Realtime publishers

"Leading the Conversation"

The Definitive Guide™ To

Building a Windows Server 2008 Infrastructure

sponsored by

Greg Shields

Copyright Statement

Chapter 10: Windows Failover Clustering

The dream of every IT administrator is an environment of servers and services that never go down. With servers that never go down, the pager never goes off, sleep is never interrupted, and vacations are never put on hold due to data center emergencies. Although the "never" in that dream is likely to remain just a dream for a long time to come, there are technologies available today that can bring it a little closer to realization.

One of those technologies is Windows Server Failover Clustering (WSFC), available with Windows Server 2008. Although WSFC isn't new to Windows, the updates it sees with the upgrade to Microsoft's newest server operating system (OS) makes it a technology that is now eminently useable by a wide range of IT organizations.

A Humorous Personal Perspective on Windows Clustering Over the Years

As a humorous anecdote, Windows clustering was in fact so painfully difficult in previous versions that this author nearly lost a job based on its implementation. Pitching high availability as a solution for file storage back in the days of Windows 2000, an early cluster using that OS version was implemented atop an existing file server. The problem was that clustering in Windows 2000 wasn't all that great and required substantial skill and patience to get it right. In the end, after numerous whole-cluster failures, a lot of painful meetings, and ultimately a de-clustering project, this author's mantra became, "As the corporate expert in Windows clustering, I recommend you don't use Windows clustering."

Thankfully, a lot has changed over the years. With each new OS version, the underlying stability of Windows clustering has improved significantly as well as its management. As you'll find in this chapter, clustering in Windows Server 2008 is a much improved experience than 8 years ago.

The central problem with Windows clustering in previous versions was its significant complexity. The Windows clusters of yesterday were complex to set up and arrived with little assistance to the installing administrator. If you weren't a specialist in Windows clustering, you were likely to have a problematic experience with getting your first few set up. Another issue with some of its earlier versions was clustering's reliance on expensive fibre channel SCSI for its shared data storage. Although fibre channel is an excellent medium for high-speed access to remote LUNs, it can be difficult to work with and requires a set of skills all its own.

Although iSCSI support was first available in Windows Server 2003 SP1, Windows Server 2008 includes expanded support for iSCSI as the medium for shared storage. This version also improves the underlying low-level mechanisms that cluster nodes use to communicate with their shared storage, greatly improving the reliability of the storage subsystem itself.

Understanding Windows Failover Clustering

Clustering, however, is not a solution for all needs. Clustering brings high availability to certain services in certain situations—some of which are not well understood by those that intend to implement clustering. Thus, before we can begin a discussion on implementing WSFC, it is important to understand at a high level exactly what Microsoft's implementation of clustering truly is. With that understanding comes a much-needed discussion on the pros and cons of using clustering in your IT environment.

Clustering in Windows Server 2008 is at its core a mechanism for bringing some forms of high availability to specific Windows services and applications. The last part of that sentence is critical. Most forms of "traditional" clustering using WSFC are not solutions for disaster recovery and clustering does not work with all services and applications.

Any service or application that runs on a Windows cluster must be *cluster-aware*. This means that the service or application has been specifically encoded to recognize that it is running atop a cluster and function appropriately.

Services such as Dynamic Host Configuration Protocol (DHCP) and File Services as well as applications such as Microsoft Exchange and SQL Server are considered cluster-aware applications, and their documentation specifically spells this out. This may not necessarily be the case with other applications. Verifying a service's or application's ability to run atop a cluster is critical before attempting to install it.

The high-availability features that arrive with Windows clustering stem from a cluster's ability to share the potential for connections to data across multiple nodes. In a Windows cluster, between 2 and 16 nodes are configured to share access to one or more storage locations. This idea of "sharing" is actually a misnomer. Only one node at a time can actually interact with a particular storage location. When the node that "owns" the connection to a particular storage location experiences an outage situation, the cluster re-hosts the ownership of the resource to a different node.

Many IT professionals ask about the differences between "active-active" and "active-passive" clustering. At its core, because of this single-node ownership behavior, essentially all clusters and their resources have an "active-passive" configuration. When services or applications leverage an "active-active" configuration, they leverage two separate instances on two separate nodes to achieve this end.

Figure 10.1 shows the simplest example of how the pieces can interconnect. In Figure 10.1, two individual servers are configured as cluster nodes. Each has a connection to the same shared storage and has one or more connections to the Local Area Network (LAN). Typically, a single LUN per resource is created at the shared storage and made available to both cluster nodes. As we'll explore later in this chapter, this LUN sharing has some implications most especially during the initial cluster installation and configuration. Cluster nodes also minimally have one or more connections to the LAN. Although a single network connection can be used for all cluster communication, it is considered a best practice for clusters to use two separate connections at a minimum—one for communication between cluster nodes and a second for communication with the rest of the network.

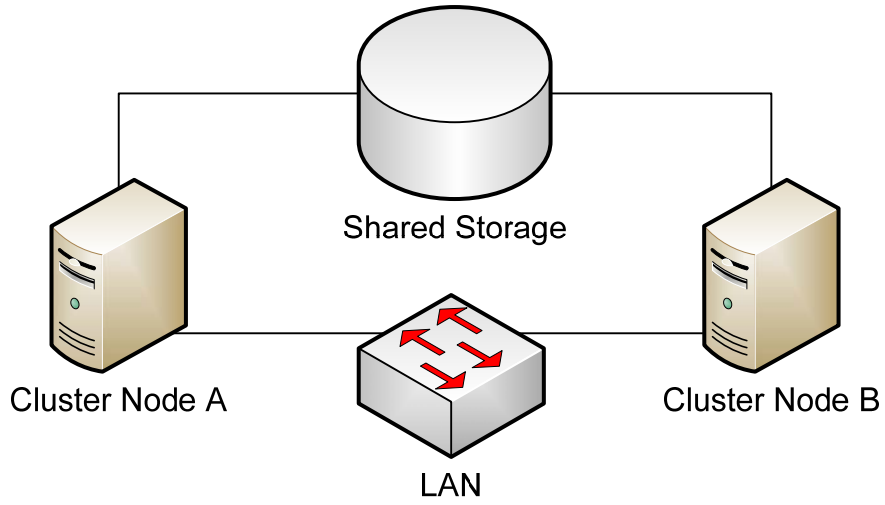

Figure 10.1: A high-level example of the resources used to build a simple two-node cluster.

Two-node clusters work well for the hosting of small numbers of cluster-aware resources. They provide a redundant location for the service to re-host when problems occur on one of the cluster nodes. However, some environments have the resource needs for more than one location. Or they might have multiple resources that they want to load balance across a larger number of hosts. In this case, with the x64 version of Windows Server 2008 Enterprise Edition, it is possible to create clusters with as many as sixteen separate nodes. Figure 10.2 shows an example of how this works with a four-node cluster.

WSFC can be installed only to the Enterprise Edition of Windows Server 2008.

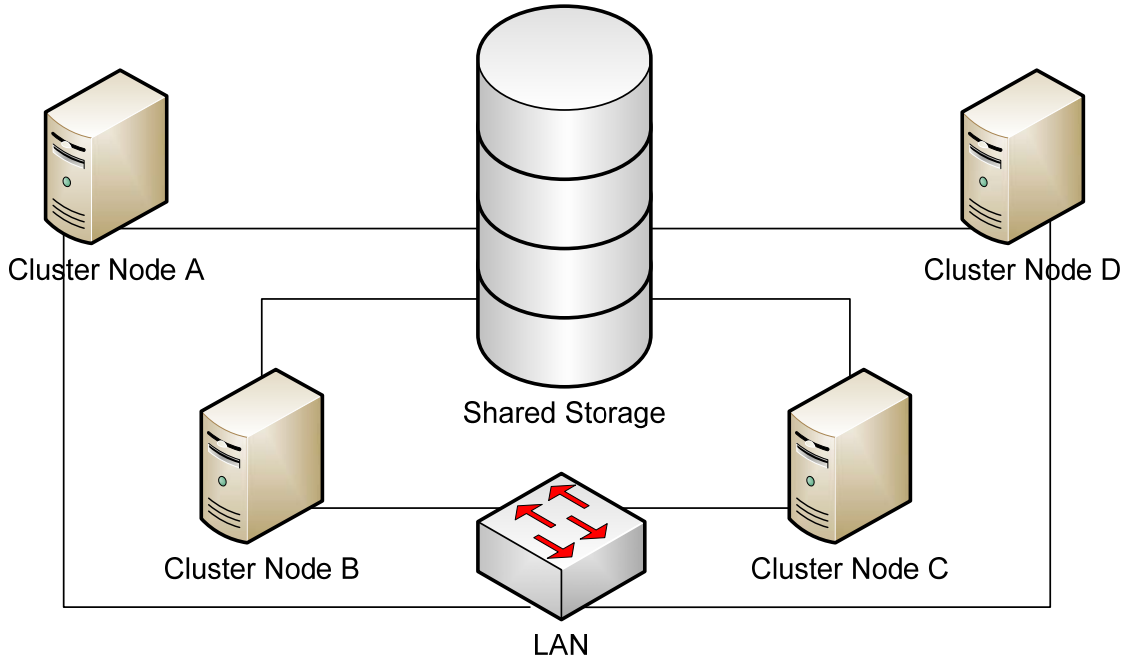

Figure 10.2: It is easy to see how clustering grows complex as more hosts are added.

In this example, four nodes are configured to operate within the cluster. There are four LUNs created, each hosting the storage needs of a particular resource. Each of the four cluster nodes has the ability to host none, some, or all of the configured resources. It is easy to see that as the number of nodes in a cluster increases, so does its complexity. Although building a small two-node cluster can be an easy solution for even small environments, the administration complexities of large clusters can quickly grow unwieldy if not planned appropriately.

Reasons to Use WSFC

Because of the ways in which clusters enable high availability, they are a compelling add-on to existing or new services. There are a few classic instances where clustering specifically adds value to the IT environment:

- *Hardware outages.* Clustering's greatest benefit to the IT environment is arguably its process of seamlessly re-hosting a cluster resource upon the outage of a cluster node. When a cluster node can no longer run its configured service because of a condition on the node—the node is down, it has crashed, it is hung, and so on—the remaining cluster nodes will identify the state and relocate any resources based on preexisting parameters. The benefit to IT is that the process happens rapidly and automatically when a node becomes unresponsive, eliminating the need for an IT technician to immediately troubleshoot and fix the problem to bring the service back to functionality.
- *Software problems.* Similar to hardware problems, software issues sometimes cause software to become unresponsive. Though this is a less-often recognized reason to move to clustering, software problems that can be resolved through an automatic restart can be made easier by hosting atop a cluster. When the software goes unresponsive, the cluster can automatically re-host its instance elsewhere, a process that involves the needed service restart.
- *Patching-related outages.* Another big plus for high-priority services is the way clusters help manage the outages involved with patching services' host server. The patching process for Microsoft products typically occurs minimally once a month and sometimes more often. Re-hosting a cluster resource to a different node prior to patching helps reduce the overall downtime associated with the patching process. It also protects the critical hosted resource from the instance where the installation of a patch causes the server to crash. Having already re-hosted the resource elsewhere gives IT time to fix the problem without seeing a loss of service.

Reasons Not to Use WSFC

For all the items noted previously, it is likely that you will only host your most critical services and applications atop Windows clusters. Due to the added complexity clustering brings to the table, there is the potential that a poorly planned migration of a service to a cluster could decrease its availability. Thus, decisions about the use of Windows clustering must involve careful planning and should be limited to services whose continuous operations are critical.

It is also important to note that most cluster architectures are not intended to be used as a form of disaster recovery. In all but one of the potential cluster configurations, each node must be physically proximate to each other as well as their data storage due to the limitations of fibre channel or Ethernet cabling. Thus, with one exception, individual cluster nodes are generally too close in physical proximity for the loss of one node due to a disaster not to affect other nodes. Additionally, when centralized shared storage is used, the loss of the storage constitutes a loss of cluster functionality. Thus, highly available centralized storage must be used if the storage subsystem is not to become a single point of failure all its own.

With the release of Windows Server 2008, Microsoft has made some very important changes to the way in which clusters can be configured and the way communication takes place between nodes. As we'll discuss later in this chapter, this change brings about the possibility for multi-site, geographically distributed clusters that can be used for disaster recovery.

Components and Prerequisites

Minimally, in order to build a simple two-node cluster, you will require two Windows Server 2008 Enterprise Edition servers to be used as candidate cluster nodes. Each of those computers will need a minimum two network cards each, one for cluster communication with a second for communication with the rest of the LAN. It is possible to aggregate these two network connections—and we will in our example later in this chapter—but it is not recommended for production deployments. A shared storage location must also be available. As stated earlier, that shared storage location can be attached via an iSCSI connection or fibre channel. In the case of iSCSI, one or more additional network cards are necessary to carry the needed iSCSI traffic from the shared storage to the host.

At minimum, on the shared storage, a single volume must be created and assigned a LUN. That LUN should be enabled for connection by all iSCSI network cards on both servers. The individual processes used for creating and exposing that LUN will differ based on the type of shared storage used and its management utility.

> ☞ If you don't have physical iSCSI or fibre channel storage, you can use a software-based iSCSI target as the location for shared storage. That software-based iSCSI target must support the use of SCSI-3 commands and persistent reservations. At the time of this writing, very few software-based iSCSI targets support both of these requirements. One of the few that will work is the StarWind iSCSI Target, available at http://www.rocketdivision.com.

Cluster Validation

Available first in Windows 2003 and significantly improved with Windows Server 2008, the Cluster Validation Wizard is a tool that runs an extended series of tests on candidate cluster nodes as well as the storage and networking components. This tool is useful because any cluster must pass all tests prior to being considered a candidate for clustering. It eliminates many of the manual guess-and-check iterations often required with previous versions and ensures that before any cluster installation begins, all prerequisites are ready for installation.

Figure 10.3 shows an example of some of the tests seen when attempting to run the Cluster Configuration Wizard. When creating a new cluster, once you have completed connecting the physical components, first run this wizard to ensure that you've completed every step correctly. To start the wizard requires that correct network connectivity to each cluster node is available using DNS resolution. It also requires that the Failover Clustering feature is installed to each candidate node prior to running the wizard.

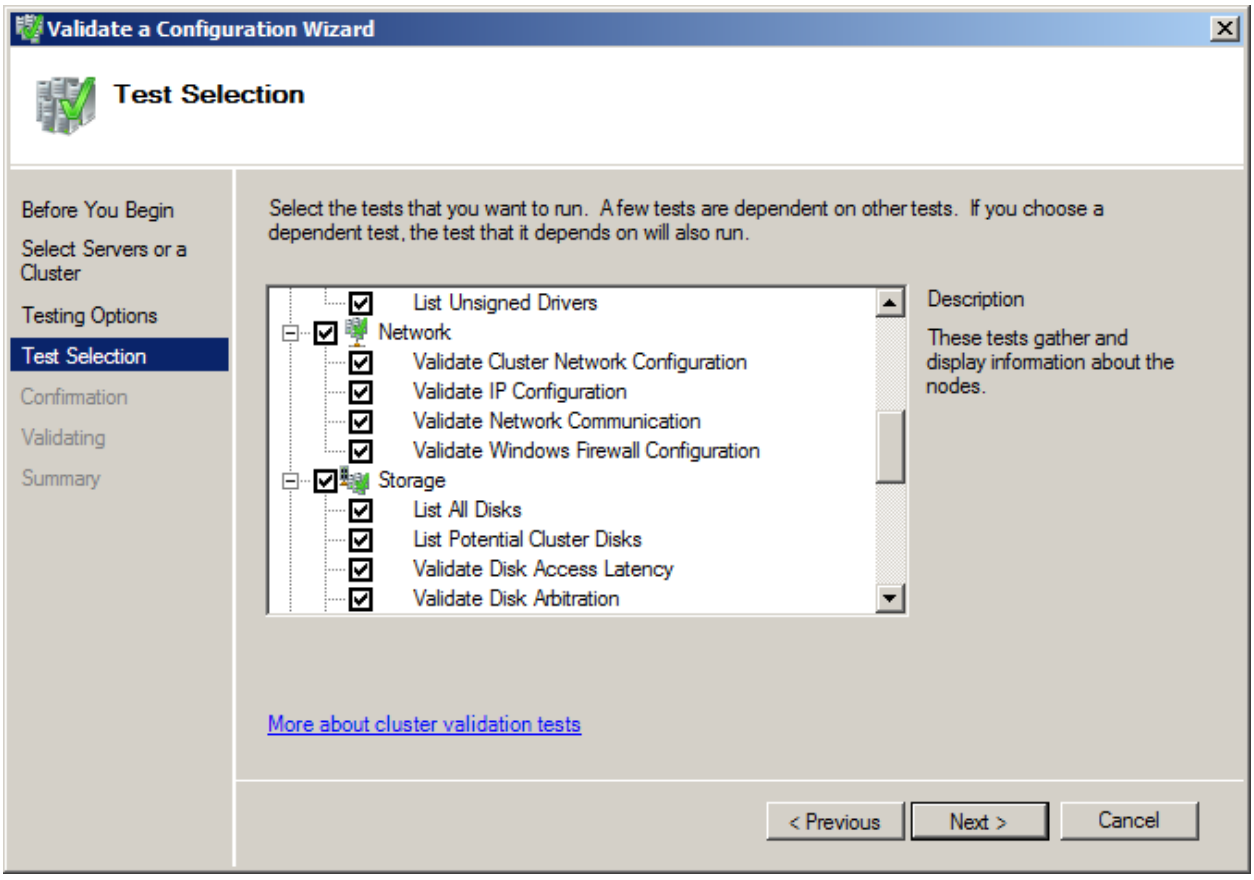

Figure 10.3: The Validate a Configuration Wizard runs an extended series of tests to ensure a successful cluster creation.

Cluster Quorum Models

Another critical part of planning your cluster implementation before any components are installed is the determination of the type of cluster quorum model to be used. Notwithstanding what kinds of resources you plan to host atop your cluster, one of the following quorum models is required.

"Quorum" in WSFC is analogous to democratic voting bodies like the US legislature or you local city council meeting. Quorum in parliamentary procedure is defined as the number of people who must be present at a meeting if that meeting is to be able to hold a vote on issues. Quorum is often defined as 50% plus one of the total members, but can be a different number as defined by the rules of the group. It is put in place to ensure that a minority of the voting body cannot make voting decisions without a large enough group of people present.

With Windows clustering, quorum is used to determine whether the cluster is really a cluster. When a cluster "has quorum" it effectively has enough functioning components in place that it can go about its business being a functioning cluster. When the number of functioning components drops below the threshold for quorum, the cluster can no longer operate as a cluster. All hosted resources then go offline.

The rules of different voting bodies define what constitutes quorum for that body; in the same manner, quorum within various cluster products is defined differently for different platforms. For WSFC in Windows Server 2008, there are actually four quorum models that can be used. Which of these models you'll use depends on the number of nodes in your cluster along with your anticipated uses for that cluster. The following sections offer a description of each of the four possible models.

The quorum model for a particular cluster is usually defined at its time of installation. However, in some cases, it is possible to change the model later within the *Failover Cluster Management* console.

Node Majority

Using the *Node Majority* model, the individual nodes that make up the cluster are given votes that count towards quorum. The cluster considers itself to have quorum when a number greater than half of the nodes are up and available. Should the operational nodes in a cluster using this model go below that magic value, the cluster will cease operations completely. Because of this focus on only the nodes, this model is typically used when the number of nodes in the cluster is odd.

Node and Disk Majority

Two-node clusters, which are the most often-implemented cluster architecture, obviously do not have an odd number of nodes. Thus, a second quorum model is available called the *Node and Disk Majority* model. In this model, each cluster node gets a vote as well as the shared storage for the cluster quorum drive. Thus, this model is typically used when the number of nodes in the cluster is even.

No Majority: Disk Only

Another alternative configuration that can be used but is not suggested is the *No Majority: Disk Only* model. This model is not recommended for use because only the quorum drive itself is given a vote in determining quorum. If the quorum drive is unavailable, the cluster must shut down. This model is effectively the model used in previous versions but is no longer considered a best practice due to the single point of failure that is the cluster quorum drive.

Node and File Share Majority

Lastly, in certain circumstances, it may be desirable to build a cluster whose quorum drive is not attached via shared storage. In the *Node and File Share Majority* model, each node gets a vote in the quorum decision as well as the quorum drive. However, the difference with this model is that the quorum drive is not shared storage. It is instead a file share that is made available somewhere on the LAN to all nodes of the cluster. This model is possible and can be implemented across subnet boundaries with Windows Server 2008 due to its new ability to use TCP-based communication for cluster heartbeat communication rather than broadcasts. The over-the-network nature of this quorum model makes it useful for creating the multi-site, geographically distributed clusters discussed earlier.

For all of these models, you'll notice that a component called the quorum drive is required. This drive is a special drive on shared storage (or via a file share) that is accessible by all cluster nodes. It is usually formatted with 500MB of space and is used exclusively by the cluster to determine quorum. It rarely if ever uses much of that assigned space.

In creating your cluster, you will need to carve out and expose one LUN of this size to all hosts for this use. Although previous versions required this drive to be labeled with a drive letter, Windows Server 2008 does not have this requirement.

Installing WSFC

For this example, we will build a two-node cluster that will host a cluster service atop the servers \\w2008a.realtime-windowsserver.com and \\w2008b.realtime-windowsserver.com. To simplify this example, we will use two iSCSI data stores and only two network cards. The first iSCSI target will serve as the quorum drive and will be configured with 500MB of space. The second iSCSI target will serve as the shared storage for the hosted cluster service and will be configured with 2G of space. Although only two network cards are used in this example for simplicity—one for the iSCSI connection and another for the production network—it is strongly recommended that additional network cards are used in production to separate traffic between that needed for the production network, the cluster heartbeat communication, and its connection to iSCSI. Moreover, because the network-based connection to its iSCSI disk can be a single point of failure with only one network card, redundancy in iSCSI network cards is similarly recommended.

Configuring Networking

It is strongly suggested that the connection to the iSCSI target be made over a different network than that which is used for the production network. An example of the IP configuration of each cluster candidate and the iSCSI target server can be set up as shown graphically in Figure 10.4. Your actual network configuration may differ, but this image shows how the networking is segregated between iSCSI and production networks. Configure the network cards that will connect to the iSCSI target with the proper IP address and subnet mask, but leave the gateway and DNS information blank. Also, remove the *Client for Microsoft Networks* service as well as the *File and Printer Sharing for Microsoft Networks* under the properties of the network card. Lastly, remove IPv6 if it is unused on this network.

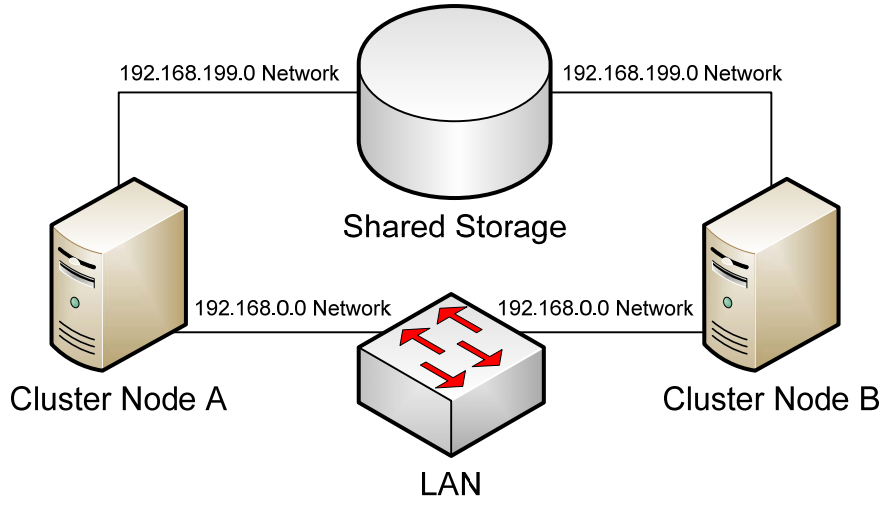

Figure 10.4: The networking configuration of our two-node cluster. This segregation of networks ensures that traffic routes through the correct network cards.

Configuring the Shared Storage

In this example, the two iSCSI targets have already been configured using the iSCSI data store's management utility. A LUN has been exposed to that iSCSI target and made available to each of the hosts. To connect to that iSCSI LUN on the first host, navigate to Start | Administrative Tools | iSCSI Initiator. The first time this tool is run, you will be prompted to start the Microsoft iSCSI Service and set it to Automatic. Click Yes to do so. You will also be prompted to unblock the Microsoft iSCSI Service so that it can operate through the Windows Firewall. Click Yes again to enable this firewall exclusion.

> If your iSCSI target has special software or device drivers required for its use, this software must be installed prior to moving to the next step.

The Microsoft iSCSI Initiator has six configuration tabs:

- *General*. This tab displays the name for the iSCSI initiator and provides a location to change that name as well as configure authentication via CHAP. In our example, we will not be configuring authentication for the sake of simplicity. However, in a production environment, this authentication protects rogue computers from connecting to exposed iSCSI LUNs over the network and its configuration is considered a best practice. The CHAP secret will need to be entered at both the iSCSI target and initiator to connect.
- *Discovery*. Click Add Portal. In the resulting screen, enter the DNS name or IP address for the iSCSI target that hosts the data storage location. Click Advanced. Ensure that the *Local adapter* is set to Microsoft iSCSI Initiator and the Source IP is set to the IP address for the network card you want to configure for use with iSCSI.
- *Targets*. If the connection was correctly made on the previous tab, selecting this tab will automatically display the available drives on the iSCSI target. A picture of how this should look is shown in Figure 10.5. Click each discovered target and then the *Log on* button. In the resulting screen, select the *Automatically restore this connection when the computer starts* check box, and click Advanced. Again, set *Local adapter* to Microsoft iSCSI Initiator, Source IP to the correct source IP for this server's network card, and *Target portal* to the iSCSI target address. If your iSCSI target uses special software that enables multiple connections to the target, you can select the *Enable multi-path* check box. Complete these steps for each discovered drive.

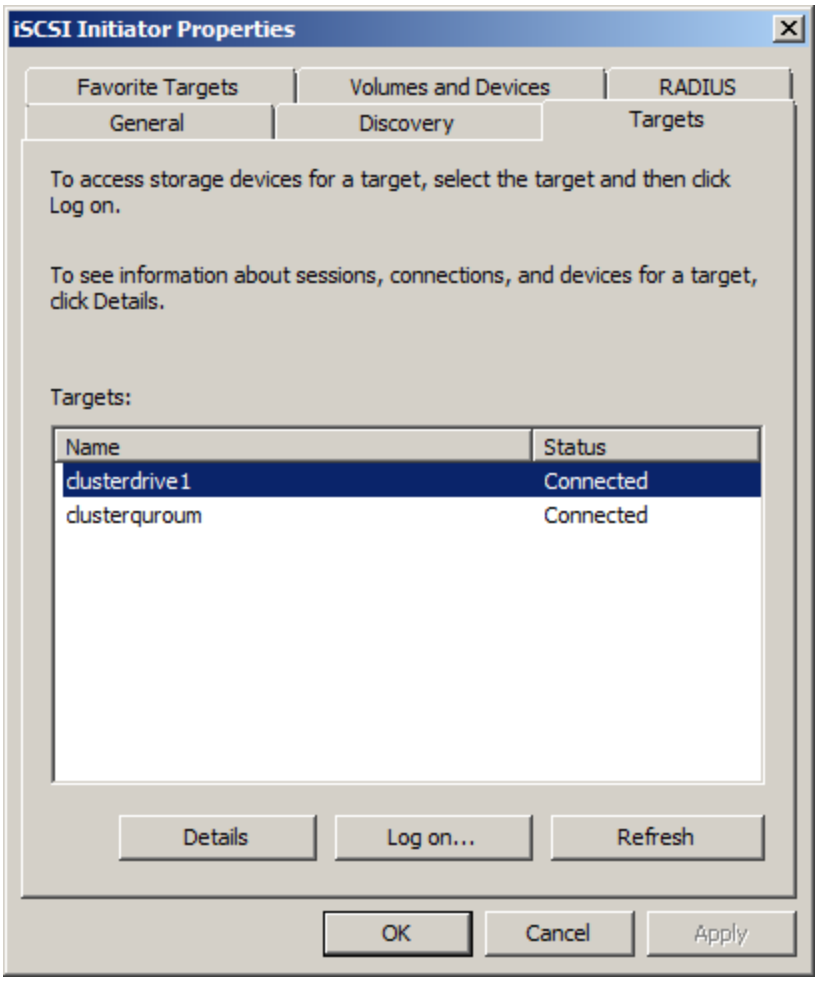

Figure 10.5: If you've configured everything correctly, your iSCSI drives should appear on the Targets tab.

- *Favorite Targets*. On this tab, you can view the properties of any connected drives. There is no further configuration to be done on this tab.
- *Volumes and Devices*. Click Autoconfigure. If everything has been set up correctly to this point, the *Volume/mount point/device* box should populate with information about the discovered drives.
- *RADIUS*. For the purposes of this example, there is nothing to do on this tab.

You will want to complete the previous steps on both candidate hosts to establish each server's connection to the iSCSI target. Note that at this point you have made a connection to a raw drive, but you have not yet initialized or formatted the drive, nor have you added a drive signature to that drive.

To do so, launch Server Manager and navigate to Storage | Disk Management on one of the two nodes. The two drives should be present on the node, but both will be displayed in black as Unallocated disks. Right-click each disk and select to bring that disk Online. Then right-click one of the disks and choose Initialize Disk. The Initialize Disk wizard will appear with both disks selected. If this disk will never grow beyond 2TB, keep the disk as a Master Boot Record (MBR) disk. If you believe the disk will grow beyond that size at some point in the future, convert the disk to a GUID Partition Table (GPT) disk. Lastly, right-click each disk and select New Simple Volume. Create a new simple volume on each disk, assign a drive letter, and format the disk as NTFS.

Validate and Create the Cluster

By bringing the disk online and formatting it, the disk can be verified by the Cluster Validation Wizard, allowing that process to complete its testing. At this point, navigate to Administrative Tools | Failover Cluster Management. There, right-click the top-level node and choose to Validate a Configuration. This will launch the Validate a Configuration Wizard, which will prompt for the names of the candidate nodes and the tests to be run. Once run, the wizard will provide an HTML report of the results similar to what is seen in Figure 10.6. If any errors appear in the running of the wizard, you will need to fix the problem and re-run the wizard until all tests are passed.

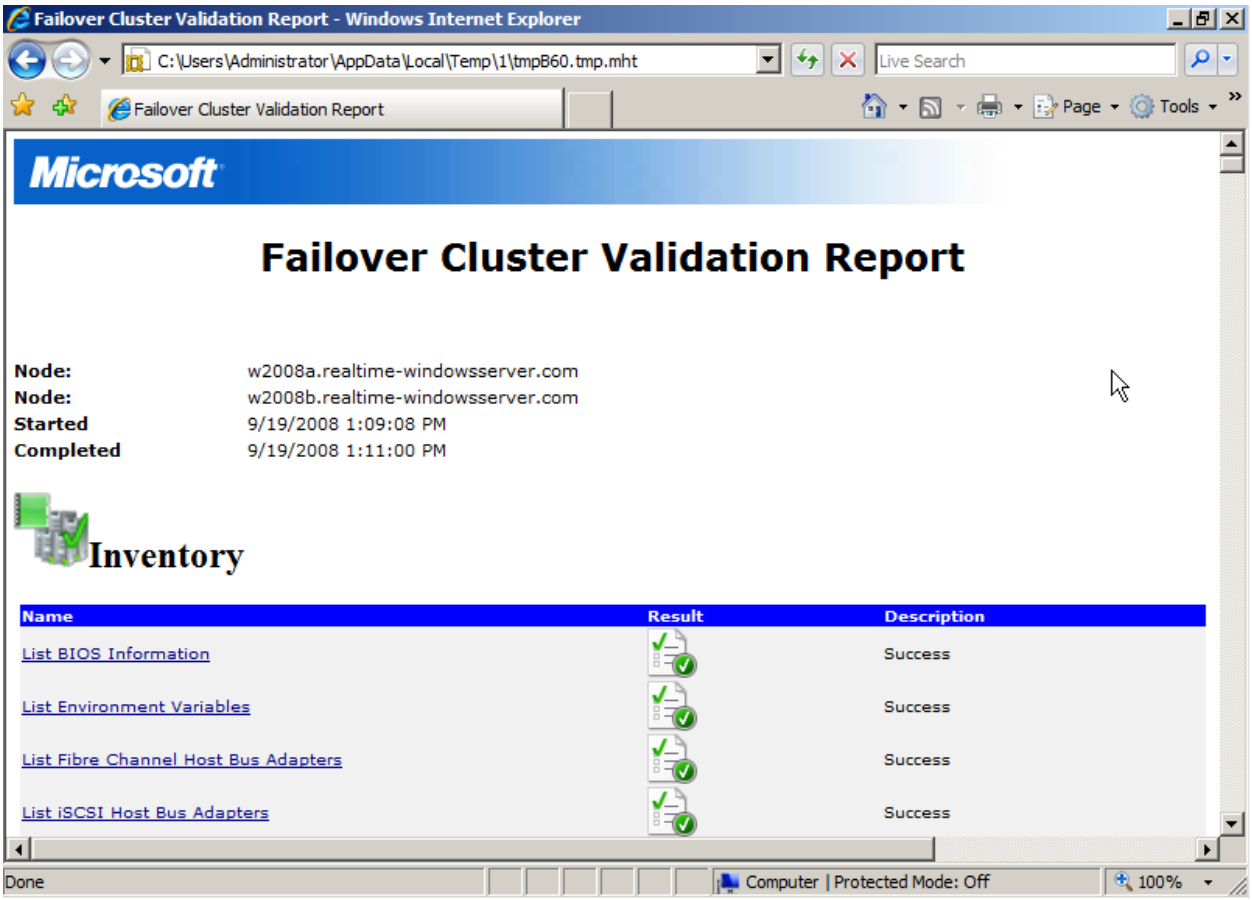

Figure 10.6: An example of the Cluster Validation Report. All cluster components must pass all tests prior to attempting to create a cluster.

Once you've completed the wizard and fixed any issues discovered in the validation process, it is time to create your cluster. Do so back in the Failover Cluster Management console by right-clicking the top-level node and choosing Create a Cluster. In the resulting wizard, enter the names of the candidate nodes. In the next screen, provide a name as well as an IP address for the cluster itself. This name and address will be used for connecting to the cluster for management. Finally, confirm the creation of the cluster. The wizard will create the cluster and return control when that process is complete.

Ensure that you click View Report after the completion of the installation to view the results of the installation process. Some clusters can be installed with warnings that later cause problems.

Post-Installation Quorum Reconfiguration

Once the cluster has completed its installation, it can be managed via the Failover Cluster Management console. Immediately after creation, navigate to this console and verify that all network and storage resources are available and visible in the interface.

The cluster installation wizard by default does not always install the quorum resource to the correct shared disk and sometimes does not always choose the correct quorum model. If either of these conditions is the case, these settings can be changed by right-clicking the cluster name and selecting More Actions | Configure Cluster Quorum Settings. In the resulting wizard, it is possible to change the quorum model as well as the shared drive that is to be used for the quorum. Figure 10.7 shows an example of the screen where the quorum drive can be changed. Click through the wizard to complete the reconfiguration.

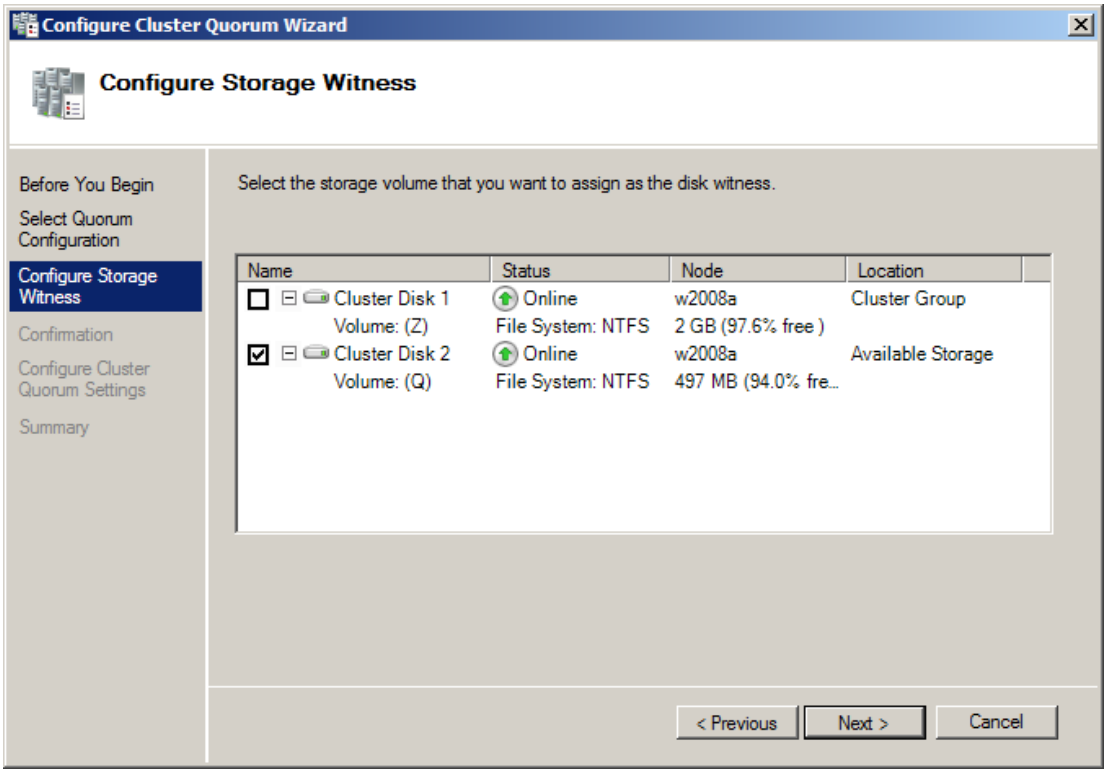

Figure 10.7: It is possible to adjust the quorum drive and model after the cluster completes its installation.

Managing WSFC

Once complete with the steps up to this point, you have successfully created your two-node Windows cluster. The creation of clusters with larger numbers of nodes happens in much the same way but with more planning of IP addresses and storage as the number of nodes increases. This completed cluster at this point, however, isn't serving any purpose. There are no services or applications running atop the cluster. In this section, we'll add a service and talk a little about the process of adding an application. What you'll find is that once the cluster is created, you're only partially ready for operations. The management of cluster resources and services requires additional care to ensure that they function in appropriate and desirable ways.

Adding a Cluster Service

By default, a standard Windows Server 2008 cluster can support 13 types of services right out of the box. Each of these services can be added to the cluster by right-clicking the Services and Applications node and selecting Configure a Service or Application. The possible services that can be added directly through the interface are:

DFS Namespace Server	DHCP Server	Distributed Transaction Coordinator
File Server	Generic Application	Generic Script
Generic Service	iSNS Server	Message Queuing
Other Server	Print Server	Virtual Machine
WINS Server		

You'll immediately notice that a number of these potential services are intended for generic instances of applications, services, or servers. These "generic" entries are used in situations where the service you want to host atop your cluster does not have its own native cluster installation routine. Prior to attempting to run an existing application as a "generic" cluster service, consult the documentation for that service to determine whether the service supports the configuration.

Some services require the accompanying role to be installed via Server Manager prior to enabling it for cluster hosting.

For the continuing example of this chapter, we will create a clustered file server by adding the File Server service, as shown in Figure 10.8. Click that link in the list to start the process of adding the clustered service. Creating a file server service first requires the creation of a network name and IP address that users will use to connect to the service. This network name and IP address are not unlike naming and addressing the server that would have hosted file services in the traditional sense. Neither name nor IP address can be the same as the existing cluster name or IP address.

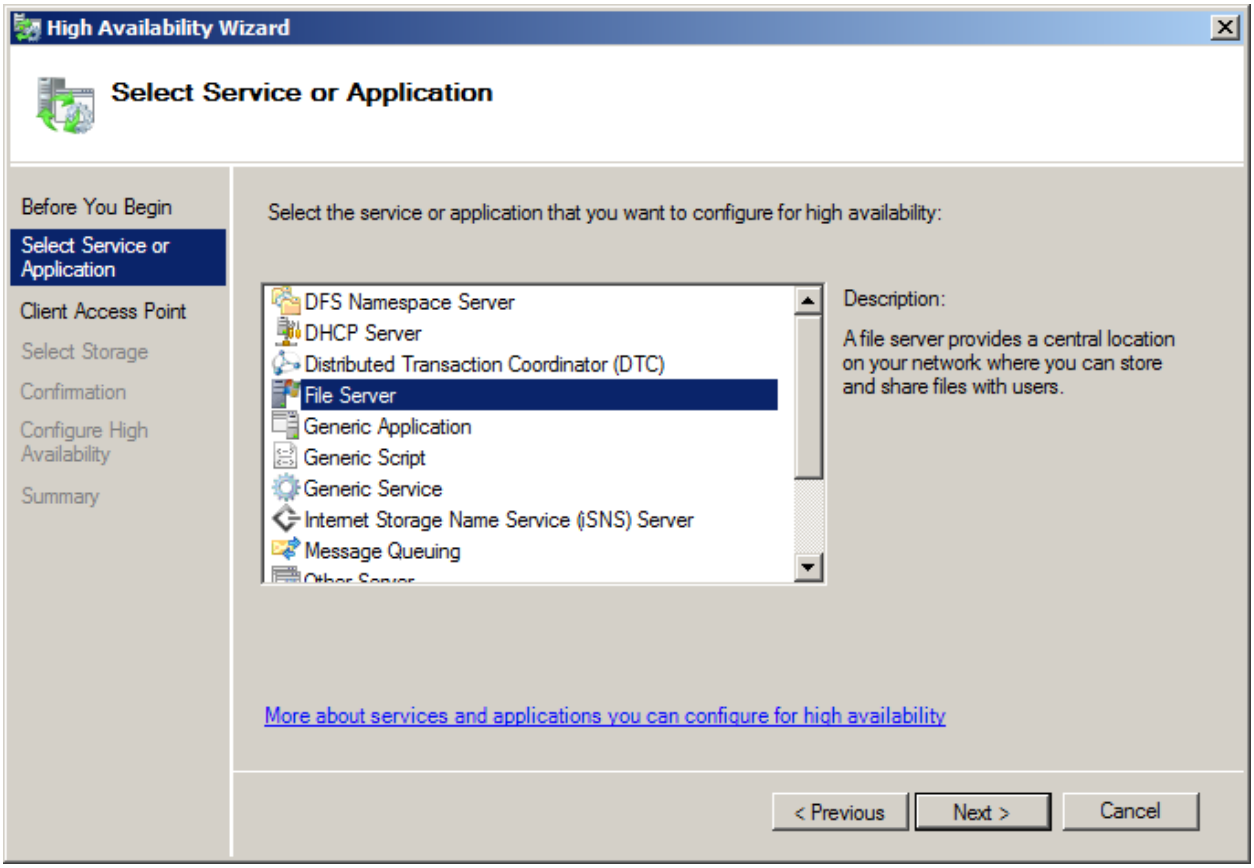

Figure 10.8: The list of possible cluster services available in the interface.

> You'll find that clusters, especially those that host multiple services, tend to consume large numbers of IP addresses. Plan accordingly.

Once a name has been given to the new service, you will need to assign it the available shared storage disk created earlier. This shared storage will be the location where files are stored by users once the service is fully configured. As you'll see, the shared storage for resources such as file services can grow exceptionally large. Thus, there is the need to ensure enough storage is available for long-term storage needs at the time the cluster is created.

Once the service has been created, clicking that service in the console brings forward status information about the service. This view is shown in Figure 10.9. There, you'll see that the file server service *REALCLUSTFS* has been created. Also shown is that the service is online and currently owned by the server \\w2008b. The service is using Cluster Disk 1 and has a single hidden share currently created. Creating a new clustered share is done by right-clicking the service, and selecting *Add a shared folder*. Doing so brings forward the same Provision a Shared Folder wizard as discussed back in Chapter 4.

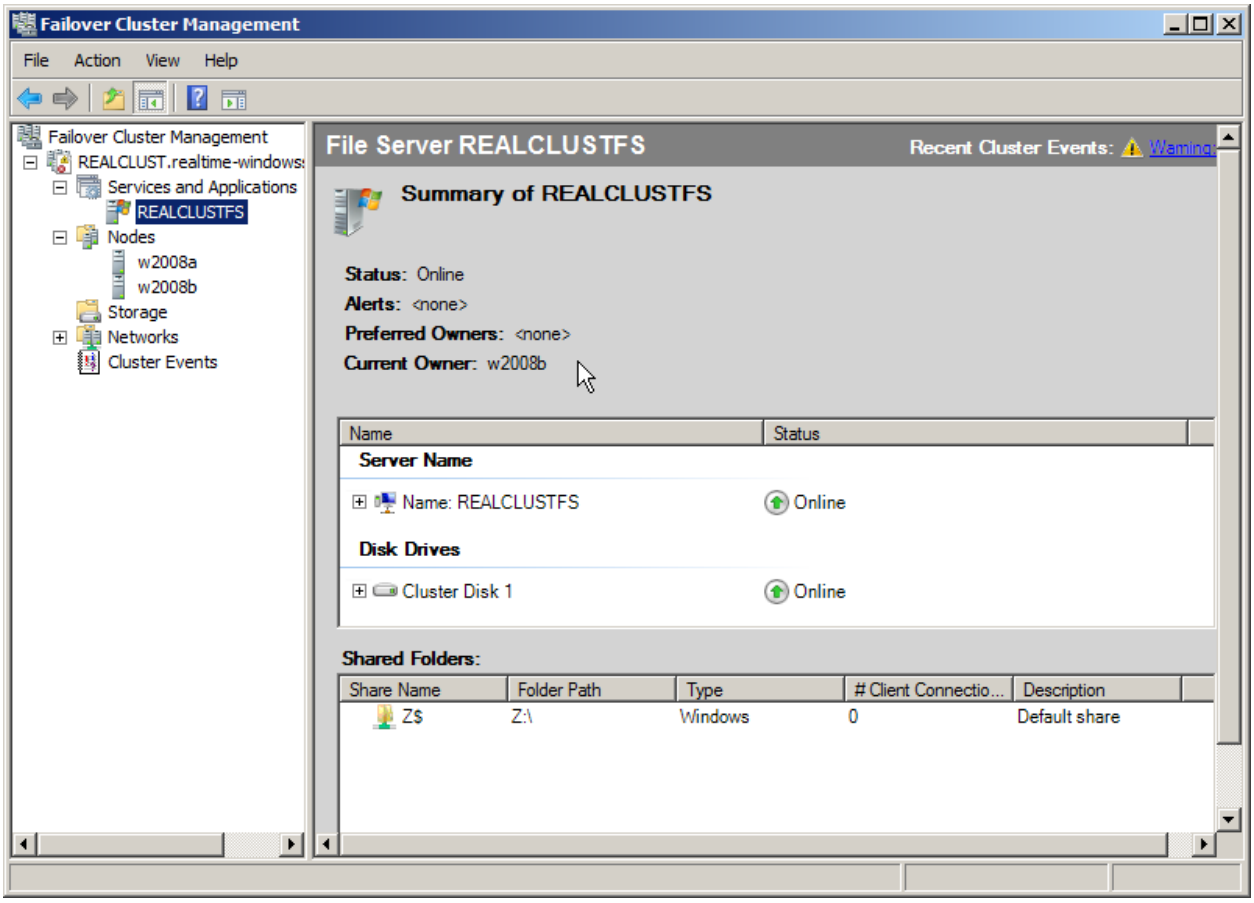

Figure 10.9: Viewing the properties of the newly created file server service.

Installing Cluster-Aware Applications

Applications are installed usually in a much different way than cluster services. As described earlier, it is possible to create a generic application service in much the same way as creating the file server service done in the previous section. However, it is worth mentioning again that when using the "generic" entities for creating new cluster applications, it is critical to verify first with the application's vendor that the application will indeed function in a clustered environment.

Some applications such as Microsoft Exchange Server and Microsoft SQL Server have installations that are cluster-aware themselves. Thus, the process to install these applications to an existing cluster is not the process discussed previously. Instead, to install these applications, run their standard installation setup file. That setup will automatically recognize that the application is being installed to a cluster and provide the necessary installation questions needed to complete the installation.

Managing Resources and Dependencies

The internal logic used by a cluster in determining whether the resource is healthy or needs to be relocated to another node is handled through a series of dependencies. Resources that make up a cluster service will have a list of dependent and antecedent resources that map together to equal the sum total of the service. This list of dependencies can be seen by right-clicking the service and choosing *Show dependency report*. Figure 10.10 shows a snippet of that report for the newly created file server service, which displays how the Network Name resource depends on the IP address resource while the Physical Disk resource stands alone.

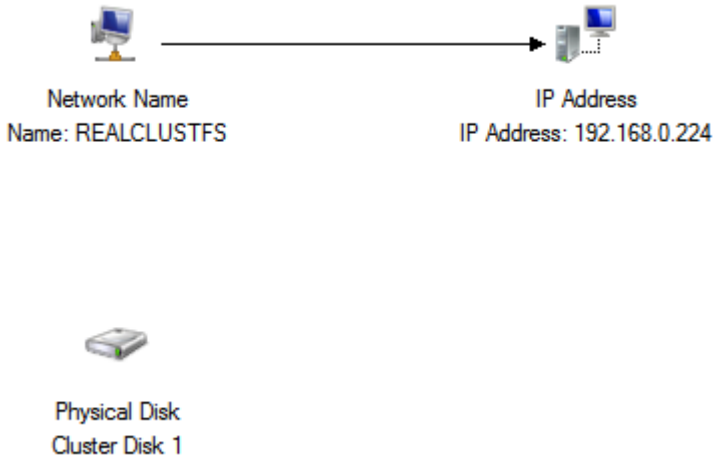

Figure 10.10: A dependency mapping of the file server service showing the three resources that make up the service.

For resources that have dependencies, any or all of their dependencies must be online for the resource to remain online. If a dependent resource goes offline, the resource itself will go offline. As we'll discuss in the next section, this can trigger a failover event. In the right pane of the screen shown in Figure 10.9, click any of the resources that make up the REALCLUSTFS service. On the General tab of the resulting screen, you will see information about that resource. Selecting the Dependencies tab will display the list of dependencies for the resource. Here, it is possible to manually add dependencies if your service architecture needs them. For example, if the outage of a completely separate IP address or service will impact the functionality of the resource, you would enter that other resource as a dependency here. Figure 10.11 shows how the Cluster Disk 1 could be added as a dependency to the REALCLUSTS service using the AND operator with the IP address.

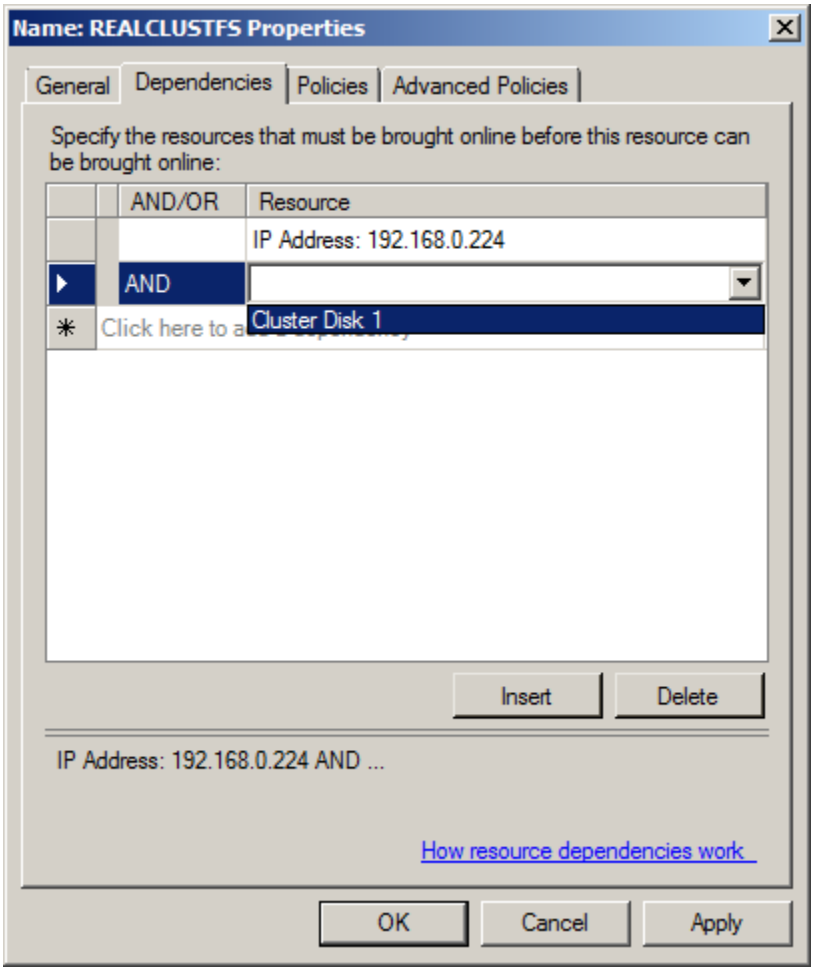

Figure 10.11: Adding dependencies to the file server service can trigger a failover if one of the dependencies fails.

Be cautious with the addition of dependencies. The outage of any dependency can trigger a failover.

Failover

All this talk of dependencies directly drives the behavior of the cluster should an outage of a resource occur. When the outage of a resource takes place, the default behavior of a cluster is to attempt to restart the resource on the current node. If that cannot be accomplished, the cluster will attempt to fail over the entire service to the alternative node. This determination of failover behavior is configured by right-clicking to view the properties of the cluster service, and selecting the Policies tab. This tab is shown on the left side of Figure 10.12.

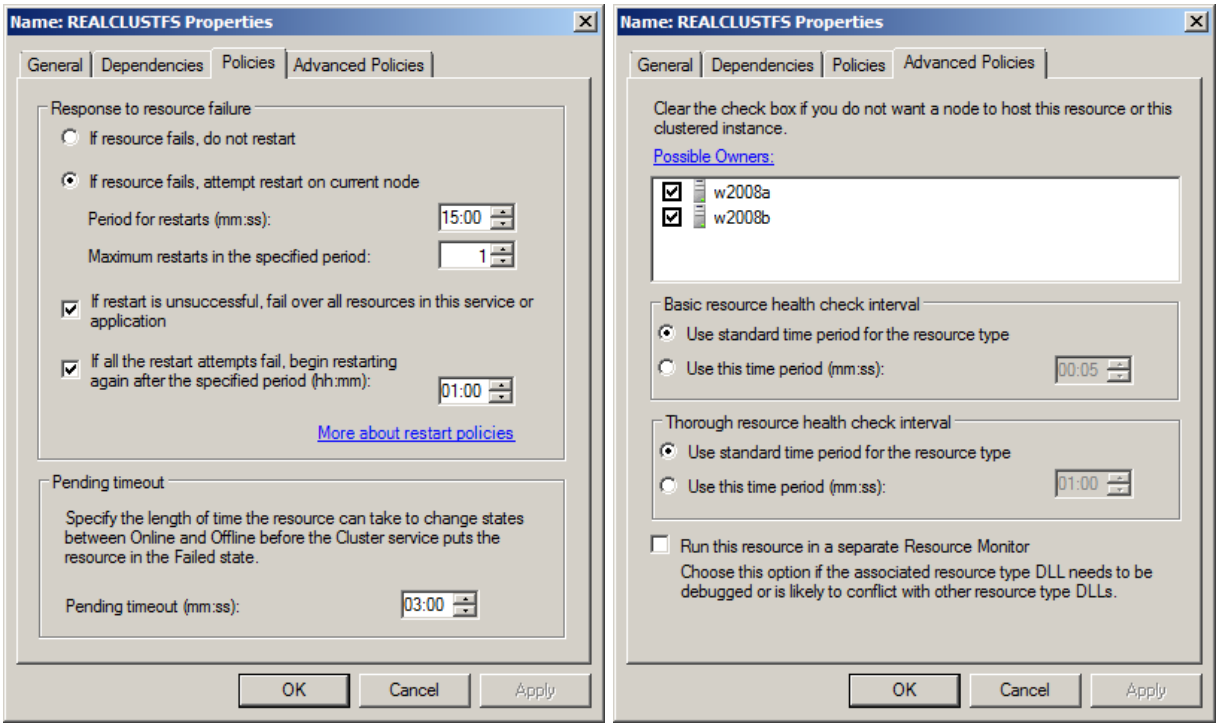

Figure 10.12: The Policies and Advanced Policies tabs of the file server service. It is within these two locations where failover behaviors for the resource are configured.

You can see on the left side of Figure 10.12 that the resource will first attempt to restart itself on the current node before failing over to the other node. This is handy in the case where a resource sees a temporary outage due to environment conditions and you don't want the service to change node ownership. The right-pane of Figure 10.12 shows the advanced policies associated with the resource. These advanced policies come into play much more in clusters that include more than two nodes. Here, it is possible to identify which cluster nodes are allowed to own the resource in the case of a failover. When a failover occurs on a multi-node cluster, the entries in this list are used to identify which node can and ultimately will become the new owner of the resource during a failover event. Also available here are options for choosing the intervals used for verifying the health of individual resources.

Complicating this configuration further, right-clicking and viewing properties on the service itself brings forward another two possible tabs that are shown in Figure 10.13. Here, it is possible to identify which cluster hosts are referred to own the resource. Although also more useful in multi-node clusters, the list of preferred owners shown on the left side of Figure 10.13 provides a place to identify which owners are preferred to own the resource. The most preferred owner is at the top of the list.

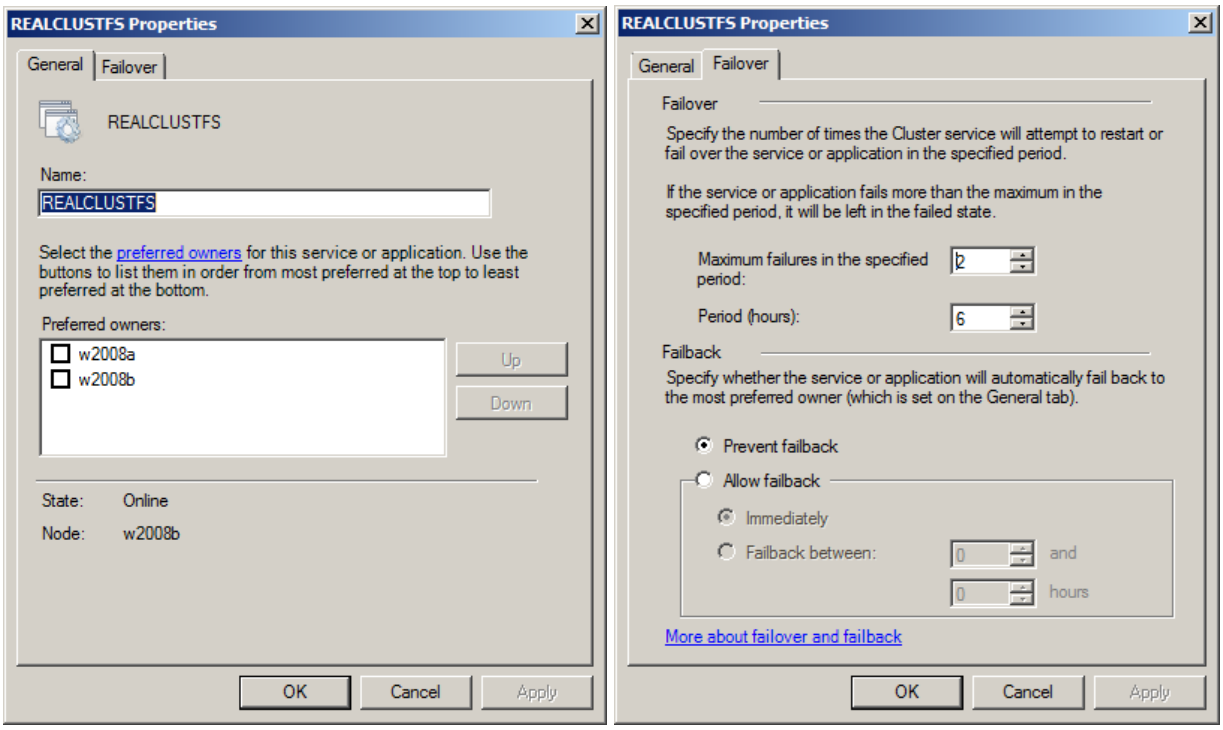

Figure 10.13: Two tabs for setting failover behavior and node ownership relating to the service itself.

On the Failover tab, it is possible to customize the behavior for what defines "failure" for the service itself. Here, it is possible to customize the number of times a failure can occur over what period before the cluster considers the resource completely failed. In the case where the cluster completely fails a resource, the cluster no longer attempts to restart the resource. This failover protection is present to prevent a condition commonly known as *bouncing* where a cluster resource that cannot be brought online continually fails over between nodes.

Failback

Failback is another configuration that can optionally occur when you want a failed resource to return back to its preferred node when that node can resume servicing the resource. Failback is by default disabled. This is the case due to the potential for the same kinds of bouncing behavior discussed earlier. When a node fails on its preferred owner, it fails over to an alternative node. If the resource successfully restarts on the alternative node, failback will either immediately or eventually fail back the resource to its preferred node.

Although this may sound like a desirable feature, be careful with the use of this configuration. In the case where the resource cannot start on its preferred node but restarts correctly on an alternative node, failback can actually cause the resource to return back to the host where it regularly fails. Once failed back, the service cannot restart, which causes another failover, ultimately resulting in a bounce condition until the cluster completely fails the resource.

Avoid failback unless you absolutely need it. Obviously, this means that some manual monitoring of cluster resources and their ownership is required. But that manual monitoring is arguably a much better solution than creating the potential for a painful bounce condition.

Geoclustering

The decision to implement a geographically distributed cluster is not one taken lightly. Whereas the process to create a simple two-node cluster is relatively trivial, once the networking and storage concerns are planned, implementing a geocluster or "geographically distributed cluster" requires a lot more work and cost. Unlike traditional clusters, geoclusters leverage the use of separated but replicated data storage at each site where a cluster node is hosted. An example of this is shown graphically in Figure 10.14.

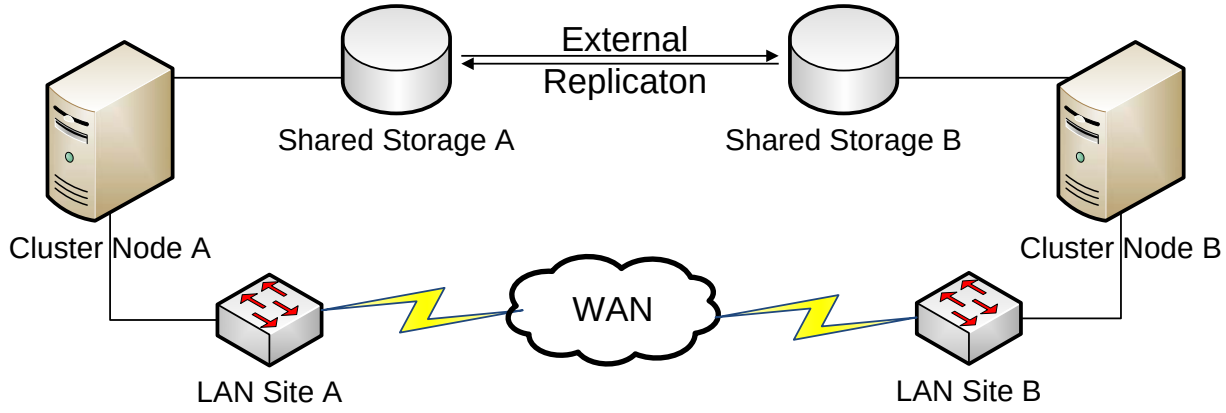

Figure 10.14: A geocluster across multiple sites is possible with Windows Server 2008. However, Microsoft does not provide the needed replication tools to ensure the separate-but-equal data stores remain consistent.

First and foremost, Microsoft does not provide the replication tools necessary to replicate the storage subsystem between the two separated nodes. This replication is necessary to ensure that both nodes see the same set of data. Third-party replication tools are necessary that guarantee a very low latency to ensure high levels of data consistency between nodes. Microsoft has augmented WSFC in Windows Server 2008 with a change to the cluster heartbeat that loosens the restrictions on network latency for the cluster heartbeat as well as a conversion of heartbeat communication to routable TCP.

> 🕮 For more information about the third-party tools that enable replication between sites as well as detailed information on developing multi-site failover clusters, check out http://www.microsoft.com/windowsserver2008/en/us/clustering-multisite.aspx.

Clustering Brings High Availability

So clustering in Windows Server 2008 finally brings low-cost high availability to specific windows services and applications. With the right iSCSI or fibre channel data storage in place, the only additional needs are the right network connections and a plan for failover of resources. Unlike essentially every previous version of Windows clustering, that which you'll find in Windows Server 2008 is a service you'll actually want to use for your highest-value services.

And thus ends our guide. This guide in all its ten chapters has attempted to assist you with the process of building your Windows Server 2008 infrastructure. Starting with the concepts necessary to build the domain, working through various services such as file serving and Terminal Services, and including a few management components such as Group Policy along the way, this guide has hopefully given you the resources you need to build your infrastructure along the lines of best practice. The rest is up to you.

Download Additional eBooks from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this eBook to be informative, we encourage you to download more of our industry-leading technology eBooks and video guides at Realtime Nexus. Please visit http://nexus.realtimepublishers.com.