



# 在 **NetApp** 儲存設備上部署 **Microsoft Hyper-V**

## NetApp Solutions

NetApp  
May 14, 2024

# 目錄

在 NetApp 儲存設備上部署 Microsoft Hyper-V .....	1
在 NetApp 儲存設備上部署 Microsoft Hyper-V .....	1
在 NetApp 儲存設備上部署 Microsoft Hyper-V : 預先申請 .....	1
在 NetApp 儲存設備上部署 Microsoft Hyper-V : 考量事項 .....	6
在 NetApp 儲存設備上部署 Microsoft Hyper-V : 資料保護 .....	12
在 NetApp 儲存設備上部署 Microsoft Hyper-V : 結論 .....	18
在 NetApp 儲存設備上部署 Microsoft Hyper-V : 移轉指令碼 .....	18

# 在 NetApp 儲存設備上部署 Microsoft Hyper-V

## 在 NetApp 儲存設備上部署 Microsoft Hyper-V

Windows Server 平台使用 Hyper-V 角色來提供虛擬化技術。Hyper-V 是 Windows Server 提供的眾多選用角色之一。

### 總覽

Hyper-V 角色可讓我們使用 Windows Server 內建的虛擬化技術、建立及管理虛擬化運算環境。Hyper-V 技術會虛擬化硬體、提供一個環境、讓您可以在一部實體電腦上同時執行多個作業系統。Hyper-V 可讓您建立及管理虛擬機器及其資源。每個虛擬機器都是獨立的虛擬化電腦系統、可以執行自己的作業系統。Hyper-V 提供基礎架構來虛擬化應用程式與工作負載、以支援各種業務目標、以提高效率並降低成本、這是 VMware® vSphere 的最佳替代方案、尤其是在企業組織在目前的市場環境中、正在尋求共存的多個 Hypervisor 時。

### 目標對象

本文件說明 Hyper-V 叢集組態與 NetApp ONTAP 系統的架構與部署程序。本文件的目標對象包括銷售工程師、現場顧問、專業服務、IT 經理、合作夥伴工程師、以及想要將 Hyper-V 部署為主要或替代 Hypervisor 的客戶。

### 架構

本文檔中介紹的架構具體包括 Microsoft® Windows Server® 2022 和 Hyper-V® 虛擬化。NetApp 強烈建議您在每次部署中都使用虛擬化軟體和基礎架構管理軟體。此組態使用每個元件的最佳實務做法、以實現可靠的企業級基礎架構。

### 使用案例摘要

本文件說明在使用 NetApp All Flash FAS 和 ASA 陣列機型的 Microsoft Windows Server 2022 上、設定 Hyper-V 叢集以最佳方式執行工作負載的部署程序和最佳實務做法。伺服器作業系統 / Hypervisor 是 Microsoft Windows Server 2022。本指南涵蓋透過儲存區域網路 (SAN) 和網路附加儲存 (NAS) 傳輸協定來處理資料的 NetApp 儲存系統。

## 在 NetApp 儲存設備上部署 Microsoft Hyper-V：預先申請

本主題提供使用 ONTAP 儲存系統設定及部署雙節點容錯移轉叢集和叢集式 Hyper-V 虛擬機器的步驟。

### 部署程序的先決條件

- 所有硬體都必須通過所執行 Windows Server 版本的認證、而且完整的容錯移轉叢集解決方案必須通過驗證組態精靈中的所有測試
- Hyper-V 節點已加入網域控制器 (建議)、而且彼此之間也有適當的連線能力。
- 每個 Hyper-V 節點的設定都應該相同。
- 在每部 Hyper-V 伺服器上設定網路介面卡和指定的虛擬交換器、用於管理、iSCSI、SMB、即時移轉的隔

離流量。

- 容錯移轉叢集功能會在每部 Hyper-V 伺服器上啟用。
- SMB 共享區或 CSV 是用來儲存 VM 及其磁碟以用於 Hyper-V 叢集的共用儲存區。
- 不同叢集之間不應共用儲存設備。針對每個叢集規劃一或多個 CSV/CIFS 共用。
- 如果 SMB 共用是用作共用儲存設備、則必須設定 SMB 共用的權限、以授予叢集中所有 Hyper-V 節點的電腦帳戶存取權。

如需詳細資訊、請參閱：

- ["Windows Server 上 Hyper-V 的系統需求"](#)
- ["驗證容錯移轉叢集的硬體"](#)
- ["部署 Hyper-V 叢集"](#)

## 安裝 Windows 功能

下列步驟說明如何安裝必要的 Windows Server 2022 功能。

- 所有主機 \*
  1. 在所有指定節點上準備 Windows OS 2022 、並提供必要的更新和裝置驅動程式。
  2. 使用安裝期間輸入的管理員密碼登入每個 Hyper-V 節點。
  3. 在工作列中的 PowerShell 圖示上按一下滑鼠右鍵、然後選取、即可啟動 PowerShell 提示 Run as Administrator 。
  4. 新增 Hyper-V 、 MPIO 和叢集功能。

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

## 設定網路

正確的網路規劃是達成容錯部署的關鍵。為每種流量設定不同的實體網路介面卡、是容錯移轉叢集的標準建議。透過新增虛擬網路介面卡、切換內嵌群組（Set）和 Hyper-V QoS 等功能、可在較少的實體介面卡上壓縮網路流量。以服務品質、備援和流量隔離為考量、設計網路組態。將 VLAN 等網路隔離技術與流量隔離技術搭配使用、可為流量和服務品質提供備援、以改善儲存流量效能並增加一致性。

建議您使用多個邏輯和 / 或實體網路來分隔和隔離特定工作負載。典型的網路流量範例通常分為幾個區段、如下所示：

- iSCSI 儲存網路。
- CSV（叢集共享 Volume）或心跳網路。
- 即時移轉
- VM 網路
- 管理網路

- 注意 \*：當 iSCSI 搭配專用 NIC 使用時、不建議使用任何群組解決方案、也不應使用 MPIO / DSM。
- 注意 \*：Hyper-V 網路最佳實務做法也不建議在 Hyper-V 環境中使用 NIC 群組功能來支援 SMB 3.0 儲存網路。

如需其他資訊、請參閱 "[規劃 Windows Server 中的 Hyper-V 網路](#)"

## 決定 Hyper-V 的儲存設計

Hyper-V 支援 NAS（SMB3.0）和區塊儲存（iSCSI / FC）作為虛擬機器的備份儲存設備。NetApp 支援 SMB3.0、iSCSI 和 FC 傳輸協定、可作為 VM 的原生儲存設備 - 使用 iSCSI / FC 和 SMB3 的叢集共用磁碟區（CSV）。對於需要直接存取儲存設備的工作負載、客戶也可以使用 SMB3 和 iSCSI 做為來賓連線儲存設備選項。ONTAP 提供靈活的選項、搭配統一化儲存設備（All Flash Array）、適用於需要混合傳輸協定存取和 SAN 最佳化儲存設備（All SAN Array、僅適用於 SAN 組態）的工作負載。

SMB3 與 iSCSI / FC 的使用決定、是由目前現有的基礎架構所決定、SMB3/iSCSI 可讓客戶使用現有的網路基礎架構。對於擁有現有 FC 基礎架構的客戶、可以利用該基礎架構、將儲存設備呈現為 FC 型叢集式共享磁碟區。

- 注意：\* 執行 ONTAP 軟體的 NetApp 儲存控制器可在 Hyper-V 環境中支援下列工作負載：
- 以持續可用的 SMB 3.0 共享區代管的 VM
- 在 iSCSI 或 FC 上執行的叢集共用磁碟區（CSV） LUN 上託管的 VM
- 來賓儲存設備、並將磁碟傳遞至來賓虛擬機器
- 注意 \*：核心 ONTAP 功能、例如精簡配置、重複資料刪除、壓縮、資料壓縮、彈性複製、無論平台或作業系統為何、快照和複寫都能在背景中順暢運作、為 Hyper-V 工作負載帶來顯著價值。這些功能的預設設定最適合 Windows Server 和 Hyper-V
- 附註 \*：如果 VM 有多個路徑可用、且已安裝及設定多重路徑 I/O 功能、則可使用來賓啟動器在來賓 VM 上支援 MPIO。
- 注意 \*：ONTAP 支援所有主要的業界標準用戶端通訊協定：NFS、SMB、FC、FCoE、iSCSI、NVMe / FC 和 S3。不過、Microsoft 不支援 NVMe / FC 和 NVMe / TCP。

## 安裝 NetApp Windows iSCSI 主機公用程式

下節說明如何執行 NetApp Windows iSCSI 主機公用程式的自動安裝。如需安裝的詳細資訊、請參閱 "[安裝 Windows Unified Host Utilities 7.2（或最新支援版本）](#)"

- 所有主機 \*
  1. 下載 "[Windows iSCSI 主機公用程式](#)"
  2. 解除封鎖下載的檔案。

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

3. 安裝主機公用程式。

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

- 注意 \* : 系統將在此過程中重新開機。

## 設定 Windows 主機 iSCSI 啟動器

下列步驟說明如何設定內建的 Microsoft iSCSI 啟動器。

- 所有主機 \*

1. 以滑鼠右鍵按一下工作列中的 PowerShell 圖示、然後選取「以系統管理員身分執行」、即可啟動 PowerShell 提示。
2. 將 iSCSI 服務設定為自動啟動。

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

3. 啟動 iSCSI 服務。

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

4. 設定 MPIO 以宣告任何 iSCSI 裝置。

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

5. 將所有新宣告的裝置的預設負載平衡原則設為循環配置資源。

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

6. 為每個控制器設定 iSCSI 目標。

```
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. 將每個 iSCSI 網路的工作階段連線至每個目標。

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

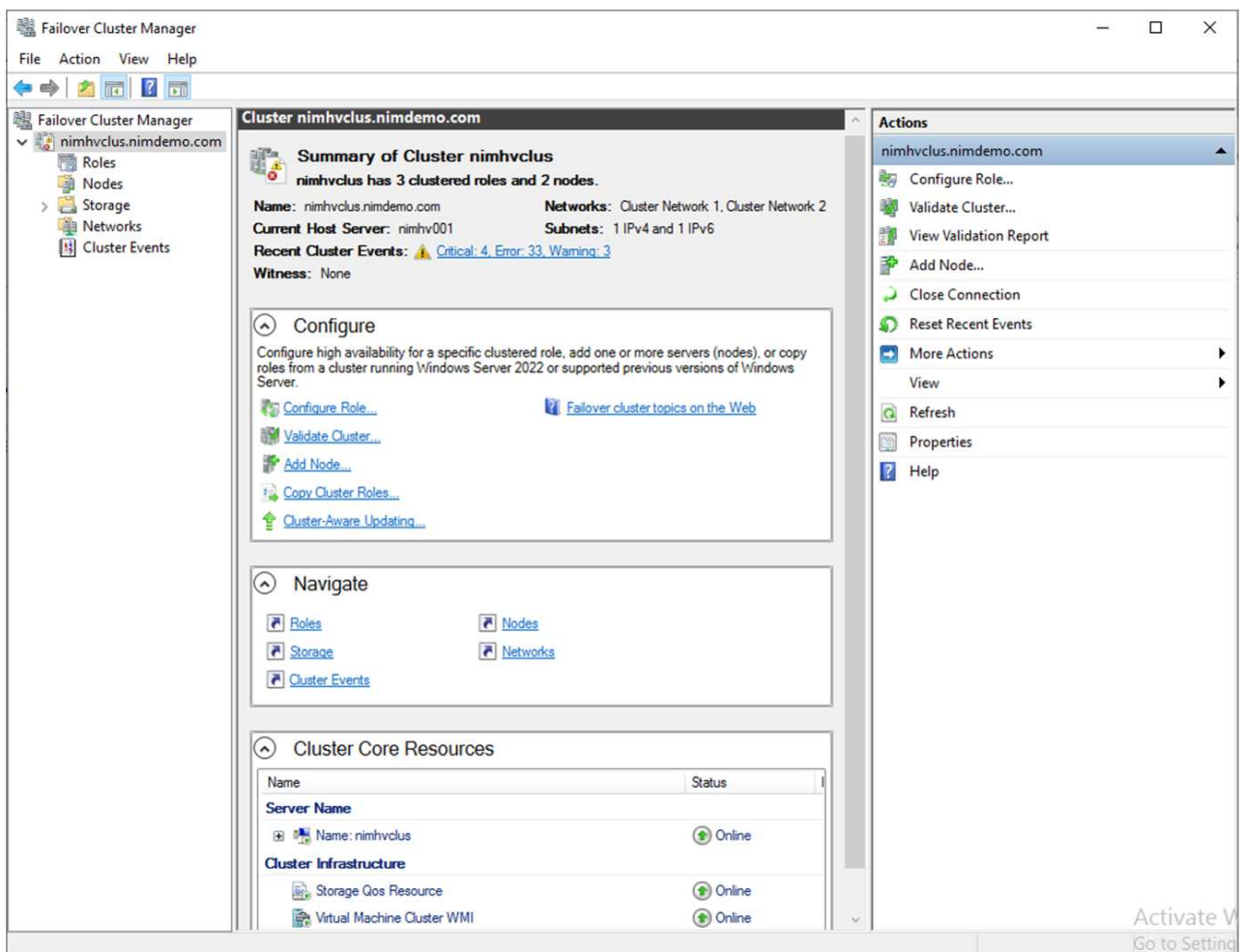
Get-IscsiTarget | Connect-IscsiTarget -IsPersistent $true
-IsMultipathEnabled $true -InitiatorPortAddress <iscsib_ipaddress>
```

- 附註 \* : 新增多個工作階段 (最少 5-8 個) 、以提升效能並運用頻寬。

## 建立叢集

- 僅限一部伺服器 \*
  1. 以滑鼠右鍵按一下 PowerShell 圖示並選取、即可啟動具有管理權限的 PowerShell 提示 Run as Administrator`。
  2. 建立新叢集。

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



3. 選擇適當的叢集網路進行即時移轉。
4. 指定 CSV 網路。

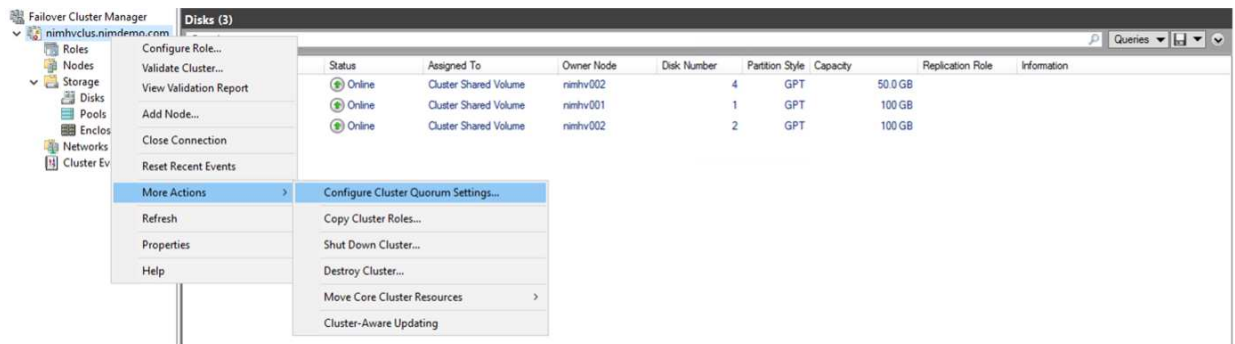
```
(Get-ClusterNetwork -Name Cluster).Metric = 900
```

5. 將叢集變更為使用仲裁磁碟。

- a. 以滑鼠右鍵按一下 PowerShell 圖示並選取「以系統管理員身分執行」、即可啟動具有系統管理權限的 PowerShell 提示。

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

- b. 在容錯移轉叢集管理程式中、選取 Configure Cluster Quorum Settings。



- c. 在「歡迎」頁面中按一下「下一步」。
- d. 選取仲裁見證、然後按一下「下一步」。
- e. 選取「設定磁碟見證」、然後按一下「下一步」。
- f. 從可用的儲存設備中選取 Disk W:、然後按一下 Next (下一步)。
- g. 在確認頁面上按一下「下一步」、然後在摘要頁面上按一下「完成」。

如需仲裁與見證的詳細資訊、請參閱 ["設定及管理仲裁"](#)

6. 從容錯移轉叢集管理程式執行叢集驗證精靈、以驗證部署。
7. 建立 CSV LUN 來儲存虛擬機器資料、並透過容錯移轉叢集管理程式中的角色來建立高可用度的虛擬機器。

## 在 NetApp 儲存設備上部署 Microsoft Hyper-V：考量事項

此步驟對於確定應用程式、服務和工作負載可在 Hyper-V 環境中有效運作至關重要。相容性檢查必須涵蓋作業系統版本、Windows 伺服器版本、應用程式相依性、資料庫系統、以及現有環境中存在的任何特定組態或自訂。



## 適當調整儲存容量

在部署工作負載或從現有 Hypervisor 移轉之前、請先確定工作負載的大小符合所需的效能。這可以透過收集每個個別 VM 的效能資料來輕鬆完成、這些資料會收集 CPU（已使用 / 已配置）、記憶體（已使用 / 已配置）、儲存（已配置 / 已使用）、網路處理量和延遲、以及讀取 / 寫入 IOPS、處理量和區塊大小的集合。這些參數是成功部署及正確調整儲存陣列和工作負載主機大小的必要參數。

- 注意 \*：為 Hyper-V 及相關工作負載調整儲存容量時、規劃 IOPS 與容量。
- 注意 \*：對於 I/O 密集的較高 VM 或需要大量資源和容量的 VM、請將作業系統和資料磁碟區隔開。作業系統和應用程式二進位檔不常變更、而且可以接受磁碟區損毀的一致性。
- 注意 \*：使用來賓連線儲存設備（也稱為來賓作業系統）來處理高效能資料磁碟、而非使用 VHD。這也有助於簡化複製程序。

## 提升虛擬機器效能

選擇適當數量的 RAM 和 vCPU、以獲得最佳效能、並將多個磁碟附加到單一虛擬 SCSI 控制器。仍建議使用固定的 VHD 作為部署虛擬磁碟的主要選擇、而且不限制使用任何類型的 VHDX 虛擬磁碟。

- 附註 \*：避免在 Windows Server 上安裝不必要的角色、以免被使用。
- 附註 \*：選擇第 2 代作為能夠從 SCSI 控制器載入 VM 的虛擬機器、並以 VMBus 和 VSP / VSC 架構為基礎、以達到開機層級、大幅提升整體 VM 效能。
- 注意 \*：避免經常進行檢查點、因為它會對 VM 效能造成負面影響。

## SMB3.0 設計與考量

SMB 3.0 檔案共用可做為 Hyper-V 的共用儲存設備 ONTAP 支援在 Hyper-V 的 SMB 共用上進行不中斷營運 Hyper-V 可以使用 SMB 檔案共用來儲存虛擬機器檔案、例如組態、快照和虛擬硬碟（VHD）檔案。針對 Hyper-V 使用專屬的 ONTAP CIFS SVM for SMB3.0 型共用用於儲存虛擬機器檔案的磁碟區必須使用 NTFS 安全性樣式磁碟區建立。如果有 Hyper-V 主機和 NetApp 陣列可用、建議在 10GB 網路上進行連線。如果是 1GB 網路連線、NetApp 建議您建立由多個 1GB 連接埠組成的介面群組。將服務 SMB 多通道的每個 NIC 連接到其專用 IP 子網路、以便每個子網路在用戶端和伺服器之間提供單一路徑。

- 重點 \*
- 在 ONTAP SVM 上啟用 SMB 多通道
- ONTAP CIFS SVM 在叢集中的每個節點上至少應有一個資料 LIF。
- 使用的共用必須使用持續可用的內容集進行設定。
- ONTAP One 現在已包含在每個 AFF（A 系列和 C 系列）、All SAN Array（ASA）和 FAS 系統上。因此不需要個別的授權。
- 若為共用的 VHD、請使用來賓連線的 iSCSI LUN
- 附註 \*：ODX 支援、可跨通訊協定運作。在檔案共用區與 iSCSI 或 FCP 附加 LUN 之間複製資料也會使用 ODX。
- 附註 \*：叢集中節點上的時間設定應相應設定。如果 NetApp CIFS 伺服器必須參與 Windows Active Directory（AD）網域、則應使用網路時間傳輸協定（NTP）。
- 注意 \*：必須透過 CIFS 伺服器啟用大 MTU 值。較小的封包大小可能會導致效能降低。

## 資源配置 SMB Volume

1. 確認儲存虛擬機器（SVM）上已啟用必要的 CIFS 伺服器選項
2. 下列選項應設為 true：啟用 SMB2 的 smb3 啟用複製卸載的 ShadowCopy-enabled is - 啟用多通道的 smb2-smb3-mtu

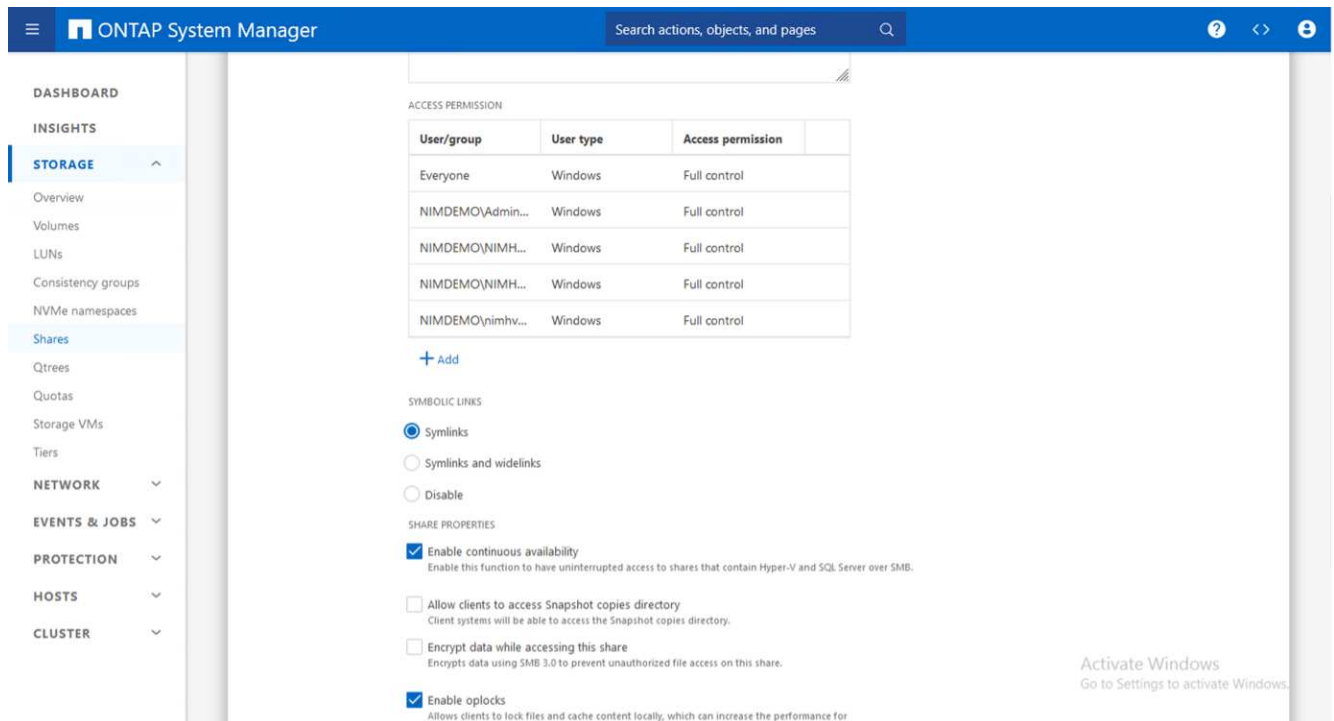
```
HV_NestedCluster::> vserver cifs options show -vserver NestedHVsvm01 -fields copy-offload-enabled, is-multichannel-enabled, is-large-mtu-enabled, smb2-enabled, smb3-enabled, copy-offload-enabled, shadowcopy-enabled
vserver      smb2-enabled smb3-enabled copy-offload-enabled shadowcopy-enabled is-multichannel-enabled is-large-mtu-enabled
-----
NestedHVsvm01 true         true         true         true         true         true
```

3. 在儲存虛擬機器（SVM）上建立 NTFS 資料磁碟區、然後設定持續可用的共享區、以搭配 Hyper-V 使用

```
HV_NestedCluster::> volume create -vserver NestedHVsvm01 -volume hvdemosmb -aggregate HV_NestedCluster_01_VM_DISK_1 -size 500GB -security-style ntfs -junction-path /hvdemosmb
[Job 169] Job succeeded: Successful
```

- 附註 \*：除非將組態中使用的磁碟區建立為 NTFS 安全型磁碟區、否則透過 SMB 執行 Hyper-V 的不中斷作業將無法正常運作。

4. 在共用上啟用持續可用的 NTFS 權限、並將其設定為包含完全控制的 Hyper-V 節點。



如需詳細的最佳實務做法指引、請參閱 "[Hyper-V 的部署準則與最佳實務做法](#)"。

如需其他資訊、請參閱 "[SMB伺服器與磁碟區需求、適用於SMB上的Hyper-V](#)"。

## 區塊傳輸協定設計與考量

- 重點 \*
- 在主機上使用多重路徑（MPIO）來管理多重路徑。視需要建立更多路徑、以促進資料移動性作業、或是利用額外的 I/O 資源、但不要超過主機作業系統可支援的最大路徑數。
- 在存取 LUN 的主機上安裝主機公用程式套件。

- 建立至少 8 個磁碟區。
- 附註 \*：每個磁碟區使用一個 LUN、因此 LUN 與 CSV 的比例對應為 1：1。
- SVM 應在每個儲存控制器上、每個乙太網路或光纖通道架構都有一個 LIF、以便使用 iSCSI 或光纖通道來提供資料。
- SVM 使用 FCP 或 iSCSI 提供資料、需要 SVM 管理介面。

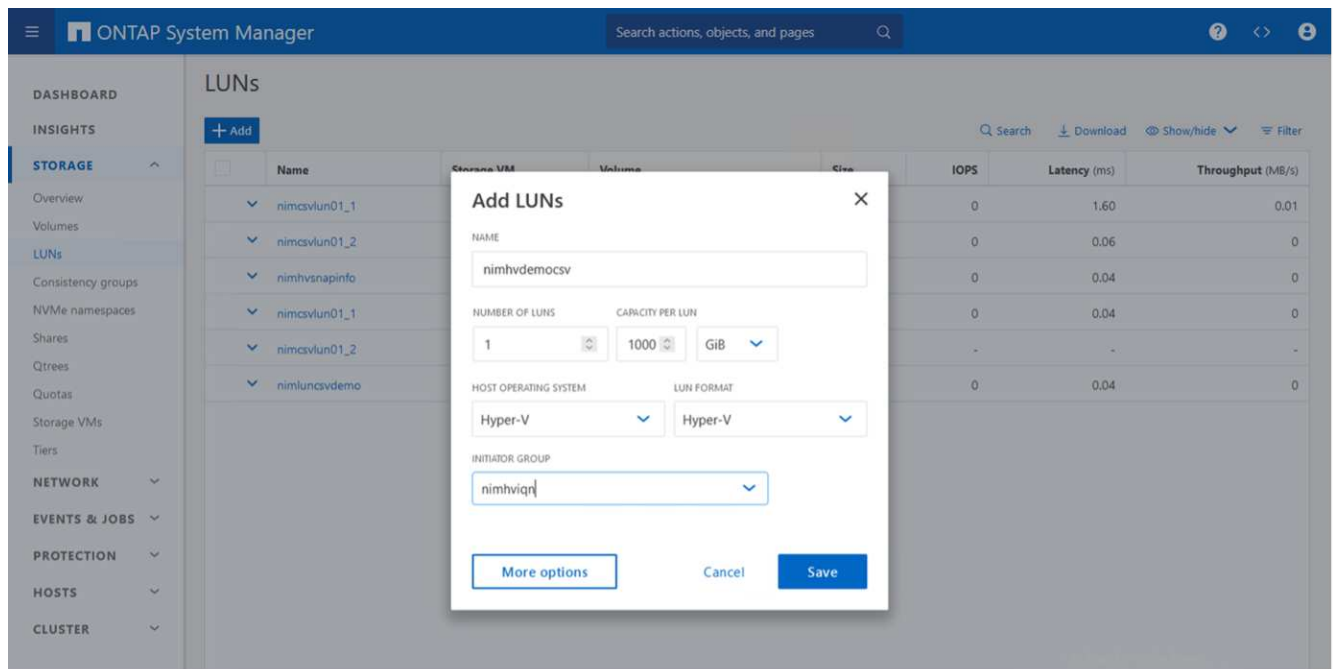
## 資源配置 iSCSI Volume

若要佈建 iSCSI Volume、請確保符合下列先決條件。

- 儲存虛擬機器（SVM）應啟用 iSCSI 傳輸協定、並建立適當的邏輯介面（生命）。
- 指定的 Aggregate 必須有足夠的可用空間來容納 LUN。
- 注意 \*：根據預設、ONTAP 使用選擇性 LUN 對應（SLM），只能透過擁有 LUN 的節點上的路徑及其高可用性（HA）合作夥伴來存取 LUN。
- 如果 LUN 移至叢集中的另一個節點、請在每個節點上設定所有 iSCSI 生命體、以實現 LUN 的移動性。

### 步驟

1. 使用系統管理員並瀏覽至 LUN 視窗（ONTAP CLI 可用於相同的作業）。
2. 按一下「建立」。
3. 瀏覽並選擇要在其中建立 LUN 的指定 SVM、並顯示 Create LUN Wizard（建立 LUN 精靈）。
4. 在「一般內容」頁面上、針對包含 Hyper-V 虛擬機器虛擬硬碟（VHD）的 LUN 選取 Hyper-V。

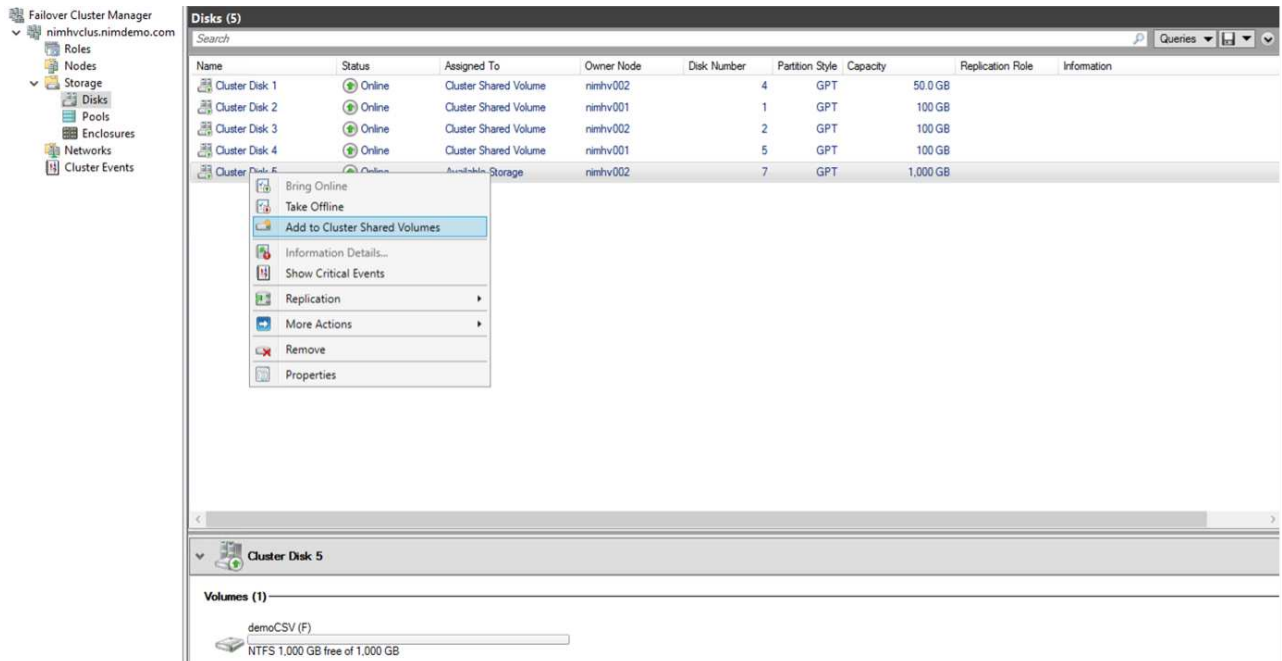


5. 在 LUN Container 頁面上、選擇現有的 FlexVol Volume、否則將建立新的 Volume。
6. 在 [ 啟動器對應 ] 頁面上按一下 [ 其他選項 ]，然後按一下 [ 新增啟動器群組 ]，在 [ 一般 ] 索引標籤上輸入必要的資訊，再在 [ 啟動器 ] 索引標籤上輸入主機的 iSCSI 啟動器節點名稱。

7. 確認詳細資料、然後按一下「完成」以完成精靈。

建立 LUN 後、請前往容錯移轉叢集管理程式。若要將磁碟新增至 CSV、必須將磁碟新增至叢集的可用儲存群組（如果尚未新增）、然後將磁碟新增至叢集上的 CSV。

- 注意 \*：在容錯移轉叢集中、預設會啟用 CSV 功能。
- 將磁碟新增至可用儲存設備：\*
  1. 在容錯移轉叢集管理程式的主控台樹狀目錄中、展開叢集名稱、然後展開儲存。
  2. 以滑鼠右鍵按一下 [ 磁碟 ]，然後選取 [ 新增磁碟 ]。隨即出現清單、顯示可新增以用於容錯移轉叢集的磁碟。
  3. 選取您要新增的磁碟、然後選取確定。
  4. 磁碟現在已指派給可用的儲存群組。
  5. 完成後、請選取剛指派給可用儲存設備的磁碟、以滑鼠右鍵按一下選取項目、然後選取新增至叢集共用磁碟區。



6. 磁碟現在已指派給叢集中的叢集共用 Volume 群組。磁碟會以編號的磁碟區（掛載點）形式、顯示在 %SystemDrive%\ClusterStorage 資料夾下。這些磁碟區會出現在 CSVFS 檔案系統中。

如需其他資訊、請參閱 "[在容錯移轉叢集中使用叢集共用磁碟區](#)"。

- 建立高可用度的虛擬機器：\*

若要建立高可用度的虛擬機器、請遵循下列步驟：

1. 在容錯移轉叢集管理程式中、選取或指定您要的叢集。確保叢集下的主控台樹狀目錄已展開。
2. 按一下角色。
3. 按一下 [ 動作 ] 窗格中的 [ 虛擬機器 ]，然後按一下 [ 新增虛擬機器 ]。此時將出現 [ 新增虛擬機器精靈 ]。按一下「下一步」

4. 在 [ 指定名稱與位置 ] 頁面上，指定虛擬機器的名稱，例如 Nimdemo 。按一下 [ 將虛擬機器儲存在其他位置 ] ，然後輸入完整路徑，或按一下 [ 瀏覽 ] 並瀏覽至共用儲存設備。
5. 將記憶體指派給與實體網路介面卡相關的虛擬交換器、並設定網路介面卡。
6. 在「Connect Virtual Hard Disk」頁面上、按一下「Create a virtual hard disk」（建立虛擬硬碟）。
7. 在「安裝選項」頁面上、按一下「從開機 CD/DVD-ROM 安裝作業系統」。在 [ 媒體 ] 下，指定媒體的位置，然後按一下 [ 完成 ] 。
8. 虛擬機器即會建立。然後、容錯移轉叢集管理程式中的高可用度精靈會自動設定虛擬機器的高可用度。

## 使用 ODX 功能快速配置虛擬磁碟

ONTAP 的 ODX 功能可複製主要 VHDX 檔案、只需複製 ONTAP 儲存系統代管的主要 VHDX 檔案、即可製作主要 VHDX 的複本。由於啟用 ODX 的複本不會將任何資料放在網路線路上、因此複製程序會在 NetApp 儲存端執行、因此速度最多可快上六到八倍。快速資源配置的一般考量包括儲存在檔案共用上的主系統重新繪製映像、以及由 Hyper-V 主機啟動的一般複本程序。

- 注意 \* : ONTAP 支援 ODX 、適用於 SMB 與 SAN 傳輸協定。
- 注意 \* : 若要利用 ODX 複本卸載傳遞與 Hyper-V 的使用案例、來賓作業系統必須支援 ODX 、而且來賓作業系統的磁碟必須是支援 ODX 的 SCSI 磁碟（SMB 或 SAN）。客體作業系統上的 IDE 磁碟不支援 ODX 傳遞。

## 效能最佳化

雖然每個 CSV 建議的虛擬機器數量是主觀的、但許多因素決定了可放置在每個 CSV 或 SMB 磁碟區上的最佳虛擬機器數量。雖然大多數系統管理員只考慮容量，但傳送至 VHD 的並行 I/O 數量是整體效能最重要的因素之一。控制效能的最簡單方法是調整每個 CSV 或共享區上放置的虛擬機器數量。如果並行虛擬機器 I/O 模式傳送太多流量給 CSV 或共享區、則磁碟佇列會填滿、並產生較高的延遲。

## SMB Volume 和 CSV 規模調整

確保解決方案的端點對端大小適當、以避免瓶頸、當建立用於 Hyper-V VM 儲存目的的磁碟區時、最佳做法是建立不超過所需容量的磁碟區。適當調整規模的磁碟區可避免在 CSV 上意外放置太多虛擬機器、並降低資源爭用的可能性。每個叢集共用 Volume（CSV）都支援一個 VM 或多個 VM。CSV 上要放置的虛擬機器數量取決於工作負載和業務偏好、以及如何使用 ONTAP 儲存功能、例如快照和複寫。在大多數部署案例中、將多個 VM 放在 CSV 上是一個良好的起點。針對特定使用案例調整此方法、以符合效能和資料保護需求。

由於磁碟區和 VHD 大小可以輕鬆增加、如果 VM 需要額外容量、就不需要調整超過所需的 CSV 大小。磁碟部分可用於擴充 CSV 大小、或是建立新的 CSV、並將所需的 VM 移轉至新的 CSV。為了達到最佳效能、最佳做法是增加 CSV 數量、而非增加 CSV 的規模作為臨時措施。

## 移轉

在目前市場情況下、最常見的使用案例之一是移轉。客戶可以使用 VMM Fabric 或其他協力廠商移轉工具來移轉 VM。這些工具會使用主機層級複本、將資料從來源平台移至目的地平台、視移轉範圍內的虛擬機器數量而定、可能會耗費大量時間。

在這類案例中使用 ONTAP、可比使用主機型移轉程序更快進行移轉。ONTAP 也能將 VM 從一個 Hypervisor 快速移轉至另一個 Hypervisor（在此案例中為 ESXi）。任何大小的 VMDK 都可以在 NetApp 儲存設備上以秒為單位轉換為 VHD。這就是我們的 PowerShell 方法：它利用 NetApp FlexClone® 技術來快速轉換 VM 硬碟。它也會處理目標 VM 和目的地 VM 的建立和組態。

此程序有助於將停機時間降至最低、並提高企業生產力。此外、它還能降低授權成本、限制及承諾單一廠商、提供更多選擇與靈活度。這對於想要最佳化 VM 授權成本並擴充 IT 預算的組織也很有幫助。

下列影片將虛擬機器從 VMware ESX 移轉至 Hyper-V 的程序虛擬化

### 從 ESX 到 Hyper-V 的零接觸移轉

如需使用 FlexClone 和 PowerShell 移轉的其他資訊、請參閱 "[用於移轉的 PowerShell 指令碼](#)"。

## 在 NetApp 儲存設備上部署 Microsoft Hyper-V：資料保護

資料保護是任何正式作業工作負載的主要租戶。本節說明如何備份及還原 Hyper-V 虛擬機器。

### 使用 NetApp 儲存快照還原

備份 VM 並快速恢復或複製 VM 是 ONTAP 磁碟區的主要優勢之一。使用 Snapshot 複本快速複製 VM 的 FlexClone 複本、甚至是整個 CSV 磁碟區、而不會影響效能。如此一來、在複製正式作業資料磁碟區、並將其裝載於 QA、移位和開發環境中時、就能處理正式作業資料、而不會有資料毀損的風險。FlexClone Volume 可用於製作正式作業資料的測試複本、而無需將複製資料所需的空間量增加一倍。

請記住、Hyper-V 節點會為每個磁碟指派唯一的 ID、並針對具有個別分割區（MBR 或 GPT）的磁碟區擷取快照、將會具有相同的唯一識別碼。MBR 使用磁碟簽章、GPT 使用 GUID（全域唯一識別碼）。如果是獨立式 Hyper-V 主機、則可以輕鬆掛載 FlexClone Volume、而不會發生任何衝突。這是因為獨立式 Hyper-V 伺服器可以自動偵測重複的磁碟 ID、並在不需要使用者介入的情況下動態變更。此方法可用於在案例需要時複製 VHD 來恢復 VM。

雖然獨立式 Hyper-V 主機很簡單、但 Hyper-V 叢集的程序卻不同。恢復程序包括將 FlexClone 磁碟區對應至獨立 Hyper-V 主機、或使用磁碟區將 FlexClone 磁碟區對應至獨立的 Hyper-V 主機、以手動變更簽名（這很重要、因為磁碟 ID 衝突會導致磁碟無法上線）、完成後、將 FlexClone Volume 對應至叢集。

### 使用協力廠商解決方案進行備份與還原

- 注意 \*：本節使用 CommVault、但這適用於其他第三方解決方案。

藉助 ONTAP 快照，CommVault IntelliSnap® 可創建基於硬件的快照。Hyper-V 備份可以根據 Hyper-V Hypervisor 或 VM 群組的組態自動進行、或是針對 VM 群組或特定 VM 手動進行。IntelliSnap 可快速保護 Hyper-V 環境、將生產力虛擬化伺服器陣列的負載降至最低。IntelliSnap 技術與虛擬伺服器代理程式（VSA）的整合、可讓 NetApp ONTAP 陣列在幾分鐘內完成大量虛擬機器和資料儲存區的備份。精細的存取功能可從儲存設備的次層和完整的客體 .vhd 檔案、提供個別的檔案和資料夾還原功能。

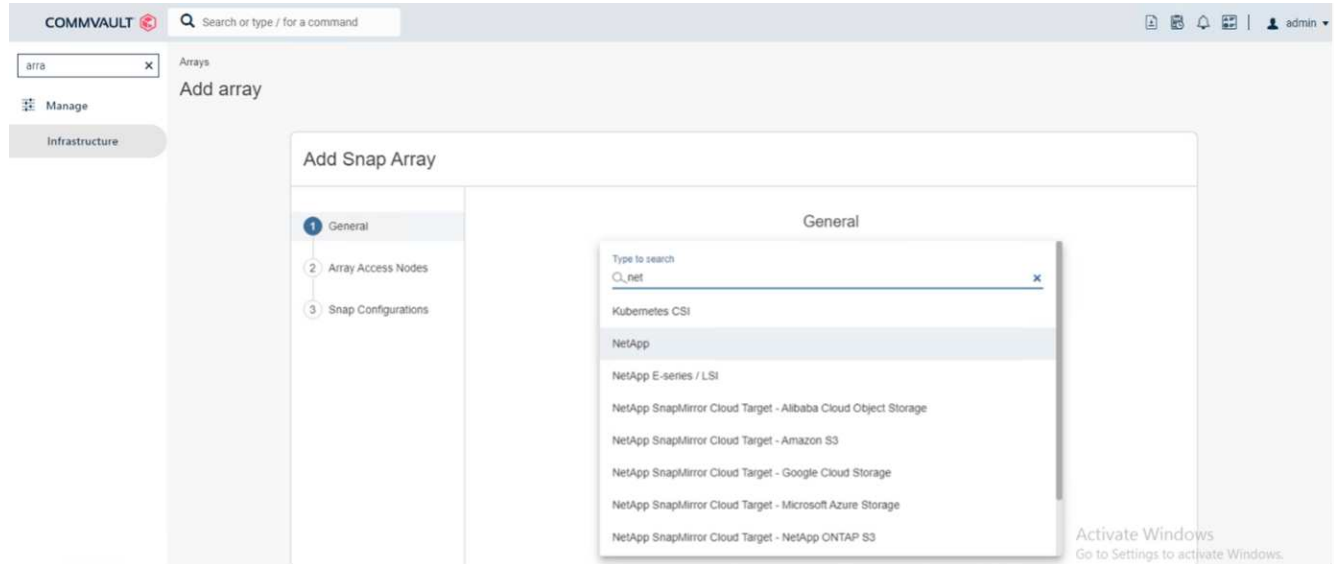
在設定虛擬化環境之前、請先部署需要與陣列進行快照整合的適當代理程式。Microsoft Hyper-V 虛擬化環境需要下列代理程式：

- MediaAgent
- 虛擬伺服器代理程式（VSA）
- VSS 硬體供應商（Windows Server 2012 及更新的作業系統）
- 使用陣列管理 \* 設定 NetApp 陣列

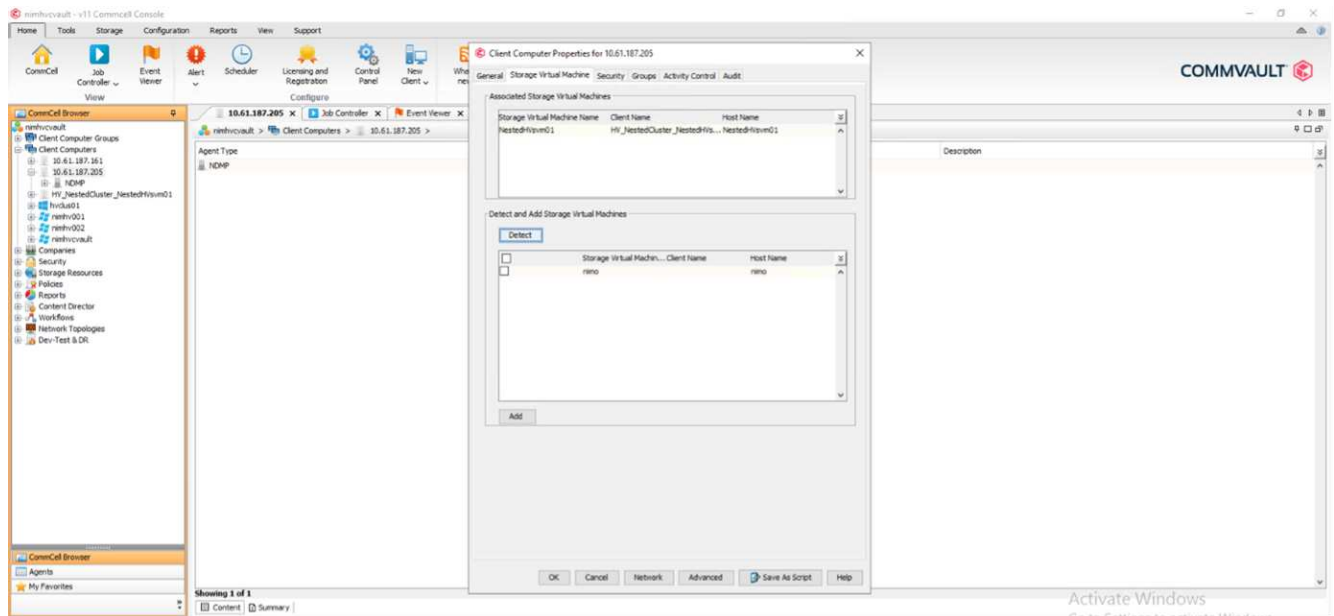
下列步驟說明如何在使用 ONTAP 陣列和 Hyper-V 的環境中設定 IntelliSnap 虛擬機器備份

1. 在「CommVault Console」的功能區上、按一下「Storage」（儲存）索引標籤、然後按一下「Array Management」（陣列管理）。
2. 此時將出現 Array Management（陣列管理）對話框。
3. 按一下「新增」

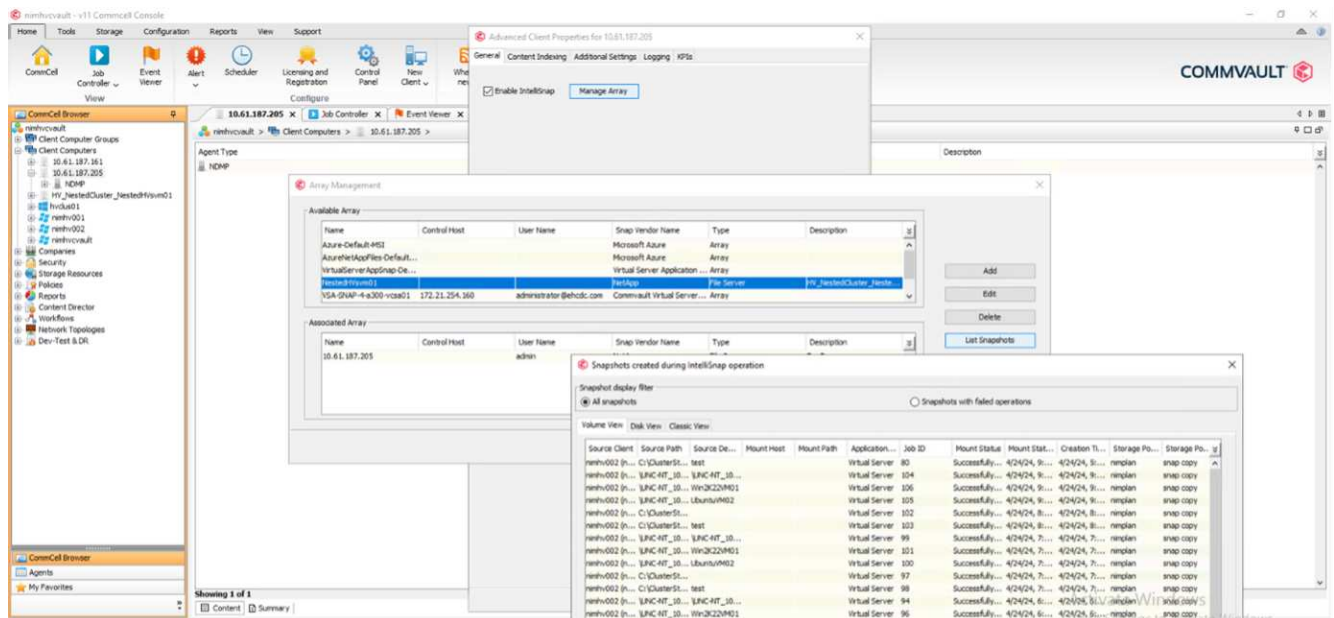
此時將出現 Array Properties（陣列屬性）對話框。



4. 在「一般」索引標籤上、指定下列資訊：
5. 從 Snap Vendor（管理廠商）清單中、選取 NetApp（管理廠商）。
6. 在名稱方塊中、輸入主檔案伺服器的主機名稱、完整網域名稱（FQDN）或 TCP/IP 位址。
7. 在 Array Access Nodes（陣列存取節點）標籤上、選取 Available media agents（可用媒體代理）。
8. 在 Snap Configuration（快照組態）索引標籤上、根據您的需求設定 Snapshot Configuration Properties（快照組態內容）。
9. 按一下「確定」。
10. <Mandatory step> 完成後、也可以使用 detect 選項在 NetApp 儲存陣列上設定 SVM、以自動偵測儲存虛擬機器（SVM）、然後選擇 SVM、並使用 add 選項、將 SVM 新增至 CommServe 資料庫、做為陣列管理項目。



11. 按一下「進階」(如下圖所示)、然後選取「啟用 IntelliSnap」核取方塊。



如需設定陣列的詳細步驟、請參閱 "設定 NetApp 陣列" 和 "在 NetApp 陣列上設定儲存虛擬機器"

- 將 Hyper-V 新增為 Hypervisor \*

下一步是新增 Hyper-V Hypervisor 並新增 VM 群組。

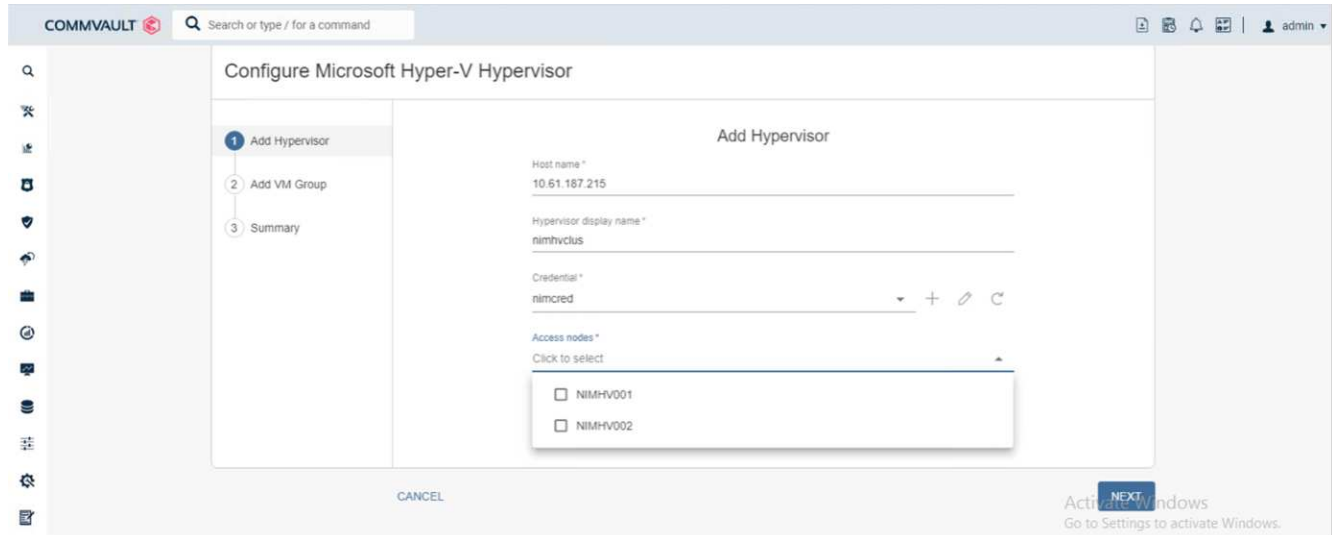
- 先決條件 \*
- Hypervisor 可以是 Hyper-V 叢集、叢集中的 Hyper-V 伺服器或獨立的 Hyper-V 伺服器。
- 使用者必須屬於 Hyper-V Server 2012 及更新版本的 Hyper-V 管理員群組。對於 Hyper-V 叢集、使用者帳戶必須具有完整的叢集權限 (讀取和完全控制)。
- 識別要安裝虛擬伺服器代理程式 (VSA) 的一或多個節點、以建立備份與還原作業的存取節點 (VSA Proxy)。若要探索 Hyper-V 伺服器、CommServe 系統必須安裝 VSA。



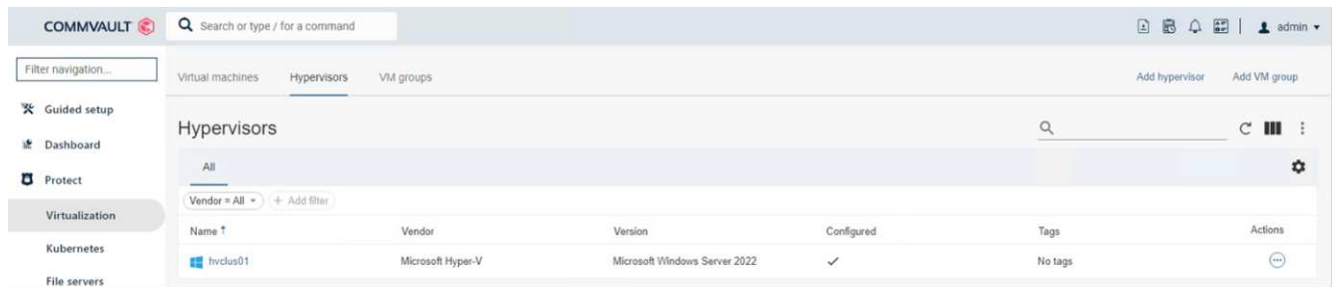
- 若要針對 Hyper-V 2012 R2 使用變更的區塊追蹤、請選取 Hyper-V 叢集中的所有節點。

下列步驟說明如何將 Hyper-V 新增為 Hypervisor。

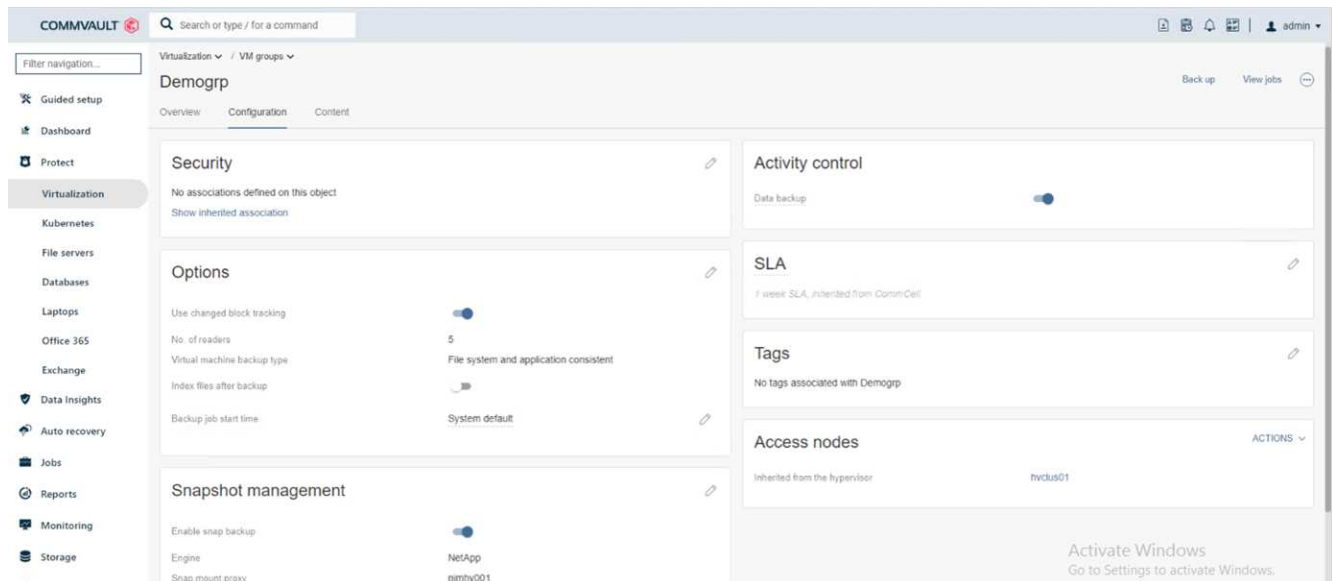
1. 核心設定完成後、在保護索引標籤上、按一下虛擬化磚。
2. 在「建立伺服器備份計畫」頁面上、輸入計畫的名稱、然後提供儲存、保留和備份排程的相關資訊。
3. 現在出現「新增 Hypervisor」頁面 > 選擇廠商：選擇 Hyper-V（輸入 IP 位址或 FQDN 和使用者認證）
4. 若為 Hyper-V 伺服器、請按一下「探索節點」。填入「節點」欄位時、請選取一個或多個要安裝 Virtual Server Agent 的節點。



5. 按一下「下一步」和「儲存」。



6. 在「Add VM Group」（新增 VM 群組）頁面上、選取要保護的虛擬機器（Demograp 是在此案例中建立的 VM 群組）、然後啟用 IntelliSnap 選項、如下所示。



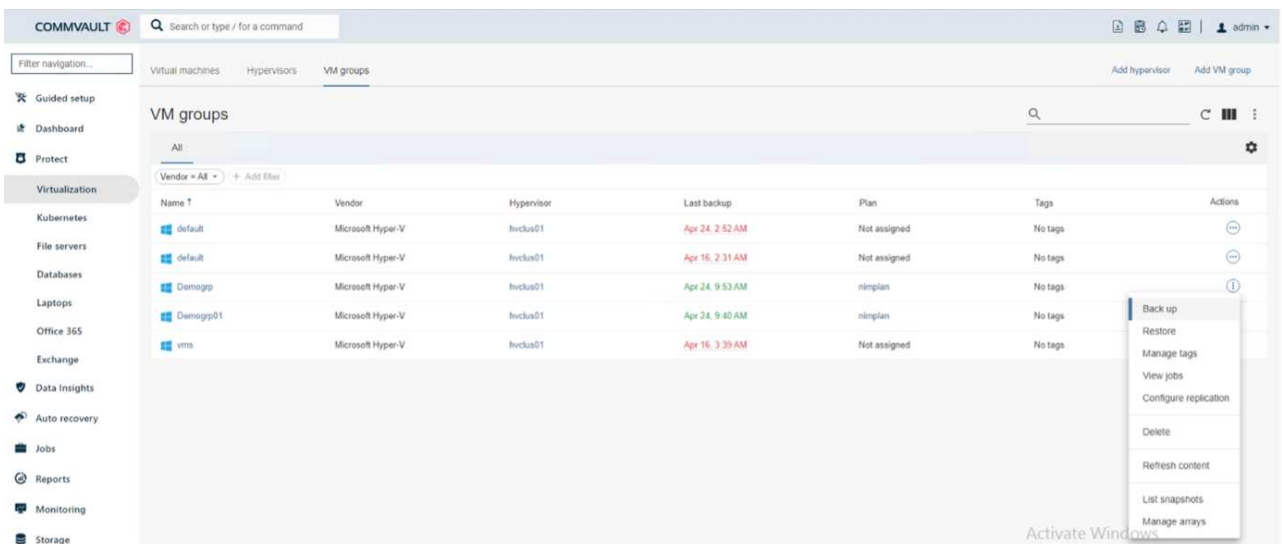
。注意 \*：在 VM 群組上啟用 IntelliSnap 時、CommVault 會自動為主要（Snap）和備份複本建立排程原則。

7. 按一下儲存。

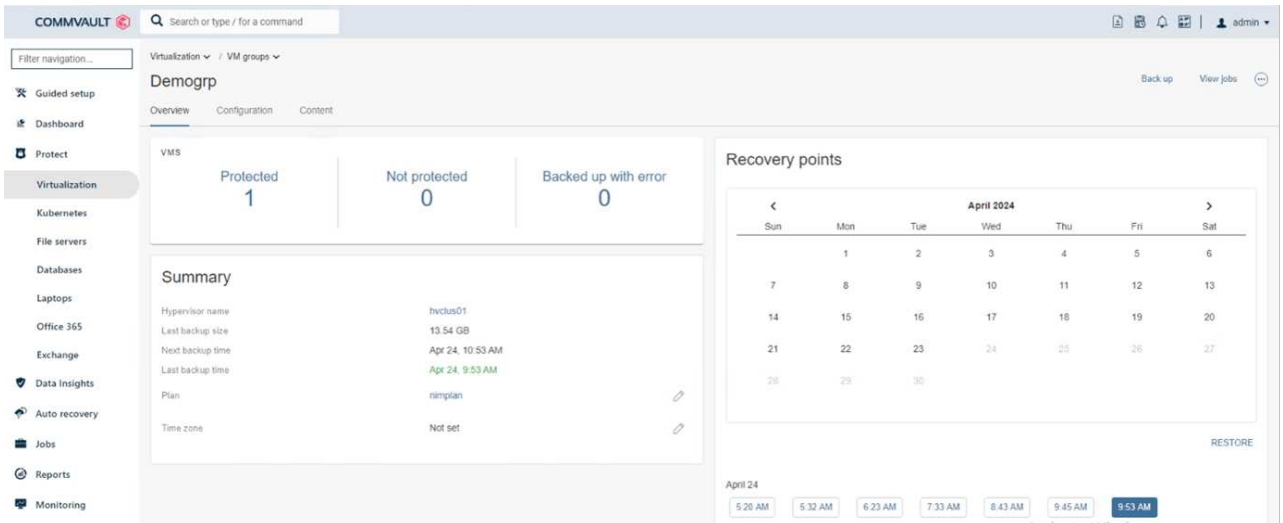
如需設定陣列的詳細步驟、請參閱 "新增 Hypervisor"。

• 執行備份：\*

1. 從導覽窗格中、移至「保護」>「虛擬化」。此時將出現 [ 虛擬機器 ] 頁面。
2. 備份 VM 或 VM 群組。在此示範中、已選取 VM 群組。在 VM 群組的列中、按一下動作按鈕 ACTION\_button、然後選取備份。在此案例中、nimplan 是與 Demograp 和 Demogra01 相關的計畫。



3. 一旦備份成功、還原點就會如畫面擷取所示。您可以從快照複本還原完整 VM、並還原客體檔案和資料夾。



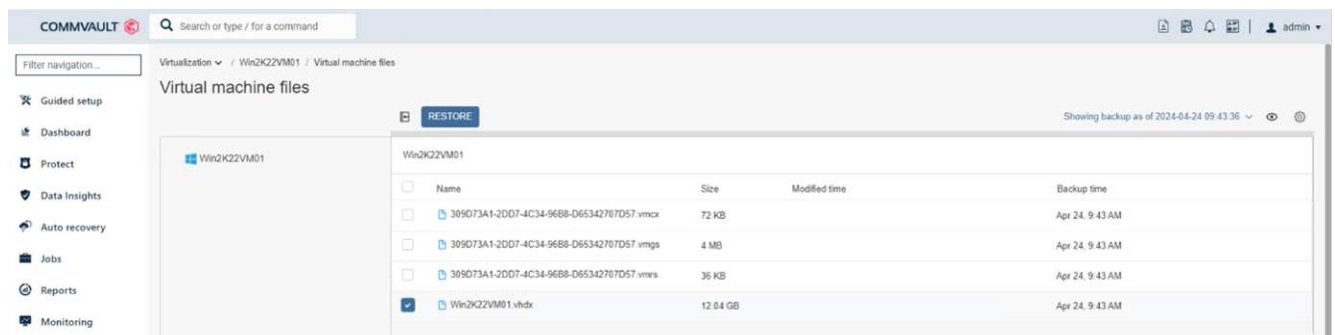
- 附註 \*：對於關鍵且使用率高的虛擬機器、每個 CSV 可保留較少的虛擬機器
- 執行還原作業：\*

透過還原點還原完整 VM、來賓檔案和資料夾、或虛擬磁碟檔案。

1. 從導覽窗格中、移至「保護」>「虛擬化」、隨即出現「虛擬機器」頁面。
2. 按一下 VM 群組索引標籤。
3. 此時將顯示 VM 群組頁面。
4. 在 VM Groups (VM 群組) 區域中、針對包含虛擬機器的 VM 群組按一下 Restore (還原)。
5. 此時會出現「選取還原類型」頁面。



6. 根據所選項目、選取來賓檔案或完整虛擬機器、然後觸發還原。



如需所有支援還原選項的詳細步驟、請參閱 ["Hyper-V 的還原"](#)。

## 進階 NetApp ONTAP 選項

NetApp SnapMirror 可實現高效的站點到站點存儲複製，從而造成災難快速、可靠且可管理的恢復能力、以符合現今的全球企業需求。SnapMirror 可透過 LAN 和 WAN 高速複寫資料、為關鍵任務應用程式提供高資料可用度和快速恢復、並提供優異的重複資料刪除和網路壓縮功能。透過 NetApp SnapMirror 技術、災難恢復可以保護整個資料中心。磁碟區可以遞增備份到異地位置。SnapMirror 會像所需的 RPO 一樣頻繁地執行遞增的區塊型複寫。區塊層級的更新可降低頻寬和時間需求、並在 DR 站台維持資料一致性。

重要步驟是建立整個資料集的一次性基準傳輸。這是執行遞增更新之前的必要條件。此作業包括在來源建立 Snapshot 複本、以及將其參照的所有資料區塊傳輸至目的地檔案系統。初始化完成後、可能會進行排程或手動觸發的更新。每個更新只會將新的和變更的區塊從來源傳輸到目的地檔案系統。此作業包括在來源 Volume 建立 Snapshot 複本、將其與基準複本進行比較、並僅將變更的區塊傳輸至目的地 Volume。新複本會成為下一次更新的基礎複本。由於複寫是定期的、SnapMirror 可以整合變更的區塊、並節省網路頻寬。對寫入處理量和寫入延遲的影響最小。

完成下列步驟即可執行恢復：

1. 連線至次要站台上的儲存系統。
2. 打破 SnapMirror 關係。
3. 將 SnapMirror 磁碟區中的 LUN 對應到次要站台上 Hyper-V 伺服器的啟動器群組（igroup）。
4. 將 LUN 對應至 Hyper-V 叢集後、請將這些磁碟設為線上。
5. 使用容錯移轉叢集 PowerShell Cmdlet、將磁碟新增至可用的儲存設備、並將其轉換為 CSV。
6. 將 CSV 中的虛擬機器匯入 Hyper-V 管理程式、使其具有高可用度、然後將其新增至叢集。
7. 開啟 VM。

## 在 NetApp 儲存設備上部署 Microsoft Hyper-V：結論

ONTAP 是部署各種 IT 工作負載的最佳共享儲存基礎。ONTAP AFF 或 ASA 平台兼具彈性與可擴充性、適用於多種使用案例和應用程式。在其中啟用 Windows Server 2022 和 Hyper-V 是虛擬化解決方案的常見使用案例、如本文件所述。ONTAP 儲存設備及其相關功能的靈活度與擴充性、讓客戶從適當大小的儲存層開始、即可隨業務需求的演進而擴充及調整。在目前的市場情況下、Hyper-V 提供完美的替代 Hypervisor 選項、可提供 VMware 提供的大部分功能。

## 在 NetApp 儲存設備上部署 Microsoft Hyper-V：移轉指令碼

本節包含可用於使用 FlexClone 移轉的 PowerShell 指令碼。

### PowerShell 指令碼

```
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 ""$directoryName"" already exists!
`n"
    }
    Else {
        Write-Error "Failed to create folder ""$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 "-----"

        $vmDiskPath = "/vol/$( $ONTAP_VOLUME_NAME )/$( $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;

```





## 版權資訊

Copyright © 2024 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變更本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

## 商標資訊

NETAPP、NETAPP 標誌及 <http://www.netapp.com/TM> 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。