

21-1Linux Operating System

Core 2 Objectives

- 1.8

Explain common OS types and their purposes.

- 1.11

Identify common features and tools of the Linux client/desktop OS.

- 4.9

Given a scenario, use remote access technologies.

A variation of UNIX is Linux (pronounced “Lih-nucks”), an OS created by Linus Torvalds when he was a student at the University of Helsinki in Finland. Basic versions of this OS are open source, and all the underlying programming instructions (called source code) are freely distributed. Linux is popular because it’s inexpensive and very stable (it seldom crashes). Linux is used as an OS for desktops, servers, mobile devices, and even IoT devices. It’s the most popular OS for server applications such as web servers and email servers. In addition, Android and Chrome OS for mobile devices are based on Linux, and bootable CDs and flash drives that contain utility software often use Linux. Versions of Linux are called distributions or flavors; the more popular ones for desktops and servers are listed in [Table 21-1](#). Hardware requirements for Linux vary widely by distribution.

Table 21-1

Popular Linux Distributions for Desktops and Servers

Name	Comments	Website
Arch Linux	Arch Linux must be manually configured. It has excellent online documentation and community support, and Linux professionals appreciate its simplicity.	archlinux.org
Fedora	Fedora has been around for a long time and is backed by a stable company. When updates are released, they tend to work well without errors. It’s a great distribution for just about any OS purpose.	getfedora.org
Linux Mint	Linux Mint is based on Ubuntu with several features added. Mint is great for users who are familiar with the Windows desktop environment, as its appearance is similar.	linuxmint.com
openSUSE	openSUSE is made for servers, desktops, and mobile devices. Applications install without a hassle; go to software.opensuse.org , select an app, and perform a Direct Install.	openSUSE.org
Red Hat Enterprise Linux	Designed for enterprise use on servers and workstations, this commercial distribution is stable and comes with long-term support.	redhat.com

Name	Comments	Website
Ubuntu	Ubuntu is one of the most popular distributions of Linux for desktops and servers, and it comes with tons of online tutorials and help.	ubuntu.com



Note 1

For more information on Linux, see linux.org as well as the websites of the different Linux distributors.

Linux itself is not a complete operating system; it is only the kernel for the OS. You also need a shell for user and application interfaces, and Linux shells vary widely by distributions. Many distributions of Linux include a GUI shell or desktop, which is called a windows manager. For example, [Figure 21-1](#) shows the desktop or windows manager for Ubuntu Desktop. Some distributions of Linux designed for server applications don't have a windows manager. For example, Ubuntu Server installs with only a command-line interface. In this module, we use Ubuntu Desktop and Ubuntu Server as our sample Linux distributions.

Figure 21-1

Ubuntu Desktop with the Mozilla Firefox browser window open

You can install Ubuntu Desktop or Ubuntu Server in a VM or on a hard drive, CD, or USB flash drive. When you install Ubuntu on a CD or USB flash drive, it is called a **Live CD** or **Live USB**. A Live CD or USB can boot up a live version of Linux, complete with Internet access and all the tools you normally have available in a hard drive installation of Linux, but without installing the OS on the hard drive.

Let's first install and explore Ubuntu Desktop with its graphical interface, and then you'll learn to install and use Ubuntu Server with its command-line interface.

Applying Concepts

Installing Ubuntu Desktop in a VM

- **Est. Time:** 1 hour
- **Core 2 Objectives:** 1.8,1.11

Follow these steps to install Ubuntu Desktop in a VM:

1. **1**

Recall that you learned how to install a hypervisor and create a VM in the module “[Installing Windows](#).” If you don’t already have a hypervisor installed, install one that you can use to manage VMs. For example, in 64-bit Windows 10/11 Pro, you can use the Programs and Features window to enable Client Hyper-V, which comes embedded in the OS. Alternately, you can download and install one of these free hypervisors:

- Oracle VirtualBox at [virtualbox.org/wiki/Downloads](https://www.virtualbox.org/wiki/Downloads)
- VMware Workstation Player at https://customerconnect.vmware.com/en/downloads/info/slug/desktop_end_user_computing/vmware_workstation_player/16_0

2. **2**

Go to ubuntu.com/download/desktop and download the free Ubuntu Desktop OS to your hard drive. The file that downloads is an ISO file.

Note 2

Ubuntu Desktop is only available as a 64-bit OS. To install a 64-bit guest OS in a VM, the host OS must also be 64 bit. As of this publication, the LTS (long-term support) Ubuntu version is 20.04.3.

1. **3**

Open your hypervisor software, and create a new VM with at least 4 GB of RAM and at least a 25 GB virtual hard drive capacity. Mount the ISO file that contains the Ubuntu Desktop download to a virtual DVD in your VM.

2. **4**

Start up the VM, and install Ubuntu Desktop in the VM, accepting all default settings. Be sure to write down the name of the VM and your Ubuntu host name, Ubuntu user name, and password. When given the option, decline to install any extra software bundled with the OS. If needed, the software can be installed later.

Note 3

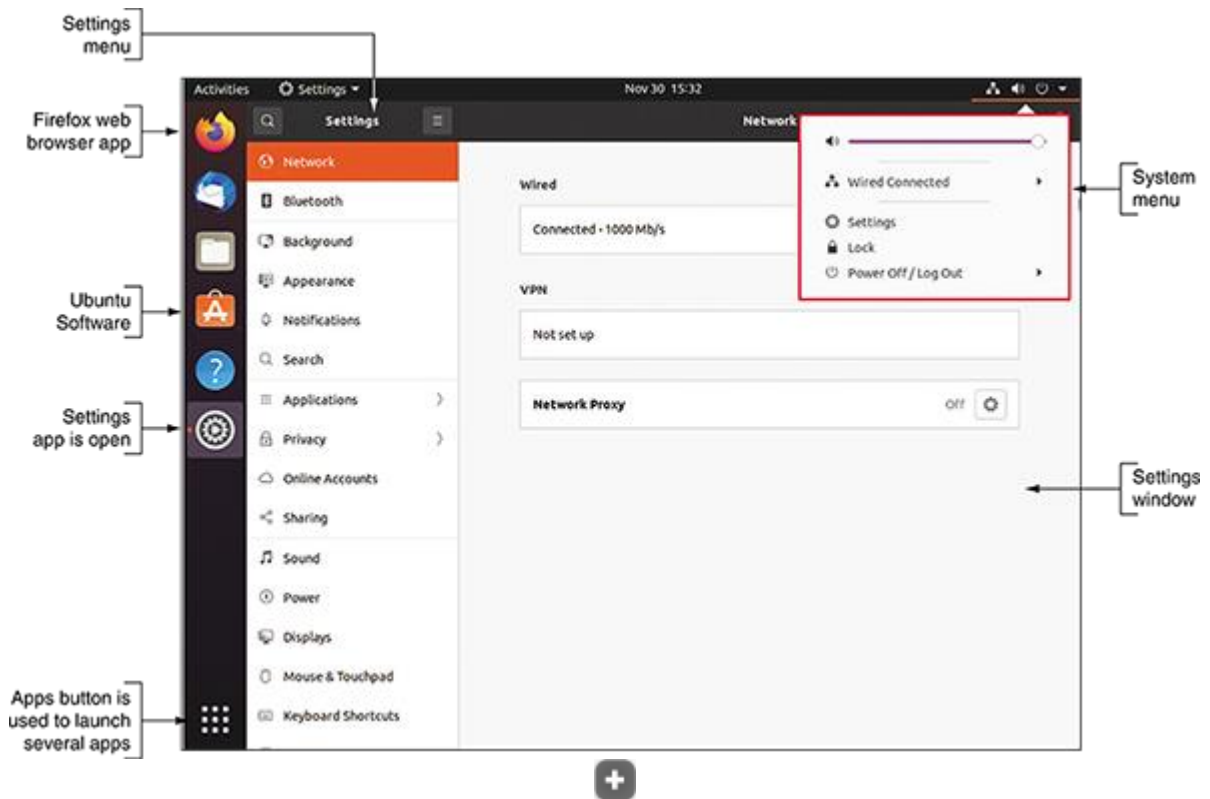
If you need help learning to use your hypervisor of choice, try searching for some tutorial videos at [youtube.com](https://www.youtube.com) or on the hypervisor manufacturer’s website.

1. **5**

When asked to restart the VM, first dismount (remove) the ISO file from the optical drive so the VM boots to the hard drive. After Ubuntu Desktop launches, log in with your user name and password. [Figure 21-2](#) shows the desktop with the Settings window active and the Network setting selected. To open this window, click the **Apps** button, and then click **Settings**. When a window is active, its menu appears at the top of the screen. Also notice the System menu is displayed. To open the System menu, click the **system icons** area in the upper-right corner of the screen.

Figure 21-2

Ubuntu Desktop with the Settings window open



2. **6**

Take a few minutes to poke around the desktop. You'll see how it resembles macOS in many ways. For example, to open a terminal window where you can enter Linux commands, click **Activities**, and type **terminal** in the search box. Click **Terminal** in the list that appears. Using the terminal, you can enter Linux commands to manage and support the OS.

3. **7**

To shut down Ubuntu Desktop, click the system icons area in the upper-right corner of the screen. The System menu opens. Click **Power Off/Log Out**, and then click **Power Off**.

The Ubuntu Desktop Guide at help.ubuntu.com/stable/ubuntu-help is an excellent resource if you want to learn more about using Ubuntu Desktop.

21-1a Linux Installs, Updates, and Backups

Core 2 Objective

- 1.11

Identify common features and tools of the Linux client/desktop OS.

When supporting a Linux system, an IT technician needs to know how to install software, update the OS and apps, and create and maintain scheduled backups. Here are a few details to get you started:

- **Install software.** To install software, click the **Ubuntu Software** icon. Ubuntu Desktop software opens. Click an item and then click **Install** to start its installation. You can also install an app by clicking a setup file you have downloaded from the web.
- **Update Ubuntu and apps.** To update, click **Updates** in the Ubuntu Software window. Updates for Ubuntu are listed first, followed by app updates. Click **Install** beside the updates you want to install. [Figure 21-3](#) shows that no updates are available.

Figure 21-3

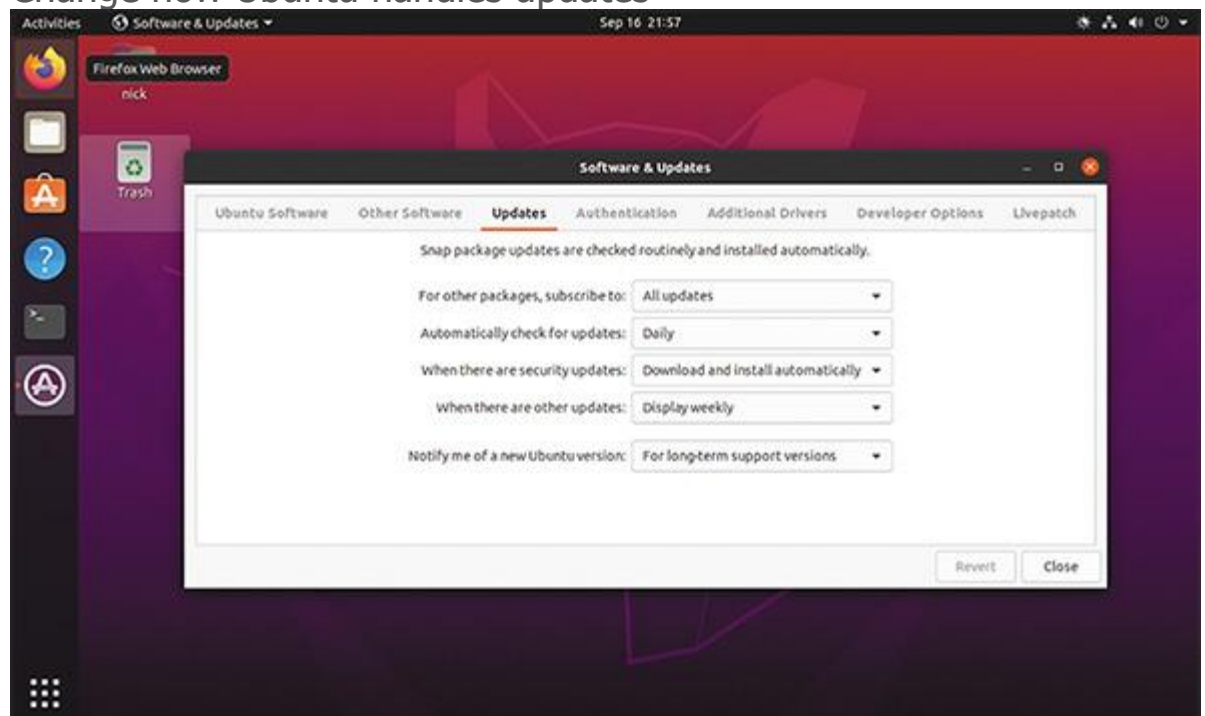
Ubuntu Desktop is up to date



- **Change update settings.** To change update settings, open the **Apps** menu, and click **Software & Updates** in the drop-down list. Then click the **Updates** tab in the Software & Updates window (see [Figure 21-4](#)).

Figure 21-4

Change how Ubuntu handles updates





- **Configure backups.** Click the **Apps** button. In the list of apps, scroll down and click **Utilities** (see [Figure 21-5](#)). Click the **Backups** utility. Using the Backups window (see [Figure 21-6](#)), you can schedule backups, decide where the backups are stored, and select which items are included in the backup.

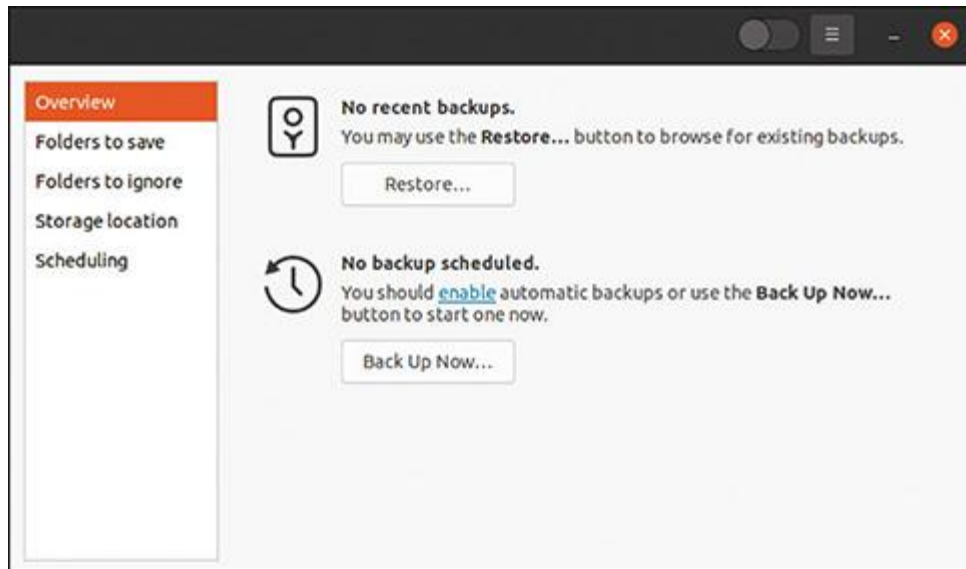
Figure 21-5

Ubuntu Desktop utilities



Figure 21-6

Schedule an Ubuntu Desktop backup



- **Antivirus and anti-malware solutions.** As with every computing environment, you should protect your Linux system with an antivirus or anti-malware solution. Examples of solutions that have been developed specifically for Linux include the ClamAV, Comodo, and Sophos applications. An antivirus solution should be updated regularly to ensure system protection is maintained. Some vendors provide updates multiple times a week or even daily. Externally connected media and downloaded files should be scanned to minimize the threat of malware infecting your system.

Applying Concepts

Installing ClamAV

- **Est. Time:** 15 minutes
- **Core 2 Objective:** 1.11

To use ClamAV to protect your previously installed VM installation of Ubuntu against malware, follow these steps:

1. **1**
Launch your VM, and log in using your root credentials.
2. **2**
Open a terminal window, and update your system by running the following command:

```
sudo apt-get update
```
3. **3**
After your system is updated, run the following command to install ClamAV:

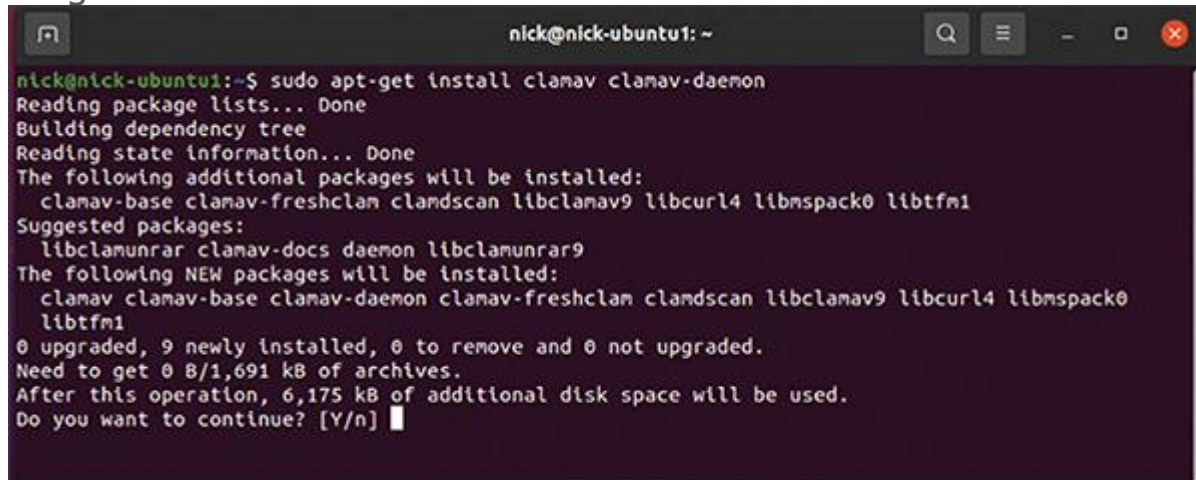
```
sudo apt-get install clamav clamav-daemon
```
4. **4**

Press **Enter** when prompted to continue. (Because the Y is uppercase, as shown in [Figure 21-7](#), the default answer to the prompt is Yes.) After the installation finishes, enter the following command to confirm that ClamAV was installed correctly:

```
clamscan --version
```

Figure 21-7

Installing ClamAV on Ubuntu

A terminal window titled 'nick@nick-ubuntu1: ~' showing the command 'sudo apt-get install clamav clamav-daemon'. The output displays the package lists, dependency tree, and a list of additional packages to be installed. It also shows the disk space requirements and a confirmation prompt 'Do you want to continue? [Y/n]' with 'Y' entered.

```
nick@nick-ubuntu1:~$ sudo apt-get install clamav clamav-daemon
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  clamav-base clamav-freshclam clamdscan libclamav9 libcurl4 libnsspack0 libtfn1
Suggested packages:
  libclamunrar clamav-docs daemon libclamunrar9
The following NEW packages will be installed:
  clamav clamav-base clamav-daemon clamav-freshclam clamdscan libclamav9 libcurl4 libnsspack0
  libtfn1
0 upgraded, 9 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/1,691 kB of archives.
After this operation, 6,175 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

If a version shows for ClamAV, then the installation was successful. In [Figure 21-8](#), we can see the current version of ClamAV is 0.103.2.

Figure 21-8

Terminal window showing ClamAV installed version

A terminal window showing the output of the 'clamscan --version' command. The output displays the ClamAV version and the date of the last update.

```
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
nick@nick-ubuntu1:~$ clamscan --version
ClamAV 0.103.2/26297/Thu Sep 16 09:59:37 2021
```

Now that you have learned some of the basics about Ubuntu Desktop and have installed an antivirus solution, let's turn our attention to Ubuntu Server.

21-1b Installing and Exploring Ubuntu Server

Core 2 Objectives

- 1.8

Explain common OS types and their purposes.

- 1.11

Identify common features and tools of the Linux client/desktop OS.

Ubuntu Server does not include a windows manager but instead uses a command-line interface called the terminal. The default shell for the terminal is the **Bash shell**, which stands for “Bourne Again Shell” and takes the best features from two previous shells, the Bourne and Korn shells. In this module, we use Ubuntu Server and its default Bash shell. In Linux, a command prompt in the terminal is called a **shell prompt**.

Note 4

To find out what shell is the default shell for the Linux system, enter the **echo \$SHELL** command. To find out which shell you are currently using, enter the **echo \$0** command.

Applying Concepts

Installing Ubuntu Server in a VM

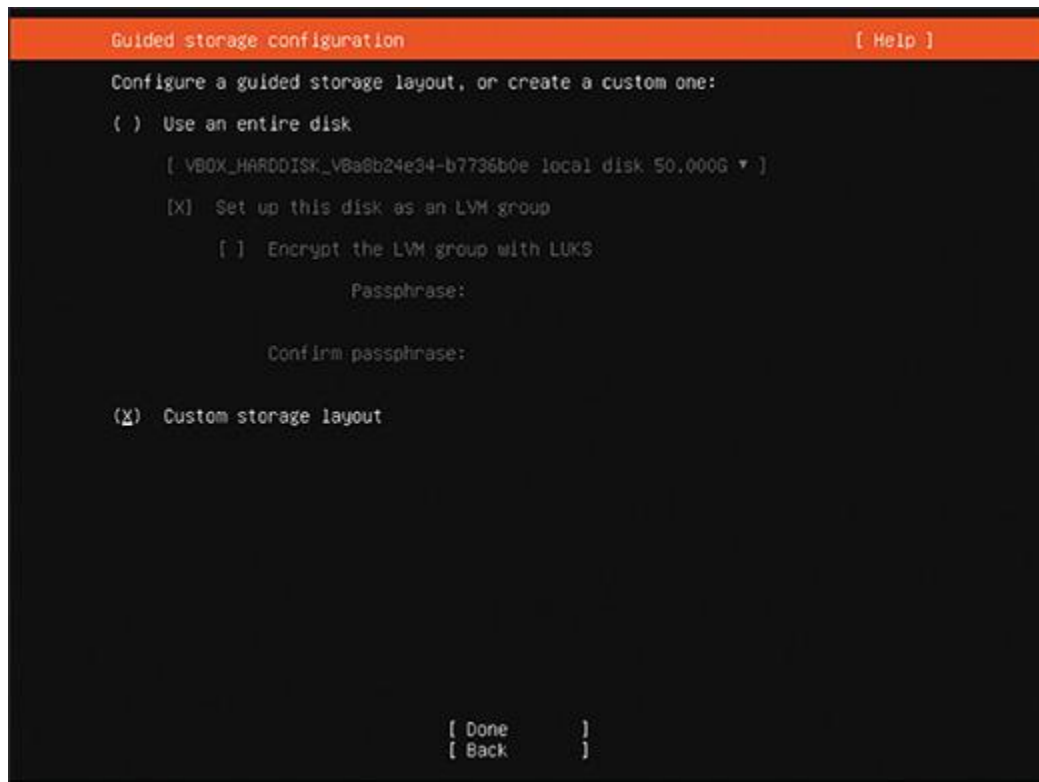
- **Est. Time:** 15 minutes
- **Core 2 Objectives:** 1.8,1.11

To practice Linux skills covered in this module, you need an installation of Ubuntu Server. Before you continue with this module, follow these steps to install Ubuntu Server in a VM on a Windows computer:

1. **1**
Go to ubuntu.com/download/server and download the Ubuntu Server OS to your hard drive. The file that downloads is an ISO file.
2. **2**
Open your hypervisor software, and create a new VM. For Ubuntu Server, you'll need at least 4 GB of RAM and at least a 25 GB virtual hard drive. For this example, we selected a 50 GB virtual hard disk. Mount the ISO file that contains the Ubuntu Server download to a virtual DVD in your VM.
3. **3**
Start up the VM, and install Ubuntu Server, accepting all default settings until you get to the Guided storage configuration screen (see [Figure 21-9](#)). To improve Ubuntu performance, you can set up a **swap partition**, which is used to hold virtual memory. In Linux, a swap file or swap partition can hold virtual memory, but a swap partition yields better performance. Select **Custom storage layout**, as shown in the figure, so you can create three partitions on the hard drive.

Figure 21-9

Partition the hard drive to include a swap file



Note 5

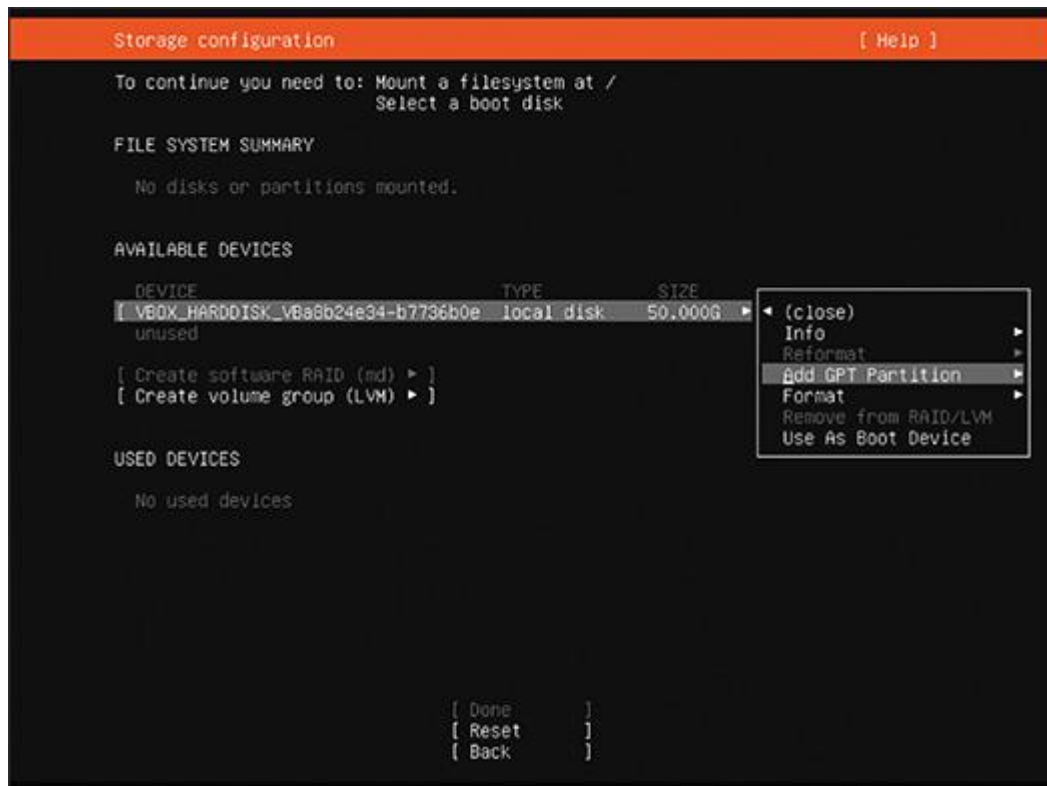
When installing Ubuntu, use the arrow keys to navigate menus, and press **Enter** to select. The **Esc** key takes you back one level in a menu.

1. 4

On the next screen, under AVAILABLE DEVICES, select the first device listed, and then press **Enter**. In the drop-down menu that appears, select **Add GPT Partition**, as shown in [Figure 21-10](#).

Figure 21-10

Create a new partition on the drive

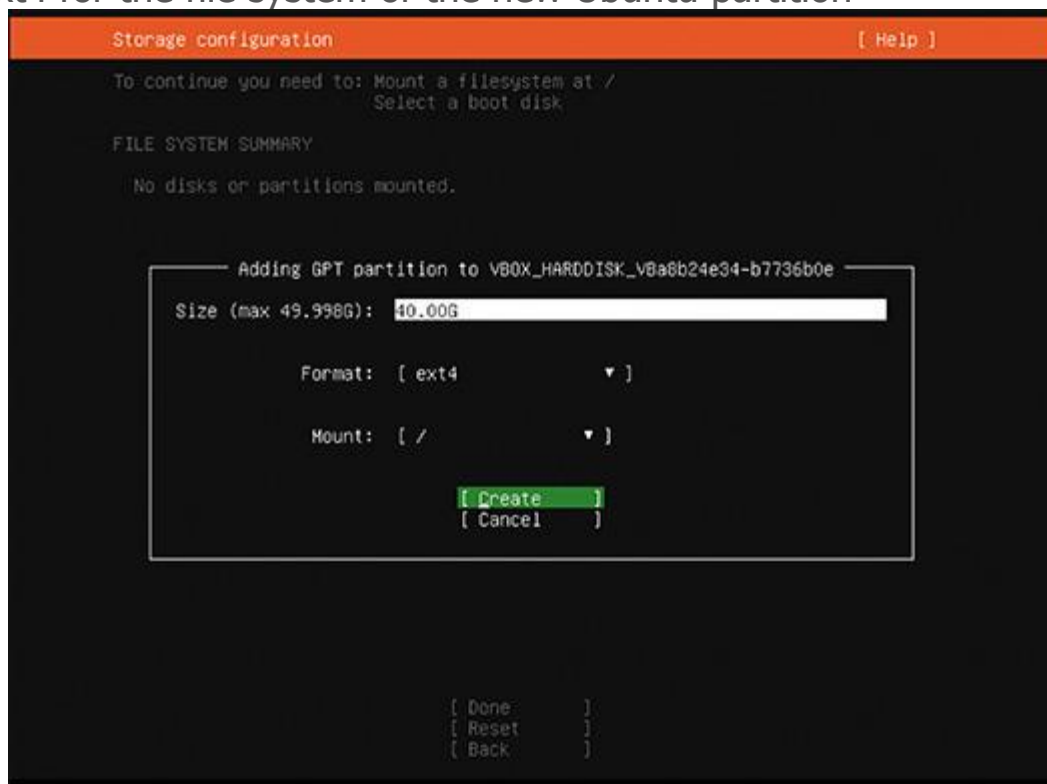


2. 5

In the Adding partition box, enter the size of the boot partition (at least 25 GB), and select the file system type **ext4**, as shown in [Figure 21-11](#).

Figure 21-11

Select ext4 for the file system of the new Ubuntu partition



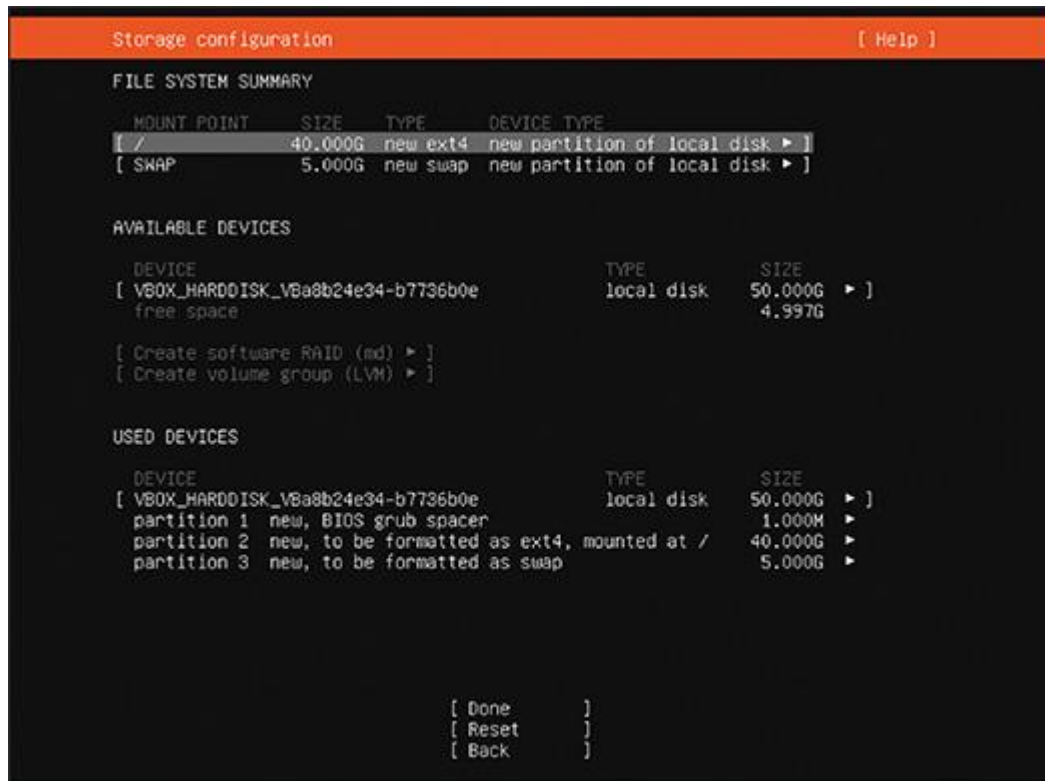
3. 6

Add a second partition, and select **swap** as the file system type. Make the partition size a little larger than the amount of installed memory. [Figure 21-12](#) shows the resulting Storage configuration screen with three partitions created. Notice that not all the hard drive is partitioned in our example:

- **Partition 1.** The small BIOS grub spacer partition is automatically created and is used to boot the system in a dual-boot environment. **GRUB (GR and Unified Bootloader)** is a boot loader used to manage dual-boot systems.
- **Partition 2.** The 40 GB partition that will hold the OS uses the ext4 file system.
- **Partition 3.** The 5 GB swap partition is used to hold virtual memory. A swap partition does not have a file system installed.

Figure 21-12

Partitions are created to hold the Ubuntu installation



4. **7** Select **Done** to complete the installation, accepting all defaults. Be sure to write down the name of the VM and your Ubuntu host name, Ubuntu user name, and password. When given the option, decline to install any extra software bundled with the OS. Notice this software includes add-ons to be used when Ubuntu Server is installed in the cloud using Amazon Web Services or Google Cloud. If needed, the software can be installed later.
5. **8** After the VM restarts, Ubuntu Server launches, and you should see the terminal shell in the VM. See [Figure 21-13](#).

Figure 21-13

When you log in to Ubuntu Server, available updates are listed

```
Sep 2021 02:07:36 +0000. DataSource DataSourceNone. Up 42.05 seconds
[ 42.069820] cloud-init[1350]: 2021-09-18 02:07:36,426 - cc_final_message.py[WARNING]: Used fallback
ck datasource
[ OK ] Finished Execute cloud user/final scripts.
[ OK ] Reached target Cloud-init target.

spock login: evan
Password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-84-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Sat 18 Sep 2021 02:17:22 AM UTC

System load:  0.0           Processes:           91
Usage of /:   13.2% of 39.12GB Users logged in:       0
Memory usage: 6%           IPv4 address for enp0s3: 192.168.128.57
Swap usage:   0%

6 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

evan@spock:~$
```

Six updates available

Shell prompt

6. **9** You might need to press **Enter** to see the shell prompt. Then enter your user name and password, and you're logged in to Ubuntu Server. In [Figure 21-13](#), the server is named spock and the logged-in user is evan. Notice that Ubuntu reports that six packages can be updated. In Ubuntu, a **package** is the collection of files needed to install software.
7. **10** When you are ready to shut down Ubuntu Server, run the **sudo shutdown now** command.

Typically, the shell prompt includes the user name, host name, and the current directory, followed by a \$. For example, in [Figure 21-13](#), the shell prompt shows the user name is evan, and the host name is spock. The ~ character indicates the user's home directory, which for the evan account is /home/evan. When you first log in to Linux, the current directory is always the home directory of the logged-in user. (In Linux, directories in a path are separated with forward slashes, in contrast to the backward slashes used by Windows.)

It's easiest to install a swap partition when you install Ubuntu. If Ubuntu is installed on a single partition on the hard drive and you want to improve performance later, you can shrink the Ubuntu partition and create a swap partition in the free space. This text does not cover how to do that.

Note 6

As you learn to use Ubuntu, know that the help.ubuntu.com website contains a wealth of information about Ubuntu and links to even more help.



Exam Tip

The A+ Core 2 exam expects you to know that a swap partition improves performance and that you can create one during or after the installation. You also need to know about the ext4 and ext3 file systems.

Recall from the module “[The Complex World of IT Professionals](#)” that each OS has file systems it can support. Linux file systems include:

- **ext4.** The current Linux file system is ext4 (fourth extended file system). This is why you selected ext4 when you installed Ubuntu Desktop and Server.
- **ext3.** The ext3 file system was the first to support journaling, which is a technique that tracks and stores changes to the hard drive and helps prevent file system corruption.
- **FAT32 and NTFS.** The FAT32 and NTFS file systems are supported for compatibility with Windows and macOS. Windows can use either FAT32 or NTFS, and macOS can use FAT32. Ubuntu should not be installed on a FAT32 or NTFS volume.

Note 7

On a local network or in a dual boot with Windows and Linux, you might want to access files in either volume from either OS. Know that Linux can access the NTFS file system on the Windows volume, but Windows cannot access the ext4 file system on the Linux volume. You can, however, install third-party software, such as Paragon ExtFS for Windows (paragon-software.com), to access the ext4 volume.

As you read along and learn about Linux commands, you can use your Ubuntu VMs to practice these commands.

Directory Structures

[Table 21-2](#) lists some important directories that are created in the root during a typical Linux installation. (Some distributions of Linux modify the directory structure.) Not all directories in the root are listed in the table. You will need to be familiar with the directories list in [Table 21-2](#) and understand their uses.

Table 21-2

Important Directories in a Typical Linux Root Directory

Directory	Description
/bin	Contains programs and commands necessary to boot the system and perform other system tasks not reserved for the administrator, such as shutdown and reboot.
/boot	Consists of components needed for the boot process, such as boot loaders.
/dev	Holds device names, which consist of the type of device and a number identifying the device; actual device drivers are located in the <code>/lib/modules /<i>[kernel version]</i>/</code> directory.

Directory	Description
/etc	Contains system configuration data, including configuration files and settings and their subdirectories; these files are used for tasks such as configuring a user account, changing system settings, and configuring a domain name resolution service. This directory should be backed up on a regular basis.
/home	Contains user data; every user of the system has a directory in the /home directory, such as /home/evan or /home/robyn, and when a user logs in, that directory becomes the current working directory. This directory should be backed up on a regular basis to protect user-saved documents and files.
/lib	Stores common libraries used by applications so more than one application can use the same library at one time; an example is the library of C programming code, without which only the kernel of the Linux system could run.
/lost+found	Stores data that is lost when files are truncated or when an attempt to fix system errors is unsuccessful.
/opt	Contains installations of third-party applications such as web browsers that do not come with the Linux OS distribution; this directory should be included in your regular backups.
/root	Serves as the home directory for the root user and contains only files specific to the root user; don't confuse this directory with the root, which contains all the directories listed in this table. This directory should also be backed up to ensure a copy of the root user account is saved.
/sbin	Stores commands required for system administration, with root-level privileges required.
/tmp	Stores temporary files, such as the ones that applications use during installation and operation.
/usr	Contains executable programs, libraries, and shared resources that are not critical to the Ubuntu system; this directory and two subdirectories should be backed up. Those subdirectories (/usr/local/bin and /usr/local/sbin) may include applications that have been installed and are not part of the /usr parent directory.
/var	Holds variable data such as logs, email, news, print spools, and administrative files; this directory should also be backed up on a regular basis.



Applying Concepts

Exploring Directories and Files

- **Est. Time:** 15 minutes
- **Core 2 Objective:** 1.11

Let's learn a few Linux commands that we can use to explore directories and files. As shown in [Figure 21-14](#), enter these commands:

1. **1**
Open a new terminal window, or run the **clear** command to clear an existing window of all its clutter.
2. **2**

Enter the `pwd` command (print working directory) to display the full path to the current directory, which is `/home/evan` in the figure.

3. **3**
Run the `cd ..` command to move up one directory to `/home`. (Note the space after the `d`.)
4. **4**
Run the `ls` command to display the list of files and subdirectories in the `/home` directory. Notice in the figure that the one subdirectory in the `/home` directory is `evan`.
5. **5**
Enter the `ls -l` command to display the results using the long format. (Note the space after the `s`.) As you can see in the figure, the results are as follows:

```
drwxr-xr-x 3 evan evan 4096 Sep 18 02:20 evan
```

Figure 21-14

Directories in the root

```
evan@spock:~$ pwd
/home/evan
evan@spock:~$ cd ..
evan@spock:/home$ ls
evan
evan@spock:/home$ ls -l
total 4
drwxr-xr-x 3 evan evan 4096 Sep 18 02:20 evan
evan@spock:/home$ cd ..
evan@spock:/$ ls -l
total 3063880
lrwxrwxrwx 1 root root      7 Aug 24 08:41 bin -> usr/bin
drwxr-xr-x 3 root root    4096 Sep 18 02:21 boot
drwxr-xr-x 2 root root    4096 Sep 18 02:00 cdrom
drwxr-xr-x 19 root root   4100 Sep 18 02:07 dev
drwxr-xr-x 97 root root    4096 Sep 18 02:20 etc
drwxr-xr-x 3 root root    4096 Sep 18 02:07 home
lrwxrwxrwx 1 root root      7 Aug 24 08:41 lib -> usr/lib
lrwxrwxrwx 1 root root      9 Aug 24 08:41 lib32 -> usr/lib32
lrwxrwxrwx 1 root root      9 Aug 24 08:41 lib64 -> usr/lib64
lrwxrwxrwx 1 root root     10 Aug 24 08:41 libx32 -> usr/libx32
drwx----- 2 root root  16384 Sep 18 01:59 lost+found
drwxr-xr-x 2 root root    4096 Aug 24 08:42 media
drwxr-xr-x 2 root root    4096 Aug 24 08:42 mnt
drwxr-xr-x 2 root root    4096 Aug 24 08:42 opt
dr-xr-xr-x 149 root root      0 Sep 18 02:07 proc
drwx----- 4 root root    4096 Sep 18 02:07 root
drwxr-xr-x 26 root root     800 Sep 18 02:17 run
lrwxrwxrwx 1 root root      8 Aug 24 08:41 sbin -> usr/sbin
drwxr-xr-x 6 root root    4096 Aug 24 08:48 snap
drwxr-xr-x 2 root root    4096 Aug 24 08:42 srv
-rw----- 1 root root 3137339392 Sep 18 02:02 swap.img
dr-xr-xr-x 13 root root      0 Sep 18 02:06 sys
drwxrwxrwt 11 root root    4096 Sep 18 02:41 tmp
drwxr-xr-x 15 root root    4096 Aug 24 08:46 usr
drwxr-xr-x 13 root root    4096 Aug 24 08:47 var
evan@spock:/$ _
```

Annotations in the figure:

- Print working directory (points to `pwd` command)
- Result of `ls` command (points to `ls` command)
- Result of `ls -l` command (points to the output of `ls -l` in the `/home` directory)
- Directories in root (points to the list of directories in the root directory)

Here is an explanation of the types of information in the list:

- **Attributes.** The first 10 characters (drwxr-xr-x) define the file or directory attributes. The first character identifies the type of item: A “d” is a directory, “a –” is a regular file, and a “1” indicates the item is a link to another location. The other nine characters (“rwxr-xr-x”) define the read, write, and execute permissions assigned to the file or directory; these permissions are explained in detail later in the module.
- **Links.** The second column lists the number of links the item has, which is three in our example. In Linux, a link is similar to a Windows shortcut to a file or directory.
- **Owners.** The third column lists the user owner, and the fourth column lists the group that owns the file or directory. In [Figure 21-14](#), the owner is evan, and the owner group is also evan.
- **Size, date, and name.** The last columns list the size of the file or directory in bytes, the date the item was last modified, and the name of the file or directory. The name of directory, evan, is listed in blue.

6. **6**

Enter the `cd ..` command again to move up to the main directory in Linux, called the root directory, which is indicated with a forward slash. Run the `ls -l` command to list the files and subdirectories in the root.

Root Account and User Accounts

A Linux system administrator is responsible for installing updates to the OS (called patches), managing backups, installing software and hardware, setting up user accounts, resetting passwords, and generally supporting the OS and users. To accomplish this, they require root privileges or access to all the functions of the OS. Two ways Linux allows for root privileges are as follows:

- **Superuser.** The principal user account is called the **root account**. Notice in [Figure 21-14](#) that all the directories and files in the root directory belong to the root account. When logged in to the root account, the user is called the **superuser**. Because the root account is so powerful, Ubuntu disables login to this account by default. The root account is similar to the Windows Administrator account.
- **Regular user account with root privileges.** Any user account can be assigned root privileges. If the user has root privileges, they can execute any command that requires root access by adding `sudo` to the beginning of the command line. An account with root privileges is similar to a Windows account that has been assigned administrative rights.

The command to switch users is `su`. To switch to the root account, use the command `sudo su root`, as shown in [Figure 21-15](#).

Figure 21-15

The user account, host name, and current directory appear in the shell prompt, along with a # or \$ to indicate the root account or other account

```
evan@spock:/$ sudo su root
[sudo] password for evan:
root@spock:/# su evan
evan@spock:/$
```

Notice in the figure that the shell prompt changes to root@spock:/#. The Linux command prompt for the root user is different from the command prompt for regular users. The root command prompt is #, and other users have the \$ command prompt. To switch back to the evan account, use the command su evan. As a general practice, never log in to Linux as root unless you have no other option; you can do a lot of damage as root.

Linux Commands

[Table 21-3](#) describes some basic Linux commands, together with simple examples of how some are used. As you read along, be aware that all commands entered in Linux are case sensitive, meaning that uppercase and lowercase matter. You can practice these commands in a Ubuntu VM, or you can practice the commands by booting a computer from the Live Ubuntu bootable USB drive you created in the Core 1 module “[Hard Drives and Other Storage Devices](#).”

Table 21-3

Some Common Linux Commands

Command	Description
adduser	Add a user to a system: adduser <username>
apt-get	Install and remove applications and other programs (called packages) in Linux. When you first install Linux, it installs with only a bare-bones set of commands and utilities, and it includes a library of packages that you can install as needed. For example, to install the SSH (Secure Shell) package so you can remotely connect to your Linux server: sudo apt-get install ssh The apt-get command requires root access, which means you must precede the command with sudo.
cat	View the contents of a file. Many Linux commands can use the redirection symbol > to redirect the output of the command. For example, use the redirection symbol with the cat command to copy a file:

Command	Description
	<pre>cat /etc/shells > newfile</pre> <p>The content of the shells file is written to newfile.</p>
cd	<p>Change the directory.</p> <p>To change the directory to /etc:</p> <pre>cd /etc</pre> <p>To move up one level in the directory tree:</p> <pre>cd ..</pre> <p>To go to the root:</p> <pre>cd /</pre>
chmod	Change modes (or permissions) for a file or directory. You'll see several examples of this command later in the module.
chown	<p>Change the owner of a file or directory.</p> <p>To change the owner of /mystuff to root:</p> <pre>chown root /mystuff</pre>
clear	Clear the screen. This command is useful when the screen has become cluttered with commands and data that you no longer need to view.
cp	<p>Copy a file:</p> <pre>cp <source> <destination></pre>
df	<p>Show the amount of disk free space available in Linux and to understand the file systems that are attached, or mounted, to the system. Use the df (disk file system) command with no parameters to report file system device names, free space, and mount points:</p> <pre>df</pre>
dig	<p>Query for DNS information. This is similar to the Windows command nslookup. To lookup the DNS information for linux.org:</p> <pre>dig linux.org</pre>
echo	<p>Display information on the screen. You can also save the information to a file. For example, to create a new file that contains text:</p> <pre>echo "Hello World" > myfile</pre>
exit	Log out; the login shell prompt appears, and you can log in again.

Command	Description
find	<p>"Walk" through the file hierarchy to find files and directories. To find a file, the syntax of the command is as follows:</p> <pre>find <directoryname> -name <filename></pre>
grep	<p>Search for a specific pattern in a file or in multiple files. This command is useful when searching through long log files:</p> <pre>grep <pattern> <file></pre>
ifconfig	<p>Troubleshoot problems with TCP/IP network connections. This command can disable and enable network adapters and assign a static IP address to an adapter. For example, to show all configuration information:</p> <pre>ifconfig -a</pre> <p>To enable or disable an adapter, use the up or down parameter. For example, to enable eth0, the first Ethernet interface:</p> <pre>sudo ifconfig eth0 up</pre> <p>To assign a static IP address to the eth0 interface:</p> <pre>ifconfig eth0 192.168.1.90</pre>
ip	<p>Display IP addresses and property information of the various network interfaces on the system. This command can also be used to configure and modify routes, tunnel over IP, and modify the interface statuses. Many Linux distributions have moved to the ip command rather than using ifconfig.</p> <p>For example, to show the ip address information:</p> <pre>ip a</pre> <p><i>or</i></p> <pre>ip addr</pre> <p>To assign the 192.168.1.100 ip address and 255.255.255.0 subnet mask to the eth0 interface:</p> <pre>ip a add 192.168.1.100/255.255.255.0 dev eth0</pre>
iwconfig	<p>Display information about the wireless adapter's configuration or to change the configuration. This command works like ifconfig but applies only to wireless networks. To set the wireless NIC to Ad-Hoc mode so other devices within range can connect directly to it, use this command, where wlan0 identifies the wireless adapter:</p> <pre>iwconfig wlan0 mode Ad-Hoc</pre> <p>To force the NIC to use channel 3:</p> <pre>iwconfig wlan0 channel 3</pre>

Command	Description
kill	<p>Kill a process instead of waiting for it to terminate. Use the ps command to list process IDs. To end a process, use the kill command followed by the PID. For example, to kill the process with a PID of 984:</p> <pre>kill 984</pre> <p>The command sends a signal to the process to end itself in an orderly way. If the process doesn't die peacefully, you can get the kernel involved to forcefully end the process (this is called a forced kill):</p> <pre>kill -kill 984</pre>
ls	<p>Display a list of directories and files (similar to the Windows dir command). For example, to list files in the /etc directory, use the long parameter for a complete listing:</p> <pre>ls -l /etc</pre> <p>To include hidden files in the list:</p> <pre>ls -la /etc</pre> <p>(Note that in Linux, hidden files begin with a period.)</p>
man	<p>Display the online help manual, called man pages. For example, to get information about the echo command:</p> <pre>man echo</pre> <p>The manual program displays information about the command. To exit the manual program, type q.</p>
mkdir	<p>Make a new directory:</p> <pre>mkdir <directory></pre>
mv	<p>Move a file or rename it, if the source and destination are the same directory:</p> <pre>mv <source> <destination></pre> <p>For example, to move myfile from the evan directory to the home directory:</p> <pre>sudo mv /home/evan/myfile /home/myfile</pre> <p>Note that because the /home directory is owned by the root account, the sudo command is required.</p>
nano	<p>Launch a full-screen editor that can be used to edit a file:</p> <pre>nano <filename></pre>
passwd	<p>Change a password. When a user enters the command, they are asked for the old password and then can change it.</p>

Command	Description
	The superuser can change the password for any account and does not need to enter the account's old password, making it possible to reset a forgotten password.
ping	<p>Test network connections by sending a request packet to a host. If a connection is successful, the host will return a response packet. For example:</p> <pre>ping 192.168.1.100</pre> <p>The ping results continue until you manually stop the process. Press Ctrl+C to break out of the process. To specify the number of pings:</p> <pre>ping 192.168.1.100 -c 4</pre>
ps	<p>Display the process table so you can identify process IDs for currently running processes. The command stands for "process status." (Once you know the process ID, you can use the kill command to terminate a process.)</p> <p>To list processes of the current user:</p> <pre>ps</pre> <p>To list processes owned by all users:</p> <pre>ps aux</pre>
pwd	<p>Show the name of the current or present working directory. The command stands for "print working directory."</p> <p>When you first log in to Linux, the directory is <i>/home/username</i>.</p>
rm	<p>Remove or delete the file or files that are specified:</p> <pre>rm <file></pre>
rmdir	<p>Remove or delete an empty directory:</p> <pre>rmdir <directory></pre>
shutdown	<p>Automatically shut down the system.</p> <p>To shut down now:</p> <pre>sudo shutdown now</pre> <p>To warn users and then shut down:</p> <pre>sudo shutdown -h +10 "Everyone log out now. The system will shut down in 10 minutes for maintenance."</pre> <p>To reboot now:</p> <pre>sudo shutdown -r now</pre>

Command	Description
su	<p>Switch to a different user account. The command stands for “substitute user” or “switch user.” When switching to superuser, add sudo to the command.</p> <p>To switch to the root account:</p> <pre>sudo su root</pre> <p>To switch back to the evan account:</p> <pre>su evan</pre>
sudo	<p>Run a command as the superuser. The command stands for “substitute user to do the command,” and it is pronounced “sue-doe” or “sue-doo.” When logged in as a normal user with an account that has the right to use root commands, you can start a command with sudo to run the command as the superuser. A user password may be required. For example:</p> <pre>sudo shutdown now</pre>
top	<p>Display the Linux processes. This command shows a dynamic, real-time view of the running processes and threads currently being managed by the Linux kernel.</p>
yum	<p>Get, install, delete, and manage software packages in RPM (Red Hat Package Manager). The yum (Yellowdog Update Manager) command is the primary tool for performing these functions in RPM. This command is similar to the apt-get update command used in this module to update distributions such as Ubuntu.</p> <p>For example, to download updates for all installed software:</p> <pre>sudo yum update</pre>



Exam Tip

The A+ Core 2 exam expects you to be familiar with these Linux commands: apt-get, cat, chmod, chown, cp, df, dig, find, grep, ip, ls, man, mv, nano, ps, pwd, rm, top, yum.

Here are a few tips when using commands at a shell prompt:

- **Retrieve previous commands.** Press the up arrow key to retrieve previously entered commands and then edit a command that appears.
- **Use wildcard characters.** Linux, similar to Windows, can use the * and ? wildcard characters in command lines. For example, use the ls *.??? command to list all files with a file extension of three characters. In addition, Linux provides a third wildcard: Brackets can give a choice of characters. For example, use the ls *. [abc]* command to list all files whose file extension begins with a, b, or c.
- **Redirect output.** Normally, output from a command displays on the screen. To redirect that output to a file, use the redirection symbol >.

For example, to redirect the output of the `ifconfig` command to myfile, use the `ifconfig > myfile` command.

- **Page the output.** Append `|more` to the end of a command line to display the results of the command on the screen one page at a time. For example, to page the `ls` command: `ls -l |more`.
- **Use `Ctrl+C`.** To break out of a command or process, press **Ctrl+C**. Use it to recover after entering a wrong command or to stop a command that requires a manual halt.

The Nano Editor

The nano editor is a text editor that allows you to easily make changes to text-based files. In this section, you learn how to create a text file in the nano editor, edit text, and save your changes.

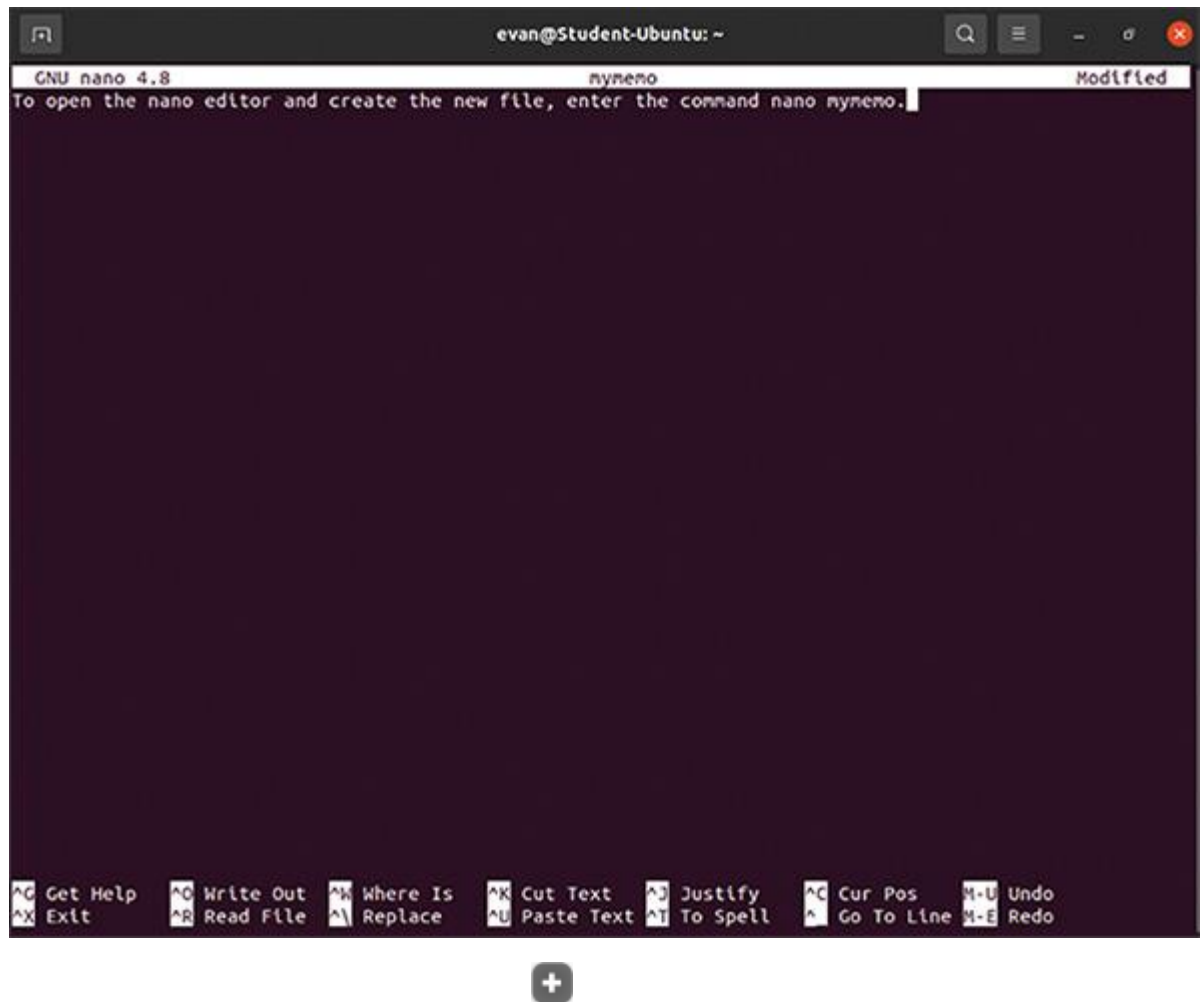
Let's create and work with a file called `mymemo`:

1. You can save a file to your home directory because you own that directory. If you are not already in your home directory, run the `cd` command to go there. For example:


```
cd /home/evan
```
2. To open the nano editor and create the new file, enter the command `nano mymemo`. The nano editor screen appears, and the file name is shown at the top center of the screen.
3. When you first open the nano editor, you can immediately edit the file by typing in the window. Type the first sentence of step 2 as the text for your memo (see [Figure 21-16](#)).

Figure 21-16

The nano text editor with commands displayed at bottom of screen



4. At the bottom of the screen are several commands. The carrot (^) symbol is used to represent the Control (Ctrl) key on your keyboard. To see all the commands available in nano, press **Ctrl+G**.
5. To save your changes, press **Ctrl+O**, and then press **Enter** to save the changes to the listed file name. To exit, press **Ctrl+X**. Alternatively, you can just use the Exit command, and nano will ask if you want to save any changes made to the file.

Applying Concepts

Installing FTP Server in Ubuntu

- **Est. Time:** 15 minutes
- **Core 2 Objective:** [1.11](#)

In the following steps, you learn to use several Linux commands to install and configure software and examine a log file. Follow these steps to set up an FTP server in Ubuntu:

1. **1**
Log in to Ubuntu Server with your user name and password.
2. **2**

To create a short file to test the FTP server, you can use the echo command with redirection. Create mymemo2 in your `/home/username` directory by entering this command:

```
echo "my typing" > /home/username/mymemo2
```

3. **3**

To install the FTP program named vsftpd, enter this command:

```
sudo apt-get install vsftpd
```

4. **4**

Respond to the prompts, and then wait for the package to install.

5. **5**

Now you need to configure the FTP program by editing the `/etc/vsftpd.conf` text file. Before you edit the file, go to the `/etc` directory and make a backup copy of the file just in case you need it later. The sudo command is needed because files in the `/etc` directory belong to root:

```
cd /etc
```

```
sudo cp vsftpd.conf vsftpd.backup
```

6. **6**

Use the nano editor to edit the FTP configuration file:

```
sudo nano vsftpd.conf
```

7. **7**

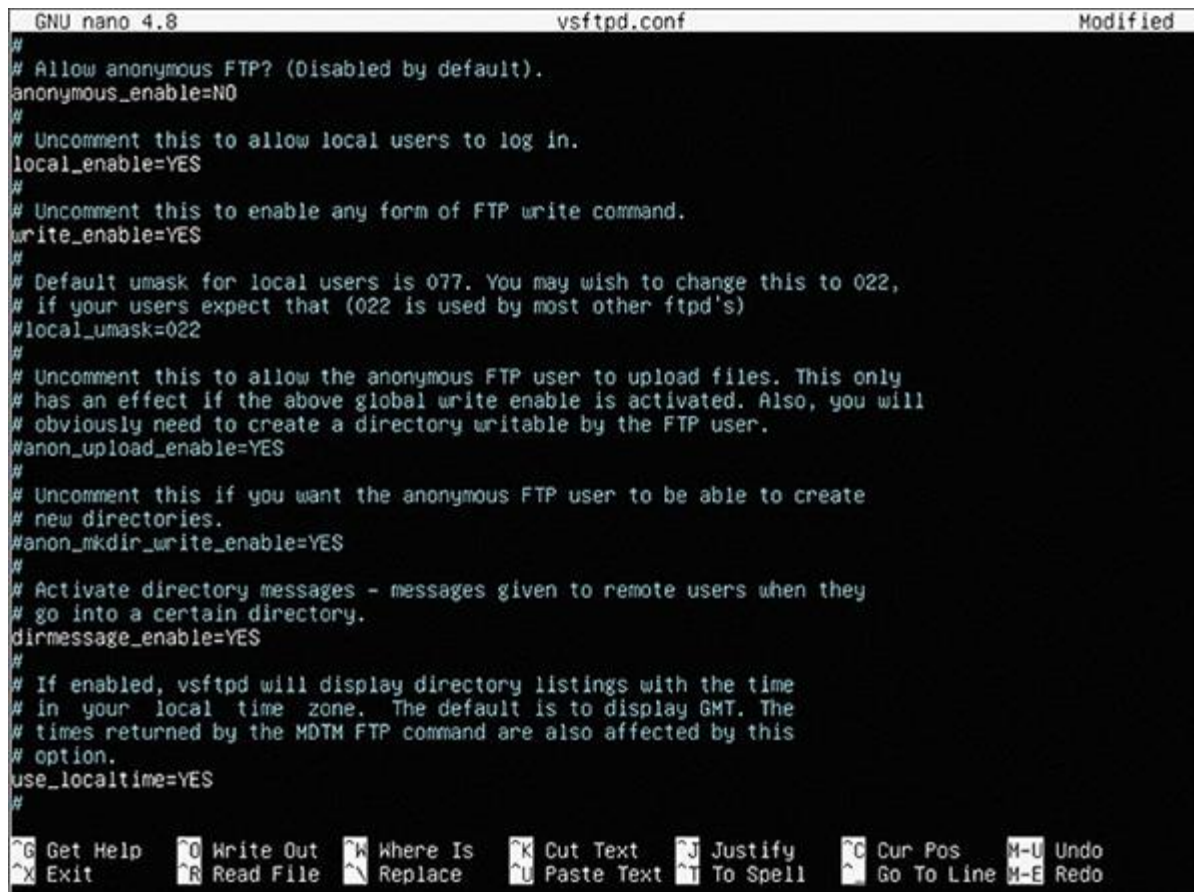
Verify and/or change three lines in the file to create the following settings:

- Disable anonymous logins: **anonymous_enable=NO**
- If necessary, remove the # to uncomment the line, and allow local users to log in: **local_enable=YES**
- If necessary, remove the # to uncomment the line, and allow users to write to a directory: **write_enable=YES**

Part of the file, including these three lines, is shown in [Figure 21-17](#).

Figure 21-17

Part of the vsftpd.conf text file



```
GNU nano 4.8          vsftpd.conf          Modified
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
#
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES
#
# If enabled, vsftpd will display directory listings with the time
# in your local time zone. The default is to display GMT. The
# times returned by the MDTM FTP command are also affected by this
# option.
use_localtime=YES
#
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text   ^J Justify   ^C Cur Pos   M-U Undo
^X Exit      ^R Read File  ^_ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line M-E Redo
```

8. **8** Exit the nano editor, saving your changes. Restart the FTP service by running this command:

```
service vsftpd restart
```

9. **9** To find out the IP address of the server, type **ifconfig**. If net-tools are not installed, you will need to install them.

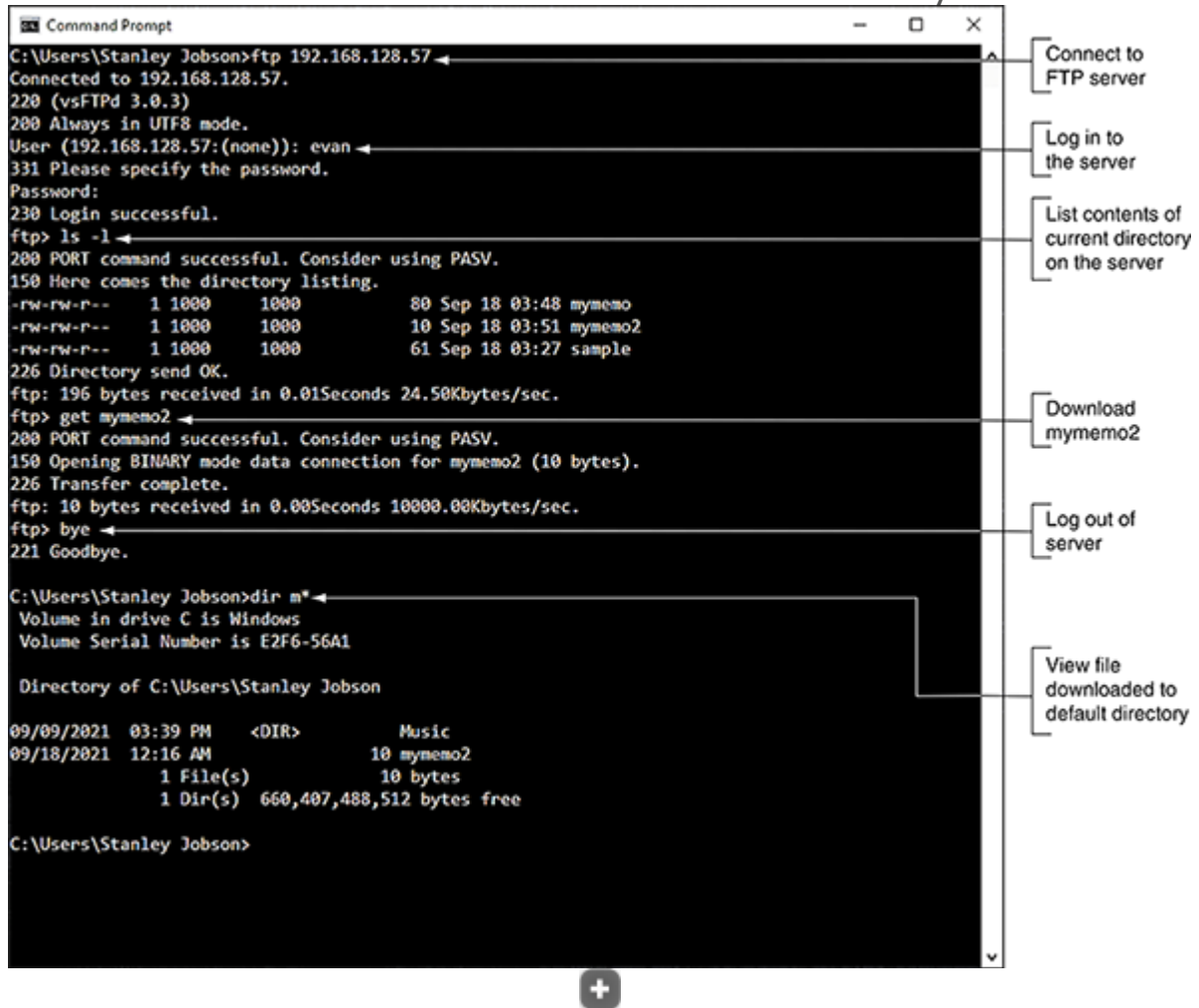
Note 8

If an error appears stating “Command ‘ifconfig’ not found, but can be installed with: apt install net-tools,” you can run the command **apt install net-tools** to install the networking tools.

1. **10** On your host Windows computer, open a command prompt window, and go to a directory on your Windows computer where you have a file stored. To test your FTP server, open an FTP session using the IP address of the server—for example: ftp 192.168.128.57 (your IP address may be different). Then enter your user name and password. The ftp> prompt appears. See [Figure 21-18](#).

Figure 21-18

Use FTP to transfer files between a Windows and Ubuntu system



```
C:\Users\Stanley Jobson>ftp 192.168.128.57
Connected to 192.168.128.57.
220 (vsFTPd 3.0.3)
200 Always in UTF8 mode.
User (192.168.128.57:(none)): evan
331 Please specify the password.
Password:
230 Login successful.
ftp> ls -l
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-rw-r-- 1 1000 1000 80 Sep 18 03:48 mymemo
-rw-rw-r-- 1 1000 1000 10 Sep 18 03:51 mymemo2
-rw-rw-r-- 1 1000 1000 61 Sep 18 03:27 sample
226 Directory send OK.
ftp: 196 bytes received in 0.01Seconds 24.50Kbytes/sec.
ftp> get mymemo2
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for mymemo2 (10 bytes).
226 Transfer complete.
ftp: 10 bytes received in 0.00Seconds 10000.00Kbytes/sec.
ftp> bye
221 Goodbye.

C:\Users\Stanley Jobson>dir m*
Volume in drive C is Windows
Volume Serial Number is E2F6-56A1

Directory of C:\Users\Stanley Jobson

09/09/2021 03:39 PM <DIR> Music
09/18/2021 12:16 AM 10 mymemo2
1 File(s) 10 bytes
1 Dir(s) 660,407,488,512 bytes free

C:\Users\Stanley Jobson>
```

Annotations on the right side of the screenshot:

- Connect to FTP server
- Log in to the server
- List contents of current directory on the server
- Download mymemo2
- Log out of server
- View file downloaded to default directory

2. **11**
Next, run the `ls -l` command to see a list of directories and files. You should see the file `mymemo2` that you created in your `/home/username` directory earlier.
3. **12**
If you want to transfer files with FTP commands, use the `get` and `put` commands. To download the `mymemo2` file, run the command `get mymemo2`. To transfer a file from your Windows computer to your Ubuntu server, run the `put` command.
4. **13**
Type `bye` to disconnect from the FTP server. At the Windows command prompt, run the `dir m*` command, as shown in [Figure 21-18](#), to verify that the file was received on the Windows computer.
5. **14**
Return to Ubuntu Server and examine the FTP log file, `/var/log/vsftpd.log`. Because the file is short, you can run the `cat` command to display the entire log. The `sudo` command is required because `/var` files belong to root:

```
sudo cat /var/log/vsftpd.log
```


6. 15

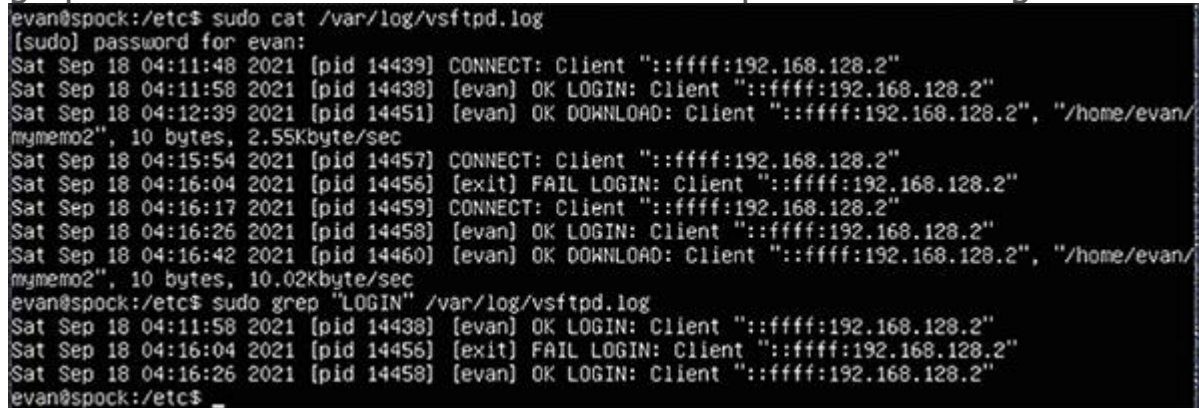
After much activity, log files can get quite long. The `grep` command can help you find a specific action, user, IP address, file name, or directory name. For example, to display lines in the log file that contain the text “LOGIN,” enter this `grep` command:

```
sudo grep "LOGIN" /var/log/vsftpd.log
```

The results of the `cat` and `grep` commands are shown in [Figure 21-19](#).

Figure 21-19

The `grep` command can be used to search for specific text in log files



```
evan@spock:/etc$ sudo cat /var/log/vsftpd.log
[sudo] password for evan:
Sat Sep 18 04:11:48 2021 [pid 14439] CONNECT: Client "::ffff:192.168.128.2"
Sat Sep 18 04:11:58 2021 [pid 14438] [evan] OK LOGIN: Client "::ffff:192.168.128.2"
Sat Sep 18 04:12:39 2021 [pid 14451] [evan] OK DOWNLOAD: Client "::ffff:192.168.128.2", "/home/evan/mymemo2", 10 bytes, 2.55Kbyte/sec
Sat Sep 18 04:15:54 2021 [pid 14457] CONNECT: Client "::ffff:192.168.128.2"
Sat Sep 18 04:16:04 2021 [pid 14456] [exit] FAIL LOGIN: Client "::ffff:192.168.128.2"
Sat Sep 18 04:16:17 2021 [pid 14459] CONNECT: Client "::ffff:192.168.128.2"
Sat Sep 18 04:16:26 2021 [pid 14458] [evan] OK LOGIN: Client "::ffff:192.168.128.2"
Sat Sep 18 04:16:42 2021 [pid 14460] [evan] OK DOWNLOAD: Client "::ffff:192.168.128.2", "/home/evan/mymemo2", 10 bytes, 10.02Kbyte/sec
evan@spock:/etc$ sudo grep "LOGIN" /var/log/vsftpd.log
Sat Sep 18 04:11:58 2021 [pid 14438] [evan] OK LOGIN: Client "::ffff:192.168.128.2"
Sat Sep 18 04:16:04 2021 [pid 14456] [exit] FAIL LOGIN: Client "::ffff:192.168.128.2"
Sat Sep 18 04:16:26 2021 [pid 14458] [evan] OK LOGIN: Client "::ffff:192.168.128.2"
evan@spock:/etc$
```

Update Linux from the Shell Prompt

In general, Linux updates don’t come as often as Windows or macOS updates. The creator of your Linux distribution publishes updates to packages in the current release of a distribution and publishes new releases of a distribution. When you first log in to the system, Linux reports the package updates that are available (refer back to [Figure 21-13](#)).

Run these commands to update the packages previously installed in your system:

1. To refresh the list of all available updates:

```
sudo apt-get update
```

2. To update only the installed packages:

```
sudo apt-get upgrade
```

A new release of a distribution contains all updates since the last release. As a Linux administrator, you need to stay aware of the latest release of the distribution you are using and decide when or if it’s appropriate to upgrade to that release. Before you upgrade to a new release, be sure you have backups of your data and a disk image (called a **clone**) of the entire Linux partition.

Here's how to upgrade to a new release for Ubuntu Server:

1. Follow the previous steps to update all packages installed in the system.
2. To make sure the latest update manager program is installed, enter the command:

```
sudo apt-get install update-manager-core
```

3. To install the latest release of Ubuntu Server, enter the command:

```
sudo do-release-upgrade
```

If a new release is available, the last command reports it, and you can follow directions to install it.

Note 9

Ubuntu Server does not have an easy-to-use backup service. Normally, the system administrator installs third-party software, such as Bacula, to perform backups. Another option is to write your own **shell scripts**, which are similar to batch files, that include tar commands to create an archive of many files and copy it to other media, such as a USB drive or network storage. Files recommended for your Linux backup processes are noted in the list of directories in [Table 21-2](#), presented earlier in the module.

Assign Permissions and Ownership of Files or Directories

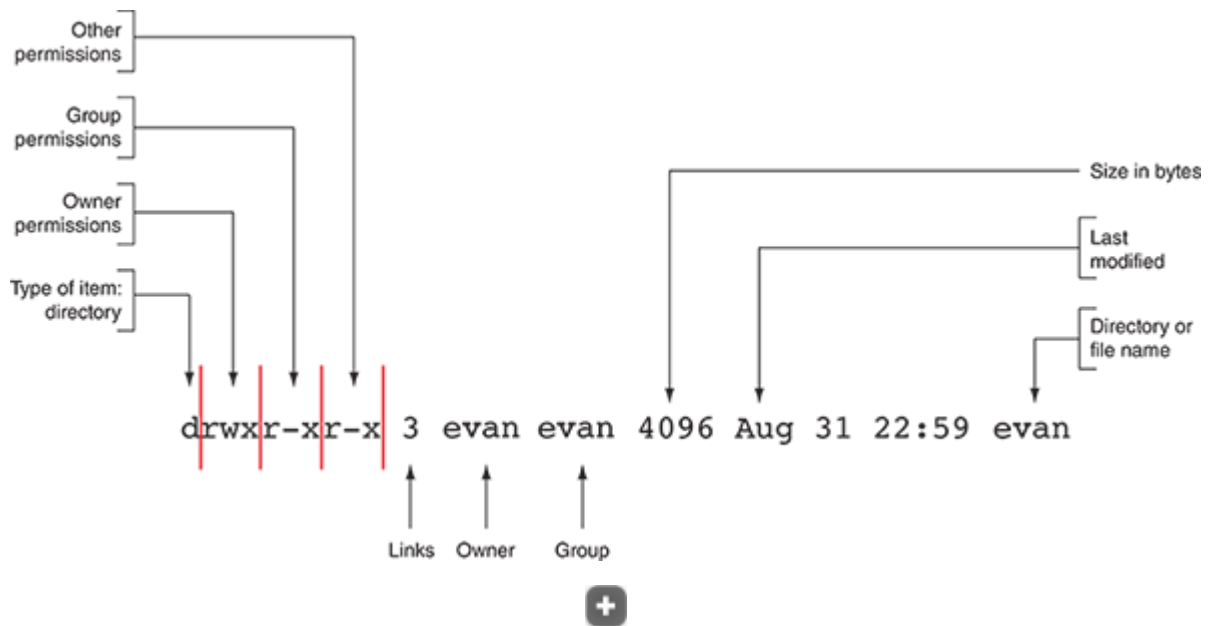
A file or directory can have read, write, and/or execute permissions assigned to it. Permissions can be assigned to

1. the owner,
2. other users in the same group as the owner, and
3. all users.

The `chmod` command is used to manage permissions for files and directories. To see current permissions, examine the 10 characters in the left column that display when you use the `ls -l` command. For example, suppose the output for the `ls -l` command on the `/home/evan` directory is that shown in [Figure 21-20](#).

Figure 21-20

Information about the `evan` directory displayed using the `ls -l` command



Here is the explanation of these characters:

- The first character identifies the type of item (`d` is a directory; `-` is a regular file).
- Characters 2–4 show the permissions assigned to the user or owner (for example, `rwx` means the user has read, write, and execute permissions).
- Characters 5–7 show the permissions assigned to the group (for example, `r-x` means the group has read and execute permissions, but not write permission).
- Characters 8–10 show the permissions for others (for example, `r-x` means other users have read and execute permissions, but not write permission. By contrast, `---` would mean others don't have read, write, or execute permission).

The `chmod` command changes these permissions. To change permissions on files and directories, you can use either the symbolic (characters) format or the absolute (octal numbers 0 through 7) format. Using the example in [Figure 21-20](#), if the user `evan` wants to give read, write, and execute permissions to everyone (group and other), he can use this symbolic formatted command:

```
chmod g=rwx,o=rwx /home/evan
```

The `g` assigns permission to the group, and the `o` assigns permissions to others. (The `u` can assign permissions to the owner.) For a folder, you must move out of the folder before you change its permissions. Also, the command has no space before or after a comma.

Let's look at an example of when you might need to change the ownership of a directory or file. Suppose you are setting up an FTP server so users who have accounts on the server can use FTP to upload files to their home

directories. [Table 21-4](#) lists commands that might be useful to set up the user directories.

Table 21-4

Commands to Set Up User Directories with Appropriate Write Permissions

Command	Description
<code>sudo adduser hector</code>	Creates the user account and its home directory
<code>sudo mkdir /home/hector/files</code>	Creates a subdirectory named files for hector to store their files when using FTP; the sudo command is required to create the directory in another user's account, and this new directory belongs to root
<code>sudo chown hector:hector /home/hector/files</code>	Changes the owner of the files directory. The user:group component of the command written as hector:hector says that the user owner is hector, and also the group owner is the hector group. Note there is no space before or after the colon.
<code>sudo chmod u+w, g-w,o-w /home/hector/files</code>	Gives write permissions to the user, and removes write permissions from the group and other

You can also use an octal format, also known as the absolute method, to change the permissions. The octal format is used to quickly use octal numbers 0 through 7 to specify the permission values for the user, group, and others. In the example, `chmod 754 myfile`, the 754 is broken into a 7 for the user, a 5 for the group, and a 4 for others. The octal values for the read, write, and execute permissions are as follows:

- 4 stands for “read”
- 2 stands for “write”
- 1 stands for “execute”
- 0 stands for “no permission”

When looking back at the 754 in our example, the user is given full permissions of read, write, and execute. The group is given the read and execute only, and others are given read only permission (4).

21-1c Telnet and SSH for Remote Access

Core 2 Objective

- 4.9

Given a scenario, use remote access technologies.

Recall that Windows uses Remote Desktop and macOS uses Screen Sharing to remotely access a computer with screen and file sharing. Windows also makes use of SSH for secure remote access, as you learned in the module “[Network Security and Troubleshooting](#).” In Linux, the primary utilities for

remote access at a shell prompt are **Telnet** and SSH (Secure Shell). Telnet does not encrypt transmissions, but SSH encrypts all transmissions. Therefore, SSH is more secure than Telnet and is the most common method to remotely access a Linux system. Use the commands in [Table 21-5](#) to install Telnet and SSH in Linux.

Table 21-5

Install and Run Telnet and SSH in Linux

Command	Description
<code>sudo apt-add-repository universe</code>	Add the universe repository to the list of places Ubuntu can find apps. This is an official repository of apps, but it is not supported by Ubuntu. (Note that when you attempt to add the repository and it is already available, a message appears, saying the component is already enabled.)
<code>sudo apt-get update</code>	Download and update all apps available to Ubuntu, including the ones in the universe repository.
<code>sudo apt-get install openssh-server</code>	Install and run the SSH server in Linux.
<code>sudo apt install telnet</code>	Install and run the Telnet server.

In Windows, SSH is enabled by default, and Telnet Client can be turned on using the Windows Features window, which is available in the Programs and Features window. Here are the steps to remotely access a Linux system from a Windows or Linux computer:

1. For Telnet using Windows, turn on **Telnet Client** in the **Windows Features** window.
2. For Telnet, at a Windows command prompt or Linux shell prompt, run the following command to open a Telnet session, substituting the IP address for the Linux system you want to remote in to:

```
telnet 192.168.1.160
```

3. The Linux login prompt appears. Enter your user name and password. To close the session, run the **logout** command.
4. To use SSH to remote in, enter the following command, substituting your user name and IP address for the remote Linux system:

```
ssh evan@192.168.1.160
```

5. Enter your password to log in to Linux. To close the session, run the **logout** command.

21-1d Samba File Servers

Core 2 Objective

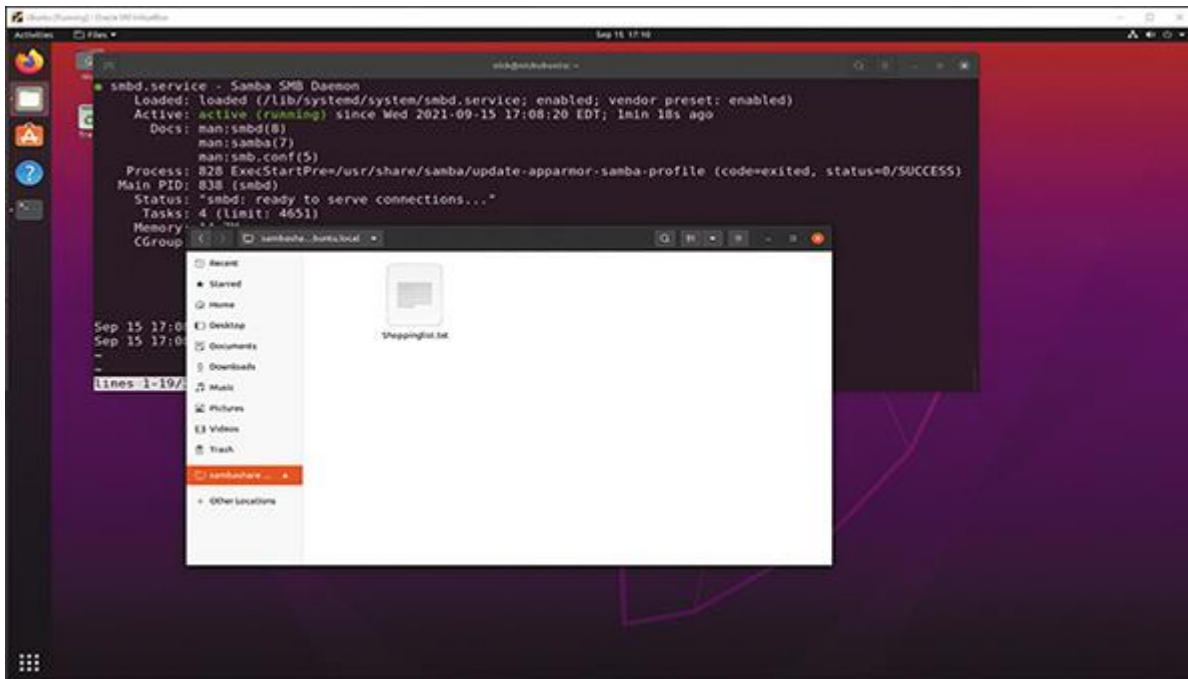
- 1.11

Identify common features and tools of the Linux client/desktop OS.

A Linux **Samba** file server allows for file sharing across different operating systems within a network. The name Samba comes from the Windows protocol server message block (SMB). Whereas SMB is used solely in a Windows environment for file and printer sharing, Samba provides the ability for Linux-hosted files and printers to be accessed across the Windows network. [Figure 21-21](#) shows the Samba service running on Ubuntu, and in the window open toward the bottom of [Figure 21-21](#), you can see the Shoppinglist.txt file that is located on the Ubuntu hard drive.

Figure 21-21

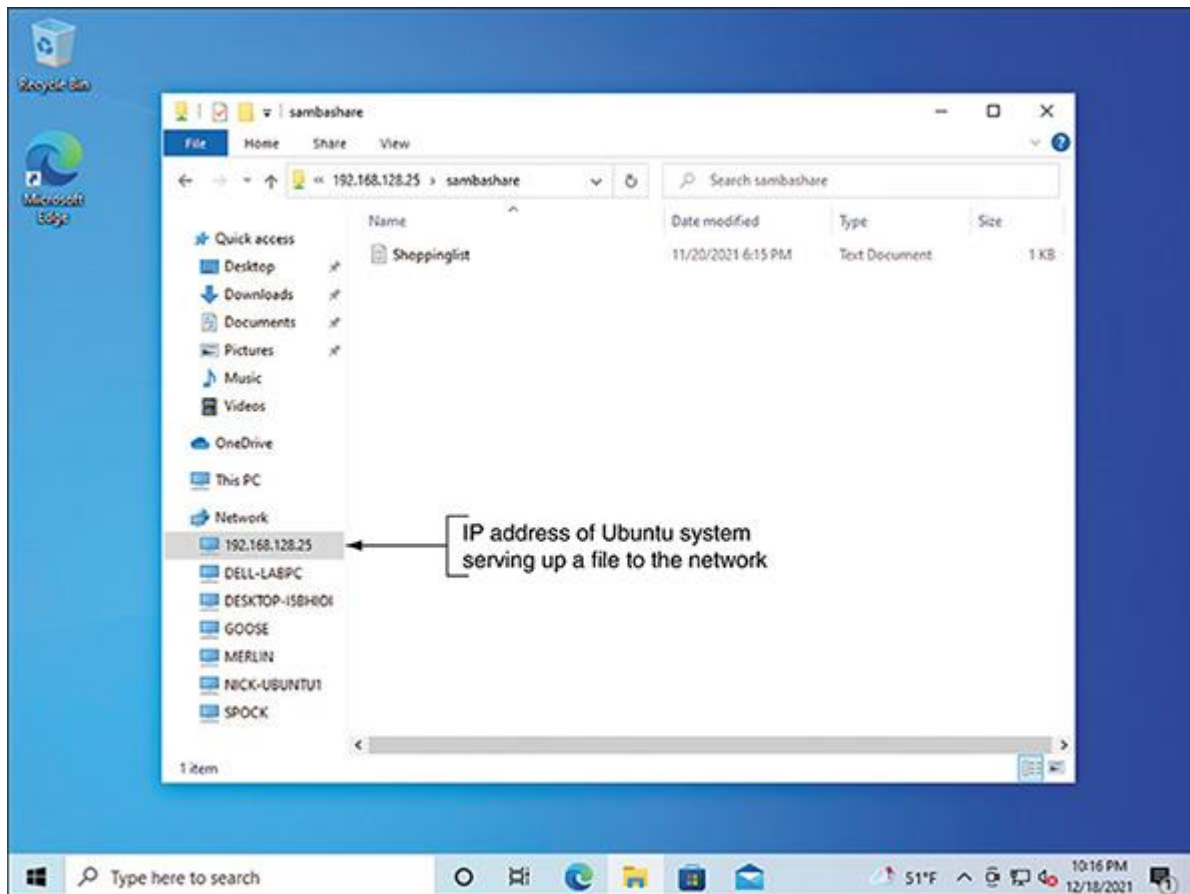
Samba service running in background and Shoppinglist text file in the sambashare directory



In [Figure 21-22](#), you can see that same Shoppinglist file being accessed from a Windows operating system. This is because the file was shared by the Ubuntu system using the Samba service.

Figure 21-22

Shoppinglist text file shown in Windows



You learn how to install and use Samba later in this module with [Real Problem 21-3](#). Now let's turn our attention to the last topic of the module, scripting.

21-2 Scripting Software and Techniques

Core 2 Objective

- 4.8

Identify the basics of scripting.

In this text, you've learned to use commands in a Windows command prompt window and Linux commands within the Linux terminal. When a technician finds themselves repetitively entering the same group of commands, they might decide to store them in a text file and execute them as a batch. The text file containing such a list of commands is called a **script**; using scripts can save time and ensures consistency (fewer errors). In this part of the module, you learn about the various script file types and then explore the basics of reading and writing scripts, which will help get you started using scripts written by others or writing your own scripts.

21-2a Script File Types

In the module “[Maintaining Windows](#),” you learned to create a batch file that contains Windows commands. This and other script file types are listed in [Table 21-6](#) with a description of the software that can read and interpret each command in a script file and execute these commands in a **run-time environment**.

Table 21-6

Types of Script Files and Scripting Software

Script File Extension	Description
.bat	A batch file contains a list of Windows commands that can be executed in a command prompt window.
.ps1	A PowerShell script contains PowerShell commands, also known as cmdlets , that are executed in Windows PowerShell. The script is written using dynamic type checking , which means each cmdlet is checked by the PowerShell interpreter as it is typed to verify that the command can be executed as it is added to the script file. Many scripting and programming applications support dynamic type checking.
.vbs	A .vbs script is written with VBScript , which is modeled after the more complex Visual Basic, a full-fledged programming language.
.sh	A UNIX or Linux script, also called a shell script, contains Linux commands and is executed in a UNIX or Linux shell.
.py	A Python script is a group of Python commands interpreted by Python. Python can also compile the commands into an executable program.
.js	A .js script written in JavaScript is a text file that contains commands normally used with webpages. These scripts can be embedded in an HTML file, which is downloaded from a web server to a browser and used to build an interactive webpage in the browser. To ensure this works, JavaScript needs to be activated or turned on from the browser settings.



Note 10

With the release of Windows 11, Microsoft now includes the Windows Terminal application, in addition to PowerShell version 7.2. You can use Windows Terminal to run commands from Command Prompt and cmdlets from PowerShell. Scripts should still be created in the PowerShell ISE.

Scripts are simpler to write and use than programs. Differences between a script and a program include the following:

- **A script is interpreted.** A script is read, interpreted, and executed command by command directly from the script file by software called an interpreter. For example, a technician types the commands into a script file, and the commands are verified to work by PowerShell or

VBScript. When a technician executes the script, PowerShell or VBScript reads, interprets, and executes each command in the file.

- **A program is compiled.** A program is first written using a programming language such as Visual Basic or Python, which interprets the code to verify that it can be executed. Then the coding file is compiled by the Visual Basic or Python compiler into a binary executable file that has an .exe file extension. The executable file can then be run or executed by an operating system.

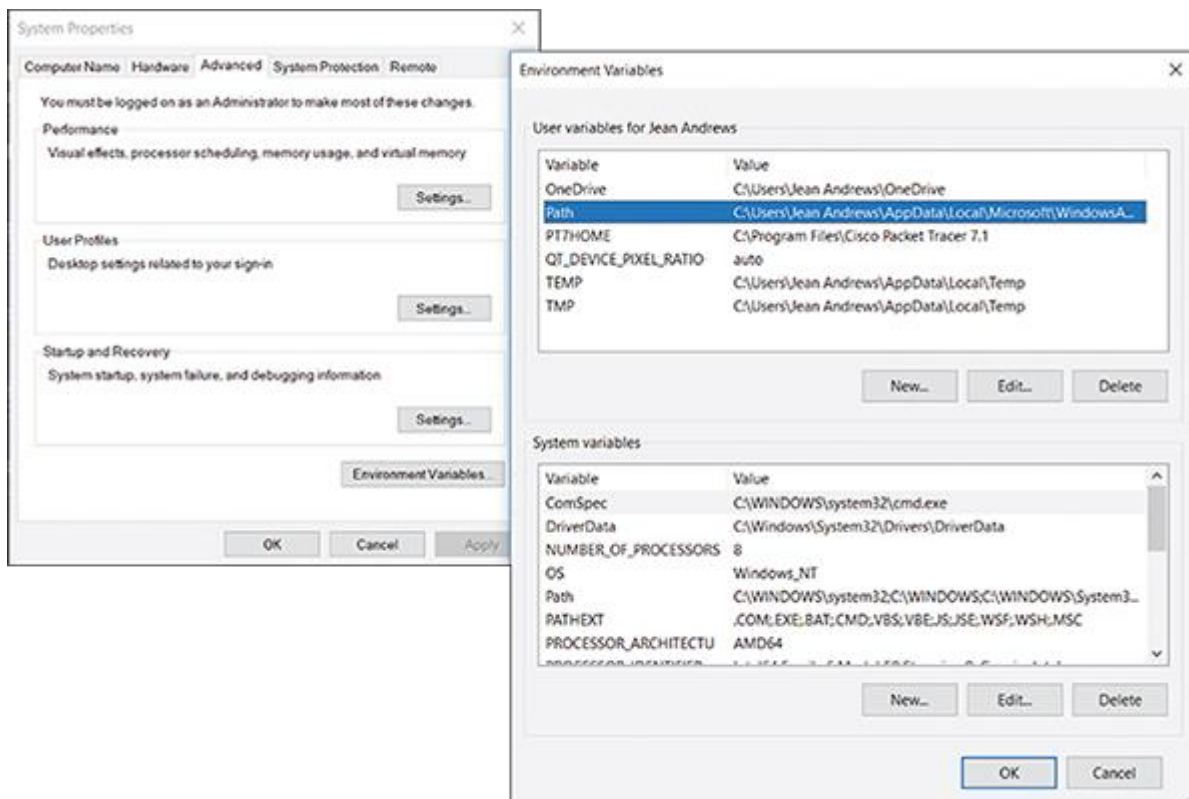
21-2b Basics of Scripting

Ready to learn a little scripting? Let's get started with some key terms:

An **environmental variable** (sometimes called a system variable) is information the OS makes available to a script. For example, the Windows and Linux PATH variable lists the paths (drives, directories, and subdirectories) the script can use. Another example is the TEMP variable, which tells a script where it can store its temporary files. To view and edit environmental variables in Windows, open the **System** window, and click **Advanced system settings**. On the Advanced tab of the System Properties dialog box, click **Environmental Variables**. See [Figure 21-23](#).

Figure 21-23

View, create, edit, and delete environmental variables



- Comments are text you put in a script to document the script. They can include your name, the date, the purpose of the script, and documentation that might help someone understand command lines in the script. **Comment syntax** refers to how you tag the text as a comment so it is not interpreted as a command. For example, in PowerShell, a line in the script file can hold a comment if you begin the line with a #.
- A **basic loop** executes the same commands multiple times until some condition is met.
- A **variable** is the name of an unknown data item and can be assigned a value, which is called initializing the variable. In PowerShell, a variable name is preceded by \$. You can assign a value to a variable using the equal symbol (=).
- A data type determines what type of value a variable can be assigned. Two common data types are **integers** (whole numbers) and **strings** (text).

In the world of scripting and programming, it is common practice that the very first script or program a person learns how to write does nothing more than say “Hello World.” Although it is a simple program, it proves you know how to create, save, and execute a script. Let’s create one in a shell script:

1. At an Ubuntu shell prompt, make sure the current directory is your home directory. Then enter this command, saving the .sh file in your home directory:

```
echo "echo Hello World" > my-script.sh
```

2. To assign execute permission to the file, run this command:

```
chmod u=rwx my-script.sh
```

3. To execute a shell script, type ./ before the script file name. Enter this command (see [Figure 21-24](#)):

```
./my-script.sh
```

Figure 21-24

A simple shell script is created and executed

```

jean@spock:~$ echo "echo Hello World" > my-script.sh
jean@spock:~$ ls -l
total 12
drwxrwxr-x 2 jean jean 4096 Sep 27 21:09 Hold
-rw-rw-r-- 1 jean jean  17 Sep 27 21:17 my-script.sh
-rw----- 1 jean jean   4 Sep 12 19:57 MyTestFile.txt
jean@spock:~$ chmod u=rwx my-script.sh
jean@spock:~$ ls -l
total 12
drwxrwxr-x 2 jean jean 4096 Sep 27 21:09 Hold
-rwxrw-r-- 1 jean jean  17 Sep 27 21:17 my-script.sh
-rw----- 1 jean jean   4 Sep 12 19:57 MyTestFile.txt
jean@spock:~$ ./my-script.sh
Hello World
jean@spock:~$ _

```

Annotations:

- ← Create the script (points to `echo "echo Hello World" > my-script.sh`)
- ← Execute the script (points to `./my-script.sh`)
- ← Script results (points to `Hello World`)



Now let's create a PowerShell script with a loop using Windows 10 **PowerShell ISE**, the PowerShell Integrated Scripting Environment, where scripts are created and tested:

1. In the Windows 10 search box, type **powershell** and then click **PowerShell ISE**. The PowerShell ISE window opens.
2. PowerShell ISE does dynamic type checking. As the command is interpreted, color coding is added to indicate the purpose of what you type. Type the following lines in the script pane (see [Table 21-7](#)):

Table 21-7

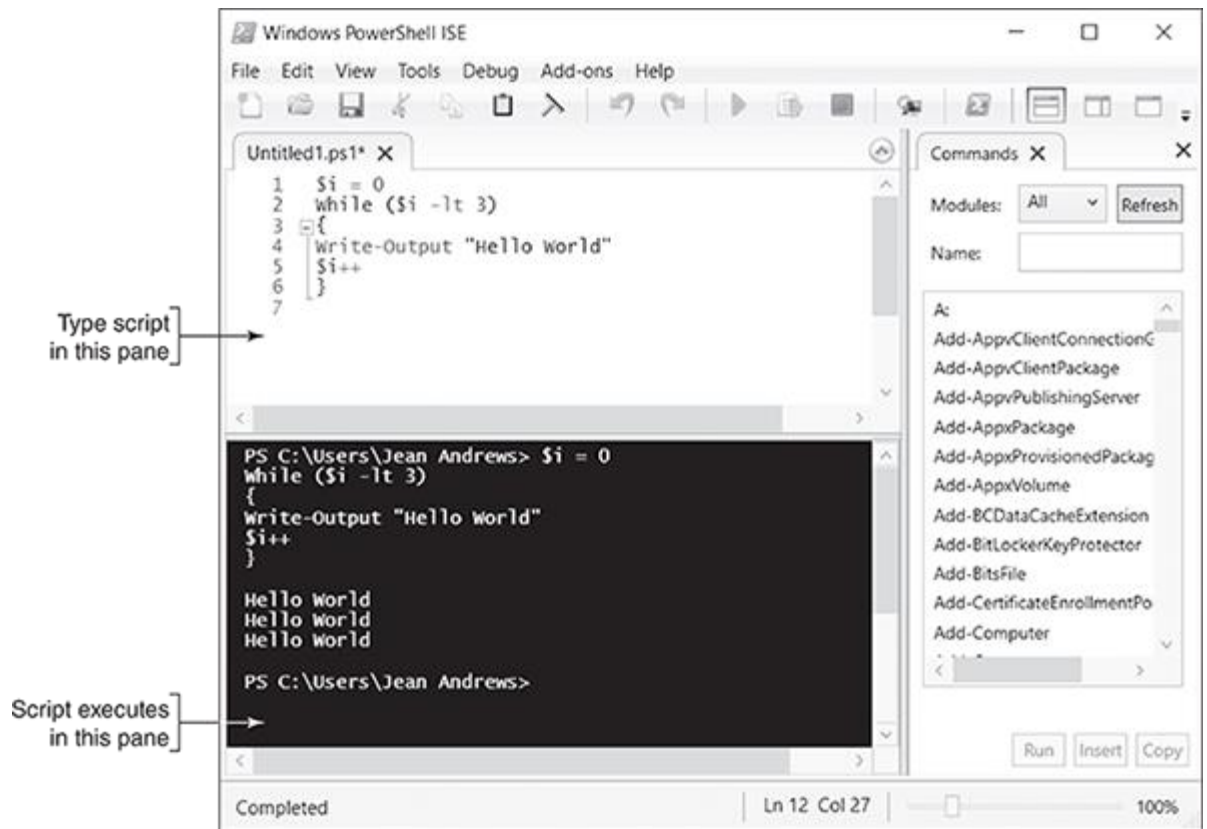
Commands to Enter for a Script with a Basic Loop

Command	Purpose
<code>\$i = 0</code>	Defines an integer variable i that is assigned the value 0
<code>While (\$i -lt 3)</code>	Continues looping as long as the variable is less than 3
<code>{</code>	Defines the beginning of the loop
<code>Write-output "Hello World"</code>	Displays "Hello World"
<code>\$i++</code>	Adds one to the i variable
<code>}</code>	Defines the end of the loop

3. To execute the script, click the **Run Script** button, or click **File** and then click **Run**. The script is executed in the lower pane. See [Figure 21-25](#).

Figure 21-25

A PowerShell script with a basic loop is executed



4. To save your script, click **File**, click **Save**, and save the script to your desktop. Name the script **MyLoopScript**. By default, the .ps1 file extension is assigned to the file.
5. Close the **PowerShell ISE** window, and open a standard **PowerShell** window.
6. By default, running PowerShell scripts is disabled. To set the execution policy so scripts will run except those downloaded from the Internet without a valid digital signature, enter this cmdlet:

Set-ExecutionPolicy RemoteSigned

7. Run this alias cmdlet to go to your Windows desktop folder:

cd desktop

8. Run the **dir** alias cmdlet to list the contents of your desktop folder. You should see the script file listed.
9. To execute a PowerShell script, begin with **./**. Enter this cmdlet to execute your script:

./MyLoopScript

Those are the basics of scripting. Now, let's discuss why IT personnel write scripts and examine various instances when you might want to use a script.

21-3 Uses for Scripting

Core 2 Objective

- 4.8

Identify the basics of scripting.

As you just learned, scripts can be used to write simple programs to perform tasks such as writing the “Hello World” statement. However, more complicated uses for scripts include the following:

- Restarting systems and services remotely
- Automating various administrative processes such as application installs, updates, and backups
- Remapping network file shares for users to access and save files across the network
- Installing and removing applications from systems
- Gathering information from network nodes, such as their IP address and MAC address information
- Initiating updates for various applications and the operating system itself

As you further explore scripting, you may want to start learning to script by writing one to rename files or collect information from computers on your network.

Applying Concepts

Using PowerShell to Assist in System Administration

- **Est. Time:** 30 minutes
- **Core 2 Objective:** 4.8

PowerShell can make Windows system administration easier through automation. In this activity, you will use the PowerShell ISE to script the following:

- Create a TestUser account without a password
- Create a MyTestDocuments directory in the My Documents folder of the TestUser
- Create four text documents named File1, File2, File3, and File4 in that newly created MyTestDocuments directory
- Check the status of DHCP Client service
- Restart the Explorer service
- Get a list of all your computer’s information

1. **1**

In the Windows 10 search box, type **powershell** and then click **Run as administrator** under the PowerShell ISE. The PowerShell ISE window opens.

2. **2**

Run the command **Set-ExecutionPolicy RemoteSigned** in the bottom window. This allows you to run locally written and signed scripts you get from the Internet.

3. **3**

Enter the commands listed in [Table 21-8](#) in the top pane of the PowerShell ISE. After all the commands have been entered, the screen should look like the one in [Figure 21-26](#).

Table 21-8

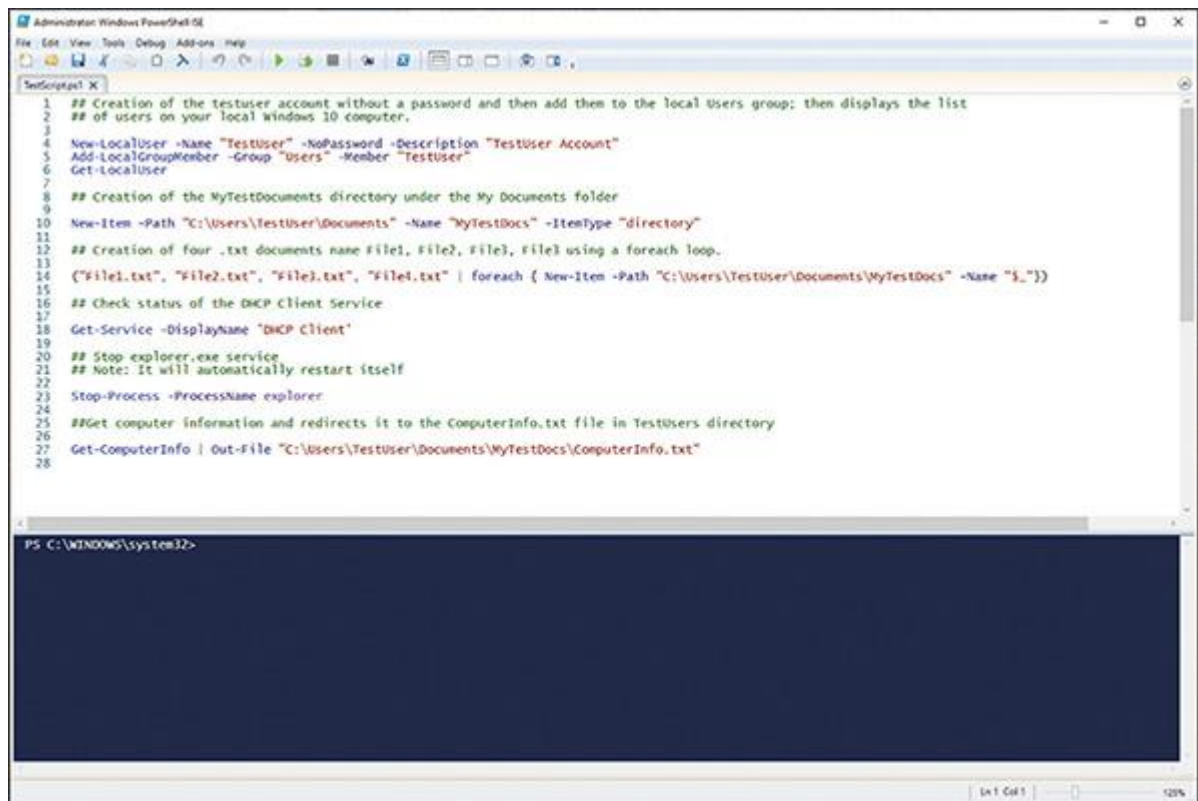
PowerShell Commands to Enter for the Script

Command	Purpose
<code>New-LocalUser -Name "TestUser" -NoPassword</code>	Creates a new local user named TestUser and sets the account to not have a password
<code>Add-LocalGroupMember -Group "Users" -Member "TestUser"</code>	Adds the TestUser account to the Users group in Windows 10
<code>Get-LocalUser</code>	Gets a list of users from your Windows 10 computer
<code>New-Item -Path "C:\Users\TestUser\Documents" -Name "MyTestDocs" -ItemType "directory"</code>	Creates a new directory named MyTestDocs in the TestUser's Documents directory
<code>("File1.txt", "File2.txt", "File3.txt", "File4.txt" foreach { New-Item -Path "C:\Users\TestUser\Documents\MyTestDocs" -Name "\$_" })</code>	Uses a foreach loop to create four new text files in the previously created MyTestDocs directory
<code>Get-Service -DisplayName 'DHCP Client' /</code>	Gets the status of the DHCP Client Service
<code>Stop-Process -ProcessName explorer</code>	Stops the explorer process (It will auto-restart.)
<code>Get-ComputerInfo Out-File "C:\Users\TestUser\Documents\MyTestDocs\ComputerInfo.txt"</code>	Gets information about your local computer and redirects it to a file



Figure 21-26

The PowerShell script is shown in the PowerShell ISE



The screenshot shows the PowerShell ISE window with a script titled 'TestScript1.ps1'. The script contains the following commands and comments:

```
1  ## Creation of the testuser account without a password and then add them to the local users group; then displays the list
2  ## of users on your local Windows 10 computer.
3
4  New-LocalUser -Name "TestUser" -NoPassword -Description "TestUser Account"
5  Add-LocalGroupMember -Group "Users" -Member "TestUser"
6  Get-LocalUser
7
8  ## Creation of the MyTestDocuments directory under the My Documents folder
9
10 New-Item -Path "C:\Users\TestUser\Documents" -Name "MyTestDocs" -ItemType "directory"
11
12 ## Creation of four .txt documents name File1, File2, File3, File4 using a foreach loop.
13
14 ("File1.txt", "File2.txt", "File3.txt", "File4.txt" | foreach { New-Item -Path "C:\Users\TestUser\Documents\MyTestDocs" -Name "$_" })
15
16 ## Check status of the DHCP Client Service
17
18 Get-Service -DisplayName "DHCP Client"
19
20 ## Stop explorer.exe service
21 ## Note: It will automatically restart itself
22
23 Stop-Process -ProcessName explorer
24
25 ##Get computer information and redirects it to the ComputerInfo.txt file in TestUsers directory
26
27 Get-ComputerInfo | Out-File "C:\Users\TestUser\Documents\MyTestDocs\ComputerInfo.txt"
28
```

The bottom pane of the ISE shows the command prompt with the prompt 'PS C:\WINDOWS\system32>'.



Note 11

In [Figure 21-26](#), the green lines with double number signs indicate that a line is a comment. Comment lines are not read during the execution of a script and are only there for documentation within the script. Comments are often used to document what the command below the comment is doing. Using comments in this way is a best practice when writing scripts. Someone may have to edit your script later, and comments can help them understand what is happening at each step in the script.

1. **4**

Save the script to your Desktop as TestScript.ps1.

2. **5**

Click the green play button in the top toolbar. Alternatively, you can press the F5 key to run the script within the ISE.

3. **6**

The script executes. You can confirm the user was created along with the files by navigating in File Explorer to C:\Users\TestUser\Documents\MyTestDocs.

The PowerShell ISE contains all the commands that can be run within your Windows environment. In enterprise networks, additional command

modules can be added to support domain-wide administrative functions, including creating domain users and email accounts and deploying Windows installations to remote systems across the network.

Let's look at another PowerShell script, shown in [Figure 21-27](#). Each line that begins with a # is a comment. In the image, you can see that the script is automating a variety of tasks.

Figure 21-27

An Advanced PowerShell script with comments



```
1
2 # This prompts the administrator to enter their credentials
3 # This ensures that the admin account is used for mounting the network drive
4 $cred = Get-Credential -Credential USERNAME
5
6 # Connects a network drive letter S to the Application Files folder on a system with
7 # IP address 192.168.128.3 by using the above credentials
8 New-PSDrive -Name "S" -PSProvider "FileSystem" -Root "\\192.168.128.3\Application Files" -Credential $cred
9
10 # Executes the installation file for Adobe Reader from the newly mounted network drive
11 # The argument list is used as answers to the installation wizard and accepts the End
12 # User License Agreement(EULA)
13 Start-Process -FilePath "S:\Application Files\PDF_Application\PDFRdrDC1800920044_en_US.exe" -r
14 -ArgumentList "/s /all /rs /rps /msi /norestart /quiet EULA_ACCEPT=YES"
15
16 # Collect System Information including Computer Name, Operating System Info,
17 # Processor Info, and Installed Memory
18
19 echo "This computers name is $env:COMPUTERNAME" `n
20
21 echo "Operating System Information"
22 Get-ComputerInfo OsName, OsVersion, OsBuildNumber, WindowsVersion
23
24 echo "Processor Information"
25 Get-CimInstance -ClassName Win32_Processor | Select-Object -ExcludeProperty "CIM*"
26
27 echo "Installed Memory in GB"
28 (Get-CimInstance -ClassName Win32_PhysicalMemory | Measure-Object -Property capacity -Sum).sum /1gb
29
30 # Install the Windows Update Powershell Module and then run updates
31 Install-Module PSWindowsUpdate -Force
32 Get-WindowsUpdate -AcceptAll -Install
33
34 # Restart the Computer after 240 seconds
35 Restart-Computer -wait -Timeout 240
```

Line 4 is asking for the administrator's log in credentials and then storing them. This is useful if there is a need for the credentials at any point in the script. Line 8 is connecting a remote network share folder to the local machine. The folder will show up as the drive letter S. This is sometimes called mounting a network share.

Lines 13 and 14 are installing an application using the .exe file that is located on the newly mounted network share. Lines 19–28 are collecting and displaying information about the computer. In this example, the computer's name, its operating system, its processor information, and the amount of installed memory will be displayed. Using a script can assist a technician in gathering information from the systems on a network to create an accurate inventory—making an administrator's life somewhat easier.

Lines 31 and 32 are installing the Windows Update module to the local system's PowerShell and using the Get-WindowsUpdate cmdlet to update

the local system. In enterprise environments, this is extremely helpful to ensure all the network computers stay updated. Finally, in line 35, the script will restart the computer after a period of 240 seconds.

Exam Tip

The A+ Core 2 exam expects you to be familiar with scripts used for basic automation, restarting machines, remapping network drives, installing applications, automating backups, gathering information, and initiating updates. You are not expected to know how to write these scripts or understand each command line in a script.

Now that we have explored PowerShell, let's turn our attention back to Linux and how scripts are written for that operating system.

Applying Concepts

Using Scripts to Assist in Backups of Linux Directories

- **Est. Time:** 45 minutes
- **Core 2 Objective:** 4.8

Scripts can also be used to simplify the process of backing up important directories in Linux. This script will create a backup of the directories identified in [Table 21-2](#) earlier in this module.

1. **1**
Start your Ubuntu VM, and log in using your user name and password.
2. **2**
Open a terminal window by selecting the Terminal app from the Apps menu.
3. **3**
Type the commands in [Table 21-9](#), pressing the **Enter** key after each line.

Table 21-9

Shell Commands to Enter for the Script

Command	Purpose
<code>#!/bin/sh</code>	Loads the correct interpreter for your script
<code>DATE=\$(date +%Y-%m-%d-%H%M%S)</code>	Uses the system's date and time and formats it so we can use it as the file name %Y is the year, %m is the month, %d is the day, %H is the hour, %M is minute, and %S is second. By using this command, we create a unique file name each time the script is run.

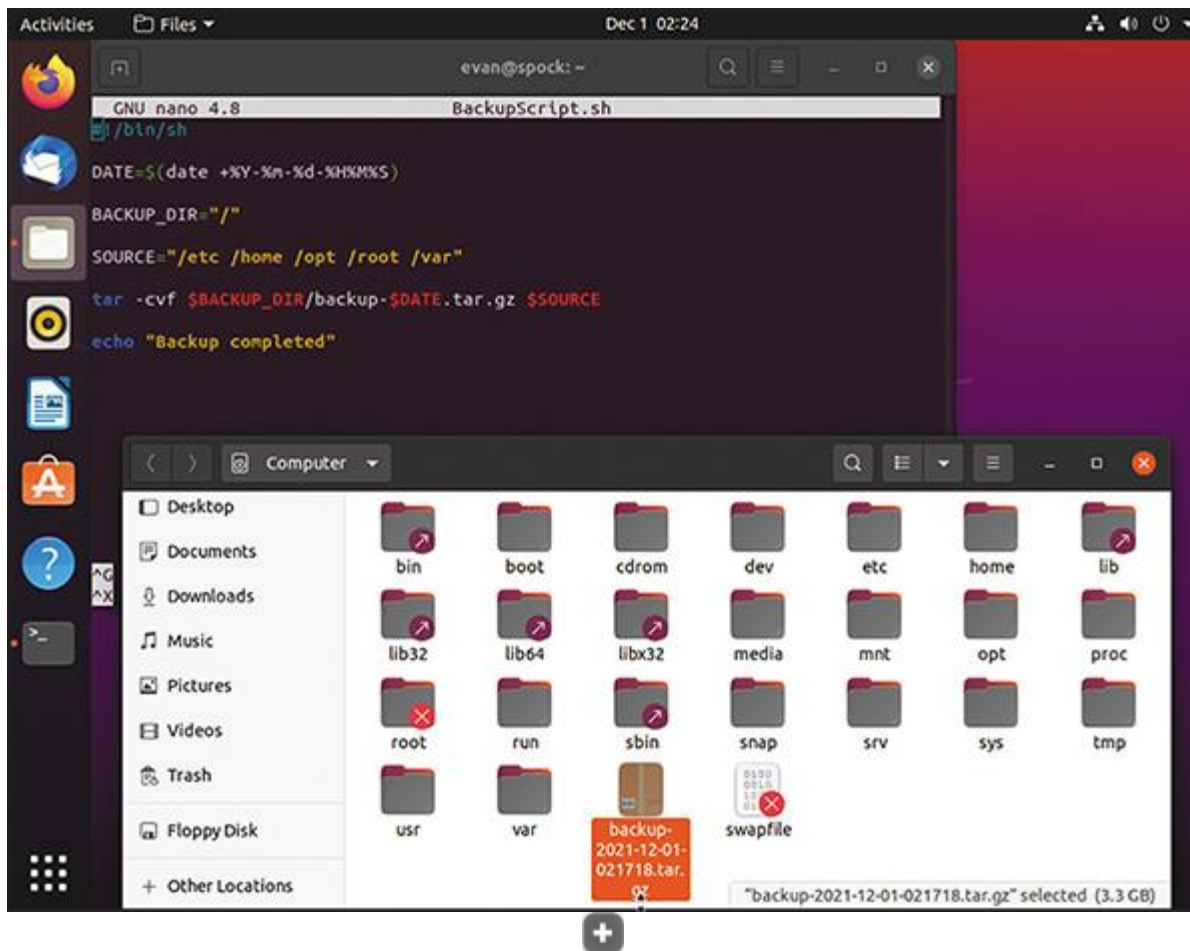
Command	Purpose
<code>BACKUP_DIR="/"</code>	Records the directory you want to save the backup to. In this example, we are saving the backup in the root directory.
<code>SOURCE="/etc /home /opt /root /var"</code>	Describes the files you want to back up
<code>tar -cvf \$BACKUP_DIR/backup-\$DATE.tar.gz \$SOURCE</code>	Creates the backup of our listed directories. The tar command stands for tape archive, and the -cvf options mean the following: <ul style="list-style-type: none"> -c: Creates the Archive -v: Displays Verbose information -f: Creates the archive with the given file name
<code>Echo "Backup completed"</code>	Displays the words "Backup completed" after the backup finishes



After it is complete, you should have something that looks like the script in the upper-left portion of [Figure 21-28](#). The lower portion of [Figure 21-28](#) shows the backup file that was created with the script. Your file name will be different because the date and time will be unique to your system and when you ran the script.

Figure 21-28

Shell script to create a backup of Linux directories



You can also make use of computer programming fundamentals to create a script in which a portion of the script only runs if a certain condition is met. An example of this would be a script that checks for the last time the computer checked for updates, and if the time since the last update check exceeds three days, it runs the update check. Other examples include checking for directories and files that have been modified or changed since a certain date and listing those files for the technician or only installing an application if it has not already been installed.

The script in [Figure 21-29](#) is used to check the status of the ClamAV installed during this module. First, there is a check to see if a network connection to the Internet has been established. Then, there is a check of the installed signature file and a comparison to the version found on the ClamAV server. The comparison of the installed signature file and the version on the ClamAV server is only able to run if there is a network connection. The installed version on the local computer is then checked and compared with the version on the server.

Figure 21-29

Script file checking network connection and file comparison

```
1 #!/bin/bash
2
3 echo
4 echo "Check if ClamAV database is up-to-date"
5 ((ping -w5 -c3 1.1.1.1 || ping -w5 -c3 8.8.8.8) > /dev/null 2>&1) && INTERNET="y" || (INTERNET="n")
6 if [ "$INTERNET" = "y" ]; then
7   echo "Check for latest update available"
8   echo -n "Date update available: "
9   DNSLKUP=$( host -t txt current.cvd.clamav.net )
10  date -d @$ ( echo $DNSLKUP | awk '{ print $4 }' | awk -F "." '{ print $4 }' )
11  echo -n "Signatures version: "
12  RMTSIGV=$( echo $DNSLKUP | awk '{ print $4 }' | awk -F "." '{ print $3 }' )
13  echo $RMTSIGV
14 else
15   echo "No connection to the Internet, Please Check Network Settings and Try Again"
16 fi
17
18 echo -n "Date when checked: "
19 date
20 echo "Currently installed on this machine."
21 CLAMINST=$( clamscan --version )
22 echo -n "Signature file version: "
23 LCLSIGV=$( echo $CLAMINST | awk -F "/" '{ print $2 }' )
24 echo $LCLSIGV
25 echo -n "Date of signature file: "
26 echo $CLAMINST | awk -F "/" '{ print $3 }'
27 echo -n "ClamAV version: "
28 echo $CLAMINST | awk -F "/" '{ print $1 }'
29 if [ "$INTERNET" = "y" ]; then
30   if [ "$LCLSIGV" = "$RMTSIGV" ]; then
31     echo "Same version of signature file as the latest on ClamAV server"
32   else
33     echo "Different version of signature file to latest on ClamAV server"
34   fi
35 fi
36 read -p "Press any key to exit..." -n1 -s
37 exit
```

Checking network connection

Checking for version installed on local computer

Comparison of installed file version with the version on the ClamAV server

Another use of these conditional operations could involve checking if a file has been modified or changed since the last backup ran, and if a file has changed, it is included in the next backup.

21-3a Scripting Security Considerations

There is also the chance that scripts could be used for malicious reasons. They can be used to download malware to a system, maliciously change system settings, or even cause a system to crash due to overloaded use of system resources.

For instance, a Microsoft Excel file with embedded scripts, called macros, stored in the cells of the spreadsheet, could contain malware. When a user opens the Excel file, the macros can activate, downloading malware from a malware server on the Internet. The malware can then be used to create a backdoor into the user's system or gather information from the network to send off to a potential adversary.

Therefore, it is best to ensure that only authorized scripts are allowed on your network and that standard users, by default, do not have permission to run scripts. This can be controlled by using local and remote group policies within the operating system or from a server that manages domain policy. How to use Group Policy in a Windows domain is covered in the module ["Securing and Sharing Windows Resources."](#)

When using scripts, you should also be mindful of the changes that are made to the system settings by the script. For example, a script may be used to uninstall several applications; however, the script might also inadvertently remove a file during the uninstall process that is needed by another program on the computer. This can be caused by applications sharing files or a newer version of an application using an old version of a file. An administrator would have to then spend time reinstalling the missing file or recovering it from a backup. This can cause a delay in the technician's already busy day.

It is also wise to review how a script is accessing and using system resources such as memory or the computer processor. Overuse or misuse of system resources can lead to an application malfunction or even a total system crash. Legitimate scripts that are written in JavaScript are sometimes found on webpages and run within your web browser. Some malicious JavaScript scripts found within a webpage can also be used to steal user credentials from your computer or personal data from your browsing history. They can even cause the browser application to crash because it malfunctioned or ran out of allocated memory. Therefore, you should be cautious when running scripts that were downloaded from the Internet or from unknown sources.

As you can see, learning to script can be helpful in completing the day-to-day tasks of a computer technician. By using a script, you can save time by automating processes that repeat themselves, such as updating a computer, collecting important system information for inventory, or simply copying files and installing or uninstalling applications.

21-4a **Module Summary**

Linux Operating System

- Distributions of Linux provide a shell prompt in the Linux terminal and might also provide a desktop with a GUI. The default command-line shell for Linux is the Bash shell.
- Ubuntu Desktop with its windows manager offers Ubuntu software to install apps and update Ubuntu.
- The root account in Linux has access to all features of the OS. When logged in to the root account, the user is called the superuser.
- Important Linux commands include apt-get, cat, chmod, chown, cp, df, dig, find, grep, ip, ls, man, mv, nano, ps, pwd, rm, top, and yum.
- Telnet and SSH can be used to remotely access a Linux computer. Telnet transmissions are not secured. All SSH transmissions are encrypted, and SSH is the preferred method of remote access for Linux and Unix systems.
- Windows and macOS are popular OSs for desktops and laptops, and Linux is popular as an application server OS. For compatibility, the FAT32 file system can be used by Windows, macOS, and Linux.

- Samba can be used to share files and share printer resources between the operating systems.

Scripting Software and Techniques

- Scripts are executed in a run-time environment without first being compiled like programs are.
- Script file types include batch files, PowerShell scripts, VBScript, shell scripts (for Linux and UNIX), Python scripts, and JavaScript.

Uses for Scripting

- Scripts may be used to automate administrative processes, restart machines, map network storage drives, install and remove applications, perform system updates, back up files, and even gather information from your network devices.
- Careful testing, reviewing, and monitoring of scripts is necessary to protect the integrity of the operating system and hardware resources.

21-4c Thinking Critically

These questions are designed to prepare you for the critical thinking required for the A+ exams and may use information from other modules and the web.

1. You are helping your friend troubleshoot a problem with their Linux server. You enter a common Linux command and discover it doesn't work exactly as you expected. What might be the problem, and what do you do next?
 1. The Linux installation is corrupted; restore the system from backup.
 2. The Linux shell is not the one you expected; use the echo \$SHELL command.
 3. You probably don't know how to use the Linux command; search the web for information about the command.
 4. The Linux shell is not the one you expected; use the echo \$0 command.
2. What is the full path to the home directory of the user account evan in Linux?
3. You are running a web server app in Ubuntu Server. Users complain that their browsers are loading webpages with errors. Where are you likely to find the log file where the web server reports its errors?
 1. /app/log
 2. /bin
 3. /var/log
 4. /root
4. In Linux, when logged in as a normal user with root privileges, which command must precede the apt-get command in the command line in order to install a program?
 1. sudo
 2. sudo user
 3. su
 4. root
5. Which file system does Linux currently use for the volume on which Linux is installed?

6. You have set up an FTP server in Ubuntu Server. Dominique, a user, calls to say she gets an error when trying to put a file in her /home/dominique/files directory. You look at the directory structure and see that you forgot to give the user ownership of the directory. Which command can fix the problem?

1. `chown dominique:dominique /home/dominique/files`
2. `sudo chmod u=rwx /home/dominique/files`
3. `sudo chown dominique:dominique /home/dominique/files`
4. `chmod u-rwx /home/dominique/files/`

7. What is the Linux nano editor command to save your changes and exit the editor?

8. You are managing an FTP server installed in Ubuntu Server. The server has created a very large log file, vsftpd.log. Which command is appropriate to search the log file for activity of the user nolan?

1. `sudo cat /var/log/vsftpd.log`
2. `grep "nolan" /var/log/vsftpd.log`
3. `sudo grep "nolan" /var/log/vsftpd.log`
4. `cat /var/log/vsftpd.log`

9. You work at an IT help desk and have been asked to set up 25 new user accounts in Active Directory. Your manager tells you to save time by using a PowerShell script that's available on a network share. You look at several script files named CreateNewUsers that are stored on the drive. Which one is likely to be the one you want?

1. The file with a .js file extension
2. The file with a .bat file extension
3. The file that is the largest
4. The file with a .ps1 file extension

10. A coworker is complaining that their connection to the Internet does not work. When you ask to examine the system, they show you a MacBook with Ubuntu Linux installed. Your initial assessment is that the Ethernet port does not have any lights flashing. You now want to check the configuration of the network card. Which of the following commands would allow you to view the necessary information?

1. `ipconfig /all`
2. `ip route`
3. `ip address`
4. `tracert`

11. Backups of files can help restore files when something goes missing. Which of the following commands can be used to copy a file named ImportantInfo.txt from the desktop to a directory named Backups in the root directory?

1. `copy Desktop/ImportantInfo.txt ./Backups`
2. `mv Desktop/ImportantInfo.txt ./Backups/ImportantInfo.txt`
3. `cp Desktop/ImportantInfo.txt ./Backups/ImportantInfo.txt`
4. `mkdir root/ImportantInfo`

12. While browsing the Internet, your web browser is not displaying the webpage correctly. The images and multimedia content are not showing or playing, and you want to watch your favorite music video on YouTube. What is one possible explanation for this issue?

1. The Internet is not working.
2. The JavaScript add-on is not activated.
3. Your wireless network card is suffering from interference.
4. The display driver needs updating.

13. You want to restart your Windows computer at the end of a PowerShell script you have built to update the operating system. Which of the following commands could be used to perform the restart in your script?

1. PowerOff Restart
2. shutdown
3. Reboot-Computer
4. Restart-Computer

14. Cait is trying to execute a script file to update a Linux workstation. The script has been written and is saved. When Cait checks the permissions using `ls -l`, the following information is displayed in the terminal window:

```
-r--r--r-- 1 evan evan 1437 Sep 18 07:01 UpdateLinux.sh
```

Cait needs to ensure that everyone can read and execute the file. Using the absolute mode format, which of the following commands would change the permissions on the UpdateLinux.sh file?

1. `chmod 555 UpdateLinux.sh`
2. `chown 757 UpdateLinux.sh`
3. `chmod u=rx,g=rx,o=rx UpdateLinux.sh`
4. `chmod 535 UpdateLinux.sh`

15. A new user is exploring the Linux terminal and wants to read an explanation of the `dig` command and see several examples. Which command would help them understand this command?

1. `dig man`
2. `man dig`
3. `help dig`
4. `dig --help`

21-4d Hands-On Projects

Hands-On Project 21-1

Practicing Linux Commands

Est. Time: 30 minutes

Core 2 Objective: 1.11

Practice the Linux commands listed in [Table 21-10](#) using the Ubuntu Server VM you created earlier in this module. You could also use the Live Ubuntu on a USB you created in the Core 1 module “[Hard Drives and Other Storage](#).” As you complete the commands, you’ll examine the directory structure, create a new directory, and put a blank file in it.

Table 21-10

Linux Commands to Practice

Task	Command	Description
1	<code>ls -l</code>	Lists files and directories in the current directory; in Linux, a directory is treated more like a file than it is in a Windows directory
2	<code>pwd</code>	Displays the full path to the current directory; when you first log in to a system, that directory is <i>/home/username</i>
3	<code>mkdir mydir</code>	Creates a directory named mydir; the directory is created in the current directory
4	<code>cd mydir</code>	Goes to the directory you just created in the <i>/home/username</i> directory
5	<code>touch myfile</code>	Creates a blank file named myfile in the current directory
6	<code>ls</code>	Lists the current directory’s contents
7	<code>cd ..</code>	Moves up one level in the directory tree
8	<code>cd /etc</code>	Changes the directory to the /etc directory, where text files are kept for configuring installed programs
9	<code>ls</code>	Lists the contents of the /etc directory
10	<code>cd /home</code>	Changes the directory to the /home directory
11	<code>ping 127.0.0.1</code>	Pings the loopback address; pinging continues until you stop it by pressing Ctrl+C
12	<code>ifconfig</code>	Displays TCP/IP configuration data
13	<code>man ifconfig</code>	Displays the page from the Linux Manual about the ifconfig command; type q to exit
14	<code>df -T</code>	Displays free space on the hard drive and the file system used
15	<code>exit</code>	Logs out; the login shell prompt appears, where you can log in again



Hands-On Project 21-2

Changing Permissions for an Ubuntu Directory

Est. Time: 30 minutes

Core 2 Objective: [1.11](#)

Follow these steps to change permissions for your home directory and then create a new user account to test these permissions:

1. **1**

Create a new user account named nadia. Log in to Ubuntu Server as nadia, and try to copy a file to your own home directory. For example, you can run this command to make a new copy of the mymemo file you created earlier in this module:

```
cp mymemo mymemo.nadia
```

When you do so, permission is denied.

2. **2**

Log back in to Ubuntu Server with your own account.

3. **3**

To install the chmod command, enter this command:

```
sudo apt-get install coreutils
```

4. **4**

Run the **chmod** command to give full read, write, and execute permissions to everyone for your home directory.

5. **5**

Log out and log back in to the system as nadia, and verify that the user nadia can now copy a file to your home directory.

Hands-On Project 21-3

Using Telnet and SSH

Est. Time: 30 minutes

Core 2 Objective: 1.11

Following directions given earlier in the module, install Telnet and SSH in your Ubuntu Server VM. If you get an error, Ubuntu Server might need updating. Commands to update Ubuntu are also given in the module.

On your Windows host computer, turn on Telnet Client from the Programs and Features area of Control Panel. Open a command prompt window, and remote in to your Ubuntu Server VM using both Telnet and SSH. Alternatively, you can use the PuTTY program shown in the module "[Securing and Sharing Windows Resources](#)" to complete this.

21-4e Real Problems, Real Solutions

Real Problem 21-1

Using Google Cloud Platform to Create a Red Hat Enterprise Linux VM and Update the OS

Est. Time: 45 minutes

Core 2 Objective: 1.11

4. **4**

You can also try using other commands, such as `df`, to check the disk space and file systems on your VM or even try creating, editing, copying, and moving files. You can refer to [Table 21-3](#) or use the manual pages to complete the following tasks:

1. **a**

Create and edit a new file in the home directory named `testfile`.

2. **b**

Create a directory in your home directory named `data` and copy the `testfile` from the home directory to the `data` directory.

3. **c**

Delete the `testfile` from the home directory.

4. **d**

Change permissions or ownership of the `testfile` that is in the `data` directory so the root account is the owner and has full permissions for the `testfile`.

5. **e**

Search for a specific word in the `testfile`.

6. **f**

View network interface configurations.

5. **5**

After you are finished, shut down the VM by running the `sudo shutdown` command.

Note 12

When you set up your Google Cloud account, your credit card information was required. If you're now done with your Google Cloud 90-day free trial, close your billing account so your credit card will not be charged at the end of your free trial period. Click the three-bar icon on the far-left side of the blue Google Cloud Platform menu bar, and click **Billing**. Then select **Account management**, and click **CLOSE BILLING ACCOUNT** at the top. Google normally will not automatically charge you after the trial period ends unless you upgrade your account to a full account, but it's better to be safe and close your billing account to prevent any unnecessary problems.

Real Problem 21-2

Configuring Network Adapters

Est. Time: 15 minutes

Core 2 Objective: 1.11

A user reports that their Ubuntu virtual machine cannot access the Internet. After taking control of the VM, you identify the following:

- The network card is enabled.

- The web browser is not loading linux.org.
- Other users are not reporting any issues browsing the Internet.

You suspect there is a problem with the network settings, and you will need to reconfigure the adapter to support full network connectivity. You decide that troubleshooting this error will be easily fixed through the terminal.

1. What command would you run to verify the current network card ip configuration?
2. You are shown that the inet address is 169.254.9.18. What does this information tell you? Describe how you might correct this problem using the terminal.
3. After performing the first two steps, you are shown a new ip address of 192.168.128.14. How can you verify connectivity to linux.org?

Real Problem 21-3

Installing and Using Samba

Est. Time: 30 minutes

Core 2 Objective: 1.11

Remember that Samba can be used to share files and printers between Linux and Windows systems. For this activity, you will need both your Windows 10 VM from the module “[Installing Windows](#)” and your Ubuntu VM from earlier in this module. Ensure both VMs are started and running, and then complete the following steps:

1. **1**
In the Ubuntu VM, open the terminal window, and run the following:

```
sudo apt update
```

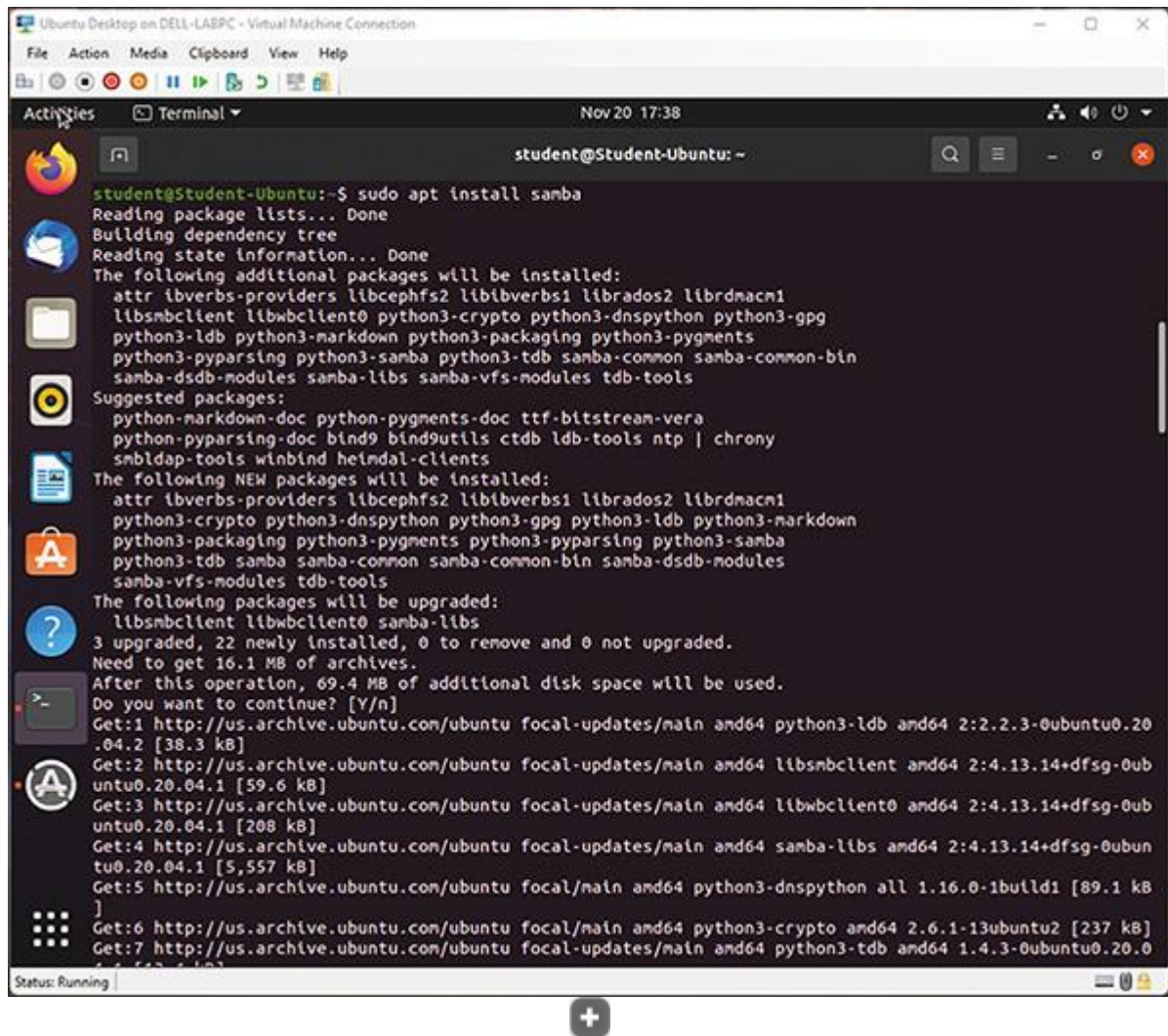
```
sudo apt install samba
```

Enter your sudo password, and then press **Enter**. When asked if you want to continue, press **Enter**.

[Figure 21-31](#) shows the installation of Samba in terminal.

Figure 21-31

Installation of the Samba service on Ubuntu



```
student@Student-Ubuntu:~$ sudo apt install samba
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  attr ibverbs-providers libcephfs2 libibverbs1 librados2 librdmacn1
  libsmclient libwbclient0 python3-crypto python3-dnspython python3-gpg
  python3-ldb python3-markdown python3-packaging python3-pygments
  python3-pyparsing python3-samba python3-tdb samba-common samba-common-bin
  samba-dsdb-modules samba-libs samba-vfs-modules tdb-tools
Suggested packages:
  python-markdown-doc python-pygments-doc ttf-bitstream-vera
  python-pyparsing-doc bind9 bind9utils ctdb ldb-tools ntp | chrony
  smbldap-tools winbind heimdal-clients
The following NEW packages will be installed:
  attr ibverbs-providers libcephfs2 libibverbs1 librados2 librdmacn1
  python3-crypto python3-dnspython python3-gpg python3-ldb python3-markdown
  python3-packaging python3-pygments python3-pyparsing python3-samba
  python3-tdb samba samba-common samba-common-bin samba-dsdb-modules
  samba-vfs-modules tdb-tools
The following packages will be upgraded:
  libsmclient libwbclient0 samba-libs
3 upgraded, 22 newly installed, 0 to remove and 0 not upgraded.
Need to get 16.1 MB of archives.
After this operation, 69.4 MB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-ldb amd64 2:2.2.3-0ubuntu0.20.04.2 [38.3 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libsmclient amd64 2:4.13.14+dfsg-0ubuntu0.20.04.1 [59.6 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libwbclient0 amd64 2:4.13.14+dfsg-0ubuntu0.20.04.1 [208 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 samba-libs amd64 2:4.13.14+dfsg-0ubuntu0.20.04.1 [5,557 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal/main amd64 python3-dnspython all 1.16.0-1build1 [89.1 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal/main amd64 python3-crypto amd64 2.6.1-13ubuntu2 [237 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-tdb amd64 1.4.3-0ubuntu0.20.04.2 [44.3 kB]
Status: Running
```

2. **2** Now that Samba is installed, you need to create a directory for Samba to share with the network. Type the following command into the terminal window, substituting your user name where `<username>` is shown:

```
mkdir /home/<username>/sambashare/
```

3. **3** Now that the directory you want to share is created, you need to ensure that the Samba configuration file includes it as a shared item for the network. Run the `sudo /nano /etc/<username>/sambashare` command in terminal, and then add the following lines to the end of the file:

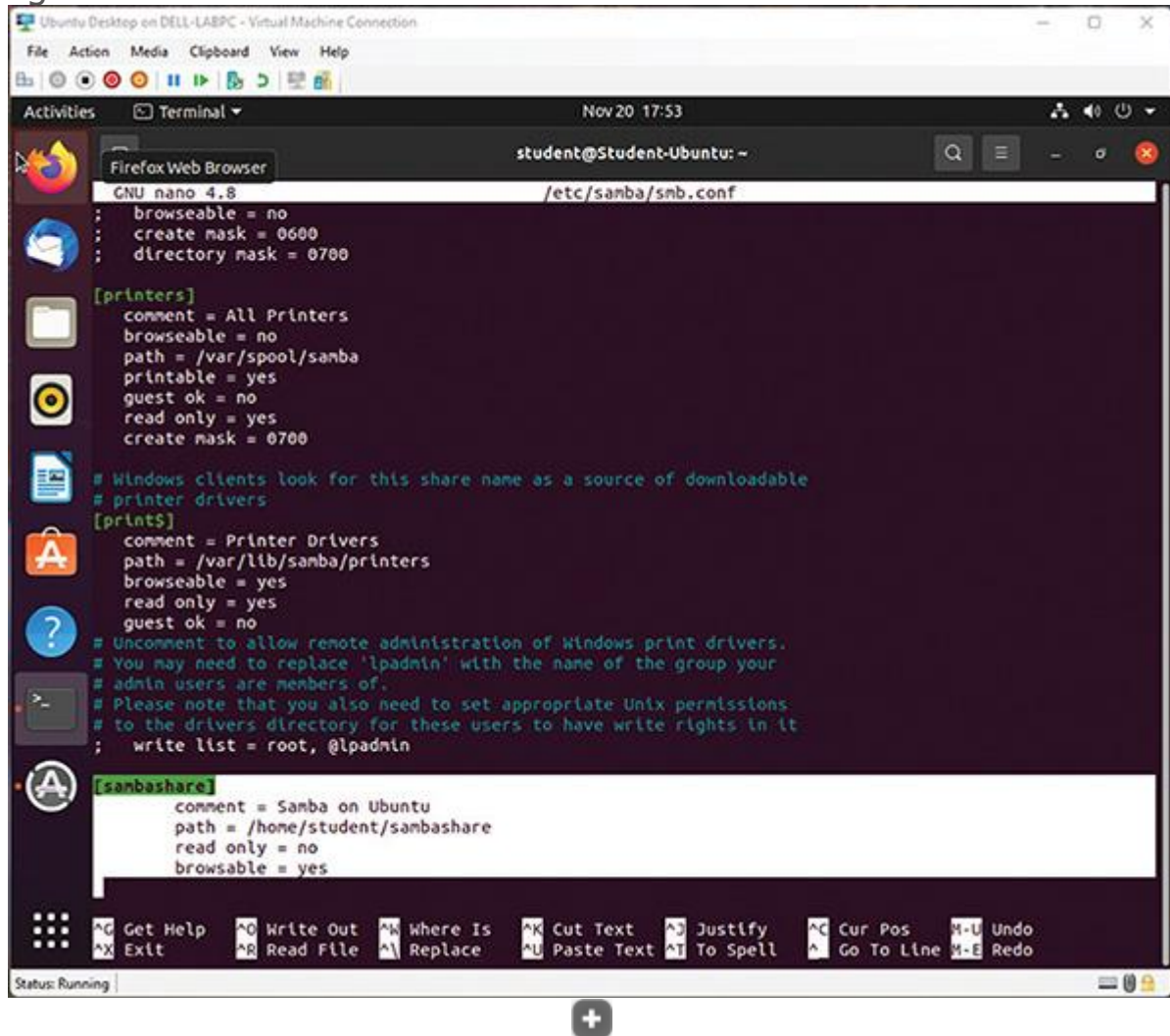
```
[sambashare]
```

- o `comment = Samba on Ubuntu`
- o `path = /home/<username>/sambashare`
- o `read only = no`
- o `browsable = yes`

Then press **Ctrl+O** to save and **Ctrl+X** to exit the nano editor. [Figure 21-32](#) shows the lines added to the end of the file. They are highlighted in white.

Figure 21-32

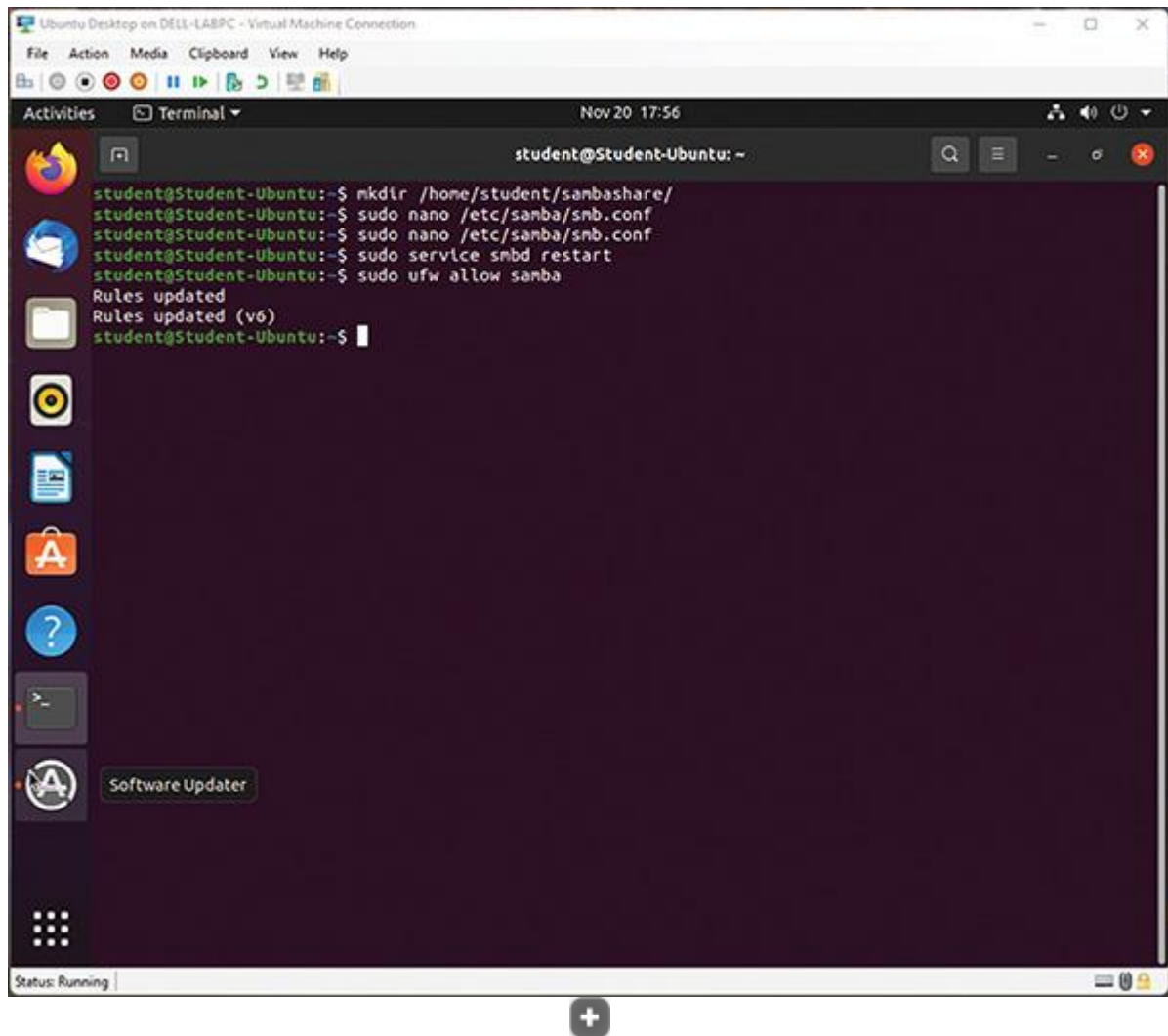
Editing the smb.conf file



4. **4** Restart the smbd service by running the `sudo service smbd restart` command. Then, allow Samba traffic through the Ubuntu firewall by running the `sudo ufw allow samba` command, as shown in [Figure 21-33](#).

Figure 21-33

Restarting the Samba service and modifying the firewall rules to allow Samba connections through



5. **5** Next, you will set up a user for the Samba service by entering the following command (using your own Ubuntu user account name):

```
sudo smbpasswd -a username
```

Enter your desired password twice, and then close the terminal. Be sure to remember this password, as you will need it to connect from your Windows VM and access the shared directory.

6. **6** To view the sambashare folder, click the **sambashare folder** icon on the left side of the Ubuntu desktop. The sambashare folder should be shown in the Home directory, as shown in [Figure 21-34](#). Any documents or files that are placed in the sambashare directory can now be accessed across the network. Add a document or file to the sharefolder by first creating the document using LibreOffice writer or another software application and then saving it in the correct sambashare directory of your home directory. You can see an example Accounts.txt file is saved to the proper directory in [Figure 21-35](#).

Figure 21-34

Sambashare directory (selected on the right) is displayed on the Ubuntu system

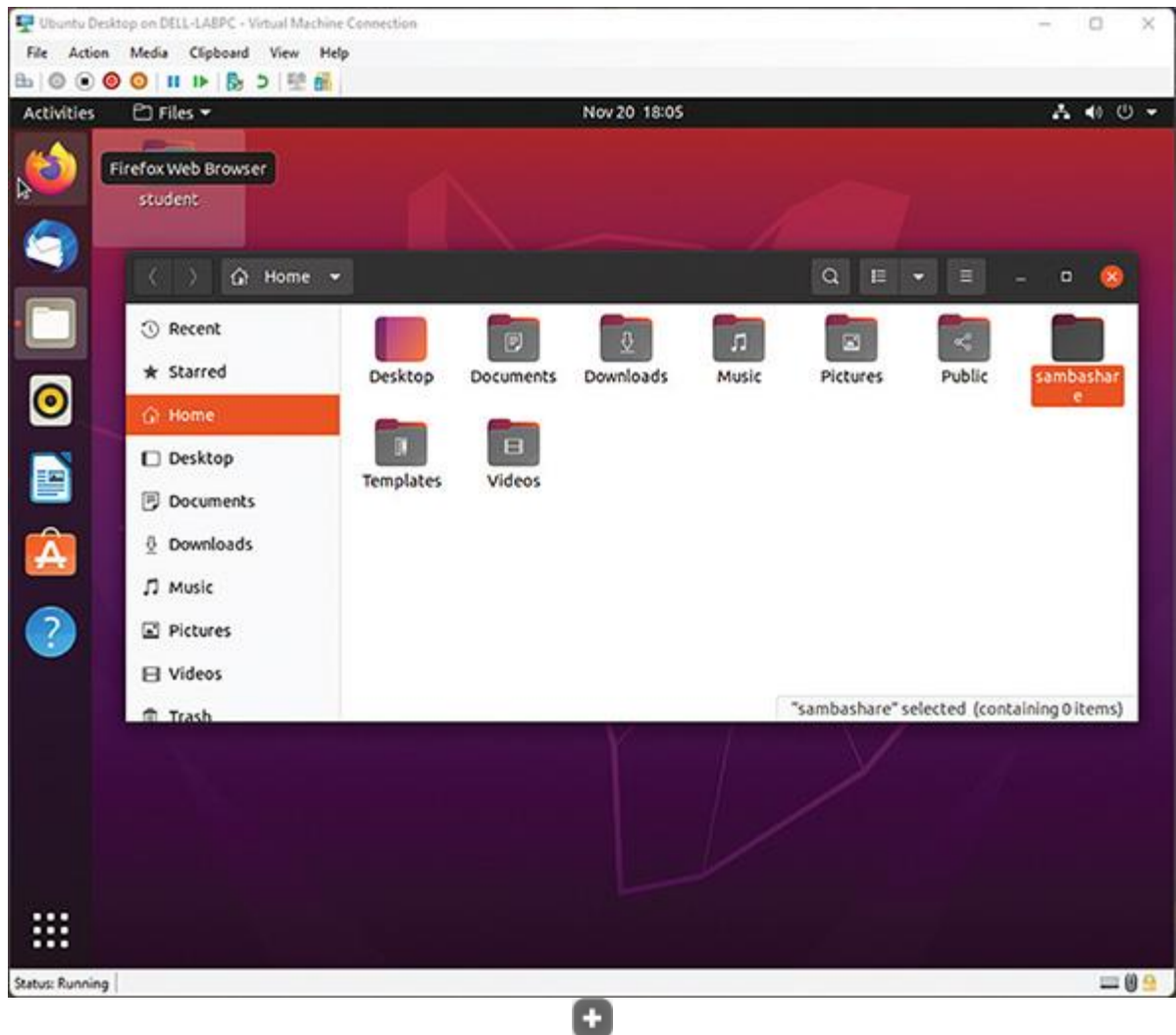
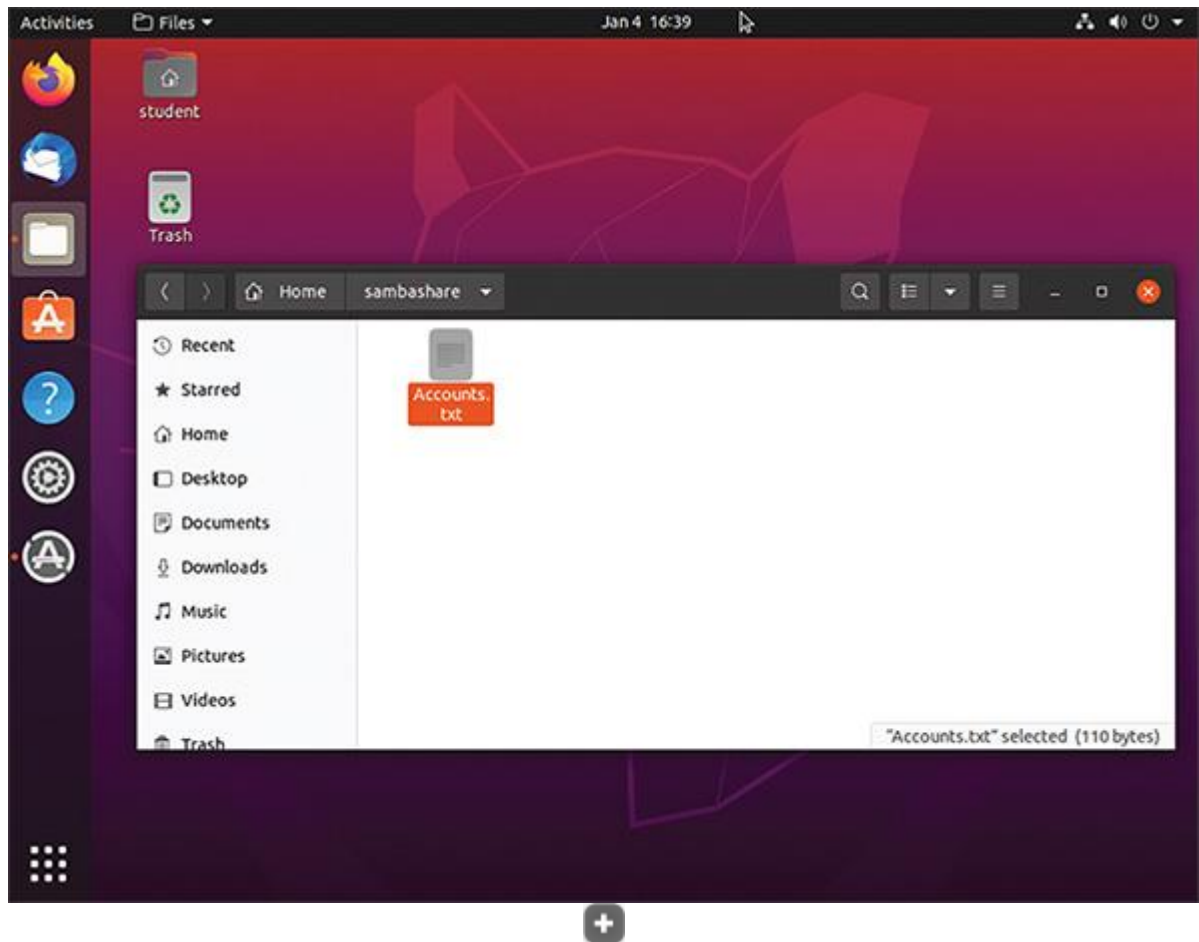


Figure 21-35

Accounts.txt file shown in the sambashare folder on Ubuntu



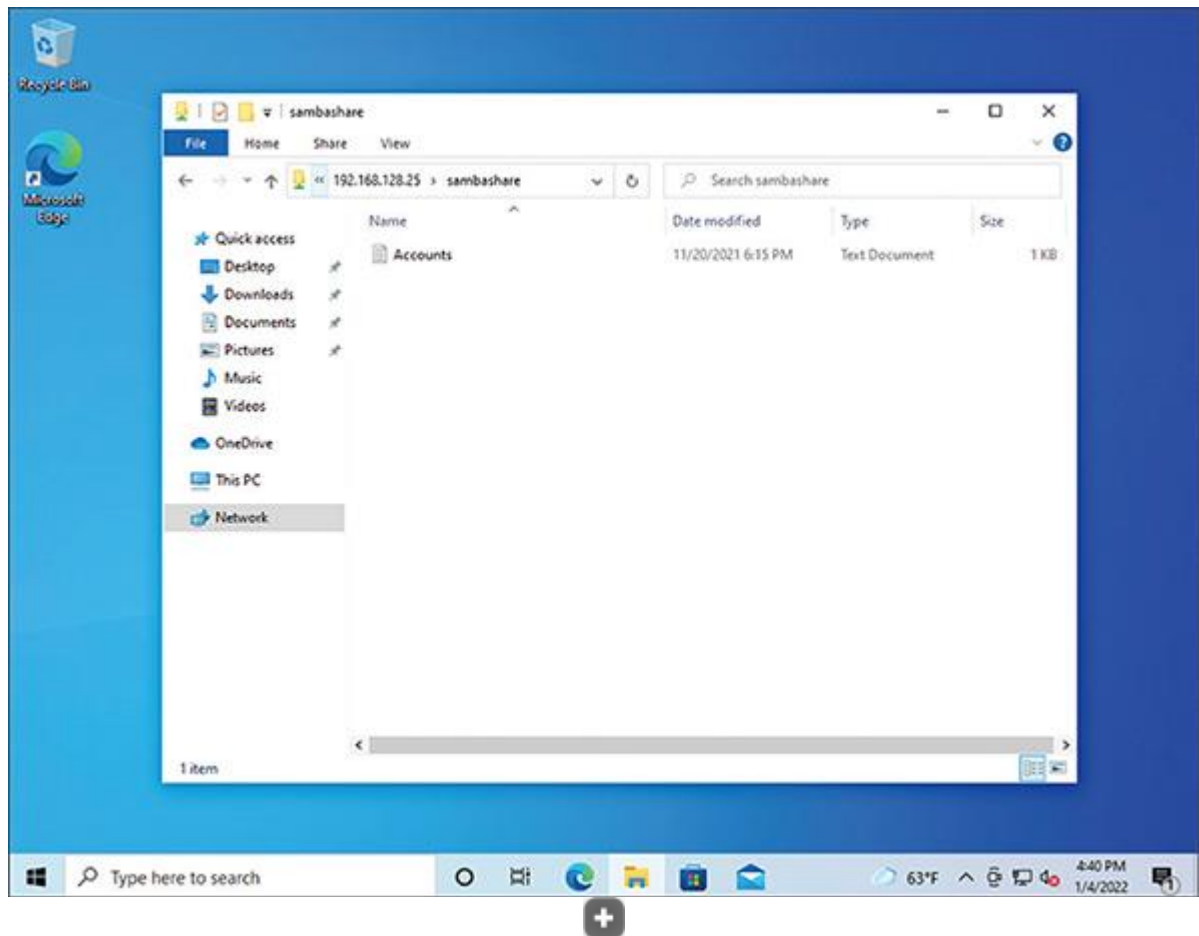
7. **7** To access the files that are shared via samba from your Windows 10 VM, first connect to your VM, and then open **File Explorer** from the taskbar. Type the following into the directory path: `\\ip-address\sambashare`.

Be sure to use the correct ip address of your Ubuntu VM in place of the word ip-address. This can be found by running the `ifconfig` command from the terminal window. You will be prompted to enter your Ubuntu user name and password.

8. **8** You should now be able to see the file you saved in [step 6](#) in File Explorer. See [Figure 21-36](#), and note that the example Accounts.txt file is shown.

Figure 21-36

Accounts text file shown on Windows



9. **9**

Alternatively, you could connect from your host machine to the sambashare directory. If you are using Windows as your host OS, follow [step 7](#). If you are on a Mac, use the Finder menu, and select **Go > Connect**, and then enter the following: **smb://ip-address/sambashare**

Again, be sure to change “ip-address” to the Ubuntu VM’s.

10. **10**

If you have a printer attached to your host machine, try printing the document as well.

Real Problem 21-4

Preparing for the A+ Core 2 Exam

Est. Time: Unknown

Core 2 Objectives: [1.1–4.9](#)

In this text, the modules “[The Complex World of IT Professionals](#)” through “[Linux and Scripting](#)” and the appendix “[Safety Procedures and Environmental Concerns](#)” prepare you for the A+ Core 2 exam. When you have completed these modules and appendix, you are ready to make your final review of the A+ Core 2 objectives and sit for the exam. Read through the objectives listed at the beginning of this text and make sure you understand each objective. If you don’t understand an objective, reread that section in the text or do a general Google search. You may also want to form a study group as you prepare for the certification exam. Your instructor might also suggest other exam-prep tasks.

cd / ls cd /etc ls cd init ls ls -l rc.conf	Change to root of OS List the contents Move into "etc" (path type is "relative") List Move into "init" List List the size of a file (281 bytes)
cd ~ ls -l /etc/init/rc.conf	Change to the user's Home dir List size of file in 1 step
cd .. pwd	Move up 1 level Verify your location (print working directory)
cd ~ cd ../..	Return to home dir Move to the / dir using dots
file /etc/init/rc.conf cat /etc/init/rc.conf clear	Determine the file type (verify it's readable) (display file types using special characters) Open with "cat" command – allows user to view the contents of a file Clear the screen
grep root /etc/passwd	Find all instances of the word "root" in the passwd file. (4 times)
ls -l pwd mkdir mydir cd mydir touch myfile mv myfile.txt /home ls cd .. cd /etc ls cd /home ping 127.0.0.1 "Ctrl C" ifconfig man ifconfig df -T exit help	List directories and files in current (file & dir attributes, ownership, and sizes) Long List Display the full path of current Create dir in present "mydir" Change to new dir Create an empty file "myfile" Move file from current location to another and you can rename the files List contents of dir Move up 1 level Change to "etc" List contents Home dir Ping to validate your loopback Quit command (stop a running loop) View status and statistics of your connections (server ip addr) Find and record your ip address "inet addr: 10.0.2.15 , Bcast:.255, Mask .0" "man page" contains info about the command and it's behavior and options View the man page for Ifconfig (q to exit) View the free space on drive & file system used Return to the login screen Display the built-in tool documentation when used with dt
rm -r <folder name> nano myfile01.txt cp myfile.txt /home ls /home/admin chmod o+w myfile02.txt	Remove folder nano and vim are two text-based file GNU editors in Linux. Opens file in nano program. Copy files and directories from current location to another List contents of the specified directory Config permissions of "others" and "write" to a file :-rw-rw-rw- or read and write for the owner/read and write for the group/read and write for others.

sudo apt install mc	<p>“sudo” command elevates the privileges of the user to a “superuser” who can perform administrative tasks. The “su” command switches the context to another user, typically used to elevate the current user to “root.” Use “su” very sparingly and instead use “sudo”.</p> <p>The “apt” command is used to install, update and remove packages in Debian-based systems. In a Redhat-based system, the “yum” command is used; however, today, “dnf” is used in these systems.</p> <p>The “mc” option is the “Midnight Commander” package to be installed.</p>
sudo apt remove mc	The “apt” command can remove software packages from Debian-based Linux systems.

Exercise 1 - Files and Directories in Linux

The bash shell can be used to manage files and directories on a linux operating system. Specific commands are used to accomplish these tasks.

Exercise 2 - Viewing and Modifying Permissions in Linux

On a Linux operating system, specific commands can be used to view and manage the permissions of files and directories.

Exercise 3 - Managing Software Packages in Linux

Software applications on Linux operating systems can be installed and managed through the bash interface. Specific commands can be used to accomplish this task.

Task 1 - Installing Packages in Linux

In this task, you will learn to install application packages in Linux.

Task 2 - Uninstalling Software Packages in Linux

In this task, you will uninstall the **Midnight Commander** software package.

Exercise 1 - Scanning for Malware

Malware is a type of software that is designed to harm a system.

Malware can cause the following:

- System lockups
- System slowness
- Application crashes
- Applications generating unexpected output

Most operating systems have built-in anti-malware programs that may help to ward-off irritants such as a virus (a malicious code), unwanted email messages, and spyware. These irritants can be a waste of time for the user, while some of them might lead to a system malfunction, resulting in data or business loss. As new malware is released

on a daily basis, it is good to keep the anti-malware application updated all the time. Most of these applications can perform an automatic update without user intervention. You should also see the possibility of configuring the application to perform an auto scan.

Which of the following can be the outcome of a malware attack? [Choose all that apply]

- ☒ Application generating unexpected output
- ☒ System lockups
- ☒ System slow down
- ☒ Application crash

Which of the following malware attributes can be viewed in the details window when a threat is detected?

- ☒ Affected items
- ☒ Details
- ☒ Date
- ☒ Alert level
- ☒ Category

Which of the following allows users to create point-in-time configuration snapshots that can be used to return a Windows machine to an early configuration?

- ☐ Task Manager
- ☒ System Restore
- ☐ Device Manager
- ☐ Windows Defender

On the **Virus:DOS/EICAR_Test_File** window, review the following malware attributes:

- **Alert level**
- **Status**
- **Date**
- **Category**
- **Details**
- **Affected items**

Exercise 3 - Implement Malware Removal Best Practices

Windows System Restore allows users to create point-in-time configuration snapshots that can be used to return a Windows machine to an early configuration. The goal is to create a restore before remediation, and if one is not created ahead of time, then one should be created after malware removal.

Exercise 1 - Creating Basic Scripts

Different operating systems use different file types that can be used for scripting. Microsoft operating system can use Powershell scripts with a file extension of .ps or Command Prompt file types with a .bat file extension. In comparison, script files for Linux operating systems use the bash shell to create a file with a .sh extension.

Task 1 - Create a Linux Script

Script files can be used to automate repetitive administration tasks to minimize administrative effort. The most commonly used file for creating a script using a Linux operating system in the bash shell is the .sh file type.

In this task, a simple Linux script will be created.

Click the **Suse** icon in the bottom lefthand corner.

Click the **System** menu and select **Konsole** in the fly-out menu.

In the **~:bash - Konsole** window, type the following and press **Enter**:

```
touch Documents/plabdoc{1..10}.txt
```

Executing the command will create files that will be backed up later using a script.

In the **~:bash - Konsole** window, type the following and press **Enter**:

```
ls -l Documents
```

Executing the command will create ten text files in the Documents folder.

In the **~:bash - Konsole** window, type the following and press **Enter**:

```
mkdir Backup
```

In the **~:bash - Konsole** window, type the following and press **Enter**:

```
touch plabbackup.sh
```

Executing the touch command will create a file that will be used to create a backup script.

In the **~:bash - Konsole** window, type the following and press **Enter**:

```
vim plabbackup.sh
```

In the **~:vim-nox11 Konsole** window, press the **"i"** key.

enables editing the file using the vim application.

In the **~:vim-nox11 Konsole** window, type the following and press **Enter**:

```
#!/bin/bash
tar cfv /home/admin/Backup/plabbackup.tar
/home/admin/Documents/
```

The vim application is used to create a basic script file that will back up the documents in the Documents folder to the backup directory.

In the **vim** application, press the **Escape** key.

Type the following and press **Enter**:

```
:wq!
```

In the **~:bash - Konsole** window, type the following and press **Enter**:

```
chmod +x plabbackup.sh
```

*The chmod command is to be used with the **+** parameter to make the script file executable.*

In the **~:bash - Konsole** window, type the following and press **Enter**:

```
ls -l plabbackup.sh
```

In the **~:bash - Konsole** window, type the following and press **Enter**:

```
./plabbackup.sh
```

The script that was created will create a backup of the user's Documents folder to the Backup folder. The script can be automated to run at a specific time.

Task 2 - Create a Basic Powershell Script

Windows Powershell can be used to create scripts to automate specific administrative tasks, for example, mapping a shared network drive. Creating scripts can reduce the administrative effort needed to complete repetitive tasks.

In the **Administrator: Windows PowerShell** window, type the following and press **Enter**:

```
notepad.exe mapdrive.ps1
```

In the **mapdrive - Notepad** window, type the following:

```
New-PSDrive -Name K -PSProvider FileSystem -Root  
"\\plabdc01\Marketing" -Persist -Scope Global
```

Used to map a network drive = New-PSDrive

Reopen the **Administrator: Windows PowerShell** window.

Type the following and press **Enter**:

```
Set-ExecutionPolicy Unrestricted
```

Type **a** and press **Enter**.

Type the following in **Administrator: Windows PowerShell** window and press **Enter**:

```
.\mapdrive.ps1
```

Task 3 - Creating a Basic Batch Script File

The **Windows Terminal** can be used to create a batch script file with a file extension of *.bat.

These script files can be used to automate tasks and reduce administrative effort.

In this task, a simple batch script file will be created through the **Windows Terminal**.

Right-click **Start** and select **Windows Terminal**.

In the **Terminal** window, type the following and press **Enter**:

```
notepad.exe mapdrive.bat
```

Type the following in the **Notepad** file:

```
net use z: \\plabdc01\Marketing
```

Reopen the **Windows Terminal** window.

Type the following and press **Enter**:

```
.\mapdrive.bat
```

The batch file was successfully executed, mapping the shared folder to the device as a drive. Like the Powershell script, a batch file can be associated with a user's profile to map a drive on login.

Which file type can be used to create a script file on a Linux operating system?

- ☐ *.ps1
- ☐ *.bat
- ☒ *.sh
- ☐ *.txt

Which command is used to create a new file on a Linux operating system?

- ☐ ls -l
- ☒ touch
- ☐ mkdir
- ☐ chmod

Which command in Windows PowerShell can be used to map a network drive?

- ☒ New-PSDrive
- ☐ touch
- ☐ Set-ExecutionPolicy
- ☐ mkdir

What is the purpose of creating a script file? [Choose all that apply]

- ☒ Decrease administrative effort
- ☐ Delete a file
- ☒ Automate repetitive tasks
- ☐ Change into a directory

You are helping your friend troubleshoot a problem with their Linux server. You enter a common Linux command and discover it doesn't work exactly as you expected. What might be the problem, and what do you do next?

- ☒ a. The Linux shell is not the one you expected; use the `echo $0` command.
- ☐ b. The Linux shell is not the one you expected; use the `echo $SHELL` command.
- ☐ c. You probably don't know how to use the Linux command; search the web for information about the command.
- ☐ d. The Linux installation is corrupted; restore the system from backup.

You are running a web server app in Ubuntu Server. Users complain that their browsers are loading webpages with errors. Where are you likely to find the log file where the web server reports its errors?

- ☐ a. `/app/log`
- ☐ b. `/bin`
- ☒ c. `/var/log`
- ☐ d. `/root`

In Linux, when logged in as a normal user with root privileges, which command must precede the `apt-get` command in the command line in order to install a program?

- ☒ a. `sudo`
- ☐ b. `root`
- ☐ c. `su`
- ☐ d. `sudo user`

You have set up an FTP server in Ubuntu Server. Dominique, a user, calls to say she gets an error when trying to put a file in her `/home/dominique/files` directory. You look at the directory structure and see that you forgot to give the user ownership of the directory. Which command can fix the problem?

- ☐ a. `chown dominique:dominique /home/dominique/files`
- ☐ b. `sudo chmod u5rwx /home/dominique/files`
- ☒ c. `sudo chown dominique:dominique /home/dominique/files`
- ☐ d. `chmod u-rwx /home/dominique/files/`

You are managing an FTP server installed in Ubuntu Server. The server has created a very large log file, vsftpd.log. Which command is appropriate to search the log file for activity of the user nolan?

- ☐ a. `sudo cat /var/log/vsftpd.log`
- ☒ b. `grep "nolan" /var/log/vsftpd.log`
- ☐ c. `sudo grep "nolan" /var/log/vsftpd.log`
- ☐ d. `cat /var/log/vsftpd.log`

You work at an IT help desk and have been asked to set up 25 new user accounts in Active Directory. Your manager tells you to save time by using a PowerShell script that's available on a network share. You look at several script files named CreateNewUsers that are stored on the drive. Which one is likely to be the one you want?

- ☐ a. The file with a .js file extension
- ☐ b. The file that is the largest
- ☐ c. The file with a .bat file extension
- ☒ d. The file with a .ps1 file extension

A coworker is complaining that their connection to the Internet does not work. When you ask to examine the system, they show you a MacBook with Ubuntu Linux installed. Your initial assessment is that the Ethernet port does not have any lights flashing. You now want to check the configuration of the network card. Which of the following commands would allow you to view the necessary information?

- ☐ a. `ipconfig /all`
- ☒ b. `ip address`
- ☐ c. `ip route`
- ☐ d. `tracert`

Backups of files can help restore files when something goes missing. Which of the following commands can be used to copy a file named ImportantInfo.txt from the desktop to a directory named Backups in the root directory?

- ☐ a. `copy Desktop/ImportantInfo.txt ./Backups`
- ☐ b. `mv Desktop/ImportantInfo.txt ./Backups/ImportantInfo.txt`
- ☒ c. `cp Desktop/Importantfile.txt ./Backups/ImportantInfo.txt`
- ☐ d. `mkdir root/ImportantInfo`

While browsing the Internet, your web browser is not displaying the webpage correctly. The images and multimedia content are not showing or playing, and you want to watch your favorite music video on YouTube. What is one possible explanation for this issue?

- ☐ a. The Internet is not working.
- ☐ b. Your wireless network card is suffering from interference.

- ☐ c. The display driver needs updating.
- ☒ d. The JavaScript add-on is not activated.

You want to restart your Windows computer at the end of a PowerShell script you have built to update the operating system. Which of the following commands could be used to perform the restart in your script?

- ☐ a. Reboot-Computer
- ☒ b. Restart-Computer
- ☐ c. PowerOff Restart
- ☐ d. shutdown

Cait is trying to execute a script file to update a Linux workstation. The script has been written and is saved. When Cait checks the permissions using `ls -l`, the following information is displayed in terminal. `-r--r--r-- 1 evan evan 1437 Sep 18 07:01 UpdateLinux.sh` Cait needs to ensure that everyone can read and execute the file. Using the absolute mode format, which of the following commands would change the permissions on the `UpdateLinux.sh` file?

- ☒ a. `chmod 555 UpdateLinux.sh`
- ☐ b. `chown 757 UpdateLinux.sh`
- ☐ c. `chmod u5rx,g5rx,o5rx UpdateLinux.sh`
- ☐ d. `chmod 535 UpdateLinux.sh`

A new user is exploring the Linux terminal and wants to read an explanation of the `dig` command and see several examples. Which command would help them understand this command?

- ☐ a. `dig --help`
- ☐ b. `help dig`
- ☒ c. `man dig`
- ☐ d. `dig man`