Chapter

1

Introducing Cloud Computing Configurations and Deployments

THE FOLLOWING COMPTIA CLOUD+ EXAM OBJECTIVES ARE COVERED IN THIS CHAPTER:

✓ **1.1 Compare and contrast the different types of cloud models.**

- Deployment models
  - Public
  - Private
  - Hybrid
  - Community
  - Cloud within a cloud
  - Multicloud
  - Multitenancy
- Service models
  - Infrastructure as a service (IaaS)
  - Platform as a service (PaaS)
  - Software as a service (SaaS)
- Advanced cloud services
  - Internet of Things (IoT)
  - Serverless
  - Machine learning/Artificial intelligence (AI)
- Shared responsibility model

✓ **1.3 Explain the importance of high availability and scaling in cloud environments.**

- Hypervisors
  - Affinity
  - Anti-affinity
- Regions and zones
- High availability of network functions
  - Switches
  - Routers
  - Load balancers
  - Firewalls
- Scalability
  - Auto-scaling

✓ **1.4 Given a scenario, analyze the solution design in support of the business requirements.**

- Environments
  - Development
  - Quality assurance (QA)
  - Staging
  - Production
- Testing techniques
  - Vulnerability testing
  - Penetration testing
  - Performance testing
  - Regression testing
  - Functional testing
  - Usability testing

- ✓ **3.1 Given a scenario, integrate components into a cloud solution.**
  - Application
    - Serverless
- ✓ **4.1 Given a scenario, configure logging, monitoring, and alerting to maintain operational status.**
  - Monitoring
    - Baselines
    - Thresholds
- ✓ **4.3 Given a scenario, optimize cloud environments.**
  - Placement
    - Geographical
    - Cluster placement
    - Redundancy
    - Colocation
- ✓ **4.4 Given a scenario, apply proper automation and orchestration techniques.**
  - Automation activities
    - Routine operations
    - Updates
    - Scaling

Introducing Cloud Computing

You'll begin your CompTIA Cloud+ (CV0-003) certification exam journey with a general overview of cloud computing. With a strong understanding of cloud terminology and architectures, you'll better understand the details of the cloud, which in turn means that you'll be better prepared for the Cloud+ exam and be effective when planning, deploying, and supporting cloud environments.

Let's start by briefly looking at where cloud sits in the broader scope of the IT world. Before cloud computing, organizations had to acquire the IT infrastructure needed to run their applications. Such infrastructure included servers, storage arrays, and networking equipment like routers and firewalls.

Options for where to locate this infrastructure were limited. An organization with an ample budget might build an expensive *data center* consisting of the following:

- Racks to hold servers and networking equipment
- Redundant power sources and backup batteries or generators
- Massive cooling and ventilation systems to keep the equipment from overheating
- Network connectivity within the data center and to outside networks such as the Internet

Organizations that are less fiscally blessed might rent physical space from a *colocation* (colo) facility, which is just a data center that leases rack space to the general public. Because colo customers lease only as much space as they need—be it a few rack units, an entire rack, or even multiple racks—this option is much cheaper than building and maintaining a data center from scratch. Customer organizations just have to deal with their own IT equipment and software. To put it in marketing terms, think of a colocation facility as "data center as a service."

Cloud computing takes the concept of a colocation facility and abstracts it even further. Instead of acquiring your *own* IT equipment and leasing space for it from a colo, an organization can simply use a *cloud service provider*. In addition to providing and managing the data center infrastructure, a cloud service provider (or just provider for short) *also* handles the IT hardware infrastructure—namely servers, storage, and networking. The *consumer* of the cloud services pays fees to use the provider's equipment, and such fees are typically based

on usage and billed monthly. In this chapter, we'll dive further into the details of how cloud computing works and how an organization might use the cloud instead of—or in addition to—a traditional data center or colo model.

Related to the data center versus cloud distinction, there are two terms that you need to know. *On-premises* (on-prem) hosting refers to an organization hosting its own hardware, be it in a data center or a colo. In contrast, cloud computing is an example of *off-premises* (off-prem) hosting, as the hardware resources are *not* controlled by the organization that uses them. To make this distinction easy to remember, just equate on-prem with data center and off-prem with cloud.

This book will reference the National Institute of Standards (NIST) SP 800-145 publication (`https://doi.org/10.6028/NIST.SP.800-145`) as the main source of cloud computing definitions. NIST defines cloud computing as follows:

> . . . a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Pay close attention to that last sentence. The reference to "minimal management effort or service provider interaction" is code for automation. Unlike a traditional data center or colocation facility, using the cloud doesn't require someone to physically go to a facility to install servers or plug in cables. There's no need for "remote hands" because cloud providers offer self-service management consoles that do the provisioning for you. When you think about it, cloud providers are really just providing automation-powered managed services for nearly every conceivable IT need.

In this study guide, you'll take a close look at all aspects of cloud computing as you progress on your journey to obtaining the Cloud+ certification. As an analogy to cloud computing, think of the services you get in your home through utilities, such as electricity and water services. You are probably not involved in the details of how these services are created and delivered; you turn the water faucet on and off, and pay only for what you use. Cloud computing follows the same principle, albeit applied to a variety of IT services.

Traditionally, computing did not follow this model. Instead, an organization would purchase all the hardware and software necessary to meet their needs. Adding to the cost, they needed to maintain a staff of specialized engineers to operate and maintain the systems. And they'd have to purchase more equipment than they needed to leave room for growth. This meant high capital outlays without a corresponding immediate payoff. As I mentioned earlier, organizations had to build and secure data centers to host, power, and cool the equipment.

Like utilities, cloud computing follows a pay-as-you-go model, where a provider sells computing resources that you consume as needed. This allows organizations to pay only for what they use, and it has many additional advantages that you'll explore throughout this book.

The market and adoption of the cloud computing business has exploded worldwide. In just the past decade, cloud computing has gone from a novelty for early adopters to a dominant position in the marketplace today. Although there are many statistics and measurements

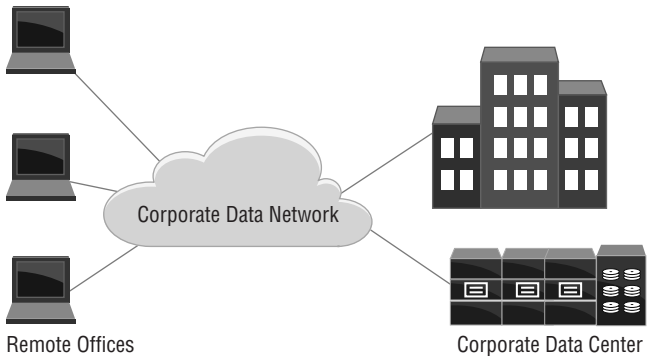
of the size, it is generally agreed that the market has been growing at least 15 percent annually worldwide. Current forecasts estimate the total cloud market worldwide will be more than $800 billion by the year 2025. What is clear is that the economics and business advantages of cloud computing are compelling companies to move more and more applications to the cloud, fueling additional growth well into the future.

There are many advantages to moving to the cloud, but three stand out as compelling business and operations alternatives to hosting computing resources internally in your own data center or in a colocation facility:

- In the past when computing resources were initially needed, there was often a long delay of procuring, installing, and configuring all the pieces needed to host an application. With a cloud solution, the equipment is already running in a cloud provider's data center, and you can begin hosting your application in record time, sometimes as short as in a few minutes.
- From a financial perspective, a company's capital expenditures can be reduced as cloud computing avoids the large up-front costs of purchasing the needed computing equipment and ongoing support expenses associated with maintaining it. Cloud computing, with its pay-as-you-go billing model, frees up a company's cash flow for other needs.
- As your computing or storage needs grow, a cloud computing model can expand almost immediately. Contrast this with the data center model in which you have to procure, install, and configure new equipment or software—a process that can take days if not weeks.

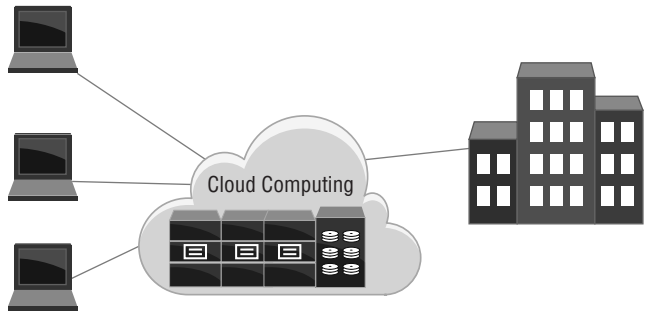
In-house computing requires a data center with the computing gear needed to support the organization's operations. The organization must hire engineers to tend to the operating systems, applications, storage, and networks. As illustrated in Figure 1.1, all computing is owned and operated by a single entity.

FIGURE 1.1 In-house computing



When moving to the cloud, you outsource many of these data center operations to a cloud service provider, as shown in Figure 1.2.

FIGURE 1.2 Cloud computing model



It's important to understand that the organization's data center does not actually move into the cloud, nor does the cloud provider take over the operations of the organization's data center. Instead, the cloud provider has its *own* data centers that already contain all the equipment that you need to host your applications and other IT services. The cloud provider can reach economies of scale by sharing its physical IT resources with many companies. This sounds similar to a colocation facility, but there's one big exception: you do not have physical access to the cloud provider's data center. The only way that you can use their computing resources is via the management interfaces—typically web consoles or application programming interfaces (APIs)—that they provide. There's a trade-off here: you give up control of the physical IT infrastructure in exchange for the convenience of a pay-as-you-go billing model.

In the distant past, computing was the realm of large *mainframe computers*, with a staff of highly specialized engineers and teams of programmers and administrators to manage and run the operations. Figure 1.3 shows a typical mainframe architecture. Mainframe computing was a highly capital-intensive operation that was needed to supply computing resources to a corporation.

As computers became smaller and more powerful, the client-server architecture shown in Figure 1.4 grew prevalent, and we saw the rise in departmental computing that was distributed throughout a company's operations.

Virtualization

Virtualization is what makes cloud computing possible. Simply put, *virtualization* is the ability to take physical data center resources such as servers, storage, and networking and abstract them as services that can be delivered as cloud offerings. A key benefit of virtualization is that it allows different customers to share the same physical IT infrastructure. Without this ability, cloud computing wouldn't be possible.

FIGURE 1.3 Mainframe computing

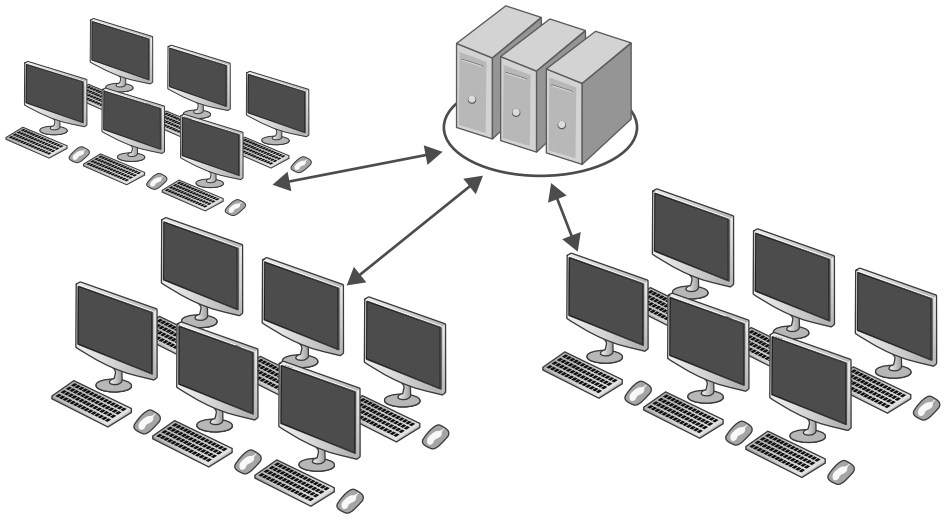
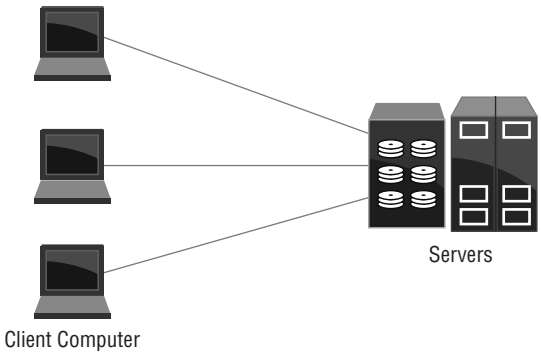


FIGURE 1.4 Client-server computing



The topic of virtualization can be confusing because people use the term in wildly different ways. In its most basic interpretation, "virtual" can mean anything not purely physical—not a very helpful distinction. Hence, it helps to break virtualization down into two different categories:

- Machine virtualization
- Network virtualization

These distinctions aren't particularly important in the context of cloud computing, but because it's easy to apply the "virtual" moniker to anything and everything, we need to draw some boundaries around when the designation is appropriate.

Machine Virtualization

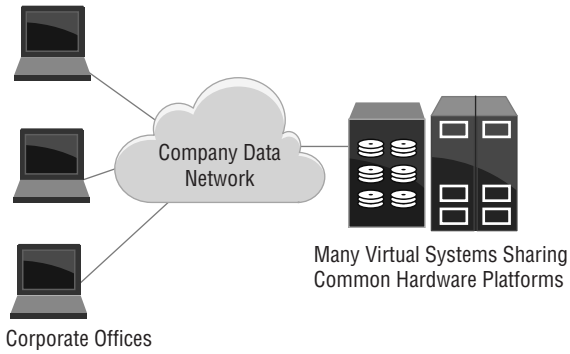
Machine virtualization—also called server virtualization—involves abstracting the resources of a single physical server into multiple virtual machines (VMs). Essentially, a VM is a software-emulated computer consisting of *virtual CPUs* (vCPUs), memory, storage, and networking. Like a real computer, a VM runs an operating system (OS) called a guest OS. The software that creates virtual machines and performs this abstraction is called a *hypervisor*. The hypervisor also defines the properties of a VM, including the following:

- The number of virtual CPUs
- The amount of random access memory (RAM)
- The type and amount of storage
- Virtual network interfaces and how they're connected

We'll discuss hypervisors in more detail in Chapter 2, "Cloud Deployments." Virtualization not only allows for more efficient use of hardware resources, but also reduces power consumption, cooling, and the server footprint in data centers. This is illustrated in Figure 1.5, where many VMs share common hardware platforms.

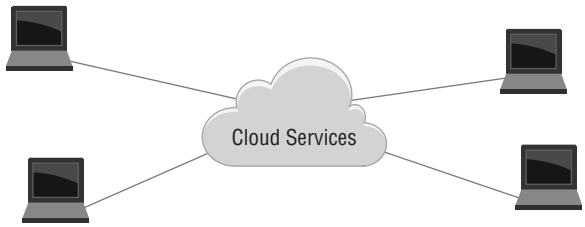
A physical server that runs VMs is called a virtualized host.

FIGURE 1.5 Virtualized computing



The proliferation of Internet-capable devices, such as smartphones and tablets, means an increased strain on IT infrastructure. Organizations that want to offer services to these customers can benefit greatly from the cloud's utility-like service model (see Figure 1.6). The cloud supports instant access in an always-on environment.

FIGURE 1.6 Cloud computing



Network Virtualization

By definition, computer networks are already virtual because they're just an abstraction of physical connections. But in the context of cloud computing, *network virtualization* refers to virtual private clouds (VPCs)—isolated private networks within the cloud that allow connectivity among virtual machines and other cloud resources. There's some overlap with machine virtualization here, because VMs have virtual network interfaces that connect to these virtual private clouds.

Cloud Service Models

If a cloud provider controls all the hardware aspects of your IT infrastructure, what do you get to control? The answer depends on the type of *cloud service model* you choose. Cloud service models fall into three categories, all of which are characterized by the term as a service:

- Software as a Service (SaaS)
- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)

Many cloud service providers use more descriptive terms in their marketing, including Communications as a Service (CaaS), Anything as a Service (XaaS), Desktop as a Service (DaaS), and Business Process as a Service (BPaaS), to name a few. However, all of these clever names fit into the SaaS, IaaS, or PaaS categories.

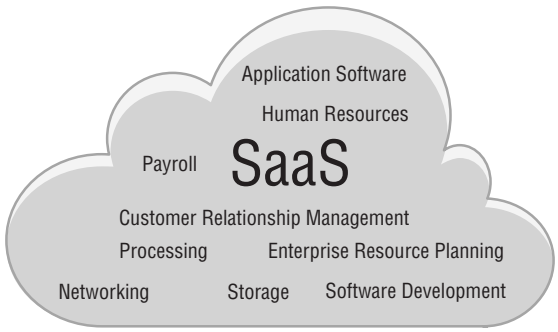
Software as a Service

Software as a Service (SaaS) most closely aligns with what used to be called a managed software service. For example, in the early days of the cloud, hosting companies would offer a hosted version of a certain popular, brand-name enterprise email system. Instead of having to buy, configure, and maintain this email system on your own servers, the hosting company would do it for you on its own servers. All you had to do was configure your client machines to connect to the appropriate server to send and receive email. NIST formalizes this concept in the following description of Software as a Service:

> The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based e-mail), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

The Software as a Service model is where the customer of the service accesses the application software that is owned and controlled by the cloud company, which has complete responsibility for the management and support of the application, as shown in Figure 1.7. You, on the other hand, have limited control over the operation and configuration of the software itself. Sticking to the earlier example, you can create and delete email inboxes and control how inbound and outbound emails are processed. But you can't upgrade the software to a newer version.

FIGURE 1.7 SaaS



Business applications are good examples of SaaS and can include customer relationship management, enterprise resource planning, human resources, payroll, and software development applications. Hosted applications such as email or calendars that are accessible from a browser or email client are examples of SaaS.

Infrastructure as a Service

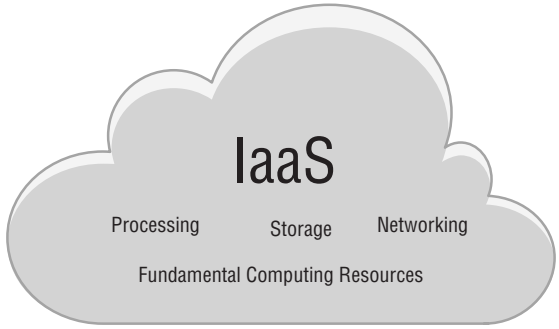
The *Infrastructure as a Service (IaaS)* model lets you create VMs and virtual networks in the cloud according to your desired specifications regarding processing power, memory, storage, and networking. The IaaS model is probably the easiest to understand because it most closely mirrors the virtualized server environments in modern data centers. NIST describes it as follows:

> The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

IaaS is really just a server and network virtualization environment offered as a service. Because of this, it offers the customer the most flexibility of any of the e-service models. You can provision to your specifications any number of VMs, on which you can run the software of your choice. Also, some IaaS offerings even allow you to choose the virtualized host on which your VMs run, giving you the ability to spread VMs across multiple hosts for resiliency.

IaaS (shown in Figure 1.8) allows the company's data center equipment to be replaced by the cloud equivalent but retains the ability to build software infrastructure on top of the hardware as can be done in a private data center.

FIGURE 1.8 IaaS



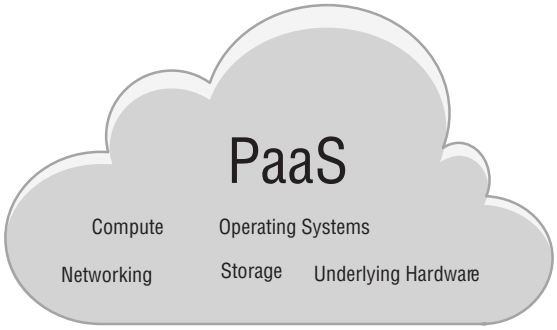
Platform as a Service

The *Platform as a Service (PaaS)* model sits somewhere in between the IaaS and SaaS models, and conceptually it is probably the most difficult to grasp because it doesn't have a clear analog in the data center. Essentially, PaaS gives you a preconfigured computing environment on which to install and run the software of your choice. You have little to no control over the configuration of the OS and VMs on which your application runs. NIST describes it thusly:

> The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations.

A strange but accurate way of saying it is that PaaS offers an operating system as a service on which customers can install their applications, as shown in Figure 1.9. The cloud provider takes responsibility up to and including the operating system, including all hardware and virtualized resources.

FIGURE 1.9 PaaS



Not surprisingly, PaaS is a popular model with software developers because they can deploy their applications quickly without having to mess with provisioning VMs and keeping up with OS maintenance.

Communications as a Service

Communications as a Service (CaaS) is a particular instance of SaaS that includes hosted voice, videoconferencing, instant messaging, email, collaboration, and all other communication services that are hosted in the cloud. These outsourced corporate communication services can support on-premises or mobile users accessing the applications hosted in the cloud.

The CaaS model allows even small to medium-sized businesses to implement advanced technologies at a reasonable metered cost. There is no need for a staff to manage these communication services since the CaaS cloud provider takes responsibility. Another common term for this service is *Unified Communications as a Service (UCaaS)*.

Database as a Service

Database as a Service (DBaaS) is a manifestation of SaaS in which the cloud provider gives you a turnkey database management system on which you can create your own databases. The provider takes care of the hardware and virtual infrastructure, the operating system, and the database software itself. You need concern yourself only with the databases that you create.

Desktop as a Service

Desktop as a Service (DaaS) provides virtual desktops that consumers can access remotely via desktop or laptop computers, mobile devices, or thin clients. This solution is sometimes called *virtual desktop infrastructure (VDI)*. All desktop applications are hosted in the cloud and can consist of any type of application, such as spreadsheets, word processing, and any other common application. You choose what software to install. The DaaS provider manages all maintenance and configurations as well as licensing and version updates. DaaS is an example of the PaaS model.

Business Process as a Service

Business Process as a Service (BPaaS) is a specialized area that outsources many of a company's day-to-day operations such as inventory, shipping, supply chain, finance, and many other services to the cloud. This allows for small and medium-sized businesses to access sometimes very expensive applications from a BPaaS service provider that pools its resources and allows for economies of scale in providing these services. BPaaS is another instance of the SaaS model.

Anything as a Service

Anything as a Service (XaaS) could best be described as offering complete IT services as a package. XaaS is the combination of the services described in this section. It is a broad term that is a catchall of the various service offerings.

Cloud Reference Designs and Delivery Models

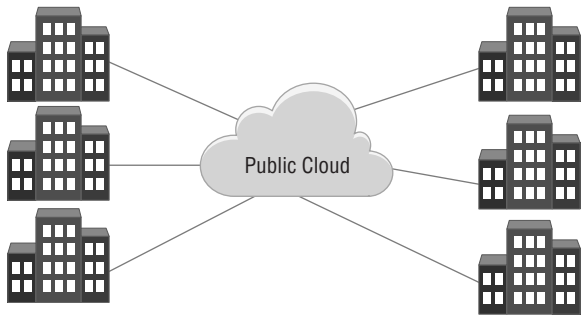
The cloud computing industry has created *reference designs* and *delivery models* to help differentiate between cloud offerings in the marketplace. By understanding the types of models, you can get the big-picture overview of the overall scope of cloud computing. This section will introduce the cloud models, and then in Chapter 2, I will expand on each one. These are the four primary cloud delivery models:

- Public
- Private
- Community
- Hybrid

Public Cloud

The primary focus of the Cloud+ certification is the *public cloud*, which is the most common delivery model deployed. The public cloud is designed for use by the general public. This is the utility-based pay-as-you-go model. Figure 1.10 illustrates a basic public cloud deployment.

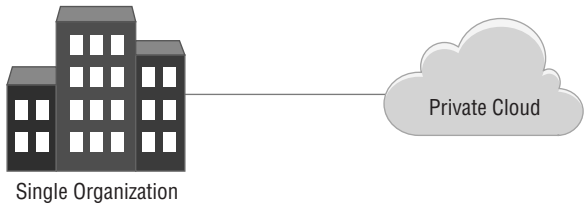
FIGURE 1.10 Public cloud



Private Cloud

A *private cloud* is for the exclusive use of a single organization, but it may be used by many units or entities inside a company. Figure 1.11 illustrates a basic private cloud deployment.

FIGURE 1.11 Private cloud

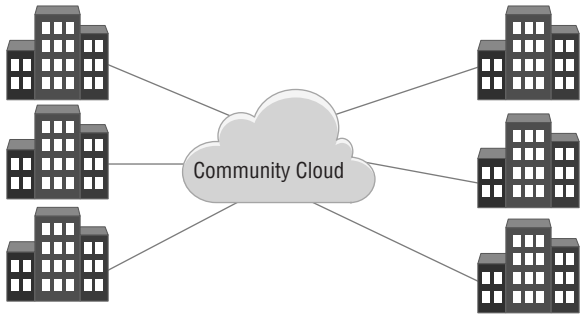


Some people have misappropriated the term *private cloud* to apply to a data center, but this is incorrect. In order to be a private cloud, it has to have the same sort of automation-powered self-service facilities as a public cloud. A traditional data center is not a cloud.

Community Cloud

Community clouds are offered for a specific community of interest and shared by companies with similar requirements for regulatory compliance, security, or policy. Examples include clouds designed for medical, financial, or government organizations that all share common use cases or require standardized architectures. Figure 1.12 shows an example of a common community cloud deployments.

FIGURE 1.12 Community cloud

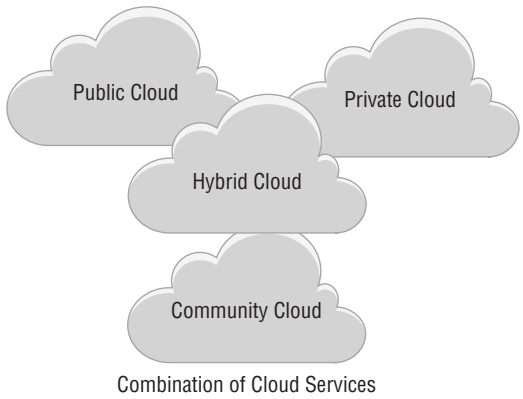


Hybrid Cloud

A *hybrid cloud* is what you get when you connect multiple cloud infrastructures that may or may not be of the same type (public, private, or community). For example, a dentist's office may use the public cloud for its email and office applications but also connect to

a community cloud shared by other dentists to access an application for storing patient records. Figure 1.13 shows examples of hybrid computing.

FIGURE 1.13 Hybrid cloud



When you use multiple cloud providers redundantly, it's called a multicloud deployment. Multicloud deployments are common when there's a need to avoid the unlikely failure of an entire provider, or to avoid cloud provider lock-in.

Colloquially, a hybrid cloud may also refer to connecting cloud-based resources to a data center or office. Although technically this isn't a hybrid cloud, understand that this is actually what most people mean when they use the term.

Introducing Cloud Concepts and Components

Cloud deployments make heavy use of on-demand self-service provisioning, resource pooling via virtualization, rapid elasticity, and a metered or pay-as-you-go pricing model. In this section, we will discuss some common cloud concepts and components.

Applications

The term *application* is broad, but it usually refers to the software that an organization's end users interact with. Some examples include databases, web servers, email, big data, and line-of-business software applications.

Automation

Automation plays a critical role in modern cloud services. Cloud providers employ proprietary automation software that automates the deployment and monitoring of cloud resources, including network, storage, and compute. Automation makes rapid deployment and teardown possible, and it gives users granular control over their cloud usage.

Compute

Simply put, the cloud services that run your applications fall under the category of *compute*. People often think of compute as just virtual machines running in the cloud, but this is only half the story. Compute may refer to one of two things: IaaS virtual machines, or so-called serverless computing.

IaaS Compute may refer to an IaaS service that lets you provision virtual machines, storage, and networking resources in the cloud.

Serverless/FaaS Compute can also refer to what the marketers call *serverless* computing and what the technophiles call *function-as-a-service* (FaaS). In this model, the cloud provider hands you a slick interface into which you can upload your own application code written in a variety of programming languages, and the cloud provider executes it on compute infrastructure that they fully manage. This model obviates the need to provision virtual machines. Instead, the cloud provider handles the compute infrastructure, so all you have to do is deal with the application code. FaaS is a type of PaaS offering.

I've never figured out why it's called *compute* instead of the more familiar *computing*. My best guess, however, is that it's to distinguish the cloud model from the data center model. The term *compute* is used almost exclusively of cloud infrastructure.

Networking

Cloud providers offer most of the traditional networking functionality that you would find in a data center. The difference is that in the cloud, the networking functions provided by traditional firewalls, routers, switches, and load balancers are implemented in the provider's proprietary software. The upside of this approach is that it allows the provider to achieve high availability for these core networking functions.

In the IaaS model, cloud providers also offer Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), and virtual private cloud networks as part of the service, so you don't have to spin up your own DNS or DHCP servers as you would in a data center environment.

Security

Just as security is a critical component in private and corporate data centers, so is it in the cloud. Cloud service providers offer many security services, including firewalls, access control, intrusion detection and prevention systems, and encryption services.

Storage

Large storage arrays and storage area networks exist in the cloud for use by cloud service consumers. Common storage media are solid-state drives (SSDs) and magnetic physical

drives. Storage types include object-based, block-based, and filesystem-based systems. Some storage is optimized for high availability and durability, and others are less expensive and offer long-term archival storage.

Connecting the Cloud to the Outside World

Cloud providers give you complete control over how open or closed your cloud resources are to the rest of the world. If you want to offer a service that's available to anyone anywhere in the world, you can do that. *Ubiquitous access* refers to the ability to access cloud resources from anywhere in the network from a variety of devices such as laptops, tables, smartphones, and thin or thick clients. On the other hand, if you want to restrict access only to those within a particular office, you can do that as well. Because most cloud providers are security-conscious, they prohibit access to your cloud resources by default. You have to explicitly allow access.

Deciding Whether to Move to the Cloud

Organizations that blindly decide to move some of their IT infrastructure to the cloud are sometimes met with an unpleasant surprise when they find out how difficult and expensive it can be. It's not necessarily that the cloud is prohibitively expensive. The surprise comes from failing to understand the dependencies that exist among different IT resources in the data center. When one IT resource moves from the data center to the cloud, it usually has to drag a few other resources with it. For example, moving a database-backed application probably requires moving the database, which might be quite large. Naturally, whoever manages that database is going to have to back it up, so backups will have to be stored in the cloud as well.

Hence, you must have a very clear and detailed understanding of what it is that you are actually moving. This means having updated documentation that reflects all aspects of your operations. To perform a migration, you must know exactly which applications you are running, their dependencies, along with any storage, network, operating system, processing, memory, and any other relevant requirements. The more detailed assessment of your current operations, the better equipped you are to decide whether it makes sense to move to a cloud-based model.

Selecting Cloud Compute Resources

Let's talk about some considerations for migrating on-premises compute resources into the cloud. In the data center, you have virtual machines. It's often possible to migrate a virtual machine directly to the cloud, but the better approach is usually to create a new virtual machine in the cloud and configure it from scratch. One reason for this is that a virtual machine running in your data center will have drivers and other software specific to the virtualization platform that you're using, which will undoubtedly be different than what the cloud provider is using. Performing a "lift and shift" migration to the cloud is asking for trouble.

The sizing of virtual machines—processor power, memory, and storage—should mirror that of your data center VMs. VMs in the cloud are fundamentally no different than the

VMs in your data center. There is no "cloud magic" that makes cloud-based VMs more resource efficient. For example, an application that requires 16 GB of memory in the data center will require just as much memory when it's running in the cloud. Make sure that there is ample input/output (I/O) and low latency for storage access, and that there is enough processing and memory allocated to ensure the proper level of performance expected.

One final word on migrating machines to the cloud. Prior to machine virtualization, each physical server ran a single operating system on *bare metal*. This often meant that one server could host only a handful of applications. Nowadays, machine virtualization is the norm, both in the data center and in the cloud. However, you may occasionally run across an old application that for whatever reason has to run on a bare-metal server. Such applications generally aren't good candidates for moving to the cloud. If you're not sure, you can sometimes smoke out such legacy applications by looking for servers that have been exempted from operating system updates.

Hypervisor Affinity Rules

Another consideration involves the *physical* placement of your data center VMs. To ensure resiliency in the event of a virtualization host failure, organizations often use *hypervisor affinity* rules to ensure that redundant VMs never run on the same hardware. For example, you may have primary and secondary SQL database servers, each running on a different VM host. If the host running the primary SQL VM fails, the secondary VM running on a different host can take over. If both are running on the same host, then both VMs fail, defeating the point of redundant VMs in the first place!

Cloud providers offer the ability to enforce hypervisor affinity rules, although they may use more user-friendly terminology, such as VM-host affinity. As you plan a cloud migration, take note of VMs that are redundant or that appear to serve the same purpose. Chances are that you have hypervisor affinity rules in place that you'll want to re-create in the cloud.

Validating and Preparing for the Move to the Cloud

To perform a successful migration to the cloud, it is important to bring together all the interested parties and stakeholders. The traditional IT groups such as development, operations, OSs, storage, networking, and security will be integral parts of the migration teams. Non-IT groups, such as finance and legal, will need to be involved, because cloud computing can significantly change the cost and accounting models of departments and possibly the entire organization. In the data center model, a healthy portion of the budget goes to the up-front expenses of purchasing physical infrastructure such as servers and networking equipment. Accountants call these capital expenditures (capex for short). When moving to the cloud, these huge capital outlays aren't necessary. Instead of spending hundreds of thousands or more on IT infrastructure, you pay the cloud provider a monthly fee based on usage. This expense is ongoing for as long as you use the services, but the amount you have to fork over up front is much less. This is called an operational expenditure (opex) model. In essence, instead of buying or leasing your own equipment, you're leasing the *use* of the cloud provider's equipment.

If the company does not have the in-house expertise with cloud migrations, it may be advised to enlist the skills of companies specializing in assisting companies moving to the cloud. Also, especially with larger projects, a project manager should be assigned to track and manage the undertaking.

Once you identify the pieces that need to move to the cloud, you need to decide what cloud delivery model you want to use. If you're hosting applications on-prem, do you continue to host the same applications in the cloud using an IaaS model? Or do you offload some of the responsibility to the cloud provider by using a PaaS model? For example, you may have a custom application written in Python. You can continue to run it in VMs as you always have, but just in the cloud using an IaaS model. Or you may instead run it in the cloud without having to bother with the VMs—the PaaS model.

Also, you may decide that now's a good time to ditch some of your current applications and use an SaaS model where the cloud provider manages the back-end IT infrastructure. With this information, you can evaluate the many different cloud company's service offerings in the marketplace.

A common practice is to determine what less critical or low-risk applications could be good candidates to move to the cloud. These applications can be used as a validation or proof-of-concept project for your company.

As part of the preparation, keep the finance group involved from the start of the project. IT expenses and budgeting often are a significant expense for any company, from the smallest local mom-and-pop shop to multibillion conglomerates. The cloud computing pay-as-you-go utility cost models shift the expenses away from the large up-front capital expenditures of equipment, services, and software. Cloud computing requires little, if any, up-front capital costs, and costs are operational based on usage.

Choosing Elements and Objects in the Cloud

Before making plans to migrate anything to the cloud, it's important to decide whether you want to use the IaaS, PaaS, or SaaS model, or any combination thereof. The model you use will determine the specific cloud building blocks you'll have to put together. As you already know, IaaS leaves much of the work in your hands, making migrations more flexible but more complicated. On the other end of the spectrum, the SaaS model makes migrations quicker and easier, but at the cost of giving up control over the infrastructure.

Once you decide on the model, identify what services and capabilities are available in the cloud that fit your needs and requirements. As service providers have expanded their offerings and capabilities, understanding all your options has become almost overwhelming. Some of the largest public cloud companies have more than a thousand objects and services to choose from, with more being added regularly.

To give you an idea of the plethora of options, let's start with the IaaS model. When it comes to virtual servers, there are a variety of prebuilt OSs to choose from. On the virtual hardware side, you can choose CPU and GPU power, memory, storage volumes, and network I/O. On the network side, you have load balancing, DNS and DHCP services, routing, firewalls, network address translation (NAT), and more. For storage, most cloud companies offer block, object, and file storage.

When it comes to PaaS and SaaS models, your options are naturally more limited because the provider takes on more responsibility for the back-end infrastructure. However, it's not unusual for a provider to offer different services that seem to be very similar to one another. For example, cloud providers offer a variety of managed database services for different types of databases. Likewise, they may offer messaging or email services that are compatible with different on-prem solutions. And of course, PaaS services and tools for developing and deploying applications are limited only by the vast number of programming languages out there.

Internet of Things

The ubiquitous access and elasticity enabled by the cloud has birthed a phenomenon called the *Internet of Things (IoT)*. IoT describes the explosion of small, purpose-built devices that typically collect data and send it to a central location for processing. Some examples of IoT devices include temperature sensors, remote-controlled thermostats, and electronic buttons you push to order a new bag of kitty litter effortlessly. Some cloud providers sell such devices that you can program and integrate with the provider's IoT services.

Machine Learning/Artificial Intelligence (AI)

Machine learning/artificial intelligence (ML/AI) is fundamentally concerned with finding patterns in data. The popularity of ML/AI is growing rapidly because of its ability to make predictions and classify or label data in datasets that are too large to work with manually.

With all of the hype surrounding ML/AI, it's important to understand what it *can't* do. It can't autonomously write a coherent novel. It can't predict tomorrow's winning lottery numbers. In fact, the applications of ML/AI are much more limited than many assume. The capabilities of ML/AI are limited to three things:

- Making predictions—for example, forecasting the weather or calculating the arrival time to a destination
- Identifying patterns—Recognizing objects in images and classifying them according to their contents
- Identifying anomalies—Detecting fraud or hacking attempts

ML/AI requires significant human input, and some applications require more human input than others. In fact, ML/AI is broken down into learning models based on the degree of human input required for them to operate—supervised and unsupervised.

Supervised Learning

Supervised learning applications include predictions and image or voice recognition. It requires creating or "training" an ML/AI model by feeding sample data into it. For example, if you want to create an application that identifies your voice in an audio file, you'd train the model using some quality samples of your voice. Subsequently, the model should be able to identify your voice accurately most of the time.

The accuracy of models depends on the size and quality of the training dataset. In terms of percentage, accuracy can be as high as the upper 90s, but rarely higher.

Unsupervised Learning

Unsupervised learning applications include anomaly detection and automatic classification. Unlike the supervised learning model, the unsupervised learning model requires no manual training. Instead, unsupervised learning automatically finds patterns in a dataset and creates groupings or clusters based on those patterns. For example, given a collection of potato pictures, an unsupervised ML/AI algorithm might classify them based on size, color, and the presence or absence of eyes.

It's important to understand the limitations of this approach. The algorithm doesn't know that it's looking at potatoes, so it might classify a bean as a potato if they look similar enough. All that the algorithm does is to label the pictures based on similarities. If you want your application to be able to take a collection of random images and identify all the potatoes, you'll have to use a supervised learning model.

Another interesting unsupervised learning application is fraud detection. If you have a credit or debit card, there's a good chance that your financial institution is using ML/AI for this purpose. It works by analyzing how you normally use your card—purchase amounts, the types of merchandise or services you purchase, and so on—to establish a baseline pattern. If your purchase activity deviates too much from this pattern, the algorithm registers it as an anomaly, probably generating a call from your financial institution.

If this sounds similar to the supervised learning model, it's because it is, but with one big difference—the model is always retraining itself with new data. To better understand this, imagine that you frequent a particular coffee shop and use your card to buy coffee. One day, the coffee shop closes its doors for good, and you end up going to another shop. This may initially register as a slight anomaly, but as you continue visiting the new coffee shop, that becomes your new baseline. The model is continually retraining itself on what your normal card usage looks like.

To put a finer point on it, in supervised learning the algorithm is looking for things it *does* recognize. In unsupervised learning, the algorithm is looking for things it *doesn't* recognize.

Creating and Validating a Cloud Deployment

In this section, I will go deeper into the technology architectures and processes that you'll find in the world of cloud computing. It is important to see how much of an effect virtualization has had in the creation of cloud computing and to understand the process of taking physical hardware resources, virtualizing them, and then assigning these resources to systems and services running in the virtualized data center.

The Cloud Shared Resource Pooling Model

Every cloud service depends on resource pooling. *Resource pooling* is when the cloud service provider virtualizes physical resources into a group, or pool, and makes these pooled resources available to customers. The underlying physical resources are then dynamically allocated and reallocated as the demand requires. Recall the NIST definition of cloud computing as "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources . . ." Resource pooling is a defining feature of the cloud.

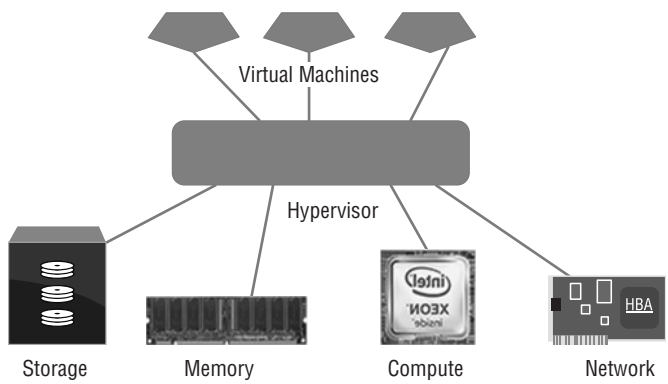
Resource pooling hides the physical hardware from the customer and allows many customers to share resources such as storage, compute power, and network bandwidth. This concept is also called multitenancy. We'll look at some examples of these pooled resources in the following sections.

Compute Pools

In the cloud provider's data center, there are numerous physical servers that each run a hypervisor. When you provision a VM in the cloud, the provider's cloud *orchestration platform* selects an available physical server to host your VM. The important point here is that it doesn't matter which particular host you get. Your VM will run the same regardless of the host. In fact, if you were to stop a running VM and then restart it, it would likely run on a completely different host, and you wouldn't be able to tell the difference. The physical servers that the provider offers up for your VMs to run on are part of a singular *compute pool*. By pooling physical servers in this way, cloud providers can flexibly dispense computing power to multiple customers. As more customers enter the cloud, the provider just has to add more servers to keep up with demand.

Drilling down a bit into the technical details, the hypervisor on each host will virtualize the physical server resources and make them available to the VM for consumption. Multiple VMs can run on a single physical host, so one of the hypervisor's jobs is to allow the VMs to share the host's resources while remaining isolated from one another so that one VM can't read the memory used by another VM, for instance. Figure 1.14 shows this relationship between the virtual machines and the hardware resources.

FIGURE 1.14 Shared resource pooling



The hypervisor's job is to virtualize the host's CPUs, memory, network interfaces, and—if applicable—storage (more on storage in a moment). Let's briefly go over how the hypervisor virtualizes CPU, memory, and network interfaces.

On any given host, there's a good chance that the number of VMs will greatly outnumber the host's physical CPU cores. Therefore, the hypervisor must coordinate or schedule the various threads run by each VM on the host. If the VMs all need to use the processor simultaneously, the hypervisor will figure out how to dole out the scarce CPU resources to the VMs contending for it. This process is automatic, and you'll probably never have to think about it. But you should know about another "affinity" term that's easy to confuse with hypervisor affinity: CPU affinity. *CPU affinity* is the ability to assign a processing thread to a core instead of having the hypervisor dynamically allocate it. A VM can have CPU affinity enabled, and when a processing thread is received by the hypervisor, it will be assigned to the CPU it originally ran on. You're not likely to see it come up on the exam, but just be aware that CPU affinity can and often does result in suboptimal performance, so it's generally best to disable it. A particular CPU assigned to a VM can have a very high utilization rate while another CPU might be sitting idle; affinity will override the hypervisor's CPU selection algorithms and be forced to use the saturated CPU instead of the underutilized one.

Now let's talk about random access memory (RAM). Just as there are a limited number of CPU cores, so there is a finite amount of RAM installed in the physical server. This RAM is virtualized by the hypervisor software into *memory pools* and allocated to virtual machines. When you provision a VM, you choose the amount of RAM to allocate to it. Unlike the CPU, which is shared, RAM is not. Whatever RAM you allocate to a VM is dedicated to that VM, and no other VM can access it. When a VM consumes all of its allocated RAM, it will begin to swap the contents of some of its RAM to storage. This *swap file*, as it is called, will be used as virtual RAM. When configuring a VM, be sure to allocate enough storage space for the swap file, and keep in mind that the storage latency of the swap file will have a negative impact on the performance of the VM.

Thus far, we've discussed compute pools from the perspective of an IaaS model. But how do compute pools work with PaaS or SaaS models? Behind the scenes, almost everything's the same. What's different is that in the PaaS and SaaS models, the cloud provider runs a user-friendly interface atop the underlying compute infrastructure. For example, if the cloud provider is offering hosted email as a service, that email system gets its computing power from the same compute pools that power the IaaS infrastructure. In fact, every service under the PaaS or SaaS model that the provider offers probably runs directly on the same IaaS infrastructure that we've been discussing. In other words, cloud providers don't reinvent the wheel for every service that they provide. They build the compute infrastructure to provide the compute pools, and everything else uses those.

Network Pools

Cloud providers also virtualize and pool network resources. If you're not familiar with the details of networking, what happens behind the scenes can be a bit difficult to grasp, so we'll start in familiar territory.

The term *network* is a loaded term because its meaning varies with context. Generally, a *network* is the infrastructure that allows communication between computing resources, such

as two servers or an end user and a server. That much you already know, but here's where it gets complicated. In the cloud, there are two different levels of networking:

The Underlay The underlying network (or *underlay*) consists of the physical network infrastructure that the cloud provider completely manages. This is transparent to you, and you have no visibility into it whatsoever.

The Overlay The cloud provider allows customers to create and manage virtual networks that run atop the provider's underlying network. These are sometimes called *overlay* networks or virtual private clouds (VPCs). Virtual networks are what you'll actually work with and connect your cloud resources to. In simple terms, a VPC is a private, software-defined network that exists in the cloud.

A virtual network consists of, at a minimum, a block of private IP addresses to be assigned to VMs and other network resources, such as DNS and DHCP servers. A virtual network can span multiple physical hosts—a VM running on one host can communicate with another VM running on a different host, as if they were on the same subnet. Naturally, you can connect a virtual network to an external network such as the Internet or a corporate network via a VPN.

It's important to understand that networking in the cloud operates quite differently than what you'll find in a traditional data center. In a data center, a VM's *virtual network interface card (vNIC)* typically connects to a *virtual switch* (vSwitch) that's associated with one or more physical network interfaces on the host. Each VM can be connected to a different virtual LAN (VLAN) for traffic segmentation. In this virtual switching paradigm, configuring VM networks is a mostly manual task, so the network configuration remains relatively fixed. Without getting into too many details, this inflexibility is due to the limitations of Ethernet. Additionally, such networks are limited to a maximum of about 1 million devices, which is more than enough for a data center but woefully lacking for a cloud provider that may need to support hundreds of millions of VMs.

If you have an application that depends on Ethernet broadcast functionality, it probably won't work in the cloud. Ethernet broadcasts pose a hindrance to scalability in the cloud, so cloud providers generally don't support them.

When you create a virtual network in the cloud, the cloud provider's orchestration platform handles the details of the behind-the-scenes connectivity. For example, suppose that you're running two VMs connected to the same virtual network. These VMs may be on different physical hosts and even in different geographic locations, but they can still communicate. The important point here is that you don't need to know anything about the underlying network infrastructure. All you have to do is to create and configure a virtual network and connect your resources to it, and you're good to go. Just as the provider's proprietary orchestration software automatically picks what server to run your VMs on, so too it dynamically handles connectivity among devices on your virtual networks. This is another example of

that "minimal management effort or service provider interaction" qualification that defines cloud computing.

Storage Pools

When you think of the term *storage*, you might think of files on a drive. That's one example of storage that most people are familiar with because it's how our daily-use computers store files. But in the cloud, you'll encounter three different types of storage:

- Block storage
- Object/file storage
- Filesystem storage

Regardless of the storage type, in the cloud data is redundantly replicated across multiple physical devices that compose a *storage pool*. Having data distributed across a storage pool allows the cloud provider to achieve exceptionally high read and write speeds.

Block Storage *Block storage* is designed to mimic a drive by storing data in the same way that a drive does. The virtual disks' (vDisks) VMs used to store data are backed up by block storage. In the data center, a *storage area network (SAN)* device is what provides block storage. A SAN consists of a collection of redundant drives (either spinning disks or SSDs) that store data in blocks, hence the term *block storage*. Even the drive in your personal computer stores data in this way.

In the cloud, because VMs may move from host to host, the VM's persistent storage is not attached to the host. In cloud provider terminology, block storage may be called *elastic block storage* or *block blob storage*. Notice how the terminology hints at the flexible and abstract nature of the storage.

To allocate space from a block storage pool, you create a volume that you can then attach to a VM as a virtual disk. Just as when you provision a VM and the cloud provider dynamically and automatically finds a host to run it on, so too the provider selects some SANs from the storage pool to hold the volume. One big advantage of storage pooling is that your data is stored redundantly on multiple physical SANs; if one fails, your VM can keep on running with no loss of data.

As you gain experience with the cloud, you'll notice that as a general rule, the more block storage you allocate to a vDisk, the better the performance you get. The reason for this is that allocating more storage means that storage is spread across more physical drives operating in parallel.

Although block storage is typically thought of as an IaaS analog to the SAN, it does come into play in some PaaS services. Some managed SQL database services make use of block storage. Although in most cases the cloud provider manages it, you still may get to choose some aspects of the volume such as speed and size.

Although the storage systems are generally external from the physical servers themselves, some cloud providers do let your VM use locally attached drives for temporary storage, like swap files or caching.

Object/File Storage As the name suggests, *object/file storage* is designed just to store files. You can use object storage to store any file of virtually any size. It's often used for file backups, but it can also be used to store web assets such as images, video, HTML files, and PDF documents. Object/file storage is intended to store files that don't change frequently, so although you can use it to store a database backup, it's not appropriate for storing a live database that's regularly written to.

The cloud provider will usually offer multiple interfaces to upload or download files. For example, they may allow you to transfer files via a web interface, command-line tool, or API. They may allow you to use HTTP to download files, effectively letting you use the object store as a static web server.

One particularly important use of object storage is storing snapshots of elastic block storage volumes. When you take a snapshot of a VM's volume for backup or cloning, it may be stored in object storage, depending on the cloud provider.

Filesystem Storage Filesystem storage is similar to object/file storage in that both are for storing files. However, *filesystem storage* is meant for files that change frequently, like a live database. Filesystem storage is a popular choice for applications that need shared read/write access to the same files.

The way that you interact with filesystem storage differs from object storage. You access filesystem storage via standardized network filesystem protocols, such as Network File System (NFS) or Server Message Block (SMB). To do this, you must configure your OS to mount the networked filesystem as a volume. The OS can then read and write files, just as it would on any other attached filesystem.

In the data center environment, it's common to have file servers dedicated to offering file storage via SMB or NFS, typically for storing user documents. You could still build your own file servers in the cloud, but most cloud providers offer this as a service under the SaaS model.

Organizational Uses of the Cloud

In the cloud, just as in the data center, you don't simply deploy your applications and forget about them. Applications have to be updated or reconfigured, and you'll probably end up adding or retiring applications over time. These operations always carry the risk of breaking working things, something that can be detrimental to your organization. Therefore, it's a best practice to test changes before rolling them out and committing to them. To achieve this, it's common to separate operations into four isolated sections of the cloud:

- Production
- Quality assurance/test
- Staging
- Development

As we walk through these categories, keep in mind that they can be broken down differently according to the needs of the organization. The point here is to protect the organization from data loss and downtime caused by changes.

Production

Production environments host the live applications that the organization uses in its normal course of business. These include email, customer-facing services, and any other line-of-business applications.

The use of multiple production environments can become important during the rollout of updates. Depending on the application, you may want to release updates only to a portion of users. For example, if you're adding a new feature to a web application, you can use load balancing to direct a small portion of users (say 10 percent) to the updated version while everyone else uses the old version. If there's an unforeseen problem with the updated version, it impacts only a fraction of your users.

Of course, this is a simple example, and this approach won't necessarily work in more complex cases. If the application update causes irreversible changes to a huge database, a more cautious approach is needed. This is where it may be necessary to replicate the full production environment and test the update there. If all goes well, you cut everyone over to the updated environment. If things don't go so well, your existing production environment remains intact and functional.

Quality Assurance/Test

Quality assurance (QA)/test environments are used for the testing of software updates or new applications. QA/test environments may closely mirror production environments to ensure the accuracy of test results. To achieve this parity, you may need to copy over production data to the QA/test environment, but the environments still remain carefully separated. We'll discuss some testing methods later in the chapter.

When sensitive data exists in the production environment, doing a verbatim copy to QA/test may not be feasible. It may be necessary to use dummy data that mimics the production data.

Staging

Staging environments are used for building out a system prior to releasing it to production. In reality, a staging environment is just a preproduction environment.

Development

Development environments are typically used by software developers for creating new applications. Organizations that don't develop their own software may not need a dedicated development environment.

Scaling and Architecting Cloud Systems Based on Requirements

One of the prime advantages of cloud computing is that it enables *on-demand computing*, allowing you to deploy and pay for the computing capacity that you are actually using. This is an attractive alternative to having to absorb the costs of servers sitting on standby to address any bursts or cyclical higher compute requirements, such as end-of-month processing or holiday sales loads if, for example, you are a retailer.

Autoscaling is a cloud feature that automatically adds and removes resources based on demand. By paying only for what you need when you need it, you can take advantage of the immense computing power of the cloud without having to pay for servers that are just sitting idle during times of low demand.

For example, let's look at a small sporting goods retailer that uses a public cloud provider to host its e-commerce website. During normal operations, the retailer runs and pays for three web servers. During times of high demand, autoscaling will provision additional web servers to match the increased load. For example, the retailer may decide to run a TV commercial on a Saturday afternoon televised game. After the commercial airs, the website experiences a huge traffic spike and an increase of online orders. Once the load subsides to normal levels, autoscaling terminates the additional web servers so that the retailer doesn't have to keep paying for them when they're not needed. This works well because the retailer can match the load on the website with the needed amount of computing, memory, storage, and other back-end resources in the cloud. Combining this pay-as-you-go model with autoscaling maximizes cost efficiency because you don't have to expend money to purchase the hardware for any peak loads or future growth. Autoscaling will just provision more capacity when needed. With automation and rapid provisioning, adding capacity can be as simple as a few clicks in a console, and the resources are immediately deployed!

Contrast this scenario with what would happen without autoscaling. If the retailer were stuck with only three web servers, during the traffic spike the servers might slow down or crash. Adding more servers would be a manual, expensive, and time-consuming process that even in a best-case scenario would take several minutes to complete. By that time, the damage would have already been done.

Understanding Cloud Performance

Cloud performance encompasses all of the individual capabilities of the various components as well as how they interoperate. The performance you are able to achieve with your deployment is a combination of the capabilities and architecture of the cloud service provider and how you design and implement your operations.

Ongoing network monitoring and management allow you to measure and view an almost unlimited number of cloud objects. If any parameter extends beyond your predefined boundaries, alarms can be generated to alert operations and even to run automated scripts to remedy the issue. Here are just a few of the things you may want to monitor:

- Database performance
- Bandwidth usage

- Network latency
- Storage I/O operations per second (IOPS)
- Memory utilization

Delivering High Availability Operations

By implementing a well-architected network using best design practices, and by selecting a capable cloud service provider, you can achieve high availability operations. You and the cloud provider share responsibility for achieving high availability for your applications running in the cloud.

The cloud provider must engineer its data centers for redundant power, cooling, and network systems, and create an architecture for rapid failover if a data center goes offline for whatever reason. As we discussed with computing, network, and storage pools, the cloud provider is responsible for ensuring high availability of these pools, which means that they're also responsible for ensuring redundancy of the physical components that compose these pools.

It's your responsibility as the cloud customer to engineer and deploy your applications with the appropriate levels of availability based on your requirements and budgetary constraints. This means using different regions and availability zones to eliminate any single point of failure. It also means taking advantage of load balancing and autoscaling to route around and recover from individual component failures, like an application server or database server going offline.

Managing and Connecting to Your Cloud Resources

By definition, your cloud resources are off-premises. This raises the question of how to connect to the remote cloud data center in a way that is both reliable and secure. You'll look at this question in this chapter. Finally, you'll learn about firewalls, a mainstay of network security, and you'll see the role of firewalls in cloud management deployments.

Managing Your Cloud Resources

It's instructive to note the distinction between managing your cloud resources and using them. Managing your cloud resources includes provisioning VMs, deploying an application, or subscribing to an SaaS service such as hosted email. You'll typically manage your cloud services in one of three ways:

- Web management interface
- Command-line interface (CLI)
- APIs and SDKs

Web Management Interface

When getting started with the cloud, one of the first ways you'll manage your cloud resources is via a web interface the cloud provider offers. You'll securely access the *web*

management interface over the Internet. Here are a few examples of what you can do with a typical cloud provider web interface:

IaaS: Provision VMs, create elastic block storage volumes, create virtual networks

PaaS: Upload and execute an application written in Python, deploy a web application from a Git repository

SaaS: Send and receive email, create and collaborate on documents

Note that when it comes to the PaaS and SaaS side of things, there's considerable overlap between managing a service and using it.

Command-Line Interface, APIs, and SDKs

Cloud providers offer one or more command-line interfaces to allow scripted/programmatic management of your cloud resources. The command-line interface is geared toward sysadmins who want to perform routine management tasks without having to log in and click around a web interface.

Command-line interfaces work by using the cloud provider's APIs. In simple terms, the API allows you to manage your cloud resources programmatically. In contrast to a web management interface, in which you're clicking and typing, an API endpoint is a web service that listens for specially structured requests. Cloud provider API endpoints are usually open to the Internet, encrypted using Transport Layer Security (TLS), and require some form of authentication.

Cloud providers offer *software development kits (SDKs)* for software developers who want to write applications that integrate with the cloud. SDKs take care of the details of communicating with the API endpoints so that developers can focus on writing their application.

Connecting to Your Cloud Resources

How you connect to your cloud resources depends on how you set them up. As I alluded to earlier, cloud resources that you create are not necessarily reachable via the Internet by default. There are three ways that you can connect to your resources:

- Internet
- VPN access
- Dedicated private connections

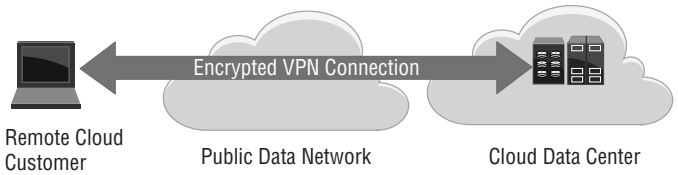
Internet

If you're hosting an application that needs to be reachable anytime and anywhere, you'll likely open it up to the Internet. If a resource is open to the Internet, it will have a publicly routable Internet IP address. This is typically going to be a web application, but it doesn't have to be. Although anywhere, anytime access can be a great benefit, keep in mind that traffic traversing the Internet is subject to high, unpredictable latency.

VPN Access

A *virtual private network (VPN)* allows for secure and usually encrypted connections over an insecure network (like the Internet), as shown in Figure 1.15. Usually, a VPN connection is set up between a customer-owned device deployment and the cloud. VPNs are appropriate for applications that do not need anywhere, anytime access. Organizations often use VPNs to connect cloud resources to offices and data centers.

FIGURE 1.15 Remote VPN access to a data center



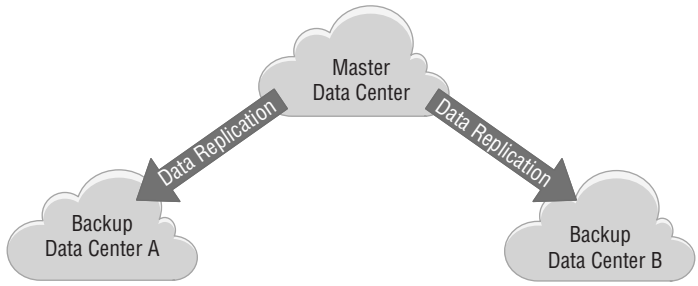
Dedicated Private Connections

Cloud providers offer connections to their data centers via private leased lines instead of the Internet. These connections offer dedicated bandwidth and predictable latency—something you can't get with Internet or VPN access. *Dedicated private connections* do not traverse the Internet, nor do they offer built-in encryption. Keep in mind that dedicated connections don't usually provide Internet access. For that, you'll need a separate Internet connection.

Is My Data Safe? (Replication and Synchronization)

Replication is the transfer and synchronization of data between computing or storage resources, and typically between multiple regions or data centers, as illustrated in Figure 1.16. For disaster recovery purposes and data security, your data must be transferred, or replicated, between data centers. Remote copies of data have traditionally been implemented with storage backup applications. However, with the virtualization of servers in the cloud, you can easily replicate complete VM instances, which allows you to replicate complete server instances, with all of the applications, service packs, and content, to a remote facility.

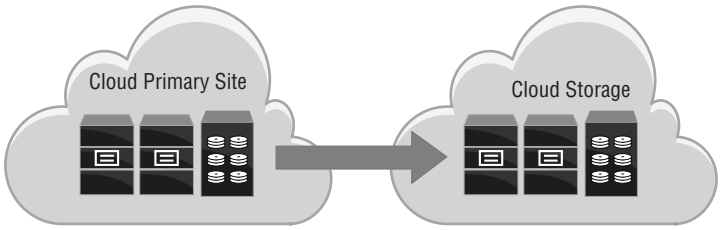
FIGURE 1.16 Site-to-site replication of data



Applications such as databases have built-in replication processes that can be utilized based on your requirements. Also, many cloud service offerings can include data replication as a built-in feature or as a chargeable option.

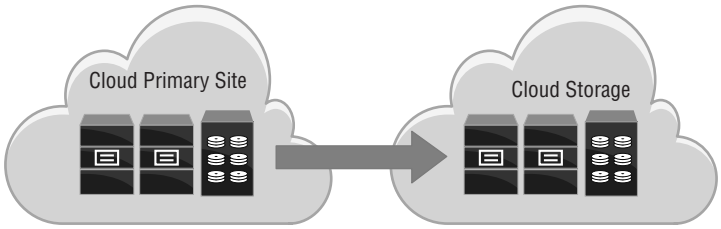
Synchronous replication is the process of replicating data in real time from the primary storage system to a remote facility, as shown in Figure 1.17. Synchronous replication allows you to store current data at a remote location from the primary data center that can be brought online with a short recovery time and limited loss of data. Relational database systems offer synchronous replication along with automatic failover to achieve high availability.

FIGURE 1.17 Synchronous replication



With *asynchronous replication*, the data is first written to the primary storage system in the primary storage facility or cloud location. After the data is stored, it is then copied to a remote location on a delayed or scheduled basis, as shown in Figure 1.18.

FIGURE 1.18 Asynchronous replication



One common use case for asynchronous replication involves taking scheduled snapshots of VM storage volumes and storing those snapshots offline. The snapshots may also be replicated to a remote location for safekeeping. If you ever need to restore the VM, you can do so from the snapshot.

Another example of asynchronous replication is the creation of database read replicas. When an organization needs to run intensive, complex reports against a database, it can tax the database server and slow it down. Rather than taxing the primary database server, which

might be performing critical business functions, you can asynchronously replicate the data to a read replica and then run your reports against the replica.

Asynchronous replication can be more cost effective than implementing a synchronous replication offering. Cloud providers often charge for data transfer between regions or availability zones. Because asynchronous replication is not in real time, there's typically less data to transfer.

Understanding Load Balancers

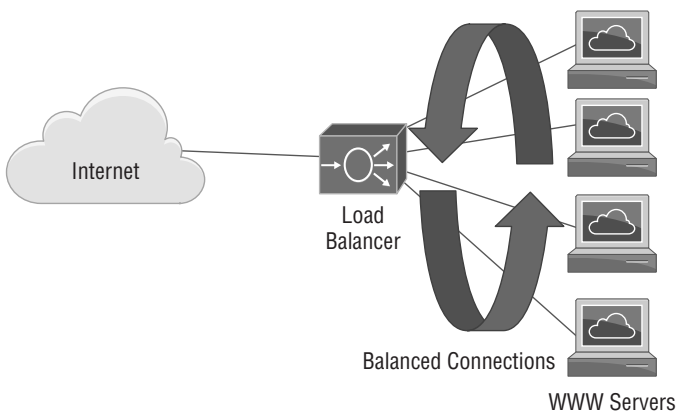
Loose coupling (also called decoupling) is a design principle in which application components are broken up in such a way that they can run on different servers. With this approach, redundant application components can be deployed to achieve high availability and scalability.

Let's take a look at a familiar example. Most database-backed web applications decouple the web component from the database so that they can run on separate servers. This makes it possible to run redundant web servers for scaling and high availability.

But loose coupling introduces a new challenge: If there are multiple web servers that users can access, how do you distribute traffic among them? And what if one of the servers fails? The answer is *load balancing*. A load balancer accepts connections from users and distributes those connections to web servers, typically in a round-robin fashion. When a load balancer sits in front of web servers, users connect to an IP address of the load balancer instead of an IP address of one of the web servers.

Other load balancing functions may include SSL/TLS termination, compression, and session tracking. Load balancers can integrate with autoscaling and server health checks so that if a web server becomes unreachable, the load balancer will detect the failure and trigger an automatic replacement or recovery action. With load balancing, you can effortlessly achieve redundancy and scalability, as shown in Figure 1.19.

FIGURE 1.19 Load balancing web servers



Cloud Testing

As you progress through this book, I will include information on the testing and validations that are required to ensure that changes and ongoing operations are valid and working as expected. In this chapter, you'll be introduced to three validations. Vulnerability and penetration tests are security-related, and I will expand my discussion of them throughout this book. You'll be introduced to load testing to ensure that your application works as expected when it is deployed into a heavily used production network.

Vulnerability Testing

Vulnerability testing is used to find objects in your cloud deployment that can be exploited or that are potential security threats. The *vulnerability scanner* is an application that has a database of known exploits and runs them against your deployment to see whether your cloud deployment may be susceptible or have security holes that need to be remediated. The scanner will detect and report on weaknesses in your cloud deployment. For example, if you're running an older version of a content management system (CMS) that's easily hacked, a vulnerability scan can alert you to this before you become a victim.

Penetration Testing

Penetration testing is the process of trying to exploit vulnerabilities that exist in your infrastructure. Pentesting is usually performed from outside your cloud deployment to assess the ability to access systems into your cloud from, for example, the Internet. Cloud providers have strict rules for how and when you can perform penetration testing, typically requiring advance permission and coordination with the provider. Some examples of penetration testing include trying default or easy-to-guess usernames and passwords and looking for open Transmission Control Protocol/User Datagram Protocol (TCP/UDP) ports.

Performance Testing

Performance testing (sometimes called *load testing*) puts a demand or load on your application or compute system and measures the response. By performing load testing, you can determine how your applications and cloud deployment can be expected to perform in times of heavy production usage. Load testing helps you determine a system's behavior under both normal and anticipated peak load conditions. All systems will fail at some point when under heavy loads, and by performing tests, you can identify and rectify any issues on your design.

Regression Testing

Frequent software updates are a part of the IT landscape. When you upgrade software to a new version, there's always a chance that a previously working function will break. This phenomenon is called a *regression*. *Regression testing* is designed to identify these regressions so that you can decide whether or not to update.

There was a time when organizations would postpone software updates because they would routinely break things and create troubleshooting headaches for everyone. Although this isn't as much of a problem anymore, it does happen from time to time, so it's important to perform regression testing in a controlled test environment prior to rolling out major software updates.

Functional Testing

Functional testing checks the functionality of software against a set of specifications that defines what the software must (or must not) do. In short, functional testing checks whether the software is capable of doing what you want it to do.

Usability Testing

Usability testing considers how easy or difficult it is for end users to use a system. A piece of software might pass all other tests, but the user may still find it difficult, confusing, or frustrating. Such flaws are usually not technical flaws in the code or architecture of a system, but rather flaws in the user interface or a process the user has to follow. Usability testing is designed to catch such flaws early.

In its simplest form, usability testing consists of having a user attempt to complete a specified task. Usability testing is, in a sense, a more subjective version of functional testing. Functional testing is designed to test whether the *software* performs a specified function. Usability testing tests whether the *user* can use the software.

Verifying System Requirements

After you have completed your assessments and needs analysis, you'll have then defined your requirements and which cloud service and deployment models best meet them. The next step is to select a pilot application to migrate to the cloud from your existing data center.

Prior to performing the migration, the engineering team should sit down and review the complete design, from the application, configuration, hardware, and networking to the storage and security. As part of this verification, it is helpful to stage the system in the cloud as a proof-of-concept design. This allows everyone to test the systems and configuration in a cloud environment prior to going live.

Correct Scaling for Your Requirements

The ability of the cloud to scale resources up or down rapidly to match demand is called *elasticity*. For IaaS services, this can be done automatically as needed using autoscaling. This allows cloud consumers to scale up automatically as their workload increases and then have the cloud remove the services after the workload subsides. For SaaS and PaaS services, dynamic allocation of resources occurs automatically and is handled by the cloud provider. (Later in the chapter, we'll discuss the division of responsibilities between you and the provider.) With elastic computing, there is no longer any need to deploy servers and storage systems designed to handle peak loads—servers and systems that may otherwise sit idle during normal operations. Now you can scale the cloud infrastructure to the normal load and automatically expand as needed when the occasion arises.

On-demand cloud services allow the cloud customer to create instantly additional servers, storage, processing power, or any other services as required. On-demand allows customers to consume cloud services only as needed and scale back when they are no longer required. For

example, if a developer needs to provision a database and application server for testing, they can quickly provision these servers and deprovision them once they're no longer needed. This is also referred to as a *just-in-time* service because it allows cloud services to be added and removed as needed.

Pay as you grow (PAYG) is like a basic utility, such as power or water, where you pay for only what you use. This is very cost effective because there are minimal up-front costs, and the ongoing costs track your actual consumption of the service. The elasticity of the cloud lets you add resources on-demand, so there's no need to overprovision for future growth. With a normal data center operation, the computing must be overprovisioned to take into account peak usage or future requirements that may never be needed.

Making Sure the Cloud Is Always Available

In this section, you'll become familiar with common deployment architectures used by many of the leading cloud providers to address availability, survivability, and resilience in their services offerings.

Regions

Major cloud providers partition their operations into regions for fault tolerance and to offer localized performance advantages. A *region* is not a monolithic data center but rather a geographical area of presence that usually falls within a defined political boundary, such as a state or country. For example, a cloud company may offer regions throughout the world, as shown in Figure 1.20. They may have regions in Sydney and Tokyo in the Asia Pacific region, and in Europe there may be regions called London and Oslo. In North America there could be regions in Boston, Ottawa, Austin, and San Jose.

FIGURE 1.20 Cloud regions

All of the regions are interconnected to each other and the Internet with high-speed optical networks but are isolated from each other, so if there is an outage in one region, it should not affect the operations of other regions.

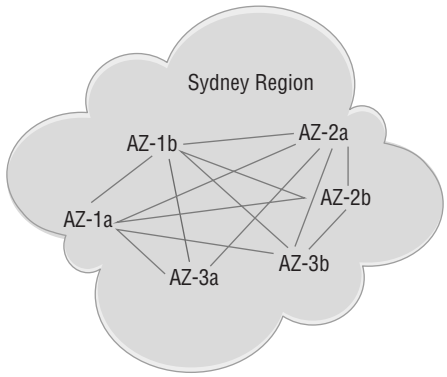
Generally, data and resources in one region aren't replicated to any other regions unless you specifically configure such replication to occur. One of the reasons for this is to address regulatory and compliance issues that require data to remain in its country of origin.

When you deploy your cloud operations, you'll be given a choice of what region you want to use. Also, for a global presence and to reduce network delays, the cloud customer can choose to replicate operations in multiple regions around the world.

Availability Zones

Regions are divided into one or more *availability zones (AZs)*. Each region will usually have two or more availability zones for fault tolerance. AZs almost always correspond to individual data centers. Each AZ has its own redundant power and network connections. Within a region, AZs may be located a greater distance apart, especially if the region is in an area prone to natural disasters such as hurricanes or earthquakes. When running redundant VMs in the cloud, it's a best practice to spread them across AZs to take advantage of the resiliency that they provide. Figure 1.21 illustrates the concept of availability zones.

FIGURE 1.21 Availability zones



Cluster Placement

As I alluded to earlier, you can choose to run VMs on different virtualization hosts for redundancy in case one host fails. However, there are times when you want VMs to run on the same host, such as when the VMs need extremely low-latency network connectivity to one another. To achieve this, you would group these VMs into the same cluster and implement a cluster placement rule to ensure that the VMs always run on the same host. If this sounds familiar, it's because this is the same principle you read about in the discussion of hypervisor affinity rules.

This approach obviously isn't resilient if that host fails, so what you may also do is to create multiple clusters of redundant VMs that run on different hosts, and perhaps even in different AZs.

Remote Management of VMs

In this section, you'll learn about remote access techniques and look at what tools are available to manage and monitor your VMs. Remember that you don't have physical access to the cloud provider's data centers, so remote access is your only option for managing your servers. Furthermore, the cloud provider will not give you direct access to the hypervisor, which is typically going to be proprietary. This is unlike a traditional data center environment in which you can install a hypervisor management application on a workstation and fully configure and manage the hypervisor, as Figure 1.22 and Figure 1.23 illustrate.

FIGURE 1.22 Local computer running the hypervisor management application

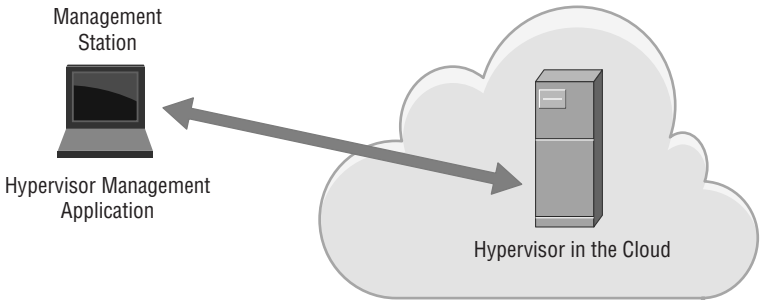
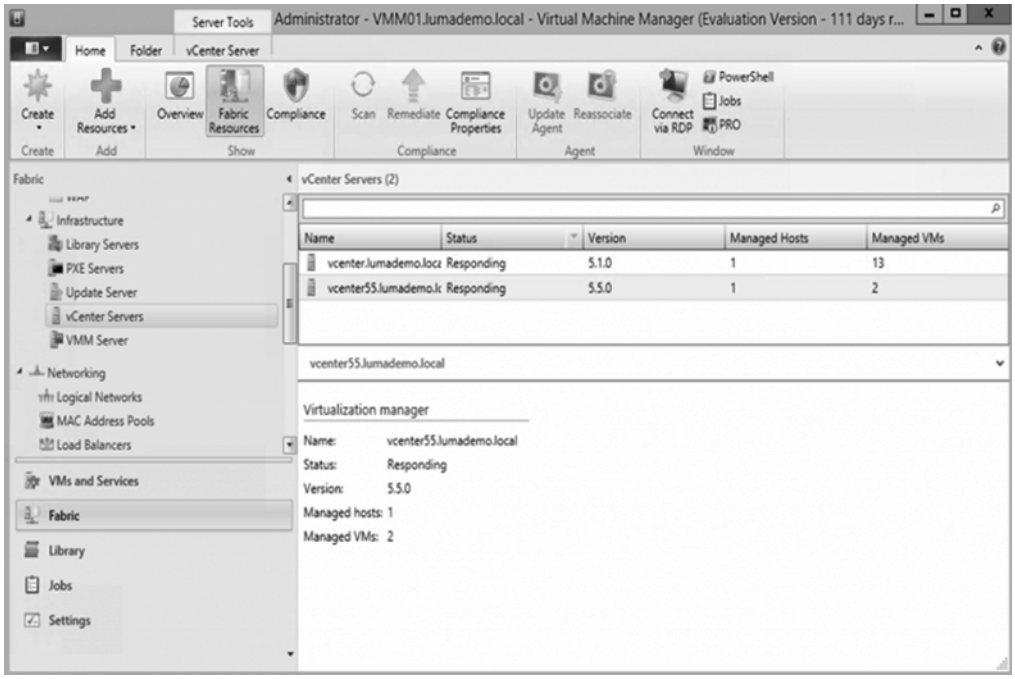


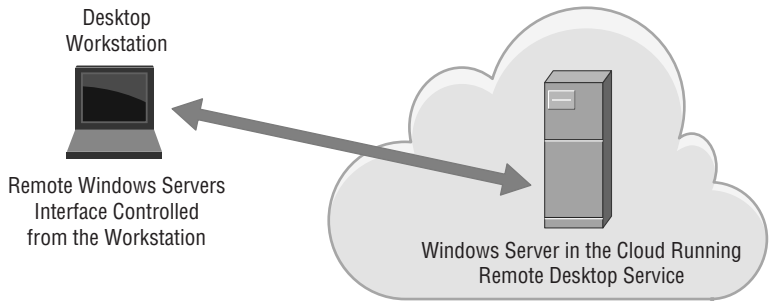
FIGURE 1.23 Remote hypervisor management application



As we've discussed, your options for managing your VMs and other cloud resources are limited to the web management interface, command-line tools, and APIs that the provider gives you. But once you provision a VM, managing the OS and applications running on it is a much more familiar task. You'll manage them in almost the exact same way as you would in a data center. In fact, the whole reason providers offer IaaS services is to replicate the data center infrastructure closely enough that you can take your existing VMs and migrate them to the cloud with minimal fuss. Let's revisit some of these management options with which you're probably already familiar.

RDP *Remote Desktop Protocol (RDP)* is a proprietary protocol developed by Microsoft to allow remote access to Windows devices, as illustrated in Figure 1.24. RDP is invaluable for managing remote Windows virtual machines, since it allows you to work remotely as if you were locally connected to the server. Microsoft calls the application *Remote Desktop Services*, formerly Terminal Services. The remote desktop application comes preinstalled on all modern versions of Windows. RDP uses the TCP port 3389.

FIGURE 1.24 Local computer running Remote Desktop Services to remotely access a Windows server graphical interface in the cloud

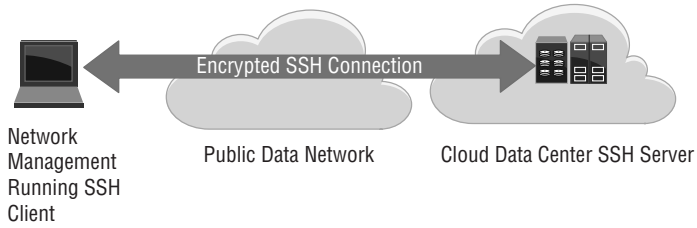


The graphical client will request the name of the remote server in the cloud, and once it's connected, you'll be presented with a standard Windows interface to log in. Then you'll see the standard Windows desktop of the remote server on your local workstation.

SSH The *Secure Shell (SSH)* protocol has largely replaced Telnet as a remote access method. SSH supports encryption, whereas Telnet does not, making Telnet insecure. To use SSH, the SSH service must be enabled on the VM. This is pretty much standard on any Linux distribution, and it can also be installed on Windows devices.

Many SSH clients are available on the market as both commercial software and freeware in the public domain. The SSH client connects over the network using TCP port 22 via an encrypted connection, as shown in Figure 1.25. Once you are connected, you have a command-line interface to manage your cloud services. SSH is a common remote connection method used to configure network devices such as switches and routers.

FIGURE 1.25 Secure Shell encrypted remote access



Monitoring Your Cloud Resources

Just because you have created an operational cloud deployment doesn't mean your work is over! You must continually monitor performance and also make sure that there are no interruptions to services. Fortunately, this function has largely been automated. As you'll learn in later chapters, cloud providers offer ways to collect and analyze performance and health metrics. You can also configure automated responses to various events, such as increased application response time or a VM going down. Also, alerts such as text messages, emails, or calls to other applicators can be defined and sent in response to such events.

Monitoring and *automation* go hand in hand in the cloud. For example, you can use autoscaling to add additional virtual CPUs to a VM if utilization has, for example, exceeded 98 percent for more than 10 minutes. And just like everything else in the cloud, you can configure this automation quickly and adjust it to meet changing requirements.

Writing It All Down (Documentation)

Documentation should be created by the many different teams involved in the cloud deployment, such as the server, virtualization, storage, networking, developer, security, and management teams, as well as the cloud provider.

Once the document is complete, it should be readily accessible, and the procedures to maintain consistency should be clear. A corporate compliance group may be formed to monitor and maintain adherence to the standardization process. Also, since the operations and deployment are constantly evolving and changing, the documentation will need to be constantly modified and updated.

Creating Baselines

Before migrating to the cloud, you should establish *baselines* for various aspects of your current infrastructure. Baselines to collect include CPU, memory, storage utilization, and database query times. Establishing baselines helps you determine how to size your cloud resources. Baselines also help you tune your monitoring. Use your baseline statistics as a reference, and if a metric goes significantly above or below that value, you can configure an alert to fire or autoscaling to add more capacity. The deviation of a metric from a baseline is called *variance*.

Because the metrics you'll be monitoring will likely change quickly, minute by minute, as well as slowly over longer periods of time, you'll want to take baseline samplings for both long-term and short-term intervals. For example, suppose a database server running in your data center averages 70 percent CPU utilization over the course of a day, but it occasionally spikes to 90 percent for about 15 minutes when someone runs a complex query. You don't necessarily want to alert every time the CPU utilization spikes. But you might want to know if the average daily CPU utilization begins creeping up to, say, 80 percent. Therefore, having separate baselines for the long term and the short term gives you a better picture of what's really going on in your environment. Keep in mind that monitoring always requires trial and error.

Your baselines help you form a starting point for determining what the normal metrics are for your environment, but they're not to be taken as gospel, since they will change over time. Also, given the huge number of cloud resources available, there are even more metrics that you can monitor. The key is to focus only on the most impactful metrics.

Shared Responsibility Model

Cloud providers operate under what's called a *shared responsibility model* that defines what you are responsible for and what the provider is responsible for. The model will vary depending on what you are contracting to use and the offerings of the service provider. Let's look at how the division of responsibilities changes with different service models.

IaaS

As the "I" in IaaS suggests, the cloud provider is responsible for the underlying infrastructure they provide as a service—all of the hardware, hypervisors, block storage, and networking. If you're running VMs, you would have responsibility for the operating system and applications, as well as configuration of your virtual network, which includes network access controls and IP routing.

To make it easier to get started, a cloud provider may provide you with a default virtual network for your IaaS resources. Even though such a default network should work right out of the gate, it's still your responsibility to make sure that it's configured properly.

PaaS

With a PaaS service, you're responsible for the applications you choose to run on the service, and the cloud provider would take responsibility for almost everything your application depends on—namely the VMs, OS, and storage. When it comes to the networking, the division of responsibilities may vary. The cloud provider may offer to configure networking for you automatically, giving you unique public and private IP addresses and corresponding DNS entries, or you may provide your own network configuration parameters.

SaaS

When offering Software as a Service, the cloud provider assumes full responsibility for everything, including the software itself. Your responsibility is limited to how you configure and use the software.

Summary

In this introductory chapter, you explored the big picture of cloud computing. You investigated the many types of service models offered in the marketplace today. The core models include IaaS, which strives to emulate the core components of a traditional data center and offer them as a service to facilitate migrations to the cloud. The PaaS and SaaS models extend the benefits of elasticity and pay-as-you-go pricing without the commitment and expertise required to pull off an IaaS deployment. In these models, the cloud provider handles the compute, storage, and network infrastructure.

Cloud delivery models are important to understand for the exam. You looked at the various types of models, including private, public, community, and hybrid clouds, and what the general use cases and differences are for each type of delivery model.

You also learned about the fundamental characteristics of cloud computing. The cloud offers on-demand, self-service provisioning of elastic resources that you pay for on a metered basis. The chapter discussed resource pooling and virtualization of resources such as storage, CPU, memory, storage, and networking.

The chapter also covered how to prepare to migrate your operations to the cloud and how it's important to test systems in the cloud—including sizing, performance, availability, connectivity, data integrity, and others—and that you document the results.

As you read the rest of the book, keep these fundamental concepts in mind, because they provide a structure that you'll build on as you progress on your journey to become Cloud+ certified.

Exam Essentials

Know that cloud computing is similar in operation to a utility. Cloud computing follows the utilities model where a provider will sell computing resources using an as-needed or as-consumed model. This allows a company or individual to pay for only what they use.

Know what cloud computing is. Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (for example, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

Understand the different cloud service models and how to differentiate between them. Cloud service models are characterized by the phrase *as a service* and are accessed by many types of devices, including web browsers, thin clients, and mobile devices. There are three primary service types. Software as a Service, Infrastructure as a Service, and Platform as a Service are the core service offerings. Many cloud service providers offer more descriptive terms in their marketing and sales offerings, including Communications as a Service, Anything as a Service, and Desktop as a Service. However, all of these newer terms fit into the SaaS, IaaS, or PaaS service model. Study the service models and know the differences among IaaS, PaaS, and SaaS as well as the other service models.

Know the primary cloud delivery models. The four primary cloud delivery models are public, private, community, and hybrid clouds. Know what each one is and its function. It is critical that you understand the way cloud services are delivered in the market today and what they offer.

Be able to identify and explain cloud components. Common cloud components include applications, automation, compute, networking, security, and storage.

Know the cloud shared resource pooling model and how it is used. Resource pooling is when the cloud service provider abstracts its physical compute, storage, and networking resources into a group, or pool. The resources from these pools are dynamically allocated to customers on-demand. Resource pooling hides the underlying physical hardware from the customers in such a way that different customers share the underlying infrastructure while their cloud resources remain isolated from each other.

Understand cloud performance components. The performance you are able to achieve with your deployment is a combination of the capabilities and architecture of the cloud service provider and how you design and implement your operations. Some metrics that you'll want to consider are bandwidth usage, network latency, storage I/O operations per second (IOPS), and memory utilization.

Be able to explain how autoscaling works. The ability to automatically and dynamically add additional resources such as storage, CPUs, memory, and even servers is referred to as elasticity. Using autoscaling, this can be done "on the fly" as needed or on a scheduled basis. This allows for cloud consumers to scale up automatically as their workload increases and then have the cloud remove the services after the workload subsides. On-demand cloud services allow the cloud customer to access a self-service portal and instantly create additional servers, storage, processing power, or any other services as required. If the computing workload increases, then additional cloud resources can be created and applied as needed. On-demand allows customers to consume cloud services only as needed and scale back when they are no longer required.

Know what regions and availability zones are. Large cloud operations partition operations into geographical regions for fault tolerance and to offer localized performance advantages. A region is not a monolithic data center but rather a geographical area of presence. The actual data centers in each region are availability zones. Each region will usually have two or more availability zones for fault tolerance. The AZs are isolated locations within cloud data center regions that public cloud providers originate and operate. Each availability zone is a physically separate data center with its own redundant power and network connections.

Written Lab

Fill in the blanks for the questions provided in the written lab. You can find the answers to the written labs in Appendix B.

1. With the _____ as a Service model, the cloud provider owns and manages all levels of the computing environment.
2. With the _____ as a Service model, the cloud provider owns and manages the computing hardware but not the operating systems or the applications.
3. With the _____ as a Service model, the cloud provider owns and manages the hardware and operating system but not the application software.
4. _____ refers to the ability to access the cloud resources from anywhere in the network from a variety of devices such as laptops, tables, smartphones, and thin or thick clients.
5. _____ is the ability to take physical data center resources such as RAM, CPU, storage, and networking and create a software representation of those resources that enables large-scale cloud offerings.
6. Private, low-latency network interconnectivity between your corporate data center and your cloud operations is accomplished using _____.
7. _____ is the transfer and synchronization of data between computing or storage resources.
8. _____ addresses the issues found when cloud workloads and connections increase to the point where a single server can no longer handle the workload by spreading the workload across multiple cloud computing resources.
9. Common remote access protocols used to manage servers in the cloud include _____ and _____.
10. Establishing _____ helps you determine how to size your cloud resources.

Review Questions

The following questions are designed to test your understanding of this chapter's material. You can find the answers in Appendix A. For more information on how to obtain additional questions, please see this book's Introduction.

1. What cloud model gives you complete control of the operating system?
   - **A.** IaaS
   - **B.** PaaS
   - **C.** SaaS
   - **D.** CaaS
2. A cloud service provider allocates resources into a group. These resources are then dynamically allocated and reallocated as the demand requires. What is this referred to as?
   - **A.** On-demand virtualization
   - **B.** Dynamic scaling
   - **C.** Resource pooling
   - **D.** Elasticity
3. What are three examples of IaaS elements you can provision in the cloud? (Choose three.)
   - **A.** CPU
   - **B.** OS ACLs
   - **C.** Memory
   - **D.** Storage
   - **E.** Scalability
   - **F.** SSH
4. Which of the following is not a valid pooled resource?
   - **A.** Memory
   - **B.** Storage
   - **C.** Security
   - **D.** Networking
   - **E.** CPU
5. What technologies are used to enable on-demand computing? (Choose two.)
   - **A.** Load balancing
   - **B.** Automation
   - **C.** Autoscaling groups
   - **D.** Virtualization

6. When you migrate your operations to the cloud and you decide to match computing resources with your current requirements, what can you take advantage of to expand your compute capacity in the future? (Choose three.)
   - **A.** Elasticity
   - **B.** On-demand computing
   - **C.** Availability zones
   - **D.** Resiliency virtualization
   - **E.** Pay as you grow
   - **F.** Regions

7. Your company has decided to use one cloud provider for a production application while using a different cloud provider for storing backups. What type of cloud delivery model is this?
   - **A.** Public
   - **B.** Hybrid
   - **C.** Community
   - **D.** Private

8. In a traditional virtualized data center, what shared network resource do VMs on the same host use to communicate with each other?
   - **A.** Virtual NIC
   - **B.** Region
   - **C.** Virtual switch
   - **D.** LAN

9. Which cloud characteristic allows you to pay for only the services used?
   - **A.** Bursting
   - **B.** Pay as you grow
   - **C.** Chargeback
   - **D.** Autoscaling

10. When migrating an application from the data center to the cloud, which of the following is a best practice?
    - **A.** Deploy to a test environment to validate functionality and performance.
    - **B.** Deploy directly to production so that end users can immediately report any problems.
    - **C.** Clone the VM running the application and upload it to the cloud.
    - **D.** Copy the application files to a fresh VM running in the cloud.

11. Which cloud characteristic allows you to access a self-service portal to create additional servers, storage, or other services instantly?

A. Bursting

B. Pay as you grow

C. Multitenancy

D. On-demand

12. In which cloud service model does the provider handle everything up to and including the application?

A. IaaS

B. PaaS

C. SaaS

D. ZaaS

13. Which of the following metrics can you typically monitor in the cloud? (Choose two.)

A. Network latency

B. Physical CPU host utilization

C. The number of available physical virtualization hosts

D. Inter-availability zone latency

E. Storage I/O operations per second

14. Cloud service providers will often segment their operations to allow for resiliency, geographic proximity, and data protection regulations. What are these geographical segmentations referred to as?

A. Regions

B. Autoscaling groups

C. Availability zones

D. Global DNS affinity

15. What are critical steps to take prior to performing a migration to the cloud? (Choose three.)

A. Baselines

B. Capacity requirements

C. Variance measurements

D. Documentation

E. Automation rollout

16. Cloud operations are the responsibility of both your organization and the cloud service provider. What is this model called?

A. Availability zone model

B. Community model

C. Shared responsibility model

D. Shared regional model

17. What is the process of testing your cloud access to determine whether there is any vulnerability that an attacker could exploit?
    - **A.** Elasticity
    - **B.** Vulnerability testing
    - **C.** Penetration testing
    - **D.** Load testing

18. A hospital wants to use a medical records application but must minimize its capital expenditure, minimize ongoing maintenance, and comply with various regulations. What type of cloud service model and cloud delivery model would you recommend that they use? (Choose two.)
    - **A.** Public
    - **B.** SaaS
    - **C.** Community
    - **D.** Private
    - **E.** IaaS

19. What systems do cloud providers implement for rapid deployment of customer-requested services?
    - **A.** RDMS
    - **B.** Orchestration
    - **C.** On-demand provisions
    - **D.** Service catalogs

20. In which cloud service model does the provider manage everything except the application?
    - **A.** IaaS
    - **B.** PaaS
    - **C.** SaaS
    - **D.** CaaS