Chapter 1

Configure and Manage High Availability

THE FOLLOWING 70-412 EXAM OBJECTIVES ARE COVERED IN THIS CHAPTER:

✓ **Configure Network Load Balancing (NLB)**

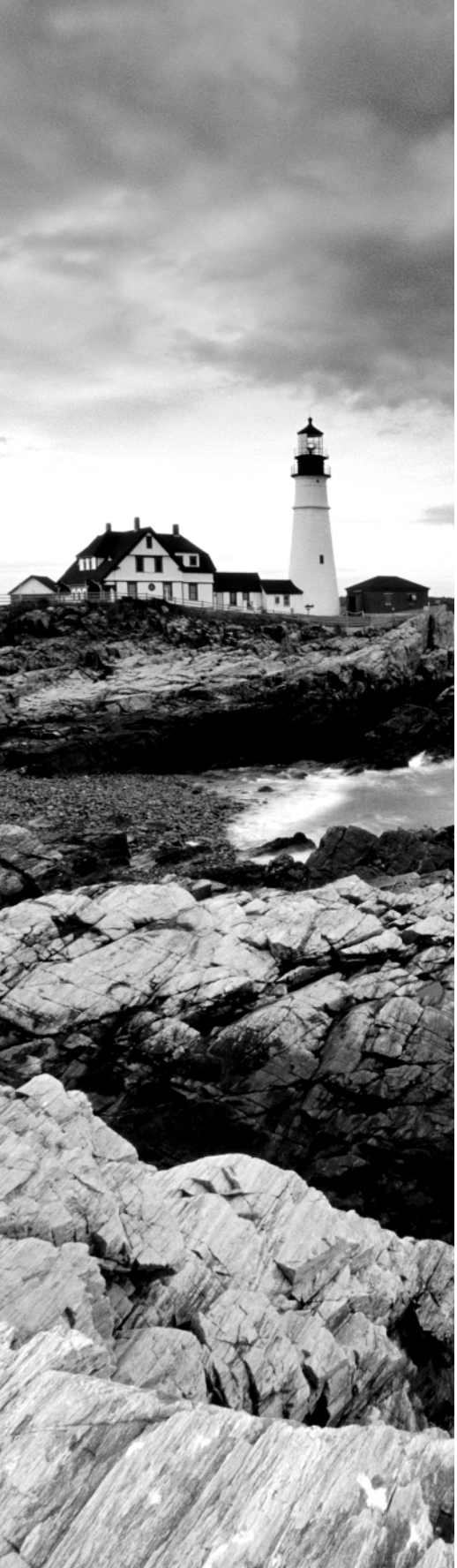
- This objective may include, but is not limited to:
  - Install NLB nodes
  - Configure NLB prerequisites
  - Configure affinity
  - Configure port rules
  - Configure cluster operation mode
  - Upgrade an NLB cluster

✓ **Configure failover clustering**

- This objective may include, but is not limited to:
  - Configure Quorum
  - Configure cluster networking
  - Restore single node or cluster configuration
  - Configure cluster storage
  - Implement cluster aware updating
  - Upgrade a cluster

✓ **Manage failover-clustering roles**

- This objective may include, but is not limited to:
  - Configure role-specific settings including continuously available shares



- Configure VM monitoring
- Configure failover and preference settings

✓ **Manage Virtual Machine (VM) Movement**

- This objective may include, but is not limited to:
  - Configure Virtual Machine network health protection
  - Configure drain on shutdown
  - Perform quick, live and storage migrations
  - Import/export/copy of VMS

The R2 update to Windows Server 2012 has improved upon the rich high availability capabilities already present in Windows Server 2012. The management, reporting, and ease of use of the feature set are all worth mentioning, but the expansion of features is the greatest benefit of the R2 update with regard to high availability.

The exam will cover the new features at a high level, and it will cover the basic configuration and operational functions for both a failover cluster and a network load balancer. This chapter will introduce how to achieve high availability with hardware and operational changes as well as how to use the high availability features of Windows Server 2012 R2.

Any discussion of high availability, network load balancers, and clustering would not be complete without a discussion of high availability in general. The chapter will first cover what it means, both from a purely technical perspective and from a business perspective.

Components of High Availability

High availability is a buzzword that many application and hardware vendors like to throw around to get you to purchase their products. Many different options are available to achieve high availability, and there also seems to be a number of definitions and variations that help vendors sell their products as high-availability solutions.

When it comes right down to it, however, high availability simply means providing services with maximum uptime by avoiding unplanned downtime. Often, *disaster recovery (DR)* is also closely lumped into discussions of high availability, but DR encompasses the business and technical processes that are used to recover once a disaster has happened.

Defining a high availability plan usually starts with a *service level agreement (SLA)*. At its most basic, an SLA defines the services and metrics that must be met for the availability and performance of an application or service. Often, an SLA is created for an IT department or service provider to deliver a specific level of service. An example of this might be an SLA for a Microsoft Exchange server. The SLA for an Exchange server might have uptime metrics on how much time during the month the mailboxes need to be available to end users, or it might define performance metrics for the amount of time it takes for email messages to be delivered.

When determining what goes into an SLA, two other factors need to be considered. However, you will often see them discussed only in the context of disaster recovery, even though they are important for designing a highly available solution. These factors are the *recovery point objective (RPO)* and the *recovery time objective (RTO)*.

An RTO is the length of time an application can be unavailable before service must be restored to meet the SLA. For example, a single component failure would have an RTO of less than five minutes, and a full-site failure might have an RTO of three hours. An RPO is essentially the amount of data that must be restored in the event of a failure. For example, in a single server or component failure, the RPO would be 0, but in a site failure, the RPO might allow for up to 20 minutes of lost data.

SLAs, on the other hand, are usually expressed in percentages of the time the application is available. These percentages are also often referred to by the number of nines the percentage includes, as shown in Table 1.1.

TABLE 1.1 Availability percentages

Availability rating	Allowed unplanned downtime/year
99 percent	3.7 days
99.9 percent	8.8 hours
99.99 percent	53 minutes
99.999 percent	5.3 minutes

Two important factors that affect an SLA are the *mean time between failure (MTBF)* and the *mean time to recovery (MTTR)*. To be able to reduce the amount of unplanned downtime, the time between failures must be increased, and the time it takes to recover must be reduced. Modifying these two factors will be addressed in the next several sections of this chapter.

Achieving High Availability

Windows Server 2012 R2 is the most secure and reliable Windows version to date. It also is the most stable, mature, and capable of any version of Windows. Although similar claims have been made for previous versions of Windows Server, you can rest assured that Windows Server 2012 R2 is much better than previous versions for a variety of reasons.

An honest look at the feature set and real-world use should prove that this latest version of Windows provides the most suitable foundation for creating a highly available solution. However, more than just good software is needed to be able to offer high availability for applications.

High Availability Foundation

Just as a house needs a good foundation, a highly available Windows server needs a stable and reliable hardware platform on which to run. Although Windows Server 2012 R2 will

technically run on desktop-class hardware, high availability is more easily achieved with server-class hardware. What differentiates desktop-class from server-class hardware? *Server-class hardware* has more management and monitoring features built into it so that the health of the hardware is capable of being monitored and maintained.

Another large difference is that server-class hardware has redundancy options. Server-class hardware often has options to protect from drive failures, such as RAID controllers, and to protect against power supply failures, such as multiple power supplies. Enterprise-class servers have even more protection.

More needs to be done than just installing Windows Server 2012 R2 to ensure that the applications remain running with the best availability possible. Just as a house needs maintenance and upkeep to keep the structure in proper repair, so too does a server. In the case of a highly available server, this means *patch management*.

Installing Patches

Microsoft releases monthly updates to fix security problems with its software, both for operating system fixes and for applications. To ensure that your highly available applications are immune to known vulnerabilities, these patches need to be applied in a timely manner during a scheduled maintenance window. Also, to address stability and performance issues, updates and service packs are released regularly for many applications, such as Microsoft SQL Server, Exchange Server, and SharePoint Portal Server. Many companies have a set schedule—daily, weekly, or monthly—to apply these patches and updates after they are tested and approved.

Desired Configuration Manager (DCM), an option in Microsoft System Center Configuration Manager 2012 and newer, is a great tool for helping to validate that your cluster nodes are patched. It can leverage the SCCM client to collect installed patches and help reporting within the enterprise on compliancy with desired system states based on the software installed.

To continue with the house analogy, if you were planning to have the master bath remodeled, would you rather hire a college student on spring break looking to make some extra money to do the job or a seasoned artisan? Of course, you would want someone with experience and a proven record of accomplishment to remodel your master bath.

Likewise, with any work that needs to be done on your highly available applications, it's best to hire only decidedly qualified individuals. This is why obtaining a Microsoft certification is definitely an excellent start to becoming qualified to configure a highly available server properly. There is no substitute for real-life and hands-on experience. Working with highly available configurations in a lab and in production will help you know not only what configurations are available but also how the changes should be made.

For example, it may be possible to use Failover Clustering for a WINS server, but in practice WINS replication may be easier to support and require less expensive hardware in order to provide high availability. This is something you would know only if you had enough experience to make this decision.

As with your house, once you have a firm and stable foundation built by skilled artisans and a maintenance plan has been put into place, you need to ascertain what more is

needed. If you can't achieve enough uptime with proper server configuration and mature operational processes, a cluster may be needed.

Windows Server 2012 R2 provides two types of clustering: *Failover Clustering* and *Network Load Balancing (NLB)*. Failover clustering is used for applications and services such as SQL Server and Exchange Server. Network Load Balancing is used for network-based services such as web and FTP servers. The remaining sections of this chapter will cover both of these clustering options in depth.

To Cluster or Not to Cluster

Clustering is often thrown into the mix when someone wants to achieve higher availability. This is a good step toward improved availability, but the return on the investment of a cluster doesn't always add up. Although Windows Server 2012 R2 greatly simplifies both the creation and management of a failover cluster, there is added complexity and cost in terms of hardware, software, and personnel.

How do you determine whether to cluster applications? Sometimes, even though it is possible to cluster applications, they perform worse when clustered. At other times, only a small improvement is made when a cluster is created. You have to balance the slight improvement over the increased hardware cost, complexity, and level of training required for administrators.

Configure Network Load Balancing

Network Load Balancing is a form of clustering where the nodes are highly available for a network-based service. This is typically a port listener configuration where a farm of, say, Microsoft Internet Information Services servers all listen on ports 80 and 443 for incoming web traffic from client endpoints. These nodes, while not fully clustered in a technical sense, are load balanced, where each node handles some of the distributed network traffic.

Network Load Balancing at the software level (as I will discuss here) is generally reserved for light loads or loads in lower-budget environments, such as a test or QA environment, for example. Generally speaking, Network Load Balancing in large production environments relies on hardware-based solutions to front-end network load balancing and distributes it on a session-based load to multiple hosts. This type of configuration, however, is out of scope for this book.

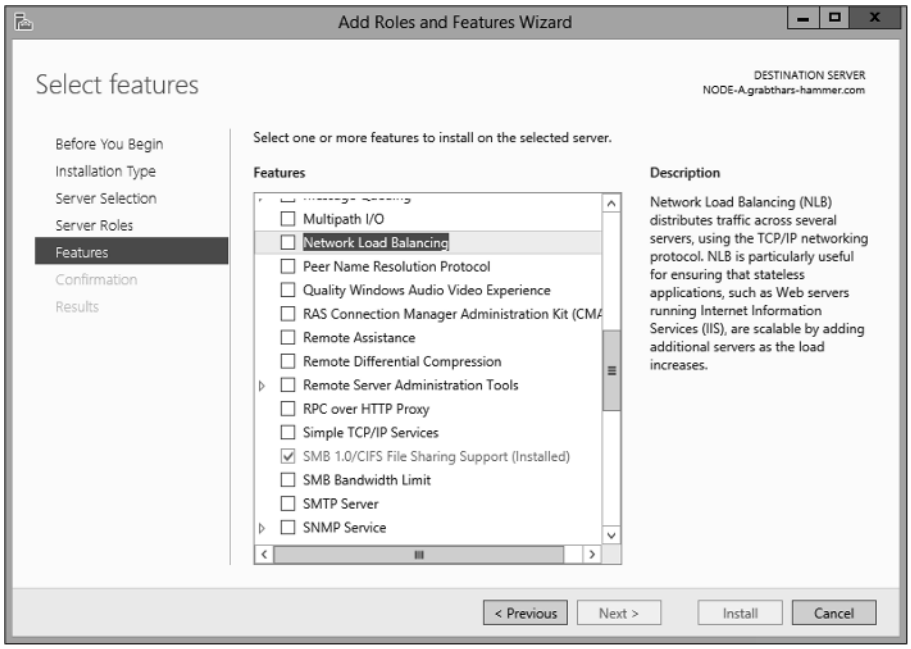
Install NLB Nodes

You can install NLB nodes like any other server build. You want the host patched, provisioned with appropriate resources (typically with multiple network interface cards for capacity and responsiveness), and monitored for health and reliability. In Exercise 1.1, you'll install NLB nodes.

EXERCISE 1.1

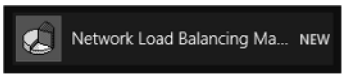
Installing NLB Nodes

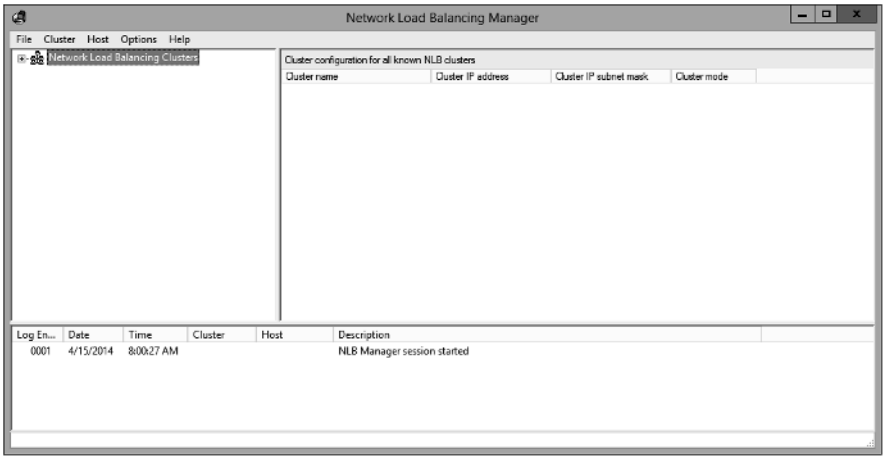
1. Once you have multiple hosts ready for the installation of NLB, simply run the Add Roles And Features Wizard and select Network Load Balancing in the Features area of the wizard.



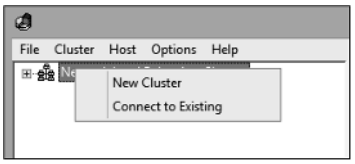
2. This wizard places a new application in your Start menu, the Network Load Balancing Manager (shown here), the execution of which loads the console.

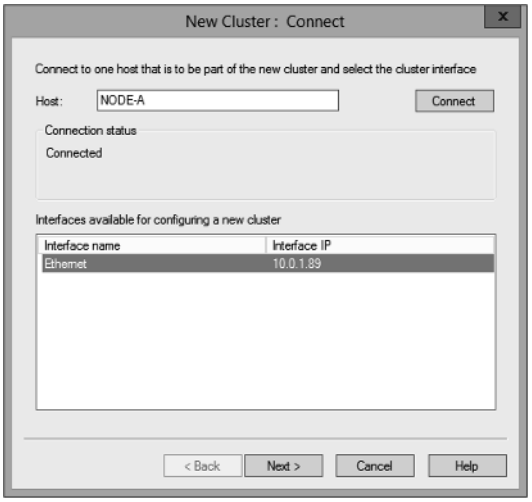
EXERCISE 1.1 *(continued)*



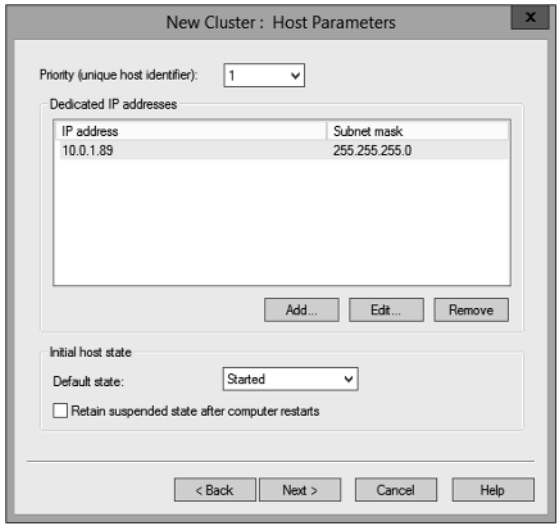


3. Right-click Network Load Balancing Clusters and select New Cluster. You are then presented with the connection wizard where you can specify the name of one of your hosts.

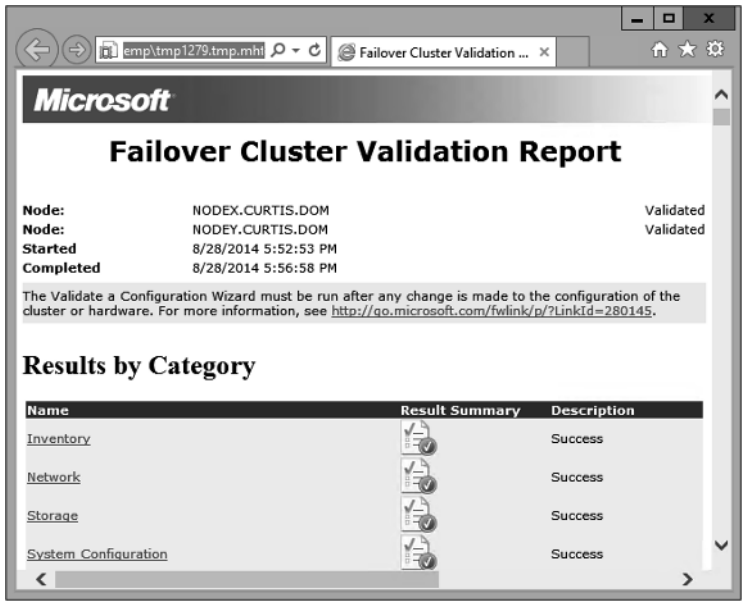




4. The next screen reveals a prompt to add any additional IPs and assign a priority level. You can do all this later, so hit Next.

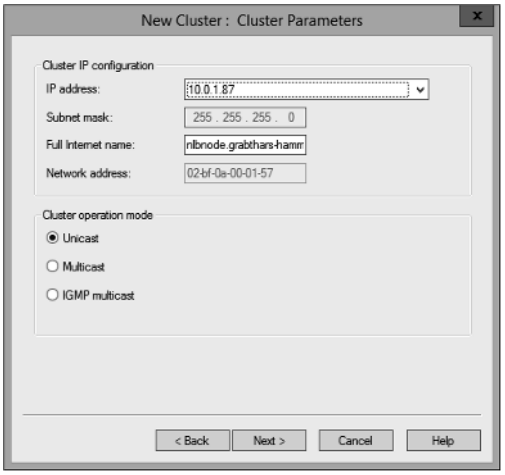


5. The next wizard screen is where you specify the cluster IP address. This is the address that the endpoints or clients or users of the NLB cluster will contact. Typically the network team will assign a cluster IP address for this use.



EXERCISE 1.1 *(continued)*

6. On the next screen, you configure the operation mode and specify a name.



With regard to the cluster operation modes, the differences between them are as follows:

Unicast

The cluster adapters for all nodes are assigned the same MAC address.

The outgoing MAC address for each packet is modified based on priority to prevent upstream switches from discovering that all nodes have the same MAC address.

Communication between cluster nodes (other than heartbeat and other administrative NLB traffic) is not possible unless there are additional adapters (because all nodes have the same MAC address).

Depending on load, this configuration can cause switch flooding since all inbound packets are sent to all ports on the switch.

Multicast

The cluster adapters for all nodes are assigned their own MAC unicast address.

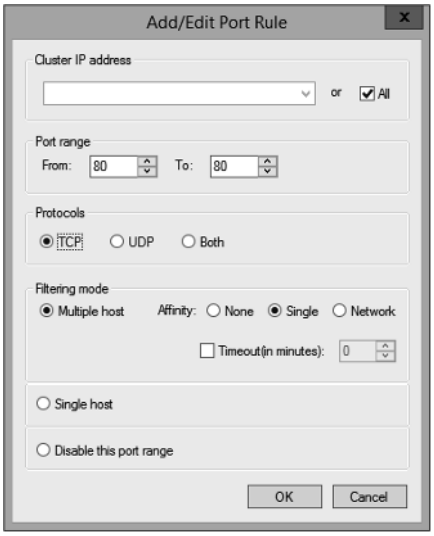
The cluster adapters for all nodes are assigned a multicast MAC address (derived from the IP of the cluster).

Non-NLB network traffic between cluster nodes works fine since they all have their own MAC address.

IGMP Multicast

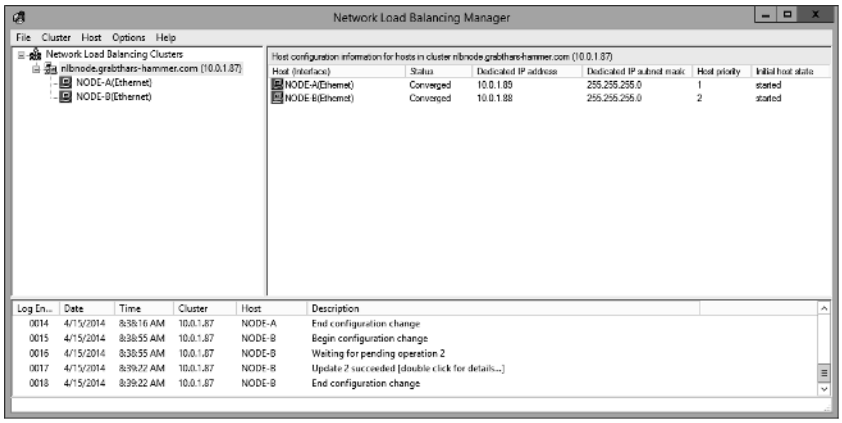
This is much like multicast, but the MAC traffic goes only to the switch ports of the NLB cluster, preventing switch flooding.

7. After selecting the appropriate settings, the next page is where port rules are configured. By default, it is set up to be wide open. Most implementations will limit NLB ports to just the ports needed for the application. For example, a web server would need port 80 enabled. It is also in this area where you can configure filtering mode.



The affinity sets a client's preference to a particular NLB host. It is not recommended to set affinity to None when UDP is an expected traffic type.

8. After clicking OK and finishing the wizard, you can add nodes to the NLB cluster by right-clicking and selecting Add Host To Cluster. Doing so presents a fairly straightforward wizard to add the host, specify a priority value, and then join the cluster. Shortly, your console will look similar to the one shown here.



Upgrading an NLB Cluster

Upgrading an NLB cluster is a fairly straightforward process. You execute a drainstop on the NLB cluster node or remove existing connections to the application on the local host, and then you can perform an in-place upgrade in a rolling manner.

Achieving High Availability with Failover Clustering

Taking high availability to the next level for enterprise services often means creating a failover cluster. In a failover cluster, all of the clustered application or service resources are assigned to one node or server in the cluster. Commonly clustered applications are SQL Server and Exchange Server; commonly clustered services are File and Print. Since the differences between a clustered application and a clustered service are primarily related to the number of functions or features, for simplicity's sake I will refer to both as *clustered applications*. Another, more frequently, clustered resource is a Hyper-V virtual machine.

If there is a failure of the primary node or if the primary node is taken offline for maintenance, the clustered application is started on another cluster node. The client requests are then automatically redirected to the new cluster node to minimize the impact of the failure.

How does Failover Clustering improve availability? By increasing the number of server nodes available on which the application or virtual machine can run, you can move the application or virtual machine to a healthy server if there is a problem, if maintenance needs to be completed on the hardware or the operating system, or if patches need to be applied. The clustered application being moved will have to restart on the new server regardless of whether the move was intentional. This is why the term *highly available* is used instead of *fault tolerant*. Virtual machines, however, can be moved from one node to another node using a process known as *live migration*. Live migration is where one or more virtual machines are intentionally moved from one node to another with their current memory state intact through the cluster network with no indicators to the virtual machine consumer that the virtual machine has moved from one server to another. However, in the event of a cluster node or virtual machine failure, the virtual machine will still fail and will then be brought online again on another healthy cluster node.

Figure 1.1 shows an example of SQL Server running on the first node of a Windows Server 2012 R2 failover cluster.

The clustered SQL Server in Figure 1.2 can be failed over to another node in the cluster and still service database requests. However, the database will be restarted.

Failover clustering is notorious for being complicated and expensive. Windows Server 2012 R2 makes strides in removing both of these concerns. Troubleshooting and other advanced concepts are outside of the scope of the Microsoft MCSA exams and thus this book, so I will cover only the basic requirements and concepts needed to configure a failover cluster.

FIGURE 1.1 Using Failover Clustering to cluster SQL Server

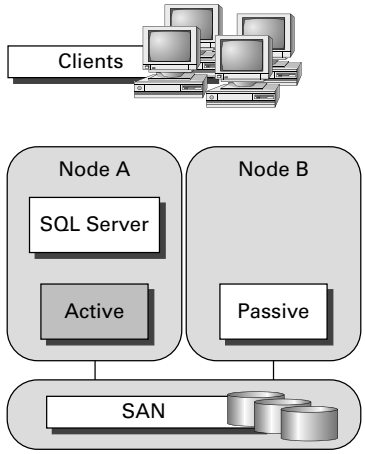
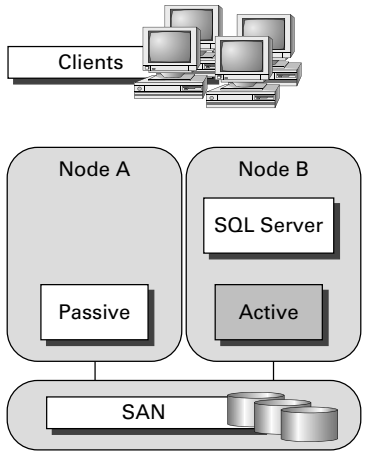


FIGURE 1.2 Failing the SQL Server service to another node



Failover Clustering Requirements

The Failover Clustering feature is available in the Datacenter, Standard, and Hyper-V editions of Windows Server 2012 R2.

To be able to configure a failover cluster, you must have the required components. A single failover cluster can have up to 64 nodes when using Windows Server 2012 R2, however, and the clustered service or application must support that number of nodes.

To create a failover cluster, an administrator must make sure that all the hardware involved meets the cluster requirements. To be supported by Microsoft, all hardware must be certified for Windows Server 2012 R2, and the complete failover cluster solution must

pass all tests in the Validate A Configuration Wizard. Although the exact hardware will depend on the clustered application, a few requirements are standard:

- Server components must be marked with the "Certified for Windows Server 2012 R2" logo.
- Although not explicitly required, server hardware should match and contain the same or similar components.
- All of the Validate A Configuration Wizard tests must pass.

The requirements for Failover Clustering storage have changed from previous versions of Windows. For example, Parallel SCSI is no longer a supported storage technology for any of the clustered disks. There are, however, additional requirements that need to be met for the storage components:

- Disks available for the cluster must be Fibre Channel, iSCSI, or Serial Attached SCSI.
- Each cluster node must have a dedicated network interface card for iSCSI connectivity. The network interface card you use for iSCSI should not be used for network communication.
- Multipath software must be based on Microsoft's Multipath I/O (MPIO).
- Storage drivers must be based on `storport.sys`.
- Drivers and firmware for the storage controllers on each server node in the cluster should be identical.
- Storage components must be marked with the "Certified for Windows Server 2012 R2" logo.

In addition, there are network requirements that must be met for Failover Clustering:

- Cluster nodes should be connected to multiple networks for communication redundancy.
- Network adapters should be the same make, use the same driver, and have the firmware version in each cluster node.
- Network components must be marked with the "Certified for Windows Server 2012 R2" logo.

There are two types of network connections in a failover cluster. These should have adequate redundancy because total failure of either could cause loss of functionality of the cluster. The two types are as follows:

Public Network This is the network through which clients are able to connect to the clustered service application.

Private Network This is the network used by the nodes to communicate with each other.

To provide redundancy for these two network types, additional network adapters would need to be added to the node and configured to connect to the networks.

In previous versions of Windows Server, support was given only when the entire cluster configuration was tested and listed on the Hardware Compatibility List. The tested configuration listed the server and storage configuration down to the firmware and driver versions. This proved to be difficult and expensive from both a vendor and a consumer perspective to deploy supported Windows clusters.

When problems did arise and Microsoft support was needed, it caused undue troubleshooting complexity as well. With Windows Server 2012 R2 Failover Clustering

and simplified requirements, including the "Certified for Windows Server 2012 R2" logo program and the Validate A Configuration Wizard, it all but eliminates the guesswork of getting the cluster components configured in a way that follows best practices and allows Microsoft support to assist you easily when needed.

Cluster Quorum

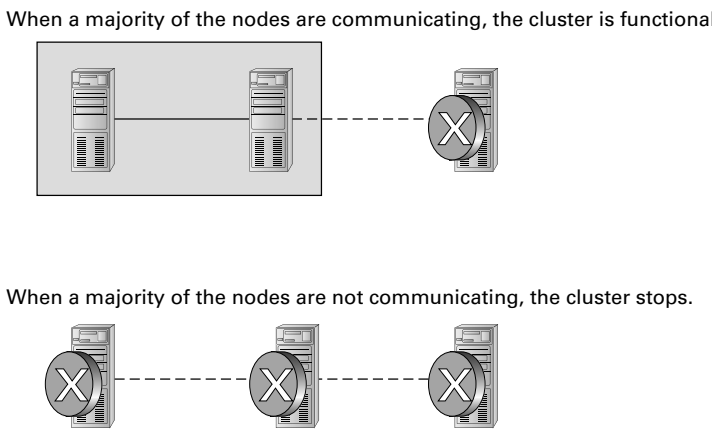
When a group of people sets out to accomplish a single task or goal, a method for settling disagreements and for making decisions is required. In the case of a cluster, the goal is to provide a highly available service in spite of failures. When a problem occurs and a cluster node loses communication with the other nodes because of a network error, the functioning nodes are supposed to try to bring the redundant service back online.

How, though, is it determined which node should bring the clustered service back online? If all the nodes are functional despite the network communications issue, each one might try. Just like a group of people with their own ideas, a method must be put in place to determine which idea, or node, to grant control of the cluster. Windows Server 2012 R2 Failover Clustering, like other clustering technologies, requires that a quorum exist between the cluster nodes before a cluster becomes available.

A *quorum* is a consensus of the status of each of the nodes in the cluster. Quorum must be achieved in order for a clustered application to come online by obtaining a majority of the votes available (see Figure 1.3). Windows Server 2012 R2 has four quorum models, or methods, for determining quorum and for adjusting the number and types of votes available:

- Node majority (no witness)
- Node majority with witness (disk or file share)
- Node and file share majority
- No majority (disk witness only)

FIGURE 1.3 Majority needed



Witness Configuration

Most administrators follow some basic rules. For example, when you configure a quorum, the voting components in the cluster should be an odd number. For example, if I set up a quorum for five elements and I lose one element, I continue to work. If I lose two elements, I continue to work. If I lose three elements, the cluster stops—as soon as it hits half plus 1, the cluster stops. This works well with an odd number.

If the cluster contains an even number of voting elements, an administrator should then configure a disk witness or a file share witness. The advantage of using a witness (disk or file share) is that the cluster will continue to run even if half of the cluster nodes simultaneously go down or are disconnected. The ability to configure a disk witness is possible only if the storage vendor supports read-write access from all sites to the replicated storage.

One of the advantages of Windows Server 2012 R2 is the advanced quorum configuration option. This option allows you to assign or remove quorum votes on a per-node basis. Administrators now have the ability to remove votes from nodes in certain configurations. For example, if your organization uses a multisite cluster, you may choose to remove votes from the nodes in the backup site. This way, those backup nodes would not affect your quorum calculations.

Dynamic Quorum Management

Another advantage in Windows Server 2012 R2 is dynamic quorum management. *Dynamic quorum management* automatically manages the vote assignment to nodes. With this feature enabled, votes are automatically added or removed from nodes when that node either joins or leaves a cluster. In Windows Server 2012 R2, dynamic quorum management is enabled by default.

Validating a Cluster Configuration

Configuring a failover cluster in Windows Server 2012 R2 is much simpler than in previous versions of Windows Server. Before a cluster can be configured, the Validate A Configuration Wizard should be run to verify that the hardware is configured in a fashion that is supportable. Before you can run the Validate A Configuration Wizard, however, the Failover Clustering feature needs to be installed using Server Manager. The account that is used to create a cluster must have administrative rights on each of the cluster nodes and have permissions to create a cluster name object in Active Directory. Follow these steps:

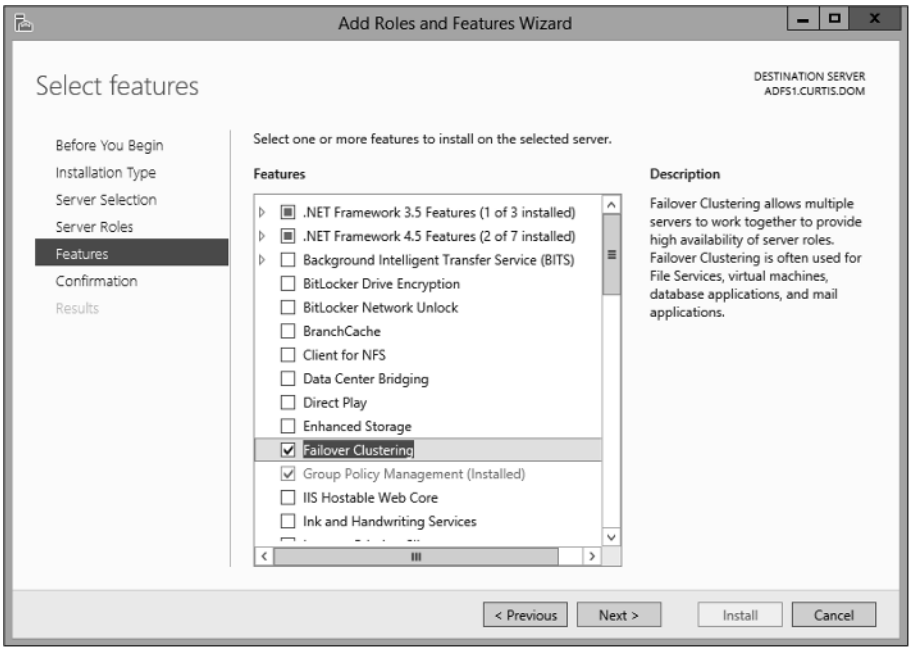
1. Prepare the hardware and software perquisites.
2. Install the Failover Clustering feature on each server.
3. Log in with the appropriate user ID and run the Validate A Configuration Wizard.
4. Create a cluster.
5. Install and cluster applications and services.

To install the Failover Clustering feature on a cluster node, follow the steps outlined in Exercise 1.2.

EXERCISE 1.2

Installing the Failover Cluster Feature

1. Press the Windows key and select Administrative Tools ➢ Server Manager.
2. Select number 2, Add Roles And Features.
3. At the Select Installation Type screen, choose a role-based or feature-based installation.
4. At the Select Destination Server screen, choose Select A Server From The Server Pool and click Next.
5. At the Select Server Roles screen, click Next.
6. At the Select Features screen, click the Failover Clustering check box. If the Add Features dialog box appears, click the Add Features button. Click Next.
7. At the Confirmation screen, click the Install button.
8. Once the installation is complete, click the Close button.
9. Close Server Manager.



Using the Validate A Configuration Wizard before creating a cluster is highly recommended. This wizard validates that the hardware configuration and the software configuration for the potential cluster nodes are in a supported configuration. Even if the configuration passes the tests, take care to review all warnings and informational messages so that they can be addressed or documented before the cluster is created.

Running the Validate A Configuration Wizard does the following:

- Conducts four types of tests (software and hardware inventory, network, storage, and system configuration)
- Confirms that the hardware and software settings are supportable by Microsoft support staff

You should run the Validate A Configuration Wizard before creating a cluster or after making any major hardware or software changes to the cluster. Doing this will help you identify any misconfigurations that could cause problems with the failover cluster.

Running the Validate a Configuration Wizard

The Validate A Configuration Wizard, shown in Figure 1.4, is simple and straightforward to use, as its "wizard" name would suggest. It should be run after the Failover Clustering feature has been installed on each of the cluster nodes, and it can be run as many times as required.

FIGURE 1.4 The Validate A Configuration Wizard

When you are troubleshooting cluster problems or have changed the configuration of the cluster hardware, it is a good idea to run the Validate A Configuration Wizard again to help pinpoint potential cluster configuration problems.

If you already have a cluster configured and want to run the Validate A Configuration Wizard, you can do so; however, you will not be able to run all of the storage tests without taking the clustered resources offline. You will be prompted either to skip the disruptive tests or to take the clustered resources offline so that the tests can complete.

Exercise 1.3 shows the exact steps to follow to run the Validate A Configuration Wizard successfully on clusters named NODEA and NODEB, which are not yet clustered.

I am using NODEA and NODEB in the exercises. You need to replace these two nodes with your own two servers to complete these exercises.

EXERCISE 1.3

Running the Validate A Configuration Wizard

1. Press the Windows key and select Administrative Tools ➢ Failover Cluster Management.
2. In the Actions pane (right side of screen), click Validate Configuration.

3. At the Before You Begin screen, click Next.
4. Type **NODEA** in the Enter Name field and click Add.
5. Type **NODEB** in the Enter Name field and click Add.

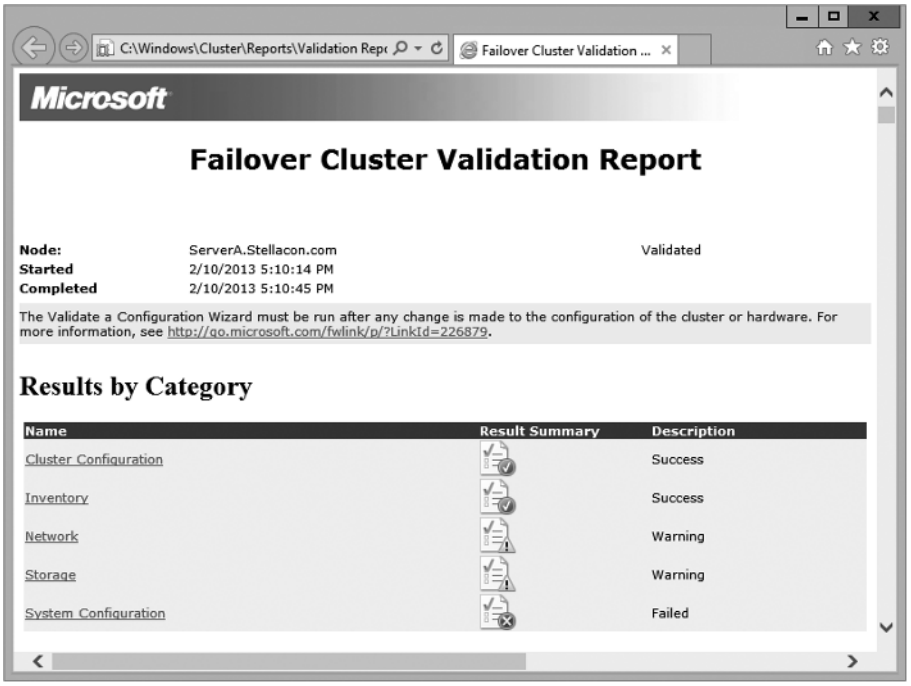
EXERCISE 1.3 *(continued)*

6. Click Next.
7. Leave Run All Tests (Recommended) selected and click Next.
8. Click Next at the Confirmation screen.
9. Let the test complete, review the report in the Summary window, and then click Finish.

Addressing Problems Reported by the Validate A Configuration Wizard

After the Validate A Configuration Wizard has been run, it will show the results, as shown in Figure 1.5. This report can also be viewed in detail later using a web browser. The report is named with the date and time the wizard was run, and it is stored in `%windir%\cluster\Reports`.

FIGURE 1.5 Validate A Configuration Wizard results



How should errors listed in the report be addressed? Often, the errors reported by the Validate A Configuration Wizard are self-explanatory; however, sometimes additional help is required. The following three guidelines should help troubleshoot the errors:

- Read all of the errors because multiple errors may be related.
- Use the checklists available in the Windows Server help files to ensure that all the steps have been completed.
- Contact the hardware vendor for updated drivers, firmware, and guidance for using the hardware in a cluster.

Multisite, Stretched, or Geographically Dispersed Clusters (Geoclustering)

One issue you may face is if you have multiple sites or if the cluster is geographically dispersed. If the failover cluster does not have a shared common disk, data replication between nodes might not pass the cluster validation "storage" tests.

Setting up a cluster in a multisite, stretched, or geocluster (these terms can be used interchangeably) configuration is a common practice. As long as the cluster solution does not require external storage to fail over, it will not need to pass the storage test to function properly.

Creating a Cluster

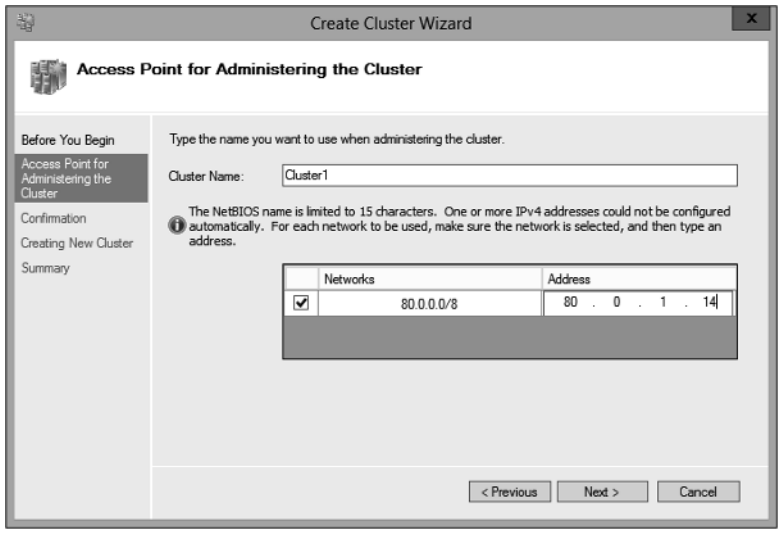
After you have successfully validated a configuration and the cluster hardware is in a supportable state, you can create a cluster. The process for creating a cluster is straightforward and similar to the process of running the Validate A Configuration Wizard. To create a cluster with two servers, follow the instructions in Exercise 1.4.

EXERCISE 1.4

Creating a Cluster

1. Open the Failover Cluster Management MMC.
2. In the Management section of the center pane, select Create A Cluster.
3. Read the Before You Begin information and click Next.
4. In the Enter Server Name box, type **NODEA** and then click Add.
5. Again, in the Enter Server Name box, type **NODEB** and then click Add. Click Next.
6. At the Validation screen, choose No for this exercise and then click Next.
7. In the Access Point For Administering The Cluster section, enter **Cluster1** for the cluster name.
8. Type an IP address and then click Next. This IP address will be the IP address of the cluster.

EXERCISE 1.4 *(continued)*



9. In the Confirmation dialog box, verify the information and then click Next.
10. On the Summary page, click Finish.

Working with Cluster Nodes

Once a cluster is created, a couple of actions are available. First you can add another node to the cluster by using the Add Node Wizard from the Failover Cluster Management Actions pane.

At this point, you also have the option to pause a node, which prevents resources from being failed over or moved to the node. You typically would pause a node when the node is involved in maintenance or troubleshooting. After a node is paused, it must be resumed to allow resources to be run on it again.

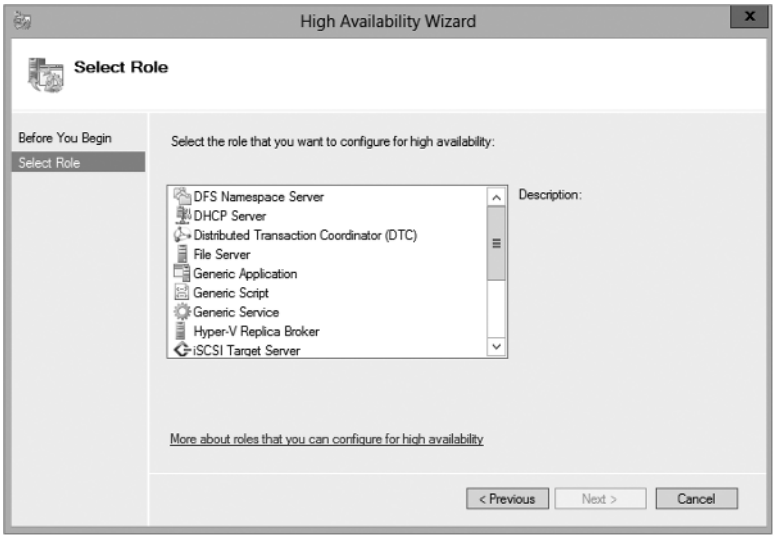
Another action available to perform on a node at this time is *evict*. Eviction is an irreversible process. Once you evict the node, it must be re-added to the cluster. You would evict a node when it is damaged beyond repair or is no longer needed in the cluster. If you evict a damaged node, you can repair or rebuild it and then add it back to the cluster using the Add Node Wizard.

Clustering Roles, Services, and Applications

Once the cluster is created, applications, services, and roles can be clustered. Windows Server 2012 R2 includes a number of built-in roles and features that can be clustered.

The following roles and features can be clustered in Windows Server 2012 R2 (see Figure 1.6):

FIGURE 1.6 High availability roles



- DFS Namespace Server
- DHCP Server
- Distributed Transaction Coordinator (DTC)
- File Server
- Generic Application
- Generic Script
- Generic Service
- Hyper-V Replica Broker
- iSCSI Target Server
- iSNS Server
- Message Queuing
- Other Server
- Virtual Machine
- WINS Server

In addition, other common services and applications can be clustered on Windows Server 2012 R2 clusters:

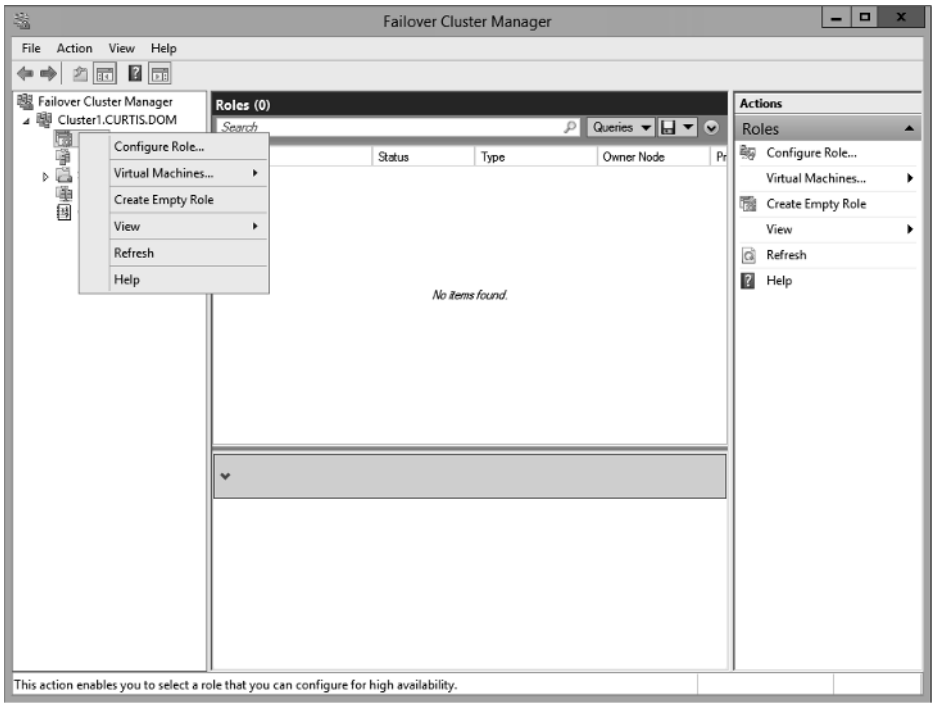
- Enterprise database services, such as Microsoft SQL Server
- Enterprise messaging services, such as Microsoft Exchange Server

To cluster a role or feature such as Print Services, the first step is to install the role or feature on each node of the cluster. The next step is to use the Configure A Service Or Application Wizard in the Failover Cluster Management tool. Exercise 1.5 shows you how to cluster the Print Services role once an appropriate disk has been presented to the cluster.

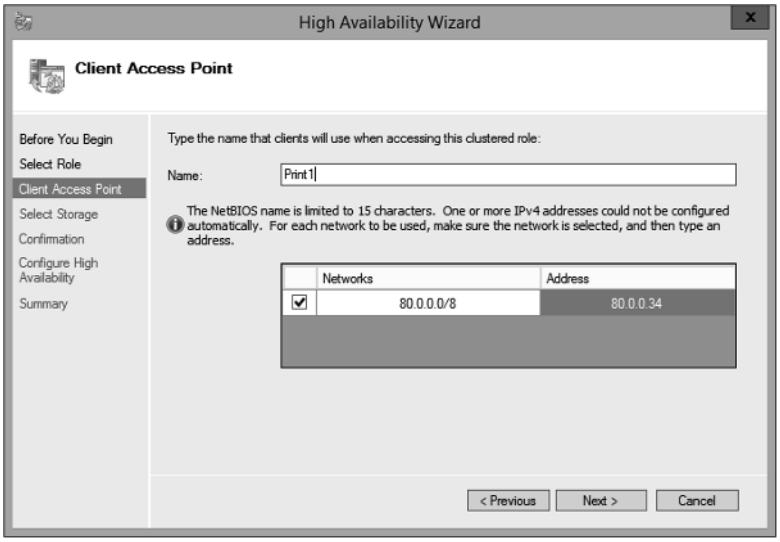
EXERCISE 1.5

Clustering the Print Services Role

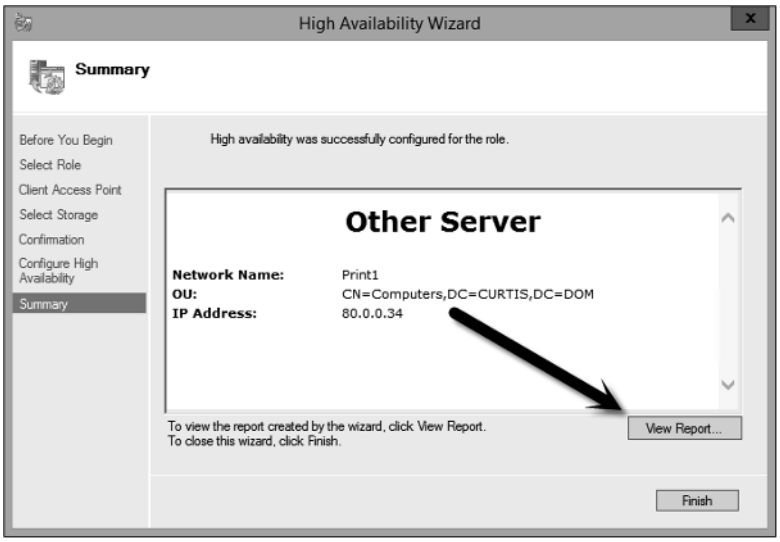
1. Open the Failover Cluster Management MMC.
2. In the console tree, click the arrow next to the cluster name to expand the items underneath it.
3. Right-click Roles and choose Configure Role.



4. Click Next on the Before You Begin page.
5. Click Other Server on the Select Role screen and then click Next.
6. Type the name of the print server, such as **Print1**, and type in the IP address that will be used to access the print service, such as **80.0.0.34**. Then click Next.



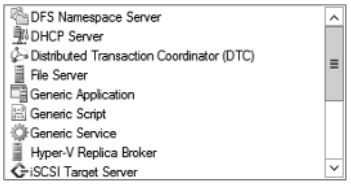
7. At the Select Storage page, just click Next.
8. Click Next at the Confirmation page.
9. After the wizard runs and the Summary page appears, you can view a report of the tasks the wizard performed by clicking View Report.



10. Close the report and click Finish.

The built-in roles and features all are configured in a similar fashion. Other applications, such as Microsoft Exchange Server 2013, have specialized cluster configuration routines that are outside the scope of this exam. Applications that are not developed to be clustered can also be clustered using the Generic Application, Generic Script, or Generic Service option in the Configure A Service Or Application Wizard, as shown in Figure 1.7.

FIGURE 1.7 Configuring a generic application



Clustered Application Settings

Windows Server 2012 R2 has options that allow an administrator to fine-tune the failover process to meet the needs of their business. These options will be covered in the next few sections.

Failover occurs when a clustered application or service moves from one node to another. The process can be triggered automatically because of a failure or server maintenance or can be done manually by an administrator. The failover process works as follows:

1. The cluster service takes all of the resources in the application offline in the order set in the dependency hierarchy.
2. The cluster service transfers the application to the node that is listed next on the application's list of preferred host nodes.
3. The cluster service attempts to bring all of the application's resources online, starting at the bottom of the dependency hierarchy.

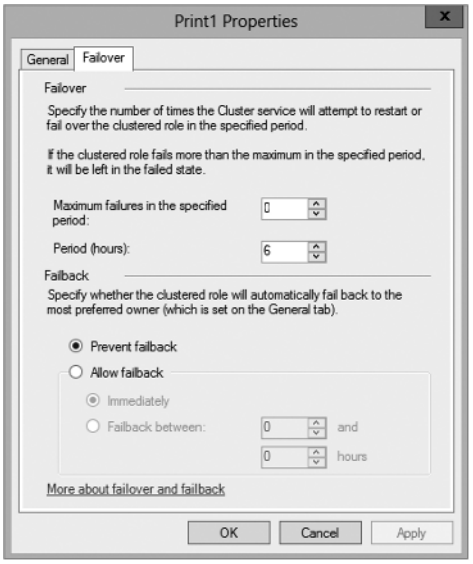
These steps can change depending on the use of Live Migration.

In a cluster that is hosting multiple applications, it may be important to set specific nodes to be primarily responsible for each clustered application. This can be helpful from a troubleshooting perspective since a specific node is targeted for hosting service. To set a preferred node and an order of preference for failover, use the General tab in the Properties dialog box of the clustered application.

Also, the order of failover is set in this same dialog box by moving the order in which the nodes are listed. If NODEA should be the primary node and NODEC should be the server that the application fails to first, NODEA should be listed first and selected as the preferred owner. NODEC should be listed second, and the remaining cluster nodes should be listed after NODEC.

As shown in Figure 1.8, a number of failover settings can be configured for the clustered service. The failover settings control the number of times a clustered application can fail in a period of time before the cluster stops trying to restart it. Typically, if a clustered application fails a number of times, some sort of manual intervention will be required to return the application to a stable state.

FIGURE 1.8 Clustered application failover settings



Specifying the maximum number of failures will keep the application from trying to restart until it is manually brought back online after the problem has been resolved. This is beneficial because if the application continues to be brought online and then fails, it may show as being functional to the monitoring system, even though it continues to fail. After the application is put in a failed state, the monitoring system will not be able to contact the application and should report it as being offline.

Figure 1.8 also shows the failback settings for Print1. Failback settings control whether and when a clustered application would fail back to the preferred cluster node once it becomes available. The default setting is Prevent Failback. If failback is allowed, two additional options are available, either to fail back immediately after the preferred node is available or to fail back within a specified time.

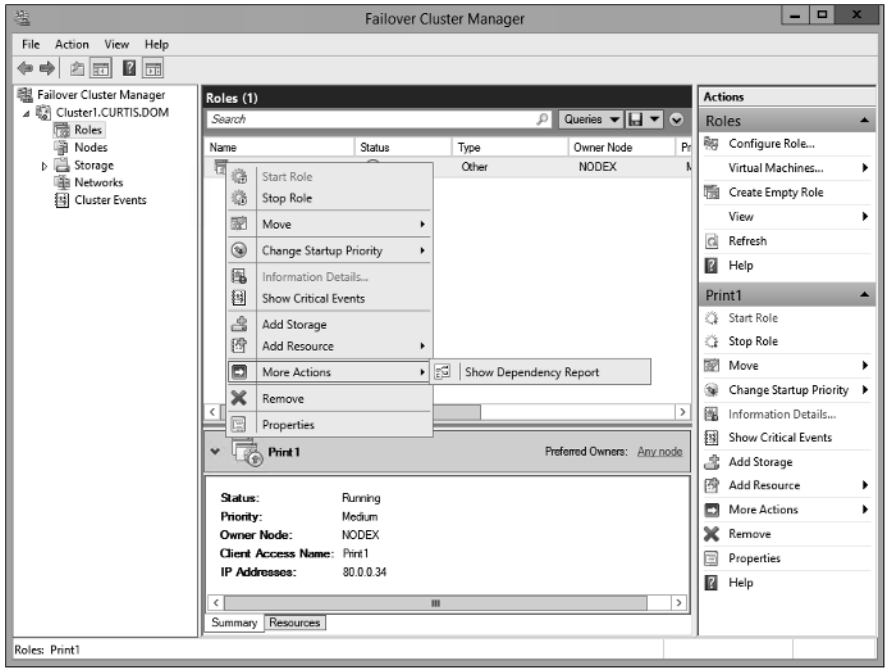
The time is specified in the 24-hour format. If you want to allow failback between 10 p.m. and 11 p.m., you would set the failback time to be between 22 and 23. Setting a failback time to off-hours is an excellent way to ensure that your clustered applications are running on the designated nodes and automatically scheduling the failover process for a time when it will impact the fewest users.

One tool that is valuable in determining how resources affect other resources is the dependency viewer. The *dependency viewer* visualizes the dependency hierarchy created for an application or service. Using this tool can help when troubleshooting why specific resources are causing failures and allow an administrator to visualize the current configuration better and adjust it to meet business needs. Exercise 1.6 will show you how to run the dependency viewer.

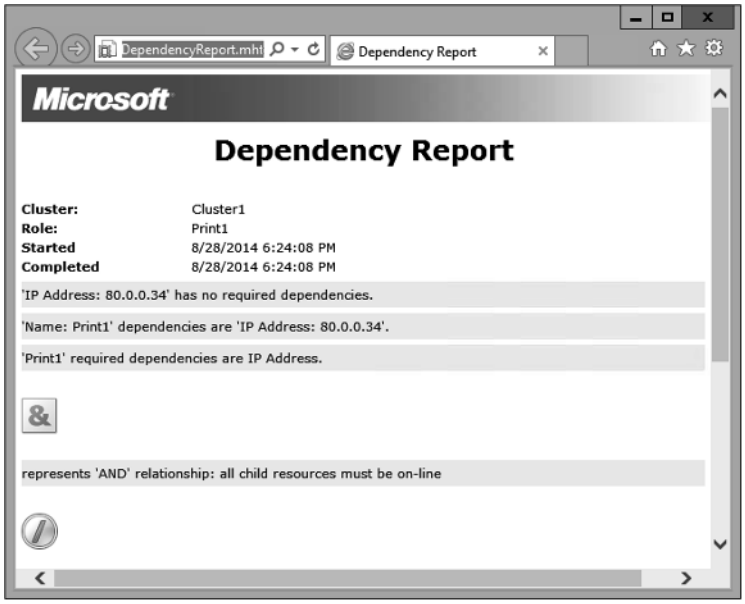
EXERCISE 1.6

Using the Dependency Viewer

1. Open the Failover Cluster Management MMC.
2. In the console tree, click the arrow to expand the cluster.
3. Click Roles.
4. Under the Roles section in the center of the screen, click one of the roles (such as Print1).
5. Right-click the role and under More Actions click Show Dependency Report.



6. Review the dependency report.



7. Close the Dependency Report and close the Failover Cluster Manager.

Exercise 1.6 generated a dependency report that shows how the print service is dependent on a network name and a clustered disk resource. The network name is then dependent on an IP address.

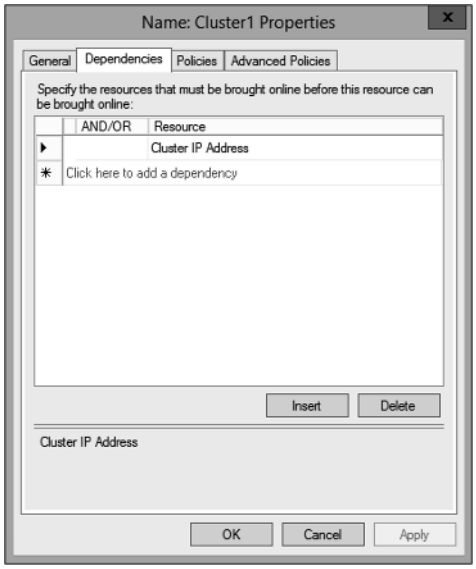
Resource Properties

Resources are physical or logical objects, such as a file share or IP address, which the failover cluster manages. They may be a service or application available to clients, or they may be part of the cluster. Resources include physical hardware devices such as disks and logical items such as network names. They are the smallest configurable unit in a cluster and can run on only a single node in a cluster at a time.

Like clustered applications, resources have a number of properties available for meeting business requirements for high availability. This section covers resource dependencies and policies.

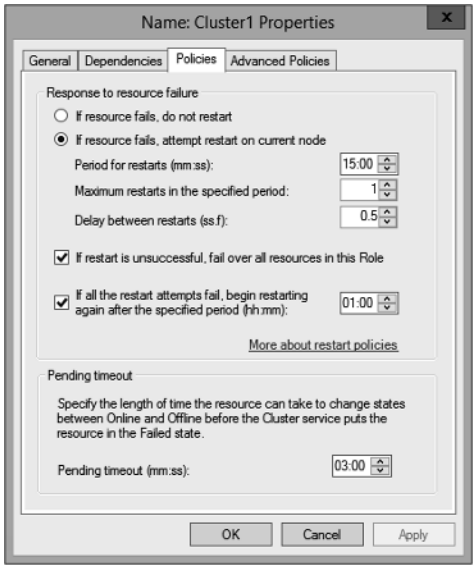
Dependencies can be set on individual resources and control how resources are brought online and offline. Simply put, a dependent resource is brought online after the resources that it depends on, and it is taken offline before those resources. As shown in Figure 1.9, dependencies can be set on a specific resource, such as the print spooler.

FIGURE 1.9 Resource dependencies



Resource policies are settings that control how resources respond when a failure occurs and how resources are monitored for failures. Figure 1.10 shows the Policies tab of a resource's Properties dialog box.

FIGURE 1.10 Resource policies



The Policies tab sets configuration options for how a resource should respond in the event of a failure. The options available are as follows:

If Resource Fails, Do Not Restart This option, as it would lead you to believe, leaves the failed resource offline.

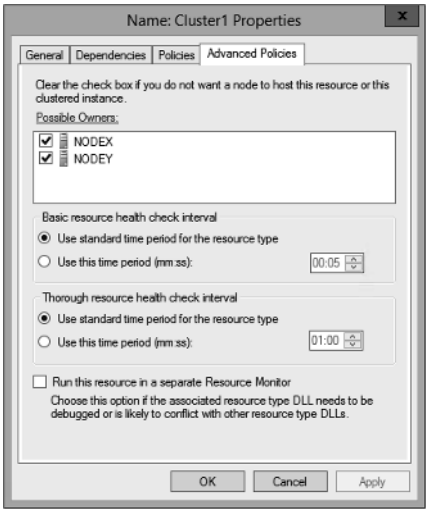
If Resource Fails, Attempt Restart On Current Node With this option set, the resource tries to restart if it fails on the node on which it is currently running. There are two additional options if this is selected so that the number of restarts can be limited. They set the number of times the resource should restart on the current node in a specified length of time. For example, if you specify 5 for Maximum Restarts In The Specified Period and 10:00 (mm:ss) for Period For Restarts, the cluster service will try to restart the resource five times during that 10-minute period. After the fifth restart, the cluster service will no longer attempt to restart the service on the active node.

If Restart Is Unsuccessful, Fail Over All Resources In This Service Or Application If this option is selected, when the cluster service is no longer trying to restart the resource on the active node, it will fail the entire service or application to another cluster node. If you wanted to leave the application or service with a failed resource on the current node, you would clear this check box.

If All The Restart Attempts Fail, Begin Restarting Again After The Specified Period (hh:mm) If this option is selected, the cluster service will restart the resource at a specified interval if all previous attempts have failed.

Pending Timeout This option is used to set the amount of time in minutes and seconds that the cluster service should wait for this resource to respond to a change in states. If a resource takes longer than the cluster expects to change states, the cluster will mark it as having failed. If a resource consistently takes longer than this and the problem cannot be resolved, you may need to increase this value. Figure 1.11 shows the Advanced Policies tab.

FIGURE 1.11 Resource Advanced Policies



The options available on the Advanced Policies tab are as follows:

Possible Owners This option allows an administrator to remove specific cluster nodes from running this resource. Using this option is valuable when there are issues with a resource on a particular node and the administrator wants to keep the applications from failing over to that node until the problem can be repaired.

Basic Resource Health Check Interval This option allows an administrator to customize the health check interval for this resource.

Thorough Resource Health Check Interval This option allows an administrator to customize the thorough health check interval for this resource.

Run This Resource In A Separate Resource Monitor If the resource needs to be debugged by a support engineer or if the resource conflicts with other resources, this option may need to be used.

Windows Server 2012 R2 Clustering Features

Many new features are included in the Windows Server 2012 R2 release for clustering. It is a rich feature set of high availability with greatly improved flexibility based on the needs of IT organizations. The new features relate to quorum behavior, virtual machine hosting, Active Directory–detached clusters, and a new dashboard.

Windows PowerShell Cmdlets for Failover Clusters As I have explained throughout this book, Windows PowerShell is a command-line shell and scripting tool. Windows Server 2012 R2 clustering has new cmdlets that provide powerful ways to script cluster configuration and management tasks. Windows PowerShell cmdlets have now replaced the `Cluster.exe` command-line interface.

Cluster Shared Volumes *Cluster Shared Volumes (CSV)* allows for the configuration of clustered virtual machines. CSV allows you to do the following:

- Reduce the number of LUNs (disks) required for your virtual machines.
- Make better use of disk space. Any VHD file on that LUN can use the free space on a CSV volume.
- More easily track the paths to VHD files and other files used by virtual machines.
- Use a few CSV volumes to create a configuration that supports many clustered virtual machines.

CSV volumes also are utilized for the Scale-Out-File-Server cluster role.

Management of Large-Scale Clusters One new advantage of Windows Server 2012 R2 clusters is the ability for Server Manager to discover and manage the nodes in a cluster. By

starting the Failover Cluster Manager from Server Manager, you can do remote multiserver management and role and feature installation. Administrators now have the ability to manage a cluster from one convenient location.

Management and Mobility of Clustered Virtual Machines Microsoft has built Windows Server 2012 R2 "from the cloud up." Microsoft, as well as the industry as a whole, is moving toward the cloud and virtualization. With that in mind, administrators can now configure settings such as prioritizing the starting or placement of virtual machines in the clustered workloads. This allows administrators to allocate resources efficiently to your cluster.

Cluster-Aware Updating One issue that every administrator has dealt with is updating a system or application while it is running. For example, if you are running Microsoft Exchange and you want to do an Exchange update, when do you take the server offline to do the update? It always seems that someone is on the system 24 hours a day. Well, Windows Server 2012 R2 clustering has a solution. *Cluster-Aware Updating (CAU)* is a new automated feature that allows system updates to be applied automatically while the cluster remains available during the entire update process.

Scale-Out File Server for Application Data By utilizing *Microsoft Storage Spaces*, you can create a highly available clustered file share that utilizes SMB 3.0 and CSV to provide scalable access to data.

Scale-out file servers are useful for storing the following application data:

- Hyper-V virtual machine storage
- SQL Server database files

Be aware that scale-out file servers are not useful at all for typical file share data because they benefit only from applications that require a persistent connection to their storage.

Shared Virtual Hard Disks In the previous versions of Windows, Failover Cluster nodes running as virtual machines had to use iSCSI or virtual HBAs to connect directly to SAN-based storage. With Windows Server 2012 R2, you can set your Hyper-V virtualized cluster to use a shared VHDX virtual disk. Shared virtual hard disks can reside on the following:

- A scale-out file server failover cluster
- Cluster CSV volumes

Shared virtual hard disks are extremely useful in providing highly available shared storage for the following virtualized workloads:

- SQL Server
- Virtual Machine Manager
- Exchange Server

Virtual Machine Drain on Shutdown When needing to perform maintenance on a Hyper-V failover cluster, you may have a lot of virtual machines on one node of a cluster. Inevitably, you will need to restart a cluster node for updates or shut it down for maintenance.

In previous versions of Windows, virtual machines running on the cluster would save their state, and then the cluster node would shut down. Windows Server 2012 R2 helps alleviate this issue by automatically draining the virtual machines running on a node before it shuts down or restarts. Windows does this by attempting to live migrate all virtual machines on the cluster node to other nodes in the cluster when at all possible.

This feature is turned on by default, but it can be disabled through PowerShell.

Active Directory–Detached Clusters Previous versions of Windows Failover Clustering have depended on Active Directory to provide computer objects for the cluster name object as well as virtual computer objects. With Active Directory–detached failover clusters, communication to the cluster-form clients will use NTLM authentication rather than the normal Kerberos authentication. This is useful in maintaining high availability should a person accidently delete a virtual computer object in Active Directory that a clustered resource depends on for Kerberos authentication.

Dynamic Witness Earlier in this chapter, I mentioned the Dynamic Quorum model and how votes were dynamically adjusted based on the number of nodes in a cluster. In Windows Server 2012 R2, there is a new feature called *dynamic witness* that is enabled by default when the cluster is configured to use a dynamic quorum. Since it is preferred to have an odd number of votes at any one time in a cluster, the dynamic witness will turn on or off the witness vote in order to ensure that there are an odd number of votes in the cluster.

Tie Breaker For 50% Node Split Like the *dynamic witness* feature just described, the Tie Breaker For 50% Node Split option in Windows Server 2012 R2 dynamically adjusts cluster node votes in order to maintain an odd number of votes in a cluster where no witness is being used.

This is useful for a cluster in a multisite, stretched, or geocluster configuration.

Global Update Manager Mode Since the first release of Microsoft Cluster Services appearing in Windows NT 4.0 Enterprise, all nodes in a cluster maintain a local database that keeps a copy of the cluster configuration. The *Global Update Manager (GUM)* is a component of the cluster that ensures that before a change is marked as being committed for the entire cluster, all nodes have received and committed that change to their local cluster database. If one or more nodes do not report back or commit a change, the cluster node is kicked out of being a member of the cluster. Another issue that can occur is that for various clustered applications, such as SQL and Exchange, their performance can be negatively impacted by the time it takes the GUM to coordinate with all the nodes of a cluster for any changes. The GUM is only as fast as the slowest node in the cluster.

With Windows Server 2012 R2, a new feature was added to Failover Clustering called *Global Update Manager mode*. This feature allows you to configure the GUM read-write modes manually in order to greatly speed up the processing of changes by the GUM and to improve the performance of certain clustered resources.

Turn Off IPsec Encryption For Inter-node Cluster Communications In network environments where IPsec is used, slow Group Policy updates and other issues can cause

Active Directory Domain Services to be temporarily unavailable to cluster nodes. If the cluster intracluster communications protocol uses IPsec encryption, then these delays could cause cluster nodes to drop out of the cluster for failure to communicate in a timely manner with the rest of the nodes in the cluster. Windows Server 2012 R2 now provides a way to turn off IPsec encryption on the cluster communication network.

Cluster Dashboard Starting with Windows Server 2012, Failover Clustering supports up to 64 nodes in a cluster. Keeping track of the status and resources on all of these nodes can be an administrative headache! Managing more than one failover cluster and determining what a certain cluster hosts can be painful as well. Fortunately, in Windows Server 2012 R2, the *Failover Cluster Manager*'s main dashboard has been updated to make it easier to see the status and health of multiple clusters.

Hyper-V Replica Broker Starting with Windows Server 2012, Hyper-V supported continuous replication of virtual machines to another server or cluster for disaster recovery purposes. The Hyper-V Recovery Broker allows for virtual machines in a cluster to be replicated. The Hyper-V Recovery Broker keeps track of which cluster nodes virtual machines are residing on and ensures that replication is maintained.

Hyper-V Manager Integration into Failover Cluster Manager In Windows Server 2012 R2, the Hyper-V Management Console is integrated with Failover Cluster Manager for managing virtual machines that are clustered. Normal Hyper-V operations such as configuring, exporting, importing, configuring replication, stopping, starting, and live migrating virtual machines are supported directly through Failover Cluster Manager.

Virtual Machine Monitoring Starting with Windows Server 2012, Failover Clustering now supports Virtual Machine Monitoring for Windows Server 2012/2012 R2 virtual machines. Virtual Machine Monitoring monitors administrator-selected Windows services running within a virtual machine and will automatically restart a service if it should fail. If the service does not start for the configured number of restart attempts, the virtual machine will fail over to another node and then restart. For example, you can configure Failover Clustering to monitor the Print Spooler service on a Windows Server 2012 R2 virtual machine. If the Print Spooler service goes offline, then the cluster will attempt to restart the Print Spooler service within the virtual machine. If the service still fails, Failover Clustering will move the virtual machine to another node.

Summary

High availability is more than just clustering. It is achieved through improved hardware, software, and processes. This chapter focused on how to configure Failover Clustering and Network Load Balancing in order to achieve high availability and scalability.

High availability should be approached through proper hardware configuration, training, and operational discipline. Failover clustering provides a highly available base for many applications, such as databases and mail servers.

Network load-balanced clusters are used to provide high availability and scalability for network-based applications, such as VPNs and web servers. Network load balanced clusters can be configured with any edition of Windows Server 2012 R2 except for the Windows Server 2012 R2 Hyper-V Edition.

Exam Essentials

Know how to modify failover and failback settings. These settings are set on the clustered service or application, but they can be modified by settings on the resources.

Know the hardware requirements for Failover Clustering and Network Load Balancing. Failover clustering and Network Load Balancing have distinct hardware requirements. Know the differences.

Review Questions

1. Which of the following editions of Windows Server 2012 R2 can be configured in a failover cluster? (Choose all that apply.)
   - **A.** Windows Server 2012 R2 Hyper-V edition
   - **B.** Windows Server 2012 R2 Standard edition
   - **C.** Windows Server 2012 R2 Foundation edition
   - **D.** Windows Server 2012 R2 Datacenter edition

2. Which of the following editions of Windows Server 2012 can be configured in a Network Load Balancing cluster? (Choose all that apply.)
   - **A.** Windows Server 2012 R2 Essentials edition
   - **B.** Windows Server 2012 R2 Standard edition
   - **C.** Windows Server 2012 R2 Hyper-V edition
   - **D.** Windows Server 2012 R2 Datacenter edition

3. What is the maximum number of nodes that can participate in a Windows Server 2012 failover cluster?
   - **A.** 2
   - **B.** 4
   - **C.** 16
   - **D.** 64

4. Which of the following actions should be performed against an NLB cluster node if maintenance needs to be performed while not terminating current connections?
   - **A.** Evict
   - **B.** Drainstop
   - **C.** Pause
   - **D.** Stop

5. What is the maximum number of nodes that can participate in a Windows Server 2012 R2 NLB cluster?
   - **A.** 4
   - **B.** 8
   - **C.** 16
   - **D.** 32

6. Which of the following applications would be better suited on a failover cluster instead of a network load-balanced cluster? (Choose all that apply.)
   - **A.** SQL Server
   - **B.** Website
   - **C.** Exchange Mailbox Server
   - **D.** VPN services

7. Which of the following applications would be better suited on a Network Load Balancing cluster instead of a failover cluster? (Choose all that apply.)
   - **A.** SQL Server
   - **B.** Website
   - **C.** Database servers
   - **D.** Terminal Services

8. To configure an NLB cluster with unicast, what is the minimum number of network adapters required in each node?
   - **A.** One
   - **B.** Two
   - **C.** Three
   - **D.** Six

9. In a four-node cluster set to a Node And File Share Majority quorum model, how many votes can be lost before quorum is lost?
   - **A.** One
   - **B.** Two
   - **C.** Three
   - **D.** Four

10. In a three-node cluster set to a Node Majority quorum model, how many cluster nodes can be offline before quorum is lost?
    - **A.** Zero
    - **B.** One
    - **C.** Two
    - **D.** Three