White Paper

Windows Server Failover Clustering (WSFC) on SimpliVity

Table of Contents

Introduction

Microsoft Windows Clustering Service, is Microsoft's traditional method of providing high availability for applications and services running on servers. Failover clustering is a role in Microsoft Windows Server operating system that allows clustering of multiple servers to provide application resiliency. Clustered servers use a shared disk between the servers so that even if one server experiences a planned or unplanned outage, another server in the cluster still has access to the data.

Even though Microsoft's High Availability (HA) designs today are moving to multiple copies of the data (DAG and AlwaysOn), there may be scenarios where Windows Server Failover Clustering technology is the preferred option. This White Paper provides details on how to implement and operate a Windows Failover Cluster on SimpliVity OmniStack systems.

Objective

The objective of this paper is to provide information and guidance on using Windows Server Failover Clustering within virtual machines (VMs) on SimpliVity hyperconverged infrastructure. Implementation of the cluster, HA scenarios of services running on the VMs and operational best practices are the main focus of the paper.

Audience

This paper is intended for Solution Architects and IT professionals who are looking to operate and protect applications running on VMs using Windows Server Failover Clustering on SimpliVity hyperconverged infrastructure.

Scope

With the main focus being high availability of services, this White Paper also covers implementation and configuration details for implementing Windows Failover clustering on SimpliVity hyperconverged infrastructure. Additionally, best practices and recommendations are provided in the document. For the purpose of explanation, this paper uses vCenter Database Availability on Microsoft SQL Server using Windows Server Failover Clustering as an example.

SimpliVity OmniStack

As an early pioneer in hyperconvergence and a recognized market and technology leader, SimpliVity offers the industry's most completed hyperconverged infrastructure platform. The SimpliVity solution goes beyond compute, storage, and storage networking to converge all IT infrastructure and advanced data services for virtualized workloads - including data efficiency, data protection, management and mobility - onto x86 servers.

SimpliVity hyperconverged infrastructure provides a single, shared resource pool across the entire IT stack, eliminating point products and inefficient siloed IT architectures.

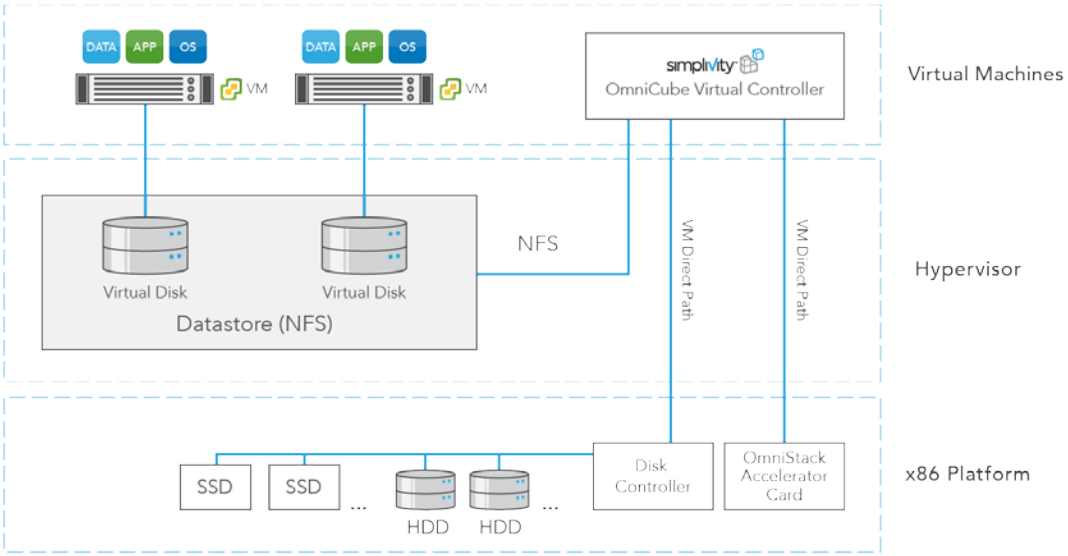
OmniStack Data Virtualization Platform

SimpliVity's foundational technology is the OmniStack Data Virtualization Platform, designed from the ground up for virtualized workloads. The OmniStack Data Virtualization Platform provides three key benefits to end-users:

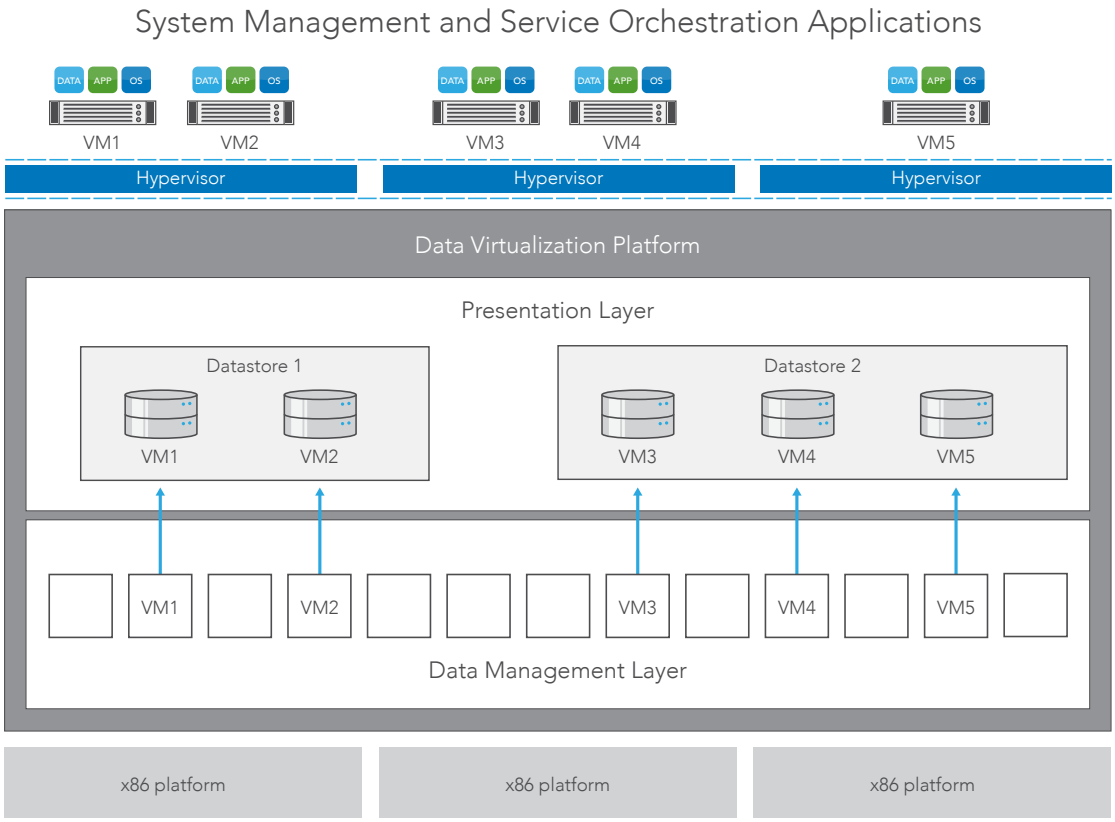
1. Guaranteed Data Efficiency - Deduplicates, compresses, and optimizes all data globally, improving performance, guaranteeing 90% capacity across storage and backup.
2. Built-in Resiliency and Data Protection - Delivers the most resilient hyperconverged infrastructure platform, including built-in backup and replication that eliminates the use of legacy solutions.
3. Global VM-Centric Management and Mobility - Delivers policy-based, VM-centric management to simplify day-to-day operations and enable data mobility, increasing productivity.

An individual OmniStack node includes:

1. A compact hardware platform - a 2U industry-standard virtualized x86 platform containing compute, memory, performance-optimized SSDs and capacity-optimized HDDs protected in hardware RAID configurations, and 10GbE network interfaces.
2. A hypervisor such as VMware vSphere/ESXi.
3. OmniStack virtual controller software running on the hypervisor.
4. An OmniStack Accelerator Card – a special-purpose PCIe card with an FPGA, flash, and DRAM, protected with super capacitors; the accelerator card offloads CPU-intensive functions such as data compression, deduplication and optimi- zation from the x86 processors.



SimpliVity OmniStack is a software-defined hyperconverged infrastructure solution. Clustering multiple OmniStack- powered hyperconverged infrastructure units forms a shared resource pool and delivers high availability, mobility, and efficient scaling of performance and capacity.



Windows Failover Clustering

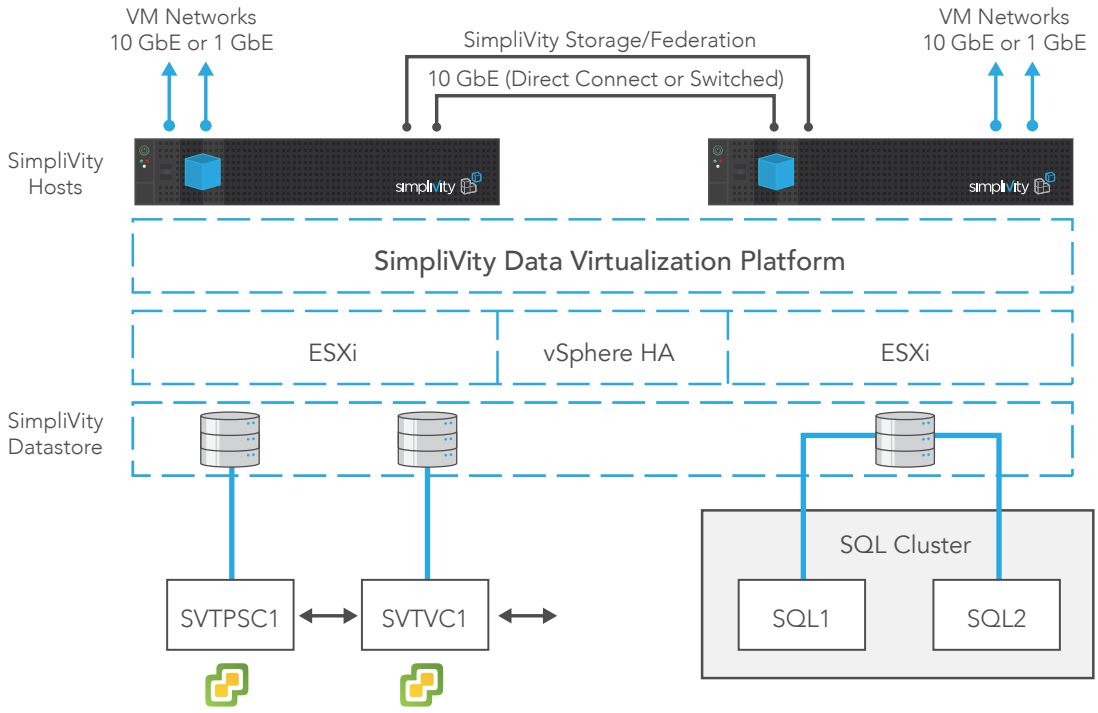
Clustering involves using two or more separate physical or virtual servers to create one logical server that is seen as the same to applications, with the members of the cluster (called nodes) able to monitor each and, if one of them goes down, its duties "fail over" to its partner without causing any disruption of service to users.

Windows Failover Cluster is a role available for install in the server distribution OS from Microsoft windows.

Solution Architecture

This section provides details about the test environment that was setup and used to validate the Windows Server Failover Clustering solution on SimpliVity. Architectural Diagram of the setup, overview of the environment, along with details of the software components used in the solution are provided in this section.

Architecture Diagram



Overview

To demonstrate the practicality of the solution, this paper uses SQL server as the application protected through Windows Server Failover Clustering. A vCenter server whose database is hosted on the SQL server is used as proof to ensure that SimpliVity supports Windows Server Failover Clustering and that the vCenter service is available through all High Availability test scenarios.

In the figure above, two OmniStack systems are hosted in a logical VMware datacenter. Two Windows Server 2012 R2 VMs running SQL server 2014 (SQL1 and SQL2) are hosted on the same OmniStack node. For shared storage, both VMs share a .vmdk that is attached to the SQL1 VM.

Another VM running vCenter Server (SVTVC1) is hosted on the second OmniStack node along with another VM running the VMware Platform Service Controller (SVTPSC1) which provides identity management for administrators and applications that interact with the vSphere platform.

Environment Setup

This sections highlights key configuration settings that were used to deploy the test environment. Configuration settings for the shared disks between the VMs, Windows Server Failover Clustering, SQL cluster and vCenter Server is provided below.

The following table summarized the technical details of the components used in this solution.

Server	Operating System	Application	VM Configuration
SVTPSC1.SVTPM.COM IP: 192.168.50.20	SUSE 11 Enterprise Linux Appliance	VMware Platform Service Controller 6.0u1b	vCPU: 2 Memory: 2 GB
SVTVC1.SVTPM.COM IP: 192.168.50.21	Windows 2012 R2 Updated and patched to current (9/30)	VMware vCenter Server 6.0u1b	vCPU: 4 Memory: 12 GB OS Disk: 100 GB
SQL1.SVTPM.COM IP: 192.168.50.22 HB IP: 172.28.10.1	Windows 2012 R2 Updated and patched to current (9/30)	Microsoft SQL Server Cluster Node SQL Server 2014 SP2	vCPU: 2 Memory: 4 GB OS Disk: 40 GB
SQL2.SVTPM.COM IP: 192.168.50.23 HB IP: 172.28.10.2	Windows 2012 R2 Updated and patched to current (9/30)	Microsoft SQL Server Cluster Node SQL Server 2014 SP2	vCPU: 2 Memory: 4 GB OS Disk: 40 GB

Shared Disk Setup

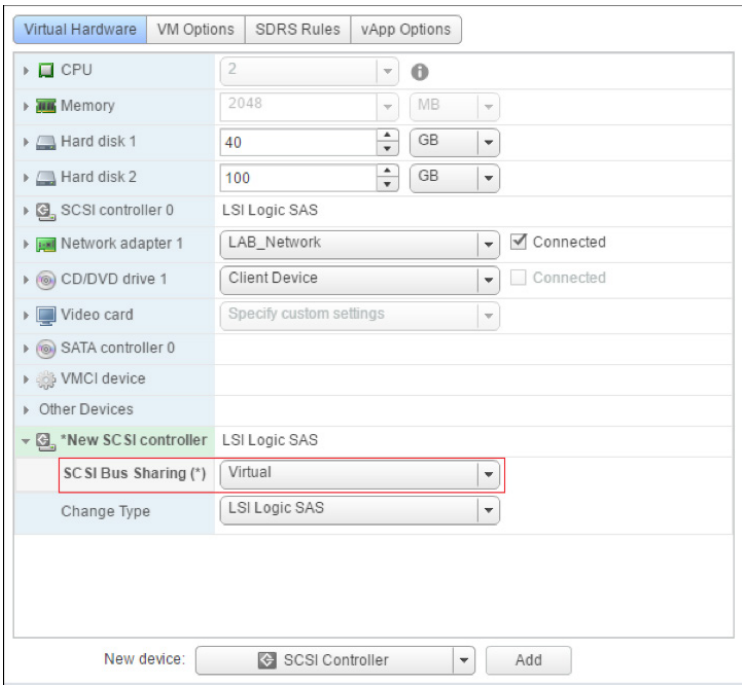
Two disks were created for this test that were shared between the two SQL Server VMs.

1. Quorum Disk
2. Data Disk

Both disks were first created on VM SQL1 and then shared with VM SQL2. The following Virtual SCSI Controller options were applied to create the shared disks:

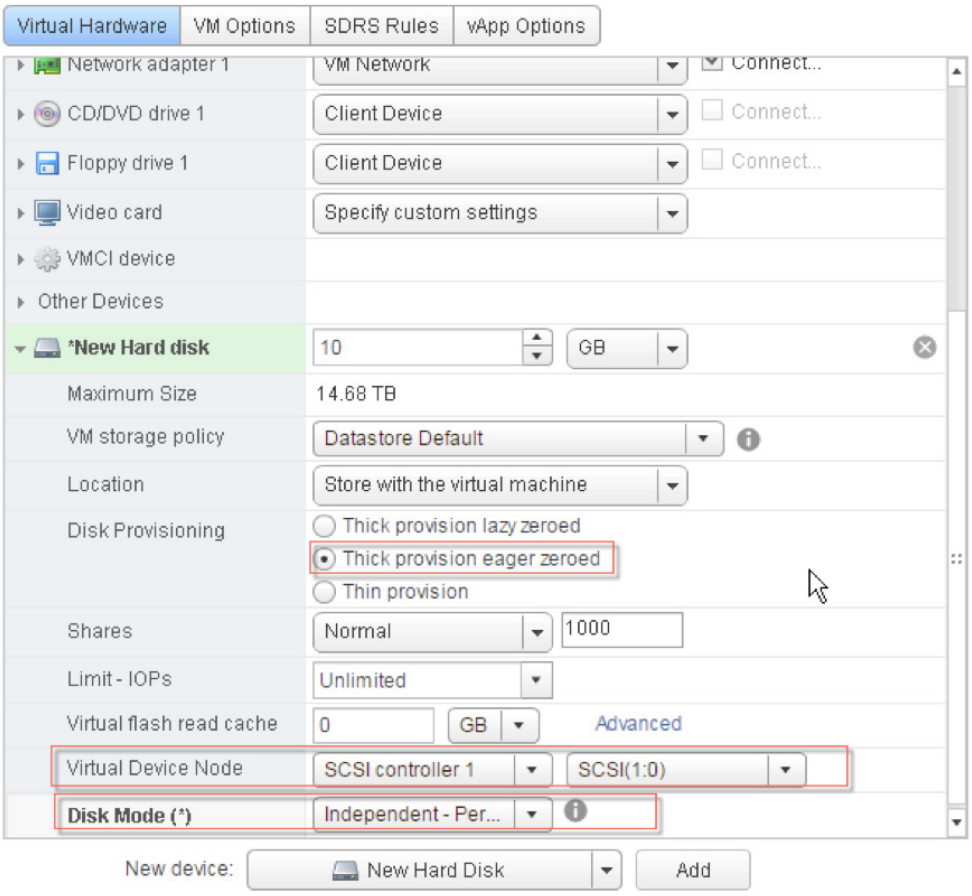
Virtual SCSI Controller for shared disks (Quorum and Data):

1. SCSI Bus Sharing: Virtual
2. Type: LSI Logic SAS



The following settings were used for each of the shared disks

1. Quorum Disk: SCSI 1:0
    a. Disk Provisioning: Thick provision lazy/eager zero
    b. Disk mode: Independent Persistent
    c. Size: 10 GB
    d. Volume: Q:
    e. Allocation Unit: 8192
2. Data Disk: SCSI 1:1
    a. Disk Provisioning: Thick provision lazy/eager zero
    b. Disk mode: Independent Persistent
    c. Size: 100 GB
    d. Volume: S:
    e. Allocation Unit: 64k



Note: All virtual disks (OS, Quorum, Data) are provisioned as Thick provision eager zeroed.

Windows Server Failover Cluster Setup

The Windows Failover Cluster was setup using the following cluster name and IP.

Cluster Name: MSCLS1

Cluster VIP: 192.168.50.24

Cluster Settings

1. Cluster Members: HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Disk\TimeOutValue = 60 (Default Setting in Windows Server 2012 R2)
2. Set Cluster Heartbeat Threshold: 11
   https://blogs.msdn.microsoft.com/clustering/2012/11/21/tuning-failover-cluster-network-thresholds/

For more information on creating a Windows Failover Cluster, refer to the following link:
https://technet.microsoft.com/en-us/library/cc730692(v=ws.11).aspx

Note: In Windows 2012 and 2012 R2, cluster validation completes with this warning: Validate Storage Spaces Persistent Reservation. You can safely ignore this warning.
https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1037959

SQL Cluster Setup

The SQL Server Cluster was setup using the following cluster name and IP

Cluster Name: SQLCLS1
Cluster VIP: 192.168.50.25

For more information on creating a SQL Server Cluster, refer to: the following link:
https://msdn.microsoft.com/en-us/library/ms179530.aspx

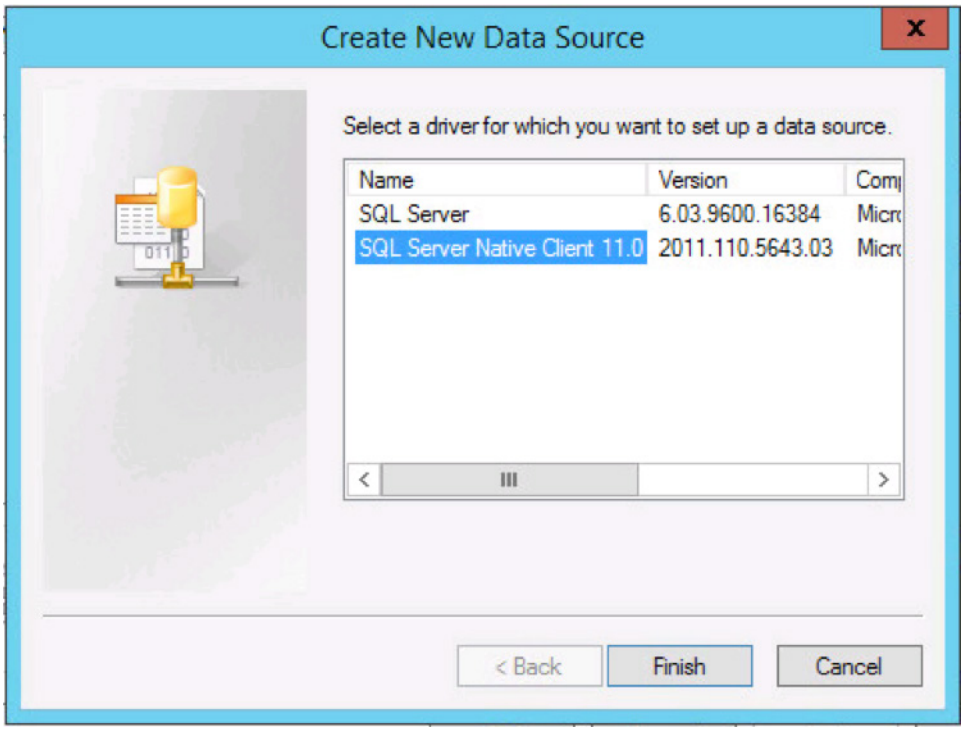
Additional settings

1. SQL Server 2014 SP2 Program installed on C:\.
2. Windows Firewall and Advanced Security has been disabled on cluster members.

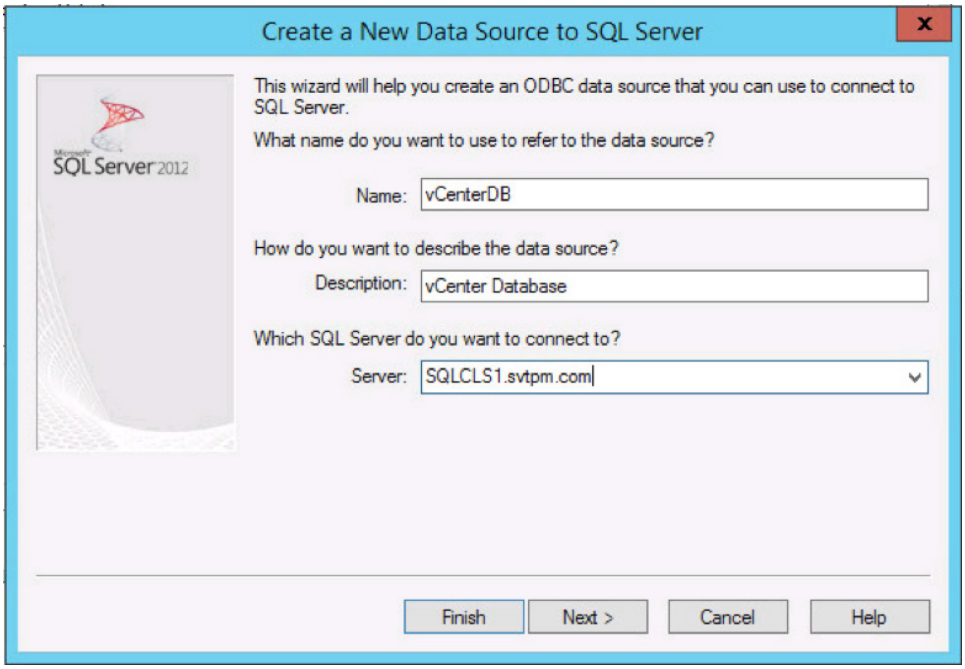
vCenter Setup

A 64-bit Data Source Name (DSN) is required while installing the vCenter Server. The following screenshots highlight the key settings used for creating the DSN.

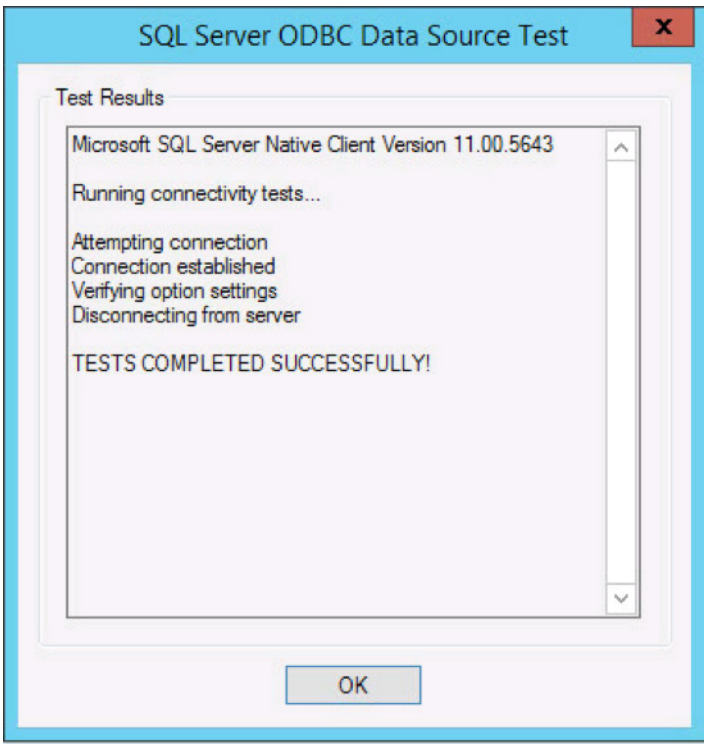
1. Select the SQL Server Native Client Driver



2. Use the Cluster Name for the Server:



3. Complete ODBC configuration for vCenter Database and test the data source:



Note: From the vCenter Server, open the vpxd.cfg file and set recoverForAllError set to True
https://kb.vmware.com/selfservice/search.do?cmd=displayKC&docType=kc&docTypeID=DT_KB_1_1&externalId=2059560

Solution Testing and Results

This section details the testing that was done to validate Microsoft Clustering Services on SimpliVity. Detailed test cases, test methodology and results are provided in this section.

Test cases

In order to validate the solution, the following use cases were tested

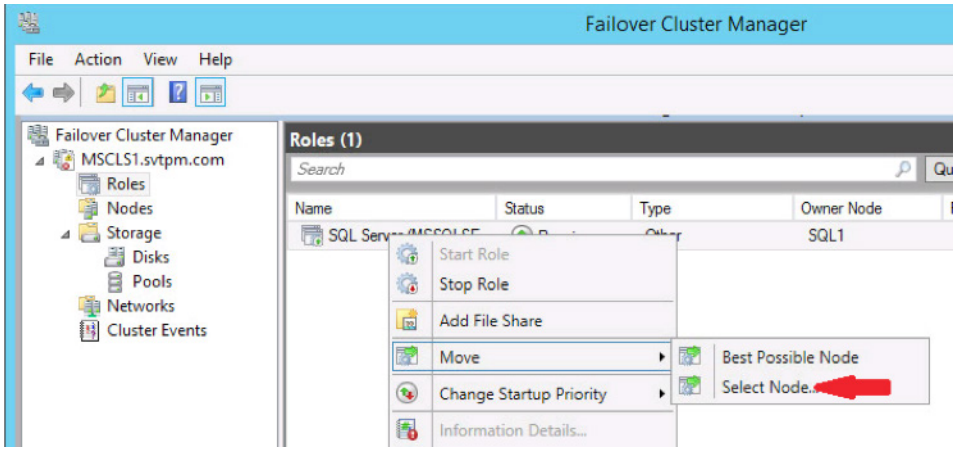
1. **SQL Cluster Failover/Failback (Planned)** – To ensure that the SQL Server role can be successfully failed over (or failed back) in case of planned maintenance work on one of the SQL Server VMs.
2. **SQL Cluster Failover/Failback (Unplanned)** – To ensure that the SQL Server role fails over to the surviving node in case of an unplanned outage to one of the SQL Server VMs.
3. **vSphere HA Failover** – To verify that the SQL cluster can failover to the surviving OmniStack node in case of a node failure.
4. **SimpliVity OmniStack Virtual Controller (OVC) Failover/Failback** – In case of failure of the OmniStack Virtual Controller verify that there is automatic IP failover of the OVC with no application downtime.
5. **SimpliVity Backups** – Verify that the SQL server cluster can be backed up using native SimpliVity data protection features.
6. **SimpliVity Restore** – Verify that SQL server cluster can be restored from SimpliVity backups.
7. **vMotion of SQL Cluster** – To move the SQL cluster to another node in the SimpliVity datacenter in case you have a planned maintenance operation on the OmniStack system.

Test Methodology and Results

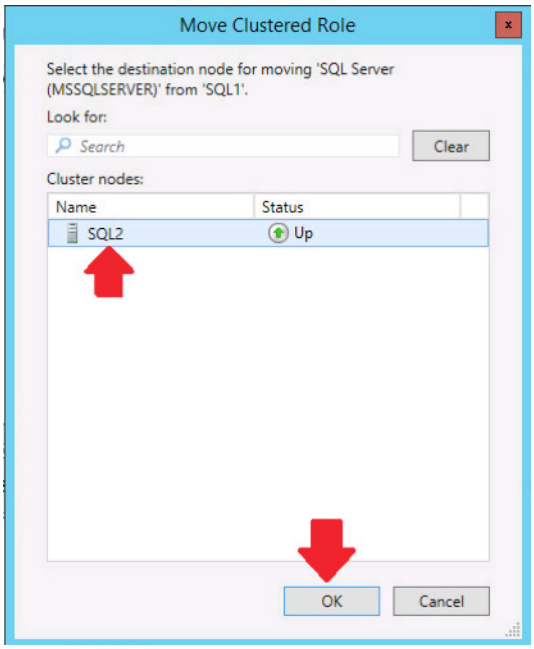
This section provides details on how various use cases were tested along with the results observed.

SQL Cluster Failover/Failback (Planned)

1. In Failover Cluster Manager -> Roles move the SQL Server Role to SQL2.



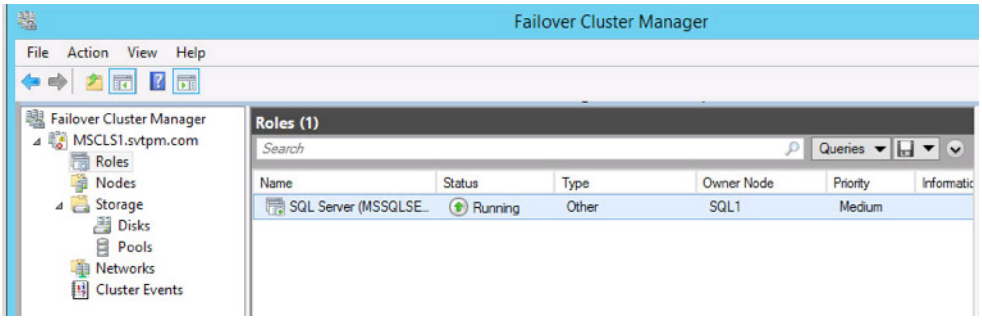
2. Select SQL2 and click OK.



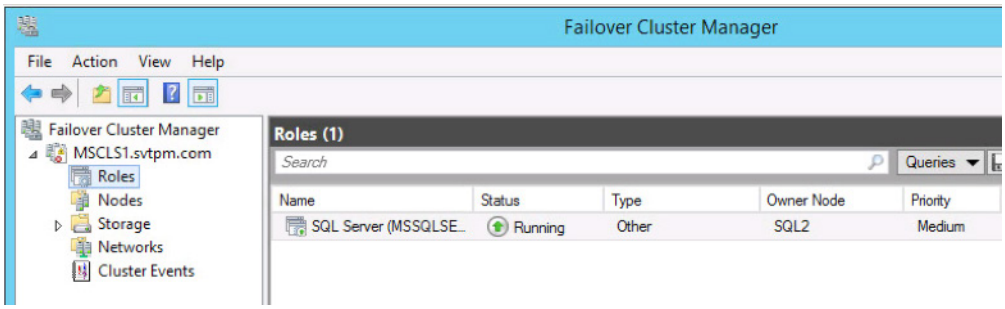
3. Verify connectivity to vCenter Database.
4. In Failover Cluster Manager -> Roles move the SQL Server Role to SQL1.
5. Verify connectivity to vCenter Database and vSphere Client.

SQL Cluster Failover/Failback (Unplanned)

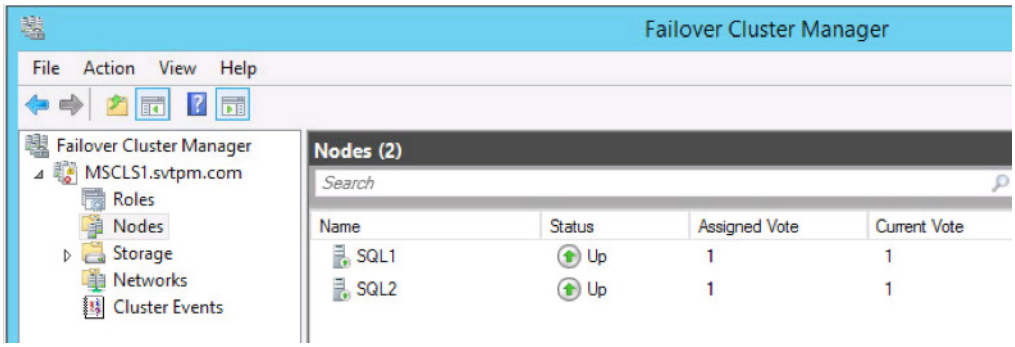
1. Verify SQL1 is the Owner Node for the SQL Server Role.



2. Using vSphere Client to Power Off SQL1.
3. SQL2 should become the Owner Node of the SQL Server Role.



4. Verify Connectivity to vCenter Server Database and vSphere Client.
5. Power On SQL1.
6. Verify SQL1 is up in Failover Cluster Manager -> Nodes.



vSphere HA Failover

In this scenario, the host running the SQL cluster is reset to ensure connectivity to the vCenter database through vSphere HA.

1. Use IDRAC to reset power of SimpliVity host running the SQL Cluster.
2. The SQL cluster members are restarted on the surviving node.
3. vCenter was inaccessible for 5 minutes till the members came back online.
4. vSphere desktop client had to be reconnected to the server.
5. vSphere Web Client did not require any reconnect.

SimpliVity OmniStack Virtual Controller (OVC) Failover

In this scenario vCenter database availability is validated by powering off the OmniStack Virtual Controller.

1. Power off OVC on SimpliVity host running the SQL Cluster.
2. The surviving node handles the IP failover successfully.
3. Database connectivity is verified through vCenter Web UI and SQL Server Management Studio.
4. OVC is powered On.

SimpliVity Backups

Both SQL1 and SQL2 VMs are backed up using the SimpliVity backup feature for this test case.

1. VM that owns the data disk and quorum disk (SQL1) completes backup successfully.
2. VM that does not own the data disk and quorum disk (SQL2) completes as partial backup.

SimpliVity Restore

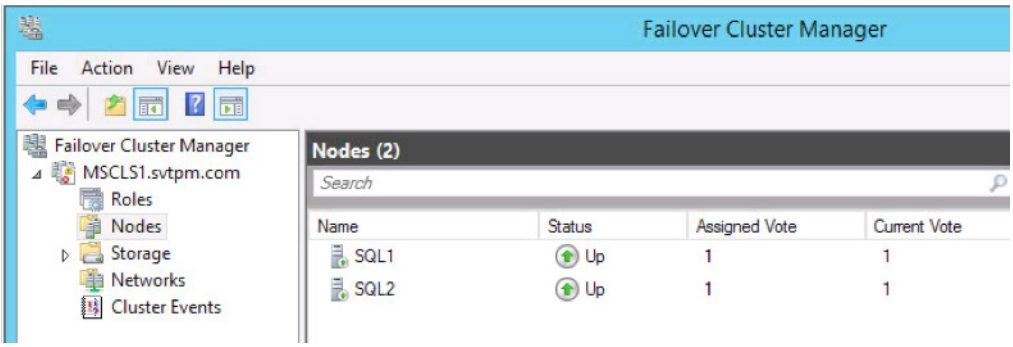
The SQL VMs that were backed up using the SimpliVity backup feature are restored and connectivity to the vCenter is re-established.

1. Shutdown both SQL VMs gracefully.
2. Note the name of the VM folder within the datastore that contains the data drive and quorum drive (SQL1 in this case).
3. Delete both SQL VMs.
4. Restore the VM that owns the data disk and quorum disk first.
5. Ensure vCenter is up and running again.
6. Restore the second VM (SQL2).
7. Before powering ON the second VM, rename the VM folder containing the data drive and quorum drive to match the name noted in step 2.
8. Power on the second VM.
9. FCM will list both nodes up and the SQL service as up and running.

vMotion of SQL Cluster

This is not supported and should only be used to move cluster nodes when performing maintenance on a host. vMotion is not available to powered on VMs using Virtual SCSI Bus Sharing.

1. Verify SQL1 is the Owner Node for the SQL Server Role.
2. Gracefully shutdown SQL2.
3. Use vMotion to Cold Migrate SQL2 to required host.
4. Power on SQL2 and verify SQL2 node is up in Failover Cluster Manager.



5. Move SQL Server Role to SQL2.
6. Gracefully shutdown SQL1.
7. Use vMotion to Cold Migrate SQL1 to required host (Same host as SQL2).
8. Power on SQL1 and verify SQL node is up in Failover Cluster Manager.

Summary of Test Plan and Results

The following table summarizes the use cases, test procedure, success criteria and test results as discussed in the previous sections.

Test	Test Procedure	Success Criteria	Result
1. SQL Cluster Failover/ Failback (Planned)	Use Failover Cluster Manager to Move SQL Clustered Role between nodes in the cluster. Verify vCenter Database connectivity.	vCenter Server is able to reconnect to vCenter Database.	vCenter accessible. Database accessible. Failover takes a few seconds to complete. No impact on operations.
2. SQL Cluster Failover/ Failback (Unplanned)	Power off cluster node with the SQL Server Role. Verify vCenter Database connectivity.	vCenter Server is able to reconnect to vCenter Database.	vCenter accessible. Database accessible. Failover takes a few seconds to complete. No impact on operations.
3. vSphere HA Failover	Use iDrac to reset power of SimpliVity host running the SQL Cluster. Verify vCenter Database connectivity.	SQL Cluster members restart on surviving SimpliVity node. vCenter Server is able to reconnect to vCenter Database.	SQL cluster members restart on surviving host. vCenter database was inaccessible for approximately 5 minutes. Once cluster members restarted databases accessible. Windows vSphere Client required reconnection to vCenter, vSphere Web Client recovered without reconnecting.

4. SimpliVity OmniStack Virtual Controller (OVC) Failover/Failback	Power off OVC on SimpliVity host running the SQL Cluster. Verify vCenter Database connectivity. Power on OVC.	OVC IP Address fails over to surviving OVC. Storage continues to be available.	IP Address failover successful. vCenter accessible. Database accessible. No impact on operations. OVC IP Address and Storage traffic failed back after power restored.
5. SimpliVity Backups	Backup SQL Cluster VMs using SimpliVity Backups.	Backups complete without failures.	Primary Cluster Node (SQL1) backed up successful. SQL SimpliVity backup pf SQL2 resulted in a warning "SimpliVity Partial VM Backup Detected."
6. SimpliVity restore	Restore SQL Cluster VMs.	Gracefully shut down SQL VMs and then delete them. Restore SQL1 first, rename folder name to match original name. Restore second VM	Restores are successful. Restoring and powering on SQL1 brings the vCenter service online again. Second VM is restored and powered on after changing the name of the VM folder for the first VM to match original name.
7. vMotion of SQL Cluster-in-a-Box	vMotion SQL Cluster Node VMs.	vMotion SQL Servers to another host in same datacenter.	Successfully moved SQL Cluster Node VMs to a new host. No impact on operations.

Conclusion

Windows Server Failover Clustering is a key technology from Microsoft leveraged by many applications for high availability. While there are newer technologies providing high availability at the application level, applications and customers may still rely on and want to use Microsoft clustering on SimpliVity.

This white paper provides details for implementing and operating a Windows Server Failover Cluster with SimpliVity hyperconverged infrastructure. A vCenter Server running on clustered SQL server VMs is used as an example to demonstrate the practicality of the solution.

Support for SimpliVity features along with high availability for Windows Server Failover Clustering is the main objective of the paper. Details around configuration settings for different components that are key to the validation of the solution are provided as well.

Several tasks and operations that IT administrators may use for managing Microsoft Failover Clusters on SimpliVity were tested successfully. High availability of the solution was validated successfully by simulating planned and unplanned failures for SimpliVity OmniStack nodes, OmniStack Virtual Controller and SQL cluster VMs. The results and observations of these tests are described in the Test and Results sections of the paper.

References

1. Setup for Failover Clustering and Microsoft Cluster Service
   http://pubs.vmware.com/vsphere-60/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-60-setup-mscs.pdf
2. vCenter Server 6.0 Availability Guide
   http://www.vmware.com/techpapers/2015/vcenter-server-60-availability-guide-10472.html
3. Enabling Microsoft SQL Clustering Service for VMware vCenter Server 5.5 database and later (2059560)
   https://kb.vmware.com/selfservice/search.do?cmd=displayKC&docType=kc&docTypeID=DT_KB_1_1&externalId=2059560

For more information, visit:

www.simplivity.com