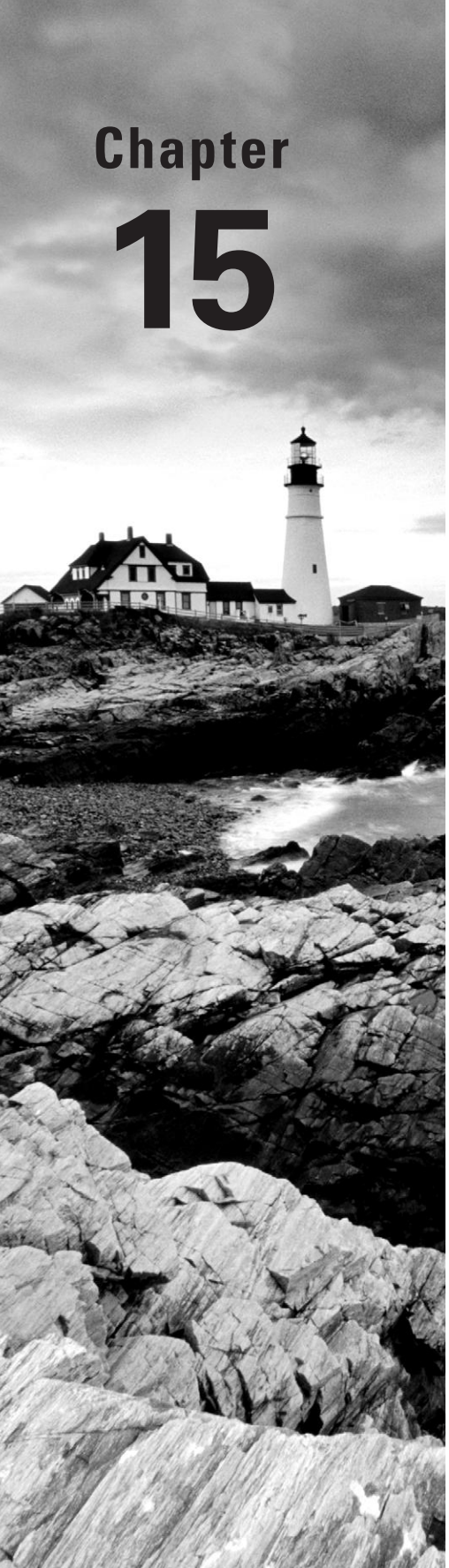
Chapter

15



Cloud Storage

TOPICS COVERED IN THIS CHAPTER:

- ✓ Defining the cloud
- ✓ The cloud computing model
- ✓ Public cloud
- ✓ Private cloud
- ✓ Hybrid cloud
- ✓ Cloud storage
- ✓ Eventual consistency
- ✓ Data durability

This chapter delves into the increasingly popular and important topic of the cloud. The chapter starts by defining the cloud at a high level and then gets into more detail. Because the concepts of cloud computing can represent a significant mindset change for many people, you'll get hands-on with some of the most popular cloud services available, including spinning up virtual machines (VMs) as well as object stores in the public cloud.

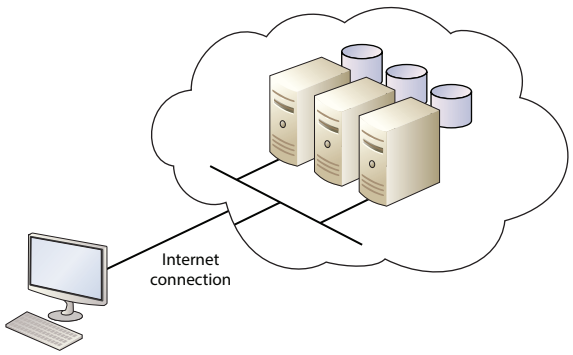
This chapter presents the three major cloud models: public cloud, private cloud, and hybrid cloud. After learning about cloud computing, you'll focus on cloud storage—its important concepts and theory as well as some hands-on activities as you create storage in the cloud and upload objects. You'll explore public cloud storage, private cloud storage, and hybrid cloud storage and some potential use cases for each.

The Cloud Overview

There isn't really any formal definition of what the *cloud* is, but don't let that deter you. The cloud is real, and it's going to get bigger in the future. Thankfully, it's pretty simple.

The cloud, or *cloud computing*, is a computing model. A major feature of this model is that IT resources are accessed remotely via the Internet, rather than locally over a local area network (LAN). This is shown in Figure 15.1.

FIGURE 15.1 High-level cloud computing model



In many ways, cloud computing is an extreme form of virtualization, where you as a consumer don't care about the underlying hardware or software. You no longer have to think about hardware maintenance, hardware upgrades, tech refresh, security patching, antivirus, support staff, or any of the typical things that you have to consider if you own the technology. You literally consume the service and don't care about the underlying specifics. It couldn't be simpler. A major goal of cloud computing is exactly that: make consumption of IT resources as simple as possible.

The term *cloud* probably came from the popular cloud image used to represent the Internet in IT and network infrastructure diagrams. One reason for representing the Internet as a cloud is that for most purposes you don't need to understand the complexities of the Internet and Internet routing; you just need to know that it's a network that reliably delivers data. The same goes for cloud computing: you pay for and receive a service and don't need to care about the details. All you care about is that it works.

Companies that provide cloud services are usually referred to as *service providers,* or *cloud providers*. They own the technology behind the service and are responsible for making sure that it all works seamlessly for you as a customer. They're also the people that you pay, usually on a monthly basis, for the privilege of using their services. For these reasons, we often refer to cloud solutions as *fully managed services*. Some of the major public cloud providers include companies such as Amazon, Microsoft, Google, and Rackspace.

One of the major advantages the cloud brings is cost. Cloud services are almost always set up so that you pay monthly based on what you have consumed that month. This operating expenditure (op-ex) model means that the financial road into the cloud is made easy. There's no requirement for large up-front capital expenditure (cap-ex) costs as there usually is in noncloud models. In noncloud models, if you want 10 servers with 64 GB RAM each and a total of 6 TB of storage, you have to lay down the cash to buy these up front (cap-ex). This isn't the case with cloud computing. In the cloud model, you can literally spin up the 10 virtual servers with the required RAM and storage (using the cloud provider's self-service tools), and then you start paying on a monthly basis. After a month or two, if you realize that you really need only three of the servers, you turn off the seven you don't want and immediately stop paying for them. Then, at a later date, if you need another 20 servers, you can spin up another 20 servers and have them instantly available. Again, there is no up-front cap-ex cost and no long lead-time on delivery of the servers.

This ability to dial up or dial down your consumption on the fly is known as *elastic computing*. You can use as much or as little of the cloud as you want, and you can change it on a daily basis. There is no need to over-spec on day one, just in case you might need more capacity in the future, only to find that that future requirement never materializes. The cloud model is totally different from more-traditional approaches to IT. It is way more flexible.

Another advantage of the cloud is that services tend to be self-service. Cloud users can order their own configurations via a simple web portal, and after they've completed their

order, the service is provisioned and almost always instantly available. For example, a cloud service will probably provide a web portal for ordering a virtual machine (VM) to your exact specifications. On the portal, you enter all of the following for your VM:

- Name
- Number of CPUs
- Amount of memory
- Number and speed of network cards
- Number and size of disks

After you've ordered your VM from the self-service portal, you click Create, and the VM is created and immediately made available to you. It is a lot like buying a new book for your Amazon Kindle device—you select the book you want, add it to your shopping cart, proceed with payment, and the book is immediately delivered to your device.

Now let's talk briefly about the three major cloud models:

- Public
- Private
- Hybrid

Public Cloud

The *public cloud* is what most people think of when they think about the cloud. In many ways, it's the purest form of the cloud.

The public cloud is usually accessed over the Internet and provides the highest levels of abstraction. By this, I mean that it's opaque: customers have no visibility of the technologies holding up the service, the server hardware, and the networking and storage technologies involved.

One thing we all need to understand and accept about the public cloud is that it's *multitenant*, meaning that all the underlying infrastructure is shared by multiple customers. You definitely don't get a physical server or physical storage array to yourself! If you place your services or data in the public cloud, it will be sitting on the same shared infrastructure as potentially thousands of other customers' services and data. This is true multitenancy!

Also, as a public cloud customer, you have very little idea of where your data is actually held. Sure, you will probably know the region that it's held in, such as North America, the European Union (EU), Asia Pacific (APAC), and so on. But you won't know exactly where, and you definitely can't turn up at the front door of your cloud provider's data centers and get inside to have a look around.

All of this is important in providing a lot of the features we expect from cloud services, such as the following:

- Low cost
- Elasticity

- Massive scalability
- Self-service
- Accessibility

However, the uber-shared nature of public cloud services can be a challenge for some use cases and organizations. You might work for an organization that has strict security policies governing whether services can be hosted in the public cloud. In these situations, you may not be allowed to use public cloud services.

Services hosted in the public cloud are accessible from anywhere in the world that has an Internet connection, making public cloud–based services among the most accessible IT services in the world. You can access them at any time, from any device, in any part of the world, so long as you have an Internet connection.

This leads us to a vital point on public cloud: security! You need to be security savvy when working with public cloud services. Make sure you encrypt your data, and make sure that you, not your cloud provider, hold the encryption keys. And make sure that you mark as "public" only any data that you want to be publically viewable.

The most popular public cloud computing service is probably Amazon Elastic Compute Cloud (EC2), which is part of Amazon Web Services (AWS). Other popular public cloud computing services include Windows Azure and Rackspace Cloud Servers.

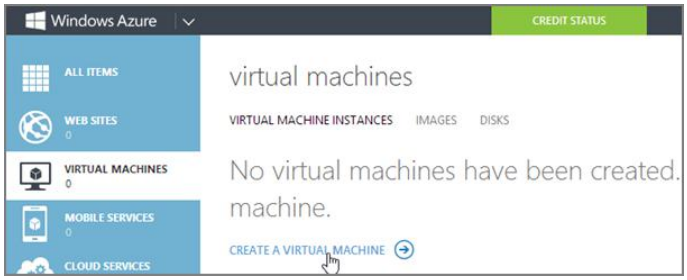
One of the first basic things you need to do in the cloud is create a virtual machine. Exercise 15.1 walks you through how to create a Windows virtual machine with Windows Azure, and Exercise 15.2 shows you how to create a Linux virtual machine with AWS.

EXERCISE 15.1

Creating a New Virtual Machine in the Cloud with Windows Azure

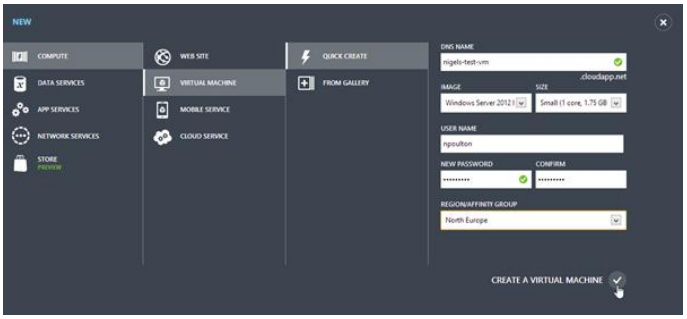
In this exercise, you'll walk through the process of creating and logging on to a Windows VM created in the Windows Azure public cloud. This exercise assumes you have a Windows Azure account.

1. In the Windows Azure web user interface, select Virtual Machines and then click the Create A Virtual Machine link.

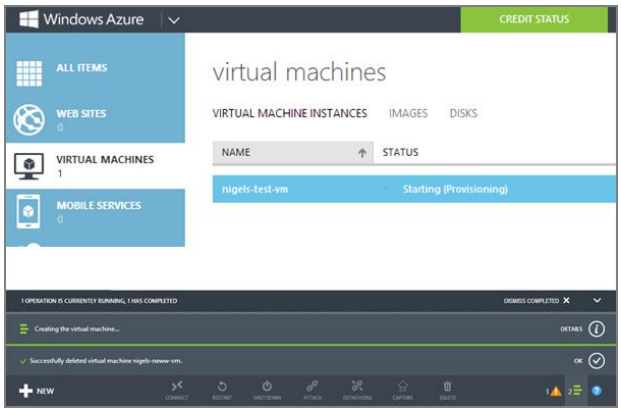


2. The properties sheet for creating a new VM appears and requires you to provide the following information for your VM:
   - DNS name
   - Operating system (referred to as *image*)
   - Size of the VM (specifies number of CPUs and amount of memory)
   - Username
   - Password
   - Region to store your VM

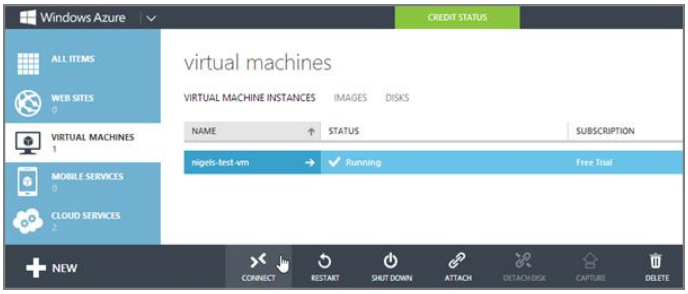
   Enter the information required to create your new VM and click the check mark next to Create A Virtual Machine. I am selecting a configuration for a new VM called `nigels-test-vm` that will run Windows Server 2012 on a small VM image with a single core and 1.75 GB of RAM, with a username and password, and region of Northern Europe. You should use whatever settings are appropriate for your scenario.



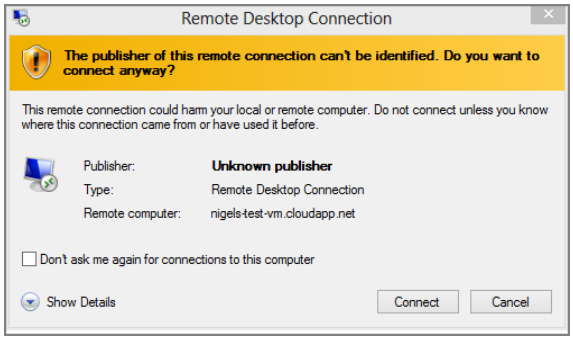
   After you have clicked Create A Virtual Machine, Windows Azure creates your new VM. The process should take only a minute or two.



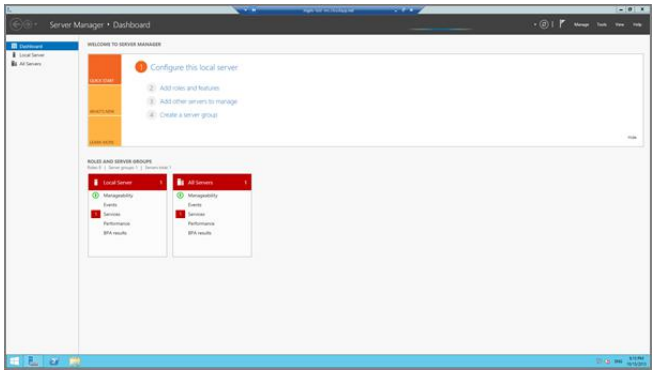
3. Now that your VM is created and running, you can log on. To log on to a Windows Azure VM, highlight the VM in the list of available VMs and click Connect.



The Windows Azure portal now downloads a Remote Desktop Protocol (RDP) app that will allow you to log on to your newly created VM. Opening the RDP app brings up the Remote Desktop Connection dialog box.



4. Click the Connect button in the Remote Desktop Connection dialog box. Then enter your username and password in the Windows Security dialog box that appears. This logs you on to your new VM.

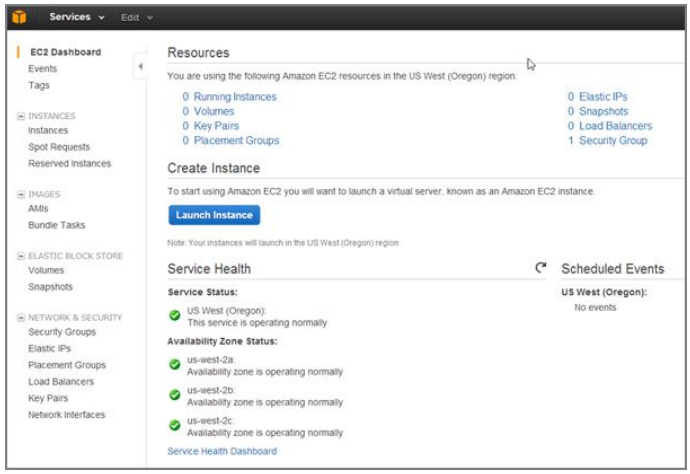


EXERCISE 15.2

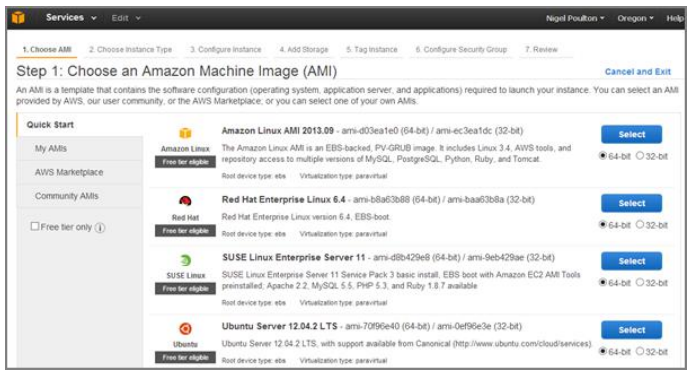
Creating a New Virtual Machine in the Cloud with Amazon Web Services

In this exercise, you'll walk through the process of creating a new Linux VM in the EC2 component of AWS. This exercise assumes you have an Amazon Web Services account.

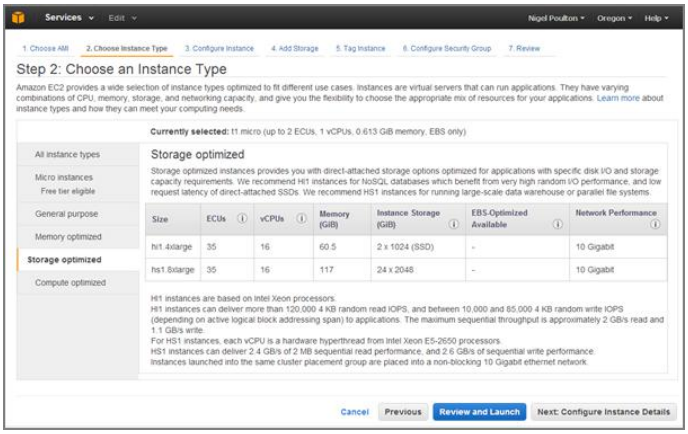
1. Once logged on to the AWS Management Console, the first thing to do is navigate to the EC2 Virtual Servers in the Cloud dashboard.
2. When in the EC2 dashboard, you need to create a new VM. In EC2, a VM is called an *instance*. In the EC2 dashboard web user interface, select the option Launch Instance.



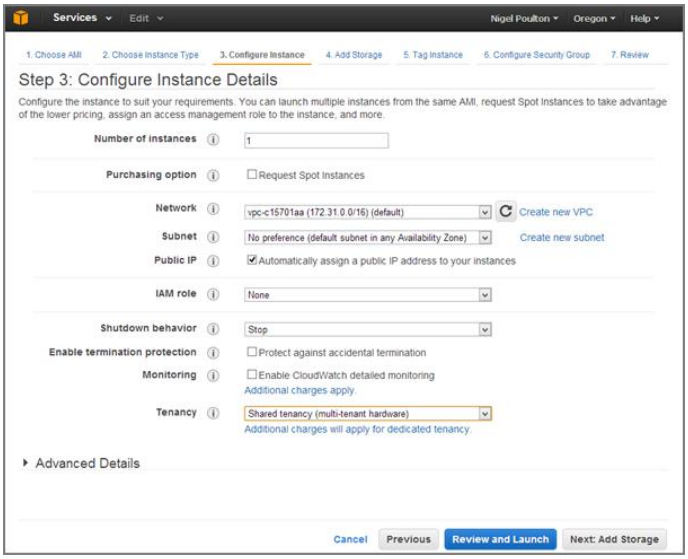
3. From the resulting screen, select the type of VM you would like to create. Options include most of the popular operating systems as well as choices of 32-bit or 64-bit. Some options are shown in the following screenshot. Select your instance type according to your operating system details.



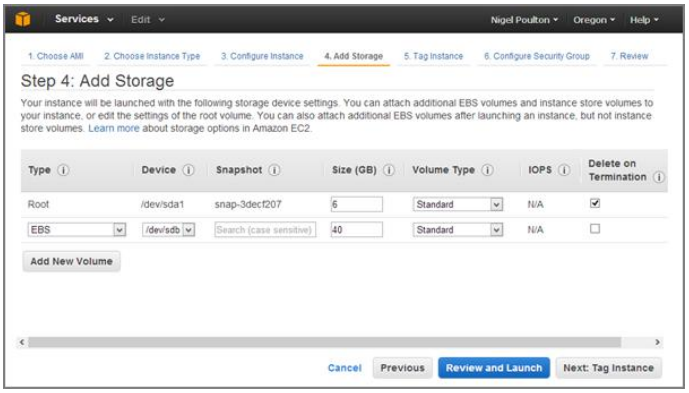
4. You are now prompted for details related to the virtual machine. EC2 gives you several options that are optimized for certain characteristics, such as Memory Optimized, Compute Optimized, and Storage Optimized. Choose the type of VM you wish to create and click Next: Configure Instance Details. The following screenshot shows a Storage Optimized VM configuration.



5. Click through the remaining screens to configure your new VM. This includes configuring the networking and shutdown details, some of which are shown here.



6. You can also add storage devices to your VM.



7. Once all of the information has been provided, click Review And Launch and your new VM will be created.

Private Cloud

The *private cloud* can have either of the following two designs:

- Your own internal cloud, with on-premise equipment that you still own and manage, but use to offer cloud-like services to your internal customers.
- Services hosted by a third party, off premise, but offering customers their own dedicated infrastructure (no multitenancy).

The term *on premise* refers to hosting IT equipment in your own data center facilities (your own premises). The term *off premise* refers to hosting IT in somebody else's data center facilities. Sometimes the terms are shortened to *off prem* and *on prem,* respectively.

In the first model, where you own the on-premise equipment and provide cloud-like services to your internal customers, these services include things such as web portals that allow users to self-provision, as well as monthly itemized billing services. This is a common approach in large organizations, where the internal IT department is under pressure from the public cloud and needs to provide cloud-like services internally in order to discourage users from opting for public cloud services rather than services offered by the internal IT department.

In the second model, third parties provide customers with dedicated off-premise equipment. This provides some of the advantages of cloud services without some of the risks

perceived around the public cloud. For example, an organization may have a desire to no longer own and manage all of its own technology infrastructure, but may also be averse to putting all of its data and services out to the public cloud on a massively shared platform. A private cloud option, in which another company takes care of all infrastructure management but provides a dedicated platform, may be a good solution. These options often allow customers to visit data center facilities and view the equipment that they are paying to use.

Hybrid Cloud

The *hybrid cloud*, as its name suggests, is a combination of the public cloud and private cloud. Many organizations aren't keen to throw all their data and services into the public cloud on day one. Instead, they prefer to start out by putting some of their less-important apps and services into the public cloud while keeping their mission-critical and sensitive apps in-house. This mixture of the public and private cloud is sometimes referred to as the *hybrid cloud*.

Hybrid clouds also allow customers to burst into the cloud as the business requires extra computing resources. For example, say your business requires a lot more computing capability at one particular time of the year—you put on an event once a year that sees a massive spike in visits to your website. Utilizing the public cloud allows you to increase your computing power once a year and then turn it off again (and stop paying for it) when it's no longer needed.

Storage and the Cloud

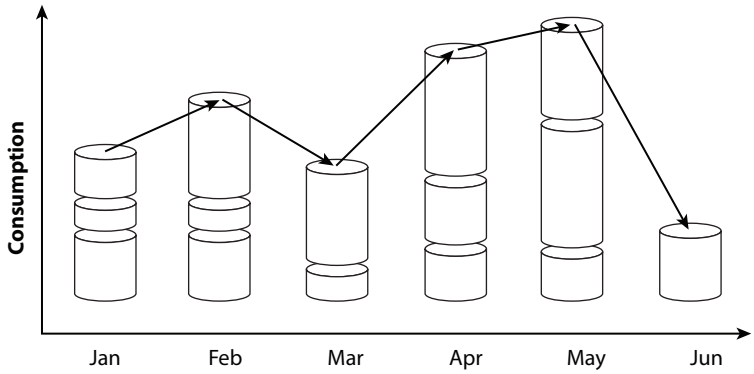
You've examined cloud computing. Now let's look more specifically at cloud storage.

Storage as a service is sometimes written as *SaaS*.

Cloud storage is accessed remotely over the Internet. You pay for it monthly using a consumption model where you only pay for what you're using.

The combination of elastic storage and consumption-based billing is shown in Figure 15.2.

FIGURE 15.2 Elasticity and consumption billing



Cloud storage has most of the features and benefits of cloud computing, including the following:

- Simplicity
- Elasticity
- Massive scalability
- Accessibility
- Fast self-service provisioning
- Built-in protection
- No cap-ex cost

Cloud storage doesn't suit every use case. Cloud storage services have some potential drawbacks that need considering. These include low performance when compared to non-cloud-based storage solutions. For example, accessing storage over an Internet connection is never going to be as fast as accessing storage over a short, low-latency Fibre Channel SAN that exists in close proximity to your servers within the confines of your own data center. Similarly, you won't want to put any latency-sensitive applications on public cloud storage. So that means no SQL-based relational databases on cloud storage.

Most cloud storage solutions are object storage–based. Although Chapter 7, "Files, NAS, and Objects," covers object storage (object stores) in detail, I'll remind you here that object storage is not designed for high-performance or structured data (databases), and it isn't ideal for data that changes a lot. However, it is good for streaming/sequential read workloads, making it ideal for storage and retrieval of rich media and other Web 2.0 types of content such as photo images, videos, audio, and other documents. Hence the reason that many of the popular social media sites use object storage systems to host their uploaded content such as images and videos, as these are uploaded once (written once), played back (read) multiple times, and almost never changed after being uploaded.

Make sure that you know whether your cloud provider charges you based on a base-2 or base-10 usage number. Base 10 thinks that 1 MB = 1,000 KB, whereas base 2 thinks that 1 MB = 1,024 KB. While it might seem a small difference with such small numbers, when you start talking about multiple terabytes (TB), the difference can be significant.

Data Durability

The concept of *data durability* is key to cloud storage. Data durability refers to the ability of an object stored in the cloud to survive failures within the cloud. You can think of this as being a little like RAID for the cloud.

The concept of data durability is borrowed from the database world, where the *D* in the well-known ACID acronym stands for *durability*. ACID, which stands for Atomicity, Consistency, Isolation, and Durability, outlines key features required to guarantee consistent and reliable database transactions.

At the time of this writing, Amazon S3 was claiming 99.999999999 percent durability. Amazon also states that this level of durability gives an average annual expected loss of objects of 0.000000001 percent. Basically, you'd be extremely unlucky to lose an object stored in S3, or as Amazon puts it, *If you store 10,000 objects with Amazon S3, you can on average expect to incur a loss of a single object once every 10,000,000 years*. Obviously, that is referring to the potential of losing objects due to failures within S3, and it doesn't cover you for user error such as accidentally deleting objects.

This kind of extreme data durability is achieved by cloud storage services storing multiple copies of each object throughout the cloud. For example, if you keep your data within the U.S. region of your chosen cloud, multiple copies of your data will be kept across multiple data centers within the U.S. region. This means that if a node, or even an entire data center, fails within the cloud's U.S. region, your data will still be available from other data centers within the same region.

Many cloud storage solutions also offer Geo replication, where objects are also replicated outside regions, across the globe. In a Geo replication scenario, your data is protected across multiple facilities within a region and then also across multiple regions. That is even more protection.

An important concept on the topic of data durability is *atomic updates*—remember, the *A* in ACID stands for *atomicity*! Atomic updates ensure that all updates to cloud storage (new objects or updates to existing objects) are stored to multiple locations within the cloud *before* the cloud confirms that the update was successful. So you know that once an object, or update to an object, is signaled as successfully committed to the cloud, that object has been stored to multiple locations.

Despite the impressive numbers that cloud storage services advertise for data durability, you will still want to back up your data. Data durability on its own does not protect you against accidental deletions or unintentional updates to objects! If a cloud storage solution provides *versioning*—whereby the previous version of an object is preserved every time the object is updated or deleted—then this *may* allow you to stop backing up data that you have stored in the cloud. However, you will want to conduct thorough research before making a decision as important as changing your backup practices.

Eventual Consistency Model

Some aspects of cloud storage platforms work on an *eventual consistency* model. In this model, objects and updates to objects take time to propagate throughout the cloud. As you learned in the section about data durability, objects in a storage cloud are replicated, or propagated, to multiple nodes and locations within the cloud in order to protect against failures within the cloud. Sometimes this propagation of objects and updates to objects occurs asynchronously (such as propagating out of region) meaning that your updates to the cloud are signaled as complete before changes are replicated throughout the entire cloud. For example, it is possible in a cloud storage solution configured for Geo replication for an object to be updated in California in the U.S. region at 10 a.m., and for someone accessing the object from a copy in the APAC region at 10:01 a.m. (California time) to still see a previous version of the object, and for it to be 10:02 a.m. before the updated object is propagated to the APAC region. That being said, atomic updates are still respected, meaning that if your cloud storage solution is configured to make three copies *in region*, these three copies will be updated before any updates to your objects are signaled as complete; it is just the propagation out of the region that should be asynchronous and may take a while to propagate. As long as you leave the system for long enough, without making changes, all copies of your data in the cloud will eventually become consistent. Obviously, if rapid global consistency is important to your organization, cloud-based object storage may not be the best solution.

Some examples of *eventual consistency* models—sometimes called *loose consistency*—include NoSQL databases and some social media and photo-sharing sites. In these solutions, it takes a short amount of time for changes and updates to be propagated throughout the system.

Public Cloud Storage

When most people think about cloud storage, they're thinking of *public cloud storage*, as it's the most popular form of cloud storage. Services such as Amazon S3, Windows Azure, and Rackspace Cloud Files are all examples of public cloud storage services.

At a high level, public cloud storage is storage that's accessed over the Internet, usually via RESTful application programming interfaces (APIs), and is paid for monthly based on what you've consumed that month. Public cloud storage is also multitenant by nature, meaning it has multiple customers sharing the same common infrastructure and storage platforms.

To help you understand how it works, Exercise 15.3 shows you how to configure some cloud storage in Amazon S3.

EXERCISE 15.3

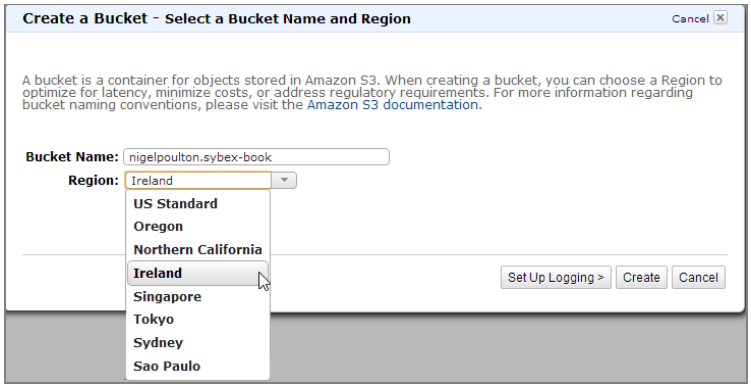
Getting Started with Public Cloud Storage with Amazon S3

In this exercise, you'll walk through the steps necessary to get up and running with Amazon Simple Storage Service (S3). Before you can get started with Amazon S3, you'll need an Amazon Web Services account. For this example, we'll assume you already have one. Now, you'll create a cloud storage bucket, upload an object, and configure custom metadata.

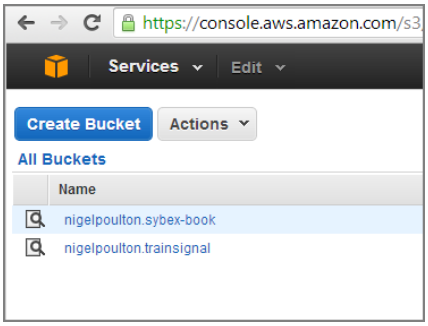
1. From within the AWS web console, navigate to the S3 service.

2. As all objects in Amazon S3 are stored in *buckets*, let's create a bucket. Click the Create Bucket button from within the S3 web interface.

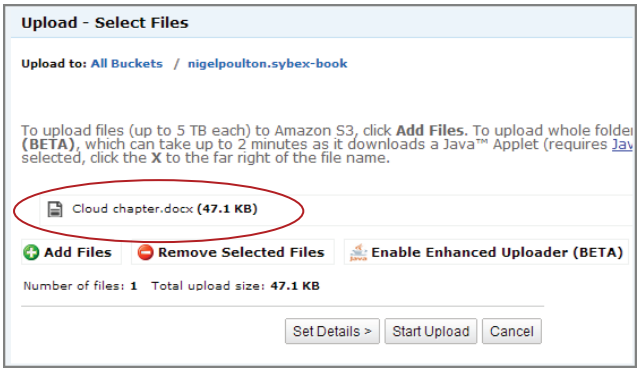
3. In the resulting Create A Bucket screen, select a name for your bucket and choose a region that you would like your bucket stored in. In this example, I'm creating a bucket called nigelpoulton.sybex-book and choosing to store it in the Ireland region.



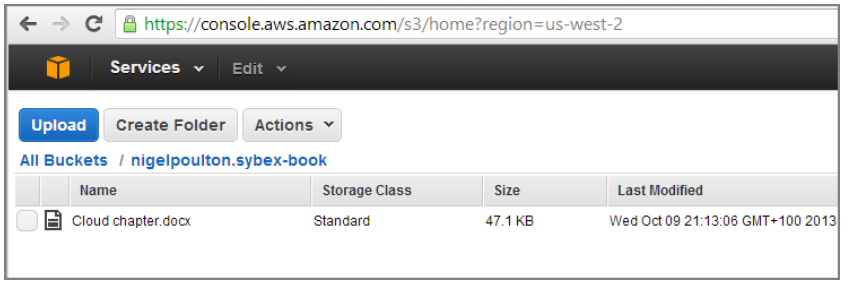
4. Click Create. The newly created bucket appears in the list of buckets.



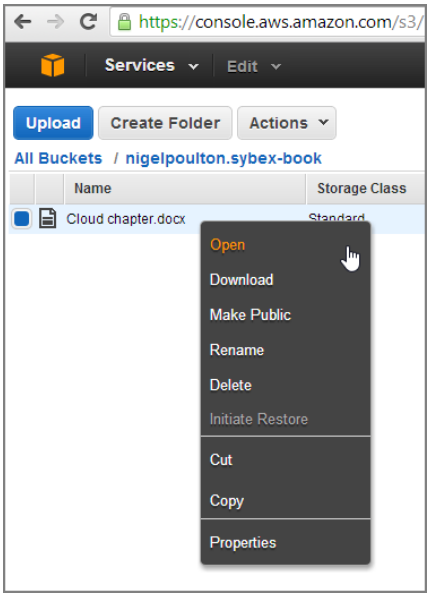
5. Now let's add an object (file) to the bucket. Choose your bucket and click Upload.
6. On the Upload screen, click Add Files and choose whatever files you want to upload to the bucket from the dialog box. As you can see in the following image, I have selected to upload the file for this chapter, currently called `Cloud chapter.docx`.



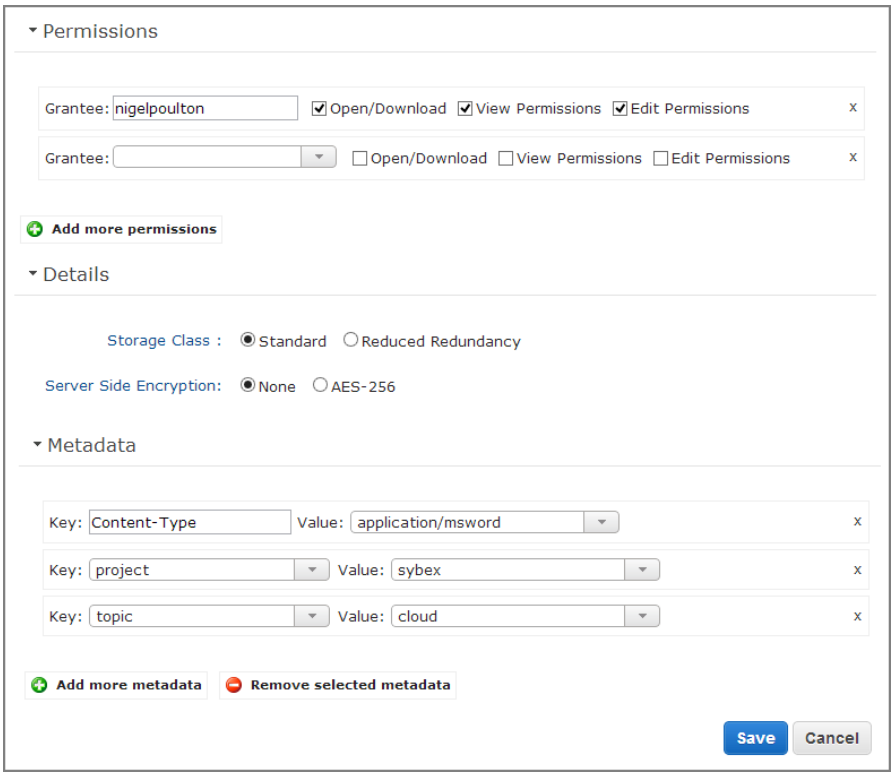
7. Once your file is selected, click Start Upload. The following image shows that my `Cloud chapter.docx` object has been uploaded to the `mynigelpoulton.sybex-book` bucket.



8. Right-click your uploaded object. You get a context menu that lists all the common options you can perform on the object.



9. From the right-click menu, select Properties. A page opens, enabling you to set common properties associated with objects stored in the Amazon S3 cloud, including these:
   - Permissions
   - Encryption settings
   - Redundancy
   - Custom metadata



And there you go. You've just created a cloud storage bucket, uploaded an object, and configured custom metadata.

API Access

As you can see from Exercise 15.3, Amazon S3 storage can be accessed via a web interface. Uploading objects, downloading objects, deleting objects, managing access to objects, and so on can all be done via a web browser. You don't need to mount cloud storage as you do with a SMB/CIFS share or an NFS export—in fact, you can't. And you don't need to format it with a filesystem as you do with a SCSI LUN. Again, you can't. Instead, you access it directly via either a web browser or application that talks directly to it using REST APIs.

As noted earlier, API is an acronym for *application programming interface*, and APIs are kind of like a published set of instructions for accessing and using web-based services. In the context of the public cloud, APIs are almost always RESTful. By RESTful, I mean a system that uses simple put/get style semantics such as the common HTTP `PUT` and `GET` verbs used so extensively on the World Wide Web. Most REST-based systems use HTTP (although they don't have to), and most, although not all, object storage systems are accessed via RESTful APIs.

If APIs are new to you, it might be useful to think of them as being similar to a set of rules and protocols you should follow in order to get something done. For example, the protocol for ordering food at a five-star restaurant is very different from ordering food at a McDonald's. In the five-star restaurant, you are respectfully ushered to your seat and given a menu, you then leisurely browse the menu and wait for the waiter to come over and take your order for drinks, and you then wait until the drinks are delivered before making your food order. You normally keep a tab open and settle the bill after you have eaten. Knowing that helps make your experience enjoyable. But try that same approach in a McDonald's restaurant and you will find yourself very hungry. There will be nobody waiting to seat you and no waiter coming to your table to take your order. And don't even bother trying to eat your food before you've paid for it! APIs aren't that much different. The Amazon S3 API gives you instructions to follow in order to access data in the S3 cloud. Follow the S3 API, and your experience with S3 will probably be good. Don't follow the S3 API, and your experience with S3 will not be so good.

APIs are also software-to-software interfaces, not user interfaces. They are designed to be used for programmatic or systematic access to a system, without requiring user assistance. On this topic, many APIs have *language bindings* to many of the popular programming languages such as Python, Java, PHP, C#, and so on. These language bindings are a higher level (simpler) than pure REST, allowing you to do RESTful operations within the native context of the programming language, making programming APIs a lot simpler.

Public Cloud Storage Performance

When it comes to performance, cloud storage is close to the bottom of the pile, which makes sense. Its lack of proximity to your on-premise systems or applications suggests that cloud storage won't be a stellar performer. Storage performance almost always gets slower as it gets further away from the CPU. For example, main memory (RAM) is close to the CPU and is extremely fast. Locally attached disk is a bit further away from the CPU (you access it over a PCI bus on the system motherboard) and is slower. Network-attached storage (SAN and NAS) is even further away and is slower again. And cloud storage is the furthest away of them all! So not surprisingly, it's the slowest of them all.

It's also interesting to note that the closer you get to the CPU, the less capacity is available, whereas the further away from the CPU, the more capacity. For example, RAM is closer to the CPU on a system than a hard drive is, and a hard drive is closer than a NAS array. From this we should know that RAM capacity is usually a lot less than hard drive capacity, which in turn is usually a lot less than the capacities provided by a NAS array. Cloud storage is true to that rule too, as it is the most scalable form of storage available today.

While we're talking about proximity to the CPU, it also appears to be true that the further away from the CPU your storage is, the more accessible and shareable it is. For example, sharing main memory (RAM) is relatively difficult compared to sharing space on a locally attached hard drive. Likewise, NAS arrays were designed with shareability in mind, as was cloud storage. Cloud storage provides the ultimate in storage accessibility and shareability.

With all of this said, cloud storage has its use cases. In some use cases, such as accessing large objects such as photos and video files, cloud storage can be a good performer. This is because access to these types of objects requires good throughput rather than low latency; a single HTTP `GET` can deliver a very large amount of data. And Internet bandwidth is getting cheaper and cheaper.

Atomic Uploads

All good public cloud storage solutions work based on the principle of *atomic uploads*. This is a good thing. Atomic uploads ensure that an object is not available to be accessed from the cloud until the upload operation is 100 percent complete. That atomic upload process usually involves making several copies of the object before signaling the upload as complete. This means that failures within the cloud should never result in you losing your objects. Atomic uploads include any associated metadata and security properties.

Geo Replication

Public cloud services often offer global replication capability, sometimes called *Geo replication*. This is where objects can be replicated across the cloud's worldwide infrastructure. This can obviously improve data durability—the ability of your data to survive large-scale local disasters and failures.

Another major use for Geo replication is local access to data. For example, if your public cloud provider has data centers in the United States, the United Kingdom, and Singapore, and you have offices in the United States, the United Kingdom, and Australia, you might want your objects replicating to all of your cloud provider's facilities so that staff in each of your locations around the world can access data without always having to come back to the United States. However, for legal or regulatory purposes, you may not be allowed to have your data leave a certain region or legal jurisdiction. For example, the European Union (EU) has laws that prohibit certain data from leaving the EU. Also, you may be concerned about government access to data that is stored within various countries.

Content Delivery Networks

On the topic of local access to data, the closer your data is to your users, the better the performance and the greater the chance of a good user experience. For example, if your public cloud provider has data centers in the United States, the United Kingdom, and Singapore, and you have users in Hong Kong, you'll probably want your users in Hong Kong to have access to data via the cloud provider's facilities in Singapore. However, it's still a long way from Hong Kong to Singapore. And this is where content delivery networks (CDNs) come to the rescue. CDNs work by caching objects closer to users. They are able to do this by having caching nodes in *edge locations* dotted around the globe. For example, Akamai Technologies, a CDN provider that partners with Rackspace, can

cache objects in Hong Kong so users in Hong Kong have local access to data rather than having to traverse backward and forward across the Internet to Singapore every time they want access to an object.

Sometimes edge locations are referred to as *points of presence (POP)*.

So CDNs take the *access your data from any device, anywhere in the world* philosophy of cloud storage a step further, making it *access your data from any device, anywhere in the world, FAST!*

You may sometimes hear CDN explained as a content *distribution* network, rather than a content *delivery* network. Both terms mean the same thing. Some cloud solutions have their own CDN, such as Amazon S3, whereas others partner with a third-party CDN specialist such as Rackspace partnering with Akamai for their Cloud Files cloud storage platform. As an example, Akamai (a CDN provider) has over 100,000 servers in more than 200 *edge locations* dotted around the globe that can cache data so that it is closer to your users. On the cost side, you will also want to check whether there are additional charges for using a CDN.

While public cloud storage might pose challenges and risks to you and your organization, there is definitely a trend within organizations to offload their lower-tier requirements and their lower-risk data and services to someone else—and that someone else is increasingly becoming the public cloud.

Real World Scenario

Using Public Cloud Storage for Small Offices

Many organizations are starting to utilize public cloud storage offerings to facilitate storage requirements in remote branch offices where no IT staff are employed. A popular trend is pushing backups to the cloud, as volumes of data in small offices tend not to be large, and are therefore well suited to backup to the cloud. For example, rather than having physical tapes, or even a small dedicated disk unit/VTL at a remote branch office to cater for backup storage, many organizations are starting to send regional office backups to the cloud, via interfaces such as AWS VTL interface (Gateway-VTL). This avoids the need to deploy dedicated hardware at remote branch office, and removes any need for manual tape handling etc.

Private Cloud Storage

Private cloud storage comes in two major flavors:

- On premise
- Off premise

A common characteristic of both on premise and off premise is that customers of private cloud storage get their own private infrastructure—their own dedicated storage systems. This is in stark contrast to public cloud storage, where thousands of customers share the same underlying infrastructure and storage systems.

Despite being the polar opposite of public cloud storage when it comes to tenancy, private cloud storage services aim to emulate many of the other features of public cloud storage, including these:

- Simplicity
- Elasticity
- Scalability
- Fast self-service provisioning
- Built-in protection/data durability
- Op-ex-based consumption billing

In an *on-premise* private cloud storage solution, you access storage that is owned by a third party but resides in your data center, over your corporate LAN. In an *off-premise* private cloud storage solution, the storage is moved to the service provider's data center and is accessed over either the Internet or a private WAN link.

For companies that are nervous about letting their corporate data out of their hands and onto a massively shared public platform, private cloud storage may be a more palatable solution than public cloud storage. It may also be a good stepping-stone or halfway house, while companies warm up to the idea of public cloud storage.

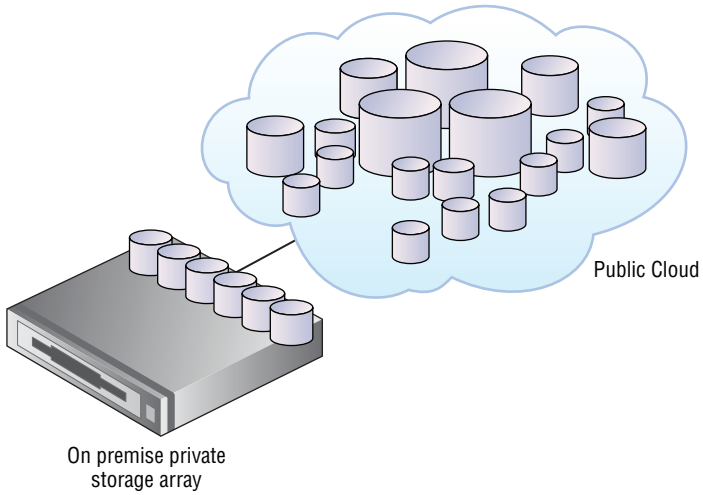
Hybrid Cloud Storage

In *hybrid cloud storage*, a single application or product uses a mix of public cloud storage and private storage, giving the best of both worlds. Some examples of a commercially available hybrid storage solution include Microsoft StorSimple, Nasuni, Panzura, and TwinStrata. As an example, Microsoft StorSimple is a hardware and software storage array that you buy and locate in your data center but that can natively use the Windows Azure cloud as a low tier of storage. At a high level, the StorSimple product comprises an on-premise hardware storage array with internal SSD and SAS drives, as well as native integration with Windows Azure cloud storage. This can give the performance benefits of a

traditional on-site storage system with SSD and SAS drives for fast access to frequently referenced data, as well as making Windows Azure appear as a tier of storage for storing less frequently accessed data. Microsoft is calling StorSimple *cloud integrated storage (CIS)*, as well as calling the use of Windows Azure *cloud as a tier (CaaT)*.

Figure 15.3 shows a simple hybrid cloud storage system such as Microsoft StorSimple.

FIGURE 15.3 Hybrid cloud storage device



Another common example of an increasingly common hybrid cloud storage solution is *hybrid cloud backup*. In a hybrid cloud backup solution, a staging appliance is located in your data center to store one to two weeks' worth of backups *locally* so they can be used to perform fast restores. Those backups are moved to the cloud after those one to two weeks.

CDMI

A standards-based API that you may come across in relation to cloud storage is *Cloud Data Management Initiative* (*CDMI*). This SNIA initiative aims to bring some standards to the world of cloud storage—standards for storing, accessing, and managing objects stored in the cloud. However, CDMI is practically irrelevant because of the massive popularity of other APIs including S3, OpenStack Swift (Cloud Files), and Azure. Most people who are coding for cloud storage are more commonly coding against S3, Swift, and Azure. Not as many people are coding against CDMI. However, more object storage systems are starting to support it.

De Facto and De Jure Standards

There are two major types of standards to be aware of:

- De jure standards
- De facto standards

De jure standards are those handed down to us by standards bodies and committees such as SNIA, IEEE, and ISO. *De facto standards* aren't controlled and handed down to us by standards bodies, but are so widely used in the real world as to be potentially more important than de jure standards. In the cloud storage space, CDMI is an example of a de jure standard that is written and regulated by a standards body, but not very widely adopted and practiced. Conversely, S3 and Swift are so widely used that they become almost unofficial standards that everybody abides by and practices.

Both have their pros and cons.

Architecturally speaking, CDMI-based solutions look and feel like any other cloud-based object store. CDMI stores objects in *containers* that are flat namespaces similar to buckets in Amazon S3. Objects are stored within containers, and containers cannot be nested. All objects within a CDMI cloud store are given a URL by which they can be accessed and manipulated via REST using the common HTTP verbs such as `PUT`, `GET`, `DELETE`, and so on. CDMI also supports other essential cloud storage features such as security and authentication.

CDMI is implemented as a client-server architecture. A CDMI client makes CDMI requests to a cloud storage service. The CDMI server in the cloud storage solution then represents data back to the CDMI client. This all occurs over HTTP/HTTPS and is very web browser and web friendly as the CDMI server is built on top of HTTP. An example of a client requesting to view a PDF file called `cloud-chapter.pdf` in a container called `sybex-book` on a cloud platform accessed at `http://cloud.provider.com` might look like the following:

```
GET /sybex-book/cloud-chapter.pdf HTTP/1.1
Host: cloud.provider.com
Accept: */*
```

As well as being able to create, retrieve, update, and delete data in the cloud, CDMI supports the ability to discover the capabilities of a particular cloud storage solution.

Summary

In this chapter, you learned the basics of the cloud computing model and the differences between the three popular forms of cloud computing: public cloud, private cloud, and hybrid cloud. You learned how the cloud is simple, elastic, scalable, self-service, and paid for on a pay-for-what-you-use model. You then learned about storage-specific cloud concepts including performance, atomic updates, data durability, and API access. Throughout the chapter, you touched on what the cloud may, and may not, be a good solution for. You also walked through examples of creating virtual servers and object stores in some of the most popular public cloud services available.

Chapter Essentials

Cloud Computing In the cloud computing model, computing resources such as CPU, network, and storage are accessed remotely via the Internet. This model has several advantages over more-traditional computing models, such as elasticity, scalability, and accessibility.

Cloud Storage Cloud storage is accessed over the Internet via APIs. It shares many of the advantages of cloud computing, including elasticity, scalability, and accessibility.

Public Cloud Public cloud services, including cloud computing and cloud storage, operate on a shared infrastructure. Multiple customers sharing the same infrastructure is known as multitenancy.

API Access Application programming interfaces (APIs) allow services such as cloud computing and cloud storage to be accessed from within an application. As long as an API for a cloud service is made publically available, you can write your own code (software) to speak directly with that cloud service.