



Managing and Protecting a Windows Server Hyper-V Environment using Dell EqualLogic PS Series Storage and Tools

Revisions

The following table describes the release history of this Deployment and Configuration Guide.

| Report | Date | Document Revision |
|--------|---------------|---|
| 1.0 | December 2011 | Initial release as TR1080 |
| 2.0 | May 2014 | Second release incorporating HIT/Microsoft 4.7 functionality. This document supersedes TR1080 |

THIS WHITE PAPER IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

© 2014 Dell Inc. All rights reserved. Reproduction of this material in any manner whatsoever without the express written permission of Dell Inc. is strictly forbidden. For more information, contact Dell.

Dell™, the Dell logo, Dell Boomi™, Dell Precision™, OptiPlex™, Latitude™, PowerEdge™, PowerVault™, PowerConnect™, OpenManage™, EqualLogic™, Compellent™, KACE™, FlexAddress™, Force10™ and Vostro™ are trademarks of Dell Inc. Other Dell trademarks may be used in this document. Cisco Nexus®, Cisco MDS®, Cisco NX-OS®, and other Cisco Catalyst® are registered trademarks of Cisco System Inc. EMC VNX®, and EMC Unisphere® are registered trademarks of EMC Corporation. Intel®, Pentium®, Xeon®, Core® and Celeron® are registered trademarks of Intel Corporation in the U.S. and other countries. AMD® is a registered trademark and AMD Opteron™, AMD Phenom™ and AMD Sempron™ are trademarks of Advanced Micro Devices, Inc. Microsoft®, Windows®, Windows Server®, Internet Explorer®, MS-DOS®, Windows Vista® and Active Directory® are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Red Hat® and Red Hat® Enterprise Linux® are registered trademarks of Red Hat, Inc. in the United States and/or other countries. Novell® and SUSE® are registered trademarks of Novell Inc. in the United States and other countries. Oracle® is a registered trademark of Oracle Corporation and/or its affiliates. Citrix®, Xen®, XenServer® and XenMotion® are either registered trademarks or trademarks of Citrix Systems, Inc. in the United States and/or other countries. VMware®, Virtual SMP®, vMotion®, vCenter® and vSphere® are registered trademarks or trademarks of VMware, Inc. in the United States or other countries. IBM® is a registered trademark of International Business Machines Corporation. Broadcom® and NetXtreme® are registered trademarks of Broadcom Corporation. Qlogic is a registered trademark of QLogic Corporation.



Table of contents

- Revisions..... 2
- Executive summary 6
- Audience..... 6
- Software information 7
- Referenced documentation..... 7
- 1 EqualLogic Host Integration Tools Microsoft Edition (HIT/Microsoft)..... 8
 - 1.1 HIT/Microsoft core components 8
 - 1.2 Installing HIT/Microsoft on a dedicated Hyper-V Management Virtual Machine 10
- 2 Dell EqualLogic Auto-Snapshot Manager Microsoft Edition (ASM/ME)..... 12
 - 2.1 Requirements and constraints..... 12
 - 2.2 ASM/ME interaction with Microsoft Volume Shadow Copy Services (VSS)..... 13
 - 2.3 Understanding ASM/ME Smart Copies..... 16
 - 2.3.1 Smart Copy types..... 17
 - 2.3.2 Backup Document Folder Location 18
 - 2.3.3 Torn Smart Copies 19
 - 2.3.4 Smart Copy data recovery options 19
- 3 Basic ASM/ME Smart Copy operations for a Hyper-V environment 20
 - 3.1 Accessing the PS Group for ASM/ME on the Management VM 20
 - 3.2 Add the Hyper-V host into to the HIT Group managed from ASM/ME 25
 - 3.3 Create Smart Copies of Hyper-V VMs..... 30
 - 3.3.1 Using ASM/ME to create a Smart Copy snapshot of a Hyper-V VM..... 31
 - 3.3.2 Using HIT/Microsoft PowerShell tools to create a Smart Copy snapshot..... 34
 - 3.3.3 Using ASM/ME to create Smart Copy Clone..... 37
 - 3.3.4 Using HIT/Microsoft PowerShell Tools to create a Smart Copy Clone 38
 - 3.4 Creating a Smart Copy schedule for virtual machines..... 39
 - 3.5 Working with a Collection of Virtual Machines 43
 - 3.5.1 Use ASM/ME to create a collection of virtual machines..... 43
 - 3.5.2 Use HIT/Microsoft PowerShell Tools to create a collection of virtual machines 45
 - 3.5.3 Use ASM/ME to create a Smart Copy of a collection of virtual machines..... 45
 - 3.5.4 Use HIT/Microsoft PowerShell Tools to create a Smart Copy of a collection of virtual machines 49
 - 3.6 Working with Clustered Shared Volumes (CSVs) 49



| | | |
|-------|--|-----|
| 3.6.1 | Use ASM/ME to create a Smart Copy of a CSV | 50 |
| 3.6.2 | Use HIT/Microsoft PowerShell tools to create a Smart Copy of a CSV | 52 |
| 3.7 | Hyper-V VM Smart Copy restore operations | 53 |
| 3.7.1 | Use ASM/ME to perform an In-Place Restore of Hyper-V VMs | 53 |
| 3.7.2 | Use HIT/Microsoft PowerShell Tools to perform an In-Place Restore of Hyper-V VMs | 55 |
| 3.7.3 | Using ASM/ME to perform a Selective Restore operation for a specific Hyper-V VM | 56 |
| 3.7.4 | Use HIT/Microsoft PowerShell Tools to perform a Selective Restore of a Hyper-V VM | 59 |
| 3.7.5 | Using ASM/ME to perform a Restore as New operation | 60 |
| 4 | Create a VM template from an ASM/ME Smart Copy | 61 |
| 4.1 | Using ASM/ME, take a Smart Copy of the VM for the template | 63 |
| 4.2 | Use ASM/ME to perform a Restore as New operation to create a new template VM | 63 |
| 4.3 | Verify that the new template VM has been created | 66 |
| 4.4 | Modify the new template VM settings as necessary | 69 |
| 4.5 | Startup the template VM using Hyper-V Manager or PowerShell | 70 |
| 4.6 | Sysprep and reboot the template virtual machine | 72 |
| 4.7 | Convert the template VM volume to an EqualLogic "Template Volume" | 75 |
| 4.8 | Updating Hyper-V environment documentation | 77 |
| 5 | Summary | 80 |
| A | Sample Hyper-V VM Selective Restore script using HIT/Microsoft PowerShell tools | 81 |
| B | Sample rapid VM deployment script | 85 |
| B.1 | Create virtual machine XML configuration file | 88 |
| B.2 | Import required modules and verify EqualLogic services | 89 |
| B.3 | Locate virtual machine XML file and read in contents | 89 |
| B.4 | Create variables to hold XML configuration information and VM and VHD path information | 90 |
| B.5 | Gather primary user input | 91 |
| B.6 | Function: SelectMethod | 93 |
| B.6.1 | } Function: ProcessCreate | 94 |
| B.6.2 | Function: ProcessCopy | 96 |
| B.7 | Function: CreateVM | 97 |
| B.8 | Function: SetupNetworking | 99 |
| B.9 | Main Logic | 100 |
| C | Provisioning the Hyper-V environment | 101 |



| | | |
|-----|--|-----|
| C.1 | Using the EqualLogic SMP integration..... | 101 |
| C.2 | Using HIT/Microsoft PowerShell tools..... | 113 |
| D | Perform a Hyper-V storage migration for a VM to a new EqualLogic iSCSI storage device..... | 119 |
| D.1 | Using Hyper-V Manager | 120 |
| D.2 | Using PowerShell | 129 |



Executive summary

With increasing pressure to reduce business costs and increase service availability, virtualization is very prevalent in today's datacenters. Microsoft Windows Server Hyper-V virtualization is commonly used by businesses to minimize total cost of ownership (TCO), streamline management of both physical and virtual machines, and increase availability of business applications. This paper focuses on using Dell EqualLogic PS Series storage and tools to help manage and protect a Microsoft Hyper-V virtualized environment. Readers will gain an understanding of the following:

- The Dell EqualLogic Host Integration Tools Microsoft Edition (HIT/Microsoft) components and capabilities
- Protecting the Hyper-V environment using Auto-Snapshot Manager/Microsoft Edition ASM/ME and PowerShell scripting
- Creating Hyper-V virtual machine templates using ASM/ME
- Rapid Virtual Machine Deployment using HIT/Microsoft and PowerShell
- Provisioning EqualLogic Storage using HIT/Microsoft integration with Microsoft Server Manager and PowerShell scripting
- Guidelines for constructing standard operating procedures that manage and protect a Hyper-V environment

Audience

This technical solution guide provides instructions for an Information Technology administrator to manage and protect a Windows Server Hyper-V environment using PowerShell and Dell EqualLogic PS Series storage and tools. This paper assumes that the reader has:

- Previous Windows Server administration experience including installation and configuration
- Understanding of virtualization technologies
- Some familiarity with Microsoft PowerShell or some other scripting language

The examples in this guide include extensive amounts of PowerShell commands. It is strongly recommended that the reader have some basic understanding of PowerShell prior to attempting the examples in the document. A detailed discussion on PowerShell and its capabilities goes beyond the scope of this paper, but it is vitally important that Windows and Hyper-V administrators get some understanding and familiarity with this powerful tool. For more information on PowerShell, consult the following resources:

- [Dell Technical Report: *Windows Command-line Automation Techniques for Dell EqualLogic PS Series Arrays*](http://en.community.dell.com/dell-groups/dtcmedia/m/mediagallery/20421645.aspx) at <http://en.community.dell.com/dell-groups/dtcmedia/m/mediagallery/20421645.aspx>
- Ed Wilson, *Windows PowerShell 3.0 Step by Step*, O'Reilly Media, Inc., Sebastopol, CA, 2013.
- Don Jones and Jeffery Hicks, *Learn Windows PowerShell 3 in a Month of Lunches*, Manning Publications Co., Shelter Island, NY, 2013.



Although Microsoft's System Center is a key component of any Hyper-V virtualization strategy, especially in larger environments, providing a detailed discussion on System Center is also beyond the scope of this paper. This paper focuses on small to medium Hyper-V deployments where System Center functionality is not required. For more information on Microsoft System Center and its integration with Dell EqualLogic Storage, consult the Dell technical report: [Automation and Integration with Microsoft System Center Virtual Machine Manager 2012 SP1 and Dell EqualLogic Storage](http://en.community.dell.com/techcenter/extras/m/white_papers/20437936.aspx) at http://en.community.dell.com/techcenter/extras/m/white_papers/20437936.aspx.

Software information

The following table shows the software and firmware used in the configurations presented in this paper.

| Vendor | Model | Software Revision |
|-----------|--|-------------------|
| Dell | Host Integration Tools for Microsoft (HIT/Microsoft) | vv4.7 |
| Dell | Dell EqualLogic Firmware | v7.0+ |
| Microsoft | Windows Server 2012 | |
| Microsoft | Windows Server 2012 R2 | |

Referenced documentation

The following table lists the documents referred to in this Technical Report. All Dell EqualLogic Technical Reports are available on the Customer Support site at eqlsupport.dell.com

| Vendor | Document Title |
|--------|---|
| Dell | Dell EqualLogic Host Integration Tools for Microsoft, Installation and User's Guide |
| Dell | Dell EqualLogic Auto-Snapshot Manager/Microsoft Edition User's Guide |
| Dell | Dell EqualLogic PowerShell Tools User Guide |
| Dell | Dell EqualLogic Host Integration Tools for Microsoft Release Notes |
| Dell | Using Dell EqualLogic Storage with Microsoft Failover Clusters and Hyper-V |
| Dell | PS Series Deployment: Configuring and Deploying Dell EqualLogic Multipath I/O Device Specific Module with Microsoft Windows |



1 EqualLogic Host Integration Tools Microsoft Edition (HIT/Microsoft)

The Dell EqualLogic Host Integration Tools Microsoft Edition (HIT/Microsoft) provides an administrator the ability to manage and configure PS Series storage arrays from the servers that use them. The HIT/Microsoft toolkit allows administrators to perform a wide variety of tasks such as initializing new arrays, creating application consistent snapshots, and scripting management operations. The EqualLogic Host Tools are available to all EqualLogic customers at no additional cost.

When used with Microsoft Windows Server Hyper-V, HIT/Microsoft can ensure robust protection and quick recovery of these virtualized environments. HIT/Microsoft integrates with Microsoft Volume Shadow Copy Services (VSS). The VSS architecture enables backup applications to create consistent point-in-time copies of virtual machines that can be used for recovery operations in the event of failures. Dell EqualLogic enables SAN protection of Hyper-V virtual machines by integrating with Microsoft Volume Shadow Copy Services and Virtual Disk Services for application-consistent data protection and quick recovery. This utilizes the built-in hardware volume snapshot capabilities of Dell EqualLogic PS Series arrays.

When HIT/Microsoft is used in conjunction with Windows Server Hyper-V Manager, and PowerShell, the Hyper-v administrator has a toolkit that enables building customizable solutions to provide functionality typically reserved for System Center Virtual Machine Manager (SCVMM). SCVMM is a large encompassing product designed to manage hundreds of virtualized servers and virtual machines. For small medium businesses where only a few servers are being virtualized, SCVMM might not be an economically viable option. However, many traditional SCVMM tasks such as rapid virtual machine deployment from a template and virtual machine backup and restore can be easily performed with the HIT/Microsoft, native Windows Server Hyper-V manager, and some simple PowerShell scripting.

1.1 HIT/Microsoft core components

Table 1 Core components installed with the HIT kit

| Installed Component | Installed Features and Services | Description |
|------------------------|---|---|
| Remote Setup Wizard | Tool | Enables an administrator to initialize a PS Series array and set up or expand a PS Series group. |
| Dell EqualLogic ASM/ME | Service: EqlASMAgent Application: EQLASMAApp.exe | Enables centralized management of Windows hosts connecting to EqualLogic SANs. Offers automated protection and recovery utilizing Smart Copies (snapshots, clones, and replicas). An alternate CLI facilitates custom operations and scripting. |
| PowerShell Tools | Libraries: EQLPSTools.dll, EQLASMTTools.dll*, EQLPSArrayTools.dll*, EQLMpioPSTools.dll* | Enables management of one or more PS Series groups through a comprehensive set of PowerShell cmdlets. *Introduced with HIT/Microsoft version 4.7 |



| Installed Component | Installed Features and Services | Description |
|---|--|---|
| Dell EqualLogic Storage Management Provider (SMP) | Service: EqlSMPHost | Enables an administrator to manage Dell EqualLogic storage directly through native Windows storage interfaces such as PowerShell cmdlets, File Services UI in the Windows Server 2012 Server Manager console, or the standard Windows Management Instrumentation API. |
| Dell EqualLogic VSS Provider | Services: EqlVss and EqlReqService | Supports VSS integration for application-consistent Smart Copies of Hyper-V VMs, SQL Server, SharePoint, and Exchange databases and NTFS |
| Dell EqualLogic VDS Provider | Service: EqlVdsHwPrv In pre-Windows 2012 Deployments | Enables an administrator to use Microsoft VDS and Microsoft Storage Manager for SANs to create and manage volumes in a PS Series group. Requires installation of the Windows feature iSCSITarget-VSS-VDS. Otherwise, EqlVdsHwPrv service is installed with a default startup state of manual. |
| Dell EqualLogic Multipath I/O DSM (MPIO DSM) | Feature: Multipath I/O Service: EHCMservice Note: multiple iSCSI host bus adapters are required to use this feature. | This feature is the driver component in the I/O stack that works in conjunction with the Microsoft MPIO driver to route I/O to the desired path. It dynamically balances the iSCSI SAN traffic load over multiple network paths between the computer and the PS Series group. When the HIT kit is installed, it automatically installs the Windows MPIO feature if it is not already installed on the Hyper-V host. It also sets the MPIO load balancing policy to least queue depth. This feature also installs the EqualLogic Host Connection Manager (EHCM) service that manages iSCSI connections and sessions from the host to the EqualLogic PS Series array. |
| Microsoft iSCSI Initiator | Service: MSiSCSI Note: Installed by Windows Server but not started. | Enables connections to iSCSI targets. Required by Remote Setup Wizard, ASM/ME, and the VDS provider. The HIT installer automatically configures this service, sets the Windows firewall rules, and configures the service for automatic startup. |
| EQL iSCSI Initiator properties tab | Tool | Provides information about the status and history of the multipath connections supported by the Multipath I/O DSM in the iSCSI tool within Server Manager. Note: Requires Minimum Server Interface on Server Core or full GUI versions of Windows Server |

One of the major benefits of HIT/Microsoft is that when it is installed, it configures all of the defaults and preferred EqualLogic settings for the MSiSCSI service and Multipath I/O (MPIO) feature. For the MSiSCSI service, the install starts the service, sets the appropriate firewall rules, and then sets the startup to **auto**. For MPIO, it verifies that the Windows Multipath I/O feature is installed on the host, and then sets the



load-balancing algorithm to the EqualLogic optimized setting of **least queue depth**. For a complete list of settings, refer to *EqualLogic PS Series Arrays: Optimizing your SAN Environment for High Availability* at <http://en.community.dell.com/dell-groups/dtcmedia/m/mediagallery/20371245/download.aspx>

The other major benefit of installing HIT/Microsoft is the set of Dell EqualLogic PowerShell cmdlets that are installed. The Dell EqualLogic PowerShell cmdlets can create powerful commands to manage PS Series groups and objects in the PowerShell environment.

1.2 Installing HIT/Microsoft on a dedicated Hyper-V Management Virtual Machine

HIT/Microsoft can be installed on the Management OS of the Hyper-V host server(s) running either Windows Server Core or Full Windows Server with GUI; and on any Hyper-V virtual machines running in the virtualized environment. Often in Hyper-V environments, a dedicated management virtual machine is set up to perform all Hyper-V management functions. This is done in accordance with Microsoft best practice recommendations for Hyper-V when the Hyper-V host server is dedicated to running the Hyper-V role. All Hyper-V environment management functions are offloaded to the dedicated management virtual machine. This paper focuses on how to perform Hyper-V management operations such as protecting and provisioning the Hyper-V environment with Windows Server tools and the Dell EqualLogic HIT kit from a dedicated management virtual machine. A single Hyper-V server layout that uses a dedicated management virtual machine may use the configuration shown in Figure 1.

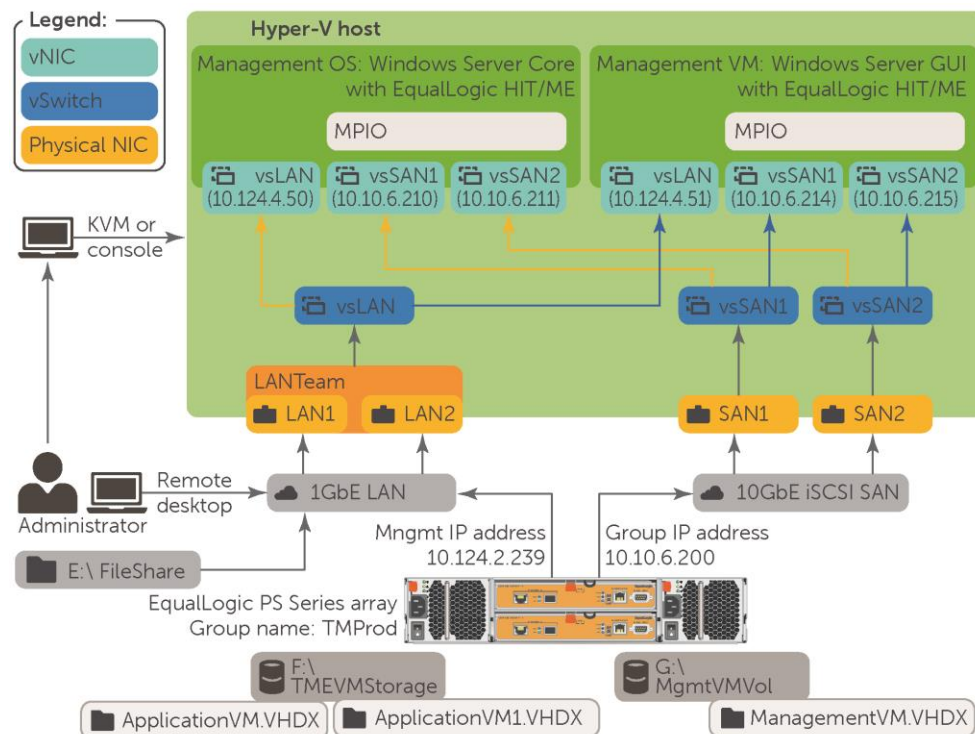


Figure 1 A Hyper-V server using a dedicated management virtual machine

This diagram shows a management OS and separate management VMs. The management VM uses a dedicated EqualLogic volume to house its VHDX and configuration files. The EqualLogic HIT kit has been installed on all of the virtual machines as well as the Management OS of the Hyper-V server.

Note: The operational steps used on a management virtual machine are identical if they were run from the Management OS of the Hyper-V host server.

This paper discusses how to set up and use HIT/Microsoft to protect the Hyper-V environment. It does not go into detail about the architecture and installation. This information is covered in detail in the *Dell EqualLogic Auto-Snapshot Manager/Microsoft Edition User Guide* in the document directory for the EqualLogic HIT kit installation (c:\Program Files\EqualLogic\Docs).

For more information on installing and using the EqualLogic HIT kit, go to the HIT/Microsoft installation directory at C:\Program Files\EqualLogic\Doc\ for the following files.

- *Dell EqualLogic Host Integration Tools for Microsoft, Installation and User's Guide*
- *Dell EqualLogic Auto-Snapshot Manager/Microsoft Edition User's Guide*
- *Dell EqualLogic PowerShell Tools User Guide*

For more information on Dell EqualLogic Multipath IO DSM, go to: [TR1036 V4.0 – PS Series Deployment: Configuring and Deploying Dell EqualLogic Multipath I/O Device Specific Module with Microsoft Windows](#)



2 Dell EqualLogic Auto-Snapshot Manager Microsoft Edition (ASM/ME)

Installing HIT/Microsoft provides administrators with ASM/ME, a powerful tool that can be used to help protect the Hyper-V environment. When ASM/ME is installed on a system running Hyper-V, point-in-time Smart Copies of virtual machines can be created and used to recover entire VMs.

For Hyper-V, the native ASM/ME can offer the following functionality:

- Define collections of virtual machines or cluster volumes, including Cluster Shared Volumes
- Set up schedules for creating Smart Copies of virtual machines, clustered volumes and cluster shared volumes, or collections of virtual machines and clustered shared volumes
- Restore smart copies of virtual machines
- Restore virtual hard drive volumes
- Centrally manage Dell EqualLogic storage configuration settings across all managed hosts and virtual machines

2.1 Requirements and constraints

There are certain requirements and constraints when using ASM/ME to protect the Hyper-V environment.

| Requirement / Constraint | Description |
|-----------------------------------|---|
| Windows Operating System | Any Hyper-V computer or virtual machine that host ASM/ME must be a 64-bit server running Windows Server 2008 or later. |
| PS Series Group firmware | Groups must run under the firmware revision specified in the Host Integration Tools for Microsoft Release Notes. |
| Sufficient snapshot reserve space | Volumes must have snapshot space reserved to create snapshots. Refer to the Dell EqualLogic Group Manager Administrator's Guide |
| Restoring a virtual machine | A restore operation will take the virtual machine offline and then try to restart it after the restore operation is complete. If the VM was in an offline state at the time of the restore, no attempt will be made to restart it. |
| Integration Services | Integration Services (Virtual Guest Services) components must be installed on every virtual machine to support online backup and data exchange. See the Microsoft Hyper-V Documentation on TechNet for installation information about Integration Services. |
| Hyper-V VSS Writer constraints | ASM/ME is constrained and limited by the current functions offered by the Hyper-V VSS writer. |



| Requirement / Constraint | Description |
|-----------------------------|--|
| Direct-attach iSCSI volumes | Using directly attached iSCSI volumes to a virtual machine is supported as long as the specific virtual machine has ASM/ME installed. This is required because directly attached iSCSI volumes to a VM are not visible to the Hyper-V host and therefore not visible to the ASM/ME components installed on the Hyper-V host or management virtual machine. |
| Pass-through volumes | Pass-through volumes are not supported for ASM/ME operations. Pass-through volumes are logged into by the host management OS but mounted only on the VM as an IDE device. In this case, the volume is not seen as an iSCSI target and is not visible as such to ASM/ME or to Hyper-V. |

2.2 ASM/ME interaction with Microsoft Volume Shadow Copy Services (VSS)

To protect Hyper-V virtual machines, ASM/ME works with Microsoft VSS. VSS operates on the block level of the file system and backs up disk information only. As a schema, it has three components: a requester, a provider and a writer. Figure 2 shows the relationship between each component and ASM/ME.



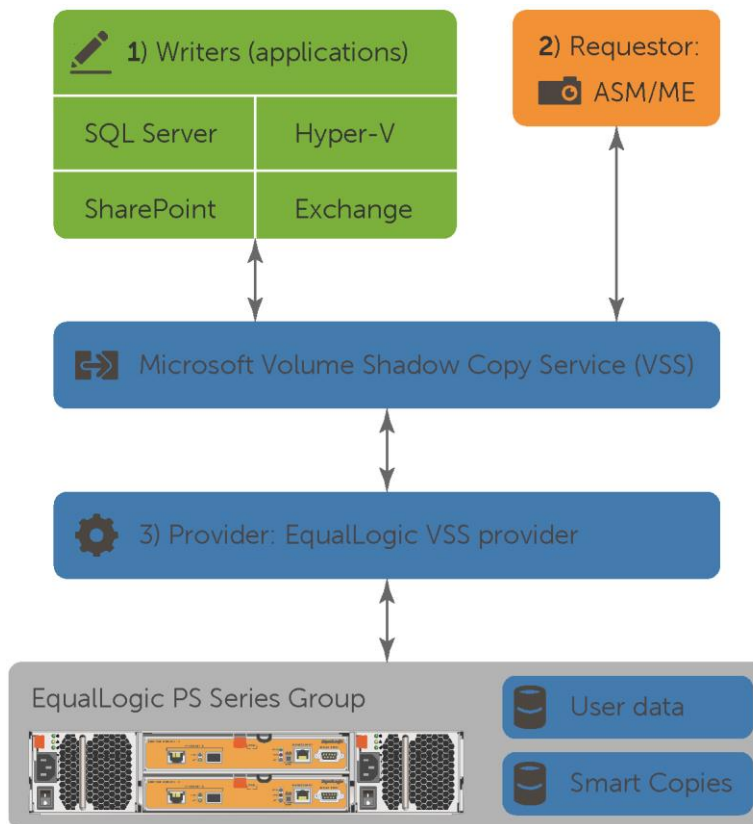


Figure 2 Relationship to the Microsoft VSS Copy Service

1. **VSS Writer**, integrated with the application, prepares the application for the backup or restore operation.
2. **VSS Requestor**, usually a backup application, requests the creation of shadow copies and provides an interface for backing up and restoring data. ASM/ME functions as a requestor.
3. **VSS Provider**, installed with the Host Integration Tools, interacts directly with the PS Series group.

Figure 3 illustrates an example of components and the process used to perform a backup of a Hyper-V virtual machine by an ASM/ME instance that is installed on a dedicated management VM.

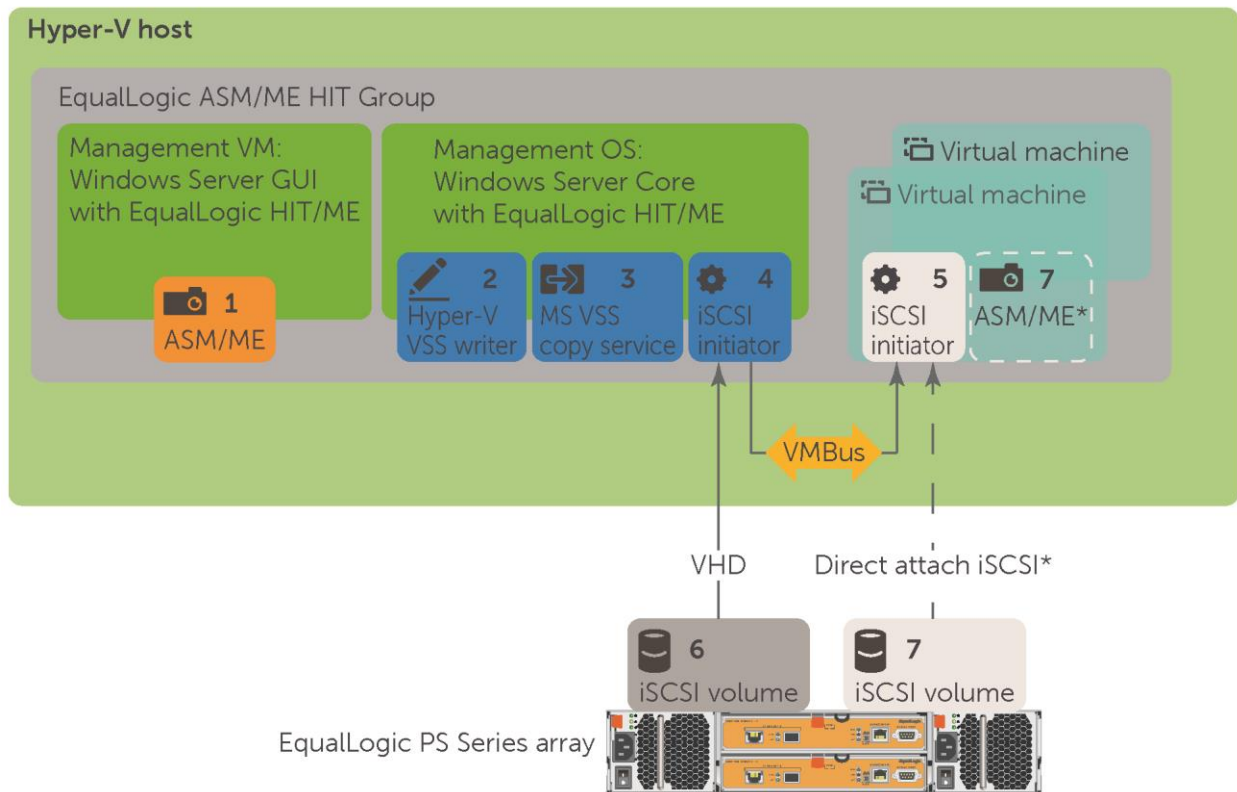


Figure 3 Hyper-V VM back up by an ASM/ME

1. ASM/ME (the requestor) on the management VM generates a request to take a smart copy of a virtual machine on the Hyper-V host. This is done using the Dell EqualLogic VSS Provider installed with the HIT kit.
2. This request is sent to the Hyper-V VSS writer, which is part of the Virtual Machine Management Service (VMMS), on the Hyper-V Server Management OS. The writer places the virtual machine into the Saved State as it starts the PrepareForSnapshot event.
3. The Microsoft VSS Copy Service performs a snapshot of the virtual machine. Once the snapshot is done, it returns the virtual machine back to its original state prior to the PrepareForSnapshot event.
4. During the snapshot, the virtual machine guest connects to the PS Series group through the iSCSI initiator installed under the Hyper-V host Management OS.
5. The virtual machine has the Backup Integration Services (Virtual Guest Services) component installed to facilitate copies of the virtual machine and the data on it while the VM is online.
6. The boot volume (C : \) for each virtual machine is a virtual hard drive (VHD) on an EqualLogic iSCSI volume located in an EQL PS Series Group. The EqualLogic VSS provider instructs the group to create clones or snapshots for the entire EQL volume that houses the VHD file.
7. The PS Series Group can also be used to directly present iSCSI volumes to the virtual machine (direct-attach). These volumes could be used for applications such as Exchange, SharePoint SQL Server, and File Servers. Direct-attach iSCSI volumes to a virtual machine are not visible to the Hyper-V Host Management OS and cannot be backed up by the Hyper-V VSS Writer and Copy Services. In order to back up these volumes to make application consistent Smart Copies, ASM/ME must be installed on

each virtual machine that uses direct attached iSCSI volumes. The backup method employed in this case is called the Guest VM Snapshot. In this case, the installed ASM/ME instance on the virtual machine uses its own VSS service to complete the backup for these direct attach volumes. For easier management, it is best to perform snapshots of these direct attached iSCSI volume on the virtual machine from the Management VM. To set this up, simply add the virtual machine into the HIT group which includes the Hyper-V Server and the management VM.

*ASM/ME installation on Hyper-V virtual machines is recommended by Dell

Note: There are some additional VM requirements and constraints for the Guest VM Snapshot method and direct-attach iSCSI volumes:

- Backup (volume snapshot) Integration Service is installed and running in the VM. The service name is **Hyper-V Volume Shadow Copy Requestor**.
- The VM must be in the running state.
- The Snapshot File Location for the VM is set to the same volume in the host operating system as the VHD files for the VM.
- All volumes in the VM are basic disks and there are no dynamic disks.
- All disks in the VM must use a file system that supports snapshots (for example, NTFS).

2.3 Understanding ASM/ME Smart Copies

The primary benefit of installing EqualLogic HIT is that the Hyper-V environment can now be protected using ASM/ME Smart Copies. A Smart Copy consists of:

- One or more snapshots, replicas, or clones, depending on the type of Smart Copy operation and the original object.
- A backup document, describing the Smart Copy.

The resulting Smart Copy can also be called a Smart Copy Set. Even if the Smart Copy operation involves only a single volume, the result is still considered a Smart Copy Set. After a Smart Copy is created, it can be imported and mounted. The object can be restored to the original or a new location using the Smart Copy. Smart Copies can be created instantly, or on a schedule.

An administrator can use ASM/ME to create Smart Copies of:

- **Volumes:** PS Series volumes formatted using the NTFS file system. These iSCSI objects are represented by nodes in the ASM/ME Tree Panel under the Volumes list node.
- **Application Components:** Parts of SQL Server databases, Microsoft Exchange Mailbox databases, Microsoft SharePoint farms, or Hyper-V virtual machines that are represented by nodes in the ASM/ME Tree Panel under the Applications list node. Because ASM/ME supports Hyper-V, you can also create Smart Copies of virtual machines (VMs), VMs residing on CSVs, and CSVs themselves.
- **Collections:** Related groups of volumes or application components that are represented by nodes in the ASM/ME Tree Panel under the Collections list node. For example, multiple volumes or components may be grouped together as a single collection, and then a Smart Copy of the collection created. This



feature is useful when an administrator wants the Smart Copies to be created simultaneously in one set.

2.3.1 Smart Copy types

An administrator can use either the ASM/ME GUI or the Group Manager GUI to view the snapshots, replicas, and clones created by a Smart Copy operation.

- **Snapshots** are point-in-time copies of a PS Series volume. Restoring a snapshot will restore a volume to the state represented by the snapshot.
- **Clones** are a new and independent volume containing the same data as the original volume at the time the clone is created. Since creating a clone creates a new and independent volume, an administrator cannot restore a clone.
- **Replicas** are also point-in-time copies of a PS Series volume. However, the original volume and the replica are located on different PS Series groups, perhaps in different datacenters, for disaster recovery. The Dell EqualLogic PS Series groups and volumes must be configured for replication to utilize a replica Smart Copy type.

Note: Thin Clone is also a Smart Copy type, but this option only pertains to template volumes.

While the Dell EqualLogic Group Manager GUI allows an administrator to create snapshots, clones, and replicas of volumes, ASM/ME provides the same capability, but additionally allows an administrator to create snapshots, clones, and replica Smart Copies of Microsoft Hyper-V virtual machines, Microsoft Exchange storage groups or mailbox databases, Microsoft SharePoint Farms and Microsoft SQL Server databases. ASM/ME can also create Smart Copies of groups or collections of those objects.

When a Smart Copy is created, it consists of one or more snapshots, replicas, or clones, depending on the operation and the original object – as well as a backup document describing the Smart Copy.

2.3.1.1 Choosing either a Full or Copy backup type when creating a Smart Copy

In ASM/ME, the user has two Smart Copy backup options, Full and Copy. The Full Smart Copy requests the VSS writer to do a VSS Full Backup. This will back up all of the files that comprise a virtual machine (VHD and configuration files) and then record the function in the backup history of each file where the backup occurred. However, if there is an application running on the Hyper-V virtual machine, a VSS Full Backup will clear the application log files after the backup completes. Clearing the application logs could create complications for third party application backup software that relies on the logs to take incremental backups. This results in the backup application losing the ability to do incremental backups and forcing a full back up every time. As a rule of thumb, chose to perform a Full back up in situations where:

- Any applications running on the virtual machine are using separate independent volumes (such as direct attach iSCSI) and have no application log data with the virtual machine C:\root partition (within the actual VM VHD).
- No third party application backup software is being used in the environment



2.3.1.2 Choosing a Copy backup type

A Copy backup requests that the VSS writer to do a VSS Copy Backup, which also performs a backup of all virtual machine files. However, any application log files on the Hyper-V VM will not be cleared and the backup history will not be updated. This means that information about changes to the files and logs will be preserved and not affect the sequence of incremental backups being used by third party application backup software. Therefore, it is best to choose Copy backups in situations where:

- Applications running on the virtual machine have files and log data within the virtual machine C:\root partition (within the actual VM's VHD)
- Third party application backup software is being used in the environment

Note: In a standalone Windows Server 2012 (or higher) environment, both full and copy backup types are available. In a Windows Server 2012 (or higher) CSV environment, only the full backup type is available.

2.3.2 Backup Document Folder Location

In ASM/ME, a network folder or shared location can be specified for storing backup documents. If an administrator intends to import Smart Copies on different hosts, using a network share accessible to all the computers that may need to import the Smart Copies is the best option. Please note that a non-clustered host should have its own backup document directory for storing backup documents. If an administrator wants to use a network folder for storing backup documents of non-clustered hosts, each non-clustered host should have its own subdirectory in the network folder. In contrast, for Hyper-V Microsoft Failover Clusters, all cluster nodes must share the same exact backup document directory path. It is a best practice to ensure that backup documents are backed up regularly. For more information about modifying the default backup document folder location, please consult the *Dell EqualLogic Auto-Snapshot Manager/Microsoft Edition User's Guide*.



2.3.3 Torn Smart Copies

ASM/ME always takes a Smart Copy of a Dell EqualLogic PS Series volume. Therefore, if multiple application components (such as VMs) are located on a common volume, all VMs are included in the backup operation and will also be restored during a restore operation. If VMs are spanned across multiple volumes, a Smart Copy operation could potentially result in a VM that is torn. Therefore, Dell recommends that all a VM data set resides on a single volume or data store.

2.3.4 Smart Copy data recovery options

The main methods of recovering data from Smart Copies are:

- In-Place Restore
- Selective Component Restore
- Manual Restore

An **In-Place Restore** is a full recovery option that restores all data in a Dell EqualLogic PS Series volume. An administrator can use this option for snapshot Smart Copies of NTFS data volumes or for specific application components, such as Hyper-V virtual machines. For Hyper-V, this operation restores all data on all PS Series volumes used by all VMs included in the Smart Copy. VMs restored in-place will experience a service interruption.

With a **Selective Component Restore** for Hyper-V, an administrator can specify one or more VMs to be restored. It is suggested to reduce the volume size of VHD/VHDX files from the default of 127 GB to a more precise size based on actual requirements of application data and user space. This will improve the time required to complete a selective restore operation.

A **Manual Restore** can be used in ASM/ME to mount a Smart Copy to a Microsoft Windows drive letter or mount point. An administrator can then manually copy data from the mounted Smart Copy to the target destination.

The ASM/ME Hyper-V restore operation has the following sequence.

1. The Hyper-V management console displays a message confirming that it is performing the restore operation.
2. The VM is taken offline and is deleted by the VSS writer. If a remote desktop to the guest VM is running, the remote desktop session is terminated.
3. ASM/ME restores the VM files.
4. When the restore operation completes, the VM is registered with the Hyper-V service, which adds it to the list of VMs in the Hyper-V management console. If the VM was running before the restore operation was initiated, ASM/ME will restart the VM.
5. If the VM crashed and then an administrator initiated a restore operation to recover it, then the VM is left in the powered-off state by the ASM/ME restore operation.

Note: A restore on a Hyper-V VM places the VM offline, regardless of the type of guest OS that is installed on the VM.



3 Basic ASM/ME Smart Copy operations for a Hyper-V environment

3.1 Accessing the PS Group for ASM/ME on the Management VM

Prior to using ASM/ME on the Management VM, an administrator must configure access of one or more PS Series groups for ASM/ME to connect to for creating Smart Copies and other functions. This section describes performing this task using the example configuration in Figure 1. For full information on PS Series group access settings for ASM/ME, consult the online ASM/ME help (F1 key) and the *Dell EqualLogic Auto-Snapshot Manager/Microsoft Edition User Guide* found in the HIT install directory at C:\Program Files\EqualLogic\Doc.

Group access settings can be specified for the following purposes.

- ASM/ME access to one or more PS Series groups may be allowed for creating and managing Smart Copies, among other functions.
- Microsoft VDS and VSS CHAP settings enable these services to access to a PS Series group.
- Smart Copy access can enable the local computer to access Smart Copies created on other computers.
- PowerShell/SMP access can authenticate PowerShell and SMP access to the group.
- Single Sign-On (SSO) to the group can be enabled if needed.

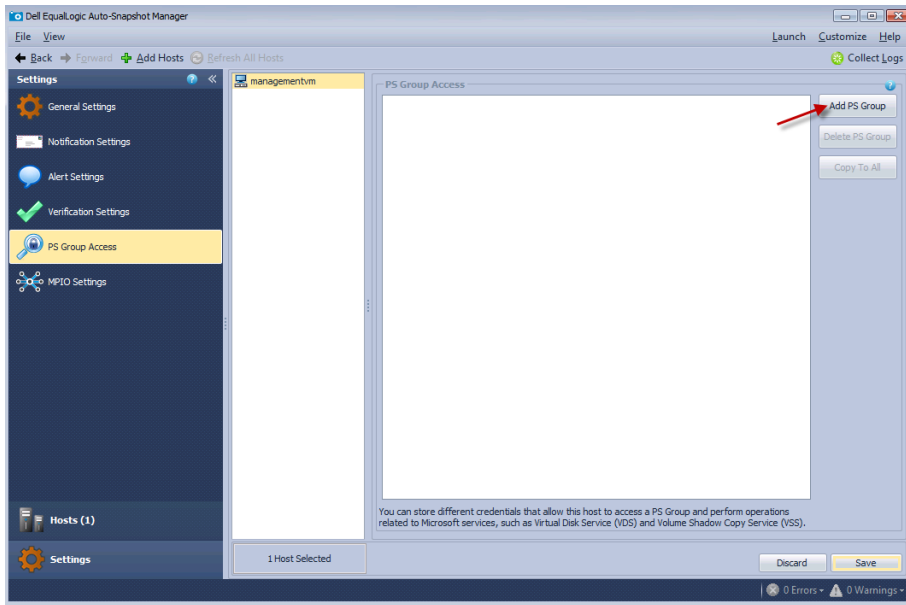
Note: To use Single Sign-On, the PS Series group must be running PS Series firmware version 6.0 (or later), and must be configured to allow SSO. See the Dell EqualLogic Group Manager Online Help for information on configuring the group for Single Sign-On. In addition, the ASM service must be configured to use accounts with domain access

Before setting these properties, CHAP must be configured on the PS Series group, either locally or on an external RADIUS authentication server. For information on configuring CHAP on the PS Series group, see the *Dell EqualLogic Group Manager Administrator's Guide*.

The following steps explain setting up ASM/ME Group access with the previous example.

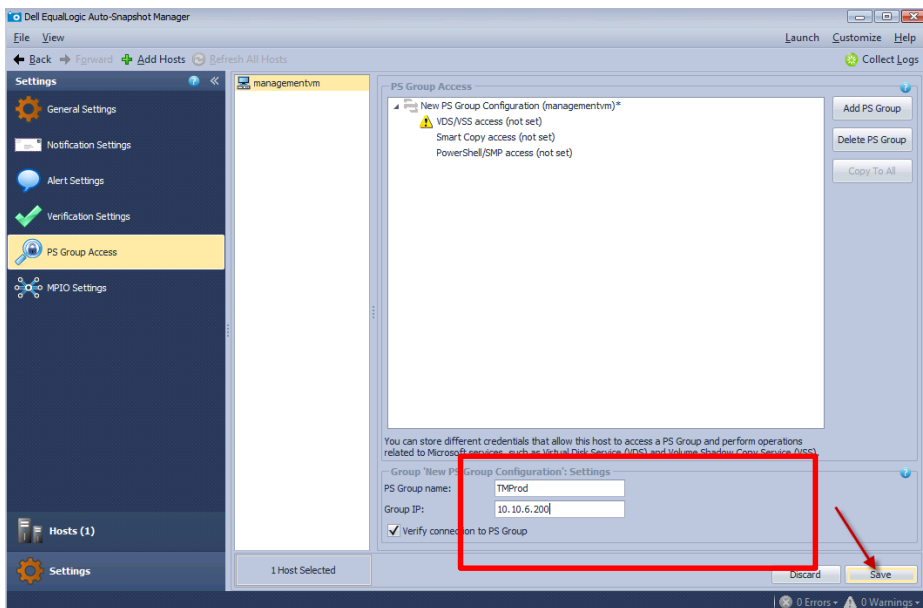
1. **Register the PS Group:** Open ASM/ME and select **Settings > PS Group Access** to open the Add PS Group Access panel. Once the panel opens, click **Add PS Group**.





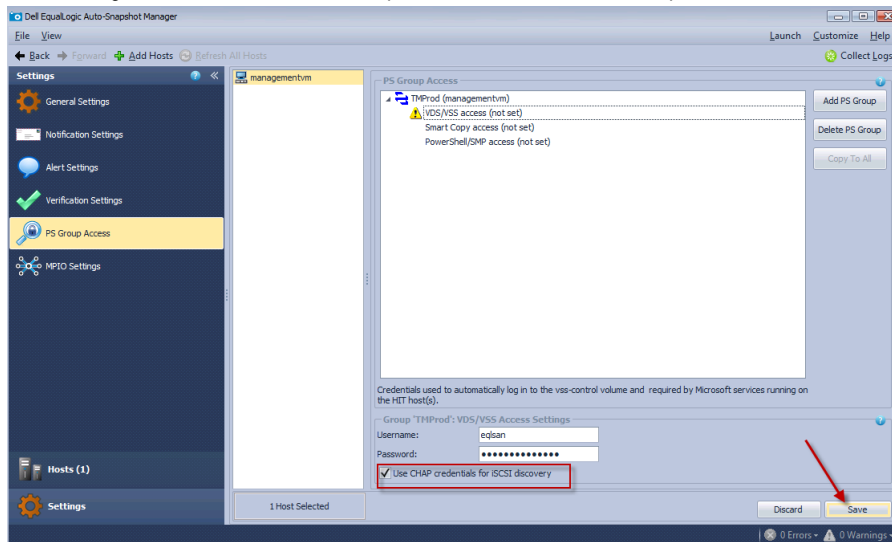
2. In the Group Settings panel, enter the group name and IP address in the fields. In the example, the PS Group name is TMProd and the Group IP is 10.10.6.200. When the information is entered, click **Save**.

Note: Press F1 to view online help for any of these dialogs.



Once the PS Group is successfully registered, the name of the PS Group appears next to the EqualLogic icon in the panel.

3. **Setup VDS/VSS Access.** On the PS Group Access panel in ASM/ME, the VSS/VDS settings enable the configuration of the Microsoft VSS and VDS services. ASM/ME depends on this configuration to function properly. Microsoft VDS and VSS services running on an ASM/ME host must be able to automatically access a PS Series group VSS Control Volume. Typically, VSS volume access is setup using one-way (unidirectional) CHAP authentication. This means that the initiator (host) has to be authenticated by the target (the PS Series storage array) for access to the VSS Volume. To setup VDS/VSS access using one-way CHAP authentication, select **Use CHAP credentials for iSCSI discovery**, enter in the PS Group CHAP username and password, and then click **Save**.



Note: If VDS/VSS access is not configured, the VDS/VSS access node displays a warning icon (yellow triangle). When configured, the icon changes to a green check mark.

A successful setup of the VDS/VSS access will result in the PS Group VSS control volume being discovered and connected as an iSCSI target to the Management VM. This can be verified in the GUI using the iSCSI initiator tool or with the **get-iscsitarget** PowerShell command.

```
[MANAGEMENTVM] PS C:\>get-iscsitarget | ft -autosize
```

```
IsConnected NodeAddress
```

```
-----
```

```
True      iqn.2001-05.com.equallogic:0-1cb196-b481da94f-d8600000042522f7-vss-control
```

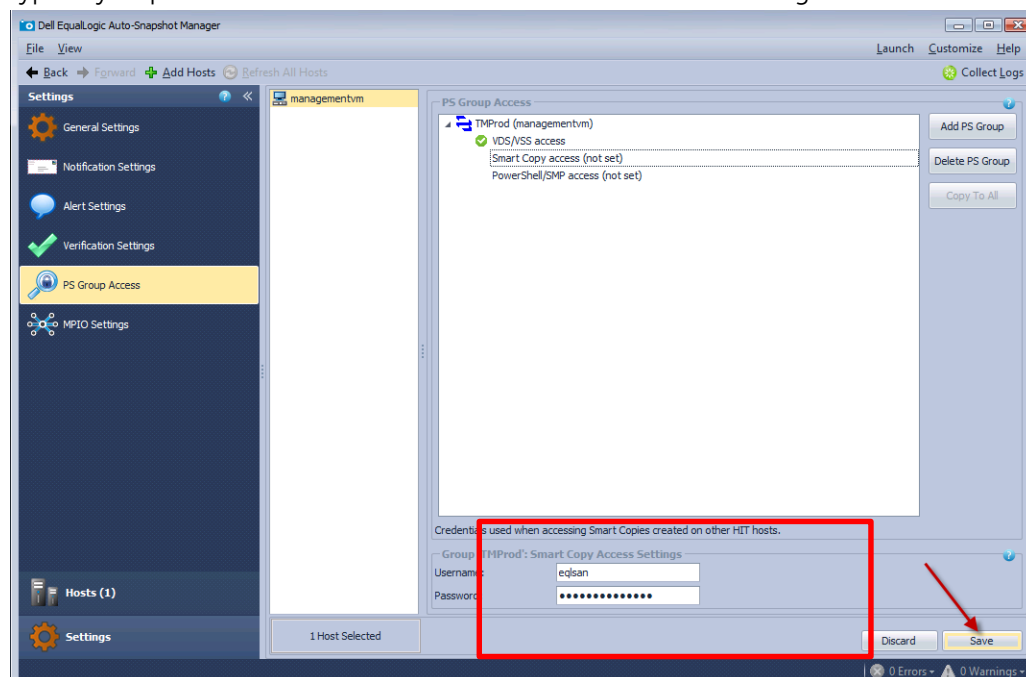
To examine the VSS Control volume's authentication type, run the following PowerShell command:

```
[MANAGEMENTVM] PS C:\>(get-iscsitarget | ? {$_.nodeaddress -like  
"*vss*"} | get-iscsisession).authenticationtype
```

```
ONEWAYCHAP
```

4. **Configure Smart Copy Access.** On the PS Group Access panel, the Smart Copy Access settings enable host configuration for accessing Smart Copies that are created on a different host. When a local host imports a Smart Copy created on a different host, the local host must automatically

present the appropriate access credentials required for accessing the imported Smart Copy. This typically requires the local CHAP user credentials that were configured with the PS Series array.

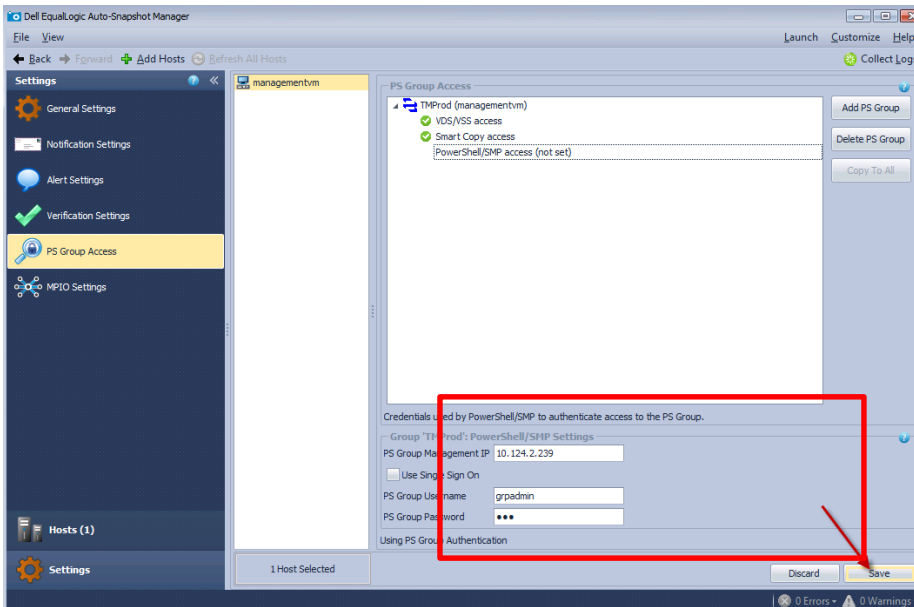


Enter the relevant credentials and click **Save**. Select F1 or click the question mark next to a frame title for online help.

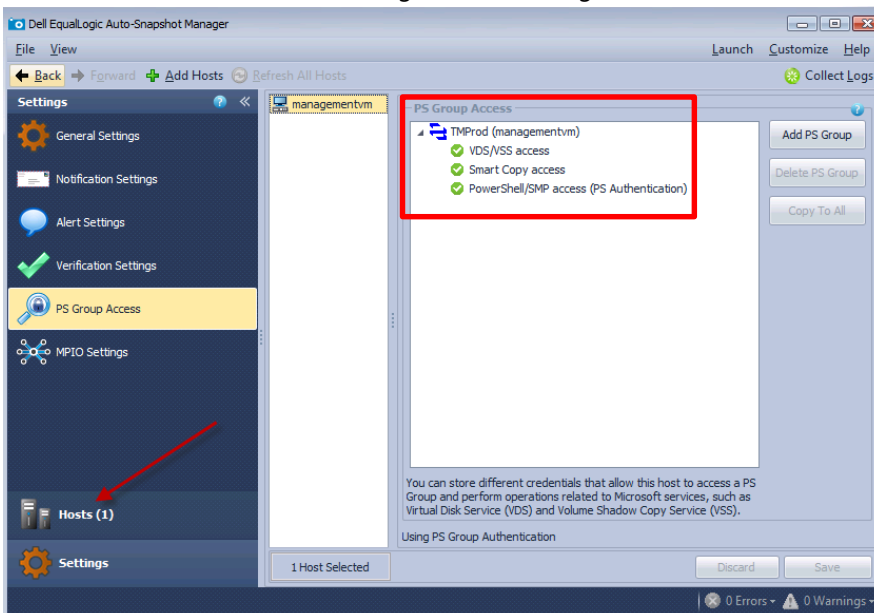
Note: For information on setting up a CHAP user on the PS Array, see the online Group Manager help by opening **Group Manager > Group Configuration > iSCSI tab > local CHAP Account** then selecting F1.

5. **Configure PowerShell and SMP Access.** On the PS Group Access panel, the PowerShell/SMP Access settings enable configuration for access to a PS Series group to:
 - Use the PowerShell tools with a particular PS Series group. The PowerShell cmdlets use different credentials from those that ASM/ME uses. You must configure PowerShell/SMP access if you want to use the PowerShell tools.
 - Use ASM/ME with a host that is running Windows Server 2012/2012 R2 or Windows 8 to perform EqualLogic PS Series array management tasks from native Windows Server Manager or SCVMM.
 - Optional: Enable Single Sign-On (SSO) to a group.

To set the PowerShell and SMP access, enter the PS Group Management IP address and the Group Manager user account and password. If the PS Group is not using a designated Management IP address, use the Group IP Address. The example in this paper uses the grpadmin account. Optionally, check to enable single sign on. Consult the Dell EqualLogic Group Manager online help for information on configuring the group for Single Sign-On. Click **save** when the necessary information is entered. Green check marks next to each access type designate that ASM/ME access on the management VM is successfully configured to the PS Group.



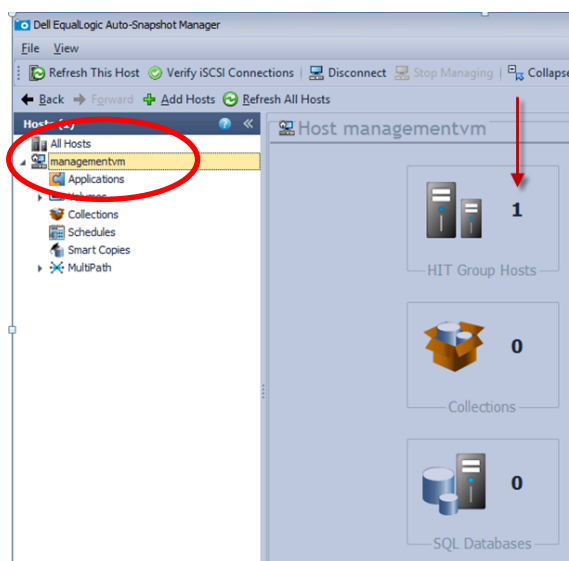
6. Once access to the PS Group is set for the management VM, select **hosts** in the lower-left of the navigation panel. This re-opens the main ASM/ME dashboard. In the HIT Group Host section of the dashboard, there should be a single host (management VM) in the HIT Group.



This is also shown in the tree panel with the managementvm being the only host displayed. The



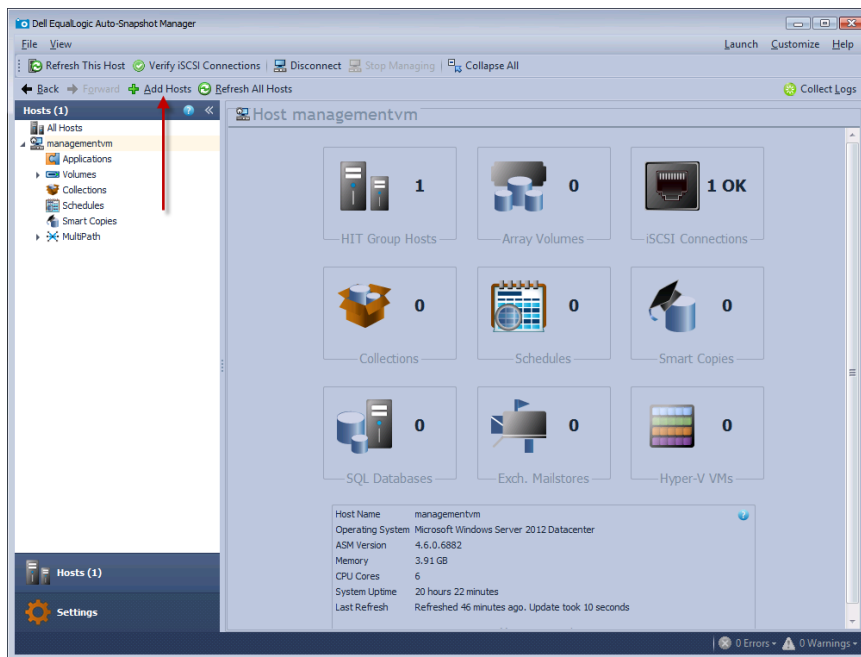
icon in the tree shows that the Management VM node is the local host node where the current instance of ASM/ME is running. Section 3.2 explains how to add the Hyper-V host into the HIT Group so that it can be managed by the ASM/ME instance on the Management VM.



3.2 Add the Hyper-V host into to the HIT Group managed from ASM/ME

A HIT Group is a group of one or more hosts (and/or virtual machines) that is protected from ASM/ME. In a Hyper-V environment that uses a dedicated virtual machine, the Hyper-V host(s) and the management VM can have the HIT kit installed and be placed into a single HIT Group. The entire Hyper-V environment can then be protected by ASM/ME on the management VM as the administrator has the ability to create and manage Smart Copies and Smart Copy schedules for the Hyper-V host itself and all of the associated VMs.

1. To add a Hyper-V host server into the HIT group, select **Add Host** in the ASM/ME main dashboard of the Management VM to open the **Add Host** wizard.



- The first screen in the **Add Host** wizard prompts for one of four discovery methods for the new host so that it can be added into the HIT Group.



- **Discover through PS Series group:** Causes ASM/ME to query any PS Series group connected to any current HIT Group members (including the local host), and displays all the hosts connected to those groups.
- **Browse network:** Select a host by browsing a network. Network permissions are required for this option.

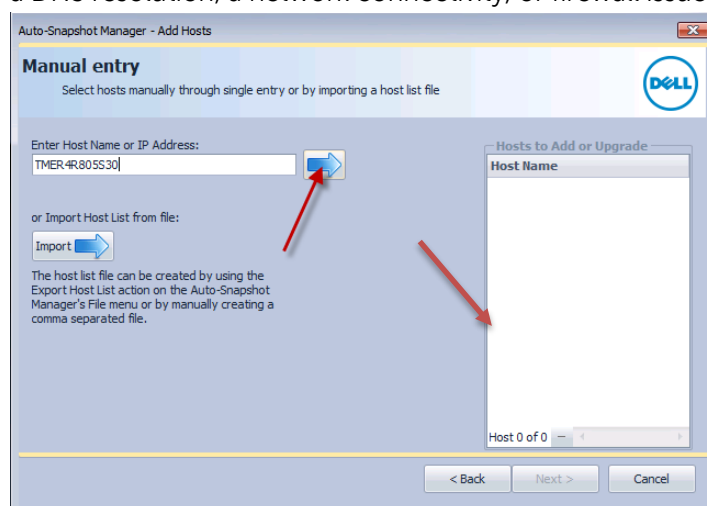


- **Manual entry:** Enter an IP address or host name for each host wanted in the HIT Group. A file could also be imported that lists all the hosts that are to be added to the HIT Group. This is very useful when multiple hosts or virtual machines need to be added to the HIT group. The file will be parsed and each host will be run through the manual entry process. This file can be generated automatically from the ASM/ME console by clicking **File** and then **Export Host List**. The import file could also be a text file that contains a comma-separated list of host IP addresses.
- **Cluster and SharePoint Farm Nodes:** Available only if ASM/ME detects a running multiple node cluster where one or more of the nodes have not been added to the HIT Group. All cluster nodes should be added to the HIT Group. If only a subset of nodes is added, CSV operations will fail.

Note: The Remote Procedure Call service must be running on a host being added to the HIT Group. This service is started by default on Windows systems, but if it is not running, or if the firewall is blocking it, the installation will not work. For full details on how to use the Add Host wizard, consult the ASM/ME online help by typing F1 from anywhere in the ASM/ME tool.

In the example, "Manual Entry" is checked and "Next" is selected

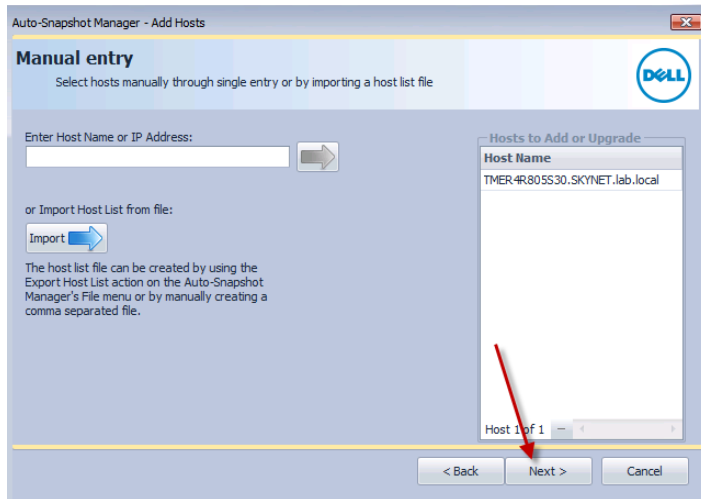
3. Enter the name or IP address of the Hyper-V host in the text box. When the right arrow is selected, the ASM/ME will attempt to ping the server using the entered information as a connection string. If the connection string can be successfully pinged, it checks to see if the HIT tools are installed. If the host connection string cannot be contacted via ping, an error occurs. If this happens, it is most likely a DNS resolution, a network connectivity, or firewall issue.



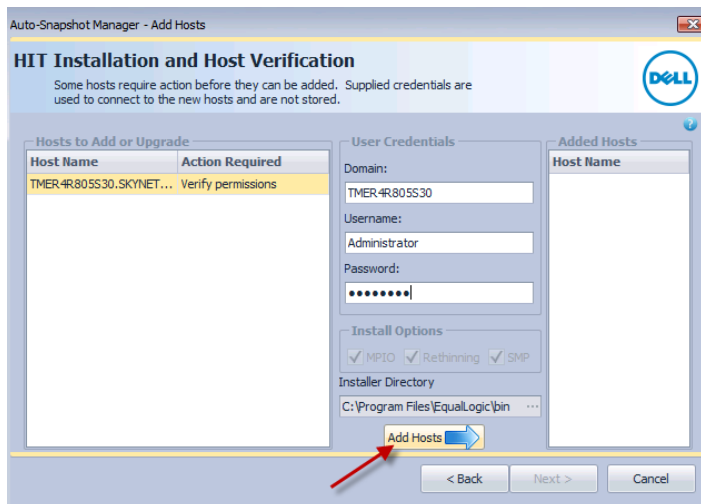
If the host can be successfully contacted, it is added to the **Host to Add or Upgrade** panel.

4. To add another host, enter the name or IP address and repeat the previous steps. Otherwise, click **Next** to display the HIT Installation and Host Verification page.



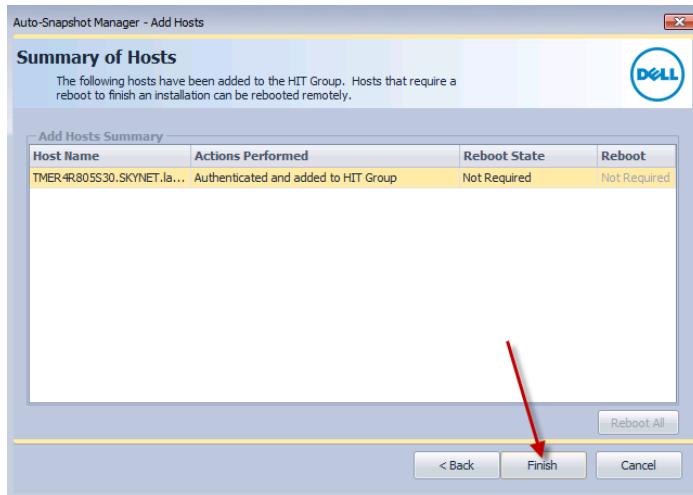



This page confirms that the HIT Tools are installed on the hosts, or prompts for installations and upgrades. It also asks for the host permissions that the ASM/ME instance uses to communicate. In the example, the domain name of the Hyper-V host is entered, as well as the host administrator username and password. In the background, the wizard will attempt to contact the host using username and password supplied. If it is successful, the **Add Hosts** → button becomes active. If the HIT kit is not installed on the host, the user is prompted for the installation directory of the Setup64.exe file to perform a remote installation.

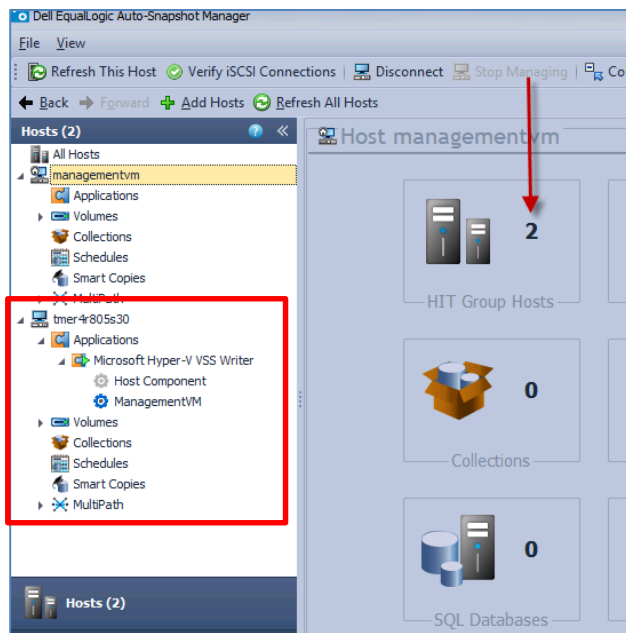


The Summary of Hosts window appears once the Hyper-V host's permissions are verified. If necessary, the HIT kit installed and the host successfully added into the HIT group is displayed.

5. Select **Finish** to close the wizard.



Once the Hyper-V host has been successfully added into the HIT Group, the number of hosts in the **HIT Group Hosts** main dashboard view increases to two. The Hyper-V host is displayed in the navigation tree as a remotely managed HIT Group host. This is designated by the  icon next to the Hyper-V host name. Expanding the tree under the Hyper-V host shows the ASM/ME supported applications. Hyper-V is displayed here as the **Microsoft Hyper-V VSS Writer** with all of the VMs associated with the Hyper-V host listed; in this case, the Management VM itself.

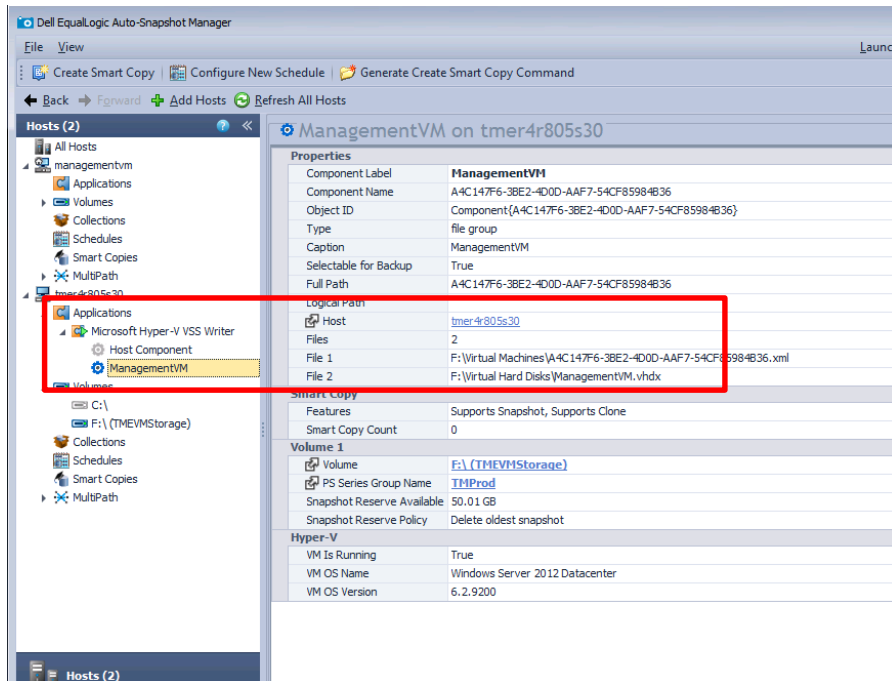


6. Select any virtual machine in the navigation tree to display detailed information about its configuration in the properties window. In the example, the Management VM is selected. In the properties window, information about the location of its virtual hard disk and configuration file is displayed as well as an indicator if it is ready for Smart Copy protection.

Note: In cluster environments, all cluster nodes in a HIT Group have a reciprocal relationship. That is, adding cluster node B to the ASM/ME instance on cluster node A will automatically add cluster node A to the ASM/ME instance on cluster node B.

An administrator must always add an entire cluster to a HIT Group as opposed to a subset of cluster nodes. ASM/ME will then automatically set up the trust relationship between each cluster node. If only one subset of cluster nodes is added to a HIT Group, then data restoration, schedule, and Smart Copy operations could result in fatal errors.

For more information about using HIT Groups in cluster environments, please consult the *Dell EqualLogic Auto-Snapshot Manager/Microsoft Edition User's Guide*.



At this point, ASM/ME is configured and ready to be the primary tool that an administrator can use to protect the Hyper-V environment associated with the Hyper-V host server.

3.3 Create Smart Copies of Hyper-V VMs

Hyper-V virtual machines appear under the Applications node in the Tree Panel. In the VM properties, **VM is running** is displayed. When using Hyper-V virtual machines, ASM/ME allows the following operations with respect to creating a smart copy.

- Create application-consistent and crash-consistent Smart Copies of virtual machines (dependent on the OS type). Note the following constraints for Hyper-V environments:
 - The copy operation (application-consistent or crash-consistent) is dependent on the type of OS and whether the Integration Services (or Virtual Guest Services) are installed on the guest.
 - On Windows Server 2008 R2, replicas are supported for Hyper-V only when it is running on a CSV.

- Volumes for highly available VMs created through SCVMM do not have user-visible mount points. However, Smart Copies of them can be created in the same way as other volumes. For more information, see the *Host Integration Tools for Microsoft Installation and User's Guide*.
- Define collections of virtual machines.
- Set up schedules for creating Smart Copies of virtual machines.

When using ASM/ME with Hyper-V, ASM/ME must be installed within the Virtual Machine Guest Operating Systems to support protection of application components, such as Microsoft SQL and Exchange. Otherwise, an administrator will only be able to manage Smart Copies of the entire VMs and not the application components themselves. For more information on using ASM/ME with Microsoft SQL and Microsoft Exchange, refer to the *Dell EqualLogic Auto-Snapshot Manager Microsoft Edition User's Guide*.

Dell recommends a multi-tiered protection approach in this scenario. For example, from the Management VM or from the Management OS on the Hyper-V server, ASM/ME performs Smart Copies to protect the virtual machines a few times a day. Additionally, ASM/ME running on the Guest Application VMs will perform Smart Copies to protect data and applications, such as SQL and Exchange, at multiple times throughout the day, depending on the data protection requirements. This provides multiple restore points to fail back to in case of a failure or disaster. This approach is a good way to ensure that both the virtual machines, as well as the application data, are optimally protected and is shown in Figure 4.

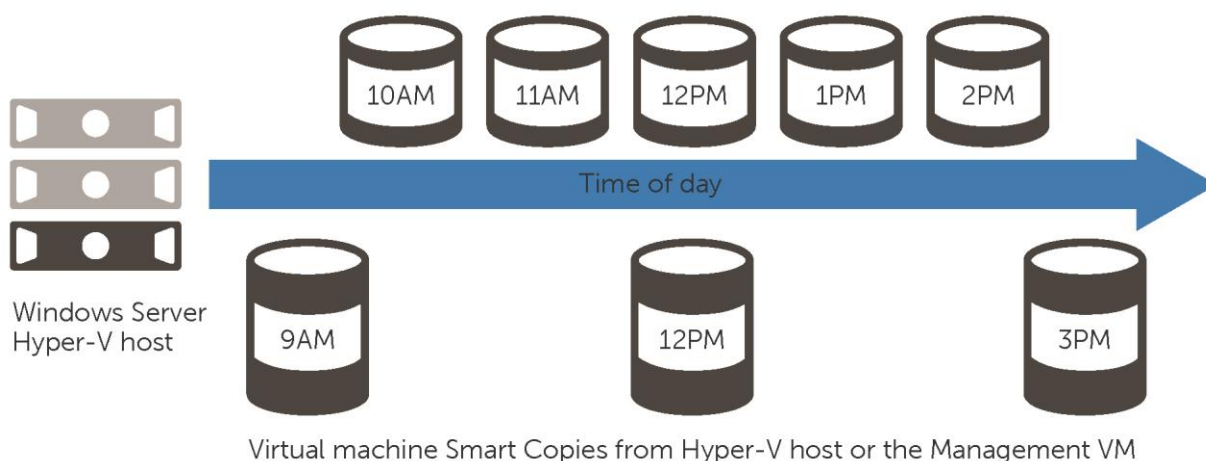
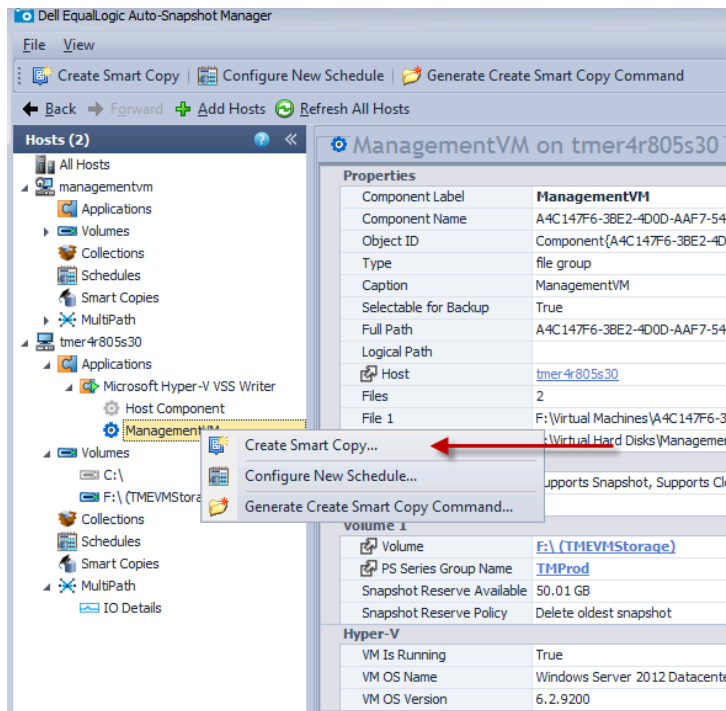


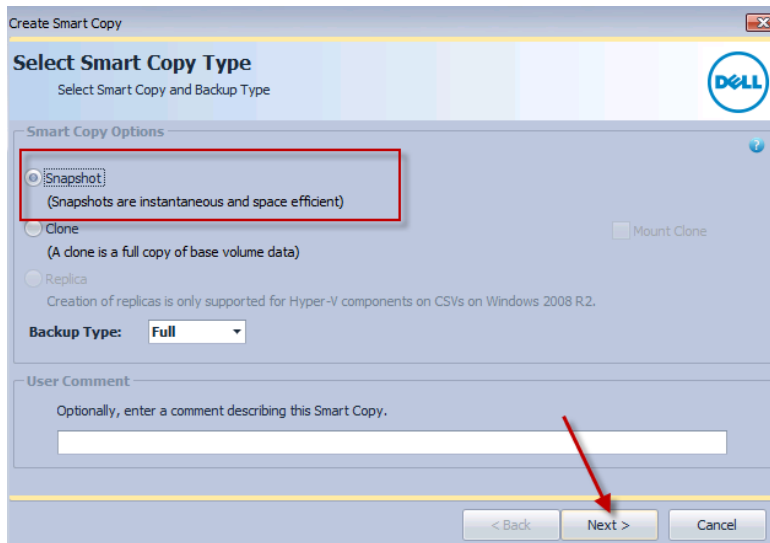
Figure 4 Multi-tiered protection

3.3.1 Using ASM/ME to create a Smart Copy snapshot of a Hyper-V VM

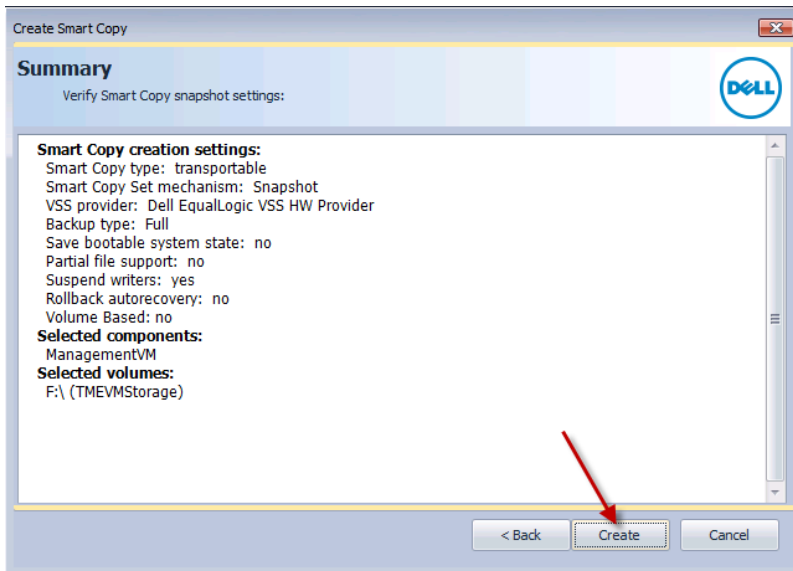
1. To create a Smart Copy of a virtual machine, right click on the virtual machine (the Management VM in the example) and select **Create Smart Copy** from the menu.



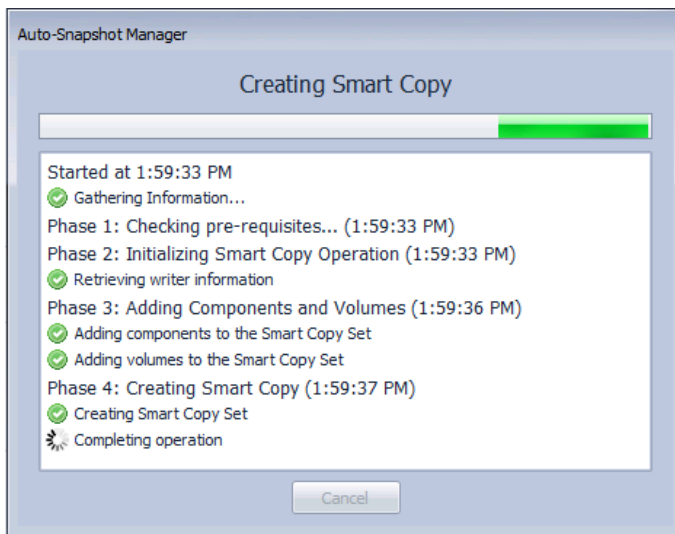
2. The **Select Smart Copy Type** window appears. In the example, a Smart Copy Snapshot is selected. Click **Next** to continue.




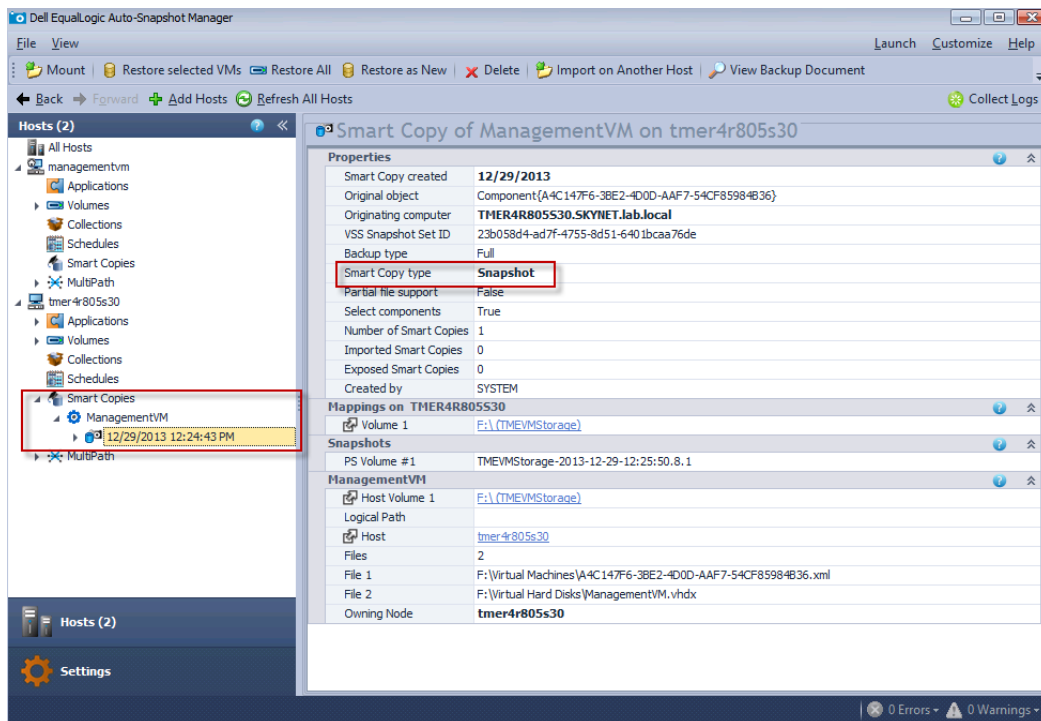
3. In the summary window, select **Create** to create the Smart Copy Snapshot.



As the Smart Copy Snapshot is being created, a progress window is displayed showing the status of each operation step.



4. The new management VM Smart Copy Snapshot appears in the ASM/ME navigation tree under the Smart Copies node. Expanding the virtual machine reveals the snapshot, designated by the  icon. The Snapshot utilizes space from the volume snap shot reserve for the EqualLogic volume that houses the virtual machine and virtual hard disk file. Highlighting the Management VM Snapshot displays its details in the properties window.



Note: In Windows 2012 R2, Microsoft changed Hyper-V backup performance from using VSS Snapshots to checkpoints. This means that every time a request is made to back up a virtual machine, a checkpoint is created and all of the writes to the VM are temporarily redirected to the checkpoints AVHDX files. This gives a backup requester a clean, crash consistent copy of the virtual machines VHDX files. After the backup, the checkpoint is merged. The checkpoint AVHDX file as well as a backupSnapshot.xml file becomes part of the ASM Smart Copy file set.

| | |
|--------------|--|
| Logical Path | |
| Host | tmer4r805s30 |
| Files | 4 |
| File 1 | H:\Virtual Machines\F53B1C20-1F3E-47A8-8B6B-C54DA98DB730.xml |
| File 2 | H:\Virtual Hard Disks\Disk0.vhdx |
| File 3 | H:\Snapshots\F53B1C20-1F3E-47A8-8B6B-C54DA98DB730-BackupSnapshot.xml |
| File 4 | H:\Virtual Hard Disks\Disk0-AutoRecovery.avhdx |

3.3.2 Using HIT/Microsoft PowerShell tools to create a Smart Copy snapshot

The EQLASMPSTools PowerShell library was released with EQL HIT/Microsoft version 4.7. This library contains a set of PowerShell commands that can perform much of the same functionality as the ASM/ME GUI. This module needs to be imported into the PowerShell session prior to running any of the new ASM Smart Copy commands. For example, to create a Smart Copy snapshot of a Hyper-V VM use the **new-ASMSmartcopy** command.

1. Ensure that the EQLASMPSTools.dll module is imported.



```
[TMER4R805S30]: PS C:\> import-module "c:\program
files\EqualLogic\bin\EQLASMPSTools.dll"
```

2. Run the New-ASMSmartCopy command.

```
[TMER4R805S30]: PS C:\> New-ASMSmartCopy -ObjectName ManagementVM
-SmartCopyType Snapshot -BackupType copy -ErrorAction Stop
```

Sample output with two key pieces of information: ObjectName and OriginalObjectId.

```
SmartCopyName      : ManagementVM (3/27/2014 3:35:45 PM)
SmartCopyType      : Snapshot
Created            : 3/27/2014 3:35:45 PM
OriginatingComputer : TMER4R805S30.SKYNET.lab.local
OriginalObjectName : ManagementVM
isAccessible       : True
ObjectName         : SnapshotSet{f3c802ee-a5f4-402a-bc93-024a2b40e98f}
OriginalObjectId   : Component{F53B1C20-1F3E-47A8-8B6B-C54DA98DB730}
```

- ObjectName: The snapshot ID given by the VSS writer to the Smart Copy. This name is unique to a specific smart copy.
- OriginalObjectId: The actual objectname of the ASM object that the smart copy was created for. In this case, it contains the virtual machine ID assigned by the VMMS for the VM.

In the above command, the virtual machine name (ManagementVM) is specified for the -ObjectName parameter. The ObjectName parameter is used to identify the ASM Object that a Smart Copy is to be created for. This ASM object can be a component, collection, volume, or other object. It is important that the value entered has a unique ObjectName parameter. This is why it is often best to specify the actual ObjectName of the component with the -objectname parameter. For example, a particular ASM protected Hyper-V environment might have a virtual machine named "Production1" also in addition to another component (such as a volume) with the same name of Production1. In this case, the new-ASMSmartCopy command would detect a redundancy and end in an error because it could not decipher which component the user wishes to create the Smart Copy for. The ObjectName will always be unique for a specific component.

3. Run the following commands to determine the actual ObjectName for any ASM object.
 - a. For a volume:

```
[TMER4R805S30]: PS C:\ > get-asmvolume | ft -Property
Volume,VolumeName,ObjectName -AutoSize
```

Sample Output:

| Volume | VolumeName | ObjectName |
|--------|-----------------|--|
| G:\ | HyperVDataStore | Volume{d265a214-e21c-4102-8a47-7af5d39868a0} G:\ |
| H:\ | ManagementVol | Volume{42147156-a3b0-11e3-80bd-90e2ba250dbd} H:\ |



b. For virtual machines on a Hyper-V server:

```
[TMER4R805S30]: PS C:\> get-asmcomponent |  
? {$_.writername -like "*Hyper-V*"} | ft -property ComponentName,ObjectName -  
AutoSize
```

Sample Output:

| ComponentName | ObjectName |
|----------------|---|
| ApplicationVM2 | Component{4BA0BF8A-D08B-4347-B6EB-B810D8257A67} |
| ApplicationVM3 | Component{5D973DC3-AEA7-41B8-9744-F0AB8B7EBB28} |
| ApplicationVM1 | Component{E0921359-9E00-477E-8D97-80A6FC122F63} |
| ManagementVM | Component{F53B1C20-1F3E-47A8-8B6B-C54DA98DB730} |
| Host | Component{Host Component} |

c. For collections:

```
[TMER4R805S30]: PS C:\ > get-asmcollection |  
ft -Property CollectionName,Component,ObjectName -AutoSize
```

Sample Output:

| CollectionName | Component | ObjectName |
|----------------|--|---------------------------|
| ProductionVMs | {ApplicationVM2, ApplicationVM3, ApplicationVM1} | Collection{ProductionVMs} |

Examples: Creating Smart Copy Snapshot for ManagementVM using its ObjectName:

- To create a Smart Copy snapshot of the HyperVDataStore volume using its objectname:

```
[TMER4R805S30]: PS C:\ > New-ASMSmartCopy -ObjectName  
"Component{F53B1C20-1F3E-47A8-8B6B-C54DA98DB730}" -SmartCopyType snapshot  
-ErrorAction Stop
```

- Or create a Smart Copy by getting the objectname for the volume dynamically using the following command:

```
[TMER4R805S30]: PS C:\ New-ASMSmartCopy -ObjectName ((get-asmcomponent |  
? {$_.componentname -like "ManagemenVM"}).objectname) -SmartCopyType snapshot  
-ErrorAction Stop
```

Note: if PowerShell commands are run from the management VM against the Hyper-V host, a remote PowerShell connection will need to be opened to the Hyper-V host from the ManagementVM. See the Dell Technical White Paper ["Deploying a Dedicated Hyper-V Management Guest using PowerShell and Dell EqualLogic Storage and Tools"](#) for more information on opening a remote PowerShell session.

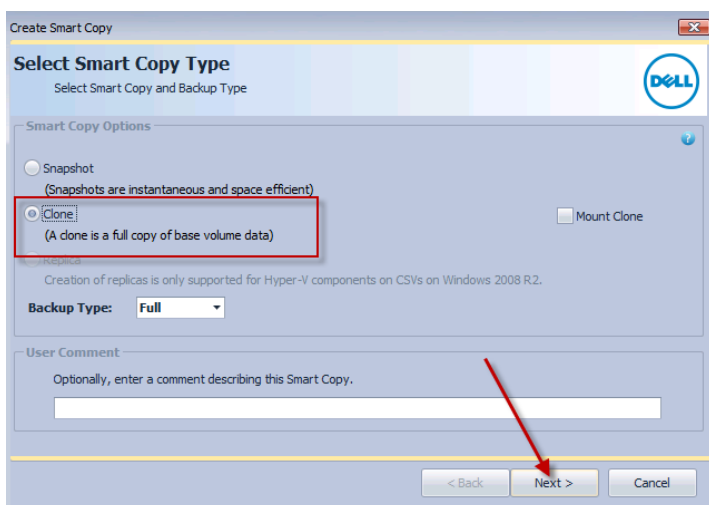
For more information on using the EQLASMPSTools PowerShell commands, go to the *Dell EqualLogic PowerShell Users Guide* found in the c:\Program Files\EqualLogic\Docs directory.



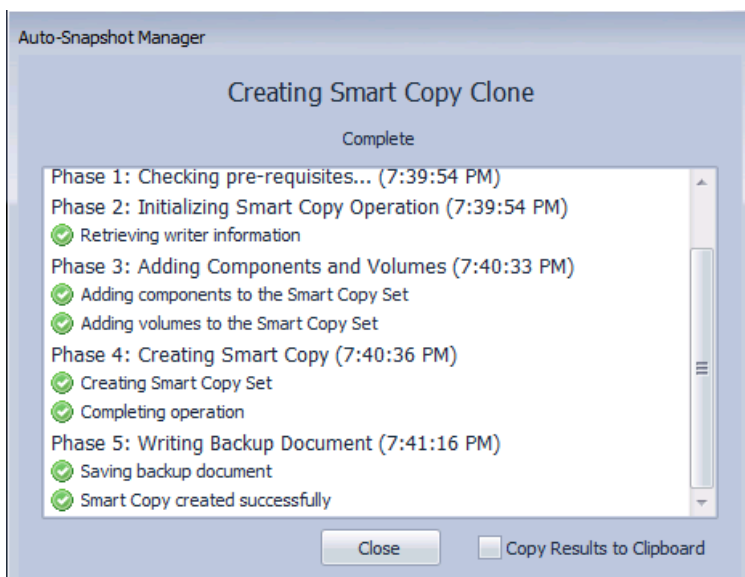
3.3.3 Using ASM/ME to create Smart Copy Clone


A Smart Copy Clone can be made of the virtual machine and is a full volume copy. This operation creates a new EqualLogic volume of the same size as the volume which houses the virtual machine and virtual hard disk.

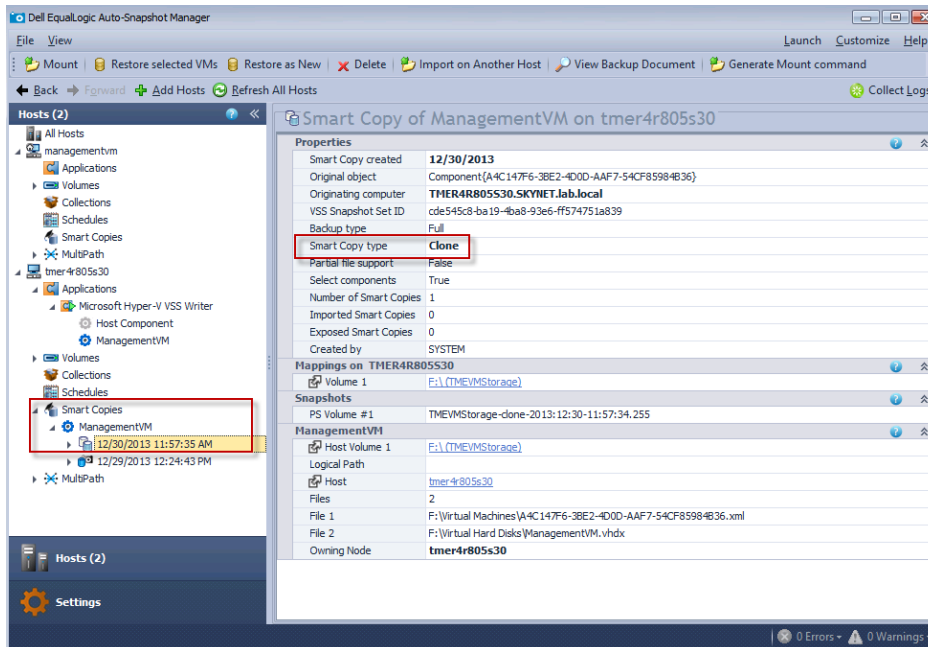
Creating a Smart Copy clone is performed by the same method as creating the Smart Copy Snapshot of the virtual machine, with the exception being that **Clone** is selected and the option to mount the clone once it is created is offered to the user.



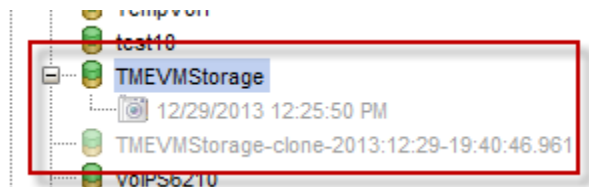
Like the Snapshot creation, a progress window is displayed while the clone is being created.



When the Smart Copy Clone of the virtual machine is complete, it appears in the virtual machine list with any other Smart Copies under the Smart Copies node in the navigation tree. The Smart Copy Clone of the Management Virtual Machine is distinguished by the  icon



In Group Manager, both the Smart Copy Snapshot and Smart Copy Clone are visible under the TMEVMStorage volume. Note that the Smart Copy Clone appears as an offline volume.



On the Hyper-V host, the Smart Copy Clone can be seen in the `get-iscsitarget` command as an unconnected iSCSI target.

```
[TMER4R805S30]: PS C:\ > get-iscsitarget | ft -AutoSize
```

```
IsConnected NodeAddress
-----
True ign.2001-05.com.equallogic:0-1cb196-b481da94f-d8600000042522f7-vss-control
True ign.2001-05.com.equallogic:0-1cb196-dcc1da94f-b06000000af525d7-tmevmstorage
False ign.2001-05.com.equallogic:0-... tmevmstorage-clone-2013:12:29-19:40:46.961
```

3.3.4 Using HIT/Microsoft PowerShell Tools to create a Smart Copy Clone

To create a Smart Copy clone of a virtual machine, the **`new-asmsmartcopy`** command is used. In this case, the **`-smartcopytype`** value is **`clone`**.

```
[TMER4R805S30]: PS C:\ > New-ASMSmartCopy -ObjectName ((get-asmcomponent |  
? {$_.componentname -like "ManagementVM"}).objectname) -SmartCopyType clone  
-ErrorAction Stop
```

Sample output:

```
SmartCopyName      : ManagementVM (3/27/2014 3:41:58 PM)  
SmartCopyType      : Clone  
Created            : 3/27/2014 3:41:58 PM  
OriginatingComputer : TMER4R805S30.SKYNET.lab.local  
OriginalObjectName : ManagementVM  
isAccessible       : True  
ObjectName         : SnapshotSet{4e97c3d7-b650-4ef5-a1f2-322f668f0e21}  
OriginalObjectId   : Component{F53B1C20-1F3E-47A8-8B6B-C54DA98DB730}
```

In the above output, notice that the Objectname remained the same and the OriginalObjectID is unique compared to the output from the previous Smart Copy snapshot operation for the ManagementVM.

3.4 Creating a Smart Copy schedule for virtual machines

A powerful feature of ASM/ME is the ability to create schedules for Smart Copy operations. The frequency of a schedule and the number of Smart Copies ASM/ME stores concurrently are specified when the schedule is created.

Whether a Smart Copy of an online VM is application-consistent or crash-consistent depends on the type of operating system and where the Integration Services (or Virtual Guest Services) are installed.

If a VM is offline, it is still copied in its offline state. The Smart Copy will reflect the state of the VM. If the VM was shutdown cleanly, then the Smart Copy will be application-consistent. If the VM crashed, or was not shutdown cleanly, then the Smart Copy will be crash-consistent. Ensure that Smart Copy schedules are modified accordingly prior to removing a VM so that related schedules or collections pertaining to that VM will not result in failures.

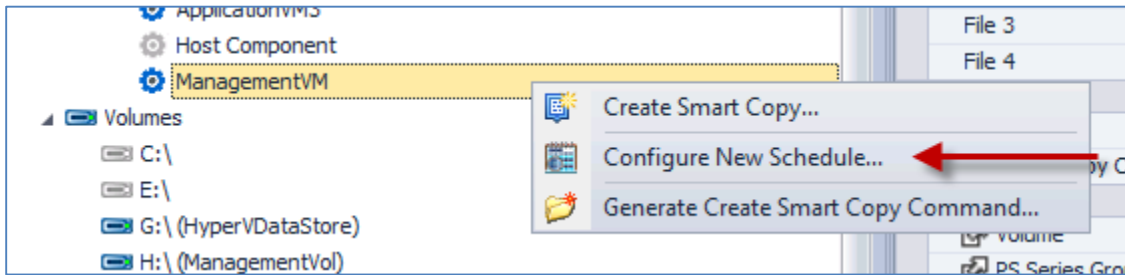
In a Hyper-V Microsoft failover cluster, ASM/ME creates, modifies and deletes scheduled tasks on all cluster nodes. Tasks are scheduled on all nodes by default, but only properly run on the node that is the current owner of the target virtual machines. If a cluster node fails or goes offline, the schedule fails over to the node that becomes the owner of the virtual machines. Additionally, changes made within Microsoft Failover Cluster Manager, such as changing the owner of a clustered resource, are automatically detected by ASM/ME.

Note: Although ASM/ME recognizes the quorum disk in a Hyper-V Microsoft Failover Cluster, schedules cannot be created for the quorum disk.

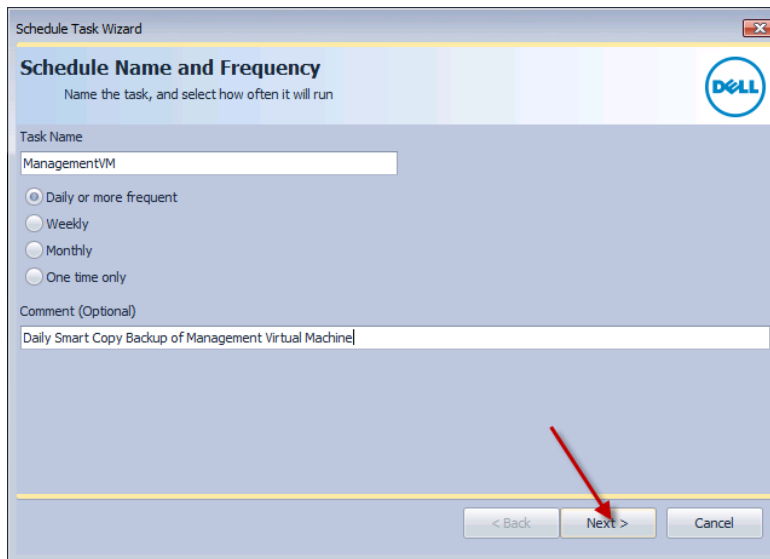
Use the following procedure to create a Smart Copy Schedule for a virtual machine. This same procedure could be done for other ASM components such as a volume or collection.



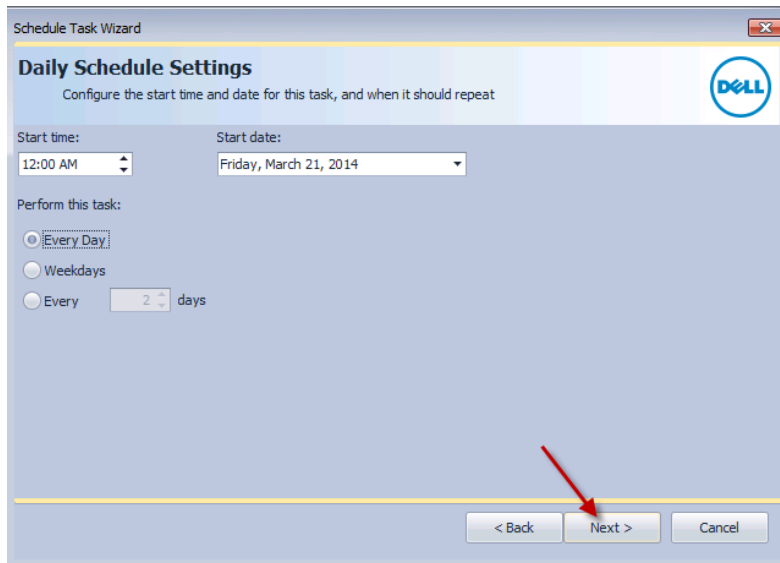
1. In the navigation tree, right-click the virtual machine and Select **Configure New Schedule**.



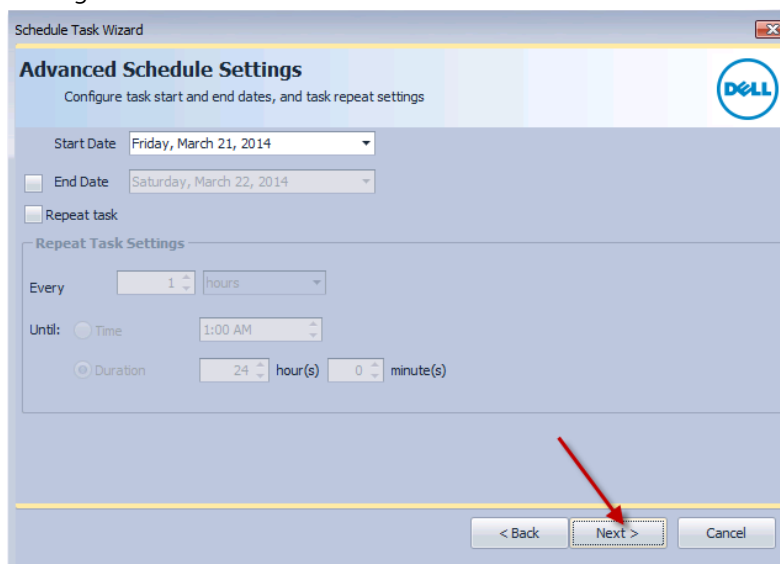
2. In the **Schedule Name and Frequency** dialog, specify the schedule name and frequency.
 - a. Select **Daily or more frequent**, **Weekly**, **Monthly** or **One time only**, depending on how frequent a Smart Copy is needed.
 - b. Optionally, enter a comment in the **Comment** field.
 - c. Click **Next**.



3. In the **Schedule Settings** dialog, select the schedule settings based on the frequency previously chosen. Click **Next**.



4. In the **Advanced Schedule Settings** dialog, configure the start and end dates, as well as repeat settings. Click **Next**.



5. In the **Smart Copy Options** dialog, select the type of Smart Copy and the number of Smart Copies to retain.

Schedule Task Wizard

Smart Copy Options
Select type of smart copy this schedule will perform, and the number of smart copies to retain at a time

Smart Copy Options

☒ Snapshot Backup Type: Full

☐ Replica

Keep Count Setting

Maximum number of smart copies to keep (0 - 512) 8

If the number of Smart Copies for this volume or collection exceeds the KeepCount value, they will be automatically deleted starting with the oldest first. To prevent deletion of Smart Copies, specify zero.

< Back Next > Cancel

6. In the **Provide User Account Information** dialog, specify the user account that will run the scheduled task.
 - In a Hyper-V Microsoft failover cluster, select **Run task as specified user** and provide login credentials with cluster administrative access.
 - Ensure that the specified account has access to the shared backup document location.
 - Click **Create** to finish creating the schedule.

Schedule Task Wizard

Provide User Account Information
Specify the user account which will be used to run this scheduled task

User account to create Smart Copy and global verification tasks

☐ Run task as system user

☒ Run task as specified user

User Information

Username: TMER-4R80SS30\Administrator

Password:

Confirm Password:

< Back Create Cancel

3.5 Working with a Collection of Virtual Machines

In ASM/ME, a collection is a logical group of components. In the case of Hyper-V virtual machines, a collection consists of either a group of virtual machines that house versions of a production application, or a set of virtual machines used for development purposes. Once a collection is created, an administrator can then perform Smart Copy operations against the entire collection.

Note: Dell recommends creating collections only for objects that would potentially be restored as a group (for example, a group of production virtual machines).

The two types of collections that can be created in ASM/ME are volume based and component based. Using a volume based collection enables any scheduled tasks for the collection to complete successfully even if the collection contents have changed. The benefit of a volume based collection is that it provides more flexibility with regards to Smart Copy schedules as well as creating and deleting virtual machines in the environment. If an administrator selects the option to create a volume based virtual machine collection, and then later, deletes or adds a virtual machine, the collection definition will update accordingly to allow any Smart Copy schedules for the collection to run successfully. However, a component based collection does not allow this. A scheduled task using the component based collection will fail if there is a change in the collection definition (such as a deleted VM). An administrator will need to manually modify the particular component based collection that contained the deleted virtual machine.

Note: A collection that contains a clustered shared volume will, by default, be a volume based collection, even if the administrator does not specify a volume based collection at the time of creation. For servers that boot to the SAN, boot volumes can now be added to collections in ASM/ME. Dell does not recommend storing Hyper-V or SQL data on boot volumes. If a Smart Copy is created of a collection containing a boot volume, a selective restore of Hyper-V, SQL or Exchange application components from the Smart Copy can be performed. However, an in-place restore operation of the boot volume from the Smart Copy cannot be performed.

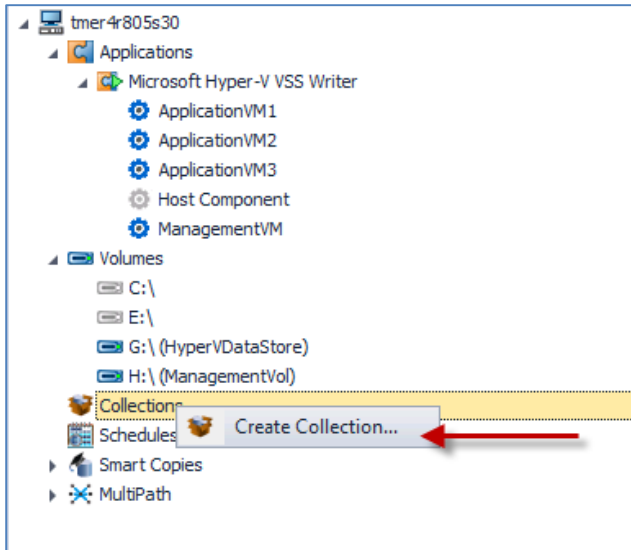
For more in-depth information on collections, please consult the Dell EqualLogic Auto-Snapshot Manager Microsoft Edition Users Guide found at `c:\Program Files\EqualLogic\Doc`

3.5.1 Use ASM/ME to create a collection of virtual machines

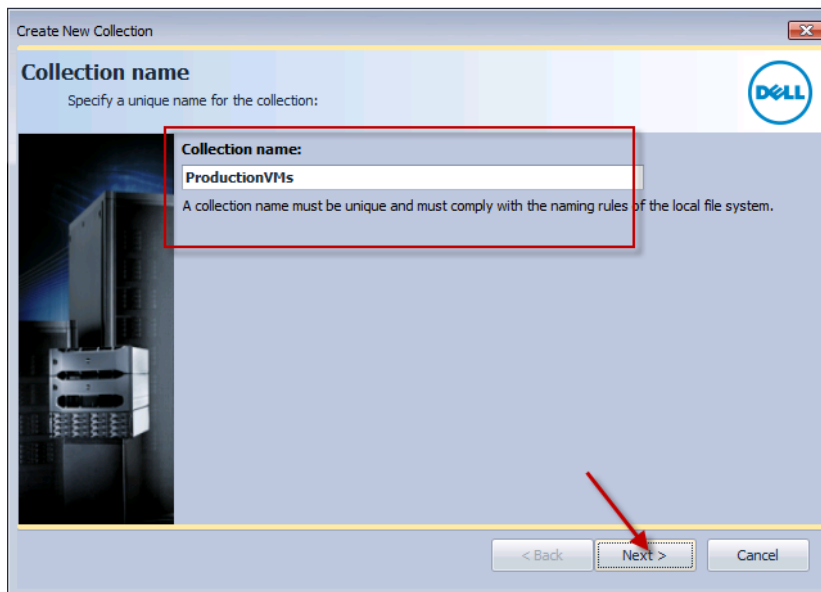
Use the following procedure to create a collection of virtual machines. In the following example, three virtual machines (ApplicationVM1, ApplicationVM2, and ApplicationVM3) share a common data store volume (G:\) and are placed into a volume based collection (ProductionVMs).

1. Right-click the **Collections** node in the navigation tree and select **Create Collection**.

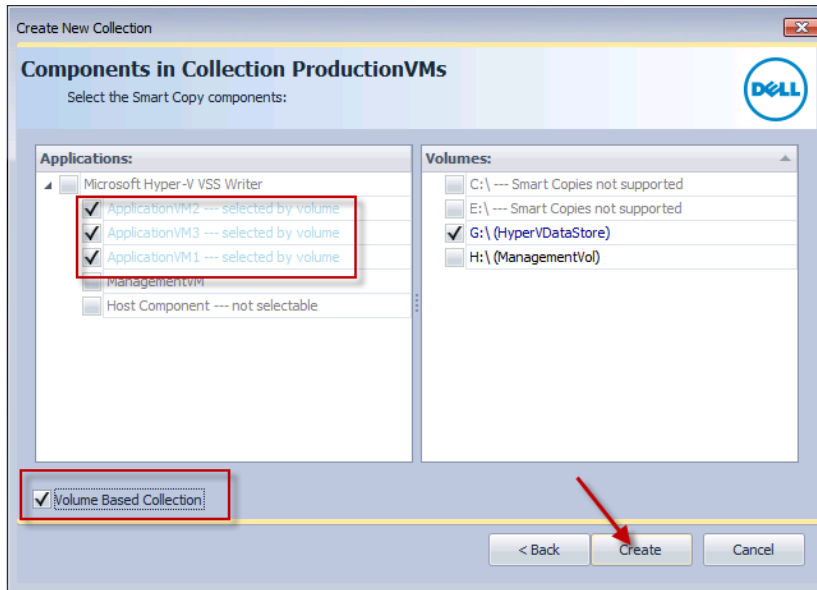




2. Enter a name for the collection and click **Next**.



3. Select the virtual machines to include in the collection from the left pane. ASM/ME will automatically select the required volumes from the right pane.
4. Select **Volume Based Collection**; The selected virtual machines will grey out.
5. Click **Create** to create the volume based collection.



3.5.2 Use HIT/Microsoft PowerShell Tools to create a collection of virtual machines

The volume based collection in the previous section can be created using the **New-ASMCollection** command found with the EQLASMPSTool.dll library.

```
[TMR4R805S30]: PS C:\> New-ASMCollection -CollectionName ProductionVMs
-Component ApplicationVM1, ApplicationVM2, ApplicationVM3 -IsVolumeBased $true
```

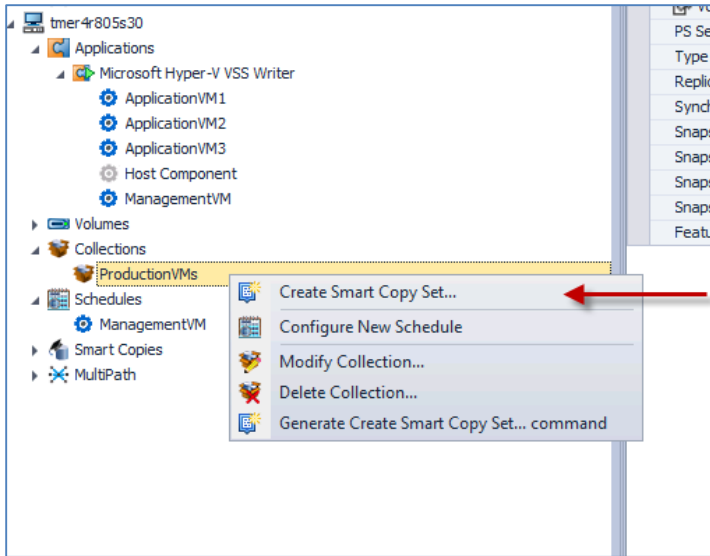
Sample output showing that the ObjectName reflects that the object is a collection. The components are an array comprised of the names of the virtual machines in the collection.

```
CollectionName : ProductionVMs
Volume         : {G:\}
Component      : {ApplicationVM1, ApplicationVM2, ApplicationVM3}
IsVolumeBased  : True
ObjectName     : Collection{ProductionVMs}
```

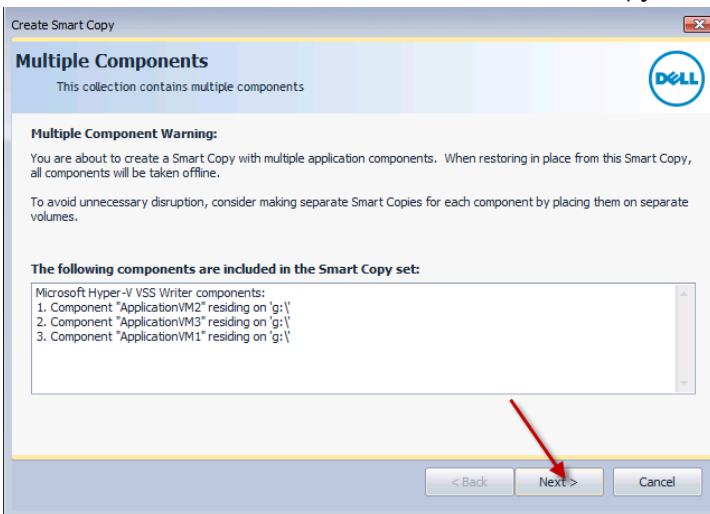
3.5.3 Use ASM/ME to create a Smart Copy of a collection of virtual machines

In ASM/ME, creating a Smart Copy of a virtual machine collection is no different than taking a smart copy of a single virtual machine.

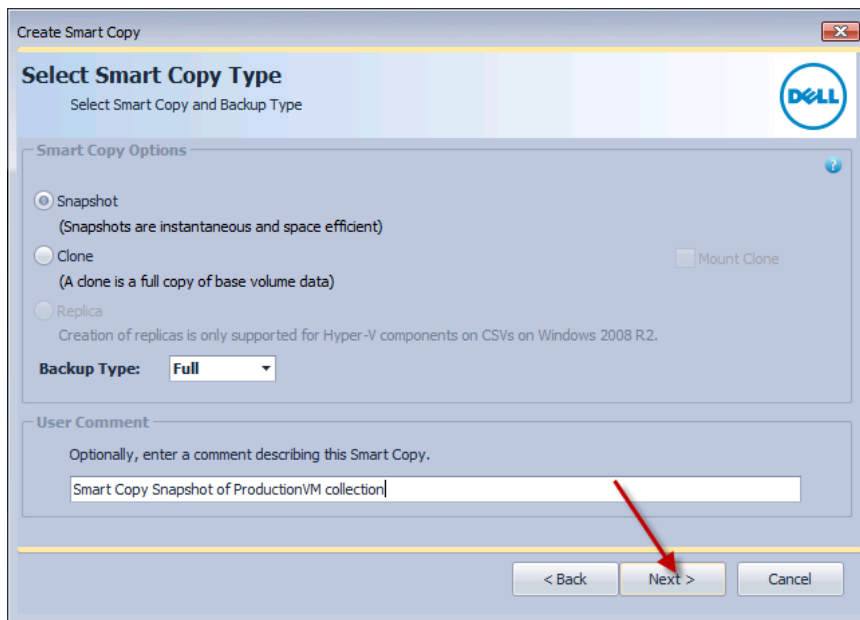
1. Right-click the collection in the navigation tree and select **Create Smart Copy Set**.



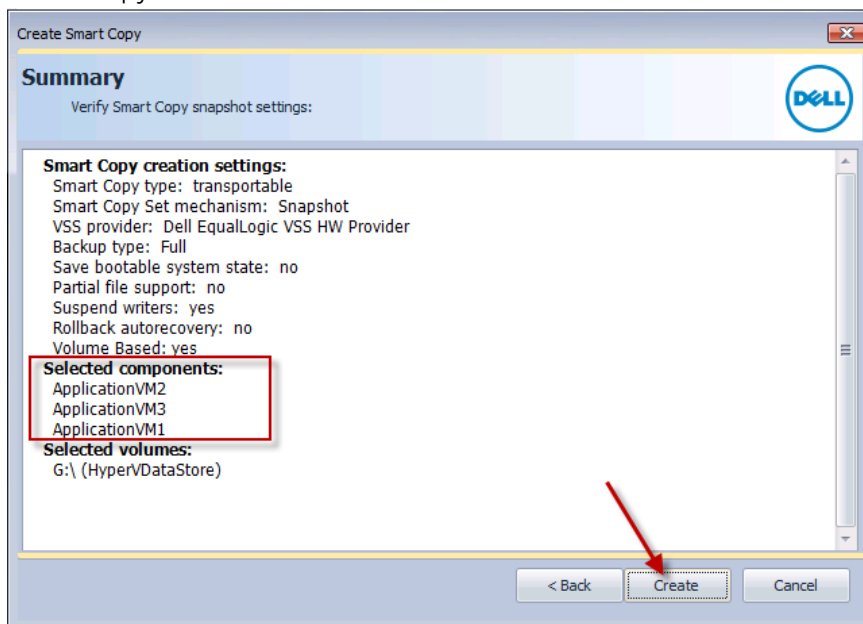
2. Click **Next** after reviewing the notification that multiple components (virtual machines) associated with the collection will be included in the Smart Copy



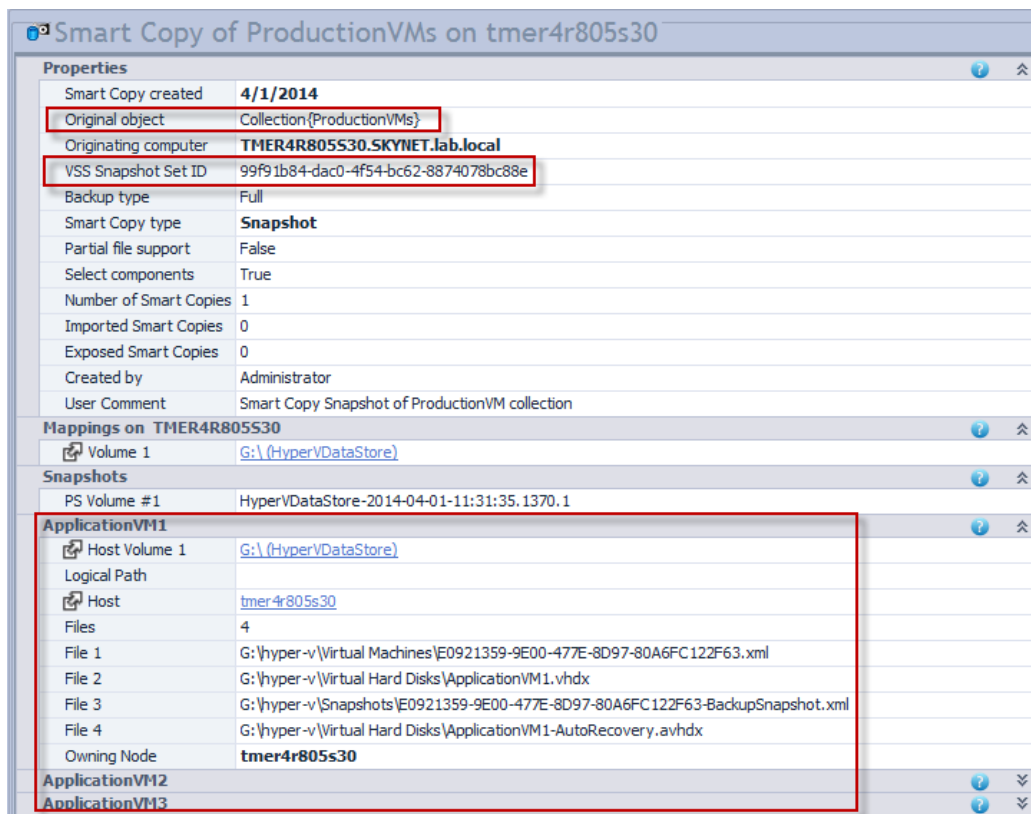
3. Choose a type of Smart Copy (Snapshot or Clone) and backup type. A descriptive comment may also be entered. Click **Next**. In this example, a Snapshot of the collection will be created.



4. In the Summary Window, review the displayed data and then click **Create** when ready to create the Smart Copy.



5. Once the Smart Copy of the collection is created, the properties may be examined by selecting it in the navigation tree.



In the above output, the original object ID (**Collection(ProductionVMs)**), the VSS Snapshot Set ID, and the three virtual machines (ApplicationVM1, ApplicationVM2, and ApplicationVM3) are listed. If the virtual machine components are expanded, the virtual machine files involved in the snapshot are shown.

This information can also be displayed using the EQLASMPSTool PowerShell toolset. Information about the specific collection can be displayed with the **Get-ASMCollection** command and using the name of the collection (ProductionVMs) as input for the **-ObjectName** parameter.

```
[TMER4R805S30]: PS C:\> Get-ASMCollection -ObjectName ProductionVMs
```

Sample Output:

```
CollectionName : ProductionVMs
Volume         : {G:\}
Component      : {ApplicationVM2, ApplicationVM3, ApplicationVM1}
IsVolumeBased  : True
ObjectName     : Collection{ProductionVMs}
```

The output from the **Get-ASMCollection** command can be passed directly into the **Get-ASMSmartCopy** command. This will return information about all of the Smart Copies associated with the specified collection.

```
[TMER4R805S30]: PS C:\ > Get-ASMCollection -ObjectName ProductionVMs | Get-ASMSmartCopy
```


Sample output showing that the SmartCopy ObjectName and OriginalObjectId of the collection are the same:

```
SmartCopyName      : ProductionVMs (4/1/2014 11:34:13 AM)
SmartCopyType      : Snapshot
Created           : 4/1/2014 11:34:13 AM
OriginatingComputer : TMER4R805S30.SKYNET.lab.local
OriginalObjectName : ProductionVMs
isAccessible       : True
ObjectName         : SnapshotSet{99f91b84-dac0-4f54-bc62-8874078bc88e}
OriginalObjectId   : Collection{ProductionVMs}
```

In the example, there is only one existing Smart Copy of the collection. If there were multiple Smart Copies of the collection, all of them would have been displayed in the output.

3.5.4 Use HIT/Microsoft PowerShell Tools to create a Smart Copy of a collection of virtual machines

The previous operation of creating a Smart Copy snapshot of a virtual machine collection could be accomplished with the **New-ASMSmartCopy** PowerShell command:

Notice that in this example, **OriginalObjectId** of the collection was used for the **-objectname** parameter. When **OriginalObjectId** is used, it needs to be surrounded by quotation marks in the command. The **OriginalObjectName (ProductionVMs)** could have also been used.

```
[TMER4R805S30]: PS C:\> New-ASMSmartCopy -ObjectName "Collection{ProductionVMs}" -smartcopytype
snapshot -backuptype copy -ErrorAction Stop
```

Sample output:

```
SmartCopyName      : ProductionVMs (4/1/2014 2:09:27 PM)
SmartCopyType      : Snapshot
Created           : 4/1/2014 2:09:27 PM
OriginatingComputer : TMER4R805S30.SKYNET.lab.local
OriginalObjectName : ProductionVMs
isAccessible       : True
ObjectName         : SnapshotSet{87bcbec7-9894-4c50-a44d-402ffb5addc2}
OriginalObjectId   : Collection{ProductionVMs}
```

3.6 Working with Clustered Shared Volumes (CSVs)

The value of Microsoft failover clustering and CSVs cannot be overstated with regards to Hyper-V environment. CSVs are beneficial in a Hyper-V Microsoft failover cluster because they enable multiple cluster nodes to access common CSVs. Therefore, multiple nodes can potentially host or run VMs residing on the same CSV to provide high availability to the Hyper-V environment.

Note: For information about working with Microsoft Failover Clusters and Dell EqualLogic Storage please see the Dell Technical White Paper: Using Dell EqualLogic Storage with Microsoft Failover Clusters and Hyper-V



In an ASM/ME Hyper-V protected environment, a CSV is another ASM/ME component, where the following Smart Copy operations are available.

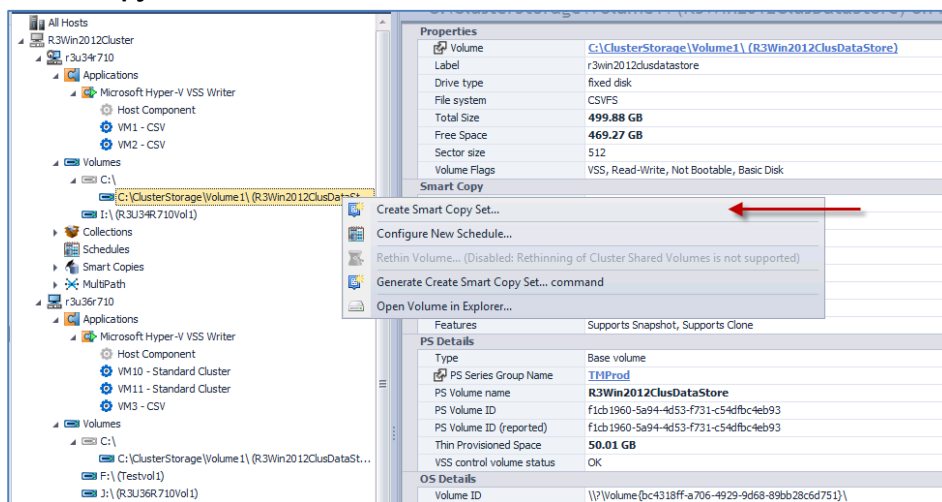
- Smart Copies of VMs residing on CSVs
- Smart Copies of CSVs
- Collections of CSVs
- Smart Copy Schedules of CSVs

ASM/ME Smart Copies in a CSV-enabled cluster provide significant performance improvements because only one schedule is needed to capture every VM residing on a CSV. Administrators no longer need to create individual schedules for each VM on the CSV which in turn greatly simplifies CSV Smart Copy schedule management. Additionally, since only a single schedule is needed to capture every VM residing on a CSV, less snapshot reserve space is consumed in the Dell EqualLogic PS Series group. Finally, a Smart Copy of a CSV taken from a cluster node contains all the virtual machines that reside on the CSV, including those that are running on other cluster nodes. Smart Copies of CSVs are file system-consistent.

3.6.1 Use ASM/ME to create a Smart Copy of a CSV

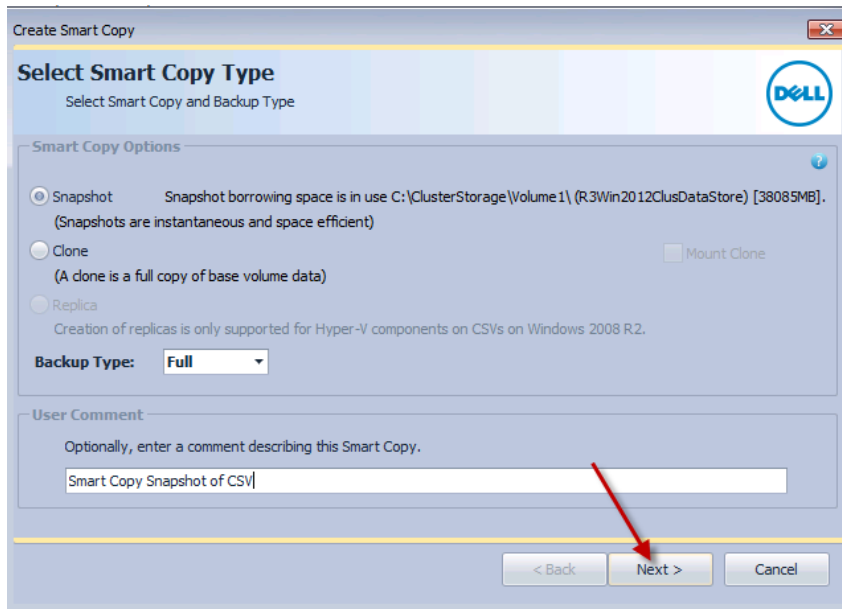
The procedure for creating a Smart Copy of a CSV in ASM/ME is the same as creating a Smart Copy of any other ASM/ME component.

1. Click to navigate to the desired Cluster Shared Volume. Right-click the volume and select **Create Smart Copy Set...**

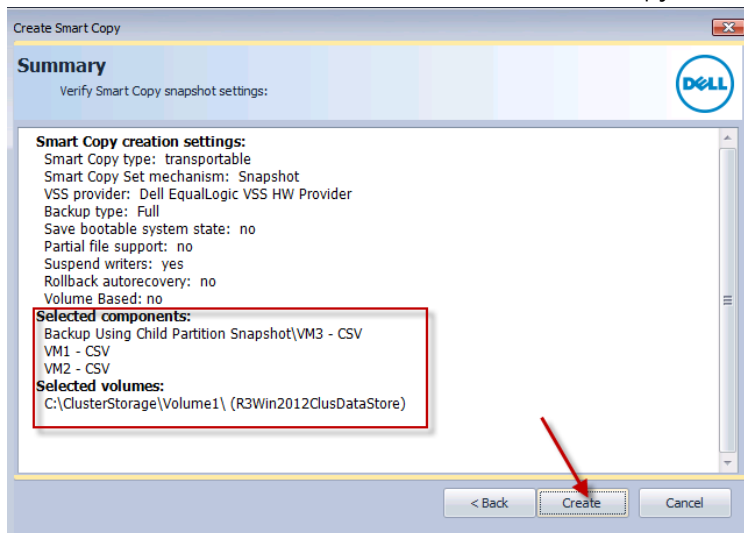


2. Select the type of Smart Copy to create (Snapshot of Clone) and optionally a comment to describe the smart copy. The only option available for backup type for a CSV is full. Click **Next** to continue



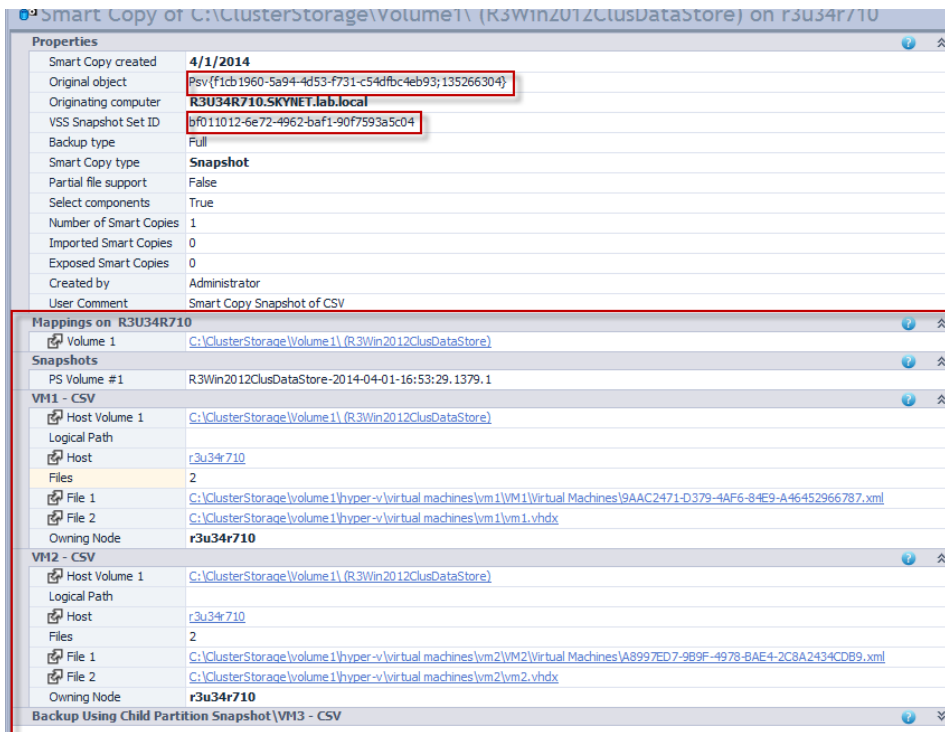


3. Review the summary screen and select **Create** when ready to create the smart copy. Note that in the example summary screen shot the component VMs owned by the node from where the Smart Copy is being taken from (VM1 – CSV and VM2 – CSV) as well as the VM owned by the other node in the cluster (VM3 – CSV) will be included in the Smart Copy.



4. Once the Smart Copy of the CSV is created, select it and examine its properties. Identify the Original Object, VSS Snapshot Set ID, and the components included in the Smart Copy





3.6.2 Use HIT/Microsoft PowerShell tools to create a Smart Copy of a CSV

1. As with the previous ASM PowerShell commands, first get the ObjectName of the CSV where the Smart Copy will be created. In the above example, the CSV is c:\ClusterStorage\Volume1.
2. Store the Object Name of the CSV into a variable

```
PS C:\> $CSVObjName = (get-asmvolume |
? {$_.volume -like "C:\ClusterStorage\Volume1\*"}).objectName
```

```
PS C:\> echo $CSVObjName
```

```
Volume{bc4318ff-a706-4929-9d68-89bb28c6d751}|C:\ClusterStorage\Volume1\
```

3. Create the Smart Copy using the **new-ASMSmartCopy** command and passing the CSV Object Name variable into the **-objectname** parameter. Note the use of quotes around the object name variable.

```
PS C:\> New-ASMSmartCopy -ObjectName "$CSVObjName" -SmartCopyType snapshot
-Backuptype full -ErrorAction Stop
```

Sample Output:

```
Starting operation...
Creating Smart Copy set for Psv{f1cb1960-5a94-4d53-f731-c54dfbc4eb93;135266304}...
```

```

SmartCopyName       : C:\ClusterStorage\Volume1\ (R3Win2012ClusDataStore) (4/2/2014 1:31...
SmartCopyType       : Snapshot
Created             : 4/2/2014 1:31:19 PM
OriginatingComputer : R3U34R710.SKYNET.lab.local
OriginalObjectName  : C:\ClusterStorage\Volume1\ (R3Win2012ClusDataStore)
isAccessible        : True
ObjectName          : SnapshotSet{7c3c7ad8-8831-405f-90b4-d9577e9a4f90}
OriginalObjectId    : Psv{f1cb1960-5a94-4d53-f731-c54dfbc4eb93;135266304}

```

3.7 Hyper-V VM Smart Copy restore operations

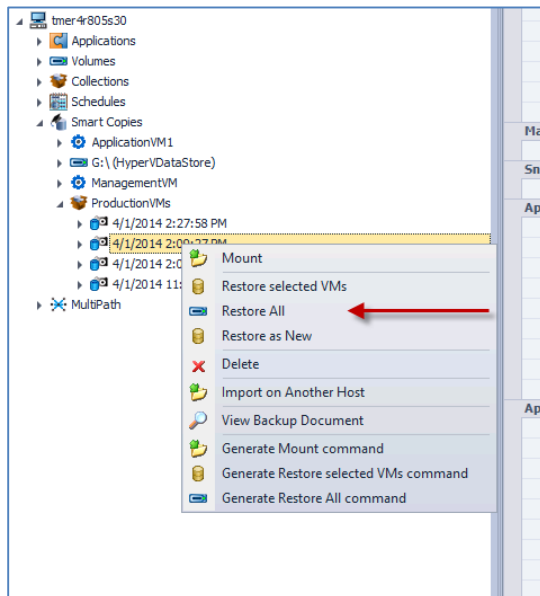
Being able to restore a virtual machine is a critical part of Hyper-V management operations. ASM/ME allows you to perform the following restore operations of Microsoft Hyper-V virtual machines as well as volumes used by the VMs.

- In-Place Restore
- Selective Restore
- Restore as New

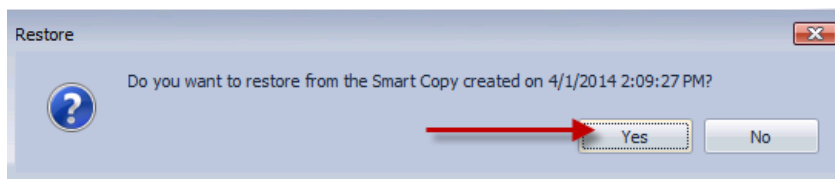
3.7.1 Use ASM/ME to perform an In-Place Restore of Hyper-V VMs

This operation will restore all data, on all PS Series volumes, used by all VMs in the Smart Copy. VMs being restored in-place will experience a service interruption. The following example will perform an in-place restore all of the virtual machines in a collection.

1. Select a Smart Copy of an existing virtual machine collection, right-click on the Smart Copy and click **Restore All**.



2. A message dialog appears asking to verify the restore all choice. Click **Yes** to begin the restore operation.



If the virtual machines associated with the collection were running prior to the restore, they are shut down and taken into the off state by the VMMS service.

3. To view a list of all the VMs associated with the Hyper-V server and their information, use the **get-vm** command.

```
[TMER4R805S30]: PS C:\> get-vm
```

| Name | State | CPUUsage (%) | MemoryAssigned (M) | Uptime | Status |
|----------------|---------|--------------|--------------------|----------|--------------------|
| ----- | ----- | ----- | ----- | ----- | ----- |
| ApplicationVM1 | Off | 0 | 0 | 00:00:00 | Operating normally |
| ApplicationVM2 | Off | 0 | 0 | 00:00:00 | Operating normally |
| ApplicationVM3 | Off | 0 | 0 | 00:00:00 | Operating normally |
| ManagementVM | Running | 3 | 4096 | 04:02:38 | Operating normally |

4. If the VMs were running prior to the restore, the restore process will attempt to restart them when completed. If the VMs were in the off state, the restore process will not attempt a restart and the VMs will have to be manually restarted.

The in-place restore process has the following steps

Started at 3:49:26 PM

Gathering Information...

Phase 1: Validating parameters (3:49:27 PM)

Phase 2: Stopping guests (3:49:28 PM)

Phase 3: Restoring guests (3:49:29 PM)

Phase 4: VSSInitializeForRestore (3:49:29 PM)

 Initializing restore operation

Phase 5: VSSSetRestoreComponents (3:49:29 PM)

Phase 6: VSSPreRestore (3:49:30 PM)

 Preparing for restore

Phase 7: LogoffLocalVolumes (3:49:34 PM)

 Restoring snapshots

 Setting volume G:\ offline on the computer

Phase 8: OfflineVolumesAndSnapshots (3:49:35 PM)

 Setting volume G:\ offline on PS Series group

Phase 9: RollbackSnapshots (3:49:35 PM)

 Restoring snapshot on PS Series group #1

Phase 10: OnlineVolumesAndSnapshots (3:49:35 PM)

 ringing volume G:\ online on the PS Series group



Phase 11: OnlineLocalVolumes (3:49:42 PM)
 Logging host in to volume G:\

Phase 12: VSSPostRestore (3:49:56 PM)
 VSS Writer clearing metadata
 VSS Writer performing post restore operations
 Checking VSS Writer status
 Completing operation

Phase 13: Starting guests (3:49:57 PM)
 Restore completed successfully.

3.7.2 Use HIT/Microsoft PowerShell Tools to perform an In-Place Restore of Hyper-V VMs

The In-Place Restore operation discussed in Section 3.7.1 can also be accomplished using the EQLASMPSTool PowerShell command **restore-asmsmartcopy**.

1. Identify the smart copies associated with the virtual machine collection (ProductionVMs) and select a smart copy to use for the restore operation.

```
[TMER4R805S30]: PS C:\> get-asmcollection | ? {$_.CollectionName -like
"ProductionVMs"} | get-asmsmartcopy | ft -property SmartCopyName,ObjectName
-AutoSize
```

| SmartCopyName | ObjectName |
|--------------------------------------|---|
| ----- | ----- |
| ProductionVMs (4/1/2014 11:34:13 AM) | SnapshotSet{99f91b84-dac0-4f54-bc62-8874078bc88e} |
| ProductionVMs (4/1/2014 2:08:09 PM) | SnapshotSet{3f4c93ea-2472-4cbb-a93d-5aec718df80b} |
| ProductionVMs (4/1/2014 2:09:27 PM) | SnapshotSet{87bcbec7-9894-4c50-a44d-402ffb5addc2} |
| ProductionVMs (4/1/2014 2:27:58 PM) | SnapshotSet{06522dec-d3df-4ccb-8b20-e42d9eb48fcc} |

2. Using the selected smart copy object name, perform the restore in-place of the collection and all of the virtual machines associated with the following **restore-asmsmartcopy** command.

```
[TMER4R805S30]: PS C:\ > Restore-ASMSmartCopy -ObjectName
"SnapshotSet{3f4c93ea-2472-4cbb-a93d-5aec718df80b}" -ErrorAction Stop
```

3. Optional: Examine restore operation logs. Examine the Microsoft-Windows-Hyper-V-VMMS-Admin log with **get-winevent**.

```
[TMER4R805S30]: PS C:\> Get-WinEvent -LogName Microsoft-Windows-Hyper-V-
VMMS-Admin
```

The EQL Trace Logs (eqltrace.log and eqltrace.0.log) can also be queried using the **get-content** command. This log can be large, so pipe the output to the where-object (?) command, focusing on the time the operation was run.



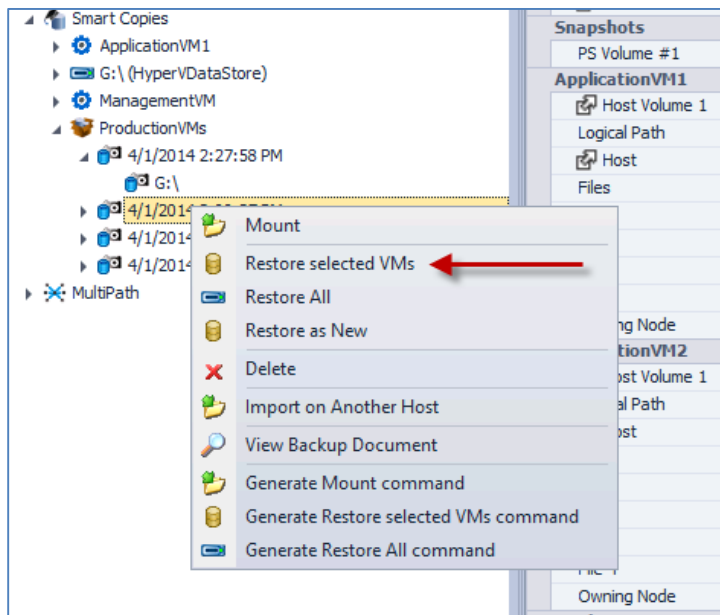
```
[TMER4R805S30]: PS C:\> Get-Content -Path
C:\ProgramData\EqualLogic\log\eqlltrace.log | ? {$_ -like "02-Apr-14 16:23*"}
```

3.7.3 Using ASM/ME to perform a Selective Restore operation for a specific Hyper-V VM

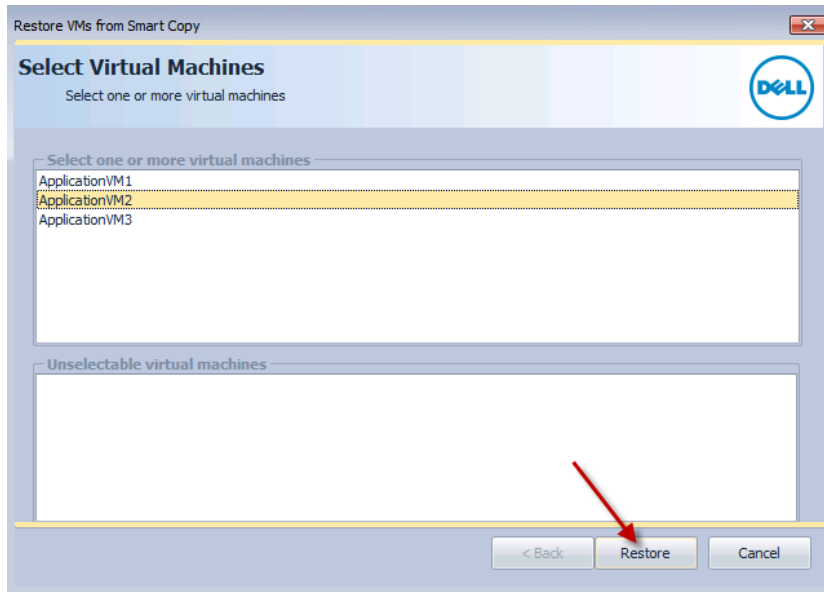
Because ASM/ME uses the Microsoft's VSS, it is important to remember that Smart Copies are volume based. If a particular virtual machine has VHDX and configuration files on an EQL volume that houses other virtual machine VHDXs, then the Smart Copies of that virtual machine will also include the other virtual machine VHDXs on the volume. This is often the case with CSVs and virtual machine volume based collections. Fortunately, an administrator can use ASM/ME take a Smart Copy of the volume or collection, and then selectively restore only a specific virtual machine, without impacting the other virtual machines in the collection or volume.

1. To restore the virtual machine from the Smart Copy Snapshot taken for the collection in the Section 3.7.2 (ProductionVMs), go to the ASM/ME navigation tree and right-click on the Smart Copy Snapshot for the collection and then select **Restore selected VMs**.

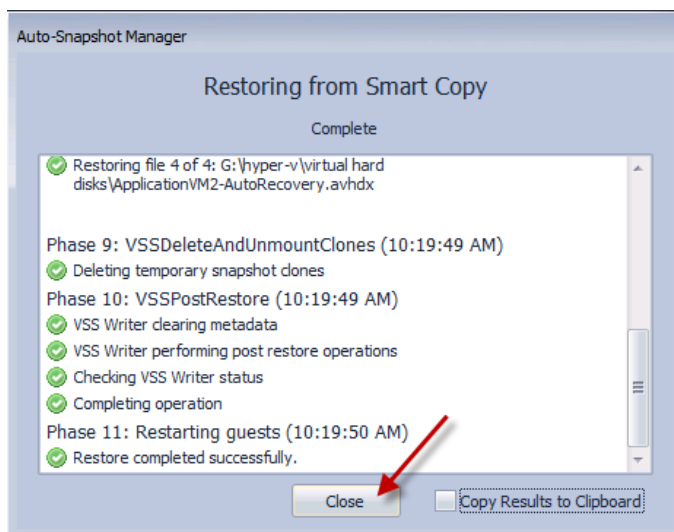
Note: the following procedure could also be used to selectively restore a virtual machine from a Smart Copy of a volume and CSV.



2. The names of the virtual machines associated with the collection Smart Copy is displayed. Select the virtual machine to be restored and then click **Restore** to start the Selective Restore operation. In this example, ApplicationVM2 is selected for restore.



3. The restore operation will freeze I/O to the virtual machine being restored. The VMMS service will then shut down the virtual machine. The VM will be deleted and restored from the Smart Copy. Depending on the state of the virtual machine when the restore was enacted, it will be restarted. The progress of the restore can be followed in the progress window. When the restore completes, click **Close**



The selective-restore process has the following steps:

Started at 10:19:00 AM
 Gathering Information...
 Phase 1: Validating parameters (10:19:00 AM)
 Phase 2: Stopping guests (10:19:02 AM)
 Phase 3: Restoring guests (10:19:03 AM)

Phase 4: VSSInitializeForRestore (10:19:03 AM)
 Initializing restore operation
 Phase 5: VSSSetRestoreComponents (10:19:03 AM)
 Phase 6: VSSPreRestore (10:19:04 AM)
 Preparing for restore
 Phase 7: VSSCloneAndMountSourceVolumes (10:19:08 AM)
 Creating temporary snapshot clones for copy operation
 Phase 8: CopyFiles (10:19:20 AM)
 Restoring file 4 of 4: G:\hyper-v\virtual hard disks\ApplicationVM2-AutoRecovery.avhdx
 Phase 9: VSSDeleteAndUnmountClones (10:19:49 AM)
 Deleting temporary snapshot clones
 Phase 10: VSSPostRestore (10:19:49 AM)
 VSS Writer clearing metadata
 VSS Writer performing post restore operations
 Checking VSS Writer status
 Completing operation
 Phase 11: Restarting guests (10:19:50 AM)
 Restore completed successfully.

Performing a selective-restore for a virtual machine that shares a volume with other virtual machines will not result in the other virtual machines being shut down. The following **get-vm** command sequence on the Hyper-V host shows that only the VM being restored (ApplicationVM2) is affected. The other VMs are not affected by the restore.

- Selective Restore Started:

```
[TMER4R805S3]: PS C:\> get-vm
```

| Name | State | CPUUsage (%) | MemoryAssigned (M) | Uptime | Status |
|----------------|---------|--------------|--------------------|----------|--------------------|
| ---- | ----- | ----- | ----- | ----- | ----- |
| ApplicationVM1 | Running | 0 | 380 | 17:56:06 | Operating normally |
| ApplicationVM2 | Running | 2 | 512 | 17:56:05 | Operating normally |
| ApplicationVM3 | Running | 0 | 372 | 17:56:04 | Operating normally |
| ManagementVM | Running | 0 | 4096 | 10:19:03 | Operating normally |

- ApplicationVM2 Shutdown:

| Name | State | CPUUsage (%) | MemoryAssigned (M) | Uptime | Status |
|----------------|---------|--------------|--------------------|----------|--------------------|
| ---- | ----- | ----- | ----- | ----- | ----- |
| ApplicationVM1 | Running | 0 | 380 | 17:56:10 | Operating normally |
| ApplicationVM2 | Off | 0 | 0 | 00:00:00 | Operating normally |
| ApplicationVM3 | Running | 0 | 372 | 17:56:08 | Operating normally |
| ManagementVM | Running | 1 | 4096 | 10:19:03 | Operating normally |

- ApplicationVM2 Removed:



```
[TMER4R805S30]: PS C:\> get-vm
```

| Name | State | CPUUsage(%) | MemoryAssigned(M) | Uptime | Status |
|----------------|---------|-------------|-------------------|----------|--------------------|
| ApplicationVM1 | Running | 0 | 380 | 17:55:22 | Operating normally |
| ApplicationVM3 | Running | 0 | 372 | 17:55:20 | Operating normally |
| ManagementVM | Running | 2 | 4096 | 10:18:19 | Operating normally |

- ApplicationVM2 Restored and Restarted:

```
[TMER4R805S3]: PS C:\> get-vm
```

| Name | State | CPUUsage(%) | MemoryAssigned(M) | Uptime | Status |
|----------------|---------|-------------|-------------------|----------|--------------------|
| ApplicationVM1 | Running | 0 | 380 | 17:56:06 | Operating normally |
| ApplicationVM2 | Running | 2 | 512 | 00:00:01 | Operating normally |
| ApplicationVM3 | Running | 0 | 372 | 17:56:04 | Operating normally |
| ManagementVM | Running | 1 | 4096 | 10:19:03 | Operating normally |

3.7.4 Use HIT/Microsoft PowerShell Tools to perform a Selective Restore of a Hyper-V VM

The same previously shown selective restore operation using a Smart Copy of a collection of VMs can be accomplished using the EQLASMPSTool PowerShell command, **restore-asmsmartcopyselective**.

1. Identify the smart copies associated with the virtual machine collection (ProductionVMs) and select a smart copy to use for the restore operation. Store the object name of the selected Smart Copy in a variable.

```
[TMER4R805S30]: PS C:\> get-asmcollection | ? {$_.CollectionName -like  
"ProductionVMs"} | get-asmsmartcopy | ft -property SmartCopyName,ObjectName  
-AutoSize
```

| SmartCopyName | ObjectName |
|--------------------------------------|---|
| ProductionVMs (4/1/2014 11:34:13 AM) | SnapshotSet{99f91b84-dac0-4f54-bc62-8874078bc88e} |
| ProductionVMs (4/1/2014 2:08:09 PM) | SnapshotSet{3f4c93ea-2472-4cbb-a93d-5aec718df80b} |
| ProductionVMs (4/1/2014 2:09:27 PM) | SnapshotSet{87bcbec7-9894-4c50-a44d-402ffb5addc2} |
| ProductionVMs (4/1/2014 2:27:58 PM) | SnapshotSet{06522dec-d3df-4ccb-8b20-e42d9eb48fcc} |

```
[TMER4R805S30]: PS C:\> $scobjectname = "SnapshotSet{3f4c93ea-2472-4cbb-a93d-  
5aec718df80b}"
```

2. Get the object name of the virtual machine to be restored using the following **Get-ASMComponent** command and store in a variable.



```
[TMER4R805S30]: PS C:\> $vmobjectname = (Get-ASMComponent |  
? {$_.componentname -like "ApplicationVM2"}).objectname
```

```
[TMER4R805S30]: PS C:\> echo $vmobjectname
```

Sample output:

```
Component{4BA0BF8A-D08B-4347-B6EB-B810D8257A67}
```

3. Perform the selective restore of the virtual machine (ApplicationVM2) from the selected Smart Copy Snapshot of the ProductionVMs collection using the EQLASMPSTools **Restore-ASMSmartcopySelective** command.

```
[TMER4R805S30]: PS C:\> Restore-ASMSmartCopySelective -ObjectName  
"$scobjectname" -Selections "$vmobjectname" -ErrorAction Stop
```

With this command, the Smart Copy Object name is passed into the command through the **-objectname** parameter. The object name of the virtual machine is passed into the command through the **-selection** parameter

Note: A sample selective restore PowerShell script is included in Appendix A of this document.

3.7.5 Using ASM/ME to perform a Restore as New operation

The Restore as New operation is available for snapshots and clones. This operation creates a new VM that will be local and non-clustered, so no restrictions are needed. The VM is created locally on the current node, and has the default VM settings and a user-specified amount of RAM that uses the original VM .vhd files.

Note: For Gen 1 virtual machines, Restore as New supports up to four .vhd files during a restore. This limitation no longer exists with Windows 2012 R2 Gen 2 virtual machines

The Restore as New process allows the user to specify a name for the virtual machine, a drive letter mount point for the Smart Copy, and the amount of startup memory used by the new VM. Configuring other virtual hardware settings, such as virtual network adapters, is not available through this process. These settings will need to be configured manually for the new virtual machine. For Windows VMs belonging to an Active Directory domain, If the original VM and the newly restored VM will be online concurrently, be sure to take all necessary precautions on both the OS and running applications, like Microsoft SQL and Microsoft Exchange, in order to avoid duplicate names when bringing the new VM online while configuring virtual network settings.

Note: The restore as new operation in ASM/ME is detailed in the next section titled, "Create a VM template from an ASM/ME Smart Copy".



4 Create a VM template from an ASM/ME Smart Copy

Virtual machine templates enable the rapid deployment of virtual machines in the environment. Without a virtual machine template, an administrator would have to create all virtual machines from scratch. Even with scripting, creating a virtual machine from scratch every time quickly becomes an arduous task that slows down an administrator's ability to quickly respond to the dynamics of a virtualized environment.

Virtual Machine templates are essentially a sysprep image of an existing virtual machine. (See the Microsoft TechNet article at [http://technet.microsoft.com/en-us/library/cc721940\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc721940(v=ws.10).aspx) for details about sysprep). A sysprep image of a virtual machine has all of the Windows specific information removed including the computer name the unique computer security identifier (SID). This is important because network and application conflicts can result from having multiple machines with the same SID and same name on the same network. A sysprep image will retain all of the installed software and utilities on a virtual machine. This makes deployment of these installed applications and utilities throughout the environment quick and easy by using the image as a deployment template.

Hyper-V administrators often have multiple templates available for use. Some examples are:

- A Windows 8 client image for virtual desktop deployments
- A Windows Server 2012 application server image
- A Windows Server 2012 Core application server image
- A Windows Server 2012 R2 or patched version for test and acceptance trials
- A Linux OS image
- A Windows management image with installed tools and utilities

Creating and deploying Windows Hyper-V virtual machine templates is easily done when using Microsoft System Center Virtual Machine Manager (SCVMM). Without SCVMM, this task is a little more complicated. Fortunately, an administrator can recreate much of the template functionality in SCVMM using Dell EqualLogic ASM/ME, standard Windows Server Hyper-V Manager, and some basic PowerShell scripting. This section shows how to create a Hyper-V virtual machine template based off of the Management VM (set up in Section 3) using these tools. The template creation process shown in this section is illustrated in Figure 5.



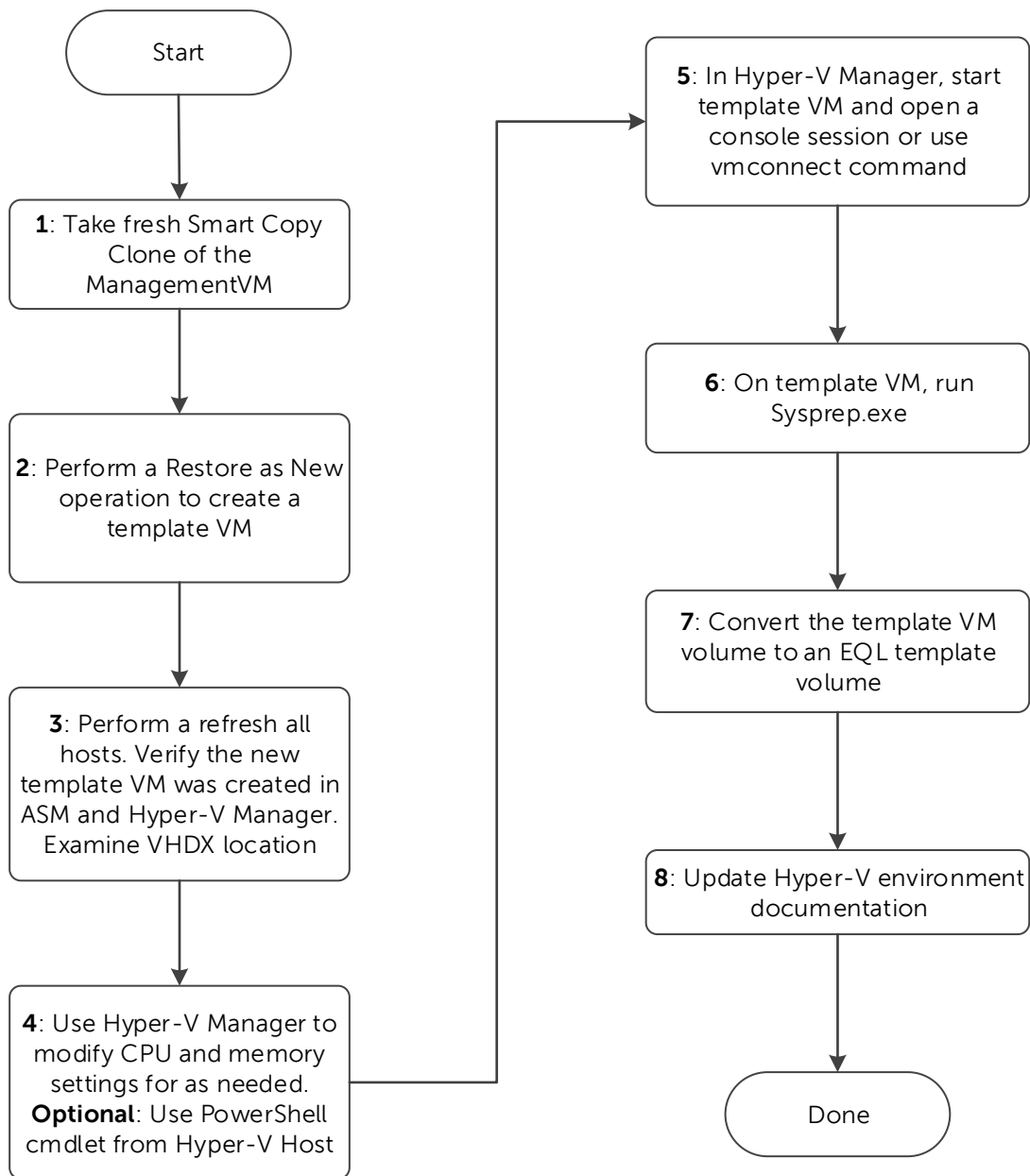


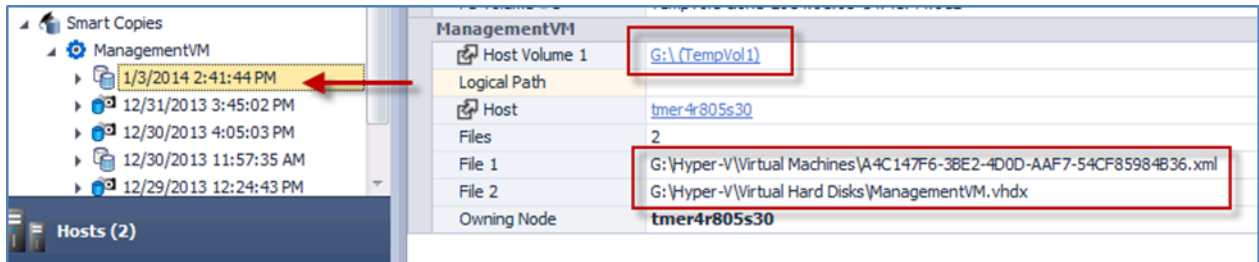
Figure 5 Template creation process

Note: For this example, the management virtual machine (ManagementVM) will be used as the basis for the template. A Smart Copy Clone of the ManagementVM will be created and used. However, a Smart Copy Snapshot can also be used.

4.1 Using ASM/ME, take a Smart Copy of the VM for the template

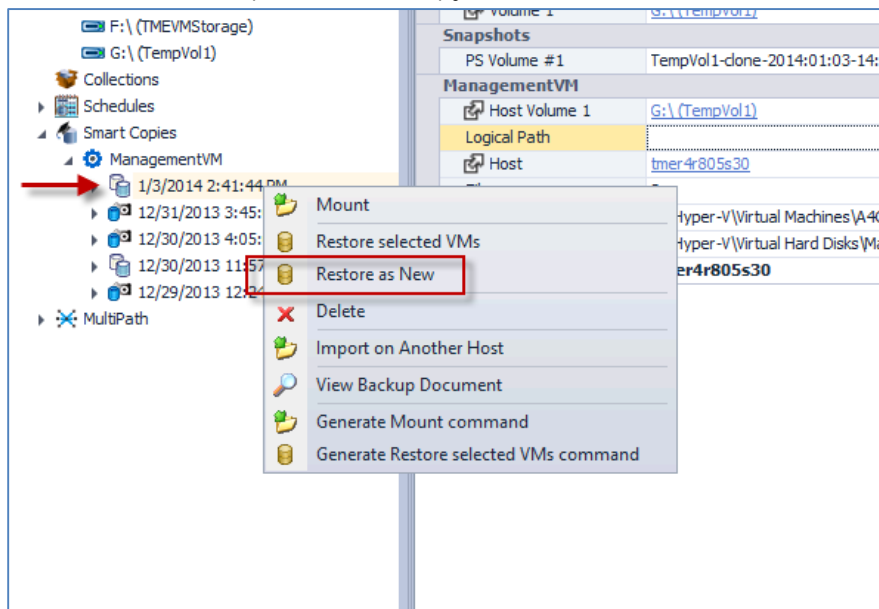
If not done previously, create a fresh Smart Copy of a Virtual Machine that will be the basis for the template. In the example, a Smart Copy clone of the ManagementVM is used. Review Section 3.3.4 of this document or consult the *Dell EqualLogic Auto-Snapshot Manager/Microsoft Edition User Guide* in the document directory for the EqualLogic HIT kit installation (c:\Program Files\EqualLogic\Docs) for details on how to do this.

Note: A Smart Copy Snapshot of the virtual machine can also be used for this operation.

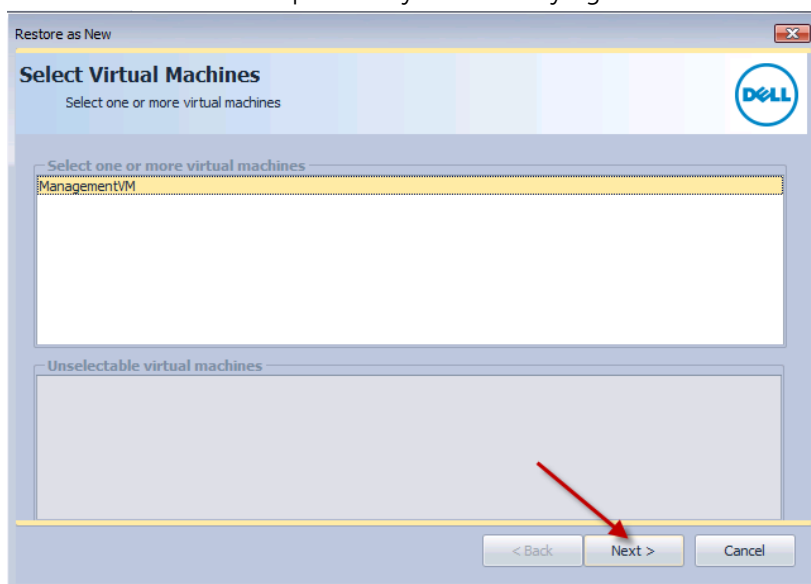


4.2 Use ASM/ME to perform a Restore as New operation to create a new template VM

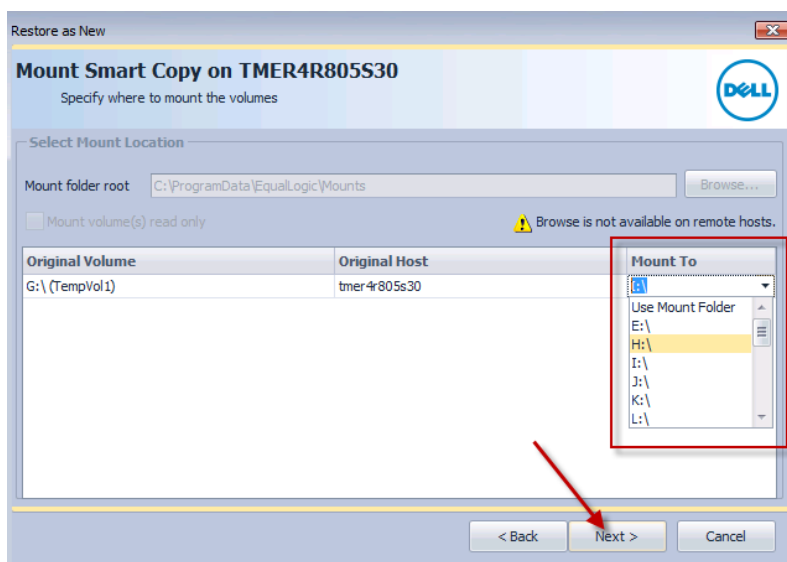
1. Use ASM/ME to perform a Restore as New operation. Select a Smart Copy (Clone or Snapshot) of the virtual machine that will be the basis of the template. Right-click the Smart Copy and select **Restore as New**. In the example, a Smart Copy clone of the VM is selected.



- Unlike the previous restore operations that are a restore of an existing virtual machine, the Restore as New operation restores a Smart Copy clone or snapshot of the virtual machine as a new virtual machine. This is accomplished by first identifying the virtual machine to be restored.



- Identify a mounting technique for the smart copy.
Restore as New from a Smart Copy clone utilizes the entire clone volume and mounts it as a new drive on the Hyper-V host.
- Select a drive letter for the mount point in the wizard, and then click **Next**.
In this example, H:\ is selected as the mount point for the Smart Copy clone.

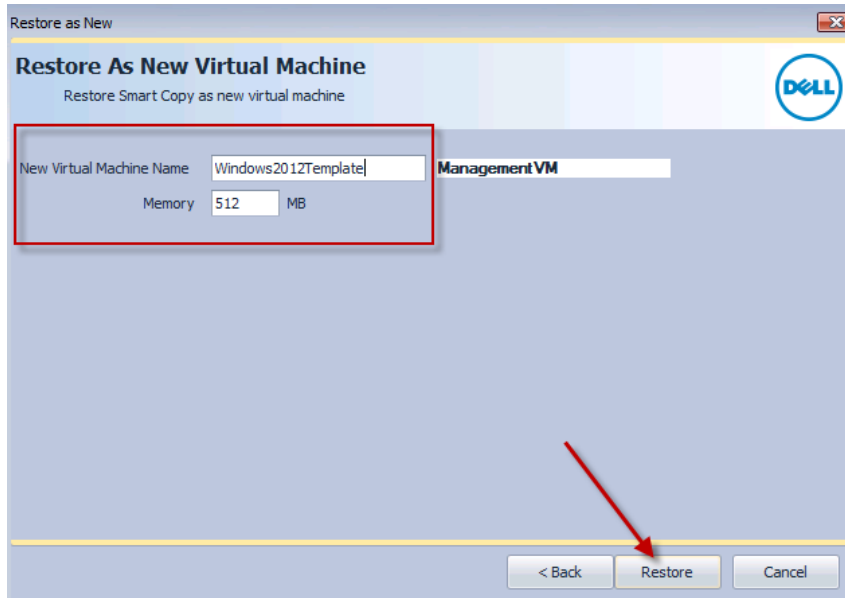


- Enter a name for the new virtual machine and the start memory size. In the example, the new virtual machine will be called "Windows2012Template" and the startup memory size will be 512 MB.

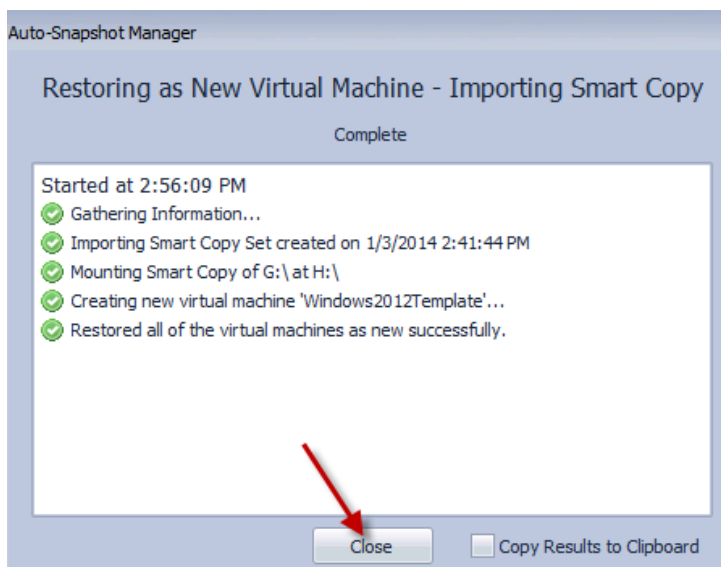


512 MB is the minimum amount of memory a Hyper-V virtual machine needs to start up. Since this virtual machine is a template, a minimum value is selected. This value can be adjusted later as necessary.

6. Select **Restore** to start the Restore as New process.



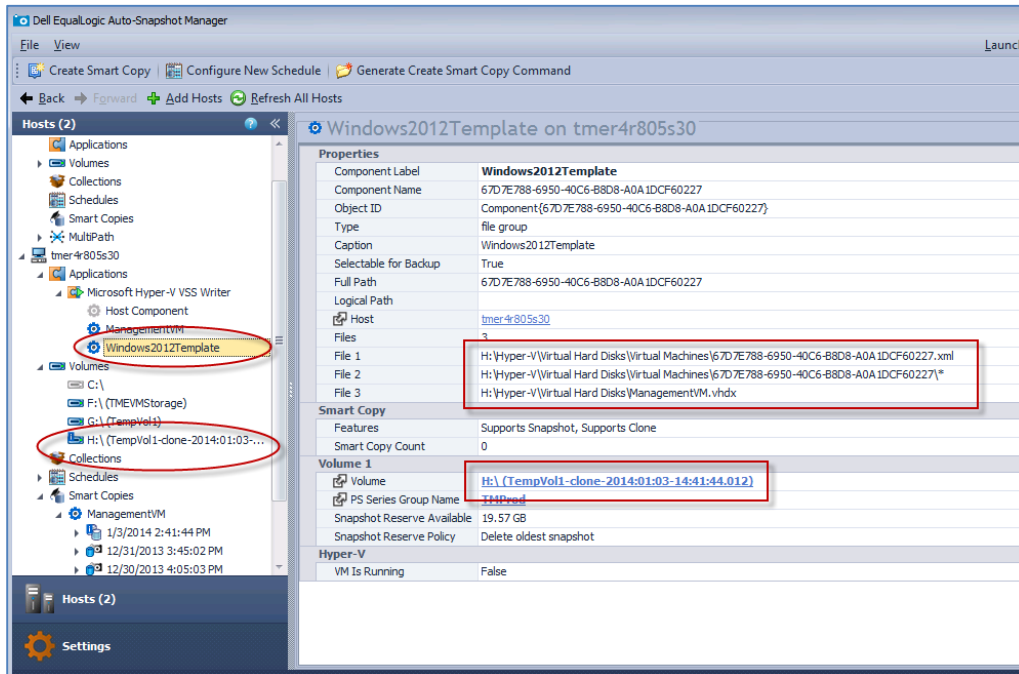
7. Monitor the progress of the Restore as New process. It will mount the clone volume on the Hyper-V host and create a new virtual machine that will act as a template. Once the operation completes, select **Close**



4.3 Verify that the new template VM has been created

After the Restore as New operation completes, the new template virtual machine (Windows2012Template) is displayed in the navigation tree for the Hyper-V VSS Writer in ASM/ME. Also, the mounted clone volume (H:\) is displayed under volumes.

1. Select the new template virtual machine and examine its properties in the properties window.



Note: Other than the mount point, the properties for the new virtual machine template are identical to the property information for the management VM itself. This includes the name of the mounted file system label and the name of the virtual hard disk (ManagementVM.VHDX).

2. For easier identification, rename the virtual hard disk file to match the name of the new template virtual machine. This can be done in Hyper-V manager or with PowerShell. A sample VHDX rename PowerShell script is shown below.

```
#####
#Enter Name for new VM
$VM = "Windows2012Template" #Hard coding template name. Could use user input.

Get-VMHardDiskDrive -vmname $vm | % {
    $oldvhdName = $_.Path
    $newvhdName = $oldvhdName.Substring(0,$oldvhdName.LastIndexOf("\")) `
        + "\" + $_.VMName + "_" + $_.ControllerType `
        + "_" + $_.ControllerNumber + "_" `
        + $_.ControllerLocation`
}
```



```

        + $oldvhdname.Substring($oldvhdname.LastIndexOf("."))
Rename-Item $oldvhdname $newvhdname
Set-VMHardDiskDrive -VMName $_.VMName -Path $newvhdname `
    -ControllerType $_.ControllerType -ControllerNumber `
    $_.ControllerNumber -ControllerLocation `
    $_.ControllerLocation
}
#End of Script
#####

```

This script renames the VHD to include the name of the new template virtual machine plus the controller number and controller location.

3. To verify the change, run the following PowerShell command.

```
[TMER4R805S30]: PS H:\> Get-VMHardDiskDrive -VMName $vm | ft -property VMNAME,Path -AutoSize
```

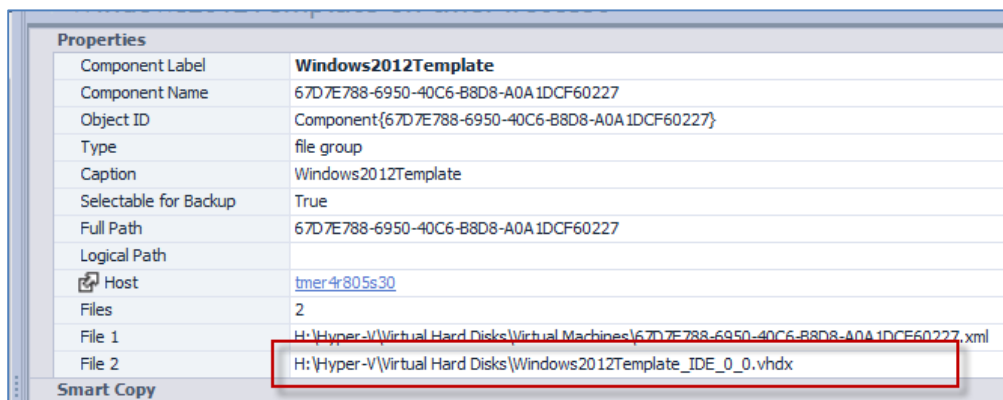
```

VMName          Path
-----
Windows2012Template H:\Hyper-V\Virtual Hard Disks\Windows2012Template_IDE_0_0.vhdx

```

4. After running the rename script, the new name of the VHD will be present in the template virtual machine property window in ASM/ME. Examine the properties window to verify that ASM/ME is properly displaying the new name of the VHD.

Note: select **Refresh all hosts** if the change is not immediately present in the properties window



On the Hyper-V host, the mount point for the template clone volume is the same as the volume that houses the Management VM as shown below in the **get-volume** command output.

```
[TMER4R805S30]: PS C:\> get-volume | ft -AutoSize
```

```

DriveLetter FileSystemLabel FileSystem DriveType HealthStatus SizeRemaining      Size
-----

```



| DriveLetter | FileSystemLabel | FileSystem | DriveType | HealthStatus | SizeRemaining | Size |
|-------------|-----------------|------------|-----------|--------------|---------------|-----------|
| | System Reserved | NTFS | Fixed | Healthy | 108.69 MB | 350 MB |
| F | TMEVMStorage | NTFS | Fixed | Healthy | 491.13 GB | 499.88 GB |
| G | TempVol1 | NTFS | Fixed | Healthy | 88.53 GB | 99.88 GB |
| H | TempVol1 | NTFS | Fixed | Healthy | 88.53 GB | 99.88 GB |
| C | | NTFS | Fixed | Healthy | 45.61 GB | 67.41 GB |
| D | | | CD-ROM | Healthy | 0 B | 0 B |

- To further minimize potential confusion, rename the file system labels to reflect the name of the virtual machines that use them. This can be done with the following **set-volume** commands.

```
[TMER4R805S30] PS C:\>set-volume -DriveLetter H -NewFileSystemLabel
WinServ2012Temp
```

```
[TMER4R805S30]: PS C:\> set-volume -driveletter G -NewFileSystemLabel
MgmtVMVol
```

- Run the **get-volume** command again to verify that the changes occurred.

```
[TMER4R805S30]: PS C:\> get-volume | ft -AutoSize
```

| DriveLetter | FileSystemLabel | FileSystem | DriveType | HealthStatus | SizeRemaining | Size |
|-------------|-----------------|------------|-----------|--------------|---------------|-----------|
| | System Reserved | NTFS | Fixed | Healthy | 108.69 MB | 350 MB |
| F | TMEVMStorage | NTFS | Fixed | Healthy | 499.73 GB | 499.88 GB |
| G | MgmtVMVol | NTFS | Fixed | Healthy | 88.53 GB | 99.88 GB |
| H | WinServ2012Temp | NTFS | Fixed | Healthy | 88.53 GB | 99.88 GB |
| C | | NTFS | Fixed | Healthy | 45.63 GB | 67.41 GB |
| D | | | CD-ROM | Healthy | 0 B | 0 B |

- This next step is optional but it is always a good idea to rename the EqualLogic clone volume so that it can be easily identified as the volume associated with the template virtual machine. This is easily done through Group Manager or by using the EqualLogic HIT PowerShell utility. In this example, the name of clone volume was TempVol1-Clone-2014*. Using the HIT PowerShell utility, a one liner can be created that uses the **get-eqlvolume** command and passes the output (the clone volume) into the **set-eqlvolume** command to rename it to the desired name (such as **WinServ2012Template**).

```
[TMER4R805S30]: PS C:\>get-eqlvolume | ? {$_.volumename -like "*tempvol1-
clone*"} | Set-EqlVolume -NewVolumeName "WinServ2012Template" -iSCSIalias
"WinServ2012Template"
```

```
Volume 'WinServ2012Template' changed successfully
```

```
[TMER4R805S30]: PS C:\> get-eqlvolume | ft -Property Volumename
```

```
Volumename
-----
...
TempVol1
```



```
WinServ2012Template
```

8. Optionally, rename the Management VM EQL volume (TempVol1) at this point as well for easier identification.

```
[TMER4R805S30]: PS C:\> Set-EqlVolume -volumename TempVol1 -NewVolumeName  
"MgmtVMVol" -iSCSIalias "MgmtVMVol"
```

```
[TMER4R805S30]: PS C:\> get-eqlvolume | ft -Property Volumename
```

```
Volumename
```

```
-----
```

```
...
```

```
MgmtVMVol
```

```
WinServ2012Template
```

4.4 Modify the new template VM settings as necessary

The settings for the virtual machine template can be modified as needed. In the example, the template virtual machine was created with 512MB of non-dynamic startup memory and a single virtual processor. Because the template virtual machine is off, these settings can be modified at this point in the **Settings** window in Hyper-V Manager or through PowerShell. For example, to change the virtual machine template settings in PowerShell to use two virtual processors and dynamic memory that can float in a range between 512MB to 8GB, use the following PowerShell command sequence.

1. Confirm that the template virtual machine is in the off state.

```
[TMER4R805S30]: PS C:\> get-vm
```

| Name | State | CPUUsage(%) | MemoryAssigned(M) | Uptime | Status |
|---------------------|---------|-------------|-------------------|------------|--------------------|
| ---- | ---- | ----- | ----- | ----- | ----- |
| ManagementVM | Running | 5 | 4000 | 3.21:46:34 | Operating normally |
| Windows2012Template | Off | 0 | 0 | 00:00:00 | Operating normally |

2. Change the template virtual machine settings with the **set-vm** command.

```
[TMER4R805S30]: PS C:\> Set-VM -name Windows2012Template -ProcessorCount 2  
-DynamicMemory -MemoryMinimumBytes 512MB -MemoryMaximumBytes 8GB
```

3. Re-examine the template virtual machine with the **get-vm** command.

```
[TMER4R805S30]: PS C:\> Get-VM -Name Windows2012Template | fl *
```

```
...
```

```
MemoryStartup : 536870912
```

```
DynamicMemoryEnabled : True
```



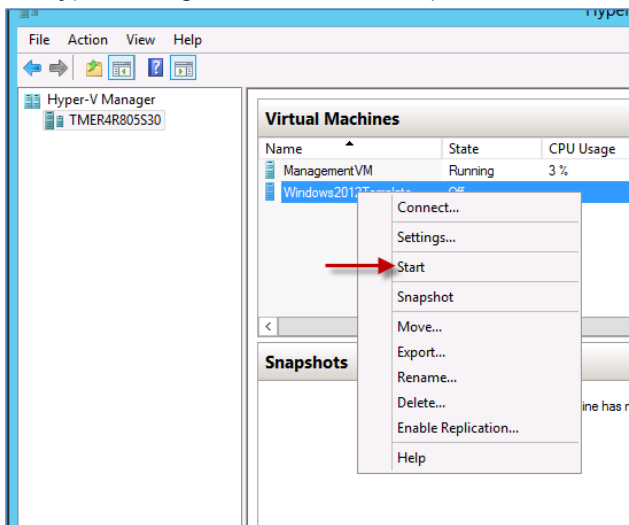
```
MemoryMinimum           : 536870912
MemoryMaximum           : 8589934592
ProcessorCount           : 2
...
```

Note: When the template virtual machine was created from the cloned source virtual machine, it was created without vNICs. Any time a virtual machine is created through an ASM/ME Restore as New operation, the new virtual machine will not have vNICs, even if the source virtual machine has vNICs created. This is done intentionally by ASM/ME to reduce the potential for IP address conflicts in the virtual environment. A best practice is to leave the template without vNICs and have the vNIC creation as part of a virtual machine creation script.

4.5 Startup the template VM using Hyper-V Manager or PowerShell

- Once the template virtual machine settings have been set and verified, start it in either Hyper-V manager or PowerShell.

- Hyper-V: Right-click on the template virtual machine and select **Start**.



- PowerShell: Use the **start-vm** command.

```
[TMER4R805S30]: PS C:\> start-vm Windows2012Template
```

- Verify that the template virtual machine started successfully.

```
[TMER4R805S30]: PS C:\> get-vm
```

| Name | State | CPUUsage(%) | MemoryAssigned(M) | Uptime | Status |
|---------------------|---------|-------------|-------------------|------------|--------------------|
| ManagementVM | Running | 9 | 4000 | 3.22:52:22 | Operating normally |
| Windows2012Template | Running | 8 | 512 | 00:00:07 | Operating normally |



Once the template virtual machine has been started, open a connection with Hyper-V Manager from the management virtual machine.

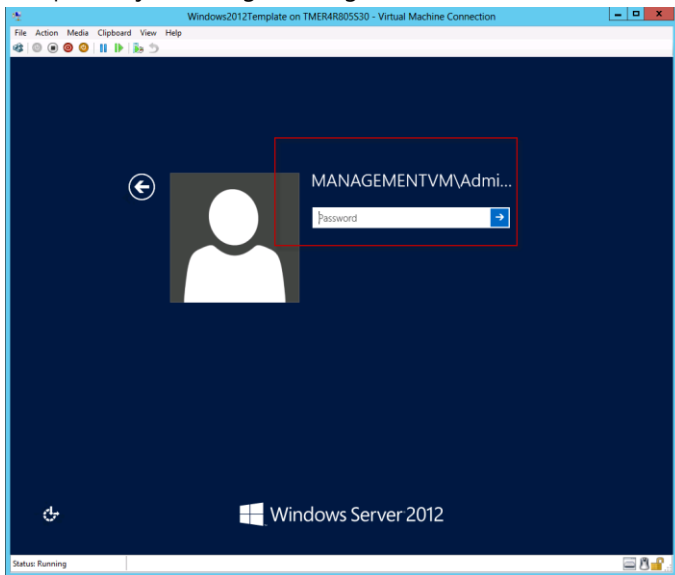
3. Right-click the template virtual machine and select **Connect** in the pop-up menu, or use the `vmconnect` command line utility:

The syntax of the `vmconnect` utility is: `vmconnect <Hyper-V Host Server Name or IP> <VM Name>`

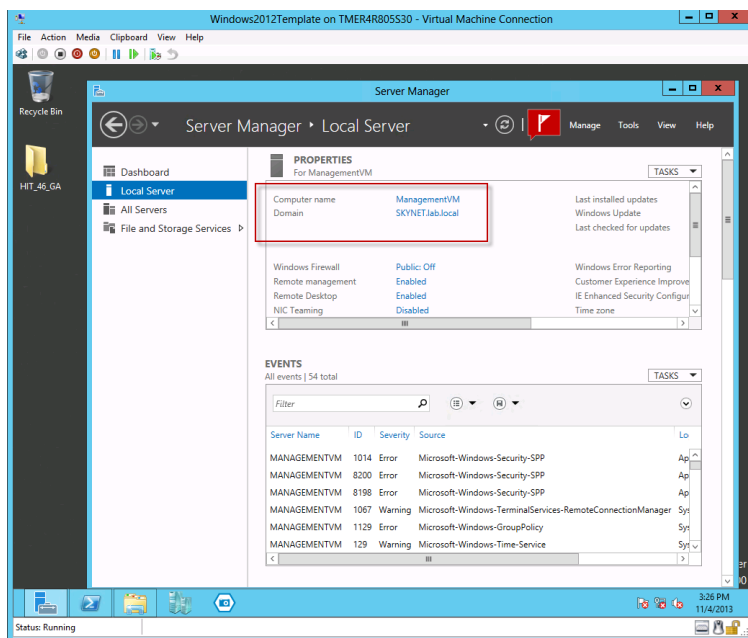
```
[MANAGEMENTVM] PS C:\>vmconnect TMER4R805S30 Windows2012Template
```

Note: The installation of the RSAT-Hyper-V-Tools includes the `vmconnect` utility. It can be also be used to open a connection window to a virtual machine on the Hyper-V host from a command line or script.

4. Once the connection to the template virtual machine is established, sign in by pressing Ctrl + Alt + Delete from the action menu on the sign in screen. This leads to the Windows Server login screen. Note that the name of the template virtual machine still reflects the name ManagementVM. The template virtual machine is also still using the ManagementVM administrator credentials. Log in to the template by entering the original source virtual machine (ManagementVM) administrator password.



Note that the hostname displayed in server manager still reflects the name of the source virtual machine (ManagementVM).



4.6 Sysprep and reboot the template virtual machine

Because the template is still using the computer name, user credentials, and computer security identifier from the source virtual machine (ManagementVM), it needs to be syspreped so these values can be reset. This allows the template to be used for creating future virtual machines which will not result in networking and application conflicts within the Hyper-V environment. Syspreping is performed by using the Sysprep utility found in the `c:\windows\system32\sysprep` directory.

1. In the connection window for the template, open a command line and go to the sysprep directory.

```
C:\> cd \windows\system32\sysprep
```

In the directory, list the contents and verify that the `sysprep.exe` utility and any answer files (such as `autounattend.xml`) that were placed into the directory on the example management virtual machine in Section 3.7.4 are present.

```
C:\windows\system32\sysprep> ls
```

```
Directory: C:\windows\system32\sysprep
```

| Mode | LastWriteTime | Length | Name |
|-------|---------------------|--------|------------------|
| d---- | 10/15/2013 12:04 PM | | Panther |
| -a--- | 9/25/2013 3:16 PM | 7050 | autounattend.xml |
| -a--- | 7/25/2012 11:08 PM | 407040 | sysprep.exe |
| -a--- | 7/25/2012 11:07 PM | 970240 | unbcl.dll |



2. To start the sysprep operation for the template virtual machine, run the following sysprep command with the included options:

```
C:\windows\system32\sysprep> ./sysprep.exe -oobe -generalize -reboot  
-unattend: ./autounattend.xml
```

During the sysprep operation, the options included in the above command do the following:

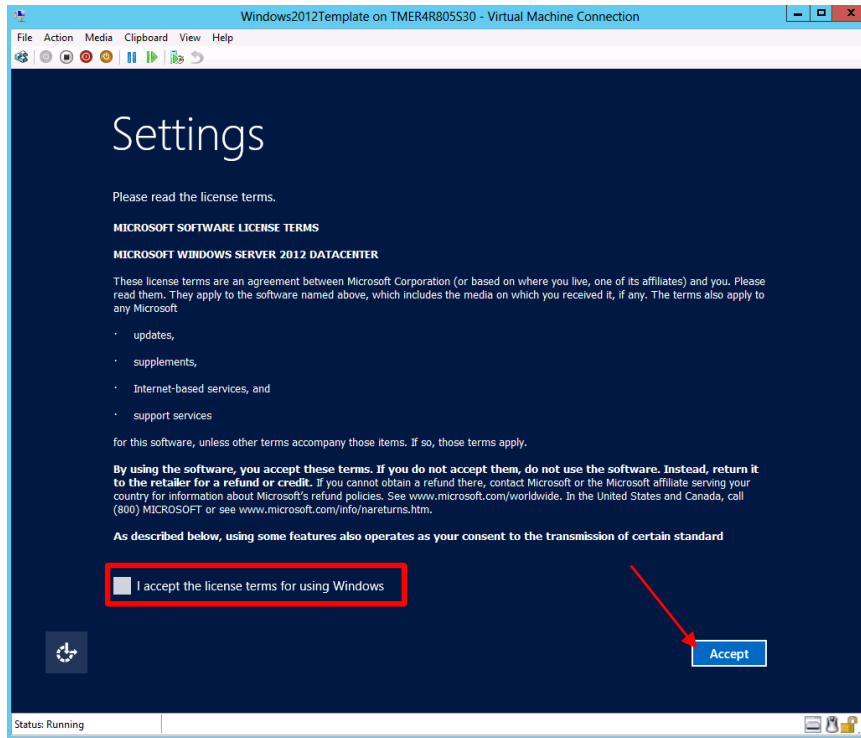
- **-oobe**: Out of the Box Experience – this enables end users to customize their windows operating system, create user accounts, and name the computer at the initial startup after reboot
- **-generalize**: Prepares the Windows installation to be imaged. Sysprep removes all unique system information from the Windows installation. Sysprep resets the security ID (SID), clears any system restore points, and deletes event logs
- **-reboot**: Restarts the computer after the sysprep operation completes - use this option to audit the computer and to verify that the first-run experience operates correctly
- **-unattend:<answer file>** - Applies settings in an answer file to Windows during an unattended installation, where <answerfile> specifies the path and file name of the answer file to use.

For more information on how to use sysprep, go to Microsoft TechNet at

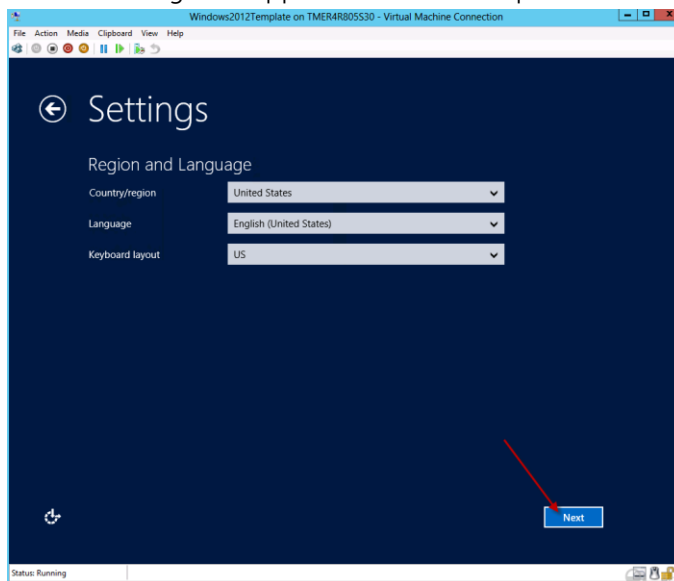
<http://technet.microsoft.com/en-us/library/hh825033.aspx>

3. After the sysprep command is complete, a status window opens and provides details on the progress of the operation until the template reboots. After the sysprep and reboot complete, re-open a connection window to the template virtual machine. The Settings Screen (License Terms and Agreement) should be visible depending on the flags and answer file used with the sysprep command. This is the first screen a user will see for any cloned virtual machines that used the template. If the answer file used in the template sysprep operation included a default administrator password and license keys, then all a user needs to log in to the template is check **I accept the license terms for using Windows** and then click **Accept**.

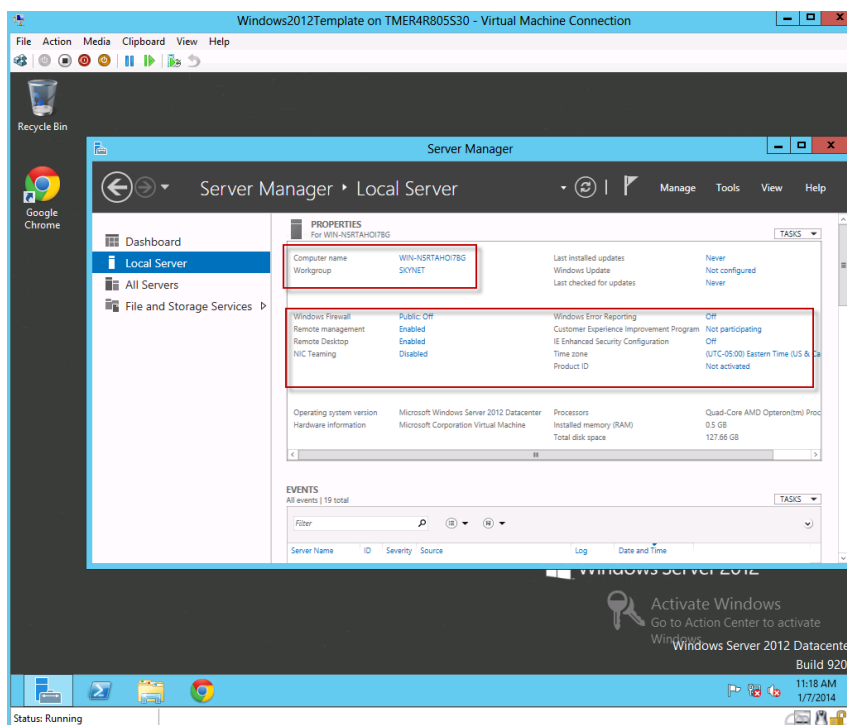




4. The default Region and Language options (which could also be specified in the answer file along with time zone, remote desktop, and firewall settings) are displayed. Click **Next** to log in to the new virtual machine using the supplied administrator password.



5. Review the template properties in Server Manager to verify that the template has been re-assigned a default computer name and the settings specified in the answer file have been applied.



6. After verifying the settings, shut down the template either through Hyper-V manager on the management virtual machine or through a remote PowerShell connection the Hyper-V server. Using PowerShell:

```
[TMER4R805S30]: PS H:\> stop-vm -vmname Windows2012Template
```

```
[TMER4R805S30]: PS H:\> get-vm
```

| Name | State | CPUUsage (%) | MemoryAssigned (M) | Uptime | Status |
|---------------------|---------|--------------|--------------------|------------|--------------------|
| ManagementVM | Running | 5 | 4000 | 7.00:40:17 | Operating normally |
| Windows2012Template | Off | 0 | 0 | 00:00:00 | Operating normally |

4.7 Convert the template VM volume to an EqualLogic "Template Volume"

The final step in creating the virtual machine template is to turn the EqualLogic volume associated with the template VM into an EqualLogic template volume. EqualLogic template volumes offer several benefits when used in a Hyper-V environment. A template volume is a type of volume that is useful when the environment requires multiple volumes that share a large amount of common data. This happens when virtual machines are cloned from an original template VM. In this case, the new virtual machines often



contain a small set of unique data that is different from the original data on the template vm. The bulk of the data between the template and the new virtual machines is the same. An EqualLogic template volume works well in this situation because if the new virtual machines are used to separate unique volumes cloned from the original template, these cloned volumes will be thin clones.

Thin clone volumes only store the deltas while common data is stored on the original template volume. This makes thin clones very efficient with regards to pool space utilization. Initially, a template volume and thin clone are identical in reported size and content. Because the PS Group allocates space to the new thin clone in the same way it allocates space to a new standard, thin provisioned volume, only the minimum volume reserve is consumed from free pool space.

Thin Clone volumes are ideal candidates for use cases such as Windows virtual desktop deployments and testing validation database deployments where available pool space comes at a premium. For more information about thin clone volumes, see the *Dell EqualLogic Group Manager Administrator's Manual*.

Note: Be sure that the template virtual machine is finalized (settings and configurations) prior to converting its volume to a template.

1. To convert the template virtual machine volume (WinServ2012Template) to a template volume, use the **ConvertTo-EqlTemplateVolume** command from the EqualLogic PowerShell tools. If necessary, first import the EqualLogic PowerShell module.

```
[TMER4R805S30]: PS C:\> import-module -name "c:\program  
files\equallogic\bin\EqlPSTools.dll"
```

2. Convert the volume (WinServ2012Template) to a template with the following command.

```
[TMER4R805S30]: PS C:\> ConvertTo-EqlTemplateVolume -GroupName TMProd -  
VolumeName WinServ2012Template
```

3. Converting to a template volume makes the original volume read only and takes it offline. This is done to protect any shared data on the template volume. However, because the template volume is offline, the VMMS service on the Hyper-V host is no longer able to access the template virtual machine. This often results in the template virtual machine being in the **Off-Critical** state in Hyper-V Manager. To clear this issue, bring the template volume back online so that it is visible to the Hyper-V host but still read only.

Note: It is often desirable that the template volume be online and read only accessible to the Hyper-V host so that the contents such as the VHDX files can still be read and copied to other locations.

4. To set the template volume back online and visible to the Hyper-V host for read only access, use the **set-EQLVolume** command.

```
[TMER4R805S30]: PS C:\> Set-EqlVolume -VolumeName WinServ2012Template -  
OnlineStatus online
```

5. Reconnect the template volume iSCSI target address to the Hyper-V host. This can be done using the following PowerShell command one liner:



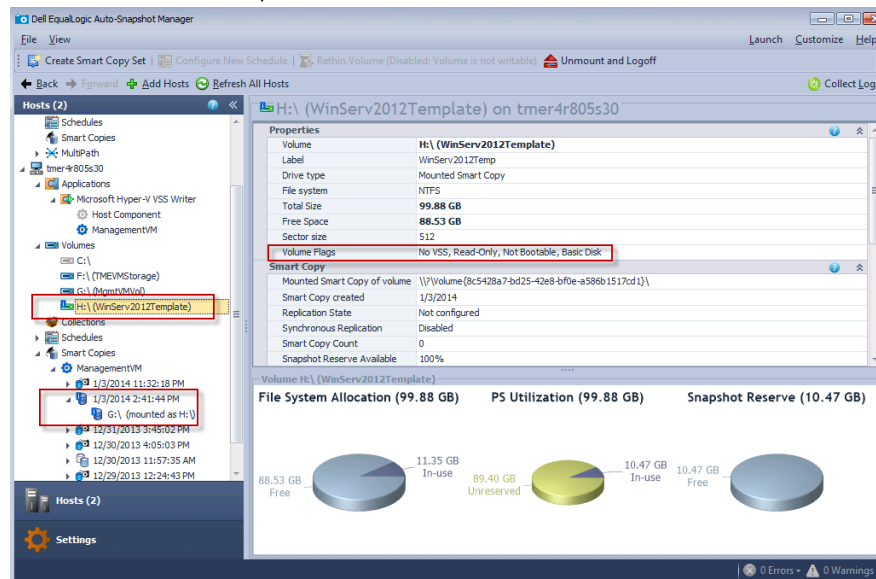
```
[TMER4R805S30]: PS C:\> connect-iscsitarget -NodeAddress (Get-EqlVolume -
VolumeName WinServ2012Template).iscsitargetname
```



- Verify that the template volume is again visible to the Hyper-V host.

```
[TMER4R805S30]: PS C:\> get-volume | ft -AutoSize
```

| DriveLetter | FileSystemLabel | FileSystem | DriveType | HealthStatus | SizeRemaining | Size |
|-------------|-----------------|------------|-----------|--------------|---------------|----------|
| H | WinServ2012Temp | NTFS | Fixed | Healthy | 88.53 GB | 99.88 GB |

- Re-examine the template volume in ASM/ME.



Examining the properties of the Template Volume (WinServ2012Template) reveals that its icon in the navigation tree has changed to  signifying that it is a read only volume. This is also stated in the volume flags properties in the property window. The associated smart copy clone that was used for the template virtual machine volume has also changed its icon to  signifying that it is now a template volume.

At this point, the template volume is ready to be used for rapid virtual machine deployments.

Note: A sample Rapid VM Deployment PowerShell script which uses EqualLogic template volumes is detailed in [Appendix B](#) of this document.

4.8 Updating Hyper-V environment documentation

As with any configuration change to the Hyper-V environment, best practices dictate that the documentation is updated to reflect the change. In the case of the example used in this document, the



new template volume (H:\) with the template virtual machine VHDX (no longer visible to Hyper-V VMMS) is added to any Hyper-V environment schematics. A sample diagram detailing this is shown in Figure 6.

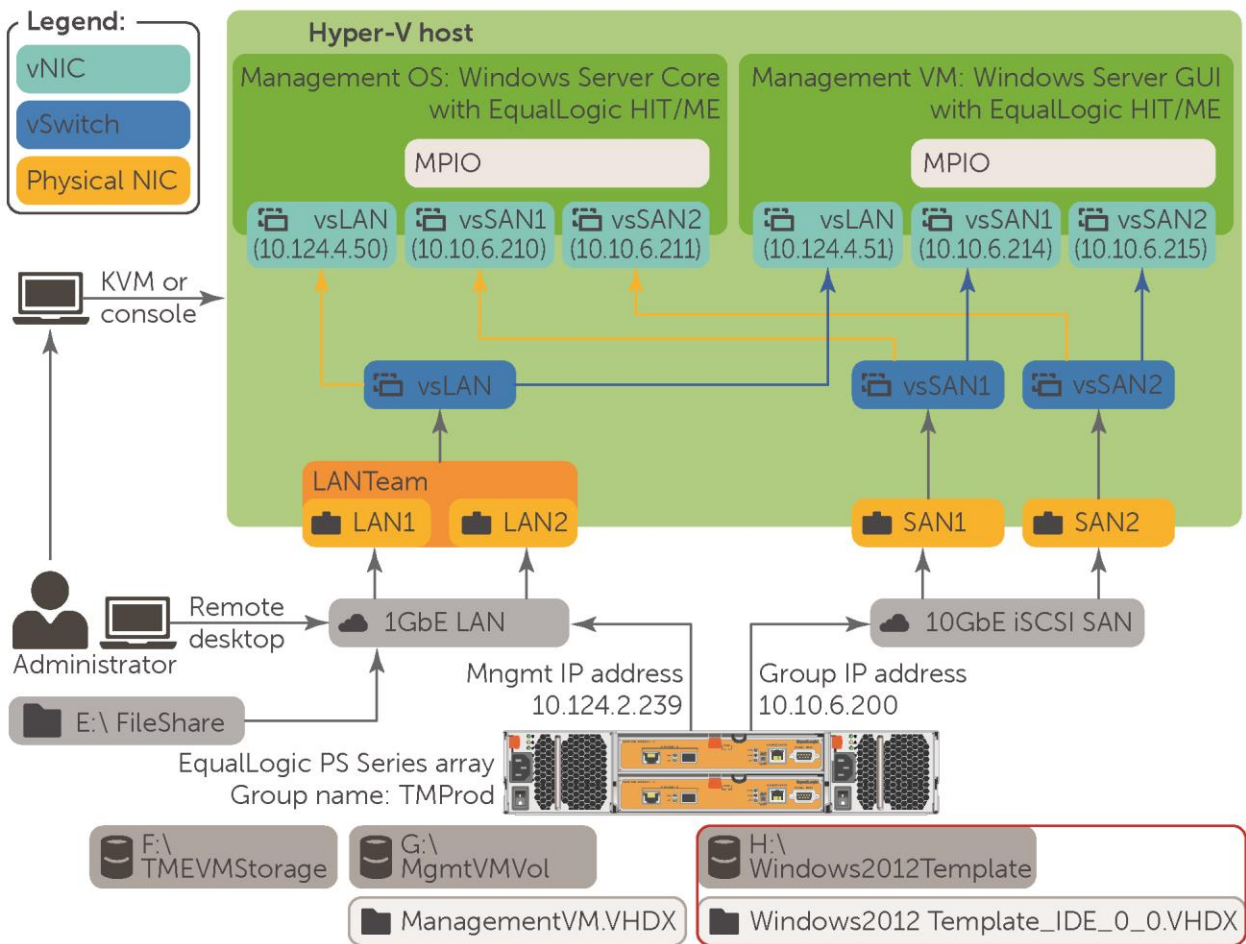


Figure 6 Hyper-V environment schematics

It is also a good idea to start keeping track of the EqualLogic volumes that house the Hyper-V environments virtual machine VHD files. Keeping track of the VHD locations will become especially important as virtual machines start proliferating. A good way to do this is to add a VHD Volume column in environment configuration documents.

Table 2 VHD Volume info in a sample environment configuration document

| Hyper-V Host | Component | VHD Volume | vNIC Name | Attached Virtual Switch | IP Address | Gateway | Subnet Mask (Prefix – address) | DNS Servers |
|--------------|---------------|------------|-----------|-------------------------|-------------|------------|--------------------------------|-----------------------|
| TMER4R805S30 | Management OS | NA | vsLAN | vsLAN | 10.124.4.50 | 10.124.4.1 | 22 – 255.255.252.0 | 10.124.6.245, 8.8.8.8 |
| | | | vsSAN1 | vsSAN1 | 10.10.6.210 | NA | 16 – 255.255.0.0 | NA |
| | | | vsSAN2 | vsSAN2 | 10.10.6.211 | NA | 16 – 255.255.0.0 | NA |



| Hyper-V Host | Component | VHD Volume | vNIC Name | Attached Virtual Switch | IP Address | Gateway | Subnet Mask (Prefix – address) | DNS Servers |
|--------------|---------------------|------------------------------|-----------|-------------------------|-------------|------------|--------------------------------|--------------------------|
| | ManagementVM | G:\ (MgmtVMVol) | vsLAN | vsLAN | 10.124.4.51 | 10.124.4.1 | 22 – 255.255.252.0 | 10.124.6.245, 8.8.8.8 |
| | | | vsSAN1 | vsSAN1 | 10.10.6.214 | NA | 16 – 255.255.0.0 | NA |
| | | | vsSAN2 | vsSAN2 | 10.10.6.215 | NA | 16 – 255.255.0.0 | NA |
| | Windows2012Template | H:\ (WinServ2012Template) | NA | NA | NA | NA | NA | NA |



5 Summary

This document has presented an in-depth overview on how to use Dell EqualLogic PS Series storage and HIT/Microsoft to manage and protect a Microsoft Hyper-V virtualized environment. Detailed examples of using the ASM/ME and PowerShell utilities have also been shown. In this document, the reader has gained an understanding of the capabilities available from using Dell EqualLogic HIT/Microsoft, so that standard operation procedures can be created and implemented based on the information provided in this guide.

The remainder of this document provides code samples as references when managing and protecting a Windows Server Hyper-V environment using Dell Equallogic PS Series storage arrays and tools.



A Sample Hyper-V VM Selective Restore script using HIT/Microsoft PowerShell tools

```
<#
.SYNOPSIS
Performs a selective virtual machine restore from an ASM Smartcopy of a Collection of
virtual machines or from Smart Copy of an individual VM.

.DESCRIPTION
This script will perform a selective Hyper-V virtual machine restore using
an ASM Smartcopy of the individual VM or using a Collection which includes the VM.
It uses the new EQLASMPSTools provided by HIT/Microsoft v4.7

.EXAMPLE OUTPUT

Name                State   CPUUsage(%) MemoryAssigned(M) Uptime   Status
-----
ApplicationVM1      Running 0         1000              00:40:46 Operating normally
ApplicationVM2      Running 0         512               00:01:13 Operating normally
ManagementVM        Running 0         4096              04:19:37 Operating normally

Select a VM to restore: ApplicationVM2
The name of the VM is ApplicationVM2

VMName : ApplicationVM2
VMId    : 4ba0bf8a-d08b-4347-b6eb-b810d8257a67
State   : Running
Path    : G:\Hyper-V

VM ApplicationVM2 has the following SmartCopies
Individual VM Smartcopies...
SmartCopyName                OriginalObjectId
ObjectName                    -----
-----
ApplicationVM2 (3/24/2014 3:57:57 PM) Component{4BA0BF8A-D08B-4347-B6EB-B810D8257A67}
SnapshotSet{80459c67-6a99-494d-997d-833d156fc38f}

Smartcopy Collections...
SmartCopyName                OriginalObjectId      ObjectName
-----
VM Collection (3/24/2014 3:44:01 PM) Collection{VM Collection} SnapshotSet{9ffb1164-4168-
4378-9e1a-24e4006b61e0}

Please enter a smartcopy ObjectName to use for the restore (use copy and paste):
SnapshotSet{9ffb1164-4168-4378-9e1a-24e4006b61e0}

objectname = SnapshotSet{9ffb1164-4168-4378-9e1a-24e4006b61e0}
```



```
ComponentId = Component{4BA0BF8A-D08B-4347-B6EB-B810D8257A67}

Restoring ApplicationVM2 from SnapshotSet{9ffb1164-4168-4378-9e1a-24e4006b61e0}. Please wait...
>>>>>>>>>>>>>>>>>>>
ApplicationVM2 restored and online

Create New Smartcopy for ApplicationVM2 ? (YES/NO): Yes
Creating new Smartcopy for ApplicationVM2

Hit any key to continue:

SmartCopyName      : ApplicationVM2 (3/24/2014 4:27:42 PM)
SmartCopyType      : Snapshot
Created            : 3/24/2014 4:27:42 PM
OriginatingComputer : R3U32R710.SKYNET.lab.local
OriginalObjectName : ApplicationVM2
isAccessible       : True
ObjectName         : SnapshotSet{a92d6500-933e-4f69-894a-60c0a596a0a2}
OriginalObjectId   : Component{4BA0BF8A-D08B-4347-B6EB-B810D8257A67}

.NOTES
This script must be run with administrator privileges.
This script requires HIT/Microsoft v4.7 or greater
This script has been written to use Smart Copies of collections and individual VMs.
It could be modified to work with CSV smart copies
#>

#####
#Import required modules

cls
Set-StrictMode -Version Latest
$ErrorActionPreference = "Stop"

Import-Module -Name "c:\program files\equallogic\bin\EqlASMPSTools.dll"
Import-Module -Name "c:\program files\equallogic\bin\EqlPSTools.dll"

#####
#Check HIT/Microsoft Verison

if ((Get-ASMVersion).version -lt "4.7") { echo "HIT/Microsoft 4.7 or greater
required.`nLoad HIT/Microsoft 4.7 and rerun script."; break}

#####
#Select a virtual machine to restore

get-vm | ft -autosize
$vmname = read-host "Select a VM to restore. (Note: If VM name includes spaces, use """"
around name i.e ""VM - Application1"")"

#If VM name has spaces, read-host needs "" around the input.
#Check to see if user input has "" and replace
if ($vmname.Contains('"')) {$vmname = $vmname.Replace('"','')}

```

```

#Check to see if VM name doesn't exists
$existingvms = (get-vm).name
While ($existingvms -notcontains $VMname) {
    $VMName = read-host "$vmname Does not exist. Please enter an existing VM name."
    if ($vmname.Contains('"')) {$vmname = $vmname.Replace('"','')}
    $existingvms = (get-vm).name
}

echo "The name of the VM is $vmname"
get-vm -name $vmname | fl * | Out-Null
get-vm -name $vmname | fl -Property VMName,VMID,state,path

#####
#Select an associated smartcopy to use in restoring the VM

echo "VM $vmname has the following SmartCopies..."

echo "Individual VM Smartcopies..."
#Get Individual Smartcopies of VM
Get-ASMSmartCopy -ObjectName "$vmname" | ft -Property
SmartCopyName,OriginalObjectID,ObjectName -AutoSize

echo "Smartcopy Collections..."
#Get Smartcopy collections which contain VM
get-asmcollection | ? {$_.component -like "$vmname"} | get-asmsmartcopy | ft -Property
SmartCopyName,OriginalObjectID,ObjectName -AutoSize

#Check to see if any smartcopies exist for VM
if (((Get-ASMSmartCopy -ObjectName "$vmname") -eq $null) -and ((get-asmcollection | ?
{$_.component -like "$vmname"}) -eq $null)) {
    echo "No ASM Smartcopies found for $vmname"; break
}

$scobjectname = read-host "Please enter a smartcopy ObjectName to use for the restore (use
copy and paste)"
#echo "smart copy objectname = $objectname" #Uncomment for Debug

$vmobjectname = (Get-ASMComponent -ObjectName $vmname).objectname
#echo "vm object name = $vmobjectname" #Uncomment for Debug

#####
#Restoring the VM

echo "Restoring $vmname from $objectname. Please wait..."

Restore-ASMSmartCopySelective -ObjectName "$scobjectname" -Selections "$vmobjectname" -
ErrorAction Stop

#Inject a pause until VM has been restored and has rebooted

while ((get-vm -name $vmname).heartbeat -ne "OkApplicationsHealthy") {write-host ">" -
NoNewline; start-sleep -Seconds 1}
echo "`n$vmname restored and online"

```



```
#####
#Optional - Create new smartcopy of restored vm
$createnewsmartcopy = read-host "Create New Smartcopy for $vmname ? (YES/NO) "

if ($createnewsmartcopy -eq "Yes") {
    echo "Creating new Smartcopy for $vmname"
    $date = date
        New-ASMSmartCopy -ObjectName $vmname -Comment "Restore point at $date" -
        ErrorAction Stop
    }

$answer = read-host "Hit any key to continue"

#End of Script
```



B Sample rapid VM deployment script

This appendix is a culmination of the technologies, processes, and techniques discussed in this document.

Being able to quickly respond to the ever-changing dynamics of a virtualized environment is a key success criterion for any administrator. One of the primary enablers for this success is being able to quickly deploy Hyper-V virtual machines. The fastest and most reliable way to perform the rapid deployment task is to script the entire deployment process. A typical script would collect user input and/or read in a VM configuration file, create the new virtual machine from a template virtual machine, and set up the VM network configuration so that at the end of the script a running virtual machine will be ready for access by the user community.

This section details a straw man script that can perform rapid Hyper-V virtual machine deployments using two-step method.

1. Create: Use EqualLogic Thin Clone technology that will use a thin clone of a dedicated template volume, mounted to the Hyper-V host server as the source VHD for a new virtual machine.
2. Copy: Copy and rename the template VHD from the dedicated template volume to a data store mounted to the Hyper-V host server.

Virtual machines built using this method are shown in Figure 7 as ApplicationVMxx. The diagram continues to build on the example configuration used in throughout this document.



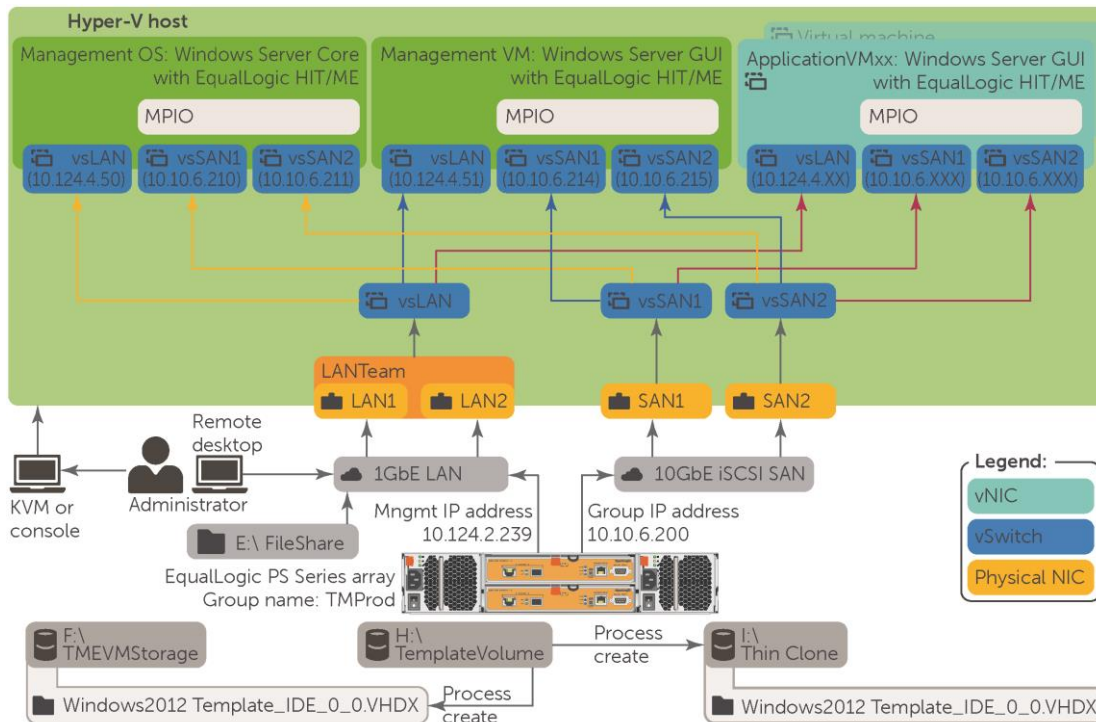


Figure 7 Configuration details

The script will use a combination of user inputs for things such as the new virtual machine name and a XML configuration file which contains standard values for the virtual network and storage environment. These values are used in the five primary functions called out in the script.

- **SelectMethod:** Select the VHD creation method; This function calls either.
 - **ProcessCreate:** Create EQL Thin Clone of a template volume.
 - **ProcessCopy:** Copy VHD from the template volume to the local VHD data store.
- **CreateVM:** Creates a virtual machine using inputted parameters and built VHD.
- **SetupNetworking:** Configures virtual machine network parameters (IP Addresses, DNS, Gateway and Subnet).

The scripts flow is shown in Figure 8.

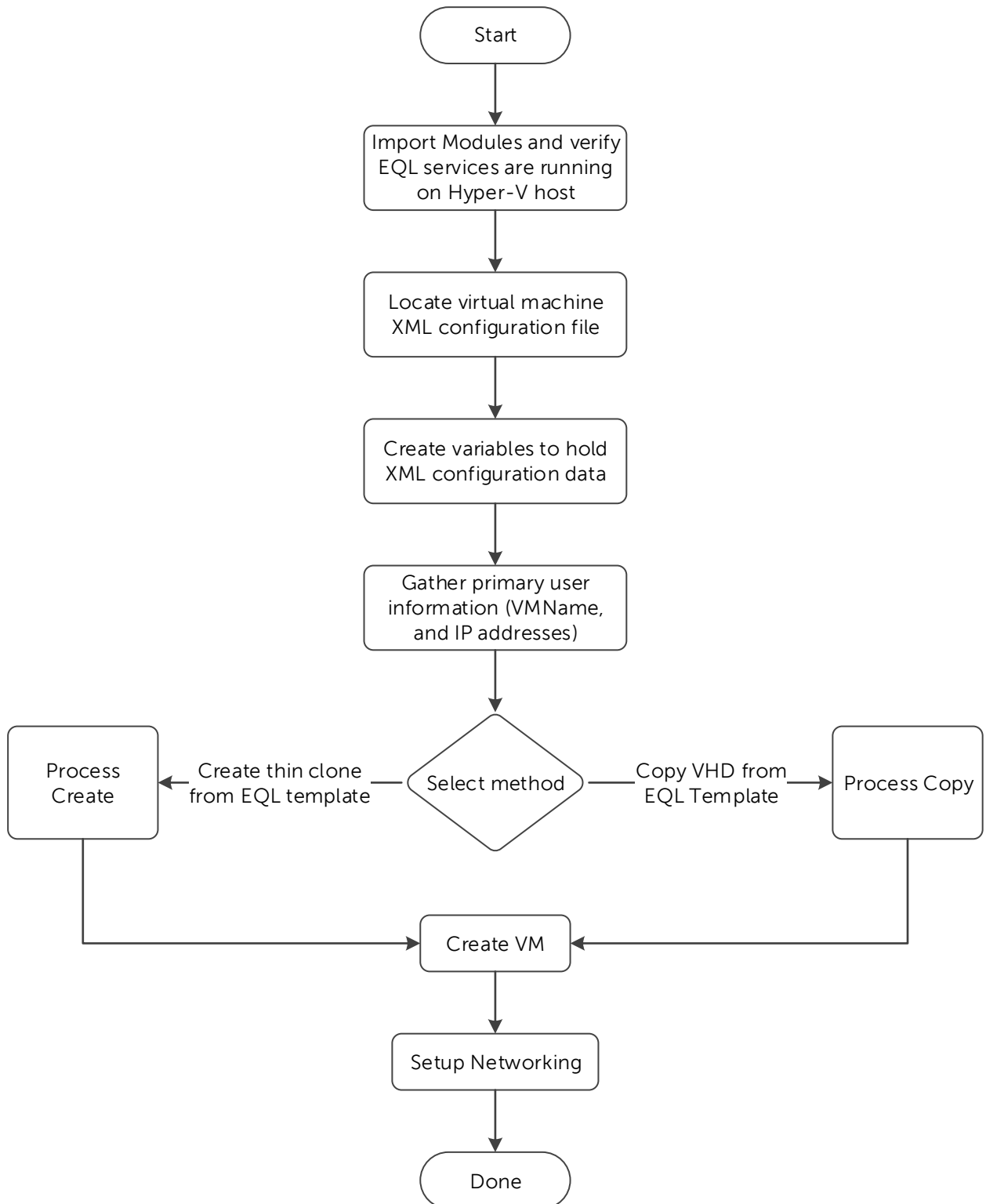


Figure 8 Rapid VM deployment script logical flow



Note: The following script should be considered a straw man because it has been set up to follow the examples used in this document. Some modification to values, paths, and formatting will most likely be required prior to result in a successful execution. The script segments detailed in this section are setup to be copied and pasted into an editor and then saved as a PowerShell script. Give careful attention to line wrapping that might have resulted due to formatting issues during publication of this document. In PowerShell, commands can consist of multiple lines when the backtick character (``) is used to join the lines. The backtick tells PowerShell that the command continues onto the next line. Commands can also be split at the pipe (|) character.

B.1 Create virtual machine XML configuration file

The first step to being able to deploy virtual machines quickly is to have a library of reusable virtual machine configuration files. These files, in either a XML or CSV format, can contain the following information.

- VM Configuration: Number of vCPUs, startup memory, dynamic memory usage, maximum memory, and minimum memory
- Default Network Configuration: Hyper-V host virtual switch names, LAN network configuration information and iSCSI network information
- EqualLogic Group Information
- EqualLogic Template Volume information

The following XML file contains this information and can be used to build a virtual machine from an EqualLogic template volume that contains a **sypred** VHD. Administrators can use files like this to build a library of reusable virtual machine XML configuration files and store them on a network file share. These XML files act as a set of standards or templates for building all virtual machines in the data center.

```
<?xml version="1.0" encoding="utf-8"?>
<Config>
  <VirtualMachine>
    <VMConfig VMCPU="2" VMStartupRAM="2GB" VMUseDynMem="YES" DynRAMMax="8GB"
DynRAMMin="512MB" />
  </VirtualMachine>
  <Network>
    <VirtualSwitchNames LAN="vsLAN" iSCSI1="vsSAN1" iSCSI2="vsSAN2" />
    <LANNetworkConfig Subnet="255.255.252.0"
DefaultGateway="10.124.4.1"
PrimaryDNS="10.124.6.245" />
    <iSCSINetworkConfig Subnet="255.255.0.0" />
  </Network>
  <EQLStorage>
    <EQLGroup Name="TMProd" />
    <EQLTemplate TempVolName="WinServ2012Template"
EQLACLCopyVol="WinServ2012Template"
TempVHDName="Windows2012Template_IDE_0_0.vhdx"
TempVHDDir="\hyper-v\virtual hard disks\"
  </EQLStorage>
</Config>
```




```
TempVMDir="\hyper-v\virtual machines\" />
    </EQLStorage>
</Config>
```

B.2 Import required modules and verify EqualLogic services

Step two of being able to deploy virtual machines quickly is importing the required modules and verifying EqualLogic services.

The opening part of the script will set the module path. It then imports the EqualLogic PowerShell tools, iSCSI, and the **set-vmnetworkconfiguration** modules. It also checks and verifies the required EqualLogic services are running on the Hyper-V host and if necessary it will restart them. Note the use of the % character which is short hand for the **foreach** cmdlet in the restart function. The \$ _ variable represents the objects returned by the **get-service** command.

```
#####
#Import Required Modules.
#####

Clear-Host
$ModulePath = "C:\Users\Administrator\Documents\WindowsPowerShell\Modules"
import-module -name "C:\program files\equallogic\bin\EqlPSTools.dll"
Import-Module -Name iSCSI

#Note: Information for following module can be found at:
#http://www.ravichaganti.com/blog/?p=2766

import-module $ModulePath\set-vmnetworkconfiguration.psm1

set-executionpolicy remotesigned -force

#####
#Verify that EQL Services are running on Hyper-V Host. Restart if necessary
#####

Write-Host "Checking to see if required EqualLogic services are running`n"
Get-Service | Where-Object {$_.displayname -like "Equal*"}

#Restart any stopped EQL Services
Get-Service | Where-Object {$_.displayname -like "Equal*"} | % {
    if($_.Status -eq "Stopped"){
        Write-Host "`nRestarting..."
        write-host $_.DisplayName
        Start-Service $_.name
    }
}
Write-Host "`nAll EqualLogic services started`n"
```

B.3 Locate virtual machine XML file and read in contents

Step three is in this section of the script; it locates and reads the virtual machine XML configuration file. In the code shown below, the user enters the drive letter that houses the file. The path and name of the file



are hard coded into the script. These could be gathered through user input or created dynamically with some further scripting. The directory path for the XML configuration file is then compiled as a concatenated string in the **\$configfilepath** variable. This variable is then used in the **ls -r** command. If the **ls -r** command locates the XML configuration file in the path, it grabs the full path string and places it into the **\$fullvmconfigfilepath** variable. The file specified in **\$fullvmconfigfilepath** is then read and parsed. If the **ls -r** command cannot locate the XML file in the specified path, an error is captured and the script halts

```
#####
#Locate and read in contents of VM configuration XML file
#####
#Get the drive letter name of the VM XML config file file share or location

Write-host "Select the drive letter which is the location of VM XML configuration file
directory"
get-psdrive -PSProvider FileSystem | ft -Property Name,DisplayRoot -AutoSize
$filessharedriveletter = read-host "Enter the Name of the Drive Letter for XML Config. Library"

#Hard coding the directory path and config file name. Could also be obtained by user input
$pathttoconfigfiles = "\hyper-v\config_files\"
$VMConfigFilename = "WinServ2012VMTempConfig.xml"

#Dynamically create path to XML config file and read in contents
#Catch errors and break script if XML file cannot be found or read
try {
    $filessharedrive = $filessharedriveletter + ":"
    $configfilepath = $filessharedrive + $pathttoconfigfiles
    $fullvmconfigfilepath = (ls -r $configfilepath -ErrorAction stop |
    ? {$_.name -eq $vmconfigfilename}).fullname

    #Read in contents from XML config file
    [xml]$configxml = get-content $fullvmconfigfilepath -ErrorAction stop

} catch {
    $ErrorMessage = $_.exception.message
    $FailedItem = $_.exception.ItemName
    write-Host "`nMiserable failure: Could not find Fileshare or
    Config File..`n$errormessage`n$faileditem"
    break
}
Write-host "VM XML configuration file found and read into script"
```

B.4 Create variables to hold XML configuration information and VM and VHD path information

Once the XML file is read and parsed, variables can be created that hold the virtual machine configuration information which will be used throughout the script. This happens in step four of enabling quick deployment of virtual machines.



Note: The XML file information is read as string data. When the virtual machine is created, the memory size data must be in an integer format. An easy way to convert the XML string data to an integer is simply add a "0" to the XML string.

```
#####
#Get values from XML file and create variables which will hold VM configuration information
#####

#Virtual Machine Settings
$CPU = $configxml.config.virtualmachine.vmconfig.vmcPU
$RAM = 0 + $configxml.config.virtualmachine.vmconfig.vmstartupram #Covert xml string to int
$maxram = 0 + $configxml.config.virtualmachine.vmconfig.dynrammax #Convert xml string to int
$minram = 0 + $configxml.config.virtualmachine.vmconfig.dynrammin #convert xml string to int
$usedynmem = $configxml.config.virtualmachine.vmconfig.vmunusedynmem

##Name of Virtual Switchs Being Used by VM
$LANNET = $configxml.config.network.virtualswitchnames.lan
$SAN1NET = $configxml.config.network.virtualswitchnames.iscsi1
$SAN2NET = $configxml.config.network.virtualswitchnames.iscsi2
## Default Network Parameters
$subnetmask = $configxml.config.network.lannetworkconfig.subnet
$defaultgateway = $configxml.config.network.lannetworkconfig.defaultgateway
$DNSServer = $configxml.config.network.lannetworkconfig.primarydns
$iSCSIsubnetmask = $configxml.config.network.iscsinetworkconfig.subnet
## EqualLogic PS Group Name
$eqlgroup = $configxml.config.eqlstorage.eqlgroup.name
## EqualLogic Template Volume Paramters
$eqltempvolname = $configxml.config.eqlstorage.eqltemplate.tempvolname
$eqlaclvolume = $configxml.config.eqlstorage.eqltemplate.eqlaclcopyvol
$tempvhddname = $configxml.config.eqlstorage.eqltemplate.tempvhddname
$tempvhddir = $configxml.config.eqlstorage.eqltemplate.tempvhddir
$tempvmdir = $configxml.config.eqlstorage.eqltemplate.tempvmdir

#Setup other path variables used in script
## Template Virtual hard disk location
$tempvhdd = $tempvhddir + $tempvhddname

#Template Volume Drive Letter and full path to template VHD
$tempvoldriveletter = (get-volume | ? {$_.FileSystemLabel -like "WinServ2012Temp*"}).DriveLetter
$fulltemppath = $tempvoldriveletter+" "+$tempvhdd

## Path to default Hyper-V Host VHD Datastore
$vhddefaultpath = (get-vmhost).virtualharddiskpath

## Hyper-V Host Computer Name
$HyperVHost = (get-vmhost).computername
```

B.5 Gather primary user input

Step five gathers the primary user input. Here the user provides input for the name of the new virtual machine. Even though the existing virtual machines are shown to the user in the **get-vm** command output, an error check is performed on the virtual machine name to make sure it is unique. The user is prompted for the name until a unique VM name is entered. Once a unique VM name is entered, the script



prompts the user if they wish to configure the virtual machine network IP addresses. If the user responds, "Yes", they are prompted to enter the LAN IP address. Note that in the script, error checking is performed against the inputted IP address format. This is done by verifying that the entered value for the IP address is an [IPAddress] data type.

```
#####
#Get name of VM from user input. Verify if VM vNIC network config is to be set up
#####

Write-host "`nThe Hyper-V host server $HyperVHost has the following virtual machines
configured:"
get-vm | ft -AutoSize

#Enter name of new virtual machine to create
$VMName = read-host "Enter the name of a new VM to create on $HyperVHost"
#PEBKAC error catch. Check to see if VM name already exists
(get-vm).name | % { while ($_ -match $VMName) {$VMName = read-host "$_ already exists.
Please enter a unique VM name."}}
#User is prompted for virtual machine LAN and iSCSI IP addresses
$configvmnet = Read-Host "Do you wish to configure the network for $VMName (YES/NO)?"
if ($configvmnet -eq "YES") {
    $vmip = read-host "`nEnter the LAN IP address of $VMName (xxx.xxx.xxx.xxx) "
    #Check LAN IP Address Format
    while ([bool]($vmip -as [ipaddress]) -eq $false) {
        $vmip = read-host "$vmip is an invalid IP format.`nEnter a valid LAN IP address"
    }
    $configiscsi = read-host "`nDo you wish to configure iSCSI vNICs for $VMname (YES/NO)?"
    if ($configiscsi -eq "YES") {
        $address = ""
        $myarray =@()
        while ($address -ne "END") {
            $address = read-host "Enter an IP address for $VMname iSCSI
vNIC (xxx.xxx.xxx.xxx). Type ""END"" to move to next section
of script"
            #Check iSCSI IP Address Format
            while ([bool]($address -as [ipaddress]) -eq $false) -and ($address -ne "END"))
            { $address = read-host "$address is an invalid IP format.`nEnter a valid iSCSI IP address or
""END"" to move on"
            }
            #Valid iSCSI IP Addresses are added to array
            if ($address -ne "END") {$myarray += $address}
        }
    } else {
        write-host "iSCSI Network will not be configured for $VMname"
    }
} else {
    Write-host "No network will be configured for the $VMName"
}
}
```



B.6 Function: SelectMethod

Step six introduces the first function used in the script that chooses a method for building the VHD. One of the main purposes PowerShell functions serve is to act like subroutines. The **selectmethod** function in the script is a routine where the user chooses the method for how the new virtual machine VHD is to be created. The methods used by the script are described in the welcome message. The user types either "Copy" or "Create" at the prompt and this value is stored in the **\$vmvhdc createmethod** variable. An error check is performed, and if the entered values are appropriate, the script will call the appropriate virtual machine VHD creation function.

The message portion of the function is optional, but printing some sort of description to the user on the status of the script is helpful. This message could be read from a file in a similar manner that the XML configuration file was read.

If ProcessCreate is the selected method, continue with the appendix section B.6.1.

If ProcessCopy is the selected method, continue with the appendix section B.6.2.

```
#####
#Function SelectMethod - select method on how the VM will be created
#####

function selectmethod {

    # Message with explanation of methods used by this script to create virtual machine's VHD
    $welcomemessage = "`nThis script will create the new virtual machine using one of two methods:`n
    The first method uses EqualLogic Templates and Thin Clone Technology. In this method a new
    virtual machine be created using by using an EqualLogic Template volume from which a new thin
    clone volume will be created and mounted to the Hyper-V host server. This new thin clone volume
    will house the template VHD which the new VM will use for its own VHD. The VHD on the thin clone
    volume will be renamed to reflect the name of the new VM. This first VM creation method could
    take a few moments to complete as a new volume needs to be created and provisioned to the Hyper-
    V host server prior to creating the new VM. This method will result in the new VM having its own
    unique volume.`n
    The second method will copy the template VHD file to the existing Hyper-V host default VHD
    datastore: $vhdefaultpath . The new VM will use the copied template VHD on the datastore volume
    as its VHD. The template VHD will be renamed to reflect the name of the new VM. The second
    method results in a faster VM create time as no new volume needs to be created and provisioned
    to the Hyper-V host server; however, the new VM could be sharing the datastore volume with other
    virtual machines.`n"

    #Display welcome message
    write-host $welcomemessage
    $vmvhdc createmethod = read-host "Select VM VHD Creation Method:
        (1) Create a new Thin Clone Volume
        (2) Copy Template VHD to Data Store`n
        Enter ""Copy"" or ""Create""

        #Error check to make sure Copy or Create are entered by user
        while (($vmvhdc createmethod -ne "Copy") -and ($vmvhdc createmethod -ne "Create")) {
            $vmvhdc createmethod = read-host "Please enter ""Copy"" or ""Create""
        }

        #Call either copy or create vm vhd creation function
        & "Process$vmvhdc createmethod"
```



B.6.1 } Function: ProcessCreate

If ProcessCreate was selected as the method for building the VHD, begin by creating a thin clone from the template. The ProcessCreate function is called by the script if the user chose to create a virtual machine which has its VHD on a unique dedicated volume. The new dedicated volume and associated VHD file is a thin clone of a template volume that is presented to the Hyper-V host. Recall from Appendix A that the template volume contains a VHD of a syspreped virtual machine. This function performs the following operations:

1. Create a thin clone of the template volume using the new-eqlthinclone command.
2. Sets the thin clone volume properties to be online and read write enabled.
3. Copies the EqualLogic access controls from the template and assigns them to the new thin clone volume.
4. Performs the required iSCSI target work including registering the iSCSI session of the thin clone and then updates the host storage cache on the Hyper-V host.
5. Prompts the user to enter a new drive letter for the thin clone volume where it is mounted on the Hyper-V host.
6. Sets the VHD path and virtual machine path variables to point to the thin clone volume; these variables are passed into the output stream.
7. Calls the CreateVM function once the thin clone volume is successfully mounted.

```
#####
#Function ProcessCreate - Create EQL Thin Clone for new VM VHD
#####

function ProcessCreate {

    write-host "`nSelection = $vmvhdcreatemethod - Thin Clone VM VHD Creation Method Selected"
    $enter = read-host "Hit any key to continue"

    Write-Host "Step 1: Creating EQL Thin Clone Volume`n"

    # Creating EQL Thin Clone Volume
    New-EqlThinClone -CloneName $VMName -VolumeName $eqltempvolname

    # Setting new EQL Thin Clone volume to read write access, online, and setting access
    controls
    Set-EqlVolume -VolumeName $VMName -OnlineStatus offline #Volume needs to offline to set RW
    Set-EqlVolume -VolumeName $VMName -AccessType read_write
    Set-EqlVolume -VolumeName $VMName -OnlineStatus online
    Get-EqlVolumeacl -VolumeName $eqlaclvolume | New-EqlVolumeAcl -VolumeName $VMName

    # Updating iSCSI Targets
    write-host "Updating iSCSI targets on $HyperVHost"
    $eqlgroupip = (get-eqlgroup).groupaddress
    Update-IscsiTargetPortal -TargetPortalAddress $eqlgroupip

    #Getting new thin clone iqn, connecting to Hyper-V host, and register iSCSI session
    $tclonetarget = (get-eqlvolume -VolumeName $VMName).iscsitargetname

    Write-host "The thin clone iSCSI target is $tclonetarget"
```



```

$enter = Read-Host "Hit any key to continue"

Write-host "Step 1a: Connecting $tclonetarget to hyper-v host $HyperVHost`n"
Connect-IscsiTarget -NodeAddress $tclonetarget
get-iscsitarget -NodeAddress $tclonetarget | get-iscsisession | register-iscsisession

#Printing iSCSI session status to screen
get-iscsitarget -NodeAddress $tclonetarget | get-iscsisession

#Optional pause in script
$enter = Read-Host "Target Connected. Hit any key to continue"

#Update Host storage cache - capture job id and print progress bar
write-host "New EQL thin clone volume for $VMName has been created and provisioned to
Hyper-V host successfully.`n"
Step 1b: Updating Host Storage Cache and setting thin clone properties.
This may take a minute..."
$jobid = (update-hoststoragecache -asjob).id
Write-host "Updating host storage cache. Please wait...`n"
While ((get-job -id $jobid).state -ne "Completed") {
    Write-host ">" -nonewline # write-progress cmdlet would be a more elegant way to do
this
    Start-sleep -seconds 5
}
Write-host "`nHost storage cache updated"

#Get new thin clone disk number and setting to online, read write access on the Hyper-V host
Write-host "`nGetting thin clone disk information and setting parameters. Please wait..."
$tclonedisknumber = (get-iscsitarget -NodeAddress $tclonetarget | get-iscsisession |
get-disk).number
set-disk -number $tclonedisknumber -IsReadOnly $false
set-disk -number $tclonedisknumber -IsOffline $false
Write-host "Thin clone disk parameters set to online and RW"

#Setting the drive letter to use for the thin clone volume on the Hyper-V host
$inusedriveletters = (get-psdrive -PSProvider FileSystem).name | sort
Write-Host "The following drive letters are in use: $inusedriveletters"
$newdriveletter = read-host "Select a new drive letter to use for the new thin clone volume"
write-host "Gathering partition information for new volume $newdriveletter. Please wait..."

$partitionnumber = (get-partition -DiskNumber $tclonedisknumber |
? {$_.type -eq "Basic"}).partitionnumber

Set-Partition -DiskNumber $tclonedisknumber -PartitionNumber $partitionnumber `
-NewDriveLetter $newdriveletter

set-volume -DriveLetter $newdriveletter -NewFileSystemLabel $VMName

write-host "Thin clone volume successfully mounted to Hyper-V host using
drive letter $newdriveletter`n"

get-volume -DriveLetter $newdriveletter | ft -AutoSize

#Verify that new drive is mapped in ps-drive
try {
    get-psdrive -name $newdriveletter -ErrorAction stop

```



```

    } catch {
        $ErrorMessage = $_.exception.message
        $FailedItem = $_.Exception.ItemName
        if ($ErrorMessage -like "Cannot find drive*") {
            $newdriveletterroot = $newdriveletter + ":\\"
            new-psdrive -name $Newdriveletter -PSProvider FileSystem -root
            $newdriveletterroot `
            -ErrorAction stop
        } else {
            write-host "Miserable failure: `n$ErrorMessage`n$FailedItem"
        }
    }

    #Setting VM VHD Path and virtual machine path
    #These variables will be passed to the CreateVM function
    $vhddpath = $newdriveletter + ":" + $tempvhdd
    $vmppath = $newdriveletter + ":" + $tempvmppath

    #Creating New Virtual Machine from VHD on thin clone
    $enter = read-host "Hit any key to begin creating new virtual machine $VMName"

    #Calling Create VM Function
    CreateVM
}

```

B.6.2 Function: ProcessCopy

If ProcessCopy was selected as the method for building the VHD, begin by copying file to the default local data store.

It is possible to copy the VHD from the template volume to the default virtual hard disk path on the Hyper-V host using the ProcessCopy function. The ProcessCopy method creates a VHD faster than the thin clone method. However, this typically results in the new virtual machine having its VHD file on a drive with other virtual machines; not an optimal solution for some applications. The default virtual hard disk path is usually a data store or CSV that will house multiple virtual machine VHD files.

Note: Recall that in the example used by this document, the default virtual hard disk path was set to **F:\Hyper-V\VirtualHardDisks**. This value is captured earlier in the script in the **\$vhdddefaultpath** variable.

This function essentially completes three operations:

1. It performs a background copy of the VHD from the template volume to the data store volume.
2. Sets the VHD Path and Virtual Machine path variables to point to the data store volume and passes the variables into the output stream.
3. Calls the CreateVM function.

```

#####
#Function ProcessCopy - Copy VHD from EQL Template Volume to DataStore location
#####
function ProcessCopy {

```




```

Write-Host "`nVHD Copy and Rename Method Selected"
Write-Host "Step 1: Copying template VHD to $vhddefaultpath`n"
$Sender = Read-Host "Hit any key to start copy"

    #Copy Template VHD to VHD Data Store location as background job and monitor progress

    $jobid = (start-job -name CopyTempVHD -ScriptBlock {cp $args[0] $args[1] -recurse -force} `
-argumentlist $fulltemppath,$vhddefaultpath).id

    Write-host "Copying VHD Template. Please wait...`n"
    While ((get-job -id $jobid).state -ne "Completed") {Write-host ">" -nonewline;
        Start-sleep -seconds 1}
    Write-host "`nTemplate VHD Copied"

    #Setting VM VHD Path and virtual machine path
    #These variables will be passed to the CreateVM function
    $vhddpath = $vhddefaultpath + $tempvhddname
    $vmppath = (get-vmhost).virtualmachinepath

    #Optional pause in script
    $Sender = read-host "Hit any key to begin creating new virtual machine $VMName"

    #Calling Create VM Function
    CreateVM
}

```

B.7 Function: CreateVM

Step seven of enabling quick deployment of virtual machines is to create a virtual machine using VHD and configuration variables from an XML file and user input.

The CreateVM function is the part of script that actually builds the new virtual machine. It uses the VHD built by both the ProcessCreate or ProcessCopy functions along with the VHD path variable, and the configuration variables created from the VM XML configuration file earlier in the script. Essentially the function performs the following operations.

1. Creates the new virtual machine with the CPU and memory settings specified from the XML configuration file and the VHD and VHD path variables from either the ProcessCopy or ProcessCreate functions.
2. Renames the VHD file to match the name of the virtual machine for easier identification.
3. Creates and attaches vNICs to each virtual switch on the Hyper-V host.
4. Starts up the new virtual machine.

```

#####
#Function CreateVM - Create Virtual Machine
#####

function CreateVM {

    #Create VM and Configure Settings
    Write-Host "Step 2: Creating VM $VMName and Configuring CPU Settings`n"

```



```

Write-Host "`n#####`n
Creating Virtual Machine with the following parameters:`n
Name: $VMName
Number of Virtual CPUs: $CPU
RAM Amount: $RAM
Use Dynamic RAM: $usedynmem
Path to VM config files: $vmopath
Path to VHDX file: $vhopath
LAN IP Address: $vmip
iSCSI IP Addresses: $myarray
`n#####`n`n"

$enter = read-host "Hit any key to continue creating new virtual machine $VMName
with above parameters"

#Create New VM
new-vm $VMName -MemoryStartupBytes $RAM -Path $vmopath -VHDPATH $vhopath
Set-VMProcessor $VMName -Count $CPU

#Set Up Dynamic Memory
if ($usedynmem = "YES") {
    Write-host "Configuring $VMName for Dynamic Memory`n"
    set-vmmemory $VMName -startupbytes 2GB -dynamicmemoryenabled $True `
-minimumbytes $minram -maximumbytes $maxram -erroraction stop
}

Write-host "Virtual Machine $VMName created successfully."

#Rename VHD to match the name of the new VM
Write-host "Step 2a: Renaming VHD to use name of $VMName"
Get-VMHardDiskDrive -vmname $VMName | % {

    $oldvhdName = $_.Path
    $newvhdName = $oldvhdName.Substring(0,$oldvhdName.LastIndexOf("\")) `
    + "\" + $_.VMName + "_" + $_.ControllerType `
    + "_" + $_.ControllerNumber + "_" `
    + $_.ControllerLocation `
    + $oldvhdName.Substring($oldvhdName.LastIndexOf("."))
    Rename-Item $oldvhdName $newvhdName
    Set-VMHardDiskDrive -VMName $_.VMName -Path $newvhdName `
    -ControllerType $_.ControllerType -ControllerNumber `
    $_.ControllerNumber -ControllerLocation `
    $_.ControllerLocation
}

#Connect Virtual Switches to VM
if ($configvmnet -eq "YES") {
    Write-host "Step 2b: Creating Virtual NIC Adapters for $VMName"
    foreach ($vswitch in (get-vmvswitch).name) {
        Write-host "`nCreating a virtual NIC adapter for virtual switch $vswitch
on $VMName"
        Add-VMNetworkAdapter -vmname $VMName -switchname $vswitch -erroraction
stop
    }
}

#Startup Virtual Machine

```



```

        get-vm | ft -AutoSize
write-host "`n Step 2c: Starting virtual machine $VMname...`n"
        start-vm -vmname $VMName

        #Monitor VM Start Up Progress
while ((get-vm -name $vmname).heartbeat -ne "OkApplicationsHealthy") {write-host ">" -NoNewline;
start-sleep -Seconds 1}

        Write-host "`n`nVirtual Machine $VMName successfully started and is ready
for remote IP configuration`n"

    }

```

B.8 Function: SetupNetworking

Step eight of enabling quick deployment of virtual machines is to set up the new virtual machine network configuration.

The final function used by the script is the SetupNetworking function. This function configures the virtual machine LAN vNIC and iSCSI vNICs using the user designated IP addresses and the network configuration information (DNS, subnets, and gateway) from the XML configuration file. The core operation of the function uses the **set-vmnetworkconfiguration** command that was imported from the **set-vmnetworkconfiguration** module. See <http://www.ravichaganti.com/blog/?p=2766> for more information.

```

#####
#Function SetupNetworking - Set up VM LAN / iSCSI SAN IP Configuration
#####
function setupnetworking {

    write-Host "Step 3: Configuring the network for $VMName using:
    IP Address: $vmip, Subnet Mask: $subnetmask, Default Gateway: $defaultgateway,
and DNS Server: $DNSserver`n`nPlease wait..."

    #Adding a 10 second delay in script incase VM has not fully brought up network interfaces
    start-sleep -seconds 10

    #Setting LAN vNIC Network Configuration
    try {
        get-vmnetworkadapter -vmname $VMName | where-object {$_.SwitchName -like "*LAN"}
    |
        set-vmnetworkconfiguration -ipaddress $vmip -subnet $subnetmask `
        -defaultgateway $defaultgateway -DNSserver $DNSserver -ErrorAction Stop
    } catch {
        $ErrorMessage = $_.exception.message
        $FailedItem = $_.Exception.ItemName
        write-host "`nMiserable LAN IP configuration failure:
`n$ErrorMessage`n$FailedItem"
        break
    }
    #Setting iSCSI vNIC Network Configuration
    if ($configiscsi -eq "YES") {
        $i=0
    }
}

```



```

        foreach ($iscsivnic in (get-vmnetworkadapter -vmname $VMName |
        where-object {$_.SwitchName -like "*SAN*"}).switchname) {
            $iscsiip = $myarray[$i]
            try {
                get-vmnetworkadapter -vmname $VMName |
where-object {$_.SwitchName -eq "$iscsivnic"} |
set-vmnetworkconfiguration -ipaddress $iscsiip `
-subnet $iSCSIsubnetmask -ErrorAction stop
            } catch {
                $ErrorMessage = $_.exception.message; $FailedItem =
$_Exception.ItemName
write-host "`nMiserable iSCSI IP Configuration failure: `n$ErrorMessage`n$FailedItem"
                break
            }
            $i++
            write-host "`n$iscsivnic IP address set to $iscsiip"
        }
    }
    Write-host "`nNetwork config finished. $VMName is can be accessed through Remote Desktop`n"
}

```

B.9 Main Logic

This final section of Appendix B that summarizes the flow of the script. The first function called is **selectmethod**; it creates or copies the VHD and then builds the virtual machine by calling the **CreateVM** function. If the user selects to configure the network settings for the new virtual machine, the **setupnetworking** function is called. At the end of the script, the user has a new running virtual machine that is network accessible.

```

#####
####
#Main Logic of Script
#####
####

#Import modules verify EQL services are running on Hyper-V host

#Locate XML Config File and import values

#Enter user information - VMName, LAN and iSCSI IP Addresses

#Call function SelectMethod
Selectmethod

#If user wishes to setup VM Network Config, call function SetupNetworking
if ($configvmnet -eq "YES") {
    setupnetworking
}

```



C Provisioning the Hyper-V environment

This appendix describes provisioning the Hyper-V environment with EqualLogic storage using HIT/Microsoft and PowerShell scripting.

C.1 Using the EqualLogic SMP integration

One of the more powerful operations that can be done from Server Manager on the management virtual machine is to directly provision EqualLogic storage using the EqualLogic SMP integration (installed with the EqualLogic Host Integration Tools). The EqualLogic SMP integration into Windows Server essentially allows an administrator to create and provision EqualLogic storage using the Windows Server Manager Console on the Hyper-V host or management virtual machine. In the example presented in this paper, Server Manager was used to:

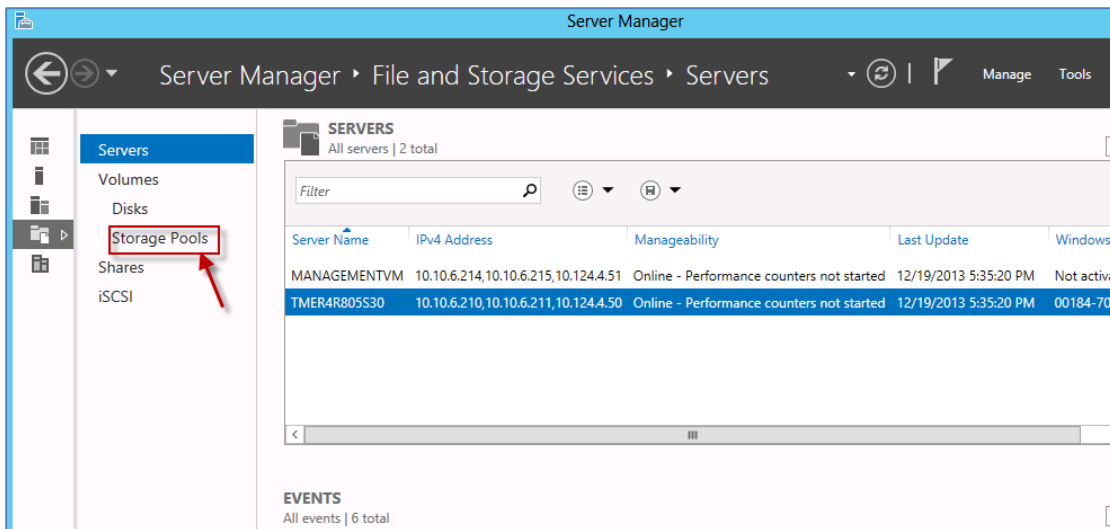
1. Create a new EqualLogic volume from the TMProd PS Group using the Gold Pool
2. Provision it to the IQN of the Hyper-V Server
3. Rescan the Hyper-V Host storage configuration
4. Initialize the new disk
5. Create a new volume on the new disk and format the volume

Opening iSCSI tools or EqualLogic Group Manager is not needed to perform these provisioning steps. A primary benefit of EqualLogic SMP integration into Windows is that it allows an administrator to use Server Manager as the single tool to manage the server, Hyper-V environment, and storage environment. Start this operation from the management virtual machine using the following steps.

1. Open Server Manager and select **File and Storage Services** on the left-side menu to open the server selection screen.
2. Highlight the Hyper-V Server and then click **Storage Pools**.

Note: This operation assumes that the Hyper-V server has been added as a managed server in Server Manager.

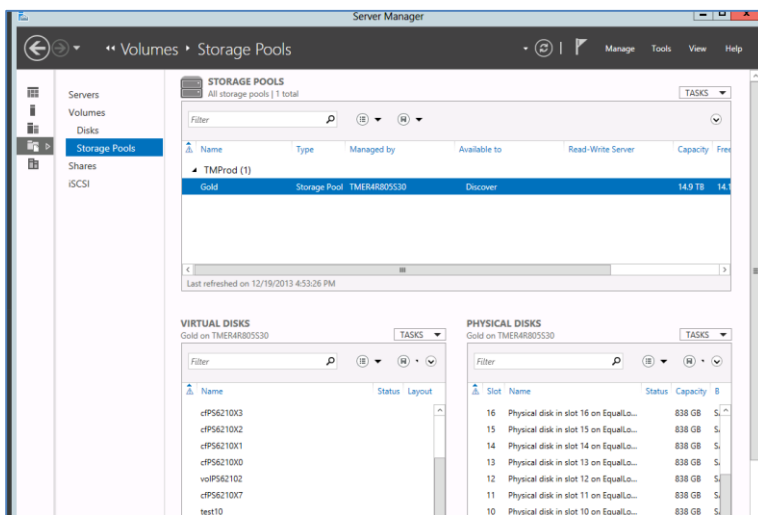




The selected storage pool information is displayed in three sections:

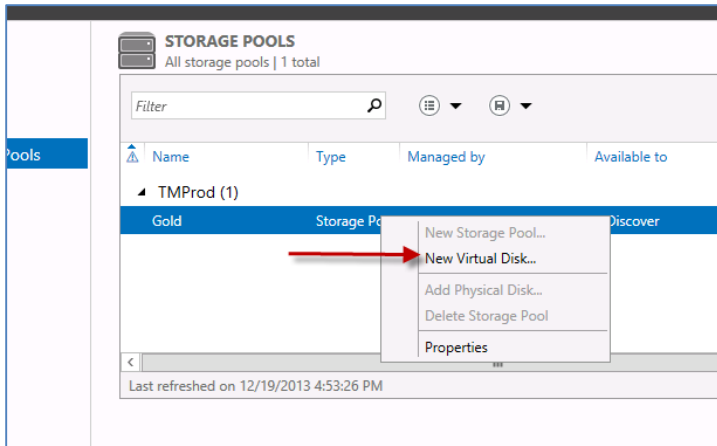
- **PHYSICAL DISKS:** EqualLogic hard disks information
- **VIRTUAL DISKS:** Existing EqualLogic volumes in the PS Group is
- **STORAGE POOLS:** Existing storage pools in the EqualLogic group

In this example, there is one storage pool labeled, "Gold" associated with the TMProd PS Group.



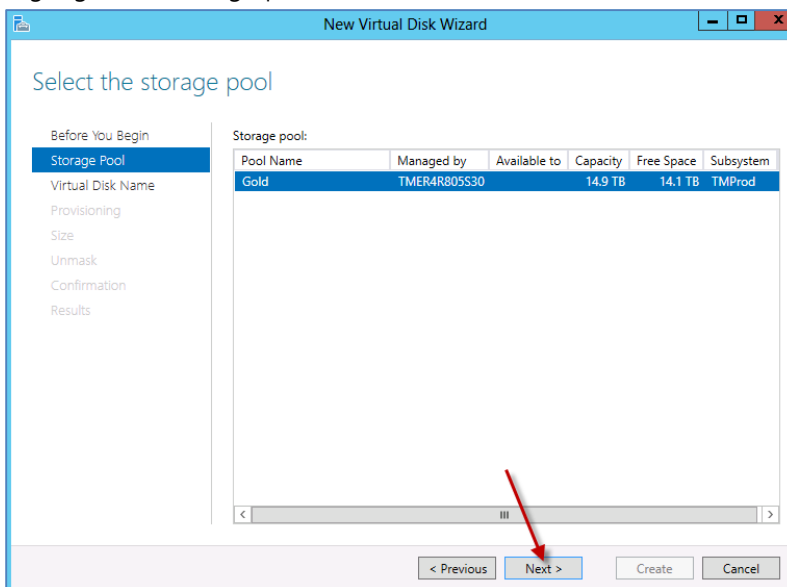
3. To start creating a new EqualLogic volume through Sever Manager, right click the EqualLogic storage pool and click **New Virtual Disk** in the pop-up menu.



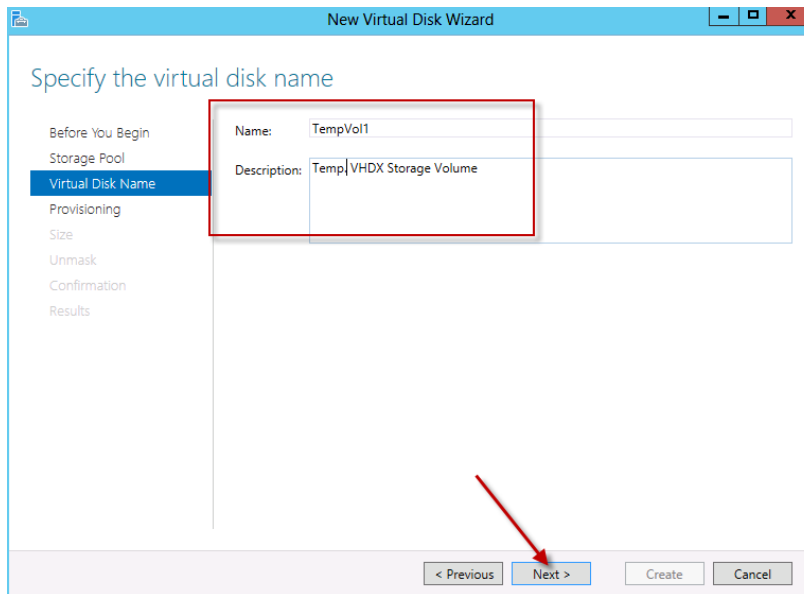


The **New Virtual Disk Wizard** is a guide for creating a new volume from the EqualLogic storage pool and presenting it to the Server Core Hyper-V host as a virtual disk. The first screen in the wizard displays the total capacity, free space, and subsystem (PS Group) for the selected EqualLogic storage pool.

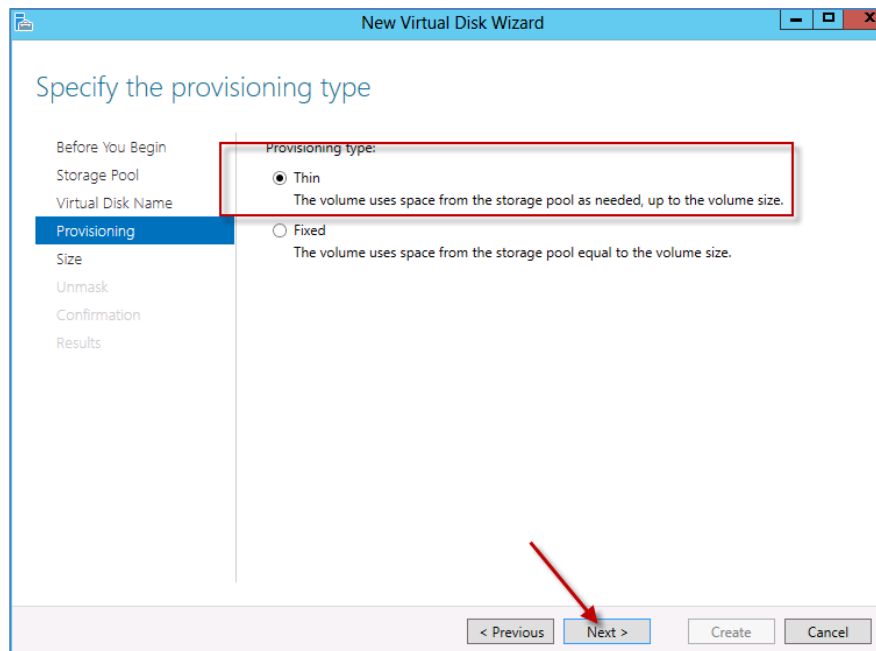
4. Highlight the storage pool and click **Next**.



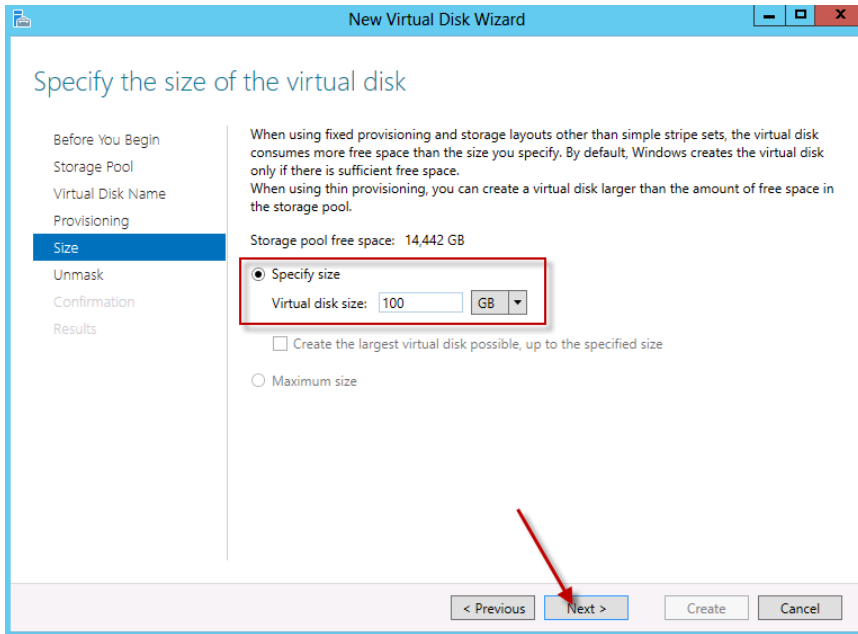
5. Enter a name for the new virtual disk and a description (optional), and then click **Next**. In the example, the new virtual disk is labeled, "TempVol1". This is the name that will be used to create the EqualLogic volume in the PS Group.



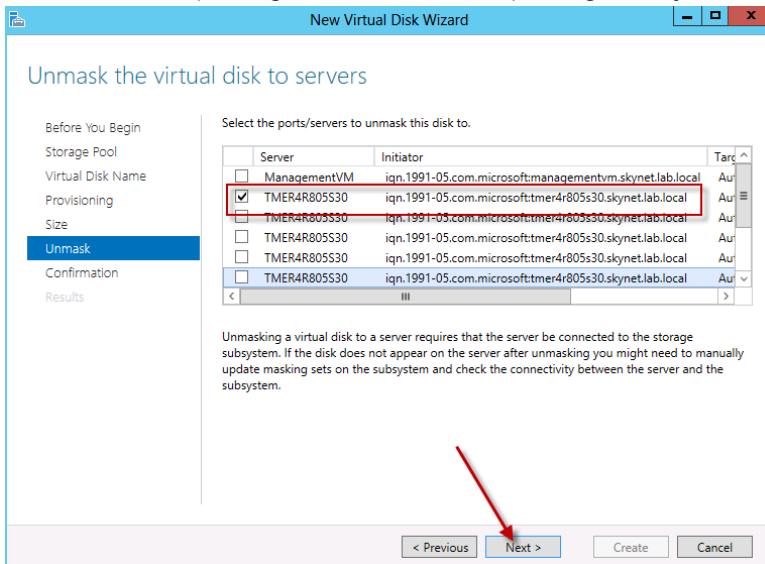
6. Specify if the virtual disk will be a thin provisioned volume, and then click **Next**. If **Thin** is selected, the volume created on the EqualLogic array will be thin provisioned and use a 100% snapshot volume reserve.



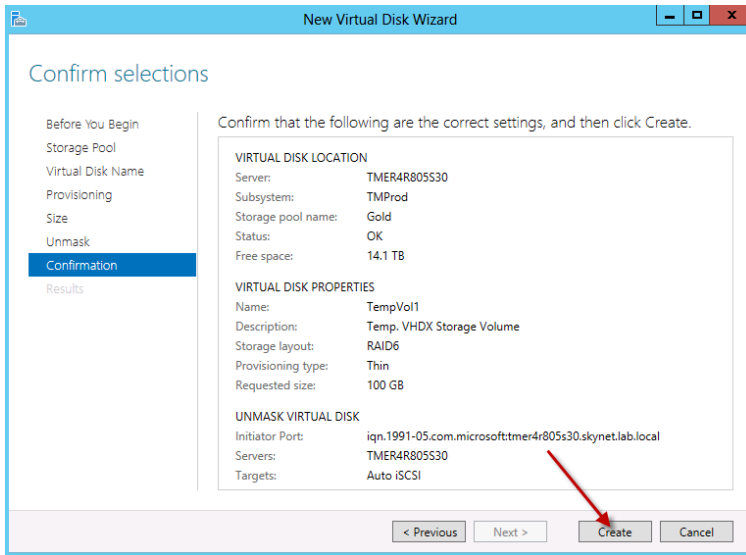
7. Enter a size for the new virtual disk (EqualLogic volume). In this example, 100GB is used. Click **Next** to continue.



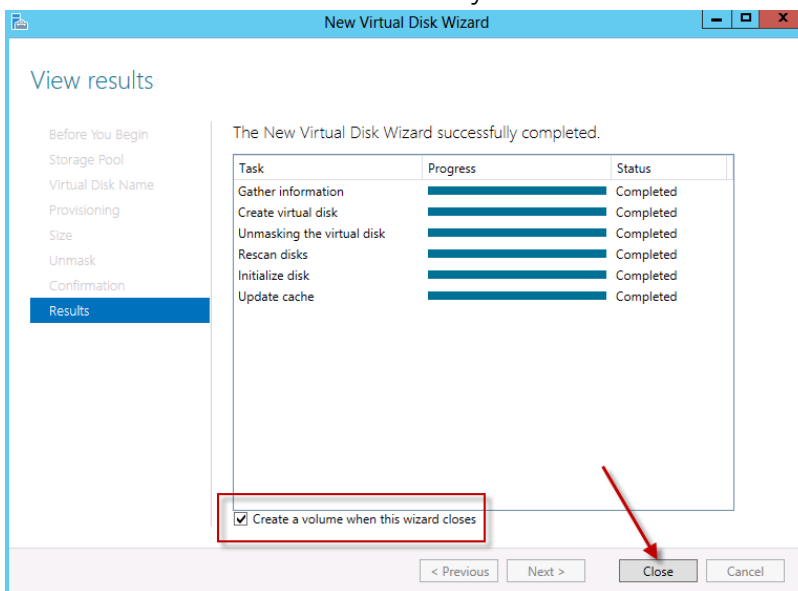
8. Select an iSCSI initiator to mask the new device to. In the example below, both the initiators for the Hyper-V host and Management VM appear. To mask the device to the Hyper-V server, select the associated iSCSI initiator and click next. This action creates an access control point for the Hyper-V server on the EqualLogic volume in the EqualLogic array.



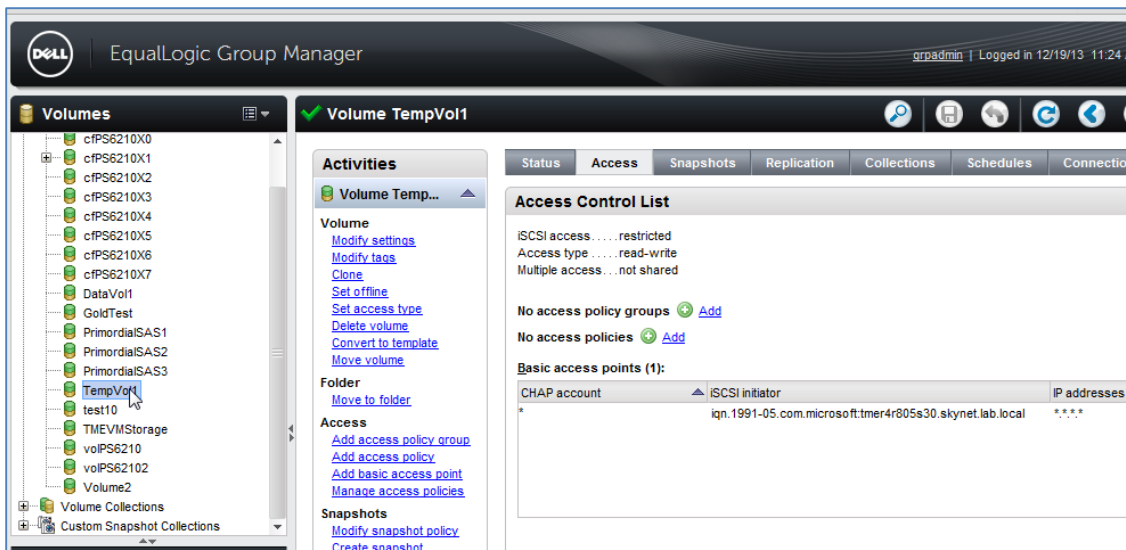
9. Review the information in the confirmation screen and click **Create** to make the new virtual disk (EqualLogic volume).



10. The steps and progress are displayed while the virtual disk is created. At the bottom of the status window, the option to create and format a volume on the new virtual disk when the wizard is closes is offered. Close the wizard when ready.

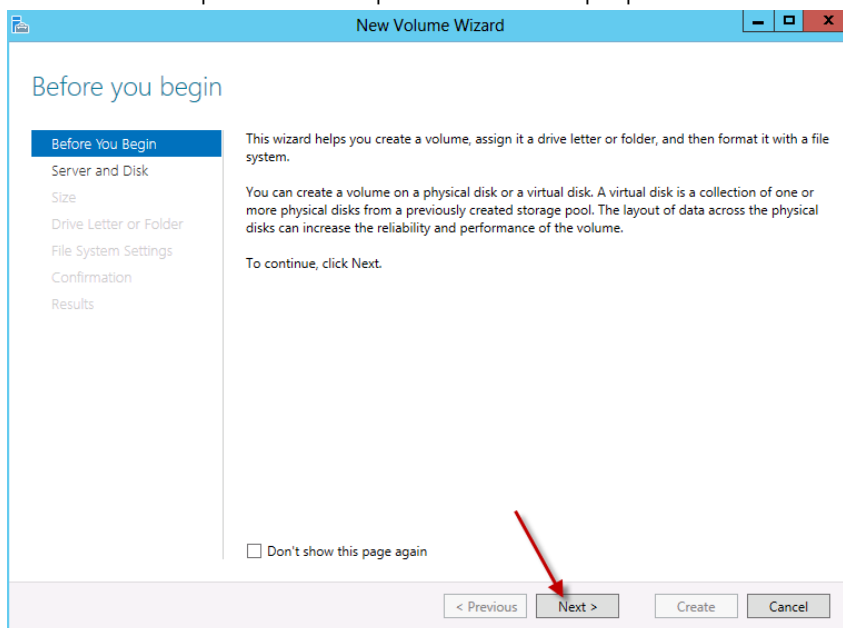


In Group Manager, the new virtual disk automatically appears as a new volume and does not require further action from the administrator. The volume can be selected and examined if needed. On the **Access** tab, the iSCSI initiator of the Hyper-V host is displayed as a basic access control point.



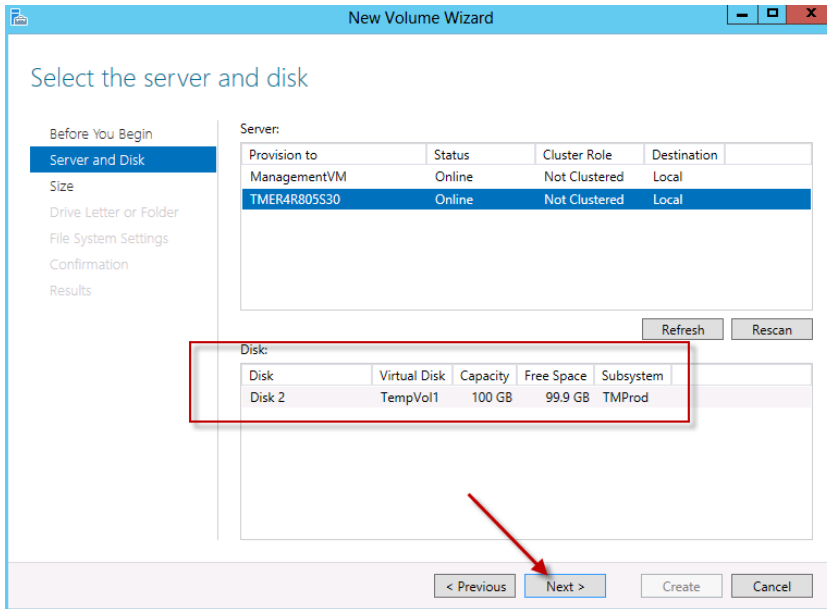
If the **Create a volume is selected when this wizard closes** option was checked in the final screen of the **Create Virtual Disk Wizard**, then the **New Volume Wizard** opens when the **Create Virtual Disk Wizard** is closed.

1. The first screen provides an explanation for the purpose of the wizard; click **Next** to begin.

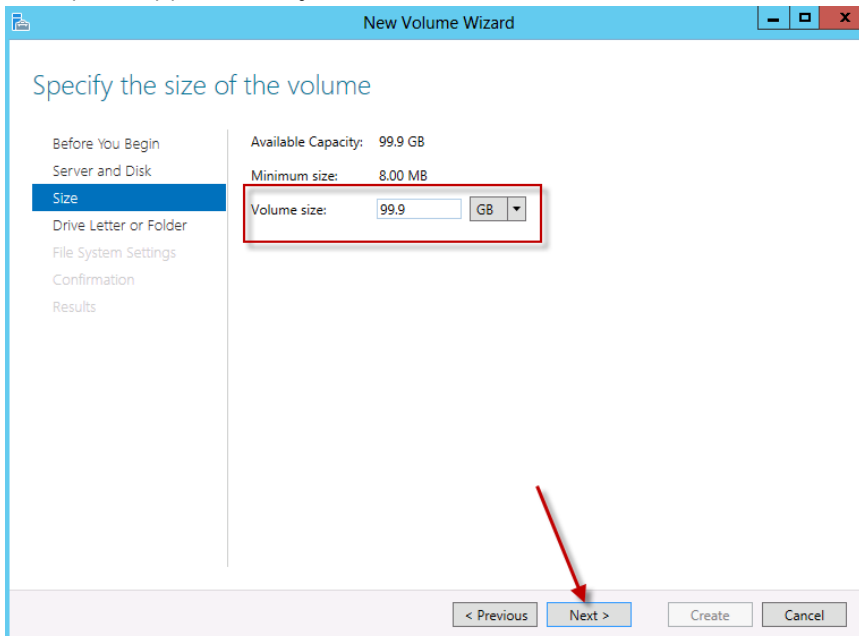


2. Select the Hyper-V server (TMER4R805S30 in the example) and disk (TempVol1) where the volume is needed. Click **Next** to continue.

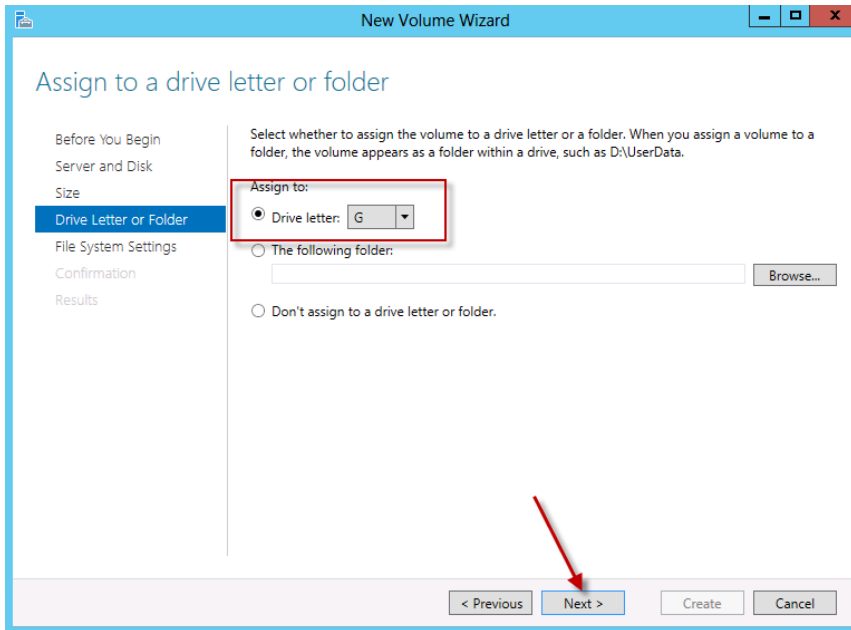




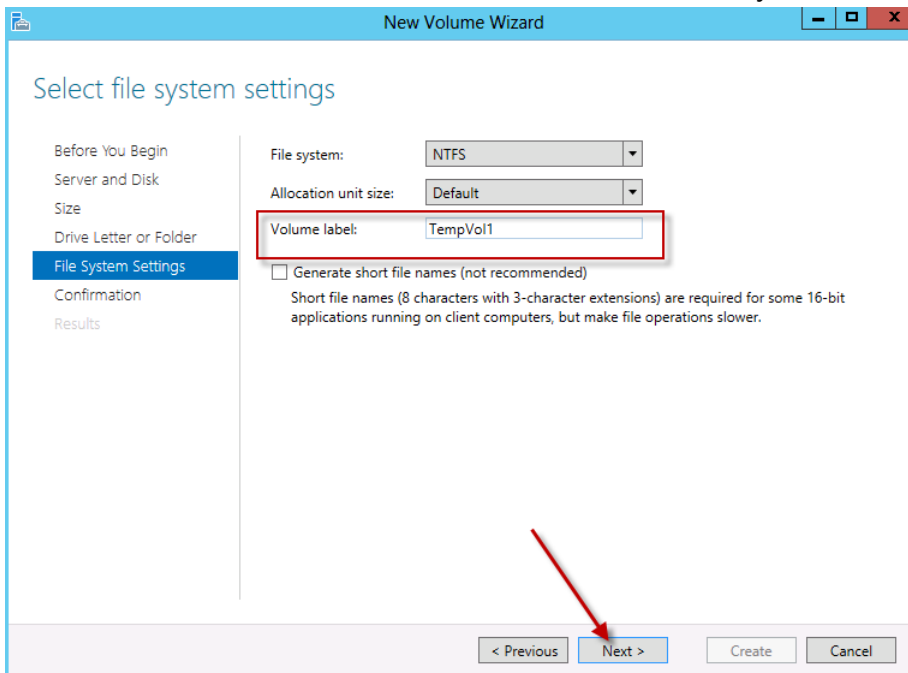
- Assign a volume size or accept the default (the maximum size available). The entire virtual disk in this example is approximately 100 GB. Click **Next**.



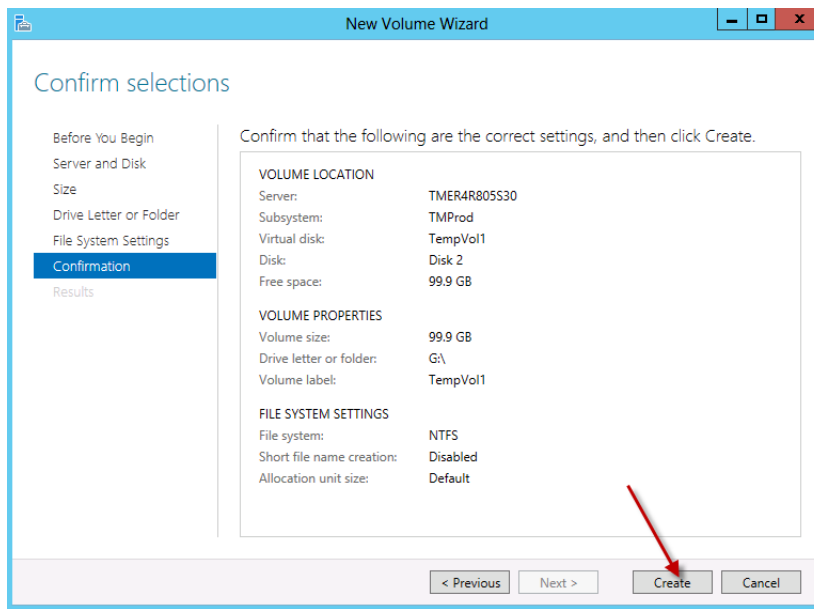
- Enter a drive letter for the volume (G in the example) and click **Next**.



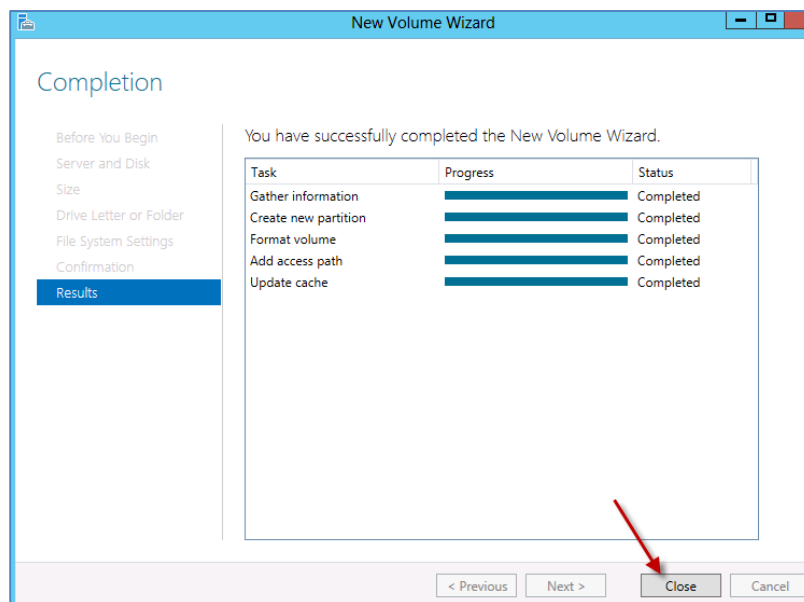
5. Select the File System type (the default NTFS is selected); a volume label is provided. In the example, the name of the virtual disk is used as the volume label for easy identification. Click **Next**.



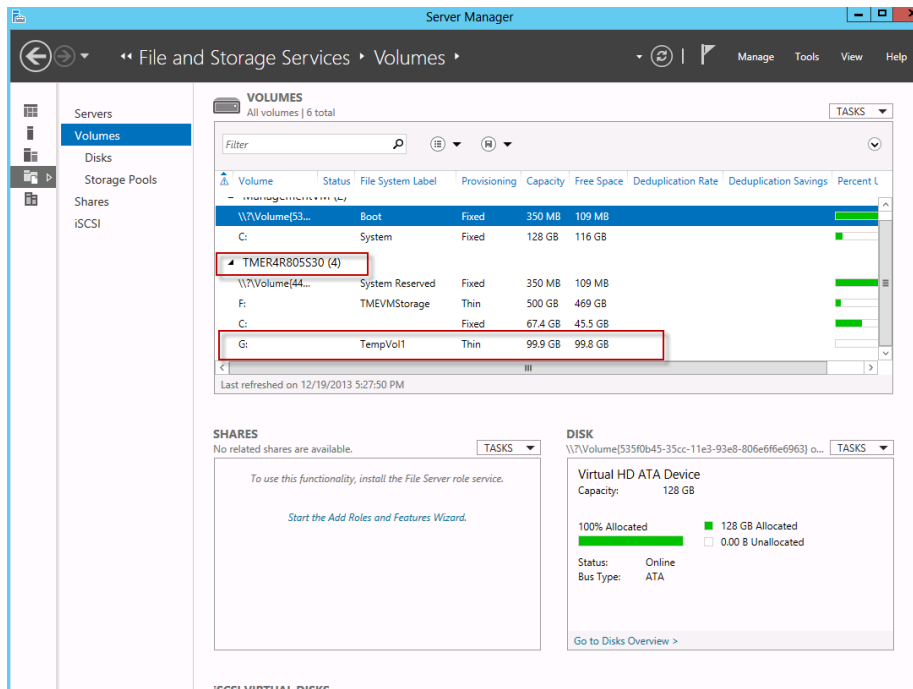
6. Review the displayed information and click **Create** to make the new volume.



7. The progress is displayed during the creation process. Once completed, click **Close** to exit the wizard.



The new volume (G:) is visible in the server manager that is associated to the Hyper-V Server (TMER4R805S30 in the example).



- The new volume can be viewed using PowerShell commands on the Hyper-V host using the **get-volume** command.

```
[TMER4R805S30] PS C:\>get-volume | ft -autosize
```

This output shows that volume G (TempVol1) is visible and ready for data from the Hyper-V server.

| DriveLetter | FileSystemLabel | FileSystem | DriveType | HealthStatus | SizeRemaining | Size |
|-------------|-----------------|------------|-----------|--------------|---------------|-----------|
| | System Reserved | NTFS | Fixed | Healthy | 108.69 MB | 350 MB |
| G | TempVol1 | NTFS | Fixed | Healthy | 99.77 GB | 99.88 GB |
| F | TMEVMStorage | NTFS | Fixed | Healthy | 468.76 GB | 499.88 GB |
| C | | NTFS | Fixed | Healthy | 45.52 GB | 67.41 GB |
| D | | | CD-ROM | Healthy | 0 B | 0 B |

- To test writing to the volume, change the directory to the volume and create a simple text file. In the example, the **get-process** command output is piped to an out file labeled, "gp.txt". The file is then readable with the **get-content** command.

```
[TMER4R805S30] PS C:\>cd G:\
```

```
[TMER4R805S30] PS G:\>get-process | out-file ./gp.txt
```

```
[TMER4R805S30] PS G:\>get-content -first 10 ./gp.txt
```

Sample output:

| Handles | NPM (K) | PM (K) | WS (K) | VM (M) | CPU (s) | Id | ProcessName |
|---------|---------|--------|--------|--------|---------|----|-------------|
| ----- | ----- | ----- | ----- | ----- | ----- | -- | ----- |



| | | | | | | | |
|-----|----|------|------|-----|-------|------|---------|
| 86 | 8 | 2872 | 7604 | 156 | 0.12 | 3460 | cmd |
| 45 | 7 | 1064 | 4808 | 68 | 0.28 | 3468 | conhost |
| 45 | 7 | 2308 | 6432 | 70 | 11.43 | 3844 | conhost |
| 218 | 12 | 1296 | 3716 | 50 | 2.59 | 760 | csrss |
| 72 | 11 | 1132 | 3380 | 45 | 0.89 | 824 | csrss |
| 126 | 10 | 1460 | 7740 | 167 | 1.65 | 2164 | csrss |
| 78 | 7 | 1068 | 4440 | 29 | 0.03 | 3976 | dllhost |

- Details about the new disk can be gathered on the Hyper-V server using the following PowerShell command that keys off of the EqualLogic volume name, "TempVol1", in the iSCSI target node address.

```
[TMER4R805S30] PS G:\> get-iscsitarget | ? {$_.nodeaddress -like
"*TempVol1*"} | get-iscsisession | get-disk | fl *
```

Sample output:

```
PartitionStyle      : GPT
ProvisioningType    : Thin
OperationalStatus   : Online
HealthStatus        : Healthy
...
AllocatedSize       : 107379425280
...
IsOffline           : False
IsReadOnly          : False
IsSystem            : False
LargestFreeExtent   : 0
Location            :
LogicalSectorSize   : 512
Manufacturer        : EQLOGIC
Model               : 100E-00
...
```

- Finally, update any Hyper-V environment configuration documentation (diagrams and /or tables) with information about the new volume provisioned to the Hyper-V host.



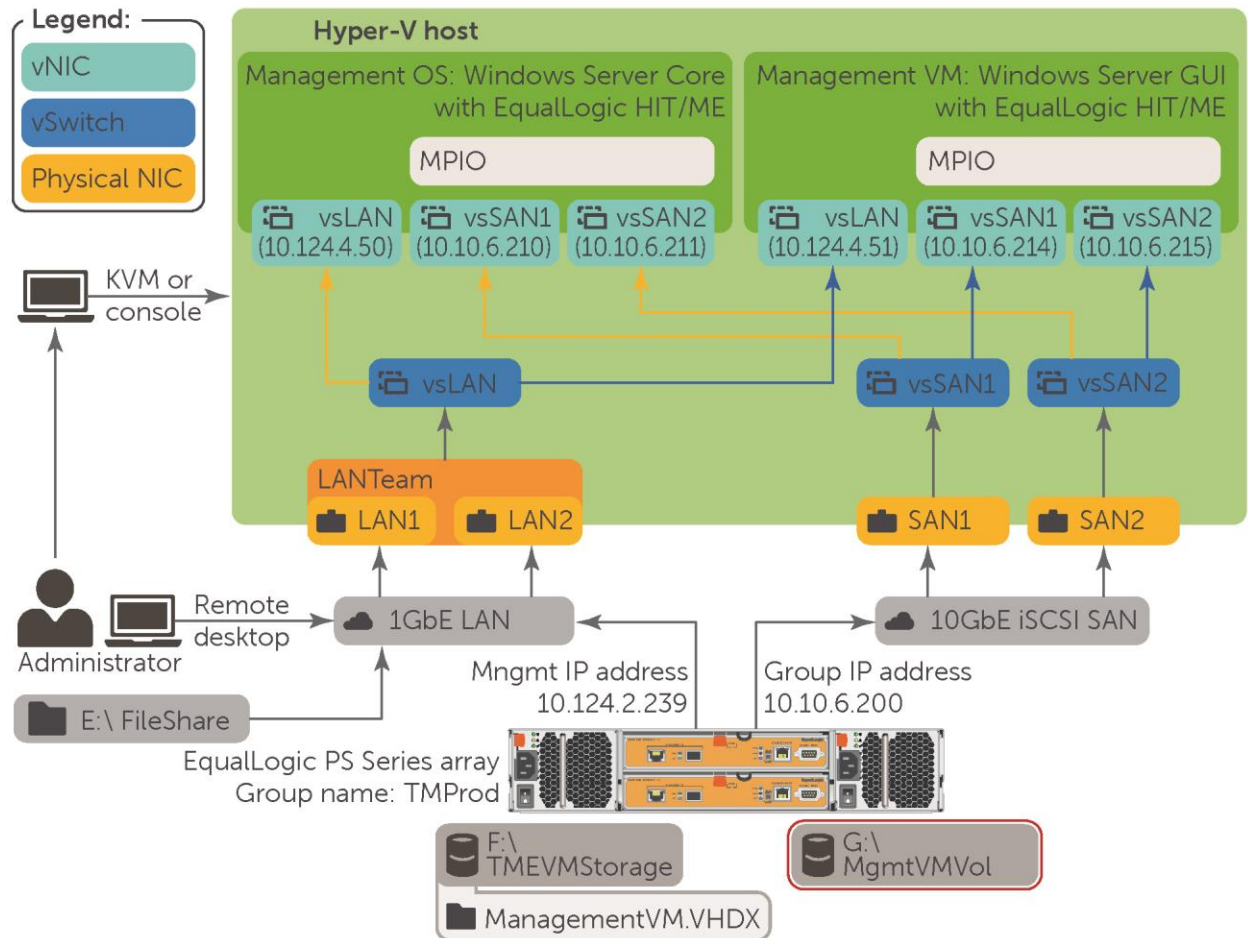


Figure 9 Updated configuration with the newly provisioned volume

C.2 Using HIT/Microsoft PowerShell tools

The following sample PowerShell script performs the same operations explained in the previous section.

```
<#
.SYNOPSIS
Creates and mounts a new EqualLogic volume to any iSCSI connected VM or host

.DESCRIPTION
This script will create and mount a new EqualLogic volume to any iSCSI connected VM or host.
It will first check for required modules and verify the required EQL services are running.
It will then prompt the user for input for EQL Pool, new volume name, new volume size, and
if thin provisioning is to be used. It then will create the volume, perform all iscsi
initiator work,
and then initialize, create a partition, assign a drive letter, and mount the new volume
to the host
or VM.
```

.EXAMPLE OUTPUT

Checking to see if required EqualLogic services are running

| Status | Name | DisplayName |
|---------|---------------|--|
| Running | EHCMSvc | EqualLogic Host Connection Manageme... |
| Running | EqlASMAgent | EqualLogic Auto-Snapshot Manager Agent |
| Running | EqlLogd | EqualLogic Trace Logging Service |
| Running | EqlReqService | EqualLogic VSS Requestor |
| Running | EqlSMPHost | EqualLogic SMP Host Service |
| Running | EqlVdsHwPrv | EqualLogic VDS Hardware Provider |
| Running | EqlVss | EqualLogic VSS Service |

All EqualLogic services started

Searching for EqualLogic Storage Pools....

The following EqualLogic storage pools were found:

| StoragePoolName | GroupName | GroupAddress | FreeSpaceMB |
|-----------------|-----------|--------------|-------------|
| Gold | TMProd | 10.10.6.200 | 11065665 |

Please select a storage pool name from the above list: Gold

Please type a name for the volume to be created: 50GBTestVol

Please select a size for the volume in MB: 50000

Use thin provisioning for volume? (Yes/No): Yes

This hosts initiator is: iqn.1991-05.com.microsoft:tmer4r805s30.skynet.lab.local

Creating a new 50000 MB EqualLogic volume called 50GBTestVol

PsApiCmdletCommon.NewVolumeAcl 50GBTestVol completed successfully.

| VolumeName | VolumeSizeMB | Groupname | StoragePoolName | HostingMembers | SessionId |
|----------------------------------|--------------|-----------|-----------------|----------------|-----------|
| 50GBTestVol | 50000 | Gold | TMProd6210X | | |
| 6019CBF194DA71B4F7221504000020C3 | | | | | |

Updating iscsi targets on host. Please wait...

Finished updating iscsi targets on host.

The iscsi target for 50GBTestVol is: iqn.2001-05.com.equallogic:0-1cb196-06435a94f-b9bbc511ddb533ec-50gbtestvol

Successfully connected to iscsitarget iqn.2001-05.com.equallogic:0-1cb196-06435a94f-b9bbc511ddb533ec-50gbtestvol

Successfully registered iscsitarget iqn.2001-05.com.equallogic:0-1cb196-06435a94f-b9bbc511ddb533ec-50gbtestvol

Performing a rescan of the host storage. Please wait...

Successfully updated host storage cache



Creating a partition on 50GBTestVol and formatting with NTFS
Script complete

Friday, April 4, 2014 10:26:06 AM

```
AuthenticationType      : NONE
InitiatorInstanceName   : ROOT\ISCSIPRT\0000_0
InitiatorNodeAddress    : iqn.1991-05.com.microsoft:tmer4r805s30.skynet.lab.local
InitiatorPortalAddress  : 0.0.0.0
InitiatorSideIdentifier : 400001370000
IsConnected             : True
IsDataDigest            : False
IsDiscovered            : True
IsHeaderDigest          : False
IsPersistent            : False
NumberOfConnections     : 1
SessionIdentifier       : fffffe00000dca430-40000137000000919
TargetNodeAddress       : iqn.2001-05.com.equallogic:0-1cb196-06435a94f-b9bbc511ddb533ec-
50gbtestvol
TargetSideIdentifier    : 2409
PSComputerName          :
```

```
DriveLetter      : J
DriveType        : Fixed
FileSystem       : NTFS
FileSystemLabel  : 50GBTestVol
HealthStatus     : Healthy
ObjectId        : \\?\Volume{b969ed7b-3289-41f6-ac4b-034b450c00f4}\
Path            : \\?\Volume{b969ed7b-3289-41f6-ac4b-034b450c00f4}\
Size            : 52302966784
SizeRemaining    : 52197855232
PSComputerName  :
```

.NOTES

This script must be run with administrator privileges.
#>

#####

#Import required modules

```
Set-StrictMode -Version Latest
$ErrorActionPreference = "Stop"
```

cls

```
import-module -name "c:\program files\equallogic\bin\EqlPSTools.dll"
Import-Module -Name iSCSI
set-executionpolicy remotesigned -force
```



```
#####
#Verify that EQL services are running

Write-Host "Checking to see if required EqualLogic services are running`n"

Get-Service | Where-Object {$_.displayname -like "Equal*"}

foreach ($svc in Get-Service | Where-Object {$_.displayname -like "Equal*"}) {
    if($svc.Status -eq "Stopped") {
        Write-Host "`nRestarting..."
        write-host $svc.DisplayName
        Start-Service $svc.name
    }
}

Write-Host "`nAll EqualLogic services started`n"

#####
#Searching for available EQL Stroage Pools

Write-Host "Searching for EqualLogic Storage Pools...."

if (get-eqlpool) {
    write-host "`nThe following EqualLogic storage pools were found:"

    get-eqlpool | ft -Property storagepoolname,groupname,groupaddress,FreespaceMB -
    AutoSize

} else {
    write-host "No EqualLogic storage pools found"
    break
}

#####
#Get key user input and store in variables

$eqlpoolname = read-host "Please select a storage pool name from the above list"
$volumename = read-host "Please type a name for the volume to be created"
[int64]$volumesize = read-host "Please select a size for the volume in MB"
$makethin = read-host "Use thin provisioning for volume? (Yes/No)"

[int64]$eqlpoolfreespace = (get-eqlpool -StoragePoolName $eqlpoolname).freespacemb
$eqlgroupname = (get-eqlpool -StoragePoolName $eqlpoolname).groupname
$eqlgroupaddress = (get-eqlgroupconfiguration).groupaddress

#####
#Verify that enough free space exists in EQL pool to create the volume

if ($volumesize -gt $eqlpoolfreespace) {
    write-host "Not enough freespace in pool for volume"
    break
}

```



```
#####
#Get IQN of host vm or server

$iqn = (get-initiatorport | where {$_.portaddress -like '*ISCSI*'}).nodeaddress
Write-Host "This hosts initiator is: $iqn`n`n"

#####
#Creating new EQL Volume

Write-Host "Creating a new $volumesize MB EqualLogic volume called $volumename`n"
New-EqlVolume -VolumeName $volumename -StoragePoolName $eqlpoolname -VolumeSizeMB
$volumesize -ThinProvision $makethin -loadbalance true -AccessType read_write -PassThru
| New-EqlVolumeAcl -InitiatorName $iqn -AclTargetType volume_and_snapshot

get-eqlvolume | where-object {$_.volumename -eq $volumename} | ft -property
Volumename,volumesizeMB,Groupname,storagepoolname,hostingmembers,sessionid -AutoSize

#####
#Performing iSCSI initiator work, update, connect, and register

get-date
Write-Host "Updating iscsi targets on host. Please wait..."
Update-IscsiTargetPortal -TargetPortalAddress $eqlgroupaddress
#Update-IscsiTarget
Write-Host "Finished updating iscsi targets on host.`n"

$iscsitarget = (Get-IscsiTarget | Where-Object -Filter {$_.nodeaddress -like
"*$volumename"}).nodeaddress
write-host "The iscsi target for $volumename is: $iscsitarget"

Connect-IscsiTarget -NodeAddress $iscsitarget
Write-Host "Successfully connected to iscsitarget $iscsitarget`n"

$iscsisession = (get-iscsitarget | where-object {$_.nodeaddress -eq $iscsitarget} | Get-
IscsiSession).SessionIdentifier
Register-IscsiSession -SessionIdentifier $iscsisession
Write-Host "Successfully registered iscsitarget $iscsitarget`n"

#####
#Performing a host storage rescan

Write-Host "Performing a rescan of the host storage. Please wait...`n"

Update-StorageProviderCache -StorageSubSystem (Get-StorageSubSystem -friendlyname
"$eqlgroupname") -DiscoveryLevel Full
Update-HostStorageCache

Write-Host "Successfully updated host storage cache`n"

#####
#Initailizing, creating a new partition, assign drive letter, and format new EQL volume
```



```

Write-Host "`nCreating a partion on $volumename and formatting with NTFS`n"

$disknumber = (get-iscsitarget | where-object {$_.NodeAddress -like "**$volumename"} | Get-
Iscsisession | get-disk).Number
Initialize-Disk -number $disknumber -PassThru | New-Partition -AssignDriveLetter -
UseMaximumSize | Format-Volume -FileSystem NTFS -newfilesystemlabel $volumename -
Confirm:$false

#####
#End of script

Write-Host "`nScript complete`n"

```



D Perform a Hyper-V storage migration for a VM to a new EqualLogic iSCSI storage device

One of the premier enhancements in Server 2012 Hyper-V is Hyper-V virtual machine mobility particularly in the areas of Live Migration and Storage Migration. A detailed discussion on this subject goes beyond the scope of this paper. There has been a significant amount of documentation written by Microsoft about Hyper-V Live Migration and Hyper-V Storage Migration on the Microsoft TechNet site at (<http://technet.microsoft.com>). This paper provides answers to why and how an administrator would perform a Hyper-V virtual machine storage migration from an existing EqualLogic volume (source device) to a new EqualLogic volume (target device).

There are many reasons why an administrator would want to perform a storage migration on a running virtual machine. Some of the reasons are:

- The virtual machine is sharing the volume with other virtual machines and is seeing performance issues due to storage I/O contention with the other VMs.
- The administrator wants to dynamically expand the virtual machine VHDX, but is limited due to physical free space restrictions on the existing volume.

Appendix section D.1 uses the example in this paper to demonstrate how to migrate the Management Virtual Machine storage to the new EqualLogic volume provisioned to the Hyper-V host in the previous section.

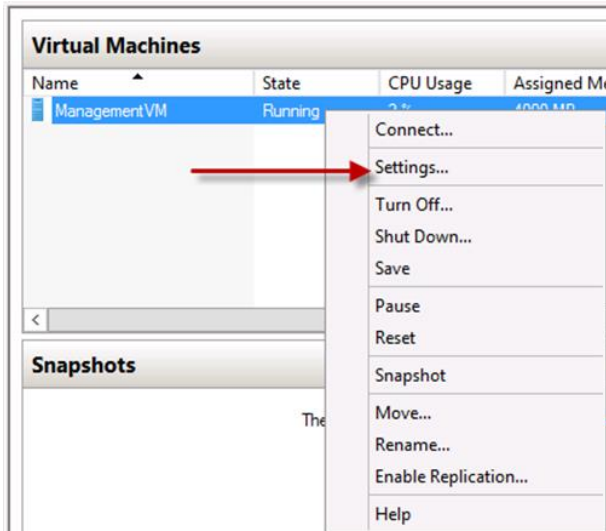
Note: The storage migration is performed using Hyper-V Manager from the Management Virtual Machine itself. A storage migration can be performed on a virtual machine while it is running as the migration moves the storage, not the virtual machine state.



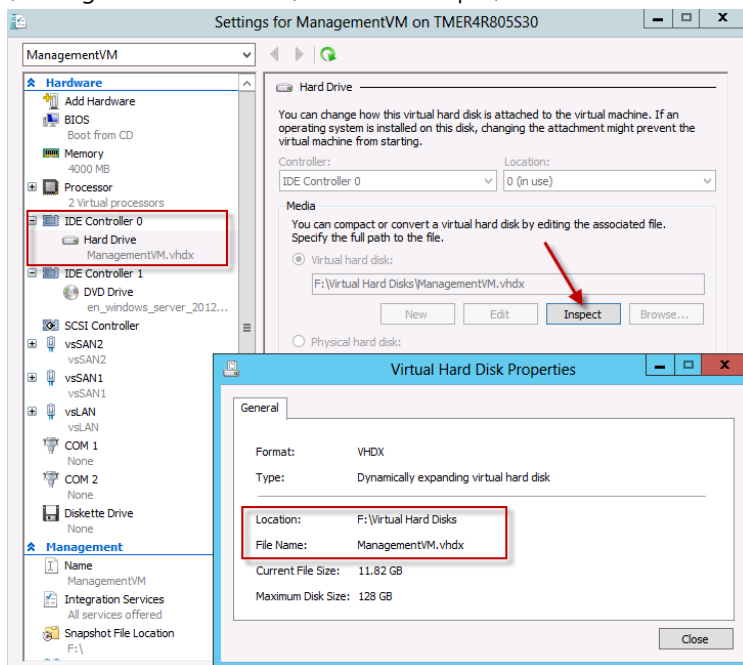
D.1 Using Hyper-V Manager

To perform a Hyper-V storage migration using Hyper-V Manager, use the following instructions.

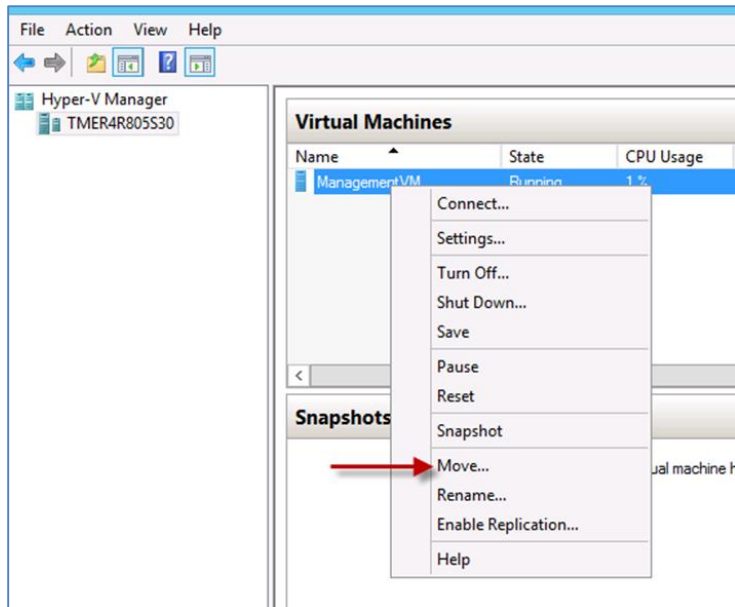
1. Review the location of the virtual machine virtual hard disk by right clicking the virtual machine and selecting **Settings** from the pop-up menu.



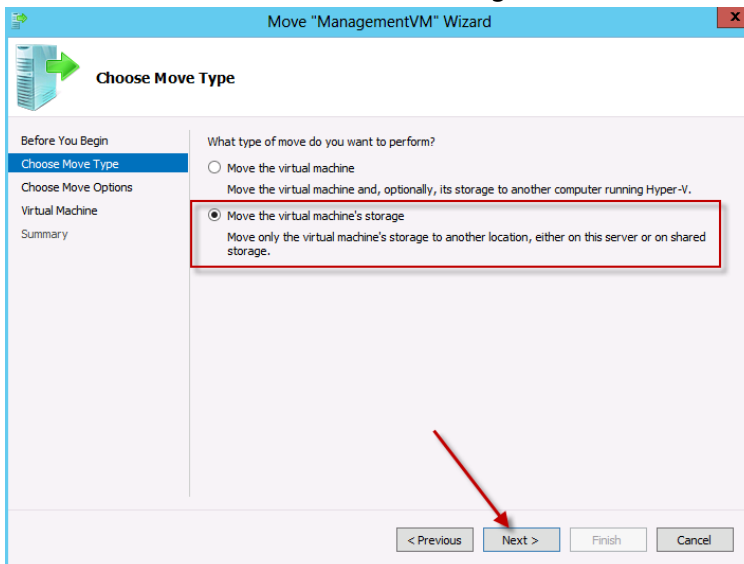
2. Examine the settings of the Management VM to verify the current location of the VHD (ManagementVM.VHDX). In the example, the location is F:\Virtual Hard Disks.



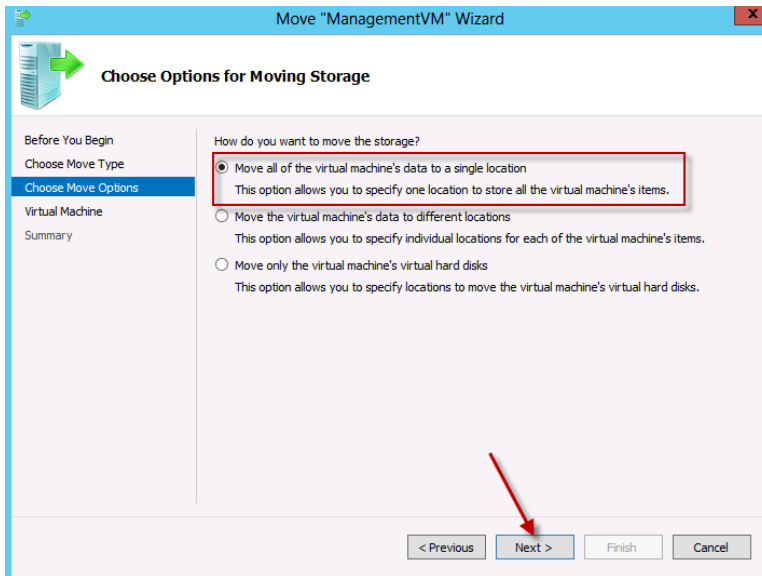
3. To launch the storage migration to the new EqualLogic iSCSI volume created and provisioned to the Hyper-V host in the last section (TempVol1), right click on the virtual machine in Hyper-V Manager and select **Move** from the pop-up menu.



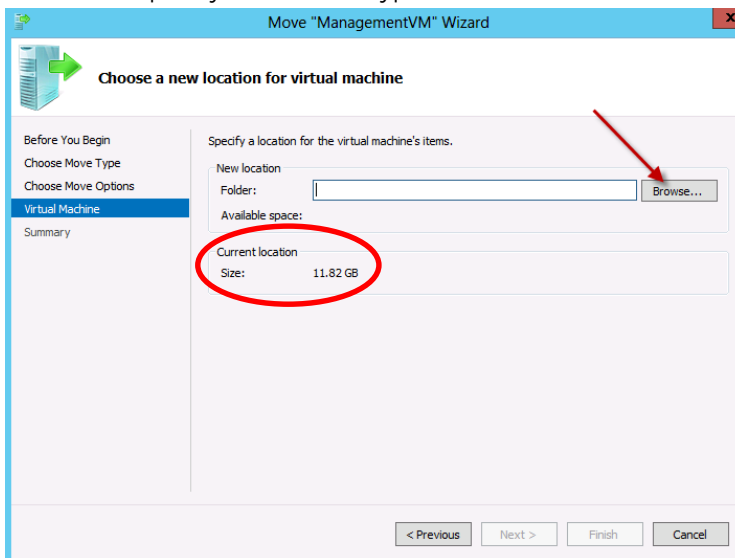
4. Select **Move the virtual machine's storage** in the wizard and click **Next**.



5. Select the storage move type
 - Move all of the virtual machines data (VHD and configuration file) to a single location
 - Move the data to different locations
 - Move only the VHD.In this example, **Move all of the virtual machine's data to a single location** is selected.

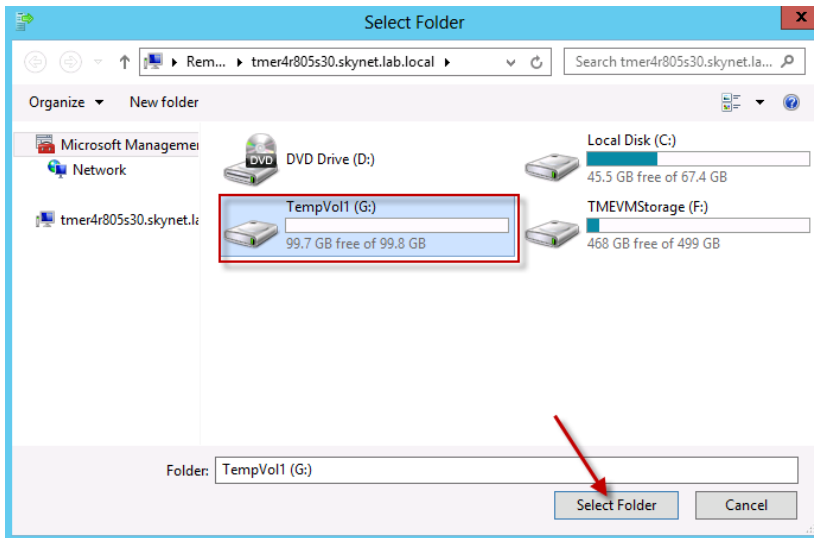


- Designate the new location for the virtual machine data. The amount of data for the virtual machine VHD and configuration file is shown on the screen. In this example, the total is 11.82 GB. This value is the amount of data that will be moved (combined total size of the VHD and configuration file). Click **Browse** to specify where the Hyper-V host will be moved.

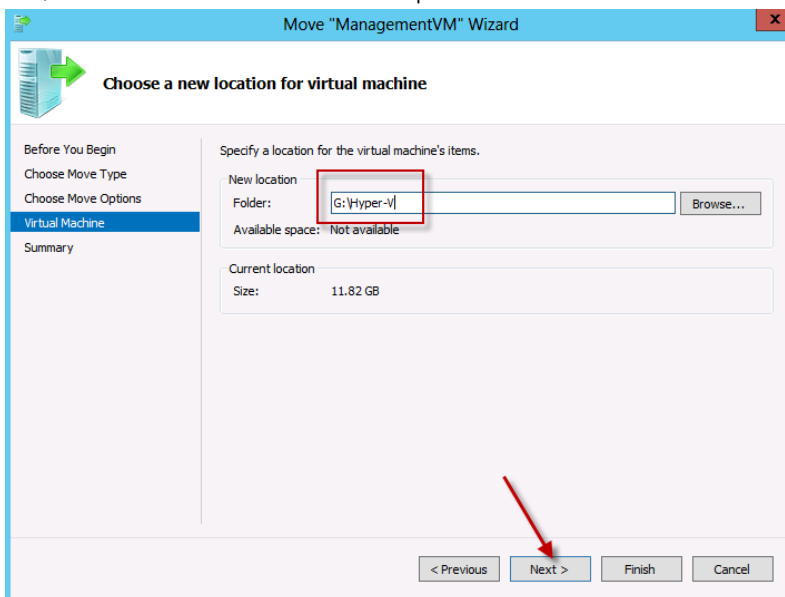


In the example, the new EqualLogic volume is TempVol1 which is mounted at G:\ on the Hyper-V host. This volume currently has 99.7 GB of free space which is more than enough for the 11.82 GB which will be moved.

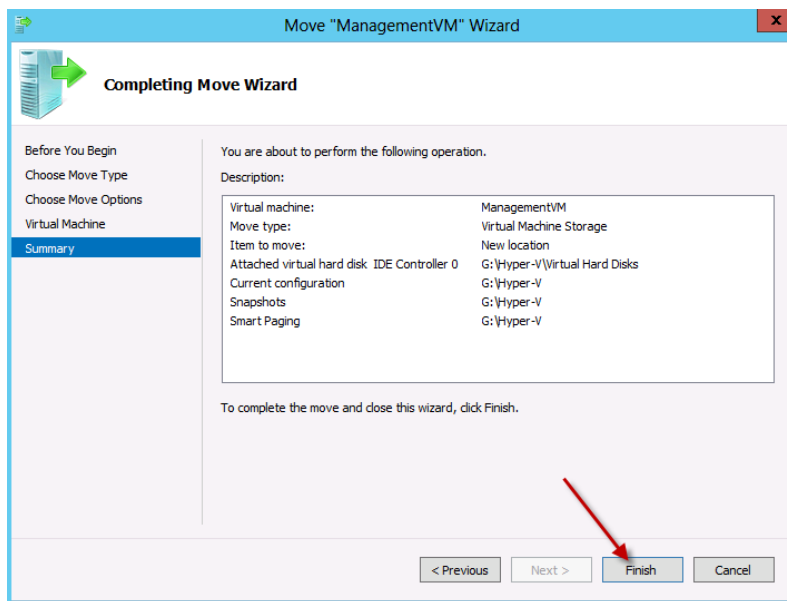
- Select the volume, and then click **Select Folder**.



8. Select or create a directory structure for the final location of the virtual machine data. In the example, the parent directory (labeled "Hyper-V") has been added. Since this parent directory does not exist on G:\, it will be created in the move process. Click **Next** to move to the wizard summary screen.

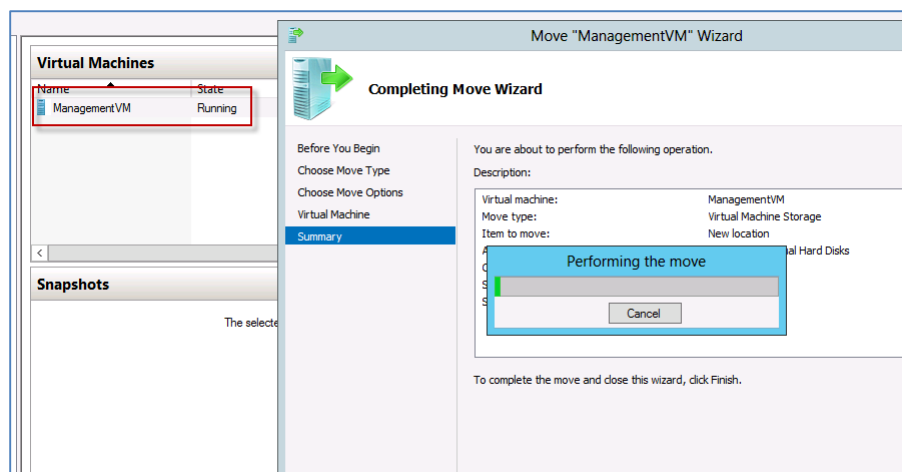


9. The wizard summary screen shows the destinations for the data. Review these locations and if acceptable, click **Finish** to start the storage move process.



Once the move starts, the virtual machine will still be online and running. The storage migration does not affect its virtual machine state because of the following workflow.

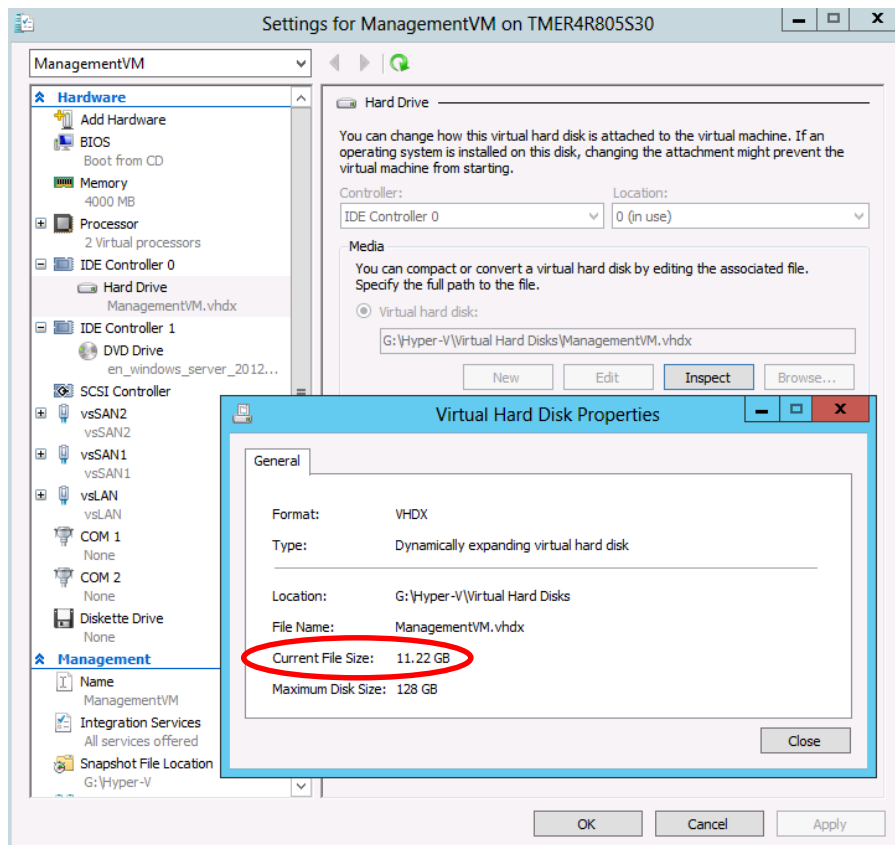
- The virtual machine read and write operations continue to go to the VHD on the source device (current location)
- The VHD is copied from the source storage device to the VHD location
- After the storage is copied, all write operations are mirrored to the source and destination locations
- After the source and destination storage are synchronized, access to the virtual machine VHD is transferred to the VHD on the destination device
- The VHD on the source device is then deleted



Note: In the example, the 11.82 GB of data is being copied from on EqualLogic volume to another EqualLogic volume within the same PS Series Array. This will invoke the Windows Offloaded Data

Transfer (ODX) mechanism on EqualLogic PS Series array which greatly enhances the speed of the data copy. ODX functionality in Windows Server 2012 maximizes the enterprise investment in EqualLogic storage arrays by enabling direct transfer data within the storage array itself, bypassing the host computer all together. By offloading the file transfer to the storage array, ODX minimizes latencies, maximizes array throughput, and reduces resource usage such as CPU and network consumption on the host computer. The result is higher data transfer rates and faster data copy/move times. For more information about ODX, go to the feature overview on Microsoft TechNet at <http://technet.microsoft.com/en-us/library/hh831628.aspx>.

10. Go back and re-examine the virtual machines storage settings. In the example, the Management VHD (ManagementVM.VHDX) is now shown in the G:\Hyper-V\Virtual Hard Disk directory. The Hyper-V parent directory was manually entered and the Virtual Hard Disk subdirectory was created as a default setting. If the Hyper-V parent directory was not specified, then the parent directory would be G:\Virtual Hard Disk. The target parent directory for a virtual machine storage move is defined by the administrator.



Note: in the previous screen shot, the **Current File Size** is 11.22 GB. This reflects the size of the Management VM VHD (ManagementVM.VHDX) in the created G:\Hyper-V\Virtual Hard Disk\ directory. The other Hyper-V virtual machine files - config (.XML), snapshot place holder (.bin), and the saved state file (.vsv) are all stored in the G:\Hyper-V\Virtual Machines\ directory. Running the **ls -r** command at G:\Hyper-V on the Hyper-V host returns this structure in the output



```
[TME4R805S30] PS G:\Hyper-V> ls -r
```

Directory: G:\Hyper-V

| Mode | LastWriteTime | Length | Name |
|-------|---------------------|--------|--------------------|
| d---- | 12/31/2013 10:30 AM | | Snapshots |
| d---- | 12/31/2013 10:31 AM | | Virtual Hard Disks |
| d---- | 12/31/2013 10:32 AM | | Virtual Machines |

Directory: G:\Hyper-V\Virtual Hard Disks

| Mode | LastWriteTime | Length | Name |
|--------|--------------------|------------|-------------------|
| -a---- | 12/31/2013 2:21 PM | 1205023539 | ManagementVM.vhdx |

Directory: G:\Hyper-V\Virtual Machines

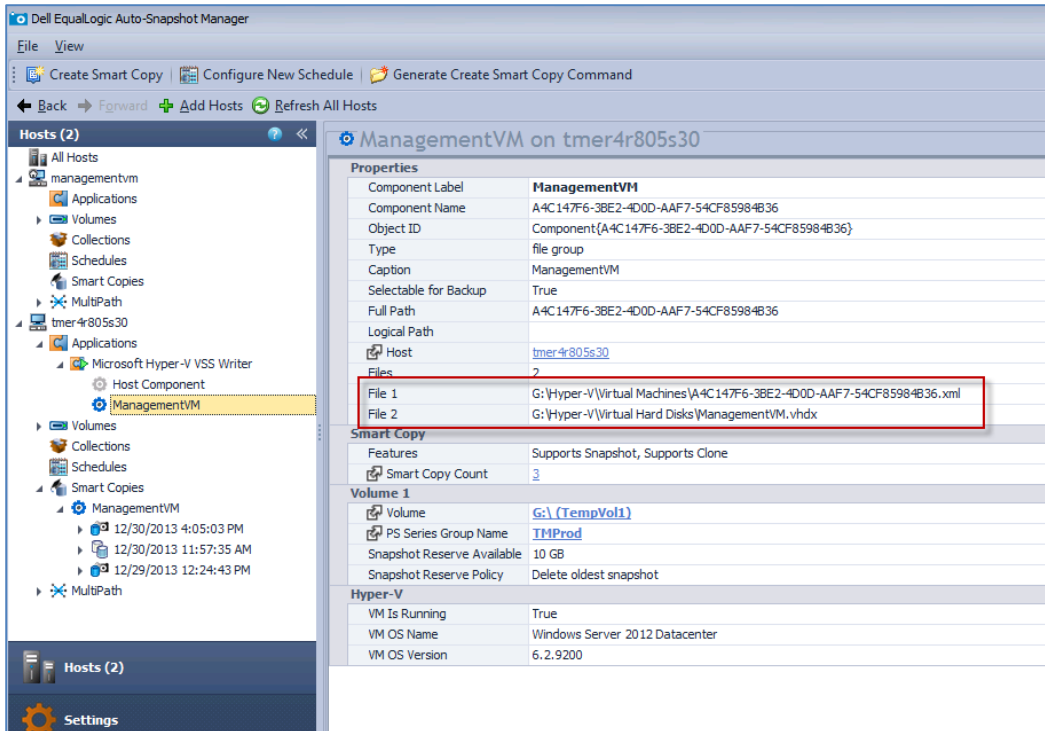
| Mode | LastWriteTime | Length | Name |
|--------|---------------------|--------|--|
| d---- | 12/31/2013 12:19 PM | | A4C147F6-3BE2-4D0D-AAF7-54CF85984B36 |
| -a---- | 12/31/2013 12:19 PM | 40702 | A4C147F6-3BE2-4D0D-AAF7-54CF85984B36.xml |

Directory: G:\Hyper-V\Virtual Machines\A4C147F6-3BE2-4D0D-AAF7-54CF85984B36

| Mode | LastWriteTime | Length | Name |
|--------|---------------------|----------|--|
| -a---- | 12/31/2013 12:19 PM | 0 | A4C147F6-3BE2-4D0D-AAF7-54CF85984B36.bin |
| -a---- | 12/31/2013 12:19 PM | 20971520 | A4C147F6-3BE2-4D0D-AAF7-54CF85984B36.vsv |

After the storage move completes, the ASM/ME application is automatically updated by the Hyper-V VSS Writer service with the new locations for ManagementVM's VHD and files. This is reflected in the properties window of the ManagementVM virtual machine in ASM/ME.

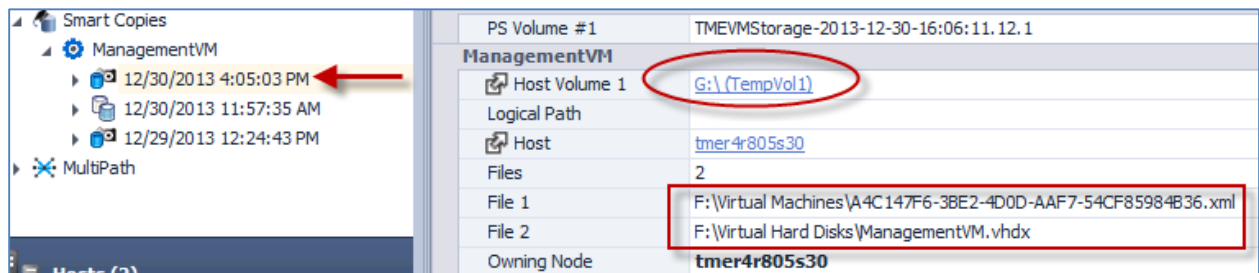




The above ASM/ME screen capture shows the new location of the ManagementVM configuration file and VHD file as G:\Hyper-V\Virtual Machines and G:\Hyper-V\Virtual Hard Disk respectively.

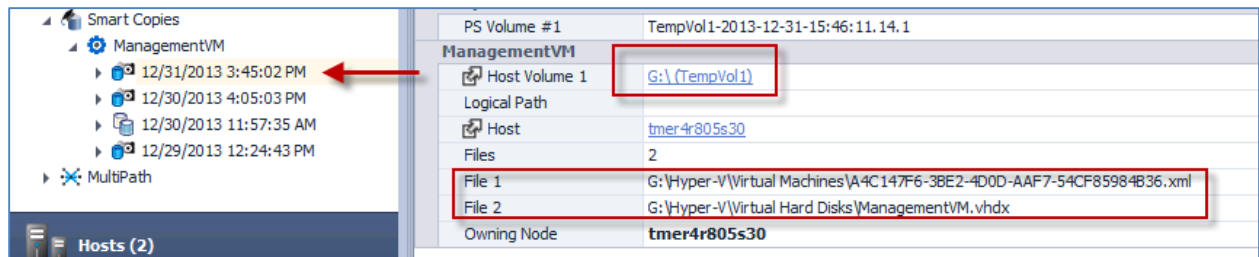
What about any existing ASM/ME Smart Copies associated with the virtual machine?

Any associated virtual machine Smart Copies (both Snapshots and Clone) created before the virtual machine storage move are updated by the Hyper-V VSS Writer to reflect the new restore target location (Host Volume) for the virtual machine storage components as shown below.

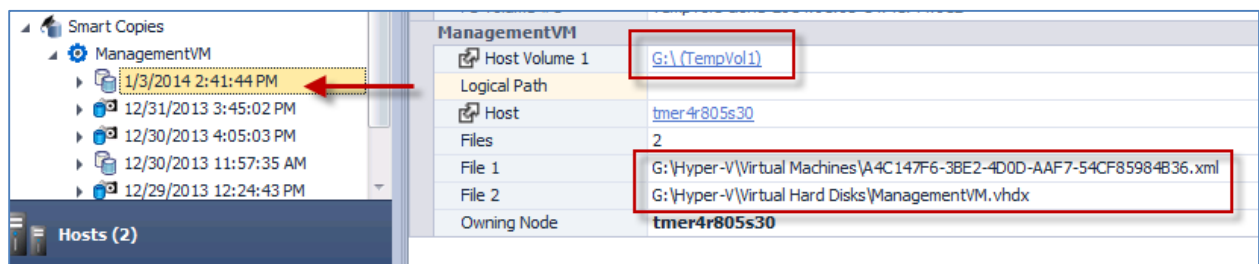


If the particular pre-move Smart Copy was used to restore the virtual machine, the restore operation would use the snapshot data from the snapshot reserve space associated with the original volume (F:\) as a source, or the clone volume if the Smart Copy was a clone. A restore operation using the pre-move Smart Copies will not relocate the virtual machine storage back to the original storage location or create two copies of the virtual machine. It will restore the VM to the state it was in when its VHD and configuration file were located on the original volume locations (F:\Virtual Machines and F:\Virtual Hard Disks). This is what is reflected by the File 1 and File 2 locations in the above output.

Any new (post move) Smart Copies will reflect the restore state of the virtual machine at its new location (G:\Hyper-V\Virtual Machines and G:\Hyper-V\Virtual Hard Disks). This is shown for a post move Smart Copy snapshot below



And a post move Smart Copy clone.



Finally, update any Hyper-V environment configuration documentation (diagrams and /or tables) with information about the new location of the ManagementVM.VHDX being on the newly provisioned volume (G:\) on the Hyper-V host.

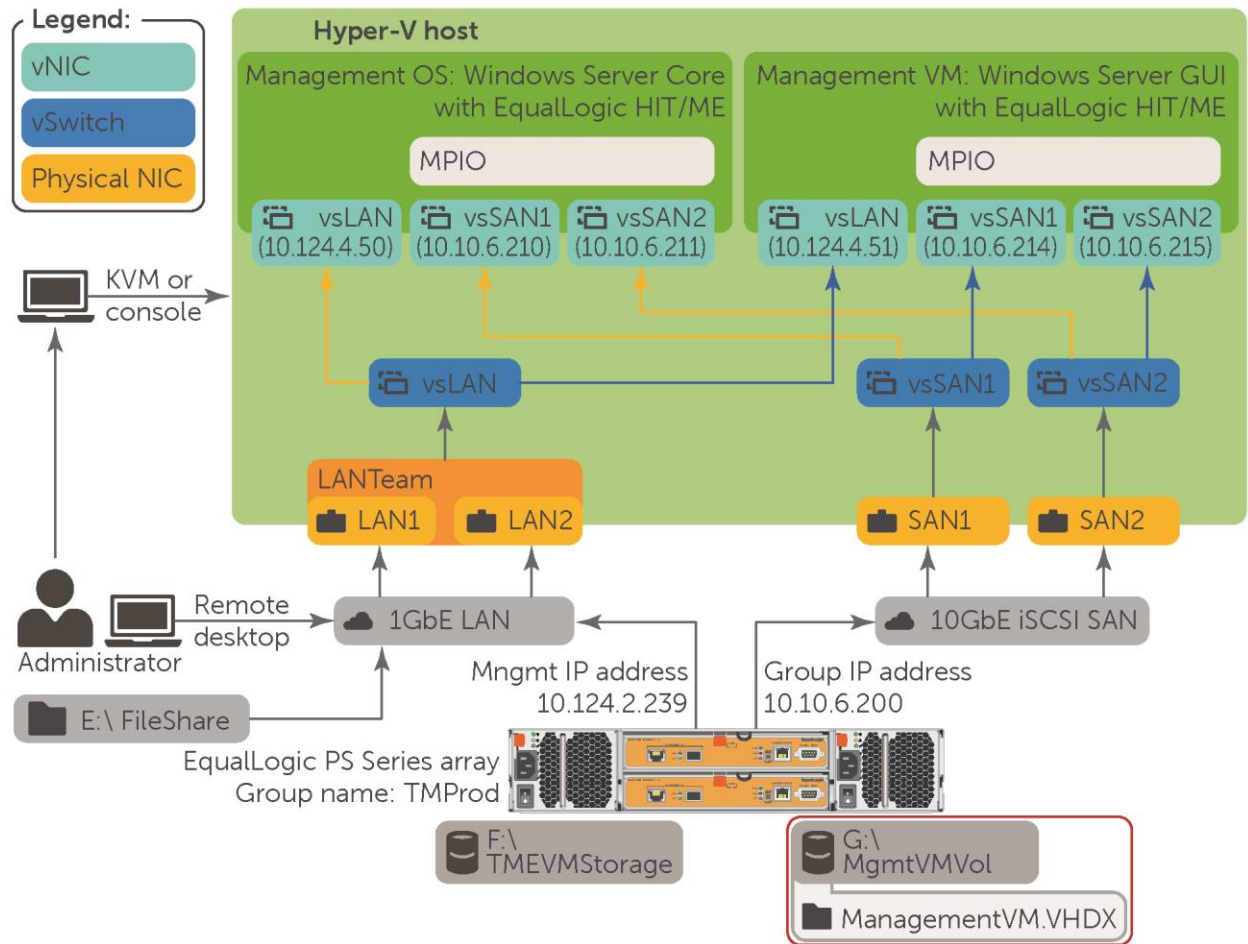


Figure 10 Final Hyper-V environment configuration

D.2 Using PowerShell

The entire virtual machine storage move described in the above example could have been accomplished with the PowerShell **move-vmstorage** command run from the Hyper-V host.

```
[TMER4R805S30] PS G:\>move-vmstorage "ManagementVM" -DestinationStoragePath
G:\Hyper-V
```

This command moves all of the files associated with the ManagementVM virtual machine to the G:\Hyper-V directory and creates the same parent and default subdirectory structure.