CHAPTER 11

NETWORK OPERATING SYSTEM FUNDAMENTALS

After reading this chapter and completing the exercises, you will be able to:

Explain the major components of an OS, including the file system, processes, and the kernel

Describe client and server operating systems and compare client and server OSs

Describe the components of virtualization and virtualization products

Plan for an OS installation and perform postinstallation tasks

From a user's standpoint, the network operating system is the focal point of a network. A computer's OS is what users interact with when accessing a network's resources. Indeed, operating systems hide the details of network access so well that users often realize they're connected to a network only when access to network resources fails. As a network administrator, you know the network operating system (NOS) is only one piece of the network puzzle, but you're likely to spend quite a bit of time with this piece because of all the network-specific services you must install and configure.

This chapter discusses components common to almost every OS, networked or not, and then describes specific network services that network operating systems provide. In addition, this chapter introduces virtualization, a technology that's integral to most medium-sized and large networks, and even many small networks. Finally, the specifics of installing Windows Server 2019 and Linux are described.

Table 11-1 summarizes what you need for the hands-on projects in this chapter.

Table 11-1 Hands-on project requirements

Hands-on project	Requirements	Time required	Notes
Hands-On Project 11-1: Navigating the Windows File System from the Command Prompt	Net-*XX*	20 minutes	
Hands-On Project 11-2: Navigating the Linux File System	Net-*XX*	20 minutes	Need a Linux Live disk or a computer with Linux already installed
Hands-On Project 11-3: Using Windows Task Manager	Net-*XX*	10 minutes	
Hands-On Project 11-4: Displaying Linux Processes	Net-*XX*	10 minutes	Need a Linux Live disk or a computer with Linux already installed
Hands-On Project 11-5: Mapping a Drive Letter	Net-*XX*	10 minutes	
Hands-On Project 11-6: Creating and Connecting to a Shared Printer	Net-*XX*	10 minutes	A computer with a shared printer to which all student computers can connect, or students can connect to each other's shared printers
Hands-On Project 11-7: Downloading and Installing VMware Workstation Player	Net-*XX*	20 minutes	Internet access
Hands-On Project 11-8: Creating a Virtual Machine in VMware Workstation Player	Net-*XX*	15 minutes	
Hands-On Project 11-9: Installing Windows Server 2019 in a VM	Net-*XX*	30 minutes or longer	The Windows Server 2019 ISO file (evaluation) downloaded from the Microsoft evaluation center

Operating System Fundamentals

A computer's OS provides a convenient interface for users and applications to access computer hardware components. It controls access to memory, CPU, storage devices, and external input/output (I/O) devices, such as printers, webcams, and scanners. To understand the network services and functions a contemporary OS provides, you

need a solid grasp of how an OS manages its local resources. The following sections expand on OS concepts that were introduced in Chapter 1:

- File systems
- Processes and services
- Kernel

The File System

A **file system** is the method by which an OS stores and organizes files and manages access to files on a storage device, such as a hard drive. File systems differ in how they allocate space for files, how files are located on a disk, what level of fault tolerance is built into the system, and how access to files is secured. Regardless of how these tasks are accomplished, contemporary file systems have the following objectives:

- Provide a convenient interface for users and applications to open and save files.
- Provide an efficient method to organize space on a drive.
- Provide a hierarchical filing method to store files.
- Provide an indexing system for fast retrieval of files.
- Provide secure access to files for authorized users.

User Interface

When a user double-clicks a file to open it, the user interface calls the file system with a request to open the file. The file type determines exactly how the file is opened. If the file is an application, the application is loaded into memory and run by the CPU. If the file is a document, the application associated with the document type is loaded into memory and opens the file. For example, on Windows computers, if you double-click a Budget.xlsx file, the Excel application is loaded into memory and then opens the file. If a user creates a file or changes an existing file and wants to save it, the application calls the file system to store the new or changed file on the disk. Most users of an OS interact with the file system by using a file manager program, such as File Explorer in Windows. As a future computer or network professional, you need to have a deeper understanding of how a file system works so that you can make informed choices when you need to install a file system or troubleshoot file system-related problems.

Disk Drive Space Organization

The storage space on a disk drive is divided into manageable chunks called "sectors." On most disk drives, each sector is 512 bytes. To make storage processing more efficient, sectors are grouped to make a disk cluster (also called a "block"). A **disk cluster** is the smallest amount of space that can be occupied by a file stored on the disk. For example, if you have a file system that groups four sectors to make a cluster, each cluster is 2048 (2K) bytes. So, if you store a file that's 148 bytes, it occupies one cluster of 2048 bytes, which wastes 1900 bytes of storage. The waste occurs because

no other file can occupy any part of a cluster already occupied by another file. If you store a file that's 10,000 bytes, it occupies five clusters, with about 240 bytes of unused, wasted space.

Note

Some file systems use optimization techniques to reduce wasted file space by allowing files to occupy unused sectors that are part of a cluster already in use.

You might think that having a smaller cluster size is the optimal way to organize your disk. This is true in some cases, but only when mostly small files are stored on the disk because data is read from and written to the disk in cluster-sized chunks. So, the smaller the clusters, the more read/write operations are required when using a file. Each read/write operation takes time, and the more operations that are required, the slower the system runs. If you store mostly large files on the disk, a larger cluster size usually results in better performance because fewer read/write operations need to be performed. In addition, smaller cluster sizes can lead to a fragmented disk, in which files are spread out all over the disk instead of being stored in consecutive locations. Fragmentation causes many more disk seek operations, which slow down file access. Recall from Chapter 1 that the disk seek time is the amount of time required to move a drive's read/write heads to the correct position on disk platters to read or write clusters.

A disk's cluster size is selected when the disk is formatted. Most OSs set the cluster size to a medium value by default; for example, Windows sets the cluster size on NTFS-formatted disks to 4 KB. However, if you know you will be storing many files under 2 KB, choose a smaller cluster size when you format to reduce wasted space. If you know you will store mostly files larger than 16 KB, choose a larger cluster size.

The formatting process groups sectors into clusters and maps all disk clusters for fast access. In addition, clusters are marked as unused. When you format a disk that contains files, the data is actually still there, but the file system can no longer access the data because the file system doesn't know how to find it. Third-party disk recovery programs can often recover data from a formatted disk by bypassing the file system and reading the data in each cluster.

Hierarchical Filing Method

Most file systems organize files in a hierarchy of folders or directories; the top of the hierarchy is called the "root" of the file system. ("Directory" is an older term for "folder" but is still used, particularly when discussing Linux file systems; however,

the term "folder" is generally used in this book.) The root of the file system often represents a disk drive or other mass storage device, such as a flash drive. Off the root of the file system can be files and folders, with folders containing files and additional folders usually referred to as "subfolders." To navigate the file system and see its hierarchy with a GUI tool, such as File Explorer, users simply double-click folders and subfolders to open them and view their contents. Figure 11-1 shows a logical diagram of a typical Windows file system.

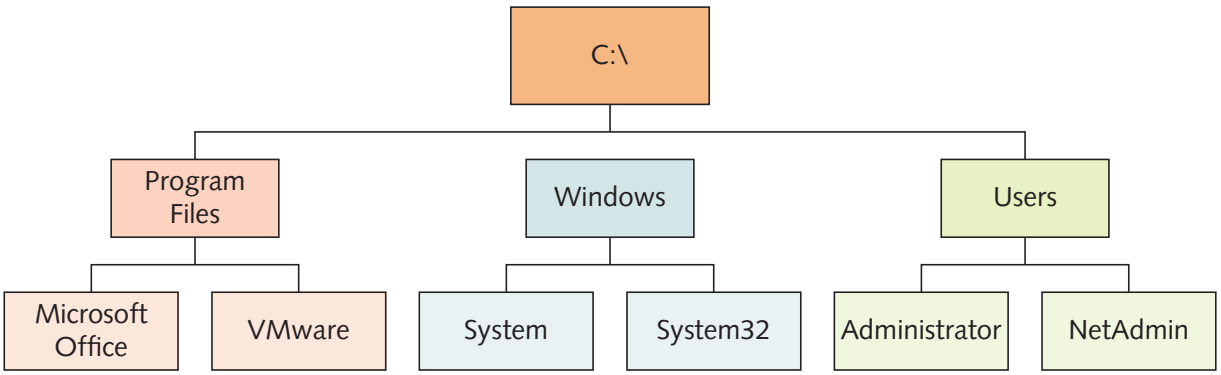

Figure 11-1 A hierarchical file system

Navigating the file system from a command prompt is a different proposition. Because users must enter the exact syntax with no typos, those who began using computers when a GUI was the standard user interface are often frustrated when trying to navigate the file system at the command prompt. However, mastering the command prompt (or shell prompt in Linux) is a valuable skill to have as a Windows administrator and essential for Linux administrators.

Note

Commercial routers and switches are often configured with their own command-line user interfaces, so skill at the command line extends beyond working with desktop and server OSs.

File-Indexing System

With today's large disk drives, more files can be stored on them, making it harder for users to find the files they need. To help solve this problem, most file systems include an indexing system that enables users to search for a file based on all or part of its filename or its contents. Users don't have to know where to go in the file system's hierarchy to find a stored file they need; they just need to know some or all of the filename or some keywords in the file. The indexing system maintains a database

that's updated as files are created, modified, and deleted. File-indexing systems can take considerable processing power, as they usually work in the background to monitor the file system and update the index as changes are made. However, they're usually configured to wait until the system is idle before performing operations. In Windows, you can configure aspects of the file system's index by clicking Indexing Options in Control Panel. Then you can select which types of files are indexed, which volumes and folders should be indexed, and whether file contents as well as file properties should be indexed. You can also change the default location of the index database, which is C:\ProgramData\Microsoft.

Secure Access to Files

Computers are often shared at home or in the workplace. Each user might want to maintain a separate set of files and documents that other users can't access. In addition, files on a computer that's part of a network are potentially accessible to all users on the network. A file system's access controls, or permissions, can be used to allow only authorized users to access certain files or folders. In addition, access controls can be used to secure OS files from accidental corruption or deletion. Not all file systems have access controls, but the ones installed by default on current OSs do. Notably, the NTFS file system in Windows supports file and folder permissions, as do Linux file systems. The older FAT16 and FAT32 file systems don't support file and folder permissions, so any user logged on to the system console has full access to all files stored on these file systems. The details of how these file systems work are beyond the scope of this book, but you learn more about using permissions in Windows and Linux in Chapter 12.

Processes and Services

A **process** is a program that's loaded into memory and run by the CPU. It can be an application a user interacts with, such as a word-processing program or a Web browser, or a program with no user interface that communicates with and provides services to other processes. This latter type of process is usually called a **service** in Windows and a "daemon" in Linux and is said to run in the background because there's no user interface. Examples of services include Client for Microsoft Networks and File and Printer Sharing for Microsoft Networks, which provide the client and server sides of Windows file sharing. Many TCP/IP Application-layer protocols, such as DNS and DHCP, also run as services.

Tip

Some OSs refer to processes as "tasks."

Network services are important because they allow your computer and applications to perform tasks they otherwise couldn't, or would need additional built-in functionality to handle. For example, a Web browser is designed to request Web pages from a Web server and display them. However, because most people use the Web server's name rather than its address, a name lookup is required before a Web browser can do its main job. If it weren't for the DNS client service running on the computer, the Web browser would have to know how to perform DNS functions. Instead, as you learned in Chapter 5, the Web browser simply sends the DNS service a request for a name lookup, and DNS returns the IP address to the Web browser. The same is true of almost every network application. Most network access is initiated by using the server name, and DNS is always running as a process to provide the name lookup service, so the application is free to do what it was designed to do. In Windows, you can use a handy tool called Task Manager to see processes and services running on your computer, check how much CPU time and memory each process is using, and stop a process from running, if necessary. In Linux, you use the System Monitor application for these tasks.

An OS can run 2, 10, 100, or more processes seemingly at once by using multitasking. **Multitasking** is an OS's capability to run more than one application or process at a time. It's what allows you to listen to a music file while browsing the Web, for example. Whether a computer has one or multiple CPUs, it multitasks by using a method called **time slicing**, which occurs when a CPU's computing cycles are divided between more than one process. Each process receives a limited number of processor cycles before the OS suspends it and activates the next process. The act of changing to another process is called **context switching**.

When a process has work to do, as when a user types at the keyboard or a Web browser submits a request for a Web page, the CPU is notified. If it already has other processes waiting, the new request is put into a queue. If this process has a higher priority than others in the queue, it jumps to the front of the line. Because a CPU can execute many billions of instructions per second, processes waiting in the queue are usually scheduled to run quickly. This activity is perceived as many applications operating simultaneously because each time slice is a very short period. People can't distinguish instances of such a brief time period, so there's an illusion that the CPU and OS are performing several tasks at once. There are two types of multitasking:

- *Preemptive*—With **preemptive multitasking**, the OS controls which process gets access to the CPU and for how long; when the assigned time slice expires or a higher-priority task has work to do, the current process is suspended, and the next process gets access to the CPU.
- *Cooperative*—With **cooperative multitasking**, the OS can't stop a process; when a process gets control of the CPU, it maintains control until it satisfies its computing needs. No other process can access the CPU until the current process releases it.

Cooperative multitasking was used in older OSs, such as Windows 3.1. An application that stopped working because of an infinite loop could bring the entire system to a screeching halt because it never gave up control of the CPU. Thankfully, all current OSs use preemptive multitasking, so the OS or the user can terminate misbehaving applications.

Many applications are now designed so that different parts can be scheduled to run separately, almost as though they were different processes. Each part that can be scheduled to run is called a **thread**, which is the smallest unit of software that can be scheduled. A **multithreaded application** has two or more threads that can be scheduled separately for execution by the CPU. For example, a multithreaded word-processing program might have one thread that waits for keyboard entry and then formats and displays the characters as they're typed, and another thread that checks the spelling of each word as it's typed.

A multithreaded application benefits most when the OS and hardware support **multiprocessing**, which allows performing multiple tasks or threads simultaneously, each by a different CPU or CPU core. All current OSs support multiprocessing. Windows 10 supports up to two physical CPUs; a physical CPU is a chip installed in a socket on the motherboard. This means Windows 10 supports two CPUs, but each CPU can have one, two, four, or more cores. Windows Server OSs support up to 64 CPUs, depending on the edition. Most Linux OSs can support up to 32 or more CPUs.

The Kernel

If the CPU is the brain of a computer, the kernel is the office manager of the OS. Just as an office manager schedules everyone and everything and manages office resources, the kernel schedules processes to run, making sure high-priority processes are taken care of first; manages memory to ensure that two applications don't attempt to use the same memory space; and makes sure I/O devices are accessed by only one process at a time, in addition to other tasks. Because the kernel performs these important tasks, its efficiency and reliability are paramount to the OS's overall efficiency and reliability. Of course, the kernel is a process like any application, but it has the highest priority of any process, so when it needs to run, it takes precedence. You can't view it in Task Manager, and you certainly can't stop it, or the whole system would come crashing down.

Operating systems are designed in layers, as network protocols are, and the kernel is usually shown as the layer just above the hardware. This structure means nothing goes in or out without passing through the kernel—or at least without the kernel's approval. Figure 11-2 is a simplified illustration of the Windows OS structure, with the kernel near the bottom of the stack and above the hardware.

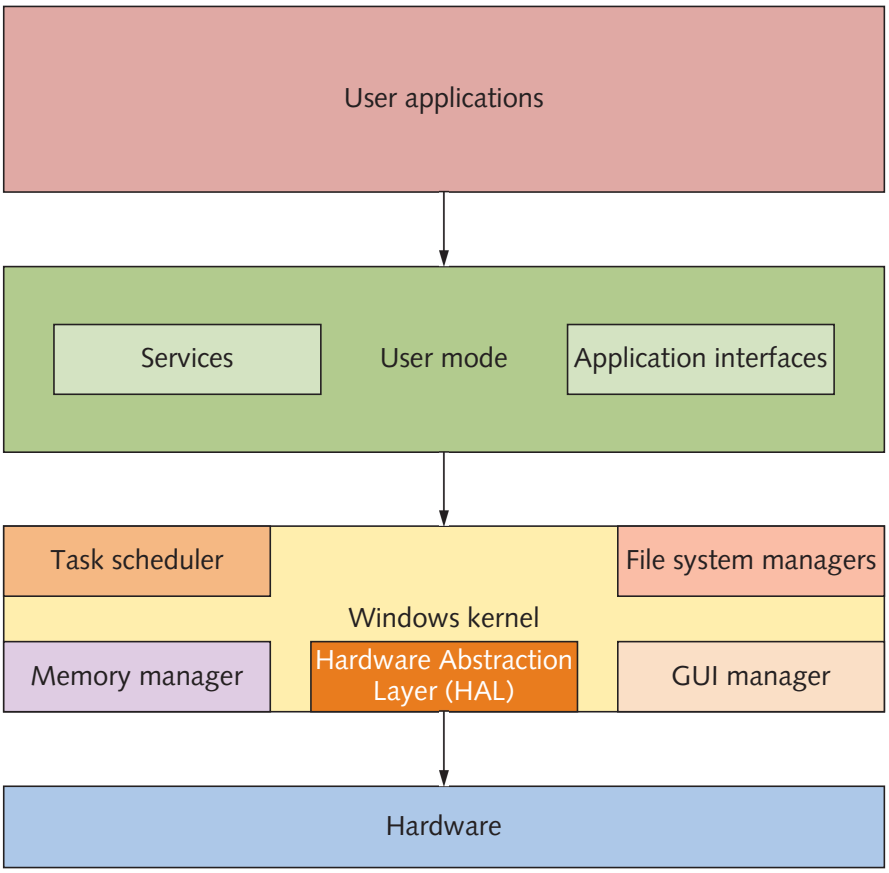

Figure 11-2 The Windows OS structure

Hands-On Project 11-1: Navigating the Windows File System from the Command Prompt

Time Required: 20 minutes
Objective: Navigate the Windows file system from the command prompt.
Required Tools and Equipment: Net-*XX*
Description: In this project, you open a command prompt window and practice navigating the Windows file system from the command prompt.

1. Log on to your computer as **NetAdmin**.
2. Open a command prompt window. The prompt indicates your current file system context. Usually, when you open a command prompt window, you're placed in the file system context of your user profile. For example, if you log on as NetAdmin, the prompt is `C:\Users\NetAdmin>`. In this prompt, `C:` indicates the drive letter, and `\Users\NetAdmin` indicates the path. Type **`dir`** and press **Enter** to see the list of files in this folder (see Figure 11-3). The `dir` command displays the same information as the NetAdmin folder in File Explorer (see Figure 11-4).

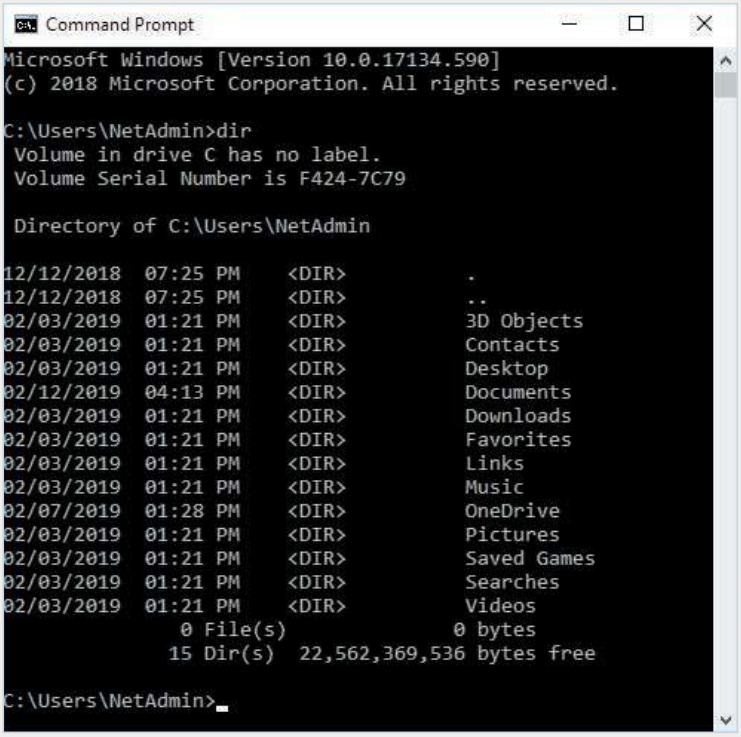

Figure 11-3 Viewing the files in C:\Users\NetAdmin

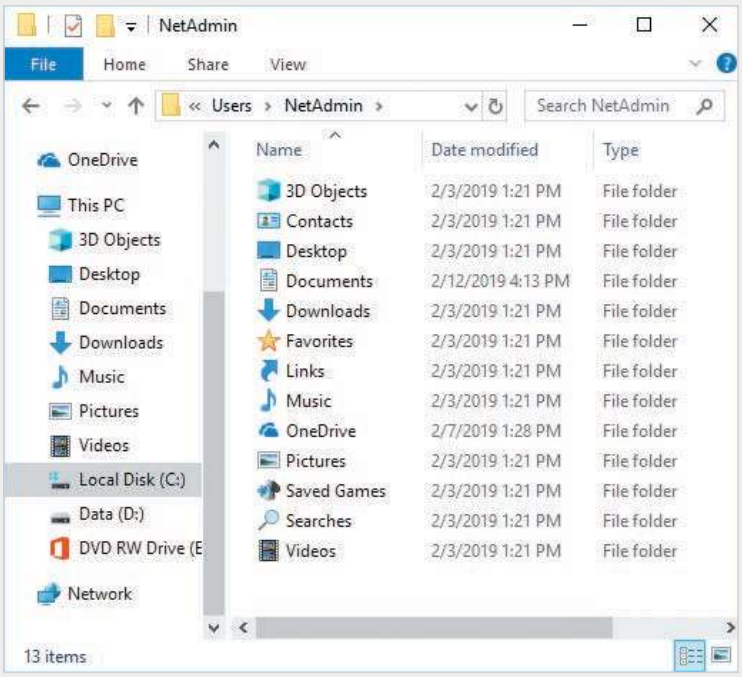

Figure 11-4 Viewing the contents of C:\Users\NetAdmin in File Explorer

3. In Windows file systems, the backslash (\) has two meanings. At the beginning of a path, it indicates the root or top of the file system. Anywhere else, it's used as a separator between folders, subfolders, and files. In Windows, the forward slash (/) is used in many command-line programs to denote options for the command. The `dir` command means "directory," which is the term used before Windows started using the word "folder"; the command lists the files and subfolders in the folder. As in File Explorer, the `dir` command doesn't display hidden files. To see hidden files, type **`dir /ah`** and press **Enter**. The `/ah` option tells `dir` to display files with the hidden attribute set. Type **`dir /a`** and press **Enter** to see all files. To see more options for the `dir` command, type **`dir /?`** and press **Enter**.

Tip

Remember to enter a space before any options you add to a command. Although not all commands require a space, many do, so it's best to get in the habit of entering one after the command.

4. To move to the root of the file system, type **`cd \`** and press **Enter**. The `cd` command means "change directory." Your prompt should now be `c:\>`. Type **`dir`** and press **Enter**. To go to C:\Windows\System32, type **`cd \windows\system32`** and press **Enter**. Notice that the prompt changes to `C:\Windows\System32>`. Type **`dir`** and press **Enter**. Several files scroll by quickly. To view them page by page, type **`dir /p`** and press **Enter**. (The `/p` option paginates the output.) Press any key to see the next page of files, or press **Ctrl**+**C** to terminate the output if you don't want to page through all the files.
5. Navigate back to the root of the file system. If you have more than one drive, you can switch drives by typing the drive letter and a colon and pressing Enter. For example, if you have a D drive, type **`D:`** and press **Enter**. The prompt changes to `D:\>`. If you don't have a D drive, you get an error stating that the drive can't be found. Type **`C:`** and press **Enter** to get back to the C drive, if necessary.
6. Next, create a folder by typing **`mkdir TestDocs`** and pressing **Enter**. The `mkdir` command means "make directory." To verify that the folder was created, type **`dir`** and press **Enter**, and then go to the new folder by typing **`cd TestDocs`** and pressing **Enter**. (*Note*: In Windows file systems, capitalization of filenames is ignored, so `TestDocs` is the same as `testdocs`; however, filenames are case sensitive in Linux.)
7. To create a subfolder, type **`mkdir SubDocs1`** and press **Enter**. Change to this subfolder by typing **`cd SubDocs1`** and pressing **Enter**. To go back to the TestDocs folder, type **`cd \TestDocs`** and press **Enter**. You must include the \ character because you're telling the file system that TestDocs is located directly under the root. To navigate to the SubDocs1 subfolder again, type **`cd SubDocs1`** and press **Enter**. You don't use the

\ character in this command because SubDocs1 is located directly under your current location. To go up one level in the file hierarchy, use the `..` notation: Type **`cd ..`** and press **Enter**, which takes you to the TestDocs folder. Type **`cd ..`** and press **Enter** again to get to the root.

8. Sometimes folder names are long and easy to mistype, so using a shortcut can be handy. Type **`cd test`** and press **Tab**. If TestDocs is the only folder name starting with "Test," the command prompt fills in the rest of the name for you. If more than one folder begins with "Test," the command prompt displays the first one in alphabetical order. Pressing Tab repeatedly cycles through all folders beginning with "Test." Press **Enter**.
9. The command prompt maintains a history of commands you've used since the window has been open. If you've been entering long commands that you need to repeat, you can scroll through the history by pressing the up arrow. Press the **up arrow** repeatedly to scroll through your recent commands. Press **Esc** when you're finished to cancel the command.
10. Type **`mkdir ARealLongFolderName`** and press **Enter**. Next, you make a mistake on purpose: misspelling the folder name. Type **`cd ARealLongFoldName`** (omitting the "er" in "Folder") and press **Enter**. You see the message "The system cannot find the path specified." To correct this error, press the **up arrow**. Press the **left arrow** until the cursor is under the "N" in Name. Type **`er`** and press **Enter**. Making a correction in this fashion is called "command-line editing."
11. To create a text file in your current folder, type **`notepad myfile.txt`** and press **Enter**. When prompted to create the file, click **Yes**. Type whatever you like in the file, and then open the **File** menu and click **Exit**. When prompted to save the file, click **Save**. Type **`dir`** and press **Enter** to verify that the file exists. To rename it, type **`ren myfile.txt newfile.txt`** and press **Enter**. To copy the file, type **`copy newfile.txt newfile1.txt`** and press **Enter**. Type **`ren newfile.txt newfile.old`** and press **Enter**. Press the **up arrow** until you see the `dir` command, and then press **Enter**.
12. To view only files with a .txt extension, type **`dir *.txt`** and press **Enter**. To see all files starting with "new," type **`dir new*`** and press **Enter**. To delete newfile.old, type **`del newfile.old`** and press **Enter**. To delete all files in the ARealLongFolderName folder, type **`del *`** and press **Enter**. Type **`y`** and press **Enter** when prompted. Press the **up arrow** until you see the `dir` command, and then press **Enter** to verify that all the files are deleted.
13. Write the commands to create a folder named **NewFolder**, move to this folder, and then list all files with the .doc extension:

    _____

    _____

    _____

14. This project has shown you the basics of using the command line to navigate the file system in Windows. As a future network administrator, you'll find yourself using the command line often. Close the command prompt window and log off Windows for the next project.

Hands-On Project 11-2: Navigating the Linux File System

Time Required: 20 minutes
Objective: Learn basic navigation of the Linux file system with the GUI and command line.
Required Tools and Equipment: A computer with Linux installed or Net-*XX* and a Linux Live disk (the Fedora Workstation 29 Live image is used in this project, but others can be used)
Description: In this project, you use the Linux GUI and command line to navigate the file system. Linux distributions use different graphical desktop interfaces, so you might need to adjust the instructions in this project to accommodate the Linux distribution and version you're using.

1. Start Linux. If you're using the Fedora Workstation Live image, press **Enter** to boot right away or wait until Fedora boots automatically. After Fedora boots, click **Try Fedora** and then click **Close**.
2. Click **Activities** at the top of the window. To begin navigating the file system, click the **Files** icon (it looks like a file cabinet) to open a file browser window that's similar to File Explorer. The Files application opens to your Home directory.
3. To navigate to the root of the Linux file system, click **+ Other Locations** in the left pane and then click **Computer** in the right pane (see Figure 11-5). You're now at the root of the file system.

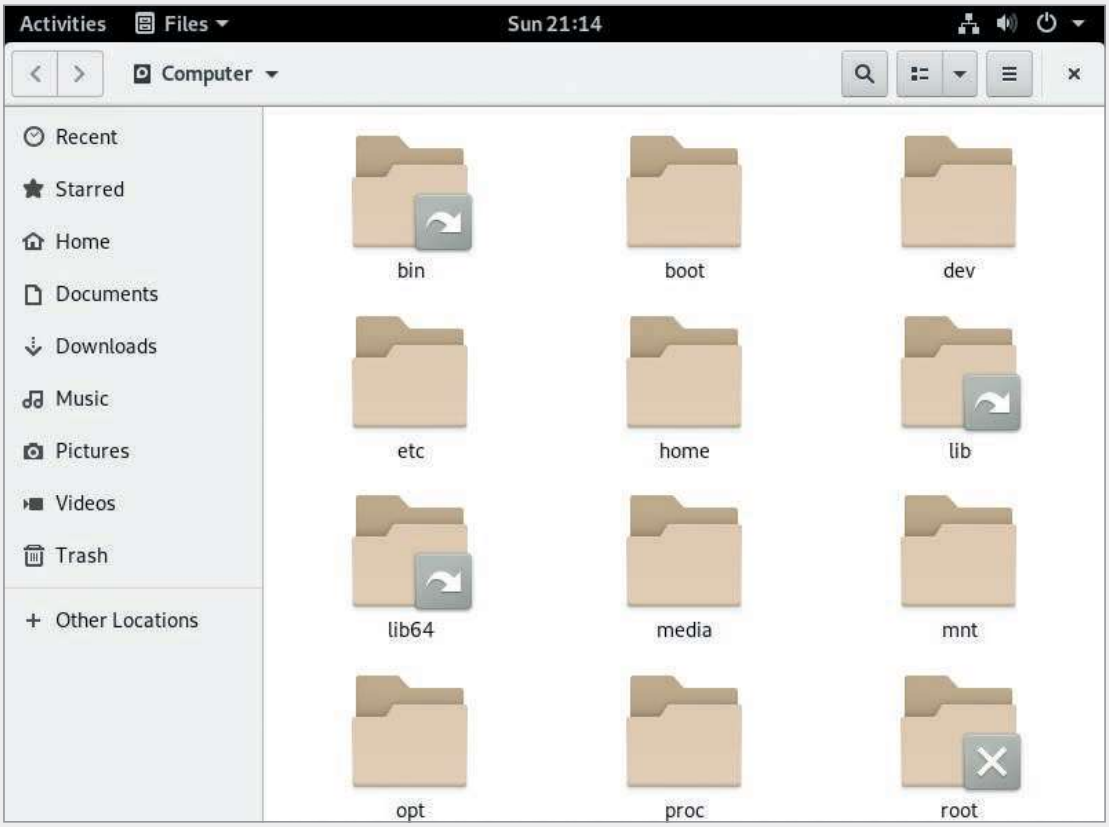

Figure 11-5 The Fedora desktop with the file browser open
Source: The Linux Foundation

Note

Remember that the file system in Linux doesn't use drive letters. All available drives are accessed as folders off the root of the file system, which in Linux is designated with a forward slash (/).

4. In the right pane, double-click the **home** folder. Home folders for all users on the system are in this folder. If you're using Fedora Live, you see a folder named liveuser, which is the user you're currently logged on as. Double-click **liveuser** (or the home folder for your installation and logon account). You see folders similar to the ones created for each user in Windows, such as Desktop, Documents, and Downloads. Close the file browser window by clicking the **X** in the upper-right corner.
5. To open a command prompt window (called a "terminal window" or "shell prompt" in Linux), click **Activities**, then type **`terminal`** in the search box. Click the **Terminal** icon shown in the search results. The prompt in a terminal window is different from the Windows command prompt. In Figure 11-6, the prompt is `[liveuser@localhost-live ~]$`. In the prompt, `liveuser` is the username of the currently logged-on user, and `localhost` after the `@` is the computer name, which is `localhost` by default. The tilde (`~`) following the computer name indicates the user's home folder, and `$` is the end of the prompt. When you open a Linux terminal window, you're usually placed in your home folder.

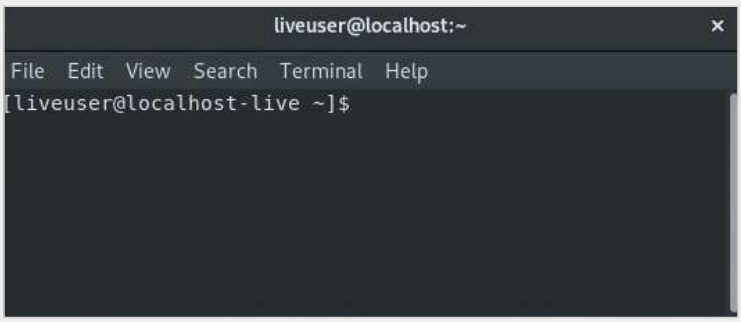

Figure 11-6 A Linux terminal window

Source: The Linux Foundation

6. Besides the terminal prompt, there are other ways to see the logged-on user's name and the computer name. To see your username, type **`whoami`** and press **Enter**. To view the computer name, type **`hostname`** and press **Enter**.
7. To move to the root of the file system, type **`cd /`** and press **Enter**. Remember that Linux uses forward slashes, and Windows uses backslashes.

Tip

If you type a backslash accidentally, Linux thinks you're continuing the command on the next line and displays a > prompt. If this happens, press Enter to get back to the normal prompt.

8. To view a list of files in the root, type **`ls`** (which means "list") and press **Enter**. You can get back to your home folder in three ways. Type **`cd /home/liveuser`** and press **Enter** (replacing `liveuser` with your logon name if you aren't using Fedora Live), and then type **`cd /`** and press **Enter** again. A second way to get to the home folder is to type **`cd`** and press **Enter**, and a third way is to type **`cd ~`** and press **Enter**.
9. Create a folder in your home folder by typing **`mkdir newfolder`** and pressing **Enter**. Type **`ls`** and press **Enter** to verify that the new folder has been created.
10. You see other folders in your home folder, including one named Documents. Type **`cd documents`** and press **Enter**. Typing `documents` with a lowercase "d" causes an error because the Linux file system is case sensitive, so `Documents` is different from `documents`. Press the **up arrow** to repeat the command. Press the **left arrow** until the cursor is over the "o" in `documents` and press **Backspace**. Type **`D`** and press **Enter**.
11. To create an empty file, type **`touch newfile`** and press **Enter**. Verify that the file was created.
12. Type **`gedit newfile`** and press **Enter** to open newfile in gedit, a Notepad-like editor. Type whatever you like, click **Save**, and then close the gedit window.
13. To view newfile's contents from the command line, type **`cat newfile`** and press **Enter**. With a long file, you can use the `more` command to paginate the output. Type **`more /etc/protocols`** and press **Enter** to see a long file. Press the **spacebar** to page through the file, and type **`q`** to quit.
14. If you can't remember the complete name of a command, you can use the Tab key to perform command completion. If the command can be found by using the letters you typed, Linux completes the command. If there's more than one match, press Tab a second time to display all matches. Type **`ge`** and press **Tab** twice. A list of commands beginning with "ge" is displayed. Press **backspace** twice to delete `ge`, and then type **`cd ..`** and press **Enter** to move back one folder. Type **`ls Doc`** and press **Tab**. Linux completes the command. Press **Enter**. You see newfile that you created.
15. Type **`cd Doc`**, press **Tab**, and then press **Enter**. The `mv` command is used to rename or move files. To rename newfile to oldfile, type **`mv newfile oldfile`** and press **Enter**.

16. The `rm` command means "remove." To remove the file you just renamed, type **`rm oldfile`** and press **Enter**. Type **`ls`** and press **Enter** to see that there are no files.
17. Write two commands to get to the home directory in Linux:

    _____

    _____

18. To shut down Linux from the terminal, type **`shutdown -h now`** and press **Enter**. The `-h` option tells Linux to shut down and halt, and the `now` option means "do it now." (You use `-r` to restart the computer.) You can also specify a number of minutes to delay before the system shuts down by using `+m` instead of `now`, replacing *m* with the number of minutes to delay. If you get the message "Must be root," you have to be an administrator to shut down the computer. The root user is the default administrator account in Linux. If you see the "Must be root" message, type **`sudo shutdown -h now`** and press **Enter**. The `sudo` command means "do this as superuser." With the Fedora Live CD, you will not have to use the `sudo` command.

Hands-On Project 11-3: Using Windows Task Manager

Time Required: 10 minutes
Objective: Use Windows Task Manager to view running processes, services, and real-time performance.
Required Tools and Equipment: Net-*XX*
Description: In this project, you use Task Manager to view processes and services.

1. Log on to your computer as **NetAdmin**.
2. Start Task Manager by right-clicking the taskbar and clicking **Task Manager**. You see a basic view of Task Manager that shows any currently running apps. If you have no running apps, you see the message "There are no running apps." Click the Search text box on the taskbar, type **notepad**, and press **Enter** to start Notepad. Switch to the Task Manager window to see Notepad in the list of running apps.
3. Click **More details** to see a more detailed view (see Figure 11-7). Right-click **Notepad** in Task Manager and click **Go to details**. The Processes tab opens, and the notepad.exe process is selected.

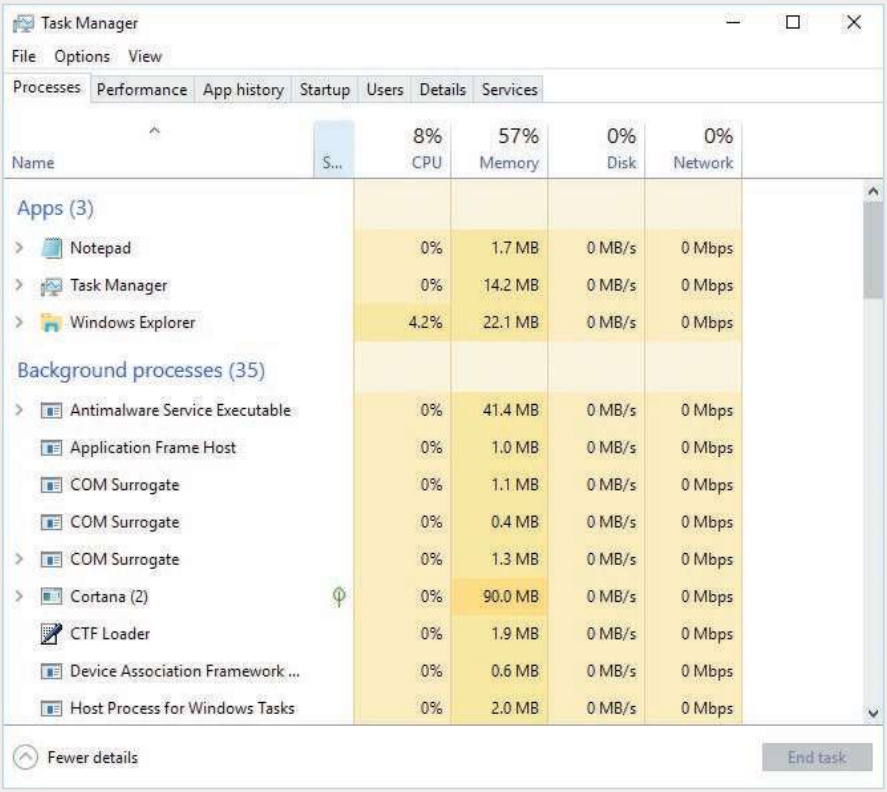

Figure 11-7 The Task Manager Processes tab

4. To sort running processes by the percentage of CPU time they're using, click the **CPU** column. If necessary, scroll to the top of the window. You'll probably see System Idle Process at the top, and its CPU percentage will be in the high 90s. When a Windows OS has no real work to do, the kernel schedules the System Idle Process with CPU time. When the CPU percentage for System Idle Process is a high value, the computer isn't very busy; if it's a low value, other processes are using the CPU a lot.
5. Right-click the **CPU** column and click **Select columns**, and then click the **CPU time** check box. This column shows you the total CPU time a process has used since it was started. Next, scroll down and click **Command line**. This column shows you the name of the actual command that loaded the process. Click **OK**.
6. Click the **CPU time** column to sort entries by overall amount of CPU time in *hours*:*minutes*:*seconds* format. Notice that several processes are named svchost.exe, which is used to start many services in Windows. The Command line column gives you a better idea of which service each svchost.exe entry refers to and can also help you track down what application a process belongs to, based on the path to the application.
7. Find one of the svchost.exe entries ending with "LocalSystemNetworkRestricted" in the Command line column. Right-click the **svchost.exe** entry and click **Go to Service(s)**. The Services tab opens, and the services started by the process are highlighted. Scroll through the Services tab to see the highlighted services.

8. In the Services tab, click the **Status** column to sort services by status. Running services are listed first, and stopped services are listed next. Scroll through the services to see how many are running. Clearly, a lot is going on behind the scenes on your computer.
9. Click and then right-click the **Dhcp** service, and then click **Go to details**. The Details tab opens, and the svchost.exe process that started the service is highlighted.
10. Click the **Performance** tab. The CPU box at the top left shows the total percentage of the CPU currently being used, and on the right is a line graph showing a one-minute history of CPU use. Figure 11-8 shows the Performance tab, which includes boxes for CPU utilization, memory use, disk utilization, and network utilization.

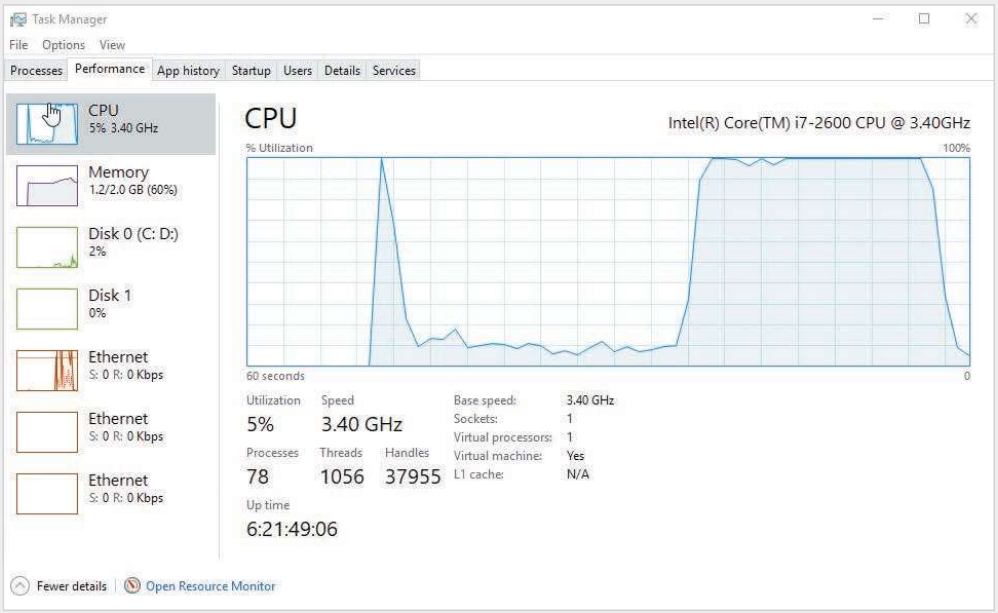

Figure 11-8 The Task Manager Performance tab

11. Click the **Processes** tab. Right-click **notepad.exe** and click **End task** to close Notepad. You can use this feature to close an application that no longer responds to the mouse or keyboard.
12. In the Details tab of Task Manager, what's the difference between the CPU column and the CPU Time column?

_____

_____

13. Close Task Manager, and log off Windows for the next project.

Hands-On Project 11-4: Displaying Linux Processes

Time Required: 10 minutes
Objective: View processes running in Linux.
Required Tools and Equipment: A computer with Linux installed or a Linux Live disk (the Fedora Workstation 29 Live image is used in this project, but others can be used)
Description: In this project, you use Linux System Monitor and the `ps` command to view processes and performance information.

1. Start Linux, and access the desktop. Click **Activities**, and then click the **Show Applications** icon.
2. Start System Monitor by typing **System Monitor** in the search box and clicking the **System Monitor** icon. Click the **Processes** tab, if necessary, and then click the **% CPU** column to sort from highest to lowest percentage of CPU use. Linux doesn't show a system idle process. System Monitor itself is probably the biggest user of the CPU now.
3. Click the **Settings** icon, and then click **Active Processes** to see only currently active processes. System Monitor is probably the only process shown. Click **My Processes** to show all processes started by the currently logged-on user.
4. Click the **Resources** tab, which shows information similar to the Performance tab in Task Manager.
5. Start Firefox by clicking **Activities** and then the **Firefox** icon (the orange and blue globe). You should see the CPU utilization and network utilization spike higher. Close Firefox and System Monitor.
6. Click **Activities**, click **Show Applications**, and type **Terminal** in the search box. Click **Terminal**. To see all currently running processes, type **`ps -A`** and press **Enter**. Type **`ps -A | less`** and press **Enter** to paginate the output. You can use the up and down arrows and Page Up and Page Down keys to scroll through the output. Type **`q`** to quit.
7. Type **`top`** and press **Enter** to see a real-time view of the top CPU users and other statistics. Type **`q`** to quit.
8. To shut down Linux, type **`shutdown -h now`** and press **Enter**.

Client and Server Operating System Overview

Client OSs, such as Windows 10 and Mac OS X, include many features once reserved for a server OS, such as file and printer sharing and file system security. Indeed, a client OS can perform as a basic server because it has these networking features built in, but an OS designed to be installed as a server still contains many additional networking and fault-tolerance features not found in client OSs. The determining factor of whether you need a server OS or a client OS is what role the computer will play in your network.

As you know, computers in a network usually play one of two roles: a client or a server. Although contemporary OSs allow servers to perform client tasks and clients to

perform server tasks, most vendors have specific versions of their OSs to fulfill these roles. The client version generally comes configured with client software, such as Web browsers, DNS and DHCP clients, and file-sharing clients. Most server versions also include client software but have server components, such as Web servers, DNS and DHCP servers, and file-sharing servers. In addition, advanced server OSs usually include directory services, remote access services, fault-tolerance features, and virtualization.

The Role of a Client Operating System

The client OS is where network users spend all their time. Its purpose is to run applications, which often access network resources. Most desktop computers run a client OS equipped with the following network client software:

- DHCP client
- DNS client
- HTTP client (Web browser)
- File-sharing client
- E-mail client

Other client software can be installed on a client OS, such as the client side of a client/server database application, but these specialized applications are beyond the scope of this book. The preceding list of client software is installed on most OSs and used by most users. The DHCP, DNS, and HTTP clients were discussed in Chapter 5, so this chapter focuses on file-sharing and e-mail clients.

File-Sharing Client

A file-sharing client allows the computer to access files and printers on the network. When a user or an application requests a resource—such as a printer or a data file—a **redirector** intercepts the request and then examines it to determine whether the resource is local (on the computer) or remote (on the network). If the resource is local, the redirector sends the request to the local software component for processing. If the resource is remote, the redirector sends the request over the network to the server hosting the resource.

With redirectors, network resources can be accessed as though they were local. For example, a user or user application doesn't distinguish between a printer connected to a local USB port and one connected to the network. In addition, with drive mapping, shared network folders are accessed just like a drive that's physically attached to the system—at least from the user's point of view. In Windows, the redirector component is part of Client for Microsoft Networks (listed in the network connection properties). This client software is designed to access shared folders and files on servers by using the Server Message Block (SMB) protocol. In Windows, the two most common ways to access a shared resource are using the UNC path or mapping a drive.

In Chapter 1, you used the UNC path to access a shared folder, which has the syntax *server-name**sharename*. The *server-name* is the name of the computer where the shared resource resides. You can also use the server's IP address in place of its name. The *sharename* is the name given to the folder or printer when it was shared. (Sharing folders and printers is discussed later in "The Role of a

Server Operating System.") Typing a UNC path in the Search text box on the taskbar opens a File Explorer window showing the shared folder's contents. You can access a subfolder or file in the share directly by continuing the UNC path, as in *\\server-name\sharename\subfolder\filename*.

Tip

Linux systems also use the UNC path to access shared resources, but on Linux systems, forward slashes are used in place of backslashes.

You can use the UNC path to access shared folders and printers, but you must type the path every time you need it or create a shortcut with the UNC path as the target. One common method of making access to shared files easier (particularly those that are used often) is drive mapping, which associates a drive letter with the UNC path to a shared folder. Drives are usually mapped by using File Explorer or the `net` command. To use File Explorer, simply type the server portion of the UNC path in the Search text box on the taskbar to see a list of shared folders and printers the server is hosting. Right-click a shared folder and click Map network drive, as shown in Figure 11-9. You can then pick a drive letter (one that's not already in use) and choose to have Windows reconnect to the share with the same drive letter every time you log on.

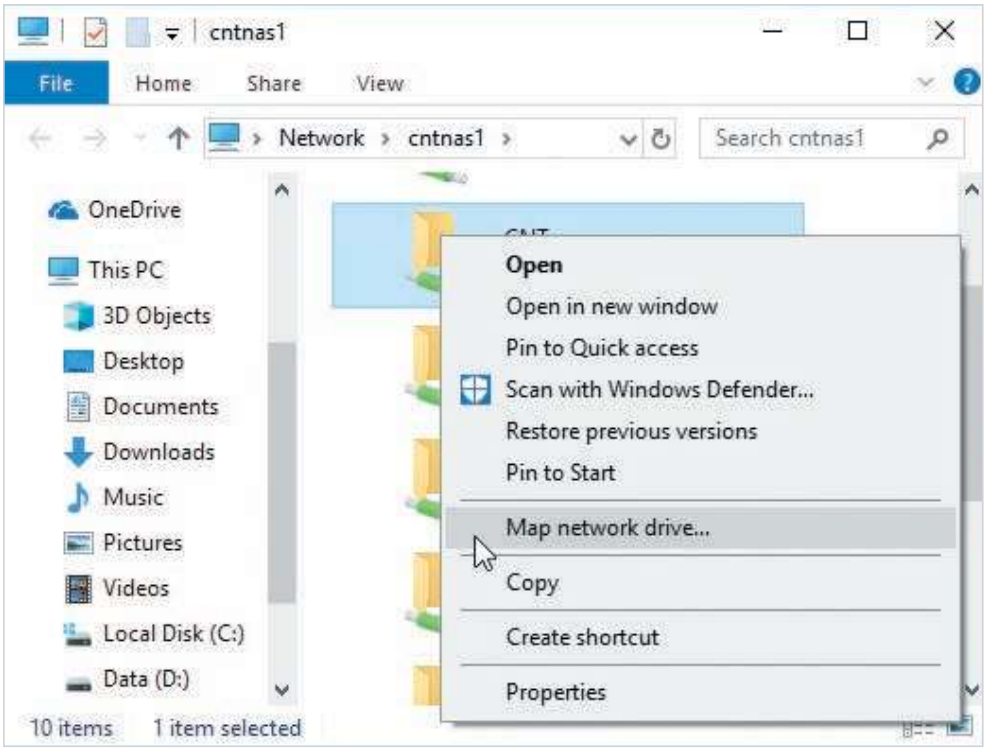

Figure 11-9 Mapping a drive in File Explorer

Another method of mapping a drive is using the `net` command. This method is often used by administrators in a **logon script**, which consists of commands that run when a user logs on to a Windows domain. The command to map a drive with the `net` command is `net use drive-letter: \\server-name\sharename`.

The *`drive-letter`* is an unused drive letter and must be followed by a colon (`:`). The command can be entered at the command prompt or placed in a batch file. A **batch file** is a text file containing a list of commands you ordinarily type at the command prompt. To run a batch file, enter its name at the command prompt or double-click the file in File Explorer. Batch files are useful for storing long, complex commands that are used often or a series of commands that are always used together.

For the sake of comparison, Linux doesn't use drive letters at all. Instead, Linux file systems are based on the concept of a file system root designator, which is simply the / character. All local and network drives and folders are accessed from the root as folders (or directories, as most Linux users call them). A drive or network share is mounted into an empty directory so that it becomes part of the file system hierarchy. So, to access a shared folder in Linux, you create a new directory at the root of the file system or in a subdirectory, and then mount the shared folder in the new directory.

The protocol used in Windows to share files and printers is SMB, also known as Common Internet File System (CIFS). Aside from file and printer sharing, SMB also provides a mechanism for interprocess communication between computers. Interprocess communication allows processes running on computers to communicate with one another for configuring and administering a computer over the network, for example.

Linux also supports SMB implemented as an installation option called Samba, but the native file-sharing protocol in the Linux environment is Network File System (NFS). NFS works much like SMB, in that NFS clients mount the shared folder into their local file system so that it appears as a local resource to both users and applications accessing it.

Using shared printers in Windows is even easier than mapping a drive. Simply right-click the shared printer in the File Explorer window and click Connect. A new printer is created in the Printers folder.

Hands-On Project 11-5: Mapping a Drive Letter

Time Required: 10 minutes
Objective: Map a drive letter by using different methods.
Required Tools and Equipment: Net-*XX*
Description: In this project, you create a shared folder, and then map a drive letter to it. You wouldn't normally map a drive letter to a share on your own computer, but this project shows you how to perform the process without using a second computer.

1. Log on to your computer as **NetAdmin**.
2. Click **Start** and then click **File Explorer**. Click **This PC** in the left pane, and then click **Local Disk (C:)**.

3. Create a folder named **MyShare**. Right-click **MyShare**, point to **Give access to**, and click **Specific people**. You see that NetAdmin is listed with the permission level Owner. You can add users who can access the share in this dialog box, but because only NetAdmin will access it, click **Share**. You're notified that the folder is shared, and you see the path listed as \\Net-*XX*\MyShare. Click **Done**.
4. Click in the Search text box on the taskbar, type **\\localhost**, and press **Enter**. The \\localhost refers to your own computer, so a window opens showing available shares, including MyShare. Normally, you wouldn't map a drive to a folder on your own computer, and you would replace "localhost" with the name of a server hosting the share. You're using localhost just for practice. Right-click **MyShare** and click **Map network drive**.
5. You can choose the drive letter to map to this share. Click the **Drive** list arrow and click **X:**. Click to clear the **Reconnect at sign-in** check box. If you leave this option selected, the drive is mapped to the share each time you log on. Notice that you can also choose to connect to the share with different credentials (username and password). Click **Finish**.
6. A File Explorer window opens, showing the share's contents. Close all windows. Right-click **Start**, and click **File Explorer**. You see the drive letter and share name listed under This PC. Right-click **MyShare (\\localhost) (X:)** and click **Disconnect** to delete the drive mapping. (You might need to press F5 to refresh the File Explorer window to see that the drive mapping has been deleted.)
7. Open a command prompt window. To map a drive letter from the command line, type **`net use x: \\localhost\MyShare`** and press **Enter**. You should see the message "The command completed successfully." To display current connections to shared resources, type **`net use`** and press **Enter**.
8. Click in the File Explorer window. The X drive letter is listed under This PC again.
9. At the command prompt, type **`net use /?`** and press **Enter** to see a list of options for the `net use` command. You can use the `/persistent` option to make a drive mapping reconnect each time you log on. You can also connect with a different set of credentials. Type **`net use x: /delete`** and press **Enter** to delete the drive mapping, and close the command prompt window.
10. To create a batch file for mapping a drive, open Notepad and type the following two lines:

```
net use x: /delete
net use x: \\localhost\Myshare
```

11. The first command deletes any existing drive mappings for the X drive. Click **File** and then click **Save As** from the menu. In the left pane of the Save As dialog box, click **Desktop**. Click the **Save as type** list arrow, and click **All Files**. In the File name text box, type **mapX.bat** and click **Save**. Close Notepad.
12. On your desktop, double-click **mapX**. In File Explorer, verify that the X drive mapping has been created. Right-click **Myshare (\\localhost) (X:)** and click **Disconnect**.

13. Write the command to map drive letter G to a share named Accounting on a server named Finance:

_____

14. Batch files can come in handy if you need to connect to another computer periodically but don't want a permanent drive mapping. They're especially useful if you often need to enter a long command because they save you the time of having to remember and enter the command each time you need it. Close all open windows, and leave Windows running for the next project.

Tip

To learn more about creating and using batch files in Windows, read the TechNet article at *https://technet.microsoft.com/en-us/library/bb490869.aspx*.

Hands-On Project 11-6: Creating and Connecting to a Shared Printer

Time Required: 10 minutes
Objective: Create and connect to a shared printer.
Required Tools and Equipment: Net-*XX*, and a computer with a shared printer to which all student computers can connect (or students can connect to each other's shared printers)
Description: In this project, you create a shared printer and then connect to it. You don't connect to an actual printer device, but this project walks you through the process of creating a printer, sharing it, and then connecting to a shared printer.

1. If necessary, log on to your computer as **NetAdmin**.
2. Click in the Search text box on the taskbar, type **Printers**, and click **Printers & scanners** in the search results. Click **Add a printer or scanner** in the Printers & scanners window. Windows searches for devices. Click **The printer that I want isn't listed**.
3. Click **Add a local printer or network printer with manual settings**, and then click **Next**. In the "Choose a printer port" window, leave the default option **Use an existing port** selected, and then click **Next**. In the "Install the printer driver" window, normally you select the printer's manufacturer and model, but because there's no physical printer, just accept the default selection, and then click **Next**.
4. In the "Type a printer name" window, click **Next**. In the Printer Sharing window, make sure **Share this printer so that others on your network can find and use it** is selected. For the share name, type **MyPrinter**, and then click **Next**. If you were actually installing a printer, you would click "Print a test page," but for this project, just click **Finish**.

5. To connect to a shared printer, ask your instructor whether a printer share is set up, or you can use another student's shared printer. In Windows, you can't connect to your own printer, as you did with the shared folder. Click in the Search text box on the taskbar, type ***computer***, and press **Enter** (replacing *computer* with the name of the computer sharing the printer, such as net-01 or net-instr).
6. When the window opens, right-click the shared printer and click **Connect**. In the Printers & scanners window, verify that the printer was created. It's listed as "*Printer make and model* on computer."
7. Close all open windows.

E-Mail Client

E-mail is the lifeblood of communication for most businesses and the people who work in them. Its complexity and importance combine to make it one of an IT department's biggest headaches. Most users, however, simply see e-mail clients as one of several communication tools they use every day and don't think much about how it works.

There's more to e-mail than just typing a message, attaching a file, and sending it to a colleague. E-mail is based on its own set of protocols, just as Web browsing and file sharing are. The most common e-mail protocols are as follows:

- *Post Office Protocol version 3 (POP3)*—E-mail clients use this protocol to download incoming messages from an e-mail server to their local desktops. POP3 clients must manage messages locally (not on the server, as they can with IMAP).
- *Simple Mail Transport Protocol (SMTP)*—This protocol is the standard protocol for sending Internet and other TCP/IP-based e-mail. POP3 is used to retrieve e-mail, and SMTP is used to send e-mail.
- *Internet Message Access Protocol (IMAP)*—This standard has advanced message controls, including the capability to manage messages locally yet store them on a server. IMAP also includes fault-tolerance features.

Sending an e-mail involves a series of steps. After a message has been written and the user clicks the Send button, the e-mail client software contacts an SMTP server. The SMTP server's address is part of the e-mail client's configuration. The SMTP server receives the message, looks up the domain of the destination address, and contacts an SMTP server at the destination domain. The destination SMTP server sends the message to the POP3 server containing the recipient's mailbox. The POP3 server deposits the message in the recipient's mailbox, where it sits until the mailbox owner instructs the e-mail client software to retrieve messages.

When you start your e-mail client or click the Get Mail (or equivalent) button in your client, the client uses POP3 to contact the POP3 server containing your mailbox.

The POP3 server forwards waiting messages to the client software and usually deletes them from the server. If you're using IMAP instead of POP3, only the message headers are sent; they include sender information and the subject. Only when you click the message header is the body of the e-mail sent. Messages aren't deleted from the server until you delete them with the client software. With IMAP, you have the advantage of being able to open and read e-mail on one computer and download the same messages on another computer. Because IMAP doesn't delete messages on the server automatically, they can be downloaded and opened from multiple locations. In addition, users' mailboxes can be backed up on the server so that users don't have to back up e-mail on client computers.

Note

Some ISPs support IMAP and some don't because of the extra space undeleted messages use on their servers. Also, POP3 has an option to leave downloaded messages on the server until they're deleted from the client computer.

Although the use of e-mail client and server software is still the norm in medium-sized and large businesses, many small businesses and home users access their e-mail by using a Web browser interface from sites such as Gmail.com and Outlook.com. In this case, the same processes occur when transmitting and receiving e-mail, except the Web server you connect to for accessing your mail performs these tasks instead of a locally installed e-mail client.

The Role of a Server Operating System

In the past, server OSs installed on servers in PC networks were dedicated to providing network services to client computers and couldn't run user applications and client network software. However, the OS installed on a desktop computer is now largely the same as that installed on a server, with the differences being the number and type of network services available and how server resources are used. For example, Windows Server is configured with Client for Microsoft Networks and DHCP and DNS client services. However, you can install DHCP and DNS server components on Windows Server as well as the Active Directory directory service; these services are unavailable in Windows client OSs. In Linux distributions, some installation programs let you choose a desktop or server configuration, but some distributions, such as Red Hat Enterprise, are designed as server OSs.

Memory, CPU, and disk use in client OSs are optimized to run user applications and client network software. In server OSs, use of these resources is typically optimized to run network services in the background to speed up responses to client requests. In addition, server OSs have more security and fault-tolerance features. The following is a

list (but by no means an exhaustive list) of the features and functions most server OSs provide in a typical network:

- Centralized user account and computer management
- Centralized storage
- Infrastructure services, such as name resolution and address assignment
- Server and network fault tolerance such as RAID and clustering
- Additional server features

Name resolution (DNS) and address assignment (DHCP) have already been covered in Chapter 5. The other server OS features are discussed in the following sections.

Centralized User Account and Computer Management

Among the most compelling reasons to design a network, even a small one, as a server-based network is centralized management of network resources, which includes the following functions:

- User authentication and authorization
- Account management
- Security policy management

User Authentication and Authorization

Authentication is the process of identifying who has access to the network. The most common form of authentication is a logon with a username and password. Other forms include digital certificates, smart cards, and biometric scanners.

Authorization is the process of granting or denying an authenticated user's access to network resources. Both authentication and authorization require users (and sometimes devices) to have an account that stores properties about the user, such as a logon name and password. A user account is also used to grant permissions for the user to access network resources.

Account Management

Most OSs, including those designed as client OSs, now incorporate account management for the purposes of authentication and authorization, but account management is centralized in the server OS. To better understand centralized account management as well as centralized authentication and authorization, consider a network in which account management is decentralized, as in a Windows workgroup network. As discussed in Chapter 1, each computer in this type of network maintains its own list of user accounts and controls access to its own resources. In a network of 10 computers, if each computer shares resources that are accessed by users on other computers, each user account must be created 10 times, once on each computer. The password for each user must also be maintained on each computer. If a user's password is changed on one computer, he or she has to remember a different password to access that computer or have the password

changed on all 10 computers. You can see how keeping up with this system could become tiresome quickly.

The server version of Windows OSs includes a centralized account management, authentication, and authorization system called Active Directory. Active Directory is a directory service that allows users to log on to the network once with their username and password, and access resources they're authorized for regardless of which computer stores the resource. When Active Directory is installed on a server, the server becomes a domain controller, and users and computers with accounts in Active Directory are referred to as domain members. Figure 11-10 shows the Active Directory Users and Computers management console. In the left pane are folders used to organize accounts and resources for easier management. In the right pane are user and group accounts, distinguished by different icons.

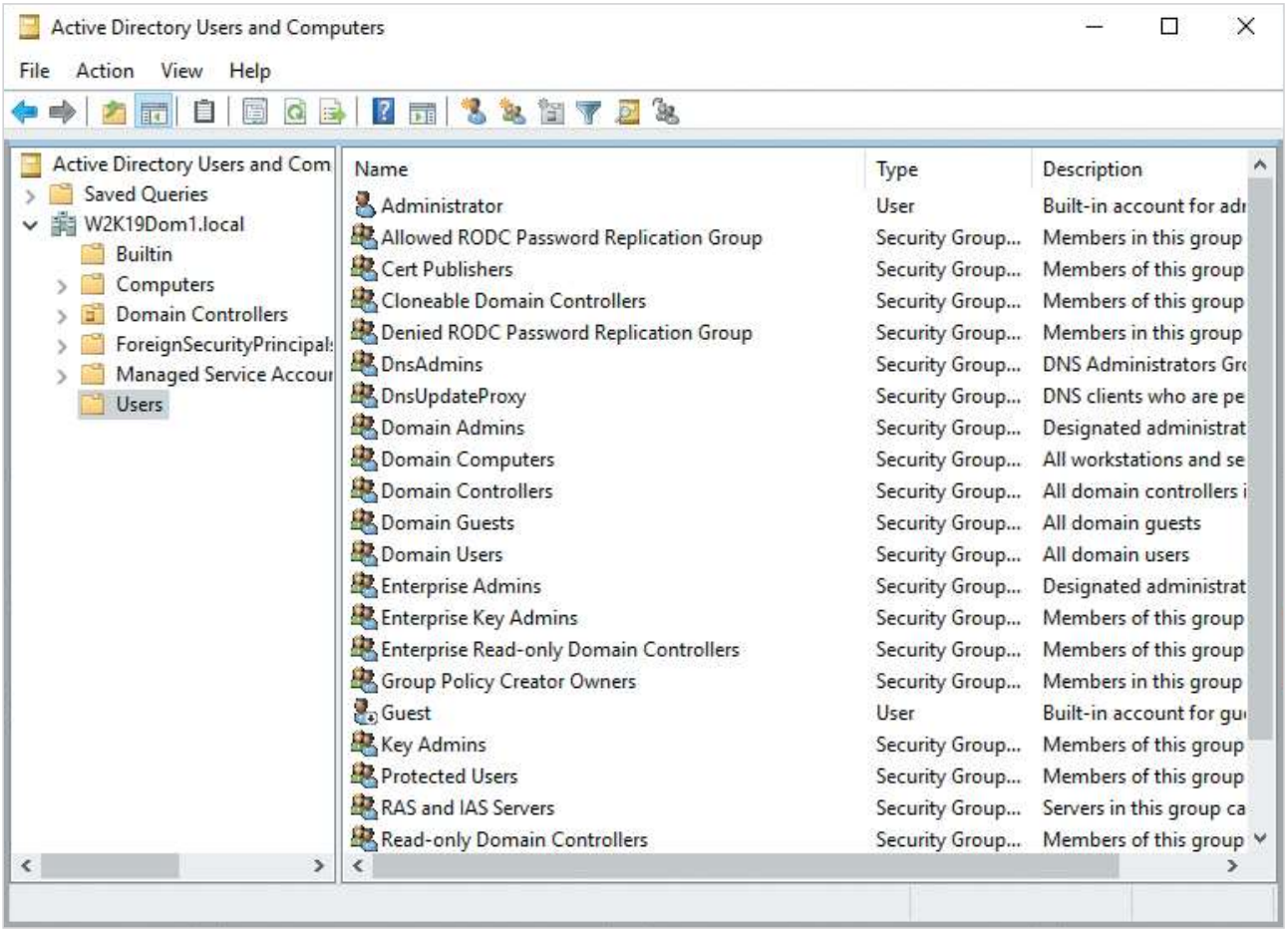

Figure 11-10 The Active Directory Users and Computers management console

When a computer running a desktop or server version of Windows becomes a domain member, an account is created for the computer in Active Directory. A computer becomes a domain member when you change its membership type from Workgroup to Domain in the Computer Name/Domain Changes dialog box accessed via the System Properties dialog box (see Figure 11-11). You work with accounts in Chapter 12.

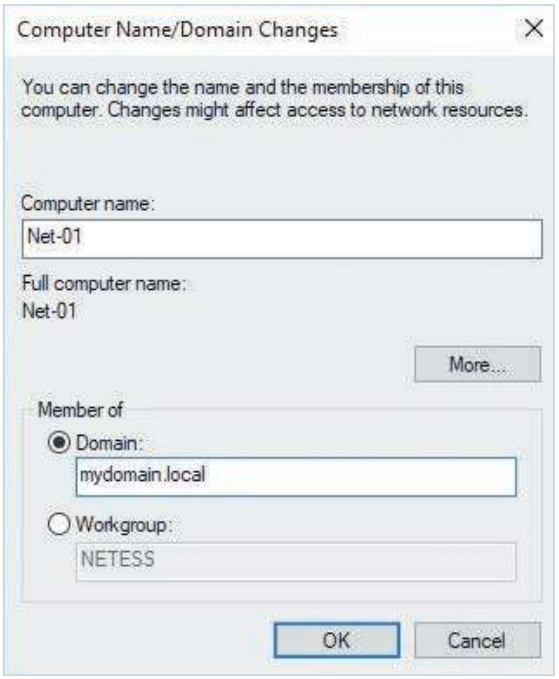

Figure 11-11 Making a computer a domain member

Security Policy Management

Aside from authentication and authorization, accounts in Active Directory are used to distribute and enforce policies for network use and security. These policies, called "group policies" in a Windows domain environment, can be applied to all domain members. Policies can range from user interface policies controlling what icons appear on the desktop and Start menu to security policies controlling password restrictions and what applications a user can run on a computer. Those are just a few examples of the power of group policies; hundreds of different policy settings are available.

Linux OSs have a basic directory service for centralized logon called Network Information Service (NIS), but Lightweight Directory Access Protocol (LDAP), which Active Directory is based on, is also commonly used in the Linux community. LDAP has the advantage of supporting both Windows and Linux user authentication and authorization.

Centralized Storage

With huge multimedia files being such a large portion of the data stored and processed on networks, network administrators are in a constant quest to better manage and maintain storage resources. Network storage includes file sharing, in which users store documents on network servers that other users can access. It also includes storing e-mail, user files, application databases, and data backups, among many other resources.

Traditional servers use locally attached disk drives to store the installed OS and applications as well as user files. However, the amount of data stored in even small to medium-sized networks is measured in dozens of terabytes, which is quite a burden for servers juggling numerous other network tasks. Although locally attached storage is still in common use, many network administrators are turning to specialized devices to help manage their storage requirements, including the following:

- Network-attached storage devices
- Storage area networks
- Cloud-based storage

Network-Attached Storage

A **network-attached storage (NAS)** device is a dedicated server device designed solely for providing shared storage for network users. An NAS could be a regular server with NAS software installed or it could be a **network appliance**, a device equipped with specialized software that performs a limited task, such as file sharing. Network appliances are often packaged without video interfaces, so you don't configure them with an attached keyboard and monitor. They have a built-in Web server to which you connect with a Web browser to configure and manage the device. Many NASs integrate with Active Directory or an LDAP-based system for user authentication and authorization.

Storage Area Network

A **storage area network (SAN)** is a high-speed, high-cost network storage solution that largely replaces locally attached drives on servers. SAN technology allows multiple servers to access an enormous amount of shared storage that appears as locally attached drives from the server and user's perspective. Servers can even boot their OSs from a SAN instead of booting from local disks. This type of centralized storage offers better reliability and fault tolerance than traditional storage methods. Additionally, because storage is shared among several servers, power requirements for maintaining these systems are lower than those needed to maintain several servers with their own locally attached storage. The most common network technologies in SANs are Fibre Channel and iSCSI. They're designed to connect large arrays of hard drive storage that can be accessed and shared by servers. Client computers access the shared data by contacting the servers via the usual method, and the servers retrieve the requested data from the SAN devices and pass it along to the client computer. Figure 11-12 shows a LAN with three servers connected to a SAN.

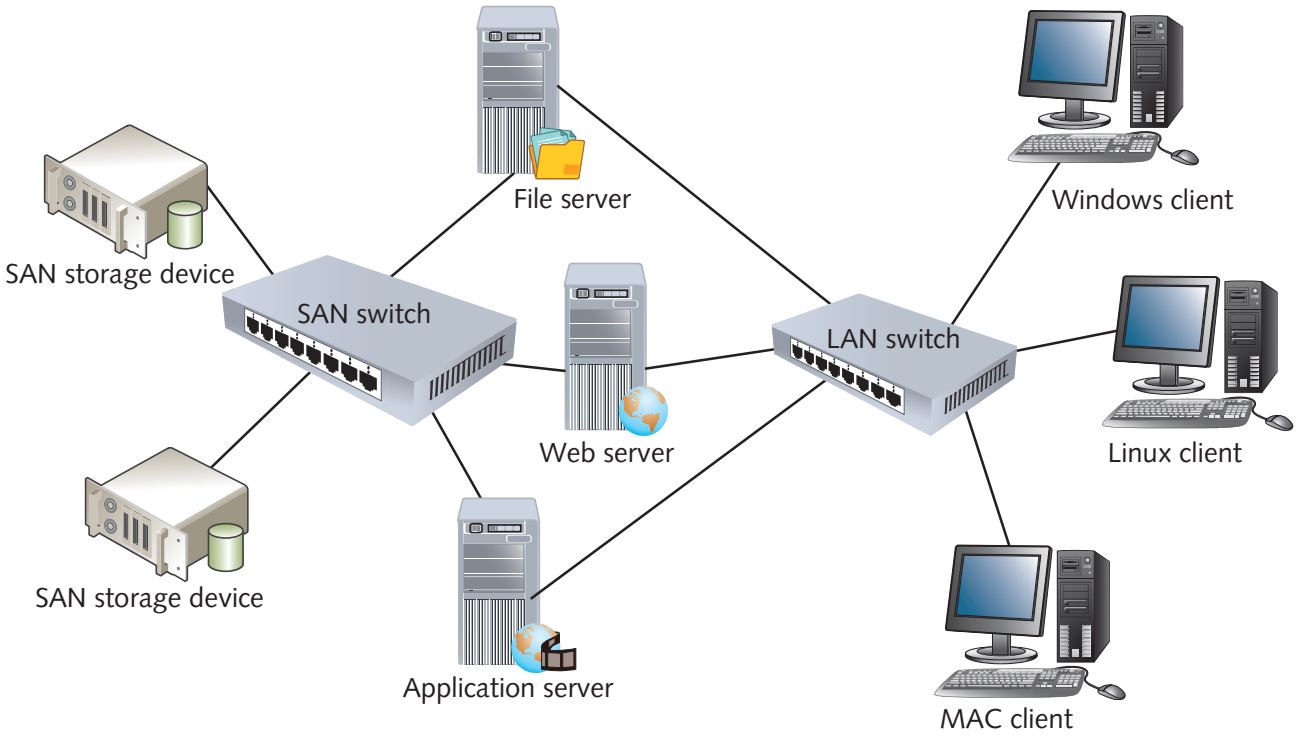

Figure 11-12 A storage area network

Cloud-Based Storage

When a company's storage needs have outgrown its storage capabilities, whether because of physical capacity limits or the lack of personnel to maintain in-house storage, the company can turn to the cloud, as you learned in Chapter 10. In this case, the computing solution is network storage. With **cloud storage**, some or all of an organization's data is stored on servers located offsite and maintained by a storage hosting company. The customer can manage storage by assigning permissions for user access and allocating storage for network applications and so forth without having to physically maintain the servers. If more storage is needed, the customer simply pays the storage hosting company for the additional space. The advantage of this approach is that the details of managing and backing up storage on local servers are offloaded to a third party, which enables a company to focus its monetary and personnel resources on business rather than IT tasks. Cloud-based storage isn't for everyone, though. The data a company maintains might be too sensitive to trust to a third party, or the data access speed might not be sufficient, just to give two examples. Cloud storage is a rather new model in network storage, but it's one that's here to stay.

Server and Network Fault Tolerance

By now, you know that a server computer is defined by the type of software installed on it. For example, Windows Server 2016 and most Linux distributions can be installed on an inexpensive laptop just as easily as they can on a $20,000 server, but laptops are not adequate for an enterprise-level network. A network's servers are critical to business operations, so keeping them running at peak performance is essential for user productivity and business transactions. For this reason, certain fault-tolerance features are built into server OSs, and only servers designed to use these features can access them. Some fault-tolerance features on a server OS that aren't usually available on client OSs include the following:

- *Support for hot-swappable devices*—A **hot-swappable device** can be removed, replaced, or added to a server while it's running. Many low-end servers support hot-swappable disk drives, but only high-end servers are likely to support hot-swappable memory and CPUs. Windows Server 2012 and later versions support hot-swappable disks, memory, and CPUs. Red Hat and other Linux distributions support hot-swappable devices (called "hotplug" in the Linux world), too.
- *Server clustering*—A **server cluster** is two or more servers configured to operate as a single unit. The most common types of server clusters are failover clusters and load-balancing clusters. A **failover cluster** is used to provide fault tolerance so that if one server fails, the other immediately takes over its functions with no or little downtime. A **load-balancing cluster** provides high-performance computing and data access by spreading the workload among multiple computers. Physically, there are multiple servers, but logically, they work as one unit. Load-balancing clusters have the added advantage that if one server fails, the others still operate, which ensures fault tolerance.
- *Redundant/high-end disk systems*—Hard drives are a critical component of a computer and one of its few moving parts, making them more susceptible to failure than other components. Most high-end servers support enterprise-class Serial Attached SCSI (SAS) disks designed for an around-the-clock duty cycle. Low-end servers and desktop computers support only Serial ATA (SATA) disks, which lack some of the performance properties of SAS. However, even high-end disk drives can fail, so most servers incorporate disk controllers capable of a disk arrangement known as **redundant array of independent disks (RAID)**. With RAID, you can configure disks in a fault-tolerant arrangement so that if one disk fails, the data is preserved, and the server can continue to operate. Even some desktop computers support variations of RAID, but the variations with higher performance and fault tolerance are standard on servers. RAID is discussed in more detail in Chapter 12.

Additional Server Features

As mentioned, OS vendors reserve many high-end applications and network services for the server version of the OS. Some applications and services usually found only on servers include the following:

- *Remote access*—A mobile workforce needs convenient access to the company network from anywhere in the world. Most server OSs support virtual private networks (VPNs) and, if necessary, the older dial-up method of remote access.
- *Database server*—Many applications rely on a database to store and retrieve vast amounts of data. Server OSs support advanced database systems, such as MySQL, SQL Server, and Oracle.
- *Client/server applications*—Client/server applications, such as e-mail systems (Microsoft Exchange, for example), must run on a server OS. Web-based applications, too, need a server OS to handle the computing and network workload of these applications.
- *Virtualization*—Virtualization is an integral part of most IT data centers. Virtualization software can run on desktop systems, but for virtualizing production servers, you need a server-based product, such as Microsoft Hyper-V or VMware vSphere. For open-source fans, Citrix Hypervisor (*www.citrix.com/products/citrix-hypervisor/*), formerly XenServer, might be a good fit with your data center. Virtualization is such an important aspect of the computing environment that the next section focuses on this topic.

The list of applications and services usually reserved for servers and server OSs continues to grow as networks play an increasingly important role in personal and work activities. For now, turn your attention to OS virtualization, one of the hottest topics in computing.

Operating System Virtualization

OS virtualization has become a mainstream technology in both small and large networks. **Virtualization** is a process that creates a software environment to emulate a computer's hardware and BIOS, allowing multiple OSs to run on the same physical computer at the same time. This environment can be installed on most current OSs, from Windows to Linux to macOS. In this case, a picture is worth a thousand words, so examine Figure 11-13. It shows a Windows 10 client running a Windows Server 2019 virtual machine, using VMware Workstation virtualization software. Notice that there are two Start buttons: one on the host desktop and one on the virtual machine.

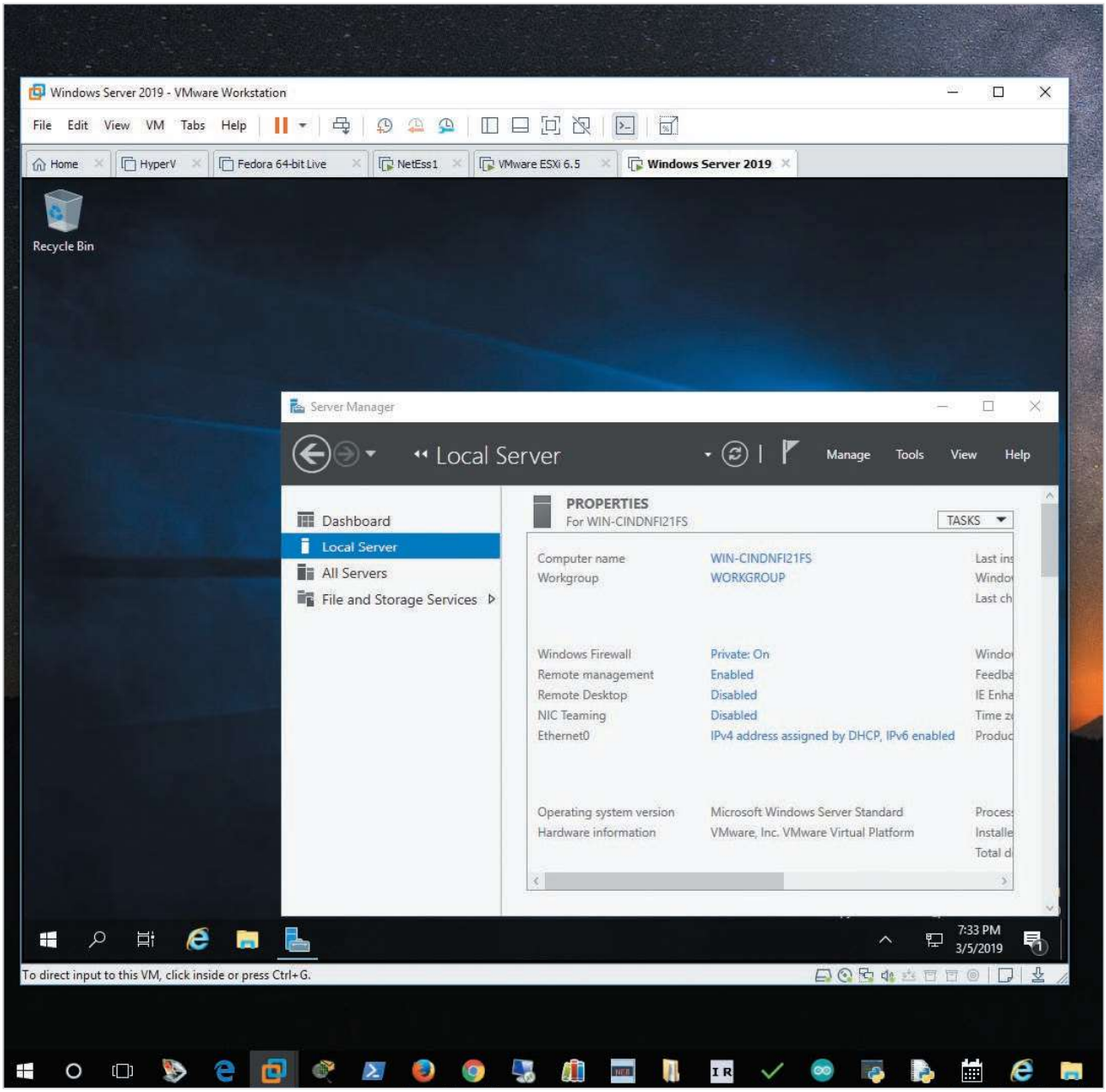

Figure 11-13 Windows Server 2019 running as a virtual machine in Windows 10

Source: VMware, Inc., *www.vmware.com*

Like all technologies, virtualization has a collection of terms that define its operation and components:

- A **virtual machine (VM)** is the virtual environment that emulates a physical computer's hardware and BIOS. A **guest OS** is the operating system installed on a VM.
- A **host computer** is the physical computer on which VM software is installed and VMs run.

- Virtualization software is the software for creating and managing VMs and creating the virtual environment in which a guest OS is installed. Examples are VMware Workstation, Oracle VirtualBox, VMware vSphere, and Microsoft Hyper-V.
- The **hypervisor** is the virtualization software component that creates and monitors the virtual hardware environment, which allows multiple VMs to share physical hardware resources. On a host computer, it acts somewhat like an OS kernel, but instead of scheduling processes for access to the CPU and other devices, it schedules VMs. It's sometimes called the "virtual machine monitor (VMM)." There are two types of hypervisors:
  - A type 1 hypervisor implements OS virtualization by running directly on the host computer's hardware and controls and monitors guest OSs. It also controls access to the host's hardware and provides device drivers for guest OSs. Also called **bare-metal virtualization**, it's used mainly for server virtualization in data centers. Examples include VMware vSphere (or ESXi), Citrix Hypervisor, and Microsoft Hyper-V Server.
  - A type 2 hypervisor implements OS virtualization by being installed in a general-purpose host OS, such as Windows 10 or Linux, and the host OS accesses host hardware on behalf of the guest OS. Also called **hosted virtualization**, it's used mostly for desktop virtualization solutions. Examples include VMware Workstation Player and Workstation, Oracle VirtualBox, and OpenVZ for Linux.
- A **virtual disk** consists of files residing on the host computer that represent a virtual machine's hard drive.
- A **virtual network** is a network configuration created by virtualization software and used by virtual machines for network communication.
- A **snapshot** is a partial copy of a VM made at a particular moment; it contains changes made since the VM was created or since the last snapshot was made, and is used to restore the VM to its state when the snapshot was taken.

Figure 11-14 illustrates the virtualization process, with a host computer connected to a physical network. The hypervisor on the host is running two VMs connected to a virtual network, which has a connection to the physical network so that the VMs can communicate on the physical network.

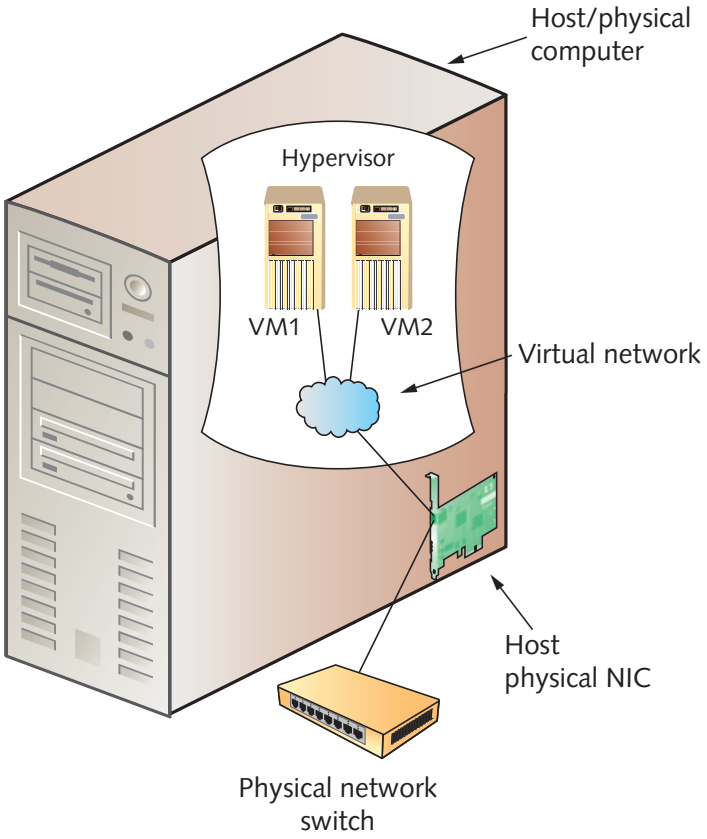

Figure 11-14 How virtualization works

One of the best ways to understand a technology is to understand the reasons it's used. The reasons to use virtualization are many and varied and are best discussed by splitting the topic into the two main types of virtualization: hosted and bare-metal.

Hosted Virtualization

As mentioned, hosted virtualization uses a type 2 hypervisor, which is installed in a standard desktop or server OS. It has the advantage of supporting a wider variety of guest OSs than bare-metal virtualization does, mostly because the guest OS uses the host OS to access host hardware, so there are few incompatibility problems between the guest OS and hardware. For example, you can run a distribution of Linux in a virtual machine on a host computer, even if you can't install Linux directly on the physical machine because of driver incompatibilities.

Another advantage of hosted virtualization is that it's easy and straightforward to use. With hosted virtualization, you install the virtualization software on your computer and begin creating virtual machines. There are few hardware requirements, and most products run on Windows versions starting with XP as well as Mac OS X and most Linux distributions. All that's required is enough memory to support the host and guest OSs, adequate CPU power, and enough free disk space to store the virtual disk. A system running Windows 10 with 4 GB of RAM, a 2.0 GHz CPU, and 40 GB of free hard drive

space can run Linux and a Windows Server 2016 virtual machine at the same time. Performance might not be stellar, but the virtual machines should work well enough for experimenting or training (one of the main reasons for using hosted virtualization).

Hosted Virtualization Applications

Hosted virtualization is so flexible and easy that its uses are varied and continuing to grow as people find different applications for it. Some common applications include the following:

- *OS training*—Whether in the classroom or at home, learning multiple OSs has often been a problem of not having enough computers or a lack of compatibility between the OS and available computers. With virtualization, a computer can have a host OS installed, such as Windows 10, and have virtual machines for numerous Linux distributions, Windows 7, Windows Server 2016, and even Novell NetWare. If you want to learn about the past, you can install Windows 3.11, DOS, or OS/2 (if you can find installation media for these very old OSs). In addition, you can run multiple VMs at the same time by using a virtual network, which enables you to work with both client and server OSs in situations that would normally take two or more physical computers.
- *Software training*—Students and employees can be trained on new software packages by giving them VMs with preinstalled software.
- *Application isolation*—Not all software works well together, so if an application conflicts with other installed software, it can be installed in its own VM, effectively isolating it from the host machine's installed software.
- *Network isolation*—Installing some networking services, such as DHCP, can wreak havoc with an existing network. Virtual networks can be isolated from the rest of the network, however, so you can experiment with these services without causing the IT department to pay you a visit.
- *Software development*—Software developers often need to design software that works on multiple OSs and OS versions. Testing on VMs makes this process easier, compared with using a physical computer for each OS to be tested.
- *What-if scenarios*—If you want to try out a software package or see whether a configuration option you read about will actually improve performance on your computer, you might not want to risk destabilizing your physical computer. You can install software and make configuration changes safely on a VM before making the commitment on your host computer.
- *Use of legacy applications*—If you have a favorite application that won't run on a newer OS, you don't have to forgo the latest hardware technology because of one application. You can install the old OS in a VM and run your legacy application on it.
- *Physical-to-virtual conversion*—Your six-year-old machine is getting slow and unreliable, so you bought a new desktop computer. However, you have several applications on your old computer and no longer have the installation media. You can convert your old computer to a virtual machine, and then maintain all the software and run it on your new desktop computer as a VM. You'll probably even see a speed boost.

As you can see, virtualization can bring plenty of benefits to your computing experience. You have many choices of products in this category, and the good news is that many are free. The following section describes some products for hosted virtualization.

Hosted Virtualization Products

Several hosted virtualization products are available. The following are the best known:

- *VMware Workstation Pro*—VMware, the virtualization pioneer in the PC world, released VMware Workstation in 1999. It isn't free, but it offers the most features, including multiple snapshots, nested virtualization (the capability to run a virtual machine inside another virtual machine), and extensive guest OS support.
- *VMware Workstation Player*—This free download from VMware has a streamlined user interface and fewer advanced features than Workstation, but it maintains excellent guest OS support.
- *VMware Fusion*—Another product from VMware, this software runs on macOS and supports the same guest OSs as VMware Workstation.
- *Parallels Desktop for Mac*—This product works on Macintosh operating systems and supports a number of guest OSs, including Windows, Linux, macOS, OS/2, and Solaris.
- *VirtualBox*—Originally developed by Innotek, VirtualBox is now developed by Oracle Corporation. Two versions are available: a proprietary version that's free for home users and can be purchased for enterprise use, and a free open-source version with a reduced feature set. VirtualBox runs on Linux, Mac OS X, or Windows hosts, and the proprietary version has features similar to VMware Workstation.

Tip

For more information on virtualization products, the platforms they run on, and supported guest OSs, review the article at *http://en.wikipedia.org/wiki/Comparison_of_platform_virtual_machines*.

These products have their strengths and weaknesses; the best approach is to work with different products to see which best serves your needs. The following sections discuss using some of these products.

Using VMware Workstation Pro

VMware Workstation Pro isn't free, but you can download a trial version at no cost and try it for 60 days. Not-for-profit educational institutions can join the VMware Academic program to give students and faculty free downloads of VMware Workstation and other VMware products.

After VMware Workstation Pro is installed, a wizard takes you through the steps of creating a virtual machine. You can choose the size of the virtual disk and set other hardware options or just accept the defaults.

Note

One convenience of installing a guest OS in a VM is being able to boot to the installation program with an ISO file rather than a DVD disk. This way, if you download the ISO file, burning a DVD to do the OS installation is unnecessary. In addition, the ISO file can be stored on a server and used by multiple users for VM installations.

A nice feature of VMware Workstation Pro is flexible networking options. You can configure the network interface card (NIC) on your VM to use one of the three virtual network options or you can create your own custom virtual network. VMware Workstation supports VMs with multiple NICs, and each NIC can be connected to a different virtual network. The three preconfigured options are as follows (see Figure 11-15):

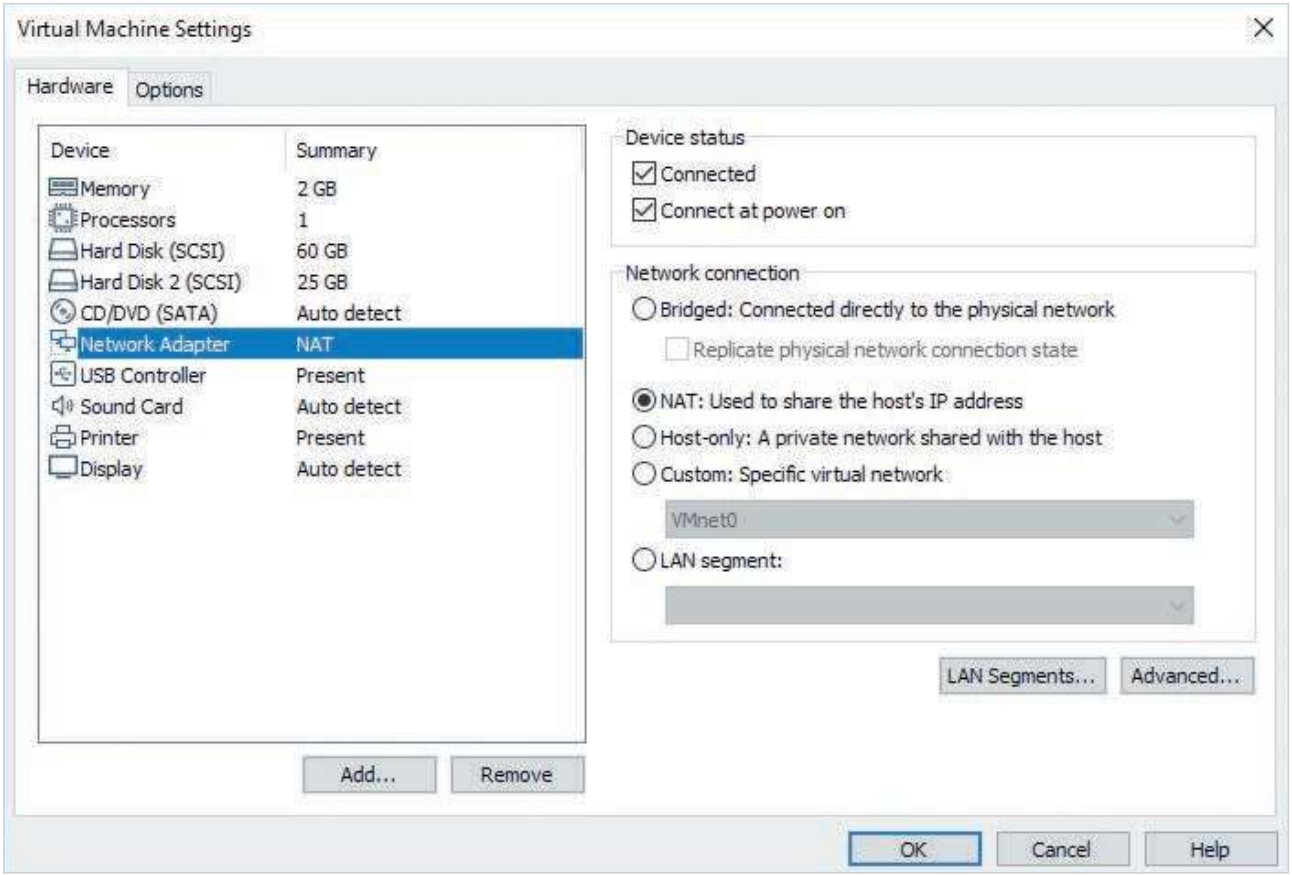

Figure 11-15 VMware virtual network options

Source: VMware, Inc., *www.vmware.com*

- *Bridged*—This option connects the VM's virtual network to the physical network, and the VM acts like any other computer on the physical network, including having an IP address on the physical network. This option is illustrated in Figure 11-14, shown previously, in which the virtual network has a (virtual) connection to the host's physical NIC.
- *NAT*—With this default option, the host computer's IP address is shared with the VM by using Network Address Translation (NAT). The main difference between the NAT and Bridged options is that VMs are assigned an IP address from the host computer rather than the physical network, and the host translates the address for incoming and outgoing packets. This option is more secure than the Bridged option because the VM isn't directly accessible. However, it's not a viable option for a VM that provides server functions to the host network.
- *Host-only*—This option isolates the VM from the host network and allows network communication only between VMs running on the host and the host computer. It's the most secure configuration and has the lowest risk of the VM causing problems with the host network. This configuration works well when you have multiple VMs that must communicate with one another but don't need to access computers or devices outside the host.

Note

Other virtualization software vendors use different terms to describe virtual networks, but the concepts are the same.

After the virtual machine is installed, you use it as you would any computer, except there are no physical on/off buttons.

VMware Tools, which is a collection of tools and drivers, should be installed in the guest OS for the best performance and ease of use. It adds optimized network, video, and disk drivers and guest-host integration tools that allow dragging and dropping files as well as cutting and pasting between the guest OS and host OS.

Other advanced features targeted to developers are available, which is why VMware Workstation Pro is generally considered the flagship hosted virtualization product. However, if you don't need all the bells and whistles and simpler is better, try VMware Workstation Player.

Using VMware Workstation Player

VMware Workstation Player is a stripped-down version of VMware Workstation, but it still offers the basics of desktop virtualization in a streamlined and easy-to-use interface. You can download it for free from the VMware Web site, and it's also included with the VMware Workstation Pro package. The opening window of VMware Workstation Player gives you an idea of its clean interface (see Figure 11-16).

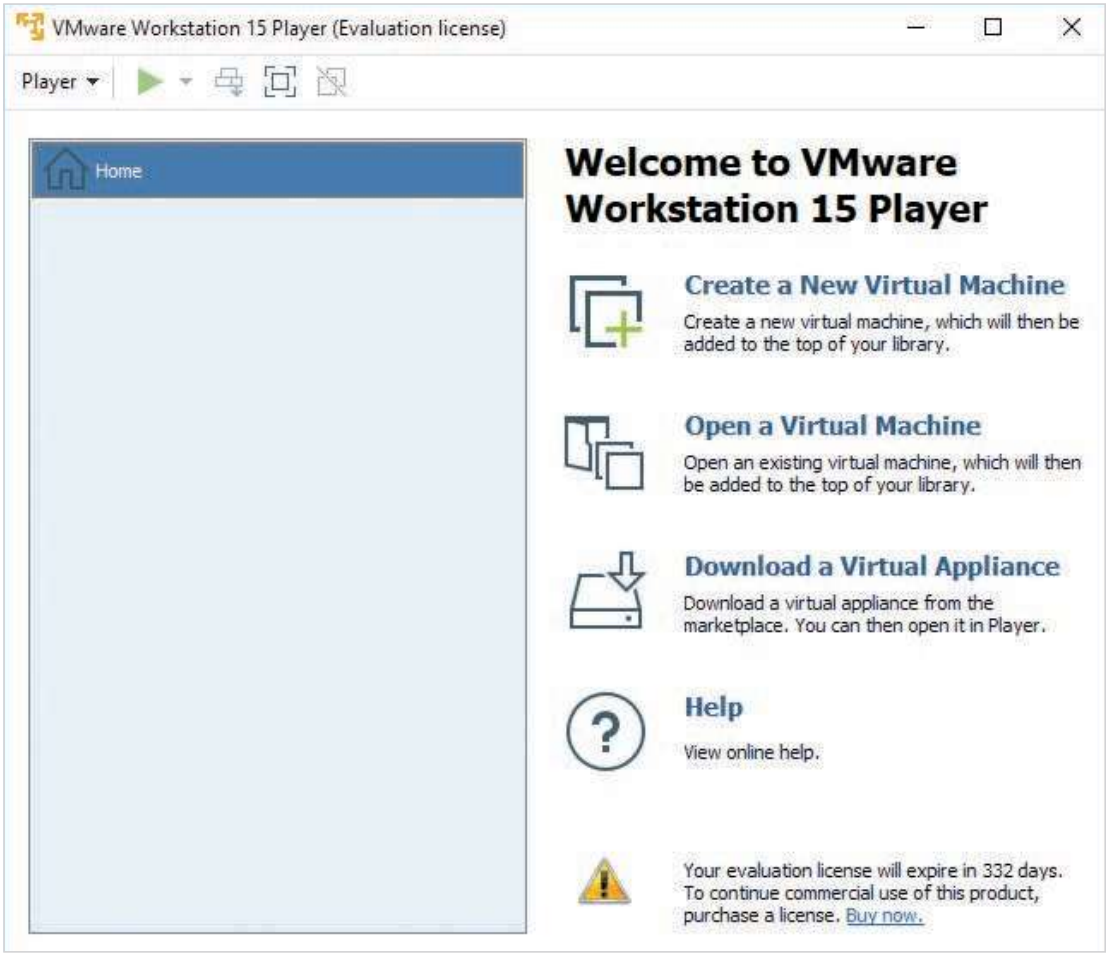

Figure 11-16 The VMware Workstation Player Welcome window

Source: VMware, Inc., *www.vmware.com*

Creating a VM in VMware Workstation Player is a wizard-based affair, nearly identical to the one in VMware Workstation Pro. Notice in Figure 11-16 the option to download a virtual appliance. Virtual appliances are ready-to-use VMs from OS and software vendors that contain a guest OS with preconfigured applications or network services. In some cases, a virtual appliance is just a preinstalled guest OS. A virtual appliance is an easy way to use and evaluate a product or configuration without having to install it yourself. Virtual appliances can be run by VMware Workstation Player or Workstation Pro and sometimes by VMware's bare-metal virtualization products.

VMware Workstation Player offers many of the same features as VMware Workstation Pro, with the exception of snapshots, customized virtual networks (although the three preconfigured network options are available), and some advanced network and virtual hardware settings. It's a good choice for new virtualization users and for classroom and training centers, where the interface's simplicity is an advantage.

Using VirtualBox

VirtualBox can be installed on Windows, Mac OS X, Linux, and Solaris hosts and supports a wide range of Windows, Linux, and other guest OSs, making it the most versatile of the products discussed. Like the other products, virtual machines are created with a wizard that walks you through selecting the guest OS and the VM's hard disk and RAM configuration; however, you can change all these settings after the VM is created. The VirtualBox user interface consists of a console where you can create VMs and view the status of all VMs. VirtualBox supports unlimited snapshots, so you can save a VM's state as you work with it and restore its state from any of the snapshots you make. You can even jump forward and backward in snapshots, meaning that if you have three snapshots, you could revert to the first one and later go back to the third snapshot.

Virtualization Software Summary

All the virtualization products discussed so far provide a type 2 hypervisor (hosted virtualization). Table 11-2 summarizes some major features and differences in these products.

Table 11-2 Comparing features of hosted virtualization software

	VMware Workstation Pro	VMware Workstation Player	Oracle VirtualBox
Price	$249 or free with Academic Program membership	Free	Free
Host OS support	Windows, Linux, Mac OS X (with VMware Fusion)	Windows, Linux	Windows, Linux, Mac OS X, Solaris
Guest OS support	Windows, several Linux distributions, NetWare, Solaris, DOS	Same as VMware Workstation Pro	Windows, several Linux distributions, Solaris, Mac OS X Server, DOS, OS/2, others
Snapshots	Unlimited	None	Unlimited
Virtual network options	Bridged, NAT, host-only, custom	Bridged, NAT, host-only	Bridged, NAT, host-only, internal
Host integration tools	VMware Tools, Unity	VMware Tools, Unity	Guest additions, seamless mode
Other features	Virtual teams, screen capture and screen movie capture, physical-to-VM conversion, developer tools		Command-line management interface, built-in remote desktop, developer programming interface, open-source edition

A benefit of these virtualization products is that you can install all of them and run them at the same time on a single host computer, so you can download and install each one and evaluate it for yourself.

Bare-Metal Virtualization

Bare-metal virtualization products (type 1 hypervisors) are targeted mainly for production virtualization in data centers. These products are installed directly on hardware and have more stringent host machine requirements than hosted products. Because they're targeted for IT departments, they have more features for managing VMs and have a performance advantage over hosted virtualization products. Their installation and use tend to require more sophisticated, knowledgeable users, too. Before learning about specific products, take a look at some applications for bare-metal virtualization products in the next section.

Bare-Metal Virtualization Applications

Bare-metal virtualization products come with a price tag for the virtualization software, the hardware to run it on, or both. So, when considering whether to use virtualization in an IT data center, most IT managers look for a return on their investment in real money or in productivity gains. The following applications show that bare-metal virtualization can deliver both:

- *Consolidate servers*—Server consolidation is probably the original reason for using bare-metal virtualization and is done for the following reasons and benefits:
  - Retire old or unreliable hardware: Converting physical machines to VMs and running them on the latest hardware means you can get rid of old hardware, thereby gaining a reliability advantage and avoiding the tedious task of reinstalling and reconfiguring a server OS on new hardware. You will also likely improve performance.
  - Make optimal use of multicore, high-performance servers: Some server roles, such as Active Directory, should be the only major network service running on a server. With multicore server CPUs, you're likely to waste a lot of the server's power if you install a single-role OS. Instead, run two, three, or more VMs on the server, making optimal use of the available performance.
  - Maintain application separation: Some applications and services run best when they're the only major application installed on an OS. You avoid OS resource conflicts and gain stability and reliability.
  - Reclaim rack or floor space: By consolidating a dozen physical servers into three or four host servers, you're no longer tripping over a plethora of towers or wondering whether your rack can handle one more server. You can even clear enough room for an easy chair and a reading lamp so that you can catch up on the latest technical journals in comfort!
  - Reduce cooling and power requirements: In most cases, by reducing the number of servers (even with higher-performance machines), you save money on cooling and powering a data center, especially when you reduce hundreds of servers down to dozens of hosts.
- *Test installations and upgrades*—Before you install a major software package or upgrade on your server, create a copy of the VM (referred to as "cloning" in some

products), and go through a test run to iron out any potential problems or conflicts. If something still goes wrong on the production VM, you can revert to a snapshot.

- *Test a preconfigured application*—Not sure whether the application the vendor is trying to sell you is right for your company? Some vendors offer virtual appliances you can use to evaluate the application without the trouble of installing it.
- *Test what-if scenarios*—You can create a virtual network and run clones of your production VMs to test ideas for improving your network's performance, functionality, and reliability. This type of testing on live production systems is never a good idea, but it's ideal on virtual machines.
- *Live migration*—Virtual machines can be migrated to a new host while they're running for performance or reliability improvements with practically no downtime. Live migration features also ensure VM fault tolerance in clustered server environments.
- *Dynamic provisioning*—Advanced VM management systems can deploy VMs and storage dynamically to meet application requirements. This advanced feature has uses in clustered computing, cloud computing, and virtual desktop infrastructure.

Caution

VMs that run distributed server applications, such as Active Directory, in which multiple servers synchronize a common database with one another, shouldn't be backed up or moved by copying the virtual hard disk, as it might result in database inconsistencies. Use only backup and migration tools approved for the virtualization software.

Bare-Metal Virtualization Products

VMware dominated the type 1 hypervisor category for years, but now you have a choice of products. The following are the most common bare-metal virtualization products:

- *Microsoft Hyper-V*—Hyper-V was introduced with Windows Server 2008 and can be installed as a server role, in which case the hypervisor is installed as a layer of software between Windows Server and the server hardware. Windows Server acts as a parent or management OS for VMs installed with Hyper-V. Hyper-V is included with Windows Server at no additional cost, or you can download the stand-alone Hyper-V Server for free from the Microsoft Web site. (You can install Hyper-V Server directly on the server, with only a command-line interface available for rudimentary management tasks; it's managed remotely by another Windows Server computer.) Hyper-V supports advanced features, such as host server clustering and live migration, and requires a 64-bit CPU with virtualization extensions enabled on the host system. Virtualization extensions offload some virtualization work to the CPU and are available on most current CPUs.

A big advantage of using Hyper-V is that Microsoft provides virtual instances of the OS with no additional licensing fees. For example, Windows Server 2019 Standard Edition allows you to run two virtual instances (two VMs) of the OS at no additional cost. Datacenter Edition allows an unlimited number of virtual instances. Hyper-V has guest OS support for Windows Server OSs (Windows 2000 Server and later versions), Ubuntu, SUSE, and Red Hat Enterprise Linux distributions, Windows client OSs (Windows XP and later), and more.

Microsoft has made Hyper-V available with Windows client OSs, too. You can enable Hyper-V in Windows 8 and later versions by going to Programs and Features in Control Panel and clicking "Turn Windows features on or off." After Hyper-V is installed, you need to restart your computer and open Hyper-V Manager (see Figure 11-17) from Administrative Tools in Control Panel.

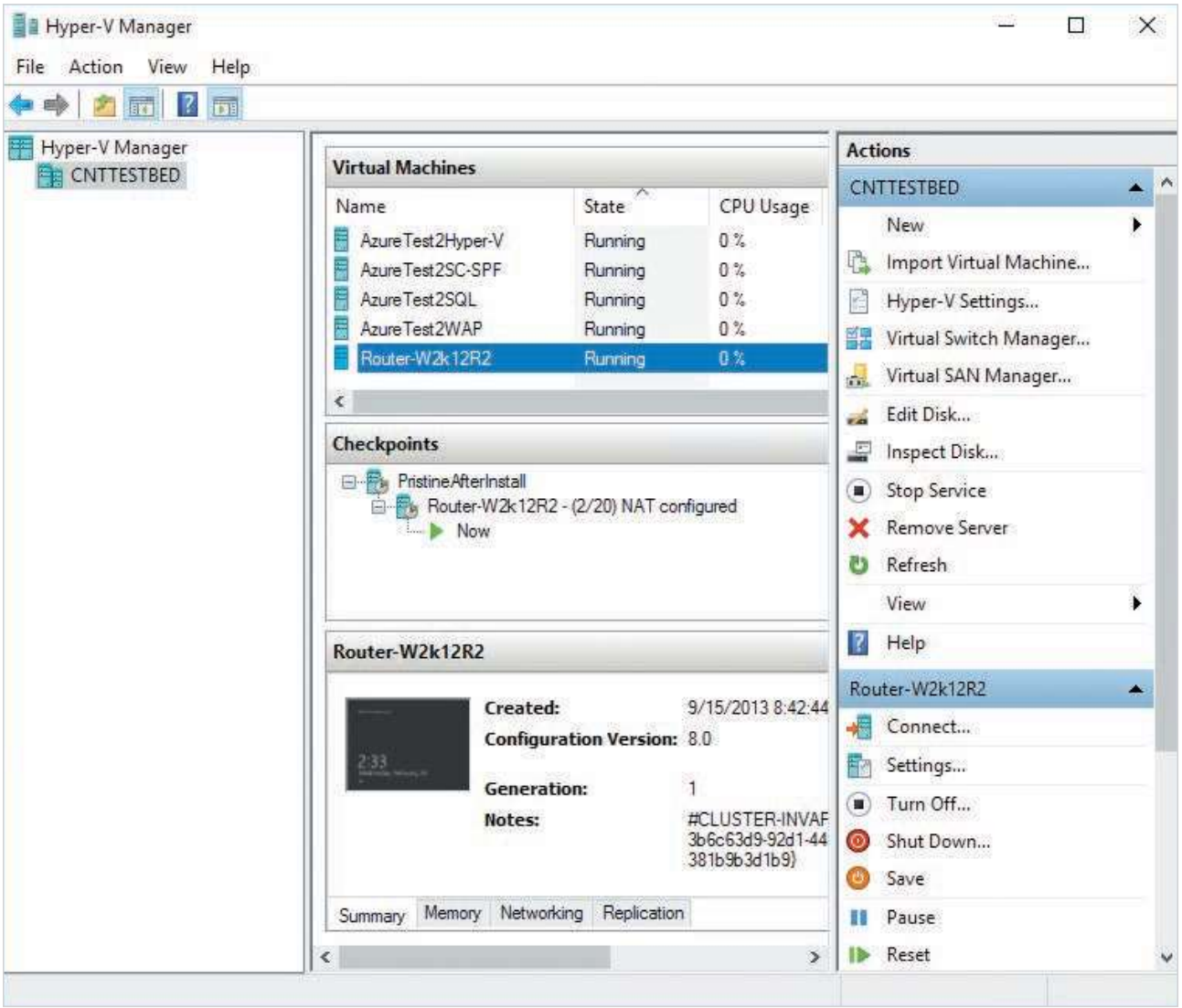

Figure 11-17 The Hyper-V Manager console

- *Citrix Hypervisor*—Formerly XenServer, this hypervisor uses Linux as a management OS on the host. Like Hyper-V, a XenServer host computer requires a 64-bit CPU with virtualization extensions to run Windows guest OSs. Guest OS support includes most Windows OSs starting with Windows Vista SP2, and SUSE, Red Hat, Oracle, Debian, Ubuntu, and CentOS Linux distributions. To manage your host and VMs, you download and install XenCenter on a Windows client or server computer.
- *VMware vSphere*—vSphere includes VMware ESX Server, which is installed directly on the physical server without a management OS. After ESX Server is installed, a basic command-line console based on Linux is available for simple configuration tasks, such as IP address configuration. Most configuration tasks are done from a Web browser-based client, or from vCenter Server, which is a VMware management platform. ESX Server has the broadest guest OS support, including Windows versions back to Windows 3.1, more than a dozen Linux distributions, Novell NetWare, Solaris, and others.

All these products have extensive management tools for managing up to hundreds of hosts and a wide array of storage resources. These tools are available for a fee from virtualization software vendors. For example, Microsoft has System Center Virtual Machine Manager (SCVMM) for managing Hyper-V and ESX Server hosts. Citrix Hypervisor offers versions with different levels of management, depending on which product you purchase, and VMware sells vCenter Server and vCloud Suite to manage an infrastructure as a service (IaaS) cloud computing environment. All these products are designed to provide a secure, reliable, and highly available virtualization infrastructure.

The basic tasks of creating and accessing VMs on bare-metal virtualization software are similar to using desktop products: A wizard walks you through the procedures. The real differences lie in host and resource management and the capability to give IT managers the tools needed to virtualize a data center, not just one or two servers. This section serves as an introduction to the available products so that you have a starting point for doing your own research in the expanding field of virtualization.

Hands-On Project 11-7: Downloading and Installing VMware Workstation Player

Time Required: 20 minutes
Objective: Download and install VMware Workstation Player.
Required Tools and Equipment: Net-*XX*, and access to the Internet
Description: In this project, you download and install VMware Workstation Player.

Note

Even if you are running this project on a virtual machine, you can install VMware Workstation Player in your virtual machine, a technology called "nested virtualization." Nested virtualization allows you to run a virtual machine within another virtual machine; it is supported by VMware Workstation Player and VMware Workstation Pro.

1. If necessary, log on to your computer as **NetAdmin**.
2. Start a Web browser, and go to **www.vmware.com**. Click **Downloads**, and under Free Product Downloads, click **Workstation Player**.
3. Click the **Download Now** button under Try Workstation 15 Player for Windows. After downloading the file, start the installation of VMware Workstation Player. Follow the prompts, using the default options.
4. When the installation is finished, double-click the **VMware Workstation Player** shortcut on your desktop to start VMware Workstation Player. In the Welcome to VMware Workstation Player window, type your e-mail address to use VMware Workstation Player for noncommercial use, click **Continue**, and then click **Finish**.
5. You're ready to create a virtual machine. Leave VMware Workstation Player running for the next project.

Hands-On Project 11-8: Creating a Virtual Machine in VMware Workstation Player

Time Required: 15 minutes
Objective: Create a virtual machine in VMware Workstation Player.
Required Tools and Equipment: Net-*XX*
Description: In this project, you create a virtual machine in VMware Workstation Player. In Hands-On Project 11-9, you install an evaluation copy of Windows Server 2019 in the virtual machine.

Note

If you're running Net-*XX* as a virtual machine, you will need to make some changes to the default settings. If you're using VMware Workstation 10 or a later version, you need to enable virtualization extensions by turning off the Net-*XX* VM. Next, from the settings page of the VM, click Processors, click Virtualize Intel VT-x/EPT or AMD-V/RVI, and click OK. Restart the Net-*XX* VM, and then run VMware Workstation Player.

1. If necessary, log on to your computer as **NetAdmin**, and start VMware Workstation Player.
2. In the VMware Workstation Player Welcome window, click **Create a New Virtual Machine** to start the New Virtual Machine Wizard.
3. In the Welcome window of the New Virtual Machine Wizard, click the **I will install the operating system later** option button (see Figure 11-18), and then click **Next**.

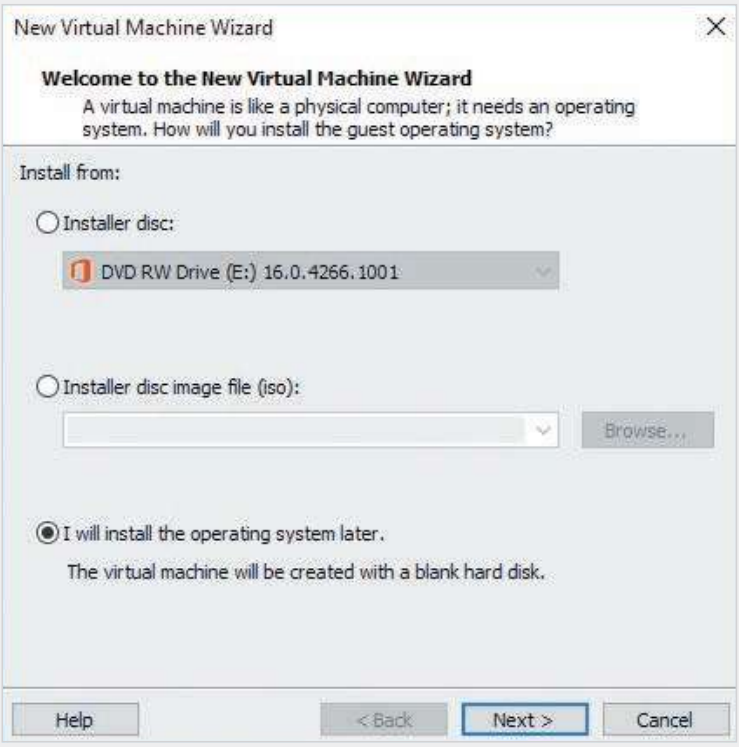

Figure 11-18 The New Virtual Machine Wizard

Source: VMware, Inc., *www.vmware.com*

4. In the Select a Guest Operating System window, make sure **Microsoft Windows** is selected, click the list arrow, and then click **Windows Server 2016**. (As of this writing, VMware Workstation Player does not yet have an option to install Windows Server 2019, but the Windows Server 2016 option will work for Windows Server 2019.) Click **Next**.
5. In the Name the Virtual Machine window, type **Windows Server 2019** and accept the default path, and then click **Next**.
6. In the Specify Disk Capacity window, accept the default **60.0** GB, and then click **Next**.
7. In the "Ready to Create Virtual Machine" window, review the options. If you wanted to change hardware options, you could click Customize Hardware, but for now, click **Finish**.
8. Before you install an OS in the virtual machine, take a look at some of VMware Workstation Player's features and options. The Windows Server 2019 VM you just created is shown in the left pane of VMware Workstation Player. Click to select **Windows Server 2019**, and in the right pane, click **Edit virtual machine settings** to open the Virtual Machine Settings dialog box (see Figure 11-19).

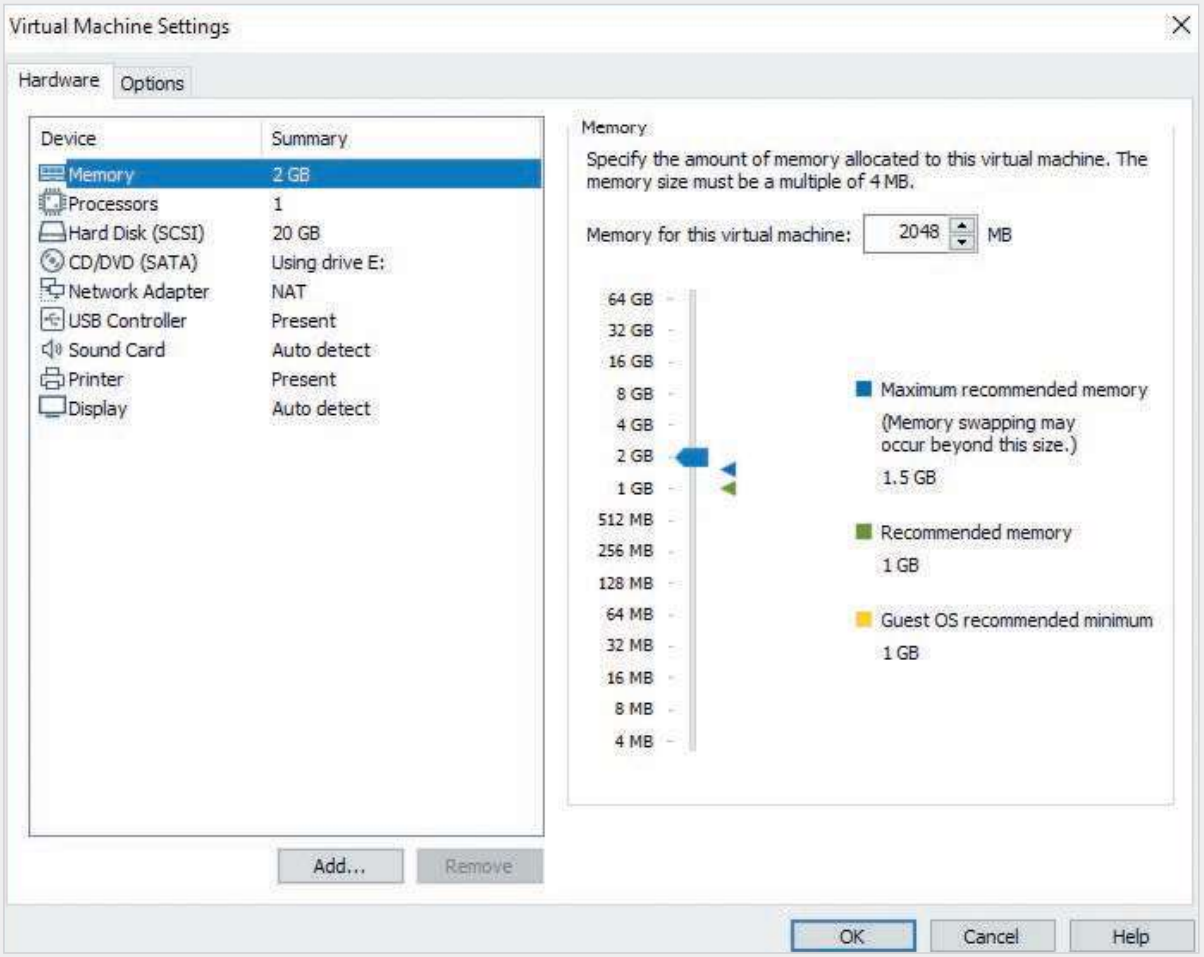

Figure 11-19 The Virtual Machine Settings dialog box

Source: VMware, Inc., *www.vmware.com*

9. In this dialog box, you can change the amount of memory allocated to the VM, change processor options, access disk utilities, change the virtual network settings, and configure many more settings. You can also add virtual hardware, such as hard disks and network adapters. Click **Network Adapter** in the list on the left. The default option is NAT, which means the VM gets an IP address from the host computer, and VMware performs NAT so that the VM can access the physical network. Click **Advanced** to see advanced network settings (see Figure 11-20). Notice that the VM is assigned a MAC address, just like a computer with a physical NIC. Click **Cancel**.

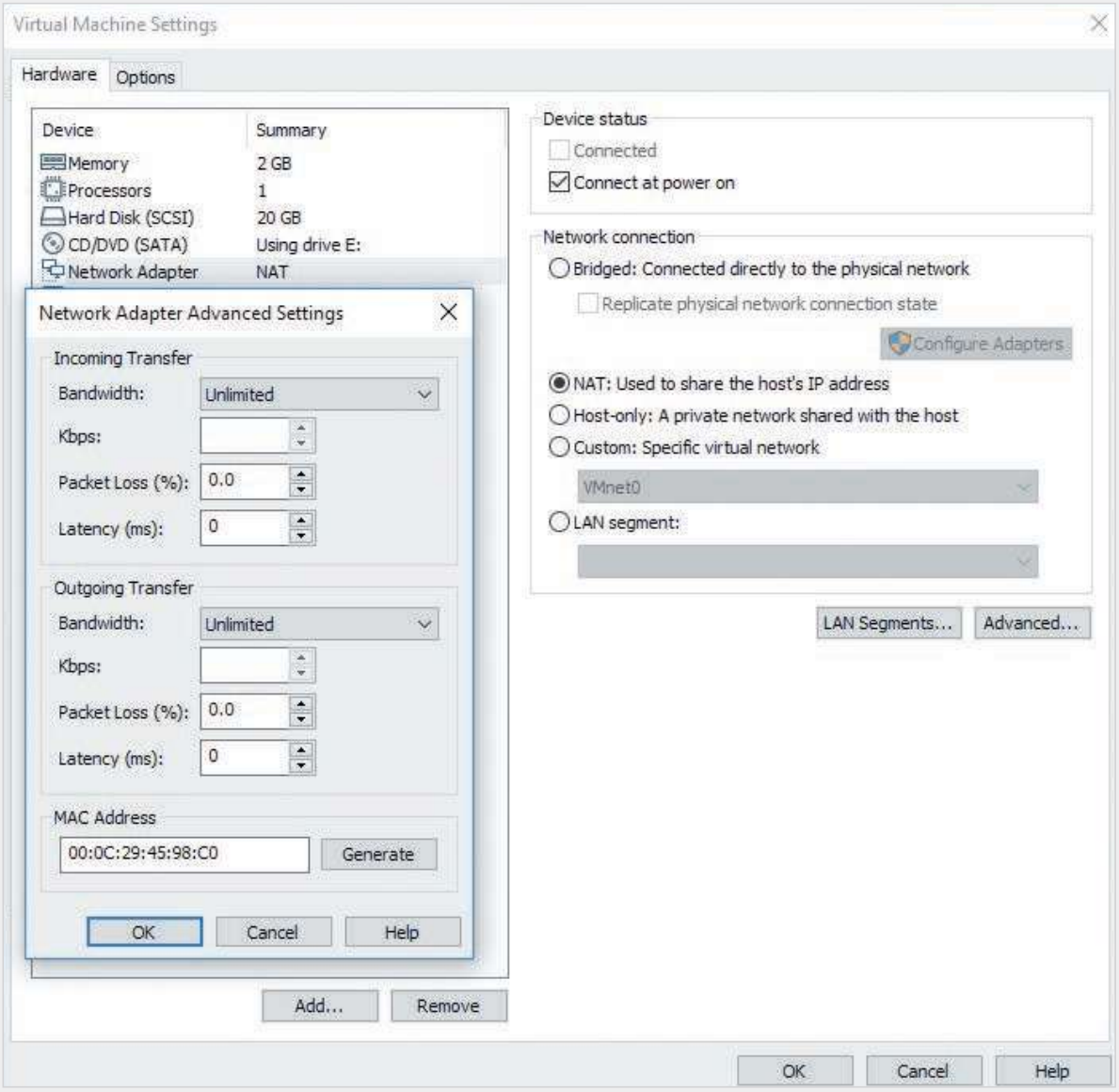

Figure 11-20 Advanced settings for network adapters

Source: VMware, Inc., *www.vmware.com*

10. In the Virtual Machine Settings dialog box, click **Add**. The Add Hardware Wizard lists virtual hardware you can add to the VM. Click **Cancel**.
11. In the Virtual Machine Settings dialog box, click the **Options** tab, where you can change options such as the VM name, the guest OS to install, power options, and shared folders. Click **Shared Folders**. If you enable shared folders, you can copy files from the host computer to the VM without having to create a network share. Click **Cancel** to close the Virtual Machine Settings dialog box.

12. Which virtual network setting should you use in VMware Workstation Player if you want the VM to be able to get an IP address from a DHCP server on the physical network?

_____

13. If you're continuing to the next project, leave VMware Workstation Player open; otherwise, log off or shut down your computer.

Installing an OS

Installing an OS, whether it's a desktop or server version, has become a no-brainer. Essentially, OSs install themselves; all you have to do is click Next and OK a few times and perhaps enter a license key and accept the license agreement. Even most Linux distributions, which in the past could stymie novices with a frustrating array of choices and options, are mostly hands-off installations now.

The real work of installing an OS, particularly on a network server, involves preinstallation and postinstallation tasks. The prerequisites for installing any OS are a copy of the installation medium and a computer that meets the installation requirements, including enough free (preferably unallocated) disk space, a CPU that meets minimum performance requirements, and enough RAM. The following sections explain the preinstallation planning process, the installation, and common postinstallation tasks for Windows Server and a common distribution of Linux.

Planning for and Installing Windows Server

Note

This section applies to most versions of Windows Server, starting with Windows Server 2008.

The role a server will play on the network is a key consideration in planning Windows Server installations. A server used only for file and printer sharing that supports a dozen users has different minimum hardware requirements than a server running Active Directory, a Web server, and a database and supporting a few hundred users. Windows Server is available in two primary editions, Standard and Datacenter, with the main differences being cost and included virtual machine licenses, so you need to determine which edition best fits your needs. After Windows is installed, you need to perform some postinstallation tasks immediately before installing additional features or applications.

Selecting Server Hardware for Windows Server

The minimum requirements for a server OS, although adequate for testing and training, are rarely satisfactory for a production server. So, a major factor to consider for a server OS installation is the server's hardware features. The following list describes a few features to consider before purchasing a server:

- *CPU architecture*—The minimum requirement is a 1.4 GHz CPU. CPUs are available in speeds well over 3 GHz, and major CPU manufacturers typically have a workstation line and a server line of processors. Depending on the expected server workload, you must also consider how many physical processors you need and how many cores each processor should have. Although Windows Server can run on just about any CPU that meets the minimum requirements, a CPU designed for servers (such as the Intel Xeon line of processors) usually has other server-specific components on the motherboard, such as high-end disk controllers and memory slots.

Note

Windows Server 2008 R2 is the first Microsoft OS that no longer supports a 32-bit CPU; you must use a 64-bit system for Windows Server 2008 R2 and later versions.

- *Disk subsystem*—For entry-level or departmental servers, SATA is a good choice because it's inexpensive and offers excellent performance. For enterprise servers or servers accessed around the clock, SAS disks have better performance and reliability but are more expensive than SATA. SAS disks are generally designed for continuous use; SATA drives tend to be designed more for consumer use, although most manufacturers have an enterprise line of SATA hard drives designed for servers. Researching current technology and your network's needs before deciding is best. RAID configurations that provide fault tolerance are inexpensive and highly recommended, considering their usefulness in the event of a disk failure. Windows Server requires only about 60 GB of free disk space, but you need additional space for data you store on the server. The OS should be installed on one disk (or RAID set), and at least one other disk (or RAID set) should be used for data and application storage. If you are running disk-intensive applications, you should strongly consider using SSDs in your server, as they provide considerably better performance than even the fastest mechanical disks.
- *Memory*—The minimum requirement is 512 MB of RAM, but only if you are installing Windows Server without a graphical user interface; the Desktop Experience installation option requires 2 GB of RAM. For testing or training purposes, Windows Server runs capably with these amounts, at least until you have more than a couple of users accessing the server or you want to install

several server roles. Server motherboards are typically equipped with more RAM slots than desktop systems are—and for good reason. After you start running database-driven Web applications, maintaining a few thousand users in Active Directory, or using virtualization on your servers, you often need 64, 128, or 256 GB (or even more) of RAM. Also, be aware that server memory usually costs more than desktop memory because it has features such as buffering and error correcting code (ECC) that make it more reliable.

This list covers just a few server hardware configurations you should consider before installing a server OS. The best advice is to forge a good relationship with a knowledgeable vendor you can consult when you need to make a purchase. This way, you can focus on managing your server, and your vendor can focus on keeping up with the latest hardware options.

Tip

To make sure your hardware selections are compatible with Windows Server, check the Windows Server Catalog at *www.windowsservercatalog.com.*

Selecting the Right Windows Edition

Windows Server 2019 comes in two primary editions, and one limited edition that targets different types of customers. These editions can be summarized as follows:

- Both Datacenter and Standard editions are full-featured server OSs that support up to 24 TB of RAM, up to 64 physical processors, and server clusters with up to 64 nodes per cluster. Only the virtual use limits and some advanced networking and storage options set them apart. For organizations using virtualization on a large scale, Datacenter Edition is clearly the best fit. A Datacenter Edition license allows you to install an unlimited number of virtual instances of the OS, meaning you can install Datacenter Edition with Hyper-V on a physical server and then install as many instances of Windows Server 2019 Datacenter Edition in virtual machines as you need. In addition, Datacenter Edition supports software-defined networking, a feature called Network Controller for virtual network management, and an advanced storage feature called Storage Spaces Direct.
- Standard Edition has all the features of Datacenter Edition except as noted above, and the same hardware limitations. The only other distinction (aside from price) is that a Standard Edition license permits only two virtual instances, so when you purchase Standard Edition, you can install it on a server, install the Hyper-V role, and then install Standard Edition on up to two virtual machines.
- Essentials Edition is aimed at small businesses with 25 or fewer users. It supports most of the roles and features in Standard and Datacenter editions, but some roles have restrictions or limited functions. For the price of the license (typically

around $500), you can install Essentials Edition one time on a physical server or a virtual machine, but not both. Essentials Edition is automatically configured as a domain controller. During installation, you're asked for the domain name, and Active Directory is installed automatically. Several other services are configured automatically in this edition: Active Directory Certificate Services, DNS, File Services, Web Server (IIS), Network Policy Server, and Remote Desktop Services. In addition, it comes with a front-end management interface called Dashboard that serves as a simplified server manager. Other features particular to this edition include client backups and Remote Web Access. This edition supports up to two physical processors and 64 GB of RAM.

Windows Server Preinstallation Decisions

When installing a new server in a network, you must make some decisions shortly after finishing the installation. Many of these configuration decisions should be made before you actually begin the installation so that you can dive right into postinstallation tasks. Some are fairly straightforward, but others take some thought and consultation. Here's a list of some decisions you need to make:

- What should you name the server? This decision is more important than it sounds. Every computer needs a name so that it can be identified on the network. A server name must be unique on the network and should include some description, such as its location or primary function. Server names should also be simple and easy to remember because users often access servers by name.
- Which network protocols and addresses should you use? By default, Windows installs both TCP/IPv4 and TCP/IPv6. You can't uninstall them, but you can disable them in a network connection's Properties dialog box. Disabling TCP/IPv6 or TCP/IPv4 isn't recommended, however, as some network services depend on these protocols. Windows has no additional protocol or client options, so if you need something that isn't already installed, you must find a third-party solution.
- How should I assign an IP address to the server? By default, Windows Server is configured to use DHCP, but a server should have a static IP address. Some server roles, such as DHCP, require assigning a static address. If you haven't devised an addressing scheme, now is the time to do that. You might want to reserve a bank of addresses in the beginning or end of the address range for your servers, such as 192.168.1.1 to 192.168.1.20 or 192.168.1.230 to 192.168.1.250. Whatever you decide, be consistent so that when more servers are added, you can assign addresses easily.
- Setting the correct time zone isn't really a decision but a task you must complete because having the wrong time zone can cause all manner of problems, particularly in a domain environment. Certain functions in a domain network, such as user authentication, depend on client and server computers having their clocks synchronized within a few minutes of each other.
- Should I use the workgroup or domain model? The Windows domain model has several advantages in usability, manageability, and security. If you've invested in a Windows server OS, it makes sense to get the most out of it by using the

domain model and installing Active Directory. With a small network of fewer than 10 users, however, the workgroup model is a viable option, particularly if the main administrator isn't familiar with Active Directory. With either model, you need a workgroup or domain name, unless you're using the workgroup model and keep the default name "Workgroup." If you're using the domain model, you need to decide whether the domain name will be registered on the Internet. If it isn't, many Active Directory administrators use the top-level domain name "local," such as mycompany.local.

- What services should you install? This decision is one of the most important because it determines how the server will be used and what network services will be available to users. Windows Server refers to services such as Active Directory, DNS, and DHCP as "server roles." With the domain model, you must install Active Directory on at least one server. Active Directory requires DNS, so the DNS Server role is installed automatically. Other basic roles to consider on a first server include DHCP (for IP address configuration) and File and Storage Services, which includes tools for sharing and managing file storage. You can install many other roles and features to meet your network and business needs.

After you have a plan, it's time to move on to the actual installation of Windows Server. Instead of reviewing installation steps here, you can do Hands-On Project 11-9, which walks you through installing Windows Server 2019 in a virtual machine.

Windows Server Postinstallation Tasks

After Windows Server is installed, it's time to attend to some postinstallation tasks. Some were discussed earlier, such as naming the server and configuring protocols and addresses. Here's a summary of the tasks you should perform immediately after installation:

- Activate Windows Server.
- Set the correct date, time, and time zone.
- Assign a static IP address.
- Assign a computer name.
- Download and install available updates.
- Add and configure roles and features.

Windows Server requires activation within 60 days after installation. After 60 days, you can't log on until you activate it. Windows Server tries to activate automatically after several days, or you can activate it manually by clicking "Activate Windows now" in the System Properties dialog box. As you can see, most of the work of installing Windows Server is in the planning and postinstallation tasks. The same is true of most Linux installations, which are covered next.

Planning for and Installing Linux

Planning for a Linux server installation isn't much different from planning a Windows Server installation. Minimum hardware requirements must be met, and more important, hardware requirements for the role the server will play in your network

must be met. Linux has come a long way in hardware compatibility but still doesn't have the broad support for different hardware that Windows does.

Tip

To research hardware compatibility for Linux distributions, go to *www.linux-drivers.org*.

One of the biggest decisions to make before you install Linux is choosing which distribution to use. There are so many distributions, each with its own target audience, that making a recommendation without knowing the intended environment is impossible. A Web site called *DistroWatch.com* lists dozens of distributions along with descriptions and links to get more information. Most Linux distributions are open source and governed by the GNU General Public License (GPL), which allows users to run the program for any purpose, make changes to the program, and redistribute the program to others under the same GPL license terms.

After deciding on a Linux distribution, the next step is downloading a disk image of the installation medium and burning it to a DVD. Many Linux distributions are offered as a Live install that you can use to boot your system (physical or virtual) from the DVD and run the OS without having to install it on a hard drive. Running a Live install isn't a replacement for installing the OS on a disk, but it's a good way to evaluate a distribution. In addition, many specialized Linux distributions are available as Live installs and contain disk and system repair utilities to help you fix a Linux or Windows installation.

Tip

You can find a list of Live Linux installs at *www.livecdlist.com*.

The preinstallation and postinstallation tasks for a Linux OS aren't very different from those for Windows Server, except there's no need to activate Linux; also, most tasks, such as IP address assignment and time zone selection, are done during the Linux installation.

Although installing Linux isn't difficult, it requires more input and decision-making during installation, whereas almost all configuration decisions in Windows are made after installation. Linux is a popular server OS, particularly for running Web applications and applications that use databases. Compared with Windows, it offers all the basic infrastructure services, such as DHCP and DNS, but lacks a comprehensive directory service, such as Active Directory. Also, although more Linux services can be

managed in a GUI, Linux still tends to make heavy use of the command line, which can be a drawback for administrators who are more at home with a GUI. Most large network environments use a combination of Windows and Linux servers, placing them in roles where they excel.

Hands-On Project 11-9: Installing Windows Server 2019 in a VM

Time Required: 30 minutes or longer
Objective: Install Windows Server 2019 in a VM.
Required Tools and Equipment: Net-*XX*, and the Windows Server 2019 ISO file
Description: In this project, you download and install the trial version of Windows Server 2019 in the VM you created earlier. You need the Windows Server 2019 ISO file (which contains an image of a DVD). Your instructor can make this file available to you, or you can download it at *https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2019*. You need to sign in to the Microsoft Web site with a Microsoft account before you can download the file.

1. If necessary, log on to your computer as **NetAdmin**, and start VMware Workstation Player. In the left pane of the Welcome window, click **Windows Server 2019**, and then click **Edit virtual machine settings**. Click **CD/DVD**, and then click **Use ISO image file**. Click the **Browse** button, navigate to where the ISO file is stored on your computer, click the ISO file, and then click **Open** (see Figure 11-21).

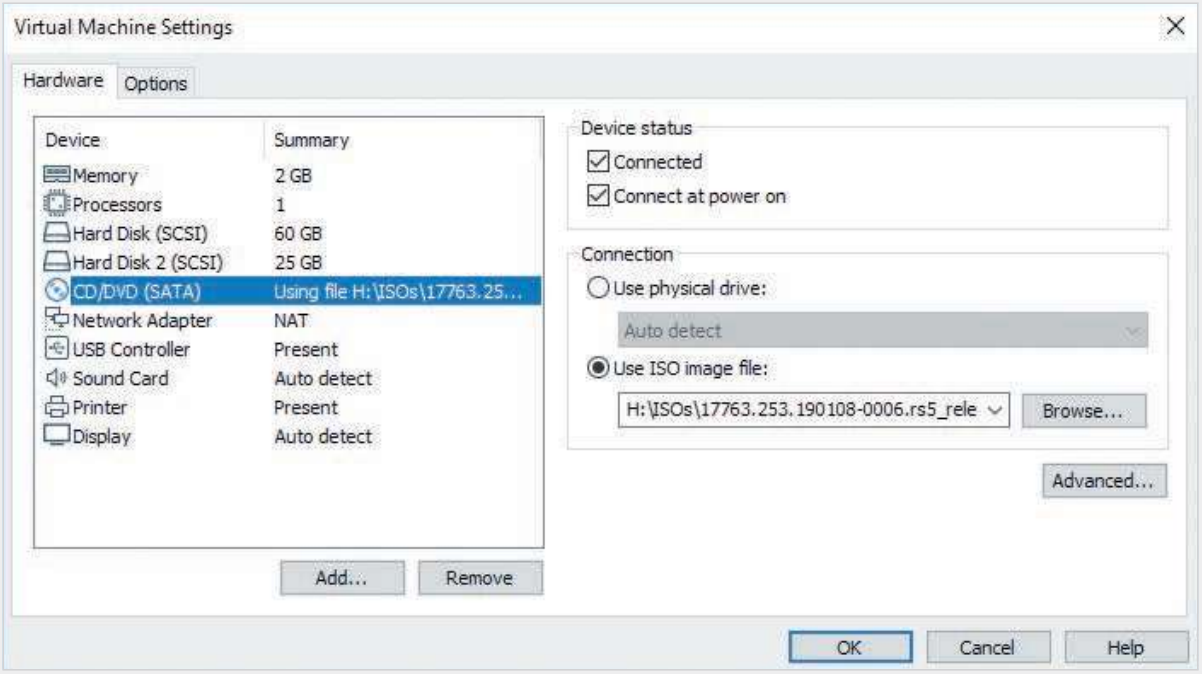

Figure 11-21 Using an ISO file to install Windows Server 2019
Source: VMware, Inc., *www.vmware.com*

2. Click **OK** to close the Virtual Machine Settings window, and then click **Play virtual machine**. You need to press a key when you see the message "Press any key to boot from CD or DVD." The VM boots to the Windows Server 2019 installation ISO file, and you see the Windows Setup window (shown in Figure 11-22).

Figure 11-22 The Windows Setup window

3. Accept the default options or change the language options, if necessary. Click **Next**, and then click **Install now**.
4. In the "Select the operating system you want to install" window, click **Windows Server 2019 Standard Evaluation (Desktop Experience)**, which is the second installation option. Click **Next**.
5. In the "License terms" window, click **I accept the license terms**, and then click **Next**.
6. In the "Which type of installation do you want?" window, click **Custom: Install Windows only (advanced)**.
7. In the "Where do you want to install Windows?" window, accept the default option, and then click **Next**. Windows begins copying files. The installation might take 10 minutes or more.
8. After the installation is finished, the VM restarts, and the Customize settings dialog is shown. Set the Administrator password by typing **Password01** twice, and then

clicking **Finish**. You see a message that Windows is finalizing your settings, and you're asked to log on.

9. When prompted to press Ctrl+Alt+Delete to sign in, instead press **Ctrl**+**Alt**+**Insert** or click the **Ctrl**+**Alt**+**Delete** icon on the VMware Workstation Player menu. Type **Password01** and press **Enter**. After a short time, you see the desktop, Server Manager opens, and then you're ready to go. If you see a Network prompt, click **Yes**.
10. Now you're ready to perform postinstallation tasks, such as setting the computer name, IP address, and time zone. In Server Manager, click **Local Server** to see the default settings.
11. You can configure the server in Challenge Lab 11-1. For now, right-click **Start** on your virtual machine, point to **Shut down or sign out**, and click **Shut down**. When you click **Continue**, the VM shuts down and VMware Workstation Player closes. Log off or shut down your computer unless you're going on to the challenge labs.

Chapter Summary

- A computer's OS provides services that enable users and devices to interact with the computer and manage the computer's resources. These services include a file system, process and service management, and the kernel.
- File systems provide a method for storing, organizing, and managing access to files on a storage device, such as a hard drive. In addition, they provide an indexing system for fast file retrieval and permissions for securing access to files.
- A process is a program that's loaded into memory and run by the CPU. It can be an application a user interacts with or a program with no user interface that communicates with and provides services to other processes. The latter type of process is called a service.
- The kernel schedules processes to run, making sure high-priority processes are taken care of first; manages memory to ensure that two applications don't attempt to use the same memory space; and makes sure I/O devices are accessed by only one process at a time, in addition to other tasks.
- Client OSs include many features once reserved for a server OS, such as file and printer sharing and file system security, but an OS designed to be installed as a server still contains many additional networking and fault-tolerance features not found in client OSs.
- Virtualization can be divided into two categories: hosted virtualization and bare-metal virtualization. Hosted virtualization products are installed on a desktop OS and include VMware Workstation, Virtual PC, and VirtualBox. Bare-metal virtualization software is used in data centers, is installed on servers, and includes products such as Microsoft Hyper-V, VMware vSphere, and Citrix XenServer.

- The real work of installing an OS consists of preinstallation and postinstallation tasks. The prerequisites for installing any OS are a copy of the installation medium and a computer that meets the hardware requirements, including enough free (preferably unallocated) disk space, a CPU meeting minimum performance requirements, and enough RAM.
- Some features to look for in a server system include CPU architecture, disk subsystem, and amount of memory. You must also be sure to select the correct edition of the OS you're going to install.
- Preinstallation decisions include the server name, the protocols to use, the networking model (domain or workgroup) you should use, and the services that should be installed. Postinstallation tasks include activating the OS if necessary, setting the correct date and time, configuring IP settings, configuring the computer name, installing updates, and installing roles and features.

Key Terms

authentication
authorization
bare-metal virtualization
batch file
cloud storage
context switching
cooperative multitasking
disk cluster
failover cluster
file system
guest OS
host computer
hosted virtualization
hot-swappable device
hypervisor
load-balancing cluster
logon script
multiprocessing
multitasking
multithreaded application
network appliance
network-attached storage (NAS)
preemptive multitasking
process
redirector
redundant array of independent disks (RAID)
server cluster
service
snapshot
storage area network (SAN)
thread
time slicing
virtual disk
virtual machine (VM)
virtual network
virtualization

Review Questions

1. Which of the following is an objective of a file system? (Choose all that apply.)
   a. Organize space on a drive.
   b. Organize files hierarchically.
   c. Schedule access to applications.
   d. Secure access to files.

2. A cluster is composed of which of the following?
   a. One or more 512-bit blocks
   b. Two or more 2K-byte sectors
   c. One or more 512-byte sectors
   d. One or more 2K-byte blocks