Building A SQL Server 2008 R2 Cluster: Part I

Installation of Windows Cluster is always a challenging task. Most of the time, we didn't get enough opportunities to install an entire cluster system on our own due to limitation of hardware/software. With the help of desktop virtualization software, now its possible to build a windows cluster on a desktop environment. The purpose of this series is to create a step-by-step deployment guide on how to build a windows Cluster and install SQL Server 2008 on it. This series is created for learning and testing purpose.

Software Used to Create the SQL Server 2008 R2 Cluster:

For the virtual environment to create the windows cluster on a client OS, we are going to use Oracle VM. Oracle VM is a free desktop virtualization software available for both 32-bit as well as 64-bit platform which support 64-bit guest operating system. If you are new to Oracle VM and don't know how to create a VM using it, you can have a look on the following article.

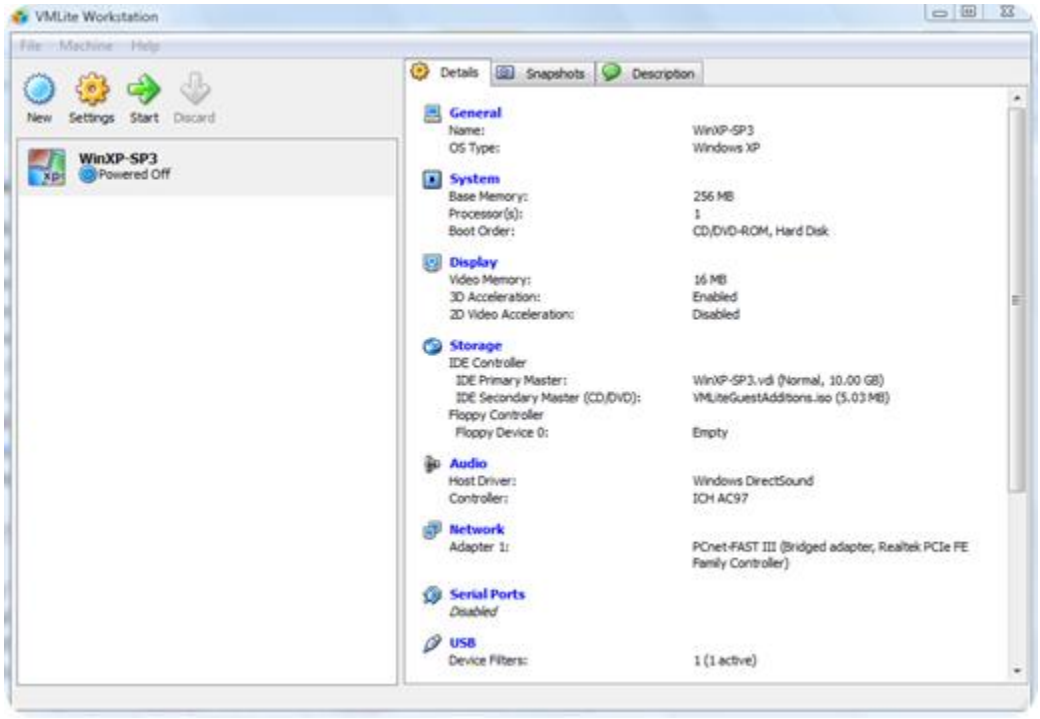
Creating a Virtual Machine using VMLite Workstation

Virtualization is a hot topic in today's IT industries. The main objectives behind the virtualization concepts are consolidation of multiple servers into a single server, which in turns reduced the total cost of ownership along with less power consumption, cooling, and manageability cost. VMware's ESX, Microsoft's Hyper-V, Citrix's Xen servers are the mostly used products when we are discussing about server virtualization. Unlike servers, sometimes, we might need to have virtual machines running on out of our desktop, while we are testing a new feature of an existing software, or installing a new software to try its functionality. I am sure, many of you, are already using some types of desktop virtualization software. Some of the available desktop virtualization software are Microsoft's Virtual PC, VMware's VMWare Workstation, Oracle's Oracle VM, VMLite's VMLite Workstation. In this post, we are going to create a Virtual Machine using VMLite Workstation.

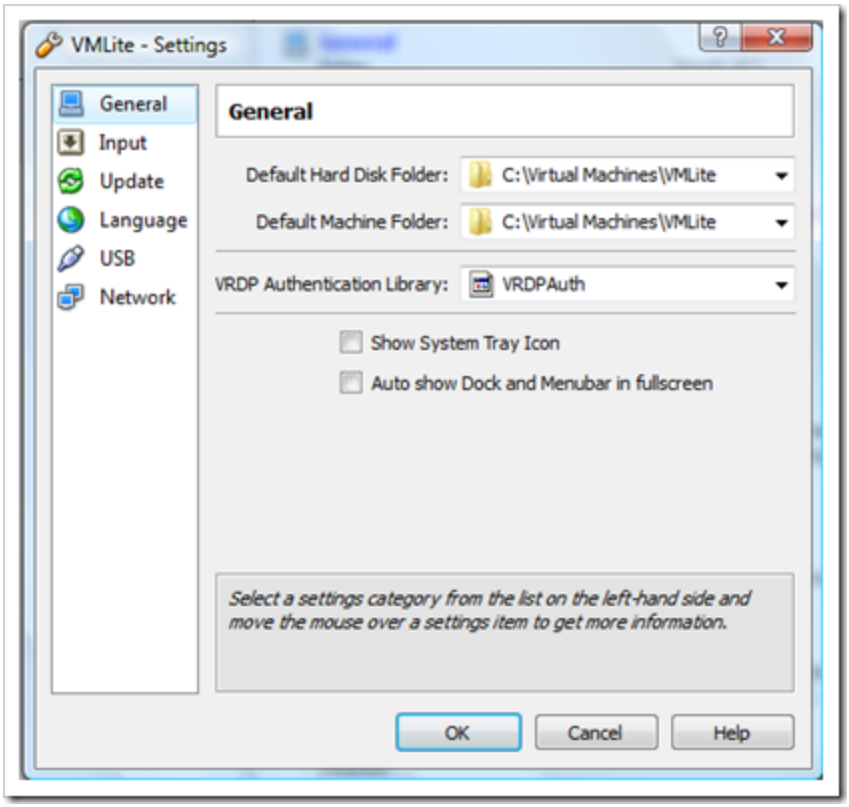
About VMLite Workstation

VMLite Workstation is a free desktop virtualization software available both for 32-bit & 64-bit platform. You can run this software on your existing Windows XP/Vista/7 machine. It also supports 64-bit guest OS, the feature which was lacking with many desktop virtualization software. The installation of VMLite is very easy, mostly you click on the *Next* button to continue the installation. Currently, I am running VMLite Workstation on a Windows Vista 64-bit machine. Once you installed VMLite Workstation, you can open it from your system's Start

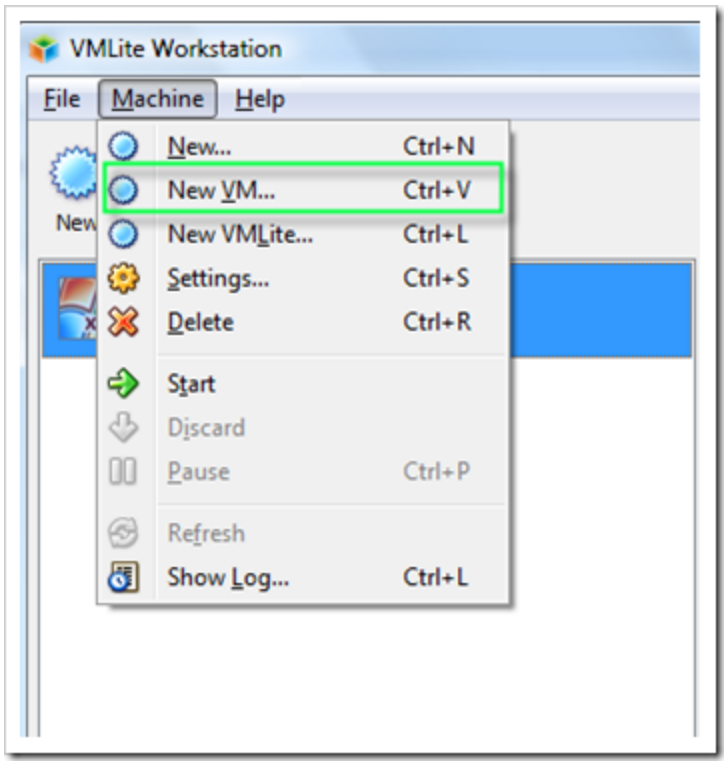
Menu (Start –> Programs –> VMLite Workstation –>VMLite Workstation). You will get the following screen after you lunch VMLite Workstation.



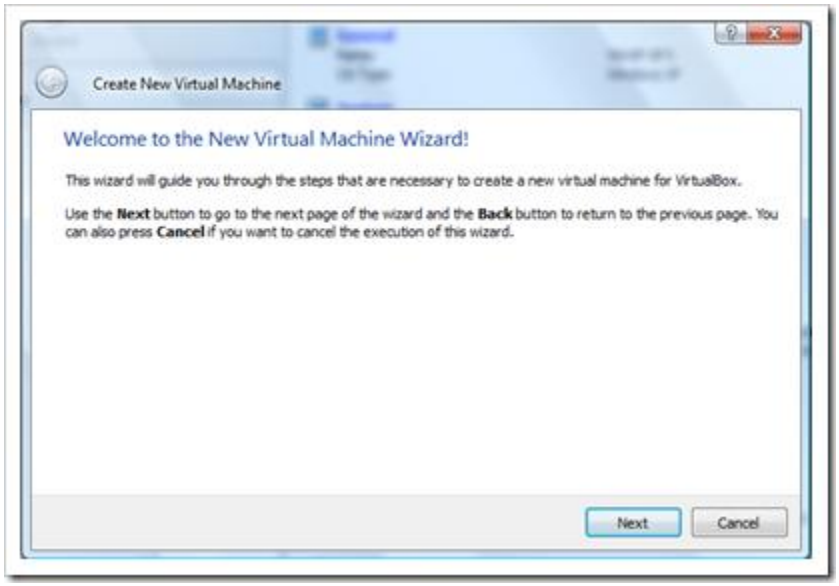
After you open the VMLite Workstation, click on the File Menu and click on *Preferences.* A new window will open, with a few options as mentioned below. While on the General tab, you can specify the location of your default Virtual Machines folder & its virtual hard disk folder. By default, the virtual hard disk extension for VMLite Workstation is ***.VDI***.



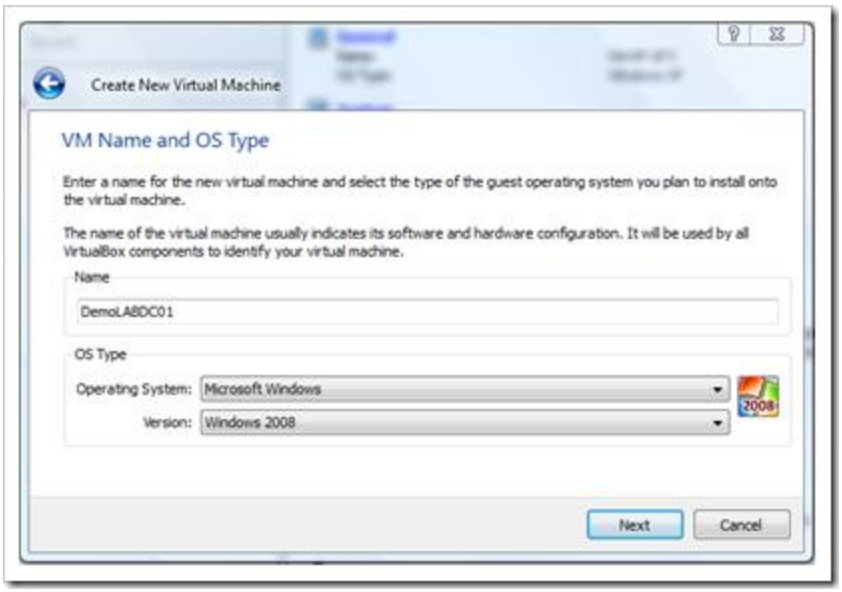
Once you specify a location for your virtual machine & its virtual disks, click on Ok to close the Preferences window. Since we have made the necessary changes, now we will look into how to create a virtual machine using VMLite. Open the Machine tab of VMLite Workstation and click on *New VM* to create a new virtual machine.



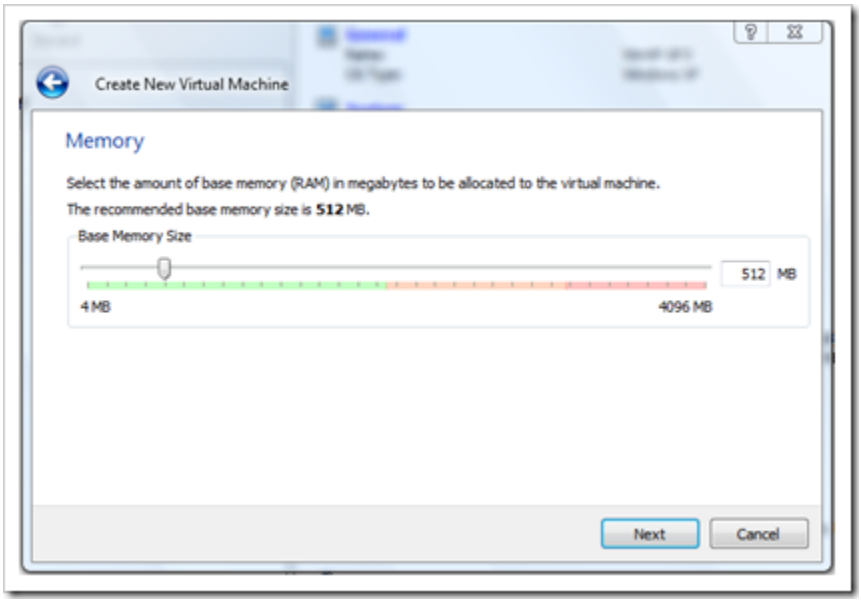
You will get the welcome screen, click on *Next* to continue.



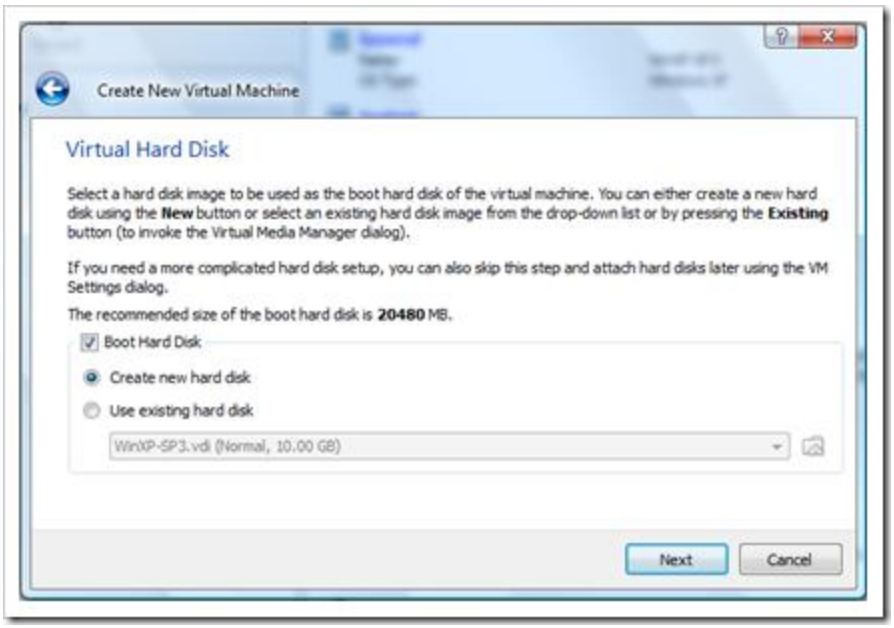
Now, we need to specify the name of the Virtual Machine and choose the OS we want to install. VMLite Workstation supports many different versions of Windows & Linux distro. In this post, we are going to create a virtual machine (VM) for Windows Server 2008. Click on *Next* to continue.



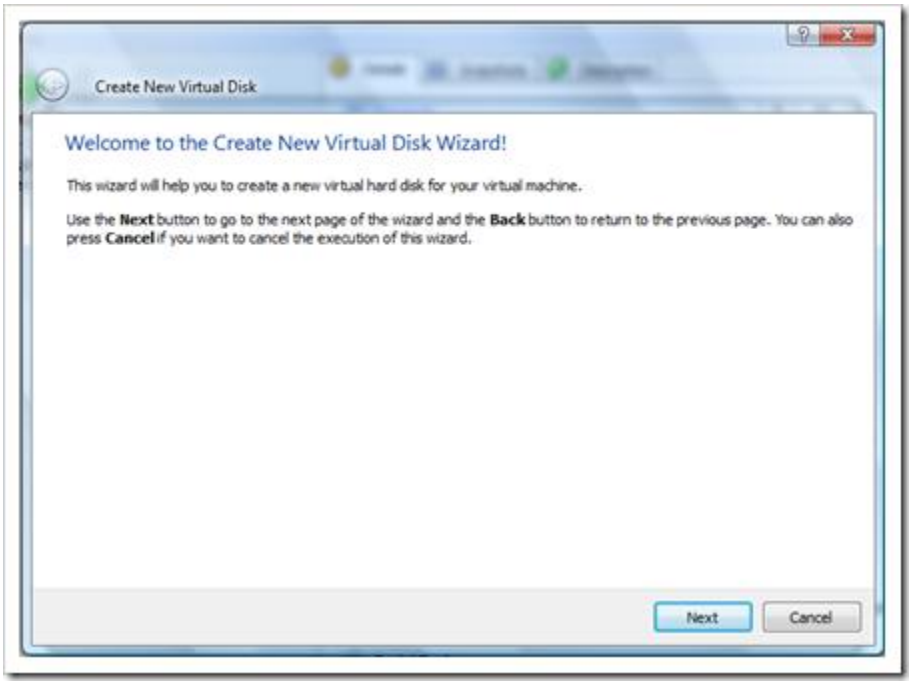
Once we choose the desired operating system, we need to assign the memory for this VM. I am going to assign 512MB of memory for my current VM, *Next* to continue.



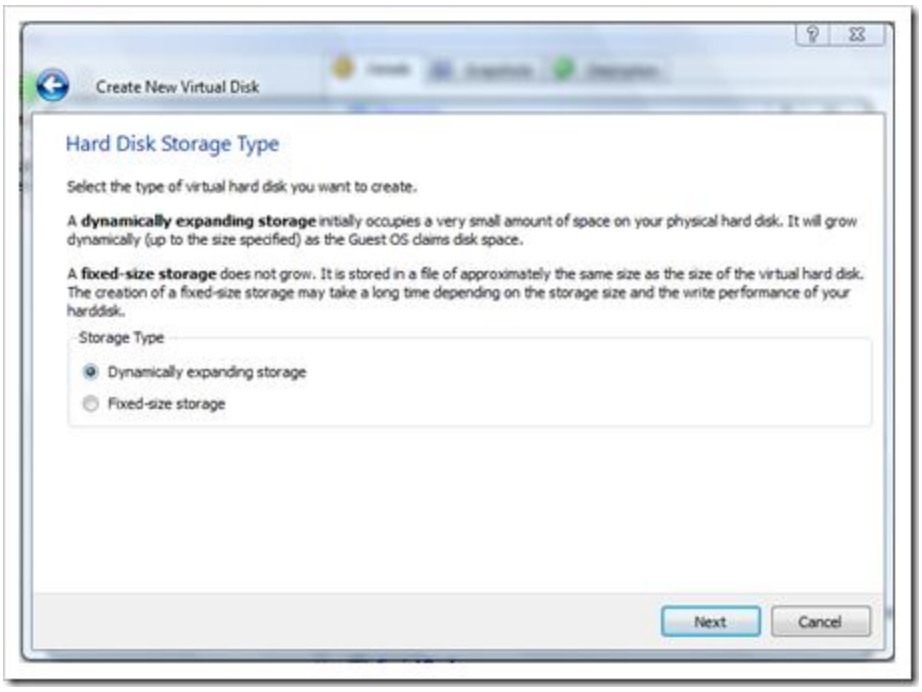
Now that we have chosen the desired OS and the amount of memory for our VM, its time to create a disk for our machine. If you have an existing virtual hard disk, you can choose it from the drop down menu else to continue to create a new virtual hard disk.



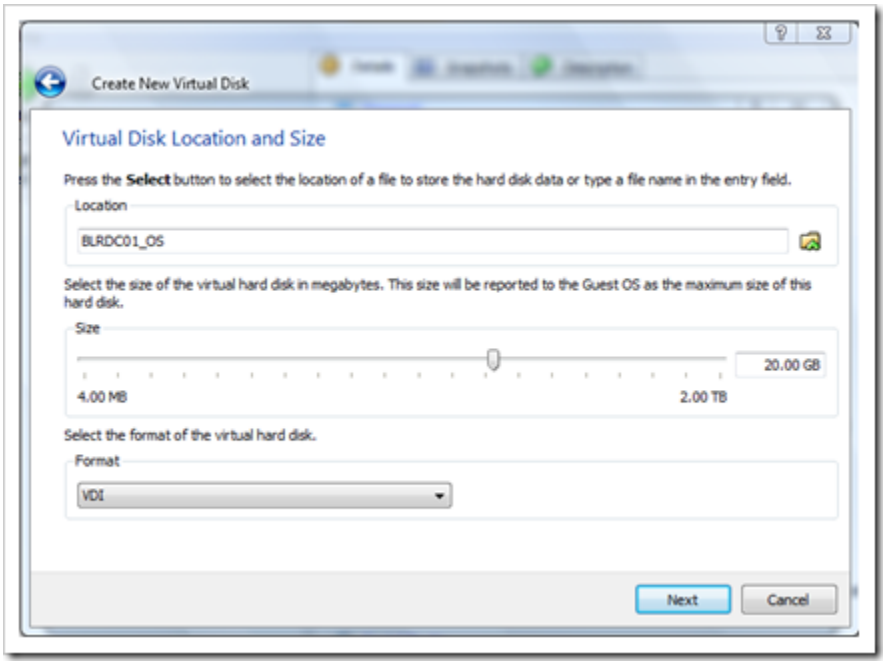
Click on *Next* on the Virtual disk wizard to continue.



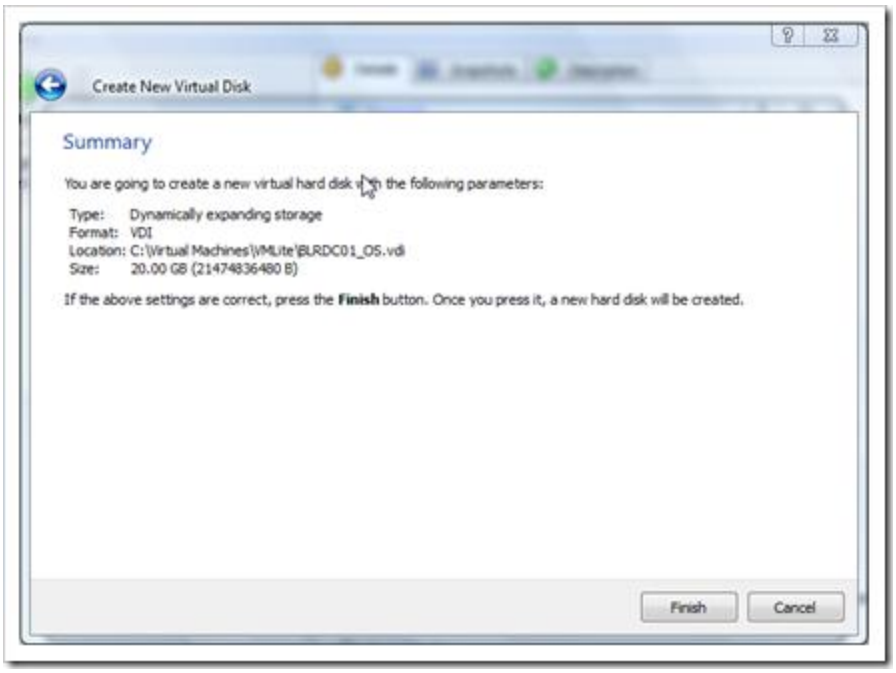
You need to choose the type of virtual disk for your VM. A *Dynamic Virtual disk* grows dynamically till its occupies maximum available free spaces available on the disk. A *Fixed-Size Storage* occupy all the assigned spaces immediately from the physical drive. Choose the type of disk drive you want to use and click on *Next* to continue



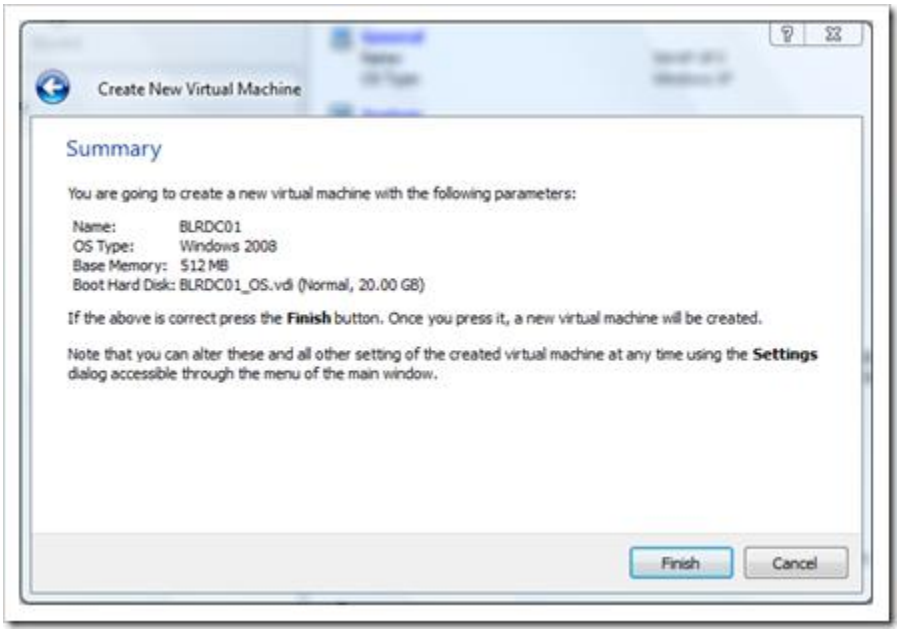
Specify the size and location of the virtual hard drive and its extension. VMLite Workstation supports the following extension for virtual disks, i.e. VDI, VHD, & VMDK. Once you enter all the information, click on *Next* to continue.



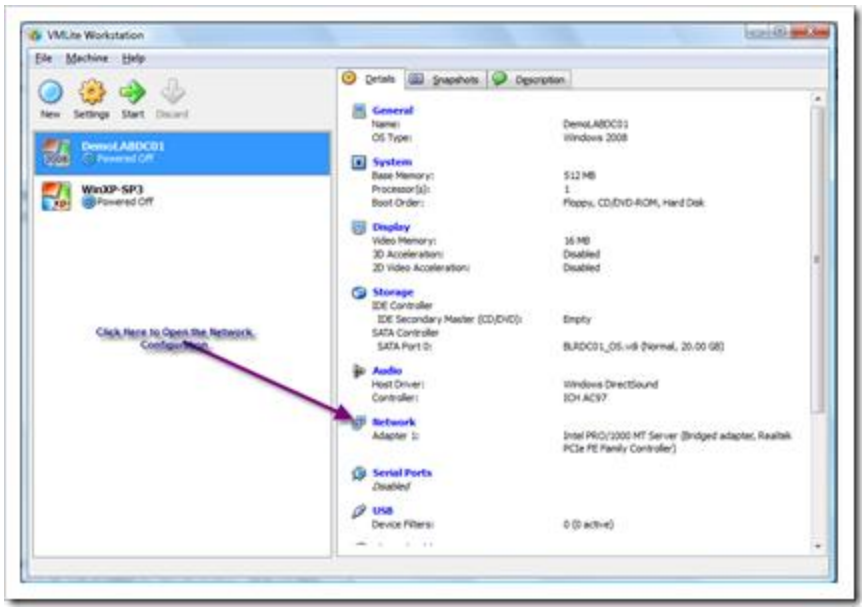
A summary of the details about the virtual hard disk will be displayed. Click on *Finish* to create the virtual hard disk for the VM.



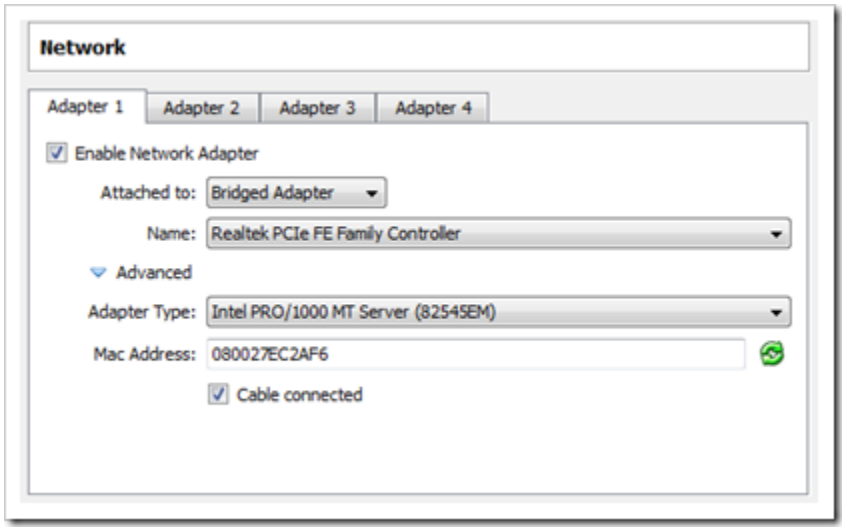
Once the virtual disk is created, you will provided the detailed summary about the configuration of your virtual machine. Click on *Finish* to create the virtual machine.



Now we have created a virtual machine, lets configure a network adapter for this machine. On the details tab in VMLite Workstation, click on *Network* to open the Network Configuration.



Configure the Network Adapter as mentioned below. If you create multiple VMs, with this network configuration, all the VMs will be able to communicate each other, if all the present in the same subnet.



VMLite Workstation 3.2.x supports the USB devices, you will be able to access your portable drives inside VMs. In a future post, we will discuss how to access a USB drive using VMLite Workstation.

I hope, this post will be helpful if you are trying to configure a virtual machine using VMLite workstation. If you are already well versed with any one of the desktop virtualization software, I am sure, you will find the implementation of VMLite Workstation is very easy.

The above article was developed using VMLite, however the same steps are applicable for Oracle VM as well. We are going to use the evaluation edition of Windows Server 2008 & SQL Server 2008 R2 to build the cluster, you can download the same from the MS website. For the storage requirement, we are going to use FreeNAS, a free storage solution to build our cluster. You can download all these software from the links provided above.

Before, we start the actual installation, let's have a quick look on our existing virtualization environment. To build a Windows Cluster, we need a Domain Controller (DC) present in our network along with a DNS server, which will help us for Name resolution. For this series, we build a domain named ***DemoLab.Local***. So we are going to create a DC named **BLRDC01**, which will be also used for DNS Name resolution. Then we are going to build a storage server in our environment using FreeNAS, and provided the disks to the cluster nodes. Next, we are going to build a two-node Windows Cluster using Windows Server 2008.

For this series of post, I am using the IP Address series 172.16.10.x/255.255.0.0 for our Public LAN. Before building the windows cluster, we will add the Backup LAN & Heartbeat LAN on our Windows Cluster Node. I have already build three virtual machines, installed Windows server 2008 evaluation on it and promoted one of them to a domain controller with AD-integrated DNS . The machine name and initial configuration as follows:

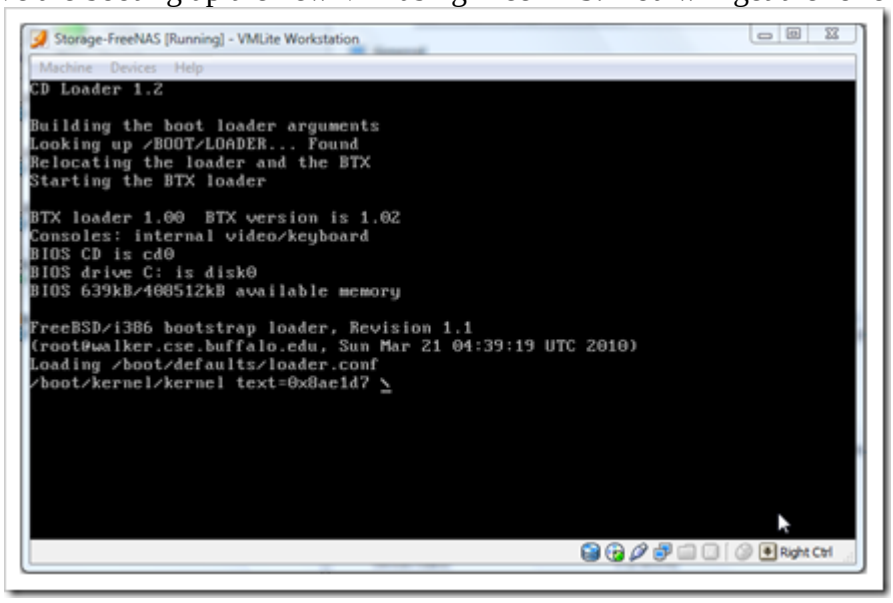
VM Name	Operating System	Functionality	Public LAN IP
BLRDC01	Windows 2008 Eval	DC, DNS	172.16.10.1/255.255.0.0
BLRC100	Windows 2008 Eval	Cluster Node 1	172.16.10.100/255.255.0.0
BLRC101	Windows 2008 Eval	Cluster Node 2	172.16.10.101/255.255.0.0

After we configured all these three machines, we will add both the servers to the *DemoLab.Local* network. Now that we have added both the servers to the *DemoLab* network, we are going to install the storage server.

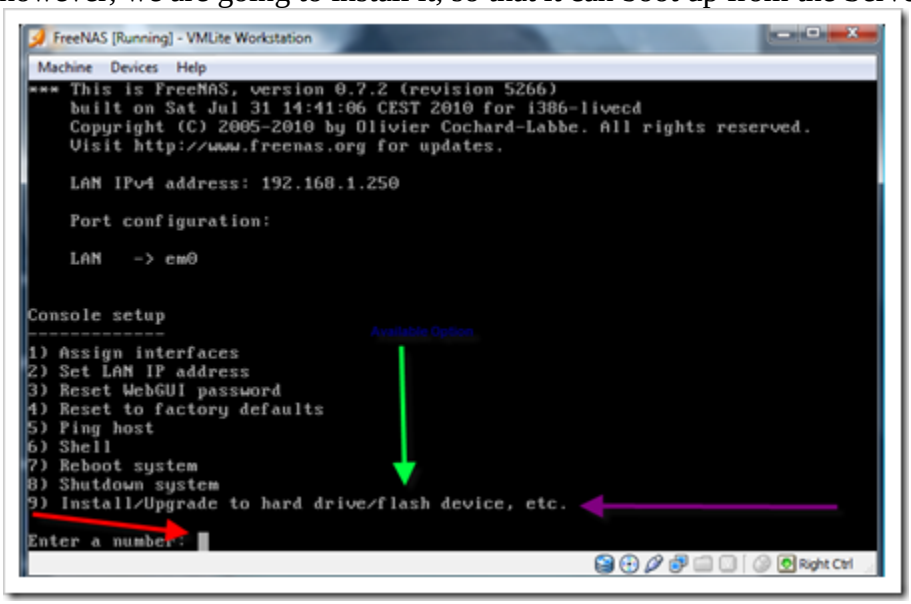
Installing the Storage Server:

Create a new virtual machine using VMLite as a storage server. While you are asked to choose the type of OS and version, select **BSD** as Operating System and version as **FreeBSD**. Our storage server is based on FreeBSD. For this demo environment, I have given 400MB of RAM

of our storage server. We are booting up the new VM using FreeNAS. You will get the following screen, during boot up:

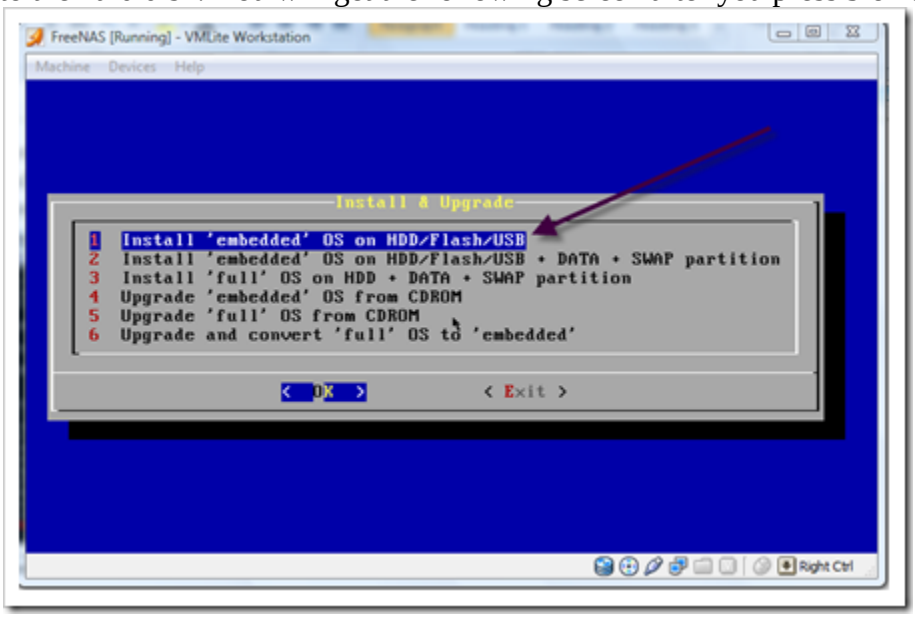


Once the storage server booted up, you will get the following screen. Currently, I am using the Live CD of FreeNAS, which will let me to use storage server without loading the entire OS, however, we are going to install it, so that it can boot up from the Server hard disk.

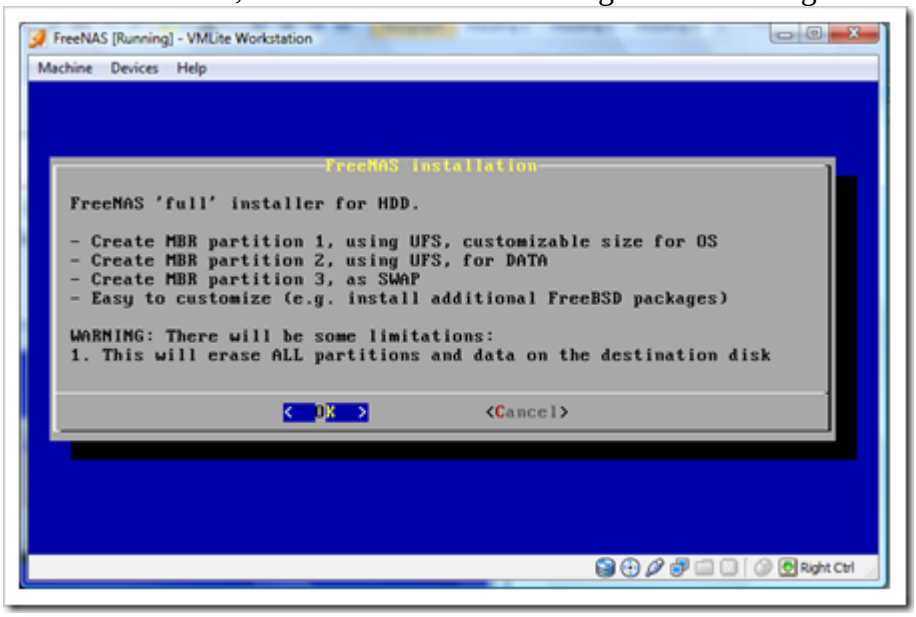


As we can see, from the above screenshot, there are 9 available options, while we booted from the FreeNAS Live CD. We need to choose the option 9, to start the installation, and load the OS

to the hard disk. You will get the following screen after you press 9 on the previous menu.

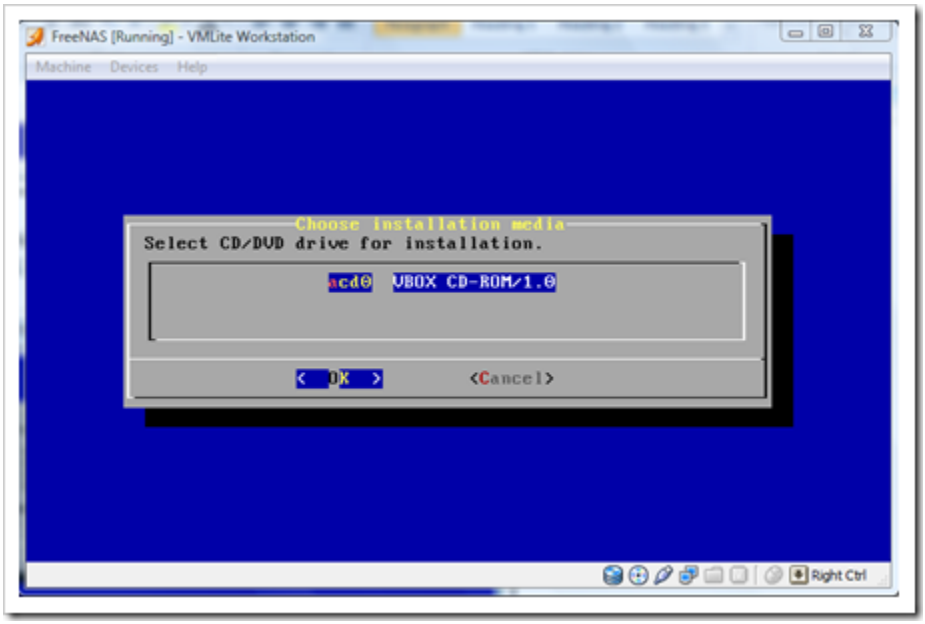


Select the third option to install the Full OS, and click on OK. You will get the following screen.

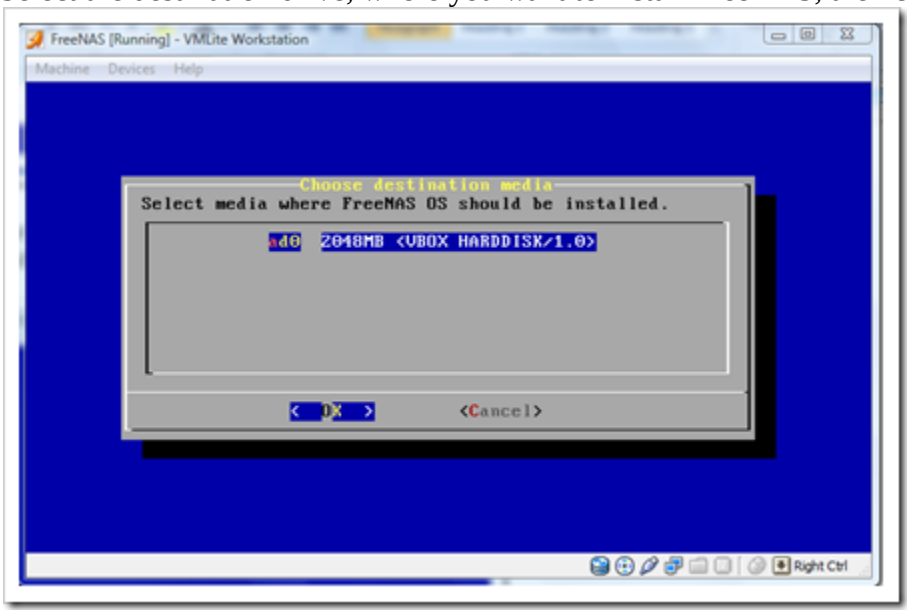


Click on OK to continue.

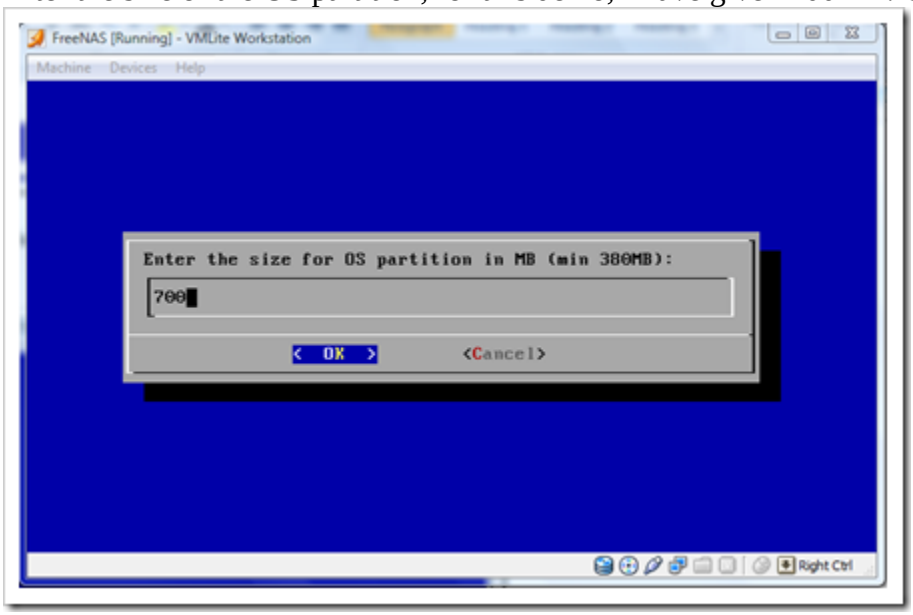
Select your Current installation media, for this installation, we are using a CD-ROM to mount the FreeNAS OS. Click on OK to continue.



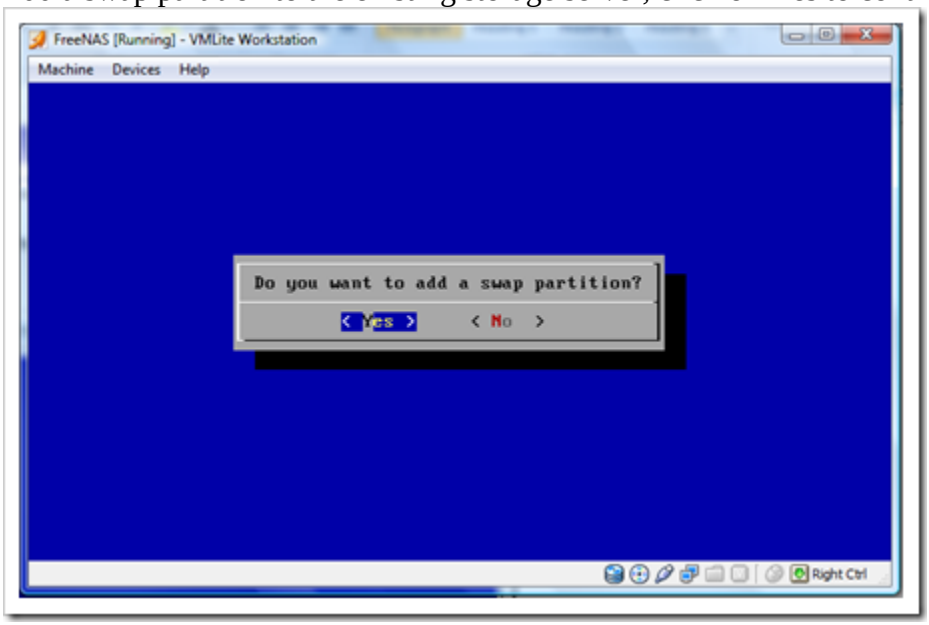
Select the destination drive, where you want to install FreeNAS, then click on Ok to continue.



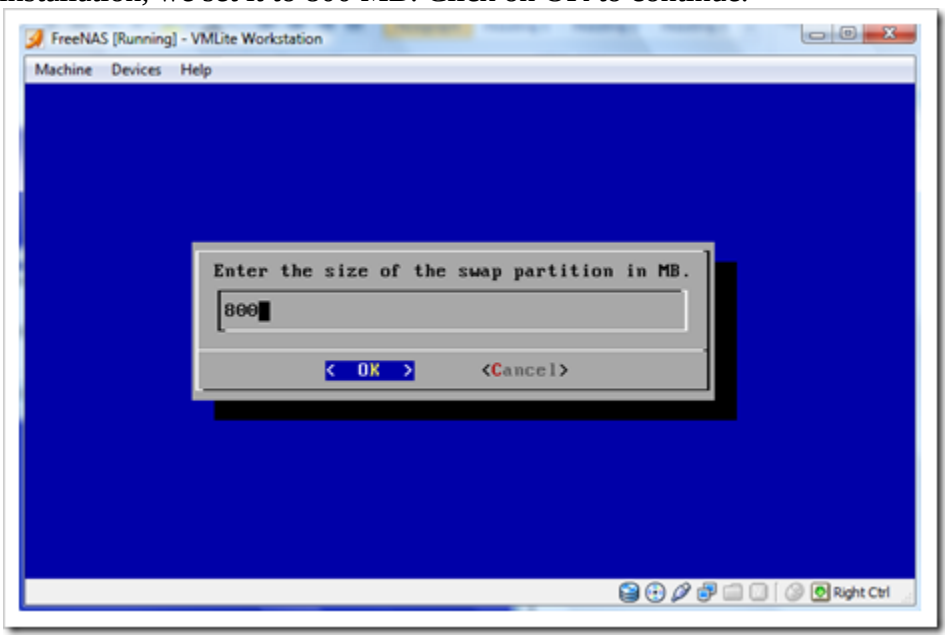
Enter the size of the OS partition, for this demo, I have given 700MB. Click on Ok to continue.



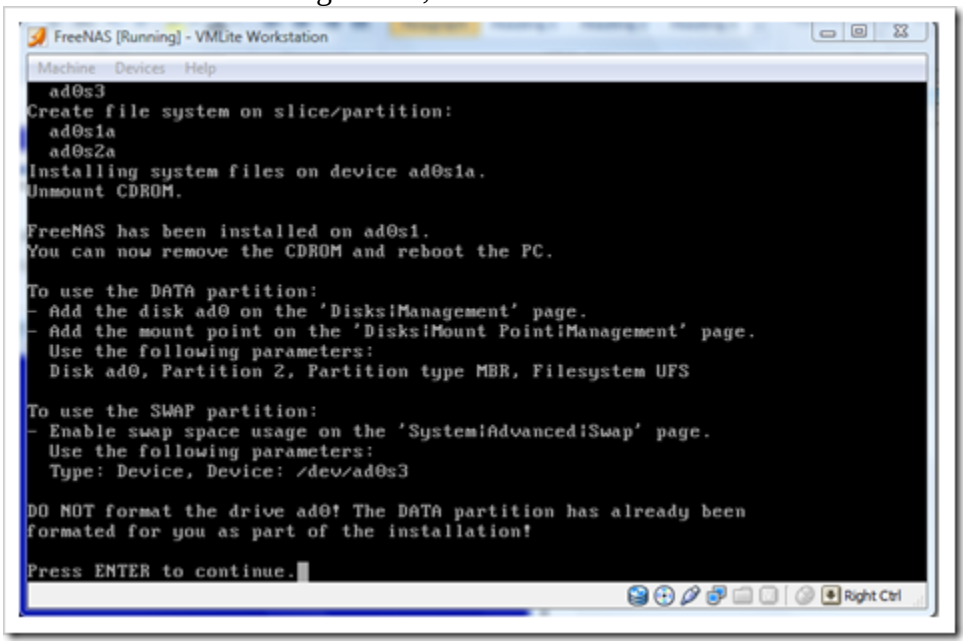
Add a swap partition to the existing storage server, click on Yes to continue.



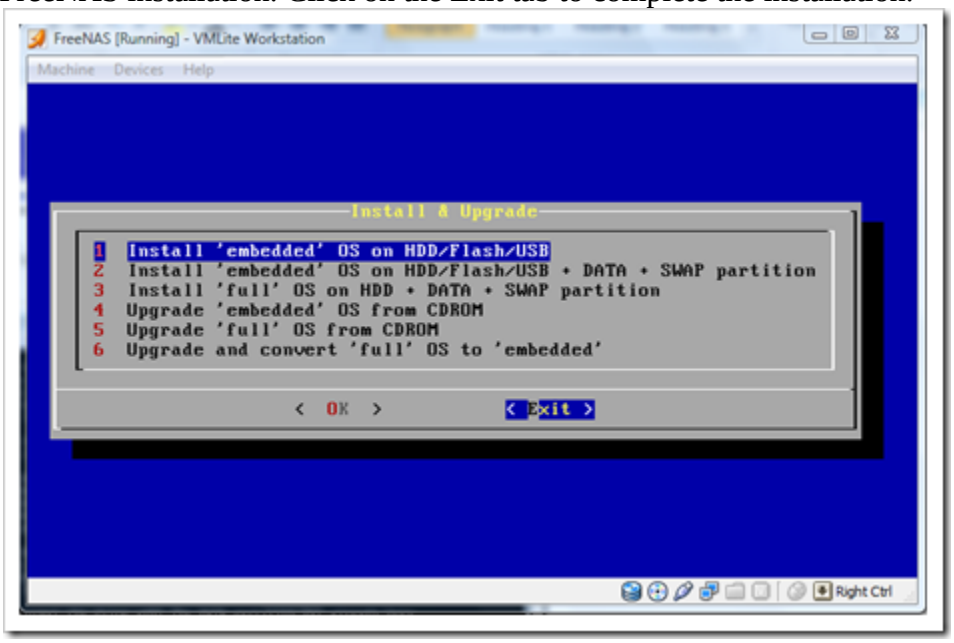
The size of swap is should be double the size of the physical RAM present in the server, for our installation, we set it to 800 MB. Click on OK to continue.



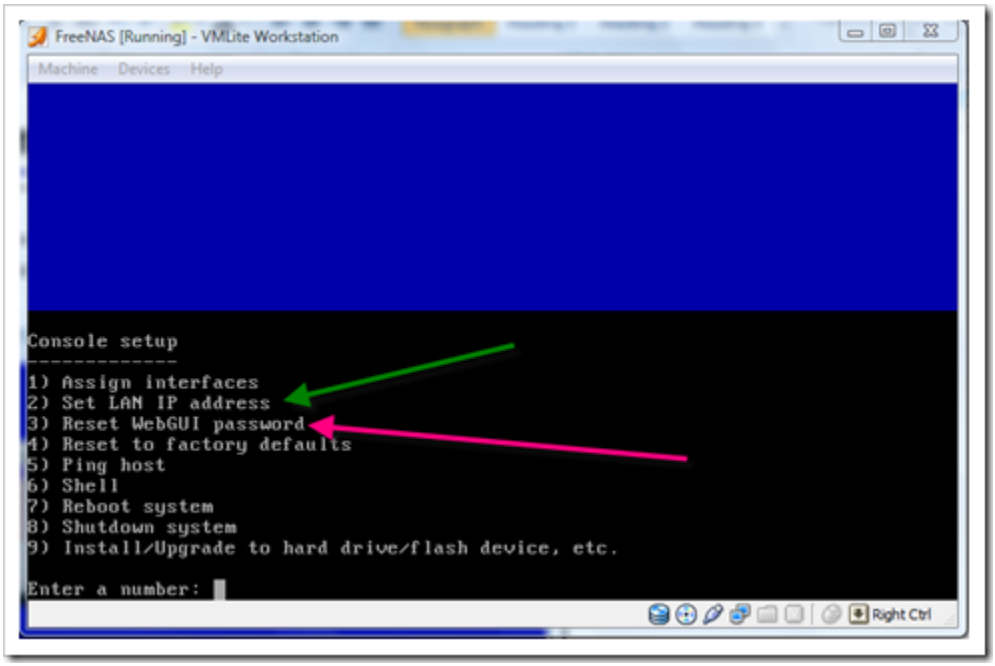
Once we set all the configuration, the installation of FreeNAS will continue.



Click on Enter to complete the installation, and you will be back to the starting screen of FreeNAS installation. Click on the Exit tab to complete the installation.

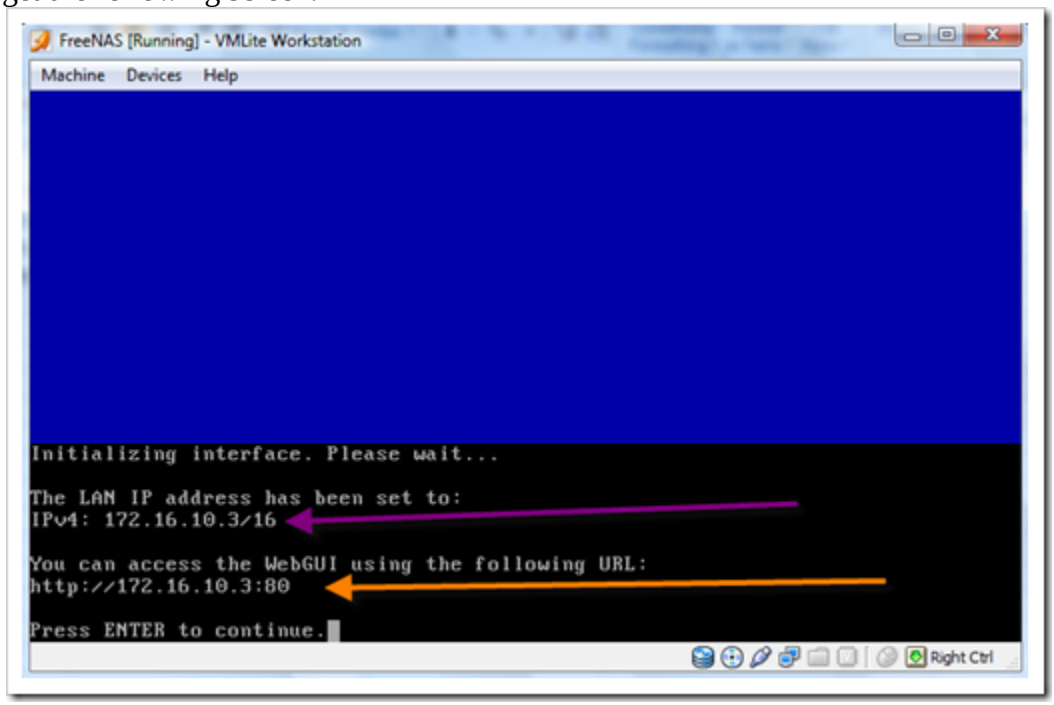


Since the installation of FreeNAS is completed, we need to set the IP Address for this server, and set the WebUI to access the server.

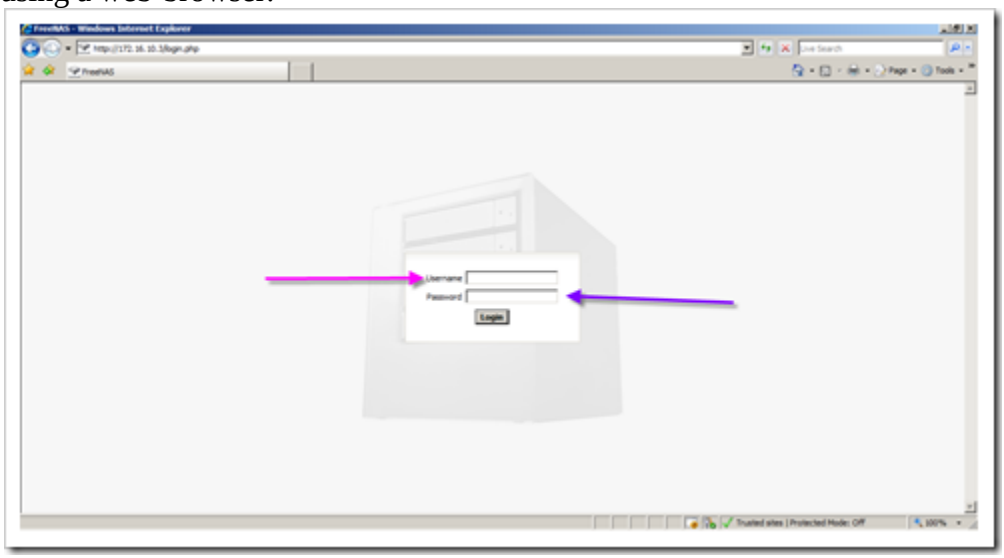


Select 2 to set an IP Address for the server. For our environment, we are going to configure the Static IP Address, Subnet Mask, Default Gateway, and DNS server for our storage server. Click on Next and provide all the information. Once you have configured all the information, you will

get the following screen.



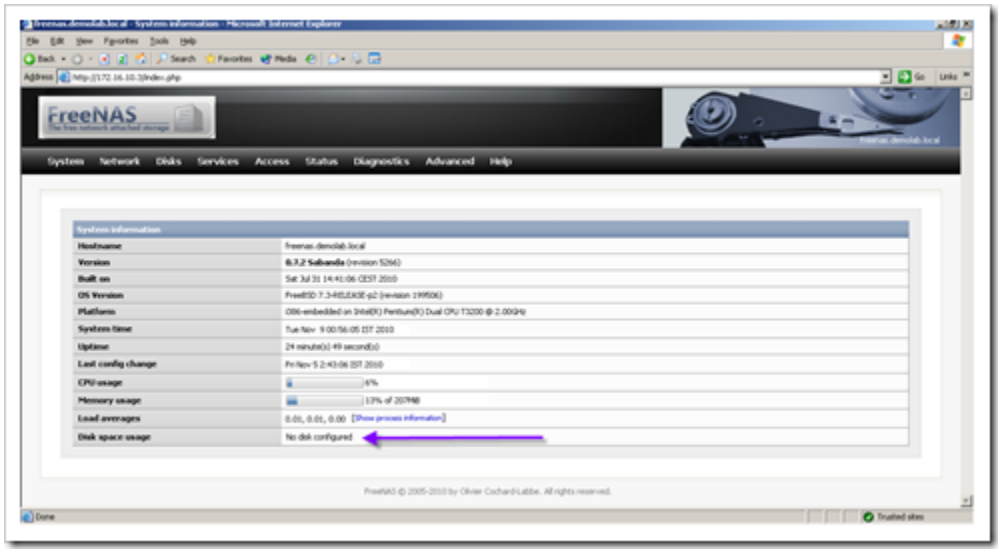
Thus, the initial configuration of our storage server is completed. Remove the iso image from the machine and restart the server. Once the server is restarted, you can access the storage server using a web-browser.



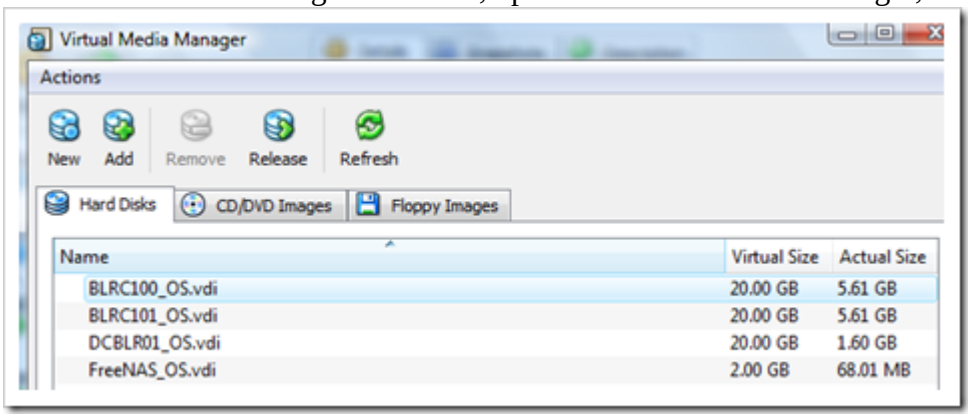
When you connect through the above URL, you will get the above screen. The initial username and password are **admin** & **freenas** respectively. In the next part of this series, we will see how to configure the storage server for the cluster deployment.

Building a SQL Server 2008 R2 Cluster: Part II

In the first part of this series, we saw the installation of our storage server for our demo cluster environment. In this part, we are going to configure our storage server. Before we configure our storage server, we need to add hard disks to our server. As you can see from the below screenshot, there are no disks available to our storage server.

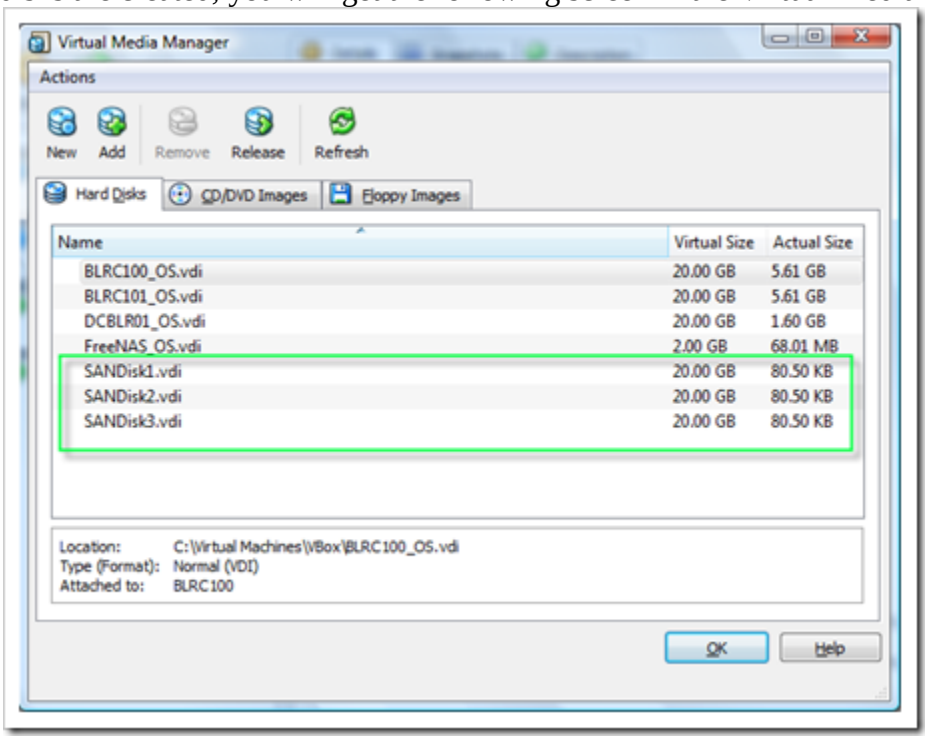


To create a new hard disk using Oracle VM, open the Virtual Media Manager, from the File menu.

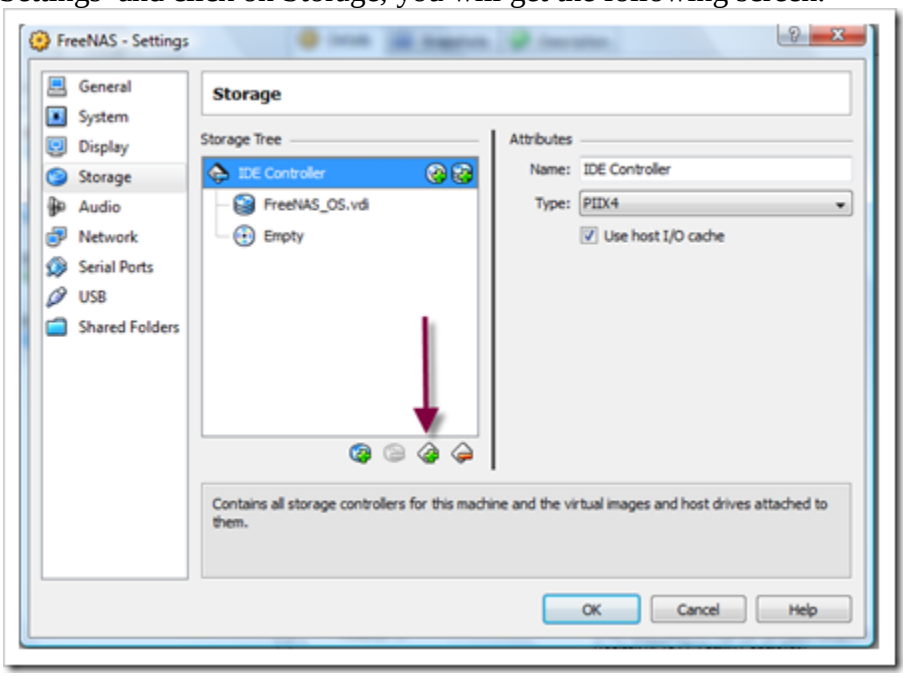


Click on New to start the create a new virtual disk wizard and follow the on-screen instruction to create a new virtual hard disk. Follow the above twice to create additional two disks. Once all the

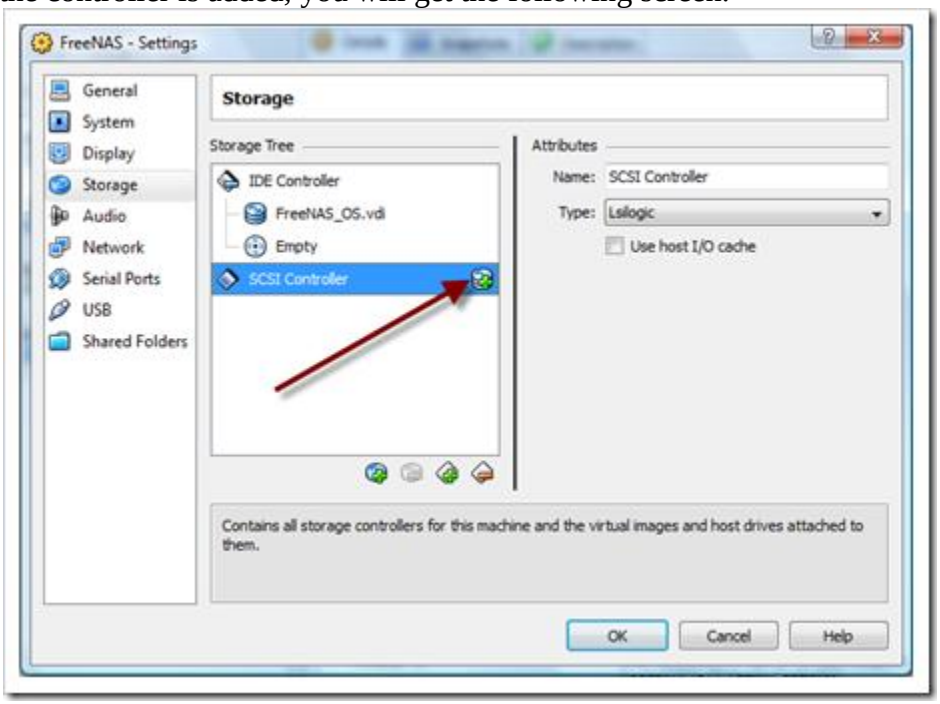
disks are created, you will get the following screen in the Virtual Media Manager.



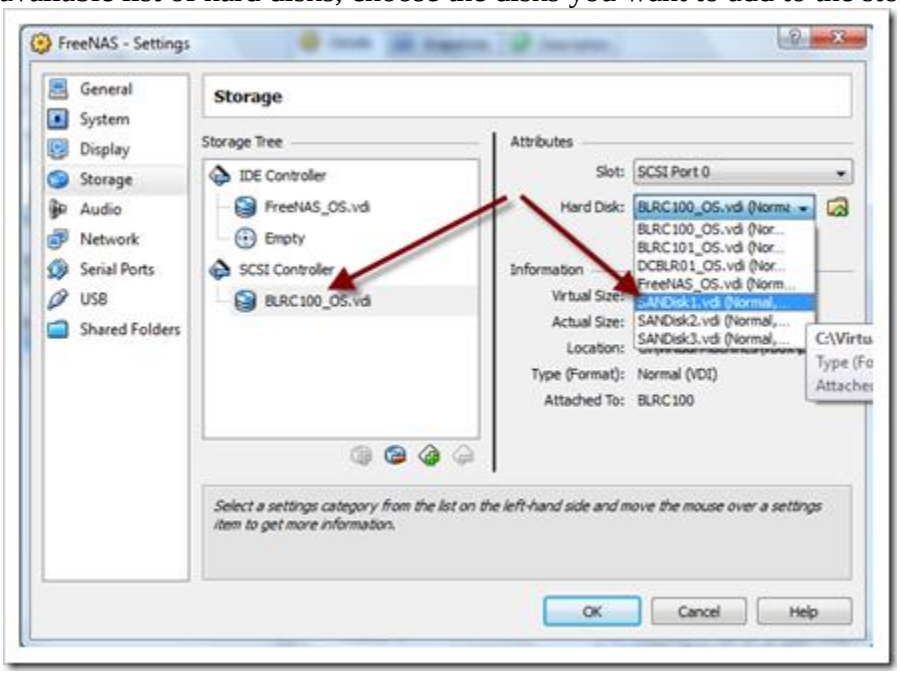
Since all the required disks for our storage server is available, now we are going to add the disks to the server. To add the disks, we need to shut down the storage server. Open the Storage server Settings and click on Storage, you will get the following screen.



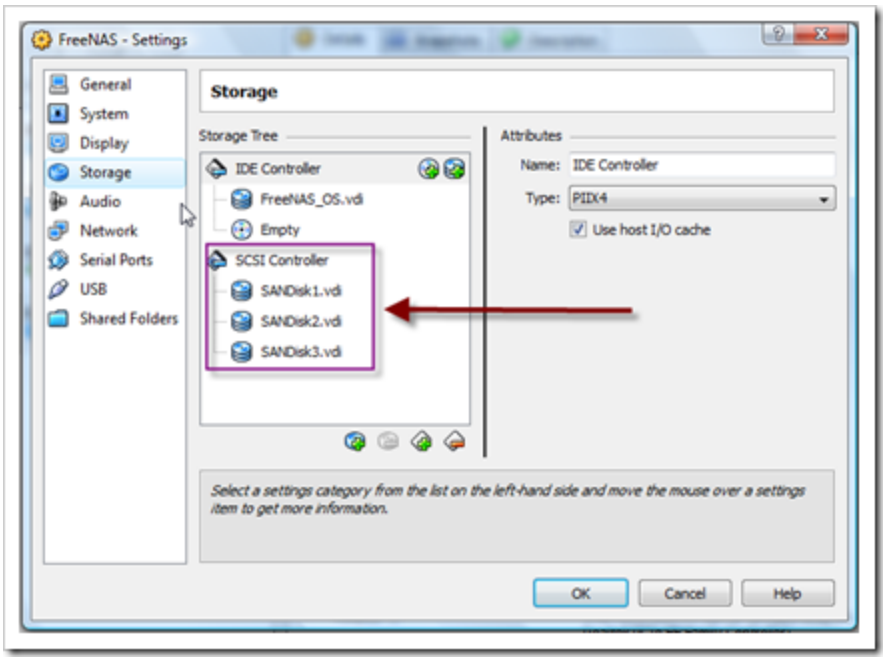
Click on the Add Attachments, and select SCSI controllers to add it to the storage server. Once the controller is added, you will get the following screen.



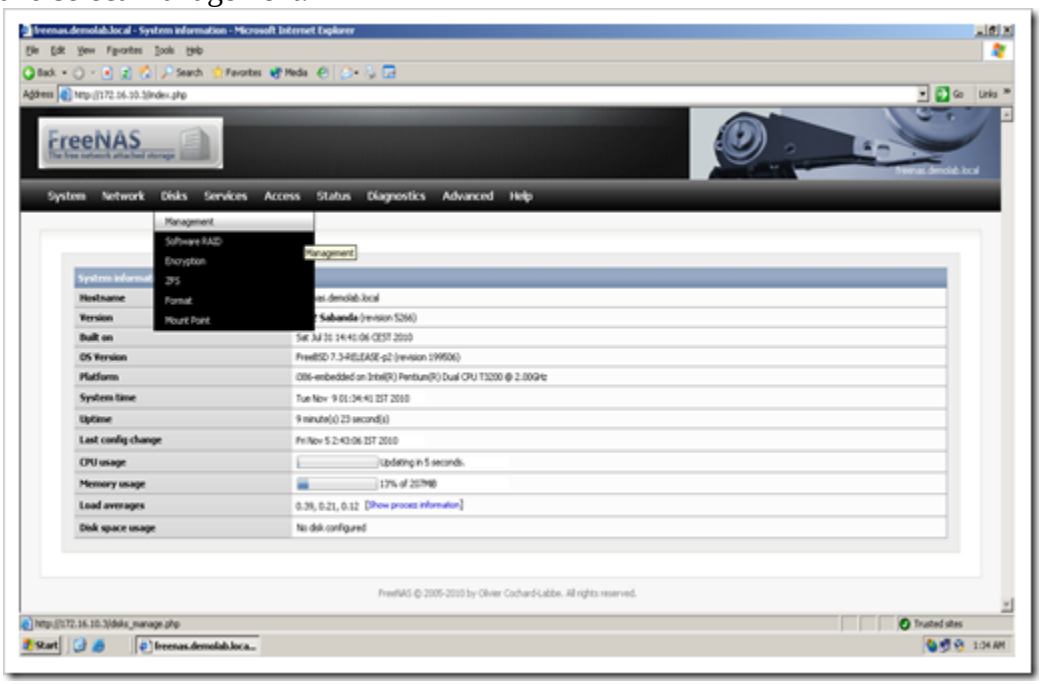
Right click on the SCSI Controller, and select Add Hard disk from the available menu. From the available list of hard disks, choose the disks you want to add to the storage server.



Once all the three disks are added, you will get the following screen, click on Ok to close the Setting window for the storage server.

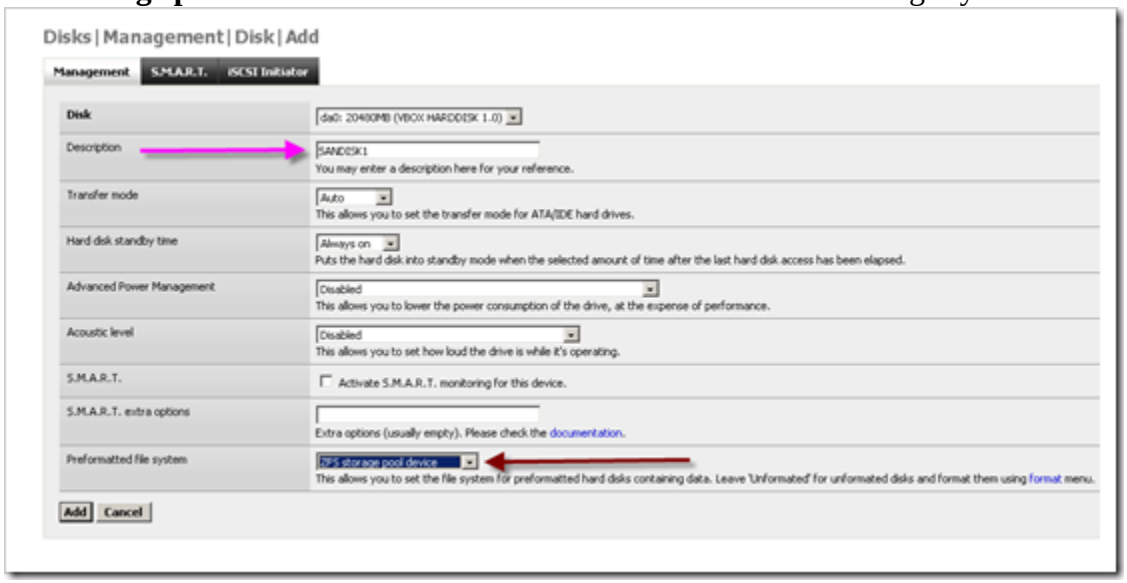


Now that we have added the disks for our storage server, its time to configure the same for our demo cluster. Start the storage server and wait until its up and running. Once the server is up and running, we can access it using the administration website, look at Part I of this series to find out the URL of this site. Our storage server can be accessed using the following URL: http://172.16.10.3:80 . Once we logged on to the website, Click on Disk from the available menu and select Management.

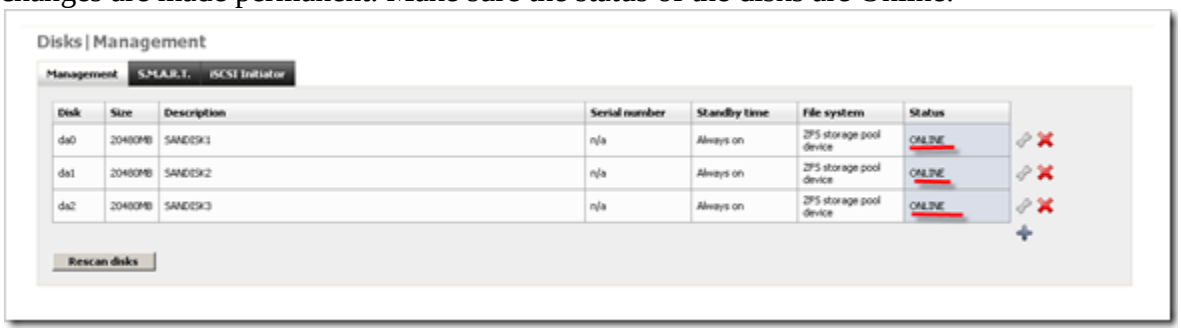


Once you select Management (Disks –>Management), you will get the following screen. Click to add the disks to the server.

Once you click on the icon as shown above, you will be presented with the below screen. Enter a description for the disk you have added to server and selected the preformatted file system as **ZFS storage pool device**. Click on Add to initialized the disk the storage system.



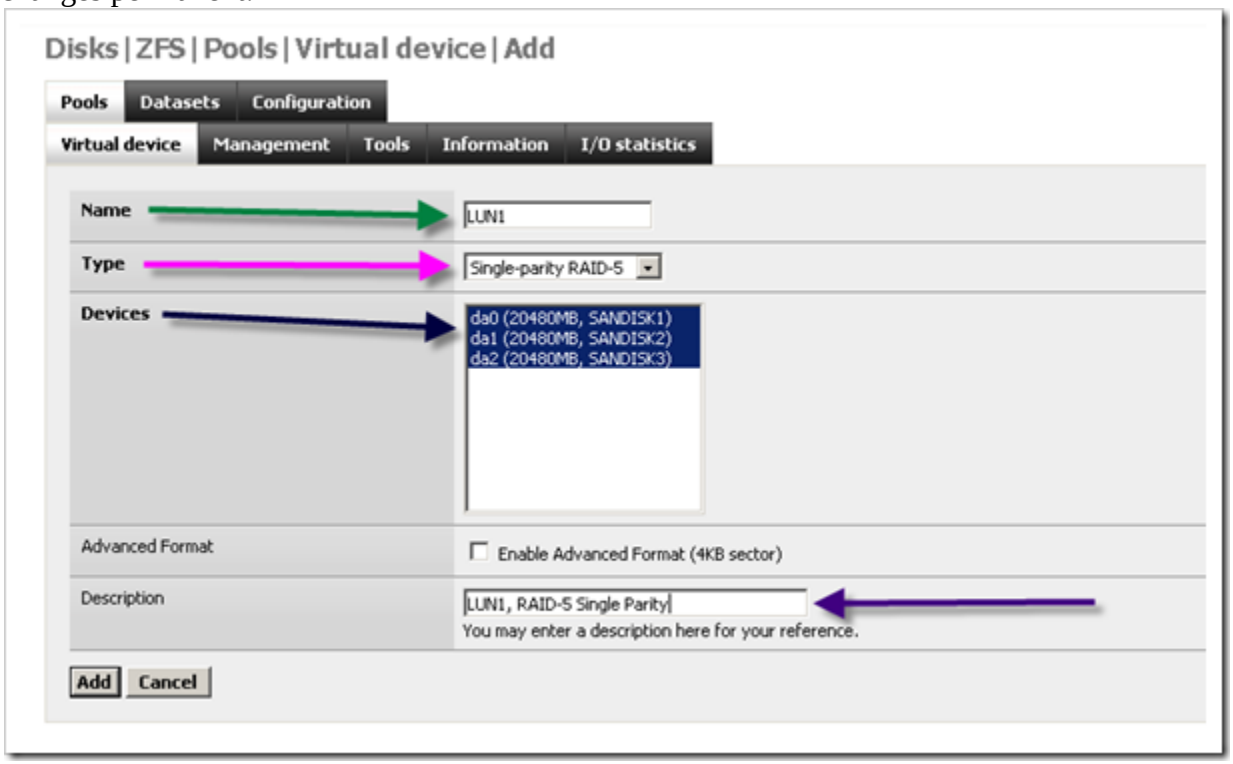
Repeat the same steps twice to add another two disks. Once all the disks are added, click on *Apply Changes* to make these changes permanent. You will get the following screen once the changes are made permanent. Make sure the status of the disks are Online.



Since all the disks are added to the server, now we are going to create a virtual RAID-5 volume for our cluster. Click on Virtual Devices (Disks –> ZFS –>Virtual Devices) from the menu, you

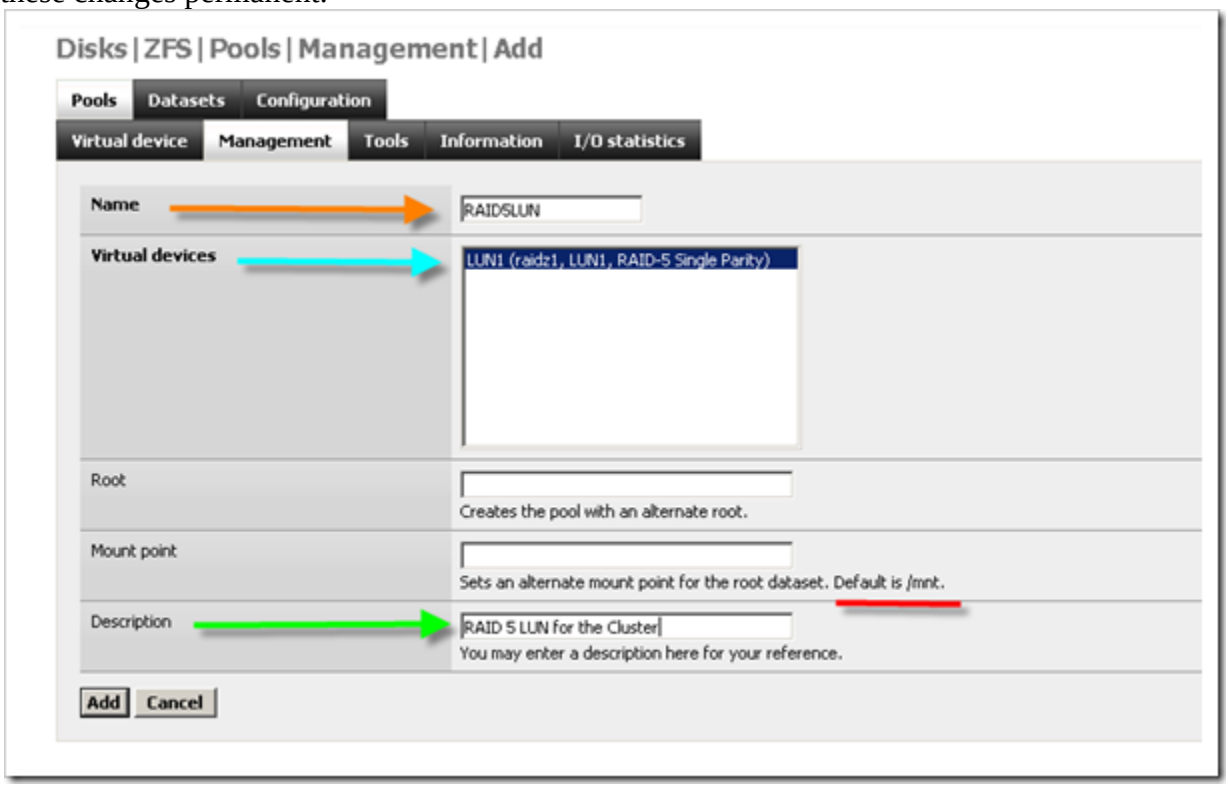
will get the following screen.

Click on the + icon to add a new virtual device. As shown in the below screenshot, we are going to create a RAID-5 Disk volume. Give a name to the virtual device and provide a description for the same and click on *Add*. Once the virtual disk is added, click on *Apply Changes* to make these changes permanent.

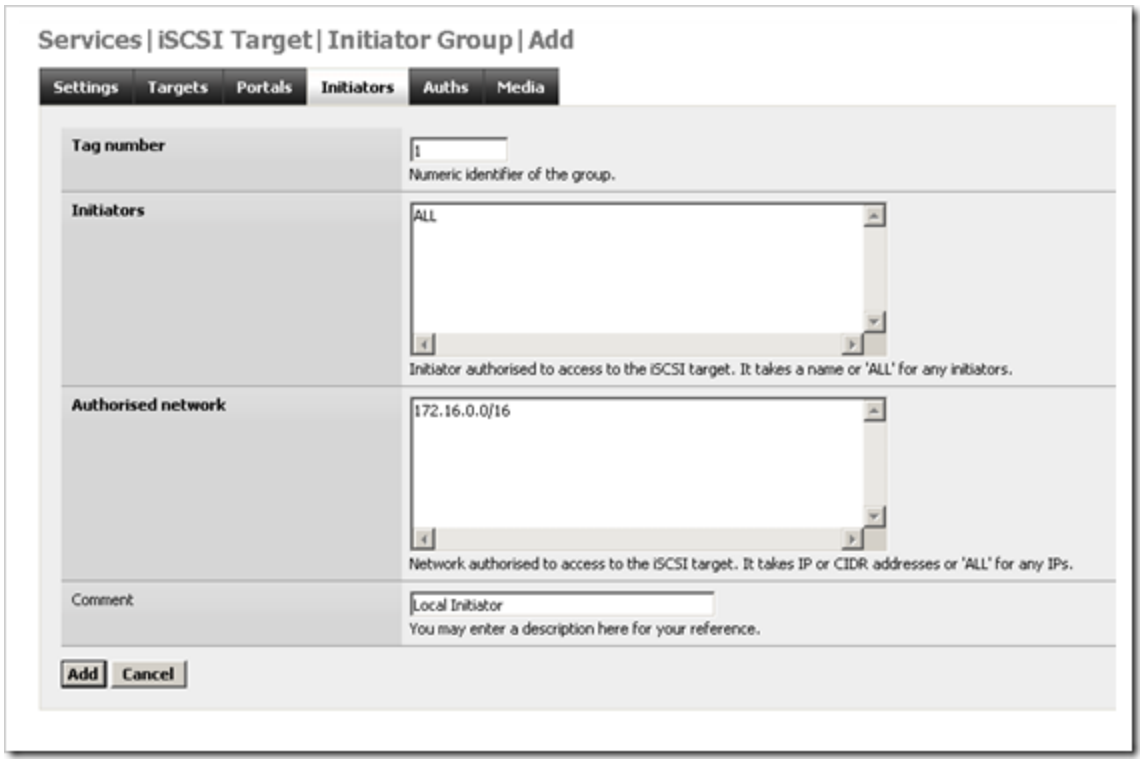


Once the virtual device is created, we need to add it to ZFS Management (Disks –>ZFS –>Management). Select the available virtual device and add a name & description for the same. The default mount point for the disk will be /mnt. Click on *Add* and *Apply Changes* to make

these changes permanent.

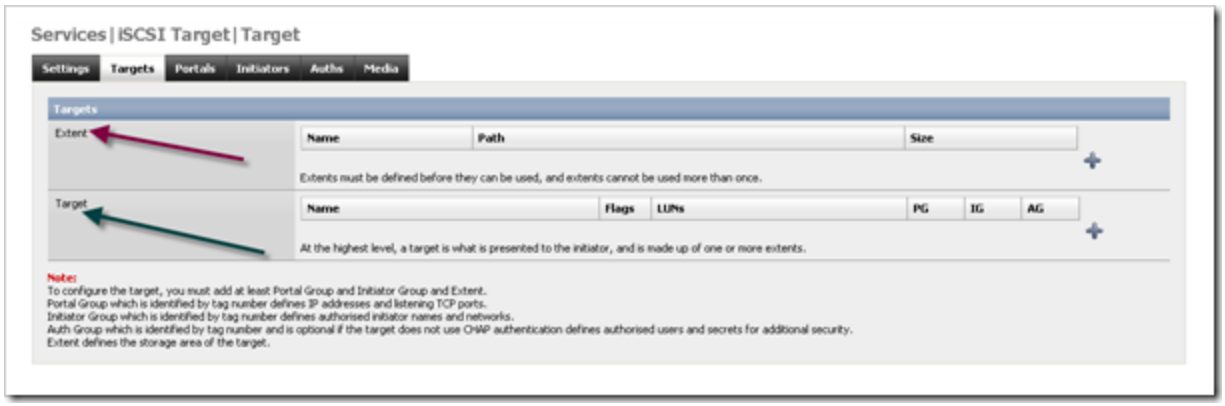


Now the RAID-5 volume is configured properly, we need to create individual disks for our cluster system. For the Windows 2008 cluster system, we are going to use the **iSCSI** service to share the disk among the cluster. To begin with, click on iSCSI Service (Service –>iSCSI Service). Click on the initiator tab and add the Network details for the local initiator. Click on *Add* and *Apply Changes* to continue.



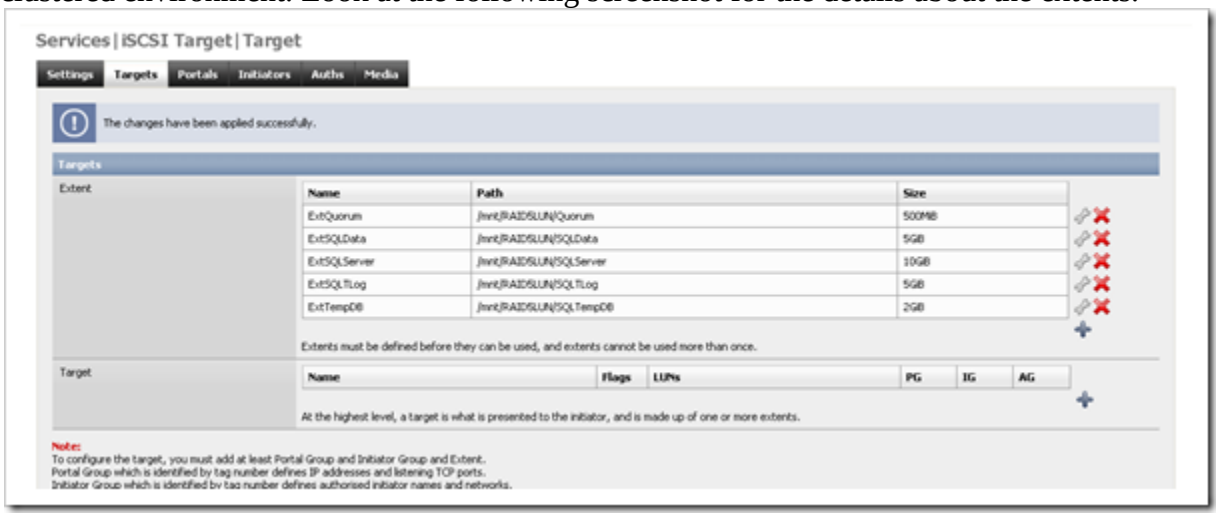
Click on Portals to configure the IP address and Port number for the storage server portal, the same address will be used by the target to access the disks. Click on *Add* and *Apply Changes* to continue.

Since we have created both Initiators and Portals for the iSCSI service, now we are going to create the target volumes for our cluster. To start with click on the Targets tab (Services –>iSCSI Service –>Targets), you will get the following screen. Here we are going to configure both extent and disk volumes.



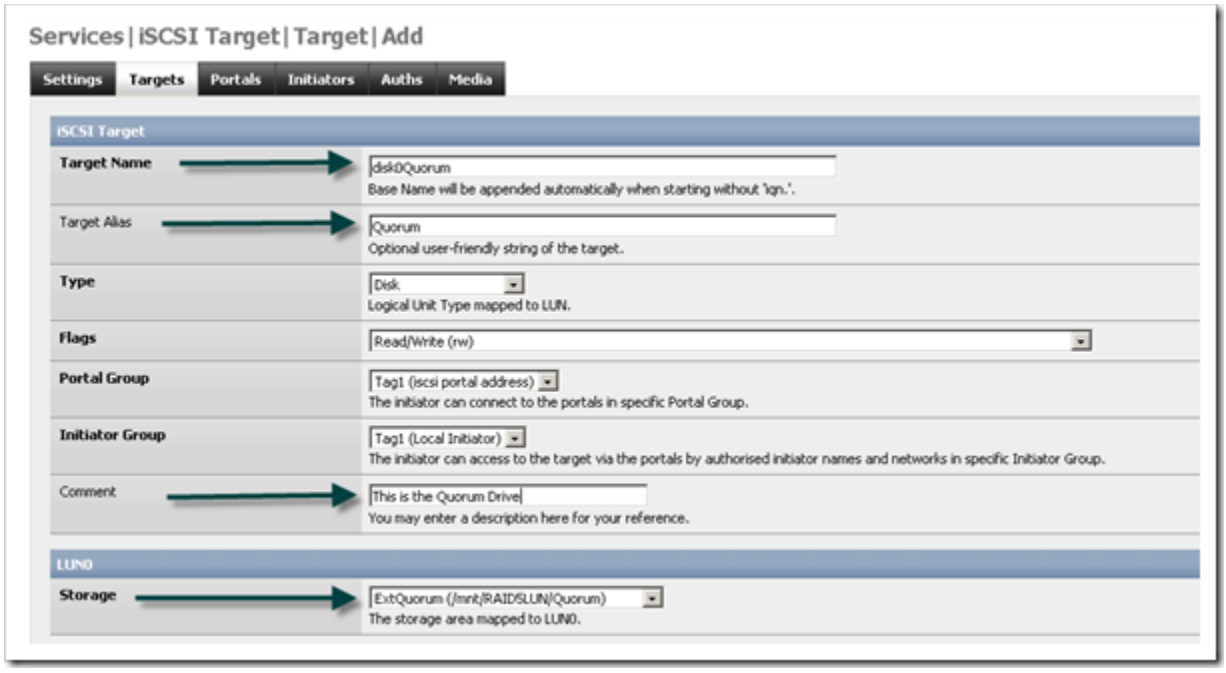
Click on the + icon in the extent to create a new extent. I have created an extent for the quorum of the cluster. Choose the type as *File*, File size as *500 MiB*, and mount point path as /mnt/RAID5LUN/Quorum. Click on Add and Apply changes to continue.

I have added another 4 extents, which will be used during the installation of SQL Server in the clustered environment. Look at the following screenshot for the details about the extents.

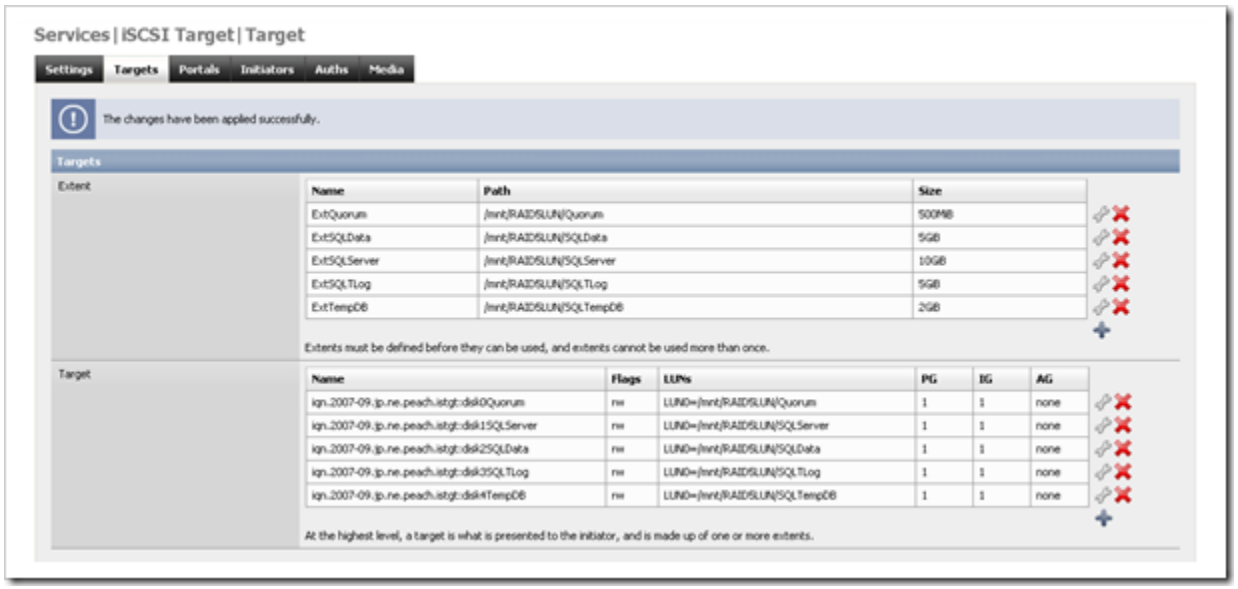


The extents are created, now we are going to create the target disks for the cluster. Click on the *Add Target* icon to create a target disk. Enter the Target Disk Name, Alias name, Extent details

and description for the disk as shown below. Click on Add and Apply changes to create the target disks.



Repeat the above steps to create the other target disks. Once all the disks are created, you will get the following screen.



So the storage server is configured and all the disks are available through iSCSI service to the cluster nodes. In the next part of this series, we are going to start the Windows Cluster installation.

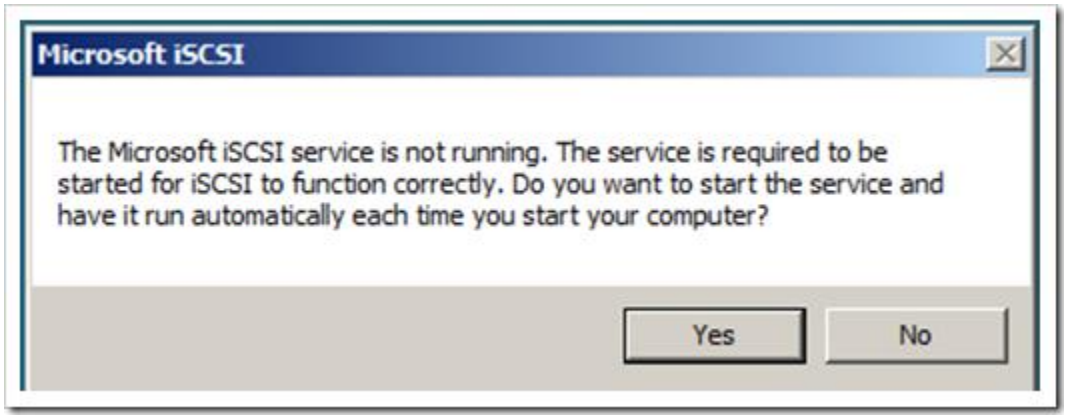
Building a SQL Server 2008 R2 Cluster: Part III

In the previous part of this post, we have configured the disks for our Windows Cluster. The disks are created from a RAID 5 LUN. As you can see, we have a LUN of 60GB however only 40GB is available for shared storage.

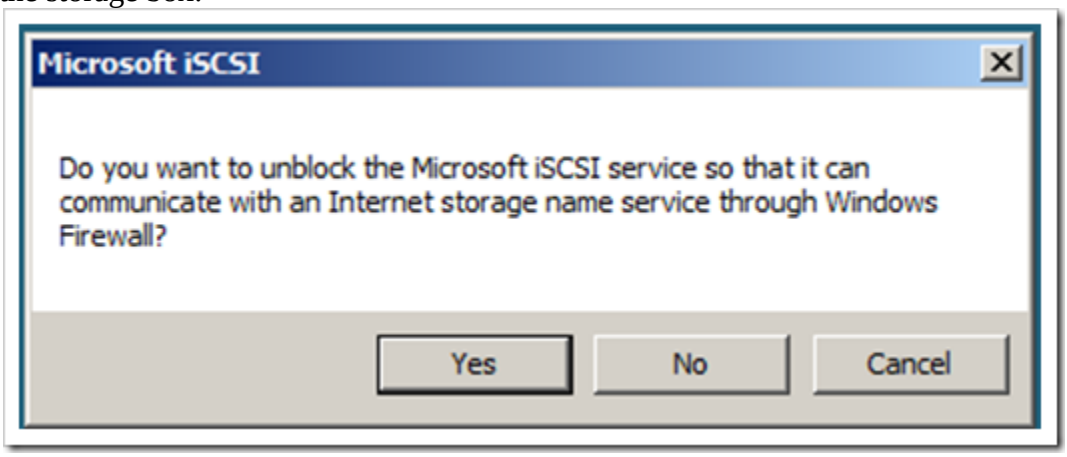
As we are going to install SQL Server on a windows cluster environment, we are going to add a new disk for the MSDTC resource. Here is a list of all the available disks available to our cluster.

Name	Flags	LUNs	PG	IG	AG
iqn.2007-09.jp.ne.peach.istgt:disk0Quorum	rw	LUN0=/mnt/RAID5LUN/Quorum	1	1	none
iqn.2007-09.jp.ne.peach.istgt:disk1SQLServer	rw	LUN0=/mnt/RAID5LUN/SQLServer	1	1	none
iqn.2007-09.jp.ne.peach.istgt:disk2SQLData	rw	LUN0=/mnt/RAID5LUN/SQLData	1	1	none
iqn.2007-09.jp.ne.peach.istgt:disk3SQLTLog	rw	LUN0=/mnt/RAID5LUN/SQLTLog	1	1	none
iqn.2007-09.jp.ne.peach.istgt:disk4TempDB	rw	LUN0=/mnt/RAID5LUN/SQLTempDB	1	1	none
iqn.2007-09.jp.ne.peach.istgt:disk5MSDTC	rw	LUN0=/mnt/RAID5LUN/MSDTC	1	1	none

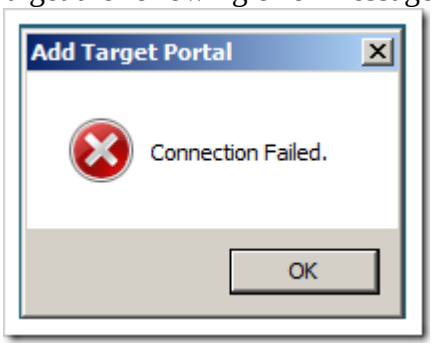
Now, we have all our disks ready, its time to add it to both our nodes. To begin with, I am going to log on to our first node, BLRC100. Both the nodes will access these shared disk with the iSCSI service available with Windows Server 2008. Open the iSCSI Initiator from the administrative tools (*Start Menu –> Administrative Tools –> iSCSI Initiator*). If the iSCSI service is not running on this server, you will get the following screen, click on Yes to start the Microsoft iSCSI service on this server.



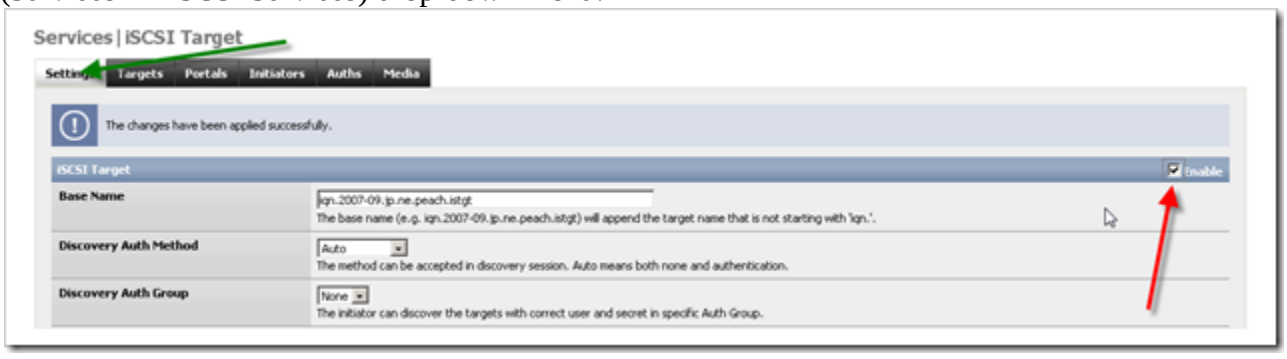
If prompted, unblock the iSCSI service in the Windows Firewall so that it can communicate with the storage box.



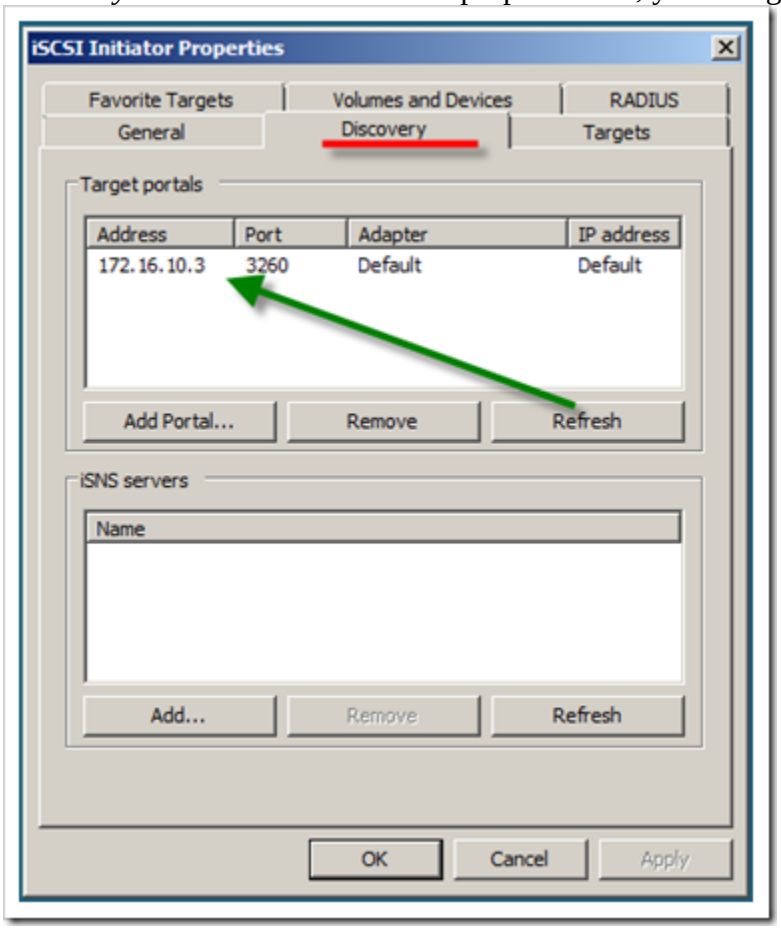
The *iSCSI Initiator* properties window will be opened. Click on the *Discovery* tab. In this tab, we are going to add the storage server portal. Click on *Add Portal*, and provide the IP address or FQDN of the storage portal and click on *OK*. If you get the following error message, make sure the iSCSI service is running on the storage server.



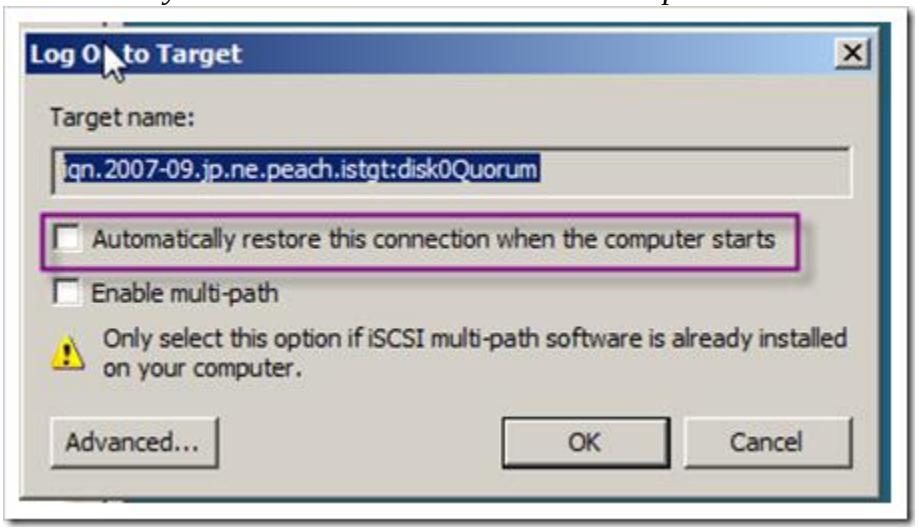
To start the iscsi service on the storage server, click on the *Setting* tab from the *Services* (*Services –> iSCSI Services*) drop-down menu.



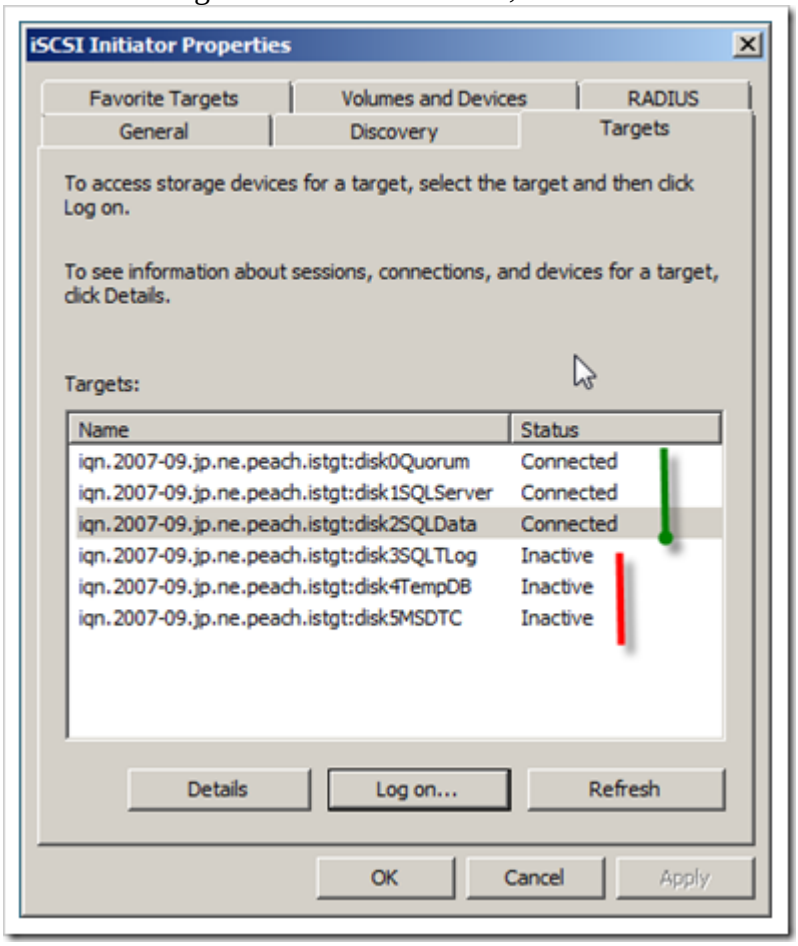
If the iSCSI service is running on the storage server, after you will add the portal on the *Discovery* tab of the iSCSI Initiator properties tab, you will get the following screen.



Now that we have added the target portal, click on the *Targets* tab of the iSCSI initiator properties. You will find all the disks in inactive state. Select one disk and click on the *Log On* button, you will get the following screen. Make sure, you select the checkbox next to *Automatically restore this connection when the computer starts*. Click on OK to continue.



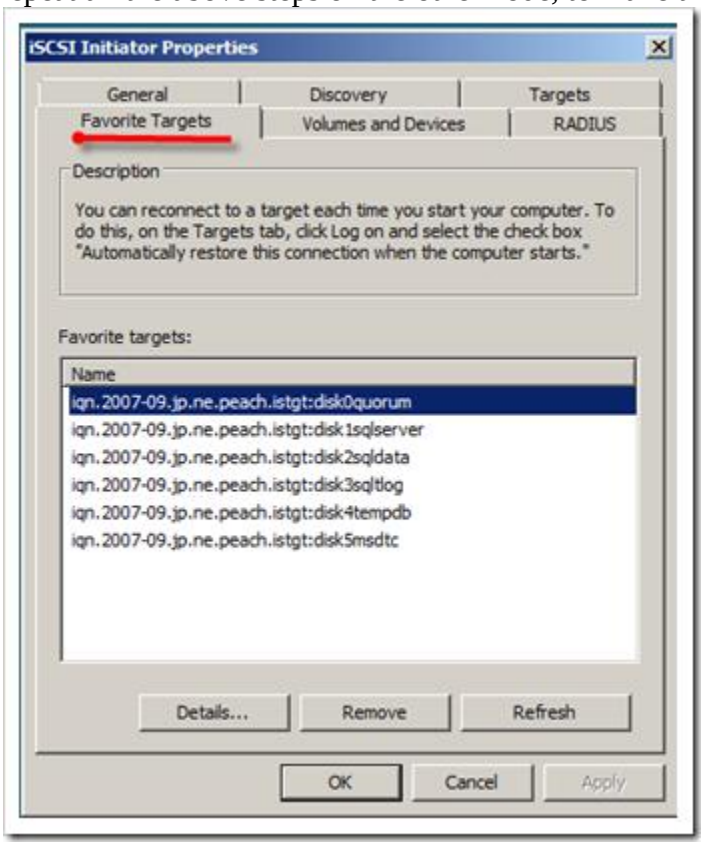
Once you have added some of these disks, you will get the following screen. In the below screenshot, the disks next to the green-line are connected; however the disks next to the red-line



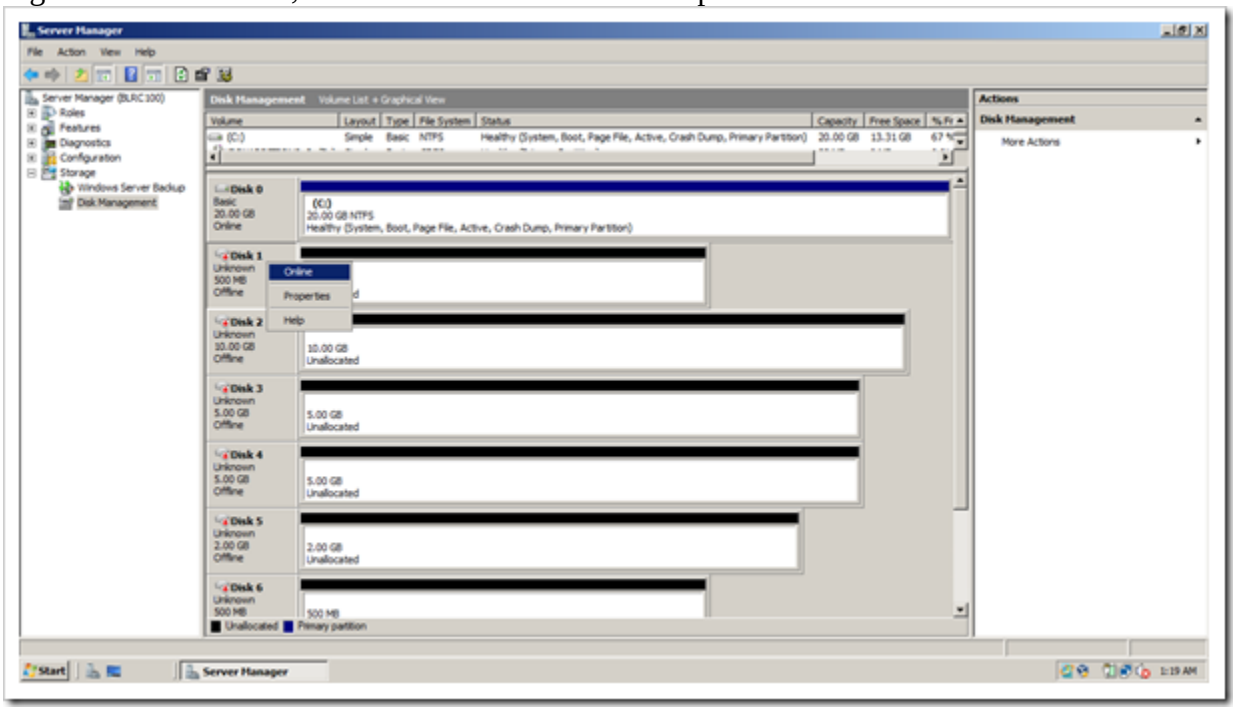
are still inactive.

Once all the disks are active, click on the *Favorite Targets* tab to list all the shared disk available to the Cluster node. Click on *Ok* to close the *iSCSI Initiator* properties window. We need to

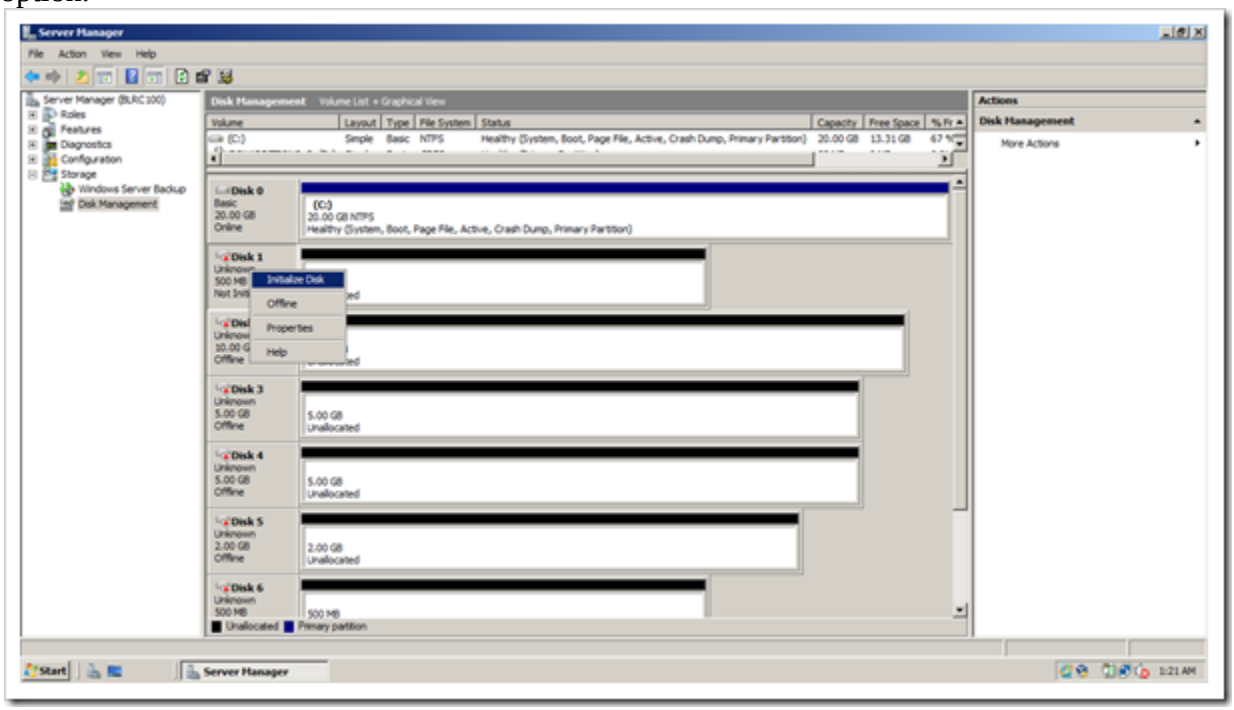
repeat all the above steps on the other node, to make these disks available.



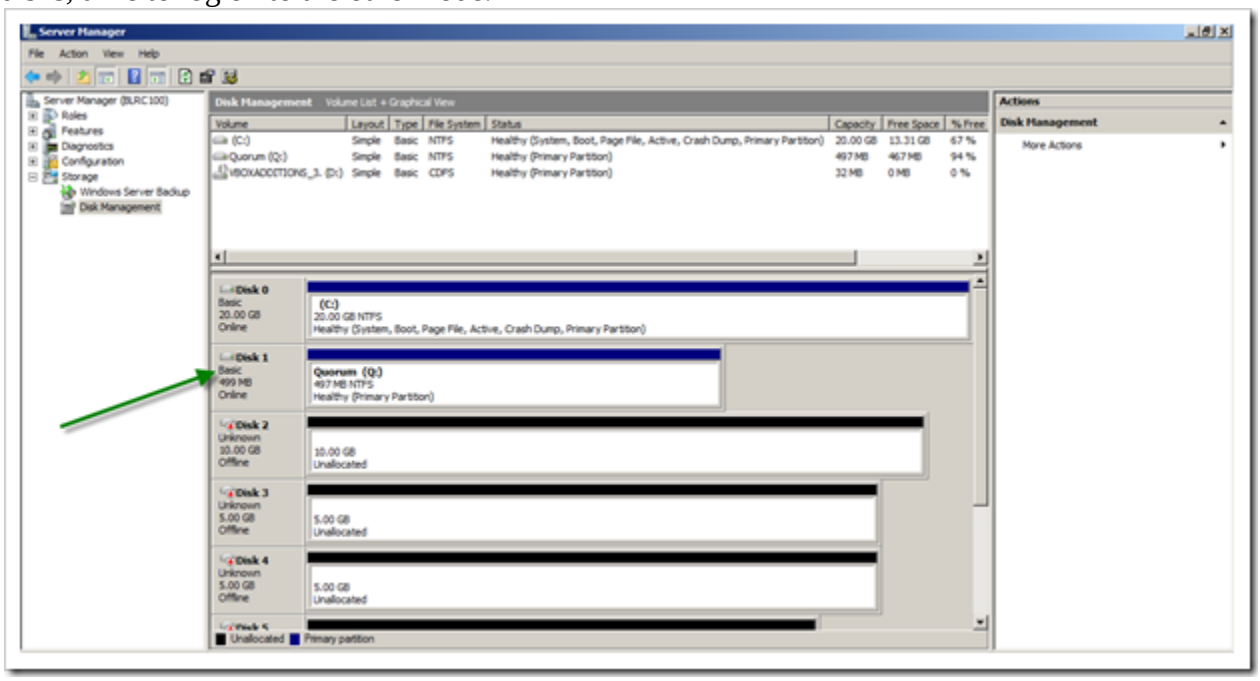
Once you open the *Server Manager* (*Start Menu –> Administrative Tools –> Server Manager*) and browse to *Storage –> Disk Management*. You will find all the disks are available offline. Right-click on the disk, and select *Online* from the drop-down menu.



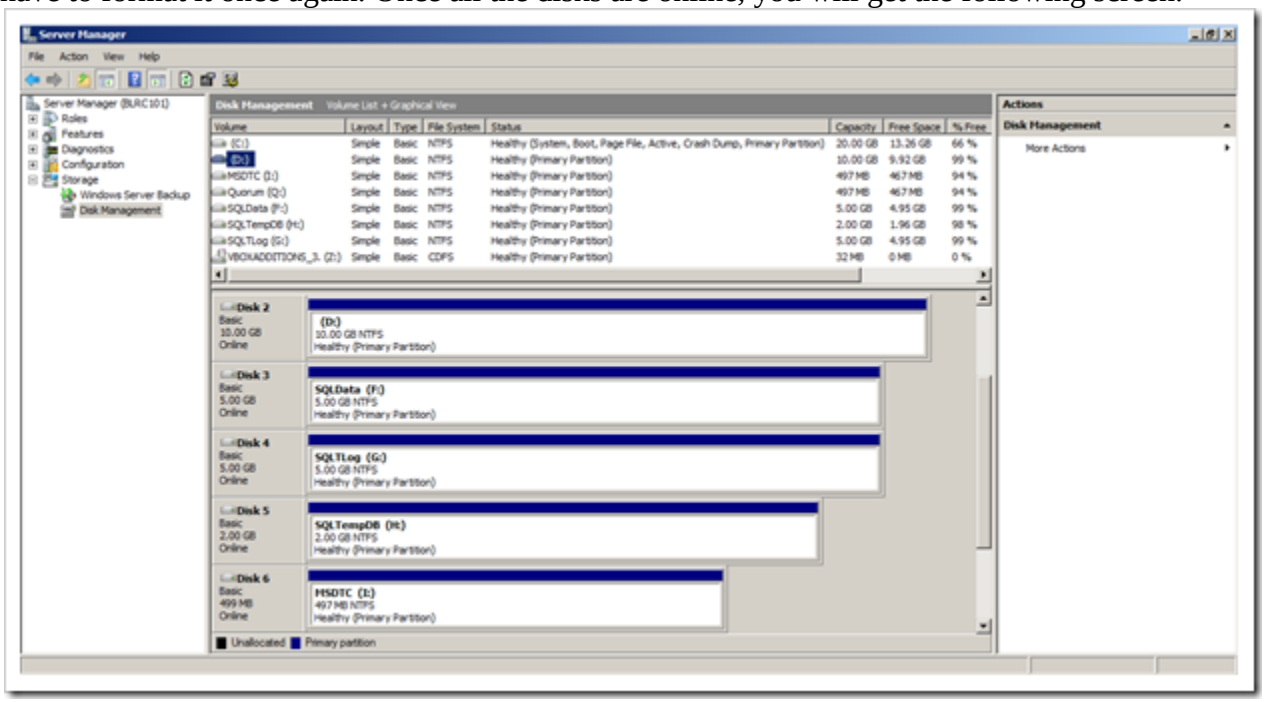
Once the disk is online, right-click on the disk and select *Initialize Disk* from the available option.



After you initialized the disk, format the disk. Make sure you have selected NTFS file system for the disk. Once the disk is formatted, bring it offline. You need to initialized all the available disks, format it with NTFS partition and then bring it offline. After you have completed all the disks, time to log on to the other node.



As we have completed all the above steps on the first node of our windows cluster, time to move to the second node of our cluster system, BLRC101. Shut down the cluster node BLRC100 & start the node BLRC101. On this node, we have to all the iSCSI service as we have done for node BLRC100, bring the disks online. As we have already formatted the disks earlier, we don't have to format it once again. Once all the disks are online, you will get the following screen.

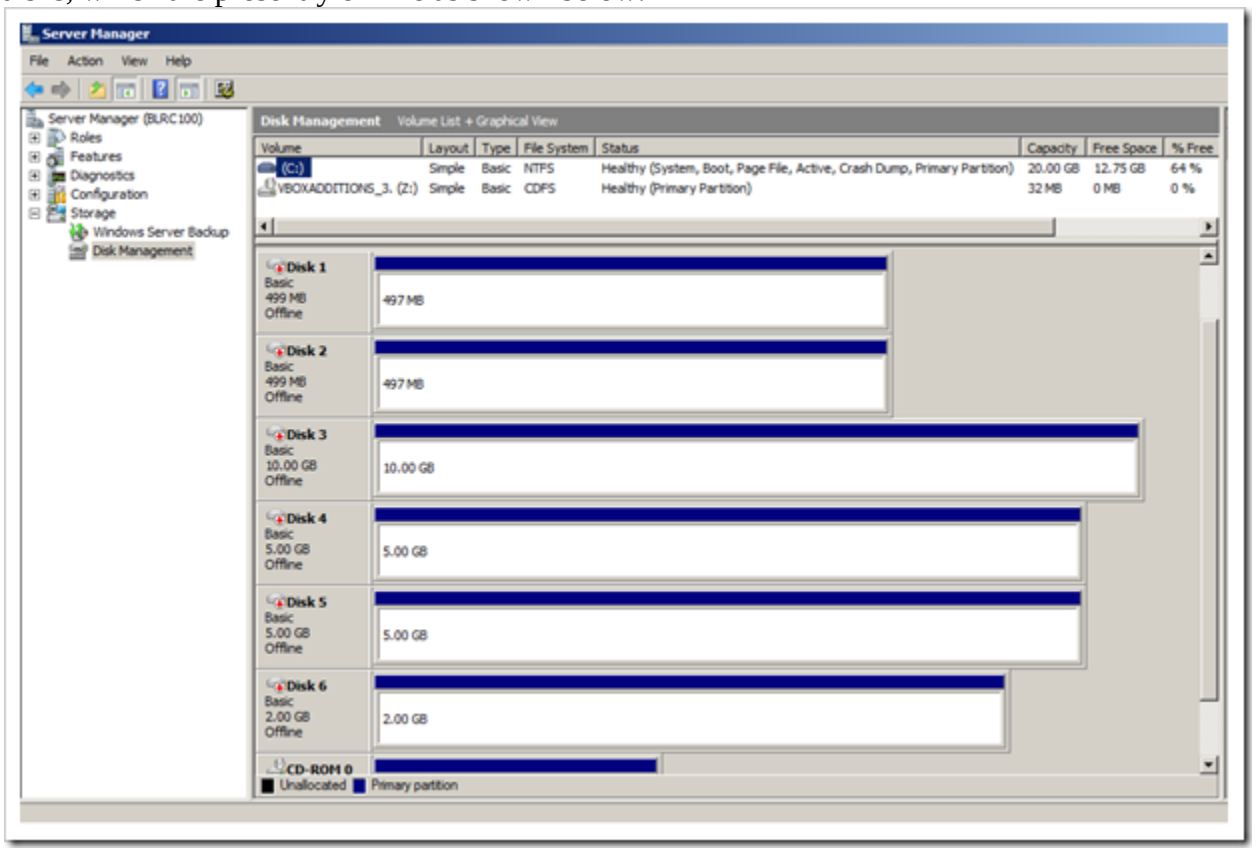


Since all the disks are ready for cluster installation, we are going to take these disks offline and shutdown this node. Then we are going to add a network adapter on each node, which will create a HeartBeat Network (10.200.41.100/8, 10.200.41.101/8) for the cluster. Once we add the network adapter, we are going to install ***Dot Net Framework v 2.0 & 3.5*** on both the nodes. Since DotNet framework is mandatory for SQL Server cluster installation, we are going to install it before starting the Windows Cluster installation.

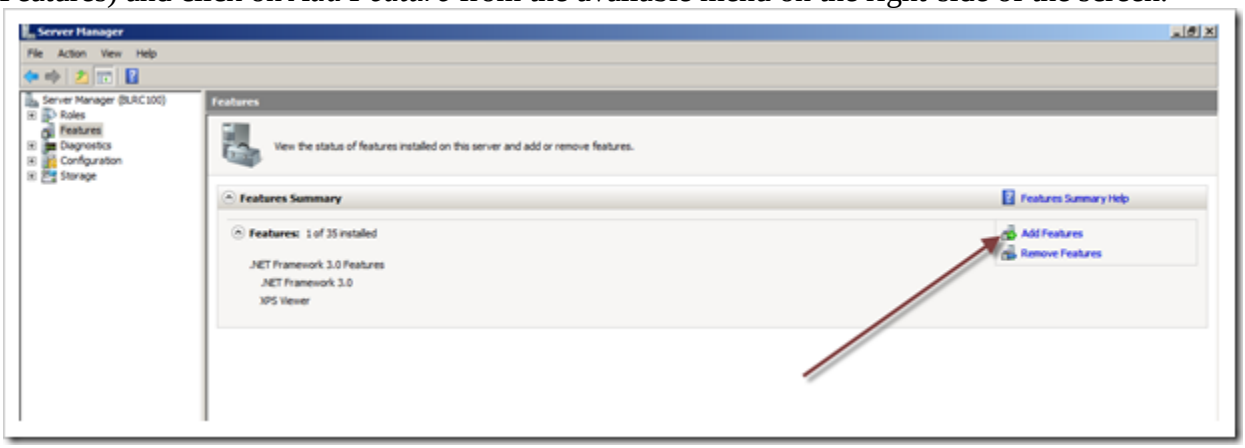
As we have installed and configured all the necessary hardware and software, its time to start the windows cluster installation. In the next post, we are going to start the installation of Windows Cluster.

Building a SQL Server 2008 R2 Cluster: Part IV

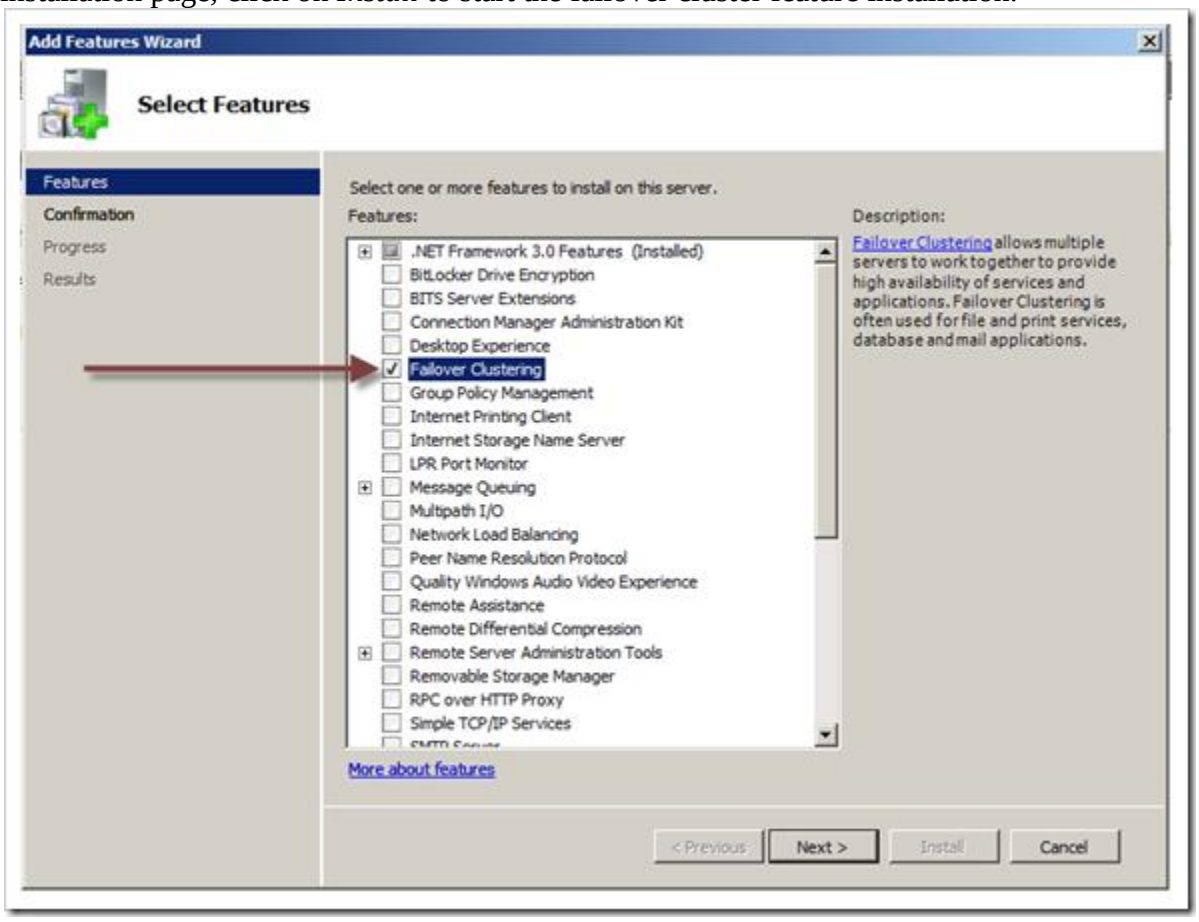
As of previous post, we have configured all the necessary parameters to start the Failover Cluster installation including dotnet framework 3.5. In this post, we are going to build the failover windows cluster on Windows Server 2008. I have started the node **BLRC100** and open the Disk Management (Server Manager –> Storage –> Disk Management). Bring Online all the available disks, which are presently offline as shown below.



Now we need to add the Failover Clustering feature. Open the Features (Server Manager –> Features) and click on *Add Feature* from the available menu on the right-side of the screen.



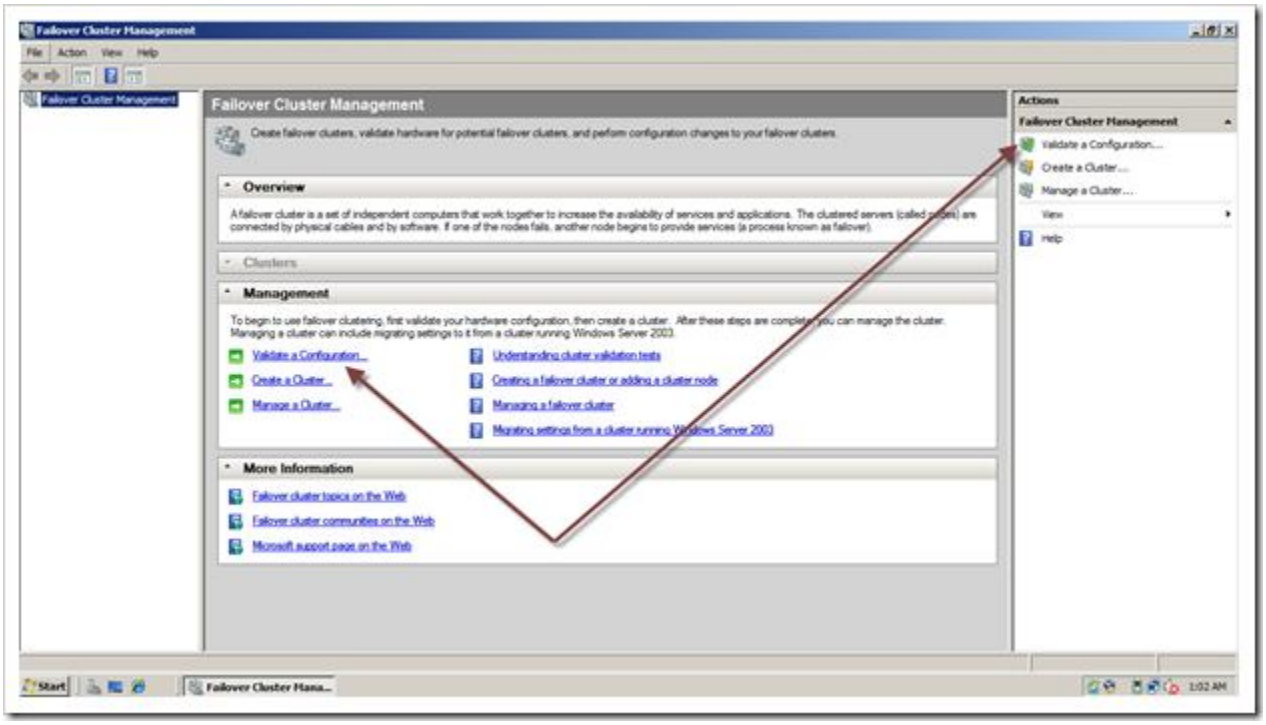
From the available option, select *Failover Clustering* and click on *Next*. On the confirm installation page, click on *Install* to start the failover cluster feature installation.



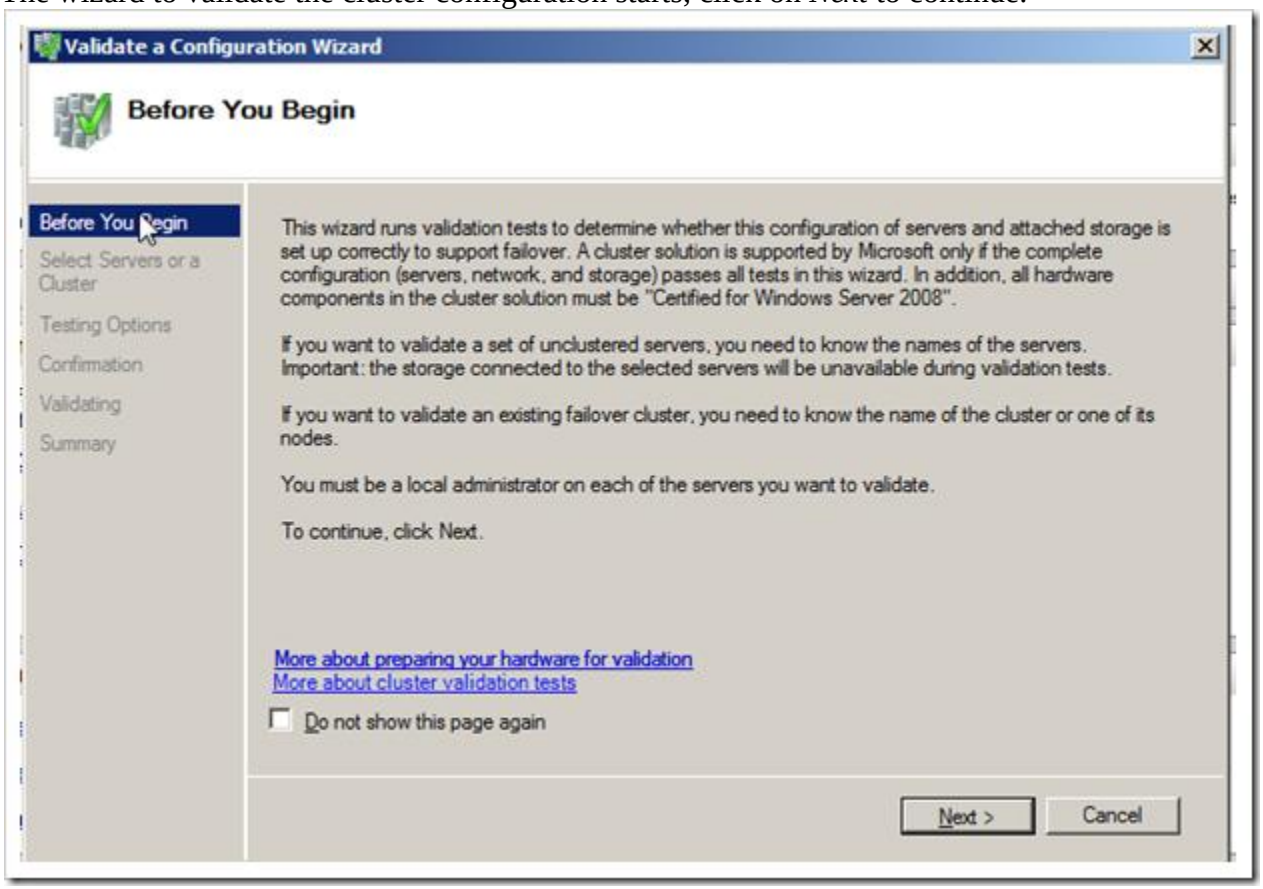
Restart the node BLRC100 once the failover cluster feature is installed. Install the failover cluster feature on the other node, in our case, BLRC101. Once you have install the failover cluster on BLRC101, shut it down. We are going to configure the failover clustering on the first node BLRC100 and later we will add the second node, BLRC101.

Validating the Failover Cluster Configuration

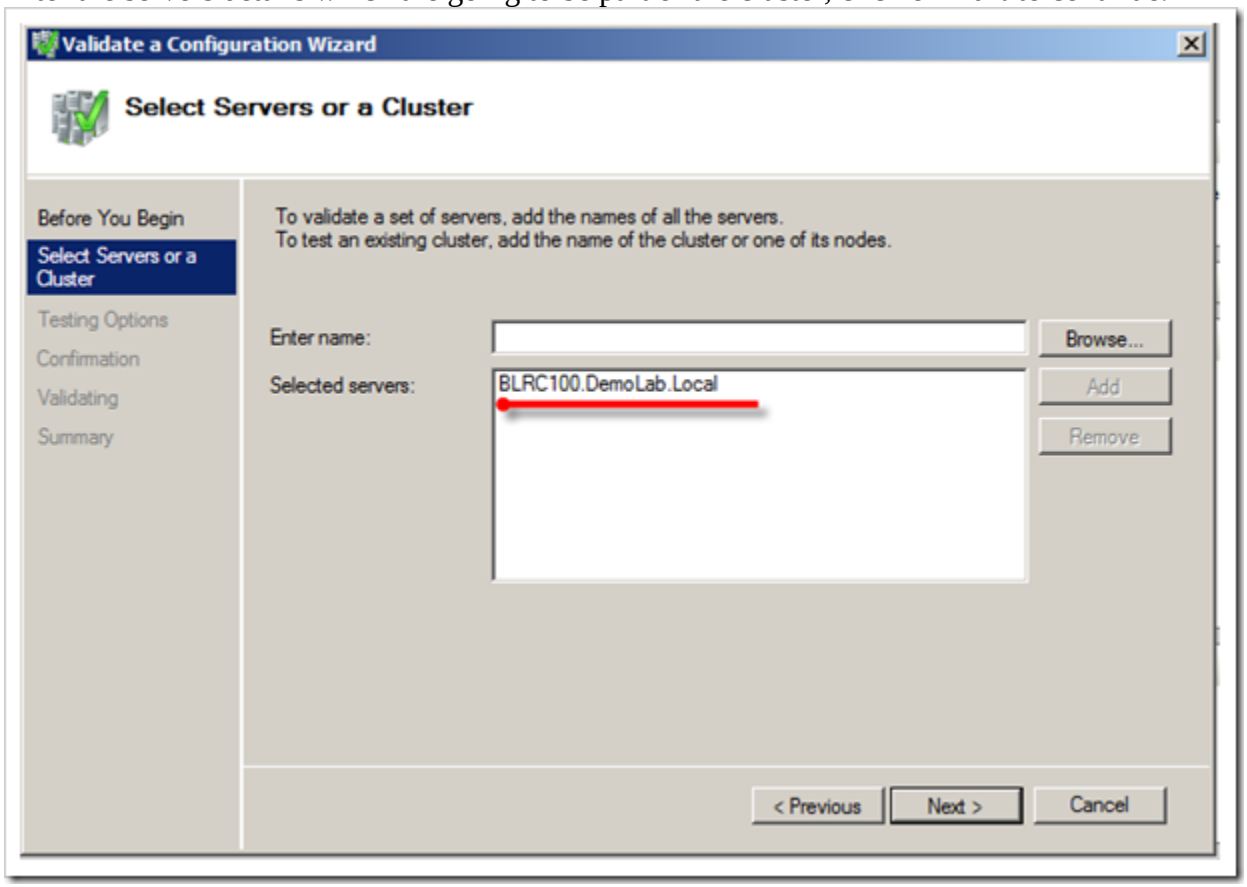
Now that I have logged on to BLRC100 using the domain administrator account and we are going to start the configuration of our Windows Cluster. To start with, open the Failover Cluster Management Console (Start Menu –> Administrative Tools –> Failover Cluster Management). We need to validate the hardware configuration of our server, click on *Validate a Configuration* as shown below.



The wizard to validate the cluster configuration starts, click on *Next* to continue.

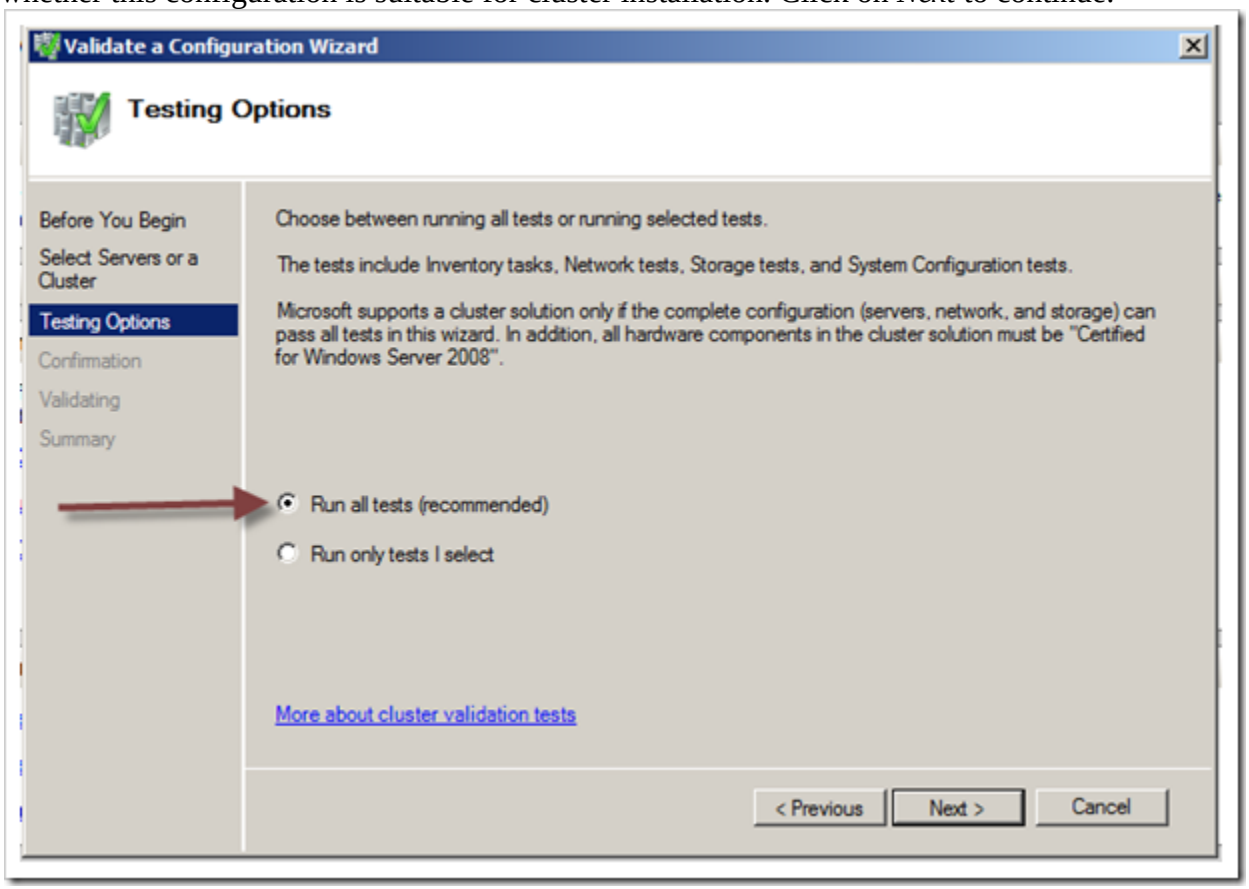


Enter the servers details which are going to be part of the cluster, click on *Next* to continue.

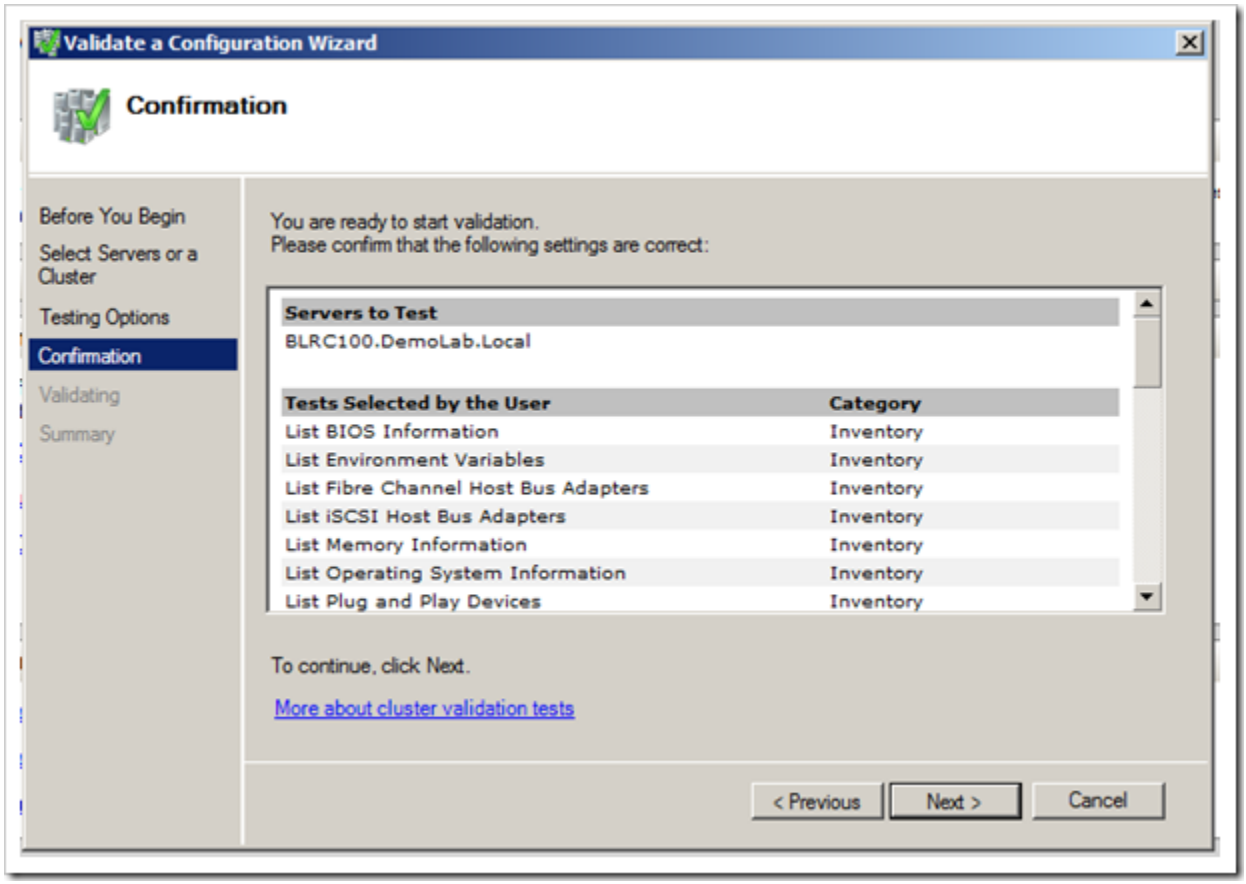


Select the testing options (default, *Run all tests*). This tests include Network tests, storage, system configurations etc. This will validate the server configuration and provide a report

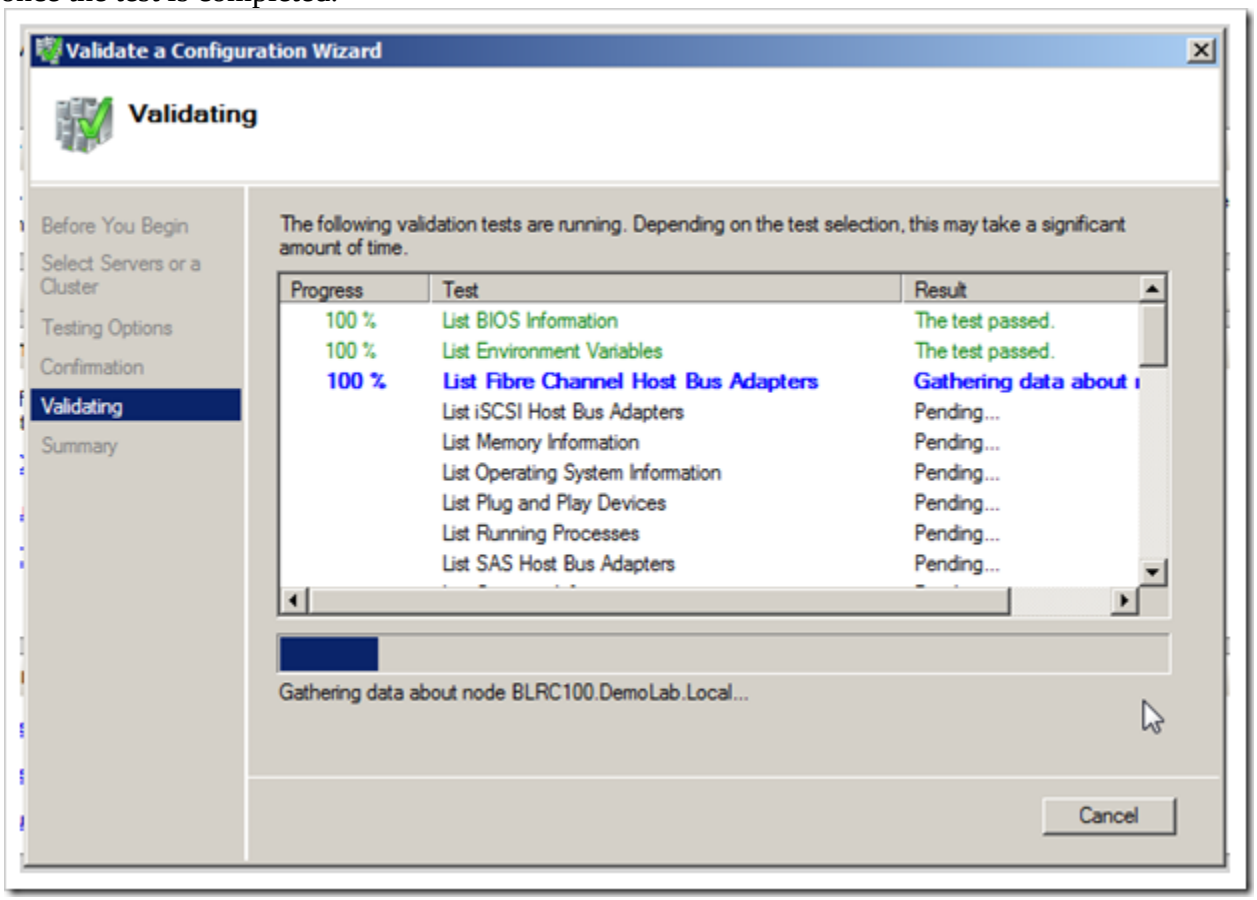
whether this configuration is suitable for cluster installation. Click on *Next* to continue.



This page provides details about all the tests which are going to run. Verify and click on *Next* to start the test.

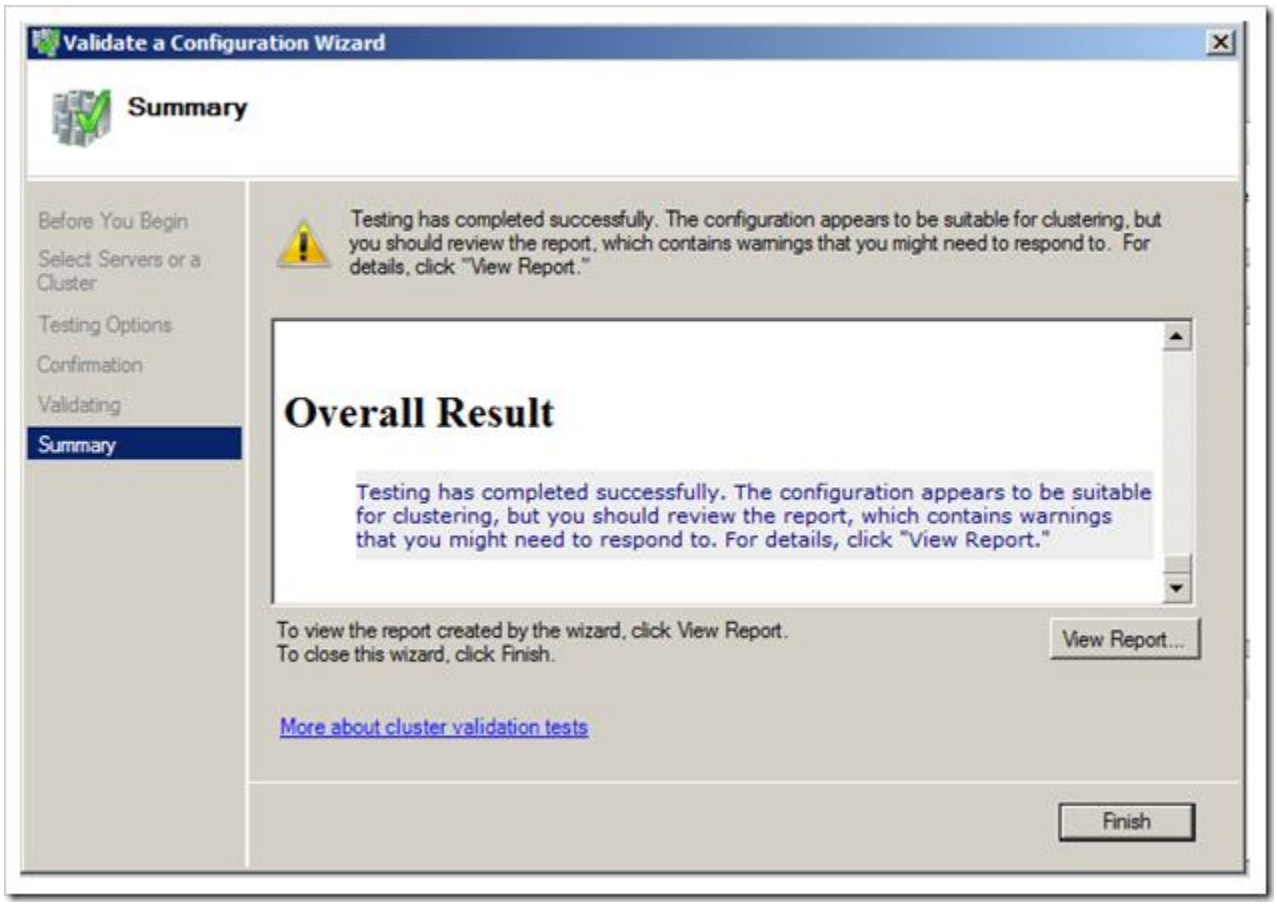


Now the test is running on the server BLRC100.Demolab.Local. You will get a detailed report once the test is completed.



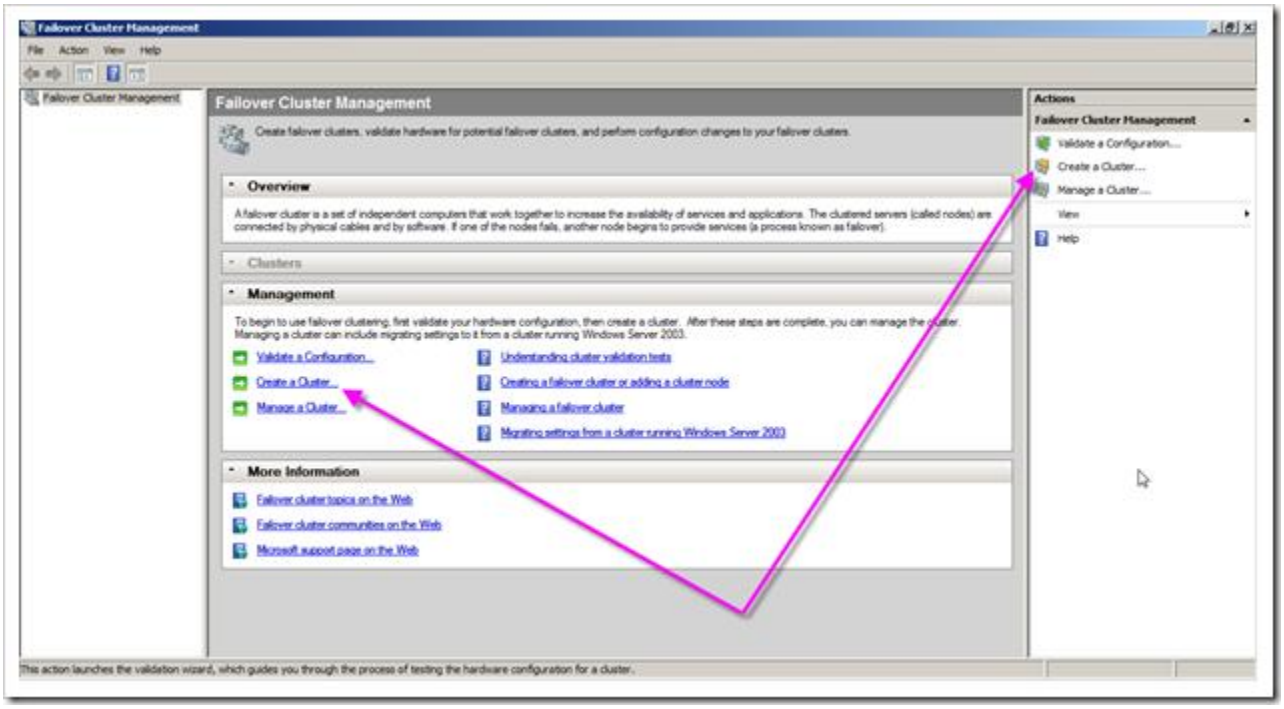
Once the test completed, it will provide a detailed report about the current hardware, network and storage configuration. The validation report of our demo server contains some warnings. Since we are going to use it for learning purposes, we are proceeding further. In case of production environment, **do not proceed** further until you fixed all the warnings and your validation report returns all success report. I have executed the validation report on the other node, thus, I am moving to the next part of installation. Click on *Finish* to close the cluster

validation wizard.

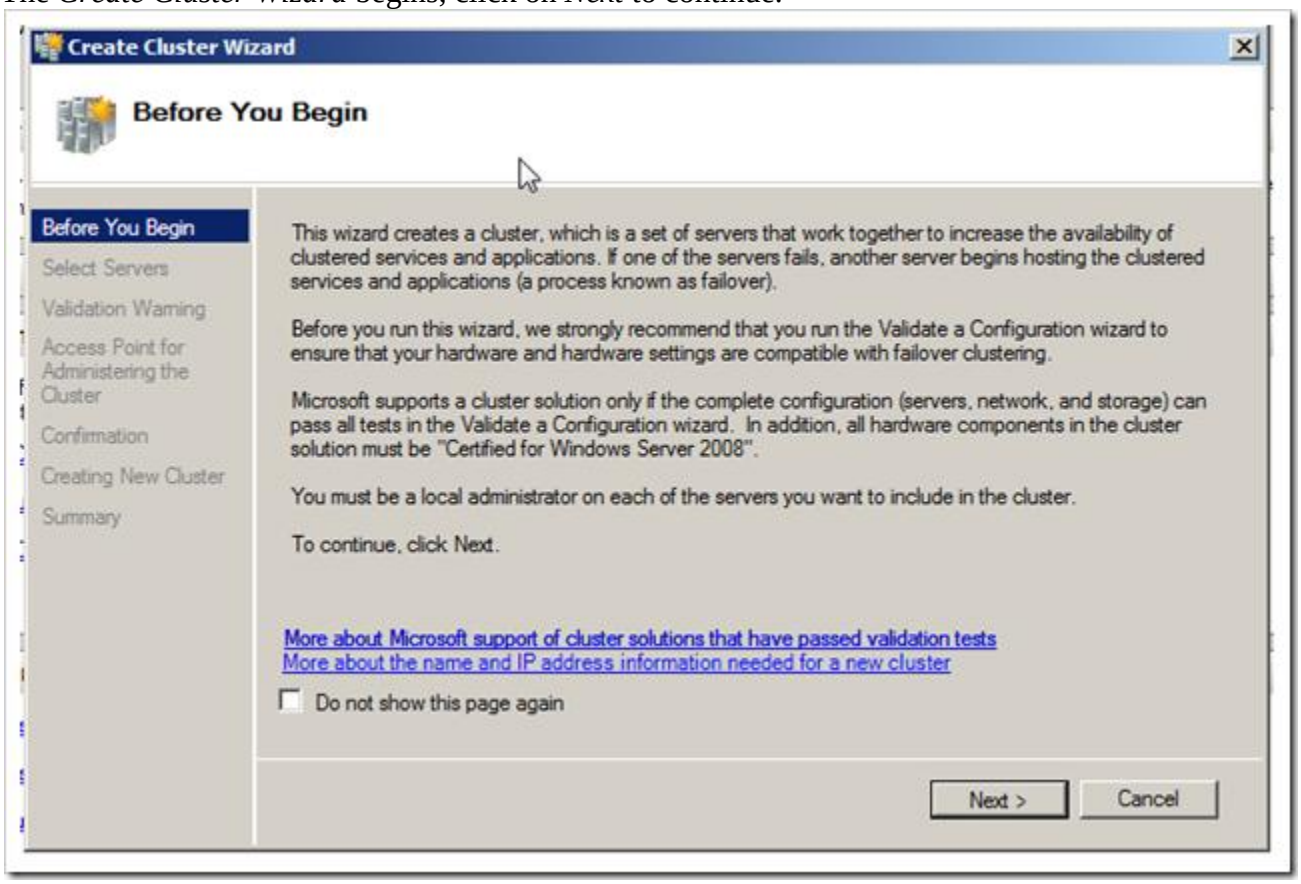


Creating a Failover Cluster

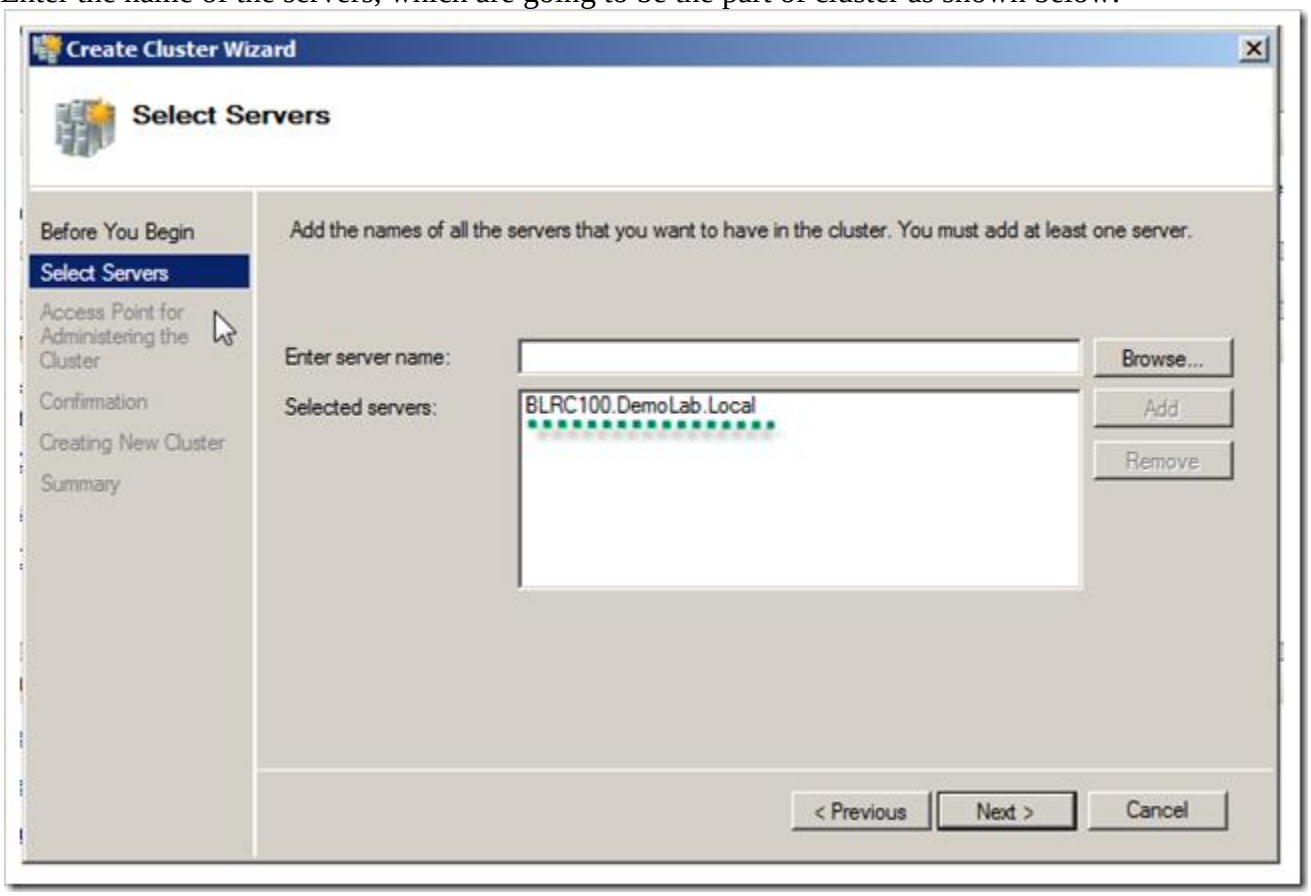
Since the validation of failover cluster wizard completed, now we are going to create the cluster. Click on *Create a Cluster* as shown below.



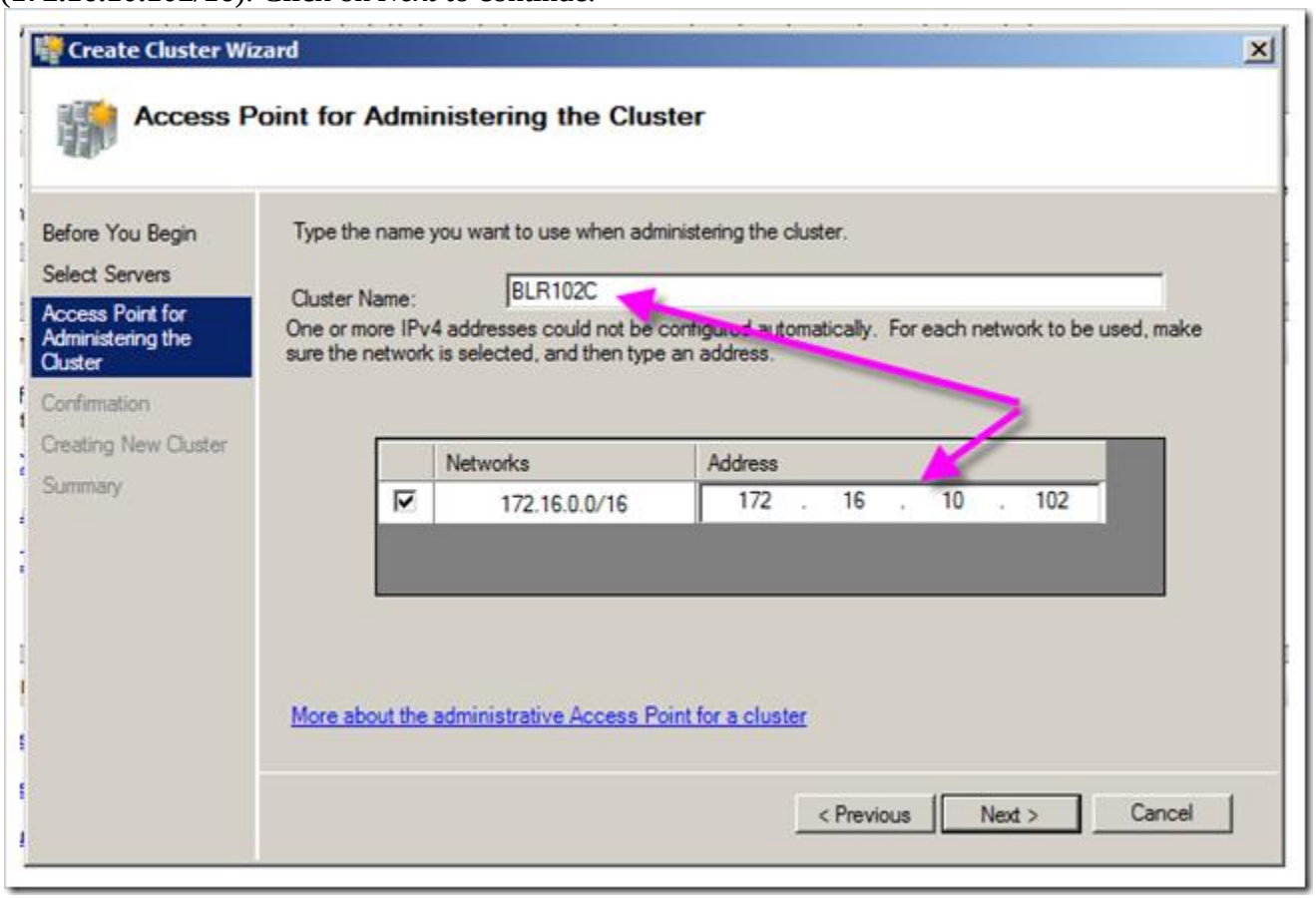
The *Create Cluster Wizard* begins, click on *Next* to continue.



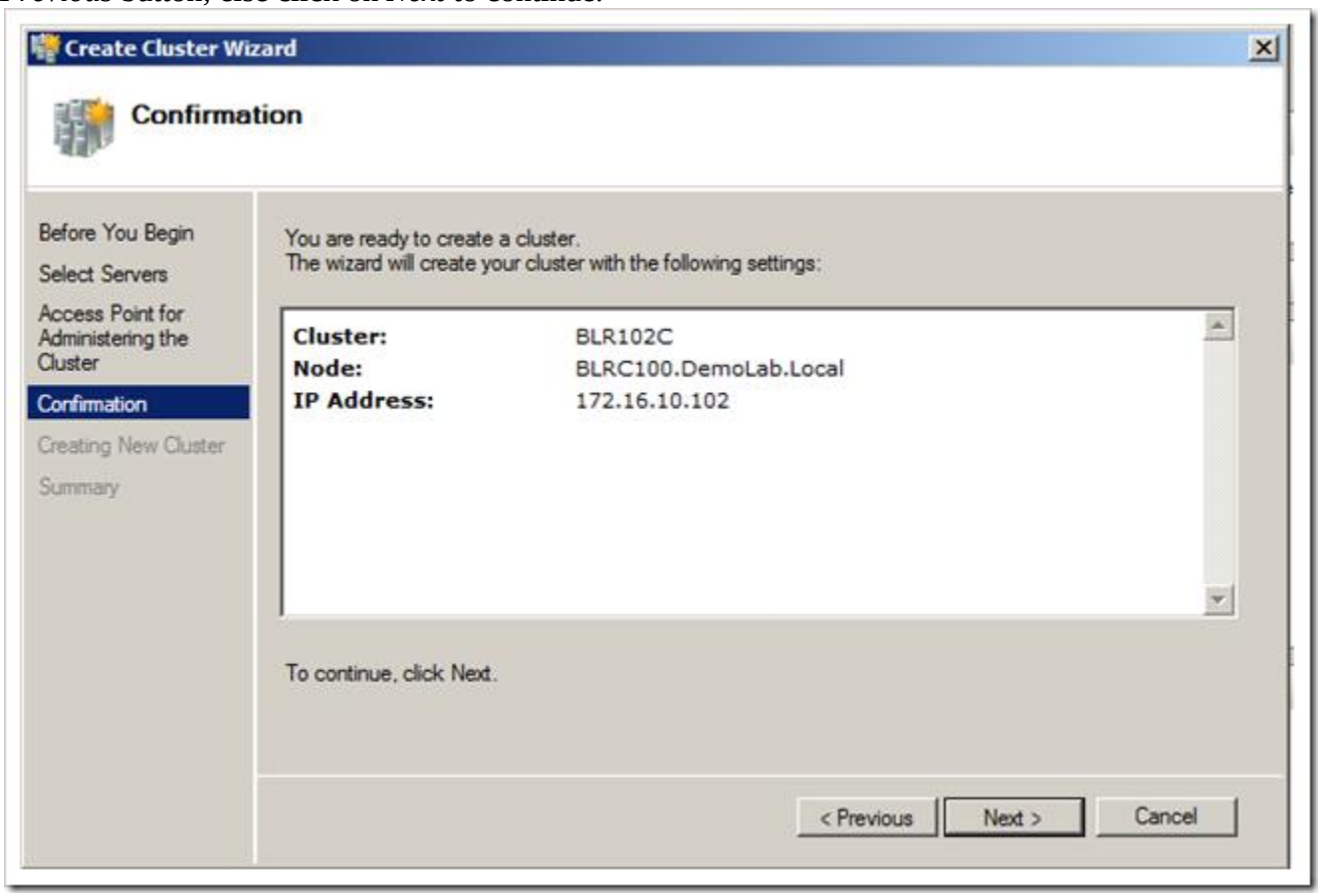
Enter the name of the servers, which are going to be the part of cluster as shown below.



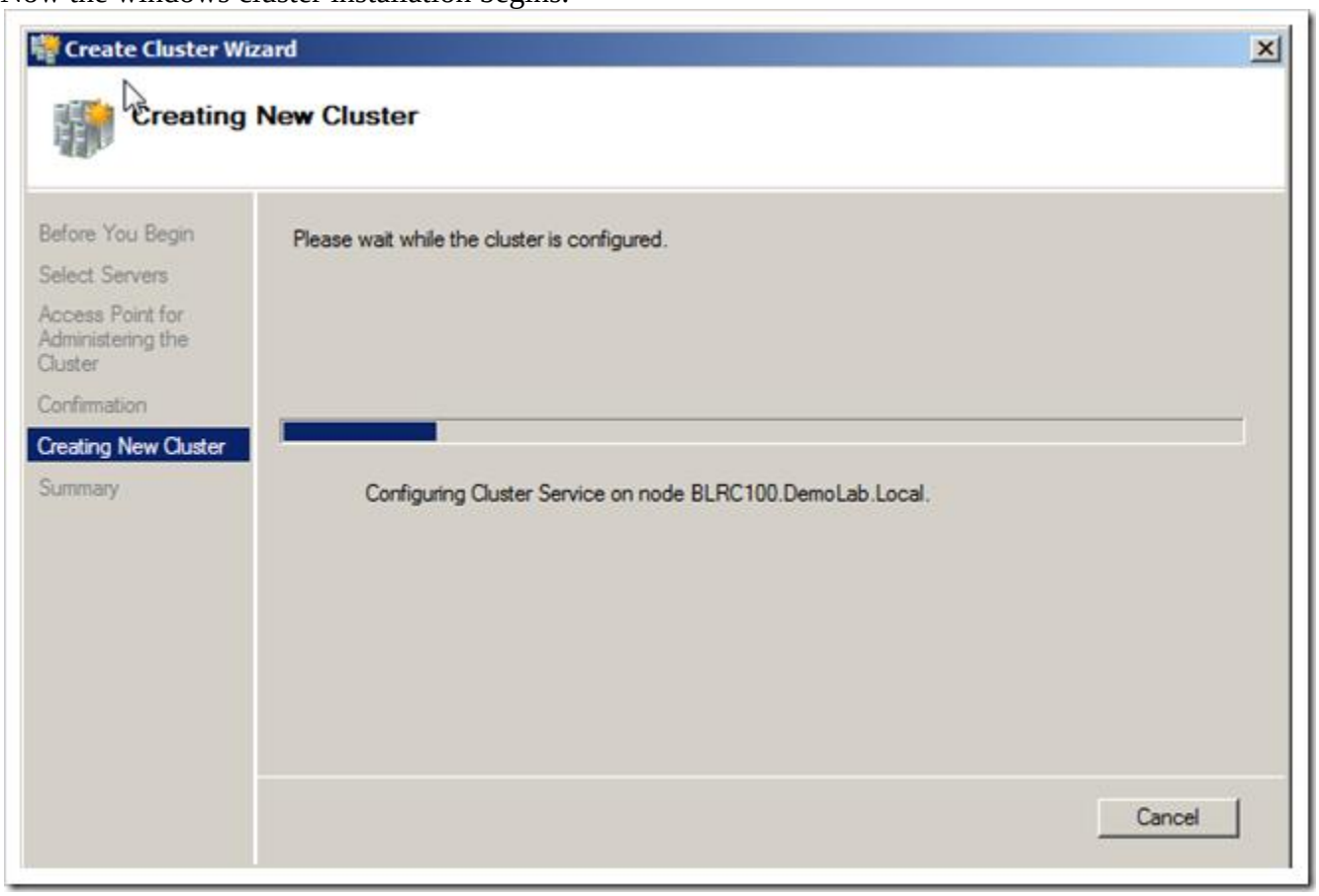
Enter the name of our Failover Cluster, **BLR102C**, and the IP Address to connect to this cluster (**172.16.10.102/16**). Click on *Next* to continue.



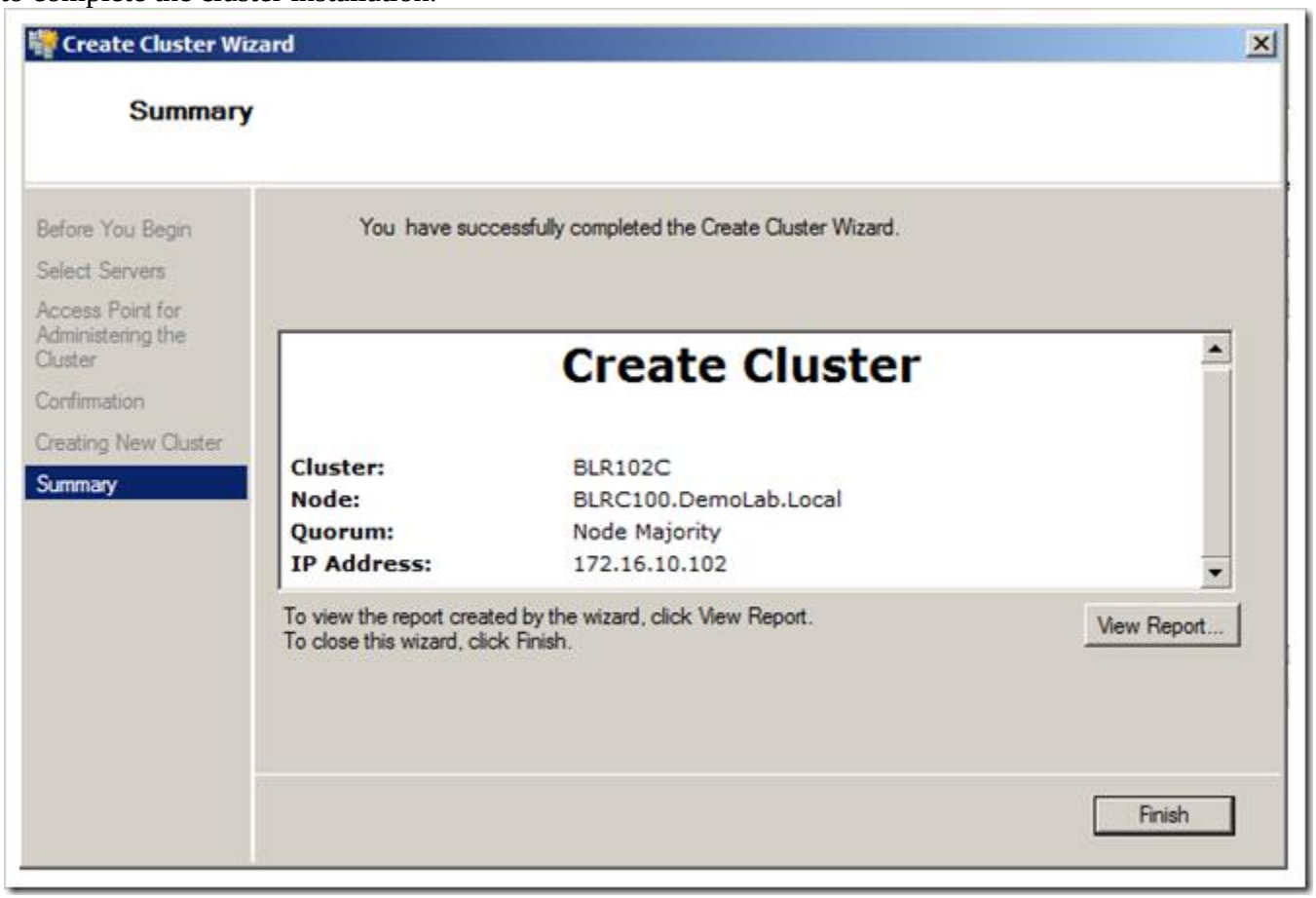
Verify all the settings for this cluster. If you want to modify the information, click on the *Previous* button; else click on *Next* to continue.



Now the windows cluster installation begins.

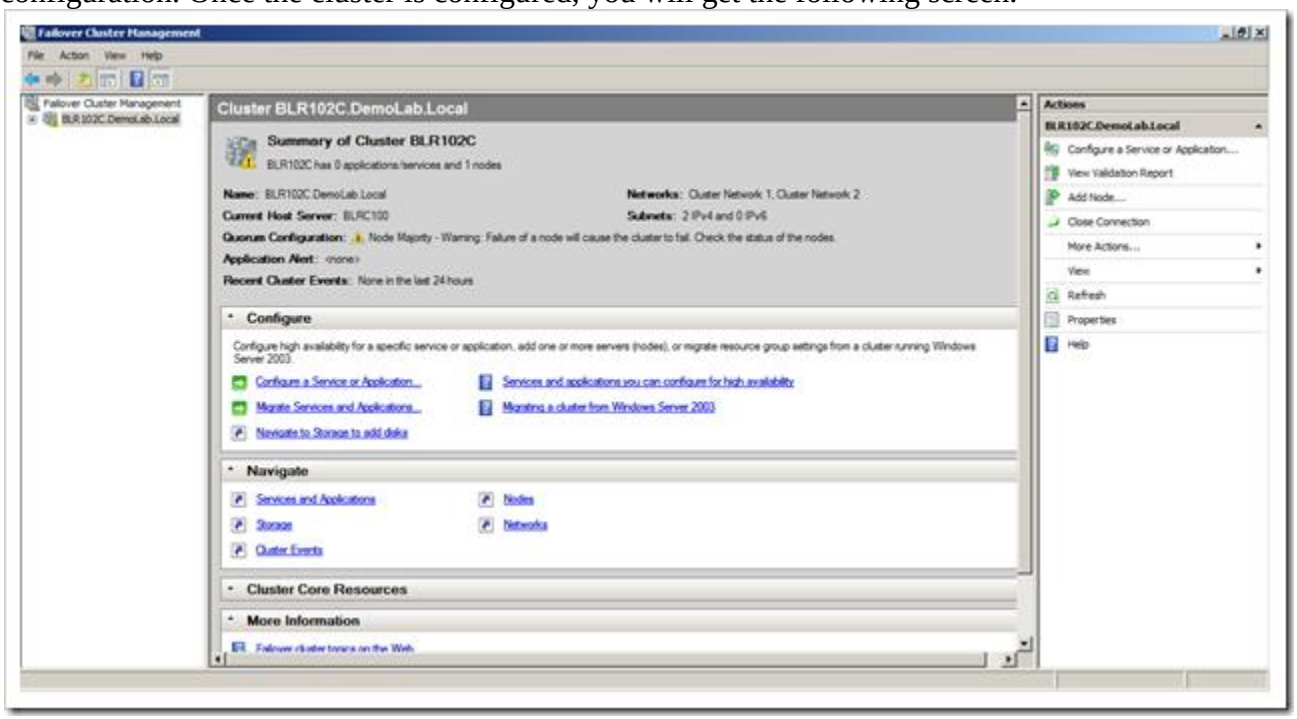


Once the installation is completed, you will get the following Summary report. Click on *Finish* to complete the cluster installation.



As you can see from the above screenshot, the quorum of this cluster is set as Node Majority. Also the steps to build a cluster on Windows Server 2008 is different than the previous versions of Windows. Since we are going to build a two-node cluster using Windows Server 2008, we can not use the Quorum type as ***Node Majority***. Later, we are going to change the Quorum

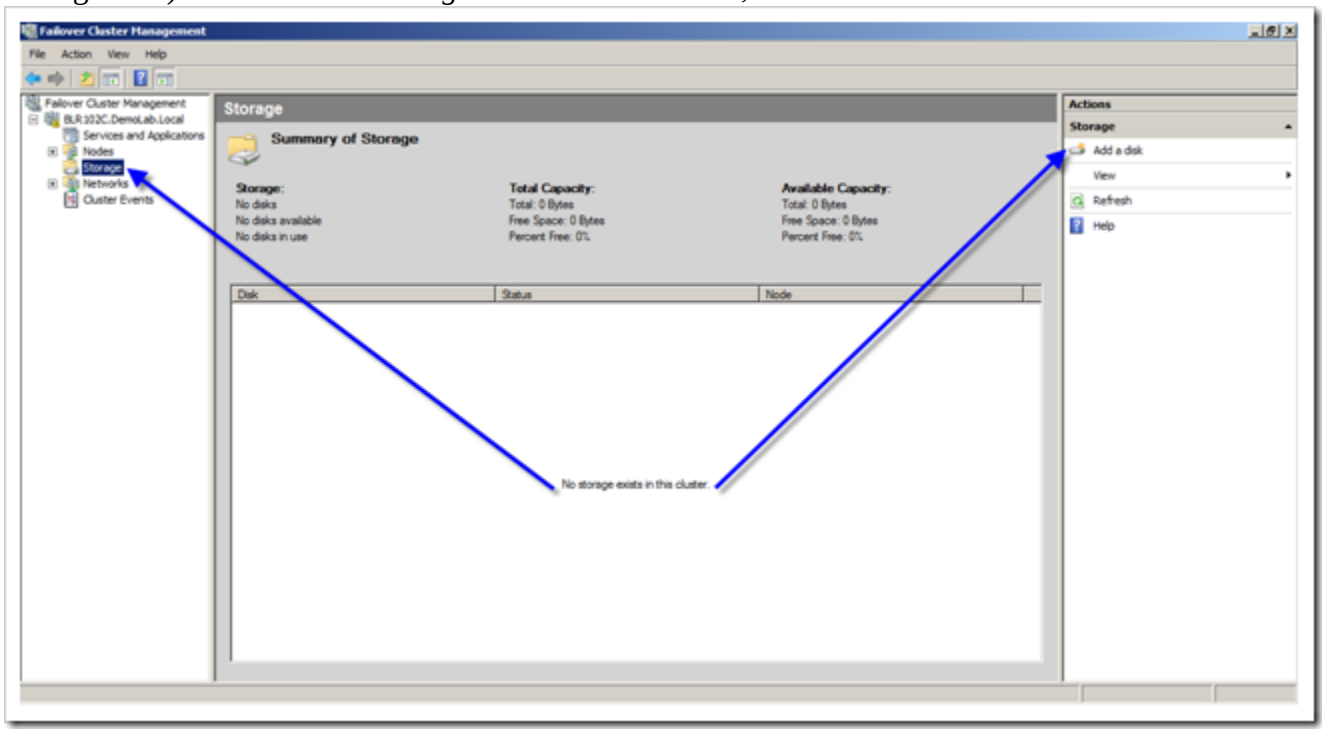
configuration. Once the cluster is configured, you will get the following screen.



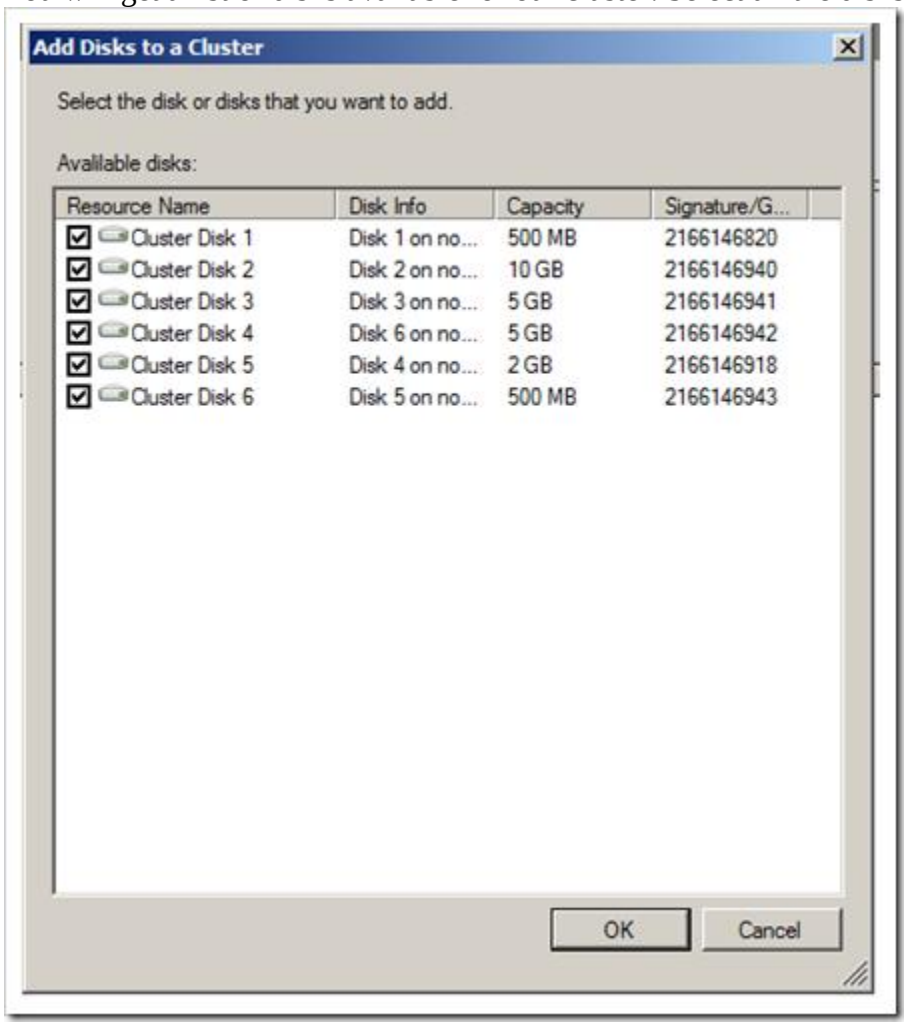
Our cluster BLR102C is ready; however, before adding the other node, we need to add the disks to our cluster, which also include the ***Quorum disk***. Once we will add the disks, then we will add the other node of the cluster.

Adding Disks to Cluster

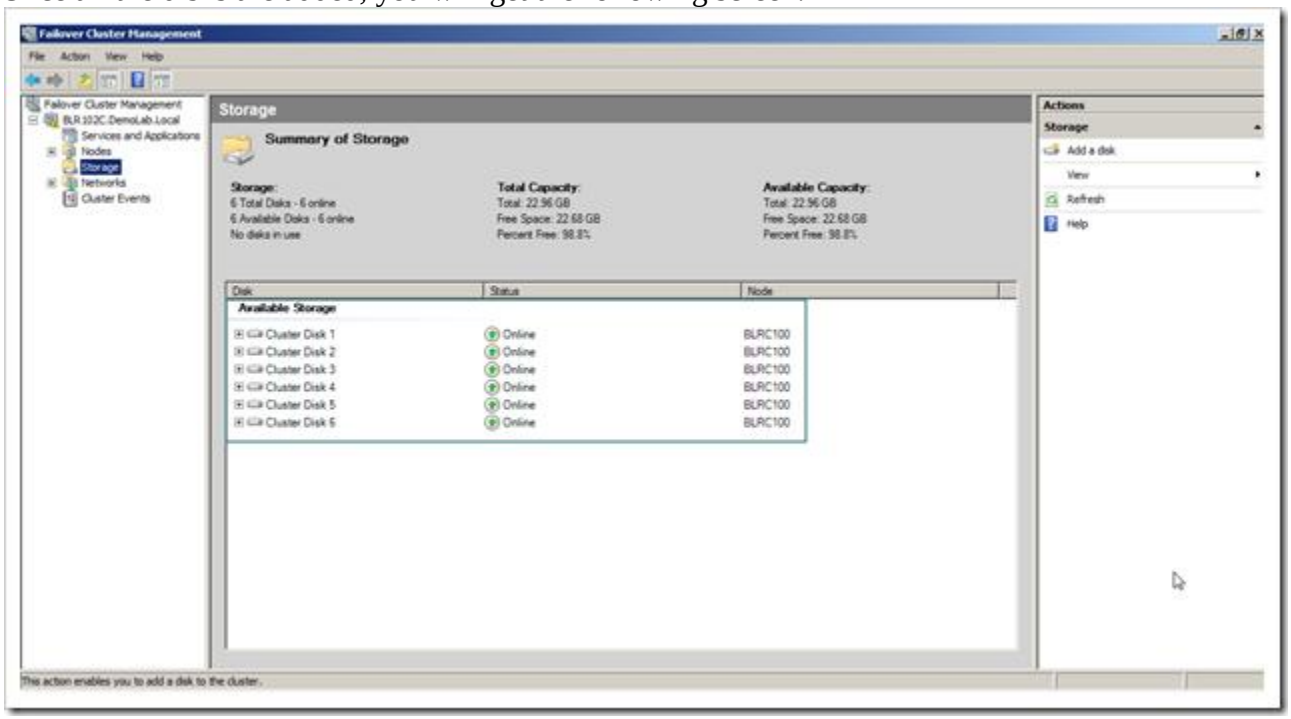
Open the *Failover Cluster Management*(Start —> Administrative Tools –> Failover Cluster Management) and browse to *Storage*. Click on *Add a disk*, to add disks to our cluster.



You will get a list of disks available for our cluster. Select all the disks and click on *OK*.



Once all the disks are added, you will get the following screen.



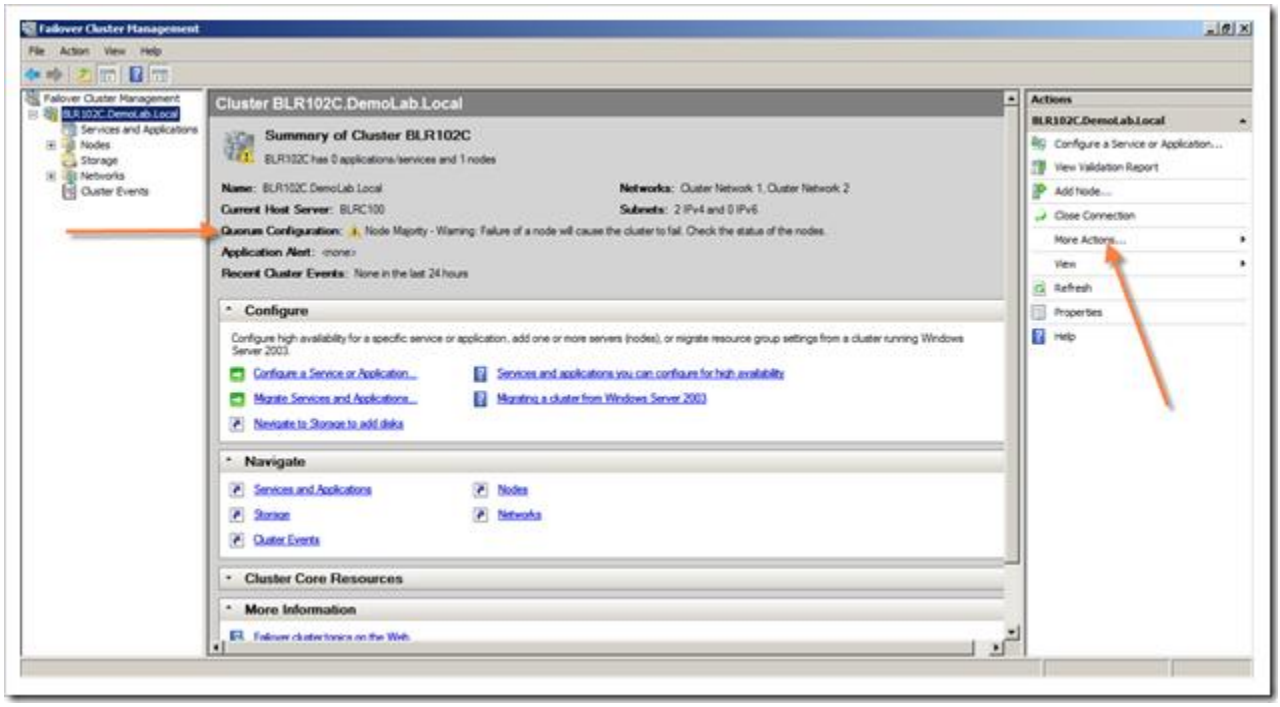
In this post, we have install failover cluster feature and create a windows cluster. We have also added all the available disks to the cluster. In the next part of this series, we are going to re-configure the Quorum setting and add the other node to complete the Windows Cluster installation.

Building a SQL Server 2008 R2 Cluster: Part V

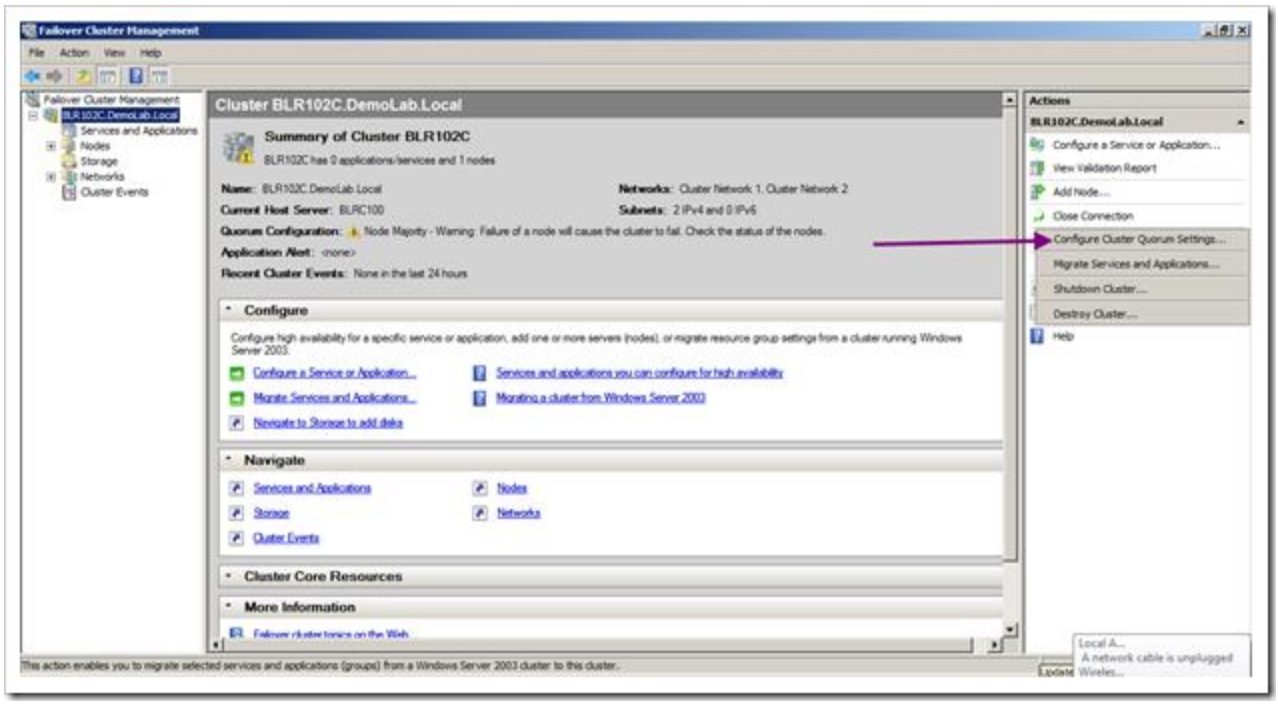
In the previous part of this series, we have created the windows cluster and added the disk resources. In this post, we are going to re-configure the Quorum setting of this cluster, and then we are going to add the second node of this cluster.

Configuring the Quorum of the Cluster

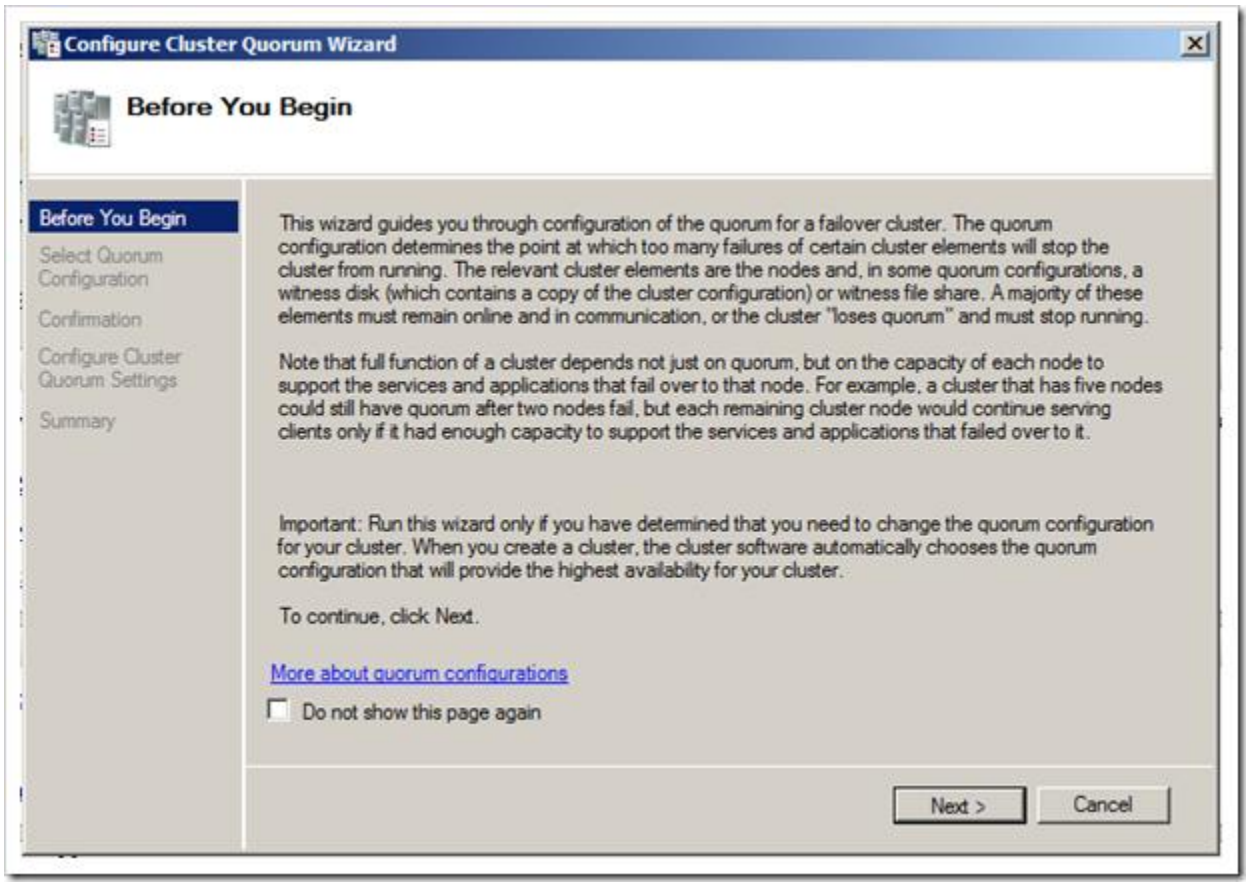
When we create a cluster using Windows Server 2008, by default, the quorum is set as Node Majority. However, this feature is not available for a two-node cluster. To change the quorum setting, click on *More Actions* as shown below.



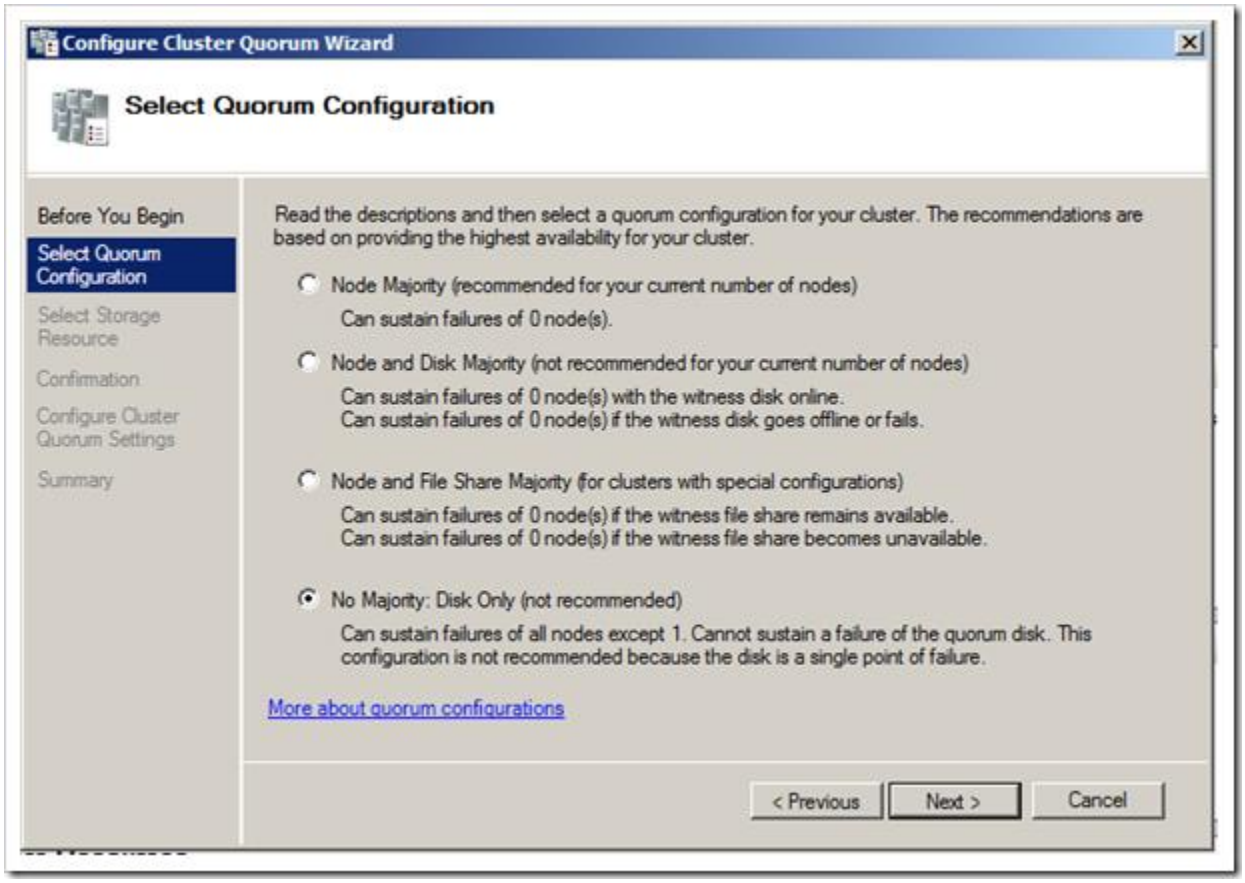
From the drop-down menu click on *Configure Cluster Quorum Settings* to change the quorum configuration setting.



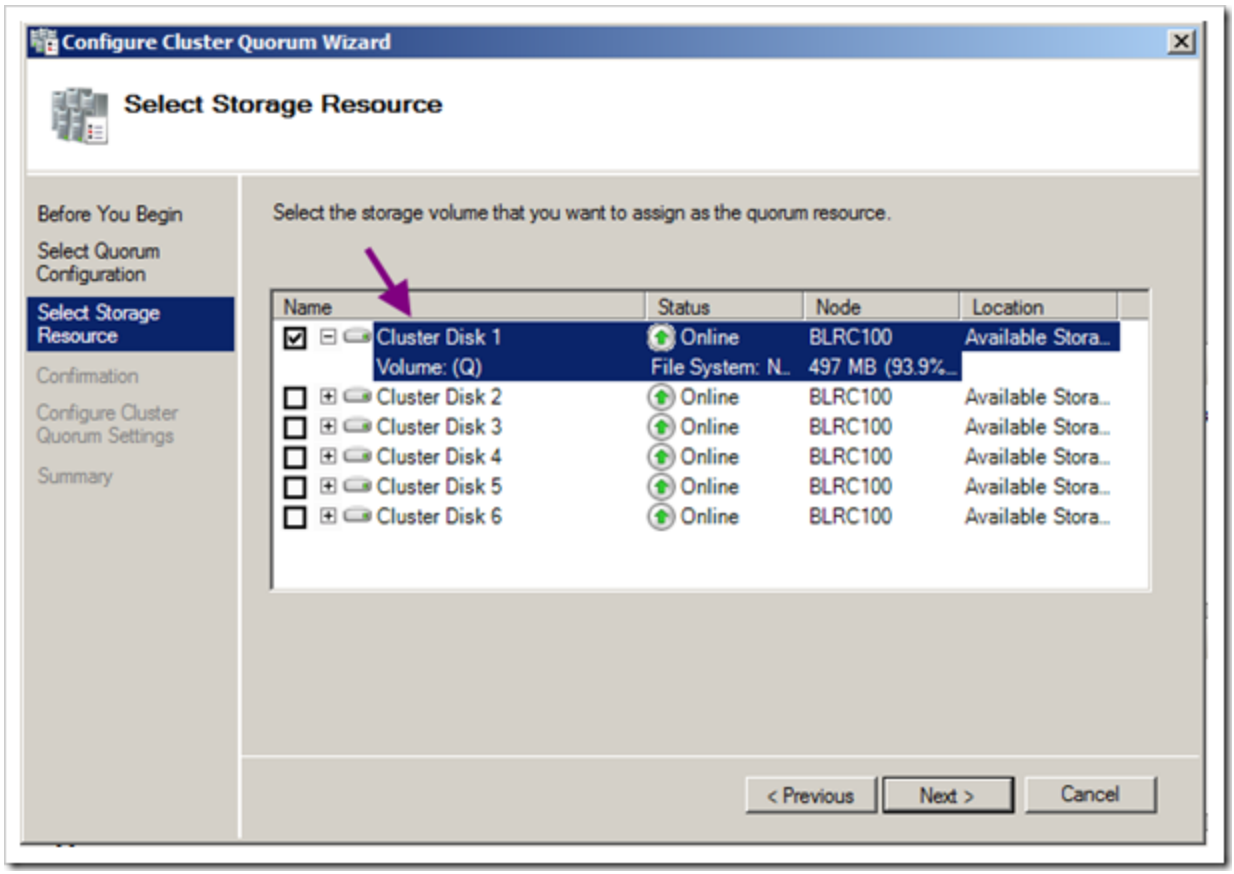
The *Configure Cluster Quorum* wizard will appear, click on *Next* to continue further.



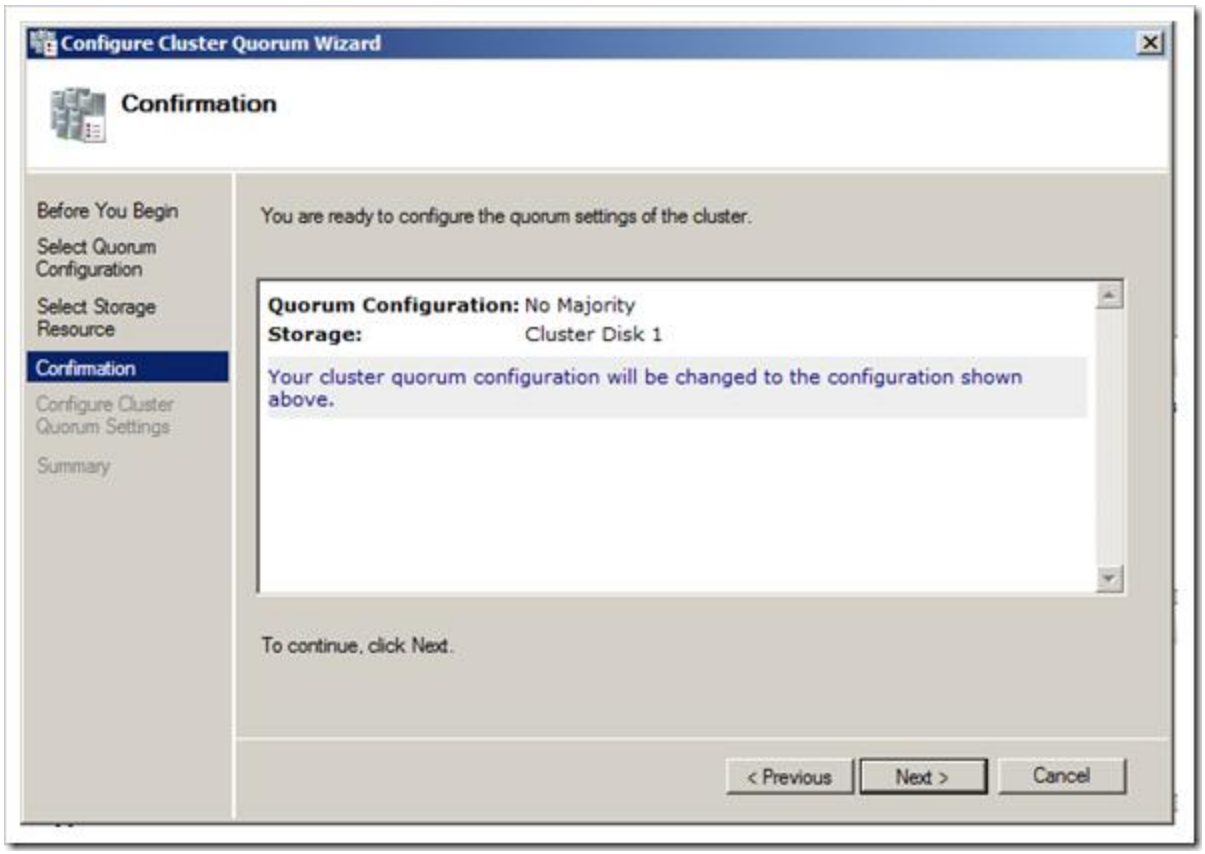
On the Quorum configuration page, select *No Majority: Disk Only* and click on *Next* to continue.



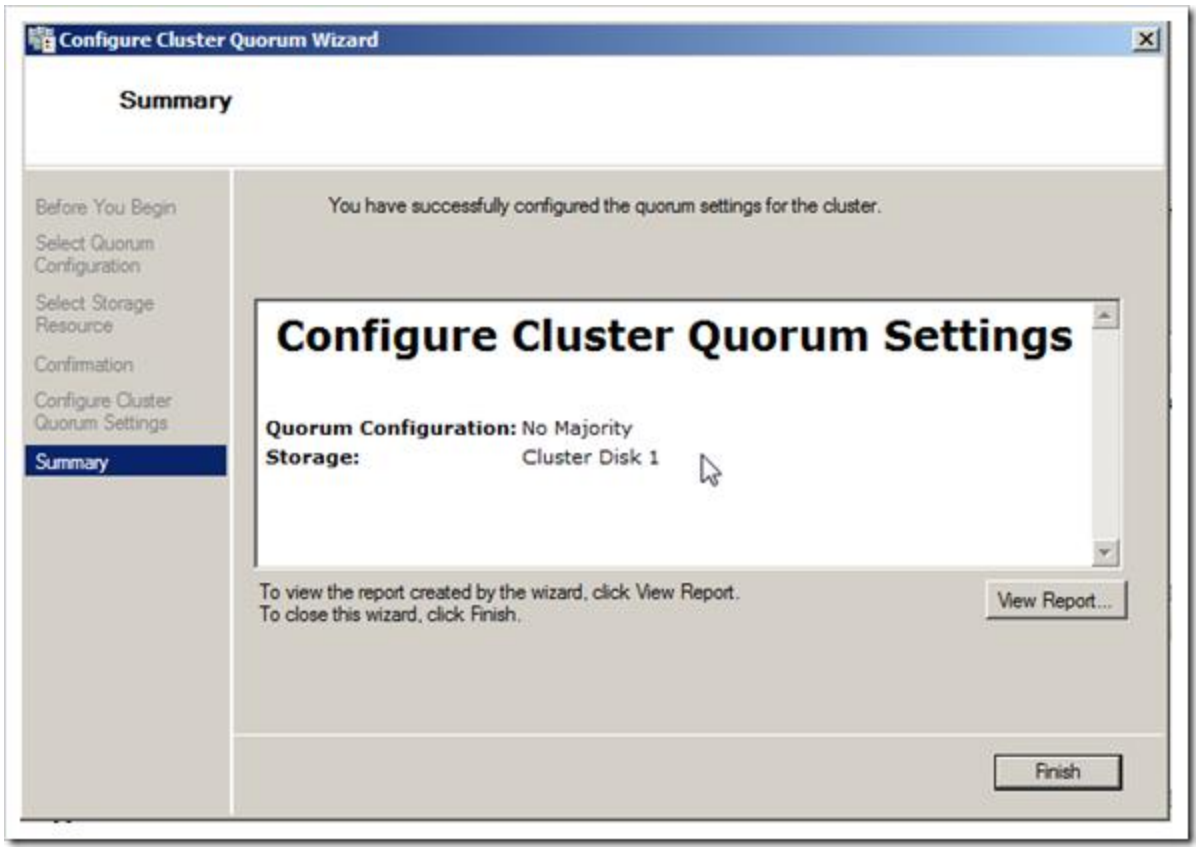
On the select storage resource page, select the disk for the Quorum from the available list of disks. In our case, the Disk 1 is used as a Quorum disk (with drive letter **Q**). Click on *Next* to continue further.



Confirm the quorum setting and click on *Next* to continue.



Once the quorum is configured, you will get the *Summary* page, click on *Finish* to complete the quorum re-configuration.

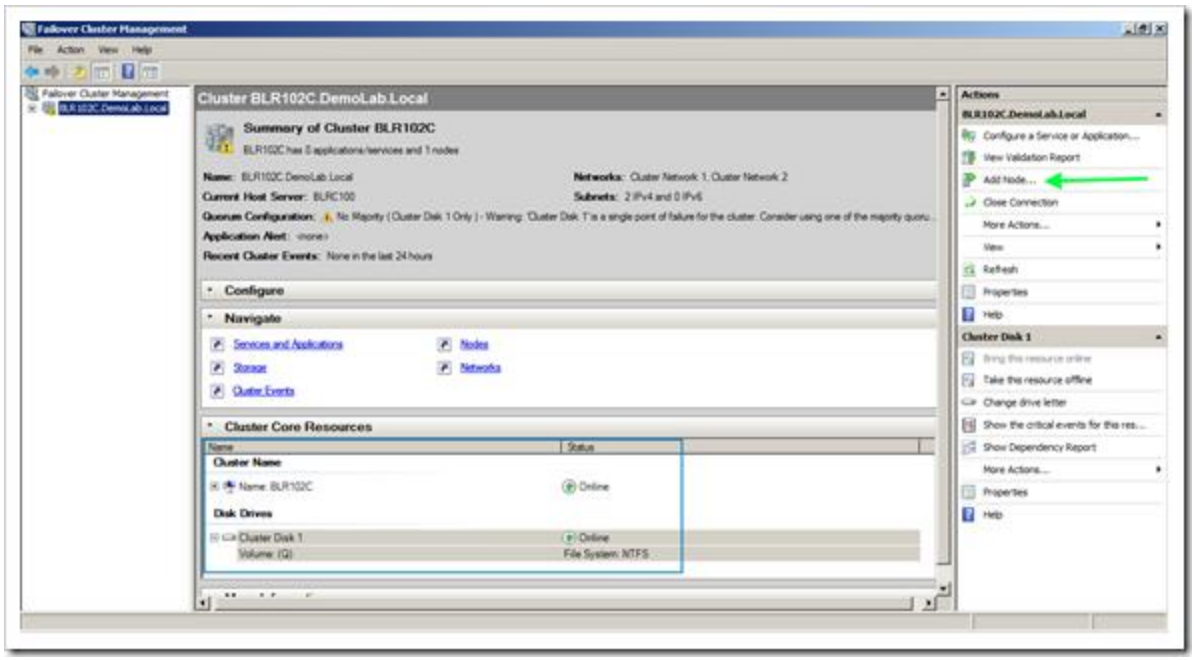


Now that we have re-configured our quorum disk, its time to add the second node of our cluster.

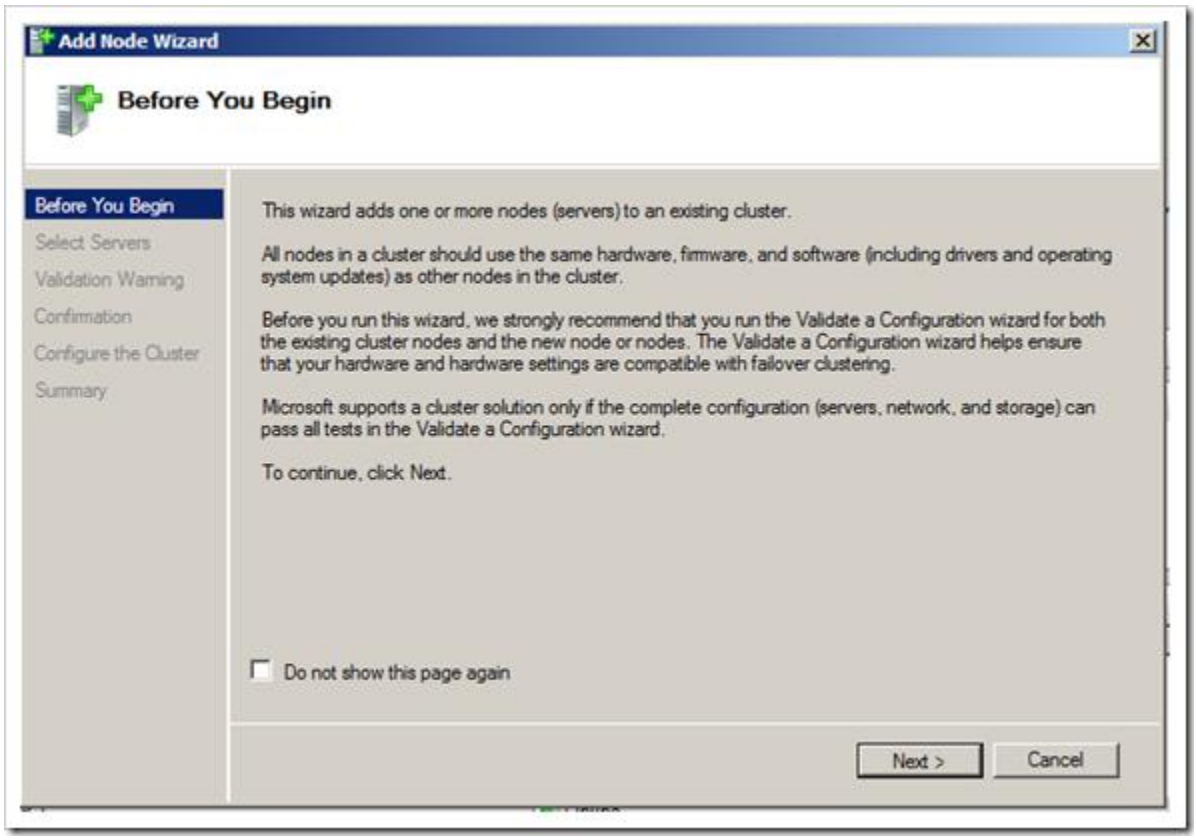
Adding Node to the Cluster

What we have done so far in this series on SQL Server Cluster installation? We have configured the storage server, shared the disks with the cluster nodes, configure the Cluster, added the disk resources, and re-configured the Quorum. Now the only remaining part of this two-node cluster installation is **Addition** of the second node.

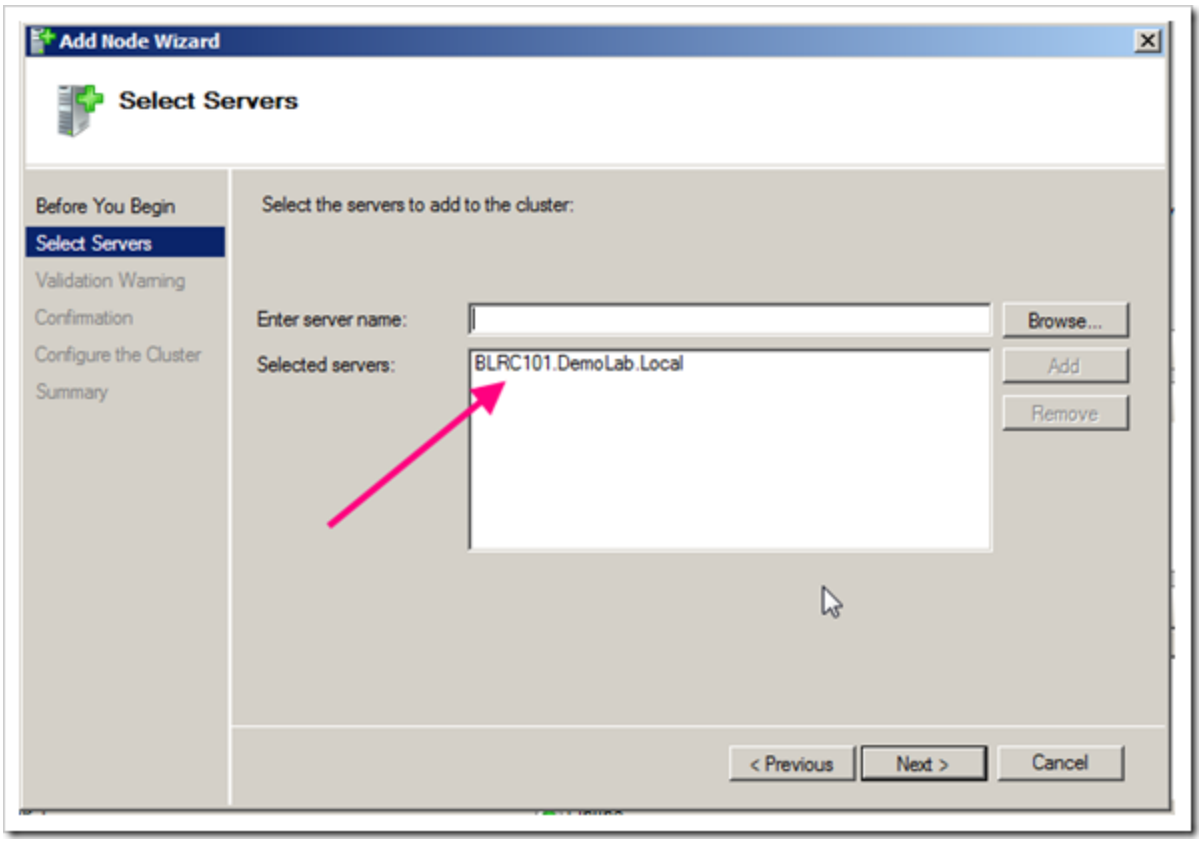
To begin with, open the Failover Cluster Management and connect to the cluster **BLR102C** and click on *Add Node* from the menu available on the right-side of the screen as shown below:



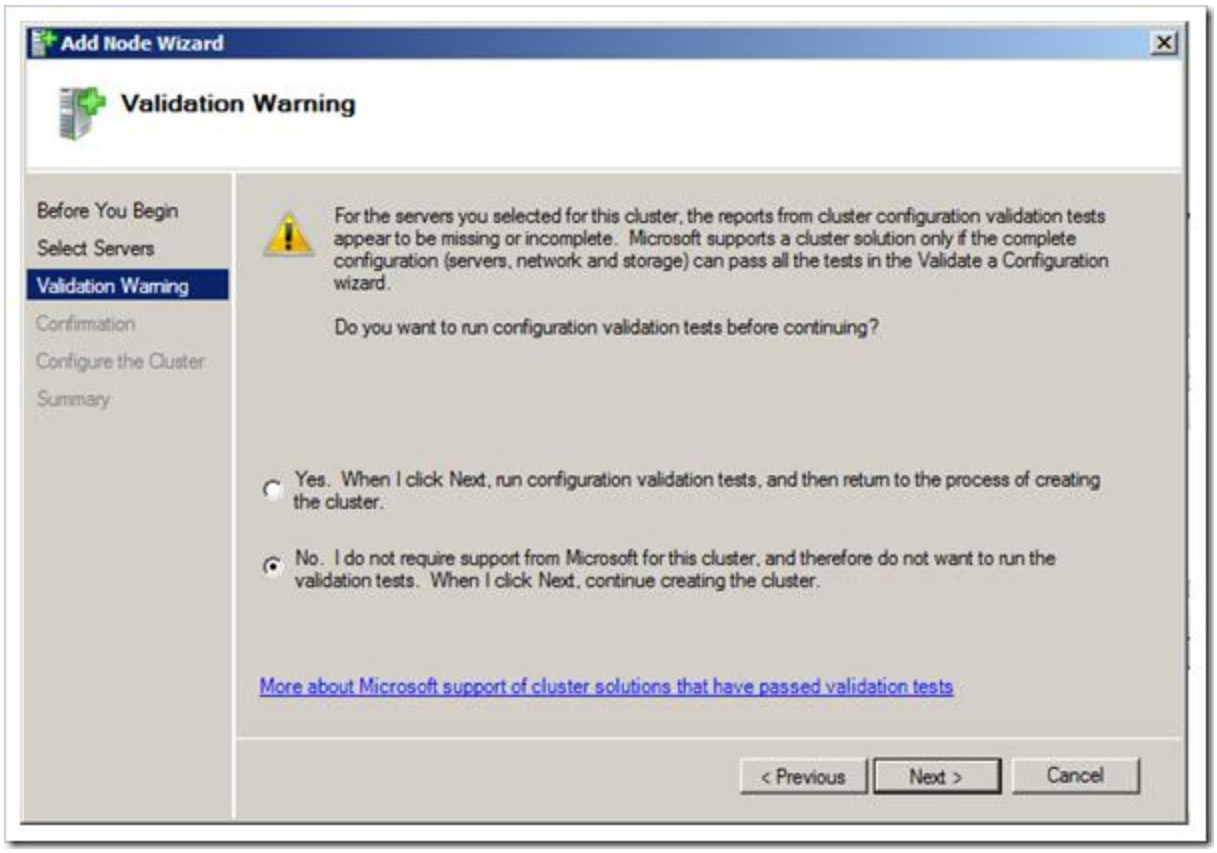
The *Add Node* wizard begins, click on *Next* to continue.



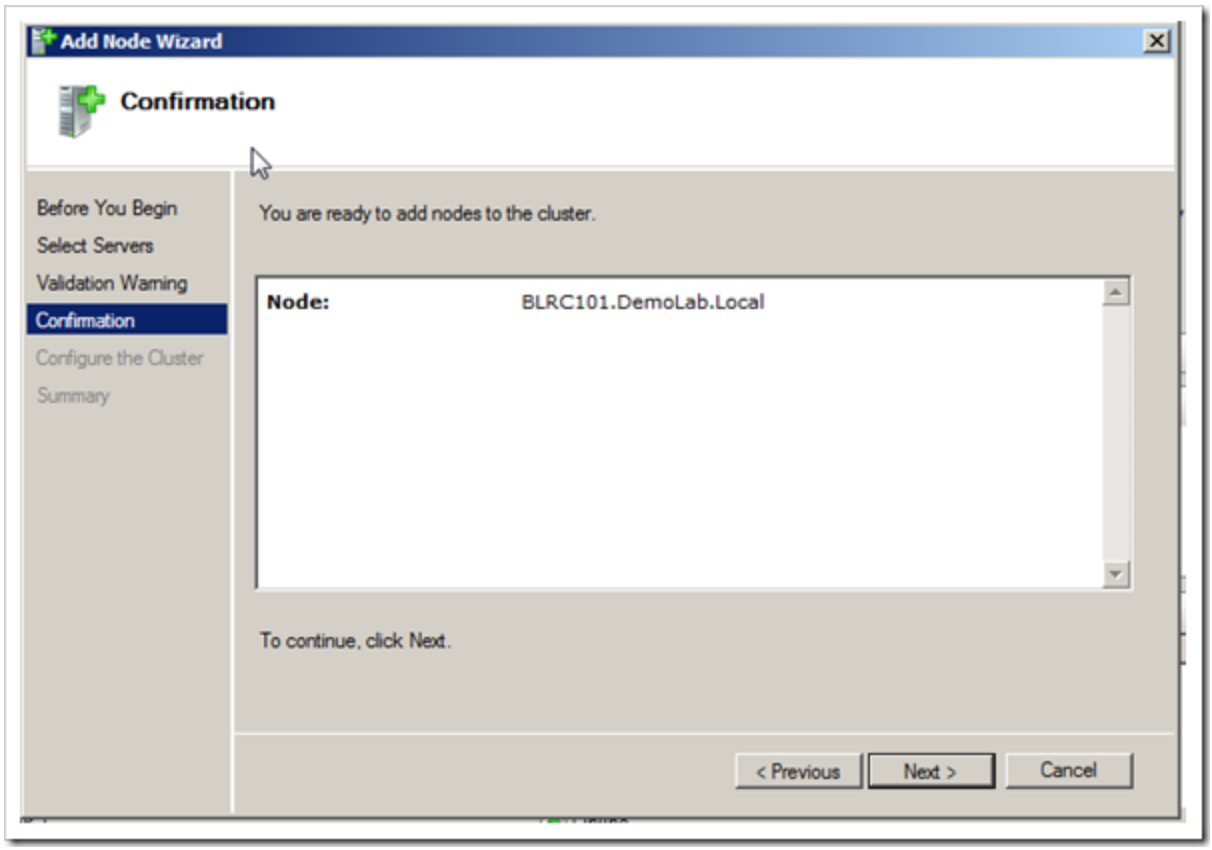
Select the node which we are going to add to the cluster, in our case, **BLRC101**.



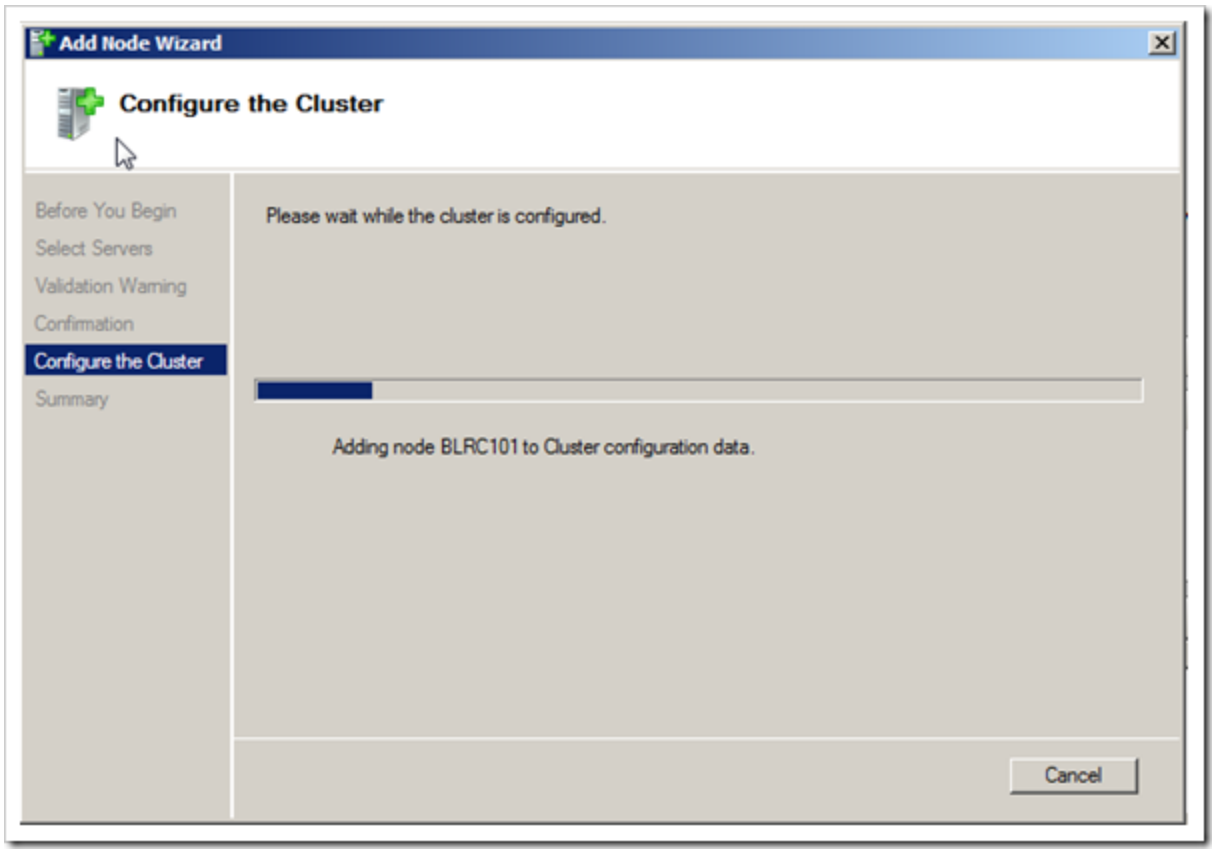
As there were some warnings in our cluster validation report, we will get the following screen. Since this is for learning purpose, I am not going to contact Microsoft Support for this cluster, so I am continuing to add the node without re-validating the cluster configuration. As mentioned earlier, for the production environment, you should make sure the validation wizard completed successfully.



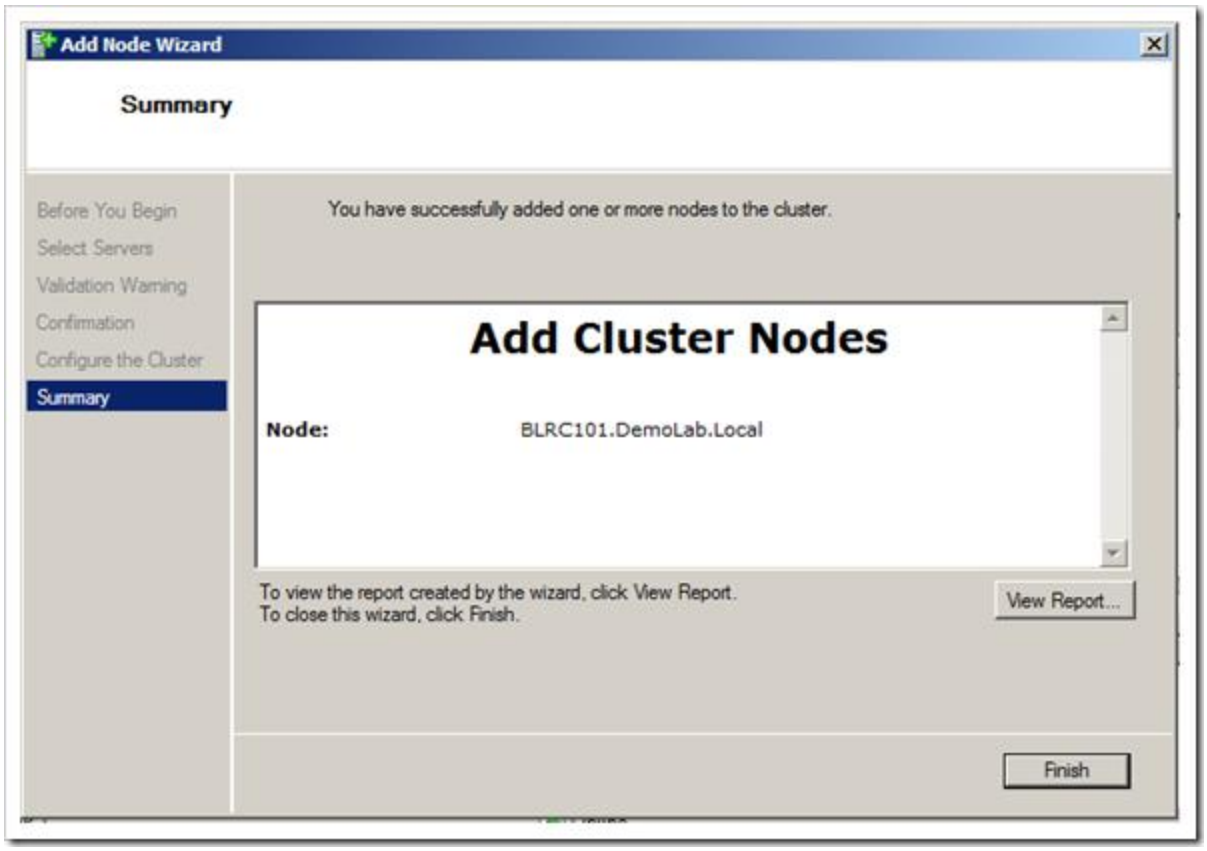
Confirm the details of the node you are going to add, and click on *Next* to continue.



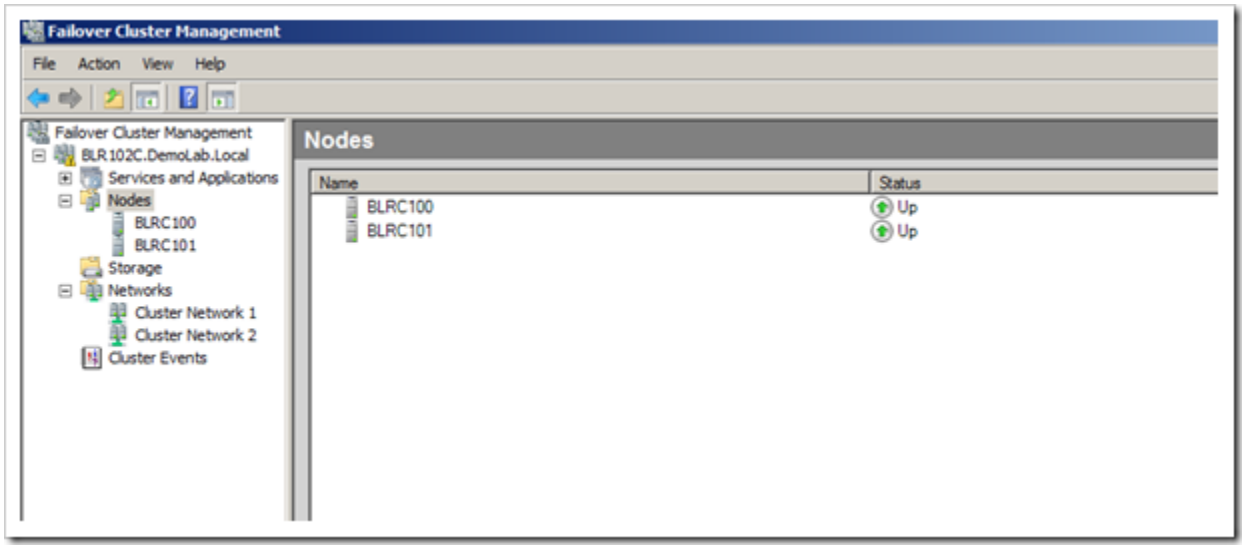
The node is getting added to the cluster BLR102C.



Once the node is added, you will get the following screen. Click on *Finish* to complete the wizard.



If you expand the *Nodes* in Failover Cluster Management, you will find both the nodes, BLRC100 & BLRC101.



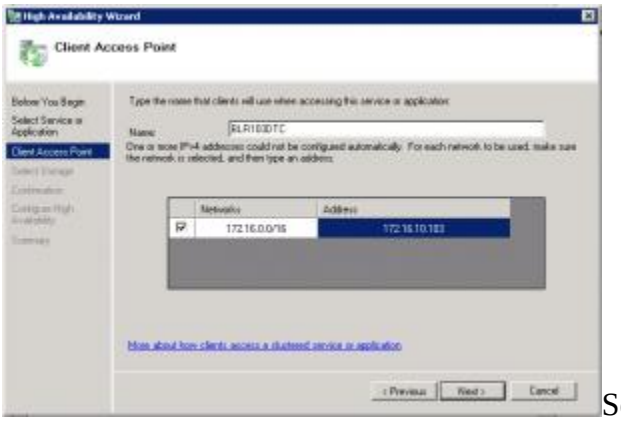
This completes the installation of the Windows Cluster using Windows Server 2008. In the next part of this series, we are going to start the installation of SQL Server in the clustered environment.

Building a SQL Server 2008 R2 Cluster: Part VI

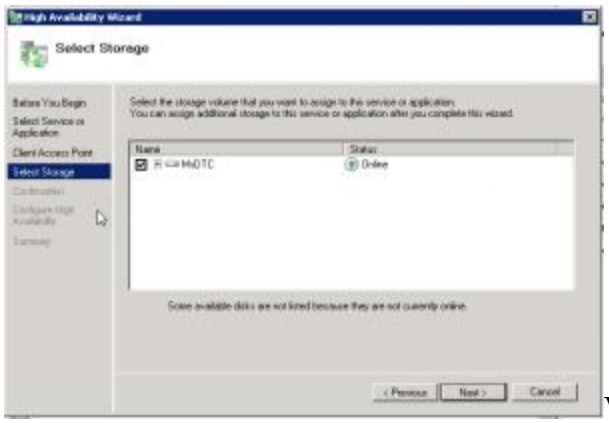
With Part-V of this series on Failover Cluster installation, the two-nodes windows 2008 cluster is up and running. Before continuing further, there is a small update on the installation prerequisites of SQL Server 2008 R2. The following three components are need to be installed on the server before starting the installation (some of these components can be added during installation; however to speed-up the time required to complete the installation, install these components before starting the failover cluster installation). The required components are **Dot Net Framework 3.5 SP1**, **Windows Installer 4.5**, and **PowerShell v 1.0**. As Windows installer 4.5 & PowerShell v 1.0 was already installed on both the node, we need to install Dot Net Framework 3.5 SP1 on both the nodes.

To begin the Failover cluster installation, first we need to create the service accounts of the SQL Server Services (Database Engine Service Account and SQL Agent Service accounts). Create two domain user accounts to run the SQL Server Services, DBC01_AG & DBC01_DE for the SQL Agent Service & SQL Server Database Engine Service Accounts. Make sure you have added the Database Engine Service Accounts to the following policy on the individual nodes, *Lock Pages in Memory* & *Perform Volume Maintenance Task*. Once the Local policies are set, assign the Database Engine Service account with read and write permission on the SQL Server installation drive and on all the drives required for further configuration. The SQL Agent Service account should have minimum read permission on the drive.

Add the MSDTC service on the primary node. Right click on the Services and Application and select *Configure a Service or Application* from the available menu. Click on *Next* and select *Distribution Transaction Coordinator* from the available menu. Provide the *Name* and *IP Address* for the DTC service as shown below and click on *Next* to continue.



Select the drive to install the DTC service, we have added a shared disk for the same and rename the disk resource as *MSDTC*. Select the drive and click on Next to continue.

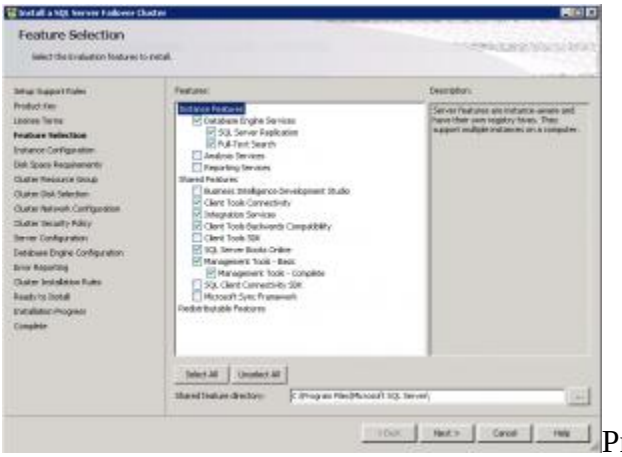


Verify all the available configuration and click on *Next* to start the DTC service installation. Once the installation is completed successfully, we will get the following screen.

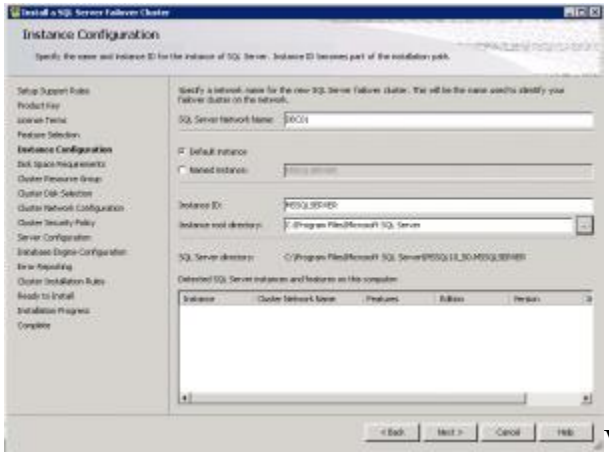
Since we have configured the *MSDTC* service successfully, time to initiate the Failover Cluster installation for SQL Server 2008 R2. To begin with, make sure all the disk resources are online and current present on your first node, in our case, its BLRC100. Start the installation from the SQL Server 2008 R2 installation media, select *New SQL Server Failover Cluster Installation* from the available option, as shown below.

The *SQL Support* files verifies the existing setup for the SQL Server installation and you will be asked to enter the product key for the SQL Server Setup. We are going to use the *SQL Server 2008 R2 Enterprise Evaluation Edition* to complete this cluster setup. *Accept the License Agreement* and continue to load the SQL Server setup

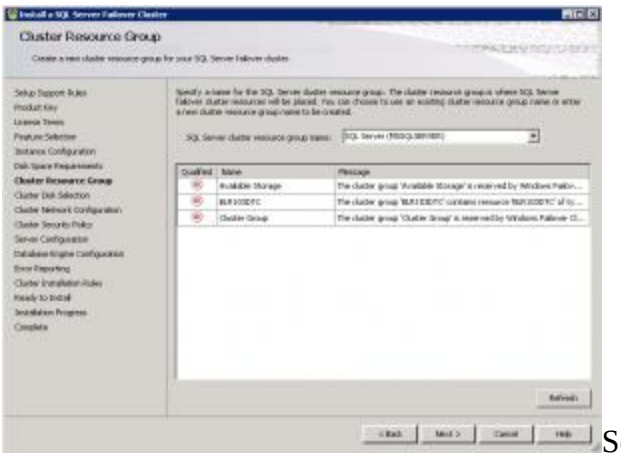
support files. It verifies the existing hardware and software requirements and provide a detailed report about any existing issues. Click on *Next* to select the features of the SQL Server you want to install as shown below:



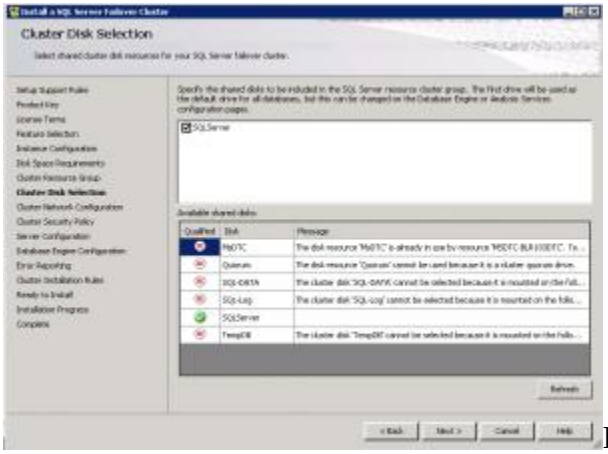
Provide the *Network Name* of the SQL Server and click on *Next* to continue.



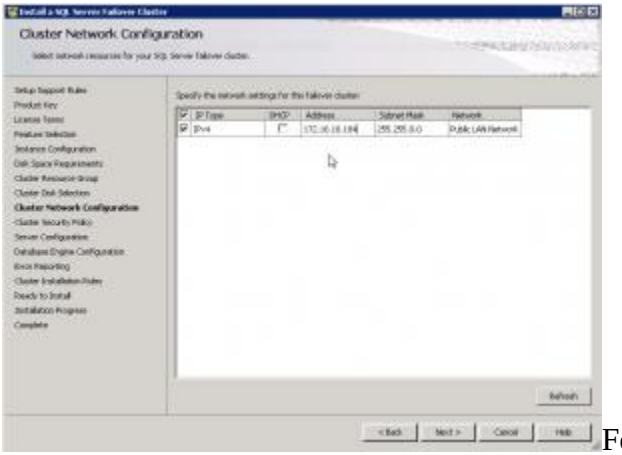
Validate the disk space requirements for the cluster and click on Next to *Create the Resource Group* for the failover cluster installation. For the current setup, we are going to continue with the default configuration.



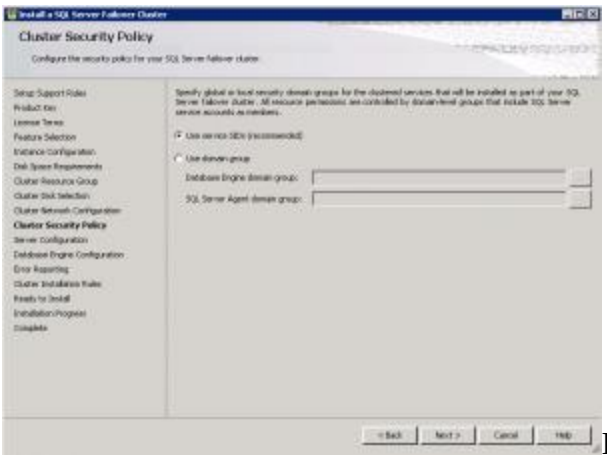
Select the shared disk drive for the failover cluster installation as shown below.



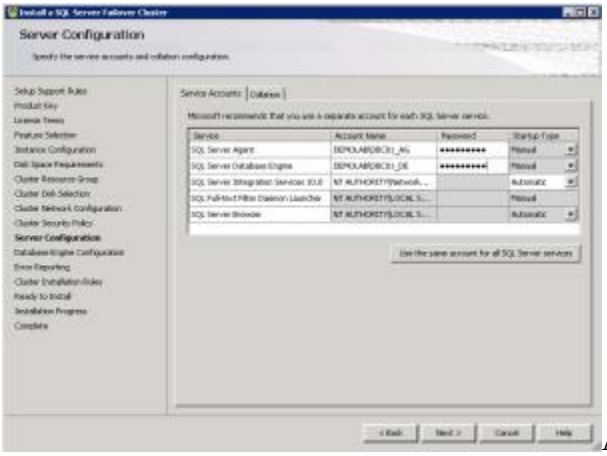
Enter the IP Address for the SQL server cluster and click on *Next* to continue.



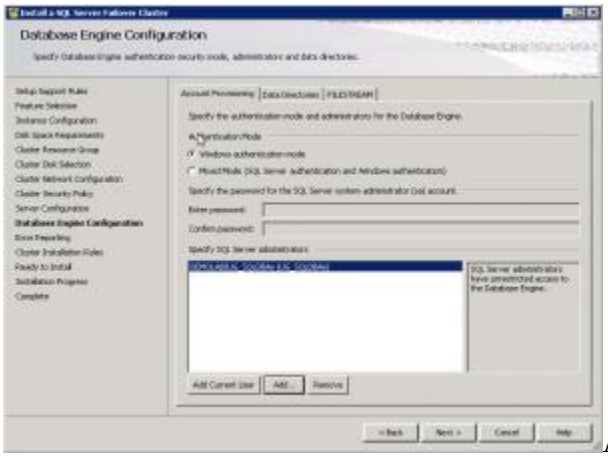
For our test environment, we are going to use *Service SIDs*. Click on *Next* to continue further.



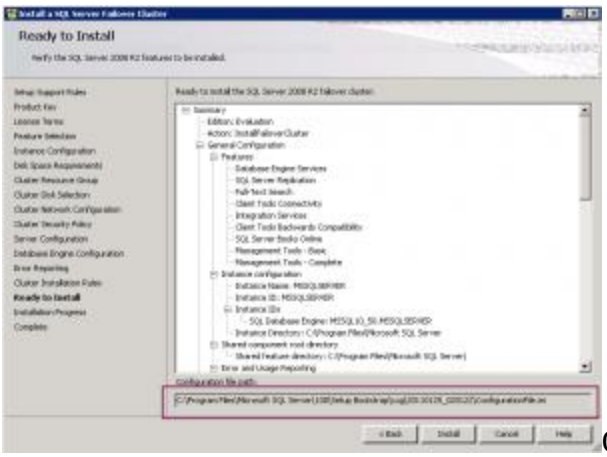
Enter the Service accounts details and password to start the SQL Server Service & SQL Server Agent Service.



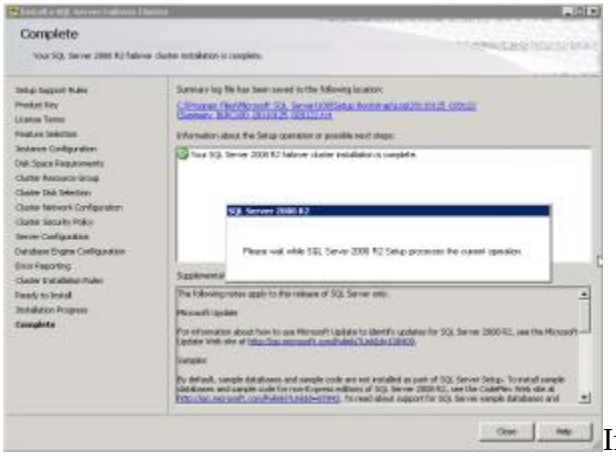
Add the user/group to the SQL Server Administrator; with SQL Server 2008 onwards, the Windows Server Administrators are, by-default not a member of SQL Server Administrators. Make sure, to enable the *Filestream* option for the Database Engine Service Account.



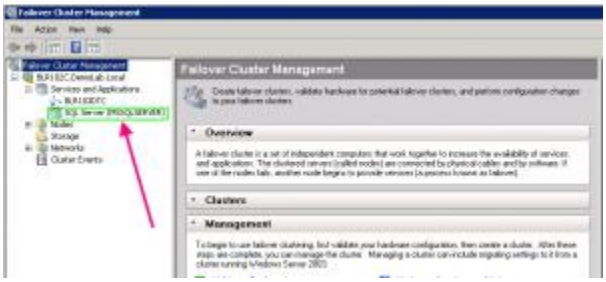
A setup of rules will be executed to verify whether the setup for the Failover Cluster installation will be blocked or not. Verify the SQL Server 2008 R2 features details and click on *Install* to start the SQL Server Failover Cluster installation. The configuration file can be used for script based installation of failover cluster.



Once the SQL Server failover cluster installation completed successfully, we will get the following screen:



If you will open the failover cluster administrator from Administrative Tools, you will see the following screen-shot, when you will connect to the cluster BLR102C.

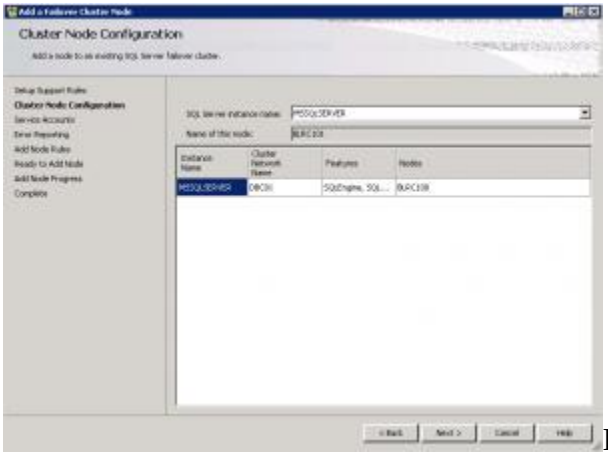


In this part, we have installed the SQL Server 2008 R2 Failover Cluster, in the next part of this series, we are going to add the second node to our Failover SQL Cluster.

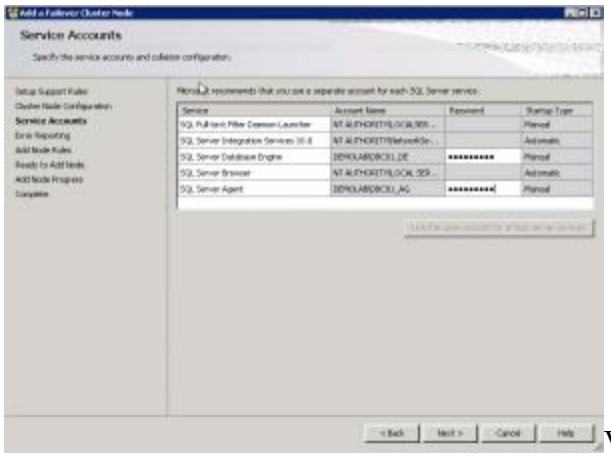
Building a SQL 2008 R2 Cluster: Part VII

In the previous part of this series, we had installed the Failover SQL Cluster. In this post, we are going to add the second node to the existing cluster. To begin with, verify that all the resources are present on the first node. Start the SQL Server setup from the installation media on the second node, and select *Add Node to a SQL Server Failover Cluster.*

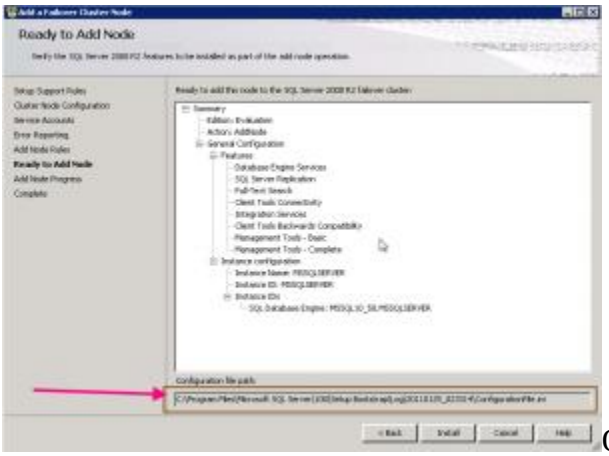
The SQL support files will be installed and the current node is verified for the SQL server installation. Provide the Product-Key and click on *Next* to continue the installation. Select the SQL Server failover cluster instance name, to which, you want to add the current node, and click on *Next* to continue further.



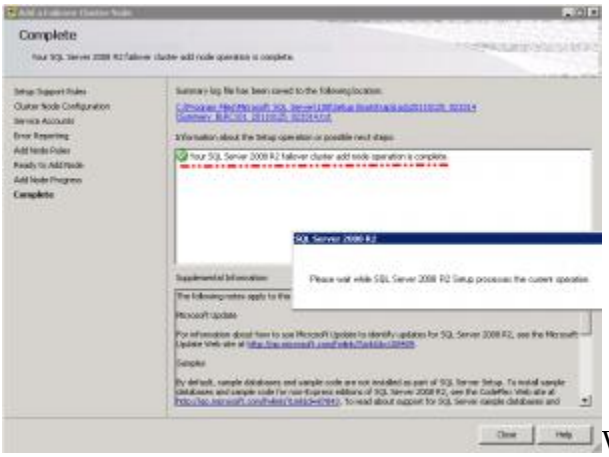
Enter the SQL Server Database Engine & SQL Agent Passwords and click on *Next* to continue further.



Verify all the features need to installed as part of the add-node installation for the Failover SQL Cluster, and click on *Install* to start the installation. The configuration file can be used for the script based installation for add new node to existing SQL Cluster.



Once the setup is completed successfully, we will get the following screen:



With this post, the series on How to install a Windows Cluster and How to deploy a SQL Server 2008 R2 Failover Cluster is completed. However, this posts are mainly targeted to create a setup for testing / learning purpose. For any production deployment, make sure the Windows Cluster validation wizard passed before

proceeding further. Verify Book Online for any additional information. You also need to provide more access for the SQL Server Service Accounts on the drives containing SQL Server files, database files, etc. Make sure the Database Engine Service account has both read and write permissions on the drive, whereas the SQL Agent Service account need read permissions on the drives. Once the cluster is up & running, you might need to configure it as per your requirements, like moving TempDB to a different drive, separate drives for User and System databases, etc.

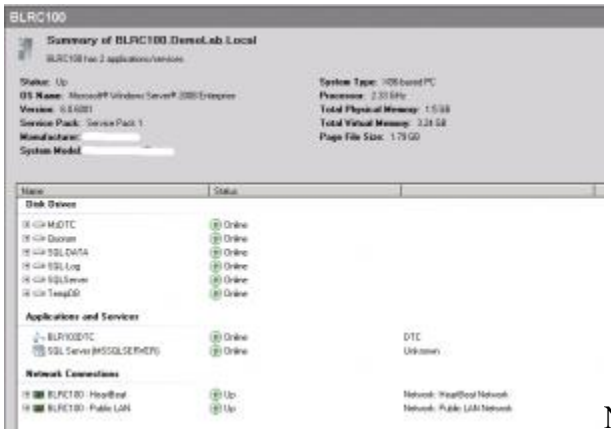
In real world scenario, we normally never shutdown a cluster; however, you may need it in your lab. If you are deployed the entire failover cluster on a Stand-alone machine as me, then you may need to shut down all the VMs before shutting down your Laptop/Desktop. In the last part of this series, we are going to discuss shutdown the entire cluster in your lab, so that it can be used only when required. Stay tuned!

Building a SQL Server 2008 R2 Cluster: Part VIII

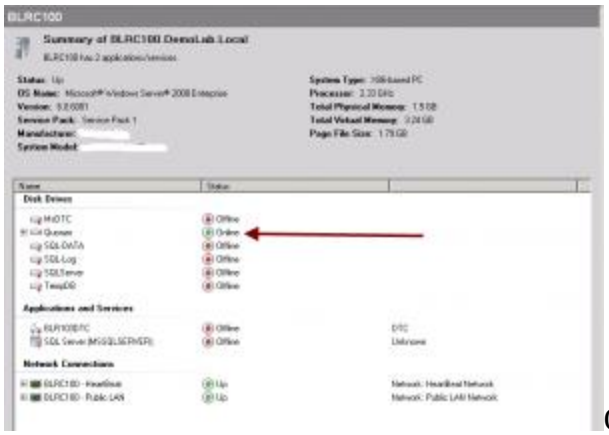
As discussed in the previous part of this series, in a production environment you should never shut down a cluster as it failed the entire purpose of High Availability. However, the current series is focusing on learning purpose. If you are running the setup in your test environment, where you don't want to run all the VMs continuously, you need to cleanly shutdown the entire setup, before shutting down your PC. Some of the following methods are still applicable for update scenarios where you may want to update the cluster node / SQL Server with future Security Patches & Service Packs.

Make sure, all the applications are running from one node. If any application is currently running from the other node, do a failover. Connect to the cluster using Failover Cluster Administrator and the browse to the *Nodes*. Select the Node, where no applications are currently running, right-click on it and select *Pause*. This will pause the selected node, and a failover to that node will not be possible.

Make sure none of the resources are present on that node, which includes the *Quorum* of the cluster. Now log on to the node, and manually shutdown the node. Now all the resources are present on the first node.



Now one node of the cluster is down, it's time to shut down the other node of the cluster. To start with, take the current applications offline. In our case, I first took the SQL Server installation *offline* followed by the *MSDTC*. **Never take the Quorum Disk offline.**



Once all the applications are offline, shutdown the current node. After the node is shut down completely, you can shutdown your storage server, followed by the Domain Controller of your Lab.

Starting the VMs:

While starting your cluster, you need to reverse the entire order of Shutdown the cluster.

1. Start the Domain Controller of your Lab environment.
2. Start the Storage Server, make sure both the DC & the Storage servers are up before proceeding further.
3. Start the node of your Windows Cluster where the quorum & offline applications reside.
4. Once the node is up and running, start the other node of the cluster.
5. Open Failover Cluster Administrator and start the ***MSDTC*** followed by the ***SQLServer***.
6. Make sure the other node is up and running (verify whether all the services with start-up type automatic are started). Once the node is up, *Resume* the cluster node.

Your cluster is back online and available for you.

The complete series is only meant for training & learning purpose. While installing a SQL Cluster in a production environment, you need to make sure that your Hardware matches the Hardware Compatibility List for installing SQL Server Failover Cluster. The cluster validation wizard should pass all the tests and if some of the tests are failed, make sure they are passed before proceeding further.