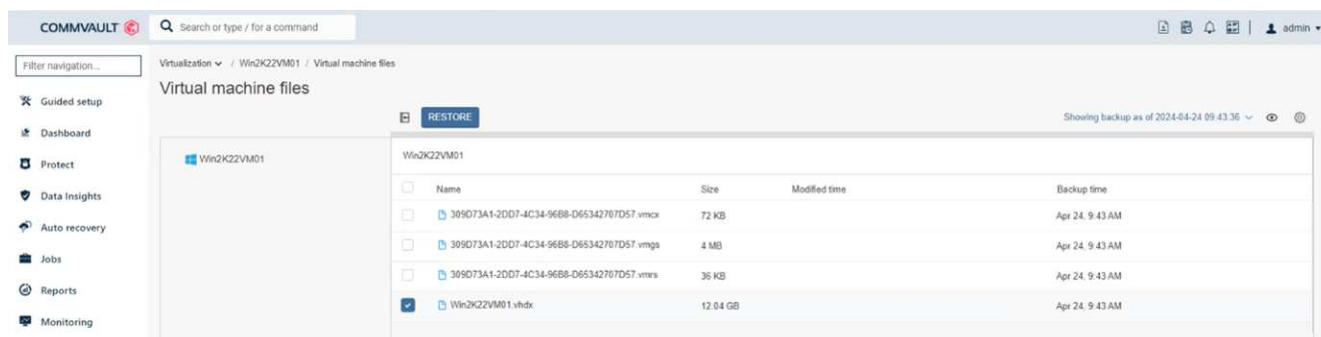
执行恢复操作：

通过还原点还原完整的虚拟机、客户文件和文件夹或虚拟磁盘文件。

1. 从导航窗格中，转到保护>虚拟化，将出现虚拟机页面。
2. 单击虚拟机组选项卡。
3. 出现 VM 组页面。
4. 在虚拟机组区域，单击包含虚拟机的虚拟机组的恢复。
5. 出现“选择还原类型”页面。



6. 根据选择选择客户文件或完整虚拟机并触发恢复。



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

高级NetApp ONTAP选项

NetApp SnapMirror支持高效的站点到站点存储复制，使灾难恢复快速、可靠且易于管理，适合当今的全球企业。SnapMirror可通过 LAN 和 WAN 高速复制数据，为关键任务应用程序提供高数据可用性和快速恢复，以及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。卷可以逐步备份到异地位置。SnapMirror按照所需的 RPO 频率执行增量、基于块的复制。块级更新减少了带宽和时间要求，并在 DR 站点保持了数据一致性。

一个重要的步骤是创建整个数据集的一次性基线传输。这是执行增量更新之前所必需的。此操作包括在源处创建 Snapshot 副本并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，可以进行计划或手动触发的更新。每次更新仅将新的和更改的块从源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，然后仅将更改的块传输到目标卷。新的副本将成为下一次更新的基线副本。由于复制是定期的，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很小。

通过完成以下步骤来执行恢复：

1. 连接到辅助站点上的存储系统。
2. 中断SnapMirror关系。
3. 将SnapMirror卷中的 LUN 映射到辅助站点上的 Hyper-V 服务器的启动器组 (igroup)。
4. 一旦 LUN 映射到 Hyper-V 集群，就使这些磁盘联机。
5. 使用故障转移群集 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 Hyper-V 管理器，使其具有高可用性，然后将其添加到集群中。
7. 打开虚拟机。

ONTAP存储系统上的 Microsoft Hyper-V 部署摘要

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，适用于多种用例和应用程序。Windows Server 2022 及其上启用的 Hyper-V 是虚拟化解决方案的一个常见用例，本文档对此进行了描述。ONTAP存储和相关功能的灵活性和可扩展性使客户能够从适当大小的存储层开始，该存储层可以随着不断变化的业务需求而增长并适应其需求。在当前的市场条件下，Hyper-V 提供了一个完美的替代虚拟机管理程序选项，它提供了 VMware 提供的大部分功能。

使用 PowerShell 脚本将虚拟机迁移到 Microsoft Hyper-V

使用 PowerShell 脚本将虚拟机从 VMware vSphere 迁移到使用FlexClone技术的 Microsoft Hyper-V。该脚本通过连接到 vCenter 和ONTAP集群、创建快照、将 VMDK 转换为 VHDX 以及在 Hyper-V 上配置虚拟机来简化迁移过程。

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=$True, HelpMessage="NetApp ONTAP NSF,SMB Volume
name")]
    [String]$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 ""$directoryName"" already exists!"
    }
    Else {
        Write-Error "Failed to create folder ""$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 "-----"

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

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

```


版权信息

版权所有 © 2026 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，本文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。