9

BACKING UP YOUR DATA

Having a reliable, well-defined, and well-implemented backup strategy is one of the best defenses any network can have against malicious or accidental data loss. Whether you mistakenly delete a folder of critical documents, an adversary executes ransomware inside your network, or a natural disaster destroys your devices, backups can save you from catastrophe.

This chapter introduces various backup considerations, including different backup types, creating backup schedules, the value of onsite and offsite backups, what to back up, and backup storage options. Finally, we'll show how to implement different solutions within your network.

Backup Types

You should consider three types of backups when implementing your backup schedule: *full*, *incremental*, and *differential*.

Full Backup

- A full backup contains a complete copy of everything you want to back up from a specific host or location (called a *backup set*). For example, you might decide you want to regularly create backup copies of everything in your user profile on your computer. Alternatively, you may want a copy of the entire disk or volume, including operating system files. Either of these options is viable and could be considered a different backup set.
- Full backups provide quick, easy restoration of all files from a backup set. Because all data is contained in a single backup, the process to restore will be faster than other backup options. However, full backups require more storage, especially if you're keeping more than one, and they take the longest amount of time to create.

Differential Backup

- Differential backups contain only copies of the data that's changed since the last full backup, thus lending themselves to more frequent use than full backups. If you decide to take a full backup once a month, it's a good idea to schedule a differential backup once a week. It's recommended to create a full backup regularly to keep the size of your differential backups under control (rather than taking one full backup of your backup set and then creating differential backups only from that point forward).
- Differential backups take up a lot of storage space, as the first backup contains copies of all the files modified since the last full backup, and then the next differential backup contains all of that data, *plus* all of the additional modified files between the first and second differential backups. Without a full backup at some point in the chain, differential backups will become exponentially larger as time goes on. Also, if one or more of the differential backups is incomplete, you won't be able to achieve a full recovery from the partial differential backup and the full backup of the data.

Incremental Backup

- Incremental backups create copies of any data that has changed since the last backup *of any kind*, whether it's a prior full, differential, or incremental. This type of backup takes the least amount of space and requires the least amount of time to create. If you create a full backup once a month and a differential backup once a week, you'd do well to create your incremental backups daily to ensure that any changes made to your data will be captured.
- Incremental backups are difficult in that it can be challenging to restore all your files across multiple backups, as each will have to be opened individually and specific files restored from the desired backup point. That means using all the available restore points to complete a full restoration, whereas a differential backup requires only the most recent differential and the most recent full backup to complete a full restoration.

Devising a Backup Schedule

Backups are of little value if not taken regularly. Your most critical data probably changes daily, so having a two-month-old backup of a document isn't helpful if it becomes corrupted or otherwise permanently unavailable. Therefore, it's essential to decide how often backups should be created. Your backup strategy will be highly individual and unique to your particular set of circumstances and requirements, though there are some best practices you can follow.

It's usually best to create full backups less frequently than differential or incremental backups, because of how much space and time they take up. Taking differential or incremental backups allows modified data to be backed up more frequently, and these backup types are less prone to failure due to time constraints than a full backup would be. As a general rule, taking a full backup of your primary systems or critical data once a month is a good place to start, and you can adjust your backup strategy as needed.

Depending on the backup software you choose, you may have access to some, all, or none of these specific scheduling options. Some software, for example, technically has all of this functionality and will allow you to restore data from your backups from different points in time, but the differential/incremental/full backup and restore options will not be displayed to you directly. Most operating systems come with an integrated backup solution (discussed later). Depending on your requirements, the built-in options might be sufficient. Otherwise, you can consider various paid solutions.

Onsite and Offsite Backups

Depending on the criticality of the data you choose to back up, keeping *offsite backups* in addition to *onsite backups* might be a good idea. Onsite backups are held in the same physical location as your original data, such as your home or office. Offsite backups are stored away from your primary location. Storing your backups in multiple locations provides data redundancy; if your onsite backups and original data are destroyed, an offsite backup may be your only option for recovery. Typically, an offsite backup will be stored offline, not connected to your network, and ideally in something like a fireproof safe. Alternatively, you might choose to use a cloud solution, but that comes with its own security risks.

Having an offsite backup may create additional administrative overhead. In most cases, you'll create your backup onsite, either by using a backup application or by copying the onsite backup in its entirety, and then remove that backup physically from the premises. This should be done on a regular schedule in the same way as onsite backups, as it affords you more options in the event of an emergency.

What to Back Up and What Storage to Use

Choosing exactly what you want to back up can be a challenge at first. Do you need to be able to recover an entire device from a previous checkpoint, or do you need only specific files? If you don't need to recover the whole operating system, it's best to determine which files and folders are critical or would be useful to recover. Consider how long you can be without the data before it really starts to impact you, your users, or your business. The time it takes to restore data from backup depends entirely on how much data you have to restore.

The other consideration when determining what to include in your backup strategy is what kind of storage, and how much, you have available to store your backups. You could choose to create backups of files on the device itself, but doing so isn't very useful if the device is lost, stolen, destroyed, or otherwise unavailable. A better option is to use an external hard drive, which can be obtained from any local computer store in any capacity that suits your needs. It's a cheap, easy option that allows you to create onsite backups and secondary backups that can be easily stored offsite. Finally, you could purchase dedicated storage for your backups in the form of a *network-attached storage (NAS)* device. An NAS connects to your network, has high storage capacity, and usually includes additional features like drive redundancy and automation. It offers greater performance and reliability than a stand-alone external hard drive but is often more expensive and requires some administration.

Whichever storage solution you choose, it should match your needs in terms of what you plan to back up. Full-disk backups of your devices will take up a lot of space, as internal storage for computers is typically up to or greater than 1TB. If you're planning to back up only critical personal files, you won't need as much storage for your backups. Chances are you'll start with an external hard drive and then upgrade your storage as your needs increase over time. The solution you choose will depend largely on your operating system, whether your endpoint is physical or virtual, and how much data you want to keep in your backup set.

If you want to maintain your own backups, most operating systems have built-in backup solutions, some of which are more fully featured than others.

#31: Using Windows Backup

To access the built-in backup solution for Windows, open **Windows Settings ▸ Update & Security ▸ Backup**. You'll need to connect a drive to the computer that will store your backup data. Once connected, click **Add a Drive** and select it, as shown in **Figure 9-1**. Note that in Windows 11, Windows Backup is found under **Windows Settings ▸ Accounts ▸ Windows backup**, and the settings are per user. You can use Windows Backup in Windows 11 to sync your files, applications, and preferences to OneDrive, but the option to back up a local drive to an external or network drive is no longer available.

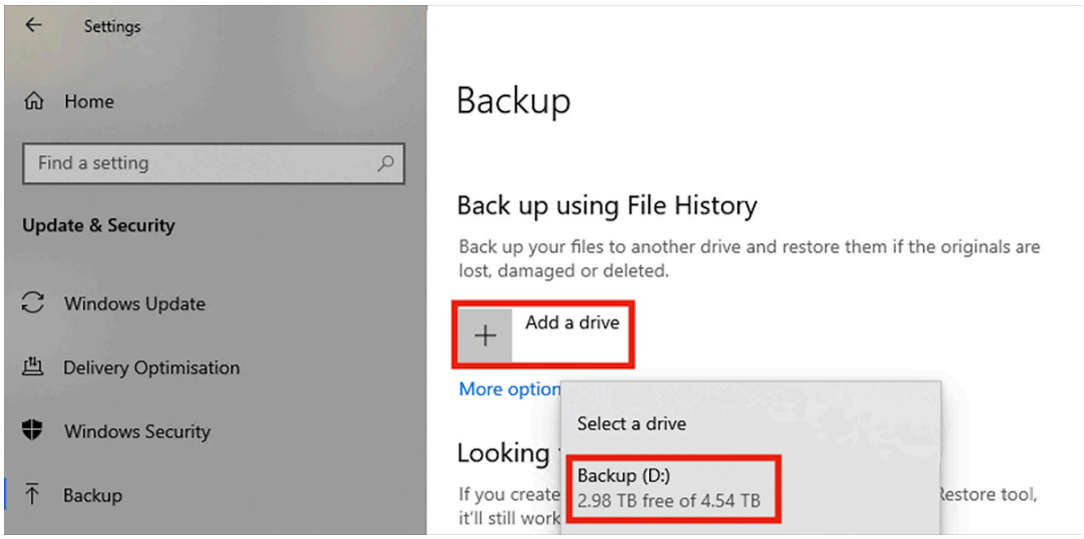

***Figure 9-1**: Windows backup to an external drive*

When you've selected the drive, the Automatically Back Up My Files option will be turned on. This will keep various copies of the more critical data in your user profile folder (*C:|Users|<username>*) such as your *Documents*, *Desktop*, and *Downloads* folders, as well as application settings from the *AppData* folder. You can choose which folders to include in your backup by clicking More Options.

Windows Backup is essentially a full backup plus differential backup strategy. It'll take a complete backup of your chosen files to start with and will then keep every new version or modification to any of those files on an hourly basis by default, forever (or until your backup drive space is exhausted). This allows you to view and restore your files from any point in the backup timeline. There are a few limitations to this, including being unable to save your backup to a network location and not being able to take full system images.

#32: Using Windows Backup and Restore

Windows Backup, discussed in the previous project, is great for backing up specific files and folders, but not for taking full system backups. Luckily, all versions of Windows since Windows 7 include *Back up and Restore*, which is useful for creating full system backups including system images that can completely restore a system (if it were to become corrupted by ransomware, for example). It's also capable of creating backups to an external or network drive, though it doesn't keep older versions of your files or file history.

In Windows 10, you can get to Back up and Restore by going to **Windows Settings ▸ Update & Security ▸ Back up ▸ Go to Back up and Restore (Windows 7)**. In Windows 11, Back up and Restore is found in **Control Panel ▸ System Security ▸ Back up and Restore (Windows 7)**. You'll be presented with the screen shown in **Figure 9-2**.

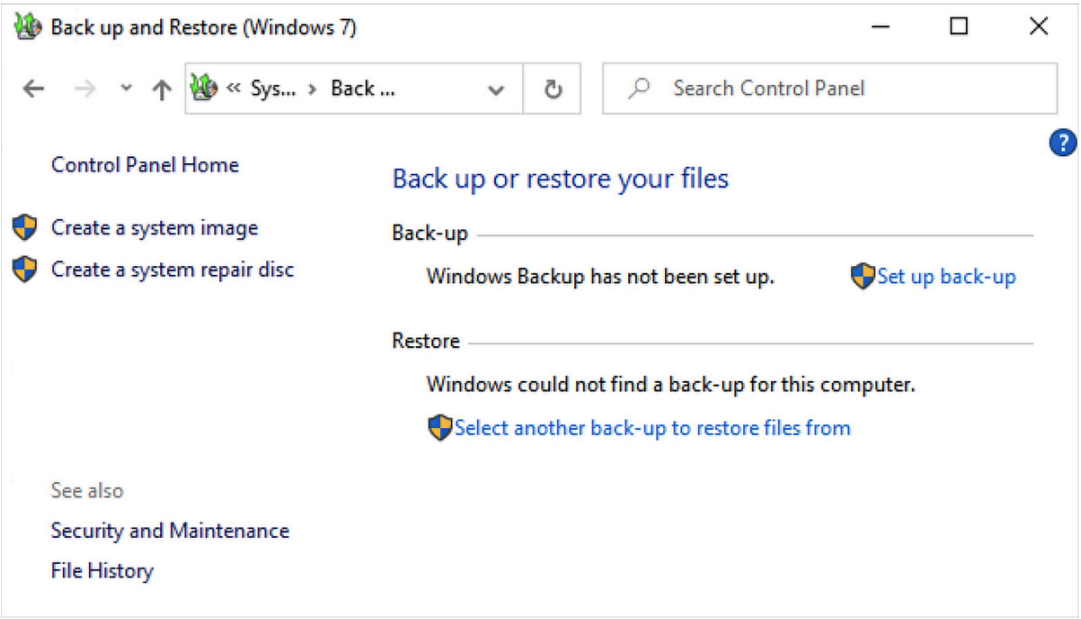

***Figure 9-2**: Windows Back up and Restore window*

On the left, you have options to create a *system image* or a *system repair disc*. In case your computer becomes inaccessible due to issues with the hardware or the operating system, you'll be able use either of these to restore the computer to its current, known-good configuration, with all of your files intact in their present state.

1. Click the **Set Up Back-up** button on the right to create a regular backup schedule of your personal and system files.
2. Attach an external drive and click **Refresh** to select it as a backup location, or specify a network drive or location by clicking **Save On a Network** and specifying the network location and the necessary username and password (if any); then click **Next**.

At this point, Windows will ask you what you want to include in your backup. By default, you can let Windows choose what to include, which will be your personal files and folders from *C:\Users\<username>*.

3. Select the **Let Me Choose** radio button and then click **Next**.
2. 4. Choose what to back up.

   a. Enable or disable backup of new users' files (assuming new user accounts will be created on this computer).
   b. Include and exclude any personal libraries such as *Documents* or *Pictures* and locations like your *Desktop* and *Downloads* folders.
   c. Select any folders from the drives on your computer.
   d. If you'd like to regularly back up your entire system, ensure that the checkbox to include a system image of your device is checked. When you're happy with the settings, click **Next**.
3. At this point, you can choose the schedule on which you want this backup to run, which can be daily, weekly, or monthly.
4. Click **Save Settings and Run Backup** to take the first full backup of your data.

You now have a regular backup of your folders to either an external or network drive on your Windows system.

#33: Using macOS Time Machine

Apple devices come with their own built-in solution for backing up your data called *Time Machine*, accessed via **System Settings ▸ Time Machine** (Figure 9-3).

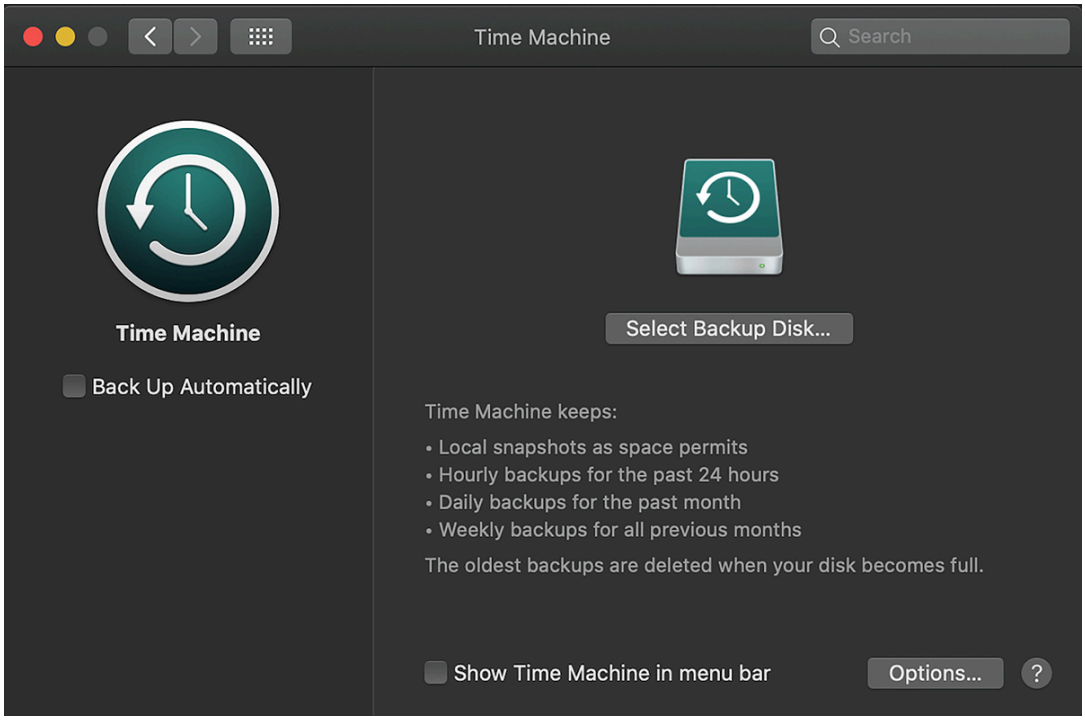

***Figure 9-3**: macOS Time Machine*

Time Machine can back up your data either to an external drive directly connected to your computer or to network-attached storage. This network storage can be in the form of an Apple Airport Time Capsule (designed explicitly for Time Machine backups), a drive connected to an Apple Airport Extreme base station, another Mac that has been shared as a Time Machine backup destination, or a dedicated NAS device that supports Time Machine over SMB. If you have any of these in your network already, I recommend using that solution. If you don't have one of these available, the simplest and cheapest solution to back up your Mac is to use an external drive rather than a network location. Typically, when you plug in a high-capacity drive to your Apple device, you'll be presented with a prompt asking if you'd like to use that drive for Time Machine backups. Alternatively, you can select a specific backup disk using the Select Backup Disk option shown in **Figure 9-3**.

Time Machine doesn't provide the option to schedule backups; it'll back up your data at predefined times. Time Machine keeps 24-hourly snapshots of your data, rolling daily backups for one month, and

rolling weekly backups for as long as there's space remaining on your backup drive. Once the space is exhausted, Time Machine will delete its oldest backup set to accommodate the more recent versions of your data. Check **Back Up Automatically** to allow Time Machine to do so.

The options for selecting the data you want to back up are somewhat limited. By default, Time Machine will back up your entire device, including system files, applications, accounts, preferences, emails, music, photos, movies, and documents. Click **Options** to exclude any of these using the dialog shown in **Figure 9-4**.

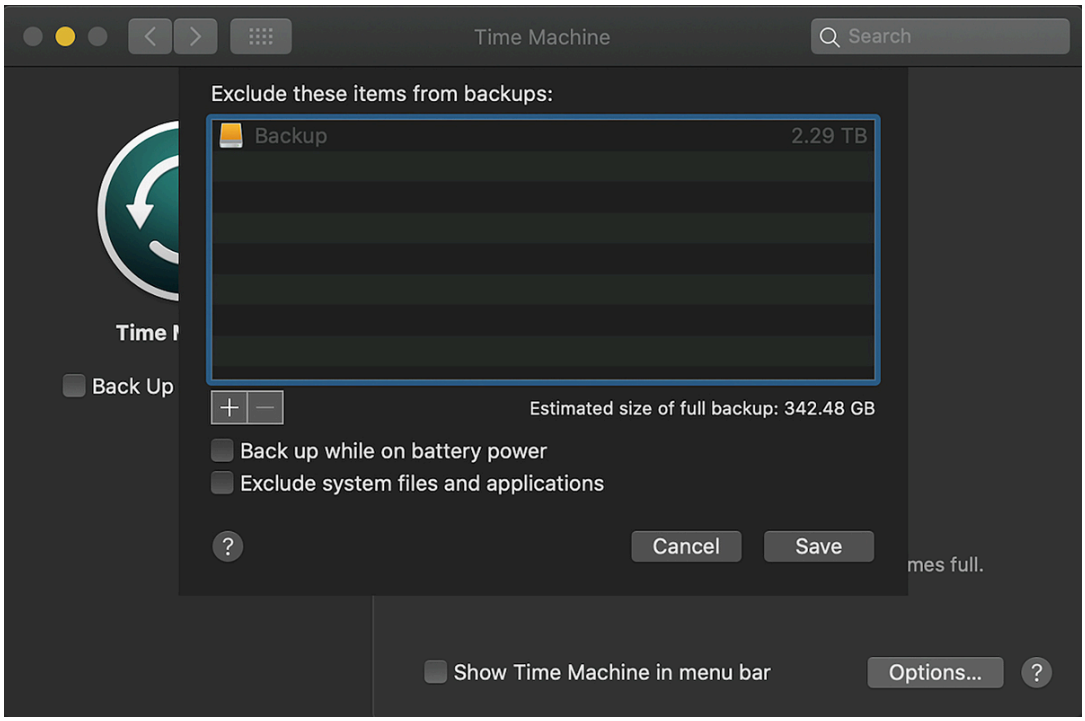

***Figure 9-4**: Time Machine backup options*

Overall, Time Machine is a robust solution for backing up and restoring your Apple endpoints.

#34: Using Linux duplicity

In Ubuntu, several utilities are available for creating backups of your files. The easiest to use is *duplicity*, a command line utility that creates full and incremental backup archives to local storage, external hard

drives, or network locations. Use the following command to install duplicity on your Ubuntu endpoints:

```
$ sudo apt install duplicity
```

Once the command completes, execute duplicity with the `-h` option to display its help file and confirm the installation was successful:

```
$ duplicity -h

Usage:
  duplicity [full|incremental] [options] source_dir target_url
  duplicity [restore] [options] source_url target_dir
  duplicity verify [options] source_url target_dir
  duplicity collection-status [options] target_url
  duplicity list-current-files [options] target_url
  duplicity cleanup [options] target_url
  duplicity remove-older-than time [options] target_url
  duplicity remove-all-but-n-full count [options] target_url
  duplicity remove-all-inc-of-but-n-full count [options] target_url
  duplicity replicate source_url target_url

--snip--
```

Read through the output to familiarize yourself with the available options and configurations. In the following sections, we'll discuss some of the most commonly used features.

Creating Local Backups with duplicity

The following example uses duplicity to create an initial full backup of a user's home directory, saving the output to the */tmp/* directory on the local system:

```
$ duplicity /home/user file:///tmp/
```

Last full backup date: none

GnuPG passphrase for decryption:

Retype passphrase for decryption to confirm:

----------[Backup Statistics]----------

StartTime 1634779305.32

EndTime 1634779305.94

ElapsedTime 0.62 (0.62 seconds)

SourceFiles 139

SourceFileSize 5793461 (5.53 MB)

NewFiles 139

NewFileSize 5793461 (5.53 MB)

DeletedFiles 0

ChangedFiles 0

ChangedFileSize 0 (0 bytes)

ChangedDeltaSize 0 (0 bytes)

DeltaEntries 139

RawDeltaSize 5465781 (5.21 MB)

TotalDestinationSizeChange 660694 (645 KB)

Errors 0

Note that the target directory (where the backup archive will be saved) must have the *file:///* prefix. The */tmp/* directory is a holding location for the backup; you should either move the backup elsewhere once completed or save the backup somewhere else. The first time this command is run, duplicity will take a full backup of the source files or folders. Subsequent execution of the same command will create incremental backups of the source data. The command outputs statistics, as shown in the listing, including the start and end time of the backup operation, how many files were included, and the total size of the backup archive. Backups created with duplicity must be protected with a passphrase.

To create another full backup, specify the `full` option, like so:

```
$ duplicity full /home/ user file:///tmp/
```

Doing so will force duplicity to create a full, rather than an incremental, backup of the data.

Creating Network Backups with duplicity

Saving backups to a network location is preferable to saving them to a local folder for several reasons. Saving backups locally is risky, because if you lose access to the endpoint or it becomes otherwise unavailable, you have no backups to restore from in other locations. Also, if an adversary gains access to the system, they've gained access

to your (encrypted) backups as well. Therefore, it's safer to save to a remote location like a fileserver. This can be achieved using the rsync functionality built in to duplicity. The following command assumes you've followed the instructions in **Chapter 1** to create SSH keys and use SSH key authentication instead of password authentication. If this is not the case, go back and do so now. SSH key authentication requires the use of a public/private key pair that's shared between the local and remote endpoints, enabling them to perform cryptographically secure communication that offers greater security than the use of password or passphrase authentication.

```
$ duplicity /home/ user
rsync:// user@server_ip//path/to/folder/
```

Once you've decided which files and folders you want to back up, and the backup location, schedule duplicity to create regular backups of your files using Crontab, a built-in Linux job discussed in detail in **Project 27** on **page 122**:

```
$ sudo crontab -e

--snip--

# m h  dom mon dow   command

0 0 * * 1 duplicity /home/ user
rsync:// user@server_ip//path/to/folder/

0 2 1 * * duplicity full /home/ user
rsync:// user@server_ip//path/to/folder/
```

The `-e` option of the Crontab application indicates that you will edit the cron file and the scheduled jobs maintained by cron. The commands shown in the Crontab in this example schedule duplicity to run at midnight every day, create an incremental backup, and force a full backup to be created on the first day of every month at 2 AM.

Restoring duplicity Backups

Use the command line to restore backups created with duplicity:

```
$ duplicity restore file:///tmp/
/home/ user/backup_folder_name/
```

Entering the `restore` command with a source and target path restores all files from the backup set to the specified location.

There are various options to restore specific files and folders from a backup set if required. Here's an example:

```
$ duplicity -t 3D --file-to-restore
/home/user/Documents/test.txt \

    file:///tmp/ /home/user/Documents/restored_file
```

In this command, we invoke duplicity, tell it to restore the version of the *test.txt* file (specified immediately after the `--file-to-restore` argument) from three days ago with the `-t 3D` parameters from the backup we created in the */tmp/* folder on the local system, and we then save the resulting file to the */home/user/Documents/* folder. For more information on restoring files and the options available, review duplicity's man page.

Additional duplicity Considerations

The duplicity utility has several other powerful options. You might want to exclude certain files or folders from your backups; for example, you'll often want to exclude system folders when creating backups of user data. Use the `--exclude` argument to exclude files and folders:

```
$ duplicity --exclude /proc --exclude /mnt / file:///tmp/
```

Once your backups have run to completion, use the `verify` parameter and swap the source and target locations from your original backup command to confirm they were created successfully:

```
$ duplicity verify file:///tmp/ /home/user/
```

Local and Remote metadata are synchronized, no sync needed.

`--snip--`

Verify complete: 325 files compared, 0 differences found.

If the output reveals no errors, your backups were successful.

There will be times when you want to delete older backups, either because they're no longer needed or to free up space for newer backups. First, review the available backups in your backup set using the `collection-status` parameter:

```
$ duplicity collection-status file:///tmp/
```

`--snip--`

Collection Status

Connecting with backend: BackendWrapper

Archive dir:
/home/user/.cache/duplicity/c2731c0788339744944161fd8afb74dd

Found 1 secondary backup chain.

Secondary chain 1 of 1:

Chain start time: Wed Oct 20 19:53:09 2022

Chain end time: Wed Oct 29 20:11:39 2022

Number of contained backup sets: 2

Total number of contained volumes: 2

Type of backup set:	Time:	Num volumes:
Full	Wed Oct 20 19:53:09 2022	1
Incremental	Wed Oct 29 20:11:39 2022	1

Found primary backup chain with matching signature chain:

Chain start time: Wed Oct 20 20:11:53 2022

Chain end time: Wed Oct 20 20:11:53 2022

Number of contained backup sets: 1

Total number of contained volumes: 1

Type of backup set:	Time:	Num volumes:
Full	Wed Oct 20 20:11:53 2022	1

No orphaned or incomplete backup sets found.

Once you know how many backups are in your backup set and how old they are, you can delete older backups based on age:

```
$ duplicity remove-older-than 3D file:///tmp/
```

The `3D` means older than three days.

You also can remove all but the desired number of full backups:

```
$ duplicity remove-all-but-n-full 1 file:///tmp/
```

Here the `1` indicates to duplicity that it should delete all but the most recent full backup from the backup set. Read duplicity's man page to become familiar with the available options for creating, restoring, or deleting backups.

Cloud Backup Solutions

Although cloud services like Google Drive and Dropbox are not true backups, they're capable of keeping a secondary copy of your local data in the cloud (akin to an offsite backup), keeping tertiary copies on your other systems, and maintaining a version history of each of your files—and they do all of this consistently and regularly. Most of these services come with some level of free storage as well, so you can try them out and upgrade to a paid plan if you decide they work for you.

Google Drive and Dropbox are generally designed for file sharing and online collaboration rather than for backing up data. Using a service geared explicitly toward backing up data, although not free, will generally provide more features, granular control, and storage at a lower cost. Backblaze and Carbonite are two reliable cloud backup services that encrypt your data locally and automatically back it up using a client application on your computer. Backblaze will back up your files, and Carbonite is capable of backing up your entire computer. In general, look for services that encrypt your data both at rest and in tran-

sit. Carbonite and Backblaze are currently available for Mac, Windows, and Linux.

Backblaze

Backblaze is a great option if you want a set-and-forget backup solution. Once downloaded and installed, it immediately begins backing up your files to the Backblaze cloud servers and will continuously do so unless set to do otherwise. Virtual machine files and folders are the only data automatically excluded, although you can remove those from the exclusions list. You'll be able to back up not only your computer's internal hard drive but any attached external drives as well. Backblaze maintains all versions of your files for the previous 30 days, which can be extended to one year or more with an additional fee. You can restore your files from the application (shown in **Figure 9-5**), the web UI, the mobile app, or by having Backblaze physically mail your files to you on a USB drive.

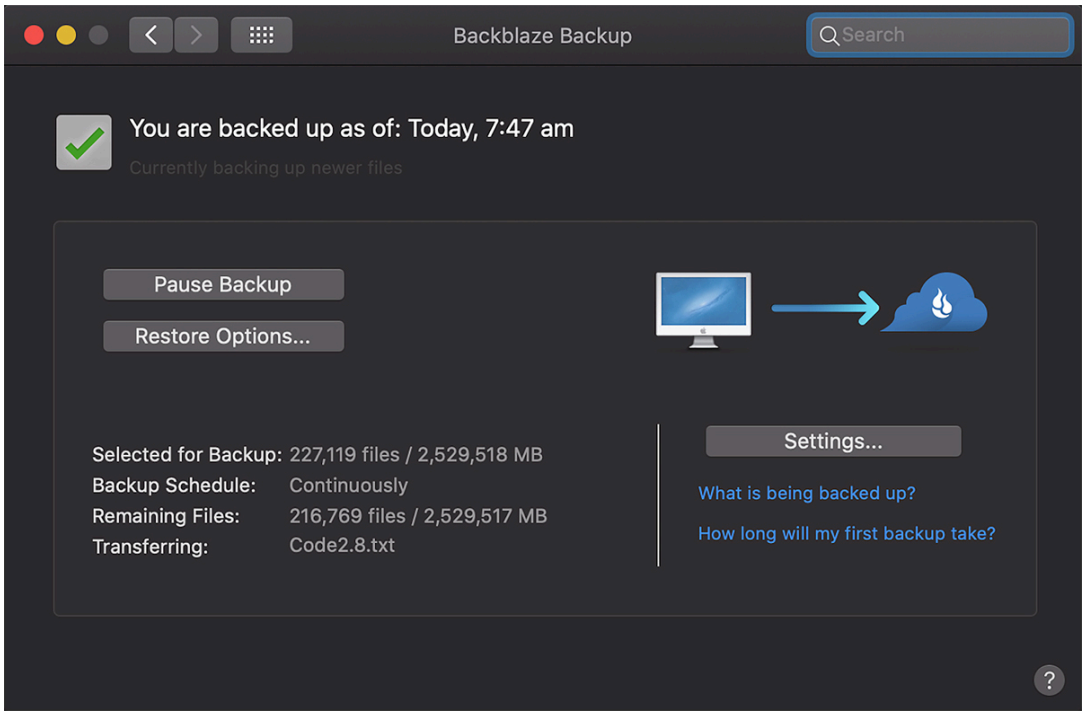

***Figure 9-5**: Backblaze GUI*

The security features Backblaze offers are worth considering. Your files are first encrypted by the application, are transferred to the cloud using SSL (encrypted transmission), and are then stored, encrypted, on the Backblaze servers. Even better, you're able to configure your own decryption key, so Backblaze itself is unable to decrypt your data, which adds another layer of complexity for any adversary who might gain access to your encrypted data. You also can add two-factor authentication so that, in addition to your password, your decryption key, and your email address, anyone who wants to access your data, including yourself, requires a one-time password. With all of those features combined, any third party trying to access your data will have a lot of layers of defense to contend with. Additionally, Backblaze is one of the cheapest cloud backup solutions available.

Carbonite

If you need a backup provider capable of restoring your entire computer in the event of catastrophic failure, Carbonite is one possible solution. In extreme cases, like the event of ransomware, having the capability to restore your computer all the way down to the operating system settings and configuration in addition to your critical files is beneficial because you might find the entire operating system becomes corrupted or otherwise unusable. Carbonite has several plans at tiered price points so you can choose the level of cover for your specific needs. As with Backblaze, Carbonite locally encrypts all the data it backs up, which is sent to the cloud over SSL and encrypted on the Carbonite servers. Depending on the plan you choose, it's capable of backing up external hard drives and provides unlimited cloud storage (in some cases). Carbonite will also keep your backed-up files for an unlimited amount of time, and it won't delete file versions older than 30 days.

Once you've downloaded and installed the application (shown in **Figure 9-6**), it'll start uploading the first backup of your data automatically. You can tell it which files to include or exclude.

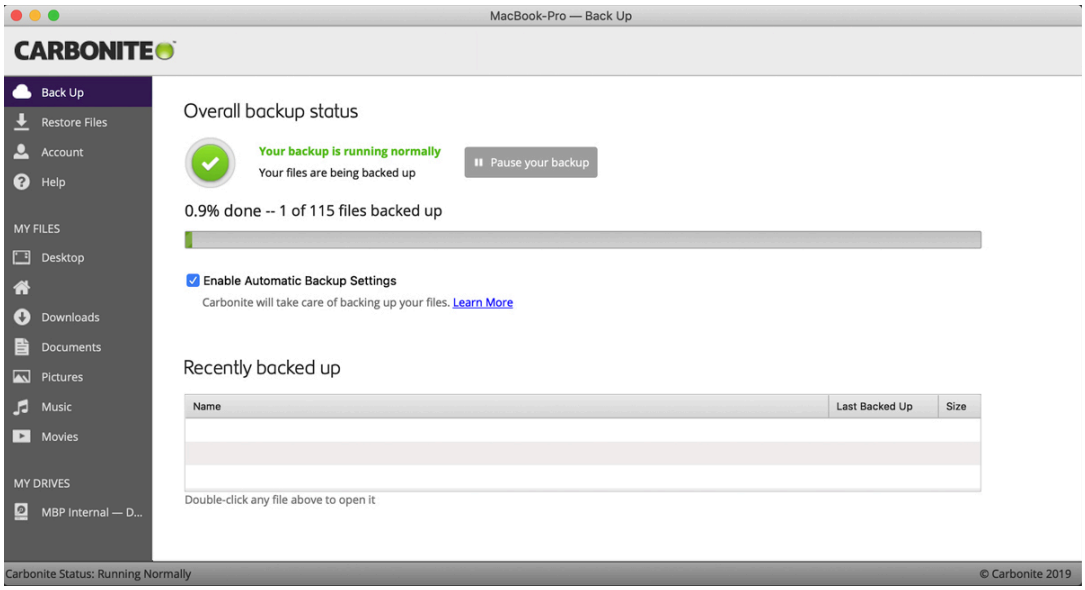

***Figure 9-6**: Carbonite GUI*

Carbonite, like Backblaze, will run in the background, continuously backing up your data. When you want to restore a file from your backup, you can do so through the application. There is no web UI or mobile app available.

Virtual Machine Snapshots

Virtual machines provide a lot of benefits not seen in physical computers. They're able to share hardware (processors and RAM), are fast to start up or reboot, and can be created with just the right amount of resources for their specific purpose. One of the best things about them is the capability to create *snapshots*.

A virtual machine snapshot is a copy of that virtual machine at a given point in time. A snapshot generally includes all information related to a virtual machine at the time the snapshot is taken, including its power state (on, off, or paused/suspended), the contents of its virtual memory, and the contents of its virtual hard disk.

Whenever you're making significant changes to a virtual machine, it's wise to create a snapshot of that machine beforehand to protect your-

self from the eventuality that your modifications will break your virtual machine. Having a snapshot just before the changes that rendered your virtual machine unusable allows you to revert to the known-good configuration as if the changes were never made in the first place. It's like a real-life undo button.

All major virtual machine software (such as VMware, VirtualBox, and Hyper-V) are capable of creating snapshots of the virtual machines they manage. **Figure 9-7** shows an example of VMware's Snapshot Manager.

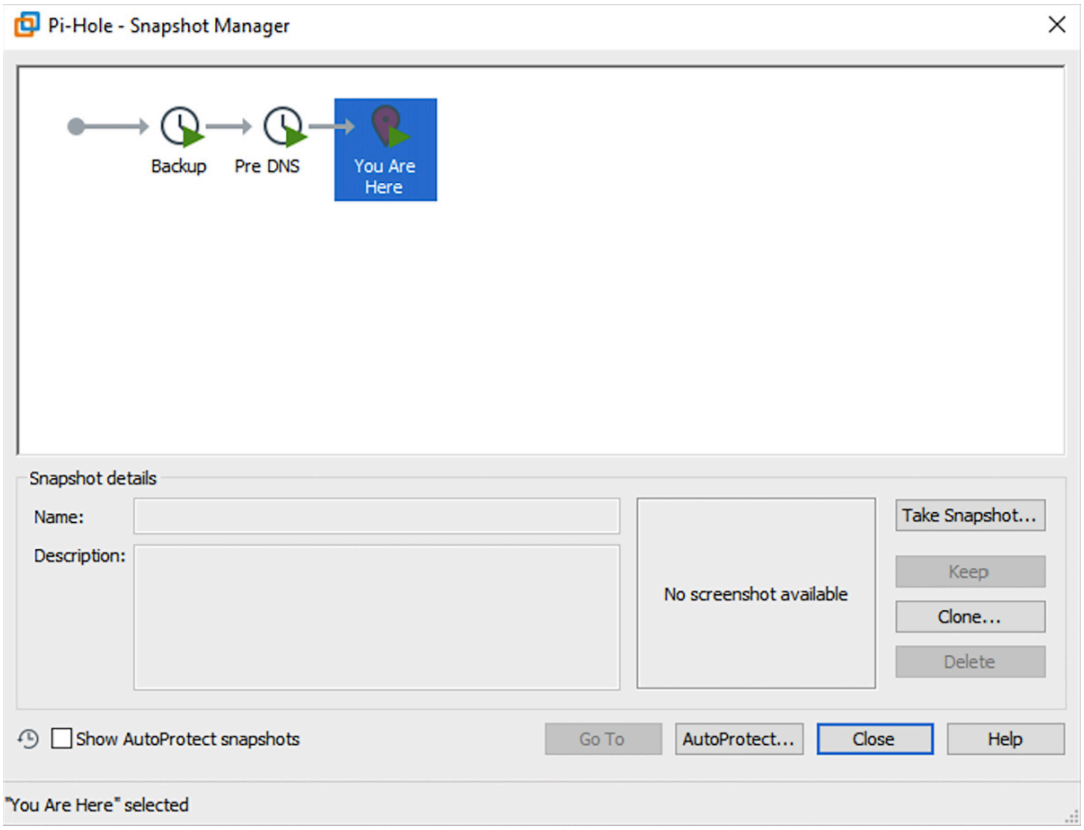

***Figure 9-7**: VMware Snapshot Manager*

From within Snapshot Manager, you can create a new snapshot with the Take Snapshot button, go to a specific snapshot (that is, revert the virtual machine to the point in time a given snapshot was taken), delete or clone a snapshot, or enable AutoProtect, a feature that creates snapshots on a predefined schedule, allowing you to revert to a

snapshot from multiple points in the past. Snapshots are available in most hypervisors, but the settings and options might differ slightly.

While snapshots aren't a backup in the literal sense, they are useful for restoring a working virtual machine configuration. You shouldn't use snapshots as your only means of backing up, but including them as part of your strategy is often handy. A sensible solution would be to include all the virtual machine files in your regular backup strategy.

Testing and Restoring Backups

Once you've created your backup strategy and your most important data is being regularly backed up onsite and offsite, it's important to test these backups at regular intervals. If you suffer a loss of data and try to restore it from your backup only to find out it has become corrupt, your backup strategy is providing no value.

To restore your files on Windows, open the **Backup and Restore** menu. Click **Restore My Files** or **Restore All Users' Files**. You can look through the contents of your backups by using the Browse for Files or Browse for Folders options. You can also search the contents of your backup via this menu.

To restore your files on a Mac using Time Machine, browse to the folder from which you want to restore files, such as your Documents or Downloads folder. Open Time Machine and then use the arrows and timeline to browse the available local snapshots and backups. Select one or more of the items you want to restore and then click **Restore**. This can include files, folders, or your entire disk. Restored items will return to their original location on your computer.

To restore files using Carbonite, Backblaze, or any other solution, open the web portal or the application GUI. Then, locate the files or folders you want to restore and follow the instructions provided.

When your first full backup of any system has been created, test a few files or folders at random to restore. It may be worth finding some larger files to include as part of this test, as the larger the file, the higher chance there is of the backup failing partway through. If you restore a sample of data and all seems well, set a reminder to do the same again in a week and then a month following. If all of your test restorations go smoothly, you can choose how often you want to test your backups from that point forward. Somewhere between one and six months is prudent.

With that done, your backup strategy should be complete and robust enough to recover from pretty much any data loss event or disaster.

Summary

The solutions discussed in this chapter will be adequate for most situations, but they aren't guaranteed to fit your needs. When looking for a backup solution, ensure that the one you choose is right for your operating system, allows for the creation of backups containing the data you want to back up (and nothing else), and is capable of the type of backups you want to create. It should also be capable of automatically creating backups on a schedule or creating a backup of your data constantly. Finally, make sure that, once backed up, you're able to restore your data within a reasonable timeframe and high confidence in the integrity of the restored data.