



**NetApp®**

Technical Report

## OnCommand Shift Best Practices Guide

Brahmanna Chowdary Kodavali and Shashanka S R, NetApp  
June 2016 | TR-4433

## TABLE OF CONTENTS

<b>1</b>	<b>Overview .....</b>	<b>3</b>
1.1	Purpose and Intended Audience .....	3
1.2	Uses and Benefits.....	3
1.3	Architecture .....	4
<b>2</b>	<b>OnCommand Shift Server Installation and Configuration Best Practices.....</b>	<b>5</b>
2.1	OnCommand Shift Server.....	5
2.2	NetApp ONTAP Storage Virtual Machine .....	6
2.3	VMware ESX .....	7
2.4	Microsoft Hyper-V .....	7
2.5	Domain Requirement.....	8
<b>3</b>	<b>Solutions.....</b>	<b>8</b>
3.1	Virtual Machine Conversion.....	8
3.2	Disk Conversion.....	11
3.3	Queuing Virtual Machine Conversion Jobs.....	11
3.4	ESX to Hyper-V .....	12
3.5	Hyper-V to ESX .....	13
3.6	Multiple Shift Servers.....	15
3.7	vCenter and System Center Environments.....	16
<b>4</b>	<b>Conclusion .....</b>	<b>21</b>
<b>Appendix .....</b>		<b>22</b>
SAN Environments.....		22
<b>References .....</b>		<b>23</b>
<b>Version History.....</b>		<b>23</b>

## LIST OF FIGURES

Figure 1) A typical OnCommand Shift server deployment architecture.....	4
Figure 2) A typical domain requirement scenario for OnCommand Shift server deployment.....	8
Figure 3) Virtual machine conversion in a multiple OnCommand Shift server environment.....	16
Figure 4) Virtual machine conversion in vCenter and System Center environments.....	17
Figure 5) Virtual machine conversion from ESX to Hyper-V in SAN environments.....	22
Figure 6) Virtual machine conversion from Hyper-V to ESX in SAN environments.....	23

# 1 Overview

NetApp® OnCommand® Shift is a standalone product that allows you to quickly migrate virtual machines (VMs) from VMware ESXi to Microsoft Hyper-V and back. It also supports disk-level conversions between different virtual disk formats. This document presents an overview of the components and features of OnCommand Shift and also provides best practices for installing, configuring, and using this application.

## 1.1 Purpose and Intended Audience

This document is for IT administrators, solution architects, technical architects, professional service engineers, and system engineers who want to migrate VMs between VMware ESXi and Microsoft Hyper-V. NetApp OnCommand Shift offers a fast and reliable solution for the migration of VMs between these two hypervisors.

This document assumes that the reader has:

- A basic understanding of NetApp ONTAP® software operational processes
- General knowledge of hypervisor technologies (VMware ESXi and Microsoft Hyper-V)
- A basic understanding of Windows PowerShell cmdlets

NetApp technical support does not provide assistance with the scripting provided in this document. Support is only available for individual OnCommand Shift cmdlet execution.

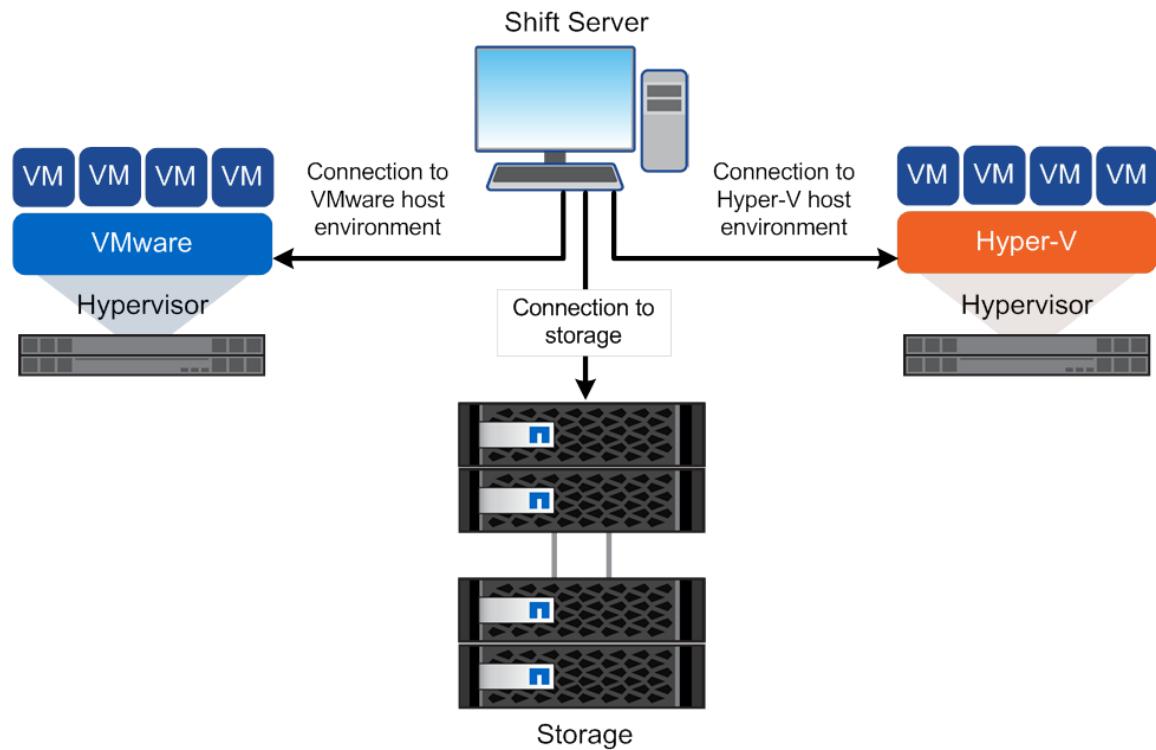
## 1.2 Uses and Benefits

NetApp OnCommand Shift is a PowerShell-based interface that enables you to migrate VMs from one hypervisor to another and to convert virtual disks from one format to another. It leverages NetApp FlexClone® technology for the rapid conversion of VM hard disks. It also handles the creation and configuration of target and destination VMs.

- OnCommand Shift provides flexibility in a multihypervisor environment by supporting bidirectional conversion between the following hypervisors:
  - VMware ESXi to Microsoft Hyper-V
  - Microsoft Hyper-V to VMware ESXi
- OnCommand Shift supports bidirectional disk-level conversions of virtual disks between hypervisors for the following disk formats:
  - VMware ESX to Microsoft Hyper-V (virtual machine disk [VMDK] to virtual hard disk [VHD] or virtual hard disk format [VHDX])
  - Microsoft Hyper-V to VMware ESX (VHD or VHDX to VMDK)
  - VMware ESX to Red Hat KVM (VMDK to qcow2)
  - Red Hat KVM to VMware ESX (qcow2 to VMDK)
- OnCommand Shift helps minimize downtime and enhances business productivity.
- OnCommand Shift offers choice and flexibility by reducing licensing costs, lock-in, and commitments to a single vendor.
- OnCommand Shift is beneficial for organizations looking to optimize VM licensing costs and extend IT budgets.
- OnCommand Shift reduces virtualization costs with data portability.
- OnCommand Shift is offered free from NetApp.
- OnCommand Shift provides fully scriptable PowerShell cmdlets that can be leveraged by automation workflows.
- OnCommand Shift replaces the MAT4Shift tool.

## 1.3 Architecture

Figure 1) A typical OnCommand Shift server deployment architecture.



OnCommand Shift works by connecting OnCommand Shift servers to Microsoft Hyper-V, VMware ESXi, and shared storage. OnCommand Shift leverages FlexClone to convert VM hard drives from one hypervisor to another by using three key NetApp technologies:

- **Single volume and multiple protocols.** With NetApp ONTAP, you can use multiple protocols to access a single volume. For example, VMware ESXi can access a volume that is enabled by OnCommand Shift with the Network File System (NFS) protocol, and Microsoft Hyper-V can access the same volume with the CIFS protocol.
- **FlexClone technology.** FlexClone allows the rapid cloning of entire files or volumes with no data copy. Common blocks on the storage system are shared between multiple files or volumes. As a result, large VM disks can be cloned very quickly.
- **VM disk conversion PowerShell cmdlets.** The NetApp PowerShell Toolkit and OnCommand Shift contain a large number of PowerShell cmdlets that you can use to perform various actions on a NetApp storage controller. Included are PowerShell cmdlets that convert virtual disks to different formats. For example, VMware VMDK can be converted to Microsoft VHDX or VHD, and vice versa. These conversions are performed with FlexClone, which enables very rapid cloning and conversion of disk formats in one step.

## 2 OnCommand Shift Server Installation and Configuration Best Practices

### 2.1 OnCommand Shift Server

The OnCommand Shift server is the Microsoft Hyper-V server or a standalone Windows server (physical or virtual) where OnCommand Shift is installed. The OnCommand Shift server executes OnCommand Shift operations by using PowerShell to communicate between VMware ESXi and Microsoft Hyper-V. For supported server hosts, see the NetApp [Interoperability Matrix](#).

To install and configure NetApp OnCommand Shift, follow the [NetApp OnCommand Shift Installation and Setup Guide](#). For more information about cmdlets supported by OnCommand Shift, see the [NetApp OnCommand Shift Cmdlet Reference Guide](#).

Before installing OnCommand Shift, review the following best practices:

- Review the [NetApp OnCommand Shift Installation and Setup Guide](#) to see all system, connectivity, software, hardware, and licensing requirements. Also listed are the supported hypervisors, the hypervisor OSs and switches, disk formats, and the guest OS.
- To verify that the Windows PowerShell version on the Shift server is 3.0 or 4.0, run the following PowerShell cmdlet on the OnCommand Shift server:

```
# Below command must show 3.0 or 4.0 for "PSVersion" field
$PSVersionTable
```

- To verify that Microsoft .NET Framework 3.5 is installed on the OnCommand Shift server, run the following PowerShell cmdlet on the OnCommand Shift server:

```
# Below command must will list the .NET Framework versions installed on the machine. Make
# sure that the version field shows both 3.5 and 2.0
Get-ChildItem 'HKLM:\SOFTWARE\Microsoft\NET Framework Setup\NDP' -recurse | Get-
ItemProperty -name Version,Release -EA 0 | Where { $_.PSChildName -match '^(!?S)\p{L}'} | Select PSChildName, Version, Release
```

- To verify that VMware vSphere PowerCLI 5.5 is installed on the OnCommand Shift server, run the following PowerShell cmdlets on the OnCommand Shift Server:

```
# Below command must show 5.5 as the PowerCLI version
Add-PSSnapin VMware.VimAutomation.Core
Get-PowerCLIVersion
```

- OnCommand Shift can be installed on a Microsoft Hyper-V server or on a stand-alone server (physical or virtual), and it is a best practice to install OnCommand Shift on its own VM. This approach allows you to target different Microsoft Hyper-V servers or VMware ESXi servers with a single OnCommand Shift server and a simple PowerShell-based configuration change.
- OnCommand Shift supports bidirectional migration of VMs between VMware ESXi and Microsoft Hyper-V. However, it is currently limited to communication between only one ESXi environment and one Hyper-V environment (without reconfiguration).

After OnCommand Shift is installed, review the following best practices before converting VMs with OnCommand Shift:

- To verify that Windows remote management is enabled and the WinRM service is running on the OnCommand Shift server, run the following PowerShell cmdlet on the OnCommand Shift server:

```
# Below command must return "WinRM is already set up for remote management on this
# computer"
winrm quickconfig
```

- To verify that the OnCommand Shift server is configured as a CredSSP client and the Hyper-V server is added as the delegated computer and trusted host on the OnCommand Shift server, run the following PowerShell cmdlet on the OnCommand Shift server:

```
# Below command must show that the Shift Server is configured as CredSSP client and
# allows delegating fresh credentials to the Hyper-V host
Get-WSManCredSSP
```

```
# Below command must show "CredSSP=true" and the "Trusted Hosts" on Shift Server
winrm get winrm/config/client
```

If these configurations are not set, then set them by running the following PowerShell cmdlet on the OnCommand Shift server:

```
# Below command enables Shift Server as CredSSP client and allows delegating fresh
credentials to the Hyper-V server
Enable-WSManCredSSP -role client -DelegateComputer <"input FQDN of hyper-v server">
# Below command adds the Hyper-V server as the trusted host on Shift Server
Set-Item WSMAN:\localhost\Client\TrustedHosts <"input FQDN of hyper-v server">
```

- To make sure that the execution policy is set to unrestricted so that the cmdlets can be executed in unrestricted mode, run the following command:

```
# Below command must show "Unrestricted"
Get-ExecutionPolicy
```

If this policy is not set correctly, then configure it by running the following PowerShell cmdlet on the OnCommand Shift server:

```
Set-ExecutionPolicy Unrestricted
```

- Make sure that the Apache Tomcat 7.0 service is running on the OnCommand Shift server:

```
# Below command must show "Running" in the status field
Get-Service | Where-Object{$_ .Name -eq "Tomcat7"}
```

If it is not running, then start the Apache Tomcat service in Control Panel > Services on the OnCommand Shift server.

- To verify that the OnCommand Shift server is connected and running, run the following PowerShell cmdlet on the Shift server:

```
# Below command must show "Shift Server is Running"
Get-ShiftServerState
```

## 2.2 NetApp ONTAP Storage Virtual Machine

An ONTAP SVM (Storage Virtual Machine, formerly known as vServer) is a logical storage server that provides data access to LUNs and/or a NAS namespace from one or more LIFs. For more information on SVMs, see the [ONTAP System Administration Guide](#).

- Although OnCommand Shift permits the use of an existing SVM with some configuration changes, it is a NetApp best practice to create a new SVM. You should also use vSphere Storage vMotion or Hyper-V Storage Live Migration to migrate VMs onto this newly created SVM. This migration is performed without downtime for the VM. With this approach, the VMs that are being migrated do not reside on the production SVM. The new SVM can be created with any of the following methods:
  - The Configuration wizard or PowerShell cmdlets provided by OnCommand Shift
  - NetApp OnCommand System Manager
  - CLI commands on the NetApp storage
  - The NetApp PowerShell Tool Kit
- OnCommand Shift only supports the conversion of VMs residing in a NAS environment (NFS/CIFS). It does not support the conversion of VMs residing in SAN environments (LUNs). To learn more about the SAN environment scenario, see the section ["SAN Environments."](#)

Follow these best practices for the ONTAP SVM used for the OnCommand Shift server:

- To verify that the NetApp controller on which the SVM resides has valid licenses for CIFS, NFS, and FlexClone, run the following PowerShell cmdlet on the OnCommand Shift server:

```
# Below command must show "CIFS, NFS and FlexClone" licenses listed in the Package field
$cclusterManagementIP = "<input cluster management IP Address of the controller>"
#-----
Import-Module DataONTAP
Connect-NcController -Name $cclusterManagementIP
Get-NcLicense
```

- Before configuring CIFS on the ONTAP SVM, make sure that the time settings are synchronized between the Windows domain controller and NetApp storage controller. NetApp recommends having a time skew between the Windows domain controller and NetApp storage controller of not more than five minutes. It is a best practice to configure the NTP server for the ONTAP cluster to synchronize with an external time source. To configure the Windows domain controller as the NTP server, run the following command on your ONTAP cluster:

```
$domainControllerIP = "<input IP Address of windows domain controller>"  
system services ntp server create -server $domainControllerIP
```

- Make sure that the CIFS/NFS share (used to store the VMs to be converted) and the destination share (used to store the converted VMs) reside on the same volume. OnCommand Shift does not support spanning on multiple volumes. To do so, run the following PowerShell cmdlets on the OnCommand Shift server:

```
# Import the Data ONTAP Module and connect to the Controller  
$clusterManagementIP = "<input cluster management IP Address of the controller>"  
$vserverIP = "<input name of the vserver used for OnCommand Shift operations>"  
#-----  
Import-Module DataONTAP  
Connect-NcController -Name $clusterManagementIP -vserver $vserverIP  
# Get the NFS export name  
Get-NcNfsExport  
# Get the CIFS share name  
Get-Nccifsshare  
# Make sure the "Pathname" field in NFS export and "Path" field in CIFS share has same  
parent volume
```

## 2.3 VMware ESX

- OnCommand Shift supports ESXi version 5.0 and later, so make sure that the ESX server is not running an ESXi OS earlier than version 5.0. To check the version number, run the following PowerShell cmdlets on the OnCommand Shift server:

```
# Below command must show 5.0.0 or above as the ESXi version  
$esxhost = "<input IP address of ESX Host>"  
$esxUserName = "<input username for ESX Host>"  
$esxPassword = "<input password for ESX Host>"  
#-----  
Add-PSSnapin VMware.VimAutomation.Core  
Connect-VIServer -Server $esxhost -User $esxUserName -password $esxPassword  
(Get-VMHost).version
```

- Provide the FQDN or IP address of a stand-alone ESX server when configuring ESX settings for OnCommand Shift. You must do this because OnCommand Shift:
  - Can only communicate with a single ESX server
  - Does not support ESX clusters
  - Does not communicate with vCenter directly

To learn more about using OnCommand Shift in a vCenter environment, see the section [“vCenter and System Center Environments.”](#)

## 2.4 Microsoft Hyper-V

- To enable Windows remote management and run the WinRM service on the Hyper-V server, run the following PowerShell cmdlet on the Hyper-V server:

```
# Below command must return "WinRM is already set up for remote management on this  
computer"  
winrm quickconfig
```

- Make sure that the Hyper-V server is configured as a CredSSP server:

```
# Below command must show that the Hyper-V host is configured as CredSSP server and  
receives credentials from a remote client computer  
Get-WsManCredSSP
```

If it is not so configured, then run the following PowerShell cmdlet on the Hyper-V server:

```
# Below command enables Hyper-V server as CredSSP server and receives credentials from a
remote client computer
Enable-WSManCredSSP -role Server
```

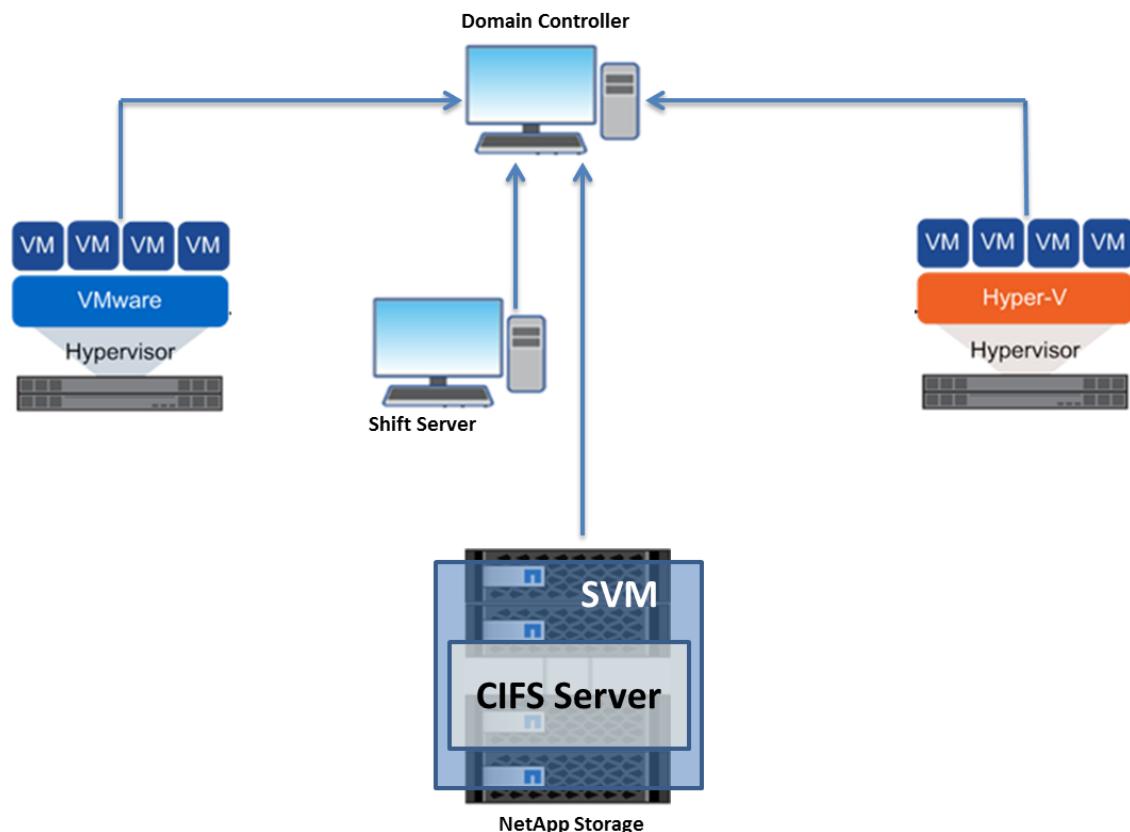
- Provide the FQDN or IP address of a stand-alone Hyper-V server when configuring Hyper-V settings for OnCommand Shift because OnCommand Shift:
  - Can only communicate with a single Hyper-V server
  - Does not support Hyper-V clusters
  - Does not communicate with System Center directly

To learn more about using OnCommand Shift in a System Center environment, see the section [“vCenter and System Center Environments.”](#)

## 2.5 Domain Requirement

Place the OnCommand Shift server, the Hyper-V server, the ESX server, and the ONTAP CIFS servers on the same Windows domain, as is depicted in Figure 2.

Figure 2) A typical domain requirement scenario for OnCommand Shift server deployment.



## 3 Solutions

### 3.1 Virtual Machine Conversion

OnCommand Shift supports bidirectional VM conversion between VMware ESX and Microsoft Hyper-V. To convert VMs with NetApp OnCommand Shift, follow the [NetApp OnCommand Shift Conversion and Administration Guide](#). The OnCommand Shift 1.2 UI wizard provides new options for VM conversion and

job tracking. The VM Conversion page provides a list of all VMs available for conversion based on the conversion direction. The Job Tracking page enables you to track or abort OnCommand Shift jobs.

Before converting VMs with OnCommand Shift, follow these best practices:

- Move the VMs you would like to convert to the newly created ONTAP SVM before conversion. Use an NFS share for Hyper-V to ESX conversion and a CIFS share for ESX to Hyper-V conversion. To verify the location of the virtual disks for all of the VMs on a hypervisor, run the following PowerShell cmdlets on the OnCommand Shift server:

- Verify Hyper-V VMs.

```
# Please Fill in the details for entries marked in red
$hypervhost = "<input FQDN of Hyper-V Host>"
$hypervusername = "<input UserName of Hyper-V Host>"
$hypervpassword = "<input Password of Hyper-V Host>"
#-----
# Create Hyper-V Credential Object
$hypervcredpwd = ConvertTo-SecureString -AsPlainText -Force $hypervpassword
$hypervcred = new-object management.automation.pscredential $hypervusername,
$hypervcredpwd
# Run a remote command on Hyper-V Host to query the location of the virtual disks of all
# the VMs
Invoke-Command -ComputerName $hypervhost -Credential $hypervcred -ScriptBlock {(Get-VM |
Select-Object VMID | Get-VHD).Path}
```

- Verify ESX VMs.

```
# Please Fill in the details for entries marked in red
$esxhost = "<input FQDN of ESX Host>"
$esxusername = "<input UserName of ESX Host>"
$esxpassword = "<input Password of ESX Host>"
#-----
# Create ESX Credential Object
$esxcredpwd = ConvertTo-SecureString -AsPlainText -Force $esxpassword
$esxcred = new-object management.automation.pscredential $esxusername, $esxcredpwd
# Add VMware Snapin to PowerShell and Connect to ESX Server
$addsnapin = Add-PSSnapin VMware.VimAutomation.Core
$connect = Connect-VIServer -Server $esxhost -User $esxUserName -password $esxPassword
# Run a remote command on ESX Host to query the location of the virtual disks of all the
# VMs
(Get-VM | Get-HardDisk ).filename
```

- Make sure that the destination path (where the converted VMs are stored) is set to an NFS share when you are converting VMs from Hyper-V to ESX. Set the destination path to a CIFS share when you are converting VMs from ESX to Hyper-V. OnCommand Shift allows the following two destination paths to be set simultaneously:
  - CIFS share path for Hyper-V
  - NFS share path for ESX

To verify the destination paths, run the following OnCommand Shift PowerShell cmdlet on the OnCommand Shift server.

```
# Make sure you connected to the Shift Server by running "Connect-ShiftServer" before
running the below cmdlet
Get-ShiftDestinationPathConfig
```

- Make sure that the VM you would like to convert is not powered off before running the conversion command because OnCommand Shift runs certain scripts on the VM to get VM and guest OS information. To identify the currently powered-on VMs on a hypervisor, run the following PowerShell cmdlets on the OnCommand Shift server:

- Verify Hyper-V VMs.

```
# Please Fill in the details for entries marked in red
$hypervhost = "<input FQDN of Hyper-V Host>"
$hypervusername = "<input UserName of Hyper-V Host>"
$hypervpassword = "<input Password of Hyper-V Host>"
#-----
# Create Hyper-V Credential Object
$hypervcredpwd = ConvertTo-SecureString -AsPlainText -Force $hypervpassword
$hypervcred = new-object management.automation.pscredential $hypervusername,
$hypervcredpwd
# Run a remote command on Hyper-V Host to query the list of powered on VMs
```

```
Invoke-Command -ComputerName $hypervhost -Credential $hypervcred -ScriptBlock {(Get-VM | Where-Object {$_.State -eq "running"}).VMName}
```

- Verify ESX VMs.

```
# Please Fill in the details for entries marked in red
$esxhost = "<input FQDN of ESX Host>"
$esxusername = "<input UserName of ESX Host>"
$esxpassword = "<input Password of ESX Host>"
#-----
# Create ESX Credential Object
$esxcredpwd = ConvertTo-SecureString -AsPlainText -Force $esxpassword
$esxcred = new-object management.automation.pscredential $esxusername, $esxcredpwd
# Add VMware Snapin to PowerShell and Connect to ESX Server
$addsnapin = Add-PSSnapin VMware.VimAutomation.Core
$connect = Connect-VIServer -Server $esxhost -User $esxUserName -password $esxPassword
# Run a remote command on ESX Host to query the list of powered on VMs
(Get-VM | Where-Object {$_.PowerState -eq "PoweredOn"}).Name
```

- Make sure that the VM guest OS satisfies the following requirements:
  - The guest OS must have PowerShell version 2.0 or later.
  - The guest user specified for conversion must have elevated access on the Windows guest OS.
- If the conversion direction is from ESX to Hyper-V, then complete the following steps:
  - To make sure that VMware tools are installed and running on the ESX VM, run the following PowerShell cmdlets on the OnCommand Shift server:

```
# Please Fill in the details for entries marked in red
$esxhost = "<input FQDN of ESX Host>"
$esxusername = "<input UserName of ESX Host>"
$esxpassword = "<input Password of ESX Host>"
#-----
# Create ESX Credential Object
$esxcredpwd = ConvertTo-SecureString -AsPlainText -Force $esxpassword
$esxcred = new-object management.automation.pscredential $esxusername, $esxcredpwd
# Add VMware Snapin to PowerShell and Connect to ESX Server
$addsnapin = Add-PSSnapin VMware.VimAutomation.Core
$connect = Connect-VIServer -Server $esxhost -User $esxUserName -password $esxPassword
# Run a remote command on ESX Host to query the VMware tools status for all the VMs
Get-VM | Format-Table @{{Expression={$_.Name};Label="VM"}, @{{Expression=($_ | Get-View).Guest.ToolsRunningStatus};Label="Status"}}
```

- By default, OnCommand Shift converts the ESX VM to a generation 1 Hyper-V VM. You can not specify the generation type for the destination Hyper-V VMs because OnCommand Shift does not support the creation of generation 2 Hyper-V VMs.
- If the conversion direction is from Hyper-V to ESX, then complete the following steps:
  - Provide the IP address of the VM in the conversion command, because OnCommand Shift must connect to the VM and run remote PowerShell scripts there.
  - Make sure that the VM is generation 1, because OnCommand Shift does not support the conversion of generation 2 Hyper-V VMs.
  - Make sure that the VM is not using a shared VHD, because ONTAP does not support shared VHDs on an SMB 3.0 stack. Therefore, OnCommand Shift also does not support them.
  - Make sure that the destination NFS data store (which stores the converted VMs) is mounted on the ESX server before running the conversion command. To check the NFS datastore mounted on the ESX server, run the following PowerShell cmdlets on the OnCommand Shift server:

```
# Please Fill in the details for entries marked in red
$esxhost = "<input FQDN of ESX Host>"
$esxusername = "<input UserName of ESX Host>"
$esxpassword = "<input Password of ESX Host>"
#-----
# Create ESX Credential Object
$esxcredpwd = ConvertTo-SecureString -AsPlainText -Force $esxpassword
$esxcred = new-object management.automation.pscredential $esxusername, $esxcredpwd
# Add VMware Snapin to PowerShell and Connect to ESX Server
$addsnapin = Add-PSSnapin VMware.VimAutomation.Core
$connect = Connect-VIServer -Server $esxhost -User $esxUserName -password $esxPassword
# Run a remote command on ESX Host to query the NFS Data store mounted on the ESX host,
# Make sure that the returned Data Store path contains the Destination path returned by the
```

```
"Get-ShiftDestinationPathConfig" cmdlet of the Shift Server  
(Get-Datastore).RemotePath
```

- OnCommand Shift can currently only convert one VM at a time. However, the program can queue 300 VM conversion jobs at a time, with VM conversions running one at a time. To learn more about the queuing of VM conversions, see the section [“Queuing VM Conversion Jobs.”](#)

## 3.2 Disk Conversion

OnCommand Shift supports Shift Foundation REST APIs and cmdlets that enables bidirectional, disk-level conversions of virtual disks between hypervisors for the following disk formats:

- VMware ESX to Microsoft Hyper-V (VMDK to VHDX)
- Microsoft Hyper-V to VMware ESX (VHDX to VMDK)
- VMware ESX to Red Hat KVM (VMDK to qcow2)
- Red Hat KVM to VMware ESX (qcow2 to VMDK)

To convert the virtual disks with NetApp OnCommand Shift, see the [NetApp OnCommand Shift Conversion and Administration Guide](#).

Shift Foundation REST APIs and cmdlets can be integrated into third party products and scripts. For more information, see the “*Overview of Shift Foundation Disk Conversion REST APIs*” in the Conversion and Administration Guide.

**Note:** Shift Foundation REST APIs only support disk conversions. They do not support VM conversion. To use the converted disk in a VM, the VM must be created manually and the disk must be attached to it.

**Note:** OnCommand Shift 1.2 does not support VM-level conversions for the KVM hypervisor. However, it does support disk conversions to and from the QCOW2/QCOW3 disk format, a virtual disk format used by the KVM hypervisor.

## 3.3 Queuing Virtual Machine Conversion Jobs

The queuing of conversion jobs can be automated with a simple PowerShell script that queries the VMs from the source hypervisor and saves them in a local file. The file must also indicate whether an individual VM is to be converted. You can do this by entering either `yes` or `no` for each VM. The file is then read and a conversion command is issued for all VMs tagged with `yes`.

The following sections provide automation scripts for queuing VMs for conversion. It is assumed that the user has installed and configured OnCommand Shift.

- Provide the following information prior to running the PowerShell script:
  - The FQDN, username, and password for the source hypervisor
  - The password for the OnCommand Shift server
  - The domain, user name, and password for the guest OS of the VM
- Upon execution, the script fetches all of the VMs that are powered on from the source hypervisor and saves them in a CSV file. The CSV file contains the following five fields, which the user can update as needed:
  - vmname.** The name of the VM.
  - guestOSDomain.** The domain of the guest OS.
  - guestOSUserName.** The user credential information for the guest OS.
  - guestOSPassword.** The user credential information for the guest OS.
  - Convert.** Takes a value of `yes` or `no`, indicating whether or not to convert a particular VM

The guestOSDomain, guestOSUserName and guestOSPassword fields are filled with the values provided by the user at the beginning of the script. If required, the user can change these values in the CSV file.

The Convert field is filled with the default option yes for all of the VMs, meaning that all the VMs are queued for conversion. If you do not want to convert certain VMs, then modify this value to no or leave it empty for that particular VM.

3. The script must receive confirmation from the user before proceeding with the conversion process. After the conversion has started, all of the conversion jobs are queued and the unique job ID for each conversion job is saved with the name of the VM in another CSV file.

**Note:** Save the following scripts with the .ps1 extension and run them from Windows PowerShell.

**Note:** The CSV files are generated by the script in the same location where the script is run.

**Note:** NetApp technical support does not provide assistance with the scripting provided in this document. Support is only available for individual OnCommand Shift cmdlets.

### 3.4 ESX to Hyper-V

```
# This script will query the VMs from a ESX server and saves the information in a csv file
# The user selects the VMs to be converted by editing the csv file with either Yes/No option
# The script will convert the selected VMs to Hyper-V
#-----
# Please Fill in the details for entries marked in red
$esxhost = "<input FQDN of ESX Host>"  
$esxUserName = "<input UserName of ESX Host>"  
$esxPassword = "<input Password of ESX Host>"  
$shiftServerUserName = "<input UserName of Shift Server>"  
$shiftServerPassword = "<input Password of Shift Server>"  
$guestOSDomain = "<input Domain of Guest OS>"  
$guestOSUserName = "<input UserName of Guest OS>"  
$guestOSPassword = "<input Password of Guest OS>"  
#-----
# Change the path for CSV files to store VM list and Conversion status if required
$VMsListFilePath = "$PSScriptRoot\VMsList.csv"  
$VMsConversionStatusFilePath = "$PSScriptRoot\VMsConversionStatus.csv"  
#-----
# Create arrays to hold VMs and Conversion Status
$vmarray= @()  
$vmstatusarray= @()  
# Delete contents of the files
If(Test-Path $VMsListFilePath) {Clear-Content $VMsListFilePath}  
If(Test-Path $VMsConversionStatusFilePath) {Clear-Content $VMsConversionStatusFilePath}  
# Create Shift Server Credential object
$shiftServerPassword = ConvertTo-SecureString $shiftServerPassword -AsPlainText -Force
$shiftServerCredential = new-object management.automation.pscredential  
$shiftServerUserName,$shiftServerPassword
# Add VMware Snapin to PowerShell and Connect to ESX Server
$addsnapin = Add-PSSnapin VMware.VimAutomation.Core
$connect = Connect-VIServer -Server $esxhost -User $esxUserName -password $esxPassword
# Run a remote command on ESX Host to query the VMs
$esxVMs = (Get-VM | Where-Object {$__.PowerState -eq "Poweredon"}).Name
# Disconnect from ESX Server
$disconnect = Disconnect-VIServer -Server $esxhost -Confirm:$false
# Create a PowerShell Object for each VM with required information
ForEach ($vm In $esxVMs)
{
    $vmarrayitem =
    [pscustomobject]@{vmname=$vm;guestOSDomain=$guestOSDomain;guestOSUserName=$guestOSUserName;guestOSPassword=$guestOSPassword;convert="yes"}
    $vmarray += $vmarrayitem
}
# Export the PowerShell objects created to a CSV File and open it for editing
$vmarray | Export-Csv $VMsListFilePath
notepad.exe $VMsListFilePath
# A small menu for confirmation to convert the selected VMs
$title = "Convert VMs"
$message = "Do you want to convert the selected VMs?"
$yes = New-Object System.Management.Automation.Host.ChoiceDescription "&Yes", "Converts all the VMs selected"
$no = New-Object System.Management.Automation.Host.ChoiceDescription "&No", "Exits the script"
```

```

$options = [System.Management.Automation.Host.ChoiceDescription[]]($yes, $no)
$result = $host.ui.PromptForChoice($title, $message, $options, 0)
If ($result -eq 0)
{
    Write-Host "You selected Yes."
}
else
{
    Write-Host "You selected No.Exiting the script."
    Exit
}
# Connect to the Shift Server
Connect-ShiftServer -credential $shiftServercredential
# Import the CSV File and for each VM issue conversion command
$VmList = Import-Csv $VmListFilePath
ForEach ($line In $VmList)
{
    # Query the details from CSV file
    $VmName = $line.VmName
    $VmDomain = $line.GuestOSDomain
    $VmUsername = $line.GuestOSUserName
    $VmPassword = $line.GuestOSPassword
    $VmConvert = $line.Convert
    # Create Guest OS Credential Object
    $GoscredPwd = ConvertTo-SecureString -AsPlainText -Force $VmPassword
    $Goscred = New-Object Management.Automation.PSCredential $VmUsername, $GoscredPwd
    # Issue the conversion command for the selected VM
    If ($VmConvert -eq "Yes" -or $VmConvert -eq "yes")
    {
        $ShiftJobId = Convert-ShiftVirtualMachine -source esx -destination hyperv
        -name $VmName -credential $Goscred -domain $VmDomain -force
        Write-Host "$VmName queued for conversion"
        # Create a PowerShell object for each VM with conversion job id
        $VmStatusArrayItem =
        [PSCustomObject]@{VmName=$VmName;ShiftJobId=$ShiftJobId;Status=""}
        $VmStatusArray += $VmStatusArrayItem
    }
}
# Export the PowerShell objects created to a CSV File
$VmStatusArray | Export-Csv $VmConversionStatusFilePath
# Disconnect from Shift Server
Disconnect-ShiftServer

```

### 3.5 Hyper-V to ESX

```

# This script will query the VMs from a Hyper-V server and saves the information in a csv
# file
# The user selects the VMs to be converted by editing the csv file with either Yes/No
# option
# The script will convert the selected VMs to ESX
#-----
# Please Fill in the details for entries marked in red
$HyperVHost = "<input FQDN of Hyper-V Host>"
$HyperVUsername = "<input UserName of Hyper-V Host>"
$HyperVPassword = "<input Password of Hyper-V Host>"
$ShiftServerUserName = "<input UserName of Shift Server>"
$ShiftServerPassword = "<input Password of Shift Server>"
$GuestOSDomain = "<input Domain of Guest OS>"
$GuestOSUserName = "<input UserName of Guest OS>"
$GuestOSPassword = "<input Password of Guest OS>"
#-----
# Change the path for CSV files to store VM list and Conversion status if required
$VmListFilePath = "$PSScriptRoot\VmList.csv"
$VmConversionStatusFilePath = "$PSScriptRoot\VmConversionStatus.csv"
#-----
# Create arrays to hold VMs and Conversion Status
$VmArray = @()
$VmStatusArray = @()
# Delete contents of the files
If (Test-Path $VmListFilePath) {Clear-Content $VmListFilePath}
If (Test-Path $VmConversionStatusFilePath) {Clear-Content $VmConversionStatusFilePath}
# Create Shift Server Credential object
$ShiftServerPassword = ConvertTo-SecureString $ShiftServerPassword -AsPlainText -Force
$ShiftServerCredential = New-Object Management.Automation.PSCredential
$ShiftServerUserName, $ShiftServerPassword
# Create Hyper-V Credential object
$HyperVcredPwd = ConvertTo-SecureString -AsPlainText -Force $HyperVPassword
$HyperVcred = New-Object Management.Automation.PSCredential $HyperVUsername,
$HyperVcredPwd

```

```

# Run a remote command on Hyper-V Host to query the VMs
$hypervVMs = Invoke-Command -ComputerName $hypervhost -Credential $hypervcred -
ScriptBlock {(Get-VM | Where-Object {$_.State -eq "running"}).VMName}
# Create a PowerShell object for each VM with required information
ForEach ($vm In $hypervVMs)
{
    $vmarrayitem =
[pscustomobject]@{vmname=$vm;guestOSDomain=$guestOSDomain;guestOSUserName=$guestOSUserName;guestOSPassword=$guestOSPassword;convert="yes"}
$vmarray += $vmarrayitem
}
# Export the PowerShell objects created to a CSV File and open it for editing
$vmarray | Export-Csv $VMsListFilePath
notepad.exe $VMsListFilePath
# A small menu for confirmation to convert the selected VMs
$title = "Convert VMs"
$message = "Do you want to convert the selected VMs?"
$yes = New-Object System.Management.Automation.Host.ChoiceDescription "&Yes", "Converts all the VMs selected"
$no = New-Object System.Management.Automation.Host.ChoiceDescription "&No", "Exits the script"
$options = [System.Management.Automation.Host.ChoiceDescription[]]($yes, $no)
$result = $host.ui.PromptForChoice($title, $message, $options, 0)
If ($result -eq 0)
{
    Write-Host "You selected Yes."
}
else
{
    Write-Host "You selected No. Exiting the script."
    Exit
}
# Connect to the Shift Server
Connect-ShiftServer -credential $shiftServercredential
# Import the CSV File and for each VM issue conversion command
$vmList = Import-Csv $VMsListFilePath
ForEach ($line In $vmList)
{
    # Query the details from CSV file
    $vmname = $line.vmname
    $vmdomain = $line.guestOSDomain
    $vmusername = $line.guestOSUserName
    $vmpassword = $line.guestOSPassword
    $vmconvert = $line.convert
    # Get the IP Address of the VM
    $gosip = Invoke-Command -ComputerName $hypervhost -Credential $hypervcred -
ScriptBlock {(Get-VMNetworkAdapter -VMName $using:vmname).IPAddresses[0]}
    # Create Guest OS Credential Object
    $goscredpwd = ConvertTo-SecureString -AsPlainText -Force $vmpassword
    $goscred = new-object management.automation.pscredential $vmusername, $goscredpwd
    # Issue the conversion command for the selected VM
    If ($vmconvert -eq "Yes" -or $vmconvert -eq "yes")
    {
        $shiftjobid = Convert-ShiftVirtualMachine -source hyperv -destination esx -name $vmname -ipAddress $gosip -credential $goscred -domain $vmdomain -force
        Write-Host "$vmname queued for conversion"
        # Create a PowerShell object for each VM with conversion job id
        $vmstatusarrayitem =
[pscustomobject]@{vmname=$vmname;shiftjobid=$shiftjobid}
        $vmstatusarray += $vmstatusarrayitem
    }
}
# Export the PowerShell objects created to a CSV File
$vmstatusarray | Export-Csv $VMsConversionStatusFilePath
# Disconnect from Shift Server
Disconnect-ShiftServer

```

## Checking Conversion Status

The following script reads the conversion status file generated in the previous scripts and provides the conversion status for each of the VMs being converted.

```

# This script will check the conversion status of the VMs being converted
#-----
# Please Fill in the details for entries marked in red
$shiftServerPassword = "<input Password of Shift Server>"
#-----

```

```

# Use the path for csv file given in the previous script to store the Conversion status
$VMsConversionStatusFilePath = "$PSScriptRoot\VMsConversionStatus.csv"
# Create array to hold Conversion Status
$vmstatusarray= @()
# Create Shift Server Credential object
$shiftServerPassword = ConvertTo-SecureString $shiftServerPassword -AsPlainText -Force
$shiftServercredential = new-object management.automation.pscredential
"admin", $shiftServerPassword
# Connect to the Shift Server
Connect-ShiftServer -credential $shiftServercredential
# Import the CSV File and for each VM get the conversion job status. Modify it according
# to your needs
$statuslist = Import-Csv $VMsConversionStatusFilePath
ForEach ($line In $statuslist)
{
    $vmname = $line.vmname
    $shiftjobid = $line.shiftjobid

    $shiftjobstatus = Get-ShiftJobstatus $shiftjobid | out-String
    if([string]::IsNullOrEmpty($shiftjobstatus))
    {
        $status = "Queued"
    }
    elseif($shiftjobstatus.Contains("FAILED"))
    {
        $status = "Failed"
    }
    elseif($shiftjobstatus.Contains("Migration Complete"))
    {
        $status = "Complete"
    }
    elseif($shiftjobstatus.Contains("Migrate Virtual Machine"))
    {
        $status = "Converting"
    }
    #Create a PowerShell object for each VM with conversion job status
    $vmstatusarrayitem = [pscustomobject]@{vmname=$vmname;status=$status}
    $vmstatusarray += $vmstatusarrayitem
}
# Output the status
$vmstatusarray
# Disconnect from Shift Server
Disconnect-ShiftServer

```

### 3.6 Multiple Shift Servers

Although OnCommand Shift can migrate bidirectionally between VMware ESXi and Microsoft Hyper-V, it can currently only communicate between one ESXi environment and one Hyper-V environment using a ONTAP volume. However, if multiple OnCommand Shift servers must be connected to the same Hyper-V server and ESX server, then each OnCommand Shift server must be configured with a different ONTAP volume. Figure 3 depicts a scenario in which multiple OnCommand Shift servers are connected to the same pair of hypervisors using different ONTAP volumes.

Figure 3) Virtual machine conversion in a multiple OnCommand Shift server environment.

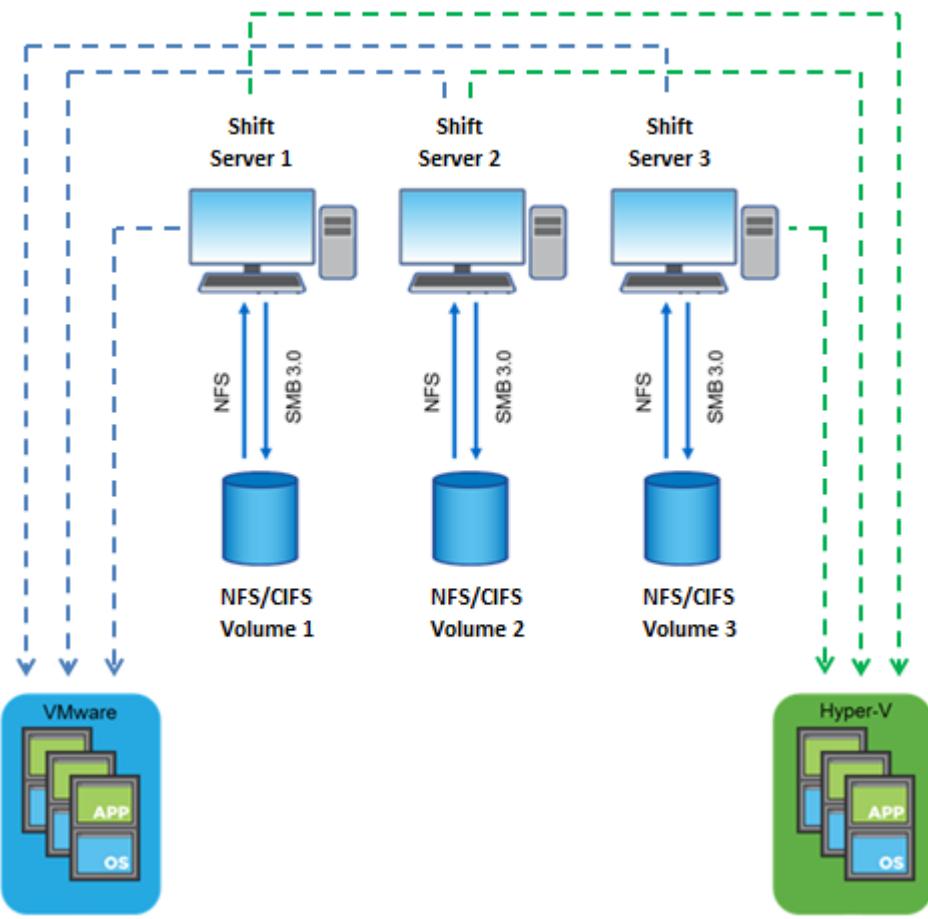
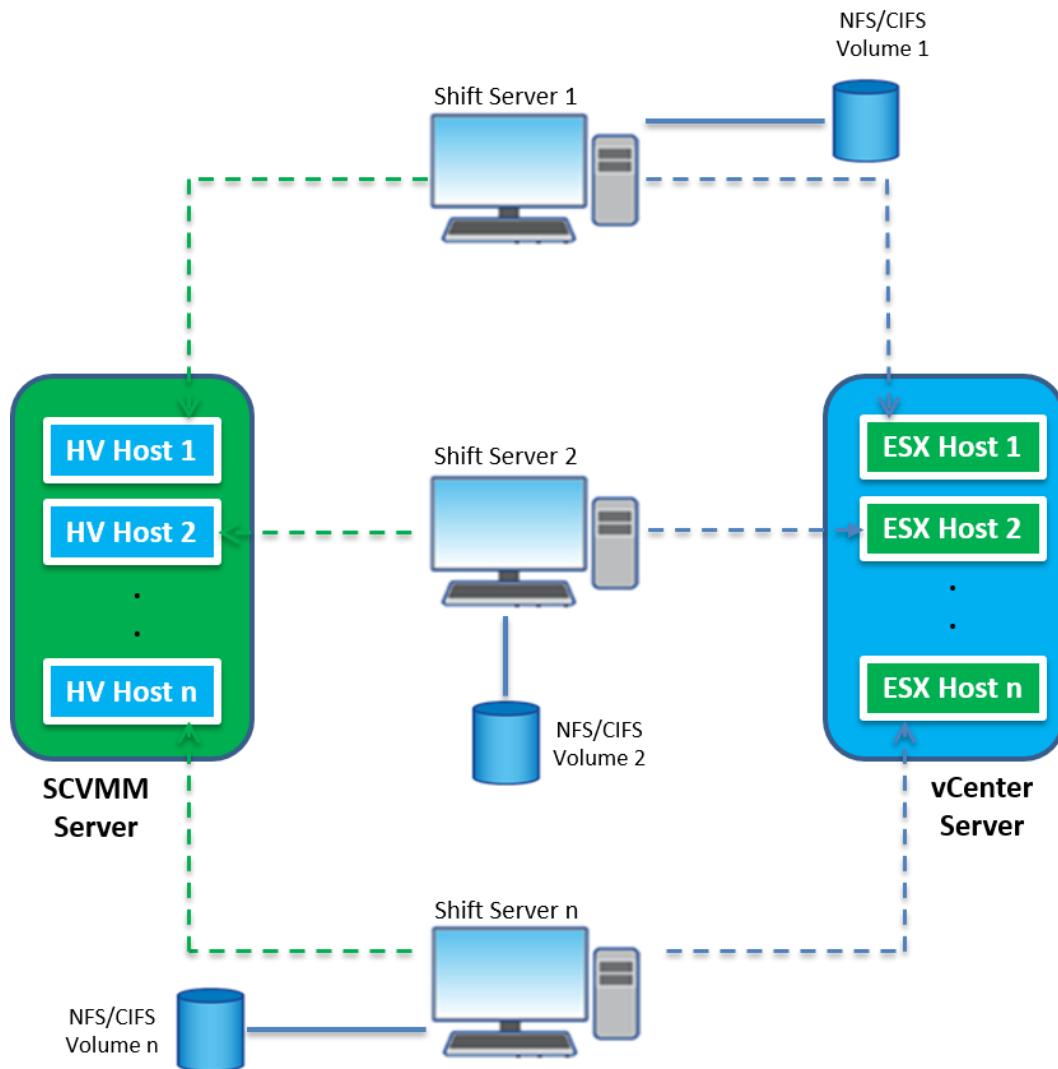


Figure 3 depicts the deployment of three shift servers connected to same set of Hyper-V and ESX servers. In this configuration, each OnCommand Shift server must be connected to a different ONTAP volume. These volumes can exist on the same SVM or on different SVMs. However, make sure that the path for the VMs to be converted on the source hypervisor and the destination path for the converted VMs on the destination hypervisor reside on the same volume.

### 3.7 vCenter and System Center Environments

As a key requirement for OnCommand Shift, the hypervisor details configured for the OnCommand Shift server must be for a single Hyper-V server and a single ESX server. The OnCommand Shift server cannot be configured with vCenter or System Center environments. To use an OnCommand Shift server in these environments, you must create multiple OnCommand Shift servers and configure them with a pair of hypervisors, as shown in Figure 4.

Figure 4) Virtual machine conversion in vCenter and System Center environments.



As is shown in Figure 4, an individual OnCommand Shift server is configured for each pair of Hyper-V and ESX servers in the environment. To automate this procedure, complete the following steps:

1. Query the Hyper-V hosts from the System Center Virtual Machine Manager (SCVMM) server by running a remote PowerShell script, and store the script in a local file.
2. Query the ESX hosts from the vCenter server with PowerCLI commands by running a remote PowerShell script, and store the script in a local file.
3. Create a remote PowerShell script that can be run on any OnCommand Shift server that takes the following inputs:
  - a. The Hyper-V host (obtained from step 1) and its credentials
  - b. The ESX host (obtained from step 2) and its credentials

The script must perform the following steps:

- Configuring the hypervisor for the OnCommand Shift server with the information obtained in step 1a and step 1b.
- Fetching the network adapters on the Hyper-V and ESX hosts and configuring the network settings for the OnCommand Shift server.

- Fetching the source hypervisor VMs and convert them to the destination hypervisor, as described in the section [“Queuing VM Conversion Jobs.”](#)

4. To completely automate this process, create an automation script to run on each pair of Hyper-V and ESX servers (obtained in step 1 and step 2). Then run the script created in step 3 on the corresponding OnCommand Shift server.

The following section provides automation scripts for queuing VMs for conversion in a SCVMM and vCenter environment. It is assumed that:

- You have installed and configured multiple OnCommand Shift servers.
- The SCVMM server, the vCenter server, and all of the Hyper-V hosts and ESX hosts reside in the same domain.
- Each OnCommand Shift server is configured with a separate ONTAP volume. Each OnCommand Shift server is connected to exactly one pair of ESX and Hyper-V hosts. Also, the VMs to be converted are moved to their respective volumes.

**Note:** NetApp technical support does not provide assistance with the scripting provided in this document. Support is only available for the execution of individual OnCommand Shift cmdlets.

```
# This script will query the Hyper-V Hosts from a SCVMM Server and ESX Hosts from a
#VCenter Server
# The user selects source and destination hypervisors from the displayed list
# The user also provides the credentials for the selected hypervisors
# The script will query the network adapters from hypervisors and displays to the user
# The user selects the network configuration information for the shift server
# This script will then fetch the VMs from the source hypervisor and saves the
# information in a csv file
# The user selects the VMs to be converted by editing the csv file with either Yes/No
# option
# The script will convert the selected VMs
#-----
# Please Fill in the details for entries marked in red
$scvmmhost = "<input FQDN of SCVMM Host>"  

$scvmmUserName = "<input UserName of SCVMM Host>"  

$scvmmPassword = "<input Password of SCVMM Host>"  

$vcCenterhost = "<input FQDN of vCenter Host>"  

$vcCenterUserName = "<input UserName of vCenter Host>"  

$vcCenterPassword = "<input Password of vCenter Host>"  

$guestOSDomain = "<input Domain of Guest OS>"  

$guestOSUserName = "<input UserName of Guest OS>"  

$guestOSPassword = "<input Password of Guest OS>"  

#-----
# Change the path for csv files to store VM list and Conversion status if required
$VMsListFilePath = "$PSScriptRoot\VMsList.csv"  

$VMsConversionStatusFilePath = "$PSScriptRoot\VMsConversionStatus.csv"  

#-----
# Create arrays to hold VMs and Conversion Status
$hypervhostsarray= @()  

$vmarray= @()  

$vmstatusarray= @()  

# Delete contents of the files
If(Test-Path $VMsListFilePath) {Clear-Content $VMsListFilePath}  

If(Test-Path $VMsConversionStatusFilePath) {Clear-Content $VMsConversionStatusFilePath}  

# Create SCVMM Credential object
$scvmmcredpwd = ConvertTo-SecureString -AsPlainText -Force $scvmmPassword  

$scvmmcred = new-object management.automation.pscredential $scvmmUserName, $scvmmcredpwd  

# Create vCenter Credential Object
$vccentercredpwd = ConvertTo-SecureString -AsPlainText -Force $vccenterpassword  

$vccentercred = new-object management.automation.pscredential $vccenterusername, $vccentercredpwd  

# Run a remote command on SCVMM Host to query the Hyper-V Hosts
$hypervhostlist = Invoke-Command -ComputerName $scvmmhost -Credential $scvmmcred -  

ScriptBlock {Get-SCVMHost -VMHost $using:scvmmhost).ComputerName}
# Add VMware Snapin to PowerShell and Connect to vCenter Server
$addssnapin = Add-PSSnapin VMware.VimAutomation.Core
$connect = Connect-VIServer -Server $vcCenterhost -Credential $vccentercred
# Run a remote command on vCenter Host to query the ESX Hosts
$esxhostlist = (Get-VMHost | Where-Object {$__.PowerState -eq "PoweredOn"}).Name
# Disconnect from the vCenter Server
$disconnect = Disconnect-VIServer -Server $vcCenterhost -Confirm:$false
do
{
    # Display menu to choose the source and destination hypervisor type and receive
    # credentials from the user for the same
```

```

$srchypervisor = Read-Host -Prompt "`nSelect source hypervisor (hyperv/esx)"
if($srchypervisor -eq "hyperv")
{
    $srchostlist = $hypervhostlist
    $desthostlist = $esxhostlist
    $srchypervisortype = "Hyper-V"
    $desthypervisortype = "ESX"
    $desthypervisortypeforpath = "esx"
}
elseif($srchypervisor -eq "esx")
{
    $srchostlist = $esxhostlist
    $desthostlist = $hypervhostlist
    $srchypervisortype = "ESX"
    $desthypervisortype = "Hyper-V"
    $desthypervisortypeforpath = "hyperv"
}
else
{
    Write-Host "Wrong input`n"
    Exit
}
# Source Hypervisor Menu
Write-Host "`nSelect a source $srchypervisortype Host"
Write-Host "Please select a $srchypervisortype Host from the following list to
fetch VMs"
$srchostlist
# Read source hypervisor details from the user
$srchostname = Read-Host -Prompt "`nInput the $srchypervisortype Host name (FQDN)"
$srchostip = Read-Host -Prompt "`nInput the $srchypervisortype IP Address"
$srcusrname = Read-Host -Prompt "Input the $srchypervisortype User name"
$srcpwd = Read-Host -Prompt "Input the $srchypervisortype Password"
if($srchypervisortype -eq "ESX")
{
    $esxdatacenter = Read-Host -Prompt "Input the data center for
$srchypervisortype Host"
}
# Destination Hypervisor Menu
Write-Host "`nSelect a destination $desthypervisortype Host"
Write-Host "Please select a $desthypervisortype Host from the following list as
destination hypervisor"
$desthostlist
# Read destination hypervisor details from the user
$desthostname = Read-Host -Prompt "`nInput the $desthypervisortype Host name
(FQDN)"
$desthostip = Read-Host -Prompt "`nInput the $desthypervisortype IP Address"
$destusrname = Read-Host -Prompt "Input the $desthypervisortype User name"
$destpwd = Read-Host -Prompt "Input the $desthypervisortype Password"
if($desthypervisortype -eq "ESX")
{
    $esxdatacenter = Read-Host -Prompt "Input the data center for
$desthypervisortype Host"
}
# Create source hypervisor Credential Object
$srccredpwd = ConvertTo-SecureString $srcpwd -AsPlainText -Force
$srccred = new-object management.automation.pscredential $srcusrname, $srccredpwd
# Create destination hypervisor Credential Object
$destcredpwd = ConvertTo-SecureString $destpwd -AsPlainText -Force
$destcred = new-object management.automation.pscredential $destusrname,
$destcredpwd
# Select the credentials based on hypervisor type
if($srchypervisor -eq "hyperv")
{
    $hypervhostname = $srchostname
    $hypervip = $srchostip
    $hypervcred = $srccred

    $esxhostname = $desthostname
    $esxip = $desthostip
    $esxcred = $destcred
}
elseif($srchypervisor -eq "esx")
{
    $hypervhostname = $desthostname
    $hypervip = $desthostip
    $hypervcred = $destcred

    $esxhostname = $srchostname
    $esxip = $srchostip
    $esxcred = $srccred
}

```

```

# Run a remote command to get the VMs and Network adapters list from a chosen
Hyper-V host
$hypervvms = Invoke-Command -ComputerName $hypervhostname -Credential $hypervcred
-ScriptBlock {(Get-VM | Where-Object {$_._State -eq "running"}).VMName}
$hypervadapters = Invoke-Command -ComputerName $hypervhostname -ScriptBlock {(Get-
VMswitch).Name}
# Run a remote command to get the VMs and Network adapters list from a chosen ESX
host
$connect = Connect-VIServer -Server $esxhostname -Credential $esxcred
$esxvms = (Get-VM | Where-Object {$_._PowerState -eq "PoweredOn"}).Name
$esxadapters = (Get-VirtualPortGroup).Name
$disconnect = Disconnect-VIServer -Server $esxhostname -Confirm:$false
# Get the shift server details from user
Write-Host "`nEnter details for the Shift server to be used for conversion"
$shiftserver = Read-Host -Prompt "`nInput the Shift server IP Address or FQDN"
$shiftusrname = Read-Host -Prompt "Input the Shift server User name"
$shiftpwd = Read-Host -Prompt "Input the Shift server Password"
# Create Shift Server Credential Object
$shiftcredpwd = ConvertTo-SecureString $shiftpwd -AsPlainText -Force
$shiftcred = new-object management.automation.pscredential
$shiftusrname,$shiftcredpwd
# Choose the VMs to convert based on hypervisor type
if($srchypervisor -eq "hyperv")
{
    $vmstoconvert = $hypervvms
}
elseif($srchypervisor -eq "esx")
{
    $vmstoconvert = $esxvms
}
# Create a PowerShell object for each VM with required information
ForEach ($vm In $vmstoconvert)
{
    $vmarrayitem =
[pscustomobject]@{vmname=$vm;guestOSDomain=$guestOSDomain;guestOSUserName=$guestOSUserName;
guestOSPassword=$guestOSPassword;convert="yes"}
    $vmarray += $vmarrayitem
}
# Export the PowerShell objects created to a CSV File
$vmarray | Export-Csv $VMsListFilePath
# Connect to Shift Server and configure Hypervisor details
Connect-ShiftServer -host $shiftserver -credential $shiftcred
Set-ShiftHypervisorConfig HYPER-V -IPaddress $hypervip -hostname $hypervhostname -
credential $hypervcred -force
Set-ShiftHypervisorConfig VMWARE -IPaddress $esxip -hostname $esxhostname -
credential $esxcred -datacenter $esxdatacenter -force
# Display the Network Adapters available on both the hypervisors
Write-Host "`nNetwork Adapters on selected ESX host"
$esxadapters
Write-Host "`nNetwork Adapters on selected Hyper-V host"
$hypervadapters
# Get the user input to map network configuration between the hypervisors for the
Shift Server
do{
    Write-Host "`nConfigure Network settings for the Shift Server"
    $netconfigname = Read-Host -Prompt "`nInput the Name for the Network
Config"
    $netconfigesx = Read-Host -Prompt "`nSelect the ESX Network Adapter (from
the list of ESX adapters displayed above)"
    $netconfighyperv = Read-Host -Prompt "`nSelect the Hyper-V Network Adapter
(from the list of Hyper-V adapters displayed above)"
    $netconfigvlanid = Read-Host -Prompt "`nEnter the VLAN ID"
    # Add a new network configuration for the Shift Server from the
information provided by the user
    Add-ShiftNetworkConfig -name $netconfigname -vmPortGroup $netconfigesx -
hypervswitch $netconfighyperv -vLanId $netconfigvlanid -force
    $subloopagain = Read-Host -Prompt 'Do you want to add another network
config? (Y/N)'
    }while($subloopagain -eq 'y' -or $subloopagain -eq 'Y')
    # Read the destination path for the converted VMs from the user and configure the
destination path settings for the Shift Server
    Write-Host "`nConfigure the Destination Path for the Shift Server"
    $destinationpath = Read-Host -Prompt "`nInput the Destination path for converted
VMs of $desthypervisortype"
    Set-ShiftDestinationPathConfig -destinationHypervisor $desthypervisortypeforpath -
destinationPath $destinationpath -force
    # Open the VMs List CSV file for editing
    notepad.exe $VMsListFilePath
    # A small menu for confirmation to convert the selected VMs
    $title = "Convert VMs"
    $message = "Do you want to convert the selected VMs?"

```

```

$yes = New-Object System.Management.Automation.Host.ChoiceDescription "&Yes",
"Converts all the VMs selected"
$no = New-Object System.Management.Automation.Host.ChoiceDescription "&No", "Exits
the script"
$options = [System.Management.Automation.Host.ChoiceDescription[]]($yes, $no)
$result = $host.ui.PromptForChoice($title, $message, $options, 0)
If ($result -eq 0)
{
    Write-Host "You selected Yes."
}
else
{
    Write-Host "You selected No.Exiting the script."
    Exit
}
# Import the CSV File and for each VM issue conversion command
$vmList = Import-Csv $VMsListFilePath
ForEach ($line In $vmList)
{
    # Query the details from csv file
    $vmname = $line.vmname
    $vmdomain = $line.guestOSDomain
    $vmusername = $line.guestOSUserName
    $vmpassword = $line.guestOSPassword
    $vmconvert = $line.convert
    # Get the IP Address of the VM for Hyper-V
    if($srchypervisor -eq "hyperv")
    {
        $gosip = Invoke-Command -ComputerName $hypervhostname -Credential
$hypervcred -ScriptBlock {(Get-VMNetworkAdapter -VMName $using:vmname).IPAddresses[0]}
    }
    # Create Guest OS Credential Object
    $goscredpwd = ConvertTo-SecureString -AsPlainText -Force $vmpassword
    $goscred = new-object management.automation.pscredential $vmusername,
$goscredpwd
    # Issue the Conversion command for the selected VM
    If ($vmconvert -eq "Yes" -or $vmconvert -eq "yes")
    {
        if($srchypervisor -eq "hyperv")
        {
            $shiftjobid = Convert-ShiftVirtualMachine -source hyperv -
destination esx -name $vmname -ipAddress $gosip -credential $goscred -domain $vmdomain -
force
        }
        elseif($srchypervisor -eq "esx")
        {
            $shiftjobid = Convert-ShiftVirtualMachine -source esx -
destination hyperv -name $vmname -credential $goscred -domain $vmdomain -force
        }
        Write-Host "$vmname queued for conversion"
        #Create a PowerShell object for each VM with conversion job id
        $vmstatusarrayitem =
[pscustomobject]@{vmname=$vmname;shiftjobid=$shiftjobid}
        $vmstatusarray += $vmstatusarrayitem
    }
}
# Export the PowerShell objects created to a CSV File
$vmstatusarray | Export-Csv $VMsConversionStatusFilePath
# Disconnect from Shift Server
Disconnect-ShiftServer
$loopagain = Read-Host -Prompt 'Do you want to convert using another set of
hypervisor hosts and shift server? (Y/N)'
}while($loopagain -eq 'y' -or $loopagain -eq 'Y')

```

After this script initiates the VM conversion process for the selected VMs, you can use the script provided in the section "[Checking Conversion status](#)" to read the conversion status file and retrieve the status of each of the VMs being converted.

## 4 Conclusion

NetApp OnCommand Shift helps an administrator to rapidly and seamlessly convert VMs from VMware to Hyper-V and back again. It can also convert just the virtual disks between the different hypervisors. Therefore, OnCommand Shift saves you several hours of effort each time that you want to move workloads from one hypervisor to the other. Organizations can now host multihypervisor environments

without having to worry about whether workloads are tied down to a single hypervisor. This capability increases flexibility and reduces licensing costs, lock-in, and commitments to a single vendor.

## Appendix

### SAN Environments

As a key requirements of OnCommand Shift, the VMs to be converted must reside in a NAS environment (NFS for ESX and SMB for Hyper-V). If the VMs reside in a SAN environment (iSCSI, FC, or FCoE), then they must be migrated to a NAS environment before conversion.

### Virtual Machine Conversion from ESX to Hyper-V

Figure 5) Virtual machine conversion from ESX to Hyper-V in SAN environments.

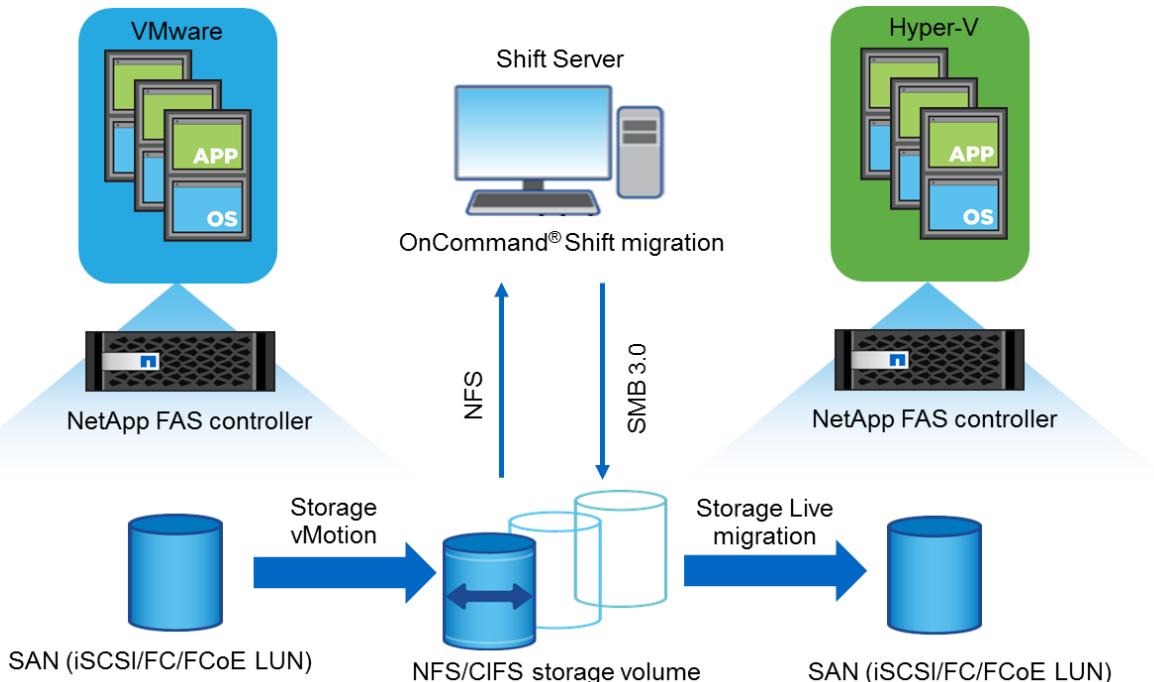


Figure 5 depicts a typical SAN environment in which VMs are stored in a SAN datastore. The VMs to be converted from ESX to Hyper-V along with their disks are first migrated to an NFS data-store with VMware vSphere Storage vMotion. OnCommand Shift uses FlexClone to convert the VMs from ESX to Hyper-V. The converted VMs (along with their disks) reside on a CIFS datastore. The converted VMs (along with their disks) are migrated back to the SAN data-store with Hyper-V Storage Live Migration.

## Virtual Machine Conversion from Hyper-V to ESX

Figure 6) Virtual machine conversion from Hyper-V to ESX in SAN environments.

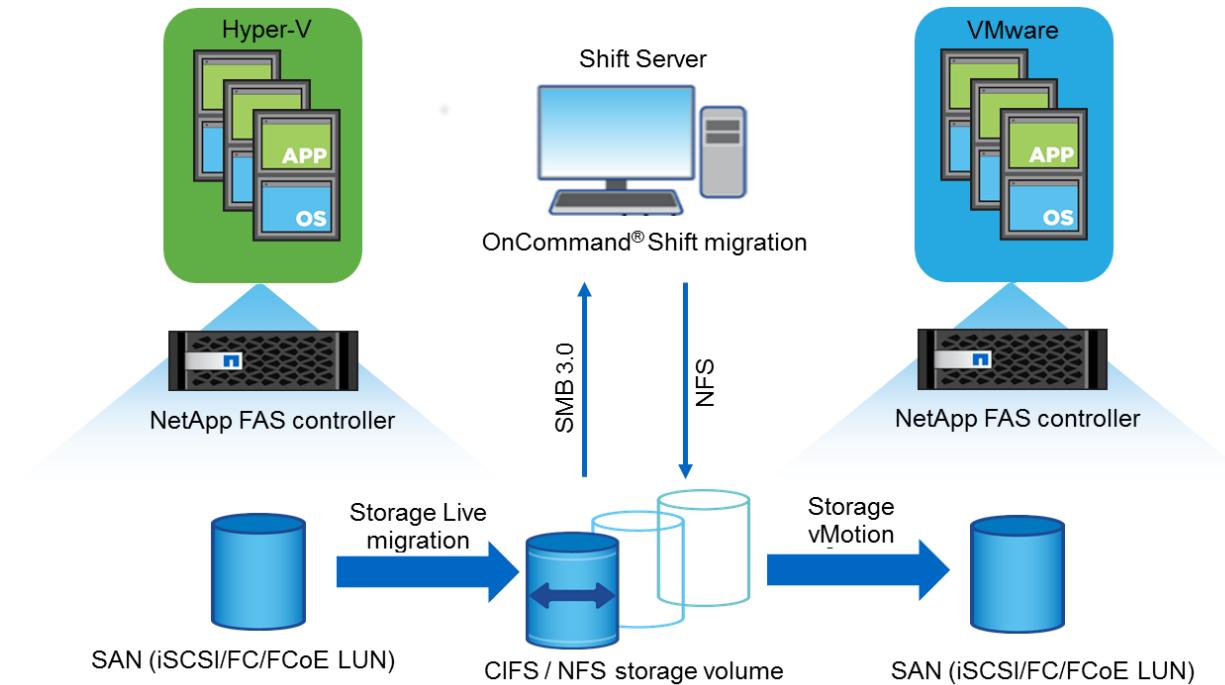


Figure 6 depicts a typical SAN environment in which VMs are stored in a SAN datastore. The VMs to be converted from Hyper-V to ESX along with their disks are first migrated to a CIFS datastore with Hyper-V Storage Live Migration. OnCommand Shift then uses FlexClone to convert the VMs from Hyper-V to ESX. The converted VMs along with their disks reside on an NFS datastore. The converted VMs along with their disks are migrated back to the SAN datastore with VMware vSphere Storage vMotion.

## References

The following references were used in this technical report:

- [NetApp OnCommand Shift 1.2 Release Notes](#)
- [NetApp OnCommand Shift 1.2 Installation and Setup Guide](#)
- [NetApp OnCommand Shift 1.2 Conversion and Administration Guide](#)
- [NetApp OnCommand Shift 1.2 Cmdlet Reference Guide](#)

## Version History

Version	Date	Document Version History
<a href="#">Version 1.0</a>	April 2015	<ul style="list-style-type: none"><li>• Conversion of VMs from ESX to Hyper-V</li><li>• Conversion of VMs from Hyper-V to ESX</li></ul>

Version	Date	Document Version History
<a href="#"><u>Version 1.1</u></a>	September 2015	<ul style="list-style-type: none"> <li>• Configuration Wizard support</li> <li>• Internationalization support for English, Japanese, German, and Simplified Chinese</li> <li>• ONTAP support for adaptive compression volumes</li> <li>• ESXi support for v6.0</li> <li>• Powershell module renamed from ShiftClient to OCShift</li> <li>• New cmdlets for rollback and uninstalling VMware tools</li> </ul>
<a href="#"><u>Version 1.1.1</u></a>	March 2016	<ul style="list-style-type: none"> <li>• Support for distributed switches</li> <li>• Support for different LIFs for CIFS and NFS protocols</li> </ul>
<a href="#"><u>Version 1.2</u></a>	June 2016	<ul style="list-style-type: none"> <li>• Support for the Linux guest OS</li> <li>• UI for conversion and job tracking</li> <li>• Support for two destination paths</li> <li>• Support for aborting conversion jobs</li> <li>• Support for Foundation REST APIs</li> <li>• Support for virtual disk conversion</li> <li>• Support for obtaining prestaging logs</li> <li>• New Shift server configuration management cmdlets</li> </ul>

Refer to the [Interoperability Matrix Tool \(IMT\)](#) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

## Copyright Information

Copyright © 1994–2016 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

## Trademark Information

NetApp, the NetApp logo, Go Further, Faster, AltaVault, ASUP, AutoSupport, Campaign Express, Cloud ONTAP, Clustered Data ONTAP, Customer Fitness, Data ONTAP, DataMotion, Flash Accel, Flash Cache, Flash Pool, FlashRay, FlexArray, FlexCache, FlexClone, FlexPod, FlexScale, FlexShare, FlexVol, FPolicy, GetSuccessful, LockVault, Manage ONTAP, Mars, MetroCluster, MultiStore, NetApp Fitness, NetApp Insight, OnCommand, ONTAP, ONTAPI, RAID DP, RAID-TEC, SANshare, SANtricity, SecureShare, Simplicity, Simulate ONTAP, SnapCenter, SnapCopy, Snap Creator, SnapDrive, SnapIntegrator, SnapLock, SnapManager, SnapMirror, SnapMover, SnapProtect, SnapRestore, Snapshot, SnapValidator, SnapVault, SolidFire, StorageGRID, Tech OnTap, Unbound Cloud, WAFL, and other names are trademarks or registered trademarks of NetApp Inc., in the United States and/or other countries. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such. A current list of NetApp trademarks is available on the web at <http://www.netapp.com/us/legal/netapptmlist.aspx>. TR-4433