DIY Hyper-V, Clustering and SQL Server

David Cobb

MCT | MCITP DBA,Dev,BI SQL 2008

daveslog.com

Who's this guy?

- David Cobb / daveslog.com / david@davidcobb.net
- Serving clients as an I.T. Consultant since '95
  - Windows Administration
  - .Net Development
  - SQL Development and Adminstration
- SQL Trainer since 2002 at Homnick Systems
- Occasional speaker
- Many hats, professional learner

Why I'm Up Here

- Learning by teaching something new
  - Hyper-V, iSCSI
- Branch out from strengths to new areas
  - You like SQL, you'll love virtualizing and clustering SQL!

Why Aren't You Up Here?

- Overcome your fears
  - Get up in front of a group and make mistakes! ☺
- Learn something new by teaching it
- Meet cool geeks

Thanks to Mobisave

- Sharing their development environment with me
- Check it out at http://mobisave.mobi/

What is Active/Active SQL Cluster and why should I care?

- High Availability, for 24/7 database servers
- Only SQL High Availability option that provides access to the whole server instance, rather than individual databases. (Denali changes this with High Availability Groups)
- Can failover node to install updates, perform maintenance on server while users access the database.
- Uses Windows Failover Clustering
  - Can also cluster File Servers, Print, etc.
  - Many cluster aware applications
- It’s just plain cool to cluster.

The Goal: SQL Active/Active Cluster Demo

Simple Step by Step Process

(Take notes, I'll wait..)

Prerequisites:
Basic knowledge of Windows administration (joining domains, adding users)
Windows Server 2008 R2 (SP1 recommended)
SQL Server 2008 R2 (SP1 recommended)

To recreate this demo you'll need a Windows Server 2008 R2 Hyper-V Environment with SSMS ran (maybe less will work)
Have an Active Directory domain and controller setup and available, with a valid login in the Domain Admins group.

You'll want to decide your Virtual Machine names, and designate several valid unused IP addresses for use in this demo. (See SQL Clustering Worksheet)

*Start in your Hyper-V Server, Hyper-V Manager***Create Virtual Networks**
You'll need two networks, an **External** one for local lan traffic (you probably already have this), and a new **Private** one for the iSCSI traffic.
Set up virtual network for iSCSI
In Hyper-V manager, click Virtual Network Manager
Click New Virtual Network
Name it iSCSI Network (See worksheet)
Type Internal Only (only VMs can access)

Create Virtual Machines
You'll need 3 Virtual Machines: SAN Host and two SQL Nodes

SAN (can be on DC or separate machine)
SQL Node 1
SQL Node 2

Create Parent Disk
I created a Virtual Machine called BASE1 with a single Dynamically Expanding VHD.
Installed Windows Server 2008 Enterprise R2 SP1 **64 bit** (32 may work) and all Windows Updates, and ran Sysprep in c:\window\system32\ (Choose OOBE, Generalize, Shutdown options)
Deleted BASE1 Virtual Machine (Virtual Hard Disk remains undeleted)

Create New Virtual Machines
For each Virtual Machine (SAN,SQL Node 1, SQL Node 2):
In Hyper-V Manager choose Create New Virtual Machine.
Choose location for the Virtual Machine files (I use V:\VMs\<VM Name>)
Grant 1024MB ram.
Choose Virtual Network (should have already created) that is configured as **External** (see above.)
Choose to create Virtual Hard Disk later and Finish.
Enter Settings for the new VM.
Select IDE0, Add a new Hard Disk, Click New.
Choose a Differencing Disk.
Enter filename under the folder for your Virtual Machine (I use V:\VMS\DaveSAN\DaveSAN.vhd)
Choose the parent disk (I use V:\VM's\BASE1.vhd) and Finish.
Repeat above steps for SQL Node 1 and SQL Node 2

Start VMs and configure networking and domain
For each Virtual Machine (SAN,SQL Node 1, SQL Node 2):
Start the Virtual Machine, allow it to configure and reboot.
Follow the defaults for the setup, but choose Skip in the activate windows screen. (Can work within 180 day trial, or activate with key later)
Set local Administrator password (P@ssw0rd is a good choice)
In Network Connections, choose properties for your Local Area Network connection..
Uncheck IPv6 (for simplicity), enter properties for IPv4, and enter IP **(Each VM need different unused IP! See worksheet under 'My Local Lan IP'**), Subnet Mask, Gateway and DNS (See your worksheet for these values)
Rename local area network connection (I use **Local Lan**)
VERIFY: In Status for this network connection you should see sent and received packets.
Join your computer to your domain, give your machine name (See worksheet), and enter valid Domain Admin login to complete the process.
Once you reboot, login as domain administrator ***from this point forward.***
Once you've logged in to the domain successfully, shutdown the VM.
In Hyper-V Manager, in the Setting for the VM , choose Add hardware, Network Adapter, click Add.
Choose the Private Network you created for iSCSI traffic (See worksheet), click OK.
Start the VM and login.
In Network Connections, choose properties for your new Local Area Network connection, not the local Lan connection.
Uncheck all by IPv4 (again for simplicity), enter properties for IPV4, and enter IP **(Each VM need different unused IP! See worksheet under 'My iSCSI Lan IP'**), Subnet Mask, **No Gateway or DNS.** Click OK
Rename local area network connection (I use **iSCSI LAN**)
Turn off Windows Firewall for all networks (for simplicity)
Repeat above steps for SQL Node 1 and SQL Node 2

VERIFY: From the SAN VM, confirm you can ping both SQL Node IPs on the iSCSI LAN network (See worksheet).

Create Snapshots (optional)
Now that networking is configured, you may wish to take a snapshot of each machine so you can return to this point as a fallback if you misconfigure iSCSI.
For each Virtual Machine (SAN,SQL Node 1, SQL Node 2):
From Hyper-V Manager, Shutdown the VM, Create Snapshot, and Start the VM

Configure iSCSI Target on SAN
Login to the SAN VM
Create a folder to hold vhd files for disks we will create. (I use C:\Disks)
Download Microsoft iSCSI Target 3.3 software from http://www.microsoft.com/downloads/en/details.aspx?FamilyID=f8e5
Run the file to extract files to any local folder (I use c:\temp)
Install the iscsitarget_public.msi file under the x64 folder, choose default options.
Create iSCSI disks
We'll create 3 disks, a 1GB disk for Quorum, and two 10GB disks for Data, one for the first SQL instance and one for the second SQL instance
(Keeping it simple. Could add additional disks for logs, tempdb, etc.)
Launch Microsoft iSCSI Software Target application
For each of 3 disks:
Right Click Devices, choose Create Virtual Disk, click Next
Enter path for the disk, I use C:\Disks\<unique>.vhd
Enter 1024 for disk size in MB
Enter LUN0 for Description
Choose Next (assign initiators later) and Finish
Repeat above steps with LUN1 and LUN2, each 10240MB (10GB)
[see image iSCSITarget - 3 LUNs.png]

Create iSCSI Target
In Microsoft iSCSI Software Target application, Rick-Click iSCSI Targets and choose Create New Target. Click Next.
Enter name and description (I use DaveSAN1, First Target), Click Next, then Click Advanced.
Click Add, choose Identifier Type as IP Address, and enter the iSCSI LAN IP addresses of SQL Node 1 (See worksheet, I use 10.1.1.11), Click Next and Finish
Repeat previous step for SQL Node 2 (I use 10.1.1.12)
Under iSCSI Targets, Right-Click your new iSCSI Target, choose Add Existing Virtual Disk to Target, and choose all disks
[see image iSCSITarget - Target with Disks.png]
Right-Click your new target, Properties, Advanced, turn off Enforce idle connection timeout
[see image iSCSITarget - Target Properties - Enforce idle connection timeout.png]

Configure iSCSI Initiators
Initiator for SQL Node 1
Login to SQL Node 1
Under Administrative Tools, Choose iSCSI Initiator, and Click Yes to start the service
In Target, enter iSCSI LAN IP of the SAN (I use 10.1.1.10) and click Quick Connect, then click Done.
Enter Disk Management (Start,Run, diskmgmt.msc) Click OK to Initialize all 3 disks.
[Note the order the disks are added is arbitrary, may not be in the order you created them. Could add disks at target and initialize one at a time at initiator to control this. Any other way?]
Locate the 1GB drive, Right-Click, Configure as a Simple Volume, Drive letter Q:, Volume Name Quorum, NTFS, Quick Format
Locate the first 10GB drive, Right-Click, Configure as a Simple Volume, Drive letter S:, Volume Name Data1, NTFS, Quick Format
Locate the 1GB drive, Right-Click, Configure as a Simple Volume, Drive letter T:, Volume Name Data2, NTFS, Quick Format
[Note the drive letters are arbitrary, and in SQL 2008 R2 unnecessary as we could use mount points. Just need to be consistent among nodes.]
[see image SQL Node 1 - Disk Management - 3 Disks.png]

Initiator for SQL Node 2
We want to verify we can connect to the disks on SQL Node 2.
Before leaving SQL Node 1, take all 3 disks offline (Right-Click greay area on left of each disk, choose Offline)
Login to SQL Node 2
Under Administrative Tools, Choose iSCSI Initiator, and Click Yes to start the service
In Target, enter iSCSI LAN IP of the SAN (I use 10.1.1.10) and click Quick Connect, then click Done.
Enter Disk Management (Start,Run, diskmgmt.msc) Click OK to Initialize all 3 disks.
[Note the disks are added with arbitrary drive letters.]
Change the drive letters, if necessary, so that the drive letters match SQL Node 1. (i.e. Quorum is Q:, Data1 is S:, and Data2 is T:)

Configure Windows Cluster
Continuing in SQL Node 2:
In Server Manager, navigate to Features, Add Feature, Failover Clustering, click Next, click Install
In Administrative Tools, click Failover Cluster Manager
Click Validate a Configuration
In the Name field, enter the Local LAN IP of SQL Node 2 (I use 10.0.1.152), click Add (it will enter the FQDN of the server)
Choose All Tests, Next, (test should pass after a few minutes with warnings but no errors), can view (and save report as MHT on desktop), finish
While you are waiting, or afterwards:
Switch over to SQL Node 1
In Server Manager, navigate to Features, Add Feature, Failover Clustering, click Next, click Install
Switch Back to SQL Node 2 and continue..

VERIFY: No errors in Cluster Validation Report
From SQL Node 2, Click Create a Cluster
In the Name field, enter the Local LAN IP of SQL Node 2: (I Use 10.0.1.152), click Add
Enter Cluster Name: (I use DaveCluster1)
Enter IP Address for your new Cluster (I use 10.0.1.160)
Next, Finish, DaveCluster1should be created
Navigate to DaveCluster1, Storage, right-click, Add a Disk. choose smaller 1GB disk
Right-click DaveCluster1, More Actions, Configure Cluster Quorum Settings, click Next, choose Node and Disk Majority, choose drive Q, click Next, Next, Finish
Right-click Storage, Add a Disk. choose 10GB Disk drive corresponding to S:, using Disk Manager to verify whether it's Disk 1, Disk 2 or Disk 3
VERIFY: S: Drive (not T:) listed under Storage, Available Storage
Right-click Nodes, Add Node, enter Local LAN IP of SQL Node 1: (I Use 10.0.1.151), click Add
Follow defaults and run all tests. (Will take longer as it tests the disk failover.)
After tests, Next and Finish, and your 2nd cluster node is installed. (Name is in lower case, not sure why or how to change.)
VERIFY: See if you can fail the resource for drive S: back and forth successfully.

Install SQL Instance1
In Active Directory Users and Computers:
Create a user for SQL Service, note the username and password. (I use DaveSQLService)
For both SQL Node 1 and SQL Node 2:
Install Framework 3.5 from Server Manager, Add Role, Application Role, and choose defaults.

From SQL Node 1:
Run setup on SQL Server 2008 R2 (Standard or Enterprise) In Windows Explorer, double-click DVD drive F: (may need to mount ISO from Hyper-V settings)
Click Installation, click New SQL Server Failover Cluster Installation
After Setup Support Files install, click Install
Click Next, leave default on Evaluation, click Next
Click accept, click Next
Setup Support Rules should pass, with 5 warnings (MSDTC and Network Binding).
Choose Database Engine, Management Tools Complete (Analysis Services is clusterable but not done for simplicity), click Next
Enter SQL Server Network Name: (I use DaveSQLCluster1)
Choose Named Instance, enter: Instance1
Click Next 4 times, accept defaults, ensure correct Disk (from Cluster, Add Disk step above) is selected.
at Cluster Network Configuration step, uncheck DHCP, enter IP Address for SQL Cluster 1 (I use 10.0.1.161)
Click Next until Server Configuration, click Same account for all services, click Browse, enter SQL Service Account (I use DaveSQLService), click Check Names, choose OK, enter password (I use P@ssw0rd)
Click Next, Click Add Current User. (Can click Data Directories tab to see the folders on the S: drive used for SQL)
(Filestream supported for Clustering, not done for simplicity)
Accepting defaults, setup should complete successfully

From SQL Node2
[Configure Clustering]
Run setup on SQL Server 2008 R2 (Standard or Enterprise) In Windows Explorer, double-click DVD drive F: (may need to mount ISO from Hyper-V settings)
Click Installation, Add Node to a SQL Server Failover Cluster
Choose defaults, enter password, complete installation
VERIFY: From Failover Cluster Manager, Under Services and Applications, Right-Click your SQL Service, More Actions, Move this Service..Move to other Node, verify success.

Install SQL Instance2
From SQL Node 2:
Run setup on SQL Server 2008 R2 (Standard or Enterprise) In Windows Explorer, double-click DVD drive F: (may need to mount ISO from Hyper-V settings)
Click Installation, click New SQL Server Failover Cluster Installation
After Setup Support Files install, click Install
Click Next, leave default on Evaluation, click Next
Click accept, click Next
Setup Support Rules should pass, with 5 warnings (MSDTC and Network Binding).
Choose Database Engine, Management Tools Complete (Analysis Services is clusterable but not done for simplicity), click Next
Enter SQL Server Network Name: (I use DaveSQLCluster2)
Choose Named Instance, enter: Instance2
Click Next 4 times, accept defaults, ensure correct Disk (from Cluster, Add Disk step above) is selected. (Should be only one unassigned)
at Cluster Network Configuration step, uncheck DHCP, enter IP Address for SQL Cluster 2 (I use 10.0.1.162)
Click Next until Server Configuration, click Same account for all services, click Browse, enter SQL Service Account (I use DaveSQLService), click Check Names, choose OK, enter password (I use P@ssw0rd)
Click Next, Click Add Current User. (Can click Data Directories tab to see the folders on the T: drive used for SQL)
(Filestream supported for Clustering, not done for simplicity)
Accepting defaults, setup should complete successfully

From SQL Node1
[Configure Clustering]
Run setup on SQL Server 2008 R2 (Standard or Enterprise) In Windows Explorer, double-click DVD drive F: (may need to mount ISO from Hyper-V settings)
Click Installation, Add Node to a SQL Server Failover Cluster
Choose defaults, enter password, complete installation
VERIFY: From Failover Cluster Manager, Under Services and Applications, Right-Click your SQL Service, More Actions, Move this Service..Move to other Node, verify success.
VERIFY: Start SQL Management Studio, View Menu, Registered Servers,, Register instances davesqlcluster1\instance1 and davesqlcluster2\instance2
RUN Query against all instances:

```
select SERVERPROPERTY('ComputerNamePhysicalNetBIOS') as NodeName
,SERVERPROPERTY('ServerName') as ServerName
,SERVERPROPERTY('MachineName') as ClusterName
,SERVERPROPERTY('IsClustered') as IsClustered
--, HOST_NAME()
, name
from sys.databases
```

DONE! (for now..)

High Level Steps

- Hyper-V Environment
- Plan. Start with a Worksheet
- Virtual Machines
  - Parent Disk
  - Differencing Disks for 3 virtual machines:
    - SAN
    - SQL Node 1
    - SQL Node 2
  - Configure each
- Other Requirements
  - Active Directory

Hyper-V Environment

- **Host Server**
  - ***Windows 2008 R2 SP1 Standard***
  - ***Hyper-V Role***
  - ***Extra 3-5GB Memory***
  - ***~50GB storage (depending on disk count, sizes)***
- **Hyper-V Virtual Networks**
  - ***Lan traffic***
    - Shared network with Host for external access
    - ex. 10.0.1.0
  - ***San traffic***
    - Private network
    - Clients use IPV4 Only
    - ex. 10.10.1.0
    - Separate networks are more work, but best practice to separate LAN and iSCSI traffic
- **Network Diagram**

Virtual Machines

- **Need Windows Server 2008 R2**
  - ***Enterprise Edition for Cluster Nodes***
  - ***Standard Edition OK for iSCSI Target***
- **Use a Parent Disk with Differencing Disks!**
  - ***Create a new VM (BASEX) of Server 2008 R2 SP1 Enterprise***
  - ***Install Windows Updates until current***
  - ***Run Sysprep***
    - c:\windows\system32\sysprep
    - Choose OOBE, Generalize, Shutdown options
  - ***Remove VM, VHD remains***
  - ***Make BASEX.vhd READ ONLY, and make a backup copy in case, as changes to parent invalidate the child disks.***
  - ***Create new VMs with no disk, add Differencing Disk based on BASEX.vhd parent***
- **DEMO New VM with Differencing Disk**

SAN - iSCSI Target

- ***Configure Networking (Local LAN, iSCSI LAN)***
- ***Install Microsoft iSCSI Target 3.3***
- ***Configure Target with Disk Devices (LUN0-LUNX)***
- ***Add Disks***
  - Quorum
  - Data disk for each SQL Node
  - Optional: Log disk for each SQL Node
  - Optional: DTC Disk
  - Example
- ***Add Target***
  - Configure access to all disks from each SQL Node
  - Turn off idle timeout
- Demo add and mount new LUN

SQL Node 1 and 2

- ***Configure Networking (Local LAN, iSCSI LAN)***
- ***Configure iSCSI Initiator***
- ***Initialize, Format and assign Volume Names, Drive Letters (Mount points supported)***
- ***Add Feature Windows Failover Clustering (Enterprise Only!)***
- ***Configure Windows Clustering***
  - ***Validate a Configuration catches issues before you cluster. DEMO***
  - ***Quorum***
  - ***Add Storage***
  - ***Add Node***
- ***Install First SQL Instance (Instance1) as SQL Failover Cluster***
  - Default Instance OK for active/passive
  - Need two instances if active/active
  - Add SQL Node to Instance1
  - TEST!
- ***Install Second SQL Instance (Instance2) as SQL Failover Cluster***
  - Add SQL Node to Instance2
  - TEST!
  - **DEMO (If time permits)**

Other Requirements

- ***Active Directory***
  - Use existing in demo, or can set up SAN virtual machine as AD controller
  - Need SQL Service User

  ALSO..

- ***Windows Server Administration Experience***
- ***SQL Server Experience***
- ***Willingness to Learn and Make Mistakes ☺***

Troubleshooting Options

- Cluster.exe for setup, scripting, diagnosis
- Cluster Configuration
  - *Evict idle node, then add again*
- Cluster Logging
  - *(as Administrator, from cmd) cluster log /g*
  - *http://blogs.msdn.com/b/clustering/archive/2008/09/24/8962934.aspx*
- SQL Installation
  - *Repair Instance*
  - *Remove Node/Add Node*
- SQL Logs
  - *If instance won't come online, check SQL Logs. If not current, it's a SAN issue, if logs are current it's a SQL issue.*

Production deployment options

- Microsoft iSCSI Target supported in production
  - *Could host on physical server for better performance*
- Other iSCSI solutions
  - *Microsoft Storage Server uses same tech as iSCSI Target*
  - *Starwind, many others*

Other High Availability Options

- Failover Clustering File Services
  http://blogs.technet.com/b/josebda/archive/2011/05/19/teched-2011-demo-install-step-by-step-hyper-v-ad-dns-iscsi-target-file-server-cluster-sql-server-over-smb2.aspx
- Denali AlwayOn Availability Groups
  http://msdn.microsoft.com/en-us/library/ff877884(v=sql.110).aspx
- Hyper-V Failover
  http://technet.microsoft.com/en-us/library/cc732181(WS.10).aspx

Q & A

- Will post slides, worksheet and STEP BY STEP instructions (larger font) to daveslog.com
- Email me david@davidcobb.net for clarification and questions.