Installing and Configuring a SQL Server 2014 Multi-Subnet Cluster on Windows Server 2012 R2

Edwin M Sarmiento

Applies to:

- SQL Server 2012
- SQL Server 2014

Copyright

Contents

Introduction 4
Assumptions 4
Network Architecture Design 5
Adding Required Windows Features 7
Discovering Target Portals 9
Connecting Targets and Configuring Multipathing 12
Initialize and Format the Disks 16
Verify the Storage Replication Process 20
Running the Failover Cluster Validation Wizard 21
Creating the Windows Server 2012 R2 Multi-Subnet Cluster 23
Tuning Cluster Heartbeat Settings 26
Install SQL Server 2014 on a Multi-Subnet Failover Cluster 27
Adding a Node on a SQL Server 2014 Multi-Subnet Cluster 39
Tuning the SQL Server 2014 Failover Clustered Instance DNS Settings 42
Testing Application Connectivity 43
About The Author 45

Introduction

A SQL Server 2014 multi-subnet failover clustered instance is a configuration where each node of the cluster is connected to a different network subnet or subnets. These network subnets can be in the same location or in a remote site commonly used for disaster recovery. This configuration provides the benefit of having both high availability and disaster recovery solution to meet business' recovery objectives for SQL Server 2014 databases.

This guide is intended for experienced Windows system administrators, IT professionals and SQL Server database administrators who would like to install and configure a 2-node Windows Server 2012 R2 Failover Cluster that will host a SQL Server 2014 multi-subnet failover clustered instance.

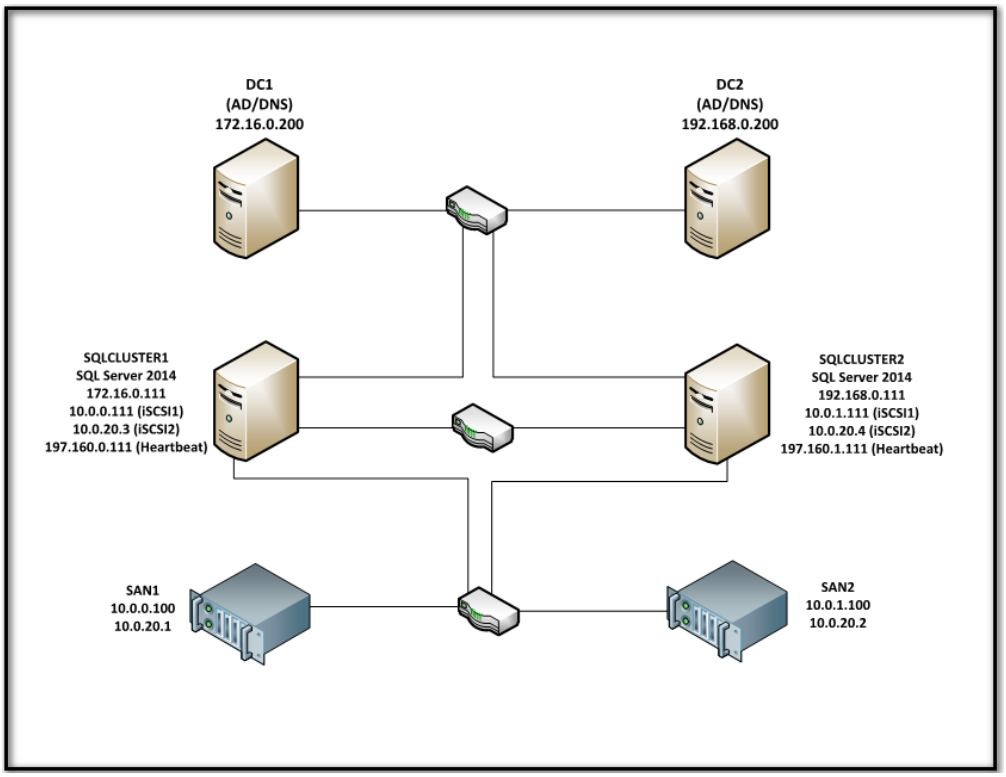
Assumptions

When using this guide, a few assumptions have been made:

- Windows Server 2012 R2 is installed on each server that will be used for the cluster and that they are joined to the same Active Directory domain.
- Configuration of the shared storage used for the cluster is outside the scope of this document. Enlist the assistance of your storage vendors and engineers to accomplish this task. For demonstration purposes, an iSCSI storage is used in this document; in particular, StarWind Virtual SAN.
- You have decided which quorum model will be used by the failover cluster. This document will use a disk witness as the quorum model.

Network Architecture Design

Proper network architecture design is key to successfully implementing a multi-subnet SQL Server 2014 failover cluster instance. Enlist the help of your network engineers to make sure that your design complies with your corporate standards and done appropriately. Below is the network diagram that will be used to implement the multi-subnet SQL Server 2014 failover clustered instance.



There are two domain controllers - **DC1** and **DC2** - in the same Active Directory domain. The domain controllers are in different network subnets, each on a dedicated Active Directory site and configured for replication. Cluster nodes **SQLCLUSTER1** and **SQLCLUSTER2** have four network adapters - one for production traffic, one for heartbeat communication and two for the iSCSI storage. Technically, there is no shared storage in a multi-subnet cluster because each node will have its own storage subsystem. However, the storage subsystem used by one node is an exact replica of the storage subsystem being used by the other nodes. In the environment described above, storage system **SAN1** is being replicated over to **SAN2** via a TCP/IP connection. A breakdown of the servers, storage systems and IP addresses is shown in the table below.

Hostname	IP Address	Purpose
DC1	172.16.0.100	Domain Controller/DNS Server
DC2	192.168.0.100	Domain Controller/DNS Server
SQLCLUSTER1	172.16.0.111	Cluster Node 1 - public traffic
	197.160.0.111	Heartbeat communication
	10.0.0.111/10.0.20.3	iSCSI communication to SAN1/SAN2
SQLCLUSTER2	192.168.0.111	Cluster Node 2 - public traffic
	197.160.1.111	Heartbeat communication
	10.0.1.111/10.0.20.4	iSCSI communication to SAN1/SAN2
SAN1	10.0.0.100/10.0.20.1	iSCSI communication
SAN2	10.0.1.100/10.0.20.2	iSCSI communication

Active Directory Domain Name: TESTDOMAIN.COM

iSCSI storage primary IP addresses: SAN1 (10.0.0.100 and 10.0.20.1) and SAN2 (10.0.1.100 and 10.0.20.2)

Cluster Nodes: SQLCLUSTER1 & SQLCLUSTER2

Cluster Disks: Q:\, F:\, G:\ & H:\

Windows Server Failover Cluster Name and IP Address: WINMULTISUBCLUS (172.16.0.112 and 192.168.0.112)

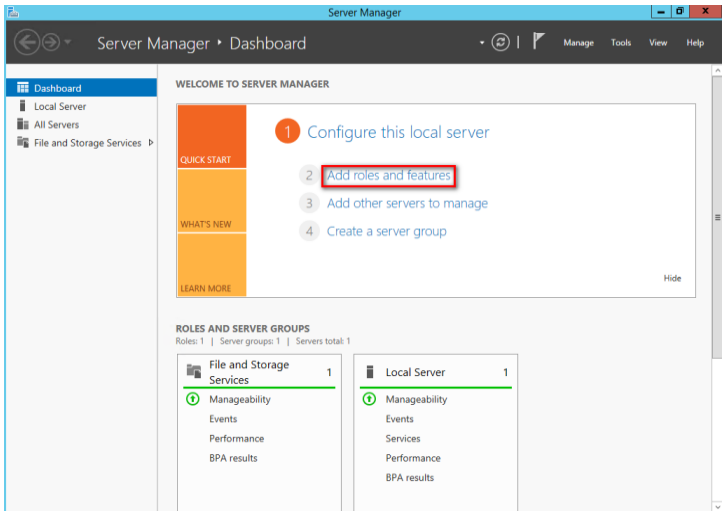
SQL Server Failover Cluster Name and IP Address: SQLCLUSTER (172.16.0.213 and 192.168.0.213)

SQL Server Service Account: TESTDOMAIN\sqlservice

Adding Required Windows Features

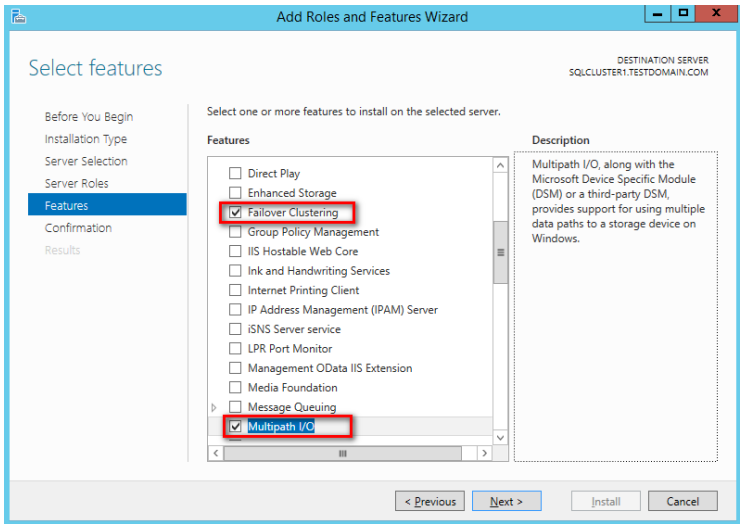
In this section, we will add the required Windows features to configure our multi-subnet failover cluster:

1. Open the **Server Manager Dashboard** and click the **Add roles and features** link. This will run the **Add Roles and Features Wizard**

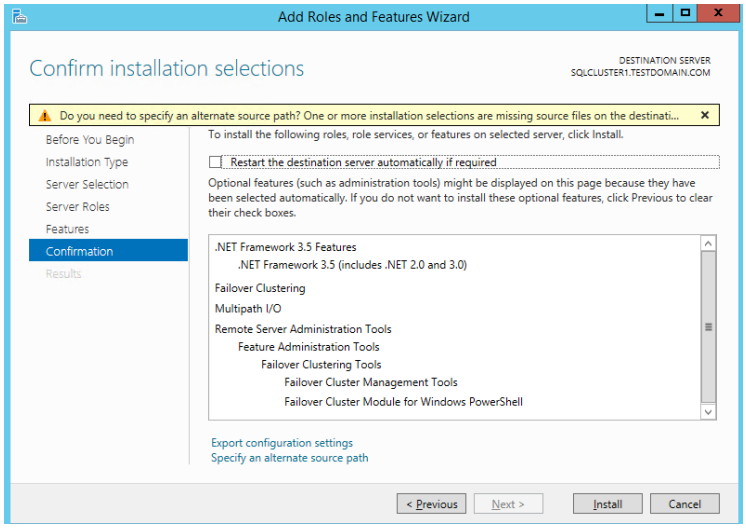


2. In the **Select Features** dialog box, select the **.NET Framework 3.5 Features** (*select only the .NET Framework 3.5 option*)**, Failover Clustering** and the **Multipath I/O** checkboxes and click **Next**.

   NOTE: The *.NET Framework 3.5* is a requirement for SQL Server 2014 and is no longer installed by the SQL Server setup process. Even if the *.NET Framework 4.5* is installed by Windows Server 2012 R2, the *.NET Framework 3.5* is not installed with it and has to be explicitly installed.



3. In the **Confirm Installation Selections** dialog box, click **Install** to confirm the selection and proceed to do the installation of the required features.

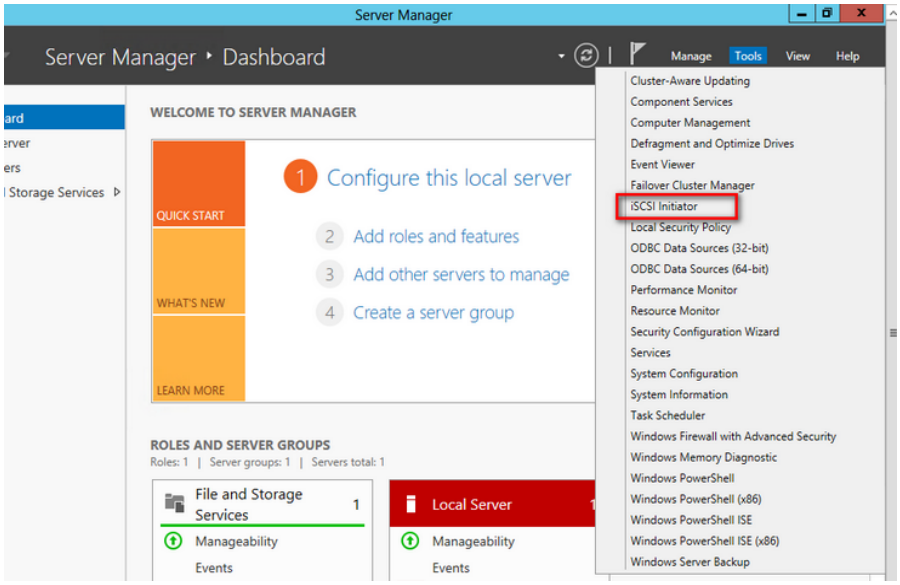


Discovering Target Portals

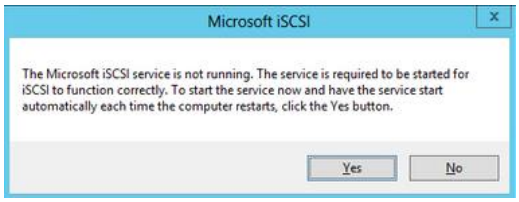
In this section, we will connect the iSCSI storage to the servers that will be added to the cluster.

NOTE: *Windows Server 2012 R2 comes with iSCSI Initiator software that enables connection of a Windows host to an external iSCSI storage array using network adapters. You can launch the tool from the* ***Server Manager*** *dashboard, under* ***Tools*** *and select* ***iSCSI Initiator****.*

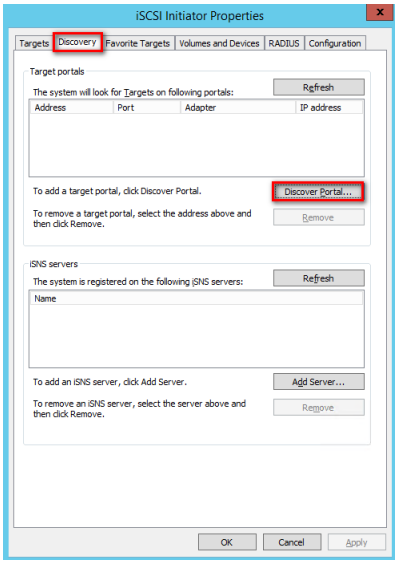
These steps have to be performed on both of the servers that will act as nodes in your failover cluster. The steps below are performed on ***SQLCLUSTER1****.*



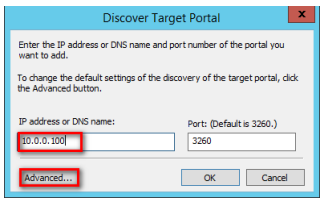
You will get a message saying that the Microsoft iSCSI service is not running. Simply click ***Yes*** *to continue.*



1. In the **iSCSI Initiator Properties** window, select the **Discovery** tab.
2. Click the **Discover Portal**… button. The **Discover Target Portal** dialog box appears.

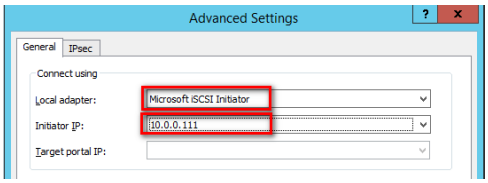


3. Type in the first IP address of the partner node you will use to connect to the highly available iSCSI devices. For this example, the IP address of **SAN1** is **10.0.0.100**



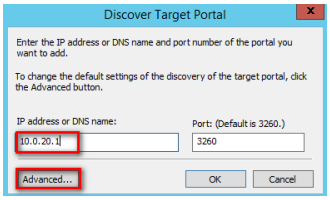
Click **Advanced**.

4. Select **Microsoft ISCSI Initiator** as your **Local adapter**. Select the **Initiator IP** in the same subnet as the IP address on the SAN server from the previous step. For this example, the first IP address of **SQLCLUSTER1** that communicates to **SAN1** is **10.0.0.111**.



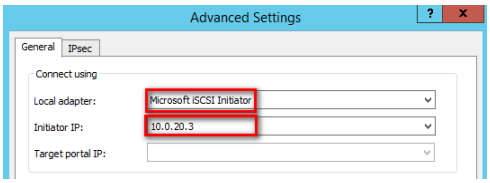
Click **OK**. Then click **OK** again to close the **Discover Target Portal** dialog box.

5. Click the **Discover Portal** button once again. The **Discover Target Portal** dialog appears.
6. Type in the second IP address of the partner node you will use to connect to the HA iSCSI devices. For this example, the IP address of **SAN1** is **10.0.20.1.**



Click **Advanced**.

7. Select **Microsoft ISCSI Initiator** as your **Local adapter**. Select the **Initiator IP** in the same subnet as the IP address on the SAN server from the previous step. For this example, the second IP address of **SQLCLUSTER1** that communicates to **SAN1** is **10.0.20.3**.

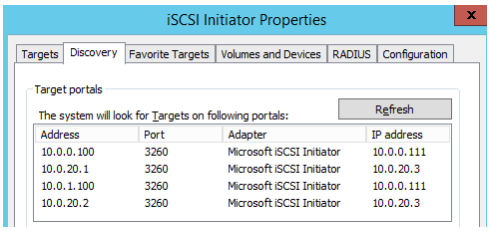


Click **OK**. Then click **OK** again to close the **Discover Target Portal** dialog box.

8. Repeat the same steps (*steps #1 to #7*) to add **SAN2** to the list of discovered targets. Note the following:

   - **10.0.1.100** and **10.0.20.2** (first and second IP addresses of **SAN2**, respectively)
   - **10.0.0.111** (first IP address of **SQLCLUSTER1** that communicates to the first IP address **SAN2**)
   - **10.0.20.3** (second IP address of **SQLCLUSTER1** that communicates to the second IP address **SAN2**)

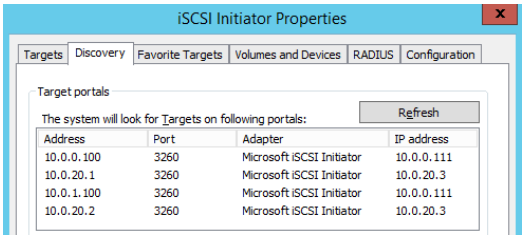
   SQLCLUSTER1 should be connected on both **SAN1** and **SAN2** via the following target portals.



9. Repeat the same steps (*steps #1 to #8*) for the second node **SQLCLUSTER2** until all the target portals have been added. Note the following:

   - **10.0.0.100** and **10.0.20.1** (first and second IP addresses of **SAN1**, respectively)
   - **10.0.1.111** (first IP address of **SQLCLUSTER2** that communicates to **SAN1**)
   - **10.0.20.4** (second IP address of **SQLCLUSTER2** that communicates to **SAN1**)
   - **10.0.1.100** and **10.0.20.2** (first and second IP addresses of **SAN2**, respectively)
   - **10.0.1.111** (first IP address of **SQLCLUSTER2** that communicates to **SAN2**)
   - **10.0.20.4** (second IP address of **SQLCLUSTER2** that communicates to **SAN2**)

   SQLCLUSTER2 should be connected on both **SAN1** and **SAN2** via the following target portals.

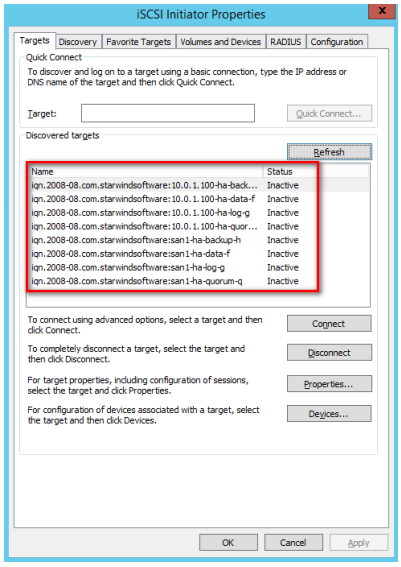


Connecting Targets and Configuring Multipathing

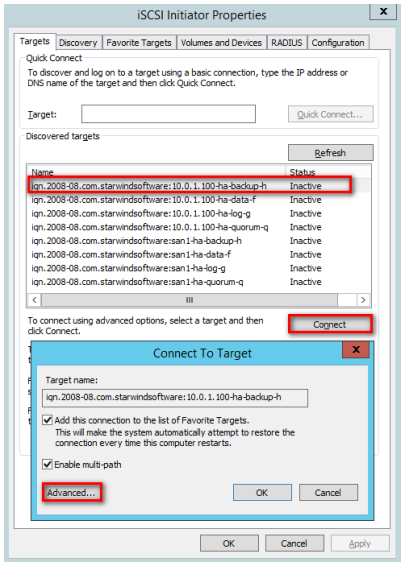
In this section, we will connect the servers to the iSCSI targets and configure multipathing:

NOTE: *These steps have to be performed on both of the servers that will act as nodes in your failover cluster. The steps below are performed on* ***SQLCLUSTER1****.*

1. In the **iSCSI Initiator Properties** window, select the **Targets** tab. The iSCSI targets configured should be listed in the **Discovered Targets** section.



2. Select the first target in the list and click **Connect**.
3. Enable both checkboxes. Click **Advanced...**



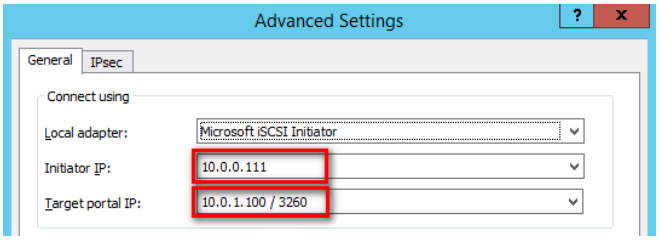
4. Select **Microsoft iSCSI Initiator** in the **Local adapter** drop down list.

In the **Initiator IP** drop down list, select the IP address of the server that connects to the corresponding initiator.

In the **Target portal IP** drop down list, select the IP address of the iSCSI Target where the Initiator IP address is mapped to.

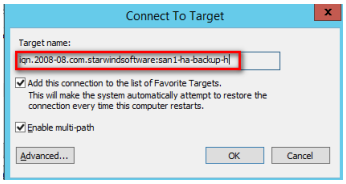
> ***NOTE:*** *The selection for* ***Initiator IP*** *and* ***Target portal IP*** *addresses depend on the iSCSI target selected in Step #2. In this example, the target*
>
> ***iqn.2008-08.com.starwindsoftware:10.0.1.100-ha-backup-h***
>
> *was selected. This corresponds to the iSCSI Qualified Name (IQN) of* ***SAN2****. The* ***Initiator IP*** *address for* ***SQLCLUSTER1*** *(****10.0.0.111****) is used to connect to* ***SAN2****.*

Click **OK**.



5. Select the partner target from the other iSCSI target node and click **Connect**. For the iSCSI target selected in Step #2, the partner target is

   iqn.2008-08.com.starwindsoftware:san1-ha-backup-h



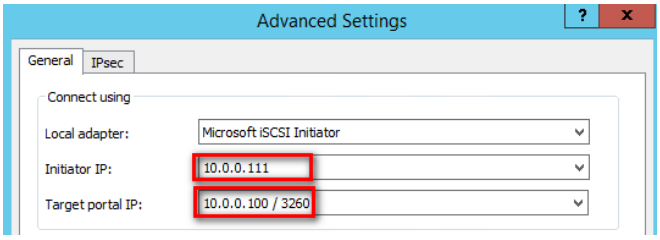
6. Enable both checkboxes. Click **Advanced...**
7. Select **Microsoft iSCSI Initiator** in the **Local adapter** drop down list.

In the **Initiator IP** drop down list, select the IP address of the server that connects to the corresponding initiator.

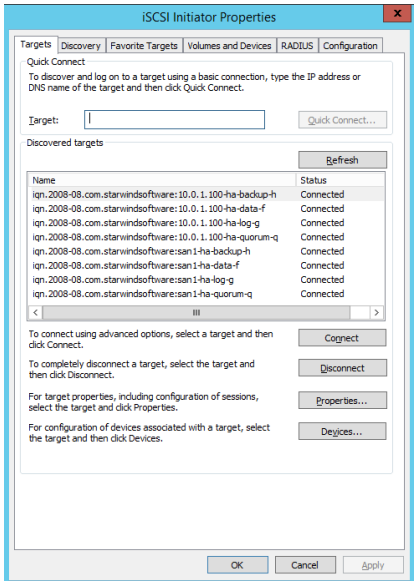
In the **Target portal IP** drop down list, select the IP address of the iSCSI Target where the Initiator IP address is mapped to.

> ***NOTE:*** *The selection for* ***Initiator IP*** *and* ***Target portal IP*** *addresses depend on the iSCSI target selected in Step #5. In this example, the target*
>
> ***iqn.2008-08.com.starwindsoftware:san1-ha-backup-h***
>
> *was selected. This corresponds to the iSCSI Qualified Name (IQN) of* ***SAN1****. The* ***Initiator IP*** *address for* ***SQLCLUSTER1*** *(****10.0.0.111****) is used to connect to* ***SAN1****.*

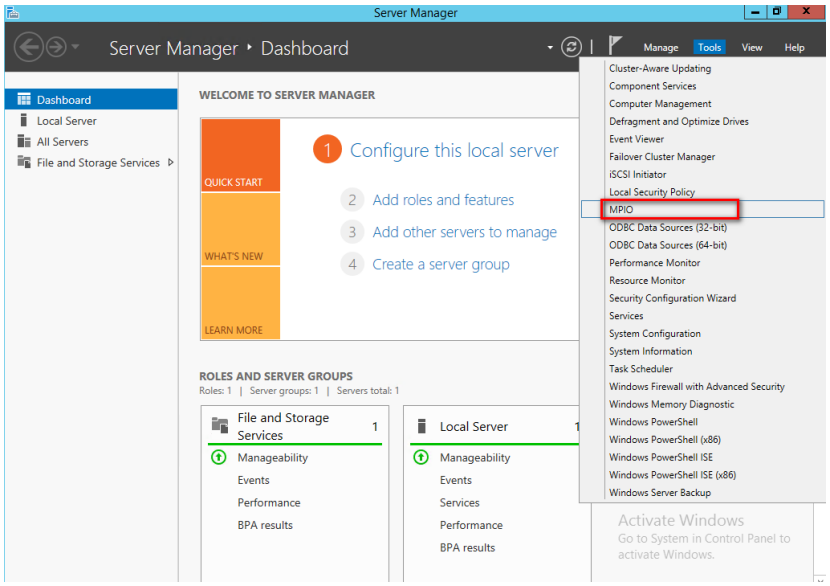
Click **OK**.



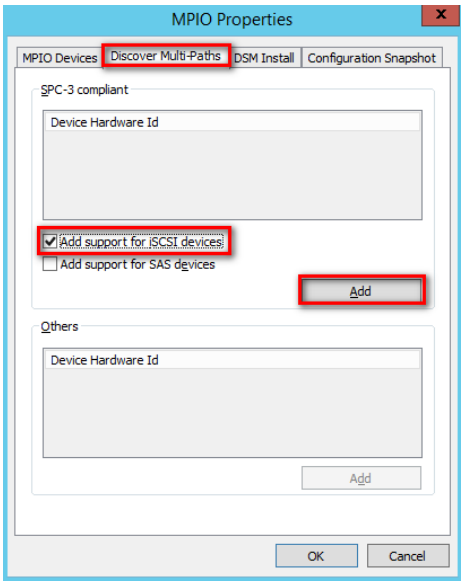
8. Repeat the ***Steps #1*** to ***#7*** with the Initiator and Target portal IPs of the remaining iSCSI targets together with their corresponding partner targets. The server should now be connected to all provisioned highly available, fault tolerant iSCSI targets. The result should look similar to the one below.



9. Repeat the ***Steps #1*** to ***#8*** on **SQLCLUSTER2**.
10. Once all targets are connected, launch the MPIO manager from the **Server Manager** dashboard, under **Tools** and select **MPIO**



11. In the **MPIO Properties** dialog box, select the **Discover Multi-Paths** tab and enable the **Add support for iSCSI devices** checkbox.
12. Click the **Add** button and click **OK**.



Reboot the server to apply the changes. Repeat ***Step #10*** to ***#12*** on **SQLCLUSTER2**.

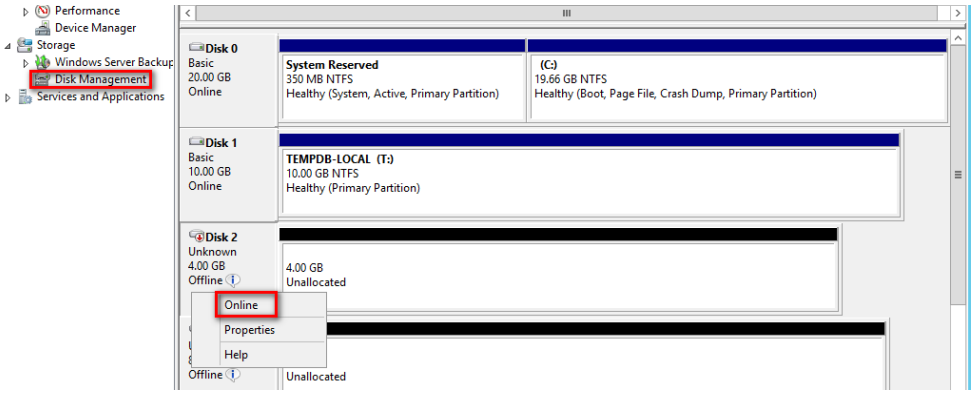
Initialize and Format the Disks

In this section, we will initialize and format the iSCSI disks. You can launch the tool from the **Server Manager** dashboard, under **Tools** and select **Computer Management.**

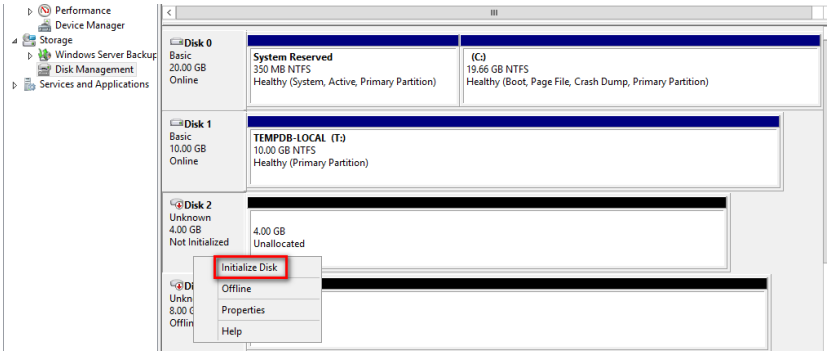
NOTE: *Going thru the disk initialization process is a great way to validate whether or not the storage replication process works as per vendor specification. Disk configuration changes made on one of the cluster nodes should be replicated over to the other nodes within the cluster.*

These steps have to be performed on both of the servers that will act as nodes in your failover cluster. The steps below are performed on ***SQLCLUSTER1****.*

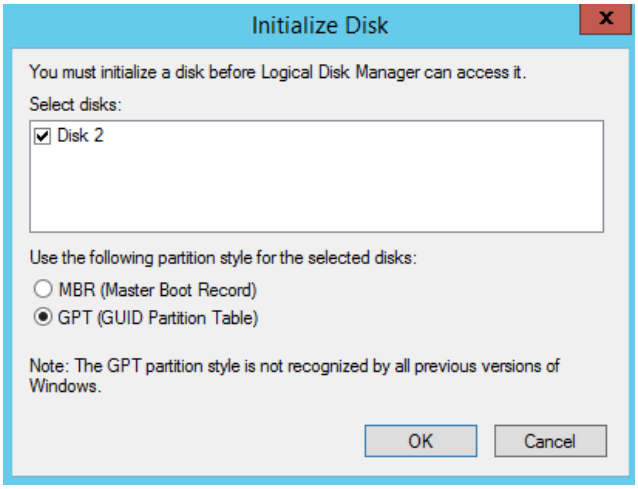
1. Expand **Storage** and select **Disk Management**.
2. Right-click any of the disks that you want to configure and select **Online**. Once the disk is brought online, it is now marked as **Not Initialized**.



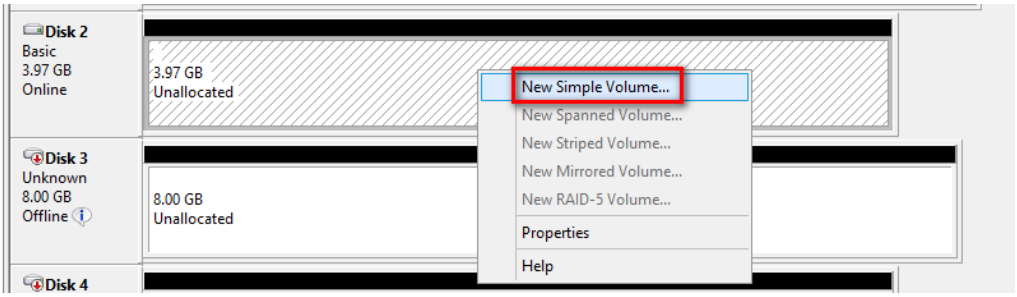
3. To initialize, right-click on the disk and select **Initialize Disk**. The Initialize Disk dialog box will appear.



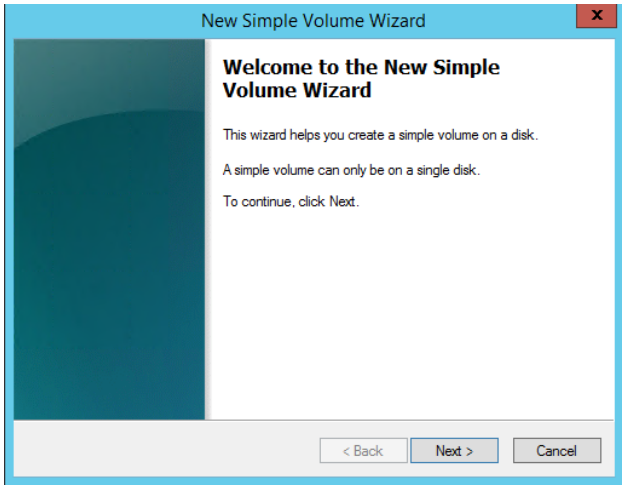
4. In the **Initialize Disk** dialog box, make sure that the correct disk is selected for initialization and then choose whether to initialize the disk using the MBR or GPT partition styles. For this configuration, we will use a **GPT** partition style. Click **OK**.



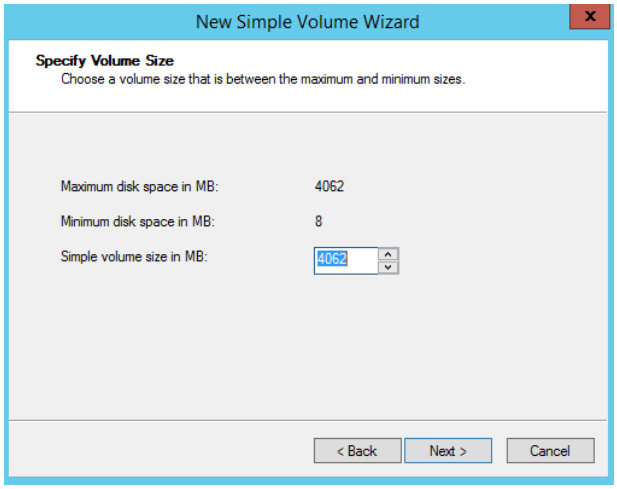
6. To create a disk partition, right-click on the unallocated space and select **New Simple Volume**.



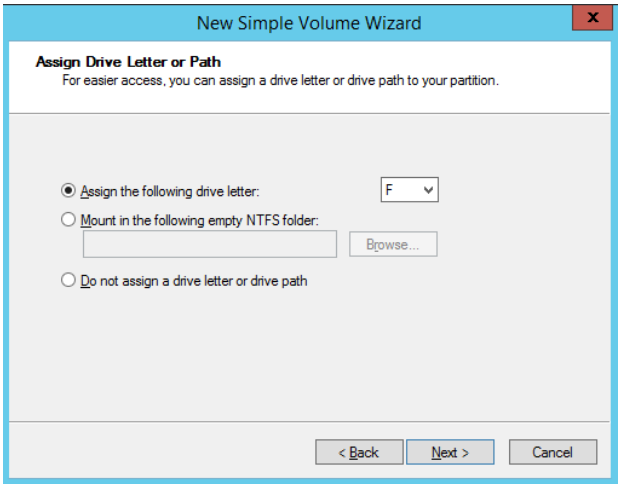
7. In the **Welcome to the New Simple Volume Wizard** dialog box, click **Next**.



8. In the **Specify Volume Size** dialog box, enter the volume size and click **Next**.

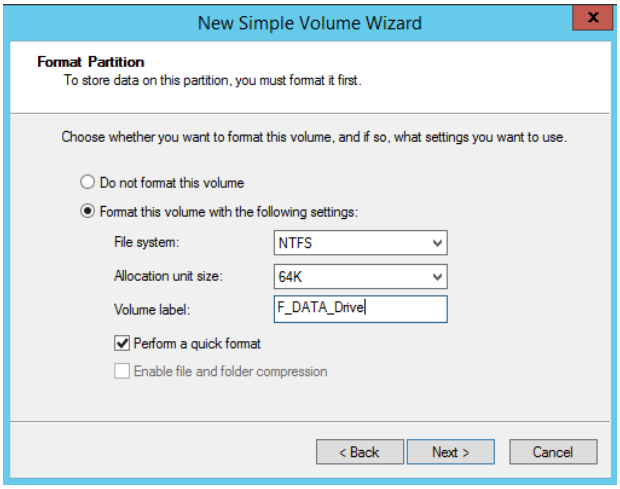


9. In the **Assign Drive Letter or Path** dialog box, specify the drive letter you would like to use and click **Next**.

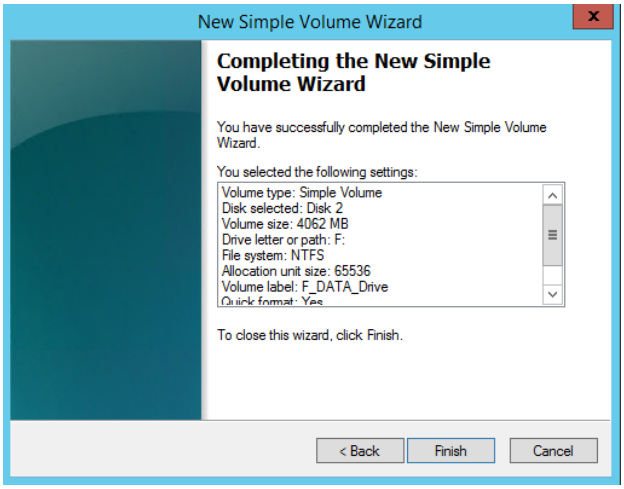


10. In the **Format Partition** dialog box,
    - Make sure that the file system selected is **NTFS**.
    - To follow Microsoft best practices on allocation unit size, select **64K**.
    - In the **Volume label:** text box, enter the appropriate name. For this example,**F_DATA_Drive** is used. Take note of this volume label because this will be used to verify the configuration on the other cluster node.

      Click **Next**



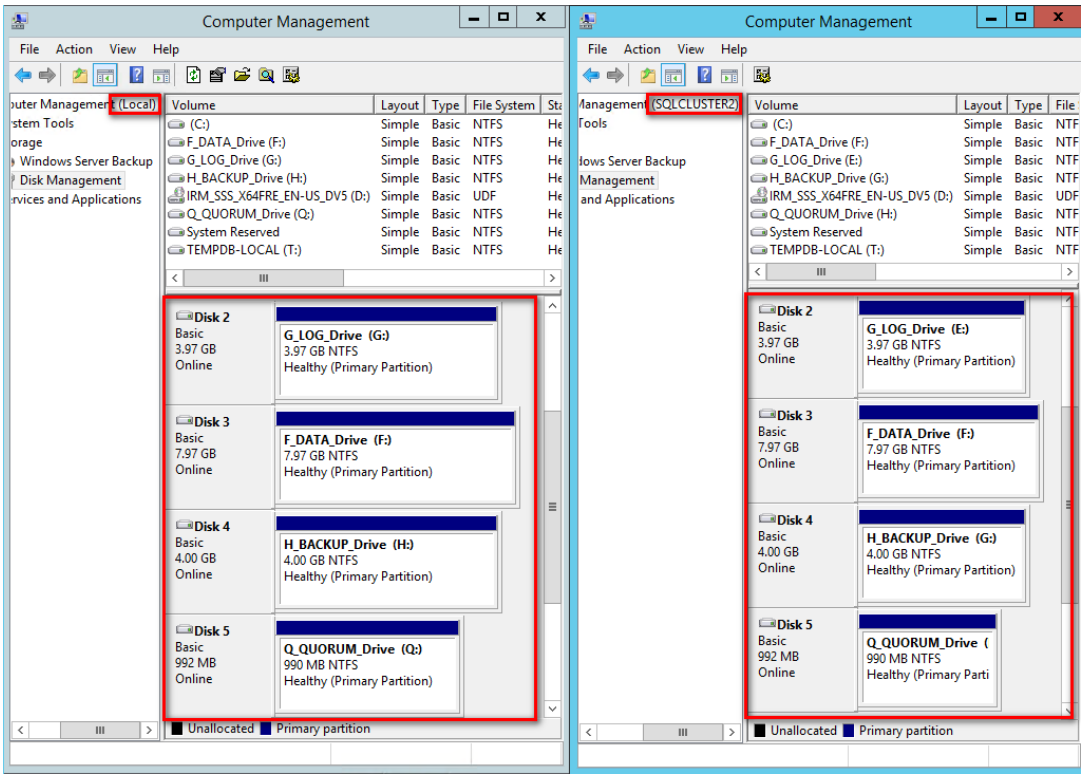
11. In the **Completing the New Simple Volume Wizard** dialog box, review the settings you have made and click **Finish**.



12. Repeat ***Steps #3*** to ***#11*** on all of the iSCSI disks that you want to configure as part of your cluster.
13. Repeat ***Step #2*** on **SQLCLUSTER2**. No need to initialize the iSCSI disks.

Verify the Storage Replication Process

In this section, we will verify the storage replication process. In order to verify this process, simply bring all of the disks on the other cluster nodes online, as per ***Step #2*** in the previous section. If the storage replication works, the volume names will be propagated on all of the cluster nodes. In this example, the clustered disks have been named **Q_QUORUM_Drive**, **F_DATA_Drive**, **G_LOG_Drive** and **H_BACKUP_Drive** on **SQLCLUSTER1**. After bringing the disks online on **SQLCLUSTER2**, the same volume properties will appear. The drive letters will not be the same because Windows will assign them from the available drive letters on the server. The drive letters will be removed since they will be defined from within the Windows Server Failover Cluster. A screenshot of the **Disk Management** console for both **SQLCLUSTER1** and **SQLCLUSTER2** is shown below.



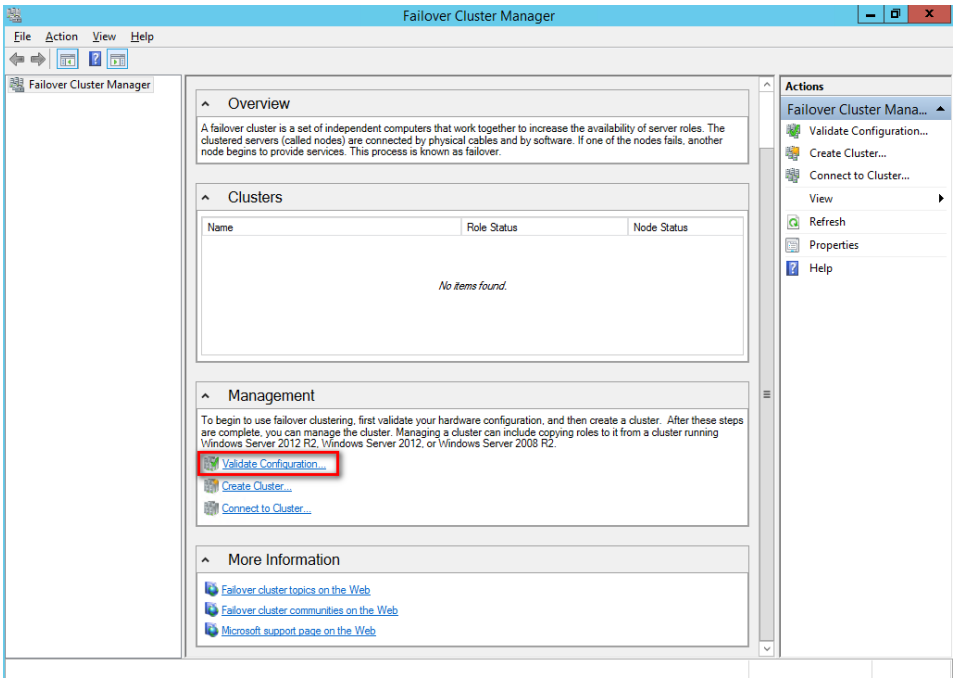
This is just a simple way to verify if the storage replication works as expected. Make sure that this verification step has been done and that all potential issues have been addressed prior to moving to the next step.

Running the Failover Cluster Validation Wizard

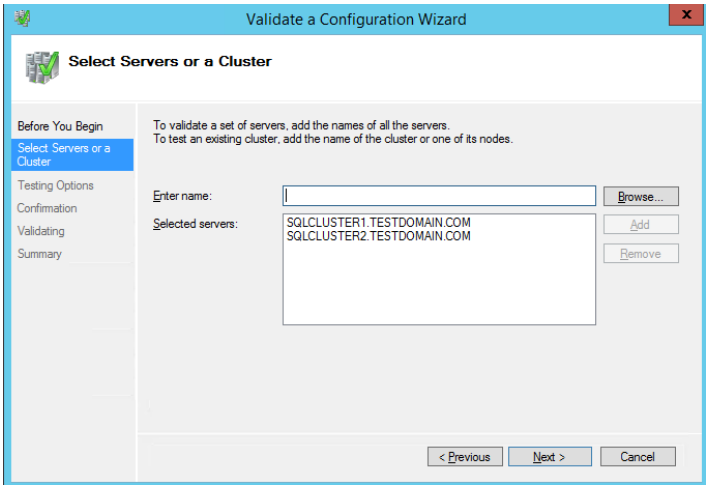
In this section we will run the Failover Cluster Validation Wizard from the Failover Cluster Management console. You can launch the tool from the **Server Manager** dashboard, under **Tools** and select **Failover Cluster Manager**.

NOTE: *These steps can be performed on any of the servers that will act as nodes in your failover cluster. The steps below are performed on* ***SQLCLUSTER1****.*

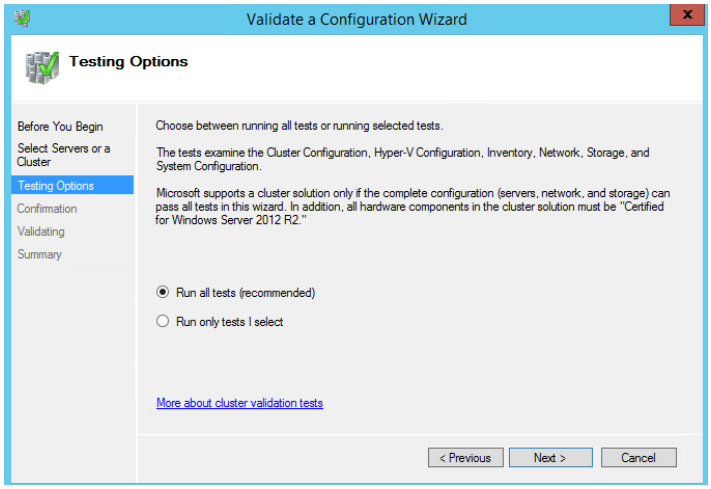
1. In the **Failover Cluster Management** console, under the **Management** section, click the **Validate Configuration** link. This will run the **Validate a Configuration Wizard**.



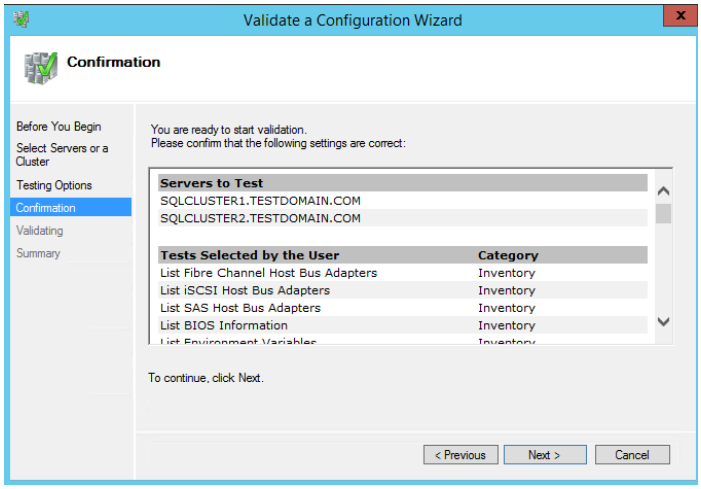
2. In the **Select Servers or a Cluster** dialog box, enter the hostnames of the nodes that you want to add as members of your cluster. Click **Next**.



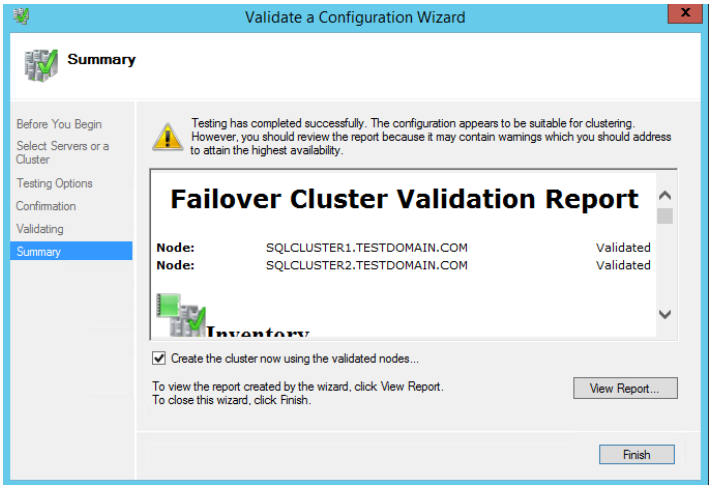
3. In the **Testing Options** dialog box, click **Next** to run all the necessary tests to validate whether or not the nodes are OK for clustering.



4. In the **Confirmation** dialog box, click **Next**. This will run all the necessary validation tests.



5. In the **Summary** dialog box, verify that all the report returns successful. Click **Finish** to create the Windows Server Failover Cluster.



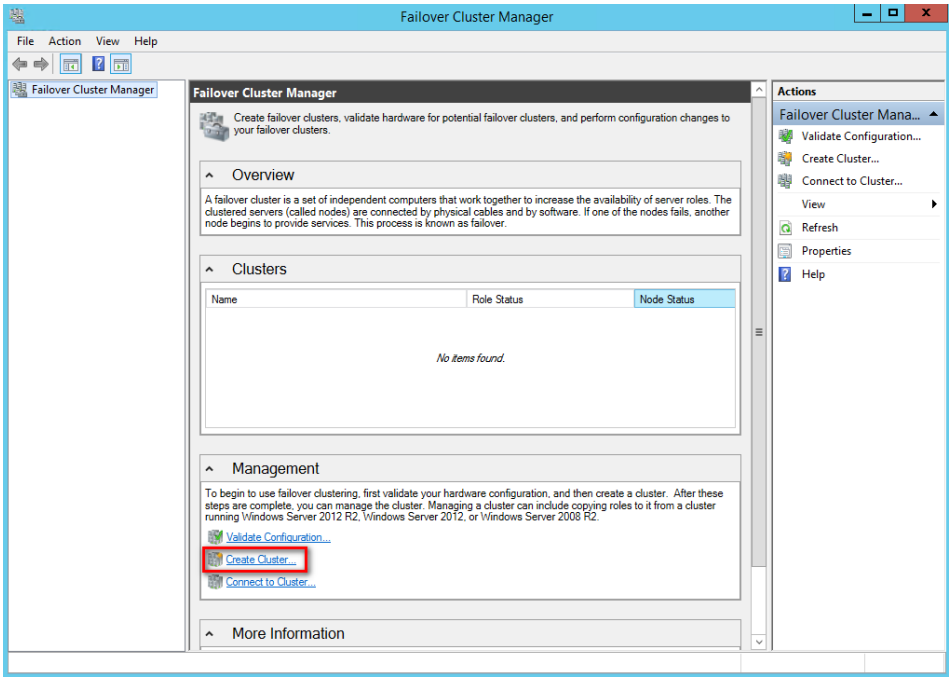
NOTE: *The* ***Cluster Validation Wizard*** *may report* ***Warning*** *messages pertaining to the network. This is because the iSCSI network is on a dedicated network segment that is not accessible from the public network traffic. You can ignore these warnings. In general, resolve all errors prior to proceeding with the next steps.*

Creating the Windows Server 2012 R2 Multi-Subnet Cluster

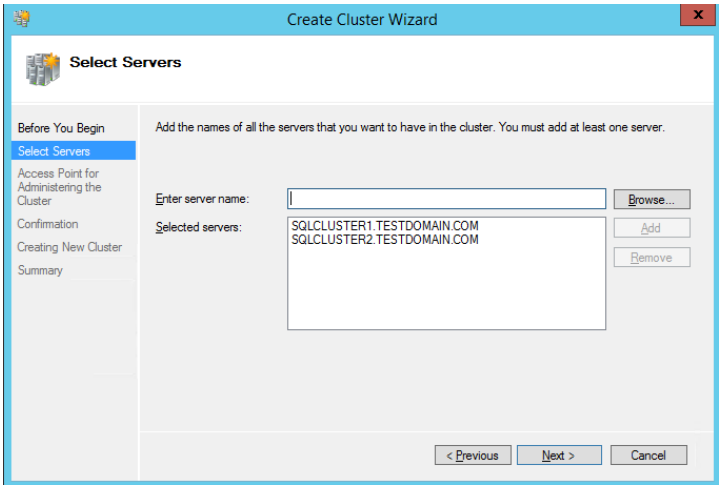
In this section we will create a Windows Server 2012 R2 Multi-Subnet Failover Cluster from the **Failover Cluster Management** console. You can launch the tool from the **Server Manager** dashboard, under Tools and select **Failover Cluster Manager**. Alternatively, the Create Cluster Wizard will automatically run after the Failover Cluster Validation Wizard runs the first time.

NOTE: *These steps can be performed on any of the servers that will act as nodes in your failover cluster. The steps below are performed on* ***SQLCLUSTER1****.*

1. Under the **Management** section, click the **Create a Cluster** link. This will run the **Create Cluster Wizard**.



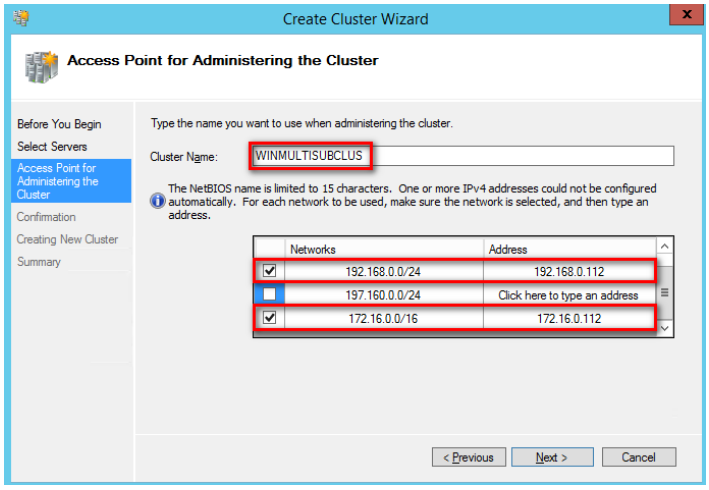
2. In the **Select Servers** dialog box, enter the hostnames of the nodes that you want to add as members of your cluster. Click **Next**.



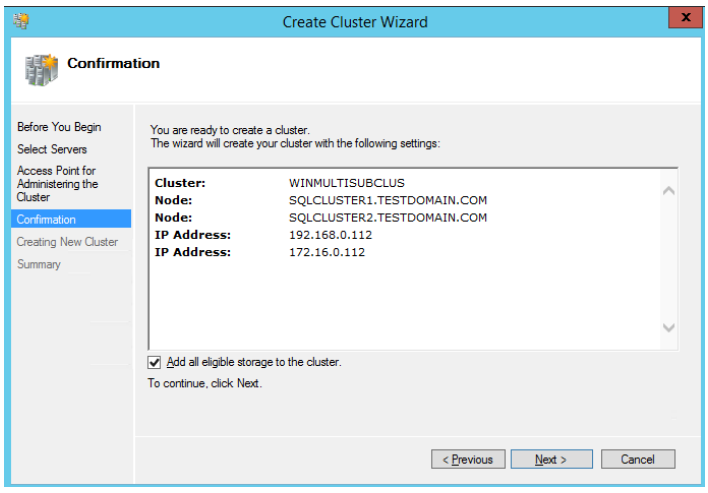
3. In the **Access Point for Administering the Cluster** dialog box, enter the Windows Server Failover Cluster virtual hostname and IP addresses that you will use to administer the cluster. Notice that you now have multiple sections for the virtual IP address - one for each subnet. Only assign virtual IP addresses for the production network.

Virtual Server Name	Networks	IP Address
WINMULTISUBCLUS	172.16.0.0/24	172.16.0.112
	192.168.0.0/24	192.168.0.112

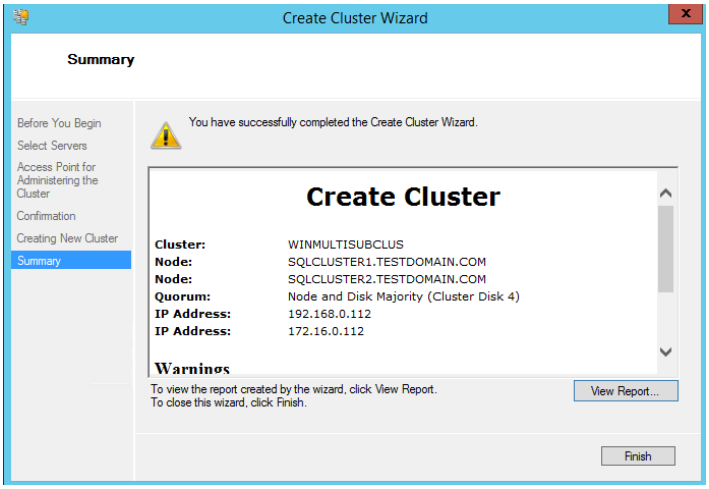
Click **Next**.



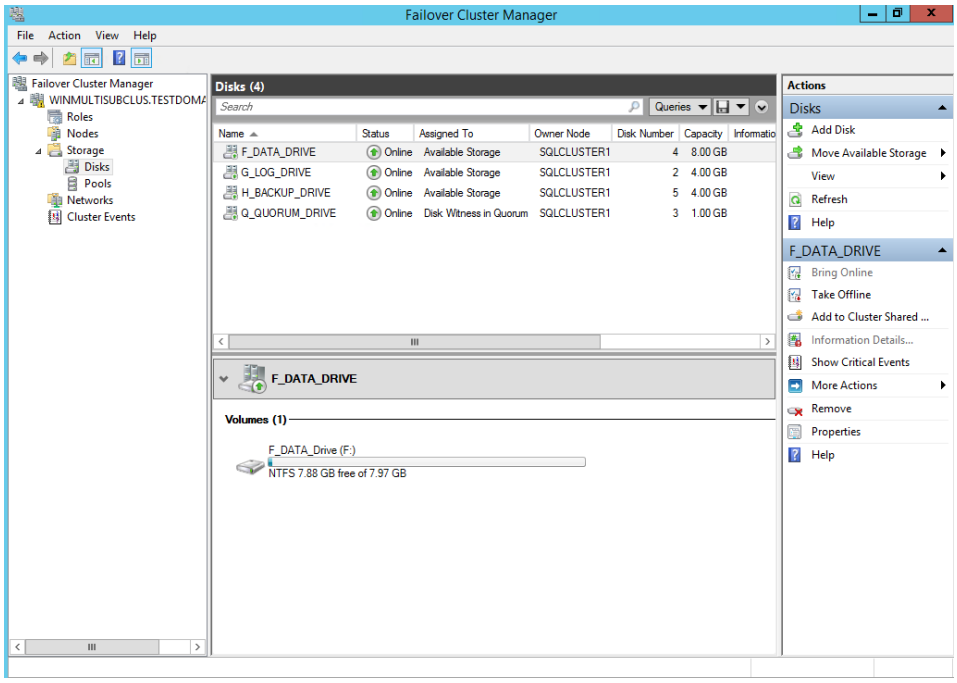
4. In the **Confirmation** dialog box, click **Next**. This will configure Failover Clustering on both nodes of the cluster, add the configured cluster storage, add Active Directory and DNS entries for the cluster virtual server name.



5. In the **Summary** dialog box, verify that the report returns successful results.



NOTE: *You may need to configure the cluster storage depending on how the local storage is configured on the server. In this example, the* ***Create Cluster Wizard*** *reported a warning because two disks are not configured as clustered storage. Each server is configured with one extra local storage that will be specifically used for the* ***tempdb*** *database. Be sure to reconfigure the cluster storage to reflect the configuration you want for your cluster. Also, name the cluster storage properly for proper identification during SQL Server 2014 failover clustered instance installation.*



Tuning Cluster Heartbeat Settings

In this section, we will tune the cluster heartbeat settings for multi-subnet clusters. We will use **Windows PowerShell** to perform the following tasks.

NOTE: *The communication between cluster nodes, more commonly known as the "**heartbeat**", needs to be properly configured for the cluster to work efficiently. Inefficient communication between cluster nodes may trigger a false failover, thus, it is necessary to properly tune the heartbeat settings.*

There are two major settings that affect heartbeat. First, the frequency at which the nodes send signals to the other nodes in the cluster (subnet delays) and, second, the number of heartbeats that a node can miss before the cluster initiates a failover (subnet threshold). Rarely do we make modifications to these settings in a single-subnet cluster because the default delay and threshold values are tolerable enough for the cluster to handle without initiating a false failover. However, in a multi-subnet cluster, when the cluster nodes are too far away from each other, the communication may take longer and could possibly miss heartbeats. The table below outlines the default values for cluster subnet delays and thresholds.

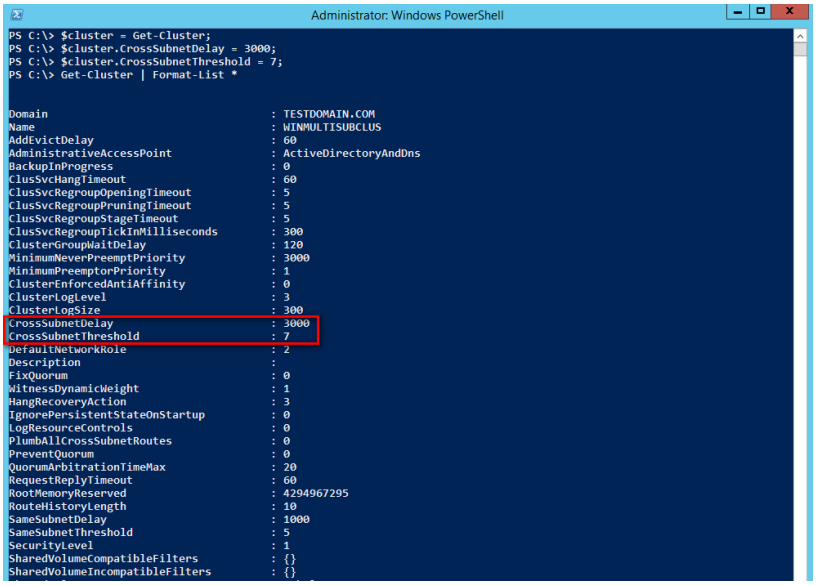
Heartbeat Parameter	Default value
SameSubnetDelay	1000 (in milliseconds)
SameSubnetThreshold	5 heartbeats
CrossSubnetDelay	1000 (in milliseconds)
CrossSubnetThreshold	5 heartbeats

*We need to increase the values for the **CrossSubnetDelay** and **CrossSubnetThreshold** parameters of the Windows Server Failover Cluster.*

*These steps can be performed on either of the nodes in your failover cluster. The steps below are performed on **SQLCLUSTER1**.*

1. Open the **Windows PowerShell** console in *Administrator mode*
2. Type the following command. This will change the cross subnet delay value to **3** seconds and the cross subnet threshold value of **7**.

```
PS C:\> $cluster = Get-Cluster;
PS C:\> $cluster.CrossSubnetDelay = 3000;
PS C:\> $cluster.CrossSubnetThreshold = 7;
```

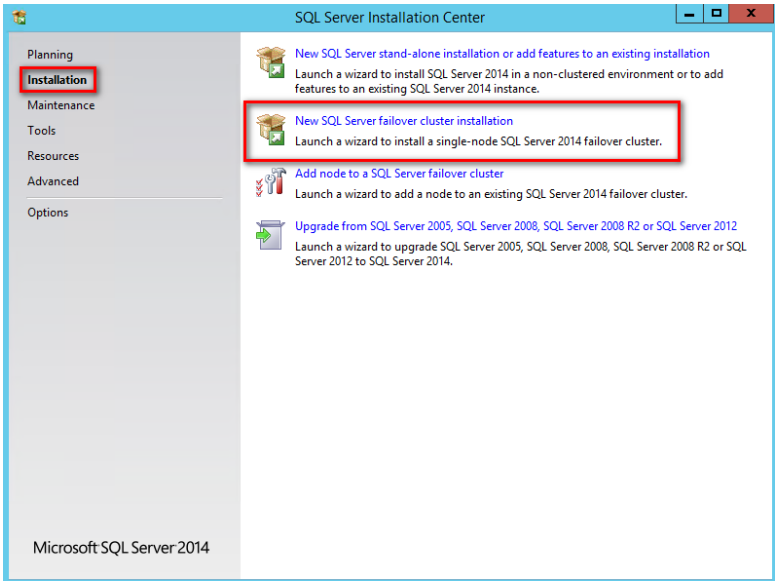


This now changes the behavior of the cluster heartbeat to be more tolerable across multiple subnets.

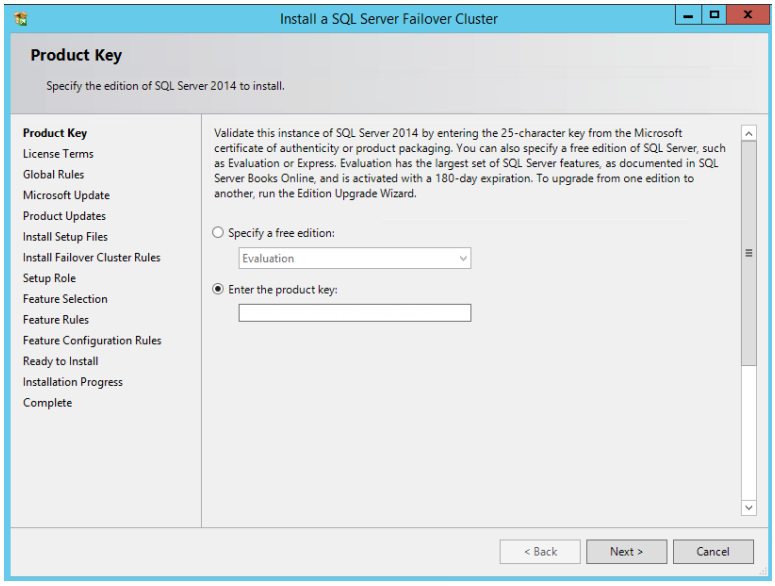
Install SQL Server 2014 on a Multi-Subnet Failover Cluster

In this section, we will install SQL Server 2014 failover clustered default instance on a multi-subnet Windows Server Failover Cluster. We will run the installation process on the first node of our cluster, **SQLCLUSTER1**.

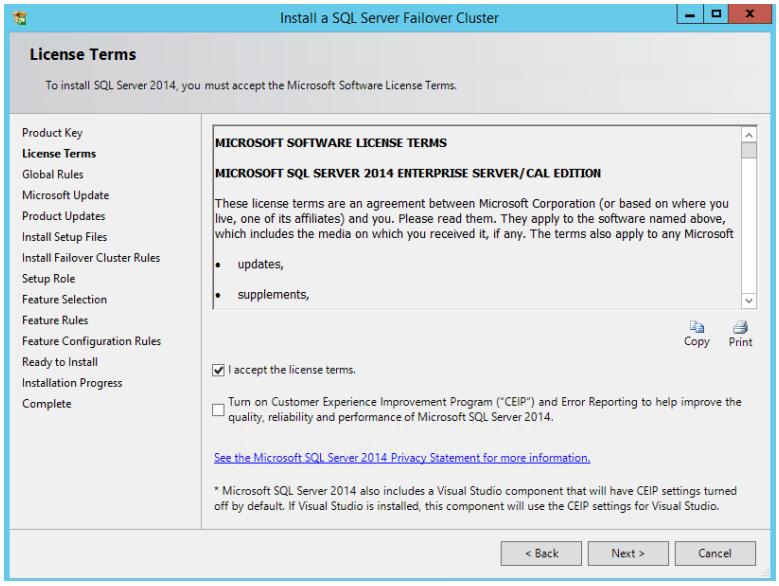
1. Run **setup.exe** from the SQL Server 2014 installation media to launch **SQL Server Installation Center**. Click on the **Installation** link on the left-hand side
2. Click the **New SQL Server failover cluster installation** link. This will run the SQL Server 2014 Setup wizard



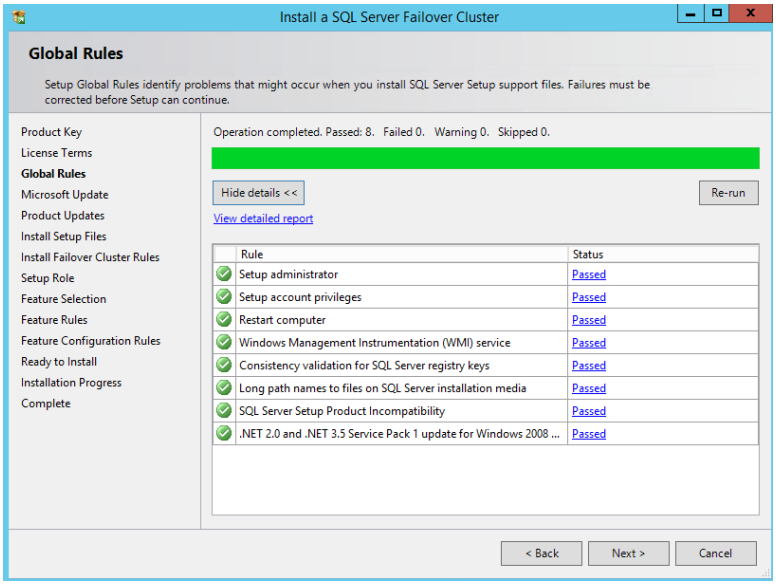
3. In the **Product Key** dialog box, enter the product key that came with your installation media and click **Next**.



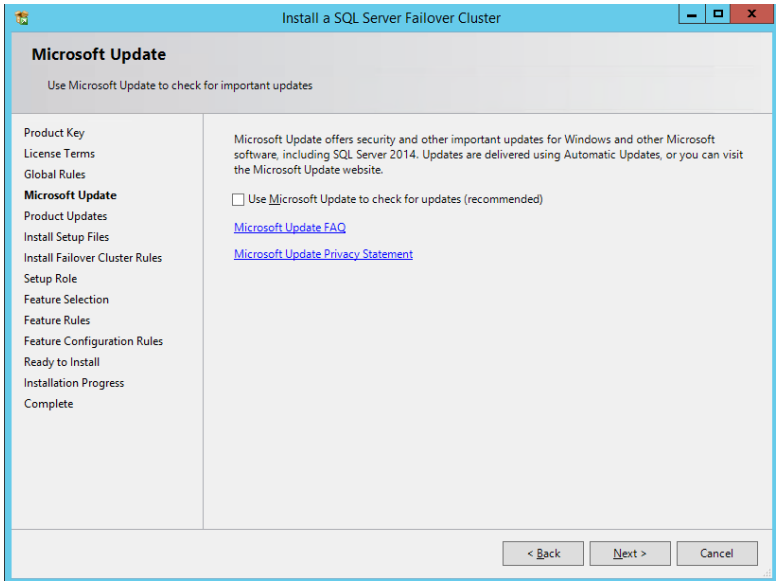
4. In the **License Terms** dialog box, click the **I accept the license terms** check box and click **Next**.



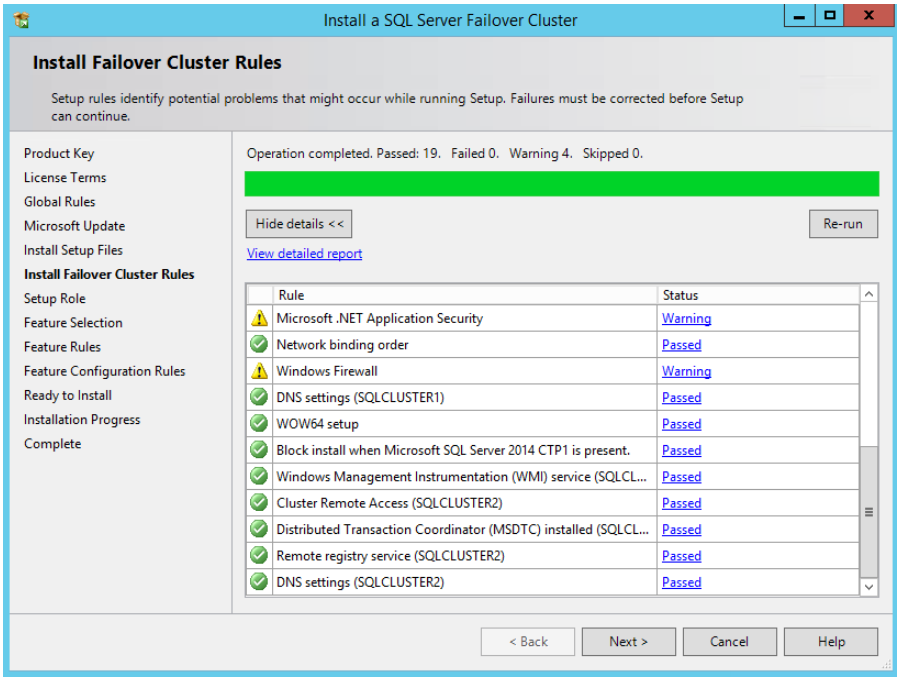
5. In the **Global Rules** dialog box, validate that the checks return successful results and click **Next**.



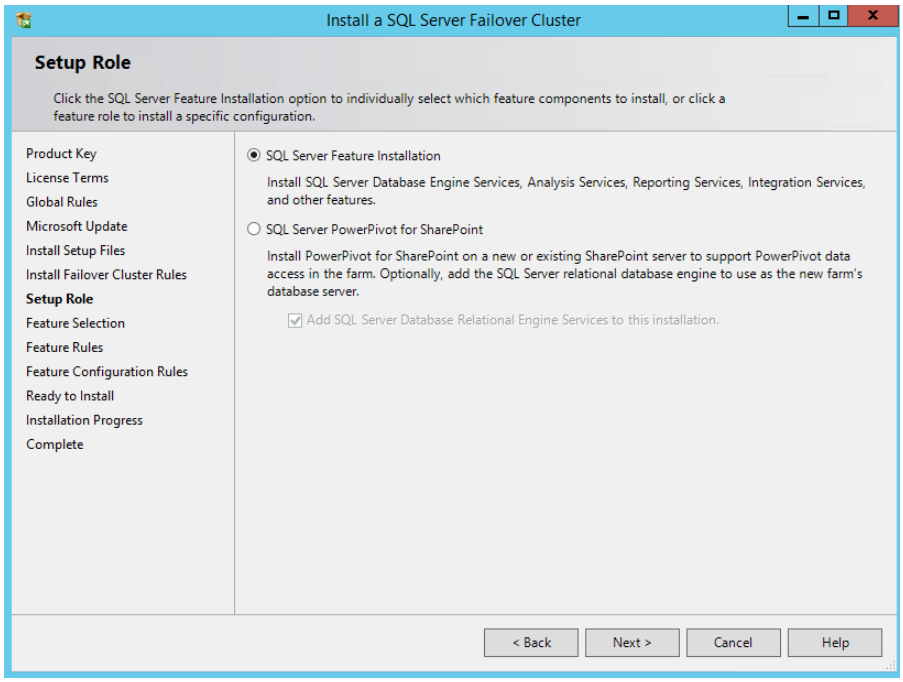
6. In the **Microsoft Update** dialog box, click **Next**.



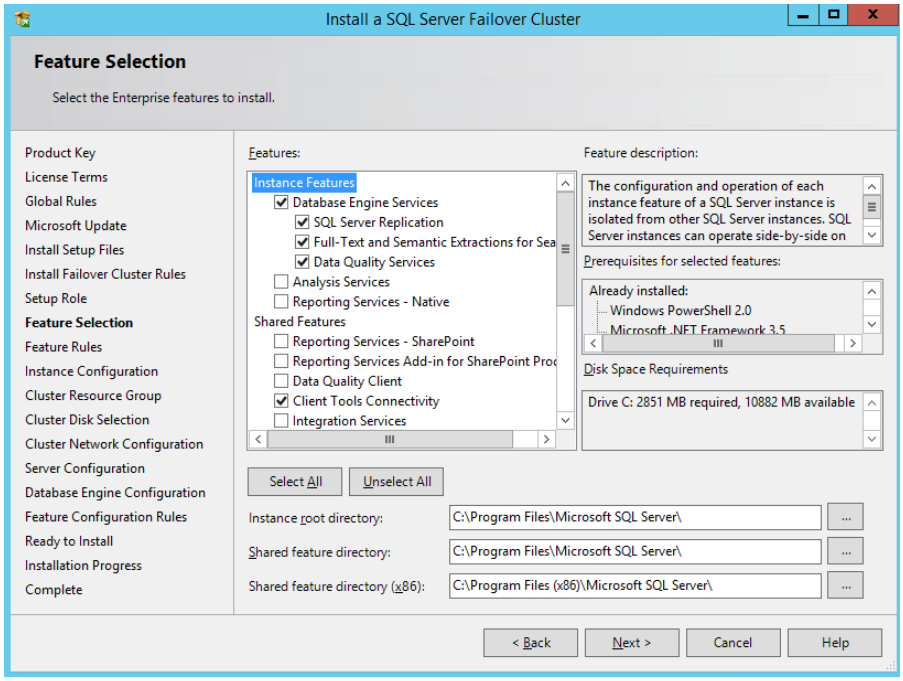
7. In the **Install Failover Cluster Rules** dialog box, validate that the checks return successful results. If the checks returned a few warnings, make sure you fix them before proceeding with the installation. Click **Next**.



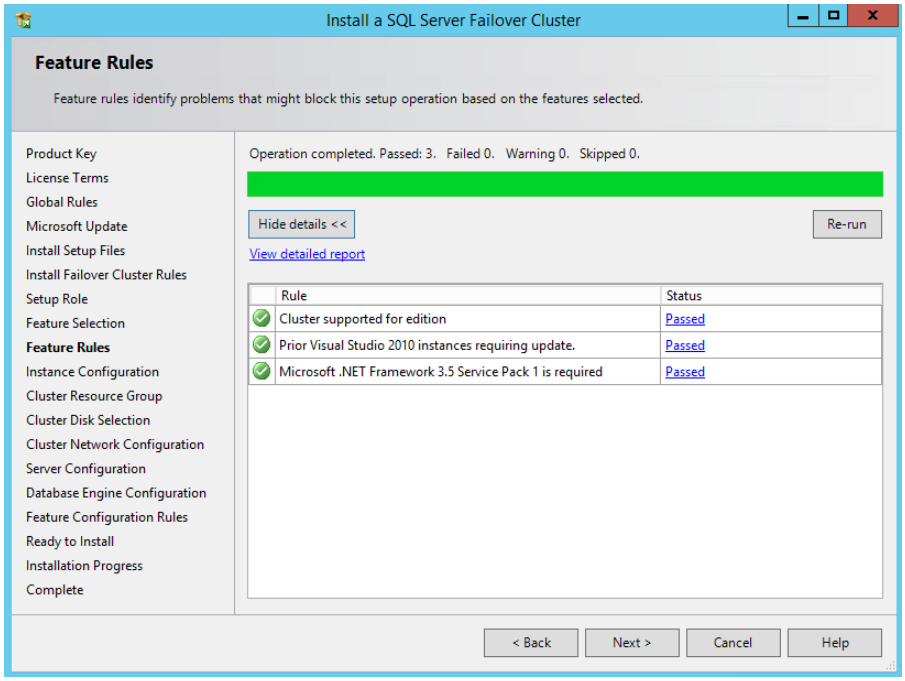
8. In the **Setup Role** dialog box, select the **SQL Server Feature Installation** option and click **Next**.



9. In the **Feature Selection** dialog box, select the following components – **Database Engine Services, Client Tools Connectivity** and **Management Tools**. Click **Next**.



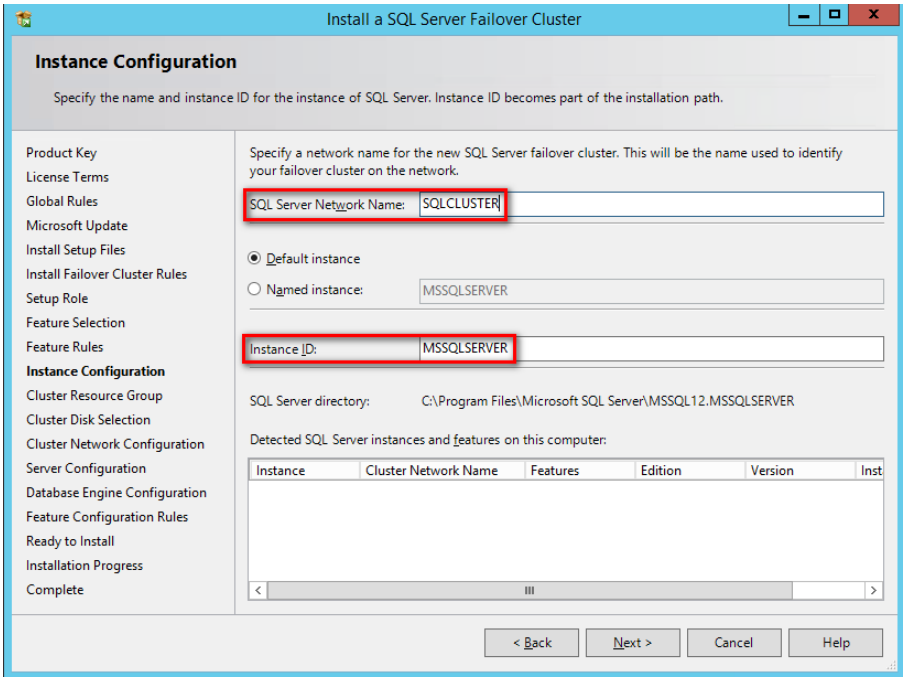
10. In the **Feature Rules** dialog box, verify that all the rules have passed. If the rules returned a few warnings, make sure you fix them before proceeding with the installation. Click **Next**.



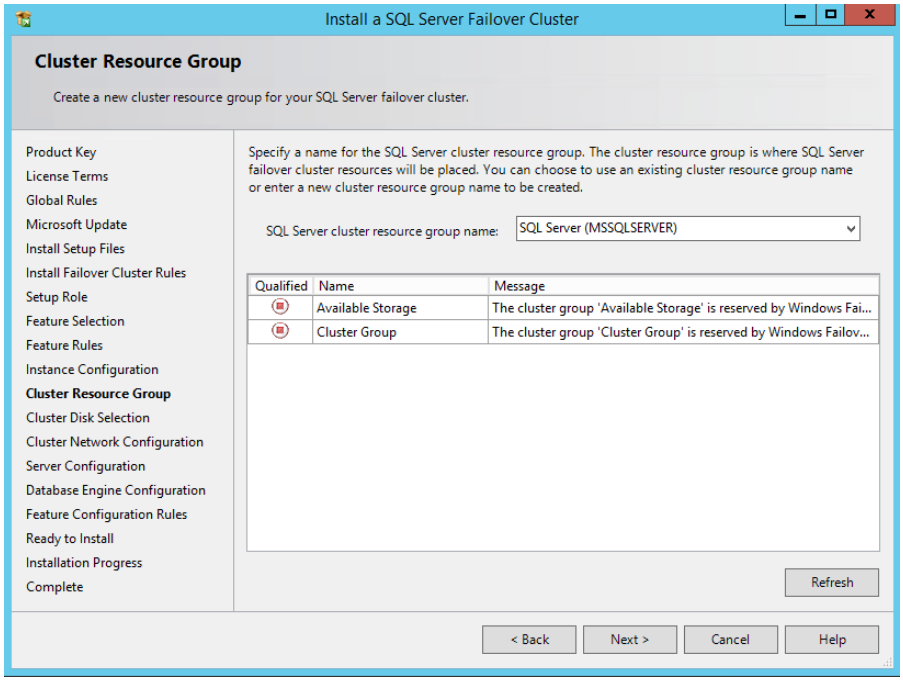
11. In the **Instance Configuration** dialog box, enter the following details:

- **SQL Server Network Name:** SQLCLUSTER
- **Instance ID**: MSSQLSERVER

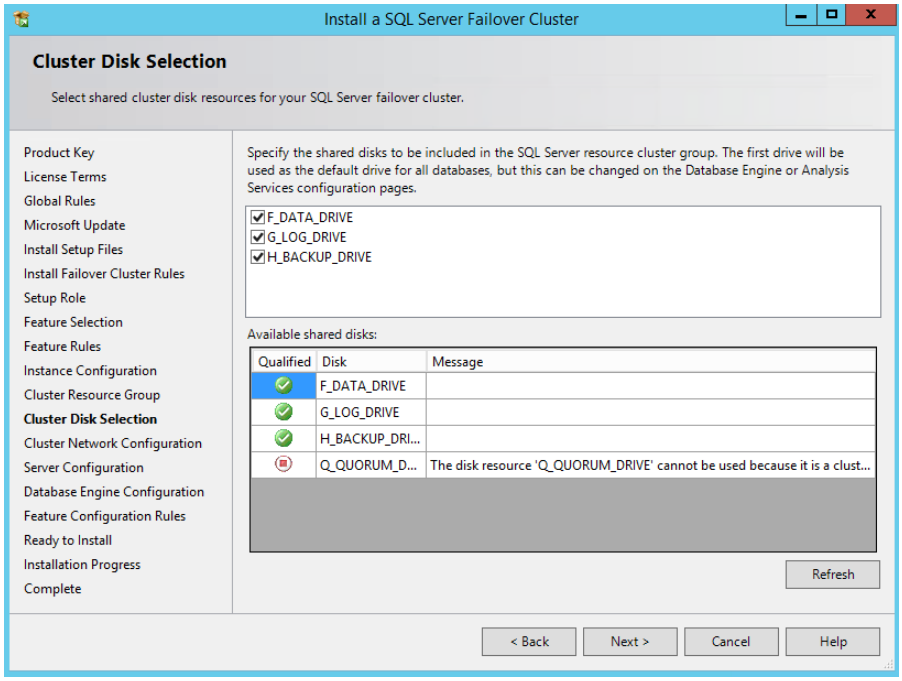
Click **Next**.



12. In the **Cluster Resource Group** dialog box, check the resources available on your Windows Server Failover Cluster. This tells you that a new Resource Group will be created on your cluster for the SQL Server instance. To specify the SQL Server cluster resource group name, you can either use the drop-down box to specify an existing group to use or type the name of a new group to create it. Accept all the defaults and click **Next**.



12. In the **Cluster Disk Selection** dialog box, select the available disk groups that are on the cluster for SQL Server 2014 to use. Click **Next**.

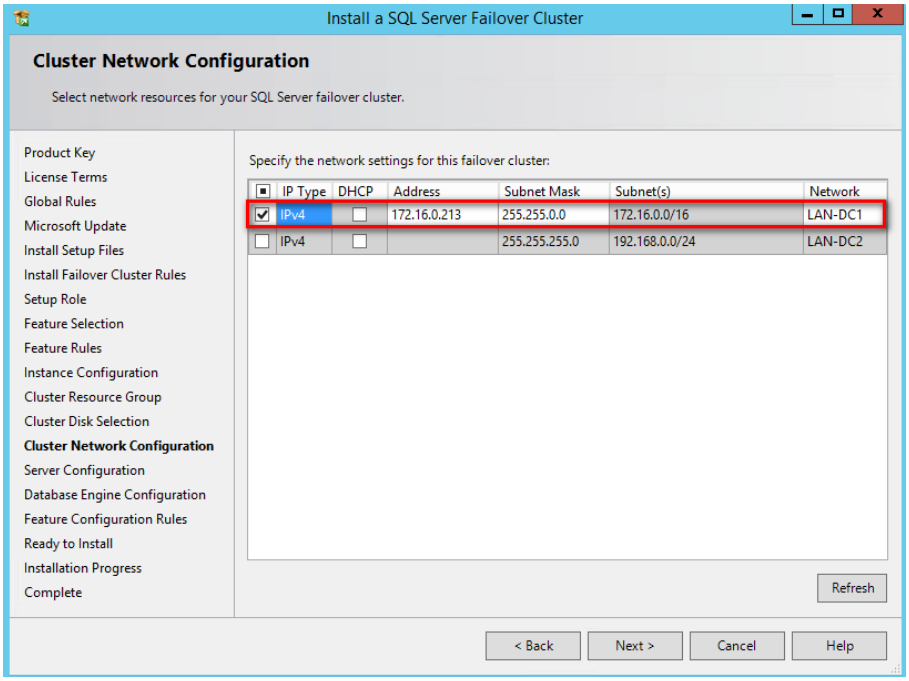


13. In the **Cluster Network Configuration** dialog box, enter the virtual IP address and subnet mask that the SQL Server 2014 cluster will use. Notice that the setup process has detected the existence of multiple network subnets. These are the names of the network adapters that have been defined in the Windows Server 2012 R2 Failover Cluster. Since the installation is performed on a cluster node that belongs to one of the network subnets, only that option will be available. The other option to assign a virtual IP address will be made available when the Add Node option is selected to install an additional node in the cluster.

    We will be using the following information for the SQL Server failover cluster instance.

Virtual Server Name	Networks	IP Address
SQLCLUSTER	172.16.0.0/24	172.16.0.213
	192.168.0.0/24	192.168.0.213

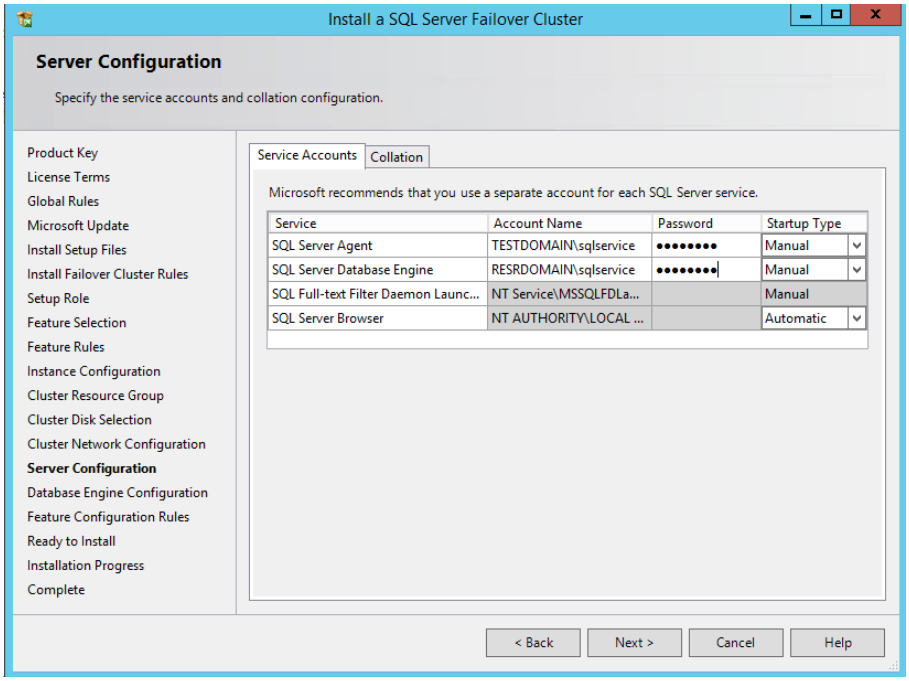
    Select the checkbox beside the IPv4 column as a static IP addresses will be used. Click **Next**.



NOTE: *The network adapter settings that will be displayed in this dialog box will depend on how the cluster network adapters are configured. Be sure to configure the iSCSI network adapters with the* ***Do not allow cluster network communication on this network*** *option.*

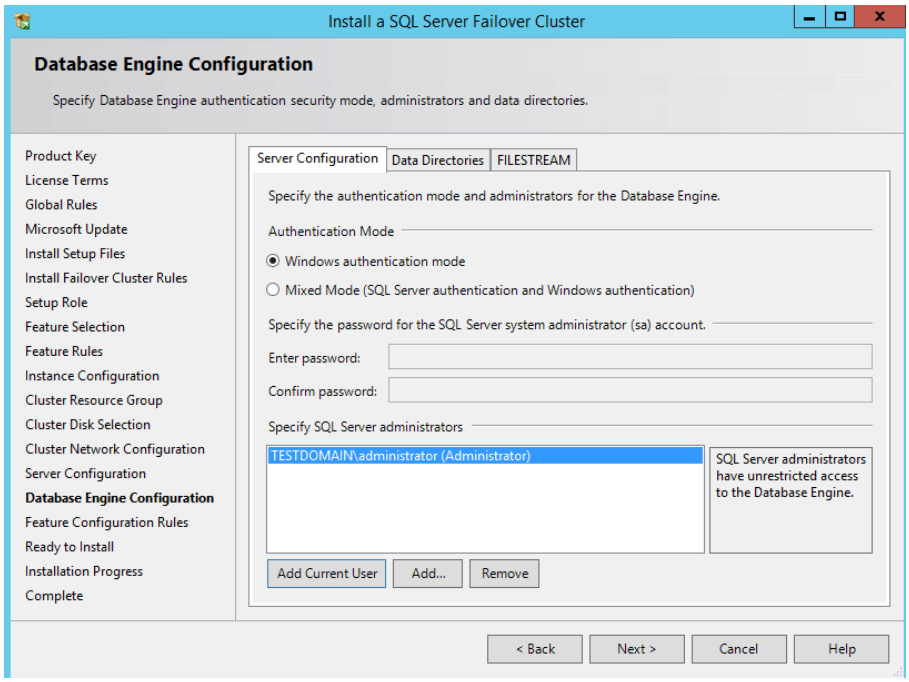
14. In the **Server Configuration** dialog box, use the following credentials for the SQL Server service accounts in the **Service Accounts** tab. Make sure that both the **SQL Server Agent** and **SQL Server Database Engine** services have a **Startup Type** of **Manual**. The Windows Server Failover Cluster will take care of stopping and starting the service. Also, set the **Collation** property for the instance according to your application requirement.

    - **SQL Server Agent:** TESTDOMAIN\sqlservice
    - **SQL Server Database Engine:** TESTDOMAIN\sqlservice



Click **Next**.

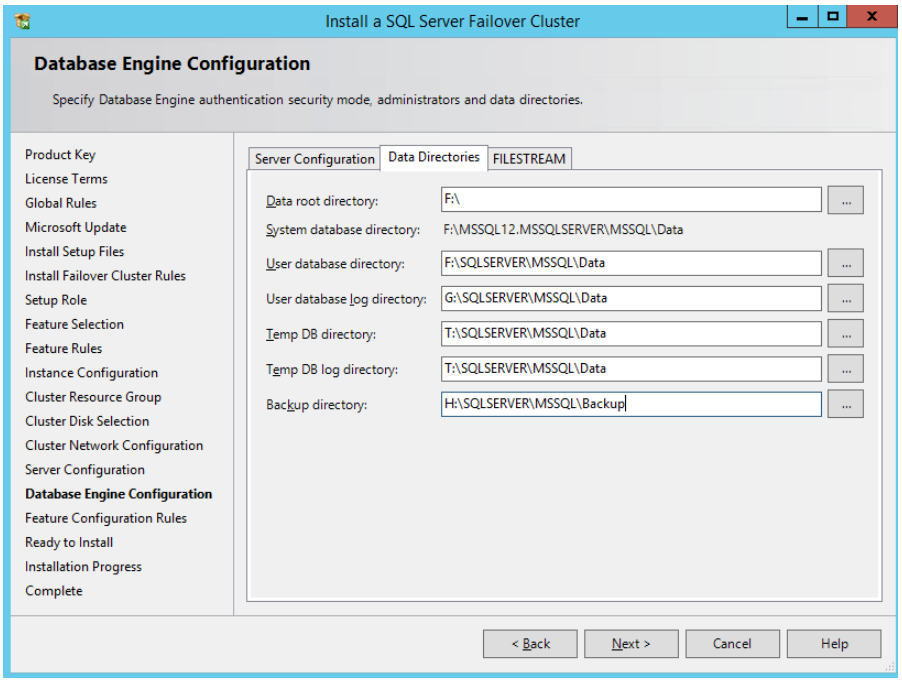
15. In the **Database Engine Configuration** dialog box, select the appropriate **Authentication Mode** in the **Server Authentication** tab. If you want to add the currently logged on user to be a part of the SQL Server administrators group, click the **Add Current User** button. Otherwise, you can add the appropriate domain accounts or security groups.



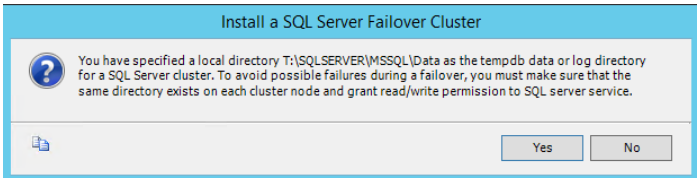
In the **Data Directories** tab, enter the following

- **Data root directory:** F:\
- **User database directory:** F:\SQLSERVER\MSSQL\Data
- **User database log directory:** G:\SQLSERVER\MSSQL\Data

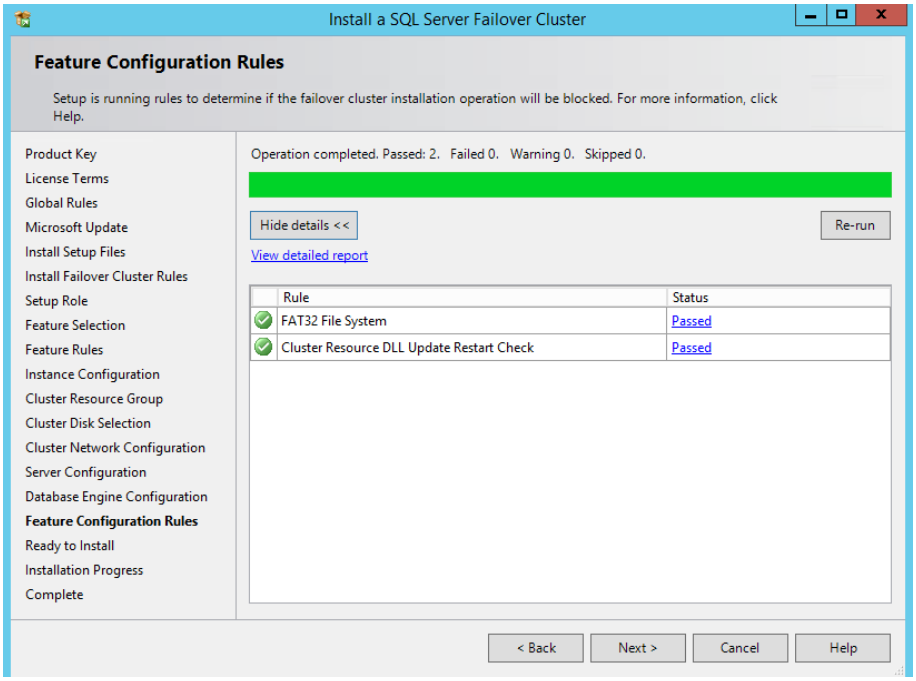
- **Temp DB directory:** T:\SQLSERVER\MSSQL\Data
- **Temp DB log directory:** T:\SQLSERVER\MSSQL\Data
- **Backup directory:** H:\SQLSERVER\MSSQL\Backup



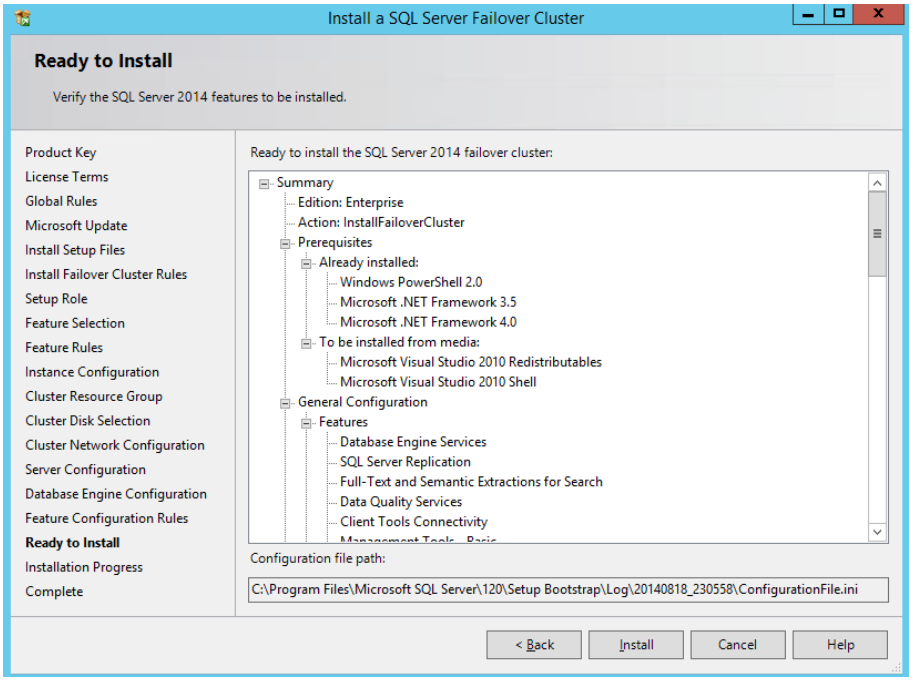
NOTE: *Introduced in SQL Server 2012 is the option to store the* ***tempdb*** *database on a local drive instead of a clustered drive. Should you decide to do so, you will get prompted to make sure that all of the nodes in the cluster contain the same directory structure and that the SQL Server service account has read/write permissions on those folders.*



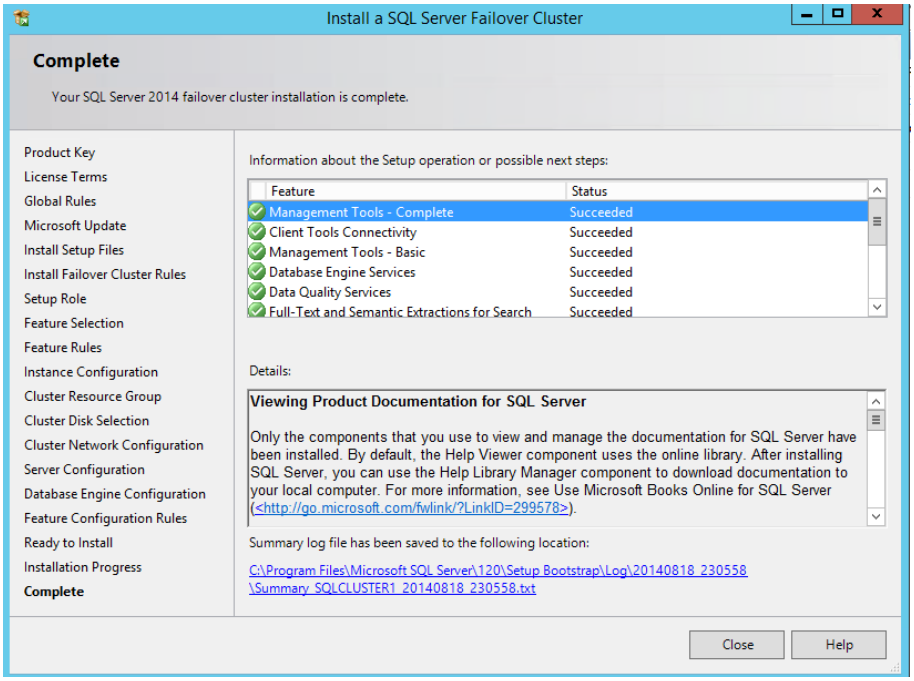
16. In the **Feature Configuration Rules** dialog box, click **Next**.



17. In the **Ready to Install** dialog box, verify that all configurations are correct. Click **Next**.



14. Once the installation finishes, in the **Complete** dialog box, click **Close**.

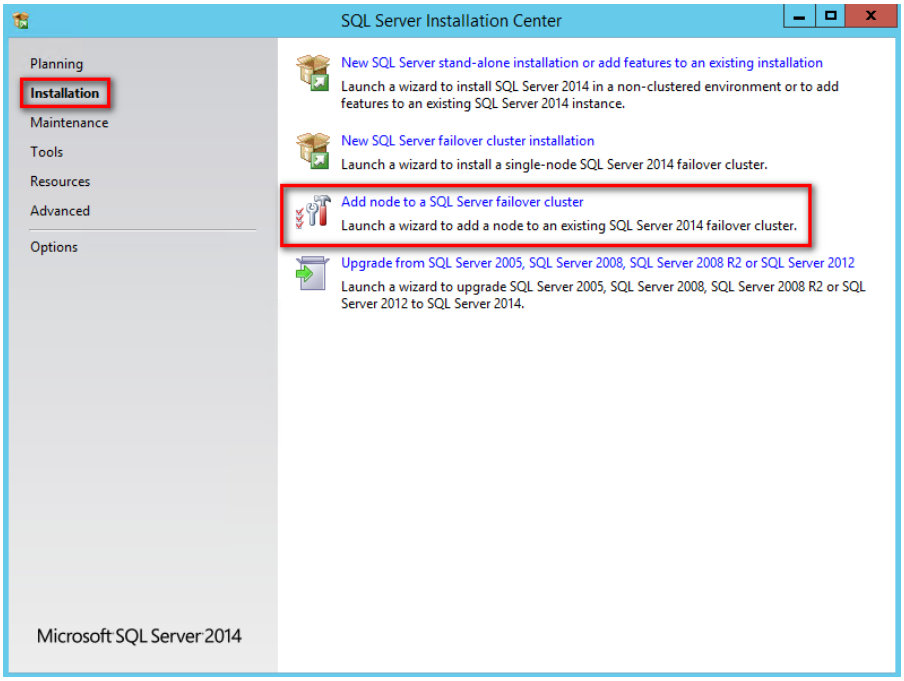

Install a SQL Server Failover Cluster
Complete
Your SQL Server 2014 failover cluster installation is complete.
Product Key
License Terms
Global Rules
Microsoft Update
Install Setup Files
Install Failover Cluster Rules
Setup Role
Feature Selection
Feature Rules
Instance Configuration
Cluster Resource Group
Cluster Disk Selection
Cluster Network Configuration
Server Configuration
Database Engine Configuration
Feature Configuration Rules
Ready to Install
Installation Progress
Complete
Information about the Setup operation or possible next steps:
Feature
Status
Management Tools - Complete
Succeeded
Client Tools Connectivity
Succeeded
Management Tools - Basic
Succeeded
Database Engine Services
Succeeded
Data Quality Services
Succeeded
Full-Text and Semantic Extractions for Search
Succeeded
Details:
Viewing Product Documentation for SQL Server
Only the components that you use to view and manage the documentation for SQL Server have been installed. By default, the Help Viewer component uses the online library. After installing SQL Server, you can use the Help Library Manager component to download documentation to your local computer. For more information, see Use Microsoft Books Online for SQL Server (<http://go.microsoft.com/fwlink/?LinkID=299578>).
Summary log file has been saved to the following location:
C:\Program Files\Microsoft SQL Server\120\Setup Bootstrap\Log\20140818_230558\Summary_SQLCLUSTER1_20140818_230558.txt
Close
Help

Adding a Node on a SQL Server 2014 Multi-Subnet Cluster

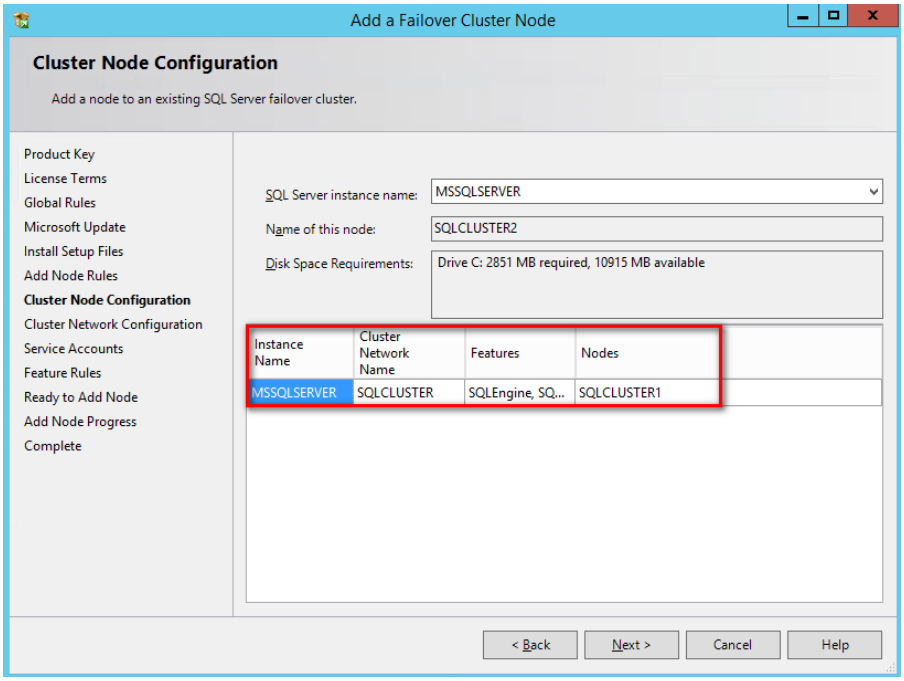
In this section, we will add a node to the SQL Server 2014 failover clustered default instance on a multi-subnet Windows Server Failover Cluster. We will run the installation process on the second node of the cluster, **SQLCLUSTER2**.

To add a node on a SQL Server 2014 multi-subnet failover clustered instance:

1. Run **setup.exe** from the installation media to launch SQL Server Installation Center.
2. Click on the **Installation** link on the left-hand side. Click the **Add node to a SQL Server failover cluster** link. This will run the SQL Server 2014 Setup wizard.

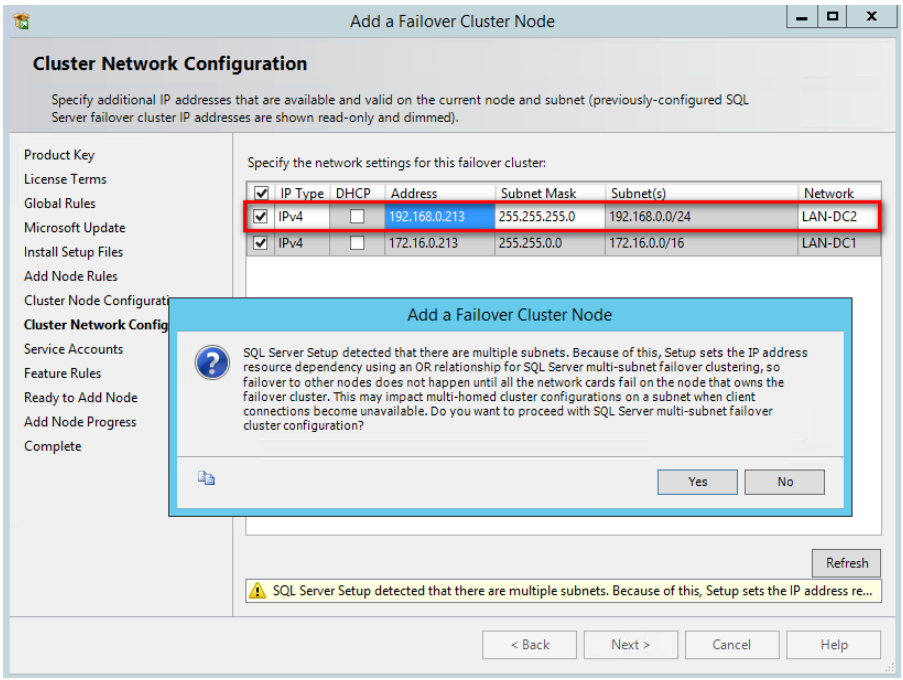


3. In the **Product Key** dialog box, enter the product key that came with your installation media and click **Next**.
4. In the **License Terms** dialog box, click the **I accept the license terms** check box and click **Next**.
5. In the **Global Rules** dialog box, validate that the checks return successful results and click **Next**.
6. In the **Microsoft Update** dialog box, click **Next**.
7. In the **Add Node Rules** dialog box, validate that the checks return successful results. If the checks returned a few warnings, make sure you fix them before proceeding with the installation. Click **Next**.
8. In the **Cluster Node Configuration** dialog box, validate that the information for the existing SQL Server 2014 failover clustered instance is correct. Click **Next**.

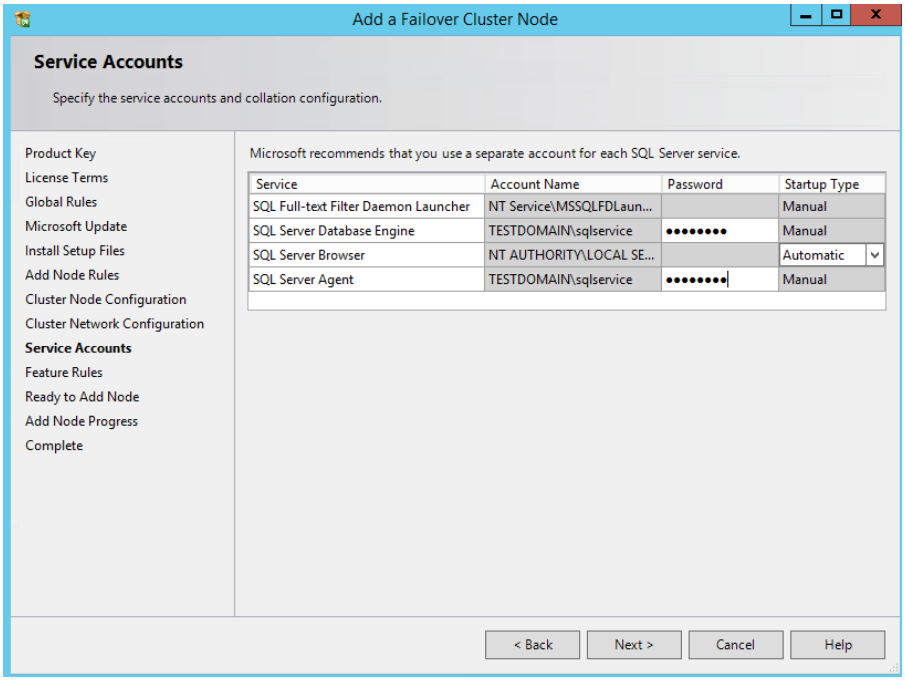


9. In the **Cluster Network Configuration** dialog box, enter the virtual IP address and subnet mask that the SQL Server 2014 failover cluster instance will use in the network subnet that the second node is in - **192.168.0.213**. Notice that the setup process also detected the existence of two network subnets. Since the virtual IP address for the **172.16.0.0/16** subnet has already been configured, that option has been disabled.

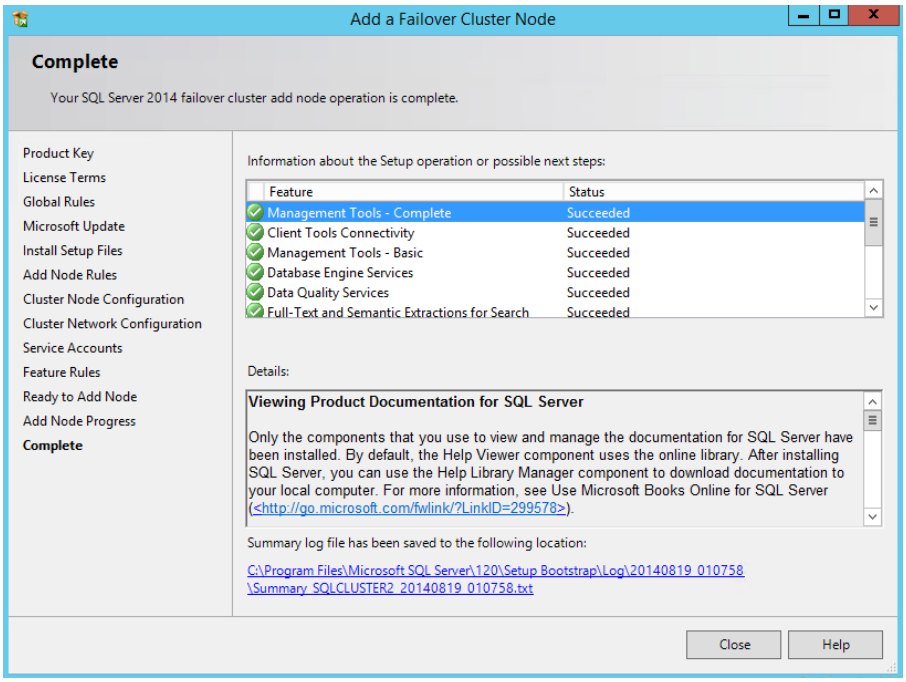
   NOTE: *A message box that gives you a brief explanation of how the OR logic dependency works will be displayed. Click the* ***Yes*** *button in the message box. Click* ***Next****.*



10. In the **Service Accounts** dialog box, verify that the information is the same as what was used to configure the first node. Click **Next**.



11. In the **Feature Rules** dialog box, click **Next**.
12. In the **Ready to Add Node** dialog box, verify that all configurations are correct and click **Install**.
13. Once the installation finishes, in the **Complete** dialog box, click **Close**. This concludes adding a node to a SQL Server 2014 Multi-Subnet Cluster.



NOTE: *When storing the* ***tempdb*** *database in a local drive instead of a clustered drive, be sure that:*

- *The same drive letter and folder structure exists in all of the nodes in the cluster*
- *The SQL Server service account has the appropriate permissions on the folder where* ***tempdb*** *will be created*

Tuning the SQL Server 2014 Failover Clustered Instance DNS Settings

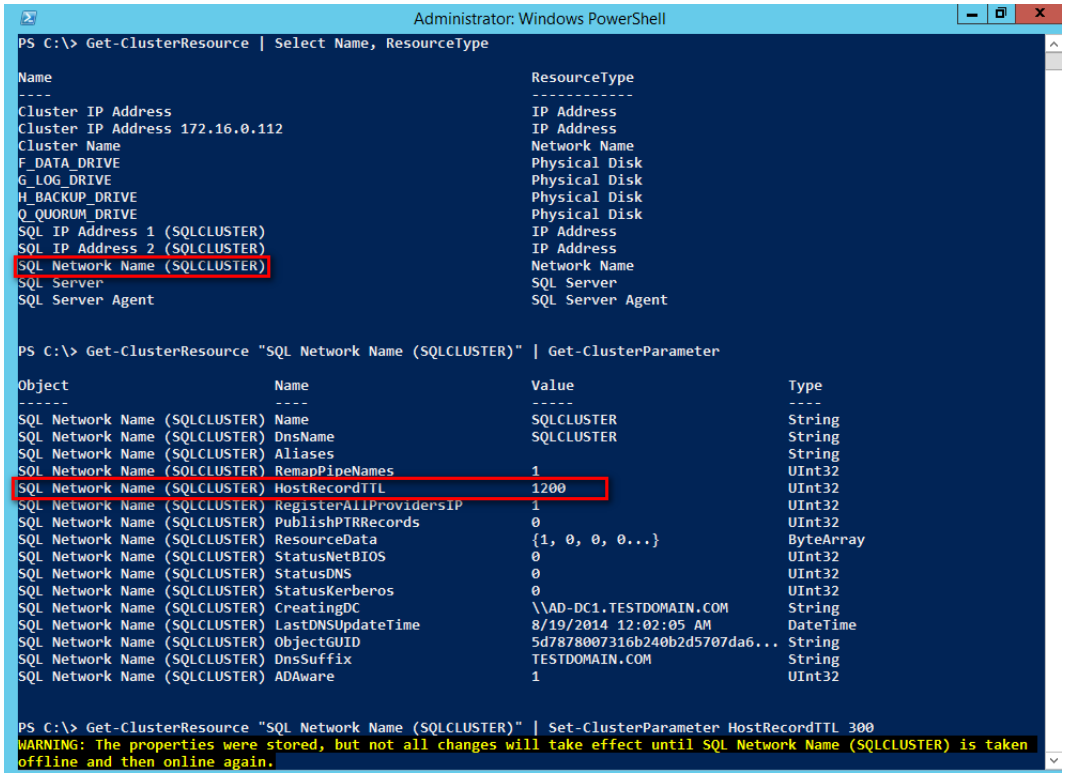
In this section, we will tune the SQL Server 2014 failover clustered instance DNS settings for multi-subnet clusters. We will use Windows PowerShell to perform the following tasks.

NOTE: *Client workstations and applications cache DNS entries for a period of time before checking with the DNS server to see if the name resolution has changed. This is called the Time-To-Live (TTL) value and, for cluster resources, the default value is 1200 seconds, or 20 minutes. This can significantly impact recovery time objective (RTO.) We can decrease the DNS TTL value of the virtual server name for the SQL Server 2014 failover clustered instance to 300 seconds or 5 minutes by changing the* ***HostRecordTTL*** *property value. Discuss this with your network engineers to make sure that they understand the impact of the change to the overall network infrastructure.*

These steps can be performed on either of the nodes in the failover cluster. The steps below are performed on ***SQLCLUSTER1****.*

1. Open the **Windows PowerShell** console in *Administrator mode*
2. Type the following command. This will change the DNS TTL value of the virtual server name for the SQL Server 2014 failover clustered instance to **300** seconds (5 minutes).

```
PS C:\>#List different cluster resources
PS C:\>Get-ClusterResource | Select Name, ResourceType
PS C:\>#List parameters and their values of the SQL Server Network name
PS C:\>Get-ClusterResource "SQL Network Name (SQLCLUSTER)" | Get-ClusterParameter
PS C:\>#Set parameter value
PS C:\>Get-ClusterResource "SQL Network Name (SQLCLUSTER)" | Set-ClusterParameter
HostRecordTTL 300
```



3. Take the virtual server name for the SQL Server 2014 failover clustered instance offline and back online for the changes to take effect.

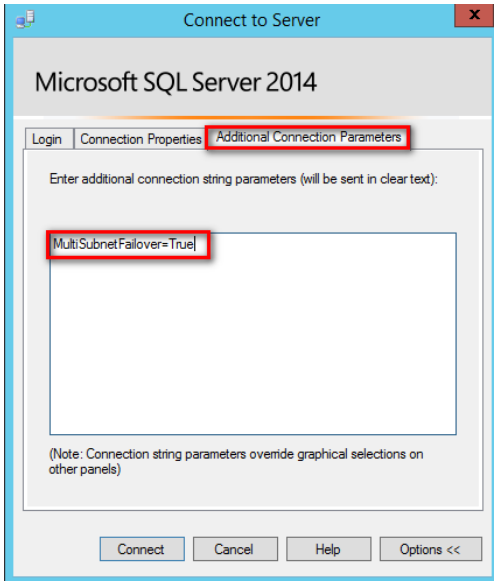
Testing Application Connectivity

In this section, we will test application connectivity for SQL Server 2014 multi-subnet failover clustered instance. We will use **SQL Server 2014 Management Studio** and **SQLCMD** to perform the following tasks.

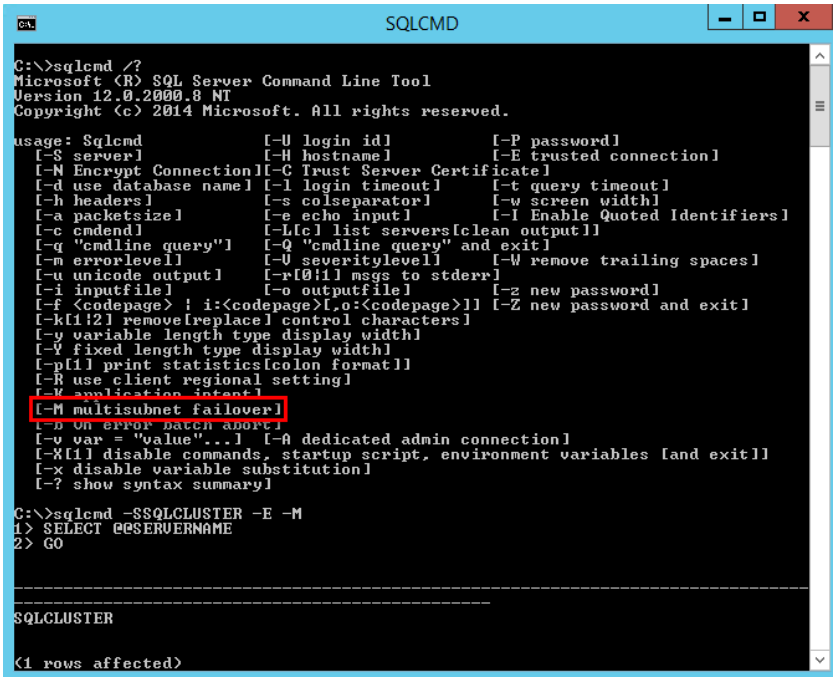
NOTE: In order for client applications to be automatically redirected during a cluster failover, they need to either be using

1. *the SQL Server 2012 Native Client or higher*
2. *the Data Provider for SQL Server in .NET Framework 4.02 or above*
3. *the Microsoft JDBC Driver 4.0 for SQL Server*

A new connection string parameter named **MultiSubnetFailover** is made available to allow applications to simultaneously try all the IP addresses assigned for the SQL Server 2014 multi-subnet failover clustered instance name and connects to the first one that responds. The parameter can be used with SQL Server Management Studio under the **Additional Connection Parameters** tab.



The **-M** parameter in **sqlcmd** can also be used as shown below.


SQLCMD
C:\>sqlcmd /?
Microsoft (R) SQL Server Command Line Tool
Version 12.0.2000.8 NT
Copyright (c) 2014 Microsoft. All rights reserved.
usage: Sqlcmd [-U login id] [-P password]
[-S server] [-H hostname] [-E trusted connection]
[-N Encrypt Connection][-C Trust Server Certificate]
[-d use database name] [-l login timeout] [-t query timeout]
[-h headers] [-s colseparator] [-w screen width]
[-a packetsize] [-e echo input] [-I Enable Quoted Identifiers]
[-c cmdend] [-L[c] list servers[clean output]]
[-q "cmdline query"] [-Q "cmdline query" and exit]
[-m errorlevel] [-V severitylevel] [-W remove trailing spaces]
[-u unicode output] [-r[0|1] msgs to stderr]
[-i inputfile] [-o outputfile] [-z new password]
[-f <codepage> | i:<codepage>[,o:<codepage>]] [-Z new password and exit]
[-k[1|2] remove[replace] control characters]
[-y variable length type display width]
[-Y fixed length type display width]
[-p[1] print statistics[colon format]]
[-R use client regional setting]
[-K application intent]
[-M multisubnet failover]
[-b on error batch abort]
[-v var = "value"...] [-A dedicated admin connection]
[-X[1] disable commands, startup script, environment variables [and exit]]
[-x disable variable substitution]
[-? show syntax summary]
C:\>sqlcmd -SSQLCLUSTER -E -M
1> SELECT @@SERVERNAME
2> GO
SQLCLUSTER
(1 rows affected)

About The Author

Edwin M Sarmiento is a **Microsoft SQL Server MVP** and **Microsoft Certified Master** from Ottawa, Canada specializing in high availability, disaster recovery and system infrastructures running on the Microsoft server technology stack - ranging from Active Directory to SharePoint and anything in between. He is very passionate about technology but has interests in music, professional and organizational development, leadership and management matters when not working with databases. He lives up to his primary mission statement – "*To help people grow and develop their full potential as God has planned for them*."

He wants the whole world to know that the FILIPINO is a world-class citizen and brings JESUS CHRIST to the world.

http://www.EdwinMSarmiento.com
edwin@EdwinMSarmiento.com
@EdwinMSarmiento
http://ca.linkedin.com/in/EdwinMSarmiento

Did this document help you? Send feedback, comments and suggestions to:
edwin@EdwinMSarmiento.com

Interested in requesting the author for training, consulting, solutions architecture, whiteboard sessions or team mentoring, send an email to: **edwin@EdwinMSarmiento.com**